Base R Replication

Sam Eckhardt

9/16/2014

Abstract

Base R Plots

Purpose

I am creating this document as a guide to you, newcomer. Welcome to R. I am your tour guide, Alfonzo. R, the code you see used in this paper, is used by many statisticians as graph making software. To start, each of the shaded areas of this paper is what I have entered into RStudio to make the graph that follows. Anything with a "#" in front of it within those shaded areas are notes, notes that you can make to help yourself. They are not necessary for the code, but when dealing with a huge string of code, it will be very helpful to you.

Data

First, we will start with creating some data. If you have a data set of your own, this is where you would use that as a source instead. For this guide, we will have three factor, or qualitative, variables, and four numeric, or quantitiative, variables.

```
## Simulate some data. This is all randomly generated data from R.

## 3 Factor Variables
FacVar1=as.factor(rep(c("Male","Female"),25))
FacVar2=as.factor(rep(c("Blonde","Brunette","Redhead"),17)[-51])
FacVar3=as.factor(rep(c("North","South","East","West"),13)[-c(51:52)])

## 4 Numeric Variables.
##R uses a seed to randomly generate data. The seed for the data we are using is seed 123.
## If you use seed 123, you will always get this exact random data.
## if you use no seed, R creates a seed off of the clock of your computer.
## Good luck reproducing it.
set.seed(123)
NumVar1=round(rnorm(n=50,mean=1000,sd=50),digits=2) ## Normal distribution
set.seed(123)
NumVar2=round(runif(n=50,min=500,max=1500),digits=2) ## Uniform distribution
set.seed(123)
NumVar3=round(rexp(n=50,rate=.001)) ## Exponential distribution
NumVar4=2001:2050

simData=data.frame(FacVar1,FacVar2,FacVar3,NumVar1,NumVar2,NumVar3,NumVar4)
```

Each of the factor variables is representative of something that cannot be readily quantified, such as male and female, blonde brunette or redhead, or something along the lines of North South East and West. We will use these in our model, for simplicity.

Now that we have our randomly generated data, lets mess with it.

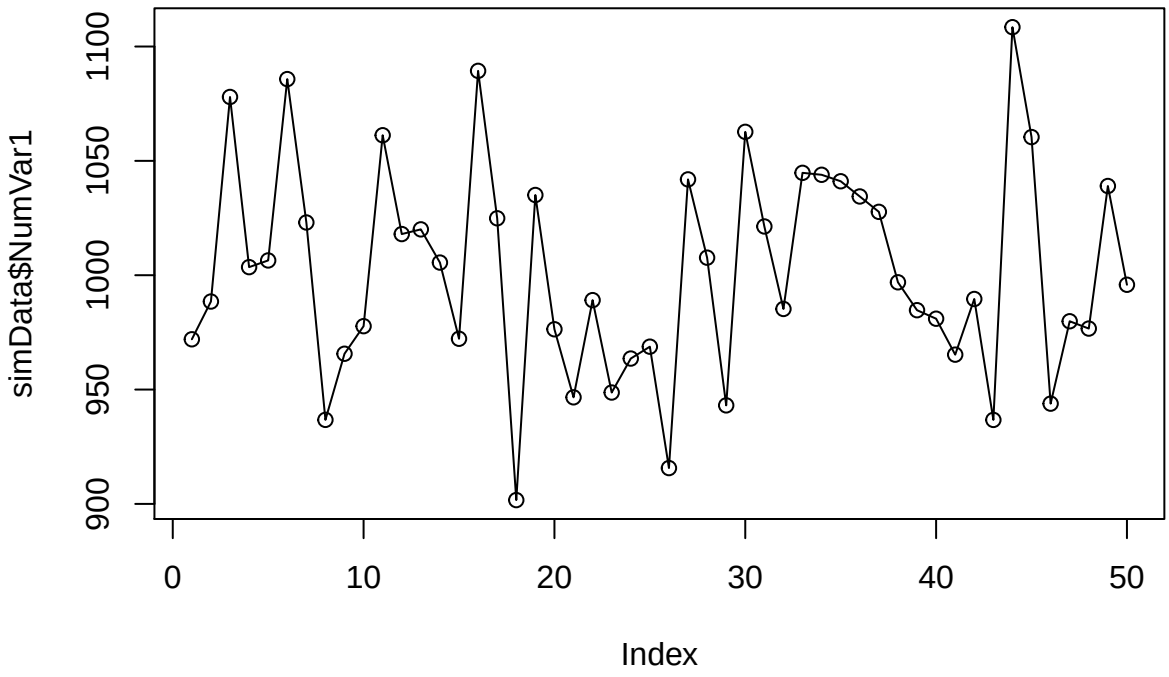
Plotting a Single Numeric Variable

While plotting a single variable may seem a bit pointless, there is lots of information we can recieve from the following graphs.

Index

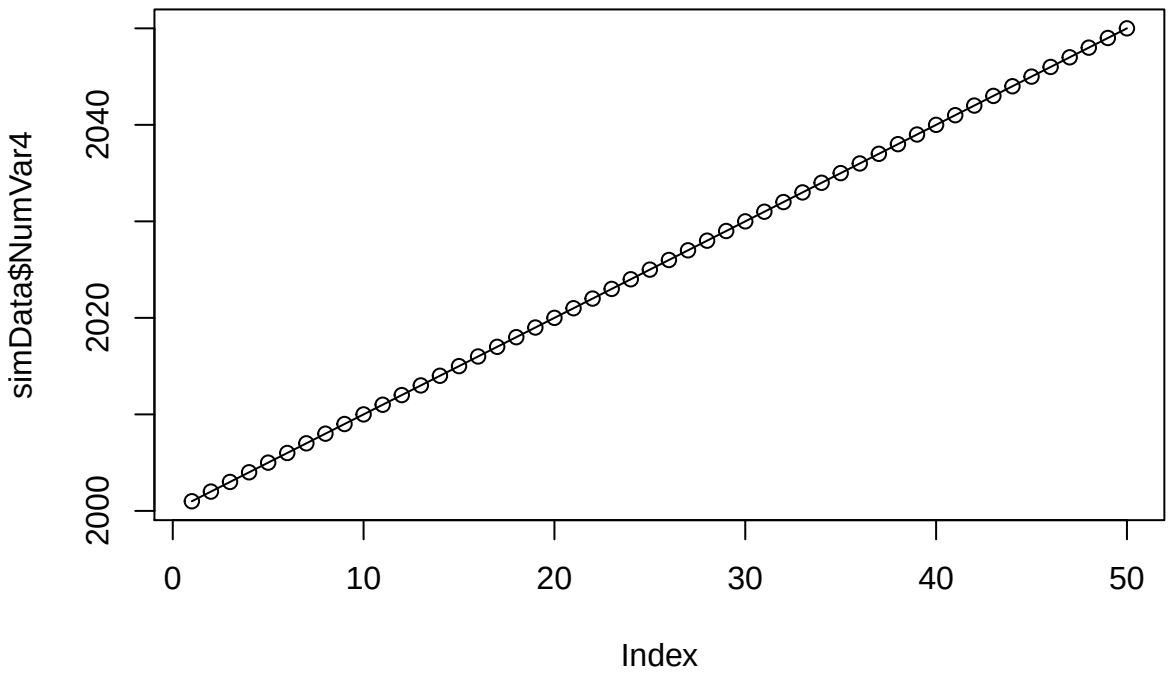
An Index plot is just plotting the points generated from R, using the number of points as the X axis. The first data point is the first number generated, the second point number two, etc. This can be helpful by quickly seeing if there is any autocorrelation in your data, or any data that is basing itself off of data from the previous period. There will be an obvious pattern if you have Autocorrelation.

```
plot(simData$NumVar1,type="o") ## Index plot
```



Here, we use NumVar1. We could use any of the 4 variables, and it would make a graph of any of those variables. Type o is a certain type of chart, specifically it adds the line to the graph. There are several types of plots; you can find them by typing "?plot" into the console.

```
plot(simData$NumVar4, type="o")
```

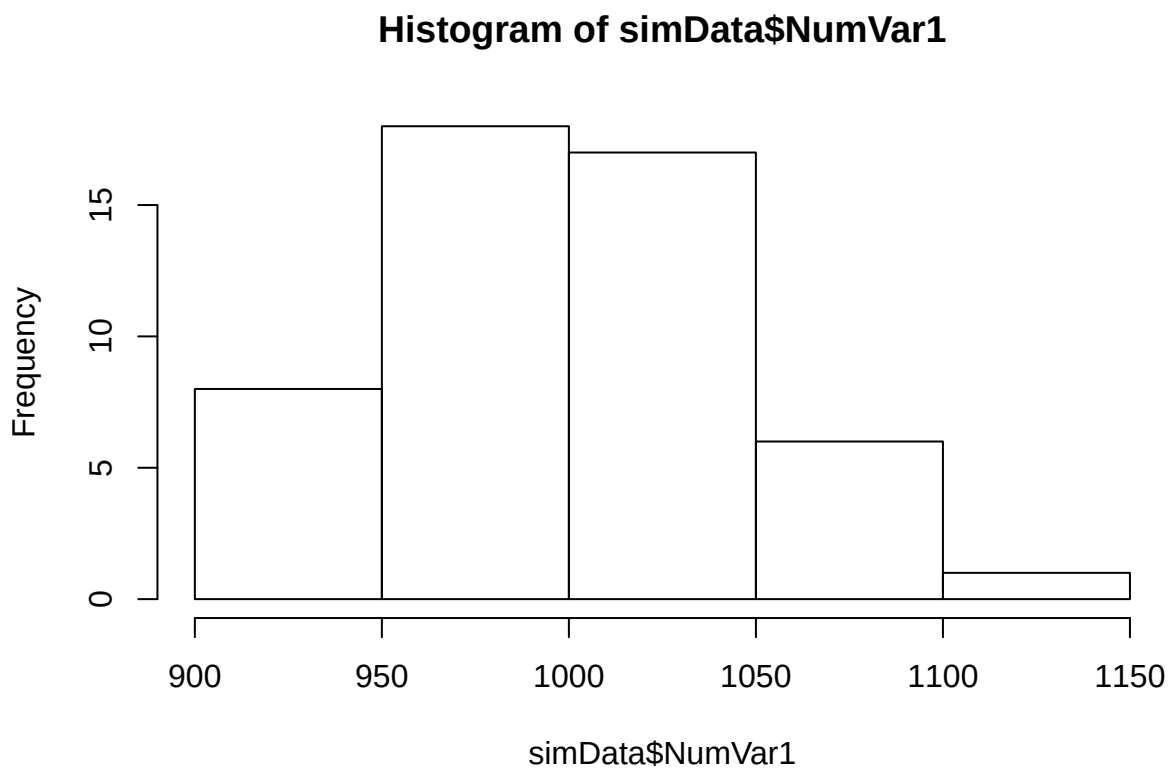


This is an example of autocorrelation. This is using the set NumVar4, which is a data set that goes from 2000-2050, increasing by one per point. Basically, X is equal to X from the previous period, plus one. Compared to the previous graph, there is a very obvious pattern, which is known as autocorrelation in time series data, which is what NumVar4 is used for later in our guide.

