Homework Assignment 1

- **제출마감: 10월 15일 월요일 저녁 11시 59분 (이 시간이 지나면 시스템에서 자동 차단하여 제출 불가함)**
- **모든 문제를 다 풀어보되 Learning-By-Yourself의 Exercise 1과 Exercise 5만 제출할 것**
- 숙제는 myclass.ssu.ac.kr에 파일을 업로드하는 형태로 제출 할 것
- 문서를 스캔을 하여 첨부파일 형태로 제출하거나 한글 또는 워드 문서는 pdf로 변환하여 제출 할 것
- 한 사람당 1개의 파일만 업로드 가능하고 100M 사이즈 제약이 있음을 유의할 것

Self-Checking Exercises

수업 내용을 잘 이해했나 스스로 확인해 볼 수 있는 문제들입니다. 강의 노트에서 대부분의 답을 찾을 수 있습니다.

1. We learned several types of data structures, i.e. atomic vector, list, matrix, data frame, array, frequently used in the R system. Please fill out the blanks in the table below - classified by dimensionality (layout of elements) and homogeneity (of elements).

	Homogeneous	**Heterogeneous**
1 dim		
2 dim		
n dim		

2. What are the most common types of atomic vectors?

3. What is the type of a vector, c("1", $1:3$, FALSE)? And why?

```
typeof(c("1", 1:3, FALSE))
```

4. If $x = 1:10$, what are the type and mode of x, respectively?

```
typeof(x)
mode(x)
```

5. If $x =$ seq(1, 2, by = 0.1), what are the type and mode of x, respectively?

```
typeof(x)
mode(x)
```

6. If $x =$ c(1:4), provide matrices created by the R scripts below:

```
matrix(x, 2, 2)
matrix(x, 2, 2, byrow = T)
```

7. Is this statement true or false? (TRUE or FALSE)

```
A list is a type of vectors.
```

8. What is the difference between two matrix operations below? Write down the results of the operations below, respectively.

```
A <- matrix(c(1,3,2,4), 2, 2)
B <- matrix(c(1,2,2,1), 2, 2)
A * B
A %*% B
```

9. What is the result of the vector operation below? And why?

```
c(1:3) + c(1:6)
```

10. What is the answer to the summation over a logical vector below? And why?

```
sum( c(F,T,T,F) )
```

11. What does the function `is.vector()` do?

```
x <- c(aname = "a", bname = "b", cname = "c")
names(x)
is.vector(x)

attributes(x) <- list( name = "abc vector")
is.vector(x)
```

12. Make a function checking whether an object is a vector or not (Hint: use `is.atomic()` and `is.list()`).

```
check.vector <- function( x ){

}
```

13. Does the statement, `A matrix is an atomic vector with dimension`, make sense? Think about it based on the following example:

```
> x <- 1:12
> dim(x) <- c(3,4)
> x
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

14. What is the difference between data frames `df1` and `df2` below? (Hint: take a look at their structures).

```
df1 <- data.frame( x = 1:3, y = letters[1:3] )
df2 <- data.frame( x = 1:3, y = letters[1:3], stringsAsFactors = FALSE )
```

15. Please write down what results would be produced from the scripts below when they are conducted.

```
rm( list = ls() )
(c(-2,2) >= 0) & (c(-2,2) <= 0)
(c(-2,2) >= 0) && (c(-2,2) <= 0)

(c(-2,2) >= 0) | (c(-2,2) <= 0)
(c(-2,2) >= 0) || (c(-2,2) <= 0)

TRUE || a
TRUE | a
```

16. What is the difference between `floor()` and `trunc()` below? Explain by using the example below.

```
> x <- c(-3.2, -1.8, 2.3, 2.9)
> floor(x)
[1] -4 -2  2  2
> trunc(x)
[1] -3 -1  2  2
```

17. Why is the difference between `round()` and `signif()` below? Explain by using the example below.

```
> round(135.789, 2)
[1] 135.79
> signif(135.789, 2)
[1] 140
```

18. Draw a histogram comparing an empirical distribution of a random sample from t distribution with df = 1 with its theoretical distribution (Hint: consult the following code where a standard normal distribution is used instead of a t distribution). What does the argument `freq` of the function `hist()` do (Hint: consult the help file of `hist()`)?

