Using Jupyter

The Jupyter Book community

Apr 20, 2023

CONTENTS

Markdown cells

This is a Markdown (documentation) cell, for adding documentation and comments to a notebook. It uses Markdown, a text styling language, as well as HTML. When a Markdown cell is "run", it converts the contents of the cell into HTML, which is then rendered and displayed.

Double-click on a Markdown cell to see its source.

This is **more** *markdown*. Surround text with double asterisk for bold, or single underline for italic.

See the Notebook named **JupyterMarkdownGuide** for details.

Code Cells

Code cells contain Python (or other language) code. When a code cell is run, it executes the code, and shows the output below the cell. The output is retained unless you explicitly clear it.

```python
x = 3
```

```python
for i in range(3):
    print(i)
```

```
0
1
2
```

```python
print("wombat")
```

```
wombat
```

```python
print(x)
```

```
3
```

It's all one program

All code is run in the same instance of the Python interpreter, so that objects created in one cell are available to other cells, as long as the first cell has been run.

```python
from datetime import date as Date
```

```python
then = Date(2011,5,22)
```

```python
print(then.year)
```

```
2011
```

```
%pinfo then
```

Getting help

Putting a ? before (or after) any object displays help for that object. Using ?? will add more detailed help, if available. (The output will be in a separate pane at the bottom of the browser window).

```
i?
```

```
Date??
```

Using Python's scientific libraries

For typical use of Python's scientific libraries, put the following at the top of the notebook in a code cell:

Other modules and packages should be included as needed.

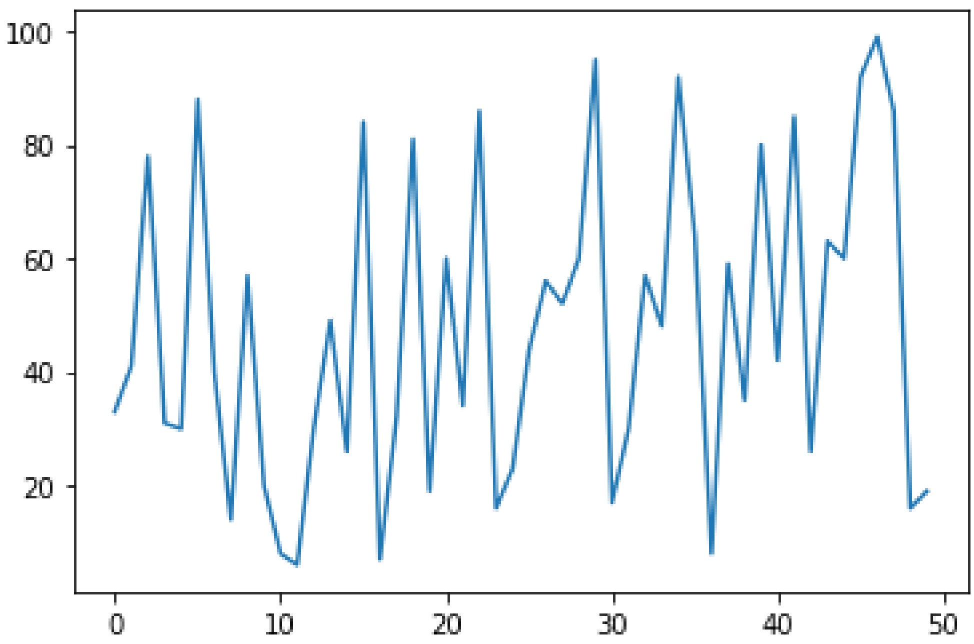
```
import pandas as pd
import scipy as sp
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
```

Inline plotting

After matplotlib is imported, use the **%matplotlib inline** magic to display figures as part of the notebook. Otherwise, they are displayed in popup windows.

```
%matplotlib inline
values = np.random.randint(1, 100, 50)
plt.plot(values)
```

```
[<matplotlib.lines.Line2D at 0x7fec61258790>]
```



Using HTML

Since Markdown is converted to HTML, any actual HTML in a Markdown cell is used.

Magics

iPython and Jupyter notebooks have *line magics*, which are line-oriented. Many of them execute commands or turn iPython settings on and off.

