

Quick start

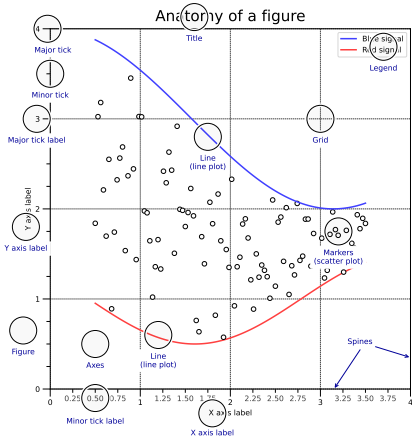
```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
X = np.linspace(0, 2*np.pi, 100)
Y = np.cos(X)
```

```
fig, ax = plt.subplots()
ax.plot(X, Y, color='green')
```

```
fig.savefig("figure.pdf")
plt.show()
```

Anatomy of a figure



Subplots layout

```
subplot[s](rows, cols, ...)
fig, axs = plt.subplots(3, 3)
```

```
G = gridspec(rows, cols, ...)
ax = G[0, :]
```

```
ax.inset_axes(extent)
```

```
d=make_axes_locatable(ax)
ax = d.new_horizontal('10%')
```

Getting help

- matplotlib.org
- github.com/matplotlib/matplotlib/issues
- discourse.matplotlib.org
- stackoverflow.com/questions/tagged/matplotlib
- https://gitter.im/matplotlib/matplotlib
- twitter.com/matplotlib
- Matplotlib users mailing list

Basic plots

```
plot([X], Y, [fmt], ...)
X, Y, fmt, color, marker, linestyle
```

```
scatter(X, Y, ...)
X, Y, [s]izes, [c]olors, marker, cmap
```

```
bar[h](x, height, ...)
x, height, width, bottom, align, color
```

```
imshow(Z, ...)
Z, cmap, interpolation, extent, origin
```

```
contour[f]([X], [Y], Z, ...)
X, Y, Z, levels, colors, extent, origin
```

```
pcolormesh([X], [Y], Z, ...)
X, Y, Z, vmin, vmax, cmap
```

```
quiver([X], [Y], U, V, ...)
X, Y, U, V, C, units, angles
```

```
pie(X, ...)
Z, explode, labels, colors, radius
```

```
text(x, y, text, ...)
x, y, text, va, ha, size, weight, transform
```

```
fill[_between][x](...)
X, Y1, Y2, color, where
```

Advanced plots

```
step(X, Y, [fmt], ...)
X, Y, fmt, color, marker, where
```

```
boxplot(X, ...)
X, notch, sym, bootstrap, widths
```

```
errorbar(X, Y, xerr, yerr, ...)
X, Y, xerr, yerr, fmt
```

```
hist(X, bins, ...)
X, bins, range, density, weights
```

```
violinplot(D, ...)
D, positions, widths, vert
```

```
barbs([X], [Y], U, V, ...)
X, Y, U, V, C, length, pivot, sizes
```

```
eventplot(positions, ...)
positions, orientation, lineoffsets
```

```
hexbin(X, Y, C, ...)
X, Y, C, gridsz, bins
```

Scales

```
ax.set_[xy]scale(scale, ...)
linear any values
symlog any values
log values > 0
logit 0 < values < 1
```

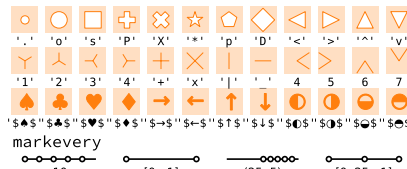
Projections

```
subplot(..., projection=p)
p='polar'
p='3d'
p=ccrs.Orthographic()
import cartopy.crs as ccrs
```

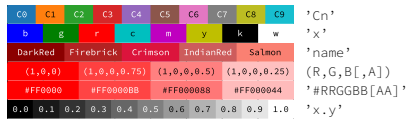
Lines

```
linestyle or ls
capstyle or dash_capstyle
"butt" "round" "projecting"
```

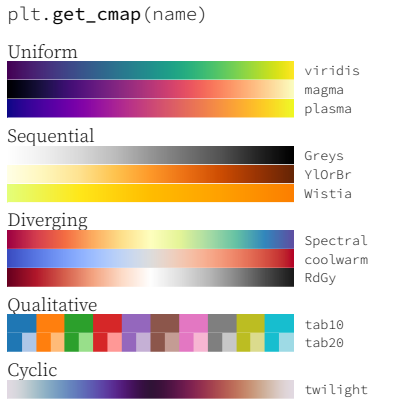
Markers



Colors



Colormaps



Tick locators

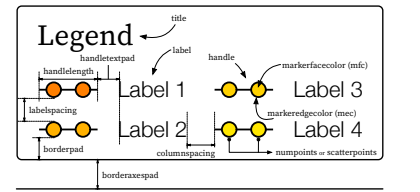
```
from matplotlib import ticker
ax.[xy]axis.set_[minor|major]_locator(locator)
ticker.NullLocator()
ticker.MultipleLocator(0.5)
ticker.FixedLocator([0, 1, 5])
ticker.LinearLocator(numticks=3)
ticker.IndexLocator(base=0.5, offset=0.25)
ticker.AutoLocator()
ticker.MaxNLocator(n=4)
ticker.LogLocator(base=10, numticks=15)
```

