Replication of Blog

Ali Walkley

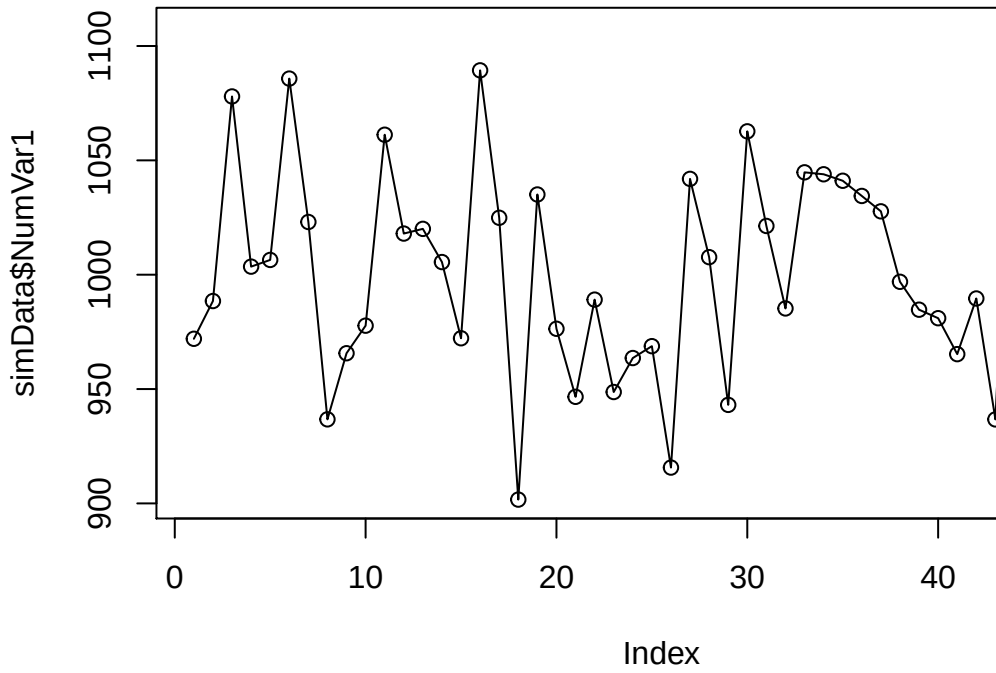
September 21, 2015

Base R Version

Data Let us begin by simulating our sample data of 3 factor variables and 4 numeric variables.

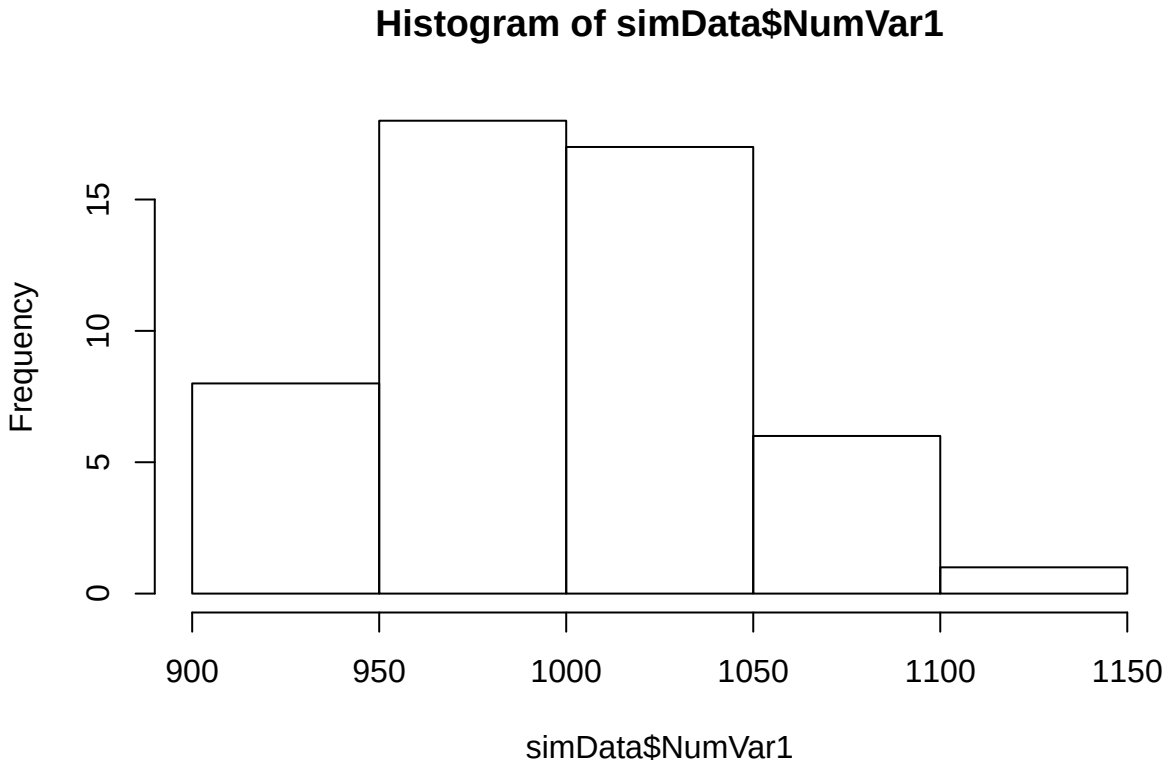
```
## Simulate some data

## 3 Factor Variables
## The function factor is used to encode a vector as a factor (the terms 'category' and 'enumerated typ
FacVar1=as.factor(rep(c("level1","level2"),25))
FacVar2=as.factor(rep(c("levelA","levelB","levelC"),17)[-51])
FacVar3=as.factor(rep(c("levelI","levelII","levelIII","levelIV"),13)[-c(51:52)])

## 4 Numeric Vars
##ceiling takes a single numeric argument x and returns a numeric vector containing the smallest intege
##.Random.seed is an integer vector, containing the random number generator
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

##This function creates data frames, tightly coupled collections of variables which share many of the p
simData=data.frame(FacVar1,FacVar2,FacVar3,NumVar1,NumVar2,NumVar3,NumVar4)
```

```
##creates a plot, histogram and density:For simple scatter plots, plot.default will be used. However, t
plot(simData$NumVar1,type="o") ## Index plot
```

