Exploratory Data Analysis

Lecture Content

- Exploratory Data Analysis
  - Visualization before Analysis,
  - Dirty Data,
  - Examining Multiple Variables

Exploratory Data Analysis

- In previous slides we learned about importing and exporting data in R, basic data types and generating descriptive statistics
  - As an example we learned how *summary()* can help analysts easily get an idea of the magnitude and range of the data
- But other aspects like linear relationship and distribution are more difficult to see from descriptive statistics
  - For example the following R code shows a summary view of a data frame data with two columns **x** and **y**
  - The output of R code below shows the range of **x** and **y** but it is not clear what the relationship may be between these two variable
- Thus the coming slides will cover this aspect of **Exploratory Data Analysis**

Creation of data Frame named **data**

```r
# Figure 3-5
x <- rnorm(50)
y <- x + rnorm(50, mean=0, sd=0.5)

data <- as.data.frame(cbind(x, y))
summary(data)
```

Console C:/Users/Z9701/Desktop/aaTeaching/CTI Courses/Fall 2016/CTI 466 Data Analytics/CIT466 R Exercises/

```
> x <- rnorm(50)
> y <- x + rnorm(50, mean=0, sd=0.5)
>
> data <- as.data.frame(cbind(x, y))
> summary(data)
       x                 y
 Min.   :-1.6419   Min.   :-1.6932
 1st Qu.:-0.3646   1st Qu.:-0.1701
 Median : 0.2359   Median : 0.2767
 Mean   : 0.2384   Mean   : 0.3066
 3rd Qu.: 0.7582   3rd Qu.: 0.9813
 Max.   : 1.8031   Max.   : 2.0512
```

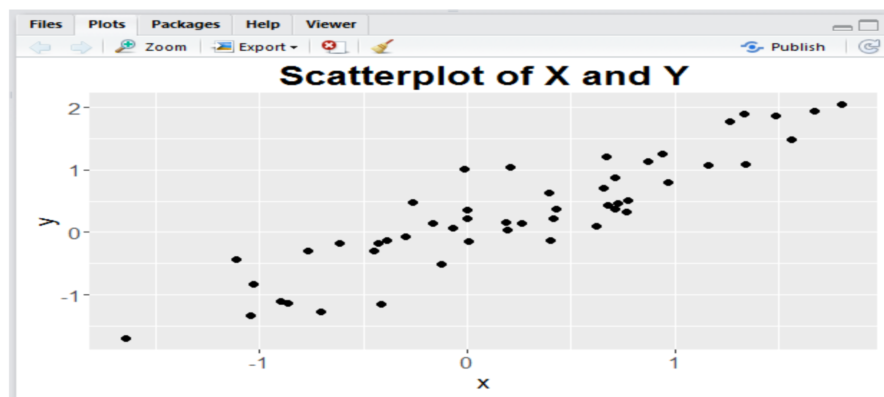
Exploratory Data Analysis

Visualization Before Analysis

Exploratory Data Analysis with Visualization

- Visualization gives a succinct , holistic view of the data that may be difficult to grasp from the numbers and summaries alone
- The R code below visualize in scatterplot the relationship between the variables **x** and **y** taken from data frame *data*
- *A scatterplot can easily show if x and y share a relation*

```r
library(ggplot2)
ggplot(data, aes(x=x, y=y)) +
  geom_point(size=2) +
  ggtitle("Scatterplot of X and Y") +
  theme(axis.text=element_text(size=12),
        axis.title = element_text(size=14),
        plot.title = element_text(size=20, face="bold"))
```



Visualization Before Analysis

- Consider the quartet which consists four data sets constructed by the statistician Francis Anscombe in 1973 to demonstrate the importance of graphs in statistical analysis
- As the nearly identical statistical properties across each data set, one might conclude these four data sets are quite similar. However the scatterplot in the next slide tells different story

Anscombe's Quartet

I		II		III		IV	
x	y	x	y	x	y	x	y
4	4.26	4	3.1	4	5.39	8	5.25
5	5.68	5	4.74	5	5.73	8	5.56
6	7.24	6	6.13	6	6.08	8	5.76
7	4.82	7	7.26	7	6.42	8	6.58
8	6.95	8	8.14	8	6.77	8	6.89
9	8.81	9	8.77	9	7.11	8	7.04
10	8.04	10	9.14	10	7.46	8	7.71
11	8.33	11	9.26	11	7.81	8	7.91
12	10.8	12	9.13	12	8.15	8	8.47
13	7.58	13	8.74	13	12.7	8	8.84
14	9.96	14	8.1	14	8.84	19	12.5

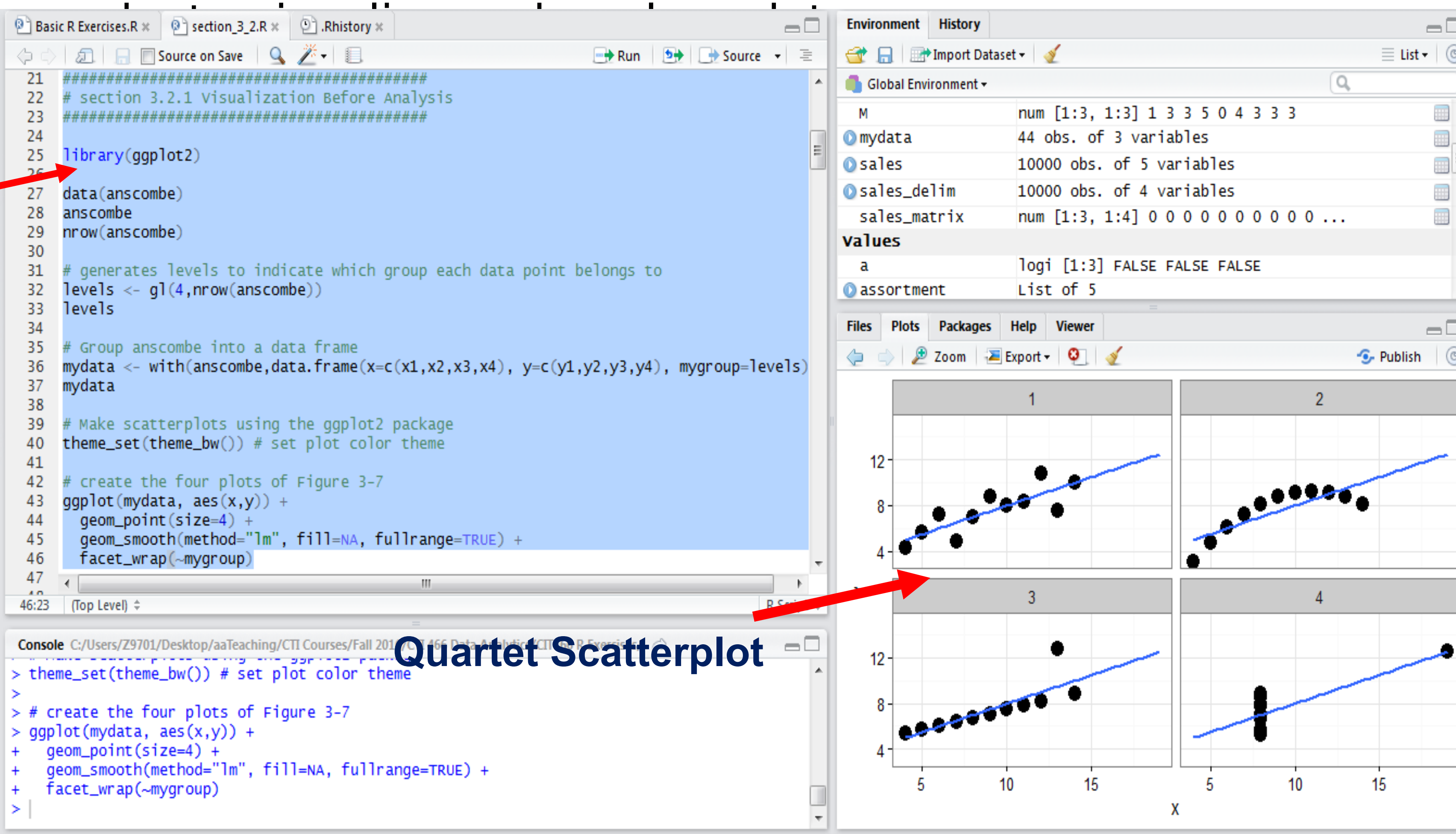
Statistical Properties of Anscombe's Quartet

