Bios 6301: Assignment 3

Dannielle Gibson

Due Tuesday, 27 September, 1:00 PM

50 points total.

Add your name as `author` to the file's metadata section.

Submit a single knitr file (named `homework3.rmd`) by email to tianyi.sun@vanderbilt.edu. Place your R code in between the appropriate chunks for each question. Check your output by using the `Knit HTML` button in RStudio.

$5^{n=day}$ points taken off for each day late.

Question 1

15 points

Write a simulation to calculate the power for the following study design. The study has two variables, treatment group and outcome. There are two treatment groups (0, 1) and they should be assigned randomly with equal probability. The outcome should be a random normal variable with a mean of 60 and standard deviation of 20. If a patient is in the treatment group, add 5 to the outcome. 5 is the true treatment effect. Create a linear model for the outcome by the treatment group, and extract the p-value (hint: see assigment1). Test if the p-value is less than or equal to the alpha level, which should be set to 0.05.

Repeat this procedure 1000 times. The power is calculated by finding the percentage of times the p-value is less than or equal to the alpha level. Use the `set.seed` command so that the professor can reproduce your results.

1. Find the power when the sample size is 100 patients. (10 points) power is 239/1000*100= 23.9%.

```
n.sim =1000

n <- 100
mean <- 60
sd <- 20
p_value = NULL
treatment.groups <-c(0,1)

set.seed(2)
for (j in 1:n.sim) {
t.s <- sample(treatment.groups, n, replace=TRUE)
outcome<- rnorm(n, mean, sd)
for (i in 1:n) {
  if (t.s[i]==1){
    outcome[i]<- outcome[i] +5}
  else {outcome[i] <- outcome[i] }
}
```

```
summary(coef(lm(outcome~t.s)))
p_value[j] = coef(summary(lm(outcome~t.s)))[2,4]
}
p_value <0.05
```

```
##    [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##   [13] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [25] FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
##   [37] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##   [49] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [61] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
##   [73] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
##   [85] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [97]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [109] FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##  [121]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
##  [133] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
##  [145] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
##  [157]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [169] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
##  [181] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [205] FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [217] FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [229] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [241] FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [253] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [265]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##  [289] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [301] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [313] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##  [325] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [337] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
##  [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [361]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [373] FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [421] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [433] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [445] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [457] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [469] FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [481]  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
##  [493] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
##  [505] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [517] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [529] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [541] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
##  [553] FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [565] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
```

```
##  [577] FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [589] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##  [601]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [613] FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
##  [637]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [649]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [661] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [673] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [685]  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##  [697] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [709] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [721]  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE
##  [733] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
##  [745] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [757]  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [769] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [781]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [793]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [805] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [817]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
##  [829] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [841] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE
##  [853] FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [865] FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [877]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
##  [889] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##  [901] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [913]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
##  [925] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##  [937] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [949] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [961] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE
##  [973] FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##  [985] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [997]  TRUE FALSE FALSE FALSE
```

```
sum(p_value <0.05)
```

```
## [1] 239
```

1. Find the power when the sample size is 1000 patients. (5 points) power is 46/1000*100= 4.6%

```
n2.sim =1000

n2 <- 1000
mean2 <- 60
sd2 <- 20
p_value2 = NULL
treatment.groups2 <-c(0,1)

set.seed(2)
for (j2 in 1:n2.sim) {
```

```r
t.s2 <- sample(treatment.groups2, n2, replace=TRUE)
outcome2<- rnorm(n2, mean2, sd2)
for (i2 in 1:n2) {
  if (t.s2[i2]==1){
    outcome2[i]<- outcome2[i2] +5}
  else {outcome2[i2] <- outcome2[i2] }
}
summary(coef(lm(outcome2~t.s2)))
p_value2[j2] = coef(summary(lm(outcome2~t.s2)))[2,4]
}
p_value2 <0.05
```

```
##    [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##   [97] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [133]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [205] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [253] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [265] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [277] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [289] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [313]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [325] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##  [337] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [349]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [409] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [469] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [481] FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
##  [493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [505] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [517] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [529] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [553] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [565] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [577] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [589] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [613] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [637]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [661] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [673]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [685] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [697] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [709] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [733] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [745] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [757] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [769] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [781] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [793] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [805] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [817] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [829] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [841] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [853] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [865] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [877] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [889] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [901] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [913] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [925] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [937] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [949] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [961] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [973]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [985] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [997] FALSE FALSE FALSE FALSE
```

```
sum(p_value2 <0.05)
```

```
## [1] 46
```

Question 2

14 points

Obtain a copy of the football-values lecture. Save the `2021/proj_wr21.csv` file in your working directory. Read in the data set and remove the first two columns.

