Homework 3

NAME

DATE

Introduction

You will be creating some basic plots using `ggplot2` package and managing data using `dplyr` package in this homework assignment. You will use two data sets, `ncbirths` which is in `openintro` package and `flights` which come part of the `nycflights13` package.

The code chunk below sets some code chunk options (using `opts_chunk` from the `knitr` package) to make your knitted report output more readable. It is good habit to load all packages and data in the first code chunk.

```
knitr::opts_chunk$set(warning=FALSE, message=FALSE, fig.height=4, fig.width=5, fig.align='center')
library(ggplot2)
library(dplyr)
flights <- nycflights13::flights
ncbirths <- openintro::ncbirths
```

Univariate plots

This section asks you to create data visualizations or summaries from the `ncbirths` data set.

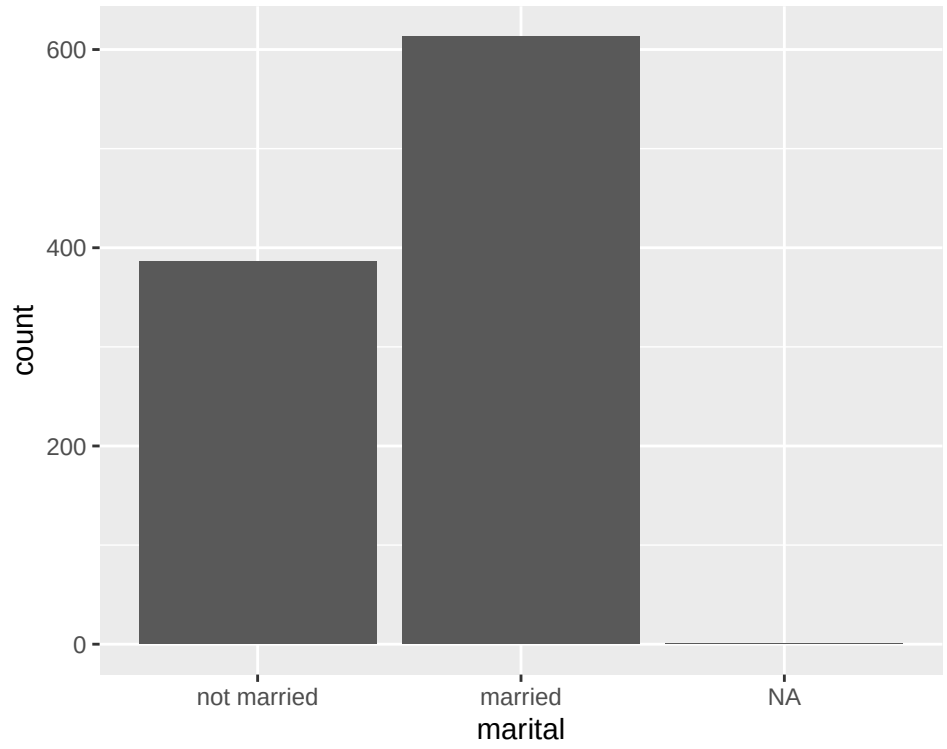
1. Create a table of marital status (`marital`) from NCbirths.

```
table(ncbirths$marital)
```

```
##
## not married     married
##         386         613
```

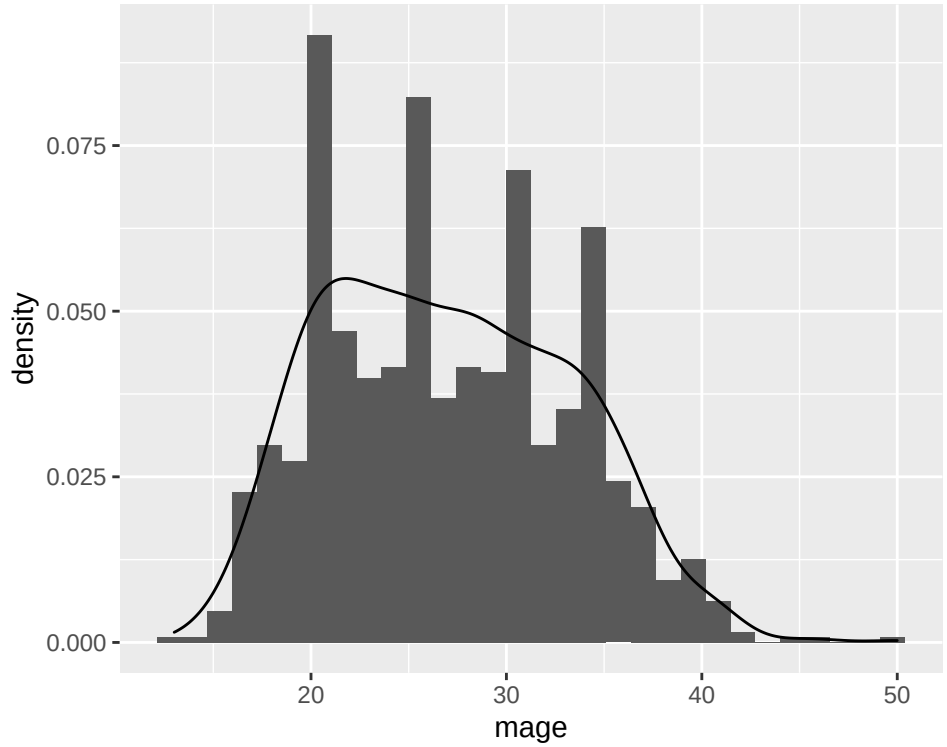
2. Create a barchart of marital status (same as above)

```
ggplot(ncbirths, aes(marital)) + geom_bar()
```



