COMP47670

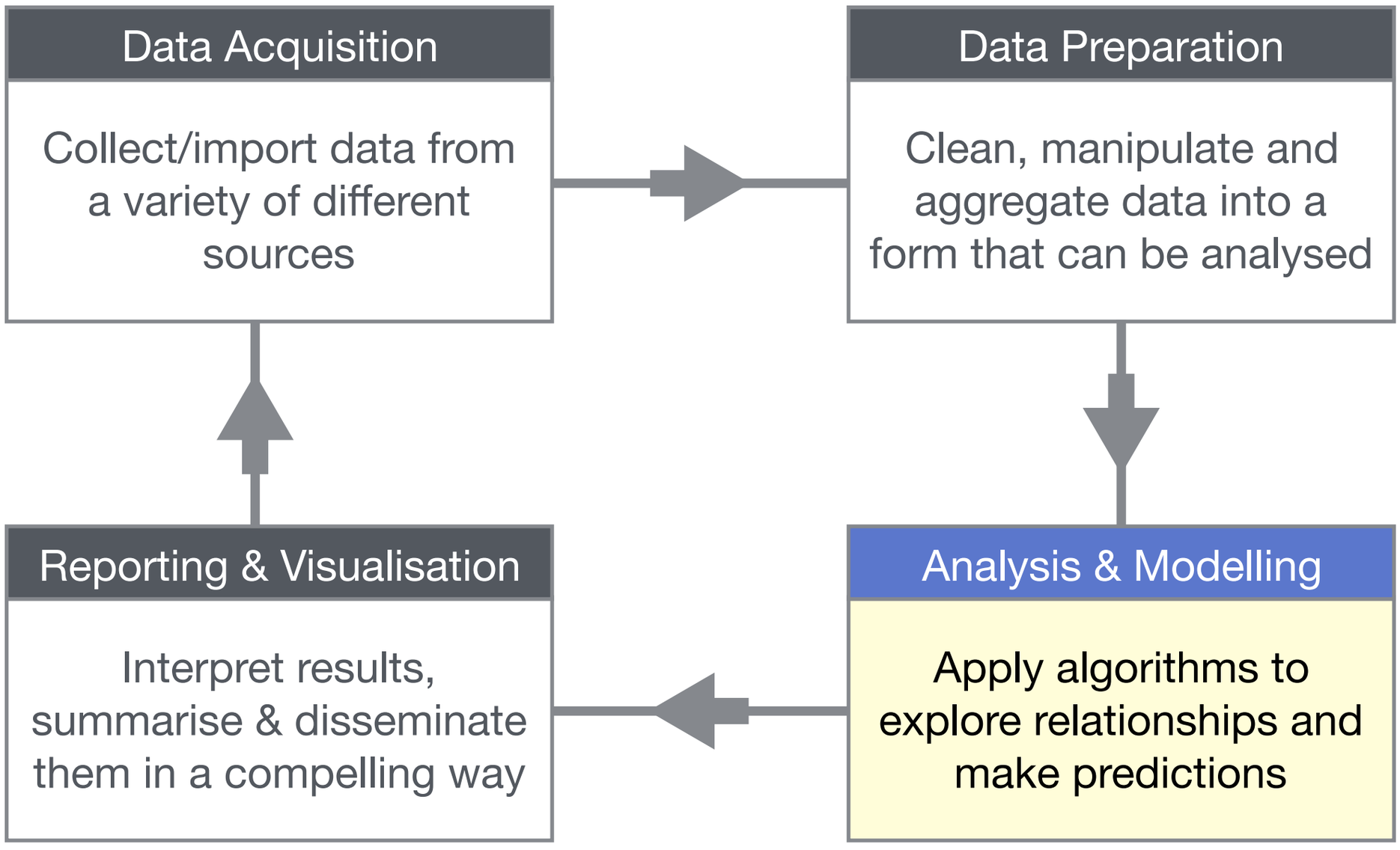
Modelling and Prediction

Slides by Derek Greene

UCD School of Computer Science

Reminder: Data Science Pipeline

- Recall the stages of the basic data science pipeline…
- Most complex component relates to data mining and modelling.

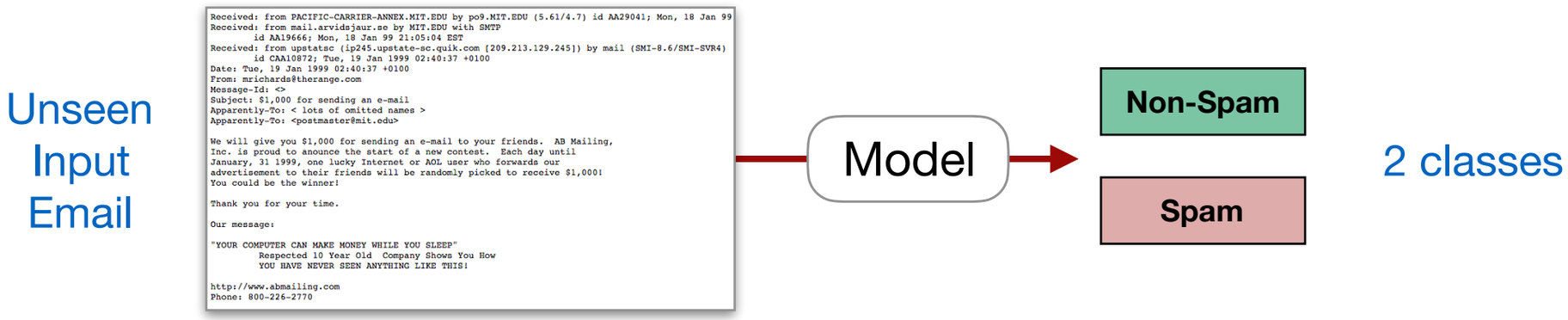


Modelling and Prediction Tasks

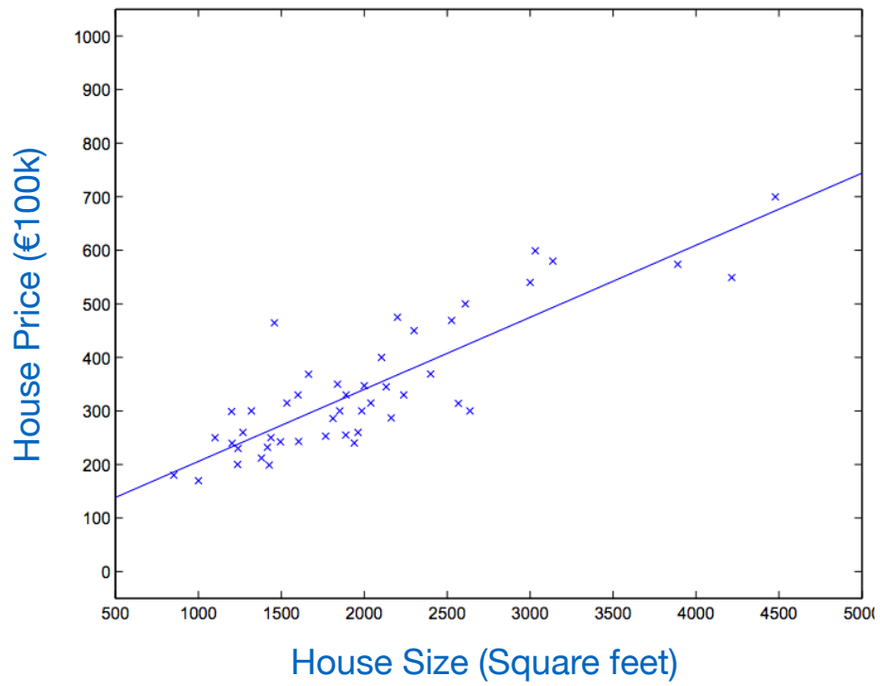
- Predictive modelling uses statistics to predict outcomes, based on historic data. Also referred to as supervised machine learning.
- **Examples:**
  - Spam filtering: predict if a new email is spam or non-spam, based on annotated examples of past spam / non-spam.
  - Car insurance: assign risk of accidents to policy holders and potential customers.
  - Healthcare: predict disease which a patient has, based on their symptoms.
  - Algorithmic trading: predictive models can be built for different assets like stocks, futures, currencies, etc, based on historic data and company information.

Supervised Learning

- Classification:
  Learn from a labelled training set to make a prediction to assign a new "unseen" example to one of a fixed number of classes.



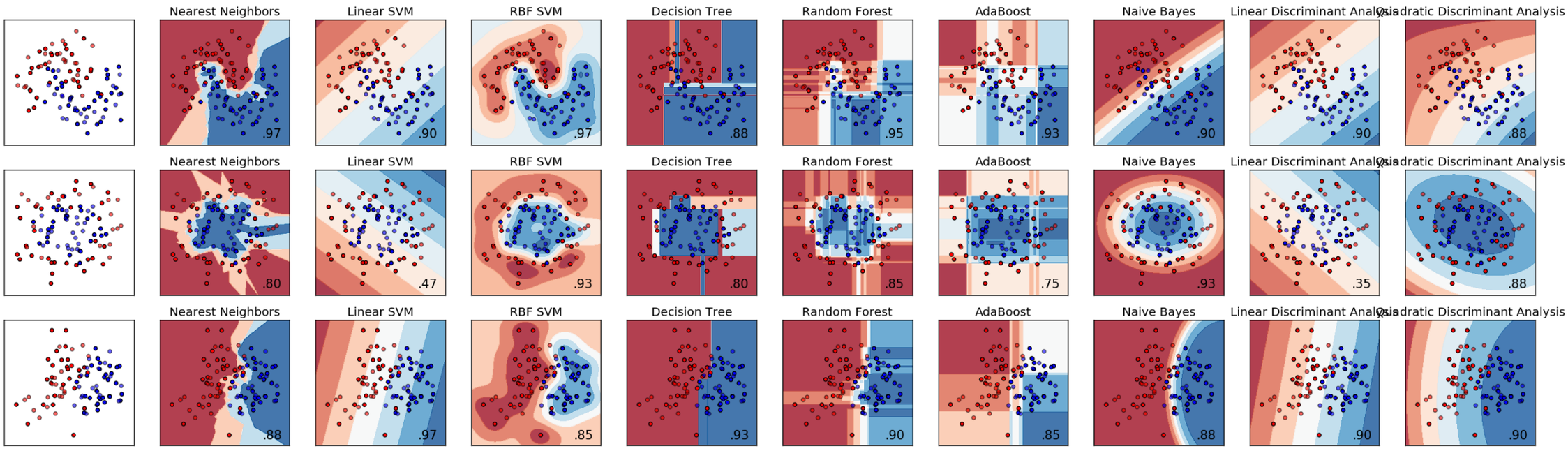
- Regression:
  Learn from a labelled training set to decide the value of a continuous output variable (i.e. the output is a number).



