3. Data Visualization in R

R has great data visualization functions that are quite simple to use. Modern R has extended the visualization capacities of R with the `ggplot2` package, discussed in Appendix A. Here we give an overview of the types of graphs you can create using standard R. To get a feel for the R's graphic capabilities, type `demo(graphics)` at the console. The purpose of this chapter is to get readers familiar with how data can be visualized in R. You might want to skim over this chapter to get the big picture, then refer back to it as you make your own graphs in the context of machine learning solutions later in the book.

In this chapter we first dicuss data visualization techniques for a single column of a data frame. Then we discuss data visualization for two columns, two dimensional visualization. Throughout the chapter we use the Titanic data for the sample graphs. As usual, you can find the code for all the graphs on the github. Additionally, there are two good resources that you should bookmark:

- An overview of graphical parameters from the stat methods site: `https://www.statmethods.net/advgraphs/parameters.html`
- Color names for colors in R provided by Professor Tian Zheng at Columbia: `http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf`

You can display graphs individually, or display them in grids. The par() function is used to set up a grid. The following code shows how to save the original parameter settings in a variable called opar, then plot graphs in a 1x2 grid, and finally restore the original parameter settings. The par() function can be used to set up any grid pattern you like, such as 3x2, etc. The graphs will be placed left-to-right, top-to-bottom, in the grid.

```
opar <- par()      # copy original settings
par(mfrow=c(1,2))  # set up 1x2 grid
hist(...           # make a plot
plot(...           # make another plot
par(opar)          # restore parameter settings
```

3.1 Data Visualization of One Variable

In the online notebook, we first load the Titanic data and do a little clean-up. A given column in a data frame is just a vector. We first look at ways to plot quantitative vectors and then look at visualizing qualitative vectors.

3.1.1 Quantitative Vectors

The most common graph type for one quantitative variable is the histogram. You can specify the bins, but in the graph below, using the default settings worked out well. Type `?hist()` at the console to see all the parameters you can modify. Another plot type that is appropriate for quantitative data is a simple scatterplot. Since we only have one variable, R will supply row index numbers for the x axis. The code and graphs are shown below.

Code 3.1.1 — Titanic data. Histogram and Scatterplot.

```
opar <- par()    # copy original settings
par(mfrow=c(1,2))
hist(df$age, col="slategray", main="Age of Titanic Passengers",
    xlab="Age")
plot(df$age, pch=21, cex=0.75,
   bg=c("snow", "slategray")[unclass(df$survived)], ylab="Age",
   main="Age (White Deceased)")
par(opar)
```

