This repository Search Pull requests Issues Gist

chelmyers / data-analysis-python

Watch 3 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs

Data analysis with Python and Jupyter

7 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

chelmyers add jump links		Latest commit 759d883 on Oct 24, 2016
wk-06-examples	reorganize examples	4 months ago
README.md	add jump links	4 months ago

README.md

Data Analysis with Python and Jupyter Cheat Sheet

- Descriptive Statistics
- Statistical Inference

Descriptive Statistics

Measures of Center

Idea of typical value.

Mean: The numerical average.

```
dataset.mean()
```

Median: The 50th percentile (50% of response are higher and lower).

```
dataset.median()
```

Mode: Returns a table. Can return multiple value.

```
dataset.mode()
```

Measures of Spread

How data varies from typical value.

Quantile: Figure out the values of the 0th, 25th, 50th, 75th, and top percentile using the quantile function.

```
five_num = [dataset["subset"].quantile(0),
            dataset["subset"].quantile(0.25),
            dataset["subset"].quantile(0.50),
            dataset["subset"].quantile(0.75),
            dataset["subset"].quantile(1)]

five_num # Will output these ^ values
```

Describe: Also return quantiles, but also returns mean, standard of deviation, min and max values.

```
dataset["subset"].describe();
```

IQR: Interquartile Range. Distance between the 3rd and the 1st quartile. Describes the data.

Boxplot: A visual representation of the five number summary and IQR.

```
dataset.boxplot(column="subset",
                return_type='axes',
                fiqsize=(8,8))

plt.text(x=0.74, y=22.25, s="3rd Quartile")
plt.text(x=0.8, y=18.75, s="Median")
plt.text(x=0.75, y=15.5, s="1st Quartile")
plt.text(x=0.9, y=10, s="Min")
plt.text(x=0.9, y=33.5, s="Max")
plt.text(x=0.7, y=19.5, s="IQR", rotation=90, size=25)
```

Variance: The average of the squared deviations (differences) from the mean. A numerical value used to indicate how widely individuals in a group vary.

```
dataset["subset"].var()
```

Standard Deviation: Extent of deviation from mean.

```
dataset["subset"].std()
```

Median Absolute Deviation (MAD): Alternative measure of spread based on median if the mean cannot be trusted (or is not a true representation).

```
abs_median_devs = abs(dataset["subset"] - dataset["subset"].median())
abs_median_devs.median() * 1.4826 #Multiplied by scale factor for normal distribution
```

Skewness and Kurtosis

Skew: Measure of skew or asymmetry of dataset. More extreme number measure is more skewed.

```
dataset["subset"].skew()
```

Kurtosis: Measure of peakedness of a distribution. Higher number is more peaked. Negative number is more uniform.

```
dataset["subset"].kurt()
```

Statistical Inference

Sample to guess the whole. The process of analyzing sample data to gain insight into the population from which the data was collected and to investigate differences between data samples.

Point Estimates

Estimate of the population based on sample data.

Sample Mean: The average of a sample. Same function for `.mean()` .

Sampling Distribution and The Central Limit Theorem

Histogram: Show data/information in "bins".

```
pd.DataFrame(dataset).hist(bins=58,
                          range=(17.5,75.5),
                          figsize=(9,9))
print( stats.skew(dataset) )
```

Central Limit Theorem: The distribution of many sample means, known as a sampling distribution, will be normally distributed. For example, get 200 samples of your sample data and make a point estimate for each. These point estimates will have a normal distribution.

```
np.random.seed(10)

point_estimates = []         # Make empty list to hold point estimates

for x in range(200):         # Generate 200 samples, each with 500 sampled values
    sample = np.random.choice(a= population_ages, size=500)
    point_estimates.append( sample.mean() ) # keep the sample mean in the list

pd.DataFrame(point_estimates).plot(kind="density",  # Plot sample mean density
                                   figsize=(9,9),
                                   xlim=(41,45))
```

Confidence Intervals

A range of values above and below a point estimate that captures the true population parameter at some predetermined confidence level. Calculate a confidence interval by taking a point estimate and then adding and subtracting a margin of error to create a range.

```
confidence_interval = (sample_mean – margin_of_error,
                       sample_mean + margin_of_error)
```

OR

```
stats.t.interval(alpha=0.95,              # Confidence level
                 df=24,                   # Degrees of freedom
                 loc=sample_mean,         # Sample mean
                 scale=sigma)             # Standard deviation estimate
```

Margin of Error: The amount allowed for miscalculation.

```
margin_of_error = z_critical * (stdev/math.sqrt(sample_size))
```

Z-critical: The number of standard deviations you'd have to go from the mean of the normal distribution to capture the proportion of the data associated with the desired confidence level. If four cover that range, the Z-critical would be 2.

```
z_critical = stats.norm.ppf(q = 0.975) #q = confidence interval of 95%
```

T-critical: Use when you do not have the standard deviation but only a sample standard deviation.

```
t_critical = stats.t.ppf(q = 0.975, df=24) #df is degrees of freedom (sample size minus 1)
                                           # q = confidence interval of 95 #
```

Degrees of Freedom: Sample size minus 1.

T-Distribution: A distribution that closely resembles the normal distribution but that gets wider and wider as the sample size falls.

Statistical Hypothesis Testing: The T-Test

What is the likelihood as a percentage that something interesting is going on (when comparing data). Checks the null hypothesis.

Null Hypothesis: Assumes nothing interesting is going on between whatever variables you are testing.

One-Sample T-Test

A one-sample t-test checks whether a sample mean differs from the population mean.

```
stats.ttest_1samp(a=dataset_sample,               # Sample data
                  popmean=dataset_whole.mean())   # All data mean
```

Two-Sample T-Test

Investigates whether the means of two independent data samples differ from one another.

```
stats.ttest_ind(a=dataset_sample,
                b=dataset_whole,
                equal_var=False)    # Assume samples have equal variance?
```

Paired T-Test

A paired t-test lets you check whether the means of samples from the same group differ.

```
before= stats.norm.rvs(scale=30, loc=250, size=100) #Generated dummy data

after = before + stats.norm.rvs(scale=5, loc=-1.25, size=100)  #Generate dummy data

stats.ttest_rel(a=before,
                b=after)
```

Type I & II Errors

Type I: False-positive. A situation where you reject the null hypothesis when it is actually true.

Type II: False-negative. A situation where you fail to reject the null hypothesis when it is actually false.