Problem Set 1 Spring 2022

Note: Grading is based both on your graphs and verbal explanations. Follow all best practices as discussed in class, including choosing appropriate parameters for all graphs. *Do not expect the assignment questions to spell out precisely how the graphs should be drawn. Sometimes guidance will be provided, but the absense of guidance does not mean that all choices are ok.*

Read *Graphical Data Analysis with R*, Ch. 3

```r
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.2     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(gridExtra)
```

```
##
## Attaching package: ’gridExtra’
```

```
## The following object is masked from ’package:dplyr’:
##
##     combine
```

```r
library(ggridges)
```

1. SATs

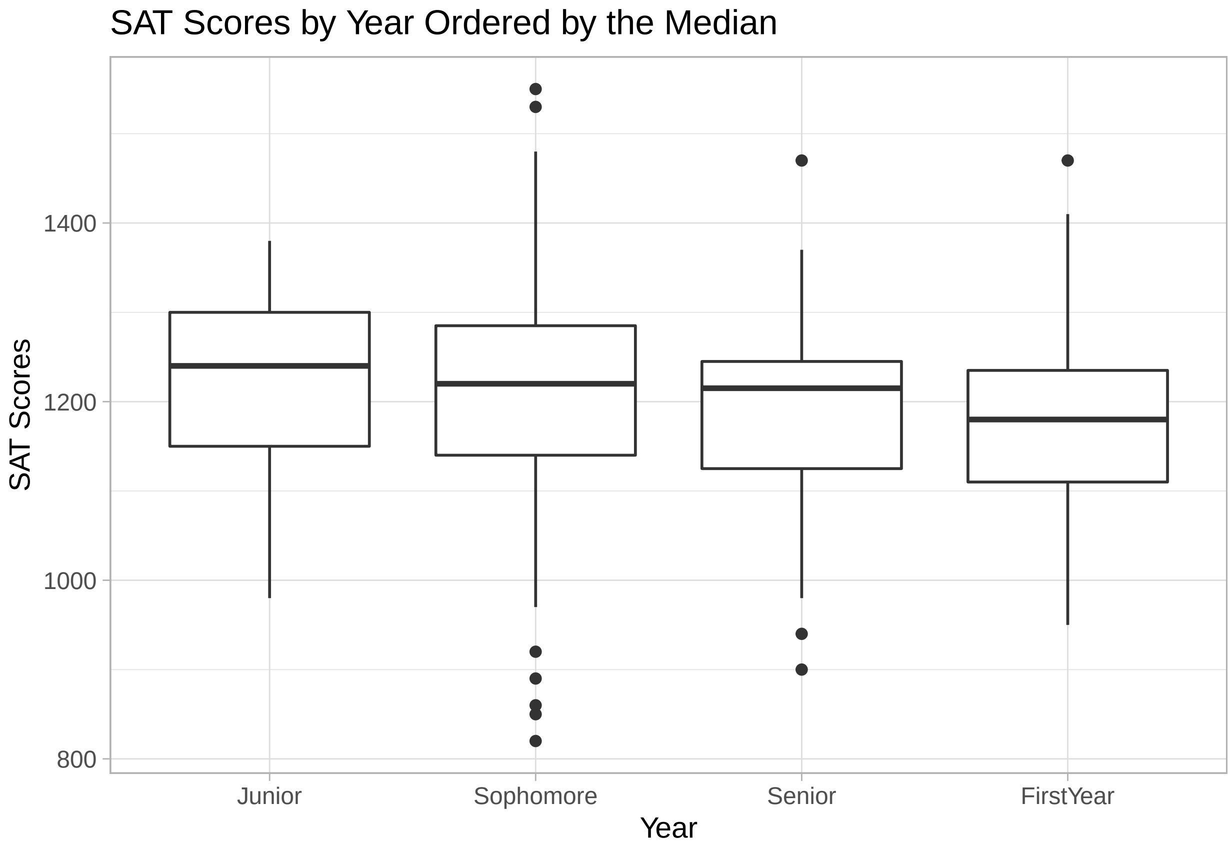
[7 points]

Data: *StudentSurvey* in **Lock5withR** package (Remember to add a proper title and labels to every plot.)

```
library(Lock5withR)
```

a) Draw multiple horizontal boxplots of `SAT`, by `Year`. What do you observe? (Hint:You can remove all blank and NAs)

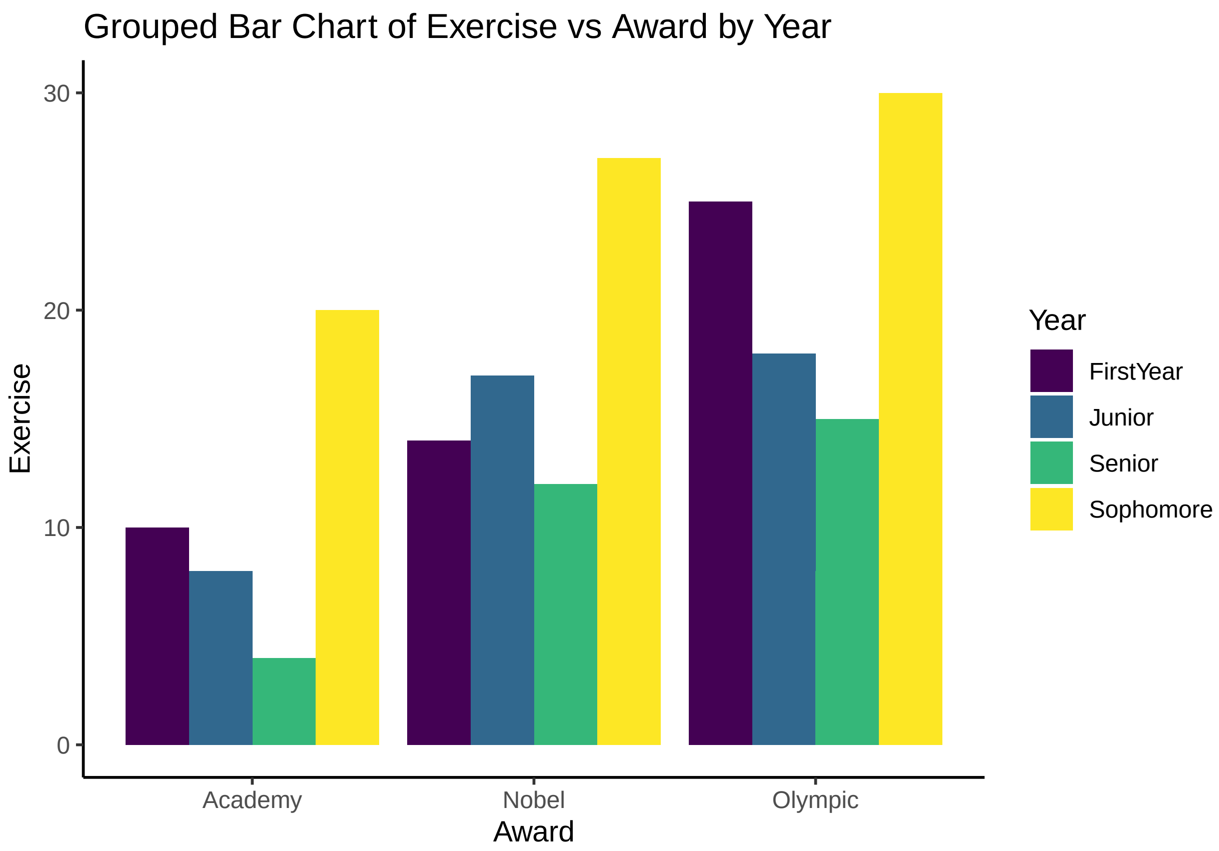
```
StudentSurvey %>% drop_na() %>%
  ggplot(aes(x=reorder(Year, -SAT, median), y=SAT)) +
  geom_boxplot() +
  xlab("Year") + ylab("SAT Scores") +
  ggtitle("SAT Scores by Year Ordered by the Median") + theme_light()
```



When ordered by median, we see that the the median SAT scores of Juniors is the highest and those of First Years is the least. We also see that there are many outliers in the Sophomore class. We also observe that the range of Senior scores is approximately the lowest, suggesting that Senior scores don't vary too much.

