SCR Exam 1: the Midterm

<Write Down Your Name Here, and ULCN Here>

1.1

1.1a

line 1: load the library of readxl line 2: create a vector with tab names i.e. c("Exam1", "Assignment", "Exam2") line 3: loop through the tab names line 4: assign the xlsx file location to file_name line 5: for each tab, read all rows, column 1 to 4 and assign the data to the corresponding variable, i.e. Exam1, Assignment, Exam2

1.1b

```
all.equal(names(Exam1), names(Assignment), names(Exam2))
```

```
## [1] "1 string mismatch"
```

1.1c

```
e1_id <- Exam1$ULCN
e2_id <- Exam2$ULCN
as_id <- Assignment$ULCN
total_id <- unique(c(e1_id ,e2_id,as_id))
length(total_id) #There are 49 unique students
```

```
## [1] 49
```

```
sum((total_id %in% e1_id) & (total_id %in% e2_id) & (total_id %in% as_id ))
```

```
## [1] 37
```

```
# 37 students took them all
```

1.1d

```
for (f in c(Exam1, Assignment, Exam2)) paste(str(f),'\n')
```

```
##  chr [1:47] "Tommy" "Francina" "Jeni" "Darell" "Estell" "Bridget" ...
##  chr [1:47] "Tejera" "Fegley" "Jameson" "Dipalma" "Ertl" "Brickey" ...
##  chr [1:47] "s1913973" "s1029448" "s2447584" "s2185592" "s1927199" ...
##  num [1:47] 47 61 47 72 58 94 72 75 92 67 ...
##  chr [1:41] "Francina" "Jeni" "Darell" "Estell" "Bridget" "Dean" ...
##  chr [1:41] "Fegley" "Jameson" "Dipalma" "Ertl" "Brickey" "Dahlen" ...
##  chr [1:41] "s1029448" "s2447584" "s2185592" "s1927199" "s2071403" ...
##  num [1:41] 69 91 54 80 53 51 50 74 41 71 ...
##  chr [1:40] "Francina" "Jeni" "Darell" "Estell" "Bridget" "Dean" ...
##  chr [1:40] "Fegley" "Jameson" "Dipalma" "Ertl" "Brickey" "Dahlen" ...
##  chr [1:40] "s1029448" "s2447584" "s2185592" "s1927199" "s2071403" ...
##  num [1:40] 100 56 50 60 79 49 67 59 100 88 ...
```

```
rm(f)
```

1.1e

```
mean(Exam1$`Final Grade`>=55) #74.5% pass Exam1
```

```
## [1] NaN
```

```r
mean(Assignment$`Final Grade`>=55) #75.6% pass Assignment
```

```
## [1] NaN
```

```r
mean(Exam2$`Final Grade`>=55) #87.5% pass Exam2
```

```
## [1] NaN
```

All statements are TRUE.

1.2

1.2a

```r
names(Exam1)[4] <- 'Exam1'
names(Assignment)[4] <- 'Assignment'
names(Exam2)[4] <- 'Exam2'
```

