Introduction to Data Science, Topic 5

- Instructor: Professor Henry Horng-Shing Lu,
  Institute of Statistics, National Chiao Tung University, Taiwan
  Email: hslu@stat.nctu.edu.tw
- WWW: http://www.stat.nctu.edu.tw/misg/hslu/course/DataScience.htm
- Reference:
  M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.
- Evaluation: Homework: 50%, Term Project: 50%
- Office hours: By appointment

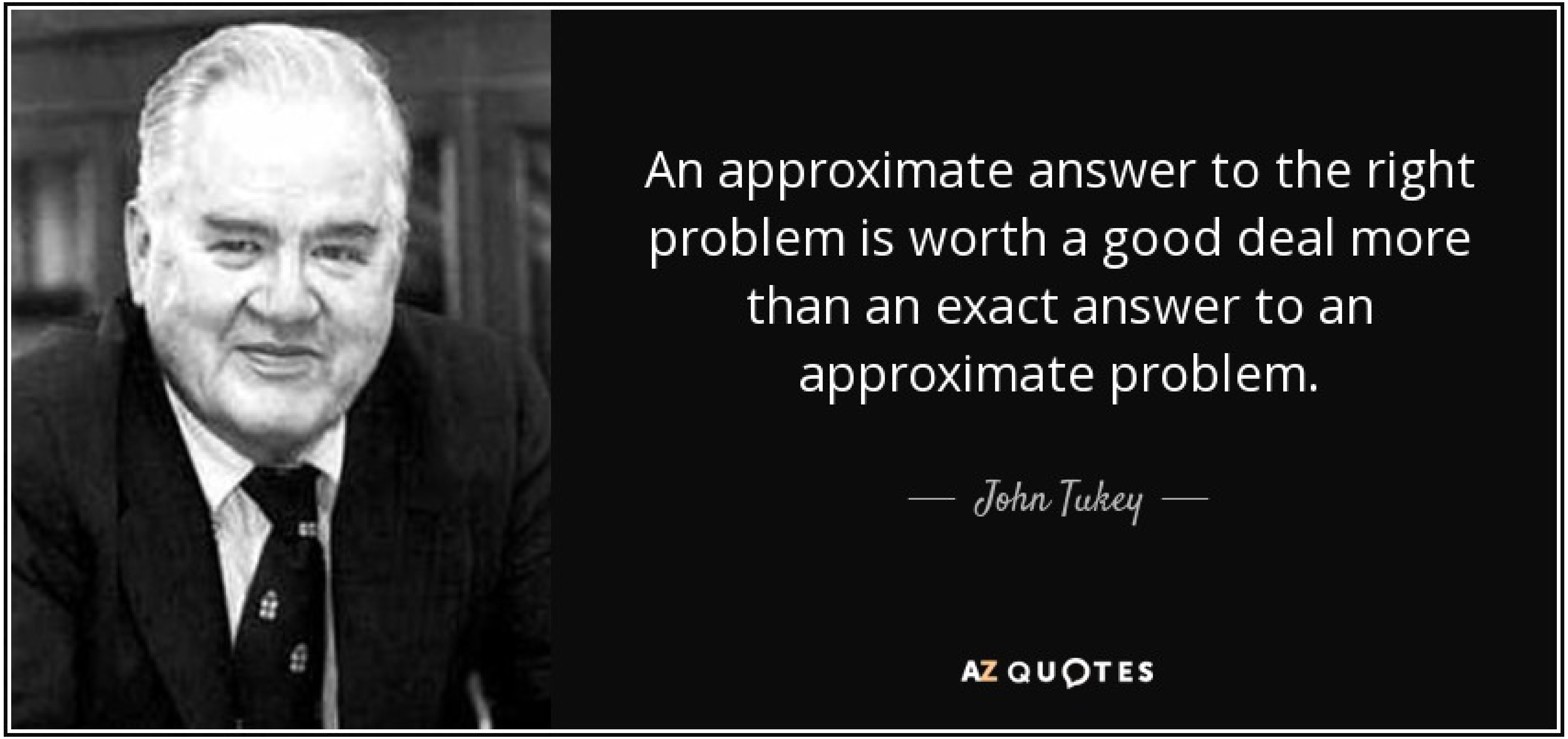
Course Outline

- **Introduction of Data Science**
- **Introduction of R**
- **Process Real Data by R**
- **Data Visualization**
- **Exploratory Data Analysis**
- **Regression**
- **Classification**
- **Text Mining**
- **Clustering**

Exploratory Data Analysis with R

References:

Ch. 5, M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.



http://www.azquotes.com/author/14847-John_Tukey

John W. Tukey wrote the book *Exploratory Data Analysis* in 1977. Tukey held that too much emphasis in statistics was placed on statistical hypothesis testing (confirmatory data analysis); more emphasis needed to be placed on using data to suggest hypotheses to test.

Exploratory Data Analysis

- ***Summary Statistics***
- ***Getting Sense of Data Distribution***
- ***Putting it all together: Outlier Detection***

In this topic, we'll use the US Census Demographic Data on Kaggle:
https://www.kaggle.com/muonneutrino/us-census-demographic-data/data#_=_

Summary Statistics – Data size

the `dim()` function output the numbers of rows and columns.

```
> census <- read.csv("acs2015_county_data.csv")
> dim(census)
[1] 3220   37
```

Our data contain 3220 records (in this census data means 3220 counties), and 37 variables.

