Week4: Univariate Linear Regression

Data Science Certificate Program

Ryerson University

Bora Caglayan

5 Feb, 2015

Announcements

- Lab answers available on Blackboard.
- Homework deadline extended: March 1st.

Outline

- Correlation Analysis
- Univariate Linear Regression
- Application of Linear Regression with R
  - Correlation Analysis
  - Definition of Formula
  - Univariate Linear Regression
- Lab

Correlation Analysis

Correlation

Definition: The degree to which two or more attributes or measurements on the same group of elements show a tendency to vary together.

Pearson Correlation

Definitions:

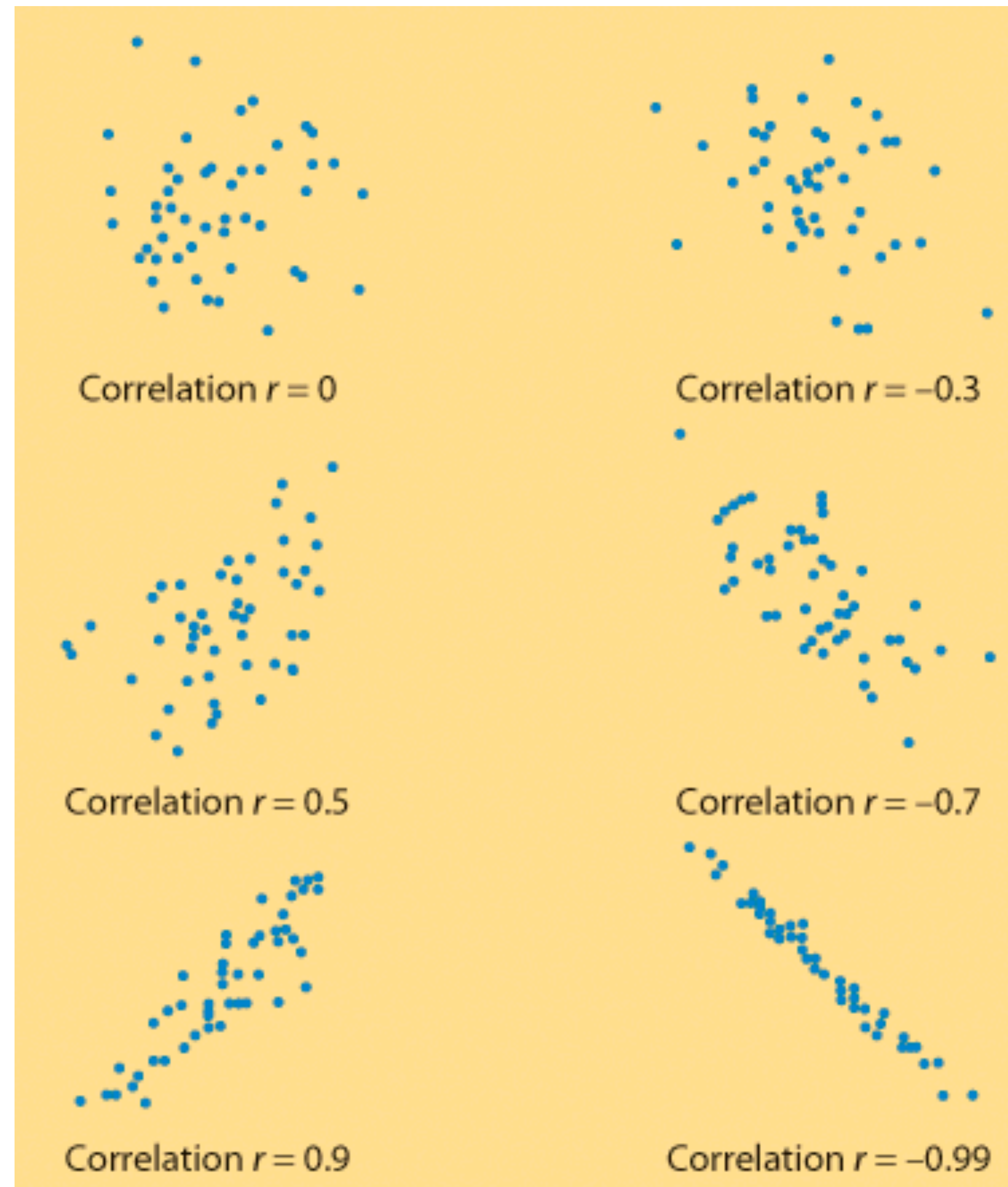
- Variance of X: $V(X)$
- Expected Value of X: $E(.)$
- mean of X: μ_x
- Covariance $\sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)] = E(XY) - \mu_X\mu_Y$
- Pearson Correlation $\rho_{XY} = \dfrac{\mathrm{cov}(X, Y)}{\sqrt{V(X)\,V(Y)}} = \dfrac{\sigma_{XY}}{\sigma_X \sigma_Y}$

Pearson Correlation

- We can approximate the strength and direction of the relation by a correlation estimation method.
- Pearson correlation method defines correlation as follows:

Strength: *how closely the points follow a straight line.*

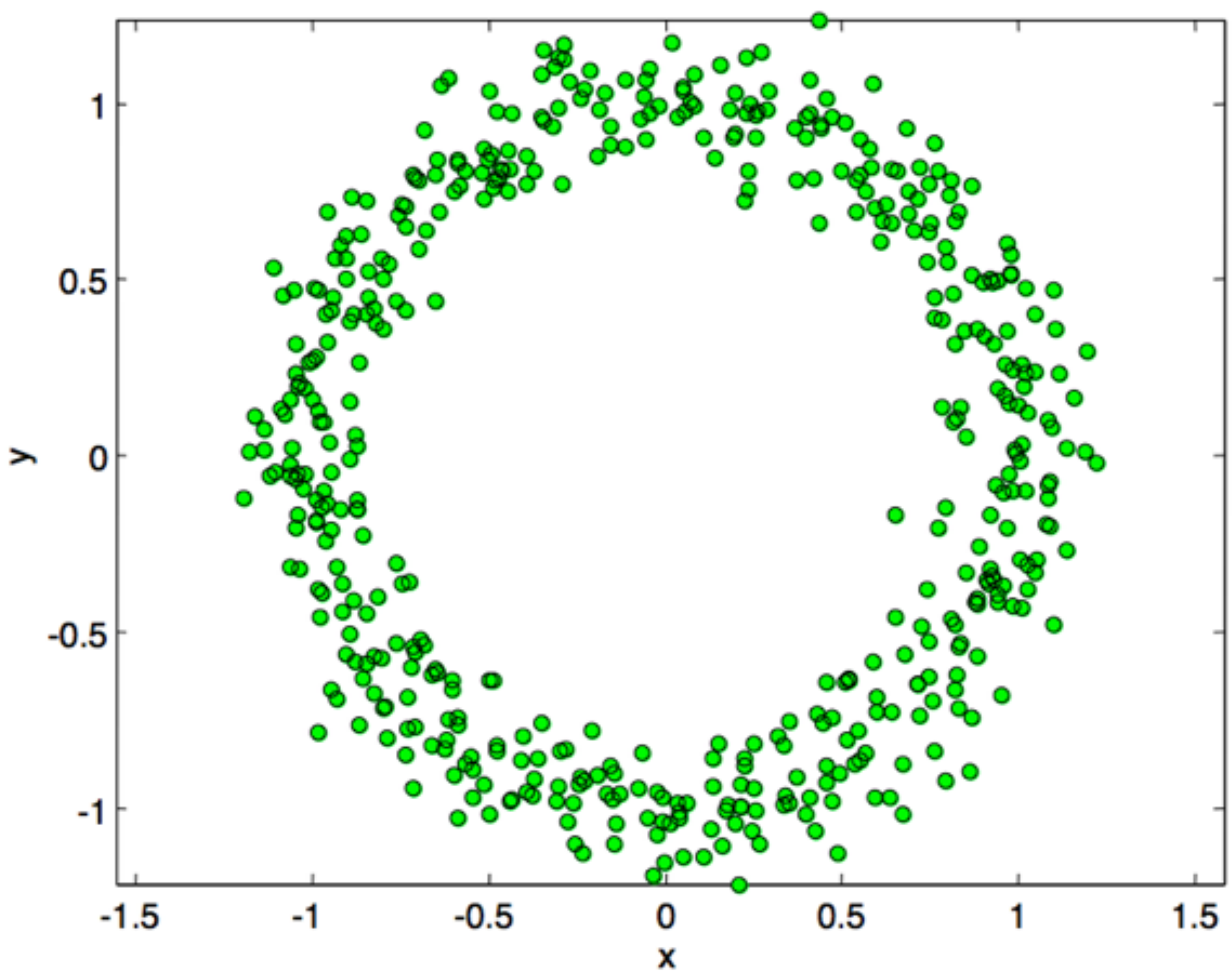
Direction*: is positive when individuals with higher X values tend to have higher values of Y.*