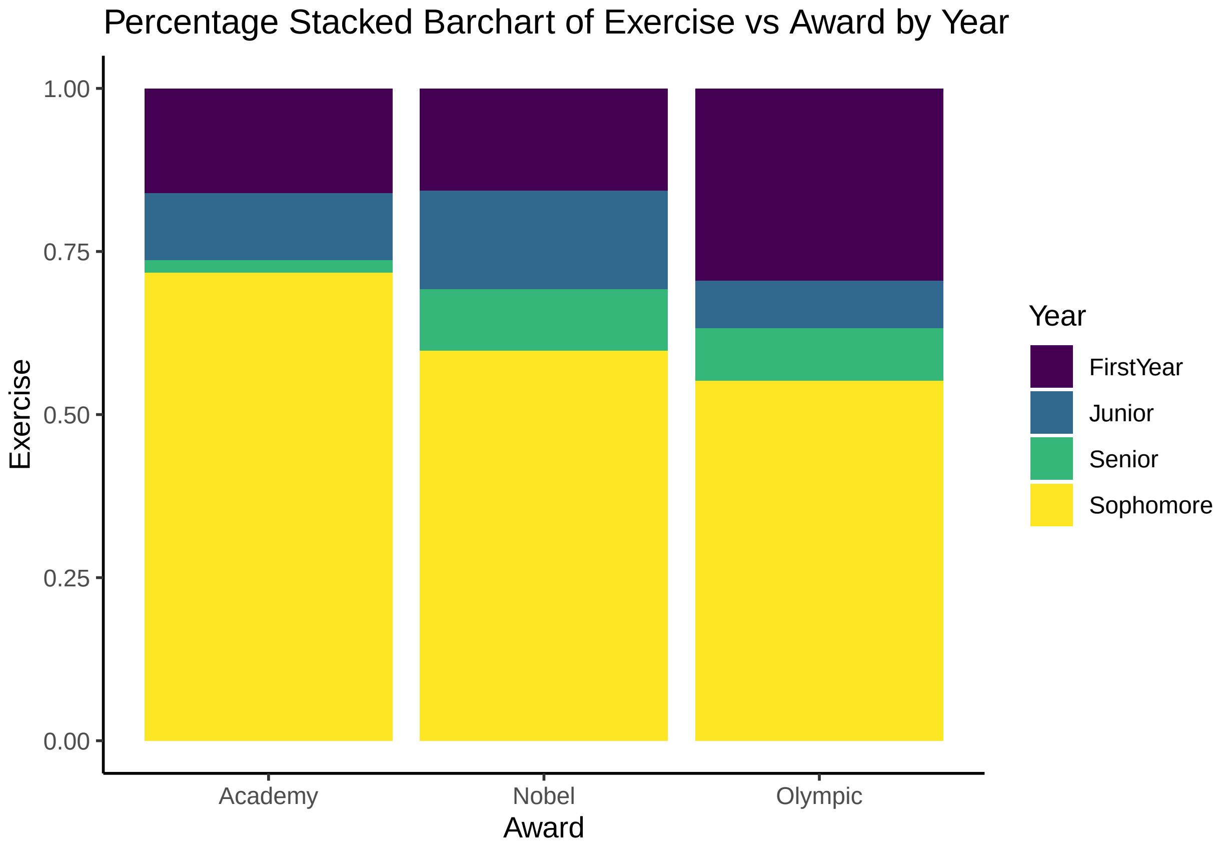
b) Draw a grouped bar chart of average `Exercise` by `Award` filled with `Year`. (You can ignore NAs.)

```
StudentSurvey %>% drop_na() %>%
  ggplot(aes(x=Award, y=Exercise, fill=Year)) +
  geom_bar(stat="identity", position="dodge") +
  ggtitle("Grouped Bar Chart of Exercise vs Award by Year") +
  scale_fill_viridis(discrete = T) +
  theme_classic()
```



c) Draw a percentage stacked barchart (each bar = 100%) of average `Exercise` by `Award` filled with `Year`. Compare to the plot in b), which one do you prefer and why?

```r
StudentSurvey %>% drop_na() %>%
  ggplot(aes(x=Award, y=Exercise, fill=Year)) +
  geom_bar(stat="identity", position="fill") +
  ggtitle("Percentage Stacked Barchart of Exercise vs Award by Year") +
  scale_fill_viridis(discrete = T) +
  theme_classic()
```



I would prefer the grouped barchart because it gives me information about the actual amount of average Exercise per Award group and Year, whereas the percent stacked bargraph doesn't give me any information about the actual quantity of average Exercise. The grouped barchart also lets me compare the amount of Exercise by award category, so I immediately know which group has lowest or highest Exercise, but I can't achieve this using the percentage stacked barchart. But sometimes it depends on what information I'm trying to glean from the plots. If I simply want to compare the amount of Exercise soley within Award groups by Year, then I would look at the percetage stacked bargraph.

2. Bad Drivers

[7 points]

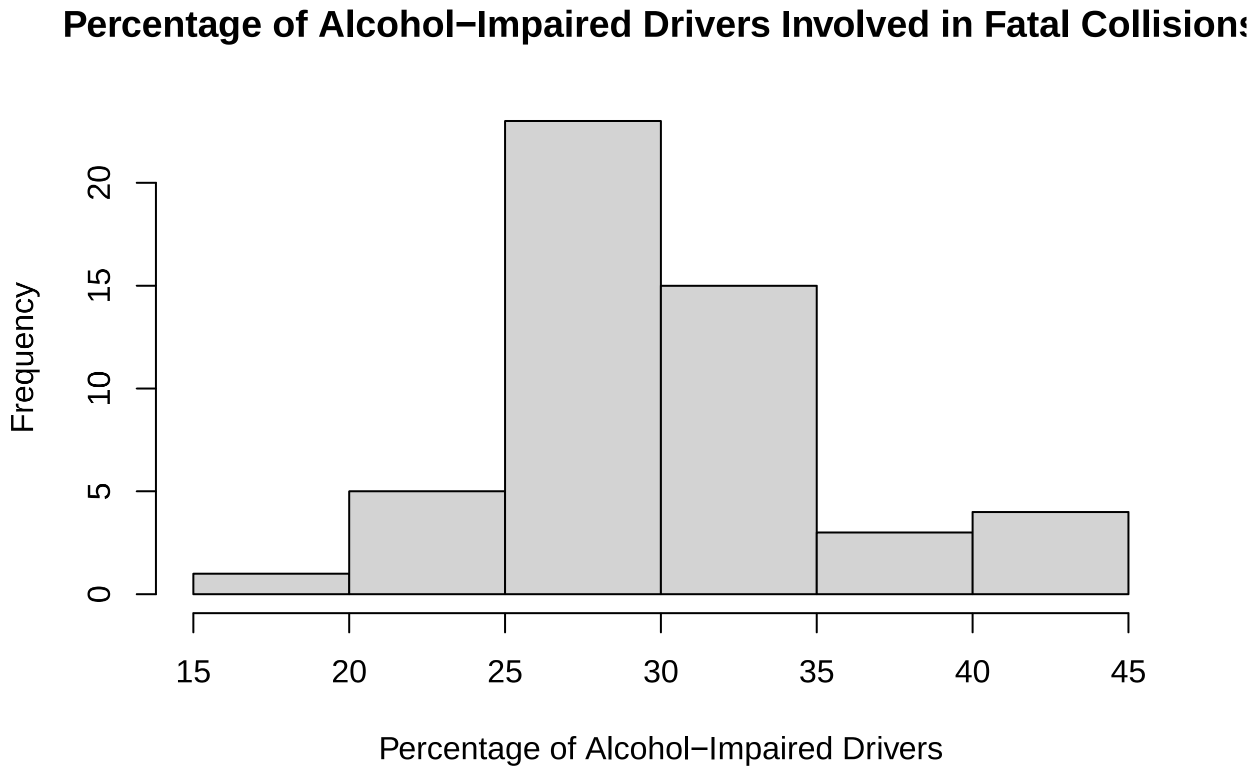
Data: *bad_drivers* in **fivethirtyeight** package

```
library("fivethirtyeight")
```

```
## Some larger datasets need to be installed separately, like senators and
## house_district_forecast. To install these, we recommend you install the
## fivethirtyeightdata package by running:
## install.packages('fivethirtyeightdata', repos =
## 'https://fivethirtyeightdata.github.io/drat/', type = 'source')
```