We can also use `nrow()` and `ncol()` function to find only the number of rows and columns, respectively.

```
> nrow(census)
[1] 3220
> ncol(census)
[1] 37
```

Summary Statistics – Summarizing the Data

The `head()`and `tail()`functions, output the first and last few entries of a data frame. It is useful to have a glimpse of data.

```
> head(census[, 1:8])
  CensusId   State  County TotalPop   Men Women Hispanic White
1     1001 Alabama Autauga    55221 26745 28476      2.6  75.8
2     1003 Alabama Baldwin   195121 95314 99807      4.5  83.1
3     1005 Alabama Barbour    26932 14497 12435      4.6  46.2
4     1007 Alabama    Bibb    22604 12073 10531      2.2  74.5
5     1009 Alabama  Blount    57710 28512 29198      8.6  87.9
6     1011 Alabama Bullock    10678  5660  5018      4.4  22.2
```

Summary Statistics – Summarizing the Data

The `summary()` functions give the brief summary for each column.

```
> summary(census[, 1:8])
    CensusId          State              County           TotalPop
 Min.   : 1001   Texas   : 254   Washington:  31   Min.   :      85
 1st Qu.:19033   Georgia : 159   Jefferson :  26   1st Qu.:   11218
 Median :30024   Virginia: 133   Franklin  :  25   Median :   26035
 Mean   :31394   Kentucky: 120   Jackson   :  24   Mean   :   99409
 3rd Qu.:46106   Missouri: 115   Lincoln   :  24   3rd Qu.:   66430
 Max.   :72153   Kansas  : 105   Madison   :  20   Max.   :10038388
                 (Other) :2334   (Other)   :3070
      Men              Women             Hispanic          White
 Min.   :     42   Min.   :     43   Min.   : 0.000   Min.   : 0.00
 1st Qu.:   5637   1st Qu.:   5572   1st Qu.: 1.900   1st Qu.:64.10
 Median :  12932   Median :  13057   Median : 3.900   Median :84.10
 Mean   :  48897   Mean   :  50512   Mean   :11.012   Mean   :75.43
 3rd Qu.:  32993   3rd Qu.:  33488   3rd Qu.: 9.825   3rd Qu.:93.20
 Max.   :4945351   Max.   :5093037   Max.   :99.900   Max.   :99.80
```

Summary Statistics – Summarizing the Data

For categorical variables like State and County, the summary contains the number of times occur in each value, so we can see that Washington occur 31 times. We can see all Washington records by:

```
> census[which(census$County == "Washington"), 1:6]
 CensusId          State     County TotalPop    Men  Women
65       1129        Alabama Washington    16997   8490   8507
183      5143       Arkansas Washington   216432 108144 108288
306      8121       Colorado Washington     4795   2482   2313
387     12133        Florida Washington    24629  13478  11151
537     13303        Georgia Washington    20785  10467  10318
595     16087          Idaho Washington    10025   5082   4943
690     17189       Illinois Washington    14457   7242   7215
785     18175        Indiana Washington    27930  13867  14063
881     19183           Iowa Washington    22017  10847  11170
                                  .
                                  .
                                  .
```

Summary Statistics – Summarizing the Data

For numeric variables, the summary contain:

- Min.—smallest value of the variable.
- 1st Qu. (Q1)—first quartile or 25th percentile
- Median—second quartile or 50th percentile
- Mean—Average value of the variable.
- 3rd Qu. (Q2)—third quartile or 75th percentile
- Max.—largest value of the variable.

These statistics are useful to get a sense of the data distribution for a variable: its range and centrality.

Summary Statistics – Ordering Data by a Variable

`sort()` function sorts vectors or data frame by a variable.

```
> sort(census$TotalPop)
  [1]    85   117   267   433   443   448   548   551   565   565
 [11]   606   643   673   675   681   705   711   733   756   769
 [21]   776   778   781   812   820   821   847   851   874   901
        ...
```

`sort()` function sorts vectors in descending order, if we set `decreasing = T`

```
> sort(census$TotalPop, decreasing = T)
  [1] 10038388  5236393  4356362  4018143  3223096  3116069
  [7]  2639042  2595259  2485003  2301139  2298032  2094769
 [13]  2045756  2035572  1914526  1868149  1843152  1825502
        ...
```

`sort()` function sorts also can sort string alphabetically.

Summary Statistics – Ordering Data by a Variable

`order()`function orders the data frame for given variable in one step.

```
> census[order(-census$TotalPop)[1:3], 1:6]
     CensusId      State      County TotalPop     Men   Women
205      6037 California Los Angeles 10038388 4945351 5093037
611     17031   Illinois        Cook  5236393 2537245 2699148
2624    48201      Texas      Harris  4356362 2166727 2189635
```

The minus sign in front of the `census$TotalPop` means sorted in reverse. `order()`function can also sort on multiple variables, like `order(variable1, varialbe2, …)`.

Summary Statistics – Group and Split Data by a Variable

We can select a subset by:

```
> which(census$State == "California")
```

But if we want to do this repeatly and perform the analysis on that, we can use `by()` function.

```
> by(census$TotalPop, census$State, mean)
census$State: Alabama
[1] 72098.81
-------------------------------------------------
census$State: Alaska
[1] 25288.79
-------------------------------------------------
...
```

Summary Statistics – Group and Split Data by a Variable

There is a similar function `split()`, which split the data and output it in a list.

```
> data.split = split(census, census$State)
> for (x in names(data.split)) {
+   dd = data.split[[x]]
+   print(x)
+   print(dd[order(-dd$TotalPop)[1:5], c(3, 4)])
+ }
[1] "Alabama"
       County TotalPop
37  Jefferson   659026
49     Mobile   414251
45    Madison   346438
51 Montgomery   228138
59     Shelby   203530
       ...
```

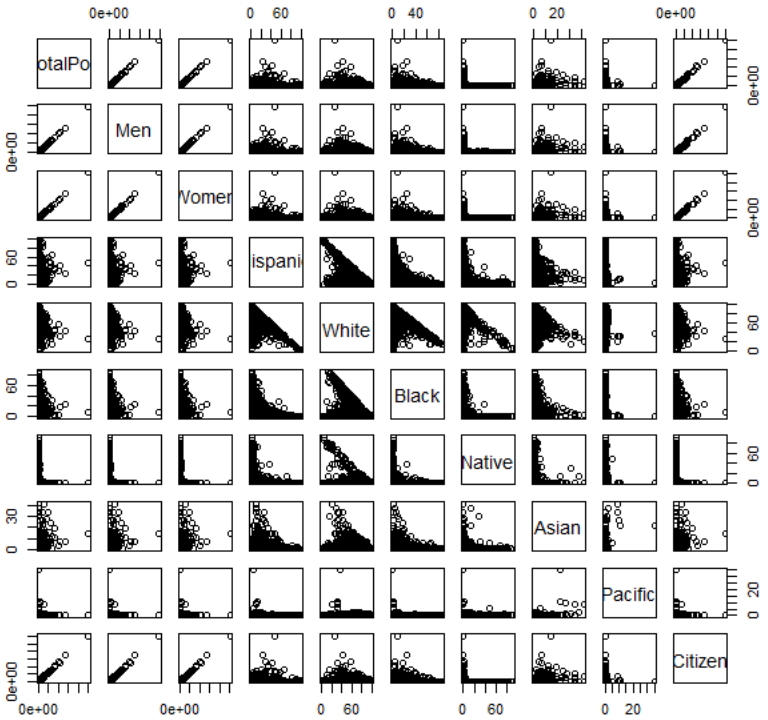
Summary Statistics – Variable Correlation

