

EDA

`import seaborn as sns` seaborn is used to set the plotting

`sns.set()` Set default Seaborn style

The "square root rule" is a commonly-used rule of thumb for choosing number of bins: choose the number of bins to be the square root of the number of samples.

`Bee swarm plot` Draw a categorical scatterplot with non-overlapping points.

`sns.swarmplot(x='colname1', y='colname2', data=df)` colname1 is categorical. y is for the numbers.

`ECDF` Empirical cumulative distribution function. It is one of the important plots for understanding the data.

`plt.plot(x, y, marker='.', linestyle='none')`

`plt.margins(0.0, 2)` Keeps data off plot edges

`np.arange(3,7)` `array([3, 4, 5, 6])`

`numpy.arange(start, stop, step, dtype=None)` Return evenly spaced values within a given interval.

numpy

`np.percentile(arrayname, [2.5, 25])` Compute the 2.5 and 25 percentiles of variable arrayname

`sns.boxplot(x=colname1, y=colname2, data=df)`

`np.var(arrayname)` compute the variance of numpy array arrayname

`np.std(arrayname)` compute the standard deviation of numpy array arrayname

numpy (cont)

`np.cov(x, y)` returns a 2D array where entries [0,1] and [1,0] are the covariances. Entry [0,0] is the variance of the data in x, and entry [1,1] is the variance of the data in y. This 2D output array is called the covariance matrix, since it organizes the self- and covariance.

`np.corrcoef()` Pearson correlation coefficient, also called the Pearson r, is often easier to interpret than the covariance. It is computed using the `np.corrcoef()` function. Like `np.cov()`, it takes two arrays as arguments and returns a 2D array. Entries [0,0] and [1,1] are necessarily equal to 1 (can you think about why?), and the value we are after is entry [0,1].

hypotheses

`permutation sampling` permutation sampling is a great way to simulate the hypothesis that two variables have identical probability distributions

`np.random.permutation(data)` Permute the concatenated array

`np.concatenate((data1, data2))` Concatenate the data sets

hypotheses (cont)

The p-value is generally a measure of: the probability of observing a test statistic equally or more extreme than the one you observed, assuming the hypothesis you are testing is true.

a permutation replicate is a single value of a statistic computed from a permutation sample.

probabilistic logic

Statistical inference involves taking your data to probabilistic conclusions about what you would expect if you took even more data, and you can make decisions based on these conclusions.

`np.random.random()` The function returns a random number between zero and one

`np.random.seed(42)` Seed the random number generator

`np.empty(10000)` Initialize an empty array, `random_numbers`, of 100,000 entries

`np.random.binomial(n=100, p=0.05, size=10000)` # Take 10,000 samples out of the binomial distribution: `n_defaults`

`np.random.poisson(10, size=10000)` Draw 10,000 samples out of Poisson distribution with a mean of 10

`np.random.normal(20, 1, size=100000)` Draw 100,000 samples from a Normal distribution that has a mean of 20 and a standard deviation of 1

`plt.hist(array, bins=100, normed=True, histtype='step')` `histtype='step'` smoothes histogram



By **elhamsh**

cheatography.com/elhamsh/

Not published yet.

Last updated 12th January, 2018.

Page 1 of 2.

Sponsored by **Readability-Score.com**

Measure your website readability!

<https://readability-score.com>

probabilistic logic (cont)

`plt.ylim(a, b)` limit the y axes between a and b

`np.random.exponential(mean, size=size)`

`slope,` found the slope and intercept of the points

`intercept =` (x,y). degree determines the degree of

`np.polyfit(x,` polynomial

`y, degree)`

`np.linspace(a, b, c)` get c points in the range between a and b

`np.empty_like(variable)` This function returns a new array with the same shape and type as a given array "variable"

Bootstrapping The use of resampled data to perform statistical inference

If we have a data set with nn repeated measurements, a bootstrap sample is an array of length nn that was drawn from the original data with replacement

`np.random.choice(array, size=n)` Generate bootstrap sample from array with size n

Confidence interval of a statistic If we repeated measurements over and over again, $p\%$ of the observed values would lie within the $p\%$ confidence interval.

A confidence interval gives bounds on the range of parameter values you might expect to get if we repeated our measurements. For named distributions, you can compute them analytically or look them up, but one of the many beautiful properties of the bootstrap method is that you can just take percentiles of your bootstrap replicates to get your confidence interval. Conveniently, you can use the `np.percentile()` function.

pairs bootstrap involves resampling pairs of data.

