ggplot2 Blog Post

Granger Moch

September 29, 2014

Abstract

Demonstrate the functionality of ggplot and proide an introduction to its syntax and uses.

Contents

Two Variables: One Factor and One Numeric Variable 17

Three Variables: Three Factor Variables 21

Three Variables: One Numeric and Two Factor Variables 22

Three Variables: Two Numeric and One Factor Variable 26

Purpose

I am creating this guide not solely for the assignment, but so that those that are new to R can recognize the extensive functionality it offers. Furthermore, I hope to provide a succinct introduction to several different graph types and their uses. for both my sake as well as those who read this guide.

Data

Simulate 3 Factor Variables

```
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)])
```

Simulate 4 Numeric Variables

```
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)
```

The ggplot2 and reshape2 libraries need to be initialized for use on this page

```
library(ggplot2)
library(reshape2)
```

General Syntax

Each ggplot2 plot will begin with the function ggplot(), which has two primary arguments:

- data The data frame containing the data to be plotted
- aes() The aesthetic mappings to assign to plot elements

The **geom_()** functions are used alongside the + operator, in order to assign a geometric object to represent the data

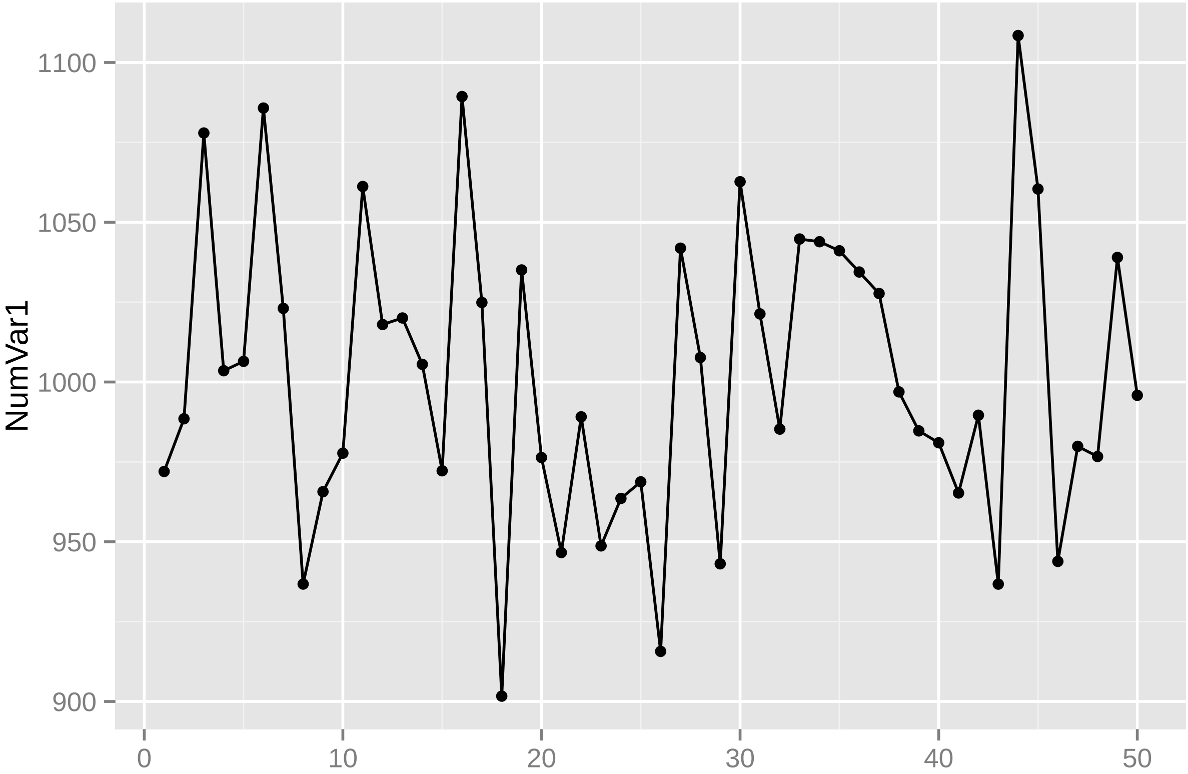
Syntax:

```
ggplot(dataset, aes(x, y))+geom_point()
```

One Variable: Numeric Variable

Index Plot of One Numeric Variable

```
ggplot(simData,aes(y=NumVar1,x=1:nrow(simData),group="NumVar1"))+geom_point()+geom_line()+ xlab("") ## I
```

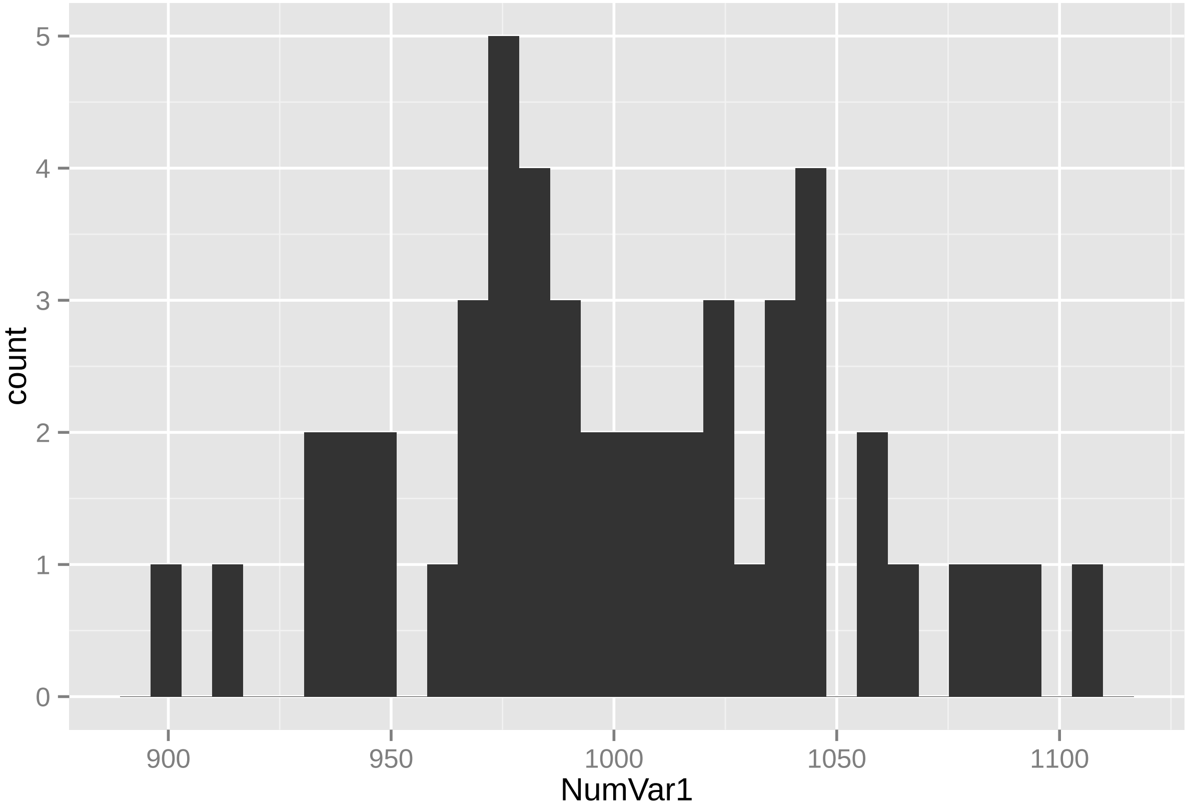


-Axis labels can be changed using **xlab()** and **ylab()** respectively for the x and y axes. In this instance, the x axis label is blank as indicated by the null value.

Histogram of One Numeric Variable

```
ggplot(simData,aes(x=NumVar1))+geom_histogram()
```

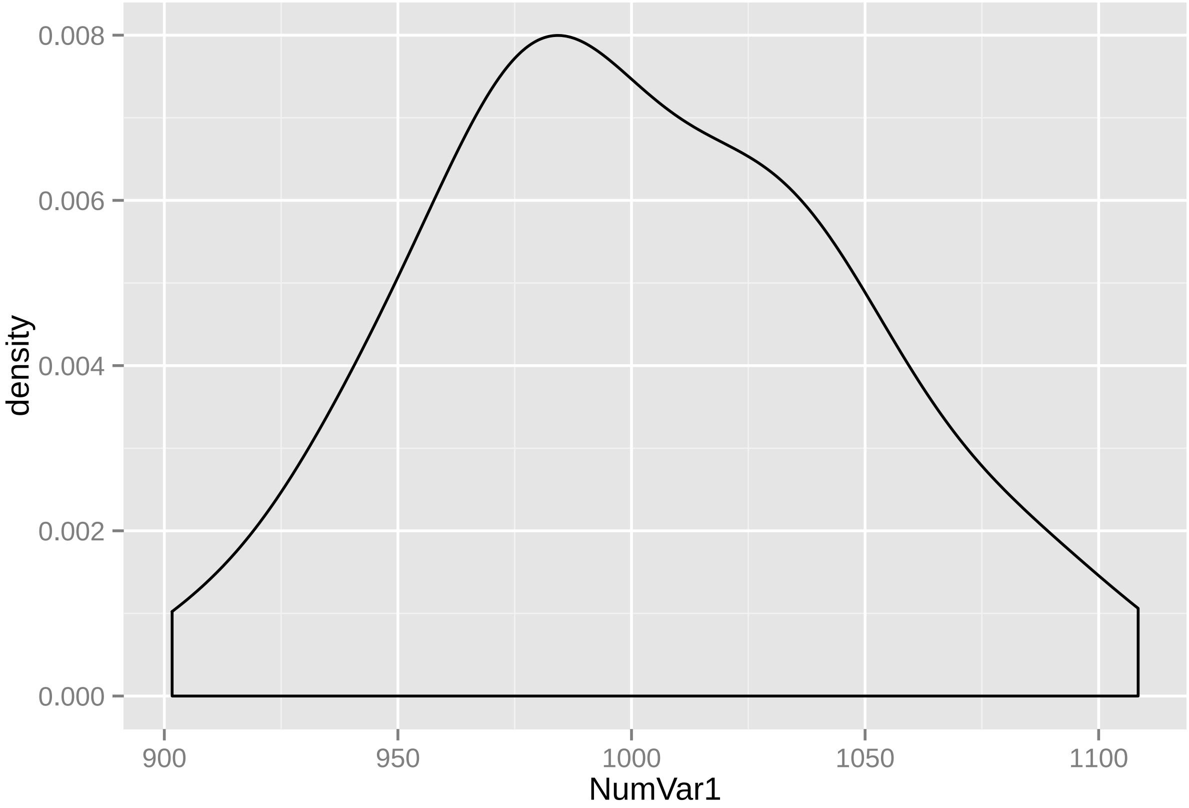
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



-This plot uses the geom_histogram(), for a list of all geoms available for use see here: http://sape.inf.usi.ch/quick-reference/ggplot2/geom

