Cheatography

Python - Seaborn Cheat Sheet

by DarioPittera (aggialavura) via cheatography.com/83764/cs/19851/

TO START

```
import seaborn as sns
# If working on a notebook
%matplotlib inline
```

DISTRIBUTION PLOTS

sns.distplot(df['col'])	distribution plot
- bin = x	number of bins
- kde = False	remove the line
sns.jointplot(x,y,df)	plot 2 variables
- kind = ''	kind of plot*
sns.pairplot(df)	plot all vars combin
- hue='categ var'	distinguish per var
- palette=''	set a color palette
sns.rugplot(df['col'])	idea of distribution
sns.kdeplot(df['col'])	kde plot

"kind=" E.g.: hex, reg, kde.

CATEGORICAL PLOTS

sns.barplot(x,y,df)	bar plot
- estimator="*	bar values
sns.countplot(x,df)	bars = count
sns.boxplot(x,y,,df)	box plot
- hue='categ var'	divide per var
- palette=''	set palette
- orient='h'	horiz. plot
sns.violinplot(x,y,df)	violin plot*
- hue='categ var'	divide per var
- palette=''	set palette
sns.stripplot(x,y,df)	bars = scatter
- jitter = True	add noise
- hue='categ var'	divide per var
- palette=''	set palette
- split = True	split by hue

CATEGORICAL PLOTS (cont)

sns.swarmplot(x,y,df)	swarm plot
- hue='categ var'	divide per var
- palette=''	set palette
- split = True	split by hue

** You can alo combine more plots by calling them one after each other.

sns.factorplot(x,y,df,kind)*
general categorical form of graph

TIP: when you call a plot function, press **"shift + tab"** to show the parameters needed.
estimator= can be, mean, std, or whatever function. It will display the bars or whatever you choose.
General form, kind=: e.g., point, bar, violin, etc.

ON CATEGORICAL PLOTS...

What is a violin plot?

> It has a similar role of a box and whisker plots. It shows the distribution of quantitative data across several levels of one (or more) categorical variables. The violin plot features a kernel density estimation of the underlying distribution.

What is a strip plot?

> It will draw a scatterplot where one variable is categorical. It is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution.

What is a swarm plot

> It is similar to a stripplot(), but the points are adjusted (only along the categorical axis) so that they don't overlap. This gives a better representation of the distribution of values, although it does not scale as well to large numbers of observations.

MATRIX PLOTS

sns.heatmap(df.corr())*	heat map plot
- annot = True	add actual values
- cmap=''	set a color palette
- linecolor=''	set borders
- linewidths=x	set border width
sns.clustermap(matrix)	hierarc. clustering
- cmap=''	set a color palette
- standard_scale = 1	normalise data

Heat map plot needs a correlation matrix, or more generally, a matrix. You can use the **pivot_table(index,columns,values)** function to convert a dataframe.

GRIDS

sns.pairplot(df)	plot all vars combination
- hue='categ var'	divide per var
- palette=''	set palette
g = **sns.PairGrid(df)**	set (empty) axis of pairplot
-g.map(plt.scatter)	populate axis with some plot
-g.map_diag(plt.hist)	set diag plots
-g.map_upper(plt.scatter)	set upper plots
-g.map_lower(sns.kdeplot)	set lower plots
g = sns.FacetGrid(df,c,r)	empty axis
-g = g.map(**plt**.hist, "c")	populate axis histograms
-..g.map(sns.distplot, "c")	populate axis with distplots

now some more complex stuff

GRIDS (cont)

- hue='categ var'	divide per var
-g = g.map(plt.scatter, "c", "c").add_legend()	

g = sns.JointGrid(x,y,df)
general form of jointplot()

g = g.plot(sns.regplot, sns.distplot)
join two plots

REGRESSION PLOTS

sns.lmplot(x,y,df)	creat reg plot
- hue='categ var'	divide per var
- palette=''	set palette
- markers='' *	set mark shape
- scatter_kws='dict' *	set marker size
sns.lmplot(x,y,df,**col**)	create a grid plot
sns.lmplot(x,y,df,**row,col**)	X*X grid
sns.lmplot(x,y,df,**row,col,hue**)	X*X*X grid
- aspect = x	choose ratio
- size = x	set size

markers='': e.g., o,v,etc.
scatter_kws=' e.g.: {'s':100}, it is a call to matplotlib. It will be hard to remember how to use these special cases, so no worries, you will have a look online.

STYLE and COLOR

sns.set_style('dar-kgrid')	apply darkgrid style
sns.set_style('ticks')	apply ticks style
sns.despine()	remove borders
sns.despine(lef-t=True)	remove left border
plt.figure(figsize=(x,x))	choose fig size
sns.set_context('talk') *	set context
sns.set_context (font_scale)	set font size

.set_context(''): e.g.: paper, poster, talk, notebook, etc.

By **DarioPittera** (aggialavura)

cheatography.com/aggialavura/
www.dariopittera.com

Not published yet.
Last updated 15th June, 2019.