A very brief Introduction to Python for Data Science

Part 2

Introduction to Data Science

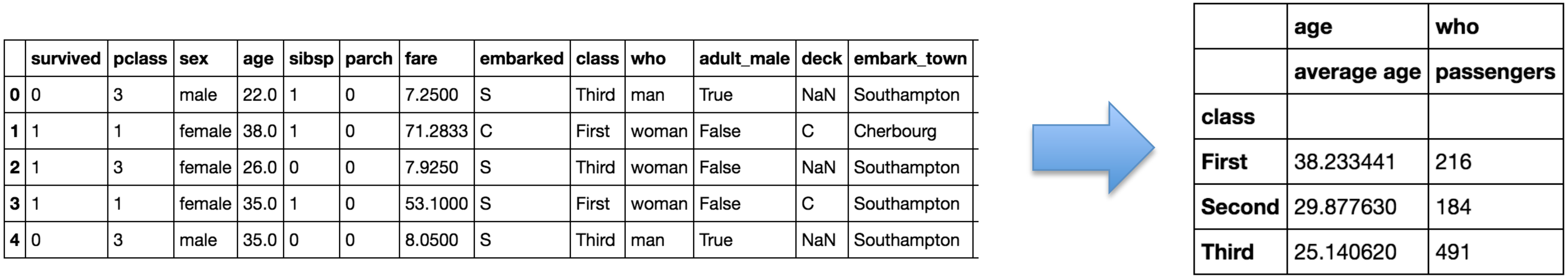
Slides by Mark Carman

Advanced Aggregation

In last week's tutorial you saw basic table manipulation and groupby commands

- This week we'll see how to run multiple aggregation operators at once

```
fun = {'who':{'passengers':'count'},'age':{'average age':'mean'}}
groupbyClass = titanic.groupby('class').agg(fun)
```



	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton

	age	**who**
	average age	**passengers**
class		
First	38.233441	216
Second	29.877630	184
Third	25.140620	491

- And how to write custom aggregators using anonymous functions:

```
fun = {'age':{'unique age count':'nunique','over 50s count':lambda x: sum(e>50 for e in x)}}
```

Slides by Mark Carman

Plotting data

- We can use the matplotlib library to plot data in Python

```
import matplotlib.pylab as plt
```

- Define a table with the data to plot:

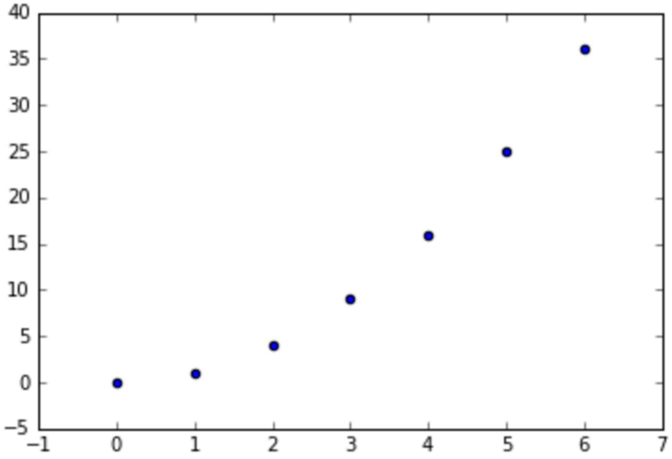
```
> df = pd.DataFrame({
'X' : [0,1,2,3,4,5,6],
'Y' : [0,1,4,9,16,25,36]})
```

- Create scatter plot

```
> plt.scatter(df['X'], df['Y'])
```

- And show it:

```
plt.show()
```



More plots

There are many other types of plots for visualising data in Python. In the tutorial we'll investigate:

