```
# Source code for Homework 2

# Exercise 1

# Part 1

library(readr)
library(dplyr)
library(magrittr)
transplant <- read_table2("transplant.txt", col_names = FALSE)[-(1:7),(1:3)]
colnames(transplant) = c("t", "type", "survive")
transplant = as.data.frame(transplant)
transplant %<>% mutate_if(is.character, as.numeric)


# Part 2

library(SMPracticals)
library(ggfortify)
library(survival)
autoplot(survfit(Surv(t,survive)~type,data=transplant))


# Part 3

mod = survreg(Surv(t, survive) ~ type, data=transplant, dist = "exponential")
summary(mod)


# Part 5

plot(survfit(Surv(t,survive) ~ type, data = transplant), conf.int=TRUE, col=c(2,3), main =
"Exponential vs K-M fits")
x <- seq(from=0, to=60, by=0.1)
lines(x, 1-pexp(x,exp(-coef(mod)[1])), col="darkred", lwd=2)
lines(x, 1-pexp(x,exp(-sum(coef(mod)))), col="darkgreen", lwd=2)


# Part 6

mod_wei <- survreg(Surv(t,survive) ~ type, data = transplant)
gamma = 1/exp(mod_wei$scale)
plot(survfit(Surv(t,survive) ~ type, data = transplant), conf.int=TRUE, col=c(2,3),
main="Weibull v. K-M fits")
lines(x,1-pweibull(x, gamma, exp(coef(mod_wei)[1])),col="darkred",lwd=2)
lines(x,1-pweibull(x, gamma, exp(sum(coef(mod_wei)))),col="darkgreen",lwd=2)
```

```r
summary(mod_wei)


# Exercise 2

# Part 1

scores = read.delim("scores.txt", sep = " ")

# (a)
(cov_a <- cov(scores, use="complete.obs"))

# (b)
(cov_b <- cov(scores, use="pairwise.complete.obs"))

# (c)
scores_imp <- scores
for (i in 1:dim(scores)[2]){
  ind <- which(is.na(scores[,i]))
  scores_imp[ind, i] <- mean(na.omit(scores[,i]))
}
(cov_c <- cov(scores_imp))

# (d)
c <- matrix(0, nrow = dim(scores)[2], ncol = dim(scores)[2])
for (i in 1:1000){
  n <- dim(scores)[1]
  new_ind <- sample(1:n, size = n, replace = TRUE)
  scores_boot <- scores[new_ind,]
  for (j in 1:dim(scores)[2]){
    ind <- which(is.na(scores_boot[,j]))
    scores_boot[ind, j] <- mean(na.omit(scores_boot[,j]))
  }
  c <- c + cov(scores_boot)
}
(cov_d <- c/1000)

# (e)
library(TestDataImputation)
set.seed(1)
scores_em <- EMimpute(scores, max.score = 1000)
(cov_e <- cov(scores_em))


# Part 2

# (a)
```

```
z <- qnorm(0.975)
n <- dim(scores)[1]
lambda_a <- max(eigen(cov_a)$values)
lambda_a_low <- lambda_a/(z*sqrt(2/n)+1)
lambda_a_high <- lambda_a/(-z*sqrt(2/n)+1)
cat(paste0("The confidence interval for eigenvalue of complete case analysis is: (",
        round(lambda_a_low, 3), ", ", round(lambda_a_high, 3), ")."))

# (b)
lambda_b <- max(eigen(cov_b)$values)
lambda_b_low <- lambda_b/(z*sqrt(2/n)+1)
lambda_b_high <- lambda_b/(-z*sqrt(2/n)+1)
cat(paste0("The confidence interval for eigenvalue of available case analysis is: (",
        round(lambda_b_low, 3), ", ", round(lambda_b_high, 3), ")."))

# (c)
lambda_c <- max(eigen(cov_c)$values)
lambda_c_low <- lambda_c/(z*sqrt(2/n)+1)
lambda_c_high <- lambda_c/(-z*sqrt(2/n)+1)
cat(paste0("The confidence interval for eigenvalue of mean imputation is: (",
        round(lambda_c_low, 3), ", ", round(lambda_c_high, 3), ")."))

# (d)
lambda_d <- max(eigen(cov_d)$values)
lambda_d_low <- lambda_d/(z*sqrt(2/n)+1)
lambda_d_high <- lambda_d/(-z*sqrt(2/n)+1)
cat(paste0("The confidence interval for eigenvalue of mean imputation with bootstrap is:
(",
        round(lambda_d_low, 3), ", ", round(lambda_d_high, 3), ")."))

# (e)
lambda_e <- max(eigen(cov_e)$values)
lambda_e_low <- lambda_e/(z*sqrt(2/n)+1)
lambda_e_high <- lambda_e/(-z*sqrt(2/n)+1)
cat(paste0("The confidence interval for eigenvalue of EM-algorithm is: (",
        round(lambda_e_low, 3), ", ", round(lambda_e_high, 3), ")."))


# Part 3

library(SMPracticals)
cov_comp <- cov(mathmarks)

lambda_comp <- max(eigen(cov_comp)$values)
lambda_comp_low <- lambda_comp/(z*sqrt(2/n)+1)
lambda_comp_high <- lambda_comp/(-z*sqrt(2/n)+1)
cat(paste0("The confidence interval for eigenvalue of the complete data is: (",
```

```
    round(lambda_comp_low, 3), ", ", round(lambda_comp_high, 3), ").")) 
