GR5702 EDAV Homework 2

Po-Chieh Liu (pl2441)

1. **Flowers**

a. Rename the column names and recode the levels of categorical variables to descriptive names. For example, “V1” should be renamed “winters” and the levels to “no” or “yes”. Display the full dataset.

Based on the dataset information page (https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/flower.html (https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/flower.html)), the variable names are as follows: winters, shadow, tubers, color, soil, preference, height, and distance. The following R codes imported the dataset and renamed the columns’ names. The binary data (winters, shadow, tubers) were converted to “Yes” and “No”. The nominal color data were converted to corresponding colors. The ordinal soil data were converted to different soil water content level.

```
data(flower)
names(flower) = c("winters","shadow","tubers","color","soil","preference","height","distance")
flower[1:3]<-as.data.frame(ifelse(flower[1:3] == 0, "No", "Yes"))
levels(flower$color) <- c( "white", "yellow", "pink", "red", "blue" )
levels(flower$soil) <- c( "dry", "normal", "wet" )
flower
```

winters <fctr>	**shadow** <fctr>	**tubers** <fctr>	**color** <fctr>	**soil** <ord>	**preference** <ord>	**height** <dbl>	**distance** <dbl>
No	Yes	Yes	red	wet	15	25	15
Yes	No	No	yellow	dry	3	150	50
No	Yes	No	pink	wet	1	150	50
No	No	Yes	red	normal	16	125	50
No	Yes	No	blue	normal	2	20	15
No	Yes	No	red	wet	12	50	40
No	No	No	red	wet	13	40	20
No	No	Yes	yellow	normal	7	100	15
Yes	Yes	No	pink	dry	4	25	15
Yes	Yes	No	blue	normal	14	100	60

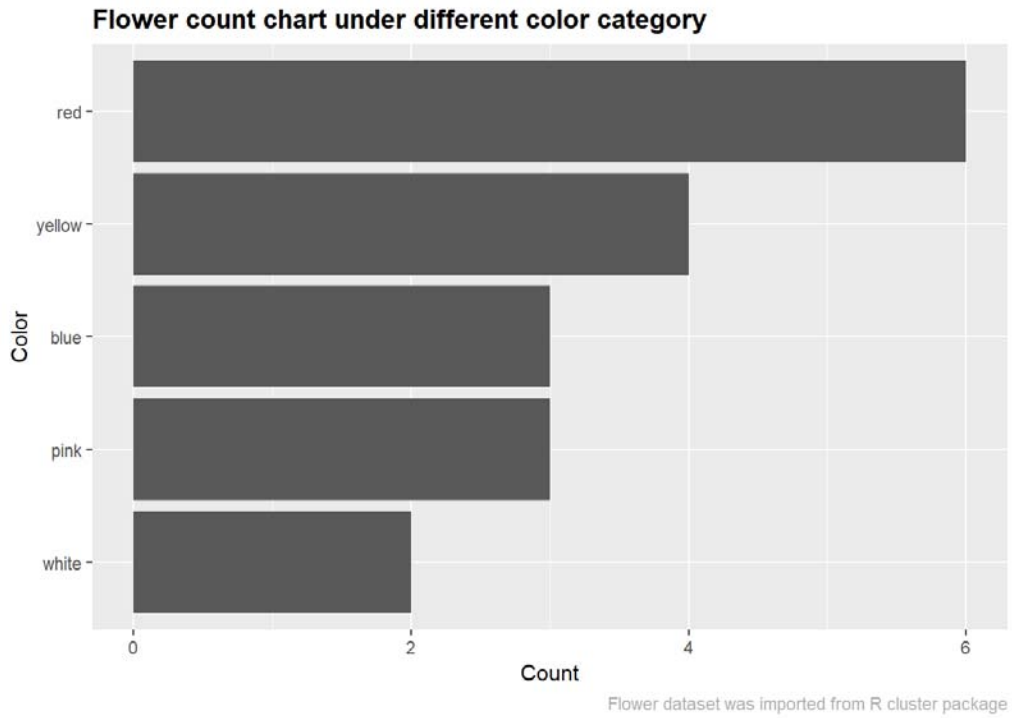
1-10 of 18 rows Previous **1** 2 Next

b. Create frequency bar charts for the `color` and `soil` variables, using best practices for the order of the bars.

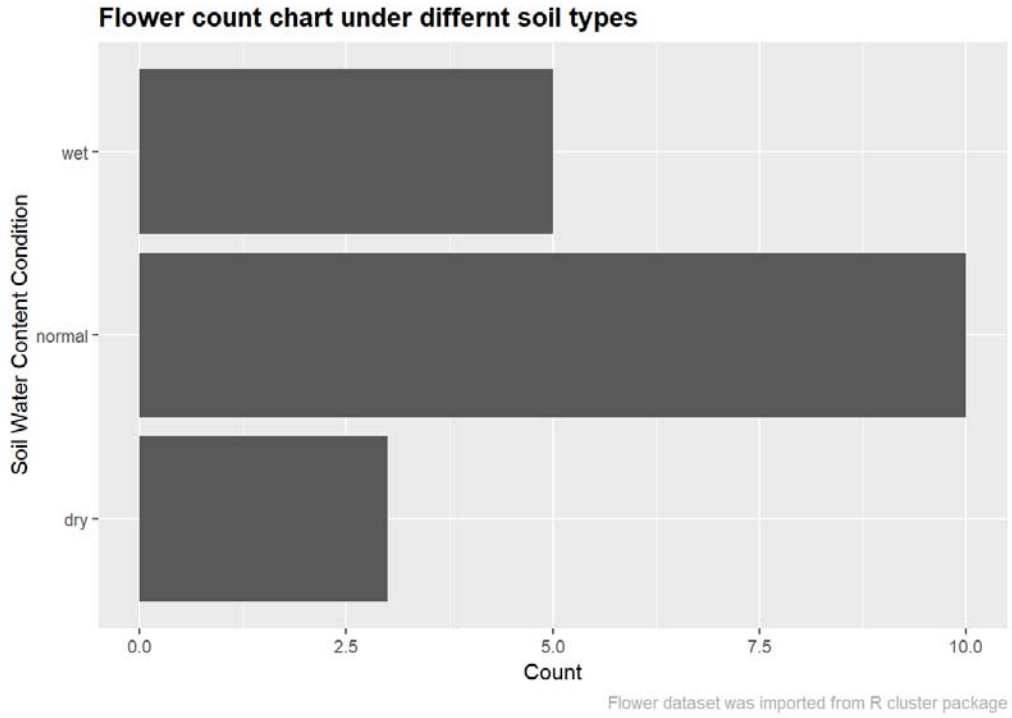
Frequency bar charts of color and soil data were generated by the following R codes. The data were preprocessed by dplyr group and summarise function and generated frequency for plotting. The bar order of color bar chart was reordered by corresponding frequency from high to low. The order of soil bar charts was ordered by ordinal data order from wet to normal to dry. The coordinate was flipped for better observation.

From the data description, the data contains 18 popular flowers with 8 different characteristics. From the color plot we can observe that red color flower has the largest number of count followed by yellow, blue, and pink. White is the less count flower. From the soil plot we can observe the normal type soil has larget number of flower count, followed by wet and dry. If the dataset was collected for investigating the flower growth condition, we can conclude that red flower might be the most easiest kind to planet. And keeping the soil under normal or wet conditions are easier for flower to growth. Moreover, we can combine both two factors together for a third plot. We can observe that only yellow and pink flowers can survive in dry soil. And blue, red and pink flowers can survive in wet soil. We might infer that yellow is drought resistant plant, and pink is easy-to-grow plant in regards to soil water content condition.

