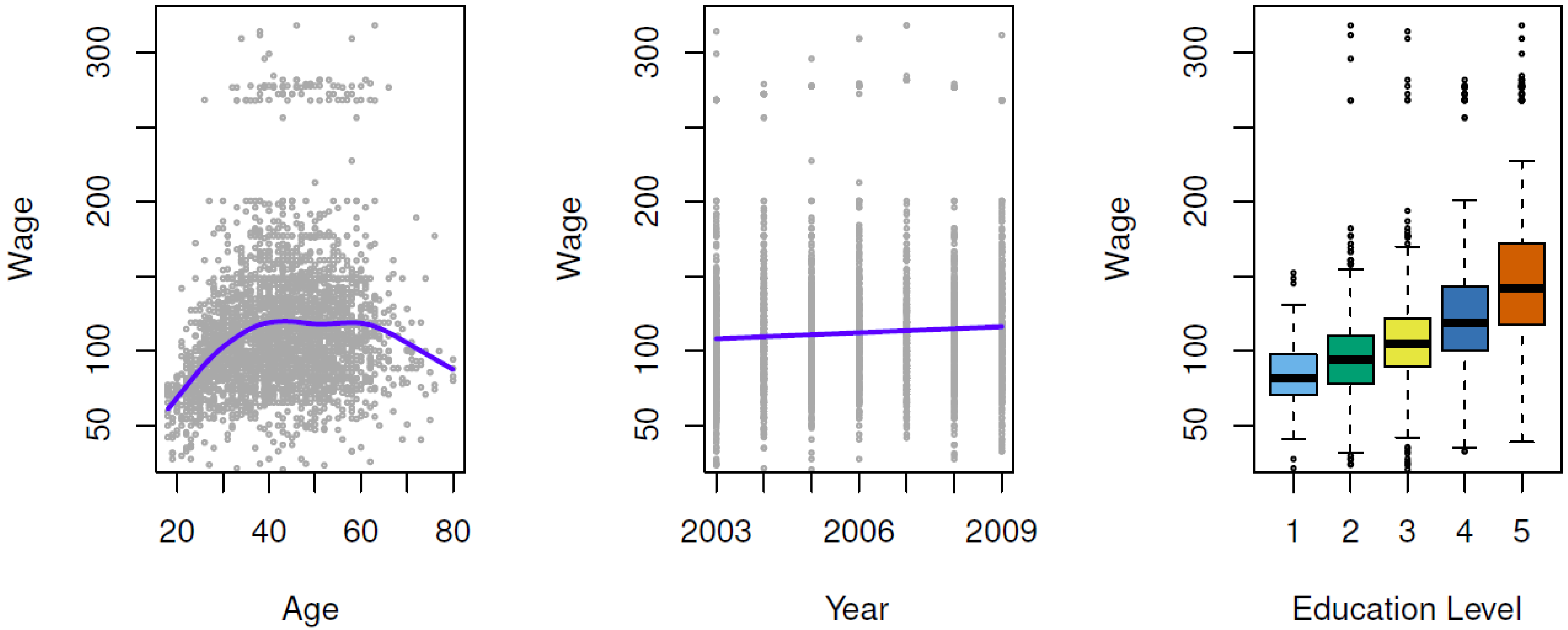
Introduction to Statistical Learning

Dr. Adel Abusitta

What is Statistical Learning

- Statistical learning is a set of tools for understanding data.
- There are two main types of statistical learning: **supervised** and **unsupervised**.
- Supervised statistical learning involves building a statistical model to predict an output based on one or more inputs.
- Unsupervised statistical learning involves learning relationships and structure from data that have inputs but no supervising output.
- **Statistical learning has applications** in various fields such as cybersecurity, medicine, and engineering.

