Johnson_Exam1

Jen

10/6/2017

```
library(nycflights13)
flights<- flights
library(ggplot2)
library(tidyverse)
```

```
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'tibble' was built under R version 3.4.1
```

```
## Warning: package 'tidyr' was built under R version 3.4.1
```

```
## Warning: package 'purrr' was built under R version 3.4.1
```

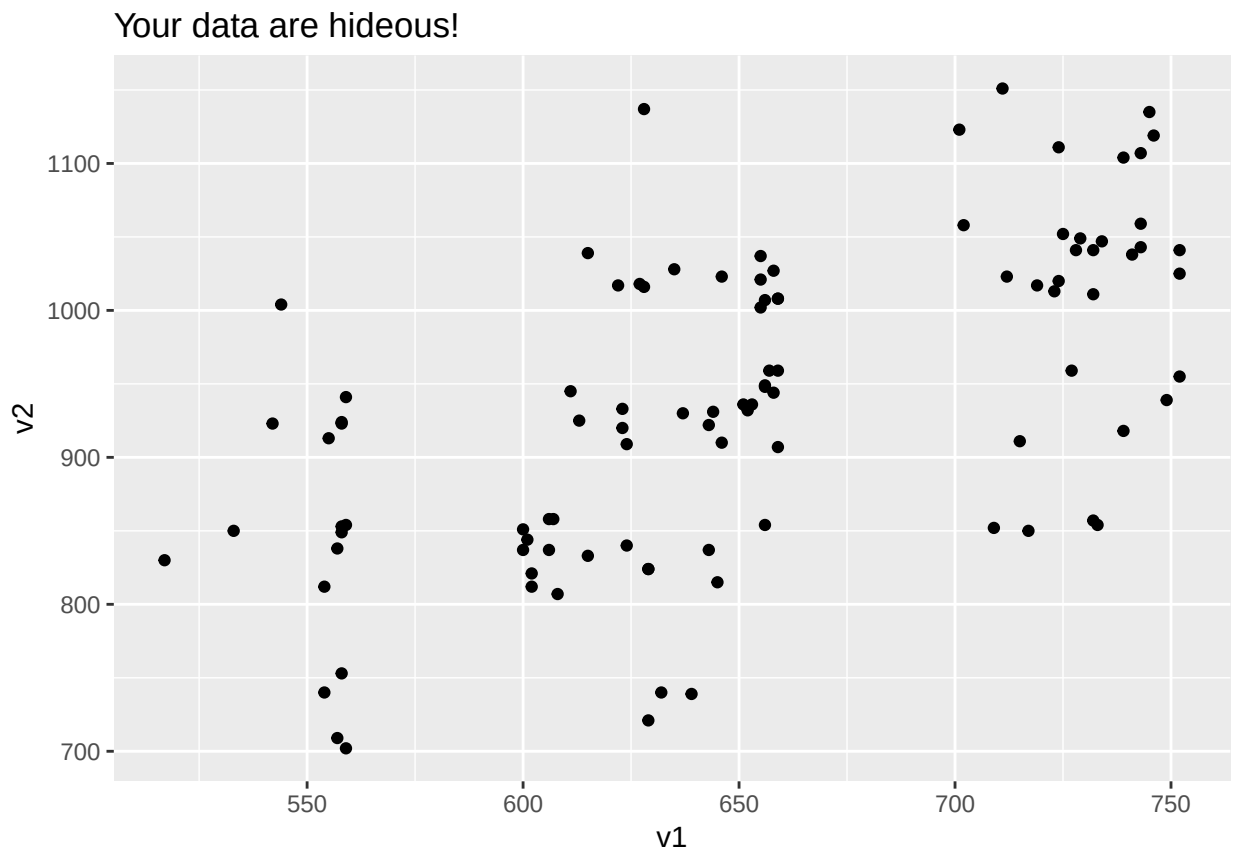
```
## Warning: package 'dplyr' was built under R version 3.4.1
```

```
## Conflicts with tidy packages ----------------------------------------------
```

