STAT3612_2312 Data Mining 1st Tutorial

Jason J. You

1/23/2017

Introduction to Data Format in R

Vector is a way to hold multiple values in a variable.

```
vec1=c(1,2,3)
vec2=c(4,5,6)
vec3=c(7,8,9)
vec4=c(vec1,vec2,vec3)
vec1
```

```
## [1] 1 2 3
```

```
vec2
```

```
## [1] 4 5 6
```

```
vec3
```

```
## [1] 7 8 9
```

```
vec4
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

Most data in statistics is in a "row-by-column" format, which can be represented by a matrix. We use cbind() and rbind() functions to bind either rows or columns together

```
mat1 = cbind(vec1, vec2)
mat2 = rbind(vec3, vec1, vec2)
mat1
```

```
##      vec1 vec2
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```
mat2
```

```
##      [,1] [,2] [,3]
## vec3    7    8    9
## vec1    1    2    3
## vec2    4    5    6
```

Another way to build a matrix is simply using {matrix} function. Columns, rows or individual elements can be extracted with bracket notation, as shown below.

```
mat3 = matrix(vec4, nrow=3, ncol=3)
mat3
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
t(mat3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
mat3[,1]
```

```
## [1] 1 2 3
```

```
mat3[3,]
```

```
## [1] 3 6 9
```

In addition to the row by column format, most statistical datasets have a mix of continuous and categorical variables. To handle this in R, we need a data frame.

```
nrow = 12

# Generate a numeric variable with normal distribution
x1 = rnorm(n = nrow, mean = 0, sd=1)

# Generate a factor variable with 3 categories
fac1 = as.factor(rep(c("a","b","c"),each=4))

# Generate a data frame
df1 = data.frame(value = x1,category = fac1)

df1
```

```
##          value category
## 1   0.48291168        a
## 2  -0.88274429        a
## 3   0.03420341        a
## 4   1.93572027        a
## 5  -0.97795676        b
```

```
## 6  -1.17272544        b
## 7   0.20980238        b
## 8  -1.18376344        b
## 9   1.67036375        c
## 10  0.40038935        c
## 11  0.28810640        c
## 12  1.68012779        c
```

```
# Look at the class of the attributes
class(df1$value)
```

```
## [1] "numeric"
```

```
class(df1$category)
```

```
## [1] "factor"
```

Read & Write Data Files in R

Function {getwd()} will help you get the current working directory

Function {setwd('/AimedWorkingDirectory')} will help you set the working directory

```
getwd()
```

```
## [1] "/Users/Jason/Desktop"
```

```
setwd("/Users/Jason/Desktop")
getwd()
```

```
## [1] "/Users/Jason/Desktop"
```

There are bunches of datafile types, the most frequent datafile format is ".csv" and ".txt". The following are examples to write your data.frame file into ".csv" and ".txt", respectively.

```
write.csv(df1, file = "MyData.csv", row.names = FALSE)
write.table(df1, file = "MyData.txt", sep=",", col.names=TRUE)
```

Load your files into R

```
read.csv("MyData.csv")
```

```
##          value category
## 1   0.48291168        a
## 2  -0.88274429        a
## 3   0.03420341        a
## 4   1.93572027        a
## 5  -0.97795676        b
## 6  -1.17272544        b
## 7   0.20980238        b
## 8  -1.18376344        b
## 9   1.67036375        c
## 10  0.40038935        c
## 11  0.28810640        c
## 12  1.68012779        c
```