Example: Wage data



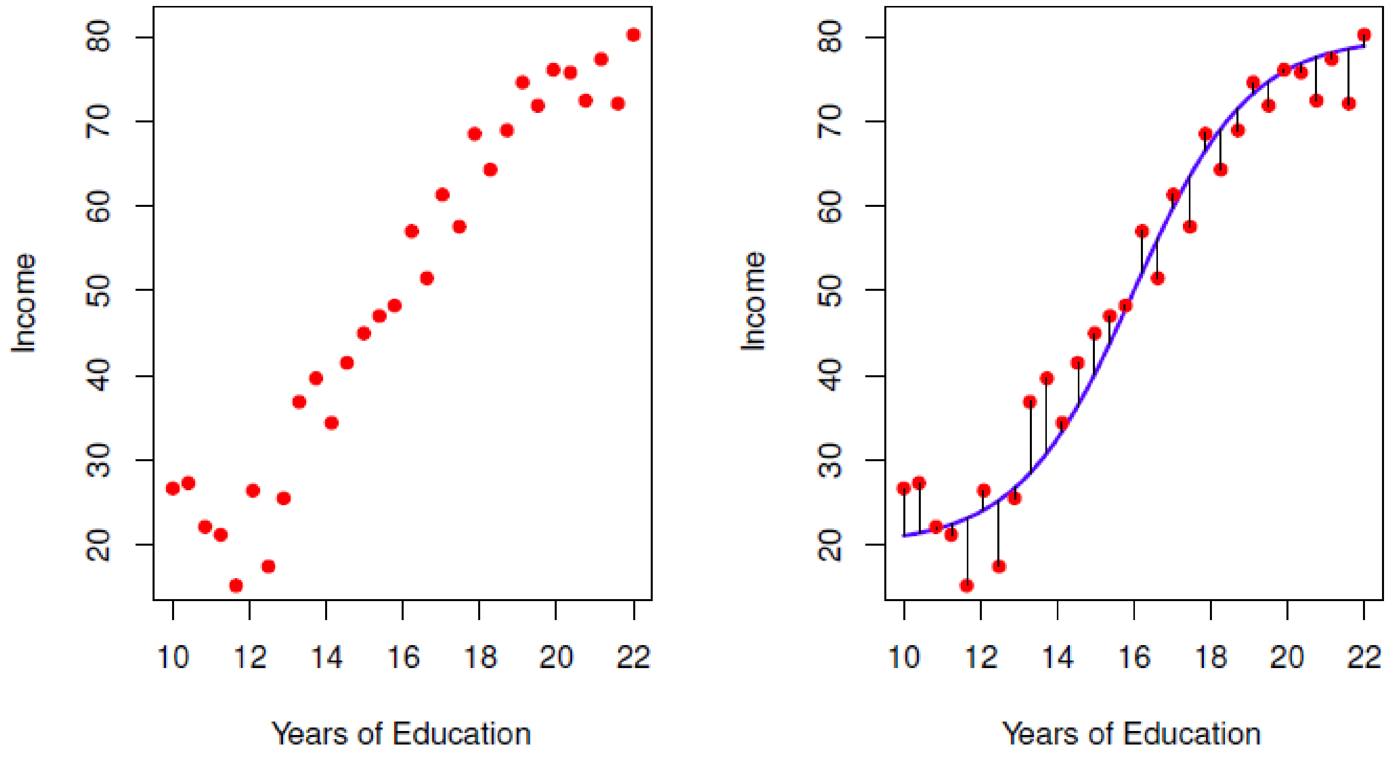
What is Statistical Learning

- More generally, suppose that we observe a quantitative response Y and p different predictors, X1, X2, . . . ,Xp. We assume that there is some relationship between Y and X = (X1, . . . ,Xp), which can be written in the very general form:

$$Y = f(X) + \epsilon.$$

Here f is some fixed but unknown function of $X_1, \ldots, X_p$, and ϵ is a random error term

Example: Income vs. Education



The Income data set. Left: The red dots are the observed values of income (in tens of thousands of dollars) and years of education for 30 individuals. Right: The blue curve represents the true underlying relationship between income and years of education

Why Estimate f?

There are two main reasons that we may wish to estimate f: prediction and inference.

- Prediction

Prediction refers to the process of using a statistical model to **estimate or forecast the outcome of a future** event or observation. In other words, it involves using available data to make predictions about what will happen in the future. For example, a stock market analyst might use historical stock price data and other market variables to predict the future price of a particular stock.

- inference

Inference, on the other hand, refers to the process of using a statistical model to **draw conclusions or make generalizations about a population based on a sample of data.** It involves using available data to learn about a larger population or phenomenon. For example, a researcher might collect a sample of data on the academic performance of students in a particular school district and use statistical inference techniques to draw conclusions about the academic performance of all students in the district.

How Do We Estimate f?