Tick formatters

```
from matplotlib import ticker
ax.[xy]axis.set_[minor|major]_formatter(formatter)
ticker.NullFormatter()
ticker.FixedFormatter(['zero', 'one', 'two', ...])
ticker.FuncFormatter(lambda x, pos: "[%2f]" % x)
ticker.FormatStrFormatter('>%d<')
ticker.ScalarFormatter()
ticker.StrMethodFormatter('{x}')
ticker.PercentFormatter(xmax=5)
```

Ornaments

```
ax.legend(...)
handles, labels, loc, title, frameon
```



```
ax.colorbar(...)
mappable, ax, cax, orientation
```

```
ax.annotate(...)
text, xytext, xycoords, textcoords, arrowprops
```

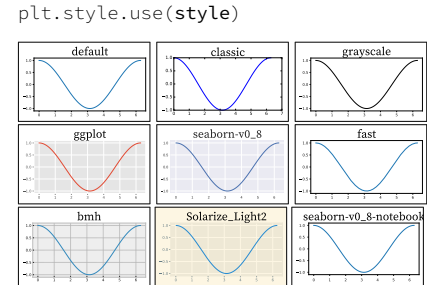
Event handling

```
fig, ax = plt.subplots()
def on_click(event):
    print(event)
fig.canvas.mpl_connect('button_press_event', on_click)
```

Animation

```
import matplotlib.animation as mpla
T = np.linspace(0, 2*np.pi, 100)
S = np.sin(T)
line, = plt.plot(T, S)
def animate(i):
    line.set_ydata(np.sin(T+i/50))
anim = mpla.FuncAnimation(plt.gcf(), animate, interval=5)
plt.show()
```

Styles



Quick reminder

```
ax.grid()
ax.set_[xy]lim(vmin, vmax)
ax.set_[xy]label(label)
ax.set_[xy]ticks(ticks, [labels])
ax.set_[xy]ticklabels(labels)
ax.set_title(title)
ax.tick_params(width=10, ...)
ax.set_axis_[on|off]()
fig.suptitle(title)
fig.tight_layout()
plt.gcf(), plt.gca()
mpl.rc('axes', linewidth=1, ...)
[fig|ax].patch.set_alpha(0)
text=r'$\frac{e^{-i\pi}}{2}$'
```

Keyboard shortcuts

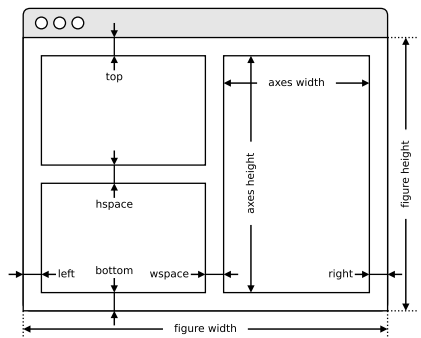
ctrl+s	Save	ctrl+w	Close plot
r	Reset view	f	Fullscreen 0/1
f	View forward	b	View back
p	Pan view	o	Zoom to rect
x	X pan/zoom	y	Y pan/zoom
g	Minor grid 0/1	G	Major grid 0/1
l	X axis log/linear	L	Y axis log/linear

Ten simple rules

1. Know your audience
2. Identify your message
3. Adapt the figure
4. Captions are not optional
5. Do not trust the defaults
6. Use color effectively
7. Do not mislead the reader
8. Avoid "chartjunk"
9. Message trumps beauty
10. Get the right tool

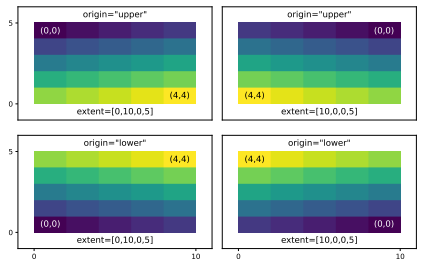
Axes adjustments API

plt.subplots_adjust(...)



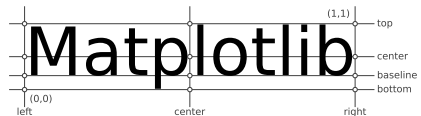
Extent & origin API

ax.imshow(extent=..., origin=...)



Text alignments API

ax.text(..., ha=..., va=..., ...)

