IE2152 Statistics for IE Project Final Project

Assoc. Prof. Özlem Şenvar / Res. Asst. M. Umut İzer

13/06/2021

**Group Number:25**

**Student 1 150818822 - Hatice Sena - TEMİZ**

**Student 2 150818821 - İremnur - BOYACI**

**Student 3 150319573 - Mert - HIRKA**

**Student 4 150816043 - Selcan - AKBULUT**

# 3. METHODOLOGY AND APPLICATION

In order to round to four decimal places and disable scientific notation, use the following command:

#To enter the data from drive document to R  
library(readxl)

## Warning: package 'readxl' was built under R version 4.0.3

url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vQETcbH5NyQ843pw6444ePjdQeVerp56-AGc0FVvv75WXGcEqM8zdeeQzD7-AajKQ/pub?output=xlsx"  
destfile <- "data\_22.xlsx"  
curl::curl\_download(url, destfile)  
data\_22 <- read\_excel(destfile)  
data\_22

## # A tibble: 42 x 7  
## Number BE CE PE OCE D GNP  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1977 668 553. 115. 144. 3810.  
## 2 2 1978 777. 627. 150. 125. 5251   
## 3 3 1979 1314. 992. 323. 199. 8504.  
## 4 4 1980 2422 1971. 451. 326. 17542.  
## 5 5 1981 3796. 3012. 784. 549. 24526.  
## 6 6 1982 4401 3272. 1129. 758. 34148   
## 7 7 1983 7575. 6032 1543. 1262. 47040.  
## 8 8 1984 12406. 10263. 2143. 2538. 73938.  
## 9 9 1985 20168. 17230. 2938. 3598. 126874.  
## 10 10 1986 32418. 27676 4742. 4982 195143.  
## # ... with 32 more rows

old<-options(digits=4,scipen=999)

## 3.1. Descriptive Statistics

Descriptive statistics allow summarizing the characteristics of a data set. While making this summary, values such as mean and median are used. Thus, data set features can be better understood and comments about the data set can be made through them.

#Write your codes here. Their results will appear after knitting the markdown file.

You can also add your explanations and comments here.

### 3.1.1. Summary Statistics

Summary statistics such as mean, median and mode etc. are computed below for every subset of data …

"SUMMARY"

## [1] "SUMMARY"

sapply(data\_22,summary,quantile.type=6)

## Number BE CE PE OCE D GNP  
## Min. 1.00 1977 668 553.3 114.7 124.9 3810  
## 1st Qu. 10.75 1987 42579 36200.0 6379.4 5730.5 265615  
## Median 21.50 1998 29873956 25915654.8 3958301.3 7662500.0 175672105  
## Mean 21.50 1998 467141360 394109262.6 73032097.7 68808754.1 2884909571  
## 3rd Qu. 32.25 2008 1044572891 883757752.2 167775049.2 150941856.2 5200070776  
## Max. 42.00 2018 2340681611 1965616053.4 375065558.1 305176267.8 18334799699

"MEAN"

## [1] "MEAN"

sapply(data\_22,mean)

## Number BE CE PE OCE D   
## 21.5 1997.5 467141360.3 394109262.6 73032097.7 68808754.1   
## GNP   
## 2884909571.2

"MEDIAN"

## [1] "MEDIAN"

sapply(data\_22,median)

## Number BE CE PE OCE D   
## 21.5 1997.5 29873956.1 25915654.8 3958301.3 7662500.0   
## GNP   
## 175672104.6

"MODE FOR CURRENT EXPENDITURES"

## [1] "MODE FOR CURRENT EXPENDITURES"

sort(table(data\_22$CE),dec=T)

##   
## 668 776.6 1314.3 2422 3795.6   
## 1 1 1 1 1   
## 4401 7575.1 12405.9 20167.5 32418.2   
## 1 1 1 1 1   
## 45966.5 75886.9 129114 247636.2 400163.9   
## 1 1 1 1 1   
## 654652.1 1162462.7 3037803.5 6263359.4 10671604   
## 1 1 1 1 1   
## 20375173.6 39372738.5 75952566 129126690.9 162283573.5   
## 1 1 1 1 1   
## 220646137.4 334498045.2 436157692.2 571203865.96 742526703.9   
## 1 1 1 1 1   
## 932560258.59 1024958715.7 1103415414.92 1140049984.45 1159028230.38   
## 1 1 1 1 1   
## 1199857540.42 1295770449.97 1465520571.62 1566222876.17 1680388903.91   
## 1 1 1 1 1   
## 1956564793.92 2340681611.43   
## 1 1

"MODE FOR PERSONNEL EXPENDITURES"

## [1] "MODE FOR PERSONNEL EXPENDITURES"

sort(table(data\_22$PE),dec=T)

##   
## 553.3 627.1 991.5 1971.1 3012.1   
## 1 1 1 1 1   
## 3271.8 6032 10262.7 17229.9 27676   
## 1 1 1 1 1   
## 39041.3 65663.5 108093.1 206942.8 341748.5   
## 1 1 1 1 1   
## 557737 973752.4 2669305.3 5509473.8 9275954.3   
## 1 1 1 1 1   
## 17756985.3 34074324.2 65793169.1 113053504.6 137522686.8   
## 1 1 1 1 1   
## 186722954.6 285819431.5 374790266.1 483139433.65 624001846.5   
## 1 1 1 1 1   
## 756015536.87 866264412.7 936237770.79 947401504.27 989460966.14   
## 1 1 1 1 1   
## 1016009248.82 1097304837.3 1240105107.68 1324714692.8 1425718505.2   
## 1 1 1 1 1   
## 1645246451 1965616053.38   
## 1 1

"MODE FOR OTHER CURRENT EXPENDITURES"

## [1] "MODE FOR OTHER CURRENT EXPENDITURES"

sort(table(data\_22$OCE),dec=T)

##   
## 114.7 149.5 322.8 450.9 783.5 1129.2   
## 1 1 1 1 1 1   
## 1543.1 2143.2 2937.6 4742.2 6925.2 10223.4   
## 1 1 1 1 1 1   
## 21020.9 40693.4 58415.4 96915.1 188710.3 368498.2   
## 1 1 1 1 1 1   
## 753885.6 1395649.7 2618188.3 5298414.3 10159396.9 16073186.2   
## 1 1 1 1 1 1   
## 24760886.7 33923182.8 48678613.7 61367426.1 88064432.31 118524857.4   
## 1 1 1 1 1 1   
## 158694303 167177644.13 169567264.24 176544721.72 183848291.6 192648480.18   
## 1 1 1 1 1 1   
## 198465612.67 225415463.94 241508183.37 254670398.71 311318342.92 375065558.05   
## 1 1 1 1 1 1

"MODE FOR DEFENCE"

## [1] "MODE FOR DEFENCE"

sort(table(data\_22$D),dec=T)

##   
## 124.9 143.6 198.9 325.8 548.6 757.9   
## 1 1 1 1 1 1   
## 1261.7 2537.9 3598.4 4982 5980 8564.7   
## 1 1 1 1 1 1   
## 12550 20140 46975 84150 126100 310265.8   
## 1 1 1 1 1 1   
## 747500 1247500 4325000 11000000 22421093 24275000   
## 1 1 1 1 1 1   
## 34870000 54000000 68918000 78000000 80145649.02 118843456   
## 1 1 1 1 1 1   
## 120047875.46 146813407.4 163327202.5 167395579.16 170498906.17 194687299.57   
## 1 1 1 1 1 1   
## 199979873.17 201795988.02 227517948.15 240860181.29 252444740.87 305176267.81   
## 1 1 1 1 1 1

"MODE FOR GNP"

## [1] "MODE FOR GNP"

sort(table(data\_22$GNP),dec=T)

##   
## 3810.5 5251 8504.5 17541.5   
## 1 1 1 1   
## 24525.6 34148 47040.2 73937.8   
## 1 1 1 1   
## 126874.2 195142.7 289106.4 485848.2   
## 1 1 1 1   
## 910058.6 1547793 2273698.1 4037702.2   
## 1 1 1 1   
## 6941224.3 16581566.8 35178971.7 63576940.3   
## 1 1 1 1   
## 117683403.8 233660805.3 407069775.4 651380055   
## 1 1 1 1   
## 1070424473 1418703263.6 1907070964 2520806747.4   
## 1 1 1 1   
## 3143699611.6 4101387190.5 4671255885.9 5128334134.4   
## 1 1 1 1   
## 5415280698.8 5649534936 6559174528.7 6915831629.3   
## 1 1 1 1   
## 7579403276.2 8840388007.6 10210731660.207 11605460378.4   
## 1 1 1 1   
## 14551761179.1426 18334799699.2   
## 1 1

"RANGE"

## [1] "RANGE"

sapply(data\_22,range)

## Number BE CE PE OCE D GNP  
## [1,] 1 1977 668 553.3 114.7 124.9 3810  
## [2,] 42 2018 2340681611 1965616053.4 375065558.1 305176267.8 18334799699

"VARIANCE"

## [1] "VARIANCE"

sapply(data\_22,var)

## Number BE CE   
## 150.5 150.5 432804216879521536.0   
## PE OCE D   
## 307348250641565056.0 10756865022613600.0 8362922193236486.0   
## GNP   
## 19844959216553959424.0

"STANDARD DEVIATION"

## [1] "STANDARD DEVIATION"

sapply(data\_22,sd)

## Number BE CE PE OCE   
## 12.27 12.27 657878573.05 554389980.65 103715307.56   
## D GNP   
## 91449014.17 4454768143.97

"INTERQUARTILE"

## [1] "INTERQUARTILE"

sapply(data\_22,IQR,type=6)

## Number BE CE PE OCE D   
## 21.5 21.5 1044530311.1 883721552.2 167768669.7 150936125.7   
## GNP   
## 5199805160.0

old<-options(digits=4,scipen=999)

…

### 3.1.2. Frequency Distribution

Frequency distribution is computed via using intervals which has equal widths according to range of data. Cumulative and relative frequencies are calculated below for all samples. A Frequency Distribution shows us that the summary data grouping is mutually divided into special classes and the number that occurs in a class. It is a way of expressing disorganized data. In this way, all categories in data 22 are specified with their classes. …

#Frequency Distribution for all samples  
n2<-nrow(data\_22)  
 k<-ceiling(log(n2,base=2))  
   
 for(i in 1:5){  
 if(i==1) {x=data\_22$CE  
 lab="Current Expenditures"}  
 if(i==2) {x=data\_22$PE  
 lab="Personnel Expenditures"}  
 if(i==3) {x=data\_22$OCE  
 lab="Other Current Expenditures"}  
 if(i==4) {x=data\_22$D  
 lab="Defence"}  
 if(i==5) {x=data\_22$GNP  
 lab="GNP"}  
   
   
 w<-round(diff(range(x))/k,1)  
 breaks<- seq(w\*floor(min(x)/w),w\*ceiling(max(x)/w),w)  
 x.cut <- cut(x, breaks,right=F)  
 x.freq <- table(x.cut)  
 x.relfreq <- x.freq / length(x)  
 x.cumfreq <- cumsum(x.freq)  
 x.cumrelfreq <- x.cumfreq / length(x)  
 print(paste("Frequency table for",lab))  
 print(cbind(x.freq, x.relfreq, x.cumfreq,x.cumrelfreq))  
 cat("\n")  
   
 }

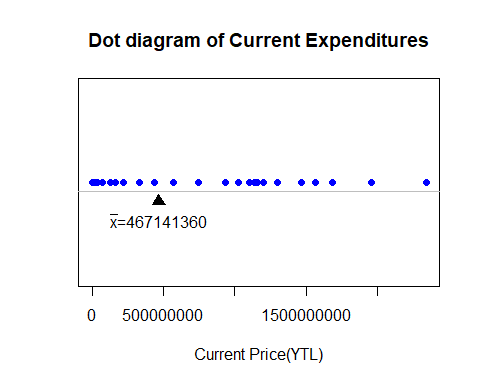
## [1] "Frequency table for Current Expenditures"  
## x.freq x.relfreq x.cumfreq x.cumrelfreq  
## [0,3.9e+08) 27 0.64286 27 0.6429  
## [3.9e+08,7.8e+08) 3 0.07143 30 0.7143  
## [7.8e+08,1.17e+09) 5 0.11905 35 0.8333  
## [1.17e+09,1.56e+09) 3 0.07143 38 0.9048  
## [1.56e+09,1.95e+09) 2 0.04762 40 0.9524  
## [1.95e+09,2.34e+09) 1 0.02381 41 0.9762  
## [2.34e+09,2.73e+09) 1 0.02381 42 1.0000  
##   
## [1] "Frequency table for Personnel Expenditures"  
## x.freq x.relfreq x.cumfreq x.cumrelfreq  
## [0,3.28e+08) 27 0.64286 27 0.6429  
## [3.28e+08,6.55e+08) 3 0.07143 30 0.7143  
## [6.55e+08,9.83e+08) 4 0.09524 34 0.8095  
## [9.83e+08,1.31e+09) 4 0.09524 38 0.9048  
## [1.31e+09,1.64e+09) 2 0.04762 40 0.9524  
## [1.64e+09,1.97e+09) 1 0.02381 41 0.9762  
## [1.97e+09,2.29e+09) 1 0.02381 42 1.0000  
##   
## [1] "Frequency table for Other Current Expenditures"  
## x.freq x.relfreq x.cumfreq x.cumrelfreq  
## [0,6.25e+07) 28 0.66667 28 0.6667  
## [6.25e+07,1.25e+08) 2 0.04762 30 0.7143  
## [1.25e+08,1.88e+08) 5 0.11905 35 0.8333  
## [1.88e+08,2.5e+08) 4 0.09524 39 0.9286  
## [2.5e+08,3.13e+08) 2 0.04762 41 0.9762  
## [3.13e+08,3.75e+08) 0 0.00000 41 0.9762  
## [3.75e+08,4.38e+08) 1 0.02381 42 1.0000  
##   
## [1] "Frequency table for Defence"  
## x.freq x.relfreq x.cumfreq x.cumrelfreq  
## [0,5.09e+07) 25 0.59524 25 0.5952  
## [5.09e+07,1.02e+08) 4 0.09524 29 0.6905  
## [1.02e+08,1.53e+08) 3 0.07143 32 0.7619  
## [1.53e+08,2.03e+08) 6 0.14286 38 0.9048  
## [2.03e+08,2.54e+08) 3 0.07143 41 0.9762  
## [2.54e+08,3.05e+08) 0 0.00000 41 0.9762  
## [3.05e+08,3.56e+08) 1 0.02381 42 1.0000  
##   
## [1] "Frequency table for GNP"  
## x.freq x.relfreq x.cumfreq x.cumrelfreq  
## [0,3.06e+09) 28 0.66667 28 0.6667  
## [3.06e+09,6.11e+09) 6 0.14286 34 0.8095  
## [6.11e+09,9.17e+09) 4 0.09524 38 0.9048  
## [9.17e+09,1.22e+10) 2 0.04762 40 0.9524  
## [1.22e+10,1.53e+10) 1 0.02381 41 0.9762  
## [1.53e+10,1.83e+10) 0 0.00000 41 0.9762  
## [1.83e+10,2.14e+10) 1 0.02381 42 1.0000

…

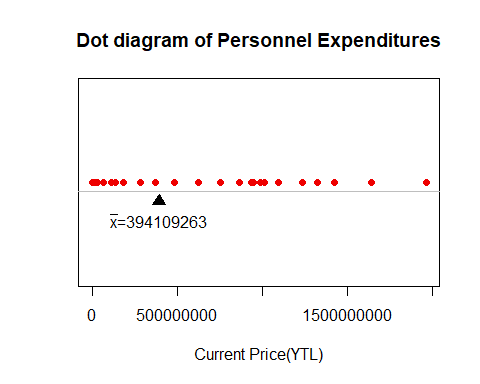
### 3.1.3. Descriptive Plots

Budget Expenditure data set such as Current Expenditure, Personnel Expenditure etc. is a continuous data type because of price is uncountable. Steam and leaf, dot plot, frequency graph, histogram, time-series plot were used for the data set to visualize the subdata one by one. …

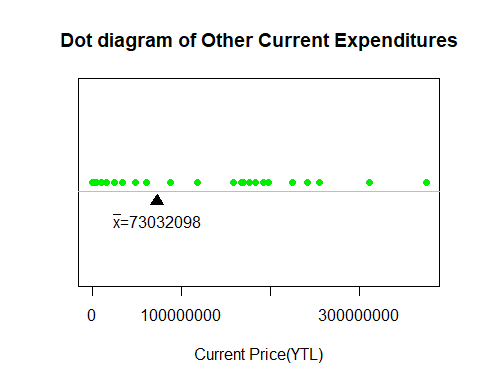
#Descriptive Plots  
 col=c('blue','red2','green2','purple', 'orange')  
  
 for(i in 1:5){  
 if(i==1) {x=data\_22$CE  
 lab="Current Expenditures"}  
 if(i==2) {x=data\_22$PE  
 lab="Personnel Expenditures"}  
 if(i==3) {x=data\_22$OCE  
 lab="Other Current Expenditures"}  
 if(i==4) {x=data\_22$D  
 lab="Defence"}  
 if(i==5) {x=data\_22$GNP  
 lab="GNP"}  
   
#Dot diagram  
stripchart(x, xlab="Current Price(YTL)",   
 main=paste("Dot diagram of",lab),  
 method="stack",   
 pch=16,   
 col=col[i])   
  
points(x=mean(x), y=0.8, pch=17, cex=1.5)   
   
abline(h=0.9,col='gray')  
  
text(x=mean(x), y=0.6, labels=bquote(bar(x)\*"=" \* .(mean(x))))  
  
#Stem-and-leaf diagram  
print(paste("Stem and leaf diagram of",lab))  
stem(x)  
 }



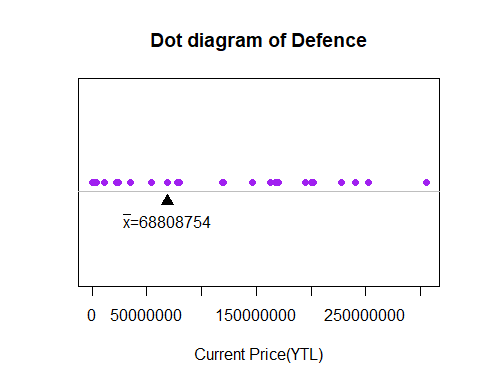
## [1] "Stem and leaf diagram of Current Expenditures"  
##   
## The decimal point is 8 digit(s) to the right of the |  
##   
## 0 | 0000000000000000001124836  
## 2 | 23  
## 4 | 47  
## 6 | 4  
## 8 | 3  
## 10 | 2046  
## 12 | 00  
## 14 | 77  
## 16 | 8  
## 18 | 6  
## 20 |   
## 22 | 4



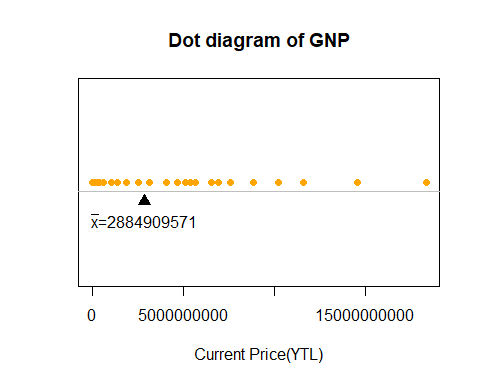
## [1] "Stem and leaf diagram of Personnel Expenditures"  
##   
## The decimal point is 8 digit(s) to the right of the |  
##   
## 0 | 00000000000000000011237149  
## 2 | 97  
## 4 | 8  
## 6 | 26  
## 8 | 7459  
## 10 | 20  
## 12 | 42  
## 14 | 3  
## 16 | 5  
## 18 | 7



## [1] "Stem and leaf diagram of Other Current Expenditures"  
##   
## The decimal point is 8 digit(s) to the right of the |  
##   
## 0 | 00000000000000000000011223  
## 0 | 569  
## 1 | 2  
## 1 | 677889  
## 2 | 034  
## 2 | 5  
## 3 | 1  
## 3 | 8

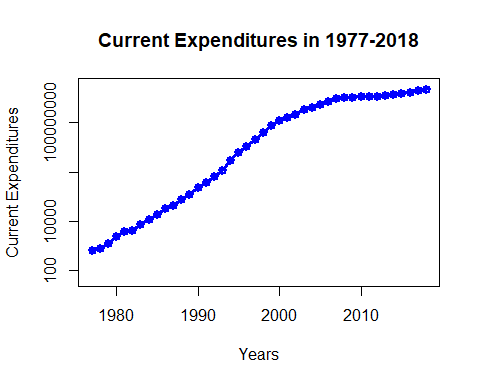


## [1] "Stem and leaf diagram of Defence"  
##   
## The decimal point is 8 digit(s) to the right of the |  
##   
## 0 | 0000000000000000000001223  
## 0 | 5788  
## 1 | 22  
## 1 | 56779  
## 2 | 0034  
## 2 | 5  
## 3 | 1

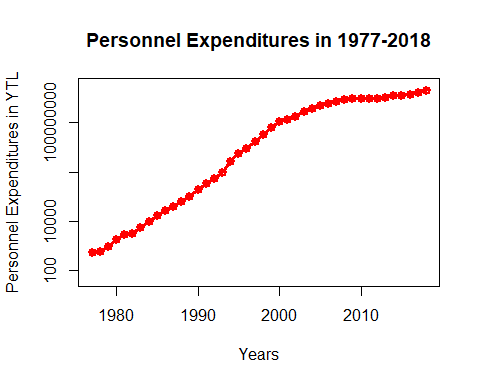


## [1] "Stem and leaf diagram of GNP"  
##   
## The decimal point is 9 digit(s) to the right of the |  
##   
## 0 | 000000000000000000011247149  
## 2 | 51  
## 4 | 17146  
## 6 | 696  
## 8 | 8  
## 10 | 26  
## 12 |   
## 14 | 6  
## 16 |   
## 18 | 3

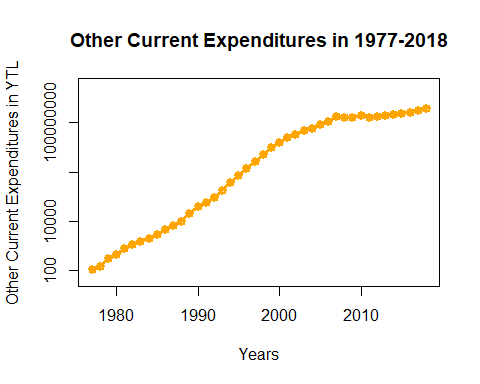
#Time-Series was plotted.  
plot(x=data\_22$BE,  
 y=data\_22$CE,  
 xlim=c(1977, 2018), ## with c()  
 ylim=c(50, 3000000000), ## with c()  
 log="y", col="blue",xlab = "Years", ylab = "Current Expenditures", main = "Current Expenditures in 1977-2018", type = "o", lwd="3")



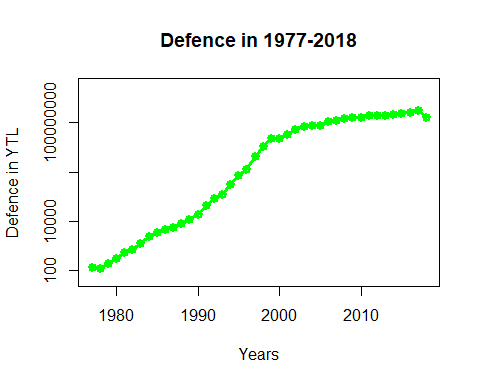
plot(x=data\_22$BE,  
 y=data\_22$PE,  
 xlim=c(1977, 2018), ## with c()  
 ylim=c(50, 3000000000), ## with c()  
 log="y", col="red",xlab = "Years", ylab = "Personnel Expenditures in YTL", main = "Personnel Expenditures in 1977-2018", type = "o", lwd="3")



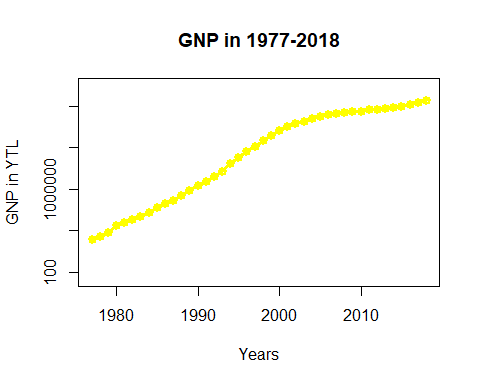
plot(x=data\_22$BE,  
 y=data\_22$OCE,  
 xlim=c(1977, 2018), ## with c()  
 ylim=c(50, 3000000000), ## with c()  
 log="y", col="orange",xlab = "Years", ylab = "Other Current Expenditures in YTL", main = "Other Current Expenditures in 1977-2018", type = "o", lwd="3")



