Graphics in R

Ira Sharenow

January 2, 2019

```
library(ggplot2) # graphing library
library(RColorBrewer)  # nice colors
```

R Markdown

This is an R Markdown document.
The purpose of this document is to show R users how to create graphs using both base R and ggplot2. In most case I will be demonstrating a graph via base R and then doing something similar with ggplot2. The ggplot2 graphs will all use the ggplot function, and none of the base R graphs will use the ggplot function.
I have divided this document by types of graphs. Generally speaking the first graph in each section is fairly straigtforward, but then the graphs become more complex/have more features/more default parameters are modified.
Another point. Typically, I generated the data by creating data frames. Then I displayed the data if it is not too big. Then I graphed the data. When appropriate, I used the set.seed() function so that when I used a random function, the data will be reproducible.
I do not provide a lot of details. Online help via ?plot or ?par is a good way to go as are Google searches.

Table of Contents

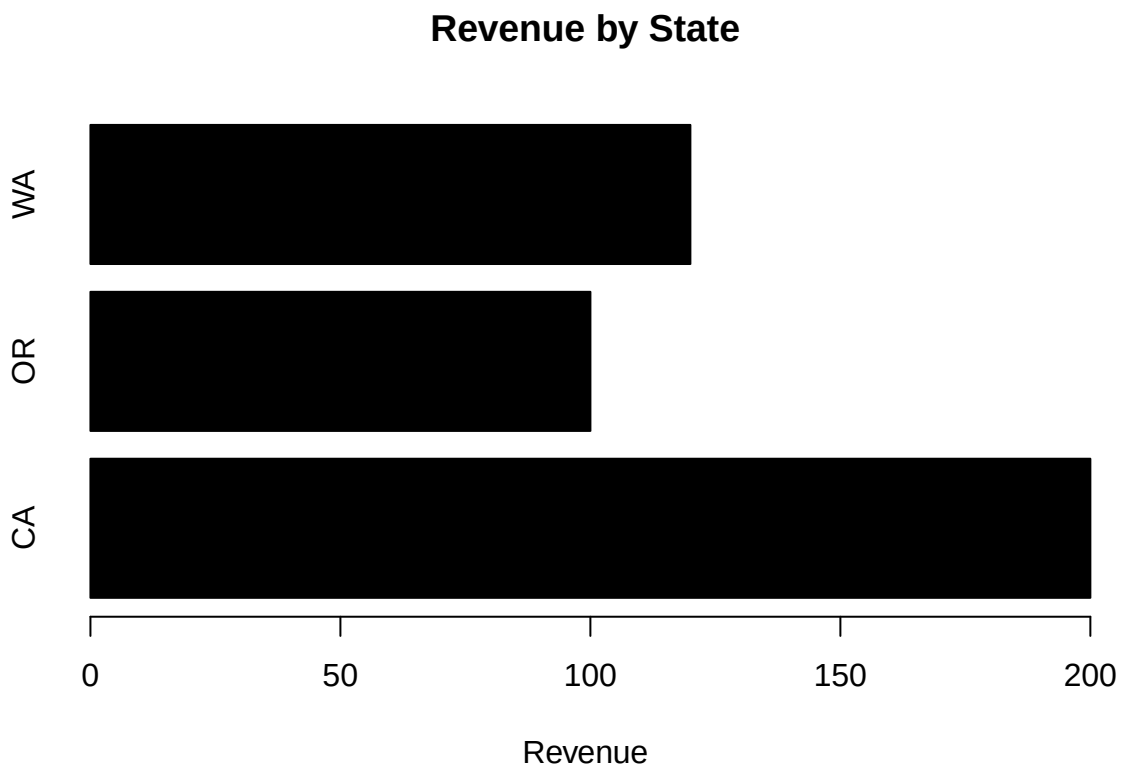
    1. Basic pie chart
    2. Pie chart with extra formatting
7. Plotting user defined functions
    1. Basic logistic curve
    2. Basic logistic curve plus math expression annotation in ggplot2
8. Faceting with ggplot2
    1. All data, no faceting
    2. Faceted chart with one factor (break data into subsets and then graph)
    3. Faceted chart with one factor (break data into subsets and then graph)
    4. Faceted chart with two factors (break data into subsets and then graph)

Bar graph 1
Bar graph 2

```
dfBar1 = data.frame(State = c('CA', 'OR', 'WA'), Revenue = c(200, 100, 120))
dfBar1
```

```
##   State Revenue
## 1    CA     200
## 2    OR     100
## 3    WA     120
```

```
barplot(dfBar1$Revenue, names.arg = dfBar1$State, col = "black", horiz = TRUE,
        main = "Revenue by State", xlab = "Revenue")
```

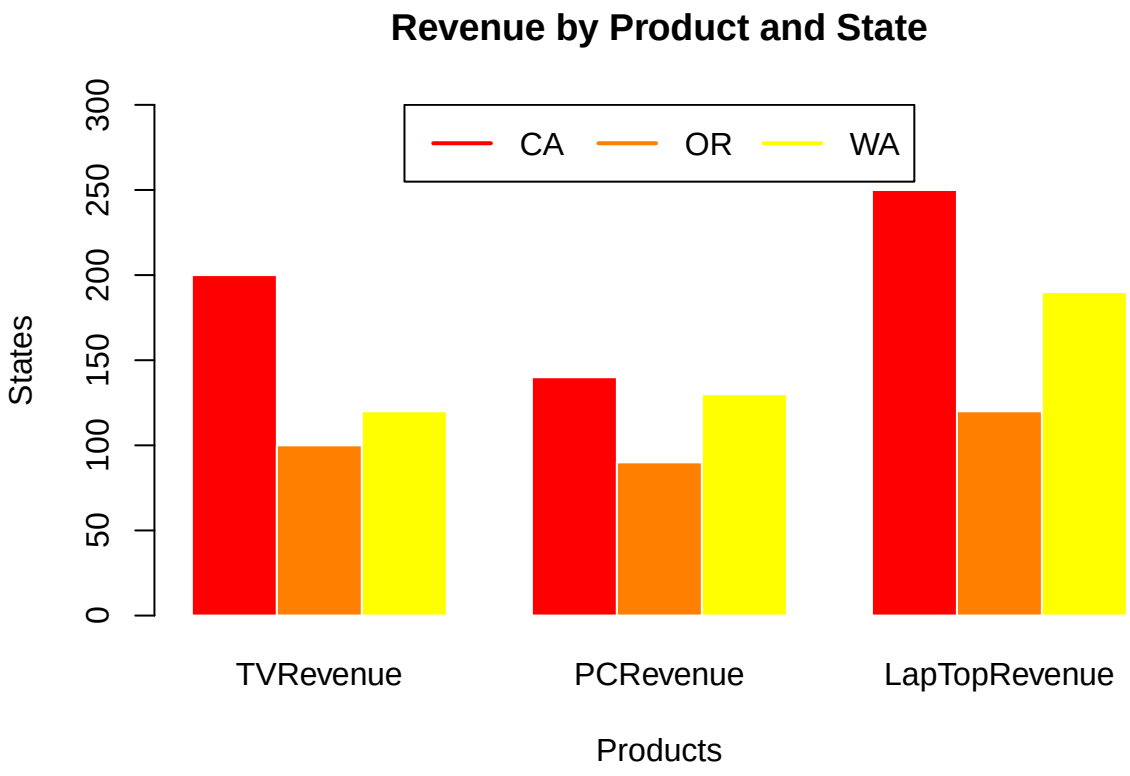



```
dfBar2 = data.frame(State = c('CA', 'OR', 'WA'), TVRevenue = c(200, 100, 120), PCRevenue = c(140,90, 13
                    LapTopRevenue = c(250, 120, 190))
```

```
dfBar2
```

```
##   State TVRevenue PCRevenue LapTopRevenue
## 1    CA       200       140           250
## 2    OR       100        90           120
## 3    WA       120       130           190
```

```
barplot(as.matrix(dfBar2[, 2:4]), beside = TRUE,
        col = heat.colors(3), border = "white", ylim = c(0, 300),
        main = "Revenue by Product and State", xlab = "Products", ylab = "States")
legend("top", legend = dfBar2$State, col = heat.colors(3), ncol = 3, lty=1,lwd=2 )
```

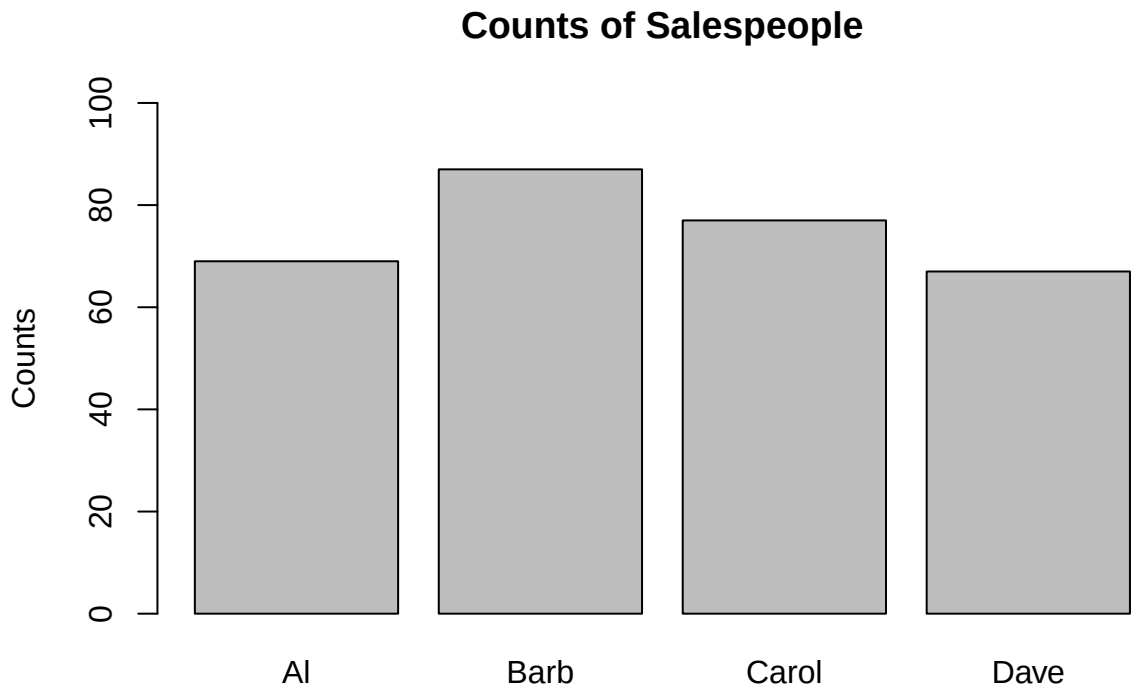


Bar graph 3 Bar graph 4

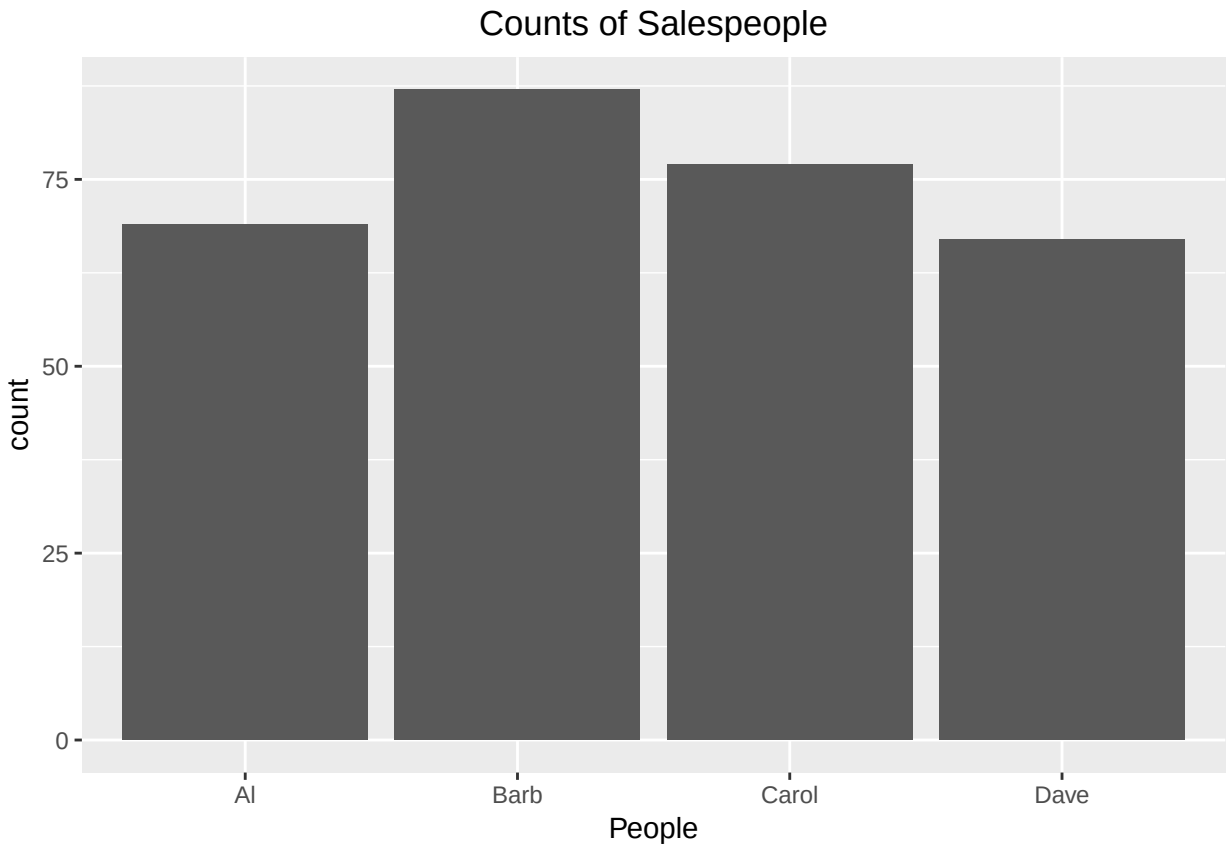
```
set.seed(2018)
Counts = 300
names4 = c("Al", "Barb", "Carol", "Dave")

salespeople = data.frame(People = sample(names4, Counts, replace = TRUE))
head(salespeople)
```

```
##   People
## 1   Barb
## 2   Barb
## 3     Al
## 4     Al
## 5   Barb
## 6   Barb
```

```r
barplot(table(salespeople$People), ylab = "Counts", main = "Counts of Salespeople", ylim = c(0,100))
```




```r
ggplot(salespeople, aes(x = factor(People))) + geom_bar() +
  ggtitle("Counts of Salespeople") + xlab("People")  + theme(plot.title = element_text(hjust=0.5))
```

