# Machine Learning - Regression

Week 8 – Part 1 – Introduction to Machine Learning

CS 457 - L1 Data Science

Zeehasham Rasheed

# Outline

- Define and describe predictive analysis.
- Compare and contrast descriptive and predictive analysis.
- Define and describe the regression process.
- Define and describe regression techniques.

# Data Mining

- Data mining is the process of analyzing data to discover patterns and relationships. If you are working with sales data, it makes sense to determine facts such as:
  - Which customer purchased the most products?
  - Which customer purchased the least?
  - What was the average sales per customer order?
  - What was the average number of days between orders?
  - What customers have not yet ordered the new product, and so on.

- Using data to **describe past events is called descriptive analytics**. Being able to analyze data to determine such historical facts is important and valuable to businesses. To perform descriptive analytics, you can use statistical tools to generate metrics, you can use visualization tools to chart data, you can use clustering to group data, and more.

# Machine Learning - Predictive Analytics

- In Machine Learning, Predictive Analysis is about using data to predict future events.
- Companies use predictive analytics for a wide range of applications:
  - Estimate the revenue opportunity associated with an upcoming product sale.
  - Predict the length of time machines that will run without failing.
  - Determine the loan amount to offer a customer.
  - Determine which customers are likely to become long-term customers.
  - Estimate for what price a beach house in Karachi will be sold.
  - And more.

# Machine Learning – Types of Learning

- **Supervised:** We are given input samples (X) and output samples (y) of a function y = f(X). We would like to "learn" a model **f**, and evaluate it on new data (recall Employee Data – Attrition column)
  - **Classification:** y is discrete (class labels).
  - **Regression:** y is continuous, e.g. (sales price, weather temperature forecast)

- **Unsupervised:** Given only samples X of the data (no output y), we compute a function **f** such that y = f(X) gives us some grouping or segments
  - **Clustering:** y is discrete (cluster labels or segment numbers)

# Supervised and Unsupervised Use Cases

- **Supervised:**
  - Is this image a cat, dog, car, house?
  - How would this user score that restaurant?
  - Is this email spam?
  - Is this application qualifies for loan approval?

- **Unsupervised**
  - Group similar images.
  - What are the top 20 topics in Twitter right now?
  - Find similar customer segments for marketing and advertisement.

# ML Techniques (Algorithms)

- **Supervised Learning:**
  - Linear Regression
  - Logistic Regression
  - Decision Trees
  - Random Forests
  - Naïve Bayes
  - Support Vector Machines
  - kNN (k Nearest Neighbors)

- **Unsupervised Learning:**
  - Clustering
  - Factor analysis
  - Topic Models

# End of Part 1

# Machine Learning - Regression

Week 8 – Part 2 – Introduction to Regression

CS 457 - L1 Data Science

Zeehasham Rasheed

# Regression Basics

- **Simple Linear Regression**:  Characterizing relationships between two variables

- **Multiple Linear Regression:** Characterizing relationships between more than two input variables

  - We are going to create a model for **<u>predicting continuous (numeric)</u>** values using a technique called **Regression**
    - such as company's projected revenue, average basketball player's height, and range of temperatures in Phoenix.

  - The goal of Regression is to produce a model (mathematical equation), which we can use to **<u>predict unknown or future values.</u>**

  - Regression uses **Correlation** and **p-Values**

# Correlation and Regression

**Correlation Analysis**: Concerned with measuring **the strength and direction of the association between variables**. The correlation of X and Y (Y and X).

**Linear Regression**: Concerned with **predicting the value of one variable based on (given) the value of the other variable**.

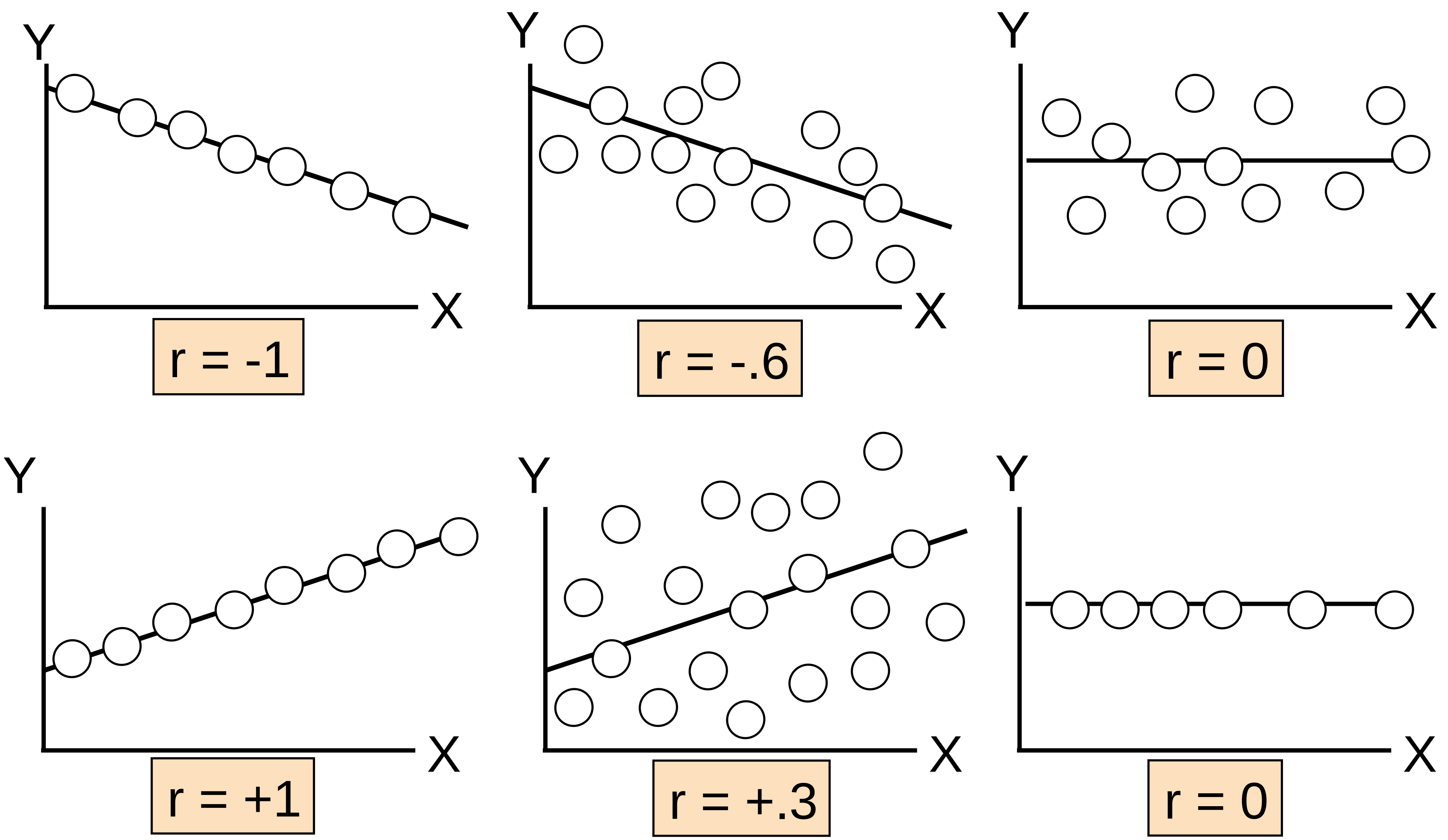
# Correlation

- Correlation measures the relative strength of the **linear relationship** between two variables
  - Unit-less
  - Ranges between –1 and 1
  - The closer to –1, the stronger the negative linear relationship
  - The closer to 1, the stronger the positive linear relationship
  - The closer to 0, the weaker or no positive/negative linear relationship

# Correlation Analysis

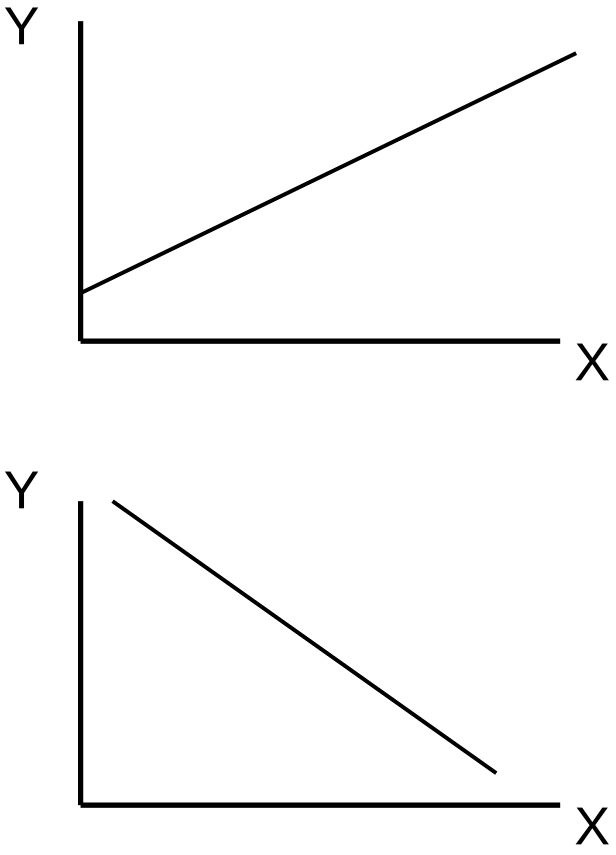
- Characterizes the **extent of linear relationship** between two variables, and the **direction**
  - **How closely does a scatterplot of the two variables produce a non-flat (with some angle and slope) straight line?**
  - **Does one variable tend to increase as the other increases ($r>0$), or decrease as the other increases ($r<0$)**
    - **$r$ is the correlation coefficient**

