Import_tidy

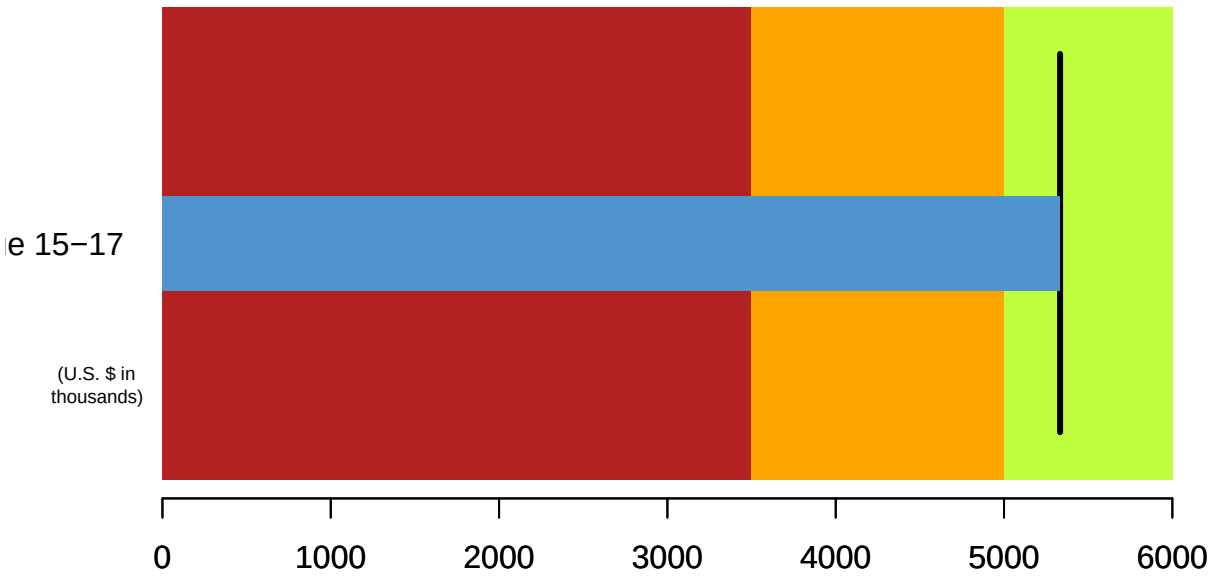
Moran Wang & Jingru Ma

January 31, 2018

Sevice Learning

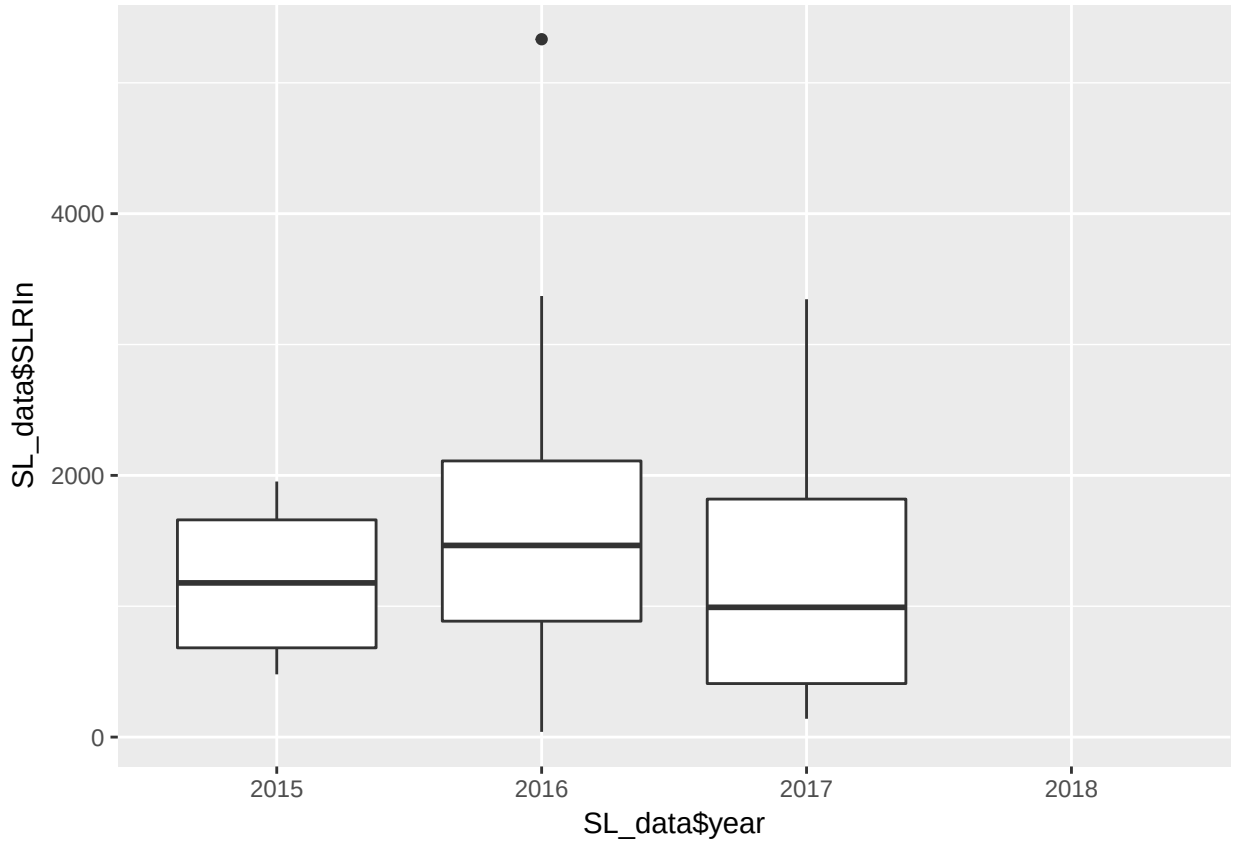
```
#read data
serviceLearning <- redata ("Service Learning.csv")

##subset the FY2017-18 to FY 2015, change header names
SL_data <- sub_range(serviceLearning,4,45,1,9)
colnames(SL_data) <-c("Month","Zone1","Zone2","Zone3","TNStudents","Tours","Classes","SLR","Reunve")
# delete the rows that are not useful
d_v <- c ('FY 16-17', '2016', 'FY 15-16','2015')
SL_data <- filter (SL_data, !SL_data$Month %in% d_v)
#str(SL_data)

#test function month_format , good
testnew <- SL_data$Month
mth <- month_format(testnew)

# test function year_format, good
yrnew <- year_format(SL_data$Month)

#try to add the mth , int 2,3,4 with yrnew, factor 2018,2017...
#hh <- as.yearmon(paste(yrnew, mth), "%Y%m", sep="-")
#dates1 <- as.Date(hh)
#SL_data$graphYearCol <- clean_date(SL_data$Month)

#change month from Feburary 18 to Feb 18
SL_data$Month <- as.yearmon(paste(year_format(SL_data$Month), month_format(SL_data$Month)), "%Y%m", sep

#add a new column in SLrevenue
SL_data$SLRIn <- make_int(SL_data$SLR)

#Zone23= zone2 + zone3
temp <- make_int(SL_data[1:14,3]) + make_int(SL_data[1:14,4])
temp2 <- make_int(SL_data[15:38,4])
SL_data$zone23 <- c(temp,temp2)

##draw graph of year and total service learning reveune
# the graph has already removed the missing value

#if the revenue is 0, i decided to calcualte an average value to replace it
m <- mean(SL_data$SLRIn,na.rm=TRUE)
SL_data$SLRIn[7] <- m

#bullet graph, Service Learning Revenue
#box plot, var
```

```
#split Feb 2018 to 2018
SL_data <- mutate(SL_data,year=sapply(str_split(SL_data$Month," "),function(x){x[2]}))
boxplot <- ggplot (SL_data, aes(x = SL_data$year, y = SL_data$SLRIn)) +
  geom_boxplot()

graphx <- max(SL_data$SLRIn, na.rm= TRUE)
bulletgraph(x=graphx,ref=5333,limits=c(0,3500,5000,6000),name= "Service Learning Revenue 15-17",subname=
```