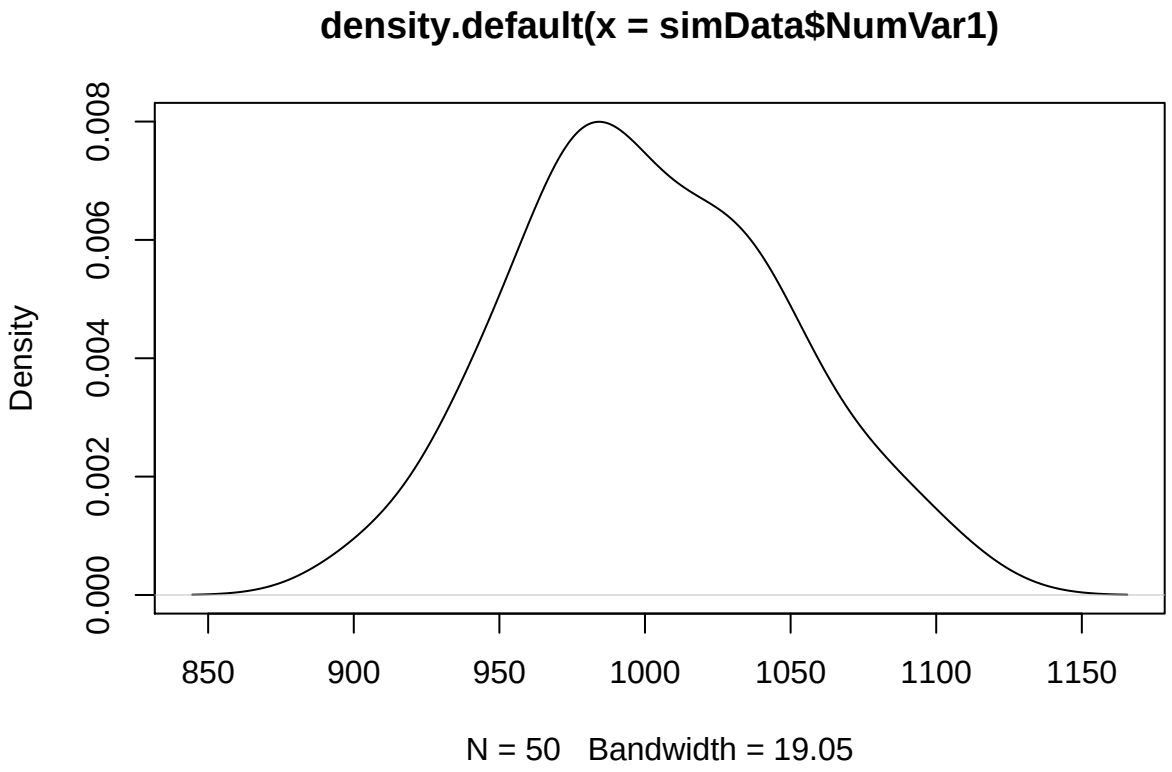


One Variable: Numeric Variable

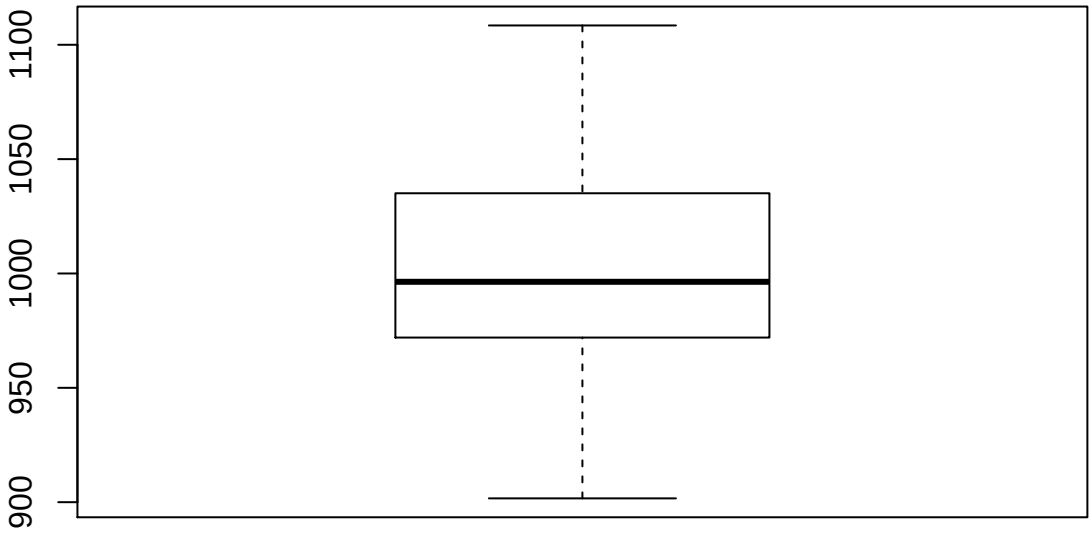
```
hist(simData$NumVar1) ## histogram
```



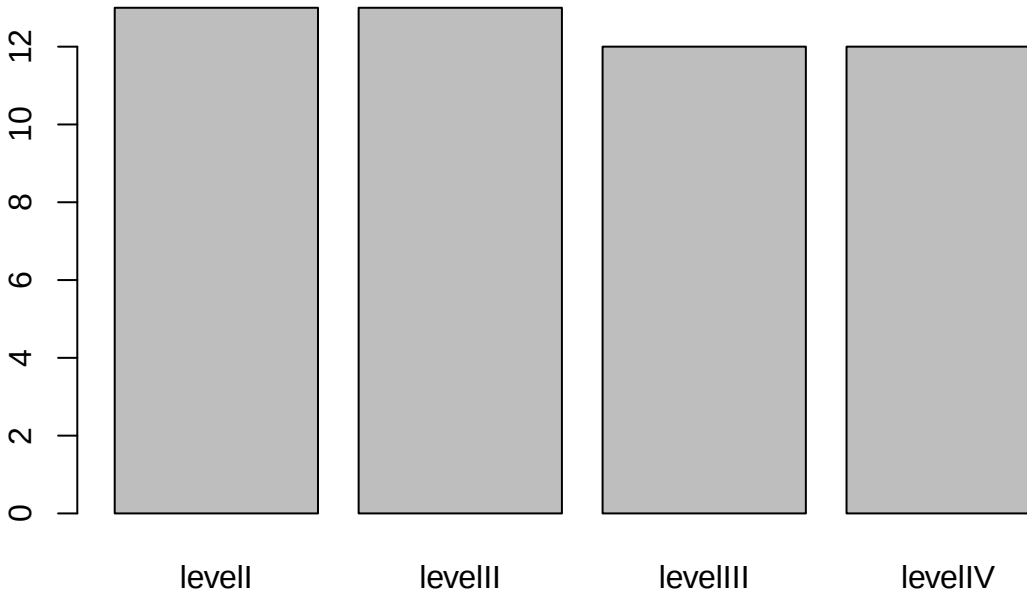
```
plot(density(simData$NumVar1)) ## Kernel density plot :The (S3) generic function density computes kernel
```




```
boxplot(simData$NumVar1) ## box plot :Produce box-and-whisker plot(s) of the given (grouped) values.
```

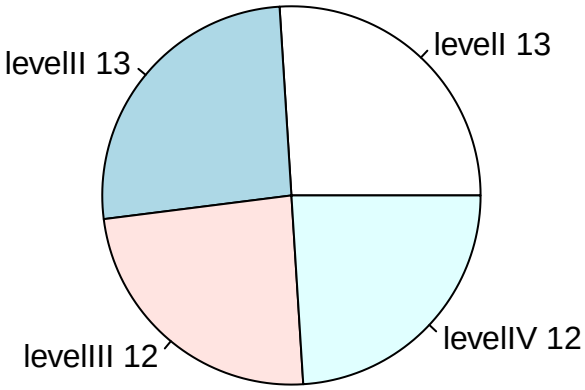



```
plot(simData$FacVar3) ## bar plot
```

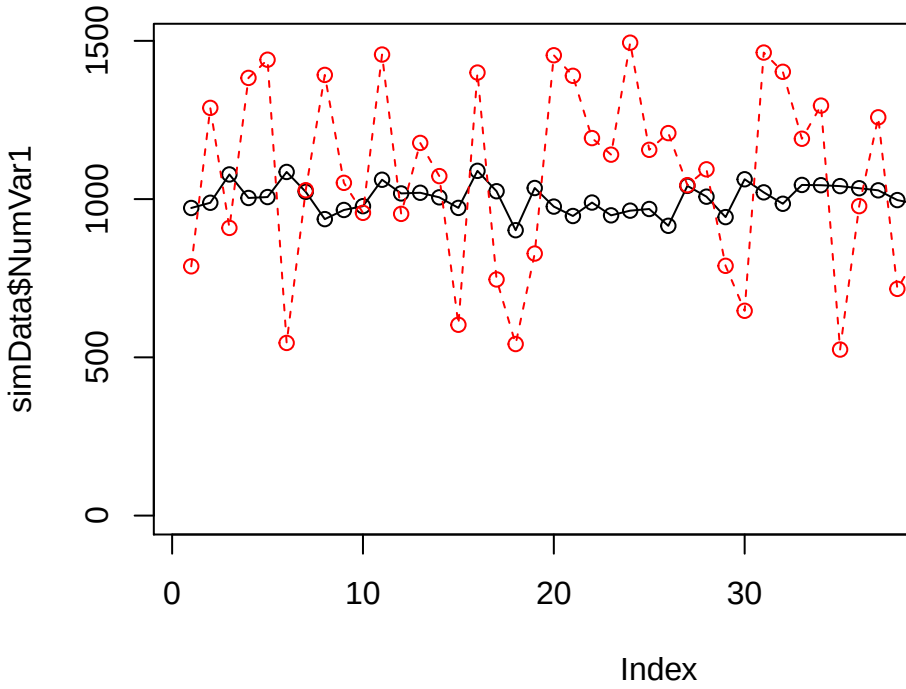


One Variable: Factor Variable

```
## pie chart - Not the best graph --- use with caution
##table uses the cross-classifying factors to build a contingency table of the counts at each combinatio
counts=table(simData$FacVar3) ## get counts
labs=paste(simData$FacVar3,counts)## create labels
##  the paste command concatenates vectors after converting to characters.
##creates a pie chart with pie command with labels and counts the amount
## labels on the key parts of the graph
pie(counts,labels=labs) ## plot
```