Statistical Properties	Values
Mean of x	9
Variance of y	11
Mean of y	7.50 (to 2 decimal points)
Variance of y	4.12 or 4.13 (to 2 decimal points)
Correlation between x and y	0.816
Linear regression line	Y=3.00+0.50*x (to 2 decimal point

Anscombe Quartet Visualized as Scatterplot

Although the quartet data sets have similar statistical properties BUT when plotting them they are completely different. That is why need

R Code



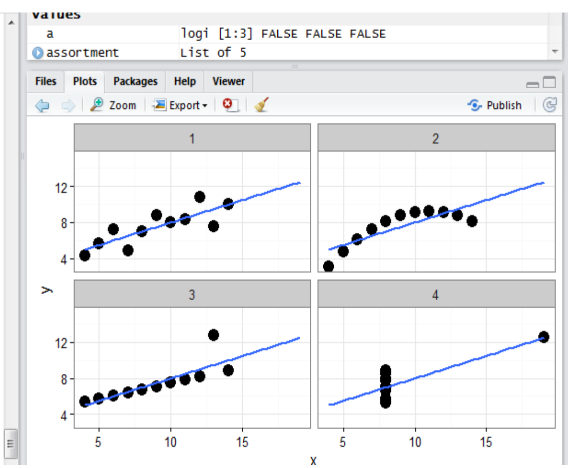
Intermediate Results of R Code

Below the intermediate R code results and scatterplot (see text book page 84 for more explanation of the code)

```
Console  C:/Users/Z9701/Desktop/aaTeaching/CTI Courses/Fall 2016/CTI 466 Data Analytics/CIT466 R Exercises/
> library(ggplot2)
>
> data(anscombe)
> anscombe
   x1 x2 x3 x4    y1   y2    y3    y4
1  10 10 10  8  8.04 9.14  7.46  6.58
2   8  8  8  8  6.95 8.14  6.77  5.76
3  13 13 13  8  7.58 8.74 12.74  7.71
4   9  9  9  8  8.81 8.77  7.11  8.84
5  11 11 11  8  8.33 9.26  7.81  8.47
6  14 14 14  8  9.96 8.10  8.84  7.04
7   6  6  6  8  7.24 6.13  6.08  5.25
8   4  4  4 19  4.26 3.10  5.39 12.50
9  12 12 12  8 10.84 9.13  8.15  5.56
10  7  7  7  8  4.82 7.26  6.42  7.91
11  5  5  5  8  5.68 4.74  5.73  6.89
> nrow(anscombe)
[1] 11
>
> # generates levels to indicate which group each data point belongs to
> levels <- gl(4,nrow(anscombe))
> levels
 [1] 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4
[41] 4 4 4 4
Levels: 1 2 3 4
```

```
> # Group anscombe into a data frame
> mydata <- with(anscombe,data.frame(x=c(x1,x2,x3,x4), y=c(y1,y2,y3,y4), mygroup=levels))
> mydata
    x     y mygroup
1  10  8.04       1
2   8  6.95       1
3  13  7.58       1
4   9  8.81       1
5  11  8.33       1
6  14  9.96       1
7   6  7.24       1
8   4  4.26       1
9  12 10.84       1
10  7  4.82       1
11  5  5.68       1
12 10  9.14       2
13  8  8.14       2
14 13  8.74       2
15  9  8.77       2
16 11  9.26       2
17 14  8.10       2
18  6  6.13       2
19  4  3.10       2
20 12  9.13       2
21  7  7.26       2
22  5  4.74       2
23 10  7.46       3
24  8  6.77       3
```

```
25 13 12.74       3
26  9  7.11       3
27 11  7.81       3
28 14  8.84       3
29  6  6.08       3
30  4  5.39       3
31 12  8.15       3
32  7  6.42       3
33  5  5.73       3
34  8  6.58       4
35  8  5.76       4
36  8  7.71       4
37  8  8.84       4
38  8  8.47       4
39  8  7.04       4
40  8  5.25       4
41 19 12.50       4
42  8  5.56       4
43  8  7.91       4
44  8  6.89       4
>
> # Make scatterplots using the ggplot2 package
> theme_set(theme_bw()) # set plot color theme
>
> # create the four plots of Figure 3-7
> ggplot(mydata, aes(x,y)) +
+   geom_point(size=4) +
+   geom_smooth(method="lm", fill=NA, fullrange=TRUE) +
+   facet_wrap(~mygroup)
>
```



Dirty Data

- Dirty data refers to data that contains erroneous information.
- Data analyst should look for anomalies, verify the data with the domain knowledge, and decide appropriate approach to *clean the data*
- Visualization in the scatterplot below shows anomalies where a large number of account holders with negative, zero and less than 10 years of ages? Therefore *data cleaning* should be performed over accounts with abnormal age values.