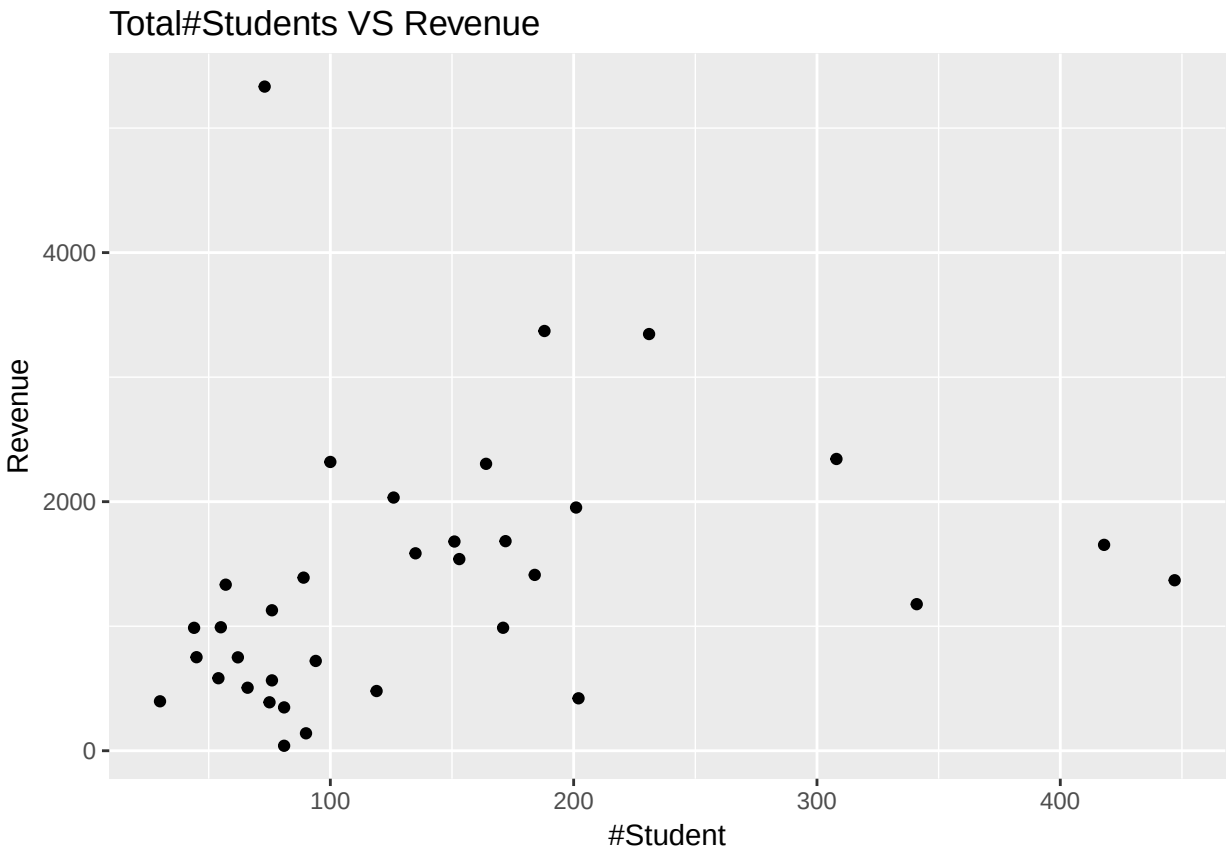



```
boxplot #looks like revenue have outlier
```

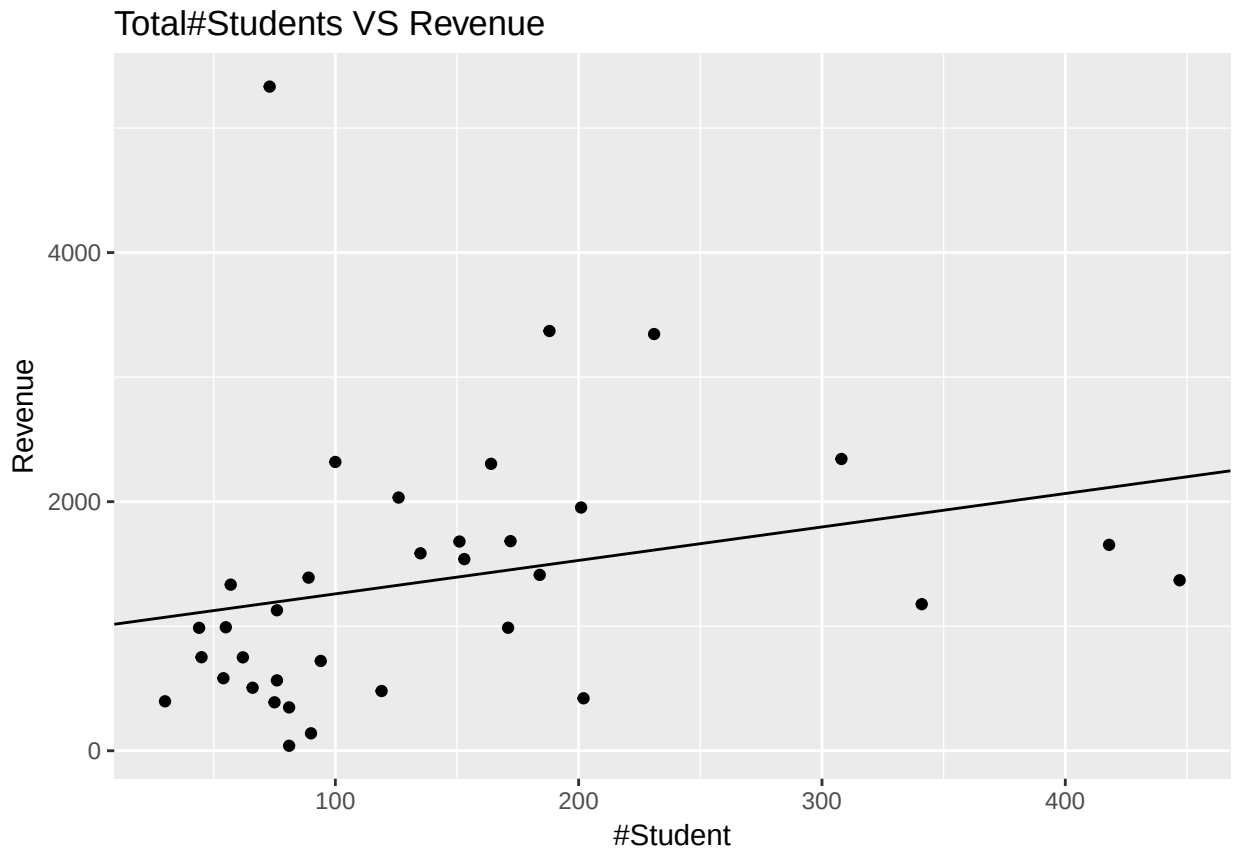
```
## Warning: Removed 3 rows containing non-finite values (stat_boxplot).
```




```
#analysis: fit linear regression model, the zone1+ zone2+zone3(TNstudents)
lr_data <- SL_data[,c(5,10)]
lr_data$TNStudents <- make_int(lr_data$TNStudents)
new.data <- lr_data[rowSums(is.na(lr_data[ ,c("TNStudents", "SLRIn")])) < 2, ]

# linear regression
p1 <- ggplot (data = new.data, aes(x= TNStudents, y = SLRIn)) + geom_point() + xlab("#Student") +
  ylab("Revenue") + ggtitle("Total#Students VS Revenue")
p1
```



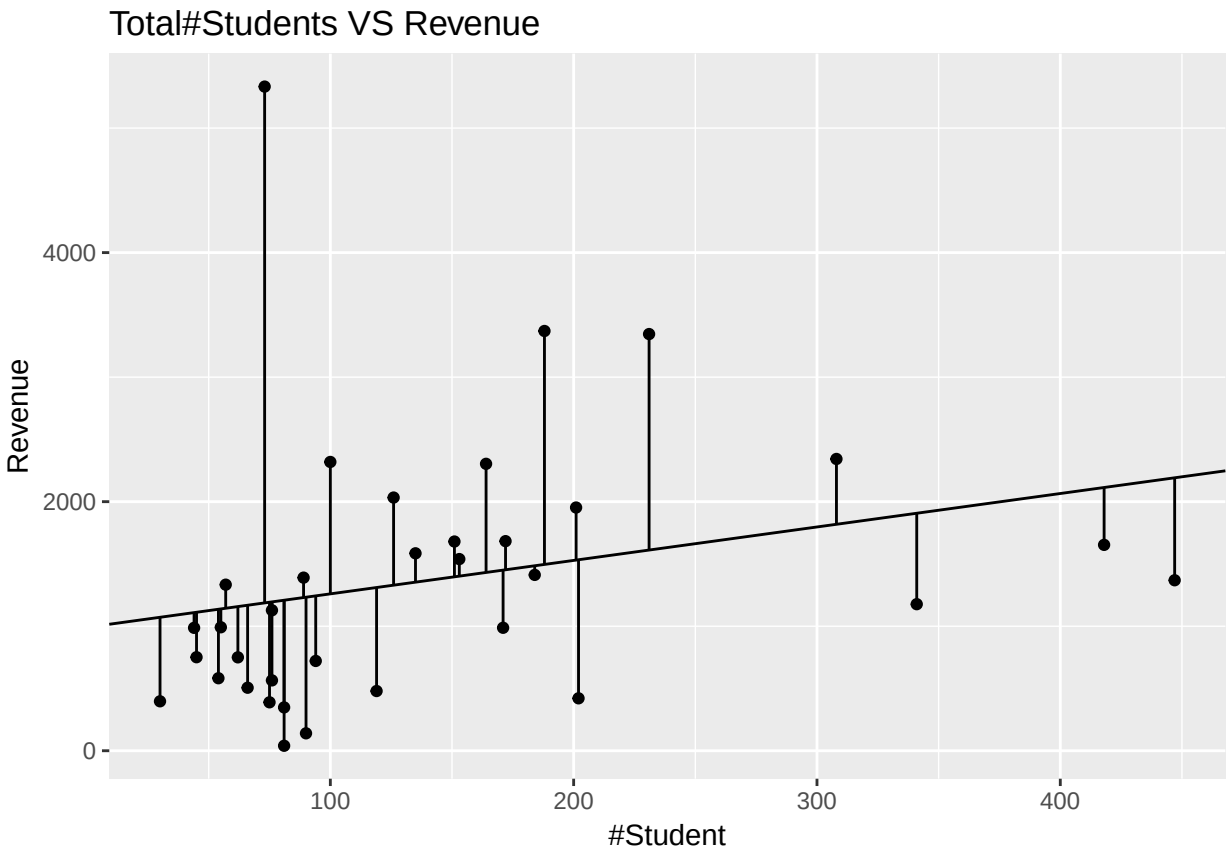

```
lm1 <- lm(new.data$SLRIn ~ new.data$TNStudents)
m <- 2.686
b <- 991.115
g <- p1 + geom_abline(slope = m, intercept = b)
g
```




```
# residual
lm1$residuals
```

```
##           1           2           3           4           5           6
##  -360.97277  -860.65890  -147.33003   189.34138 -1092.83043  -674.68689
##           7           8           9          10          11          12
##  1059.31231   872.42587 -1112.63171  1734.48224   -67.23027  -122.28705
##          13          14          15          16          17          18
##  4145.82691  -803.54454   136.96885   159.85530  -553.84430 -1168.65890
##          19          20          21          22          23          24
##   231.61191  -729.94758   703.48344   524.68136  1874.56845  -522.57333
##          25          26          27          28          29          30
##   422.05402  -822.63451   229.94006  -407.63011  -630.23027  -662.37301
##          31          32          33          34          35
##   283.34030  -460.44846  -831.21648   -73.28865  -462.87421
```