Scikit-Learn Package

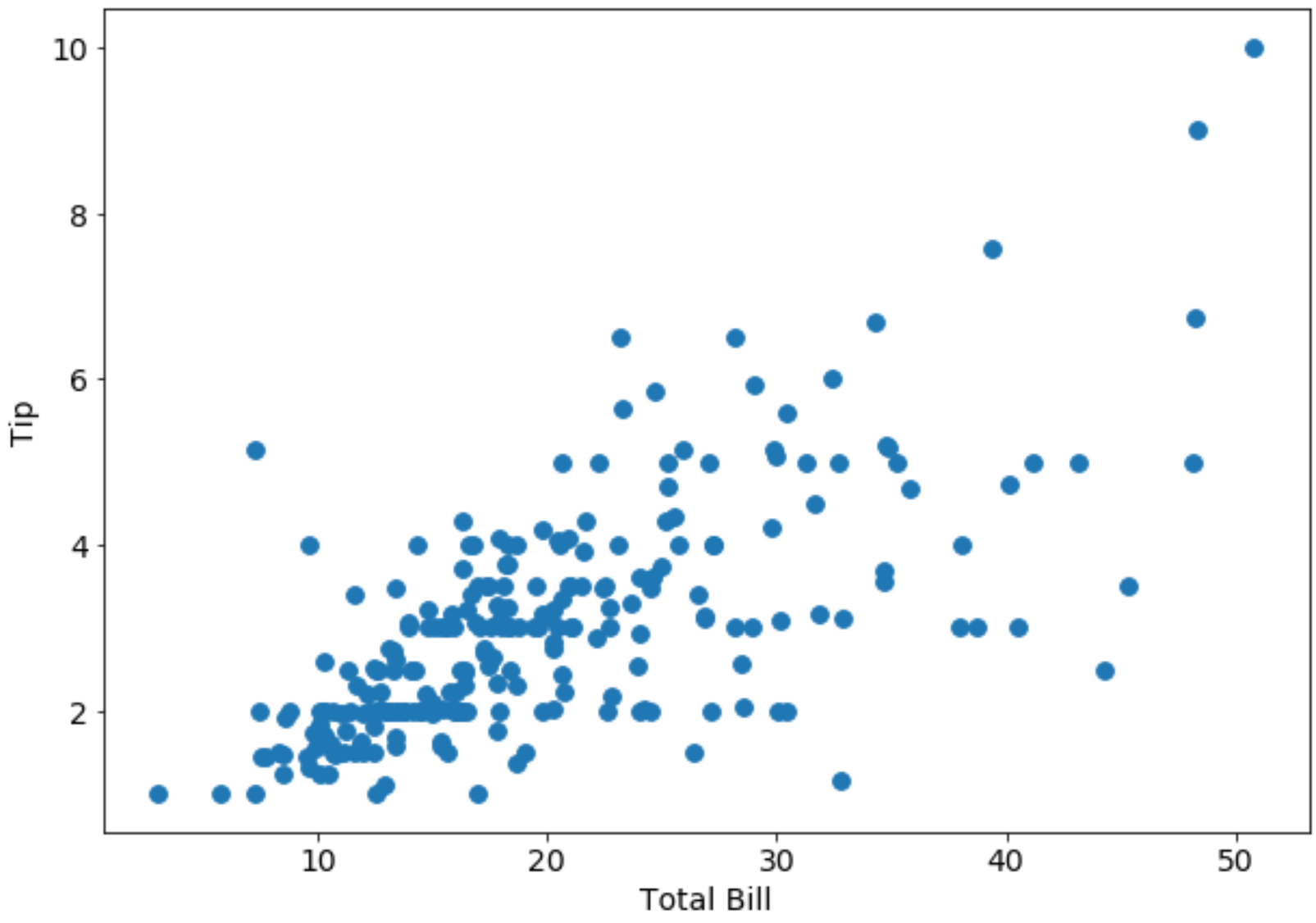
- Scikit-learn is a comprehensive open source Python package for machine learning and data analysis: http://scikit-learn.org
- Anaconda includes Scikit-learn as part of its free distribution.
- Scikit-learn algorithm inputs and outputs are usually represented as NumPy arrays, although we can also work with input data as Pandas Data Frames.

```
import sklearn
```



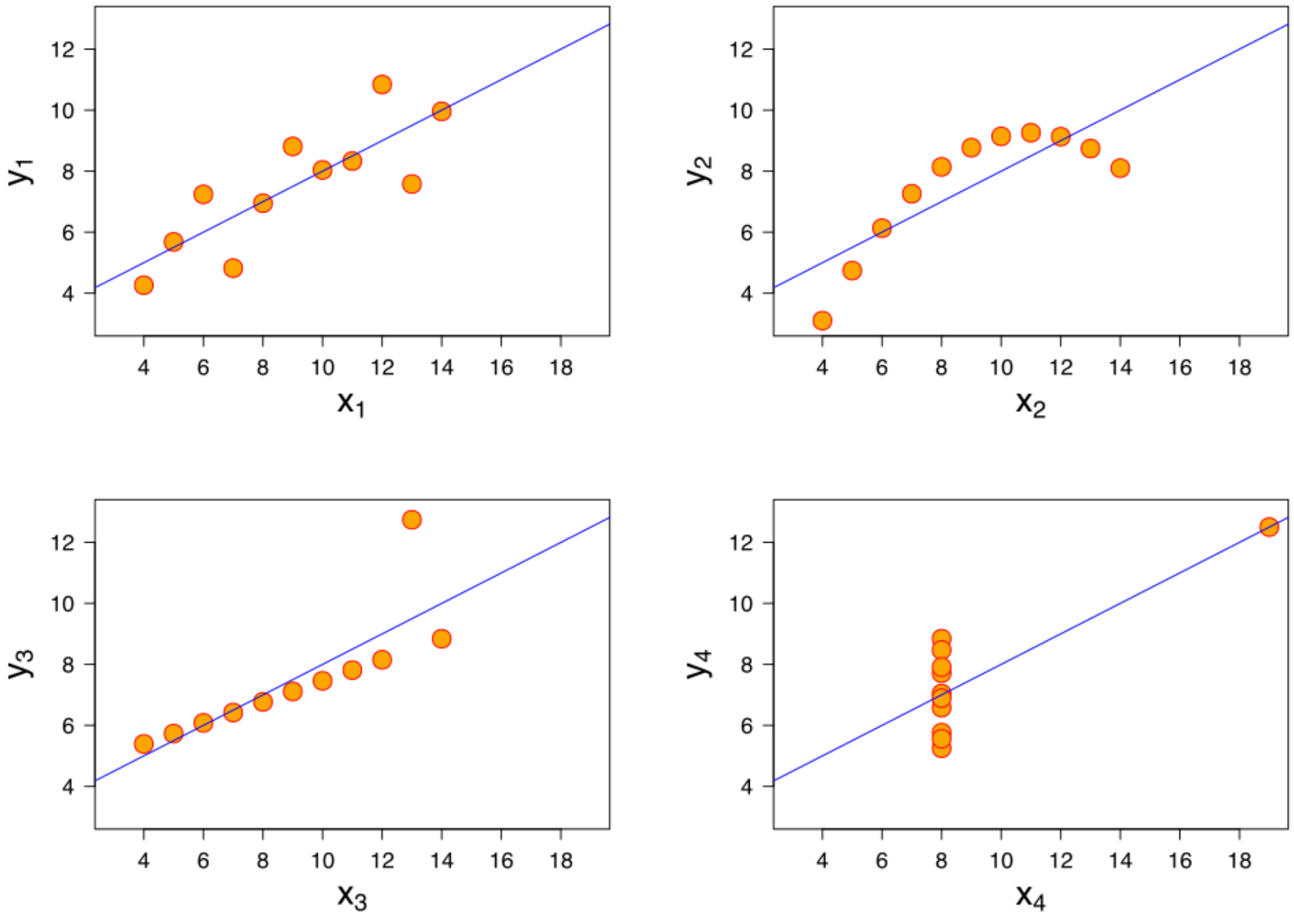
Finding Patterns in Data

- When analysing a new dataset, as a first step we might visualise the data to identify any obvious patterns.
- **Example:** Dataset of 244 meals, with details of total meal bill and tip amount. What can we say about the data?



Anscombe's Quartet

- 4 datasets with nearly identical statistical properties. Yet, each expresses quite different relationships between Y and X. Important to look at the data visually before building a model!

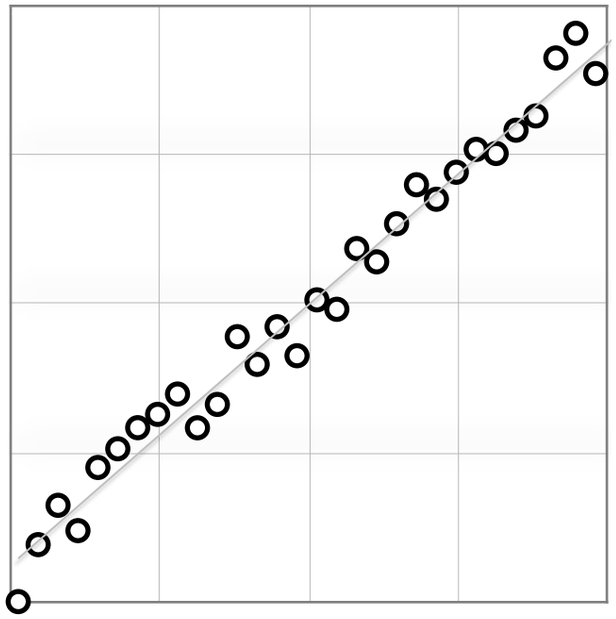


Wikipedia

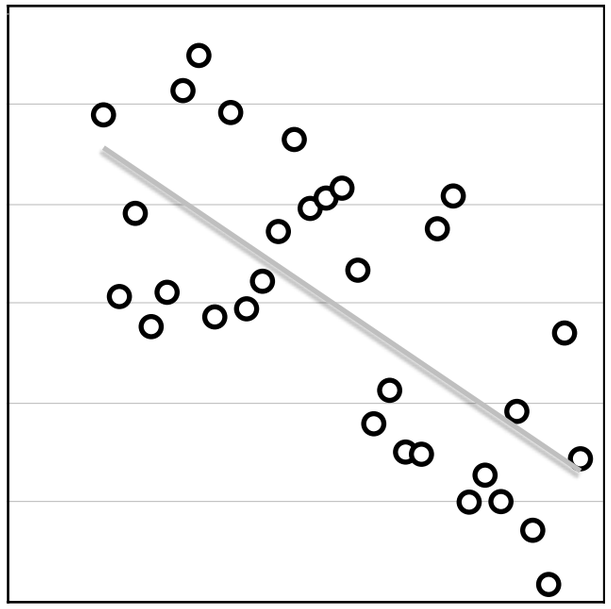
Correlation

- Visualising data using scatter plots can provide us with a sense of the relationship between two variables. Relationships can have different directions (+ve, -ve) and different strengths.
- How do we quantify this numerically?