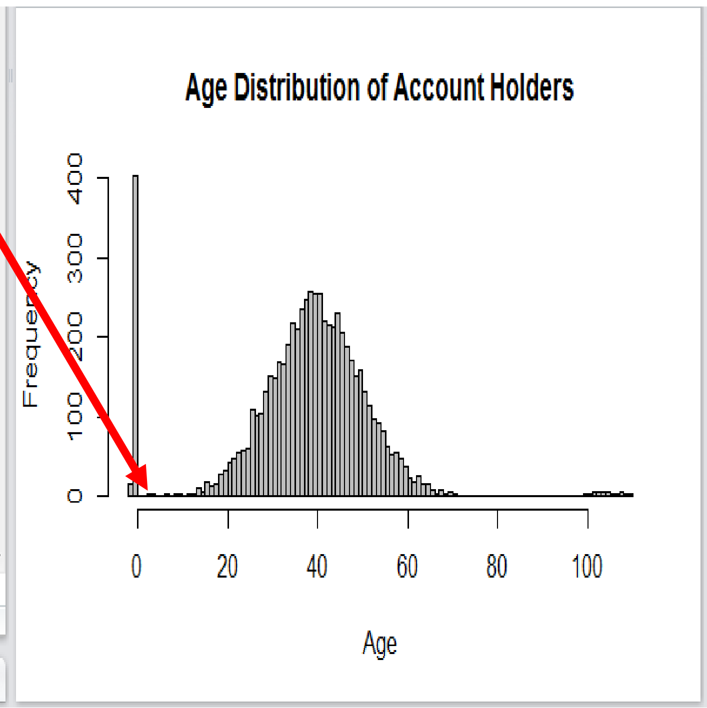
```r
##########################################
# section 3.2.2 Dirty Data
##########################################

age <- rnorm(6000, mean=40, sd=10)
age <- c( age, runif(20, min=-2, max=0),
          rep(0,400),
          runif(40, min=100, max=110))
age <- round(age)

hist(age, breaks=100, main="Age Distribution of Account Holders",
     xlab="Age", ylab="Frequency", col="gray")
```

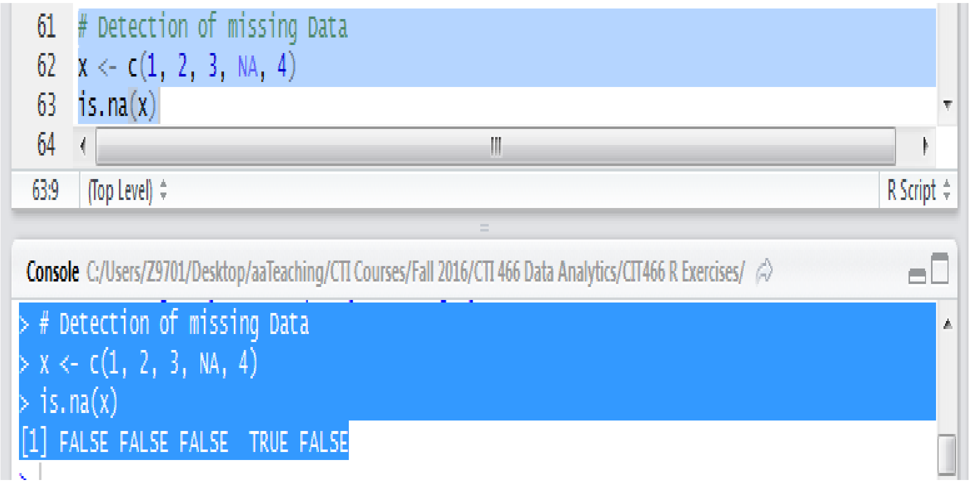
66:1 (Top Level) R Script

Console



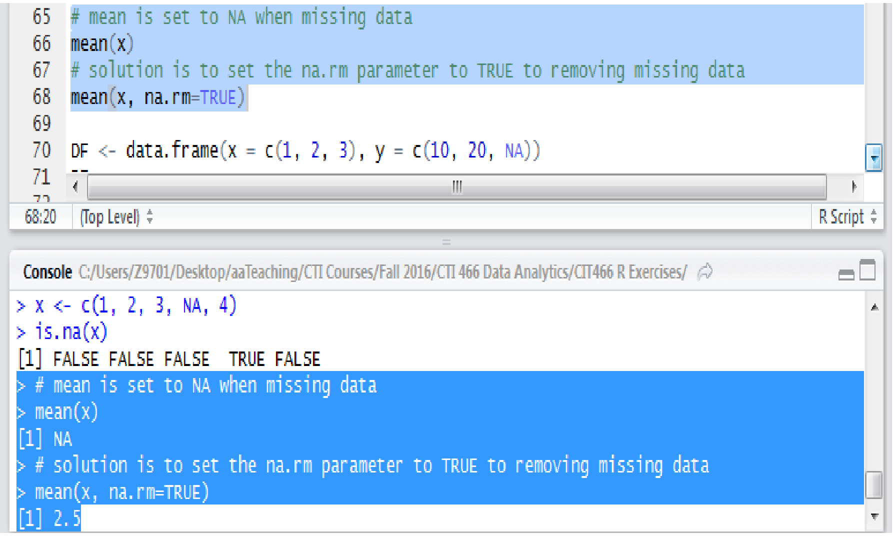
Missing data

- In statistics, **missing data**, or **missing values**, occur when no data value is stored for the variable in a data set.
- In R the *is.na()* function provide tests for missing data values,
- The following example creates a vector **x** where the fourth value is not available (NA) and by using *is.na()* function it returns TRUE at each NA value and FALSE otherwise



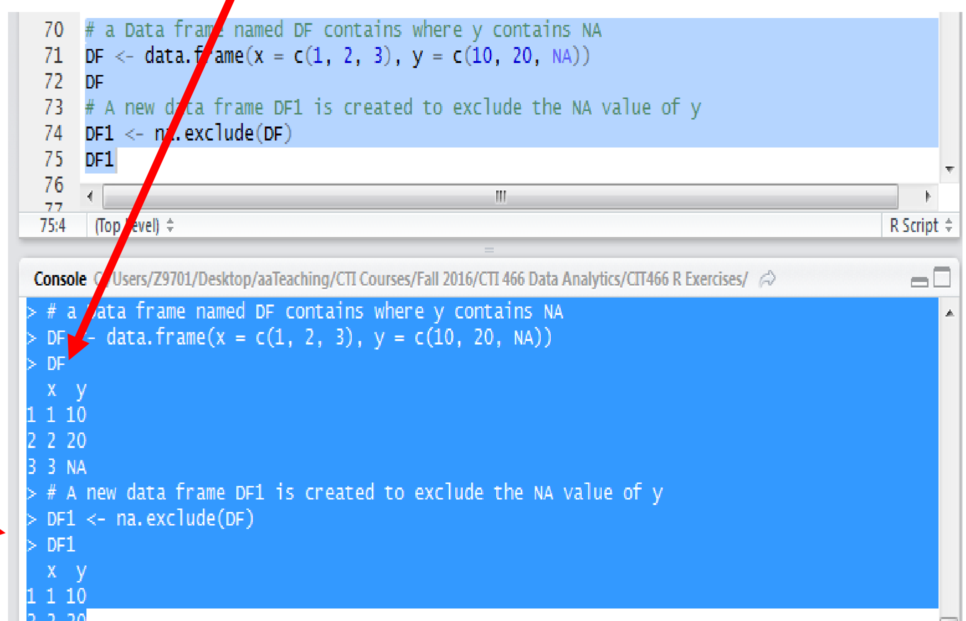
Solutions to avoid impact of missing data

- Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.
- Some arithmetic function such as *means()* applied to data containing missing values can yield to NA results
- To prevent this, it is advised to set *na.rm* parameter to TRUE to remove the missing value during the arithmetic function



More on avoiding impact of missing data

- Another solution provide by R is the *na.exclude()* function
- This function return the object/data set with missing/incomplete data cases completely removed
- In the R code below DF data frame containing NA is replaced by DF1 from which the NA is excluded

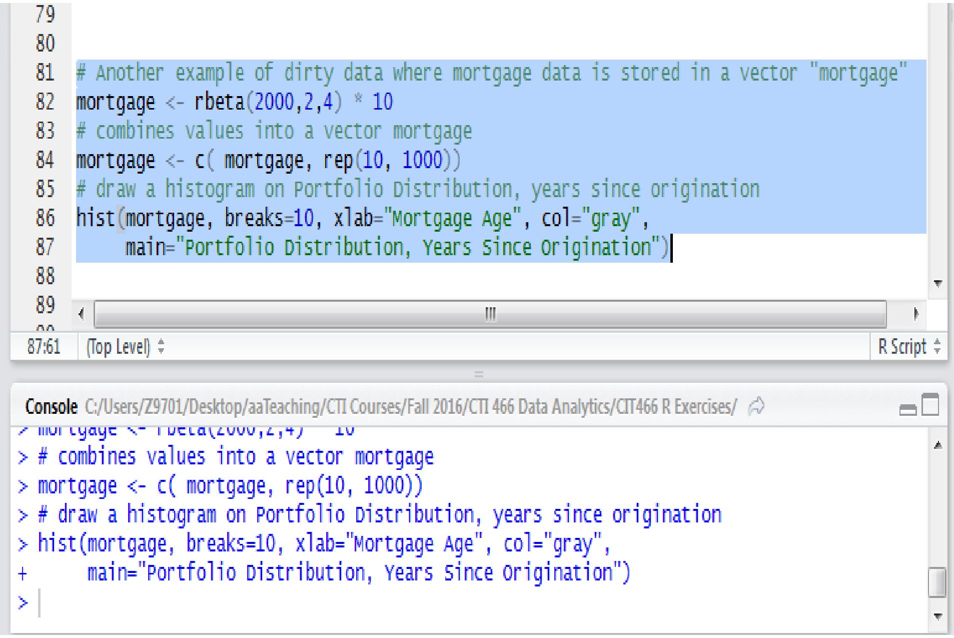


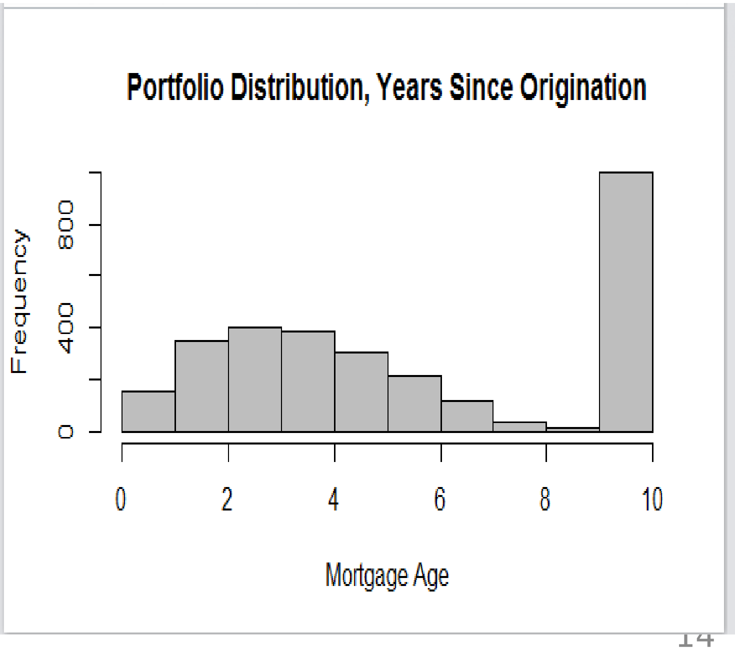
More on missing data Solution

- Another example of missing data could happen with account holder data set where there are account holder older than 100 years.
- This could be caused by
  - Typos in writing account holders ages,
  - Passed to the heirs of the original account holder but not updated,
  - Etc.
- Solutions:
  - Further examine the data and conduct cleansing if necessary,
  - Could be simply removed,
  - If removal is not possible, the data analyst can look for pattern within data and develop a set of heuristic to tackle the problem of dirty data,
  - Wrong age could be replaced with approximation based on the nearest neighbor/similar account holder
  - Investigate the different exiting mathematical and statistical techniques for missing data e.g. List wise & Pairwise deletion, Single Imputation Methods, Model-Based Methods,

Example of Dirty Data

- Histogram below shows that there are no more loans than 10 years old, and these 10 years-old loans have a disproportionate frequency compared to the rest of the population, one possible explanation is that the 10 years-old loans do not include only loans originated 10 years. In other words, the 10 in the x-axis actually means ≥10.
- Dirty data can occur due to acts of omission, like in sales data used in previous example as the minimum number of orders is fixed to 1 and sales amount was fixed to $30.02. So there is a strong possibility the data set does not include all customers





Exploratory Data Analysis

Visualizing a Single Variable

Visualizing a Single Variable

- Using visual representation is a hallmark of exploratory analysis: letting data explain itself its audience rather an interpretation of the data a priori
- R has many functions available to examine a single variable. Some of them are listed below

Function	Purpose
Plot (data)	Scatterplot where x is the index and y is the value; Suitable for low volume data
barplot (data)	Bar plot with vertical or horizontal bars
dotchart (data)	Cleveland dot plot
hist (data)	Histogram
plot(density (data))	Density plot (a continuous histogram)
stem (data)	Stem-and-leaf plot
rug(data)	Add a rug representation (1-d plot) of the data to an existing plot

R Code & Drawing of Dotchart

- Dotchart portray continuous values with labels from a discrete variable,
- Dotchart is drawn using *dotchart(x, label=...)* function where x is a numeric vector and label is a vector of categorical labels for x,

