Statistical measurements using numpy, pandas, SciPy

+ Code + Text

Here are some of the most commonly used statistical formulas in NumPy, Pandas, and other libraries:

NumPy:

- **Descriptive statistics**:
- Mean: np.mean(data)
- Median: np.median(data)
- Mode: stats.mode(data)
- Variance: np.var(data)
- Standard deviation: np.std(data)
- Minimum value: np.min(data)
- Maximum value: np.max(data)
- Range: np.max(data) - np.min(data)
- **Correlation:**
- Pearson correlation coefficient: np.corrcoef(x, y)
- Probability distributions:
- Normal distribution: np.random.normal(loc, scale, size)
- Binomial distribution: np.random.binomial(n, p, size)
- Poisson distribution: np.random.poisson(lam, size)
- Uniform distribution: np.random.uniform(low, high, size)
- **Skewness with numpy:**
- numpy.skew(data) - Computes the skewness of a data set.
- **Kurtosis with numpy:**
- numpy.kurtosis(data) - Computes the kurtosis of a data set.
- **Standard error of the mean with numpy:**
- numpy.std(data, ddof=1) / numpy.sqrt(len(data)) - Computes the standard error of the mean of a data set.
- **Pandas:**
- **Descriptive statistics:**

- Mean: df.mean()
- Median: df.median()
- Mode: df.mode()
- Variance: df.var()
- Standard deviation: df.std()
- Minimum value: df.min()
- Maximum value: df.max()
- Range: df.max() - df.min()
- **Correlation:**
- Pearson correlation coefficient: df.corr()
- **Probability distributions:**
- Normal distribution: pd.DataFrame(np.random.normal(loc, scale, size))
- Binomial distribution: pd.DataFrame(np.random.binomial(n, p, size))
- Poisson distribution: pd.DataFrame(np.random.poisson(lam, size))
- Uniform distribution: pd.DataFrame(np.random.uniform(low, high, size))
- **Skewness with pandas**:
- df.skew() - Computes the skewness of a data set.
- **Kurtosis with pandas:**
- df.kurtosis() - Computes the kurtosis of a data set.
- **Standard error of the mean with pandas:**
- df.sem(ddof=1) - Computes the standard error of the mean of a data set.
- Note that the ddof parameter is used to specify the degrees of freedom for the calculation of the standard deviation. In general, if you are working with a sample (as opposed to a population), you should use ddof=1 to correct for bias in the estimation of the population standard deviation.
- **SciPy:**
- **Hypothesis testing:**
- One-sample t-test: scipy.stats.ttest_1samp(data, popmean)
- Two-sample t-test for independent samples: scipy.stats.ttest_ind(data1, data2)
- Two-sample t-test for dependent samples: scipy.stats.ttest_rel(data1, data2)
- **Probability distributions:**
- Normal distribution: scipy.stats.norm.pdf(x, loc, scale)
- Binomial distribution: scipy.stats.binom.pmf(k, n, p)

- Poisson distribution: scipy.stats.poisson.pmf(k, mu)
- Uniform distribution: scipy.stats.uniform.pdf(x, loc, scale)
- **Other statistical measures:**
- Skewness: scipy.stats.skew(data)
- Kurtosis: scipy.stats.kurtosis(data)
- Standard error of the mean: scipy.stats.sem(data)
- I hope you find these formulas helpful!

✓ 0s completed at 06:16