Histogram

Histograms are to Quantitative Data what Bar Charts are to Qualitative Data.

```
hist(simData$NumVar1) ## Histogram
```

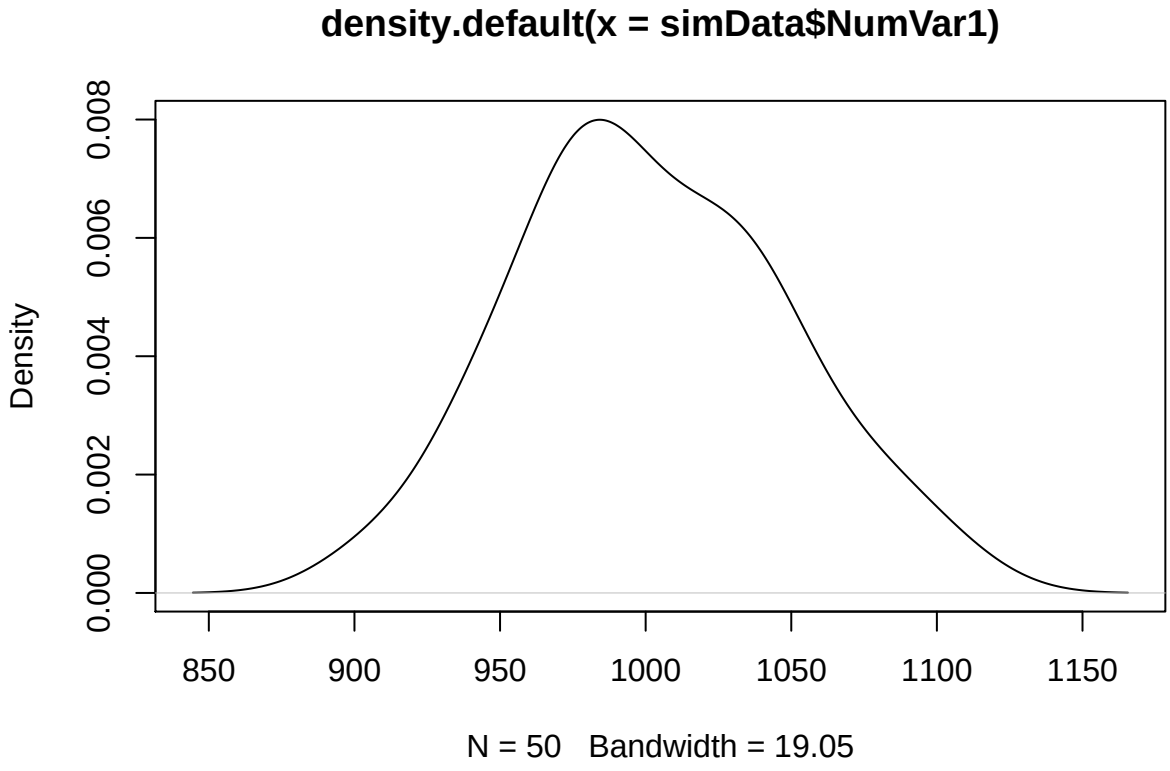


In this example, for every range of 50 points on the X axis, there are a certain number of points that have been randomly generated within those ranges, and the graph tracks that frequency on the Y axis. Histograms are very useful. For example, if this was sale values of the same good at different auctions, we can see that the most popular price for the sale is between 950 and 1000, so we should use this information as a basis for a price for our good.

Kernel Density Plot

Density Plots show the density of the distribution to each individual number. For example, at 1000, there are ~.008 points of data. Since there are only 50 points of data, and a range of over 300 in the data, the density is quite low; if we had 500, or 5000 data points, the density would increase, as well as the information gained form the graph. This graph is great to see if there are any little spikes at certain sales values, such as a spike at even numbers, or odd, or every ten dollars, etc.

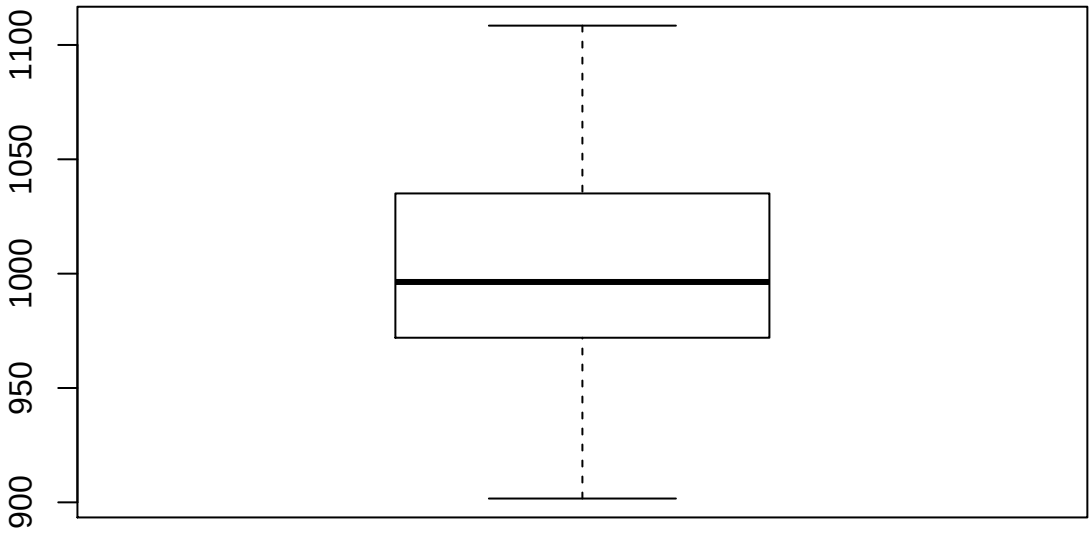
```
plot(density(simData$NumVar1)) ## Kernel density plot
```



Box Plot

A Box Plot is a plot that shows the distribution of the data along the range of numbers. From the bottom line to the bottom of the box, that is the lowest quartile of points. From the bottom of the box to the thick black line is the second quartile, and the Black Line is the median. Black line to top of box is the third quartile, and from top of box to top line is the fourth quartile.
Occasionally, there will be dots above (below) your top (bottom) line. These are outliers, and R determines those as outliers mathematically (also known as magic).

```
boxplot(simData$NumVar1) ## box plot
```



This sort of graph can be used to figure out where the middle 50% of your sales are, as well as the median and the quartiles.

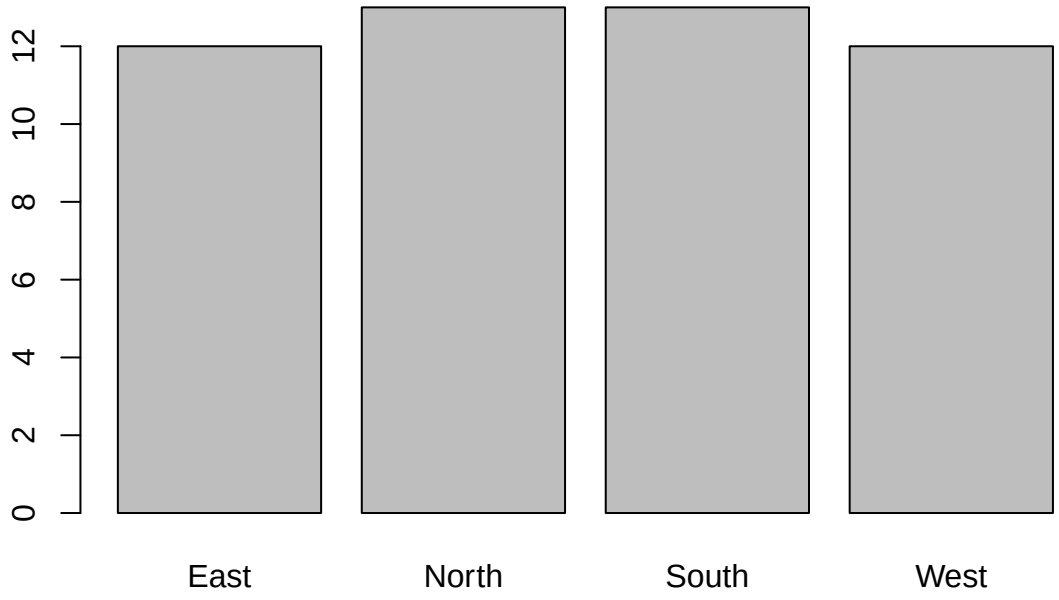
Plotting a Single Factor Variable

Plotting one qualitative variable basically becomes a frequency graph, or how many times each of the factor variables shows up. This can be good for many things, a major one being "which one of my products is the most popular." Lets look at a few

Bar Plot

This bar plot shows the number of times each level shows up in Factor Variable 3.

```
plot(simData$FacVar3) ## Bar Plot
```

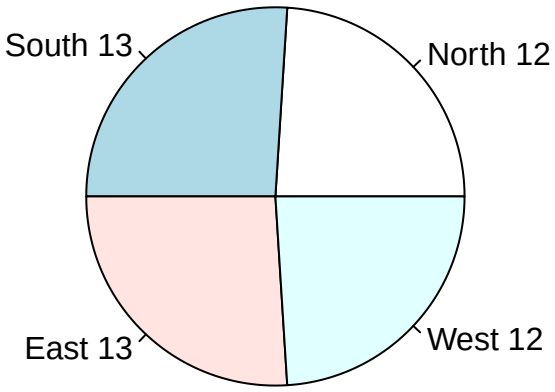


This is the classic example of the bar chart. We have our four variables, North, South, East, and West, and they have bars along the x axis, corresponding to the amount of times they show up, which is shown on the Y axis.

