FOUNDATION OF DATA SCIENCE - UNIT I - INTRODUCTION

Data Science: Benefits and uses – facets of data - Data Science Process: Overview – Defining research goals – Retrieving data – Data preparation - Exploratory Data analysis – Build the model– presenting findings and building applications – Data Mining - Data Warehousing – Basic Statistical descriptions of Data

Data
In computing, data is information that has been translated into a form that is efficient for movement or processing

Data Science
Data science is an evolution ary extension of statistics capable of dealing with the massive amounts of data produced today. It adds methods from computer science to the repertoire of statistics.

Benefits and uses of data science
Data science and big data are used almost everywhere in both commercial and non commercial Settings
- Commercial companies in almost every industry use data science and big data to gain insights into their customers, processes, staff, completion, and products.
- Many companies use data science to offer customers a better user experience, as well as to cross-sell, up -sell, and personalize their of ferings.
- Governmental organizations are also aware of data's value. Many governmental organizations not only rely on internal data scientists to discover valuable information, but also share their data with the public.
- Nongovernmental organizations (NGOs) use it to raise money and defend their causes.
- Universities use data science in their research but also to enhance the study experience of their students. The rise of massive open online courses (MOOC) produces a lot of data, which allow sun iversities to study how this type of learning can complement traditional classes.

Facets of data
In data science and big data you'll come across many different types of data, and each of them tends to require different tools and techniques. The main categories of data are these:
- Structured
- Unstructured
- Natural language
- Machine-generated
- Graph-based
- Audio, video, and images
- Streaming

Let's explore all these interesting data types.

Structured data
- Structured data is data that depends on a data model and resides in a fixed field within a record. As such, it's often easy to store structured data in tables with in data bases or Excel files
- SQL, or Structured Query Language, is the preferred way to manage and query data that resides in data bases.

Indicator ID	Dimension List	Timeframe	Numeric Value	Missing Value Flag	Confidence Inte
214390830	Total (Age-adjusted)	2008	74.6%		73.8%
214390833	Aged 18-44 years	2008	59.4%		58.0%
214390831	Aged 18-24 years	2008	37.4%		34.6%
214390832	Aged 25-44 years	2008	66.9%		65.5%
214390836	Aged 45-64 years	2008	88.6%		87.7%
214390834	Aged 45-54 years	2008	86.3%		85.1%
214390835	Aged 55-64 years	2008	91.5%		90.4%
214390840	Aged 65 years and over	2008	94.6%		93.8%
214390837	Aged 65-74 years	2008	93.6%		92.4%
214390838	Aged 75-84 years	2008	95.6%		94.4%
214390839	Aged 85 years and over	2008	96.0%		94.0%
214390841	Male (Age-adjusted)	2008	72.2%		71.1%
214390842	Female (Age-adjusted)	2008	76.8%		75.9%
214390843	White only (Age-adjusted)	2008	73.8%		72.9%
214390844	Black or African American only (Age-adjusted)	2008	77.0%		75.0%
214390845	American Indian or Alaska Native only (Age-adjusted)	2008	66.5%		57.1%
214390846	Asian only (Age-adjusted)	2008	80.5%		77.7%
214390847	Native Hawaiian or Other Pacific Islander only (Age-adjusted)	2008	DSU		
214390848	2 or more races (Age-adjusted)	2008	75.6%		69.6%

Unstructured data

Unstructured data is data that isn't easy to fit into a data model because the content is context-specific or varying. One example of unstructured data is your regular email
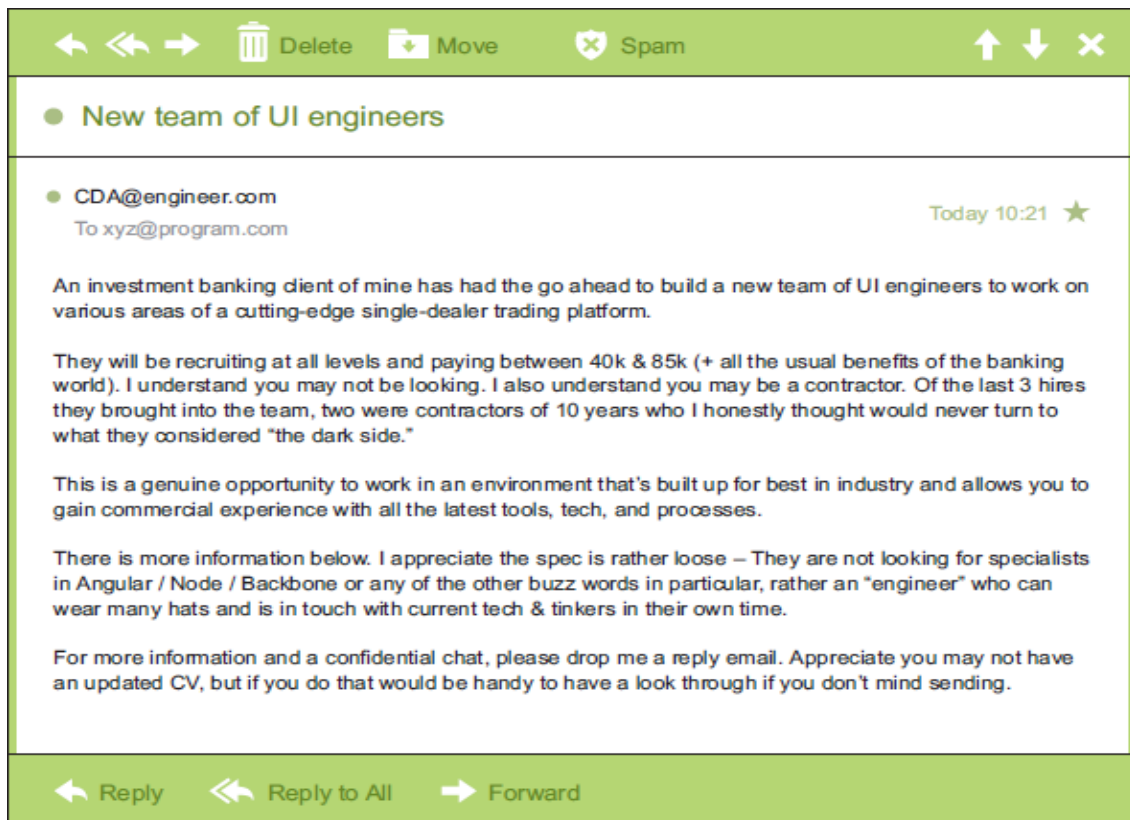
Natural language

- Natural language is a special type of unstructured data; it's challenging to process because it requires knowledge of specific data science techniques and linguistics.

- The natural language processing community has had success in entity recognition, topic recognition, summarization, text completion, and sentiment analysis, but models trained in one domain don't generalize well to other domains.
- Even state-of-the-art techniques aren't able to decipher the meaning of every piece of text.

Machine-generated data

- Machine-generated data is information that's automatically created by a computer, process, application, or other machine without human intervention.
- Machine-generated data is becoming a major data resource and will continue to do so.
- The analysis of machine data relies on highly scalable tools, due to its high volume and speed. Examples of machine data are web server logs, called tail records, network event logs, and telemetry.

```
CSIPERF:TXCOMMIT;313236
2014-11-28 11:36:13, Info            CSI     00000153 Creating NT transaction (seq
69), objectname [6]"(null)"
2014-11-28 11:36:13, Info            CSI     00000154 Created NT transaction (seq 69)
result 0x00000000, handle @0x4e54
2014-11-28 11:36:13, Info            CSI     00000155@2014/11/28:10:36:13.471
Beginning NT transaction commit...
2014-11-28 11:36:13, Info            CSI     00000156@2014/11/28:10:36:13.705 CSI perf
```

Graph-based or network data

- "Graph data" can be a confusing term because any data can be show nina graph.
- Graphornetworkdatais,in short, datathat focuseson therelationshipor adjacencyof objects.
- Thegraphstructuresuse nodes,edges,andpropertiestorepresentandstoregraphicaldata.
- Graph-baseddataisanaturalwaytorepresentsocialnetworks,anditsstructureallows you tocalculatespecificmetricssuchasthe influenceofapersonand the shortestpath betweentwo people.
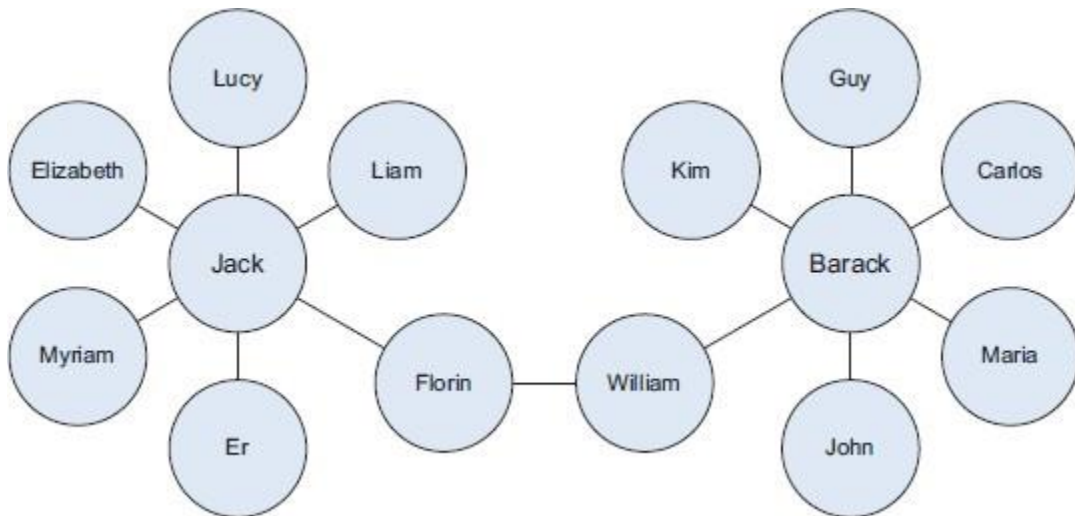
Audio,image, andvideo

- Audio,image,andvideo aredatatypesthat posespecificchallengestoadatascientist.
- Tasksthataretrivialforhumans, suchasrecognizing objectsinpictures,turnouttobe challengingforcomputers.
- MLBAM(MajorLeagueBaseballAdvancedMedia)announcedin2014thatthey'llincreasevideocaptureto approximately7 TBpergame for thepurposeoflive,in-gameanalytics.
- RecentlyacompanycalledDeepMindsucceededatcreatinganalgorithmthat'scapableoflearninghow to playvideogames.
- Thisalgorithmtakesthevideoscreenasinputandlearnstointerpreteverythingviaacomplexofdeep learning.

Streamingdata
- Thedataflowsintothesystemwhenaneventhappensinsteadofbeingloadedintoadatastoreinabatch.
- Examplesarethe"What'strending"onTwitter,livesportingormusicevents,andthestockmarket.

DataScienceProcess

Overviewofthedatascienceprocess

Thetypicaldatascienceprocessconsistsofsix stepsthrough whichyou'll iterate,asshowninfigure
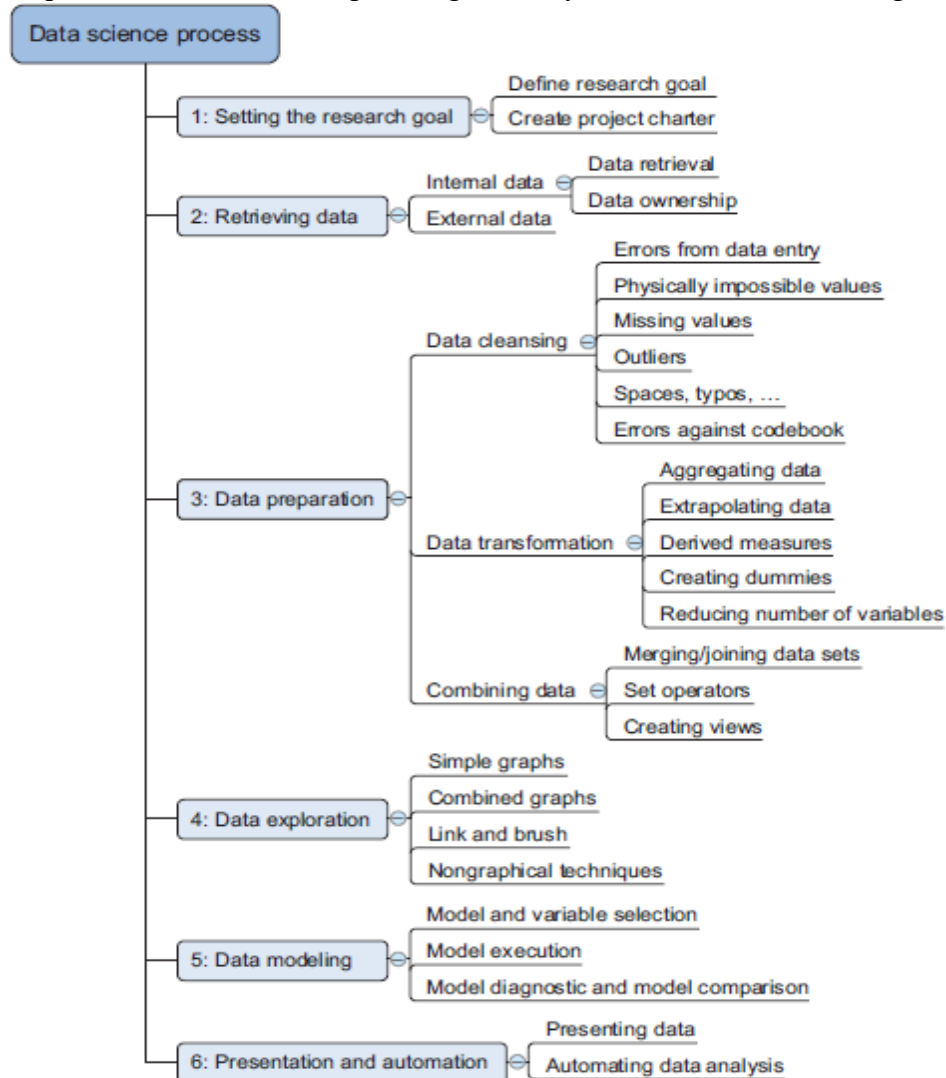
1. The first step of this process is setting a research goal. The main purpose here is making sure all thestakeholders understand the what, how, and why of the project. In every serious project this will resultina project charter.
2. The second phase is data retrieval. You want to have data available for analysis, so this step includesfinding suitable data and getting access to the data from the data owner. The result is data in its rawform,which probablyneeds polishingand transformation before it becomes usable.
3. Now that you have the raw data, it's time to prepare it. This includes transforming the data from a rawform into data that's directly usable in your models. To achieve this, you'll detect and correct differentkinds of errors in the data, combine data from different data sources, and transform it. If you havesuccessfullycompleted this step,youcan progress to data visualizationand modeling.

4. The fourth step is data exploration. The goal of this step is to gain a deep understanding of the data.You'll look for patterns, correlations, and deviations based on visual and descriptive techniques. Theinsightsyougain fromthis phasewillenableyou to start modeling.
5. Finally, we get to model building (often referred to as "data modeling" throughout this book). It is nowthat you attempt to gain the insights or make the predictions stated in your project charter. Now is thetime to bring out the heavy guns, but remember research has taught us that often (but not always) acombination of simple models tends to outperform one complicated model. If you've done this phaseright,you'realmost done.
6. The last step of the data science model is presenting your results and automating the analysis, ifneeded. One goal of a project is to change a process and/or make better decisions. You may still needtoconvincethebusinessthatyourfindingswillindeedchangethebusinessprocessasexpected.Thisis where you can shine in your influencer role. The importance of this step is more apparent in projectson a strategic and tactical level. Certain projects require you to perform the business process over andoveragain, soautomatingthe project willsavetime.

Definingresearchgoals

A project starts by understanding the *what*, the *why*, and the *how* of your project. The outcome should be aclear research goal, a good understanding of the context, well-defined deliverables, and a plan of action with atimetable.This informationis then best placedinaproject charter.

Spendtimeunderstandingthegoalsandcontextof yourresearch

- An essential outcome is the research goal that states the purpose of your assignment in a clear andfocusedmanner.
- Understandingthebusinessgoalsand contextiscriticalforprojectsuccess.
- Continue asking questions and devising examples until you grasp the exact business expectations,identify howyour project fits in the bigger picture, appreciate howyour research is going to changethebusiness,and understand how they'lluseyourresults

Createaprojectcharter

Aprojectcharterrequiresteamwork,andyourinputcoversatleastthefollowing:

- Aclearresearchgoal
- Theprojectmission and context
- Howyou'regoingtoperformyouranalysis
- Whatresourcesyou expecttouse
- Proofthatit's anachievable project,orproof ofconcepts
- Deliverablesandameasureofsuccess
- Atimeline

Retrievingdata

- Thenextstepindatascienceistoretrievetherequireddata.Sometimesyouneedtogointothefieldanddesignadata collectionprocessyourself,butmostofthetime youwon'tbeinvolvedinthisstep.
- Manycompanieswillhave alreadycollectedandstoredthe data foryou,andwhattheydon't havecanoftenbebought from third parties.
- Moreandmoreorganizationsaremakingevenhigh-qualitydatafreelyavailableforpublicandcommercialuse.
- Datacanbestoredinmanyforms,rangingfromsimpletextfilestotablesinadatabase.Theobjectivenowis acquiringall the datayou need.

Startwithdatastoredwithinthecompany(Internaldata)

- Most companies have a program for maintaining key data, so much of the cleaning work may alreadybe done. This data can be stored in official data repositories such as databases, data marts, datawarehouses,anddatalakes maintained byateam ofIT professionals.
- Data warehouses and data marts are home to preprocessed data, data lakes contain data in its natural orrawformat.
- Finding data even within your own company can sometimes be a challenge. As companies grow, theirdata becomes scattered around many places. the data may be dispersed as people change positions andleavethecompany.
- Getting access to data is another difficult task. Organizations understand the value and sensitivity ofdataand oftenhavepolicies inplacesoeveryonehas accessto what theyneed andnothingmore.
- These policies translate into physical and digital barriers called *Chinese walls*. These "walls" aremandatoryand well-regulated forcustomer data inmost countries.

ExternalData
- If data isn't available inside your organization, look outside your organizations. Companies providedata so that you, in turn, can enrich their services and ecosystem. Such is the case with Twitter,LinkedIn,andFacebook.
- Moreandmore governmentsand organizationssharetheirdatafor freewiththeworld.
- Alistofopendataproviders thatshouldgetyoustarted.

Open data site	Description
Data.gov	The home of the US Government's open data
https://open-data.europa.eu/	The home of the European Commission's open data
Freebase.org	An open database that retrieves its information from sites like Wikipedia, MusicBrains, and the SEC archive
Data.worldbank.org	Open data initiative from the World Bank
Aiddata.org	Open data for international development
Open.fda.gov	Open data from the US Food and Drug Administration

DataPreparation(Cleansing,Integrating,TransformingData)

Your model needs the data in a specific format, so data transformation will always come into play. It's a goodhabit to correct data errors as early on in the process as possible. However, this isn't always possible in arealisticsetting, soyou'llneedto take correctiveactions inyour program.

Cleansing data

Datacleansingisasubprocessofthedatascienceprocessthatfocusesonremovingerrorsinyourdatasoyourdata becomes atrueand consistent representationoftheprocessesit originatesfrom.
- Thefirsttypeistheinterpretationerror,such aswhenyoutake the valueinyourdatafotgranted,likesayingthat aperson'sageis greater than 300years.
- Thesecondtypeoferrorpointstoinconsistenciesbetweendatasourcesoragainstyourcompany'sstandardized values.

Anexampleofthisclassoferrorsisputting"Female"inonetableand"F"inanotherwhentheyrepresentthesame thing: that the person is female.

Overviewofcommonerrors

General solution	
Try to fix the problem early in the data acquisition chain or else fix it in the program.	
Error description	**Possible solution**
Errors pointing to false values within one data set	
Mistakes during data entry	Manual overrules
Redundant white space	Use string functions
Impossible values	Manual overrules
Missing values	Remove observation or value
Outliers	Validate and, if erroneous, treat as missing value (remove or insert)
Errors pointing to inconsistencies between data sets	
Deviations from a code book	Match on keys or else use manual overrules
Different units of measurement	Recalculate
Different levels of aggregation	Bring to same level of measurement by aggregation or extrapolation

Sometimes you'll use more advanced methods, such as simple modeling, to find and identify data errors;diagnostic plots can be especially insightful. For example, in figurewe use a measure to identify data pointsthat seem out of place. We do a regression to get acquainted with the data and detect the influence ofindividualobservations on theregression line.

A single outlier can throw off a regression estimate.

Regression line influenced by outlier

Normal regression line
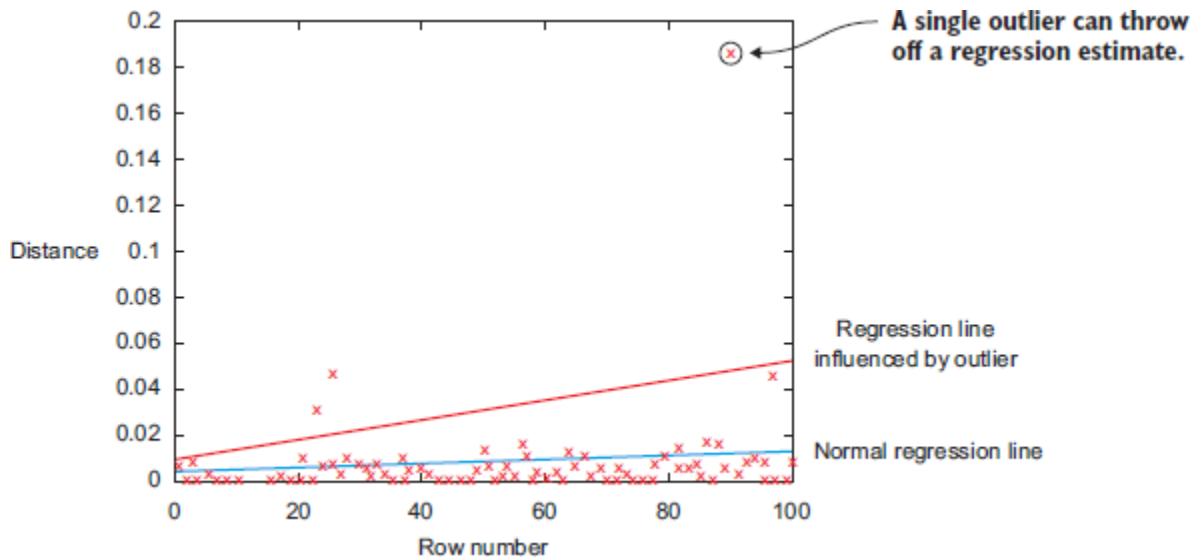
The encircled point influences the model heavily and is worth investigating because it can point to a region where you don't have enough data or might indicate an error in the data, but it also can be a valid data point.

DataEntryErrors
- Datacollectionanddataentryareerror-proneprocesses.Theyoftenrequirehumanintervention,andintroducean errorinto thechain.

- Datacollectedbymachinesorcomputersisn'tfreefromerrors.Errorscanarisefromhumansloppiness,whereas others aredueto machine orhardwarefailure.
- Detecting data errors when the variables you study don't have many classes can be done by tabulatingthedatawith counts.
- Whenyou have a variable that can take only two values: "Good" and"Bad",you can create afrequency table and see if those are truly the only two values present. In tablethe values "Godo" and"Bade"point out somethingwent wrongin at least16cases.

Value	Count
Good	1598647
Bad	1354468
Godo	15
Bade	1

Most errors of this type are easy to fix with simple assignment statements and if-thenelserules:

```
ifx=="Godo":
x ="Good"
ifx =="Bade":
x ="Bad"
```

RedundantWhitespace
- Whitespacestendtobehardtodetectbutcause errorslikeotherredundantcharacterswould.
- The whitespace cause the miss match in the string such as "FR " – "FR", dropping the observationsthatcouldn't bematched.
- If you know to watch out for them, fixing redundant whitespaces is luckily easy enough in mostprogramming languages. They all provide string functions that will remove the leading and trailingwhitespaces. For instance, in Python you can use the strip() function to remove leading and trailingspaces.

FixingCapitalLetterMismatches
Capitallettermismatchesarecommon.Mostprogramminglanguagesmakeadistinctionbetween"Brazil"and"brazil".
Inthiscaseyoucansolvetheproblembyapplyingafunctionthatreturnsbothstringsinlowercase,suchas .lower()inPython."Brazil".lower()=="brazil".lower()shouldresultin true.

ImpossibleValuesandSanityChecks
Hereyoucheckthevalueagainstphysicallyortheoreticallyimpossible valuessuchaspeopletallerthan3metersorsomeone with an ageof299years. Sanitychecks can bedirectlyexpressedwith rules:
check=0<= age <=120

Outliers
An outlier is an observation that seems to be distant from other observations or, more specifically, oneobservation that follows a different logic or generative process than the other observations. The easiest way tofindoutliers is to useaplot oratablewiththe minimum and maximumvalues.
The plot on the top shows no outliers, whereas the plot on the bottom shows possibleoutliers on the uppersidewhen anormal distribution is expected.

Distribution with outliers
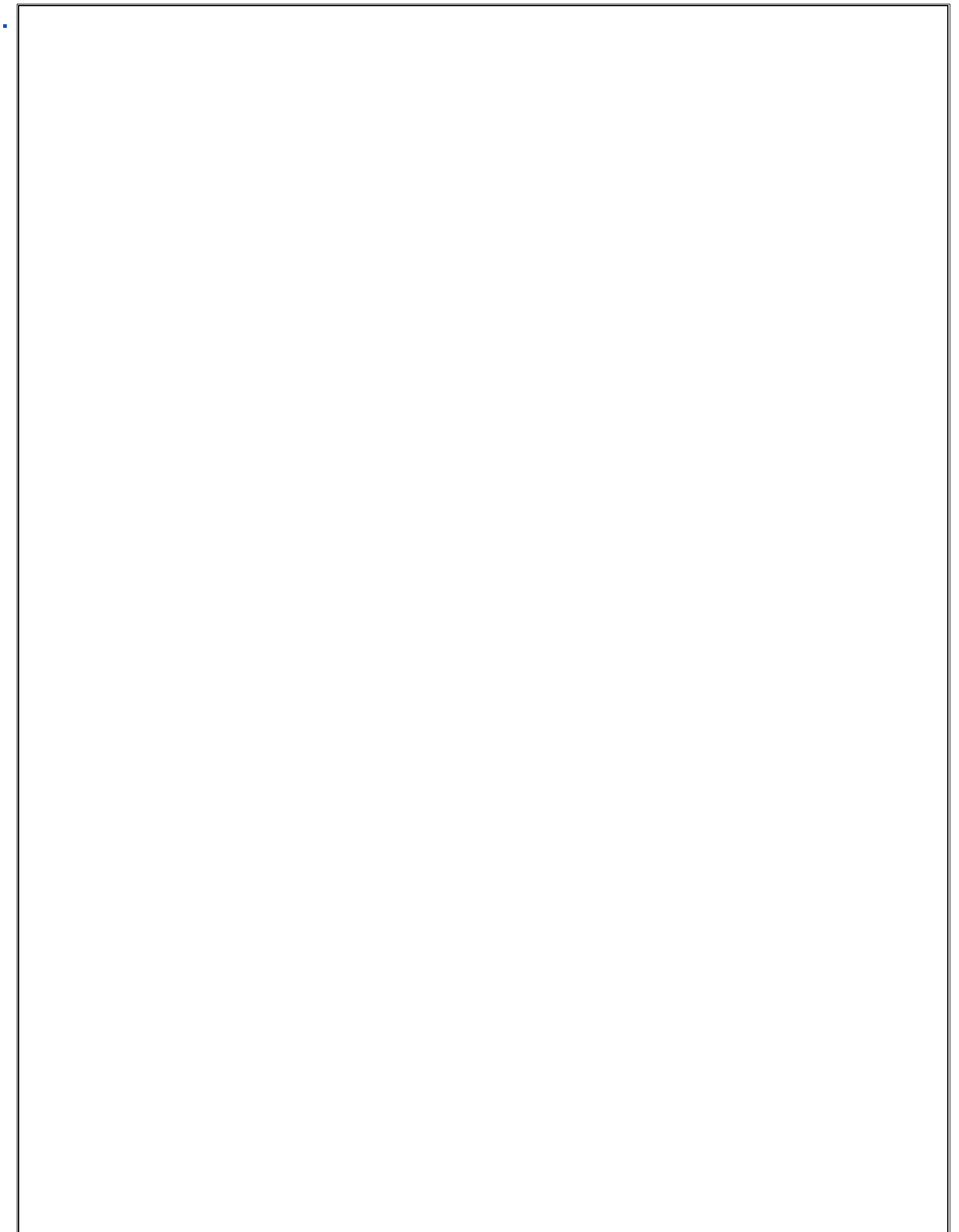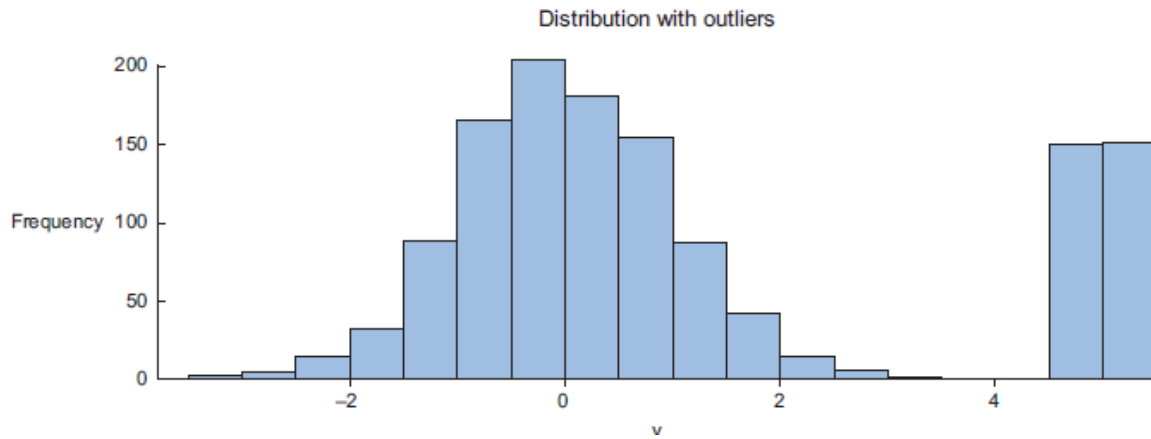
DealingwithMissingValues

Missing values aren't necessarily wrong, but you still need to handle them separately;certain modelingtechniques can't handle missing values. They might be an indicator that something went wrong in your datacollection or that an error happened in the ETL process. Common techniques data scientists use are listed intable

Integratingdata

Your data comes from several different places, and in this substep we focus on integrating these differentsources.Datavariesinsize,type, andstructure,rangingfromdatabasesandExcelfilestotextdocuments.

TheDifferentWaysof CombiningData

Youcanperformtwo operationstocombineinformation fromdifferentdatasets.

- Joining
- Appendingorstacking

JoiningTables

- Joiningtablesallowsyoutocombinetheinformationofoneobservationfoundinonetablewiththeinformationt hatyoufindin anothertable. Thefocus is onenrichinga singleobservation.
- Let'ssaythatthefirsttablecontainsinformationaboutthepurchasesofacustomerandtheothertablecontainsinf ormation about theregion whereyourcustomer lives.
- Joiningthetablesallowsyoutocombinetheinformationsothatyoucanuseitforyourmodel,asshownin figure.
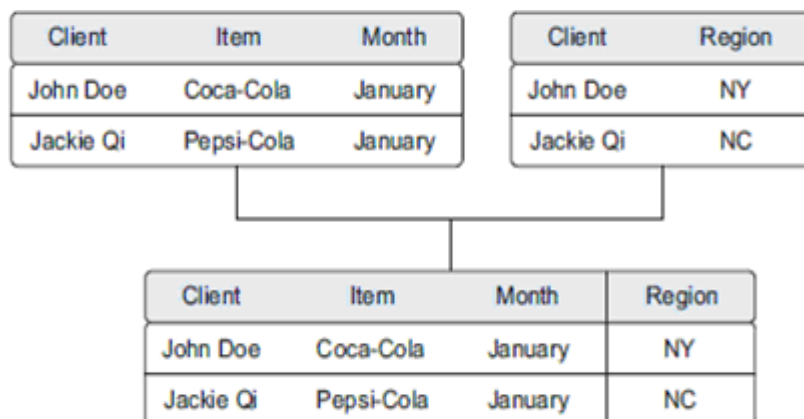

Figure. Joining two tables on the item and region

key9

To join tables, you use variables that represent the same object in both tables, such as a date, a country name,or a Social Security number. These common fields are known as keys. When these keys also uniquely definetherecords in the table theyare called*primarykeys*.

Thenumberofresultingrowsin theoutputtabledependsontheexactjointypethatyouuse

AppendingTables
- Appendingor stackingtables iseffectivelyadding observations from onetable toanothertable.
- OnetablecontainstheobservationsfromthemonthJanuaryandthesecondtablecontainsobservations   from the month February. The result of appending these tables is a larger one with theobservationsfrom Januaryaswell as February.
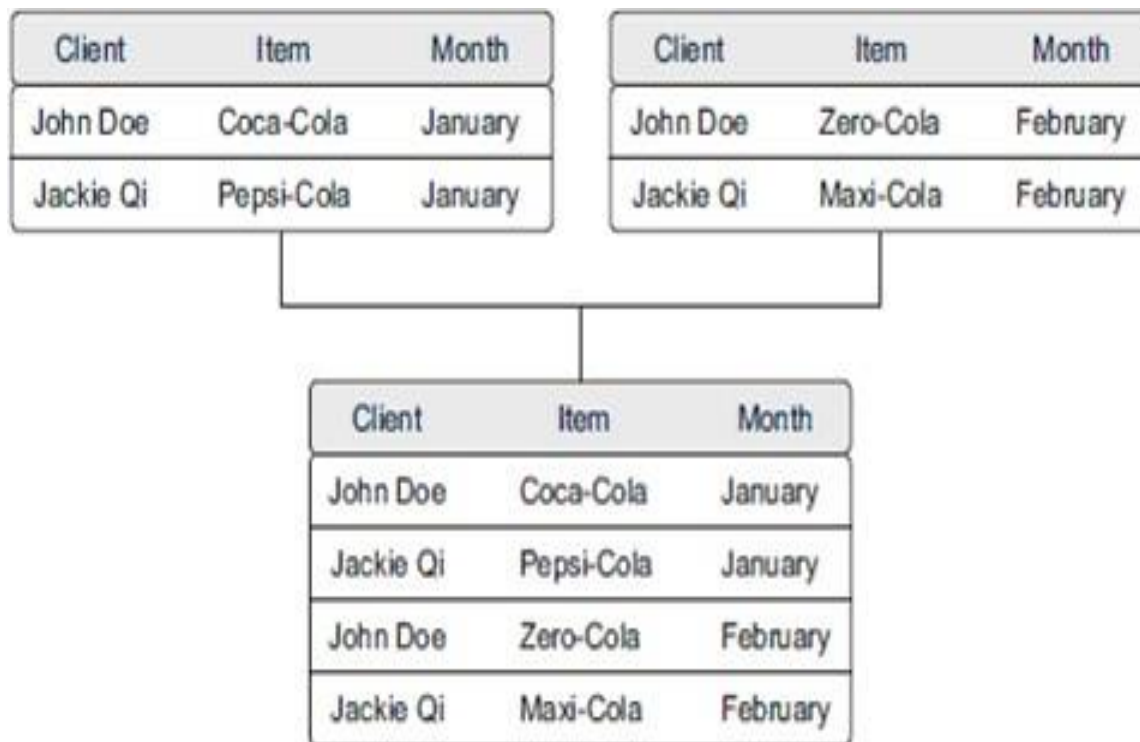
Client	Item	Month
John Doe	Coca-Cola	January
Jackie Qi	Pepsi-Cola	January

Client	Item	Month
John Doe	Zero-Cola	February
Jackie Qi	Maxi-Cola	February

Client	Item	Month
John Doe	Coca-Cola	January
Jackie Qi	Pepsi-Cola	January
John Doe	Zero-Cola	February
Jackie Qi	Maxi-Cola	February

Figure. Appending data from tables is a common operation but requires an equal structure in the tables beginappended,

Transformingdata

Certain models require their data to be in a certain shape. Transforming your data so it takes a suitable formfordata modeling.

Relationships between an input variable and an output variable aren't always linear. Take, for instance, arelationshipof the form$y = ae^{bx}$. Taking the log of the independentvariables simplifies the estimationproblem dramatically. Transforming the input variables greatly simplifies the estimation problem. Other timesyoumight want to combinetwo variables into anew variable.

x	1	2	3	4	5	6	7	8	9	10
log(x)	0.00	0.43	0.68	0.86	1.00	1.11	1.21	1.29	1.37	1.43
y	0.00	0.44	0.69	0.87	1.02	1.11	1.24	1.32	1.38	1.46
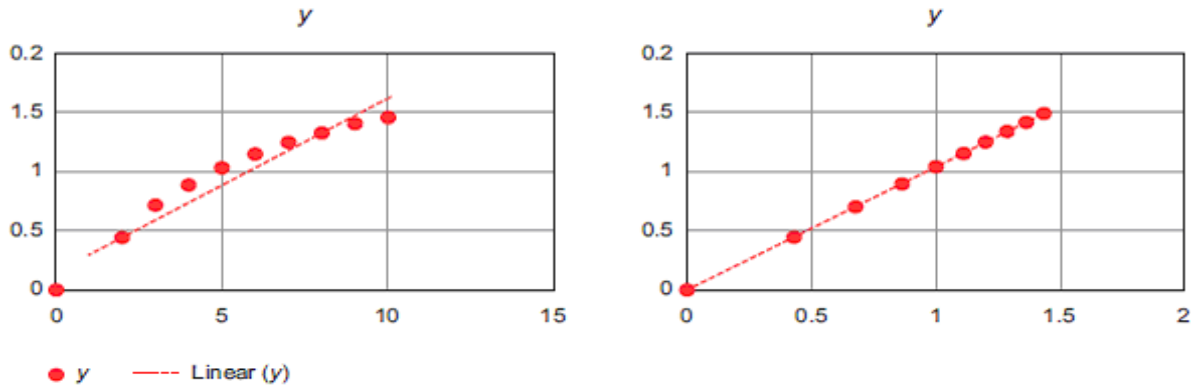
Transforming x to log x makes the relationship between x and y linear (right), compared with the non-log x (left).

ReducingtheNumberof Variables

- Having too many variables in your model makes the model difficult to handle, and certain techniquesdon't perform well when you overload them with too many input variables. For instance, all thetechniquesbased onaEuclidean distanceperformwell onlyup to 10 variables.
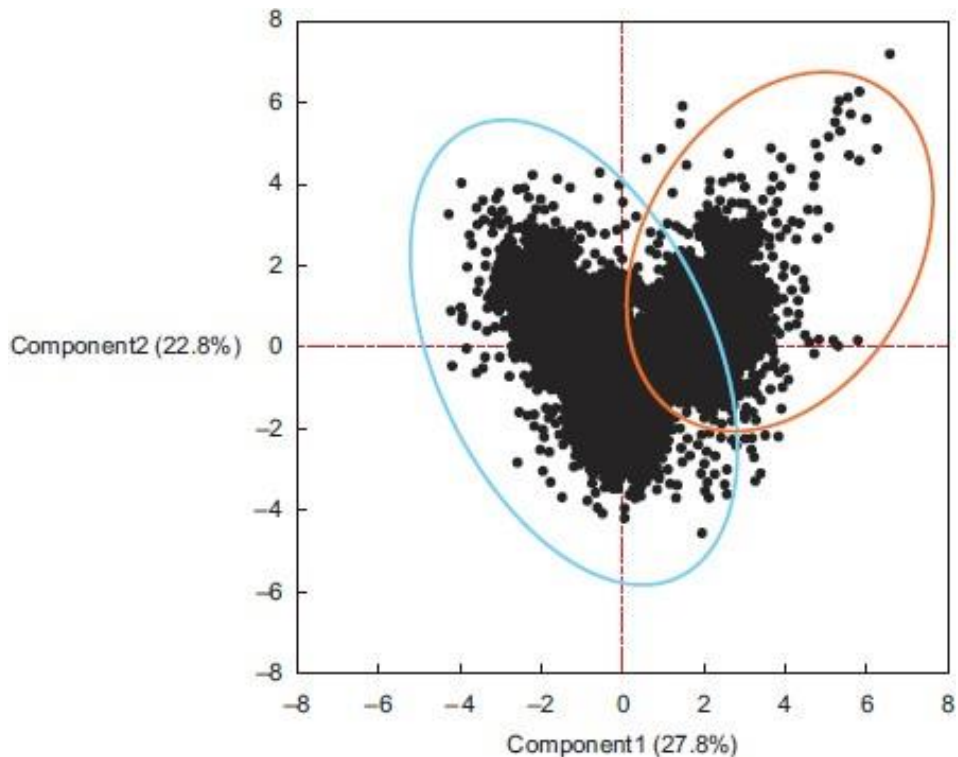- Data scientists use special methods to reduce the number of variables but retain the maximum amountofdata.

Figure shows how reducing the number of variables makes it easier to understand thekey values.It alsoshows how two variables account for 50.6% of the variation within the data set (component1 = 27.8% +component2 = 22.8%). These variables, called"component1" and "component2," are both combinations oftheoriginal variables.They'retheprincipal componentsofthe underlyingdata structure

TurningVariablesintoDummies

- *Dummyvariables*canonlytaketwovalues:true(1) orfalse(0).They're usedtoindicatetheabsenceofacategorical effect thatmayexplain the observation.
- Inthiscaseyou'llmakeseparatecolumnsfortheclassesstoredinone variable andindicate itwith1iftheclass is present and 0otherwise.
- AnexampleisturningonecolumnnamedWeekdaysintothecolumnsMondaythroughSunday.Youuseanindi catorto showiftheobservation wasonaMonday;you put1on Mondayand0 elsewhere.
- Turningvariablesintodummiesisatechniquethat'susedinmodelingandispopularwith,butnotexclusiveto, economists.

Customer	Year	Gender	Sales
1	2015	F	10
2	2015	M	8
1	2016	F	11
3	2016	M	12
4	2017	F	14
3	2017	M	13

M   F

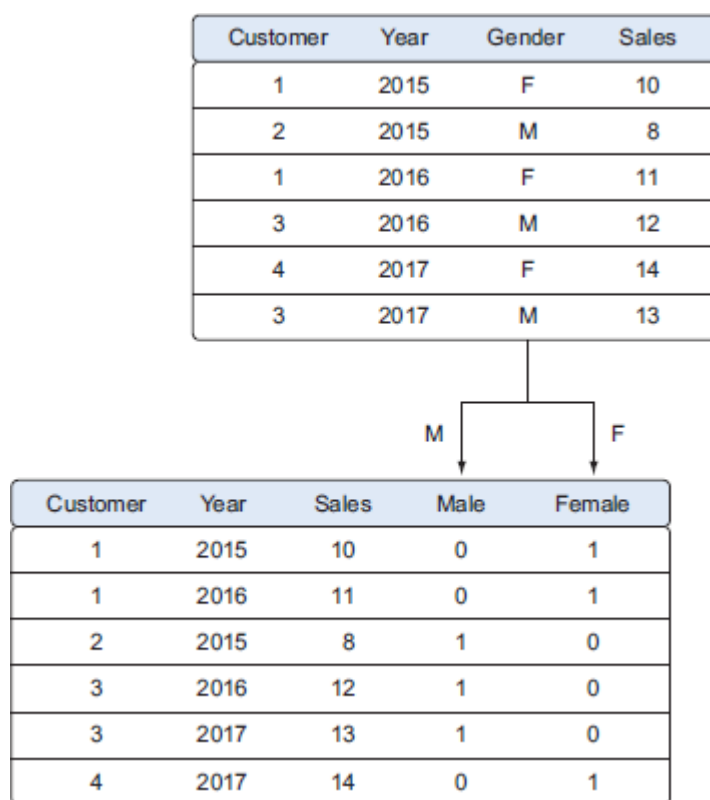| Customer	Year	Sales	Male	Female
1	2015	10	0	1
1	2016	11	0	1
2	2015	8	1	0
3	2016	12	1	0
3	2017	13	1	0
4	2017	14	0	1

Figure. Turning variables into dummies is a data transformation that breaks a variable that has multipleclasses into multiplevariables, each havingonlytwo possiblevalues: 0 or1

Exploratorydataanalysis

During exploratory data analysis you take a deep dive into the data (see figure below). Informationbecomes much easier to grasp when shown in a picture, therefore you mainly use graphical techniques togainan understandingofyourdata and the interactionsbetweenvariables.
The goal isn't to cleanse the data, but it's common that you'll still discover anomalies you missed before,forcingyou to takeastepback and fixthem.
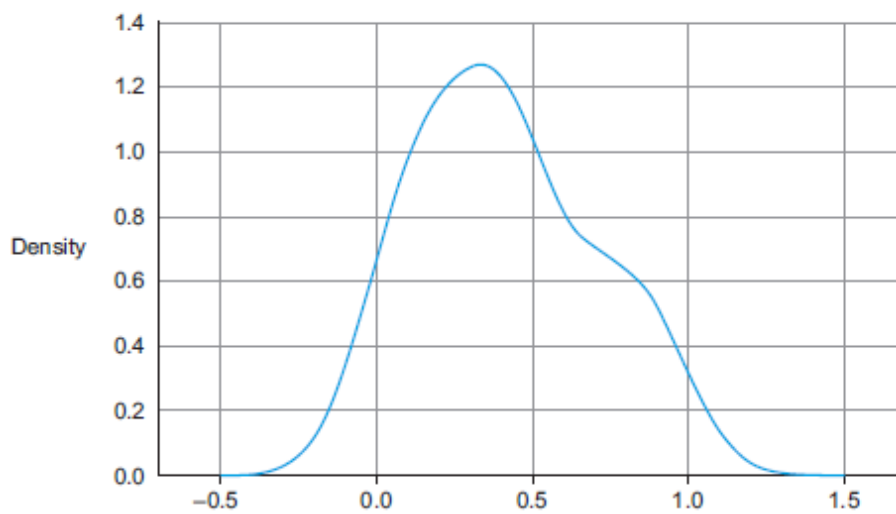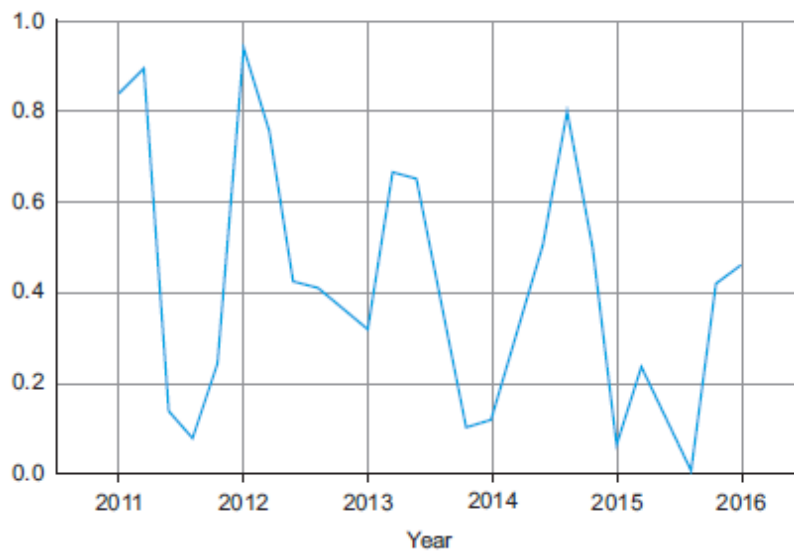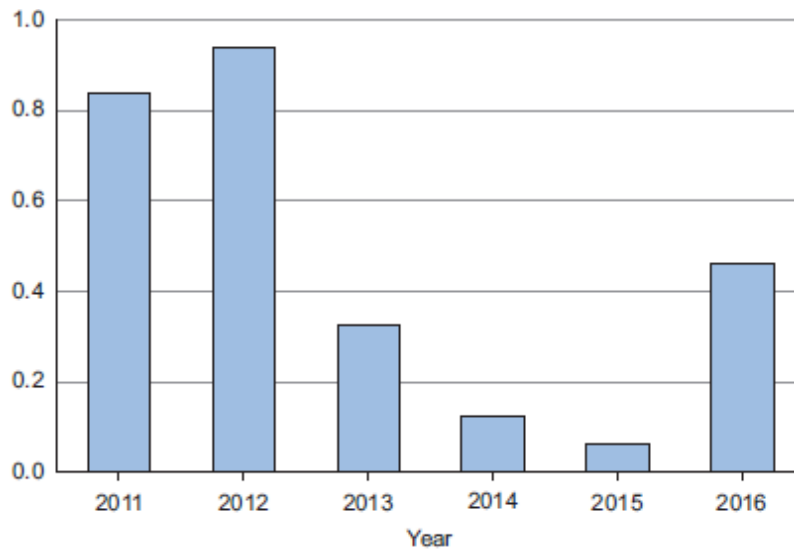
From top to bottom, a bar chart, a line plot, and a distribution are some of the graphs used in exploratory analysis.