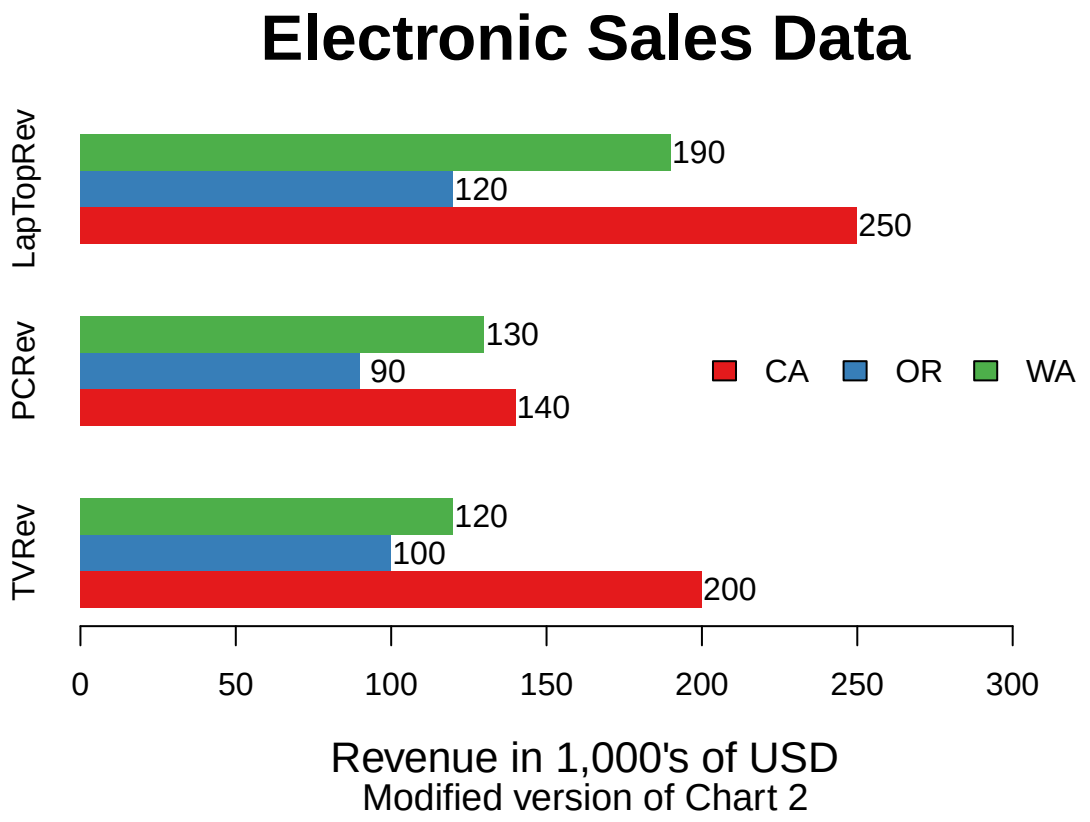


Bar graph 5

```
# Same as dfBar2
dfBar5 = data.frame(State = factor(c('CA', 'OR', 'WA'), ordered = TRUE),
                    TVRev = c(200, 100, 120), PCRev = c(140,90, 130),
                    LapTopRev = c(250, 120, 190))

dfBar5
```

```
##   State TVRev PCRev LapTopRev
## 1    CA   200   140       250
## 2    OR   100    90       120
## 3    WA   120   130       190
```

```
x = barplot(as.matrix(dfBar5[, 2:4]), beside = TRUE, horiz = TRUE,
        border = FALSE, space = c(0, 2), # no space between bars in same group; 5 units between groups
        col=brewer.pal(3,"Set1"),
        xlim = c(0, 325), xlab = "Revenue in 1,000's of USD",
        main = "Electronic Sales Data",
        sub = "Modified version of Chart 2",
        cex.main=2, cex.lab=1.3, cex.sub=1.2) # Change font size
legend("right", ncol = 3, legend = dfBar5$State, bty = "n",
       fill = brewer.pal(3, "Set1"))
y = as.matrix(dfBar5[, 2:4])
text(y+9, x,labels=as.character(y))
```

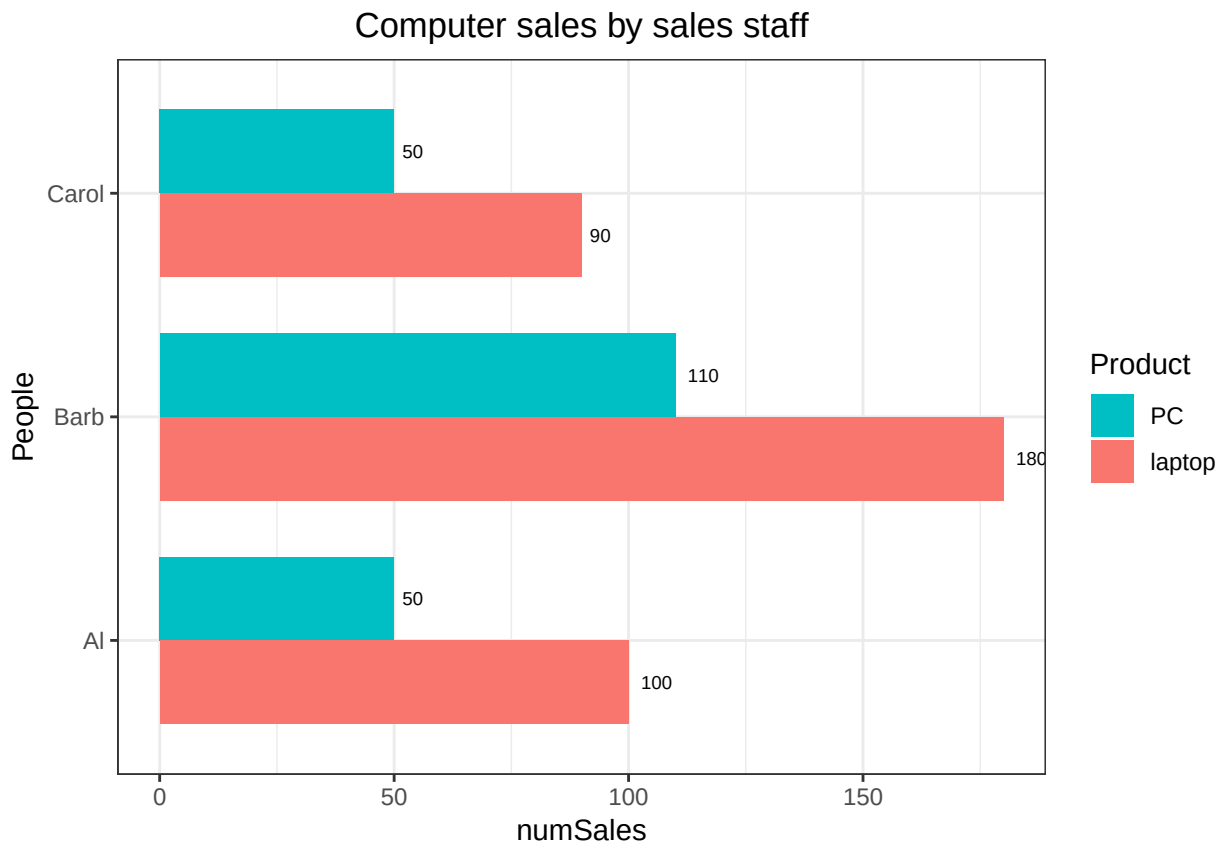


Bar graph 6
Bar graph 7

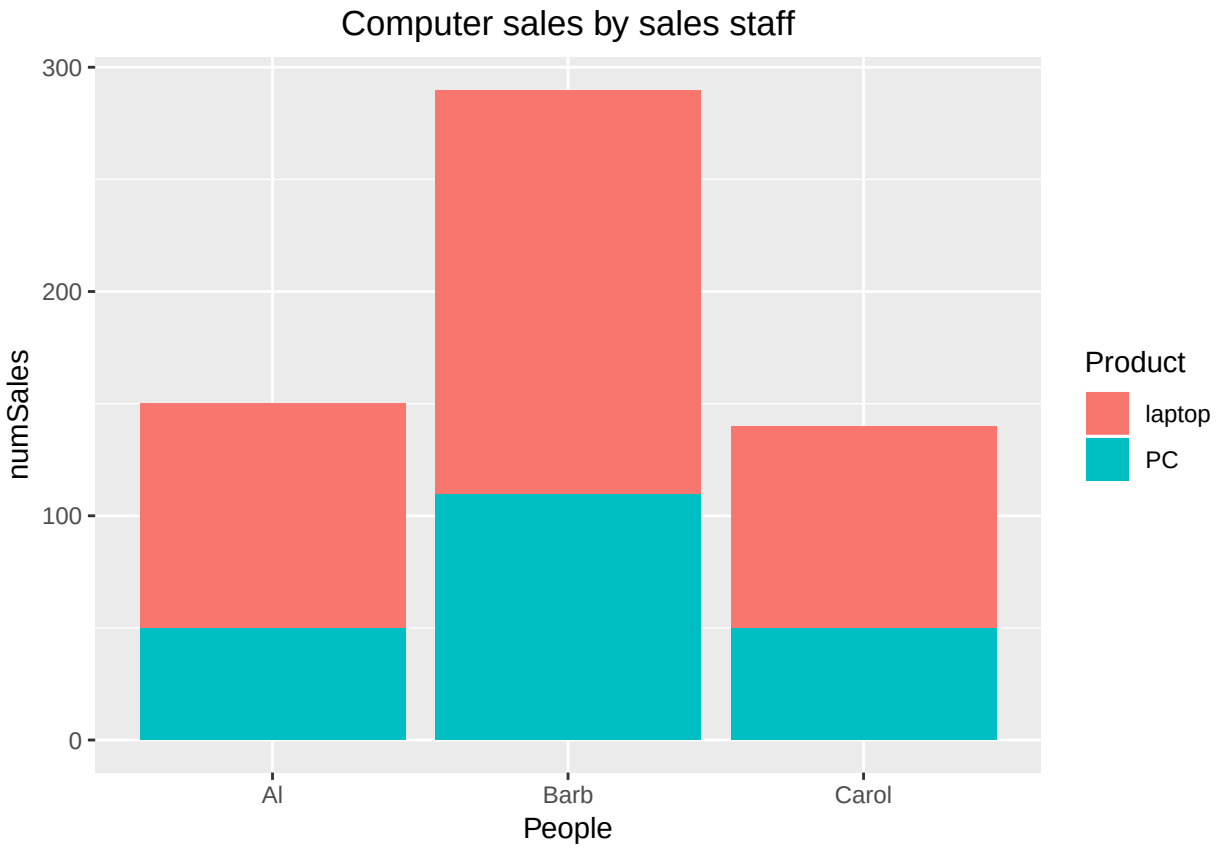
```
salespeople2 = data.frame(People = c('Al', 'Barb', 'Carol'), Product = c(rep('laptop',3), rep('PC', 3))
                          numSales = c(100, 180, 90, 50, 110, 50))
salespeople2
```

```
##   People Product numSales
## 1     Al  laptop      100
## 2   Barb  laptop      180
## 3  Carol  laptop       90
## 4     Al      PC       50
## 5   Barb      PC      110
## 6  Carol      PC       50
```

```
position = position_dodge(width = .75)
width = .75

ggplot(salespeople2, aes(x = People, y = numSales, label = numSales, fill = Product)) +
  geom_bar(width = width, stat = 'identity', position = position) +
  geom_text(hjust = -0.4, size = 2.5, position = position) +
  guides( fill = guide_legend( reverse = TRUE)) +
  coord_flip() +
  theme_bw() +
  ggtitle('Computer sales by sales staff') + theme(plot.title = element_text(hjust=0.5))
```




```
# Stacked bar graph
ggplot(salespeople2, aes(x = People, y = numSales, fill = Product)) + geom_bar(stat = "identity") +ggti
```