`cor()`function computes the correlation of a pair of variable.

```
> cor(census$TotalPop, census$Men)
[1] 0.9998772
```

We can use the `cor()`function to obtain the pairwise correlation between a set of numeric variables

```
> cor(census[, 4:13])
> pairs(census[, 4:13])
```

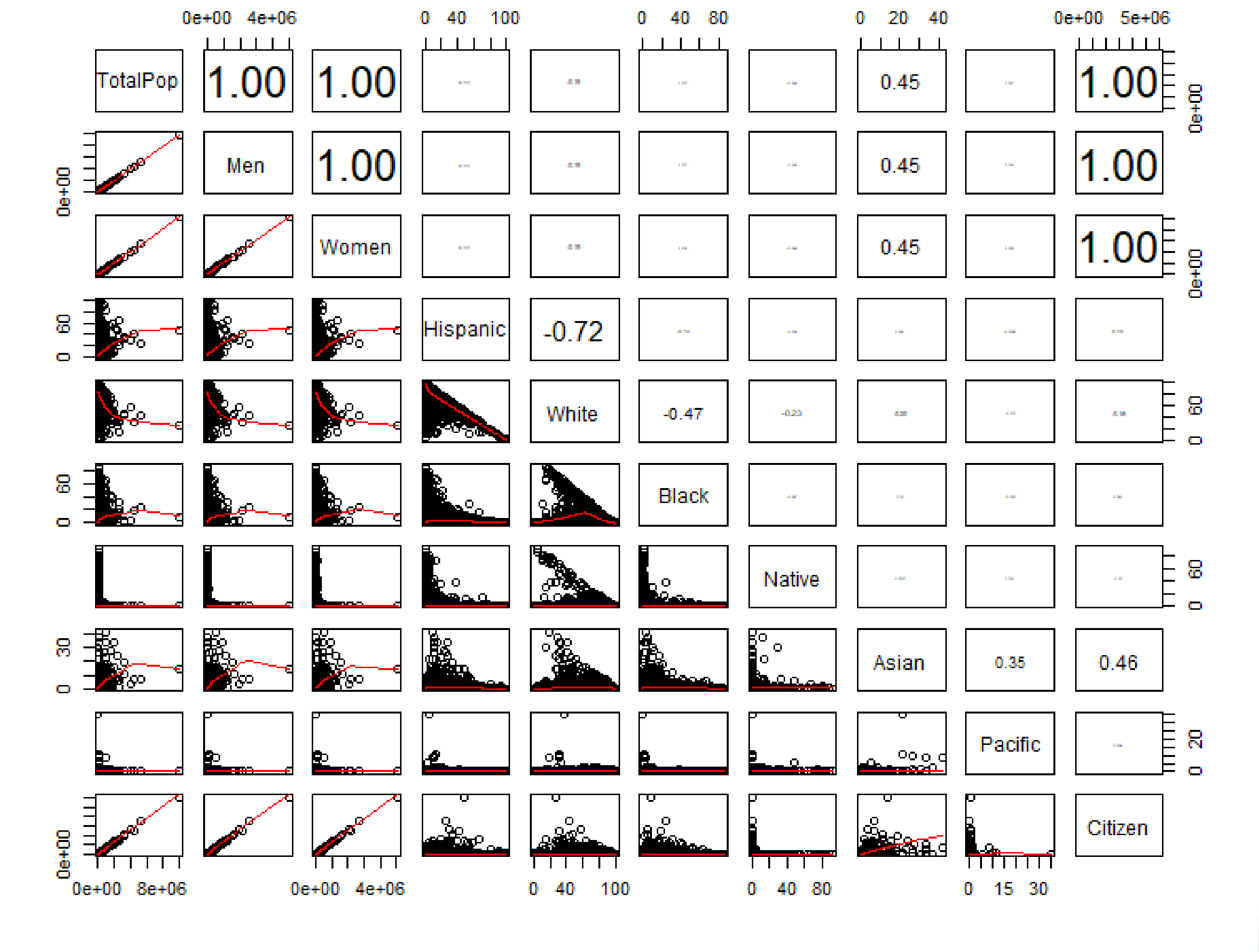


Summary Statistics – Variable Correlation

Or we can customized pairs plot:

```
> panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor =
10, ...) {
+   usr <- par("usr"); on.exit(par(usr))
+   par(usr = c(0, 1, 0, 1))
+   r <- cor(x, y)
+   txt <- format(c(r, 0.123456789), digits = digits)[1]
+   txt <- paste(prefix, txt, sep = "")
+   if(missing(cex.cor)) cex.cor <- 1/strwidth(txt)
+   text(0.5, 0.5, txt, cex = cex.cor * abs(r))
+ }
> pairs(census[, 4:13], lower.panel = panel.smooth,
+       upper.panel = panel.cor)
```

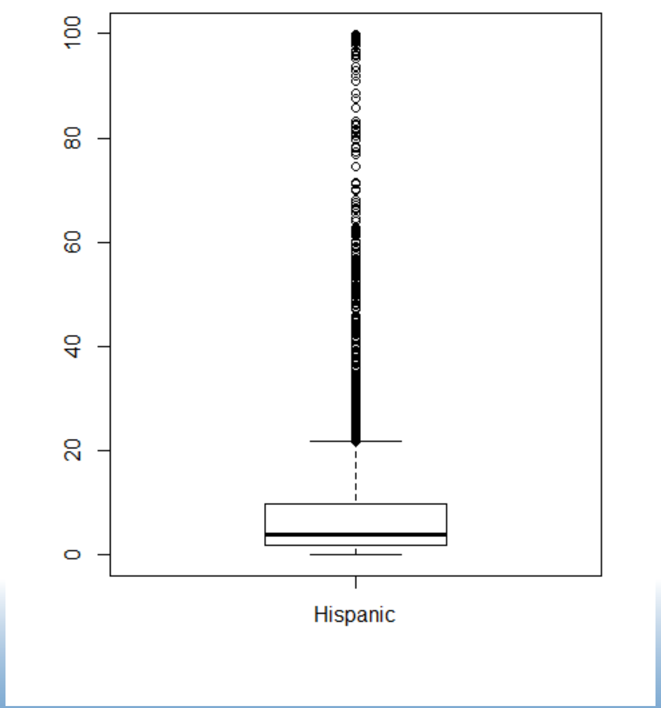
Summary Statistics – Variable Correlation