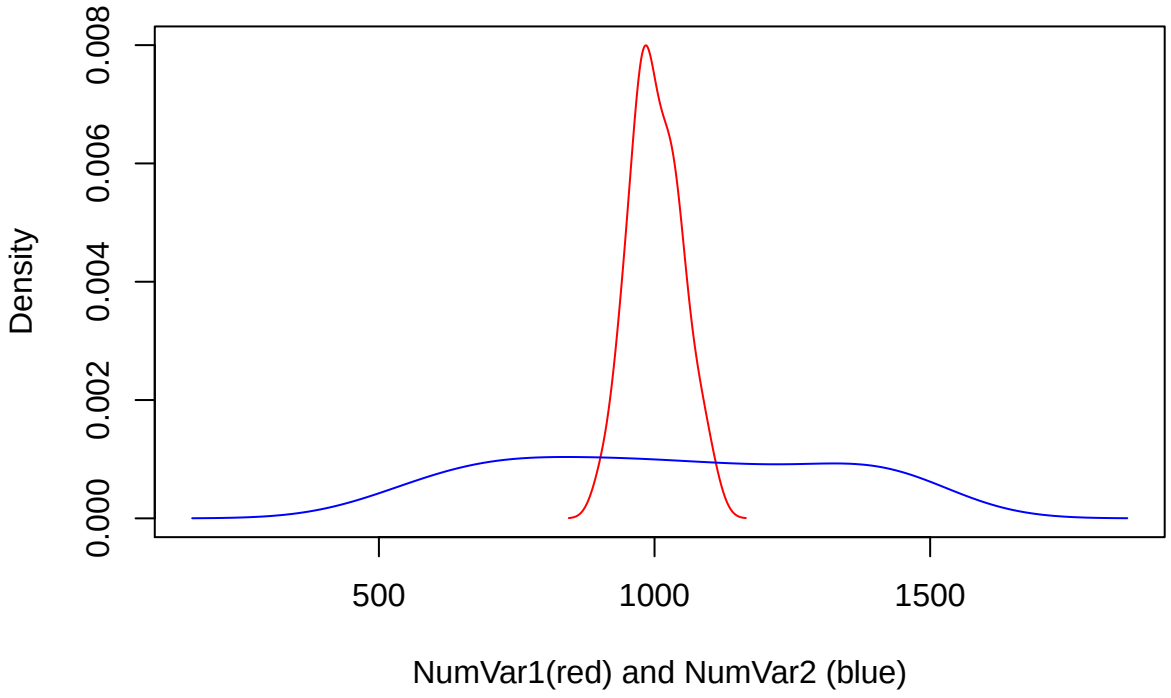



```
plot(simData$NumVar1,type="o",ylim=c(0,max(simData$NumVar1,simData$NumVar2)))## index plot with one vari
##lines command: A generic function taking coordinates given in various ways and joining the correspondi
##creates a red line that will compare the relationship between the two variables, shows contrast
lines(simData$NumVar2,type="o",lty=2,col="red")## add another variable
```

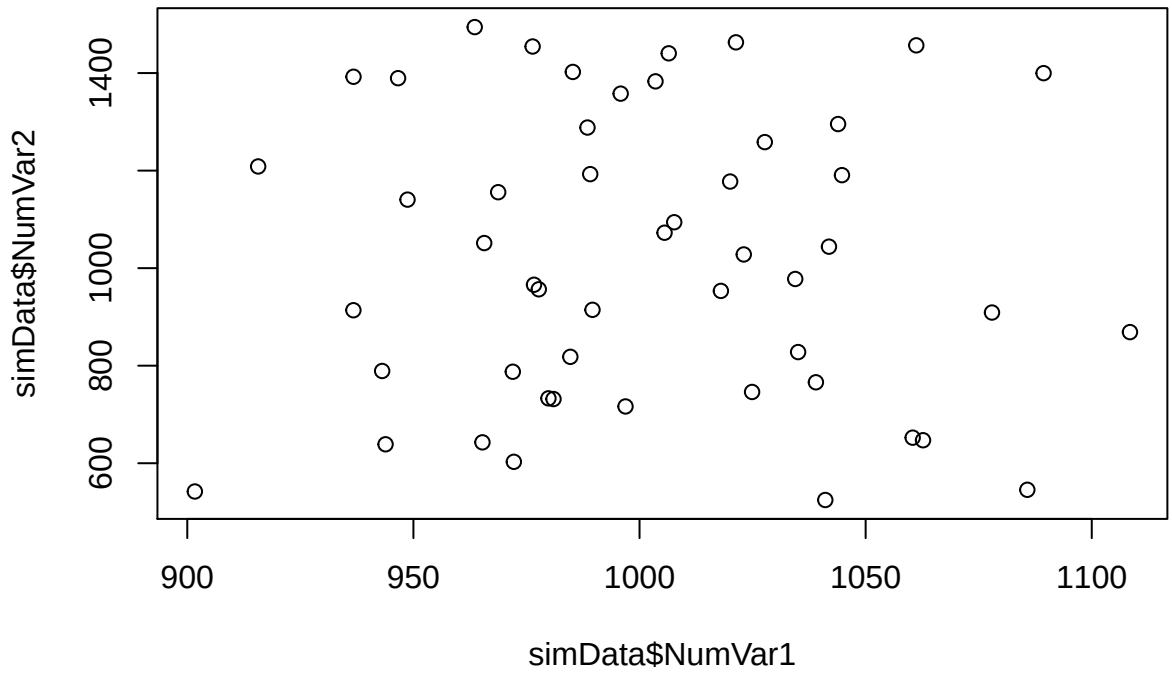


Two Variables: Two Numeric Variables

```
## 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")
## there will be a blue line and a red line that compares data in the density graphs
#plots the lines
##switches the x and y axis from the previous graph
lines(dv1, col = "red")
lines(dv2, col = "blue")
```

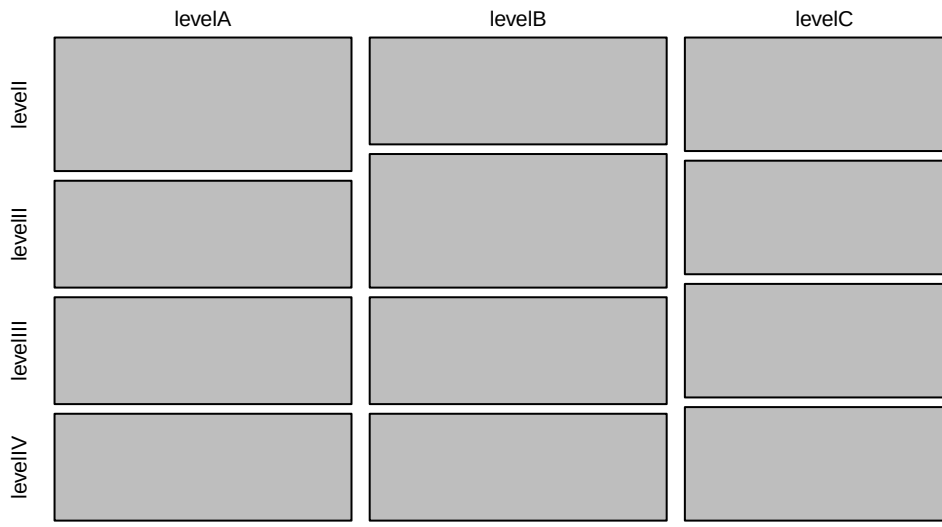



```
## scatterplots
## turns the data above into a scatter plot rather than a density plot
plot(simData$NumVar1,simData$NumVar2)
```