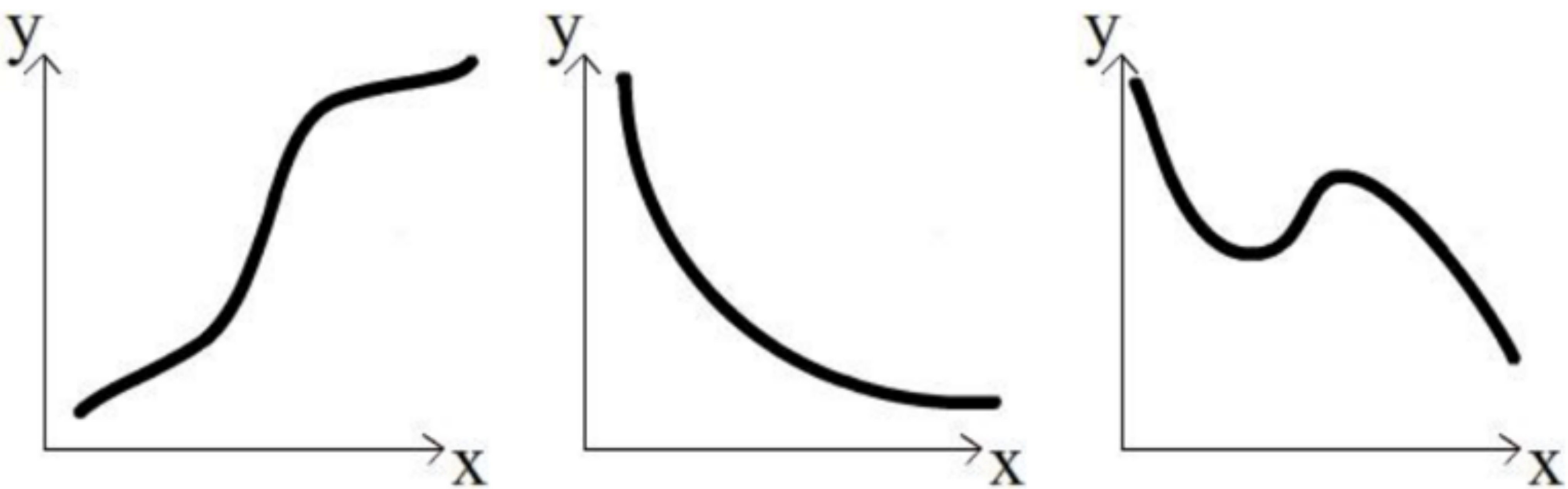
Properties of the Pearson Correlation Coefficient

- Pearson Correlation coefficient measures only linear relationship.
- Works well if both of the variables are normally distributed.
- High correlation does not imply causality.

- No correlation does not imply lack of patterns.



Spearman's Rank Correlation

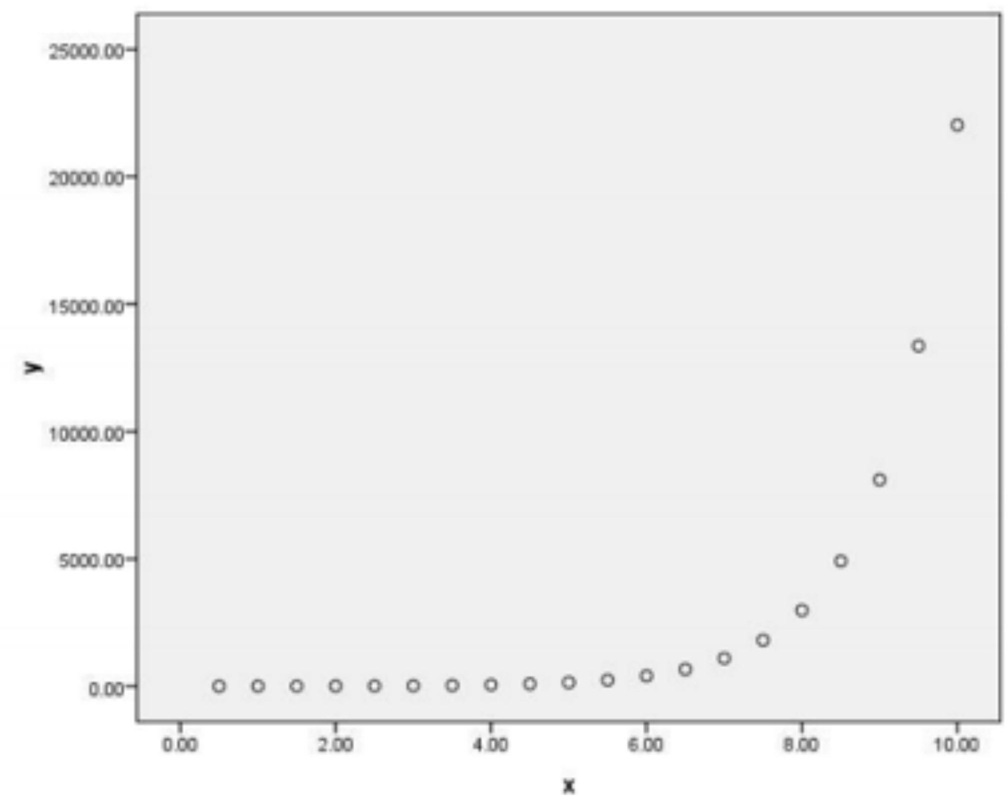


Monotonically increasing Monotonically decreasing Not monotonic

- Monotonically increasing - as the x variable increases the y variable never decreases.
- Monotonically decreasing - as the x variable increases the y variable never increases.
- Not monotonic - as the x variable increases the y variable sometimes decreases and sometimes increases.

