R Handout: Descriptive Statistics

This handout will introduce a basic use of R with an example describing a data. This pdf file and the R program with the example data are provided on Blackboard.

1. Load a data. Here, I will show two different ways to load data. 1) Loading data from a CSV file. 2) Directly inputing data into R. You can also load a data from other types of data source, such as Excel, SQL server and etc.. Because I am not going to cover all of them, if you are interested in other methods, please see:

   https://cran.r-project.org/doc/manuals/r-devel/R-data.html

   This CSV file contains a data monitoring a manufacturing process about Automobile Engine Piston Rings [1]. Theoretically, each row is identically and independently distributed with each other.

   - R code (Loading from a CSV file)

```
# Load table from a CSV file
# Locate the CSV file
addr <- 'https://raw.githubusercontent.com/
bolus123/R-handout/master/DescriptiveStatistics/example.csv'
# The link can be replaced by your local address
# such as 'C:/yourfolder/example.csv'

# Load the data in R and transfer it into a matrix
data <- as.matrix(read.csv(file = addr))

# Show the data
data
```

- R code (Directly inputing into R)

```
# Directly input a data as a matrix
data <- matrix(c(
74.03 ,74.002 ,74.019 ,73.992 ,74.008 ,
73.995 ,73.992 ,74.001 ,74.011 ,74.004 ,
73.988 ,74.024 ,74.021 ,74.005 ,74.002 ,
74.002 ,73.996 ,73.993 ,74.015 ,74.009 ,
73.992 ,74.007 ,74.015 ,73.989 ,74.014 ,
74.009 ,73.994 ,73.997 ,73.985 ,73.993 ,
73.995 ,74.006 ,73.994 ,74 ,74.005 ,
73.985 ,74.003 ,73.993 ,74.015 ,73.988 ,
74.008 ,73.995 ,74.009 ,74.005 ,74.004 ,
73.998 ,74 ,73.99 ,74.007 ,73.995 ,
73.994 ,73.998 ,73.994 ,73.995 ,73.99 ,
74.004 ,74 ,74.007 ,74 ,73.996 ,
73.983 ,74.002 ,73.998 ,73.997 ,74.012 ,
74.006 ,73.967 ,73.994 ,74 ,73.984 ,
74.012 ,74.014 ,73.998 ,73.999 ,74.007 ,
74 ,73.984 ,74.005 ,73.998 ,73.996 ,
73.994 ,74.012 ,73.986 ,74.005 ,74.007 ,
74.006 ,74.01 ,74.018 ,74.003 ,74 ,
73.984 ,74.002 ,74.003 ,74.005 ,73.997 ,
74 ,74.01 ,74.013 ,74.02 ,74.003 ,
73.982 ,74.001 ,74.015 ,74.005 ,73.996 ,
74.004 ,73.999 ,73.99 ,74.006 ,74.009 ,
74.01 ,73.989 ,73.99 ,74.009 ,74.014 ,
74.015 ,74.008 ,73.993 ,74 ,74.01 ,
73.982 ,73.984 ,73.995 ,74.017 ,74.013
), ncol = 5, byrow = T)

# Show the data
data
```

2. Describe this data with graphs

   - R code

```
# Graph this data
# histogram for the whole data
# with the maximum number of breaks 10
# in other words, the maximum number of bins is 11
hist(data, breaks = 10)
# Notice that y-axis is frequency

# boxplot for the whole data
boxplot(as.vector(data))

# boxplot for each column
boxplot(data)
# boxplot for each row
boxplot(t(data))
# They are just examples and please notice that
# it makes less sense if you draw boxplots for each column
# or for each row at the setting of this data
```

3. Describe this data with statistics

   - R code

```
# Basic statistics
mean(data) # grand mean
colMeans(data) # means for each column
rowMeans(data) # means for each row

var(as.vector(data)) # grand variance
var(data) # covariance matrix for each column
var(t(data)) # covariance matrix for each row

# percentiles including 1%, 5%, 10%, 25%,
# 50%, 75%, 90%, 95% and 99%
```

```
quantile(data
, c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99))
```

4. Fit a distribution. Suppose we guess this data is following a normal distribution.

   - R code

```
# fit a normal distribution for the whole data
# estimate parameters
mu <- mean(data) # grand mean
sigma <- sqrt(var(as.vector(data))) # standard deviation

# draw a histogram for the whole data
# with 10 breaks (11 bins)
# and y-axis, density (pdf)
hist(data, freq = FALSE, ylim = c(0, 40), breaks = 10)
# add a normal pdf with estimated parameers to the histograme
curve(dnorm(x, mean = mu, sd = sigma), add = T, col = 'blue')
```

5. Check the normality. We need to verify our normal assumption, because there is no guarantee that we are right. Here, I will show a verification by a Q-Q plot.

   - R code

```
# check the normality (Q-Q plot)
# we need to have the empirical quantile and the theoretical
# quantile based on the empirical probability

# 1. we need to know the whole sample size
n <- dim(data)[1] * dim(data)[2]

# 2. sort the data and obtain their frequencies
e.d <- table(as.vector(data))

# 3. the empirical quantiles is the names of this vector
e.q <- as.numeric(names(e.d))
```

```
# 4. calculate the empirical p.d.f.
e.p <- e.d / n

# 5. calculate the empirical c.d.f.
e.c <- cumsum(e.p)

# 6. find out the theoretical quantile
t.q <- qnorm(e.c, mean = mu, sd = sigma)

# 7. draw a Q-Q plot
# draw a scatter plot with x-axis the empirical quantile
# and y-axis the theoretical quantile
plot(e.q, t.q, xlab = 'Empirical', ylab = 'Theoretical'
, main = 'Q-Q plot')
# reference line
points(c(0, 100), c(0, 100), type = 'l', col = 'blue')
```

■References References

[1] Douglas C. Montgomery, *Introduction to Statistical Quality Control*, Wiley, NJ, 6th edition, 2009.