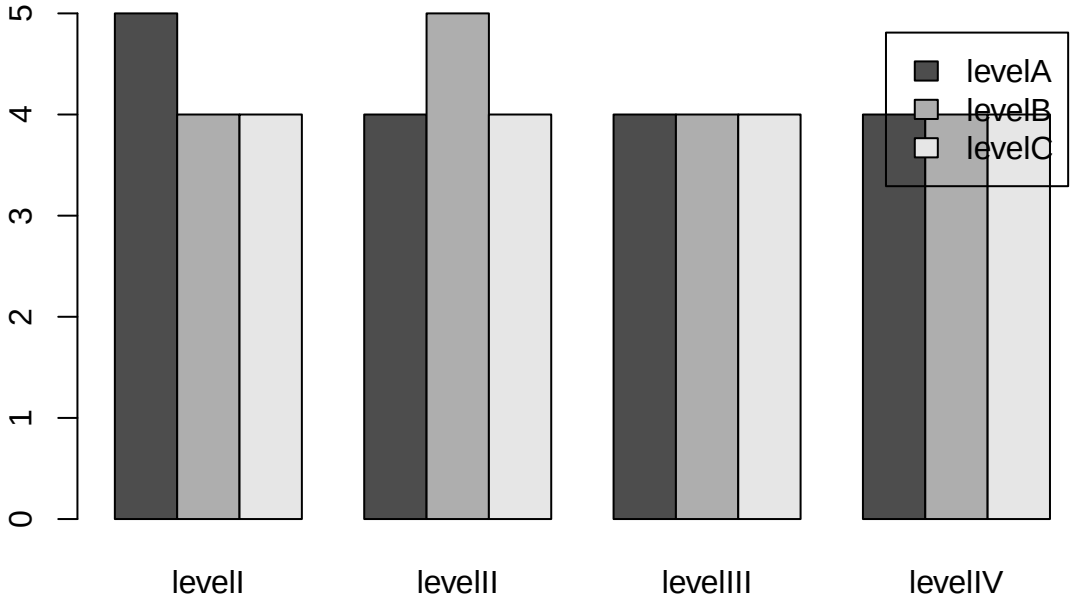

```
## Mosaic plot
## Plots the data into a table with level 1-4 from the scatterplot, there is level A,B,& C
```

```
##Personally I think this is not a very efficient type of graph
## Plots the three factors
plot(table(simData$FacVar2,simData$FacVar3))
```

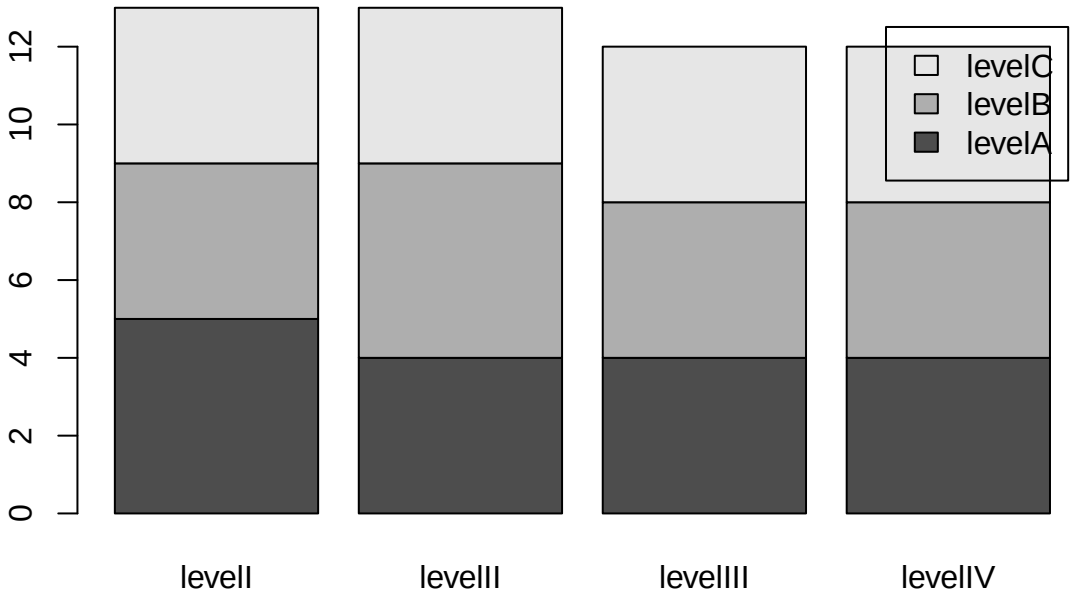


Two Variables: Two Factor Variables

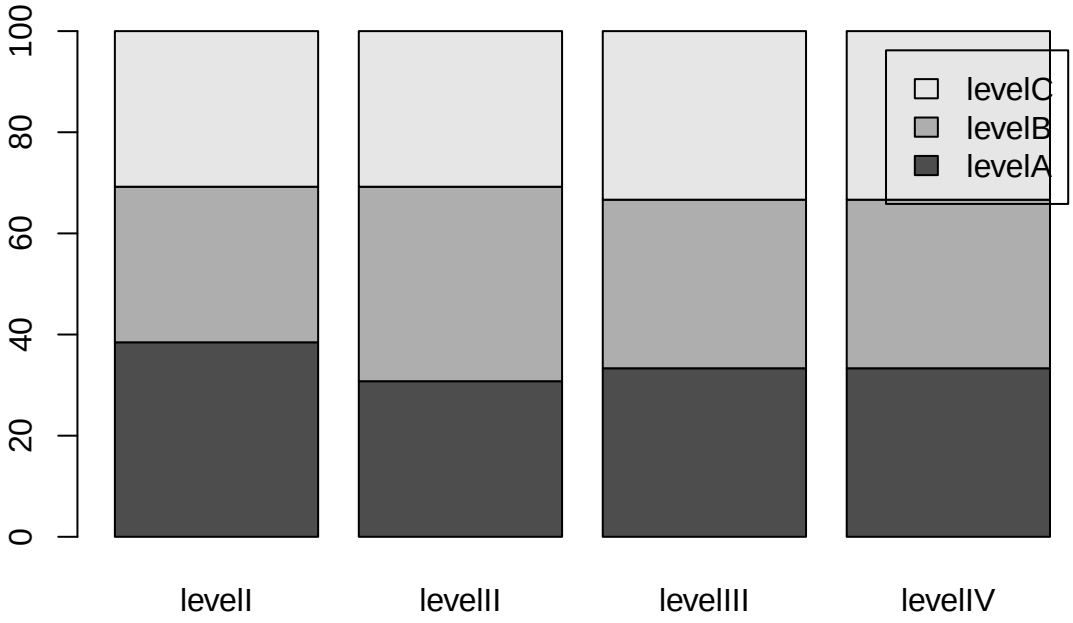
```
## barplots-Creates a bar plot with vertical or horizontal bars. In this case the bars are stacked to c
## Legend command: This function can be used to add legends to plots. Note that a call to the function
bartable=table(simData$FacVar2,simData$FacVar3) ## get the cross tab
barplot(bartable,beside=TRUE, legend=levels(unique(simData$FacVar2))) ## plot
```