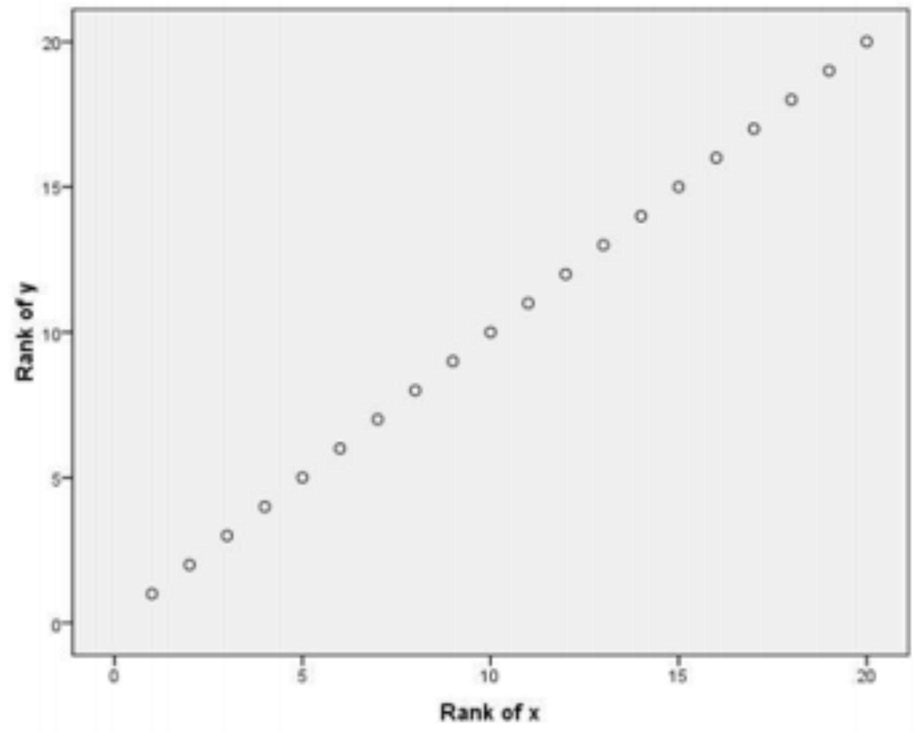
Spearman's Rank Correlation

	x	y
1	.5	1.6
2	1.0	2.7
3	1.5	4.5
4	2.0	7.4
5	2.5	12.2
6	3.0	20.1
7	3.5	33.1
8	4.0	54.6
9	4.5	90.0
10	5.0	148.4
11	5.5	244.7
12	6.0	403.4
13	6.5	665.1
14	7.0	1096.6
15	7.5	1808.0
16	8.0	2981.0
17	8.5	4914.8
18	9.0	8103.1
19	9.5	13359.7
20	10.0	22026.5



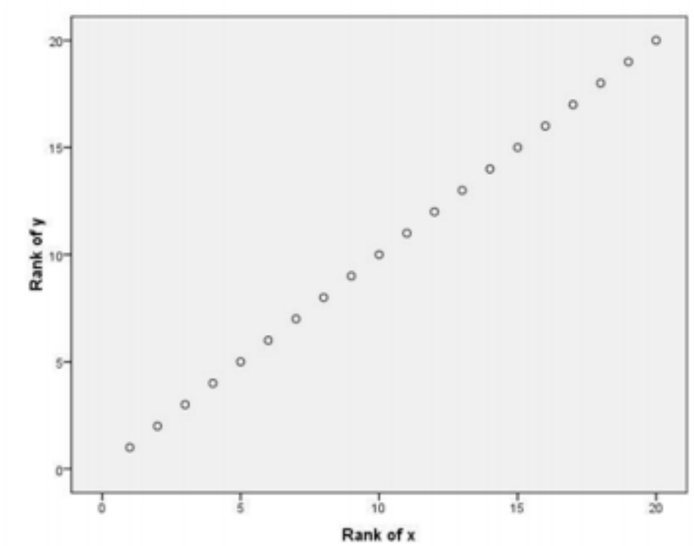
Spearman's Rank Correlation

	x	Rank of x	y	Rank of y
1	.5	1	1.6	1
2	1.0	2	2.7	2
3	1.5	3	4.5	3
4	2.0	4	7.4	4
5	2.5	5	12.2	5
6	3.0	6	20.1	6
7	3.5	7	33.1	7
8	4.0	8	54.6	8
9	4.5	9	90.0	9
10	5.0	10	148.4	10
11	5.5	11	244.7	11
12	6.0	12	403.4	12
13	6.5	13	665.1	13
14	7.0	14	1096.6	14
15	7.5	15	1808.0	15
16	8.0	16	2981.0	16
17	8.5	17	4914.8	17
18	9.0	18	8103.1	18
19	9.5	19	13359.7	19
20	10.0	20	22026.5	20



Spearman's Rank Correlation

	x	Rank of x	y	Rank of y
1	.5	1	1.6	1
2	1.0	2	2.7	2
3	1.5	3	4.5	3
4	2.0	4	7.4	4
5	2.5	5	12.2	5
6	3.0	6	20.1	6
7	3.5	7	33.1	7
8	4.0	8	54.6	8
9	4.5	9	90.0	9
10	5.0	10	148.4	10
11	5.5	11	244.7	11
12	6.0	12	403.4	12
13	6.5	13	665.1	13
14	7.0	14	1096.6	14
15	7.5	15	1808.0	15
16	8.0	16	2981.0	16
17	8.5	17	4914.8	17
18	9.0	18	8103.1	18
19	9.5	19	13359.7	19
20	10.0	20	22026.5	20



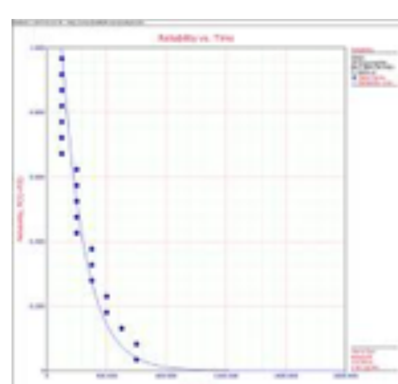
Formula: $$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}.$$

d_i: Rank difference

n: number of samples

Comparison of Spearman and Pearson Correlation

- Spearman checks monotonic relationship.
- Pearson checks linear relationship.
- If you want to explore your data it is best to compute both.
- Spearman > Pearson implies monotonic non linear relationship.
    - Example:



Statistical Significance of Correlation

- Significance tests: May the event be seen by chance or not?
  - Direction: Is there a positive or negative relationship?
    - We form null and alternate hypothesis
    - We use t-test to check the significance of correlation direction.
    - Based on n-2(degree of freedom) and desired probability critical probability value there is a **threshold**.
    - If t >threshold the direction is significance.