Getting a Sense of Data Distribution–Box Plots

Box plots based on the five-number summary statistics of a variable (minimum, Q1, median, Q3, maximum)

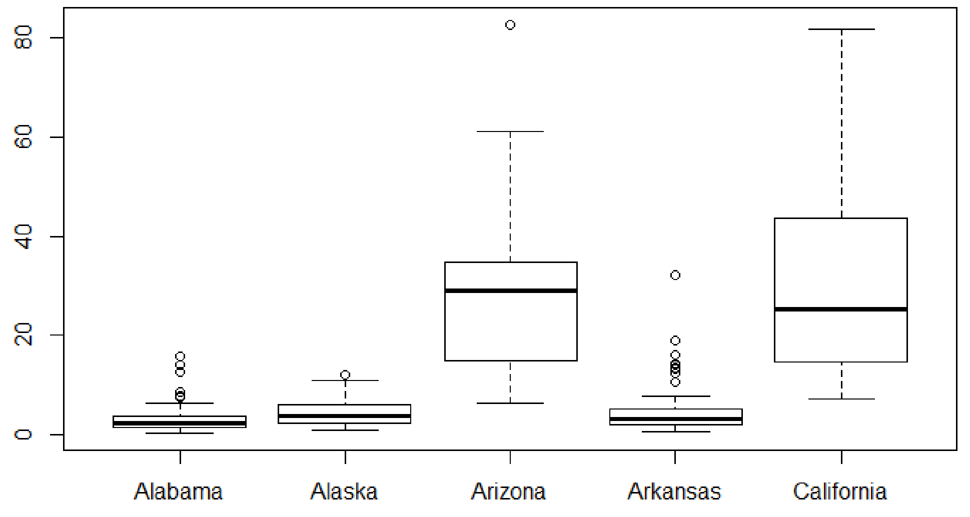
```
boxplot(census$Hispanic, names = c("Hispanic"), show.names = T)
```



Getting a Sense of Data Distribution– Box Plots

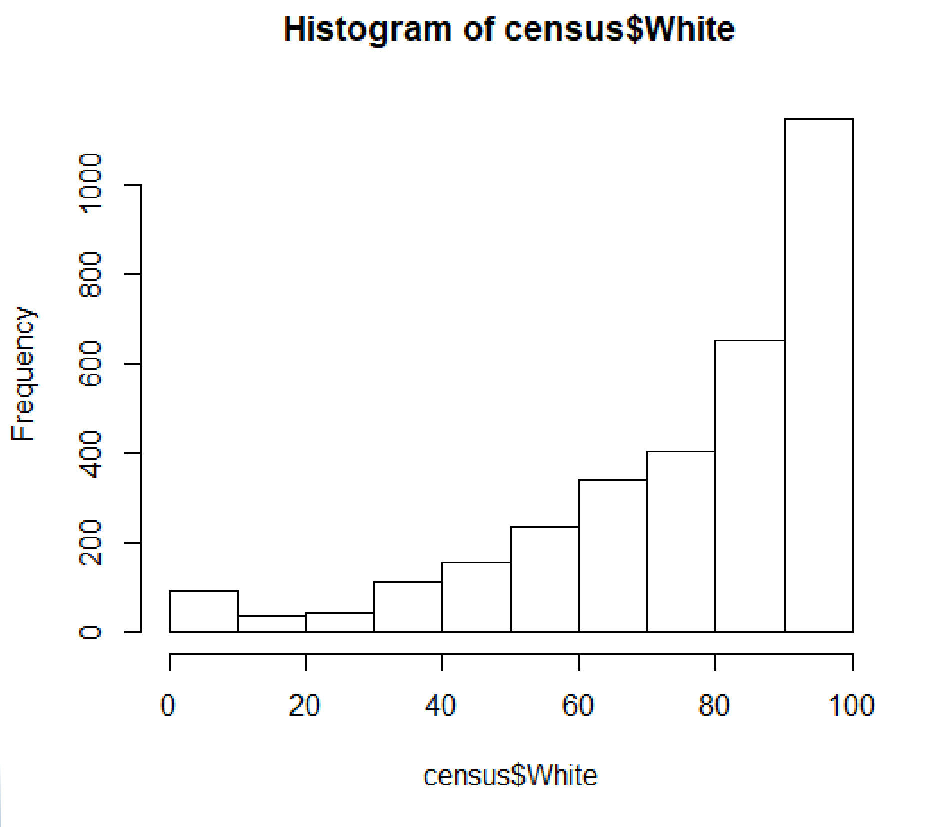
We also can draw boxplot against different States.

```
> census_micro <- subset(census, census$State %in%
unique(census$State)[1:5])
> census_micro$State <- as.factor(as.character(census_micro$State))
> boxplot(census_micro$Hispanic ~ census_micro$State)
```