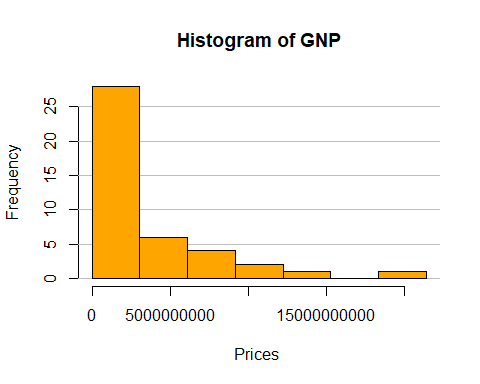
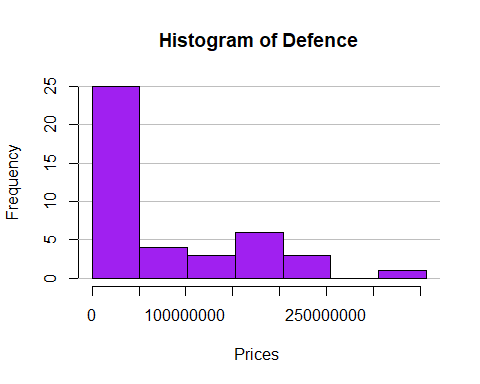
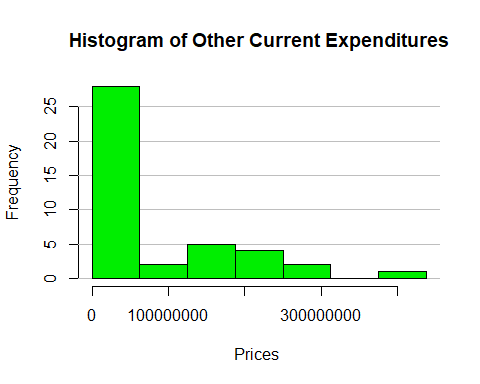
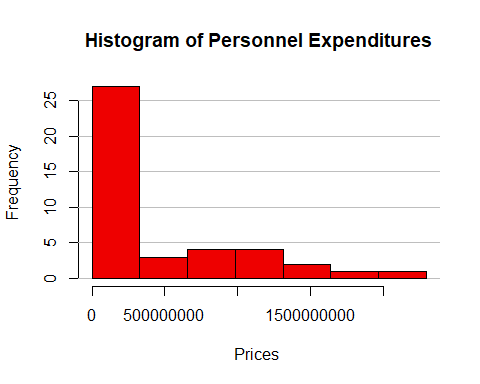
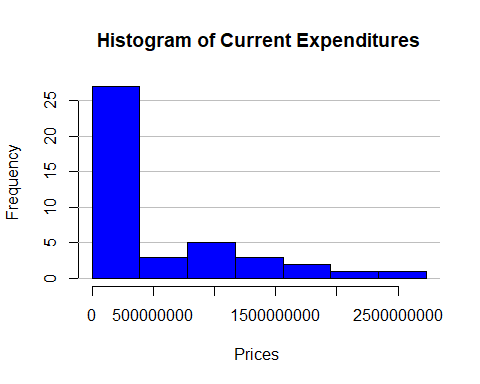
plot(x=data\_22$BE,  
 y=data\_22$D,  
 xlim=c(1977, 2018), ## with c()  
 ylim=c(50, 3000000000), ## with c()  
 log="y", col="green",xlab = "Years", ylab = "Defence in YTL", main = "Defence in 1977-2018", type = "o", lwd="3")



plot(x=data\_22$BE,  
 y=data\_22$GNP,  
 xlim=c(1977, 2018), ## with c()  
 ylim=c(50, 90000000000), ## with c()  
 log="y", col="yellow",xlab = "Years", ylab = "GNP in YTL", main = "GNP in 1977-2018", type = "o", lwd="3")



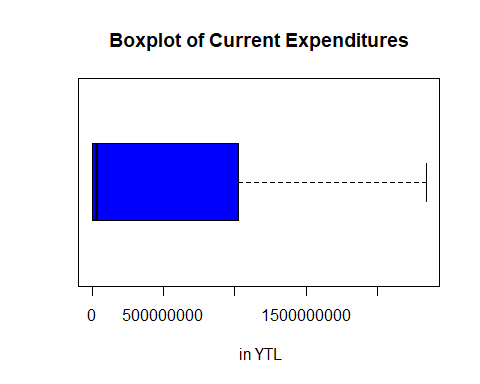
#Histogram  
  
for(i in 1:5){  
 if(i==1) {x=data\_22$CE  
 lab="Current Expenditures"}  
 if(i==2) {x=data\_22$PE  
 lab="Personnel Expenditures"}  
 if(i==3) {x=data\_22$OCE  
 lab="Other Current Expenditures"}  
 if(i==4) {x=data\_22$D  
 lab="Defence"}  
 if(i==5) {x=data\_22$GNP  
 lab="GNP"}  
 w<-round(diff(range(x))/k,1)  
 breaks<- seq(w\*floor(min(x)/w),w\*ceiling(max(x)/w),w)  
   
 hist(x,breaks,right=F,  
 xlab="Prices ",  
 main=paste("Histogram of",lab))  
   
 abline(h=seq(0,60,5),col='gray')  
 hist(x,breaks,col=col[i],add=T,right=F)  
}



#Box PlotS  
  
BE\_list <- (data\_22$BE)  
PE\_list <- (data\_22$PE)  
CE\_list <- (data\_22$CE)  
OCE\_list <- (data\_22$OCE)  
D\_list <- (data\_22$D)  
GNP\_list <- (data\_22$GNP)  
  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

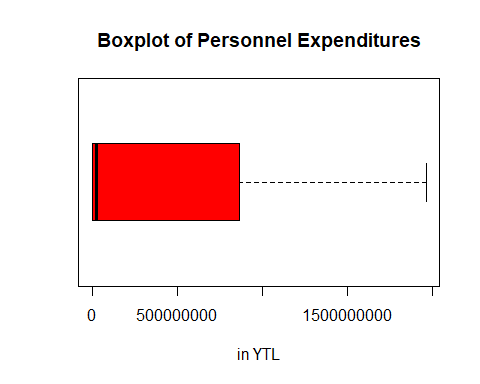
boxplot(CE\_list,main='Boxplot of Current Expenditures',pch="\*",xlab=' in YTL',col='blue',horizontal=T)



"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

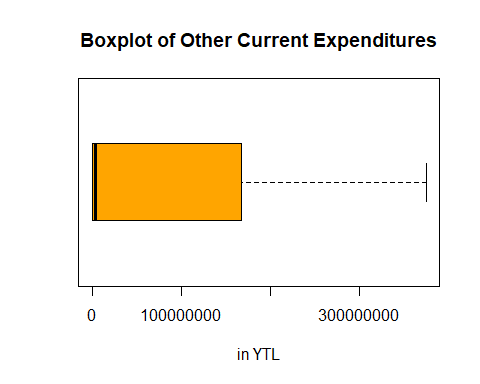
boxplot(PE\_list,main='Boxplot of Personnel Expenditures',pch="\*",xlab='in YTL',col='red',horizontal=T)



"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

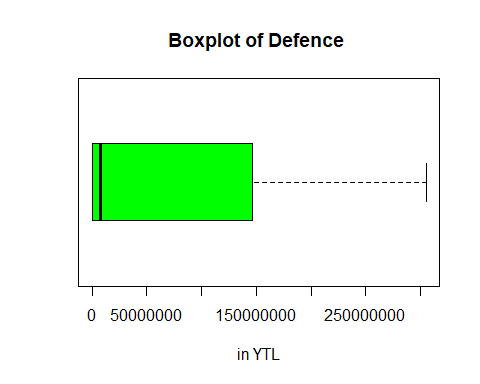
boxplot(OCE\_list,main='Boxplot of Other Current Expenditures',pch="\*",xlab='in YTL',col='orange',horizontal=T)



"FOR DEFENCE"

## [1] "FOR DEFENCE"

boxplot(D\_list,main='Boxplot of Defence ',pch="\*",xlab='in YTL',col='green',horizontal=T)



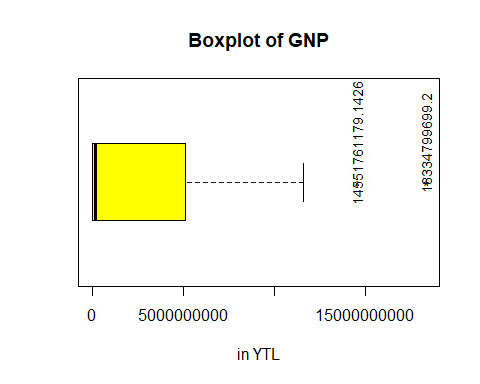
"FOR GNP"

## [1] "FOR GNP"

boxplot(GNP\_list,main='Boxplot of GNP',pch="\*",xlab='in YTL',col='yellow',horizontal=T)  
GNP.out<-boxplot.stats(GNP\_list)$out; GNP.out

## [1] 14551761179 18334799699

text(cex=0.9,y=1.2,x=GNP.out,labels=as.list(GNP.out),srt=90)

  
…

## 3.2. Inferential Statistics

…

#Write your codes here. Their results will appear after knitting the markdown file.

…

### 3.2.1. Confidence Intervals

Confidence interval estimation (CI) is the range of possible values for the population parameter depends on point estimates such as, sample mean and level of confidence and standard deviation. …

#Write your codes here. Their results will appear after knitting the markdown file.

…

#### 3.2.1.1. Confidence Intervals for the Population Mean/Proportion of a Single Sample

Confidence interval for the population means are calculated based on sample size, mean, and standard deviation of the sample data. n>30 sample size is calculated in z test function where variance is unknown and also prediction intervals were calculated to compare the confidence intervals result …

"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

xbar1<-mean(data\_22$CE); xbar1

## [1] 467141360

sd1<-sd(data\_22$CE); sd1

## [1] 657878573

#Find the size of the sample  
n1<-42  
#Since sample size is greater than 42, we can approximately use z-statistic and the sample standard deviation.  
  
#The margin of error is found for the large sample confidence interval as follows:  
E1 <- qnorm(.975)\*sd1/sqrt(n1); E1

## [1] 198961565

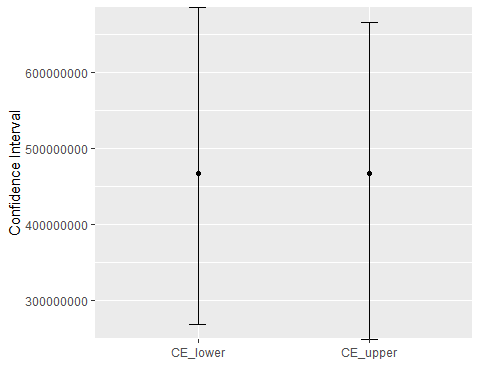
#The two-sided large sample confidence bound for mu is:  
xbar1 + c(-E1, E1)

## [1] 268179796 666102925

#Using the z-test function:  
z.test(data\_22$CE,alternative="two.sided",sigma.x =sd1,conf.level = 0.95)$conf.int

## [1] 268179796 666102925  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('CE\_lower', 'CE\_upper'),xbar1,c(E1,Inf),c(Inf,E1),'add')



#If we used t-statistic instead of z-statistic, the CI will be found as:  
E12 <- qt(.975,df=n1-1)\*sd1/sqrt(n1); E12

## [1] 205009395

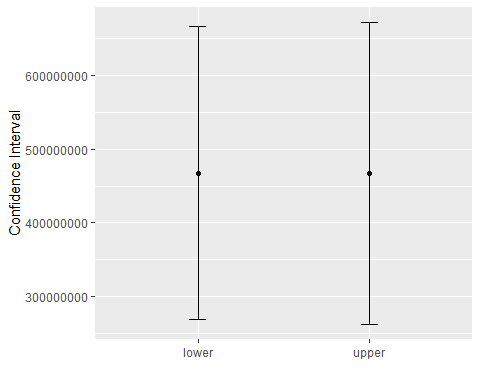
xbar1 + c(-E12, E12)

## [1] 262131965 672150756

#Alternatively, t.test built-in function can be called.  
t.test(data\_22$CE,alternative="two.sided", conf.level=0.95)$conf.int

## [1] 262131965 672150756  
## attr(,"conf.level")  
## [1] 0.95

#We can plot both confidence intervals  
ci.plot(c('lower', 'upper'),xbar1,c(E1,E12),c(E1,E12),'add')



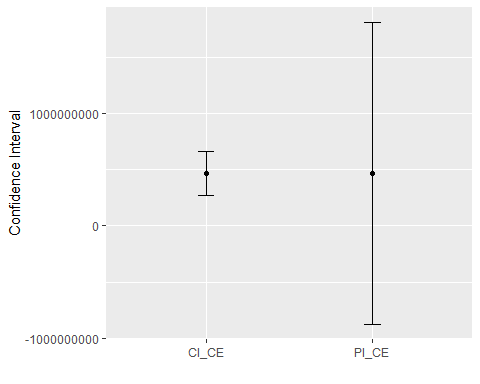
#Calculate the 95% prediction interval for a new observation  
P <- qt(.975,df=n1-1)\* sd1\*sqrt(1+1/n1); P

## [1] 1344336507

xbar1 + c(-P, P)

## [1] -877195146 1811477867

#We can plot both the confidence interval and the prediction interval  
ci.plot(c('CI\_CE', 'PI\_CE'),xbar1,c(E1, P),c(E1,P),'add')



"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

xbar2<-mean(data\_22$PE); xbar2

## [1] 394109263

sd2<-sd(data\_22$PE); sd2

## [1] 554389981

n2<-42  
E2 <- qnorm(.975)\*sd2/sqrt(n2); E2

## [1] 167663612

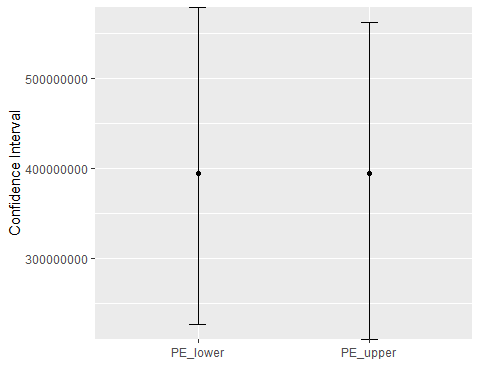
xbar2 + c(-E2, E2)

## [1] 226445650 561772875

z.test(data\_22$PE,alternative="two.sided",sigma.x =sd2,conf.level = 0.95)$conf.int

## [1] 226445650 561772875  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('PE\_lower', 'PE\_upper'),xbar2,c(E2,Inf),c(Inf,E2),'add')



#If we used t-statistic instead of z-statistic, the CI will be found as:  
E22 <- qt(.975,df=n2-1)\*sd2/sqrt(n2); E22

## [1] 172760080

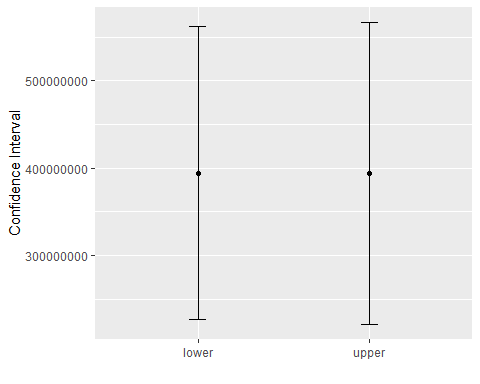
xbar2 + c(-E22, E22)

## [1] 221349183 566869342

t.test(data\_22$PE,alternative="two.sided", conf.level=0.95)$conf.int

## [1] 221349183 566869342  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('lower', 'upper'),xbar2,c(E2,E22),c(E2,E22),'add')



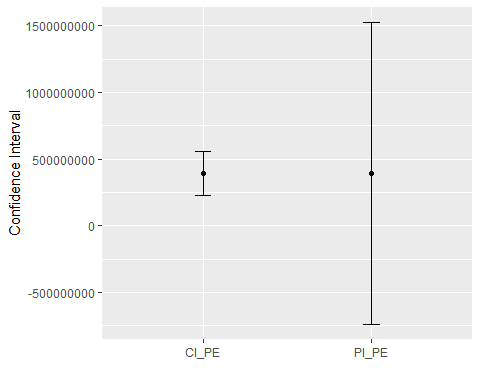
P <- qt(.975,df=n2-1)\* sd2\*sqrt(1+1/n2); P

## [1] 1132863602

xbar2 + c(-P, P)

## [1] -738754339 1526972865

ci.plot(c('CI\_PE', 'PI\_PE'),xbar2,c(E2, P),c(E2,P),'add')



"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

xbar3<-mean(data\_22$OCE); xbar3

## [1] 73032098

sd3<-sd(data\_22$OCE); sd3

## [1] 103715308

n3<-42  
E3 <- qnorm(.975)\*sd3/sqrt(n3); E3

## [1] 31366518

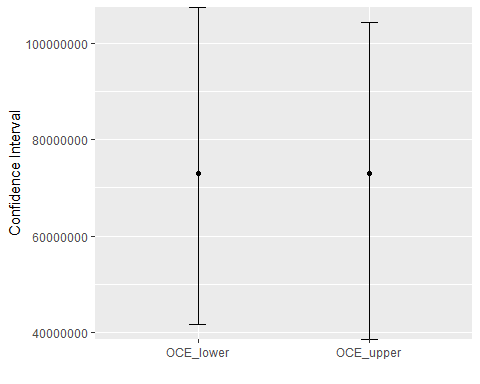
xbar3 + c(-E3, E3)

## [1] 41665580 104398615

z.test(data\_22$OCE,alternative="two.sided",sigma.x =sd3,conf.level = 0.95)$conf.int

## [1] 41665580 104398615  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('OCE\_lower', 'OCE\_upper'),xbar3,c(E3,Inf),c(Inf,E3),'add')



#If we used t-statistic instead of z-statistic, the CI will be found as:  
E32 <- qt(.975,df=n3-1)\*sd3/sqrt(n3); E32

## [1] 32319965

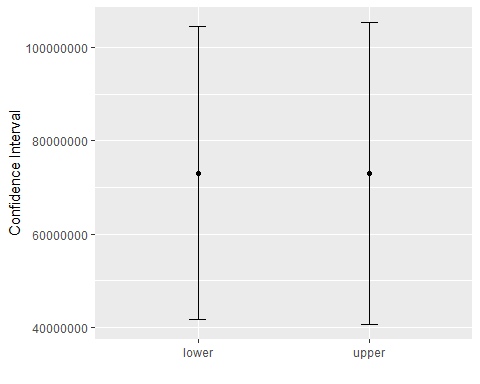
xbar3 + c(-E32, E32)

## [1] 40712133 105352063

t.test(data\_22$OCE,alternative="two.sided", conf.level=0.95)$conf.int

## [1] 40712133 105352063  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('lower', 'upper'),xbar3,c(E3,E32),c(E3,E32),'add')



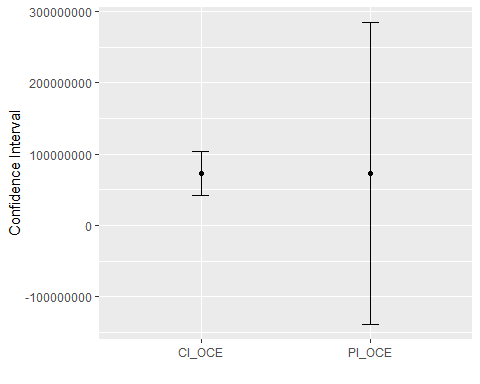
P <- qt(.975,df=n3-1)\* sd3\*sqrt(1+1/n3); P

## [1] 211936184

xbar3 + c(-P, P)

## [1] -138904086 284968282

ci.plot(c('CI\_OCE', 'PI\_OCE'),xbar3,c(E3, P),c(E3,P),'add')



"FOR DEFENCE"

## [1] "FOR DEFENCE"

xbar4<-mean(data\_22$D); xbar4

## [1] 68808754

sd4<-sd(data\_22$D); sd4

## [1] 91449014

n4<-42  
E4 <- qnorm(.975)\*sd4/sqrt(n4); E4

## [1] 27656835

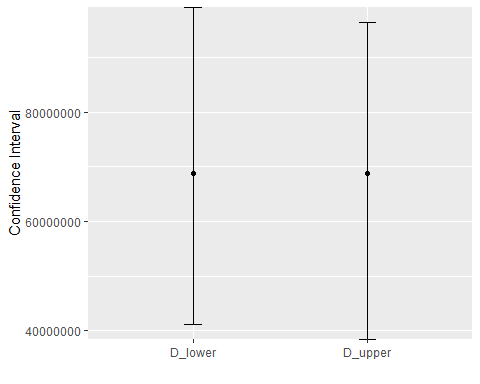
xbar4 + c(-E4, E4)

## [1] 41151919 96465589

z.test(data\_22$D,alternative="two.sided",sigma.x =sd4,conf.level = 0.95)$conf.int

## [1] 41151919 96465589  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('D\_lower', 'D\_upper'),xbar4,c(E4,Inf),c(Inf,E4),'add')



#If we used t-statistic instead of z-statistic, the CI will be found as:  
E42 <- qt(.975,df=n4-1)\*sd4/sqrt(n4); E42

## [1] 28497519

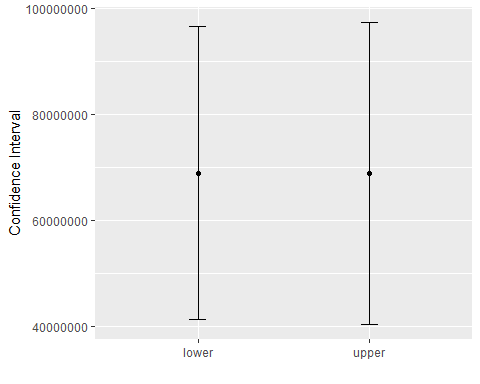
xbar4 + c(-E42, E42)

## [1] 40311235 97306273

t.test(data\_22$D,alternative="two.sided", conf.level=0.95)$conf.int

## [1] 40311235 97306273  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('lower', 'upper'),xbar4,c(E4,E42),c(E4,E42),'add')



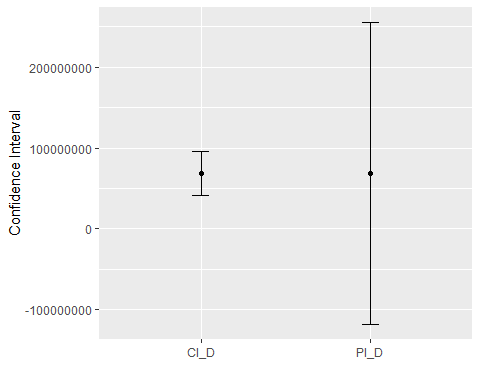
P <- qt(.975,df=n4-1)\* sd4\*sqrt(1+1/n4); P

## [1] 186870729

xbar4 + c(-P, P)

## [1] -118061974 255679483

ci.plot(c('CI\_D', 'PI\_D'),xbar4,c(E4, P),c(E4,P),'add')



"FOR GNP"

## [1] "FOR GNP"

xbar5<-mean(data\_22$GNP); xbar5

## [1] 2884909571

sd5<-sd(data\_22$GNP); sd5

## [1] 4454768144

n5<-42  
E5 <- qnorm(.975)\*sd5/sqrt(n5); E5

## [1] 1347251113

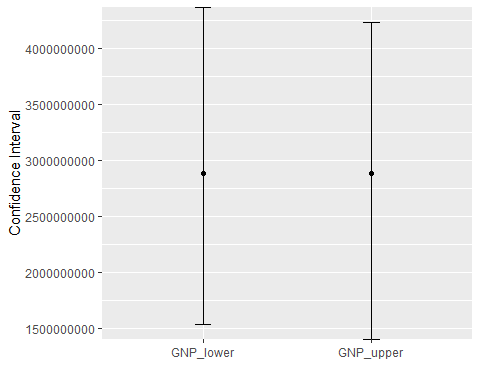
xbar5 + c(-E5, E5)

## [1] 1537658458 4232160685

z.test(data\_22$GNP,alternative="two.sided",sigma.x =sd5,conf.level = 0.95)$conf.int

## [1] 1537658458 4232160685  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('GNP\_lower', 'GNP\_upper'),xbar5,c(E5,Inf),c(Inf,E5),'add')



#If we used t-statistic instead of z-statistic, the CI will be found as:  
E52 <- qt(.975,df=n5-1)\*sd5/sqrt(n5); E52

## [1] 1388203478

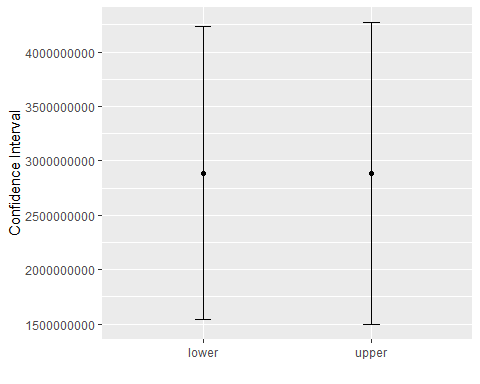
xbar5 + c(-E52, E52)

## [1] 1496706093 4273113050

t.test(data\_22$GNP,alternative="two.sided", conf.level=0.95)$conf.int

## [1] 1496706093 4273113050  
## attr(,"conf.level")  
## [1] 0.95

ci.plot(c('lower', 'upper'),xbar5,c(E5,E52),c(E5,E52),'add')



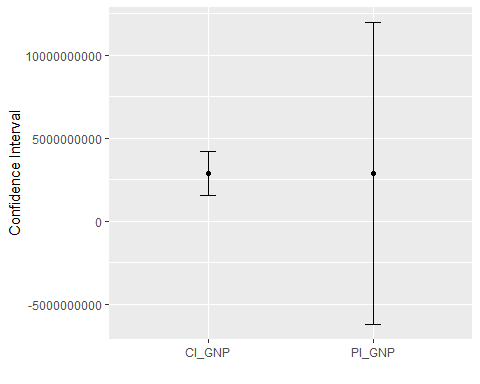
P <- qt(.975,df=n5-1)\* sd5\*sqrt(1+1/n5); P

## [1] 9103058969

xbar5 + c(-P, P)

## [1] -6218149398 11987968540

ci.plot(c('CI\_GNP', 'PI\_GNP'),xbar5,c(E5, P),c(E5,P),'add')

  
…

#### 3.2.1.2. Confidence Intervals for the Population Variance of a Single Sample

Instead of the normally distributed sample calculation, the calculation is made according to the chi square distribution via using sample mean, variance, and standard deviation to find confidence intervals for the population variance. All results were commented in word documents …

"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

#Find sample size, sample mean and sample variance  
n<-length(data\_22$CE); n

## [1] 42

xbar<-mean(data\_22$CE); xbar

## [1] 467141360

var<-var(data\_22$CE); var

## [1] 432804216879521536

sd<- sd(data\_22$CE); sd

## [1] 657878573

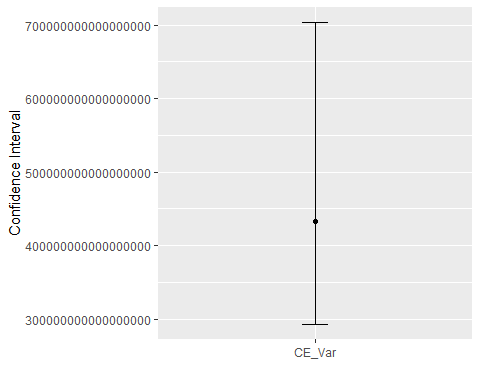
#Alternatively, we can use varTest() function from EnvStats package.  
require(EnvStats)  
sigmasqCI<-varTest(data\_22$CE, alternative="two.sided",conf.level=0.95,sigma.squared = var)$conf.int; sigmasqCI