$$H_0 : \rho = 0$$

$$H_1 : \rho > 0$$

$$t = r\sqrt{\frac{n-2}{1-r^2}}$$

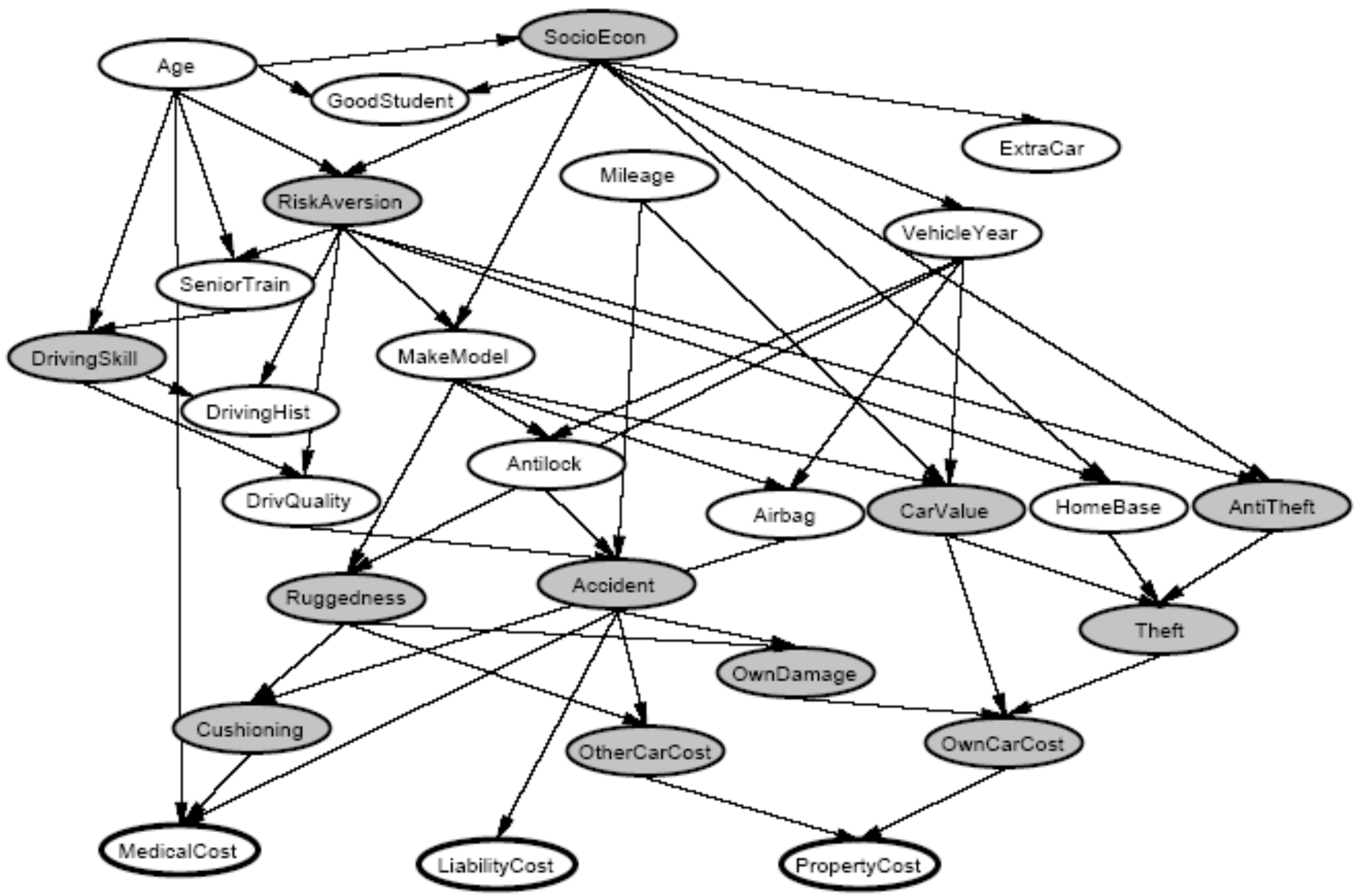
r: correlation

t: number of samples

- Strength: How strong is the relation? (high, low, very high etc.)
  - There are different guidelines for different disciplines.

In Real Projects…

There are more than 2 variables.

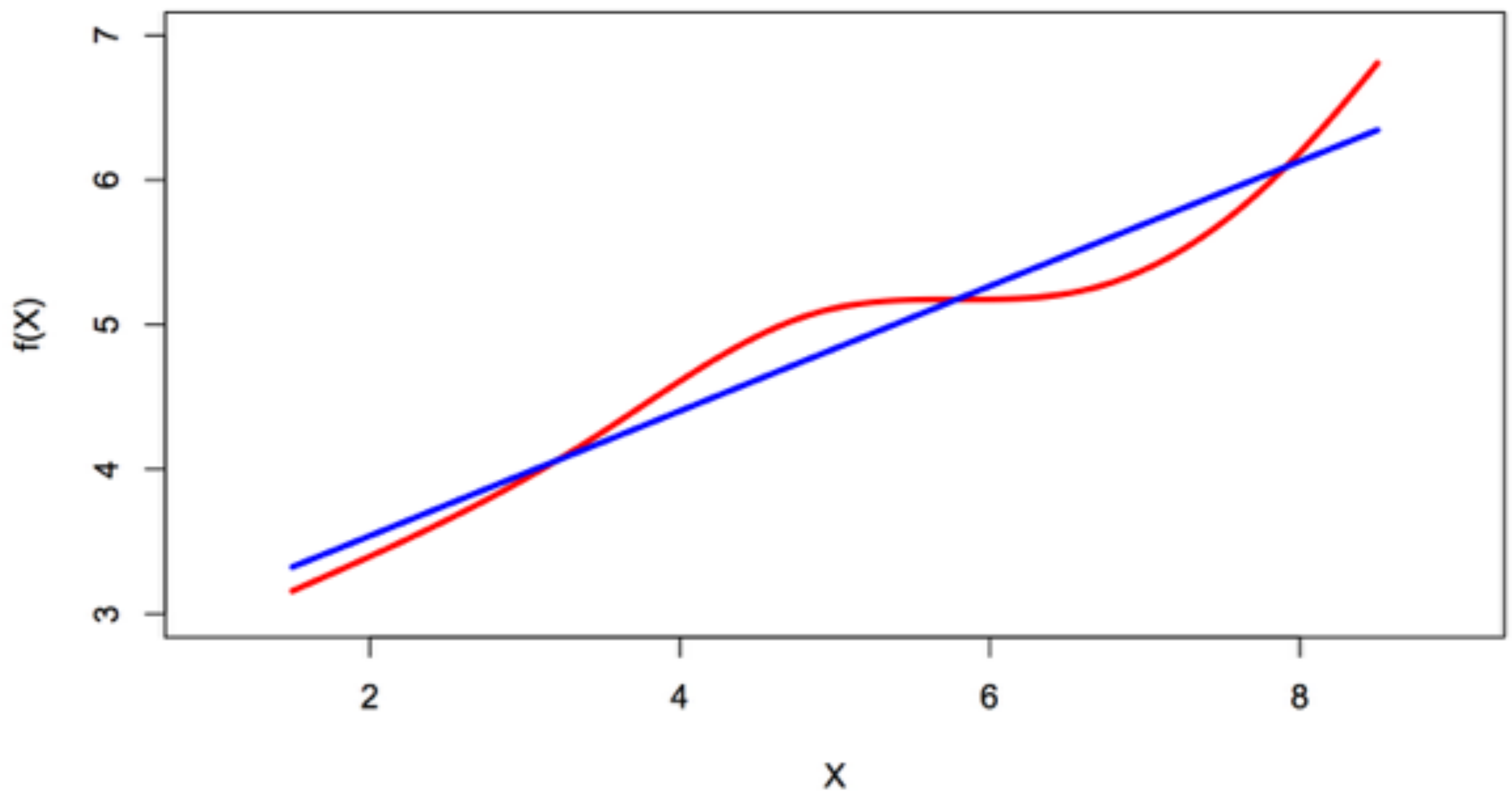


Univariate Linear Regression

"Essentially, all models are wrong, but some are useful."

–George Box

Linear Regression



- Linear regression is a simple approach to supervised learning. It assumes that the dependence of Y on X_1, X_2, . . . X_p is linear.

- True regression functions are never linear!

- Although it may seem overly simplistic, linear regression is extremely useful both conceptually and practically.

Univariate Linear Regression

Formally we define the problem as follows:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters

Intercept

objective function

Input variables

Regression Problem

Goal: Minimize cost function.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Cost function

(estimation-actual)^2

Regression Problem

Goal: Minimize cost function.

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Find theta for which $\frac{\partial}{\partial \theta_j} J(\theta) = 0$

How do we find the coefficients?