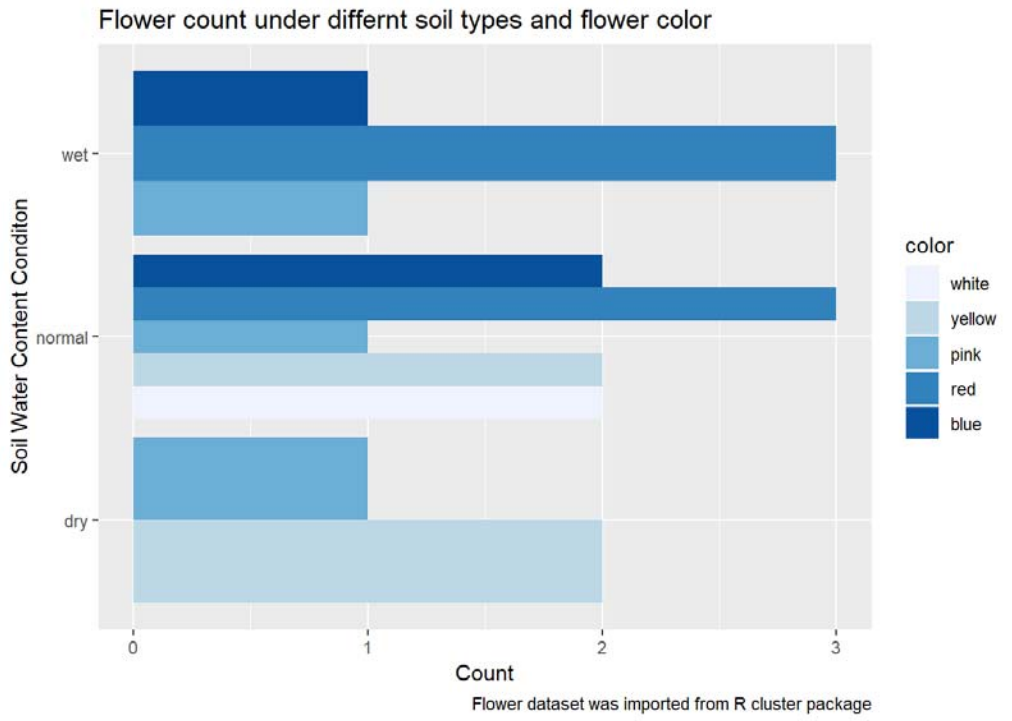
```
df <- flower %>% group_by(color) %>% summarise(n=n())
ggplot(df, aes(x= fct_reorder(color,n),y=n))+
  geom_bar(stat="identity") +
  labs(y="Count",x="Color",caption = "Flower dataset was imported from R cluster package") +
  ggtitle("Flower count chart under different color category") +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35")) +
  theme(plot.caption = element_text(color = "grey68")) +
  coord_flip()
```




```
df <- flower %>% group_by(soil) %>% summarise(n=n())
ggplot(df, aes(x=soil,y=n))+
  geom_bar(stat="identity") +
  labs(y="Count",x="Soil Water Content Condition",caption = "Flower dataset was imported from R cluster package") +
  ggtitle("Flower count chart under differnt soil types")+
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35")) +
  theme(plot.caption = element_text(color = "grey68"))+
  coord_flip()
```



```
df <- flower %>% group_by(soil, color) %>% summarise(Total = n())
ggplot(df , aes(x=soil, y = Total, fill = color)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(y="Count",x="Soil Water Content Conditon",caption = "Flower dataset was imported from R cluster package") +
  ggtitle("Flower count under differnt soil types and flower color") +
  scale_fill_brewer(palette=1) +
  coord_flip()
```



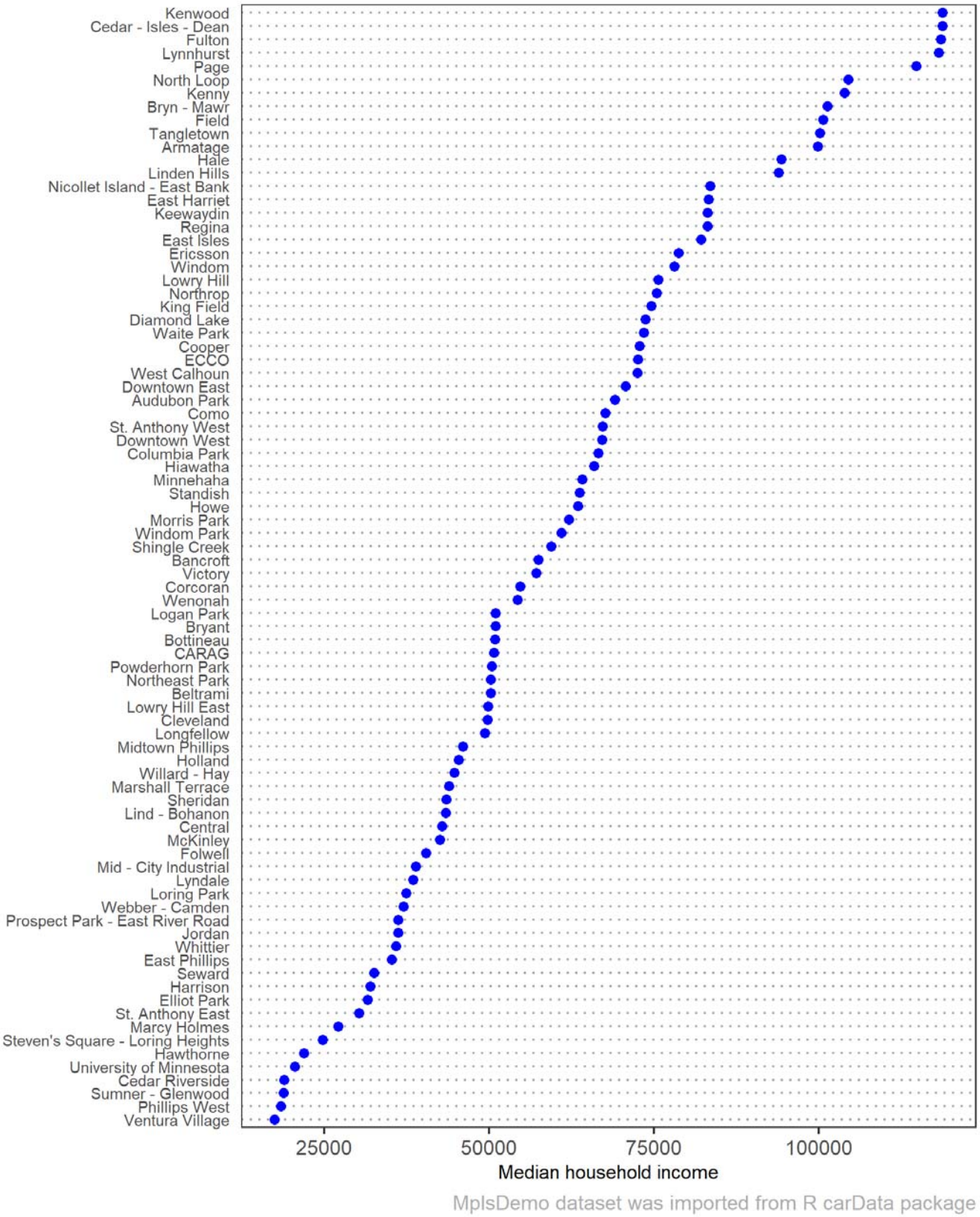
2. Minneapolis

a. Create a Cleveland dot plot showing estimated median household income by neighborhood.