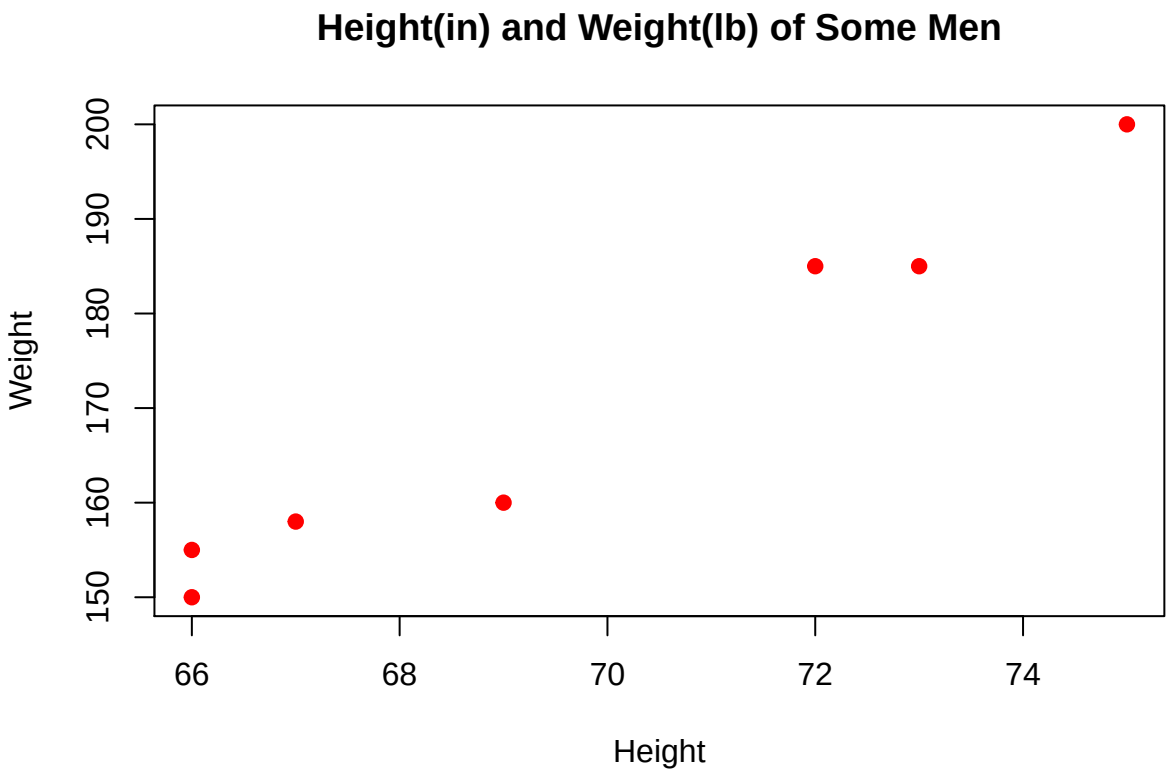


Point Graph 1

```
dfPoint1 = data.frame(Height = c(66, 66, 69, 72, 67, 73, 75),
                      Weight = c(155, 150, 160, 185, 158, 185, 200))
dfPoint1
```

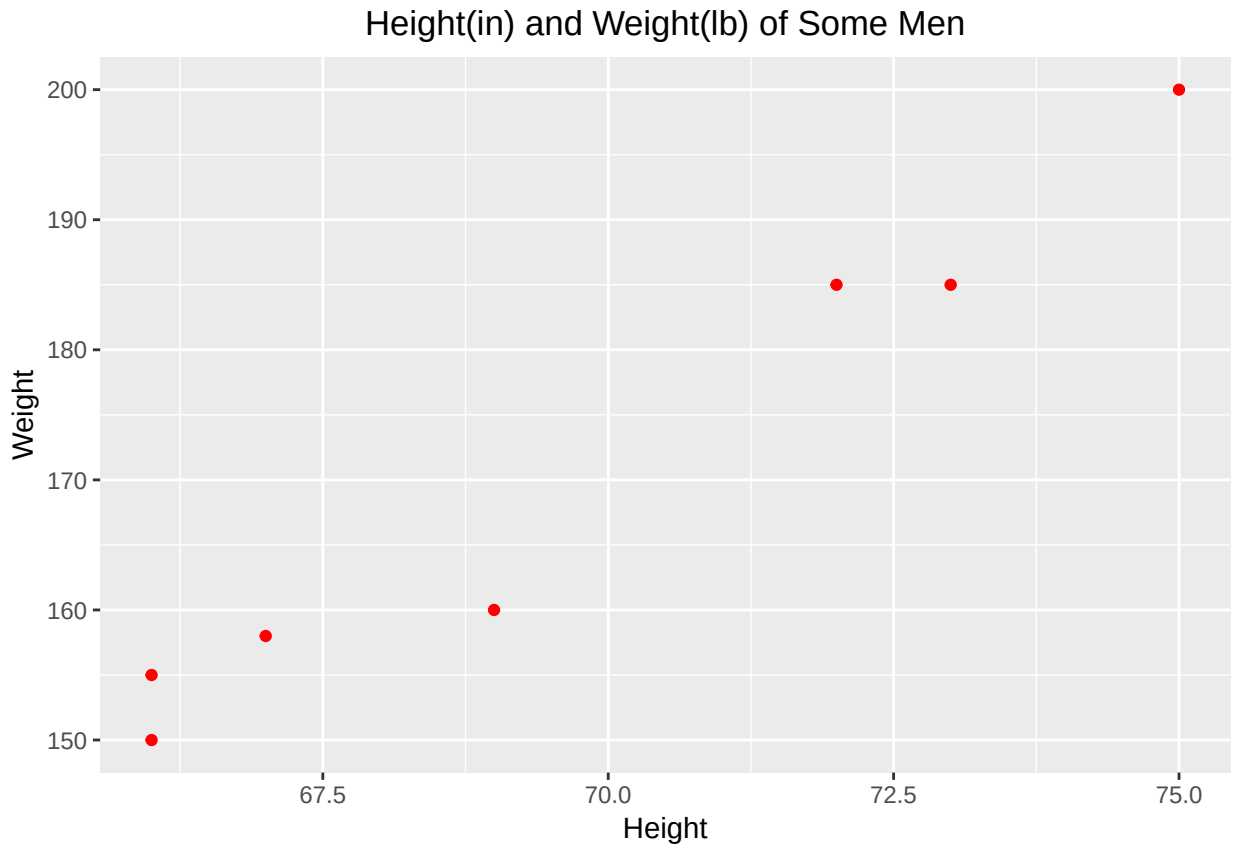
```
##   Height Weight
## 1     66    155
## 2     66    150
## 3     69    160
## 4     72    185
## 5     67    158
## 6     73    185
## 7     75    200
```

```
plot(dfPoint1$Weight ~ dfPoint1$Height, pch = 19, col = "red",
                      main = "Height(in) and Weight(lb) of Some Men",
                      xlab = "Height", ylab = "Weight")
```



Point Graph 2 (ggplot)

```
ggplot(dfPoint1, aes(x = Height, y = Weight)) + geom_point(color = "red") + ggtitle("Height(in) and Wei
```

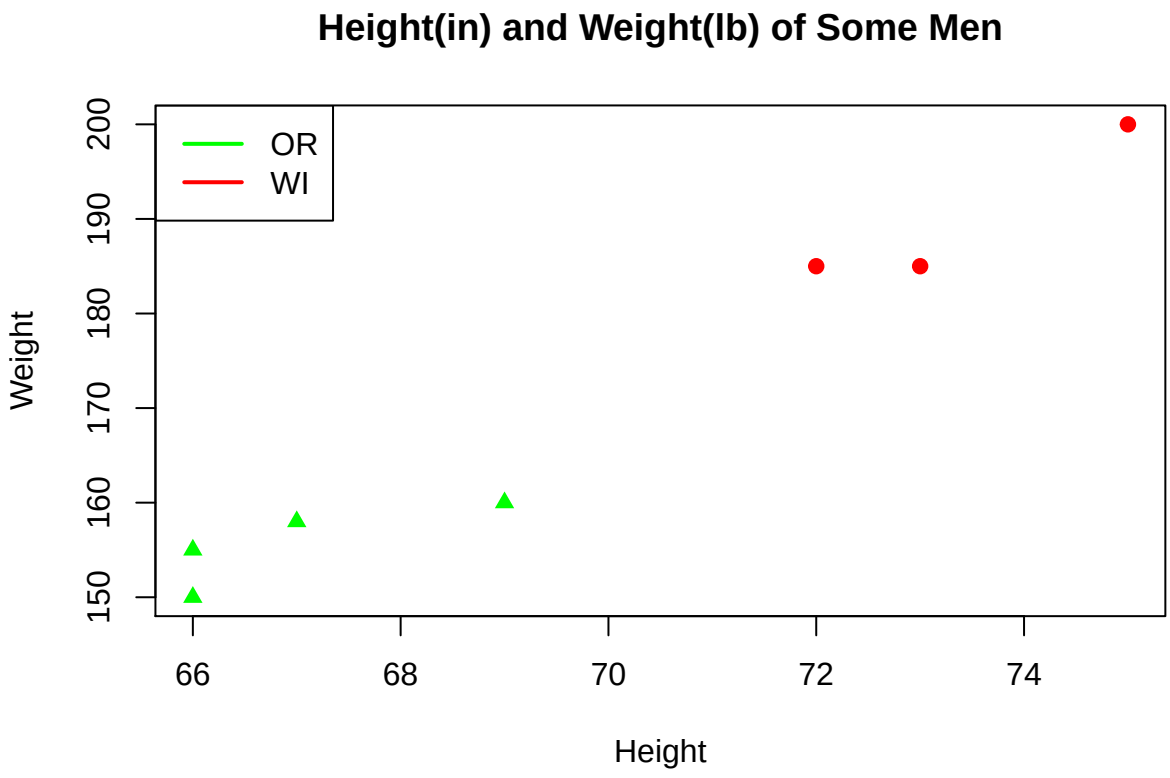


Point Graph 3

```
# Need to get everything matching up. Then need to do legend
# https://stackoverflow.com/questions/12919816/plotting-in-different-shapes-using-pch-argument
#
dfPoint3 = data.frame(Height = c(66, 66, 69, 72, 67, 73, 75),
                      Weight = c(155, 150, 160, 185, 158, 185, 200),
                      State = factor(c('OR', 'OR', 'OR', 'WI', 'OR', 'WI', 'WI')))
dfPoint3
```

```
##   Height Weight State
## 1     66    155    OR
## 2     66    150    OR
## 3     69    160    OR
## 4     72    185    WI
## 5     67    158    OR
## 6     73    185    WI
## 7     75    200    WI
```

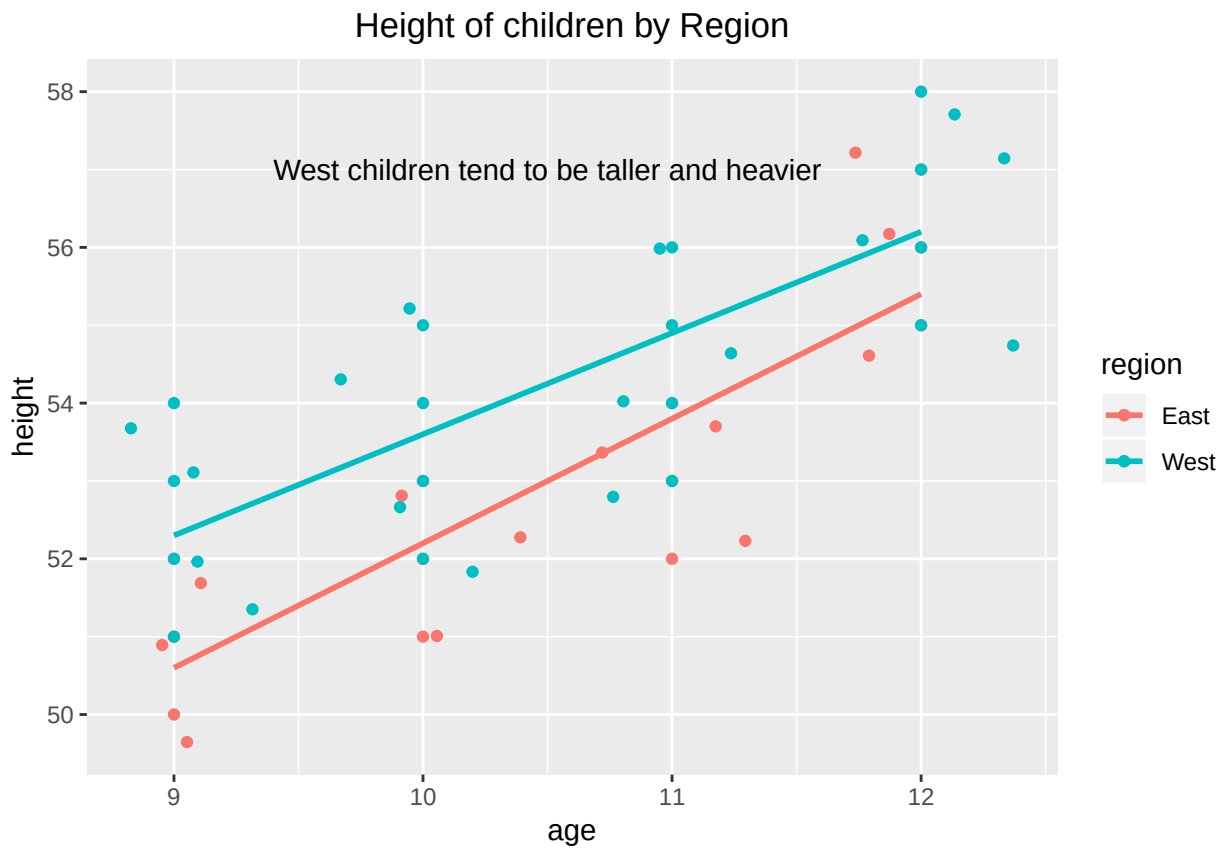
```
plot(dfPoint3$Weight ~ dfPoint3$Height, col = c('green', 'red')[as.numeric(dfPoint3$State)],
     pch = c(17, 19)[as.numeric(dfPoint3$State)],
     main = "Height(in) and Weight(lb) of Some Men",
     xlab = "Height", ylab = "Weight")
legend("topleft", legend = as.vector(unique(dfPoint3$State)), col = c('green', 'red'), lty=1,lwd=2)
```




```
# https://stackoverflow.com/questions/46390628/show-color-coded-legend-beside-scatter-plot
# https://stackoverflow.com/questions/8891172/matching-color-in-legend-in-r-plot <<<<<
```