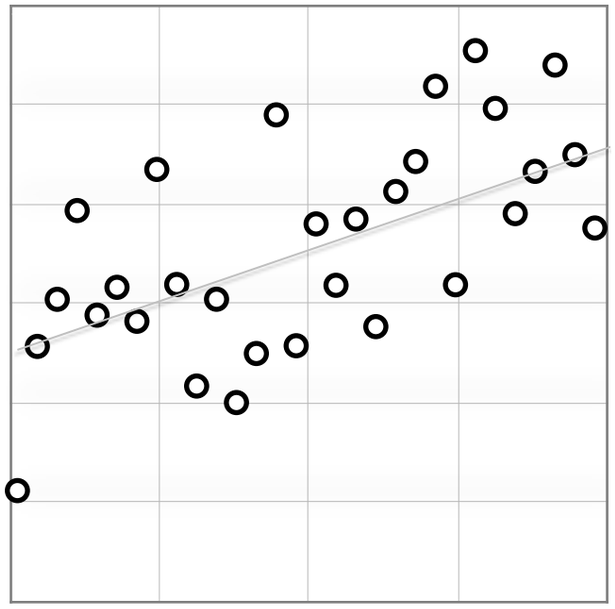
strong *+ve* relationship



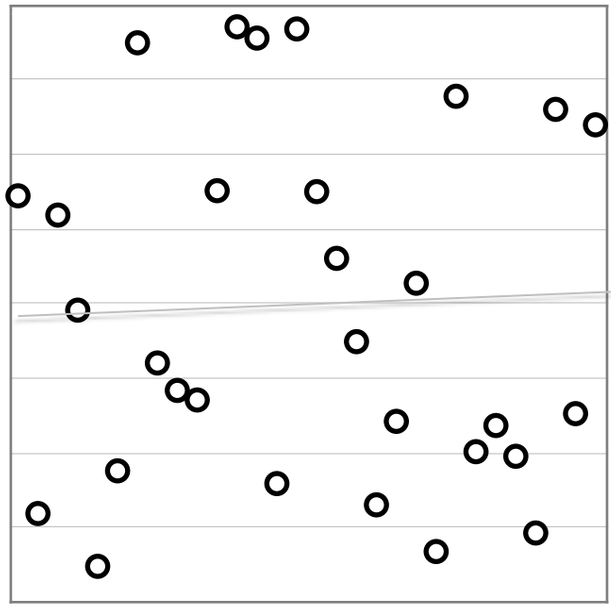
weak *-ve* relationship



weak *+ve* relationship



no relationship



Correlation in Python

- Often useful to know the strength of relationship between Y and X, but independent of the units of measurement.
- The Correlation between Y and X is a statical measure of how strongly two variables are related. It is dimensionless, i.e. a unit-free measure of the relationship between variables.
- Takes a value in [-1,+1], where 1 is total positive correlation, 0 is no correlation, –1 is total negative correlation.

$$Cor(Y,X) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{y_i-\bar{y}}{s_y}\right)\left(\frac{x_i-\bar{x}}{s_x}\right) \quad \text{where} \quad s_y^2 = \frac{\sum_i^n (y_i-\bar{y})^2}{n-1}$$

```
x = np.array([0.1, 0.3, 0.4, 0.8, 0.9])
y = np.array([3.2, 2.4, 2.4, 0.1, 5.5])
np.corrcoef(x,y)
```

```
array([[ 1.        , -0.95363007],
       [-0.95363007,  1.        ]])
```

Calling the NumPy `corrcoef(x,y)` function will create a 2x2 Pearson correlation coefficient matrix.

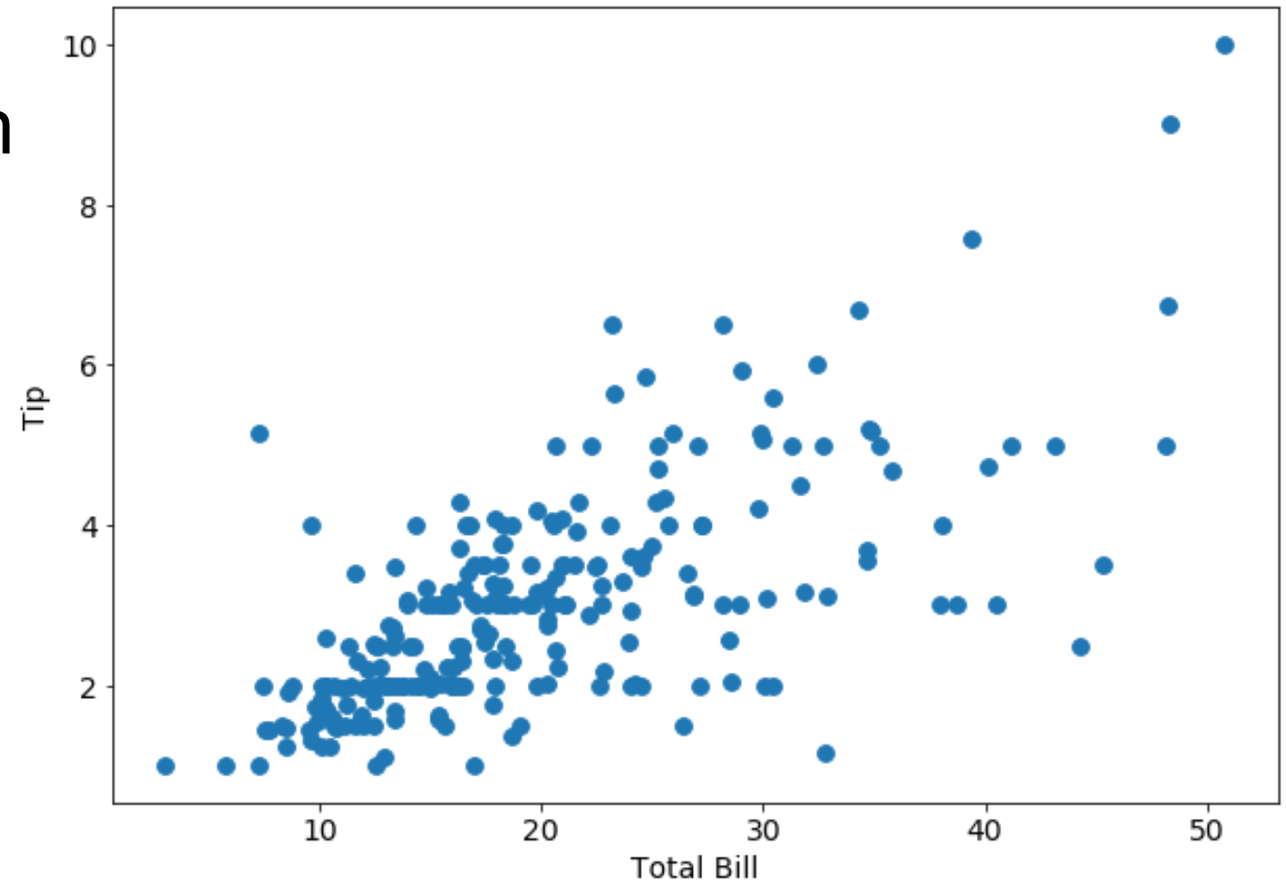
The off-diagonals indicate strength of relationships.

Correlation in Python

- If our data is stored in a Pandas DataFrame, we can also use the `df.corr()` function.

```
df.corr()
```

	total_bill	tip
total_bill	1.000000	0.675734
tip	0.675734	1.000000



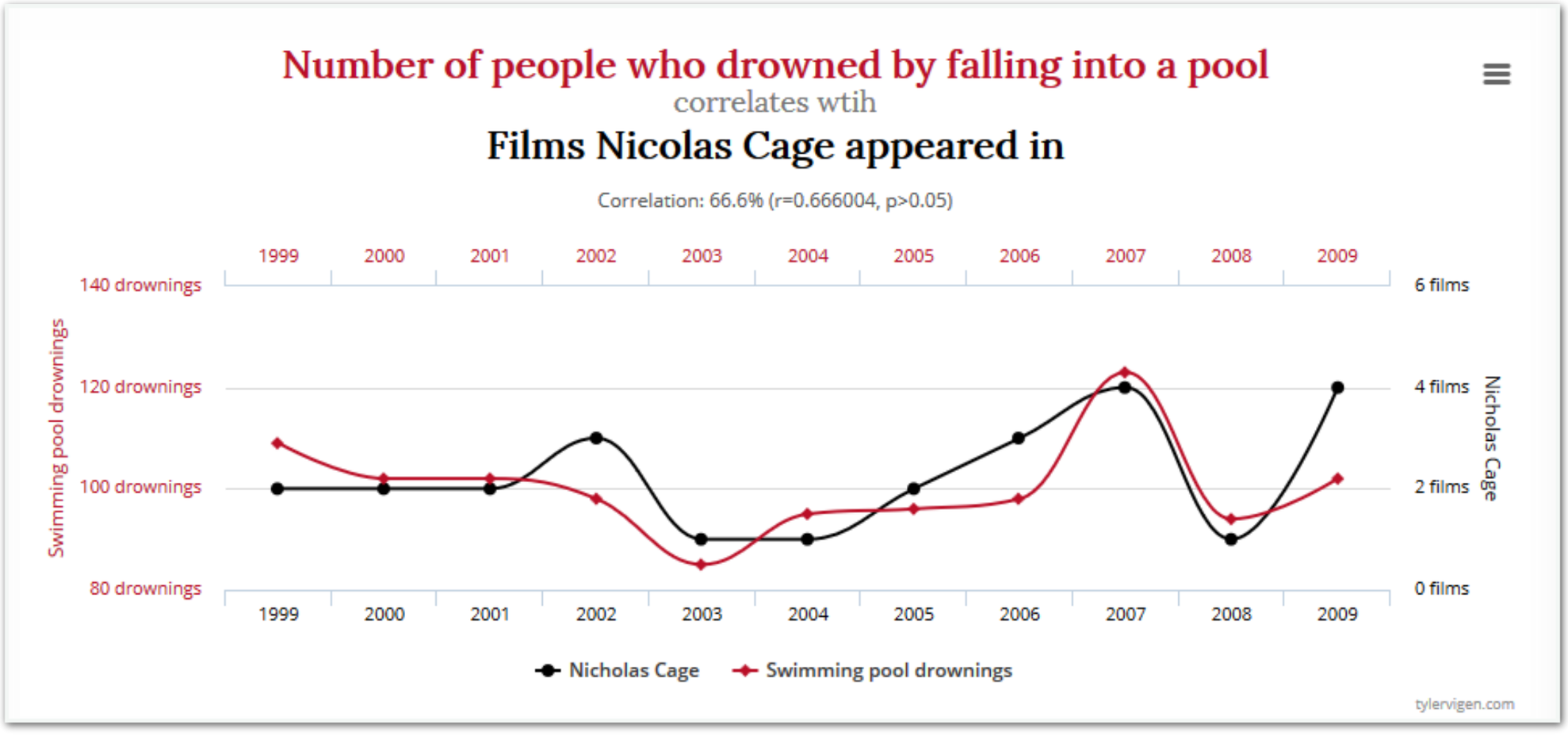
The off-diagonals indicate strength of relationships.