1. Show the correlation matrix of this data set. (4 points)

```r
getwd()
```

```
## [1] "C:/Users/gibsondk/Dannielle-Gibson-BIOS6301"
```

```r
proj_wr21<- data.frame(read.csv(file = "proj_wr21(original).csv"))
df<- proj_wr21[-c(1,2)]
df
```

```
##     rec_att rec_yds rec_tds rush_att rush_yds rush_tds fumbles  fpts
## 1     100.2  1436.2    12.3     12.9     82.4      0.8     0.7 229.2
## 2     122.5  1462.5    12.8      0.0      0.0      0.0     0.8 221.3
## 3     113.5  1474.5     9.3      2.4     15.6      0.1     1.2 202.4
## 4      99.3  1415.3     8.8      4.6     26.2      0.1     0.8 196.1
## 5     122.1  1467.7     7.6      0.9      4.4      0.0     1.5 190.2
## 6      86.6  1301.3     9.9      0.3      1.3      0.0     1.4 187.2
## 7      93.5  1392.3     7.9      0.4      1.5      0.0     0.8 185.2
## 8      84.5  1267.9     9.8      0.5      7.5      0.0     0.9 184.5
## 9     111.2  1208.6     7.9      3.0     16.3      0.1     1.0 168.4
## 10     76.1  1113.5     9.4      0.0      0.0      0.0     0.7 166.5
## 11     91.5  1235.7     6.5      1.8     12.1      0.1     0.7 162.9
## 12     98.3  1223.4     6.9      0.3      0.3      0.0     0.7 162.6
## 13     89.8  1187.7     7.2      2.6     11.7      0.0     0.7 161.9
## 14     79.6  1021.5     9.2      3.2     14.6      0.3     0.7 158.9
## 15     97.3  1080.4     6.0     17.0    106.5      0.9     0.8 158.4
## 16     84.6  1099.7     6.6      8.5     61.2      0.6     0.8 157.7
## 17     93.4  1063.1     8.4      1.0      2.7      0.0     0.7 155.6
## 18     77.8  1102.8     7.2      0.5     -0.5      0.0     0.7 152.3
## 19     82.3  1186.1     5.4      3.1     20.1      0.1     0.8 151.9
## 20     92.8  1066.6     7.1      3.9     23.7      0.1     0.7 150.9
## 21     82.1  1067.3     7.5      0.3      2.0      0.0     0.8 150.1
## 22     79.2  1058.3     6.4      2.3     12.4      0.0     0.8 144.0
## 23     71.1   994.3     6.5      5.6     50.6      0.3     0.7 143.7
## 24     73.1   926.3     6.5     14.6     82.4      0.7     0.7 142.8
## 25     69.9  1033.1     6.3      0.0      0.0      0.0     0.7 139.9
## 26     93.9   981.0     6.6      4.4     27.4      0.1     1.4 137.7
## 27     71.9  1041.7     5.8      1.5      8.2      0.0     1.2 137.7
## 28     69.6   892.1     6.8      8.3     44.3      0.6     0.9 136.6
## 29     70.0   992.9     6.3      1.7      9.4      0.0     0.7 136.6
## 30     75.3  1052.5     5.2      0.8      6.8      0.0     0.7 135.9
## 31     73.1  1030.8     5.5      0.0      0.0      0.0     0.8 134.5
## 32     70.4   898.6     4.7     13.5     99.3      0.7     0.8 130.8
## 33     80.3   991.3     5.2      2.5     12.3      0.1     0.8 130.6
## 34     89.3   901.9     6.7      0.0      0.0      0.0     0.9 128.6
## 35     70.6   941.6     5.9      0.3      3.3      0.0     0.6 128.4
## 36     64.1   941.7     5.9      0.0      0.0      0.0     0.6 128.3
## 37     84.0   932.0     5.5      2.1     15.7      0.0     0.8 126.5
## 38     56.7   925.1     5.7      0.3      0.5      0.0     0.6 125.8
## 39     75.0   825.8     4.9     19.7    108.7      0.5     0.7 124.5
## 40     68.7   876.4     6.3      0.3      0.0      0.0     0.6 124.3
## 41     61.2   824.6     7.0      1.5      8.0      0.0     0.6 124.1
## 42     62.3   735.2     4.8     24.6    128.5      1.1     0.7 120.4
```

```