Jupyter has *cell magics*, which apply to the entire cell.

```
%lsmagic
```

```
Available line magics:
%alias  %alias_magic  %autoawait  %autocall  %automagic  %autosave  %bookmark
 ↪%cat  %cd  %clear  %colors  %conda  %config  %connect_info  %cp  %debug  %dhist
 ↪%dirs  %doctest_mode  %ed  %edit  %env  %gui  %hist  %history  %killbgscripts
 ↪%ldir  %less  %lf  %lk  %ll  %load  %load_ext  %loadpy  %logoff  %logon
 ↪%logstart  %logstate  %logstop  %ls  %lsmagic  %lx  %macro  %magic  %man
 ↪%matplotlib  %mkdir  %more  %mv  %notebook  %page  %pastebin  %pdb  %pdef  %pdoc␣
 ↪ %pfile  %pinfo  %pinfo2  %pip  %popd  %pprint  %precision  %prun  %psearch
 ↪%psource  %pushd  %pwd  %pycat  %pylab  %qtconsole  %quickref  %recall  %rehashx␣
 ↪ %reload_ext  %rep  %rerun  %reset  %reset_selective  %rm  %rmdir  %run  %save
 ↪%sc  %set_env  %store  %sx  %system  %tb  %time  %timeit  %unalias  %unload_ext
 ↪%who  %who_ls  %whos  %xdel  %xmode

Available cell magics:
%%!  %%HTML  %%SVG  %%bash  %%capture  %%debug  %%file  %%html  %%javascript  %%js␣
 ↪ %%latex  %%markdown  %%perl  %%prun  %%pypy  %%python  %%python2  %%python3  %
 ↪%ruby  %%script  %%sh  %%svg  %%sx  %%system  %%time  %%timeit  %%writefile
```

(continues on next page)

(continued from previous page)

```
Automagic is ON, % prefix IS NOT needed for line magics.
```

```
!hostname
h = !hostname
print(h)
```

```
Johns-Macbook.attlocal.net
```

```
['Johns-Macbook.attlocal.net']
```

Running external Python scripts

Use the **%run** magic to launch an external Python script

```
%run ../EXAMPLES/my_vars.py
```

```
<Figure size 432x288 with 0 Axes>
```

```
print(user_name)
print(animal)
print(snake)
```

```
Susan
wombat
Eastern Racer
```

Loading scripts into cells

Use the **%load** magic to read a separate Python script into the current cell. After it's loaded, it can be run like any other cell.

Once the code is loaded into the cell, the **%load** command is commented out

```
# %load ../EXAMPLES/read_tyger.py

with open("../DATA/tyger.txt", "r") as tyger_in:  # tyger_in is return value of open()
    for raw_line in tyger_in:  # tyger_in is iterable of lines in the file
        line = raw_line.rstrip()   # remove \n
        print(line)
```

```
           The Tyger

Tyger! Tyger! burning bright
```

(continues on next page)

(continued from previous page)

```
In the forests of the night,
What immortal hand or eye
Could frame thy fearful symmetry?

In what distant deeps or skies
Burnt the fire of thine eyes?
On what wings dare he aspire?
What the hand dare seize the fire?

And what shoulder, & what art,
Could twist the sinews of thy heart?
And when thy heart began to beat,
What dread hand? & what dread feet?

What the hammer? what the chain?
In what furnace was thy brain?
What the anvil? what dread grasp
Dare its deadly terrors clasp?

When the stars threw down their spears
And water'd heaven with their tears,
Did he smile his work to see?
Did he who made the Lamb make thee?

Tyger! Tyger! burning bright
In the forests of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?

                    by William Blake
```

```
%load imports.py
```

Using LaTeX

Markdown cells can render LaTeX via MathJax. Put the LaTeX code inside a pair of dollar signs: **ρ:**

ρ, ρ, ρ your boat

$$\mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \end{vmatrix}$$

$$\left(\sum_{k=1}^n a_k b_k \right)^2 \leq \left(\sum_{k=1}^n a_k^2 \right) \left(\sum_{k=1}^n b_k^2 \right)$$

Can you read this limerick?