Remember, 1 is total positive correlation, 0 is no correlation, –1 is total negative correlation.

	total_bill	tip
0	16.99	1.01
1	10.34	1.66
2	21.01	3.50
3	23.68	3.31
4	24.59	3.61

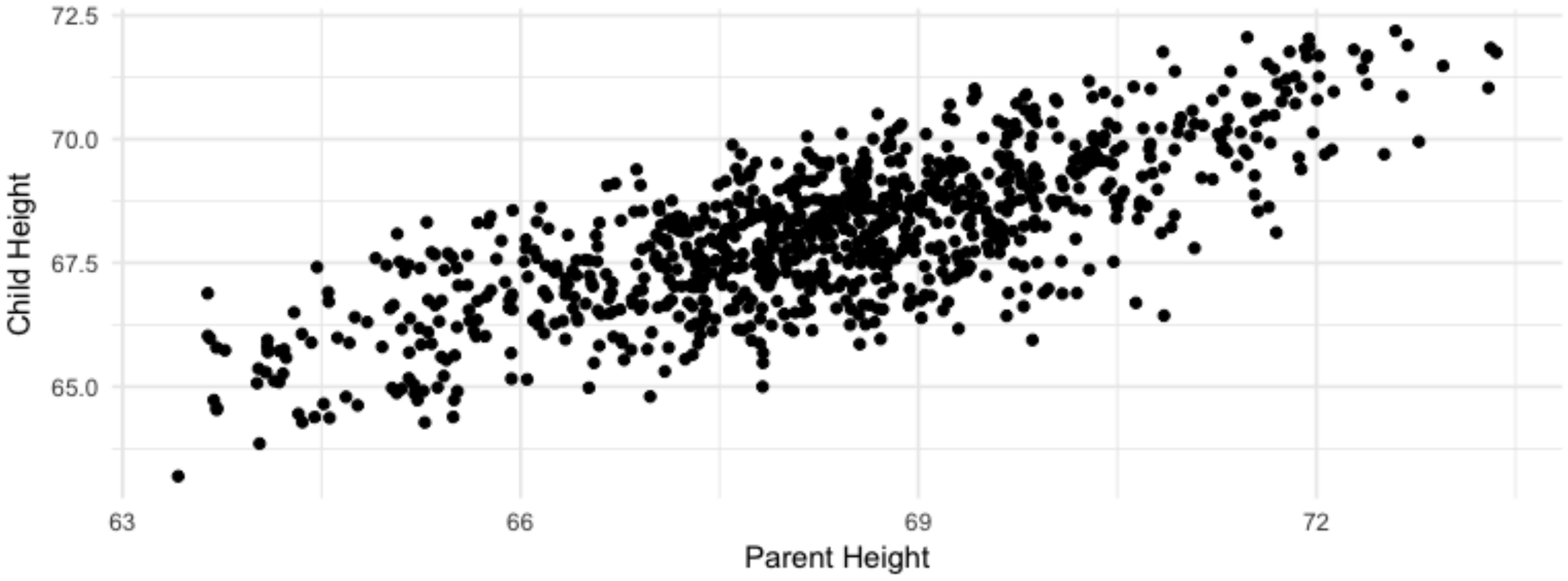
Correlation vs Causation

- Causation: indicates that one event is the result of the occurrence of the other event - i.e. there is a causal relationship between the two events.
- But a correlation between variables does not automatically mean that the change in one variable is the cause of the change in the values of the other variable.



Regression

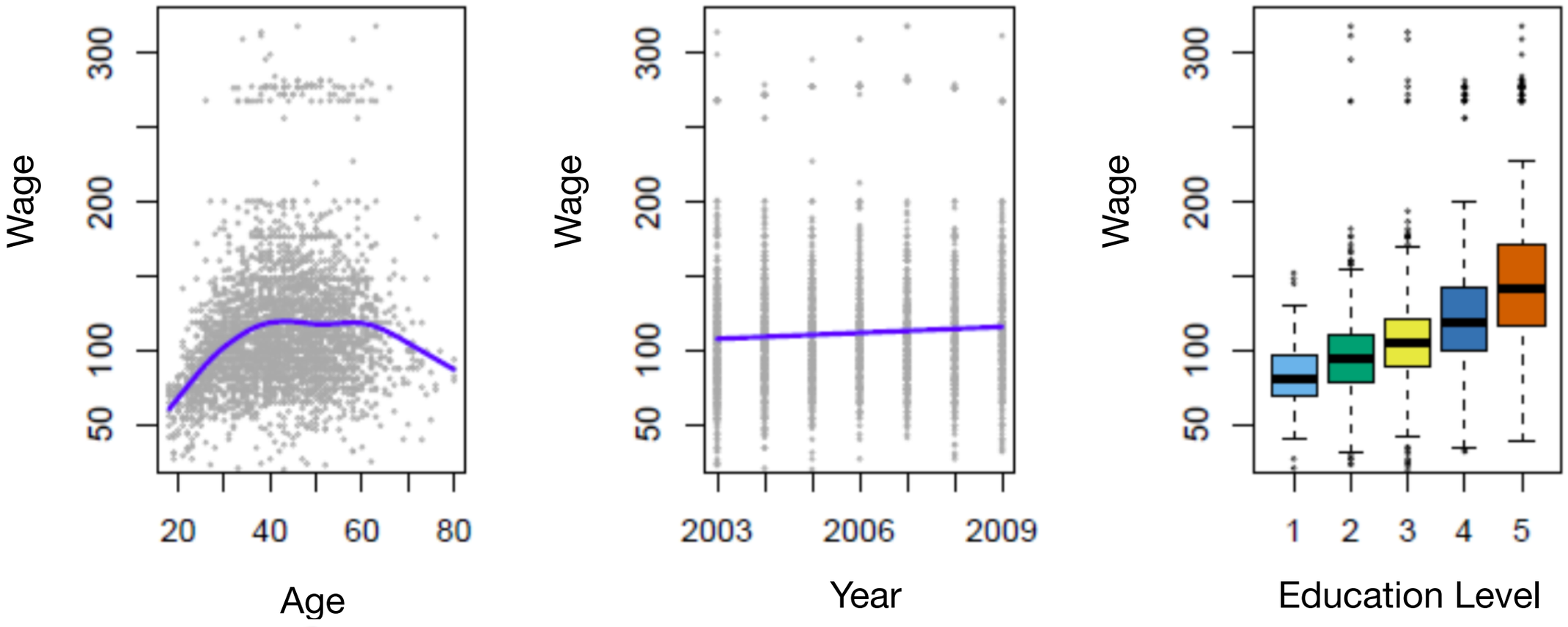
- Regression analysis: A common statistical process for estimating the relationships between variables. This can allow us to make numeric predictions based on past data.
- **Simple example**: Can we predict a child's height, based on their parent's height?



https://web.stanford.edu

Regression

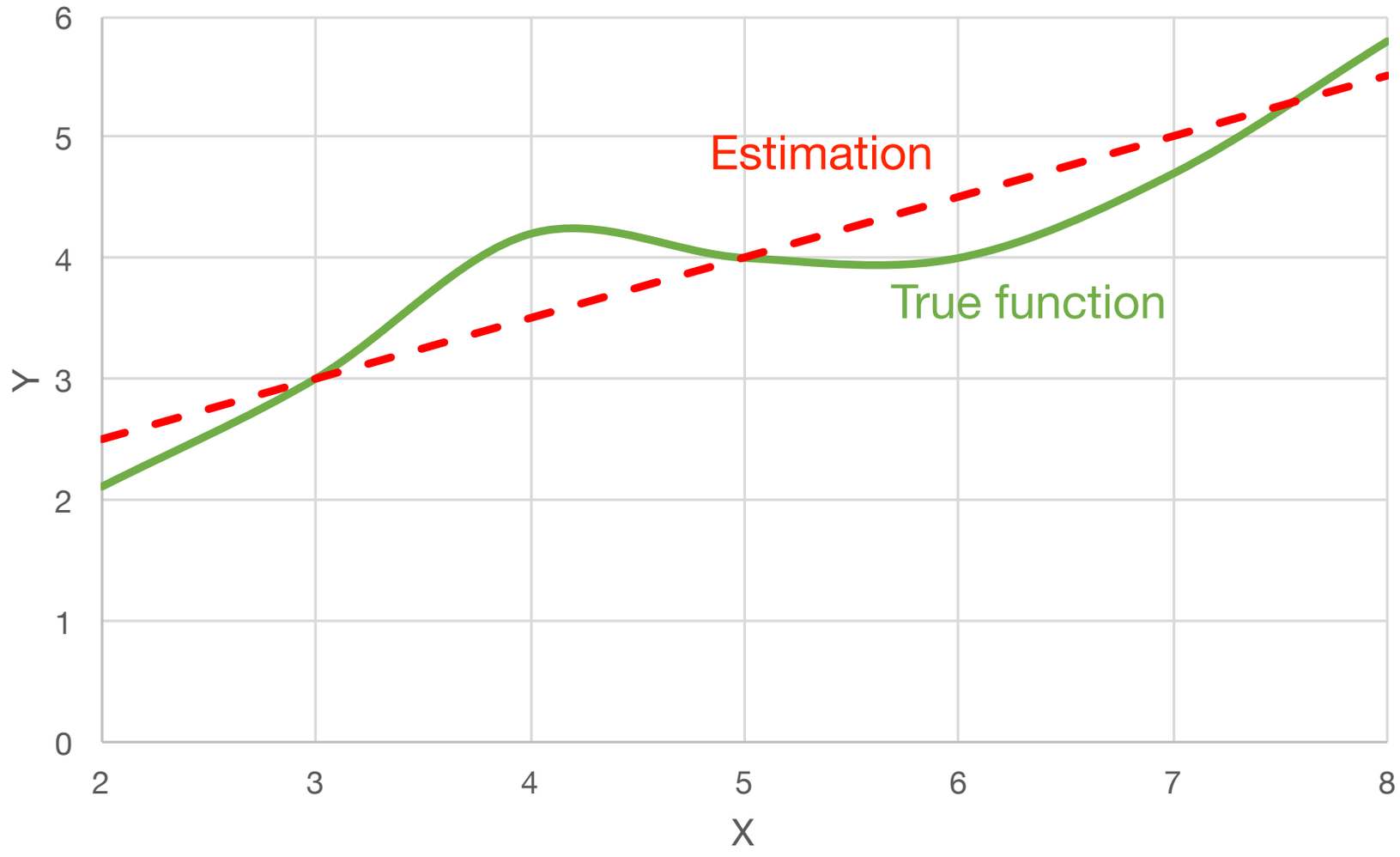
- Regression analysis: A common statistical process for estimating the relationships between variables. This can allow us to make numeric predictions based on past data.
- **Example**: Can we establish a relationship between salary level and demographic variables in population survey data?