Getting a Sense of Data Distribution–Histograms

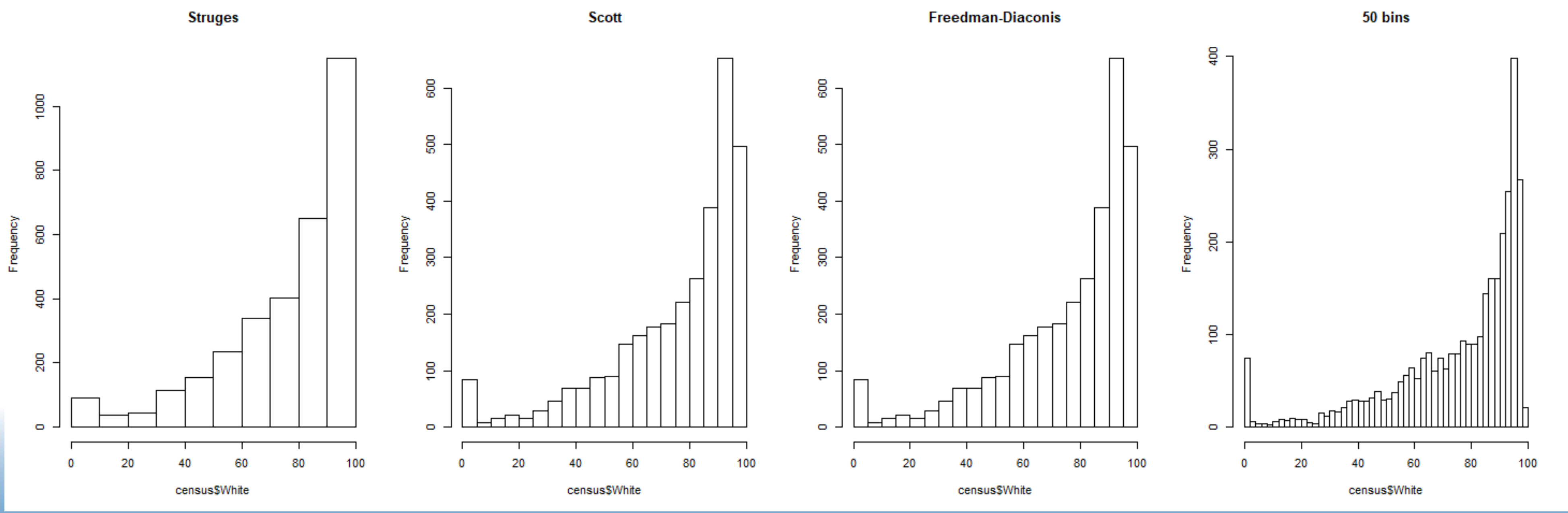
```
> hist(census$White)
```



Getting a Sense of Data Distribution– Histograms

We also can use different break.

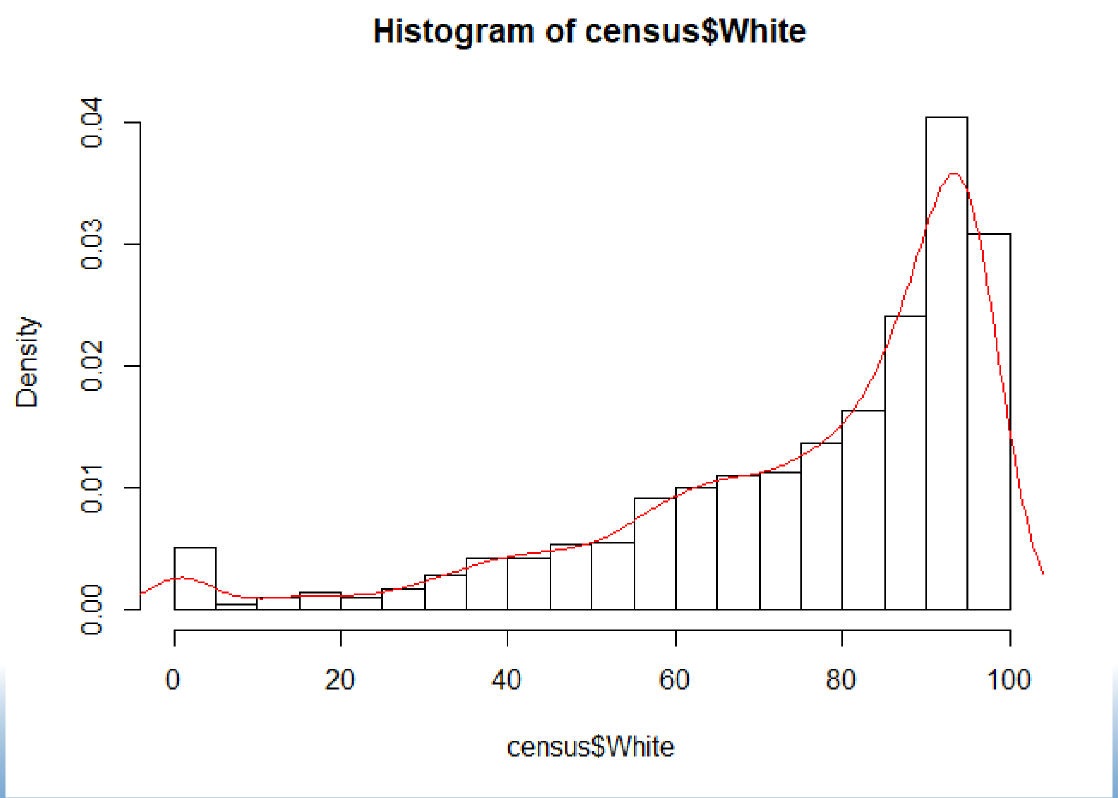
```
> par(mfrow = c(1, 4))
> hist(census$White, breaks = "sturges", main = "Struges")
> hist(census$White, breaks = "scott", main = "Scott")
> hist(census$White, breaks = "fd", main = "Freedman-Diaconis")
> hist(census$White, breaks = 50, main = "50 bins")
```



Getting a Sense of Data Distribution–Histograms

Add density to histogram.

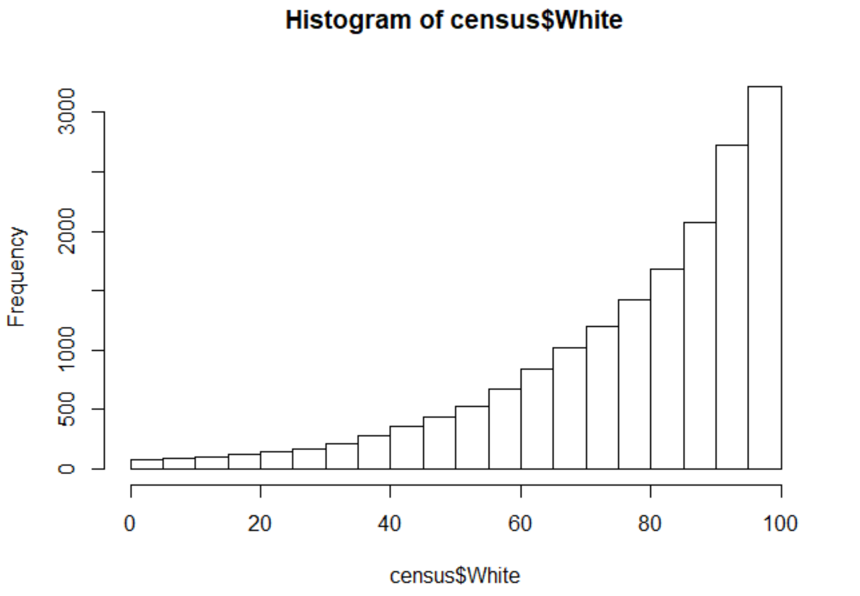
```
> hist(census$White, breaks = "FD", freq = F)
> points(density(census$White), type = "l", col = "red")
```