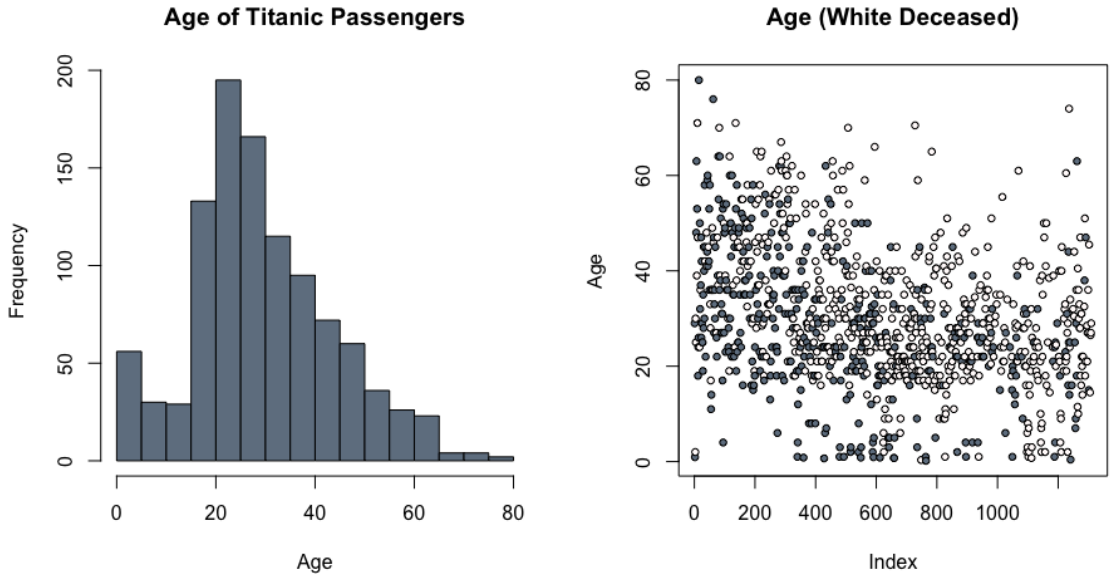

Figure 3.1: Plotting a Quantitative Vector

You can create the histogram with code as simple as `hist(df$age)` to get the visualization you need for the data. Later you can add the colors, titles, etc. For the scatter plot we added color, conditioned on the survival status of the passengers.

Another option for a quantitative vector is the kernel density plot. This plot gives you similar information as the histogram, but smoothing has been applied. In the code below, we first create the density vector, then use it for the plot() function. The last line of code below fills in the curve with a polygon.

Code 3.1.2 — Titanic data. Kernel Density Plot.

```
d <- density(df$age, na.rm = TRUE)
plot(d, main="Kernel Density Plot for Age", xlab="Age")
polygon(d, col="wheat", border="slategrey")
```

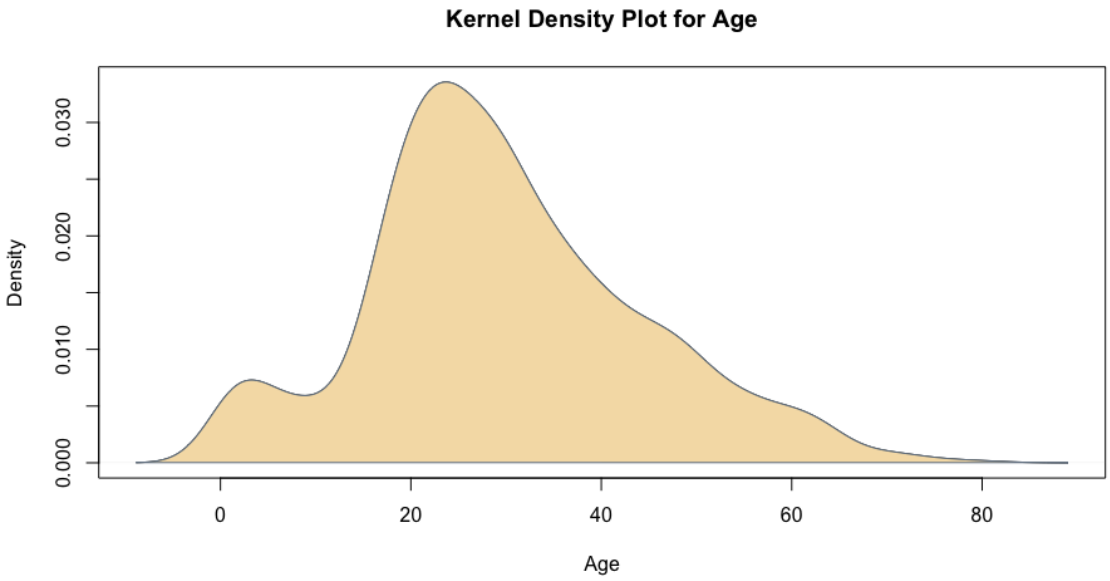

Figure 3.2: Kernel Density Plot

We can overlay several kernel density plots using package sm. First we subset the data frame to the two columns of interest so that we can use complete.cases() to get rid of NAs.