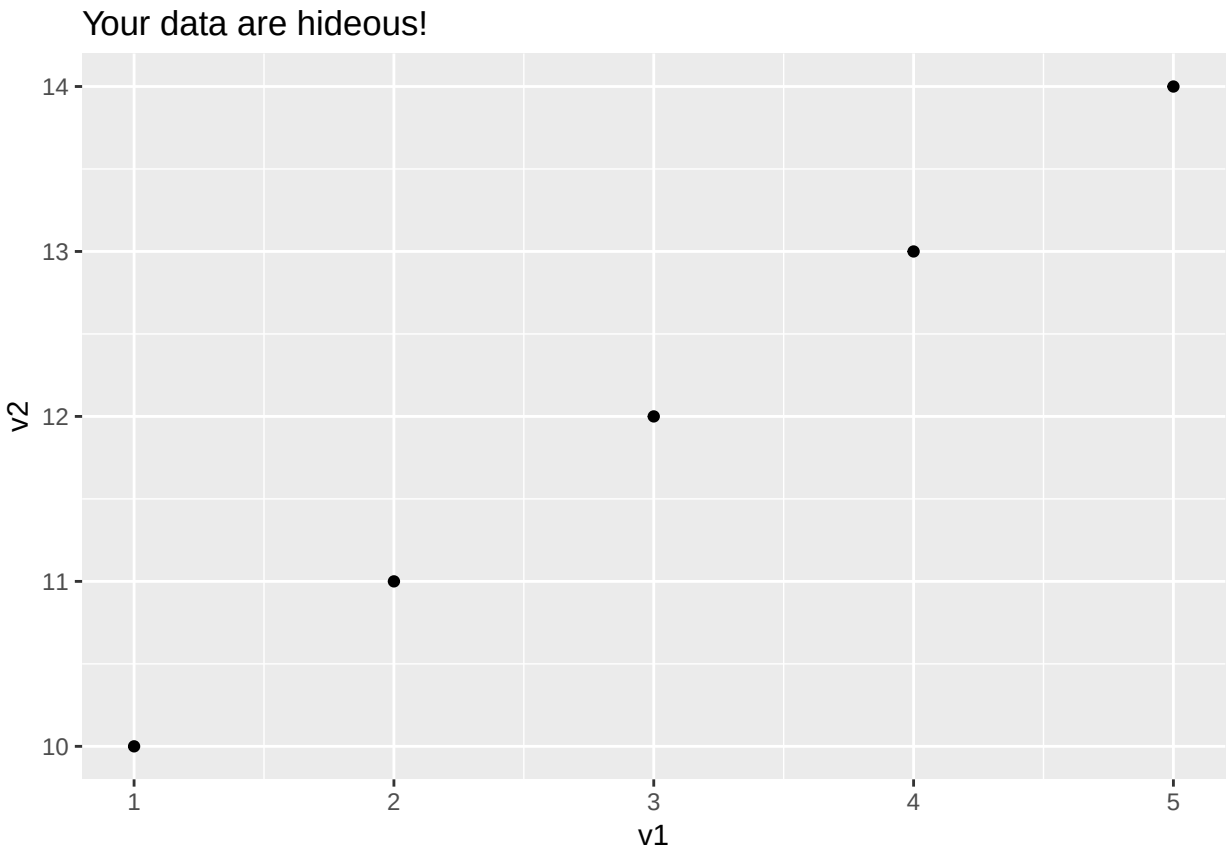
```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

Question 1

```
scatter_plot <- function(v1, v2){
  d <- cbind(v1, v2)
  data <- data.frame(d)
  ggplot(data, mapping = aes(x = v1, y = v2)) + geom_point() + ggtitle("Your data are hideous!")
}