The Cleveland dot plot of estimated median household income from different neighborhood was generated by the following R codes.

```
data("MplsDemo")
theme_dotplot <- theme_bw(18) +
    theme(axis.text.y = element_text(size = rel(.75)),
          axis.ticks.y = element_blank(),
          axis.title.x = element_text(size = rel(.75)),
          panel.grid.major.x = element_blank(),
          panel.grid.major.y = element_line(linetype=3, color="darkgray"),
          panel.grid.minor.x = element_blank(),
          plot.caption = element_text(color = "grey68"))

ggplot(MplsDemo, aes(x = hhIncome, y = fct_reorder(neighborhood,hhIncome))) +
  geom_point(color = "blue", size =3)+
  labs(y="",x="Median household income",caption = "MplsDemo dataset was imported from R carData package") +
  theme_dotplot
```



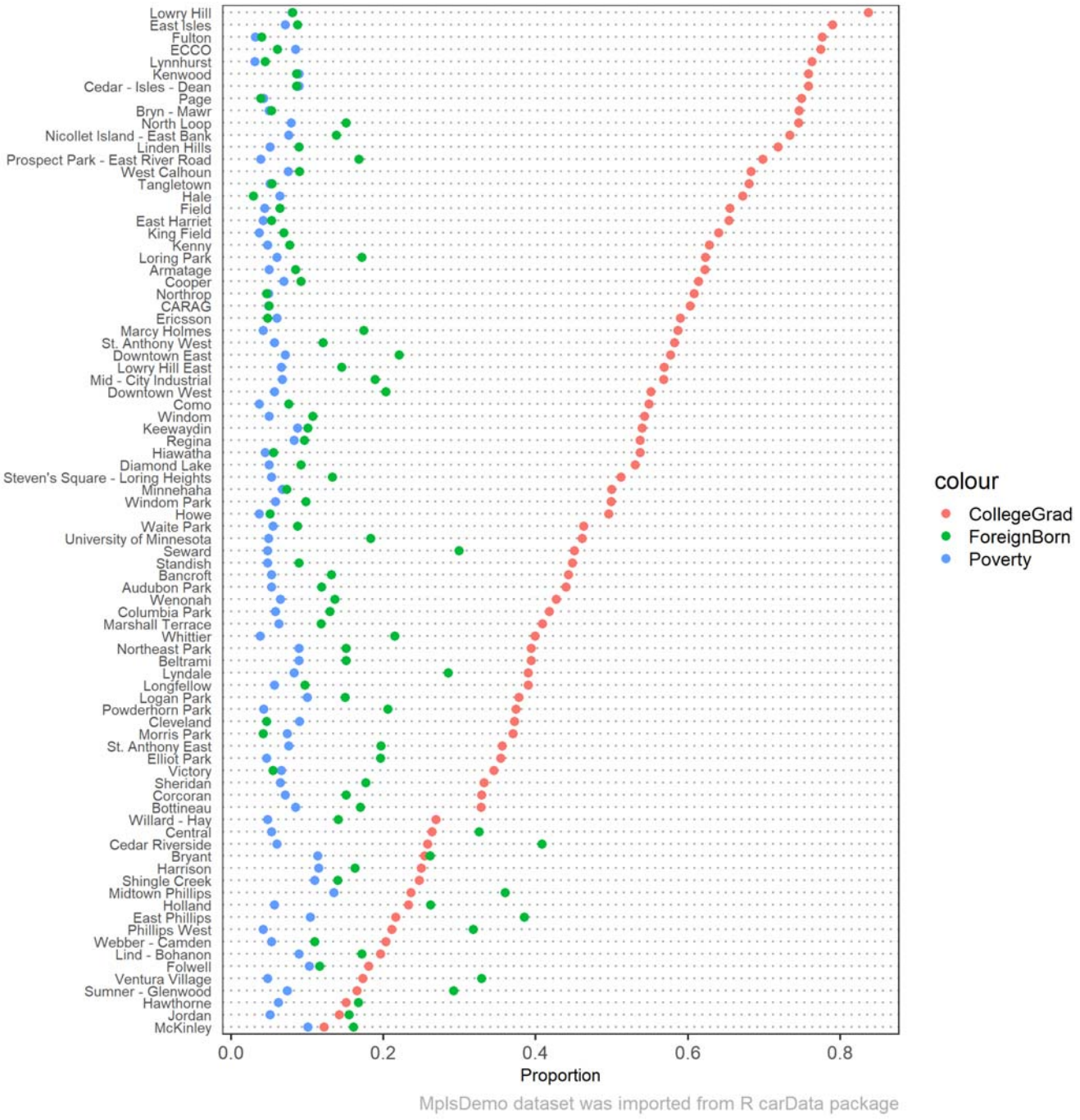
b. Create a Cleveland dot plot to show percentage of foreign born, earning less than twice the poverty level, and with a college degree in different colors. Data should be sorted by college degree.

The following R codes generated a Cleveland dot plot shows the percentage of foreign born, earning less than twice the poverty level, and college degree in regards to different neighborhoods. The neighborhoods were sorted by college degree in a descending manner.

```
library(reshape2)
df <- MplsDemo%>% select(neighborhood, foreignBorn, poverty, collegeGrad)

theme_dotplot <- theme_bw(18) +
    theme(axis.text.y = element_text(size = rel(.75)),
          axis.ticks.y = element_blank(),
          axis.title.x = element_text(size = rel(.75)),
          panel.grid.major.x = element_blank(),
          panel.grid.major.y = element_line(linetype=3, color="darkgray"),
          panel.grid.minor.x = element_blank(),
          plot.caption = element_text(color = "grey68"))

cols <- c("CollegeGrad"="#f04546","Poverty"="#3591d1","ForeignBorn"="#62c76b")
ggplot(df)+
  geom_point( aes(x = collegeGrad, y = fct_reorder(neighborhood, collegeGrad),colour="CollegeGrad"),size = 3) +
  geom_point( aes(x = poverty, y= neighborhood, colour = "Poverty"),size = 3) +
  geom_point( aes(x= foreignBorn, y = neighborhood, colour = "ForeignBorn"),size = 3) +
  theme_dotplot +
  labs(y="",x="Proportion",caption = "MplsDemo dataset was imported from R carData package")
```



c. What patterns do you observe? What neighborhoods do not appear to follow these patterns?

Since we sort the neighborhoods by the fraction of having college degree, the maximum and minimum fraction are around 82% and 12%. The fraction difference is large in compared with other two variables. Also, there are two obvious gaps between “Howe” and “White Park”, as well as “Bottineau” and Willard-Hay”. The poverty fractions visually do not differ much across all neighborhoods. One visually notable pattern is that foreign born proportion is always lower then college graduate proportion except for neighborhoods with college graduate proportion lower than 30%. Also, the foreign born fraction is visually weakly negatively associated to college grad fraction.

3. Taxis

Draw four scatterplots of `tip_amount` vs. `far_amount` with the following variations:

The following R codes were used ahead before generating Rmarkdown and html file. Because of the runtime of loading data under Rmarkdown environment is too long, the data was preprocessed including randomly sampled 10,000 records and negative fare and tip records are excluded due to possible data quality issue.