```
read.table("MyData.txt", sep=",")
```

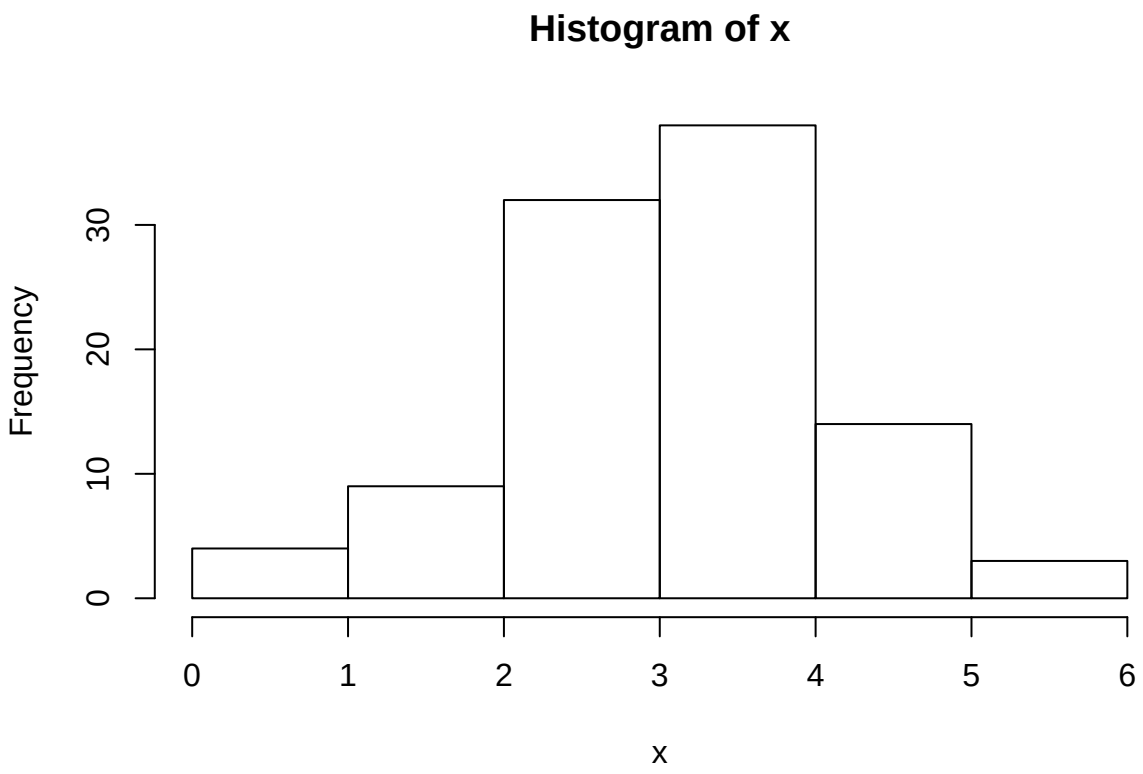
```
##          value category
## 1   0.48291168        a
## 2  -0.88274429        a
## 3   0.03420341        a
## 4   1.93572027        a
## 5  -0.97795676        b
## 6  -1.17272544        b
## 7   0.20980238        b
## 8  -1.18376344        b
## 9   1.67036375        c
## 10  0.40038935        c
## 11  0.28810640        c
## 12  1.68012779        c
```

Data Generation

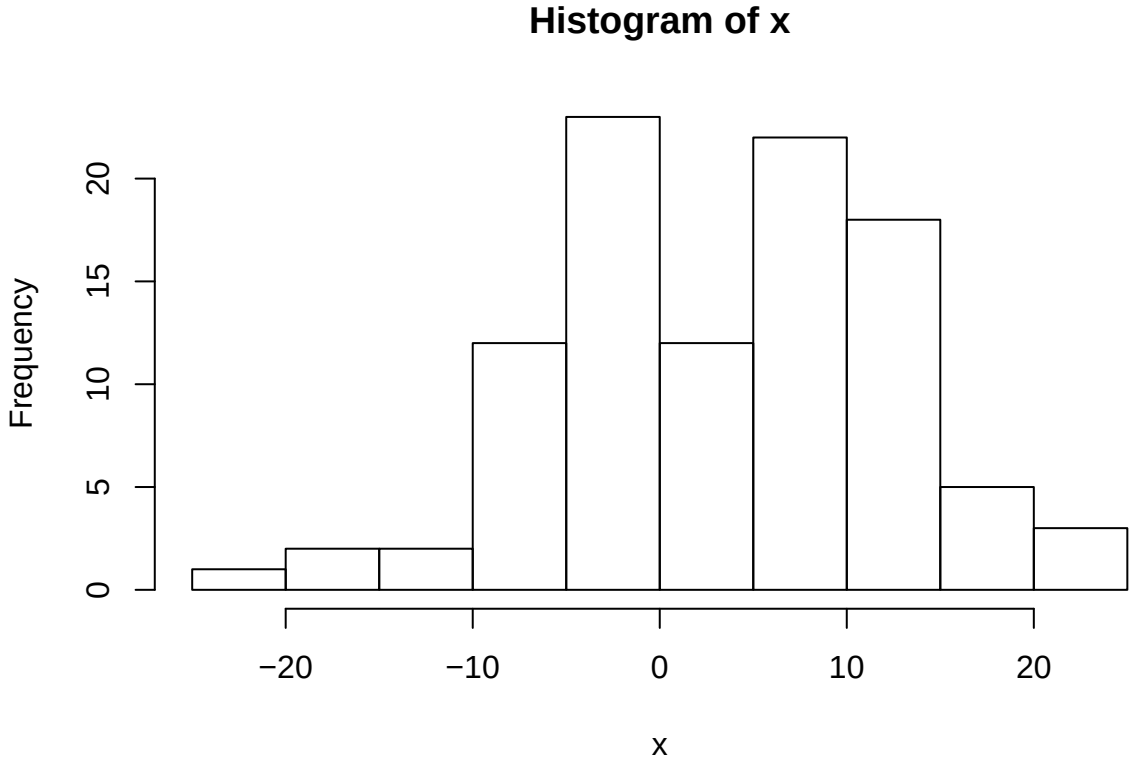
```
?rnorm
?runif
?rbeta
?gamma
```

 # Randomly generate 100 samples following normal distribution with mean 3 and standard deviation 1 and 10

