Data Visualization Homework 3

Matt Isaac - A01515095

November 29, 2017

i

a. Load all required R packages to answer this question. Show your R code.

```
library(tidyr)
library(ggplot2)
library(DAAG)
library(lattice)
library(graphics)
library(ggthemes)
library(dplyr)

palette("default")
```

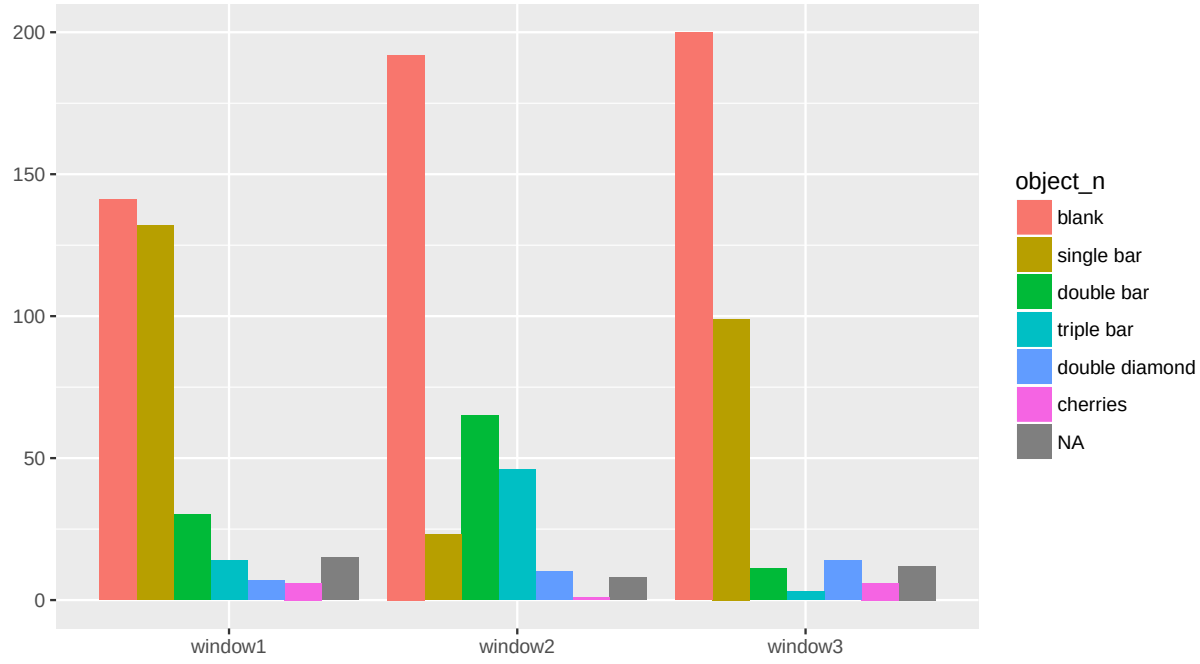
b. Process the data to get the data in the proper format for the graphs. Show R code and final data structure.

```
data(vlt)
vlt_long <- gather(vlt, window, object, window1:window3, factor_key = TRUE)
object_names <- c("blank", "single bar", "double bar", "triple bar", NA,
                  "double diamond", "cherries", "7")
vlt_long$object_n <- as.factor(object_names[vlt_long$object + 1])

vlt_long$object_n <- factor(vlt_long$object_n, levels = c("blank",
                                                          "single bar",
                                                          "double bar",
                                                          "triple bar",
                                                          "double diamond",
                                                          "cherries"))
# vlt_long is formatted for the ggplot package

head(vlt_long)
```

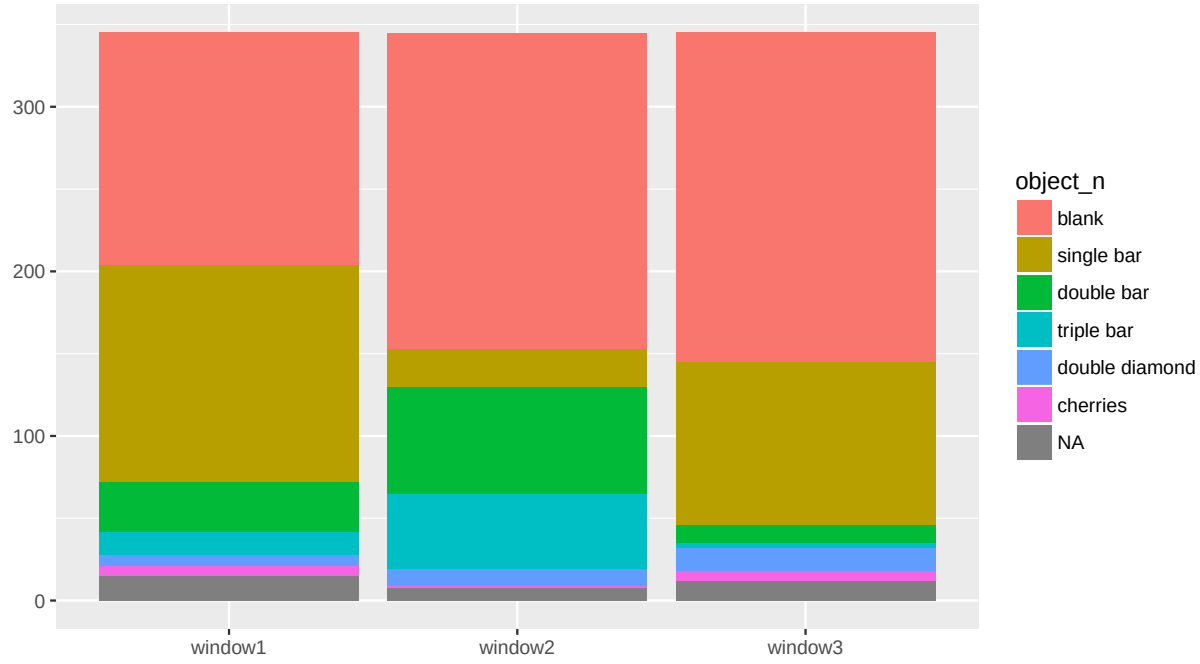
```
##   prize night  window object   object_n
## 1     0     1 window1      2 double bar
## 2     0     1 window1      0      blank
## 3     0     1 window1      0      blank
## 4     0     1 window1      2 double bar
## 5     0     1 window1      0      blank
## 6     0     1 window1      0      blank
```

c. Draw equally scaled bar charts to see if the distributions of frequencies are the same for each window. Describe important features. Try at least four di???erent layouts of your bar charts. No need to further re???ne all of these. Just choose the layout that best answers this question and re???ne this bar chart by adjusting scales, axis labels, title, etc.

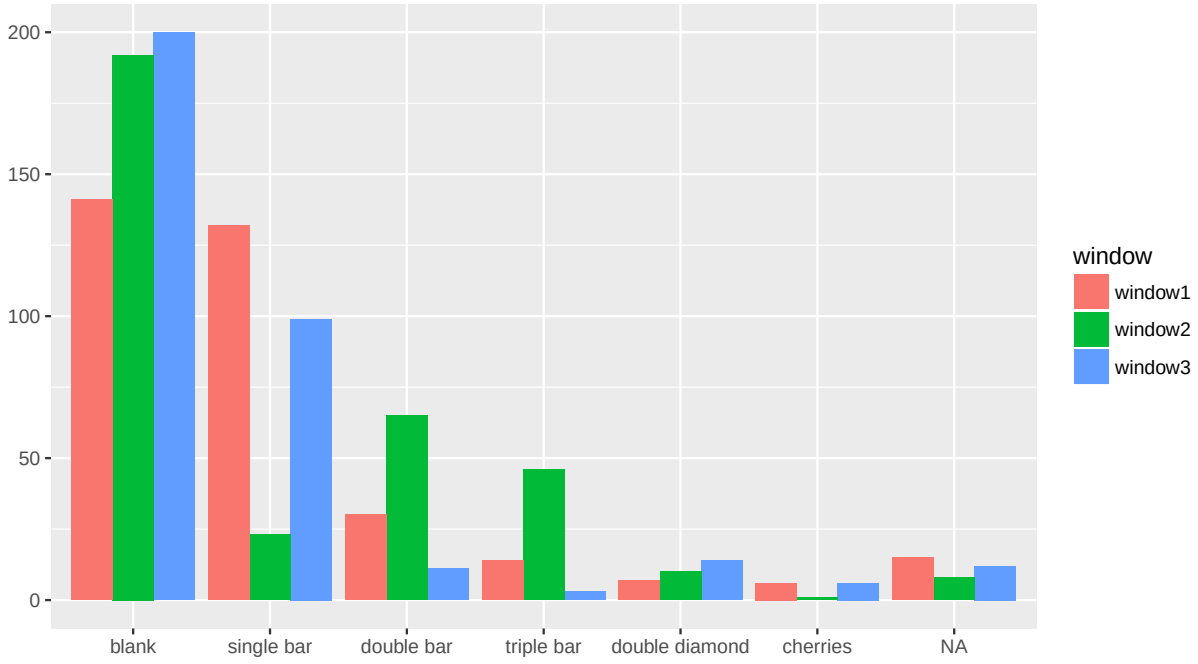
```
ggplot(vlt_long, aes(x = window)) +
  xlab("") +
  ylab("") +
  geom_bar(aes(fill = object_n), position = "dodge")
```