$$\frac{12+144+20+3\sqrt{4}}{7} + (5*11) = 9^2 + 0$$

See text of limerick at the bottom of this notebook

Getting info

```
from scipy import stats
sp.info(stats)
```

```
.. _statsrefmanual:

==========================================
Statistical functions (:mod:`scipy.stats`)
==========================================

.. currentmodule:: scipy.stats

This module contains a large number of probability distributions,
summary and frequency statistics, correlation functions and statistical
tests, masked statistics, kernel density estimation, quasi-Monte Carlo
functionality, and more.

Statistics is a very large area, and there are topics that are out of scope
for SciPy and are covered by other packages. Some of the most important ones
are:

- `statsmodels <https://www.statsmodels.org/stable/index.html>`__:
  regression, linear models, time series analysis, extensions to topics
  also covered by ``scipy.stats``.
- `Pandas <https://pandas.pydata.org/>`__: tabular data, time series
  functionality, interfaces to other statistical languages.
- `PyMC <https://docs.pymc.io/>`__: Bayesian statistical
  modeling, probabilistic machine learning.
- `scikit-learn <https://scikit-learn.org/>`__: classification, regression,
  model selection.
- `Seaborn <https://seaborn.pydata.org/>`__: statistical data visualization.
- `rpy2 <https://rpy2.github.io/>`__: Python to R bridge.


Probability distributions
=========================

Each univariate distribution is an instance of a subclass of `rv_continuous`
(`rv_discrete` for discrete distributions):

.. autosummary::
   :toctree: generated/

   rv_continuous
   rv_discrete
   rv_histogram

Continuous distributions
------------------------

.. autosummary::
   :toctree: generated/

   alpha             -- Alpha
   anglit            -- Anglit
```

(continues on next page)

(continued from previous page)

```
  arcsine           -- Arcsine
  argus             -- Argus
  beta              -- Beta
  betaprime         -- Beta Prime
  bradford          -- Bradford
  burr              -- Burr (Type III)
  burr12            -- Burr (Type XII)
  cauchy            -- Cauchy
  chi               -- Chi
  chi2              -- Chi-squared
  cosine            -- Cosine
  crystalball       -- Crystalball
  dgamma            -- Double Gamma
  dweibull          -- Double Weibull
  erlang            -- Erlang
  expon             -- Exponential
  exponnorm         -- Exponentially Modified Normal
  exponweib         -- Exponentiated Weibull
  exponpow          -- Exponential Power
  f                 -- F (Snecdor F)
  fatiguelife       -- Fatigue Life (Birnbaum-Saunders)
  fisk              -- Fisk
  foldcauchy        -- Folded Cauchy
  foldnorm          -- Folded Normal
  genlogistic       -- Generalized Logistic
  gennorm           -- Generalized normal
  genpareto         -- Generalized Pareto
  genexpon          -- Generalized Exponential
  genextreme        -- Generalized Extreme Value
  gausshyper        -- Gauss Hypergeometric
  gamma             -- Gamma
  gengamma          -- Generalized gamma
  genhalflogistic   -- Generalized Half Logistic
  genhyperbolic     -- Generalized Hyperbolic
  geninvgauss       -- Generalized Inverse Gaussian
  gibrat            -- Gibrat
  gompertz          -- Gompertz (Truncated Gumbel)
  gumbel_r          -- Right Sided Gumbel, Log-Weibull, Fisher-Tippett, Extreme␣
↪Value Type I
  gumbel_l          -- Left Sided Gumbel, etc.
  halfcauchy        -- Half Cauchy
  halflogistic      -- Half Logistic
  halfnorm          -- Half Normal
  halfgennorm       -- Generalized Half Normal
  hypsecant         -- Hyperbolic Secant
  invgamma          -- Inverse Gamma
  invgauss          -- Inverse Gaussian
  invweibull        -- Inverse Weibull
  johnsonsb         -- Johnson SB
  johnsonsu         -- Johnson SU
  kappa4            -- Kappa 4 parameter
  kappa3            -- Kappa 3 parameter
  ksone             -- Distribution of Kolmogorov-Smirnov one-sided test statistic
  kstwo             -- Distribution of Kolmogorov-Smirnov two-sided test statistic
  kstwobign         -- Limiting Distribution of scaled Kolmogorov-Smirnov two-
↪sided test statistic.
```

(continues on next page)