```
x <- rnorm(n=100, mean=3, sd=1)
hist(x)
```




```
x <- rnorm(n=100, mean=3, sd=10)
hist(x)
```



quantile

```
qnorm(0.95, 0, 1, lower.tail = T)
```

```
## [1] 1.644854
```

```
qnorm(0.975, 0, 1, lower.tail = T)
```

```
## [1] 1.959964
```

probility

```
pnorm(1.64, 0, 1, lower.tail = T)
```

```
## [1] 0.9494974
```

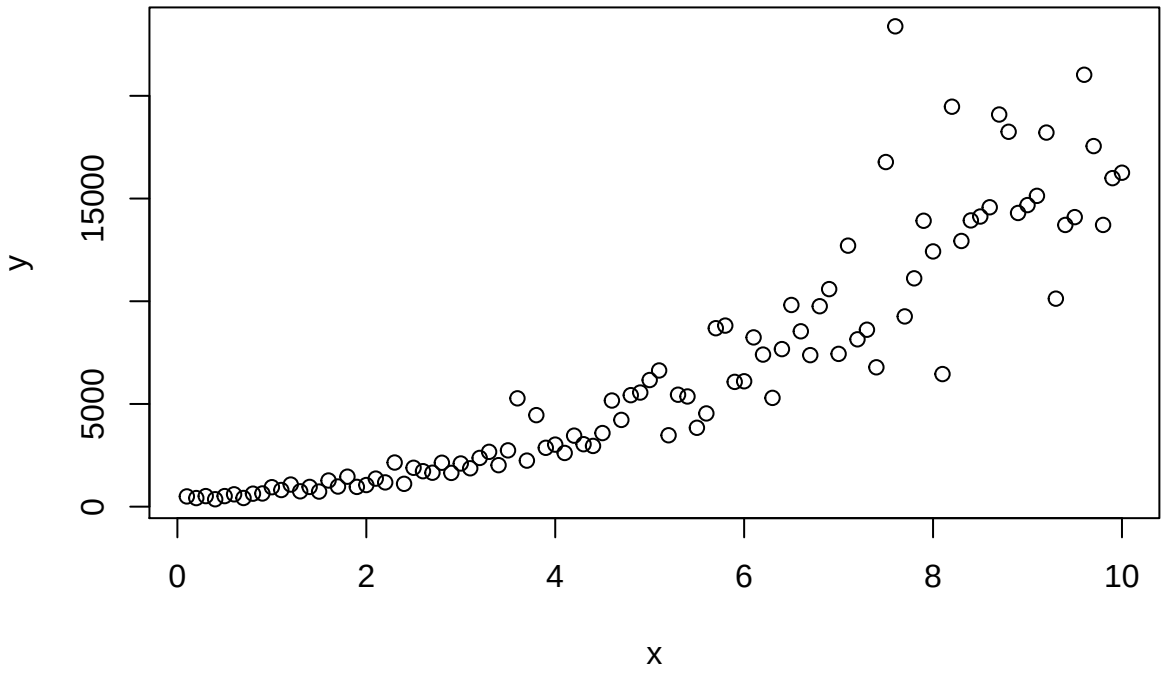
```
pnorm(1.96, 0, 1, lower.tail = T)
```