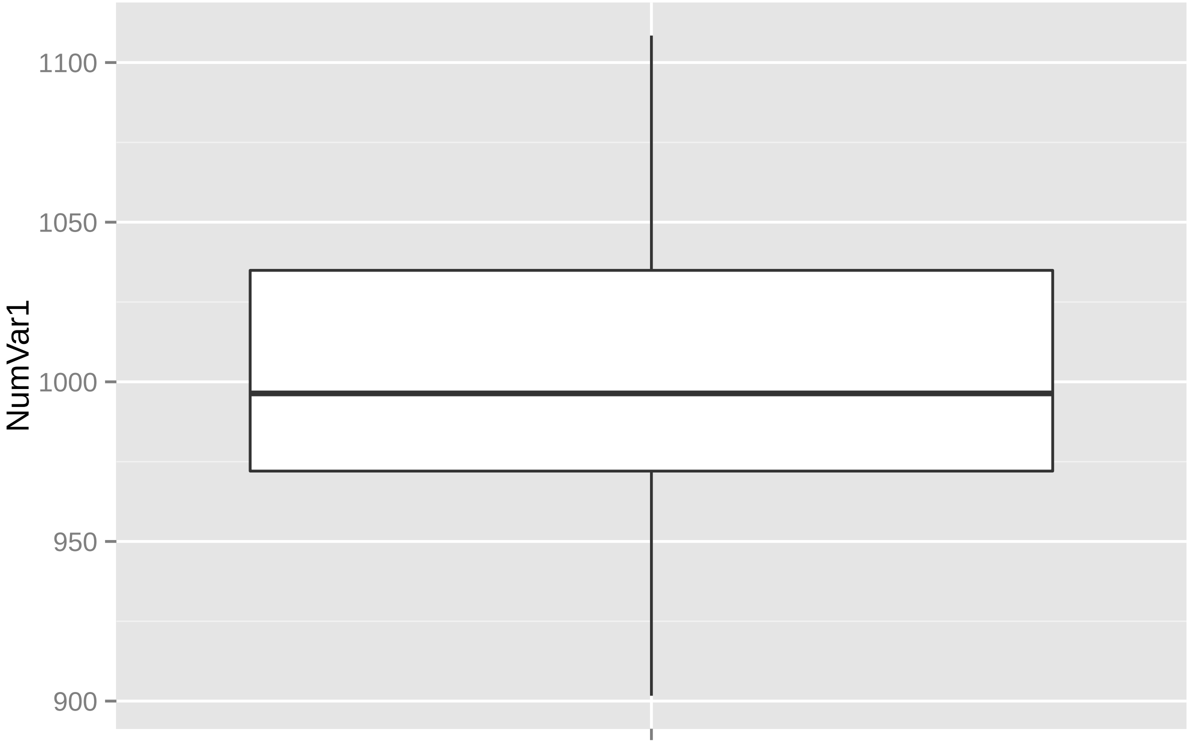
Kernel Density Plot of One Numeric Variable

```
ggplot(simData,aes(x=NumVar1))+geom_density()
```



Box Plot of One Numeric Variable

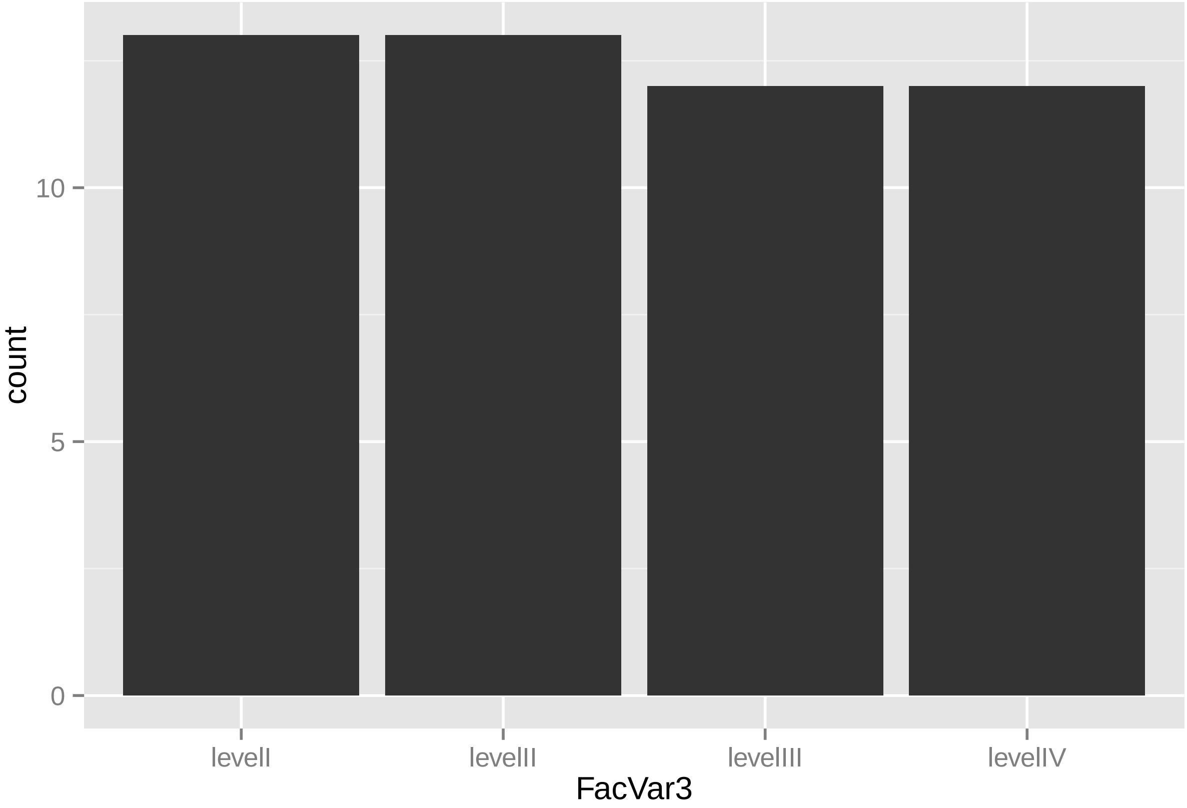
```
ggplot(simData,aes(x=factor(""),y=NumVar1))+geom_boxplot()+ xlab("") ## box plot
```



Plotting One Variable: Factor Variable

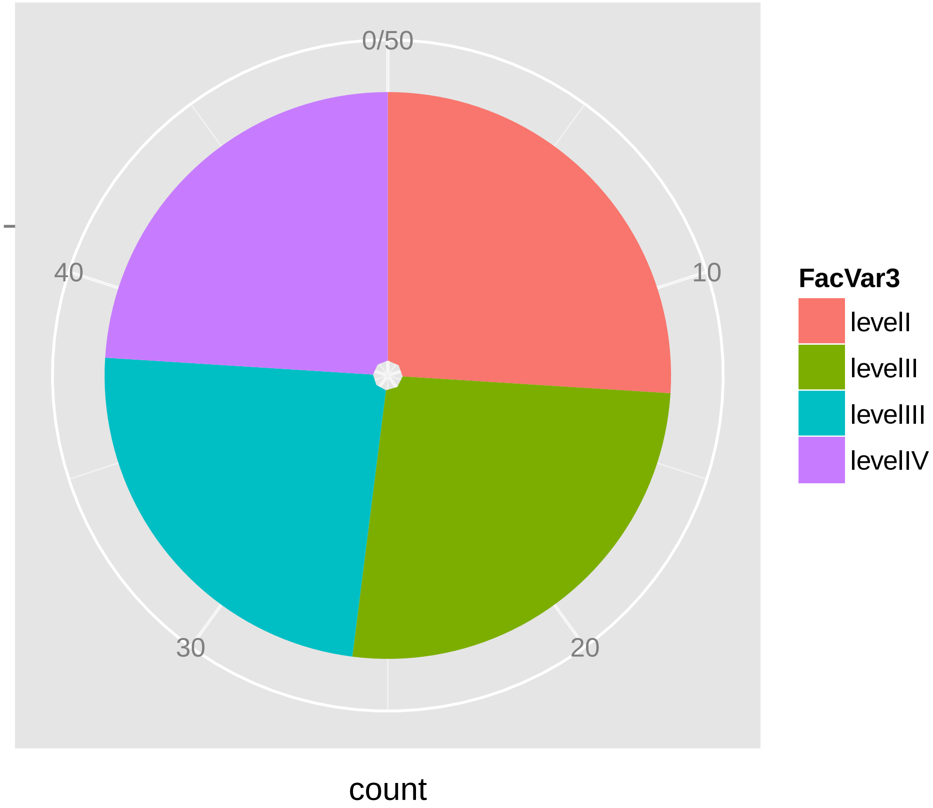
Barplot of One Factor Variable

```
ggplot(simData,aes(x=FacVar3))+geom_bar()
```



Pie chart of One Factor Variable

```
ggplot(simData,aes(x = factor(""), fill=FacVar3, label=FacVar3))+geom_bar()+ coord_polar(theta = "y")  +
```



The coordinate polar system (coord_polar) is most typically used for pie charts, as seen above.

Syntax:

coord_polar(theta = “x”, start = 0, direction = 1)

Arguments: -*theta*= variable to map angle to (x or y) -*start*= offset of starting point from 12 o’clock in radians -*direction*= 1, clockwise; -1, anticlockwise

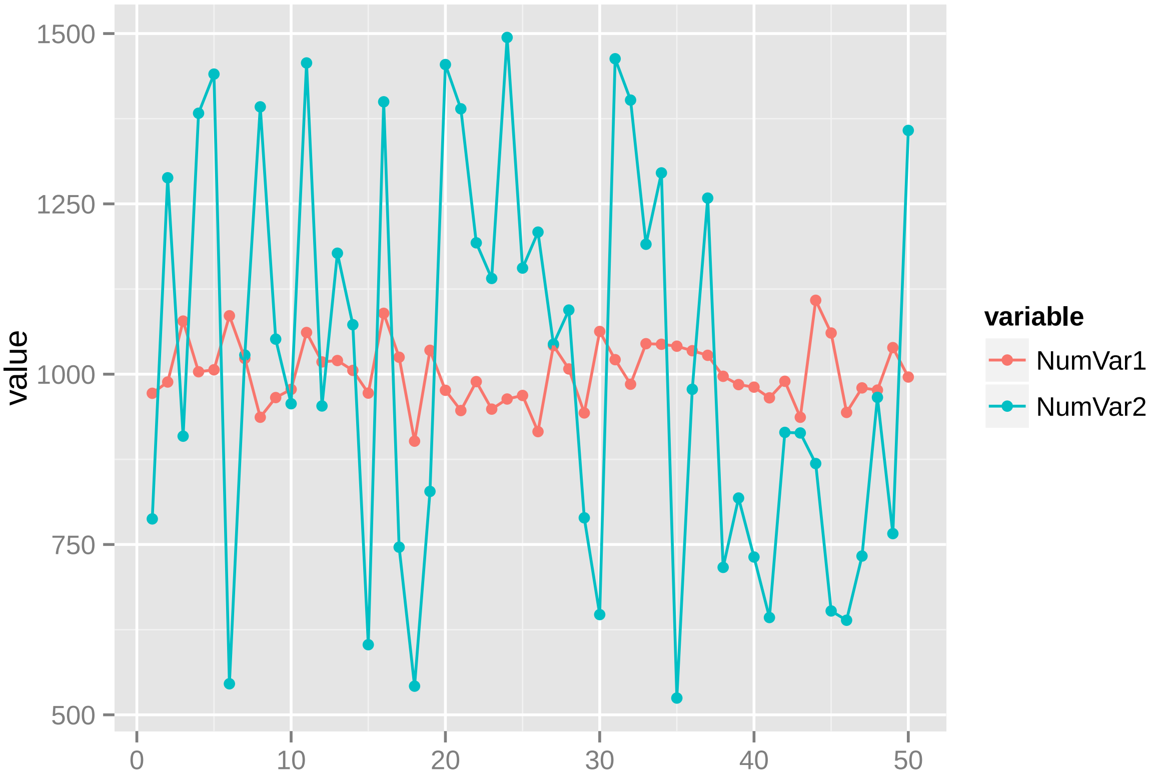
-In this instance, the variable angle is mapped to y.