- The visualization techniques you use in this phase range from simple line graphs or histograms, asshownin below figure,to morecomplexdiagrams such as Sankeyandnetworkgraphs.
- Sometimes it's useful to compose a composite graph from simple graphs to get even more insightinto the data   Other times the graphs can be animated or made interactive to make it easier and,let's admit it, waymorefun
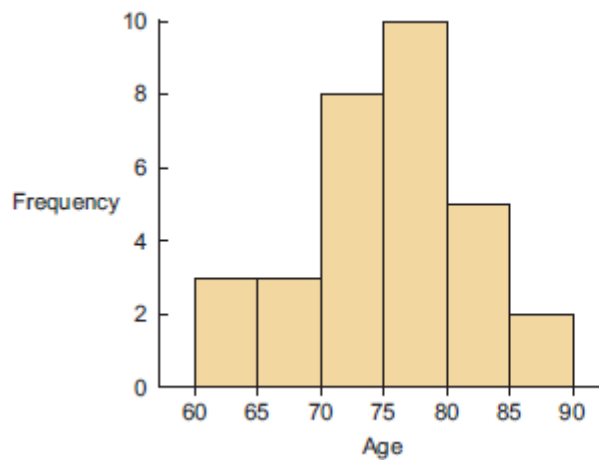
Example histogram: the number of people in the age-groups of 5-year intervals

The techniques we described in this phase are mainly visual, but in practice they're certainly not limited tovisualizationtechniques.Tabulation,clustering,andothermodelingtechniquescanalsobeapartofexploratoryanalysis. Even buildingsimplemodels can beapart ofthis step.

Buildthemodels

- With clean data in place and a good understanding of the content,you're ready to build modelswiththe goal of making better predictions, classifying objects, or gaining an understanding of the systemthatyou'remodeling.
- This phase is much more focused than the exploratory analysis step, because you know what you'relookingforand whatyouwant theoutcometobe.

Building a model is an iterative process. The way you build your model depends on whether you go withclassic statistics or the somewhat more recent machine learning school, and the type of technique you want touse.Eitherway, most models consist ofthe followingmain steps:
- Selectionofamodelingtechnique andvariables toenterin themodel
- Executionofthemodel
- Diagnosisandmodelcomparison

Modelandvariableselection
You'llneedtoselectthevariablesyouwanttoincludeinyourmodelandamodelingtechnique.You'llneedto   consider model performance and whetheryour project meets allthe requirements to useyour model, aswellas other factors:
- Mustthe model be movedto a production environmentand, if so, wouldit be easytoimplement?
- Howdifficultis themaintenanceonthemodel:how longwill itremain relevantif leftuntouched?
- Doesthe model need to beeasyto explain?

Modelexecution
- Onceyou'vechosen amodelyou'llneedtoimplementitincode.

- Most programming languages, such as Python, already have libraries such as StatsModels or Scikit-learn.Thesepackages useseveral of themost populartechniques.
- Coding a model is a nontrivial task in most cases, so having these libraries available can speed up theprocess.Asyoucanseeinthefollowingcode,it'sfairlyeasytouselinearregressionwithStatsModelsor Scikit-learn
- Doing this yourself would require much more effort even for the simple techniques. The followinglistingshows the execution of alinear predictionmodel.

```python
import statsmodels.api as sm          Imports required
import numpy as np                     Python modules.
predictors = np.random.random(1000).reshape(500,2)
target = predictors.dot(np.array([0.4, 0.6])) + np.random.random(500)
lmRegModel = sm.OLS(target,predictors)
result = lmRegModel.fit()              Fits linear        Creates random data for
result.summary()                       regression         predictors (x-values) and
                      Shows model      on data.           semi-random data for
                      fit statistics.                     the target (y-values) of the
                                                          model. We use predictors as
                                                          input to create the target so
                                                          we infer a correlation here.
```

Modeldiagnosticsandmodelcomparison
- You'llbebuildingmultiplemodelsfromwhichyouthenchoosethebestonebasedonmultiplecriteria.Working with aholdout sample helpsyoupick thebest-performingmodel.
- A holdoutsampleisapartofthedatayouleaveoutofthemodelbuildingsoitcanbeusedtoevaluatethemodel afterward.
- Theprinciplehereissimple:themodelshouldworkonunseendata.Youuseonlyafractionofyourdatato estimatethemodel and theotherpart, theholdout sample, is kept out of theequation.
- Themodelisthen unleashedonthe unseendataand errormeasuresarecalculated toevaluateit.
- Multipleerrormeasuresareavailable,andinfigureweshowthegeneralideaoncomparingmodels.Theerror measureusedin the exampleis themean squareerror.

Formulaformeansquareerror.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y_i} - Y_i)^2$$

Meansquareerrorisasimplemeasure:checkforeverypredictionhowfaritwasfromthetruth,squarethiserror,and add up theerror ofeveryprediction.

n	Size	Price	Predicted model 1	Predicted model 2	Error model 1	Error model 2
1	10	3				
2	15	5				
3	18	6				
4	14	5				
…	…					
800	9	3				
801	12	4	12	10	0	2
802	13	4	12	10	1	3
…						
999	21	7	21	10	0	11
1000	10	4	12	10	−2	0
				Total	5861	110225

80% train (rows 1–800), 20% test (rows 801–1000)

A holdout sample helps you compare models and ensures that you can generalize results to data that the model has not yet seen.

Above figure compares the performance of two models to predict the order size from the price. The first model is $size=3 * price$ and the second model is $size=10$.

- To estimate the models, we use 800 randomly chosen observations out of 1,000 (or 80%), without showing the other 20% of data to the model.
- Once the model is trained, we predict the values for the other 20% of the variables based on those for which we already know the true value, and calculate the model error with an error measure.
- Then we choose the model with the lowest error. In this example we chose model 1 because it has the lowest total error.

Many models make strong assumptions, such as independence of the inputs, and you have to verify that these assumptions are indeed met. This is called *model diagnostics*.

Presenting findings and building applications

- Sometimes people get so excited about your work that you'll need to repeat it over and over again because they value the predictions of your models or the insights that you produced.
- This doesn't always mean that you have to redo all of your analysis all the time. Sometimes it's sufficient that you implement only the model scoring; other times you might build an application that automatically updates reports, Excel spreadsheets, or PowerPoint presentations. The last stage of the data science process is where your *soft skills* will be most useful, and yes, they're extremely important.

Data mining
Data mining is the process of discovering actionable information from large sets of data. Data mining uses mathematical analysis to derive patterns and trends that exist in data. Typically, these patterns cannot be discovered by traditional data exploration because the relationships are too complex or because there is too much data.

These patterns and trends can be collected and defined as a *data mining model*. Mining models can be applied to specific scenarios, such as:
- **Forecasting**: Estimating sales, predicting server loads or server downtime

16

- **Risk and probability**: Choosing the best customers for targeted mailings, determining the probable break-even point for risk scenarios, assigning probabilities to diagnoses or other outcomes
- **Recommendations**: Determining which products are likely to be sold together, generating recommendations
- **Finding sequences**: Analyzing customer selections in a shopping cart, predicting next likely events
- **Grouping**: Separating customers or events into cluster of related items, analyzing and predicting affinities

Building a mining model is part of a larger process that includes everything from asking questions about the data and creating a model to answer those questions, to deploying the model into a working environment. This process can be defined by using the following six basic steps:

1. Defining the Problem
2. Preparing Data
3. Exploring Data
4. Building Models
5. Exploring and Validating Models
6. Deploying and Updating Models

The following diagram describes the relationships between each step in the process, and the technologies in Microsoft SQL Server that you can use to complete each step.
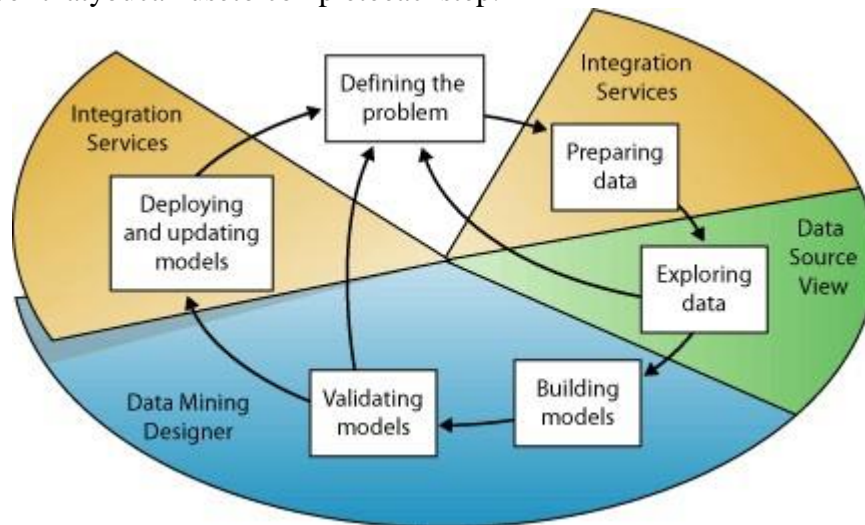
Defining the Problem

The first step in the data mining process is to clearly define the problem, and consider ways that data can be utilized to provide an answer to the problem.

This step includes analyzing business requirements, defining the scope of the problem, defining the metrics by which the model will be evaluated, and defining specific objectives for the data mining project. These tasks translate into questions such as the following:

- What are you looking for? What types of relationships are you trying to find?
- Does the problem you are trying to solve reflect the policies or processes of the business?
- Do you want to make predictions from the data mining model, or just look for interesting patterns and associations?
- Which outcome or attribute do you want to try to predict?

17

- What kind of data do you have and what kind of information is in each column? If there are multipletables, how are the tables related? Do you need to perform any cleansing, aggregation, or processing tomakethe data usable?
- How is the data distributed? Is the data seasonal? Does the data accurately represent the processes ofthe business?

PreparingData
- The second step in the data mining process is to consolidate and clean the data that was identified intheDefiningthe Problemstep.
- Data can be scattered across a company and stored in different formats, or may contain inconsistenciessuchas incorrect or missingentries.
- Data cleaning is not just about removing bad data or interpolating missing values, but about findinghidden correlations in the data, identifying sources of data that are the most accurate, and determiningwhichcolumns arethemost appropriate for useinanalysis

ExploringData
Exploration techniques include calculating the minimum and maximum values, calculating mean and standarddeviations, and looking at the distribution of the data. For example, you might determine by reviewing themaximum, minimum, and mean values that the data is not representative of your customers or businessprocesses, and that you therefore must obtain more balanced data or review the assumptions that are the basisfor your expectations. Standard deviations and other distribution values can provide useful information aboutthe stabilityand accuracyof theresults.

BuildingModels
The mining structure is linked to the source of data, but does not actually contain any data until you process it.Whenyouprocesstheminingstructure,SQLServerAnalysisServicesgeneratesaggregatesandotherstatistical information that can be used for analysis. This information can be used by any mining model that isbasedon the structure.

ExploringandValidatingModels
Before you deploy a model into a production environment, you will want to test how well the model performs.Also, when you build a model, you typically create multiple models with different configurations and test allmodelsto seewhichyieldsthebest results foryourproblem andyour data.

DeployingandUpdatingModels
After the mining models exist in a production environment, you can perform many tasks, depending on yourneeds.Thefollowingare some ofthe tasksyou canperform:
- Usethemodelstocreate predictions,whichyou canthenusetomakebusinessdecisions.
- Createcontentqueriestoretrievestatistics,rules,orformulasfromthemodel.
- Embed data mining functionality directly into an application. You can include Analysis ManagementObjects (AMO), which contains a set of objects that your application can use to create, alter, process,and deleteminingstructures andminingmodels.
- Use Integration Services to create a package in which a mining model is used to intelligently separateincomingdata into multipletables.
- Createa report that lets users directlyqueryagainst an existingminingmodel
- Updatethemodels afterreviewandanalysis. Anyupdaterequiresthatyoureprocess themodels.
- Update the models dynamically, as more data comes into the organization, and making constantchangesto improvetheeffectivenessof thesolution shouldbepartofthedeployment strategy.

Datawarehousing

Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructedby integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or adhocqueries,anddecisionmaking.Datawarehousinginvolvesdatacleaning,dataintegration,anddataconsolidations.

Characteristicsof datawarehouse

Themain characteristics of adata warehouseareas follows:

- Subject-Oriented

  A data warehouse is subject-oriented since it provides topic-wise information rather than theoverallprocesses ofabusiness. Such subjectsmaybesales,promotion, inventory,etc

- Integrated

  A data warehouse is developed by integrating data from varied sources into a consistent format.The data must be stored in the warehouse in a consistent and universally acceptable manner in terms ofnaming,format, and coding. Thisfacilitates effectivedata analysis.

- Non-Volatile

  Data once entered into a data warehouse must remain unchanged. All data is read-only. Previousdata is not erased whencurrent data is entered.This helpsyou to analyze what has happened andwhen.

- Time-Variant

  The data stored in a data warehouse is documented with an element of time, either explicitly orimplicitly. An example of time variance in Data Warehouse is exhibited in the Primary Key, whichmusthavean element oftime like theday,week,ormonth.

Databasevs.DataWarehouse

Although a data warehouse and a traditional database share some similarities, they need not be the same idea.The main difference is that in a database, data is collected for multiple transactional purposes. However, in adata warehouse, data is collected on an extensive scale to perform analytics. Databases provide real-time data,whilewarehouses storedata to beaccessedforbig analytical queries.

DataWarehouseArchitecture

Usually,datawarehouse architecturecomprises athree-tierstructure.

BottomTier

Thebottomtierordatawarehouseserverusuallyrepresentsarelationaldatabasesystem.Back-endtoolsareusedto cleanse, transform and feed data into this layer.

MiddleTier

Themiddletierrepresentsan OLAPserver thatcan beimplemented intwo ways.
TheROLAPorRelationalOLAPmodelisanextendedrelationaldatabasemanagementsystemthatmapsmultidimensionaldataprocess to standardrelational process.
TheMOLAPormultidimensionalOLAPdirectlyactsonmultidimensionaldataandoperations.

TopTier

This is the front-endclient interfacethatgets data out from thedatawarehouse.Itholds varioustools likequerytools, analysis tools, reportingtools, anddataminingtools.

How DataWarehouseWorks

Data Warehousing integrates data and information collected from various sources into one comprehensivedatabase. For example, a data warehouse might combine customer information from an organization's point-of-salesystems,itsmailinglists,website,andcommentcards.Itmightalsoincorporateconfidential
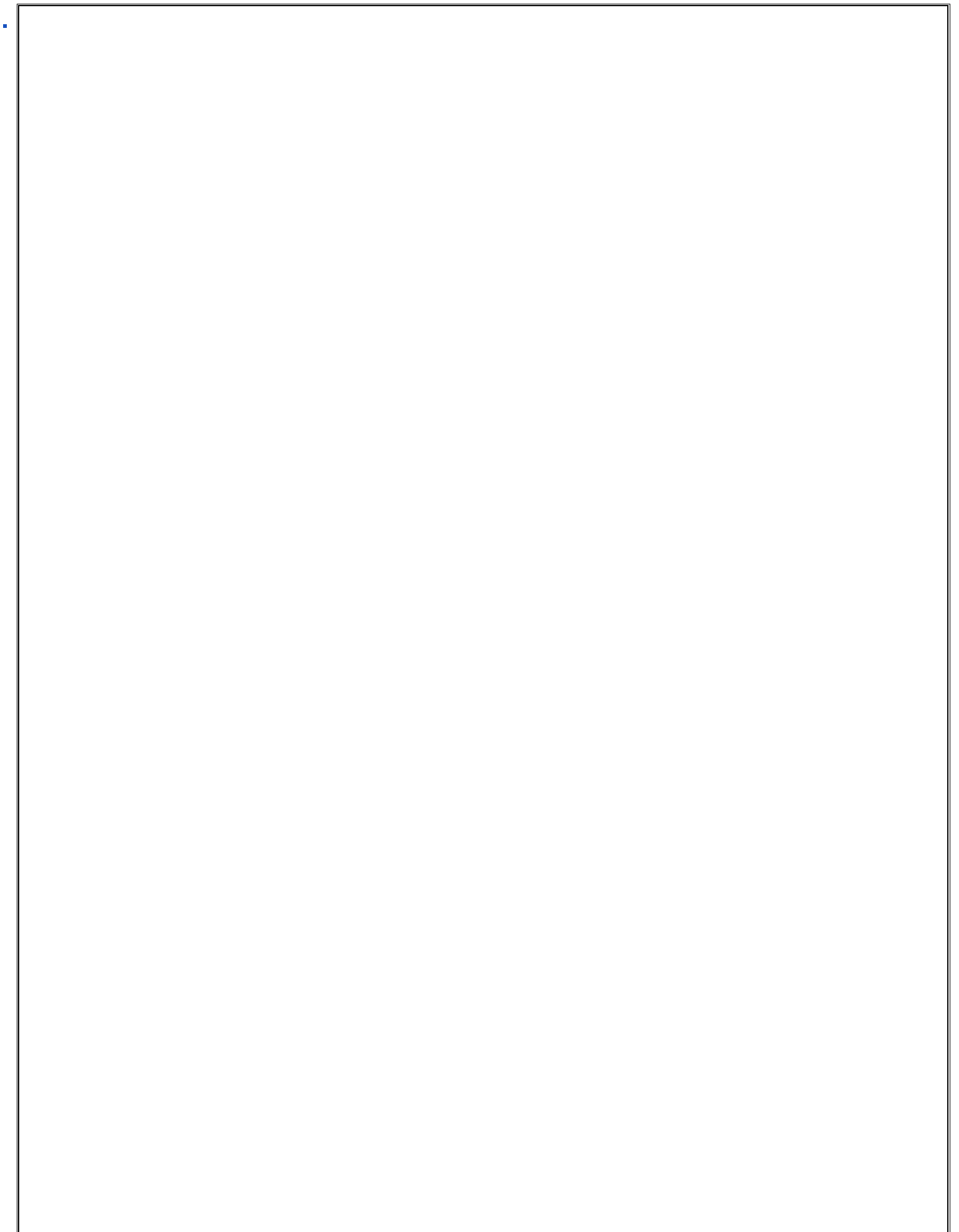
information about employees, salary information, etc. Businesses use such components of data warehouse toanalyze customers.

Data mining is one of the features of a data warehouse that involves looking for meaningful data patterns invastvolumes ofdataanddevisinginnovative strategies forincreased salesand profits.

Typesof DataWarehouse
Therearethreemain typesof datawarehouse.

EnterpriseDataWarehouse(EDW)
This type of warehouse serves as a key or central database that facilitates decision-support services throughoutthe enterprise. The advantage to this type of warehouse is that it provides access to cross-organizationalinformation,offersaunified approach todatarepresentation,and allows runningcomplexqueries.

OperationalDataStore (ODS)
=his type of data warehouse refreshes in real-time. It is often preferred for routine activities like storingemployeerecords.Itisrequired whendatawarehousesystems donotsupportreportingneedsofthebusiness.

DataMart
A data mart is a subset of a data warehouse built to maintain a particular department, region, or business unit.Every department of a business has a central repository or data mart to store data. The data from the data martisstored intheODSperiodically. TheODSthen sendsthedatato theEDW, whereit is storedandused.

Summary
Inthischapteryoulearnedthedatascienceprocessconsistsofsixsteps:
- *Settingtheresearchgoal*—Definingthewhat,thewhy,andthehowofyourprojectinaprojectcharter.
- *Retrievingdata*—Findingandgettingaccesstodataneededinyourproject.Thisdataiseitherfoundwithinthe companyor retrieved from a third party.
- *Datapreparation*—Checking andremediatingdataerrors,enrichingthedatawithdatafromotherdatasources,andtransformingit into asuitableformatforyour models.
- *Dataexploration*—Divingdeeperintoyourdata usingdescriptivestatisticsandvisualtechniques.
- *Datamodeling*—Usingmachinelearningandstatisticaltechniquestoachieveyourprojectgoal.
- *Presentationandautomation*— Presentingyourresultstothestakeholdersandindustrializingyouranalysisprocess for repetitive reuse andintegration withothertools.
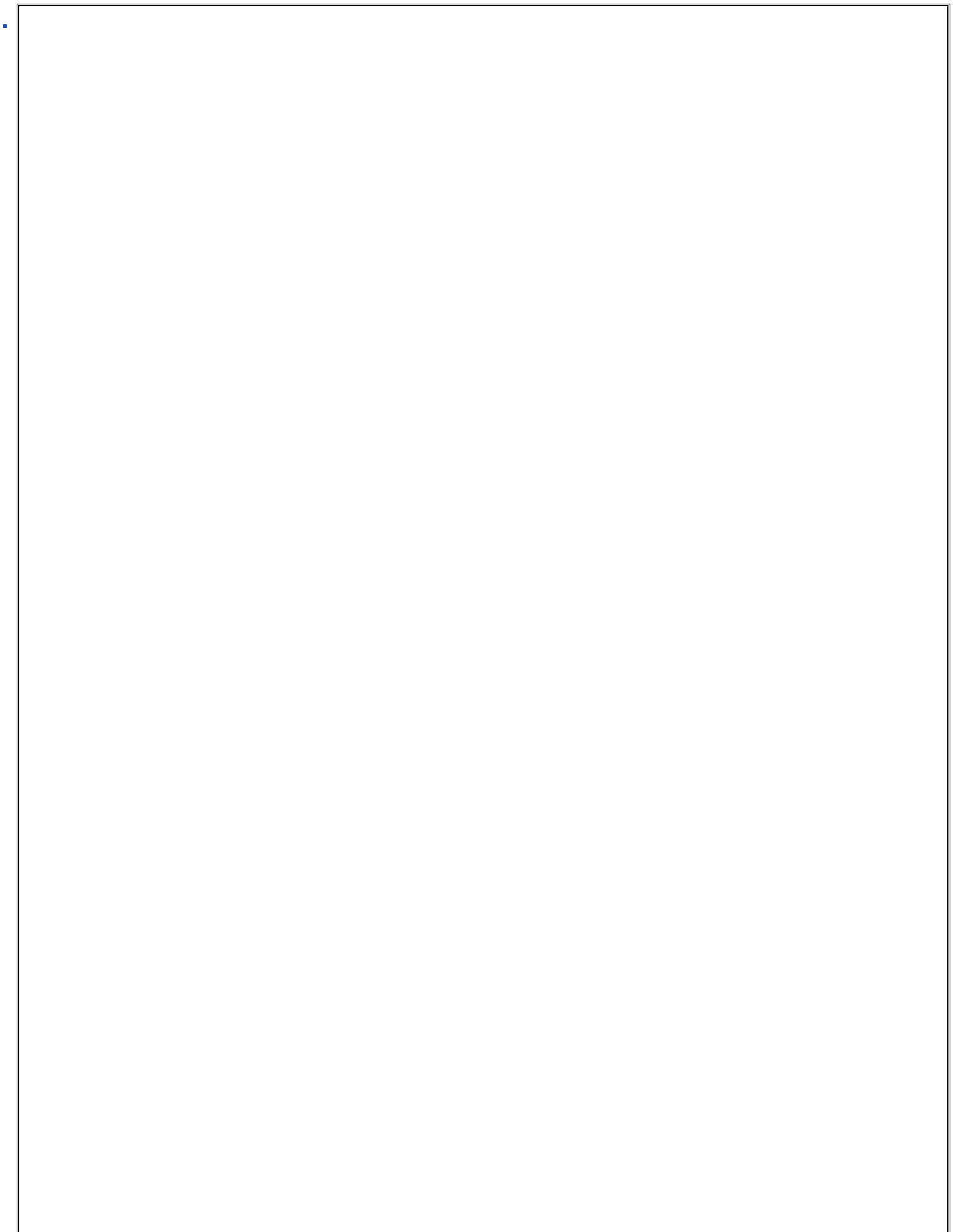
Unit– II
DESCRIBINGDATA

TypesofData-Typesof Variables-DescribingDatawithTablesandGraphs–DescribingDatawithAverages - DescribingVariability-Normal Distributionsand Standard (z) Scores

THREETYPESOFDATA

- **Qualitative data**consistof words(YesorNo),letters(Y orN), or numericalcodes(0or 1) thatrepresenta class or category.
- **Rankeddata**consistofnumbers(1st,2nd,...40thplace)thatrepresentrelativestandingwithinagroup.
- **Quantitative data**consistofnumbers(weightsof238,170,...185lbs)thatrepresentanamountor acount.To determinethe typeof data, focus onasingleobservation inanycollectionofobservations

TYPESOFVARIABLES

A**variable** is a characteristicorpropertythat cantakeon different values.

- Theweightscanbedescribednotonlyasquantitativedatabutalsoasobservationsforaquantitativevariable,sin cethe various weights takeon different numerical values.
- Bythesametoken,therepliescanbedescribedasobservationsforaqualitativevariable,sincetherepliesto theFacebookprofilequestion takeon different values of either Yes orNo.
- Giventhisperspective,anysingleobservationcanbedescribedasaconstant,sinceittakesononlyonevalue.

DiscreteandContinuousVariables

Quantitativevariables canbefurtherdistinguishedas**discreteorcontinuous.**

A**discretevariable** consistsofisolated numbersseparated bygaps.
Discretevariablescanonlyassumespecificvaluesthatyoucannotsubdivide.Typically,youcountdiscretevalues,and the results areintegers.
Examples

- Counts-suchasthe numberof childrenin a family.(1,2,3,etc., butnever 1.5)
- Thesevariablescannothavefractional ordecimalvalues.You canhave20or21cats,butnot20.5
- Thenumberofheads in asequenceofcoin tosses.
- Theresult of rollingadie.
- Thenumberof patients in ahospital.
- Thepopulationof acountry.

While discrete variables have no decimal places, the average of these values can be fractional. For example,families can have only adiscrete number of children: 1, 2, 3, etc. However, the average number of childrenper familycan be2.2.

A**continuousvariable** consistsofnumberswhosevalues,atleastintheory,have norestrictions.
Continuousvariablescanassumeanynumericvalueandcanbemeaningfullysplitintosmallerparts.Consequently, they have valid fractional and decimal values. In fact, continuous variables have an infinitenumberof potential valuesbetweenanytwo points.Generally,you measurethemusingascale.

Examplesofcontinuousvariablesincludeweight, height,length,time,andtemperature.
Durations, such as the reaction times of grade school children to a fire alarm; and standardized test scores,suchas thoseon the ScholasticAptitudeTest (SAT).

IndependentandDependentVariablesIndependentVariable

*Inanexperiment,an***independentvariable** *isthetreatmentmanipulatedbytheinvestigator.*

- Independentvariables(IVs)aretheonesthatyouincludeinthemodeltoexplainorpredictchangesinthedependent variable.
- Independentindicates thattheystand aloneand other variablesin themodel do notinfluencethem.
- Independent variables are also known as predictors, factors, treatment variables, explanatory variables,input variables, x-variables, and right-hand variables—because they appear on the right side of theequalssign in aregression equation.
- Itisavariablethatstandsaloneandisn'tchangedbytheothervariablesyouaretryingtomeasure.

For example, someone'sage might be an independentvariable. Other factors (such as what they eat, howmuchtheygo to school,how much television theywatch)

The impartial creation of distinct groups, which differ only in terms of the independent variable, has a mostdesirableconsequence.Oncethedatahavebeencollected,anydifferencebetweenthegroupscanbeinterpreted as being*caused* bytheindependent variable.

DependentVariable

When a variable is believed to have been influenced by the independent variable, it is called a **dependentvariable.**Inanexperimentalsetting,thedependentvariableismeasured,counted,orrecordedbytheinvestigator.

- Thedependentvariable(DV)iswhatyouwanttousethemodeltoexplainorpredict.Thevaluesofthisvariabledepend on othervariables.
- It'salsoknownastheresponsevariable,outcomevariable,andleft-handvariable.Graphsplacedependentvariables on thevertical, or Y, axis.
- adependentvariableisexactlywhatitsounds like. It issomethingthat dependson otherfactors.

Forexamplethebloodsugartestdependsonwhatfoodyouate,atwhichtimeyouateetc.
Unlike the independent variable, the dependent variable isn't manipulated by the investigator. Instead, itrepresentsan outcome: thedata produced bytheexperiment.

ConfoundingVariable

An uncontrolled variable that compromises the interpretation of a study is known as a **confounding variable.**Sometimes a confounding variable occurs because it's impossible to assign subjects randomly to differentconditions.

DescribingDatawithTablesandGraphs
FrequencyDistributionsforQuantitativeData

- A frequency distribution is a collection of observationsproduced by sorting observations into classes and showingtheirfrequency(f)of occurrencein each class.
- When observations are sorted into classes of singlevalues, as in Table 2.1, the result is referred to as a frequencydistributionfor ungrouped data.
- The frequency distribution shown in Table 2.1 is only partiallydisplayedbecausetherearemorethan100possiblevaluesbetweenthelargest and smallest observations.

Frequencydistribution tableis much moreinformativeifpossible

observedvaluesislessthen20.Ifmoreentryisobservedthengroupedda ta is used.

GroupedData

According to their frequency of occurrence. When observations are sortedintoclasses of morethanonevalueresultis referredto as afrequency for groupeddata.(Shownintable2.2)

- Thegeneralstructureof thisfrequencydistribution isthedata'saregroupedinto class intervalswith 10 possiblevalueseach.
- Thefrequency(f) column shows thefrequencyofobservations in eachclassand, atthebottom, thetotalnumber ofobservationsin allclasses.

Table 2.2 FREQUENCY DISTRIBUTION (GROUPED DATA)	
WEIGHT	f
240–249	1
230–239	0
220–229	3
210–219	0
200–209	2
190–199	4
180–189	3
170–179	7
160–169	12
150–159	17
140–149	1
130–139	3
Total	53

GUIDELINES

Table 2.3 FREQUENCY DISTRIBUTION WITH TOO MANY INTERVALS	
WEIGHT	f
245–249	1
240–244	0
235–239	0
230–234	0
225–229	2
220–224	1
215–219	0
210–214	0
205–209	2
200–204	0
195–199	0
190–194	4
185–189	1
180–184	2
175–179	2
170–174	5
165–169	7
160–164	5
155–159	9
150–154	8
145–149	1
140–144	0
135–139	2
130–134	1
Total	53

GUIDELINES FOR FREQUENCY DISTRIBUTIONS

Essential

1. *Each observation should be included in one, and only one, class.*
   Example: 130–139, 140–149, 150–159, etc. It would be incorrect to use 130–140, 140–150, 150–160, etc., in which, because the boundaries of classes overlap, an observation of 140 (or 150) could be assigned to either of two classes.

2. *List all classes, even those with zero frequencies.*
   Example: Listed in Table 2.2 is the class 210–219 and its frequency of zero. It would be incorrect to skip this class because of its zero frequency.

3. *All classes should have equal intervals.*
   Example: 130–139, 140–149, 150–159, etc. It would be incorrect to use 130–139, 140–159, etc., in which the second class interval (140–159) is twice as wide as the first class interval (130–139).

Optional

4. *All classes should have both an upper boundary and a lower boundary.*
   Example: 240–249. Less preferred would be 240–above, in which no maximum value can be assigned to observations in this class. (Nevertheless, this type of open-ended class is employed as a space-saving device when many different tables must be listed, as in the *Statistical Abstract of the United States*. An open-ended class appears in the table "Two Age Distributions" in Review Question 2.17 at the end of this chapter.)

5. *Select the class interval from convenient numbers, such as 1, 2, 3, . . . 10, particularly 5 and 10 or multiples of 5 and 10.*
   Example: 130–139, 140–149, in which the class interval of 10 is a convenient number. Less preferred would be 130–142, 143–155, etc., in which the class interval of 13 is not a convenient number.

6. *The lower boundary of each class interval should be a multiple of the class interval.*
   Example: 130–139, 140–149, in which the lower boundaries of 130, 140, are multiples of 10, the class interval. Less preferred would be 135–144, 145–154, etc., in which the lower boundaries of 135 and 145 are not multiples of 10, the class interval.

7. *Aim for a total of approximately 10 classes.*
   Example: The distribution in Table 2.2 uses 12 classes. Less preferred would be the distributions in Tables 2.3 and 2.4. The distribution in Table 2.3 has too many classes (24), whereas the distribution in Table 2.4 has too few classes (3).

Table 2.4 FREQUENCY DISTRIBUTION WITH TOO FEW INTERVALS	
WEIGHT	f
200–249	6
150–199	43
100–149	4
Total	53

CONSTRUCTING FREQUENCY DISTRIBUTIONS

1. **Find the range, that is,** the difference between the largest and smallest observations. The range of weights in Table 1.1 is $245 - 133 = 112$.

2. **Find the class interval required to span the range** by dividing the range by the desired number of classes (ordinarily 10). In the present example,

$$Class\ interval = \frac{range}{desired\ number\ of\ classes} = \frac{112}{10} = 11.2$$

3. **Round off to the nearest convenient interval** (such as 1, 2, 3, . . . 10, particularly 5 or 10 or multiples of 5 or 10). In the present example, the nearest convenient interval is 10.

4. **Determine where the lowest class should begin.** (Ordinarily, this number should be a multiple of the class interval.) In the present example, the smallest score is 133, and therefore the lowest class should begin at 130, since 130 is a multiple of 10 (the class interval).

5. **Determine where the lowest class should end** by adding the class interval to the lower boundary and then subtracting one unit of measurement. In the present example, add 10 to 130 and then subtract 1, the unit of measurement, to obtain 139—the number at which the lowest class should end.

6. **Working upward, list as many equivalent classes as are required to include the largest observation.** In the present example, list 130–139, 140–149, . . . , 240–249, so that the last class includes 245, the largest score.

7. **Indicate with a tally the class in which each observation falls.** For example, the first score in Table 1.1, 160, produces a tally next to 160–169; the next score, 193, produces a tally next to 190–199; and so on.

8. **Replace the tally count for each class with a number—the frequency (f)—and show the total of all frequencies.** (Tally marks are not usually shown in the final frequency distribution.)

9. **Supply headings for both columns and a title for the table.**

OUTLIERS

Anoutlierisanextremelyhighorextremelylowdatapointrelativetothenearestdatapointandtherestofthe neighboringco-existingvalues in a data graph ordatasetyou'reworkingwith.

Outliersare extremevalues thatstand out greatlyfrom theoverall patternof values ina datasetorgraph.

RELATIVEFREQUENCYDISTRIBUTIONS

Relative frequency distributions show the frequency of eachclass as a part or fraction of the total frequency for the entiredistribution.

Thistypeofdistributionisespeciallyhelpfulwhenyoumustcomp are two or more distributions based on different totalnumbersof observations.

The conversion to relative frequencies allows a directcomparison of the shapes oftwo distributions withoutadjust otherobservations.

ConstructingRelativeFrequencyDistributions

To convert a frequency distribution into a relative frequencydistribution,divide the frequencyfor each classbythe totalfrequencyfortheentiredistribution.
Table2.5illustratesarelativefrequencydistributionbasedonthew eight distribution of Table 2.2.

Table 2.5
RELATIVE FREQUENCY DISTRIBUTION

WEIGHT	f	RELATIVE f
240–249	1	.02
230–239	0	.00
220–229	3	.06
210–219	0	.00
200–209	2	.04
190–199	4	.08
180–189	3	.06
170–179	7	.13
160–169	12	.23
150–159	17	.32
140–149	1	.02
130–139	3	.06
Total	53	1.02*

The sum does not equal 1.00 because of rounding-off errors.

PercentagesorProportions

Some people prefer to deal with percentages rather than proportions because percentages usually lackdecimal points. A proportion always varies between 0 and 1, whereas a percentage always varies between0percentand 100 percent.

To convert the relative frequencies, multiply each proportion by 100; that is, move the decimal point twoplacesto theright.

CUMULATIVEFREQUENCYDISTRIBUTIONS

Cumulative frequency distributions show the total number of observations in each class and in all lower-rankedclasses.

Cumulative frequencies are usually converted, in turn, to cumulative percentages. Cumulative percentagesareoften referred to as percentileranks.

ConstructingCumulativeFrequencyDistributions

To convert a frequency distribution into a cumulative frequency distribution, add to the frequency of eachclassthe sum of thefrequencies ofall classes ranked below it.

Table 2.6
CUMULATIVE FREQUENCY DISTRIBUTION

WEIGHT	f	CUMULATIVE f	CUMULATIVE PERCENT
240–249	1	53	100
230–239	0	52	98
220–229	3	52	98
210–219	0	49	92
200–209	2	49	92
190–199	4	47	89
180–189	3	43	81
170–179	7	40	75
160–169	12	33	62
150–159	17	21	40
140–149	1	4	8
130–139	3	3	6
Total	53		

CumulativePercentages

Ashasbeensuggested,ifrelativestandingwithinadistributionisparticularlyimportant,thencumulativefrequenci esareconverted to cumulative percentages

Toobtainthiscumulativepercentage,thecumulativefrequencyoftheclassshouldbedividedbythetotalfrequencyof the entiredistribution.

PercentileRanks

Whenusedtodescribetherelativepositionofanyscorewithinitsparentdistribution,cumulativepercentagesareref erred to as percentile ranks.

The percentile rank of a score indicates the percentage of scores in the entire distribution with similar orsmaller values than that score. Thus a weight has a percentile rank of 80 if equal or lighter weightsconstitute80 percent of theentiredistribution.

FREQUENCYDISTRIBUTIONSFORQUALITATIVE(NOMINAL)DATA

Frequency distributions for qualitative data are easy to construct.Simply determine the frequency with which observations occupyEachclass, andreportthesefrequenciesasshowninTable2.7forthe Facebook profile survey

Qualitative data have an ordinal level of measurement becauseObservationscanbeorderedfromleasttomost,thatordershou ldbepreserved in thefrequencytable

Table 2.7 FACEBOOK PROFILE SURVEY	
Response	**f**
Yes	56
No	27
Total	83

RelativeandCumulativeDistributionsforQualitativeData

Frequencydistributionsforqualitativevariablescanalwaysbeconvertedintorelativefrequencydistributions. ifmeasurementisordinalbecauseobservationscanbeorderedfromleasttomost,cumulativefrequencies(andcum ulativepercentages) can beused.

Table 2.8
RANKS OF OFFICERS IN THE U.S. ARMY (PROJECTED 2016)

RANK	f	PROPORTION	CUMULATIVE PERCENT
General	311	.004*	100.0
Colonel	13,156	.167	99.6
Major	16,108	.204	82.9
Captain	29,169	.370	62.5
Lieutenant	20,083	.255	25.5
Total	78,827		

To avoid a value of .00 for General, proportions are carried three places to the right of the decimal point.

GRAPHS

Datacanbedescribedclearlyandconciselywiththeaidofawell-constructedfrequencydistribution.Anddata can often bedescribed even morevividlybyconvertingfrequencydistributions into graphs.

GRAPHSFORQUANTITATIVEDATA

Histograms

Abar-typegraphforquantitativedata.Thecommonboundariesbetweenadjacentbarsemphasizethecontinuityof thedata, aswith continuous variables.
Ahistogramisadisplayofstatisticalinformationthatusesrectanglestoshowthefrequencyofdataitemsinsuccessiv enumericalintervals of equal size.

Importantfeaturesofhistograms

- Equalunitsalongthehorizontalaxis(theXaxis,orabscissa)reflectthevariousclassintervalsofthefrequencydistribution.
- Equalunitsalongtheverticalaxis(theYaxis,orordinate)reflectincreasesinfrequency.(Theunitsalongthe verticalaxis donot haveto be thesame width as those alongthehorizontal axis.)
- Theintersectionofthetwo axesdefinestheoriginatwhich bothnumericalscalesequal0.
- Numericalscalesalwaysincreasefromlefttorightalongthehorizontalaxisandfrombottomtotopalongthe verticalaxis
- Thebodyofthehistogramconsistsofaseriesofbarswhoseheightsreflectthefrequenciesforthevariousclasses.
- Theadjacentbarsinhistogramshavecommonboundariesthatemphasizethecontinuityofquantitativedata forcontinuous variables.
- Theintroductionofgapsbetweenadjacentbarswouldsuggestanartificialdisruptioninthedatamoreappropriate fordiscrete quantitative variables orforqualitativevariables.
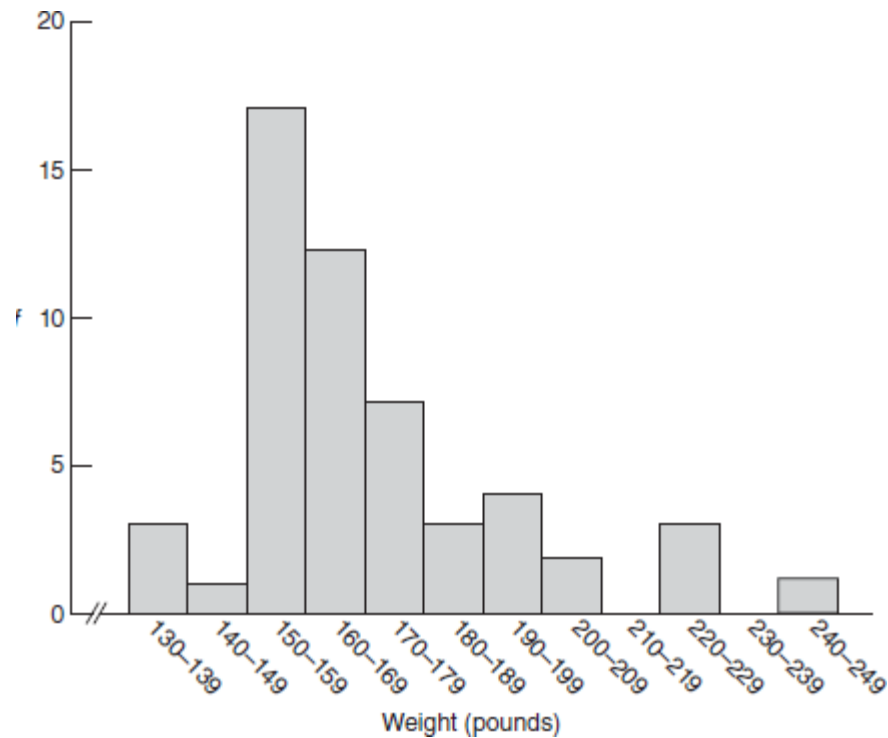
Figure:Histogram

FrequencyPolygon

Animportantvariationonahistogramisthefrequencypolygon,orlinegraph.Frequencypolygonsmaybeconstructed directlyfrom frequencydistributions.

Step-by-steptransformation of ahistogramintoafrequencypolygon

A. Thispanelshowsthehistogramforthe weightdistribution.
B. Placedotsatthemidpointsofeachbartopor,intheabsenceofbartops,atmidpointsforclassesonthehorizontal axis, and connect them with straight lines.
C. c. Anchor the frequency polygon to the horizontal axis. First, extend the upper tail to the midpoint ofthe first unoccupied class on the upper flank of the histogram. Then extend the lower tail to themidpoint of the first unoccupied class on the lower flank of the histogram. Now all of the area underthefrequencypolygon is enclosed completely.
D. Finally,eraseallof thehistogrambars, leavingonlythe frequencypolygon.
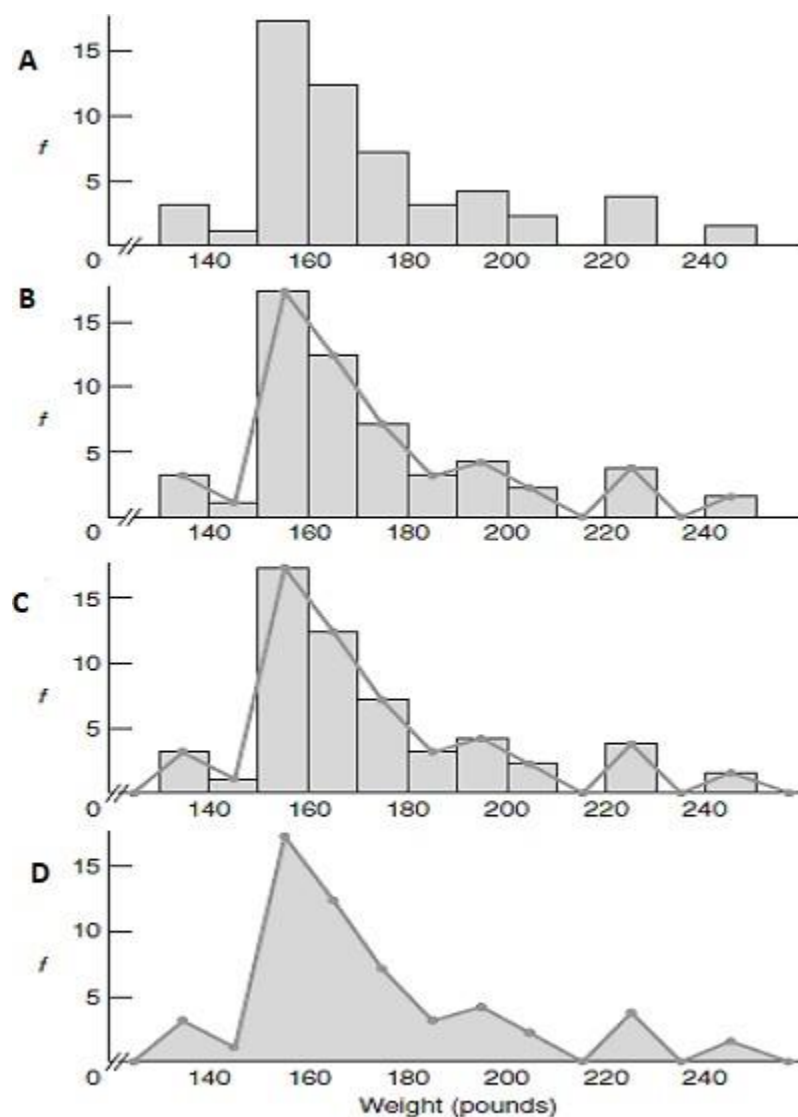
FIGURE
Transition from histogram to frequency polygon.

