Project 3 Solutions

(Abhimanyu Agarwal)

Collaborators: N/A

TA help:

1) Melissa: Helped on understanding the which() function.

Online resources used: N/A

Question 1

```
#Loads into dataframe called "splash_mountain" using read.csv()
splash_mountain <-read.csv("/class/datamine/data/disney/splash_mountain.csv")

#Columns/features in the dataset
str(splash_mountain)
```

```
'data.frame':   223936 obs. of  4 variables:
 $ date    : chr  "01/01/2015" "01/01/2015" "01/01/2015" "01/01/2015" ...
 $ datetime: chr  "2015-01-01 07:51:12" "2015-01-01 08:02:13" "2015-01-01 08:09:12" "2015-01-01 08:16:1
 $ SACTMIN : int  NA NA NA NA NA NA NA NA NA 4 ...
 $ SPOSTMIN: int  5 5 5 5 5 5 5 5 5 NA ...
```

```
#Either works, just out of curiosity i used both to see the difference in the information being conveye

#Returns the two values, the first value = rows and second value = columns:
dim(splash_mountain)
```

```
[1] 223936      4
```

Question 2

```
#Code to find SPOTSMIN
head(splash_mountain$SPOSTMIN)
```

```
[1] 5 5 5 5 5 5
```

```
#Code to find SACTMIN
head(splash_mountain$SACTMIN)
```

```
[1] NA NA NA NA NA NA
```

```
#Code to estimate the mean by removing NA values
mean(splash_mountain$SPOSTMIN, na.rm=TRUE)
```

```
[1] -71.70373
```

```
#Obtained mean is: -71.70373
```

```r
#Code to estimate the standard deviation by removing NA values
sqrt(var(splash_mountain$SPOSTMIN, na.rm=TRUE))
```

```
[1] 328.0586
```

```r
#Obtained Standard deviation is: 328.0586

#Result explanation here
# The data set column given to us talks about wait time.
# Waiting time expected cannot be negative quantity.
# Mean obtained is negative, which is impossible in reality.
# Hence we can say that the data set comprises of large negative values
# or significantly small positive values
```

Question 3

```r
#Here we are using TRUE's and FALSE's as indexes
#Code to estimate the mean by removing NA values
mean(splash_mountain$SPOSTMIN[splash_mountain$SPOSTMIN != -999], na.rm=TRUE)
```

```
[1] 43.3892
```

```r
#Newly obtained mean is: 43.3892

#Code to estimate the standard deviation by removing NA values
sqrt(var(splash_mountain$SPOSTMIN[splash_mountain$SPOSTMIN != -999], na.rm=TRUE))
```

```
[1] 31.74894
```

```r
#Newly obtained standard deviation is: 31.74894

#Result explanation here
#Yes, this solves the problem. Now that, negative value -999 has been eliminated
#the computed mean is a positive and believable value. Earlier, when the mean
#was negative, it didn't make sense to have a negative mean for the wait time.
```