Pie Chart

This shows the exact same information as the chart above. However, since the human brain is much better at determining length instead of area, it is harder to determine which slice of the pie is biggest. This terrible malady of the human brain also leads to countless fights among children over who got the biggest slice of Caramel Apple Pie.

```
## The only good pie is Caramel Apple, and Pumpkin between October 15th and December 31st. pie chart - I
counts=table(simData$FacVar3) ## get counts
labs=paste(simData$FacVar3,counts)## create labels
pie(counts,labels=labs) ## plot
```



Ew. . .

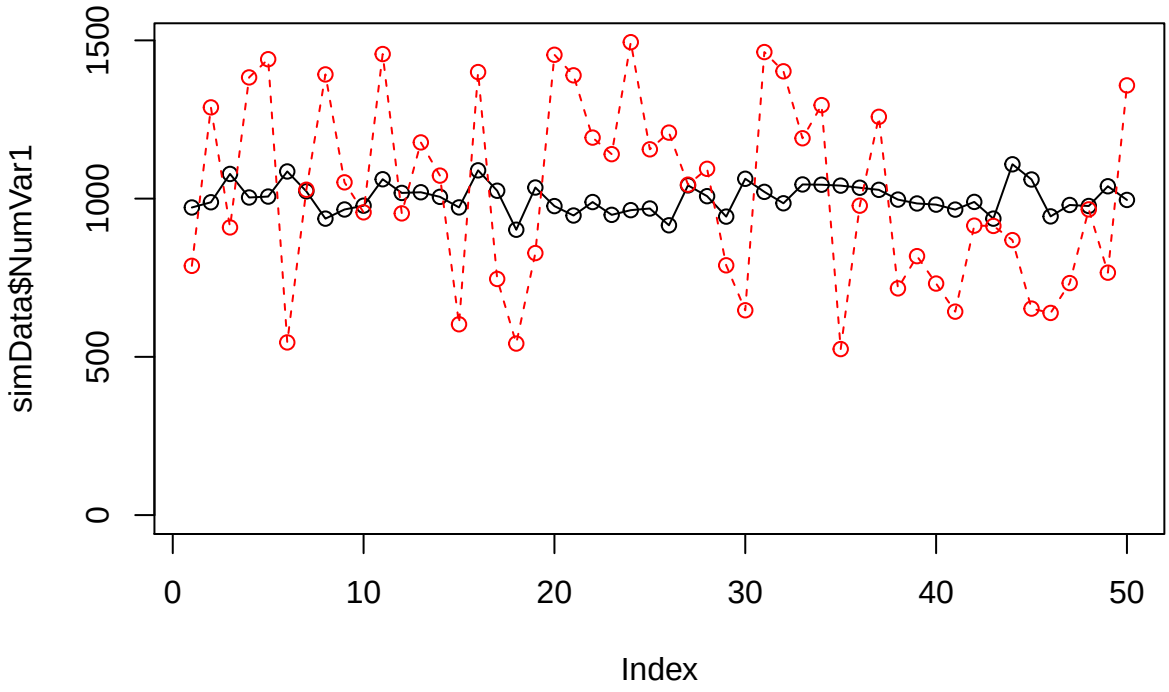
Two Variables: Numeric

Comparing two variables is a major reason for graphs over a data set; it is much easier to comprehend the relationship between two variables in picture form than in number and word form. The idea of pictures being better than words is also the main reason why I prefer coloring to text books.

Index

Just like last time, we are plotting the points generated by R, corresponding to the number in which it was generated. The black line is the same as the first line, NumVar1; the red line is NUmVar2

```
plot(simData$NumVar1,type="o",ylim=c(0,max(simData$NumVar1,simData$NumVar2)))## index plot with one vari
lines(simData$NumVar2,type="o",lty=2,col="red")## add another variable
```

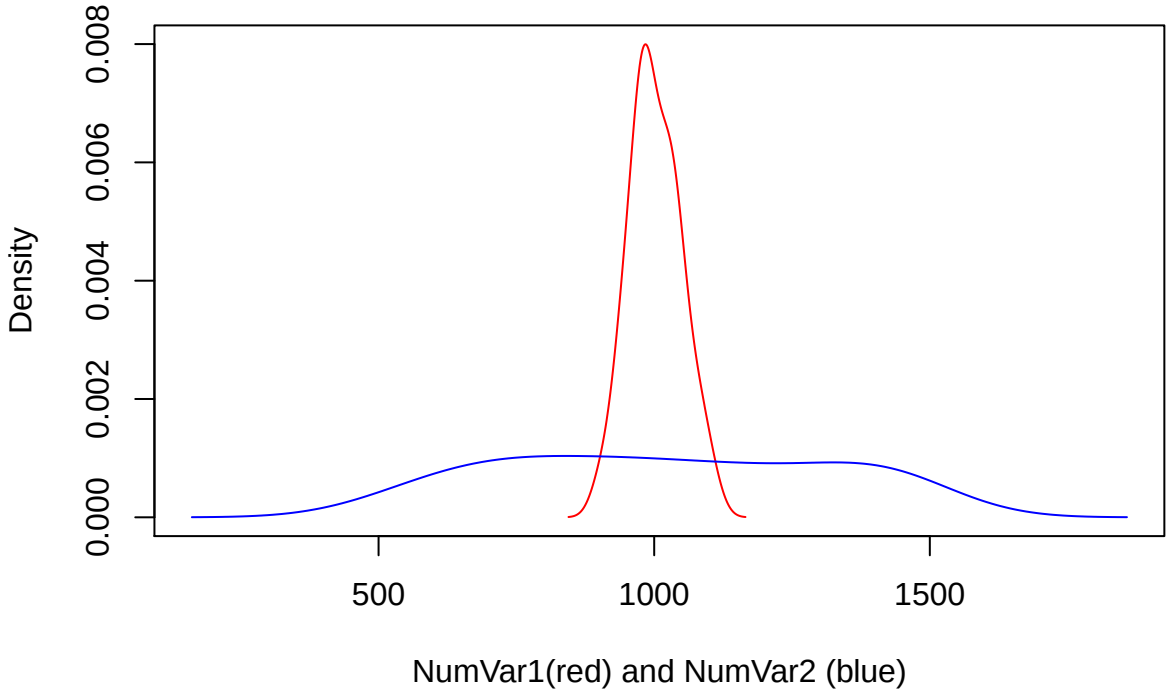


As you can see, what looked like major swings in NumVar1 in the single variable graph looks like minor swings in the second graph compared to NumVar2. This brings up a major point in data visualization. Instead of NumVar2 being a range from ~500 to ~1500, what if it were a range from ~500 to ~15000? Then the line for NumVar2 would basically be a solid black line along the bottom of the graph.
This shows the importance of having similar ranges in your graphs. if NumVar2 had values up to 15000, and was on the same graph as NumVar1, it would make the data for NumVar1, which may be vastly important, seem insignificant to NumVar2. It is important that the data is able to be read, even if it means making 2 graphs instead of one.

Density Plots

This is repeating what we did in the single variable density plot, but adding a second variable.

```
## Let's draw density plots : https://stat.ethz.ch/pipermail/r-help/2006-August/111865.html
dv1=density(simData$NumVar1)
dv2=density(simData$NumVar2)
plot(range(dv1$x, dv2$x),range(dv1$y, dv2$y), type = "n", xlab = "NumVar1(red) and NumVar2 (blue)",
     ylab = "Density")
lines(dv1, col = "red")
lines(dv2, col = "blue")
```

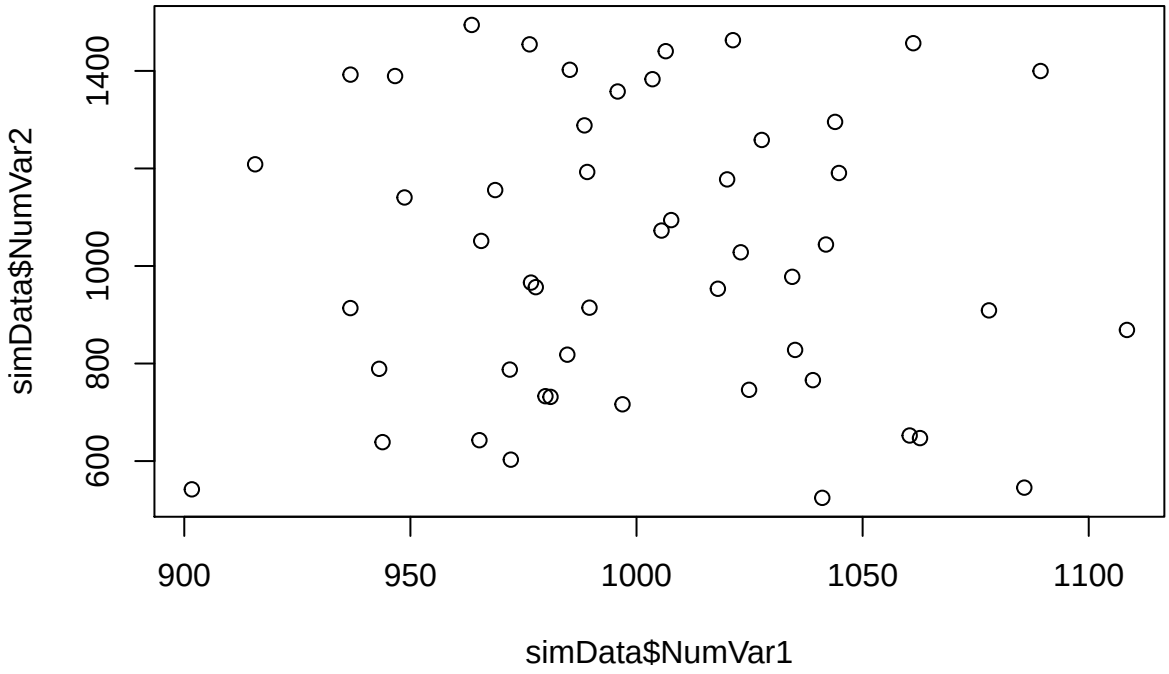


The same malady that plagued the previous graph plagues this one as well. Since the range for the blue line, NumVar2, is much larger than NumVar1, the line looks very flat and it is hard to determine any real information of value from it. As well, the line for NumVar1 is now so steep compared to the initial graph, it is also hard to determine much information other than "It gets dense around 1000." Once again, having two graphs in this case would be a great benefit for the transfer of quality information.