```
dat <- rnorm(1000)
xx <- seq(-3, 3, by = 0.1)
yy <- dnorm(xx)
hist(dat, freq = F, col = "grey50", main = "Histogram of a random sample from N(0,1)")
lines(xx, yy, col = 4, lwd = 3)
```

Learning-By-Yourself

수업 내용을 기반으로 혼자서 더 심도있는 학습을 할 수 있도록 도움을 주는 문제들입니다. 강의 노트와 R help files (만약 필요하다면 관련 책, 기사, 웹사이트 등)을 찾아보며 충분히 고민하면서 문제를 풀어보세요.

Exercise 1: `factor()` and `levels()`

- `factor()` 함수의 help file을 참조해서 함수의 `levels`와 `labels` arguments에 대해 공부하세요.

```
?factor
```

- `factor()` 함수에 대해 잘 이해하였다고 생각하면, 다음 코드를 실행한 후, 각각의 경우에 level들의 순서, 각 level의 이름, 그리고 "a", "b"에 어떤 명목적인 숫자가 부여되었는지 확인해 보세요.

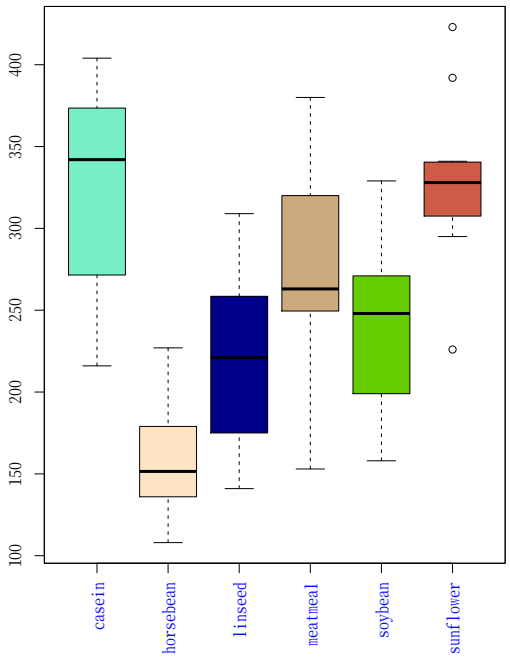
```
f1 <- factor(c("a","b")); f1; as.numeric(f1)
f2 <- factor(c("a","b"), labels = c("c","d")); f2; as.numeric(f2)
f3 <- factor(c("a","b"), levels = c("b","a")); f3; as.numeric(f3)
f4 <- factor(c("a","b"), levels = c("b","a"), labels = c("c","d")); f4; as.numeric(f4)

cbind(f1,f2,f3,f4)
```

- `levels()` 함수의 help file을 읽어본 후 다음 코드를 실행하여 결과를 예측해 보세요. `f1`과 비교하여 `ff`의 level과 label에 어떤 변화가 생기나요?

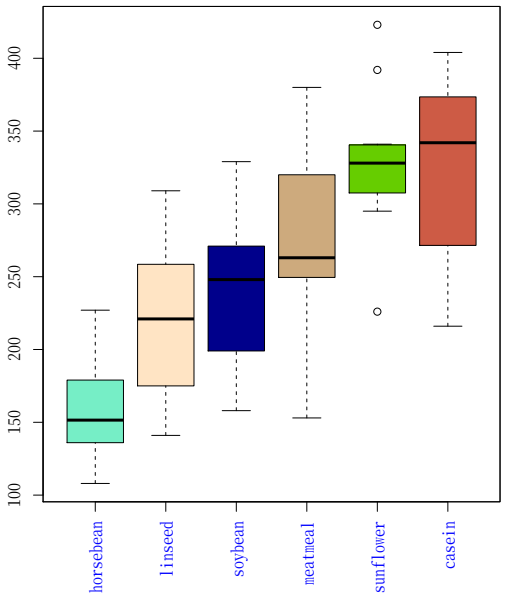
```
ff <- f1
levels(ff) <- c("b", "a")
ff
```

- 위의 내용을 잘 이해하였다면 다음 문제를 문제없이 풀 수 있을 거예요. 일단 아래의 R 코드를 실행해보면 boxplot이 그려질 거예요.