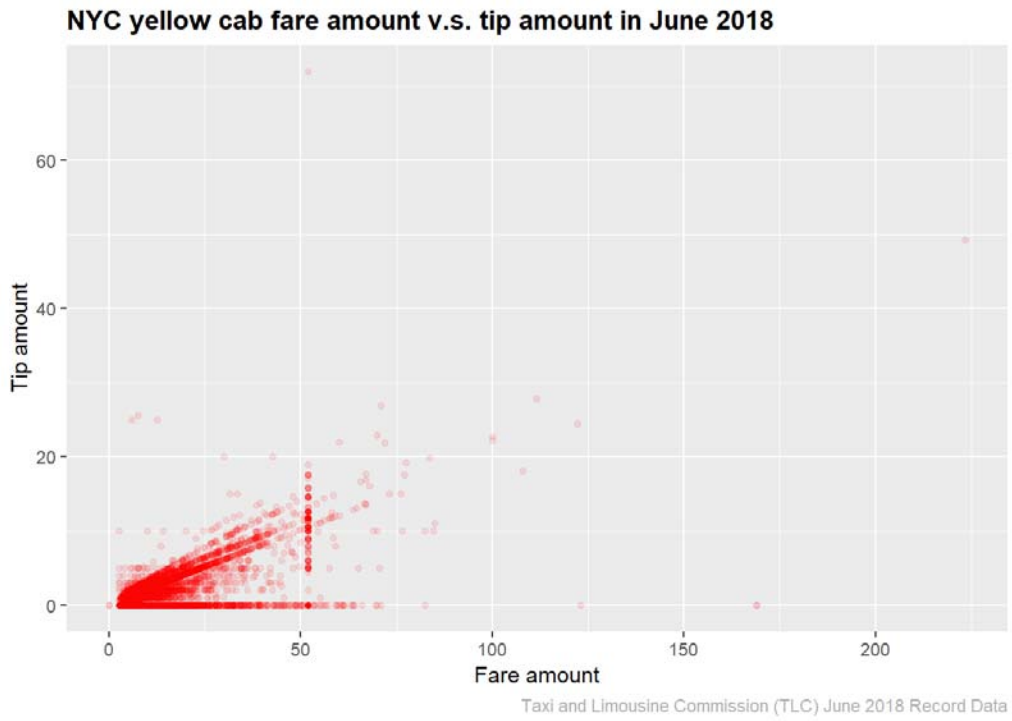
```
# Following R codes were used to randomly select 10,000 data points of fare amount and tip amout from 2018 June yellow cab d
ataset. The negative fare and tipe amounts were filtered out.

# yellowCab <- read_csv("yellow_tripdata_2018-06.csv")
# df<-yellowCab %>% select(tip_amount, fare_amount) %>%  filter(tip_amount >= 0 & fare_amount >= 0)
#yellowCab <- sample_n( df, 10000)
# save(yellowCab,file="yellowCab.Rda")
```

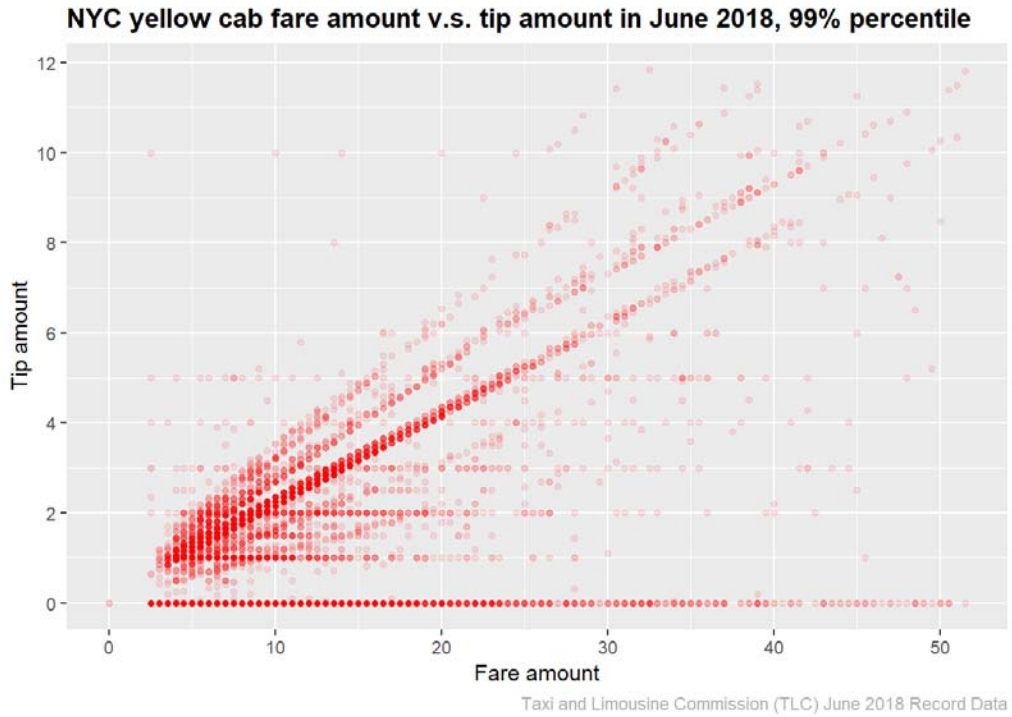
a. Points with alpha blending

The following R codes were used to generate scatter plots of fare amount vs. tip amount. The axis of first plot were stretched by extreme and rare values. The next plot only included on fare amount and tip amount under 99% percentile. In this plot we can have better visualization of the pattern of most data points.

```
load("yellowCab.Rda")
ggplot(yellowCab, aes(x=fare_amount,y=tip_amount)) +
  geom_point(alpha = 0.1, color = "red", stroke = 0) +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018") +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35")) +
  theme(plot.caption = element_text(color = "grey68"))
```



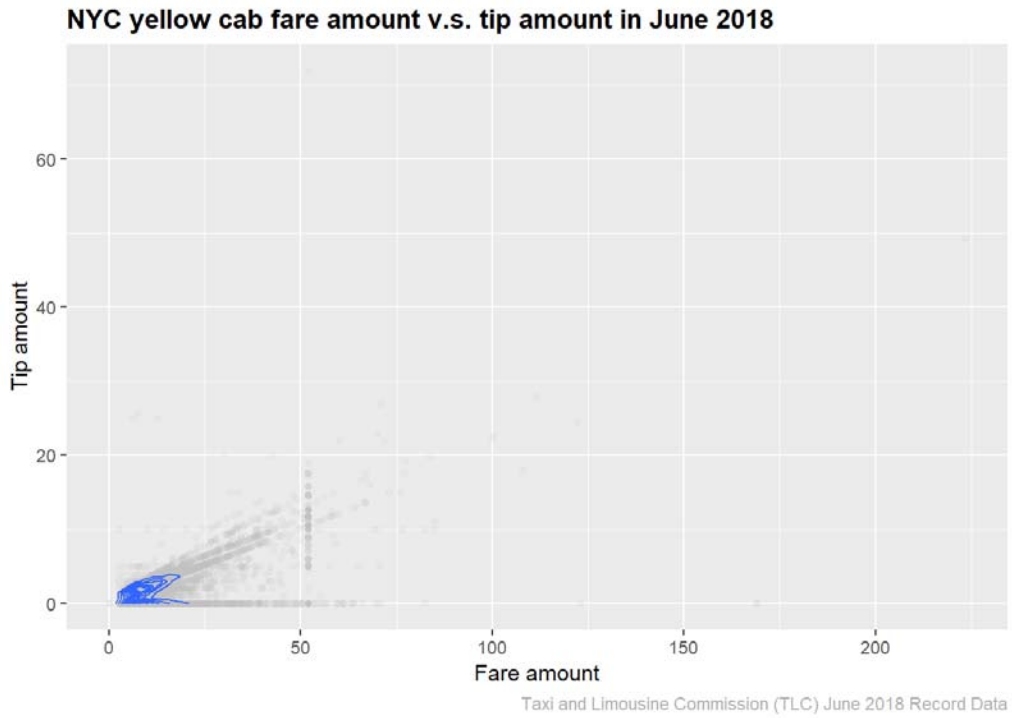
```
df <- yellowCab %>%  filter(fare_amount < quantile(fare_amount, .99) & tip_amount < quantile(tip_amount, .99))
ggplot(df, aes(x=fare_amount,y=tip_amount)) +
  geom_point(alpha = 0.1, color = "red", stroke = 0) +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  scale_y_continuous(
    labels = c("0","2","4","6","8","10","12"),
    breaks = seq(0, 12, len = 7)) +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018, 99% percentile") +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35")) +
  theme(plot.caption = element_text(color = "grey68"))
```



b. Points with alpha blending + density estimate contour lines

The following R codes generated the density contours plots with alpha blending points. The point color was changed to grey in order to have better view of the contour. Same as part (a), first and second figure generated by full and 99% data, respectively.

```
ggplot(yellowCab, aes(x=fare_amount,y=tip_amount)) +
  geom_point(alpha = 0.1, color = "grey", stroke = 0) +
  geom_density2d() +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018") +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35")) +
  theme(plot.caption = element_text(color = "grey68"))
```