1.2b

```r
temp <- merge(Exam1, Assignment,all = T)
```

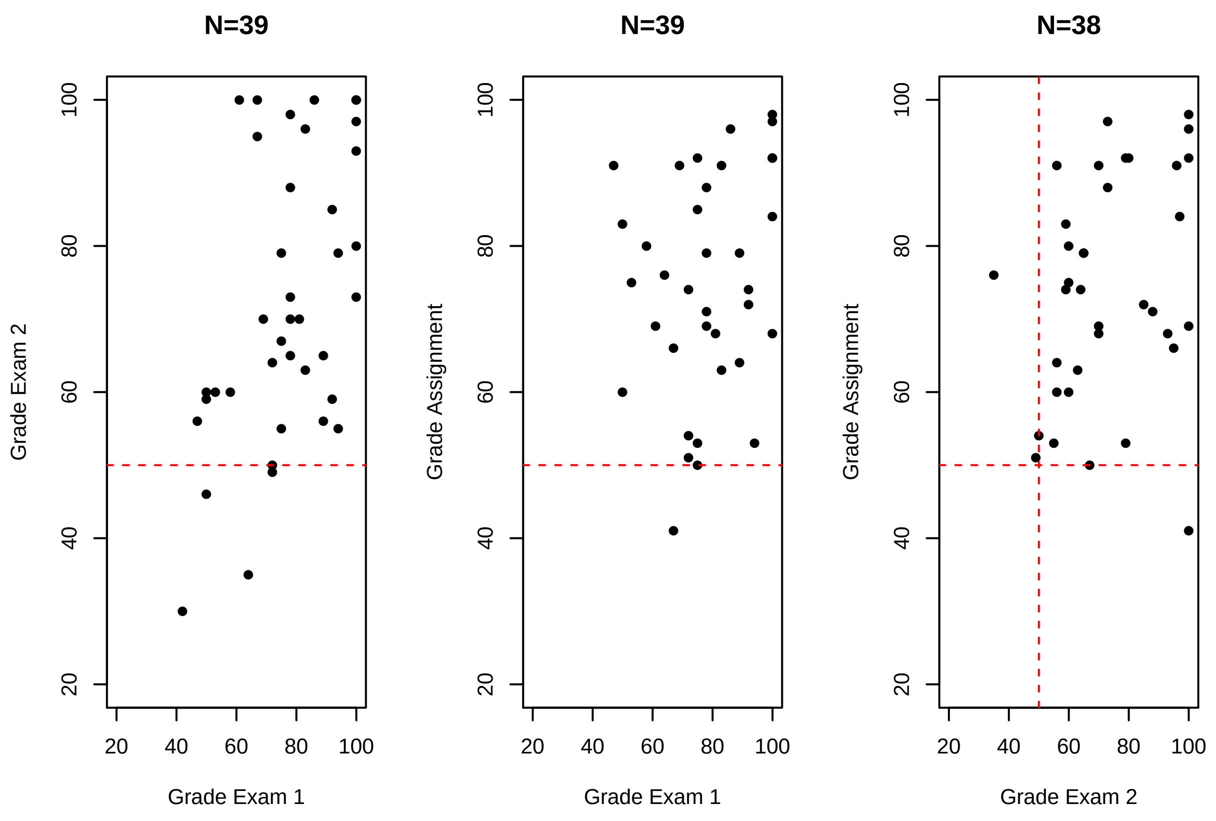
1.2c

```r
all_grade <- merge(temp, Exam2, all = T)
rm(temp)
all.equal(all_grade,the_grades)
```

```
## [1] TRUE
```

1.3

```r
par(mfrow=c(1, 3))
plot(all_grade$Exam2 ~ all_grade$Exam1, pch= 16,main = 'N=39',ylim=c(20,100),
     xlim=c(20,100),xlab='Grade Exam 1', ylab ='Grade Exam 2')
abline(h=50, col='red', lty = 2)
plot(all_grade$Assignment ~ all_grade$Exam1, pch= 16,main = 'N=39',ylim=c(20,100),
     xlim=c(20,100),xlab='Grade Exam 1', ylab ='Grade Assignment')
abline(h=50, col='red', lty = 2)
plot(all_grade$Assignment ~ all_grade$Exam2, pch= 16,main = 'N=38',ylim=c(20,100),
     xlim=c(20,100),xlab='Grade Exam 2', ylab ='Grade Assignment')
abline(h=50, col='red', lty = 2)
abline(v=50, col='red', lty = 2)
```



1.4

1.4a

```
sum(rowSums(is.na(all_grade)[,c('Exam1', 'Assignment','Exam2')]) == 0)
```

```
## [1] 37
```

```
# 37 students attend all
```