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

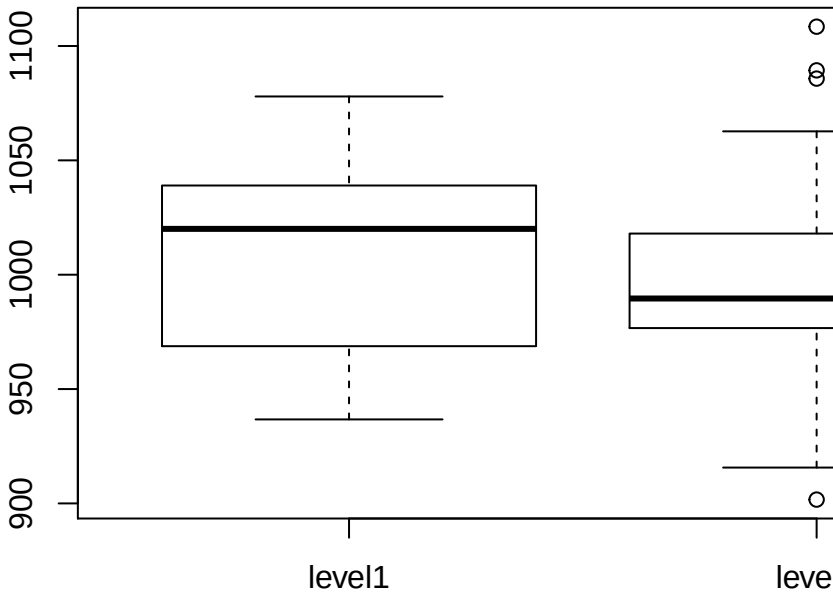



```
barplot(prop.table(bartable,2)*100, legend=levels(unique(simData$FacVar2))) ## stacked 100%
```



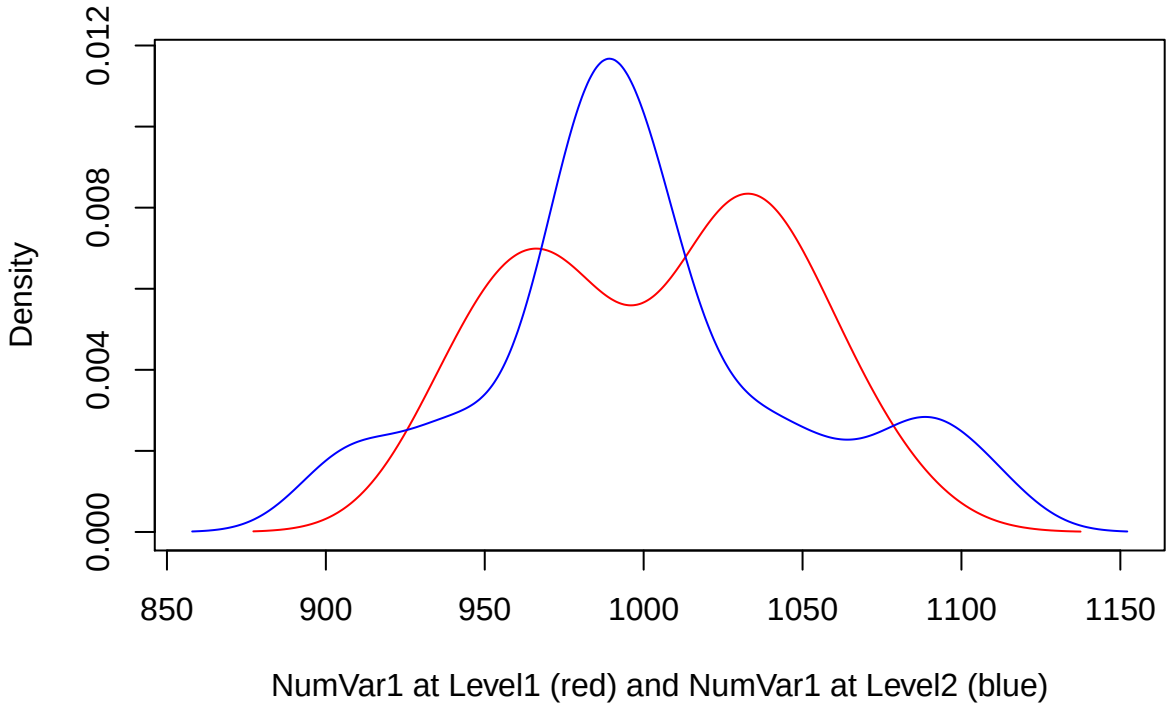

```
## unique command: unique returns a vector, data frame or array like x but with duplicate elements/rows
```

```
## Box plots for the numeric var over the levels of the factor var
## compares two types of box plots within the same graph
plot(simData$FacVar1,simData$NumVar1)
```