Point Graph 4

```
dfPoint4 = data.frame(region = c(rep('East', 12), rep('West', 16)),
                      age = c(rep(9:12, each = 3), rep(9:12, each = 4)),
                      height = c(50:52, 51:53, 52:54, 55, 56, 57, 51:54,
                                 52:55, 53:56, 55:58))


p = ggplot(dfPoint4, aes(x = age, y = height, color = region)) + geom_point() + geom_jitter() +
  stat_smooth(method = lm, se = FALSE) +
  ggtitle("Height of children by Region") + theme(plot.title = element_text(hjust=0.5))

p + annotate("text", x = 10.5, y = 57, label = "West children tend to be taller and heavier")
```



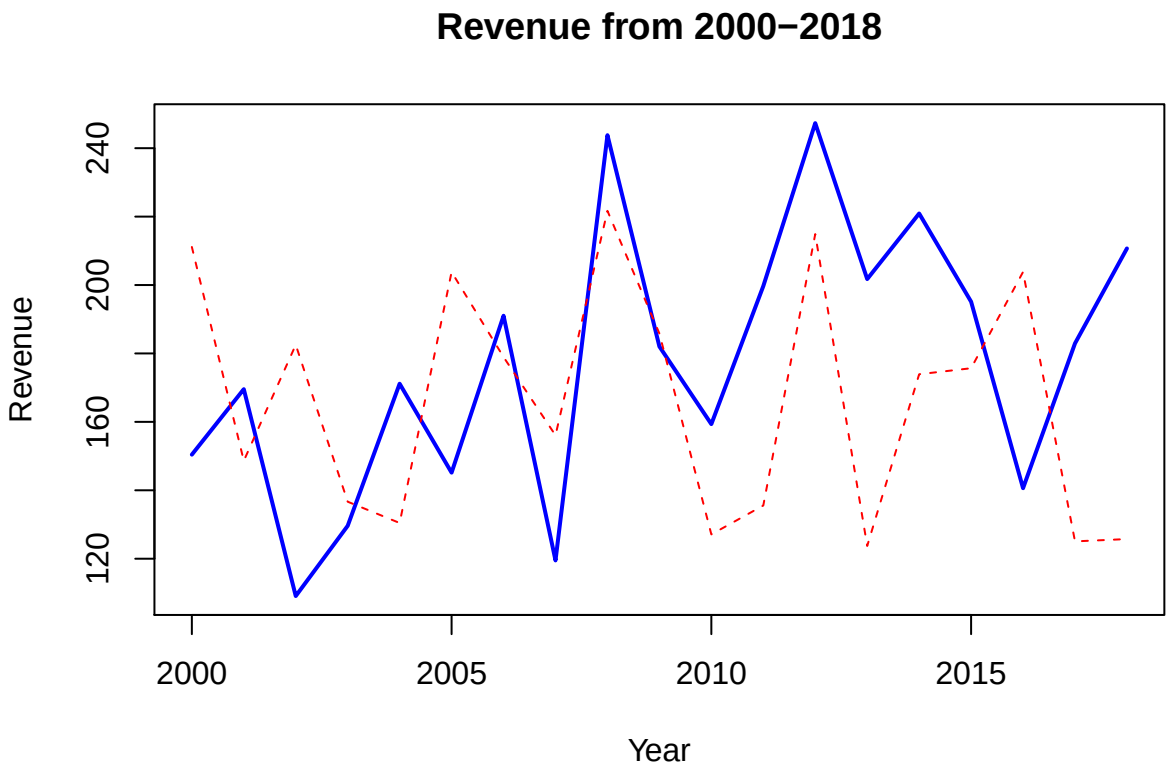
Line Graph 1

```
set.seed(2018) # for reproducibility
# create random uniforms between 100 and 250
# There are two revenue streams
dfLine1 = data.frame(Year = 2000:2018,
                     Revenue1 = c(runif(19, min = 100, max = 250)),
                     Revenue2 = c(runif(19, min = 120, max = 230)))
dfLine1
```

```
##    Year Revenue1 Revenue2
## 1  2000 150.4230 211.1240
## 2  2001 169.5585 148.6111
## 3  2002 109.0878 182.5196
## 4  2003 129.6150 136.6648
## 5  2004 171.1471 130.4431
## 6  2005 145.1573 203.8108
## 7  2006 191.0138 178.9521
## 8  2007 119.5018 156.0669
## 9  2008 243.7982 221.6792
## 10 2009 182.0274 185.8143
## 11 2010 159.3424 127.1100
## 12 2011 199.6808 135.5505
## 13 2012 247.3168 214.8859
## 14 2013 201.7323 123.7213
## 15 2014 220.9042 173.9225
## 16 2015 195.1270 175.6798
```

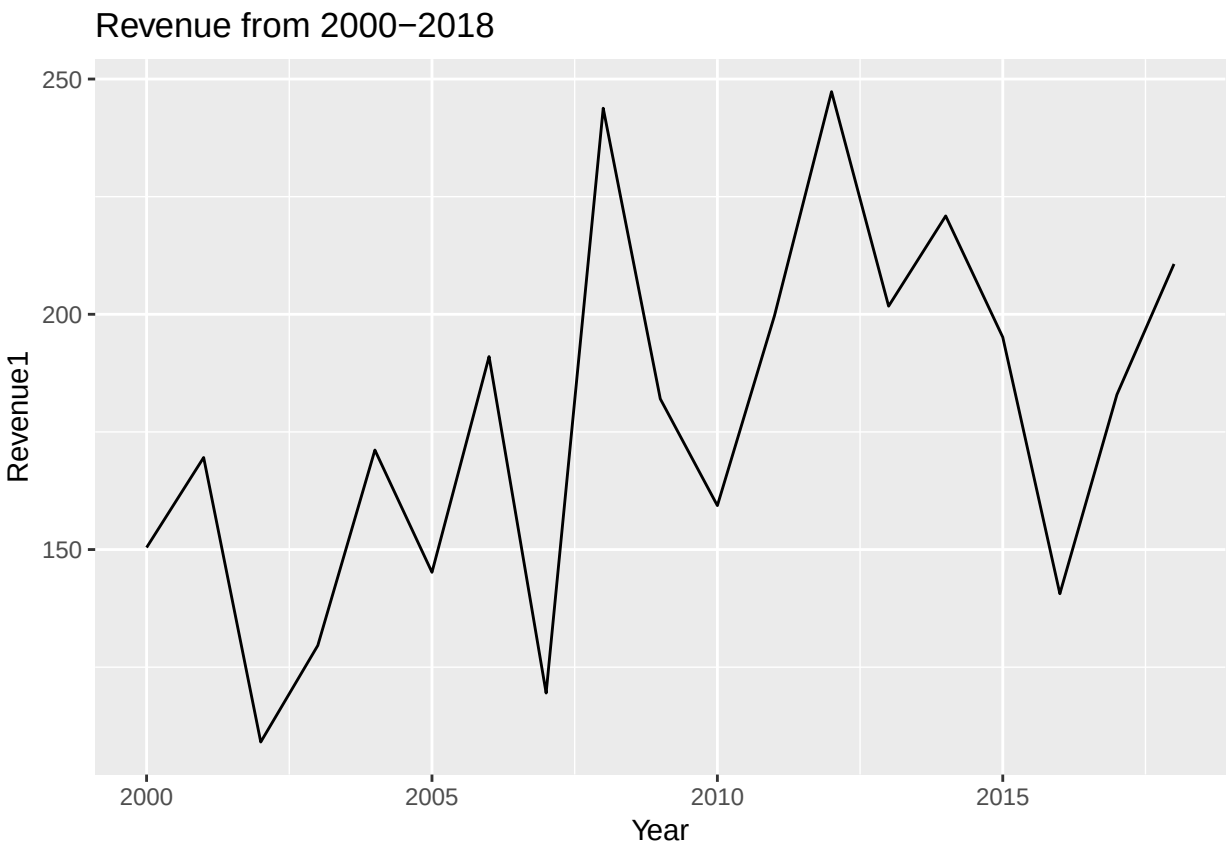
```
## 17 2016 140.6105 203.8238
## 18 2017 182.9356 125.0662
## 19 2018 210.6934 125.7391
```

```
plot(dfLine1$Year, dfLine1$Revenue1, type = "l",
     main = "Revenue from 2000-2018", xlab = "Year", ylab = "Revenue",
     col = "blue", lwd = 2)
lines(dfLine1$Year, dfLine1$Revenue2, col = "red", lty = 2)
```



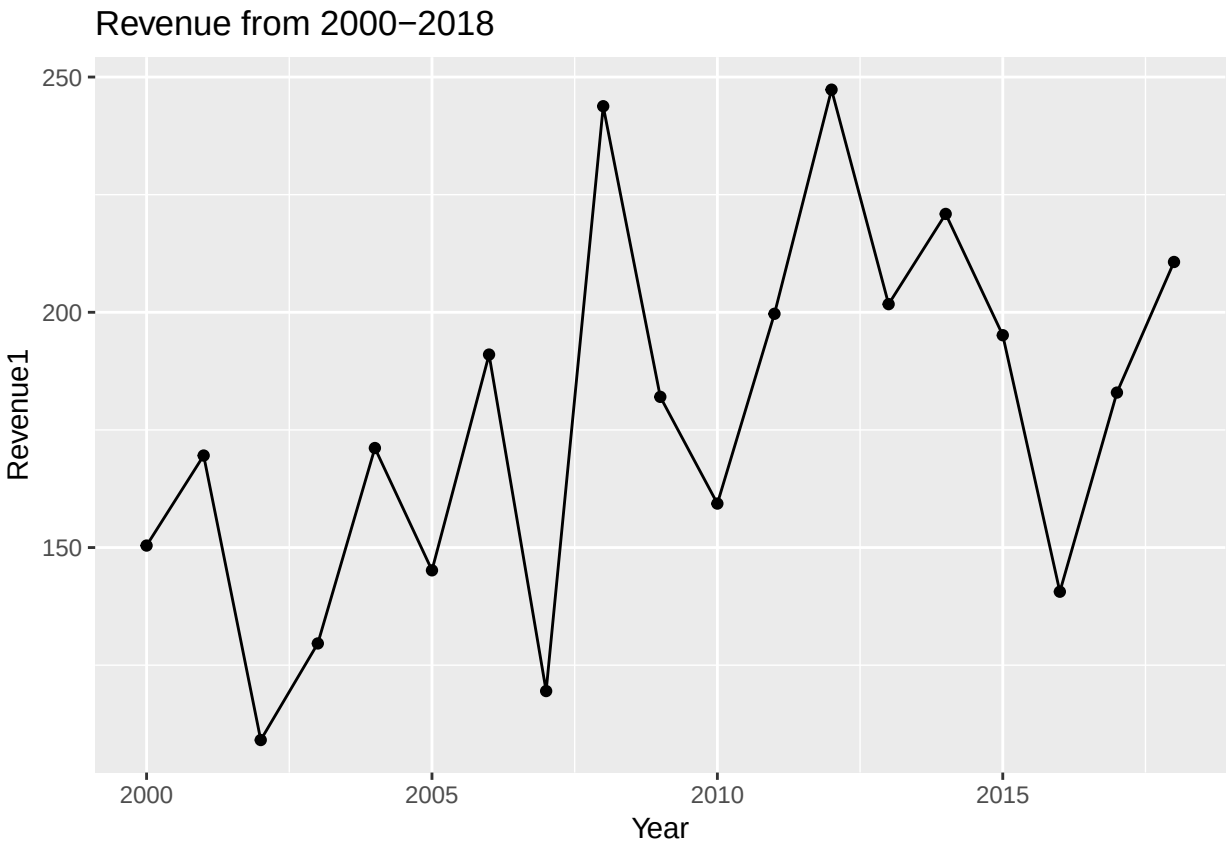
Line Graph 2 (ggplot)

```
ggplot(dfLine1, aes(x = Year, y = Revenue1)) + geom_line() + ggtitle("Revenue from 2000-2018")
```