## LCL UCL   
## 293011977656920768 703760129104281856   
## attr(,"conf.level")  
## [1] 0.95

sigmaCI<-sqrt(sigmasqCI); sigmaCI

## LCL UCL   
## 541305808 838904124   
## attr(,"conf.level")  
## [1] 0.95

ci.plot('CE\_Var',var,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

n1<-length(data\_22$PE); n1

## [1] 42

xbar1<-mean(data\_22$PE); xbar1

## [1] 394109263

var1<-var(data\_22$PE); var1

## [1] 307348250641565056

sd1<- sd(data\_22$PE); sd1

## [1] 554389981

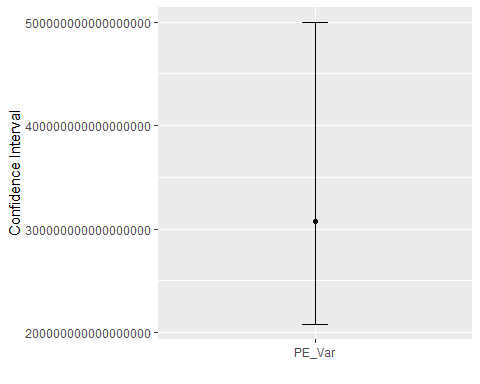
#Alternatively, we can use varTest() function from EnvStats package.  
require(EnvStats)  
sigmasqCI<-varTest(data\_22$PE, alternative="two.sided",conf.level=0.95,sigma.squared = var1)$conf.int; sigmasqCI

## LCL UCL   
## 208077267359316768 499762793696840512   
## attr(,"conf.level")  
## [1] 0.95

sigmaCI<-sqrt(sigmasqCI); sigmaCI

## LCL UCL   
## 456154872 706939031   
## attr(,"conf.level")  
## [1] 0.95

ci.plot('PE\_Var',var1,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

n2<-length(data\_22$OCE); n2

## [1] 42

xbar2<-mean(data\_22$OCE); xbar2

## [1] 73032098

var2<-var(data\_22$OCE); var2

## [1] 10756865022613600

sd2<- sd(data\_22$OCE); sd2

## [1] 103715308

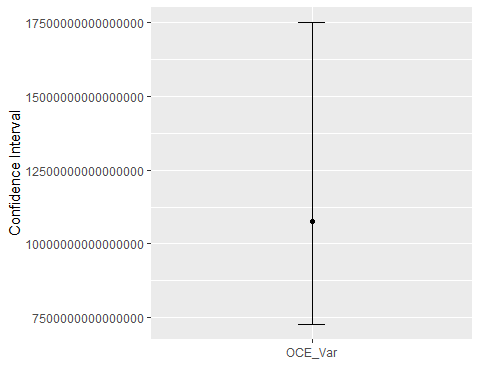
require(EnvStats)  
sigmasqCI<-varTest(data\_22$OCE, alternative="two.sided",conf.level=0.95,sigma.squared = var2)$conf.int; sigmasqCI

## LCL UCL   
## 7282485176298436 17491171346833688   
## attr(,"conf.level")  
## [1] 0.95

sigmaCI<-sqrt(sigmasqCI); sigmaCI

## LCL UCL   
## 85337478 132254192   
## attr(,"conf.level")  
## [1] 0.95

ci.plot('OCE\_Var',var2,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR DEFENCE"

## [1] "FOR DEFENCE"

n3<-length(data\_22$D); n3

## [1] 42

xbar3<-mean(data\_22$D); xbar3

## [1] 68808754

var3<-var(data\_22$D); var3

## [1] 8362922193236486

sd3<- sd(data\_22$D); sd3

## [1] 91449014

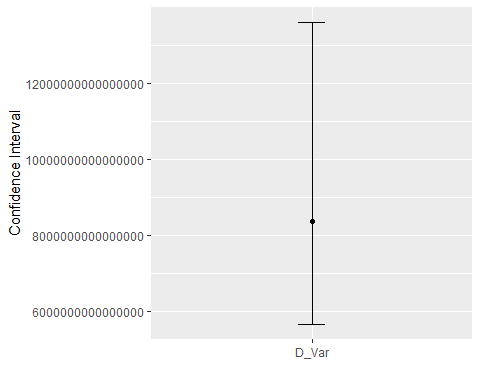
require(EnvStats)  
sigmasqCI<-varTest(data\_22$D, alternative="two.sided",conf.level=0.95,sigma.squared = var3)$conf.int; sigmasqCI

## LCL UCL   
## 5661766395204270 13598507068242126   
## attr(,"conf.level")  
## [1] 0.95

sigmaCI<-sqrt(sigmasqCI); sigmaCI

## LCL UCL   
## 75244710 116612637   
## attr(,"conf.level")  
## [1] 0.95

ci.plot('D\_Var',var3,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR GNP"

## [1] "FOR GNP"

n4<-length(data\_22$GNP); n4

## [1] 42

xbar4<-mean(data\_22$GNP); xbar4

## [1] 2884909571

var4<-var(data\_22$GNP); var4

## [1] 19844959216553959424

sd4<- sd(data\_22$GNP); sd4

## [1] 4454768144

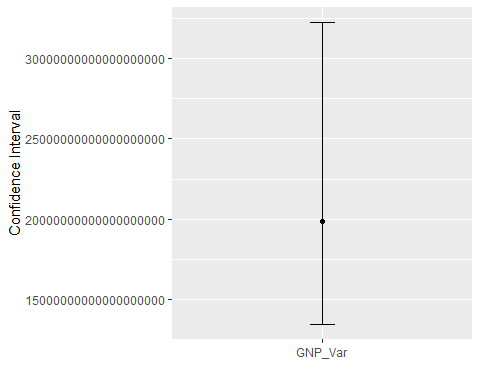
require(EnvStats)  
sigmasqCI<-varTest(data\_22$GNP, alternative="two.sided",conf.level=0.95,sigma.squared = var4)$conf.int; sigmasqCI

## LCL UCL   
## 13435198918549504000 32268842390227734528   
## attr(,"conf.level")  
## [1] 0.95

sigmaCI<-sqrt(sigmasqCI); sigmaCI

## LCL UCL   
## 3665405696 5680567084   
## attr(,"conf.level")  
## [1] 0.95

ci.plot('GNP\_Var',var4,sigmasqCI[1],sigmasqCI[2],'multiply')

  
…

#### 3.2.1.3. Confidence Intervals for the Difference of Population Means/Proportions of Two Samples

Since, population standard deviation is unknown and our sample size is 42, z test was applied for calculating confidence intervals for difference of population mean of two sample data. …

"FOR PERSONNEL EXPENDITURES and OTHER CURRENT EXPENDITURES CONFIDENCE INTERVAL ON MEANS"

## [1] "FOR PERSONNEL EXPENDITURES and OTHER CURRENT EXPENDITURES CONFIDENCE INTERVAL ON MEANS"

#Samples size, samples mean and standard deviation of samples was called.  
n10<- length(data\_22$PE) ; n10

## [1] 42

n11<- length(data\_22$OCE) ;n11

## [1] 42

xbar10<-mean(data\_22$PE); xbar10

## [1] 394109263

xbar11<- mean(data\_22$OCE); xbar11

## [1] 73032098

sd10<-sd(data\_22$PE); sd10

## [1] 554389981

sd11<-sd(data\_22$OCE); sd11

## [1] 103715308

#zsum.test() function from the BSDA package was used. Since sample size is greater than 30 , we can approximately use z-statistic and then the 95% CI was found.  
  
zsum.test(xbar10, sd10, n10, xbar11, sd11, n11, alternative="two.sided", conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 3.7, p-value = 0.0002  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 150504760 491649570  
## sample estimates:  
## mean of x mean of y   
## 394109263 73032098

#The standard error of the difference in means was calculated.  
sed10<-sqrt(sd10^2/n10+sd11^2/n11); sed10

## [1] 87028336

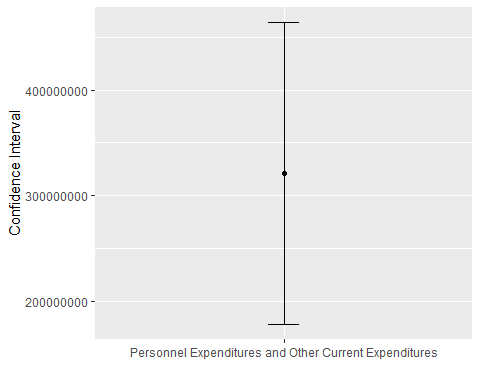
# zalpha/2 was multiplied with the standard error of the difference in means and the margin of error was obtained.  
E10 <- qnorm(.95)\*sed10; E10

## [1] 143148875

#The difference of sample means was added and so,the confidence interval was found.  
xbar10-xbar11 + c(-E10, E10)

## [1] 177928290 464226040

#Then confidence intervals was plotted for both.  
ci.plot("Personnel Expenditures and Other Current Expenditures",xbar10-xbar11,E10,E10,'add')



"FOR PERSONNEL EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON MEANS"

## [1] "FOR PERSONNEL EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON MEANS"

#Samples size, samples mean and standard deviation of samples was called.  
n12<- length(data\_22$PE) ; n12

## [1] 42

n13<- length(data\_22$D) ;n13

## [1] 42

xbar12<-mean(data\_22$PE); xbar12

## [1] 394109263

xbar13<- mean(data\_22$D); xbar13

## [1] 68808754

sd12<-sd(data\_22$PE); sd12

## [1] 554389981

sd13<-sd(data\_22$D); sd13

## [1] 91449014

#zsum.test() function from the BSDA package was used. Since sample size is greater than 30 , we can approximately use z-statistic and then the 95% CI was found.  
zsum.test(xbar12, sd12, n12, xbar13, sd13, n13, alternative="two.sided", conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 3.8, p-value = 0.0002  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 155371149 495229868  
## sample estimates:  
## mean of x mean of y   
## 394109263 68808754

#The standard error of the difference in means was calculated.  
sed11<-sqrt(sd12^2/n12+sd13^2/n13); sed11

## [1] 86700246

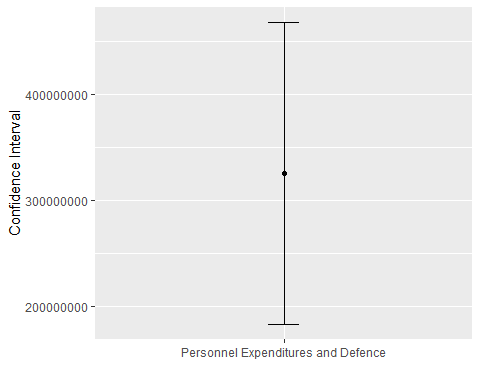
# zalpha/2 was multiplied with the standard error of the difference in means and the margin of error was obtained.  
E11 <- qnorm(.95)\*sed11; E11

## [1] 142609214

#The difference of sample means was added and so,the confidence interval was found.  
xbar12-xbar13 + c(-E11, E11)

## [1] 182691294 467909723

#Then confidence intervals was plotted for both.  
ci.plot("Personnel Expenditures and Defence",xbar12-xbar13,E11,E11,'add')



"FOR OTHER CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON MEANS"

## [1] "FOR OTHER CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON MEANS"

#Samples size, samples mean and standard deviation of samples was called.  
n14<- length(data\_22$OCE) ; n14

## [1] 42

n15<- length(data\_22$D) ;n15

## [1] 42

xbar14<-mean(data\_22$OCE); xbar14

## [1] 73032098

xbar15<- mean(data\_22$D); xbar15

## [1] 68808754

sd14<-sd(data\_22$OCE); sd14

## [1] 103715308

sd15<-sd(data\_22$D); sd15

## [1] 91449014

#zsum.test() function from the BSDA package was used. Since sample size is greater than 30 , we can approximately use z-statistic and then the 95% CI was found.  
zsum.test(xbar14, sd14, n14, xbar15, sd15, n15, alternative="two.sided", conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 0.2, p-value = 0.8  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -37594822 46041509  
## sample estimates:  
## mean of x mean of y   
## 73032098 68808754

#The standard error of the difference in means was calculated.  
sed12<-sqrt(sd14^2/n14+sd15^2/n15); sed12

## [1] 21336191

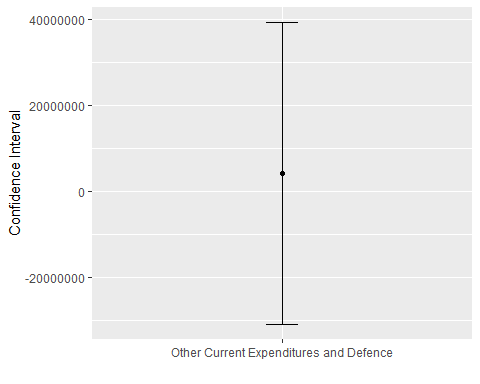
# zalpha/2 was multiplied with the standard error of the difference in means and the margin of error was obtained.  
E12 <- qnorm(.95)\*sed12; E12

## [1] 35094910

#The difference of sample means was added and so,the confidence interval was found.  
xbar14-xbar15 + c(-E12, E12)

## [1] -30871567 39318254

#Then confidence intervals was plotted for both.  
ci.plot("Other Current Expenditures and Defence",xbar14-xbar15,E12,E12,'add')



"FOR CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON MEANS"

## [1] "FOR CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON MEANS"

#Samples size, samples mean and standard deviation of samples was called.  
n16<- length(data\_22$CE) ; n16

## [1] 42

n17<- length(data\_22$D) ;n17

## [1] 42

xbar16<-mean(data\_22$CE); xbar16

## [1] 467141360

xbar17<- mean(data\_22$D); xbar17

## [1] 68808754

sd16<-sd(data\_22$CE); sd16

## [1] 657878573

sd17<-sd(data\_22$D); sd17

## [1] 91449014

#zsum.test() function from the BSDA package was used. Since sample size is greater than 30 , we can approximately use z-statistic and then the 95% CI was found.  
zsum.test(xbar16, sd16, n16, xbar17, sd17, n17, alternative="two.sided", conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 3.9, p-value = 0.0001  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 197458007 599207206  
## sample estimates:  
## mean of x mean of y   
## 467141360 68808754

#The standard error of the difference in means was calculated.  
sed13<-sqrt(sd16^2/n16+sd17^2/n17); sed13

## [1] 102488924

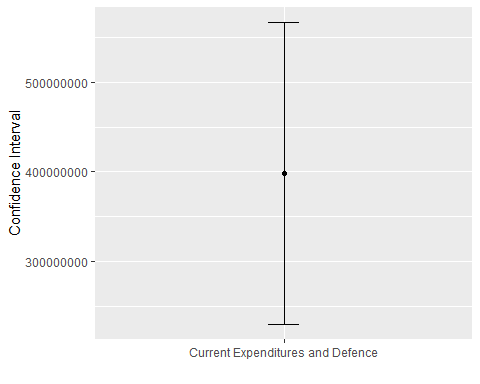
# zalpha/2 was multiplied with the standard error of the difference in means and the margin of error was obtained.  
E13 <- qnorm(.95)\*sed13; E13

## [1] 168579278

#The difference of sample means was added and so,the confidence interval was found.  
xbar16-xbar17 + c(-E13, E13)

## [1] 229753328 566911884

#Then confidence intervals was plotted for both.  
ci.plot("Current Expenditures and Defence",xbar16-xbar17,E13,E13,'add')

  
…

#### 3.2.1.4. Confidence Intervals for the Ratio of Population Variances of Two Samples

F test is applied for determining calculating confidence intervals. CI are calculated according to population variances of two samples. …

"FOR PERSONNEL EXPENDITURES and OTHER CURRENT EXPENDITURES CONFIDENCE INTERVALS ON VARIANCES "

## [1] "FOR PERSONNEL EXPENDITURES and OTHER CURRENT EXPENDITURES CONFIDENCE INTERVALS ON VARIANCES "

#Samples size and standard deviation of samples was called.  
n1<-length(data\_22$PE)  
n2<-length(data\_22$OCE)  
sd1<-sd(data\_22$PE);sd1

## [1] 554389981

sd2<-sd(data\_22$OCE);sd2

## [1] 103715308

#Since, we have two variances,F statistic was used.Then 95% CI was calculated for the two population variances.  
fclo <- qf(0.975,n2-1,n1-1,lower=F); fclo

## [1] 0.5375

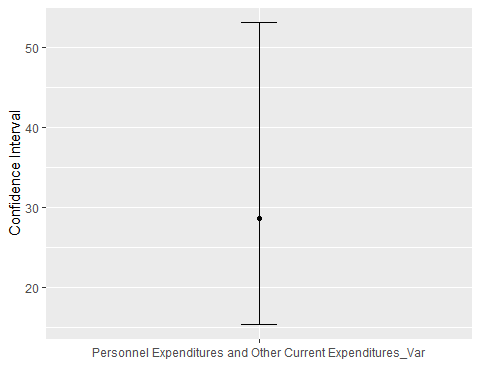
fcup <- qf(0.025,n2-1,n1-1,lower=F); fcup

## [1] 1.86

sigmasqCI <-(sd1/sd2)^2\*c(fclo,fcup); sigmasqCI

## [1] 15.36 53.16

#Confidence interval was plotted.  
ci.plot('Personnel Expenditures and Other Current Expenditures\_Var',(sd1/sd2)^2,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR PERSONNEL EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON VARIANCES "

## [1] "FOR PERSONNEL EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON VARIANCES "

#Samples size and standard deviation of samples was called.  
n3<-length(data\_22$PE);n3

## [1] 42

n4<-length(data\_22$D);n4

## [1] 42

sd3<-sd(data\_22$PE);sd3

## [1] 554389981

sd4<-sd(data\_22$D);sd4

## [1] 91449014

#Since, we have two variances,F statistic was used.Then 95% CI was calculated for the two population variances.  
fclo <- qf(0.975,n4-1,n3-1,lower=F); fclo

## [1] 0.5375

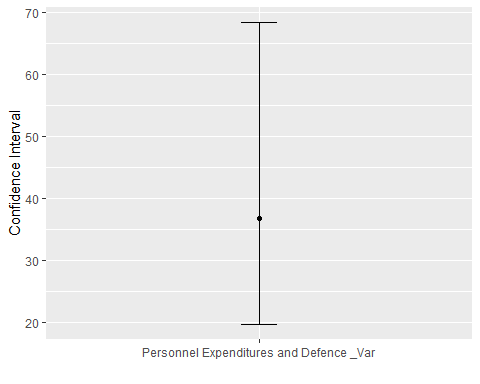
fcup <- qf(0.025,n4-1,n3-1,lower=F); fcup

## [1] 1.86

sigmasqCI <-(sd3/sd4)^2\*c(fclo,fcup); sigmasqCI

## [1] 19.75 68.37

#Confidence interval was plotted.  
ci.plot('Personnel Expenditures and Defence \_Var',(sd3/sd4)^2,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR OTHER CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON VARIANCES "

## [1] "FOR OTHER CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON VARIANCES "

#Samples size and standard deviation of samples was called.  
n5<-length(data\_22$OCE);n5

## [1] 42

n6<-length(data\_22$D);n6

## [1] 42

sd5<-sd(data\_22$OCE);sd5

## [1] 103715308

sd6<-sd(data\_22$D);sd6

## [1] 91449014

#Since, we have two variances,F statistic was used.Then 95% CI was calculated for the two population variances.  
fclo <- qf(0.975,n6-1,n5-1,lower=F); fclo

## [1] 0.5375

fcup <- qf(0.025,n6-1,n5-1,lower=F); fcup

## [1] 1.86

sigmasqCI <-(sd5/sd6)^2\*c(fclo,fcup); sigmasqCI

## [1] 0.6914 2.3929

#Confidence interval was plotted.  
ci.plot('Other Current Expenditures and Defence \_Var',(sd5/sd6)^2,sigmasqCI[1],sigmasqCI[2],'multiply')



"FOR CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON VARIANCES "

## [1] "FOR CURRENT EXPENDITURES and DEFENCE CONFIDENCE INTERVALS ON VARIANCES "

#Samples size and standard deviation of samples was called.  
n7<-length(data\_22$CE);n7

## [1] 42

n8<-length(data\_22$D);n8

## [1] 42

sd7<-sd(data\_22$CE);sd7

## [1] 657878573

sd8<-sd(data\_22$D);sd8

## [1] 91449014

#Since, we have two variances,F statistic was used.Then 95% CI was calculated for the two population variances.  
fclo <- qf(0.975,n8-1,n7-1,lower=F); fclo

## [1] 0.5375

fcup <- qf(0.025,n8-1,n7-1,lower=F); fcup

## [1] 1.86

sigmasqCI <-(sd7/sd8)^2\*c(fclo,fcup); sigmasqCI

## [1] 27.82 96.28

#Confidence interval was plotted.  
ci.plot('Current Expenditures and Defence \_Var',(sd7/sd8)^2,sigmasqCI[1],sigmasqCI[2],'multiply')

  
…

### 3.2.2. Hypothesis Tests

Hypothesis testing is the method used to determine the accuracy of a hypothesis within a statistical reliability range. To test hypothesis made about population parameters, sample statistics was used. …

#Write your codes here. Their results will appear after knitting the markdown file.

…

#### 3.2.2.1. Hypothesis Tests on the Population Mean/Proportion of a Single Sample

Z test is applied to conduct hypothesis test, since n>30 and sample standard deviation is known.Hypothesized values were determined and also beta tests were conducted the measure the power and judgment of the decision …

#Write your codes here. Their results will appear after knitting the markdown file.  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

#We want to test the null hypothesis H0: mu=500000000 against the less-than one sided alternative using critical value approach  
#Define problem data  
n<-42 #sample size  
mu0<-500000000 #hypothesized value  
sigma<-sd(data\_22$CE) #population standard deviation  
xbar<-mean(data\_22$CE) #sample mean  
alpha<-0.05 #significance level  
#Calculate the critical value at 0.05 significance level  
zc<-qnorm(alpha); zc

## [1] -1.645

#Calculate the test statistic Z0:  
z0<-(xbar-mu0)/(sigma/sqrt(n)); z0

## [1] -0.3237

#Use our test.critical function to give the decision  
testmean.cv(zc,z0,mu0,alpha,alt="less",theta="mu")

## [1] "H0: mu = 500000000 is not rejected at 0.05 significance level"

###We want to test the null hypothesis H0: mu=500000000 against the less-than one sided alternative using p-value approach  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(mean.x=xbar, sigma.x=sigma, n.x=n, alt='less', mu=mu0,conf.level=0.95)

##   
## One-sample z-Test  
##   
## data: Summarized x  
## z = -0.32, p-value = 0.4  
## alternative hypothesis: true mean is less than 500000000  
## 95 percent confidence interval:  
## NA 634115169  
## sample estimates:  
## mean of x   
## 467141360

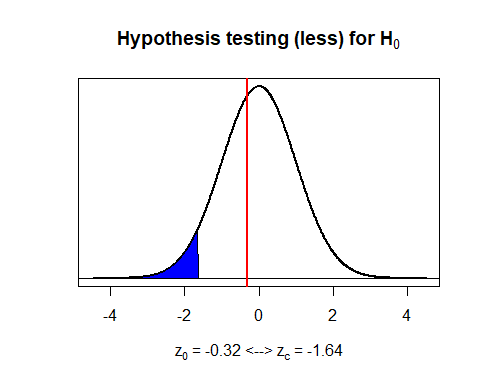
#For manual calculation, calculate the p-value  
pval<-pnorm(z0); pval

## [1] 0.3731

#Use our test.pval function to give the decision  
test.pval(pval,mu0,alpha,theta="mu")

## [1] "H0: mu = 500000000 is not rejected at 0.05 significance level"

#Use our plotmean.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z0,zc,n,type='z',alt='less')



"TEST FOR CURRENT EXPENDITURES"

## [1] "TEST FOR CURRENT EXPENDITURES"

#Define true parameter value and delta, and calculate power of the test with pwr.norm.test function:   
mu01 <- 480000000  
delta <- mu01-mu0; delta

## [1] -20000000

pwrNorm <- pwr.norm.test(delta/sigma,n,alpha,power=NULL,alt='less'); pwrNorm

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = -0.0304  
## n = 42  
## sig.level = 0.05  
## power = 0.07383  
## alternative = less

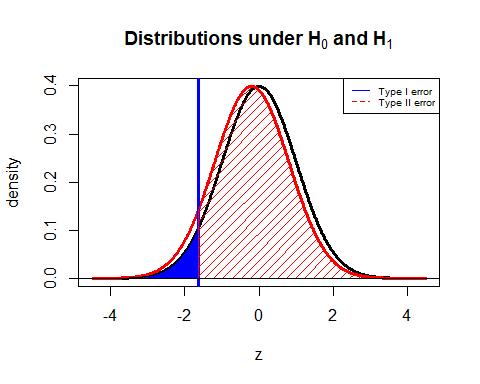
beta <- 1-pwrNorm$power; beta

## [1] 0.9262

power <- 1-beta; power

## [1] 0.07383

plotmean.beta(sigma,zc,delta,n,type='z',alt='less')



"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

#We want to test the null hypothesis H0: mu=400000000.0 against the less-than one sided alternative using critical value approach  
#Define problem data  
n<-42 #sample size  
mu1<-400000000.0 #hypothesized value  
sigma1<-sd(data\_22$PE) #population standard deviation  
xbar1<-mean(data\_22$PE) #sample mean  
alpha<-0.05 #significance level  
#Calculate the critical value at 0.05 significance level  
zc<-qnorm(alpha); zc

## [1] -1.645

#Calculate the test statistic Z0:  
z1<-(xbar1-mu1)/(sigma1/sqrt(n)); z1

## [1] -0.06886

#Use our test.critical function to give the decision  
testmean.cv(zc,z1,mu1,alpha,alt="less",theta="mu")

## [1] "H0: mu = 400000000 is not rejected at 0.05 significance level"

###We want to test the null hypothesis H0: mu=400000000.0 against the less-than one sided alternative using p-value approach  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(mean.x=xbar1, sigma.x=sigma1, n.x=n, alt='less', mu=mu1,conf.level=0.95)

##   
## One-sample z-Test  
##   
## data: Summarized x  
## z = -0.069, p-value = 0.5  
## alternative hypothesis: true mean is less than 400000000  
## 95 percent confidence interval:  
## NA 534817002  
## sample estimates:  
## mean of x   
## 394109263

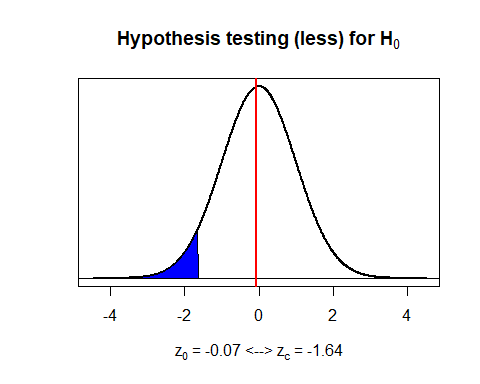
#For manual calculation, calculate the p-value  
pval1<-pnorm(z1); pval1

## [1] 0.4725

#Use our test.pval function to give the decision  
test.pval(pval1,mu1,alpha,theta="mu")

## [1] "H0: mu = 400000000 is not rejected at 0.05 significance level"

#Use our plotmean.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z1,zc,n,type='z',alt='less')



"TEST FOR PERSONNEL EXPENDITURES"

