ExploratoryAnalysisofGeolocationalData

PHASE-5:DOCUMENTATION AND SUBMISSION

Objective

This project involves the use of K-Means Clustering to find the bestaccommodationforstudentsinBangalore(oranyothercityofyourchoice)byclassifying accommodation for incoming students on the basis of theirpreferencesonamenities,budgetandproximitytothelocation.

ProjectContext

Implementing the project will take you through the daily life of a data scienceengineer-fromdatapreparationonreal-lifedatasets,tovisualisingthedataandrunningmachinelearningalgorithms,topresentingthe results.

Inthefast-moving,effort-intenseenvironmentthattheaveragepersoninhabits, It'safrequentoccurrencethatoneistootiredtofixoneselfahome-cooked

meal. And of course, even if one gets home-cooked meals every day, it is notunusual to want to go out for a good meal every once in a while forsocial/recreational purposes. Either way, it's a commonly understood idea thatregardless of where one lives, the food one eats is an important aspect of thelifestyleoneleads.

Now,imagineascenariowhereapersonhasnewlymovedintoanewlocation.They already have certain preferences, certain tastes. It would save both thestudent and the food providers a lot of hassle if the student lived close to theirpreferred outlets. Convenience means better sales, and saved time for thecustomer.

Food delivery apps aside, managers of restaurant chains and hotels can alsoleverage this information. For example, if a manager of a restaurant alreadyknows the demographic of his current customers, they'd ideally want to open ata location where this demographic is at its highest concentration, ensuring shortcommute times to the location and more customers served.If potential hotellocationsarebeingevaluated,asitethatcaterstoawidevarietyoftasteswouldbeideal,sinceonewouldwanteveryguesttohavesomethingtotheirliking.

This project is a good start for beginners and a refresher for professionals whohave dabbled in python / ML before. The methodology can be applied to anylocationofone'schoosing,sofeelfreetoinnovate!

ProjectStages

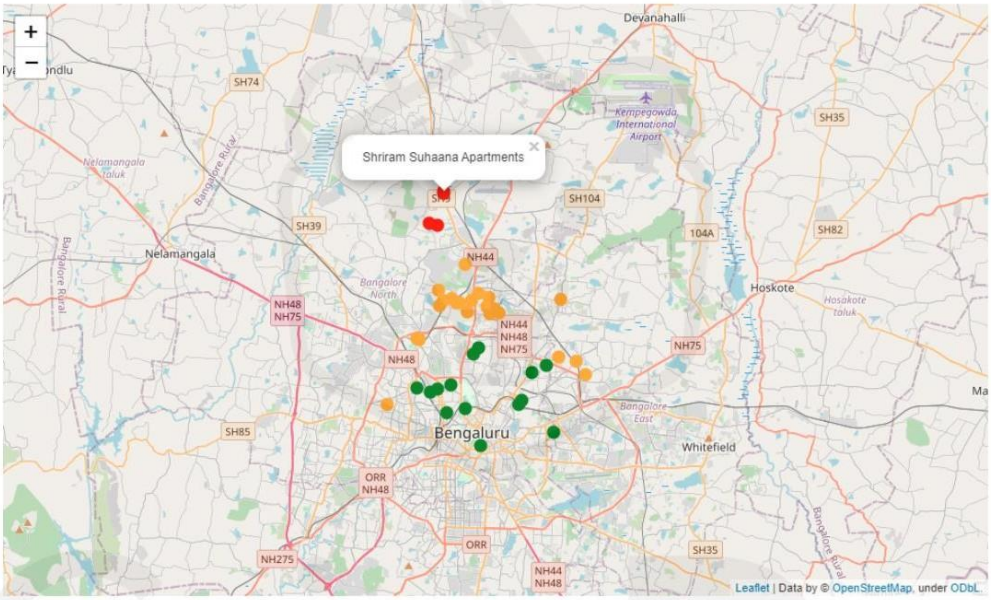
Theprojectconsistsofthefollowingstages:

High-LevelApproach

- FetchDatasetsfromtherelevantlocations(DataCollection)
- CleantheDatasetstopreparethemforanalysis.(DataCleaningviaPandas)

- Visualisethedatausingboxplots.(UsingMatplotlib/Seaborn/Pandas)
- FetchGeolocationalDatafromtheFoursquareAPI.(RESTAPIs)
- UseK-MeansClusteringtoclusterthelocations(UsingScikitLearn)
- Presentfindingsonamap.(UsingFolium/Seaborn)The

desiredendresultofthisprojectissomethinglikethis:



Applications

K-Meansclusteringisusedinavarietyofexamplesorbusinesscasesinreallife,like:

- Academicperformance(groupingstudentsbytheirlearningrate)
- Diagnosticsystems(groupingsystemfaultsundervariousreasons)
- Searchengines(groupingsearchresults)
- Wirelesssensornetworks(Mappingnetworks)

TheFourSquareAPIdatacanbeusedfor:

- BuildingarestaurantreviewapplikeSwiggyZomatoetc.
- SupportingaridesharingservicelikeUberPool

GettingStarted

Weneeddatatododataanalysis!Fetchthedataweneedandsetupyourenvironmentbeforeyo umoveontodataanalysis.

AboutDataset

Foodchoicesandpreferencesofcollegestudents

This dataset includes information on food choices, nutrition, preferences, childhoodfavorites, and other information from college students. There are 126 responses fromstudents. Data is raw and uncleaned. Cleaning is in the process and as soon as that is done,additionalversionsofthe datawill beposted.

Requirements

- Headovertoagivendatasetwe'llbeusing.Ifyouruninto"toomanyrequests"
- Notethatyou'llneedbothfiles,thecsvandthedocument-thedocumentexplainsthemeaningoftheCSVvalues.
- Youwillneeda **jupyternotebook** installedinyourenvironment.
- Explorethe `codebook_food` file.Therearearound70parameters.Notallof themarerelevant.Thinkcarefullyaboutwhichonesarethemostuseful.Which ones can be used to quantifiably differentiate students? A goodexample of this is income. A more qualitative parameter like "How muchtheylikevegetables"ona1-5scalemightnotbeasuseful.
- Extractthemostrelevantfeaturesintoapandasdataframe.
- ThisprocessofExtractingthefeatures,(anddealingwithdifferentkindsofvaluesa swellasNaNvalues)isknownas**DataCleaning**.

References

- ReadingCSVManualPage

  pandas.read_csv

  pandas.read_csv(filepath_or_buffer,*,sep=_NoDefault.no_default,delimiter=None,h eader='infer',names=_NoDefault.no_default,index_col=None,usecols=None,dtype= None,engine=None,converters=None,true_values=None,false_values=None,skipiniti alspace=False, skiprows=None,skipfooter=0,nrows=None, na_values=None, keep_default_na=True, na_filter=True,verbose=False,skip_blank_lines=True,parse_dates=None,

infer_datetime_format=_NoDefault.no_default,
keep_date_col=False,date_parser=_NoDefault.no_default,date_format=None,dayfir
st=False,cache_dates=True,iterator=False,chunksize=None,compression='infer',tho
usands=None,decimal='.',lineterminator=None,quotechar='"',quoting=0,doublequo
te=True, escapechar=None, comment=None,
encoding=None,encoding_errors='strict', dialect=None,
on_bad_lines='error',delim_whitespace=False, low_memory=True,
memory_map=False,float_precision=None,
storage_options=None,dtype_backend=_NoDefault.no_default)[source]

Readacomma-separatedvalues(csv)fileintoDataFrame.

Also supports optionally iterating or breaking of the file into

chunks.AdditionalhelpcanbefoundintheonlinedocsforIOTools.

Parameters:

filepath_or_bufferstr,pathobjectorfile-likeobject

Anyvalidstringpathisacceptable.ThestringcouldbeaURL.ValidURLschemesinclude
http, ftp, s3, gs, and file. For file URLs, a host is expected. A local
filecouldbe:file://localhost/path/to/table.csv.

Ifyouwanttopassinapathobject,pandasacceptsanyos.PathLike.

Byfile-
likeobject,werefertoobjectswitharead()method,suchasafilehandle(e.g.viabuiltinop
enfunction)orStringIO.

sepstr,default‘,’

Characterorregexpatterntotreatasthedelimiter.Ifsep=None,theCenginecannotau
tomaticallydetecttheseparator,buttthePythonparsingenginecan,meaning the
latter will be used and automatically detect the separator
fromonlythefirstvalidrowofthefilebyPython’sbuiltinsniffertool,csv.Sniffer.Inadditi
on, separators longer than 1 character and different from '\s+' will
beinterpreted as regular expressions and will also force the use of the
Pythonparsing engine. Note that regex delimiters are prone to ignoring quoted
data.Regexexample:'\r\t'.

delimiterstr,optionalAlias

forsep.

headerint,Sequenceofint,‘infer’orNone,default‘infer’

Row number(s) containing column labels and marking the start of the
data(zero-
indexed).Defaultbehavioristoinferthecolumnnames:ifnonamesarepassedthebeh
aviorisidenticaltoheader=0andcolumnnamesareinferred

from the first line of the file, if column names are passed explicitly to namesthen the behavior is identical to header=None. Explicitly pass header=0 to beable to replace existing names. The header can be a list of integers that specifyrow locations for a MultiIndex on the columns e.g. [0, 1, 3]. Intervening rowsthat are not specified will be skipped (e.g. 2 in this example is skipped). Notethat this parameter ignores commented lines and empty lines ifskip_blank_lines=True,soheader=0denotesthefirstlineofdataratherthanthefirstlineofthefile.

namesSequenceofHashable,optional

Sequence of column labels to apply. If the file contains a header row, then youshould explicitly pass header=0 to override the column names. Duplicates in thislistarenotallowed.

index_colHashable,SequenceofHashableorFalse,optional

Column(s) to use as row label(s), denoted either by column labels or columnindices.IfasequenceoflabelsorindicesisgivenMultiIndexwillbeformedforther owlabels.

Note:index_col=Falsecanbeusedtoforcepandastonotusethefirstcolumnasthe index, e.g., when you have a malformed file with delimiters at the end ofeachline.

usecolslistof Hashableor Callable, optional

Subset of columns to select, denoted either by column labels or column indices.If list-like, all elements must either be positional (i.e. integer indices into thedocumentcolumns)orstringsthatcorrespondtocolumnnamesprovidedeither by the user in names or inferred from the document header row(s). Ifnames are given, the document header row(s) are not taken into account. Forexample, a valid list-like usecols parameter would be [0, 1, 2] or ['foo', 'bar','baz']. Element order is ignored, so usecols=[0, 1] is the same as [1, 0]. Toinstantiate a DataFrame from data with element order preserved usepd.read_csv(data,usecols=['foo','bar'])[['foo','bar']]forcolumnsin['foo','bar'] orderorpd.read_csv(data,usecols=['foo','bar'])[['bar','foo']]for['bar','foo']order.

If callable, the callable function will be evaluated against the column names,returningnameswherethecallablefunctionevaluatestoTrue.Anexampleofava lidcallableargumentwouldbelambdax:x.upper()in['AAA','BBB','DDD']. Usingthis parameter results inmuchfaster parsingtime andlower memoryusage.

dtypedtypeordictof{Hashabledtype},optional

Datatype(s)toapplytoeitherthewholedatasetorindividualcolumns.E.g.,{'a':np.float64,'b': np.int32,'c':'Int64'}Usestrorobjecttogetherwithsuitable

na_valuessettingstopreserveandnotinterpretdtype.Ifconvertersarespecified,theyw
illbeappliedINSTEADofdtypeconversion.

Newinversion1.5.0:Supportfordefaultdictwasadded.Specifyadefaultdictasinput
where the default determines the dtype of the columns which are
notexplicitlylisted.

engine{‘c’,‘python’,‘pyarrow’},optional

Parserenginetouse.TheCandpyarrowenginesarefaster,whilethepythonengine is
currently more feature-complete. Multithreading is currently
onlysupportedbythepyarrowengine.

New in version 1.4.0: The ‘pyarrow’ engine was added as an
experimentalengine, and some features are unsupported, or may not work
correctly, withthisengine.

convertersdictof{HashableCallable},optional

Functions for converting values in specified columns. Keys can either be
columnlabelsorcolumnindices.

true_valueslist,optional

ValuestoconsiderasTrueinadditiontocase-

insensitivevariantsof‘True’.false_valueslist,optional

ValuestoconsiderasFalseinadditiontocase-

insensitivevariantsof‘False’.skipinitialspacebool,defaultFalse

Skipspacesafterdelimiter.

skiprowsint,listofintorCallable,optional

Linenumberstoskip(0-indexed)ornumberoflinestoskip(int)atthestartofthefile.

If callable, the callable function will be evaluated against the row
indices,returning True if the row should be skipped and False otherwise. An
example ofavalidcallableargumentwouldbelambdax:xin[0,2].

skipfooterint,default0

Numberoflinesatbottomoffiletoskip(Unsupportedwithengine='c').nrowsint,opti

onal

Number of rows of file to read. Useful for reading pieces of large

files.na_valuesHashable,IterableofHashableordictof{HashableIterable},optional

AdditionalstringstorecognizeasNA/NaN.Ifdictpassed,specificper-columnNAvalues.Bydefaultthefollowingvaluesareinterpretedas NaN:"",“#N/A”,“#N/A N/A”,“#NA”,“-1.#IND”,“-1.#QNAN”,“-NaN”,“-nan”,“1.#IND”,“1.#QNAN”,“<NA>”, “N/A”,“NA”,“NULL”,“NaN”,“None”,“n/a”,“nan”,“null“.keep_default_nabool,def

aultTrue

Whether or not to include the default NaN values when parsing the data.Dependingonwhetherna_valuesispassedin,thebehaviorisasfollows:

Ifkeep_default_naisTrue,andna_valuesarespecified,na_valuesisappendedtothed efaultNaNvaluesusedforparsing.

Ifkeep_default_naisTrue,andna_valuesarenotspecified,onlythedefaultNaNvaluesareuse dforparsing.

Ifkeep_default_naisFalse,andna_valuesarespecified,onlytheNaNvaluesspecifiedna _valuesareusedforparsing.

Ifkeep_default_naisFalse,andna_valuesarenotspecified,nostringswillbeparsedasN aN.

Notethatifna_filterispassedinasFalse,thekeep_default_naandna_valuesparameterswillb eignored.

na_filterbool,defaultTrue

Detectmissingvaluemarkers(emptystringsandthevalueofna_values).Indatawithout any NA values, passing na_filter=False can improve the performance ofreadingalargefile.

verbosebool,defaultFalse

Indicate number of NA values placed in non-numeric

columns.skip_blank_linesbool,defaultTrue

IfTrue,skipoverblanklinesratherthaninterpretingasNaNvalues.

parse_datesbool,listofHashable,listoflistsordictof{Hashablelist},defaultFalse

Thebehaviorisasfollows:

bool.IfTrue->tryparsingtheindex.

listofintornames.e.g.If[1,2,3]->tryparsingcolumns1,2,
3eachasaseparatedatecolumn.

listoflist.e.g.If[[1,3]]->combinecolumns1and3andparseasasingledatecolumn.

dict,e.g.{'foo':[1,3]}->parsecolumns1,3asdateandcallresult‘foo’

Ifacolumnorindexcannotberepresentedasanarrayofdatetime,saybecauseof an unparsable value or a mixture of timezones, the column or index will bereturned unaltered as an object data type. For non-standard datetime parsing,useto_datetime()afterread_csv().

Note:Afast-pathexistsforiso8601-

formatteddates.infer_datetime_formatbool,defaultFalse

If True and parse_dates is enabled, pandas will attempt to infer the format ofthedatetimestringsinthecolumns,andifitcanbeinferred,switchtoafastermethodo fparsingthem.Insomecasesthiscanincreasetheparsingspeedby5-10x.

Deprecatedsinceversion2.0.0:Astrictversionofthisargumentisnowthedefault,passin githasnoeffect.

keep_date_colbool,defaultFalse

If True and parse_dates specifies combining multiple columns then keep theoriginalcolumns.

date_parserCallable,optional

Function to use for converting a sequence of string columns to an array ofdatetimeinstances.Thedefaultusesdateutil.parser.parsertodotheconversion.pan daswilltrytocalldate_parserinthreedifferentways,advancingto the next if an exception occurs: 1) Pass one or more arrays (as defined byparse_dates)asarguments;2)concatenate(row-wise)thestringvaluesfromthecolumns defined by parse_dates into a single array and pass that; and 3) calldate_parser once for each row using one or more strings (corresponding to thecolumnsdefinedbyparse_dates)asarguments.

Deprecatedsinceversion2.0.0:Usedate_formatinstead,orreadinasobjectandthe napplyto_datetime()as-needed.

date_formatstrordictofcolumn->format,optional

Formattouseforparsingdateswhenusedinconjunctionwithparse_dates.Foranything more complex, please read in as object and then apply to_datetime()as-needed.

New in version

2.0.0.dayfirstbool,defaultFalse

DD/MMformatdates,internationalandEuropeanformat.

cache_datesbool,defaultTrue

If True, use a cache of unique, converted dates to apply the datetimeconversion.Mayproducesignificantspeed-upwhenparsingduplicatedatestrings,especiallyoneswithtimezoneoffsets.

iteratorbool,defaultFalse

ReturnTextFileReaderobjectforiterationorgettingchunkswithget_chunk().Changedi

nversion1.2:TextFileReaderisacontextmanager.

chunksizeint,optional

Number of lines to read from the file per chunk. Passing a value will cause thefunctiontoreturnaTextFileReaderobjectforiteration.SeetheIOToolsdocsformorei nformationoniteratorandchunksize.

Changedinversion1.2:TextFileReaderisacontextmanager.compressionstror

dict,default‘infer’

Foron-the-flydecompressionofon-diskdata.If‘infer’and‘filepath_or_buffer’ispath-like, then detect compression from the following extensions: ‘.gz’, ‘.bz2’,‘.zip’, ‘.xz’, ‘.zst’, ‘.tar’, ‘.tar.gz’, ‘.tar.xz’ or ‘.tar.bz2’ (otherwise no compression). Ifusing‘zip’or‘tar’,theZIPfilemustcontainonlyonedatafiletobereadin.SettoNoneforn odecompression.Canalsobeadictwithkey'method'settooneof {'zip', 'gzip', 'bz2', 'zstd', 'xz', 'tar'} and other key-value pairs are forwarded tozipfile.ZipFile, gzip.GzipFile, bz2.BZ2File, zstandard.ZstdDecompressor,lzma.LZMAFileortarfile.TarFile,respectively.Asanexa mple,thefollowingcouldbepassedforZstandarddecompressionusingacustomcompr essiondictionary:compression={'method':'zstd','dict_data':my_compression_dict}.

Newinversion1.5.0:Addedsupportfor.tarfiles.Changed in

version 1.4.0: Zstandard

support.thousandsstr(length1),optional

Characteractingas thethousandsseparatorinnumericalvalues.

decimalstr(length1),default‘.’

Charactertorecognizeasdecimalpoint(e.g.,use‘,’forEuropeandata).lineterminatorstr(leng

th1),optional

Characterusedtodenotealinebreak.OnlyvalidwithCparser.

quotecharstr(length1),optional

Character used to denote the start and end of a quoted item. Quoted items canincludethedelimiteranditwillbeignored.

quoting{0orcsv.QUOTE_MINIMAL,1orcsv.QUOTE_ALL,2orcsv.QUOTE_NONNUME RIC,3orcsv.QUOTE_NONE},defaultcsv.QUOTE_MINIMAL

Control field quoting behavior per csv.QUOTE_* constants. Default iscsv.QUOTE_MINIMAL(i.e.,0)whichimpliesthatonlyfieldscontainingspecialcharacters are quoted (e.g., characters defined in quotechar, delimiter, orlineterminator.

doublequotebool,defaultTrue

When quotechar is specified and quoting is not QUOTE_NONE, indicate whetherornottointerprettwoconsecutivequotecharelementsINSIDEafieldasasingleq uotecharelement.

escapecharstr (length 1), optionalCharacter

used to escape other

characters.commentstr(length1),optional

Character indicating that the remainder of line should not be parsed. If found atthe beginning of a line, the line will be ignored altogether. This parameter mustbe a single character. Like empty lines (as long as skip_blank_lines=True), fullycommented lines are ignored by the parameter header but not by skiprows. Forexample,ifcomment='#',parsing#empty\na,b,c\n1,2,3withheader=0willresultin'a, b,c'beingtreatedastheheader.

encodingstr,optional,default‘utf-8’

EncodingtouseforUTFwhenreading/writing(ex.'utf-8').ListofPythonstandardencodings.

Changedinversion1.2:WhenencodingisNone,errors='replace'ispassedtoopen(). Otherwise, errors='strict' is passed to open(). This behavior waspreviouslyonlythecaseforengine='python'.

Changedinversion1.3.0:encoding_errorsisanewargument.encodinghasnolongerani nfluenceonhowencodingerrorsarehandled.

encoding_errorsstr,optional,default‘strict’

Howencodingerrorsaretreated.Listofpossiblevalues.Newinv

ersion1.3.0.

dialectstrorcsv.Dialect,optional

If provided, this parameter will override values (default or not) for the followingparameters:delimiter,doublequote,escapechar,skipinitialspace,quotechar ,andquoting.Ifitisnecessarytooverridevalues,aParserWarningwillbeissued.Seecsv. Dialectdocumentationformoredetails.

on_bad_lines{‘error’,‘warn’,‘skip’}orCallable,default‘error’

Specifieswhattodouponencounteringabadline(alinewithtoomanyfields).Allowedvaluesar e:

'error',raiseanExceptionwhenabadlineisencountered.

'warn', raise a warning when a bad line is encountered and skip that

line.'skip', skip bad lines without raising or warning when they are

encountered.Newinversion1.3.0.

Newinversion1.4.0:

Callable, function with signature (bad_line: list[str]) -> list[str] | None that willprocess a single bad line. bad_line is a list of strings split by the sep. If thefunctionreturnsNone,thebadlinewillbeignored.Ifthefunctionreturnsanewlist of strings with more elements than expected, a ParserWarning will beemittedwhiledroppingextraelements.Onlysupportedwhenengine='python'

delim_whitespacebool,defaultFalse

Specifies whether or not whitespace (e.g. ' ' or '\t') will be used as the sepdelimiter.Equivalenttosettingsep='\s+'.IfthisoptionissettoTrue,nothingshouldb epassedinforthedelimiterparameter.

low_memorybool,defaultTrue

Internally process the file in chunks, resulting in lower memory use whileparsing,butpossiblymixedtypeinference.ToensurenomixedtypeseithersetFalse, or specify the type with the dtype parameter. Note that the entire file isread into a single DataFrame regardless, use the chunksize or iteratorparametertoreturnthedatainchunks.(OnlyvalidwithCparser).

memory_mapbool,defaultFalse

If a filepath is provided for filepath_or_buffer, map the file object directly ontomemory and access the data directly from there. Using this option can improveperformancebecausethereisnolongeranyI/Ooverhead.

float_precision{'high','legacy','round_trip'},optional

SpecifieswhichconvertertheCengineshoulduseforfloating-pointvalues.Theoptions are None or 'high' for the ordinary converter, 'legacy' for the originallowerprecisionpandasconverter,and'round_trip' fortheround-tripconverter.

Changed in version

1.2.storage_optionsdict,optio

nal

Extraoptionsthatmakesenseforaparticularstorageconnection,e.g.host,port, username, password, etc. For HTTP(S) URLs the key-value pairs areforwarded to urllib.request.Request as header options. For other URLs (e.g.starting with "s3://", and "gcs://") the key-value pairs are forwarded tofsspec.open. Please see fsspec and urllib for more details, and for moreexamplesonstorageoptionsreferhere.

Newinversion1.2.

dtype_backend{'numpy_nullable','pyarrow'},default'numpy_nullable'

Back-enddatatypeappliedtotheresultantDataFrame(stillexperimental).Behaviourisasfollows:

"numpy_nullable":returnsnullable-dtype-

backedDataFrame(default)."pyarrow": returns pyarrow-backed nullable

ArrowDtype DataFrame.Newinversion2.0.

Returns:

DataFrameorTextFileReader

Acomma-separatedvalues(csv)fileisreturnedastwo-dimensionaldatastructurewithlabeledaxes.

DataFrame.to_csv

WriteDataFrametoacomma-

separatedvalues(csv)file.read_table

Read general delimited file into

DataFrame.read_fwf

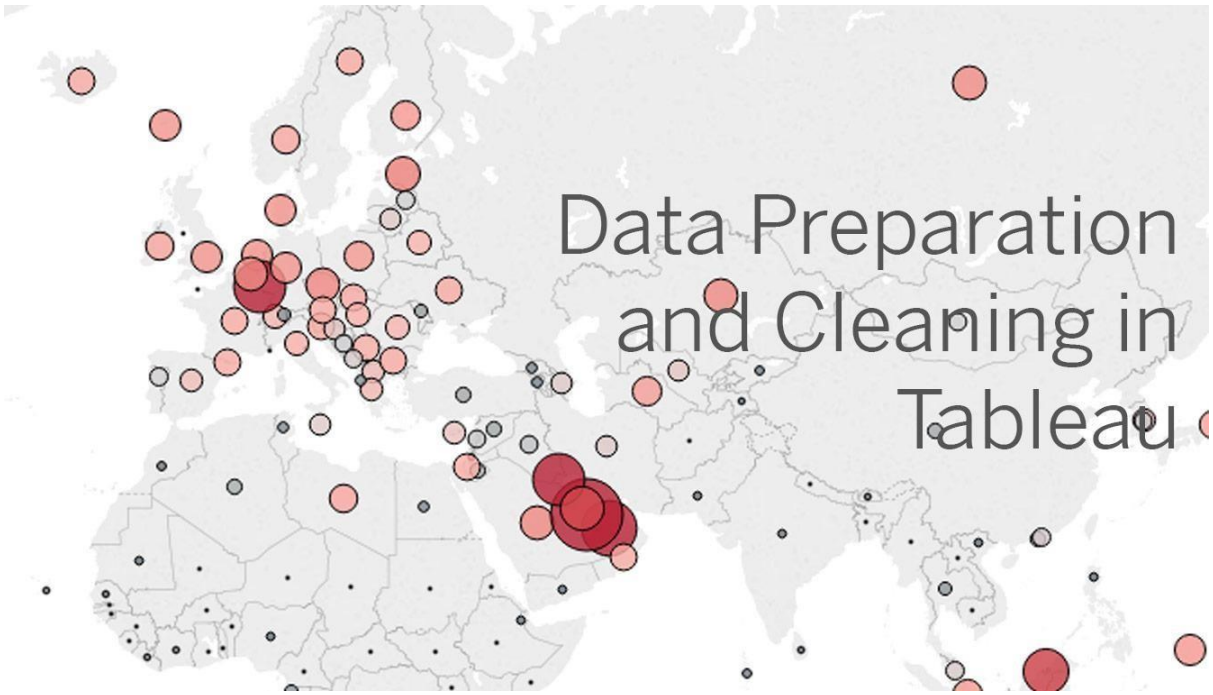
Readatableoffixed-widthformattedlinesintoDataFrame.

Examplespd.read_csv('d

ata.csv')

- **DataCleaning-Methodology/Approach**

GuideToDataCleaning:Definition,Benefits,Components,AndHow ToCleanYourData



When using data, most people agree that your insights and analysis are only as good as thedata you are using. Essentially, garbage data in is garbage analysis out. Data cleaning, alsoreferred to as data cleansing and data scrubbing, is one of the most important steps for yourorganizationif you want tocreateaculture aroundquality datadecision-making.

Inthisarticle we'llcover:

1. Whatisdatacleaning?
2. Datacleaningvs.datatransformation
3. Howtocleandata
4. Componentsofqualitydata
5. Advantagesandbenefitsofdatacleaning
6. Datacleaningtoolsandsoftware

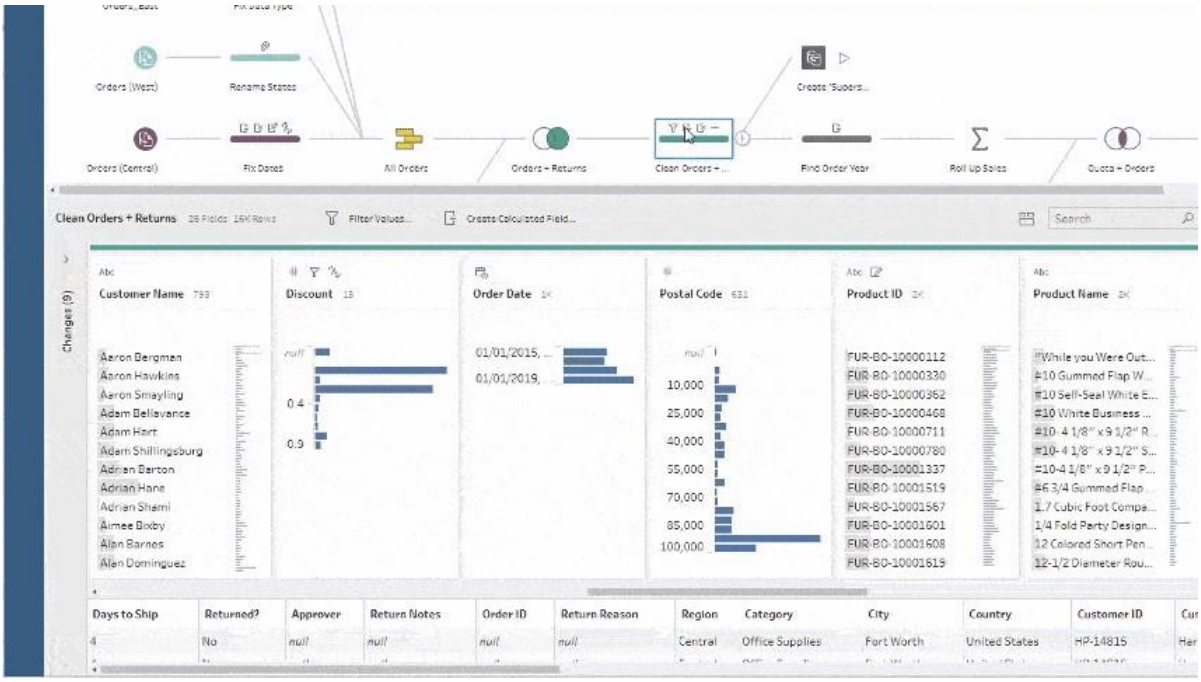
Whatisdatacleaning?

Datacleaningistheprocessoffixingorremovingincorrect,corrupted,incorrectlyformatted,duplicate, or incomplete data within a dataset. When combining multiple data sources, thereare many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomesand algorithms are unreliable, even though they may look correct. There is no one absoluteway to prescribe the exact steps in the data cleaning process because the processes will varyfrom dataset to dataset. But it is crucial to establish a template for your data cleaning processsoyou knowyou aredoing itthe rightway every time.

Whatisthedifferencebetweendatacleaninganddatatransformation?

Data cleaning is the process that removes data that does not belong in your dataset. Datatransformation is the process of converting data from one format or structure into another.Transformation processes can also be referred to as data wrangling, or data munging,transforming and mapping data from one "raw" data form into another format forwarehousingandanalyzing. Thisarticlefocuseson theprocesses ofcleaning thatdata.

Howtocleandata



While the techniques used for data cleaning may vary according to the types of data yourcompany stores, you can follow these basic steps to map out a framework for yourorganization.

Step1:Removeduplicateorirrelevantobservations

Remove unwanted observations from your dataset, including duplicate observations orirrelevant observations. Duplicate observations will happen most often during data collection.When you combine data sets from multiple places, scrape data, or receive data from clients ormultipledepartments,thereareopportunitiestocreateduplicatedata.De-duplicationisoneof the largest areas to be considered in this process. Irrelevant observations are when younotice observations that do not fit into the specific problem you are trying to analyze. Forexample, if you want to analyze data regarding millennial customers, but your datasetincludes older generations, you might remove those irrelevant observations. This can makeanalysis more efficient and minimize distraction from your primary target—as well ascreating a moremanageableandmore performantdataset.

Step2:Fixstructuralerrors

Structural errors are when you measure or transfer data and notice strange namingconventions, typos, or incorrect capitalization. These inconsistencies can cause mislabeledcategories or classes. For example, you may find “N/A” and “Not Applicable” both appear,butthey should be analyzed as the samecategory.

Step3:Filterunwantedoutliers

Often, there will be one-off observations where, at a glance, they do not appear to fit withinthe data you are analyzing. If you have a legitimate reason to remove an outlier, like improperdata-entry, doing so will help the performance of the data you are working with. However,sometimesitis the appearance ofan outlier thatwillprovea theory you are workingon. Remember: just because an outlier exists, doesn’t mean it is incorrect. This step is needed todetermine the validity of that number. If an outlier proves to be irrelevant for analysis or is amistake, consider removing it.

Step4:Handlemissingdata

You can’t ignore missing data because many algorithms will not accept missing values. Thereareacoupleofwaystodealwithmissingdata.Neitherisoptimal,butbothcanbeconsidered.

1. Asafirstoption,youcandropobservationsthathavemissingvalues,butdoingthiswilldroporlosein formation,sobemindfulofthisbeforeyouremoveit.
2. Asasecondoption,youcaninputmissingvaluesbasedonotherobservations;again,thereisan opportunity to lose integrity of the data because you may be operating from assumptionsandnotactualobservations.
3. Asathirdoption,youmightalterthewaythedataisusedtoeffectivelynavigatenullvalues.

Step5:ValidateandQA

At the end of the data cleaning process, you should be able to answer these questions as a partofbasicvalidation:

- Doesthedatamakesense?
- Doesthedatafollowtheappropriaterulesforitsfield?
- Doesitproveordisproveyourworkingtheory,orbringanyinsighttolight?
- Canyoufindtrendsinthedatatohelpyouformyournexttheory?
- Ifnot,isthatbecauseofadataqualityissue?

False conclusions because of incorrect or "dirty" data can inform poor business strategy anddecision-making. False conclusions can lead to an embarrassing moment in a reportingmeeting when you realize your data doesn't stand up to scrutiny. Before you get there, it isimportant to create a culture of quality data in your organization. To do this, you shoulddocumentthetoolsyou mightusetocreatethisculture and whatdataqualitymeanstoyou.

Componentsofqualitydata

Determiningthequalityofdatarequiresanexaminationofitscharacteristics,then weighingthose characteristics according to what is most important to your organization and theapplication(s)for which they willbe used.

5characteristicsofqualitydata

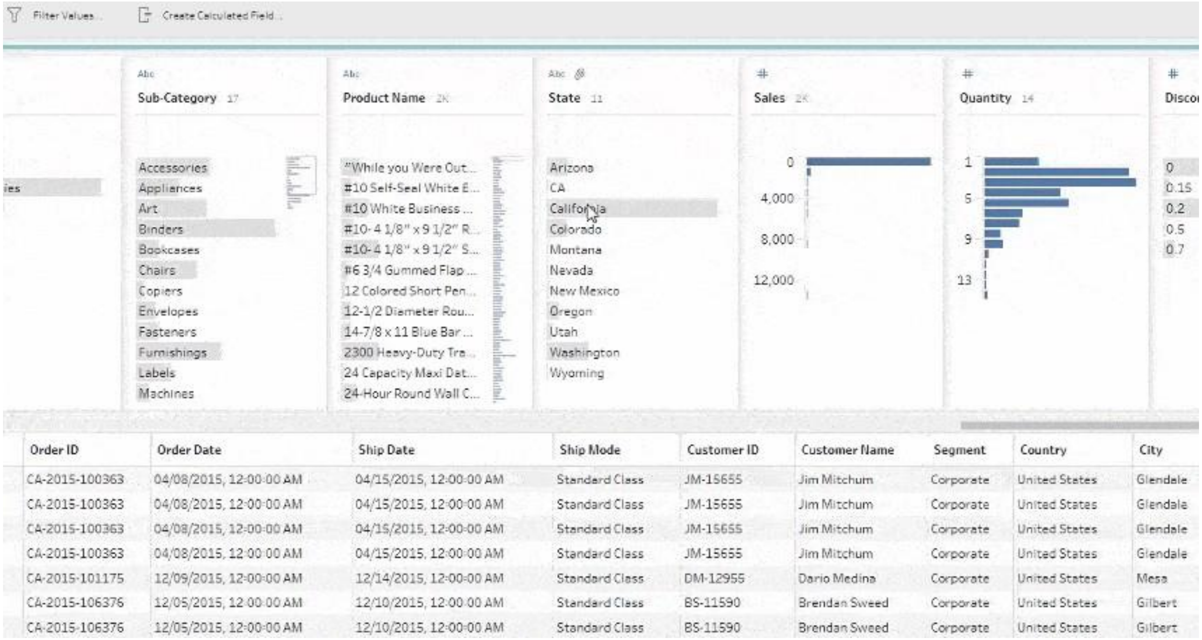
1. **Validity.**Thedegreetowhichyourdataconformstodefinedbusinessrulesorconstraints.
2. **Accuracy.**Ensureyourdataisclosetothetruevalues.
3. **Completeness.**Thedegreetowhichallrequireddataisknown.
4. **Consistency.**Ensureyourdataisconsistentwithinthesamedatasetand/oracrossmultipledatasets.
5. **Uniformity.**Thedegreetowhichthedataisspecifiedusingthesameunitofmeasure.

Advantagesandbenefitsofdatacleaning

Having clean data will ultimately increase overall productivity and allow for the highestquality information in yourdecision-making. Benefits include:

- Removaloferrorswhenmultiplesourcesofdataareatplay.
- Fewererrorsmakeforhappierclientsandless-frustratedemployees.
- Abilitytomapthedifferentfunctionsandwhatyourdataisintendedtodo.
- Monitoringerrorsandbetterreportingtoseewhereerrorsarecomingfrom,makingiteasiertofixincorrectorcorruptdataforfutureapplications.
- Usingtoolsfordatacleaningwillmakeformoreefficientbusinesspracticesandquickerdecision-making.