R Code

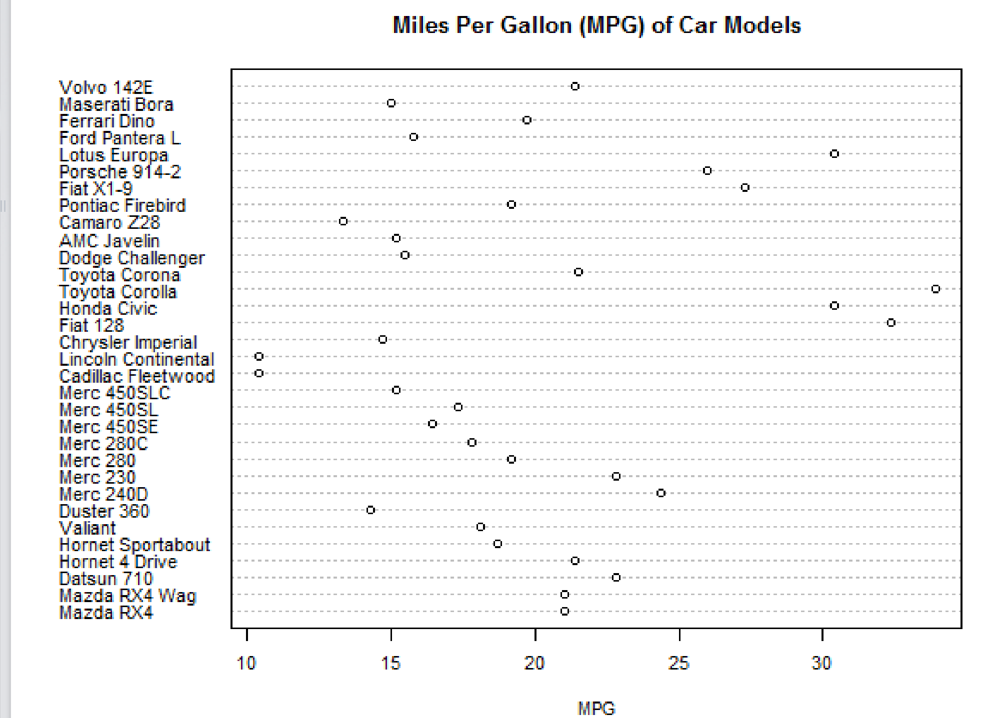
```
###########################################
# section 3.2.3 Visualizing a Single Variable
###########################################
data(mtcars)

## Dotchart and Barplot ##

dotchart(mtcars$mpg,labels=row.names(mtcars),cex=.7,
         main="Miles Per Gallon (MPG) of Car Models",
         xlab="MPG")
```

mtcars dataset comes with R distribution

Dotchart



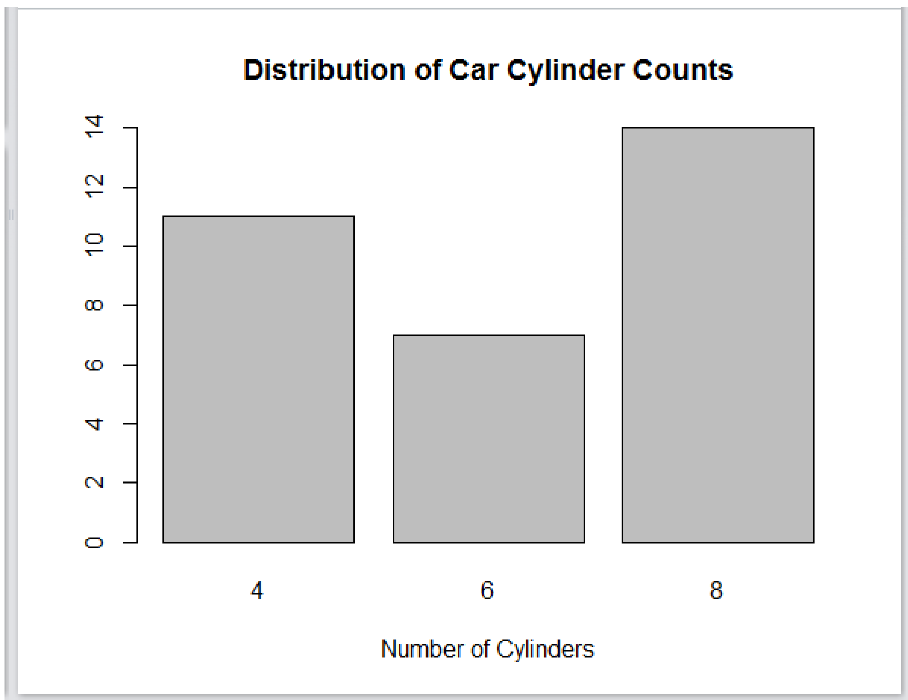
R Code and drawing of Barplot

- Barplot portray continuous values with labels from a discrete variable,
- A barplot is drawn using *barplot(height)* function where height represents a vector or matrix.

R Code

```
data(mtcars)
## Barplot ##
barplot(table(mtcars$cyl), main="Distribution of Car Cylinder Counts",
        xlab="Number of Cylinders")
```

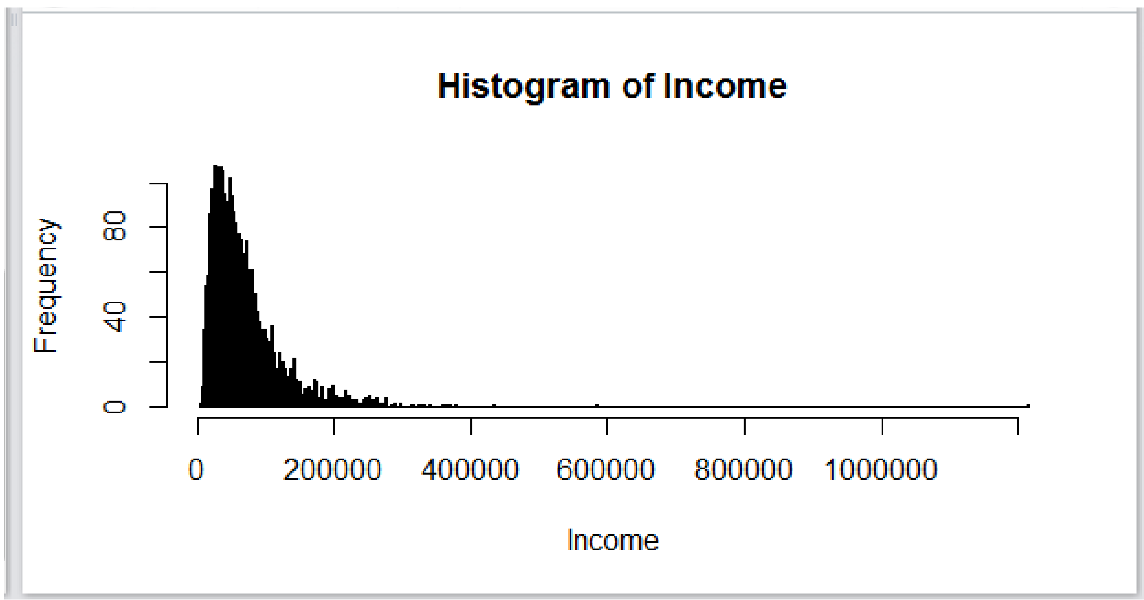
Barplot



Histogram

- The histogram shows a clear concentration of how household income on the left and the long tail of the higher incomes on the right.