Question 4

```r
#Change SPOSTMIN to posted_min_wait time
head(splash_mountain)
```

```
        date            datetime SACTMIN SPOSTMIN
1 01/01/2015 2015-01-01 07:51:12      NA        5
2 01/01/2015 2015-01-01 08:02:13      NA        5
3 01/01/2015 2015-01-01 08:09:12      NA        5
4 01/01/2015 2015-01-01 08:16:12      NA        5
5 01/01/2015 2015-01-01 08:23:12      NA        5
6 01/01/2015 2015-01-01 08:29:12      NA        5
```

```r
colnames(splash_mountain)[which(colnames(splash_mountain)=="SPOSTMIN")] <- "posted_min_wait_time"
#View the name change
head(splash_mountain)
```

```
        date            datetime SACTMIN posted_min_wait_time
1 01/01/2015 2015-01-01 07:51:12      NA                    5
2 01/01/2015 2015-01-01 08:02:13      NA                    5
```

```
3 01/01/2015 2015-01-01 08:09:12      NA                    5
4 01/01/2015 2015-01-01 08:16:12      NA                    5
5 01/01/2015 2015-01-01 08:23:12      NA                    5
6 01/01/2015 2015-01-01 08:29:12      NA                    5
```

```
#Change SACTMIN to actual_wait_time
head(splash_mountain)
```

```
        date            datetime SACTMIN posted_min_wait_time
1 01/01/2015 2015-01-01 07:51:12      NA                    5
2 01/01/2015 2015-01-01 08:02:13      NA                    5
3 01/01/2015 2015-01-01 08:09:12      NA                    5
4 01/01/2015 2015-01-01 08:16:12      NA                    5
5 01/01/2015 2015-01-01 08:23:12      NA                    5
6 01/01/2015 2015-01-01 08:29:12      NA                    5
```

```
colnames(splash_mountain)[which(colnames(splash_mountain)=="SACTMIN")] <- "actual_wait_time"
#View the name change
head(splash_mountain)
```

```
        date            datetime actual_wait_time posted_min_wait_time
1 01/01/2015 2015-01-01 07:51:12               NA                    5
2 01/01/2015 2015-01-01 08:02:13               NA                    5
3 01/01/2015 2015-01-01 08:09:12               NA                    5
4 01/01/2015 2015-01-01 08:16:12               NA                    5
5 01/01/2015 2015-01-01 08:23:12               NA                    5
6 01/01/2015 2015-01-01 08:29:12               NA                    5
```

Question 5

```
#Loading the samedata in a dataframe "df"
myDF <- read.csv("/class/datamine/data/disney/splash_mountain.csv")
head(myDF)
```

```
        date            datetime SACTMIN SPOSTMIN
1 01/01/2015 2015-01-01 07:51:12      NA        5
2 01/01/2015 2015-01-01 08:02:13      NA        5
3 01/01/2015 2015-01-01 08:09:12      NA        5
4 01/01/2015 2015-01-01 08:16:12      NA        5
5 01/01/2015 2015-01-01 08:23:12      NA        5
6 01/01/2015 2015-01-01 08:29:12      NA        5
```

```
#Now using the cut() functions
quarter <- cut(as.Date(myDF$date, "%m/%d/%Y"), "quarter")

#Estimating the possible combinations
nlevels(quarter)
```

```
[1] 20
```

```
#There are 20 quarters as we can see.

#Using factor() function we put labels like "q1", "q2"...etc
levels(quarter) <- factor(paste0("q", 1:nlevels(quarter)))

#Now we add a new column called "quarter"
myDF$quarter<-quarter
```

```r
#Now we check and observe that there is indeed a new column by the name "quarter"
head(myDF)
```

```
        date            datetime SACTMIN SPOSTMIN quarter
1 01/01/2015 2015-01-01 07:51:12      NA        5      q1
2 01/01/2015 2015-01-01 08:02:13      NA        5      q1
3 01/01/2015 2015-01-01 08:09:12      NA        5      q1
4 01/01/2015 2015-01-01 08:16:12      NA        5      q1
5 01/01/2015 2015-01-01 08:23:12      NA        5      q1
6 01/01/2015 2015-01-01 08:29:12      NA        5      q1
```

```r
tail(myDF)
```

```
             date            datetime SACTMIN SPOSTMIN quarter
223931 12/31/2019 2020-01-01 00:27:02      NA        5     q20
223932 12/31/2019 2020-01-01 00:34:02      NA        5     q20
223933 12/31/2019 2020-01-01 00:41:02      NA        5     q20
223934 12/31/2019 2020-01-01 00:48:02      NA        5     q20
223935 12/31/2019 2020-01-01 00:55:02      NA        5     q20
223936 12/31/2019 2020-01-01 01:01:02      NA        5     q20
```

Question 6

```r
#Including the statement
cat ("I acknowledge that the STAT 19000/29000/39000 1-credit Data Mine seminar will be recorded and pos
```

```
I acknowledge that the STAT 19000/29000/39000 1-credit Data Mine seminar will be recorded and posted on
```

Submitting deliverables: project03.RMD, project03.R and project03.pdf

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together - We are Purdue.