Week 2 homework assignment

Student Name

Due Date

DSI-EDA

Professor Michael Shepherd

Homework Assignment

(Due Date: Wednesday before class, 11: 59pm CST)

Let's practice what we've learned about data.table, tidyverse, and summarizing data so far. We're going to be working with one of R's built-in datasets. These data are included with the base installation of R for learning purposes. Loading the library `datasets` makes 30+ data objects available to your R session. Today's dataset is named `UCBAdmissions` and provides data on student admissions to University of California, Berkeley.

To submit this assignment, knit it to github flavored markdown and submit all your work on GitHub.

Question 1: What kind of dataset is UCBAdmissions? Include the command you used to find out. Convert `UCBAdmissions` to a data.table object and a data.frame object.

```
## The type of dataset is...?
print(class(UCBAdmissions))
```

```
## [1] "table"
```

```
#The type of the dataset is "double"
data <- UCBAdmissions
## Convert to data.table
admissions_dt <- data.table::as.data.table(data)
admissions_dt
```

```
##        Admit Gender Dept   N
##  1: Admitted   Male    A 512
##  2: Rejected   Male    A 313
##  3: Admitted Female    A  89
##  4: Rejected Female    A  19
##  5: Admitted   Male    B 353
##  6: Rejected   Male    B 207
##  7: Admitted Female    B  17
##  8: Rejected Female    B   8
##  9: Admitted   Male    C 120
## 10: Rejected   Male    C 205
## 11: Admitted Female    C 202
## 12: Rejected Female    C 391
## 13: Admitted   Male    D 138
## 14: Rejected   Male    D 279
## 15: Admitted Female    D 131
## 16: Rejected Female    D 244
## 17: Admitted   Male    E  53
## 18: Rejected   Male    E 138
## 19: Admitted Female    E  94
## 20: Rejected Female    E 299
## 21: Admitted   Male    F  22
## 22: Rejected   Male    F 351
## 23: Admitted Female    F  24
## 24: Rejected Female    F 317
##        Admit Gender Dept   N
```

```
## Convert to data.frame
admissions_df <- as.data.frame(admissions_dt)
admissions_df
```

```
##       Admit Gender Dept   N
## 1  Admitted   Male    A 512
## 2  Rejected   Male    A 313
## 3  Admitted Female    A  89
## 4  Rejected Female    A  19
## 5  Admitted   Male    B 353
## 6  Rejected   Male    B 207
## 7  Admitted Female    B  17
## 8  Rejected Female    B   8
## 9  Admitted   Male    C 120
## 10 Rejected   Male    C 205
## 11 Admitted Female    C 202
## 12 Rejected Female    C 391
## 13 Admitted   Male    D 138
## 14 Rejected   Male    D 279
## 15 Admitted Female    D 131
## 16 Rejected Female    D 244
## 17 Admitted   Male    E  53
## 18 Rejected   Male    E 138
## 19 Admitted Female    E  94
## 20 Rejected Female    E 299
## 21 Admitted   Male    F  22
## 22 Rejected   Male    F 351
## 23 Admitted Female    F  24
## 24 Rejected Female    F 317
```

Question 2: Using data.table syntax, sum the number of applicants by department and save the output as a new data object. Then, using tidyverse syntax, again sum the number of applicants by department and save the output as a new data object. Make sure you use the right type of object (data.table or data.frame) with the right syntax!

```
## Sum using data.table syntax
sum_dt <-admissions_dt[,.(N.total.sum =sum(N)), by=Dept][order(Dept)]
sum_dt
```

```
##    Dept N.total.sum
## 1:    A         933
## 2:    B         585
## 3:    C         918
## 4:    D         792
## 5:    E         584
## 6:    F         714
```

```
## Sum using tidyverse syntax
sum_df <- admissions_df %>%
  group_by(Dept) %>%
  summarise(N.total.sum = sum(N))

sum_df
```

```
## # A tibble: 6 x 2
##   Dept  N.total.sum
##   <chr>       <dbl>
## 1 A             933
## 2 B             585
## 3 C             918
## 4 D             792
## 5 E             584
## 6 F             714
```

Question 3: You can use the help operator, `?`, to get help with any function in R. For example, if you wanted to get help with the `names()` function, you would use `?names()`. You can also get help with a whole library. For example, you could use `?tidyverse` to get help with the tidyverse library. Using the help function, describe as best you can the differences between the data.table and tidyverse methods above. Do you prefer one over the other? Why or why not?

```
?data.table
```

```
## starting httpd help server ... done
```

```
?tidyverse
```

Difference between data.table and tidyverse operations assume tb is the tibble and dt is data table x and y are column headings

data.table	tidyverse
1. Creates a tibble or a data frame	1. Creates a data table
2. read_csv	2. fread
3. select(tb,x,y)	3. dt[,.(x,y)]
4. slice(tb,1:3)	4. dt[1:3,,]
5. filter(tb, state_condition)	5. dt[state_condition,]
6. arrange(tb, x)	6. dt[order(x),]
7. tb <- mutate(tb, var = formula)	7. dt[, var:= formula]

I prefer tidyverse as the functions are selfexplainatory and easier to undertand and remember.

Question 4: Using either method above, find the average department admittance rate for observations with `Female` in the gender column.

```
female_app <- filter(admissions_dt, Gender ==  "Female") %>% group_by(Dept) %>% summarize(applicant=sum
print(female_app)
```

```
## # A tibble: 6 x 2
##   Dept  applicant
##   <chr>     <dbl>
## 1 A           108
## 2 B            25
## 3 C           593
## 4 D           375
## 5 E           393
## 6 F           341
```

```
female_adm <- filter(admissions_dt, Gender ==  "Female", Admit == "Admitted") %>% group_by(Dept) %>% su
print(female_adm)
```

```
## # A tibble: 6 x 2
##   Dept  applicant
##   <chr>     <dbl>
## 1 A            89
## 2 B            17
## 3 C           202
## 4 D           131
## 5 E            94
## 6 F            24
```

```
female_rate <- mutate(female_adm, ad_rate = female_adm$applicant/female_app$applicant)
print(female_rate)
```

```
## # A tibble: 6 x 3
##   Dept  applicant ad_rate
##   <chr>     <dbl>   <dbl>
## 1 A            89  0.824
## 2 B            17  0.68
## 3 C           202  0.341
## 4 D           131  0.349
## 5 E            94  0.239
## 6 F            24  0.0704
```