Approach 1: Closed Form Solution

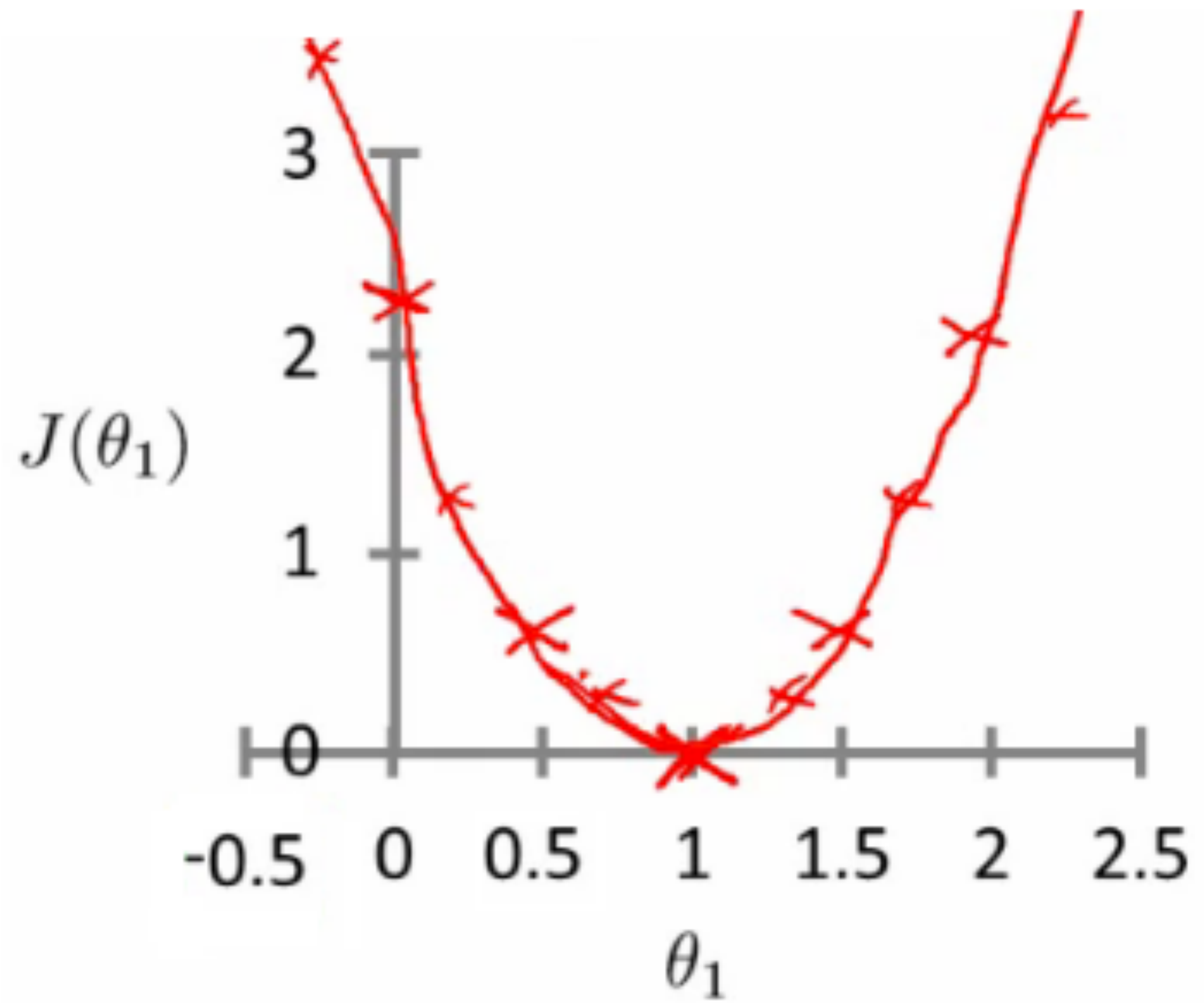
- Calculate theta that minimizes the cost function in a single step.

$$\theta = (X^T X)^{-1} X^T Y$$

X : Feature matrix
Y : Target vector

How do we find the coefficients?

- Approach 2: Gradient Descent

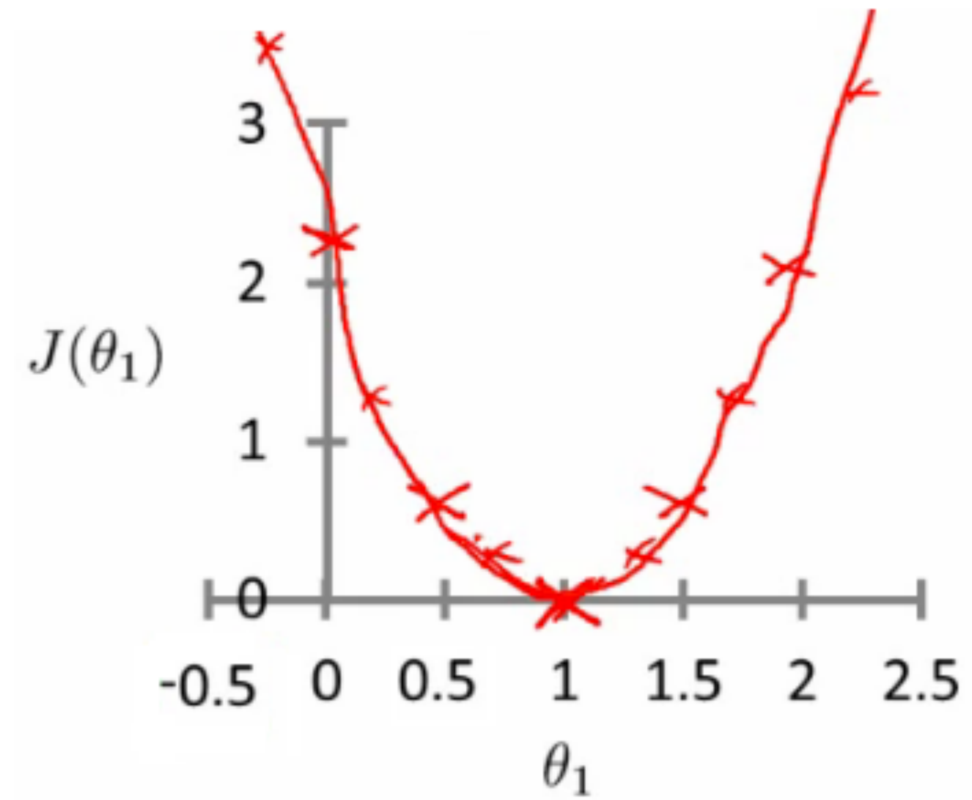


How do we find the coefficients?

Approach 2: Gradient Descent

- Do the following until convergence:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$



alpha term

What happens if alpha is too small or too large?

