Assignment 3

James Lopez

October 11, 2016

Grade: 48/50

2

part 1

```
set.seed(100)

loops <- 1e3
exp.pvals <- numeric(loops)
for(i in seq(loops)) {
  group <- rnorm(100, 60, 20)
  assign <- sample(c(0,1), 100, replace = TRUE)
  exp <- data.frame(group, assign)
  for(x in 1:100) {
    if(exp[x,'assign'] == 1) {
      exp[x,1] <- exp[x,1] +5
    }
    x <- x+1
  }
  mod.sim <- lm(group~assign, data = exp)
  summ <- summary(mod.sim)
  coefs <- coef(summ)
  exp.pvals[i] <- coefs[2,4]
  i <- i+1
}

sum(exp.pvals<=0.05)/loops
```

```
## [1] 0.245
```

The power for 100 samples is 0.231.

part 2

```
set.seed(1000)
loops_2 <- 1e3
exp.pvals_2 <- numeric(loops_2)
for(j in seq(loops_2)) {
  group_2 <- rnorm(1000, 60, 20)
  assign_2 <- sample(c(0,1), 1000, replace = TRUE)
  exp_2 <- data.frame(group_2, assign_2)
  for(y in 1:1000) {
```

```
    if(exp_2[y,'assign_2'] == 1) {
      exp_2[y,1] <- exp_2[y,1] +5
    }

  }
  mod.sim_2 <- lm(group_2~assign_2, data = exp_2)
  summ_2 <- summary(mod.sim_2)
  coefs_2 <- coef(summ_2)
  exp.pvals_2[j] <- coefs_2[2,4]

}

sum(exp.pvals_2<=0.05)/loops_2
```

```
## [1] 0.98
```

The power for 1000 sampes is 0.98.

3

The football-values data set is stored in data frame footbal, which is then converted into football with the first two columns of the original data set removed.

```
footbal <- read.csv('proj_wr16.csv')
football <- footbal[,-1:-2]
```

part 1

The correlation matrix of football is shown using:

```
cor(football)
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##            fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

part 2

To make a data set with a similar correlation structure, we use mvrnorm since the football data set consists of multiple variables and corresponding means and covariances. The resulting correlation matrix of the simulated data is contained in keep.i.

```
library(MASS)
rho <- cor(football)
vcov <- var(football)
means <- colMeans(football)

keep.i <- 0
loops <- 1e4
for(i in seq(loops)) {
  football.sim <- mvrnorm(30, mu=means, Sigma=vcov)
  keep.i <- keep.i + cor(football.sim)/loops
}

keep.i
```

```
##           rush_att  rush_yds   rush_tds    rec_att   rec_yds    rec_tds
## rush_att 1.0000000 0.9902341 0.88241167 0.19362908 0.1418744 0.13289492
## rush_yds 0.9902341 1.0000000 0.90964176 0.18401614 0.1348112 0.12530075
## rush_tds 0.8824117 0.9096418 1.00000000 0.06921105 0.0315649 0.03213981
## rec_att  0.1936291 0.1840161 0.06921105 1.00000000 0.9896449 0.96653342
## rec_yds  0.1418744 0.1348112 0.03156490 0.98964486 1.0000000 0.98149861
## rec_tds  0.1328949 0.1253007 0.03213981 0.96653342 0.9814986 1.00000000
## fumbles  0.1827721 0.1861464 0.10810836 0.42843163 0.3965239 0.35290690
## fpts     0.1732943 0.1665474 0.06553721 0.98711766 0.9975202 0.99026166
##            fumbles       fpts
## rush_att 0.1827721 0.17329432
## rush_yds 0.1861464 0.16654738
## rush_tds 0.1081084 0.06553721
## rec_att  0.4284316 0.98711766
## rec_yds  0.3965239 0.99752024
## rec_tds  0.3529069 0.99026166
## fumbles  1.0000000 0.37620335
## fpts     0.3762033 1.00000000
```

part 3

The procedure for making a simulated data set with the SAME correlation structure as the football data set is the same, except we include the argument 'empirical' and set it equal to TRUE to show that the means and variances we use to create our simulated data are empirical means and variances rather than true means and variances.

```
keep.j <- 0
loops <- 1e4
for(j in seq(loops)) {
  football.sim <- mvrnorm(30, mu=means, Sigma=vcov, empirical = TRUE)
  keep.j <- keep.j + cor(football.sim)/loops
}
keep.j
```

```
##          rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##           fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

JC Grading -2 Looking for a single generated dataset.

Question 4

part 1

$$P(B) = \sum_j P(B|A_j)P(A_j), \Rightarrow P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \tag{1}$$

part 2

$$\hat{f}(\zeta) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x \zeta} dx \tag{2}$$

part 3

$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = \left[\frac{\partial \mathbf{f}}{\partial x_1} \cdots \frac{\partial \mathbf{f}}{\partial x_n}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{3}$$