Machine Learning

TP 1 : données et statistiques

python

Useful tools

- virtualenv
- pip
- ipython
- ipython notebook
- `conda.pydata.org`

python

Notes

- `pip install -r requirements.txt`
- ipython offers tab completion (vs python)
- ipython notebook opens in a browser, caches cell output but not cell state

pandas

```
import pandas as pd
import numpy as np
import scipy
import matplotlib.pyplot as plt
```

pandas

Dataframe has many constructors. For example,

```
In [5]: pd.DataFrame({ 'A' : 1.,
               'B' : pd.Timestamp('20161209'),
               'C' : pd.Series(1,index=list(range(4)),dtype='float32'),
               'D' : np.array([3] * 4, dtype='int32'),
               'E' : pd.Categorical(["test","train","test","train"]),
               'F' : 'hello' })
Out[5]:
   A          B  C  D      E      F
0  1 2016-12-09  1  3   test  hello
1  1 2016-12-09  1  3  train  hello
2  1 2016-12-09  1  3   test  hello
3  1 2016-12-09  1  3  train  hello
```

pandas

Viewing data

```
In [16]: dates = pd.date_range('20161209', periods=4, freq='1w')

In [17]: df = pd.DataFrame(np.random.randn(4,5), index=dates,
                           columns=list('ABCDE'))

In [18]: df.head()
Out[18]:
                   A         B         C         D         E
2016-12-11 -1.303610 -1.235823  0.621914  0.379340 -0.326934
2016-12-18 -1.218197 -1.113826  0.546314 -0.255001 -0.135573
2016-12-25 -0.124625  0.337268 -0.406295  0.587049 -0.904906
2017-01-01 -0.283182 -0.866213  0.051509  0.693037 -0.661055
```

pandas

Basic data exploration

```
In [19]: df.describe()
Out[19]:
              A         B         C         D         E
count  4.000000  4.000000  4.000000  4.000000  4.000000
mean  -0.732403 -0.719648  0.203361  0.351106 -0.507117
std    0.614672  0.721194  0.478728  0.424558  0.342755
min   -1.303610 -1.235823 -0.406295 -0.255001 -0.904906
25%   -1.239550 -1.144325 -0.062942  0.220755 -0.722018
50%   -0.750689 -0.990019  0.298912  0.483195 -0.493995
75%   -0.243543 -0.565343  0.565214  0.613546 -0.279094
max   -0.124625  0.337268  0.621914  0.693037 -0.135573
```

pandas

Select a column (series)

```
In [20]: df.loc[dates[1]]
Out[20]:
A    -1.218197
B    -1.113826
C     0.546314
D    -0.255001
E    -0.135573
Name: 2016-12-18 00:00:00, dtype: float64
```

pandas

Select a range

```
In [21]: df.loc[:, ['A', 'C']]
Out[21]:
                   A         C
2016-12-11 -1.303610  0.621914
2016-12-18 -1.218197  0.546314
2016-12-25 -0.124625 -0.406295
2017-01-01 -0.283182  0.051509
```

pandas

Boolean selection criteria

```
In [23]: df[df.D > 0]
Out[23]:
                   A         B         C         D         E
2016-12-11 -1.303610 -1.235823  0.621914  0.379340 -0.326934
2016-12-25 -0.124625  0.337268 -0.406295  0.587049 -0.904906
2017-01-01 -0.283182 -0.866213  0.051509  0.693037 -0.661055
```

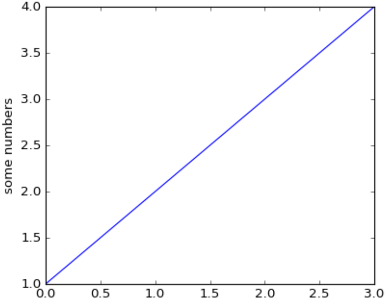
pandas

Recommended http://www.gregreda.com/2013/10/26/intro-to-pandas-data-structures/

Plotting

Draw a line

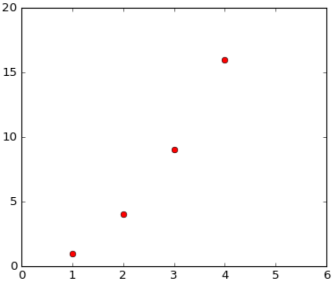
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```



Plotting

Draw a line

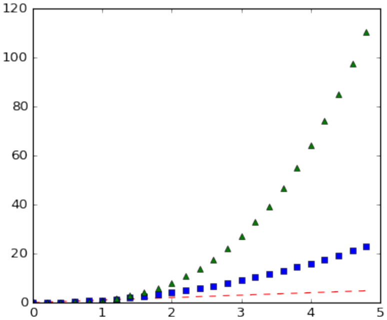
```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.ylabel('some numbers')
plt.show()
```