Scatterplots

Scatterplots are the main graph used in two numeric variable graphs. Since each of these points has two different numbers for each variable, we plot one of them, NumVar1, on the X axis, and the other, NumVar2, on the Y axis. This type of graph is helpful to see if there is any pattern to the data, or any correlation to the data.
Since this is two randomly generated data sets, there should be no correlation.

```
## scatterplots
plot(simData$NumVar1,simData$NumVar2)
```



Thankfully, there is not. That would be one heck of a coincidence.

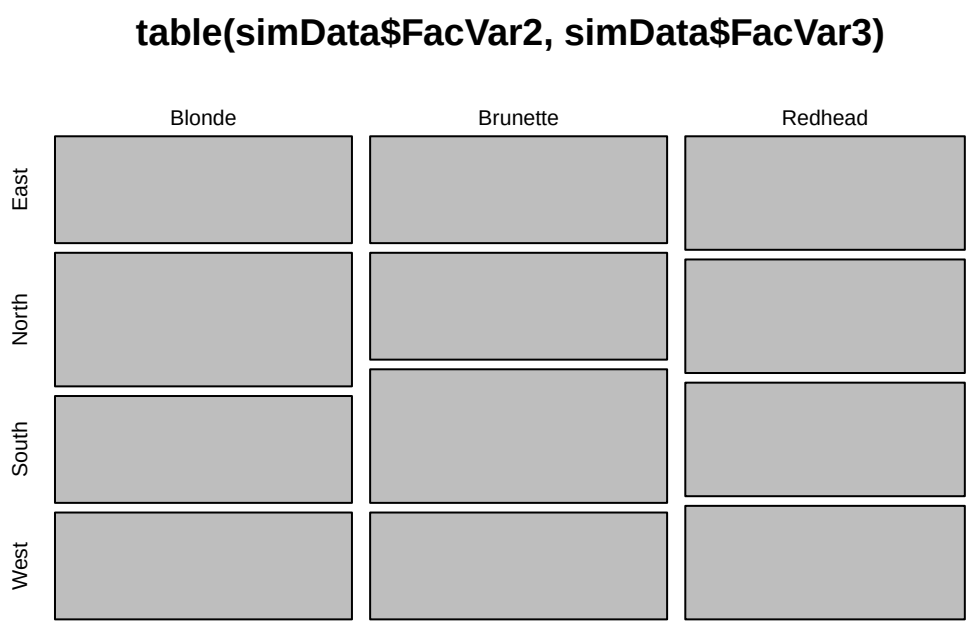
Two Variables: Factor

These graphs show the relationship between two Qualitative sets of data.

Mosaic

This type of graph shows the relationship between each hair color with each geographical area. Since the relationship is quantified in terms of the total area of the boxes, it is difficult to see which box is bigger than any other box.

```
## Mosaic plot
plot(table(simData$FacVar2,simData$FacVar3))
```

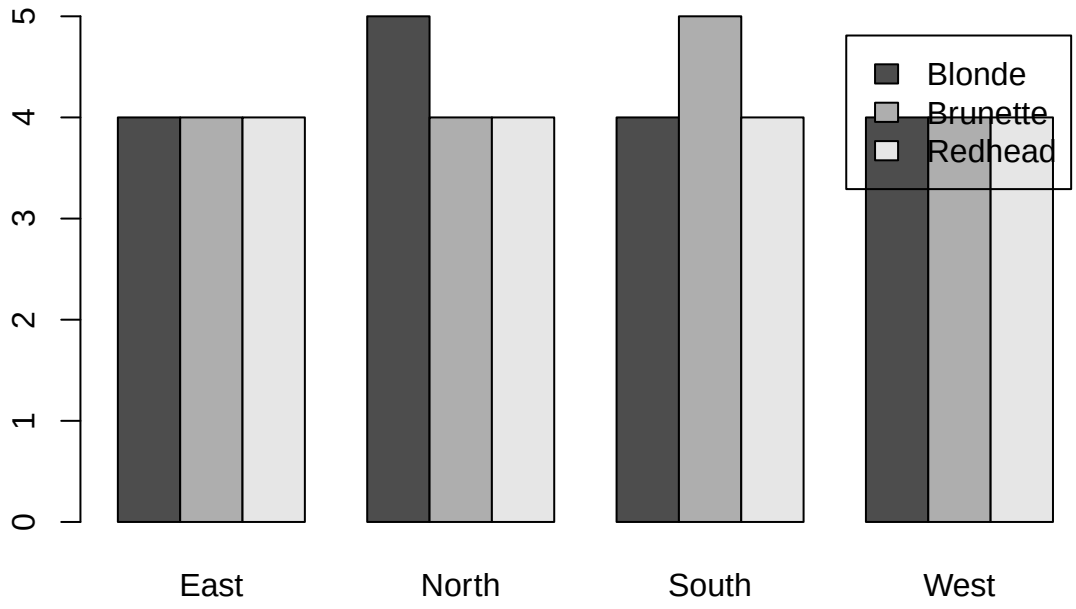


I am going to refer to this as a brownie chart, since it kind of looks like a brownie pan, and it is no more than a glorified pir chart. I'm also really into dessert. It also suffers from the major downfall of the pie chart, in that it uses area as it major determinate of size, which we have already talk about being awful. Unlike brownies.

Bar Plots

Bar plots are by far the best way to show a relationship between two Qualitative variables. In this bar plot, geographical area is shown on the X axis, hair color is each individual bar, and the Y axis is the number of people of each hair color in each geographical area.

```
## barplots
bartable=table(simData$FacVar2,simData$FacVar3) ## get the cross tab
barplot(bartable,beside=TRUE, legend=levels(unique(simData$FacVar2))) ## plot
```

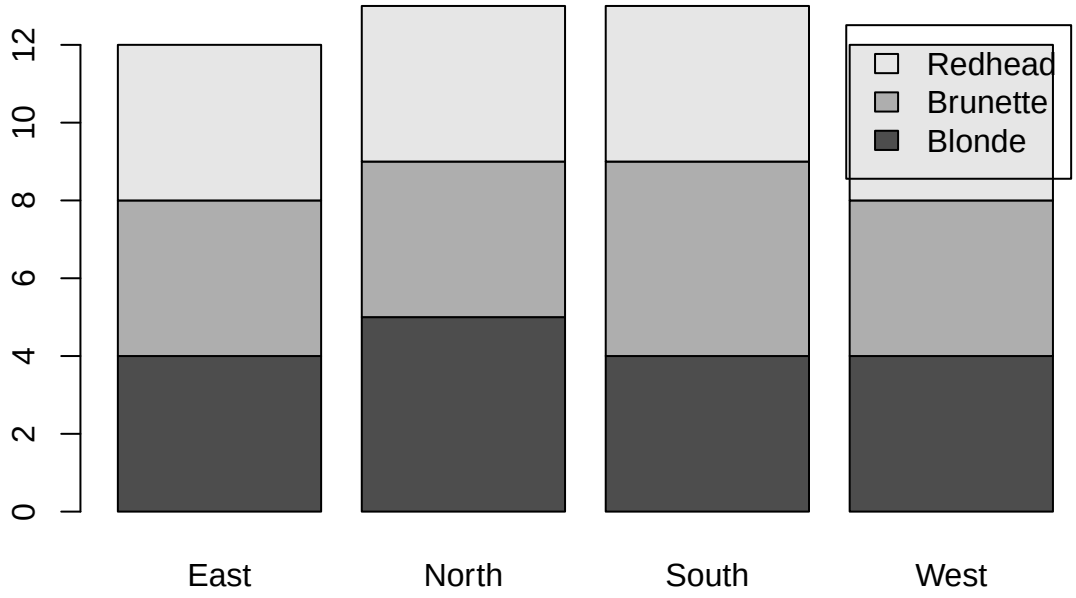


This is a much better graph than the one before it. We can easily determine the hair colors and which bar correlates to each color, which geographical location they are in, and most importantly, the number of each hair color in each area, which is the major downfall of the previous graph.

Stacked Bar Plots

Stacked bar plots show the same information as normal bar plots, but stacked. This is a great way to visualize both which geographical area has the most points, as well as the composition of hair color in each geographical area. However, it is not as descriptive, and it is somewhat hard to determine what portion is larger or smaller than another, so it should not be used to show specific data without labels.

```
barplot(bartable, legend=levels(unique(simData$FacVar2))) ## stacked
```

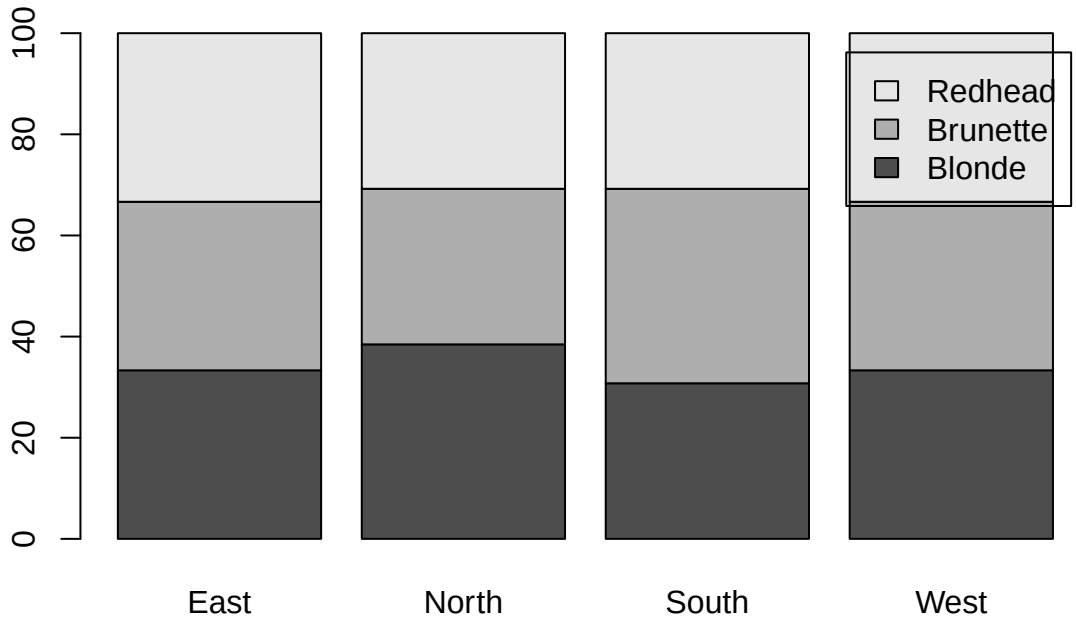


This graph now becomes a dual use graph, where we see both the overall values, as well as the composition of those values. In certain cases, dual use graphs can be extremely valuable, but with more information given, there is more information to then confuse. Use extra caution in using any dual use graph.