Generally, two steps:

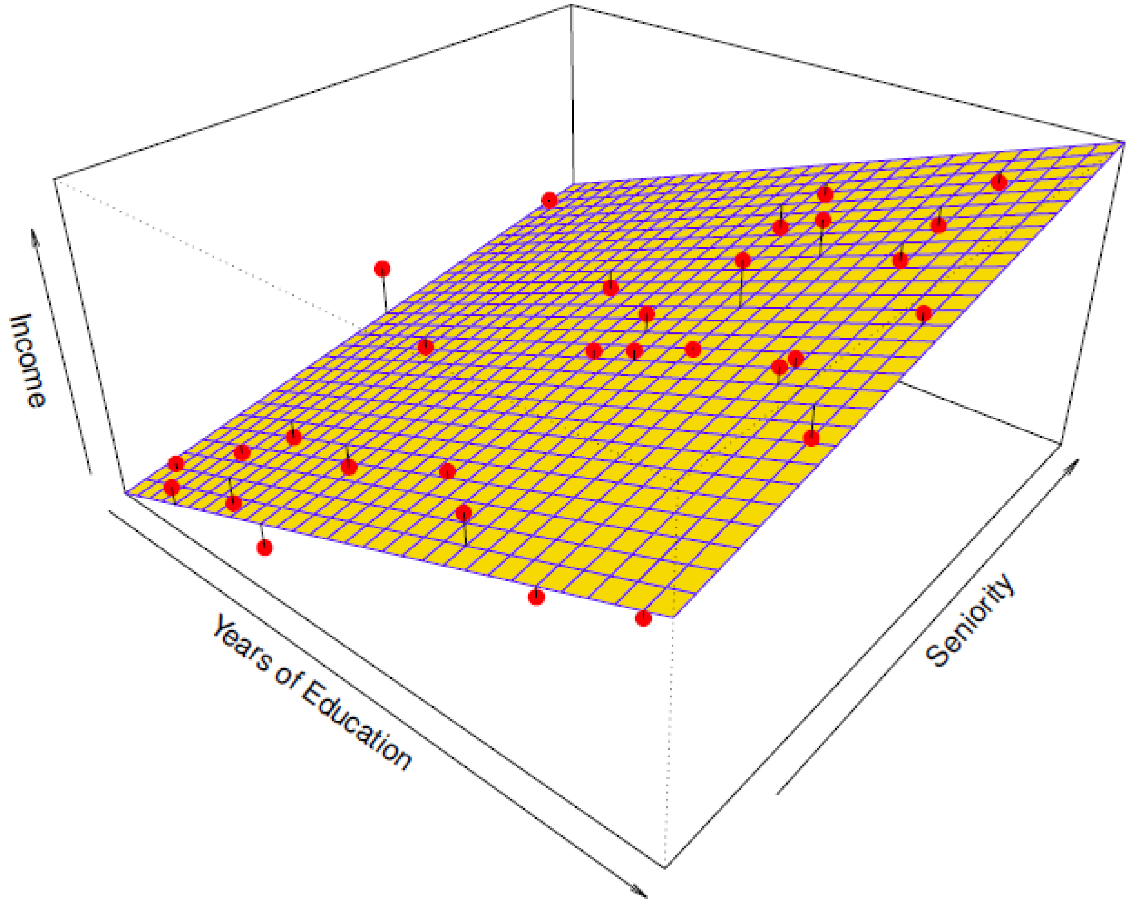
1. First, we make an assumption about the functional form, or shape, of f. For example, one very simple assumption is that f is linear in X:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

2. After a model has been selected, we need a procedure that uses the training data to fit or train the model. In the case of the linear model fit train , we need to estimate the parameters β0,β1, . . . ,βp. That is, we want to find values of these parameters such that:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Example: Income data



$$\texttt{income} \approx \beta_0 + \beta_1 \times \texttt{education} + \beta_2 \times \texttt{seniority}.$$