```

```
# Exercise 3

# Part 3

library(stringr)
data <- read.csv('CentralPark.csv')
levels(data$NAME)
july_date <- c()
july_prcp <- c()
for (i in 1:nrow(data)){
  if (substr(data$DATE[i],1,1) == 7){
    july_date <- c(july_date, toString(data$DATE[i]))
    july_prcp <- c(july_prcp, data$PRCP[i])
  }
}

prcp <- data.frame(DATE = july_date, PRCP = july_prcp, RAIN =
numeric(length(july_prcp)))
for (i in 1:nrow(prcp)){
  if (prcp$PRCP[i] > 1.5){
    prcp$RAIN[i] = 0
  }
  else{
    prcp$RAIN[i] = 1
  }
}
count00 <- 0
count01 <- 0
count10 <- 0
count11 <- 0
for (i in 1:(nrow(prcp)-1)){
  this_year <- str_sub(prcp$DATE[i],-2,-1)
  next_year <- str_sub(prcp$DATE[i+1],-2,-1)
  if (this_year == next_year){
    if (prcp$RAIN[i] == 0 && prcp$RAIN[i+1] == 0){
      count00 <- count00 + 1
    }
    else if (prcp$RAIN[i] == 0 && prcp$RAIN[i+1] == 1){
      count01 <- count01 +1
    }
    else if (prcp$RAIN[i] == 1 && prcp$RAIN[i+1] == 0){
      count10 <- count10 + 1
    }
    else if (prcp$RAIN[i] == 1 && prcp$RAIN[i+1] == 1){
```

```
      count11 <- count11 + 1
    }
  }
}
c(count00, count01, count10, count11)

# Part 4

p00 <- 0.2971098
p11 <- 0.7763401
p01 <- 0.7028902
p10 <- 0.2236599
n0 <- 865
n1 <- 2705
pp <- (n0*p00+n1*p11)/(n0+n1)
pnorm((p00-p11)/sqrt(pp*(1-pp)*(1/n0+1/n1)))

# Part 5

count000 <- 0
count001 <- 0
count010 <- 0
count011 <- 0
count100 <- 0
count101 <- 0
count110 <- 0
count111 <- 0

for (i in 1:(nrow(prcp)-2)){
  this_year <- str_sub(prcp$DATE[i],-2,-1)
  next_year <- str_sub(prcp$DATE[i+1],-2,-1)
  one_more_year <- str_sub(prcp$DATE[i+2],-2,-1)
  if (this_year == next_year && next_year == one_more_year){
    if (prcp$RAIN[i] == 0 && prcp$RAIN[i+1] == 0 && prcp$RAIN[i+2] == 0){
      count000 <- count000 + 1
    }
    else if (prcp$RAIN[i] == 0 && prcp$RAIN[i+1] == 0 && prcp$RAIN[i+2] == 1){
      count001 <- count001 + 1
    }
    else if (prcp$RAIN[i] == 0 && prcp$RAIN[i+1] == 1 && prcp$RAIN[i+2] == 0){
      count010 <- count010 +1
    }
    else if (prcp$RAIN[i] == 0 && prcp$RAIN[i+1] == 1 && prcp$RAIN[i+2] == 1){
      count011 <- count011 + 1
    }
    else if (prcp$RAIN[i] == 1 && prcp$RAIN[i+1] == 0 && prcp$RAIN[i+2] == 0){
      count100 <- count100 + 1
```

```
    }
    else if (prcp$RAIN[i] == 1 && prcp$RAIN[i+1] == 0 && prcp$RAIN[i+2] == 1){
      count101 <- count101 + 1
    }
    else if (prcp$RAIN[i] == 1 && prcp$RAIN[i+1] == 1 && prcp$RAIN[i+2] == 0){
      count110 <- count110 + 1
    }
    else if (prcp$RAIN[i] == 1 && prcp$RAIN[i+1] == 1 && prcp$RAIN[i+2] == 1){
      count111 <- count111 + 1
    }
  }
}

p000 <- count000/(count000+count001)
p001 <- count001/(count000+count001)
p010 <- count010/(count010+count011)
p011 <- count011/(count010+count011)
p100 <- count100/(count100+count101)
p101 <- count101/(count100+count101)
p110 <- count110/(count110+count111)
p111 <- count111/(count110+count111)
c(count000,count001,count010,count011,count100,count101,count110,count111)
c(p000,p001,p010,p011,p100,p101,p110,p111)

obs_TS <- 2 * (count000 * log(p000/p00) + count001 * log(p001/p01) +
               count010 * log(p010/p10) + count011 * log(p011/p11) +
               count100 * log(p100/p00) + count101 * log(p101/p01) +
               count110 * log(p110/p10) + count111 * log(p111/p11))
obs_TS

pchisq(obs_TS, df = 2)


# Exercise 5

# Part 5

library(readr)
library(dplyr)
library(magrittr)
transplant <- read_table2("transplant.txt", col_names = FALSE)[-(1:7),(1:3)]
colnames(transplant) = c("t", "type", "survive")
transplant = as.data.frame(transplant)
transplant %<>% mutate_if(is.character, as.numeric)
transplant <- transplant[order(transplant$t),]

ka_ind <- which(transplant$survive == 1)
```

```r
y <- c()
e <- c()
v <- c()
for (i in 1:length(ka_ind)){
  if (transplant$type[ka_ind[i]] == 1){
    y <- c(y, 1)
  }else{
    y <- c(y, 0)
  }
  rest <- transplant[ka_ind[i]:nrow(transplant),]
  n_a <- sum(rest$type == 1)
  n_b <- sum(rest$type == 2)
  n <- n_a + n_b
  e <- c(e, n_a/n)
  v <- c(v, n_a*n_b*(n-1)/(n*n*(n-1)))
}
z <- sum(y-e)/sqrt(sum(v))
cat("The p-value is :", pnorm(z)*2)
```