Stacked Percentage

This stacked bar plot shows what percentage of hair colors are in each geographic area. Without labels, this is a difficult graph to decipher specific percentages, so should not be used to show specific data.

```
barplot(prop.table(bartable,2)*100, legend=levels(unique(simData$FacVar2))) ## stacked 100%
```



Same idea as the last graph, be careful with dual use graphs.

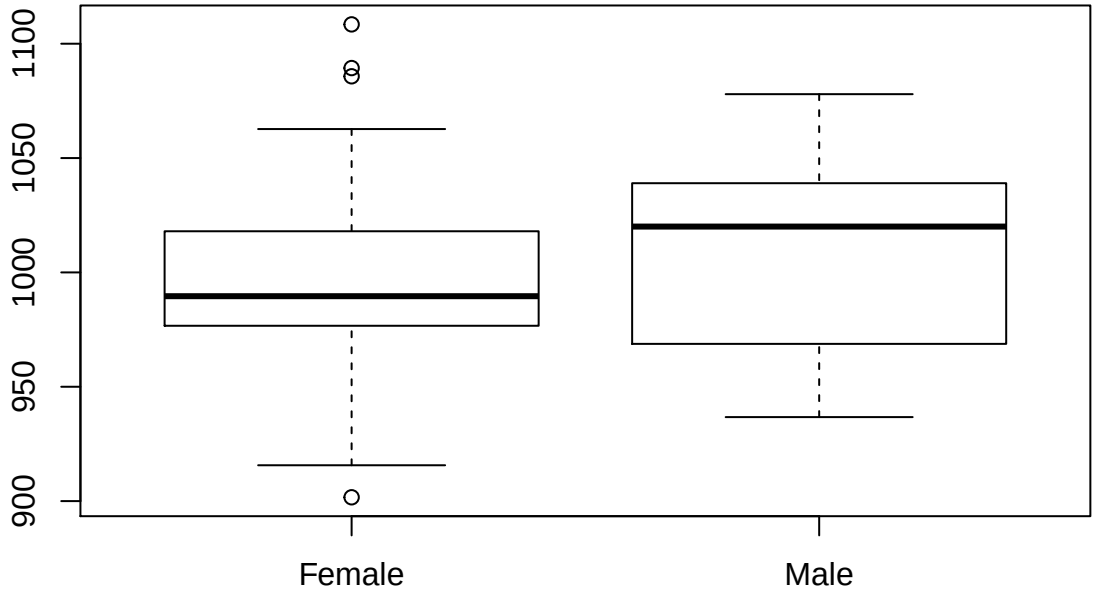
Two Variables: One Factor, One Numeric

Finding correlation between quantitative and qualitative data is a very important part of any study. For our example, any study using both men and women will want to have some sort of distinction in their findings if there is a difference in results between the two. An easy way to determine so is by plotting the quantitative data against the qualitative.

Box Plots

This box plot uses the same ideas as the one in the single numeric variable. Now, it uses the range of results in NumVar1 as the Y axis, and uses the variables of Female and Male on the X axis.

```
## Box plots for the numeric var over the levels of the factor var
plot(simData$FacVar1,simData$NumVar1)
```



As you can see, there are now outliers! Cool! Females now have one low outlier, and three high outliers. While being an outlier may sound like a bad thing, it does not necessarily mean so, or that your data is bad. The program has just determined that the graph is more accurate to your data as a whole, if these few points are thrown out of the equation. This leads to a truer representation of your data. This chart may be used to determine the dollar value of individual sales to a specific gender of customer, which can help in targeting your products to a specific group of people in an effort to maximize profits.

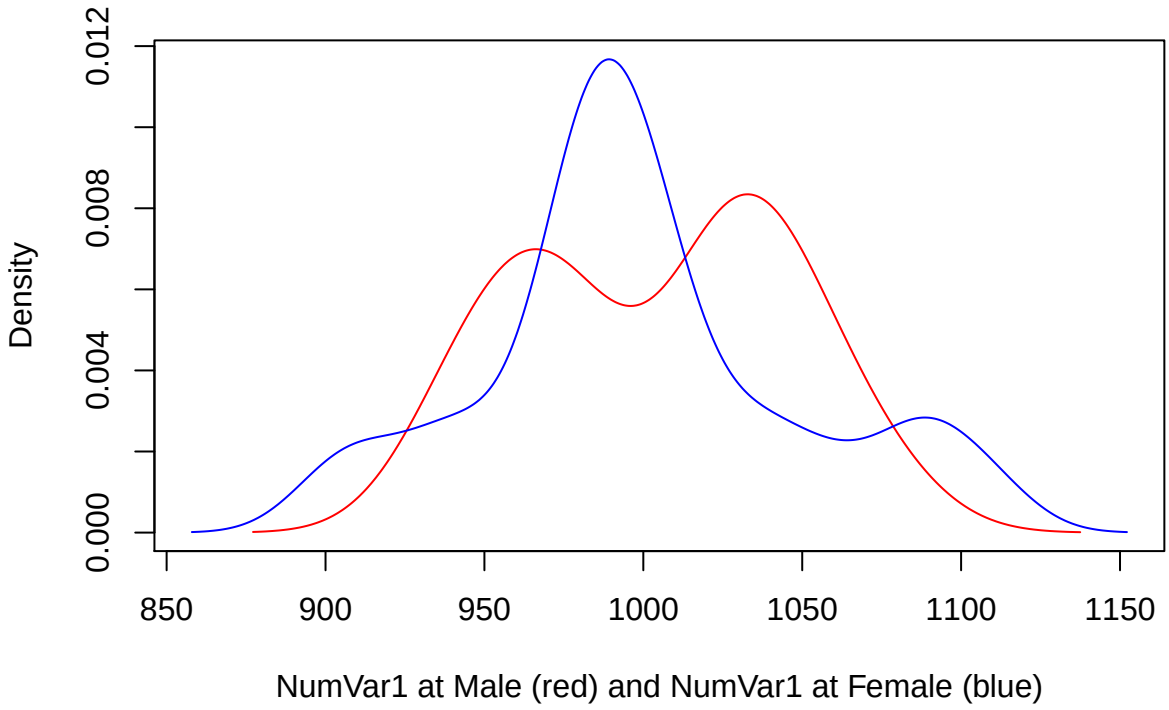
Density Chart

The Density Chart shown here is the same as the one shown in the one numeric variable chart; now, it splits the data between Male and Female, shown in red and blue respectively.

```
## density plot of numeric var across multiple levels of the factor var
Male=simData[simData$FacVar1=="Male",]
Female=simData[simData$FacVar1=="Female",]

dv3=density(Male$NumVar1)
dv4=density(Female$NumVar1)

plot(range(dv3$x, dv4$x),range(dv3$y, dv4$y), type = "n", xlab = "NumVar1 at Male (red) and NumVar1 at F
lines(dv3, col = "red")
lines(dv4, col = "blue")
```



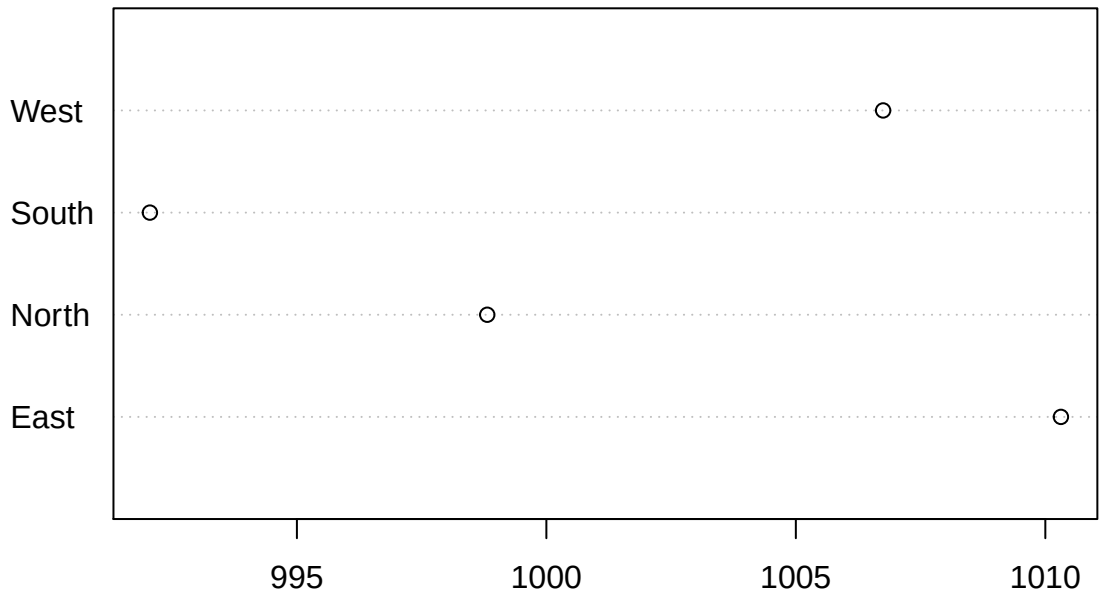
With a clearer picture of how this data is distributed, we can determine several things. For example, in our randomly selected data, males frequently hover around 1000, while there is a dip there for females, who see increases at 950 and 1050. This information could be used to increase sales, showing that men like paying in even numbers, a clean 1000, while perhaps a $50 coupon would help increase sales among women.

