Week4_Practice

ChrisCoussa

10/22/2018

```
###### Week 4 Practice, Monday######
library(DevFarn2)
```

```
## Loading required package: qcc
```

```
## Package 'qcc' version 2.7
```

```
## Type 'citation("qcc")' for citing this R package in publications.
```

```
library(qcc)
library(readxl)
library(gplots)
```

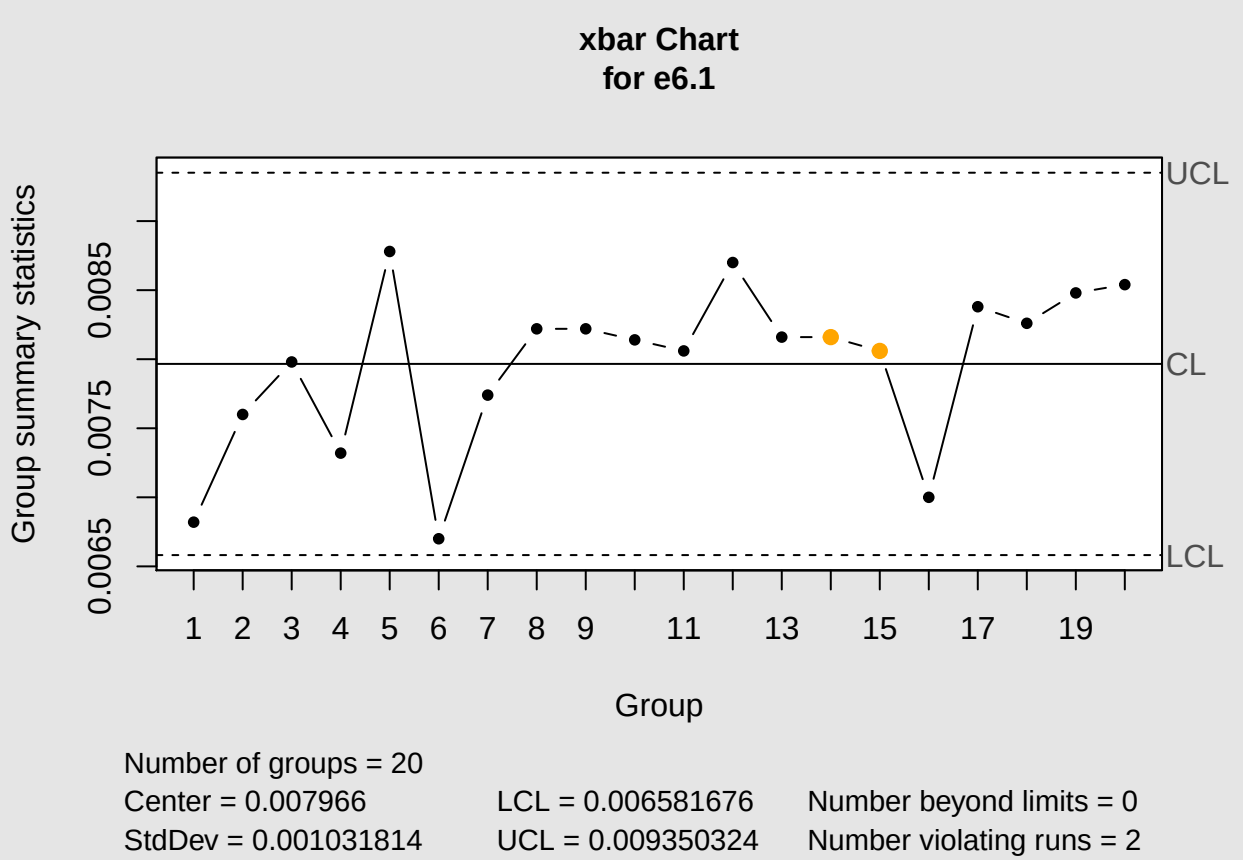
```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
data(e6.1)
# View(e6.1)
class(e6.1)
```

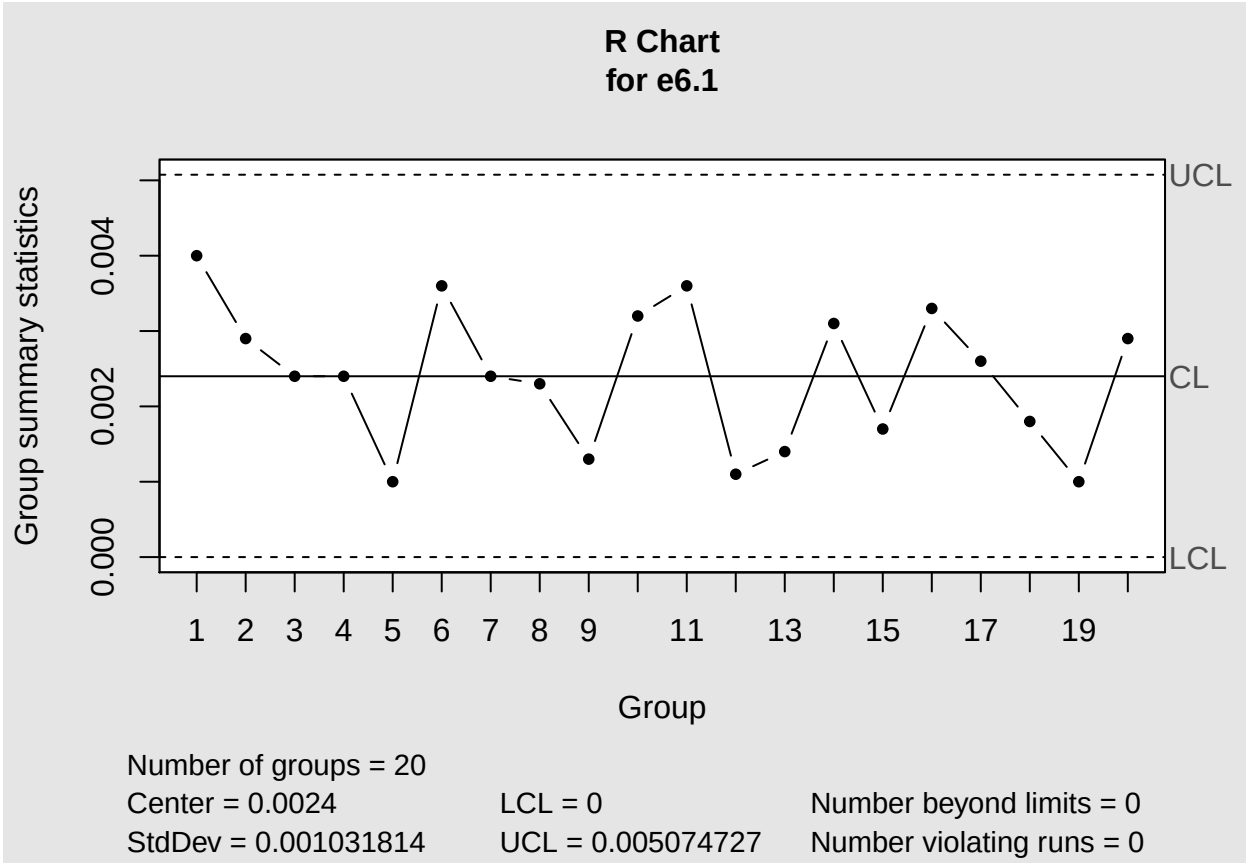
```
## [1] "data.frame"
```

```
qcc(e6.1, type = "xbar") # Creating an xbar chart form the dataset (defaults to xbar)
```




```
## List of 11
##  $ call      : language qcc(data = e6.1, type = "xbar")
##  $ type      : chr "xbar"
##  $ data.name : chr "e6.1"
##  $ data      : num [1:20, 1:5] 0.0061 0.0088 0.008 0.0067 0.0087 0.0071 0.0078 0.0087 0.0074 0.0081
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:20] 0.00682 0.0076 0.00798 0.00732 0.00878 0.0067 0.00774 0.00822 0.0082