## [1] "TEST FOR PERSONNEL EXPENDITURES"

#Define true parameter value and delta, and calculate power of the test with pwr.norm.test function:   
mu02 <- 380000000.0  
delta1 <- mu02-mu1; delta1

## [1] -20000000

pwrNorm1 <- pwr.norm.test(delta1/sigma1,n,alpha,power=NULL,alt='less'); pwrNorm1

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = -0.03608  
## n = 42  
## sig.level = 0.05  
## power = 0.07911  
## alternative = less

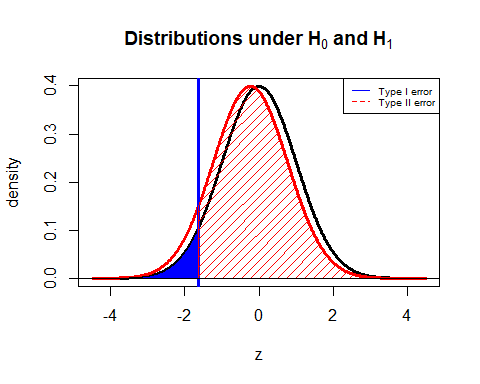
beta1 <- 1-pwrNorm1$power; beta1

## [1] 0.9209

power1 <- 1-beta1; power1

## [1] 0.07911

plotmean.beta(sigma1,zc,delta1,n,type='z',alt='less')



"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

#We want to test the null hypothesis H0: mu=80000000.0 against the less-than one sided alternative using critical value approach  
#Define problem data  
n<-42 #sample size  
mu2<-80000000.0 #hypothesized value  
sigma2<-sd(data\_22$OCE) #population standard deviation  
xbar2<-mean(data\_22$OCE) #sample mean  
alpha<-0.05 #significance level  
#Calculate the critical value at 0.05 significance level  
zc<-qnorm(alpha); zc

## [1] -1.645

#Calculate the test statistic Z0:  
z2<-(xbar2-mu2)/(sigma2/sqrt(n)); z2

## [1] -0.4354

#Use our test.critical function to give the decision  
testmean.cv(zc,z2,mu2,alpha,alt="less",theta="mu")

## [1] "H0: mu = 80000000 is not rejected at 0.05 significance level"

###We want to test the null hypothesis H0: mu=80000000.0 against the less-than one sided alternative using p-value approach  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(mean.x=xbar2, sigma.x=sigma2, n.x=n, alt='less', mu=mu2,conf.level=0.95)

##   
## One-sample z-Test  
##   
## data: Summarized x  
## z = -0.44, p-value = 0.3  
## alternative hypothesis: true mean is less than 80000000  
## 95 percent confidence interval:  
## NA 99355709  
## sample estimates:  
## mean of x   
## 73032098

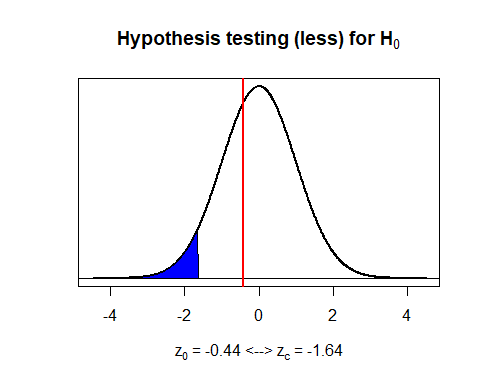
#For manual calculation, calculate the p-value  
pval2<-pnorm(z2); pval2

## [1] 0.3316

#Use our test.pval function to give the decision  
test.pval(pval2,mu2,alpha,theta="mu")

## [1] "H0: mu = 80000000 is not rejected at 0.05 significance level"

#Use our plotmean.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z2,zc,n,type='z',alt='less')



"TEST FOR OTHER CURRENT EXPENDITURES"

## [1] "TEST FOR OTHER CURRENT EXPENDITURES"

#Define true parameter value and delta, and calculate power of the test with pwr.norm.test function:   
mu03 <- 76000000.0  
delta2 <- mu03-mu2; delta2

## [1] -4000000

pwrNorm2 <- pwr.norm.test(delta2/sigma2,n,alpha,power=NULL,alt='less'); pwrNorm2

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = -0.03857  
## n = 42  
## sig.level = 0.05  
## power = 0.08152  
## alternative = less

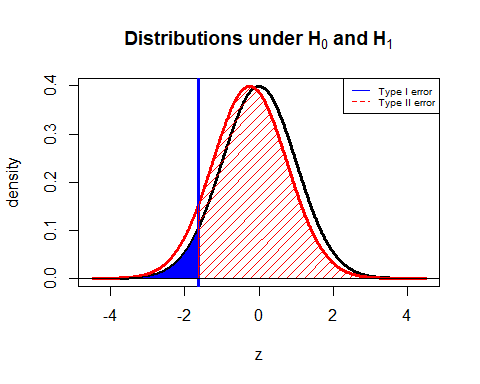
beta2 <- 1-pwrNorm2$power; beta2

## [1] 0.9185

power2 <- 1-beta2; power2

## [1] 0.08152

plotmean.beta(sigma2,zc,delta2,n,type='z',alt='less')



"FOR DEFENCE"

## [1] "FOR DEFENCE"

#We want to test the null hypothesis H0: mu=70000000.0 against the less-than one sided alternative using critical value approach  
#Define problem data  
n<-42 #sample size  
mu3<-70000000.0 #hypothesized value  
sigma3<-sd(data\_22$D) #population standard deviation  
xbar3<-mean(data\_22$D) #sample mean  
alpha<-0.05 #significance level  
#Calculate the critical value at 0.05 significance level  
zc<-qnorm(alpha); zc

## [1] -1.645

#Calculate the test statistic Z0:  
z3<-(xbar3-mu3)/(sigma3/sqrt(n)); z3

## [1] -0.08442

#Use our test.critical function to give the decision  
testmean.cv(zc,z3,mu3,alpha,alt="less",theta="mu")

## [1] "H0: mu = 70000000 is not rejected at 0.05 significance level"

###We want to test the null hypothesis H0: mu=70000000.0 against the less-than one sided alternative using p-value approach  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(mean.x=xbar3, sigma.x=sigma3, n.x=n, alt='less', mu=mu3,conf.level=0.95)

##   
## One-sample z-Test  
##   
## data: Summarized x  
## z = -0.084, p-value = 0.5  
## alternative hypothesis: true mean is less than 70000000  
## 95 percent confidence interval:  
## NA 92019101  
## sample estimates:  
## mean of x   
## 68808754

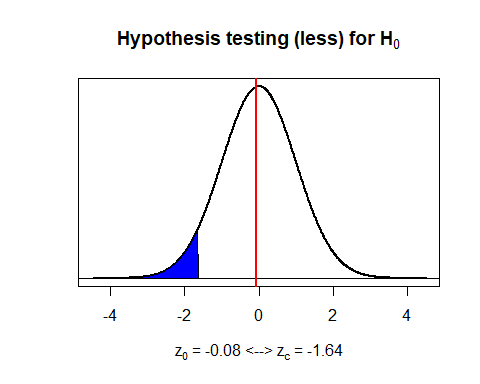
#For manual calculation, calculate the p-value  
pval3<-pnorm(z3); pval3

## [1] 0.4664

#Use our test.pval function to give the decision  
test.pval(pval3,mu3,alpha,theta="mu")

## [1] "H0: mu = 70000000 is not rejected at 0.05 significance level"

#Use our plotmean.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z3,zc,n,type='z',alt='less')



"TEST FOR DEFENCE"

## [1] "TEST FOR DEFENCE"

#Define true parameter value and delta, and calculate power of the test with pwr.norm.test function:   
mu04 <- 69000000.0  
delta3 <- mu04-mu3; delta3

## [1] -1000000

pwrNorm3 <- pwr.norm.test(delta3/sigma3,n,alpha,power=NULL,alt='less'); pwrNorm3

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = -0.01094  
## n = 42  
## sig.level = 0.05  
## power = 0.05775  
## alternative = less

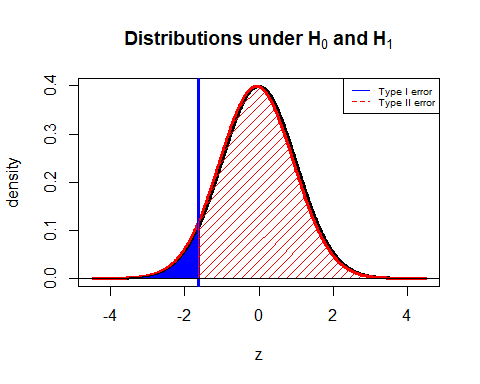
beta3 <- 1-pwrNorm3$power; beta3

## [1] 0.9423

power3 <- 1-beta3; power3

## [1] 0.05775

plotmean.beta(sigma3,zc,delta3,n,type='z',alt='less')



"FOR GNP"

## [1] "FOR GNP"

#We want to test the null hypothesis H0: mu=3000000000 against the less-than one sided alternative using critical value approach  
#Define problem data  
n<-42 #sample size  
mu4<-3000000000 #hypothesized value  
sigma4<-sd(data\_22$GNP) #population standard deviation  
xbar4<-mean(data\_22$GNP) #sample mean  
alpha<-0.05 #significance level  
#Calculate the critical value at 0.05 significance level  
zc<-qnorm(alpha); zc

## [1] -1.645

#Calculate the test statistic Z0:  
z4<-(xbar4-mu4)/(sigma4/sqrt(n)); z4

## [1] -0.1674

#Use our test.critical function to give the decision  
testmean.cv(zc,z4,mu4,alpha,alt="less",theta="mu")

## [1] "H0: mu = 3000000000 is not rejected at 0.05 significance level"

###We want to test the null hypothesis H0: mu=3000000000 against the less-than one sided alternative using p-value approach  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(mean.x=xbar4, sigma.x=sigma4, n.x=n, alt='less', mu=mu4,conf.level=0.95)

##   
## One-sample z-Test  
##   
## data: Summarized x  
## z = -0.17, p-value = 0.4  
## alternative hypothesis: true mean is less than 3000000000  
## 95 percent confidence interval:  
## NA 4015558347  
## sample estimates:  
## mean of x   
## 2884909571

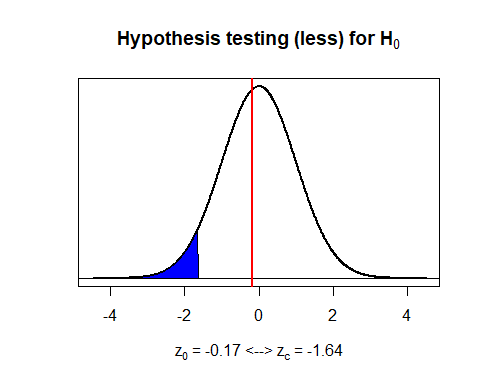
#For manual calculation, calculate the p-value  
pval4<-pnorm(z4); pval4

## [1] 0.4335

#Use our test.pval function to give the decision  
test.pval(pval4,mu4,alpha,theta="mu")

## [1] "H0: mu = 3000000000 is not rejected at 0.05 significance level"

#Use our plotmean.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z4,zc,n,type='z',alt='less')



"TEST FOR GNP"

## [1] "TEST FOR GNP"

#Define true parameter value and delta, and calculate power of the test with pwr.norm.test function:   
mu05 <- 2900000000  
delta4 <- mu05-mu4; delta4

## [1] -100000000

pwrNorm4 <- pwr.norm.test(delta4/sigma4,n,alpha,power=NULL,alt='less'); pwrNorm4

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = -0.02245  
## n = 42  
## sig.level = 0.05  
## power = 0.06689  
## alternative = less

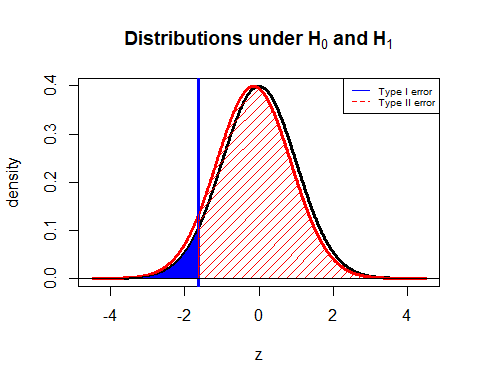
beta4 <- 1-pwrNorm4$power; beta4

## [1] 0.9331

power4 <- 1-beta4; power4

## [1] 0.06689

plotmean.beta(sigma4,zc,delta4,n,type='z',alt='less')

  
…

#### 3.2.2.2. Hypothesis Tests on the Population Variance of a Single Sample

Chi-squared test is used to conduct Hypothesis Tests. It is a method used to test whether the expected frequencies are actually the same as the expected frequency. …

#Write your codes here. Their results will appear after knitting the markdown file.  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

#In order to test the null hypothesis H0: sigma^2=450000000000000000 against the one-sided (greater) alternative using critical value approach  
n<-42 #sample size  
var7<-var(data\_22$CE);var7 #sample variance

## [1] 432804216879521536

sigmasq7<-450000000000000000 #hypothesized value  
alpha<-0.05 #significance level  
  
#The critical values at 0.05 significance level was calculated  
chisqc7<-qchisq(1-alpha,df=n-1); chisqc7

## [1] 56.94

#Also,the critical value by setting lower=FALSE is calculated becaues we the upper tail is needed.  
chisqc7<-qchisq(alpha,df=n-1,lower=F); chisqc7

## [1] 56.94

#Then,the test statistic was calculated:  
chisq7<-(n-1)\*var7/sigmasq7; chisq7

## [1] 39.43

#The test statistic with the critical values was compared and then a decision was made.  
testvar.cv(cvup=chisqc7,cvlo=NULL,chisq7,sigmasq7,alpha,alt="g")

## [1] "H0: sigma^2 = 450000000000000000 is not rejected at 0.05 significance level"

# The p-value approach was done in order to test the null hypothesis H0: sigma^2=450000000000000000 against the one-sided (greater) alternative  
pval7<-1-pchisq(chisq7,df=n-1); pval7

## [1] 0.5404

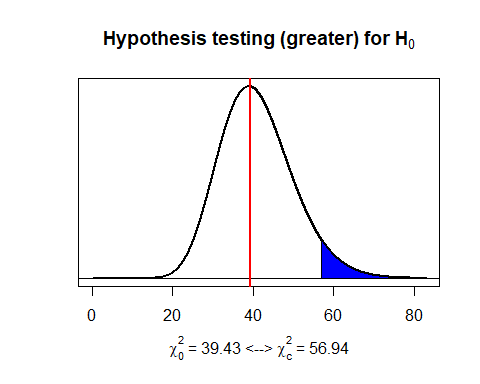
pval7<-pchisq(chisq7,df=n-1,lower=F); pval7

## [1] 0.5404

#P-value with the significance level was compared and then a decision was made.  
test.pval(pval7,sigmasq7,alpha,theta="sigma^2")

## [1] "H0: sigma^2 = 450000000000000000 is not rejected at 0.05 significance level"

# plotvar.hypothesis function was used in order to plot the hypothesis test.  
plotvar.hypothesis(chisq7,alpha,n-1,type='chisq',alt='g')



"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

#In order to test the null hypothesis H0: sigma^2=350000000000000000 against the one-sided (greater) alternative using critical value approach  
n<-42 #sample size  
var8<-var(data\_22$PE);var8 #sample variance

## [1] 307348250641565056

sigmasq8<-350000000000000000 #hypothesized value  
alpha<-0.05 #significance level  
  
#The critical values at 0.05 significance level was calculated  
chisqc8<-qchisq(1-alpha,df=n-1); chisqc8

## [1] 56.94

#Also,the critical value by setting lower=FALSE is calculated becaues we the upper tail is needed.  
chisqc8<-qchisq(alpha,df=n-1,lower=F); chisqc8

## [1] 56.94

#Then,the test statistic was calculated:  
chisq8<-(n-1)\*var8/sigmasq8; chisq8

## [1] 36

#The test statistic with the critical values was compared and then a decision was made.  
testvar.cv(cvup=chisqc8,cvlo=NULL,chisq8,sigmasq8,alpha,alt="g")

## [1] "H0: sigma^2 = 350000000000000000 is not rejected at 0.05 significance level"

# The p-value approach was done in order to test the null hypothesis H0: sigma^2=450000000000000000 against the one-sided (greater) alternative  
pval8<-1-pchisq(chisq8,df=n-1); pval8

## [1] 0.692

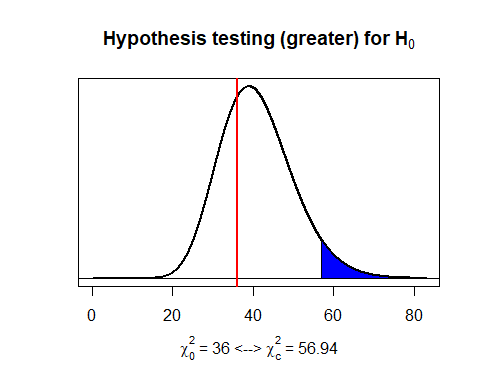
pval8<-pchisq(chisq8,df=n-1,lower=F); pval8

## [1] 0.692

#P-value with the significance level was compared and then a decision was made.  
test.pval(pval8,sigmasq8,alpha,theta="sigma^2")

## [1] "H0: sigma^2 = 350000000000000000 is not rejected at 0.05 significance level"

# plotvar.hypothesis function was used in order to plot the hypothesis test.  
plotvar.hypothesis(chisq8,alpha,n-1,type='chisq',alt='g')



"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

#In order to test the null hypothesis H0: sigma^2=12000000000000000 against the one-sided (greater) alternative using critical value approach  
n<-42 #sample size  
var9<-var(data\_22$OCE);var9 #sample variance

## [1] 10756865022613600

sigmasq9<-12000000000000000 #hypothesized value  
alpha<-0.05 #significance level  
  
#The critical values at 0.05 significance level was calculated  
chisqc9<-qchisq(1-alpha,df=n-1); chisqc9

## [1] 56.94

#Also,the critical value by setting lower=FALSE is calculated becaues we the upper tail is needed.  
chisqc9<-qchisq(alpha,df=n-1,lower=F); chisqc9

## [1] 56.94

#Then,the test statistic was calculated:  
chisq9<-(n-1)\*var9/sigmasq9; chisq9

## [1] 36.75

#The test statistic with the critical values was compared and then a decision was made.  
testvar.cv(cvup=chisqc9,cvlo=NULL,chisq9,sigmasq9,alpha,alt="g")

## [1] "H0: sigma^2 = 12000000000000000 is not rejected at 0.05 significance level"

# The p-value approach was done in order to test the null hypothesis H0: sigma^2=12000000000000000 against the one-sided (greater) alternative  
pval9<-1-pchisq(chisq9,df=n-1); pval9

## [1] 0.6598

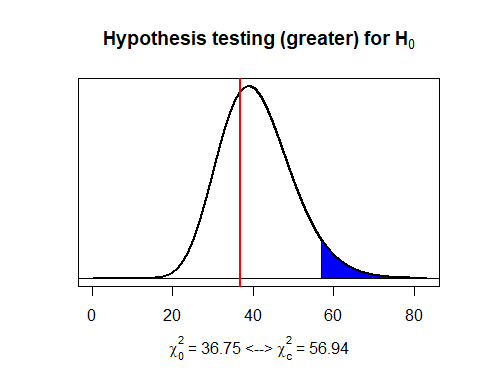
pval9<-pchisq(chisq9,df=n-1,lower=F); pval9

## [1] 0.6598

#P-value with the significance level was compared and then a decision was made.  
test.pval(pval9,sigmasq9,alpha,theta="sigma^2")

## [1] "H0: sigma^2 = 12000000000000000 is not rejected at 0.05 significance level"

# plotvar.hypothesis function was used in order to plot the hypothesis test.  
plotvar.hypothesis(chisq9,alpha,n-1,type='chisq',alt='g')



"FOR DEFENCE"

## [1] "FOR DEFENCE"

#In order to test the null hypothesis H0: sigma^2=8500000000000000 against the one-sided (greater) alternative using critical value approach  
n<-42 #sample size  
var10<-var(data\_22$D);var10 #sample variance

## [1] 8362922193236486

sigmasq10<-8500000000000000 #hypothesized value  
alpha<-0.05 #significance level  
  
#The critical values at 0.05 significance level was calculated  
chisqc10<-qchisq(1-alpha,df=n-1); chisqc10

## [1] 56.94

#Also,the critical value by setting lower=FALSE is calculated becaues we the upper tail is needed.  
chisqc10<-qchisq(alpha,df=n-1,lower=F); chisqc10

## [1] 56.94

#Then,the test statistic was calculated:  
chisq10<-(n-1)\*var10/sigmasq10; chisq10

## [1] 40.34

#The test statistic with the critical values was compared and then a decision was made.  
testvar.cv(cvup=chisqc10,cvlo=NULL,chisq10,sigmasq10,alpha,alt="g")

## [1] "H0: sigma^2 = 8500000000000000 is not rejected at 0.05 significance level"

# The p-value approach was done in order to test the null hypothesis H0: sigma^2=8500000000000000 against the one-sided (greater) alternative  
pval10<-1-pchisq(chisq10,df=n-1); pval10

## [1] 0.4998

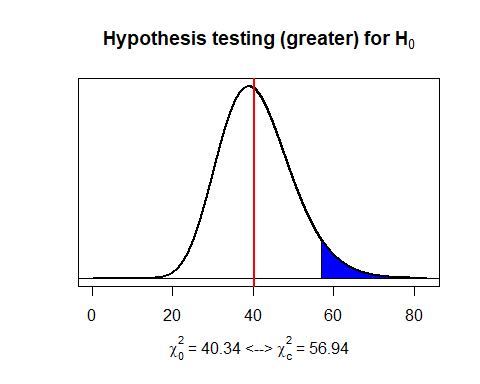
pval10<-pchisq(chisq10,df=n-1,lower=F); pval10

## [1] 0.4998

#P-value with the significance level was compared and then a decision was made.  
test.pval(pval10,sigmasq10,alpha,theta="sigma^2")

## [1] "H0: sigma^2 = 8500000000000000 is not rejected at 0.05 significance level"

# plotvar.hypothesis function was used in order to plot the hypothesis test.  
plotvar.hypothesis(chisq10,alpha,n-1,type='chisq',alt='g')



"FOR GNP"

## [1] "FOR GNP"

#In order to test the null hypothesis H0: sigma^2=21000000000000000000 against the one-sided (greater) alternative using critical value approach  
n<-42 #sample size  
var11<-var(data\_22$GNP);var11 #sample variance

## [1] 19844959216553959424

sigmasq11<-21000000000000000000 #hypothesized value  
alpha<-0.05 #significance level  
  
#The critical values at 0.05 significance level was calculated  
chisqc11<-qchisq(1-alpha,df=n-1); chisqc11

## [1] 56.94

#Also,the critical value by setting lower=FALSE is calculated becaues we the upper tailis needed.  
chisqc11<-qchisq(alpha,df=n-1,lower=F); chisqc11

## [1] 56.94

#Then,the test statistic was calculated:  
chisq11<-(n-1)\*var11/sigmasq11; chisq11

## [1] 38.74

#The test statistic with the critical values was compared and then a decision was made.  
testvar.cv(cvup=chisqc10,cvlo=NULL,chisq11,sigmasq11,alpha,alt="g")

## [1] "H0: sigma^2 = 21000000000000000000 is not rejected at 0.05 significance level"

# The p-value approach was done in order to test the null hypothesis H0: sigma^2=21000000000000000000 against the one-sided (greater) alternative  
pval11<-1-pchisq(chisq11,df=n-1); pval11

## [1] 0.5713

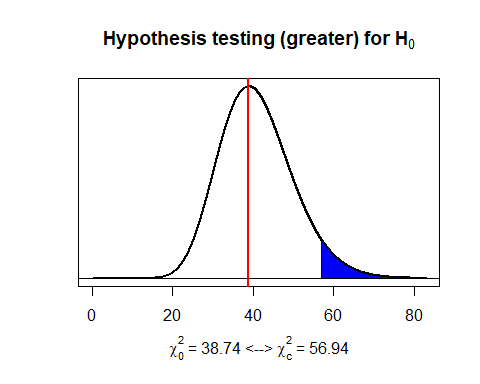
pval11<-pchisq(chisq11,df=n-1,lower=F); pval11

## [1] 0.5713

#P-value with the significance level was compared and then a decision was made.  
test.pval(pval11,sigmasq11,alpha,theta="sigma^2")

## [1] "H0: sigma^2 = 21000000000000000000 is not rejected at 0.05 significance level"

# plotvar.hypothesis function was used in order to plot the hypothesis test.  
plotvar.hypothesis(chisq11,alpha,n-1,type='chisq',alt='g')

  
…

#### 3.2.2.3. Hypothesis Tests on the Difference of Population Means/Proportions of Two Samples

Difference between population means is determined, population standard deviation is unknown and n>=30 so z test is applied for hypothesis tests. …

#Write your codes here. Their results will appear after knitting the markdown file.  
  