Mean Value of Numeric Over Factor

This chart shows what the mean value of the numeric variable is in each of the four geographical regions.

```r
## Mean of one numeric var over levels of one factor var
meanagg=aggregate(simData$NumVar1, list(simData$FacVar3), mean)

dotchart(meanagg$x,labels=meanagg$Group.1) ## Dot Chart
```

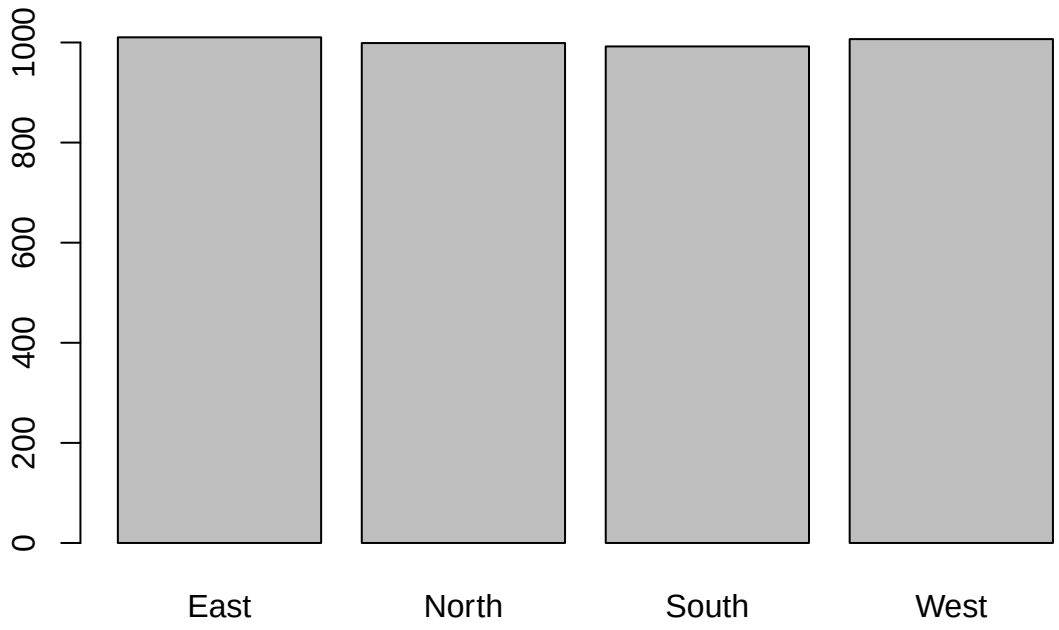


This can be used to easily compare and contrast the mean values in different areas, among different genders, hair colors, or whatever other qualitaitve variable you have.

Bar Plot

This graph shows the same data as the previous one, but in bar graph form.

```
barplot(meanagg$x,names.arg=meanagg$Group.1)## Bar plot
```



Question: Is a bar plot even appropriate when displaying a mean— a point?

Excellent question, hypothetical student in the front row! Looking at these two graphs, which one is easier to get information from? Which is more accurate? Which level in the Bar Graph has the highest mean? The lowest?
Bar Graphs are great for many things; this is not one of them.First off, using bar charts, one should always use zero where the axes meet. In this case, when we are looking at little differences between values in the thousands, this graph does not really help us. Secondly, using Tufte's theory, use as little ink as possible that is not necessary. With that philosophy, it is much better to plot the points instead of using the bar graphs.

Three Variables: Three Factor

The more the merrier as far as I'm concerned!

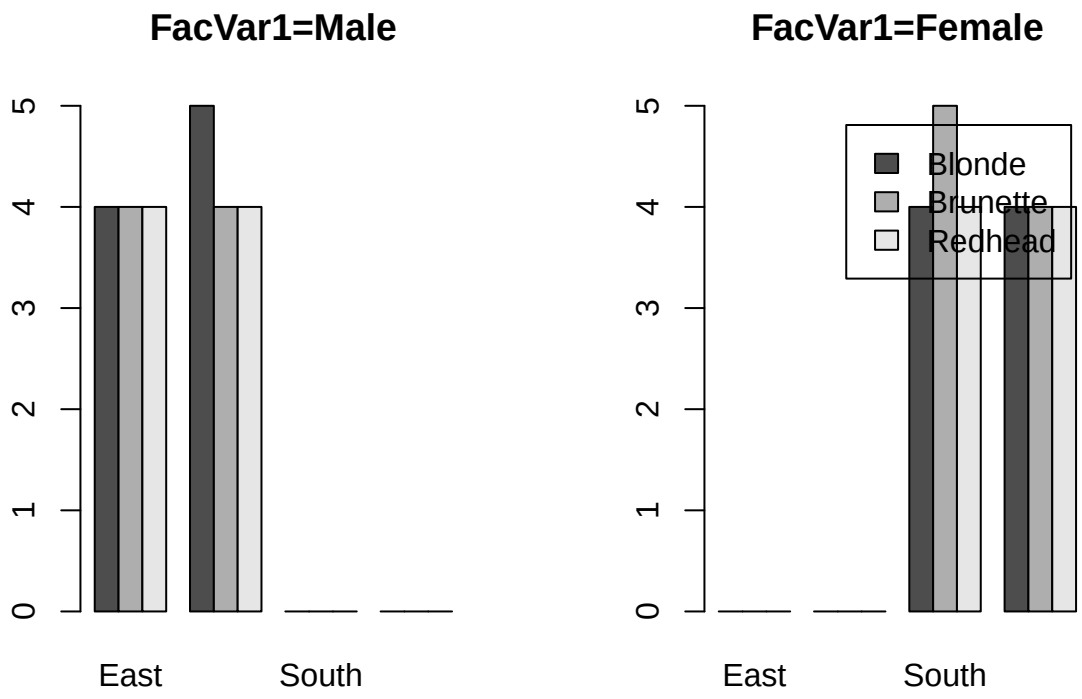
Double Bar

Double bar plot has 2 bar plots, with geographic area as the X axis on both graphs, the bars themselves as hair color, and each individual graph is a a gender. Each bar is the number of points in both area and hair color, in either Male or Female.

```
par(mfrow=c(1,2))

bar1table=table(Male$FacVar2,Male$FacVar3)
barplot(bar1table,beside=TRUE, main="FacVar1=Male")

bar2table=table(Female$FacVar2,Female$FacVar3)
barplot(bar2table,beside=TRUE, main="FacVar1=Female", legend=levels(unique(Female$FacVar2)))
```



In this graph, we are just getting more descriptive in our analysis, seeing, for example, in which area blonde women buy our product, and how many orders by blonde women in this area there were. More descriptive can be very valuable, but can make things more confusing.

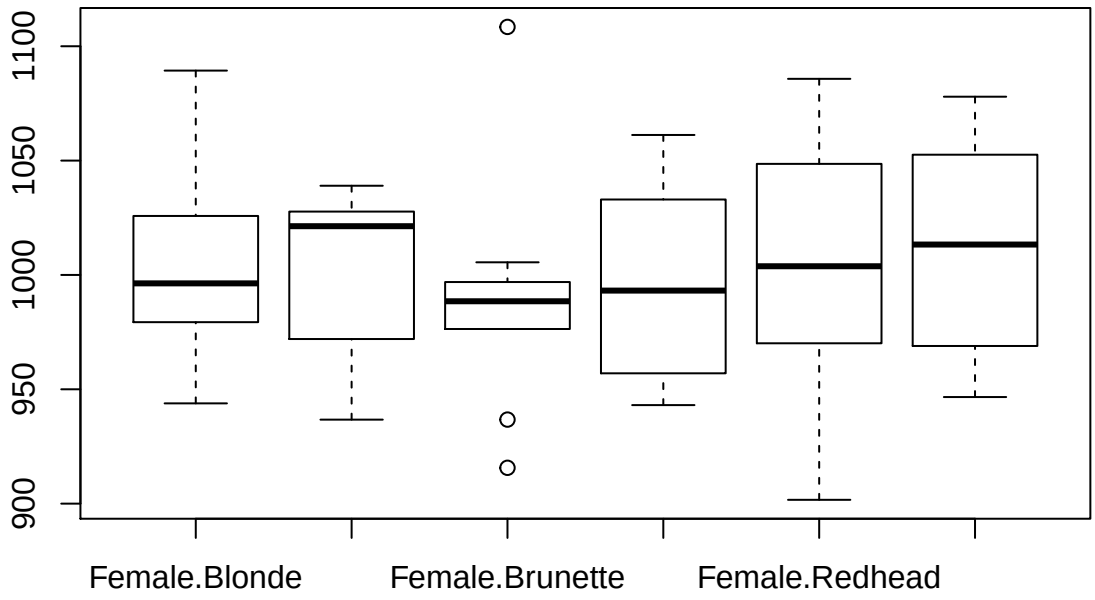
Three Variables: Two Factor, One Numeric

This allows us to compare two factor variables in relation to one numeric. For example, test scores among men and women, who have either blonde or brown hair. Alternatively, sales data in two different areas on weekends or during the week. There are many uses for the following graphs.

Box Plot

Same set up as before, but now each box correlates to a different factor variable in each of the numeric variables.

```
par(mfrow=c(1,1))
## boxplot of NumVar1 over an interaction of 6 levels of the combination of FacVar1 and FacVar2
boxplot(NumVar1~interaction(FacVar1,FacVar2),data=simData)
```



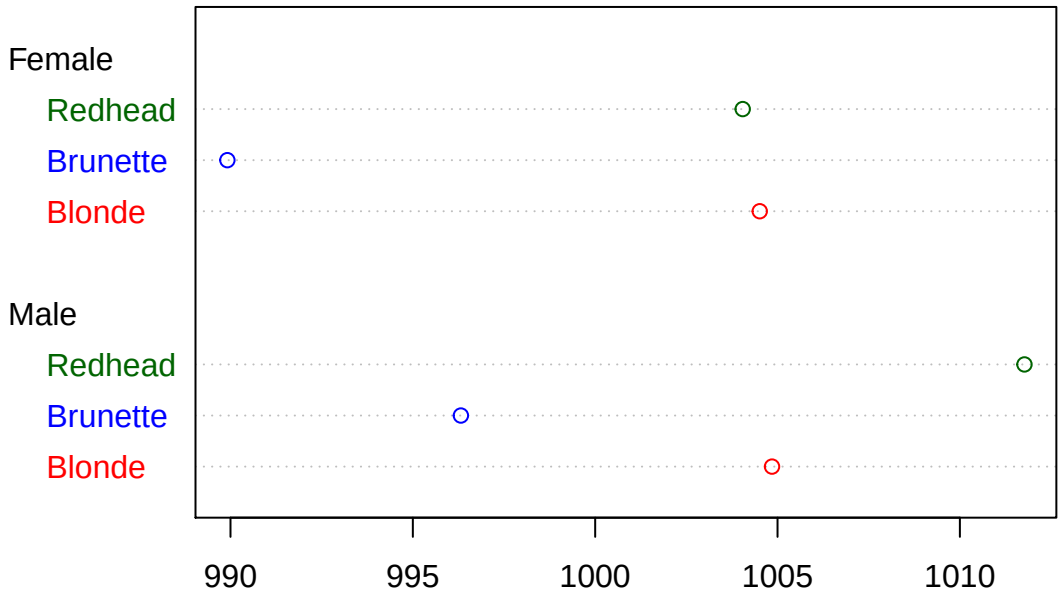
To make this slightly easier to read, it goes blonde females, then males, then brunette females, brunette males, etc. Now, we find that our outliers belong exclusively to brunette women in our randomly generated example.