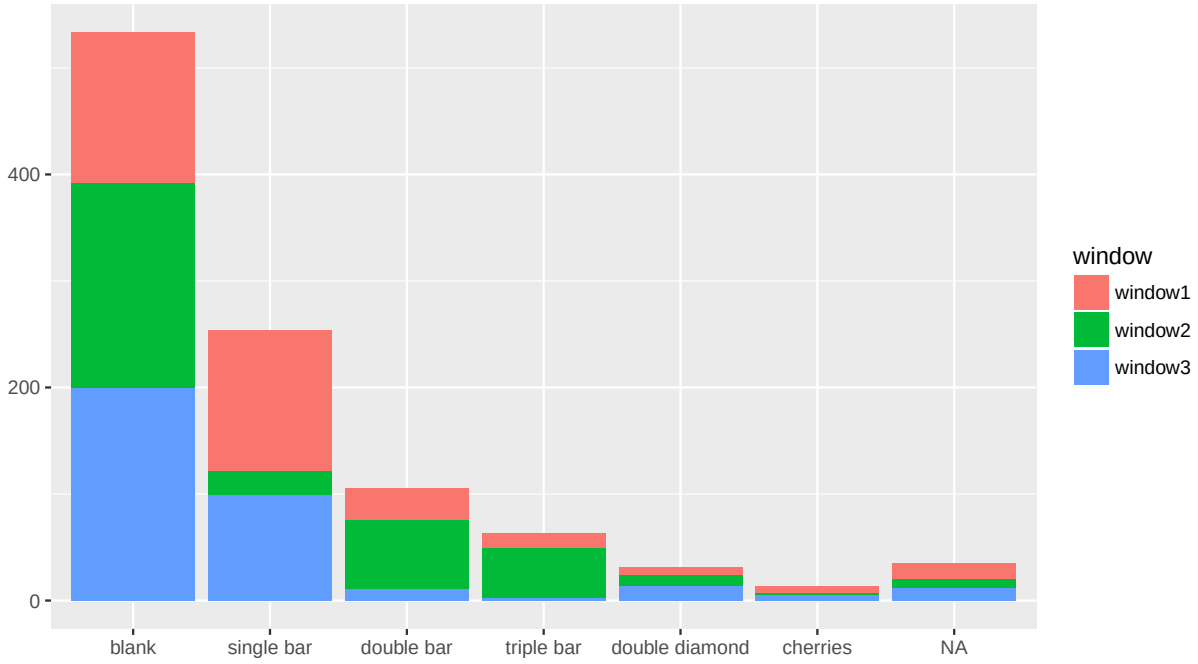

```
ggplot(vlt_long, aes(x = window)) +
  xlab("") +
  ylab("") +
  geom_bar(aes(fill = object_n), position = "stack")
```




```
ggplot(vlt_long, aes(x = object_n)) +
  xlab("") +
  ylab("") +
  geom_bar(aes(fill = window), position = "dodge")
```



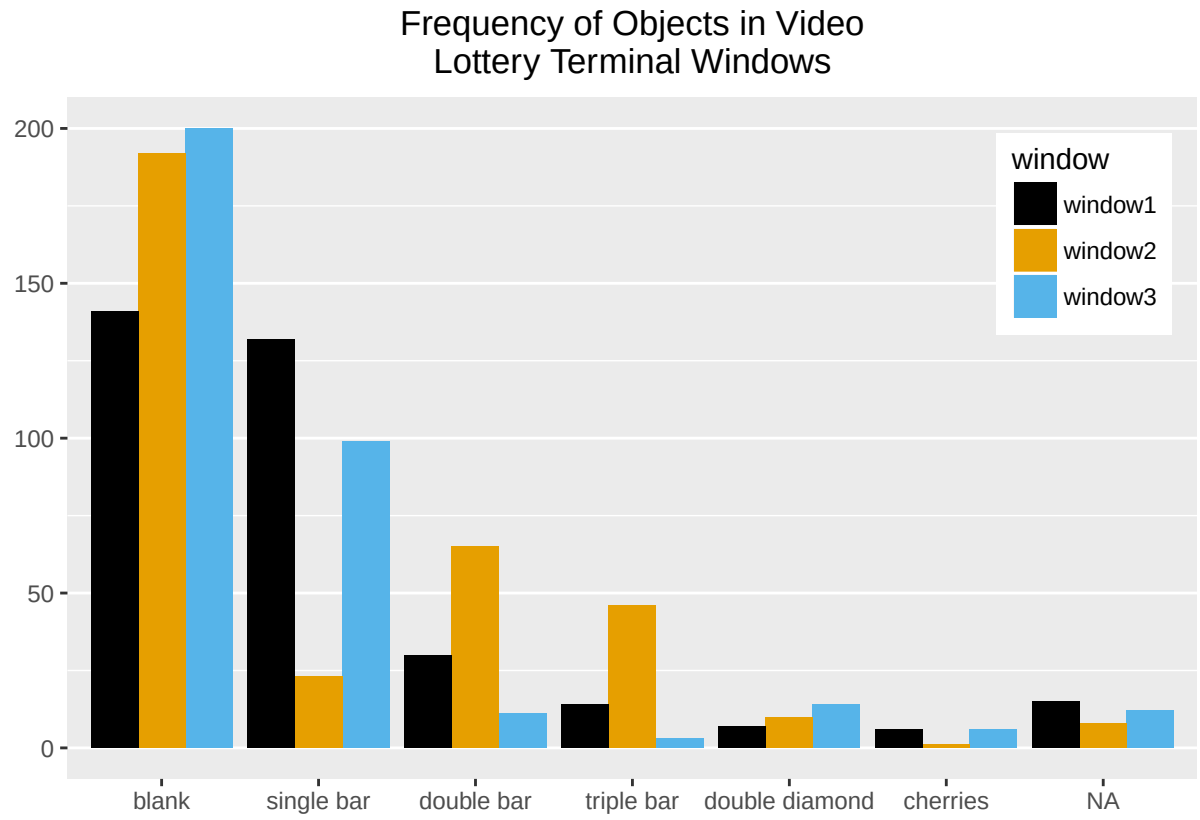

```
ggplot(vlt_long, aes(x = object_n)) +
  xlab("") +
  ylab("") +
  geom_bar(aes(fill = window), position = "stack")
```



Above are four possible bar chart layouts. The side-by-side (cluster) bar charts are better choices than the stacked bar charts, as the stacked bar charts make for difficult comparisons without a common top baseline and more than two categories. Of the two side-by-side bar charts, I think that the better choice is to group on the object shown. This way, one can easily compare the frequence of single bars (for example) for the

three windows because the three bars are right next to each other.

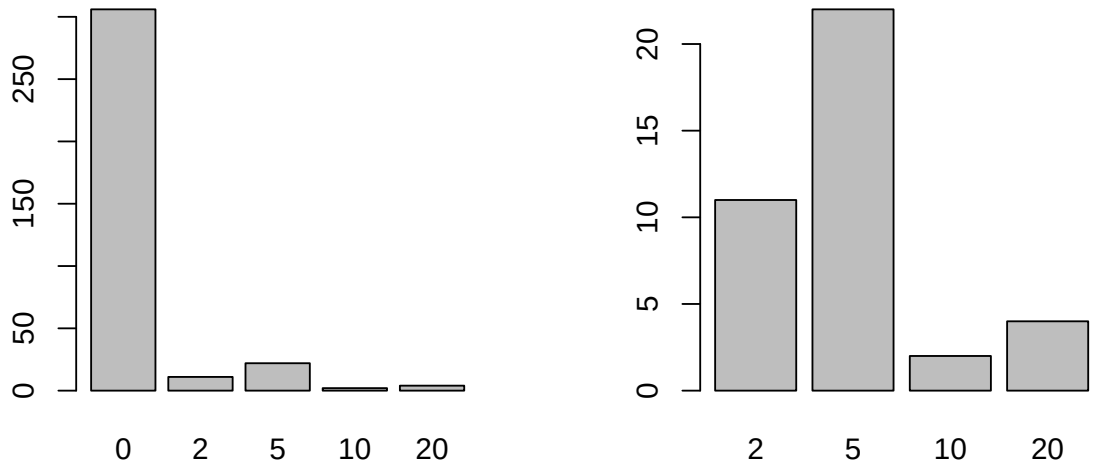
```
ggplot(vlt_long, aes(x = object_n)) +
  xlab("") +
  ylab("") +
  scale_y_continuous(breaks = seq(0, 200, by = 50))+
  ggtitle("Frequency of Objects in Video\nLottery Terminal Windows") +
  theme(plot.title = element_text(hjust = 0.5), legend.position = c(0.9, 0.8), panel.grid.major.x = ele
  scale_fill_colorblind() +
  geom_bar(aes(fill = window), position = "dodge")
```