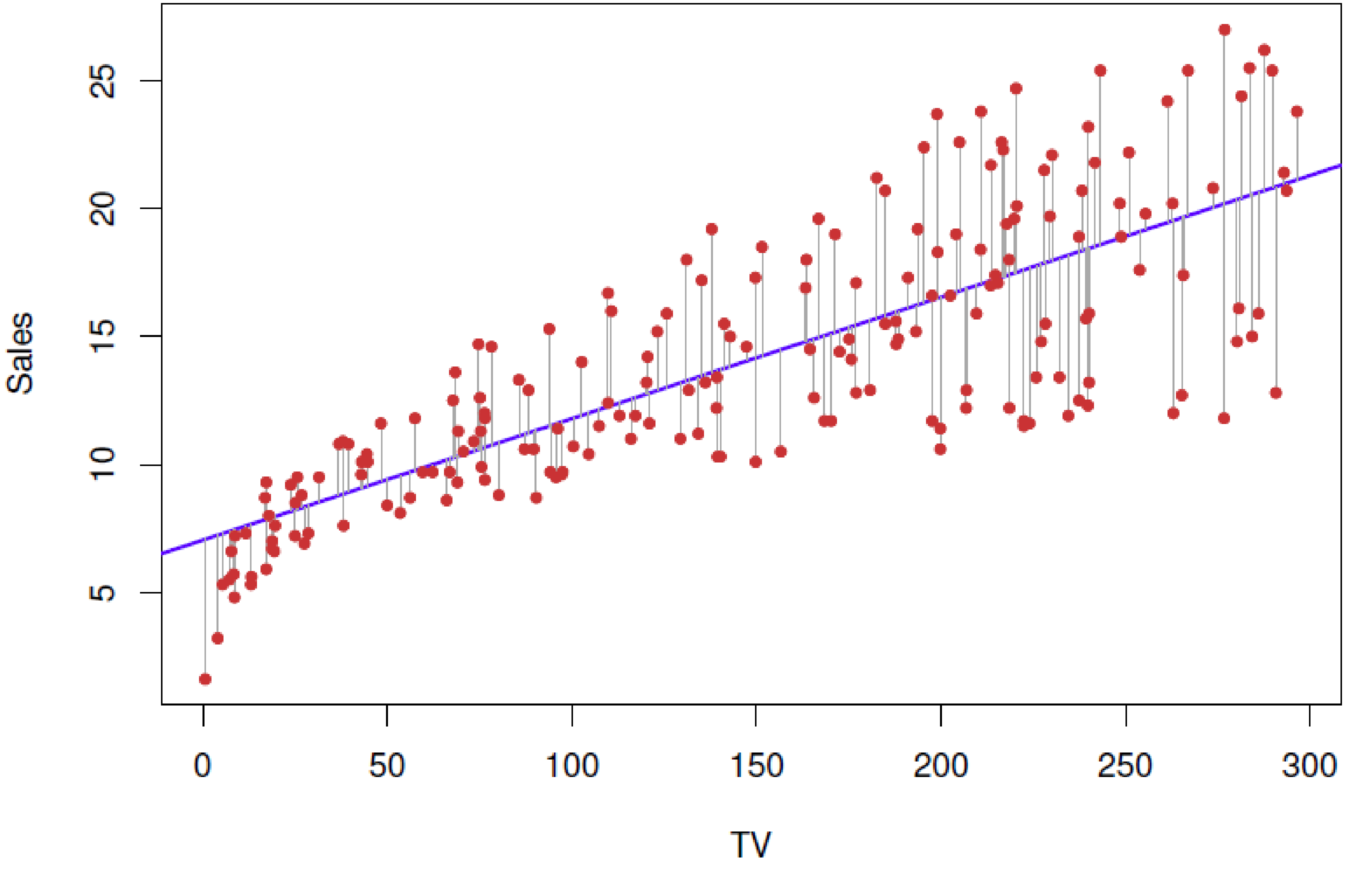
Measuring the Quality of Fit

In order to evaluate the performance of a statistical learning method on a given data set, we need some way to measure how well its predictions actually match the observed data. In the regression setting, the most commonly-used measure is the **mean squared error (MSE),**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

where $\hat{f}(x_i)$ is the prediction that $\hat{f}$ gives for the ith observation.