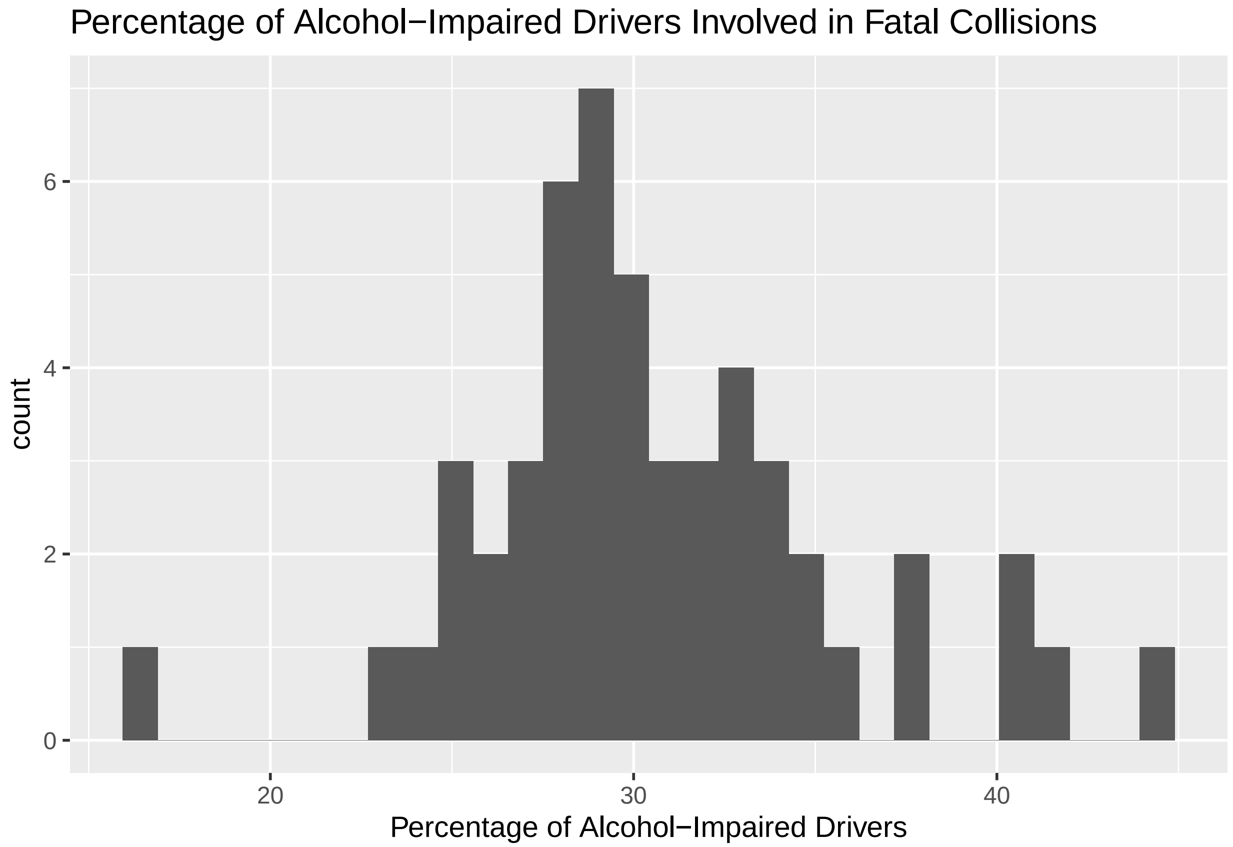
a) Draw two histograms–one with base R and the other with **ggplot2**–of the variable representing the Percentage of drivers involved in fatal collisions who were alcohol-impaired without setting any parameters. What is the default method each uses to determine the number of bins? (For base R, show the calculation.) Which do you think is a better choice for this dataset and why?

```
#base R
hist(bad_drivers$perc_alcohol, main="Percentage of Alcohol-Impaired Drivers Involved in Fatal Collision
```



```
#?hist

#ggplot
bad_drivers %>% ggplot(aes(x=perc_alcohol)) + geom_histogram() + xlab("Percentage of Alcohol-Impaired D
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

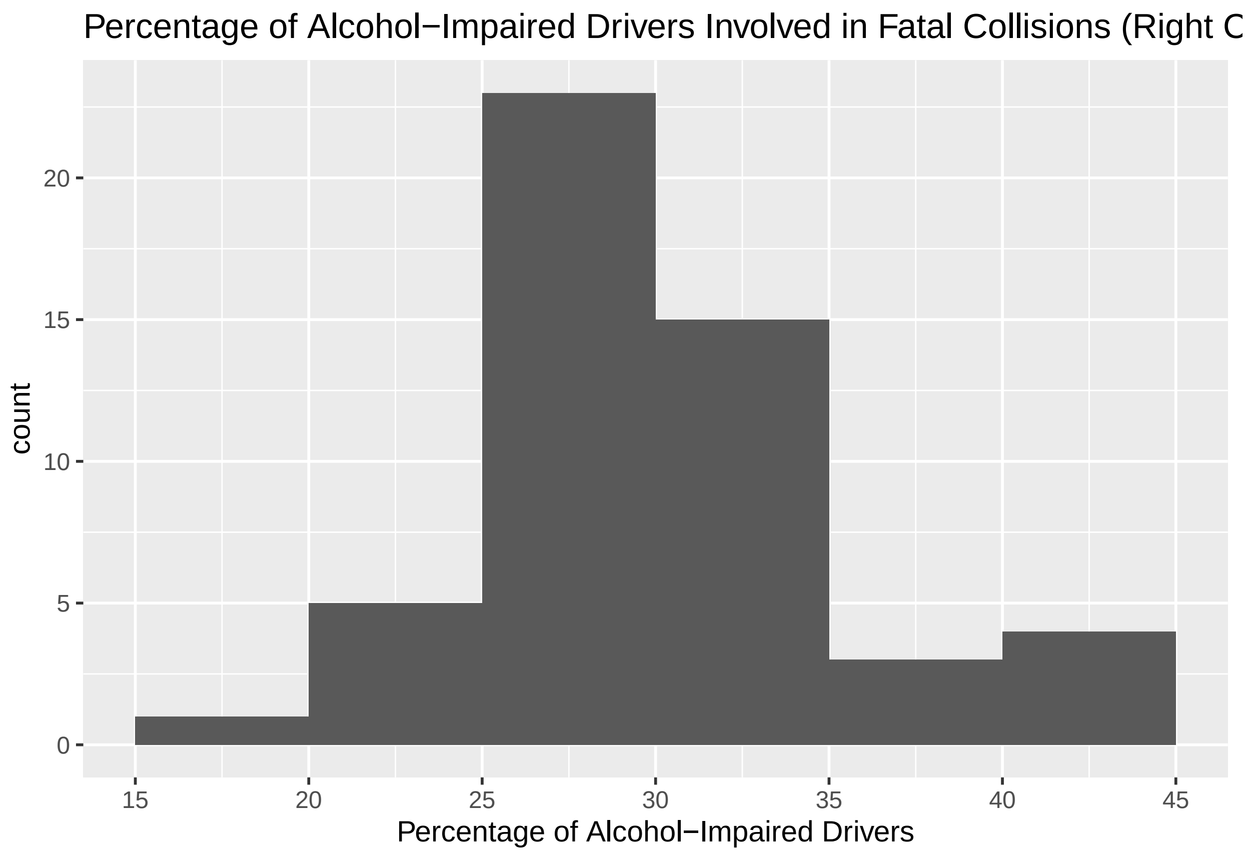


In Base R, the default breaks = "Sturges", which means R uses the formula ceiling(log2(length(x)) + 1). So, breaks = ceiling(log2(51) + 1) = ceiling(5.67 + 1) = 7. This means that there are 6 bins. ggplot by default sets number of bins to 30.

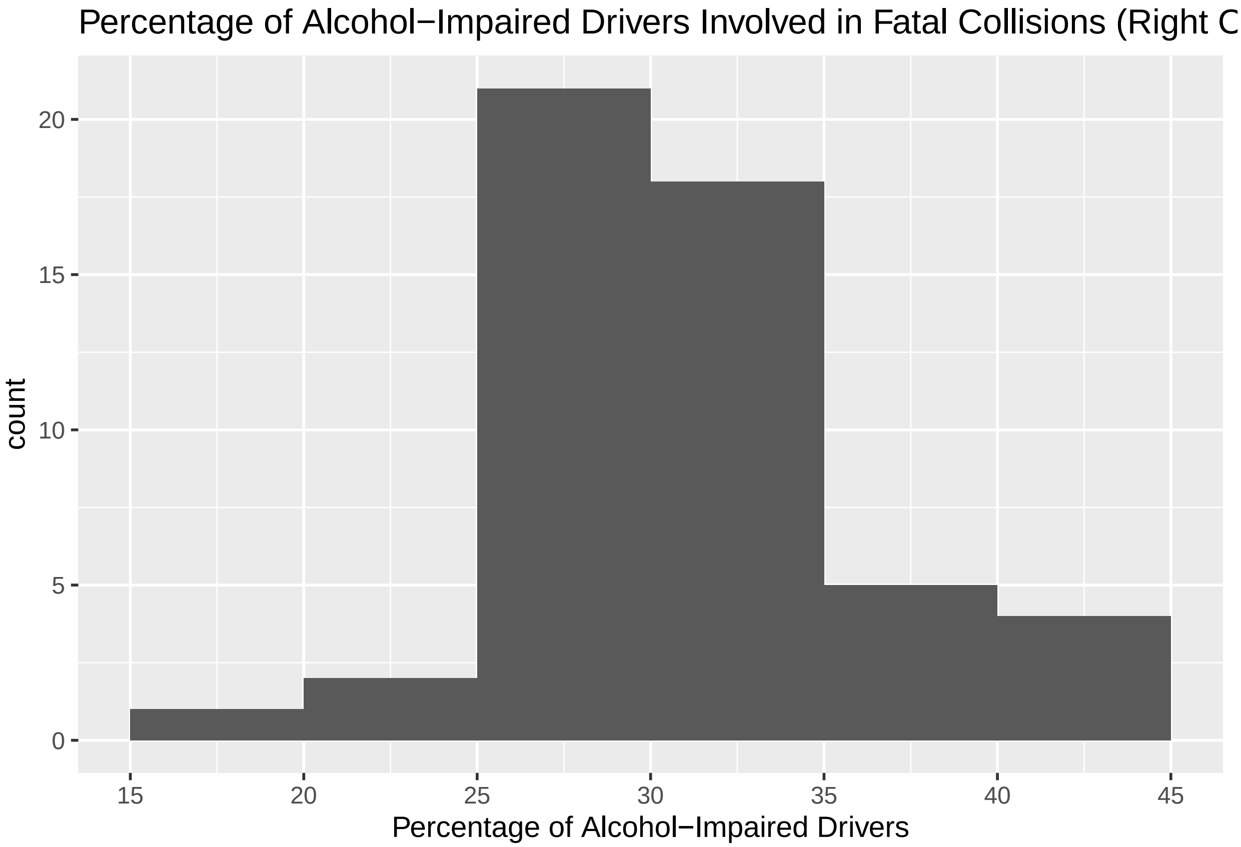
In this case, I prefer the Base R histogram because it shows a smoother representation of the histogram and I can better understand the general shape of the data.

b) Draw two histograms of the `perc_alcohol` variable with boundaries at multiples of 5, one right closed and one right open. Every boundary should be labeled (15, 20, 25, etc.)