## 43     73.6   885.3     5.0      1.9      9.7       0.3      0.8 120.1
## 44     63.7   895.8     5.1      1.0      6.2       0.0      1.2 118.3
## 45     66.6   849.1     5.3      0.0      0.0       0.0      0.6 115.6
## 46     65.5   833.0     5.5      0.0      0.0       0.0      0.7 115.0
## 47     71.8   857.0     4.9      0.3     -0.8       0.0      0.7 113.5
## 48     63.2   824.7     4.5      3.4     22.7       0.0      0.7 110.8
## 49     68.2   887.1     3.9      0.0      0.0       0.0      0.8 110.5
## 50     52.1   796.1     4.4     10.4     58.3       0.1      1.7 109.2
## 51     52.4   732.1     5.5      6.4     43.8       0.1      1.7 108.0
## 52     63.5   711.5     6.0      2.0     11.4       0.0      0.6 106.9
## 53     65.5   778.0     4.5      2.7     15.5       0.0      0.7 105.1
## 54     56.0   802.5     4.3      0.0      0.0       0.0      0.7 104.7
## 55     51.8   788.9     4.1      1.4      8.7       0.0      0.0 104.6
## 56     57.1   764.6     4.7      1.5      8.6       0.0      1.2 103.2
## 57     56.1   743.0     5.0      0.0      0.0       0.0      0.6 102.9
## 58     57.1   696.3     5.0      1.1      6.7       0.1      0.6  99.1
## 59     64.0   737.8     4.4      0.0      0.0       0.0      0.6  99.1
## 60     54.5   730.9     4.1      3.1     17.5       0.1      0.7  98.3
## 61     50.6   714.5     4.5      0.3      1.3       0.0      0.7  97.2
## 62     66.3   720.9     4.1      1.3      6.1       0.0      0.6  95.9
## 63     59.5   719.1     3.7      0.7      4.2       0.0      0.4  93.9
## 64     42.7   629.5     5.0      0.3      0.0       0.0      0.1  92.8
## 65     56.1   714.2     3.5      0.7      3.6       0.0      0.5  91.9
## 66     42.1   596.1     5.1      0.3      0.5       0.0      0.1  89.8
## 67     49.5   684.3     3.8      0.5      2.0       0.0      1.0  89.3
## 68     49.8   643.3     4.1      0.0      0.0       0.0      0.4  88.4
## 69     62.1   636.4     3.5      3.3     26.0       0.3      0.6  87.7
## 70     51.7   633.6     4.2      0.0      0.0       0.0      0.7  86.9
## 71     50.6   612.6     4.3      1.5      7.6       0.0      0.7  86.6
## 72     54.1   663.7     3.6      1.1      6.1       0.0      1.2  86.3
## 73     48.6   632.4     3.5      0.5      2.8       0.0      1.2  82.4
## 74     54.0   605.4     3.7      0.3      3.3       0.0      0.7  81.5
## 75     45.1   545.3     3.8      3.9     26.9       0.0      0.6  79.4
## 76     46.1   559.2     3.3      4.8     30.1       0.2      0.4  78.9
## 77     35.5   561.0     3.6      2.8     17.1       0.1      0.6  78.8
## 78     41.1   543.0     4.1      0.8      4.8       0.0      0.6  78.0
## 79     40.8   551.3     3.7      1.3      9.7       0.0      0.6  76.9
## 80     42.4   558.9     3.4      0.0      0.0       0.0      0.3  75.7
## 81     42.5   566.1     3.3      0.0      0.0       0.0      0.3  75.6
## 82     41.2   556.5     3.5      0.8      2.6       0.0      0.6  75.5
## 83     46.1   545.8     2.8      2.2     10.3       0.1      0.8  71.5
## 84     40.3   487.2     3.4      1.4      5.8       0.0      0.1  69.8
## 85     43.4   519.6     3.1      0.0      0.0       0.0      0.6  69.2
## 86     37.5   525.1     2.7      0.0      0.0       0.0      0.7  67.4
## 87     43.0   522.8     2.6      0.0      0.0       0.0      0.3  67.2
## 88     37.6   462.0     3.5      0.8      7.5       0.0      0.6  66.9
## 89     36.1   469.4     3.1      0.8      4.5       0.0      0.6  64.9
## 90     25.7   438.1     3.3      2.5     14.8       0.0      0.6  64.4
## 91     33.3   451.7     2.9      0.0      0.0       0.0      0.6  61.4
## 92     30.7   415.3     3.3      0.3      2.0       0.0      0.1  61.4
## 93     35.7   460.9     2.7      0.0      0.0       0.0      0.5  61.1
## 94     35.3   423.3     3.3      0.0      0.0       0.0      0.6  60.9
## 95     16.0   188.0     1.0     54.0    225.0       2.0      0.0  59.3
## 96     32.6   391.7     2.4      8.9     43.9       0.2      0.6  57.9
```

```