"FOR PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES"

#We want to test the null hypothesis H0: mu1-mu2=0 against the greater than one-sided alternative using critical value approach  
#Define problem data  
n1<-42 #sample size from population 1  
n2<-42 #sample size from population 2  
delta0<-0 #hypothesized value  
sigma1<-sd(data\_22$PE) #sample 1 standard deviation  
sigma2<-sd(data\_22$OCE) #sample 2 standard deviation  
xbar18<- mean(data\_22$PE) #sample mean from population 1  
xbar19<-mean(data\_22$OCE) #sample mean from population 2  
alpha<-0.05 #significance level  
  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(xbar18, sigma1, n1, xbar19, sigma2, n2, alt='g', conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 3.7, p-value = 0.0001  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## 177928290 NA  
## sample estimates:  
## mean of x mean of y   
## 394109263 73032098

#For manual calculation, calculate the standard error of the difference in means  
sed<-sqrt(sigma1^2/n1+sigma2^2/n2);sed

## [1] 87028336

#Calculate the critical value at 0.05 significance level  
zc<-qnorm(1-alpha); zc

## [1] 1.645

#Calculate the test statistic Z0 and use our testmean.cv function to give the decision:  
z0<-(xbar18-xbar19-delta0)/sed; z0

## [1] 3.689

testmean.cv(zc,z0,delta0,alpha,alt="g",theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is rejected at 0.05 significance level"

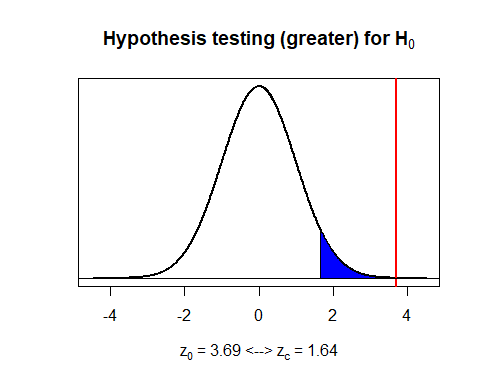
#Calculate the p-value and use our test.pval function to give the decision:  
pval<-(1-pnorm(z0)); pval

## [1] 0.0001124

test.pval(pval,delta0,alpha,theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is rejected at 0.05 significance level"

#Use the plot.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z0,zc,type='z',alt='g')



#Define true difference in population means, and compute the power of the test with pwr.norm.test function:  
delta<-(xbar18-xbar19)  
d<-(delta-delta0)/sqrt(sigma1^2+sigma2^2)  
pwr.norm.test(d,n1,0.05,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.5693  
## n = 42  
## sig.level = 0.05  
## power = 0.9795  
## alternative = greater

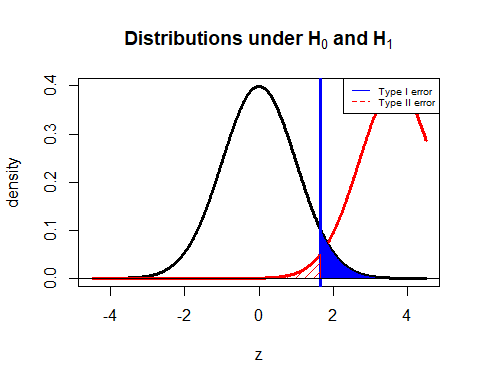
#We can also compute the type II error and power of the test manually:  
beta <- pnorm(zc-((delta-delta0)/sed)); beta

## [1] 0.02045

power<-1-beta; power

## [1] 0.9795

#We can use our plotbeta function with the required parameters:  
plotmean.beta(sed,zc,delta,n=1,type='z',alt='g')



#In the second part, we need to determine the required sample size  
#Again, we can use the pwr.norm.test function as seen below:  
pwr.norm.test(d,n=NULL,0.05,power=0.90,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.5693  
## n = 26.43  
## sig.level = 0.05  
## power = 0.9  
## alternative = greater

#We can calculate the required sample size also manually.   
#The parameters beta and zbeta are determined.  
beta2 <- 0.10  
zbeta <-qnorm(1-beta2); zbeta

## [1] 1.282

#The required sample size is calculated.  
nreq<-((zc+zbeta)/d)^2; nreq

## [1] 26.43

ceiling(nreq)

## [1] 27

"FOR PERSONNEL EXPENDITURES AND DEFENCE"

## [1] "FOR PERSONNEL EXPENDITURES AND DEFENCE"

#We want to test the null hypothesis H0: mu1-mu2=0 against the greater than one-sided alternative using critical value approach  
#Define problem data  
n1<-42 #sample size from population 1  
n2<-42 #sample size from population 2  
delta0<-0 #hypothesized value  
sigma1<-sd(data\_22$PE) #sample 1 standard deviation  
sigma2<-sd(data\_22$D) #sample 2 standard deviation  
xbar20<- mean(data\_22$PE) #sample mean from population 1  
xbar21<-mean(data\_22$D) #sample mean from population 2  
alpha<-0.05 #significance level  
  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(xbar20, sigma1, n1, xbar21, sigma2, n2, alt='g', conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 3.8, p-value = 0.00009  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## 182691294 NA  
## sample estimates:  
## mean of x mean of y   
## 394109263 68808754

#For manual calculation, calculate the standard error of the difference in means  
sed<-sqrt(sigma1^2/n1+sigma2^2/n2);sed

## [1] 86700246

#Calculate the critical value at 0.05 significance level  
zc<-qnorm(1-alpha); zc

## [1] 1.645

#Calculate the test statistic Z0 and use our testmean.cv function to give the decision:  
z0<-(xbar20-xbar21-delta0)/sed; z0

## [1] 3.752

testmean.cv(zc,z0,delta0,alpha,alt="g",theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is rejected at 0.05 significance level"

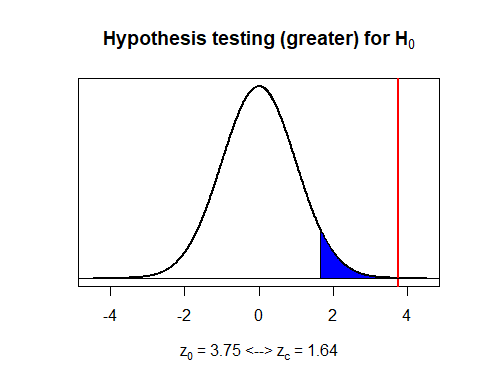
#Calculate the p-value and use our test.pval function to give the decision:  
pval<-(1-pnorm(z0)); pval

## [1] 0.00008771

test.pval(pval,delta0,alpha,theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is rejected at 0.05 significance level"

#Use the plot.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z0,zc,type='z',alt='g')



#Define true difference in population means, and compute the power of the test with pwr.norm.test function:  
delta<-(xbar20-xbar21)  
d<-(delta-delta0)/sqrt(sigma1^2+sigma2^2)  
pwr.norm.test(d,n1,0.05,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.5789  
## n = 42  
## sig.level = 0.05  
## power = 0.9824  
## alternative = greater

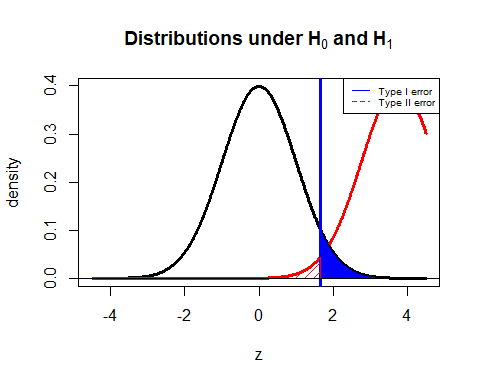
#We can also compute the type II error and power of the test manually:  
beta <- pnorm(zc-((delta-delta0)/sed)); beta

## [1] 0.01755

power<-1-beta; power

## [1] 0.9824

#We can use our plotbeta function with the required parameters:  
plotmean.beta(sed,zc,delta,n=1,type='z',alt='g')



#In the second part, we need to determine the required sample size  
#Again, we can use the pwr.norm.test function as seen below:  
pwr.norm.test(d,n=NULL,0.05,power=0.90,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.5789  
## n = 25.55  
## sig.level = 0.05  
## power = 0.9  
## alternative = greater

#We can calculate the required sample size also manually.   
#The parameters beta and zbeta are determined.  
beta2 <- 0.10  
zbeta <-qnorm(1-beta2); zbeta

## [1] 1.282

#The required sample size is calculated.  
nreq<-((zc+zbeta)/d)^2; nreq

## [1] 25.55

ceiling(nreq)

## [1] 26

"FOR OTHER CURRENT EXPENDITURES AND DEFENCE"

## [1] "FOR OTHER CURRENT EXPENDITURES AND DEFENCE"

#We want to test the null hypothesis H0: mu1-mu2=0 against the greater than one-sided alternative using critical value approach  
#Define problem data  
n1<-42 #sample size from population 1  
n2<-42 #sample size from population 2  
delta0<-0 #hypothesized value  
sigma1<-sd(data\_22$OCE) #sample 1 standard deviation  
sigma2<-sd(data\_22$D) #sample 2 standard deviation  
xbar22<- mean(data\_22$OCE) #sample mean from population 1  
xbar23<-mean(data\_22$D) #sample mean from population 2  
alpha<-0.05 #significance level  
  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(xbar22, sigma1, n1, xbar23, sigma2, n2, alt='g', conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 0.2, p-value = 0.4  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## -30871567 NA  
## sample estimates:  
## mean of x mean of y   
## 73032098 68808754

#For manual calculation, calculate the standard error of the difference in means  
sed<-sqrt(sigma1^2/n1+sigma2^2/n2);sed

## [1] 21336191

#Calculate the critical value at 0.05 significance level  
zc<-qnorm(1-alpha); zc

## [1] 1.645

#Calculate the test statistic Z0 and use our testmean.cv function to give the decision:  
z0<-(xbar22-xbar23-delta0)/sed; z0

## [1] 0.1979

testmean.cv(zc,z0,delta0,alpha,alt="g",theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is not rejected at 0.05 significance level"

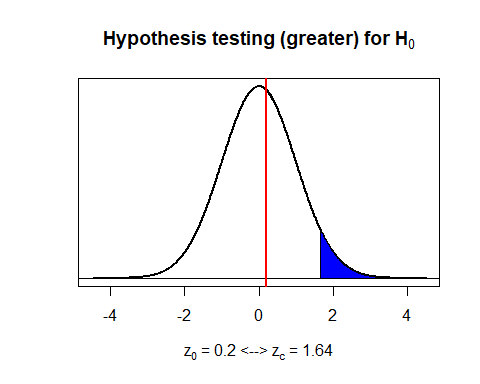
#Calculate the p-value and use our test.pval function to give the decision:  
pval<-(1-pnorm(z0)); pval

## [1] 0.4215

test.pval(pval,delta0,alpha,theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is not rejected at 0.05 significance level"

#Use the plot.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z0,zc,type='z',alt='g')



#Define true difference in population means, and compute the power of the test with pwr.norm.test function:  
delta<-(xbar22-xbar23)  
d<-(delta-delta0)/sqrt(sigma1^2+sigma2^2)  
pwr.norm.test(d,n1,0.05,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.03054  
## n = 42  
## sig.level = 0.05  
## power = 0.07396  
## alternative = greater

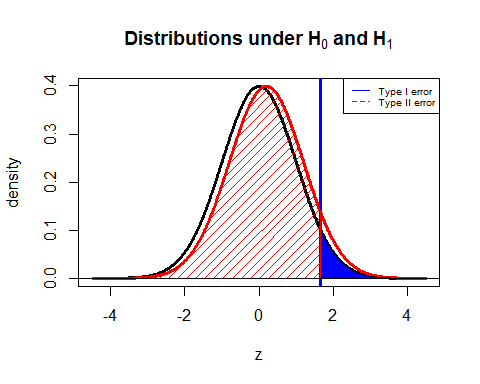
#We can also compute the type II error and power of the test manually:  
beta <- pnorm(zc-((delta-delta0)/sed)); beta

## [1] 0.926

power<-1-beta; power

## [1] 0.07396

#We can use our plotbeta function with the required parameters:  
plotmean.beta(sed,zc,delta,n=1,type='z',alt='g')



#In the second part, we need to determine the required sample size  
#Again, we can use the pwr.norm.test function as seen below:  
pwr.norm.test(d,n=NULL,0.05,power=0.90,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.03054  
## n = 9180  
## sig.level = 0.05  
## power = 0.9  
## alternative = greater

#We can calculate the required sample size also manually.   
#The parameters beta and zbeta are determined.  
beta2 <- 0.10  
zbeta <-qnorm(1-beta2); zbeta

## [1] 1.282

#The required sample size is calculated.  
nreq<-((zc+zbeta)/d)^2; nreq

## [1] 9180

ceiling(nreq)

## [1] 9180

"FOR CURRENT EXPENDITURES AND DEFENCE"

## [1] "FOR CURRENT EXPENDITURES AND DEFENCE"

#We want to test the null hypothesis H0: mu1-mu2=0 against the greater than one-sided alternative using critical value approach  
#Define problem data  
n1<-42 #sample size from population 1  
n2<-42 #sample size from population 2  
delta0<-0 #hypothesized value  
sigma1<-sd(data\_22$CE) #sample 1 standard deviation  
sigma2<-sd(data\_22$D) #sample 2 standard deviation  
xbar24<- mean(data\_22$CE) #sample mean from population 1  
xbar25<-mean(data\_22$D) #sample mean from population 2  
alpha<-0.05 #significance level  
  
#We can use the zsum.test() function from the BSDA package to test the null hypothesis.  
zsum.test(xbar24, sigma1, n1, xbar25, sigma2, n2, alt='g', conf.level=0.95)

##   
## Two-sample z-Test  
##   
## data: Summarized x and y  
## z = 3.9, p-value = 0.00005  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## 229753328 NA  
## sample estimates:  
## mean of x mean of y   
## 467141360 68808754

#For manual calculation, calculate the standard error of the difference in means  
sed<-sqrt(sigma1^2/n1+sigma2^2/n2);sed

## [1] 102488924

#Calculate the critical value at 0.05 significance level  
zc<-qnorm(1-alpha); zc

## [1] 1.645

#Calculate the test statistic Z0 and use our testmean.cv function to give the decision:  
z0<-(xbar24-xbar25-delta0)/sed; z0

## [1] 3.887

testmean.cv(zc,z0,delta0,alpha,alt="g",theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is rejected at 0.05 significance level"

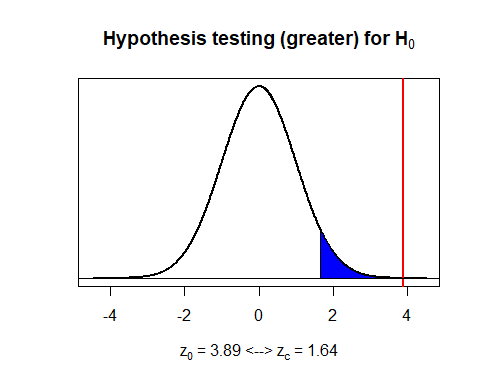
#Calculate the p-value and use our test.pval function to give the decision:  
pval<-(1-pnorm(z0)); pval

## [1] 0.00005083

test.pval(pval,delta0,alpha,theta="mu1-mu2")

## [1] "H0: mu1-mu2 = 0 is rejected at 0.05 significance level"

#Use the plot.hypothesis function to plot the hypothesis test  
plotmean.hypothesis(z0,zc,type='z',alt='g')



#Define true difference in population means, and compute the power of the test with pwr.norm.test function:  
delta<-(xbar24-xbar25)  
d<-(delta-delta0)/sqrt(sigma1^2+sigma2^2)  
pwr.norm.test(d,n1,0.05,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.5997  
## n = 42  
## sig.level = 0.05  
## power = 0.9875  
## alternative = greater

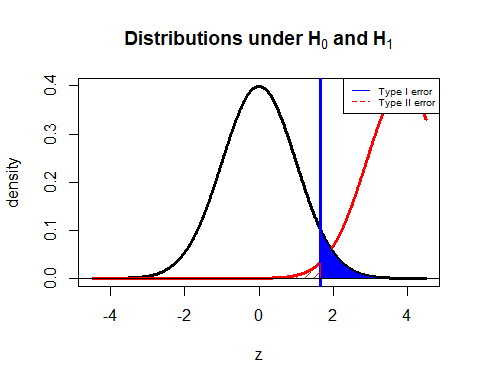
#We can also compute the type II error and power of the test manually:  
beta <- pnorm(zc-((delta-delta0)/sed)); beta

## [1] 0.01249

power<-1-beta; power

## [1] 0.9875

#We can use our plotbeta function with the required parameters:  
plotmean.beta(sed,zc,delta,n=1,type='z',alt='g')



#In the second part, we need to determine the required sample size  
#Again, we can use the pwr.norm.test function as seen below:  
pwr.norm.test(d,n=NULL,0.05,power=0.90,alt='g')

##   
## Mean power calculation for normal distribution with known variance   
##   
## d = 0.5997  
## n = 23.81  
## sig.level = 0.05  
## power = 0.9  
## alternative = greater

#We can calculate the required sample size also manually.   
#The parameters beta and zbeta are determined.  
beta2 <- 0.10  
zbeta <-qnorm(1-beta2); zbeta

## [1] 1.282

#The required sample size is calculated.  
nreq<-((zc+zbeta)/d)^2; nreq

## [1] 23.81

ceiling(nreq)

## [1] 24

…

#### 3.2.2.4. Hypothesis Tests on the Ratio of Population Variances of Two Samples

F test was applied since two populations are independent and populations perform normal distribution for testing on the ratio of population variances of two samples. …

#Write your codes here. Their results will appear after knitting the markdown file.  
  
"FOR PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES"

#We want to test the null hypothesis H0: sigma1^2=sigma2^2 against the one-sided (greater) alternative.  
#Define problem data.  
var1<-var(data\_22$PE)  
var2<-var(data\_22$OCE)  
n1<-42  
n2<-42  
alpha<-0.05  
  
#Find the upper and lower critical values  
fcup <- qf(alpha/2,n1-1,n2-1,lower=F); fcup

## [1] 1.86

fclo <- qf(1-alpha/2,n1-1,n2-1,lower=F); fclo

## [1] 0.5375

#Calculate the test statistic  
f0 <- (var1/var2); f0

## [1] 28.57

#Give a decision with our testvar.cv function  
testvar.cv(fclo,fcup,f0,theta0=1,alpha=0.05,alt='g')

## [1] "H0: sigma^2 = 1 is rejected at 0.05 significance level"

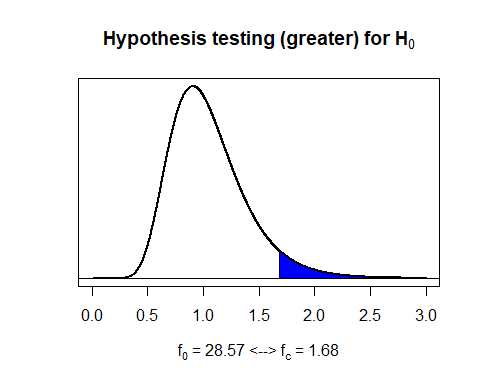
#Calculate the p-value and give a decision with our test.pval function  
pval <- pf(f0,n1-1,n2-1); pval

## [1] 1

#Alternatively, we can use var.test() built-in function.  
var.test(data\_22$PE,data\_22$OCE,alt='g')

##   
## F test to compare two variances  
##   
## data: data\_22$PE and data\_22$OCE  
## F = 29, num df = 41, denom df = 41, p-value <0.0000000000000002  
## alternative hypothesis: true ratio of variances is greater than 1  
## 95 percent confidence interval:  
## 16.99 Inf  
## sample estimates:  
## ratio of variances   
## 28.57

#Use our plotvar.hypothesis function to plot the hypothesis test  
plotvar.hypothesis(f0,alpha,df1=n1-1,df2=n2-1,type='f',alt='g')



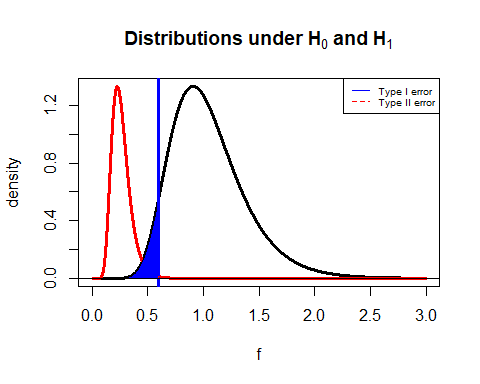
"FOR TEST ON PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES "

## [1] "FOR TEST ON PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES "

#Define scale parameter and compute the power of the test with our power.var.test() function:  
lambda <- 0.5  
power.var.test(n1,lambda,sig.level=0.05,power=NULL,type="t",alt="o")

##   
## Two-sample F Variance test power calculation   
##   
## n = 42  
## lambda = 0.5  
## sig.level = 0.05  
## power = 0.9967  
## alternative = one.sided  
##   
## NOTE: n is number in \*each\* group

#We can use our plotvar.beta function with the required parameters:  
plotvar.beta(lambda,alpha,n1-1,n2-1,type='f',alt='o')



"FOR PERSONNEL EXPENDITURES AND DEFENCE"

## [1] "FOR PERSONNEL EXPENDITURES AND DEFENCE"

#We want to test the null hypothesis H0: sigma1^2=sigma2^2 against the one-sided (greater) alternative.  
#Define problem data.  
var1<-var(data\_22$PE)  
var3<-var(data\_22$D)  
n1<-42  
n2<-42  
alpha<-0.05  
  
  
  
#Calculate the test statistic  
f1 <- (var1/var3); f1

## [1] 36.75

#Give a decision with our testvar.cv function  
testvar.cv(fclo,fcup,f1,theta0=1,alpha=0.05,alt='g')

## [1] "H0: sigma^2 = 1 is rejected at 0.05 significance level"

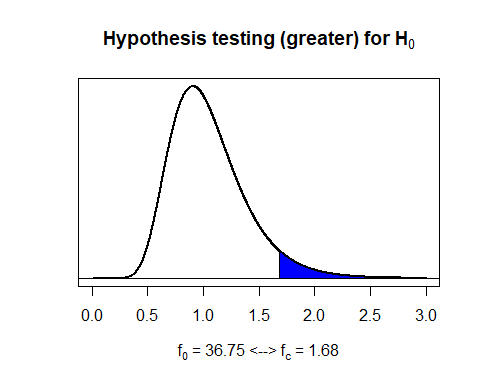
#Calculate the p-value and give a decision with our test.pval function  
pval1 <- pf(f1,n1-1,n2-1); pval1

## [1] 1

#Alternatively,--e can use var.test() built-in function.  
var.test(data\_22$PE,data\_22$D,alt='g')

##   
## F test to compare two variances  
##   
## data: data\_22$PE and data\_22$D  
## F = 37, num df = 41, denom df = 41, p-value <0.0000000000000002  
## alternative hypothesis: true ratio of variances is greater than 1  
## 95 percent confidence interval:  
## 21.85 Inf  
## sample estimates:  
## ratio of variances   
## 36.75

#Use our plotvar.hypothesis function to plot the hypothesis test  
plotvar.hypothesis(f1,alpha,df1=n1-1,df2=n2-1,type='f',alt='g')



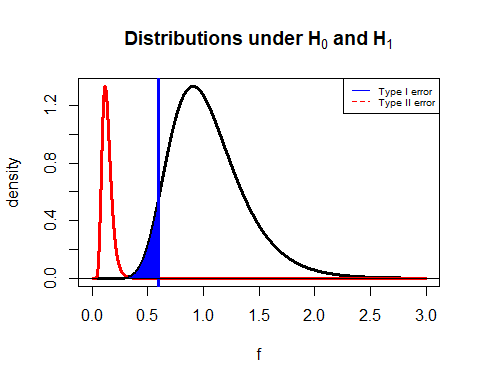
"FOR TEST ON PERSONNEL EXPENDITURES AND DEFENCE "

## [1] "FOR TEST ON PERSONNEL EXPENDITURES AND DEFENCE "

#Define scale parameter and compute the power of the test with our power.var.test() function:  
lambda1 <- 0.36  
power.var.test(n1,lambda1,sig.level=0.05,power=NULL,type="t",alt="o")

##   
## Two-sample F Variance test power calculation   
##   
## n = 42  
## lambda = 0.36  
## sig.level = 0.05  
## power = 1  
## alternative = one.sided  
##   
## NOTE: n is number in \*each\* group

#We can use our plotvar.beta function with the required parameters:  
plotvar.beta(lambda1,alpha,n1-1,n2-1,type='f',alt='o')



"FOR OTHER CURRENT EXPENDITURES AND DEFENCE"

## [1] "FOR OTHER CURRENT EXPENDITURES AND DEFENCE"

#We want to test the null hypothesis H0: sigma1^2=sigma2^2 against the one-sided (greater) alternative.  
#Define problem data.  
var2<-var(data\_22$OCE)  
var3<-var(data\_22$D)  
n1<-42  
n2<-42  
alpha<-0.05  
  
  
  
#Calculate the test statistic  
f2 <- (var2/var3); f2

## [1] 1.286

#Give a decision with our testvar.cv function  
testvar.cv(fclo,fcup,f2,theta0=1,alpha=0.05,alt='g')

## [1] "H0: sigma^2 = 1 is not rejected at 0.05 significance level"

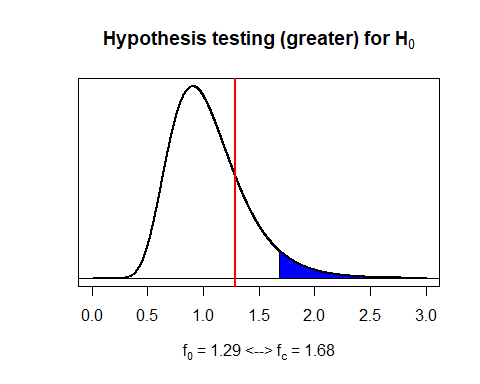
#Calculate the p-value and give a decision with our test.pval function  
pval2 <- pf(f2,n1-1,n2-1); pval2

## [1] 0.7881

#Alternatively, we can use var.test() built-in function.  
var.test(data\_22$OCE,data\_22$D,alt='g')

##   
## F test to compare two variances  
##   
## data: data\_22$OCE and data\_22$D  
## F = 1.3, num df = 41, denom df = 41, p-value = 0.2  
## alternative hypothesis: true ratio of variances is greater than 1  
## 95 percent confidence interval:  
## 0.7649 Inf  
## sample estimates:  
## ratio of variances   
## 1.286

#Use our plotvar.hypothesis function to plot the hypothesis test  
plotvar.hypothesis(f2,alpha,df1=n1-1,df2=n2-1,type='f',alt='g')



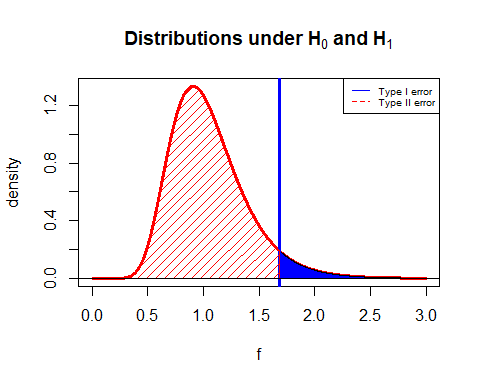
"FOR TEST ON OTHER CURRENT EXPENDITURES AND DEFENCE "

## [1] "FOR TEST ON OTHER CURRENT EXPENDITURES AND DEFENCE "

#Define scale parameter and compute the power of the test with our power.var.test() function:  
lambda2 <- 1  
power.var.test(n1,lambda2,sig.level=0.05,power=NULL,type="t",alt="o")

##   
## Two-sample F Variance test power calculation   
##   
## n = 42  
## lambda = 1  
## sig.level = 0.05  
## power = 0.05  
## alternative = one.sided  
##   
## NOTE: n is number in \*each\* group

#We can use our plotvar.beta function with the required parameters:  
plotvar.beta(lambda2,alpha,n1-1,n2-1,type='f',alt='o')



"FOR DEFENCE AND CURRENT EXPENDITURES "

## [1] "FOR DEFENCE AND CURRENT EXPENDITURES "

#We want to test the null hypothesis H0: sigma1^2=sigma2^2 against the one-sided (less) alternative.  
#Define problem data.  
var3<-var(data\_22$D)  
var4<-var(data\_22$CE)  
n1<-42  
n2<-42  
alpha<-0.05  
  