8

Stemand LeafDisplays
Another technique for summarizing quantitative data is a stem and leaf display. Stem and leaf displays areidealforsummarizingdistributions,suchasthatforweightdata,withoutdestroyingtheidentitiesofindividualobser vations.

Constructing StemandLeaf Display

Theleftmostpaneloftablere-createstheweights.
To construct the stem and leaf display for the table given below, first note that, when counting by tens, theweightsrangefrom the130s to the 240s.
Arrange a column of numbers, the stems, beginning with 13 (representing the 130s) and ending with 24(representing the 240s). Draw a vertical line to separate the stems, which represent multiples of 10, from thespaceto be occupied bytheleaves, which represent multiples of 1.

Forexample
Enter each raw score into the stem and leaf display. As suggested by the shaded coding in Table 2.9, the firstraw score of 160 reappears as a leaf of 0 on a stem of 16. The next raw score of 193 reappears as a leaf of 3 ona stem of 19, and the third raw score of 226 reappears as a leaf of 6 on a stem of 22, and so on, until each rawscorereappears asaleafon its appropriatestem.

Table 2.9
CONSTRUCTING STEM AND LEAF DISPLAY FROM WEIGHTS OF MALE STATISTICS STUDENTS

RAW SCORES				STEM AND LEAF DISPLAY	
160	165	135	175		
193	168	245	165	13	3 5 5
226	169	170	185	14	5
152	160	156	154	15	2 7 1 7 8 0 2 0 2 6 9 8 2 6 4 7 6
180	170	160	179	16	0 3 5 8 9 0 0 0 6 5 5 5
205	150	225	165	17	2 0 0 0 2 5 9
163	152	190	206	18	0 0 5
157	160	159	165	19	3 0 0 0
151	190	172	157	20	5 6
157	150	190	156	21	
220	133	166	135	22	6 0 5
145	180	158		23	
158	152	152		24	5
172	170	156			

TYPICALSHAPES
Whetherexpressedasahistogram,afrequencypolygon,orastemandleafdisplay,animportantcharacteristic   of   a frequency distribution is its shape. Below figure shows some of the more typical shapes forsmoothedfrequencypolygons (which ignoretheinevitable irregularities ofreal data).
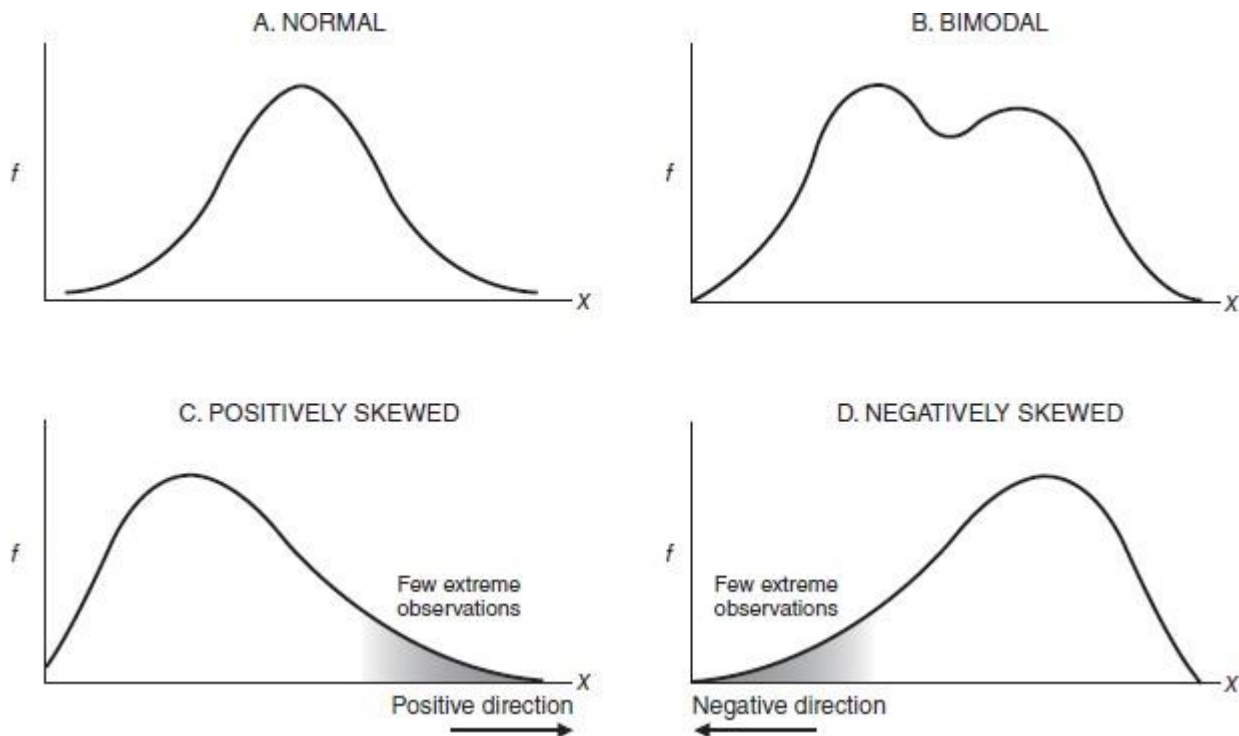
AGRAPH FORQUALITATIVE(NOMINAL)DATA

- As with histograms, equal segments along the horizontal axis are allocated to the different words orclasses that appear in the frequency distribution for qualitative data. Likewise, equal segments alongthe vertical axis reflect increases in frequency. The body of the bar graph consists of a series of barswhoseheights reflect thefrequencies forthevarious words or classes.
- A person's answer to the question "Do you have a Facebook profile?" is either Yes or No, not someimpossibleintermediate value, such as 40percent Yes and60 percentNo.
- Gapsareplacedbetweenadjacentbarsofbargraphstoemphasizethediscontinuousnatureofqualitativedata.

MISLEADINGGRAPHS

Graphscanbeconstructedin anunscrupulousmanner tosupport aparticularpoint ofview.
Popularsayingssays,including"Numbersdon'tlie,butstatisticiansdo"and"Therearethreekindsoflies—lies,damned lies,andstatistics."

10
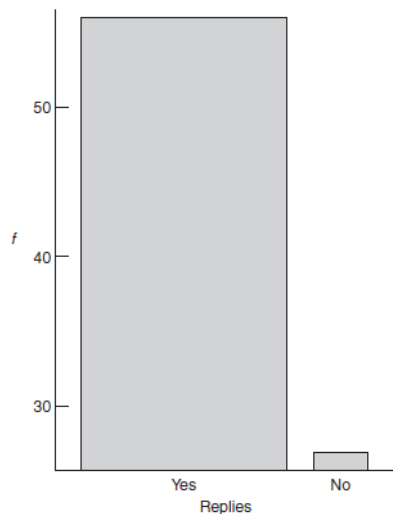
CONSTRUCTING GRAPHS

1. ***Decide on the appropriate type of graph,*** recalling that histograms and frequency polygons are appropriate for quantitative data, while bar graphs are appropriate for qualitative data and also are sometimes used with discrete quantitative data.

2. ***Draw the horizontal axis, then the vertical axis,*** remembering that the vertical axis should be about as tall as the horizontal axis is wide.

3. ***Identify the string of class intervals that eventually will be superimposed on the horizontal axis.*** For qualitative data or ungrouped quantitative data, this is easy—just use the classes suggested by the data. For grouped quantitative data, proceed as if you were creating a set of class intervals for a frequency distribution. (See the box "Constructing Frequency Distributions" on page 27.)

4. ***Superimpose the string of class intervals (with gaps for bar graphs) along the entire length of the horizontal axis.*** For histograms and frequency polygons, be prepared for some trial and error—use a pencil! Do not use a string of empty class intervals to bridge a sizable gap between the origin of 0 and the smallest class interval. Instead, use wiggly lines to signal a break in scale, then begin with the smallest class interval. Also, do not clutter the horizontal scale with excessive numbers—use just a few convenient numbers.

5. ***Along the entire length of the vertical axis, superimpose a progression of convenient numbers,*** beginning at the bottom with 0 and ending at the top with a number as large as or slightly larger than the maximum observed frequency. If there is a considerable gap between the origin of 0 and the smallest observed frequency, use wiggly lines to signal a break in scale.

6. Using the scaled axes, ***construct bars (or dots and lines) to reflect the frequency of observations within each class interval.*** For frequency polygons, dots should be located above the midpoints of class intervals, and both tails of the graph should be anchored to the horizontal axis, as described under "Frequency Polygons" in Section 2.8.

7. ***Supply labels for both axes and a title (or even an explanatory sentence) for the graph.***

DescribingDatawithAverages

MODE

Themode reflects thevalue ofthemost
frequentlyoccurringscore.Inother words
Amodeisdefinedasthevaluethathasahigherfrequencyinagivensetofvalues.Itisthevaluethatappearsthe
mostnumberof times.
Example:
Inthegivenset ofdata: 2,4, 5, 5,6, 7,themodeofthe dataset is5 sinceithas appearedin theset twice.

Typesof Modes
Bimodal,Trimodal&Multimodal(Morethanonemode)

* Whentherearetwomodesin adataset, thentheset iscalled**bimodal**

For example, The mode of Set A = {2,2,2,3,4,4,5,5,5} is 2 and 5, because both 2 and 5 is repeated three
timesinthegiven set.

* Whentherearethreemodes inadataset,then the setiscalled**trimodal**

Forexample, the modeof set A={2,2,2,3,4,4,5,5,5,7,8,8,8} is 2, 5 and 8

* Whentherearefouror moremodesina dataset,thenthe setiscalled**multimodal**

Example:Thefollowingtablerepresentsthenumberofwicketstakenbyabowlerin10matches.Findthemodeofthegive
n set ofdata.

Match No.	1	2	3	4	5	6	7	8	9	10
No. of Wickets	2	1	1	3	2	3	2	2	4	1

Itcanbeseenthat2wicketsweretakenbythebowlerfrequentlyindifferentmatches.Hence,themode ofthegivendata is
2.

MEDIAN

Themedianreflectsthe middlevaluewhenobservationsareorderedfromleasttomost.
Themediansplitsa setofordered observationsintotwoequal parts,theupper andlowerhalves.

FindingtheMedian
* Orderscores fromleast tomost.
* If thetotalnumberofobservationgivenisodd,thenthe formulatocalculate themedian

        is:Median=$\{(n+1)/2\}^{th}$term / observation

* If thetotalnumberofobservationiseven, thenthe

        medianformulais:Median= $1/2[(n/2)^{th}$term + $\{(n/2)+1\}^{th}$term]

Example1:

Findthemedianofthe following:

4, 17, 77, 25, 22, 23, 92, 82, 40, 24, 14, 12, 67, 23,29

Solution:

n=15

Whenweputthosenumbers intheorderwehave:

4, 12, 14, 17, 22, 23, 23, 24, 25, 29, 40, 67, 77, 82, 92,

Median={(n+1)/2}thterm

=(15+1)/2

=8

The8thterm in the list is24

Themedian value ofthis set ofnumbers is 24.

Example2:

Findthemedianofthe following:

9,7,2,11,18,12,6,4

Solution

n=8

Whenweputthosenumbers intheorderwehave:

2, 4, 6, 7, 9,11, 12, 18

Median=1/2[(n/2)thterm+{(n/2)+1}thterm]

=½[(8/2)term+((8/2)+1)term]

=1/2[4thterm+5thterm]   (in ourlist 4thtermis 7 and 5thtermis 9)

=½[7+9]

=1/2(16)

=8

Themedian value ofthis set ofnumbers is 8.

MEAN

Themean is found byaddingall scores andthen dividingbythenumberofscores.

Meanistheaverageofthegivennumbersandiscalculatedbydividingthesumofgivennumbersbythetotalnumberof numbers.

$$Mean = \frac{sum\ of\ all\ scores}{number\ of\ scores}$$

Typesofmeans

- Samplemean
- Populationmean

SampleMean

The sample mean is a central tendency measure. The arithmetic average is computed using samples or randomvalues taken from the population. It is evaluated as the sum of all the sample variables divided by the totalnumberof variables.

SAMPLE MEAN

$$\bar{X} = \frac{\sum X}{n}$$

PopulationMean

Thepopulationmeancanbecalculatedbythe sumofallvalues inthegivendata/populationdividedbyatotalnumberofvalues in thegiven data/population.

POPULATION MEAN

$$\mu = \frac{\sum X}{N}$$

AVERAGES FORQUALITATIVEANDRANKEDDATA

Mode

Themodealwayscanbe usedwithqualitativedata.

Median

Themediancanbeusedwheneveritispossibletoorderqualitativedatafromleasttomostbecausethelevelofmeasurement is ordinal.

DescribingVariability

RANGE

Therangeisthedifferencebetweenthelargest andsmallestscores.

Therangeinstatisticsforagivendatasetisthedifferencebetweenthehighestandlowestvalues.Forexample,if thegiven data set is{2,5,8,10,3}, then the rangewill be10– 2=8.

Example1:Findtherangeof

givenobservations:32,41,28,54,35,26,23,33,38,40.Solution:Let us

firstarrangethegivenvalues in ascendingorder.

23, 26, 28, 32, 33, 35, 38, 40, 41,54

Since23is thelowest valueand 54is thehighest value,therefore, therange oftheobservations willbe;Range

(X)=Max(X) –Min (X)

=54 – 23

=31

VARIANCE

Varianceisameasureofhowdatapointsdifferfromthemean.Avarianceisameasureofhowfarasetofdata(numbers) arespread out from their mean (average)value.

Formula

$\sigma = \Sigma(x-\mu)^2$ or

Variance=(Standarddeviation)²=σ² = >$\sigma^2 = \Sigma(x-\mu)^2/n$

thevalues ofallscores must beadded and thendivided bythetotal

numberofscores.Example

X=5, 8,6, 10, 12, 9, 11,10, 12, 7

Solution

Mean=sum(x)/nn

=10

sum(x)=5+8+6+10+12+9+11+10+12+7

=90

Mean=>μ=90/10=9Devi
ationfrommean

x-μ =-4, -1,-3,1, 3, 0, 2,1,3,-2

$(x-μ)^2$=16,1,9,1,9,0,4,1,9,4

$Σ(x-μ)^2$=16+1+9+1+9+0+4+1+9+4
    =54

$σ^2=Σ(x-μ)^2/n$

  =54/10
  =5.4

STANDARDDEVIATION

Thestandarddeviation, thesquarerootofthemean ofallsquared deviationsfrom themean, thatis,

Standarddeviation =√variance

StandardDeviation: Arough measureoftheaverage(orstandard) amount bywhich scores deviate

StandardDeviation:AMeasureof Distance
Themeanisameasureofposition,butthestandarddeviationisameasureofdistance(oneithersideofthemeanof
thedistribution).

SumofSquares (SS)

Calculatingthestandarddeviationrequiresthatweobtainfirstavalueforthevariance.However,calculatingthevarianc
erequires, in turn, that weobtain the sum of the squared deviationscores.
Thesum of squared deviation scoresor moresimplythe sumof squares, symbolized bySS

SUM OF SQUARES (*SS*) FOR POPULATION (DEFINITION FORMULA)

$$SS = \sum(X - \mu)^2 \hspace{3cm} (4.1)$$

"Thesumofsquaresequalsthesumofallsquareddeviationscores."Youcanreconstructthisformulabyrememberingthefollo
wingthreesteps:

1. Subtractthepopulationmean,μ,from eachoriginalscore,X,to obtainadeviationscore,X−μ.
2. Squareeachdeviationscore,(X−μ)2,toeliminate negativesigns.
3. Sumallsquareddeviationscores,Σ(X −μ)2.

VARIANCE FOR POPULATION

$$\sigma^2 = \frac{SS}{N} \hspace{3cm} (4.5)$$

STANDARD DEVIATION FOR POPULATION

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{SS}{N}} \hspace{3cm} (4.6)$$

Table 4.1
CALCULATION OF POPULATION STANDARD DEVIATION σ
(DEFINITION FORMULA)

A. COMPUTATION SEQUENCE
Assign a value to N **1** representing the number of X scores
Sum all X scores **2**
Obtain the mean of these scores **3**
Subtract the mean from each X score to obtain a deviation score **4**
Square each deviation score **5**
Sum all squared deviation scores to obtain the sum of squares **6**
Substitute numbers into the formula to obtain population variance, σ^2 **7**
Take the square root of σ^2 to obtain the population standard deviation, σ **8**

B. DATA AND COMPUTATIONS

X	**4** $X - \mu$	**5** $(X - \mu)^2$
13	3	9
10	0	0
11	1	1
7	−3	9
9	−1	1
11	1	1
9	−1	1

1 $N = 7$ **2** $\Sigma X = 70$ **6** $SS = \Sigma (X - \mu)^2 = 22$

3 $\mu = \dfrac{70}{7} = 10$

7 $\sigma^2 = \dfrac{SS}{N} = \dfrac{22}{7} = 3.14$ **8** $\sigma = \sqrt{\dfrac{SS}{N}} = \sqrt{\dfrac{22}{7}} = \sqrt{3.14} = 1.77$

Sum of Squares Formulas for Sample
Sample notation can be substituted for population notation in the above two formulas without causing any essential changes:

SUM OF SQUARES (SS) FOR SAMPLE (DEFINITION FORMULA)

$$SS = \Sigma(X - \bar{X})^2 \tag{4.3}$$

VARIANCE FOR SAMPLE

$$s^2 = \frac{SS}{n-1} \tag{4.7}$$

STANDARD DEVIATION FOR SAMPLE

$$s = \sqrt{s^2} = \sqrt{\frac{SS}{n-1}} \tag{4.8}$$

Table 4.3
CALCULATION OF SAMPLE STANDARD DEVIATION (S)
(DEFINITION FORMULA)

A. COMPUTATION SEQUENCE

Assign a value to n **1** representing the number of X scores
Sum all X scores **2**
Obtain the mean of these scores **3**
Subtract the mean from each X score to obtain a deviation score **4**
Square each deviation score **5**
Sum all squared deviation scores to obtain the sum of squares **6**
Substitute numbers into the formula to obtain the sample variance, s^2 **7**
Take the square root of s^2 to obtain the sample standard deviation, s **8**

B. DATA AND COMPUTATIONS

X	**4** $X - \bar{X}$	**5** $(X - \bar{X})^2$
7	4	16
3	0	0
1	−2	4
0	−3	9
4	1	1

1 $n = 5$ **2** $\Sigma X = 15$ **6** $SS = \Sigma(X - \bar{X})^2 = 30$

3 $\bar{X} = \dfrac{15}{5} = 3$

7 $s^2 = \dfrac{SS}{n-1} = \dfrac{30}{4} = 7.50$ **8** $s = \sqrt{\dfrac{SS}{n-1}} = \sqrt{\dfrac{30}{4}} = \sqrt{7.50} = 2.74$

Table 4.5
TWO ESTIMATES OF POPULATION VARIABILITY

WHEN μ IS UNKNOWN ($\bar{X} = 3$)			WHEN μ IS KNOWN ($\mu = 2$)		
X	$X - \bar{X}$	$(X - \bar{X})^2$	X	$X - \mu$	$(X - \mu)^2$
7	$7 - 3 = 4$	16	7	$7 - 2 = 5$	25
3	$3 - 3 = 0$	0	3	$3 - 2 = 1$	1
1	$1 - 3 = -2$	4	1	$1 - 2 = -1$	1
0	$0 - 3 = -3$	9	0	$0 - 2 = -2$	4
4	$4 - 3 = 1$	1	4	$4 - 2 = 2$	4

$\Sigma(X - \bar{X}) = 0$ $\Sigma(X - \bar{X})^2 = 30$ $\Sigma(X - \mu) = 5$ $\Sigma(X - \mu)^2 = 35$

$df = n - 1 = 5 - 1 = 4$ $df = n = 5$

$s^2 (df = n-1) = \dfrac{\Sigma(X - \bar{X})^2}{n-1} = \dfrac{30}{4} = 7.50$ $s^2 (df = n) = \dfrac{\Sigma(X - \mu)^2}{n} = \dfrac{35}{5} = 7.00$

DEGREESOFFREEDOM(*df*)

- Degrees of freedom (df) refers to the number of values that are free to vary, given one or moremathematicalrestrictions, in asample beingusedto estimateapopulation characteristic.
- Degrees of freedom are the number of independent variables that can be estimated in a statisticalanalysis.Thesevaluesofthesevariablesarewithoutconstraint,althoughthevaluesdoimpostrestrictionson othervariables if thedata set is to complywith estimateparameters.
- Degreesof Freedom(df) The number of valuesfree tovary,givenone or more mathematicalrestrictions.

Formula
Degreeoffreedom   **df= n-1**

Example
Consideradata setconsists offive positiveintegers. Thesumof thefive integersmust bethe multipleof6.Thevalues arerandomlyselected as3, 8, 5,and 4.
Thesumofthisforvaluesis20.Sowehavetochoosethefifthintegertomakethesumdivisibleby6.Thereforethefifth element is 10.

The number of degrees of Degrees of Freedom (df) The number of values free to vary, given one or moremathematical restrictions. Freedom—in the numerator, as in the formulas for s2 and s. In fact, we can usedegreesof freedomto rewritetheformulas for thesample varianceandstandard deviation:

VARIANCE FOR SAMPLE

$$s^2 = \frac{SS}{n-1} = \frac{SS}{df} \qquad (4.9)$$

STANDARD DEVIATION FOR SAMPLE

$$s = \sqrt{\frac{SS}{n-1}} = \sqrt{\frac{SS}{df}} \qquad (4.10)$$

INTERQUARTILERANGE(IQR)

The interquartile range (IQR), is simply the range for the middle 50 percent of the scores. More specifically,the IQR equals the distance between the third quartile (or 75th percentile) and the first quartile (or 25thpercentile), that is, after the highest quarter (or top 25 percent) and the lowest quarter (or bottom 25 percent)have been trimmed from the original set of scores. Since most distributions are spread more widely in theirextremitiesthan theirmiddle, the IQRtends to be less than half thesizeofthe range.

Simply, The IQR describes the middle 50% of values when ordered from lowest to highest. To find theinterquartile range (IQR), first find the median (middle value) of the lower and upper half of the data. Thesevaluesarequartile 1(Q1) and quartile 3(Q3). TheIQR isthe differencebetweenQ3and Q1.

Table 4.6
CALCULATION OF THE IQR

A. INSTRUCTIONS
1. Order scores from least to most.
2. To determine how far to penetrate the set of ordered scores, begin at either end, then add 1 to the total number of scores and divide by 4. If necessary, round the result to the nearest whole number.
3. Beginning with the largest score, count the requisite number of steps (calculated in step 2) into the ordered scores to find the location of the third quartile.
4. The third quartile equals the value of the score at this location.
5. Beginning with the smallest score, again count the requisite number of steps into the ordered scores to find the location of the first quartile.
6. The first quartile equals the value of the score at this location.
7. The IQR equals the third quartile minus the first quartile.

B. EXAMPLE
1. 7, 9, 9, 10, 11, 11, 13
2. $(7 + 1)/4 = 2$
3. 7, 9, 9, 10, 11, 11, 13
            ↑
         2 1

4. third quartile = 11
5. 7, 9, 9, 10, 11, 11, 13
         ↑
      1 2

6. first quartile = 9
7. IQR = 11 − 9 = 2

NormalDistributionsandStandard(z)Scores

THENORMALCURVE

The normal distribution is a continuous probability distribution that is symmetrical on both sides of the mean, so the right side of the center is a mirror image of the left side.

Propertiesof theNormalCurve
- The normal curve is a theoretical curve defined for a continuous variable, as described in Section 1.6, and noted for its symmetrical bell-shaped form, as revealed in below figure
- Because the normal curve is symmetrical, its lower half is the mirror image of its upper half.
- The normal curve peaks above a point midway along the horizontal spread and then tapers off gradually in either direction from the peak (without actually touching the horizontal axis, since, in theory, the tails of a normal curve extend infinitely far).
- The values of the mean, median (or 50th percentile), and mode, located at a point midway along the horizontal spread, are the same for the normal curve.

Propertiesofanormaldistribution
- The mean, mode and median are all equal.
- The curve is symmetric at the center (i.e. around the mean, μ).
- Exactly half of the values are to the left of center and exactly half the values are to the right.
- The total area under the curve is 1.

19
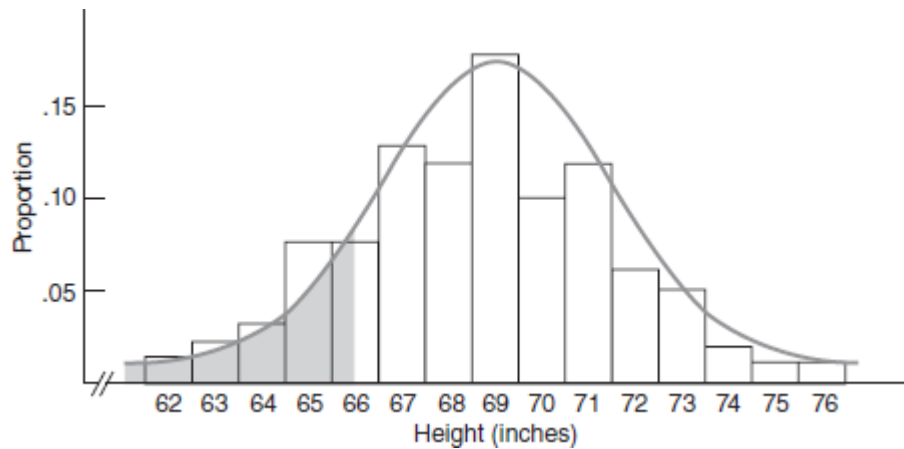
DifferentNormalCurves

As a theoretical exercise, it is instructive to note the various types of normal curves that are producedbyanarbitrarychangeinthe value ofeither themean (μ) orthe standard deviation (σ).

Obvious differences in appearance among normal curves are less important than you might suspect.Because of their common mathematical origin, every normal curve can be interpreted in exactly the same wayonceanydistancefrom themean is expressed in standard deviation units.

A. Different Means, Same Standard Deviation   B. Same Mean, Different Standard Deviations
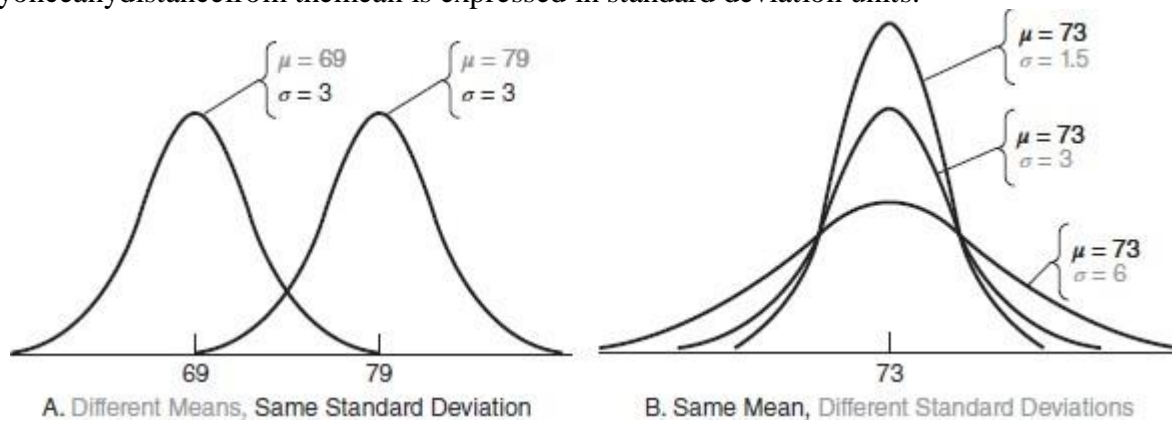
z SCORES

A z score is a unit-free, standardized score that, regardless of the original units of measurement, indicates howmanystandard deviations ascoreis aboveor below the mean of its distribution.

Azscorecanbedefinedasameasureofthenumberofstandarddeviationsbywhichascoreisbeloworabove the mean of a distribution. In other words, it is used to determine the distance of a score from the mean.If the z score is positive it indicates that the score is above the mean. If it is negative then the score will bebelowthe mean. However, ifthe z scoreis 0 it denotes that the data pointis the same as the mean.

To obtain a z score, express any original score, whether measured in inches, milliseconds, dollars, IQ points,etc., as a deviation from its mean (by subtracting its mean) and then split this deviation into standard deviationunits (bydividingbyits standard deviation),

$$z = \frac{X - \mu}{\sigma}$$

WhereXistheoriginalscoreandμandσarethemeanandthestandarddeviation,respectively,forthenormaldistribution oftheoriginalscores.Sinceidenticalunitsofmeasurementappearinboththenumerator

anddenominatoroftheratioforz,theoriginalunitsofmeasurementcanceleachotherandthezscoreemergesas a unit-freeorstandardized number,often referred toas astandardscore.

Azscoreconsists oftwoparts:
1. Apositive ornegativesignindicatingwhetherit's aboveor belowthe mean;and
2. Anumberindicatingthesizeofits deviationfrom themean instandarddeviationunits.

Converting   to   z ScoresExample
Suppose on a GRE test a score of 1100 is obtained. The mean score for the GRE test is1026 and thepopulation standard deviation is 209. In order to find how well a person scored with respect to the score of anaveragetest taker, thezscorewill haveto bedetermined.

Thesteps tocalculatethe z scoreareas follows:
- Step1: Write thevalueofthe raw scorein thezscoreequation.$z=(1100-\mu)/\sigma$
- Step2:Write themean andstandarddeviation ofthepopulationinthe zscoreformula.$z =(1100-1026)/209$
- Step3:Performthecalculationstogettherequiredz score.$z=0.345$
- Step 4: A z score table can be used to find the percentage of test-takers that are below the score of theperson. Using the first two digits of the z score, determine the row containing these digits of the ztable. Now using the 2nd digit after the decimal, find the corresponding column. The intersection ofthis row and column will give a value. As shown below, this value will be 0.6368 for the givenexample.
- Step 5: Use the value from step 5 and multiply it by 100 to get the required percentage. 0.6368 * 100 =63.68%.This showsthat 63.68%oftest-takers scores arelesser than the given raw score.

STANDARDNORMALCURVE
If the original distribution approximates a normal curve, then the shift to standard or z scores will alwaysproduce a new distribution that approximates the standard normal curve. This is the one normal curve forwhich atableis actuallyavailable**.**

Althoughthereisaninfinitenumberofdifferentnormalcurves,eachwithitsownmeanandstandarddeviation, thereis onlyonestandard normal curve,with a mean of 0andastandarddeviation of 1.

Forastandardnormal curve
    Mean=0

$$\text{Mean of } z = \frac{X-\mu}{\sigma} = \frac{\mu-\mu}{\sigma} = \frac{0}{\sigma} = 0$$

    Standarddeviation =1

$$\text{Standard deviation of } z = \frac{X-\mu}{\sigma} = \frac{\mu+1\sigma-\mu}{\sigma} = \frac{1\sigma}{\sigma} = 1$$

StandardNormalTable
Thestandardnormal tableconsistsof columnsof zscorescoordinated with columnsofproportions

UsingtheTopLegendof theTable
Notice that columns are arranged in sets of three, designated as A, B, and C in the legend at the top of thetable. When using the top legend, all entries refer to the upper half ofthe standard normal curve. The entriesincolumn A arezscores,beginningwith 0.00andendingwith 4.00

Given a z score of zero or more, columns B and C indicate how the z score splits the area in the upper half ofthenormalcurve.Assuggestedbytheshadinginthetoplegend,columnBindicatestheproportionofareabetweenthe meanandthezscore,andcolumnCindicatestheproportionofareabeyondthezscore,intheuppertail of thestandardnormal curve.

UsingtheBottomLegend of theTable
Now the columns are designated as A′, B′, and C′ in the legend at the bottom of the table. When using thebottomlegend, all entries referto thelower halfofthe standard normalcurve.
A negative z score, columns B′ and C′ indicate how that z score splits the lower half of the normal curve. Assuggested by the shading in the bottom legend of the table, column B′ indicates the proportion of area betweenthe mean and the negative z score, and column C′ indicates the proportion of area beyond the negative z score,inthelower tail of thestandard normal curve.

TABLE 5.1
PROPORTIONS (OF AREAS) UNDER THE STANDARD NORMAL CURVE FOR VALUES OF z (FROM TABLE A OF APPENDIX C)

A z	B	C	A z	B	C	A z	B	C
0.00	.0000	.5000	0.40	.1554	.3446	0.80	.2881	.2119
0.01	.0040	.4960	0.41	.1591	.3409	0.81	.2910	.2090
						0.99	.3389	.1611
						1.00	.3413	.1587
						1.01	.3438	.1562
0.38	.1480	.3520	0.78	.2823	.2711	1.18	.3810	.1190
0.39	.1517	.3483	0.79	.2852	.2148	1.19	.3830	.1170

FINDINGPROPORTIONS
FindingProportionsforOneScore
- Sketchanormal curveand shadeinthetarget area,
- Plan yoursolutionaccordingtothenormaltable.
- Convert*X*to*z*.

$$z = \frac{X - \mu}{\sigma}$$

Find: Proportion Below 66   **Solution:**

Target Area

.1587
(from column C')

66   69   *X*   −1.00   0   *z*
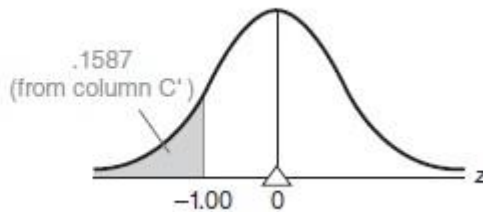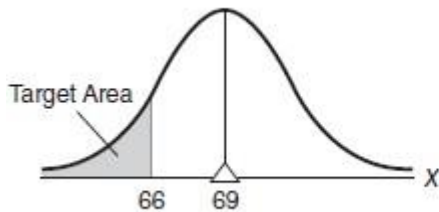
Answer: .1587

- Findthetargetarea.

FindingProportionsbetweenTwoScores
- Sketchanormalcurveandshadeinthetargetarea,(example,findproportionbetween245to 255)
- Plan yoursolutionaccordingtothenormaltable.
- Convert *X*to*z* byexpressing255 as

$$z = \frac{255 - 270}{15} = \frac{-15}{15} = -1.00$$

and by expressing 245 as

$$z = \frac{245 - 270}{15} = \frac{-25}{15} = -1.67$$

Find: Proportion Between 245 and 255

Target Area

245 255   270   *X*

Solution:

.1587
(from column C')

.0475
(from column C')

−

−1.00   0   *z*   −1.67   0   *z*

Answer: .1587
−.0475
.1112

- Findthetargetarea.

23

FINDINGSCORES

Sofar,wehaveconcentratedonnormalcurveproblemsforwhichTableAmustbeconsultedtofindtheunknown proportion(ofarea)associated with some known scoreorpair ofknown scores

NowwewillconcentrateontheoppositetypeofnormalcurveproblemforwhichTableAmustbeconsultedto find theunknown scoreor scores associatedwith some known proportion.

Forthistypeofproblemrequiresthatwereverseouruseof TableAbyenteringproportionsincolumnsB, C, B′, or C′and findingzscoreslistedincolumns A or A′.

FindingOneScore

- Sketchanormalcurveand,onthecorrectsideofthemean,drawalinerepresentingthetargetscore,as in figure
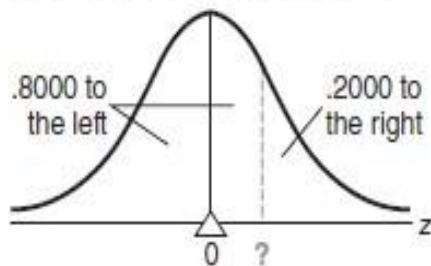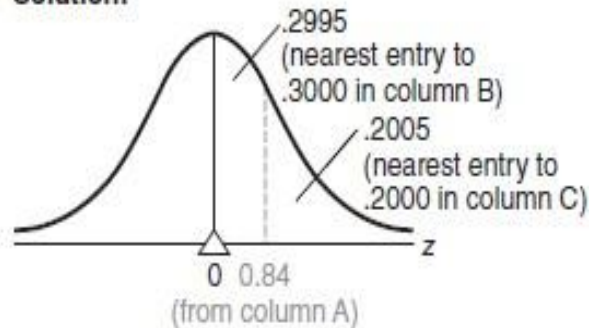
It's often helpful to visualize the target score as splitting the total area into two sectors—one to the left of(below)the target scoreand oneto theright of (above) thetarget score

- Plan yoursolution accordingtothenormaltable.

In problems of this type, you must plan how to find the z score for the target score. Because the target score ison the right side of the mean, concentrate on the area in the upper half of the normal curve, as described incolumnsBand C.

- Findz.
- Convertztothetargetscore.

When converting z scores to original scores, you will probably find it more efficient to use the followingequation

CONVERTING z SCORE TO ORIGINAL SCORE

$$X = \mu + (z)(\sigma)$$

FindingTwoScores

- Sketchanormalcurve.Oneithersideofthemean,drawtwolinesrepresentingthetwotargetscores,as infigure

Find: Pairs of Scores for the Extreme 2.5%
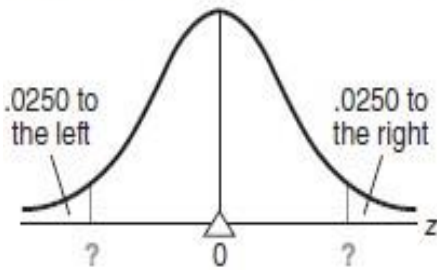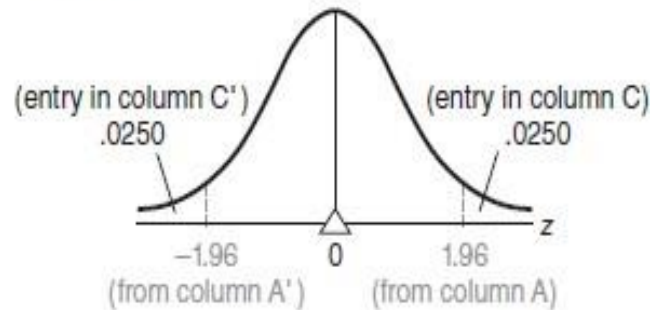
.0250 to the left   .0250 to the right

Solution:

(entry in column C') .0250   (entry in column C) .0250

−1.96 (from column A')   0   1.96 (from column A)

Answer: $X_{min} = \mu + (z)(\sigma)$
$= 22 + (-1.96)(4)$
$= 22 - 7.84$
$= 14.16$

Answer: $X_{max} = \mu + (z)(\sigma)$
$= 22 + (1.96)(4)$
$= 22 + 7.84$
$= 29.84$

- Plan yoursolutionaccordingtothenormaltable.
- Findz.
- Convertztothetargetscore.

PointstoRemember

1. range=largest value –smallest value in alist
2. classinterval= range/ desirednoofclasses
3. relative frequency= frequency(f)/ε(f)
4. Cumulativefrequency- *add tothefrequencyof eachclass thesumofthefrequenciesofallclassesranked belowit.*
5. Cumulativepercentage=(f/cumulativef)*100
6. Histograms
7. Constructionoffrequencypolygon
8. Stem and leaf display
9. Mode-*Thevalueof the most frequentscore.*
10. Forodd no ofterms **Median={(n+1)/2}thterm/observation**.Foreven noof terms**Median = 1/2[(n/2)thterm+ {(n/2)+1}thterm]**
11. Mean=sumof all scores/numberofscores

SAMPLE MEAN   **POPULATION MEAN**

$\bar{X} = \dfrac{\sum X}{n}$ $\mu = \dfrac{\sum X}{N}$

Variance $\sigma = \sum(x-\mu)^2$ or
Variance=(Standarddeviation)^2=σ^2 = >σ^2=$\sum(x-\mu)^2$/n

12. Range(X) =Max (X)–Min(X)

STANDARD DEVIATION FOR POPULATION

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{SS}{N}}$$

(4.6)

SUM OF SQUARES (SS) FOR POPULATION (DEFINITION FORMULA)

$$SS = \Sigma(X - \mu)^2$$

(4.1)

VARIANCE FOR SAMPLE

$$s^2 = \frac{SS}{n-1}$$

(4.7)

STANDARD DEVIATION FOR SAMPLE

$$s = \sqrt{s^2} = \sqrt{\frac{SS}{n-1}}$$

(4.8)

SUM OF SQUARES (SS) FOR SAMPLE (DEFINITION FORMULA)

$$SS = \Sigma(X - \bar{X})^2$$

(4.3)

13. Degreeoffreedom   **df= n-1**

14. Typesofnormalcurve

A. Different Means, Same Standard Deviation   B. Same Mean, Different Standard Deviations
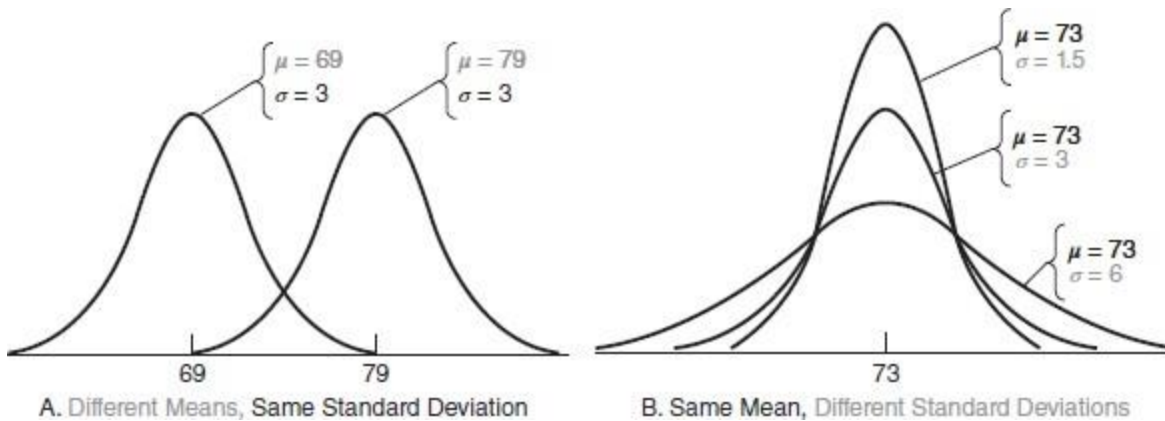
15. z –score

z SCORE

$$z = \frac{X - \mu}{\sigma}$$

(5.1)

26

16. Standardnormalcurve; **mean=0,standarddeviation=1**

17. Findingproportion

$$z = \frac{X - \mu}{\sigma}$$

18. Findingproportion

    1. Foronescore

Find: Proportion Below 66   **Solution:**

Target Area

.1587
(from column C')

66 69   X   −1.00 0   z
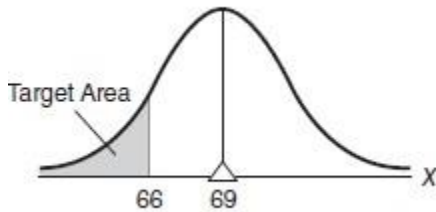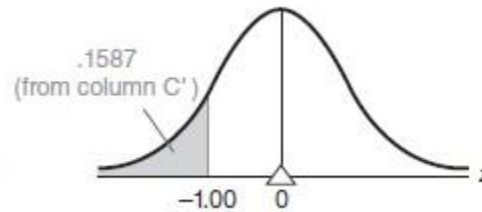
Answer: .1587

    2. Forbetween twoscore

Find: Proportion Between 245 and 255

Target Area

245 255 270   X

Solution:

.1587
(from column C')

−1.00 0   z

.0475
(from column C')

−1.67 0   z
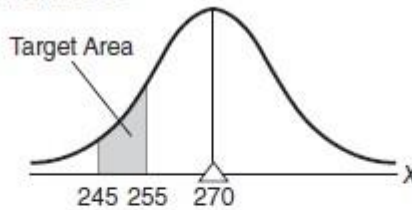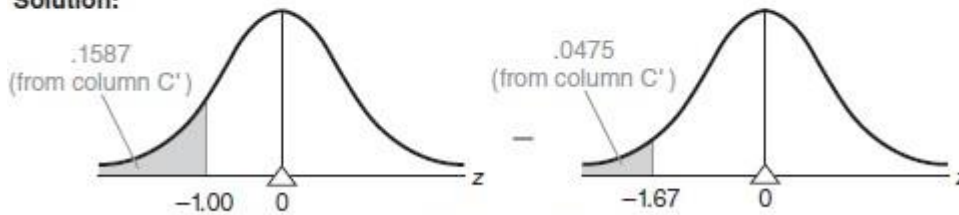
Answer: .1587
− .0475
.1112

19. Findingscores

CONVERTING z SCORE TO ORIGINAL SCORE

$$X = \mu + (z)(\sigma)$$

20. Findingscores – one score

Find: Lowest Score in Upper 20%   **Solution:**

.8000 to
the left   .2000 to
the right

0 ?   z

.2995
(nearest entry to
.3000 in column B)

.2005
(nearest entry to
.2000 in column C)

0 0.84   z
(from column A)
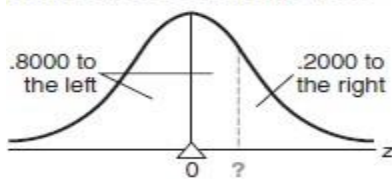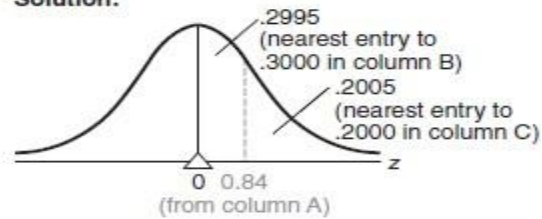
Answer: $X = \mu + (z)(\sigma)$
$= 230 + (0.84)(50)$
$= 230 + 42$
$= 272$

Twoscores

Find: Pairs of Scores for the Extreme 2.5%   **Solution:**
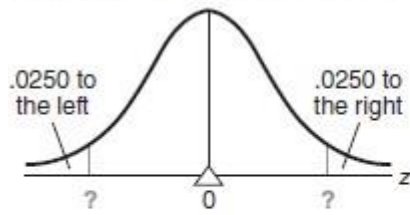
.0250 to
the left   .0250 to
                                  the right

?   0   ?   z

(entry in column C')   (entry in column C)
   .0250   .0250

−1.96   0   1.96   z
(from column A')   (from column A)

Answer: $X_{min} = \mu + (z)(\sigma)$
$= 22 + (-1.96)(4)$
$= 22 - 7.84$
$= 14.16$

Answer: $X_{max} = \mu + (z)(\sigma)$
$= 22 + (1.96)(4)$
$= 22 + 7.84$
$= 29.84$
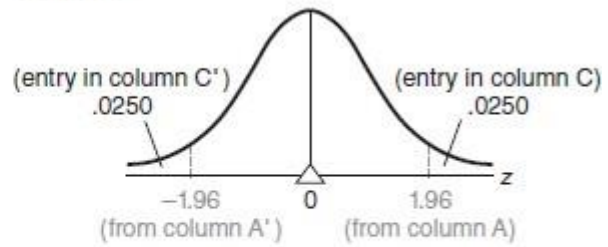
Unit –
IIIDESCRIBINGRELATIONSHI
PS

Correlation – Scatter plots – correlation coefficient for quantitative data – computational formula for correlationcoefficient–Regression–regressionline–leastsquaresregressionline–Standarderrorofestimate– interpretationof r2 – multipleregression equations– regression towardsthemean

Correlation

Correlation refers to a process for establishing the relationships betweentwo variables. You learned a way toget a general idea about whether or not two variables are related, is to plot them on a "scatter plot". While therearemanymeasuresofassociationforvariableswhicharemeasuredattheordinalorhigherlevelofmeasurement,correlation is the most commonlyused approach.

