

# Python

## Introduction

Jeff Abrahamson

8 mars    15 mars    29 mars 2019

# scikit-learn

# Datasets (1)

```
import sklearn as sk
from sklearn import datasets

iris = datasets.load_iris() digits =
datasets.load_digits()
```

Look at

- `iris.data.size`
- `iris.data.shape`

## Datasets (2)

`load_diabetes`

`load_boston`

`load_linnerud`

`load_wine`

`load_breast_cancer`

## Datasets (3)

- `data : n_samples, n_features`
- `target : response variables`
- `target_names`
- `DESCR : try print()`

# Classification

```
from sklearn import svm  
clf = svm.SVC(gamma=0.001, C=100.)  
clf.fit(digits.data[:-1], digits.target[:-1])  
clf.predict(digits.data[-1:])
```

# matplotlib

```
import numpy as np
%matplotlib inline # Optional: Jupyter notebook
from matplotlib import pyplot as plt

x = np.arange(1, 11)
y = 2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x, y)
plt.show()
```

# matplotlib

```
import numpy as np
%matplotlib inline # Optional: Jupyter notebook
from matplotlib import pyplot as plt

x = np.arange(1, 11)
y = 2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x, y, 'dr')
plt.show()
```



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

x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
plt.title("sine wave form")
plt.plot(x, y, 'g')
plt.show()
```

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

x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

# Set up a subplot grid that has height 2 and width
1,
# and set the first such subplot as active.
plt.subplot(2, 1, 1)
# Make the first plot
plt.plot(x, y_sin)
plt.title('Sine')
# Set the second subplot as active, and make the
second plot.
plt.subplot(2, 1, 2)
plt.plot(x, y_cos)
plt.title('Cosine')

plt.show()
```

```
x = [5,8,10]
y = [12,16,6]

x2 = [6,9,11]
y2 = [6,15,7]
plt.bar(x, y, align = 'center')
plt.bar(x2, y2, color = 'g', align = 'center')
plt.title('Bar graph')
plt.ylabel('Y axis')
plt.xlabel('X axis')

plt.show()
```

[https://matplotlib.org/tutorials/  
introductory/sample\\_plots.html](https://matplotlib.org/tutorials/introductory/sample_plots.html)

<https://matplotlib.org/gallery/>

# seaborn

```
import seaborn as sns

tips = sns.load_dataset("tips")
sns.relplot(x="total_bill", y="tip",
            data=tips)

plt.show()
```

```
sns.relplot(x="total_bill", y="tip",  
            hue="smoker", data=tips)
```

```
sns.relplot(x="total_bill", y="tip",  
            hue="smoker", style="smoker", data=tips);
```



```
sns.relplot(x="total_bill", y="tip",  
            size="size", sizes=(15, 200), data=tips);
```

# seaborn and random walks

```
df = pd.DataFrame(dict(time=np.arange(500),  
                        value=np.random.randn(500).cumsum()))  
g = sns.relplot(x="time", y="value",  
kind="line", data=df)  
g.fig.autofmt_xdate()
```

# More random walks

```
df = pd.DataFrame(np.random.randn(500, 2) .  
    cumsum(axis=0), columns=["x", "y"])  
sns.relplot(x="x", y="y", sort=False,  
    kind="line", data=df);
```

# More seaborn examples

<https://seaborn.pydata.org/tutorial/relational.html>

# Seaborn and categorical data

[https:](https://seaborn.pydata.org/tutorial/categorical.html)

[//seaborn.pydata.org/tutorial/categorical.html](https://seaborn.pydata.org/tutorial/categorical.html)

# Seaborn and distributions

<https://seaborn.pydata.org/tutorial/distributions.html>

# Pour la prochaine fois

**la même chose mais plus sophistiqué**

J2/travail-pour-la-prochaine-fois.txt

N'oubliez pas le tutoriel d'Addfor.

# Questions?