#Find the critical value  
fc <- qf(1-alpha,n1-1,n2-1,lower=F); fc

## [1] 0.5947

#Calculate the test statistic  
f3 <- (var3/var4); f3

## [1] 0.01932

#Give a decision with our testvar.cv function  
testvar.cv(fc,cvup=NULL,f3,theta0=1,alpha=0.05,alt='l')

## [1] "H0: sigma^2 = 1 is rejected at 0.05 significance level"

#Calculate the p-value and give a decision with our test.pval function  
pval3 <- pf(f3,n1-1,n2-1); pval3

## [1] 0.0000000000000000000000004717

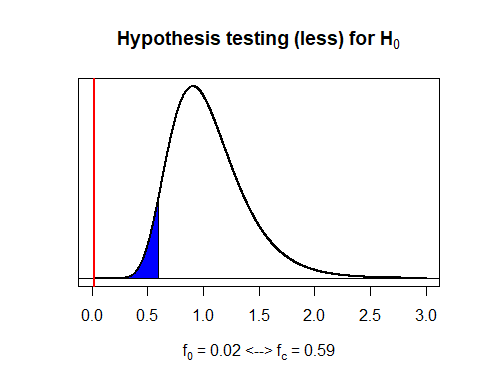
test.pval(pval3,1,alpha,theta='sigma^2')

## [1] "H0: sigma^2 = 1 is rejected at 0.05 significance level"

#Alternatively, we can use var.test() built-in function.  
var.test(data\_22$D,data\_22$CE,alt='l')

##   
## F test to compare two variances  
##   
## data: data\_22$D and data\_22$CE  
## F = 0.019, num df = 41, denom df = 41, p-value <0.0000000000000002  
## alternative hypothesis: true ratio of variances is less than 1  
## 95 percent confidence interval:  
## 0.00000 0.03249  
## sample estimates:  
## ratio of variances   
## 0.01932

#Use our plotvar.hypothesis function to plot the hypothesis test  
plotvar.hypothesis(f3,alpha,df1=n1-1,df2=n2-1,type='f',alt='l')



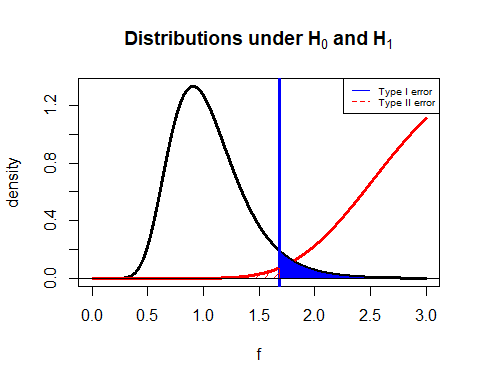
"FOR TEST ON DEFENCE AND CURRENT EXPENDITURES "

## [1] "FOR TEST ON DEFENCE AND CURRENT EXPENDITURES "

#Define scale parameter and compute the power of the test with our power.var.test() function:  
lambda3 <- 2  
power.var.test(n1,lambda3,sig.level=0.05,power=NULL,type="t",alt="o")

##   
## Two-sample F Variance test power calculation   
##   
## n = 42  
## lambda = 2  
## sig.level = 0.05  
## power = 0.9967  
## alternative = one.sided  
##   
## NOTE: n is number in \*each\* group

#We can use our plotvar.beta function with the required parameters:  
plotvar.beta(lambda3,alpha,n1-1,n2-1,type='f',alt='o')

  
…

### 3.2.3. Goodness of Fit Tests and Contingency Analyses

The goodness of fit test is used to determine whether sample data are consistent with a hypothesized distribution. If X2 is greater than critical value, the null hypothesis will be rejected …

#For current expenditures  
  
k <- 7  
xbar <- mean(data\_22$CE)   
sd <- sd(data\_22$CE)  
z <- qnorm(0:k/k); z

## [1] -Inf -1.0676 -0.5659 -0.1800 0.1800 0.5659 1.0676 Inf

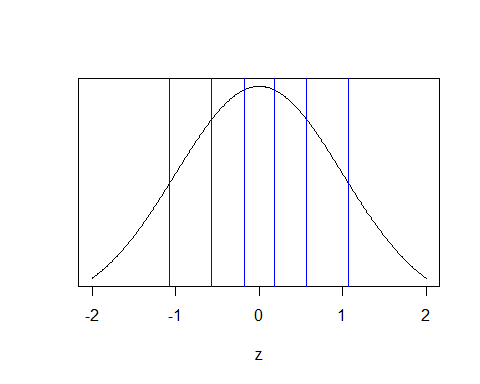
interval <- xbar+z\*sd; interval

## [1] -Inf -235190413 94815757 348715079 585567641 839466964 1169473133  
## [8] Inf

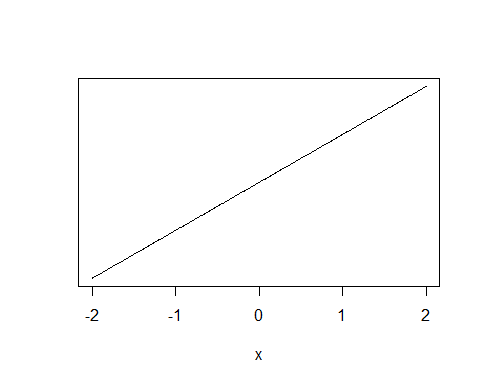
gftCE<-(data\_22$CE)  
gftCE.breaks <- seq(650,2340682000,by=293000000)   
gftCE.cut<- cut(gftCE,gftCE.breaks,right = F)  
gftCE.freq<- table((gftCE.cut))  
gftCE.freq

##   
## [650,2.93e+08) [2.93e+08,5.86e+08) [5.86e+08,8.79e+08) [8.79e+08,1.17e+09)   
## 26 3 1 5   
## [1.17e+09,1.47e+09) [1.47e+09,1.76e+09) [1.76e+09,2.05e+09)   
## 2 3 1

#Plot the probabilities of each cell from standard normal distribution with k=7 equal bins.  
x <- seq(-2,2,0.01)  
plot(x,dnorm(x),type="l",yaxt='n',xlab='z',ylab='')  
abline(v = z, col = "blue")



#Plot the probabilities of each cell from normal distribution with k=7 equal bins.  
plot(x,dnorm(x, xbar, sd),type="l",yaxt='n',ylab='')  
abline(v = interval, col = "blue")



#Observed frequencies of class i are entered into a table and expected frequencies are calculated for pi=1/7.  
cut<-cut(0, interval, right=F)  
voltage.freq<- c(26,3,1,5,2,3,1)  
names(voltage.freq)<-levels(cut)  
voltage.prob<-rep(1/k,k)  
n<-42  
voltage.exp<-n\*voltage.prob  
  
#View the observed and expected frequencies together.  
cbind(Observed=voltage.freq,Expected=voltage.exp)

## Observed Expected  
## [-Inf,-2.35e+08) 26 6  
## [-2.35e+08,9.48e+07) 3 6  
## [9.48e+07,3.49e+08) 1 6  
## [3.49e+08,5.86e+08) 5 6  
## [5.86e+08,8.39e+08) 2 6  
## [8.39e+08,1.17e+09) 3 6  
## [1.17e+09, Inf) 1 6

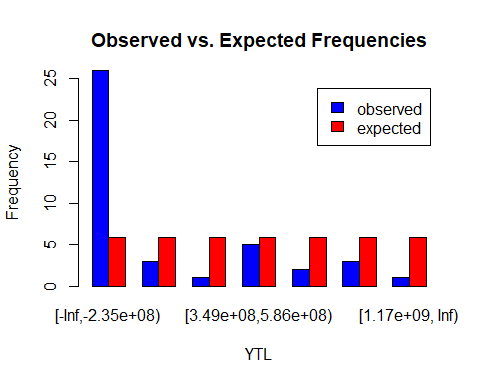
#We do the Goodness of Fit test with 7-2-1=4 degrees of freedom since two parameters of normal dist. is estimated.  
#We need to readjust the degrees of freedom by using the following functions.  
voltage.Xsq<-chisq.test(voltage.freq,p=voltage.prob)  
voltage.Xsq$parameter <- c(df=4)  
voltage.Xsq$p.value <- pchisq(voltage.Xsq$statistic, df=4, lower=F)  
  
#We can view the result of the test, observed and expected frequencies.  
voltage.Xsq

##   
## Chi-squared test for given probabilities  
##   
## data: voltage.freq  
## X-squared = 83, df = 4, p-value <0.0000000000000002

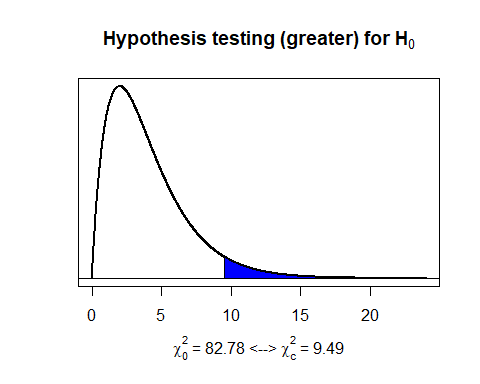
voltage.table<-rbind(observed=voltage.Xsq$observed, expected=voltage.Xsq$expected); t(voltage.table)

## observed expected  
## [-Inf,-2.35e+08) 26 5.857  
## [-2.35e+08,9.48e+07) 3 5.857  
## [9.48e+07,3.49e+08) 1 5.857  
## [3.49e+08,5.86e+08) 5 5.857  
## [5.86e+08,8.39e+08) 2 5.857  
## [8.39e+08,1.17e+09) 3 5.857  
## [1.17e+09, Inf) 1 5.857

barplot(voltage.table,ylim=c(0,25),beside=TRUE,legend=TRUE,main="Observed vs. Expected Frequencies",col=c("blue", "red"),xlab="YTL", ylab="Frequency")



#We can plot the hypothesis testing for goodness of fit using plotvar.hypothesis function that we defined before.  
plotvar.hypothesis(voltage.Xsq$statistic,alpha=0.05,df1=4,type='c',alt='g')



#Write your codes here. Their results will appear after knitting the markdown file.  
  
#For Personnel expenditures  
  
k <- 7  
xbar <- mean(data\_22$PE)   
sd <- sd(data\_22$PE)  
z <- qnorm(0:k/k); z

## [1] -Inf -1.0676 -0.5659 -0.1800 0.1800 0.5659 1.0676 Inf

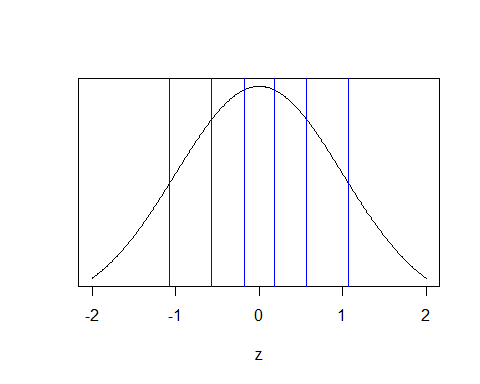
interval <- xbar+z\*sd; interval

## [1] -Inf -197741139 80352906 294312208 493906317 707865619 985959665  
## [8] Inf

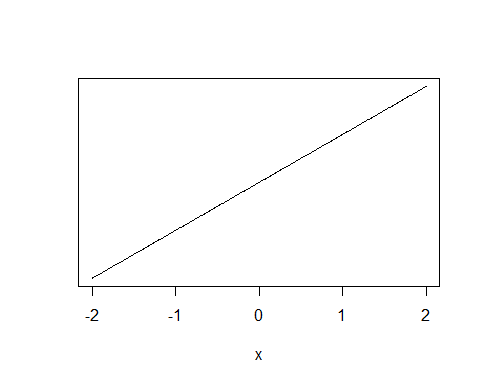
gftPE<-(data\_22$CE)  
gftPE.breaks <- seq(550,1965616054,by=245702000)   
gftPE.cut<- cut(gftPE,gftPE.breaks,right = F)  
gftPE.freq<- table((gftPE.cut))  
gftPE.freq

##   
## [550,2.46e+08) [2.46e+08,4.91e+08) [4.91e+08,7.37e+08) [7.37e+08,9.83e+08)   
## 26 2 1 2   
## [9.83e+08,1.23e+09) [1.23e+09,1.47e+09) [1.47e+09,1.72e+09)   
## 5 2 2

#Plot the probabilities of each cell from standard normal distribution with k=7 equal bins.  
x <- seq(-2,2,0.01)  
plot(x,dnorm(x),type="l",yaxt='n',xlab='z',ylab='')  
abline(v = z, col = "blue")



#Plot the probabilities of each cell from normal distribution with k=7 equal bins.  
plot(x,dnorm(x, xbar, sd),type="l",yaxt='n',ylab='')  
abline(v = interval, col = "blue")



#Observed frequencies of class i are entered into a table and expected frequencies are calculated for pi=1/7.  
cut<-cut(0, interval, right=F)  
voltage.freq<- c(26,2,1,2,5,2,2)  
names(voltage.freq)<-levels(cut)  
voltage.prob<-rep(1/k,k)  
n<-42  
voltage.exp<-n\*voltage.prob  
  
#View the observed and expected frequencies together.  
cbind(Observed=voltage.freq,Expected=voltage.exp)

## Observed Expected  
## [-Inf,-1.98e+08) 26 6  
## [-1.98e+08,8.04e+07) 2 6  
## [8.04e+07,2.94e+08) 1 6  
## [2.94e+08,4.94e+08) 2 6  
## [4.94e+08,7.08e+08) 5 6  
## [7.08e+08,9.86e+08) 2 6  
## [9.86e+08, Inf) 2 6

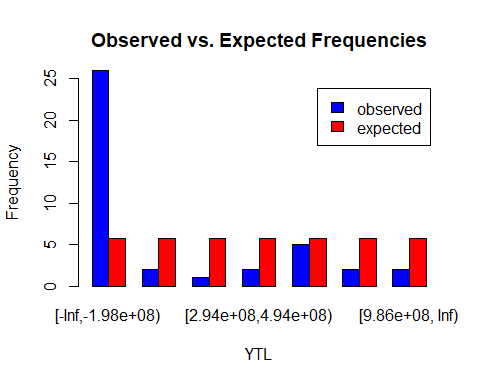
#We do the Goodness of Fit test with 7-2-1=4 degrees of freedom since two parameters of normal dist. is estimated.  
#We need to readjust the degrees of freedom by using the following functions.  
voltage.Xsq<-chisq.test(voltage.freq,p=voltage.prob)  
voltage.Xsq$parameter <- c(df=4)  
voltage.Xsq$p.value <- pchisq(voltage.Xsq$statistic, df=4, lower=F)  
  
#We can view the result of the test, observed and expected frequencies.  
voltage.Xsq

##   
## Chi-squared test for given probabilities  
##   
## data: voltage.freq  
## X-squared = 86, df = 4, p-value <0.0000000000000002

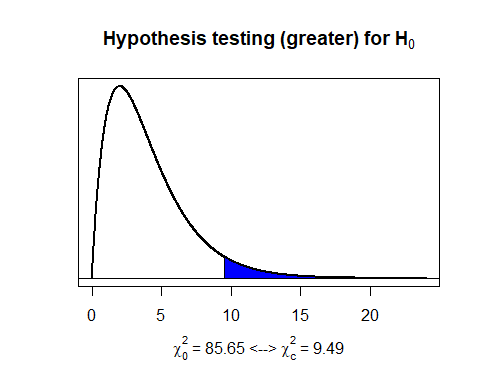
voltage.table<-rbind(observed=voltage.Xsq$observed, expected=voltage.Xsq$expected); t(voltage.table)

## observed expected  
## [-Inf,-1.98e+08) 26 5.714  
## [-1.98e+08,8.04e+07) 2 5.714  
## [8.04e+07,2.94e+08) 1 5.714  
## [2.94e+08,4.94e+08) 2 5.714  
## [4.94e+08,7.08e+08) 5 5.714  
## [7.08e+08,9.86e+08) 2 5.714  
## [9.86e+08, Inf) 2 5.714

barplot(voltage.table,ylim=c(0,25),beside=TRUE,legend=TRUE,main="Observed vs. Expected Frequencies",col=c("blue", "red"),xlab="YTL", ylab="Frequency")



#We can plot the hypothesis testing for goodness of fit using plotvar.hypothesis function that we defined before.  
plotvar.hypothesis(voltage.Xsq$statistic,alpha=0.05,df1=4,type='c',alt='g')



#Write your codes here. Their results will appear after knitting the markdown file.  
  
#For Other Current expenditures  
  
k <- 7  
xbar <- mean(data\_22$OCE)   
sd <- sd(data\_22$OCE)  
z <- qnorm(0:k/k); z

## [1] -Inf -1.0676 -0.5659 -0.1800 0.1800 0.5659 1.0676 Inf

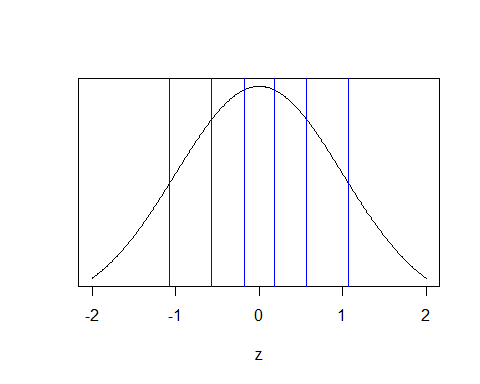
interval <- xbar+z\*sd; interval

## [1] -Inf -37691308 14334542 54362059 91702136 131729654 183755503  
## [8] Inf

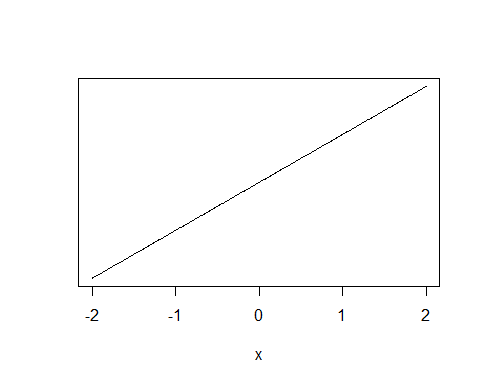
gftOCE<-(data\_22$OCE)  
gftOCE.breaks <- seq(110,375065559,by=53581000)   
gftOCE.cut<- cut(gftOCE,gftOCE.breaks,right = F)  
gftOCE.freq<- table((gftOCE.cut))  
gftOCE.freq

##   
## [110,5.36e+07) [5.36e+07,1.07e+08) [1.07e+08,1.61e+08) [1.61e+08,2.14e+08)   
## 27 2 2 6   
## [2.14e+08,2.68e+08) [2.68e+08,3.21e+08)   
## 3 1

#Plot the probabilities of each cell from standard normal distribution with k=7 equal bins.  
x <- seq(-2,2,0.01)  
plot(x,dnorm(x),type="l",yaxt='n',xlab='z',ylab='')  
abline(v = z, col = "blue")



#Plot the probabilities of each cell from normal distribution with k=7 equal bins.  
plot(x,dnorm(x, xbar, sd),type="l",yaxt='n',ylab='')  
abline(v = interval, col = "blue")



#Observed frequencies of class i are entered into a table and expected frequencies are calculated for pi=1/8.  
cut<-cut(0, interval, right=F)  
voltage.freq<- c(27,2,2,2,8,3,1)  
names(voltage.freq)<-levels(cut)  
voltage.prob<-rep(1/k,k)  
n<-42  
voltage.exp<-n\*voltage.prob  
  
#View the observed and expected frequencies together.  
cbind(Observed=voltage.freq,Expected=voltage.exp)

## Observed Expected  
## [-Inf,-3.77e+07) 27 6  
## [-3.77e+07,1.43e+07) 2 6  
## [1.43e+07,5.44e+07) 2 6  
## [5.44e+07,9.17e+07) 2 6  
## [9.17e+07,1.32e+08) 8 6  
## [1.32e+08,1.84e+08) 3 6  
## [1.84e+08, Inf) 1 6

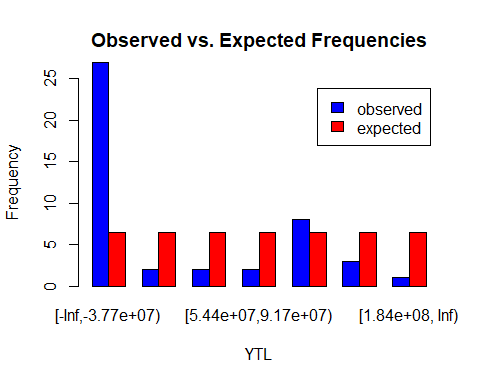
#We do the Goodness of Fit test with 7-2-1=4 degrees of freedom since two parameters of normal dist. is estimated.  
#We need to readjust the degrees of freedom by using the following functions.  
voltage.Xsq<-chisq.test(voltage.freq,p=voltage.prob)  
voltage.Xsq$parameter <- c(df=4)  
voltage.Xsq$p.value <- pchisq(voltage.Xsq$statistic, df=4, lower=F)  
  
#We can view the result of the test, observed and expected frequencies.  
voltage.Xsq

##   
## Chi-squared test for given probabilities  
##   
## data: voltage.freq  
## X-squared = 82, df = 4, p-value <0.0000000000000002

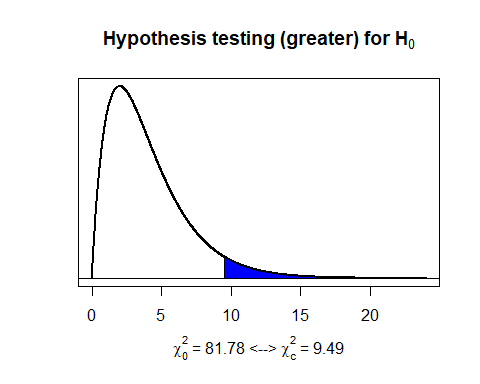
voltage.table<-rbind(observed=voltage.Xsq$observed, expected=voltage.Xsq$expected); t(voltage.table)

## observed expected  
## [-Inf,-3.77e+07) 27 6.429  
## [-3.77e+07,1.43e+07) 2 6.429  
## [1.43e+07,5.44e+07) 2 6.429  
## [5.44e+07,9.17e+07) 2 6.429  
## [9.17e+07,1.32e+08) 8 6.429  
## [1.32e+08,1.84e+08) 3 6.429  
## [1.84e+08, Inf) 1 6.429

barplot(voltage.table,ylim=c(0,25),beside=TRUE,legend=TRUE,main="Observed vs. Expected Frequencies",col=c("blue", "red"),xlab="YTL", ylab="Frequency")



#We can plot the hypothesis testing for goodness of fit using plotvar.hypothesis function that we defined before.  
plotvar.hypothesis(voltage.Xsq$statistic,alpha=0.05,df1=4,type='c',alt='g')



#Write your codes here. Their results will appear after knitting the markdown file.  
  
#For Defence  
  
k <- 7  
xbar <- mean(data\_22$D)   
sd <- sd(data\_22$D)  
z <- qnorm(0:k/k); z

## [1] -Inf -1.0676 -0.5659 -0.1800 0.1800 0.5659 1.0676 Inf

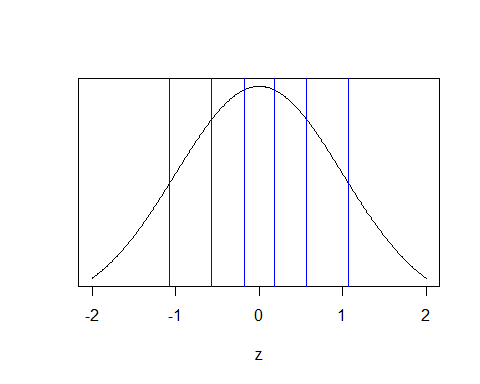
interval <- xbar+z\*sd; interval

## [1] -Inf -28819518 17053292 52346800 85270708 120564216 166437026  
## [8] Inf

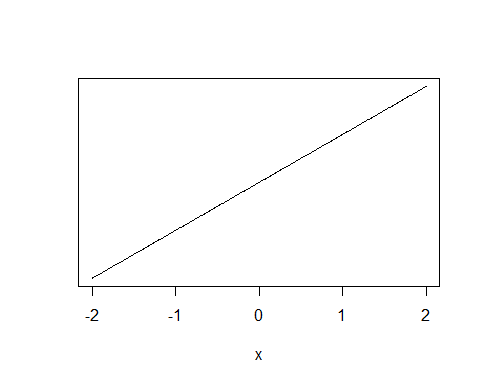
gftD<-(data\_22$D)  
gftD.breaks <- seq(140,170499000,by=21320000)   
gftD.cut<- cut(gftD,gftD.breaks,right = F)  
gftD.freq<- table((gftD.cut))  
gftD.freq

##   
## [140,2.13e+07) [2.13e+07,4.26e+07) [4.26e+07,6.4e+07) [6.4e+07,8.53e+07)   
## 21 3 1 3   
## [8.53e+07,1.07e+08) [1.07e+08,1.28e+08) [1.28e+08,1.49e+08)   
## 0 2 1

#Plot the probabilities of each cell from standard normal distribution with k=7 equal bins.  
x <- seq(-2,2,0.01)  
plot(x,dnorm(x),type="l",yaxt='n',xlab='z',ylab='')  
abline(v = z, col = "blue")



#Plot the probabilities of each cell from normal distribution with k=7 equal bins.  
plot(x,dnorm(x, xbar, sd),type="l",yaxt='n',ylab='')  
abline(v = interval, col = "blue")



#Observed frequencies of class i are entered into a table and expected frequencies are calculated for pi=1/7.  
cut<-cut(0, interval, right=F)  
voltage.freq<- c(21,3,1,3,0,2,1)  
names(voltage.freq)<-levels(cut)  
voltage.prob<-rep(1/k,k)  
n<-42  
voltage.exp<-n\*voltage.prob  
  
#View the observed and expected frequencies together.  
cbind(Observed=voltage.freq,Expected=voltage.exp)

## Observed Expected  
## [-Inf,-2.88e+07) 21 6  
## [-2.88e+07,1.71e+07) 3 6  
## [1.71e+07,5.23e+07) 1 6  
## [5.23e+07,8.53e+07) 3 6  
## [8.53e+07,1.21e+08) 0 6  
## [1.21e+08,1.66e+08) 2 6  
## [1.66e+08, Inf) 1 6

#We do the Goodness of Fit test with 7-2-1=4 degrees of freedom since two parameters of normal dist. is estimated.  
#We need to readjust the degrees of freedom by using the following functions.  
voltage.Xsq<-chisq.test(voltage.freq,p=voltage.prob)

## Warning in chisq.test(voltage.freq, p = voltage.prob): Chi-squared approximation  
## may be incorrect

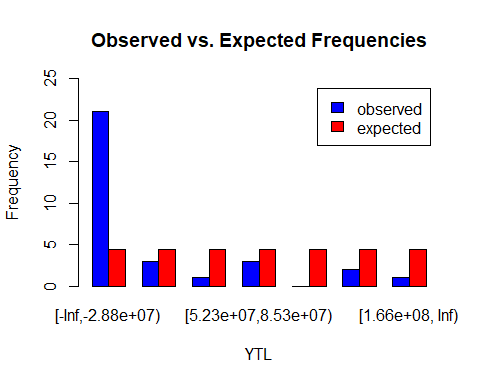
voltage.Xsq$parameter <- c(df=4)  
voltage.Xsq$p.value <- pchisq(voltage.Xsq$statistic, df=4, lower=F)  
  
#We can view the result of the test, observed and expected frequencies.  
voltage.Xsq

##   
## Chi-squared test for given probabilities  
##   
## data: voltage.freq  
## X-squared = 74, df = 4, p-value = 0.000000000000003

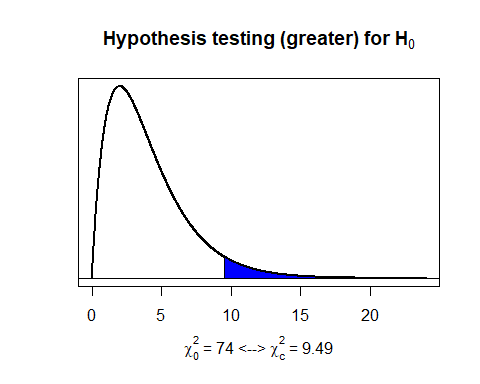
voltage.table<-rbind(observed=voltage.Xsq$observed, expected=voltage.Xsq$expected); t(voltage.table)

## observed expected  
## [-Inf,-2.88e+07) 21 4.429  
## [-2.88e+07,1.71e+07) 3 4.429  
## [1.71e+07,5.23e+07) 1 4.429  
## [5.23e+07,8.53e+07) 3 4.429  
## [8.53e+07,1.21e+08) 0 4.429  
## [1.21e+08,1.66e+08) 2 4.429  
## [1.66e+08, Inf) 1 4.429

barplot(voltage.table,ylim=c(0,25),beside=TRUE,legend=TRUE,main="Observed vs. Expected Frequencies",col=c("blue", "red"),xlab="YTL", ylab="Frequency")



#We can plot the hypothesis testing for goodness of fit using plotvar.hypothesis function that we defined before.  
plotvar.hypothesis(voltage.Xsq$statistic,alpha=0.05,df1=4,type='c',alt='g')



#Write your codes here. Their results will appear after knitting the markdown file.  
  