```
## [1] 0.9750021
```

Given x, generate

$$y = (x+5)^{(e^{1.3}+N(0,0.1))}$$

```
x=seq(0.1,10,0.1)
set.seed(2017)
rand = rnorm(100,0,0.1)
y = (x+5)^(exp(1.3)+rand)
plot(x,y)
```



Exploratory Data Analysis (EDA)

```
summary(x,y)
```

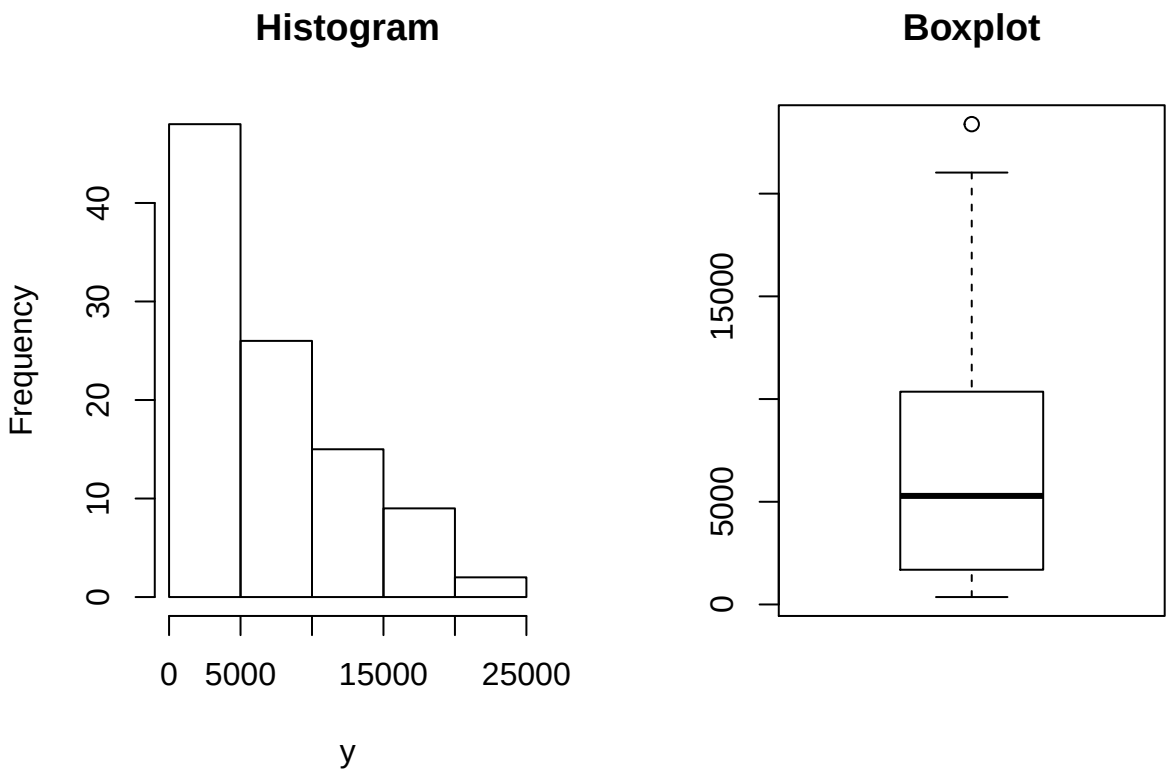
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   2.575   5.050   5.050   7.525  10.000
```

```
cor(x,y)
```

```
## [1] 0.9029026
```