## 97     31.6  449.8    2.0    0.3    0.8    0.0    0.1  57.2
## 98     27.4  419.3    2.3    2.5   10.6    0.1    0.1  56.9
## 99     22.8  369.9    2.6    2.5   15.0    0.0    0.0  54.3
## 100    26.5  395.9    2.6    0.0    0.0    0.0    0.6  53.8
## 101    32.7  373.2    2.6    1.1    5.5    0.0    0.0  53.5
## 102    27.4  381.3    2.4    0.5    3.0    0.0    0.6  51.5
## 103    28.5  375.1    2.5    0.0    0.0    0.0    0.6  51.2
## 104    26.8  392.2    2.0    0.3    1.4    0.0    0.1  50.9
## 105    29.4  351.5    1.9    6.5   39.7    0.2    0.3  50.9
## 106    28.3  379.0    2.1    0.0    0.0    0.0    0.0  50.7
## 107    35.6  371.9    2.2    1.3    5.7    0.0    0.5  49.7
## 108    35.2  411.0    2.1    1.0    5.5    0.5    4.0  49.5
## 109    27.3  371.5    2.1    0.0    0.0    0.0    0.2  49.5
## 110    29.3  372.1    2.0    0.0    0.0    0.0    0.2  49.2
## 111    23.6  303.2    2.9    0.0    0.0    0.0    0.6  46.3
## 112    26.5  350.3    1.8    0.0    0.0    0.0    0.2  45.7
## 113    27.0  304.7    1.6    7.3   47.9    0.1    0.2  45.2
## 114    23.7  311.8    2.2    0.0    0.0    0.0    0.2  44.1
## 115    24.1  427.0    1.4    1.5    0.0    0.0    4.0  43.2
## 116    29.5  339.7    1.7    0.3    2.0    0.0    0.6  43.1
## 117    30.3  331.0    1.6    0.0    0.0    0.0    0.1  42.4
## 118    24.2  328.6    1.4    0.0    0.0    0.0    0.2  41.1
## 119    22.6  290.2    2.0    0.0    0.0    0.0    0.2  40.8
## 120    17.0  244.6    1.8    1.4    7.7    0.1    0.1  36.5
## 121    21.4  270.4    1.5    0.0    0.0    0.0    0.1  35.8
## 122    18.9  200.3    1.9    5.7   28.6    0.2    0.1  35.5
## 123    18.3  275.6    1.5    1.0    4.0    0.0    1.0  35.0
## 124    19.2  249.1    1.2    0.0    0.0    0.0    0.1  31.9
## 125    19.6  231.1    1.5    0.0    0.0    0.0    0.0  31.8
## 126    18.0  221.4    1.3    2.0   12.4    0.1    0.1  31.5
## 127    19.8  233.0    1.2    0.0    0.0    0.0    0.1  30.4
## 128    20.2  233.1    1.2    0.0    0.0    0.0    0.2  30.3
## 129    16.2  196.4    1.4    0.9    5.2    0.0    0.1  28.8
## 130    15.2  190.4    1.5    0.0    0.0    0.0    0.1  27.8
## 131    17.2  190.2    1.3    1.7   11.7    0.0    0.1  27.6
## 132    15.7  207.4    1.1    0.0    0.0    0.0    0.1  27.0
## 133    15.4  189.9    1.1    0.0    0.0    0.0    0.0  25.6
## 134    13.4  154.7    0.8    6.2   33.9    0.2    0.1  24.7
## 135    13.2  138.0    0.9    8.5   39.1    0.1    0.1  23.2
## 136    14.5  155.5    1.2    0.7    3.6    0.0    0.1  23.2
## 137    10.6  120.7    0.9    2.5   14.9    0.1    0.1  19.3
## 138    10.2  133.3    0.9    0.0    0.0    0.0    0.1  18.6
## 139    15.0  146.7    0.7    0.0    0.0    0.0    0.1  18.5
## 140    10.1  108.6    0.6    4.6   26.1    0.1    0.1  17.7
## 141    12.4  145.1    0.5    0.0    0.0    0.0    0.1  17.4
## 142    11.3  113.1    0.7    3.0   19.2    0.1    0.2  17.3
## 143    11.0  136.6    0.6    0.0    0.0    0.0    0.1  17.2
## 144     9.1  123.4    0.7    0.8    4.9    0.0    0.1  17.1
## 145    11.2  131.1    0.7    0.0    0.0    0.0    0.1  17.0
## 146     9.7  126.7    0.7    0.0    0.0    0.0    0.1  16.6
## 147     8.7  110.2    0.8    0.0    0.0    0.0    0.1  15.8
## 148     9.0  104.2    0.8    1.3    6.7    0.0    0.0  15.6
## 149     9.2  114.8    0.6    0.0    0.0    0.0    0.0  15.2
## 150     8.9  109.1    0.7    0.0    0.0    0.0    0.0  14.8
```

```
## 151      9.6   107.1     0.6       0.0       0.0       0.0      0.0  14.4
## 152      8.6   105.9     0.6       0.0       0.0       0.0      0.1  14.3
## 153      5.6    70.7     0.4       4.7      28.9       0.2      0.1  13.5
## 154      8.0    99.9     0.6       0.0       0.0       0.0      0.1  13.5
## 155      8.2   100.6     0.5       0.0       0.0       0.0      0.1  13.2
## 156      7.5    93.7     0.6       0.0       0.0       0.0      0.1  12.7
## 157      6.9    90.6     0.6       0.0       0.0       0.0      0.0  12.5
## 158      7.1    86.7     0.6       