# Scatter Plots of Data with Various Correlation Coefficients

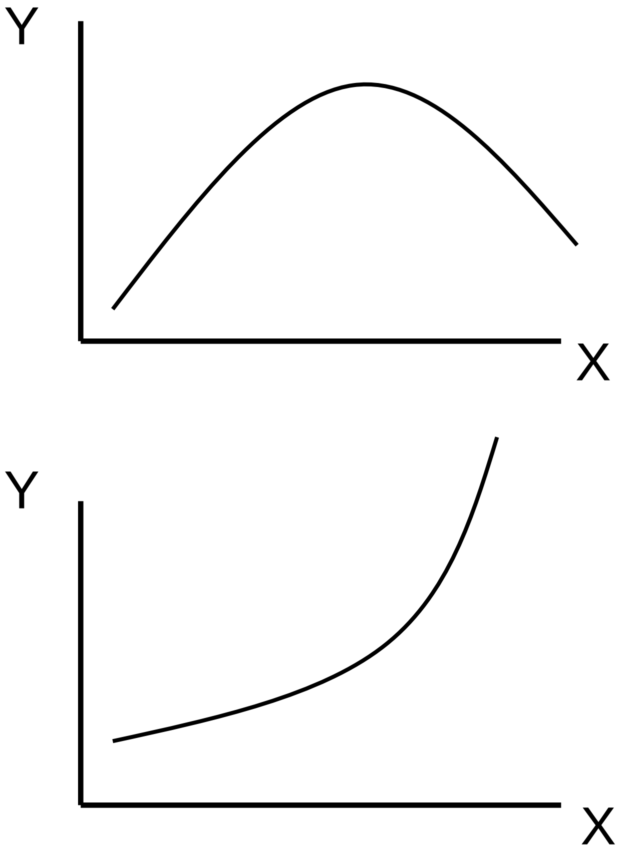


# Linear Correlation

Linear relationships



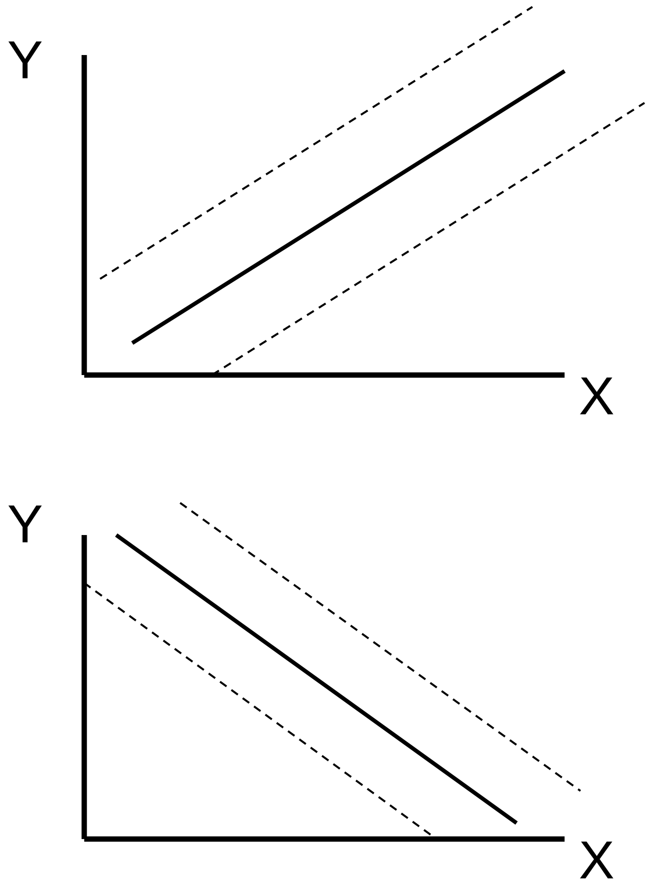

Curvilinear relationships



# Strong and Weak Relationship

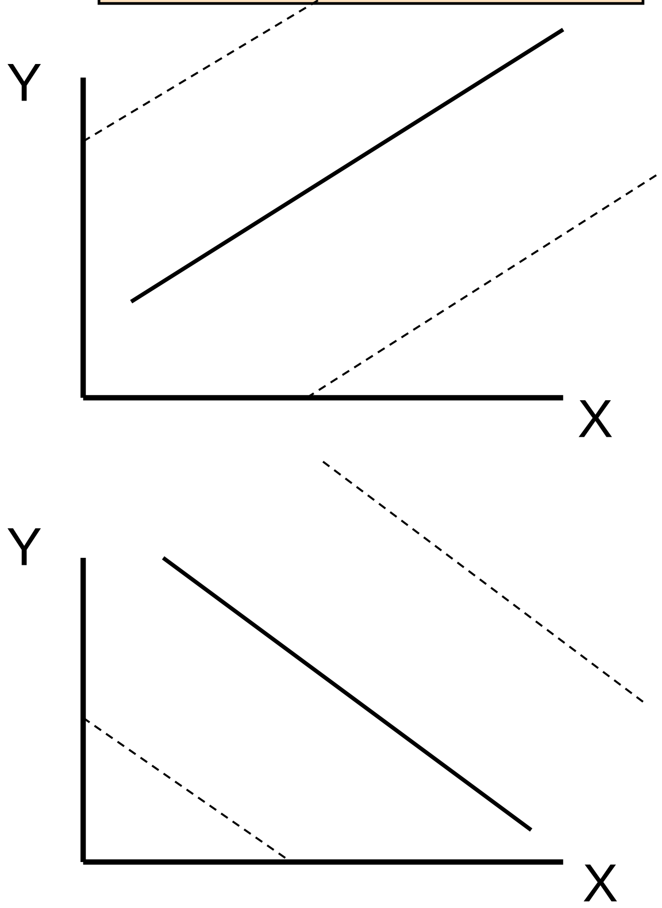
- Distance between points is small

Strong relationships



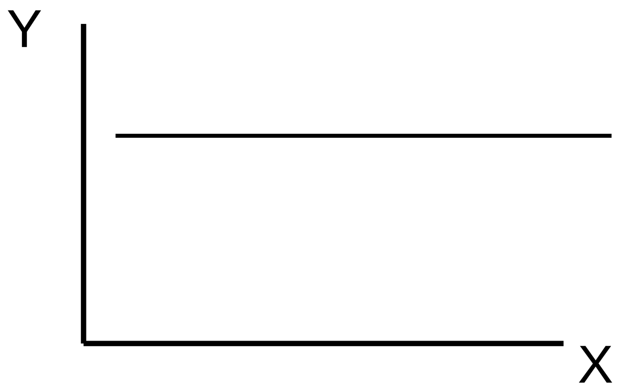

- Distance between points is large

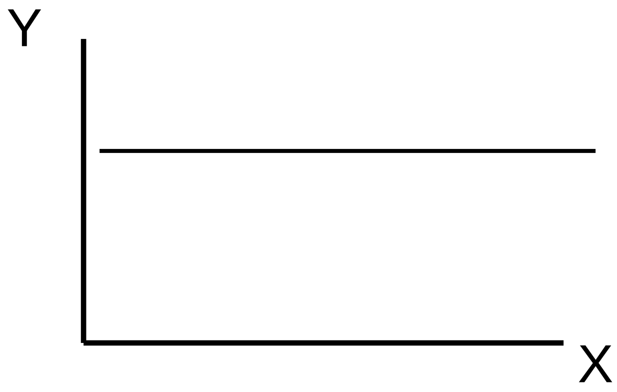
Weak relationships



# No Relationship

No relationship



# Recall: Interpreting Covariance

- Interpreting Covariance

- cov(X,Y) > 0 X and Y are positively correlated
- cov(X,Y) < 0 X and Y are negatively/inversely correlated
- cov(X,Y) = 0 X and Y are independent

$$\mathrm{cov}(x, y) = \frac{\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{n-1}$$

# Variance

- Variance measures how far a data set is spread out.
- It is mathematically defined as the average of the squared differences from the mean.

$$\text{Sample Variance} = s^2 = \frac{\Sigma(X - \overline{X})^2}{n - 1}$$

# Correlation Coefficient

- Pearson's Correlation Coefficient is standardized covariance (unitless):

$$r = \frac{\operatorname{cov}ariance(x, y)}{\sqrt{\operatorname{var} x}\sqrt{\operatorname{var} y}}$$

# Calculation by Hand

$$\hat{r} = \frac{\operatorname{cov}ariance(x,y)}{\sqrt{\operatorname{var} x}\sqrt{\operatorname{var} y}} = \frac{\dfrac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1}}{\sqrt{\dfrac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}\sqrt{\dfrac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{n-1}}}$$

$$\hat{r} = \frac{\dfrac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\cancel{n-1}}}{\sqrt{\dfrac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{\cancel{n-1}}}\sqrt{\dfrac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{\cancel{n-1}}}} =$$

$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} = \frac{SS_{xy}}{\sqrt{SS_x SS_y}}$$

Numerator of covariance

$$\hat{r} = \frac{SS_{xy}}{\sqrt{SS_x SS_y}}$$

Numerators of variance

# Independent vs Dependent Variables

- **In Correlation**
  - the two variables are treated as equals.

- **In Regression**
  - one variable is considered independent variable X (=predictor) and
  - the other variable is the the dependent variable *Y* (=outcome)

# Linear Regression – Use Cases

- How precisely can we predict Y given X?