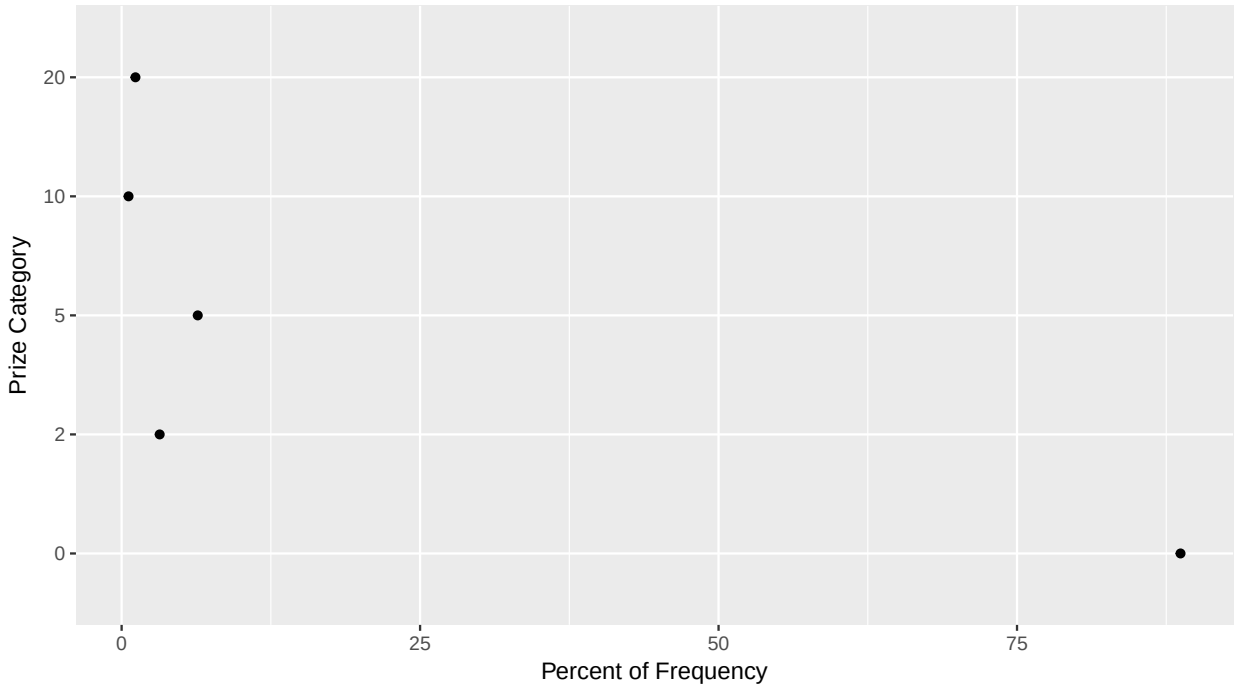
I have refined this bar chart (above) adding a title, changing the color scheme, and repositioning the legend. These plots are very informative, especially if we are looking to see if the distributions of object appearance are the same for each window. A quick glance will suggest otherwise. Upon closer inspection, we see that the distributions are quite differnt. It is true that blanks are the mode for each window; however, there is a difference of about 50 between window 1 and windows 2 and 3. This difference appears even more vast when we add the second object, single bar, to the comparison. In window 1, blanks are just barely more frequent than single bar. This is very differnt than windows 2 and 3 where the blanks are about 8 times and 2 times more frequent than the single bars. The differences in the frequencies of blanks and single bars is the most notable feature of these distributions. Differences between the other four objects exist, but are less extreme. For example, window 2 had over 60 double bars appear, while window three had about 12. Window two had almost 50 triple bars, while window 4 had about 5. Inference could be performed to be sure, but it appears to me that these differences are more than could be explained by chance.

d. Treat prize as a categorical variable with 5 outcomes. Draw at least three different graphs that show the distribution of prizes. Refine one.

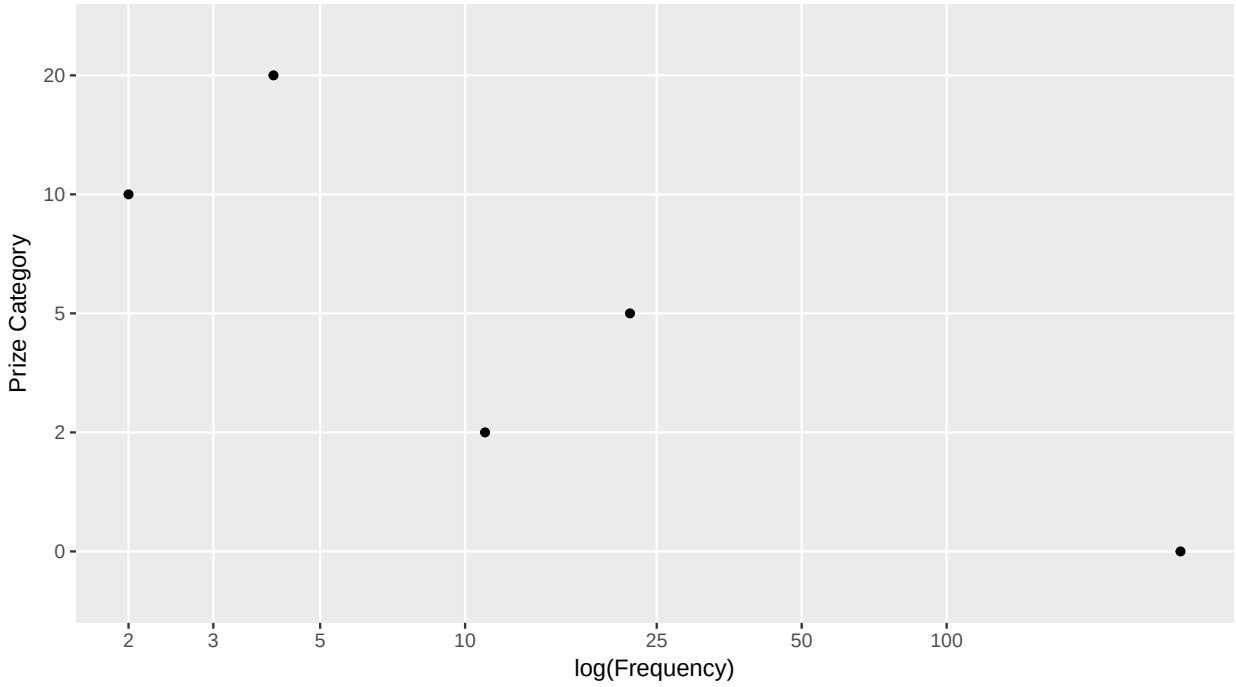
```r
par(mfrow = c(1, 2))
barplot(summary(as.factor(vlt$prize)))
barplot(summary(as.factor(vlt$prize[which(vlt$prize == 2 | vlt$prize == 5
                                          | vlt$prize == 10 | vlt$prize == 20)])))
```




```r
vlt_long$prize <- as.factor(vlt_long$prize)
vlt_long$night <- as.factor(vlt_long$night)

pn_table <- table(vlt_long$prize)
pn <- margin.table(pn_table, 1)
pn <- pn / 3

pn_df <-  data.frame(pn)
pn_df$prize <- as.factor(pn_df$Var1)
pn_df$percFreq <- (pn_df$Freq / nrow(vlt)) * 100
pn_df <- dplyr::select(pn_df, prize, percFreq, Freq)

ggplot(pn_df, aes(x = percFreq, y = prize )) +
  ylab("Prize Category") +
  xlab("Percent of Frequency") +
  geom_point()
```



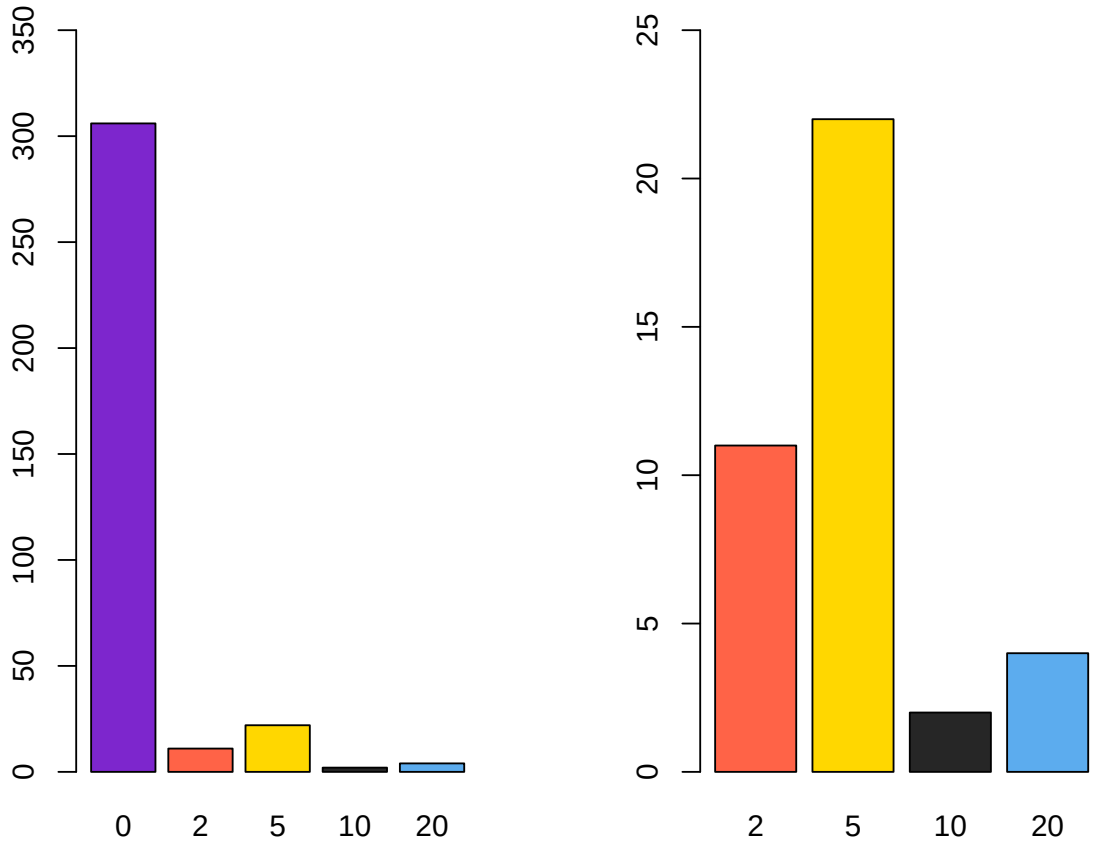

```
ggplot(pn_df, aes(x = Freq, y = prize )) +
  ylab("Prize Category") +
  xlab("log(Frequency)") +
  scale_x_log10(breaks = c(1, 2, 3, 5, 10, 25, 50, 100)) +
  theme(panel.grid.minor.x = element_blank()) +
  geom_point()
```