Basic Visualization histogram & Boxplot

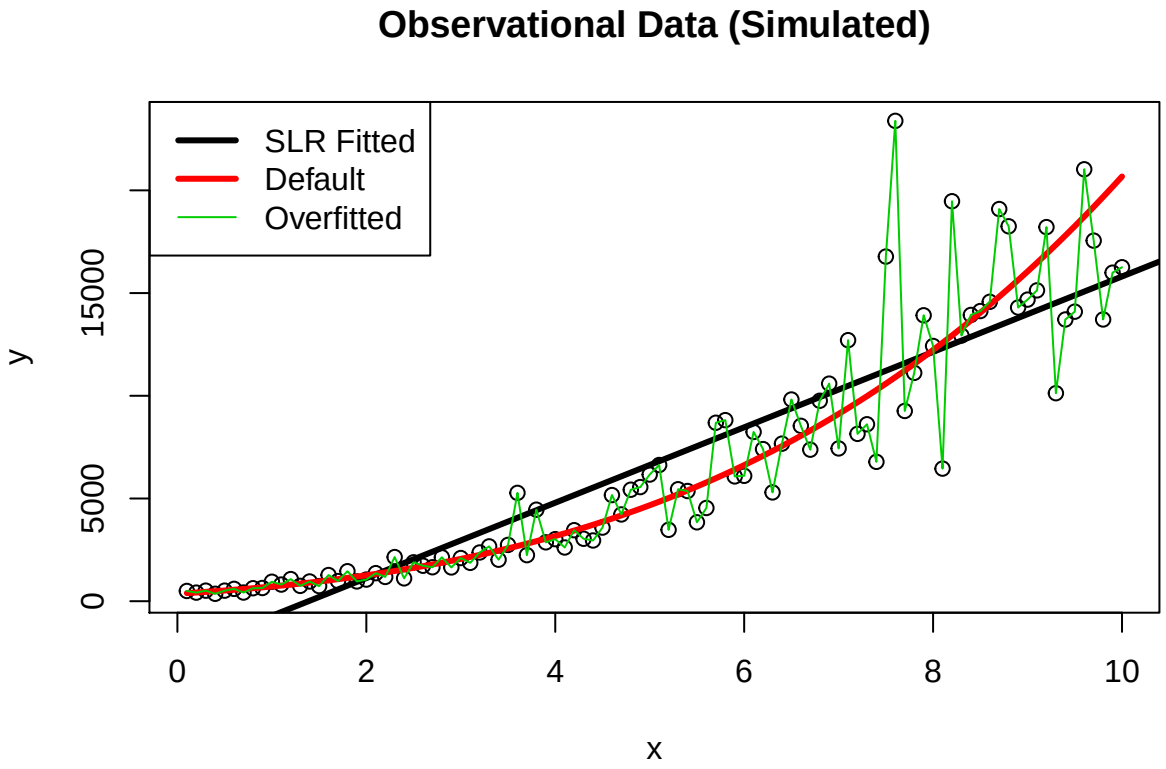
```
par(mfrow=c(1,2))
hist(y, main = "Histogram")
boxplot(y, main = "Boxplot")
```



Plot the data with fitted line

```
par(mfrow=c(1,1))
plot(x,y, pch=1, main="Observational Data (Simulated)")
fit1 = lm(y~x)
```

```
abline(fit1$coefficients,lwd=3, col=1)
yhat = (x+5)^(exp(1.3))
lines(x,yhat, lwd=3,  col=2)
lines(x,y, lwd=1, col=3)
legend('topleft', c("SLR Fitted","Default","Overfitted"), lwd=c(3,3,1), col=c(1,2,3))
```



To find the best fitted line, our initial aim is to minimize the MSE,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{f}(x)]^2$$