- **Can we predict**
  - Total Lung Capacity (TLC) <u>given</u> height (m)
    - Do people of taller height tend to have a larger total lung capacity?
  - Predict Cognitive Function Score with Vitamin D level
  - Predict performance rating <u>given</u> aptitude test score given prior to hiring
  - Predict stock price <u>given</u> open/close price.
  - Predict sales based on product demand

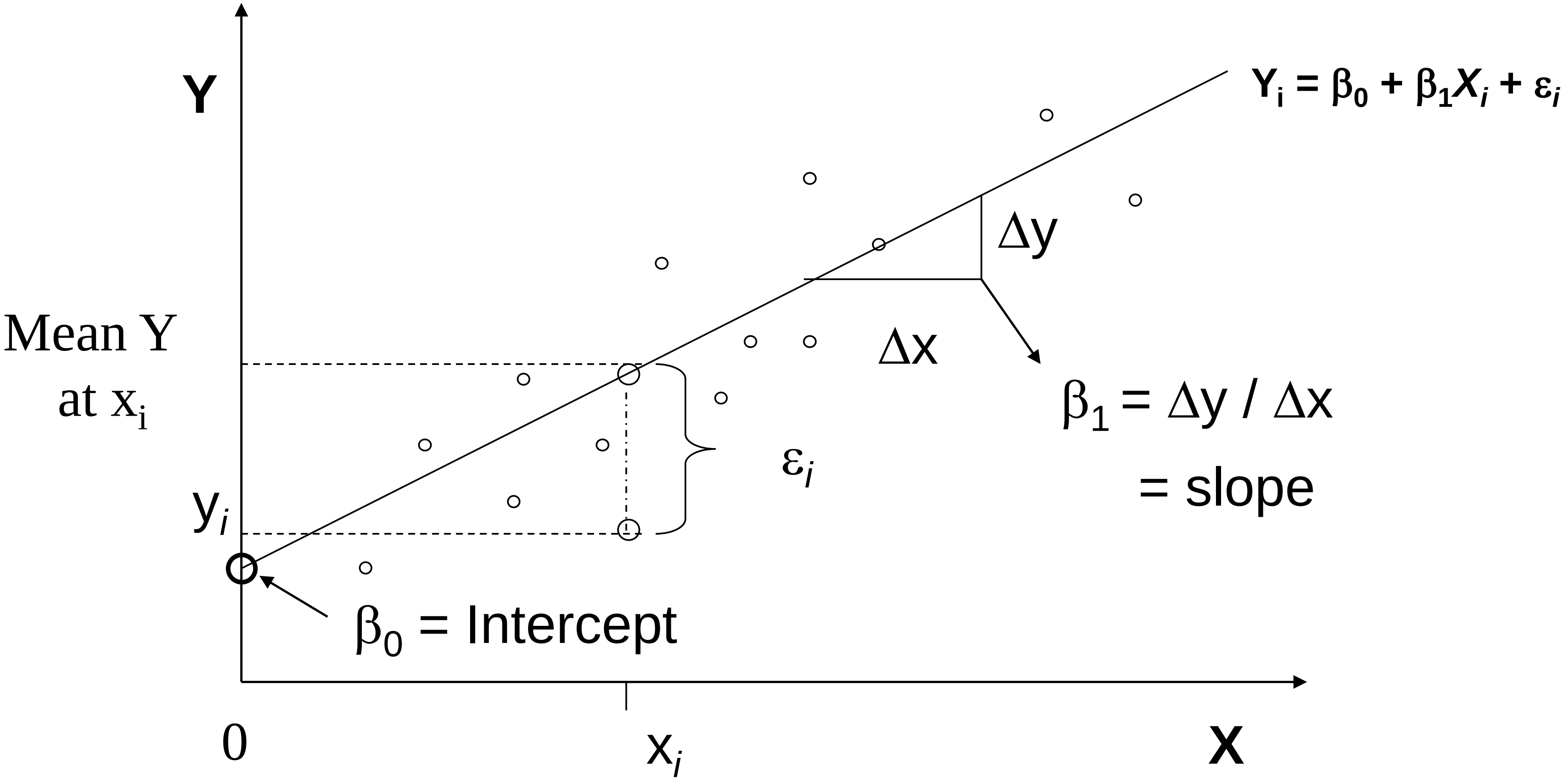
# Linear Regression –Terminology

- Outcome, Y
  - Dependent variable
  - Response variable

- Explanatory variable / Input variable / Predictor, X
  - Independent variable
  - Covariate

  - What is the relationship between Y and X?
    - Regression “models” this as a **line**
    - We care about “slope” size and direction
    - Slope=0 corresponds to “no association”

# Linear Regression - Relationship




Model:

$$\mathbf{Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i}$$

# Assumptions

- **Regression** is a statistical procedure that determines the equation for the straight line that best fits a specific set of data.
  - Any straight line can be represented by an equation of the form $\mathbf{Y_i = \beta_0 + \beta_1} \boldsymbol{X_i}$ , where $\mathbf{Y_i = \beta_0}$ and $\boldsymbol{\beta_1}$ are constants.
  - The value of $\boldsymbol{\beta_1}$ is called the slope constant and determines the direction and degree to which the line is tilted.
  - The value of $\boldsymbol{\beta_0}$ is called the Y-intercept and determines the point where the line crosses the Y-axis.

- Linear regression assumes that
  - The relationship between X and Y is linear
  - The observations are independent

# Linear Regression - Relationship

## In words

- Intercept $\beta_0$ is mean Y at X=0
- Slope $\beta_1$ is change in mean Y per 1 unit difference in X

## **Inference**: We develop best guesses at $\beta_0, \beta_1$ using our data

- Step 1: Find the **"least squares"** line
  - Tracks through the middle of the data "as best possible"
  - Has intercept $b_0$ and slope $b_1$ that make sum of $[Y_i - (b_0 + b_1 X_i)]^2$ smallest (smallest difference or error)
- Step 2: Use the slope and intercept of the least squares line **to predict unknown/future values**

# The Model

- The first order linear model

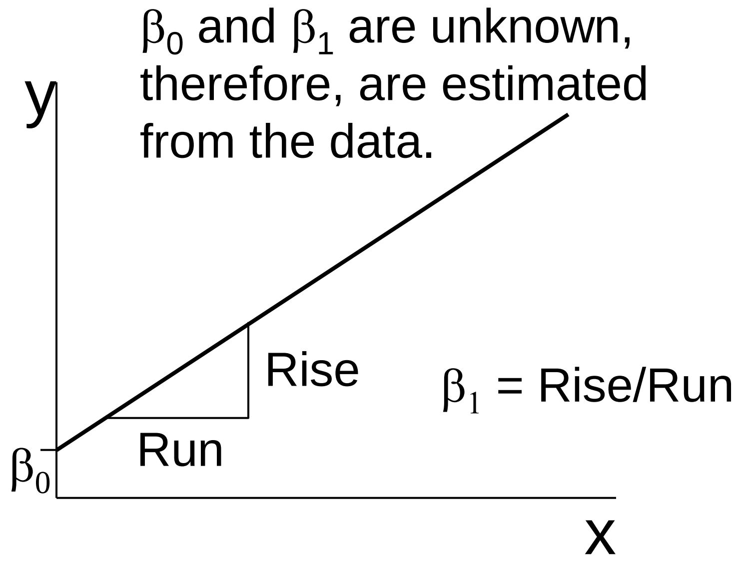
  y = dependent variable

  x = independent variable

  $\beta_0$ = y-intercept

  $\beta_1$ = slope of the line

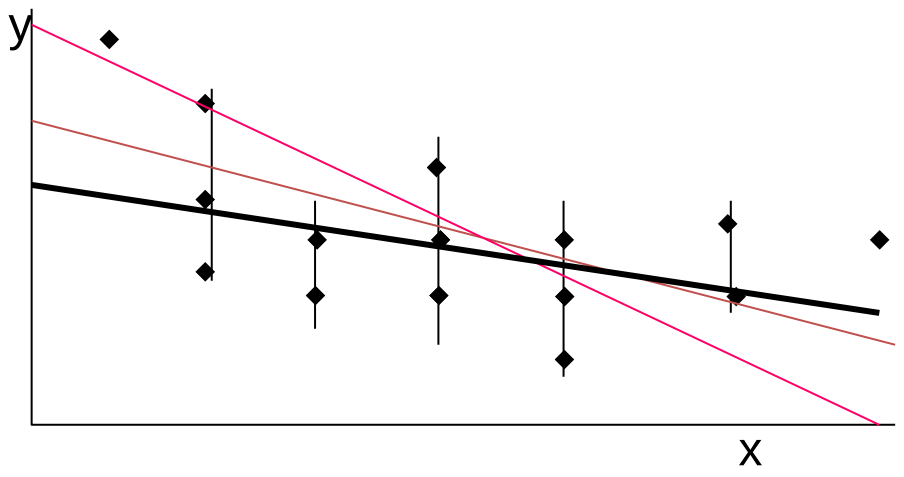
- $\beta_0$ , $\beta_1$ are also called coefficients




$$y = \beta_0 + \beta_1 x$$

# Estimating the Coefficients

- The estimates are determined by
  - drawing a sample from the population of interest,
  - calculating sample statistics.
  - producing a straight line that cuts into the data.



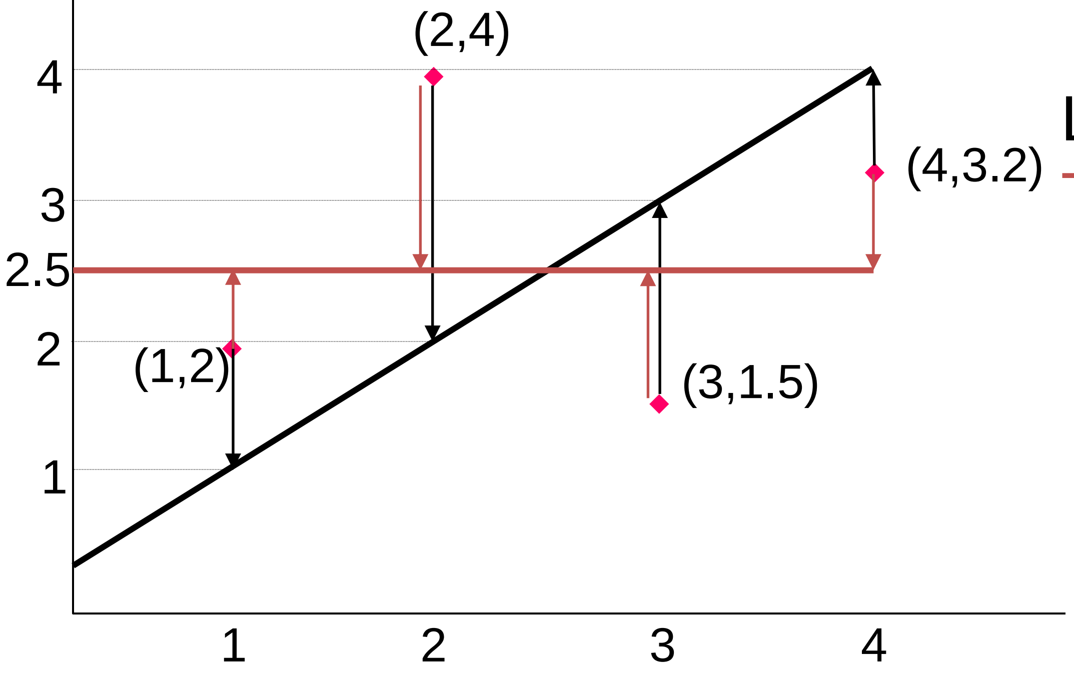

The question is:
Which straight line fits best?