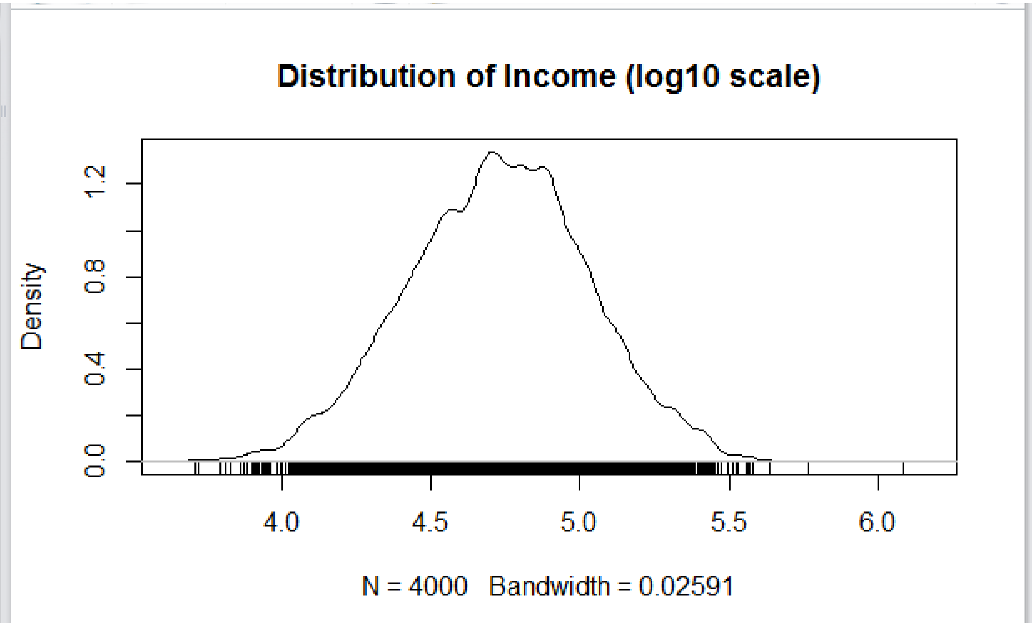
```
# Histogram
# randomly generate 4000 observations from the log normal distribution
income <- rlnorm(4000, meanlog = 4, sdlog = 0.7)
summary(income)
income <- 1000*income
summary(income)
# plot the histogram
hist(income, breaks=500, xlab="Income", main="Histogram of Income")
```



Density Plot

- Below R code and a Density Plot of the logarithm of household income values, which emphasizes the distribution
- The income distribution is concentrated in the center portion of the graph
- The rug() function creates a one dimension density plot on the bottom of the graph to emphasize the distribution of the observation

```r
# density plot
plot(density(log10(income), adjust=0.5),
     main="Distribution of Income (log10 scale)")
# add rug to the density plot
rug(log10(income))
```



Recommendation for Data preparation

- The data analyst should look for signs of dirty data,
- Examine if the data is unimodal or multi-model to give an idea how many distinct population with different behavior might be mixed into an overall population,
- Many modelling techniques assume that data follows a normal distribution, therefore it is important to know if the available dataset can match that assumption before applying any of those modelling techniques,
- Sometimes it is useful to plot graph using the logarithm of the initial data can detect structural that might be overlooked by a graph with a regular, non-logarithmic scale- See Example in next slides

Examining Multiple Variables

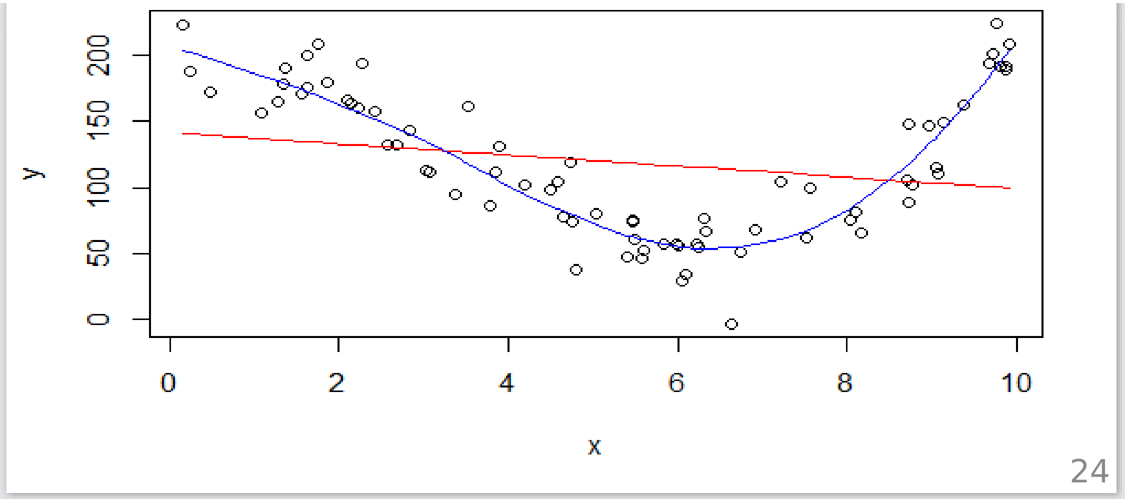
Examining Multiple Variables

- Scatterplot is simple and widely used visualization for finding the relationship among multiple variables
- It can represent data with up to five variables using x-axis, y-axis, size, color and shape
  - But usually only two to four variable are represented to avoid confusion
- When examining scatterplot, one needs to give attention to the possible relationship between variables
- If the functional relationship between variable is well pronounced, then the data may lie on straight line, parabola or exponential curve,
- If the variable y is related exponentially to x, then the plot x versus log y is approximately linear,
- If the plot looks more like a cluster without a pattern, the corresponding variables may have a weak relationship.