##   ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
##  $ sizes     : Named int [1:20] 5 5 5 5 5 5 5 5 5 5 ...
##   ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
##  $ center    : num 0.00797
##  $ std.dev   : num 0.00103
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0.00658 0.00935
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
qcc(e6.1, type = "R") # Creating a R chart from the dataset
```

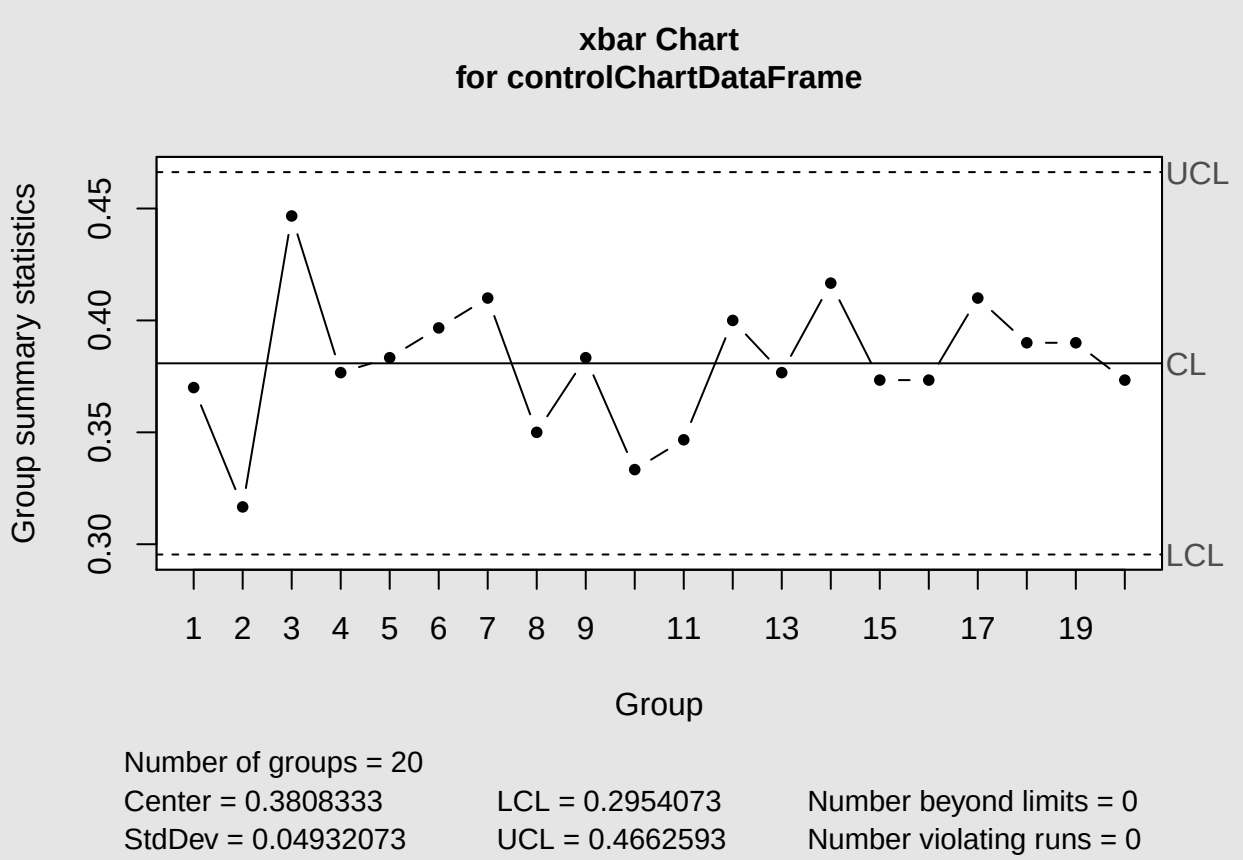



```
## List of 11
##  $ call      : language qcc(data = e6.1, type = "R")
##  $ type      : chr "R"
##  $ data.name : chr "e6.1"
##  $ data      : num [1:20, 1:5] 0.0061 0.0088 0.008 0.0067 0.0087 0.0071 0.0078 0.0087 0.0074 0.0081
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:20] 0.004 0.0029 0.0024 0.0024 0.001 0.0036 0.0024 0.0023 0.0013 0.0032