Two Variables: Two Numeric Variables

Line Plot of Two Numeric Variables

```
simtmp=simData[,c(4:5)] ## 4th and 5th columns are NumVar1 and NumVar2
simtmp$index=1:nrow(simtmp)
simtmpmelt=melt(simtmp,id=c("index"))

## line plots with observation number as index
ggplot(simtmpmelt,aes(y=value,x=index,color=variable))+geom_point()+geom_line()+xlab("")
```

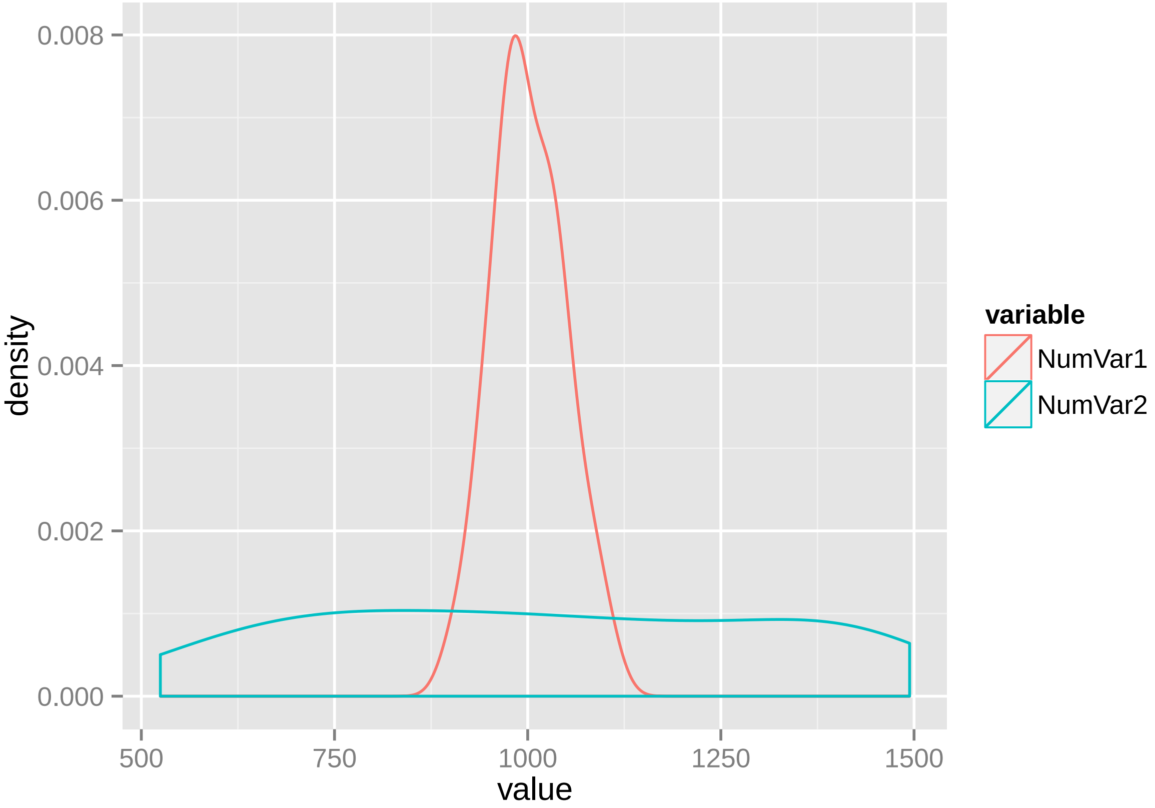


With this lineplot, the lines are colored according to the variable. This is specified as an arugment in the aes() function as “color=variable.”

The **melt** function, as part of the Reshape2 package, essentially “melts” the data you provide it so that each row is a unique id-variable combination. In other words, the “wide-format” data is “melted” into what is called “long-form data”. As these formats relate to ggplot, ggplot2 actually requires long-form data(also known as *tidy data*); this is why the **melt** function was used here. The other core function of Reshape2 is **cast**, which does just the opposite. **Cast** takes long-format data and casts it into wide-format data.

Density Plot of Two Numeric Variables

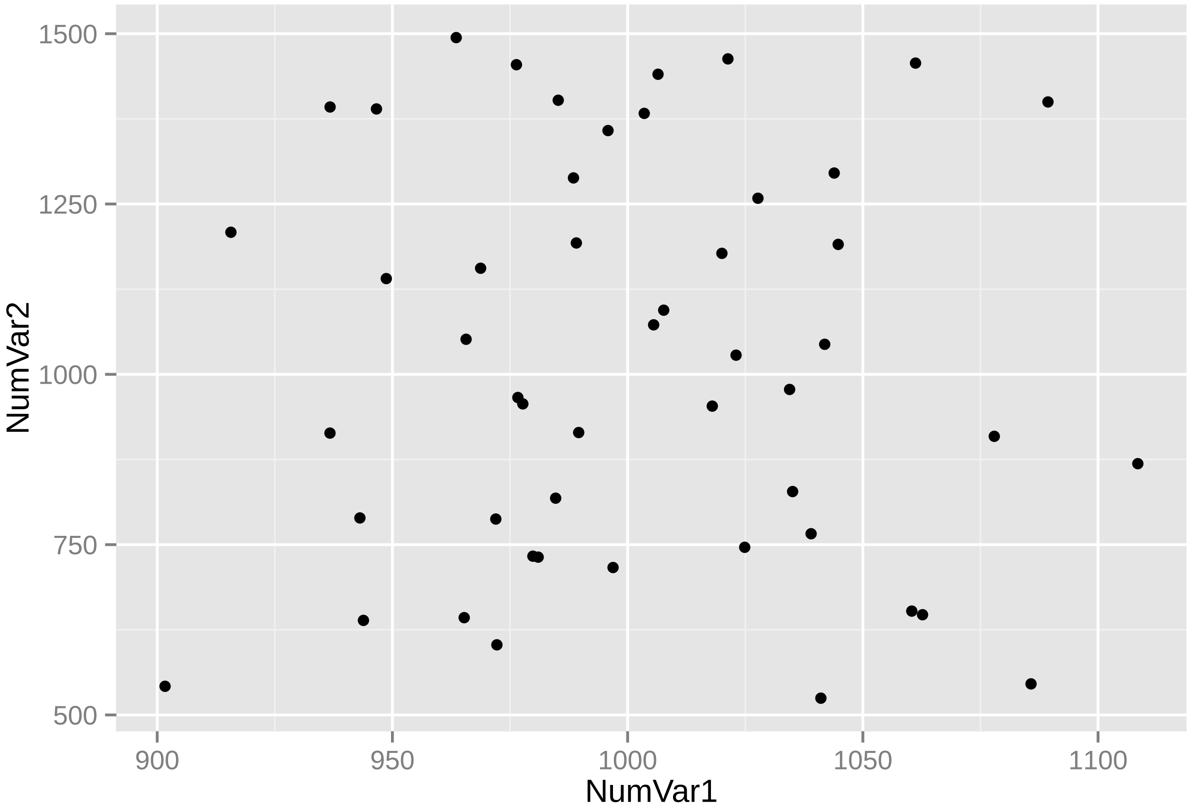
```
ggplot(simtmpmelt,aes(x=value,color=variable))+geom_density()
```



This Density Plot uses the simptmpmelt dataset created on the previous page.

Scatter Plot of Two Numeric Variables

```
ggplot(simData,aes(x=NumVar1,y=NumVar2))+geom_point()
```



Two Variables: Two Factor Variables

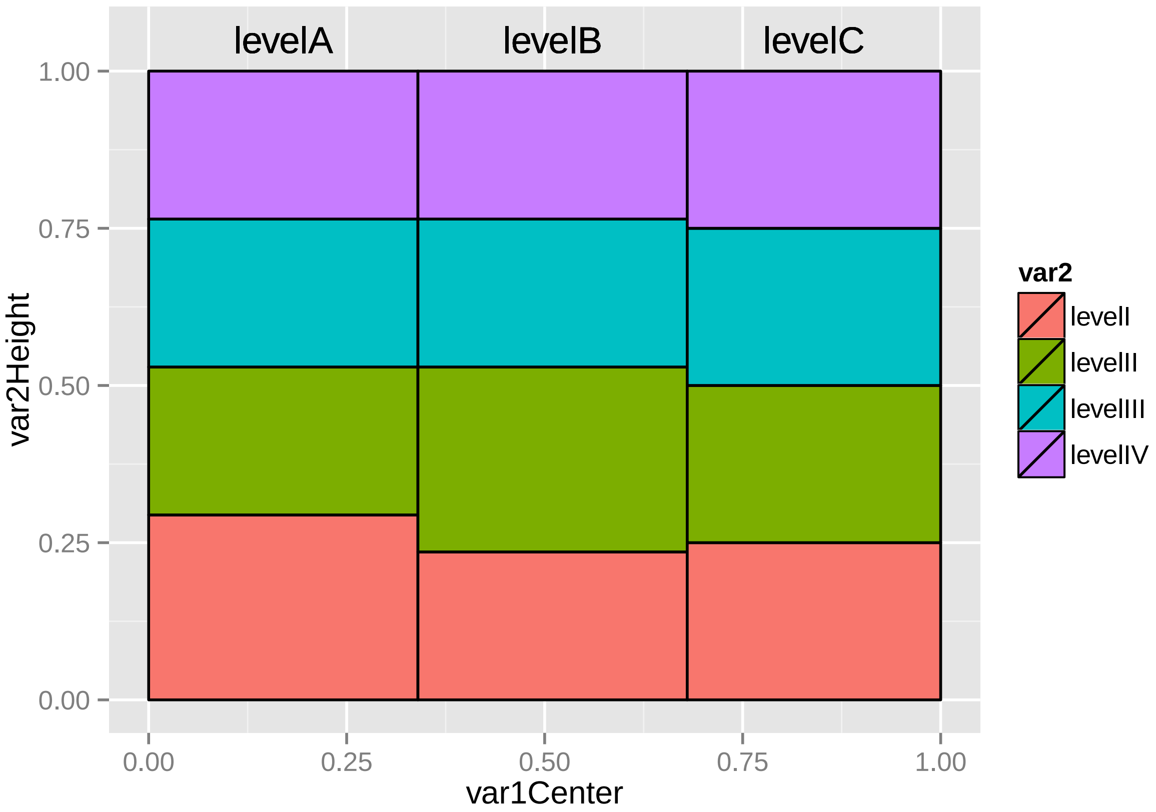
Mosaic Plot of Two Factor Variables

```
ggMMplot <- function(var1, var2){
  require(ggplot2)
  levVar1 <- length(levels(var1))
  levVar2 <- length(levels(var2))

  jointTable <- prop.table(table(var1, var2))
  plotData <- as.data.frame(jointTable)
  plotData$marginVar1 <- prop.table(table(var1)) ##creates a table of proportions
  plotData$var2Height <- plotData$Freq / plotData$marginVar1
  plotData$var1Center <- c(0, cumsum(plotData$marginVar1)[1:levVar1 -1]) +
    plotData$marginVar1 / 2

  ggplot(plotData, aes(var1Center, var2Height)) +
    geom_bar(stat = "identity", aes(width = marginVar1, fill = var2), col = "Black") +
    geom_text(aes(label = as.character(var1), x = var1Center, y = 1.05))
}
ggMMplot(simData$FacVar2, simData$FacVar3)
```