Above are three figures that describe the distribution of prizes - one figure consisting of two barplots, and two dotplots. I think that the figure with two bar plots is particularly effective at visualizing the distribution

of these data, and this is the plot I have chosen to refine.
In this figure, the first plot displays all the data. Because there are so many "prizes" awarded for $0, we don't see much detail in the prize categories that were awarded less frequently. The second plot displays a sort of "zoomed in" version of the first plot that shows all the prize categories other than $0. This allows us to see a lot more detail in the distribution of the the last four categories. This is admittedly a risky thing to do, as some people may not realize that the scale has been changed. However, I have taken measures to mitigate this risk. On the optimzed version of this plot, I have included a note explicitly stating that the axis scales have been changed. I have also kept the same labeling and color coding between both graphs. I think that between these two additions and the included graph description that this is an effective and meaningful plot. The refined plot is shown below.

```
par(mfrow = c(1, 2))
barplot(summary(as.factor(vlt$prize)), col = c("purple3", "tomato", "gold", "gray15", "steelblue2"), yli
barplot(summary(as.factor(vlt$prize[which(vlt$prize == 2 |
                                          vlt$prize == 5 |
                                          vlt$prize == 10 |
                                          vlt$prize == 20)])),
        col = c("tomato", "gold", "gray15", "steelblue2"),
        ylim = c(0, 25))
mtext("**Note change in y-axis scale**", side = 1, line = -2,
      outer = TRUE)
```

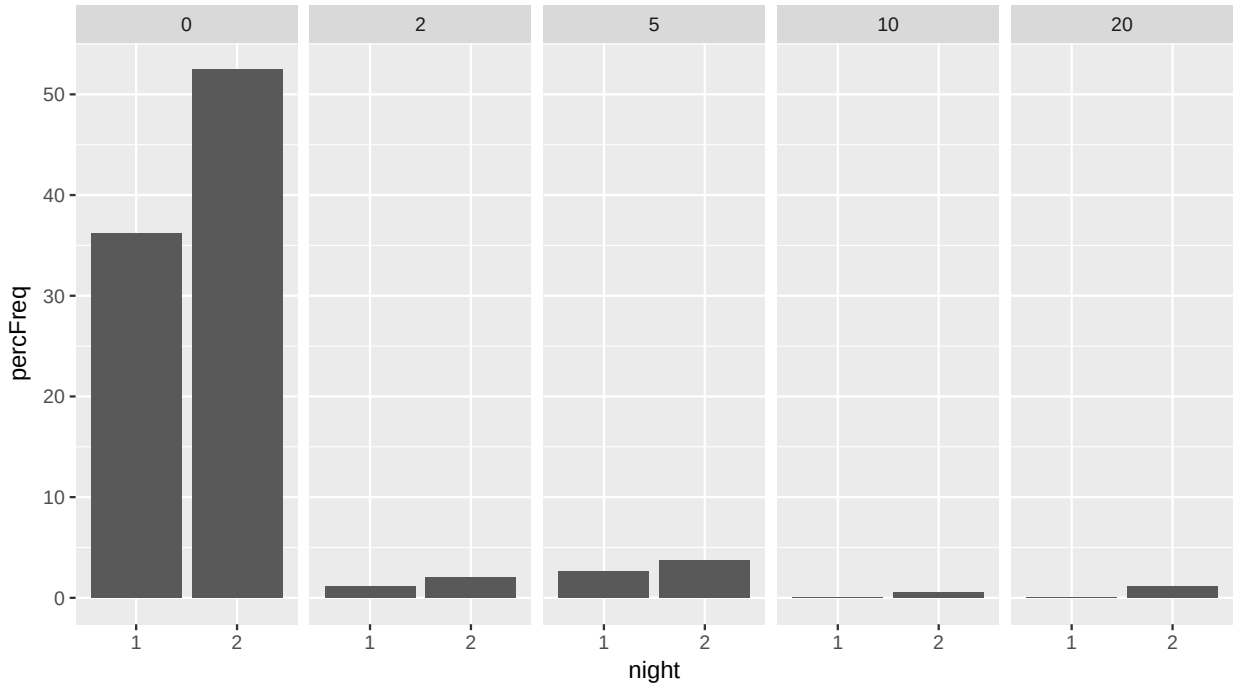


Explanation for figure above: Category '0' has been omitted from second plot so the distribution of the other categories can be seen with greater detail.

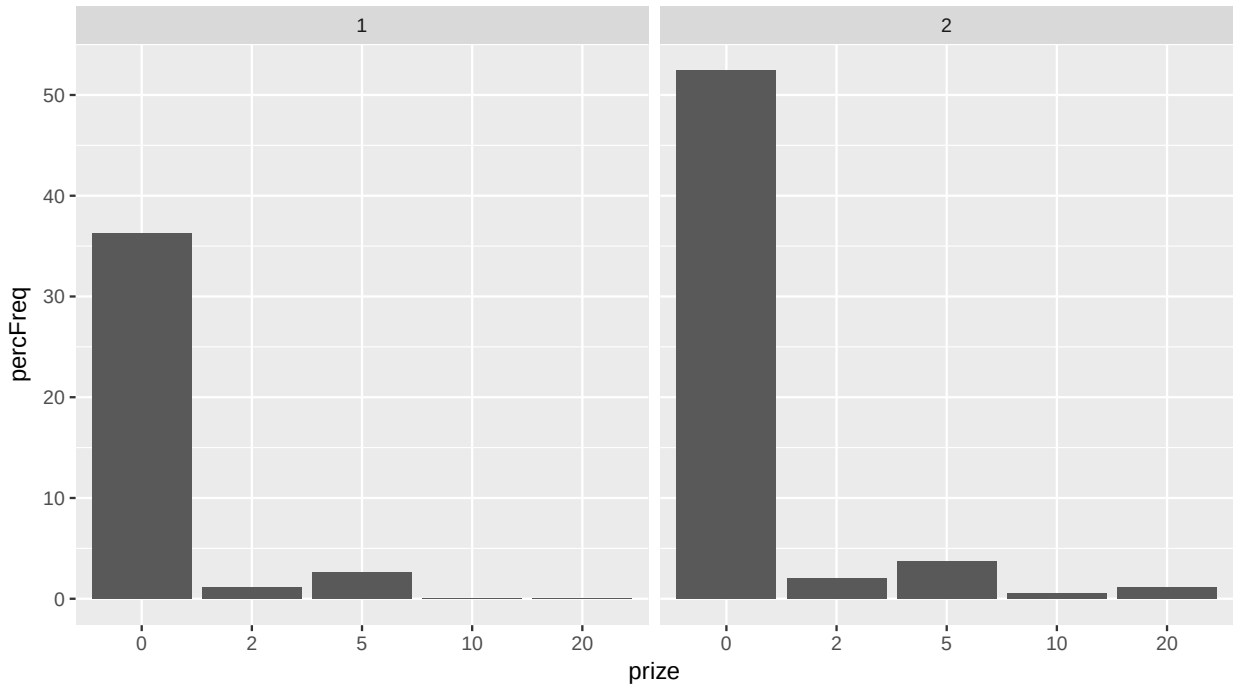
e. Draw at least three graphs of your choice to answer the question whether the variables prize and night are independent.

```
vlt_long$prize <- as.factor(vlt_long$prize)
vlt_long$night <- as.factor(vlt_long$night)

pn_table <- table(vlt_long$prize, vlt_long$night)
pn <- margin.table(pn_table, 1:2)
pn <- pn / 3

pn_df <-  data.frame(pn)
pn_df$prize <- as.factor(pn_df$Var1)
pn_df$night <- as.factor(pn_df$Var2)
pn_df$percFreq <- (pn_df$Freq / nrow(vlt)) * 100
pn_df <- dplyr::select(pn_df, night, prize, percFreq)

ggplot(data = pn_df, aes(x = night, y = percFreq)) +
  geom_bar(stat = "identity") +
```

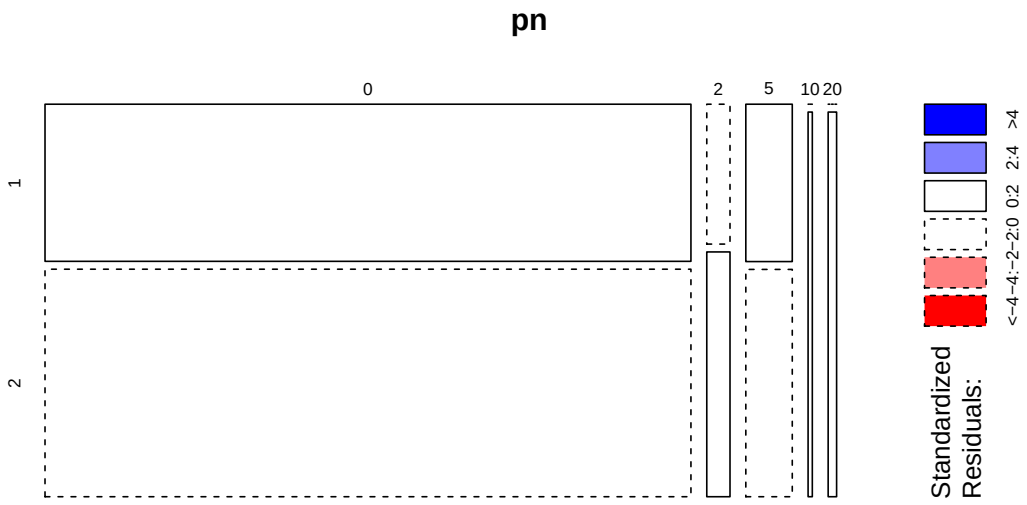
```
  facet_grid(.~ prize)
```




```
ggplot(data = pn_df, aes(x = prize, y = percFreq)) +
  geom_bar(stat = "identity") +
  facet_grid(.~ night)
```