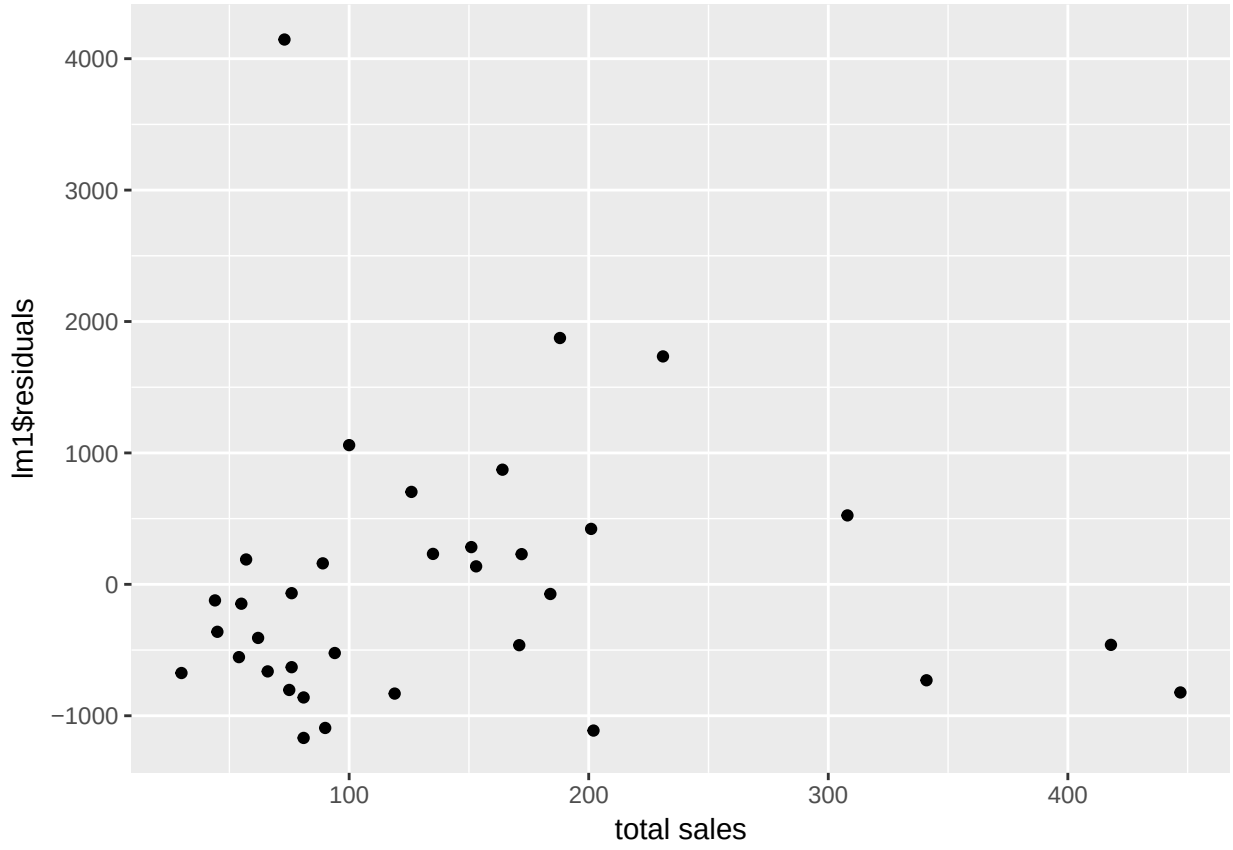
```
data.resid1 <- bind_cols(new.data,fitted = lm1$fitted.values)
g<-g + geom_linerange(data=data.resid1,aes(x=data.resid1$TNStudents,
                                           ymin=data.resid1$SLRIn,ymax=data.resid1$fitted))
g
```



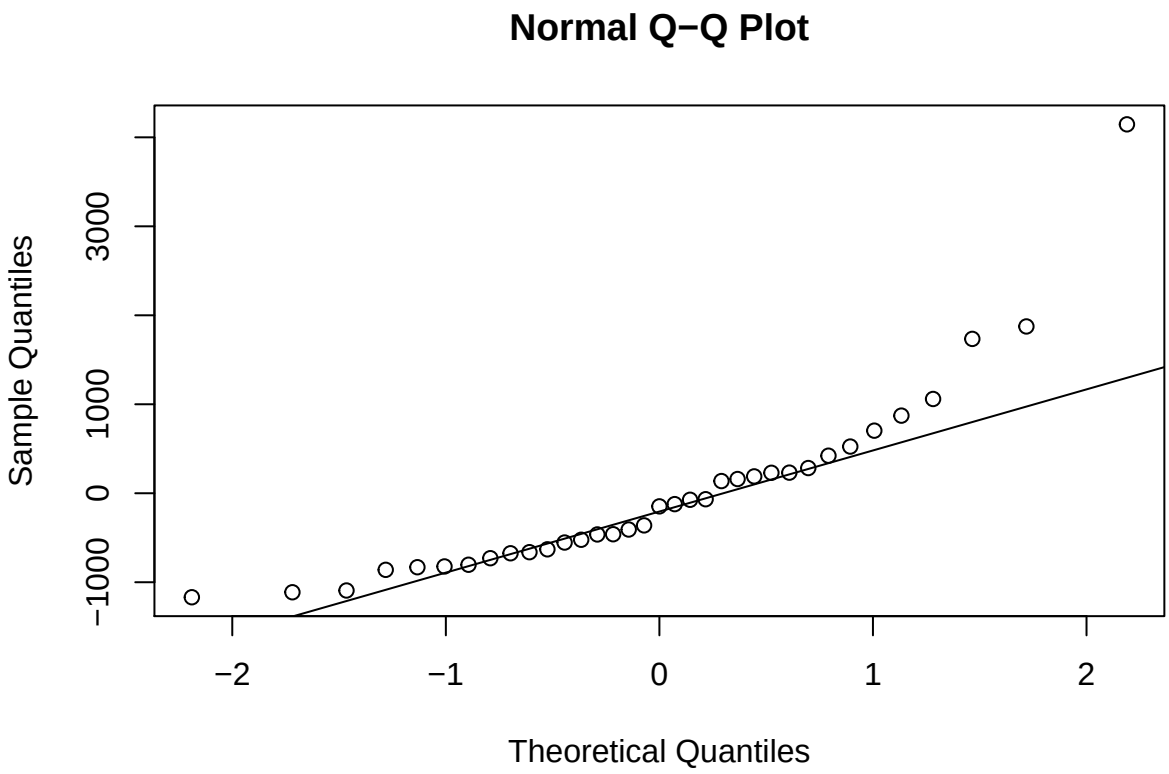

```
# R- square, residuals
summary(lm1)
```

```
##
## Call:
## lm(formula = new.data$SLRIn ~ new.data$TNStudents)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1168.7  -668.5  -147.3   257.5  4145.8
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          991.115    305.628   3.243  0.00271 **
## new.data$TNStudents    2.686      1.754   1.531  0.13530
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1052 on 33 degrees of freedom
## Multiple R-squared:  0.06632,    Adjusted R-squared:  0.03802
## F-statistic: 2.344 on 1 and 33 DF,  p-value: 0.1353
```

```
qplot(new.data$TNStudents, lm1$residuals) + xlab("total sales")
```




```
qqnorm(lm1$residuals)
qqline(lm1$residuals)
```




```
# I think linear line is not a good way to fit this relation
```