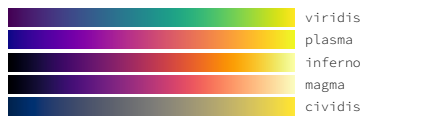


Text parameters API

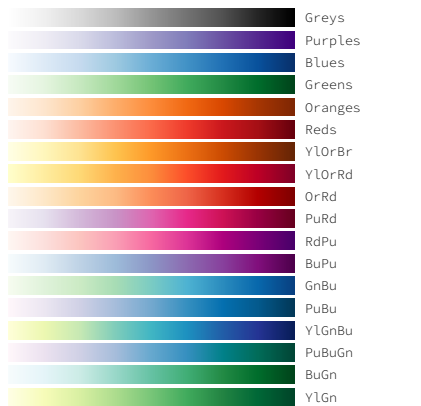
ax.text(..., family=..., size=..., weight=...)
ax.text(..., fontproperties=...)



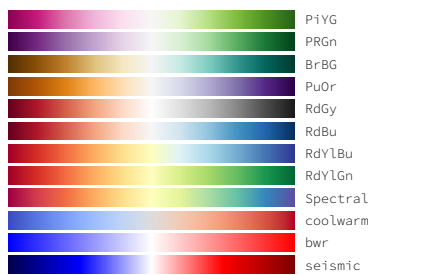
Uniform colormaps



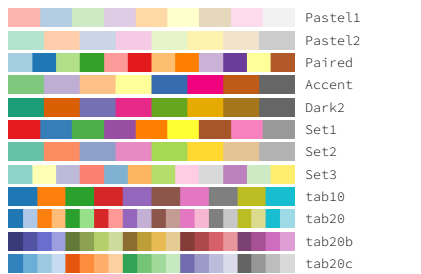
Sequential colormaps



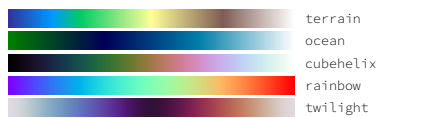
Diverging colormaps



Qualitative colormaps



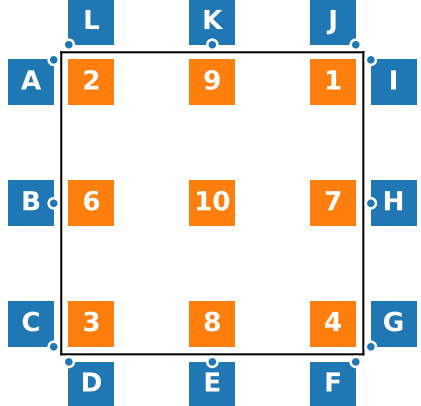
Miscellaneous colormaps



Color names API



Legend placement



ax.legend(loc="string", bbox_to_anchor=(x, y))

2: upper left 9: upper center 1: upper right
 6: center left 10: center 7: center right
 3: lower left 8: lower center 4: lower right

A: upper right / (-0.1, 0.9) B: center right / (-0.1, 0.5)
 C: lower right / (-0.1, 0.1) D: upper left / (0.1, -0.1)
 E: upper center / (0.5, -0.1) F: upper right / (0.9, -0.1)
 G: lower left / (1.1, 0.1) H: center left / (1.1, 0.5)
 I: upper left / (1.1, 0.9) J: lower right / (0.9, 1.1)
 K: lower center / (0.5, 1.1) L: lower left / (0.1, 1.1)

How do I ...

- resize a figure? → fig.set_size_inches(w, h)
- save a figure? → fig.savefig("figure.pdf")
- save a transparent figure? → fig.savefig("figure.pdf", transparent=True)
- clear a figure/an axes? → fig.clear() → ax.clear()
- close all figures? → plt.close("all")
- remove ticks? → ax.set_xticks([])
- remove tick labels? → ax.set_xticklabels([])
- rotate tick labels? → ax.tick_params(axis="x", rotation=90)
- hide top spine? → ax.spines['top'].set_visible(False)
- hide legend border? → ax.legend(frameon=False)
- show error as shaded region? → ax.fill_between(X, Y+error, Y-error)
- draw a rectangle? → ax.add_patch(plt.Rectangle((0, 0), 1, 1))
- draw a vertical line? → ax.axvline(x=0.5)
- draw outside frame? → ax.plot(..., clip_on=False)
- use transparency? → ax.plot(..., alpha=0.25)
- convert an RGB image into a gray image? → gray = 0.2989*R + 0.5870*G + 0.1140*B
- set figure background color? → fig.patch.set_facecolor("grey")
- get a reversed colormap? → plt.get_cmap("viridis_r")
- get a discrete colormap? → plt.get_cmap("viridis", 10)
- show a figure for one second? → fig.show(block=False), time.sleep(1)

Performance tips

```
scatter(X, Y, marker="o", ls="")    slow
plot(X, Y, marker="o", ls="")    fast
for i in range(n): plot(i, X[i], "o")    slow
plot(X, marker="o", ls="")    fast
cla(); imshow(...); canvas.draw()    slow
im.set_data(...); canvas.draw()    fast
```