##   ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
##  $ sizes     : Named int [1:20] 5 5 5 5 5 5 5 5 5 5 ...
##   ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
##  $ center    : num 0.0024
##  $ std.dev   : num 0.00103
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0 0.00507
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```r
# Using data inputed by hand to create a dataframe to use with Control Charts
controlChart_x1 = c(.36, .3, .51, .42, .39, .33, .43, .41, .37, .26, .36, .38, .29,
                    .44, .38, .31, .39, .43, .40, .40)
controlChart_x2 = c(.39, .35, .41, .37, .38, .41, .39, .32, .42, .42, .32, .47, .45,
                    .38, .37, .43, .49, .36, .45, .40)
controlChart_x3 = c(.36, .30, .42, .34, .38, .45, .41, .32, .36, .32, .36, .35, .39,
                    .43, .37, .38, .35, .38, .32, .32)
# Creating a dataframe from the variables
controlChartDataFrame = data.frame(controlChart_x1, controlChart_x2, controlChart_x3)
```

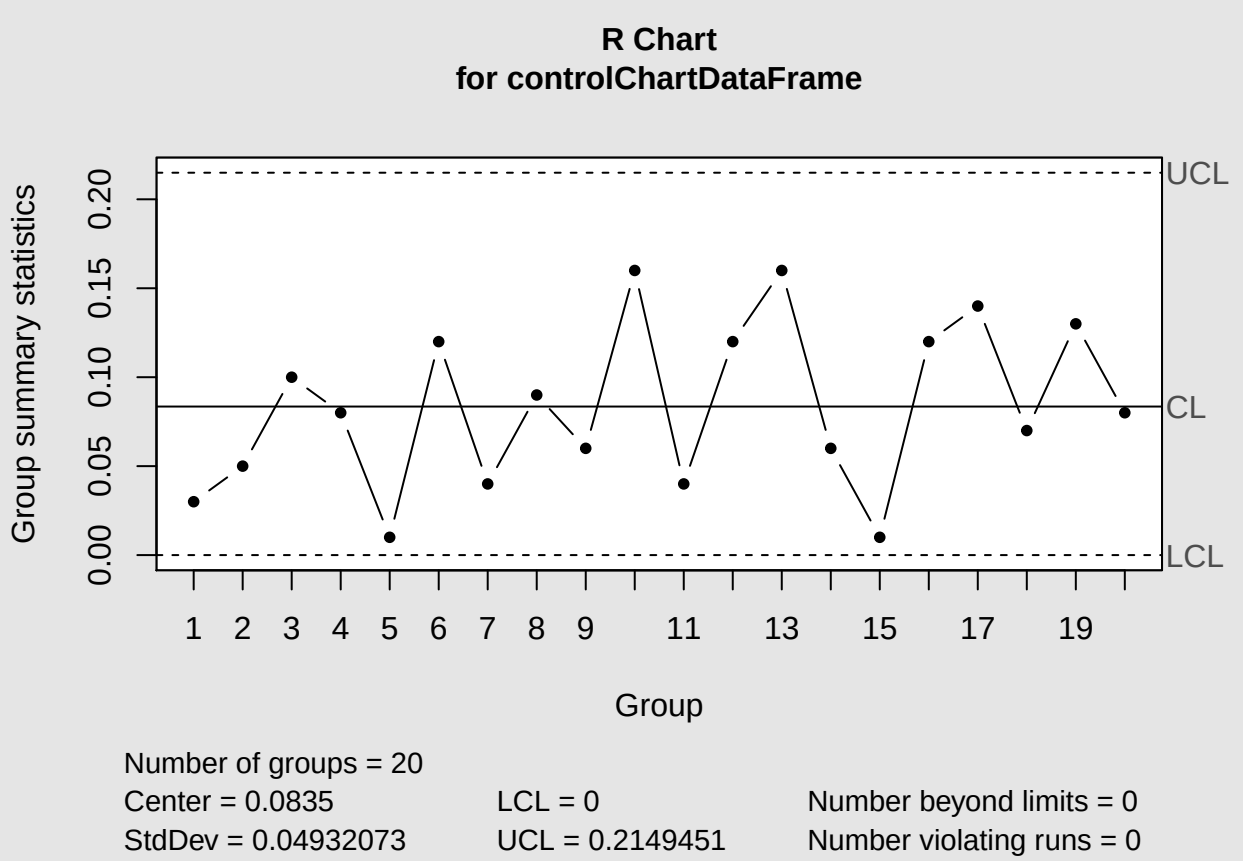
```r
# Creating an xbar Chart
qcc(controlChartDataFrame)
```

```