TypesofCorrelation

- **Positive Correlation** – when the values of the two variables move in the same direction so that anincrease/decrease in the value of one variable is followed by an increase/decrease in the value of theothervariable.
- **Negative Correlation** – when the values of the two variables move in the opposite direction so that anincrease/decrease in the value of one variable is followed by decrease/increase in the value of the othervariable.
- **NoCorrelation**– when thereisno linear dependenceor norelation between the two variables.
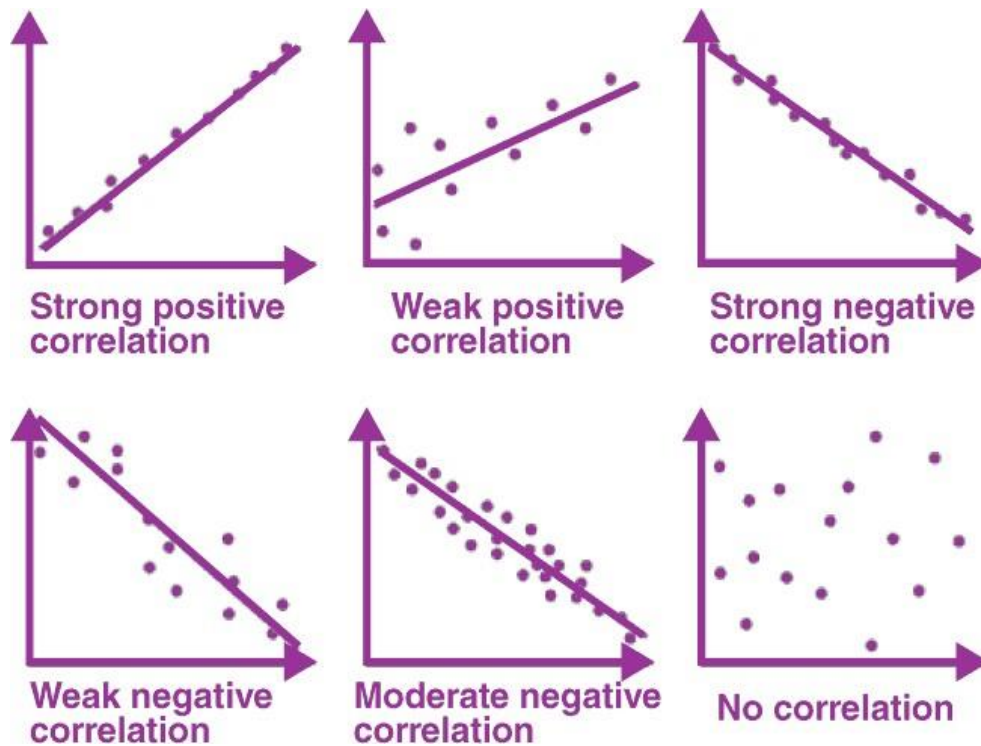
Table 6.2 THREE TYPES OF RELATIONSHIPS		
A. POSITIVE RELATIONSHIP		
FRIEND	**SENT**	**RECEIVED**
Doris	13	14
Steve	9	18
Mike	7	12
Andrea	5	10
John	1	6
B. NEGATIVE RELATIONSHIP		
FRIEND	**SENT**	**RECEIVED**
Doris	13	6
Steve	9	10
Mike	7	14
Andrea	5	12
John	1	18
C. LITTLE OR NO RELATIONSHIP		
FRIEND	**SENT**	**RECEIVED**
Doris	13	10
Steve	9	18
Mike	7	12
Andrea	5	6
John	1	14

SCATTERPLOTS

A scatter plot is a graph containing a cluster of dots that represents all pairs of scores. In other wordsScatterplotsarethegraphsthatpresenttherelationshipbetweentwovariablesinadata- set.Itrepresentsdatapointson atwo-dimensional planeor onaCartesian system.

Constructionofscatterplots

- Theindependent variable or attributeis plottedon theX-axis.   Fig6.1
- Thedependent variableis plotted onthe Y-axis.

1
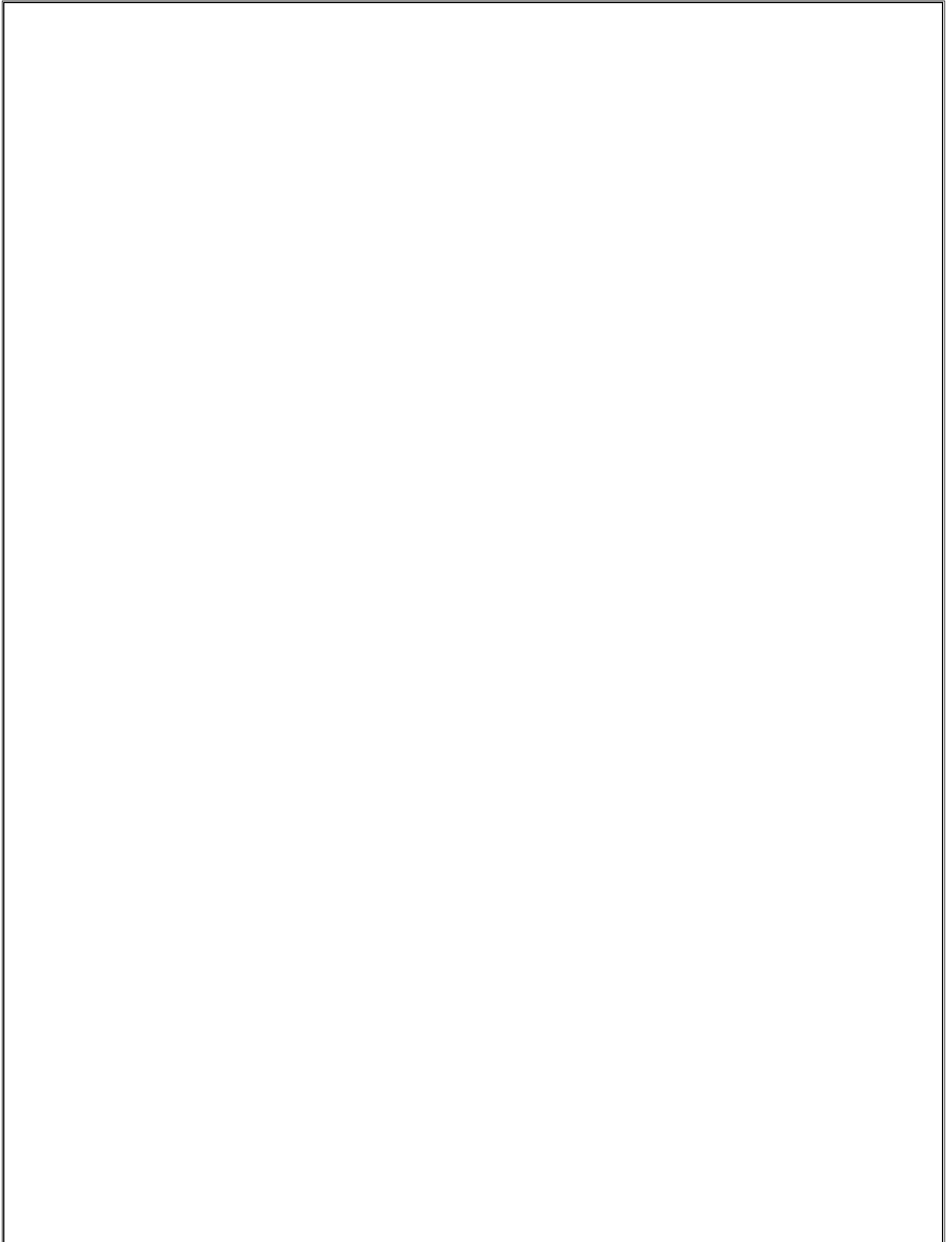
- Useeachpair ofscores tolocate adot withinthescatterplot

Positive,Negative,orLittleorNo Relationship?

Thefirst stepis tonote thetilt or slope,if any,of adot cluster.
Adotclusterthathasaslopefromthelowerlefttotheupperright,asinpanelAofbelowfigurereflectsa **positiverelationship.**

Adotclusterthathasaslopefromtheupperlefttothelowerright,asinpanelBofbelowfigurereflectsa **negativerelationship.**

Adotcluster thatlacks anyapparent slope,asin panelC ofbelow figurereflects **littleorno relationship.**
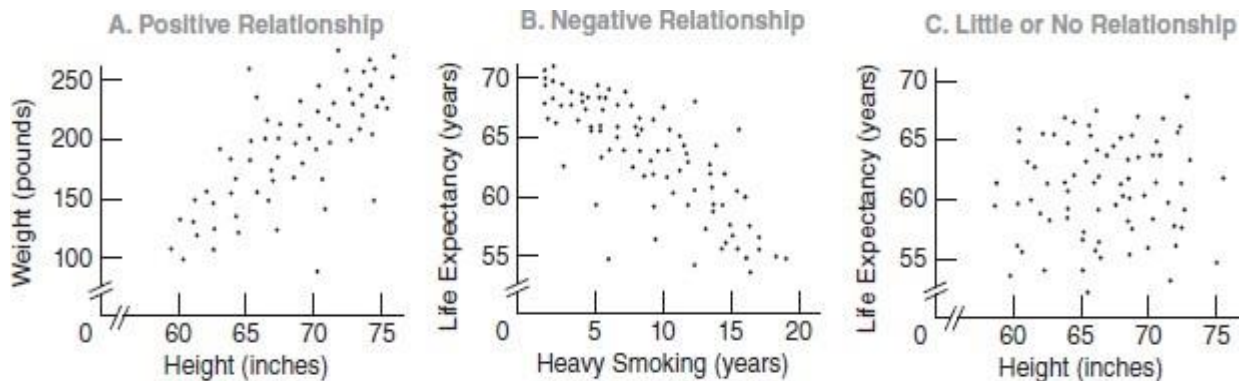
PerfectRelationship
A dot cluster that equals (rather than merely approximates) a straight line reflects a perfect relationship betweentwovariables.

CurvilinearRelationship
The previous discussion assumes that a dot cluster approximates a straight line and, therefore, reflects a **linearrelationship.** But this is not always the case. Sometimes a dot cluster approximates a bent or curved line, as inbelowfigure,and thereforereflectsa**curvilinearrelationship.**
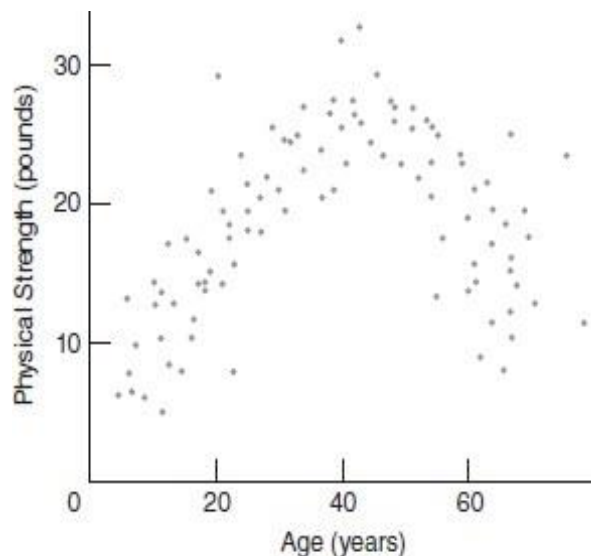
ACORRELATIONCOEFFICIENTFORQUANTITATIVEDATA: *r*

Thecorrelationcoefficient,r,isasummarymeasurethatdescribestheextentofthestatisticalrelationshipbetween twointerval or ratio level variables.

Propertiesofr

- Thecorrelationcoefficientisscaledsothat itisalwaysbetween -1and+1.
- When r is close to 0 this means that there is little relationship between the variables and the farther awayfrom 0 r is, in either the positive or negative direction, the greater the relationship between the twovariables.
- Thesignof rindicates thetypeoflinearrelationship, whetherpositiveornegative.
- Thenumericalvalueof r, without regardto sign,indicatesthestrengthofthelinearrelationship.
- A number with a plus sign (or no sign) indicates a positive relationship, and a number with a minus signindicatesanegativerelationship

COMPUTATIONFORMULAFOR*r*

Calculate avalueforrbyusingthe followingcomputation formula:

CORRELATION COEFFICIENT (COMPUTATION FORMULA)

$$r = \frac{SP_{xy}}{\sqrt{SS_x SS_y}}$$

Wherethetwo sumofsquares termsin thedenominatoraredefinedas

$$SS_x = \Sigma\left(X - \bar{X}\right)^2 = \Sigma X^2 - \frac{\left(\Sigma X\right)^2}{n}$$

$$SS_y = \Sigma\left(Y - \bar{Y}\right)^2 = \Sigma Y^2 - \frac{\left(\Sigma Y\right)^2}{n}$$

Thesumof theproductsterminthe numerator,*SPxy*, isdefinedin belowformula

$$SP_{xy} = \Sigma\left(X - \bar{X}\right)\left(Y - \bar{Y}\right) = \Sigma XY - \frac{\left(\Sigma X\right)\left(\Sigma Y\right)}{n}$$

Orthe formulais writtenas

$$r = \frac{n\left(\sum xy\right) - \left(\sum x\right)\left(\sum y\right)}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right]\left[n\sum y^2 - \left(\sum y\right)^2\right]}}$$

Wheren =Number of Information

Σx= Totalofthe

FirstVariableValueΣy=TotaloftheSecon

d VariableValue

Σxy=SumoftheProductoffirst&Second ValueΣx^2

=Sum of the Squaresof theFirst Value

Σy^2 =Sumof theSquares of theSecond Value
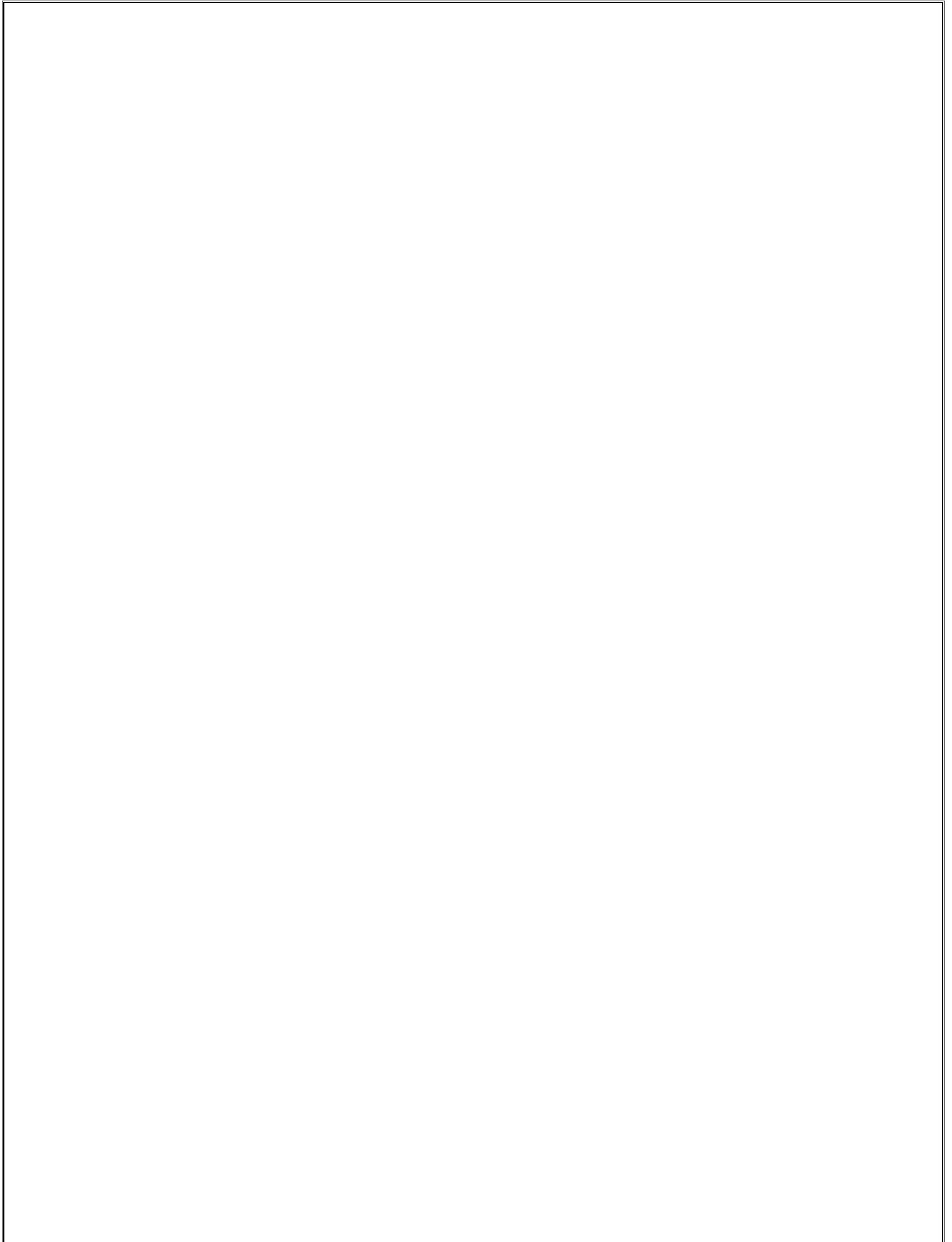
Table 6.3
CALCULATION OF r: COMPUTATION FORMULA

A. COMPUTATIONAL SEQUENCE

Assign a value to n (1), representing the number of pairs of scores.
Sum all scores for X (2) and for Y (3).
Find the product of each pair of X and Y scores (4), one at a time, then add all of these products (5).
Square each X score (6), one at a time, then add all squared X scores (7).
Square each Y score (8), one at a time, then add all squared Y scores (9).
Substitute numbers into formulas (10) and solve for SP_{xy}, SS_x, and SS_y.
Substitute into formula (11) and solve for r.

B. DATA AND COMPUTATIONS

		CARDS	4	6	8
FRIEND	SENT, X	RECEIVED, Y	XY	X^2	Y^2
Doris	13	14	182	169	196
Steve	9	18	162	81	324
Mike	7	12	84	49	144
Andrea	5	10	50	25	100
John	1	6	6	1	36

(1) $n = 5$ (2) $\Sigma X = 35$ (3) $\Sigma Y = 60$ (5) $\Sigma XY = 484$ (7) $\Sigma X^2 = 325$ (9) $\Sigma Y^2 = 800$

(10) $SP_{xy} = \Sigma XY - \dfrac{(\Sigma X)(\Sigma Y)}{n} = 484 - \dfrac{(35)(60)}{5} = 484 - 420 = 64$

$SS_x = \Sigma X^2 - \dfrac{(\Sigma X)^2}{n} = 325 - \dfrac{(35)^2}{5} = 325 - 245 = 80$

$SS_y = \Sigma Y^2 - \dfrac{(\Sigma Y)^2}{n} = 800 - \dfrac{(60)^2}{5} = 800 - 720 = 80$

(11) $r = \dfrac{SP_{xy}}{\sqrt{SS_x SS_y}} = \dfrac{64}{\sqrt{(80)(80)}} = \dfrac{64}{80} = .80$

REGRESSION

A regression is a statistical technique that relates a dependent variable to one or more independent(explanatory) variables. A regression model is able to show whether changes observed in the dependent variableareassociated with changes in oneor moreof the explanatoryvariables.

Regression captures the correlation between variables observed in a data set, and quantifies whetherthosecorrelations arestatisticallysignificant or not.

ARegression Line

aregressionlineisalinethatbestdescribesthebehaviourofasetofdata.Inotherwords,it'salinethatbestfitsthetrend ofagiven data.
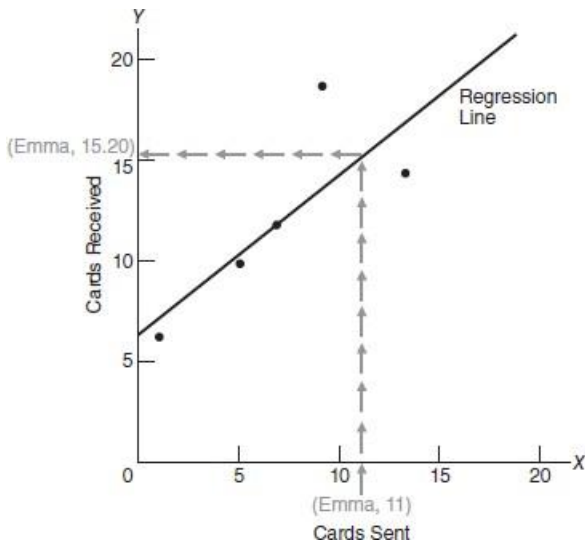
The purpose of the line is to describe the interrelation of adependentvariable(Yvariable)withoneormanyindependentvariables(X variable).By using the equationobtainedfromtheregressionlineananalystcanforecastfuturebehavioursofthedependentvariablebyinputtingdifferentvalues fortheindependent ones.

Typesofregression
Thetwobasictypesof regressionare
- Simplelinearregression
Simplelinearregressionusesoneindependentvariabletoexplainorpredictthe outcome ofthe dependent variableY
- Multiplelinearregression
Multiplelinearregressionsusetwoormoreindependentvariablesto predict theoutcome

PredictiveErrors
    Predictionerrorreferstothedifferencebetweenthepredictedvaluesmadebysomemodelandtheactualvalues.
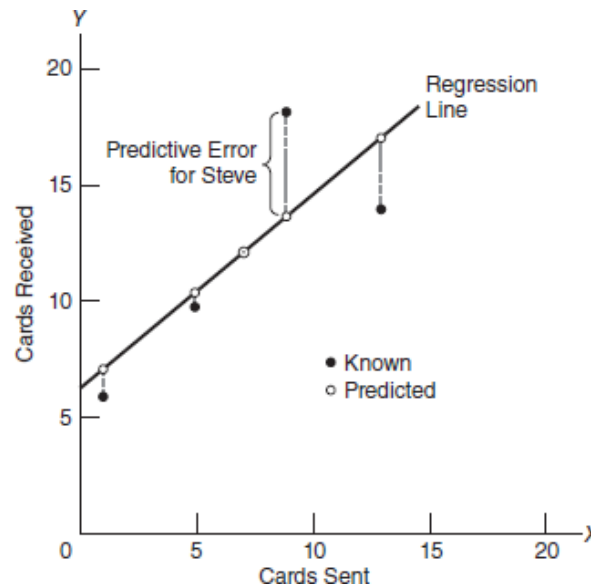
LEASTSQUARESREGRESSIONLINE
        The placement of the regression line minimizes not the total predictive error but the total squaredpredictive error, that is, the total for all squared predictive errors. When located in this fashion, the regressionlineis often referred to as the leastsquaresregression line.
        The Least Squares Regression Line is the line that minimizes the sum of the residuals squared. Theresidual is the vertical distance between the observed point and the predicted point, and it is calculated bysubtracting ŷ fromy.

Formula

y'=bx+a   b– slope , a–yintercept

$$b=\frac{N\Sigma(xy)-\Sigma x \Sigma y}{N\Sigma(x^2)-(\Sigma x)^2}$$

$b = \underline{\Sigma y - m\Sigma x}$
        N

Example

"x"   "y"
2   4
3   5
5   7
7   10
9   15

Step1: Foreach (x,y)calculatex^2andxy:

x	y	x^2	xy
2	4	4	8
3	5	9	15
5	7	25	35
7	10	49	70
9	15	81	135

Step2: Sumx,y, x2 and xy(gives usΣx, Σy,$\Sigma x2$ and Σxy):

Σx: 26Σy: 41Σx²:168Σxy: 263

Step3: CalculateSlope b

$b = \dfrac{N \Sigma(xy) - \Sigma x}{\Sigma y N\Sigma(x^2) - (\Sigma x)^2}$

$= \dfrac{5 \times 263 - 26 \times 41}{5 \times 168 - 262}$

$= \dfrac{1315 - 1066}{840 - 676}$

$= \dfrac{249}{164}$

b=1.5183.

Step 4: Calculate Intercept

$aa = \dfrac{\Sigma y - b\Sigma x}{N}$

$= \dfrac{41 - 1.5183 \times 26}{5}$

a=0.3049.

Step5:y'=bx+a
        y'=1.518x+0.305
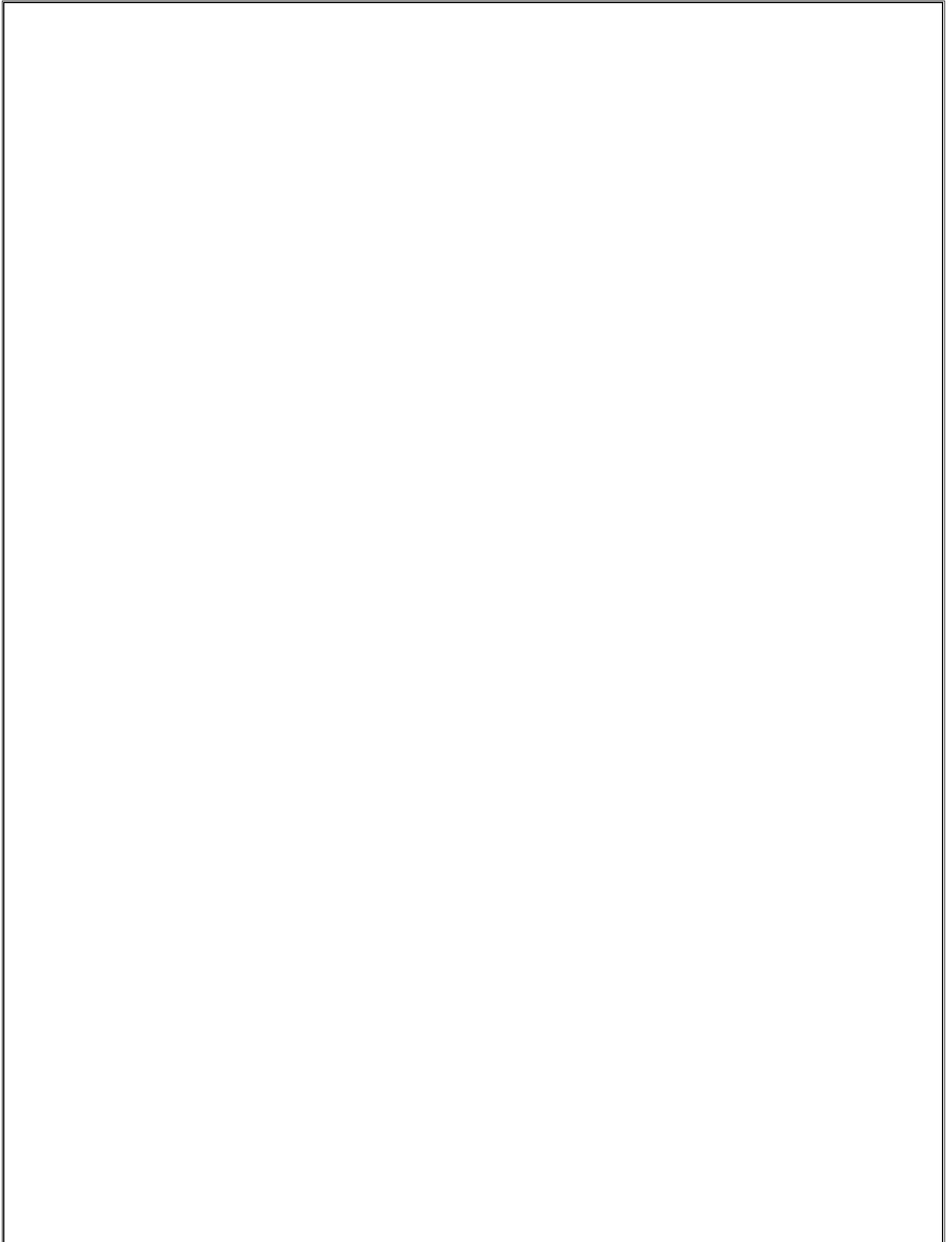
x	y	y=1.518x+0.305	error
2	4	3.34	−0.66
3	5	4.86	−0.14
5	7	7.89	0.89
7	10	10.93	0.93
9	15	13.97	−1.03

Topredicttheyvaluewecanassume
anyvalueforx.Assume x=8.
Theny=1.518 x8+0.305
=12.45

STANDARDERROROFESTIMATE ,$s_{y|x}$

The standard error of the estimate is a measure of the accuracy of predictions. The regression line is theline that minimizes the sum of squared deviations of prediction (also called the sum of squares error), and thestandarderrorof theestimateis thesquare root ofthe averagesquareddeviation.

The standard error of estimate and symbolized as $s_{y|x}$, this estimate of predictive error complies withthegeneralformatforanysamplestandarddeviation,thatis,thesquarerootofasumofsquarestermdividedbyits degrees offreedom.

$$SS_{ylx} = \Sigma(Y - Y')^2$$
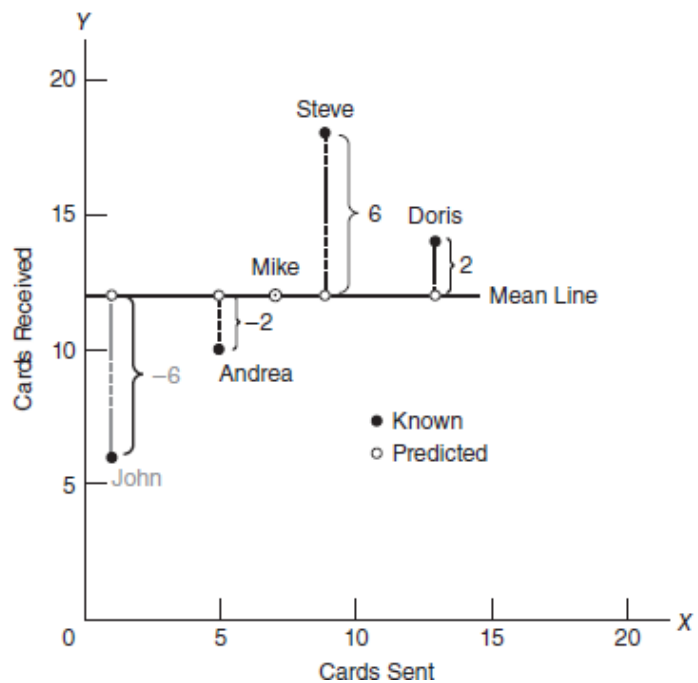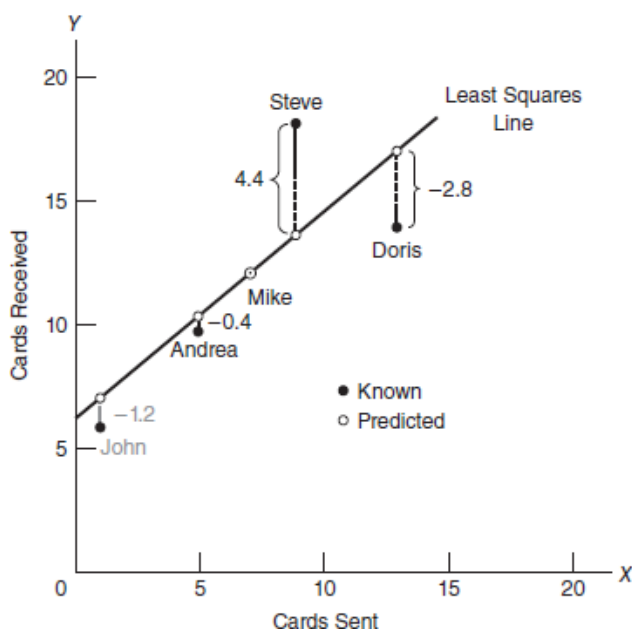
Fig.Predictiveerrorsfor fivefriends

Example

Calculatethestandard errorof estimatefor thegiven Xand Yvalues. X=1,2,3,4,5Y=2,4,5,4,5

Solution
Createfivecolumnslabeledx, y,y',y–y',(y–y')2andN=5

x	y	x^2	xy	Y'= bx+a	y-y'	(y–y')^2
1	2	1	2	2.8	-0.8	0.64
2	4	4	8	3.4	0.6	0.36
3	5	9	15	4.0	1	1
4	4	16	16	4.6	-0.6	0.36
5	5	25	25	5.2	-0.2	0.04
Σx:15	**Σy:20**	**Σx²:55**	**Σxy:66**			**Σ(y–y')^2 = 2.4**

Note:for findingb valuewehaveto find xy and x^2, so add xy andx^{2}column intable

b =$\underline{N \, \Sigma(xy) - \Sigma x}$
    $\Sigma y N\Sigma(x^2)$
    $-(\Sigma x)^2$

b=$\underline{5(66)-15x20}$
   $5(55)-(15)^2$

$=\underline{330- \, 300}$
  275-225

b= 30/50= 0.6

a=$\underline{\Sigma y- \, b\Sigma x}$
        N
 =$\underline{20- \, (0.6 \, x15)}$
            5
 = $\underline{20- \, 11}$
        5
 a= 9/5 = 2.2

SSy/x= $\sqrt{((y-y')^2 /n-2)}$

=√(2.4/3)

SSy/x=0.894

INTERPRETATIONOF*r$_2$*
        R-Squared (R² or the coefficient of determination) is a statistical measure in a regression model thatdetermines the proportion of variance in the dependent variable that can be explained by the independentvariable. Inother words,r-squaredshows howwell thedatafit theregression model(thegoodness of fit).
        R-squared can take any values between 0 to 1. Although the statistical measure provides some usefulinsights regarding the regression model, the user should not rely only on the measure in the assessment of astatisticalmodel.

In addition, it does not indicate the correctness of the regression model. Therefore, the user shouldalwaysdrawconclusionsaboutthemodelbyanalyzingr-squaredtogetherwiththeothervariablesinastatisticalmodel.

Themostcommoninterpretationofr-squaredishowwelltheregressionmodelexplainsobserved data.

$$r^2 = \frac{SS_{Y'}}{SS_Y} = \frac{SS_Y - SS_{Y|X}}{SS_Y}$$

$$SS_{y'} = \sum(Y' - \overline{Y})^2$$

MULTIPLEREGRESSIONEQUATIONS

Multiple regression is a statistical technique applied on datasets dedicated to draw out a relationshipbetweenoneresponseordependent variable andmultiple independent variables.

Multiple regression works by considering the values of the available multiple independent variables andpredictingthevalue of onedependent variable.

Example:
A researcher decides tostudy students' performance from a school over a period of time. He observed thatasthe lectures proceed to operate online, the performance of students started to decline as well. The parameters forthe dependentvariable"decrease inperformance" are various independentvariables like "lack of attention,moreinternet addiction, neglectingstudies"andmuch more.

Formulatofindmultipleregression
$$y= b1x1 + b2x2 + …bnxn + a$$

REGRESSIONTOWARDTHEMEAN

Regressiontowardthemeanreferstoatendencyforscores,particularlyextremescores,to shrinktowardthemean.
In statistics, regression toward the mean (also called reversion to the mean, and reversion to mediocrity) is aconceptthat refersto thefactthat ifonesampleof arandom variableis extreme,thenext samplingof thesamerandomvariableis likelyto be closer to its mean.

Example
A military commander has two units return, one with 20% casualties and another with 50% casualties. Hepraises the first and berates the second. The next time, the two units return with the opposite results. From thisexperience,he"learns"that praiseweakens performanceandberatingincreasesperformance.

TheRegressionFallacy

The regression fallacy is committed whenever regression toward the mean is interpreted as a real, ratherthanachance,effect.
Theregression fallacycan beavoided bysplittingthe subsetof extremeobservationsintotwo groups

Table 7.4
REGRESSION TOWARD THE MEAN: BATTING AVERAGES OF TOP 10 HITTERS IN MAJOR LEAGUE BASEBALL DURING 2014 AND HOW THEY FARED DURING 2015

TOP 10 HITTERS (2014)	BATTING AVERAGES*		REGRESS TOWARD MEAN?
	2014	2015	
1. J. Alture	.341	.313	Yes
2. V. Martinez	.335	.282	Yes
3. M. Brantley	.327	.310	Yes
4. A. Beltre	.324	.287	Yes
5. J. Abreu	.317	.290	Yes
6. R. Cano	.314	.287	Yes
7. A. McCutchen	.314	.292	Yes
8. M. Cabrera	.313	.338	No
9. B. Posey	.311	.318	No
10. B. Revere	.306	.306	No

UNITIV
PYTHONLIBRARIES FORDATAWRANGLING

Basicsof Numpy arrays–aggregations–computations on arrays–comparisons, masks, booleanlogic–fancyindexing–structuredarrays–DatamanipulationwithPandas–dataindexingandselection–operatingondata–missingdata– Hierarchicalindexing –combiningdatasets –aggregationandgrouping–pivot tables

NumPy(shortforNumerical Python)providesanefficientinterfacetostoreandoperate ondensedatabuffers.NumPy arrays are like Python's built-in list type, but NumPy arrays provide much more efficient storage anddataoperations as the arraysgrowlarger in size.

We'llcoverafewcategoriesofbasicarraymanipulationshere:

Attributesofarrays
> Determiningthesize,shape,memoryconsumption,anddatatypesof arrays

Indexingofarrays
> Gettingand settingthe value ofindividual arrayelements

Slicingofarrays
> Gettingandsettingsmallersubarrays withinalargerarray

Reshapingofarrays
> Changingthe shapeof agivenarray

Joiningandsplittingofarrays
> Combiningmultiple arraysinto one, andsplittingonearrayinto many

NumPyArrayAttributes

- ndim(thenumberof dimensions),
- shape(thesizeof each dimension)
- size(thetotal sizeof thearray)

Example

```
np.random.seed(0)#seed forreproducibility
x1=np.random.randint(10,size=6)#One-dimensionalarray
x2=np.random.randint(10,size=(3,4))#Two-dimensionalarray
x3=np.random.randint(10,size=(3,4,5))#Three-dimensionalarray
print("x3      ndim:      ",
x3.ndim)print("x3  shape:",
x3.shape)print("x3size:
",x3.size)

print("dtype:",x3.dtype)

print("itemsize:",x3.itemsize,"bytes")p
rint("nbytes:",x3.nbytes,"bytes")
```
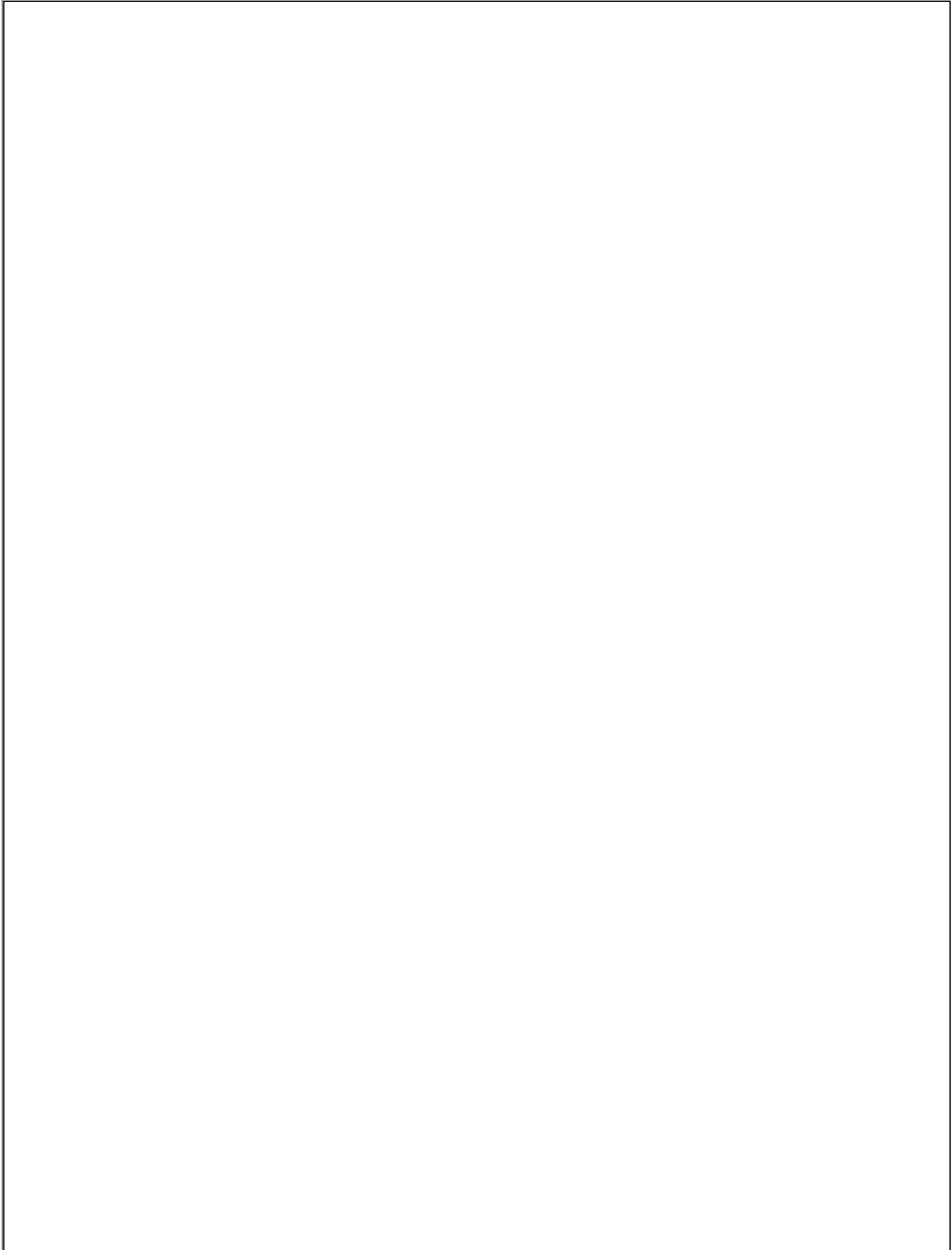
ArrayIndexing:
- AccessingSingleElements

AccessingSingleElements
- Indexingin NumPywillfeel quitefamiliarlikelist indexing,

- In aone-dimensional array,youcan accesstheithvalue(countingfromzero)byspecifyingthedesiredindex in squarebrackets,just as with Python lists
- Toindexfromtheendof thearray,youcanusenegativeindices
- In amultidimensional array,you accessitemsusingacomma-separatedtupleofindices
- Unlike Python lists, NumPy arrays have a fixed type. This means, for example, that if you attempt toinserta floating-point valueto an integerarray,thevalue will besilentlytruncated.

ArraySlicing:AccessingSubarrays

Just as we can use square brackets to access individual array elements, we can also use them to access subarrayswiththeslicenotation, marked bythecolon (:) character.
TheNumPyslicingsyntaxfollows that ofthe standard Python list; to accessa sliceofan arrayx, usethis:

x[start:stop:step]
start – startingarrayindex
stop– arrayindextostop (lastvaluewillnot
beconsidered)step– terms has to beprinted from start to stop
Defaulttothevaluesstart=0, stop=sizeof dimension,step=1.

Example
x =
np.arange(10)x

array([0,1, 2, 3, 4, 5,6, 7, 8, 9])

x[:5]*#printsfirstfiveelements*
array([0,1,2,3, 4])

x[5:]*#elementsafterindex5*
array([5,6,7,8, 9])

x[4:7]*#middle subarray(from4th indexto6thindex)*
array([4,5,6])

Whileusingnegativeindicesthe defaults forstartand stopareswapped. Thisbecomes
aconvenientwaytoreverseanarray
x[::-1]*#allelements,reversed*
array([9,8, 7, 6, 5, 4,3, 2, 1, 0])

x[5::-2]*#reversedeveryotherfromindex5*
array([5,3,1])

Multidimensionalsubarrays

Multidimensionalslicesworkinthesameway,withmultipleslicesseparatedbycommas.Forexa
mple:
x2
array([[12,5,2, 4],
       [7,6,8,8],
       [1,6,7,7]])

2

x2[:2, :3] # two rows, three
columnsarray([[12,5,2],
       [7,6,8]])

x2[:3, ::2] # all rows, every other column(every second
column)array([[12,2],
       [7,8],
       [1,7]])
Finally,subarraydimensionscan evenbereversedtogether
x2[::-1, ::-1]
array([[7,7,6,1],
       [8,8, 6,7],
       [4,2,5,12]])

ReshapingofArrays
Themostflexiblewayofdoingthis iswith the**reshape()**method. For example, if you wanttoput
thenumbers1through 9 in a 3×3grid,you can do thefollowing
grid = np.arange(1, 10).reshape((3,
3))print(grid)
       [[12 3]
       [456]
       [7 89]]

ArrayConcatenationandSplitting
Concatenationofarrays
Concatenation, or joining of two arrays in NumPy, is primarily accomplished through the
routines**np.concatenate, np.vstack,** and **np.hstack.** np.concatenate takes a tuple or list of arrays as its first
argument.x=np.array([1,2,3])
y = np.array([3, 2,
1])np.concatenate([x,
y])

array([1,2,3,3,2, 1])

Youcanalsoconcatenatemorethantwoarraysatonce
       z =[99,99, 99]
       print(np.concatenate([x, y,

z]))[123321999999]

np.concatenatecanalsobeusedfortwo-dimensionalarrays
grid =np.array([[1,2,3],
                        [4,5, 6]])
np.concatenate([grid,grid])

array([[1,2, 3],
       [4,5,6],
       [1,2,3],

3

```
                    [4,5,6]])
```

Concatenatealongthe secondaxis (zero-indexed)
```
          np.concatenate([grid,grid],axis=1)
```

```
array([[1,2,3,1,2,3],
        [4,5,6,4,5,6]])
```

np.vstack(verticalstack)functions

```
x=np.array([1,2,3])
grid =np.array([[9,8,7],
                       [6,5,4]])
np.vstack([x,grid])
```

```
array([[1,2, 3],
        [9,8, 7],
        [6,5,4]])
```

np.hstack(horizontalstack)functions

```
y=np.array([[99],
               [99]])
np.hstack([grid,y])
```

```
array([[ 9,8,7, 99],
        [6,5,4,99]])
```

Splittingofarrays
Theoppositeofconcatenationissplitting, whichisimplementedbythefunctions**np.split,np.hsplit**,and
np.vsplit.For eachof these, wecan passa list ofindicesgivingthe split points
```
x=[1,2,3,99,99,3,2,1]
x1, x2, x3 = np.split(x, [3,
5])print(x1, x2,x3)
```

```
[1 23][9999][321]
```

Noticethat Nsplit pointslead toN +1subarrays.Therelatedfunctions np.hsplitand np.vsplitaresimilar
```
grid = np.arange(16).reshape((4,
4))grid
array([[ 0,1,2,3],
        [ 4,5,6,7],
        [8,9, 10,11],
        [12,13,14,15]])
```

```
upper, lower=np.vsplit(grid,[2])
print(upper)
print(lower)
```

```
[[012 3]
[456 7]]
[[891011]
[12131415]]

left,right=np.hsplit(grid,[2])
print(left)p
rint(right)

[[01]
[45]
[89]
[1213]]
[[23]
[67]
[1011]
[1415]]
```

ComputationonNumPyArrays:UniversalFunctions
IntroducingUFuncs
NumPyprovidesaconvenientinterfaceintojustthiskindofstaticallytyped,compiledroutine.Thisisknownasavectorized operation.