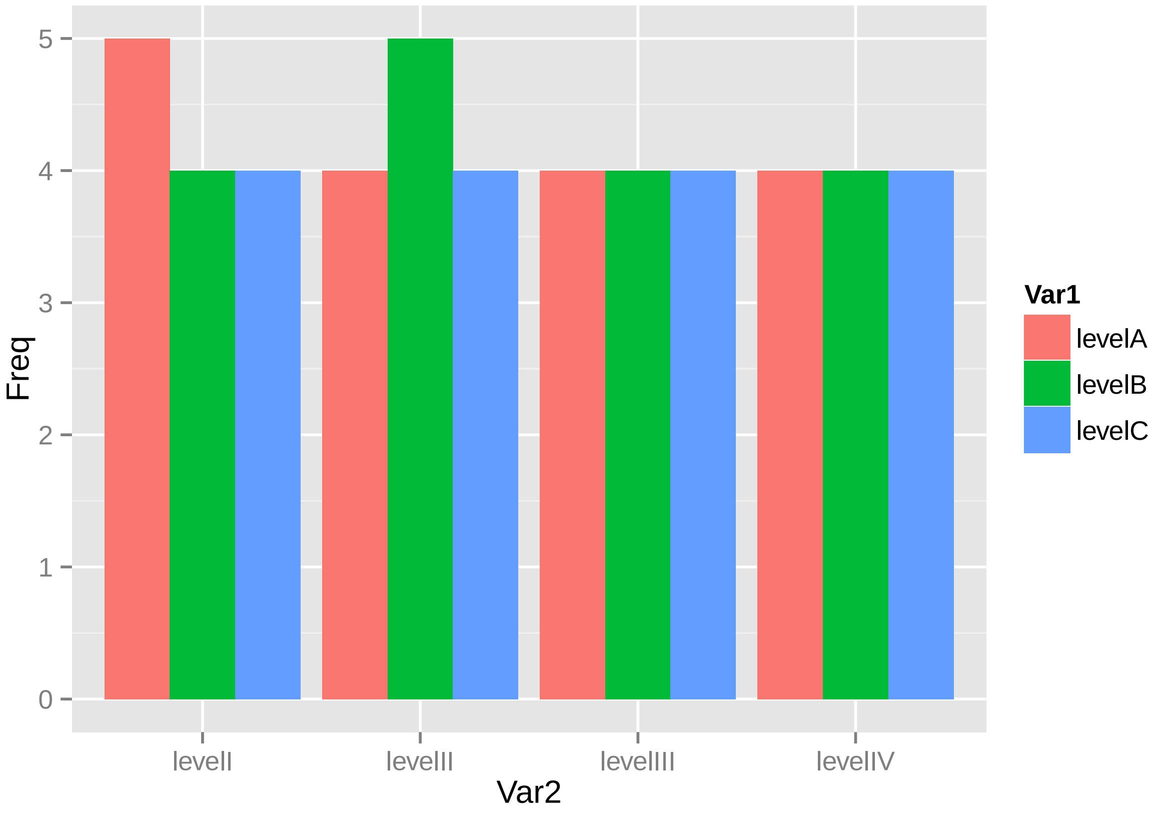
```
## Warning: position_stack requires constant width: output may be incorrect
```



Note:Because ggplot2 does not currently support mosaic plots, the function ggMMplot is created from scratch to create the plot.

Bar Plot of Two Factor Variables

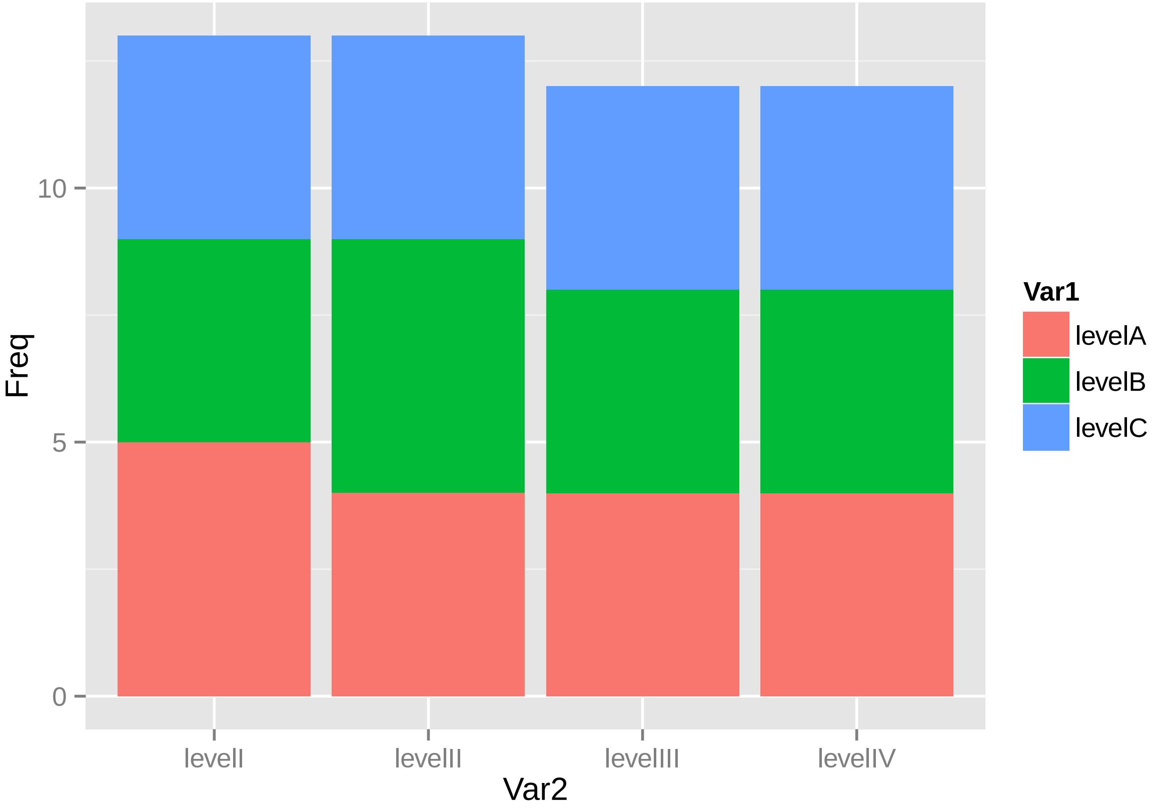
```
bartabledat = as.data.frame(table(simData$FacVar2, simData$FacVar3)) ## get the cross tab
ggplot(bartabledat,aes(x=Var2,y=Freq,fill=Var1))+geom_bar(position="dodge", stat="identity")
```



```
## added stat="identity" (since we are in fact, mapping a variable to y)
```

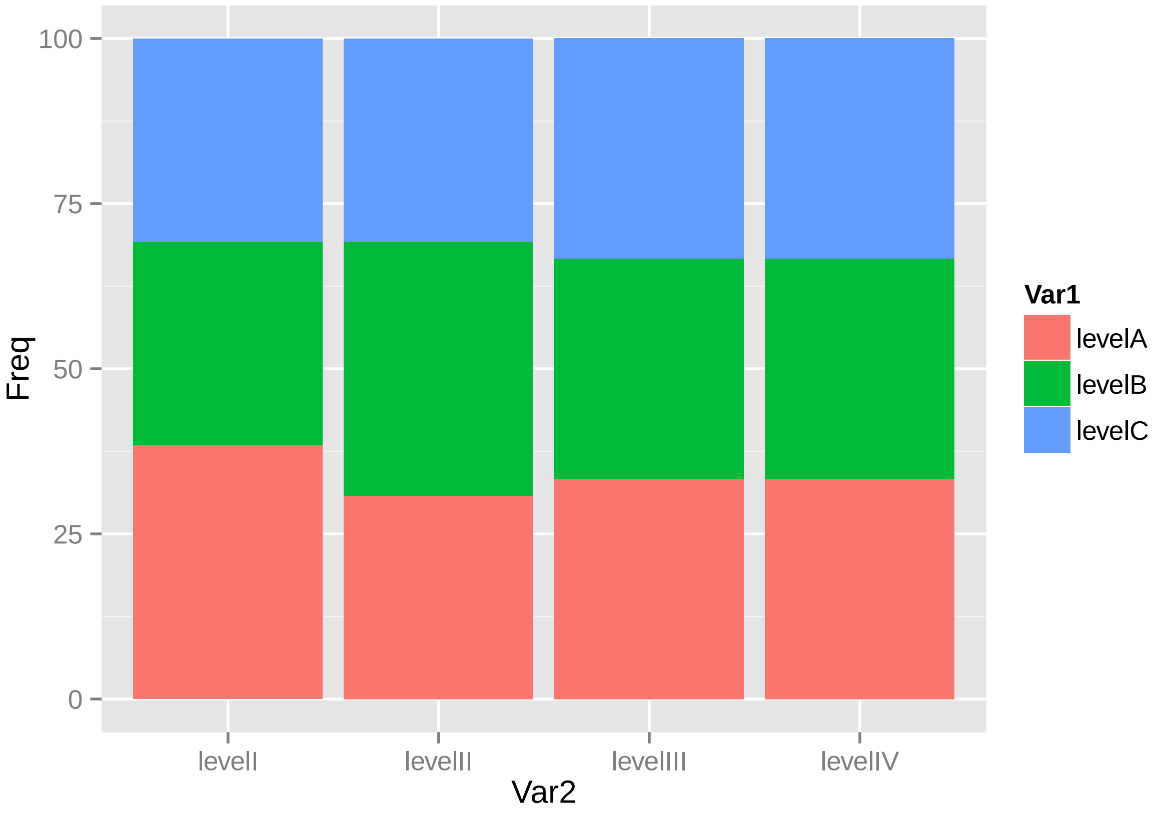
Stacked Bar Plot of Two Factor Variables

```
ggplot(bartabledat,aes(x=Var2,y=Freq,fill=Var1))+geom_bar(stat="identity") ## stacked
```