Example of examining two variables

R code that uses alternative visualizations to find accurate relationship between two variables x & y

```r
########################################
# section 3.2.4 Examining Multiple Variables
########################################
# runif generates 75 numbers between 0 and 10 of uniform distribution
# with random deviates
x <- runif(75, 0, 10)
x <- sort(x)
# rnorm (75,0,20) generates 75 number that conforma to normal distrbution with
# means equal Zero and deviation equal 20
y <- 200 + x^3 - 10 * x^2 + x + rnorm(75, 0, 20)
lr <- lm(y ~ x) # linear regression
poly <- loess(y ~ x) # LOESS is used to fit a nonlinear line to the data
fit <- predict(poly) # fit a nonlinear line
plot(x,y)
# points drwas a sequence of points at specified coordinate,here it draws
# the fitted line for the linear regression with type=1 means solid line
points(x, lr$coefficients[1] + lr$coefficients[2] * x, type = "l", col = 2)
# draw the fitted line with LOESS
points(x, fit, type = "l", col = 4)
```



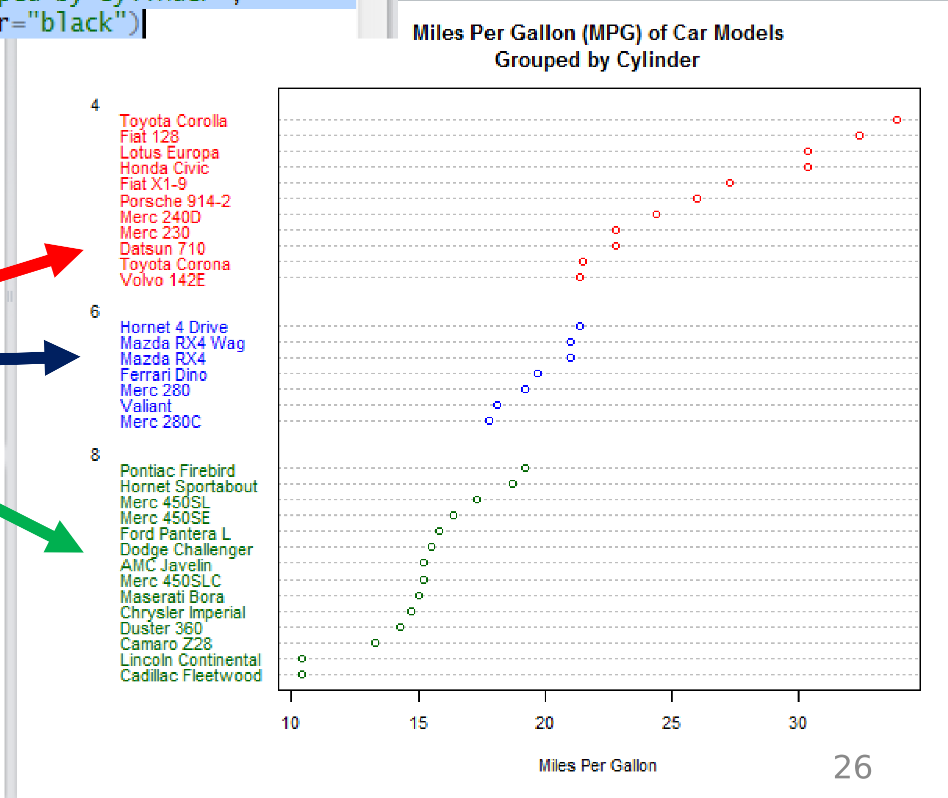
Dotchart & Barplot

- Dotchart and bar plot can visualize multiple variables.
- Both use color as an additional dimension for visualizing the data
- Below an example of Dotchart that plot **mtcars** dataset using color to distinguish between different cylinders

Dotchart for Grouping variables

```r
## Dotchart and Barplot ##
# Dotchart
# sort by mpg
cars <- mtcars[order(mtcars$mpg),]
# grouping car cylinder "cyl" variable must be a factor
cars$cyl <- factor(cars$cyl)
cars$color[cars$cyl==4] <- "red"
cars$color[cars$cyl==6] <- "blue"
cars$color[cars$cyl==8] <- "darkgreen"
dotchart(cars$mpg, labels=row.names(cars), cex=.7, groups= cars$cyl,
         main="Miles Per Gallon (MPG) of Car Models\nGrouped by Cylinder",
         xlab="Miles Per Gallon", color=cars$color, gcolor="black")
```

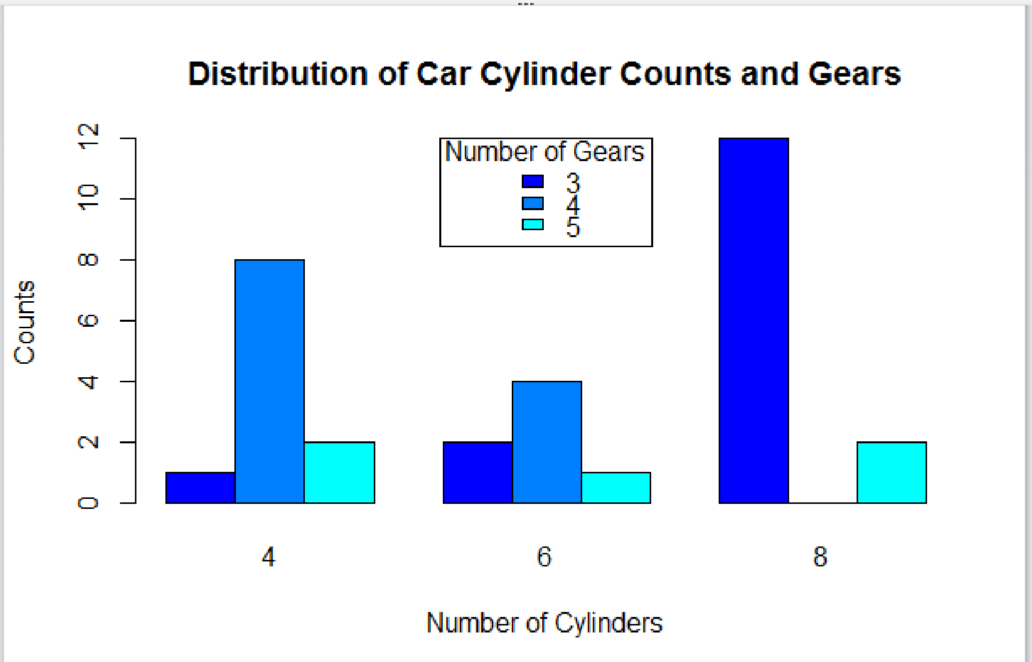
- Dotchart groups vehicle cylinders at the y-axis and use color to distinguish different cylinders
- The vehicles are sorted according to their MPG values



Barplot Example

- Barplot visualizes the distribution of car cylinders counts and number of gears
- The x-axis represent the number of cylinders and the color represent the number of gears

```r
# Barplot
counts <- table(mtcars$gear, mtcars$cyl)
barplot(counts, main="Distribution of Car Cylinder Counts and Gears",
        xlab="Number of Cylinders", ylab="Counts",
        col=c("#0000FFFF", "#0080FFFF", "#00FFFFFF"),
        legend = rownames(counts), beside=TRUE,
        args.legend = list(x="top", title = "Number of Gears"))
```



Box-and-Whisker Plot

- Box-and-Whisker Plot shows distribution of a continuous variables for each value of a discrete variable
- The example in the next slide visualize mean household income as a function of region in the US.
  - The first digit of the US postal (“ZIP”) code correspond to a geographical region in the US,
  - Each data point corresponds to the means household income from a particular zip code,
  - The horizontal axis represents the first digital code, ranging from 0 to 9,
  - The vertical axis represents the logarithm of mean household income.