# Best Line Example

The best line is the one that minimizes the sum of squared vertical differences between the points and the line.

**Sum of squared differences = $(2 - 1)^2 + (4 - 2)^2 + (1.5 - 3)^2 + (3.2 - 4)^2 = 6.89$**

**Sum of squared differences = $(2 - 2.5)^2 + (4 - 2.5)^2 + (1.5 - 2.5)^2 + (3.2 - 2.5)^2 = 3.99$**



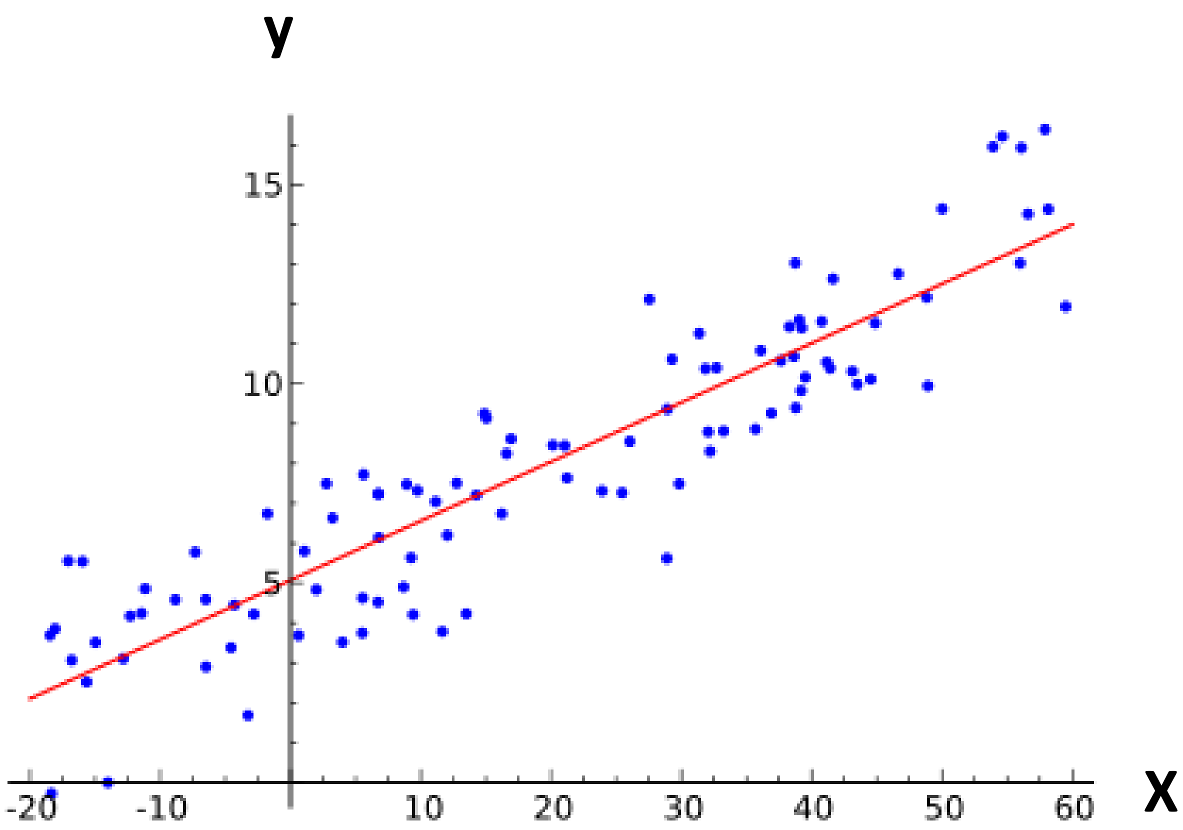

Let us compare two lines

The second line is horizontal

The smaller the sum of squared differences the better the fit of the line to the data.

# Best Line

We want to find the “best” line (linear function y=f(X)) to explain the data.



# Calculating Slope and Intercept

- The first order linear model

  y = dependent variable

  x = independent variable

  $\beta_0$ = y-intercept

  $\beta_1$ = slope of the line

To calculate the estimates of the coefficients that minimize the differences between the data points and the line, use the formulas:

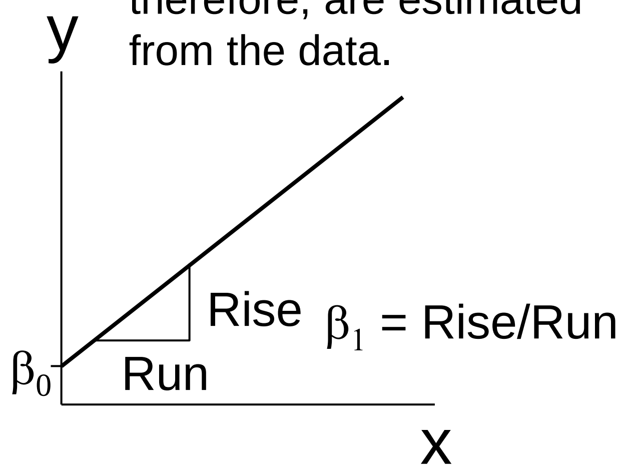
$$\beta_1 = \frac{\mathrm{cov}(X,Y)}{s_x^2}$$

$$\beta_0 = \bar{y} - b_1\bar{x}$$

The regression equation that estimates the equation of the first order linear model is:

$$\hat{y} = \beta_0 + \beta_1 x$$

$\beta_0$ and $\beta_1$ are unknown, therefore, are estimated from the data.




$$y = \beta_0 + \beta_1 x$$

# End of Part 2

# Machine Learning - Regression

Week 8 – Part 3 – Regression Examples

CS 457 - L1 Data Science

Zeehasham Rasheed

# Example: Relationship between odometer reading and a used car's selling price.

- A car dealer wants to find  the relationship between  the odometer reading and  the selling price of used cars.
- A random sample of 100 cars is selected, and the data  recorded.
- **Find the regression line.**

| Car | Odometer | Price |
|---|---|---|
| 1 | 37388 | 5318 |
| 2 | 44758 | 5061 |
| 3 | 45833 | 5008 |
| 4 | 30862 | 5795 |
| 5 | 31705 | 5784 |
| 6 | 34010 | 5359 |
| . | . | . |
| . | . | . |
| . | . | . |

Independent variable  x

Dependent variable  y

# Solution: Calculation

- To calculate $b_0$ and $b_1$ we need to calculate several statistics first

$$\bar{x} = 36{,}009.45; \qquad s_x^2 = \frac{\sum (x_i - x)^2}{n-1} = 43{,}528{,}688$$

$$\bar{y} = 5{,}411.41; \qquad \text{cov}(X,Y) = \frac{\sum (x_i - x)(y_i - y)}{n-1} = -1{,}356{,}256$$

where n = 100.

$$b_1 = \frac{\text{cov}(X,Y)}{s_x^2} = \frac{-1{,}356{,}256}{43{,}528{,}688} = -.0312$$

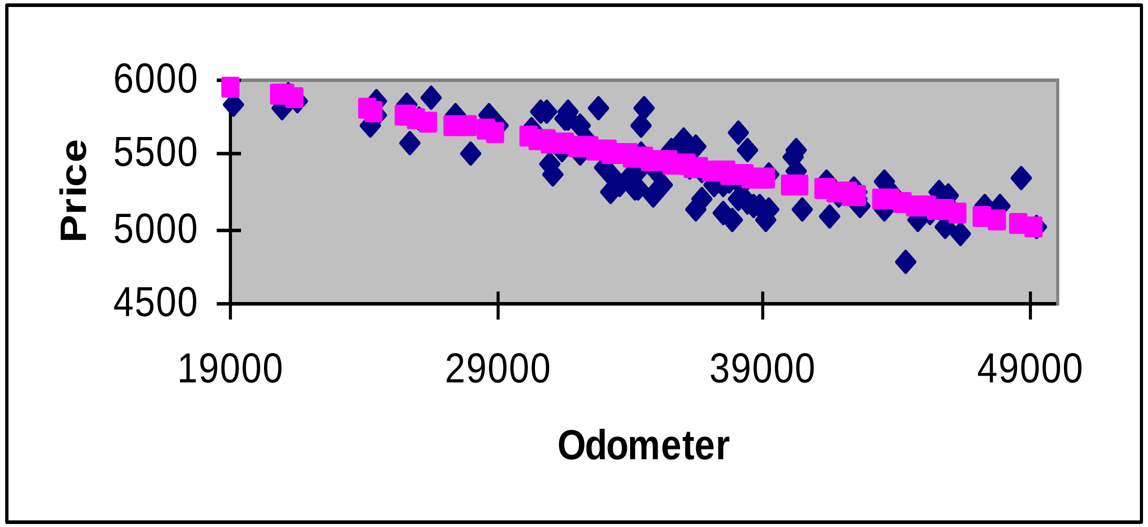
$$b_0 = \bar{y} - b_1\bar{x} = 5411.41 - (-.0312)(36{,}009.45) = 6{,}533$$

$$\boxed{\hat{y} = b_0 + b_1 x = 6{,}533 - .0312x}$$

# Output - Coefficients

SUMMARY OUTPUT

| Regression Statistics | |
|---|---|
| Multiple R | 0.806308 |
| R Square | 0.650132 |
| Adjusted F | 0.646562 |
| Standard E | 151.5688 |
| Observatio | 100 |




$$\hat{y} = 6{,}533 - .0312x$$

ANOVA

| | df | SS | MS | F | Significance F |
|---|---|---|---|---|---|
| Regressior | 1 | 4183528 | 4183528 | 182.1056 | 4.4435E-24 |
| Residual | 98 | 2251362 | 22973.09 | | |
| Total | 99 | 6434890 | | | |

| | Coefficients | Standard Erro | t Stat | P-value |
|---|---|---|---|---|
| Intercept | 6533.383 | 84.51232 | 77.30687 | 1.22E-89 |
| Odometer | -0.03116 | 0.002309 | -13.4947 | 4.44E-24 |

# Output – P values

The P-value is, as usual, the probability of observing the data under the null hypothesis of no linear relationship.