1.4b

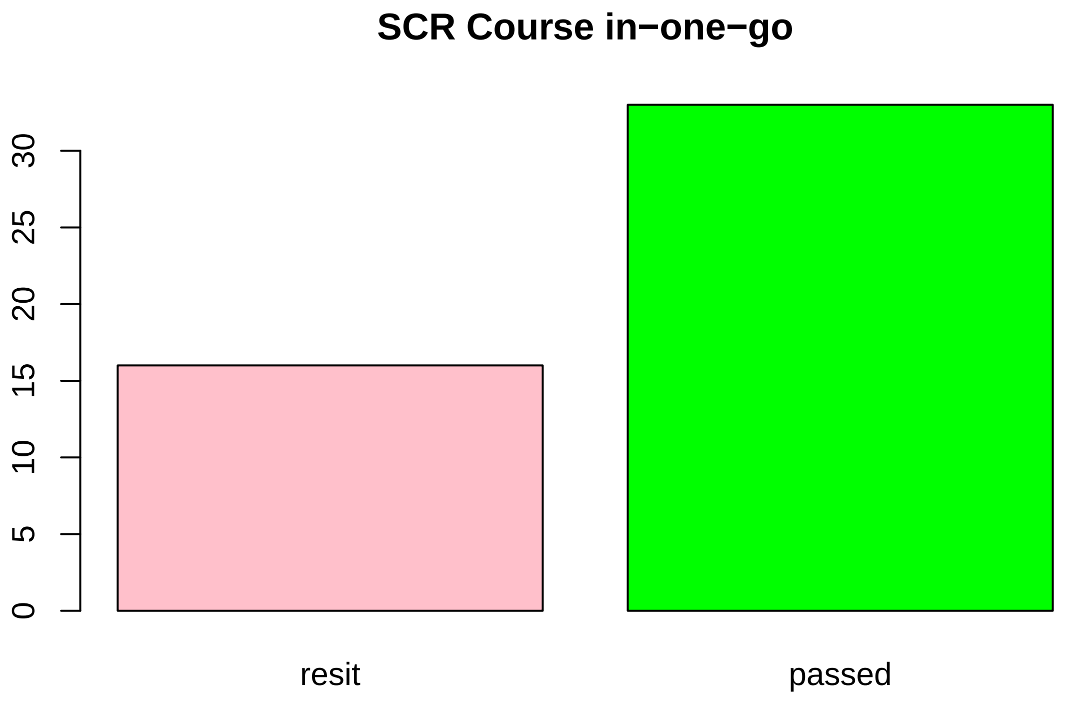
```
grade_analysis <- function(the_grades){
 list_grade <- list()
 E_1 <- the_grades$Exam1
 E_1[is.na(E_1)] <- 10
 E_2 <- the_grades$Exam2
 E_2[is.na(E_2)] <- 10
 E <- (E_2 + max(E_1, E_2)) / 2
 A <- the_grades$Assignment
 A[is.na(A)] <- 10
 Y <- (2*E + A)/3
 Y <- ifelse((E>=50 & E_2 >=50 & A>=50), Y, min(50,Y))
 list_grade[['if_pass']] <- factor(Y>=55, labels =c('resit','passed'))
 list_grade[['final_grade']] <- Y
 return(list_grade)
}
```

1.4c

```
grade_factor <- grade_analysis(the_grades)[['if_pass']]
table(grade_factor)
```

```
## grade_factor
##  resit passed
##     16     33
```

```
barplot(table(grade_factor),main='SCR Course in-one-go',col=c('pink','green'))
```



1.4d

```
tapply(
X = final_grades$CourseGrade,
INDEX = final_grades$pass,
FUN = mean)
```

```
##    resit     pass
## 33.14583 77.43434
```

2.1

2.1a

```
ULCN_nrs = 1000001:2499999
length(ULCN_nrs) # 1499999 numbers
```

```
## [1] 1499999
```

```
table(ULCN_nrs < 1700000) #800000 larger or equal, 699999 smaller
```

```
##
##  FALSE   TRUE
## 800000 699999
```

2.1b

```
prob_old <- 0.25 / 699999
prob_new <- 0.75 / 800000
probs_ULCN <- c(rep(prob_old,699999),rep(prob_new,800000))
sum(probs_ULCN) # equal to 1
```

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

2.1c

```
set.seed(20191030)
id_number <- sample(ULCN_nrs,49,prob=probs_ULCN)
```

```
# Another way to do this
set.seed(20191030)
population = c(rep(1000001:1699999,1),rep(1700000:2499999,3))
sample(population, 49, replace = F)
```

```
##  [1] 1794160 2116631 1844577 2490660 1423256 2061089 2003637 2164328
##  [9] 1880273 2120627 1936873 2493431 1946215 2373786 1724240 2077710
## [17] 1264230 1767247 2115580 1543840 1960860 2118549 1197305 1229172
## [25] 1856566 2250809 2436760 1863997 2449273 1003623 1637398 1835497
## [33] 1767891 1743916 2101635 2080556 1732299 1224804 1614123 1544143
## [41] 1427226 2047647 2288461 2269676 2316136 1771671 1896297 2285427
## [49] 2172916
```