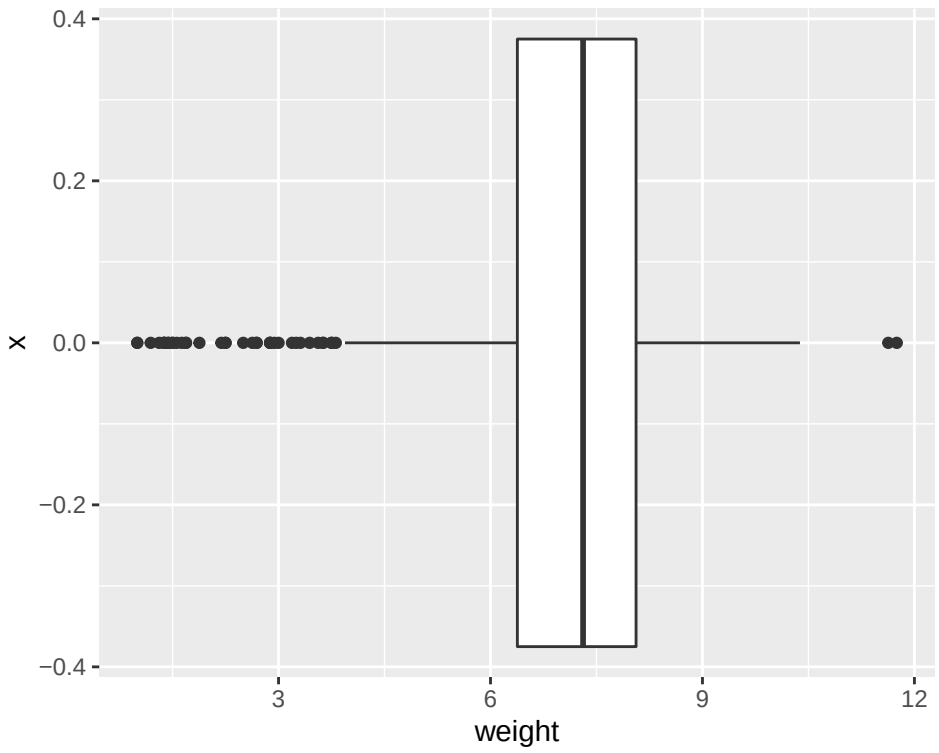
3. Create a histogram of mothers age (`mage`) with an overlaid density plot in a different color. Be sure that both density curve and histogram can be seen.

```
ggplot(ncbirths, aes(mage)) + geom_histogram(aes(y=..density..)) + geom_density()
```



4. Create a horizontal boxplot of weight of the baby (`weight`) I don't care how they make this plot, so long as it looks like a horizontal boxplot.

```
ggplot(ncbirths, aes(0, weight)) + geom_boxplot() + coord_flip()
```



Bivariate plots (This section still uses the `ncbirth` data set.)