VectorizedoperationsinNumPy are implemented via ufuncs,whosemainpurpose istoquickly executerepeated operations on values in NumPy arrays. Ufuncs are extremely flexible—before we saw an operationbetweenascalar andan array, but wecanalso operate betweentwo arrays

ExploringNumPy'sUFuncs

Ufuncs exist in two flavors: unary ufuncs, which operate on a single input, and binary ufuncs, which operate ontwoinputs.We'll seeexamplesof both thesetypesof functionshere.

Arrayarithmetic
NumPy'sufuncsmakeuseofPython'snativearithmeticoperators.Thestandardaddition,subtraction,multiplication,and divisioncan all beused.

```
x=np.arange(4)print(
"x =", x)print("x + 5
=", x + 5)print("x - 5
=", x -
5)print("x*2=",x*2)
```

Operator	Equivalentufunc	Description
+	np.add	Addition(e.g.,1+1=2)
-	np.subtract	Subtraction(e.g.,3-2=1)
-	np.negative	Unarynegation (e.g.,-2)
*	np.multiply	Multiplication(e.g.,2*3=6)
/	np.divide	Division(e.g.,3 /2 =1.5)

5

//	np.floor_divide	Floordivision (e.g.,3 //2 =1)
**	np.power	Exponentiation(e.g.,2 **3 =8)
%	np.mod	Modulus/remainder(e.g.,9%4 =1)

Absolutevalue
JustasNumPyunderstandsPython'sbuilt-inarithmeticoperators,italsounderstandsPython'sbuilt-inabsolutevaluefunction.

- np.abs()
- np.absolute()

```
x = np.array([-2, -1, 0, 1,
2])abs(x)
```

```
array([2, 1,0,1,2])
```

ThecorrespondingNumPyufuncis np.absolute,which is also available under thealias np.abs
```
np.absolute(x)arra
y([2,1,0,1,2])
```

```
np.abs(x)
array([2, 1,0,1,2])
```

Trigonometricfunctions
NumPyprovidesalargenumberofusefulufuncs,andsomeofthemostusefulforthedatascientistarethetrigonometric functions.

- np.sin()
- np.cos()
- np.tan()

inversetrigonometricfunctions

- np.arcsin()
- np.arccos()
- np.arctan()

Defininganarrayofangles:theta=np.linspace(0,np.pi, 3)
Computesometrigonometricfunctions like
```
print("theta = ",
theta)print("sin(theta) = ",
np.sin(theta))print("cos(theta) = ",
np.cos(theta))print("tan(theta)=",n
p.tan(theta))
```

Exponentsandlogarithms
Anothercommon typeofoperation availablein aNumPyufuncaretheexponentials.

- np.exp(x) – calculateexponent ofall elementsin theinput arrayie e^x(e=2.7182)
- np.exp2(x)–calculate 2**x forallxbeingthe arrayelements
- np.power(x,y) – calculates thepower as x^y

```
x=[1,2, 3]
print("x =",x)
print("e^x=",np.exp(x))
```

```
print("2^x=",np.exp2(x))
print("3^x =",np.power(3,x))
```

Theinverseoftheexponentials,thelogarithms,arealsoavailable.Thebasicnp.loggivesthenaturallogarithm;if you prefer to computethe base-2logarithm or thebase-10 logarithmas.

- np.log(x)-isamathematicalfunction thathelpsusertocalculateNatural logarithmofx wherexbelongsto all theinput arrayelements
- np.log2(x)-tocalculateBase-2logarithmofx
- np.log10(x)-tocalculateBase-10logarithmofx

```
x=[1, 2,4, 10]
print("x =",x)
print("ln(x)=",np.log(x))
print("log2(x)=",np.log2(x))
print("log10(x)=",np.log10(x))
```

Specializedufuncs

NumPyhasmanymore ufuncs availablelike

- Hyperbolictrigfunctions,
- Bitwisearithmetic,
- Comparisonoperators,
- Conversionsfromradianstodegrees,
- Roundingand remainders,and muchmore

Morespecializedandobscureufuncsisthesubmodulescipy.special.Ifyouwanttocomputesomeobscuremathematicalf unction onyour data, chancesareitisimplemented in scipy.special.

- Gammafunction

AdvancedUfuncFeatures

Specifyingoutput

Ratherthancreatingatemporaryarray,youcanusethistowritecomputationresultsdirectlytothememorylocationwhereyou'dl ikethemtobe.Forallufuncs,youcandothisusingthe outargumentofthefunction.

```
x =
np.arange(5)y=n
p.empty(5)
np.multiply(x,10,out=y)
print(y)
```

[0.10. 20. 30. 40.]
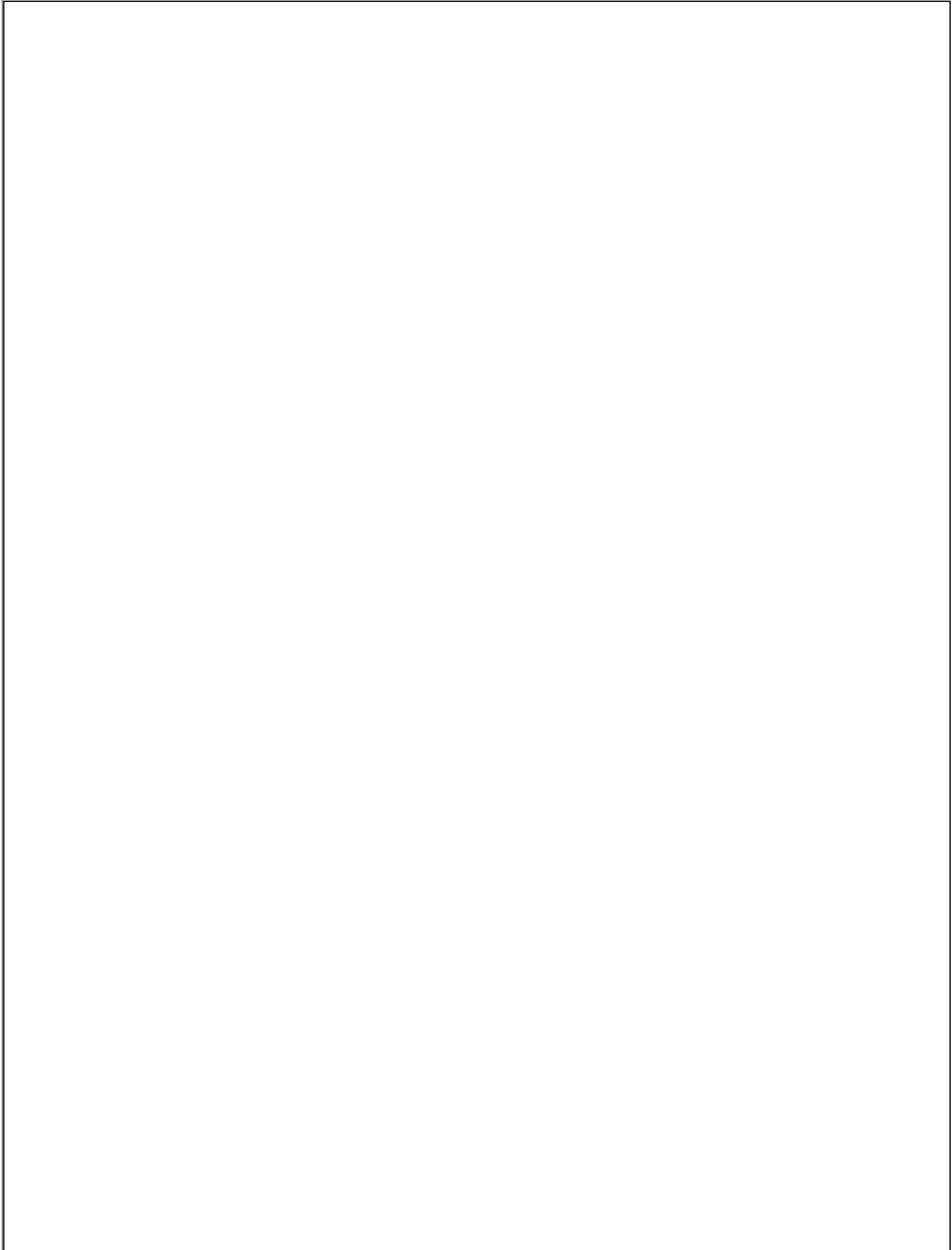
Aggregates

Toreduceanarraywithaparticular operation,wecanusethereducemethodofanyufunc.Areducerepeatedlyappliesagiven operationto theelements of anarrayuntil onlyasingle result remains.

```
x = np.arange(1,
6)np.add.reduce(x)
```

Similarly,callingreduce onthemultiplyufuncresultsintheproductofallarrayelements

```
np.multiply.reduce(x)
120
```

If we'dliketostorealltheintermediateresultsofthecomputation, wecaninsteaduseAccumulate
np.add.accumulate(x)a
rray([1,3,6,10,15])

Outerproducts
ufunc can compute the output of all pairs of two different inputs using the outer method. This allows you, in oneline,to do things likecreateamultiplication table.

x = np.arange(1,
6)np.multiply.outer(x,
x)

array([[1,2, 3,4,5],
        [2, 4, 6, 8, 10],
        [3, 6, 9, 12, 15],
        [4, 8, 12, 16, 20],
        [5,10, 15, 20, 25]])

Aggregations:Min,Max,andEverythinginBetween
MinimumandMaximum
Pythonhasbuilt-inminandmaxfunctions,usedtofindtheminimumvalueandmaximumvalueof anygivenarray.
Formin,max,sum,andseveralotherNumPyaggregates,ashortersyntaxistousemethodsofthearrayobj ectitself.
   • np.min() –finds theminimum(smallest)valueinthe array
   • np.max()–finds

themaximum(largest)valueinthearrayExample

x=[1,2,3,4]
np.min(x)
1
np.max(x)
4
Multidimensionalaggregates
Onecommontypeofaggregation operationisanaggregatealongaroworcolumn.
Bydefault,eachNumPyaggregationfunctionwillreturntheaggregateovertheentirearray.ie.Ifweusethenp.sum()it will calculates the sum of all elementsof thearray.

Example
        m=np.random.random((3,4))
        print(M)

        [[0.89675760.03783739   0.75952519   0.06682827]
         [0.8354065   0.99196818   0.19544769   0.43447084]
         [0.668593070.15038721   0.37911423   0.6687194]]

M.sum()6.085055566
7307118

Aggregation functions take an additional argument specifying the axis along which the aggregate is computed.Theaxisnormallytakeseither0or1.iftheaxis=0thenitrunsalongwithcolumns,ifaxis=1itrunsalongwith rows.

Example
Wecan find theminimum valuewithineachcolumn byspecifyingaxis=0

M.min(axis=0)
array([0.66859307, 0.03783739, 0.19544769, 0.06682827])

Similarly,wecan findthemaximumvaluewithineach
rowM.max(axis=1)
array([0.8967576 , 0.99196818,0.6687194])

Otheraggregationfunctions
NumPy provides many other aggregation functions most aggregates have a NaN-safe counterpart thatcomputes the result while ignoring missing values, which are marked by the special IEEE floating-point NaNvalue.

FunctionName	NaN-safeVersion	Description
np.sum	np.nansum	Computesumofelements
np.prod	np.nanprod	Computeproductofelements
np.mean	np.nanmean	Computemedian ofelements
np.std	np.nanstd	Computestandarddeviation
np.var	np.nanvar	Computevariance
np.min	np.nanmin	Findminimumvalue
np.max	np.nanmax	Findmaximum value
np.argmin	np.nanargmin	Findindexofminimumvalue
np.argmax	np.nanargmax	Findindexofmaximumvalue
np.median	np.nanmedian	Computemedian ofelements
np.percentile	np.nanpercentile	Computerank-basedstatisticsofelements
np.any	N/A	Evaluatewhether anyelementsaretrue
np.all	N/A	Evaluatewhetherallelementsaretrue

ComputationonArrays:Broadcasting
Broadcastingissimplyasetofrulesforapplyingbinaryufuncs(addition,subtraction,multiplication,etc.)on arrays ofdifferent sizes.
For arrays of the same size, binary operations are performed on an element-by-element
basis.a=np.array([0, 1, 2])
b = np.array([5, 5,
5])a+b

array([5,6,7])
Broadcasting allows these types of binary operations to be performed on arrays of different
sizes.a+5

array([5,6,7])

Wecanthinkofthisasanoperationthatstretchesorduplicatesthevalue5intothearray[5,5,5],andaddstheresults.Theadvantage of NumPy'sbroadcastingisthatthisduplicationofvalues doesnotactuallytakeplace.

Wecansimilarlyextendthistoarraysofhigherdimension.Observetheresultwhenweaddaone-dimensionalarraytoatwo-dimensional array.

Example
     M=np.ones((3,3))
     M

      array([[1., 1.,1.],
            [1., 1., 1.],
            [1.,1., 1.]])

     M+a

      array([[1.,2.,3.],
            [1., 2., 3.],
            [1.,2., 3.]])
Heretheone-dimensionalarrayisstretched,orbroadcast,acrosstheseconddimensioninordertomatchtheshapeofM.

Justasbeforewestretchedorbroadcastedonevaluetomatchtheshapeoftheother,herewe'vestretchedbothaand b to matcha common shape, and the result is a two dimensionalarray.
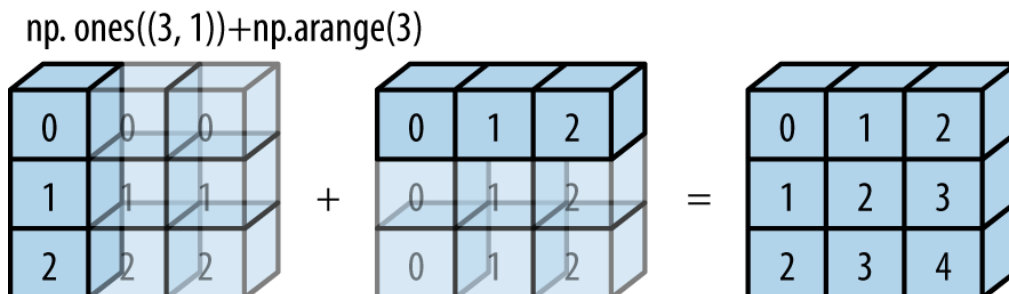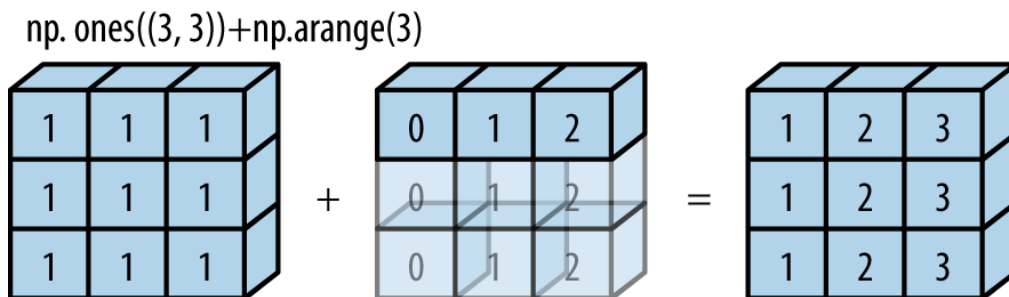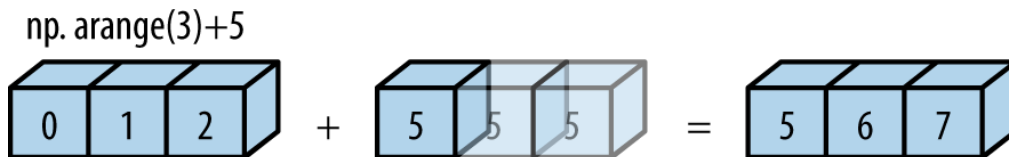
Thelightboxes representthebroadcasted values:again,this extramemoryis notactuallyallocated inthe courseoftheoperation, but it can beuseful conceptuallyto imaginethat it is.

Rulesof Broadcasting
BroadcastinginNumPyfollowsastrictset ofrules todeterminetheinteractionbetween thetwo arrays.

• Rule1:Ifthetwoarraysdifferintheir numberofdimensions,theshapeoftheonewithfewerdimensionsispadded with ones on its leading (left)side.
• Rule2:Iftheshapeof thetwo arraysdoesnotmatch inanydimension,the arraywithshapeequalto 1in thatdimensionis stretched to match theothershape.
• Rule3:Ifinanydimensionthesizesdisagree and neitherisequalto 1,anerrorisraised.

Broadcastingexample1
Let'slook ataddingatwo-dimensional arrayto aone-dimensionalarray:

        M = np.ones((2,
        3))a=np.arange(3)

Let'sconsider
        anoperationonthesetwoarrays.Theshapesofthearraysare:M.shape=(2,
        3)
        a.shape= (3,)
Weseebyrule1 that thearrayahas fewer dimensions, so wepad it onthe left with
        ones:M.shape->(2, 3)
        a.shape->(1,3)
By rule 2, we now see that the first dimension disagrees, so we stretch this dimension to
        match:M.shape-> (2, 3)
        a.shape->(2,3)
Theshapes match,andweseethatthe final shapewill be(2, 3):
        M+a
         array([[1.,2.,3.],
                [1.,2., 3.]])

Broadcastingexample2
Let'stakealookatanexamplewherebotharrays
        needtobebroadcast:a=np.arange(3).reshape((3, 1))
        b=np.arange(3)

Again,we'llstartbywritingouttheshapeofthearrays:a.sha
        pe=(3, 1)
        b.shape=(3,)

Rule1sayswemustpadtheshapeofbwithones:a.shap
        e->(3, 1)
        b.shape-> (1,3)

Andrule2 tellsusthat weupgrade each oftheseones tomatchthecorrespondingsizeof theotherarray:a.shape-
        >(3, 3)
        b.shape-> (3,3)

Becausethe resultmatches,theseshapesarecompatible.Wecanseethishere:a+b
```
array([[0,1,2],
        [1, 2,3],
        [2, 3, 4]])
```

Comparisons,Masks,andBooleanLogic

ComparisonOperatorsasufuncs.

Wesawthatusing+,-,*,/,andothersonarraysleadstoelement-
wiseoperations.NumPyalsoimplementscomparisonoperators such as <(less than) and > (greater than) aselement-
wiseufuncs.
Theresultofthese comparisonoperatorsisalways anarraywitha
Booleandatatype.Allsixof thestandardcomparison operations areavailable:

```
x = np.array([1, 2, 3, 4,
5])x<3#less than
array([ True, True, False, False, False],
dtype=bool)x>3#greater than
array([False,False,False,True,True],dtype=bool)

x<=3#lessthan orequal
array([True,True,True, False, False],dtype=bool)

x>=3#greater thanorequal
array([False,False,True, True,True],dtype=bool)

x!=3#not equal
array([True,True,False, True,True],dtype=bool)

x==3#equal
array([False,False,True,False,False],dtype=bool)
```

Operator	Equivalentufunc
==	np.equal
!=	np.not_equal
<	np.less
<=	np.less_equal
>	np.greater
>=	np.greater_equal

Justasinthecaseofarithmeticufuncs,thesewillworkonarraysofanysizeandshape.Hereisatwo-dimensionalexample

```
rng =
np.random.RandomState(0)x=rng
.randint(10,size=(3,4))
x

  array([[5,0,3,3],
         [7, 9, 3,5],
         [2, 4, 7, 6]])
```

x<6

array([[True, True, True,
        True],[False,False,True,T
        rue],
        [True,True, False,False]],dtype=bool)

TheresultisaBooleanarray,andNumPyprovidesanumberofstraightforwardpatternsforworkingwiththeseBooleanresults.

WorkingwithBooleanArrays
- np.count_nonzero()
- np.sum()
- np.sum(x,axis)
- np.any()
- np.all()
- np.all(x ,axis)

Booleanoperators
Operator

	Equivalentufunc
&	np.bitwise_and
|	np.bitwise_or
^	np.bitwise_xor
~	np.bitwise_not

Example

np.sum((inches>0.5)&(inches<1))inc
hes>(0.5&inches)<1
np.sum(~((inches<=0.5) |(inches>=1)))

BooleanArrays as Masks
A more powerful pattern is to use Boolean arrays as masks, to select particular subsets of the data
themselves.Returningtoourxarrayfrombefore,supposewewantanarrayofallvaluesinthearraythatarelessthan,say,5
Wecanobtain   a
Booleanarrayforthisconditioneasily,aswe'vealreadyseenExample

x
array([[5,0,3,3],
        [7, 9, 3,5],
        [2, 4, 7, 6]])

x<5
array([[False, True, True,
        True],[False,False,True,F
        alse],
        [True,True, False,False]],dtype=bool)

Maskingoperation
Toselectthesevaluesfromthearray,wecansimplyindexonthisBooleanarray;thisisknownasamaskingoperation.
x[x<5]

array([0,3,3, 3,2,4])
What isreturnedisa one-dimensionalarrayfilledwithallthe values thatmeetthiscondition;inother words,allthevalues in positions at which themask arrayisTrue.

FancyIndexing

Fancyindexingislikethesimpleindexingwe'vealreadyseen,butwepassarraysofindicesinplaceofsinglescalars . Thisallows usto veryquicklyaccess andmodifycomplicated subsetsof anarray'svalues.

ExploringFancyIndexing
Fancyindexingisconceptuallysimple:itmeanspassinganarrayofindicestoaccessmultiplearrayelementsatonce. Typesof fancyindexing.

- Indexing/ accessingmorevalues
- Arrayof indices
- Inmultidimensional
- Standardindexing

Example
```
importnumpyasnp
rand =
np.random.RandomState(42)x =
rand.randint(100, size=10)print(x)
```

[51 92 14 71 60 20 82 8674 74]

Indexing/accessingmorevalues
Supposewewantto accessthreedifferent elements.Wecoulddo it
        likethis:[x[3],x[7],x[2]]

    [71, 86, 14]

Arrayof indices
We can pass a single list or array of indices to obtain the same
        result.ind =[3, 7, 4]
        x[ind]

        array([71,86,60])

Inmultidimensional
Fancyindexingalso works in multipledimensions. Consider thefollowing array.
        X = np.arange(12).reshape((3,
        4))X

    array([[0,1,2, 3],
            [4, 5, 6, 7],
            [8,9, 10, 11]])

Standardindexing
Like with standard indexing, the first index refers to the row, and the second to the
        column.row=np.array([0, 1, 2])
        col=np.array([2, 1,3])

X[row,col]array(

[2,5,11])

CombinedIndexing

For even more powerful operations, fancy indexing can be combined with the other indexing schemes we'veseen.

Examplearray

```
print(X)
        [[ 0123]
        [4567]
        [ 8910 11]]
```

- Combinefancyand simple indices
  ```
  X[2,[2,0, 1]]
  array([10,8, 9])
  ```

- Combinefancyindexing with slicing
  ```
  X[1:,[2,0, 1]]

  array([[ 6,4,5],
  [10,8,9]])
  ```
- Combine fancy indexing with
  ```
  maskingmask = np.array([1, 0, 1, 0],
  dtype=bool)X[row[:,np.newaxis],mask]

  array([[ 0,2],
          [ 4,6],
          [8,10]])
  ```

ModifyingValueswithFancyIndexing

Justas fancyindexingcan beused to accessparts ofanarray, itcanalso be used tomodifyparts ofanarray.Changesomevalue in anarray

Modifyparticularelementbyindex

Forexample,imaginewehaveanarrayof indicesandwe'dliketosetthe correspondingitemsinanarraytosomevalue.

```
x= np.arange(10)
i = np.array([2,1,8, 4])
x[i] =99
print(x)
```

[09999399 567999]

Usingassignmentoperator

Wecanuseanyassignment-typeoperatorforthis.Forexample

```
x[i] -=10
print(x)
```

[08989389 567899]

Usingat()
Usethe at()methodof ufuncsforotherbehavior of modifications.

```
x             =
np.zeros(10)np.
add.at(x,       i,
1)print(x)
```

[0.0.1. 2.3.0.0. 0.0.0.]

SortingArrays

SortinginNumPy:np.sortandnp.argsort
Python has built-in sort and sorted functions to work with lists, we won't discuss them here because NumPy'snp.sort function turns out to be much more efficient and useful for our purposes. By default np.sort uses an O[*N*log *N]*, quicksort algorithm, though mergesort and heapsort are also available. For most applications, the defaultquicksort is morethan sufficient.

Sortingwithoutmodifyingtheinput.
Toreturn asorted version ofthearraywithoutmodifyingtheinput,you canusenp.sort

```
x=
np.array([2,1,4,3,5])np.so
rt(x)
```

```
array([1, 2,3,4,5])
```

Returnssortedindices
Arelated functionis argsort,which insteadreturnsthe*indices*ofthesortedelements

```
x= np.array([2,1,4,3,5])i
=np.argsort(x)
print(i)
```

```
[1 0324]
```

Sortingalongrowsorcolumns
A useful feature of NumPy's sorting algorithms is the ability to sort along specific rows or columns of amultidimensionalarrayusingthe axis argument. For example

```
rand =
np.random.RandomState(42)X =
rand.randint(0, 10, (4, 6))print(X)
```

```
[[637469]
[267437]
[725417]
[5 14095]]
```

```
np.sort(X,

axis=0)array([[2,1,4,0,
```

1,5],

16

```
        [5,2,5,4,3, 7],
        [6,3,7,4,6, 7],
        [7,6,7,4, 9,9]])
```

np.sort(X,

axis=1)array([[3,4,6,6,

7,9],

```
        [2,3,4,6,7,7],
        [1,2,4,5,7,7],
        [0,1,4,5,5,9]])
```

PartialSorts:Partitioning

Sometimes we're not interested in sorting the entire array, but simply want to find the K smallest values in thearray. NumPy provides this in the np.partition function. np.partition takes an array and a number K; the result isa new array with the smallest K values to the left of the partition, and the remaining values to the right, inarbitraryorder

```
        x= np.array([7,2, 3,1,6,5, 4])
        np.partition(x,3)
```

```
        array([2, 1,3,4,6,5,7])
```

Note that the first three values in the resulting array are the three smallest in the array, and the remaining arraypositionscontain the remainingvalues. Within thetwo partitions, theelements havearbitraryorder.

Partitioninginmultidimensionalarray

Similarlyto sorting,we canpartition alongan arbitraryaxis ofamultidimensionalarray.

np.partition(X,2, axis=1)

array([[3,4,6,7,6,9],

```
        [2,3,4,7,6,7],
        [1,2,4,5,7,7],
        [0,1,4,5,9,5]])
```

StructuredArrays

ThissectiondemonstratestheuseofNumPy'sstructuredarraysandrecordarrays,whichprovideefficientstoragefor compound,heterogeneous data.

NumPydata types

Character	DescriptionExample
'b'	Bytenp.dtype('b')
'i'	Signedintegernp.dtype('i4') ==np.int32
'u'	Unsignedintegernp.dtype('u1')==np.uint8
'f'	Floatingpointnp.dtype('f8') ==np.int64
'c'	Complex floatingpointnp.dtype('c16')==np.complex128
'S', 'a'	stringnp.dtype('S5')
'U'	
	Unicodestringnp.dtype('U')==np.str_'V
'Rawdata (void)	np.dtype('V') ==np.void

Consider if we have several categories of data on a number of people (say, name, age, and weight), and we'dlike to store these values for use in a Python program.Itwould be possible to store these in three separatearrays.

```
name = ['Alice', 'Bob', 'Cathy',
'Doug']age =[25,45,37,19]
weight =[55.0, 85.5,68.0,61.5]
```

Creatingstructuredarray
NumPycanhandlethisthroughstructuredarrays,whicharearrayswithcompounddatatypes.createastructuredarrayusingacompound datatypespecificationas follows.

```
data=np.zeros(4,dtype={'names':('name','age','weight'),
                                'formats':('U10','i4','f8')})

print(data.dtype)
[('name', '<U10'), ('age', '<i4'), ('weight',
'<f8')]U10-Unicodestring
ofmaximumlength10i4-4-byte(i.e., 32 bit)
integer
f8-8-byte(i.e., 64bit)float
```

Nowwecan fill the arraywith our listsofvalues
```
    data['name'] =
    namedata['age'] =
    agedata['weight'] =
    weightprint(data)
```

[('Alice',25,55.0)('Bob',45,85.5)('Cathy',37,68.0)('Doug',19, 61.5)]

Refervaluesthrough indexorname
Thehandythingwithstructured arrays isthatyoucannow refer to values either byindex or byname.
i. data['name']#byname

```
    array(['Alice','Bob','Cathy','Doug'],dtype='<U10')
```
ii. data[0]#byindex

```
    ('Alice',25,55.0)
```

UsingBooleanmasking
    Thisallowsto dosome moresophisticated operationssuch asfilteringonanyfields.
```
    data[data['age']<30]['name']
```

```
    array(['Alice','Doug'],dtype='<U10')
```

CreatingStructuredArrays

Dictionarymethod
```
np.dtype({'names':('name','age','weight'),
        'formats':('U10', 'i4',

'f8')})dtype([('name','<U10'),('age','<i4'),('weight','<f8')]
)
```

NumericaltypescanbespecifiedwithPythontypes
```
np.dtype({'names':('name','age','weight'),
            'formats':((np.str_, 10), int,

np.float32)})dtype([('name','<U10'),('age','<i8'),('weigh

t','<f4')])
```

Listof tuples
```
np.dtype([('name','S10'),('age','i4'),('weight','f8')])

dtype([('name','S10'),('age','<i4'),('weight','<f8')])
```
Specifythetypesalone
```
np.dtype('S10,i4,f8')

dtype([('f0','S10'),('f1','<i4'),('f2','<f8')])
```

DataManipulationwithPandas

Pandas is a newer package built on top of NumPy, and provides an efficient implementation of aDataFrame. DataFrames are essentially multidimensional arrays with attached row and column labels, and oftenwithheterogeneoustypes and/or missingdata.

Pandas, and in particular its Series and DataFrame objects, builds on the NumPy array structure andprovidesefficient accessto thesesortsof"data munging"tasks thatoccupymuchofadatascientist's time.

Herewewillfocusonthemechanicsofusing Series, DataFrame, andrelatedstructureseffectively.

IntroducingPandasObjects

PandasobjectscanbethoughtofasenhancedversionsofNumPystructuredarraysinwhichtherowsandcolumnsareidentified with labelsratherthan simple integer indices.

Pandasprovideahost ofusefultools,methods, andfunctionalityontop of the basicdatastructures.Threefundamental Pandas datastructures: theSeries,DataFrame,andIndex

ThePandas SeriesObject

APandas Seriesis aone-dimensional arrayof indexed data. Itcanbecreatedfrom alist or arrayas follows:
```
data = pd.Series([0.25, 0.5, 0.75,
1.0])data

00.25
10.50
20.75
31.00
        dtype:float64
```
- **Findingvalues**
  Thevalues aresimplyafamiliarNumPyarray
  ```
  data.values

  array([ 0.25,0.5,0.75,1.])
  ```
- **Findingindex**
  Theindex isanarray-likeobjectoftypepd.Index
  ```
  data.index
  ```
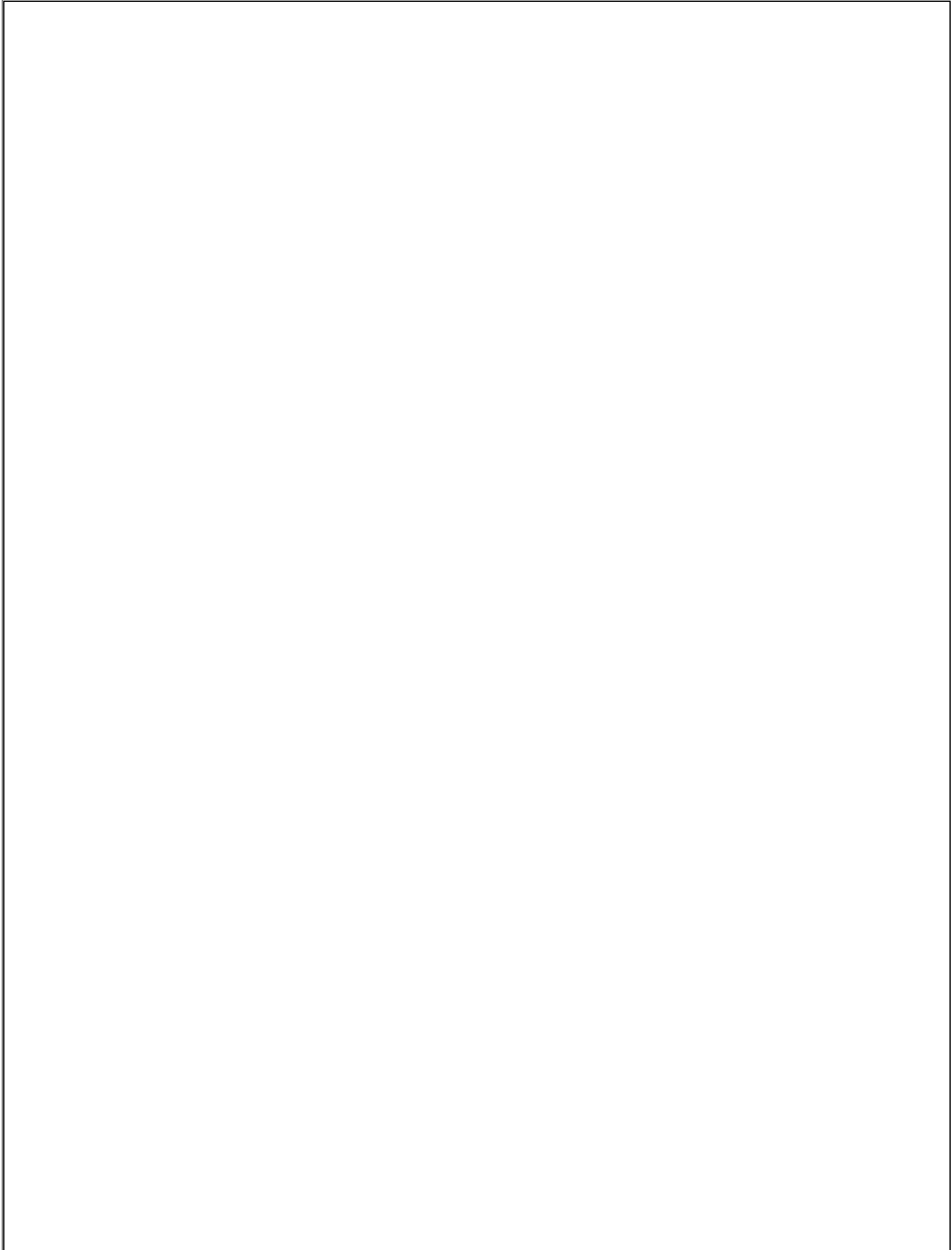
RangeIndex(start=0,stop=4,step=1)

- **Accessbyindex**

  LikewithaNumPyarray,data canbeaccessedbythe associatedindexvia thefamiliarPythonsquare-bracket notation

  data[1]

  0.5

  data[1:3]

  10.50
  20.75
      dtype:float64

SeriesasgeneralizedNumPyarray

theNumPyarrayhasanimplicitlydefinedintegerindexusedtoaccessthevalues,thePandasSerieshasanexplicitlydefined indexassociated withthe values.

ThisexplicitindexdefinitiongivestheSeriesobjectadditionalcapabilities.Forexample,theindexneednotbeaninteger, but can consist of values ofanydesiredtype.

Forexample, if wewish,wecan usestrings asan index.

Stringsasan index

    data=pd.Series([0.25,0.5, 0.75,1.0],
    index=['a', 'b', 'c',
    'd'])data

    a 0.25
    b0.50
    c 0.75
    d1.00
    dtype:float64

Noncontiguousornon sequentialindices.

    data=pd.Series([0.25,0.5, 0.75,1.0],
    index=[2, 5, 3,
    7])data

    20.25
    50.50
    30.75
    71.00
    dtype:float64

Seriesasspecializeddictionary

A dictionary is a structure that maps arbitrary keys to a set of arbitrary values, and a Series is a structure thatmapstyped keys to a setof typedvalues.

just as the type-specific compiled code behind a NumPy array makes it more efficient than a Python list forcertainoperations,thetypeinformationofaPandasSeriesmakesitmuchmoreefficientthanPythondictionariesforcer tainoperations.

WecanmaketheSeries-as-dictionaryanalogyevenmoreclearbyconstructingaSeriesobjectdirectlyfromaPythondictionary. Forexample

```
sub1={'sai':90,'ram':85,'kasim':92,'tamil':89}
mark=pd.Series(sub1)
mark
```

```
sai       90
ram       85
kasim     92
tamil     89
dtype:int64
```

Dictionary-styleitemaccess
```
Mark['ram']
85
```

Array-styleslicing
```
Mark[ 'sai':'kasim']
sai       90
ram       85
kasim     92
```

ConstructingSeriesobjects

- **ListorNumPyarray**
```
pd.Series([2, 4,6])

02
14
26
dtype:int64
```

  - **Repeatedtofillthespecifiedindex**
```
pd.Series(5,index=[100,200,300])

100 5
200 5
300 5
dtype:int64
```

  - **Datacanbeadictionary,in whichindexdefaultstothesorteddictionarykeys**
```
pd.Series({2:'a',1:'b',3:'c'})

1 b
2 a
3 c
dtype:object
```

  - **Theindexcanbeexplicitlysetifadifferentresultispreferred**

```
pd.Series({2:'a',1:'b',3:'c'},index=[3,2])
```

```
3c
2a
dtype:object
```

ThePandasDataFrameObject

ThefundamentalstructureinPandasistheDataFrame.TheDataFrame can bethoughtof eitherasageneralizationofaNumPyarray, or as aspecialization ofaPythondictionary.

DataFrameasageneralizedNumPyarray

ADataFrameisananalogofatwo-dimensionalarraywithbothflexiblerowindicesandflexiblecolumnnames. Just as you might think of a two-dimensional array as an ordered sequence of aligned one-dimensionalcolumns,you can think of a DataFrame as a sequence of aligned Series objects. Here, by "aligned" we meanthat theysharethe same index.

Todemonstratethis,let'sfirstconstructanewSerieslistingthe marksofsubject2.

```
sub2={'sai':91,'ram':95,'kasim':89,'tamil':90}
```

Wecan useadictionaryto construct a single two-dimensional object containingthis information.

```
result=pd.DataFrame({'DS':sub1,'FDS':sub2})r
esult
```

	DS	FDS
sai	90	91
ram	85	95
kasim	92	89
tamil	89	90

DataFramehasanindexattribute

LiketheSeries object,theDataFramehasanindexattributethatgives accesstothe indexlabels
result.index

```
Index(['sai','ram','kasim','tamil'],dtype='object')
```

DataFramehasacolumnsattribute.

TheDataFramehasa columnsattribute,whichisan Indexobjectholdingthecolumnlabels.
result.columns

```
Index(['DS','FDS'],dtype='object')
```

DataFrameasspecializeddictionary

WecanalsothinkofaDataFrameasaspecializationofadictionary.Whereadictionarymapsakeytoavalue,aDataFramem aps a column name to aSeries ofcolumn data.
result['DS']

```
sai90
ram85
kasim92
```

```
tamil89
Name:DS,dtype:int64
```

Note

Inatwo-dimensionalNumPyarray,data[0]willreturnthefirstrow.ForaDataFrame,data['col0']willreturnthe   first column. Because of this, it is probably better to think about DataFrames as generalized dictionariesratherthangeneralized arrays, thoughboth ways of lookingatthesituationcan beuseful.

ConstructingDataFrameobjects

APandasDataFramecanbeconstructedinavarietyofways.Herewe'llgiveseveralexamples.

- **FromasingleSeriesobject.**
- **Fromalistofdicts.**
- **FromadictionaryofSeriesobjects.**
- **Fromatwo-dimensionalNumPyarray.**
- **FromaNumPystructuredarray.**

FromasingleSeriesobject.

ADataFrameisacollectionofSeriesobjects,andasinglecolumnDataFramecanbeconstructedfromasingleSeries.

```
sub1=pd.Series({'sai':90,'ram':85,'kasim':92,'tamil':89})
pd.DataFrame(sub1,columns=['DS'])


            DS
sai         90
ram         85
kasim       92
tamil       89
```


Fromalistofdicts.

AnylistofdictionariescanbemadeintoaDataFrame.We'lluseasimplelistcomprehensiontocreatesomedata

```
data = [{'a': i, 'b': 2 *
i}for i in
range(3)]pd.DataFram
e(data)


 a
b000
112
224
```

EvenifsomekeysinthedictionaryaremissingˌPandaswillfill theminwithNaN(i.e.ˌ"notanumber")values.

```
pd.DataFrame([{'a': 1, 'b': 2}, {'b': 3, 'c':

 4}])abc
01.02NaN
1NaN 34.0
```


FromadictionaryofSeriesobjects.

Aswesaw before,aDataFramecanbeconstructedfroma dictionaryofSeries objects aswell.

```
pd.DataFrame({'DS':sub1,'FDS':sub2})
```

```
              DS     FDS
sai           90     91
ram           85     95
kasim   92    89
tamil   89    90
```

Fromatwo-dimensionalNumPyarray.

Givenatwo-dimensionalarrayofdata, wecancreateaDataFrame withanyspecifiedcolumnandindexnames.If omitted, an integerindexwill be used foreach.

```
pd.DataFrame(np.random.rand(3,
2),columns=['food', 'water'],
index=['a','b','c'])
```

```
  food    water
a0.8652570.213169
b0.4427590.108267
c0.0471100.905718
```

FromaNumPystructuredarray.

APandasDataFrameoperatesmuchlikeastructuredarray,andcanbe createddirectly.

```
A = np.zeros(3, dtype=[('A', 'i8'), ('B',
'f8')])A
```

```
array([(0,0.0),(0,0.0),(0,0.0)],
dtype=[('A','<i8'),('B','<f8')])
```

```
pd.DataFrame(A)
```

```
  AB
000.0
100.0
200.0
```

ThePandasIndex Object

We have seen here that both the Series and DataFrame objects contain an explicit index that lets you referenceand modify data. This Index object is an interesting structure in itself, and it can be thought of either as animmutablearrayoras anordered set.

```
ind = pd.Index([2, 3, 5, 7,
11])ind
```

```
Int64Index([2,3,5,7,11],dtype='int64')
```

- **Indexasimmutablearray**
  TheIndexobjectinmanywaysoperateslikeanarray.Forexample,wecanusestandardPythonin dexingnotation to retrievevalues orslices.

```
ind[1]
3
ind[::2]
Int64Index([2,5,11],dtype='int64')
```

- **Indexasorderedset**

Pandasobjectsaredesignedtofacilitateoperationssuchasjoinsacrossdatasets,whichdependonmanyaspectsof setarithmetic.

TheIndexobjectfollowsmanyoftheconventionsusedbyPython'sbuilt-insetdatastructure,sothatunions,intersections,differences,and other combinationscan becomputedin afamiliar way.

```
indA=pd.Index([1,3,5,7,9])
indB = pd.Index([2, 3, 5, 7,
11])indA&indB#intersection

Int64Index([3, 5, 7],

dtype='int64')indA|indB#union
Int64Index([1,2,3,5,7,9, 11],dtype='int64')

indA^ indB#symmetricdifference
Int64Index([1,2,9,11],dtype='int64')
```

DataIndexingandSelection

DataSelectioninSeries

ASeriesobjectactsinmanywayslikeaonedimensionalNumPyarray,andinmanywayslikeastandardPythondictionary.It willhelpustounderstandthepatterns ofdata indexingandselectioninthese arrays.

- Seriesasdictionary
- Seriesasone-dimensionalarray
- Indexers:loc,iloc, andix

- **Seriesasdictionary**

Likeadictionary,theSeriesobjectprovidesamappingfromacollectionofkeystoacollectionofvalues.

```
data=pd.Series([0.25,0.5, 0.75,1.0],
index=['a', 'b', 'c',
'd'])data

a 0.25
b0.50
c 0.75
d1.00
dtype:float64

data['b']
```

0.5

Examinethekeys/indicesandvalues

Wecanalsousedictionary-likePythonexpressionsand methodstoexaminethekeys/indicesandvalues

i. 'a'**in**data

True

ii. data.keys()

Index(['a','b','c','d'],dtype='object')

iii. list(data.items())

[('a',0.25),('b',0.5),('c',0.75),('d',1.0)]

Modifyingseriesobject

Seriesobjectscanevenbemodifiedwithadictionary-likesyntax.Justasyoucanextendadictionarybyassigningto a new key,youcan extend aSeries byassigningto a new indexvalue.

data['e'] =
1.25data

a 0.25
b0.50
c 0.75
d1.00
e1.25
dtype:float64

- **Seriesasone-dimensionalarray**

ASeriesbuildsonthisdictionary-likeinterfaceandprovidesarray-styleitemselectionviathesamebasicmechanismsas NumPyarrays—that is, slices, masking, andfancyindexing.

Slicingbyexplicitindex

data['a':'c']

a 0.25
b0.50
c 0.75
dtype:float64

Slicingbyimplicitintegerindex

data[0:2]

a 0.25
b0.50
dtype:float64

Masking

data[(data >0.3)&(data<0.8)]

b0.50
c0.75

dtype:float64

Fancyindexing

data[['a','e']]

a 0.25
e1.25
dtype:float64

- **Indexers:loc,iloc,andix**

Pandasprovidessomespecialindexerattributesthatexplicitlyexposecertainindexingschemes.Thesearenotfunctional methods, but attributes thatexposeaparticularslicinginterfacetothe data in theSeries.

data = pd.Series(['a', 'b', 'c'], index=[1, 3, 5])data

1a
3b
5c
dtype:object

loc-the locattributeallows indexingandslicingthat always referencestheexplicitindex.

data.loc[1]
'a'

data.loc[1:3]
1a
3b
dtype:object

iloc-Theilocattributeallows indexingand slicingthatalways referencesthe implicitPython-style index.

data.iloc[1]
'b'

data.iloc[1:3]
3b
5c
dtype:object

ix-ixisa hybridofthetwo,andfor Seriesobjectsis equivalent tostandard [ ]-basedindexing.

DataSelectioninDataFrame

- **DataFrameasadictionary**
- **DataFrameastwo-dimensionalarray**
- **Additionalindexingconventions**

DataFrameasadictionary

Thefirst analogywewill consider is the DataFrameas a dictionaryof related Seriesobjects.

TheindividualSeriesthatmakeupthecolumnsoftheDataFramecanbeaccessedviadictionary-styleindexingofthecolumn name.