Too small

- Take baby steps
- Takes too long

Too large

- Can overshoot the minimum and fail to converge

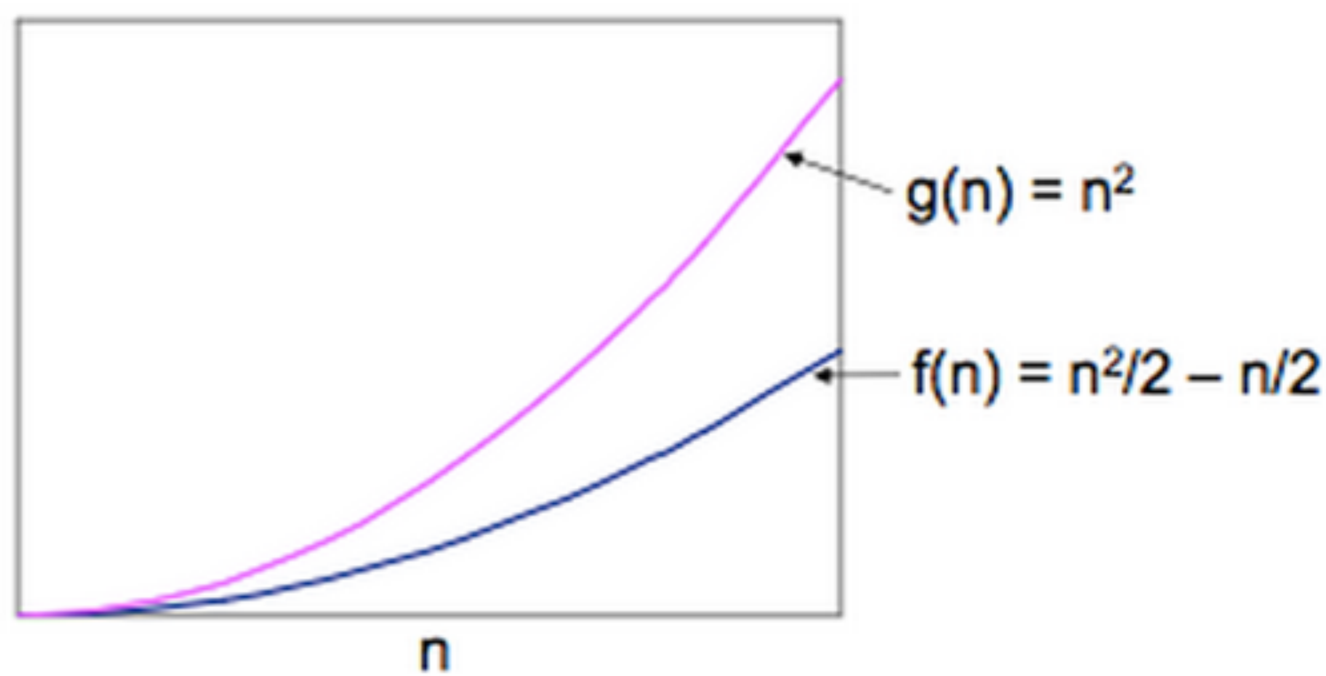
Interlude: Very Short Introduction to Big O Notation

Big-O Notation

- $f(n) = O(g(n))$ if there exist positive constants c and n_0 such that $f(n) <= cg(n)$ for all $n >= n_0$
- Example: $f(n) = n^2/2 - n/2$ is $O(n^2)$, because

$$n^2/2 - n/2 \;<=\; n^2 \quad \text{for all } n >= 0.$$

c = 1 $\qquad$ $n_0 = 0$



- Big-*O* notation specifies an *upper bound* on a function f(n) as n grows large.

Big-O Notation

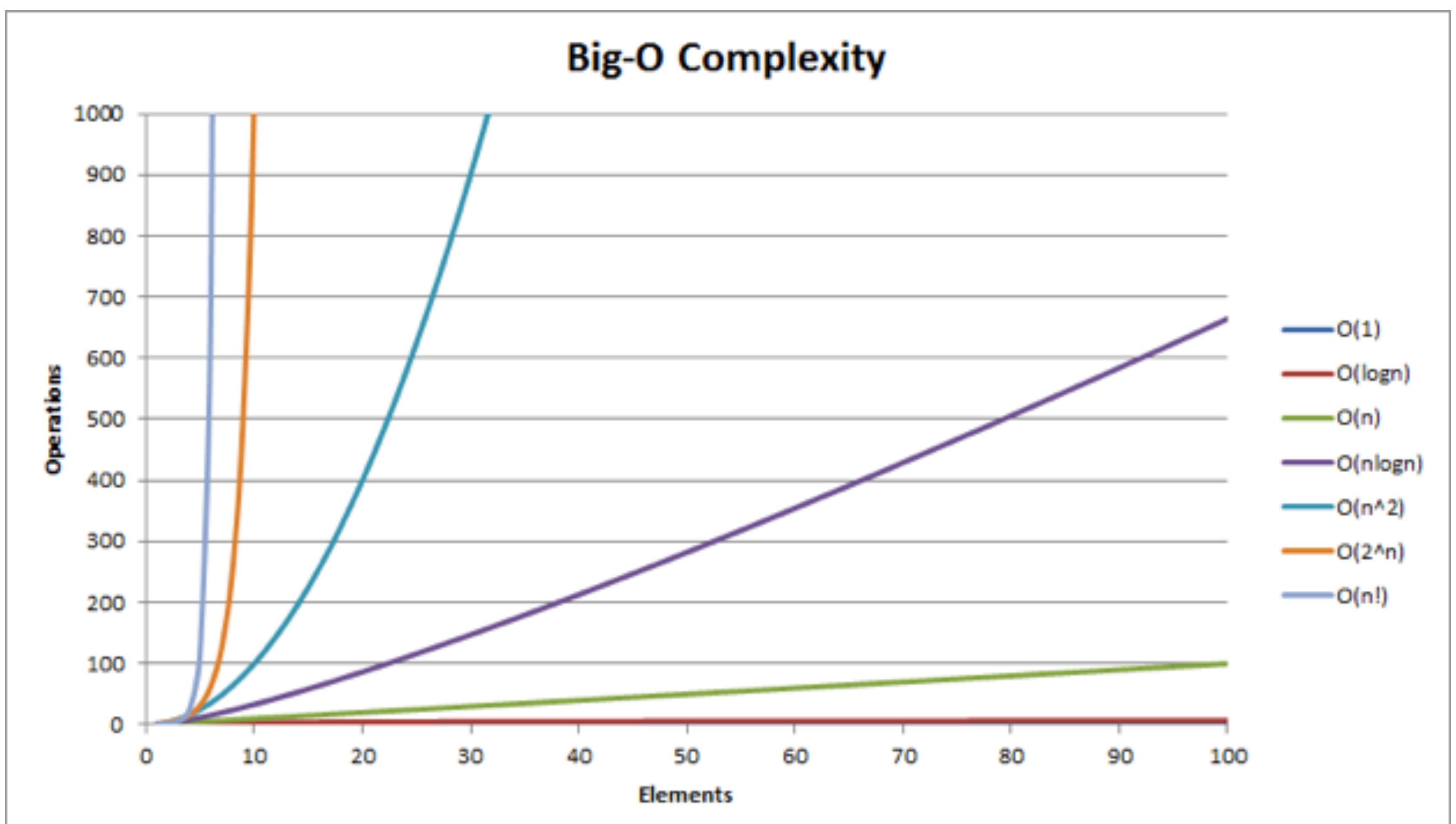
Examples:

Complexity	10	20	30
n	0.00001 sec	0.00002 sec	0.00003 sec
n^2	0.0001 sec	0.0004 sec	0.0009 sec
n^3	0.001 sec	0.008 sec	0.027 sec
n^5	0.1 sec	3.2 sec	24.3 sec
2^n	0.001 sec	1.0 sec	17.9 min
3^n	0.59 sec	58 min	6.5 years

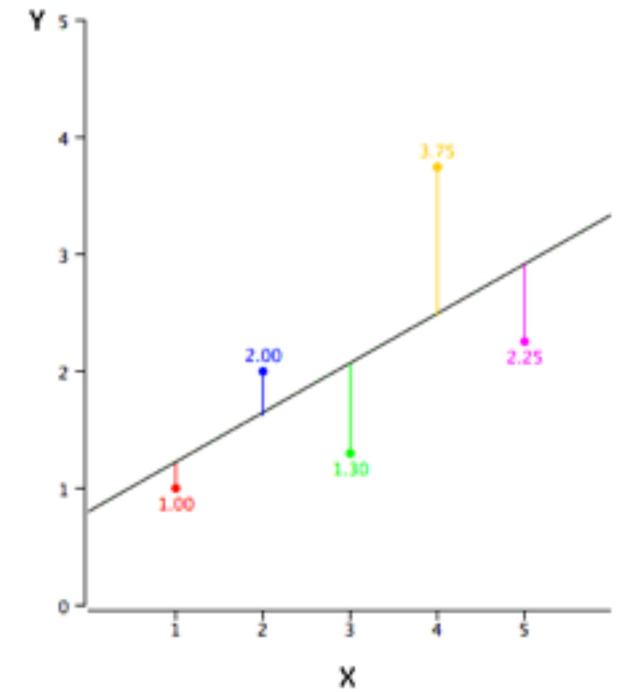
Problem of Closed Form Solution

- Calculation of $(X*X^T)$ and $(X*X^T)^{-1}$ is computationally expensive.