#For GNP  
  
k <- 7  
xbar <- mean(data\_22$GNP)   
sd <- sd(data\_22$GNP)  
z <- qnorm(0:k/k); z

## [1] -Inf -1.0676 -0.5659 -0.1800 0.1800 0.5659 1.0676 Inf

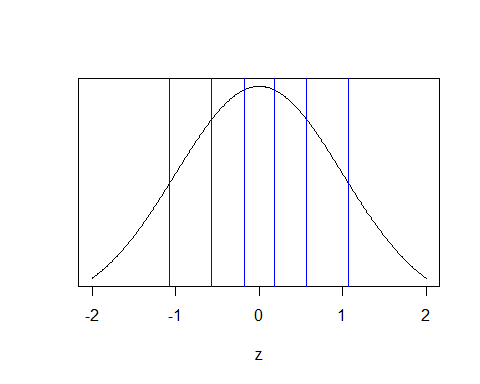
interval <- xbar+z\*sd; interval

## [1] -Inf -1870869590 363738788 2082996201 3686822942 5406080354  
## [7] 7640688732 Inf

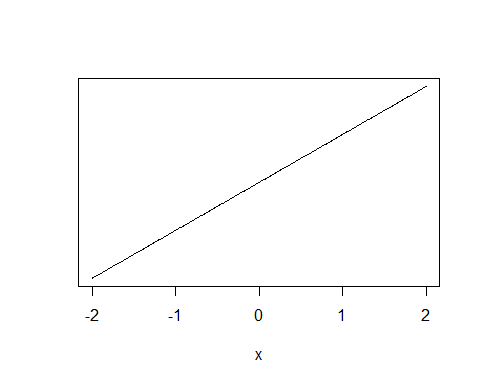
gftGNP<-(data\_22$GNP)  
gftGNP.breaks <- seq(3800,18334799700,by=2291849487)   
gftGNP.cut<- cut(gftD,gftGNP.breaks,right = F)  
gftGNP.freq<- table((gftGNP.cut))  
gftGNP.freq

##   
## [3.8e+03,2.29e+09) [2.29e+09,4.58e+09) [4.58e+09,6.88e+09) [6.88e+09,9.17e+09)   
## 33 0 0 0   
## [9.17e+09,1.15e+10) [1.15e+10,1.38e+10) [1.38e+10,1.6e+10) [1.6e+10,1.83e+10)   
## 0 0 0 0

#Plot the probabilities of each cell from standard normal distribution with k=7 equal bins.  
x <- seq(-2,2,0.01)  
plot(x,dnorm(x),type="l",yaxt='n',xlab='z',ylab='')  
abline(v = z, col = "blue")



#Plot the probabilities of each cell from normal distribution with k=7 equal bins.  
plot(x,dnorm(x, xbar, sd),type="l",yaxt='n',ylab='')  
abline(v = interval, col = "blue")



#Observed frequencies of class i are entered into a table and expected frequencies are calculated for pi=1/7.  
cut<-cut(0, interval, right=F)  
voltage.freq<- c(27,3,5,3,1,1,2)  
names(voltage.freq)<-levels(cut)  
voltage.prob<-rep(1/k,k)  
n<-42  
voltage.exp<-n\*voltage.prob  
  
#View the observed and expected frequencies together.  
cbind(Observed=voltage.freq,Expected=voltage.exp)

## Observed Expected  
## [-Inf,-1.87e+09) 27 6  
## [-1.87e+09,3.64e+08) 3 6  
## [3.64e+08,2.08e+09) 5 6  
## [2.08e+09,3.69e+09) 3 6  
## [3.69e+09,5.41e+09) 1 6  
## [5.41e+09,7.64e+09) 1 6  
## [7.64e+09, Inf) 2 6

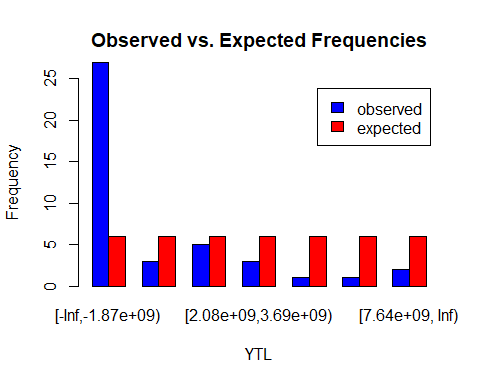
#We do the Goodness of Fit test with 7-2-1=4 degrees of freedom since two parameters of normal dist. is estimated.  
#We need to readjust the degrees of freedom by using the following functions.  
voltage.Xsq<-chisq.test(voltage.freq,p=voltage.prob)  
voltage.Xsq$parameter <- c(df=4)  
voltage.Xsq$p.value <- pchisq(voltage.Xsq$statistic, df=4, lower=F)  
  
#We can view the result of the test, observed and expected frequencies.  
voltage.Xsq

##   
## Chi-squared test for given probabilities  
##   
## data: voltage.freq  
## X-squared = 88, df = 4, p-value <0.0000000000000002

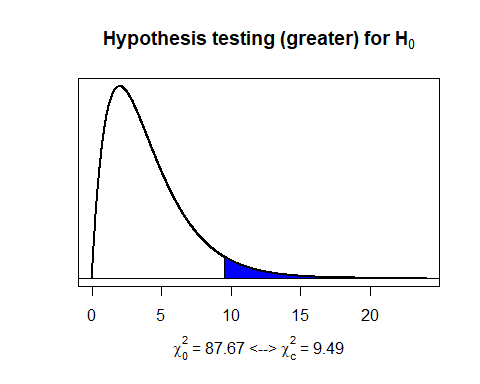
voltage.table<-rbind(observed=voltage.Xsq$observed, expected=voltage.Xsq$expected); t(voltage.table)

## observed expected  
## [-Inf,-1.87e+09) 27 6  
## [-1.87e+09,3.64e+08) 3 6  
## [3.64e+08,2.08e+09) 5 6  
## [2.08e+09,3.69e+09) 3 6  
## [3.69e+09,5.41e+09) 1 6  
## [5.41e+09,7.64e+09) 1 6  
## [7.64e+09, Inf) 2 6

barplot(voltage.table,ylim=c(0,25),beside=TRUE,legend=TRUE,main="Observed vs. Expected Frequencies",col=c("blue", "red"),xlab="YTL", ylab="Frequency")



#We can plot the hypothesis testing for goodness of fit using plotvar.hypothesis function that we defined before.  
plotvar.hypothesis(voltage.Xsq$statistic,alpha=0.05,df1=4,type='c',alt='g')



#Write your codes here. Their results will appear after knitting the markdown file.  
  
#Define the observed contingency table of CE, PE, OCE, D and GNP  
contce <- c(data\_22$CE)  
contpe <- c(data\_22$PE)  
contoce <- c(data\_22$OCE)  
contd <- c(data\_22$D)  
contgnp <- c(data\_22$GNP)  
  
contall <- rbind(contce,contpe,contoce,contd,contgnp); addmargins(contall)

##   
## contce 668.0 776.6 1314.3 2422.0 3795.6 4401.0 7575 12406 20168  
## contpe 553.3 627.1 991.5 1971.1 3012.1 3271.8 6032 10263 17230  
## contoce 114.7 149.5 322.8 450.9 783.5 1129.2 1543 2143 2938  
## contd 143.6 124.9 198.9 325.8 548.6 757.9 1262 2538 3598  
## contgnp 3810.5 5251.0 8504.5 17541.5 24525.6 34148.0 47040 73938 126874  
## Sum 5290.1 6929.1 11332.0 22711.3 32665.4 43707.9 63452 101288 170808  
##   
## contce 32418 45967 75887 129114 247636 400164 654652 1162463 3037804  
## contpe 27676 39041 65664 108093 206943 341749 557737 973752 2669305  
## contoce 4742 6925 10223 21021 40693 58415 96915 188710 368498  
## contd 4982 5980 8565 12550 20140 46975 84150 126100 310266  
## contgnp 195143 289106 485848 910059 1547793 2273698 4037702 6941224 16581567  
## Sum 264961 387019 646187 1180837 2063205 3121001 5431156 9392250 22967440  
##   
## contce 6263359 10671604 20375174 39372739 75952566 129126691 162283574  
## contpe 5509474 9275954 17756985 34074324 65793169 113053505 137522687  
## contoce 753886 1395650 2618188 5298414 10159397 16073186 24760887  
## contd 747500 1247500 4325000 11000000 22421093 24275000 34870000  
## contgnp 35178972 63576940 117683404 233660805 407069775 651380055 1070424473  
## Sum 48453191 86167648 162758751 323406282 581396000 933908437 1429861620  
##   
## contce 220646137 334498045 436157692 571203866 742526704 932560259  
## contpe 186722955 285819432 374790266 483139434 624001847 756015537  
## contoce 33923183 48678614 61367426 88064432 118524857 176544722  
## contd 54000000 68918000 78000000 80145649 118843456 120047875  
## contgnp 1418703264 1907070964 2520806747 3143699612 4101387191 4671255886  
## Sum 1913995538 2644985054 3471122132 4366252993 5705284054 6656424279  
##   
## contce 1024958716 1103415415 1140049984 1159028230 1199857540 1295770450  
## contpe 866264413 936237771 947401504 989460966 1016009249 1097304837  
## contoce 158694303 167177644 192648480 169567264 183848292 198465613  
## contd 146813407 167395579 163327203 199979873 194687300 201795988  
## contgnp 5128334134 5415280699 5649534936 6559174529 6915831629 7579403276  
## Sum 7325064973 7789507108 8092962107 9077210863 9510234010 10372740164  
##   
## contce 1465520572 1566222876 1680388904 1956564794 2340681611  
## contpe 1240105108 1324714693 1425718505 1645246451 1965616053  
## contoce 225415464 241508183 254670399 311318343 375065558  
## contd 227517948 240860181 252444741 305176268 170498906  
## contgnp 8840388008 10210731660 11605460378 14551761179 18334799699  
## Sum 11998947099 13584037594 15218682927 18770067035 23186661828  
## Sum  
## contce 19619937132  
## contpe 16552589029  
## contoce 3067348103  
## contd 2889967673  
## contgnp 121166201989  
## Sum 163296043926

contall.prop <- prop.table(contall)  
  
#Compute ui, vj and expected frequencies.  
u <- rowSums(contall.prop); u

## contce contpe contoce contd contgnp   
## 0.12015 0.10137 0.01878 0.01770 0.74200

v <- colSums(contall.prop); v

## [1] 0.00000003240 0.00000004243 0.00000006940 0.00000013908 0.00000020004  
## [6] 0.00000026766 0.00000038857 0.00000062027 0.00000104600 0.00000162258  
## [11] 0.00000237005 0.00000395715 0.00000723126 0.00001263475 0.00001911253  
## [16] 0.00003325957 0.00005751670 0.00014064909 0.00029671993 0.00052767750  
## [21] 0.00099670970 0.00198049061 0.00356038019 0.00571911244 0.00875625389  
## [26] 0.01172101597 0.01619748397 0.02125662109 0.02673826559 0.03493828703  
## [31] 0.04076292431 0.04485757767 0.04770175027 0.04956006228 0.05558745114  
## [36] 0.05823921867 0.06352107445 0.07347971703 0.08318656881 0.09319688684  
## [41] 0.11494502000 0.14199157108

E <- sum(contall)\*u%o%v; addmargins(E)

##   
## contce 635.60 832.5 1361.5 2728.8 3924.7 5251.5 7624 12170 20522  
## contpe 536.23 702.4 1148.7 2302.1 3311.1 4430.5 6432 10267 17314  
## contoce 99.37 130.2 212.9 426.6 613.6 821.0 1192 1903 3208  
## contd 93.62 122.6 200.6 401.9 578.1 773.5 1123 1793 3023  
## contgnp 3925.27 5141.4 8408.4 16851.9 24237.8 32431.4 47082 75156 126740  
## Sum 5290.10 6929.1 11332.0 22711.3 32665.4 43707.9 63452 101287 170808  
##   
## contce 31835 46500 77639 141877 247893 374987 652551 1128474 2759526  
## contpe 26858 39230 65501 119696 209138 316362 550532 952050 2328107  
## contoce 4977 7270 12138 22181 38755 58625 102019 176424 431420  
## contd 4689 6849 11436 20898 36514 55235 96119 166221 406471  
## contgnp 196602 287170 479473 876185 1530905 2315793 4029936 6969080 17041916  
## Sum 264961 387019 646187 1180837 2063205 3121001 5431156 9392250 22967440  
##   
## contce 5821626 10352999 19555382 38857101 69854436 112208627 171797151  
## contpe 4911483 8734429 16498126 32782247 58933510 94666118 144938672  
## contoce 910143 1618571 3057256 6074854 10920925 17542509 26858479  
## contd 857511 1524971 2880459 5723554 10289384 16528050 25305291  
## contgnp 35952427 63936679 120767529 239968526 431397746 692963133 1060962028  
## Sum 48453191 86167648 162758751 323406282 581396000 933908437 1429861620  
##   
## contce 229965596 317793617 417053569 524603090 685486995 799766012  
## contpe 194013160 268110296 351852113 442587522 578319107 674731934  
## contoce 35952436 49683322 65201456 82015568 107167888 125034079  
## contd 33873357 46810205 61430948 77272723 100970520 117803534  
## contgnp 1420190988 1962587615 2575584045 3239774090 4233339544 4939088721  
## Sum 1913995538 2644985054 3471122132 4366252993 5705284054 6656424279  
##   
## contce 880102854 935905341 972365306 1090622297 1142649809 1246279487  
## contpe 742509048 789587468 820347343 920116234 964009852 1051438240  
## contoce 137593806 146317873 152017963 170506063 178639957 194841247  
## contd 129636949 137856516 143226978 160645937 168309459 183573852  
## contgnp 5435222316 5779839909 6005004517 6735320333 7056624933 7696607338  
## Sum 7325064973 7789507108 8092962107 9077210863 9510234010 10372740164  
##   
## contce 1441667429 1632115250 1828517061 2255214066 2785865698  
## contpe 1216279558 1376953086 1542649767 1902637677 2350328122  
## contoce 225387871 255162164 285867294 352576389 435537576  
## contd 212354007 240406495 269335990 332187392 410351050  
## contgnp 8903258235 10079400599 11292312816 13927451511 17204579382  
## Sum 11998947099 13584037594 15218682927 18770067035 23186661828  
## Sum  
## contce 19619937132  
## contpe 16552589029  
## contoce 3067348103  
## contd 2889967673  
## contgnp 121166201989  
## Sum 163296043926

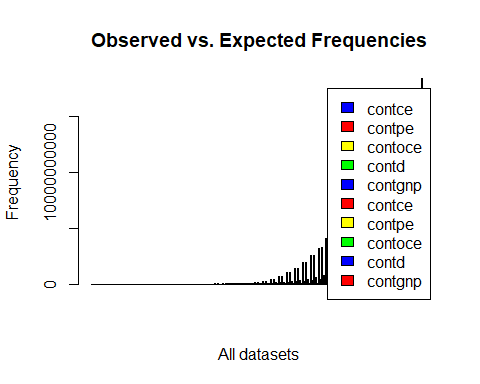
#We do the Goodness of Fit test with (42-1)\*(5-1)=164 degrees of freedom  
Xsq <- chisq.test(contall)  
  
#We can view the result of the test, observed and expected frequencies.  
Xsq

##   
## Pearson's Chi-squared test  
##   
## data: contall  
## X-squared = 965698073, df = 164, p-value <0.0000000000000002

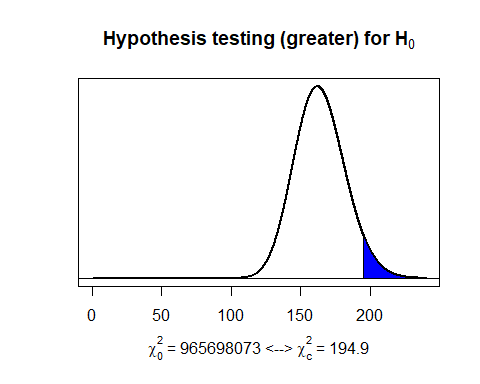
cont.table<-rbind(observed=Xsq$observed, expected=Xsq$expected); cont.table

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
## contce 668.00 776.6 1314.3 2422.0 3795.6 4401.0 7575 12406 20168 32418  
## contpe 553.30 627.1 991.5 1971.1 3012.1 3271.8 6032 10263 17230 27676  
## contoce 114.70 149.5 322.8 450.9 783.5 1129.2 1543 2143 2938 4742  
## contd 143.60 124.9 198.9 325.8 548.6 757.9 1262 2538 3598 4982  
## contgnp 3810.50 5251.0 8504.5 17541.5 24525.6 34148.0 47040 73938 126874 195143  
## contce 635.60 832.5 1361.5 2728.8 3924.7 5251.5 7624 12170 20522 31835  
## contpe 536.23 702.4 1148.7 2302.1 3311.1 4430.5 6432 10267 17314 26858  
## contoce 99.37 130.2 212.9 426.6 613.6 821.0 1192 1903 3208 4977  
## contd 93.62 122.6 200.6 401.9 578.1 773.5 1123 1793 3023 4689  
## contgnp 3925.27 5141.4 8408.4 16851.9 24237.8 32431.4 47082 75156 126740 196602  
## [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19]  
## contce 45967 75887 129114 247636 400164 654652 1162463 3037804 6263359  
## contpe 39041 65664 108093 206943 341749 557737 973752 2669305 5509474  
## contoce 6925 10223 21021 40693 58415 96915 188710 368498 753886  
## contd 5980 8565 12550 20140 46975 84150 126100 310266 747500  
## contgnp 289106 485848 910059 1547793 2273698 4037702 6941224 16581567 35178972  
## contce 46500 77639 141877 247893 374987 652551 1128474 2759526 5821626  
## contpe 39230 65501 119696 209138 316362 550532 952050 2328107 4911483  
## contoce 7270 12138 22181 38755 58625 102019 176424 431420 910143  
## contd 6849 11436 20898 36514 55235 96119 166221 406471 857511  
## contgnp 287170 479473 876185 1530905 2315793 4029936 6969080 17041916 35952427  
## [,20] [,21] [,22] [,23] [,24] [,25] [,26]  
## contce 10671604 20375174 39372739 75952566 129126691 162283574 220646137  
## contpe 9275954 17756985 34074324 65793169 113053505 137522687 186722955  
## contoce 1395650 2618188 5298414 10159397 16073186 24760887 33923183  
## contd 1247500 4325000 11000000 22421093 24275000 34870000 54000000  
## contgnp 63576940 117683404 233660805 407069775 651380055 1070424473 1418703264  
## contce 10352999 19555382 38857101 69854436 112208627 171797151 229965596  
## contpe 8734429 16498126 32782247 58933510 94666118 144938672 194013160  
## contoce 1618571 3057256 6074854 10920925 17542509 26858479 35952436  
## contd 1524971 2880459 5723554 10289384 16528050 25305291 33873357  
## contgnp 63936679 120767529 239968526 431397746 692963133 1060962028 1420190988  
## [,27] [,28] [,29] [,30] [,31] [,32]  
## contce 334498045 436157692 571203866 742526704 932560259 1024958716  
## contpe 285819432 374790266 483139434 624001847 756015537 866264413  
## contoce 48678614 61367426 88064432 118524857 176544722 158694303  
## contd 68918000 78000000 80145649 118843456 120047875 146813407  
## contgnp 1907070964 2520806747 3143699612 4101387191 4671255886 5128334134  
## contce 317793617 417053569 524603090 685486995 799766012 880102854  
## contpe 268110296 351852113 442587522 578319107 674731934 742509048  
## contoce 49683322 65201456 82015568 107167888 125034079 137593806  
## contd 46810205 61430948 77272723 100970520 117803534 129636949  
## contgnp 1962587615 2575584045 3239774090 4233339544 4939088721 5435222316  
## [,33] [,34] [,35] [,36] [,37] [,38]  
## contce 1103415415 1140049984 1159028230 1199857540 1295770450 1465520572  
## contpe 936237771 947401504 989460966 1016009249 1097304837 1240105108  
## contoce 167177644 192648480 169567264 183848292 198465613 225415464  
## contd 167395579 163327203 199979873 194687300 201795988 227517948  
## contgnp 5415280699 5649534936 6559174529 6915831629 7579403276 8840388008  
## contce 935905341 972365306 1090622297 1142649809 1246279487 1441667429  
## contpe 789587468 820347343 920116234 964009852 1051438240 1216279558  
## contoce 146317873 152017963 170506063 178639957 194841247 225387871  
## contd 137856516 143226978 160645937 168309459 183573852 212354007  
## contgnp 5779839909 6005004517 6735320333 7056624933 7696607338 8903258235  
## [,39] [,40] [,41] [,42]  
## contce 1566222876 1680388904 1956564794 2340681611  
## contpe 1324714693 1425718505 1645246451 1965616053  
## contoce 241508183 254670399 311318343 375065558  
## contd 240860181 252444741 305176268 170498906  
## contgnp 10210731660 11605460378 14551761179 18334799699  
## contce 1632115250 1828517061 2255214066 2785865698  
## contpe 1376953086 1542649767 1902637677 2350328122  
## contoce 255162164 285867294 352576389 435537576  
## contd 240406495 269335990 332187392 410351050  
## contgnp 10079400599 11292312816 13927451511 17204579382

barplot(cont.table,beside=TRUE,legend=TRUE,main="Observed vs. Expected Frequencies",  
 col=c("blue", "red","yellow","green"),xlab="All datasets", ylab="Frequency")



#We can plot the hypothesis testing for goodness of fit using plot.hypothesis function that we defined before.  
plotvar.hypothesis(Xsq$statistic,alpha=0.05,df1=164,type='c',alt='g')

  
…

### 3.2.4. Nonparametric Test Procedures

Nonparametric tests are also called distribution-free tests used for data does not follow a specific distribution. …

#Write your codes here. Their results will appear after knitting the markdown file.

…

#### 3.2.4.1. Sign Test

The sign test is used to test the null hypothesis that the median of a distribution is equal to some value here median= mean is tested …

"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

mediance <-(mean(data\_22$CE))  
expenditurece <-(data\_22$CE)  
alpha <- 0.05  
  
SIGN.test(expenditurece, md = mediance)

##   
## One-sample Sign-Test  
##   
## data: expenditurece  
## s = 14, p-value = 0.04  
## alternative hypothesis: true median is not equal to 467141360  
## 95 percent confidence interval:  
## 436697 421563901  
## sample estimates:  
## median of x   
## 29873956   
##   
## Achieved and Interpolated Confidence Intervals:   
##   
## Conf.Level L.E.pt U.E.pt  
## Lower Achieved CI 0.9116 654652 334498045  
## Interpolated CI 0.9500 436697 421563901  
## Upper Achieved CI 0.9564 400164 436157692

"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

medianpe <-(mean(data\_22$PE))  
expenditurepe <-(data\_22$PE)  
alpha <- 0.05  
  
SIGN.test(expenditurepe, md = medianpe)

##   
## One-sample Sign-Test  
##   
## data: expenditurepe  
## s = 14, p-value = 0.04  
## alternative hypothesis: true median is not equal to 394109263  
## 95 percent confidence interval:  
## 372755 362018022  
## sample estimates:  
## median of x   
## 25915655   
##   
## Achieved and Interpolated Confidence Intervals:   
##   
## Conf.Level L.E.pt U.E.pt  
## Lower Achieved CI 0.9116 557737 285819432  
## Interpolated CI 0.9500 372755 362018022  
## Upper Achieved CI 0.9564 341749 374790266

"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

medianoce <-(mean(data\_22$OCE))  
expenditureoce <-(data\_22$OCE)  
alpha <- 0.05  
  
SIGN.test(expenditureoce, md = medianoce)

##   
## One-sample Sign-Test  
##   
## data: expenditureoce  
## s = 14, p-value = 0.04  
## alternative hypothesis: true median is not equal to 73032098  
## 95 percent confidence interval:  
## 63942 59545879  
## sample estimates:  
## median of x   
## 3958301   
##   
## Achieved and Interpolated Confidence Intervals:   
##   
## Conf.Level L.E.pt U.E.pt  
## Lower Achieved CI 0.9116 96915 48678614  
## Interpolated CI 0.9500 63942 59545879  
## Upper Achieved CI 0.9564 58415 61367426

"FOR DEFENCE"

## [1] "FOR DEFENCE"

mediand <-(mean(data\_22$D))  
expenditured <-(data\_22$D)  
alpha <- 0.05  
  
SIGN.test(expenditured, md = mediand)

##   
## One-sample Sign-Test  
##   
## data: expenditured  
## s = 16, p-value = 0.2  
## alternative hypothesis: true median is not equal to 68808754  
## 95 percent confidence interval:  
## 52312 76696230  
## sample estimates:  
## median of x   
## 7662500   
##   
## Achieved and Interpolated Confidence Intervals:   
##   
## Conf.Level L.E.pt U.E.pt  
## Lower Achieved CI 0.9116 84150 68918000  
## Interpolated CI 0.9500 52312 76696230  
## Upper Achieved CI 0.9564 46975 78000000

"FOR GNP"

## [1] "FOR GNP"

mediangnp <-(mean(data\_22$GNP))  
expendituregnp <-(data\_22$GNP)  
alpha <- 0.05  
  
SIGN.test(expendituregnp, md = mediangnp)

##   
## One-sample Sign-Test  
##   
## data: expendituregnp  
## s = 14, p-value = 0.04  
## alternative hypothesis: true median is not equal to 2884909571  
## 95 percent confidence interval:  
## 2526930 2432701660  
## sample estimates:  
## median of x   
## 175672105   
##   
## Achieved and Interpolated Confidence Intervals:   
##   
## Conf.Level L.E.pt U.E.pt  
## Lower Achieved CI 0.9116 4037702 1907070964  
## Interpolated CI 0.9500 2526930 2432701660  
## Upper Achieved CI 0.9564 2273698 2520806747

#Write your codes here. Their results will appear after knitting the markdown file.