Mean Plot

Mean plot, but now we have 2 factor variables, so 2 different graphs of means here in one graph.

```
## Mean of 1 Numeric over levels of two factor vars
meanaggg=aggregate(simData$NumVar1, list(simData$FacVar1,simData$FacVar2), mean)
meanaggg=meanaggg[order(meanaggg$Group.1),]
meanaggg$color[meanaggg$Group.2=="Blonde"] = "red"
meanaggg$color[meanaggg$Group.2=="Brunette"] = "blue"
meanaggg$color[meanaggg$Group.2=="Redhead"] = "darkgreen"

dotchart(meanaggg$x,labels=meanaggg$Group.2, groups=meanaggg$Group.1,color=meanaggg$color) ## dotchart
```

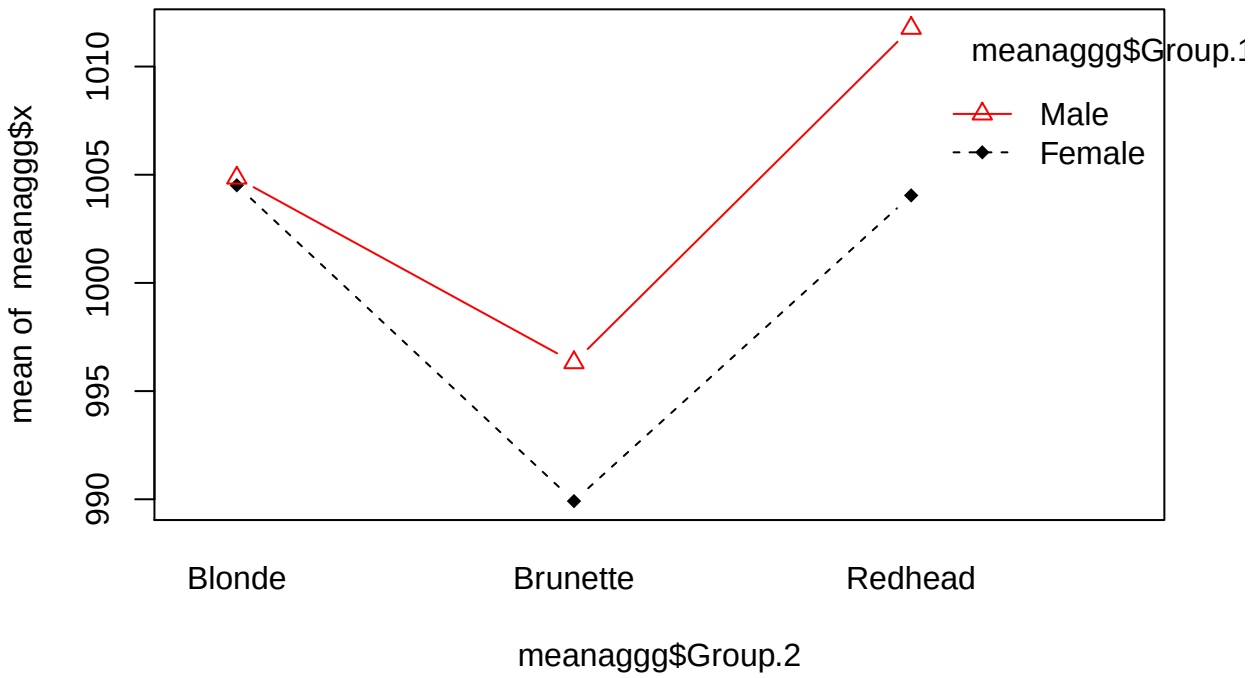


We can now compare the means of not only Males and Females, but by hair color too! How convenient!

Line Plot of Means

This is a line plot of the means, between each Factor variable sorted with each numeric variable.

```
interaction.plot(meanaggg$Group.2,meanaggg$Group.1,meanaggg$x,type="b", col=c(1:2),pch=c(18,24)) ## int
```



This shows the same information as the previous graph, but in line chart form. This would be valueable information to see a line if instead of hair color, we had monthly data or yearly data, to see a line of the rate of change in each month or year.

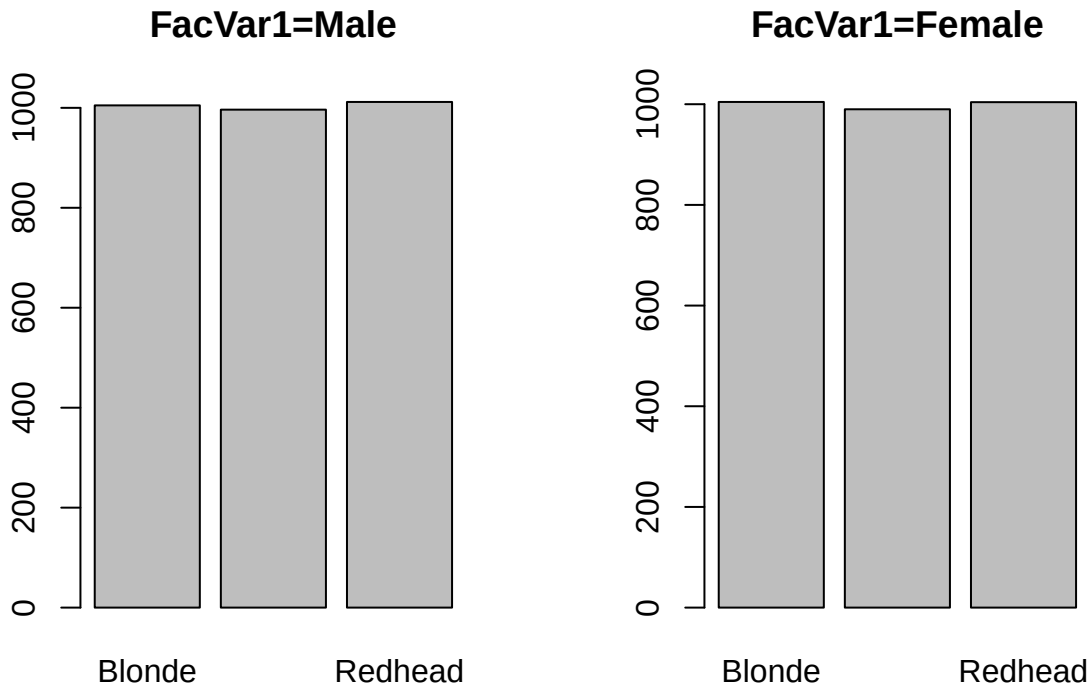
Dual Bar Plot

Once again, the means in a bar plot set up. Not nearly as informative as the mean chart or mean line chart.

```
## some a bar plot
par(mfrow=c(1,2))

Male=meanaggg[meanaggg$Group.1=="Male",]
Female=meanaggg[meanaggg$Group.1=="Female",]

barplot(Male$x,names.arg=Male$Group.2, main="FacVar1=Male")
barplot(Female$x,names.arg=Female$Group.2, main="FacVar1=Female")
```



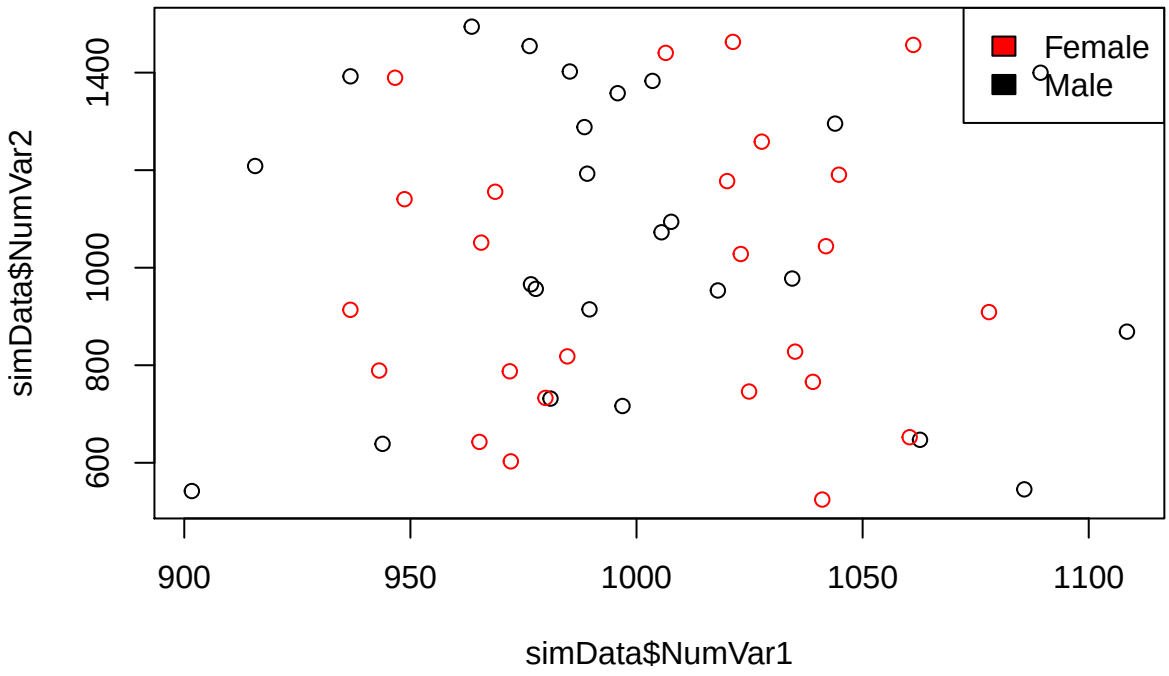
Three Variables: Two Numeric, One Factor

We like numbers, lets compare them with one factor! Yay!

Scatterplot

This scatterplot between NumVar1 and NumVar2 is the same graph as we saw earlier; now, we have two different colors showing the corresponding gender to each point.

```
## Scatter plot with color identifying the factor variable
par(mfrow=c(1,1))
plot(simData$NumVar1,simData$NumVar2, col=simData$FacVar1)
legend("topright",levels(simData$FacVar1),fill=simData$FacVar1)
```



This information can be very useful in determining if, while there not be a pattern in the data as a whole, if there is a pattern once we throw in gender, area, or hair color into the equation.