```
par(mar=c(5,5,1,1))
n.lev <- length(levels(chickwts$feed)); n.lev
boxplot(chickwts$weight ~ chickwts$feed, col = colors()[c(1:n.lev)*10], xaxt = "n")
axis(side = 1, at = 1:6, labels = levels(chickwts$feed), col.axis = 4, las = 2)
box()
```



1. 먼저 위의 boxplot을 그리는데 사용된 `chickwts` 자료를 이해해 봅시다.
   - `chickwts` 객체는 R 자료의 종류(type) 중 어떤 종류에 해당하나요?
   - 어떤 변수들이 자료에 포함되어 있나요?
   - 먹이와 연관된 `feed` 변수의 수준(level)은 몇개이며 각 수준의 이름(label)은 무엇인가요?
2. 위의 boxplot에는 feed 변수의 수준들(levels)에 따라 weight 변수의 분포를 일목정연하게 잘 정리해서 보여줍니다. 그림의 x 축의 각 수준의 이름은 영어 알파벳 순서에 따라 정렬되어 있습니다. 만약 x 축의 feed 변수의 순서를 weight 분포의 중앙값의 오른차순으로 변경하여 그림을 그리기를 원한다면, 어떻게 하면 될까요? 한번 아래와 같이 그림을 그려보세요.



Exercise 2: Write Your Own Function Carrying Out the two sample t-test

- 기초 통계학 시간에 "Two sample t-test under an equal variance assumption"에 대해서 배웠을 거예요. 만약 생각이 잘 안난다면 아래의 설명을 참조하세요.

  If we assume that μ_1과 μ_2 represent the means of the two populations of interest, the null hypothesis for comparing the two means is $H_0 : \mu_1 = \mu_2$. The alternative hypothesis can be any one of

$$H_1 : \mu_1 \neq \mu_2$$
$$H_1 : \mu_1 > \mu_2$$
$$H_1 : \mu_1 < \mu_2.$$

  Let's use α to stand for a type I error probability. Also, let's use $\boldsymbol{x} = (x_1, \ldots, x_{n_1})$ and $\boldsymbol{y} = (y_1, \ldots, y_{n_2})$ to denote random samples from two populations, respectively. Then a t-statistic used for comparing two population means is defined as

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{(n_1-1)s_X^2 + (n_2-1)s_Y^2}{n_1+n_2-2}\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

  where $\bar{x}$ and $\bar{y}$ are sample means, s_X^2 and s_Y^2 are sample variances, and the t-statistic has a t distribution with $n_1 + n_2 - 2$ degrees of freedom. The null hypothesis is rejected in favor of the alternative if
  for $H_1 : \mu_1 \neq \mu_2$,

$$|t| > t_{\alpha/2}$$

  for $H_1 : \mu_1 > \mu_2$,

$$t > t_{1-\alpha}$$

  for $H_1 : \mu_1 < \mu_2$,

$$t < t_{\alpha}.$$

- 두 모집단의 평균비교에 사용되는 t-test 함수를 직접 만들어보세요(이 때 두 모집단의 분산은 같다고 가정할 것). 자신이 만든 함수를 사용해서, 평균과 분산이 각각 1인 그리고 평균과 분산이 2와 1인 정규분포에서 각각 $n_1 = 30, n_2 = 35$인 자료를 생성해서 가설 검정을 해 보세요(아래의 R script는 참조만 할 것! 반듯이 자신의 힘으로 함수를 만들어 보세요).

```
myttest <- function( x, y, test = "two-sided", alpha = 0.05){

  n1 <- length(x)
  n2 <- length(y)
  ndf <- n1 + n2 - 2
  s2 <- ((n1 - 1) * var(x) + (n2 - 1) * var(y) ) / ndf
  tstat <- ( mean(x) - mean(y) ) / sqrt( s2 * (1/n1 + 1/n2))
  tail.area <- switch( test,
                       "two-sided" = 2 * (1 - pt( abs(tstat), ndf )),
                       "lower" = pt(tstat, ndf),
                       "upper" = 1 - pt(tstat, ndf),
                       { warning( ``test must be 'two-sided', 'lower', or 'upper' " ) } )
  list( tstat = tstat, df = ndf, reject = if(!is.null(tail.area)) tail.area < alpha, tail.area =
      tail.area )
  }

x <- rnorm(30); y <- rnorm(35) + 1;
myttest( x, y )
myttest(x, y , test = "a")
```

- R에는 two sample *t*-test를 실행하는데 사용하는 `t.test`라는 함수가 있어요. R의 `t.test` 함수를 사용해서 테스트를 해보고, 같은 결과가 도출되는지 확인해 보세요.
- R의 `t.test` 함수는 어떻게 생겼을까요? 이 함수가 S3 객체임을 확인해 보고, 함수의 구체적인 내용을 출력해서 확인해보세요. 직접 만든 함수와 많이 다르게 생겼나요?