1. Create a two-way frequency table of maturity status (`mature`) against smoking `habit`

```
table(ncbirths$mature, ncbirths$habit)
```

```
##
##                nonsmoker smoker
##   mature mom         121     11
##   younger mom        752    115
```

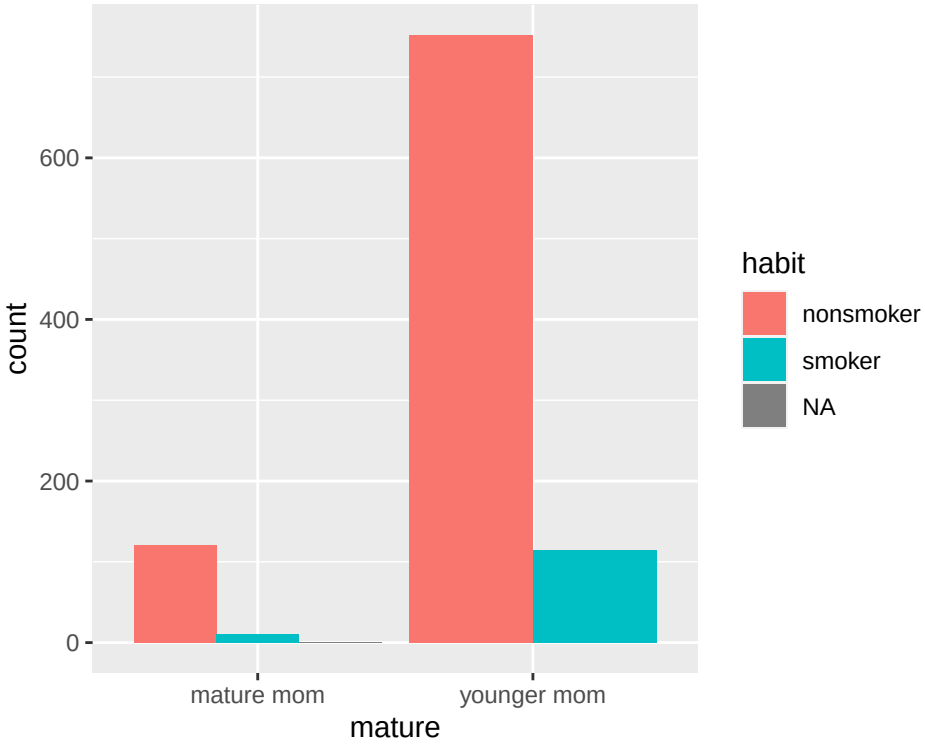
2. Create a proportion table of smoking habit *within* maturity status. Round to 3 digits.

```
table(ncbirths$mature, ncbirths$habit) %>% prop.table(margin=1) %>% round(3)
```

```
##
##                nonsmoker smoker
##   mature mom       0.917  0.083
##   younger mom      0.867  0.133
```

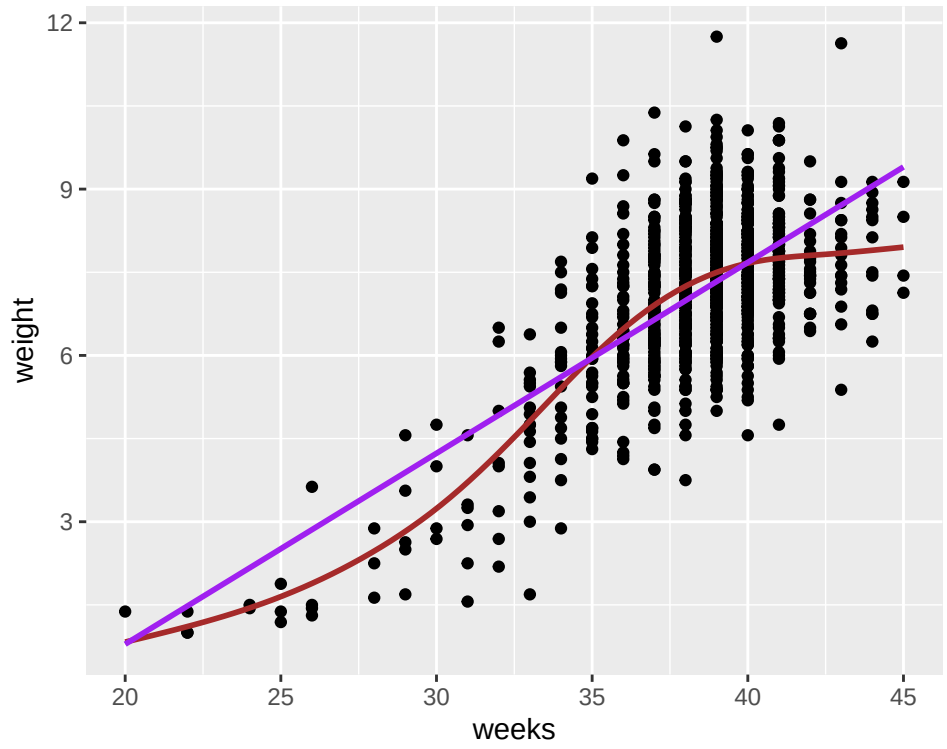
3. Create a grouped barchart that reflects the frequencies you calculated above. Think carefully which variable goes on the `x` axis, and which one is used for the `fill`

```
ggplot(ncbirths, aes(mature, fill=habit)) + geom_bar(position="dodge")
```