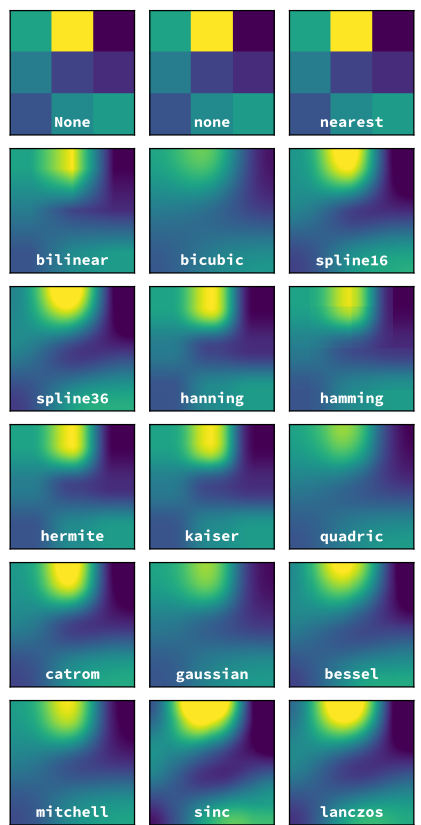
Beyond Matplotlib

- Seaborn: Statistical data visualization
- Cartopy: Geospatial data processing
- yt: Volumetric data visualization
- mpld3: Bringing Matplotlib to the browser
- Datashader: Large data processing pipeline
- plotnine: A grammar of graphics for Python

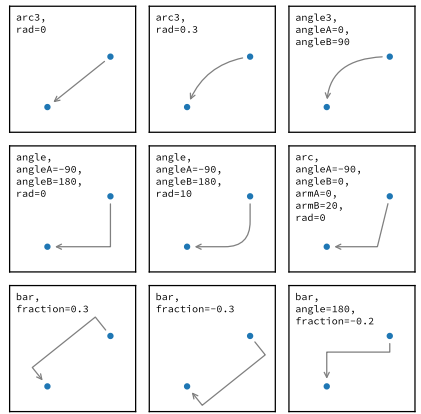
Matplotlib Cheatsheets
 Copyright (c) 2021 Matplotlib Development Team
 Released under a CC-BY 4.0 International License



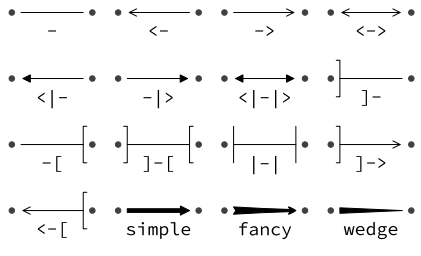
Image interpolation API



Annotation connection styles API



Annotation arrow styles API



Data Wrangling

with pandas Cheat Sheet
<http://pandas.pydata.org>

Pandas [API Reference](#) Pandas [User Guide](#)

Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

Specify values for each row.

		a	b	c
N	v			
D	1	4	7	10
	2	5	8	11
e	2	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [('d', 1), ('d', 2),
         ('e', 2)], names=['n', 'v']))
```

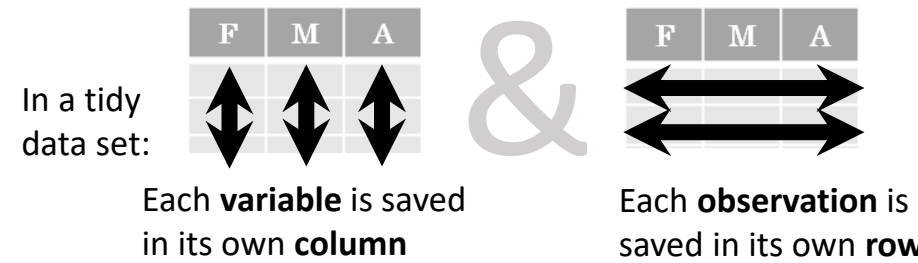
Create DataFrame with a MultiIndex

Method Chaining

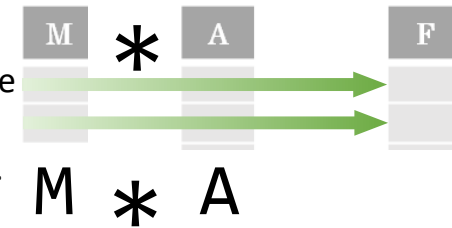
Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
      .rename(columns={
          'variable': 'var',
          'value': 'val'})
      .query('val >= 200'))
```

Tidy Data – A foundation for wrangling in pandas



Tidy data complements pandas's **vectorized operations**. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.



Reshaping Data – Change layout, sorting, reindexing, renaming



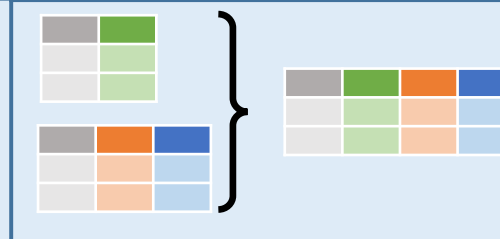
`pd.melt(df)`
Gather columns into rows.



`df.pivot(columns='var', values='val')`
Spread rows into columns.