## Warning in qcc(controlChartDataFrame): chart 'type' not specified, assuming
## "xbar"
```



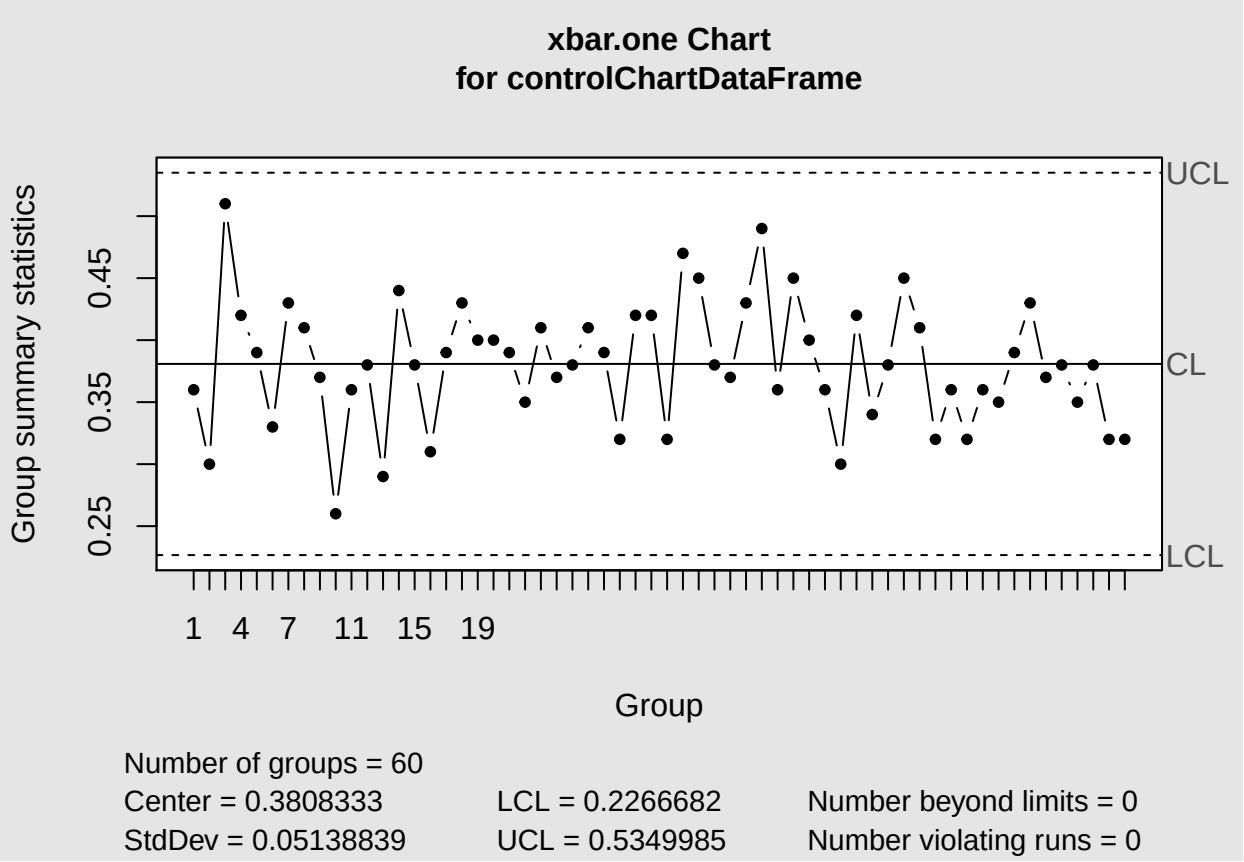

```
## List of 11
##  $ call      : language qcc(data = controlChartDataFrame)
##  $ type      : chr "xbar"
##  $ data.name : chr "controlChartDataFrame"
##  $ data      : num [1:20, 1:3] 0.36 0.3 0.51 0.42 0.39 0.33 0.43 0.41 0.37 0.26 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:20] 0.37 0.317 0.447 0.377 0.383 ...
##   ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
##  $ sizes     : int [1:20] 3 3 3 3 3 3 3 3 3 3 ...
##  $ center    : num 0.381
##  $ std.dev   : num 0.0493
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0.295 0.466
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```r
# Creating a R-Chart
qcc(controlChartDataFrame, type = "R")
```




```
## List of 11
##  $ call      : language qcc(data = controlChartDataFrame, type = "R")
##  $ type      : chr "R"
##  $ data.name : chr "controlChartDataFrame"
##  $ data      : num [1:20, 1:3] 0.36 0.3 0.51 0.42 0.39 0.33 0.43 0.41 0.37 0.26 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:20] 0.03 0.05 0.1 0.08 0.01 ...