Food Box

```
FoodBox <- redata ("FoodBox.csv")
FB_data <- sub_range(FoodBox,2,164,1,8)
colnames(FB_data) <-c("Date","TotalBoxes","BasicBoxes","FamilyBoxes","Zone1","Zone2","Zone3","Revenue")

Month17 <- c('January','February','March','April','May','June', 'July', 'August', 'September','October'
FB_Month17 <- filter (FB_data, FB_data$Date %in% Month17)

d_v <- c ('FY 16-17', '2016', 'FY 15-16','2015', 'FY 17-18',  'YTD' , 'January','February','March','Apr
FB_data <- filter (FB_data, !FB_data$Date %in% d_v)
#str(FB_data)

#convert totalboxes, basic boxes, family boxes, zone1 ,zone2, zone3 to numeric number
FB_data[2:7] <- lapply(FB_data[2:7], function(x) as.numeric(as.character(x)))

#date is a factor, convert it to a date
FB_data$Date <- as.Date(FB_data$Date, format = "%m/%d/%Y")

FB_data <- na.omit(FB_data)

str(FB_data)
```

```
## 'data.frame':    142 obs. of  8 variables:
##  $ Date       : Date, format: "2017-11-27" "2017-11-20" ...
##  $ TotalBoxes : num  60 60 75 70 70 64 69 65 71 71 ...
##  $ BasicBoxes : num  8 61 61 55 55 51 56 55 53 55 ...
##  $ FamilyBoxes: num  6 14 14 15 15 13 13 10 18 16 ...
##  $ Zone1      : num  8 10 10 10 10 9 4 4 7 7 ...
##  $ Zone2      : num  6 6 6 6 6 6 11 10 11 13 ...
##  $ Zone3      : num  46 59 59 54 54 49 53 51 53 51 ...
##  $ Revenue    : Factor w/ 18 levels "","$4,881.63 ",..: 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:4] 1 2 3 4
##   .. ..- attr(*, "names")= chr [1:4] "1" "2" "3" "4"
```

```
#aggregate the total number of box in different month by year
# how to convert a date to a short date: FB_data$shortdate <- strftime(FB_data$Date, format="%Y/%m")
sb_FB_data <- subset(FB_data[2:7])
sum_year_month <- aggregate(sb_FB_data,by=list((substr(FB_data$Date,1,7))),sum)
str(sum_year_month) # the group.1 is a charactor , i need to change it to the date to plot
```

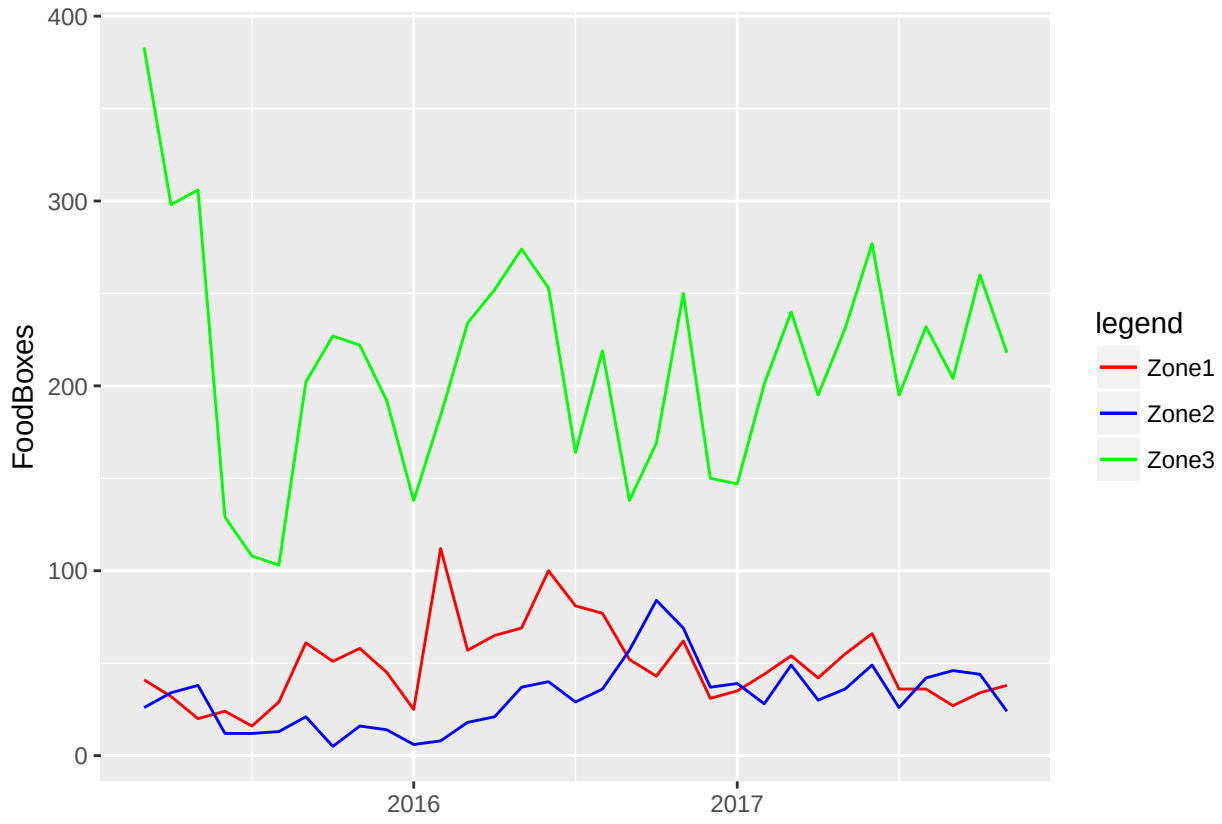
```
## 'data.frame':    33 obs. of  7 variables:
##  $ Group.1    : chr  "2015-03" "2015-04" "2015-05" "2015-06" ...
##  $ TotalBoxes : num  450 366 364 165 142 151 303 283 296 271 ...
##  $ BasicBoxes : num  308 255 260 111 86 92 216 206 223 216 ...
##  $ FamilyBoxes: num  142 111 104 54 66 53 80 77 73 55 ...
##  $ Zone1      : num  41 32 20 24 16 29 61 51 58 45 ...
##  $ Zone2      : num  26 34 38 12 12 13 21 5 16 14 ...
##  $ Zone3      : num  383 298 306 129 108 103 202 227 222 192 ...
```

```
#aggregate_month(FB_data,2:7,FB_data$Date) #this function return the argument must have sum length erro

#convert charactor as date
sum_year_month$Group.1 <- as.yearmon(sum_year_month$Group.1)
str(sum_year_month)
```

```
## 'data.frame':    33 obs. of  7 variables:
##  $ Group.1    :Class 'yearmon'  num [1:33] 2015 2015 2015 2015 2016 ...
##  $ TotalBoxes : num  450 366 364 165 142 151 303 283 296 271 ...
##  $ BasicBoxes : num  308 255 260 111 86 92 216 206 223 216 ...
##  $ FamilyBoxes: num  142 111 104 54 66 53 80 77 73 55 ...
##  $ Zone1      : num  41 32 20 24 16 29 61 51 58 45 ...
##  $ Zone2      : num  26 34 38 12 12 13 21 5 16 14 ...
##  $ Zone3      : num  383 298 306 129 108 103 202 227 222 192 ...
```