Plotting

Draw a line

```python
import numpy as np
import matplotlib.pyplot as plt
t = np.arange(0., 5., 0.2)
# r-- red dashes
# bs  blue squares
# g^  green triangles
plt.plot(t, t,
          'r--', t,
          t**2, 'bs',
          t, t**3, 'g^')
plt.show()
```

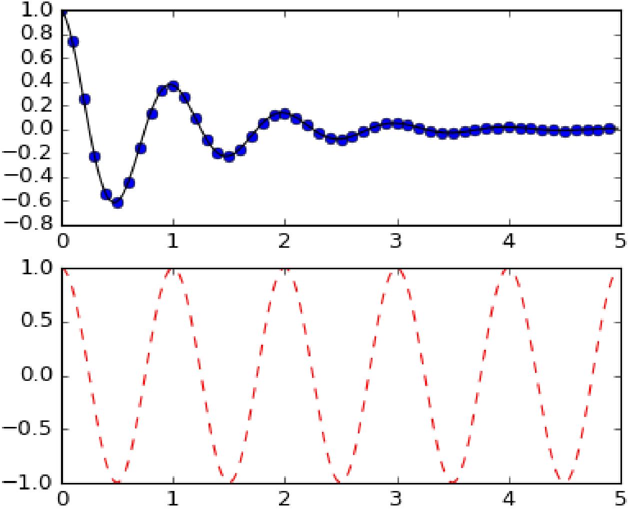


Plotting

Draw two curves

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

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure(1)
plt.subplot(211)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
```

Plotting

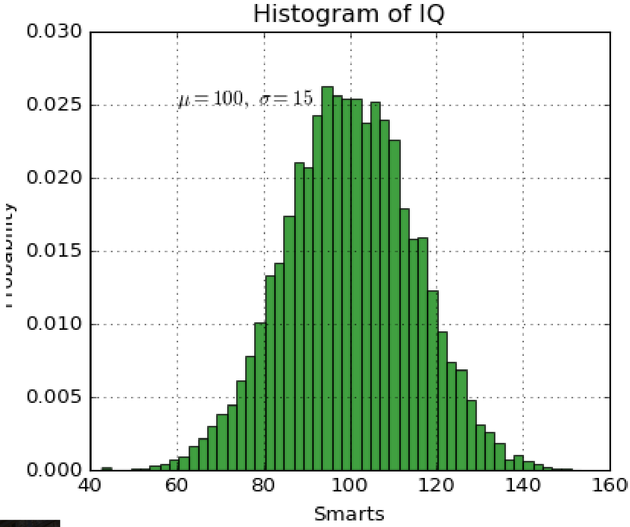


Plotting

Histogram

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

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

n, bins, patches = plt.hist(x, 50, normed=1, facecolor='g', alpha=0.75)

plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

Plotting

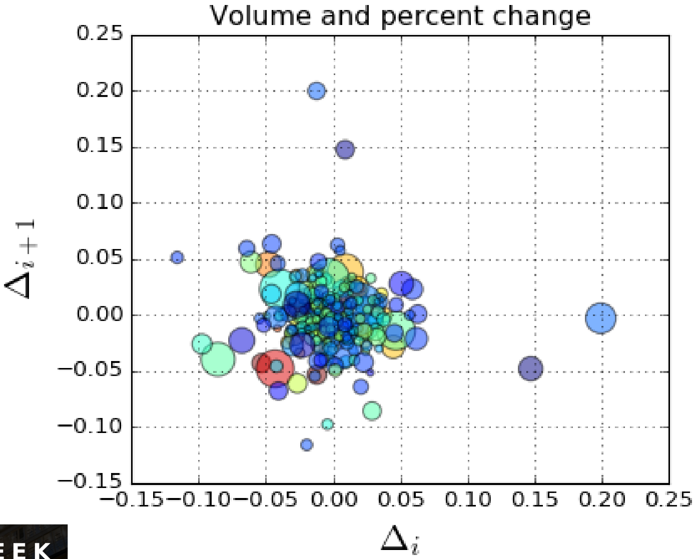

Histogram of IQ
$\mu=100,\ \sigma=15$
Probability
0.030
0.025
0.020
0.015
0.010
0.005
0.000
40
60
80
100
120
140
160
Smarts

Plotting

Scatter plot

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

fig, ax = plt.subplots()
ax.scatter(delta1[:-1], delta1[1:], c=close, s=volume, alpha=0.5)

ax.set_xlabel(r'$\Delta_i$', fontsize=20)
ax.set_ylabel(r'$\Delta_{i+1}$', fontsize=20)
ax.set_title('Volume and percent change')

ax.grid(True)
fig.tight_layout()

plt.show()
```

Plotting


Volume and percent change
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Δ_{i+1}
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
Δ_i

Plotting

http://matplotlib.org/users/pyplot_tutorial.html
http://matplotlib.org/users/beginner.html