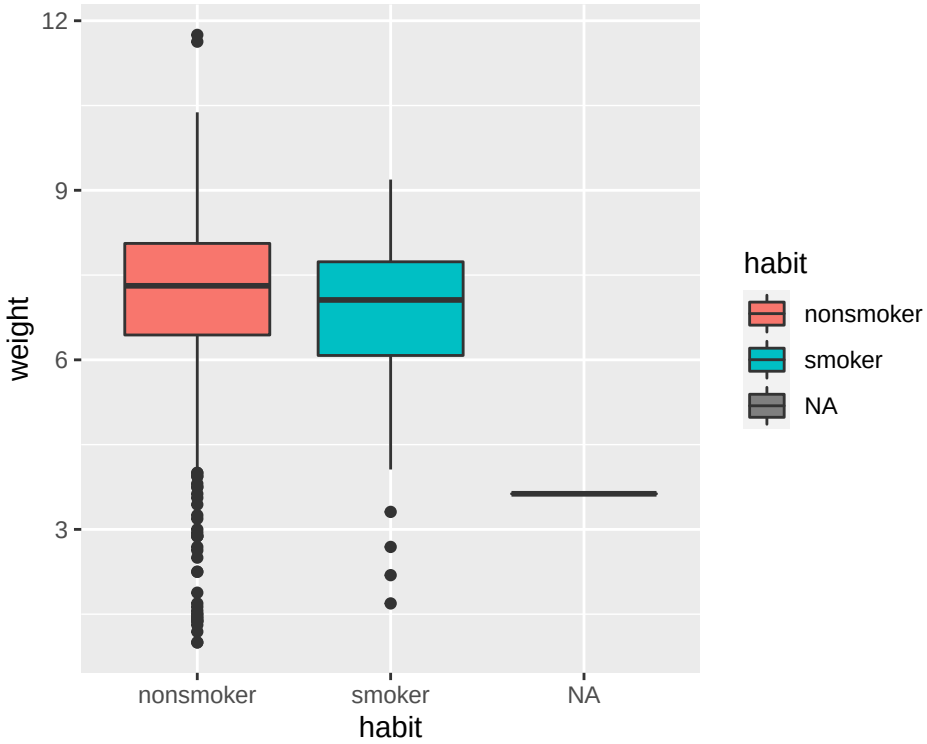
4. Create a scatterplot of length of pregnancy in `weeks` and the babies `weight`. Include a smoother line in brown, and a best fit linear model line in purple

```
ggplot(ncbirths, aes(weeks, weight)) +
  geom_point() +
  geom_smooth(se=FALSE, color="brown") +
  geom_smooth(se=FALSE, method="lm", color="purple")
```