Code 3.1.3 — Age by Class. Overlaying Kernel Density Plots.

```
library(sm)
# subset the data and remove NAs
df_subset <- df[,c(1,5)]
df_subset <- df_subset[complete.cases(df_subset),]
# create the plots
sm.density.compare(df_subset$age, df_subset$pclass,
     col=c("seagreen", "wheat", "sienna3"), lwd=2)
title(main="Age by Passenger Class")
legend("topright", inset=0.05, legend=c(1:3),
     fill=c("seagreen", "wheat", "sienna3"))
```

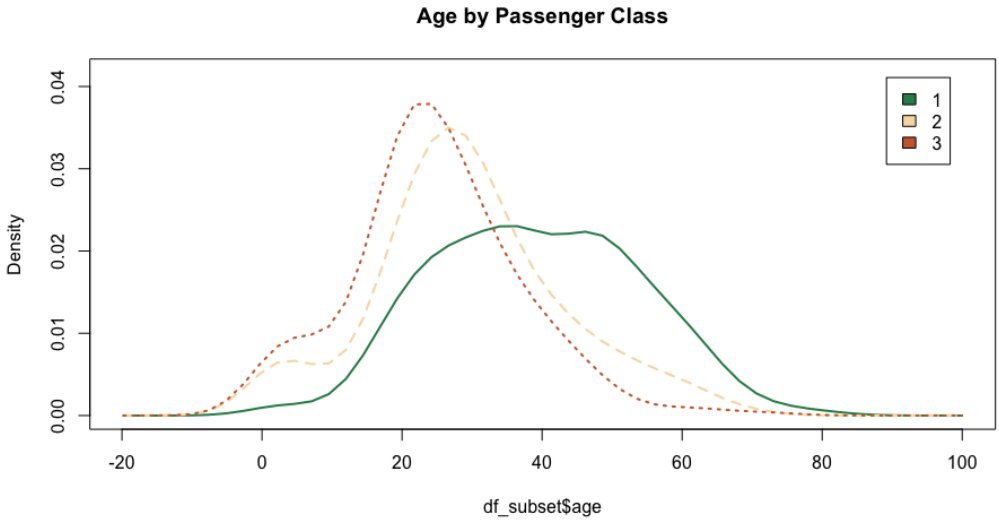

Figure 3.3: Kernel Density Plots

A boxplot is another graph type that can represent quantitative data. A box plot is more commonly vertical but below we show a horizontal example. The box shows the 2nd and 3rd quartiles of the data. The "whiskers" at either end of the dashed lines show the 1st and 4th quartiles. Dots beyond a whisker indicate suspected outliers. The bold line through the box indicates the median.

Code 3.1.4 — Age Data. Horizontal Box Plot.

```
boxplot(df$age, col="slategray", horizontal=TRUE, xlab="Age",
    main="Age of Titanic Passengers")
```

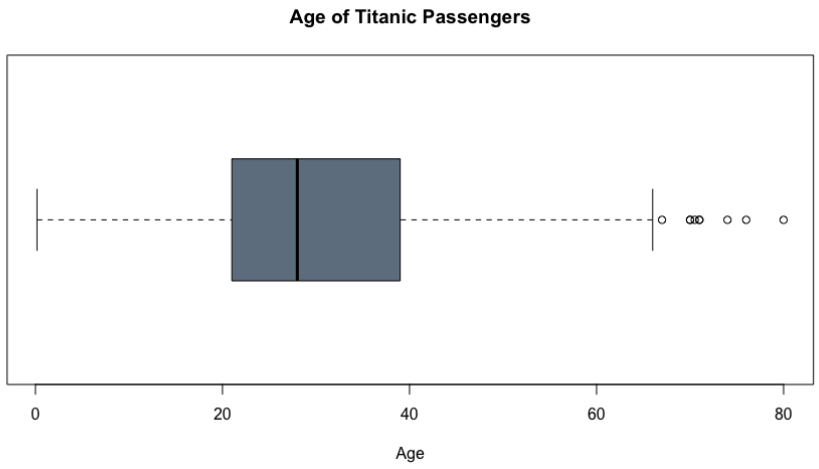

Figure 3.4: Box Plot

3.1.2 Plotting Qualitative Vectors

Barplots are often used for qualitative vectors. They can be vertical or horizontal. In the code below, adding parameter `horiz=TRUE` will cause the bars to be displayed horizontally instead of vertically. First, we make counts from the passenger class vector, then use those to create the bar plot.