Line Graph 3 (ggplot)

```
ggplot(dfLine1, aes(x = Year, y = Revenue1)) + geom_line() + geom_point() + ggtitle("Revenue from 2000-
```



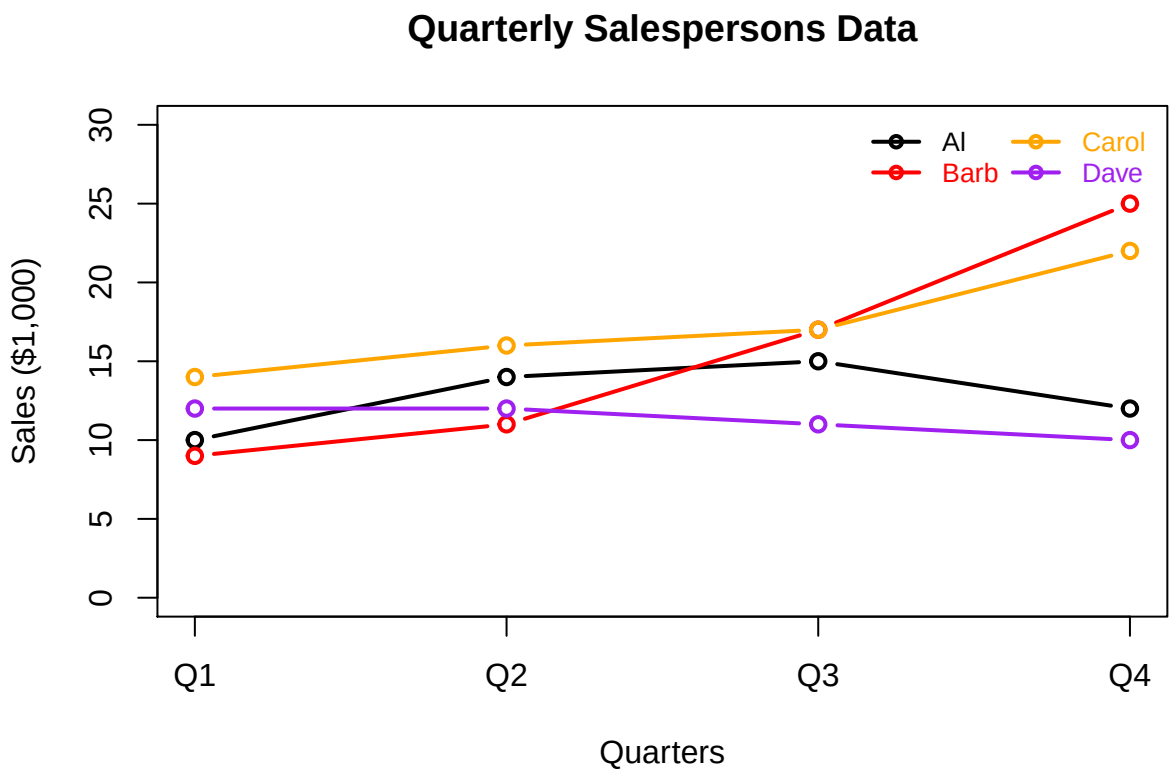
Line Graph 4

```
dfLine2 = data.frame(Quarters = paste('Q',1:4, sep = ""),
                     Al = c(10, 14, 15, 12),
                     Barb = c(9, 11, 17, 25),
                     Carol = c(14, 16, 17, 22),
                     Dave = c(12, 12, 11, 10))

dfLine2
```

```
##   Quarters Al Barb Carol Dave
## 1       Q1 10    9    14   12
## 2       Q2 14   11    16   12
## 3       Q3 15   17    17   11
## 4       Q4 12   25    22   10
```

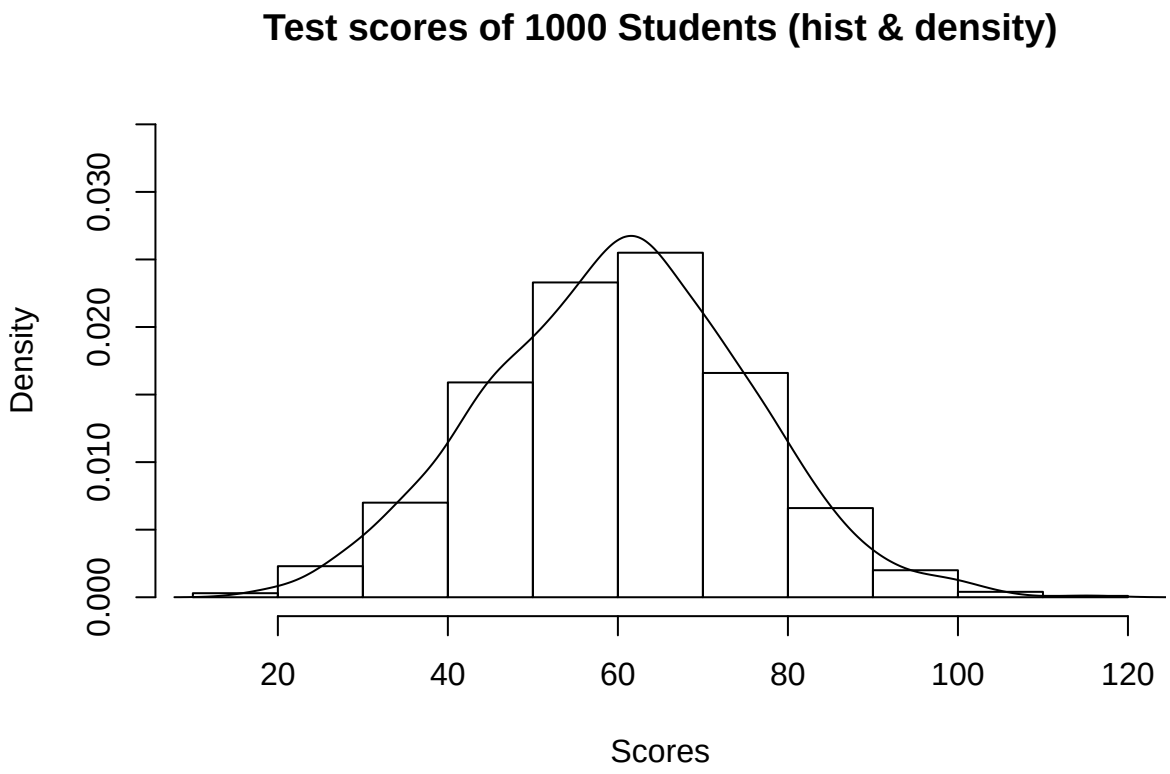
```
plot(dfLine2$Al,type="b",lwd=2,
     xaxt="n",ylim=c(0,30),col="black",
     xlab="Quarters",ylab="Sales ($1,000)",
     main="Quarterly Salespersons Data")
axis(1,at=1:length(dfLine2$Quarters),labels=dfLine2$Quarters)

lines(dfLine2$Barb,col="red",type="b",lwd=2)
lines(dfLine2$Carol,col="orange",type="b",lwd=2)
lines(dfLine2$Dave,col="purple",type="b",lwd=2)

legend("topright",legend=c("Al","Barb","Carol","Dave"),
```

```
       lty=1,lwd=2,pch=21,col=c("black","red","orange","purple"),
       ncol=2,bty="n",cex=0.8,
       text.col=c("black","red","orange","purple"),
       inset=0.01)
```



Histogram 1

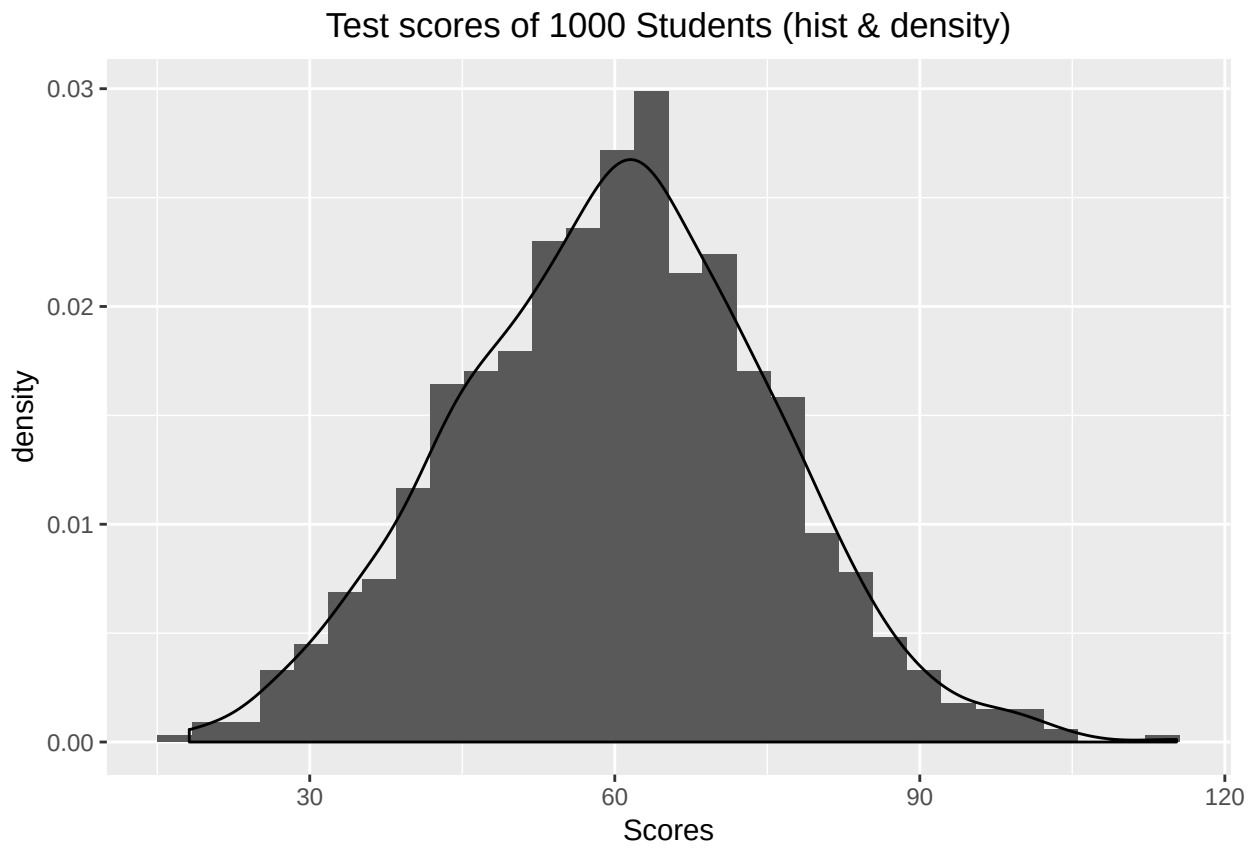
```
set.seed(2018)
dfHist1 = data.frame(Scores = rnorm(n = 1000, mean = 60, sd = 15))
hist(dfHist1$Scores, freq = FALSE, ylim = c(0, 0.035),
     main = "Test scores of 1000 Students (hist & density)", xlab = "Scores")
lines(density(dfHist1$Scores))
```