Estimating the Coefficients



- We define the residual sum of squares (RSS) as

$$\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2,$$

or equivalently as

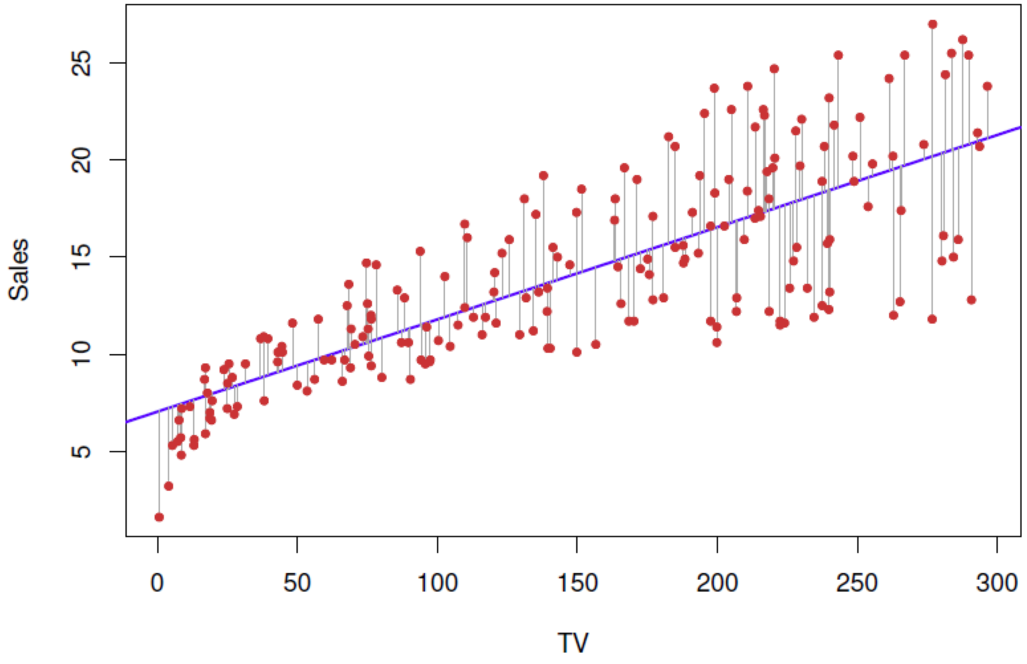
$$\mathrm{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \cdots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2.$$

Estimating the Coefficients

The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the Residual Sum of Squares (RSS). Using some calculus, one can show that the minimizers are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

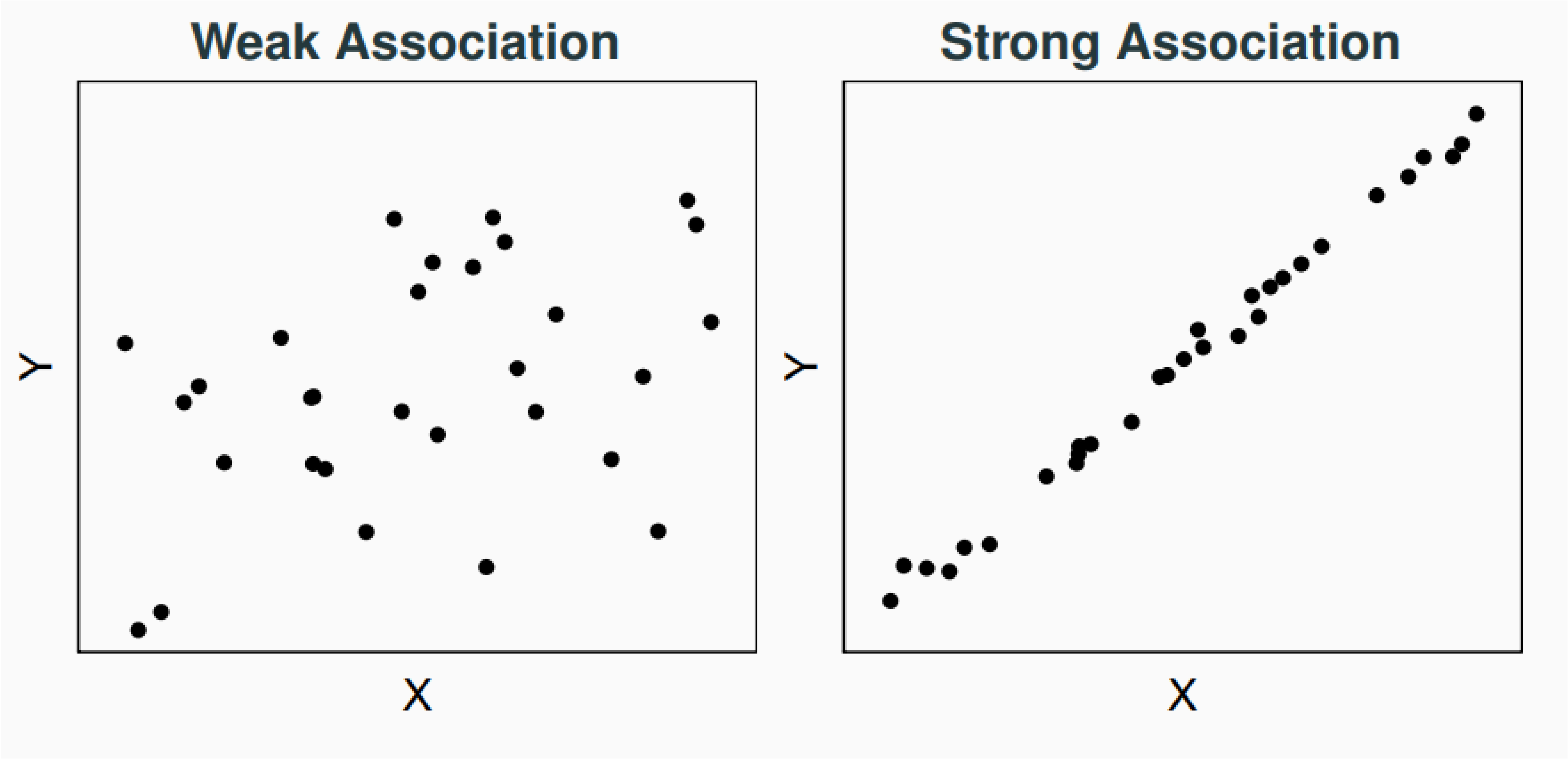
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$