Getting a Sense of Data Distribution–Histograms

Cumulative Histogram.

```
> cumhist = function(x) {
+   h = hist(census$White, "FD", plot = F)
+   h$counts = cumsum(h$counts)
+   plot(h)
+ }
> cumhist(census$White)
```



Getting a Sense of Data Distribution–
Measuring Data Symmetry Using Skewness and Kurtosis

Skewness, Kurtosis and Gini.

```
> library(moments)
> skewness(census[, 4:13])
 TotalPop       Men     Women  Hispanic     White     Black 
14.287924 14.395679 14.182423  3.216816 -1.431692  2.321071 
   Native     Asian   Pacific   Citizen 
 8.053653  7.169789 37.178879 12.652094 
> kurtosis(census[, 4:13])
   TotalPop         Men       Women    Hispanic       White 
 344.368589  348.934737  339.835909   13.610745    4.667022 
      Black      Native       Asian     Pacific     Citizen 
   8.368248   76.003223   75.057308 1676.754633  277.026915 
> library(reldist)
> gini(census$TotalPop)
[1] 0.7510696
```

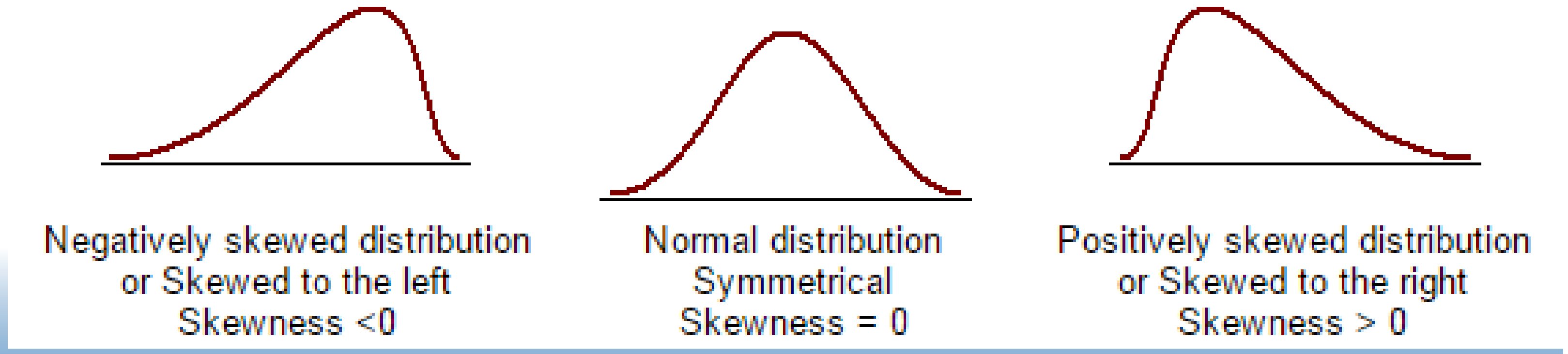
Getting a Sense of Data Distribution–
Measuring Data Symmetry Using Skewness and Kurtosis

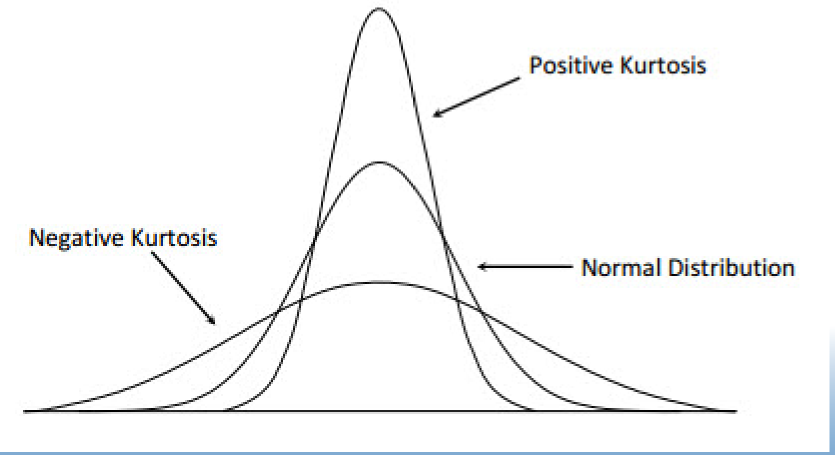
Skewness, Kurtosis and Gini.

Skewness is a measure of symmetry. (https://en.wikipedia.org/wiki/Skewness)
Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. (https://en.wikipedia.org/wiki/Kurtosis)
For univariate data $Y_1, Y_2, \cdots, Y_N$, the formula for skewness and kurtosis are:

$$skewness = \frac{\sum_{i=1}^{N}(\mathrm{Y}_i - \bar{Y})^3/N}{s^3}, \qquad kurtosis = \frac{\sum_{i=1}^{N}(\mathrm{Y}_i - \bar{Y})^4/N}{s^4}$$

where $\bar{Y}$ is the mean, s is the standard deviation, and N is the number of data points.