(continued from previous page)

```
   laplace           -- Laplace
   laplace_asymmetric    -- Asymmetric Laplace
   levy              -- Levy
   levy_l
   levy_stable
   logistic          -- Logistic
   loggamma          -- Log-Gamma
   loglaplace        -- Log-Laplace (Log Double Exponential)
   lognorm           -- Log-Normal
   loguniform        -- Log-Uniform
   lomax             -- Lomax (Pareto of the second kind)
   maxwell           -- Maxwell
   mielke            -- Mielke's Beta-Kappa
   moyal             -- Moyal
   nakagami          -- Nakagami
   ncx2              -- Non-central chi-squared
   ncf               -- Non-central F
   nct               -- Non-central Student's T
   norm              -- Normal (Gaussian)
   norminvgauss      -- Normal Inverse Gaussian
   pareto            -- Pareto
   pearson3          -- Pearson type III
   powerlaw          -- Power-function
   powerlognorm      -- Power log normal
   powernorm         -- Power normal
   rdist             -- R-distribution
   rayleigh          -- Rayleigh
   rice              -- Rice
   recipinvgauss     -- Reciprocal Inverse Gaussian
   semicircular      -- Semicircular
   skewcauchy        -- Skew Cauchy
   skewnorm          -- Skew normal
   studentized_range    -- Studentized Range
   t                 -- Student's T
   trapezoid         -- Trapezoidal
   triang            -- Triangular
   truncexpon        -- Truncated Exponential
   truncnorm         -- Truncated Normal
   truncweibull_min  -- Truncated minimum Weibull distribution
   tukeylambda       -- Tukey-Lambda
   uniform           -- Uniform
   vonmises          -- Von-Mises (Circular)
   vonmises_line     -- Von-Mises (Line)
   wald              -- Wald
   weibull_min       -- Minimum Weibull (see Frechet)
   weibull_max       -- Maximum Weibull (see Frechet)
   wrapcauchy        -- Wrapped Cauchy

Multivariate distributions
==========================

.. autosummary::
   :toctree: generated/

   multivariate_normal    -- Multivariate normal distribution
   matrix_normal          -- Matrix normal distribution
```

(continues on next page)

(continued from previous page)

```
   dirichlet              -- Dirichlet
   wishart                -- Wishart
   invwishart             -- Inverse Wishart
   multinomial            -- Multinomial distribution
   special_ortho_group    -- SO(N) group
   ortho_group            -- O(N) group
   unitary_group          -- U(N) group
   random_correlation     -- random correlation matrices
   multivariate_t         -- Multivariate t-distribution
   multivariate_hypergeom -- Multivariate hypergeometric distribution

Discrete distributions
----------------------

.. autosummary::
   :toctree: generated/

   bernoulli                -- Bernoulli
   betabinom                -- Beta-Binomial
   binom                    -- Binomial
   boltzmann                -- Boltzmann (Truncated Discrete Exponential)
   dlaplace                 -- Discrete Laplacian
   geom                     -- Geometric
   hypergeom                -- Hypergeometric
   logser                   -- Logarithmic (Log-Series, Series)
   nbinom                   -- Negative Binomial
   nchypergeom_fisher       -- Fisher's Noncentral Hypergeometric
   nchypergeom_wallenius    -- Wallenius's Noncentral Hypergeometric
   nhypergeom               -- Negative Hypergeometric
   planck                   -- Planck (Discrete Exponential)
   poisson                  -- Poisson
   randint                  -- Discrete Uniform
   skellam                  -- Skellam
   yulesimon                -- Yule-Simon
   zipf                     -- Zipf (Zeta)
   zipfian                  -- Zipfian

An overview of statistical functions is given below.  Many of these functions
have a similar version in `scipy.stats.mstats` which work for masked arrays.

Summary statistics
==================

.. autosummary::
   :toctree: generated/

   describe          -- Descriptive statistics
   gmean             -- Geometric mean
   hmean             -- Harmonic mean
   pmean             -- Power mean
   kurtosis          -- Fisher or Pearson kurtosis
   mode              -- Modal value
   moment            -- Central moment
   skew              -- Skewness
   kstat             --
   kstatvar          --
```

(continues on next page)

(continued from previous page)

```
   tmean             -- Truncated arithmetic mean
   tvar              -- Truncated variance
   tmin              --
   tmax              --
   tstd              --
   tsem              --
   variation         -- Coefficient of variation
   find_repeats
   trim_mean
   gstd              -- Geometric Standard Deviation
   iqr
   sem
   bayes_mvs
   mvsdist
   entropy
   differential_entropy
   median_abs_deviation

Frequency statistics
====================

.. autosummary::
   :toctree: generated/

   cumfreq
   percentileofscore
   scoreatpercentile
   relfreq

.. autosummary::
   :toctree: generated/

   binned_statistic     -- Compute a binned statistic for a set of data.
   binned_statistic_2d  -- Compute a 2-D binned statistic for a set of data.
   binned_statistic_dd  -- Compute a d-D binned statistic for a set of data.

Correlation functions
=====================

.. autosummary::
   :toctree: generated/

   f_oneway
   alexandergovern
   pearsonr
   spearmanr
   pointbiserialr
   kendalltau
   weightedtau
   somersd
   linregress
   siegelslopes
   theilslopes
   multiscale_graphcorr

Statistical tests
```