Stacked Bar Plot of Two Factor Variables (100%)

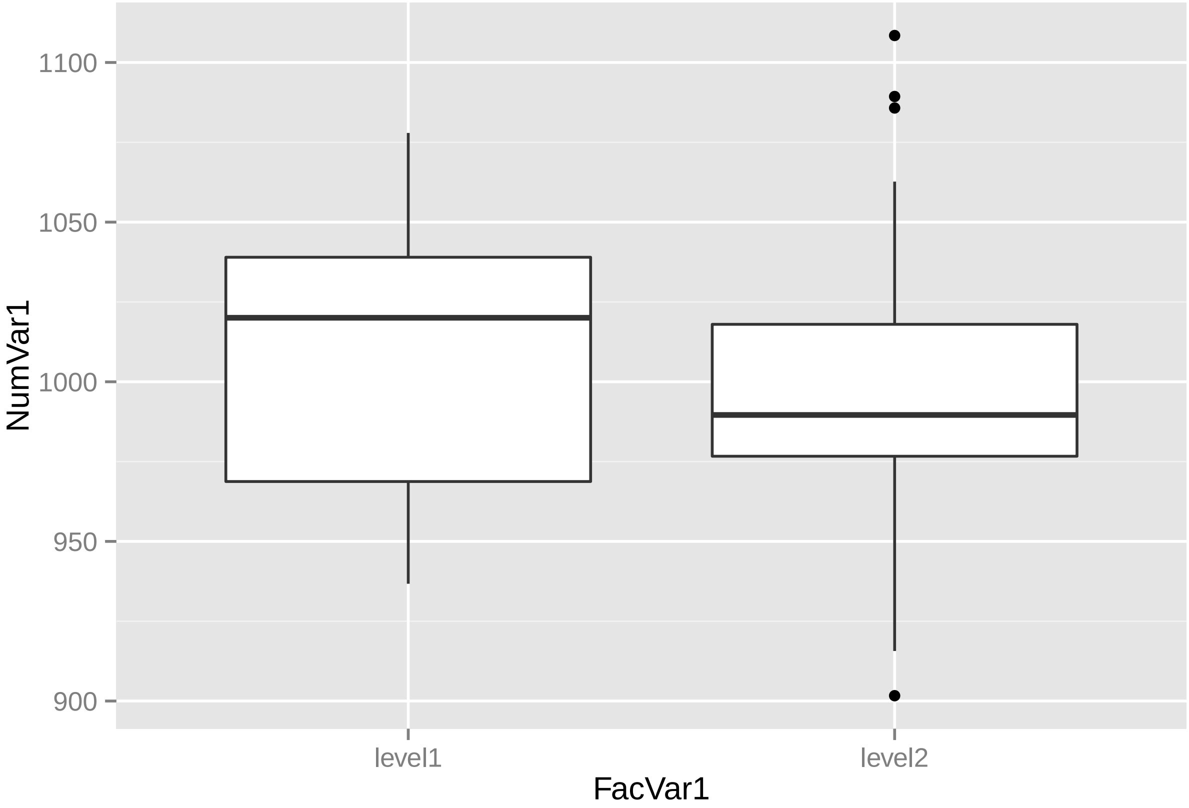
```
bartableprop =as.data.frame(prop.table(table(simData$FacVar2, simData$FacVar3),2)*100)
ggplot(bartableprop,aes(x=Var2,y=Freq,fill=Var1))+geom_bar(stat="identity")##added stat="identity"
```



Two Variables: One Factor and One Numeric Variable

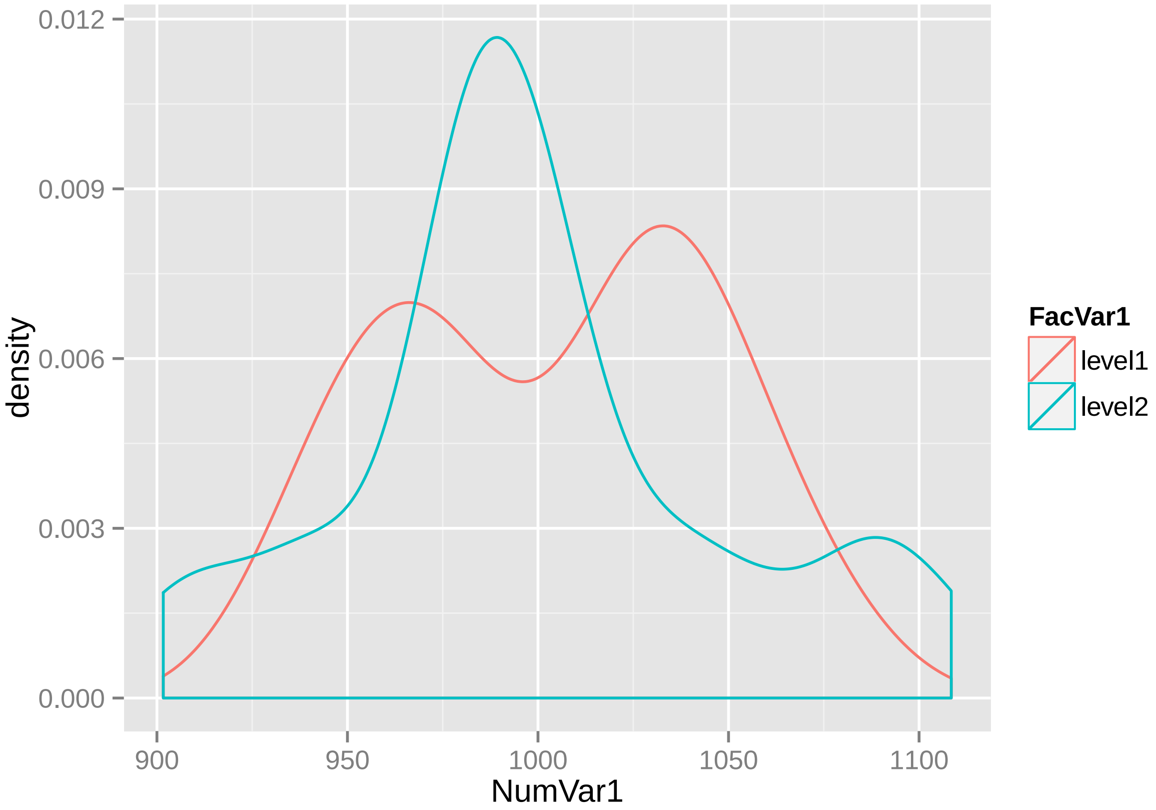
Box Plot of One Factor and One Numeric Variable

```r
ggplot(simData,aes(x=FacVar1,y=NumVar1))+geom_boxplot()
```



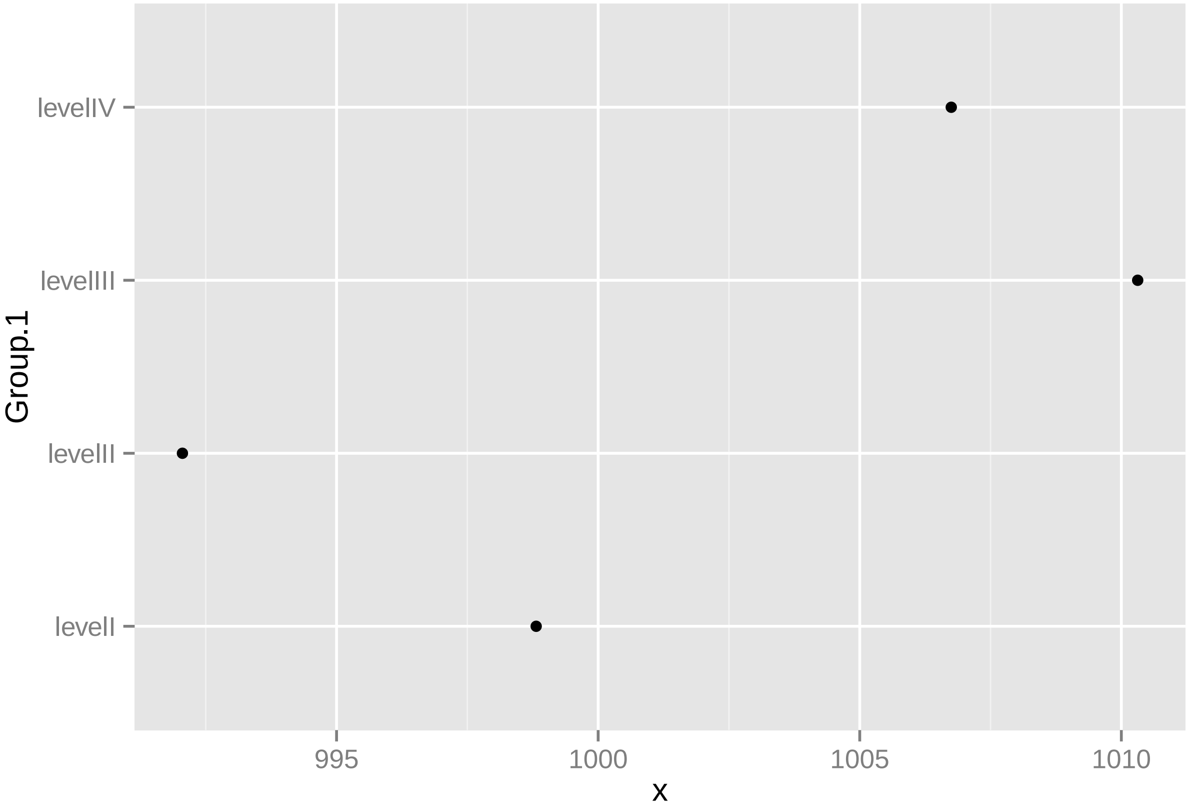
Density Plot of One Factor and One Numeric Variable

```
ggplot(simData,aes(x=NumVar1,color=FacVar1))+geom_density()
```