Histogram 2

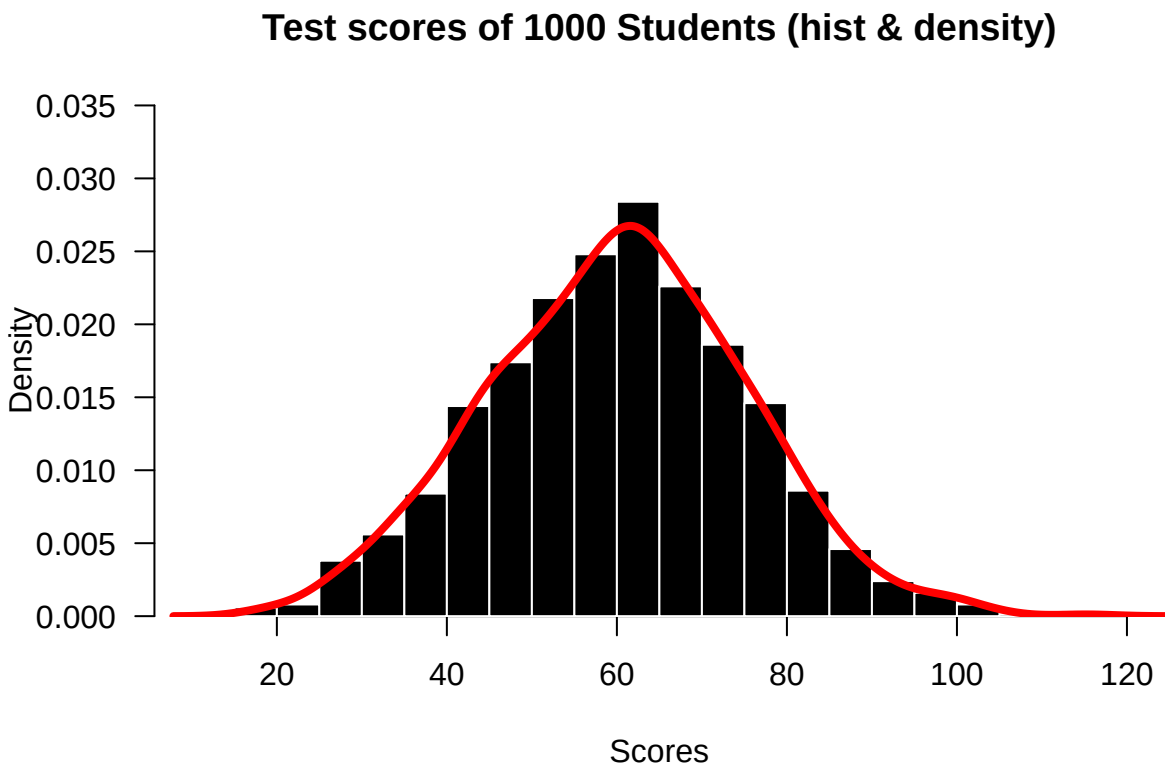
```
ggplot(dfHist1) + geom_histogram(aes(x = Scores, y = ..density..)) +
  geom_density(aes(x=Scores,y=..density..)) +
  ggtitle("Test scores of 1000 Students (hist & density)") + theme(plot.title = element_text(hjust=0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



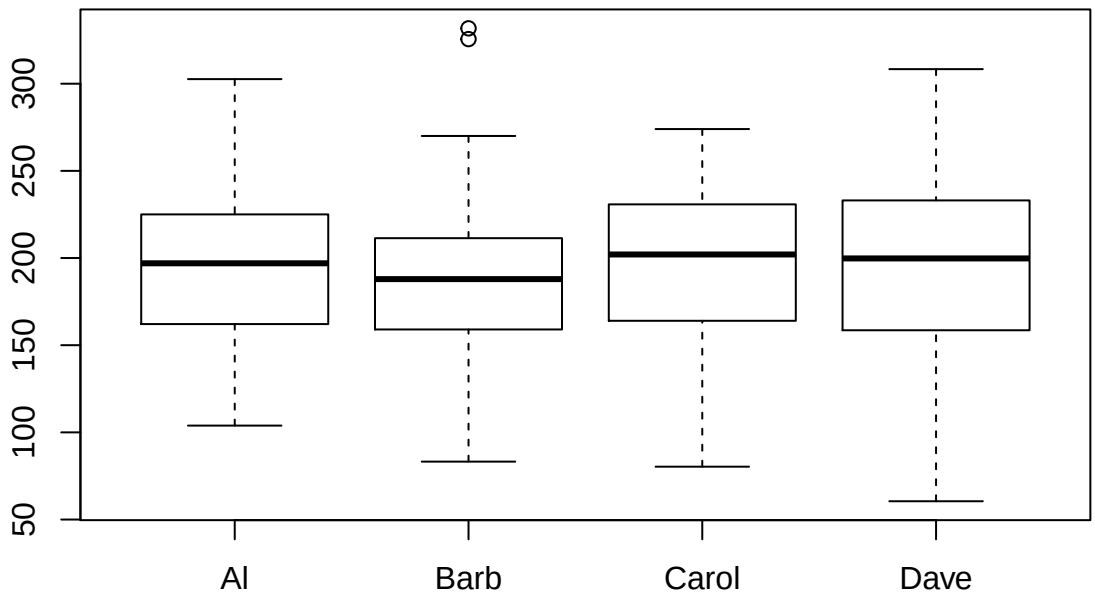
Histogram 3

```
set.seed(2018)
dfHist2 = data.frame(Scores = rnorm(n = 1000, mean = 60, sd = 15))
par(yaxs="i",las=1)
hist(dfHist2$Scores, freq = FALSE,
     ylim = c(0, 0.035),
     breaks = seq(10, 120, by = 5),
     col = "black", border = "white",
     main = "Test scores of 1000 Students (hist & density)", xlab = "Scores")
lines(density(dfHist1$Scores), col = "red", lwd = 4)
```

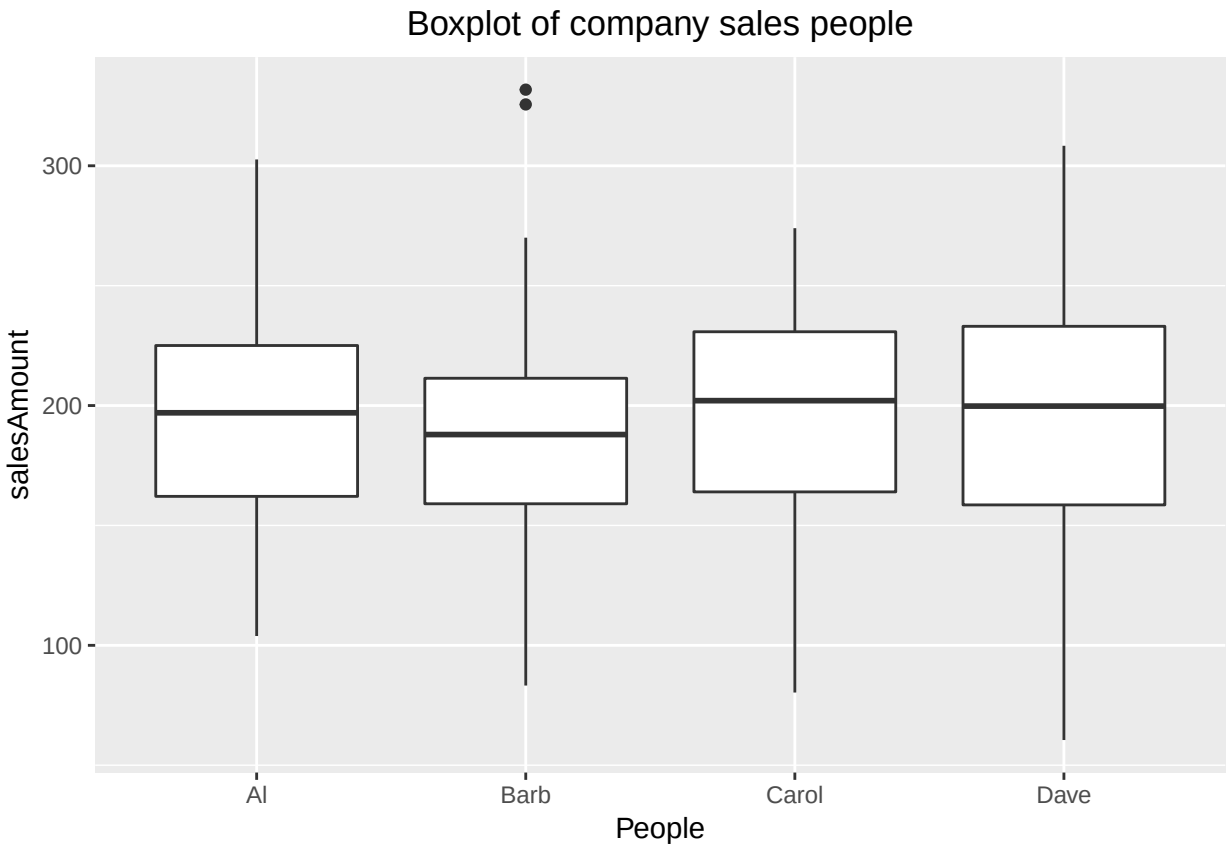


Boxplot 1
Boxplot 2

```
set.seed(2018)
Counts = 300
names4 = c("Al", "Barb", "Carol", "Dave")
states3 = c("CA", "WA", "OR")
salesData = data.frame(People = sample(names4, Counts, replace = TRUE),
                       salesState = sample(states3, Counts, replace = TRUE),
                       salesAmount = rnorm(Counts, mean = 200, sd = 50))

boxplot(salesAmount ~ People, data = salesData, main = "Boxplot of company sales people")
```

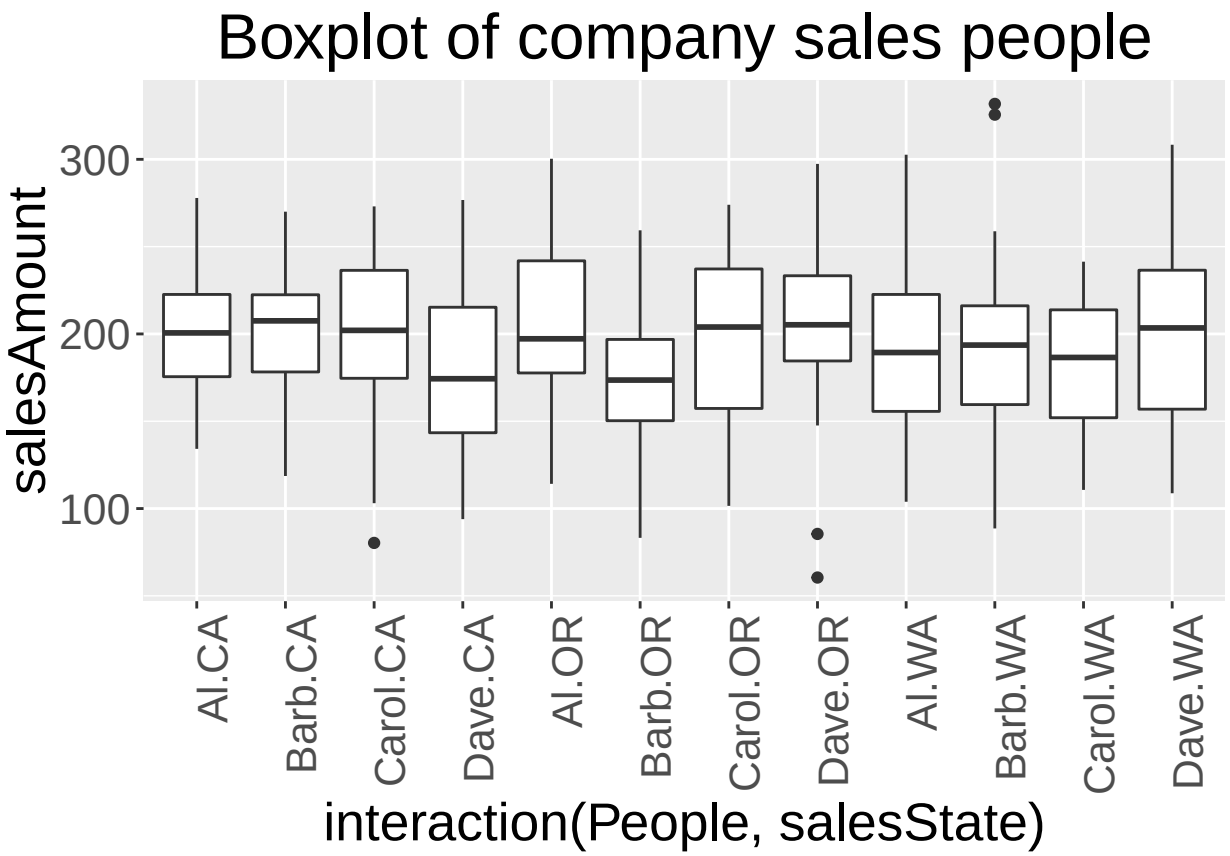



```
ggplot(salesData, aes(x = People, y = salesAmount)) + geom_boxplot() +
  ggtitle("Boxplot of company sales people") + theme(plot.title = element_text(hjust=0.5))
```



Boxplot 3

```
ggplot(salesData, aes(x = interaction(People, salesState), y = salesAmount)) + geom_boxplot() +
  ggtitle("Boxplot of company sales people") + theme(plot.title = element_text(hjust=0.5)) +
  theme(text = element_text(size=20),
        axis.text.x = element_text(angle=90, hjust=1))
```

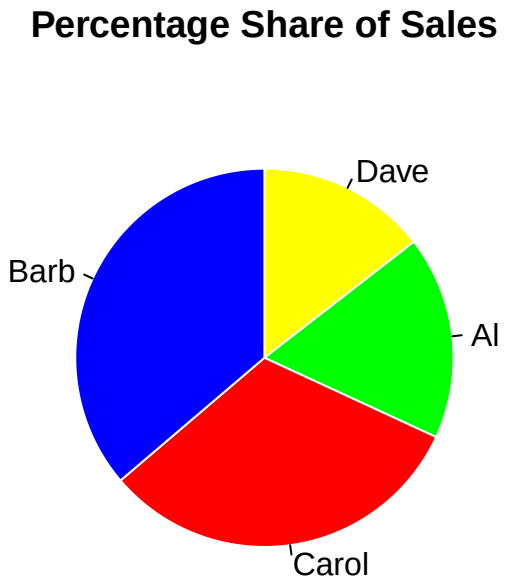


Pie chart 1