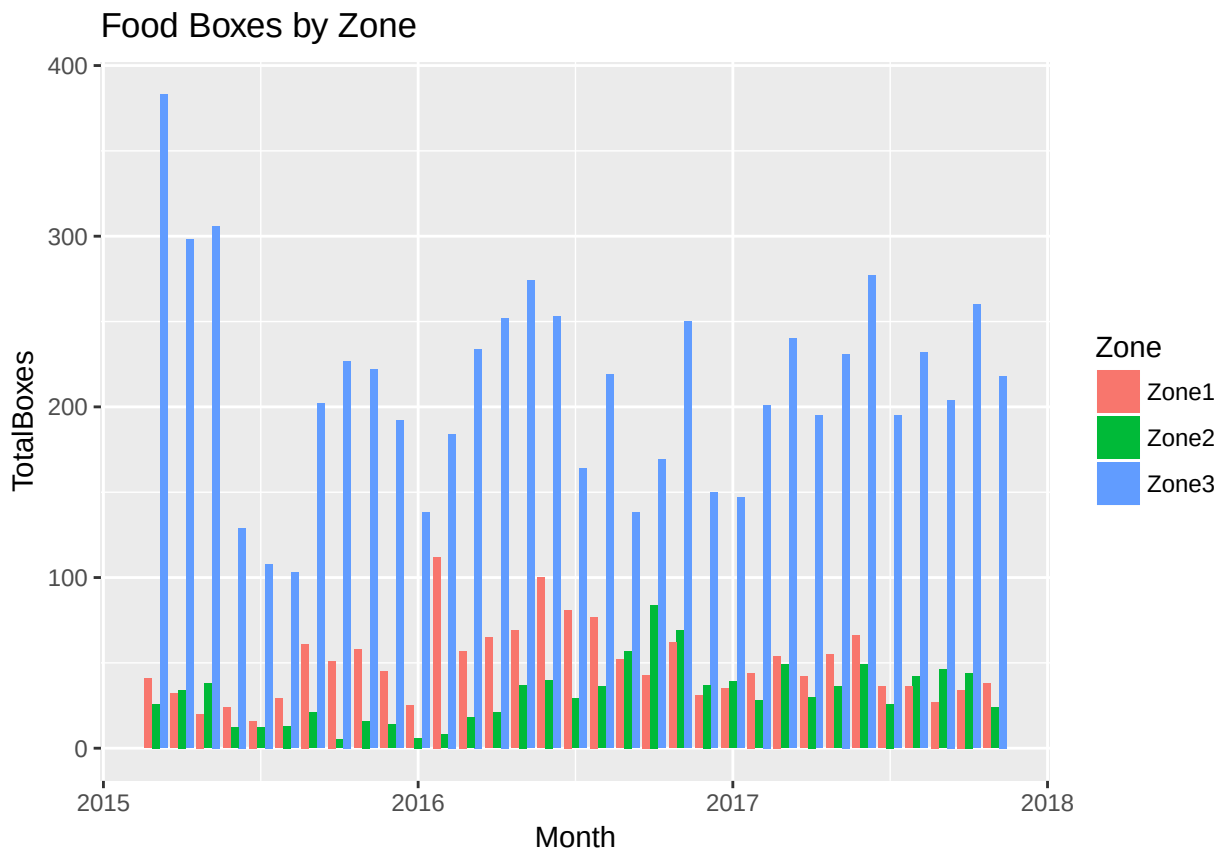
```
#basically, except the 2017-11, the other month's total boxes = bassic boxes + family boxes
variable <- c(sum_year_month$Zone1, sum_year_month$Zone2 , sum_year_month$Zone3)
g <- ggplot (sum_year_month, aes (sum_year_month$Group.1,sum_year_month$Zone1)) + geom_line(aes(color =
g <- g + geom_line(aes(y = sum_year_month$Zone2, color = "Zone2"))
g <- g + geom_line(aes(y = sum_year_month$Zone3, color = "Zone3"))
g <- g + scale_color_manual(values = c("red", "blue", "green")) +
  labs(color="legend")
g
```

```
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.
```

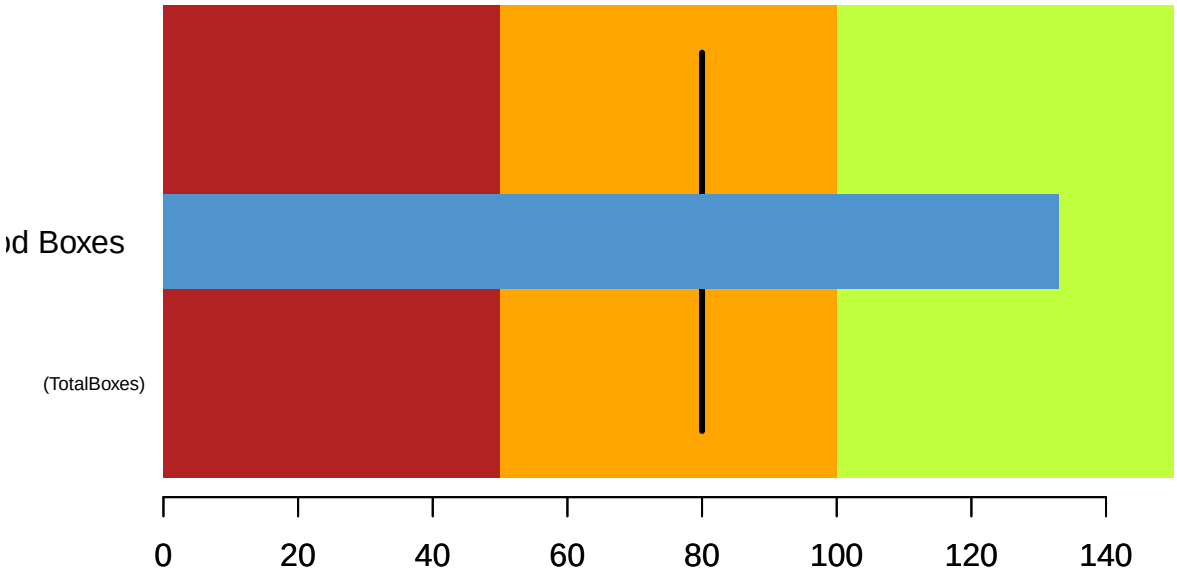



```
# for most of the sum zone1, zone2 and zone3, their value equal to total boxes. But for some of them, t

#need to deal with the table contained the revenue through 2015-2017

# try to create a bar graph same with excel, year, zone boxes, zones (like zone1, zone2, zone3)
new_graph <- sum_year_month %>% gather ('Zone1','Zone2','Zone3',key = "Date", value = 'TotalBoxes')
ngraph <- new_graph[c(1,5:6)]

ggplot(data=ngraph, aes(x=Group.1, y=TotalBoxes, fill=ngraph$Date)) +
geom_bar(stat="identity", position=position_dodge()) +
labs(x="Month", title = "Food Boxes by Zone", fill = "Zone")
```

```
## Don't know how to automatically pick scale for object of type yearmon. Defaulting to continuous.
```




```
#KPI
bulletgraph(x=max(FB_data$TotalBoxes),ref=80,limits=c(0,50,100,150),name= "Total Food Boxes",subname="("
```



Market

```
Market <- redata ("Market.csv")
Market_data <- sub_range(Market,11,166,1,9)
colnames(Market_data) <-c("Date","Zone1","Zone1Sale","Outside","Zone2","Zone2Sale","Zone3","Zone3Sale",

d_v <- c ('FY 16-17', '2016', 'FY 15-16','2015', 'FY 17-18',  'YTD' , 'January','February','March','Apri
Market_data <- filter (Market_data, !Market_data$Date %in% d_v)

#convert factor to number
Market_data[2:9] <- lapply(Market_data[2:9], function(x) as.numeric(as.character(x)))
```

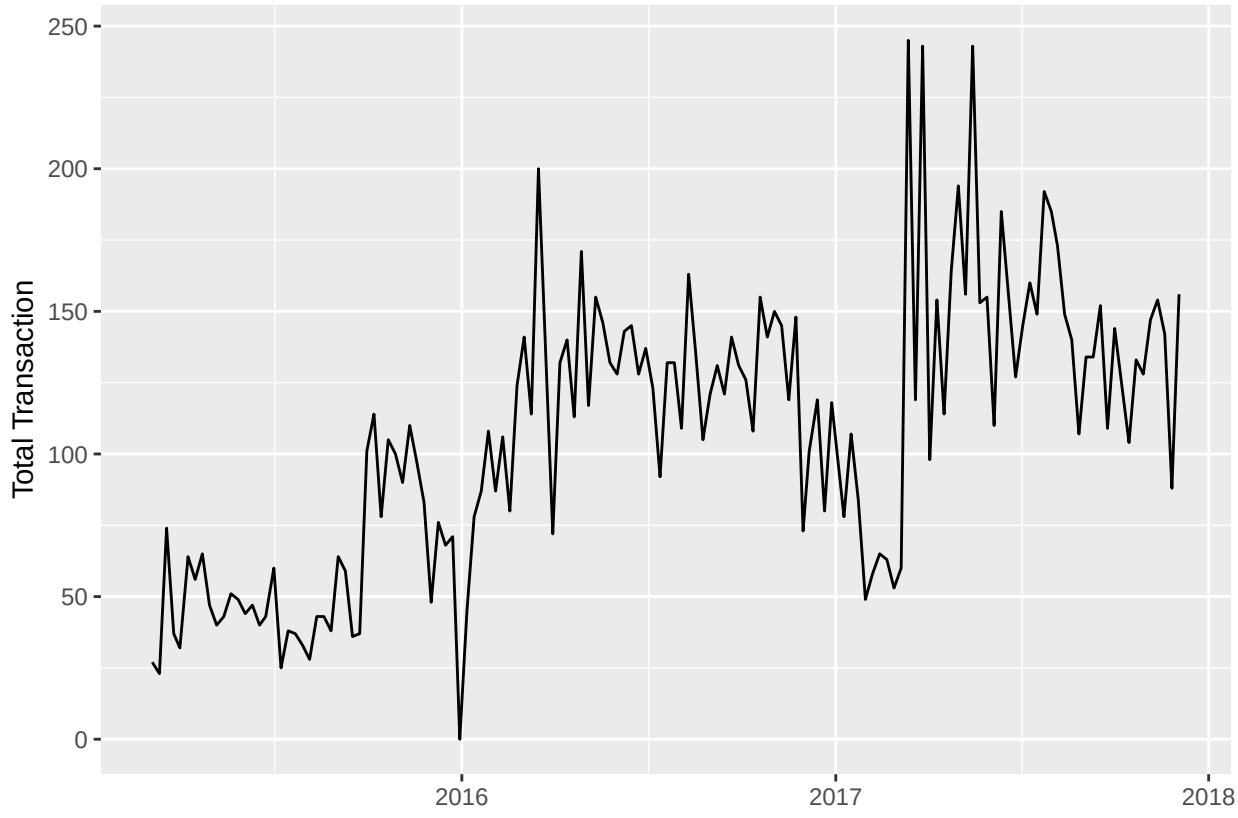
```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
#convert the factor date to date
Market_data$Date <- as.Date(Market_data$Date, format = "%m/%d/%Y")

str(Market_data)
```

```
## 'data.frame':    143 obs. of  9 variables:
##  $ Date      : Date, format: "2017-12-03" "2017-11-26" ...
##  $ Zone1     : num  87 63 63 91 62 62 64 52 52 58 ...
##  $ Zone1Sale : num  551 624 546 776 483 ...
##  $ Outside   : num  69 25 79 63 85 66 69 52 72 86 ...
##  $ Zone2     : num  48 15 58 37 57 42 35 41 44 49 ...
```