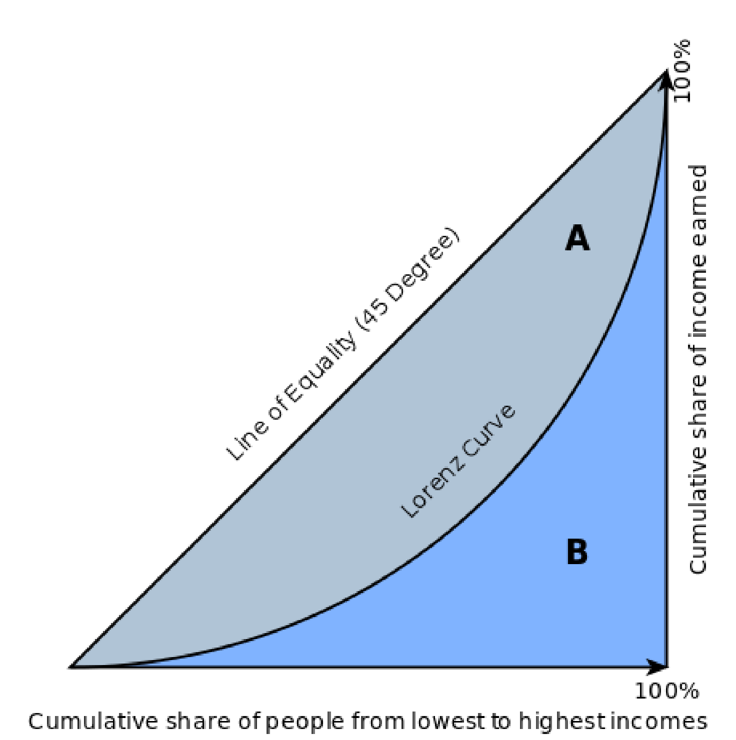


Getting a Sense of Data Distribution–
Measuring Data Symmetry Using Skewness and Kurtosis

Gini coeddicient:

The Gini coefficient measures the inequality among values of a frequency. A Gini coefficient of zero expresses perfect equality, where all values are the same. A Gini coefficient of 1 expresses maximal inequality among values. (https://en.wikipedia.org/wiki/Gini_coefficient)

$$Gini = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2n\sum_{i=1}^{n}x_i}$$



Graphical representation of the Gini coefficient

The graph shows that the Gini coefficient is equal to the area marked *A* divided by the sum of the areas marked *A* and *B*, that is, Gini = *A* / (*A* + *B*). It is also equal to 2*A* and to 1 − 2*B* due to the fact that *A* + *B* = 0.5 (since the axes scale from 0 to 1).

Putting It All Together: Outlier Detection

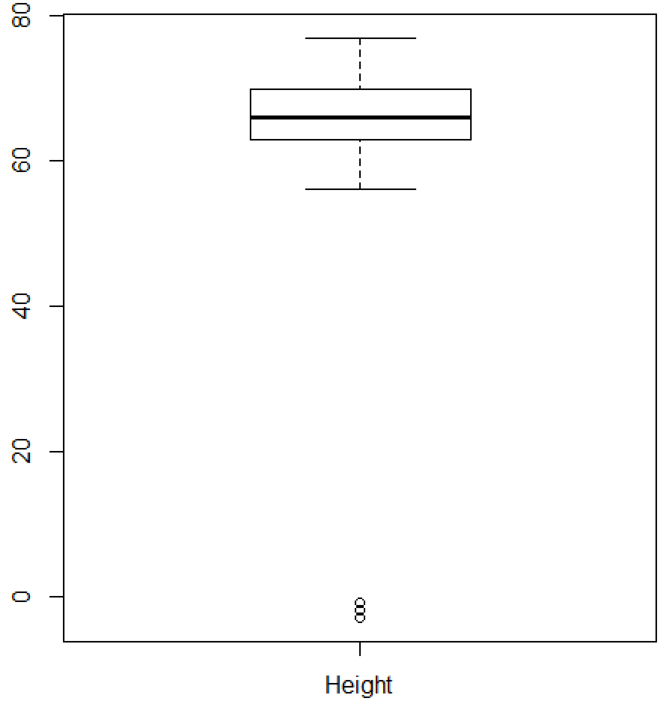
```
> health <- read.csv("ehresp_2014.csv")
> summary(health[, 22:29])
     eusnap           eugenhth          eugroshp            euhgt      
 Min.   :-3.000   Min.   :-3.000   Min.   :-3.000   Min.   :-3.00  
 1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.:63.00  
 Median : 2.000   Median : 2.000   Median : 1.000   Median :66.00  
 Mean   : 1.868   Mean   : 2.477   Mean   : 1.503   Mean   :65.63  
 3rd Qu.: 2.000   3rd Qu.: 3.000   3rd Qu.: 2.000   3rd Qu.:70.00  
 Max.   : 2.000   Max.   : 5.000   Max.   : 3.000   Max.   :77.00  
    euinclvl        euincome2            eumeat             eumilk      
 Min.   :5.000   Min.   :-3.0000   Min.   :-2.0000   Min.   :-3.000  
 1st Qu.:5.000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.000  
 Median :5.000   Median :-1.0000   Median : 1.0000   Median : 2.000  
 Mean   :5.177   Mean   :-0.2313   Mean   : 0.5293   Mean   : 1.158  
 3rd Qu.:5.000   3rd Qu.: 1.0000   3rd Qu.: 1.0000   3rd Qu.: 2.000  
 Max.   :6.000   Max.   : 3.0000   Max.   : 2.0000   Max.   : 2.000  
```

It seems that there is something abnormal in “euhgt”, the min value is -3, but height can’t be negative.

Putting It All Together: Outlier Detection

Beside using `summary()` to observe the data, we also can detect if there is an outlier by drawing some graph.

```
> boxplot(health$euhgt, names = c("Height"), show.names = T)
```



Interactive Visualizations Using Shiny

- ***UI***
  - library(shiny)
  - shinyUI(fluidPage(
  - #fluid page for dynamically adapting to screens of different resolutions.
  - titlePanel("Iris Dataset"),
  - sidebarLayout(
  - sidebarPanel(
  - #implementing radio buttons
  - radioButtons("p", "Select column of iris dataset:",
  - list("Sepal.Length"='a', "Sepal.Width"='b', "Petal.Length"='c', "Petal.Width"='d')),
  - #slider input for bins of histogram
  - sliderInput("bins",
  - "Number of bins:",
  - min = 1,
  - max = 50,
  - value = 30)
  - # Show a plot of the generated distribution
  -),
  - mainPanel(
  - plotOutput("distPlot")
  -)
  -)
  -))