Dot Chart of One Factor and One Numeric Variable

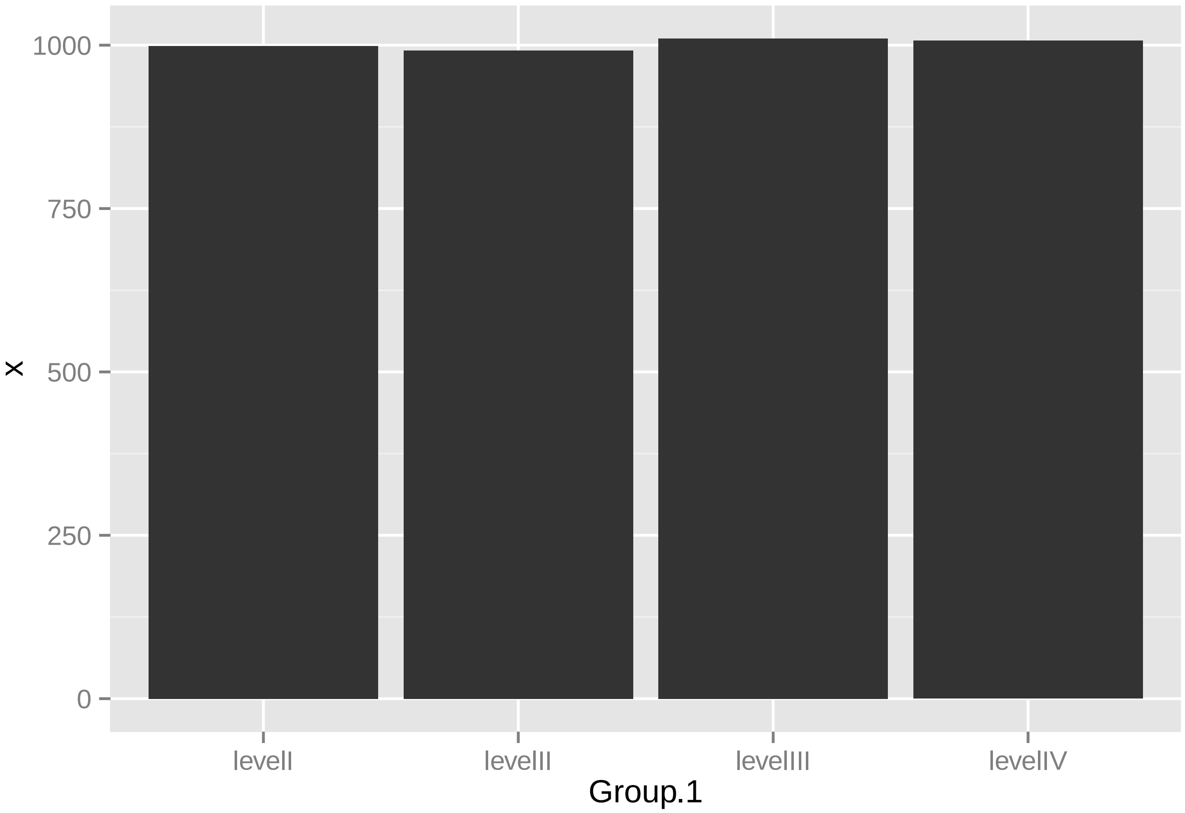
```
meanagg = aggregate(simData$NumVar1, list(simData$FacVar3), mean) ##aggregate function groups data by tl
ggplot(meanagg,aes(x=Group.1,y=x))+geom_point()+coord_flip() ## Dot Chart equivalent
```



The **coord_flip()** function flips cartesian coordinates so that horizontal becomes vertical and vice-versa. This is particularly useful for creating boxplots and other horizontal, interval geoms.

Bar Plot of One Factor and One Numeric Variable

```
meanagg = aggregate(simData$NumVar1, list(simData$FacVar3), mean)
ggplot(meanagg,aes(x=Group.1,y=x))+geom_bar(stat="identity")
```

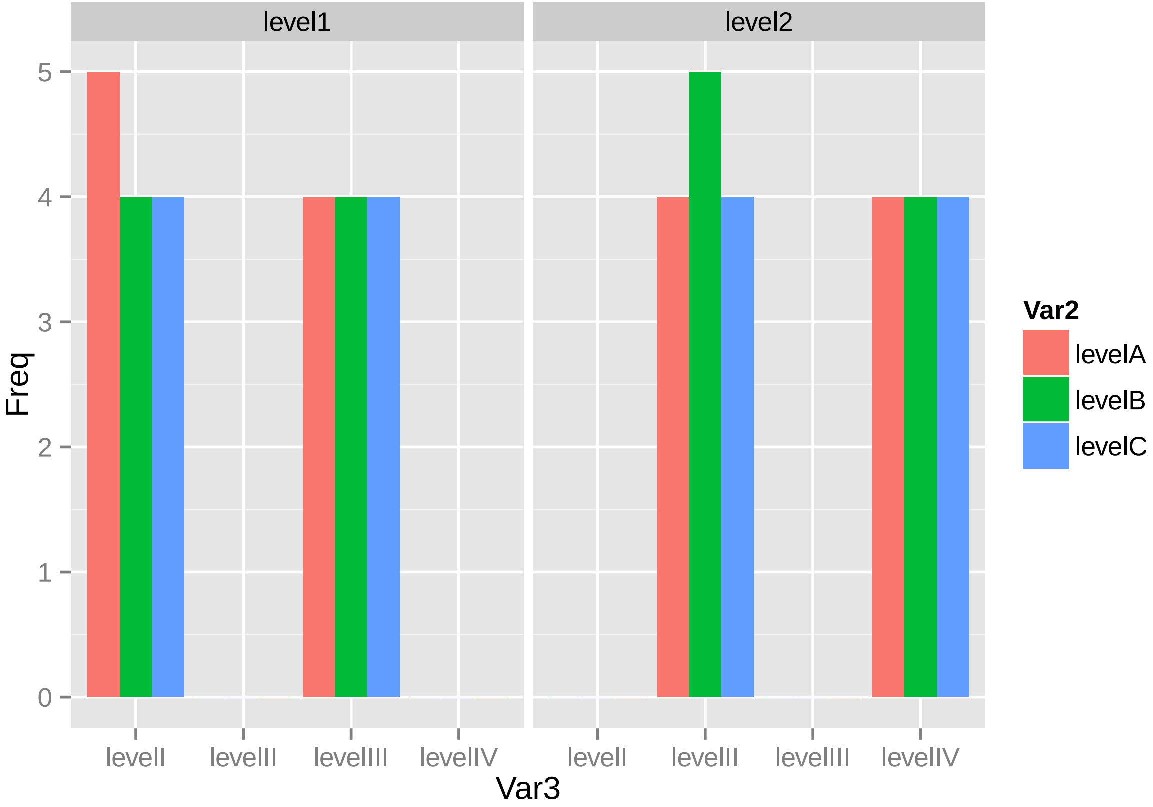


```
##added stat="identity"
```

Three Variables: Three Factor Variables

Bar Plot of Three Factor Variables

```
Threebartable = as.data.frame(table(simData$FacVar1, simData$FacVar2, simData$FacVar3)) ## CrossTab
ggplot(Threebartable,aes(x=Var3,y=Freq,fill=Var2))+geom_bar(position="dodge",stat="identity")+facet_wra
```

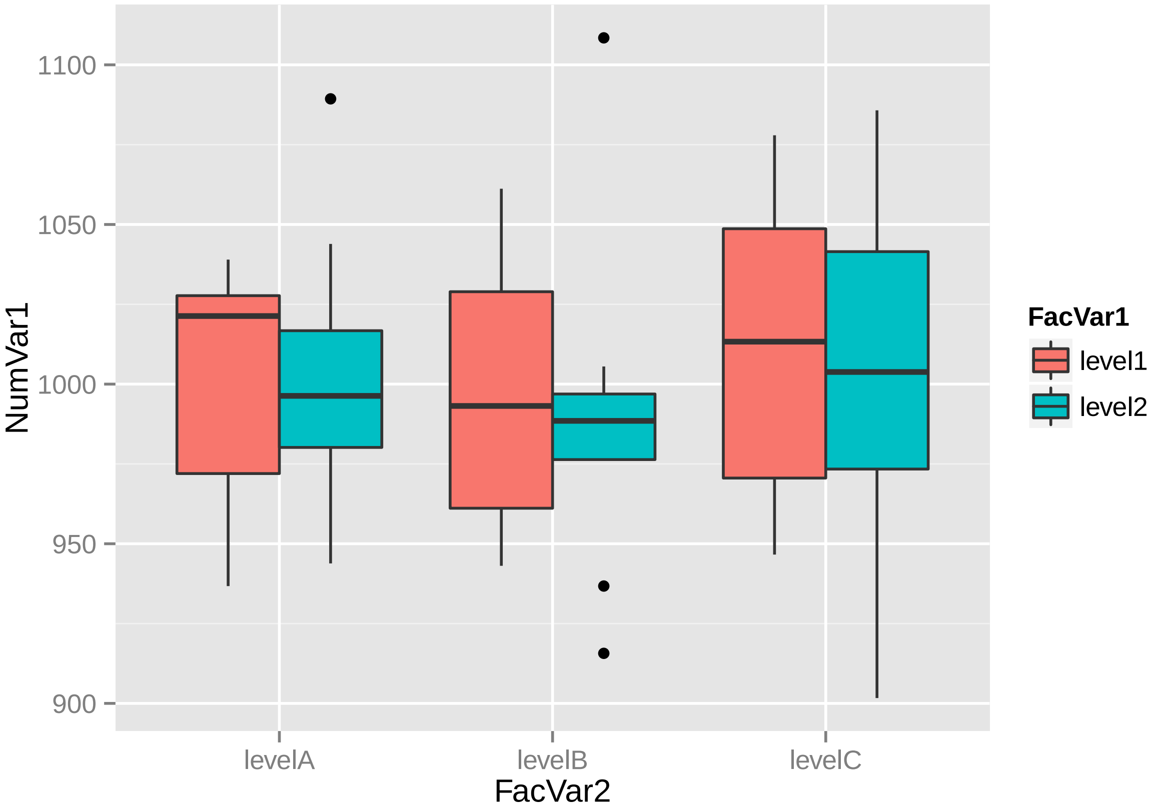


The facet_wrap() function wraps a 1d ribbon of panels into a 2d ribbon.