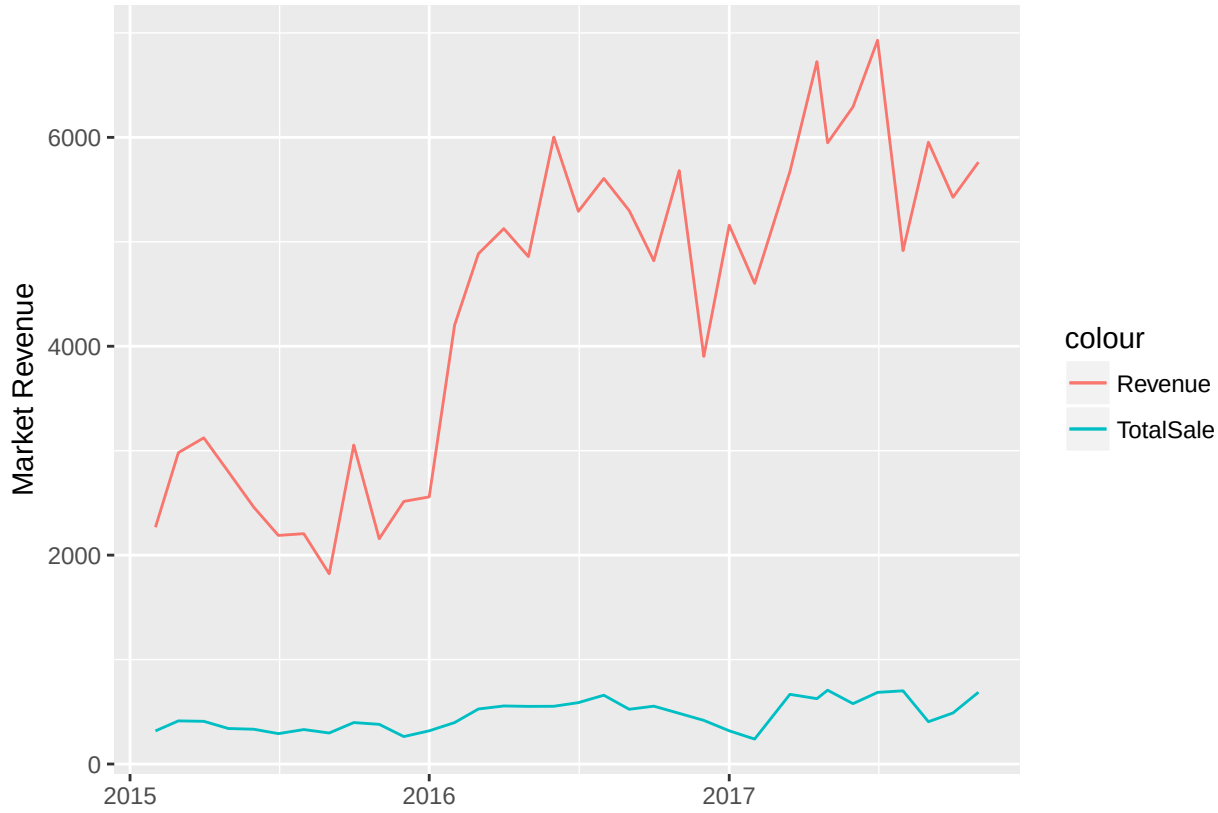
```
##  $ Zone2Sale : num  351 394 515 265 464 ...
##  $ Zone3     : num  21 10 21 26 28 24 34 11 28 37 ...
##  $ Zone3Sale : num  462 117 371 252 364 ...
##  $ TotalSales: num  156 88 142 154 147 128 133 104 124 144 ...
```

```
#add total by month by year
aggregate_month <- function (data, x, y){
  #x is the range of the data you want to add
  #y is the column of the year month
  temp_data <- subset (data[x])
  sum1 <- aggregate(temp_data,by = list(substr(data$y,1,7)),sum)
  return (sum1)
}

#remove all columns with na
nmd <- Market_data[ , colSums(is.na(Market_data)) == 0]


g <- ggplot (nmd, aes (nmd$Date,nmd$TotalSales)) + geom_line()  + xlab(" ") + ylab("Total Transaction")
g
```

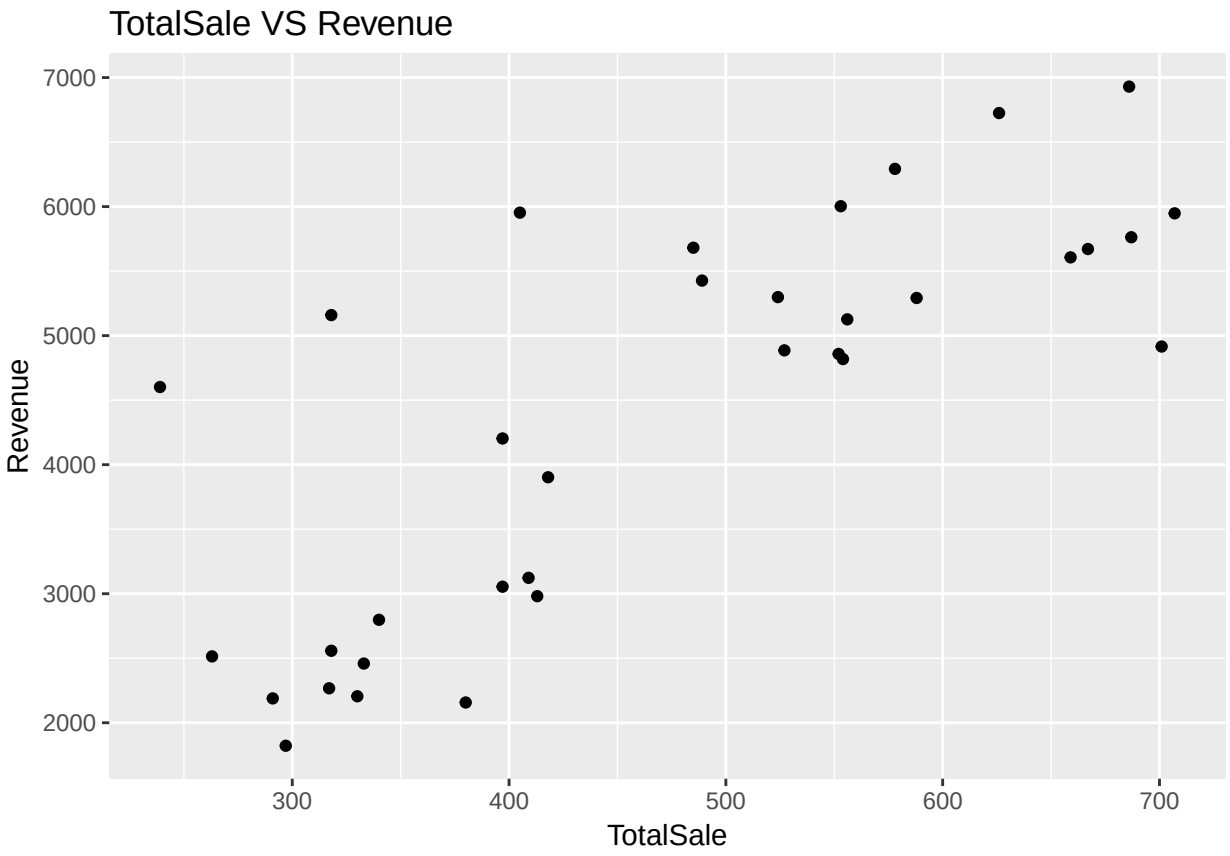



```
#need to read the table includes revenue and total boxes by month
Market_Revenue <- sub_range(Market,4,40,11,13)
#delete blank value
Market_Revenue <- Market_Revenue[-which(Market_Revenue$MONTHLY == ""),]
mr <- Market_Revenue [-20, ]
```