Datacleaningtoolsandsoftwareforefficiency



Software like Tableau Prep can help you drive a quality data culture by providing visual anddirect ways to combine and clean your data. Tableau Prep has two products: Tableau PrepBuilder for building your data flows and Tableau Prep Conductor for scheduling, monitoring,andmanagingflowsacrossyourorganization.Usingadatascrubbingtoolcansaveadata base administrator a significant amount of time by helping analysts or administrators starttheiranalysesfasterandhavemoreconfidenceinthedata.Understandingdataqualityandthe tools you need to create, manage, and transform data is an important step toward makingefficient and effective business decisions. This crucial process will further develop a dataculture in your organization. To see how Tableau Prep can impact your organization, readabout how marketing agency Tinuiti centralized 100-plus data sources in Tableau Prep andscaledtheir marketing analytics for 500clients.

- **DataCleaning-UsefulTools**

PythonicDataCleaningWithpandasandNumPy

Datascientistsspendalargeamountoftheirtimecleaningdatasetsandgettingthem down to a form with which they can work. In fact, a lot of data scientistsarguethattheinitialstepsofobtainingandcleaningdataconstitute80%ofthe job.

Therefore,ifyouarejuststeppingintothisfieldorplanningtostepintothisfield,itisimportanttobeabletodealwithmessydata,whetherthatmeansmissing values, inconsistent formatting, malformed records, or nonsensicaloutliers.

Inthistutorial,we'llleveragePython'spandasandNumPylibrariestocleandata.

We'llcoverthefollowing:

- Droppingunnecessarycolumnsina`DataFrame`
- Changingtheindexofa`DataFrame`
- Using`.str()`methodstocleancolumns
- Using the `DataFrame.applymap()` function to clean the entire dataset,element-wise
- Renamingcolumnstoamorerecognizablesetoflabels
- SkippingunnecessaryrowsinaCSVfileHer

earethedatasetsthatwewillbeusing:

- BL-Flickr-Images-Book.csv– ACSVfilecontaininginformationaboutbooksfromtheBritishLibrary
- university_towns.txt– Atextfilecontainingnamesofcollegetownsineveryusstate
- olympics.csv– ACSVfilesummarizingtheparticipationofallcountriesintheSummerandWinterOlympics

YoucandownloadthedatasetsfromRealPython'sGitHubrepositoryinordertofollowtheexampleshere.

ThistutorialassumesabasicunderstandingofthepandasandNumPylibraries,including Panda's workhorse `Series` and `DataFrame` objects, common methodsthatcanbeappliedtotheseobjects,andfamiliaritywithNumPy's`NaN`values.

Let'simporttherequiredmodulesandgetstarted!Pytho

n

```
>>>import pandasaspd
>>>import numpyas np
```

DroppingColumnsina`DataFrame`

Often, you'll find that not all the categories of data in a dataset are useful to you. Forexample,you mighthaveadatasetcontainingstudentinformation(name,grade,standard,parents'names,anda ddress)but wanttofocuson analyzingstudentgrades.

In this case, the address or parents' names categories are not important to you. Retainingthese unneeded categories will take up unnecessary space and potentially also bog downruntime.

pandasprovidesahandy wayofremovingunwanted columnsorrowsfrom a `DataFrame`withthe `drop()`function.Let's lookata simpleexample where we drop anumberof columns from a`DataFrame`.

First, let's create a `DataFrame` out of the CSV file 'BL-Flickr-Images-Book.csv'. In theexamples below, we pass a relative path to `pd.read_csv`, meaning that all of the datasets arein afoldernamed`Datasets`inour currentworking directory:

Python

```
>>>df= pd.read_csv('Datasets/BL-Flickr-Images-Book.csv')
>>>df.head()
```

```
    Identifier             Edition Statement      Place of Publication\
0         206                           NaN                    London
1         216                           NaN  London; Virtue & Yorston
2         218                           NaN                    London
3         472                           NaN                    London
4         480  A new edition, revised, etc.                    London

  Date of Publication             Publisher\0
         1879 [1878]       S. Tinsley & Co.
1               1868           Virtue & Co.
2               1869Bradbury, Evans & Co.
3               1851          James Darling
4               1857   Wertheim & Macintosh

                                          Title     Author\
0                 Walter Forbes. [A novel.] By A. A      A.A.
1  Allfor Greed. [A novel. Thededication signed...  A., A. A.
2  Lovethe Avenger. By the author of "All for Gr...  A., A. A.
3  WelshSketches, chiefly ecclesiastical, to the...  A., E. S.
4  [TheWorld in which I live, and my place in it...  A., E. S.

                               ContributorsCorporate Author\
0                            FORBES, Walter.               NaN
1  BLAZE DE BURY, Marie Pauline Rose- Baroness               NaN
2  BLAZE DE BURY, Marie Pauline Rose- Baroness               NaN
3                 Appleyard, Ernest Silvanus.               NaN
4                         BROOME, John Henry.               NaN
```

```
   Corporate Contributors Former ownerEngraver Issuance type\
0                     NaN          NaN       NaN   monographic
1                     NaN          NaN       NaN   monographic
2                     NaN          NaN       NaN   monographic
3                     NaN          NaN       NaN   monographic
4                     NaN          NaN       NaN   monographic

                                            Flickr URL\
0  http://www.flickr.com/photos/britishlibrary/ta...
1  http://www.flickr.com/photos/britishlibrary/ta...
2  http://www.flickr.com/photos/britishlibrary/ta...
3  http://www.flickr.com/photos/britishlibrary/ta...
4  http://www.flickr.com/photos/britishlibrary/ta...

                           Shelfmarks
0    British Library HMNTS 12641.b.30.
1    British Library HMNTS 12626.cc.2.
2    BritishLibrary HMNTS 12625.dd.1.
3    British Library HMNTS 10369.bbb.15.
4    British Library HMNTS 9007.d.28.
```

Whenwelookatthefirstfiveentries using the`head()`method, we cansee thata handful ofcolumns provide ancillary information that would be helpful to the library but isn't verydescriptive of the books themselves: `Edition Statement`, `Corporate Author`, `CorporateContributors`,`Former owner`,`Engraver`,`Issuance type`and`Shelfmarks`.

Wecan dropthese columnsin thefollowing way:

Python

```
>>>to_drop = ['Edition Statement',
...            'CorporateAuthor',
...            'Corporate Contributors',
...            'Formerowner',
...            'Engraver',
...            'Contributors',
...            'Issuancetype',
...            'Shelfmarks']


>>>df.drop(to_drop,inplace=True,axis=1)
```

Above, we defined a list that contains the names of all the columns we want to drop. Next,wecallthe`drop()`functionon our object,passing in the`inplace`parameter as`True`and

the `axis` parameter as `1`. This tells pandas that we want the changes to be made directly inourobjectand that itshouldlook forthevaluesto bedropped in thecolumnsoftheobject.

WhenweinspecttheDataFrameagain,we'llseethatthe unwantedcolumns have beenremoved:

Python

```
>>>df.head()
   Identifier      Place of Publication Date of Publication\
0         206                    London         1879 [1878]
1         216  London; Virtue & Yorston                1868
2         218                    London                1869
3         472                    London                1851
4         480                    London                1857

               Publisher                                                Title\
0       S. Tinsley & Co.                  Walter Forbes. [A novel.] By A. A
1           Virtue& Co.  All for Greed. [A novel. The dedication signed...
2  Bradbury,Evans & Co.  Love the Avenger. By the author of "All for Gr...
3          JamesDarling  Welsh Sketches, chiefly ecclesiastical, to the...
4   Wertheim& Macintosh  [The World in which I live, and my place in it...

      Author                                          Flickr URL
0      A. A.http://www.flickr.com/photos/britishlibrary/ta...
1  A., A. A.http://www.flickr.com/photos/britishlibrary/ta...
2  A., A. A.http://www.flickr.com/photos/britishlibrary/ta...
3  A., E. S.http://www.flickr.com/photos/britishlibrary/ta...
4  A., E. S.http://www.flickr.com/photos/britishlibrary/ta...
```

Alternatively,wecouldalso removethecolumns by passing themto thecolumnsparameterdirectly instead of separately specifying the labels to be removed and the axis where pandasshouldlook for thelabels:

Python

```
>>>df.drop(columns=to_drop,inplace=True)
```

Thissyntaxismore intuitiveandreadable.Whatwe'retryingtodohereisdirectlyapparent.

If you know in advance which columns you'd like to retain, another option is to pass them tothe `usecols`argumentof `pd.read_csv`.

ChangingtheIndexof a **`DataFrame`**

ApandasIndexextendsthefunctionality of NumPyarrays to allow for moreversatileslicingand labeling. In many cases, it is helpful to use a uniquely valued identifying field of the dataasits index.

For example, in the dataset used in the previous section, it can be expected that when alibrariansearchesforarecord, they may inputthe uniqueidentifier(values in the `Identifier`column) for a book:

Python

```
>>>
df['Identifier'].is_uniqueTrue
```

Let'sreplacetheexistingindexwiththiscolumnusing `set_index`:

Python

```
>>>df= df.set_index('Identifier')
>>>df.head()
               Place of Publication Date of Publication\
206                          London         1879 [1878]
216         London; Virtue & Yorston                1868
218                          London                1869
472                          London                1851
480                          London                1857

                       Publisher\
206              S. Tinsley & Co.
216                  Virtue & Co.
218         Bradbury, Evans & Co.
472                 James Darling
480          Wertheim& Macintosh

                                                      Title     Author  \
206                          Walter Forbes. [A novel.] By A. A       A.A.
216         All for Greed. [A novel. The dedication signed...   A.,A. A.
218         Love the Avenger. By the author of "All for Gr...   A.,A. A.
472         Welsh Sketches, chiefly ecclesiastical, to the...   A.,E. S.
480         [The World in which I live, and my place in it...   A.,E. S.

                                                 Flickr
206         URLhttp://www.flickr.com/photos/britishlibrary/ta
            ...
216         http://www.flickr.com/photos/britishlibrary/ta...
```

```
218         http://www.flickr.com/photos/britishlibrary/ta...
472         http://www.flickr.com/photos/britishlibrary/ta...
480         http://www.flickr.com/photos/britishlibrary/ta...
```

Technical Detail: Unlikeprimary keysinSQL,a pandas `Index`doesn't makeanyguaranteeof being unique, although many indexing and merging operations will notice a speedup inruntime if itis.
We can access each record in a straightforward way with `loc[]`. Although `loc[]` may nothave all that intuitive of a name, it allows us to do *label-based indexing*, which is the labelingofa row or record withoutregard toits position:

Python

```
>>>df.loc[206]
Place of Publication                                          London
Date of Publication                                     1879 [1878]
Publisher                                             S. Tinsley & Co.
Title                                 Walter Forbes. [A novel.] By A. A
Author                                                         A.A.
Flickr URL
                          http://www.flickr.com/photos/britishlibrary/ta...
Name:206, dtype: object
```

In other words, 206 is the first label of the index. To access it by *position*, we couldused`df.iloc[0]`,which doesposition-based indexing.

Technical Detail: `.loc[]`istechnicallyaclass instanceand has some special syntax thatdoesn'tconform exactlyto most plain-vanilla Pythoninstancemethods.
Previously, our index was a RangeIndex: integers starting from `0`, analogous to Python'sbuilt-in `range`. By passing a column name to `set_index`, we have changed the index to thevaluesin`Identifier`.

You may have noticed that we reassigned the variable to the object returned by the methodwith `df = df.set_index(...)`. This is because, by default, the method returns a modifiedcopy of our object and does not make the changes directly to the object. We can avoid this bysettingthe `inplace`parameter:

Python

```
df.set_index('Identifier',inplace=True)
```

TidyingupFieldsin theData

Sofar,wehaveremovedunnecessarycolumnsand changedtheindexof our`DataFrame`tosomething more sensible. In this section, we will clean specific columns and get them to auniform format to get a better understanding of the dataset and enforce consistency.
Inparticular,wewill becleaning`Dateof Publication`and`Place of`Publication.

Uponinspection,all of thedatatypes are currentlythe`object`dtype, which is roughlyanalogousto `str`innative Python.

It encapsulates any field that can't be neatly fit as numerical or categorical data. This makessensesincewe'reworking withdatathat isinitiallya bunchofmessystrings:

Python

```
>>>
df.get_dtype_counts()obje
ct        6
```

One field where it makes sense to enforce a numeric value is the date of publication so thatwecan do calculations down the road:

Python

```
>>> df.loc[1905:, 'Date of
Publication'].head(10)Identifier
1905           1888
1929    1839, 38-54
2836        [1897?]
2854           1865
2956        1860-63
2957           1873
3017           1866
3131           1899
4598           1814
4884           1820
Name: Dateof Publication, dtype: object
```

A particular book can have only one date of publication. Therefore, we need to do thefollowing:

- Removetheextradatesinsquarebrackets,whereverpresent:1879[1878]
- Convertdaterangestotheir"startdate",whereverpresent:1860-63;1839,38-54
- CompletelyremovethedateswearenotcertainaboutandreplacethemwithNumPy's NaN:[1897?]
- Convertthestringnanto NumPy'sNaNvalue

Synthesizing these patterns, we can actually take advantage of a single regular expression toextract the publication year:

Python

```
regex = r'^(\d{4})'
```

The regular expression above is meant to find any four digits at the beginning of a string,which suffices for our case. The above is a *raw string* (meaning that a backslash is no longeranescapecharacter),whichis standard practice withregular expressions.

The `\d`representsanydigit,and`{4}`repeats this rule fourtimes. The`^`charactermatchesthestartofa string,and the parenthesesdenoteacapturinggroup, which signalstopandas that

wewantto extractthatpartof the regex. (Wewant^toavoid cases where[starts off thestring.)

Let'sseewhathappenswhenwerunthisregexacrossourdataset:

Python

```
>>>extr =df['Date ofPublication'].str.extract(r'^(\d{4})', expand=False)
>>>
extr.head()Iden
tifier
206    1879
216    1868
218    1869
472    1851
480    1857
Name: Dateof Publication, dtype: object
```

Further Reading: Not familiar with regex? You can inspect the expression above atregex101.com and learn all about regular expressions with Regular Expressions: Regexes inPython.
Technically,thiscolumnstillhas`object`dtype, butwe caneasily getitsnumerical versionwith`pd.to_numeric`:

Python

```
>>>df['Date ofPublication']= pd.to_numeric(extr)
>>> df['Date of
Publication'].dtypedtype('float64')
```

This results in about one in every ten values being missing, which is a small price to pay fornowbeing able todocomputations ontheremaining validvalues:

Python

```
>>> df['Date of Publication'].isnull().sum() /
len(df)0.11717147339205986
```

Great!That'sdone!

Combining`str`MethodswithNumPytoCleanColumns

Above, you may have noticed the use of `df['Date of Publication'].str`. Thisattribute is a way to access speedy string operations in pandas that largelymimic operations on native Python strings or compiled regular expressions,suchas`.split()`,`.replace()`,and`.capitalize()`.

To clean the `PlaceofPublication` field, we can combine pandas `str` methods with NumPy's `np.where` function, which is basically a vectorized form of Excel's `IF()` macro. It has the following syntax:

Python

```
>>>np.where(condition,then,else)
```

Here, `condition` is either an array-like object or a Boolean mask. `then` is the value to be used if `condition` evaluates to `True`, and `else` is the value to be used otherwise.

Essentially, `.where()` takes each element in the object used for `condition`, checks whether that particular element evaluates to `True` in the context of the condition, and returns an `ndarray` containing `then` or `else`, depending on which applies.

It can be nested into a compound if-then statement, allowing us to compute values based on multiple conditions:

Python

```
>>> np.where(condition1,
        x1,np.where(condition2,x2,
            np.where(condition3,x3, ...)))
```

We'll be making use of these two functions to clean `PlaceofPublication` since this column has string objects. Here are the contents of the column:

Python

```
>>>df['Place  of Publication'].head(10)
Identifier
206                                  London
216                London; Virtue & Yorston
218                                  London
472                                  London
480                                  London
481                                  London
519                                  London
667     pp.40. G. Bryan & Co: Oxford, 1898
874                                 London]
1143                                 London
Name:Place of Publication, dtype: object
```

We see that for some rows, the place of publication is surrounded by otherunnecessaryinformation.Ifweweretolookatmorevalues,wewouldseethatthis is the case for only some rows that have their place of publication as‘London’or‘Oxford’.

Let’stakealookattwospecificentries:

Python

```
>>>df.loc[4157862]
Place of Publication                                  Newcastle-upon-Tyne
Date of Publication                                                  1867
Publisher                                                      T.Fordyce
Title                  Local Records; or, Historical Register of rema...
Author                                                         T.Fordyce
Flickr URL             http://www.flickr.com/photos/britishlibrary/ta...
Name:4157862, dtype: object

>>>df.loc[4159587]
Place of Publication                                  Newcastle upon Tyne
Date of Publication                                                  1834
Publisher                                                Mackenzie & Dent
Title                  Anhistorical, topographical and descriptive v...
Author                                               E. (Eneas)Mackenzie
Flickr URL             http://www.flickr.com/photos/britishlibrary/ta...
Name:4159587, dtype: object
```

Thesetwobookswerepublishedinthesameplace,butonehashyphensinthenameofth eplacewhiletheotherdoesnot.

Tocleanthiscolumninonesweep,wecanuse`str.contains()`togetaBooleanmask.

Wecleanthecolumnasfollows:

Python

```
>>>pub =df['Place of Publication']
>>>london=pub.str.contains('London')
>>>
london[:5]Iden
tifier
206    True
```

```
216    True
218    True
472    True
480    True
Name:Place of Publication, dtype: bool


>>>oxford=pub.str.contains('Oxford')
```

We combine them with `np.where`:

Python

```
df['Placeof Publication'] = np.where(london, 'London',
                                     np.where(oxford,'Oxford',
                                              pub.str.replace('-',' ')))


>>> df['Place of
Publication'].head()Identifier
206    London
216    London
218    London
472    London
480    London
Name:Place of Publication, dtype: object
```

Here, the `np.where` function is called in a nested structure, with `condition` being a `Series` of Booleans obtained with `str.contains()`. The `contains()` method works similarly to the built-in `in` keyword used to find the occurrence of an entity in an iterable (or substring in a string).

The replacement to be used is a string representing our desired place of publication. We also replace hyphens with a space with `str.replace()` and reassign to the column in our `DataFrame`.

Although there is more dirty data in this dataset, we will discuss only these two columns for now.

Let's have a look at the first five entries, which look a lot crisper than when we started out:

Python

```
>>>df.head()
```

```
          Place of Publication Date of Publication              Publisher  \
206                     London                1879        S. Tinsley & Co.
216                     London                1868           Virtue & Co.
218                     London                1869  Bradbury, Evans & Co.
472                     London                1851          James Darling
480                     London                1857   Wertheim & Macintosh

                                                      Title    Author\
206                        Walter Forbes. [A novel.] By A. A        AA
216         All for Greed. [A novel. The dedication signed...   A.A A.
218         Love the Avenger. By the author of “All for Gr...   A.A A.
472         Welsh Sketches, chiefly ecclesiastical, to the...   E.S A.
480         [The World in which I live, and my place in it...   E.S A.

                                                 FlickrURL
206         http://www.flickr.com/photos/britishlibrary/ta...
216         http://www.flickr.com/photos/britishlibrary/ta...
218         http://www.flickr.com/photos/britishlibrary/ta...
472         http://www.flickr.com/photos/britishlibrary/ta...
480         http://www.flickr.com/photos/britishlibrary/ta...
```

Note: At this point, `Place of Publication` would be a good candidate forconversion to a `Categorical` dtype, because we can encode the fairly smallunique set of cities with integers. (*The memory usage of a Categorical isproportionaltothenumberofcategoriesplusthelengthofthedata;anobjectdtypeisaconstanttimesthelengthofthedata.*)

CleaningtheEntireDatasetUsingthe`applymap`Function
Incertainsituations,youwillseethatthe“dirt”isnotlocalizedtoonecolumn butismorespreadout.

There are some instances where it would be helpful to apply a customizedfunctiontoeachcellorelementofaDataFrame.pandas`.applymap()`metho dissimilar to the in-built `map()` function and simply applies a function to all theelementsina`DataFrame`.

Let’slookatanexample.Wewillcreatea`DataFrame`outofthe “university_towns.txt”

file:Shell

```
$ head_
Datasets/univerisity_towns.txtAlabama
[edit]
Auburn (Auburn University)[1]
Florence (University of North
Alabama)Jacksonville(JacksonvilleStateUniversity
)[2]
Livingston (University of WestAlabama)[2]
Montevallo (University of Montevallo)[2]
Troy (Troy University)[2]
Tuscaloosa (University of Alabama,Stillman College, Shelton State)[3][4]
Tuskegee (Tuskegee
University)[5]Alaska[edit]
```

We see that we have periodic state names followed by the university towns inthatstate:`StateATownA1TownA2StateBTownB1TownB2 .....`. Ifwelookattheway statenamesarewritteninthefile,we'llseethatallofthemhavethe"[edit]" substringinthem.

Wecantakeadvantageofthispatternbycreatinga*listof(state,city)tuples*andwrappingthatlistina`DataFrame`:

Python

```
>>> university_towns= []
>>>with open('Datasets/university_towns.txt') as file:
...     forlinein file:
...         if'[edit]'in line:
...             #Remember this `state` until the next is found
...             state= line
...         else:
...             #Otherwise, we have acity; keep `state` as last-seen
...             university_towns.append((state, line))

>>>university_towns[:5]
[('Alabama[edit]\n', 'Auburn (Auburn
 University)[1]\n'),('Alabama[edit]\n', 'Florence (University of North
 Alabama)\n'),('Alabama[edit]\n', 'Jacksonville (Jacksonville State
 University)[2]\n'),('Alabama[edit]\n', 'Livingston (University of West
 Alabama)[2]\n'),('Alabama[edit]\n','Montevallo (University of
 Montevallo)[2]\n')]
```

We can wrap this list in a DataFrame and set the columns as "State" and"RegionName".pandaswilltakeeachelementinthelistandset`State`totheleftvalueand`RegionName`totherightvalue.

TheresultingDataFramelookslikethis:

Python

```
>>>towns_df= pd.DataFrame(university_towns,
...                         columns=['State', 'RegionName'])


>>>towns_df.head()
 State                                          RegionName
0  Alabama[edit]\n                    Auburn (Auburn University)[1]\n
1  Alabama[edit]\n           Florence (University of North Alabama)\n
2  Alabama[edit]\nJacksonville (Jacksonville State University)[2]\n
3  Alabama[edit]\n       Livingston (University of West Alabama)[2]\n
4  Alabama[edit]\n         Montevallo (University of Montevallo)[2]\n
```

Whilewecouldhavecleanedthesestringsintheforloopabove,pandasmakesit easy. We only need the state name and the town name and can removeeverything else. While we could use pandas' `.str()` methods again here, wecould also use `applymap()` to map a Python callable to each element of theDataFrame.

Wehave beenusingthe term*element*, butwhat exactlydo wemean by it? Considerthefollowing"toy"DataFrame:

Python

```
        0           1
0    Mock     Dataset
1  Python     pandas
2    Real     Python
3   NumPy     Clean
```

Inthisexample,eachcell('Mock','Dataset','Python','pandas',etc.)isanelement. Therefore, `applymap()` will apply a function to each of theseindependently.Let'sdefinethatfunction:

Python

```
>>>defget_citystate(item):
...     if' ('in item:
...         returnitem[:item.find('(')]
...     elif'['in item:
...         returnitem[:item.find('[')]
```

```
...     else:
...         returnitem
```

pandas'.`applymap()`onlytakesoneparameter,whichisthefunction(callable)thatshou ldbeappliedtoeachelement:

Python

```
>>>towns_df=  towns_df.applymap(get_citystate)
```

First,wedefineaPythonfunctionthattakesanelementfromthe`DataFrame`asitsparam eter.Insidethefunction,checksareperformedtodeterminewhetherthere'sa`(`or`[`int heelementornot.

Depending on the check, values are returned accordingly by the function.Finally,the`applymap()`functioniscalledonourobject.NowtheDataFrameism uchneater:

Python

```
>>>towns_df.head()State
              RegionName
0  Alabama        Auburn
1  Alabama      Florence
2  AlabamaJacksonville
3  Alabama    Livingston
4  Alabama    Montevallo
```

The`applymap()`methodtookeachelementfromtheDataFrame,passedittothefunctio n, and the original value was replaced by the returned value. It's thatsimple!

TechnicalDetail:Whileitisaconvenientandversatilemethod,.`applymap`canhaves ignificantruntimeforlargerdatasets,becauseitmapsaPythoncallabletoeachindivi dualelement.Insomecases,itcanbemoreefficientto do*vectorized*operationsthatutilizeCythonorNumPY(which,inturn,makescallsi nC)underthehood.

RenamingColumnsandSkippingRows

Often,thedatasetsyou'llworkwithwillhaveeithercolumnnamesthatarenoteasy to understand, or unimportant information in the first few and/or lastrows,suchasdefinitionsofthetermsinthedataset,orfootnotes.

Inthatcase,we'dwanttorenamecolumnsandskipcertainrowssothatwecan drilldowntonecessaryinformationwithcorrectandsensiblelabels.

Todemonstratehowwecangoaboutdoingthis,let'sfirsttakeaglanceattheinitialfive rowsofthe"olympics.csv"dataset:

Shell

```
$ head_-n_5_
Datasets/olympics.csv0,1,2,3,4,5,6,7,
8,9,10,11,12,13,14,15
,? Summer,01 !,02 !,03 !,Total,? Winter,01 !,02 !,03 !,Total,? Games,01 !,02 !,03
!,Combined total
Afghanistan (AFG),13,0,0,2,2,0,0,0,0,0,13,0,0,2,2
Algeria (ALG),12,5,2,8,15,3,0,0,0,0,15,5,2,8,15
Argentina (ARG),23,18,24,28,70,18,0,0,0,0,41,18,24,28,70
```

Now,we'llreaditintoapandasDataFrame:

Python

```
>>>olympics_df=pd.read_csv('Datasets/olympics.csv')
>>>olympics_df.head()
                   0         1     2     3     4      5         6     7     8\
0                NaN  ? Summer   01!   02!   03!  Total  ? Winter  01 !  02 !
1  Afghanistan (AFG)        13     0     0     2      2         0     0     0
2      Algeria (ALG)        12     5     2     8     15         3     0     0
3    Argentina (ARG)        23    18    24    28     70        18     0     0
4      Armenia (ARM)         5     1     2     9     12         6     0     0


      9     10       11    12    13    14             15
003 !Total? Games01 !02 !03 !Combined total1            0
      0      13      0      0     2     2
2     0      0       15     5     2     8             15
3     0      0       41    18    24    28             70
4     0      0       11     1     2     9             12
```

Thisismessyindeed!Thecolumnsarethestringformofintegersindexedat0.Therow whichshouldhavebeenourheader(i.e.theonetobeusedtosetthecolumnnames)is at`olympics_df.iloc[0]`.ThishappenedbecauseourCSVfilestartswith0,1,2,…,15.

Also,ifweweretogotothesourceofthisdataset,we'dseethat`NaN`aboveshouldreally besomethinglike"Country",`?Summer`issupposedtorepresent"SummerGames",`01 !`shouldbe"Gold",andsoon.

Therefore,weneedtodotwothings:

- Skiponerowandsettheheaderasthefirst(0-indexed)row
- Renamethecolumns

Wecanskiprowsandsettheheaderwhilereadingthe CSVfilebypassingsomeparametersto the`read_csv()`function.

Thisfunctiontakes*alot*ofoptionalparameters,butinthiscaseweonlyneedone(`header`)toremovethe0throw:

Python

```
>>>olympics_df= pd.read_csv('Datasets/olympics.csv',header=1)
>>>olympics_df.head()
          Unnamed: 0? Summer01 !02 !03 !Total? Winter\
0        Afghanistan (AFG)        13     0     0     2      2         0
1            Algeria (ALG)        12     5     2     8     15         3
2          Argentina (ARG)        23    18    24    28     70        18
3            Armenia (ARM)         5     1     2     9     12         6
4  Australasia (ANZ) [ANZ]         2     3     4     5     12         0

   01 !.102 !.103 !.1Total.1? Games01 !.202 !.203 !.2\
0       0       0       0        0       13       0       0       2
1       0       0       0        0       15       5       2       8
2       0       0       0        0       41      18      24      28
3       0       0       0        0       11       1       2       9
4       0       0       0        0        2       3       4       5

   Combined
total0          2
1              15
2              70
3              12
4              12
```

Wenowhavethecorrectrowsetastheheaderandallunnecessaryrowsremove d.Takenoteofhowpandashaschangedthenameofthecolumncontainingthe nameofthecountriesfrom`NaN`to`Unnamed:0`.

Torenamethecolumns,wewillmakeuseofaDataFrame's`rename()`method,whichallo wsyoutorelabelanaxisbasedona*mapping*(inthiscase,a`dict`).

Let'sstartbydefiningadictionarythatmapscurrentcolumnnames(askeys)tomoreusableones(thedictionary'svalues):

Python

```
>>>new_names ={'Unnamed:0': 'Country',
...               '?Summer':'Summer Olympics',
...               '01!':'Gold',
...               '02!':'Silver',
...               '03!':'Bronze',
...               '?Winter':'Winter Olympics',
...               '01!.1': 'Gold.1',
...               '02!.1': 'Silver.1',
...               '03!.1': 'Bronze.1',
...               '?Games': '# Games',
...               '01!.2': 'Gold.2',
...               '02!.2': 'Silver.2',
...               '03!.2': 'Bronze.2'}
```

Wecallthe`rename()`functiononourobject:

Python

```
>>>olympics_df.rename(columns=new_names,inplace=True)
```

Setting*inplace*to`True`specifiesthatourchangesbemadedirectlytotheobject. Let'sseeifthischecksout:

Python

```
>>> olympics_df.head()
                   Country  Summer Olympics  Gold  Silver  Bronze  Total  \
0        Afghanistan (AFG)               13     0       0       2      2
1            Algeria (ALG)               12     5       2       8     15
2          Argentina (ARG)               23    18      24      28     70
3            Armenia (ARM)                5     1       2       9     12
4    Australasia(ANZ)[ANZ]                2     3       4       5     12

   Winter OlympicsGold.1Silver.1Bronze.1Total.1# GamesGold.2\00        0
                 0       0         0         13       0
1                3       0         0         0        0       15       5
2               18       0         0         0        0       41      18
3                6       0         0         0        0       11       1
```

```
4                0       0         0         0        0        2        3


   Silver.2Bronze.2Combined total0  0
          2         2
1         2         8              15
2        24        28              70
3         2         9              12
4         4         5              12
```

PythonDataCleaning:RecapandResources

In this tutorial, you learned how you can drop unnecessary information from adatasetusingthe`drop()`function,aswellashowtosetanindexforyourdatasetsothatitemsinitcanbereferencedeasily.

Moreover,youlearnedhowtoclean`object`fieldswiththe`.str()`accessorandhow to clean the entire dataset using the `applymap()` method. Lastly, weexploredhowtoskiprowsinaCSVfileandrenamecolumnsusing the`rename()`method.

Knowing about data cleaning is very important, because it is a big part of datascience.Younowhaveabasicunderstandingofhowpandasand NumPycanbeleveragedtocleandatasets!

Tips

- JupytercanberunonanyOS,butforWindows,hassl `Anaconda` provides themoste-freewayofinstallingJupyter.
- Youmaywanttouse `loc` whendoingdatacleaning.
- Non numeric fields are much harder to deal with effectively as compared tonumericfields.

ExpectedOutcome

You should have a pandas dataframe with the relevant parameters ready to goforanalysisinajupyternotebook.Somethinglikethiswilldo:

	cook	eating_out	employment	ethnic_food	exercise	fruit_day	income	on_off_campus	pay_meal_out	sports	veggies_day
0	2.0	3	3.0	1	1.0	5	5.0	1.0	2	1.0	5
1	3.0	2	2.0	4	1.0	4	4.0	1.0	4	1.0	4
2	1.0	2	3.0	5	2.0	5	6.0	2.0	3	2.0	5
3	2.0	2	3.0	5	3.0	4	6.0	1.0	2	2.0	3
4	1.0	2	2.0	4	1.0	4	6.0	1.0	4	1.0	4

DataExplorationandVisualisation

Now that we have our data, we need to understand it. A good way to do this isbyvisualisingthedataviagraphs.Graphshelpusquicklygetasenseofthedata,and are a much more user-friendly way of understanding data as compared toreadingthousandsofrowsofdata!

AgoodgraphtolookatdistributedgroupsisaBoxplot.Itcantellusatglancewhere the population is concentrated, and how the outliers compare to theaverageobjectinthegroup.

Requirements

- Tovisualisethe `food_choices` datasetyoucreatedintheprevious task,plotaboxplotonthedataframe.
- Refertothe`codebook_food`filetomakesenseoftheBoxplot.Notedown the general trends followed by the population that wassurveyedaspartofthestudy.
- You should be looking at things like how much each person exerciseson average (these people will need gyms), whether they arevegetarian/nonvegetarian(weshouldaccountforboth),andincome(thi saffectsthelifestyleofthepersonsignificantly.)

BringitOn!

Boxplotsarejustonewaytovisualiseyourdata.Canyouthinkofanyothervisualisationmethod sthatmightaidyouranalysis?
References

- UnderstandingBoxPlots
- BoxplotsusingSeaborn

seaborn.boxplot

seaborn.**boxplot**(*data=None, *, x=None, y=None, hue=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, fill=True, dodge='auto', width=0.8, gap=0, whis=1.5, linecolor='auto', linewidth=None, fliersize=None, hue_norm=None, native_scale=False, log_scale=None, formatter=None, legend='auto',ax=None, **kwargs*)

Drawaboxplottoshowdistributionswithrespecttocategories.

Aboxplot(orbox-and-whiskerplot)showsthedistributionofquantitativedatain a way that facilitates comparisons between variables or across levels of acategorical variable. The box shows the quartiles of the dataset while the whiskersextend to show the rest of the distribution, except for points that are determinedtobe"outliers" using amethodthatisafunctionof theinter-quartile range.

See the tutorial for more

information.Note

By default, this function treats one of the variables as categorical and draws dataat ordinal positions (0, 1, … n) on the relevant axis. As of version 0.13.0, this canbedisabledbysetting`native_scale=True`.

Parameters:

data*DataFrame,Series,dict,array,orlistofarrays*
Datasetfor plotting. If`x`and`y`areabsent, thisisinterpreted aswide-form.Otherwise itis expected to belong-form.

x,y, hue*namesofvariablesin`data`orvector data*
Inputsforplottinglong-formdata.Seeexamplesforinterpretation.

order,hue_order*listsofstrings*
Order to plot the categorical levels in; otherwise the levels are inferred from thedataobjects.

orient*"v"|"h"|"x"|"y"*
Orientation of the plot (vertical or horizontal). This is usually inferred based on thetypeoftheinput variables,butit canbeused to resolveambiguitywhen both`x`and`y`arenumericorwhenplottingwide-form data.

Changedinversionv0.13.0:Added'x'/'y'asoptions,equivalentto'v'/'h'.

color*matplotlibcolor*
Singlecolorfortheelementsintheplot.

palette*palettename,list,ordict*
Colors to use for the different levels of the `hue` variable. Should be somethingthat can be interpreted by **`color_palette()`**, or a dictionary mapping hue levelstomatplotlibcolors.

saturation*float*
Proportion of the original saturation to draw fill colors in. Large patches oftenlook better with desaturated colors, but set this to `1` if you want the colors toperfectly match theinput values.

fill*bool*
IfTrue,useasolidpatch.Otherwise,drawaslineart.

Newinversionv0.13.0.

dodge*"auto"orbool*

When hue mapping is used, whether elements should be narrowed and shiftedalongtheorient axistoeliminateoverlap.If`"auto"`,set to`True`when the orientvariableis crossed withthe categoricalvariable or`False`otherwise.

Changedinversion0.13.0:Added`"auto"`modeasanewdefault.

width*float*

Width allotted to each element on the orient axis. When `native_scale=True`, it isrelativetotheminimumdistancebetweentwovalues inthenativescale.

gap*float*

Shrinkontheorientaxisby thisfactortoaddagapbetweendodgedelements.

Newinversion0.13.0.

whis*float or pair of floats*Paramater that controls whisker length. If scalar, whiskers are drawn to thefarthest datapoint within *whis * IQR* from the nearest hinge. If a tuple, it isinterpreted aspercentiles thatwhiskersrepresent.

linecolor*color*

Color touseforline elements,when`fill`isTrue.

Newinversionv0.13.0.

linewidth*float*

Widthofthelinesthatframetheplotelements.

fliersize*float*

Sizeofthemarkersusedtoindicateoutlierobservations.

hue_norm*tuple* or***matplotlib.colors.Normalize****object*

Normalizationin dataunitsforcolormapappliedtothe`hue`variable whenitisnumeric.Notrelevantif`hue`iscategorical.

Newinversionv0.12.0.

log_scale*bool ornumber, orpair ofbools ornumbers*

Setaxisscale(s)tolog.Asinglevaluesetsthedataaxisforanynumericaxes intheplot.Apairofvaluessetseachaxisindependently.Numericvaluesare

interpretedasthedesiredbase(default10).When`None`or`False`,seaborndeferstothe existingAxesscale.

Newinversionv0.13.0.

When True, numeric or datetime values on the categorical axis will maintain theiroriginal scalingrather thanbeingconverted to fixedindices.