```
#right-closed - [15, 20]
bad_drivers %>% ggplot(aes(x=perc_alcohol)) +
  geom_histogram(closed = "right", binwidth = 5, boundary = 15) + ggtitle("Percentage of Alcohol-Impaire
  scale_x_continuous(breaks = seq(15,45,by=5)) + xlab("Percentage of Alcohol-Impaired Drivers")
```

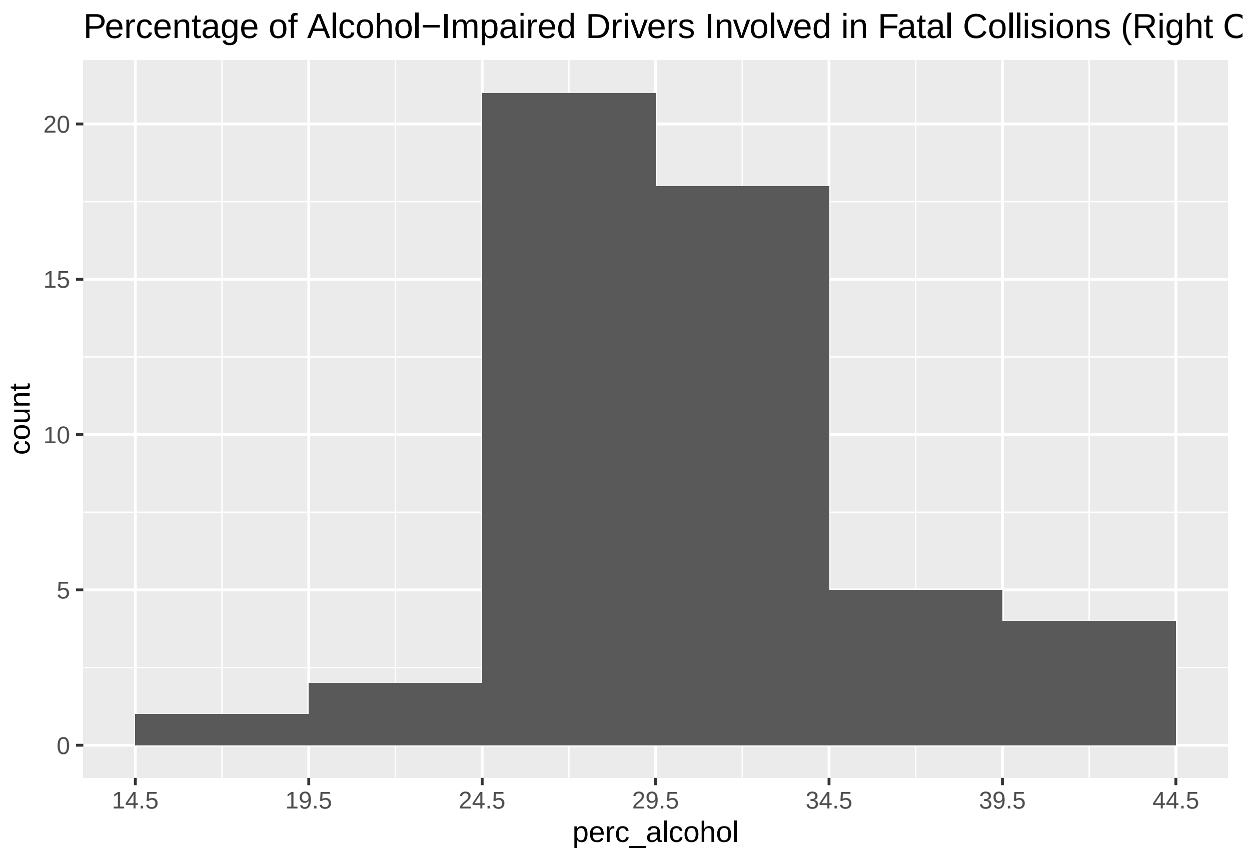