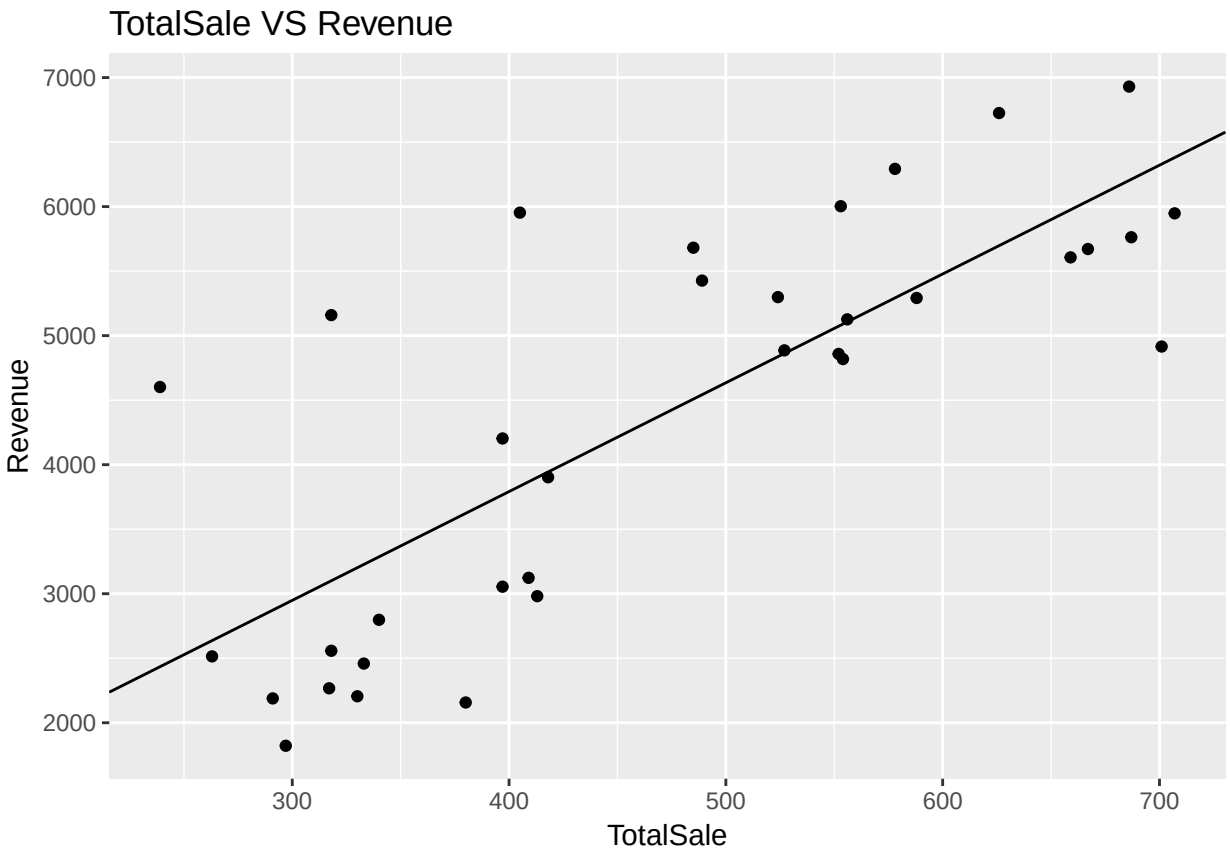
```r
mr$X.9 <- make_int(mr$X.9)
mr$X.10 <- make_int(mr$X.10)

mr$MONTHLY <- as.Date(mr$MONTHLY, format = "%m/%d/%Y")

g3 <- ggplot (mr, aes (mr$MONTHLY,mr$X.10)) + geom_line(aes(color = "Revenue"))  + xlab(" ") + ylab("Ma
g3 <- g3 + geom_line(aes(y = mr$X.9, color = "TotalSale"))
g3
```




```r
#linear regression , total sale and revenue
#density plot, check distribution
p1 <- ggplot (data = mr, aes(x= mr$X.9, y = mr$X.10)) + geom_point() + xlab("TotalSale") +
  ylab("Revenue") + ggtitle("TotalSale VS Revenue")
p1
```



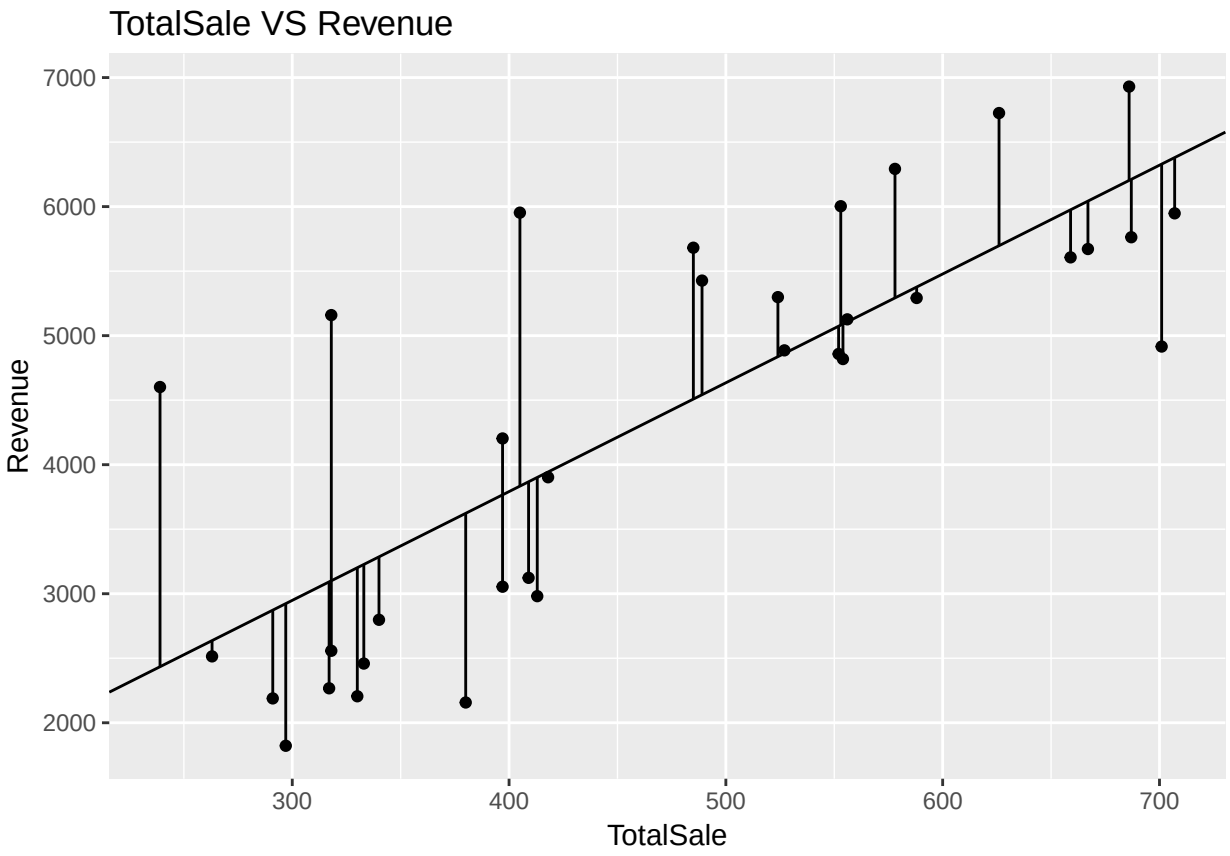

```
lm.sr <- lm(mr$X.10 ~ mr$X.9)
m <- 8.431
b <- 419.221
g <- p1 + geom_abline(slope = m, intercept = b)
g
```




```
# residual
lm.sr$residuals
```

```
##           1           2           3           4           5           6
##  -824.97557  -920.42023  -744.68462  -487.98783  -768.06801  -684.00910
##           7           8           9          10          11          12
##  -996.28380 -1101.57751  -712.11779 -1466.00394  -122.24982  -542.54697
##          13          14          15          16          17          18
##   436.66221    23.43986    19.20918  -215.11521   921.33339   -84.91571
##          19          20          21          22          23          24
##  -368.87530   461.24407  -271.58802  1173.02877   -41.03724  2058.76303
##          25          26          27          28          29          30
##  2167.56384  -371.56652  1027.73099  -432.32263   999.77832   726.89682
##          31          32          33          34
## -1414.08422  2119.25099   884.39316  -448.86458
```