##   ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
##  $ sizes     : int [1:20] 3 3 3 3 3 3 3 3 3 3 ...
##  $ center    : num 0.0835
##  $ std.dev   : num 0.0493
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0 0.215
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
qcc(controlChartDataFrame, type = "xbar.one")
```

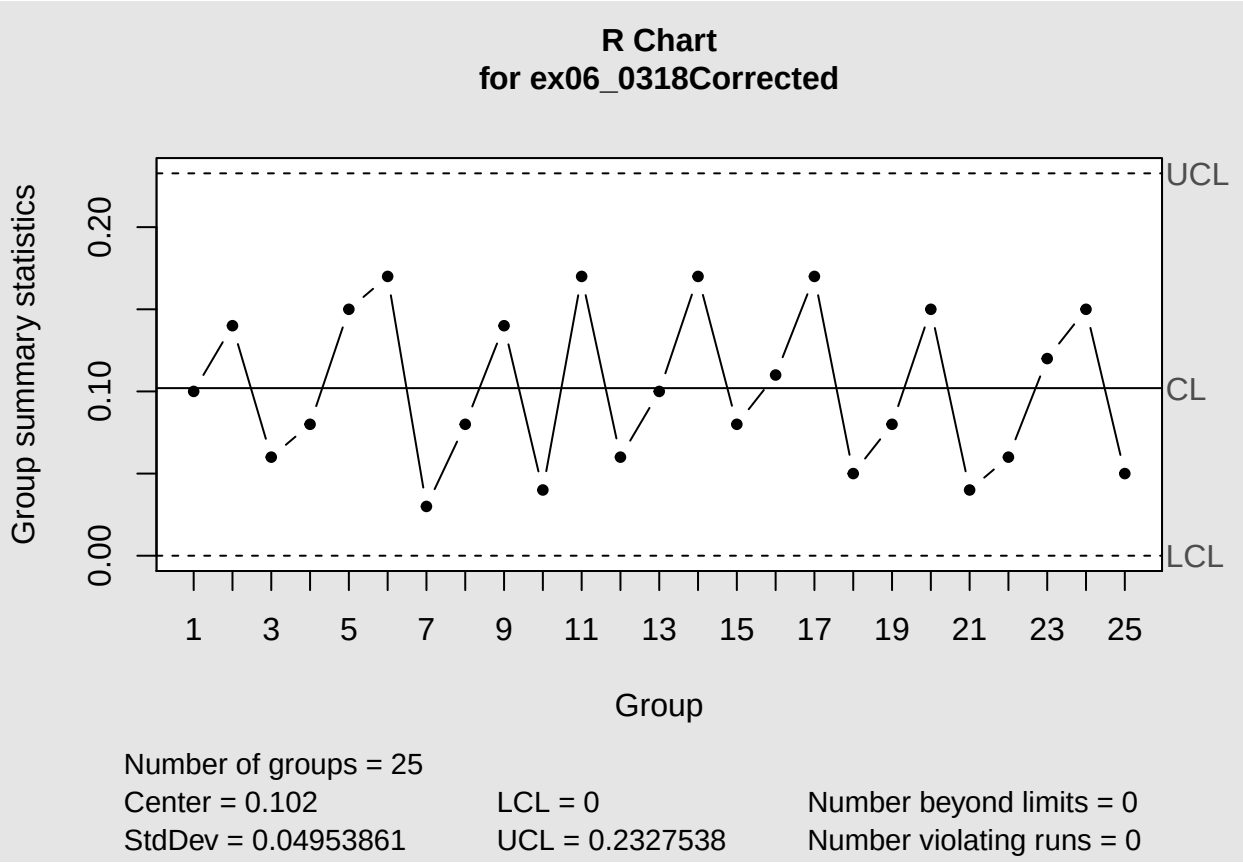



```
## List of 11
##  $ call      : language qcc(data = controlChartDataFrame, type = "xbar.one")
##  $ type      : chr "xbar.one"
##  $ data.name : chr "controlChartDataFrame"
##  $ data      : num [1:20, 1:3] 0.36 0.3 0.51 0.42 0.39 0.33 0.43 0.41 0.37 0.26 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:60] 0.36 0.3 0.51 0.42 0.39 0.33 0.43 0.41 0.37 0.26 ...
##   ..- attr(*, "names")= chr [1:60] "1" "2" "3" "4" ...
##  $ sizes     : int [1:20] 3 3 3 3 3 3 3 3 3 3 ...
##  $ center    : num 0.381
##  $ std.dev   : num 0.0514
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0.227 0.535
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
# NOTE: If you import an excel spreadsheet you can use the following line to eliminate
# columns you don't want to use
# testData = test[, -c(1,5)] <---- the -c(1,5) tells R not to use columns 1 & 5

# Sec 6.3, number 18 from the book.
# part a.
# Importing the dataset from the data_set folder. Make sure to add the
# library(readxl) at the top of the code.
ex06_0318 <- read_excel("/Users/ccoussa717/Desktop/College Classes/National University/CSC220_AppliedSta
```

