Introduction to Python

Tan Sein Jone

University of British Columbia

July 16, 2024

Table of contents

Table of Contents

Statasmodel

- Statsmodels is a library for statistical modeling
- It provides support for linear regression, logistic regression, and time series analysis
- It allows you to fit models, make predictions, and evaluate results

Linear Regression

- Linear regression is a statistical method for modeling the relationship between two variables
- It is used to predict the value of one variable based on the value of another variable
- It is used to estimate the coefficients of the regression equation

Linear Regression

```
import statsmodels.api as sm
import numpy as np

data = {
    "age": np.random.normal(50, 10, 100),
    "income": np.random.normal(50000, 10000, 100)
}
data = pd.DataFrame(data)
X = data["age"]
y = data["income"]
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
```

Hypothesis Testing

- Hypothesis testing is a statistical method for testing the validity of a hypothesis
- It is used to determine whether a hypothesis is true or false
- It is used to make inferences about a population based on a sample

Hypothesis Testing

```
import statsmodels.api as sm

data = {
    "age": np.random.normal(50, 10, 100),
    "income": np.random.normal(50000, 10000, 100)
}

data = pd.DataFrame(data)
X = data["age"]
y = data["income"]
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
print(model.pvalues)
```

Table of Contents

Matplotlib

- Matplotlib is a library for creating static, animated, and interactive visualizations
- It provides support for line plots, bar plots, scatter plots, and histograms
- It allows you to customize the appearance of plots

Line Plot

- A line plot is a type of plot that displays data as a series of points connected by lines
- It is used to show trends, patterns, and relationships in data
- It is used to visualize the relationship between two variables

Line Plot

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

x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
plt.show()
```

Seaborn

- Seaborn is a library for creating statistical data visualizations
- It provides support for line plots, bar plots, scatter plots, and histograms
- It allows you to customize the appearance of plots

Bar Plot

- A bar plot is a type of plot that displays data as a series of bars
- It is used to compare the values of different categories
- It is used to visualize the distribution of a categorical variable

Bar Plot

```
import seaborn as sns

data = {
    "age": np.random.normal(50, 10, 100),
    "income": np.random.normal(50000, 10000, 100)
}
data = pd.DataFrame(data)
sns.barplot(x="age", y="income", data=data)
plt.savefig("barplot.png")
plt.show()
```

Table of Contents

Pandas Best Practices

- Use the `read_csv` function to read CSV files
- Use the `head` function to display the first few rows of a DataFrame
- Use the `info` function to display information about a DataFrame
- Use the `describe` function to display summary statistics of a DataFrame

Pandas Best Practices

- Always have an original dataframe for your import data
- Use the `copy` function to create a copy of a DataFrame
- Write to csv only when necessary
- You may run into memory issues if you have a large dataset

Pandas Best Practices

- It's always a good idea to save memory by overriding unused variables
- Restart the kernel if you run into memory issues
- Use the `apply` function when you need to apply a function to a DataFrame

Pandas Best Practices

```
import pandas as pd

df = pd.read_csv("data/province_weather.csv")
df.head()
df.info()

weather = df.copy()
weather.describe()
```

Pandas Best Practices

```
weather['fahrenheit'] = weather['temperature'].apply(lambda x: x
weather.to_csv("data/province_weather_fahrenheit.csv", index=Fal
```