Three Variables: One Numeric and Two Factor Variables

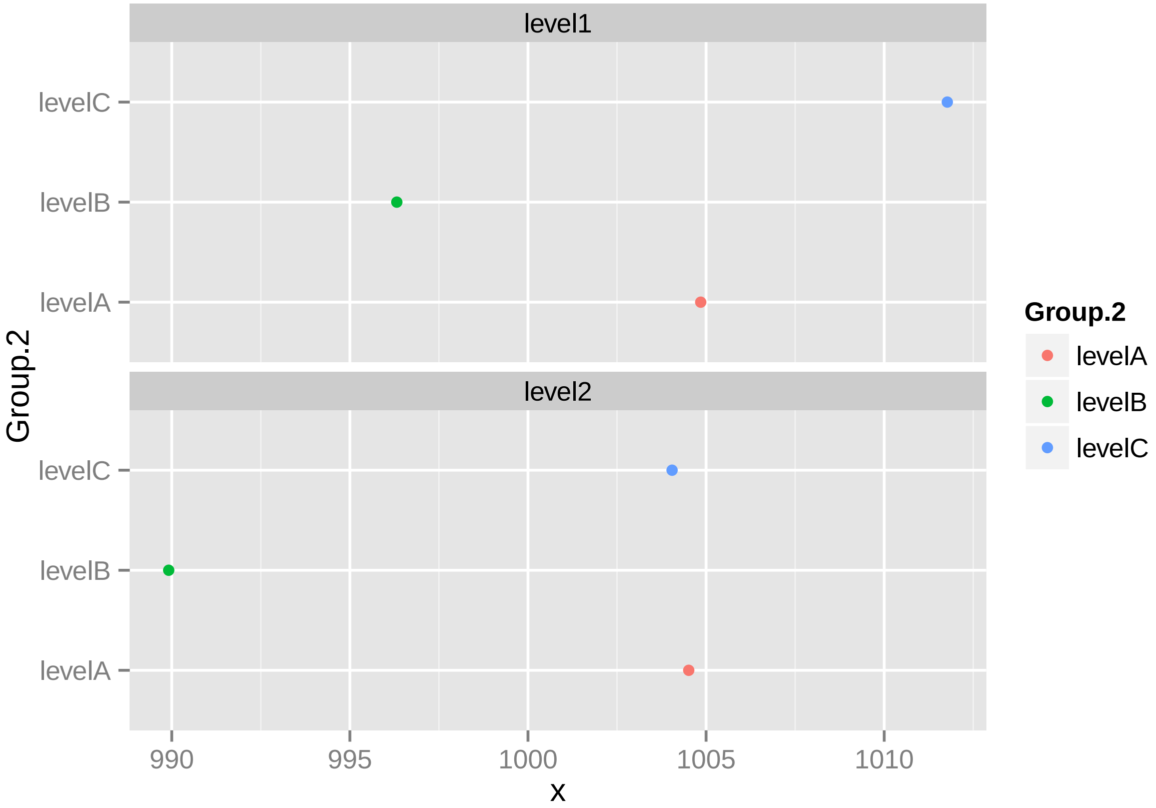
Box Plot of One Numeric and Two Factor Variables

```
## boxplot of NumVar1 over an interaction of 6 levels of the combination of FacVar1 and FacVar2
ggplot(simData,aes(x=FacVar2,y=NumVar1, fill=FacVar1))+geom_boxplot()
```



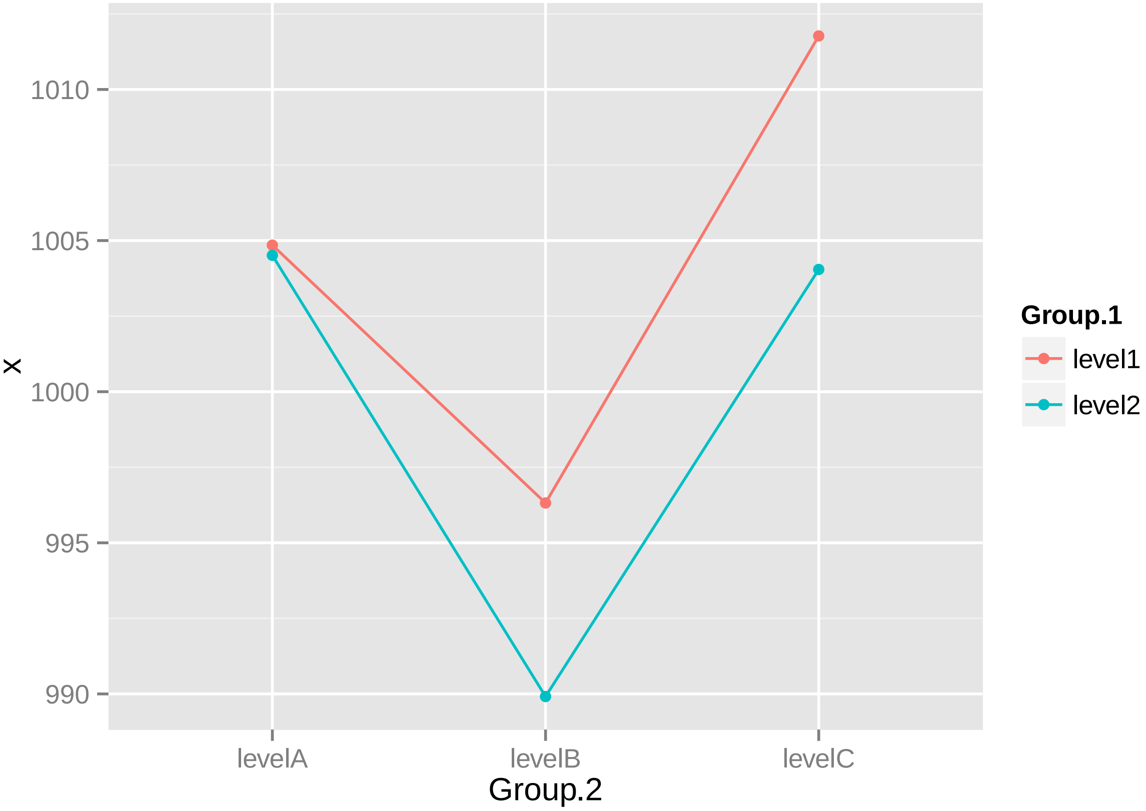
Dotchart of One Numeric and Two Factor Variables

```
## Mean of 1 Numeric over levels of two factor vars
meanaggg = aggregate(simData$NumVar1, list(simData$FacVar1, simData$FacVar2), mean)
## Dot Chart equivalent
ggplot(meanaggg,aes(x=Group.2,y=x,color=Group.2))+geom_point()+coord_flip()+facet_wrap(~Group.1, ncol=1
```



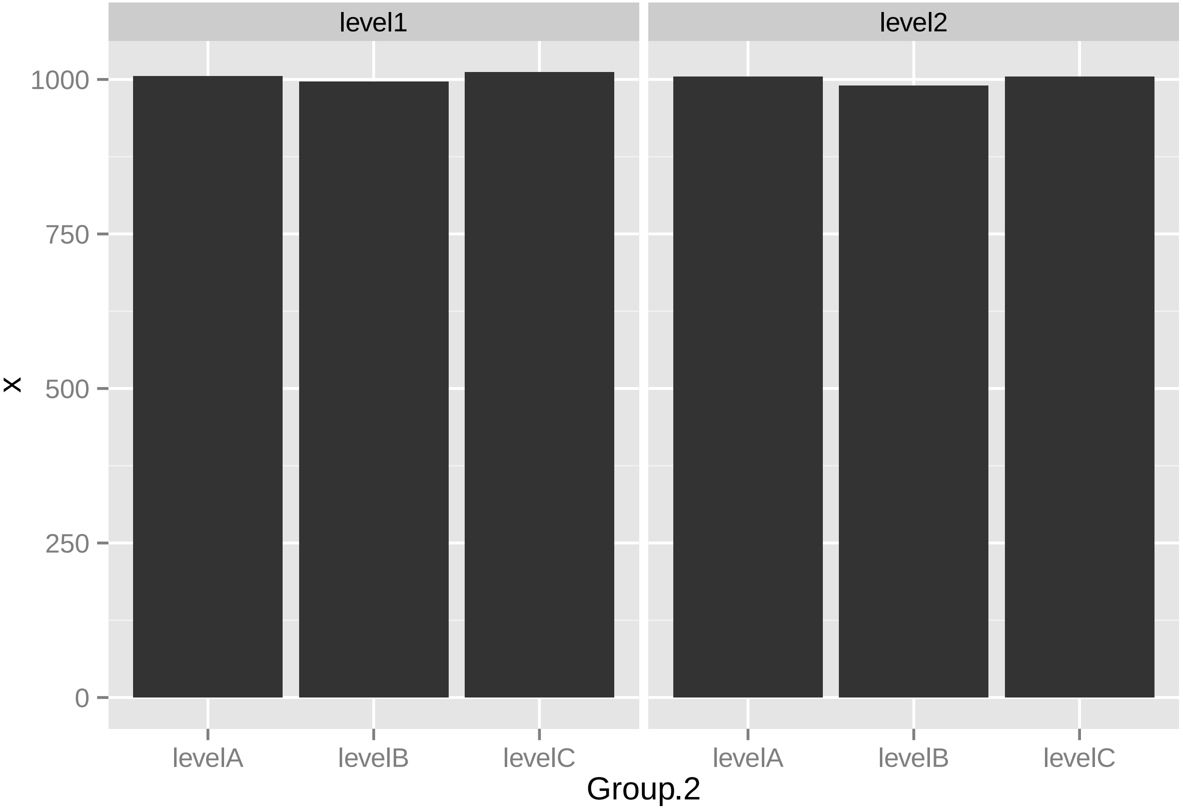
Interaction Line Chart of One Numeric and Two Factor Variables

```
## Interaction chart - line chart
ggplot(meanaggg,aes(x=Group.2,y=x,color=Group.1, group=Group.1))+geom_point()+geom_line()
```



Bar Plot of One Numeric and Two Factor Variables

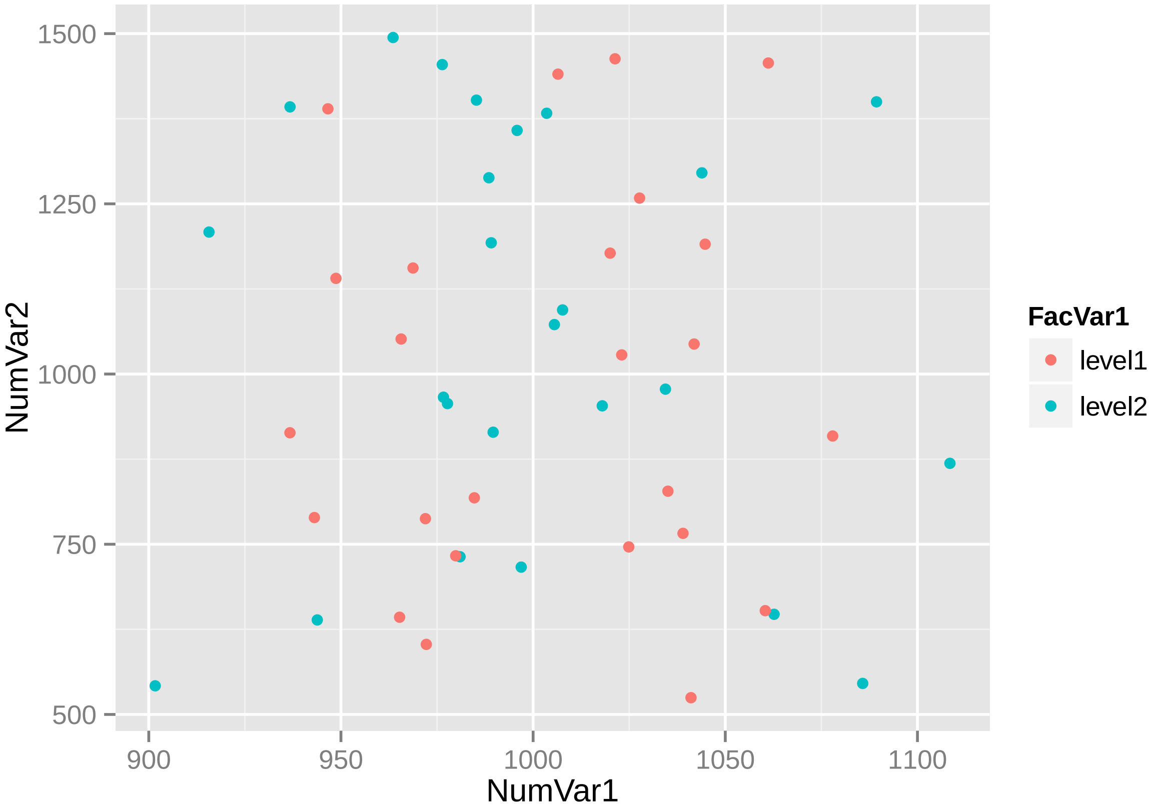
```
ggplot(meanaggg,aes(x=Group.2,y=x))+geom_bar(stat="identity")+facet_wrap(~Group.1)
```



