01 - Intro to R

Stephen Blatti

October 7, 2017

1. Calculate the following numerical results to three decimal places:

a) (7 - 8) + 5^3 - 5 / 6 + sqrt(62)
b) ln 3 + sqrt(2)sin(pi) - e^3

```
#a
round( (7 - 8) + 5^3 - 5 / 6 + sqrt(62),3 )
```

```
## [1] 131.041
```

```
#b
round( log(3) + sqrt(2)*sin(pi) - exp(3),3 )
```

```
## [1] -18.987
```

3. Create a vector named Treatment with the entries "Treatment One" appearing 20 times, "Treatment Two" appearing 18 times, and "Treatment Three" appearing 22 times.

```
Treatment <- rep(c("Treatment One", "Treatment Two", "Treatment Three"), c(20, 18, 22))
xtabs(~Treatment)
```

```
## Treatment
##   Treatment One Treatment Three   Treatment Two
##              20              22              18
```

5. Vectors, sequences, and logical operators

```
#b Create the vectors u = (1,2,5,4) and v = (2,2,1,1) using the c() and scan() fctns.
u = c(1,2,5,4)
v = c(2,2,1,1)
#c find which component of u is equal to 5
which(u == 5)
```

```
## [1] 3
```

```
#i calculate the scalar product (dot product) of q = (3,0,1,6) and r = (1,0,2,4)
q = c(3,0,1,6)
r = c(1,0,2,4)
q %*% r
```

```
##      [,1]
## [1,]   29
```

```
#j Define the matrix X whose rows are the vectors u and v
X = matrix(rbind(u,v),2,4)
#or X = rbind(u,v)

#k Define the matrix Y whose columns are the vectors u and v
Y = cbind(u,v)

#l Find the matrix product  X by Y and name is W
(W = X %*% Y)
```

```
##       u  v
```

```
## [1,] 46 15
## [2,] 15 10
```

```
#m Compute the inverse matrix of W and the transpose of that inverse
solve(W)
```

```
##           [,1]        [,2]
## u  0.04255319 -0.06382979
## v -0.06382979  0.19574468
```

```
t( solve(W))
```

```
##                u           v
## [1,]  0.04255319 -0.06382979
## [2,] -0.06382979  0.19574468
```

7. The data frame wheatUSA2004 from the PASWR package has the USA wheat harvested crop surfaces in 2004 by states. It has 2 variables, STATE for state and ACRES for 1000s of acres.

a) Attach the data frame wheatUSA2004 and use the fctn row.names() to define the states as the row names.

```
library(PASWR2)
```

```
## Warning: package 'PASWR2' was built under R version 3.4.2
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
STATES = WHEATUSA2004$states
row.names(WHEATUSA2004) = STATES
head(WHEATUSA2004)
```

```
##    states acres
## AR     AR   620
## CA     CA   320
## CO     CO  1700
## DE     DE    47
## GA     GA   190
## ID     ID   700
```

b) Define a new var called ha for the surface area given in hectors where 1 acre = 0.40468564224 hectares.

```
WHEATUSA2004$ha <- WHEATUSA2004$acres * 0.40468564224
head(WHEATUSA2004)
```

```
##    states acres        ha
## AR     AR   620 250.90510
## CA     CA   320 129.49941
## CO     CO  1700 687.96559
## DE     DE    47  19.02023
## GA     GA   190  76.89027
## ID     ID   700 283.27995
```

c) Sort the file according to the harvested surface area in acres.

```
sort_wheatUSA2004 <- WHEATUSA2004[order(WHEATUSA2004$acres), ]
head(sort_wheatUSA2004)
```

```
##    states acres       ha
## DE     DE    47 19.02023
```

```
## NY     NY   100 40.46856
## MS     MS   135 54.63256
## PA     PA   135 54.63256
## MD     MD   145 58.67942
## SC     SC   180 72.84342
```

d) Which states fall in the top 10% of states for harvested surface area?

```
top10 <- quantile(WHEATUSA2004$acres, prob = 0.9)
result <- WHEATUSA2004[WHEATUSA2004$acres > top10, ]
row.names(result)
```

```
## [1] "KS" "OK" "TX"
```

e) Save the contents of wheatUSA2004 in a new file called wheatUSA.txt. Then, remove wheatUSA2004 from your workspace and check that the contents of wheatUSA2004 can be recovered from wheatUSA.txt.

```
dump("WHEATUSA2004", "WHEATUSA.txt")
rm(WHEATUSA2004)
source("WHEATUSA.txt")
head(WHEATUSA2004)
```

```
##    states acres        ha
## AR     AR   620 250.90510
## CA     CA   320 129.49941
## CO     CO  1700 687.96559
## DE     DE    47  19.02023
## GA     GA   190  76.89027
## ID     ID   700 283.27995
```