(Statistical Learning - Hastie & Tibshirani)

Linear Regression

- Linear Regression: a simple approach to predictive modelling. It assumes that the dependence of a response (dependent) variable *Y* on input (independent) variables X_1, X_2, … is linear.
- While true regression functions are never linear, simple linear regression is still often useful.



Linear Regression: Terminology

X independent variables

Y dependent variables

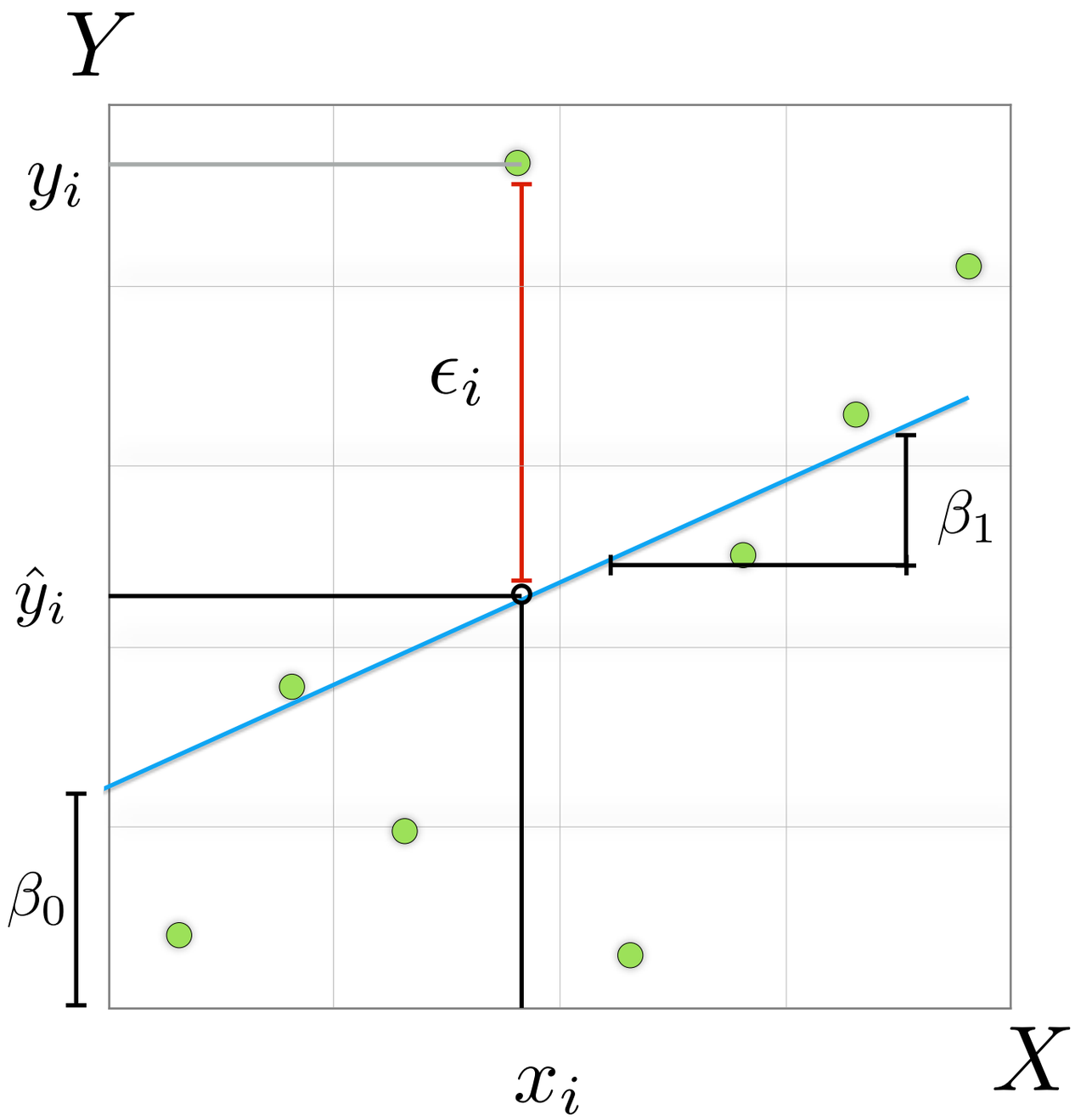
y_i observed value of Y

$\hat{y}_i$ predicted value of Y

β_0 intercept

β_1 slope

ϵ_i error



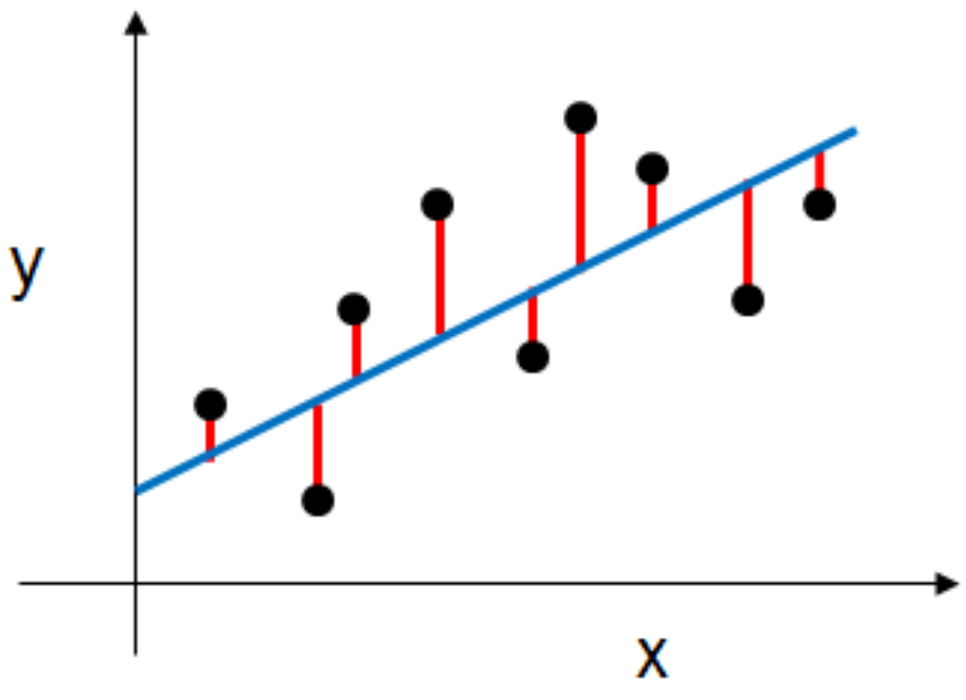
Simple Linear Regression

- Simple Linear Regression: Method for predicting a numeric response using a single input variable (feature).
- The model is: $y = \beta_0 + \beta_1 x$

y : the response variable

x : the input feature

β_0, β_1 : the model coefficients (intercept and slope)



Goal is to learn the model coefficients from existing data.

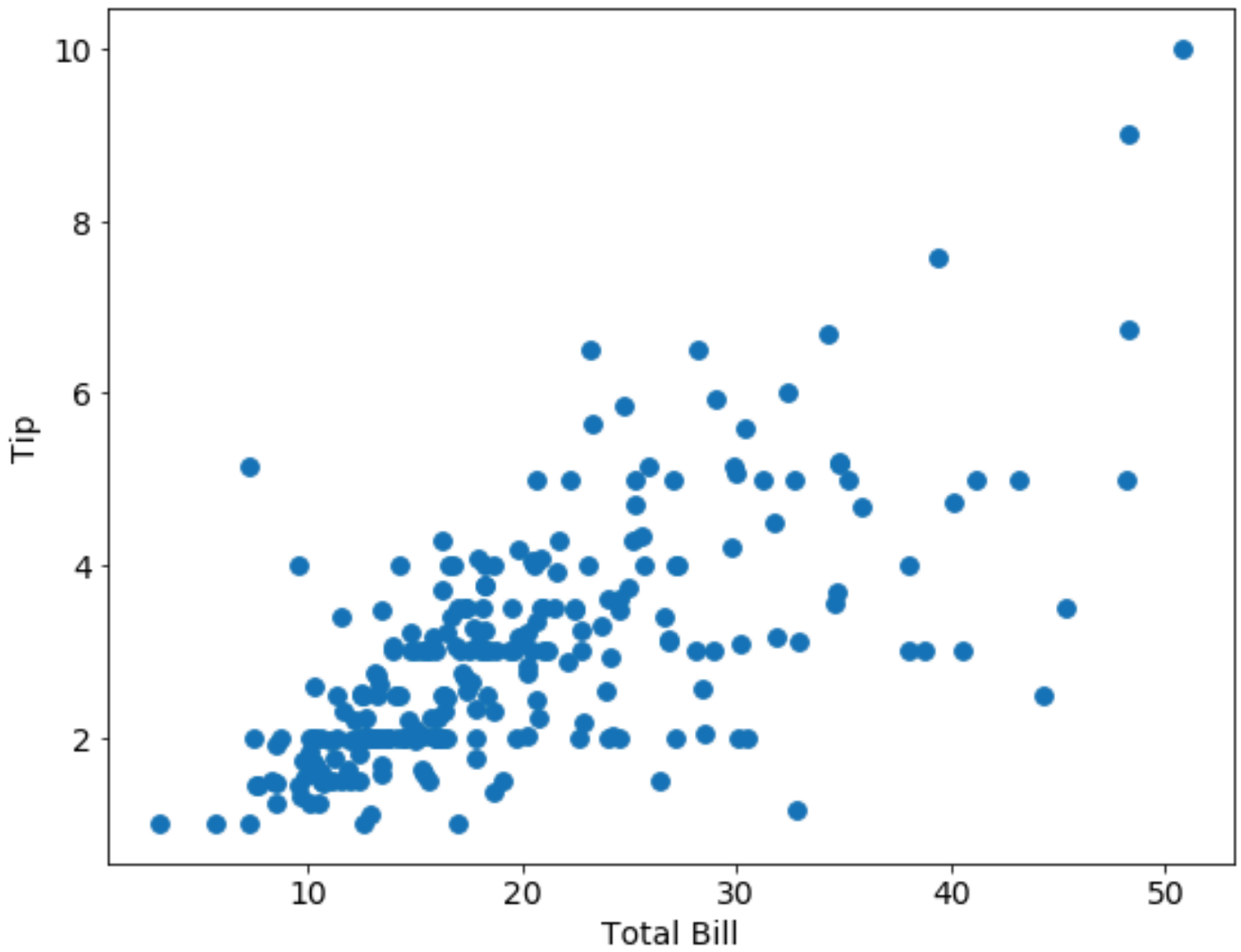
Once we have learned the model, we can make future predictions.