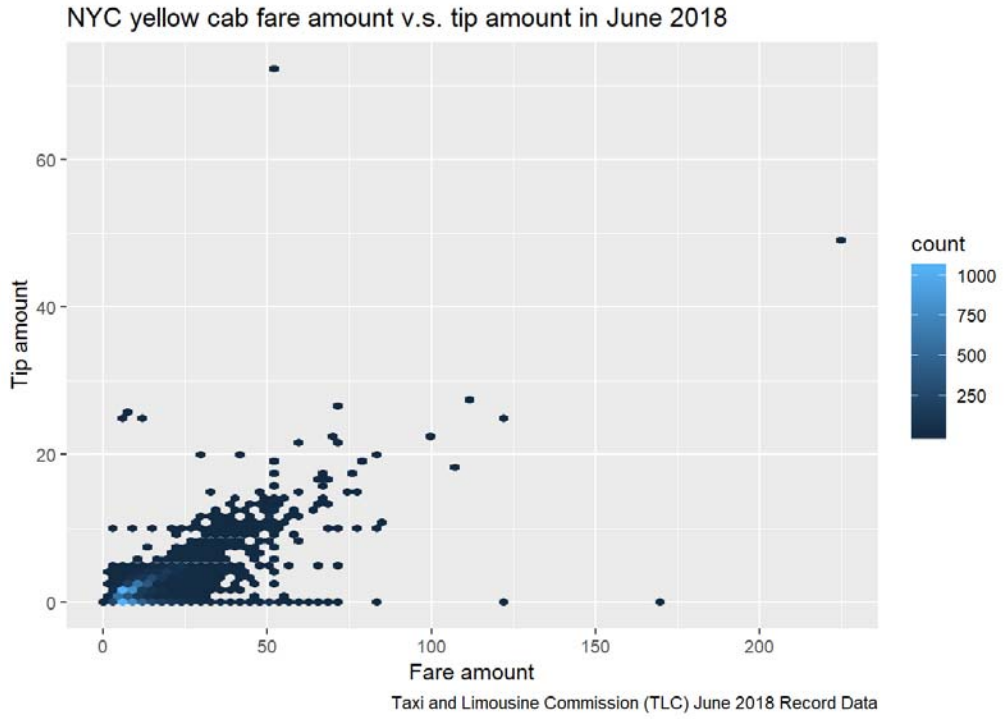



```
ggplot(df, aes(x=fare_amount,y=tip_amount)) +
  geom_point(alpha = 0.1, color = "grey", stroke = 0) +
  geom_density2d() +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018, 99% percentile") +
  scale_y_continuous(
    labels = c("0","2","4","6","8","10","12"),
    breaks = seq(0, 12, len = 7)) +
  theme(plot.title = element_text(face = "bold")) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35")) +
  theme(plot.caption = element_text(color = "grey68"))
```

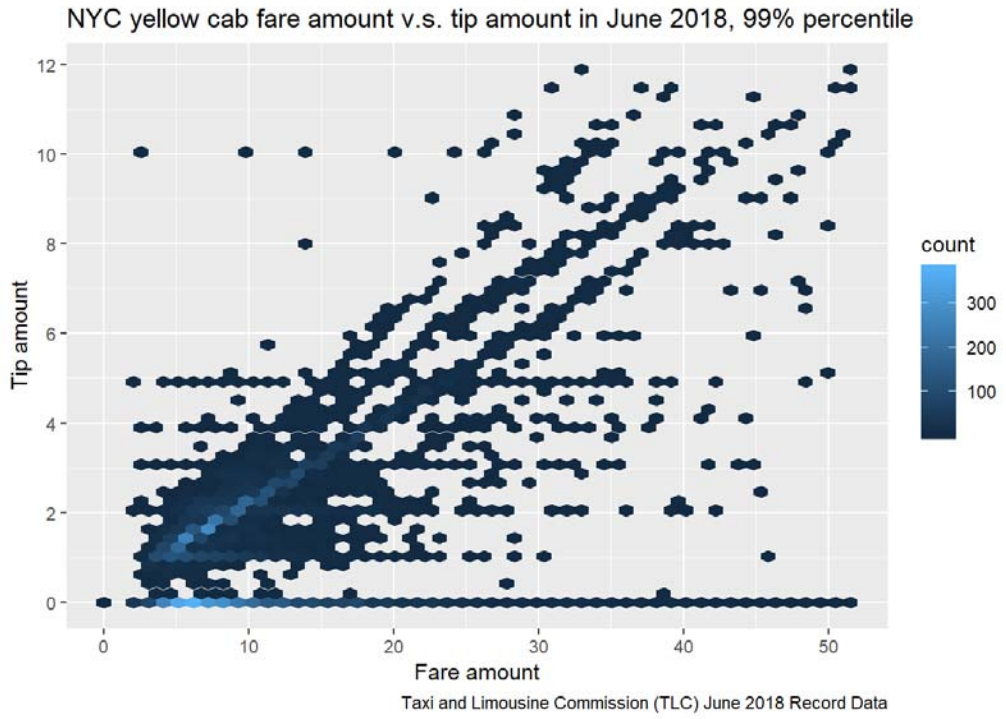
c. Hexagonal heatmap of bin counts

The following R codes generated hexagonal heatmap of bin counts plots. Same as part (a), first and second figure generated by full and 99% data, respectively.

```
ggplot(yellowCab, aes(x=fare_amount,y=tip_amount)) +
  stat_bin_hex(bins = 75) +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018")
```