- Basic plots

```
plt.plot(df.col_name)
```

- Histograms

```
df.col_name.hist(bins=200)
```

- Boxplots

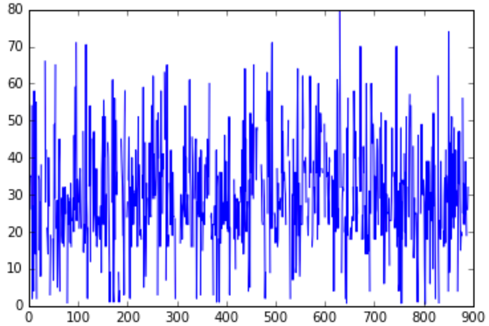
```
df.boxplot(column=‘col_name’)
```

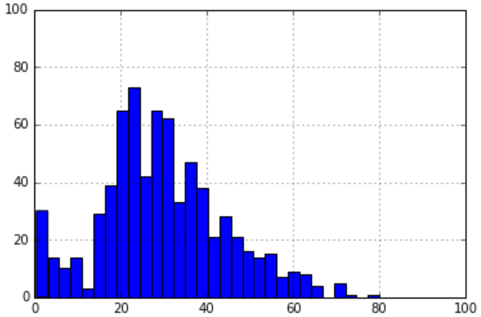
- Bar Charts

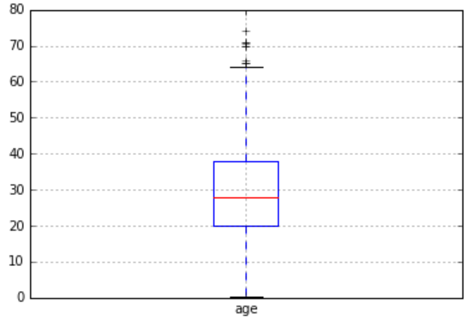
```
plt.bar((1,2,3),df['col_name’])
```

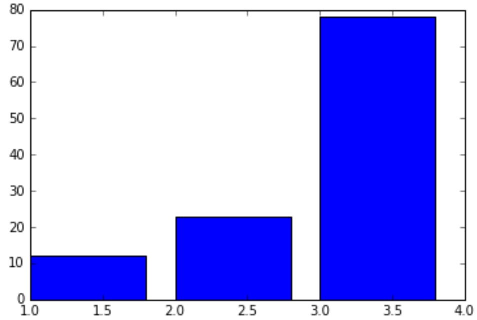
- Motion charts:

```
See code from Week 2 tutorial ...
```







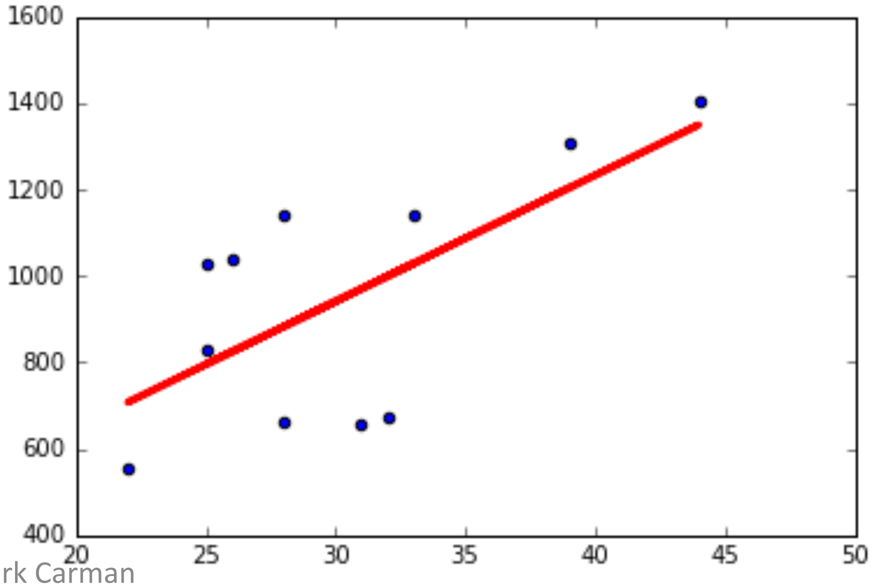


Slides by Mark Carman

Linear Regression

In the tutorial we'll show you how to compute a linear regression through data:

```
from scipy.stats import linregress
slope, intercept, r_value, p_value, std_err =
linregress(df['Age'],df['Runs'])
line = [slope*xi + intercept for xi in df['Age']]
```



End of Introduction

- We'll be playing around with Python in this week's Tutorial

- There are MANY excellent Python resources online if you'd like to learn more
  - for example: lynda.com