We learn the model by finding the best line (coefficients) which minimises the squared distance between our examples and the line.

Example: Linear Regression

Q. Can we predict the tip amount, from the total bill?

- Independent variable: the total bill amount X
- Dependent variable: the tip amount Y ("response" variable)



Use our historic dataset of 244 meals as the training data to build a new regression model which can then be used to make predictions.

Example: Linear Regression

- Scikit-Learn provides functions to apply linear regression to NumPy arrays. To build a model for input *x* and response *y*:

```
from sklearn.linear_model import LinearRegression
```

Create and fit the model based on the training data

```
model = LinearRegression()
model.fit(x, y)
```

Get the intercept coefficient

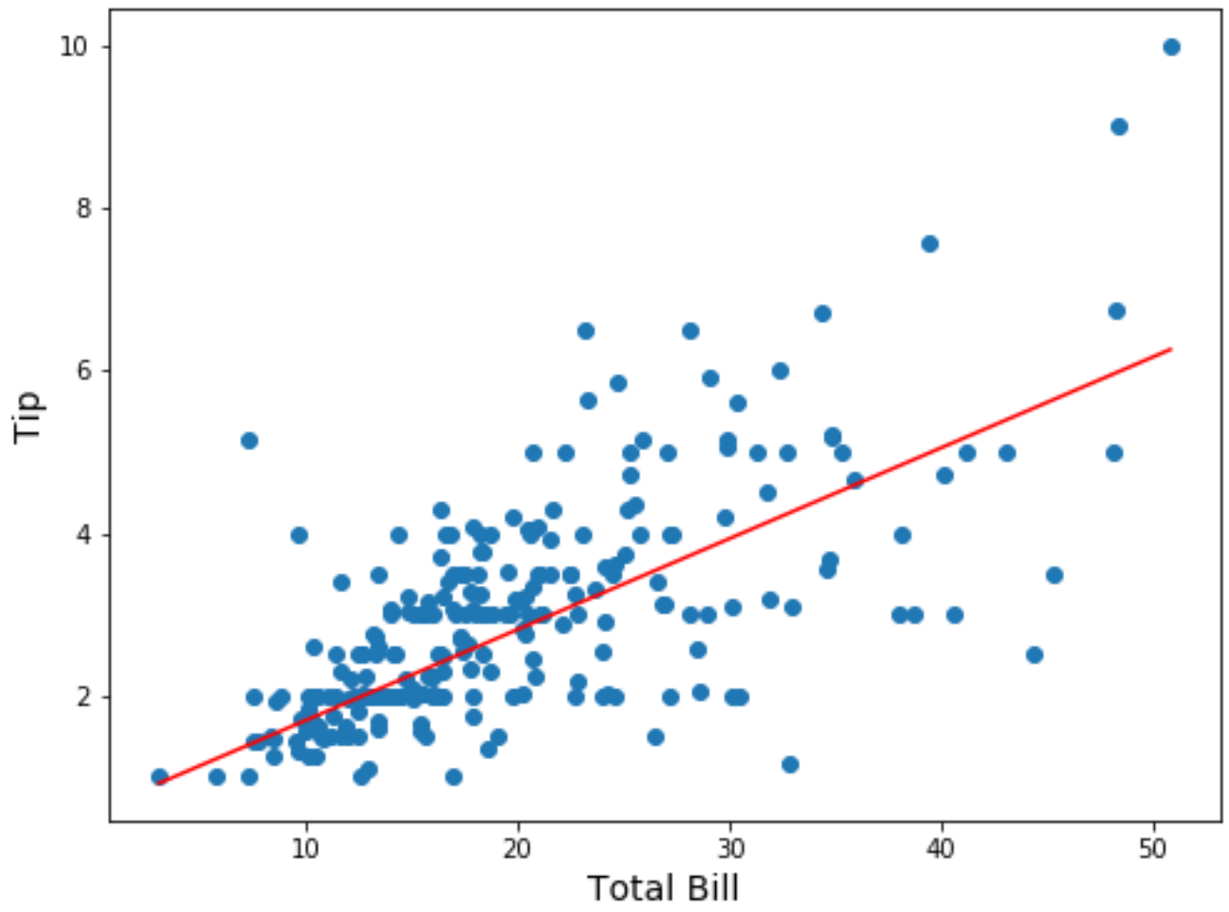
```
model.intercept_
```

```
0.92026961
```

Get the slope coefficient

```
model.coef_[0]
```

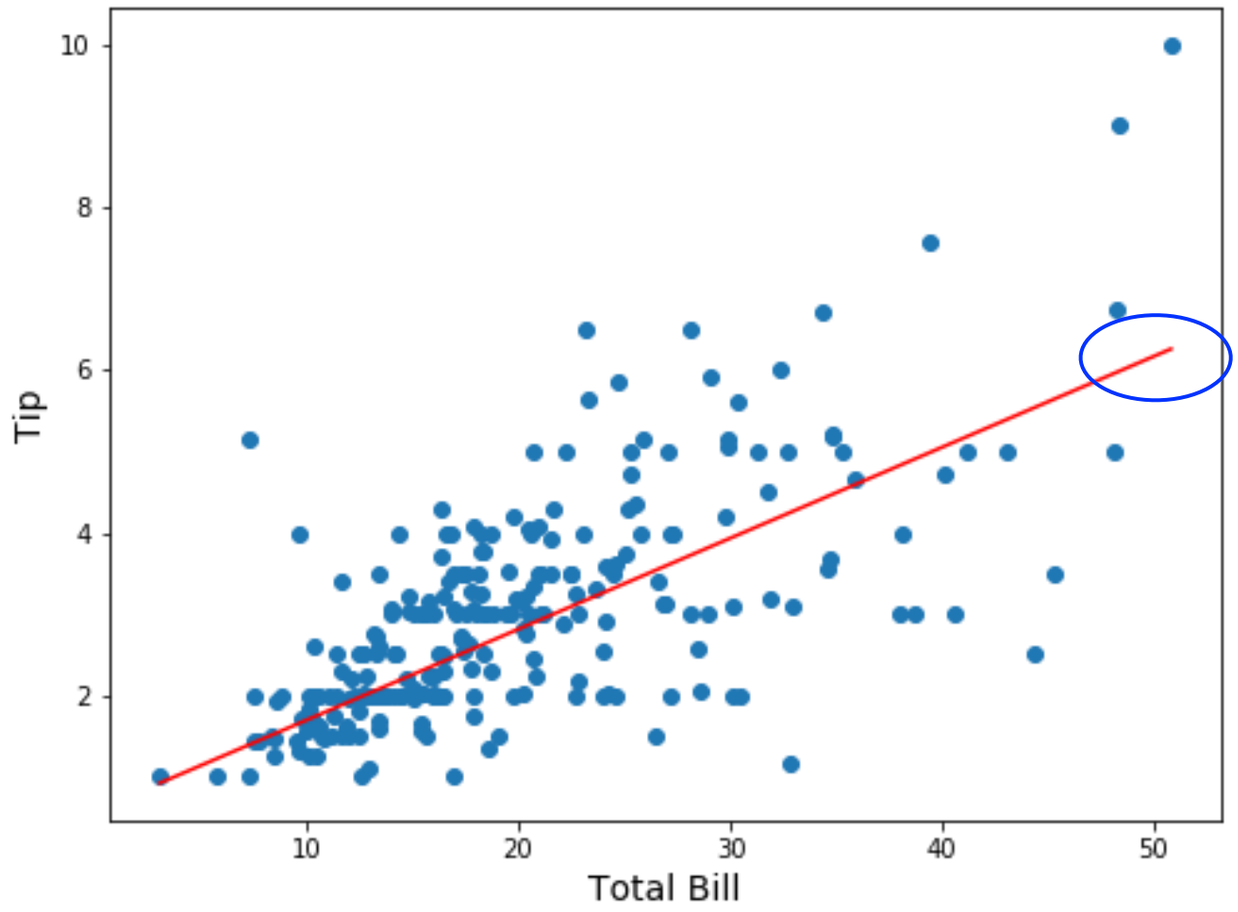
```
0.10502452
```



Example: Linear Regression

- Now that we have a built our regression model, we can use it to make predictions - i.e. predict the tip amount, based on some specified amount for the total bill.