```
methods(t.test)
getAnywhere("t.test.default")
```

Exercise 3: Local vs Global Variables

- What is the main difference between $<-$ and $<<-$? Explain what changes in the global environment using the functions, `fun1` and `fun2`, below.

```
fun1 <- function(x){
  y <- 1 + x
  return(y)
}

fun2 <- function(x){
  y <<- 1 + x
  return(y)
}
```

- 함수를 만들 때 가장 흔한 실수 중 하나는 local variable과 global variable의 차이에 대한 이해가 부족해서 일어납니다. 다음 코드를 실행해 보고 함수 `fun()`에 어떤 문제가 있는지 생각해보세요.

```
x <- 1:10
fun <- function( z ){
  res <- z + x^2
  return(res)
}
fun( 10 )

rm( list = ls() )
fun( 10 )
```

Exercise 4: Regular Expression

강의 노트의 Regular Expression을 꼼꼼히 복습한 후 다음 문제를 풀어보세요. 정규표현식을 잘 이해해야 빅데이터 분석 및 전처리시 활용할 수 있습니다.

1. 아래의 코드를 실행하면 R 프로그램에 공헌을 한 저자들의 이름이 저장된 문자열들이 `authors` 객체에 부여되게 됩니다. 정규표현식을 사용해서 모든 저자들의 성(last name)들만 출력해 보세요.

```
dir(R.home("doc"))

authors <- readLines(file.path(R.home("doc"),"AUTHORS"))[-(1:8)]
```

2. What would be produced as a result if the R script below is conducted? What would be the value(s) assigned to `res`?

```
x <- c("a", 1, "d", FALSE)
res <- grep(pattern = "[^[:alpha:]]", x = x, value = FALSE)
if(res) cat("Not all alphabetic characters!")
```

3. 강의 노트의 word count 예제를 따라해 보고 각 line의 의미를 생각해보세요.

```
infile <- "ullyses.txt"
dat <- readLines(infile, n = 100)

words <- unlist(strsplit(dat, split = "[[:space:][:punct:]]"))

words <- tolower(words)

words[grep(pattern = "[0-9]", x = words)]
words[grep(pattern = "\\d", x = words)]

words <- gsub("[0-9]", "", words)

words <- words[words != ""]

wordcount <- table(words)

wordcount
```

Exercise 5: `apply()`, `tapply()`, `sapply()`, `lapply()`, `by()`

1. 다음의 사이트에서 https://www.r-bloggers.com/apply-lapply-rapply-sapply-functions-in-r 글을 읽고 아래의 각 함수에 대해 이해해 보세요.

```
apply()
tapply()
sapply()
lapply()
by()
```

2. 각 함수를 적용한 적절한 예 찾아 적어보세요 (위의 글에 사용된 예는 제외).

3. `tapply()` 함수는 hadoop의 MapReduce를 이해하는데 도움이 됩니다. `tapply()` 함수의 help file을 읽고 예제들을 따라서 실행해 보세요.

Exercise 6: Invoking R from the Command Line

- R 프로그램은 다양한 방법으로 사용할 수 있습니다. 지금까지 수업에서는 RStudio 프로그램 안에서 인터렉트 모드로 R을 사용해 왔지만, 윈도우 컴퓨터의 경우 명령프롬프트(cmd) 창에서 (맥의 경우 터미널(terminal) 창에서) R을 실행할 수 있어요. 학생들이 가장 많이 사용하는 윈도우 OS를 기준으로해서 설명할께요.
- How to invoke R from the command line
  1. Add R path to windows environment variable
     - 잘 모르는 사람은 위의 문구를 구글링해서 찾아보세요!
  2. Open a CMD window
     - 잘 모르면 역시 구글링!
  3. Type "R" to start R program
  4. Type "q()" if you want to quite R program
- 이제 R을 배치모드(Batch Mode)로 실행해 볼까요?
  1. 윈도우 컴퓨터에서 명령프롬프트(CMD) 창을 실행
  2. R script가 저장된 파일이 있는 directory로 이동함
     - 명령프롬프트 창에서 cd [path] 입력. 잘 모르면 구글링하세요!
  3. 명령프롬프트 창에 다음을 입력 (R scritp가 정장된 파일 이름이 "prac.R"이라고 가정)

     ```
     R CMD BATCH prac.R output out.txt
     ```

  4. R script가 실행된 모든 결과가 "out.txt" 파일 안에 저장됨 (결과 파일의 이름은 물론 자신이 원하는 이름으로 지정하면 됨)