```
#right-open - [15, 20)
bad_drivers %>% ggplot(aes(x=perc_alcohol)) +
  geom_histogram(closed = "left", binwidth = 5, boundary = 15) + ggtitle("Percentage of Alcohol-Impaire
  scale_x_continuous(breaks = seq(15,45,by=5)) + xlab("Percentage of Alcohol-Impaired Drivers")
```

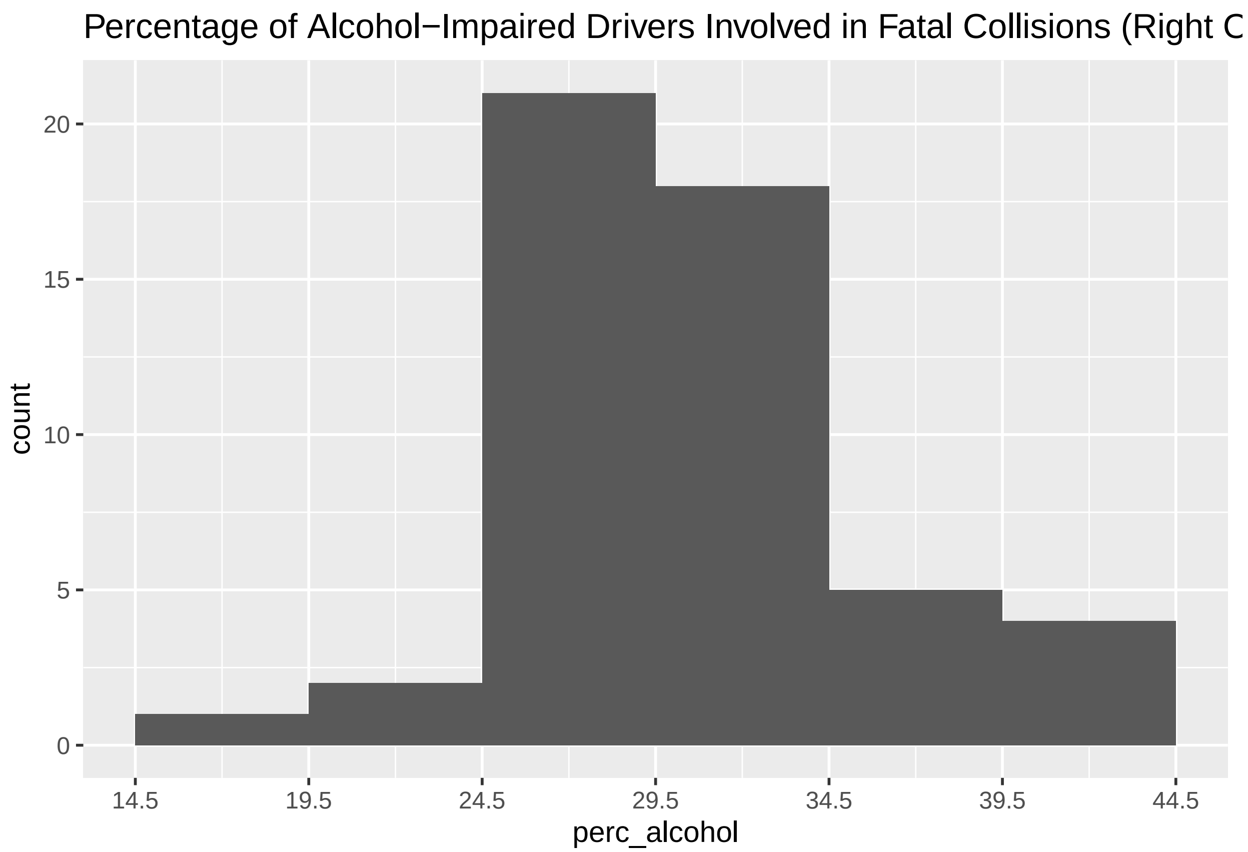


c) Adjust parameters–the same for both–so that the right open and right closed versions become identical. Explain your strategy.

```
#right-closed
bad_drivers %>% ggplot(aes(x=perc_alcohol)) +
  geom_histogram(closed = "right", binwidth = 5, boundary = 14.5) + ggtitle("Percentage of Alcohol-Impai
  scale_x_continuous(breaks = seq(14.5,44.5,by=5))
```



```
#right-open
bad_drivers %>% ggplot(aes(x=perc_alcohol)) +
  geom_histogram(closed = "left", binwidth = 5, boundary = 14.5) + ggtitle("Percentage of Alcohol-Impai
  scale_x_continuous(breaks = seq(14.5,44.5,by=5))
```



I figured that there were some values on the break values, so I decided to change the break values to increments of 5 starting with 14.5. I choose to end it at 44.5 because max(bad_drivers$perc_alcohol) = 44, so not including 45 was okay in this situation.

3.Titanic Survival

[8 points]

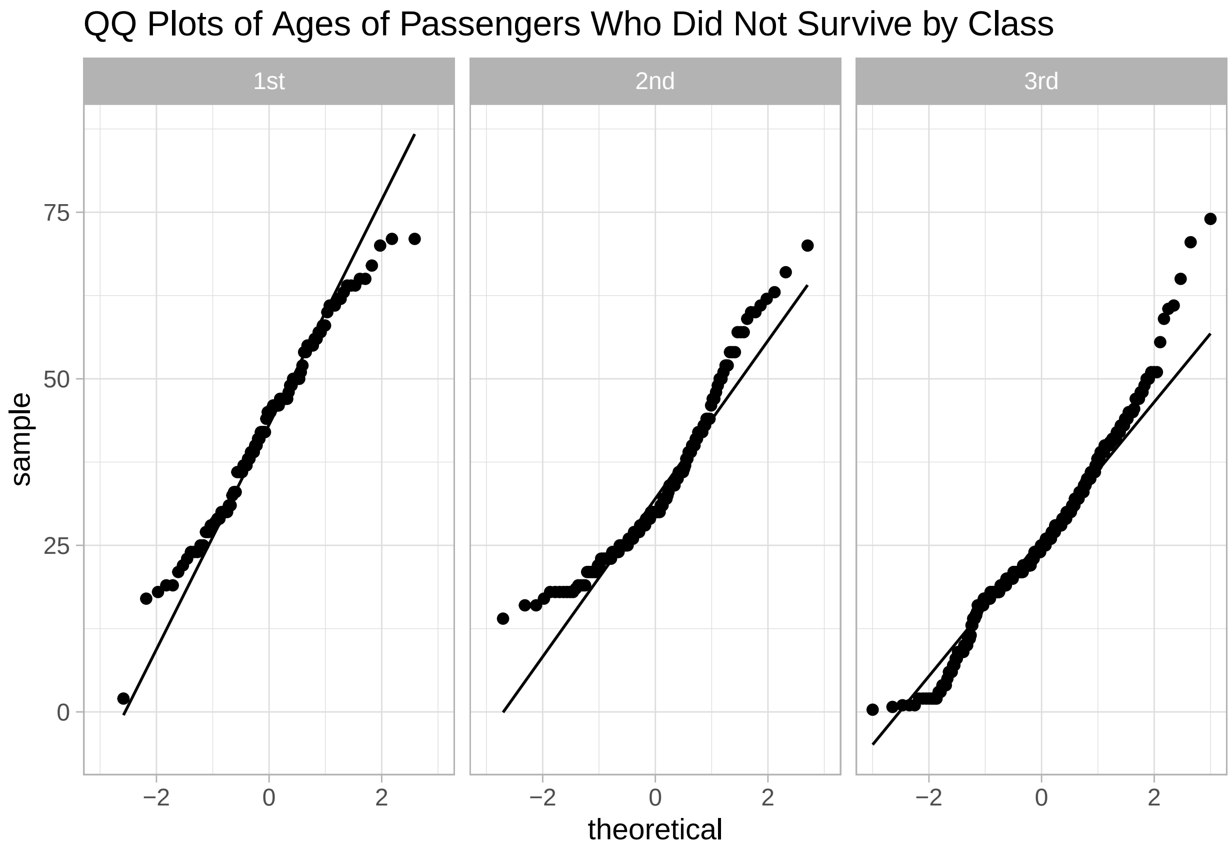
Data: *TitanicSurvival* in **carData** package

```
library(carData)
```

a) Use QQ (quantile-quantile) plots with theoretical normal lines to compare `age` of **passengers who did not survive from Titanic** for the three different levels of `passengerClass`. What are some findings and for which class does the distribution of the `age` variable appear to be closest to a normal distribution?