```
dfPie1 = data.frame(Salespersons = c("Al", "Barb", "Carol", "Dave"), SalesTotals = c(12, 25, 22, 10))
dfPie1 = dfPie1[order(dfPie1[,2]), ]
dfPie1
```

```
##   Salespersons SalesTotals
## 4         Dave          10
## 1           Al          12
## 3        Carol          22
## 2         Barb          25
```

```
pie(dfPie1[, 2],
    labels = dfPie1[, 1],
    clockwise = TRUE,
    col = c('yellow', 'green', 'red', 'blue'),
    border = 'white',
    main = 'Percentage Share of Sales')
```

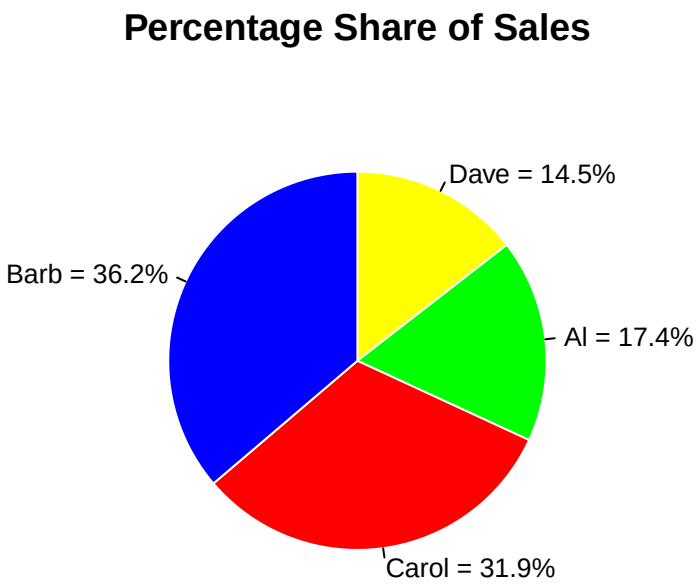


Pie chart 2

```
pielabels <- sprintf("%s = %3.1f%s", dfPie1[,1],
                     100*dfPie1[,2]/sum(dfPie1[,2]), "%")
pielabels
```

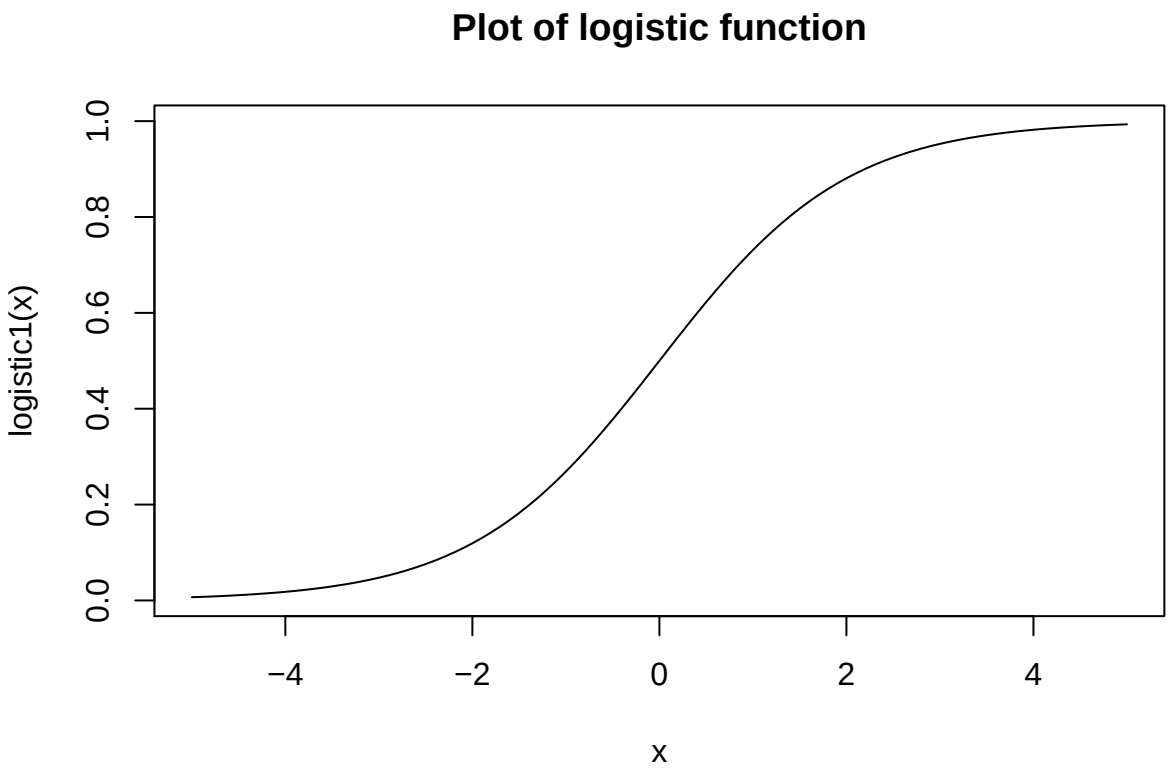
```
## [1] "Dave = 14.5%"  "Al = 17.4%"    "Carol = 31.9%" "Barb = 36.2%"
```

```
pie(dfPie1[, 2],
    labels = pielabels,
    clockwise = TRUE,
    col = c('yellow', 'green', 'red', 'blue'),
    border = 'white',
    cex = 0.8,
    main = 'Percentage Share of Sales')
```

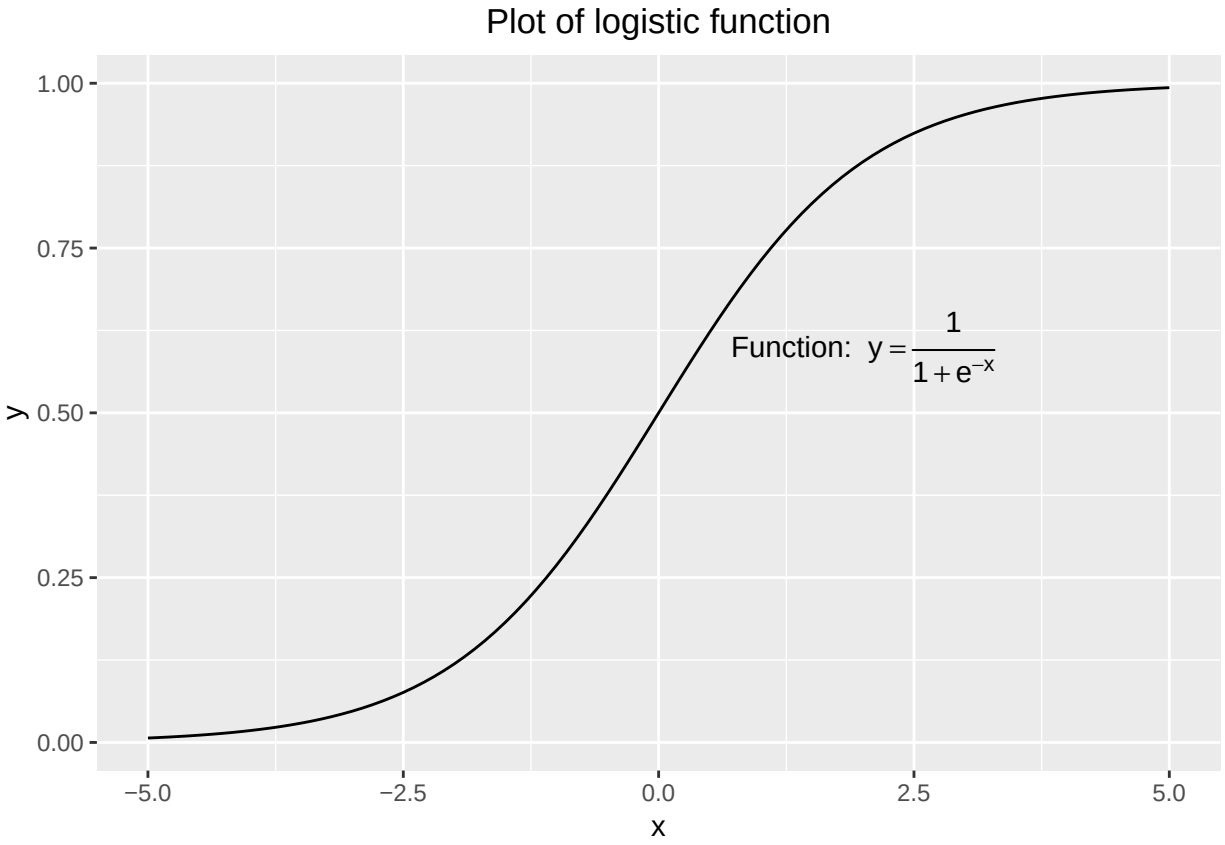


User defined function 1
User defined function 2

```
logistic1 = function(x) 1/(1 + exp(-x))

curve(logistic1, from = -5, to = 5, main = "Plot of logistic function")
```




```
p = ggplot(data.frame(x = c(-5, 5)), aes(x = x)) + stat_function(fun = logistic1, geom = "line") +
  ggtitle("Plot of logistic function")  + theme(plot.title = element_text(hjust=0.5))

p + annotate("text", x = 2, y = 0.6, parse = TRUE, size = 4, label = "'Function:  ' * y == frac(1, 1 + e
```



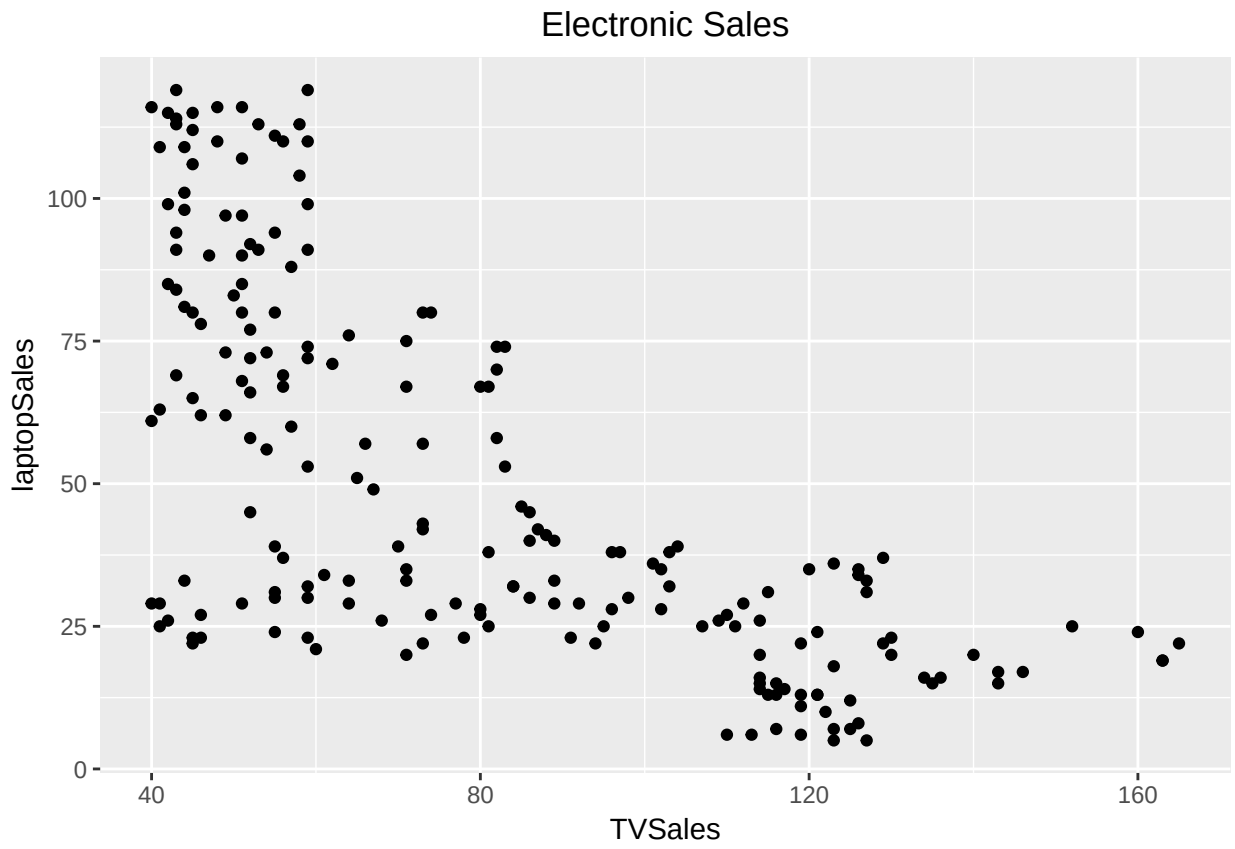
Point chart with facets 1
Point chart with facets 2
Point chart with facets 3
Point chart with facets 4