Code 3.1.5 — Passenger Class Data. Bar Plot.

```
counts <- table(df$pclass)
barplot(counts, xlab="Passenger Class", ylab="Frequency",
     col=c("seagreen","wheat","sienna3"))
```

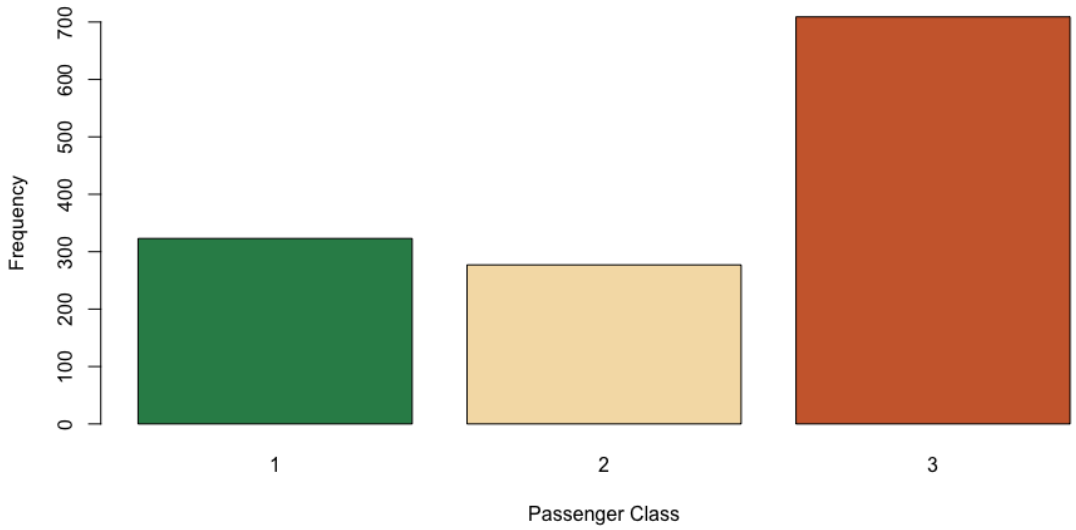

Figure 3.5: Bar Plot

A pie chart can be made with relative frequencies of a quantitative variable. First we specify frequencies for each of the 3 classes, then supply labels. With slices and labels defined, we can make a pie chart.

Code 3.1.6 — Passenger Class Data. Pie Chart.

```
slices <- c(sum(df$pclass==1, na.rm = TRUE), sum(df$pclass==2,
    na.rm = TRUE), sum(df$pclass==3, na.rm = TRUE))
lbls <- c("Class 1", "Class 2", "Class 3")
pie(slices, labels=lbls, main="Passenger Classes",
    col=c("seagreen","wheat","sienna3"))
```

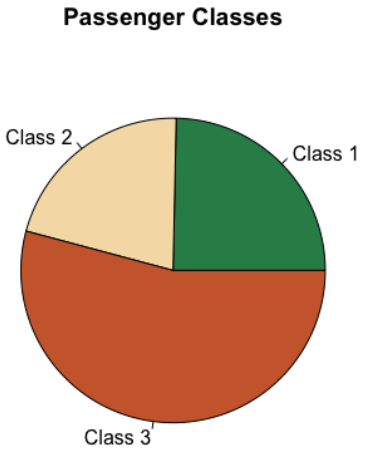

Figure 3.6: Pie Chart

3.2 Data Visualization of Two Vectors

If we have two vectors, X and Y, then there are four possible combinations of quantitative and qualitative vectors, listed below. In this section we look at graphs that are appropriate for each combination.

- both X and Y are qualitative
- X is qualitative, Y is quantitative
- X is quantitative, Y is qualitative
- X and Y are quantitative

3.2.1 Both X and Y are Qualitative

When both variables are qualitative, mosaic plots are the most common type of graph used. A related type is the association graph, which gives additional visual information about the deviation of the data from a uniform distribution. Both types of plots can be created with the `vcd` package, visualizing categorical data. First, we look at a mosaic example plotting the survived and pclass columns. The mosaic() function wants the first argument to be a table or formula, so we surround the subsetted data frame with table(). SHADE=TRUE gives you a color graph, FALSE gives you a greyscale graph.

The mosaic plot shows each group in tiles. The area of the tiles is proportional to its counts in the data.