```
survived_no <- TitanicSurvival %>% filter(survived == "no")
survived_no %>%
  ggplot(aes(sample=age)) +
  geom_qq() + geom_qq_line() +
  facet_grid(.~passengerClass) +
  theme_light() +
  ggtitle("QQ Plots of Ages of Passengers Who Did Not Survive by Class")
```

```
## Warning: Removed 190 rows containing non-finite values (stat_qq).
## Warning: Removed 190 rows containing non-finite values (stat_qq_line).
```

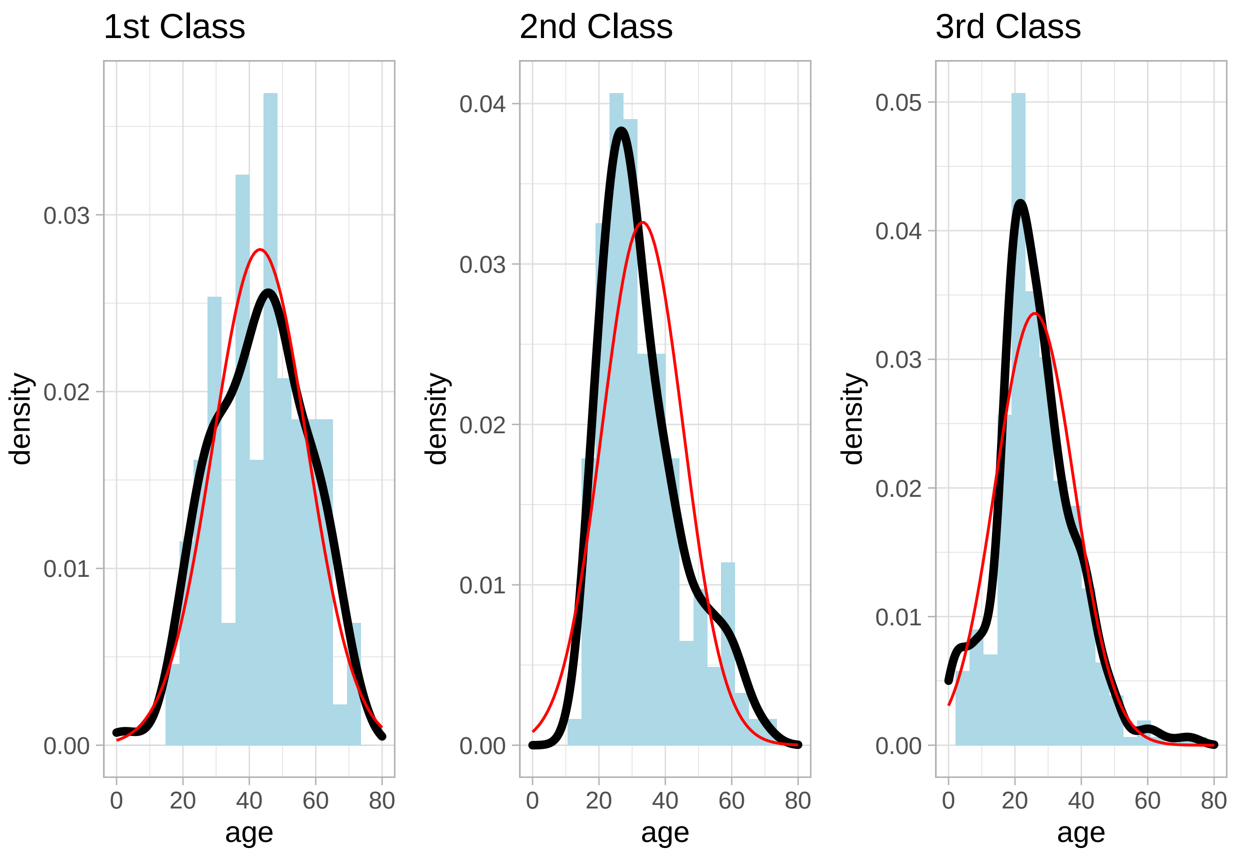


None of the ages in the 3 classes seem to be normally distributed. But ages in class 1 looks closest to one with a little deviation at the ends.

b) Draw density histograms with density curves and theoretical normal curves overlaid of `age` for the three passenger classes.

```r
#1st
first <- survived_no %>% filter(passengerClass == "1st")
first_plot <- first %>%
  ggplot(aes(x=age)) +
  geom_histogram(aes(y=..density..), fill="lightblue", bins = 20) +
  geom_density(lwd=1.5) +
  stat_function(fun = dnorm, args = list(mean = mean(first$age, na.rm = T), sd = sd(first$age, na.rm = 
  xlim(c(0, 80)) +
  ggtitle("1st Class") + theme_light()

#2nd
second <- survived_no %>% filter(passengerClass == "2nd")
second_plot <- second %>%
  ggplot(aes(x=age)) +
  geom_histogram(aes(y=..density..), fill="lightblue", bins = 20) +
  geom_density(lwd=1.5) +
  stat_function(fun = dnorm, args = list(mean = mean(second$age, na.rm = T), sd = sd(second$age, na.rm 
  xlim(c(0, 80)) +
  ggtitle("2nd Class") + theme_light()

#3rd
third <- survived_no %>% filter(passengerClass == "3rd")
third_plot <- third %>%
  ggplot(aes(x=age)) +
  geom_histogram(aes(y=..density..), fill="lightblue", bins = 20) +
  geom_density(lwd=1.5) +
  stat_function(fun = dnorm, args = list(mean = mean(third$age, na.rm = T), sd = sd(third$age, na.rm =
  xlim(c(0, 80)) +
  ggtitle("3rd Class") + theme_light()

grid.arrange(first_plot, second_plot, third_plot, nrow = 1)
```

```
## Warning: Removed 20 rows containing non-finite values (stat_bin).

## Warning: Removed 20 rows containing non-finite values (stat_density).

## Warning: Removed 2 rows containing missing values (geom_bar).

## Warning: Removed 12 rows containing non-finite values (stat_bin).

## Warning: Removed 12 rows containing non-finite values (stat_density).

## Warning: Removed 2 rows containing missing values (geom_bar).

## Warning: Removed 158 rows containing non-finite values (stat_bin).

## Warning: Removed 158 rows containing non-finite values (stat_density).

## Warning: Removed 2 rows containing missing values (geom_bar).
```



None of the plots look exactly normal. Maybe passenger class 1 almost looks normal with couple of deviations.

c) Use a statistical method of your choice, such as the Shapiro-Wilk test, to determine which `age` distribution is closest to a normal distribution.