0.0       0.0       0.0      0.0  12.0
## 159      6.3    85.8     0.5       0.7       3.7       0.0      0.0  11.9
## 160      7.2    90.5     0.5       0.0       0.0       0.0      0.0  11.7
## 161      7.7    85.4     0.5       0.0       0.0       0.0      0.0  11.5
## 162      7.3    91.1     0.4       0.0       0.0       0.0      0.0  11.5
## 163      7.3    75.2     0.5       0.0       0.0       0.0      0.0  10.4
## 164      6.0    75.2     0.4       0.0       0.0       0.0      0.0  10.0
## 165      5.4    71.7     0.4       0.0       0.0       0.0      0.0   9.7
## 166      4.3    56.8     0.5       0.7       3.9       0.0      0.1   9.1
## 167      4.5    50.5     0.3       2.4      13.6       0.1      0.0   8.6
## 168      4.5    48.8     0.3       2.5      14.3       0.1      0.0   8.4
## 169      4.6    58.2     0.4       0.0       0.0       0.0      0.0   7.9
## 170      5.1    62.6     0.3       0.0       0.0       0.0      0.0   7.9
## 171      4.0    58.1     0.3       0.0       0.0       0.0      0.0   7.8
## 172      4.2    57.2     0.3       0.0       0.0       0.0      0.0   7.8
## 173      5.3    56.2     0.3       0.0       0.0       0.0      0.0   7.5
## 174      4.1    52.2     0.4       0.0       0.0       0.0      0.0   7.5
## 175      3.7    46.3     0.2       1.7       9.0       0.1      0.0   7.3
## 176      4.4    52.5     0.3       0.0       0.0       0.0      0.1   6.8
## 177      4.2    47.9     0.3       0.0       0.0       0.0      0.0   6.6
## 178      3.5    45.6     0.3       0.0       0.0       0.0      0.0   6.6
## 179      3.7    46.6     0.3       0.0       0.0       0.0      0.0   6.5
## 180      4.1    46.9     0.3       0.0       0.0       0.0      0.0   6.3
## 181      3.9    38.0     0.3       0.0       0.0       0.0      0.0   5.7
## 182      3.9    39.7     0.2       0.0       0.0       0.0      0.0   5.4
## 183      2.9    37.4     0.2       0.0       0.0       0.0      0.0   5.0
## 184      2.5    32.9     0.2       0.0       0.0       0.0      0.0   4.7
## 185      2.4    30.5     0.3       0.0       0.0       0.0      0.0   4.6
## 186      2.4    28.1     0.1       1.0       5.8       0.0      0.0   4.3
## 187      2.3    29.5     0.2       0.0       0.0       0.0      0.0   3.9
## 188      2.3    27.0     0.1       0.0       0.0       0.0      0.0   3.5
## 189      2.1    24.1     0.2       0.0       0.0       0.0      0.0   3.4
## 190      2.0    25.2     0.2       0.0       0.0       0.0      0.0   3.4
## 191      2.0    22.5     0.2       0.0       0.0       0.0      0.0   3.3
## 192      1.9    24.5     0.1       0.0       0.0       0.0      0.0   3.2
## 193      1.8    22.9     0.1       0.0       0.0       0.0      0.0   3.1
## 194      1.6    16.7     0.2       0.0       0.0       0.0      0.0   2.8
## 195      1.5    19.3     0.1       0.0       0.0       0.0      0.0   2.7
## 196      1.7    21.4     0.1       0.0       0.0       0.0      0.0   2.7
## 197      1.3    17.1     0.2       0.0       0.0       0.0      0.0   2.7
## 198      1.2    17.3     0.2       0.0       0.0       0.0      0.0   2.7
## 199      1.5    19.0     0.1       0.0       0.0       0.0      0.0   2.4
## 200      0.9     9.1     0.0       1.8       9.4       0.1      0.0   2.3
## 201      1.5    17.6     0.1       0.0       0.0       0.0      0.0   2.1
## 202      1.1    12.1     0.0       1.2       6.4       0.0      0.0   1.9
## 203      1.2    15.2     0.1       0.0       0.0       0.0      0.0   1.8
## 204      1.6    15.0     0.1       0.0       0.0       0.0      0.0   1.8
```

```