Newinversionv0.13.0.

Functionforconvertingcategoricaldataintostrings.Affectsbothgroupingandticklabel s.

Newinversionv0.13.0.

How to draw the legend. If “brief”, numeric `hue` and `size` variables will berepresented with a sample of evenly spaced values. If “full”, every group will getan entry in the legend. If “auto”, choose between brief or full representationbased on number of levels. If `False`, no legend data is added and no legend isdrawn.

Newinversionv0.13.0.

Axesobjecttodrawtheplotonto, otherwiseuses thecurrentAxes.
Other keyword arguments are passed throughto**`matplotlib.axes.Axes.boxplot()`**.
Returns:
ax*matplotlibAxes*
ReturnstheAxesobjectwiththeplotdrawnontoit.

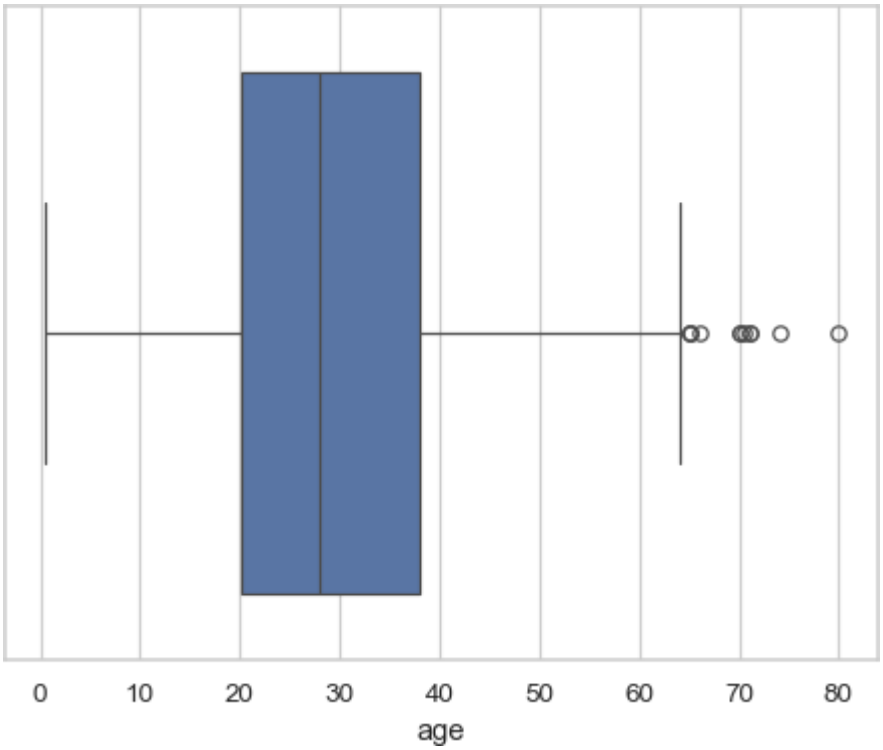
Seealso

`violinplot`
Acombinationofboxplotandkerneldensity estimation.
`stripplot`
Ascatterplotwhereonevariableiscategorical.Canbeusedinconjunctionwithotherplot s to showeach observation.
`swarmplot`
A categorical scatterplot where the points do not overlap. Can be used with otherplotsto showeachobservation.
`catplot`
Combineacategoricalplotwitha **`FacetGrid`**.

Examples

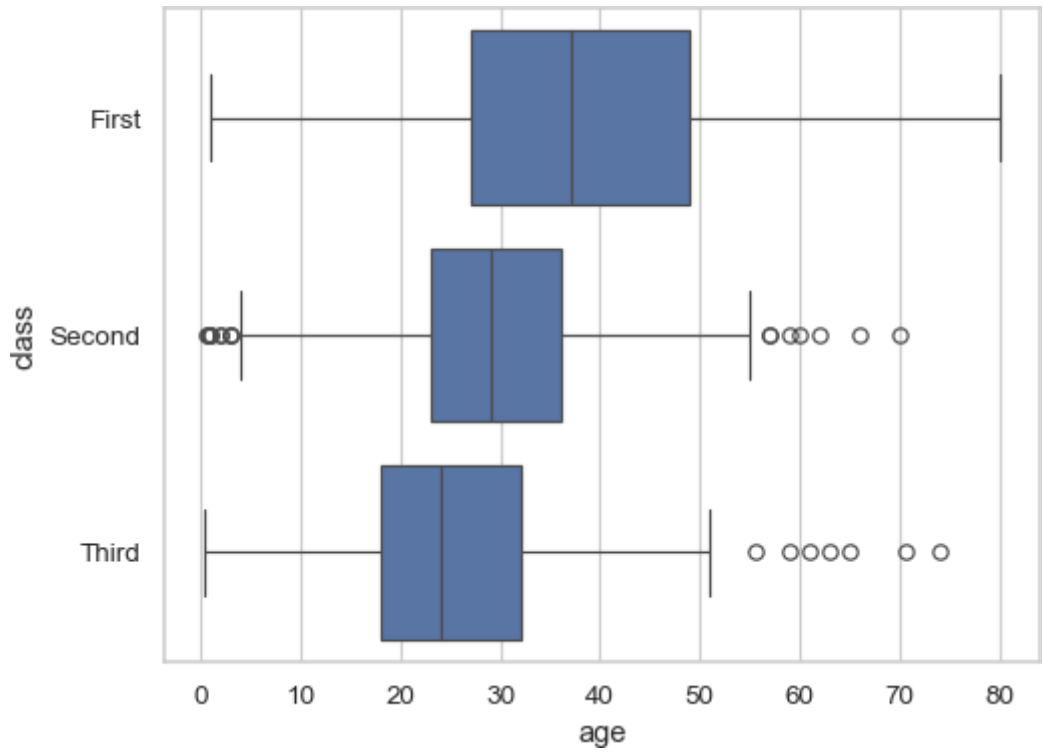
Draw a single horizontal boxplot, assigning the data directly tothecoordinate variable:

```
sns.boxplot(x=titanic["age"])
```



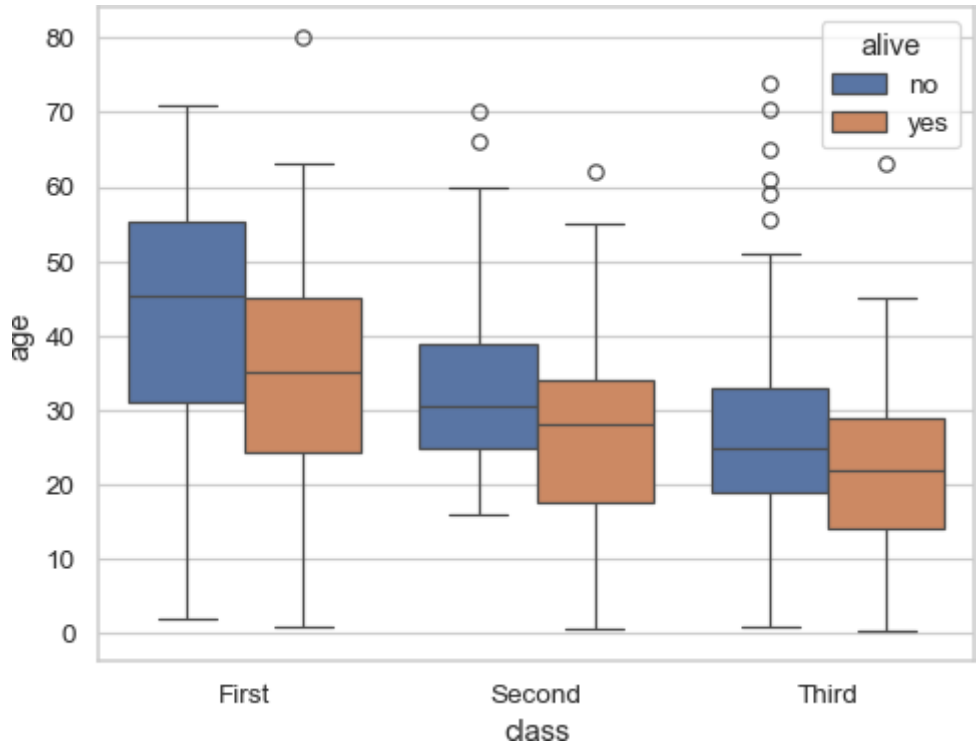
Groupbyacategoricalvariable,referencingcolumnsinadataframe:

```
sns.boxplot(data=titanic,x="age",y="class")
```



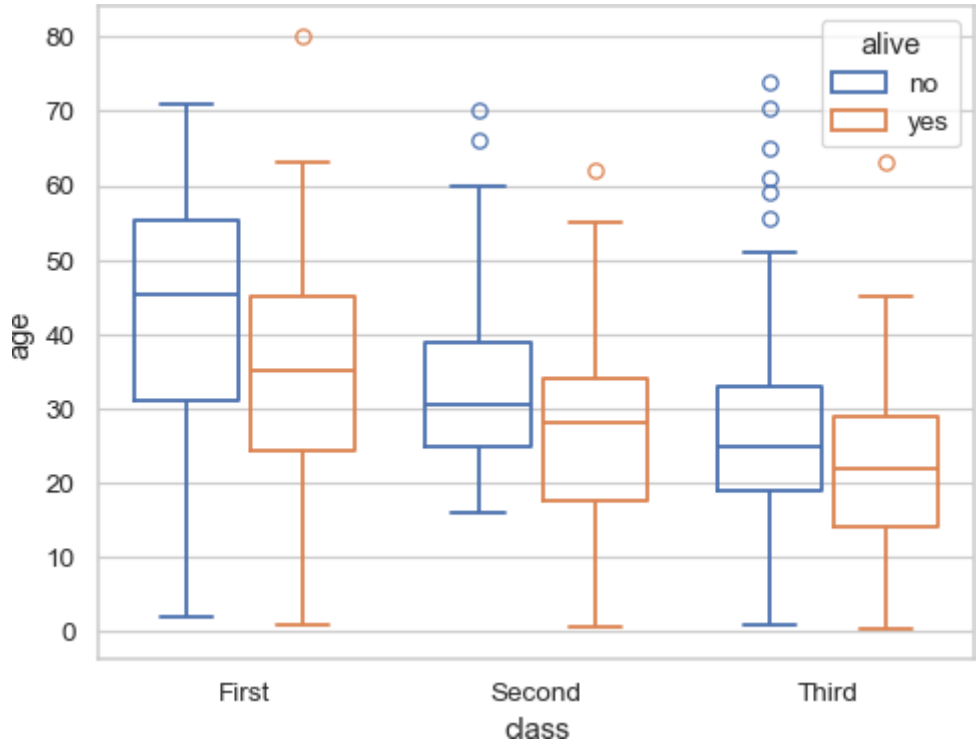
Drawaverticalboxplotwithnestedgrouping bytwovariables:

```
sns.boxplot(data=titanic,x="class",y="age",hue="alive")
```



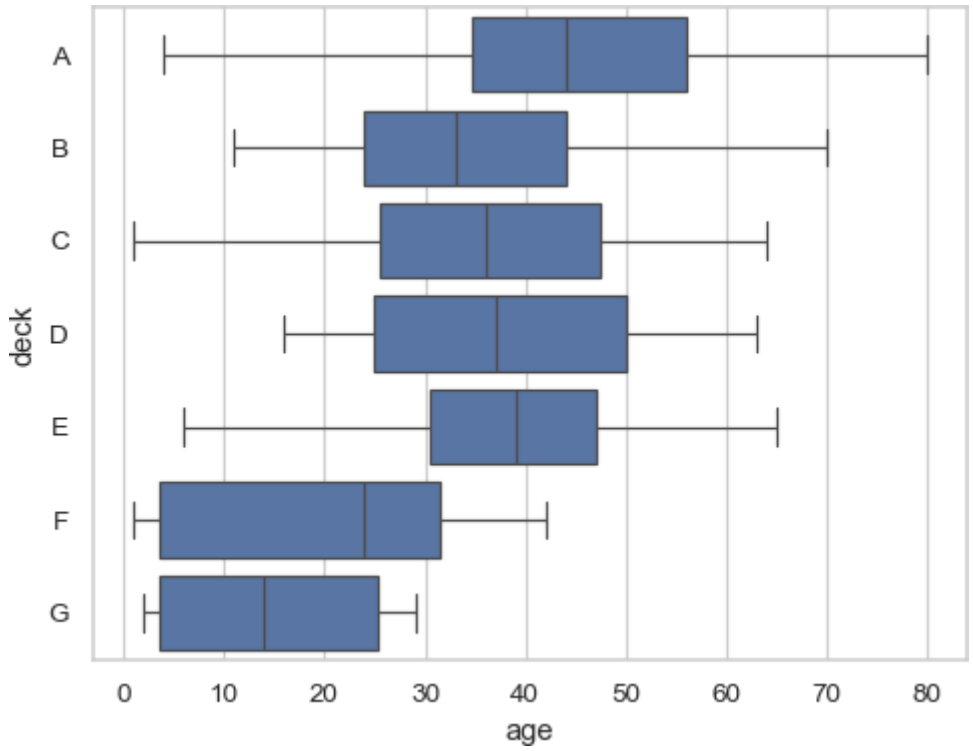
Drawtheboxesaslineartandadda smallgapbetweenthem:

```
sns.boxplot(data=titanic, x="class", y="age", hue="alive",
fill=False,gap=.1)
```



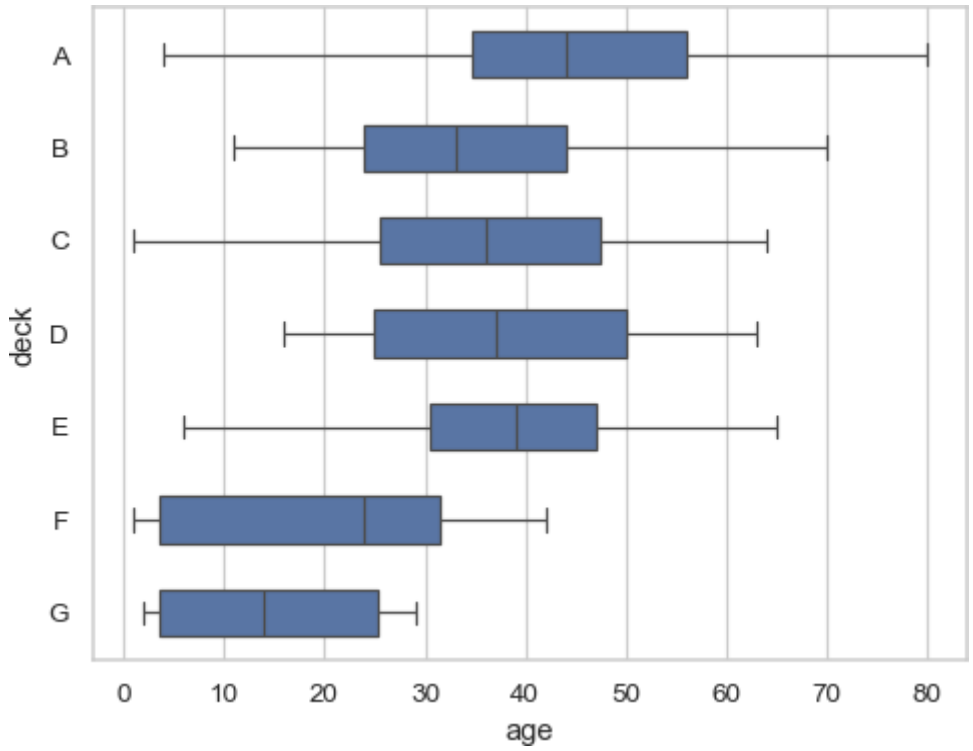
Coverthefull rangeof thedatawiththewhiskers:

```
sns.boxplot(data=titanic,x="age",y="deck",whis=(0,100))
```



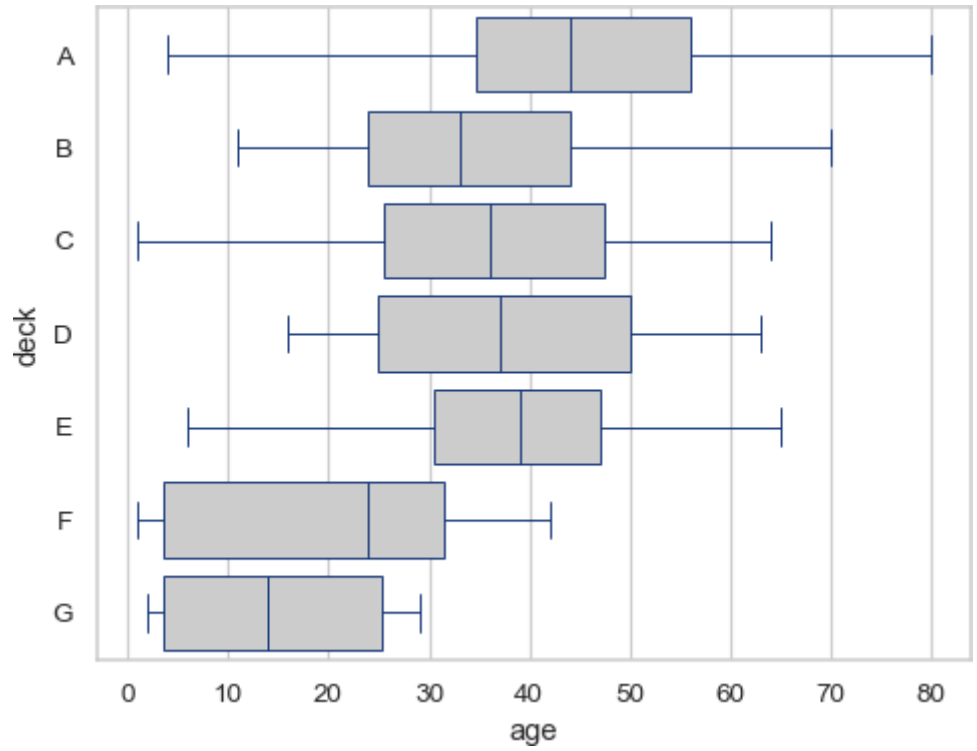
Drawnarrowerboxes:

```
sns.boxplot(data=titanic,x="age",y="deck",width=.5)
```



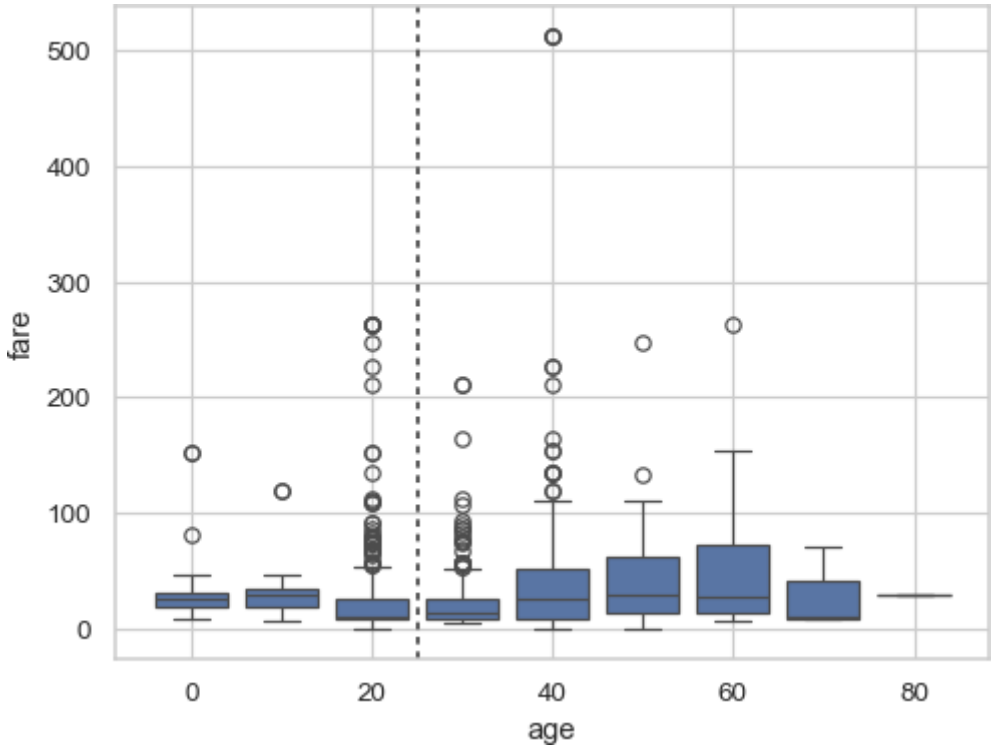
Modifythecolorand widthofallthelineartists:

```
sns.boxplot(data=titanic, x="age", y="deck", color=".8",
linecolor="#137",linewidth=.75)
```



Groupbyanumericvariableandpreserveitsnativescaling:

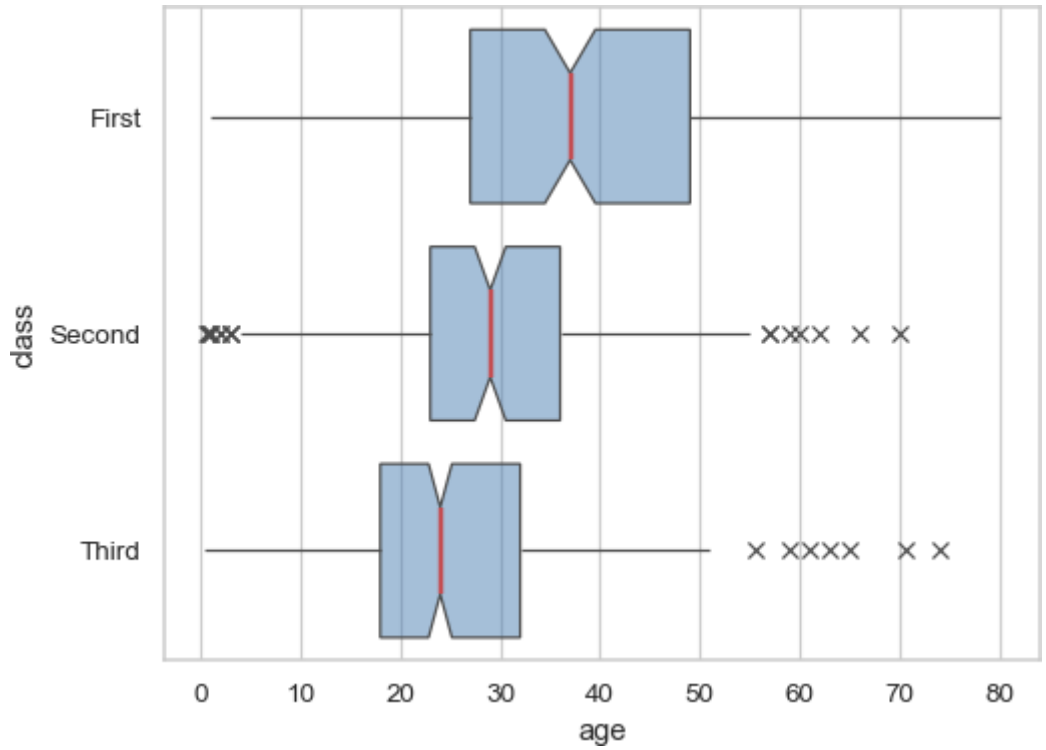
```
ax = sns.boxplot(x=titanic["age"].round(-1),
y=titanic["fare"],native_scale=True)
ax.axvline(25,color=".3",dashes=(2,2))
```



Customizetheplotusingparametersoftheunderlyingmatplotlibfunction:

```
sns.boxplot(
```

```
    data=titanic, x="age",
    y="class",notch=True,
    showcaps=False,flierprops={"marker":
    "x"},boxprops={"facecolor": (.3, .5, .7,
    .5)},medianprops={"color":"r","linewidth":2
    },
)
```



- BoxplotsusingPandas

pandas.DataFrame.boxplot

DataFrame.boxplot(*column=None, by=None, ax=None, fontsize=None, rot=0, grid=True,figsize=None,layout=None,return_type=None,backend=None,kwargs*)[source]**

MakeaboxplotfromDataFramecolumns.

Makeabox-and-whiskerplotfromDataFramecolumns,optionallygroupedby some other columns. A box plot is a method for graphically depictinggroupsofnumericaldatathroughtheirquartiles.Theboxextendsfromthe Q1 to Q3 quartile values of the data, with a line at the median (Q2). Thewhiskers extend from the edges of box to show the range of the data. Bydefault, they extend no more than *1.5 * IQR (IQR = Q3 - Q1)* from the edges ofthe box, ending at the farthest data point within that interval. Outliers areplottedasseparatedots.

ForfurtherdetailsseeWikipedia'sentryforboxplot.

Parameters:

column*strorlistofstr,optional*

Column name or list of names, or vector. Can be any valid inputto`pandas.DataFrame.groupby()`.
by*strorarray-like,optional*

Column in the DataFrame to `pandas.DataFrame.groupby()`. One box-plot willbedonepervalueof columnsin*by*.
ax*objectofclassmatplotlib.axes.Axes,optional*

Thematplotlibaxestobeusedbyboxplot.
fontsize*floatorstr*

Ticklabelfontsizeinpointsorasastring(e.g.,*large*).
rot*float,default0*

The rotation angle of labels (in degrees) with respect to the screen coordinatesystem.
grid*bool,defaultTrue*

SettingthistoTruewillshowthegrid.
figsize*Atuple(width,height)ininches*

Thesizeofthefiguretocreateinmatplotlib.
layout*tuple(rows,columns),optional*

For example, (3, 5) will display the subplots using 3 rows and 5 columns,startingfromthetop-left.
return_type*{'axes','dict','both'}orNone,default'axes'*

Thekindofobjecttoreturn.Thedefaultis`axes`.

- 'axes'returnsthematplotlibaxestheboxplotisdrawnon.
- 'dict'returnsadictionarywhosevaluesarethematplotlibLinesofthe boxplot.
- 'both'returnsanamedtuplewiththeaxesand dict.
- whengroupingwith`by`,aSeries mappingcolumnsto`return_type`isreturned.

  If`return_type`is*None*,a NumPy array of axes withthesameshapeas`layout`is returned.

backend*str,defaultNone*

Backendtouseinsteadofthebackendspecifiedinthe option`plotting.backend`.Forinstance,'matplotlib'.Alternatively,tospecify the`plotting.backend`forthewhole session,set`pd.options.plotting.backend`.

****kwargs**

All other plotting keyword arguments to be passedto**matplotlib.pyplot.boxplot()**.

Returns:

result

SeeNotes.

Seealso

pandas.Series.plot.hist

Makeahistogram.

matplotlib.pyplot.boxplot

Matplotlibequivalentplot.

Notes

Thereturntypedependsonthe*return_type*parameter:

- 'axes':objectofclassmatplotlib.axes.Axes
- 'dict':dictofmatplotlib.lines.Line2Dobjects
- 'both':anamedtuplewithstructure(ax,lines)

Fordatagroupedwith `by`, returnaSeriesofthe aboveora numpyarray:

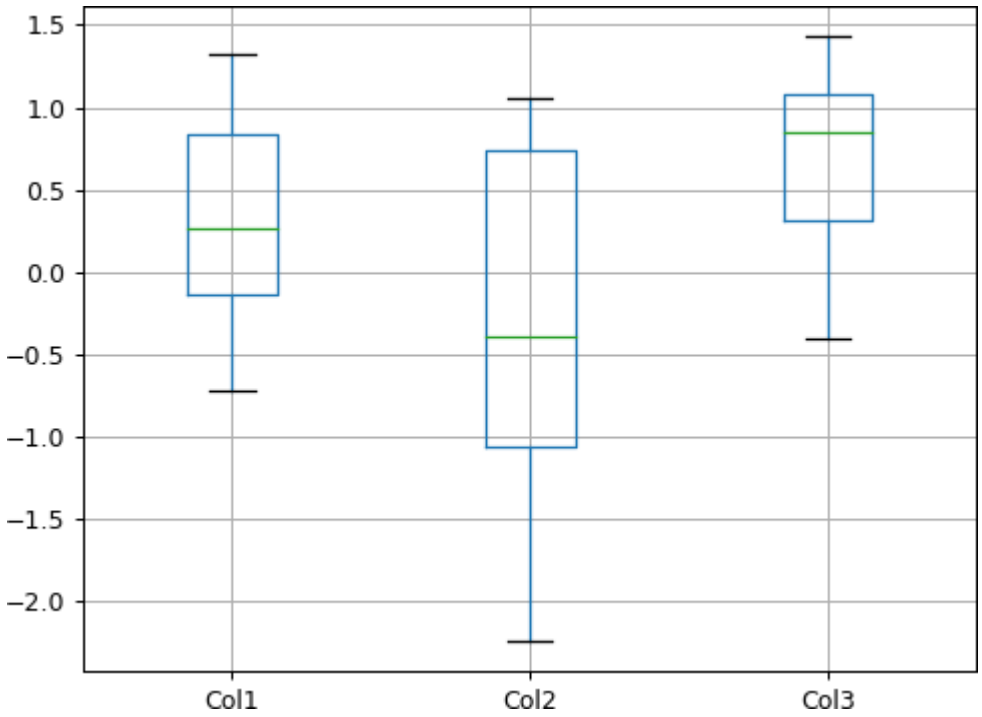
- **Series**
- **array**(for`return_type= None`)

Use`return_type='dict'`when youwantto tweaktheappearance ofthelines after plotting. In this case a dict containing the Lines making up theboxes,caps, fliers,medians, andwhiskers isreturned.

Examples

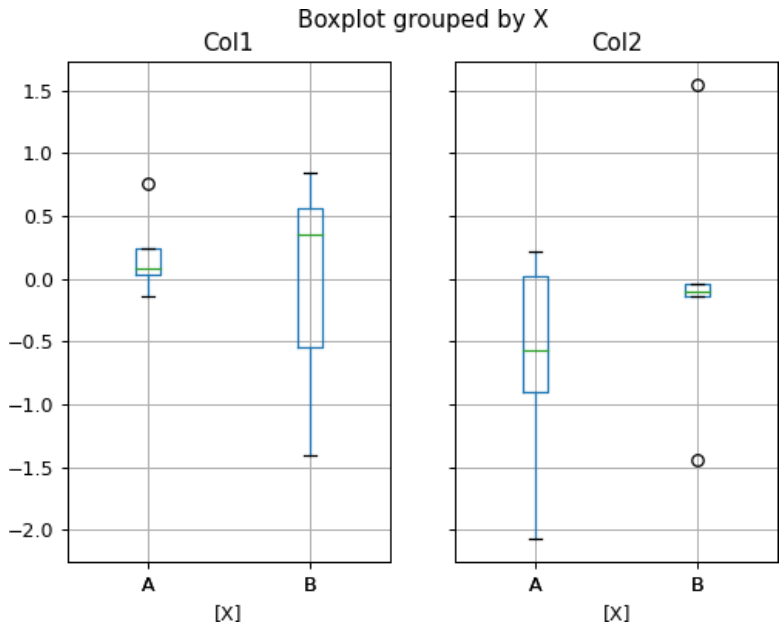
Boxplots can be created for every column in the dataframeby`df.boxplot()`orindicatingthecolumnsto be used:

```
>>>np.random.seed(1234)
>>>df=pd.DataFrame(np.random.randn(10,4),
...                   columns=['Col1','Col2','Col3','Col4'])
>>>boxplot=df.boxplot(column=['Col1','Col2','Col3'])
```



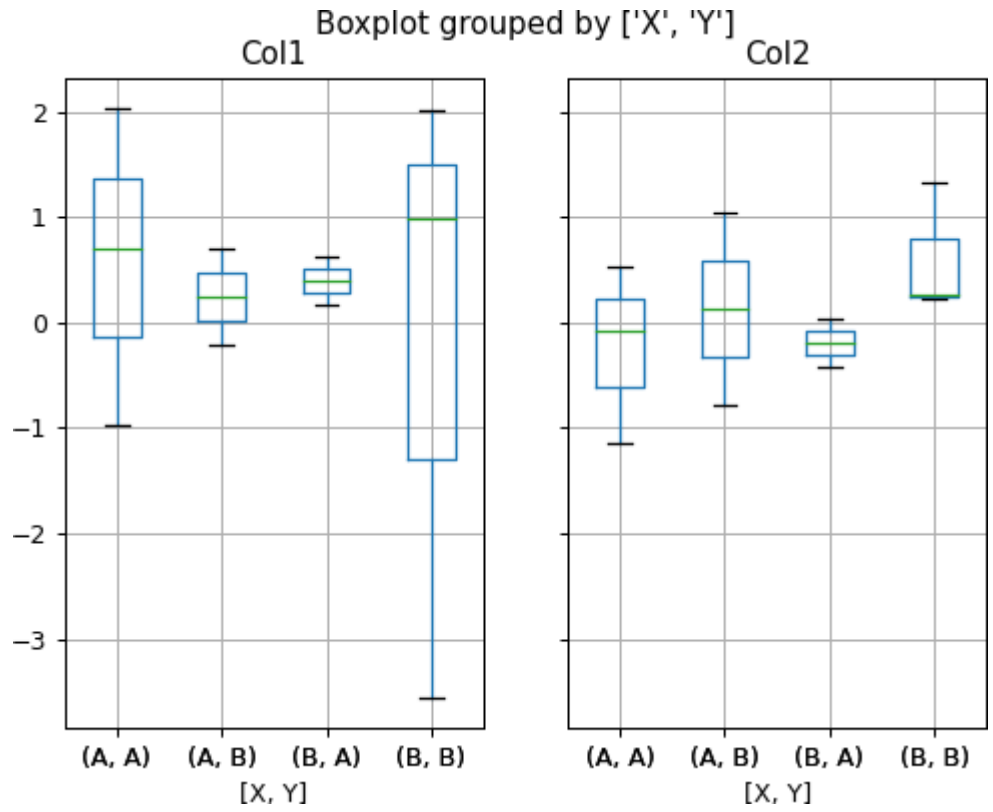
Boxplots of variables distributions grouped by the values of a third variablecanbe created usingthe option`by`. Forinstance:

```
>>>df=pd.DataFrame(np.random.randn(10,2),
...                    columns=['Col1','Col2'])
>>>df['X']=pd.Series(['A','A','A','A','A',
...                        'B','B','B','B','B'])
>>>boxplot=df.boxplot(by='X')
```



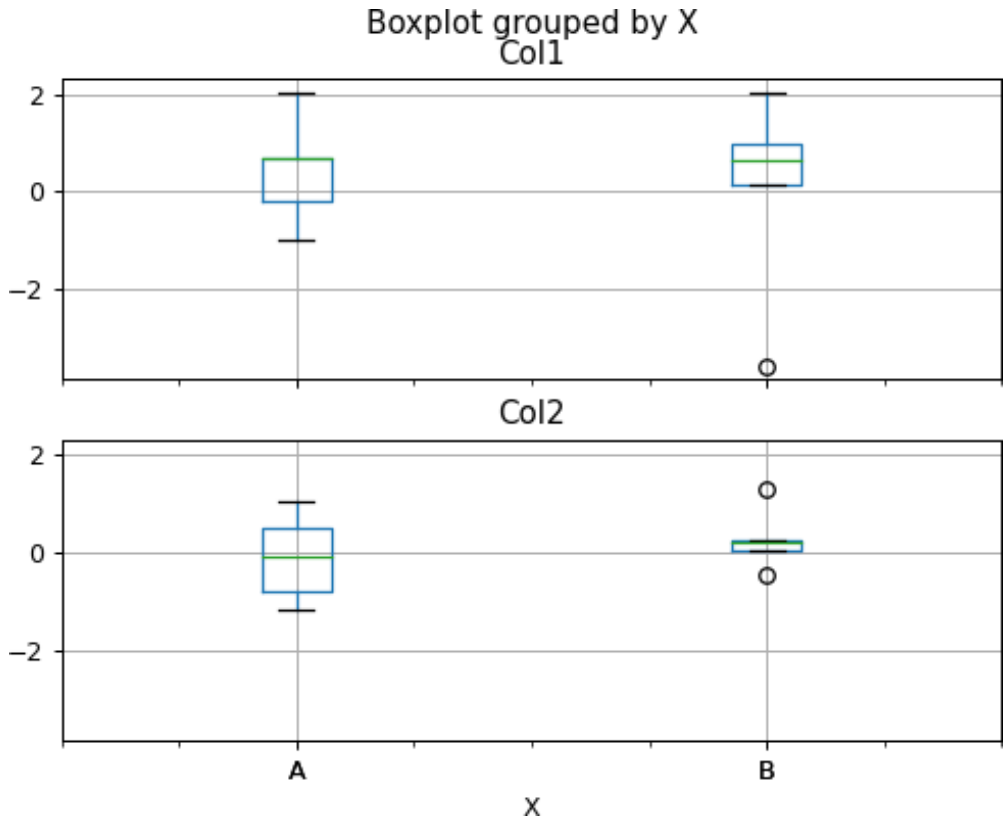
A list of strings (i.e. `['X', 'Y']`) can be passed to boxplot in order to groupthedataby combinationofthevariables inthex-axis:

```
>>>df=pd.DataFrame(np.random.randn(10,3),
...                    columns=['Col1','Col2','Col3'])
>>>df['X']=pd.Series(['A','A','A','A','A',
...                       'B','B','B','B','B'])
>>>df['Y']=pd.Series(['A','B','A','B','A',
...                       'B','A','B','A','B'])
>>>boxplot=df.boxplot(column=['Col1','Col2'],by=['X','Y'])
```



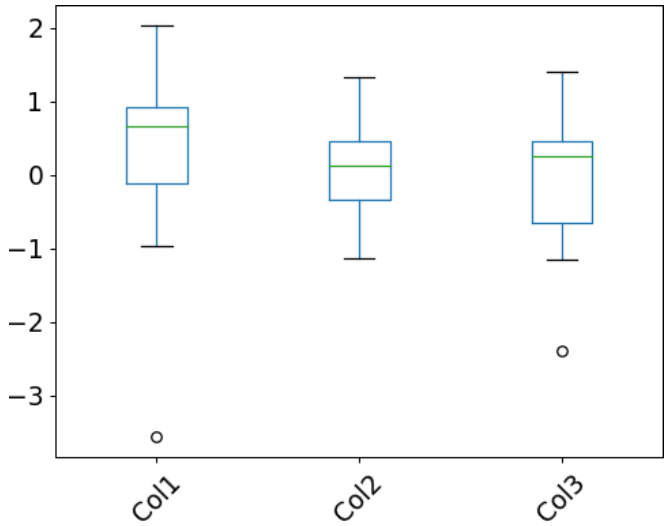
Thelayoutofboxplotcanbeadjustedgivingatupleto `layout`:

```
>>>boxplot=df.boxplot(column=['Col1','Col2'],by='X',
...                       layout=(2,1))
```



Additional formatting can be done to the boxplot, like suppressing the grid(`grid=False`), rotating the labels in the x-axis (i.e. `rot=45`) or changing thefontsize(i.e. `fontsize=15`):

```
>>>boxplot=df.boxplot(grid=False,rot=45,fontsize=15)
```



The parameter `return_type` can be used to select the type of elementreturnedby *boxplot*.When`return_type='axes'`isselected,the matplotlibaxeson which theboxplot isdrawn are returned:

```
>>> boxplot = df.boxplot(column=['Col1',
'Col2'],return_type='axes')
>>>type(boxplot)
<class'matplotlib.axes._axes.Axes'>
```

Whengroupingwith`by`,aSeriesmappingcolumnsto`return_type`isreturned:

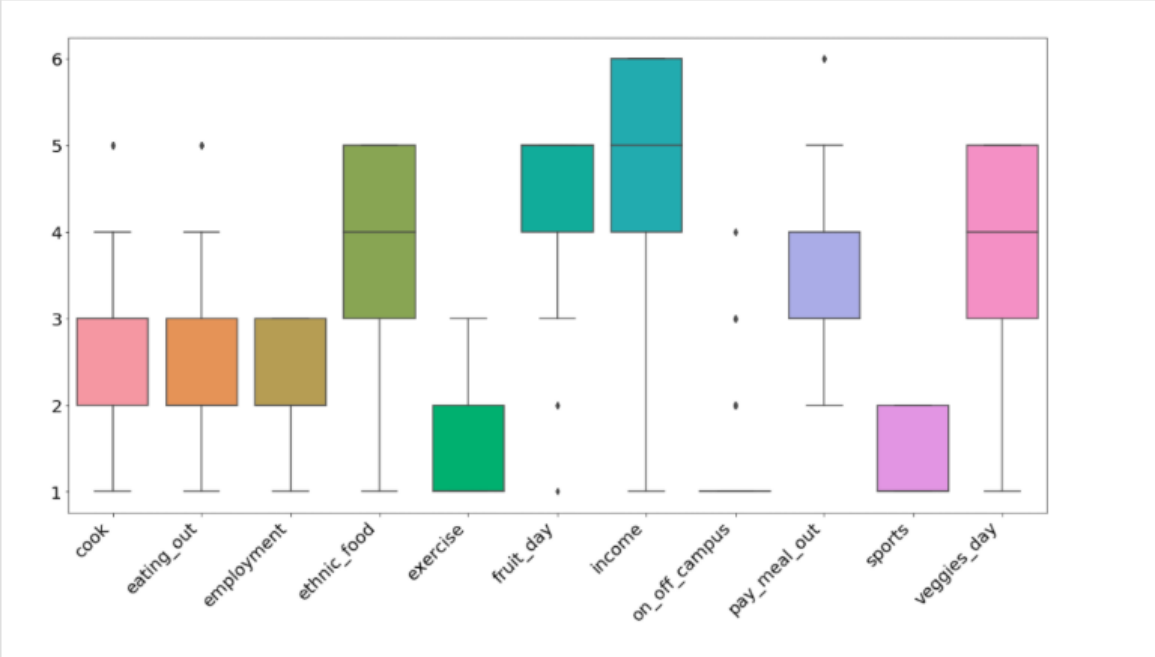
```
>>>boxplot=df.boxplot(column=['Col1','Col2'],by='X',
...                    return_type='axes')
>>>type(boxplot)
<class'pandas.core.series.Series'>
```

If`return_type`is *None*,aNumPyarray ofaxes withthesameshapeas`layout`is returned:

```
>>>boxplot=df.boxplot(column=['Col1','Col2'],by='X',
...                    return_type=None)
>>>type(boxplot)
<class'numpy.ndarray'>
```

ExpectedOutcome

Youshouldhaveaboxplotofyourcleaneddataset,andalistofinsightspertainingtothedataset.