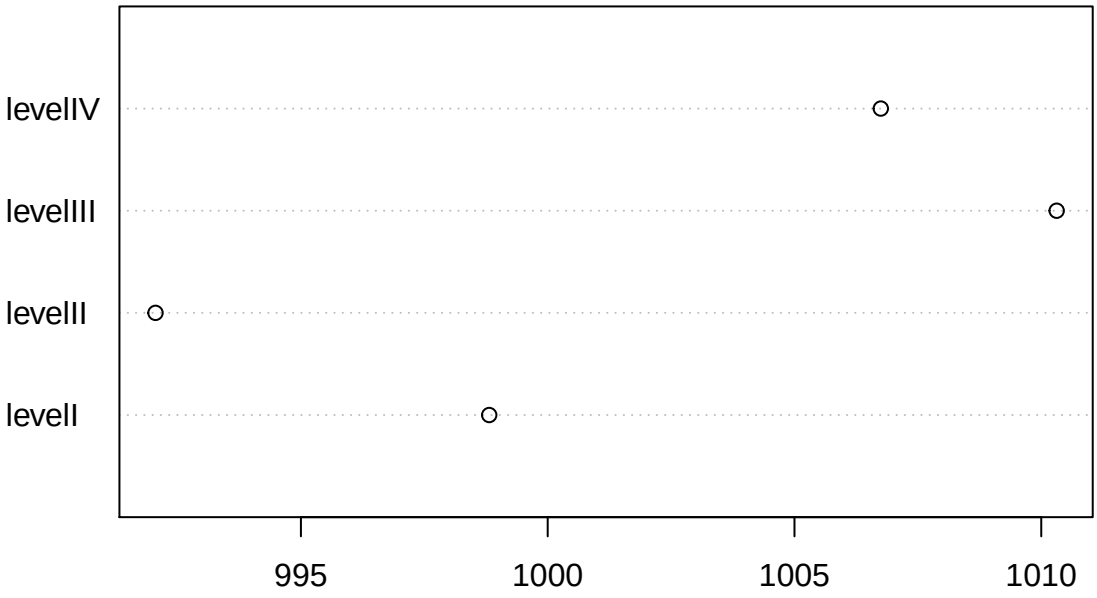


Two Variables: One Factor and One Numeric

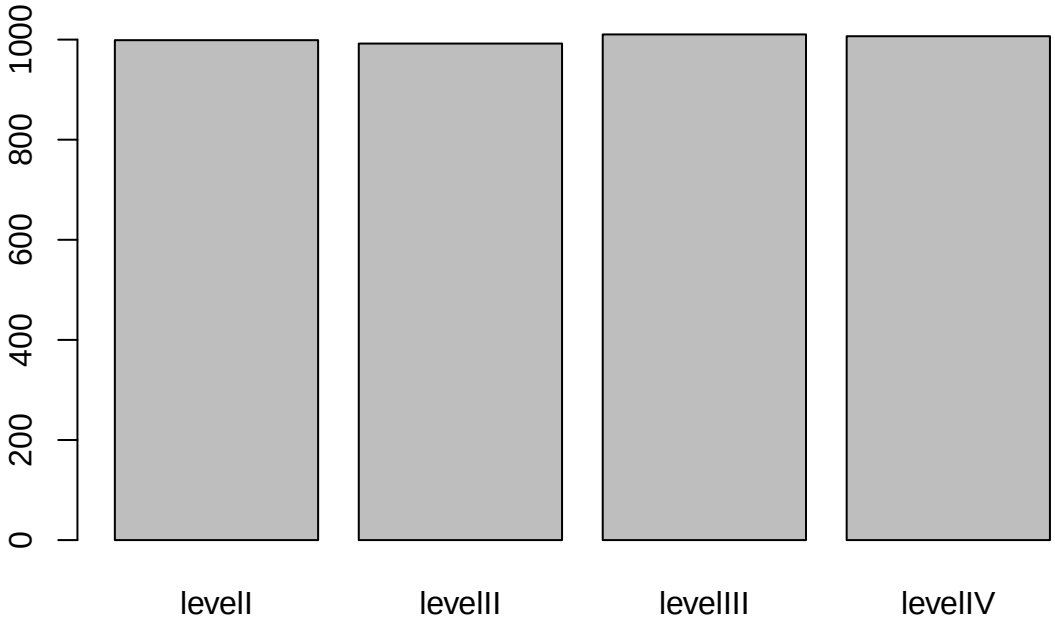
```
## density plot of numeric var across multiple levels of the factor var
level1=simData[simData$FacVar1=="level1",]
level2=simData[simData$FacVar1=="level2",]

##The (S3) generic function density computes kernel density estimates. Its default method does so with
dv3=density(level1$NumVar1)
dv4=density(level2$NumVar1)

#Range command: range returns a vector containing the minimum and maximum of all the given arguments.
#The lines are red (level 1) and blue (level 2)
plot(range(dv3$x, dv4$x),range(dv3$y, dv4$y), type = "n", xlab = "NumVar1 at Level1 (red) and NumVar1 a
lines(dv3, col = "red")
lines(dv4, col = "blue")
```




```
## Mean of one numeric var over levels of one factor var
#Splits the data into subsets, computes summary statistics for each, and returns the result in a conven
meanagg=aggregate(simData$NumVar1, list(simData$FacVar3), mean)

dotchart(meanagg$x,labels=meanagg$Group.1) ## Draw a Cleveland dot plot.
```

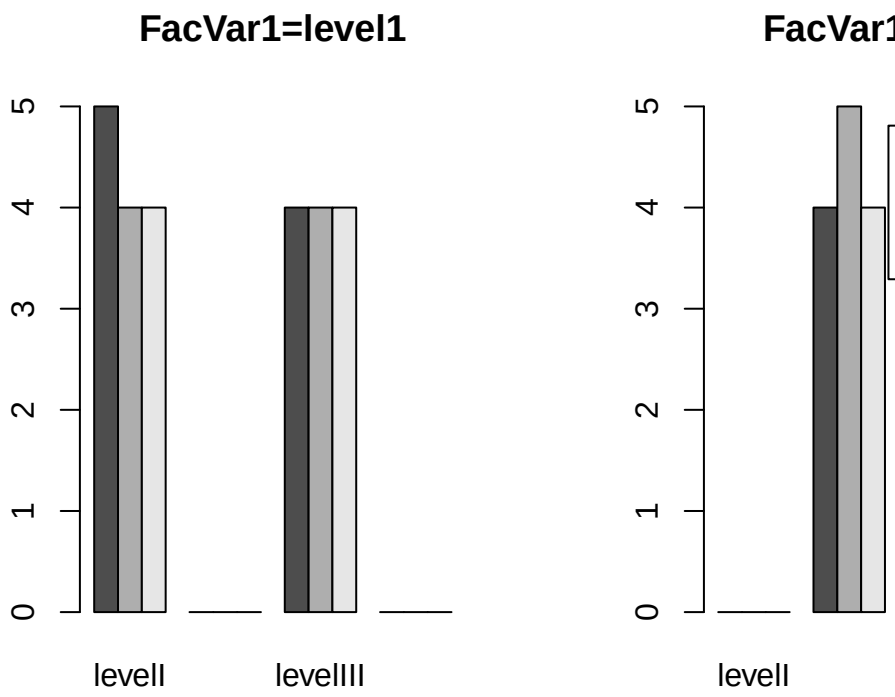



```
barplot(meanagg$x,names.arg=meanagg$Group.1)## Bar plot
```




```
## Question: Is a bar plot even appropriate when displaying a mean--- a point?
```

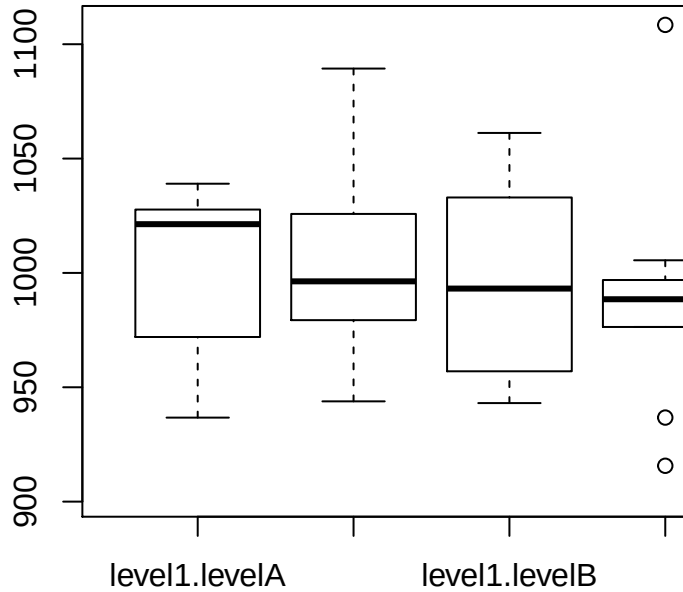
```
par(mfrow=c(1,2))
##par can be used to set or query graphical parameters. Parameters can be set by specifying them as argu
##seperates into two different graphs : one graph has narrow bars comparing the three sets of data and t
bar1table=table(level1$FacVar2,level1$FacVar3)
barplot(bar1table,beside=TRUE, main="FacVar1=level1")

bar2table=table(level2$FacVar2,level2$FacVar3)
barplot(bar2table,beside=TRUE, main="FacVar1=level2", legend=levels(unique(level2$FacVar2)))
```



Three Variables: Three Factor Variables

```
##creates a legend or key for the user to see
```

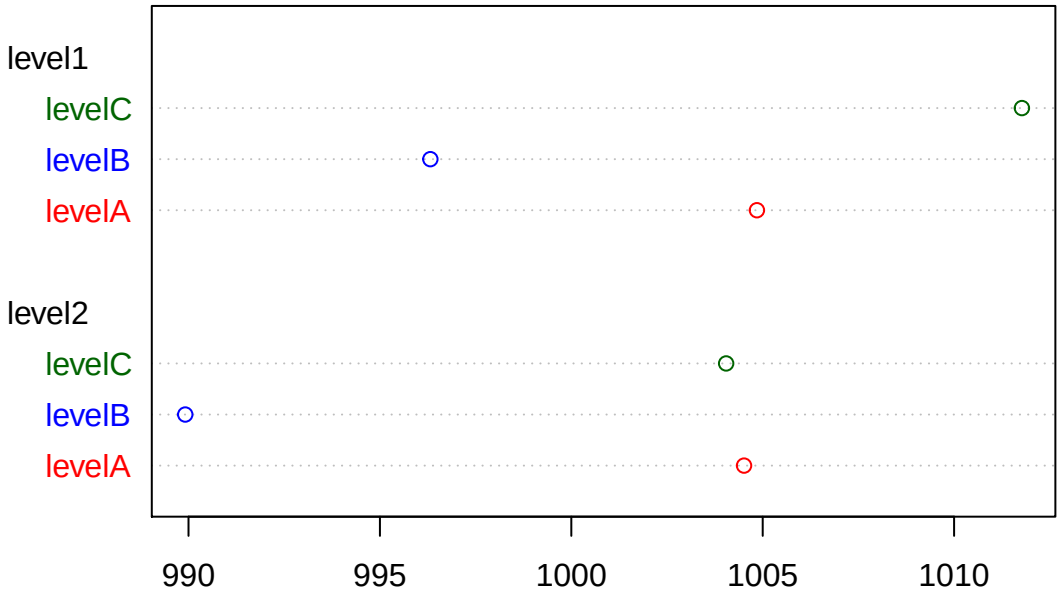
```
par(mfrow=c(1,1))
## boxplot of NumVar1 over an interaction of 6 levels of the combination of FacVar1 and FacVar2
boxplot(NumVar1~interaction(FacVar1,FacVar2),data=simData)
```