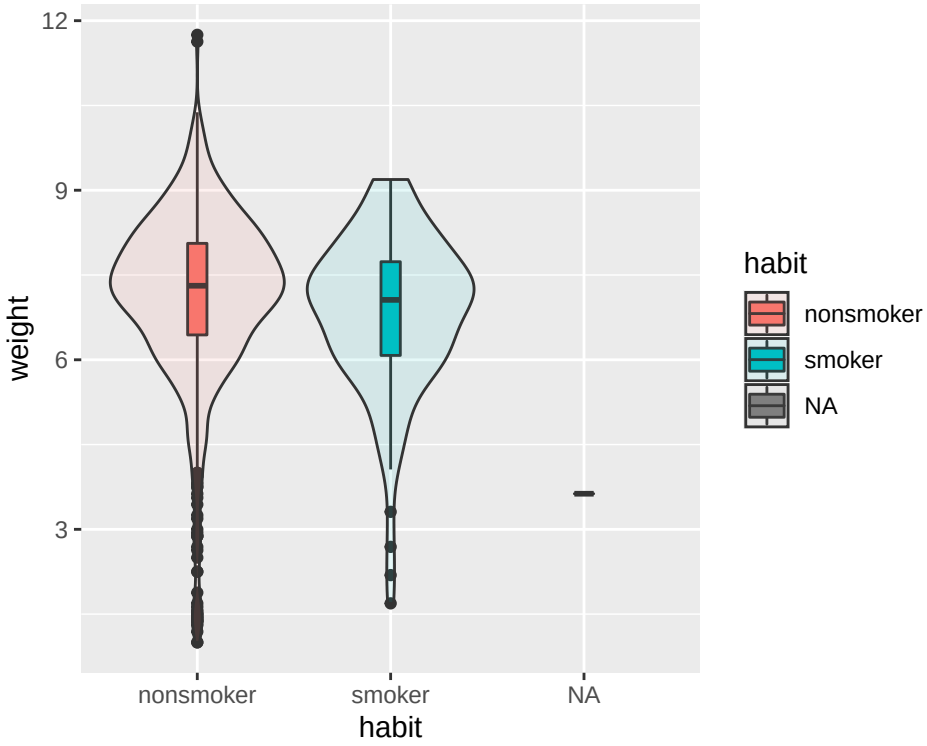
5. Creat a grouped boxplots of baby `weight` by mothers smoking `habit`. Make sure you `fill` the boxes by `habit` as well.

```
ggplot(ncbirths, aes(habit, weight, fill=habit)) + geom_boxplot()
```



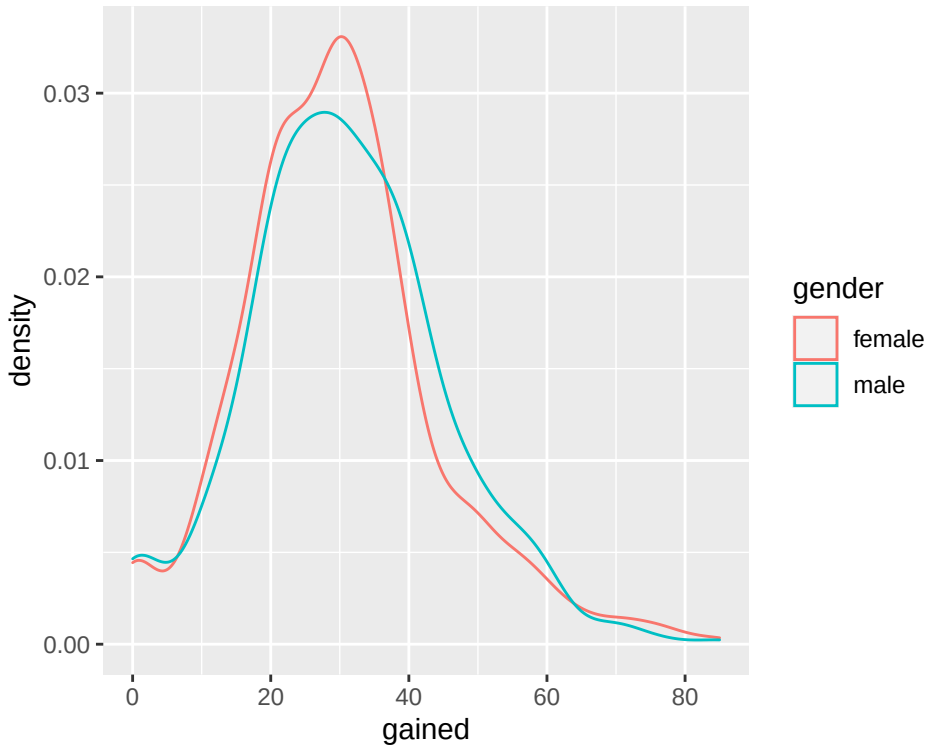
6. Replicate the same plot as above, but overlay a violin plot and change the transparency of both violin and boxplot layers.

```
ggplot(ncbirths, aes(habit, weight, fill=habit)) +
  geom_boxplot(width=.1) +
  geom_violin(alpha=.1)
```



7. Create an overlaid density plots of weight `gained` by babies `gender`. Do not apply a `fill`, only use the `color` aesthetic.

```
ggplot(ncbirths, aes(gained, color=gender)) + geom_density()
```



Data management and aggregation

This section uses the `dplyr` and `nycflights13` packages. Use the `flights` data set for the next few exercises.

- At each step use the assignment operator `<-` to store the results into a new data table and use that data in the next step.
- At each step, print out the resulting data frame so you can see the results.