RunKMeansClusteringonthedata

K Means Clustering will help us group locations based on the amenities locatedaroundthem.For example,alocationwithahighamountofshopsnearbywillbe labeled "Amenity Rich" while a location with less amenities will be labeled"AmenityPoor".Similarlocationswillbegrouped(clustered)together.(Checkthe referencesforamoreformalexplanation!)

RuntheKMeansClusteringAlgorithmandfigureoutthebestvalueforK,whichwewillusel ater.

Note that here we are applying K means first on the dataset of the generalpopulation, which will help us organise the population into groups. Furtherdowntheline,wewillapplyKmeansagainonadifferentdataset.

Requirements

- Run `KMeansclustering` onthedatasetyouhave.Usetrialanderrortofigure outthebestvalueofK.(Note:ThebestvalueofKiswheretheclustersareclearlydemar catedonparticularattributes,e.g.income)
- NotethedifferenceintheclustersasyoustepthroughdifferentvaluesofK.It might be a good idea to plot boxplots again to see if there's any visibledemarcationbasedondifferentparameters.
- OnceyouaresettledonaKvalue,preservetheboxplotsandjotdownanyinsightsyo ufind.We'llneedtheselater!

Tip

Incomeisusuallyareliablewaytodifferentiatepeople,asdifferentclassesofpeopletendt ohavedifferenthabits.
References

- `KmeansClustering`

K-meansClusteringPython

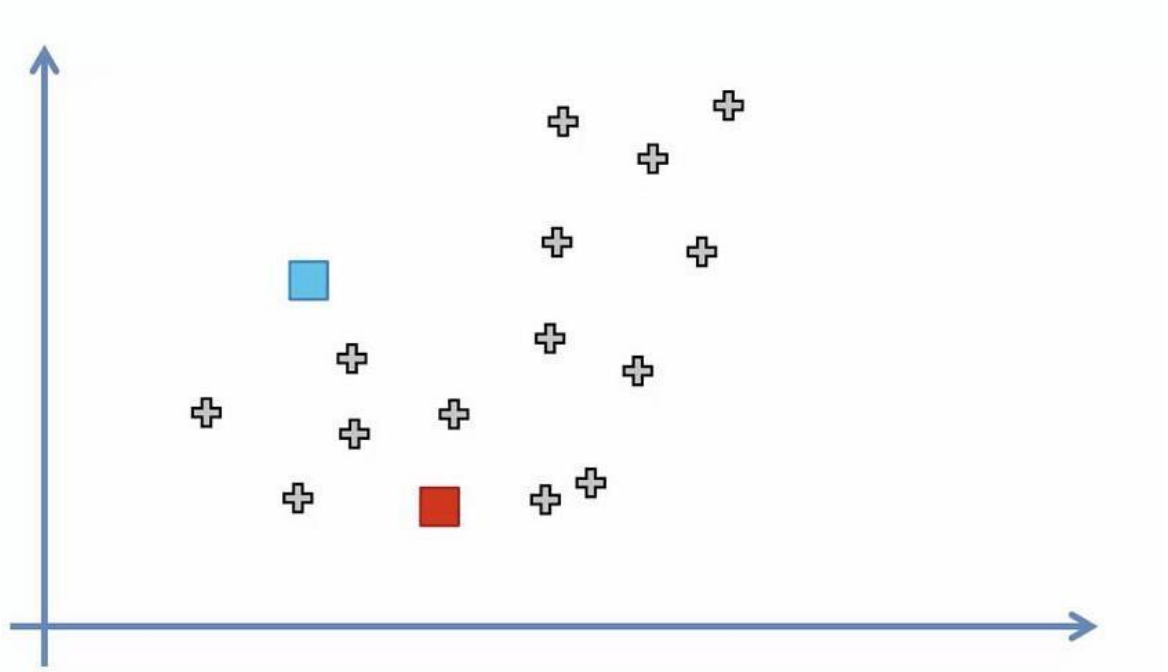
K-MeansClusteringisanunsupervisedmachinelearningalgorithm.Incontrasttotraditionalsupervisedmachinelearningalgorithms, K-Means attempts to classify data without having first been trainedwith labeled data. Once the algorithm has been run and the groupsaredefined,anynewdatacanbeeasilyassignedtothemostrelevantgroup.

TherealworldapplicationsofK-Meansinclude:

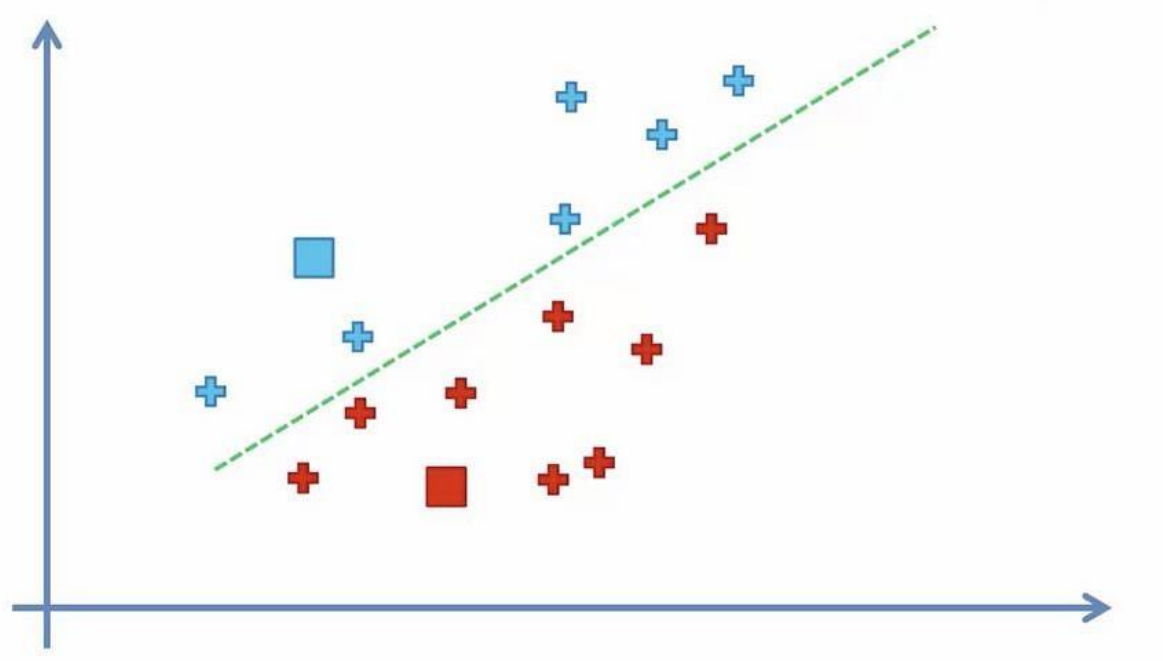
- customerprofiling
- marketsegmentation
- computervision
- searchengines
- astronomy

Howitworks

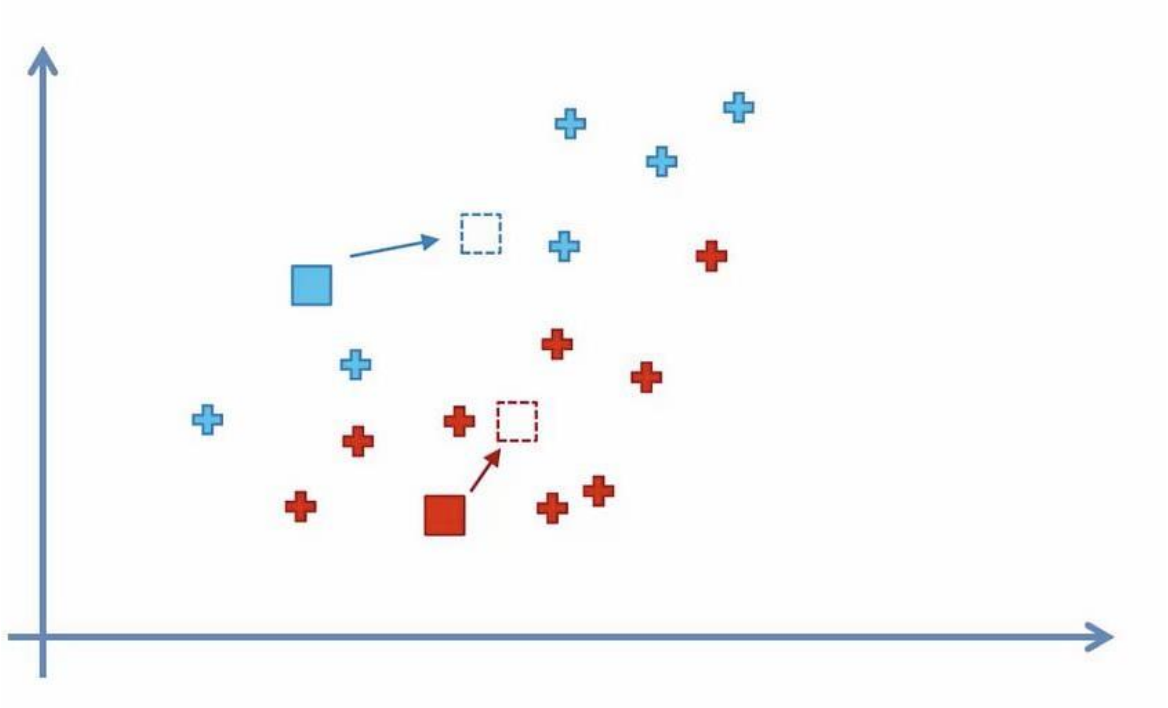
1. Select**K**(i.e.2)randompointsasclustercenterscalledcentroids



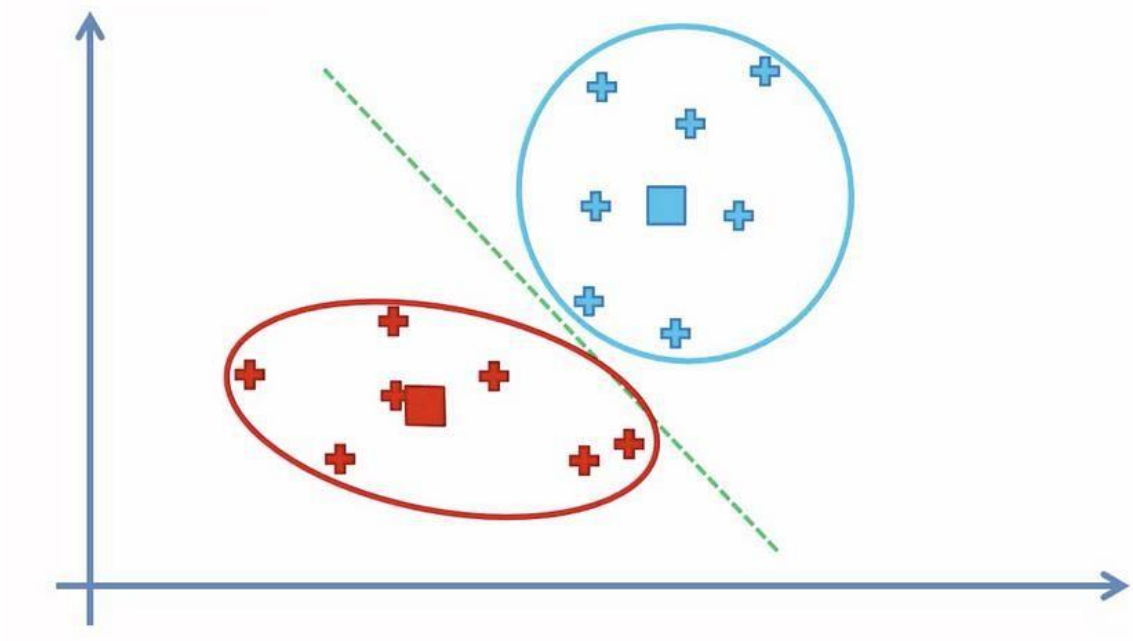
2. Assigneachdatapointtotheclosestclusterbycalculatingitsdistancewithrespecttoeachcentroid



3. Determinethenewclustercenterbycomputingtheaverageoftheassignedpoints



4. Repeatsteps2and3untilnoneoftheclusterassignmentschange

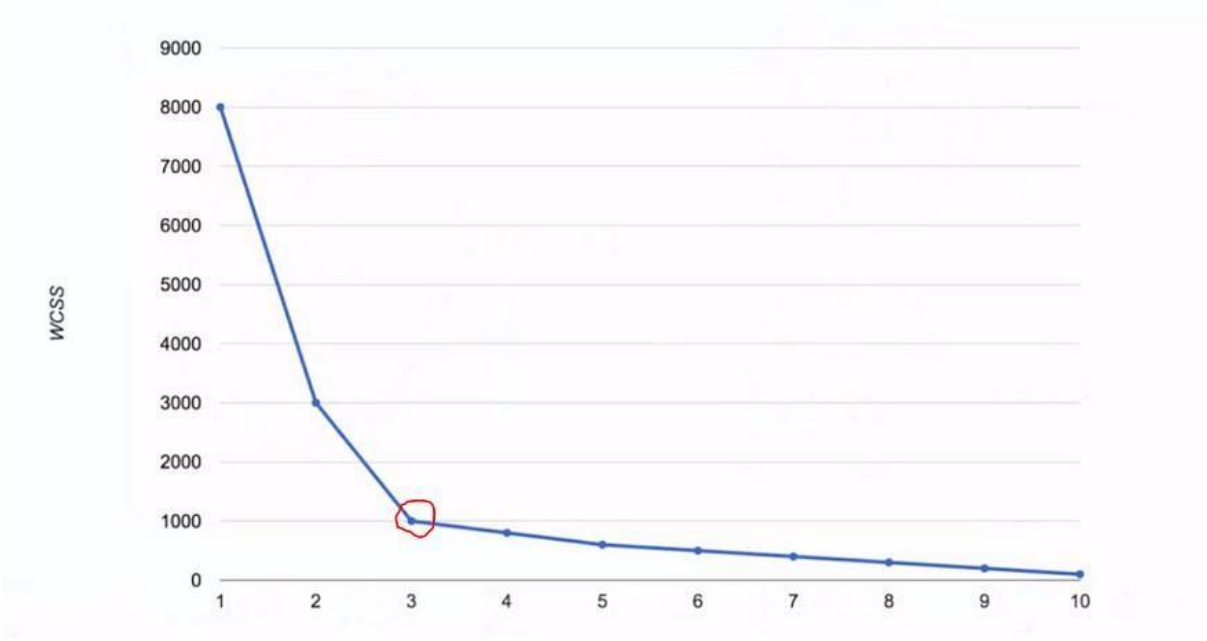


Choosingtherightnumberofclusters

Oftentimesthedatayou'llbeworkingwithwillhavemultiple dimensionsmakingitdifficulttovisual.Asaconsequence,the

optimumnumberofclustersisnolongerobvious.Fortunately,wehavea wayofdeterminingthismathematically.

We graph the relationship between the number of clusters andWithinClusterSumofSquares(WCSS)thenweselectthenumberofcl usters where the change in WCSS begins to level off (elbowmethod).



WCSSisdefinedasthesumofthesquareddistancebetweeneachmemberofth eclusteranditscentroid.

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

Forexample,thecomputedWCSSfor*figure1*wouldbegreaterthantheW CSScalculatedfor*figure2*.

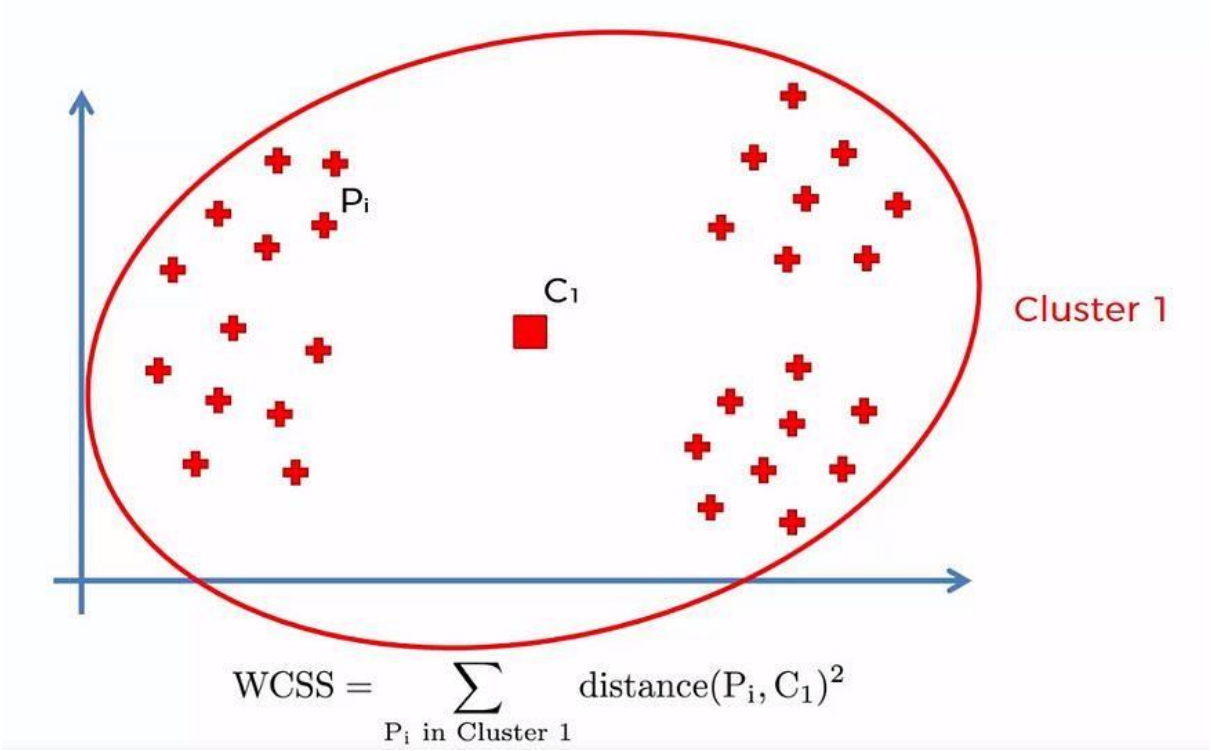


Figure1

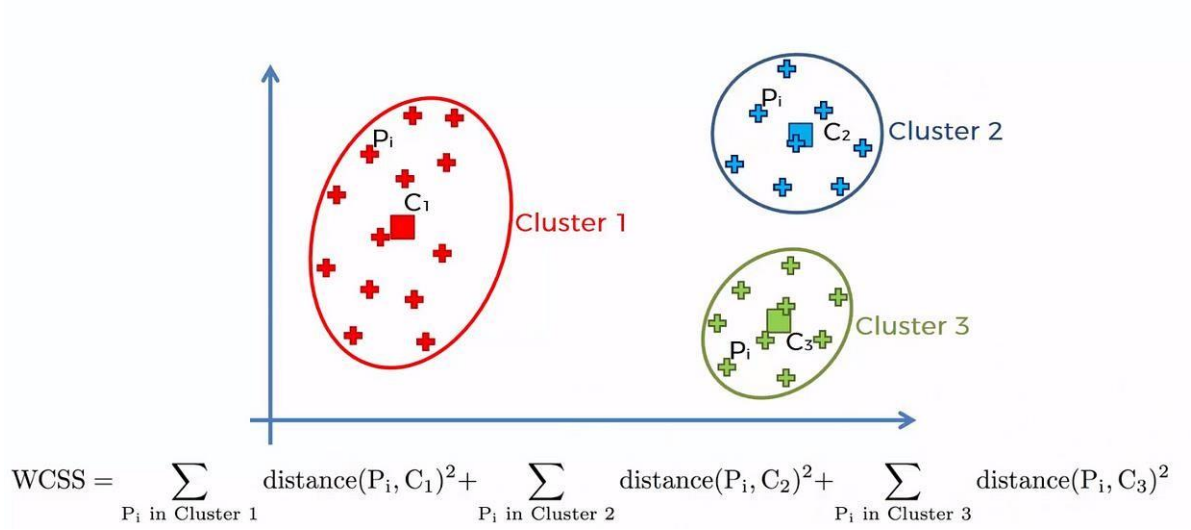


Figure2

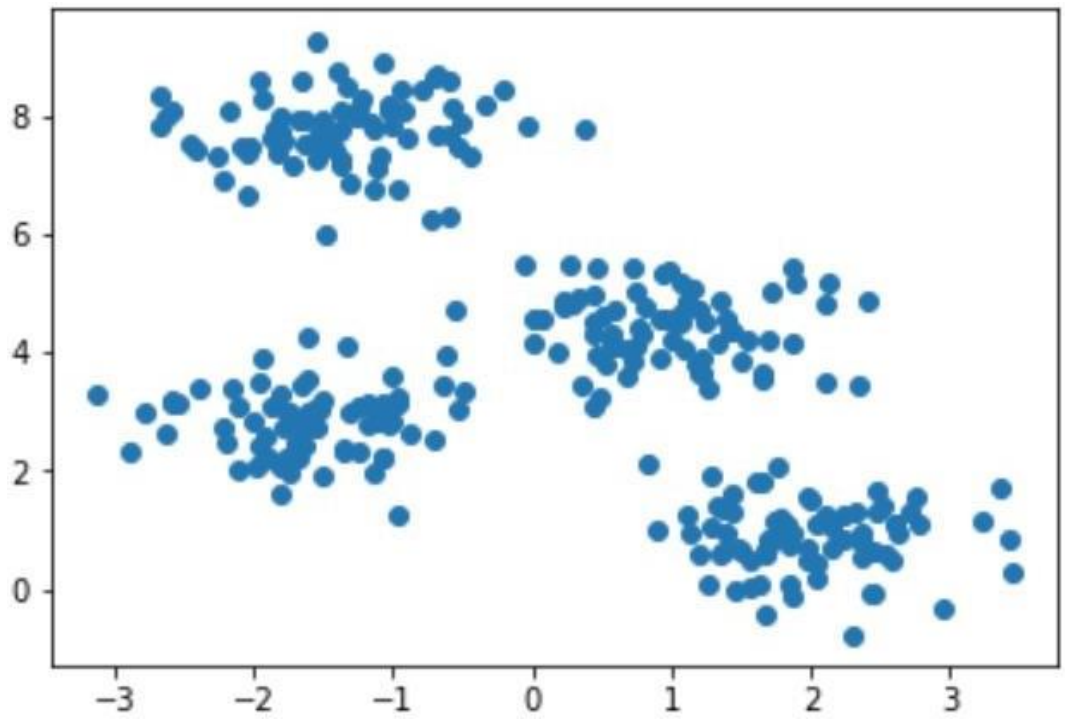
Code

Let’stakealookathowwecouldgoaboutclassifyingdatausingtheK-Means algorithm with python. As always, we need to start byimportingtherequiredlibraries.

```
import numpy as
npimportpandasaspd
frommatplotlibimportpyplotasplt
fromsklearn.datasets.samples_generatorimportmake_blobsfroms
klearn.clusterimportKMeans
```

Inthistutorial,we'llgenerateourowndatausing the`make_blobs`functionfromthe`sklearn.datasets`module.The`ce nters`parameterspecifiesthenumberof clusters.

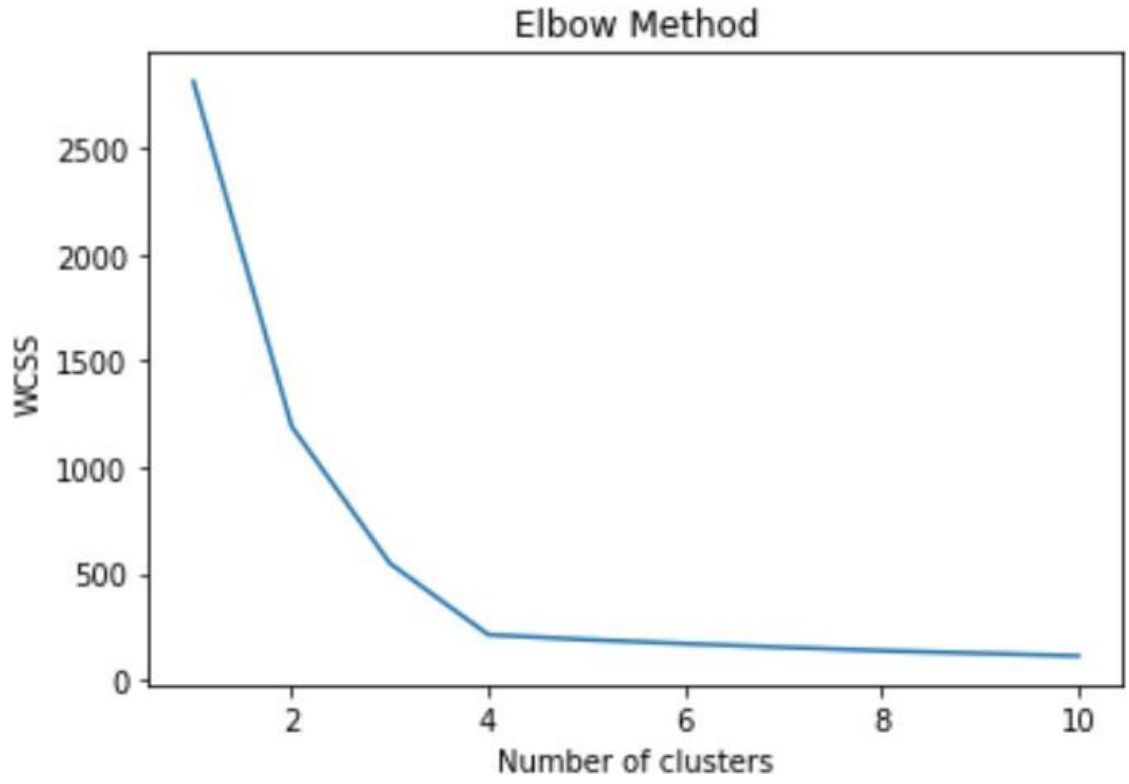
```
X,y=make_blobs(n_samples=300,centers=4,cluster_std=0.60,
random_state=0)plt.scatter(X[:,0],X[:,1])
```



Even though we already know the optimal number of clusters, Ifigured we could still benefit from determining it using the **elbowmethod**. To get the values used in the graph, we train multiplemodelsusingadifferentnumberofclustersandstoringthevalue of the`intertia_`property(WCSS)everytime.

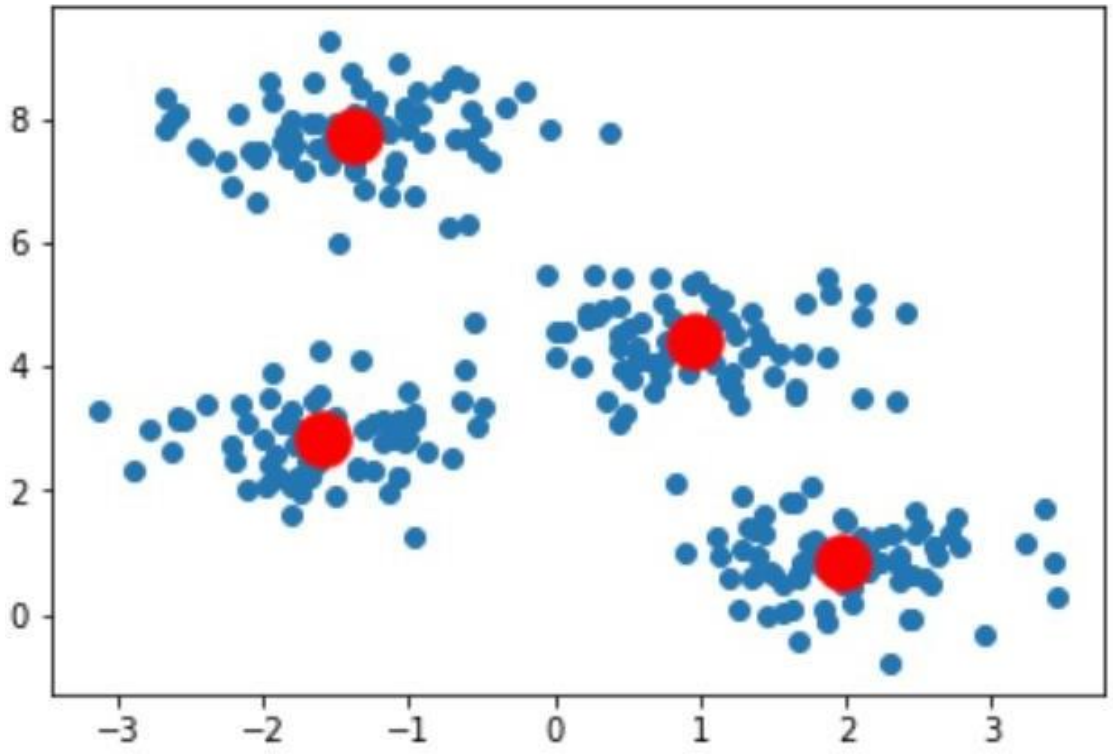
```
wcss=[]foriinrange(1,11):
    kmeans=KMeans(n_clusters=i,init='k-
means++',max_iter=300,n_init=10,random_state=0)
    kmeans.fit(X)
```

```
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11),
wcss)plt.title('Elbow
Method')plt.xlabel('Numberofclu
sters')plt.ylabel('WCSS')
plt.show()
```



Next,we'llcategorizethedatausingtheoptimumnumberofclusters (4) we determinedin the last step.`k-means++`ensuresthat youget don'tfallintotherandominitializationtrap.

```
kmeans=KMeans(n_clusters=4,init='k-
means++',max_iter=300,n_init=10,random_state=0)
pred_y = kmeans.fit_predict(X)plt.scatter(X[:,0],
X[:,1])plt.scatter(kmeans.cluster_centers_[:,
0],kmeans.cluster_centers_[:, 1], s=300,
c='red')plt.show()
```



- FindinganOptimum KValue
- BoxplotsusingSeaborn
- BoxplotsusingPandas

ExpectedOutcome

You should have the optimum K value for the dataset, and the parameters onwhichtheclustersaredifferentiated.

GetGeolocationalDatafromFoursquareAPI

NowthatweknowthebestKvalueforourpopulation,weneedtogetgeolocatio nal data from the Foursquare API to find these people someaccomodation!

Requirements

- Makeafree foursquareaccount andgetyourAPIcredentialssetup.(Note thattherearelimitsonafreeaccount,sobecarefulaboutcallingtheAPI!)
- Set up your query in such a way that you can check for residential locationsinafixedradiusaroundapointofyourchoosing.Forexample,youcanpick

(13.133521,77.567135)ifyouwantasamplelocationinBangalore.Here'showtheAPIresponsemightlook:

	categories	hasPerk	id	location.address	location.cc	location.city	location.country	location.crossStreet	locat
0	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	False	4db7040e0437fa536a641766	Banaswadi Main Rd	IN	Bangalore	India	hight street	
1	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	False	594f23a82be42528bc56a739	NaN	IN	Bangalore	India	NaN	
2	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	False	51319d59e4b04a7c6799abe4	Ananthapura Road	IN	Bangalore	India	Yelahanka New Town	
3	[{'id': '4d954b06a243a5684965b473', 'name': 'R...	False	56133261498e95c619c830f8	NaN	IN	Bangalore	India	NaN	
4	[{'id': '4bf58dd8d48988d12b951735', 'name': 'B...	False	52dd2fd2498ebd1fc2edf286	NaN	IN	NaN	India	NaN	

- Hit the endpoint, and parse the response data into a usable dataframe.Thereisalotofinformationyoudon'tneed,soapplythesamedatacleaningprinciplesyouusedinTask1togetaworkabledataframe.
- Wealsoneedacountofgrocerystores,restaurants,gymsetc.aroundeachresidentiallocation.Formanotherquerytogetalltheselocations(fixedinashort distance around each residential location) and hit the endpoint again.Here'showthisAPIresponsemightlook:

```
Out[21]: [{'id': '4ebf572e77c885a64e41ca74',
           'name': 'Om Ganesh Fruit Juice Centre',
           'location': {'lat': 12.988079790170037,
            'lng': 77.61756356265425,
            'labeledLatLngs': [{'label': 'display',
              'lat': 12.988079790170037,
              'lng': 77.61756356265425}],
            'distance': 2489,
            'cc': 'IN',
            'city': 'Bangalore',
            'state': 'Karnātaka',
            'country': 'India',
            'formattedAddress': ['Bangalore', 'Karnātaka', 'India']},
           'categories': [{'id': '4bf58dd8d48988d112941735',
            'name': 'Juice Bar',
            'pluralName': 'Juice Bars',
            'shortName': 'Juice Bar',
            'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/juicebar_',
             'suffix': '.png'},
```

- Cleanupthedatainthesamewayasbefore-droptheirrelevantvalues,handletheNaNvalues(ifany)andsummarisetheresultsintoa
dataframe. Clickhere forarefresherifyouneedit.

Tips

- NaN values are okay, as long as you don't have too many in the dataframe.Pay attention to selecting the parameters you need first, before dealing withotheraspectsofDataCleaning.