# subset to test function
x <- flights$dep_time[1:100]
y <- flights$arr_time[1:100]
scatter_plot(x, y)
```



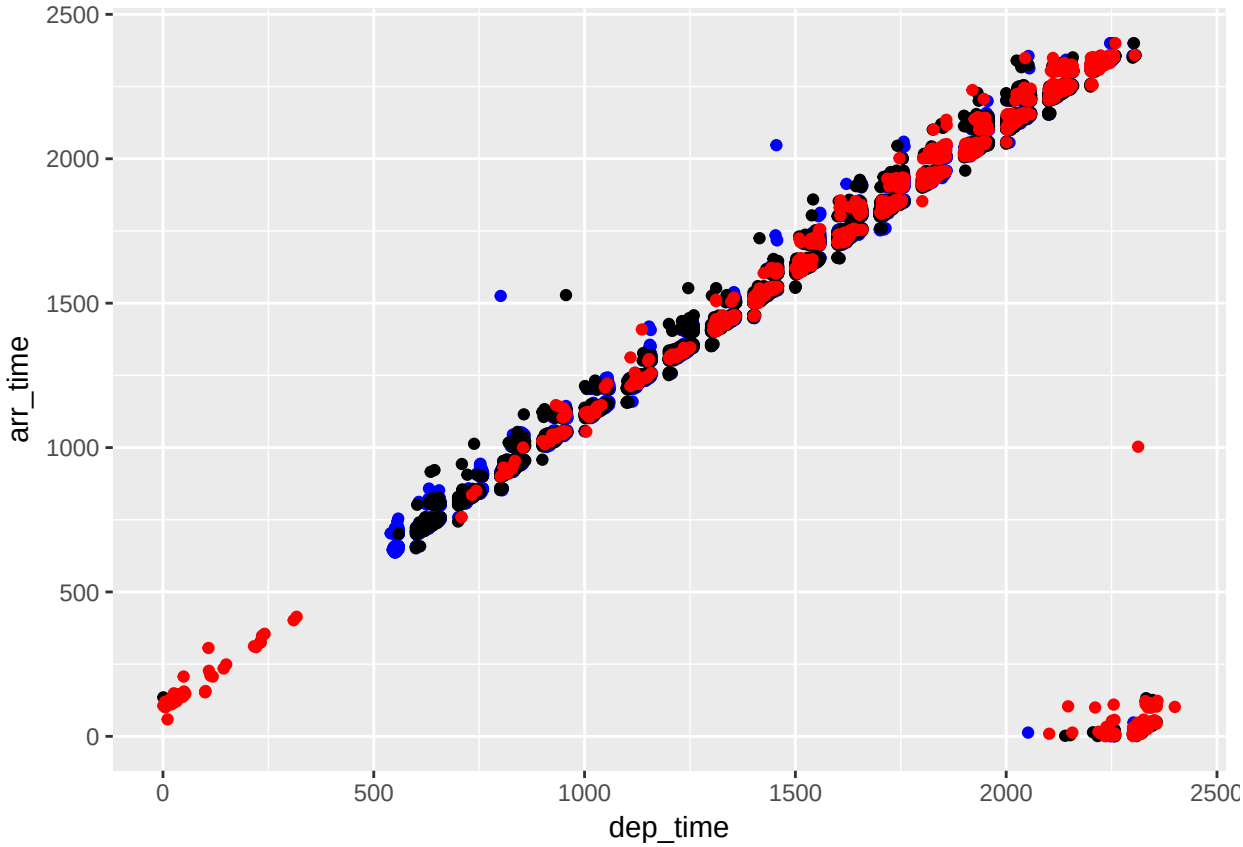

```
# another test
a <- c(1:5)
b <- c(10:14)
scatter_plot(a, b)
```



Question 2

```
base.data <- flights %>% filter(dest == "BOS") %>% select(dep_time, arr_time, dep_delay)

early.data <- base.data %>% filter(dep_delay<0)

late.data <- base.data %>% filter(between(dep_delay, 0, 60))

very.late.data <- base.data %>% filter(dep_delay>60)