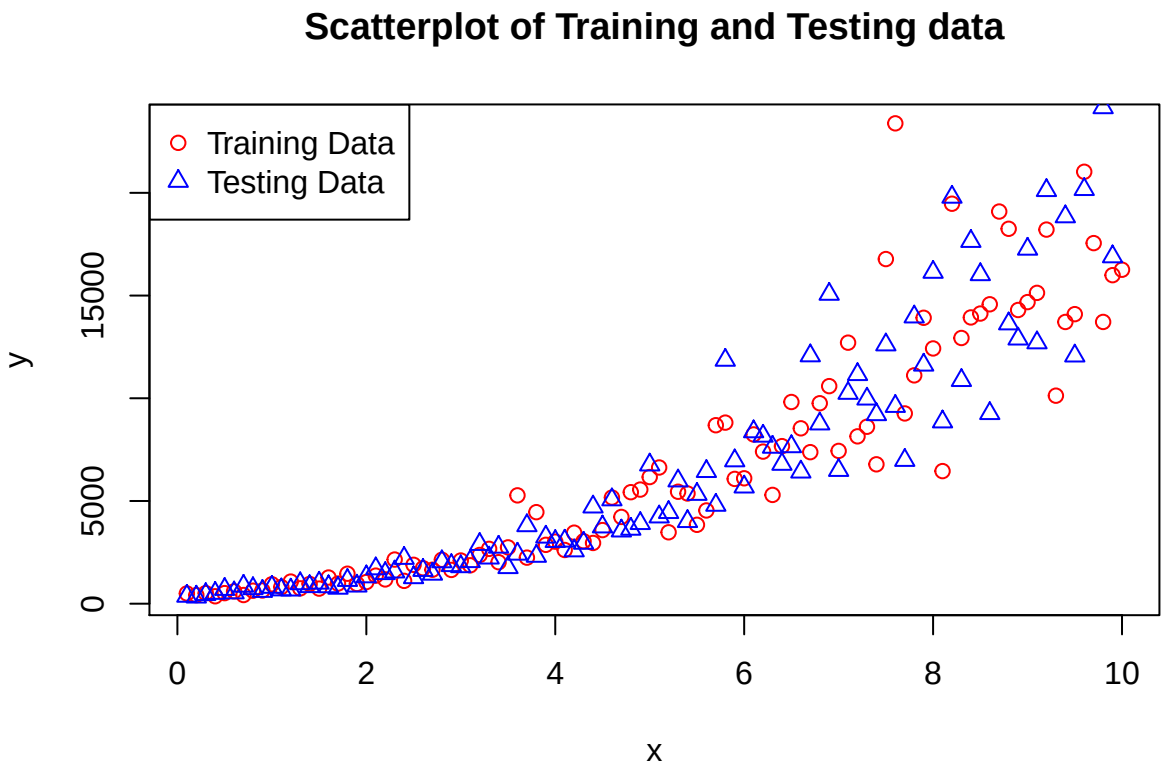
```
MSE=NULL
n = length(x)
MSE[1] = sum((y-fit1$fitted.values)^2)/n
MSE[2] = sum((y-yhat)^2)/n
MSE[3] = sum((y-y)^2)/n
MSE
```

```
## [1] 6361162 5733505       0
```

Generate test data based on

$$y = (x+5)^{(e^{1.3}+N(0,0.1))}$$

```r
x=seq(0.1,10,0.1)
set.seed(2018)
rand = rnorm(100,0,0.1)
y.test = (x+5)^(exp(1.3)+rand)
plot(x,y, col=2, pch=1, main="Scatterplot of Training and Testing data")
points(x,y.test,col=4, pch=2)
legend("topleft", c("Training Data", "Testing Data"), pch=c(1,2), col=c(2,4))
```



New MSE based on test data

```r
MSE.new=NULL
n = length(x)
MSE.new[1] = sum((y.test-fit1$fitted.values)^2)/n
MSE.new[2] = sum((y.test-yhat)^2)/n
MSE.new[3] = sum((y.test-y)^2)/n
MSE.new
```

```
## [1] 11804231  7344370 11365977
```

Regularized Least Squares:

$$minimize\{\frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{f}(x)]^2 + \lambda \int |f''|\mu|^2 d\mu\}$$