Dictionary-style indexing of the column
name.result=pd.DataFrame({'DS':sub1,'FDS':sub2})result['DS']

```
            DS
sai         90
ram         85
kasim       92
tamil       89
```

Attribute-styleaccesswithcolumnnamesthatarestrings
result.DS

```
            DS
sai         90
ram         85
kasim       92
tamil       89
```

Comparingattributestyleanddictionarystyleaccesses
result.DSisresult['DS']

```
            True
```

Modifythe object
Like with the Series objects this dictionary-style syntax can also be used to modify the object, in this case to addanew column:

result['TOTAL']=result['DS']+result['FDS']
result

	DS	FDS	TOTAL
sai	90	91	181
ram	85	95	180
kasim	92	89	181
tamil	89	90	179

DataFrameastwo-dimensionalarray
- **Transpose**

Wecantransposethe fullDataFrametoswaprows andcolumns.
result.T

	sai	ram	kasim	tamil
DS	90	85	92	89
FDS	91	95	89	90
TOTAL	181	180	181	179

Pandas again uses the loc, iloc, and ix indexers mentioned earlier. Using the iloc indexer, we can index theunderlying array as if it is a simple NumPy array (using the implicit Python-style index), but the DataFrameindex and column labels aremaintained intheresult

- **loc**
  result.loc[:'ram',:'FDS']

	DS	FDS
sai	90	91
ram	85	95

- **iloc**
  result.iloc[:2,:2]

	DS	FDS
sai	90	91
ram	85	95

- **ix**
  result.ix[:2,:'FDS']

	DS	FDS
sai	90	91
ram	85	95

- **MaskingandFancyindexing**

  Inthe locindexerwecancombinemaskingand fancyindexingas in thefollowing:
  result.loc[result.total>180,['DS','FDS']]

	DS	FDS
sai	90	91
kasim92	89	

- **Modifyingvalues**

  Indexingconventionsmayalsobeusedtosetormodifyvalues;thisisdoneinthestandardwaythatyoumight be accustomed to from workingwith NumPy.

  result.iloc[1,1]=70

	DS	FDS	TOTAL
sai	90	91	181
ram	85	**70**	180
kasim	92	89	181
tamil	89	90	179

Additional indexing conventionsSlicing rowwise

result['sai':'kasim']

	DS	FDS	TOTAL
sai	90	91	181
ram	85	**70**	180
kasim 92	89	181	

Suchslices can also referto rows bynumber rather than byindex:
```
result[1:3]
```

	DS	FDS	TOTAL
ram	85	**70**	180
kasim92	89	181	

Maskingrowwise
```
result[result.total>180]
```

	DS	FDS	TOTAL
sai	90	91	181
kasim92	89	181	

OperatingonDatainPandas

Pandas inherits much of this functionality from NumPy, and the ufuncs.   So Pandas having the ability toperform quick element-wise operations, both with basic arithmetic (addition, subtraction, multiplication, etc.)andwithmoresophisticatedoperations(trigonometricfunctions, exponentialandlogarithmicfunctions, etc.).

For unary operations like negation and trigonometric functions, these ufuncs will preserve index and columnlabelsin the output.

For binary operations such as addition and multiplication, Pandas will automatically align indices when passingtheobjects to the ufunc.

Hereweare goingto see howtheuniversal functionsareworkinginseries and DataFramesby

- **Indexpreservation**
- **Indexalignment**

IndexPreservation

Pandas is designed to work with NumPy, any NumPy ufunc will work on Pandas Series and DataFrame objects.We can use all arithmetic and special universal functions as in NumPy on pandas. In outputs the index willpreserved(maintained) as shown below.

Forseries
```
x=pd.Series([1,2,3,4])
x

01
12
23
34
dtype:int64
```

ForDataFrame
```
df=pd.DataFrame(np.random.randint(0,10,(3,4)),
                           columns=['a','b','c','d'])
```

30

```
df
```

```
        a     b     c     d
0       1     4     1     4
1       8     4     0     4
2       7     7     7     2
```

```
Foruniversalfunction.(hereweuseexponentasexample)
```

Ufuncsforseries

```
np.exp(ser)
```

```
08103.083928
154.598150
2403.428793
320.085537
dtype:float64
```

UfuncsforDataFrame

```
np.exp(df)
```

	a	b	c	d
0	2.718282	54.598150	2.718282	54.598150
1	2980.957987	54.598150	1.000000	54.598150
2	1096.633158	1096.633158	1096.633158	7.389056
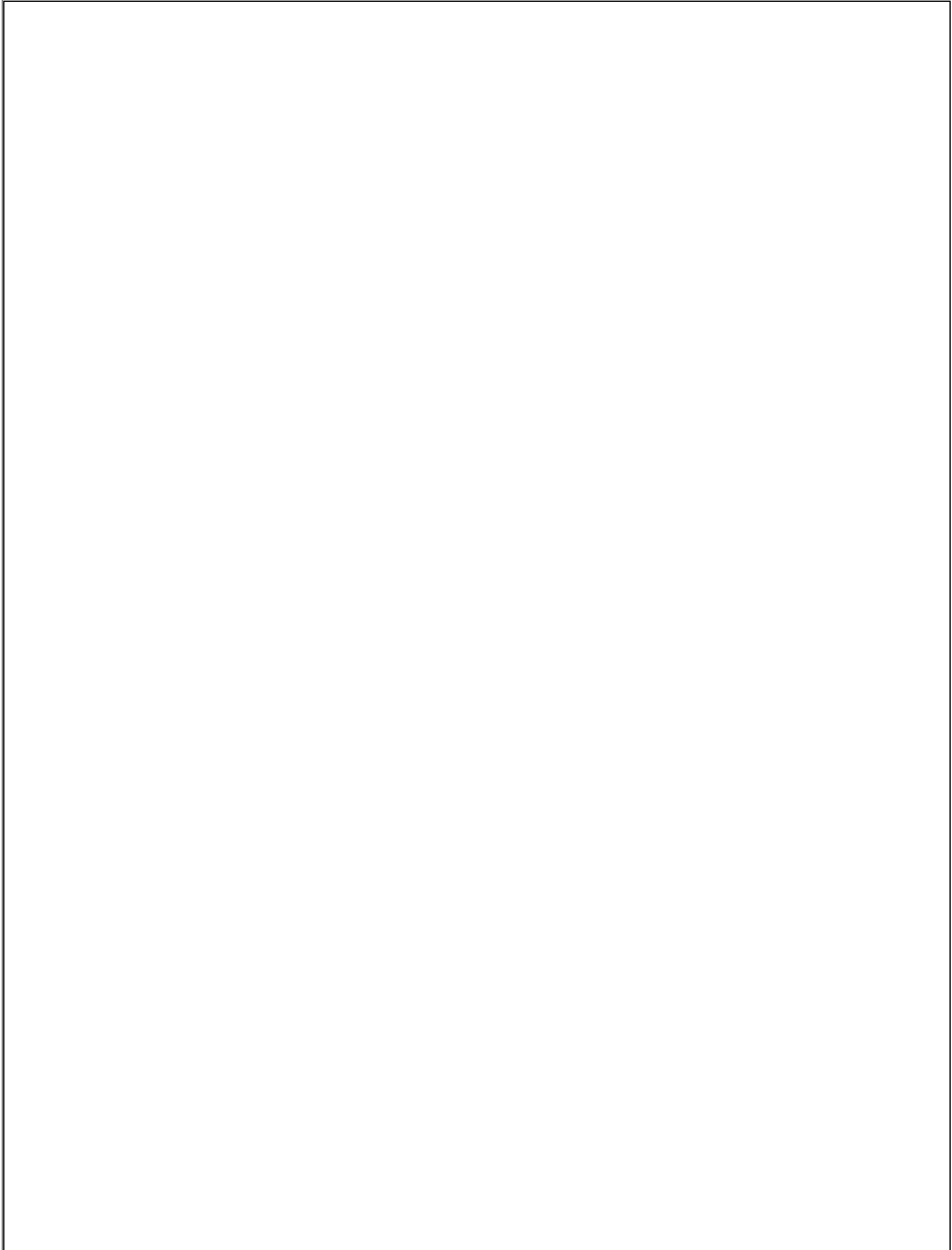
IndexAlignment

Pandaswillalignindicesintheprocessofperformingtheoperation.Thisisveryconvenientwhenyouareworkingwithincomplete data, aswe'll.

IndexalignmentinSeries

supposewearecombiningtwodifferentdata sources,thentheindexwillalignedaccordingly.

```
x=pd.Series([2,4,6],index=[1,3,5])
y=pd.Series([1,3,5,7],index=[1,2,3,4])
x+y
```

```
13.0
2
NaN39
.0
4 NaN
5 NaN
dtype:float64
```

The resulting array contains the union of indices of the two input arrays, which we could determine usingstandardPython set arithmetic on theseindices.

Any item for which one or the other does not have an entry is marked with NaN, or "Not a Number," which ishowPandas marks as missingdata.

Fillvalueinmissingdata(fill_value)

IfusingNaNvaluesisnotthedesiredbehavior,wecanmodifythefillvalueusingappropriateobjectmethodsinplaceof theoperators.

```
x.add(y,fill_value=0)
```

```
13.0
23.0
39.0
47.0
56.0
dtype:float64
```

IndexalignmentinDataFrame

A similar type of alignment takes place for both columns and indices when you are performing operations onDataFrames.

```
A=pd.DataFrame(rng.randint(0,20,(2,2)),columns=list('AB'))A
```

```
 AB0
111
151
```

```
B = pd.DataFrame(rng.randint(0, 10, (3,
3)),columns=list('BAC'))
B
```

```
 B A
C0409
1580
2926
```

```
A+B
```

```
 A  B  C
01.015.0NaN
113.06.0
NaN2NaNNaNNa
N
```

Notice that indices are aligned correctly irrespective of their order in the two objects,and indices in the resultare sorted. As was the case with Series, we can use the associated object's arithmetic method and pass anydesiredfill_value tobeused inplaceofmissingentries.Herewe'll fill withthemeanof all valuesin A.

```
     fill =
     A.stack().mean()A.add
     (B,fill_value=fill)


      A  B  C
     0 1.015.0 13.5
     113.06.04.5
     2 6.513.5 10.5
```

Mapping between Python operators and Pandas
methods.Pythonoperator   Pandas method(s)
     +   add()
     -                  sub(),subtract()
     *   mul(),multiply()
     /   truediv(),div(),divide()
     //   floordiv()
     %   mod()
     **   pow()

OperationsbetweenData FrameandSeries

When you are performing operations between a DataFrame and a Series, the index and column alignment issimilarly maintained. Operations between a DataFrame and a Series are similar to operations between a two-dimensionaland one-dimensionalNumPyarray.

```
     A = rng.randint(10, size=(3,
     4))A
     array([[3,8, 2, 4],
     [2, 6,4, 8],
     [6, 1,3,8]])


     A - A[0]
     array([[ 0,0,0,0],
     [-1,-2,2,4],
     [3,-7,1,4]])
```

HandlingMissing Data

A number of schemes have been developed to indicate the presence of missing data in a table or DataFrame.Generally, they revolve around one of two strategies: using a **mask** that globally indicates missing values, orchoosinga **sentinel value** that indicates amissing entry.

In the masking approach, the mask might be an entirely separate Boolean array, or it may involve appropriationofonebit in thedata representation to locallyindicate thenull status of avalue.

In the sentinel approach, the sentinel value could be some data-specific convention, such as indicating a missinginteger value with –9999 or some rare bit pattern, or it could be a more global convention, such as indicating amissing floating-point value with NaN (Not a Number), a special value which is part of the IEEE floating-pointspecification.

MissingDatainPandas
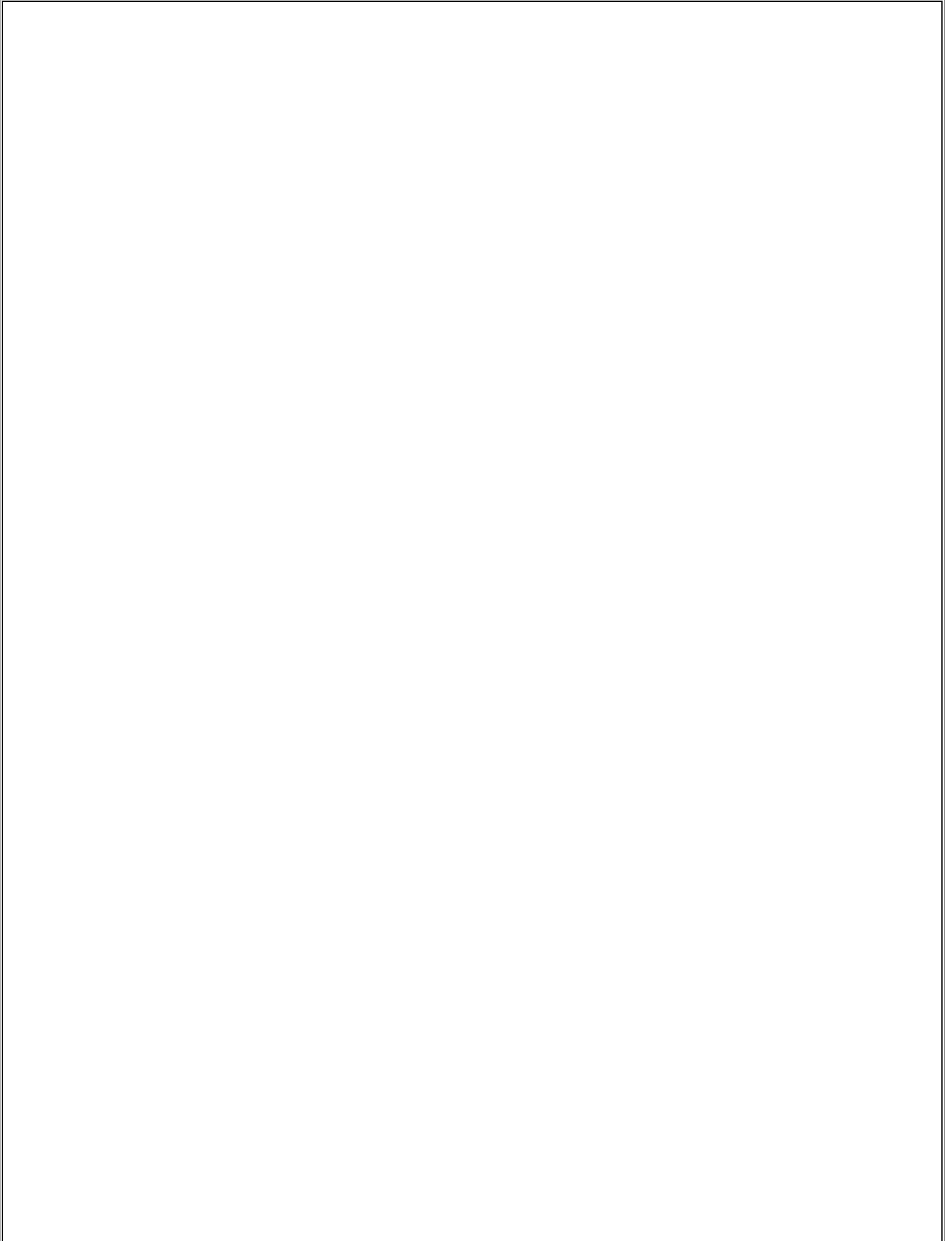
The way in which Pandas handles missing values is constrained by its NumPy package, which does not have abuilt-innotion of NAvalues for nonfloating-point datatypes.

NumPysupportsfourteenbasicintegertypesonceyouaccountforavailableprecisions,signedness,andendianness   of the encoding. Reserving a specific bit pattern in all available NumPy types would lead to anunwieldy amount of overhead in special-casing various operations for various types, likely even requiring a newfork of theNumPypackage.

Pandas chose to use sentinels for missing data, and further chose to use two already-existing Python null values:the special floatingpoint NaN value, and the Python None object. This choice has some side effects, as we willsee,but in practiceendsup being agoodcompromisein most cases ofinterest.

None:Pythonicmissingdata
The first sentinel value used by Pandas is None, a Python singleton object that is often used for missing data inPython code. Because None is a Python object, it cannot be used in any arbitrary NumPy/Pandas array, but onlyinarrays with datatype'object' (i.e., arrays of Pythonobjects)

This dtype=object means that the best common type representation NumPy could infer for the contents of thearrayis that theyarePython objects.

NaN:Missingnumericaldata
NaN is a special floating-point value recognized by all systems that use the standardIEEE floating-pointrepresentation.

    vals2 = np.array([1, np.nan, 3,
    4])vals2.dtype

    dtype('float64')
You should be aware that NaN is a bit like a data virus—it infects any other object it touches. Regardless of theoperation,the result of arithmetic with NaNwill beanotherNaN

    1 +

    np.nannan
    0 *np.nan

    Nan

NaNandNoneinPandas
NaNand Nonebothhave theirplace,and Pandas isbuilt to handlethe two ofthemnearlyinterchangeably.

    pd.Series([1, np.nan, 2,
    None])01.0
    1 NaN
    22.0
    3NaN
    dtype:float64
For types that don't have an available sentinel value, Pandas automatically type-casts when NA values arepresent. For example, if we set a value in an integer array to np.nan, it will automatically be upcast to a floating-pointtypeto accommodate theNA

```
x       =       pd.Series(range(2),
dtype=int)x
00
11
dtype:
int64x[0]   =
Nonex
0NaN
11.0
dtype:float64
```

Noticethatinadditionto castingtheinteger arraytofloatingpoint,Pandasautomaticallyconvertsthe None toaNaNvalue.

PandashandlingofNAsbytype

Typeclass	ConversionwhenstoringNAs	NAsentinelvalue
floating	Nochange	np.nan
object	Nochange	Noneornp.nan
integer	Casttofloat64	np.nan
boolean	Casttoobject	Noneornp.nan

Note: InPandas,stringdata isalwaysstored withanobjectdtype**.**

OperatingonNullValues

thereareseveralusefulmethodsfordetecting,removing, andreplacingnullvalues inPandasdatastructures.Theyare:

- isnull()-Generate aBooleanmask indicatingmissingvalues
- notnull()-Oppositeof isnull()
- dropna()-Returna filteredversion ofthedata
- fillna()-Returnacopyof thedata with missingvalues filled orimputed

Detectingnullvalues

Pandasdatastructureshavetwo usefulmethodsfordetectingnulldata:isnull()andnotnull().isnull()

```
data = pd.Series([1, np.nan, 'hello',
None])data.isnull()

0 False
1 True
2 False
3 Truedtyp
e:bool
```

notnull()

```
data.notnull()

0 True
1 False
2 True
```

35

```
3 Falsedtyp
e:bool
```

Droppingnullvalues

```
dropna()
     data.dropna()

     01
     2hellodtype:
     object

     Droppingnullvalues indataframe

     df = pd.DataFrame([[1, np.nan,
     2],[2, 3,5],
     [np.nan, 4,
     6]])Df

      012
     0 1.0NaN 2
     1 2.03.0 5
     2 NaN 4.06

     df.dropna()

     012
     1 2.03.0 5
```

Dropvaluesin columnorrow
Wecan dropNAvalues along adifferent axis;axis=1 dropsall columnscontaininganull value.

```
     df.dropna(axis='columns')

     02
     15
     26
```

Rowsorcolumnshavingallnullvalues
Youcan also specifyhow='all', whichwillonlydrop rows/columns that areallnull values.

```
     df[3] =
     np.nandf

     0123
     0 1.0NaN2NaN
     1 2.03.05 NaN
     2NaN 4.06NaN
```

```
df.dropna(axis='columns',how='all')
```

```
012
0 1.0NaN 2
1 2.03.0 5
2 NaN 4.06
```

Specificnoofnullvalues(thresh)

thethreshparameter letsyouspecifyaminimumnumberofnon-nullvaluesfortherow/column tobekept

```
df.dropna(axis='rows',thresh=3)
```

```
0123
1 2.03.05 NaN
```

Fillingnullvalues

Sometimes rather than dropping NA values, you'd rather replace them with a valid value. This value might be asingle number like zero, or it might be some sort of imputation or interpolation from the good values. You coulddo this in-place using the isnull() method as a mask, but because it is such a common operation Pandas providesthefillna() method, which returns acopyof thearraywith thenull valuesreplaced.

```
data = pd.Series([1, np.nan, 2, None, 3],
index=list('abcde'))data
```

```
a 1.0
bNaN
c 2.0
dNaN
```

Fillwithsinglevalue

Wecanfill NAentries withasingle value,such aszero

```
data.fillna(0)
```

```
a 1.0
b0.0
c 2.0
d0.0
e3.0
dtype:float64
```

Fillwith previous value

Wecan specifyaforward-fillto propagatethepreviousvalue forward

```
data.fillna(method='ffill')
```

```
a 1.0
b1.0
c 2.0
d2.0
e3.0
dtype:float64
```

Fillwith nextvalue

Wecanspecifyaback-fill to propagate thenext values backward.

```
data.fillna(method='bfill')


a 1.0
b2.0
c 2.0
d3.0
e3.0
dtype:float64
```

HierarchicalIndexing

Up to this point we've been focused primarily on one-dimensional and twodimensional data, stored in PandasSeries and DataFrame objects, respectively. Often it is useful to go beyond this and store higher-dimensionaldata—thatis, data indexed bymorethan oneor two keys.

PandasdoesprovidePanelandPanel4Dobjectsthatnativelyhandlethree-dimensionalandfour-dimensional,afarmorecommonpatterninpracticeistomakeuseofhierarchicalindexing(alsoknownasmulti-indexing)toincorporatemultiple indexlevels within a singleindex.

In this way, higher-dimensional data can be compactly represented within the familiar one-dimensional Seriesandtwo-dimensional DataFrame objects.

Here we'll explore the direct creation of MultiIndex objects; considerations around indexing, slicing, andcomputing statistics across multiply indexed data; anduseful routinesfor converting betweensimple andhierarchicallyindexed representationsofyour data.

AMultiplyIndexedSeries
PandasMultiIndex

Pandas provides a better way. Our tuple-based indexing is essentially a rudimentary multi-index, and the PandasMultiIndex type gives us the type of operations we wish to have. We can create a multi-index from the tuples asfollows

```
index=[('California',2000),('California',2010),
('NewYork', 2000),('NewYork', 2010),
('Texas',2000),('Texas',2010)]
populations=[33871648,37253956,
18976457,19378102,
20851820,25145561]
pop = pd.Series(populations,
index=index)pop

(California,2000)33871648
(California,2010)37253956
(NewYork,2000)18976457
(NewYork,2010)19378102
(Texas,2000)20851820
```

```
#creatingmultiindex
index =
pd.MultiIndex.from_tuples(index)index

MultiIndex(levels=[['California','NewYork','Texas'],[2000,2010]],
labels=[[0,0,1,1,2,2],[0, 1,0,1,0,1]])
```

Hierarchicalrepresentationof thedata

```
pop =
pop.reindex(index)pop

California200033871648
            201037253956
NewYork      200018976457
            201019378102
Texas   200020851820
            201025145561
dtype:int64
```

Herethefirsttwocolumnsof theSeriesrepresentationshowthemultipleindexvalues,whilethethirdcolumnshowsthe data.

Accessalldatawithsecondindex

```
pop[:,2010]

California37253956
NewYork19378102
Texas
25145561dtype
: int64
```

MultiIndexasextradimension

wecouldeasilyhavestoredthesamedatausingasimpleDataFramewithindexandcolumnlabels.The **unstack()**methodwill quicklyconvert amultiplyindexedSeriesintoaconventionallyindexed DataFrame.

```
pop_df=pop.unstack()
pop_df

            20002010
California3387164837253956
NewYork      1897645719378102
Texas   2085182025145561
```

The**stack()**method provides theoppposite operation.

```
pop_df.stack()

California200033871648
201037253956
NewYork200018976457
201019378102
Texas200020851820
```

```
201025145561
dtype:int64
```

Addanewcolumninmultidimensionaldataframe.

```python
pop_df=pd.DataFrame({'total':pop,
                     'under18':[9267089,9284094,
                                4687374,4318033,
                                5906301,6879014]})
pop_df
```

		total	under18
California	2000	33871648	9267089
	2010	37253956	9284094
NewYork	2000	18976457	4687374
	2010	19378102	4318033
Texas	2000	20851820	5906301
	2010	25145561	6879014

Universal functions

Alltheufuncsandother functionalityworkwithhierarchicalindices.

```python
f_u18 = pop_df['under18'] /
pop_df['total']f_u18.unstack()
```

	2000	2010
California	0.273594	0.249211
NewYork	0.247010	0.222831
Texas	0.283251	0.273568

MethodsofMultiIndex Creation

ToconstructamultiplyindexedSeriesorDataFrameistosimplypassalistoftwoormoreindexarraystotheconstructor.

```python
df = pd.DataFrame(np.random.rand(4,
2),index=[['a', 'a','b','b'],[1,2,1,2]],
columns=['data1',
'data2'])df
```

		data1	data2
a	1	0.554233	0.356072
	2	0.925244	0.219474
b	1	0.441759	0.610054
	2	0.171495	0.886688

ifyoupassadictionarywithappropriatetuplesaskeys,PandaswillautomaticallyrecognizethisanduseaMultiIndexbydefault.

```python
data={        ('California',2000):33871648,
              ('California',2010):37253956,
              ('Texas',2000):20851820,
              ('Texas',2010):25145561,
              ('NewYork',2000):18976457,
```

```
                                ('NewYork',2010):19378102}
        pd.Series(data)


        California        200033871648
                                201037253956
        NewYork            200018976457
                                201019378102
        Texas            200020851820
                                201025145561
        dtype:int64
```

ExplicitMultiIndexconstructors
Youcanconstruct theMultiIndexfrom asimplelist ofarrays,givingtheindexvalues withineachlevel.

```
        pd.MultiIndex.from_arrays([['a','a','b','b'],[1,2,1,2]])


        MultiIndex(levels=[['a','b'],[1,2]],
        labels=[[0,0, 1,1], [0,1,0, 1]])
```

Multiindexfromalistoftuples,

```
        pd.MultiIndex.from_tuples([('a',1),('a',2),('b',1),('b',2)])


        MultiIndex(levels=[['a','b'],[1,2]],
                    labels=[[0,0, 1,1], [0,1,0, 1]])
```

Multiindex fromCartesianproduct.

```
        pd.MultiIndex.from_product([['a','b'],[1,2]])


        MultiIndex(levels=[['a','b'],[1,2]],
                    labels=[[0,0,1,1], [0, 1,0, 1]])
```

MultiIndexlevelnames
ItisconvenienttonamethelevelsoftheMultiIndex.Youcanaccomplishthisbypassingthenamesargumenttoanyof theaboveMultiIndexconstructors, orbysettingthe namesattributeof theindexafterthe fact.

```
        pop.index.names = ['state',
        'year']pop


        state        year
        California        200033871648
                                201037253956
        NewYork            200018976457
                                201019378102
        Texas            200020851820
                                201025145561
        dtype:int64
```

MultiIndexforcolumns
InaDataFrame,therowsandcolumnsarecompletelysymmetric,andjustastherowscanhavemultiplelevelsofindices, the columns can havemultiple levels as well.

41

#hierarchicalindicesandcolumns
index = pd.MultiIndex.from_product([[2013, 2014], [1,
2]],names=['year','visit'])
columns = pd.MultiIndex.from_product([['Bob', 'Guido', 'Sue'], ['HR',
'Temp']],names=['subject', 'type'])

#mocksomedata
data=np.round(np.random.randn(4,6),1)
data[:, ::2]*=10
data+=37

#createtheDataFrame
      health_data = pd.DataFrame(data, index=index,
      columns=columns)health_data

      subject   Bob
            GuidoSuetype   HR
            Temp HR Temp HR Tempyear
      visit
      2013131.038.732.036.735.037.2
          2   44.037.7 50.035.029.036.7
      2014130.037.439.037.861.036.9
          2   47.037.8 48.037.351.036.5

IndexingandSlicingaMultiIndex
IndexingandslicingonaMultiIndexisdesignedtobeintuitive,andithelpsifyouthinkabouttheindicesasaddeddimensions.We'll firstlook at indexingmultiplyindexed Series,andthen multiplyindexed DataFrames.

Multiplyindexed Series
      Pop

      state   year
      California   200033871648
                    201037253956
      NewYork   200018976457
                    201019378102
      Texas   200020851820
                    201025145561
      dtype:int64

- Accesssingleelements
  Wecan access singleelements byindexingwithmultiple terms
        pop['California',2000]

        33871648
- Partialindexing
  TheMultiIndexalsosupports partialindexing,or indexingjustoneofthelevelsintheindex
        pop['California']

```
        year
        200033871648
        201037253956
        dtype:int64
```

- Partialslicing
  Partialslicingisavailableaswell,aslongasthe MultiIndexissorted.
  pop.loc['California':'NewYork']

```
state           year
California      200033871648
                   201037253956
NewYork            200018976457
                   201019378102
dtype:int64
```

- Sortedindices
  Withsortedindices,wecanperformpartialindexingonlowerlevelsbypassinganemptysliceinthefirstindex
  pop[:,2000]

```
        state
        California      33871648
        NewYork             18976457
        Texas          20851820
        dtype:int64
```

- OthertypesofindexingandselectionSe
  lectionbasedonBooleanmasks
  pop[pop >22000000]

```
state           year
California       200033871648
                    201037253956
Texas            201025145561
dtype:int64
```

Selectionbased on fancyindexing
      pop[['California','Texas']]

```
state           year
California      200033871648
                   201037253956
Texas          200020851820
                   201025145561
dtype:int64
```

RearrangingMulti-Indices

Wesawabriefexampleofthisinthestack()andunstack()methods,buttherearemanymorewaystofinelycontroltherearrangement of databetweenhierarchicalindicesandcolumns,and we'llexplorethemhere.

- **Sortedandunsortedindices**

We'll start bycreatingsome simple multiplyindexed data wherethe indices are *not lexographicallysorted*:

index = pd.MultiIndex.from_product([['a', 'c', 'b'], [1,
2]])data = pd.Series(np.random.rand(6),
index=index)data.index.names=['char','int']
data

```
      charint
      a      1      0.003001
             2      0.164974
      c      1      0.741650
```

Pandas provides a number of convenience routines to perform this type of sorting; examples are the sort_index()andsortlevel()methods oftheDataFrame. We'llusethe simplest,sort_index(), here:

```
      data =
      data.sort_index()data

      charint
      a      1      0.003001
             2      0.164974
      b      1      0.001693
             2      0.526226
      c      1      0.741650
             2      0.569264
      dtype:float64
```

Withtheindexsortedinthisway,partialslicingwillworkasexpected:

```
      data['a':'b']

      char int
      a      1      0.003001
             2      0.164974
      b      1      0.001693
             2      0.526226
      dtype:float64
```

- **Stackingandunstackingindices**

itispossibletoconvertadatasetfromastackedmulti-indextoasimpletwo-dimensionalrepresentation,optionallyspecifyingthelevel to use.

```
      pop.unstack(level=0)

      state California New York
      Texasyear
      2000 33871648 18976457 20851820
      2010 37253956 19378102 25145561
```

```
pop.unstack(level=1)
```

```
year           2000   2010
state
California      3387164837253956
NewYork               1897645719378102
Texas          2085182025145561
```

Theopposite ofunstack() is stack(),which herecan beusedto recover theoriginalseries:

```
pop.unstack().stack()
```

```
stateyear
California      200033871648
                201037253956
NewYork         200018976457
                201019378102
Texas           200020851820
                201025145561
dtype:int64
```

- **Indexsettingandresetting**

Another way to rearrange hierarchical data is to turn the index labels into columns; this can be accomplishedwith the reset_index method. Calling this on the populationdictionary will result in a DataFramewith a stateand year column holding the information that was formerly in the index. For clarity, we can optionally specifythenameofthe data for thecolumn representation.

```
pop_flat =
pop.reset_index(name='population')pop_flat
```

```
          state   yearpopulation
 0California      200033871648
 1California      201037253956
 2NewYork         200018976457
 3NewYork         201019378102
 4 Texas                 200020851820
 5 Texas                 201025145561
```

DataAggregationsonMulti-Indices

We'vepreviouslyseenthatPandashasbuilt-indataaggregationmethods,suchasmean(),sum(),andmax().For hierarchically indexed data, these can be passed a level parameter that controls which subset of the data theaggregateis computed on.

Forexample,let'sreturntoourhealthdata: (youcancreateyourowndataframe/series)

```
health_data
```

```
subject         Bob             GuidoSue
type    HR      Temp HR          TempHR      Temp
year visit
2013131.0 38.7         32.036.7 35.037.2
     2        44.037.7         50.035.0 29.036.7
```

2014130.0 37.4   39.037.8 61.036.9
        2   47.037.8   48.037.3 51.036.5

Calculatetheaverageasfollows

```
data_mean =
health_data.mean(level='year')data_mean
```

```
subject          Bob              GuidoSue
type    HRTemp          HR Temp          HR
Tempyear
2013    37.538.241.035.8532.036.95
2014    38.537.643.537.5556.036.70
```

Byfurther makinguseof the axiskeyword, wecantakethemeanamonglevels onthe columnsaswell:

```
data_mean.mean(axis=1,
level='type')type          HR          Temp
year
2013    36.83333337.000000
2014    46.00000037.283333
```

CombiningDatasets

ConcatandAppend

SimpleConcatenationwithpd.concat

Pandashasafunction,pd.concat(),which hasasimilarsyntax
tonp.concatenatebutcontainsanumberofoptionsthat we'll discuss momentarily
pd.concat()canbeusedforasimpleconcatenationofSeriesorDataFrame objects,justasnp.concatenate()canbeused
forsimple concatenations of arrays

```
ser1 =pd.Series(['A','B','C'],index=[1,2,3])
ser2 = pd.Series(['D', 'E', 'F'], index=[4, 5,
6])pd.concat([ser1, ser2])
```

```
1 A
2 B
3 C
4 D
5 E
6 F
dtype:object
```

Concatenationindataframe.

```
df1=make_df('AB',[1,2])
df2=make_df('AB',[3,4])
print(df1);print(df2);print(pd.concat([df1,df2]))
```

```
df3              df4              pd.concat([df3,df4],axis='col')
```

```
AB   CD    ABCD
    0 A0B00C0D0 0A0B0C0D0
    1 A1B11C1D1 1A1B1C1D1
```

Duplicateindices
One important difference between np.concatenate and pd.concat is that Pandas concatenation preserves indices,evenif theresult will haveduplicate indices! Considerthis simple example.

```
x=make_df('AB',[0,1])
        y= make_df('AB',[2,3])


        y.index=x.index#makeduplicateindices!
        print(x);print(y);print(pd.concat([x,y]))


        x              y              pd.concat([x,y])
          AB    AB     AB
        0 A0B00A2 B20        A0 B0
        1 A1B11A3 B31        A1 B1
        0 A2B2
        1 A3B3
```

Theappend()method
SeriesandDataFrameobjectshaveanappendmethodthatcanaccomplishthesamethingin fewerkeystrokes.Forexample, rather thancallingpd.concat([df1, df2]), you can simplycall df1.append(df2):

```
        print(df1); print(df2);
        print(df1.append(df2))df1      df2
                       df1.append(df2)
          AB    AB    AB
        1 A1B1 3A3B31A1B1
        2 A2B2 4A4B42A2B2
        3 A3B3
        4 A4B4
```

MergeandJoin
Oneessential featureoffered byPandas isits high-performance, in-memoryjoin and mergeoperations.

CategoriesofJoins
- One-to-onejoins
- Many-to-onejoins
- Many-to-manyjoins
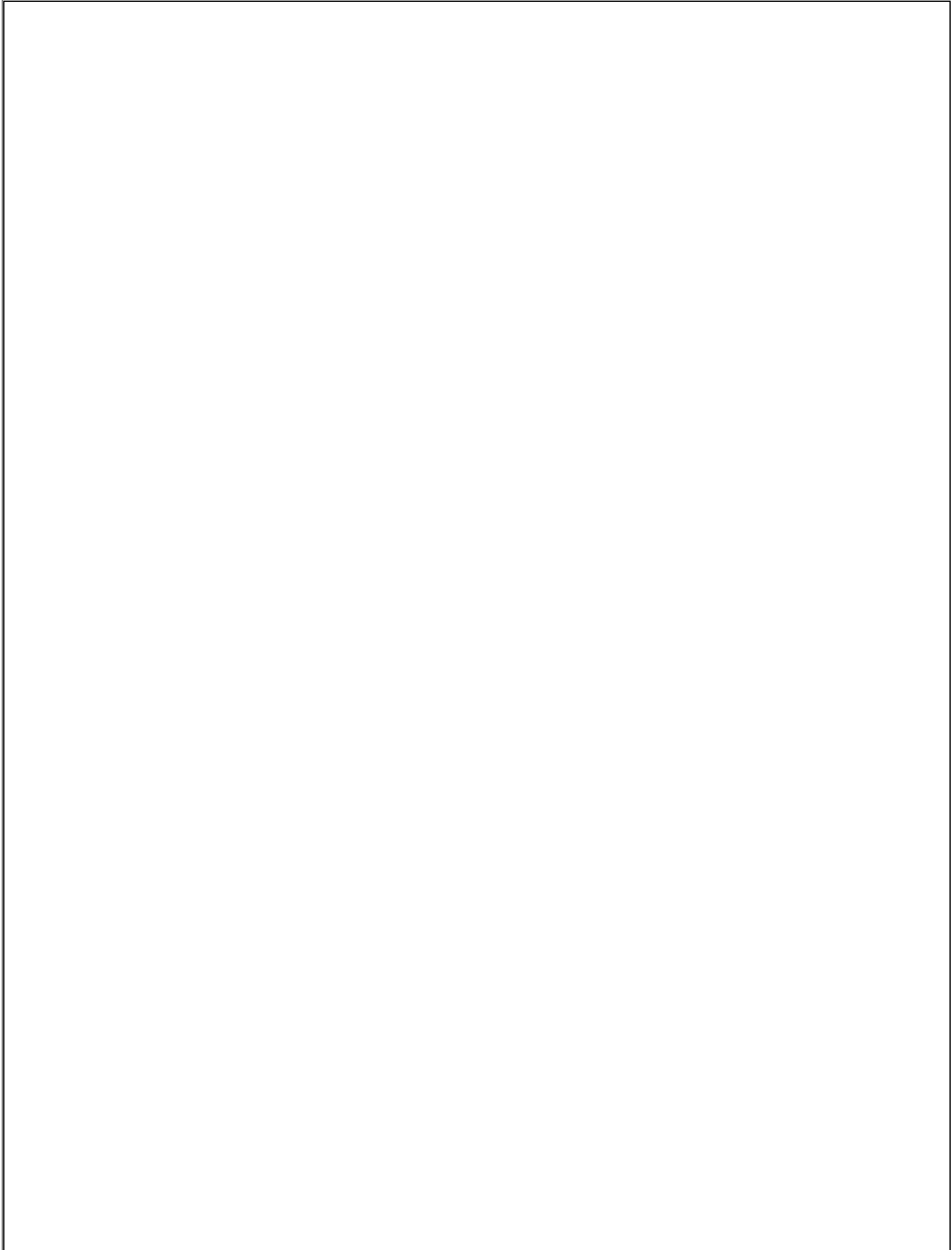
One– to– onejoins
 The simplest type of merge expression is the one-to-one join, which is in many ways very similar to thecolumn-wiseconcatenation.
```
df1=pd.DataFrame({'employee':['Bob','Jake','Lisa','Sue'],
'group':['Accounting','Engineering','Engineering','HR']})


df2 = pd.DataFrame({'employee': ['Lisa', 'Bob', 'Jake',
'Sue'],'hire_date':[2004,2008,2012,2014]})
```

print(df1);**print**(df2)

```
        df1                                        df2
            employee      group          employee      hire_date
    0       Bob           Accounting     0 Lisa2004
    1       Jake     Engineering    1Bob    2008
    2       Lisa     Engineering    2 Jake  2012
    3       Sue           HR                    3Sue2014
```

Tocombinethisinformationintoasingle DataFrame,wecanusethepd.merge()function

    df3 = pd.merge(df1,
    df2)df3

```
            employee      group          hire_date
    0       Bob           Accounting     2008
    1       Jake     Engineering    2012
    2       Lisa     Engineering    2004
    3       Sue           HR                    2014
```

Many-to-onejoins

Many-to-onejoinsarejoinsinwhichoneofthetwokeycolumnscontainsduplicateentries.Forthemany-to-onecase, theresultingDataFrame willpreservethoseduplicate entries as appropriate.

    df4=pd.DataFrame({'group':['Accounting','Engineering','HR'],
                        'supervisor':['Carly','Guido', 'Steve']})

    pd.merge(df3,df4)

```
            employee      group          hire_datesupervisor
    0       Bob           Accounting     2008   Carly
    1       Jake     Engineering    2012   Guido
    2       Lisa     Engineering    2004   Guido
    3       Sue           HR                    2014   Steve
```

The resulting DataFrame has an additional column with the "supervisor" information, where the information isrepeatedin one ormorelocations as required bytheinputs.

Many-to-manyjoins

Many-to-many joins are a bit confusing conceptually, but are nevertheless well defined. If the key column inboth the left and right array contains duplicates, then the result is a many-to-many merge. This will be perhapsmostclear with aconcreteexample.

    df5 = pd.DataFrame({'group': ['Accounting', 'Accounting', 'Engineering', 'Engineering', 'HR', 'HR'],
    'skills':['math','spreadsheets','coding','linux', 'spreadsheets','organization']})
    pd.merge(df1,df5)

```
            employee      group          skills
    0       Bob           Accounting     math
    1       Bob           Accounting     spreadsheets
    2       Jake     Engineering    coding
    3       Jake     Engineering    linux
    4       Lisa     Engineering    coding
```

5	Lisa	Engineering	linux
6	Sue	HR	spreadsheets
7	Sue	HR	organization

AggregationandGrouping

Computingaggregationslikesum(),mean(),median(), min(),andmax(), inwhich a single numbergives insightinto thenatureof apotentiallylargedataset.

SimpleAggregationinPandas

AswithaonedimensionalNumPyarray,foraPandasSeriesthe aggregatesreturnasinglevalue.

```
rng =
np.random.RandomState(42)ser
=pd.Series(rng.rand(5))
ser
```

```
00.374540
10.950714
20.731994
30.598658
40.156019
dtype:float64
```

Sum

```
ser.sum()2.81192549
17081569
```

Mean

```
ser.mean()0.56238509
834163142
```

ThesameoperationsalsoperformedinDataFrame
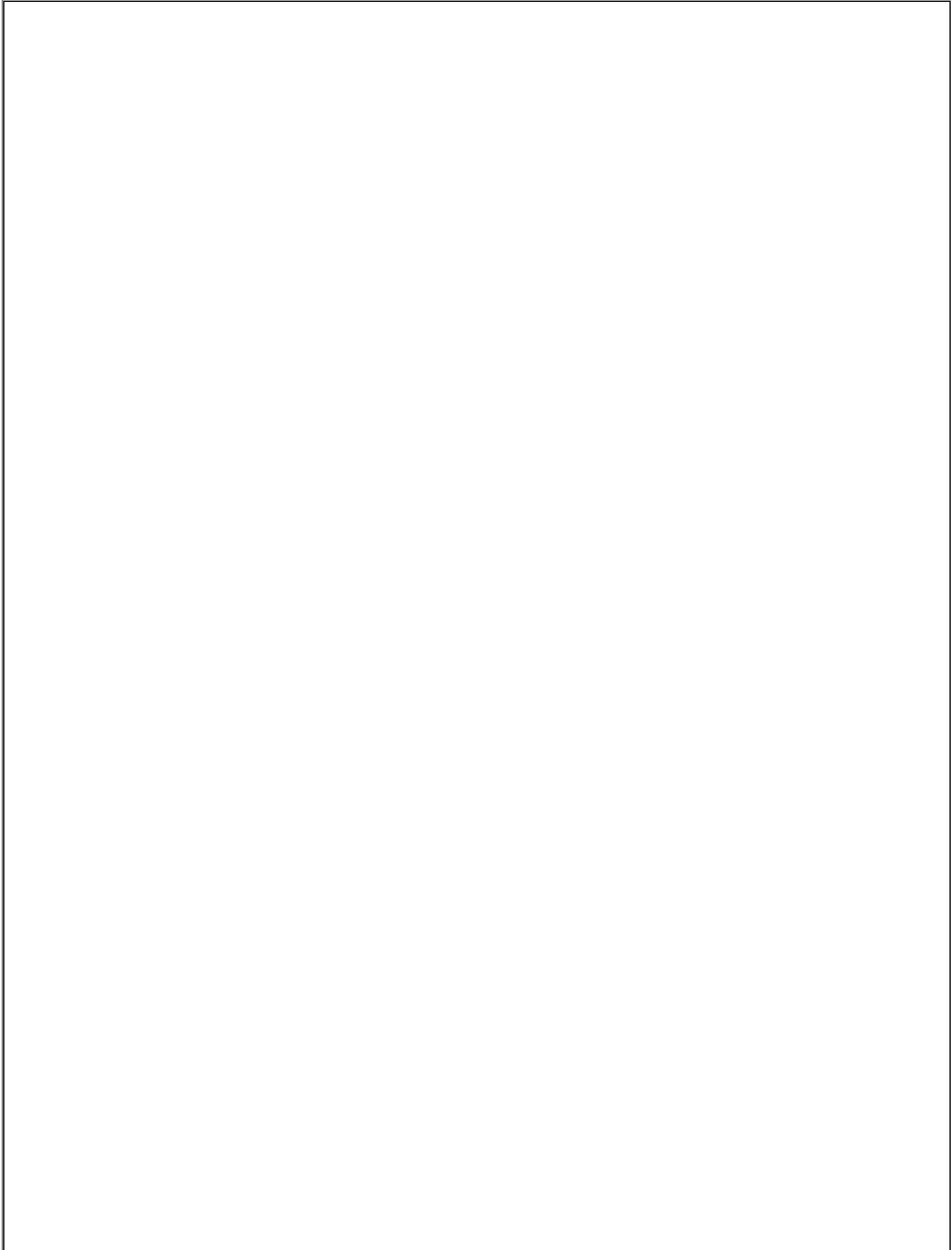
ListingofPandasaggregationmethodsA

ggregation	Description
count()	Totalnumberofitems
first(),last()	
	Firstandlastitem
mean(),median()	Meanandmedian
min(),max()	Minimum and maximum
std(),var()	Standarddeviationandvariance
mad()	Meanabsolutedeviation
prod()	Productofallitems
sum()	Sumofallitems

GroupBy:Split,Apply,Combine

Simple aggregations can give you a flavor of your dataset, but often we would prefer to aggregate conditionallyon some label or index: this is implemented in the socalled groupby operation. The name "group by" comesfrom a command in the SQL database language, but it is perhaps more illuminative to think of it in the termsfirst coined byHadleyWickham of Rstats fame: split, apply, combine.

• Thesplitstepinvolvesbreakingupandgroupinga DataFramedependingonthevalueofthespecifiedkey.
• Theapplystepinvolvescomputingsomefunction,usuallyanaggregate,transformation,orfiltering,withintheindividual groups.

• Thecombinestepmergestheresultsoftheseoperationsintoanoutputarray.
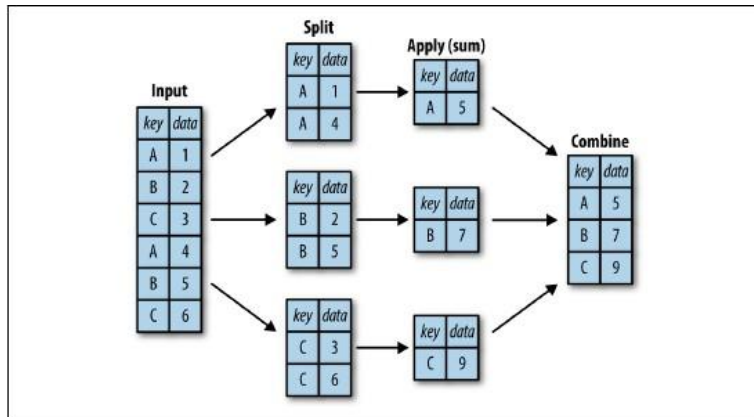
Example
df=pd.DataFrame({'key':['A','B','C','A','B','C'],
                          'data':range(6)},columns=['key','data'])
Df

```
         keydata
    0     A     0
    1     B     1
    2     C     2
    3     A     3
    4     B     4
    5     C     5
```

TheGroupByobject
The GroupByobjectisaveryflexibleabstraction.Themostimportantoperationsmade
availablebyaGroupByareaggregate, filter, transform, and apply.