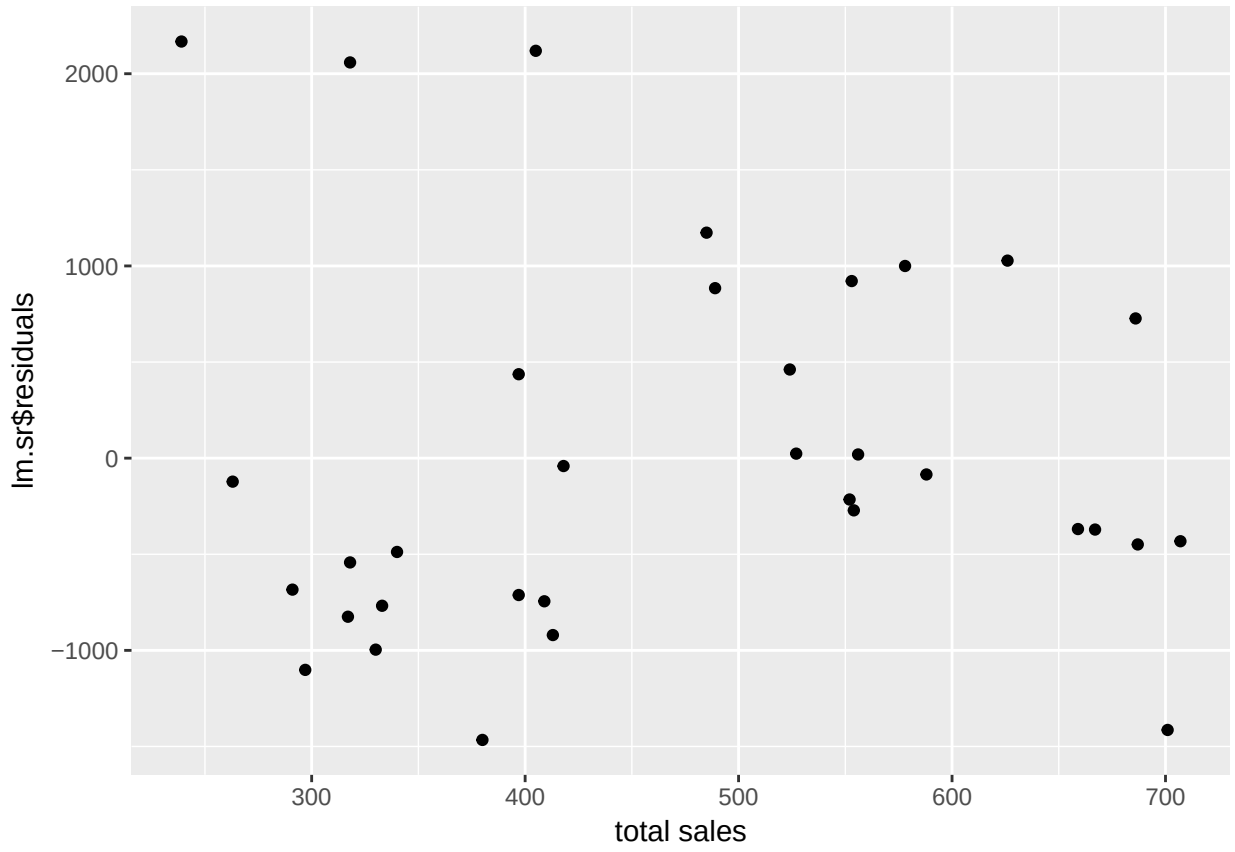
```
data.resid <- bind_cols(mr,fitted = lm.sr$fitted.values)
g<-g + geom_linerange(data=data.resid,aes(x=data.resid$X.9,
                                          ymin=data.resid$X.10,ymax=data.resid$fitted))
g
```



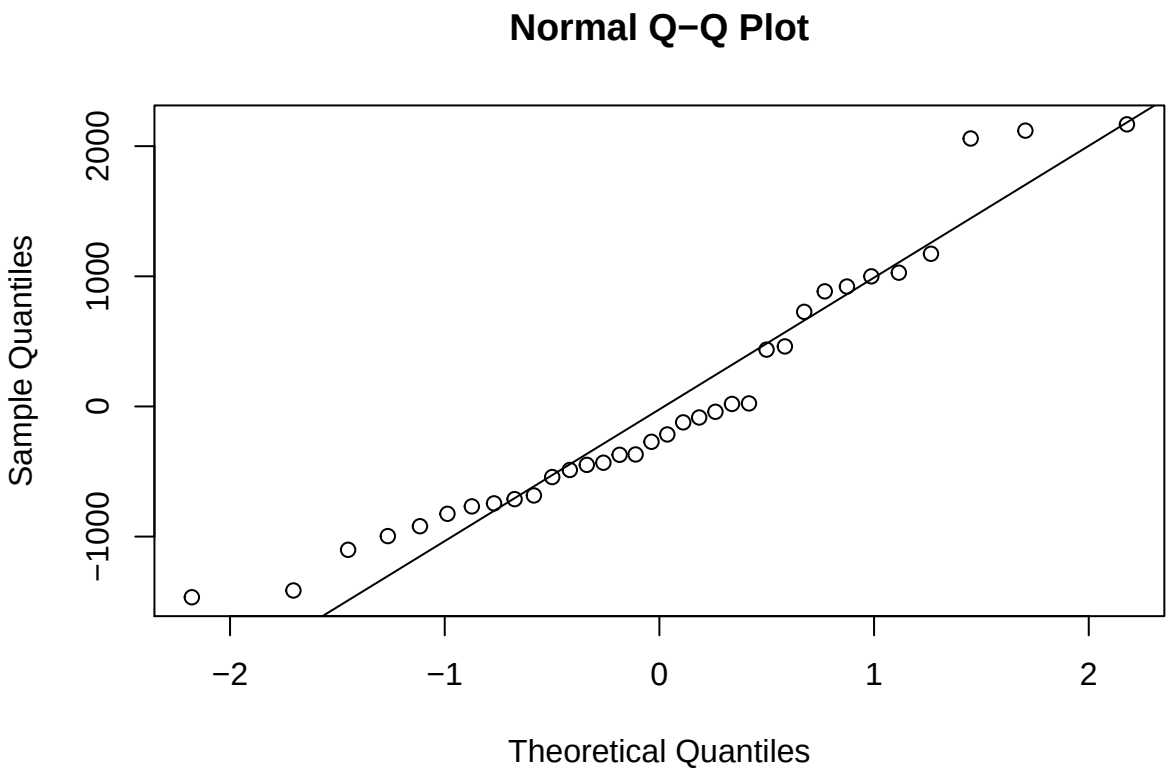

```
# R- square, residuals
summary(lm.sr)
```

```
##
## Call:
## lm(formula = mr$X.10 ~ mr$X.9)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1466.0  -705.1  -243.3   660.5  2167.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  419.221    585.093   0.717    0.479
## mr$X.9         8.431      1.191   7.082 4.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 980.5 on 32 degrees of freedom
## Multiple R-squared:  0.6105, Adjusted R-squared:  0.5983
## F-statistic: 50.15 on 1 and 32 DF,  p-value: 4.941e-08
```

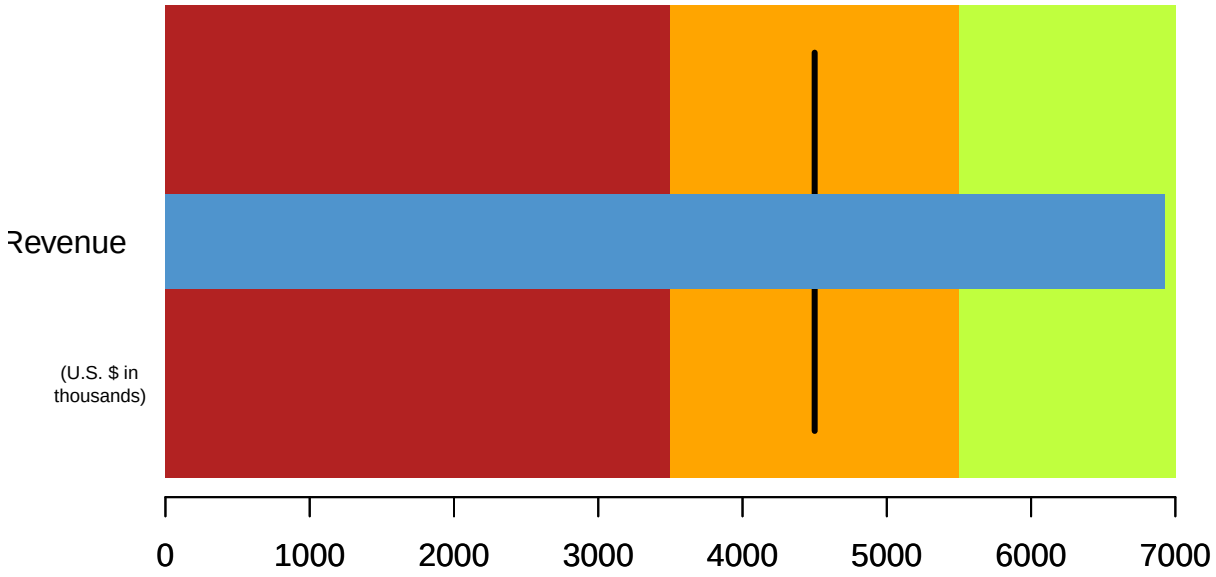
```
qplot(mr$X.9, lm.sr$residuals) + xlab("total sales")
```




```
qqnorm(lm.sr$residuals)
qqline(lm.sr$residuals)
```




```
# KPI
bulletgraph(x=max(mr$X.10, na.rm=TRUE),ref=4500,limits=c(0,3500,5500,7000),name= "Market Revenue",subna
```


Revenue
(U.S. $ in thousands)
0
1000
2000
3000
4000
5000
6000
7000