## 205      0.8      9.4     0.0       0.9       4.8       0.0      0.0   1.4
## 206      0.9      9.5     0.0       0.9       4.3       0.0      0.0   1.4
## 207      0.7      8.1     0.0       0.9       4.7       0.0      0.0   1.3
## 208      0.8     10.6     0.0       0.0       0.0       0.0      0.0   1.1
## 209      0.9     10.1     0.0       0.0       0.0       0.0      0.0   1.0
## 210      0.7      9.8     0.0       0.0       0.0       0.0      0.0   1.0
## 211      0.9      9.7     0.0       0.0       0.0       0.0      0.0   1.0
## 212      0.9      9.6     0.0       0.0       0.0       0.0      0.0   1.0
## 213      0.8      9.5     0.0       0.0       0.0       0.0      0.0   1.0
## 214      0.9      9.5     0.0       0.0       0.0       0.0      0.0   1.0
## 215      0.7      9.4     0.0       0.0       0.0       0.0      0.0   0.9
## 216      0.9      9.4     0.0       0.0       0.0       0.0      0.0   0.9
## 217      0.9      9.3     0.0       0.0       0.0       0.0      0.0   0.9
## 218      0.9      9.2     0.0       0.0       0.0       0.0      0.0   0.9
## 219      0.8      9.2     0.0       0.0       0.0       0.0      0.0   0.9
## 220      0.8      9.2     0.0       0.0       0.0       0.0      0.0   0.9
## 221      0.7      9.1     0.0       0.0       0.0       0.0      0.0   0.9
## 222      0.9      9.0     0.0       0.0       0.0       0.0      0.0   0.9
## 223      0.8      9.0     0.0       0.0       0.0       0.0      0.0   0.9
## 224      0.8      9.0     0.0       0.0       0.0       0.0      0.0   0.9
## 225      0.7      8.8     0.0       0.0       0.0       0.0      0.0   0.9
## 226      0.7      8.7     0.0       0.0       0.0       0.0      0.0   0.9
## 227      0.7      8.6     0.0       0.0       0.0       0.0      0.0   0.9
## 228      0.7      8.5     0.0       0.0       0.0       0.0      0.0   0.9
## 229      0.8      8.4     0.0       0.0       0.0       0.0      0.0   0.8
## 230      0.7      8.4     0.0       0.0       0.0       0.0      0.0   0.8
## 231      0.7      8.3     0.0       0.0       0.0       0.0      0.0   0.8
## 232      0.7      8.3     0.0       0.0       0.0       0.0      0.0   0.8
## 233      0.7      8.3     0.0       0.0       0.0       0.0      0.0   0.8
## 234      0.7      8.0     0.0       0.0       0.0       0.0      0.0   0.8
## 235      0.8      8.0     0.0       0.0       0.0       0.0      0.0   0.8
## 236      0.7      8.0     0.0       0.0       0.0       0.0      0.0   0.8
## 237      0.7      8.0     0.0       0.0       0.0       0.0      0.0   0.8
## 238      0.7      8.0     0.0       0.0       0.0       0.0      0.0   0.8
## 239      0.7      7.7     0.0       0.0       0.0       0.0      0.0   0.8
## 240      0.7      7.6     0.0       0.0       0.0       0.0      0.0   0.8
## 241      0.7      7.6     0.0       0.0       0.0       0.0      0.0   0.8
## 242      0.8      7.1     0.0       0.0       0.0       0.0      0.0   0.7
```

```
cor(df, method= c("pearson"))
```

```
##             rec_att   rec_yds   rec_tds  rush_att  rush_yds  rush_tds   fumbles
## rec_att   1.0000000 0.9889836 0.9620513 0.2242480 0.2810831 0.2312038 0.6423627
## rec_yds   0.9889836 1.0000000 0.9720400 0.2062038 0.2614786 0.2115013 0.6487247
## rec_tds   0.9620513 0.9720400 1.0000000 0.2004448 0.2540571 0.2151580 0.6021914
## rush_att  0.2242480 0.2062038 0.2004448 1.0000000 0.9779751 0.9308512 0.1446322
## rush_yds  0.2810831 0.2614786 0.2540571 0.9779751 1.0000000 0.9298581 0.1761579
## rush_tds  0.2312038 0.2115013 0.2151580 0.9308512 0.9298581 1.0000000 0.1809564
## fumbles   0.6423627 0.6487247 0.6021914 0.1446322 0.1761579 0.1809564 1.0000000
## fpts      0.9863078 0.9957911 0.9842850 0.2610623 0.3162080 0.2677821 0.6288445
##                fpts
## rec_att   0.9863078
## rec_yds   0.9957911
## rec_tds   0.9842850
```

```
## rush_att 0.2610623
## rush_yds 0.3162080
## rush_tds 0.2677821
## fumbles  0.6288445
## fpts     1.0000000
```