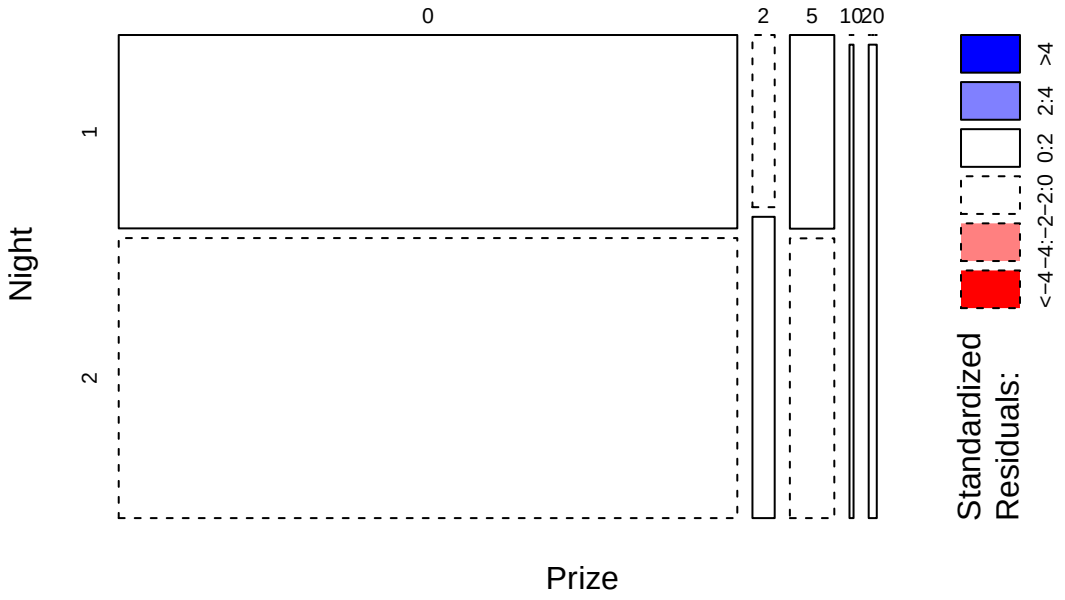


```
mosaicplot(pn, shade = TRUE)
```



I think that the mosaic plot best displays whether night and prize are independent. Below is the refined version of the plot and my explanation and conclusions. Not much was needed other than label changes.

```
mosaicplot(pn, shade = TRUE, main = "", xlab = "Prize", ylab = "Night")
```



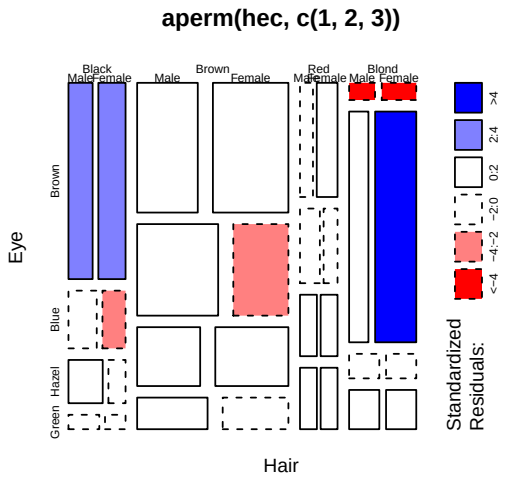
Displaying the standardized residuals gives us information about whether a particular cell has more or less observations than we would expect if the variables are independent. If there are more observations than we'd expect, the cell will be colored blue. If there are less observations than we'd expect, the cell will be colored red. In the mosaic plot of prize by night, none of the cells are colored red or blue. All the cells are white - some are drawn with a dashed line, some are drawn with a solid line. This indicates that there are somewhere between two less and zero less (for dashed lines) or zero to two more (for solid lines) than we'd expect. In this case, we can see that there might be slightly more or less observations than we'd expect if prize and night are independent.However, I don't think that it is extreme enough to say that prize and night are independent. From this plot, I would conclude that prize and night are indeed independent.

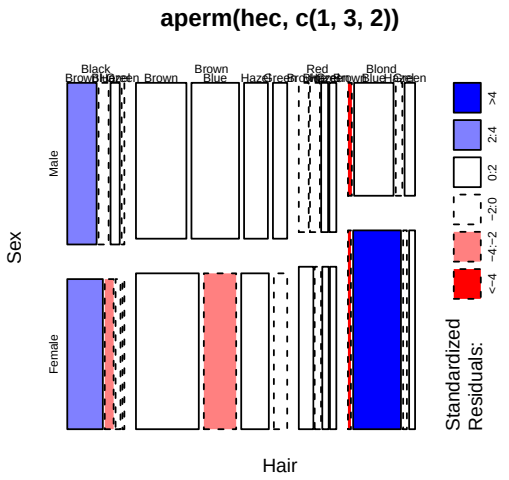
ii

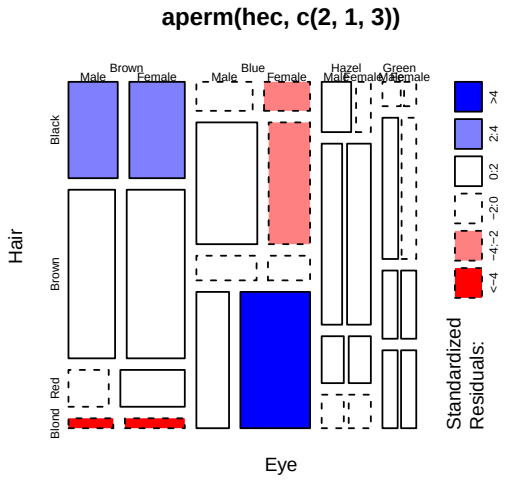
```
hec <- HairEyeColor #[h]air [e]ye [c]olor
```

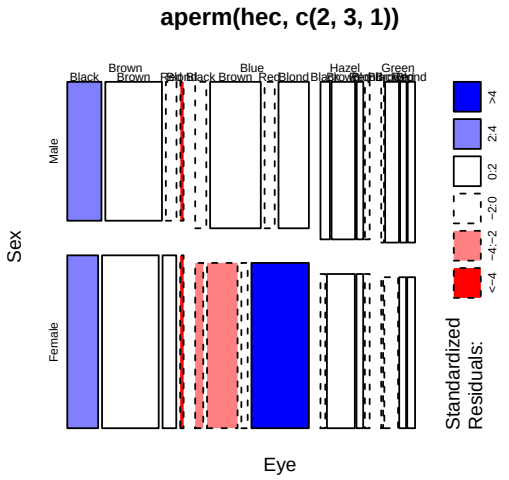
a. Create six different mosaic plots that show all possible layouts for the three variables using base R. Also show the standardized residuals.

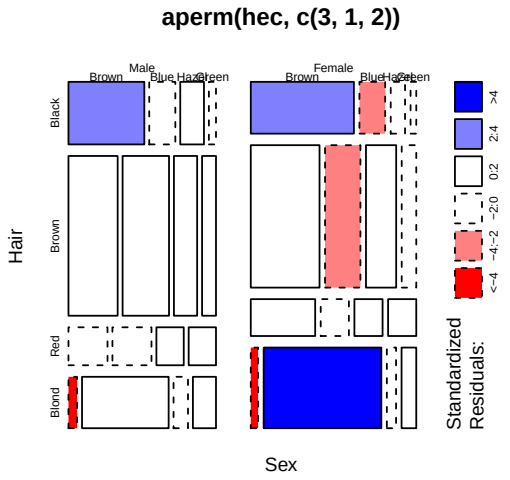
```
par(mfrow = c(3, 2))
mosaicplot(aperm(hec, c(1, 2, 3)), shade = TRUE)
mosaicplot(aperm(hec, c(1, 3, 2)), shade = TRUE)
mosaicplot(aperm(hec, c(2, 1, 3)), shade = TRUE)
mosaicplot(aperm(hec, c(2, 3, 1)), shade = TRUE)
mosaicplot(aperm(hec, c(3, 1, 2)), shade = TRUE)
mosaicplot(aperm(hec, c(3, 2, 1)), shade = TRUE)
```

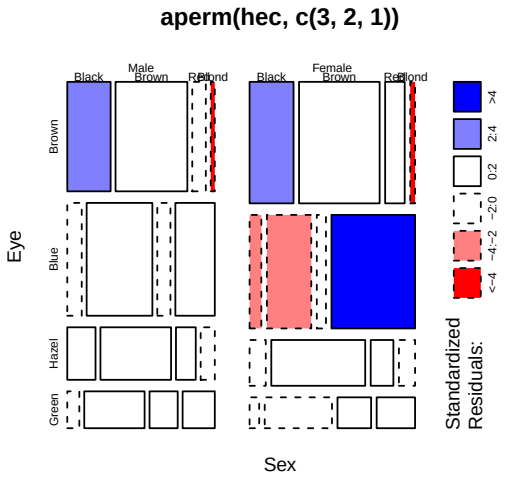

aperm(hec, c(1, 2, 3))
Eye
Hair
Standardized Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4


aperm(hec, c(1, 3, 2))
Sex
Hair
Standardized Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4


aperm(hec, c(2, 1, 3))
Hair
Eye
Standardized Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4


aperm(hec, c(2, 3, 1))
Sex
Eye
Standardized Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4