```
#1st
shapiro.test(first$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  first$age
## W = 0.98537, p-value = 0.3174
```

```
#2nd
shapiro.test(second$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  second$age
## W = 0.92686, p-value = 8.135e-07
```

```r
#3rd
shapiro.test(third$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  third$age
## W = 0.97267, p-value = 1.923e-06
```

For all 3 tests, H0: Data is nornally distributed Ha: Data is not nornally distributed

We fail to reject H0 for 1st class passengers, which means that age in this case could be normally distributed. For passenger classes 2 and 3, we reject H0, and conclude that ages in these 2 groups are not normally distributed.

d) Did all of the methods for testing for normality (a, b, and c) produce the same results? Briefly explain.

They approximately produced the same results. But none of them were conclusive. But from performing all of them, I was able to figure out that ages in passenger class 1 could be normally distributed. The density plot gave me the best picture and the Shapiro-Wilk test gave me an approximate probability that it is normally distributed.

4. Birds

[8 points]

Data: *birds* in **openintro** package

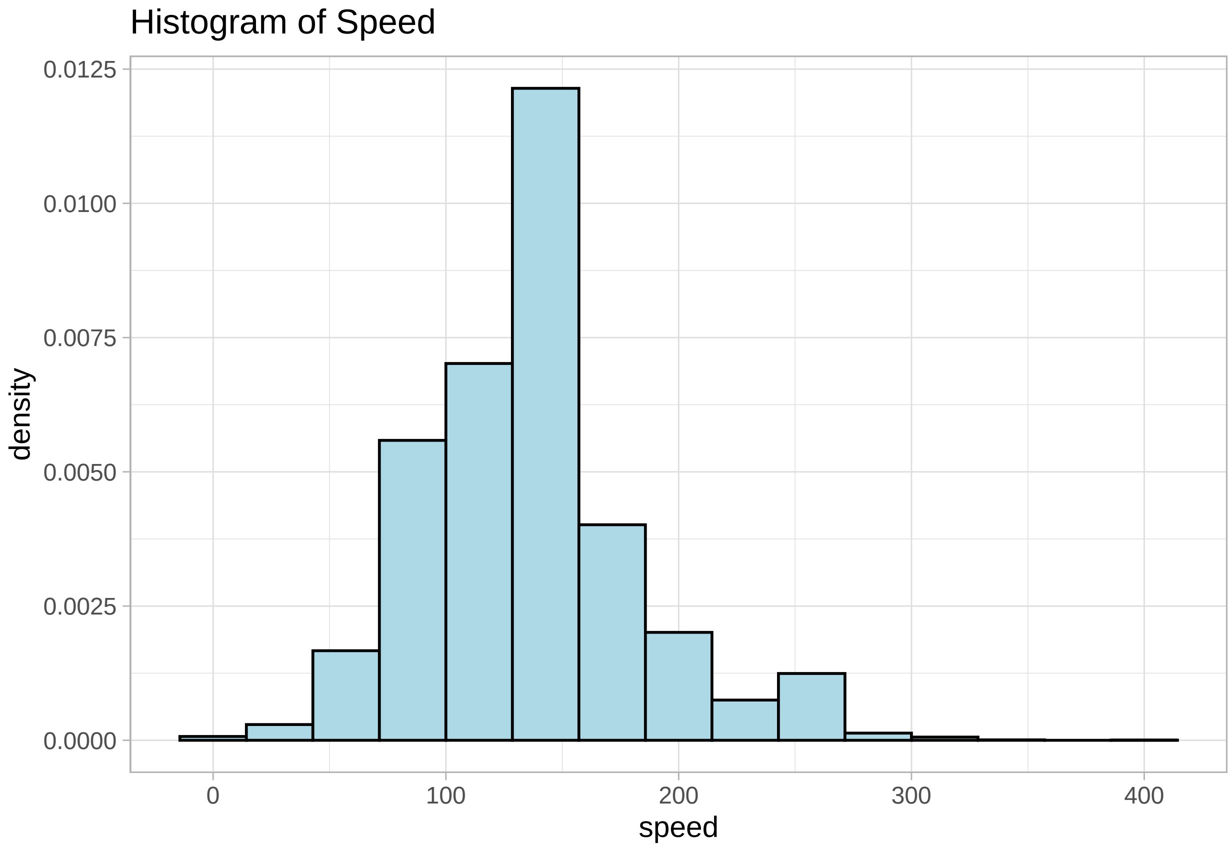
```r
library(openintro)
```

```
## Loading required package: airports

## Loading required package: cherryblossom

## Loading required package: usdata

##
## Attaching package: 'openintro'

## The following object is masked from 'package:fivethirtyeight':
##
##     drug_use
```

a) Use appropriate techniques to describe the distribution of the `speed` variable noting interesting features.

```r
birds %>% ggplot(aes(x=speed)) +
  geom_histogram(aes(y=..density..), bins=15, fill="lightblue", color="black") +
  theme_light() + ggtitle("Histogram of Speed")
```

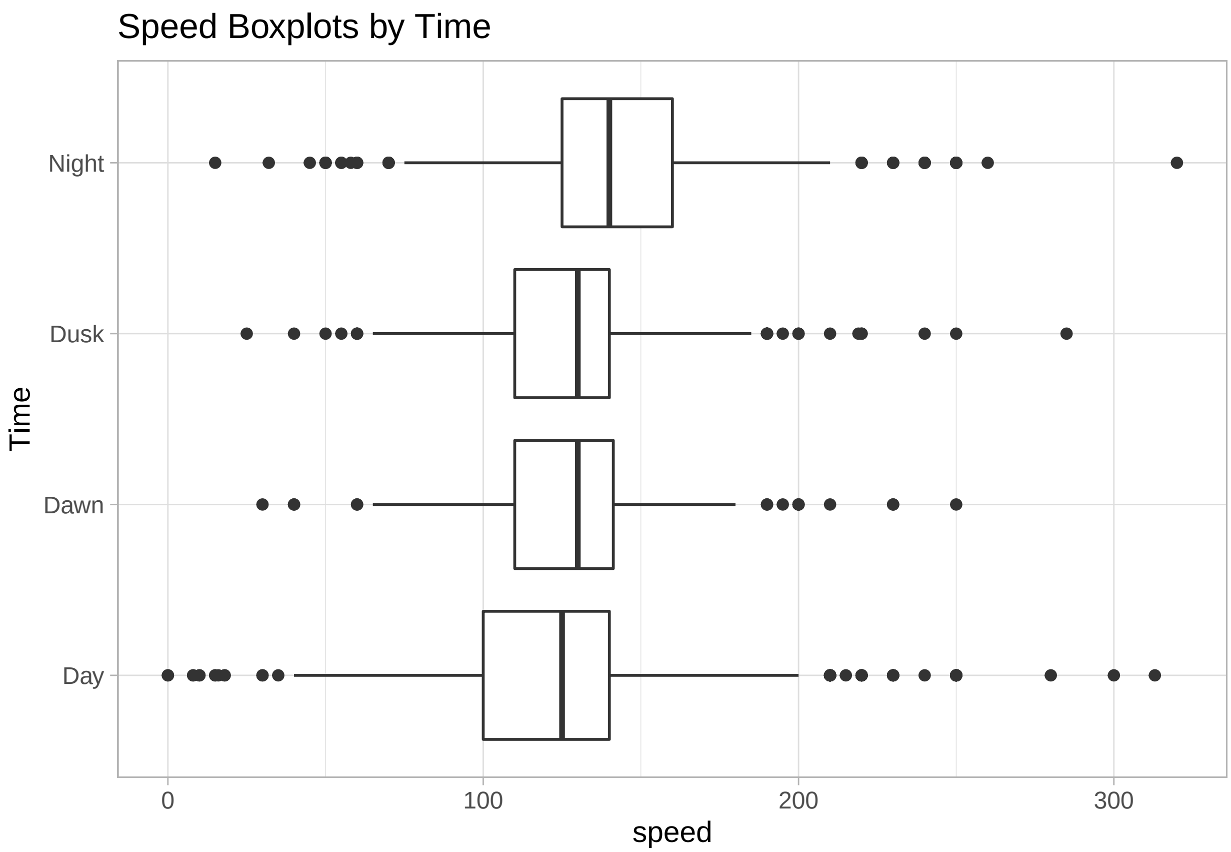
```
## Warning: Removed 7008 rows containing non-finite values (stat_bin).
```



There data seems right skewed, which mekaes sense because planes traveling very high speeds is more unusual than those that are slower. And majority of the planes seem have speeds between 80 and 150.

b) Create horizontal boxplots of `speed`, one for each level of `time_of_day`.

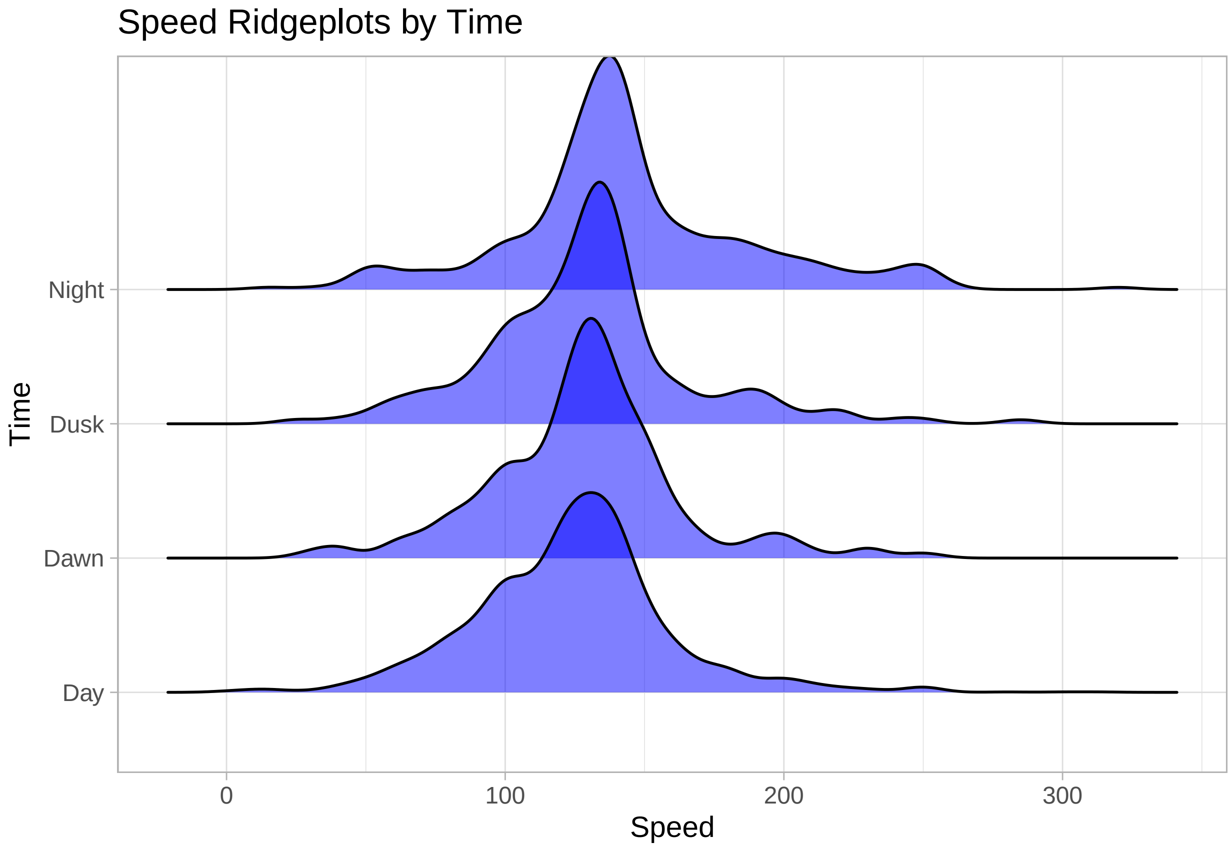
```
bp <- birds %>% drop_na() %>% ggplot(aes(x=reorder(time_of_day, speed, median), y=speed)) + geom_boxplo
  ggtitle("Speed Boxplots by Time") + xlab("Time") + theme_light()