ggplot(data = base.data, aes(x = dep_time, y = arr_time)) + geom_point(data = early.data, color = "blue
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
## Warning: Removed 3 rows containing missing values (geom_point).
## Warning: Removed 6 rows containing missing values (geom_point).
```



The distribution of point colors is even from the bottom left corner to the top right corner, although the density of red/very late points is higher in the top right corner, suggesting that flights later in the day are more likely to ve very late. However, flights that arrive in the early morning are also very late. The bottom left corner represents flights short early morning flights, while the bottom right corner represents red eye flights that left the night before and arrive the next morning.

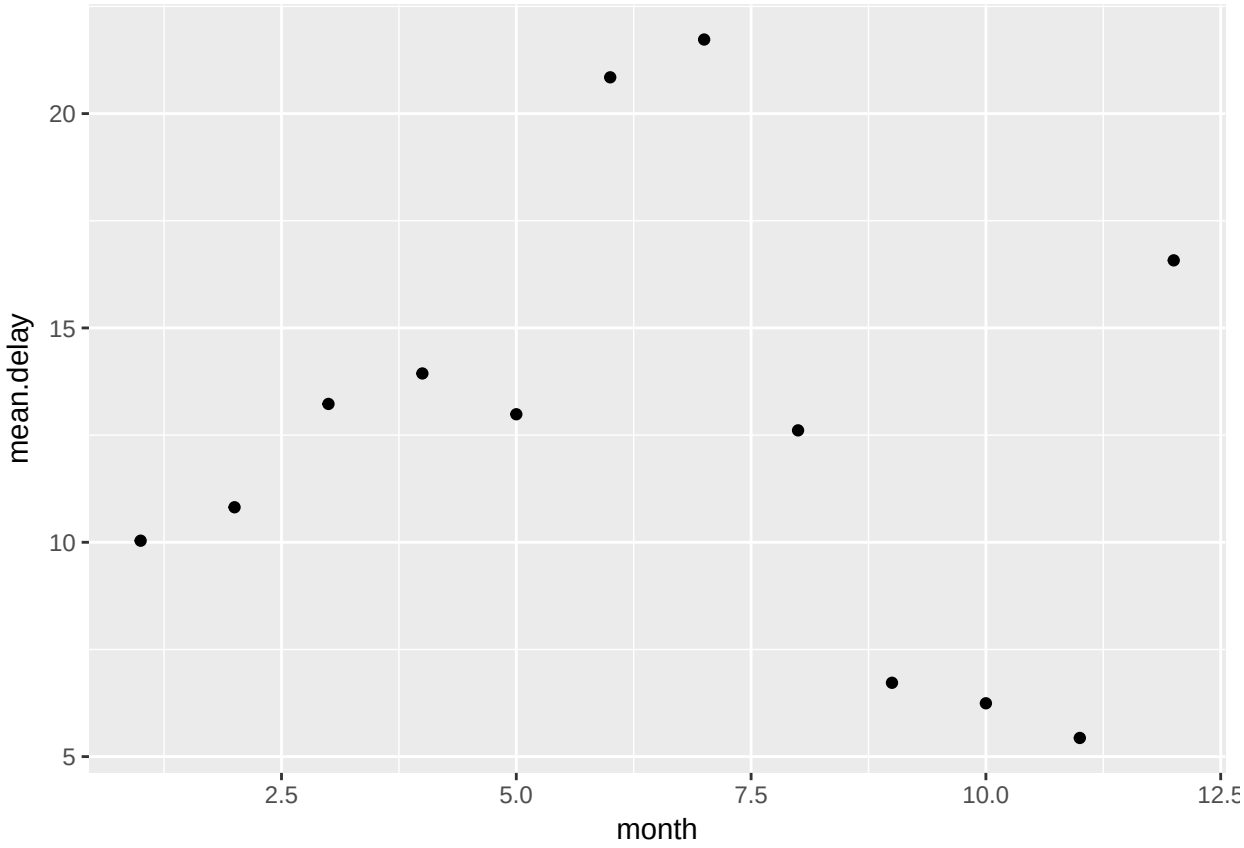
Question 3

```
months.data <- flights %>%
  group_by(month) %>%
  summarize(mean.delay = mean(dep_delay, na.rm = TRUE))
```

July has the highest average departure delay, followed June. The next highest month is December. Many people are vacationing in the summer (as well as during the Winter Break), so the large volume of travellers and flights may slow down the process and increase delays.

Question 4

```
ggplot(months.data, aes(x = month, y = mean.delay)) + geom_point()
```



I chose a scatter plot because the even though the months are a counted variable, they represent time so it makes sense to plot them as such. As mentioned in the previous question, the delays are highest during the late summer and the winter holiday. The delays are the smallest in the fall. After a peak in December, the average delap in January is low. Then, the mean delay increases to peak again in the summer.

Question 5

```
delta <- flights %>%
  filter(carrier =="DL")

t.test(delta$dep_delay, mu = 5)
```

```
##
##  One Sample t-test
##
## data:  delta$dep_delay
## t = 23.455, df = 47760, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  8.908139 9.620870
## sample estimates:
## mean of x
##  9.264505
```

The null hypothesis is that the mean delay for Delta flights is 5 minutes. The hypothesis is that the mean delay is different than 5 minutes. I do not know whether this difference (if there is one) is earlier or later

than 5 minutes so I used the default 2 tailed single sample t test. The data were collected randomly. The p value is $<$ 2.2e -16. This means that there is a miniscule chance that the larger observed mean delay of 9.26 minutes is due to chance. There is a very large chance that the difference between the means is significant.

Bonus

```
arrival.delays <- flights %>% select(arr_delay)

# for i in range 1: len(data)-50
#   make a sliding window starting from i of len(50)
#   subset arrival.delays using that window

# Method 1
#   plot distribution of delays in the window

# Method 2
# calculate n*p >=10 and n*(1-p) >=10 to see if it is normally distributed
# where n is 50.
```

I have neither given nor received unauthorized aid on this exam. Jennifer Johnson