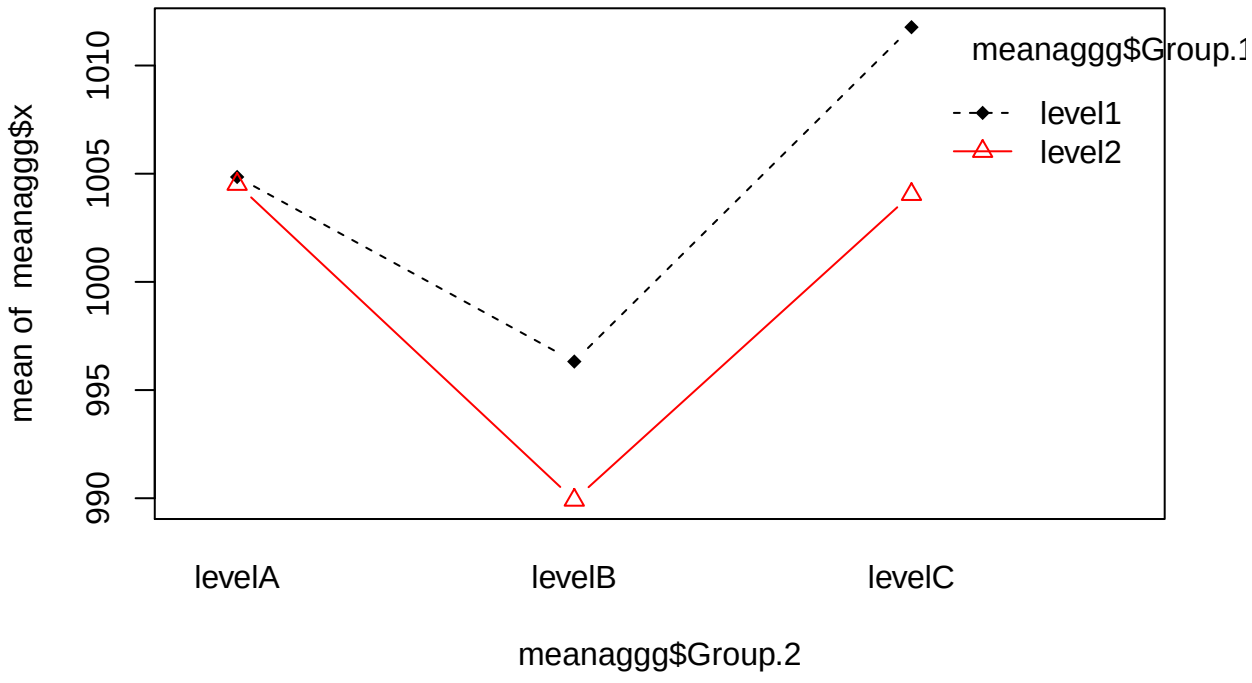
Three Variables: One Numeric and Two Factor Variables

```
## Mean of 1 Numeric over levels of two factor vars
##order returns a permutation which rearranges its first argument into ascending or descending order, bi
meanaggg=aggregate(simData$NumVar1, list(simData$FacVar1,simData$FacVar2), mean)
```

```
meanaggg=meanaggg[order(meanaggg$Group.1),]
meanaggg$color[meanaggg$Group.2=="levelA"] = "red"
meanaggg$color[meanaggg$Group.2=="levelB"] = "blue"
meanaggg$color[meanaggg$Group.2=="levelC"] = "darkgreen"

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

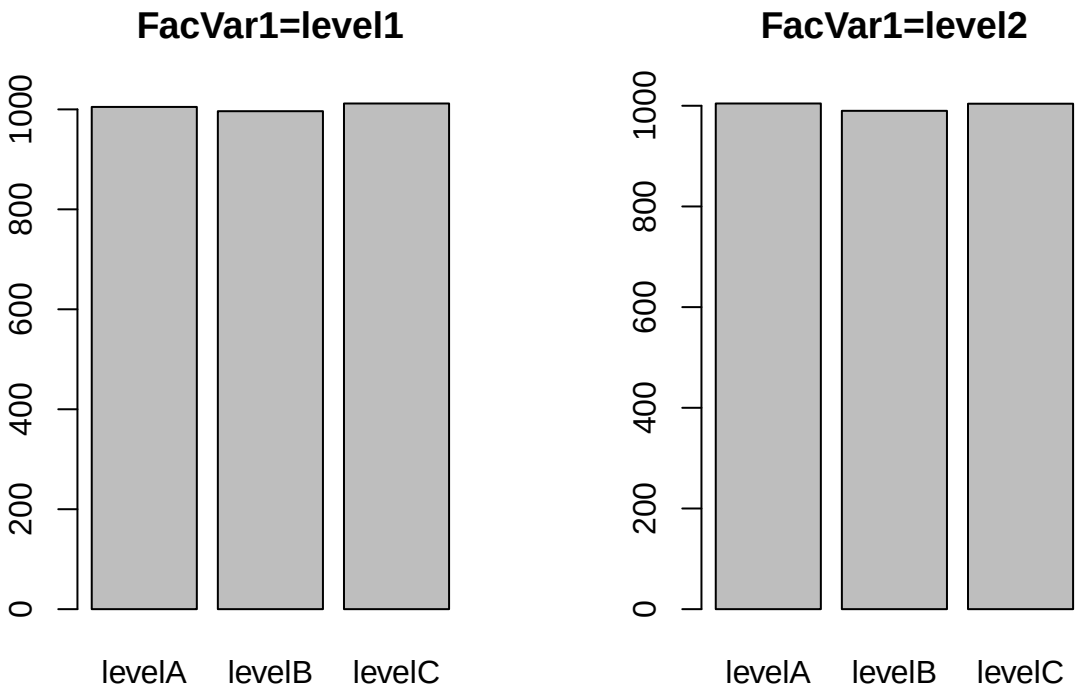



```
interaction.plot(meanaggg$Group.2,meanaggg$Group.1,meanaggg$x,type="b", col=c(1:2),pch=c(18,24)) ## int
```

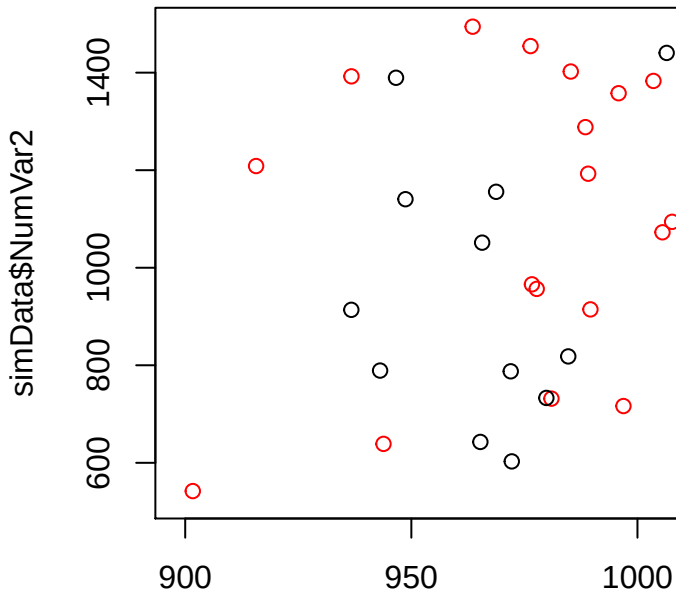