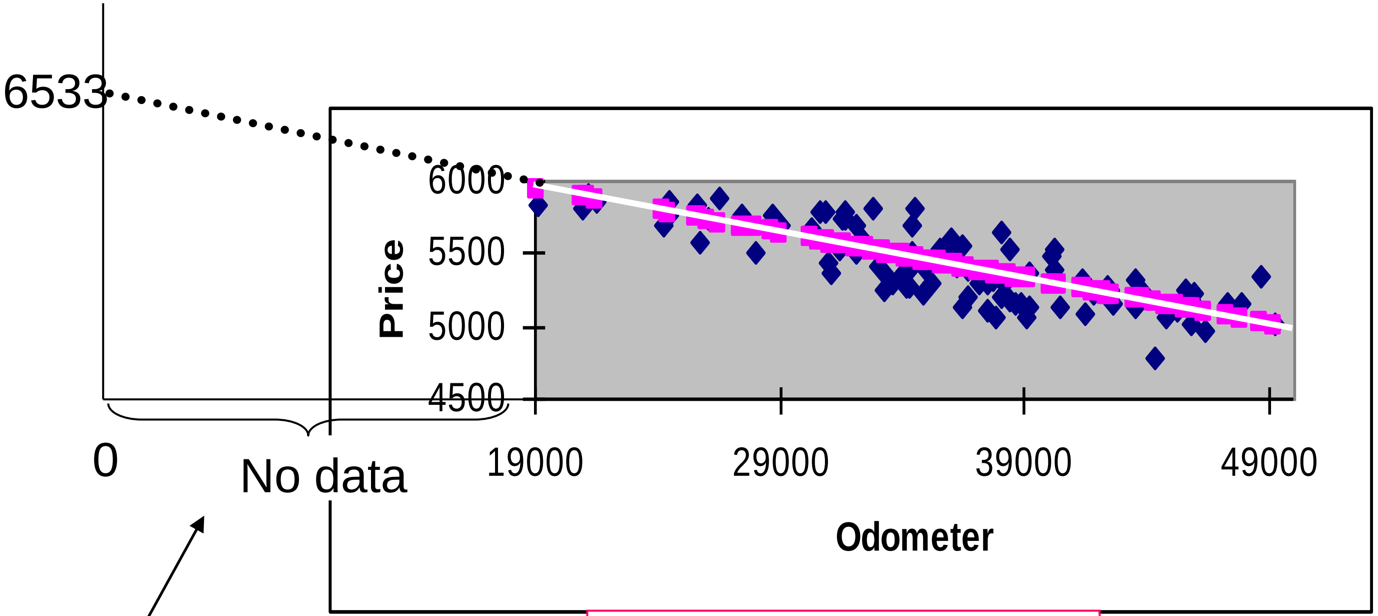
If **p is small**, say less than 0.05, we conclude that **there is a linear relationship**. (smaller the p-value, stronger and important is the relationship)

If **p is large**, say greater than 0.05, we conclude that **there is a no relationship**

| | Coefficients | Standard Erro | t Stat | P-value |
|---|---|---|---|---|
| Intercept | 6533.383 | 84.51232 | 77.30687 | 1.22E-89 |
| Odometer | -0.03116 | 0.002309 | -13.4947 | 4.44E-24 |

This shows Odometer has a strong relationship with selling price of the car based on p-value

# Result Interpretation



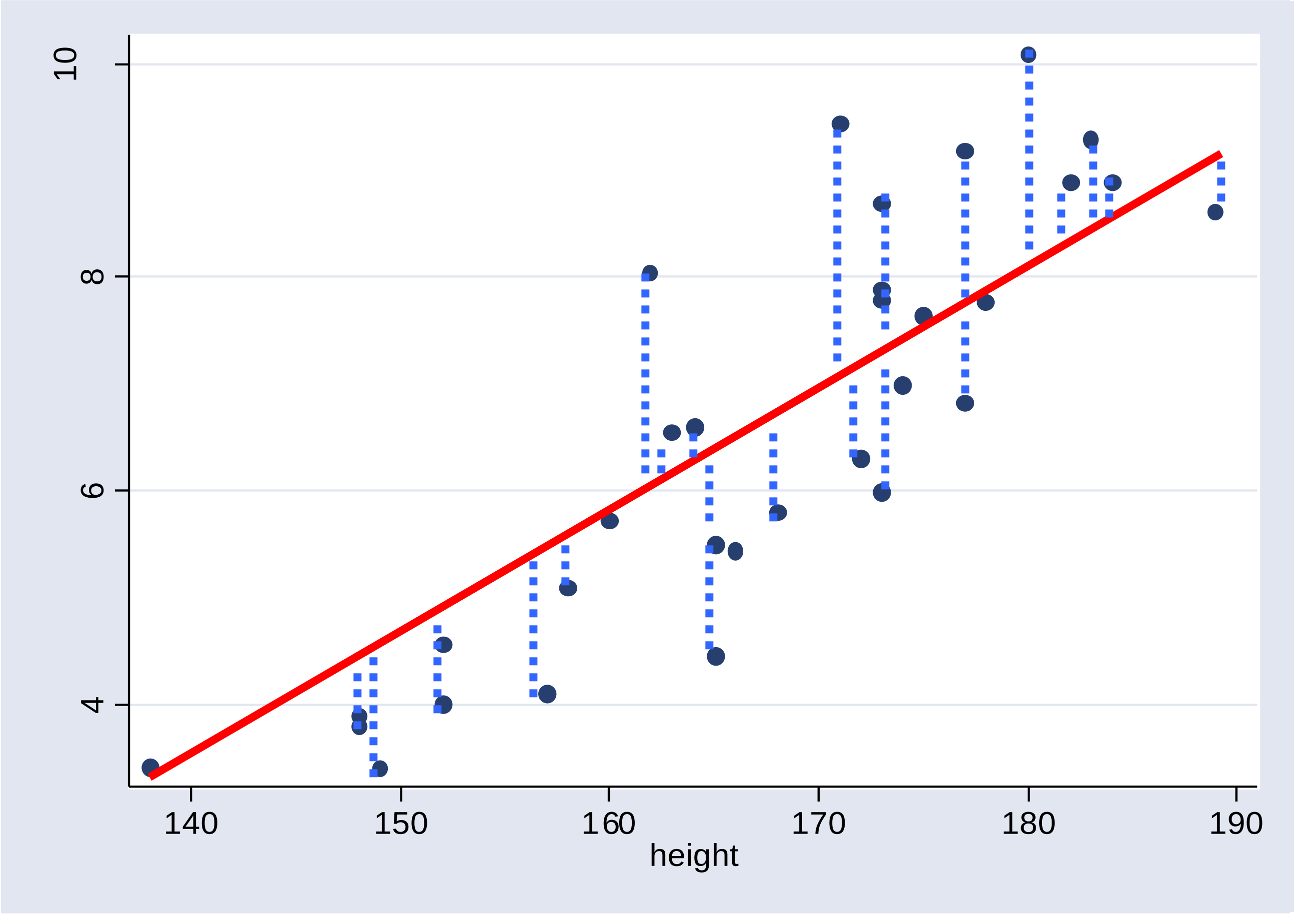

$$\hat{y} = 6{,}533 - .0312x$$

The intercept is $b_0 = 6533$.

This is the slope of the line.
For each additional mile on the odometer,
the price decreases by an average of \$0.0312

Do not interpret the intercept as the "Price of cars that have not been driven"

# Lung Capacity Example for Regression



# Lung Capacity Data - Intercept

In STATA - "regress" command:

Syntax "regress yvar xvar"

```
. regress tlc height

      Source |       SS       df       MS              Number of obs =      32
-------------+------------------------------           F(  1,    30) =   89.12
       Model |  93.7825029     1  93.7825029           Prob > F      =  0.0000
    Residual |  31.5694921    30   1.0523164           R-squared     =  0.7482
-------------+------------------------------           Adj R-squared =  0.7398
       Total |  125.351995    31  4.04361274           Root MSE      =  1.0258

------------------------------------------------------------------------------
         tlc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   .1417377    .015014     9.44   0.000     .1110749    .1724004
       _cons |  -17.10484   2.516234    -6.80   0.000    -22.24367     -11.966
------------------------------------------------------------------------------
```

$b_0$

TLC of -17.1 liters among persons of height = 0

# Lung Capacity Data - Coefficients

In STATA - "regress" command:

Syntax "regress yvar xvar"

```
. regress tlc height

      Source |       SS       df       MS              Number of obs =      32
-------------+------------------------------           F(  1,    30) =   89.12
       Model |  93.7825029     1  93.7825029           Prob > F      =  0.0000
    Residual |  31.5694921    30   1.0523164           R-squared     =  0.7482
-------------+------------------------------           Adj R-squared =  0.7398
       Total |  125.351995    31  4.04361274           Root MSE      =  1.0258

------------------------------------------------------------------------------
         tlc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   .1417377    .015014     9.44   0.000     .1110749    .1724004
       _cons |  -17.10484   2.516234    -6.80   0.000    -22.24367     -11.966
------------------------------------------------------------------------------
```

$b_1$

On average, TLC increases by 0.142 liters per cm for every one cm increase in height.

# Lung Capacity Data – p-Value

```
. regress tlc height

      Source |       SS       df       MS              Number of obs =      32
-------------+------------------------------           F(  1,    30) =   89.12
       Model |  93.7825029     1  93.7825029           Prob > F      =  0.0000
    Residual |  31.5694921    30   1.0523164           R-squared     =  0.7482
-------------+------------------------------           Adj R-squared =  0.7398
       Total |  125.351995    31  4.04361274           Root MSE      =  1.0258

------------------------------------------------------------------------------
         tlc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   .1417377    .015014     9.44   0.000     .1110749    .1724004
       _cons |  -17.10484   2.516234    -6.80   0.000    -22.24367     -11.966
------------------------------------------------------------------------------
```

pvalue for the slope smaller that 0.05 significance level

**We reject the null hypothesis** of 0 slope (null hypothesis says there is no linear relationship). We conclude that **data support a strong relationship and tendency for TLC to increase with height.**

# Lung Capacity Data – Confidence Intervals

```
. regress tlc height

      Source |       SS       df       MS              Number of obs =      32
-------------+------------------------------           F(  1,    30) =   89.12
       Model |  93.7825029     1  93.7825029           Prob > F      =  0.0000
    Residual |  31.5694921    30   1.0523164           R-squared     =  0.7482
-------------+------------------------------           Adj R-squared =  0.7398
       Total |  125.351995    31  4.04361274           Root MSE      =  1.0258

------------------------------------------------------------------------------
         tlc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   .1417377    .015014     9.44   0.000     .1110749    .1724004
       _cons |  -17.10484   2.516234    -6.80   0.000    -22.24367     -11.966
------------------------------------------------------------------------------
```

We are 95% confident that the interval (0.111, 0.172) includes the true slope.
Data shows an average increase in TLC ranging between 0.111 and 0.172 for every one cm of height
The data support a tendency for TLC to increase with height.