where $\bar{y} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} \equiv \frac{1}{n}\sum_{i=1}^{n} x_i$ are the sample means.

Correlation

Correlation

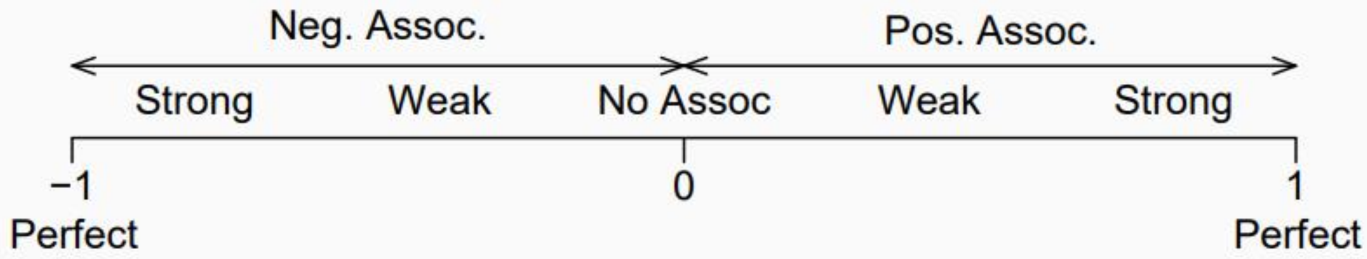


Credit: Yibi Huang - https://www.stat.uchicago.edu/~yibi/teaching/stat220/17aut/Lectures/L22.pdf

Correlation r is a numerical measure of the *direction* and *strength* of the **linear** relationship between two numerical variables.