The legend indicates the Pearson residuals. The "null" model would consider an even distribution into the cells but clearly we don't have that case here. The blue indicates we have more observations than expected, the red indicates fewer than expected, and gray is about what is expected given a null hypothesis. We didn't have to specify legend=TRUE because that is the default.

Code 3.2.1 — Passenger Class and Survived. Mosaic and Association Plots.

```
library(vcd)
mosaic(table(df[,c(2,1)]),  shade=TRUE, legend=TRUE)
assoc(table(df[,c(1,2)]), shade=TRUE)
```

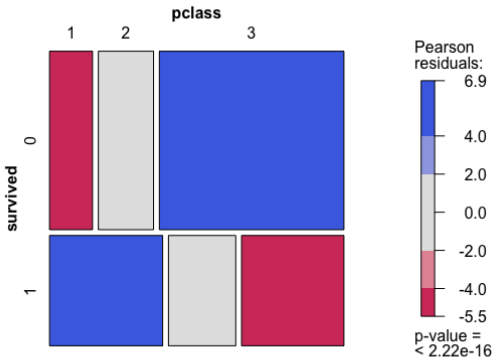

Figure 3.7: Mosaic Plot

What would happen if we reversed the order of columns 2 and 1? We would get the same information, but with the graph flipped around.

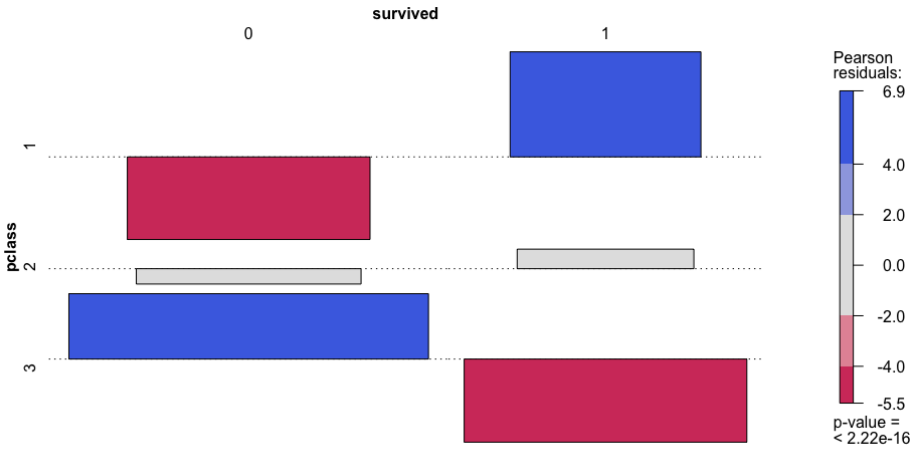

Figure 3.8: Association Plot

An association plot visualizes the residuals of an independence model. Each tile has an area that is proportional to the difference in observed and expected frequencies. The dotted line is the baseline. Tiles above the line have a frequency greater than what was expected, those below have a frequency below what was expected. In the plot above, pclass 1 survived more than expected, pclass 3 less than expected.

3.2.2 X is Qualitative, Y is Quantitative

When X is qualitative (a factor), and Y is quantitative, box plots are good choices. Notches at the median can be added with the notch=TRUE parameter. If the notches do not overlap, then it is likely that medians differ.

Code 3.2.2 — Passenger Class and Age. Box Plot.

```
plot(df$survived, df$age, varwidth=TRUE, main="Survival and Age",
    xlab="Survived", ylab="Age")
# the following would create an identical plot
boxplot(df$age~df$survived, varwidth=TRUE, main="Survival and Age",
    xlab="Survived", ylab="Age")
```