```r
# This line takes the dataset and removes the first column from being used.
ex06_0318Corrected = ex06_0318[, -c(1)]
# View(ex06_0318Corrected)
qcc(ex06_0318Corrected, type = "R")
```

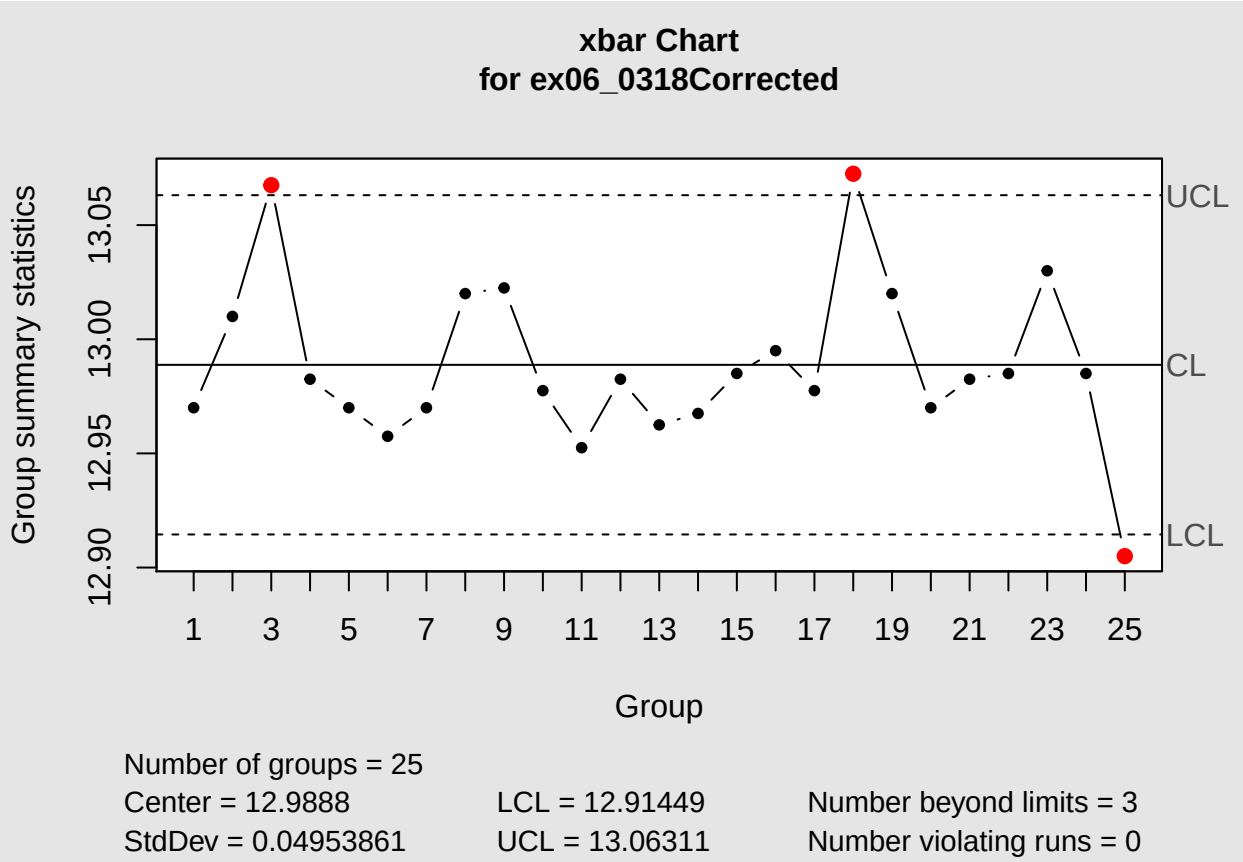



```
## List of 11
##  $ call      : language qcc(data = ex06_0318Corrected, type = "R")
##  $ type      : chr "R"
##  $ data.name : chr "ex06_0318Corrected"
##  $ data      : num [1:25, 1:4] 13 13 13 13 13 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 0.1 0.14 0.06 0.08 0.15 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 4 4 4 4 4 4 4 4 4 4 ...
##  $ center    : num 0.102
##  $ std.dev   : num 0.0495
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0 0.233
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```r
# In the R Chart there are no out-of-control signals.

# part b
qcc(ex06_0318Corrected)
```

```
## Warning in qcc(ex06_0318Corrected): chart 'type' not specified, assuming
## "xbar"
```




```
## List of 11
##  $ call      : language qcc(data = ex06_0318Corrected)
##  $ type      : chr "xbar"
##  $ data.name : chr "ex06_0318Corrected"
##  $ data      : num [1:25, 1:4] 13 13 13 13 13 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 13 13 13.1 13 13 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 4 4 4 4 4 4 4 4 4 4 ...
##  $ center    : num 13
##  $ std.dev   : num 0.0495
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 12.9 13.1
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
# Yes, there are three out-of-control signals in this dataset.

###### Calculating x-bar (mean of the mean) and R-bar (mean of the range) ######
# This line gets the means for all the rows of data
rowMeans(ex06_0318Corrected)
```

```
##  [1] 12.9700 13.0100 13.0675 12.9825 12.9700 12.9575 12.9700 13.0200
##  [9] 13.0225 12.9775 12.9525 12.9825 12.9625 12.9675 12.9850 12.9950
```

```
## [17] 12.9775 13.0725 13.0200 12.9700 12.9825 12.9850 13.0300 12.9850
## [25] 12.9050
```

```
# This row calculates the mean of the means, or x-bar (CENTERLINE)
mean(rowMeans(ex06_0318Corrected)) # <---- 12.9888
```

```
## [1] 12.9888
```

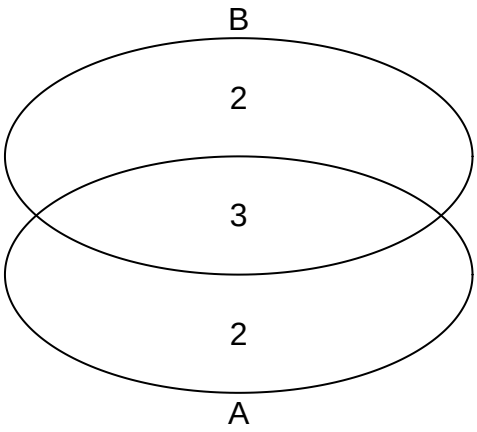
```
# This line gets the range of the dataset
apply(ex06_0318Corrected, 1, max) - apply(ex06_0318Corrected, 1, min)
```

```
##  [1] 0.10 0.14 0.06 0.08 0.15 0.17 0.03 0.08 0.14 0.04 0.17 0.06 0.10 0.17
## [15] 0.08 0.11 0.17 0.05 0.08 0.15 0.04 0.06 0.12 0.15 0.05
```

```
# This line gets the mean of the range, or R-bar (CENTERLINE)
mean(apply(ex06_0318Corrected, 1, max) - apply(ex06_0318Corrected, 1, min)) # <--- 0.102
```

```
## [1] 0.102
```