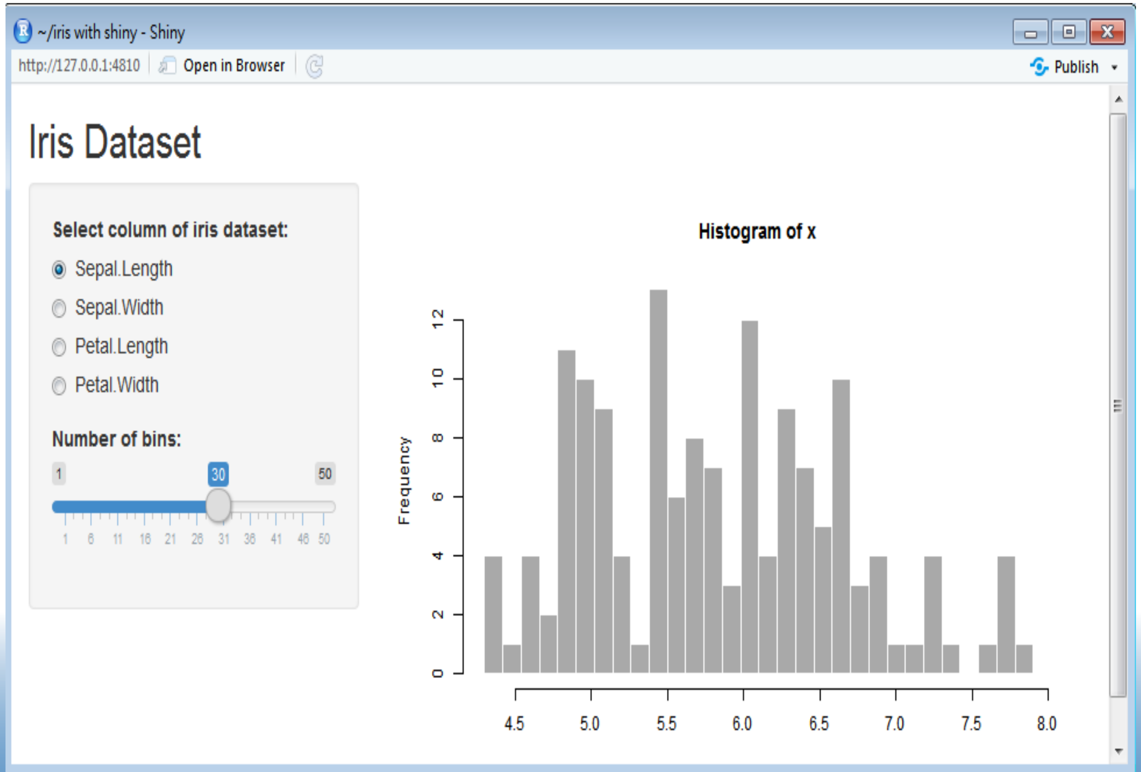
Interactive Visualizations Using Shiny

- ***Server***

```
library(shiny)
#writing server function
shinyServer(function(input, output) {
  #referring output distPlot in ui.r as output$distPlot
  output$distPlot <- renderPlot({
    #referring input p in ui.r as input$p
    if(input$p=='a'){
      i<-1
    }
    if(input$p=='b'){
      i<-2
    }
    if(input$p=='c'){
      i<-3
    }
    if(input$p=='d'){
      i<-4
    }
    x    <- iris[, i]
    #referring input bins in ui.r as input$bins
    bins <- seq(min(x), max(x), length.out = input$bins + 1)
    #producing histogram as output
    hist(x, breaks = bins, col = 'darkgray', border = 'white')
  })
})
```

Interactive Visualizations Using Shiny

- Save to R file UI.R and Server.R into same folder
- Execute the command
  - runApp(“folder path")
  - Ex: runApp("C:/Users/USER/Documents/iris with shiny/")



Demo video

- https://www.youtube.com/watch?v=bVY804VA5ak

3D dynamic plots with iris

- install.packages(c("rgl", "car"))
- library(car)
- attach(iris)
- scatter3d(x = iris$Sepal.Length,
-         y = iris$Sepal.Width,
-         z = iris$Petal.Length)
- scatter3d(x = iris$Sepal.Length,
-         y = iris$Sepal.Width,
-         z = iris$Petal.Length,
-         groups = iris$Species)
- scatter3d(x = iris$Sepal.Length,
-         y = iris$Sepal.Width,
-         z = iris$Petal.Length,
-         groups = iris$Species,
-         surface=FALSE, ellipsoid = TRUE)

Demo video

- https://www.youtube.com/watch?v=6oFg0tuIAxU
- https://www.youtube.com/watch?v=ZQGjJFvDSXY
- https://www.youtube.com/watch?v=gIgkaFAJoGE

Homework

- Basic
  - Find a dataset you want to analysis.
  - Do EDA on this dataset, like summary statistics, box plot and histogram…
  - Detect if there have any outlier in this dataset.
- Advanced
  - If there have any outlier in this dataset, how would you dial with it and why?
  - Give your point of view what you found in this dataset.

Homework 5 (submitted to e3.nctu.edu.tw before Oct 15, 2019)

- Use R and/or other software to visualize the data set with missing data (NA) that you select
- Explain the results you obtain
- Discuss possible problems you plan to investigate for future studies
- Possible source of open data:

  UCI Machine Learning Repository

  (http://archive.ics.uci.edu/ml/datasets.html)

References

1. https://en.wikipedia.org/wiki/Exploratory_data_analysis#Development
2. https://www.kaggle.com/muonneutrino/us-census-demographic-data/data#_=_
3. https://www.kaggle.com/bls/eating-health-module-dataset/data
4. https://en.wikipedia.org/wiki/Box_plot
5. Christie, M. (2001). The Ozone layer: A philosophy of science perspective. United Kingdom: Cambridge University Press. (Chap. 6).
6. Dorfman, R. (1979). A formula for the Gini coefficient. The review of economics and statistics. The Review of Economics and Statistics, 61, 146–149.
7. Most major U.S. cities show population declines. USA Today, June 2011.
8. Size 8 is the new 7: Why our feet are getting bigger. Time Magazine, Oct 2012.
9. Sugary sodas high in diabetes-linked compound. http://abcnews.go.com/Health/Healthday/story?id=4508420&page=1#.UUzdKFt34eF. March 2007.
10. To his credit, charge card king doesn’t cash in. Los Angeles Times, Dec 2004.