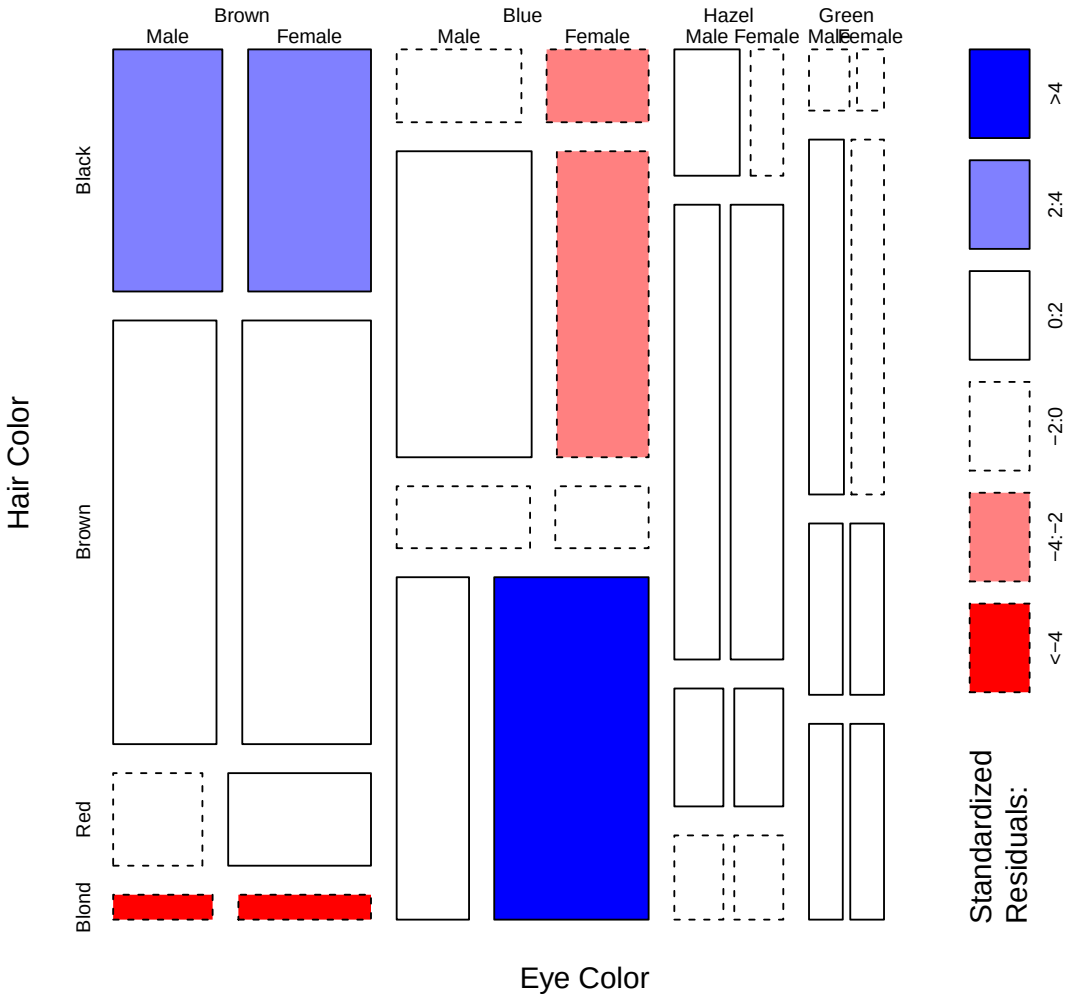

aperm(hec, c(3, 1, 2))
Hair
Sex
Standardized Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4


aperm(hec, c(3, 2, 1))
Eye
Sex
Standardized Residuals:
>4
2:4
0:2
-2:0
-4:-2
<-4

b.

Optimize mosaic plot that best displays the 3 pairs and 1 unique combination.

```
mosaicplot(aperm(hec, c(2, 1, 3)), shade = TRUE, main = "",
           xlab = "Eye Color",
           ylab = "Hair Color")
```



c. Describe and explain your mosaic plot from (b) above. What can be seen? How can we best interpret the three pairs and the unique combination? Isn???t there an important lurking variable that is missing from this data set, but that would help to even better explain the observed pattern? Which variable is this ??? and how could it be used to explain the pattern?

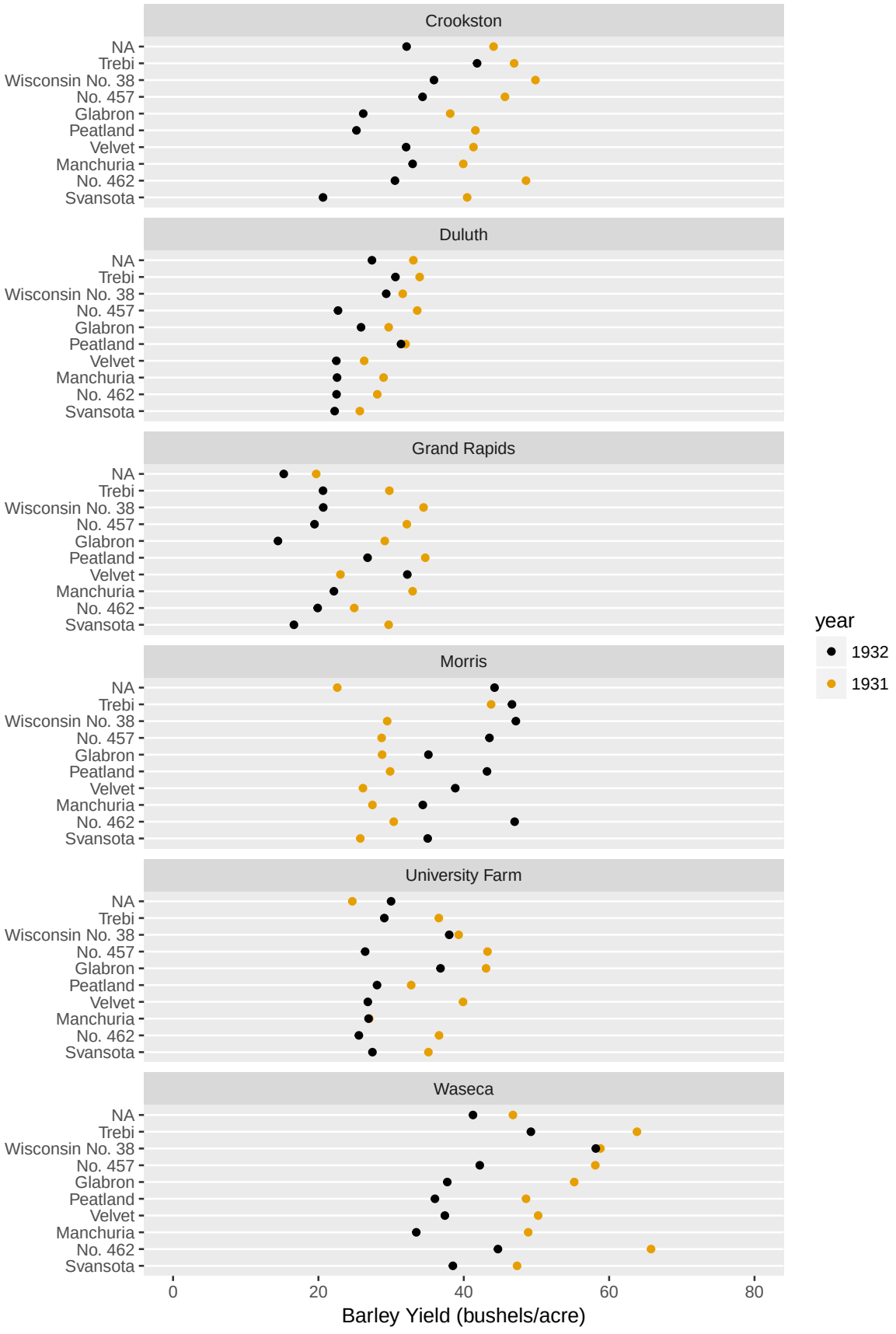
The mosaic plot above shows the relationship between three variables - gender, hair color, and eye color. Each box represents a unique combination of the three variables. For example, the top left box represents the males in the data set with black hair, brown eyes. The color of the box is also significant. The color gives us information about whether a particular cell has more or less observations than we would expect if the variables are independent. If there are more observations than we'd expect, the cell will be colored blue. If

there are less observations than we'd expect, the cell will be colored red. We can see from the plot above, since some of the boxes are colored red or blue that hair color, eye color, and gender are not all mutually independent. There are three pairs of colored squares. The two light blue squares in the top left indicate that there are more males and females with black hair and brown eyes than we'd expect if the variables were independent. There are two red boxes at the bottom left corner of the plot which are related to the first pair of blue boxes. The red boxes tell us that there are less individuals with blonde hair and brown eyes than we would expect of the variables are independent. Next, we see that in the top center of the plot there are two light red boxes. These boxes tell us that there are less females with black hair and blue eyes and less females with brown hair and blue eyes than we'd expect. Lastly, the large blue rectangle shows that there are more females with blonde hair and blue eyes than we would expect if the variables are independent.
There is a descriptor for all of these people that, if added, could help to explain why we see fewer people with blonde hair and brown eyes and more people with blonde hair and blue eyes than we would initially expect. This added descriptor (or variable) is ethnicity (could also be explained with skin color). People that are of Latino or African ethnicity are more likely to have black or brown hair and brown eyes. Likewise, people of European origin ("white") are more likely to have blonde hair and blue eyes. This would explain the pairs of colored squares we observe in this mosaic plot above.

iii

a. Reconstruct and optimize the barley data dot plot.

```
data(barley)


barley$variety <- factor(barley$variety,
                         levels = c("Svansota", "No. 462", "Manchuria", "No.
                                    475", "Velvet", "Peatland", "Glabron",
                                    "No. 457", "Wisconsin No. 38", "Trebi"))
barley$site <- factor(barley$site,
                      levels = c("Crookston", "Duluth", "Grand Rapids", "Morris",
                                 "University Farm", "Waseca"))

ggplot(barley, aes(x = yield, y = variety,  color = year)) +
  geom_point() +
  xlab("Barley Yield (bushels/acre)") +
  ylab("") +
  scale_colour_colorblind() +
  xlim(0,80) +
  theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank()) +
  facet_wrap(~ site, ncol = 1)
```