```
###### Creating Venn Diagrams ######
# In order to create a Venn diagram you need to install the gplots package and
# activate the library at the top.
# a & b represent sets of 3 coin flips
a = c("HHH", "HTT", "THT", "TTT", "HHT")
b = c("THH", "HTT", "HHT", "TTT", "TTH")
# Creates a list from variables a & b
v.list = list(a, b)
# Create the venn diabram from the list of data
venn(v.list)
```



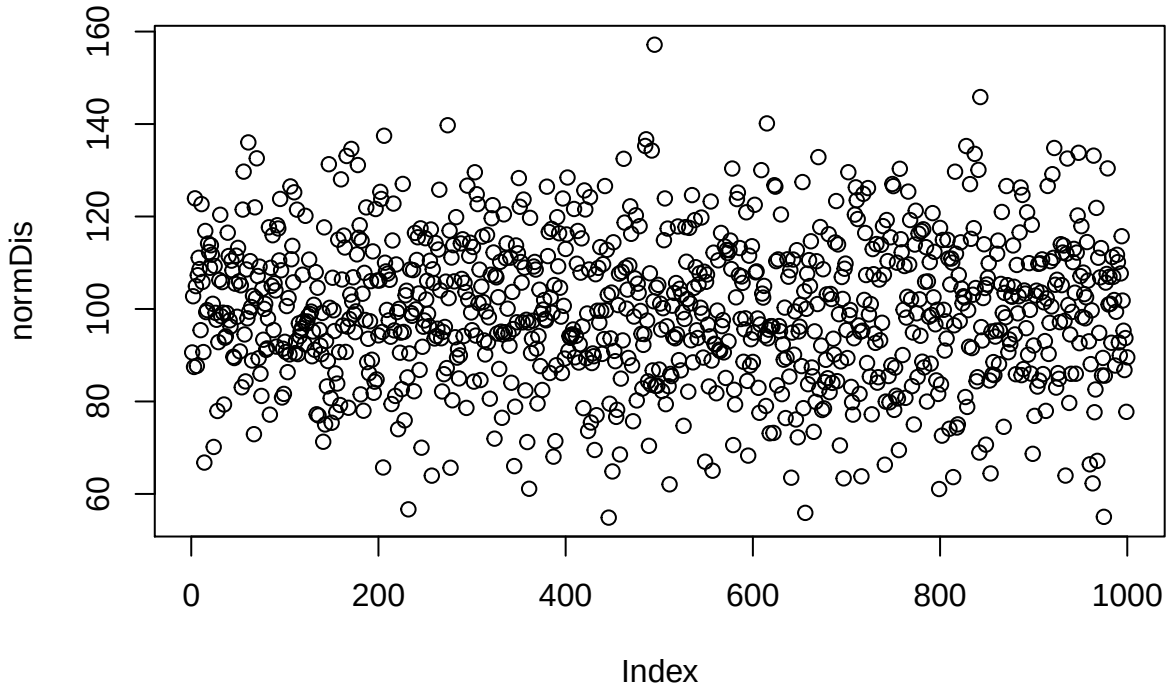