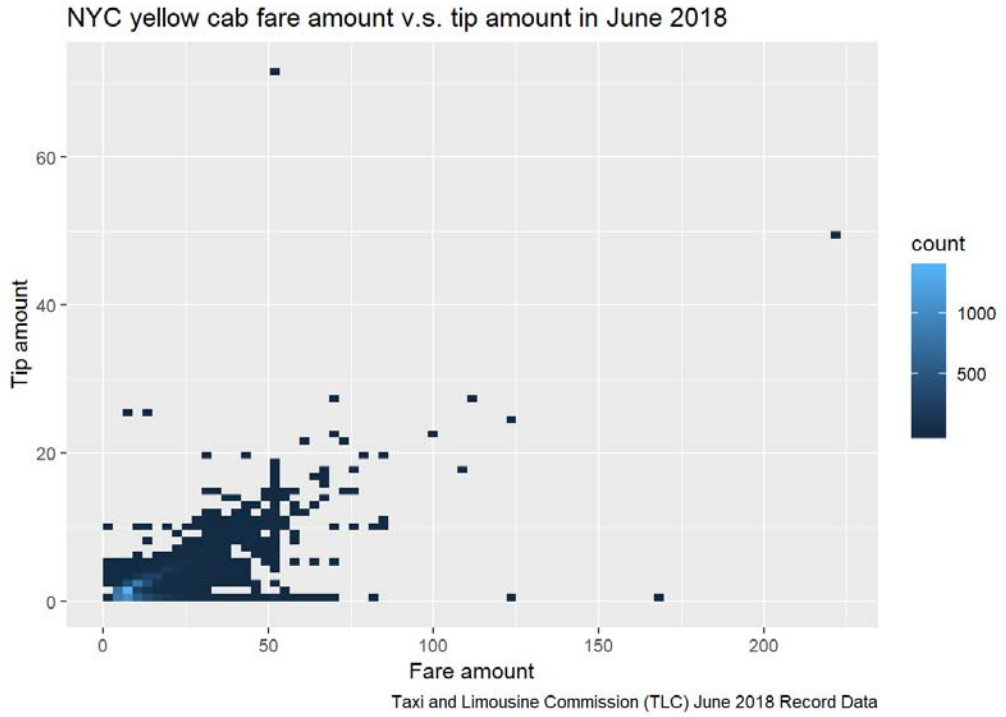

```
ggplot(df, aes(x=fare_amount,y=tip_amount)) +
  stat_bin_hex(bins = 50) +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018, 99% percentile") +
  scale_y_continuous(
    labels = c("0","2","4","6","8","10","12"),
    breaks = seq(0, 12, len = 7))
```



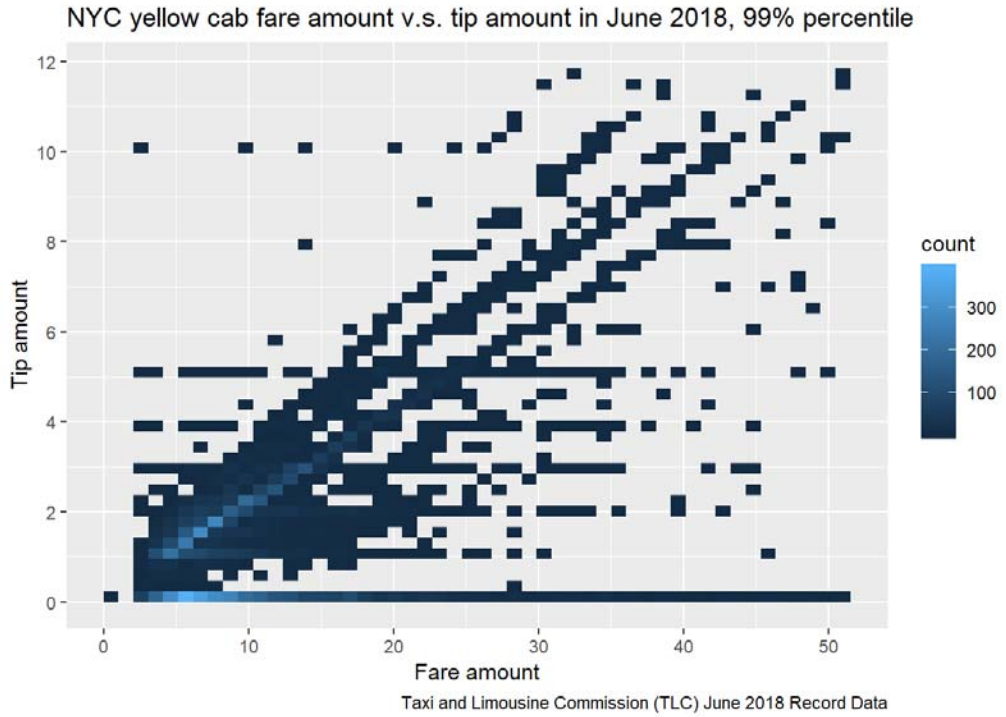
d. Square heatmap of bin counts

The following R codes generated square heatmap of bin counts plots. Same as part (a), first and second figure generated by full and 99% data, respectively

```
ggplot(yellowCab, aes(x=fare_amount,y=tip_amount)) +
  stat_bin_2d(bins = 75) +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018")
```




```
ggplot(df, aes(x=fare_amount,y=tip_amount)) +
  stat_bin_2d(bins = 50) +
  labs(y="Tip amount",x="Fare amount",caption = "Taxi and Limousine Commission (TLC) June 2018 Record Data") +
  ggtitle("NYC yellow cab fare amount v.s. tip amount in June 2018, 99% percentile") +
  scale_y_continuous(
    labels = c("0","2","4","6","8","10","12"),
    breaks = seq(0, 12, len = 7))
```



e. Describe noteworthy features of the data, using the “Movie ratings” example on page 82 (last page of Section 5.3) as a guide.

From above plots, we can observe following patterns:

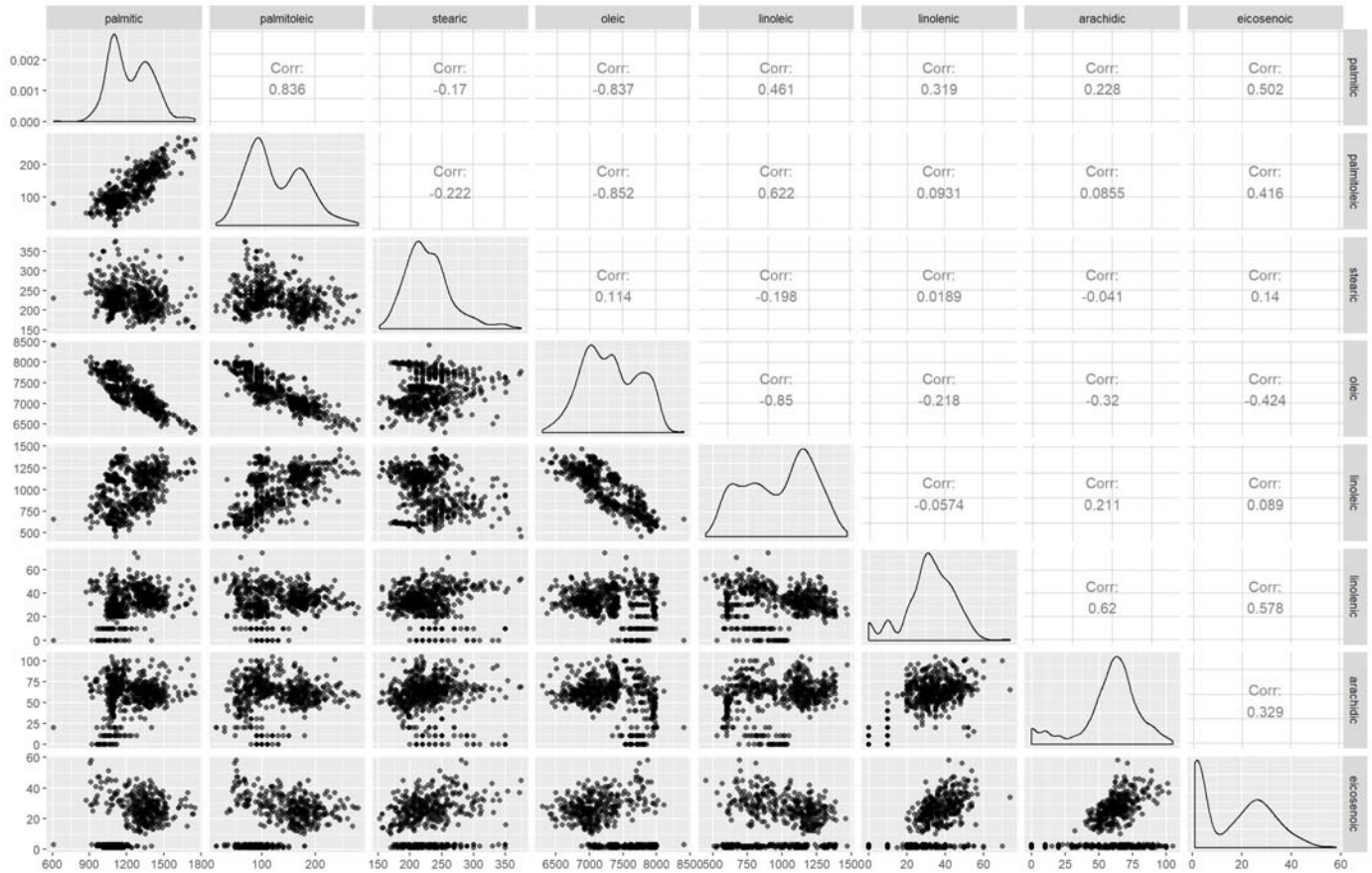
1. In first plot of (a), there are lots points line up vertical around $52, which indicate the tip amounts differ a lot around fare amount $52. After searching online, those data points might be the night surcharge ($50) related fee. The vertical line indicates some customers may give tips based on the total charge (With surcharge), and some may give tips without considering the surcharge.
2. There are couple linearly lined up dots/bins we can observe from dot plots, heat maps. Roughly estimated we can find the tip/fare ratios are 15%, 20%, and 25%. Those are recommended tip ratios after using the charge machine in Cap, which indicates a portion of people might choose the default tip ratios. Also, $0 tip line is visually observable, which means a portion of rider tend to give no tip in the end.
3. There are some horizontal lines we can observe from scatter plots and heat maps. $0, $1, $2, $3, $4, $5 and $10 are frequently show up in all plots. Those show a fraction of riders tend to give fixed amount tip or no tip instead of the ratio of fare amount.
4. From the density plots and heat maps, we can observe that most frequent recorded data points line up on 20% tip/fare line in the region under $20 fare amount, and $0 tip line under $10 fare amout. We can infer from these that a large portion of rider take cab for short distance (under $20) travel, these are the most frequent fare/tip combination we observed in the data.