Crookston
Duluth
Grand Rapids
Morris
University Farm
Waseca
NA
Trebi
Wisconsin No. 38
No. 457
Glabron
Peatland
Velvet
Manchuria
No. 462
Svansota
0
20
40
60
80
Barley Yield (bushels/acre)
year
1932
1931

iv

a. Load all packages required to answer this question

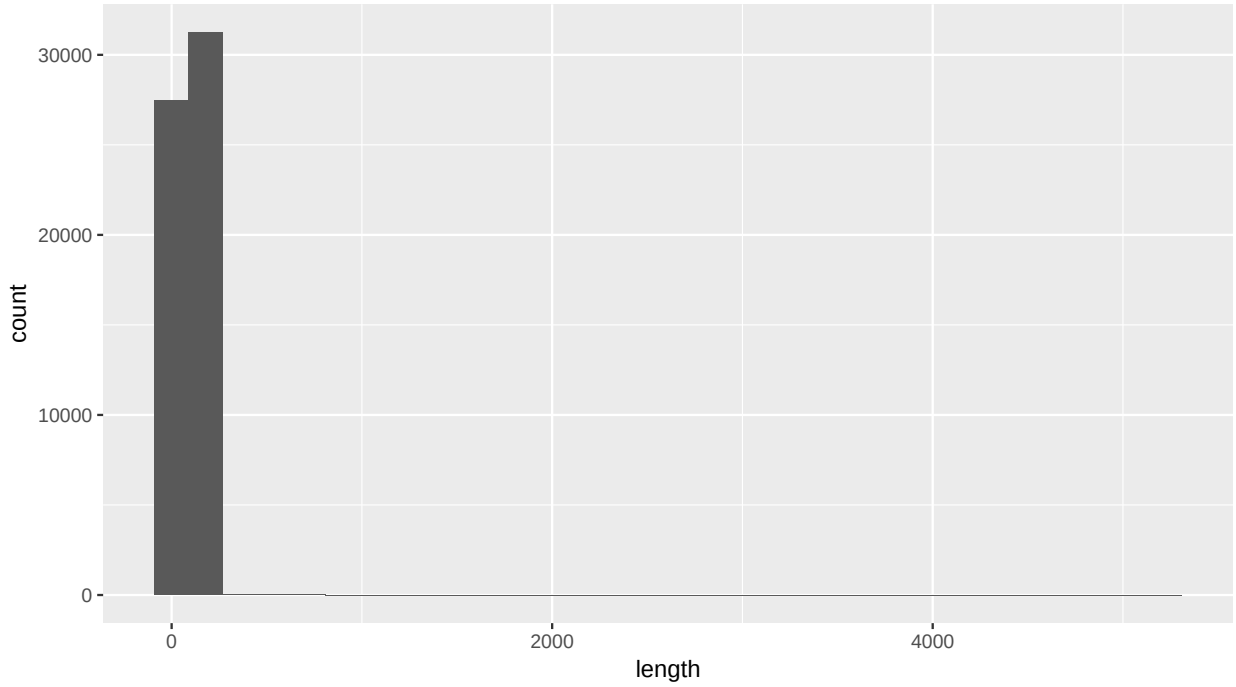
```
library(ggplot2movies)
library(ggplot2)
library(dplyr)
library(vioplot)
library(lvplot)
data(movies)

n_movies <-  nrow(movies)
```

Contrary to what is listed on the help page (28819) There are actually 58788 movies included in this data set.

b. Create default histogram via ggplot2. Do you believe the information in the histogram?

```
ggplot(movies, aes(x = length)) +
  geom_histogram()
```

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

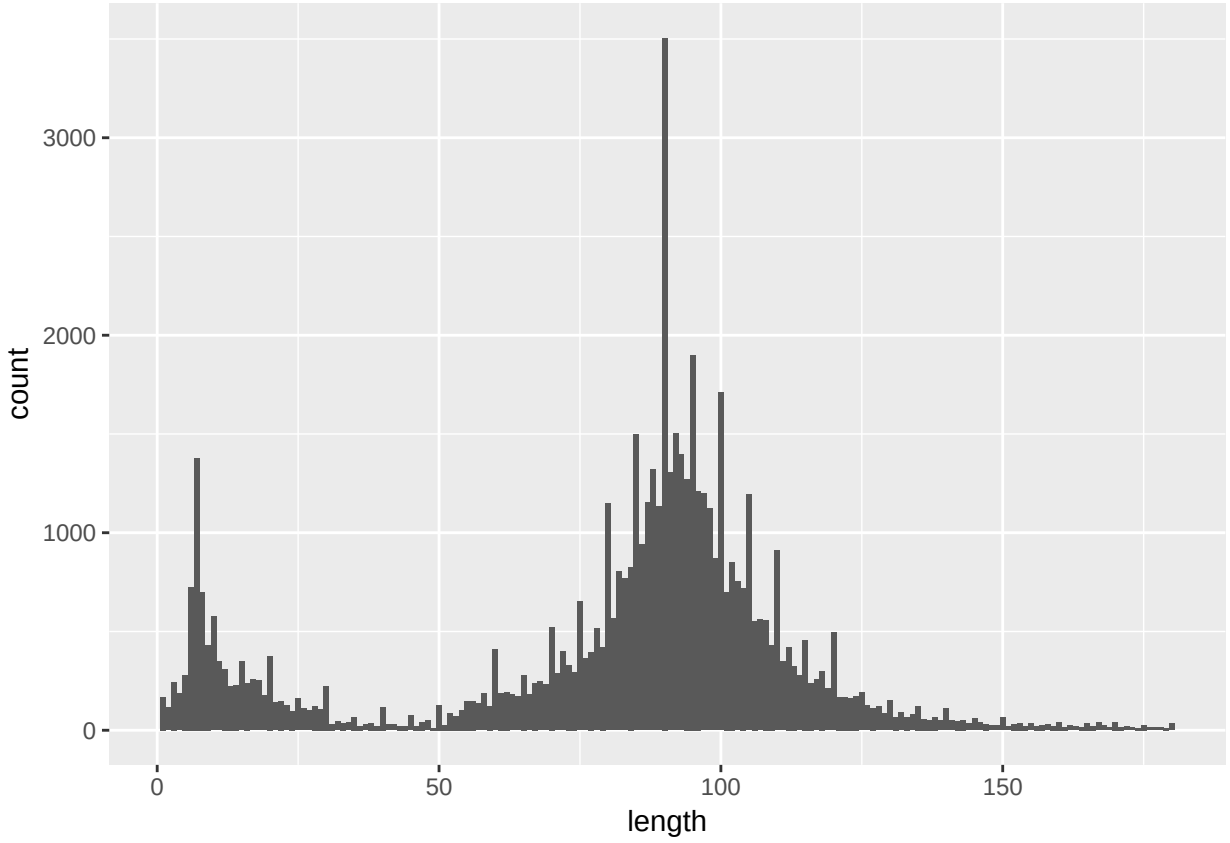



```
max_len <- max(movies$length)
```

Its hard to see anything significant from this histogram. It appears to be a uni-modal distribution, without much spread. However, when we look at the scale, the grid lines are every 1000 minutes. My first thought is that there must be an outlier to make our scale so wild. Sure enough, there is a movie that is 5220 minutes long. The default histogram is misleading because the actual structure in the main body of our data is obscured by the unnecessarily large breaks on the x-axis scale. In order to find out what the distribution really looks like, I would have to plot only the main body of the data from 0 to 300 or 400 minutes and make the bin width smaller. Only then will we begin to understand these data.

c. Create a histogram via ggplot2 of those movies that are 180 min (3 hours) or less in length. Use a bin width of 1 min, centered at 0 min. Describe and interpret this histogram.

```
lt180 <- filter(movies, length<=180)

lt180_bw1 <- ggplot(lt180, aes(x = length)) +
  geom_histogram(binwidth = 1, center = 0)
lt180_bw1
```

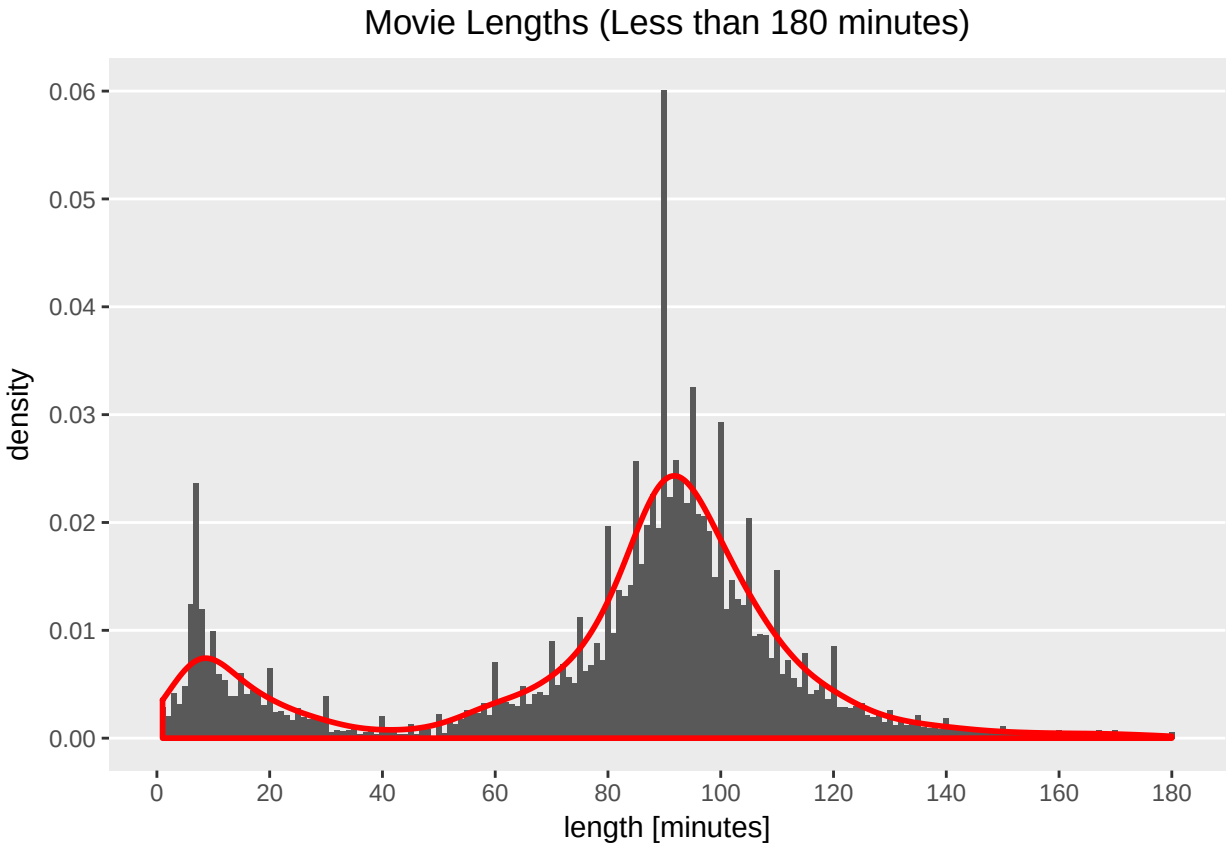


This histogram shows that our data are bi-modal - one mode is around 10 minutes, and the other is around 90 minutes. We have relatively few movies with lengths longer than 150 minutes, and obviously we have no movies with negative lengths. There is a strange pattern in these data; we observe a local mode every 5 minutes, and a very extreme mode at 90 minutes. Thinking about the "normal" lengths of movies and human nature in recording these data, this phenomenon is most likely due to some human-imposed rounding to a "nice" number, and that it does not necessarily reflect the true lengths of the movies. The large mode at 90 minutes (1.5 hours) reflects the fact that movies that are an hour and a half long (or close to it) are quite common.