```
###### Creating random Normally Distributed data ######
# Each time you run the rnorm command it will create a new set of random numbers
# so that the sd and mean will be different.
# If you add the set.see(1) command it will always generate the same numbers
set.seed(1)
normDis = rnorm(1000, mean = 100, sd = 15)
mean(normDis) # <---- Very close to 100 at 99.82528
```

```
## [1] 99.82528
```

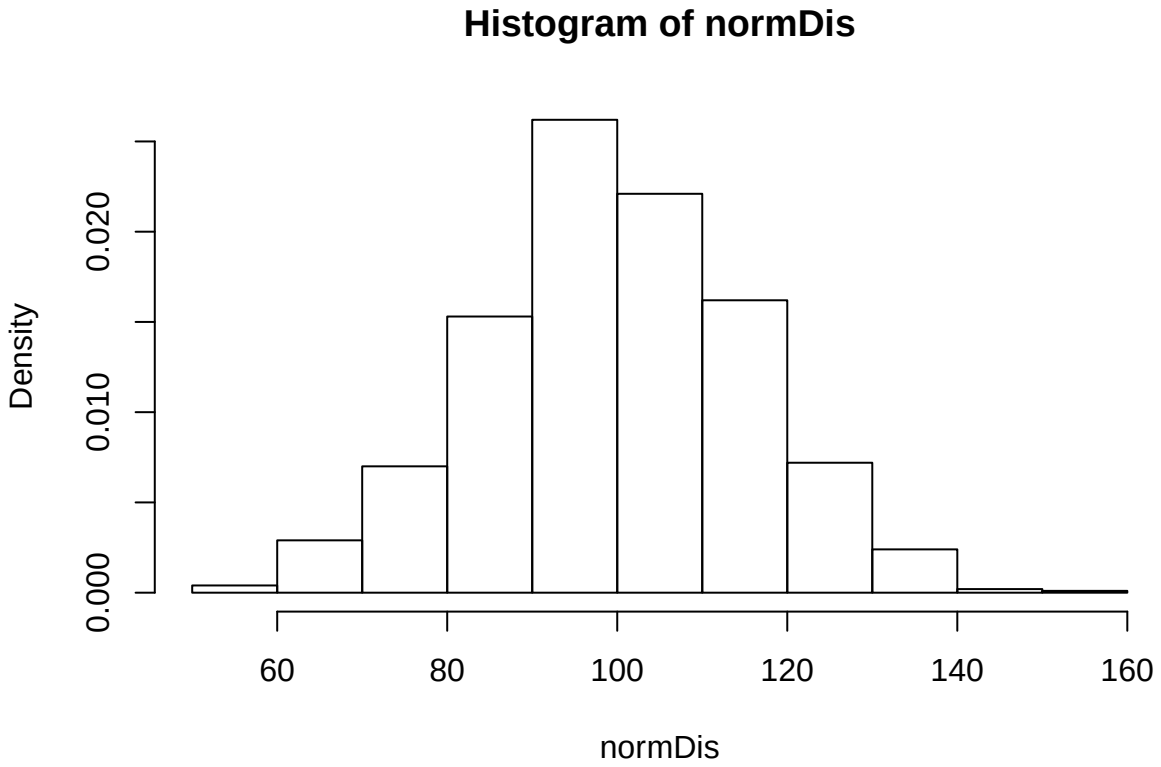
```
sd(normDis) # <---- Very close to 15 at 15.52374
```

```
## [1] 15.52374
```

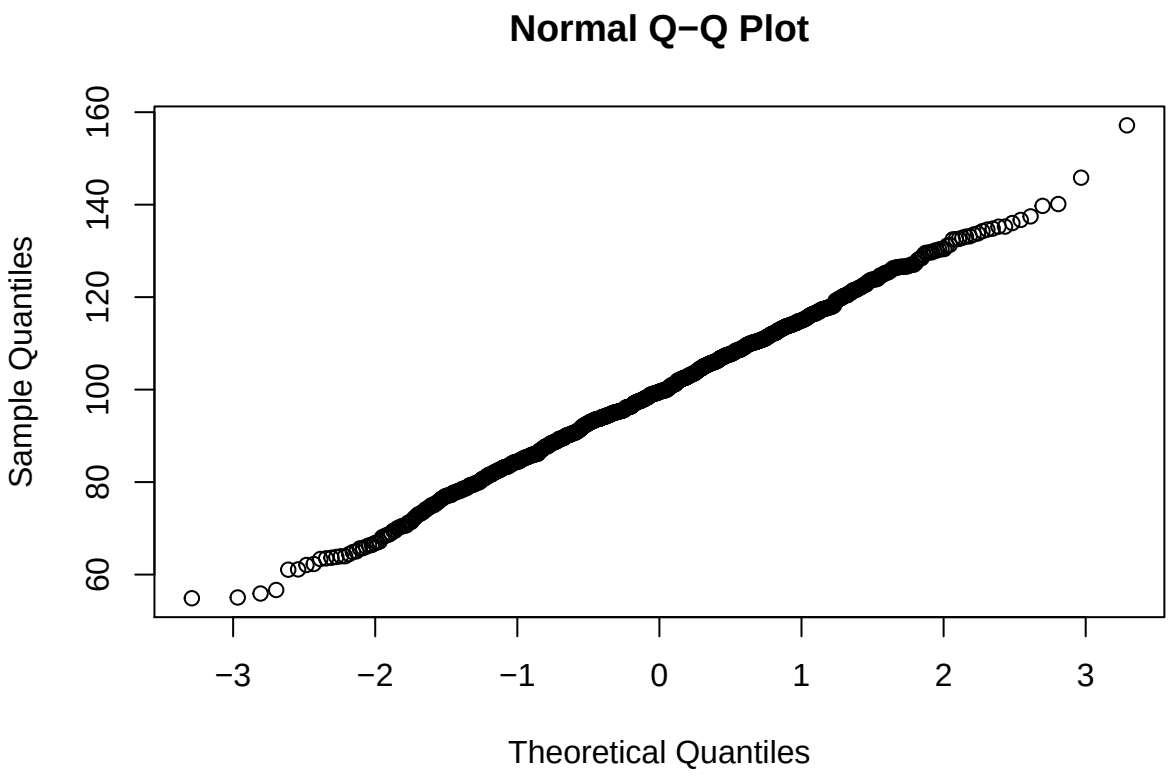
```r
plot(normDis)
```




```r
#### Visualizing the distribution of the data
# Histogram of normDis dataset, shows the distribution of the dataset
hist(normDis, probability = TRUE)
```




```r
# Q-Q plot of the dataset, shows how close to Normally Distributed your data is.
qqnorm(normDis)
```


Normal Q−Q Plot
Sample Quantiles
60
80
100
120
140
160
−3
−2
−1
0
1
2
3
Theoretical Quantiles