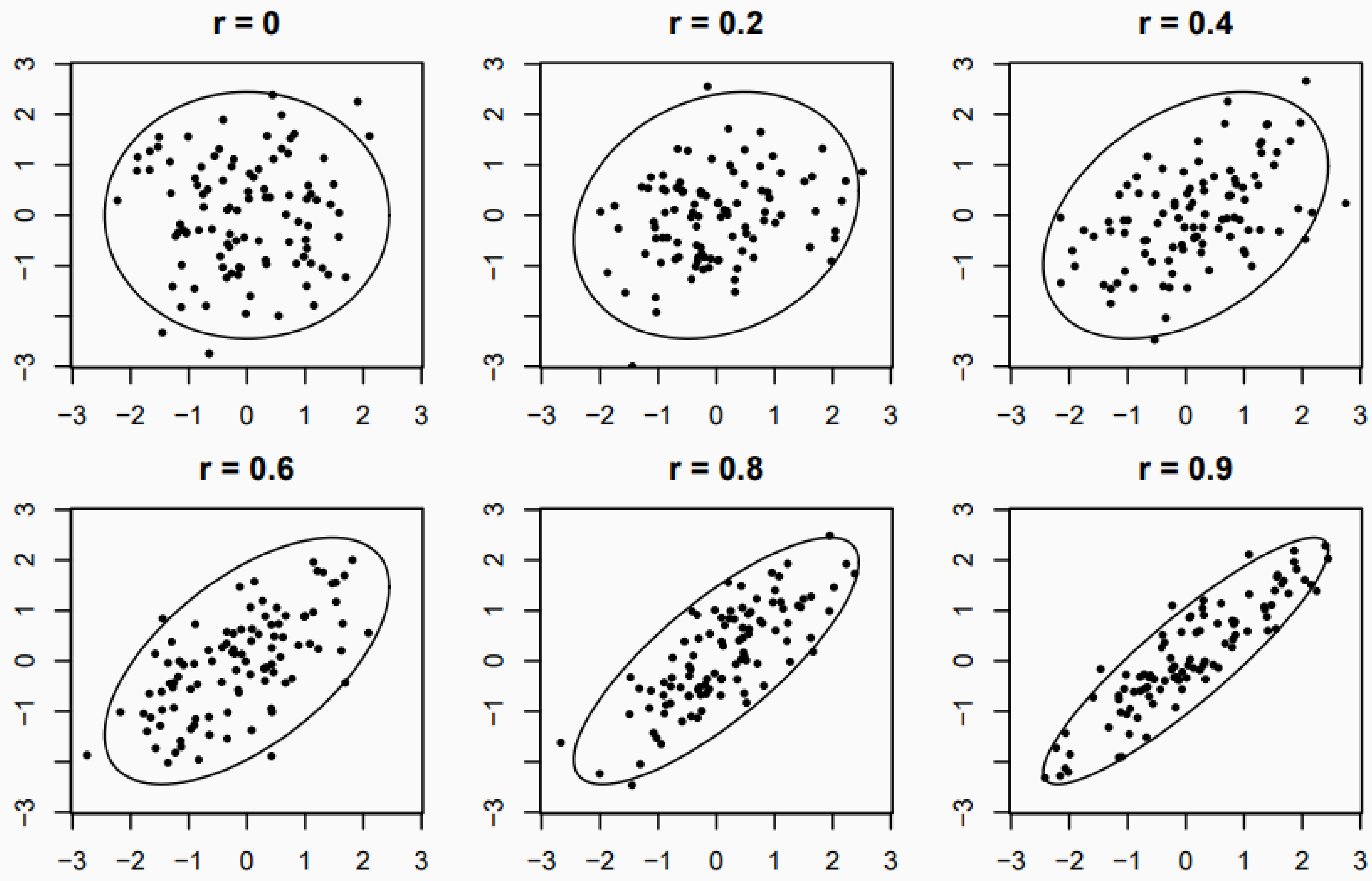
"r" always lies between -1 and 1; the strength increases as you move away from 0 to either -1 or 1.

- $r > 0$: positive association
- $r < 0$: negative association
- $r \approx 0$: very weak linear relationship
- large $|r|$: strong linear relationship
- $r = -1$ or $r = 1$: *only* when all the data points on the scatterplot lie exactly along a *straight line*



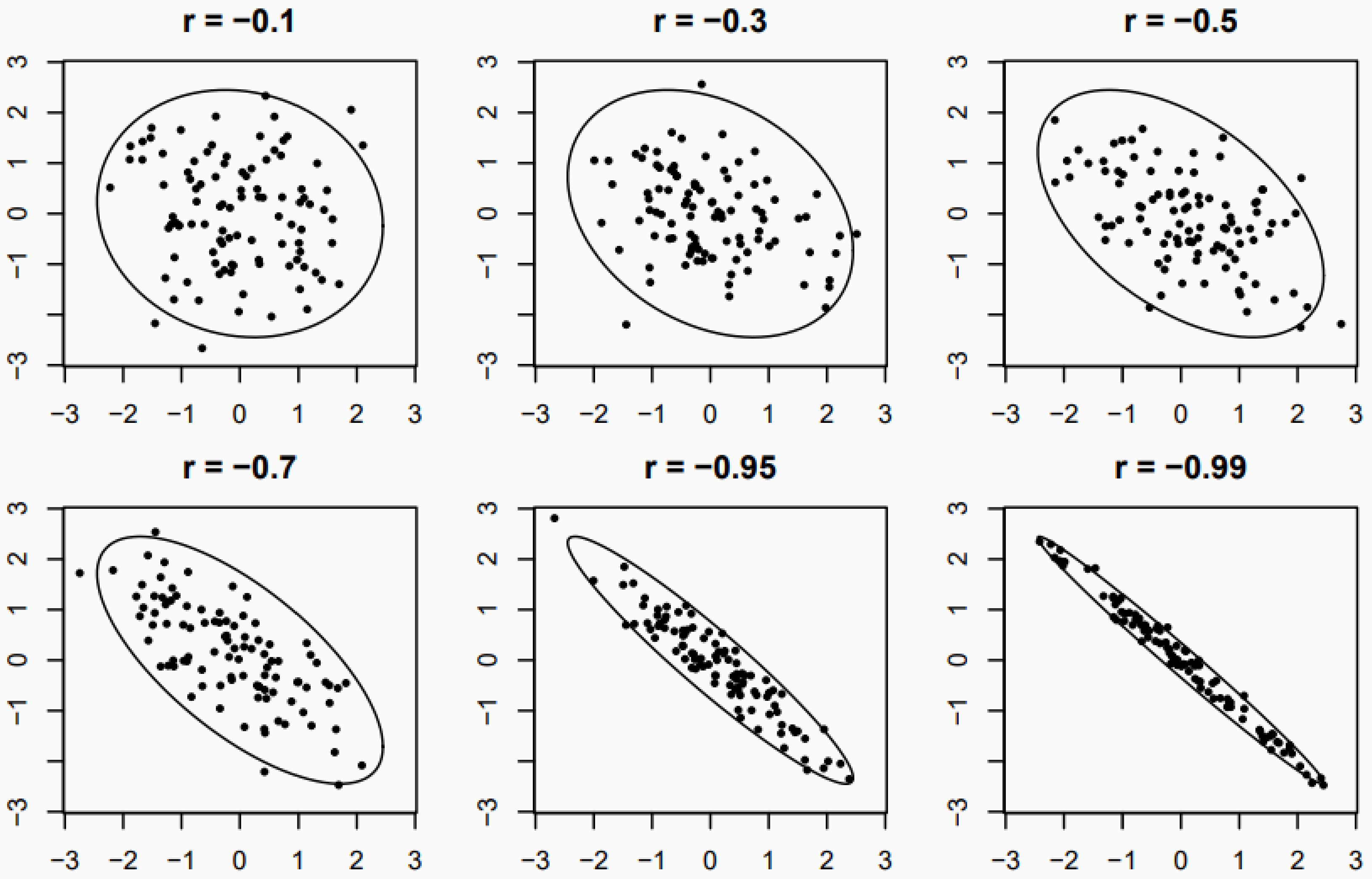
Credit: Yibi Huang - https://www.stat.uchicago.edu/~yibi/teaching/stat220/17aut/Lectures/L22.pdf