4. Olive Oil

a. Draw a scatterplot matrix of the eight continuous variables. Which pairs of variables are strongly positively associated and which are strongly negatively associated?

The following R codes generated the scatterplot matrix of the eight continuous variables. From the plot we can observe the association between each pair of variables. Based on the matrix plot, we can observe that palmtic and palmitoleic are strongly positively associated due the highly linearly trend from bottom left corner to top right corner. The correlation coefficient is 0.836. linoleic are weakly positively associated with palmtic and palmitoleic due to wider scatter plot trend. The correlation coefficients are 0.461 and 0.622, respectively. Also we can observe that palmitic, palmitoleic, and linoleic are strongly negatively associated with oleic. The plots show the linearly trend from top left to bottom right. The correlation coefficients are varied around -0.85. The rest of plots do not show the visually distinguishable linear relationship.

```
library(GGally)
df <- olives %>% select(palmitic,palmitoleic,stearic,oleic,linoleic,linolenic,arachidic,eicosenoic)
ggpairs(df, aes(alpha=0.3))
```



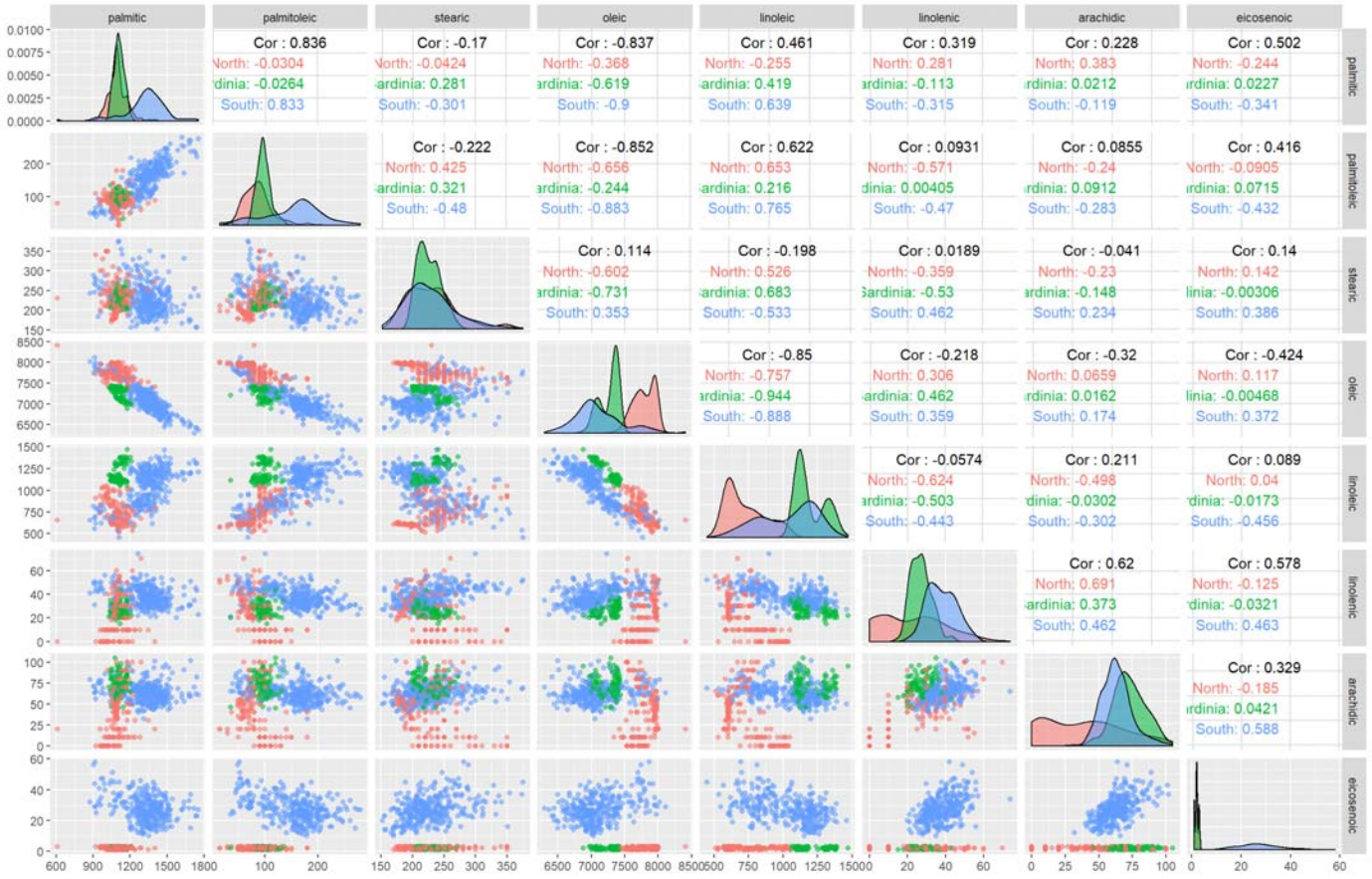
b. Color the points by region. What do you observe?