Groupbysupports thebasic operations like.
- Columnindexing.
- Iterationovergroups.
- Dispatchmethods.
- Aggregate, filter,transform,apply

Columnindexing.
The GroupBy object supports column indexing in the same way as the DataFrame, and returns a modified
GroupByobject.For example
df=pd.read_csv('D:\iris.csv')df.groupby('va
riety')

```
<pandas.core.groupby.generic.DataFrameGroupByobjectat0x00000
23BAADE84C0>
```

df.groupby('variety)['petal.length'']

```
<pandas.core.groupby.generic.SeriesGroupByobjectat0x0000023BAADE8490>
```
50

df.groupby('variety')["petal.length"].sum()

```
     varietySetosa73.1
     Versicolor     213.0
     Virginica      277.6
     Name:petal.length,dtype:float64
```

Iterationovergroups.
TheGroupByobjectsupportsdirect iterationover thegroups,returningeachgroupasaSeriesorDataFrame. Thiscanbeusefulfordoingcertainthingsmanually,thoughitisoftenmuchfastertousethebuilt-inapplyfunctionality,whichwe willdiscussmomentarily.

Dispatchmethods.
Through some Python class magic, any method not explicitly implemented by the GroupBy object will be passed throughandcalledonthegroups,whethertheyareDataFrameorSeriesobjects.Forexample,youcanusethedescribe()methodofDat aFrames to performasetof aggregations thatdescribeeachgroupinthedata.

Example
df.groupby('variety')['petal.length'].describe().unstack()

```
     variety
count     Setosa     50.000000
          Versicolor     50.000000
          Virginica      50.000000
mean      Setosa         1.462000
          Versicolor     4.260000
          Virginica      5.552000
std       Setosa         0.173664
          Versicolor     0.469911
          Virginica      0.551895
min       Setosa         1.000000
          Versicolor     3.000000
          Virginica      4.500000
25%       Setosa         1.400000
          Versicolor     4.000000
          Virginica      5.100000
50%       Setosa         1.500000
          Versicolor     4.350000
          Virginica      5.550000
75%       Setosa         1.575000
          Versicolor     4.600000
          Virginica      5.875000
max       Setosa         1.900000
          Versicolor     5.100000
          Virginica      6.900000
dtype: float64
```

Aggregate,filter,transform,andapply
```
     rng=np.random.RandomState(0)
     df=pd.DataFrame({'key':['A','B','C','A','B','C'],
```

```
'data1':range(6),
'data2':rng.randint(0,10,6)},
columns = ['key', 'data1',
'data2'])df
```

```
        keydata1data2
0       A       0       5
1       B       1       0
2       C       2       3
3       A       3       3
4       B       4       7
5       C       5       9
```

Aggregation.

We're now familiar with GroupBy aggregations with sum(), median(), and the like, but the aggregate() methodallows for even more flexibility. It can take a string, a function, or a list thereof, and compute all the aggregatesatonce. Hereis a quick examplecombiningall these:

```
df.groupby('key').aggregate(['min', np.median,

        max])data1    data2

        minmedianmaxmin medianmax
key
A       0       1.5     3       3       4.0     5
B       1       2.5     4       0       3.5     7
C       2       3.5     5       3       6.0     9
```

Filtering.

Afilteringoperationallowsyoutodropdatabasedonthegroupproperties.Forexample,wemightwanttokeepallgroups inwhichthe standard deviationislarger thansomecriticalvalue.

Thefilter() functionshouldreturnaBooleanvalue specifyingwhetherthegroup passesthefiltering.

Transformation.

While aggregation must return a reduced version of the data, transformation can return some transformedversion of the full data to recombine. For such a transformation, the output is the same shape as the input. Acommon example is to centerthe databysubtractingthegroup-wisemean:

```
df.groupby('key').transform(lambdax:x-x.mean())
```

```
        data1   data2
0       -1.5    1.0
1       -1.5    -3.5
2       -1.5    -3.0
3       1.5             -1.0
4       1.5     3.5
5       1.5             3.0
```

Theapply()method.

The apply() method lets you apply an arbitrary function to thegroup results. The function should take aDataFrame,andreturneitheraPandasobject(e.g.,DataFrame,Series)orascalar;thecombineoperationwillbetailored to thetypeofoutput returned.

PivotTables

A pivot table is a similar operation that is commonly seen in spreadsheets and other programs that operate ontabular data. The pivot table takes simple column wisedata as input, and groups the entries into a two-dimensional table that provides a multidimensional summarization of the data. The difference between pivottables and GroupBy can sometimes cause confusion; it helps me to think of pivot tables asessentially amultidimensional version of GroupBy aggregation. That is, you split apply- combine, but both the split and thecombinehappenacross not a one-dimensional index, but across atwo-dimensionalgrid.

PivotTableCreation

```
import numpy as
npimportpandasasp
d
df=pd.read_csv('D:\diabetes.csv')df.pivot_table('preg',index='age',columns='Class'
).sample(10)
```

#herediabetes data sethas largenoofrows sowe usesample()

Class	tested_negative	tested_positive
age		
63	5.500000	NaN
28	3.440000	2.000000
61	7.000000	4.000000
69	5.000000	NaN
45	7.285714	7.375000
62	6.500000	1.000000
53	2.000000	6.250000
68	8.000000	NaN
23	1.516129	1.857143

Class	tested_negative	tested_positive
age		
52	13.000000	3.428571

UNITV
DATAVISUALIZATION

Importing Matplotlib – Line plots – Scatter plots – visualizing errors – density and contour plots – Histograms – legends – colors – subplots – text and annotation – customization – three dimensional plotting - Geographic DatawithBasemap-Visualization with Seaborn.

SimpleLinePlots

Thesimplestofallplotsisthevisualizationofasinglefunctiony=fx.Herewewilltakeafirstlookatcreatingasimpleplot of this type.

The*figure*(aninstanceoftheclassplt.Figure)canbethoughtofasasinglecontainerthatcontainsalltheobjectsrepresentingaxes, graphics, text, and labels.

The*axes*(aninstanceoftheclassplt.Axes)iswhatweseeabove:aboundingboxwithticksandlabels,whichwilleventuallycontain theplot elements that makeupour visualization.

LineColorsand Styles
- Thefirstadjustmentyoumight wishtomaketoaplotistocontrolthelinecolorsandstyles.
- Toadjustthecolor,youcanusethecolorkeyword,whichacceptsastringargumentrepresentingvirtuallyanyimaginable color. Thecolor can bespecified in avarietyofways
- If nocoloris specified,Matplotlibwillautomaticallycyclethroughasetofdefaultcolorsformultiplelines

  Differentformsof colorrepresentation.
  specifycolorbyname   -color='blue'
  shortcolorcode (rgbcmyk)   -color='g'
  Grayscalebetween0and1   -color='0.75'
  Hex code (RRGGBB from 00 to FF)-
  color='#FFDD44'RGBtuple,values 0and1   -
  color=(1.0,0.2,0.3)allHTMLcolor namessupported   -
  color='chartreuse'

- Wecan adjustthe line styleusingthelinestyle keyword.
  Differentlinestyles
    linestyle='solid'line
    style='dashed'lines
    tyle='dashdot'lines
    tyle='dotted'

  Shortassignment
    linestyle='-' #
    *solid*linestyle='--' #
    *dashed*linestyle='-
    .'#dashdot*linestyle=':'
    # dotted*

- linestyleandcolorcodescanbe combinedintoasinglenonkeyword argumenttotheplt.plot()function
    plt.plot(x, x + 0, '-g') *# solid*
    *green*plt.plot(x, x + 1, '--c') *# dashed*
    *cyan*plt.plot(x, x + 2, '-.k') *# dashdot*
    *black*plt.plot(x, x+3, ':r'); *# dottedred*

AxesLimits

- Themost basicwayto adjustaxislimits isto usetheplt.xlim()and

plt.ylim()methodsExample

       plt.xlim(10,0)

     plt.ylim(1.2,-1.2);

- Theplt.axis()methodallowsyoutosetthexandylimitswithasinglecall,bypassingalistthatspecifies[xmin,xmax,ymin,ymax]

      plt.axis([-1,11,-1.5,1.5]);

- Aspectratioequalis usedtorepresentoneunitin xisequal tooneunit iny.plt.axis('equal')

LabelingPlots

     Thelabelingof plots includestitles, axislabels,

     andsimplelegends.Title   -plt.title()

       Label-plt.xlabel()

         plt.ylabel()

      Legend-plt.legend()

Example programsLinecolor

```
importmatplotlib.pyplotasplti
mport numpyas np
fig=plt.figure()a
x=plt.axes()
x=np.linspace(0,10,1000)ax
.plot(x,np.sin(x));
plt.plot(x, np.sin(x - 0), color='blue') # specify color by
nameplt.plot(x, np.sin(x - 1), color='g') # short color code
(rgbcmyk)plt.plot(x,np.sin(x-
2),color='0.75')#Grayscalebetween0and1
plt.plot(x,np.sin(x-3),color='#FFDD44')#Hex code(RRGGBBfrom00
toFF)plt.plot(x, np.sin(x - 4), color=(1.0,0.2,0.3)) # RGB tuple, values 0 and
1plt.plot(x,np.sin(x-5),color='chartreuse');#all HTMLcolornames supported
```
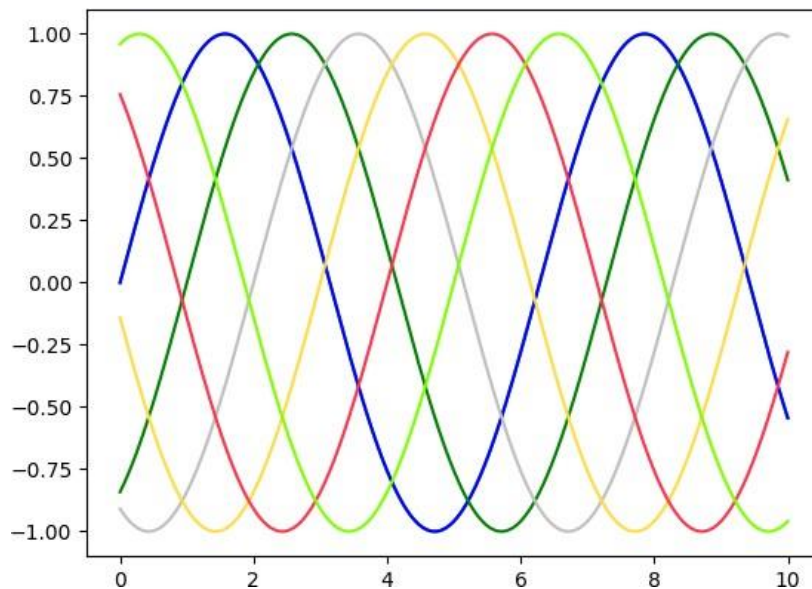

Linestyle

   importmatplotlib.pyplotasplt

```
importnumpyasnpfi
g=plt.figure()
ax=plt.axes()
x=np.linspace(0, 10,
1000)plt.plot(x, x + 0,
linestyle='solid')plt.plot(x, x + 1,
linestyle='dashed')plt.plot(x,x
+2,linestyle='dashdot')plt.plot(x,x
+3,linestyle='dotted');
#Forshort,youcanusethefollowingcodes:plt.pl
ot(x, x + 4, linestyle='-') # solidplt.plot(x, x +
5, linestyle='--') # dashedplt.plot(x, x + 6,
linestyle='-.') #
dashdotplt.plot(x,x+7,linestyle=':'); #dotted
```
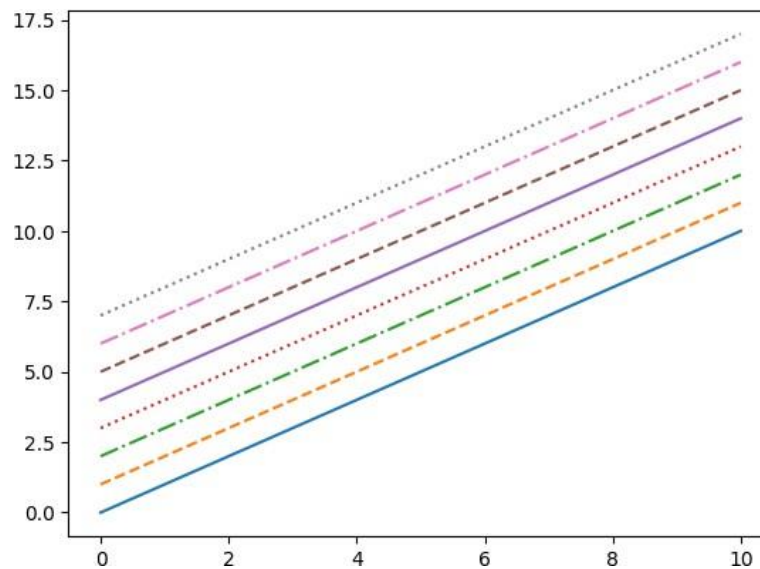

Axislimitwithlabelandlegend

```
import matplotlib.pyplot as
pltimport numpyas np
fig=plt.figure()a
x=plt.axes()
x=np.linspace(0,10, 1000)
plt.xlim(-1,11)
plt.ylim(-1.5,1.5);
plt.plot(x,np.sin(x),'-g',label='sin(x)')
plt.plot(x,np.cos(x),':b',label='cos(x)')pl
t.title("ASineCurve")
plt.xlabel("x")plt.yla
bel("sin(x)");plt.lege
nd();
```
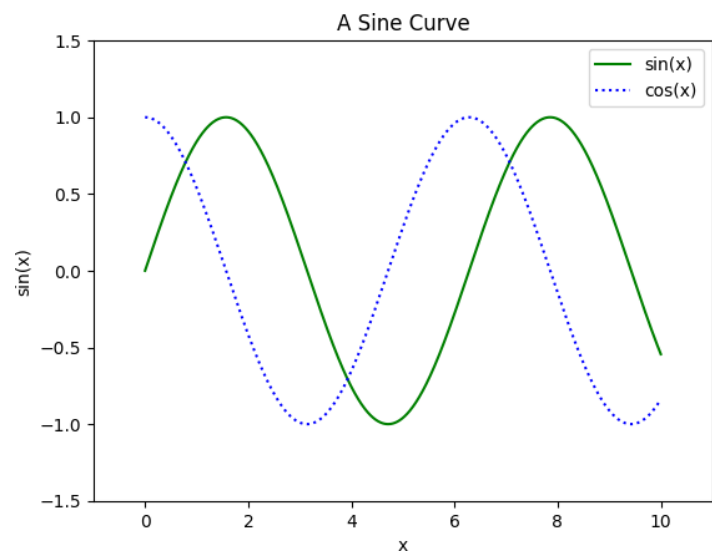

SimpleScatterPlots

Anothercommonlyusedplottypeisthesimplescatterplot,aclosecousinofthelineplot.Insteadofpointsbeingjoined bylinesegments,herethe points arerepresented individuallywith adot, circle, orother shape.

    Syntax

        plt.plot(x,y,'typeofsymbol ',color);

    Example

        plt.plot(x,y,'o',color='black');

- The third argument in the function call is a character that represents the type of symbol used for the plotting.Just as you can specify options such as '-' and '--' to control the line style, the marker style has its own set ofshort stringcodes.

Example

- Varioussymbolsusedto specify['o', '.', ',','x', '+', 'v', '^', '<','>','s', 'd']

- Shorthandassignmentof line,symbolandcolor alsoallowed.

  plt.plot(x,y,'-ok');

- Additionalargumentsinplt.plot()
  Wecanspecifysomeotherparametersrelatedwithscatterplotwhichmakesitmoreattractive.Theyarecolor,marker size, linewidth, markerfacecolor, markeredge color, markeredgewidth,etc

Example

    plt.plot(x, y, '-p',
    color='gray',markersize=15,
    linewidth=4,markerfacecolor
    ='white',markeredgecolor='g
    ray',markeredgewidth=2)plt.
    ylim(-1.2,1.2);

ScatterPlotswithplt.scatter

- A second, more powerful method of creating scatter plots is the plt.scatter function, which can be used verysimilarlyto theplt.plot function
  plt.scatter(x,y,marker='o');
- The primary difference of plt.scatter from plt.plot is that it can be used to create scatter plots where theproperties of each individual point (size, face color, edge color, etc.) can be individually controlled ormappedto data.
- Notice that the color argument is automatically mapped to a color scale (shown here by the colorbar()command),and the sizeargument isgiven inpixels.
- Cmap–colormapusedin scatterplotgivesdifferentcolorcombinations.

PerceptuallyUniformSequential
['viridis','plasma','inferno','magma']
Sequential
['Greys','Purples','Blues','Greens','Oranges','Reds','YlOrBr','YlOrRd',
'OrRd','PuRd','RdPu','BuPu','GnBu','PuBu','YlGnBu','PuBuGn','BuGn','YlGn'
]
Sequential(2)
['binary','gist_yarg','gist_gray','gray','bone','pink','spring','summer',
'autumn','winter','cool','Wistia','hot','afmhot','gist heat','copper']

4

```
Diverging
['PiYG','PRGn','BrBG','PuOr','RdGy','RdBu','RdYlBu','RdYlGn','Spectral','coolw
arm','bwr','seismic']
Qualitative
['Pastel1','Pastel2','Paired','Accent','Dark2','Set1','Set2','Set3','tab1
0','tab20','tab20b','tab20c']
Miscellaneous
['flag','prism','ocean','gist_earth','terrain','gist_stern','gnuplot',
'gnuplot2','CMRmap','cubehelix','brg','hsv','gist_rainbow','rainbow','jet
','nipy_spectral','gist_ncar']
```

Exampleprograms.

Simplescatterplot.

```
import numpyas np
importmatplotlib.pyplotaspltx
=np.linspace(0,10, 30)
y=np.sin(x)
plt.plot(x,y,'o',color='black');
```
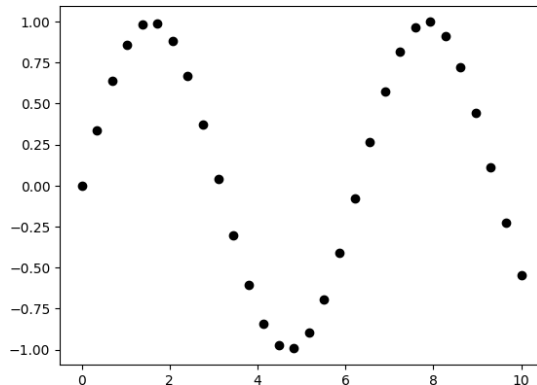
Scatter plot with edge color, face color, size,andwidthof marker.(Scatterplotwithline)

```
import numpyas np
importmatplotlib.pyplotaspltx
=np.linspace(0,10, 20)
y=np.sin(x)
plt.plot(x,y,'-
o',color='gray',markersize=15
,
linewidth=4,markerfacecolor
='yellow',markeredgecolor='r
ed',markeredgewidth=4)plt.yl
im(-1.5,1.5);
```
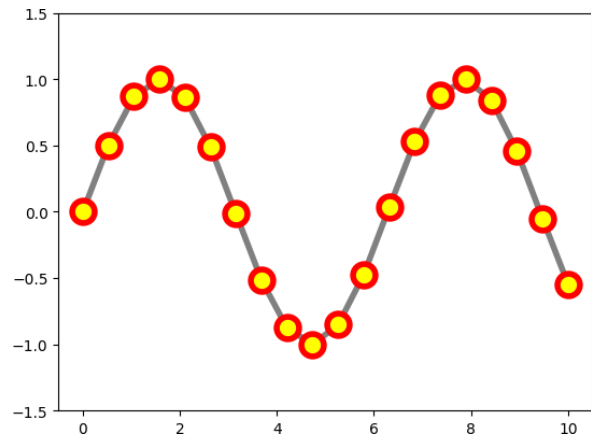
Scatterplotwithrandomcolors,sizeandtransparency
```
import numpyas np
import matplotlib.pyplot as
pltrng=np.random.RandomState(
0)x=rng.randn(100)
y =
rng.randn(100)colors=
rng.rand(100)
sizes=1000 *rng.rand(100)
plt.scatter(x,y,c=colors,s=sizes,alpha=0.3,
map='viridis')plt.colorbar()
```
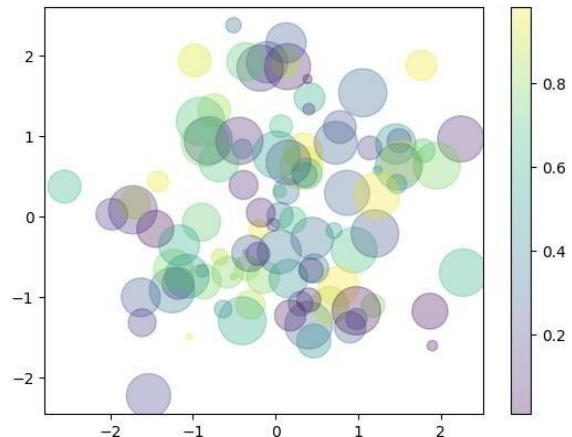
VisualizingErrors

For any scientific measurement, accurate accounting for errors is nearly as important, if not more important, thanaccurate reporting of the number itself. For example, imagine that I am using some astrophysical observations toestimatethe HubbleConstant, thelocal measurement ofthe expansion rateof theUniverse.

Invisualization   ofdata   and   results,   showingtheseerrorseffectivelycanmakeaplotconveymuch morecompleteinformation.

Typesoferrors
- BasicErrorbars
- ContinuousErrors

BasicErrorbars

Abasicerrorbar canbecreated withasingle Matplotlibfunctioncall.

```python
import matplotlib.pyplot as
pltplt.style.use('seaborn-
whitegrid')importnumpy asnp
x= np.linspace(0,10,50)
dy = 0.8
y = np.sin(x) + dy *
np.random.randn(50)plt.errorbar(x,y,
yerr=dy,fmt='.k');
```
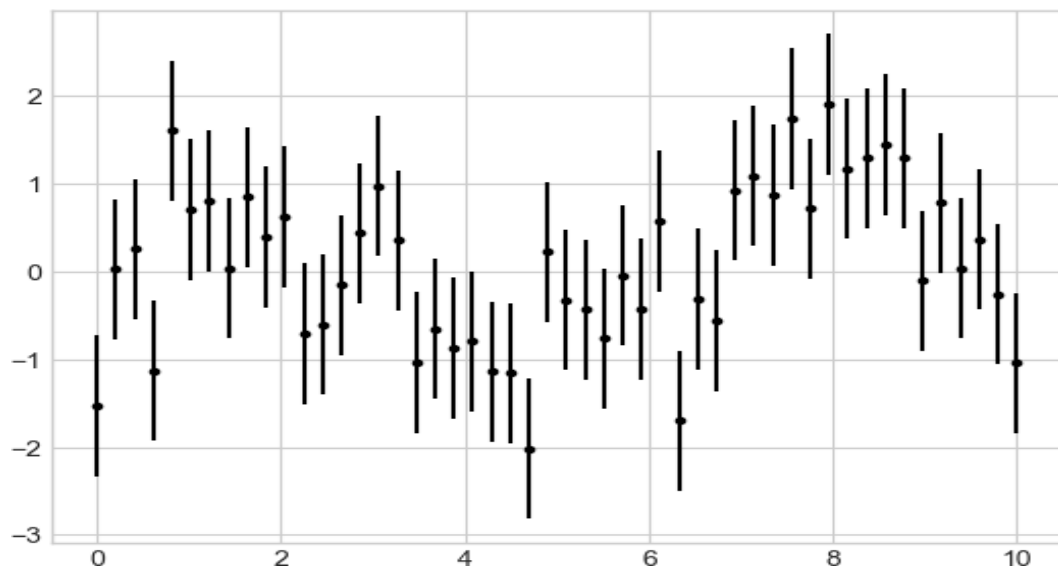

- Herethe fmt isaformatcodecontrollingthe appearanceof linesand points,and hasthe samesyntaxastheshorthandused in plt.plot()
- In addition to these basic options, the errorbar function has many options to fine tune the outputs. Usingtheseadditional options you caneasilycustomizethe aesthetics ofyour errorbar plot.

```python
plt.errorbar(x,y,yerr=dy,fmt='o',color='black',ecolor='lightgray',elinewidth=3,capsize=0);
```
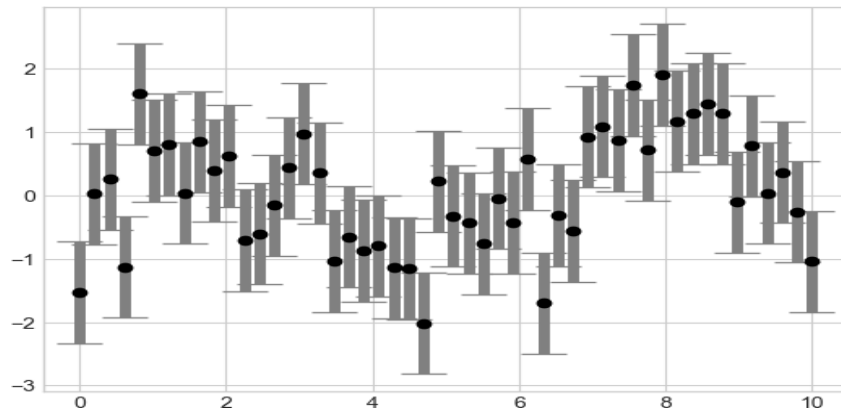
ContinuousErrors

- In some situations it is desirable to show errorbars on continuous quantities. Though Matplotlib does nothave abuilt-in convenience routinefor this typeof application, it'srelatively easy to combineprimitiveslikeplt.plotand plt.fill_betweenforauseful result.
- Here we'll perform a simple Gaussian process regression (GPR), using the Scikit-Learn API. This is amethodoffittingaveryflexiblenonparametricfunctiontodatawithacontinuousmeasureoftheuncertainty.

Densityand ContourPlots

To display three-dimensional data in two dimensions using contours or color-coded regions.TherearethreeMatplotlib functions that canbehelpful forthis task:

- plt.contourfor contourplots,
- plt.contourfforfilled contourplots, and
- plt.imshowforshowingimages.

VisualizingaThree-DimensionalFunction

Acontourplotcanbecreatedwiththeplt.contourfunction.

It

takesthreearguments:

- agrid ofxvalues,
- agrid ofyvalues,and
- agrid ofzvalues.

Thexandyvaluesrepresentpositionsontheplot,andthe

z

valueswillberepresented bythecontourlevels.

The way to prepare such data is to use the np.meshgridfunction,whichbuildstwo-dimensionalgridsfromone-dimensionalarrays:

Example

```
def f(x,y):
        return np.sin(x) ** 10 + np.cos(10 + y * x) *
np.cos(x)x=np.linspace(0,5,50)
y=np.linspace(0,5,40)X,
Y  =  np.meshgrid(x,
y)Z=f(X,Y)
plt.contour(X,Y,Z,colors='black');
```
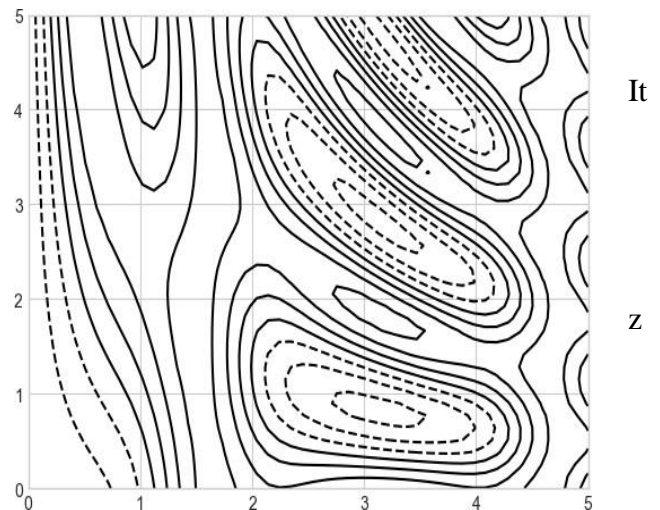
- Noticethatbydefaultwhenasinglecolorisused,negativevalues   arerepresentedby dashed lines, andpositivevalues bysolid lines.
- Alternatively,youcancolor-codethelinesbyspecifyinga colormapwith thecmapargument.
- We'llalso specifythat wewantmorelines to bedrawn—20 equallyspacedintervals within thedata range.

```
plt.contour(X,Y,Z,20,cmap='RdGy');
```

- Onepotentialissuewiththisplotisthatitisabit"splotchy."That is,thecolorstepsarediscreteratherthancontinuous,which is not always what is desired.
- Youcouldremedythisbysettingthenumberofcontourstoaveryhighnumber,butthisresultsinaratherinefficientplot : Matplotlib must render anewpolygon for each step in thelevel.
- Abetterwaytohandlethisistousetheplt.imshow()function,whichinterpretsatwo-dimensionalgridofdataas an image.

Thereareafewpotentialgotchaswithimshow().
- plt.imshow() doesn't accept an xandy grid, soyou must manually specify theextent [xmin, xmax,ymin,ymax] of theimageon theplot.
- plt.imshow()bydefaultfollowsthestandardimagearraydefinitionwheretheoriginisintheupperleft,notinthelowerleft as inmost contour plots.Thismust bechangedwhenshowing gridded data.
- plt.imshow()willautomaticallyadjusttheaxisaspectratiotomatchtheinputdata;youcanchangethisbysetting,for example, plt.axis(aspect='image') tomakexandyunits match.

Finally,itcansometimesbeusefultocombinecontourplotsandimageplots.we'lluseapartiallytransparentbackgroundimage(withtransparencysetviathealphaparameter)andover-plotcontourswithlabelsonthecontoursthemselves (usingthe plt.clabel()function):

```
contours = plt.contour(X, Y, Z, 3,
colors='black')plt.clabel(contours, inline=True,
fontsize=8)plt.imshow(Z, extent=[0, 5, 0, 5],
origin='lower',cmap='RdGy', alpha=0.5)
plt.colorbar();
```
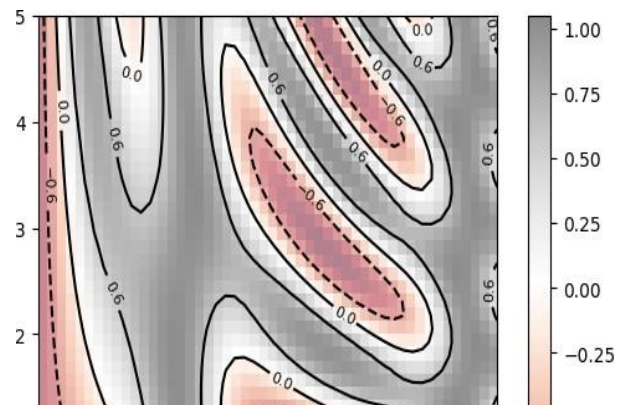
ExampleProgram

```
import numpyasnp
importmatplotlib.pyplotaspltd
eff(x, y):
returnnp.sin(x)**10+np.cos(10 +y*
x)*np.cos(x)
x= np.linspace(0,5,50)
y = np.linspace(0, 5,
40)X,       Y       =
np.meshgrid(x,     y)Z
=f(X,Y)
plt.imshow(Z, extent=[0, 10, 0,
10],origin='lower',
cmap='RdGy')plt.colorbar()
```
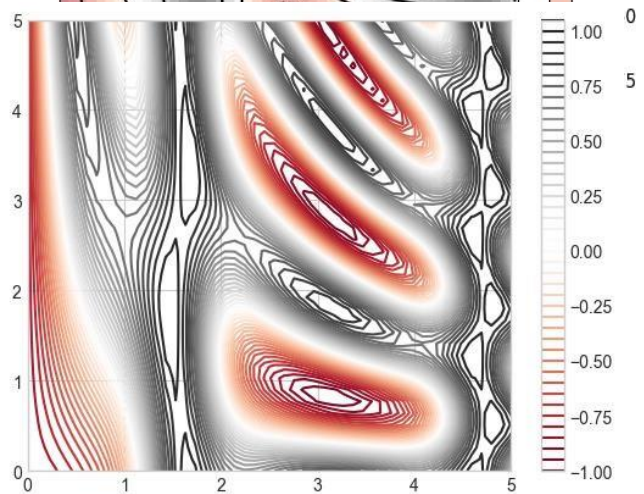
Histograms

- Histogramisthesimpleplottorepresentthelargedataset.Ahistogramisagraphshowingfrequencydistributions. It isagraph showingthe number ofobservations within eachgiven interval.

Parameters
- plt.hist()isusedtoplothistogram.Thehist()functionwilluseanarrayofnumberstocreateahistogram,thearrayis sent into thefunction as an argument.

- **bins** - A histogram displays numerical data by grouping data into "bins" of equal width. Each bin is plottedas a bar whose height corresponds to how many data points are in that bin. Bins are also sometimes called"intervals","classes", or"buckets".
- **normed** - Histogram normalization is a technique to distribute the frequencies of the histogram over a widerrangethan thecurrent range.
- **x** -(n,) array or sequence of (n,) arrays Input values, this takes either a single array or a sequence of arrayswhicharenotrequired tobeof thesame length.
- **histtype** - {'bar', 'barstacked', 'step', 'stepfilled'}, optionalThetypeof histogram todraw.

  - 'bar'isa traditionalbar-typehistogram.If multipledataare giventhebars arearrangedsidebyside.
  - 'barstacked'isabar-type histogramwheremultipledata arestackedontopofeachother.
  - 'step'generatesalineplotthat isbydefaultunfilled.
  - 'stepfilled'generatesalineplotthat
  isbydefaultfilled.Defaultis'bar'
- **align**-{'left', 'mid', 'right'},optionalControlshowthehistog ramisplotted.

  - 'left':barsarecenteredontheleftbinedges.
  - 'mid':barsarecenteredbetweenthebinedges.
  - 'right':bars arecentered
  ontherightbinedges.Defaultis'mid'
- **orientation**-{'horizontal','vertical'},optional
  If'horizontal',barh willbe usedforbar-typehistograms andthebottomkwargwillbetheleft edges.
- **color**-colororarray_likeofcolorsor None, optional
  Colorspecor sequenceofcolor specs,oneperdataset.Default (None)usesthe standardlinecolorsequence.

  DefaultisNone
- **label**-strorNone,optional. DefaultisNone

Otherparameter
- ****kwargs** -Patch properties, it allows us to pass avariablenumberofkeywordargumentstoapythonfu nction.**denotesthistypeoffunction.

Example
```
importnumpyasnp
import matplotlib.pyplot as
pltplt.style.use('seaborn-
white')data =
np.random.randn(1000)plt.hist
(data);
```
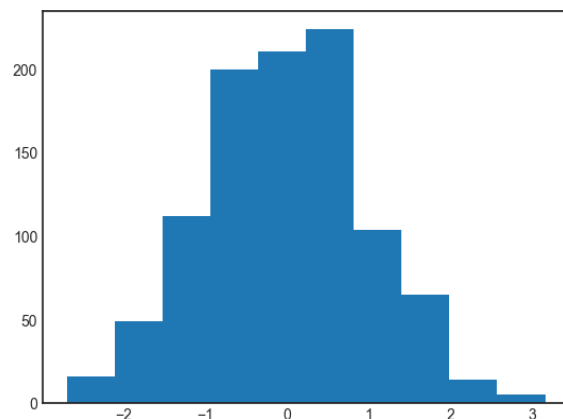

Thehist()functionhasmanyoptionstotuneboththecalculationandthedisplay;here'sanexampleofamorecustomizedhistogram.

```
plt.hist(data,bins=30,alpha=0.5,histtype='stepfilled',color='steelblue',edgecolor='none');
```

The plt.hist docstring has more information on other customization options available. I find this combination ofhisttype='stepfilled' along with some transparency alpha to be very useful when comparing histograms of severaldistributions

```
x1=np.random.normal(0,0.8,1000)
x2=np.random.normal(-2,1,1000)
x3=np.random.normal(3,2,1000)
kwargs = dict(histtype='stepfilled', alpha=0.3,
bins=40)plt.hist(x1, **kwargs)
plt.hist(x2,
**kwargs)plt.hist(x3,*
```
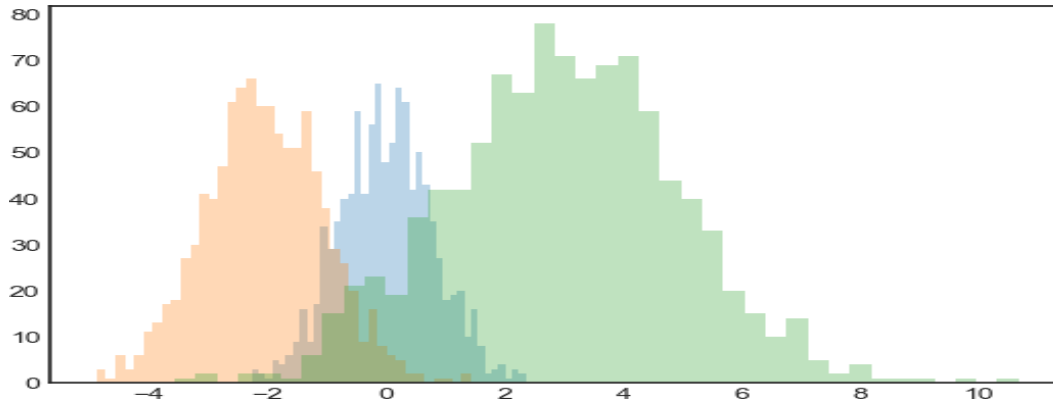

*kwargs);

Two-DimensionalHistogramsandBinnings

- Wecancreate histograms in two dimensions bydividingpoints amongtwodimensional bins.
- WewoulddefinexandyvaluesHereforexampleWe'llstartbydefiningsomedata—anxandyarraydrawnfrom amultivariateGaussiandistribution:
- Simplewaytoplotatwo-dimensionalhistogramistouseMatplotlib's plt.hist2d()function

Example

```
mean =[0, 0]
cov =[[1,1],[1,2]]
x,y=np.random.multivariate_normal(mean,cov,1000).Tplt.hist2d(x,
y,bins=30, cmap='Blues')
cb =
plt.colorbar()cb.set_label('
countsinbin')
```
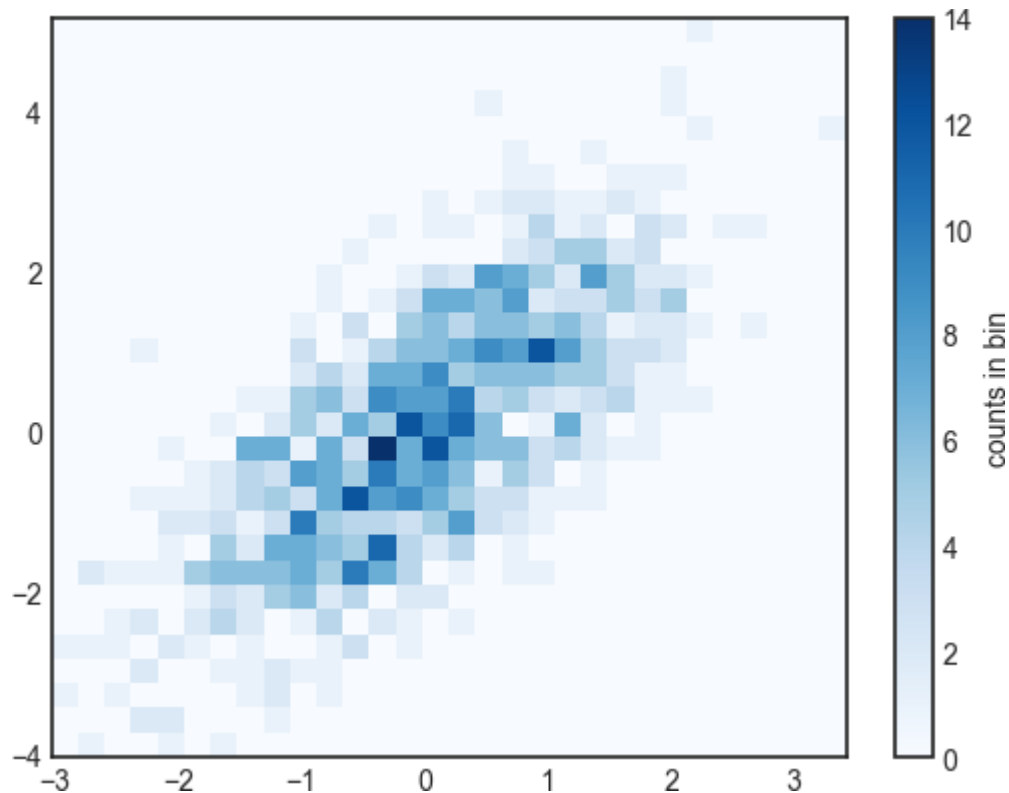
Legends

Plot legends give meaning to a visualization, assigning labels to the various plot elements. We previously saw howto create a simple legend; here we'll take a look at customizing the placement and aesthetics of the legend inMatplotlib.

Plot legends give meaning to a visualization, assigning labels to the various plot elements. We previously saw howto create a simple legend; here we'll take a look at customizing the placement and aesthetics of the legend inMatplotlib

```
plt.plot(x,np.sin(x),'-b',label='Sine')
plt.plot(x, np.cos(x), '--r',
label='Cosine')plt.legend();
```

CustomizingPlotLegends

Location and turn off the frame - We can specify the location and turn off the frame. By the parameter loc andframon.

```
ax.legend(loc='upper left',
frameon=False)fig
```

Numberofcolumns-Wecan usethencol command to specifythe number ofcolumns in the legend.

```
ax.legend(frameon=False, loc='lower center',
ncol=2)fig
```

Roundedbox,shadowandframetransparency

We can use a rounded box (fancybox) or add a shadow, change the transparency (alpha value) of the frame, orchangethe paddingaround the text.

```
ax.legend(fancybox=True, framealpha=1, shadow=True,
borderpad=1)fig
```

ChoosingElementsfortheLegend

- The legendincludesalllabeledelementsbydefault.We canchangewhichelementsandlabelsappearinthelegend byusingthe objects returned byplot commands.
- Theplt.plot()commandisabletocreatemultiplelinesatonce,andreturnsalistofcreatedlineinstances. Passinganyof theseto plt.legend()willtell itwhichtoidentify,alongwiththelabelswe'dliketospecify

```
y = np.sin(x[:, np.newaxis] + np.pi * np.arange(0, 2,
0.5))lines =plt.plot(x,y)
plt.legend(lines[:2],['first','second']);

# Applying label
individually.plt.plot(x,y[:,0],l
abel='first')
plt.plot(x,y[:,1],label='second')
plt.plot(x, y[:,
2:])plt.legend(framealpha=1,frameon=Tr
ue);
```
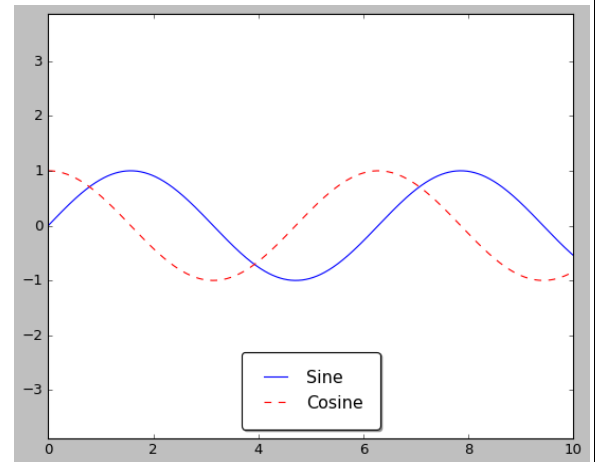

Multiplelegends

It is only possible to create a single legend for the entire plot. If youtry to create a second legend using plt.legend() or ax.legend(), it willsimplyoverridethefirstone.Wecanworkaroundthisbycreatinga new legend artist from scratch, and then using the lower-level ax.add_artist() method to manually add the secondartist to the plot

Example

```
importmatplotlib.pyplotaspltp
lt.style.use('classic')
importnumpyasnp
x= np.linspace(0,10,1000)
ax.legend(loc='lowercenter',frameon=True,shadow=True,borderpad=1,fancybox=True)fi
g
```

ColorBars

InMatplotlib,acolorbarisaseparateaxesthatcanprovideakeyforthemeaningofcolorsinaplot.Forcontinuouslabels basedon the color ofpoints,lines, or regions, alabeledcolorbarcan beagreattool.

Thesimplestcolorbarcan becreatedwith theplt.colorbar()function.

CustomizingColorbars

Choosingcolormap.

We can specify the colormap using the cmap argument to the plotting function that is creating the visualization.Broadly,wecan knowthreedifferent categories of colormaps:

- **Sequentialcolormaps**- Theseconsistofonecontinuoussequenceof colors(e.g.,binaryorviridis).
- **Divergentcolormaps**- These usuallycontaintwodistinctcolors,which show positiveandnegativedeviations from a mean(e.g., RdBu or PuOr).
- **Qualitativecolormaps** -Thesemixcolorswithnoparticularsequence (e.g.,rainboworjet).

Colorlimitsandextensions

- Matplotliballowsforalargerangeofcolorbarcustomization.Thecolorbaritselfissimplyaninstanceofplt.Axes,soall of theaxesand tick formattingtricks we'velearnedareapplicable.
- Wecannarrowthecolorlimitsandindicatetheout-of-boundsvalueswithatriangulararrowatthetopandbottombysettingtheextend property.

```
plt.subplot(1, 2,
2)plt.imshow(I,
cmap='RdBu')plt.colorbar(e
xtend='both')plt.clim(-1,1);
```
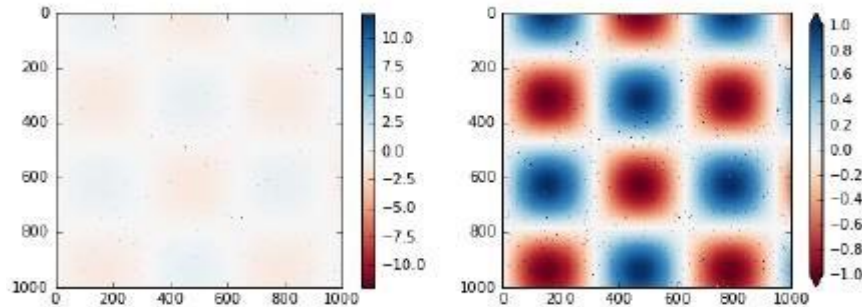
iscretecolorbars

Colormapsarebydefaultcontinuous,butsometimesyou'dliketorepresentdiscretevalues.Theeasiestwaytodothisistousetheplt.cm.get_cmap() function, and pass the name of a suitable colormapalongwith thenumber of desired bins.

```
plt.imshow(I, cmap=plt.cm.get_cmap('Blues',
6))plt.colorbar()
plt.clim(-1,1);
```
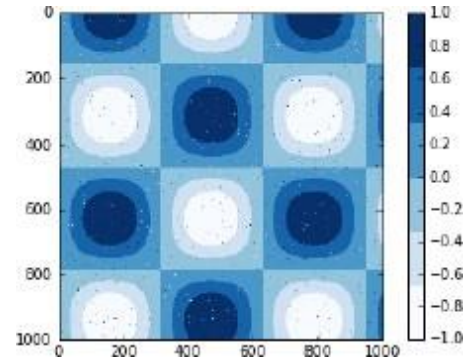
Subplots

- Matplotlibhastheconcept ofsubplots:groupsof smalleraxesthat canexisttogetherwithin asingle figure.
- Thesesubplotsmight be insets, gridsofplots,or othermorecomplicatedlayouts.
- We'llexplorefourroutinesforcreatingsubplotsinMatplotlib.
  - plt.axes: Subplots byHand
  - plt.subplot:Simple Gridsof Subplots
  - plt.subplots:TheWholeGrid inOneGo
  - plt.GridSpec:MoreComplicatedArrangements

plt.axes:Subplotsby Hand

- Themostbasicmethodofcreatinganaxesistousetheplt.axesfunction.Aswe'veseenpreviously,bydefaultthis creates astandard axes objectthat fills theentirefigure.
- plt.axesalsotakes anoptionalargumentthat isalist offournumbers inthe figurecoordinatesystem.
- These numbers represent [bottom, left, width,height] in the figure coordinate system, which ranges from 0 atthebottom left of the figureto 1 at the top right of thefigure.