Example *(not run)*

```
p1 <- planes %>% select(type)
p1
```

1. Use `select()` to extract the following variables: `origin`, `distance`, and `air_time`, `dest`. Save this result as a data set named `f1`.

```
f1 <- flights %>% select(origin, distance, air_time, dest)
f1
```

```
## # A tibble: 336,776 x 4
##    origin distance air_time dest
##    <chr>     <dbl>    <dbl> <chr>
##  1 EWR        1400      227 IAH
##  2 LGA        1416      227 IAH
##  3 JFK        1089      160 MIA
##  4 JFK        1576      183 BQN
```

```
##  5 LGA         762      116 ATL
##  6 EWR         719      150 ORD
##  7 EWR        1065      158 FLL
##  8 LGA         229       53 IAD
##  9 JFK         944      140 MCO
## 10 LGA         733      138 ORD
## # ... with 336,766 more rows
```

2. Take the `f1` data set and `filter()` to select only the flights whose destination (`dest`) is Atlanta (ATL). Save this result as `f2`. *Hint: the destination variable is a character variable, so think carefully about how you specify ATL.*

```
f2 <- f1 %>% filter(dest=="ATL")
f2
```

```
## # A tibble: 17,215 x 4
##    origin distance air_time dest
##    <chr>     <dbl>    <dbl> <chr>
##  1 LGA         762      116 ATL
##  2 LGA         762      134 ATL
##  3 JFK         760      128 ATL
##  4 EWR         746      120 ATL
##  5 LGA         762      126 ATL
##  6 LGA         762      126 ATL
##  7 JFK         760      126 ATL
##  8 LGA         762      132 ATL
##  9 LGA         762      123 ATL
## 10 LGA         762      129 ATL
## # ... with 17,205 more rows
```

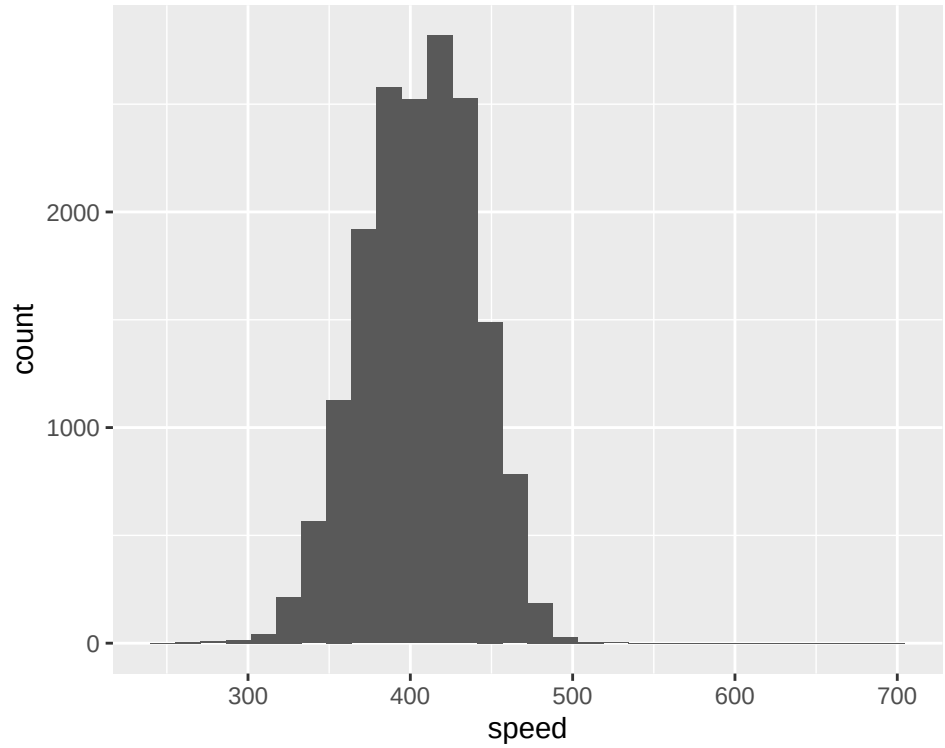
3. Take the `f2` data set and use `mutate()` to create a new variable `speed` that calculates speed of the plane as `distance/air_time*60`. Save this result as `f3`.

```
f3 <- f2 %>% mutate(speed = distance / air_time*60)
f3
```

```
## # A tibble: 17,215 x 5
##    origin distance air_time dest  speed
##    <chr>     <dbl>    <dbl> <chr> <dbl>
##  1 LGA         762      116 ATL    394.
##  2 LGA         762      134 ATL    341.
##  3 JFK         760      128 ATL    356.
##  4 EWR         746      120 ATL    373
##  5 LGA         762      126 ATL    363.
##  6 LGA         762      126 ATL    363.
##  7 JFK         760      126 ATL    362.
##  8 LGA         762      132 ATL    346.
##  9 LGA         762      123 ATL    372.
## 10 LGA         762      129 ATL    354.
## # ... with 17,205 more rows
```