- Closed form solution: **O(**$N^{2.373}$**)**



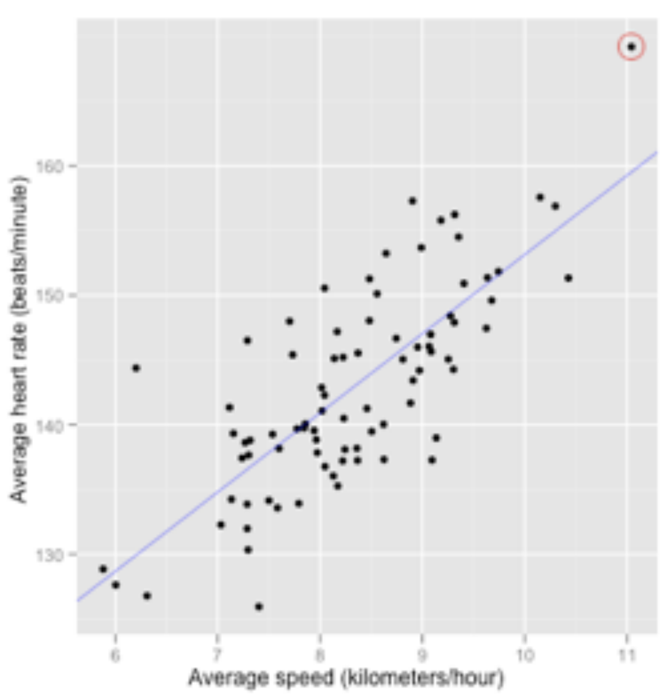
Performance Criteria

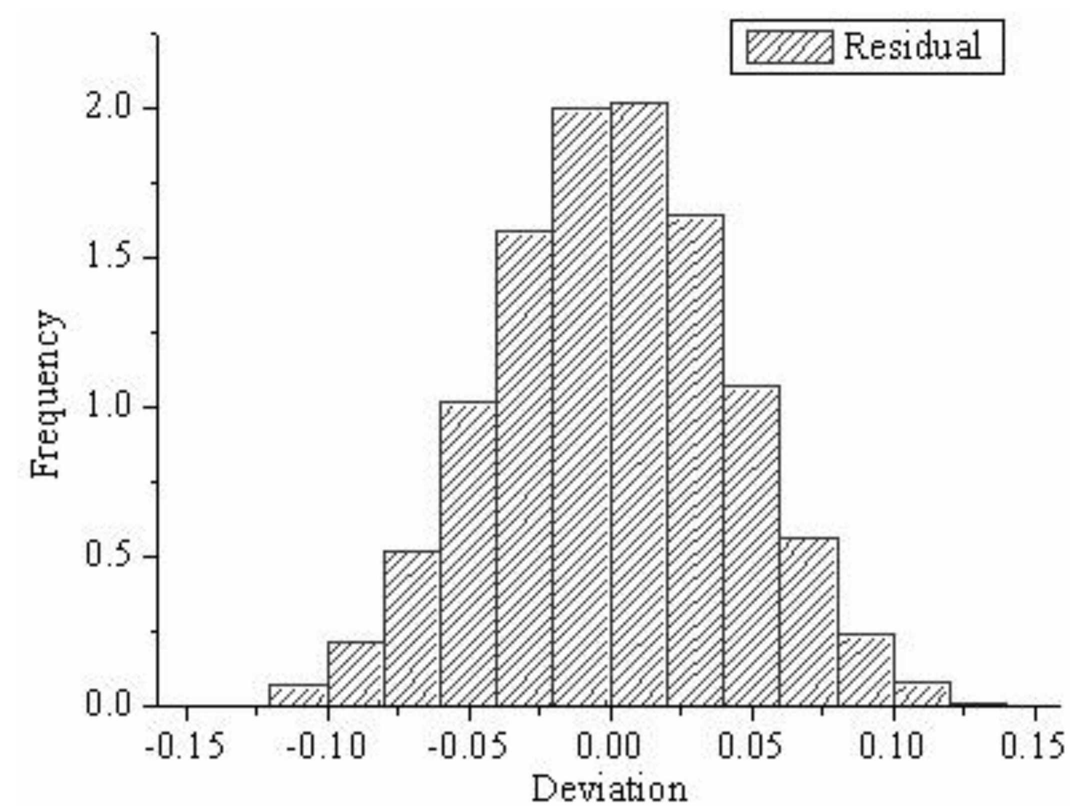


Distribution of Residuals

Check the distribution of errors

$Error_{(i)} = h(x_{(i)}) - y_{(i)}$





Performance Criteria

RMSE (Root mean square error)

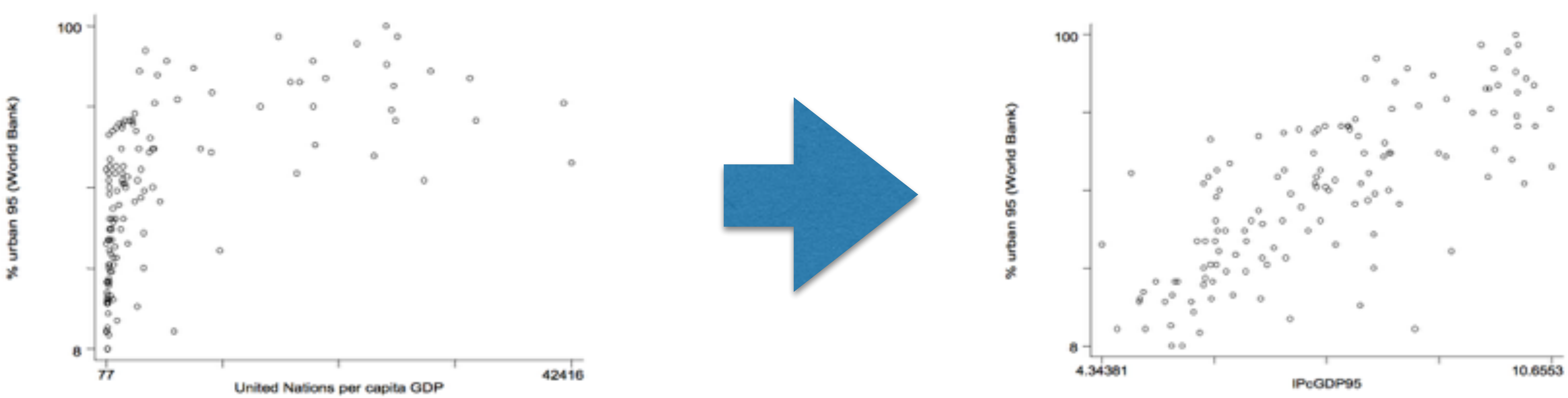
$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

pred(X)

- Percentage of predictions with at most X percent error rate.
- Might be used if small errors are not important.

Changing Scales

We may be able to change the relation to linear by changing the scales of the variables.



This type of data manipulation is a simple case of **data pre-processing**.