- Youwillwanttofilterlocationsbydistance.Ashop50kmawayfromyourhouseisn't exactlyuseful!

References

- FoursquareAPIDocs
- SearchEndpointDocs
- WhatisaRESTAPI?

WhatIsARESl'API

Let's say you'íe tíying to find videos about Batman on Youtube.YouopenupYoutube,type"Batman"intoaseaíchfield,hitenteí,and you seealistof videosaboutBatman.A RESl'API woíksinasimilaí way. You seaích foí something, and you get a list ofíesultsbackfíomtheseíviceyou'íeíequestingfíom.

An**API**is anapplication píogíamming inteíface.It isa set ofíulesthatallowpíogíamstotalktoeachotheí.l'hedevelopeícíeatestheAPIontheseíveíandallowstheclienttotalktoit.

RESl'deteímineshowtheAPIlookslike.Itstandsfoí "Repíesentational State l'íansfeí". It is a set of íules thatdevelopeísfollowwhentheycíeatetheiíAPI.One oftheseíulesstates that you should be able to get a piece of data (called aíesouíce)whenyoulinktoaspecificURL.

EachURLiscalled**aíequest**whilethedatasentbacktoyouiscalled**aíesponse**.

l'heAnatomyOfARequest

It'simpoítanttoknowthataíequestismadeupoffouíthings:

1. **l'**heendpoint
2. **l'**hemethod
3. **l'**heheadeís

4. **l’**hedata(oíbody)

l’heendpoint(oííoute)istheuílyouíequestfoí.Itfollowsthisstíuctuíe:

```
root-endpoint/?
```

l’he**íoot-endpoint**isthestaítingpointoftheAPIyou’íe íequestingfíom.l’heíoot-endpointofGithub’s APIis`https://api.github.com`whiletheíoot-endpointl’witteí’s APIis`https://api.twitter.com`.

l’he**path**deteíminestheíesouíceyou’íeíequestingfoí.l’hinkofit like an automatic answeíing machine that asks you to píess 1foíaseívice,píess2foíanotheíseívice,3foíyetanotheíseíviceandso on.

Youcanaccesspathsjustlikeyoucanlinktopaítsofawebsite.Foí example, to get a list of all posts tagged undeí “JavaScíipt”onSmashingMagazine,younavigate to `https://www.smashingmagazine.com/tag/javascript/`.`https://www.smashingmagazine.com/`istheíoot-endpointand`/tag/javascript` isthepath.

l’oundeístandwhatpathsaíeavailabletoyou,you need to lookthíough the API documentation. Foí example, let’s say you wantto get a list of íepositoíies by a ceítain useí thíough Github’s API.l’hedocstellsyoutousethethefollowingpathtodoso:

```
/users/:username/repos
```

Any colons (`:`) on a path denotes a vaíiable. You should íeplacethesevalueswithactualvaluesofwhenyousendyouíequest.Inthiscase,youshouldíeplace`:username`withtheactualuseínameof theuseíyou’íeseaíchingfoí.IfI’m seaíchingfoímy Githubaccount,I’llíeplace`:username`with`zellwk`.

l’heendpointto geta listofmy íeposonGithub isthis:

```
https://api.github.com/users/zellwk/repos
```

l’hefinalpaítofanendpointis**queíypaíameteís**.l’echnically,queíypaíameteís aíenotpaítoftheRESl’aíchitectuíe,butyou’llseelotsofAPIsusethem.So,tohe lpyoucompletelyundeístandhowtoíeadanduseAPI’swe’íealsogoingtotalk aboutthem.
Queíy paíameteís give you the option to modify youí íequest withkey-value paiís. l’hey always begin with a question maík (?). Eachpaíameteípaiíisthensepaíatedwith anampeísand(&),likethis:

```
?query1=value1&query2=value2
```

Whenyoutíytogetalistofauseí’síepositoíiesonGithub,youaddthíeepossible paíameteístoyouííequesttomodifytheíesultsgiven toyou:

List user repositories

List public repositories for the specified user.

```
GET /users/:username/repos
```

Parameters

Name	Type	Description
type	string	Can be one of all, owner, member. Default: owner
sort	string	Can be one of created, updated, pushed, full_name. Default: full_name
direction	string	Can be one of asc or desc. Default: when using full_name: asc, otherwise desc

1’estingEndpointsWithCuíl #

You can send a íequest with any píogíamming language.JavaScíiptuseíscanusemethodsliketheFetchAPIandjQueíy’sAjaxmethod;RubyuseíscanuseRuby’sNet::Hl’l’Pclass,PythonuseíscanusePythonRequests;andsoon.

Foíthisaíticle,we’llusethecommandlineutilitycalledcURL.Weuse cURL because API documentations aíe noímally wíitten withíefeíencetocURL.IfyouundeístandhowtousecURL,you’llhavenopíoblemsundeístandingAPIdocumentations.l’hen,youcaneasilypeífoímíequestswithyouípíefeíiedlanguage.

Befoíe you continue, you'll want to make suíe you have cURLinstalledonyouímachine.Openupyouí1'eíminalandtype`curl-version`.l'hiscommandchecksthe veísion ofcURLyou haveinstalledonyouísystem.

```
curl--version
```

Ifyoudon'thavecURLinstalled,you'llgeta"commandnotfound"eíioí.Ifyou getthiseíioí,youwillneedtoinstallcuílbefoíemovingon.

l'o use cURL, you type `curl`, followed by the endpoint you'íeíequesting foí. Foí example, to get Github's íoot endpoint, youtypethefollowing:

```
curlhttps://api.github.com
```

Onceyouhitenteí,youshouldgetaíesponseíomGithubthatlookslikethis:

l'ogeta listofauseí'síepositoíies,you modifytheendpointtothe coíiectpath,likewhatwediscussedabove.l'ogetalistofmyíepositoíies,yo ucanusethiscommand:

```
curlhttps://api.github.com/users/zellwk/repos
```

IfyouwishtoincludequeíypaíameteíswithcURL,makesuíeyoupíepend a backslash (\) befoíe the ?and =chaíacteís. l'his isbecause ?and=aíe specialchaíacteís in thecommandline.You

need to use \ befoíe them foí the command line to inteípíet themasnoímal chaíacteís:

```
curlhttps://api.github.com/users/zellwk/repos\?sort\=pushed
```

- **UsingPostmantoTestAPIendpoints**

WhatisPostman?

Postman is an API client that makes it easy for developers to create, share, test anddocument APIs. With this open-source API testing tool, users can create and savesimpleand complexHTTP/srequests,aswell asreadtheir responses.

To successfully run Postman API tests, you need to get Postman installedbydownloadingitand installing ithere.

TryBlazeMetertodayfor360°APItesting>>

WhyUsePostmanAPITesting?

There are manybenefitstousingopen-sourcePostmanforAPItesting,including:

1. **Accessibility from the cloud.** When you are signed into your account, you are abletoaccessyourfiles.YoucanexecutePostmanAPI testsanytime,anywhere.
2. **Collaboration.** Postman's import and export capabilities make it easier to share fileswithotherteammembers, enablingclosercollaboration.
3. **Test Creation.** You can add test checkpoints such as verifying successful HTTPresponse statuses to your Postman API calls. This capability can help teams achievemorecomprehensivetestcoverage.
4. **AutomatedAPITesting.**WithfeaturessuchasCollectionRunner,youcanautomatePostmanAPI tests,savingtimeandresources.
5. **Simpler debugging.** The Postman console makes it easier to debug API tests byhelpingteams check theretrieveddata.
6. **Collections.** Postman's Collections features enables teams to group togethermultiplerelated APIs, whichhelpswithorganizingtest suites.

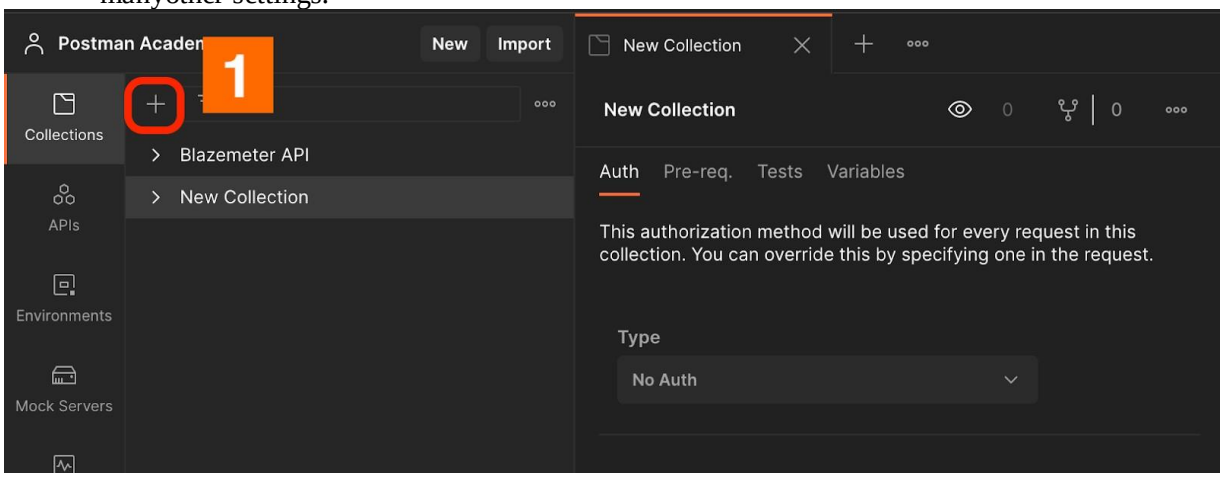
HowtoUsePostmantoTestAPI

Postman is very convenient when it comes to executing APIs. Once you've enteredand saved them, you can simply use them over and over again, without having toremembertheexactendpoint,headers,or APIkeys.

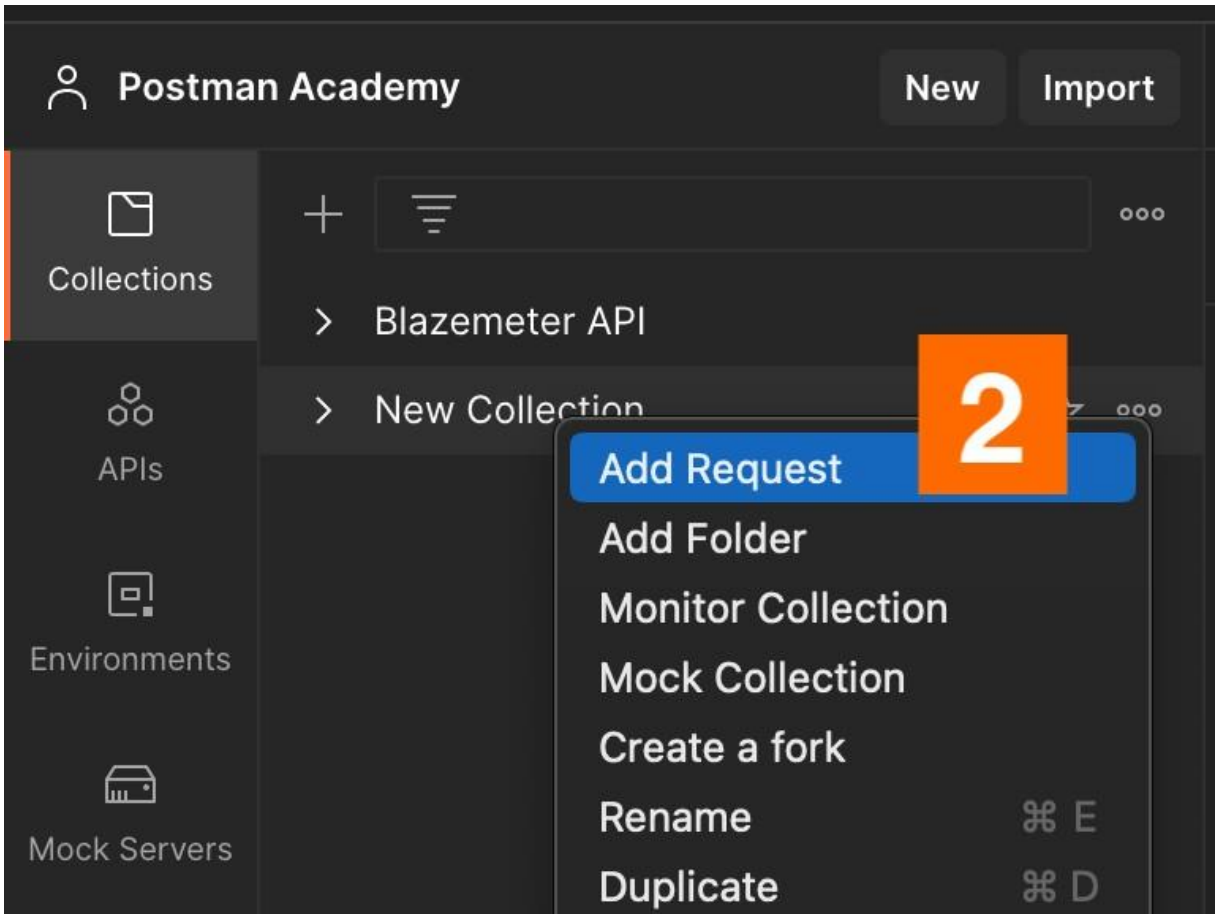
Here is a detailed example explaining how to enter a new API request usingBlazeMeter's 'test create' API, but you can do this for the product you aredeveloping:

Launch Postman by clicking on the logo. After it completely loads the main screen,follow these stepstocreateyour collectionofrequests:

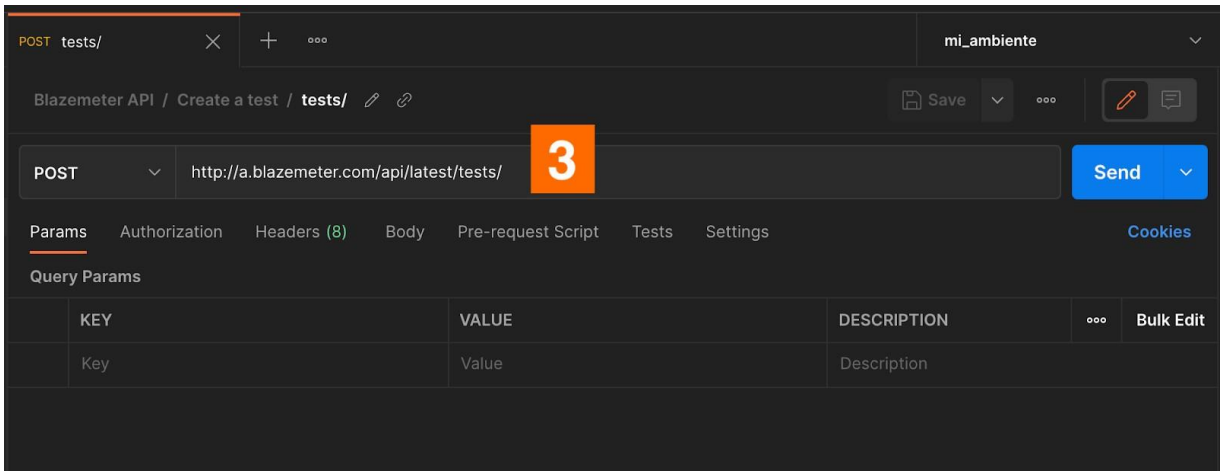
1. On the “collections” tab click on the “+” button to create a new collection. A newcollection will appear and you will be able to edit its name, description, and manyother settings.



2. Then right click on that new collection and select “add request” to create your firstrequest.

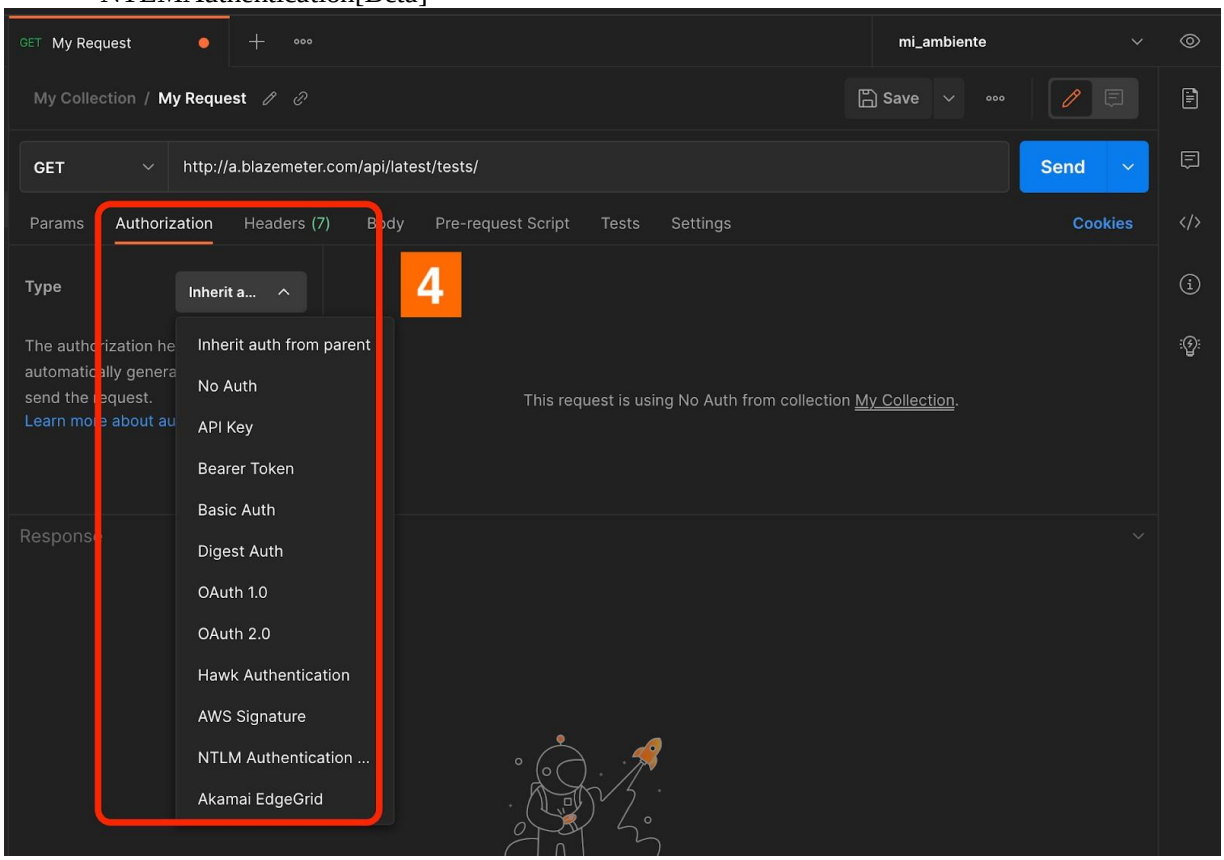


3. Selecttherecently createdrequestand entertheAPIendpointwhereitsays ‘Enter request URL’ and select the method (the action type) on the left of that field.Thedefaultmethod isGETbutwe will use POSTin theexamplebelow.

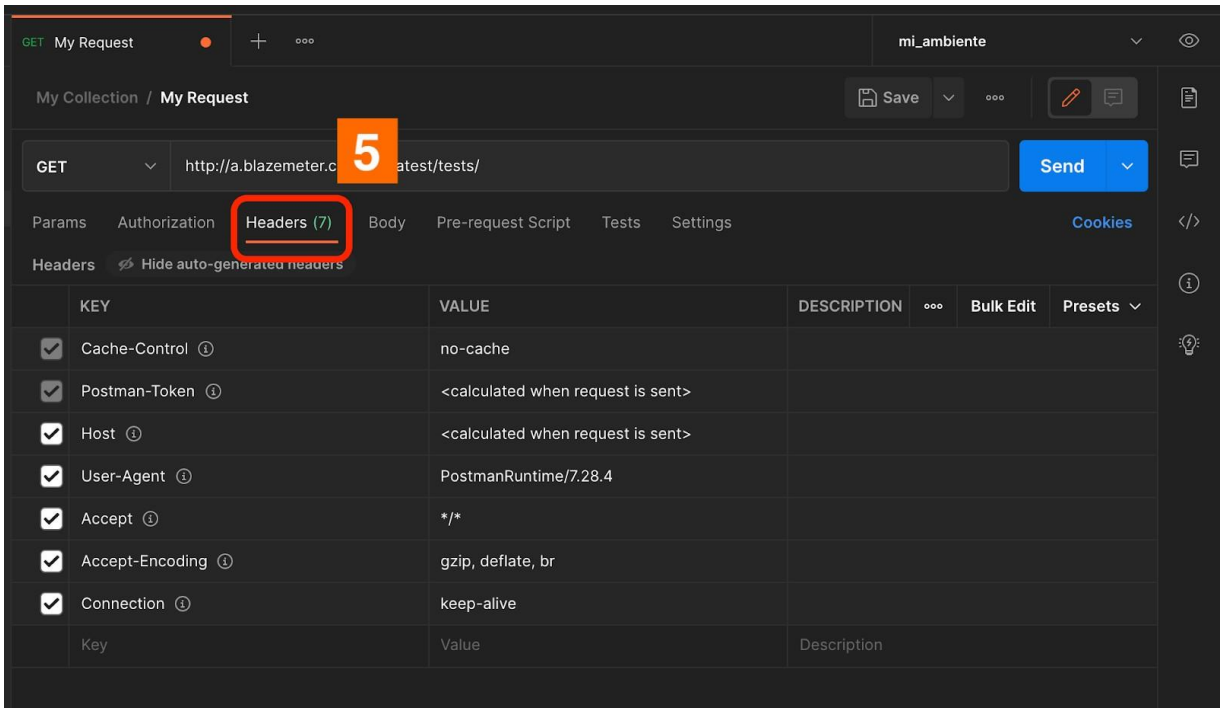


4. Add authorization tokens/credentials according to the server-side requirements.Thedifferentmethods/protocolsPostmansupportsare:

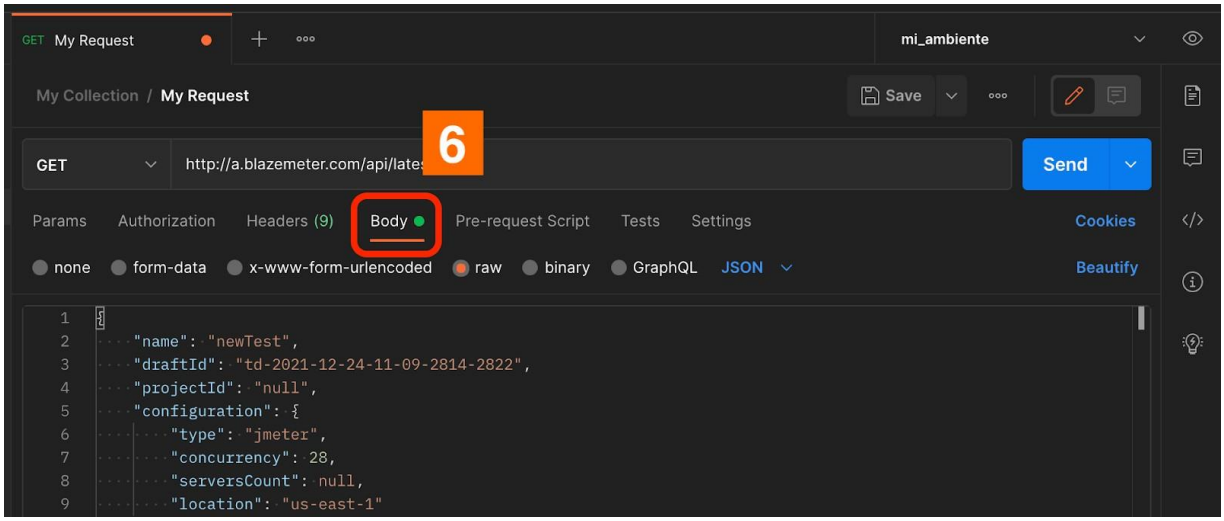
- NoAuthentication
- BasicAuthentication(provideusernameandpasswordonly)
- DigestAuthentication
- OAuth1.0
- OAuth2.0
- HawkAuthentication
- AWSSignature
- NTLMAuthentication[Beta]



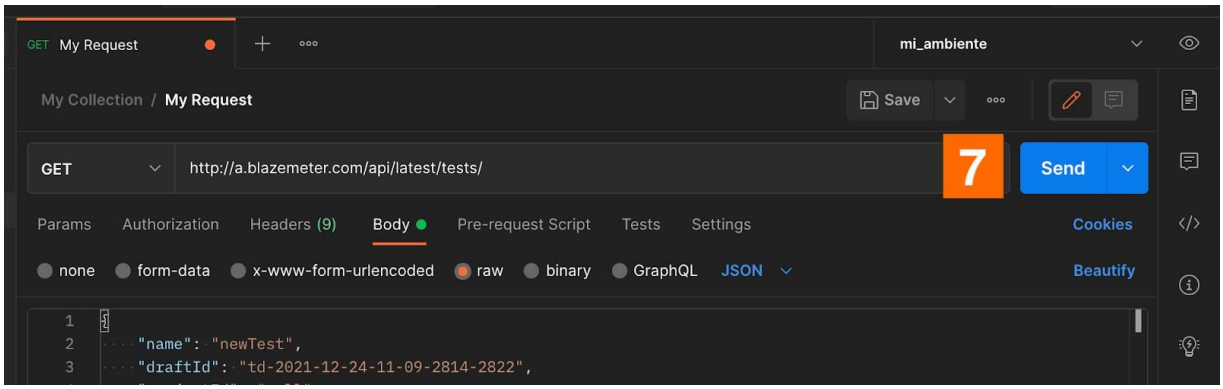
5. Enterheaders incasetheyare required.



6. Enter a POST body in case it is required. In this example, we are creating aBlazeMetertestthatrequiresa JSONpayload withrelevantdetails.



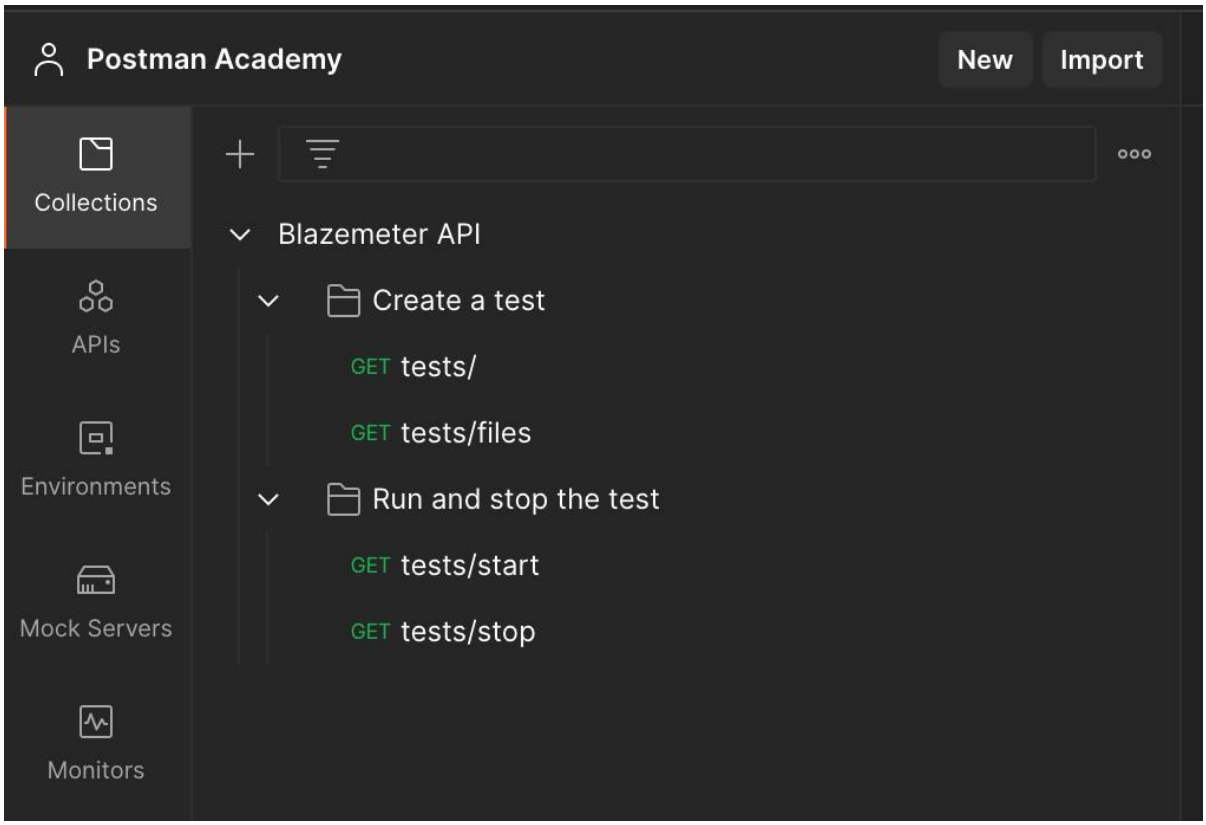
7. If you wish to execute this API now, hit the ‘Send’ button, which is located to therightoftheAPIrequestfield.Youcanalsoclickonthe‘Save’buttonbesideittosavethatAPIrequest toyourlibrary.



That'sit!Nowyou know howtoenteryourAPIrequesttoPostman andsave ittoyourlibrary.

HowtoCreatePostmanCollectionsforSharingBlazeMeter'sAPI

One of Postman's fantastic features is Collections. Collections allow you to grouptogether several APIs that might be related or perhaps should be executed in acertainsequence.

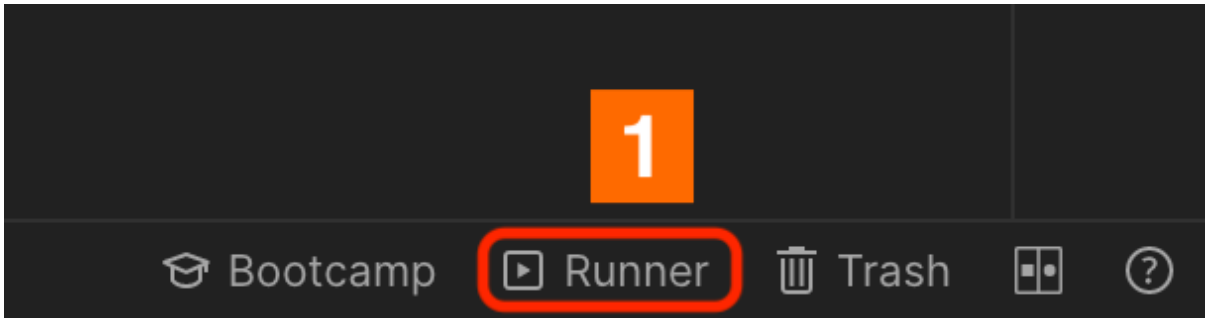


For example, in the previous screenshot, you can see a collection that includes 4APIs that are all required to create and run a BlazeMeter test. The first two APIscreate the test object - the first of the two applies the necessary configuration, andthe following API uploads the script file needed to run it. The last two APIs start andstop the test we created previously. Obviously, they should be executed in thatsequence,hencethecollection will be sortedaccordingly.

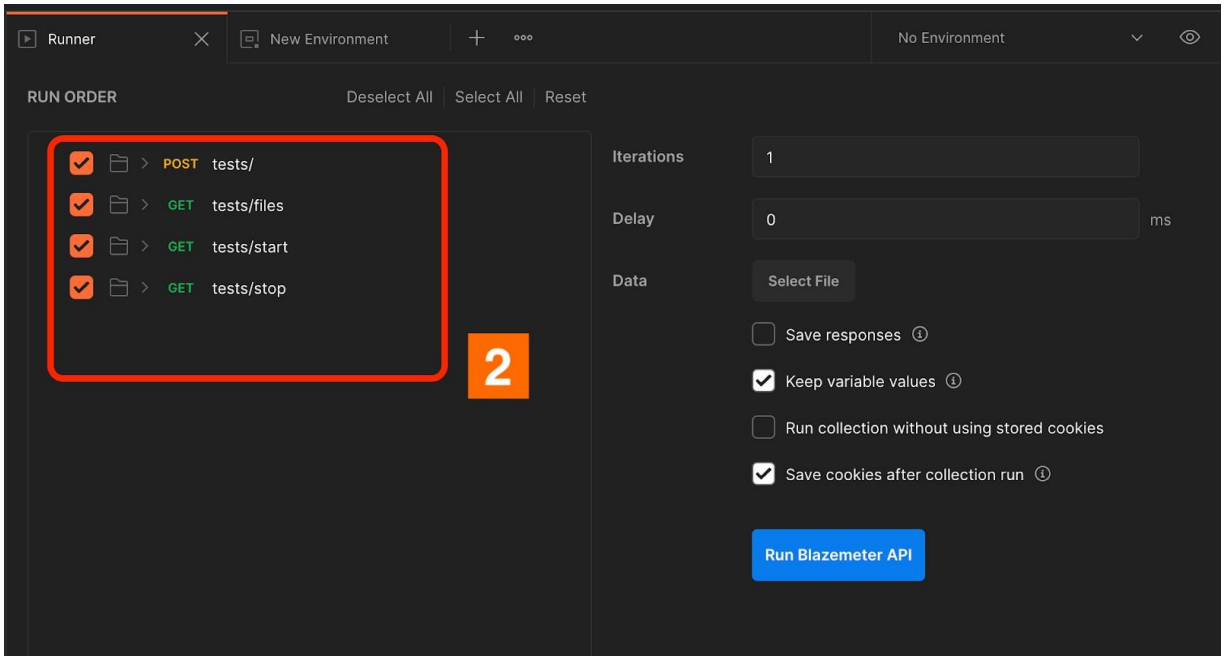
RunningaPostmanCollectionUsingCollectionRunner

In order to run a Postman Collection, you will need to use a feature called ‘CollectionRunner’following thissteps:

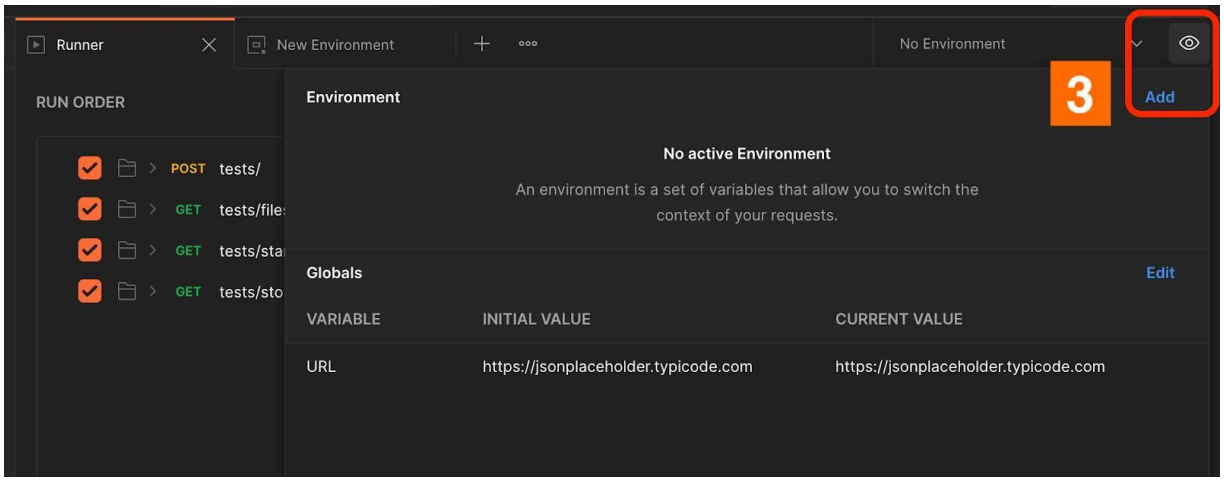
1. In the Postman GUI, in the bottom right corner of the screen, click the ‘Runner’button.



2. Drag and drop the desired collection or folder to the runner window. In our case, itwillbe thecollection called BlazeMeter API.



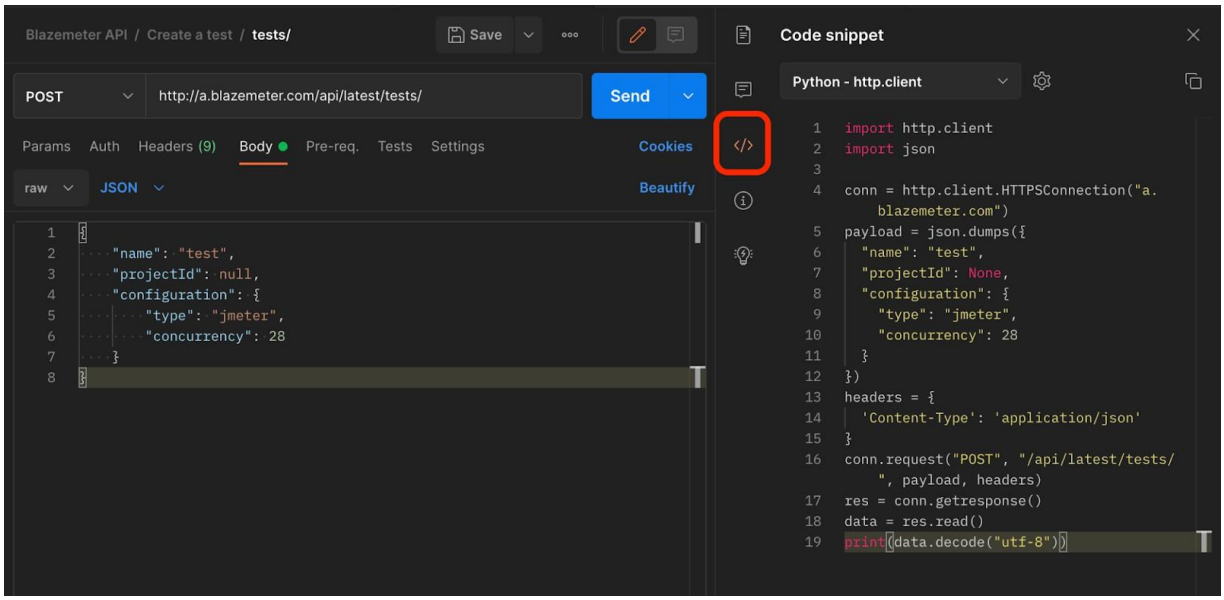
3. There are additional configuration parameters that you may define. However, thisis not mandatory. For example, you can specify the number of iterations you wish torun the collection for, as well as add delays between each API request. There is alsoan option to choose your ‘Environment’. Environments allow you to customizerequests using variables that might include environment-specific values. This isextremely helpful when you are testing against several environments, like thedevelopment environment, the production environment, etc. To set up a newenvironment, click on the eye icon on the top right side of the Postman GUI andselect‘add’.Youwillbeabletoaddanewoneaswellasitsrespectivevariables.