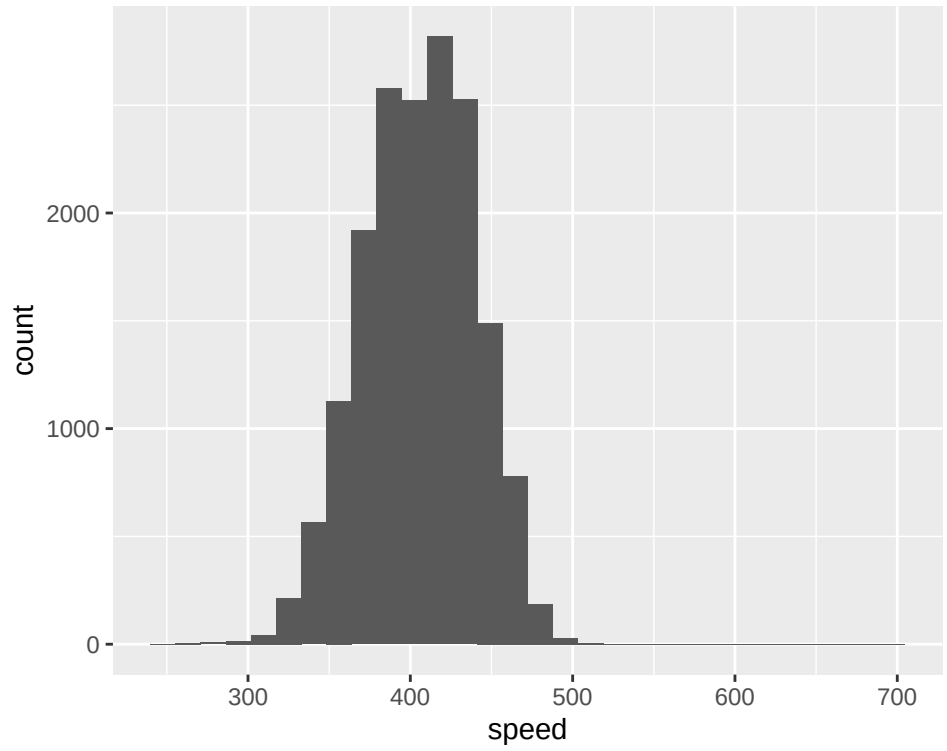
4. Use `ggplot` to plot the distribution of the planes speed on it's way to Atlanta using a histogram.

```
library(ggplot2)
ggplot(f3, aes(x=speed)) +  geom_histogram()
```



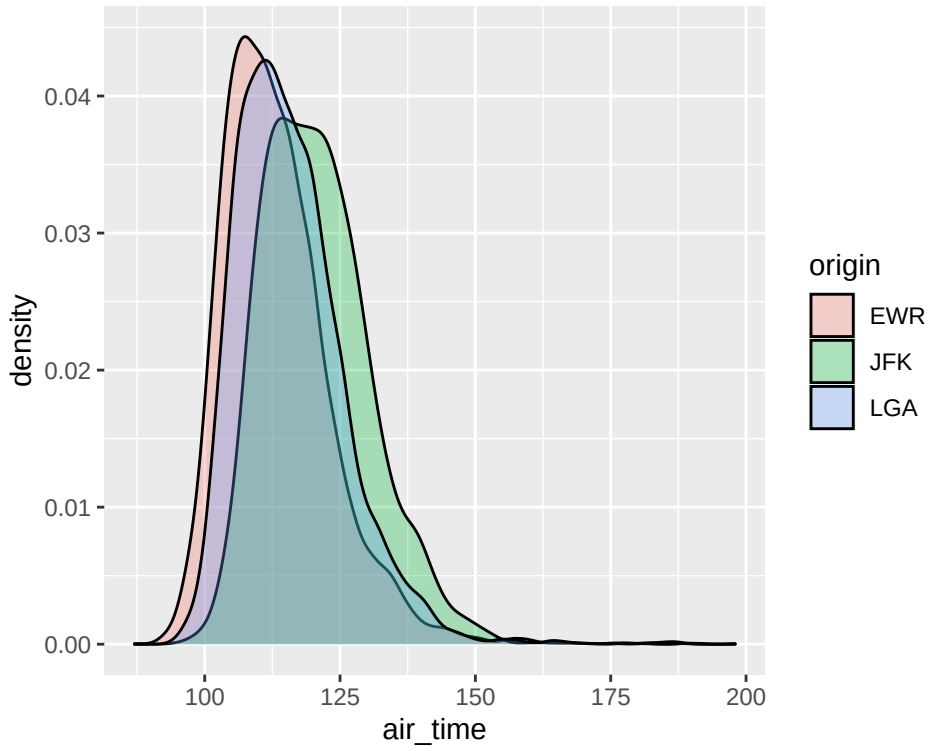
5. Use `dplyr` chaining magic (%>%) to combine questions 1-4 in one step.

```
flights %>% select(origin, distance, air_time, dest) %>% filter(dest=="ATL") %>%
            mutate(speed = distance / air_time*60) %>%
            ggplot(aes(x=speed)) +  geom_histogram()
```