```
# From the basic plot, we see a strong negative correlation between TV sales and laptop sales
# When conditioing on weeknd, we see strong laptop sales when not weekend; we see strong
# TV sales when weekend
# We see that the amount of sales is based on state size and each state seems to
# have two distributions (we will see that is the weekend effect)
# With both factors used, we see that both are very important in driving the distribution
salesData2DF = data.frame(weekend = c(rep('Yes', 80), rep('No', 120)),
                          state =
                            c(rep('CA', 40), rep('WA', 20), rep('OR', 20),
                              rep('CA', 60), rep('WA', 30), rep('OR', 30)),
                          TVSales = round(c(runif(40, 70, 130), runif(20, 110, 170), runif(20, 110, 130)
                                            runif(60, 40, 60),runif(30, 50, 90), runif(30, 40, 80)),0),
                          laptopSales = round(c(runif(40, 20, 40), runif(20, 15, 25), runif(20, 5, 15),
                                                runif(60, 60, 120),runif(30, 40, 80), runif(30, 20, 40)),0)
                          )
head(salesData2DF)
```

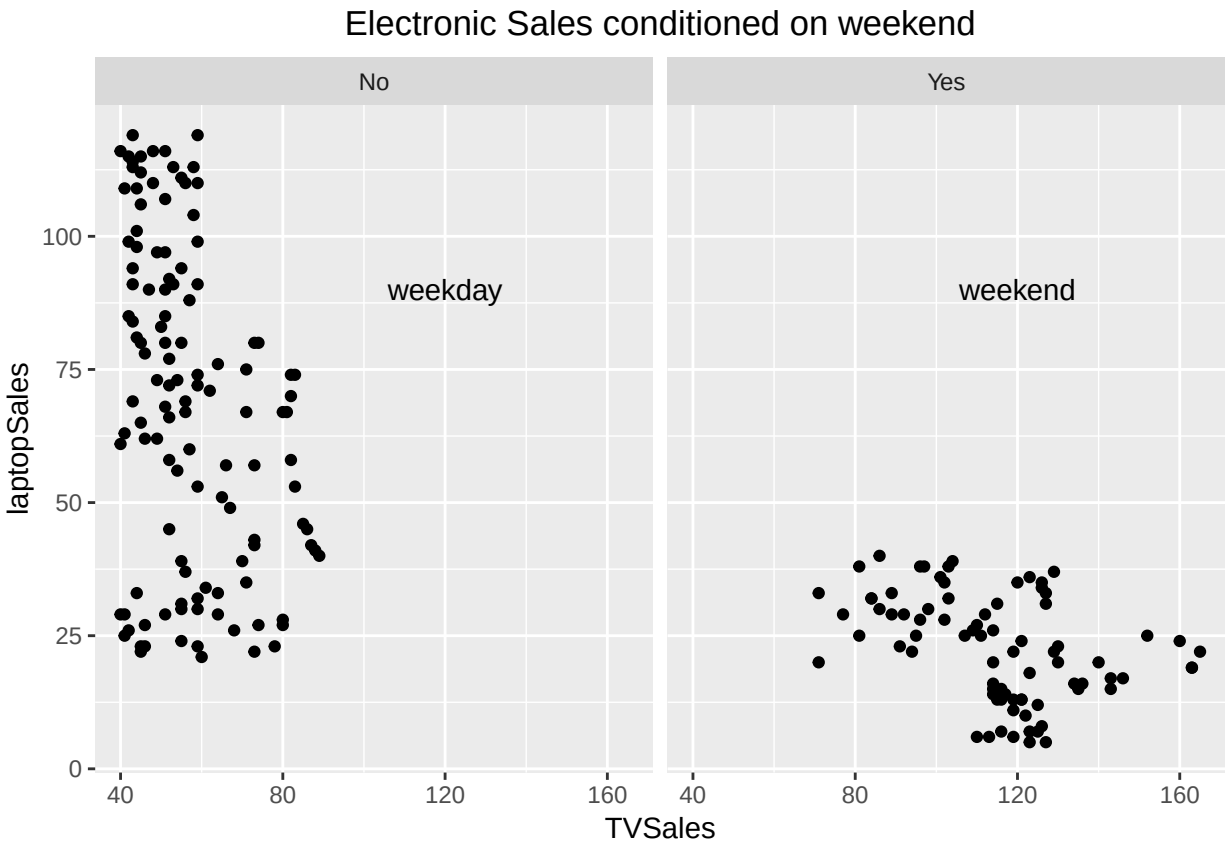
```
##   weekend state TVSales laptopSales
## 1     Yes    CA     129          37
## 2     Yes    CA      77          29
## 3     Yes    CA      96          38
## 4     Yes    CA      84          32
```

```
## 5     Yes    CA      96          28
## 6     Yes    CA      86          30
```

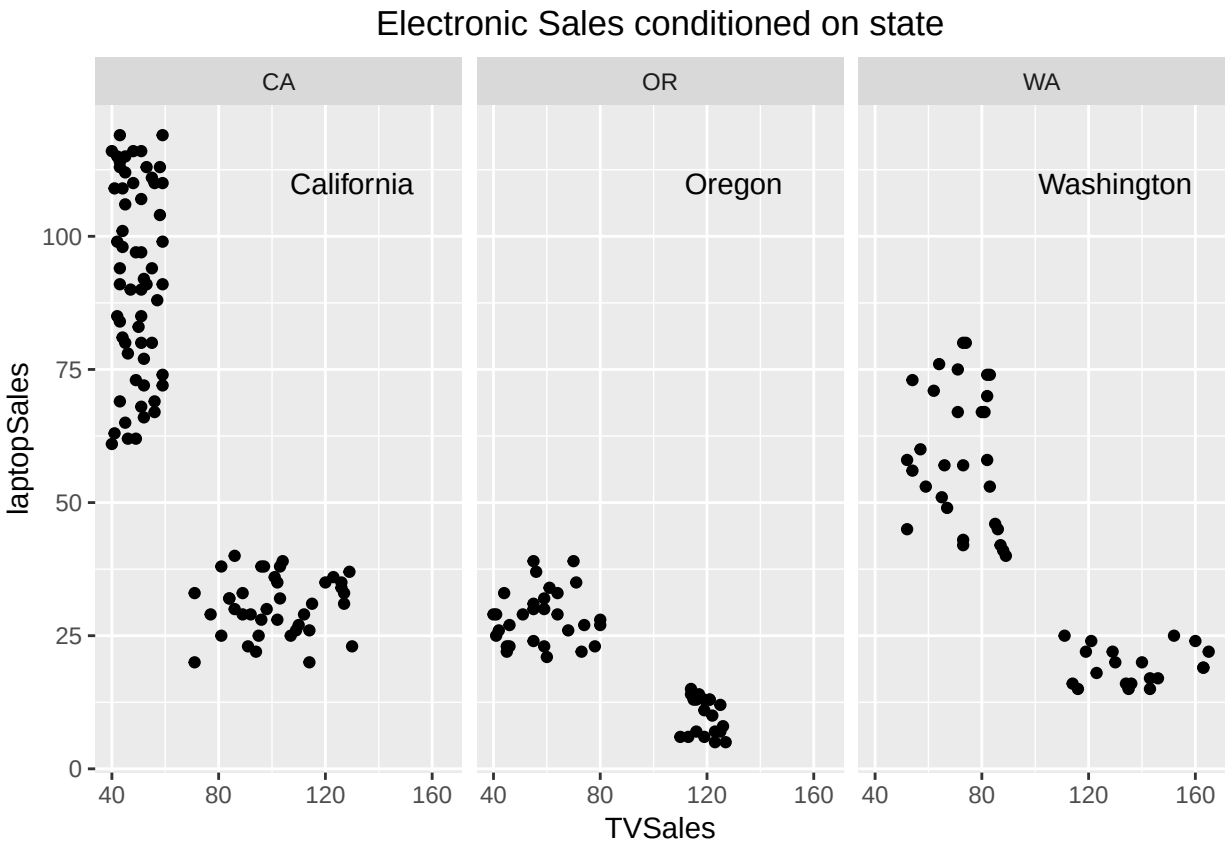
```r
# Basic plot
p1 <- ggplot(salesData2DF, aes(x = TVSales, y = laptopSales)) + geom_point() +
  ggtitle("Electronic Sales")  + theme(plot.title = element_text(hjust=0.5))
p1
```



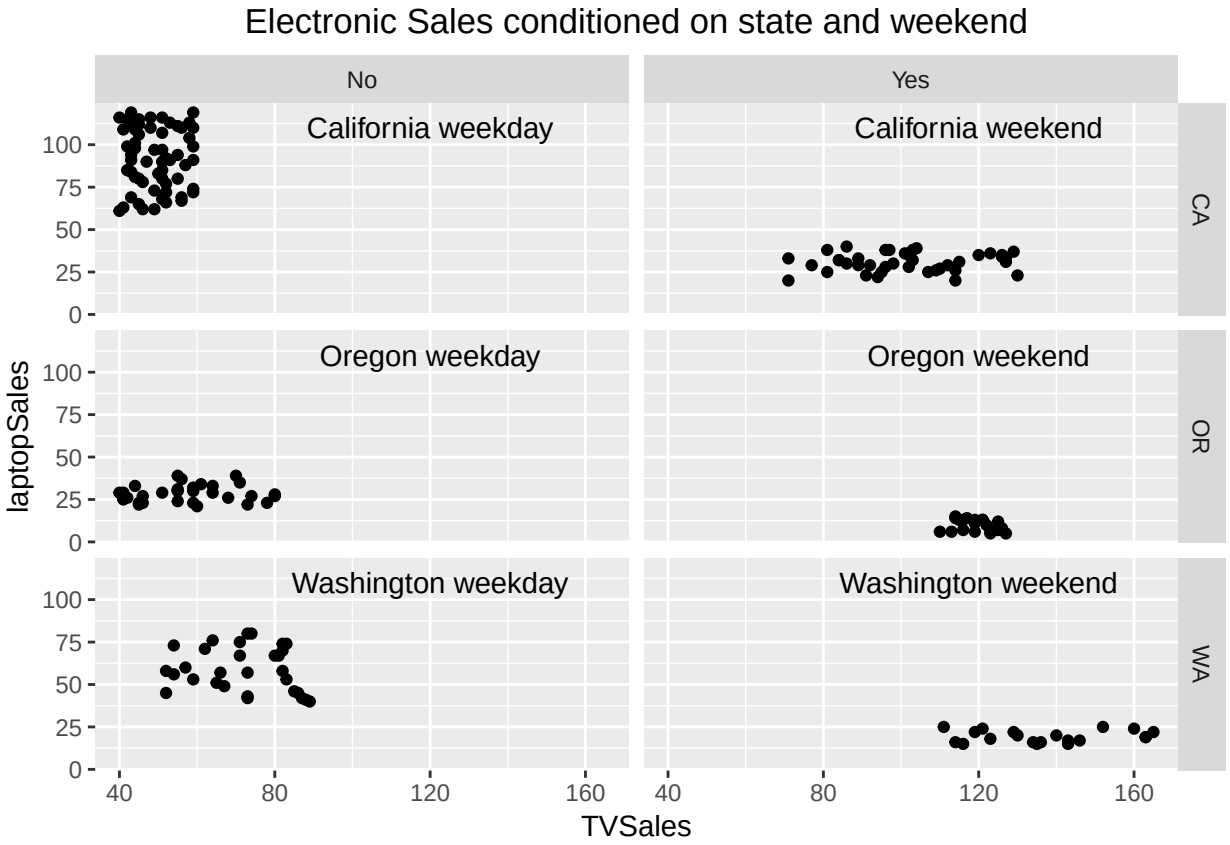

```r
# Basic plot + facet on weekend
p2W <- ggplot(salesData2DF, aes(x = TVSales, y = laptopSales)) + geom_point() +
  ggtitle("Electronic Sales conditioned on weekend")  + theme(plot.title = element_text(hjust=0.5))
f_labels <- data.frame(weekend = c("No", "Yes"), label = c("weekday", "weekend"))
p2W + facet_grid(. ~ weekend) + geom_text( x = 120, y = 90, aes(label = label), data = f_labels)
```




```
# Basic plot + facet on state
p2W <- ggplot(salesData2DF, aes(x = TVSales, y = laptopSales)) + geom_point() +
  ggtitle("Electronic Sales conditioned on state")  + theme(plot.title = element_text(hjust=0.5))
f_labels <- data.frame(state = c("CA", "WA", "OR"), label = c("California", "Washington", "Oregon"))
p2W + facet_grid(. ~ state) + geom_text( x = 130, y = 110, aes(label = label), data = f_labels)
```




```
# Basic plot + facet on both state and weekend
# https://stackoverflow.com/questions/11889625/annotating-text-on-individual-facet-in-ggplot2
p3 <- ggplot(salesData2DF, aes(x = TVSales, y = laptopSales)) + geom_point() +
  ggtitle("Electronic Sales conditioned on state and weekend")  + theme(plot.title = element_text(hjust=
f_labels3 <- data.frame(state = c("CA", "WA", "OR"),
                        weekend = c("No", "Yes"),
                        label = c("California weekday", "Washington weekend", "Oregon weekday",
                                  "California weekend", "Washington weekday", "Oregon weekend"))
p3 + facet_grid(state ~ weekend) + geom_text( x = 120, y = 110, aes(label = label), data = f_labels3)
```


Electronic Sales conditioned on state and weekend
No
Yes
California weekday
California weekend
Oregon weekday
Oregon weekend
Washington weekday
Washington weekend
CA
OR
WA
laptopSales
TVSales
0
25
50
75
100
40
80
120
160