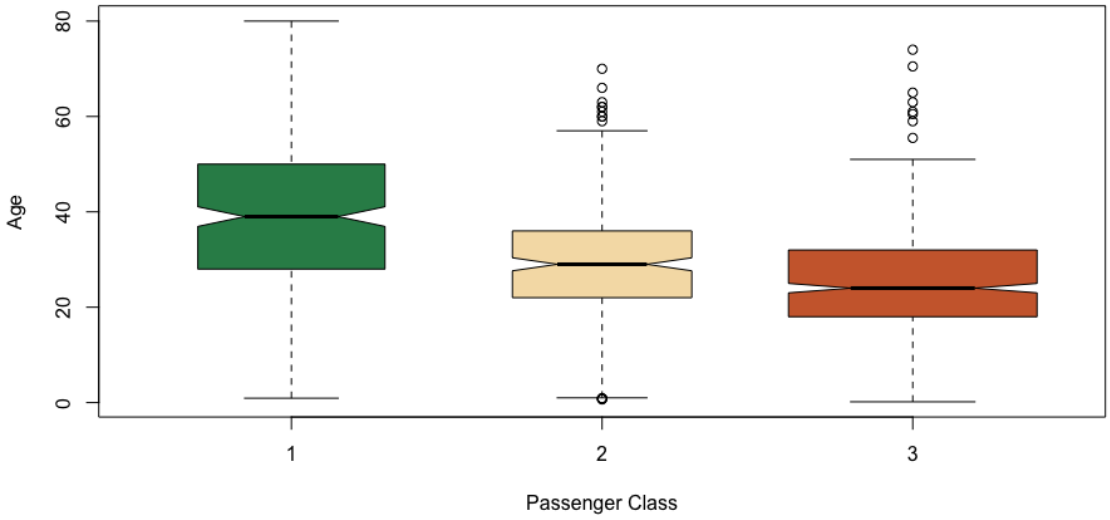

Figure 3.9: Association Plot

You can also create violin plots with package vioplot. Violin plots are a combination of a boxplot and a kernel density plot. This plot does not like NAs so we remove them.

Code 3.2.3 — Passenger Class and Age. Violin Plot.

```
library(vioplot)
df_subset <- df[,c(1,2,5)]
df_subset <- df_subset[complete.cases(df_subset),]
x1 <- df_subset$age[df_subset$pclass==1]
x2 <- df_subset$age[df_subset$pclass==2]
x3 <- df_subset$age[df_subset$pclass==3]
vioplot(x1, x2, x3, col="wheat",
    names=c("Class 1", "Class 2", "Class 3"))
```

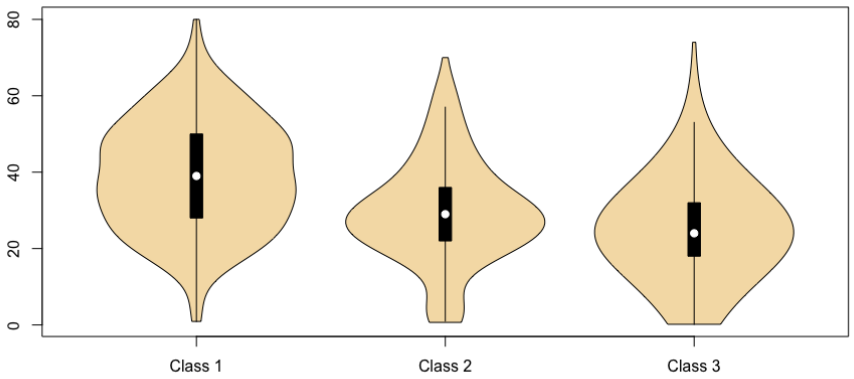

Figure 3.10: Association Plot

3.2.3 X is Quantitative, Y is Qualitative

When X is quantitative and Y is qualitative, a conditional density plot can be used. The following plot shows how survived changes over the various ages. Note that if we switched the order of age and survived, we would get a row of dots at the top of the graph for one class and a row of dots at the bottom of the graph for the other class, not terribly informative.