f) Use the command write.table() to store the contents of wheatUSA2004 in a file with the namw wheatUSA.dat

```
write.table(WHEATUSA2004, "WHEATUSA.dat")
```

g) Find the total harvested surface area in acres for the bottom 10% of states.

```
bottom10 <- quantile(WHEATUSA2004$acres, prob = 0.1)
result <- WHEATUSA2004[WHEATUSA2004$acres < bottom10, ]
result
```

```
##    states acres       ha
## DE     DE    47 19.02023
## NY     NY   100 40.46856
```

```
total_HA <- sum(result[, "acres"])
total_HA
```

```
## [1] 147
```

9. Use the data frame EPIURALf to answer the following questions:

a) How many patients have been treated with the Hamstring Stretch?

```
head(EPIDURALF)
```

```
##   doctor  kg  cm       ease           treatment oc complications
## 1      B 116 172  Difficult Traditional Sitting  0          None
## 2      C  86 176       Easy   Hamstring Stretch  0          None
## 3      B  72 157  Difficult Traditional Sitting  0          None
## 4      B  63 169       Easy   Hamstring Stretch  2          None
```

```
## 5      B 114 163 Impossible Traditional Sitting  0          None
## 6      B 121 163  Difficult   Hamstring Stretch  3          None
```

```
xtabs(~treatment, data = EPIDURALF)
```

```
## treatment
##   Hamstring Stretch Traditional Sitting
##                 171                 171
```

```
xtabs(~treatment, data = EPIDURALF)[1]
```

```
## Hamstring Stretch
##               171
```

b) What proportion of the patients treated with Hamsrting Stretch were classified as each of easy, difficult, and impossible?

```
Treatment1 <- xtabs(~treatment + ease, data = EPIDURALF)
Treatment1
```

```
##                      ease
## treatment             Difficult Easy Impossible
##   Hamstring Stretch          63  100          8
##   Traditional Sitting        51  107         13
```

```
prop.table(Treatment1[1, ]) * 100
```

```
##  Difficult       Easy Impossible
##  36.842105  58.479532   4.678363
```

c) What proportion of the patients classified as easy to palpitate were assigned to the Traditional Sitting position?

```
Treatment1
```

```
##                      ease
## treatment             Difficult Easy Impossible
##   Hamstring Stretch          63  100          8
##   Traditional Sitting        51  107         13
```

```
prop.table(Treatment1[, "Easy"])[2] * 100
```

```
## Traditional Sitting
##            51.69082
```

d) What is the mean weight for each cell in a contingency table created with the var's Ease and Treatment?

```
head(EPIDURALF)
```

```
##   doctor  kg  cm       ease           treatment oc complications
## 1      B 116 172  Difficult Traditional Sitting  0          None
## 2      C  86 176       Easy   Hamstring Stretch  0          None
## 3      B  72 157  Difficult Traditional Sitting  0          None
## 4      B  63 169       Easy   Hamstring Stretch  2          None
## 5      B 114 163 Impossible Traditional Sitting  0          None
## 6      B 121 163  Difficult   Hamstring Stretch  3          None
```

```
tapply(EPIDURALF$kg, list(EPIDURALF$ease, EPIDURALF$treatment), mean)
```

```
##            Hamstring Stretch Traditional Sitting
## Difficult           92.66667            94.27451
## Easy                78.67000            79.40187
```

```
## Impossible         127.87500           113.61538
```

e) What proportion of the patients have a body mass index (BMI = kg/(cm/100)^2) less than 25 and are classified as Easy to palpitate?

```
EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2
EPIDURALF[1:5, 3:8]
```

```
##    cm       ease           treatment oc complications      BMI
## 1 172  Difficult Traditional Sitting  0          None 39.21038
## 2 176       Easy   Hamstring Stretch  0          None 27.76343
## 3 157  Difficult Traditional Sitting  0          None 29.21011
## 4 169       Easy   Hamstring Stretch  2          None 22.05805
## 5 163 Impossible Traditional Sitting  0          None 42.90715
```

```
mean(EPIDURALF$ease =="Easy" & EPIDURALF$BMI < 25) * 100
```

```
## [1] 9.064327
```

11. Use a for loop to convert a sequence of temperatures (18 to 28 by 2) from degrees centigrade to degrees fahrenheit.

```
for (celsius in seq(from = 18, to = 28, by = 2)) {
  print(c(celsius, 9/5 * celsius + 32))
  }
```

```
## [1] 18.0 64.4
## [1] 20 68
## [1] 22.0 71.6
## [1] 24.0 75.2
## [1] 26.0 78.8
## [1] 28.0 82.4
```