# Linear Regression – Prediction Performance  Evaluation

- Need to evaluate how good is the model prediction.
  - What is the linear **regression prediction of Y given X**?
    - Plug X into the regression equation
    - The prediction $\hat{\mathbf{y}} = b_0 + b_1X$
  - The "residual" ε = data-prediction error = $\hat{\mathbf{y}}$ – Y
  - **Sum of Square Error**
    - This is the sum of differences between points and the regression line.
    - It can serve as a measure of how well the line fits the data.

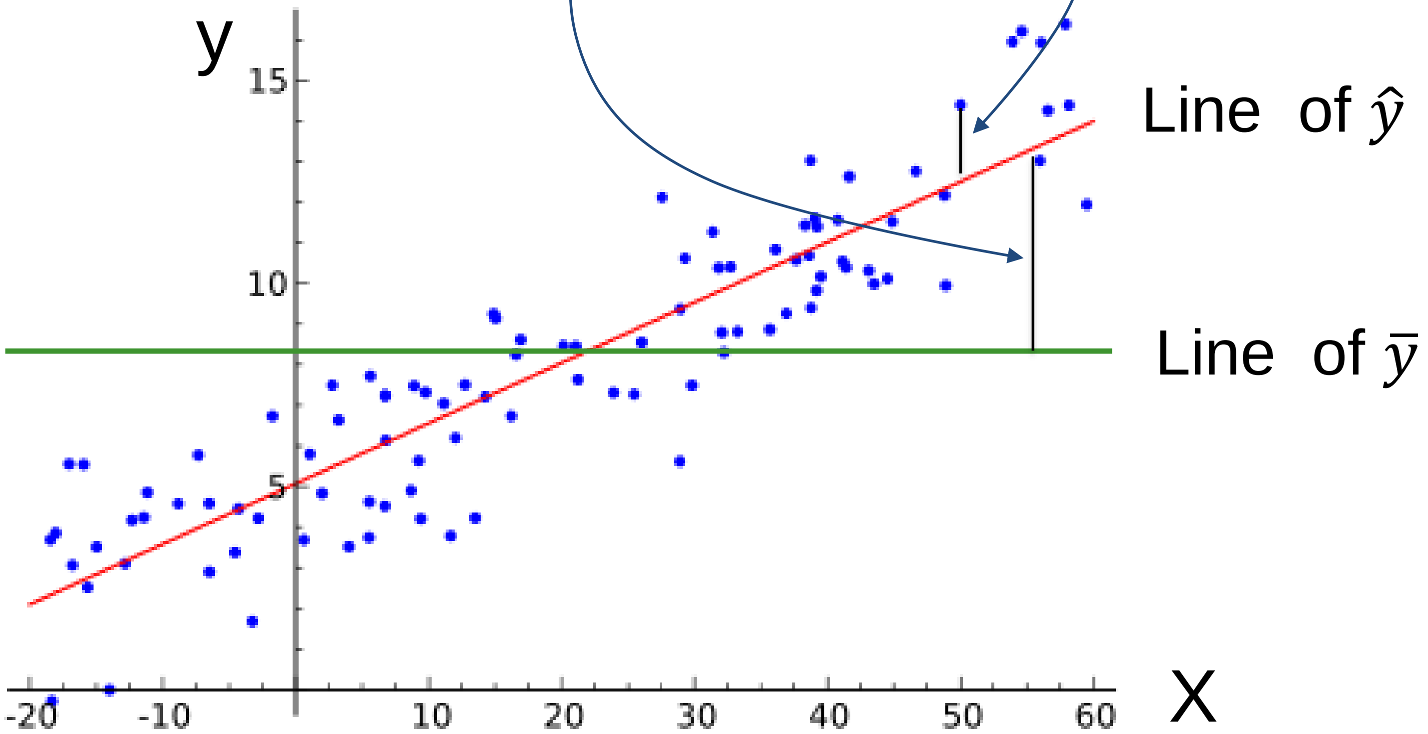
$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

  - Least squares minimizes the sum of squared residuals, e.g. makes predicted $\hat{\mathbf{y}}$ as close to actual Y as possible **(Smaller SSE, Better Prediction Performance)**

# R-squared - Formula

**SSE: Small if good fit**

$$R^2 = 1 - \frac{\boxed{\sum(y_i - \hat{y}_i)^2}}{\sum(y_i - \bar{y})^2}$$



# $R^2$ Value

**R-squared:** a suitable measure for evaluation. Let $\hat{y} = X\hat{\beta}$ be a predicted value, and $\bar{y}$ be the sample mean. Then the R-squared value is

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Can be described as the fraction of the total variance not explained by the model.

$$R^2 = \frac{[\mathrm{cov}(X,Y)]^2}{s_x^2 s_y^2} \quad or \quad R^2 = 1 - \frac{SSE}{\sum(y_i - \bar{y})^2}$$

**$R^2$ = 0:** Bad model. No evidence of a linear relationship.

**$R^2$ = 1: G**ood model. The line perfectly fits the data.

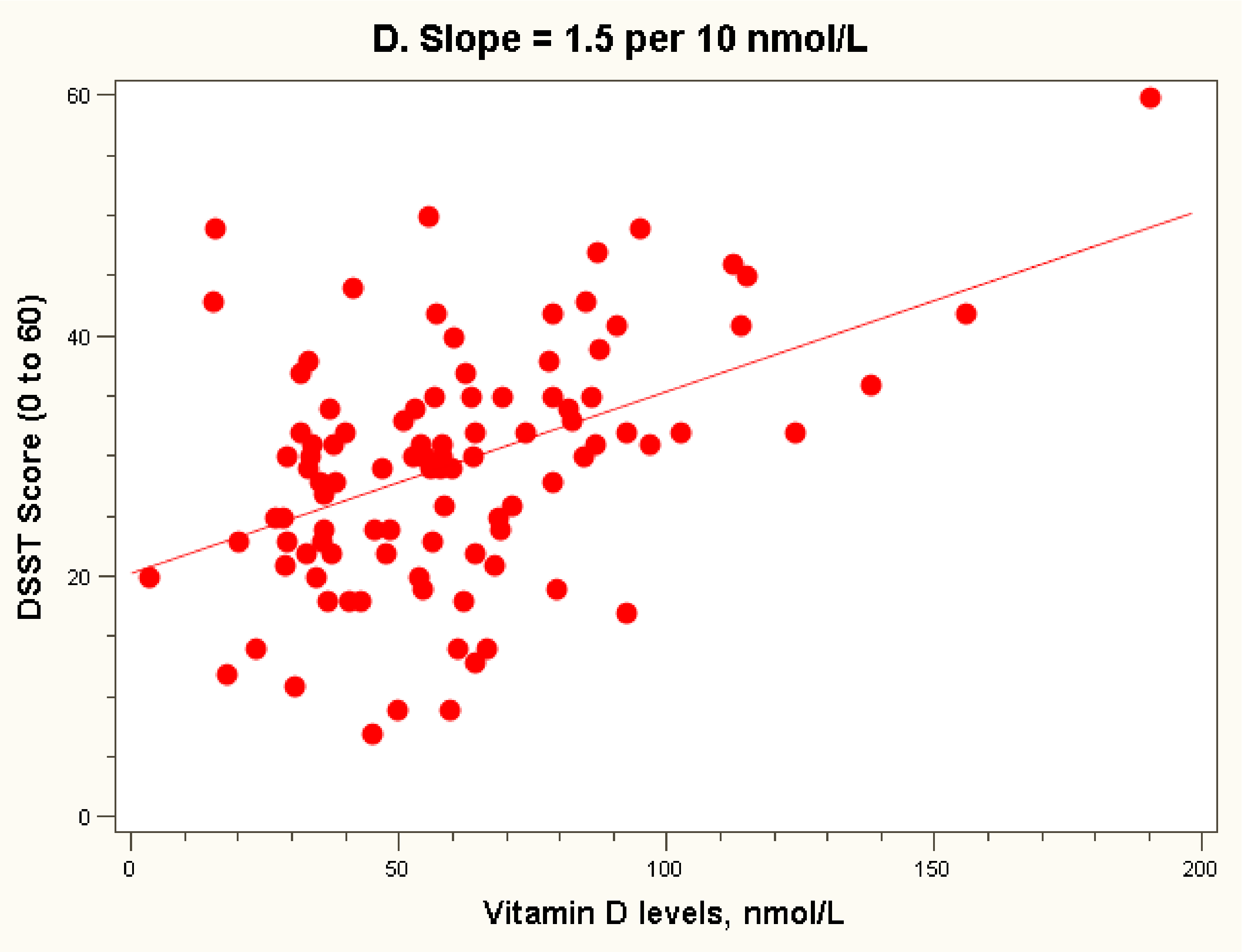
# R-Squared Example

```
. regress tlc height

      Source |       SS       df       MS              Number of obs =      32
-------------+------------------------------           F(  1,    30) =   89.12
       Model |  93.7825029     1  93.7825029           Prob > F      =  0.0000
    Residual |  31.5694921    30   1.0523164           R-squared     =  0.7482
-------------+------------------------------           Adj R-squared =  0.7398
       Total |  125.351995    31  4.04361274           Root MSE      =  1.0258

------------------------------------------------------------------------------
         tlc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      height |   .1417377    .015014     9.44   0.000     .1110749    .1724004
       _cons |  -17.10484   2.516234    -6.80   0.000    -22.24367    -11.966
------------------------------------------------------------------------------
```

R-squared = 0.748: 74.8 % of variation in TLC is characterized by the regression on height.

This corresponds to **correlation of sqrt(0.748) = 0.865** between predictions and actual TLCs. This is a precise prediction.