Code 3.2.4 — Passenger Class and Age. Conditional Density Plot.

```
cdplot(df_subset$age, df_subset$survived, col=c("snow", "gray"))
```

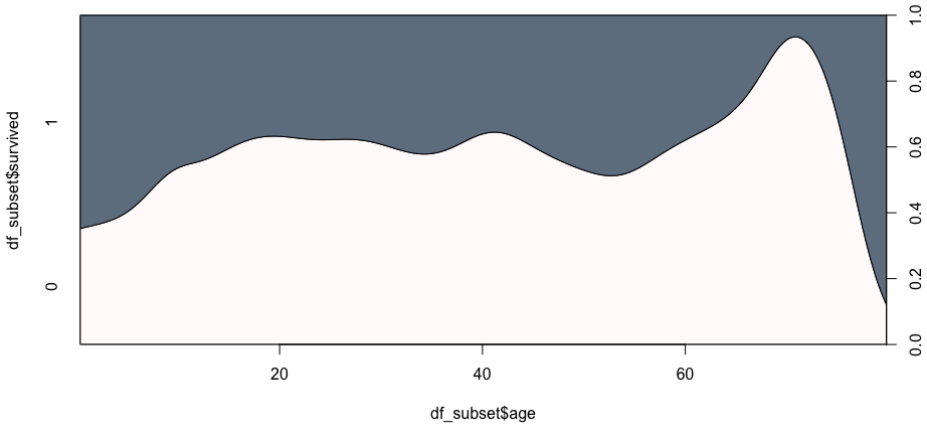

Figure 3.11: Association Plot

3.2.4 X and Y are both Quantitative

If X and Y are both quantitative, scatter plots are recommended. Here we have crosses for the points in blue, 75% of the usual size. We would have to dig further into the Titanic data to understand this chart. Why do so many passengers seem to have a fare of 0? And why did a few passengers pay 500? Perhaps the 500 fares paid for several people and the 0 fares reflect passengers whose fares were paid by a spouse or parent or adult child? Further investigation is required to understand this.

Code 3.2.5 — Fare and Age. Violin Plot.

```
plot(df$age, df$fare, pch='+', cex=0.75, col="blue",
    xlab="Age", ylab="Fare")
```

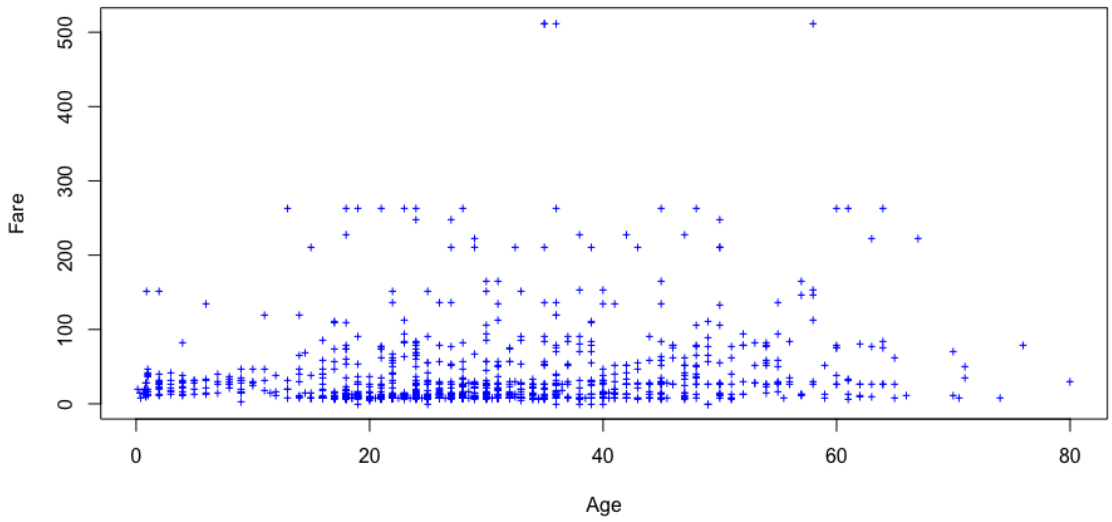

Figure 3.12: Association Plot

3.3 Summary

This chapter demonstrated how to create informative graphs in basic R. We have really just scratched the surface of what can be done in R. The data visualization capabilities of R are one of many reasons that it is heavily used in industry and academia. Data visualization is useful first for a researcher's understanding of the data, and then to communicate what has been learned about the data to others. There are many online resources for advanced data visualization techniques in R. Advanced graphics can be achieved with the power ggplot2() package, described in Appendix A.