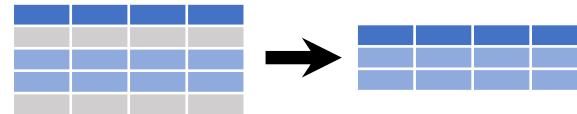
`pd.concat([df1, df2])`
Append rows of DataFrames



`pd.concat([df1, df2], axis=1)`
Append columns of DataFrames

- `df.sort_values('mpg')`
Order rows by values of a column (low to high).
- `df.sort_values('mpg', ascending=False)`
Order rows by values of a column (high to low).
- `df.rename(columns = {'y': 'year'})`
Rename the columns of a DataFrame
- `df.sort_index()`
Sort the index of a DataFrame
- `df.reset_index()`
Reset index of DataFrame to row numbers, moving index to columns.
- `df.drop(columns=['Length', 'Height'])`
Drop columns from DataFrame

Subset Observations - rows



- `df[df.Length > 7]`
Extract rows that meet logical criteria.
- `df.drop_duplicates()`
Remove duplicate rows (only considers columns).
- `df.sample(frac=0.5)`
Randomly select fraction of rows.
- `df.sample(n=10)`
Randomly select n rows.
- `df.nlargest(n, 'value')`
Select and order top n entries.
- `df.nsmallest(n, 'value')`
Select and order bottom n entries.
- `df.head(n)`
Select first n rows.
- `df.tail(n)`
Select last n rows.

Subset Variables - columns



- `df[['width', 'length', 'species']]`
Select multiple columns with specific names.
- `df['width']` or `df.width`
Select single column with specific name.
- `df.filter(regex='regex')`
Select columns whose name matches regular expression *regex*.

Using query

- `query()` allows Boolean expressions for filtering rows.
- `df.query('Length > 7')`
- `df.query('Length > 7 and Width < 8')`
- `df.query('Name.str.startswith("abc")', engine="python")`

Subsets - rows and columns

- Use `df.loc[]` and `df.iloc[]` to select only rows, only columns or both.
- Use `df.at[]` and `df.iat[]` to access a single value by row and column. First index selects rows, second index columns.
- `df.iloc[10:20]`
Select rows 10-20.
- `df.iloc[:, [1, 2, 5]]`
Select columns in positions 1, 2 and 5 (first column is 0).
- `df.loc[:, 'x2': 'x4']`
Select all columns between x2 and x4 (inclusive).
- `df.loc[df['a'] > 10, ['a', 'c']]`
Select rows meeting logical condition, and only the specific columns.
- `df.iat[1, 2]` Access single value by index
- `df.at[4, 'A']` Access single value by label

Logic in Python (and pandas)

<	Less than	<code>!=</code>	Not equal to
>	Greater than	<code>df.column.isin(values)</code>	Group membership
==	Equals	<code>pd.isnull(obj)</code>	Is NaN
<=	Less than or equals	<code>pd.notnull(obj)</code>	Is not NaN
>=	Greater than or equals	<code>&, , ~, ^, df.any(), df.all()</code>	Logical and, or, not, xor, any, all

regex (Regular Expressions) Examples

<code>'\.'</code>	Matches strings containing a period '.'
<code>'Length\$'</code>	Matches strings ending with word 'Length'
<code>'^Sepal'</code>	Matches strings beginning with the word 'Sepal'
<code>'^x[1-5]\$'</code>	Matches strings beginning with 'x' and ending with 1,2,3,4,5
<code>'^(?!Species\$).*</code>	Matches strings except the string 'Species'

Summarize Data

`df['w'].value_counts()`

Count number of rows with each unique value of variable

`len(df)`

of rows in DataFrame.

`df.shape`

Tuple of # of rows, # of columns in DataFrame.

`df['w'].nunique()`

of distinct values in a column.

`df.describe()`

Basic descriptive and statistics for each column (or GroupBy).



pandas provides a large set of **summary functions** that operate on different kinds of pandas objects (DataFrame columns, Series, GroupBy, Expanding and Rolling (see below)) and produce single values for each of the groups. When applied to a DataFrame, the result is returned as a pandas Series for each column. Examples:

`sum()`

Sum values of each object.

`count()`

Count non-NA/null values of each object.

`median()`

Median value of each object.

`quantile([0.25,0.75])`

Quantiles of each object.

`apply(function)`

Apply function to each object.

`min()`

Minimum value in each object.

`max()`

Maximum value in each object.

`mean()`

Mean value of each object.

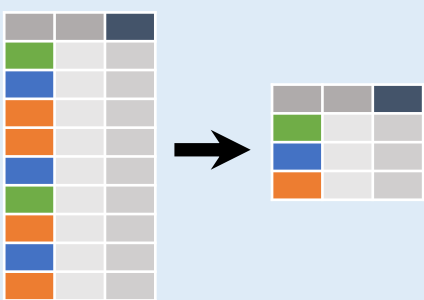
`var()`

Variance of each object.

`std()`

Standard deviation of each object.