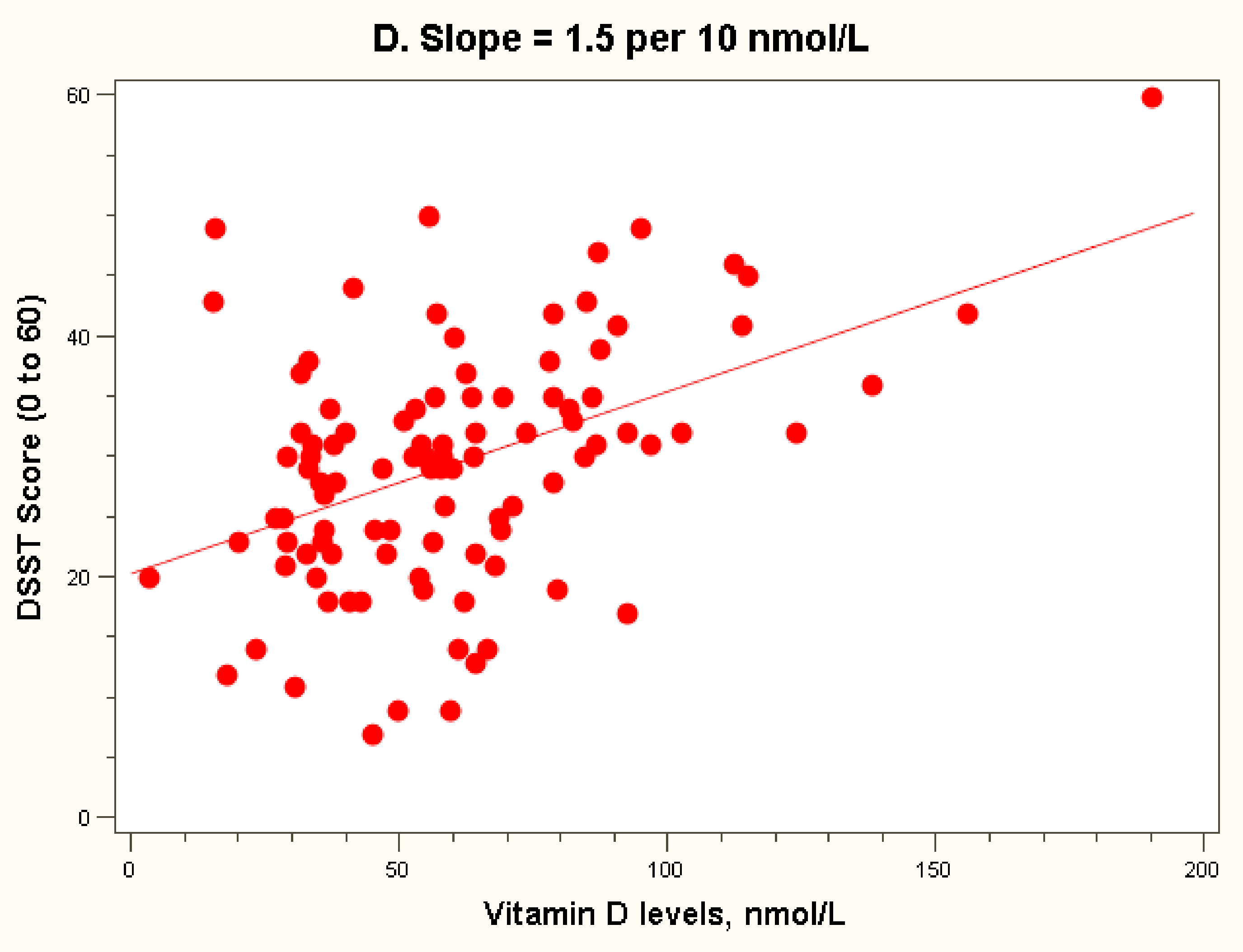
# Example: Cognitive Function and Vitamin D




Regression equation:

E(Yi) = 20 + 1.5*vit Di (in 10 nmol/L)

## Example: Cognitive Function and Vitamin D




**SDx** = 33 nmol/L

**SDy =** 10 points

**Cov(X,Y)** = 163 points*nmol/L

**Beta (ß)** = $163/33^2$ = 0.15 points per nmol/L

= 1.5 points per 10 nmol/L

$\mathbf{R^2}$ = 163/(10*33) = 0.49

or

$\mathbf{R^2}$ = 0.15 * (33/10) = 0.49

# Predict output for a New Input

This is our Regression model for the data

$$\hat{y}_i = 20 + 1.5x_i$$

**New Input whose DSST is unknown:**

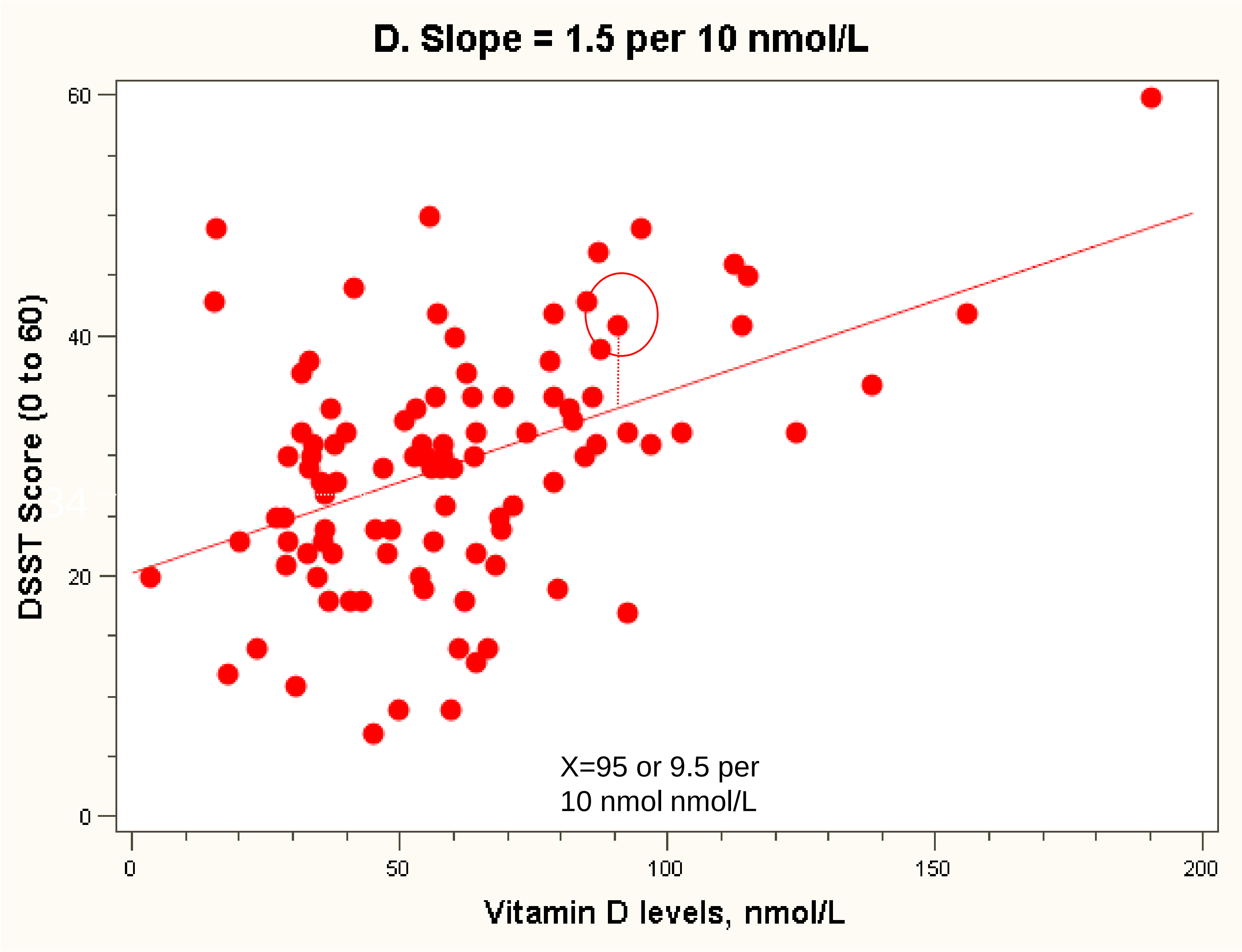
For an input of Vitamin D = 9.5 nmol/L, Predict the DSST Score

Plug the value of new input into the regression equation

$$\hat{y}_i = 20 + 1.5(9.5) = 34$$

*For the new input, the predicted DSST is 34*

# Residuals for Testing Model




**Residual = actual - predicted**

$y_i = 48$ Take any actual observed record from data

$\hat{y}_i = 34$ Pass this actual data as an input to model equation and get predicted value

$y_i - \hat{y}_i = 14$ Calculate residual by subtracting observed and predicted value. This tells the correctness of model

# Stock Market Prediction

- Estimate the market model for Nortel company stock traded in the Toronto Stock Exchange.

SUMMARY OUTPUT

| *Regression Statistics* | |
|---|---|
| Multiple R | 0.560079 |
| R Square | 0.313688 |
| Adjusted F | 0.301855 |
| Standard E | 0.063123 |
| Observatio | 60 |

ANOVA

This is a measure of the stock's market related risk. In this sample, for each 1% increase in the TSE return, the average increase in Nortel's return is .8877%.

This is a measure of the total risk embedded in the Nortel stock, that is market-related. Specifically, 31.37% of the variation in Nortel's return are explained by the variation in the TSE's returns.

| | *Coefficients* | *tandard Err* | *t Stat* | *P-value* |
|---|---|---|---|---|
| Intercept | 0.012818 | 0.008223 | 1.558903 | 0.12446 |
| TSE | 0.887691 | 0.172409 | 5.148756 | 3.27E-06 |

# Multiple Linear Regression

- Regression with more than one independent input variables (predictors)

- **"Multiple" Linear Regression**
  - More than one independent (input) variables X (example: height, age etc.)
  - With only one independent variable X we have "simple" linear regression

  - $\mathbf{Y_i = \beta_0 + \beta_1 \boldsymbol{X_{i1}} + \beta_2 \boldsymbol{X_{i2}} + \ldots + \beta_p \boldsymbol{X_{ik}}}$

- Intercept $\beta_0$ is mean Y with all X=0
- Slope $\beta_k$ is change in mean Y per 1 unit difference in $X_k$

# Multiple Linear Regression (2)

- In the same way that linear regression produces an equation that uses values of X to predict values of Y, **multiple regression** produces an equation that uses two different variables (X1, X2,…) to predict values of Y.