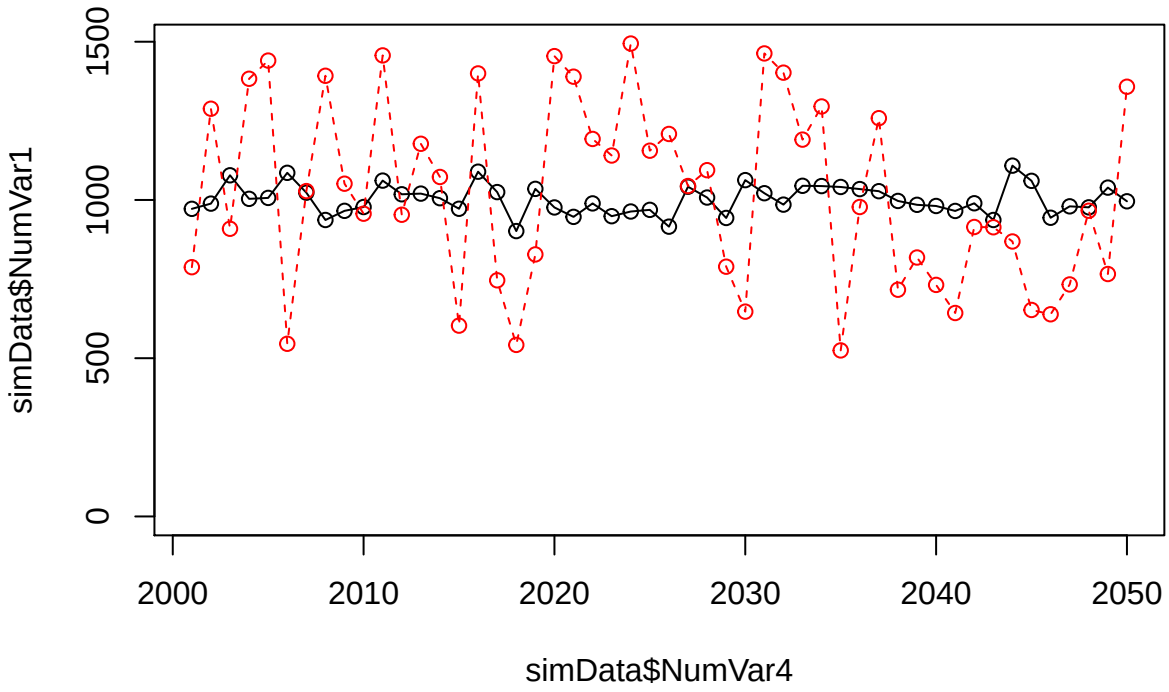
Three Variables: Three Numeric

Time Series

We are now using the graph we have in the two numeric variable section under index, and we have put NumVar4, or a time series set, against it, instead of just the points. Now, we have a year by year change in both NumVar1 and 2.

```
## NumVar4 is 2001 through 2050... possibly, a time variable - use that as the x-axis
plot(simData$NumVar4,simData$NumVar1,type="o",ylim=c(0,max(simData$NumVar1,simData$NumVar2)))## join do

lines(simData$NumVar4,simData$NumVar2,type="o",lty=2,col="red")## add another line
```

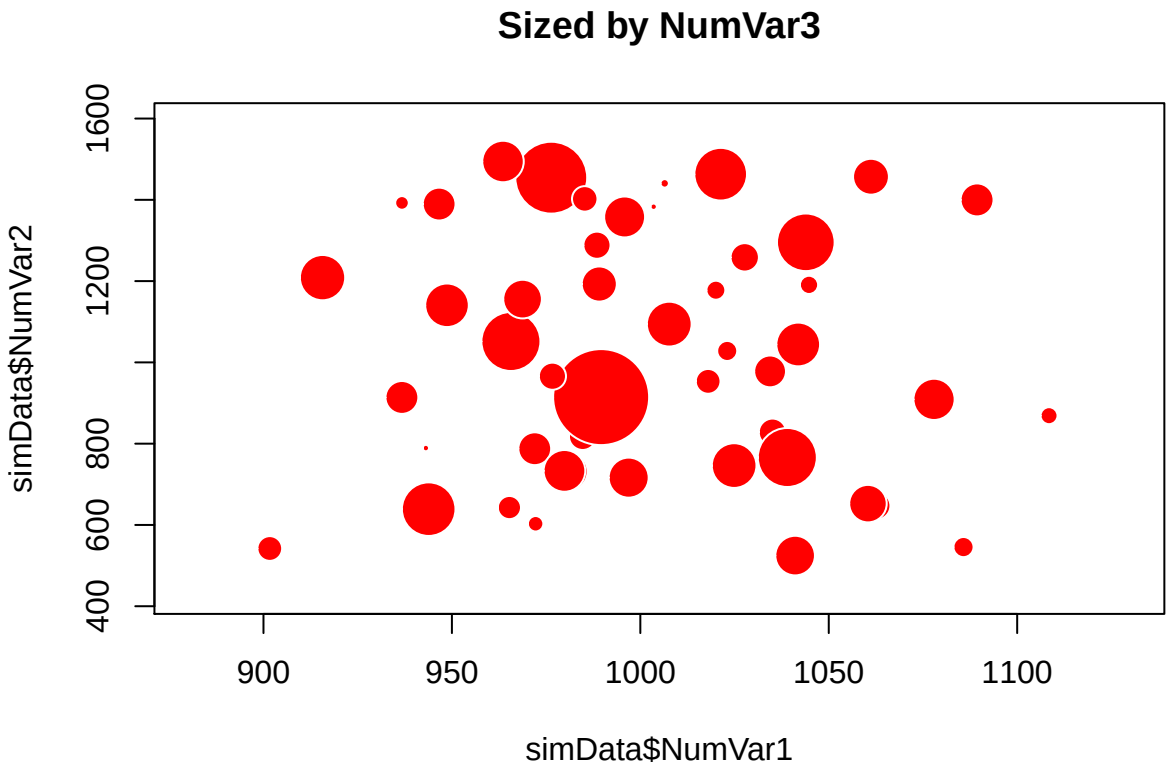


Time series data is incredibly valuable in seeing how your data changes as the months or years go by. If sales started to decline, when did this happen? Does it correspond to any change in company policy? Consumer preferences? Law changes? A great many things could be causing this change in the graph, but we can use time series data and graphs to figure out when things happen.

Bubble Plot

A bubble plot works by having the scatterplot of NumVar1 and 2, then adding the corresponding data from NumVar3 and changing the size of each of the points in the set. This makes it so we can relate the information from NumVar3 in the scatterplot correlation from NumVar1 and 2.

```
## Bubble plot - scatter plot of NumVar1 and NumVar2 with individual observations sized by NumVar3
# http://flowingdata.com/2010/11/23/how-to-make-bubble-charts/
radius <- sqrt( simData$NumVar3/ pi )
symbols(simData$NumVar1,simData$NumVar2,circles=radius, inches=.25,fg="white", bg="red", main="Sized by
```

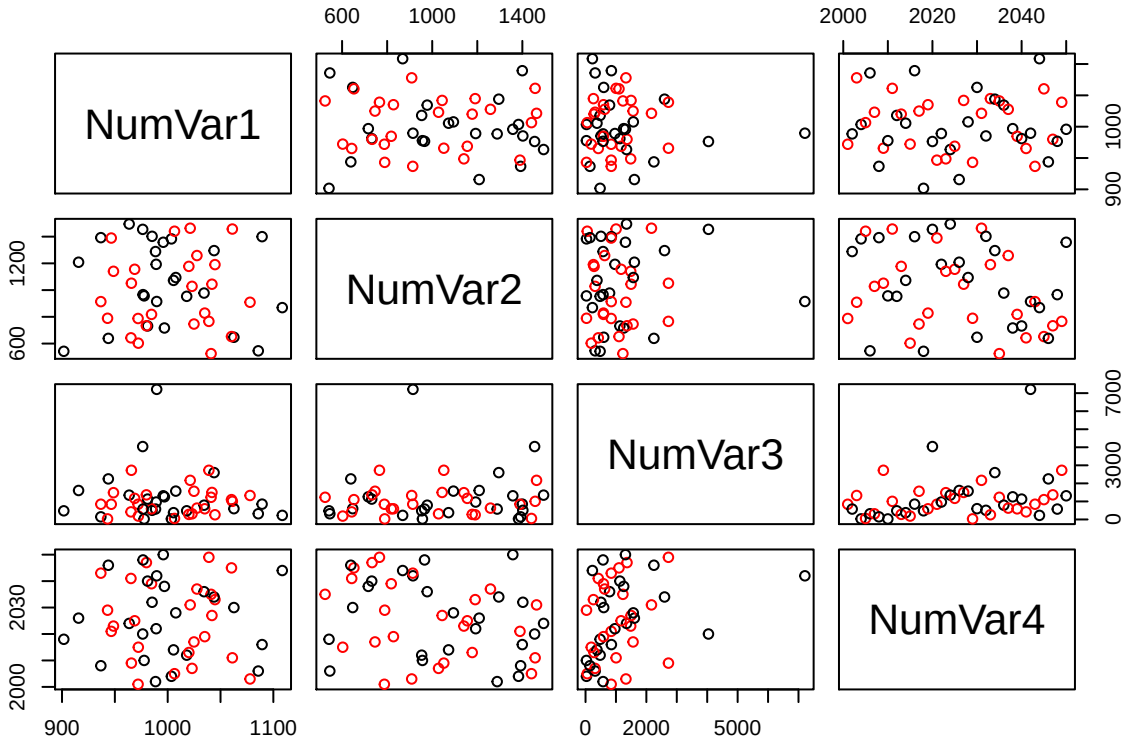


This information and this graph looks great, but once again, we cannot really differentiate between the sizes of the bubbles, so it makes it much harder for us to put a large practical use on this graph type.

Matrix Of Scatterplots

This is a matrix of all of the scatterplots between two of the numerical variables we have. It looks pretty cool, but unless you are comparing all of the graph in a particular row or column, it is hard to relate one scatterplot to the next.

```
pairs(simData[,4:7], col=simData$FacVar1)
```



Conclusion

There are many different types of graphs, to use with many different types of data. By using this guide, you now have at your fingertips many different types of graphs to use in any situation that may arise. From single numeric index charts, to a matrix of scatterplots to look at, you now know the pros and cons of many different graph types and how to use those graph types correctly. Use the information in this guide in the real world, and make graphs that look as fantastic as the data it is showing off.