bp
```



c) Create ridgeline plots for the same data as in b)

```
rp <- birds %>% drop_na() %>% ggplot(aes(y=reorder(time_of_day, speed, mean), x=speed)) + geom_density_
  ggtitle("Speed Ridgeplots by Time") + ylab("Time") + xlab("Speed") + theme_light()

rp
```

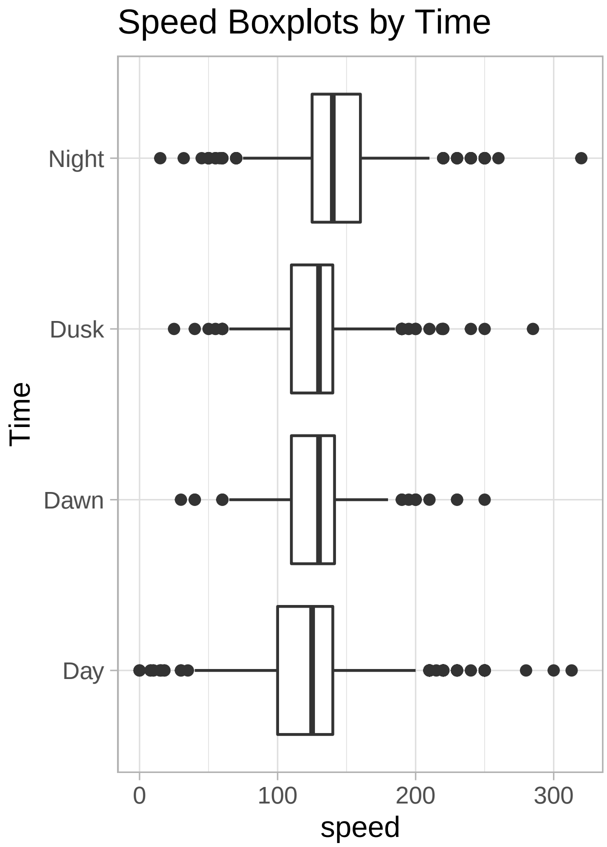
```
## Picking joint bandwidth of 7.01
```

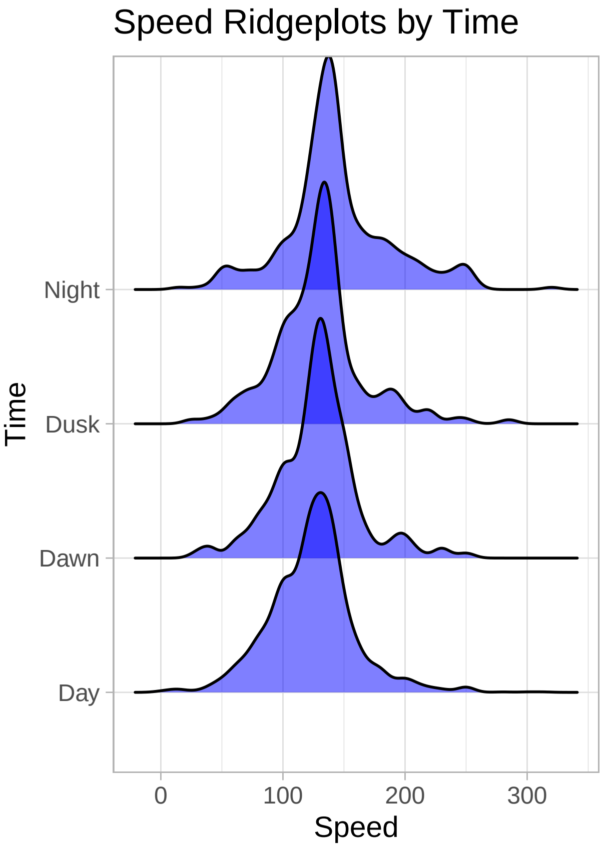


d) Compare the boxplot plots and the ridgeline plots.

```
grid.arrange(bp, rp, nrow=1)
```

```
## Picking joint bandwidth of 7.01
```


Speed Boxplots by Time


Speed Ridgeplots by Time

From the ridge plots, we see that the speeds are approximately normally distributed. But there is more modality during night, dusk and dawn than the day distribution. From the boxplot, we see that there are many outliers on both ends of speed for no matter the time of day. It is also easier to see that the speed is highest during night time. We can not glean this information as easily from the ridge plot. It is interesting to see that an average the flights at night are faster than those during the day.