Forexample,
we might create an inset axes at the top-right corner of anotheraxes by setting the x and y position to 0.65 (that is, starting at65% of the width and 65% of the height of the figure) and the xand y extents to 0.2 (that is, the size of the axes is 20% of thewidthand 20%of theheightof the figure).

```python
importmatplotlib.pyplotasplti
mport numpyas np
ax1=plt.axes()#standardaxesax2=p
lt.axes([0.65,0.65,0.2,0.2])
```
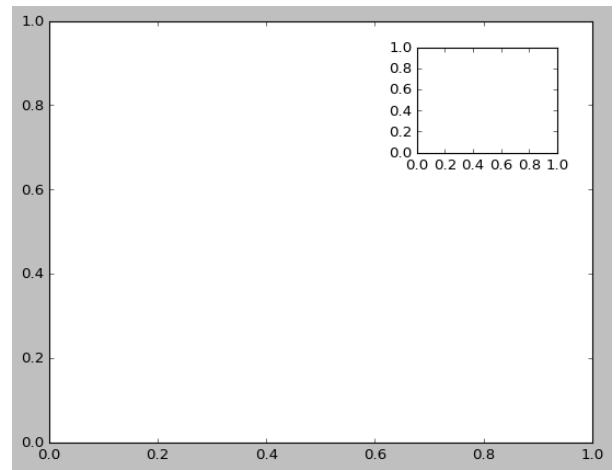
Verticalsubplot

Theequivalentofplt.axes()commandwithintheobject-
oriented interface is ig.add_axes(). Let's use thisto create two verticallystackedaxes.

```python
fig=plt.figure()
ax1 =fig.add_axes([0.1,0.5, 0.8,0.4],
xticklabels=[],ylim=(-1.2,1.2))
ax2 =fig.add_axes([0.1,0.1, 0.8,0.4],
ylim=(-1.2,1.2))
x = np.linspace(0,
10)ax1.plot(np.sin(x)
)
ax2.plot(np.cos(x));
```

- We now have two axes (the top with no ticklabels) that are just touching: the bottom of theupper panel (at position 0.5) matches the top ofthelower panel (at position0.1+0.4).
- If the axis value is changed in second plot boththe plots are separated with each other, exampleax2 =fig.add_axes([0.1,0.01,0.8, 0.4

plt.subplot:SimpleGridsofSubplots

- Matplotlibhasseveralconvenience routines toaligncolumnsor rowsofsubplots.
- Thelowestleveloftheseisplt.subplot(),which creates asinglesubplotwithinagrid.

- Thiscommandtakesthreeintegerarguments—
  the number of rows, the numberof columns, and the index of the plot to becreated in this scheme, which runs from theupperleft to thebottom right

```python
for i in range(1,
7):plt.subplot(2,3,i
)
plt.text(0.5,0.5,str((2,3,i)),fo
ntsize=18,ha='center')
```

plt.subplots:TheWholeGrid in OneGo

- The approach just described can become quite tedious when you're creating a largegrid of subplots,especiallyifyou'dliketohidethex-andy-axislabels on theinner plots.
- For this purpose, plt.subplots() is the easiertoolto use(notethes at theend of subplots).
- Ratherthancreatingasinglesubplot,thisfunction creates a full grid of subplots in asingle line, returningthem inaNumPyarray.
- The arguments are the number of rows andnumberofcolumns,alongwithoptionalkeyw ords sharex and sharey, which allow youto specify the relationships between differentaxes.
- Herewe'llcreatea2×3gridofsubplots,where all axes in the same row share their y-axis scale, and all axes in the same columnsharetheirx-axis scale

```
fig, ax = plt.subplots(2, 3,
    sharex='col',sharey='row')
```

Note that by specifying sharex and sharey, we'veautomatically removed inner labels on the grid tomakethe plot cleaner.
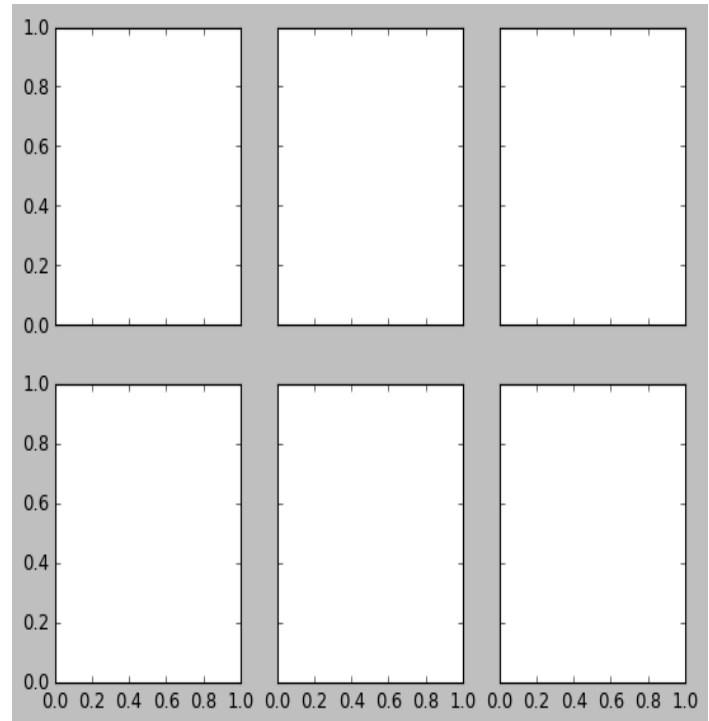
plt.GridSpec:MoreComplicatedArrangements

Togobeyondaregulargridtosubplotsthatspanmultiplerowsandcolumns,plt.GridSpec() isthebesttool.Theplt.GridSpec()objectdoesnotcreateaplotbyitself;itissimplyaconvenientinterfacethatisrecognized bythe plt.subplot() command.

Forexample,agridspecforagridoftworowsandthreecolumnswithsomespecifiedwidthandheightspacelooks likethis:

```
grid = plt.GridSpec(2, 3, wspace=0.4,
hspace=0.3)Fromthiswecanspecifysubplotlocationsan
dextentsplt.subplot(grid[0, 0])
plt.subplot(grid[0,1:])
plt.subplot(grid[1,:2])
plt.subplot(grid[1,2]);
```
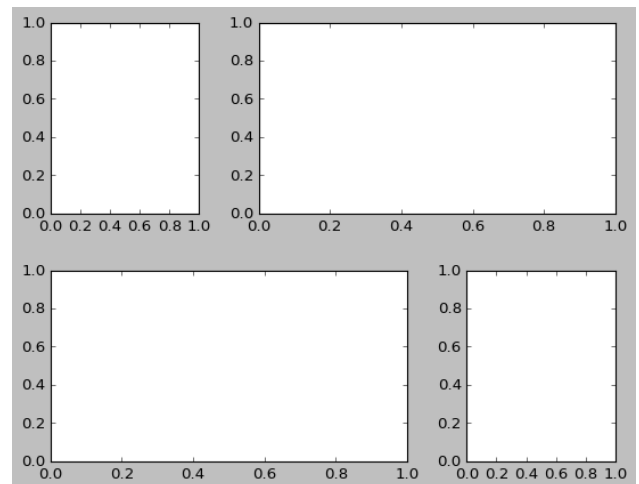
TextandAnnotation

- The most basic types of annotations we will use are axes labels and titles, here we will see some morevisualizationand annotation information's.

15

- Textannotationcanbedonemanuallywiththeplt.text/ax.textcommand,whichwillplacetextataparticularx/yvalue.
- The ax.text method takes an x position, a y position, a string, and then optional keywords specifying thecolor, size, style, alignment, and other properties of the text. Here we used ha='right' and ha='center', wherehais short for horizontal alignment.

TransformsandTextPosition

- We anchored our text annotations to data locations. Sometimes it's preferable to anchor the text to a positiononthe axes orfigure, independentofthedata.InMatplotlib, wedothis bymodifyingthe transform.
- Anygraphics displayframeworkneeds somescheme fortranslatingbetween coordinatesystems.
- Mathematically, such coordinate transformations are relatively straightforward, and Matplotlib has a well-developedsetoftoolsthatitusesinternallytoperformthem(thetoolscanbeexploredinthematplotlib.transformssub module).
- There arethreepredefined transformsthat canbeusefulin thissituation.

    o ax.transData   -Transformassociated withdatacoordinates
    o ax.transAxes-Transform associated with theaxes (inunits of axesdimensions)
    o fig.transFigure-Transform associatedwith thefigure(in unitsof figuredimensions)

Example
```
import matplotlib.pyplot as
pltimport matplotlib as
mplplt.style.use('seaborn-
whitegrid')import numpyas np
importpandasaspd
fig,ax=plt.subplots(facecolor='lightgray')a
x.axis([0, 10, 0,10])
#transform=ax.transDataisthedefault,butwe'llspecifyitanywayax.text(1,
5,".Data: (1, 5)",transform=ax.transData)
ax.text(0.5,0.1,".Axes:(0.5,0.1)",transform=ax.transAxes)
ax.text(0.2,0.2,".Figure:(0.2,0.2)",transform=fig.transFigure);
```
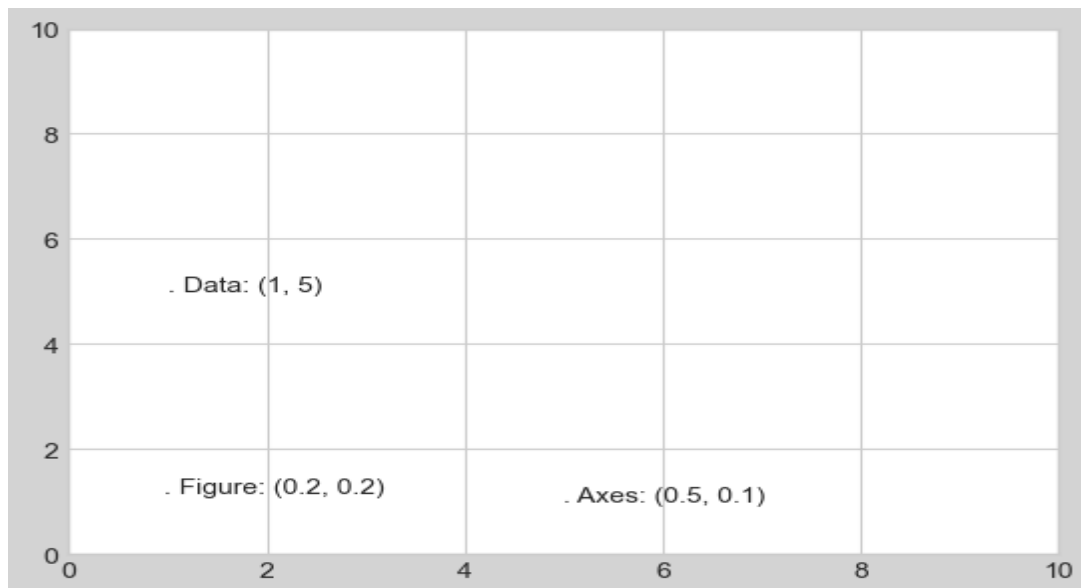
Note that by default, the text is aligned above and to the left of the specified coordinates; here the "." at thebeginningofeach stringwill approximatelymarkthe givencoordinate location.

The transData coordinates give the usual data coordinates associated with the x- and y-axis labels. The transAxescoordinates give the location from the bottom-left corner of the axes (here the white box) as a fraction of the axessize.

The transfigure coordinates are similar, but specify the position fromthe bottom left of the figure(here the graybox)as afraction ofthefiguresize.
Notice now that if we change the axes limits, it is only the transData coordinates that will be affected, while theothersremain stationary.

ArrowsandAnnotation
- Alongwith tickmarksandtext, anotheruseful annotation markis thesimplearrow.
- Drawingarrowsin Matplotlibisnotmuch harderbecause thereisaplt.arrow() functionavailable.
- ThearrowsitcreatesareSVG(scalablevectorgraphics)objectsthatwillbesubjecttothevaryingaspectratioofyour plots,and theresultis rarelywhat theuser intended.
- Thearrowstyleiscontrolledthroughthe arrowpropsdictionary,whichhas numerousoptionsavailable.

Three-DimensionalPlottinginMatplotlib

Weenable three-dimensional plots byimportingthemplot3d toolkit, includedwith themainMatplotlib installation.

```
importnumpyasnp
import matplotlib.pyplot as
pltfrommpl_toolkitsimportmplot3
dfig=plt.figure()
ax=plt.axes(projection='3d')
```

With this 3D axes enabled, we can now plot a varietyofthree-dimensional plot types.
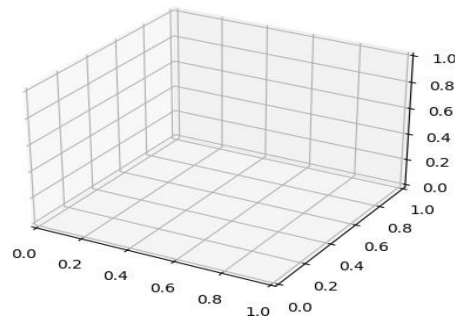
Three-DimensionalPointsandLines
Themostbasicthree-dimensionalplotis alineorscatterplotcreatedfromsetsof(x,y,z)triples.
Inanalogywiththemorecommontwo-dimensionalplotsdiscussedearlier,wecancreatetheseusingtheax.plot3D andax.scatter3Dfunctions

```
importnumpyasnp
import matplotlib.pyplot as
pltfrommpl_toolkitsimportmplot3
dax= plt.axes(projection='3d')
# Data for a three-dimensional
linezline = np.linspace(0, 15,
1000)xline =np.sin(zline)
yline =
np.cos(zline)ax.plot3D(xline,yline,
zline,'gray')
# Data for three-dimensional scattered
pointszdata=15*np.random.random(100)
xdata = np.sin(zdata) + 0.1 *
np.random.randn(100)ydata=np.cos(zdata)+0.1*np.ra
ndom.randn(100)
```
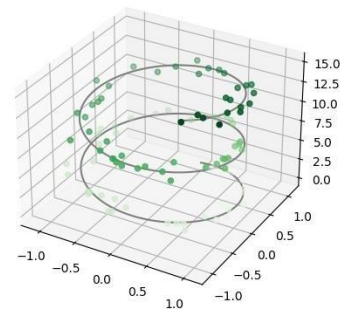
```
ax.scatter3D(xdata,ydata,zdata,c=zdata,cmap='Greens');plt.show()
```

Noticethat bydefault, thescatter pointshavetheir transparencyadjusted togiveasenseofdepth onthepage.

Three-DimensionalContourPlots

- mplot3dcontainstools tocreatethree-dimensionalrelief plotsusingthe same inputs.
- Liketwo-dimensionalax.contourplots,ax.contour3Drequiresalltheinputdatatobeintheformoftwo-dimensionalregular grids, with theZdataevaluated ateach point.
- Herewe'llshowathree-dimensionalcontourdiagramofathree dimensional sinusoidal function

```
importnumpyasnp
import matplotlib.pyplot as
pltfrommpl_toolkitsimportmplot3
ddeff(x, y):
    returnnp.sin(np.sqrt(x**
2+y**2))x=np.linspace(-6,6,30)
y = np.linspace(-6, 6,
30)X, Y = np.meshgrid(x,
y)Z =f(X,Y)
fig=plt.figure()
ax =
plt.axes(projection='3d')ax.contour3D(X
,Y,Z,50,cmap='binary')ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
```
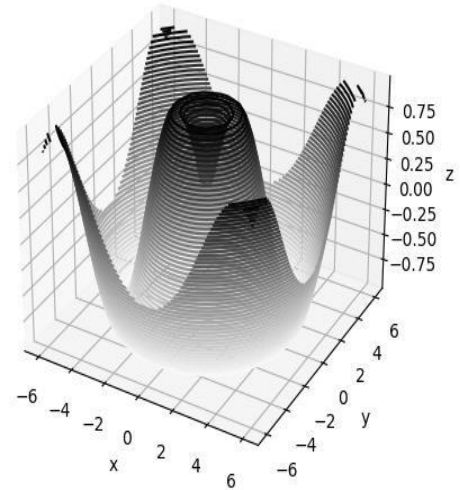
Sometimesthedefaultviewingangleisnotoptimal,inwhichcasewecanusetheview_initmethodtosettheelevationand azimuthal angles.

```
ax.view_init(60,
35)fig
```

WireframesandSurfacePlots

- Twoothertypesof three-dimensionalplotsthat workon griddeddataarewireframes andsurfaceplots.
- Thesetakeagridofvaluesandprojectitontothespecifiedthreedimensionalsurface,andcanmaketheresultingthree-dimensional forms quiteeasytovisualize.

```
importnumpyasnp
import matplotlib.pyplot as
pltfrom mpl_toolkits import
mplot3dfig=plt.figure()
ax =
plt.axes(projection='3d')ax.plot_wirefra
me(X,Y,Z,color='black')ax.set_title('wire
frame');
plt.show()
```
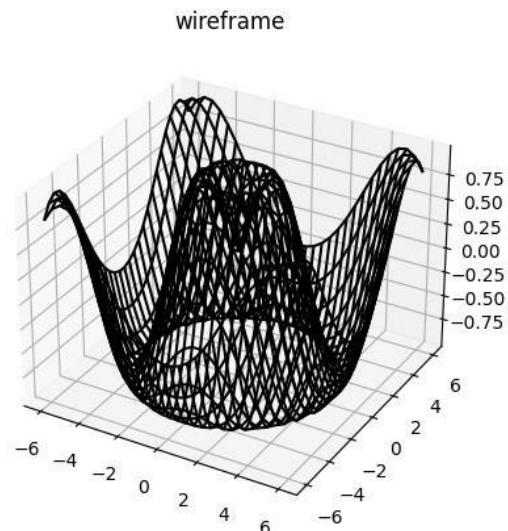
- Asurfaceplotislikeawireframeplot,buteachfaceofthewi reframe is afilled polygon.

- Addingacolormap tothefilled polygons canaid perception ofthetopologyofthesurfacebeingvisualized

```python
importnumpyasnp
import matplotlib.pyplot as
pltfrommpl_toolkitsimportmplot3
dax= plt.axes(projection='3d')
ax.plot_surface(X, Y, Z, rstride=1,
cstride=1,cmap='viridis',
edgecolor='none')ax.set_title('surface')
plt.show()
```
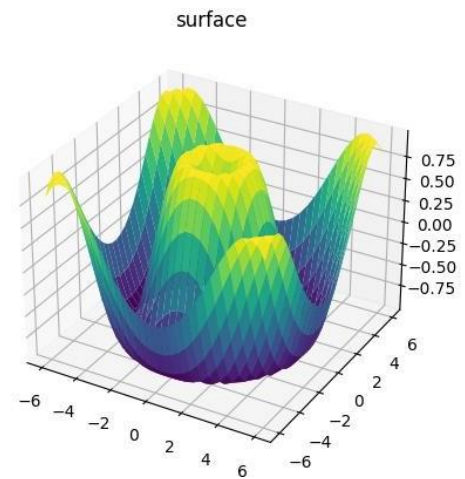
SurfaceTriangulations

- For some applications, the evenly sampled grids required bytheprecedingroutinesareoverlyrestrictiveandinconvenient.
- Inthesesituations,thetriangulation-basedplots can beveryuseful.

```python
importnumpyasnp
import matplotlib.pyplot as
pltfrommpl_toolkitsimportmplot3
d
theta=2*np.pi*np.random.random(1000)r
=6*np.random.random(1000)
x      =      np.ravel(r      *
np.sin(theta))y = np.ravel(r *
np.cos(theta))z =f(x, y)
ax=plt.axes(projection='3d')
ax.scatter(x,y,z,c=z,cmap='viridis',linewidth=0.5)
```
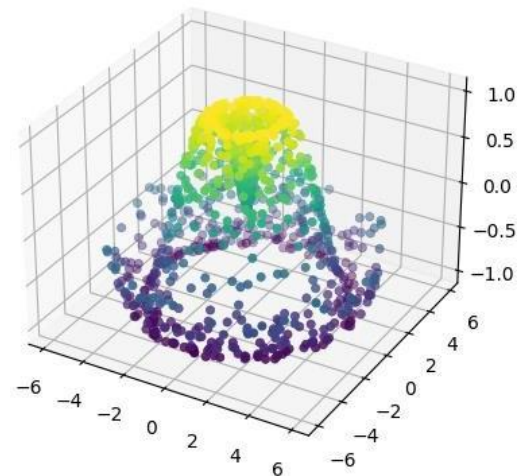
GeographicDatawithBasemap

- Onecommontypeofvisualizationindatascienceisthatofgeog raphicdata.
- Matplotlib'smaintoolforthistypeofvisualizationistheBasemaptoolkit,whichisoneofseveralMatplotlibtoolkits that liveunder thempl_toolkitsnamespace.
- Basemapisa usefultoolforPythonusers tohaveintheir virtualtoolbelts
- InstallationofBasemap.OnceyouhavetheBasemaptoolkitinstalledandimported,geographicplotsalsorequirethe PILpackageinPython2,orthepillowpackage inPython3.

```python
import numpy asnp
import matplotlib.pyplot asplt
from mpl_toolkits.basemap import
Basemapplt.figure(figsize=(8,8))
m = Basemap(projection='ortho',
resolution=None,lat_0=50, lon_0=-100)
m.bluemarble(scale=0.5);
```

- Matplotlibaxesthatunderstandssphericalcoordinatesandallo ws us toeasilyover-plot data on themap

- We'lluseanetopoimage(whichshowstopographicalfeaturesbothonlandandundertheocean)asthemapbackground

Programto displaypticulararareaofthemapwith latitudeandlongitudelines


```
importnumpyasnp
importmatplotlib.pyplotasplt
frommpl_toolkits.basemapimportBasemapfr
om itertoolsimportchain
fig=plt.figure(figsize=(8,8))
m = Basemap(projection='lcc',
resolution=None,width=8E6,height=8E6,
lat_0=45, lon_0=-
100,)m.etopo(scale=0.5,
alpha=0.5)defdraw_map(m,sc
ale=0.2):
# draw a shaded-relief
   imagem.shadedrelief(scale=scale)
# lats and longs are returned as a
dictionarylats=m.drawparallels(np.linspace(-
90,90,13))
lons = m.drawmeridians(np.linspace(-180, 180,
13))# keyscontaintheplt.Line2D instances
lat_lines = chain(*(tup[1][0] for tup in
lats.items()))lon_lines=chain(*(tup[1][0]for
tupinlons.items()))all_lines=chain(lat_lines,lon_lines)
#cyclethroughtheselinesandsetthedesired stylefor
lineinall_lines:
   line.set(linestyle='-',alpha=0.3,color='r')
```

MapProjections

TheBasemappackageimplementssseveraldozensuchprojections,allreferencedbya shortformatcode.Herewe'llbrieflydemonstratesomeof themorecommon ones.

- Cylindricalprojections
- Pseudo-cylindricalprojections
- Perspectiveprojections
- Conicprojections

Cylindricalprojection

- Thesimplestof mapprojectionsare cylindricalprojections,inwhichlinesof constantlatitude andlongitudearemapped to horizontal and vertical lines, respectively.
- Thistypeofmappingrepresentsequatorialregionsquitewell,butresultsinextremedistortionsnearthepoles.
- Thespacingoflatitudelinesvariesbetweendifferentcylindricalprojections,leadingtodifferentconservationproperties,and different distortionnear thepoles.
- OthercylindricalprojectionsaretheMercator(projection='merc')andthecylindricalequal-area(projection='cea')projections.
- TheadditionalargumentstoBasemapforthisviewspecifythelatitude(lat)andlongitude(lon)ofthelower-leftcorner (llcrnr) and upper-rightcorner (urcrnr)for thedesired map,in unitsof degrees.
  ```
  importnumpy asnp
  importmatplotlib.pyplot asplt
  frommpl_toolkits.basemap importBasemap
  ```

```
fig=plt.figure(figsize=(8,6),edgecolor='w')
m = Basemap(projection='cyl',
resolution=None,llcrnrlat=-90,urcrnrlat=90,
llcrnrlon=-180, urcrnrlon=180,
)draw_map(m)
```
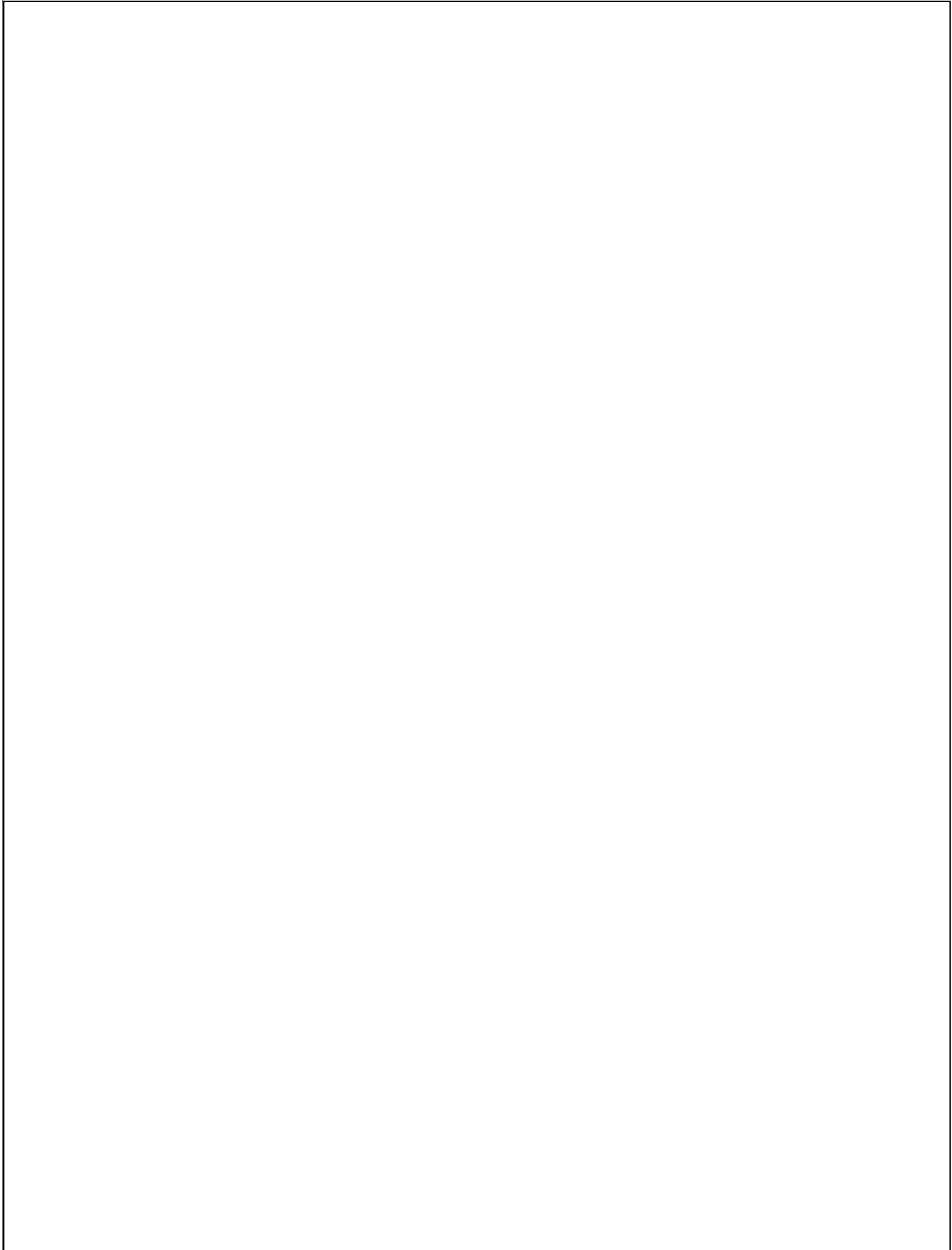

Pseudo-cylindricalprojections

- Pseudo-cylindricalprojectionsrelaxtherequirementthatmeridians(linesofconstantlongitude)remainvertical;this cangive betterproperties nearthepoles ofthe projection.
- TheMollweideprojection(projection='moll')isonecommonexampleofthis,inwhichallmeridiansareellipticalarcs
- Itisconstructedsoasto
- preserve area across the map: though there aredistortions near the poles, the area of smallpatchesreflects thetruearea.
- Otherpseudo-cylindricalprojectionsarethesinusoidal(projection='sinu')andRobinson(projection='robin')projections.

- The extra arguments to Basemap here refer tothecentrallatitude(lat_0)andlongitude(lon_0) forthe desired map.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import
Basemapfig=plt.figure(figsize=(8,6),edgecolor
='w')
m = Basemap(projection='moll',
resolution=None,lat_0=0,lon_0=0)
draw_map(m)
```

Perspectiveprojections

- Perspective projections are constructed using a particular choice of perspective point, similar to if youphotographed the Earth from a particular point in space (a point which, for some projections, technically lieswithinthe Earth!).

- One common example is the orthographic projection (projection='ortho'), which shows one side of the globeasseen from aviewer ataverylongdistance.
- Thus,itcan showonlyhalftheglobeat atime.
- Otherperspective-basedprojectionsincludethegnomonicprojection(projection='gnom')andstereographicprojection (projection='stere').
- These are often the most useful for showing smallportions of themap.

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
fig = plt.figure(figsize=(8,8))
m = Basemap(projection='ortho',
resolution=None, lat_0=50, lon_0=0)
draw_map(m);
```

Conicprojections

- Aconicprojection projectsthemaponto asingle cone, whichis thenunrolled.
- Thiscanleadtoverygoodlocalproperties,butregionsfarfromthefocuspointoftheconemaybecomeverydistorted.
- Oneexampleofthisisthe Lambert conformalconicprojection(projection='lcc').
- It projects the map onto a cone arranged in such a way that two standard parallels (specified in Basemap bylat_1 and lat_2) have well-represented distances, with scale decreasing between them and increasing outsideofthem.
- Otherusefulconicprojectionsaretheequidistantconic(projection='eqdc')andtheAlbersequal-area(projection='aea')projection

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
fig = plt.figure(figsize=(8,8))
m=Basemap(projection='lcc',resolution=None,
lon_0=0, lat_0=50, lat_1=45, lat_2=55, width=1.6E7,
height=1.2E7)draw_map(m)
```

DrawingaMapBackground

TheBasemappackagecontainsarangeofusefulfunctionsfordrawingbordersofphysicalfeatureslikecontinents,oceans,lakes, andrivers,as wellas politicalboundaries such ascountries andUSstates andcounties.

ThefollowingaresomeoftheavailabledrawingfunctionsthatyoumaywishtoexploreusingIPython'shelpfeatures:

• Physical boundaries and bodies of
waterdrawcoastlines()-
Drawcontinentalcoastlines
drawlsmask() - Draw a mask between the land and sea, for use with projecting images on one or the
otherdrawmapboundary() -Draw themapboundary, includingthe fillcolorfor oceans
drawrivers() -Drawriversonthemap
fillcontinents()-Fill the continentswith agiven color; optionallyfilllakeswithanothercolor

• Politicalboundaries
drawcountries() - Draw country
boundariesdrawstates() - Draw US state
boundariesdrawcounties()-
DrawUScountyboundaries

• Mapfeatures
drawgreatcircle() -Drawagreat
circlebetweentwopointsdrawparallels() - Draw lines of
constant latitudedrawmeridians() - Draw lines of constant
longitudedrawmapscale()-Drawa linear scaleon the map

• Whole-globeimages
bluemarble() -
ProjectNASA'sbluemarbleimageontothemapshadedrelief() -
Project a shaded relief image onto the mapetopo()-Drawan
etopo reliefimageonto themap
warpimage()-Projectauser-provided imageontothemap

PlottingDataonMaps
- TheBasemap toolkitis the abilityto over-plot avarietyof dataonto a mapbackground.
- There are many map-specific functions available as methods of the Basemap

instance.Someofthesemap-specificmethods are:
contour()/contourf()-
Drawcontourlinesorfilledcontoursimshow()-Draw an
image
pcolor()/pcolormesh()-Draw
apseudocolorplotforirregular/regularmeshesplot()-Draw lines and/ormarkers
scatter()-
Drawpointswithmarkersquiver()-
Draw vectors
barbs() - Draw wind
barbsdrawgreatcircle()-
Drawagreatcircle

VisualizationwithSeaborn

ThemainideaofSeabornisthatitprovideshigh-

levelcommandstocreateavarietyofplottypesusefulforstatisticaldataexploration, and even somestatistical modelfitting.

23

levelcommandstocreateavarietyofplottypesusefulforstatisticaldataexploration, and even somestatistical modelfitting.

Histograms,KDE,anddensities

- Instatisticaldatavisualization,allyouwantistoplothistograms and joint distributions of variables. We haveseenthat thisis relativelystraightforwardin Matplotlib

- Rather than a histogram, we can get a smooth estimate ofthe distribution using a kernel density estimation, whichSeaborndoes with sns.kdeplot

```
import pandas as
pdimportseabornassn
s
data=np.random.multivariate_normal([0,0],[[5,2],[2,
2]],size=2000)
data=pd.DataFrame(data,columns=['x','y'])
forcolin'xy':
        sns.kdeplot(data[col],shade=True)
```


- HistogramsandKDE canbecombined usingdistplot
```
sns.distplot(data['x'])
sns.distplot(data['y']);
```


- Ifwepassthefulltwo-dimensionaldatasettokdeplot,wewillgetatwo-dimensional visualization ofthe data.
- Wecanseethe jointdistributionand themarginaldistributions togetherusing sns.jointplot.
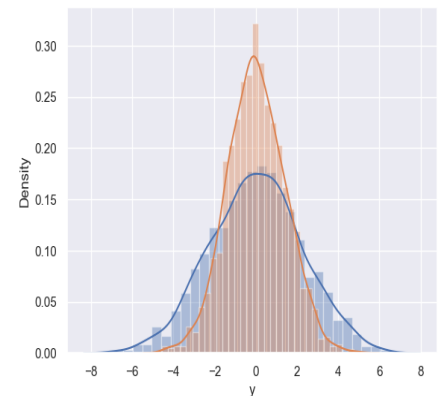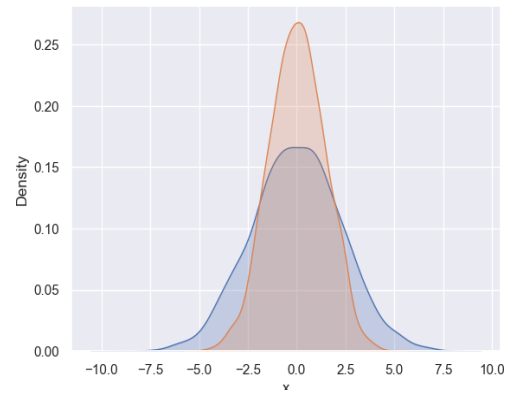
Pairplots

Whenyougeneralizejointplotstodatasetsoflargerdimensions,youendupwithpairplots.Thisisvery usefulforexploringcorrelationsbetweenmultidimensionaldata,whenyou'dliketoplotallpairsofvaluesagainst eachother.

We'lldemothiswith theIrisdataset, whichlistsmeasurementsof petalsandsepalsof threeirisspecies:

```
importseabornas sns
iris
=sns.load_dataset("iris")sns.pairplot(iris,
hue='species',size=2.5);
```
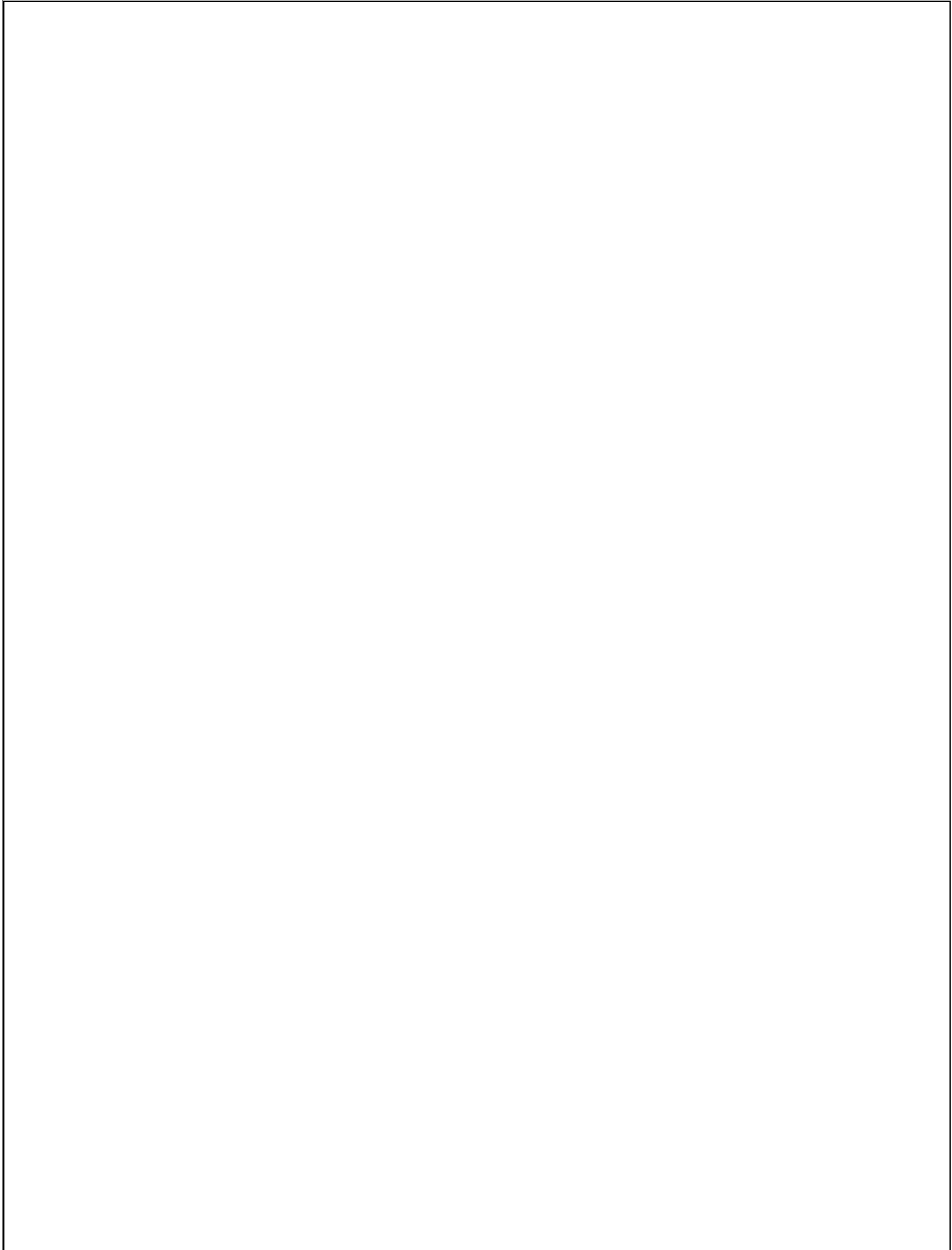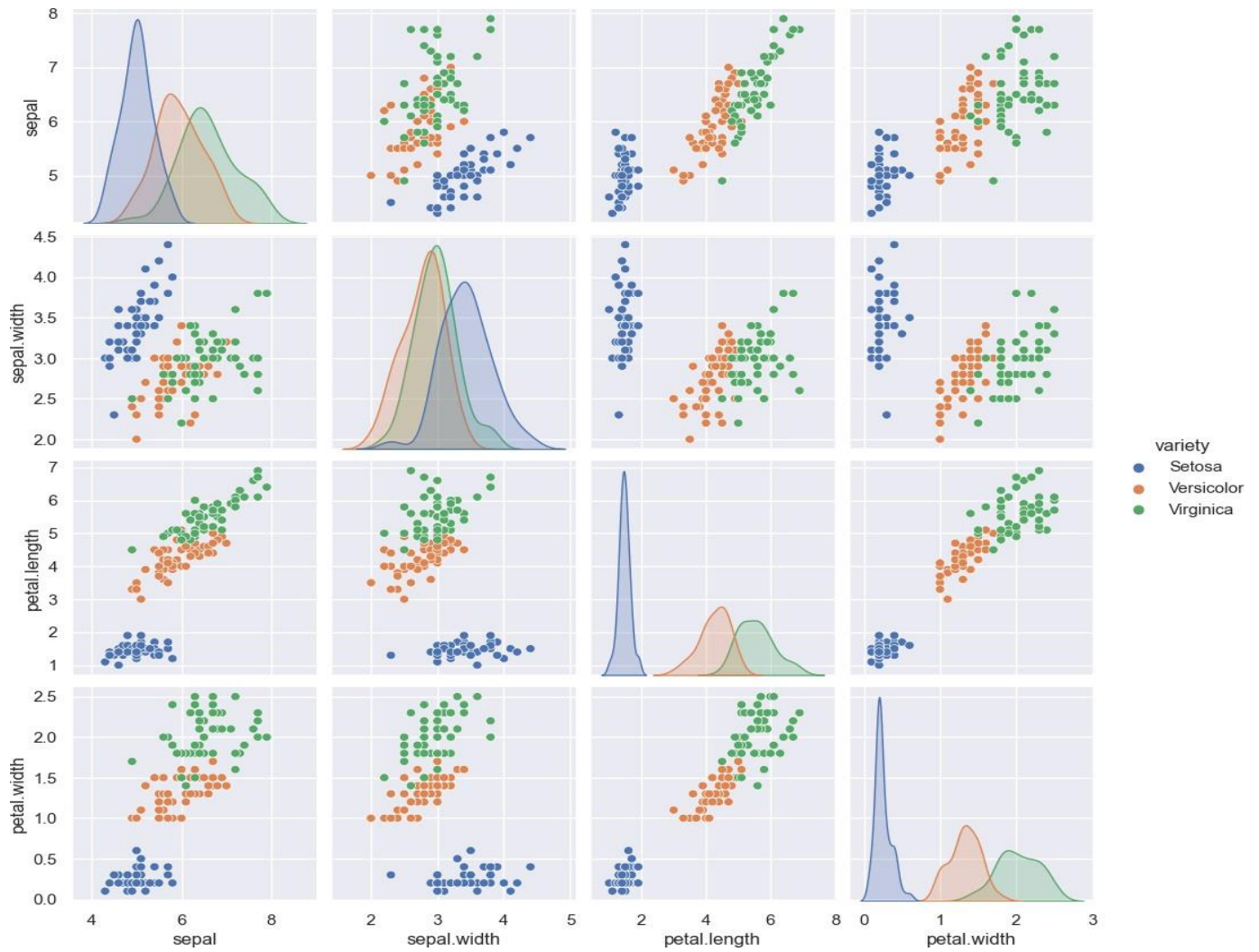
## Facetedhistograms

- Sometimesthebestwaytoviewdataisviahistogramsofsubsets.Seaborn'sFacetGridmakesthisextremelysimple.
- We'lltakealookatsomedatathatshowstheamountthatrestaurantstaffreceiveintipsbasedonvariousindicatordata

25
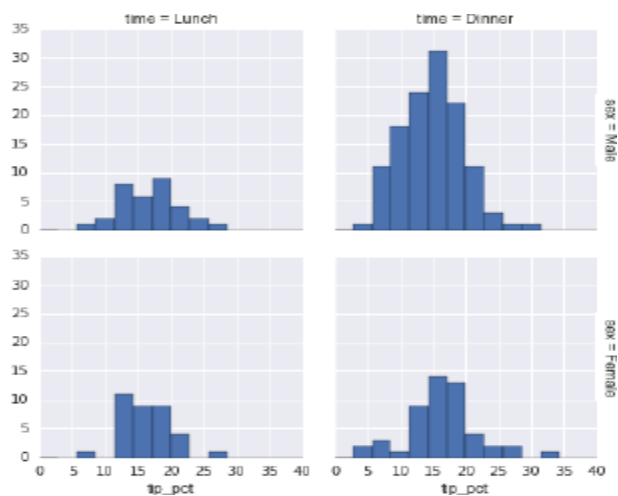
Factorplots

Factorplotscanbeusefulforthiskindofvisualizationaswell.Thisallowsyoutoviewthedistributionofaparameter within bins defined byanyotherparameter.
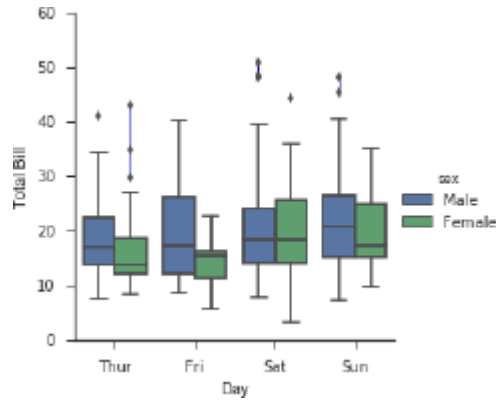
Jointdistributions

Similartothepairplotwesawearlier,wecanusesns.jointplottoshowthejointdistributionbetweendifferentdatasets, alongwith theassociated marginal distributions.
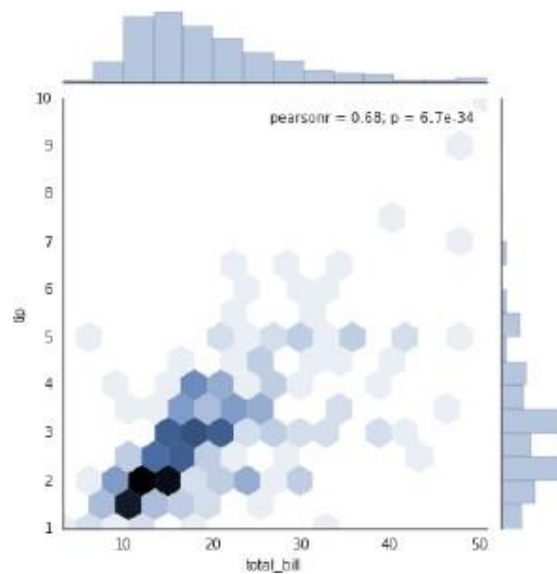
Barplots

Timeseriescanbeplottedwithsns.factorplot.

26