```
## Col command: Returns a matrix of integers indicating their column number in a matrix-like object, or
## some a bar plot
```

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

level1=meanaggg[meanaggg$Group.1=="level1",]
level2=meanaggg[meanaggg$Group.1=="level2",]

barplot(level1$x,names.arg=level1$Group.2, main="FacVar1=level1")
barplot(level2$x,names.arg=level2$Group.2, main="FacVar1=level2")
```



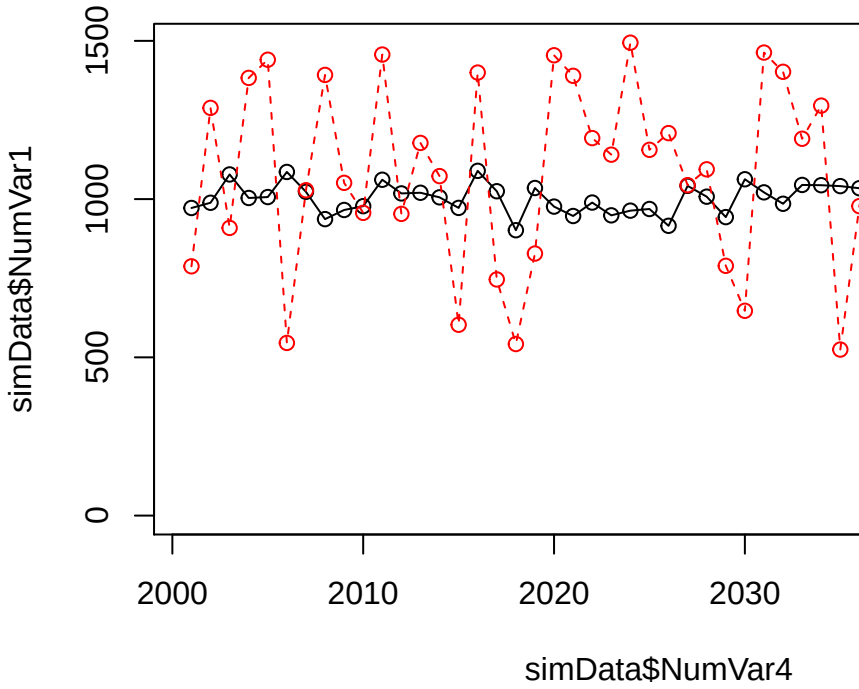

```r
## Scatter plot with color identifying the factor variable
par(mfrow=c(1,1))
plot(simData$NumVar1,simData$NumVar2, col=simData$FacVar1)
legend("topright",levels(simData$FacVar1),fill=simData$FacVar1)
```



Three Variables: Two Numeric and One Factor Variables

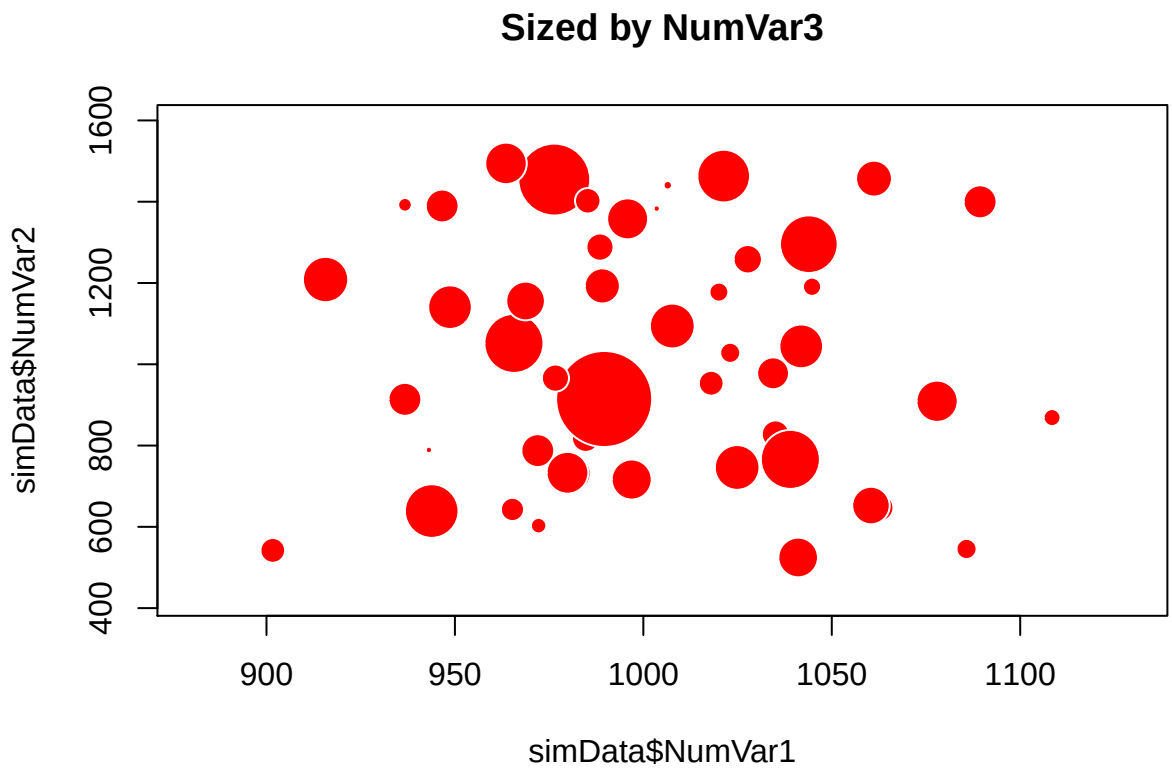
```
## the legend is placed on the top right corner of the graph
#These functions provide information about the uniform distribution on the interval from min to max. du
```

```
## 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
```

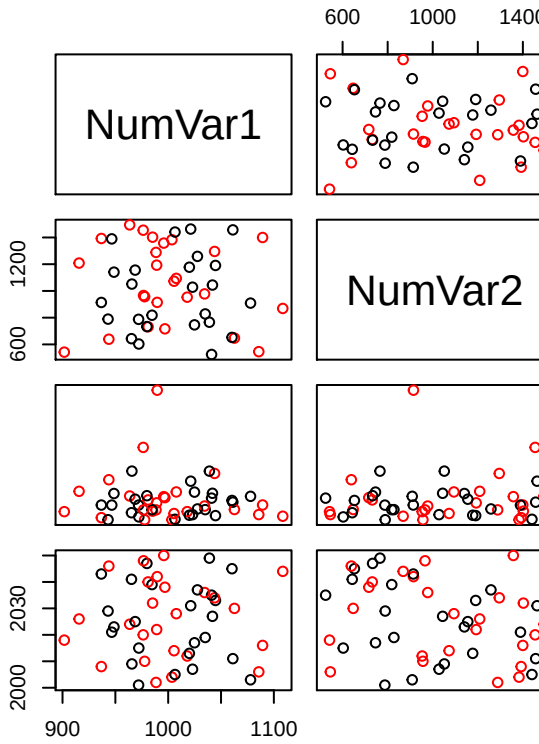


Three Variables: Three Numeric Variables

```
## 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/
## the radius is squared with the sqrt command
## the colors used are red and black, no white? Could this be the background?
radius <- sqrt( simData$NumVar3/ pi )
symbols(simData$NumVar1,simData$NumVar2,circles=radius, inches=.25,fg="white", bg="red", main="Sized by
```




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



Scatterplot Matrix of all Numeric Vars, colored by a Factor variable

```
##A matrix of scatterplots is produced with pairs command.
```

References Besides the link from flowingdata.com referred to in the context of the bubble plot, additional websites were used as references. http://www.harding.edu/fmccown/r/ http://www.statmethods.net/