Positive Correlations



Credit: Yibi Huang - https://www.stat.uchicago.edu/~yibi/teaching/stat220/17aut/Lectures/L22.pdf

Negative Correlations



Credit: Yibi Huang - https://www.stat.uchicago.edu/~yibi/teaching/stat220/17aut/Lectures/L22.pdf

Formula for Computing the Correlation Coefficient "r"

(x_1, y_1)

(x_2, y_2)

(x_3, y_3)

$\vdots$

(x_n, y_n)

The **correlation coefficient** r
(or simply, **correlation**) is defined as:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \underbrace{\left(\frac{x_i - \bar{x}}{s_x}\right)}_{z\text{-score of } x_i} \underbrace{\left(\frac{y_i - \bar{y}}{s_y}\right)}_{z\text{-score of } y_i} .$$

where s_x and s_y are respectively the sample SD of X and of Y.

Usually, we find the correlation using softwares rather than by manual computation.

Credit: Yibi Huang - https://www.stat.uchicago.edu/~yibi/teaching/stat220/17aut/Lectures/L22.pdf

Fit model using R

```
library(ggplot2)

# Create the dataset
experience <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
salary <- c(45000, 50000, 60000, 65000, 70000, 75000, 80000, 85000, 90000,
df <- data.frame(experience, salary)

# Fit the linear model
model <- lm(salary ~ experience, data = df)

# View the model summary
summary(model)

# Plot the data and fitted model
ggplot(df, aes(x = experience, y = salary)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Years of Experience", y = "Salary (USD)")
```

Fit model using Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm

# Create the dataset
experience = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
salary = np.array([45000, 50000, 60000, 65000, 70000, 75000, 80000, 85000, 9
df = pd.DataFrame({'experience': experience, 'salary': salary})

# Fit the linear model
model = sm.OLS(df['salary'], sm.add_constant(df['experience'])).fit()

# View the model summary
print(model.summary())

# Plot the data and fitted model
sns.regplot(x='experience', y='salary', data=df)
plt.xlabel('Years of Experience')
plt.ylabel('Salary (USD)')
plt.show()

# Predict salary for a new value of experience
new_experience = 11
predicted_salary = model.predict(sm.add_constant(pd.Series(new_experience)))

print(predicted_salary)
```

References

- An introduction to statistical learning with applications in Rby Gareth James, Daniela Witten, TrevorHastie, and Robert Tibshirani, New York, Springer Science and Business Media, 2013, eISBN:978-1-4614-7137-7