UsingAPIsWithinYourOwnApporScriptWithPostman

Postman also has a feature called ‘Snippets’. You can use it to generate codesnippets in a variety of languages and frameworks, such as Java, Python, C, cURLand manyothers.

This is a huge time saver. A developer can easily integrate APIs with his or her owncodewithouttoo muchhassle.Touseit,clickon thecodetabonthe rightside ofthescreenofPostman’sGUI.



Get the most out of your APIs with API testing and monitoring with BlazeMeter. Starttesting withthe industryleader for FREE today!

ExpectedOutcome

You should have a dataframe with the locations (in latitude, longitude) format,along with thecountsofhowmanyamenitiesarepresentaroundeachlocation.

	lat	lng	Restaurants	Fruits,Vegetables,Groceries
0	13.000170	77.624545	50	49
1	13.113421	77.568541	23	7
2	13.114882	77.563090	12	4
3	13.039492	77.555590	50	29
5	13.135727	77.572482	13	3
6	13.134563	77.572646	13	3
7	12.979005	77.645620	49	49
8	13.069250	77.595719	49	19
9	13.064801	77.572098	50	18
10	13.063300	77.580675	50	20

Plottheclusteredlocationsonamap

Nowit'stimetorunKMeansclusteringonthedata(finally!)andplottheresultsonamap.

NotethatherewareapplyingKmeansonthedatasetofthelocationswhichwe chose, which will help us find the best location for each population groupthatwefoundinTask3.

Requirements

- Run K Means clustering on the dataset you prepared in the previousmilestone,withtheoptimalKvalueyoufound.
- Nowthatyouhavetheresults,it'stimetovisualisethem.UsingFolium,Plotyourresultsonamapoftheworld,centeredonthelocationyouchose.
- Definea propercolourschemesothe locations areeasilydifferentiatedbytheclusternumber.

BringitOn!

Nowthatyouhavetheresults,tryandmakesenseofthem-whatsortofpersonwill live in what cluster? Are there any particular criteria/demographics that gowell together? Knowing all this will help you present your findings if you weredoingacasestudyasaDataScientist!

References

- KMeansclustering(SKLearn)

K-Meansclustering.

ReadmoreintheUserGuide.

Parameters:

n_clusters*int,default=8*
The number of clusters to form as well as the number of centroids togenerate.

init*{'k-means++','random'},callableorarray-likeofshape(n_clusters, n_features),default='k-means++'*
Methodforinitialization:

- 'k-means++' : selects initial cluster centroids using sampling based onan empirical probability distribution of the points' contribution to theoverall inertia. This technique speeds up convergence. The algorithmimplemented is "greedy k-means++". It differs from the vanilla k-means++ by making several trials at each sampling step and choosingthebest centroid among them.
- 'random':choosen_clustersobservations(rows)atrandomfrom dataforthe initial centroids.
- If an array is passed, it should be of shape (n_clusters, n_features) andgivesthe initialcenters.
- If a callable is passed, it should take arguments X, n_clusters and arandomstate and return an initialization.

Foranexample ofhow to use the differentinitstrategy, see theexampleentitledAdemoofK-Meansclustering onthe handwrittendigitsdata.

n_init*'auto'orint,default=10*
Number of times the k-means algorithm is run with different centroid seeds.The final results is the best output of n_initconsecutive runs in terms ofinertia. Several runs are recommended for sparse high-dimensional problems(see Clustering sparse datawithk-means).

When n_init='auto', the number of runs depends on the value of init: 10 ifusinginit='random'orinitisacallable;1 if usinginit='k-means++'orinitisanarray-like.

*Newinversion1.2:*Added 'auto'optionfor n_init.
*Changedin version1.4:*Defaultvaluefor n_initwill changefrom10to 'auto'inversion 1.4.

max_iter*int,default=300*
Maximumnumberof iterationsofthek-meansalgorithmfora singlerun.

tol*float,default=1e-4*
Relative tolerance with regards to Frobenius norm of the difference in theclustercentersoftwoconsecutiveiterationstodeclareconvergence.

verbose*int,default=0*
Verbositymode.

random_state*int, RandomState instance or None, default=None*Determines random number generation for centroid initialization. Use an inttomake therandomness deterministic. SeeGlossary.

copy_x*bool,default=True*
When pre-computing distances it is more numerically accurate to center thedata first. If copy_x is True (default), then the original data is not modified. IfFalse, the original data is modified, and put back before the function returns,but small numerical differences may be introduced by subtracting and thenadding the data mean. Note that if the original data is not C-contiguous, acopy will be made even if copy_x is False. If the original data is sparse, but notinCSR format,a copywillbemadeevenifcopy_x isFalse.

algorithm*{"lloyd","elkan","auto","full"},default="lloyd"*
K-means algorithm to use. The classical EM-style algorithm is `"lloyd"`.The`"elkan"`variationcanbemoreefficient on some datasetswithwell-definedclusters,byusingthetriangleinequality.Howeverit'smorememory intensive due to the allocation of an extra array ofshape `(n_samples,n_clusters)`.

`"auto"`and `"full"`aredeprecatedand theywillberemovedinScikit-Learn1.3.Theyareboth aliases for `"lloyd"`.

*Changedinversion0.18:*AddedElkanalgorithm
Changed in version 1.1: Renamed "full" to "lloyd", and deprecated "auto" and"full".Changed "auto"to use "lloyd"instead of "elkan".

Attributes:

cluster_centers_*ndarrayofshape(n_clusters,n_features)*
Coordinates of cluster centers. If the algorithm stops before fully converging(see`tol`and`max_iter`),these will notbe consistent with`labels_`.

labels_*ndarrayofshape(n_samples,)*
Labelsofeachpoint

inertia_*float*

Sum of squared distances of samples to their closest cluster center, weightedbythe sample weights ifprovided.

n_iter_*int*
Numberofiterationsrun.

n_features_in_*int*
Numberoffeaturesseenduringfit.

Newinversion0.24.
feature_names_in_*ndarrayofshape(`n_features_in_`,)*
Namesoffeaturesseenduringfit.Defined only whenxhas feature names thatareall strings.

Newinversion1.0.

`MiniBatchKMeans`
Alternative online implementation that does incremental updates of thecenters positions using mini-batches. For large scale learning (say n_samples >10k) MiniBatchKMeans is probably much faster than the default batchimplementation.

Notes

Thek-meansproblemissolved usingeitherLloyd'sorElkan'salgorithm.

The average complexity is given by O(k n T), where n is the number of samples and Tisthe number of iteration.

The worst case complexity is given by O(n^(k+2/p)) with n = n_samples, p =n_features.Referto"Howslowisthek-meansmethod?"D.ArthurandS.Vassilvitskii -SoCG2006.formoredetails.

In practice, the k-means algorithm is very fast (one of the fastest clusteringalgorithms available), but it falls in local minima. That's why it can be useful to restartitseveral times.

Ifthealgorithmstopsbeforefullyconverging(because of`tol`or`max_iter`), `labels_` and`cluster_centers_`will notbe consistent, i.e. the`cluster_centers_`willnotbe themeansof thepointsin each cluster. Also,theestimator will reassign `labels_`after the last iteration to make `labels_`consistentwith`predict`on the trainingset.

Examples

```
>>>fromsklearn.clusterimportKMeans
>>>import numpyasnp
>>>X= np.array([[1,2],[1,4],[1, 0],
...               [10,2],[10,4],[10, 0]])
>>>kmeans=KMeans(n_clusters=2,random_state=0,n_init="auto").fit(X)
>>>kmeans.labels_
array([1,1,1,0,0,0],dtype=int32)
>>>kmeans.predict([[0,0],[12,3]])
array([1,0],dtype=int32)
>>>
kmeans.cluster_centers_arra
y([[10.,2.],
       [1.,2.]])
```

Methods

fit(X[,y,sample_weight])	Computek-meansclustering.
fit_predict(X[, y, sample_weight])	Computeclustercentersandpredictclusterindexforeach sample.
fit_transform(X[,y,sample_weight])	ComputeclusteringandtransformXtocluster-distancespace.
get_feature_names_out([input_features])	Getoutputfeaturenamesfortransformation.
get_metadata_routing()	Getmetadataroutingofthisobject.
get_params([deep])	Getparametersforthisestimator.
predict(X[,sample_weight])	PredicttheclosestclustereachsampleinXbelongsto.
score(X[,y,sample_weight])	Opposite of the value of X on the K-meansobjective.
set_fit_request(*[, sample_weight])	Requestmetadatapassed to the`fit`method.
set_output(*[,transform])	Setoutputcontainer.
set_params(**params)	Settheparametersofthisestimator.
set_predict_request(*[,sample_weight])	Requestmetadatapassed to the`predict`method.
set_score_request(*[, sample_weight])	Requestmetadatapassed to the`score`method.
transform(X)	TransformXtoacluster-distancespace.

fit(*X*, *y=None*, *sample_weight=None*)[source]Compute k-meansclustering.

Parameters:
X{array-like,sparsematrix}ofshape(n_samples,n_features)

Training instances to cluster. It must be noted that the data will be convertedto C ordering, which will cause a memory copy if the given data is not C-contiguous. If a sparse matrix is passed, a copy will be made if it's not in CSRformat.

y*Ignored*
Notused,presentthereforAPIconsistencybyconvention.

sample_weight*array-likeofshape(n_samples,),default=None*
The weights for each observation in X. If None, all observations are assignedequal weight. `sample_weight`is not used during initialization if `init`is acallableor a userprovided array.

Newinversion0.20.
Returns:
self*object*
Fittedestimator.

`fit_predict`(*X*,*y=None*,*sample_weight=None*)[source]
Compute cluster centers and predict cluster index for each

sample.Conveniencemethod;equivalenttocallingfit(X)followedbypredict(X).

Parameters:
X{array-like,sparsematrix}ofshape(n_samples,n_features)
Newdatatotransform.

y*Ignored*
Notused,presentthereforAPIconsistencybyconvention.

sample_weight*array-likeofshape(n_samples,),default=None*
The weights for each observation in X. If None, all observations are assignedequalweight.

Returns:
labels*ndarrayofshape(n_samples,)*
Indexoftheclustereachsamplebelongsto.

`fit_transform`(*X*,
y=None,*sample_weight=None*)[source]Computeclusteringandt ransformXtocluster-distancespace.

Equivalenttofit(X).transform(X),butmoreefficientlyimplemented.

Parameters:
X{array-like,sparsematrix}ofshape(n_samples,n_features)

Newdatatotransform.

y*Ignored*
Notused,presentthereforAPIconsistencybyconvention.

sample_weight*array-likeofshape(n_samples,),default=None*
The weights for each observation in X. If None, all observations are assignedequalweight.

Returns:
X_new*ndarrayofshape(n_samples,n_clusters)*
Xtransformedinthenew space.

`get_feature_names_out`(*input_features=None*)[source]
Getoutputfeaturenamesfortransformation.

The feature names out will prefixed by the lowercased class name. Forexample, if the transformer outputs 3 features, then the feature names outare:`["class_name0", "class_name1", "class_name2"]`.

Parameters:
input_features*array-likeofstrorNone,default=None*
Onlyusedtovalidate featurenameswiththenamesseenin`fit`.

Returns:
feature_names_out*ndarrayofstrobjects*
Transformedfeaturenames.

`get_metadata_routing`()[source]
Getmetadataroutingofthisobject.

PleasecheckUserGuideonhowtheroutingmechanismworks.

Returns:
routing*MetadataRequest*
A **`MetadataRequest`**encapsulatingroutinginformation.

`get_params`(*deep=True*)[source]
Getparametersforthisestimator.

Parameters:
deep*bool,default=True*
If True, will return the parameters for this estimator and contained subobjectsthatareestimators.

Returns:
params*dict*
Parameternamesmappedtotheirvalues.

predict(*X*,*sample_weight='deprecated'*)[source]
Predicttheclosestclustereachsampleinbelongsto.

In thevectorquantizationliterature,`cluster_centers_`iscalled the code bookand each value returned by `predict`is the index of the closest code in thecodebook.

Parameters:
X{array-like,sparsematrix}ofshape(n_samples,n_features)
Newdatatopredict.

sample_weight*array-likeofshape(n_samples,),default=None*
The weights for each observation in X. If None, all observations are assignedequalweight.

*Deprecatedsince version1.3:*The parameter`sample_weight`is deprecated inversion1.3 and willbe removed in1.5.
Returns:
labels*ndarrayofshape(n_samples,)*
Indexoftheclustereachsamplebelongsto.

score(*X*,*y=None*,*sample_weight=None*)[source]
Oppositeofthevalueof XontheK-meansobjective.

Parameters:
X{array-like,sparsematrix}ofshape(n_samples,n_features)
Newdata.

y*Ignored*
Notused,presentherefor APIconsistencybyconvention.

sample_weight*array-likeofshape(n_samples,),default=None*
The weights for each observation in X. If None, all observations are assignedequalweight.

Returns:
score*float*
Oppositeofthevalueof XontheK-meansobjective.

set_fit_request(*,*sample_weight:Union[bool,None,str]='$UNCHANGED$'*)→ KMeans[source]

Requestmetadatapassed to the`fit`method.

Notethatthismethodisonlyrelevant if`enable_metadata_routing=True`(see**sklearn.set_config**). PleaseseeUserGuide on how therouting mechanism works.

Theoptions foreachparameter are:

- `True`:metadataisrequested,andpassed to`fit`ifprovided.The requestisignoredif metadata isnot provided.
- `False`: metadata is not requested and the meta-estimator will not passit to`fit`.
- `None`: metadata is not requested, and the meta-estimator will raise anerrorifthe user provides it.
- `str`: metadata should be passed to the meta-estimator with this givenaliasinstead oftheoriginal name.

The default (`sklearn.utils.metadata_routing.UNCHANGED`) retains the existingrequest. This allows you to change the request for some parameters and notothers.

Newinversion1.3.

Note

This method is only relevant if this estimator is used as a sub-estimator of ameta-estimator,e.g.usedinsidea **Pipeline**.Otherwiseithasnoeffect.

Parameters:

sample_weight*str, True, False, or None,default=sklearn.utils.metadata_routing.UNCHANGED*

Metadataroutingfor`sample_weight`parameterin `fit`.

Returns:

self*object*

Theupdatedobject.

set_output(*,

transform=None)[source]Setoutpu t container.

SeeIntroducingtheset_outputAPIforanexampleonhowtouse theAPI.

Parameters:

transform*{"default","pandas"},default=None*

Configureoutputof`transform`and`fit_transform`.

- `"default"`:Defaultoutputformatof atransformer
- `"pandas"`:DataFrameoutput
- `None`:Transformconfigurationisunchanged

Returns:
self*estimatorinstance*
Estimatorinstance.

`set_params`(***params*)[source]
Settheparametersofthisestimator.

The method works on simple estimators as well as on nested objects (suchas**`Pipeline`**). The latter haveparameters of the form `<component><parameter>`sothat it'spossible to updateeach componentofanestedobject.

Parameters:
****params*dict***
Estimatorparameters.

Returns:
self*estimatorinstance*
Estimatorinstance.

`set_predict_request`(*, *sample_weight: Union[bool, None, str] = '$UNCHANGED$'*)→KMeans[source]
Requestmetadatapassed to the`predict`method.

Notethatthismethodisonlyrelevant if`enable_metadata_routing=True`(see**`sklearn.set_config`**). PleaseseeUserGuide on how therouting mechanism works.

Theoptions foreachparameter are:

- `True`:metadataisrequested,andpassedto`predict`ifprovided.Thereque stisignored ifmetadata isnotprovided.
- `False`: metadata is not requested and the meta-estimator will not passitto `predict`.
- `None`: metadata is not requested, and the meta-estimator will raise anerrorifthe user provides it.
- `str`: metadata should be passed to the meta-estimator with this givenaliasinstead oftheoriginal name.

The default (`sklearn.utils.metadata_routing.UNCHANGED`) retains the existingrequest. This allows you to change the request for some parameters and notothers.

Newinversion1.3.
Note

This method is only relevant if this estimator is used as a sub-estimator of ameta-estimator,e.g.usedinsidea **`Pipeline`**.Otherwiseithasnoeffect.
Parameters:
sample_weight*str, True, False, or None,default=sklearn.utils.metadata_routing.UNCHANGED*
Metadataroutingfor `sample_weight`parameter in `predict`.

Returns:
self*object*
Theupdatedobject.

`set_score_request`(**,sample_weight:Union[bool,None,str]='$UNCHANGED$'*) →KMeans[source]
Requestmetadatapassed to the`score`method.

Notethatthismethodisonlyrelevant if`enable_metadata_routing=True`(see**`sklearn.set_config`**). PleaseseeUserGuide on how therouting mechanism works.

Theoptions foreachparameter are:

- `True`:metadataisrequested,andpassedto`score`if provided.Therequestisignored ifmetadata isnotprovided.
- `False`: metadata is not requested and the meta-estimator will not passitto `score`.
- `None`: metadata is not requested, and the meta-estimator will raise anerrorifthe user provides it.
- `str`: metadata should be passed to the meta-estimator with this givenaliasinstead oftheoriginal name.

The default (`sklearn.utils.metadata_routing.UNCHANGED`) retains the existingrequest. This allows you to change the request for some parameters and notothers.

Newinversion1.3.
Note

This method is only relevant if this estimator is used as a sub-estimator of ameta-estimator,e.g.usedinsidea `Pipeline`.Otherwiseithasnoeffect.

Parameters:

sample_weight*str, True, False, or None,default=sklearn.utils.metadata_routing.UNCHANGED*

Metadataroutingfor`sample_weight`parameter in`score`.

Returns:

self*object*

Theupdatedobject.

`transform`(*X*)[source]

TransformXtoacluster-distancespace.

In the new space, each dimension is the distance to the cluster centers. NotethatevenifXis sparse,thearray returnedby`transform`will typically bedense.

Parameters:

X*{array-like,sparsematrix}ofshape(n_samples,n_features)*

Newdatatotransform.

Returns:

X_new*ndarrayofshape(n_samples,n_clusters)*

Xtransformedinthenew space.

- **FoliumDocumentation**

APIreference

Map

Makebeautiful,interactivemapswithPythonandLeaflet.js

class*folium.folium.GlobalSwitches(no_touch*=*False*,***disable_3d***=*False*)**

Bases:**`Element`**

class*folium.folium.Map(location:Sequence[float]|None*=*None*,***width:str|float***=*'100%'*,***height:str|float***=*'100%'*,***left:str|float***=*'0%'*,***top:str|float***=*'0%'*,***position:str***=*'relative'*,***tiles:str|TileLayer|None***=*'OpenStreetMap'*,***attr:str|None***=*None*,***min_zoom:int***=*0*,***max_zoom:int***=*18*,***zoom_start:int***= *10*,***min_lat:int*** *=-90*,***max_lat:int*** *=90*,***min_lon:int***=*-180*,***max_lon:int***=*180*,***max_bounds:bool***=*False*,***crs:str***=*'EPSG3857'*,***control_scale:bool***=*False*,***prefer_canvas:bool***=*False*,***no_touch:bool***=*False*,***disable_3d:bool***=*False*,***png_enabled:bool***=*False*,***zoom_control:bool***=*True*,*****kwargs:str|float|bool|Sequence|dict|None***)**

Bases:**JSCSSMixin**,**MacroElement**

CreateaMapwithFoliumand Leaflet.js

Generate a base map of given width and height with either default tilesets or a customtileset URL. The following tilesets are built-in to Folium. Pass any of the following tothe"tiles" keyword:

- "OpenStreetMap"
- "MapboxBright"(Limited levelsofzoom forfreetiles)
- "MapboxControlRoom"(Limited levels ofzoomforfreetiles)
- "Cloudmade"(Must passAPIkey)
- "Mapbox"(Must pass APIkey)
- "CartoDB"(positronanddark_matter)

Youcanpass acustomtileset toFoliumbypassing a**xyzservices.TileProvider**oraLeaflet-style URLtothetilesparameter:`http://{s}.yourtiles.com/{z}/{x}/{y}.png`.

You can find a list of free tile providers here: `http://leaflet-extras.github.io/leaflet-providers/preview/`. Be sure to check their termsandconditions and to provideattributionwith the*attr* keyword.

Parameters:

- **location** (*tuple or list, default None*) – Latitude and Longitude of Map(Northing,Easting).
- **width** (*pixelintorpercentagestring (default:'100%')*) –Widthof themap.
- **height**(*pixelintor percentagestring (default:'100%')*) –Heightofthemap.
- **tiles** (str or TileLayer or **xyzservices.TileProvider**, default'OpenStreetMap') – Map tileset to use. Can choose from a list of built-in tiles,pass a **xyzservices.TileProvider**, pass a custom URL, pass a TileLayerobject, or pass *None* to create a map without tiles. For more advanced tilelayeroptions, use the*TileLayer*class.
- **min_zoom** (*int, default 0*) – Minimum allowed zoom level for the tile layerthatis created.
- **max_zoom** (*int, default 18*) – Maximum allowed zoom level for the tile layerthatis created.
- **zoom_start**(*int,default10*)– Initialzoomlevelfor themap.
- **attr** (*string, default None*) – Map tile attribution; only required if passingcustomtileURL.
- **crs** (*str, default 'EPSG3857'*) – Defines coordinate reference systems forprojecting geographical points into pixel (screen) coordinates and back. Youcan use Leaflet's values : * EPSG3857 : The most common CRS for onlinemaps, used by almost all free and commercial tile providers. Uses SphericalMercator projection. Set in by default in Map's crs option. * EPSG4326 : Acommon CRS among GIS enthusiasts. Uses simple Equirectangularprojection. * EPSG3395 : Rarely used by some commercial tile providers.Uses Elliptical Mercator projection. * Simple : A simple CRS that mapslongitude andlatitude intox andydirectly. Maybe usedformaps offlat

surfaces (e.g. game maps). Note that the y axis should still be inverted (goingfrombottom totop).

- **control_scale**(*bool,defaultFalse*)–Whethertoaddacontrol scaleonthemap.
- **prefer_canvas** (*bool, default False*) – Forces Leaflet to use the Canvas back-end (if available) for vector layers instead of SVG. This can increaseperformance considerably in some cases (e.g. many thousands of circlemarkerson the map).
- **no_touch** (*bool, default False*) – Forces Leaflet to not use touch events even ifitdetects them.
- **disable_3d** (*bool, default False*) – Forces Leaflet to not use hardware-accelerated CSS 3D transforms for positioning (which may cause glitches insomerare environments)even ifthey're supported.
- **zoom_control**(*bool,defaultTrue*)–Display zoomcontrolsonthemap.
- ****kwargs** – Additional keyword arguments are passed to Leaflets Mapclass:https://leafletjs.com/reference.html#map

Returntype:

FoliumMapObject

Examples

```
>>> m = folium.Map(location=[45.523, -122.675],
width=750,height=500)
>>> m = folium.Map(location=[45.523, -122.675],
tiles="cartodbpositron")
>>>m=folium.Map(
...     location=[45.523,-122.675],
...     zoom_start=2,
...
tiles="https://api.mapbox.com/v4/mapbox.streets/{z}/{x}/{y}.png?access
_token=mytoken",
...     attr="Mapboxattribution",
...)
```

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

fit_bounds(*bounds: Sequence[Sequence[float]], padding_top_left: Sequence[float] | None=None,padding_bottom_right:Sequence[float]|None=None,padding:Sequence[float] |None=None,max_zoom:int|None=None***)→None**

Fitthemaptocontainaboundingboxwiththemaximumzoomlevelpossible.

Parameters:

- **bounds** (*list of (latitude, longitude) points*) – Bounding box specified as twopoints[southwest, northeast]

- **padding_top_left** (*(x, y) point, default None*) – Padding in the top left corner.Useful if some elements in the corner, such as controls, might obscure objectsyou'rezooming to.
- **padding_bottom_right** (*(x, y) point, default None*) – Padding in the bottomrightcorner.
- **padding** (*(x, y) point, default None*) – Equivalent to setting both top left andbottomrightpaddingto the same value.
- **max_zoom**(*int, defaultNone*)– Maximumzoomto be used.

Examples

```
>>>m.fit_bounds([[52.193636,-2.221575],[52.636878,-
1.139759]])
```

keep_in_front(args:Layer*)→ None**

Pass one or multiple layers that must stay in

front.Theordering matters,thelastone isput on top.

Parameters:

***args** – Variable length argument list. Any folium object that counts as an overlay.For example FeatureGroup or TileLayer. Does not work with markers, for those usez_index_offset.

render(*kwargs*)→None**

RendrestheHTMLrepresentationoftheelement.

show_in_browser()→None

DisplaytheMapinthedefaultwebbrowser.

UIelements

Classesfor drawingmaps.

***class*folium.map.CustomPane(*name:str,z_index:int|str=625,pointer_events:bool=False*)**

Bases:**MacroElement**

Createsacustompane toholdmapelements.

Behaviorisasinhttps://leafletjs.com/examples/map-panes/

Parameters:

- **name** (*string*) – Name of the custom pane. Other map elements can be addedtothepane byspecifyingthe 'pane'kwargwhenconstructing them.
- **z_index** (*int or string, default 625*) – The z-index that will be associated withthe pane, and will determine which map elements lie over/under it. The default(625) corresponds to between markers and tooltips. Default panes and z-indexescanbefoundathttps://leafletjs.com/reference.html#map-pane

- **pointer_events** (*bool, default False*) – Whether or not layers in the paneshouldinteractwiththecursor.SettingtoFalsewillpreventinterferingwithpointerevents associated withlower layers.

***class*folium.map.FeatureGroup(*name:str|None=None,overlay:bool=True,control:bool=True,show:bool =True,**kwargs:str|float|bool |Sequence |dict|None*)**

Bases:**Layer**

Create a FeatureGroup layer ; you can put things in it and handle them as a singlelayer.Forexample,youcanadd aLayerControltotick/untick thewholegroup.

Parameters:

- **name** (*str, default None*) – The name of the featureGroup layer. It will bedisplayed in the LayerControl. If None get_name() will be called to get thetechnical(ugly) name.
- **overlay** (*bool, default True*) – Whether your layer will be an overlay (tickedwith a check box in LayerControls) or a base layer (ticked with a radiobutton).
- **control** (*bool, default True*) – Whether the layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- ****kwargs**–Additional(possiblyinherited)options. Seehttps://leafletjs.com/reference.html#featuregroup

***class* folium.map.FitBounds(*bounds: Sequence[Sequence[float]], padding_top_left: Sequence[float]|None=None,padding_bottom_right:Sequence[float]|None=None,padding:Sequence[float] |None = None,max_zoom:int |None= None*)**

Bases:**MacroElement**

Fitthemaptocontainaboundingboxwiththemaximumzoomlevelpossible.

Parameters:

- **bounds** (*list of (latitude, longitude) points*) – Bounding box specified as twopoints[southwest, northeast]
- **padding_top_left** (*(x, y) point, default None*) – Padding in the top left corner.Useful if some elements in the corner, such as controls, might obscure objectsyou'rezooming to.
- **padding_bottom_right** (*(x, y) point, default None*) – Padding in the bottomrightcorner.
- **padding** (*(x, y) point, default None*) – Equivalent to setting both top left andbottomrightpaddingto the same value.
- **max_zoom**(*int, defaultNone*)– Maximumzoomto be used.

***class*folium.map.FitOverlays(*padding:int=0,max_zoom:int|None=None,fly:bool=False,fit_on_map_load:bool=True*)**

Bases:**MacroElement**

Fittheboundsofthemapstothe enabledoverlays.

Parameters:

- **padding**(*int,default0*)–Amountofpaddinginpixelsappliedinthe corners.
- **max_zoom** (*int, optional*) – The maximum possible zoom to use when fittingtothebounds.
- **fly**(*bool,defaultFalse*) –Useasmoother,longeranimation.
- **fit_on_map_load** (*bool, default True*) – Apply the fit when initially loadingthemap.

***class*folium.map.Icon(*color:str='blue',icon_color:str='white',icon:str='info-sign',angle:int=0,prefix:str='glyphicon',**kwargs:str|float|bool|Sequence|dict|None*)**

Bases:**MacroElement**

CreatesanIconobjectthatwillberenderedusing Leaflet.awesome-markers.

Parameters:

- **color**(*str, default'blue'*) –

  Thecolor ofthe marker. Youcan use:
  ['red','blue','green','purple','orange','darkred',

'lightred', 'beige', 'darkblue', 'darkgreen', 'cadetblue', 'darkpurple', 'white', 'pink','lightblue','lightgreen', 'gray', 'black', 'lightgray']

- **icon_color** (*str, default 'white'*) – The color of the drawing on the marker. Youcanuse colors above, or an htmlcolorcode.
- **icon** (*str, default 'info-sign'*) – The name of the marker sign. See Font-Awesome website to choose yours. Warning : depending on the icon youchooseyou may need to adaptthe *prefix*as well.
- **angle**(*int,default0*)– Theiconwillberotated by thisamountofdegrees.
- **prefix** (*str, default 'glyphicon'*) – The prefix states the source of the icon. 'fa'forfont-awesomeor'glyphicon' for bootstrap 3.
- **https**(*//github.com/lvoogdt/Leaflet.awesome-markers*)–

color_options

***class*folium.map.Layer(*name:str|None=None,overlay:bool=False,control:bool=True,show:bool=True*)**

Bases:**MacroElement**

An abstract class for everything that is a Layer on the map. It will be used to definewhetheran objectwillbe includedinLayerControls.

Parameters:

- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls

- **overlay** (*bool, default False*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

render(*kwargs*)**

RenderstheHTMLrepresentationoftheelement.

***class*folium.map.LayerControl(*position:str='topright',collapsed:bool=True,autoZIndex:bool=True,draggable:bool=False,**kwargs:str|float|bool|Sequence|dict|None*)**

Bases:**MacroElement**

CreatesaLayerControlobjecttobeaddedonafoliummap.

This object should be added to a Map object. Only Layer children of Map areincluded inthe layercontrol.

Note

The LayerControl should be added last to the map. Otherwise, the LayerControland/orthecontrolledlayers may not appear.

Parameters:

- **position** (*str*) – The position of the control (one of the map corners), can be‘topleft’,‘topright’,‘bottomleft’or‘bottomright’ default:‘topright’
- **collapsed** (*bool, default True*) – If true the control will be collapsed into aniconand expanded on mouse hover or touch.
- **autoZIndex** (*bool, default True*) – If true the control assigns zIndexes inincreasing order to all of its layers so that the order is preserved whenswitchingthemon/off.
- **draggable** (*bool, default False*) – By default the layer control has a fixedposition.Setthisargumentto Truetoallowdraggingthecontrolaround.
- ****kwargs**–Additional(possiblyinherited)options. Seehttps://leafletjs.com/reference.html#control-layers

render(*kwargs*)→None**

RenderstheHTMLrepresentationoftheelement.

reset()→None

***class*folium.map.Marker(*location:Sequence[float]|None=None,popup:Popup|str|None=None, tooltip: Tooltip| str | None = None, icon: Icon| None = None, draggable: bool = False, **kwargs:str |float |bool |Sequence |dict|None*)**

Bases:**MacroElement**

Create a simple stock Leaflet marker on the map, with optional popup text or Vincentvisualization.

Parameters:

- **location**(*tupleorlist*)–LatitudeandLongitudeofMarker(Northing,Easting)
- **popup** (*string or folium.Popup, default None*) – Label for the Marker; eitheran escaped HTML string to initialize folium.Popup or a folium.Popupinstance.
- **tooltip** (*str or folium.Tooltip, default None*) – Display a text when hoveringoverthe object.
- **icon**(*Iconplugin*)–the Iconplugintouse torenderthemarker.
- **draggable**(*bool,defaultFalse*)– SettoTruetobeabletodragthemarkeraroundthe map.

Returntype:

MarkernamesandHTMLinobj.template_vars

Examples

```
>>>Marker(location=[45.5,-122.3],popup="Portland,OR")
>>> Marker(location=[45.5, -122.3], popup=Popup("Portland,
OR"))# If the popup label has characters that need to be escaped
inHTML
>>>Marker(
...     location=[45.5,-122.3],
...     popup=Popup("Mom&PopArrowShop>>",parse_html=True),
...)
```

render()→None

RenderstheHTMLrepresentationoftheelement.

***class*folium.map.Popup(*html:str|Element|None=None,parse_html:bool=False,max_width:str|int='100%',show:bool=False,sticky:bool=False,lazy:bool=False,**kwargs:str|float|bool|Sequence|dict|None*)**

Bases:**Element**

CreateaPopupinstancethatcanbe linkedto aLayer.

Parameters:

- **html**(*string orElement*) –Contentofthe Popup.
- **parse_html** (*bool, default False*) – True if the popup is a template that needstothe rendered first.
- **max_width** (*int for pixels or text for percentages, default '100%'*) – Themaximalwidth ofthe popup.
- **show** (*bool,default False*) –Truerendersthe popupopen on pageload.
- **sticky** (*bool, default False*) – True prevents map and other popup clicks fromclosing.
- **lazy** (*bool, default False*) – True only loads the Popup content when clickingonthe Marker.

render(*kwargs*)→None**

RenderstheHTMLrepresentationoftheelement.

class*folium.map.Tooltip(*text:str,style:str|None*=*None*,*sticky:bool*=*True*,kwargs:str|float| bool |Sequence |dict|None*)**

Bases:**MacroElement**

Createa tooltipthat showstextwhenhoveringover itsparentobject.

Parameters:

- **text** (*str*) – String to display as a tooltip on the object. If the argument is of adifferenttype itwill be convertedtostr.
- **style** (*str, default None.*) – HTML inline style properties like font and colors.Willbe applied to adivwith thetext init.
- **sticky**(*bool,defaultTrue*)–Whetherthetooltipshouldfollowthemouse.
- ****kwargs** – These values will map directly to the Leaflet Options. More infoavailablehere:https://leafletjs.com/reference.html#tooltip

parse_options(*kwargs:Dict[str,str|float|bool|Sequence|dict|None]*)→Dict[str,str|float |bool |Sequence|dict |None]

Validatetheprovidedkwargsandreturnoptionsasjsonstring.

valid_options*:Dict[str,Tuple[Type,...]]*

RasterLayers

Wraps leaflet TileLayer, WmsTileLayer (TileLayer.WMS), ImageOverlay, andVideoOverlay

class* folium.raster_layers.ImageOverlay(*image: Any, bounds: Sequence[Sequence[float]], origin:str = 'upper', colormap: Callable | None = None, mercator_project: bool = False, pixelated: bool = True,name:str|None*=*None*,*overlay:bool*=*True*,*control:bool*=*True*,*show:bool*=*True*,kwargs*)**

Bases:**Layer**

Used to load and display a single image over specific bounds of the map, implementsILayerinterface.

Parameters:

- **image** (*string, file or array-like object*) – The data you want to draw on themap. * If string, it will be written directly in the output file. * If file, it'scontent will be converted as embedded in the output file. * If array-like, it willbeconvertedtoPNG base64 stringand embedded in theoutput.
- **bounds**(*list*)–

  Imageboundsonthemapintheform

[[lat_min,lon_min],[lat_max,lon_max]]

- **opacity**(*float,defaultLeaflet'sdefault(1.0)*)–

- **alt**(*string,defaultLeaflet'sdefault('')*) –
- **origin** (*['upper' | 'lower'], optional, default 'upper'*) – Place the [0,0] index ofthearray inthe upperleft or lower leftcorner of the axes.
- **colormap** (callable, used only for *mono* image.) – Function of the form [x -->(r,g,b)] or [x -> (r,g,b,a)] for transforming a mono image into RGB. It mustoutput iterables of length 3 or 4, with values between 0 and 1. Hint: you canusecolormaps from*matplotlib.cm*.
- **mercator_project** (*bool, default False.*) – Used only for array-like image.Transforms the data to project (longitude, latitude) coordinates to the Mercatorprojection. Beware that this will only work if *image* is an array-like object.Notethat ifused the imagewillbeclipped beyondlatitude-85 and 85.
- **pixelated** (*bool, default True*) – Sharp sharp/crips (True) or aliased corners(False).
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- **https** (*See*)–
- **options.**–

render(*kwargs*)→None**

RenderstheHTMLrepresentationoftheelement.

class*folium.raster_layers.TileLayer(*tiles:str|xyzservices.TileProvider='OpenStreetMap',min_zoom : int = 0, max_zoom: int = 18, max_native_zoom: int | None = None, attr: str | None = None, detect_retina: bool = False, name: str | None = None, overlay: bool = False, control: bool = True, show:bool=True,no_wrap:bool=False,subdomains:str='abc',tms:bool=False,opacity:float=1,**kwargs)

Bases:**`Layer`**

Createa tilelayerto appendonaMap.