Three Variables: Two Numeric and One Factor Variable

Scatter Plot of Two Numeric and One Factor Variable

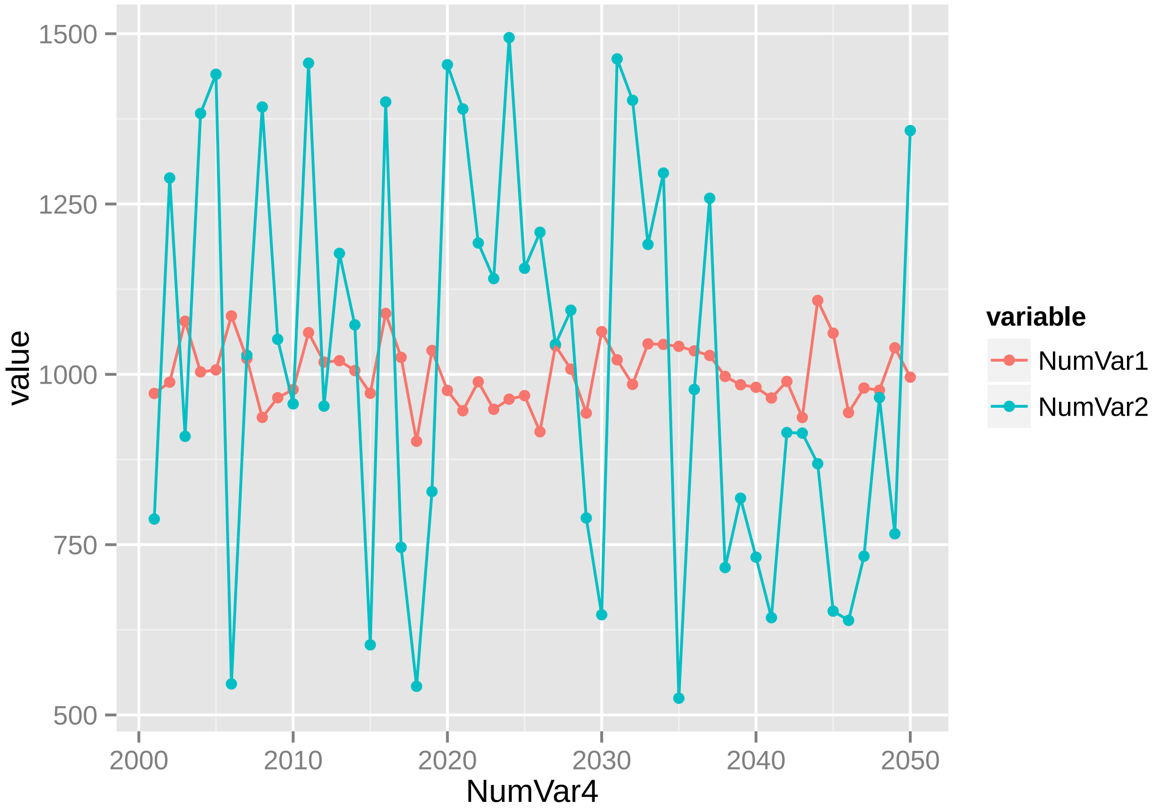
```
ggplot(simData,aes(x=NumVar1,y=NumVar2,color=FacVar1))+geom_point()
```



```
## Scatter plot with color identifying the factor variable
```

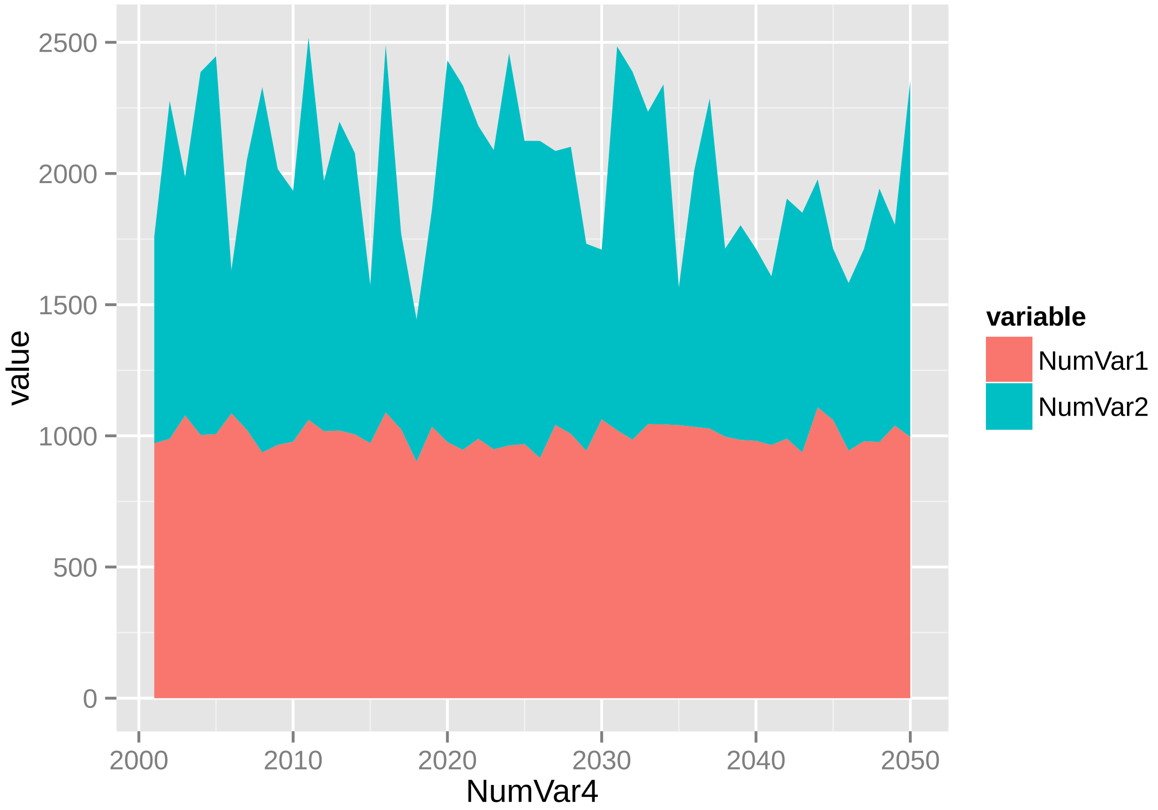
#Three Variables: Three Numeric Variables ##Name this Graph

```
# NumVar4 is 2001 through 2050... possibly, a time variable - use that as the x-axis
simtmpp=simData[,c(4,5,7)]
simtmppmelt=melt(simtmpp,id=c("NumVar4"))
ggplot(simtmppmelt,aes(x=NumVar4,y=value,color=variable,group=variable))+geom_point()+geom_line()
```



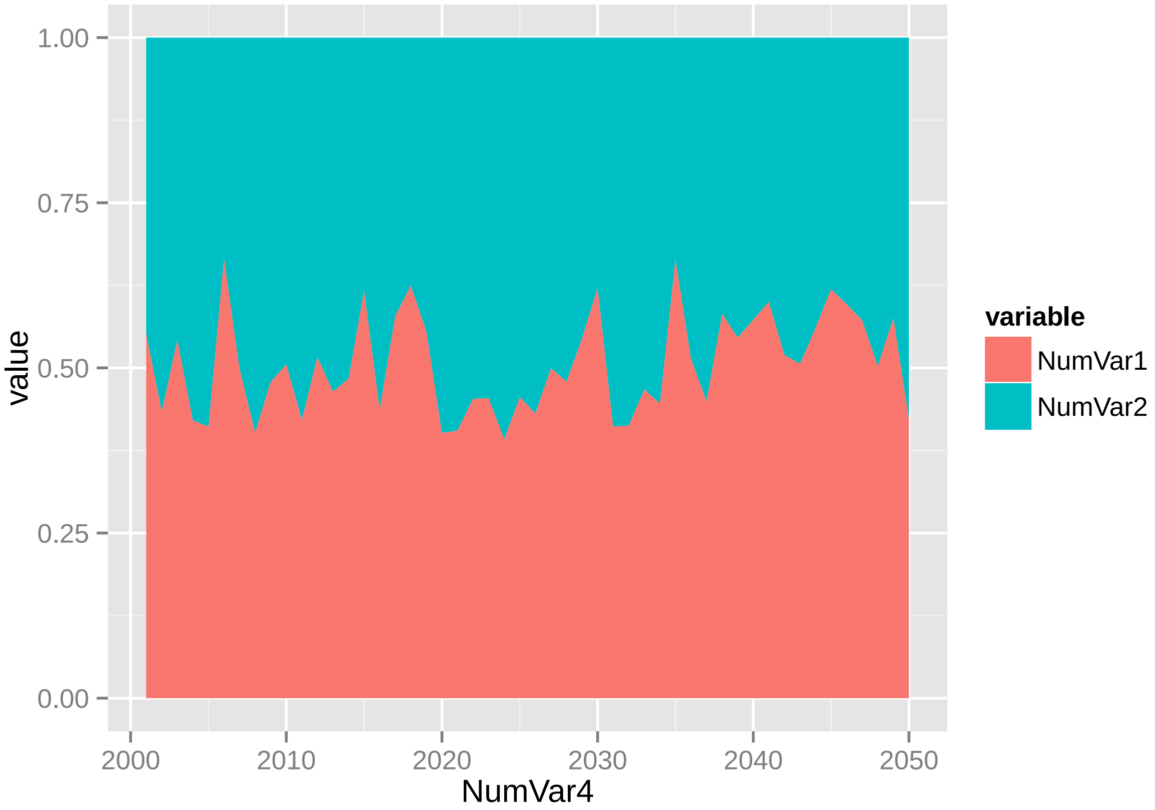
Stacked Area Graph of Three Numeric Variables

```r
ggplot(simtmppmelt,aes(x=NumVar4,y=value,fill=variable))+geom_area(position="stack")
```



Stacked Area Graph of Three Numeric Variables(100%)

```
ggplot(simtmppmelt,aes(x=NumVar4,y=value,fill=variable))+geom_area(position="fill")
```


value
NumVar4
variable
NumVar1
NumVar2