Bill	*Predicted Tip*
€10.00	€1.97
€15.00	€2.50
€20.00	€3.02
€25.00	€3.55
€30.00	€4.07
€35.00	€4.60
€40.00	€5.12
€45.00	€5.65
€50.00	€6.17
€55.00	€6.70
€60.00	€7.22
€65.00	€7.75

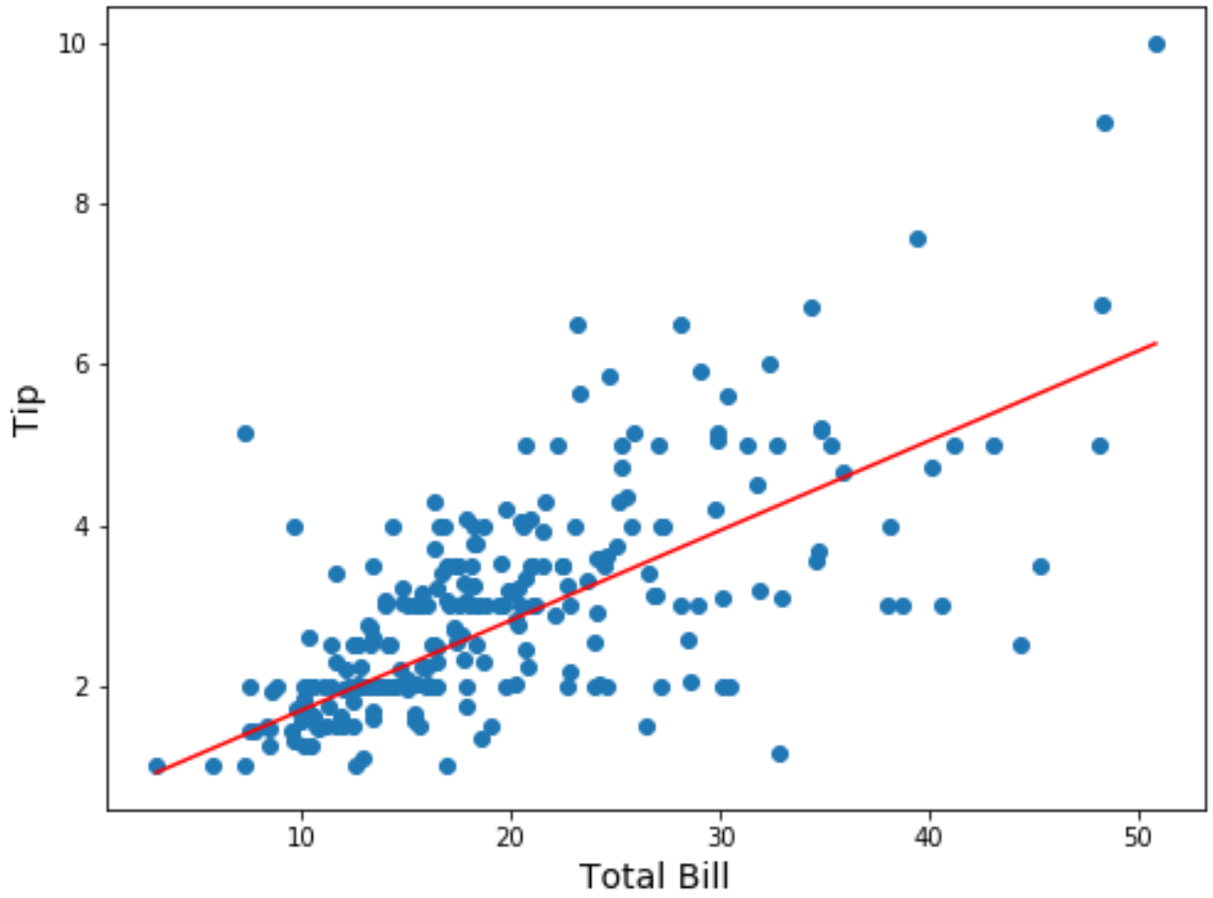


e.g. Predicted tip for €50 meal is €6.17

Example: Linear Regression

- We can also compare the output of our model with the original data to see if it agrees. When we look at the first 10 rows of the training data, we see there are some errors. Our regression model does not fit the data perfectly.

Bill	*Predicted Tip*	*Actual Tip*
€16.99	€2.70	€1.01
€10.34	€2.01	€1.66
€21.01	€3.13	€3.50
€23.68	€3.41	€3.31
€24.59	€3.50	€3.61
€25.29	€3.58	€4.71
€8.77	€1.84	€2.00
€26.88	€3.74	€3.12
€15.04	€2.50	€1.96
€14.78	€2.47	€3.23



Example: Linear Regression

Example: What is the relationship between budget spent on different advertising media and product sales?

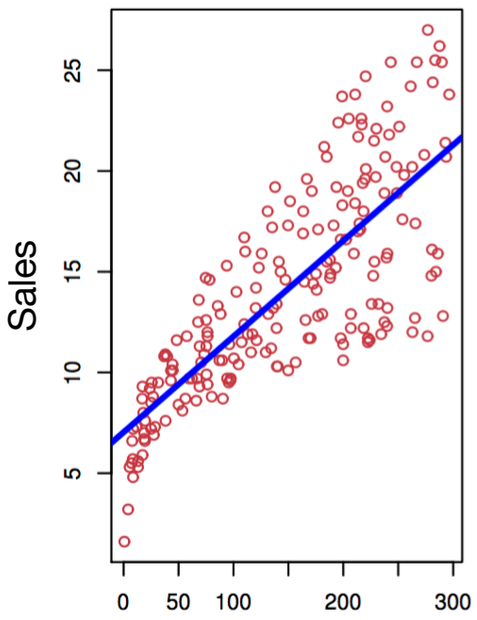
TV	Radio	Newspaper	Sales
230.10	37.80	69.20	22.10
44.50	39.30	45.10	10.40
17.20	45.90	69.30	9.30
151.50	41.30	58.50	18.50
180.80	10.80	58.40	12.90

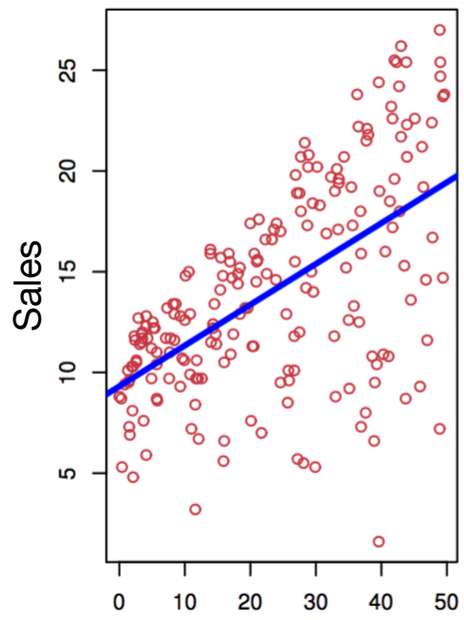
Input Features:

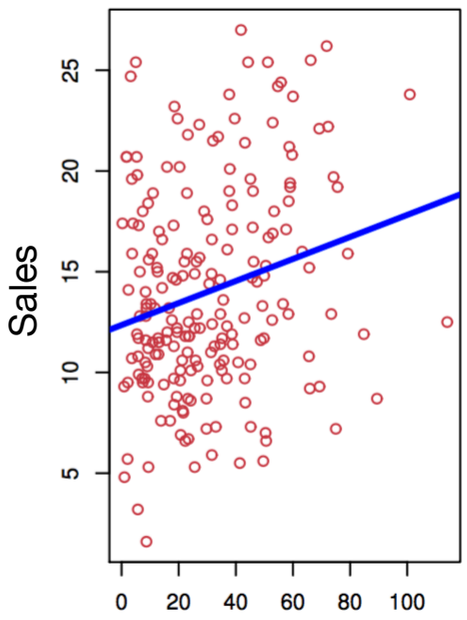
1. TV: Budget (€) spent on TV advertising
2. Radio: Budget (€) spent on Radio advertising
3. Newspaper: Budget (€) spent on print advertising

Response:

- Sales: Overall product sales (€)







Which advertising medium has the greatest impact on product sales?

Can we predict future sales?

We could examine scatter plots of each feature vs sales from our historic data

(Hastie & Tibshirani)

Example: Linear Regression

- **Example:** What is the relationship between budget spent on different advertising media and product sales?

```
from sklearn.linear_model import LinearRegression
df = pd.read_csv("advertising.csv", index_col=0)
x = df[["TV"]]
```

Let's examine the relationship between TV budget and sales

Create and fit the model

```
model = LinearRegression()
model.fit(x, df["Sales"])
```

Get the intercept coefficient

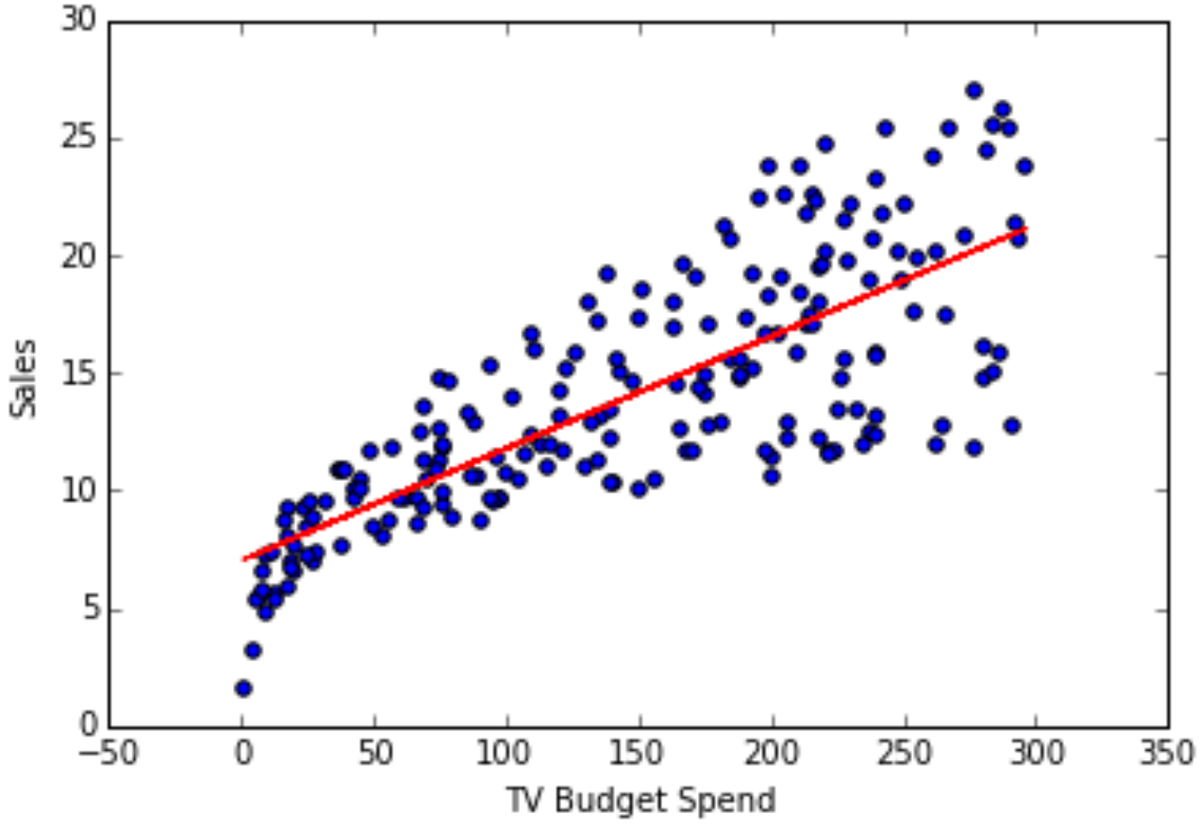
```
model.intercept_
```

```
7.03259354913
```