The following R codes generate the scatter plot matrix and colored by region. First thing we can observe is that South region data points are widely spread in all plots. The strongly positively and negatively relationship between palmltic, palmitpleic, oleic, and linoleic are mainly represented by south region data points. In contrast, Sardinia and North Region data points are concentrated in a smaller area. The obviously notable plots are

eicosenoic vs. other variables, most data points for Sardinia and North region are zeros. For North region, we can also observe lots of horizontally or vertically spread data in arachidic and linolenic rows, which is very special pattern. For Sardinia, most data points are concentrated in a small area. After coloring the data points by regions, the colored plots illustrate the differences between regions clearly. There are also some notable opposite correlations we can observe from colored plot. Following Opposite correlations were foud:

1. Stearic vs. palmitoleic: negative for South but positive for the other two regions
2. oleic vs. Stearic: Positive for South but negative for the other two regions
3. linoleic vs. Stearic: Negative for South but positive for the other two regions
4. linolenic vs. stearic: Positive for South but negative for the other two regions

```
df2 <- olives %>% select(palmitic,palmitoleic,stearic,oleic,linoleic,linolenic,arachidic,eicosenoic,Region)
ggpairs(df2, aes(colour=Region, alpha=0.3),columns = 1:8)
```

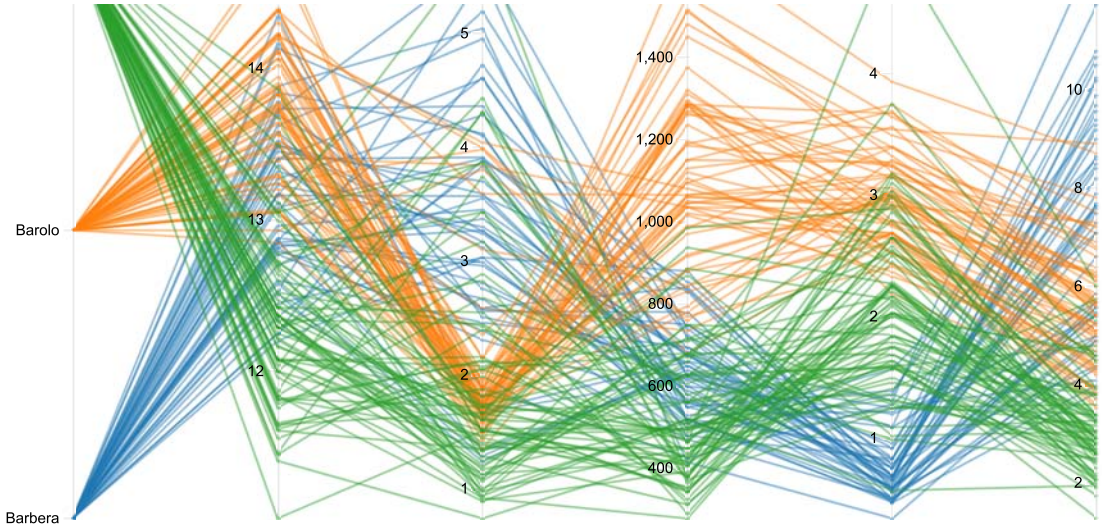


5. Wine

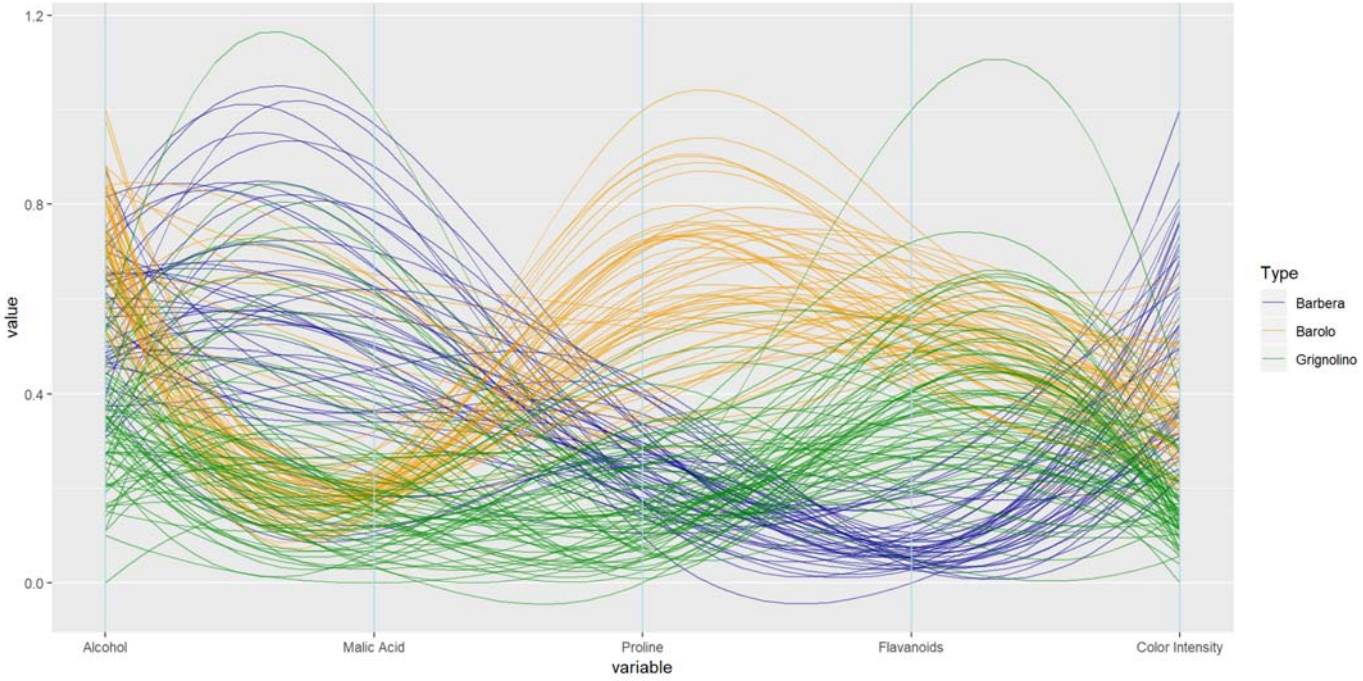
a. Use parallel coordinate plots to explore how the variables separate the wines by `Type`. Present the version that you find to be most informative. You do not need to include all of the variables.

The following R codes were used to generate two parallel coordinate plots of wine dataset. Only 6 variables were manually selected by myself to present the plots. The selection rationale is provided in part (b)

```
library(parcoords)
wine  %>% arrange(Type) %>% select(Type, Alcohol, `Malic Acid`, `Proline`, `Flavanoids`, `Color Intensity`) %>%
  parcoords(
    rownames = F
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    , alpha = .5
    , color = list(colorBy = "Type",colorScale = htmlwidgets::JS("d3.scale.category10()") )
    )
```




```
df <- wine  %>% arrange(Type) %>% select(Type, Alcohol, `Malic Acid`, `Proline`, `Flavanoids`, `Color Intensity`)
ggparcoord(df, columns = 2:6, groupColumn = 1, scale = "uniminmax",splineFactor = 10, alpha=0.5) +
  geom_vline(xintercept = 1:5, color = "lightblue") +
  scale_color_manual(values=c("blue4", "orange2", "green4"))
```



b. Explain what you discovered.

The 6 selected variables are Type, Alcohol, Malic Acid, Proline, Flavanoids, Color Intensity. The selection creteria is that I can visually observe the differences between three types of wine from the variable. From the plots, alcohol fraction and Proline values are highest in Barolo, following by Barbera and Grignolino. Barbera has the highest clustered values of Color Intensity and Malic Acid, followed by Barolo and Grignolio. The Flavanoids shows another trend that Barolo has highest clustered values then followed by Barbera and Grignolino. The rest of variables which were not selected here are not visually distinguishable for me. The variables might be useful for determine unknown type wine or for recommending wines for customers who have specific preference. Or we can understand which wine properties/characteristics may drive sales in specific wine store.