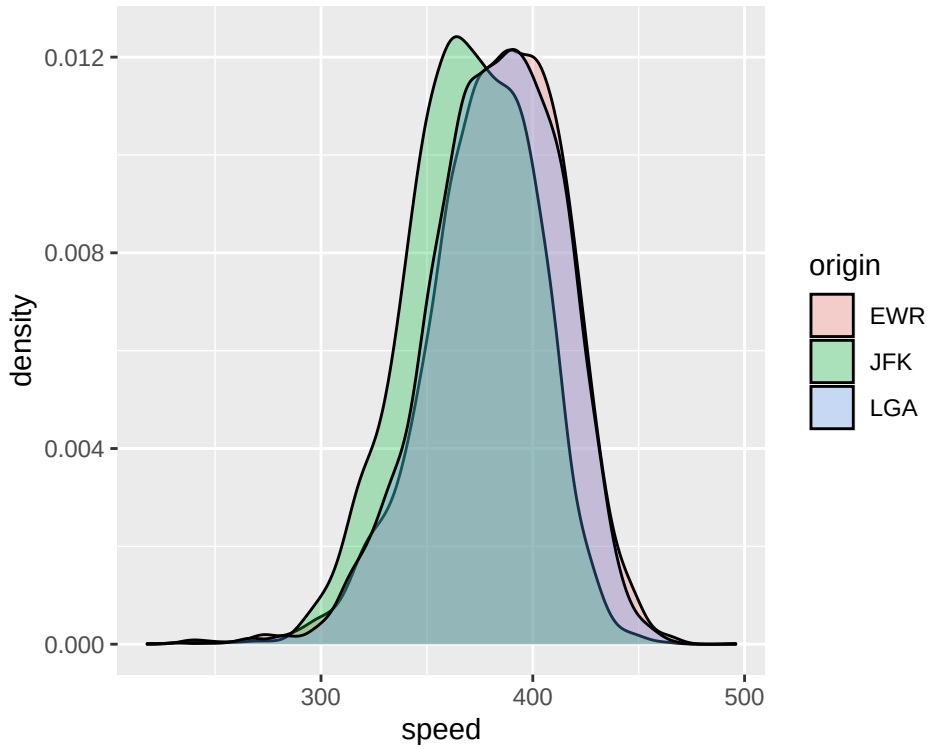
6. The three airports in the NYC region are all pretty close together. Do they all have the same travel time to Chicago O'Hare (ORD)? Use the same tactic as you did in steps 1-4 (or 5) by subsetting to the desired destination, then create overlapping density plots of `air_time` where each density plot is colored (or filled) by the variable `origin`.

```
flights %>% filter(dest=="ORD") %>%
            ggplot(aes(x=air_time, fill=origin)) +  geom_density(alpha=.3)
```



7. Repeat the process in problem 6, but when creating overlapping density plots, instead of `air_time`, define and use `speed` as in problem 3.

```
flights %>% select(origin, distance, air_time, dest) %>% filter(dest=="ORD") %>%
            mutate(speed = distance / air_time*60) %>%
            ggplot(aes(x=speed, fill=origin)) +  geom_density(alpha=.3)
```


0.012
0.008
0.004
0.000
density
300
400
500
speed
origin
EWR
JFK
LGA