



Introduction to Data Science in Python

文件导入

```
# Import pandas
import pandas as pd
# Load the 'ransom.csv' into a DataFrame
r = pd.read_csv('ransom.csv')
# Display DataFrame
print(r)
```

筛选

```
# Select the location column in credit_records
location = credit_records["location"]
```

筛选 bloom

```
# Select the dogs where Age is greater than 2
greater_than_2 = mpr[mpr.Age > 2]
print(greater_than_2)
# Select the dogs whose Status is equal to Still Missing
still_missing = mpr[mpr.Status == 'Still Missing']
print(still_missing)
# Select all dogs whose Dog Breed is not equal to Poodle
not_poodle = mpr[mpr['Dog Breed'] != 'Poodle']
print(not_poodle)
```

画折线图

```
# From matplotlib, import pyplot under the alias plt
from matplotlib import pyplot as plt
# Plot Officer Deshaun's hours_worked vs. day_of_week
plt.plot(deshaun.day_of_week, deshaun.hours_worked, label='Deshaun', color="DarkCyan", linestyle=':', marker='s')
```

- You can change `linestyle` to dotted (':'), dashed('-.'), or no line ('').

- You can change the `marker` to circle (`'o'`), diamond(`'d'`), or square (`'s'`).

```
# Add a title
plt.title("title")
# Add y-axis label
plt.xlabel("Letter"), plt.ylabel("hours_worked")
# Legend
plt.legend()
# Display Deshaun's plot
plt.show()
```

添加文本

```
# Add annotation "Missing June data" at (2.5, 80)
plt.text(2.5, 80, "Missing June data")
```

```
# Explore the data
print(cellphone.head())
```

散点图

```
# Change the transparency to 0.1
plt.scatter(cellphone.x, cellphone.y,
            color='red',
            marker='s',
            alpha=0.1)
```

- `alpha` 是透明度

柱形图

```
# Create a bar plot from the DataFrame hours
plt.bar(hours.officer, hours.avg_hours_worked,
# Add error bars
yerr=hours.std_hours_worked)
```

长条图比较（两个对象）

```
# Plot the number of hours spent on desk work
plt.bar(hours.officer, hours.desk_work, label='Desk Work')
# Plot the hours spent on field work on top of desk work
plt.bar(hours.officer, hours.field_work,
        bottom=hours.desk_work, label='Field Work')
# Add a legend
plt.legend()
# Display the plot
plt.show()
```

柱形图

```
# Create a histogram
plt.hist(gravel.radius,
bins=40,
range=(2, 8),
density=True)
# Label plot
plt.xlabel('Gravel Radius (mm)')
plt.ylabel('Frequency')
plt.title('Sample from Shoeprint')
# Display histogram
plt.show()
```

- `range` sets the minimum and maximum datapoints that we will include in our histogram.
- `bins` sets the number of points in our histogram
- `density` Normalize your histogram so that the sum of the bins adds to 1.