Setting up the Experiment

Random split strategy:

Steps:

1. Pick 80 percent of the values randomly.

2. Train the model. Find the intercept and theta.

    Example: *If theta is 2 and intercept is 32, the regression function is:* ***2*x+32****)*

3. Test your regression function on the remaining 20 percent of the data.

4. Performance criteria: RMSE

Setting up the Experiment

k-fold Cross validation strategy (better for generalizability):

Steps:

1. Divide your data into k groups.
2. for group=1..k:
    A. training_set = dataset - group[k]
    B. test_set = group[k]
    C. train the model. Find the intercept and theta.
    D. Test your regression function with the test set.
3. Check the findings for every fold
4. Performance criteria: RMSE

Summary

Uni-variate linear regression

- **Definition:** Predict the value of a numeric variable based on a single input variable.
- **Exploratory analysis:** Check correlations
- **Preprocessing:** Changing scales
- **Algorithms:** Gradient descent, closed form solution with linear algebra
- **Experiment:** 10-fold cross validation, random split
- **Performance criteria:** RMSE, pred(x), error distribution
- **Advantages:** Simplicity, low computation cost, explains relation between input and target variable well, good base line model
- **Disadvantage:** May not be a good fit for most data.

References

- [Book] A basic statistics textbook: http://ca.wiley.com/WileyCDA/WileyTitle/productCd-EHEP002914.html

- Introduction to big O complexity: http://pages.cs.wisc.edu/~vernon/cs367/notes/3.COMPLEXITY.html

- In depth analysis of linear regression: http://cs229.stanford.edu/notes/cs229-notes1.pdf

Week 4 Application Part

February 5, 2015

Finding Correlation Coefficients

Spearman:

```r
cor(iris$Sepal.Length, iris$Petal.Length,
    method="spearman")
```

```
## [1] 0.8818981
```

Pearson:

```r
cor(iris$Sepal.Length, iris$Petal.Length,
    method="pearson")
```

```
## [1] 0.8717538
```

Correlation Matrix

```
cor(iris[,1:3])
```

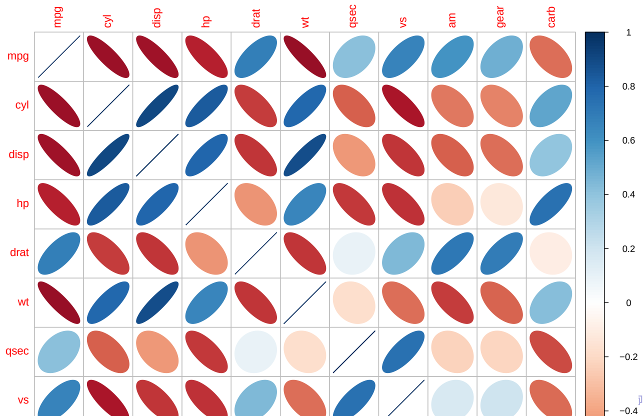
```
##              Sepal.Length Sepal.Width Petal.Length
## Sepal.Length    1.0000000  -0.1175698    0.8717538
## Sepal.Width    -0.1175698   1.0000000   -0.4284401
## Petal.Length    0.8717538  -0.4284401    1.0000000
```

Visualizing Correlation

```
## Loading required package: corrplot
```

```
{r setup, echo=FALSE}rel library("knitr")
opts_chunk$set(dev = 'pdf')
```

```
corrplot(cor(mtcars), method="ellipse")
```



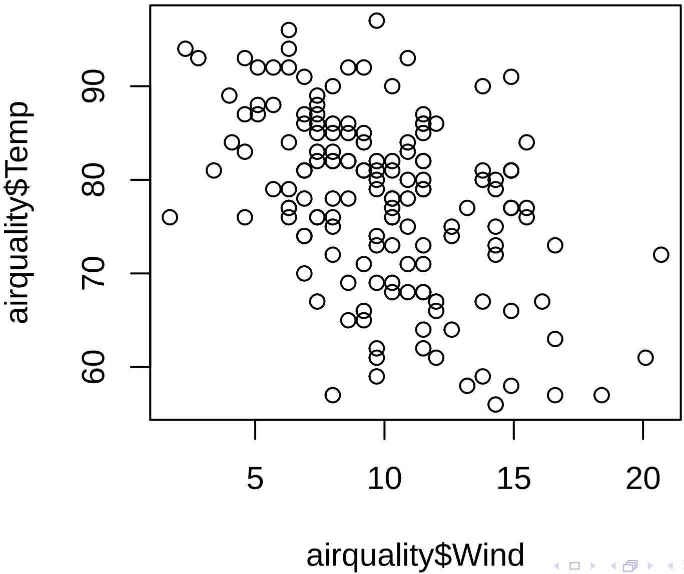
Significance of Correlation Coefficients

```
cor.test(iris$Sepal.Length, iris$Petal.Length,
         method = c("pearson"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  iris$Sepal.Length and iris$Petal.Length
## t = 21.646, df = 148, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to
## 95 percent confidence interval:
##  0.8270363 0.9055080
## sample estimates:
##       cor
## 0.8717538
```

Linear Regression: Fitting the Model

```
plot(airquality$Wind, airquality$Temp)
```



Linear Regression: Fitting the Model

```r
model_ulm <- lm(Wind~Temp, data=airquality)
summary(model_ulm)
```

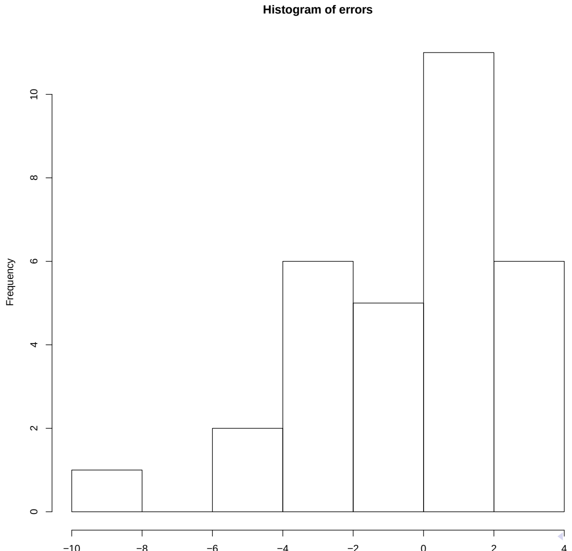
```
##
## Call:
## lm(formula = Wind ~ Temp, data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5784 -2.4489 -0.2261  1.9853  9.7398
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.23369    2.11239  10.999  < 2e-16 ***
## Temp        -0.17046    0.02693  -6.331 2.64e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
##
```

Linear Regression: Prediction

```
rn_train <- sample(nrow(airquality),
                   floor(nrow(airquality)*0.8))
train <- airquality[rn_train,c("Wind","Temp")]
test <- airquality[-rn_train,]
model_ulm <- lm(Wind~Temp, data=train)
prediction <- predict(model_ulm, interval="prediction",
                      newdata =test)
```

Linear Regression: Error Distribution

```r
errors <- prediction[,"fit"] - test$Wind
hist(errors)
```



Linear Regression: RMSE

```r
sqrt(sum((prediction[,"fit"] - test$Wind)^2)/nrow(test))
```

```
## [1] 3.011712
```

Linear Regression: PRED(25)

Find the percentage of cases with less than 25 percent error:

```r
rel_change <- 1 - ((test$Wind - abs(errors)) / test$Wind)
table(rel_change<0.25)["TRUE"] / nrow(test)
```

```
##      TRUE
## 0.6774194
```

Lab

Preparation Required Libraries

```
install.packages("corrplot")
require("corrplot")
```

Lab

Preparation Data load

```
library(RCurl)
u <- getURL("http://vincentarelbundock.github.io/Rdatasets/
c_prices <- read.csv(text = u)
```

Lab Questions

1- Find spearman correlation between hard disk space and ram.

2- Visualize the correlation of the numeric columns in the computer prices dataset.

3- Choose a single variable to predict price and build an univariate linear regression model.

4- Experiment with 30 percent split of the data. Report error distribution, RMSE and pred(25)