d. Overlay a density plot on your previous histogram from (c)

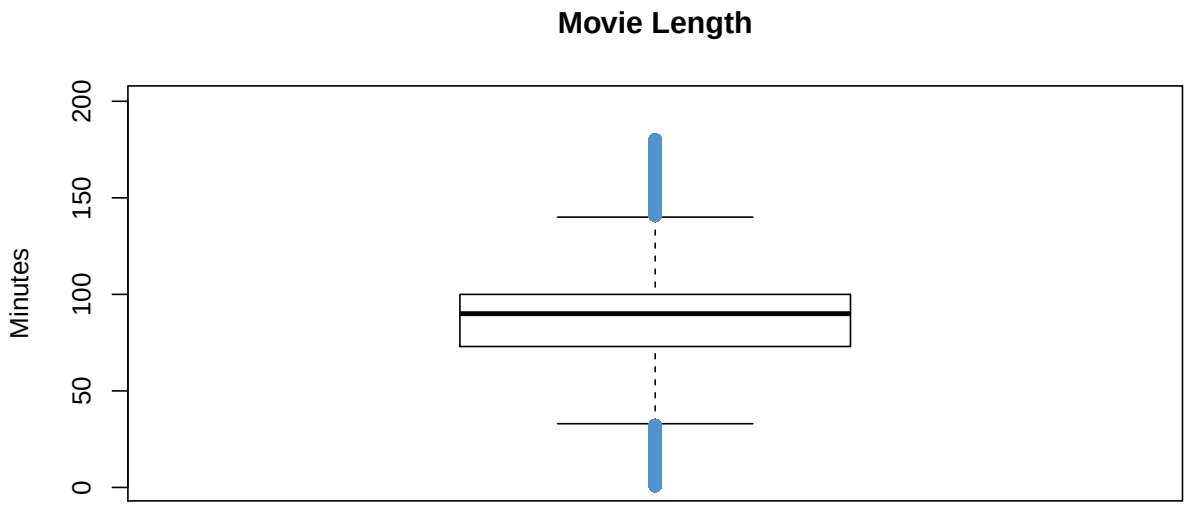
```
ggplot(lt180, aes(x = length)) +
  xlim(0, 180) +
  xlab("length [minutes]") +
  ylab("density") +
  scale_x_continuous(breaks = seq(0, 180, 20)) +
  scale_y_continuous(breaks = seq(0, 0.06, 0.01)) +
  geom_histogram(aes(y = ..density..), binwidth = 1, center = 0) +
  ggtitle("Movie Lengths (Less than 180 minutes)") +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank(),
```

```
        panel.grid.minor.y = element_blank()) +
  geom_density(bw = 5, col = "red", size = 1)
```

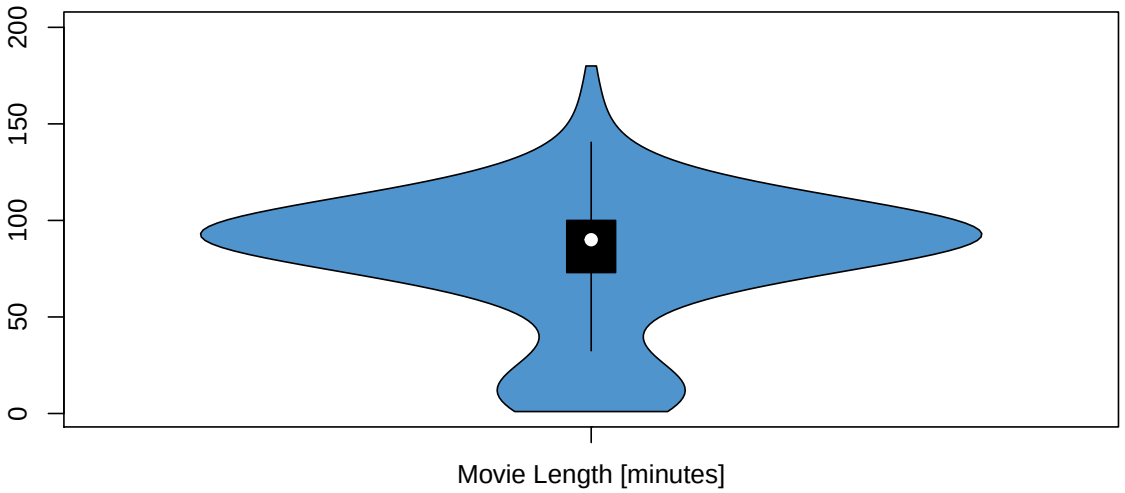


e. Create a regular box plot, violin plot, and letter value boxplot of those movies that are 180 min (3 hours) or less in length. Optimize each of these. Choose an R package of your choice for each of these three graphs. Include your ???nal ???gures and your R code. Does any of these three graphs ???match???your previous histogram, i.e., lead to a similar interpretation of the data? Explain your answer.

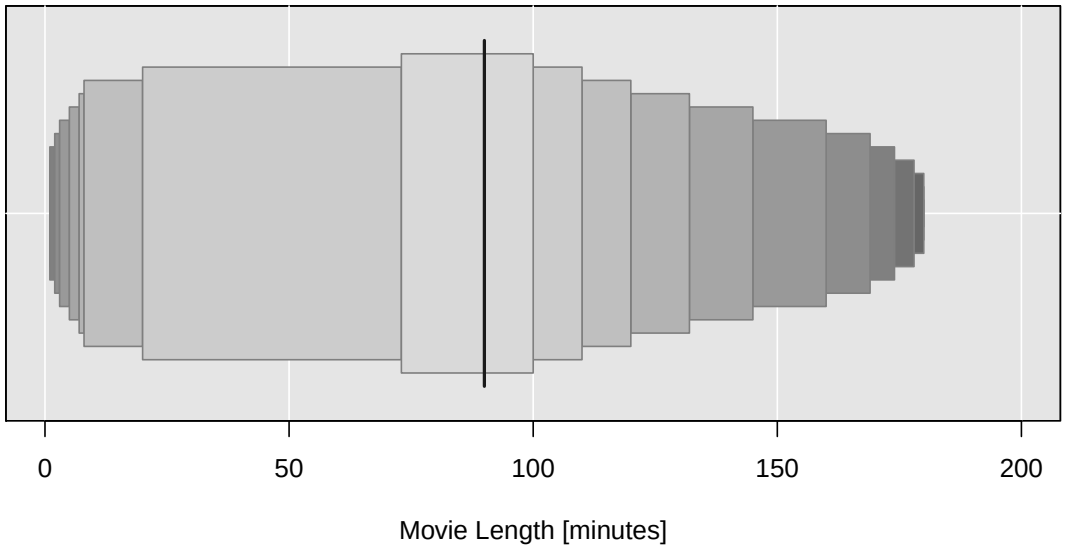
```
# boxplot
boxplot(lt180$length, ylab = "Minutes", ylim = c(1, 200), xlab = "",
        main = "Movie Length", outcol = "steelblue3")
```




```
# violin plot
vioplot(lt180$length, h = 15, names = "Movie Length [minutes]",
        col = "steelblue3", ylim = c(1, 200))
```




```
# letter value boxplot
LVboxplot(lt180$length, xlim = c(0, 200), xlab = "Movie Length [minutes]")
```



I chose to create the boxplot in *Base R*, the violin plot in *vioplot*, and the letter value plot with the `LVBoxplot()` function in *graphics*.

Of these three plots, I think that the violin plot best "matches" the histogram for these data and leads to the same conclusion. I think that the way in which the violin plot combines the density curve and the boxplot is extremely valuable and effective in illustrating the distribution of these data. The violin plot shows that these data are clearly bimodal, with most of our movie lengths being around 90 minutes long, and the other mode centered around 10 minutes or so. All the important features that are visible in the histogram are visible in the violin plot.

On the other hand, the boxplot and letter value boxplot do display certain aspects of these data well. The boxplot indicates data that are relativley symetric, with a hint of left-skewness. The boxplot also does a good job at highlighting extreme values/outliers in the data. However, if we only use the boxplot to summarize these data, we would be left without any idea that bimodality exists in these data.

The letter value boxplot shows us the bimodality of our data as well. The largest indicator of this is the largest rectangel on the plot (to the left of the median). This tells us that 12.5% of our data cover the range from about 20 minutes to 70 minutes. In other words, there is not much data in this range. The rectangels get a lot smaller down from 0 to about 15 minutes, so we know that there is another mode there. However, in this case I believe that the bimodality is much less obvious on the letter value plot than on the violin plot.

Because all of the information shown on the boxplot and letter value plot is also shown on the violin plot, and because the violin plot is easier to understand, I would suggest a violin plot for visualizing these data. I think that the violin plot most closely matches the interpretation from the histogram.