Group Data



`df.groupby(by="col")`

Return a GroupBy object, grouped by values in column named "col".

`df.groupby(level="ind")`

Return a GroupBy object, grouped by values in index level named "ind".

All of the summary functions listed above can be applied to a group. Additional GroupBy functions:

`size()`

Size of each group.

`agg(function)`

Aggregate group using function.

Windows

`df.expanding()`

Return an Expanding object allowing summary functions to be applied cumulatively.

`df.rolling(n)`

Return a Rolling object allowing summary functions to be applied to windows of length n.

Handling Missing Data

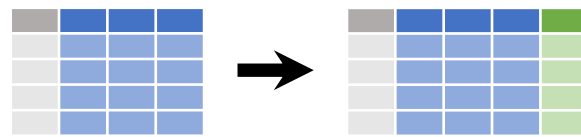
`df.dropna()`

Drop rows with any column having NA/null data.

`df.fillna(value)`

Replace all NA/null data with value.

Make New Columns



`df.assign(Area=lambda df: df.Length*df.Height)`

Compute and append one or more new columns.

`df['Volume'] = df.Length*df.Height*df.Depth`

Add single column.

`pd.qcut(df.col, n, labels=False)`

Bin column into n buckets.



pandas provides a large set of **vector functions** that operate on all columns of a DataFrame or a single selected column (a pandas Series). These functions produce vectors of values for each of the columns, or a single Series for the individual Series. Examples:

`max(axis=1)`

Element-wise max.

`min(axis=1)`

Element-wise min.

`clip(lower=-10, upper=10)`

Trim values at input thresholds

`abs()`

Absolute value.

The examples below can also be applied to groups. In this case, the function is applied on a per-group basis, and the returned vectors are of the length of the original DataFrame.

`shift(1)`

Copy with values shifted by 1.

`shift(-1)`

Copy with values lagged by 1.

`rank(method='dense')`

Ranks with no gaps.

`cumsum()`

Cumulative sum.

`rank(method='min')`

Ranks. Ties get min rank.

`cummax()`

Cumulative max.

`rank(pct=True)`

Ranks rescaled to interval [0, 1].

`cummin()`

Cumulative min.

`rank(method='first')`

Ranks. Ties go to first value.

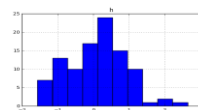
`cumprod()`

Cumulative product.

Plotting

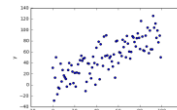
`df.plot.hist()`

Histogram for each column



`df.plot.scatter(x='w', y='h')`

Scatter chart using pairs of points



Combine Data Sets

`adf`

x1	x2
A	1
B	2
C	3



`bdf`

x1	x3
A	T
B	F
D	T



Standard Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NaN

`pd.merge(adf, bdf, how='left', on='x1')`
Join matching rows from bdf to adf.

x1	x2	x3
A	1.0	T
B	2.0	F
D	NaN	T

`pd.merge(adf, bdf, how='right', on='x1')`
Join matching rows from adf to bdf.

x1	x2	x3
A	1	T
B	2	F

`pd.merge(adf, bdf, how='inner', on='x1')`
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NaN
D	NaN	T

`pd.merge(adf, bdf, how='outer', on='x1')`
Join data. Retain all values, all rows.

Filtering Joins

x1	x2
A	1
B	2

`adf[adf.x1.isin(bdf.x1)]`
All rows in adf that have a match in bdf.

x1	x2
C	3

`adf[~adf.x1.isin(bdf.x1)]`
All rows in adf that do not have a match in bdf.

`ydf`

x1	x2
A	1
B	2
C	3



`zdf`

x1	x2
B	2
C	3
D	4



Set-like Operations

x1	x2
B	2
C	3

`pd.merge(ydf, zdf)`
Rows that appear in both ydf and zdf (Intersection).

x1	x2
A	1
B	2
C	3
D	4

`pd.merge(ydf, zdf, how='outer')`
Rows that appear in either or both ydf and zdf (Union).

x1	x2
A	1

`pd.merge(ydf, zdf, how='outer', indicator=True)`
`.query('_merge == "left_only"')`
`.drop(columns=['_merge'])`
Rows that appear in ydf but not zdf (Setdiff).