(continues on next page)

(continued from previous page)

```
=================

.. autosummary::
   :toctree: generated/

   ttest_1samp
   ttest_ind
   ttest_ind_from_stats
   ttest_rel
   chisquare
   cramervonmises
   cramervonmises_2samp
   power_divergence
   kstest
   ks_1samp
   ks_2samp
   epps_singleton_2samp
   mannwhitneyu
   tiecorrect
   rankdata
   ranksums
   wilcoxon
   kruskal
   friedmanchisquare
   brunnermunzel
   combine_pvalues
   jarque_bera
   page_trend_test
   tukey_hsd

.. autosummary::
   :toctree: generated/

   ansari
   bartlett
   levene
   shapiro
   anderson
   anderson_ksamp
   binom_test
   binomtest
   fligner
   median_test
   mood
   skewtest
   kurtosistest
   normaltest


Quasi-Monte Carlo
=================

.. toctree::
   :maxdepth: 4

   stats.qmc
```

(continues on next page)

(continued from previous page)

```
Resampling Methods
==================

.. autosummary::
   :toctree: generated/

   bootstrap
   permutation_test
   monte_carlo_test

Masked statistics functions
===========================

.. toctree::

   stats.mstats


Other statistical functionality
===============================

Transformations
---------------

.. autosummary::
   :toctree: generated/

   boxcox
   boxcox_normmax
   boxcox_llf
   yeojohnson
   yeojohnson_normmax
   yeojohnson_llf
   obrientransform
   sigmaclip
   trimboth
   trim1
   zmap
   zscore
   gzscore

Statistical distances
---------------------

.. autosummary::
   :toctree: generated/

   wasserstein_distance
   energy_distance

Sampling
--------

.. toctree::
   :maxdepth: 4
```

(continues on next page)

(continued from previous page)

```
   stats.sampling

Random variate generation / CDF Inversion
-----------------------------------------

.. autosummary::
   :toctree: generated/

   rvs_ratio_uniforms

Distribution Fitting
--------------------

.. autosummary::
   :toctree: generated/

   fit

Circular statistical functions
------------------------------

.. autosummary::
   :toctree: generated/

   circmean
   circvar
   circstd

Contingency table functions
---------------------------

.. autosummary::
   :toctree: generated/

   chi2_contingency
   contingency.crosstab
   contingency.expected_freq
   contingency.margins
   contingency.relative_risk
   contingency.association
   fisher_exact
   barnard_exact
   boschloo_exact

Plot-tests
----------

.. autosummary::
   :toctree: generated/

   ppcc_max
   ppcc_plot
   probplot
   boxcox_normplot
   yeojohnson_normplot
```

(continues on next page)

(continued from previous page)

```
Univariate and multivariate kernel density estimation
-----------------------------------------------------

.. autosummary::
   :toctree: generated/

   gaussian_kde

Warnings / Errors used in :mod:`scipy.stats`
--------------------------------------------

.. autosummary::
   :toctree: generated/

   DegenerateDataWarning
   ConstantInputWarning
   NearConstantInputWarning
   FitError
```

```
/var/folders/p7/_ryqngjd3jn_ppndvnhdzqch0000gn/T/ipykernel_33503/3286860601.py:2:␣
 ↪DeprecationWarning: scipy.info is deprecated and will be removed in SciPy 2.0.0,␣
 ↪use numpy.info instead
  sp.info(stats)
```

Benchmarking

The **%%timeit** cell magic will execute the code in the cell and report the average time it took to execute it.

```
fruits = ["pomegranate", "cherry", "apricot", "date", "Apple",
"lemon", "Kiwi", "ORANGE", "lime", "Watermelon", "guava",
"Papaya", "FIG", "pear", "banana", "Tamarind", "Persimmon",
"elderberry", "peach", "BLUEberry", "lychee", "GRAPE" ]
```

Benchmark with *for* loop

```
%%timeit 100
f1 = []
for f in fruits:
    f1.append(f[:3])
```

```
2.51 µs ± 23.4 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

Benchmark with list comprehension

```
%%timeit 100
f2 = [f[:3] for f in fruits]
```

```
3.23 µs ± 1.92 µs per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

Images

You can insert images into doc cells using the Markdown image tag:

The following uses a Markdown table to arrange the images.

Guido	Tim the Enchanter	Wombat

The limerick

A dozen, a gross, and a score Plus three times the square root of four Divided by seven Plus five times eleven Is nine squared and not a bit more.