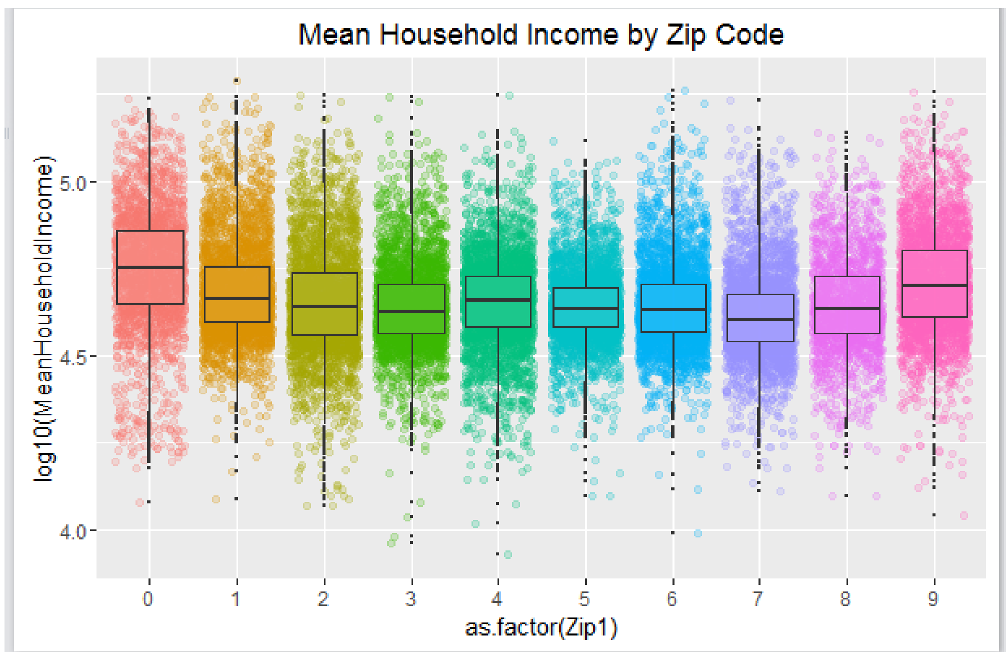
Box-and-Whisker Plot

R Code for Drawing BOX-and-Whisker plot below

```r
## Box-and-Whisker Plot ##
DF <- read.csv("c:/data/zipIncome.csv", header=TRUE, sep=",")
# Remove outliers
DF <- subset(DF, DF$MeanHouseholdIncome > 7000 & DF$MeanHouseholdIncome < 200000)
summary(DF)

library(ggplot2)
# plot the jittered scatterplot w/ boxplot
# color-code points with zip codes
# the outlier.size=0 prevents the boxplot from plotting the outlier
ggplot(data=DF, aes(x=as.factor(Zip1), y=log10(MeanHouseholdIncome)) +
  geom_point(aes(color=factor(Zip1)), alpha=0.2, position="jitter") +
  geom_boxplot(outlier.size=0, alpha=0.1) +
  guides(colour=FALSE) +
  ggtitle ("Mean Household Income by Zip Code")
```

- The scatterplot is displayed beneath the box-and-whisker
- The box of the box-and-whisker contains the central 50% of the data and the line inside the box is the location of the median value
- The upper and lower hinges of the box correspond to the first and third quartiles

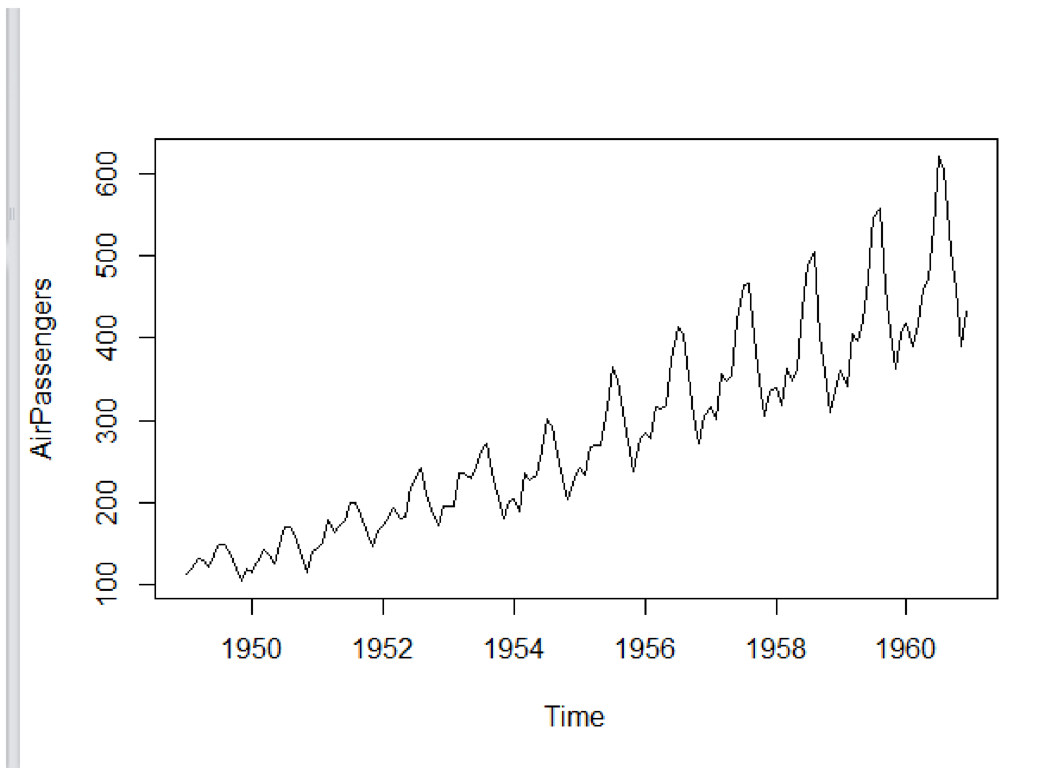


Analyzing a variable overtime

- Visualizing a variable over time is the same as visualizing any pair of variables, but in this case the goal is to identify time-specific patterns
- Example below show air passengers over time

R Code

```
## Analyzing a Variable over Time ##
plot(AirPassengers)
```



Observations:
•Large picks occur mid Year around July–August Possibly due to holiday
•Such phenomenon is Referred as seasonality effect which can be tackled using Time Series techniques.