Get the slope coefficient

```
model.coef_[0]
```

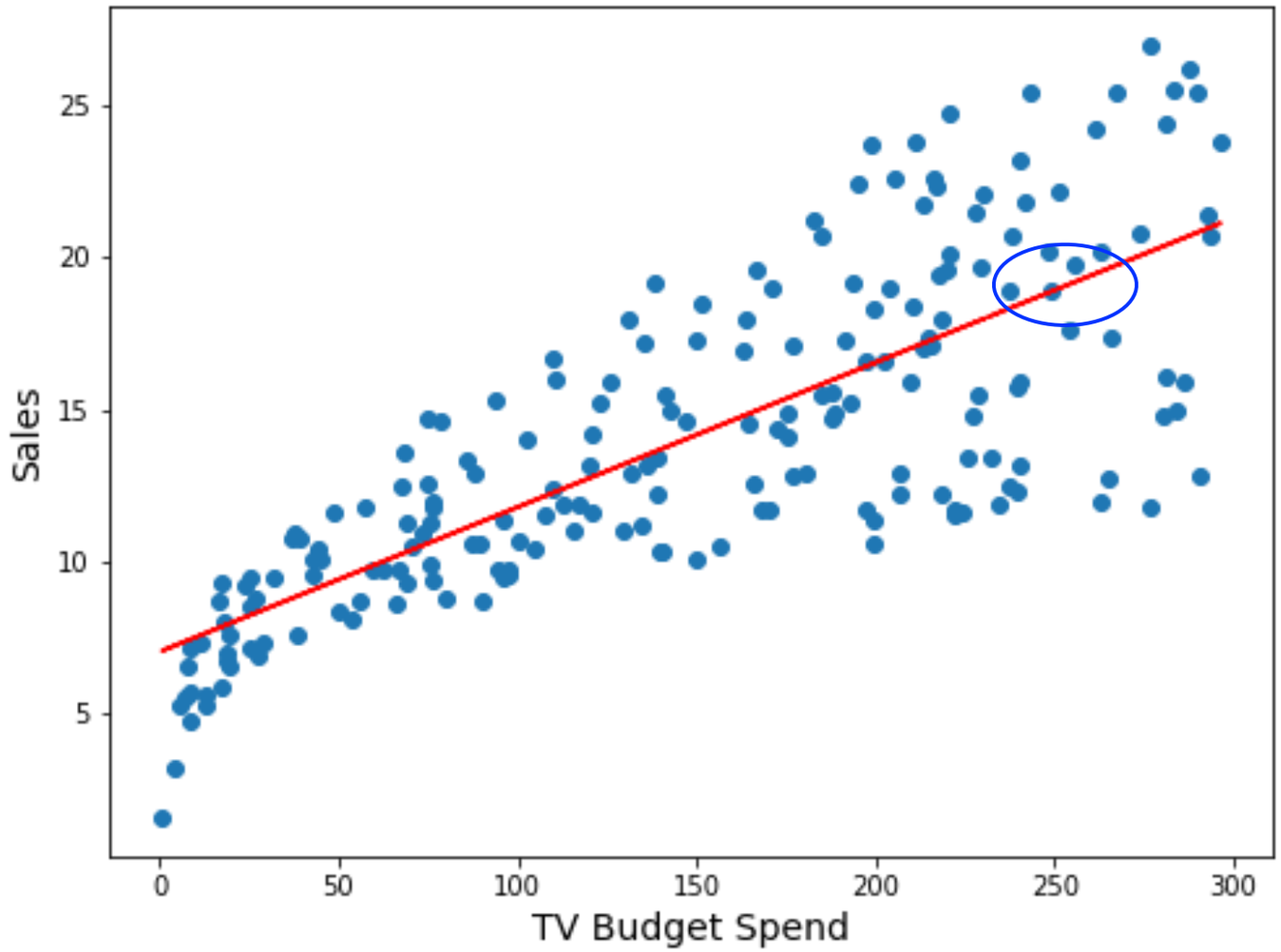
```
0.047536640433
```



Example: Linear Regression

- Now that we have a built our regression model, we can use it to make predictions - i.e. predict sales revenue, based on TV advertising budget spend.

Budget	*Predicted Sales*
€0.00	€7.03
€50.00	€9.41
€100.00	€11.79
€150.00	€14.16
€200.00	€16.54
€250.00	€18.92
€300.00	€21.29
€350.00	€23.67



e.g. Predicted sales for €250 budget is €18.92

Multiple Regression

- Multiple linear regression: Simple linear regression can easily be extended to include multiple features, where we try to learn a model: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$
- Each feature x_i has its own coefficient β_i

TV	Radio	Newspaper	Sales
230.10	37.80	69.20	22.10
44.50	39.30	45.10	10.40
17.20	45.90	69.30	9.30
151.50	41.30	58.50	18.50
180.80	10.80	58.40	12.90

e.g. Predict sales based on 3 features

$$y = \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{Radio} + \beta_3 \times \text{Newspaper}$$

Remove column we want to predict

```
x = df.drop("Sales",axis=1)
```

Fit the model based on all 3 features

```
model = LinearRegression()
model.fit(x,df["Sales"])
```

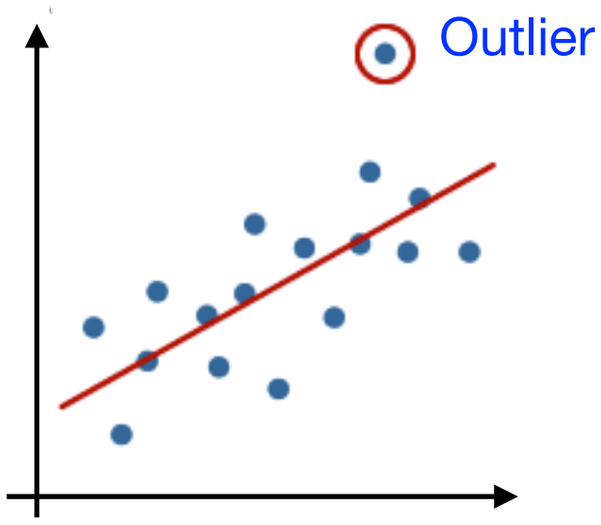
We can now use the model to make sales predictions for unseen values of *(TV, Radio, Newspaper)*

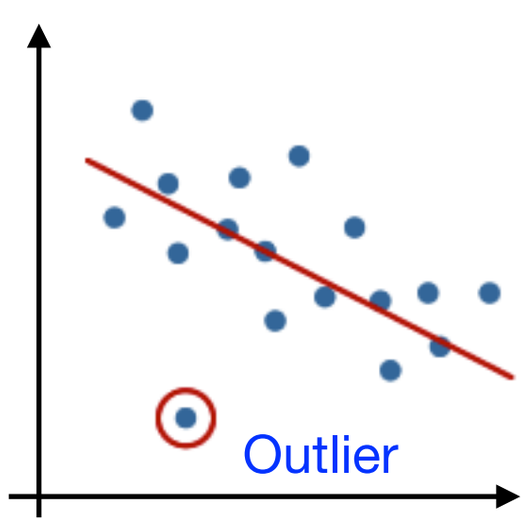
```
model.predict(test_x)
```

```
20.52
```

Outliers in Regression

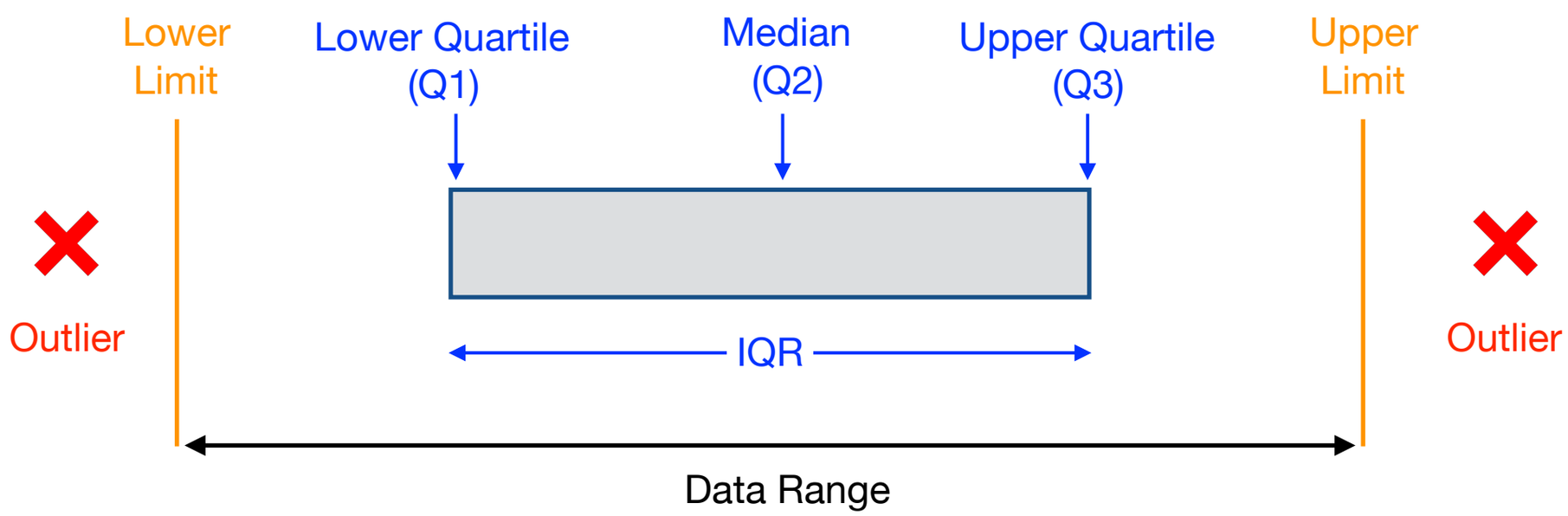
- Outlier: An observed data point that has a dependent variable value which is very different to the value predicted by the regression equation.
- Linear regression models assume there should be no significant outliers, as they can lead to a very poor regression fit.
- In some cases we might drop the outlier and recalculate the regression model. But it is always important to investigate the nature of the outlier before deciding whether to drop.





Finding Outliers

- Box plot diagram: a graphical method which helps in defining the upper limit and lower limits beyond which any data points lying will be considered as outliers.
- Median (Q2): the middle value of the dataset.
- Lower quartile (Q1): the median of the lower half of the dataset.
- Upper quartile (Q3): the median of the upper half of the dataset.
- Interquartile range (IQR): the spread of the middle 50% of the data values = Q3 - Q1



Boxplots in Python

- We can create boxplots in Python to visualise the distributions of values for different variables, and look for any potential outliers.

Create a boxplot with Matplotlib:

```
plt.figure()
plt.boxplot(data)
```

Create a boxplot from all of the columns in a Pandas Data Frame:

```
df.boxplot()
```

- Remember, it is always important to investigate the nature of an outlier before deciding whether to drop.

Possible outliers?