Таблица значений интегральной функции Лапласа

x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$
0,00	0,0000	0,50	0,1915	1,00	0,3413	1,50	0,4332	2,00	0,4772	3,00	0,49865
0,01	0,0040	0,51	0,1950	1,01	0,3438	1,51	0,4345	2,02	0,4783	3,20	0,49931
0,02	0,0080	0,52	0,1985	1,02	0,3461	1,52	0,4357	2,04	0,4793	3,40	0,49966
0,03	0,0120	0,53	0,2019	1,03	0,3485	1,53	0,4370	2,06	0,4803	3,60	0,499841
0,04	0,0160	0,54	0,2054	1,04	0,3508	1,54	0,4382	2,08	0,4812	3,80	0,499928
0,05	0,0199	0,55	0,2088	1,05	0,3531	1,55	0,4394	2,10	0,4821	4,00	0,499968
0,06	0,0239	0,56	0,2123	1,06	0,3554	1,56	0,4406	2,12	0,4830	4,50	0,499997
0,07	0,0279	0,57	0,2157	1,07	0,3577	1,57	0,4418	2,14	0,4838	5,00	0,499997
0,08	0,0319	0,58	0,2190	1,08	0,3599	1,58	0,4429	2,16	0,4846		
0,09	0,0359	0,59	0,2224	1,09	0,3621	1,59	0,4441	2,18	0,4854		
0,10	0,0398	0,60	0,2257	1,10	0,3643	1,60	0,4452	2,20	0,4861		
0,11	0,0438	0,61	0,2291	1,11	0,3665	1,61	0,4463	2,22	0,4868		
0,12	0,0478	0,62	0,2324	1,12	0,3686	1,62	0,4474	2,24	0,4875		
0,13	0,0517	0,63	0,2357	1,13	0,3708	1,63	0,4484	2,26	0,4881		
0,14	0,0557	0,64	0,2389	1,14	0,3729	1,64	0,4495	2,28	0,4887		
0,15	0,0596	0,65	0,2422	1,15	0,3749	1,65	0,4505	2,30	0,4893		
0,16	0,0636	0,66	0,2454	1,16	0,3770	1,66	0,4515	2,32	0,4898		
0,17	0,0675	0,67	0,2486	1,17	0,3790	1,67	0,4525	2,34	0,4904		
0,18	0,0714	0,68	0,2517	1,18	0,3810	1,68	0,4535	2,36	0,4909		
0,19	0,0753	0,69	0,2549	1,19	0,3830	1,69	0,4545	2,38	0,4913		
0,20	0,0793	0,70	0,2580	1,20	0,3849	1,70	0,4554	2,40	0,4918		
0,21	0,0832	0,71	0,2611	1,21	0,3869	1,71	0,4564	2,42	0,4922		
0,22	0,0871	0,72	0,2642	1,22	0,3883	1,72	0,4573	2,44	0,4927		
0,23	0,0910	0,73	0,2673	1,23	0,3907	1,73	0,4582	2,46	0,4931		
0,24	0,0948	0,74	0,2703	1,24	0,3925	1,74	0,4591	2,48	0,4934		
0,25	0,0987	0,75	0,2734	1,25	0,3944	1,75	0,4599	2,50	0,4938		
0,26	0,1026	0,76	0,2764	1,26	0,3962	1,76	0,4608	2,52	0,4941		
0,27	0,1064	0,77	0,2794	1,27	0,3980	1,77	0,4616	2,54	0,4945		
0,28	0,1103	0,78	0,2823	1,28	0,3997	1,78	0,4625	2,56	0,4948		
0,29	0,1141	0,79	0,2852	1,29	0,4015	1,79	0,4633	2,58	0,4951		
0,30	0,1179	0,80	0,2881	1,30	0,4032	1,80	0,4641	2,60	0,4953		
0,31	0,1217	0,81	0,2910	1,31	0,4049	1,81	0,4649	2,62	0,4956		
0,32	0,1255	0,82	0,2939	1,32	0,4066	1,82	0,4656	2,64	0,4959		
0,33	0,1293	0,83	0,2967	1,33	0,4082	1,83	0,4664	2,66	0,4961		
0,34	0,1331	0,84	0,2995	1,34	0,4099	1,84	0,4671	2,68	0,4963		
0,35	0,1368	0,85	0,3023	1,35	0,4115	1,85	0,4678	2,70	0,4965		
0,36	0,1406	0,86	0,3051	1,36	0,4131	1,86	0,4686	2,72	0,4967		
0,37	0,1443	0,87	0,3078	1,37	0,4147	1,87	0,4693	2,74	0,4969		
0,38	0,1480	0,88	0,3106	1,38	0,4162	1,88	0,4699	2,76	0,4971		
0,39	0,1517	0,89	0,3133	1,39	0,4177	1,89	0,4706	2,78	0,4973		
0,40	0,1554	0,90	0,3159	1,40	0,4192	1,90	0,4713	2,80	0,4974		
0,41	0,1591	0,91	0,3186	1,41	0,4207	1,91	0,4719	2,82	0,4976		
0,42	0,1628	0,92	0,3212	1,42	0,4222	1,92	0,4726	2,84	0,4977		
0,43	0,1664	0,93	0,3238	1,43	0,4236	1,93	0,4732	2,86	0,4979		
0,44	0,1700	0,94	0,3264	1,44	0,4251	1,94	0,4738	2,88	0,4980		
0,45	0,1736	0,95	0,3289	1,45	0,4265	1,95	0,4744	2,90	0,4981		
0,46	0,1772	0,96	0,3315	1,46	0,4279	1,96	0,4750	2,92	0,4982		
0,47	0,1808	0,97	0,3340	1,47	0,4292	1,97	0,4756	2,94	0,4984		
0,48	0,1844	0,98	0,3365	1,48	0,4306	1,98	0,4761	2,96	0,4985		
0,49	0,1879	0,99	0,3389	1,49	0,4319	1,99	0,4767	2,98	0,4986		