2.1d

```
ran_ULCNs <- paste0('s',sort(id_number))
```

2.2

2.2a

```
df <- read.table('0_data/names.txt',sep=' ',stringsAsFactors=F,
                 col.names=c('voornaam','achternaam'))
str(df)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ voornaam  : chr  "Tommy" "Francina" "Jeni" "Darell" ...
##  $ achternaam: chr  "Tejera" "Fegley" "Jameson" "Dipalma" ...
```

```
key20191030 <- {
  set.seed(20191030)
  data.frame(
    decrypt = c(letters, LETTERS, 0:9, " ", NA, ".", "/", "-"),
    encrypt = sample(c(letters, LETTERS, 0:9, " ", NA, ".", "/", "-")),
    stringsAsFactors = FALSE
  )
}
```

```
cat(sapply(strsplit('Francina',split = "")[[1]], function(ch) key$encrypt[key$decrypt %in% ch]),sep='')

sapply(strsplit('Francina Love',split = "")[[1]], function(ch) key$encrypt[key$decrypt %in% ch])
```

2.2b

```
Encrypt <- function(names_decr, key = key20191030) { # names_decr <- student_names$V1
# key <- key20191030
raw_names <- strsplit(names_decr, split = "")
map_to_decr <- 1:nrow(key)
names(map_to_decr) <- key$decrypt
names_encr <- sapply(raw_names, function(decr_name) { # decr_name <- raw_names[[1]]
name_encr <- key$encrypt[map_to_decr[decr_name]]
name_encr <- paste(name_encr, collapse = "")
return(name_encr)
})
return(names_encr) }
```

```
Encrypt(c("Tommy", "Francina"))
```

```
## [1] "2755 "    "loVnQ/nV"
```

2.2c

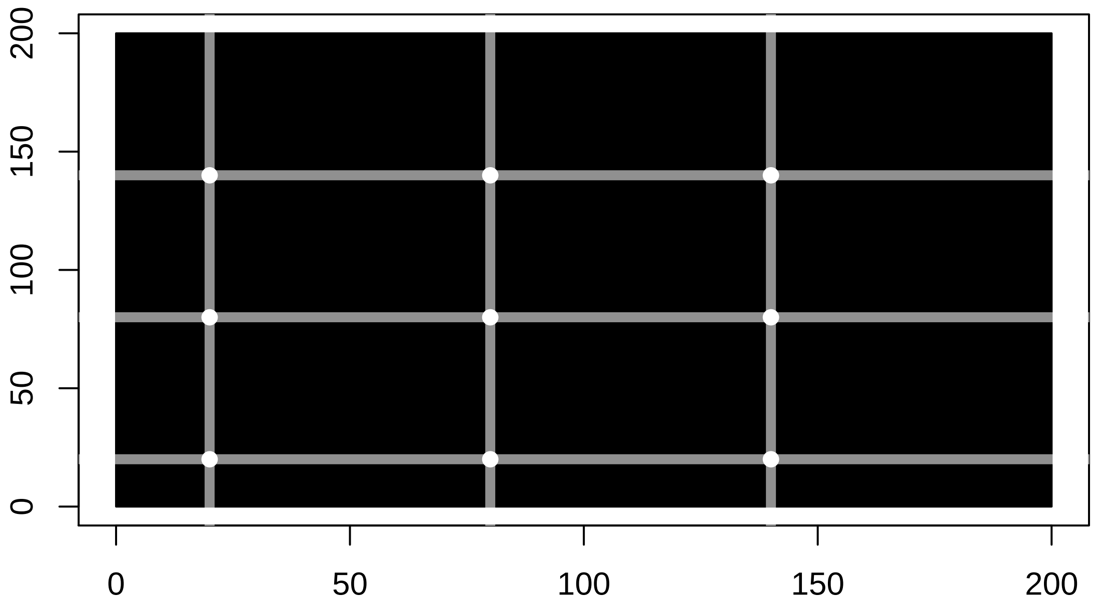
```
df_encrypted <- lapply(df,function(n) Surrogate(n,key20191030))
```

2.2d

```
write.table(df_encrypted, file="0_data/encry.txt",col.names=F, row.names = F,sep=" ", quote =F)
```

3

```r
plot(c(0, 200), c(0, 200), type= "n", xlab = "", ylab = "")
rect(0, 0, 200, 200,col='black')
abline(v=20, col='#FFFFFF90',lwd=5)
abline(v=80, col='#FFFFFF90',lwd=5)
abline(v=140, col='#FFFFFF90',lwd=5)
abline(h=20, col='#FFFFFF90',lwd=5)
abline(h=80, col='#FFFFFF90',lwd=5)
abline(h=140, col='#FFFFFF90',lwd=5)
points(rep(c(20,80,140),times=3),rep(c(20,80,140),each=3),col='white',pch=19)
```


0
50
100
150
200
0
50
100
150
200