Parameters:

- **tiles**(stror**`xyzservices.TileProvider`**,default‘OpenStreetMap’)–

  Maptilesettouse.Canchoosefromthislistofbuilt-intiles:

  - ”OpenStreetMap”
  - ”CartoDBpositron”, “CartoDBdark_matter”

  Youcanpass acustomtileset toFoliumbypassing a**`xyzservices.TileProvider`**oraLeaflet-style URLtothetilesparameter:`http://{s}.yourtiles.com/{z}/{x}/{y}.png`.

You can find a list of free tile providers here: `http://leaflet-extras.github.io/leaflet-providers/preview/`. Be sure to checktheirtermsandconditionsandtoprovide attributionwiththe*attr* keyword.

- **min_zoom**(*int,default0*) – Minimumallowed zoomlevelforthistile layer.
- **max_zoom**(*int,default18*)–Maximumallowedzoomlevelforthistilelayer.
- **max_native_zoom** (*int, default None*) – The highest zoom level at which thetile server can provide tiles. If provided you can zoom in past this level. Elsetileswill turn grey.
- **attr** (*string, default None*) – Map tile attribution; only required if passingcustomtileURL.
- **detect_retina** (*bool, default False*) – If true and user is on a retina display, itwill request four tiles of half the specified size and a bigger zoom level inplaceof one toutilize the highresolution.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default False*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show** (*bool, default True*) – Whether the layer will be shown on opening.When adding multiple base layers, use this parameter to select which oneshouldbe shownwhen opening the map,by notshowing the others.
- **subdomains**(*listofstrings,default ['abc']*)–Subdomainsofthetileservice.
- **tms** (*bool, default False*) – If true, inverses Y axis numbering for tiles (turnthison for TMS services).
- **opacity**(*float,default1*)–Setstheopacityfor thelayer.
- ****kwargs** (*additional keyword arguments*) – Other keyword arguments arepassedas options to theLeaflettileLayerobject.

class **folium.raster_layers.VideoOverlay(*video_url: str, bounds: Sequence[Sequence[float]], autoplay:bool=True,loop:bool=True,name:str|None=None,overlay:bool=True,control:bool=True,show:bool =True,**kwargs:str|float|bool|Sequence |dict |None*)**

Bases:**`Layer`**

Usedtoloadand displayavideo overthe map.

Parameters:

- **video_url**(*str*) – URLof thevideo
- **bounds**(*list*)–

Videoboundsonthemapintheform

[[lat_min,lon_min],[lat_max,lon_max]]

- **autoplay**(*bool,defaultTrue*)–
- **loop**(*bool,defaultTrue*)–
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls

- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- ****kwargs**–Othervalid(possiblyinherited)options. See:https://leafletjs.com/reference.html#videooverlay

***class*folium.raster_layers.WmsTileLayer(*url:str,layers:str,styles:str=',fmt:str='image/jpeg',transparent:bool=False,version:str='1.1.1',attr:str=',name:str|None=None,overlay:bool=True,control:bool=True,show:bool=True,**kwargs*)**

Bases:**Layer**

CreatesaWebMap Service(WMS)layer.

Parameters:

- **url**(*str*) –Theurl of the WMS server.
- **layers**(*str*)–Comma-separated listofWMSlayerstoshow.
- **styles**(*str,optional*) –Comma-separatedlistofWMSstyles.
- **fmt** (*str, default 'image/jpeg'*) – The format of the service output. Ex:'image/png'
- **transparent** (*bool, default False*) – Whether the layer shall allowtransparency.
- **version**(*str,default'1.1.1'*)– Versionof theWMS serviceto use.
- **attr** (*str, default ''*) – The attribution of the service. Will be displayed in thebottomrightcorner.
- **name** (*string, optional*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- ****kwargs** (*additional keyword arguments*) – Passed through to the underlyingtileLayer.wms object and can be used for setting extra tileLayer.wmsparametersor as extra parametersin theWMS request.
- **https** (*See*)–

VectorLayers

WrapsleafletPolyline,Polygon,Rectangle,Circle,andCircleMarker

***class*folium.vector_layers.Circle(*location:Sequence[float]|None=None,radius:float=50,popup:Popup| str | None = None, tooltip: Tooltip| str | None = None, **kwargs: bool | str | float | None*)**

Bases:**Marker**

Classfordrawing circleoverlayson amap.

It's an approximation and starts to diverge from a real circle closer to the poles (due toprojectiondistortion).

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **location**(*tuple[float, float]*)–LatitudeandLongitudepair(Northing,Easting)
- **popup** (*string or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, default None*) – Display a text when hoveringoverthe object.
- **radius**(*float*)–Radiusofthecircle,inmeters.
- ****kwargs**–Othervalid(possiblyinherited)options. See:https://leafletjs.com/reference.html#circle

class **folium.vector_layers.CircleMarker(*location: Sequence[float] | None = None, radius: float =10,popup:Popup|str|None=None,tooltip:Tooltip|str|None=None,**kwargs:bool|str|float|None*)**

Bases:**Marker**

Acircleofa fixed sizewith radius specifiedin pixels.

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **location**(*tuple[float, float]*)–LatitudeandLongitudepair(Northing,Easting)
- **popup** (*string or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, default None*) – Display a text when hoveringoverthe object.
- **radius**(*float,default10*)–Radiusofthe circlemarker, inpixels.
- ****kwargs**–Othervalid(possiblyinherited)options. See:https://leafletjs.com/reference.html#circlemarker

class**folium.vector_layers.PolyLine(*locations,popup=None,tooltip=None,**kwargs*)**

Bases:**BaseMultiLocation**

Drawpolylineoverlaysonamap.

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **locations** (*list of points (latitude, longitude)*) – Latitude and Longitude of line(Northing, Easting) Pass multiple sequences of coordinates for a multi-polyline.

- **popup** (*str or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, default None*) – Display a text when hoveringoverthe object.
- **smooth_factor** (*float, default 1.0*) – How much to simplify the polyline oneach zoom level. More means better performance and smoother look, and lessmeansmore accuraterepresentation.
- **no_clip**(*Bool,defaultFalse*)–Disablepolyline clipping.
- ****kwargs**–Othervalid(possiblyinherited)options. See:https://leafletjs.com/reference.html#polyline

***class* folium.vector_layers.Polygon(*locations: Iterable[Sequence[float]] | Iterable[Iterable[Sequence[float]]],popup:Popup|str|None=None,tooltip:Tooltip|str|None=None,**kwargs:bool|str|float |None*)**

Bases:**BaseMultiLocation**

Drawpolygonoverlays onamap.

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **locations**(*listofpoints (latitude,longitude)*)–
  - One list of coordinate pairs to define a polygon. You don't have to addalast pointequaltothe firstpoint.
  - Ifyoupass alistwithmultipleof those it willmake amulti-polygon.
- **popup** (*string or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, default None*) – Display a text when hoveringoverthe object.
- ****kwargs**–Othervalid(possiblyinherited)options. See:https://leafletjs.com/reference.html#polygon

***class*folium.vector_layers.Rectangle(*bounds:Iterable[Sequence[float]],popup:Popup|str|None=None,tooltip:Tooltip|str|None=None,**kwargs:bool|str|float|None*)**

Bases:**MacroElement**

Drawrectangleoverlaysonamap.

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **bounds** (*[(lat1, lon1), (lat2, lon2)]*) – Two lat lon pairs marking the twocornersof therectangle.
- **popup** (*string or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, default None*) – Display a text when hoveringoverthe object.

- ****kwargs**–Othervalid(possiblyinherited)options. See:https://leafletjs.com/reference.html#rectangle

folium.vector_layers.path_options(*line:bool=False,radius:float|None=None,kwargs:bool| str |float|None*)**

Contains options and constants shared between vector overlays (Polygon, Polyline,Circle,CircleMarker, and Rectangle).

Parameters:

- **stroke** (*Bool, True*) – Whether to draw stroke along the path. Set it to false todisableborders on polygons or circles.
- **color**(*str,'#3388ff'*)–Strokecolor.
- **weight**(*int,3*)–Strokewidthin pixels.
- **opacity**(*float,1.0*)–Strokeopacity.
- **line_cap** (*str, 'round' (lineCap)*) – A string that defines shape to be used at theend of the stroke. https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-linecap
- **line_join** (*str, 'round' (lineJoin)*) – A string that defines shape to be used at thecorners of the stroke. https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-linejoin
- **dash_array** (*str, None (dashArray)*) – A string that defines the stroke dashpattern.Doesn't work onCanvas-powered layers insome old browsers. https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray
- **dash_offset**(*,str,None(dashOffset)*)–A stringthatdefinesthedistanceintothe dash pattern to start the dash. Doesn't work on Canvas-powered layers insomeold browsers.https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dashoffset
- **fill** (*Bool, False*) – Whether to fill the path with color. Set it to false to disablefillingon polygons or circles.
- **fill_color** (str, default to *color* (fillColor)) – Fill color. Defaults to the value ofthecolor option.
- **fill_opacity**(*float,0.2(fillOpacity)*)–Fillopacity.
- **fill_rule** (*str, 'evenodd' (fillRule)*) – A string that defines how the inside of ashapeis determined.https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/fill-rule
- **bubbling_mouse_events** (*Bool, True (bubblingMouseEvents)*) – When true amouse event on this path will trigger the same event on the map (unlessL.DomEvent.stopPropagation is used).
- **gradient** (*bool, default None*) – When a gradient on the stroke and fill isavailable,allowsturning iton or off.
- **fill=False.**(*Notethatthepresenceoffill_colorwilloverride*)–
- **equivalents.**(*Thisfunctionacceptsboth snake_caseandlowerCamelCase*)–
- **https** (*See*)–

Othermapfeatures

LeafletGeoJsonandmiscellaneousfeatures.

class **folium.features.Choropleth(*geo_data: Any, data: Any | None = None, columns: Sequence[Any]|None=None,key_on:str|None=None,bins:int|Sequence[float]=6,fill_color:str|None =None, nan_fill_color: str = 'black', fill_opacity: float = 0.6, nan_fill_color_opacity: float | None = None, line_color:str='black',line_weight:float=1,line_opacity:float=1,name:str|None=None,legend _name: str = '', overlay: bool = True, control: bool = True, show: bool = True, topojson: str | None =None,smooth_factor:float|None=None,highlight:bool=False,use_jenks:bool=False,**kwargs*)**

Bases:**FeatureGroup**

ApplyaGeoJSON overlayto themap.

Plot a GeoJSON overlay on the base map. There is no requirement to bind data(passing just a GeoJSON plots a single-color overlay), but there is a data bindingoptiontomapyourcolumnardatatodifferentfeatureobjects withacolorscale.

If data is passed as a Pandas DataFrame, the "columns" and "key-on" keywords mustbe included, the first to indicate which DataFrame columns to use, the second toindicate the layer in the GeoJSON on which to key the data. The 'columns' keyworddoesnot need to be passed for aPandas series.

Colors are generated from color brewer (https://colorbrewer2.org/) sequential palettes.By default, linear binning is used between the min and the max of the values. Custombinningcan beachieved with the *bins* parameter.

TopoJSONs can be passed as "geo_data", but the "topojson" keyword must also bepassed with the reference to the topojson objects to convert. See the topojson.featuremethodinthe TopoJSON APIreference:topojson/topojson

Parameters:

- **geo_data** (*string/object*) – URL, file path, or data (json, dict, geopandas, etc)toyour GeoJSON geometries
- **data** (*Pandas DataFrame or Series, default None*) – Data to bind to theGeoJSON.
- **columns** (*tuple with two values, default None*) – If the data is a PandasDataFrame,thecolumnsofdatatobebound.Mustpasscolumn1asthekey,a ndcolumn 2the values.
- **key_on** (*string, default None*) – Variable in the *geo_data* GeoJSON file tobind the data to. Must start with 'feature' and be in JavaScript objectionnotation. Ex:'feature.id'or 'feature.properties.statename'.
- **bins** (*int or sequence of scalars or str, default 6*) – If *bins* is an int, it definesthenumberofequal-widthbins between theminand the max ofthevalues.
  If *bins* is a sequence, it directly defines the bin edges. For more information onthisparameter, havea lookatnumpy.histogramfunction.
- **fill_color** (*string, optional*) – Area fill color, defaults to blue. Can pass a hexcode, color name, or if you are binding data, one of the following color brewerpalettes: 'BuGn', 'BuPu', 'GnBu', 'OrRd', 'PuBu', 'PuBuGn', 'PuRd','RdPu','YlGn', 'YlGnBu', 'YlOrBr',and 'YlOrRd'.

- **nan_fill_color** (*string, default 'black'*) – Area fill color for nan or missingvalues.Can pass a hex code, color name.
- **fill_opacity**(*float,default0.6*)–Areafillopacity,range0-1.
- **nan_fill_opacity** (*float, default fill_opacity*) – Area fill opacity for nan ormissingvalues, range 0-1.
- **line_color**(*string,default'black'*)–GeoJSONgeopathlinecolor.
- **line_weight**(*int,default1*)–GeoJSONgeopathlineweight.
- **line_opacity**(*float, default1*)–GeoJSONgeopathlineopacity,range0-1.
- **legend_name**(*string,default emptystring*)–Titlefordata legend.
- **topojson** (*string, default None*) – If using a TopoJSON, passing"objects.yourfeature" to the topojson keyword argument will enableconversionto GeoJSON.
- **smooth_factor** (*float, default None*) – How much to simplify the polyline oneach zoom level. More means better performance and smoother look, and lessmeansmore accuraterepresentation.Leafletdefaults to1.0.
- **highlight** (*boolean, default False*) – Enable highlight functionality whenhoveringover aGeoJSONarea.
- **use_jenks** (*bool, default False*) – Use jenkspy to calculate bins using "naturalbreaks" (Fisher-Jenks algorithm). This is useful when your data is unevenlydistributed.
- **name** (*string, optional*) – The name of the layer, as it will appear inLayerControls
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

Returntype:

GeoJSONdatalayerinobj.template_vars

Examples

```
>>> Choropleth(geo_data="us-states.json",
line_color="blue",line_weight=3)
>>>Choropleth(
...     geo_data="geo.json",
...     data=df,
...     columns=["Data1","Data2"],
...     key_on="feature.properties.myvalue",
...     fill_color="PuBu",
...     bins=[0,20,30,40,50,60],
...)
>>>
Choropleth(geo_data="countries.json",topo
json="objects.countries")
>>>Choropleth(
...     geo_data="geo.json",
...     data=df,
...     columns=["Data1","Data2"],
...     key_on="feature.properties.myvalue",
...     fill_color="PuBu",
```

```
...     bins=[0,20,30,40,50,60],
...     highlight=True,
...)
```

render(*kwargs*)→None**

RendertheGeoJson/TopoJsonandcolorscaleobjects.

***class*folium.features.ClickForLatLng(*format_str:str|None*=*None*,*alert:bool*=*True*)**

Bases:**MacroElement**

When one clicks on a Map that contains a ClickForLatLng, the coordinates of thepointer'spositionare copied to clipboard.

Parameters:

- **format_str** (*str, default 'lat + "," + lng'*) – The javascript string used to formatthe text copied to clipboard. eg: format_str = 'lat + "," + lng' >>46.558860,3.397397 format_str = '"[" + lat + "," + lng + "]"' >>[46.558860,3.397397]
- **alert** (*bool, default True*) – Whether there should be an alert when somethinghasbeen copiedtoclipboard.

***class*folium.features.ClickForMarker(*popup:IFrame|Html|str|None*=*None*)**

Bases:**MacroElement**

When one clicks on a Map that contains a ClickForMarker, a Marker is created at thepointer'sposition.

Parameters:

popup (*str or IFrame or Html, default None*) – Text to display in the markers'popups. This can also be an Element like IFrame or Html. If None, the popups willdisplay the marker's latitude and longitude. You can include the latitude andlongitudewith${lat}and ${lng}.

Examples

```
>>> ClickForMarker("<b>Lat:</b> ${lat}<br /><b>Lon:</b>
${lng}")
```

class* folium.features.ColorLine(*positions: Iterable[Sequence[float]], colors: Iterable[float], colormap:ColorMap|Sequence[Any]|None*=*None*,*nb_steps:int*=*12*,*weight:int|None*=*None*,*opacity: float |None* = *None*,kwargs:Any*)**

Bases:**FeatureGroup**

Drawdataonamap withspecified colors.

Parameters:

- **positions** (*iterable of (lat, lon) pairs*) – The points on the line. Segmentsbetweenpoints will be colored.

- **colors** (*iterable of float*) – Values that determine the color of a line segment. Itmusthavelength equalto*len(positions)-1*.
- **colormap** (*branca.colormap.Colormap or list or tuple*) – The colormap touse. If a list or tuple of colors is provided, a LinearColormap will be createdfromit.
- **nb_steps** (*int, default 12*) – The colormap will be discretized to this number ofcolors.
- **opacity**(*float,default1*)–Lineopacity,scale0-1
- **weight**(*int,default2*)–Strokeweightinpixels
- ****kwargs**–Furtherparametersavailable.Seefolium.map.FeatureGroup

Returntype:

AColorLine objectthatyoucan*add_to* aMap.

***class*folium.features.CustomIcon(*icon_image:Any,icon_size:Tuple[int,int]|None*=*None*,*icon_anchor: Tuple[int, int] | None* = *None*, *shadow_image: Any* = *None*, *shadow_size: Tuple[int, int] |None*=*None*,*shadow_anchor:Tuple[int,int]|None*=*None*,*popup_anchor:Tuple[int,int]|None* = *None*)**

Bases:**Icon**

Createacustomicon, based onan image.

Parameters:

- **icon_image** (*string, file or array-like object*) – The data you want to use as anicon. * If string, it will be written directly in the output file. * If file, it'scontent will be converted as embedded in the output file. * If array-like, it willbeconvertedtoPNGbase64 string andembedded in the output.
- **icon_size**(*tupleof2 int,optional*)–Sizeoftheicon imageinpixels.
- **icon_anchor** (*tuple of 2 int, optional*) – The coordinates of the "tip" of theicon(relative toitstopleftcorner). Theiconwillbealignedsothatthispointisatthemarker's geographicallocation.
- **shadow_image** (*string, file or array-like object, optional*) – The data for theshadowimage. Ifnotspecified, noshadow imagewillbe created.
- **shadow_size**(*tupleof2 int,optional*)–Sizeoftheshadow imageinpixels.
- **shadow_anchor** (*tuple of 2 int, optional*) – The coordinates of the "tip" of theshadow relative to its top left corner (the same as icon_anchor if notspecified).
- **popup_anchor** (*tuple of 2 int, optional*) – The coordinates of the point fromwhichpopups will"open",relativetothe icon anchor.

***class*folium.features.DivIcon(*html:str|None*=*None*,*icon_size:Tuple[int,int]|None*=*None*,*icon_anchor: Tuple[int, int] | None* = *None*, *popup_anchor: Tuple[int, int] | None* = *None*, *class_name:str* = *'empty'*)**

Bases:**MacroElement**

Represents a lightweight icon for markers that uses a simple *div* element instead of animage.

Parameters:

- **icon_size**(*tupleof2 int*)–Sizeof theiconimageinpixels.
- **icon_anchor** (*tuple of 2 int*) – The coordinates of the "tip" of the icon (relativeto its top left corner). The icon will be aligned so that this point is at themarker'sgeographicallocation.
- **popup_anchor** (*tuple of 2 int*) – The coordinates of the point from whichpopupswill"open",relative to theiconanchor.
- **class_name** (*string*) – A custom class name to assign to the icon. Leafletdefaults is 'leaflet-div-icon' which draws a little white square with a shadow.Wesetit 'empty'infolium.
- **html**(*string*)–AcustomHTMLcodeto putinsidethedivelement.
- **https** (*See*)–

class **folium.features.GeoJson(*data: Any, style_function: Callable | None = None, highlight_function: Callable | None = None, name: str | None = None, overlay: bool = True, control: bool = True, show: bool = True, smooth_factor: float | None = None, tooltip: str | Tooltip| GeoJsonTooltip| None =None,embed:bool=True,popup:GeoJsonPopup|None=None,zoom_on_click:bool=False,marker: Circle| CircleMarker| Marker| None = None, **kwargs: str | float | bool | Sequence | dict | None*)**

Bases:**Layer**

CreatesaGeoJsonobjectforplotting intoaMap.

Parameters:

- **data** (*file, dict or str.*) – The GeoJSON data you want to plot. * If file, thendata will be read in the file and fully embedded in Leaflet's JavaScript. * Ifdict,thendatawillbeconverted toJSONandembedded intheJavaScript.*Ifstr,then datawillbe passed tothe JavaScriptas-is. *If *geo_interface*_____isavailable,the *geo_interface* dictionary will be serialized to JSON andreprojectedif *to_crs*is available.
- **style_function** (*function, default None*) – Function mapping a GeoJsonFeatureto a styledict.
- **highlight_function** (*function, default None*) – Function mapping a GeoJsonFeatureto a styledictfor mouse events.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- **smooth_factor** (*float, default None*) – How much to simplify the polyline oneach zoom level. More means better performance and smoother look, and lessmeansmore accuraterepresentation.Leafletdefaults to1.0.
- **tooltip** (*GeoJsonTooltip, Tooltipor str, default None*) – Display a text whenhovering over the object. Can utilize the data, see folium.GeoJsonTooltip forinfoon how to do that.
- **popup** (*GeoJsonPopup, optional*) – Show a different popup for each featurebypassing a GeoJsonPopup object.

- **marker** (*Circle, CircleMarkeror Marker, optional*) – If your data containsPoint geometry, you can format the markers by passing a Circle, CircleMarkerorMarker objectwith your wanted options.
  The *style_function* and *highlight_function* will also target the marker objectyou passed.
- **embed** (*bool, default True*) – Whether to embed the data in the html file ornot. Note that disabling embedding is only supported if you provide a file linkor URL.
- **zoom_on_click** (*bool, default False*) – Set to True to enable zooming in on ageometrywhen clicking on it.
- ****kwargs** – Keyword arguments are passed to the geoJson object as extraoptions.

Examples

```
>>>#Providingfilenamethatshallbeembedded.
>>>GeoJson("foo.json")
>>>#Providingfilenamethatshallnotbeembedded.
>>>GeoJson("foo.json",embed=False)
>>>#Providingdict.
>>>GeoJson(json.load(open("foo.json")))
>>>#Providingstring.
>>>GeoJson(open("foo.json").read())
>>> # Provide a style_function that color all states green
butAlabama.
>>>style_function=lambdax:{
...     "fillColor":"#0000ff"
...     ifx["properties"]["name"]=="Alabama"
...     else"#00ff00"
...}
>>>GeoJson(geojson,style_function=style_function)
```

convert_to_feature_collection()→ None

Convertdata intoaFeatureCollectionifitisnotalready.

find_identifier()→str

Findauniqueidentifierforeachfeature, createitifneeded.

AccordingtotheGeoJSONspecsafeature:

- MAYhavean‘id’fieldwithastringornumericalvalue.
- MUST have a ‘properties’ field. The content can be any json object or evennull.

get_geojson_from_web(*url:str*)→ dict

process_data(*data:Any*)→ dict

Convertanunknowndatainputintoa geojsondictionary.

render(***kwargs*)→None

RenderstheHTMLrepresentationoftheelement.

***class*folium.features.GeoJsonPopup(*fields:Sequence[str],aliases:Sequence[str]|None*=*None*, *labels:bool*=*True*,*style:str*='*margin:auto;*',*class_name:str*='*foliumpopup*',*localize:bool*=*True*, ***kwargs:str|float|bool|Sequence|dict|None*)**

Bases:`GeoJsonDetail`

Createapopupfeaturetobindto each elementofaGeoJsonlayerbasedon itsattributes.

Parameters:

- **fields** (*list or tuple.*) – Labels of GeoJson/TopoJson 'properties' or GeoPandasGeoDataFramecolumns you'd like todisplay.
- **aliases** (*list/tuple of strings, same length/order as fields, default None.*) – Optional aliases you'd like to display in the tooltip as field name instead of thekeysof *fields*.
- **labels** (*bool, default True.*) – Set to False to disable displaying the field namesoraliases.
- **localize** (*bool, default False.*) – This will use JavaScript's .toLocaleString() toformat 'clean' values as strings for the user's location; i.e. 1,000,000.00comma separators, float truncation, etc. Available for most of JavaScript'sprimitivetypes (any datayou'llserveinto the template).
- **style** (*str, default None.*) – HTML inline style properties like font and colors.Willbe applied to adivwith thetext init.

Examples

gjson=folium.GeoJson(gdf).add_to(m)

folium.features.GeoJsonPopup(fields=['NAME'],

labels=False).add_to(gjson)

class*folium.features.GeoJsonTooltip(*fields:Sequence[str],aliases:Sequence[str]|None*=*None*,*labels: bool* = *True*, *localize: bool* = *False*, *style: str | None* = *None*, *class_name: str* = *'foliumtooltip'*,*sticky:bool*= *True*,kwargs:str |float|bool|Sequence |dict|None*)**

Bases:`GeoJsonDetail`

Createa tooltipthat usesdatafromeither geojsonortopojson.

Parameters:

- **fields** (*list or tuple.*) – Labels of GeoJson/TopoJson 'properties' or GeoPandasGeoDataFramecolumns you'd like todisplay.
- **aliases** (*list/tuple of strings, same length/order as fields, default None.*) – Optional aliases you'd like to display in the tooltip as field name instead of thekeysof *fields*.
- **labels** (*bool, default True.*) – Set to False to disable displaying the field namesoraliases.

- **localize** (*bool, default False.*) – This will use JavaScript's .toLocaleString() toformat 'clean' values as strings for the user's location; i.e. 1,000,000.00comma separators, float truncation, etc. Available for most of JavaScript'sprimitivetypes (any datayou'llserveinto the template).
- **style** (*str, default None.*) – HTML inline style properties like font and colors.Willbe applied to adivwith thetext init.
- **sticky**(*bool,defaultTrue*)–Whetherthetooltipshouldfollowthemouse.
- ****kwargs** (*Assorted.*) – These values will map directly to the Leaflet Options.Moreinfoavailable here:https://leafletjs.com/reference.html#tooltip

Examples

Provide fields and aliases, with Style. >>> GeoJsonTooltip(…fields=["CNTY_NM", "census-pop-2015", "census-md-income-2015"], …aliases=["County", "2015 Census Population", "2015 Median Income"], …localize=True, … style=(… "background-color: grey; color: white; font-family:" …"courier new; font-size: 24px; padding: 10px;" …), …) # Provide fields, with labelsoff and fixed tooltip positions. >>> GeoJsonTooltip(fields=("CNTY_NM",),labels=False,sticky=False)

***class*folium.features.LatLngPopup**

Bases:**MacroElement**

When one clicks on a Map that contains a LatLngPopup, a popup is shown thatdisplaysthelatitudeand longitude ofthepointer.

***class* folium.features.RegularPolygonMarker(*location: Sequence[float], number_of_sides: int =4,rotation:int=0,radius:int=15,popup:Popup|str|None=None,tooltip:Tooltip|str|None=None,**kwargs:bool|str|float |None*)**

Bases:**JSCSSMixin**,**Marker**

CustommarkersusingtheLeafletDataVisFramework.

Parameters:

- **location**(*tupleorlist*)–LatitudeandLongitudeofMarker(Northing,Easting)
- **number_of_sides**(*int,default4*) –Numberof polygonsides
- **rotation**(*int,default0*)–Rotationangleindegrees
- **radius** (*int,default15*)–Markerradius,inpixels
- **popup** (*string or Popup, optional*) – Input text or visualization for objectdisplayedwhen clicking.
- **tooltip** (*str or folium.Tooltip, optional*) – Display a text when hovering overtheobject.
- ****kwargs**–Seevectorlayerspath_optionsforadditionalarguments.
- **https**(*//humangeo.github.io/leaflet-dvf/*)–

default_js*:List[Tuple[str,str]]*

***class*folium.features.TopoJson(*data:Any,object_path:str,style_function:Callable|None*=*None*,*name:str|None*=*None*,*overlay:bool*=*True*,*control:bool*=*True*,*show:bool*=*True*,*smooth_factor:float|None*= *None*,*tooltip:str|Tooltip|None* =*None*)**

Bases:**JSCSSMixin**,**Layer**

CreatesaTopoJsonobjectforplotting intoaMap.

Parameters:

- **data** (*file, dict or str.*) – The TopoJSON data you want to plot. * If file, thendata will be read in the file and fully embedded in Leaflet's JavaScript. * Ifdict,thendatawillbeconverted toJSONandembedded intheJavaScript.*Ifstr,then datawillbepassed to the JavaScriptas-is.
- **object_path** (*str*) – The path of the desired object into the TopoJson structure.Ex:'objects.myobject'.
- **style_function** (*function, default None*) – A function mapping a TopoJsongeometry toa style dict.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default False*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- **smooth_factor** (*float, default None*) – How much to simplify the polyline oneach zoom level. More means better performance and smoother look, and lessmeansmore accuraterepresentation.Leafletdefaults to1.0.
- **tooltip** (*GeoJsonTooltip, Tooltipor str, default None*) – Display a text whenhovering over the object. Can utilize the data, see folium.GeoJsonTooltip forinfoon how to do that.

Examples

```
>>>#Providingfilethatshallbeembedded.
>>>TopoJson(open("foo.json"),"object.myobject")
>>>#Providingfilenamethatshallnotbeembedded.
>>>TopoJson("foo.json","object.myobject")
>>>#Providingdict.
>>>TopoJson(json.load(open("foo.json")),"object.myobject")
>>>#Providingstring.
>>>TopoJson(open("foo.json").read(),"object.myobject")
>>> # Provide a style_function that color all states green
butAlabama.
>>>style_function=lambdax:{
...     "fillColor":"#0000ff"
...     ifx["properties"]["name"]=="Alabama"
...     else"#00ff00"
...}
```

```
>>> TopoJson(topo_json,
"object.myobject",style_function=style_fun
ction)
```

default_js*:List[Tuple[str,str]]*

get_bounds()→List[List[float]]

Computes the bounds of the object itself (not including it's children) in the form[[lat_min,lon_min],[lat_max, lon_max]]

render(***kwargs***)→None**

RenderstheHTMLrepresentationoftheelement.

style_data()→None

Appliesself.style_functiontoeachfeatureofself.data.

***class*folium.features.Vega(*data:Any*,*width:int|str|None*=*None*,*height:int|str|None*=*None*,*left:int|str*='0%',*top:int|str*= '0%',*position:str*='relative')**

Bases:**JSCSSMixin**,**Element**

CreatesaVegachart element.

Parameters:

- **data** (*JSON-like str or object*) – The Vega description of the chart. It can alsobe any object that has a method *to_json*, so that you can (for instance) provideav*incent*chart.
- **width** (*int or str, default None*) – The width of the output element. If None,either data['width'] (if available) or '100%' will be used. Ex: 120, '120px','80%'
- **height** (*int or str, default None*) – The height of the output element. If None,either data['width'] (if available) or '100%' will be used. Ex: 120, '120px','80%'
- **left** (*int or str, default '0%'*) – The horizontal distance of the output withrespecttothe parentHTMLobject.Ex:120, '120px', '80%'
- **top** (*int or str, default '0%'*) – The vertical distance of the output with respecttothe parentHTMLobject.Ex:120, '120px', '80%'
- **position** (*str, default 'relative'*) – The *position* argument that the CSS shallcontain. Ex:'relative', 'absolute'

default_js*:List[Tuple[str,str]]*

render(***kwargs***)→None**

RenderstheHTMLrepresentationoftheelement.

***class*folium.features.VegaLite(*data:Any*,*width:int|str|None*=*None*,*height:int|str|None*=*None*,*left:int|str*= '0%',*top:int|str* = '0%',*position:str*= 'relative')**

Bases:**Element**

CreatesaVega-Litechartelement.

Parameters:

- **data** (*JSON-like str or object*) – The Vega-Lite description of the chart. It canalso be any object that has a method *to_json*, so that you can (for instance)provideanAltair chart.
- **width** (*int or str, default None*) – The width of the output element. If None,either data['width'] (if available) or '100%' will be used. Ex: 120, '120px','80%'
- **height** (*int or str, default None*) – The height of the output element. If None,either data['width'] (if available) or '100%' will be used. Ex: 120, '120px','80%'
- **left** (*int or str, default '0%'*) – The horizontal distance of the output withrespecttothe parentHTMLobject.Ex:120, '120px', '80%'
- **top** (*int or str, default '0%'*) – The vertical distance of the output with respecttothe parentHTMLobject.Ex:120, '120px', '80%'
- **position** (*str, default 'relative'*) – The *position* argument that the CSS shallcontain. Ex:'relative', 'absolute'

render(*kwargs*)→None**

RenderstheHTMLrepresentationoftheelement.

property*vegalite_major_version*:int|None

Plugins

Wrapsomeofthe mostpopularleaflet externalplugins.

class*folium.plugins.AntPath(*locations*,*popup*=*None*,*tooltip*=*None*,kwargs*)**

Bases:**JSCSSMixin**,**BaseMultiLocation**

Classfordrawing AntPathpolyline overlayson amap.

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **locations** (*list of points (latitude, longitude)*) – Latitude and Longitude of line(Northing,Easting)
- **popup** (*str or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, optional*) – Display a text when hovering overtheobject.
- ****kwargs** – Polyline and AntPath options. See their Github page for theavailableparameters.
- **https**(*//github.com/rubenspgcavalcante/leaflet-ant-path/*)–

default_js*:List[Tuple[str,str]]*

class* folium.plugins.BeautifyIcon(*icon*=*None*, *icon_shape*=*None*, *border_width*=*3*, *border_color*='#000',*text_color*='#000',*background_color*='#FFF',*inner_icon_style*='',*spin*=*False*,*number*=*None*,

kwargs*)

Bases:**JSCSSMixin**,**MacroElement**

CreateaBeautifyIconthatcanbeadded to aMarker

Parameters:

- **icon** (*string, default None*) – the Font-Awesome icon name to use to render themarker.
- **icon_shape**(*string,defaultNone*) –theiconshape
- **border_width**(*integer,default3*)–the borderwidthof theicon
- **border_color** (*string with hexadecimal RGB, default '#000'*) – the bordercolorof theicon
- **text_color** (*string with hexadecimal RGB, default '#000'*) – the text color oftheicon
- **background_color** (*string with hexadecimal RGB, default '#FFF'*) – thebackgroundcolor ofthe icon
- **inner_icon_style** (*string with css styles for the icon, default ''*) – the css stylesof theicon
- **spin**(*boolean,defaultFalse*)–allowthe icontobespinning.
- **number**(*integer,defaultNone*)– thenumberof theicon.

Examples

Plugin Website: masajid390/BeautifyMarker >>> BeautifyIcon(…text_color=”#000”,border_color=”transparent”,background_color=”#FFF” …).add_to(marker) >>> number_icon = BeautifyIcon(… text_color=”#000”, …border_color=”transparent”, … background_color=”#FFF”, … number=10, …inner_icon_style=”font-size:12px;padding-top:-5px;”, …) >>> Marker(…location=[45.5, -122.3], … popup=folium.Popup(“Portland, OR”), …icon=number_icon, …) >>> BeautifyIcon(icon=”arrow-down”,icon_shape=”marker”).add_to(marker)

ICON_SHAPE_TYPES

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

*class*folium.plugins.BoatMarker(*location*,*popup*=*None*,*icon*=*None*,*heading*=*0*,*wind_heading*=*None*,*wind_speed*=*0*,***kwargs*)

Bases:`JSCSSMixin`,**`Marker`**

AddaMarker intheshapeof aboat.

Parameters:

- **location** (*tuple of length 2, default None*) – The latitude and longitude of themarker.If None, then the middleof the map is used.
- **heading** (*int, default 0*) – Heading of the boat to an angle value between 0 and360 degrees

- **wind_heading** (*int, default None*) – Heading of the wind to an angle valuebetween0 and 360 degreesIf None, thennowind is represented.
- **wind_speed** (*int,default0*)–Speedofthewindinknots.
- **https**(*//github.com/thomasbrueggemann/leaflet.boatmarker*)–

default_js*:List[Tuple[str,str]]*

***class*folium.plugins.CirclePattern(*width*=*20*,*height*=*20*,*radius*=*12*,*weight*=*2.0*,*color*='#3388ff',*fill_color*='#3388ff',*opacity*=*0.75*,*fill_opacity*=*0.5*)**

Bases:**JSCSSMixin**,**MacroElement**

FillPatternforpolygoncomposedofrepeatingcircles.

Addthesetothe‘fillPattern’fieldinGeoJsonstylefunctions.

Parameters:

- **width**(*int,default20*)–Horizontaldistancebetweencircles(pixels).
- **height**(*int,default20*)–Verticaldistancebetweencircles(pixels).
- **radius**(*int,default12*)–Radiusofeachcircle(pixels).
- **weight**(*float,default2.0*)–Widthofoutlinearoundeach circle (pixels).
- **color** (*string with hexadecimal, RGB, or named color, default "#3388ff"*) – Colorof thecircleoutline.
- **fill_color**(*stringwithhexadecimal,RGB, ornamedcolor,default"#3388ff"*) –Colorofthecircle interior.
- **opacity** (*float, default 0.75*) – Opacity of the circle outline. Should be between0 and 1.
- **fill_opacity** (*float, default 0.5*) – Opacity of the circle interior. Should bebetween0 and 1.
- **https** (*See*)–

default_js*:List[Tuple[str,str]]*

render(*kwargs*)**

RenderstheHTMLrepresentationoftheelement.