- The technique is used to predict the value of one variable (the dependent variable - Y) based on the value of other variables (independent variables $x_1$, $x_2$,…$x_k$)

# Other Regression Techniques

- Different Regression techniques are also available to predict the continuous outputs
  - Decision Tree Regressor
  - Random Forest Regressor
  - Support Vector Regressor (SVR)

# Examples from book

SimpleLR - Notepad

File Edit Format View Help

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array([[0],[1],[2],[3]])
y = np.array([2,3,4,5])

model = LinearRegression()
clf = model.fit(X, y)
print ('Coefficient: ', clf.coef_)
print('Y intercept: ', clf.intercept_)
```

Command Prompt

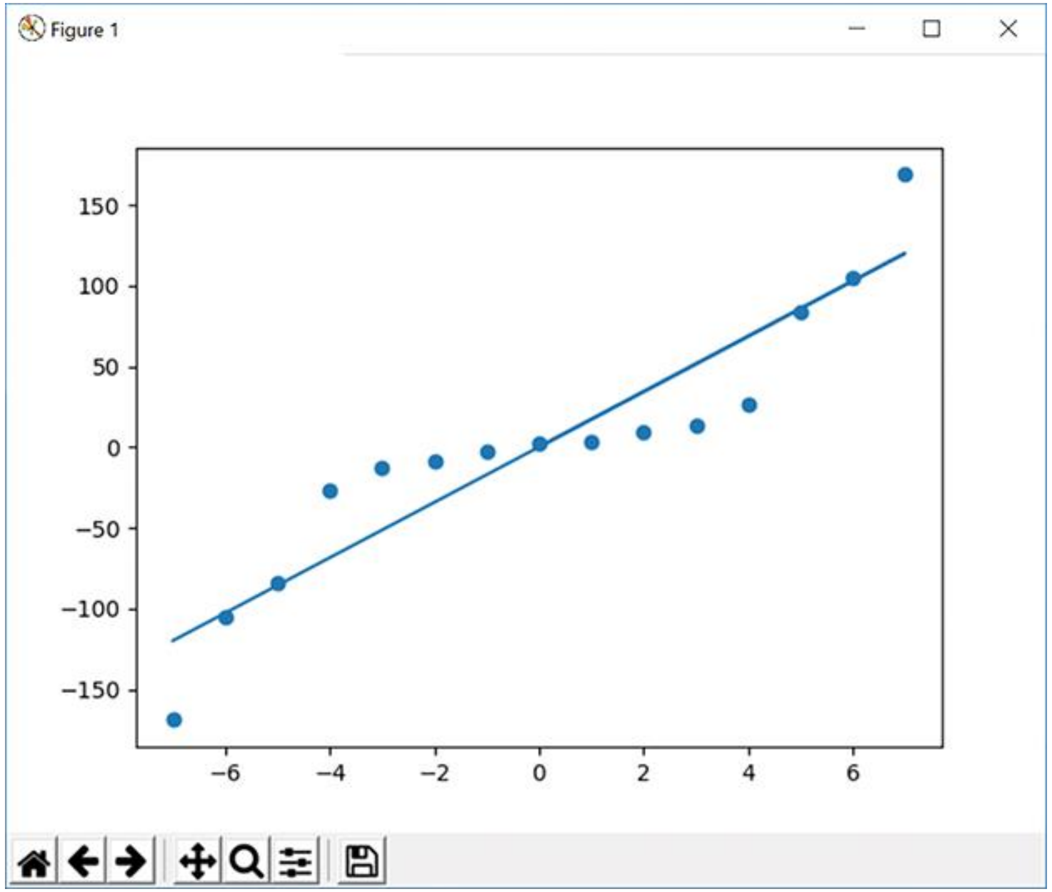
```
C:\Python>python simplelr.py
Coefficient:  [1.]
Y intercept:  2.0

C:\Python>
```

# Using Linear Regression to Draw a Line Using Python

PlotLR - Notepad

File Edit Format View Help

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array([[0],[1],[2],[3],[4],[5],[6],[7]])
y = np.array([2,3,9,13,27,84,105,169])

plt.scatter(X,y)

model = LinearRegression()
clf = model.fit(X, y)
predictions = np.dot(X, clf.coef_)

for index in range(len(predictions)):
 predictions[index] = predictions[index] +
clf.intercept_

plt.plot(X, predictions)
```



# Using Linear Regression to Predict MPG Based on Weight Using Python

WeightMPG - Notepad

File Edit Format View Help

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

data = pd.read_csv('auto-mpg.csv')

X = data[['weight']].values
y = data['mpg']

model = LinearRegression(fit_intercept=False)
clf = model.fit(X, y)
print ('Coefficient: ', clf.coef_)

predictions = model.predict(X)
for index in range(len(predictions)):
  print('Actual: ', y[index], 'Predicted: ',
predictions[index], 'Weight: ', X[index,0])
```

Command Prompt

```
C:\Python>python weightmpg.py
Coefficient:  [0.00669058]
Actual:  18.0 Predicted:  23.443794170864074 Weight:  3504
Actual:  15.0 Predicted:  24.708313890696637 Weight:  3693
Actual:  18.0 Predicted:  22.988834694945478 Weight:  3436
Actual:  16.0 Predicted:  22.968762953360834 Weight:  3433
Actual:  17.0 Predicted:  23.07581224181227 Weight:  3449
Actual:  15.0 Predicted:  29.04381007297972 Weight:  4341
Actual:  14.0 Predicted:  29.130787619846508 Weight:  4354
Actual:  14.0 Predicted:  28.849783237661494 Weight:  4312
Actual:  14.0 Predicted:  29.605818837349748 Weight:  4425
Actual:  15.0 Predicted:  25.758735033626333 Weight:  3850
Actual:  15.0 Predicted:  23.838538422028734 Weight:  3563
Actual:  14.0 Predicted:  24.14630512632661 Weight:  3609
Actual:  15.0 Predicted:  25.163273366615233 Weight:  3761
Actual:  14.0 Predicted:  20.647131510070356 Weight:  3086
Actual:  24.0 Predicted:  15.870057012925107 Weight:  2372
Actual:  22.0 Predicted:  18.954414636432055 Weight:  2833
Actual:  18.0 Predicted:  18.559670385267392 Weight:  2774
Actual:  21.0 Predicted:  17.308531826491254 Weight:  2587
Actual:  27.0 Predicted:  14.250936525097167 Weight:  2130
Actual:  26.0 Predicted:  12.277215269273851 Weight:  1835
Actual:  25.0 Predicted:  17.877231171389496 Weight:  2672
Actual:  24.0 Predicted:  16.258110683561558 Weight:  2430
Actual:  25.0 Predicted:  15.890128754509751 Weight:  2375
Actual:  26.0 Predicted:  14.946756900031488 Weight:  2234
Actual:  21.0 Predicted:  17.716657238712347 Weight:  2648
Actual:  10.0 Predicted:  30.87702913771053 Weight:  4615
```

# Calculating Coefficients Using Multiple Regression Using Python

MultipleLR - Notepad

File Edit Format View Help

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array([[0, 6, 11],[2, 7, 12],[3, 8, 13],[4, 9, 14],
[5, 10, 15]])
y = np.array([46,52,58,64,70])

model = LinearRegression()
clf = model.fit(X, y)
print ('Coefficient: ', clf.coef_)
print('Y intercept: ', clf.intercept_)
```

Command Prompt

```
C:\Python>python MultipleLR.py
Coefficient:  [3.62415977e-15 3.00000000e+00 3.00000000e+00]
Y intercept:  -5.0

C:\Python>
```

# Using Multiple Regression to Predict MGP Using Python

AutoMPGMR - Notepad

File Edit Format View Help

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

data = pd.read_csv('auto-mpg.csv')

X = data[['weight', 'horsepower', 'cylinders', 'acceleration',
'displacement','model year', 'origin']].values
y = data['mpg']

model = LinearRegression(fit_intercept=False)
clf = model.fit(X, y)
print ('Coefficient: ', clf.coef_)

y2 = model.predict(X)
for index in range(len(y2)):
  print('Actual: ', y[index], 'Predicted: ', y2[index], 'Weight: ', X[index,0])
```

Command Prompt

```
C:\Python>python AutoMPGMR.py
Coefficient:  [-0.00607987 -0.03489977 -0.62739129 -0.06569545  0.01
930388  0.5804971
  1.10007702]
Actual:  18.0 Predicted:  16.012841738633064 Weight:  3504.0
Actual:  15.0 Predicted:  14.505168381684879 Weight:  3693.0
Actual:  18.0 Predicted:  16.006316011283417 Weight:  3436.0
Actual:  16.0 Predicted:  15.688605843780202 Weight:  3433.0
Actual:  17.0 Predicted:  16.000260921193057 Weight:  3449.0
Actual:  15.0 Predicted:  11.037267543836581 Weight:  4341.0
Actual:  14.0 Predicted:  10.738726828451451 Weight:  4354.0
Actual:  14.0 Predicted:  10.93117374797421 Weight:  4312.0
Actual:  14.0 Predicted:  10.086165376642974 Weight:  4425.0
Actual:  15.0 Predicted:  13.647375623237803 Weight:  3850.0
Actual:  15.0 Predicted:  15.856624446673248 Weight:  3563.0
Actual:  14.0 Predicted:  15.22727188082479 Weight:  3609.0
Actual:  15.0 Predicted:  15.711818401075053 Weight:  3761.0
Actual:  14.0 Predicted:  18.22711675124657 Weight:  3086.0
Actual:  24.0 Predicted:  24.884430791034266 Weight:  2372.0
Actual:  22.0 Predicted:  20.234654447429214 Weight:  2833.0
Actual:  18.0 Predicted:  20.542871368040004 Weight:  2774.0
Actual:  21.0 Predicted:  22.0850611615269 Weight:  2587.0
Actual:  27.0 Predicted:  26.324044291498677 Weight:  2130.0
Actual:  26.0 Predicted:  28.089147588978072 Weight:  1835.0
```

# End of Part 3