Таблица производных	Таблица интегралов
1. $(C)' = 1, (x)' = 1$	1. $\int dx = x + C$
2. $(x^\alpha)' = \alpha x^{\alpha-1}$	2. $\int x^k dx = \frac{x^{k+1}}{k+1} + C, (k \neq -1)$
3. $(\sqrt{x})' = \frac{1}{2\sqrt{x}}, \left(\frac{1}{x}\right)' = -\frac{1}{x^2}$	3. $\int \frac{dx}{\sqrt{x}} = 2\sqrt{x} + C, \int \frac{dx}{x} = \ln x + C$
4. $(a^x)' = a^x \ln a, (e^x)' = e^x$	4. $\int a^x dx = \frac{a^x}{\ln a} + C, \int e^x dx = e^x + C$
5. $(\log_a x)' = \frac{1}{x \ln a}, (\ln x)' = \frac{1}{x}$	5. $\int \frac{dx}{\sin x} = \ln \left \operatorname{tg} \frac{x}{2} \right + C$
6. $(\sin x)' = \cos x$	6. $\int \frac{dx}{\cos x} = \ln \left \operatorname{tg} \left(\frac{x}{2} + \frac{\pi}{4} \right) \right + C$
7. $(\cos x)' = -\sin x$	7. $\int \sin x dx = -\cos x + C$
8. $(\operatorname{tg} x)' = \frac{1}{\cos^2 x}$	8. $\int \cos x dx = \sin x + C$
9. $(\operatorname{ctg} x)' = -\frac{1}{\sin^2 x}$	9. $\int \frac{dx}{\cos^2 x} = \operatorname{tg} x + C$
10. $(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$	10. $\int \frac{dx}{\sin^2 x} = -\operatorname{ctg} x + C$
11. $(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$	11. $\int \frac{dx}{\sqrt{a^2-x^2}} = \arcsin \frac{x}{a} + C$
12. $(\operatorname{arctg} x)' = \frac{1}{1+x^2}$	12. $\int \frac{dx}{\sqrt{x^2 \pm a^2}} = \ln \left x + \sqrt{x^2 \pm a^2} \right + C$
13. $(\operatorname{arcctg} x)' = -\frac{1}{1+x^2}$	13. $\int \frac{dx}{x^2 - a^2} = \frac{1}{2a} \ln \left \frac{x-a}{x+a} \right + C$
	14. $\int \frac{dx}{x^2 + a^2} = \frac{1}{a} \operatorname{arctg} \frac{x}{a} + C$

Правила дифференцирования и интегрирования		Некоторые тригонометрические формулы:	
а)	Правила дифференцирования	1	$\sin^2 x + \cos^2 x = 1; 1 - \sin^2 x = \cos^2 x;$ $1 - \cos^2 x = \sin^2 x.$
1	Производная суммы: $(u \pm v)' = u' \pm v'$	2	$\sin^2 x = \frac{1 - \cos 2x}{2}, \cos^2 x = \frac{1 + \cos 2x}{2},$
2	Производная произведения: $(u \cdot v)' = u' \cdot v + u \cdot v'$	3	$\sin \alpha \cdot \sin \beta = \frac{1}{2}(\cos(\alpha - \beta) - \cos(\alpha + \beta))$
3	$(c \cdot u)' = c \cdot u'$	4	$\sin \alpha \cdot \cos \beta = \frac{1}{2}(\sin(\alpha - \beta) + \sin(\alpha + \beta))$
	Производная частного: $\left(\frac{u}{v}\right)' = \frac{u' \cdot v - u \cdot v'}{v^2}$	5	$\cos \alpha \cdot \cos \beta = \frac{1}{2}(\cos(\alpha + \beta) + \cos(\alpha - \beta))$
5	Сложная функция: $((f(\varphi(x))))' = f'_{\varphi} \cdot \varphi'(x)$	Формулы сокращённого умножения	
б)	Правила интегрирования	1	$(a \pm b)^2 = a^2 \pm 2ab + b^2$
1	$\int (f(x) \pm g(x))dx = \int f(x)dx \pm \int g(x)$		
2	$\int k \cdot f(x)dx = k \int f(x)dx$	2	$a^2 - b^2 = (a - b)(a + b)$
3	$\int f(ax + b)dx =$ $= \frac{1}{a} \int f(ax + b)d(ax + b)$		
4	$\int f(\varphi(x)) \cdot \varphi'(x)dx \pm \int f(\varphi(x)) d\varphi(x)$	3	$(a \pm b)^3 = a^3 \pm 3a^2b + 3ab^2 \pm b^3$ $a^3 \pm b^3 = (a \pm b)(a^2 \mp ab + b^2)$
5	Интегрирование по частям $\int u \cdot dv = u \cdot v - \int v \cdot du$		