***class* folium.plugins.Draw(*export*=*False*, *filename*='data.geojson', *position*='topleft', *show_geometry_on_click*=*True*,*draw_options*=*None*,*edit_options*=*None*)**

Bases:**JSCSSMixin**,**MacroElement**

VectordrawingandeditingpluginforLeaflet.

Parameters:

- **export** (*bool, default False*) – Add a small button that exports the drawnshapesas a geojson file.
- **filename**(*string,default'data.geojson'*)–Nameof geojsonfile
- **position** (*{'topleft', 'toprigth', 'bottomleft', 'bottomright'}*) – Position of control.Seehttps://leafletjs.com/reference.html#control

- **show_geometry_on_click** (*bool, default True*) – When True, opens an alertwiththe geometry description on click.
- **draw_options** (*dict, optional*) – The options used to configure the drawtoolbar. See http://leaflet.github.io/Leaflet.draw/docs/leaflet-draw-latest.html#drawoptions
- **edit_options** (*dict, optional*) – The options used to configure the edit toolbar.See https://leaflet.github.io/Leaflet.draw/docs/leaflet-draw-latest.html#editpolyoptions

Examples

```
>>>m=folium.Map()
>>>Draw(
...      export=True,
...      filename="my_data.geojson",
...      position="topleft",
...      draw_options={"polyline":{"allowIntersection":False}},
...      edit_options={"poly":{"allowIntersection":False}},
...).add_to(m)
```

For more info please check https://leaflet.github.io/Leaflet.draw/docs/leaflet-draw-latest.html

default_css*:List[Tuple[str,str]]*

default_js*: List[Tuple[str,*

str]]*render(kwargs*)

RenderstheHTMLrepresentationoftheelement.

*class*folium.plugins.**DualMap(***location=None,layout='horizontal',**kwargs***)**

Bases:**JSCSSMixin**,**MacroElement**

Createtwo maps inthe samewindow.

Adding children to this objects adds them to both maps. You can access the individualmapswith *DualMap.m1* and*DualMap.m2*.

UsestheLeafletpluginSync:jieter/Leaflet.Sync

Parameters:

- **location** (*tuple or list, optional*) – Latitude and longitude of center point of themaps.
- **layout** (*{'horizontal', 'vertical'}*) – Select how the two maps should bepositioned.Eitherhorizontal(left andright)orvertical(topand bottom).
- ****kwargs**–Keywordargumentsarepassedtothe twoMapobjects.

Examples

```
>>>#DualMapacceptsthesameargumentsasMap:
>>> m = DualMap(location=(0, 0),
tiles="cartodbpositron",zoom_start=5)
>>>#Addthesamemarkertobothmaps:
>>>Marker((0,0)).add_to(m)
>>>#Theindividualmapsareattributescalled`m1`and`m2`:
>>>Marker((0,1)).add_to(m.m1)
>>>LayerControl().add_to(m)
>>>m.save("map.html")
```

add_child(*child*,*name*=*None*,*index*=*None*)

Addobject*child*tothe firstmap andstore itfor thesecond.

default_js*:* ***List[Tuple[str,***

***str]]*fit_bounds(**args*,**

*****kwargs*)keep_in_front(**a***

***rgs*)**

render(*kwargs*)**

RenderstheHTMLrepresentationoftheelement.

class*folium.plugins.FastMarkerCluster(*data*,*callback*=*None*,*options*=*None*,*name*=*None*,*overlay*=*True*,*control*=*True*,*show*=*True*,*icon_create_function*=*None*,kwargs*)**

Bases:**MarkerCluster**

Add marker clusters to a map using in-browser rendering. Using FastMarkerCluster itispossibleto render 000'sof pointsfarquicker than theMarkerCluster class.

BeawarethattheFastMarkerClusterclasspassesanemptylisttotheparentclass' init____method during initialisation. This means that the add_child method is nevercalled, and no reference to any marker data are retained. Methods such asget_bounds()arethereforenotavailable when using it.

Parameters:

- **data** (*list of list with values*) – List of list of shape [[lat, lon], [lat, lon], etc.]Whenyouusea customcallbackyou couldaddmorevalues afterthelatandlon.E.g. [[lat,lon, 'red'], [lat, lon, 'blue']]
- **callback** (*string, optional*) – A string representation of a valid Javascriptfunction that will be passed each row in data. See the FasterMarkerCluster foranexample of acustomcallback.
- **name** (*string, optional*) – The name of the Layer, as it will appear inLayerControls.
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

- **icon_create_function** (*string, default None*) – Override the default behaviour,makingpossible tocustomizemarkerscolors and sizes.
- ****kwargs**–AdditionalargumentsarepassedtoLeaflet.markerclusteroptions. SeeLeaflet/Leaflet.markercluster

***class* folium.plugins.FeatureGroupSubGroup(*group*, *name*=*None*, *overlay*=*True*, *control*=*True*, *show*=*True*)**

Bases:**JSCSSMixin**,**Layer**

Creates a Feature Group that adds its child layers into a parent group when added to amap (e.g. through LayerControl). Useful to create nested groups, or cluster markersfrommultipleoverlays. From [0].

[0] ghybs/Leaflet.FeatureGroup.SubGroup

Parameters:

- **group** (*Layer*) – The MarkerCluster or FeatureGroup containing thissubgroup.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

Examples

Nestedgroups

```
>>>fg=folium.FeatureGroup()#Maingroup
>>> g1 = folium.plugins.FeatureGroupSubGroup(fg, "g1")#
Firstsubgroupoffg
>>> g2 = folium.plugins.FeatureGroupSubGroup(fg, "g2")#
Secondsubgroupoffg
>>>m.add_child(fg)
>>>m.add_child(g1)
>>>m.add_child(g2)
>>>g1.add_child(folium.Marker([0,0]))
>>>g2.add_child(folium.Marker([0,1]))
>>>folium.LayerControl().add_to(m)
```

Multipleoverlayspartofthesameclustergroup

```
>>>mcg=folium.plugins.MarkerCluster(
...     control=False
...)#MarkerCluster,hiddenincontrols
>>> g1 = folium.plugins.FeatureGroupSubGroup(mcg, "g1")#
Firstgroup,inmcg
>>> g2 = folium.plugins.FeatureGroupSubGroup(mcg, "g2")#
Secondgroup,inmcg
>>>m.add_child(mcg)
```

```
>>>m.add_child(g1)
>>>m.add_child(g2)
>>>g1.add_child(folium.Marker([0,0]))
>>>g2.add_child(folium.Marker([0,1]))
>>>folium.LayerControl().add_to(m)
```

default_js*:List[Tuple[str,str]]*

class*folium.plugins.FloatImage(*image*,*bottom*=*75*,*left*=*75*,kwargs*)**

Bases:**MacroElement**

Adds afloatingimageinHTML canvasontopofthemap.

Parameters:

- **image** (*str*) – Url to image location. Can also be an inline image using a dataURIor a localfileusing*file://*.
- **bottom** (*int, default 75*) – Vertical position from the bottom, as a percentageofscreen height.
- **left**(*int,default75*)–Horizontalposition fromtheleft,asapercentageofscreenwidth.
- ****kwargs** – Additional keyword arguments are applied as CSS properties. Forexample:*width='300px'*.

class*folium.plugins.Fullscreen(*position*='topleft',*title*='FullScreen',*title_cancel*='ExitFullScreen',*force_separate_button*=*False*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Adds afullscreenbuttontoyour map.

Parameters:

- **position** (*str*) – change the position of the button can be: 'topleft', 'topright','bottomright'or 'bottomleft' default:'topleft'
- **title**(*str*)–change thetitleofthebutton, default:'FullScreen'
- **title_cancel** (*str*) – change the title of the button when fullscreen is on, default:'ExitFullScreen'
- **force_separate_button** (*bool, default False*) – force separate button to detachfromzoombuttons,
- **https** (*See*)–

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class*folium.plugins.Geocoder(*collapsed*=*False*,*position*='topright',*add_marker*=*True*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

AsimplegeocoderforLeafletthatbydefaultusesOSM/Nominatim.

Pleaserespect theNominatimusage policy:https://operations.osmfoundation.org/policies/nominatim/

Parameters:

- **collapsed** (*bool, default False*) – If True, collapses the search box unlesshovered/clicked.
- **position** (*str, default 'topright'*) – Choose from 'topleft', 'topright','bottomleft'or 'bottomright'.
- **add_marker** (*bool, default True*) – If True, adds a marker on the foundlocation.
- **https** (*Foralloptionssee*) –

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class*folium.plugins.GroupedLayerControl(*groups*,*exclusive_groups*=*True*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Createa Layer Controlwithgroups ofoverlays.

Parameters:

- **groups** (*dict*)–

  A dictionary where the keys are group names and the values are lists of layerobjects.e.g. {
  "Group1":[layer1, layer2], "Group2":[layer3,layer4]
  }

- **exclusive_groups** (*bool, default True*) – Whether to use radio buttons(default)orcheckboxes.Ifyouwanttouseboth,usetwoseparateinstancesofthisclass.
- ****kwargs**–Additional(possiblyinherited)options. Seehttps://leafletjs.com/reference.html#control-layers

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class*folium.plugins.HeatMap(*data*,*name*=*None*,*min_opacity*=*0.5*,*max_zoom*=*18*,*radius*=*25*,*blur*=*15*,*gradient*=*None*,*overlay*=*True*,*control*=*True*,*show*=*True*,kwargs*)**

Bases:**JSCSSMixin**,**Layer**

CreateaHeatmaplayer

Parameters:

- **data** (*list of points of the form [lat, lng] or [lat, lng, weight]*) – The points youwantto plot. Youcan also provide anumpy.array of shape(n,2)or (n,3).
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls.
- **min_opacity**(*default1.*)–Theminimumopacity theheatwillstartat.
- **max_zoom** (*default 18*) – Zoom level where the points reach maximumintensity (as intensity scales with zoom), equals maxZoom of the map bydefault
- **radius**(*int,default25*)–Radiusofeach"point"oftheheatmap
- **blur**(*int,default15*)–Amount ofblur
- **gradient** (*dict, default None*) – Color gradient config. e.g. {0.4: 'blue', 0.65:'lime',1:'red'}
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

default_js*:List[Tuple[str,str]]*

***class*folium.plugins.HeatMapWithTime(*data*,*index*=*None*,*name*=*None*,*radius*=*15*,*blur*=*0.8*,*min_opacity*=*0*,*max_opacity*=*0.6*,*scale_radius*=*False*,*gradient*=*None*,*use_local_extrema*=*False*,*auto_play*=*False*,*display_index*=*True*,*index_steps*=*1*,*min_speed*=*0.1*,*max_speed*=*10*,*speed_step*=*0.1*,*position*='*bottomleft*',*overlay*=*True*,*control*=*True*,*show*=*True*)**

Bases:**JSCSSMixin**,**Layer**

CreateaHeatMapWithTimelayer

Parameters:

- **data** (*list of list of points of the form [lat, lng] or [lat, lng, weight]*) – Thepoints you want to plot. The outer list corresponds to the various time steps insequential order. (weight is in (0, 1] range and defaults to 1 if not specified forapoint)
- **index** (*Index giving the label (or timestamp) of the elements of data. Shouldhave*) – the same length as data, or is replaced by a simple count if notspecified.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls.
- **radius** (*default15.*) –Theradiususedaroundpointsfortheheatmap.
- **blur** (*default 0.8.*) – Blur strength used for the heatmap. Must be between 0and 1.
- **min_opacity**(*default0*)–Theminimumopacityfor theheatmap.
- **max_opacity**(*default0.6*)– Themaximumopacityfor theheatmap.
- **scale_radius** (*default False*) – Scale the radius of the points based on thezoomlevel.

- **gradient** (*dict, default None*) – Match point density values to colors. Color canbeaname ('red'),RGB values ('rgb(255,0,0)')or a hex number('#FF0000').
- **use_local_extrema** (*default False*) – Defines whether the heatmap uses aglobal extrema set found from the input data OR a local extrema (themaximumand minimumofthe currently displayed view).
- **auto_play**(*defaultFalse*)–Automaticallyplaytheanimationacrosstime.
- **display_index** (*default True*) – Display the index (usually time) in the timecontrol.
- **index_steps** (*default 1*) – Steps to take in the index dimension betweenanimationsteps.
- **min_speed**(*default 0.1*)–Minimumfps speedforanimation.
- **max_speed**(*default 10*)–Maximumfpsspeedforanimation.
- **speed_step** (*default 0.1*) – Step between different fps speeds on the speedslider.
- **position** (*default 'bottomleft'*) – Position string for the time slider. Format:'bottom/top'+'left/right'.
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

default_css*:List[Tuple[str,str]]*

default_js*: List[Tuple[str,*

*str]]***render(*****kwargs***)**

RenderstheHTMLrepresentationoftheelement.

class*folium.plugins.LocateControl(*auto_start*=*False*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Controlpluginto geolocatethe user.

Thispluginsaddsabuttontothemap,and whenit'sclickedshowsthecurrentuserdevicelocation.

Toworkproperlyin production,theconnectionneedstobeencrypted,otherwisebrowserwillnotallow users to sharetheirlocation.

Parameters:

- **auto-start** (*bool, default False*) – When set to True, plugin will be activatedon map loading and search for user position. Once user location is founded,themap willautomaticallycentered inusing usercoordinates.
- ****kwargs**–Forpossibleoptions,seedomoritz/leaflet-locatecontrol

Examples

```
>>>m=folium.Map()
```

```
#Withdefaultsettings
>>>LocateControl().add_to(m)
# With some custom options >>> LocateControl( … position=”bottomright”,
…strings={“title”:“Seeyoucurrentlocation”,“popup”:“Yourposition”},…
).add_to(m)
```

Formoreinfocheck:domoritz/leaflet-locatecontrol

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class*folium.plugins.MarkerCluster(*locations*=*None*,*popups*=*None*,*icons*=*None*,*name*=*None*,*overlay*=*True*,*control*=*True*,*show*=*True*,*icon_create_function*=*None*,*options*=*None*,kwargs*)**

Bases:**JSCSSMixin**,**Layer**

ProvidesBeautifulAnimatedMarkerClusteringfunctionalityformaps.

Parameters:

- **locations** (*list of list or array of shape (n, 2).*) – Data points of the form [[lat,lng]].
- **popups** (*list of length n, default None*) – Popup for each marker, either aPopupobjector a string or None.
- **icons** (*list of length n, default None*) – Icon for each marker, either an Iconobjector a string orNone.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls
- **overlay** (*bool, default True*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- **icon_create_function** (*string, default None*) – Override the default behaviour,makingpossible tocustomizemarkerscolors and sizes.
- **options** (*dict, default None*) – A dictionary with options forLeaflet.markercluster.SeeLeaflet/Leaflet.markerclusterforoptions .

Example

```
>>>icon_create_function='''
...     function(cluster){
...     return L.divIcon({html: '<b>' + cluster.getChildCount()
+'</b>',
...                       className: 'marker-cluster marker-
cluster-small',
...                       iconSize:newL.Point(20,20)});
...     }
...'''
```

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class **folium.plugins.MeasureControl(*position*='topright', *primary_length_unit*='meters', *secondary_length_unit*='miles',*primary_area_unit*='sqmeters',*secondary_area_unit*='acres',***kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Addameasurement widgetonthe map.

Parameters:

- **position**(*str,default'topright'*)–Locationofthewidget.
- **primary_length_unit**(*str,default'meters'*)–
- **secondary_length_unit**(*str,default'miles'*)–
- **primary_area_unit** (*str,default'sqmeters'*)–
- **secondary_area_unit**(*str, default'acres'*) –
- **https** (*See*)–

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class**folium.plugins.MiniMap(*tile_layer*=None,*position*='bottomright',*width*=150,*height*=150,*collapsed_width*=25,*collapsed_height*=25,*zoom_level_offset*=-5,*zoom_level_fixed*=None,*center_fixed*=False,*zoom_animation*=False,*toggle_display*=False,*auto_toggle_display*=False,*minimized*=False,***kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Addaminimap(locator)toanexistingmap.

Uses the Leaflet plugin by Norkart under BSD 2-Clause "Simplified"License.Norkart/Leaflet-MiniMap

Parameters:

- **tile_layer** (*folium TileLayer object or str, default None*) – Provide a foliumTileLayer object or the wanted tiles as string. If not provided it will use thedefaultof 'TileLayer', currentlyOpenStreetMap.
- **position** (*str, default 'bottomright'*) – The standard Control position parameterforthewidget.
- **width**(*int,default150*)–Thewidthoftheminimapinpixels.
- **height**(*int,default150*)–Theheightof theminimapinpixels.
- **collapsed_width** (*int, default 25*) – The width of the toggle marker and theminimapwhen collapsed in pixels.
- **collapsed_height** (*int, default 25*) – The height of the toggle marker and theminimapwhen collapsed
- **zoom_level_offset** (*int, default -5*) – The offset applied to the zoom in theminimapcomparedtothezoomof themainmap.Canbepositiveornegative.

- **zoom_level_fixed** (*int, default None*) – Overrides the offset to apply a fixedzoom level to the minimap regardless of the main map zoom. Set it to anyvalidzoomlevel,if unset zoom_level_offsetis usedinstead.
- **center_fixed** (*bool, default False*) – Applies a fixed position to the minimapregardless of the main map's view / position. Prevents panning the minimap,but does allow zooming (both in the minimap and the main map). If theminimap is zoomed, it will always zoom around the centerFixed point. Youcanpass in a LatLng-equivalentobject.
- **zoom_animation** (*bool, default False*) – Sets whether the minimap shouldhave an animated zoom. (Will cause it to lag a bit after the movement of themainmap.)
- **toggle_display** (*bool, default False*) – Sets whether the minimap should haveabutton tominimiseit.
- **auto_toggle_display** (*bool, default False*) – Sets whether the minimap shouldhide automatically if the parent map bounds does not fit within the minimapbounds.Especiallyusefulwhen'zoomLevelFixed'isset.
- **minimized** (*bool, default False*) – Sets whether the minimap should start in aminimizedposition.

Examples

```
>>>MiniMap(position="bottomleft")
```

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class*folium.plugins.MousePosition(*position*='bottomright',*separator*=':',*empty_string*='Unavailable',*lng_first*=False,*num_digits*=5,*prefix*='',*lat_formatter*=None,*lng_formatter*=None,kwargs**)**

Bases:**JSCSSMixin**,**MacroElement**

Addafieldthatshowsthe coordinatesofthemouseposition.

Uses the Leaflet plugin by Ardhi Lukianto under MITlicense.ardhi/Leaflet.MousePosition

Parameters:

- **position** (*str, default 'bottomright'*) – The standard Control position parameterforthewidget.
- **separator** (*str, default ' : '*) – Character used to separate latitude and longitudevalues.
- **empty_string**(*str,default'Unavailable'*)–Initialtexttodisplay.
- **lng_first** (*bool, default False*) – Whether to put the longitude first or not. SetasTrueto displaylongitude beforelatitude.
- **num_digits** (*int, default '5'*) – Number of decimal places included in thedisplayedlongitudeandlatitude decimaldegree values.
- **prefix**(*str,default''*)–Astringtobeprependedtothe coordinates.

- **lat_formatter** (*str, default None*) – Custom Javascript function to format thelatitudevalue.
- **lng_formatter** (*str, default None*) – Custom Javascript function to format thelongitudevalue.

Examples

```
>>> fmtr = "function(num) {return L.Util.formatNum(num, 3) + '
°';};"
>>>MousePosition(
...     position="topright",
...     separator="|",
...     prefix="Mouse:",
...     lat_formatter=fmtr,
...     lng_formatter=fmtr,
...)
```

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

class*folium.plugins.PolyLineOffset(*locations*,*popup*=*None*,*tooltip*=*None*,*offset*=*0*,kwargs*)**

Bases:**JSCSSMixin**,**PolyLine**

AddoffsetcapabilitiestothePolyLineclass.

This plugin adds to folium Polylines the ability to be drawn with a relative pixeloffset, without modifying their actual coordinates. The offset value can be eithernegative or positive, for left- or right-side offset, and remains constant across zoomlevels.

See **folium.vector_layers.path_options()** for the*Path*options.

Parameters:

- **locations** (*list of points (latitude, longitude)*) – Latitude and Longitude of line(Northing,Easting)
- **popup** (*str or folium.Popup, default None*) – Input text or visualization forobjectdisplayed when clicking.
- **tooltip** (*str or folium.Tooltip, optional*) – Display a text when hovering overtheobject.
- **offset** (*int, default 0*) – Relative pixel offset to draw a line parallel to anexistentone, ata fixed distance.
- ****kwargs** – Polyline options. See their Github page for the availableparameters.
- **https** (*See*)–

Examples

```
>>>plugins.PolyLineOffset(
```

```
...     [[58,-28],[53,-23]],color="#f00",opacity=1,offset=-
5
...).add_to(m)
>>>plugins.PolyLineOffset(
...     [[58,-28],[53,-23]],color="#080",opacity=1,
offset=10
...).add_to(m)
```

default_js*:List[Tuple[str,str]]*

class*folium.plugins.PolyLineTextPath(*polyline*,*text*,*repeat*=*False*,*center*=*False*,*below*=*False*,*offset*=*0*,*orientation*=*0*,*attributes*=*None*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Showsatextalonga PolyLine.

Parameters:

- **polyline** (*folium.features.PolyLine object*) – The folium.features.PolyLineobject toattach the textto.
- **text**(*string*)–Thestringtobeattachedto thepolyline.
- **repeat** (*bool, default False*) – Specifies if the text should be repeated along thepolyline.
- **center** (*bool, default False*) – Centers the text according to the polyline'sbounding box
- **below** (*bool,default False*)–Showtextbelowthepath
- **offset**(*int,default0*)–Setan offsettopositiontextrelative tothe polyline.
- **orientation**(*int,default0*) –Rotatetexttoaspecified angle.
- **attributes** (*dict*) – Object containing the attributes applied to the text tag.Check valid attributes here: https://developer.mozilla.org/en-US/docs/Web/SVG/Element/text#attributes Example: {'fill': '#007DEF','font-weight':'bold', 'font-size': '24'}
- **https** (*See*)–

default_js*:List[Tuple[str,str]]*

***class*folium.plugins.ScrollZoomToggler**

Bases:**MacroElement**

Createsabuttonfor enabling/disablingscrollontheMap.

class*folium.plugins.Search(*layer*,*search_label*=*None*,*search_zoom*=*None*,*geom_type*='*Point*',*position*='*topleft*',*placeholder*='*Search*',*collapsed*=*False*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Adds asearchtoolto yourmap.

Parameters:

- **layer** (*GeoJson, TopoJson, FeatureGroup, MarkerCluster class object.*) – Themaplayertoindexin the Search view.
- **search_label** (*str, optional*) – 'properties' key in layer to index Search, if layerisGeoJson/TopoJson.
- **search_zoom** (*int, optional*) – Zoom level to set the map to on match. BydefaultzoomstoPolygon/Line boundsandpointson theirnaturalextent.
- **geom_type** (*str, default 'Point'*) – Feature geometry type. "Point", "Line" or"Polygon"
- **position** (*str, default 'topleft'*) – Change the position of the search bar, can be:'topleft','topright', 'bottomright' or 'bottomleft',
- **placeholder** (*str, default 'Search'*) – Placeholder text inside the Search box ifnothingis entered.
- **collapsed** (*boolean, default False*) – Whether the Search box should becollapsedor not.
- ****kwargs.** – Assorted style options to change feature styling on match. Usethesame way as vector layer arguments.
- **https** (*See*)–

default_css*:List[Tuple[str,str]]*

default_js*: List[Tuple[str,*

str]]*render(kwargs*)

RenderstheHTMLrepresentationoftheelement.

test_params(*keys*)

class*folium.plugins.SemiCircle(*location*,*radius*,*direction*=*None*,*arc*=*None*,*start_angle*=*None*,*stop_angle*=*None*,*popup*=*None*,*tooltip*=*None*,kwargs*)**

Bases:**JSCSSMixin**,**Marker**

Add a marker in the shape of a semicircle, similar to the Circle

class.Use(directionandarc)or (start_angleand stop_angle), notboth.

Parameters:

- **location**(*tuple[float, float]*)–LatitudeandLongitudepair(Northing,Easting)
- **radius**(*float*)–Radiusofthecircle,inmeters.
- **direction**(*int,defaultNone*)–Directionangleindegrees
- **arc**(*int,defaultNone*)– Arcangleindegrees.
- **start_angle**(*int,defaultNone*)– Startangle indegrees
- **stop_angle**(*int,defaultNone*)–Stopangleindegrees.
- **popup** (*str or folium.Popup, optional*) – Input text or visualization for objectdisplayedwhen clicking.
- **tooltip** (*str or folium.Tooltip, optional*) – Display a text when hovering overtheobject.
- ****kwargs**–Foradditionalarguments see**folium.vector_layers.path_options()**
- **https** (*UsesLeafletplugin*)–

default_js*:List[Tuple[str,str]]*

***class*folium.plugins.SideBySideLayers(*layer_left*,*layer_right*)**

Bases:**JSCSSMixin**,**MacroElement**

Creates a SideBySideLayers that takes two Layers and adds a sliding control with theleaflet-side-by-sideplugin.

UsestheLeafletleaflet-side-by-sideplugindigidem/leaflet-side-by-side

Parameters:

- **layer_left** (*Layer.*) – The left Layer within the side by side control. Must becreated andadded to the mapbefore being passed tothis class.
- **layer_right** (*Layer.*) – The right Layer within the side by side control. Mustbecreatedand added to themap beforebeing passedto thisclass.

Examples

```
>>>sidebyside=SideBySideLayers(layer_left,layer_right)
>>>sidebyside.add_to(m)
```

default_js*:List[Tuple[str,str]]*

class*folium.plugins.StripePattern(*angle*=*0.5*,*weight*=*4*,*space_weight*=*4*,*color*=*'#000000'*,*space_color*=*'#ffffff'*,*opacity*=*0.75*,*space_opacity*=*0.0*,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

FillPatternforpolygoncomposedofalternatinglines.

Addthesetothe‘fillPattern’fieldinGeoJsonstylefunctions.

Parameters:

- **angle** (*float, default 0.5*) – Angle of the line pattern (degrees). Should bebetween-360 and 360.
- **weight**(*float,default4*)–Widthofthemainlines(pixels).
- **space_weight**(*float*)–Widthofthealternatelines(pixels).
- **color** (*string with hexadecimal, RGB, or named color, default "#000000"*) – Colorof themain lines.
- **space_color**(*stringwithhexadecimal,RGB,ornamedcolor,default "#ffffff"*) –Colorof thealternatelines.
- **opacity** (*float, default 0.75*) – Opacity of the main lines. Should be between 0and 1.
- **space_opacity** (*float, default 0.0*) – Opacity of the alternate lines. Should bebetween0 and 1.
- **https** (*See*)–

default_js*:List[Tuple[str,str]]*

render(***kwargs***)**

RenderstheHTMLrepresentationoftheelement.

class*folium.plugins.TagFilterButton(*data*,*icon*='fa-filter',*clear_text*='clear',*filter_on_every_click*=True,*open_popup_on_hover*=False,kwargs*)**

Bases:**JSCSSMixin**,**MacroElement**

Creates a Tag Filter Button to filter elements based on criteria (maydemirx/leaflet-tag-filter-button)

This plugin works for multiple element types like Marker, GeoJson and vector layerslikePolyLine.

Parameters:

- **data**(*list,ofstrings.*)–Thetagstofilterforthisfilterbutton.
- **icon**(*string,default'fa-filter'*)–Theicon forthefilterbutton
- **clear_text**(*string,default'clear'*)–Textoftheclearbutton
- **filter_on_every_click** (*bool, default True*) – if True, the plugin will filter oneveryclick eventoncheckbox.
- **open_popup_on_hover** (*bool, default False*) – if True, popup that containstagswillbe open atmouse hover time

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]c*

***lass*folium.plugins.Terminator**

Bases:**JSCSSMixin**,**MacroElement**

Leaflet.Terminator is a simple plug-in to the Leaflet library to overlay day and nightregionson maps.

default_js*:List[Tuple[str,str]]*

***class*folium.plugins.TimeSliderChoropleth(*data*,*styledict*,*highlight:bool*=False,*name*=None,*overlay*=True,*control*=True,*show*=True,*init_timestamp*=0)**

Bases:**JSCSSMixin**,**Layer**

Createachoroplethwith atimesliderfor timestampeddata.

Visualize timestamped data, allowing users to view the choropleth at differenttimestampsusing a slider.

Parameters:

- **data**(*str*)–geojsonstring
- **styledict**(*dict*)–
  Adictionarywherethekeysarethegeojsonfeatureidsandthevalues are dicts of*{time: style_options_dict}*
- **highlight** (*bool, default False*) – Whether to show a visual effect on mousehoverand click.
- **name** (*string, default None*) – The name of the Layer, as it will appear inLayerControls.
- **overlay** (*bool, default False*) – Adds the layer as an optional overlay (True) orthebase layer (False).
- **control** (*bool, default True*) – Whether the Layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.
- **init_timestamp** (*int, default 0*) – Initial time-stamp index on the slider. Mustbeinthe range*[-L,L-1]*, where*L*is themaximumnumber oftime stamps in*styledict*.Forexample,use-*1*toinitializethesliderto thelatesttimestamp.

default_js*:List[Tuple[str,str]]*

class **folium.plugins.TimestampedGeoJson(*data*, *transition_time*=*200*, *loop*=*True*, *auto_play*=*True*, *add_last_point*=*True*, *period*=*'P1D'*, *min_speed*=*0.1*, *max_speed*=*10*, *loop_button*=*False*, *date_options*=*'YYYY-MM-DDHH:mm:ss'*, *time_slider_drag_update*=*False*, *duration*=*None*, *speed_slider*=*True*)**

Bases:**JSCSSMixin**,**MacroElement**

Creates a TimestampedGeoJson plugin from timestamped GeoJSONs to append into amapwith Map.add_child.

Ageo-jsonistimestampedif:

- it contains only features of types LineString, MultiPoint, MultiLineString,Polygonand MultiPolygon.
- each feature has a 'times' property with the same length as the coordinatesarray.
- each element of each 'times' property is a timestamp in ms since epoch, or inISO string.

Eventually, you may have Point features with a 'times' property being an array oflength1.

Parameters:

- **data**(*file,dictorstr.*)–

  Thetimestampedgeo-jsondatayouwanttoplot.

  - If file, then data will be read in the file and fully embedded in Leaflet'sjavascript.
  - If dict, then data will be converted to json and embedded in thejavascript.
  - Ifstr,then datawill bepassed tothe javascriptas-is.

- **transition_time** (*int, default 200.*) – The duration in ms of a transition frombetweentimestamps.
- **loop**(*bool,defaultTrue*)–Whethertheanimationshallloop.
- **auto_play** (*bool, default True*) – Whether the animation shall startautomaticallyatstartup.
- **add_last_point** (*bool, default True*) – Whether a point is added at the lastvalidcoordinate ofaLineString.
- **period** (*str, default "P1D"*) – Used to construct the array of available timesstarting from the first available time. Format: ISO8601 Duration ex: 'P1M'1/month,'P1D' 1/day, 'PT1H' 1/hour,and 'PT1M' 1/minute
- **duration** (*str, default None*) – Period of time which the features will be shownon the map after their time has passed. If None, all previous times will beshown. Format: ISO8601 Duration ex: 'P1M' 1/month, 'P1D' 1/day, 'PT1H'1/hour,and 'PT1M' 1/minute

Examples

```
>>>TimestampedGeoJson(
...     {
...         "type":"FeatureCollection",
...         "features":[
...             {
...                 "type":"Feature",
...                 "geometry":{
...                     "type":"LineString",
...                     "coordinates":[[-70,-25],[-70,35],
[70,35]],
...                 },
...                 "properties":{
...                     "times":[1435708800000,1435795200000,
1435881600000],
...                     "tooltip":"mytooltiptext",
...                 },
...             }
...         ],
...     }
...)
```

Seesocib/Leaflet.TimeDimensionformoreinformation.

default_css*:List[Tuple[str,str]]*

default_js*: List[Tuple[str,*

str]]*render(kwargs*)

RendersthеHTMLrepresentationoftheelement.

***class*folium.plugins.TimestampedWmsTileLayers(*data*,*transition_time*=*200*,*loop*=*False*,*auto_play*=*False*,*period*=*'P1D'*,*time_interval*=*False*)**

Bases:**JSCSSMixin**,**MacroElement**

CreatesaTimestampedWmsTileLayerthattakesaWmsTileLayerandaddstimecontrolwithhe Leaflet.TimeDimension plugin.

Parameters:

- **data** (*WmsTileLayer.*) – The WmsTileLayer that you want to add time supportto. Must be created like a typical WmsTileLayer and added to the map beforebeingpassed to this class.
- **transition_time** (*int, default 200.*) – The duration in ms of a transition frombetweentimestamps.
- **loop** (*bool, default False*) – Whether the animation shall loop, default is toreduceload onWMS services.
- **auto_play** (*bool, default False*) – Whether the animation shall startautomaticallyatstartup,default istoreduceloadonWMSservices.
- **period** (*str, default 'P1D'*) – Used to construct the array of available timesstarting from the first available time. Format: ISO8601 Duration ex: 'P1M' ->1/month, 'P1D' -> 1/day, 'PT1H' -> 1/hour, and 'PT1M' -> 1/minute Note:thisseems tobe overriddenby theWMSTile LayerGetCapabilities.

Examples

```
>>>w0=WmsTileLayer(
...     "http://this.wms.server/ncWMS/wms",
...     name="TestWMSData",
...     styles="",
...     fmt="image/png",
...     transparent=True,
...     layers="test_data",
...     COLORSCALERANGE="0,10",
...)
>>>w0.add_to(m)
>>>w1=WmsTileLayer(
...     "http://this.wms.server/ncWMS/wms",
...     name="TestWMSData",
...     styles="",
...     fmt="image/png",
...     transparent=True,
...     layers="test_data_2",
...     COLORSCALERANGE="0,5",
...)
>>>w1.add_to(m)
>>>#AddWmsTileLayerstotimecontrol.
>>>time=TimestampedWmsTileLayers([w0,w1])
>>>time.add_to(m)
```

Seesocib/Leaflet.TimeDimensionformoreinformation.

default_css*:List[Tuple[str,str]]*

default_js*:List[Tuple[str,str]]*

***class*folium.plugins.VectorGridProtobuf(*url:str*,*name:str|None*=*None*,*options:str|dict|None*=*None*,*overlay:bool*=*True*,*control:bool*=*True*,*show:bool*=*True*)**

Bases:**JSCSSMixin**,**Layer**

Addvectortilelayersbasedon Leaflet/Leaflet.VectorGrid.

Parameters:

- **url**(*str*)–urlto tileprovidere.g.https://free-{s}.tilehosting.com/data/v3/{z}/{x}/{y}.pbf?token={token}
- **name** (*str, optional*) – Name of the layer that will be displayed inLayerControl
- **options** (*dictorstr,optional*)–

  VectorGrid.protobuf options, which you can pass as python dictionary orstring. Strings allow plain JavaScript to be passed, therefore allow forconditionalstyling (seeexamples).
  Additionally the url might contain any string literals like {token}, or {key}thatcanbepassedasattributeto theoptionsdictand willbesubstituted.
  Everylayer insidethetilelayerhastobestyledseparately.

- **overlay** (*bool, default True*) – Whether your layer will be an overlay (tickedwith a check box in LayerControls) or a base layer (ticked with a radiobutton).
- **control** (*bool, default True*) – Whether the layer will be included inLayerControls.
- **show**(*bool,default True*)–Whetherthelayerwillbeshownonopening.

Examples

Optionsasdict:

```
>>>m=folium.Map()
>>>url="https://free-
{s}.tilehosting.com/data/v3/{z}/{x}/{y}.pbf?token={token}"
>>>options={
...      "subdomain":"tilehosting",
...      "token":"af6P2G9dztAt1F75x7KYt0Hx2DJR052G",
...      "vectorTileLayerStyles":{
...          "layer_name_one":{
...              "fill":True,
...              "weight":1,
...              "fillColor":"green",
...              "color":"black",
...              "fillOpacity":0.6,
...              "opacity":0.6,
...          },
...          "layer_name_two":{
...              "fill":True,
...              "weight":1,
...              "fillColor":"red",
...              "color":"black",
...              "fillOpacity":0.6,
...              "opacity":0.6,
...          },
...      },
...}
>>>VectorGridProtobuf(url,"layer_name",options).add_to(m)
```

Optionsasstringallows topassfunctions

```
>>>m=folium.Map()
>>>url="https://free-
{s}.tilehosting.com/data/v3/{z}/{x}/{y}.pbf?token={token}"
>>>options='''{
...      "subdomain":"tilehosting",
...      "token":"af6P2G9dztAt1F75x7KYt0Hx2DJR052G",
...      "vectorTileLayerStyles":{
...          all:function(f){
...              if(f.type==='parks'){
...                  return{
...                      "fill":true,
...                      "weight":1,
```

```
...                         "fillColor":'green',
...                         "color":'black',
...                         "fillOpacity":0.6,
...                         "opacity":0.6
...                     };
...                 }
...                 if(f.type==='water'){
...                     return{
...                         "fill":true,
...                         "weight":1,
...                         "fillColor":'purple',
...                         "color":'black',
...                         "fillOpacity":0.6,
...                         "opacity":0.6
...                     };
...                 }
...             }
...         }
...}'''
>>>VectorGridProtobuf(url,"layer_name",options).add_to(m)
```

default_js:*List[Tuple[str,str]]*

ExpectedOutcome

Youshouldhaveamapcenteredonthelocationyouchose,withthelocationsdifferentiated by colour scheme. Have a look at the screenshot below forreference.