1. Generate a data set with 30 rows that has a similar correlation structure. Repeat the procedure 1,000 times and return the mean correlation matrix. (10 points)

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

df.sim <- mvrnorm(20, mu = means, Sigma = vcov)
df.sim = as.data.frame(df.sim)

(rho.sim<- cor(df.sim))
```

```
##              rec_att      rec_yds     rec_tds     rush_att     rush_yds
## rec_att   1.00000000  0.992489481  0.97896713 -0.043684584  0.016840089
## rec_yds   0.99248948  1.000000000  0.97874980 -0.052980964  0.007534164
## rec_tds   0.97896713  0.978749804  1.00000000 -0.062834766 -0.006750880
## rush_att -0.04368458 -0.052980964 -0.06283477  1.000000000  0.984445400
## rush_yds  0.01684009  0.007534164 -0.00675088  0.984445400  1.000000000
## rush_tds -0.11465390 -0.126411818 -0.11797805  0.927982073  0.921362860
## fumbles   0.52346889  0.544682123  0.49390683 -0.141140735 -0.067348051
## fpts      0.99190752  0.996348694  0.98753692  0.002749317  0.062123836
##             rush_tds     fumbles         fpts
## rec_att  -0.11465390  0.52346889  0.991907519
## rec_yds  -0.12641182  0.54468212  0.996348694
## rec_tds  -0.11797805  0.49390683  0.987536916
## rush_att  0.92798207 -0.14114074  0.002749317
## rush_yds  0.92136286 -0.06734805  0.062123836
## rush_tds  1.00000000 -0.13901227 -0.067538742
## fumbles  -0.13901227  1.00000000  0.516869426
## fpts     -0.06753874  0.51686943  1.000000000
```

```r
rho
```

```
##            rec_att   rec_yds   rec_tds  rush_att  rush_yds  rush_tds   fumbles
## rec_att  1.0000000 0.9889836 0.9620513 0.2242480 0.2810831 0.2312038 0.6423627
## rec_yds  0.9889836 1.0000000 0.9720400 0.2062038 0.2614786 0.2115013 0.6487247
## rec_tds  0.9620513 0.9720400 1.0000000 0.2004448 0.2540571 0.2151580 0.6021914
## rush_att 0.2242480 0.2062038 0.2004448 1.0000000 0.9779751 0.9308512 0.1446322
## rush_yds 0.2810831 0.2614786 0.2540571 0.9779751 1.0000000 0.9298581 0.1761579
## rush_tds 0.2312038 0.2115013 0.2151580 0.9308512 0.9298581 1.0000000 0.1809564
## fumbles  0.6423627 0.6487247 0.6021914 0.1446322 0.1761579 0.1809564 1.0000000
## fpts     0.9863078 0.9957911 0.9842850 0.2610623 0.3162080 0.2677821 0.6288445
##               fpts
## rec_att  0.9863078
## rec_yds  0.9957911
## rec_tds  0.9842850
```

```
## rush_att 0.2610623
## rush_yds 0.3162080
## rush_tds 0.2677821
## fumbles  0.6288445
## fpts     1.0000000
```

```
keep.1=0
loops=10000

for (i in 1:loops) {
      df.sim = mvrnorm(20, mu = means, Sigma = vcov)
      keep.1=keep.1+cor(df.sim)/loops  }
keep.1-rho
```

```
##                rec_att       rec_yds       rec_tds      rush_att      rush_yds
## rec_att  -9.381385e-14 -6.187748e-04 -2.146039e-03 -8.784522e-03 -9.652938e-03
## rec_yds  -6.187748e-04 -9.381385e-14 -1.672419e-03 -8.013501e-03 -8.880588e-03
## rec_tds  -2.146039e-03 -1.672419e-03 -9.381385e-14 -8.312121e-03 -9.249453e-03
## rush_att -8.784522e-03 -8.013501e-03 -8.312121e-03 -9.381385e-14 -1.193939e-03
## rush_yds -9.652938e-03 -8.880588e-03 -9.249453e-03 -1.193939e-03 -9.381385e-14
## rush_tds -7.278976e-03 -6.544112e-03 -7.086723e-03 -3.330494e-03 -3.082884e-03
## fumbles  -8.857264e-03 -9.514761e-03 -9.926437e-03 -6.127319e-03 -6.425819e-03
## fpts     -8.017956e-04 -2.498811e-04 -9.746911e-04 -9.264005e-03 -1.003897e-02
##               rush_tds       fumbles          fpts
## rec_att  -7.278976e-03 -8.857264e-03 -8.017956e-04
## rec_yds  -6.544112e-03 -9.514761e-03 -2.498811e-04
## rec_tds  -7.086723e-03 -9.926437e-03 -9.746911e-04
## rush_att -3.330494e-03 -6.127319e-03 -9.264005e-03
## rush_yds -3.082884e-03 -6.425819e-03 -1.003897e-02
## rush_tds -9.381385e-14 -6.421741e-03 -7.817458e-03
## fumbles  -6.421741e-03 -9.381385e-14 -9.744527e-03
## fpts     -7.817458e-03 -9.744527e-03 -9.381385e-14
```

Question 3

21 points

Here's some code:

```
nDist <- function(n = 100) {
    df <- 10
    prob <- 1/3
    shape <- 1
    size <- 16
    list(
        beta = rbeta(n, shape1 = 5, shape2 = 45),
        binomial = rbinom(n, size, prob),
        chisquared = rchisq(n, df),
        exponential = rexp(n),
        f = rf(n, df1 = 11, df2 = 17),
        gamma = rgamma(n, shape),
        geometric = rgeom(n, prob),
        hypergeometric = rhyper(n, m = 50, n = 100, k = 8),
```

```
        lognormal = rlnorm(n),
        negbinomial = rnbinom(n, size, prob),
        normal = rnorm(n),
        poisson = rpois(n, lambda = 25),
        t = rt(n, df),
        uniform = runif(n),
        weibull = rweibull(n, shape)
    )
}
```

1. What does this do? (3 points)

```
round(sapply(nDist(500), mean), 2)
```

```
##          beta       binomial     chisquared    exponential              f
##          0.10           5.44           9.92           1.06           1.18
##         gamma      geometric hypergeometric      lognormal    negbinomial
##          0.92           2.01           2.67           1.81          32.39
##        normal        poisson              t        uniform        weibull
##          0.01          24.79          -0.06           0.50           1.03
```

```
  Sapply is  simplifying the output by converting the list of functions into a vector. The mean of
```

2. What about this? (3 points)

```
sort(apply(replicate(20, round(sapply(nDist(10000), mean), 2)), 1, sd))
```

```
##          beta        uniform              f    exponential          gamma
##   0.000000000    0.003663475    0.006958524    0.009665457    0.009679060
##        normal        weibull hypergeometric              t       binomial
##   0.009986833    0.012096106    0.012311740    0.012396944    0.013869694
##     lognormal      geometric     chisquared        poisson    negbinomial
##   0.020749128    0.021588252    0.053555186    0.065652594    0.116821952
```

‘

```
  Sapply is  simplifying the output by converting the list of functions into a vector. The mean of
```

In the output above, a small value would indicate that `N=10,000` would provide a sufficent sample size as to estimate the mean of the distribution. Let’s say that a value *less than 0.02* is “close enough”.

3. For each distribution, estimate the sample size required to simulate the distribution’s mean. (15 points)

```
nDist <- function(n = 100) {
    df <- 10
    prob <- 1/3
    shape <- 1
    size <- 16
    list(
        beta = rbeta(n, shape1 = 5, shape2 = 45),
        binomial = rbinom(n, size, prob),
        chisquared = rchisq(n, df),
        exponential = rexp(n),
```

```r
        f = rf(n, df1 = 11, df2 = 17),
        gamma = rgamma(n, shape),
        geometric = rgeom(n, prob),
        hypergeometric = rhyper(n, m = 50, n = 100, k = 8),
        lognormal = rlnorm(n),
        negbinomial = rnbinom(n, size, prob),
        normal = rnorm(n),
        poisson = rpois(n, lambda = 25),
        t = rt(n, df),
        uniform = runif(n),
        weibull = rweibull(n, shape)
    )
}
round(sapply(nDist(500), mean), 2)
```

```
##           beta       binomial     chisquared    exponential              f
##           0.10           5.27          10.07           1.00           1.13
##          gamma      geometric hypergeometric      lognormal    negbinomial
##           0.94           1.94           2.65           1.76          32.14
##         normal        poisson              t        uniform        weibull
##          -0.01          25.25           0.04           0.50           0.99
```

Don't worry about being exact. It should already be clear that N < 10,000 for many of the distributions. You don't have to show your work. Put your answer to the right of the vertical bars (|) below.

distribution	N
beta	?
binomial	?
chisquared	?
exponential	?
f	?
gamma	?
geometric	?
hypergeometric	?
lognormal	?
negbinomial	?
normal	?
poisson	?
t	?
uniform	?
weibull	?