…

#### 3.2.4.2. Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used to compare two related samples. Here again median is tested …

"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

mediance <-(mean(data\_22$CE))  
expenditurece <-(data\_22$CE)  
wilcox.test(expenditurece, alt="t",mu=mediance)

##   
## Wilcoxon signed rank exact test  
##   
## data: expenditurece  
## V = 428, p-value = 0.8  
## alternative hypothesis: true location is not equal to 467141360

"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

medianpe <-(mean(data\_22$PE))  
expenditurepe <-(data\_22$PE)  
wilcox.test(expenditurepe, alt="t",mu=medianpe)

##   
## Wilcoxon signed rank exact test  
##   
## data: expenditurepe  
## V = 424, p-value = 0.7  
## alternative hypothesis: true location is not equal to 394109263

"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

medianoce <-(mean(data\_22$OCE))  
expenditureoce <-(data\_22$OCE)  
wilcox.test(expenditureoce, alt="t",mu=medianoce)

##   
## Wilcoxon signed rank exact test  
##   
## data: expenditureoce  
## V = 445, p-value = 0.9  
## alternative hypothesis: true location is not equal to 73032098

"FOR DEFENCE"

## [1] "FOR DEFENCE"

mediand <-(mean(data\_22$D))  
expenditured <-(data\_22$D)  
wilcox.test(expenditured, alt="t",mu=mediand)

##   
## Wilcoxon signed rank exact test  
##   
## data: expenditured  
## V = 430, p-value = 0.8  
## alternative hypothesis: true location is not equal to 68808754

"FOR GNP"

## [1] "FOR GNP"

mediangnp <-(mean(data\_22$GNP))  
expendituregnp <-(data\_22$GNP)  
wilcox.test(expendituregnp, alt="t",mu=mediangnp)

##   
## Wilcoxon signed rank exact test  
##   
## data: expendituregnp  
## V = 352, p-value = 0.2  
## alternative hypothesis: true location is not equal to 2884909571

…

#### 3.2.4.3. Wilcoxon Rank-Sum Test

Wilcoxon Rank Sum Test, is used to test whether two samples are likely to derive from the same population. Here Personnel Expenditures-Other Current Expenditures,Personnel Expenditures and Defence, Other Current expentitures and defence are tested …

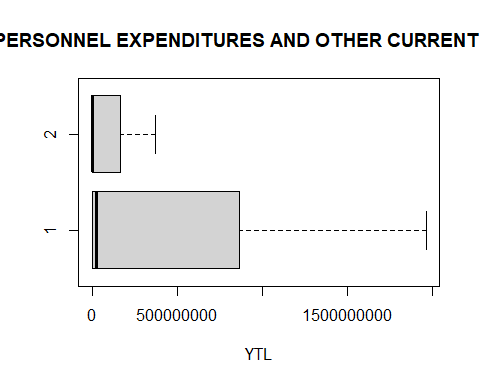
"FOR PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES"

axial<-cbind(data\_22$PE, data\_22$OCE); axial

## [,1] [,2]  
## [1,] 553.3 114.7  
## [2,] 627.1 149.5  
## [3,] 991.5 322.8  
## [4,] 1971.1 450.9  
## [5,] 3012.1 783.5  
## [6,] 3271.8 1129.2  
## [7,] 6032.0 1543.1  
## [8,] 10262.7 2143.2  
## [9,] 17229.9 2937.6  
## [10,] 27676.0 4742.2  
## [11,] 39041.3 6925.2  
## [12,] 65663.5 10223.4  
## [13,] 108093.1 21020.9  
## [14,] 206942.8 40693.4  
## [15,] 341748.5 58415.4  
## [16,] 557737.0 96915.1  
## [17,] 973752.4 188710.3  
## [18,] 2669305.3 368498.2  
## [19,] 5509473.8 753885.6  
## [20,] 9275954.3 1395649.7  
## [21,] 17756985.3 2618188.3  
## [22,] 34074324.2 5298414.3  
## [23,] 65793169.1 10159396.9  
## [24,] 113053504.6 16073186.2  
## [25,] 137522686.8 24760886.7  
## [26,] 186722954.6 33923182.8  
## [27,] 285819431.5 48678613.7  
## [28,] 374790266.1 61367426.1  
## [29,] 483139433.6 88064432.3  
## [30,] 624001846.5 118524857.4  
## [31,] 756015536.9 176544721.7  
## [32,] 866264412.7 158694303.0  
## [33,] 936237770.8 167177644.1  
## [34,] 947401504.3 192648480.2  
## [35,] 989460966.1 169567264.2  
## [36,] 1016009248.8 183848291.6  
## [37,] 1097304837.3 198465612.7  
## [38,] 1240105107.7 225415463.9  
## [39,] 1324714692.8 241508183.4  
## [40,] 1425718505.2 254670398.7  
## [41,] 1645246451.0 311318342.9  
## [42,] 1965616053.4 375065558.1

boxplot(axial,horizontal=T,xlab="YTL",main="Boxplot for PERSONNEL EXPENDITURES AND OTHER CURRENT EXPENDITURES")



wilcox.test(data\_22$PE, data\_22$OCE, alt="t", exact=F, correct=F)

##   
## Wilcoxon rank sum test  
##   
## data: data\_22$PE and data\_22$OCE  
## W = 1106, p-value = 0.05  
## alternative hypothesis: true location shift is not equal to 0

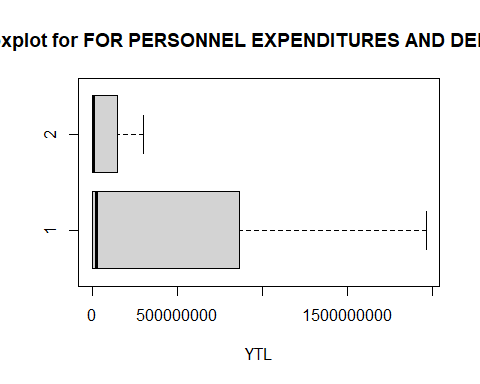
"FOR PERSONNEL EXPENDITURES AND DEFENCE"

## [1] "FOR PERSONNEL EXPENDITURES AND DEFENCE"

axial<-cbind(data\_22$PE, data\_22$D); axial

## [,1] [,2]  
## [1,] 553.3 143.6  
## [2,] 627.1 124.9  
## [3,] 991.5 198.9  
## [4,] 1971.1 325.8  
## [5,] 3012.1 548.6  
## [6,] 3271.8 757.9  
## [7,] 6032.0 1261.7  
## [8,] 10262.7 2537.9  
## [9,] 17229.9 3598.4  
## [10,] 27676.0 4982.0  
## [11,] 39041.3 5980.0  
## [12,] 65663.5 8564.7  
## [13,] 108093.1 12550.0  
## [14,] 206942.8 20140.0  
## [15,] 341748.5 46975.0  
## [16,] 557737.0 84150.0  
## [17,] 973752.4 126100.0  
## [18,] 2669305.3 310265.8  
## [19,] 5509473.8 747500.0  
## [20,] 9275954.3 1247500.0  
## [21,] 17756985.3 4325000.0  
## [22,] 34074324.2 11000000.0  
## [23,] 65793169.1 22421093.0  
## [24,] 113053504.6 24275000.0  
## [25,] 137522686.8 34870000.0  
## [26,] 186722954.6 54000000.0  
## [27,] 285819431.5 68918000.0  
## [28,] 374790266.1 78000000.0  
## [29,] 483139433.6 80145649.0  
## [30,] 624001846.5 118843456.0  
## [31,] 756015536.9 120047875.5  
## [32,] 866264412.7 146813407.4  
## [33,] 936237770.8 167395579.2  
## [34,] 947401504.3 163327202.5  
## [35,] 989460966.1 199979873.2  
## [36,] 1016009248.8 194687299.6  
## [37,] 1097304837.3 201795988.0  
## [38,] 1240105107.7 227517948.2  
## [39,] 1324714692.8 240860181.3  
## [40,] 1425718505.2 252444740.9  
## [41,] 1645246451.0 305176267.8  
## [42,] 1965616053.4 170498906.2

boxplot(axial,horizontal=T,xlab="YTL",main="Boxplot for FOR PERSONNEL EXPENDITURES AND DEFENCE")



wilcox.test(data\_22$PE, data\_22$D, alt="t", exact=F, correct=F)

##   
## Wilcoxon rank sum test  
##   
## data: data\_22$PE and data\_22$D  
## W = 1106, p-value = 0.05  
## alternative hypothesis: true location shift is not equal to 0

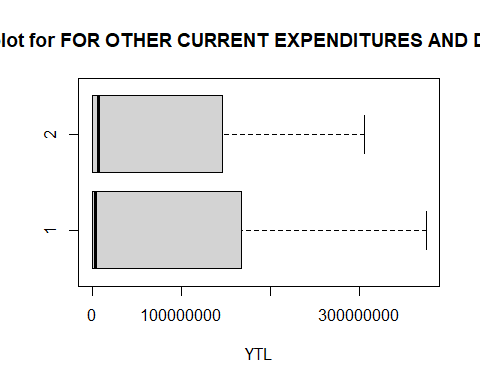
"FOR OTHER CURRENT EXPENDITURES AND DEFENCE"

## [1] "FOR OTHER CURRENT EXPENDITURES AND DEFENCE"

axial<-cbind(data\_22$OCE, data\_22$D); axial

## [,1] [,2]  
## [1,] 114.7 143.6  
## [2,] 149.5 124.9  
## [3,] 322.8 198.9  
## [4,] 450.9 325.8  
## [5,] 783.5 548.6  
## [6,] 1129.2 757.9  
## [7,] 1543.1 1261.7  
## [8,] 2143.2 2537.9  
## [9,] 2937.6 3598.4  
## [10,] 4742.2 4982.0  
## [11,] 6925.2 5980.0  
## [12,] 10223.4 8564.7  
## [13,] 21020.9 12550.0  
## [14,] 40693.4 20140.0  
## [15,] 58415.4 46975.0  
## [16,] 96915.1 84150.0  
## [17,] 188710.3 126100.0  
## [18,] 368498.2 310265.8  
## [19,] 753885.6 747500.0  
## [20,] 1395649.7 1247500.0  
## [21,] 2618188.3 4325000.0  
## [22,] 5298414.3 11000000.0  
## [23,] 10159396.9 22421093.0  
## [24,] 16073186.2 24275000.0  
## [25,] 24760886.7 34870000.0  
## [26,] 33923182.8 54000000.0  
## [27,] 48678613.7 68918000.0  
## [28,] 61367426.1 78000000.0  
## [29,] 88064432.3 80145649.0  
## [30,] 118524857.4 118843456.0  
## [31,] 176544721.7 120047875.5  
## [32,] 158694303.0 146813407.4  
## [33,] 167177644.1 167395579.2  
## [34,] 192648480.2 163327202.5  
## [35,] 169567264.2 199979873.2  
## [36,] 183848291.6 194687299.6  
## [37,] 198465612.7 201795988.0  
## [38,] 225415463.9 227517948.2  
## [39,] 241508183.4 240860181.3  
## [40,] 254670398.7 252444740.9  
## [41,] 311318342.9 305176267.8  
## [42,] 375065558.1 170498906.2

boxplot(axial,horizontal=T,xlab="YTL",main="Boxplot for FOR OTHER CURRENT EXPENDITURES AND DEFENCE")



wilcox.test(data\_22$OCE, data\_22$D, alt="t", exact=F, correct=F)

##   
## Wilcoxon rank sum test  
##   
## data: data\_22$OCE and data\_22$D  
## W = 892, p-value = 0.9  
## alternative hypothesis: true location shift is not equal to 0

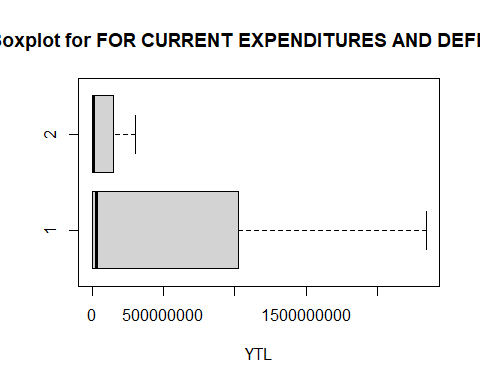
"FOR CURRENT EXPENDITURES AND DEFENCE"

## [1] "FOR CURRENT EXPENDITURES AND DEFENCE"

axial<-cbind(data\_22$CE, data\_22$D); axial

## [,1] [,2]  
## [1,] 668.0 143.6  
## [2,] 776.6 124.9  
## [3,] 1314.3 198.9  
## [4,] 2422.0 325.8  
## [5,] 3795.6 548.6  
## [6,] 4401.0 757.9  
## [7,] 7575.1 1261.7  
## [8,] 12405.9 2537.9  
## [9,] 20167.5 3598.4  
## [10,] 32418.2 4982.0  
## [11,] 45966.5 5980.0  
## [12,] 75886.9 8564.7  
## [13,] 129114.0 12550.0  
## [14,] 247636.2 20140.0  
## [15,] 400163.9 46975.0  
## [16,] 654652.1 84150.0  
## [17,] 1162462.7 126100.0  
## [18,] 3037803.5 310265.8  
## [19,] 6263359.4 747500.0  
## [20,] 10671604.0 1247500.0  
## [21,] 20375173.6 4325000.0  
## [22,] 39372738.5 11000000.0  
## [23,] 75952566.0 22421093.0  
## [24,] 129126690.9 24275000.0  
## [25,] 162283573.5 34870000.0  
## [26,] 220646137.4 54000000.0  
## [27,] 334498045.2 68918000.0  
## [28,] 436157692.2 78000000.0  
## [29,] 571203866.0 80145649.0  
## [30,] 742526703.9 118843456.0  
## [31,] 932560258.6 120047875.5  
## [32,] 1024958715.7 146813407.4  
## [33,] 1103415414.9 167395579.2  
## [34,] 1140049984.4 163327202.5  
## [35,] 1159028230.4 199979873.2  
## [36,] 1199857540.4 194687299.6  
## [37,] 1295770450.0 201795988.0  
## [38,] 1465520571.6 227517948.2  
## [39,] 1566222876.2 240860181.3  
## [40,] 1680388903.9 252444740.9  
## [41,] 1956564793.9 305176267.8  
## [42,] 2340681611.4 170498906.2

boxplot(axial,horizontal=T,xlab="YTL",main="Boxplot for FOR CURRENT EXPENDITURES AND DEFENCE")



wilcox.test(data\_22$CE, data\_22$D, alt="t", exact=F, correct=F)

##   
## Wilcoxon rank sum test  
##   
## data: data\_22$CE and data\_22$D  
## W = 1121, p-value = 0.03  
## alternative hypothesis: true location shift is not equal to 0

#Write your codes here. Their results will appear after knitting the markdown file.

…

## 3.3. Exploratory Data Analyses

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It can also help determine if the statistical techniques you are considering for data analysis are appropriate …

#Write your codes here. Their results will appear after knitting the markdown file.

…

### 3.3.1. Correlation Analyses

Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables …

#Write your codes here. Their results will appear after knitting the markdown file.

…

#### 3.3.1.1. Hypothesis Tests on the Correlation Coefficient

We perform a hypothesis test of the “significance of the correlation coefficient” to decide whether the linear relationship in the sample data is strong enough to use to model the relationship in the population. R squared values are compared …

#Write your codes here. Their results will appear after knitting the markdown file.  
m <- length(data\_22$CE)  
GNP <- c(data\_22$GNP)  
ct.mm <- qt(0.05/2,m-2,lower=F); ct.mm

## [1] 2.021

cor.test(data\_22$CE,GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 34, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9693 0.9912  
## sample estimates:  
## cor   
## 0.9835

cor.test(data\_22$CE,GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 34, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9693 0.9912  
## sample estimates:  
## cor   
## 0.9835

"FOR Personnel EXPENDITURES and GNP"

## [1] "FOR Personnel EXPENDITURES and GNP"

m <- length(data\_22$PE)  
GNP <- c(data\_22$GNP)  
ct.mm <- qt(0.05/2,m-2,lower=F); ct.mm

## [1] 2.021

cor.test(data\_22$PE,GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 34, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9693 0.9911  
## sample estimates:  
## cor   
## 0.9835

"FOR Other Current EXPENDITURES and GNP"

## [1] "FOR Other Current EXPENDITURES and GNP"

m <- length(data\_22$OCE)  
GNP <- c(data\_22$GNP)  
ct.mm <- qt(0.05/2,m-2,lower=F); ct.mm

## [1] 2.021

cor.test(data\_22$OCE,GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 32, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9657 0.9901  
## sample estimates:  
## cor   
## 0.9816

"FOR DEFENCE and GNP"

## [1] "FOR DEFENCE and GNP"

m <- length(data\_22$D)  
GNP <- c(data\_22$GNP)  
ct.mm <- qt(0.05/2,m-2,lower=F); ct.mm

## [1] 2.021

cor.test(data\_22$D,GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 14, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8347 0.9499  
## sample estimates:  
## cor   
## 0.9082

…

#### 3.3.1.2. Confidence Intervals for the Correlation Coefficient

A confidence interval gives an estimated range of r values which is likely to include an unknown population ρ, the estimated range being calculated from a given set of sample data.The confidence intervals of correlation coefficient are computed based on the sample mean r and sample standard deviation. …

#Write your codes here. Their results will appear after knitting the markdown file.  
cor.test(data\_22$CE,data\_22$GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 34, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9693 0.9912  
## sample estimates:  
## cor   
## 0.9835

cor.test(data\_22$PE,data\_22$GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 34, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9693 0.9911  
## sample estimates:  
## cor   
## 0.9835

cor.test(data\_22$OCE,data\_22$GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 32, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9657 0.9901  
## sample estimates:  
## cor   
## 0.9816

cor.test(data\_22$D,data\_22$GNP)

##   
## Pearson's product-moment correlation  
##   
## data: x and y  
## t = 14, df = 40, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8347 0.9499  
## sample estimates:  
## cor   
## 0.9082

…

### 3.3.2. Regression Analyses

Regression analysis is a set of statistical methods used for the estimation of relationships between a dependent variable and one or more independent variables. …

#Write your codes here. Their results will appear after knitting the markdown file.

…

#### 3.3.2.1. Regression Modeling

Regression analysis is a predictive modelling technique that analyzes the relation between the target or dependent variable and independent variable in a dataset. GNP is dependent variable and other metrics are independent so simple model and 3d model are drawn for analysis of relationship …

require(scatterplot3d)

## Loading required package: scatterplot3d

## Warning: package 'scatterplot3d' was built under R version 4.0.3

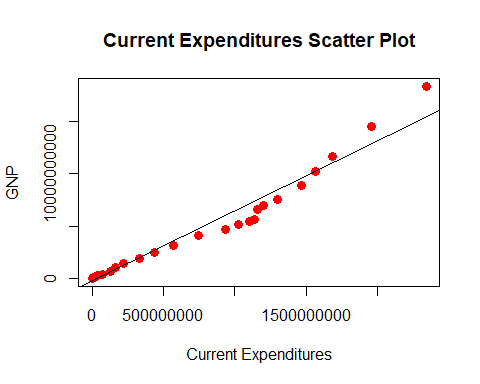
#Write your codes here. Their results will appear after knitting the markdown file.  
n <- 42  
y <- data\_22$GNP  
  
x1 <- data\_22$CE  
  
x2<-data\_22$PE  
  
x3 <- data\_22$OCE  
  
x4<-data\_22$D  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

#for defining model  
model1.1 <- lm(data\_22$GNP~data\_22$CE); model1.1

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$CE)  
##   
## Coefficients:  
## (Intercept) data\_22$CE   
## -226165667.00 6.66

#for plotting  
plot(data\_22$CE,data\_22$GNP, pch = 16, cex = 1.3, col = "red", main = "Current Expenditures Scatter Plot", xlab = "Current Expenditures", ylab = "GNP")  
abline(model1.1$coefficients)



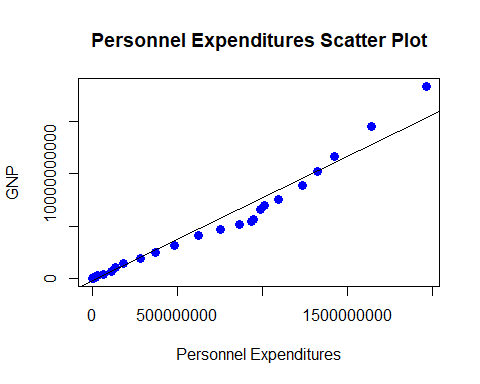
"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

#for defining model  
model1.2 <- lm(data\_22$GNP~data\_22$PE); model1.2

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$PE)  
##   
## Coefficients:  
## (Intercept) data\_22$PE   
## -229627474.9 7.9

#for plotting  
plot(data\_22$PE,data\_22$GNP, pch = 16, cex = 1.3, col = "blue", main = "Personnel Expenditures Scatter Plot", xlab = "Personnel Expenditures", ylab = "GNP")  
abline(model1.2$coefficients)



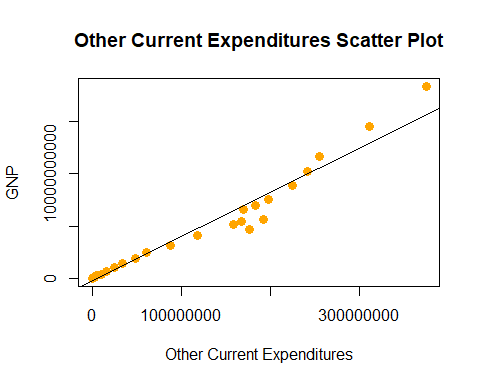
"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

#for defining model  
model1.3 <- lm(data\_22$GNP~data\_22$OCE); model1.3

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$OCE)  
##   
## Coefficients:  
## (Intercept) data\_22$OCE   
## -194110949.5 42.2

#for plotting  
plot(data\_22$OCE,data\_22$GNP, pch = 16, cex = 1.3, col = "orange", main = "Other Current Expenditures Scatter Plot", xlab = "Other Current Expenditures", ylab = "GNP")  
abline(model1.3$coefficients)



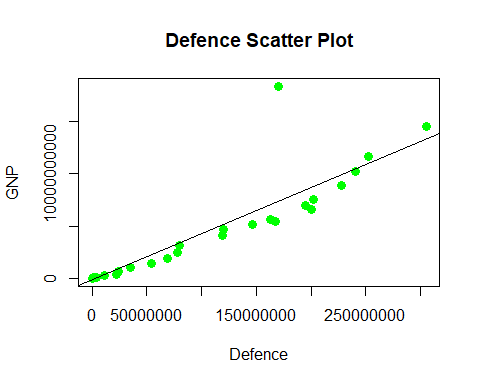
"FOR DEFENCE"

## [1] "FOR DEFENCE"

#for defining model  
model1.4 <- lm(data\_22$GNP~data\_22$D); model1.4

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$D)  
##   
## Coefficients:  
## (Intercept) data\_22$D   
## -159406019.1 44.2

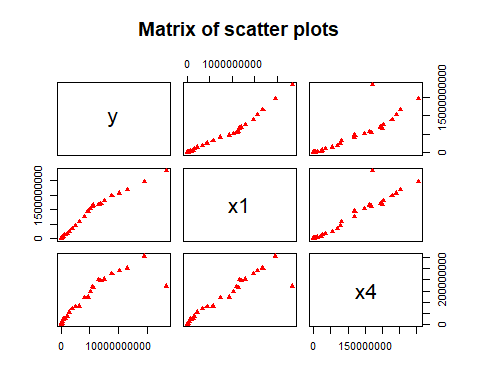
#for plotting  
plot(data\_22$D,data\_22$GNP, pch = 16, cex = 1.3, col = "green", main = "Defence Scatter Plot", xlab = "Defence", ylab = "GNP")  
abline(model1.4$coefficients)



#Create the multiple regression model for GNP, CE and D.  
model1<-lm(y~x1+x4); model1

##   
## Call:  
## lm(formula = y ~ x1 + x4)  
##   
## Coefficients:  
## (Intercept) x1 x4   
## -110211313.75 8.92 -17.04

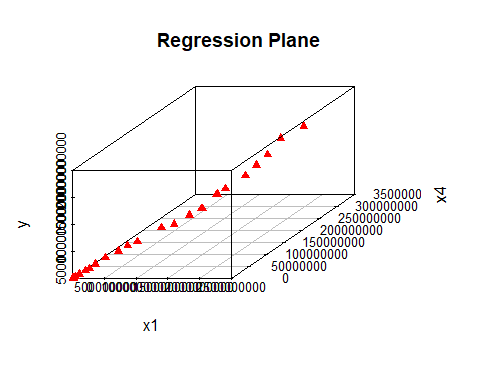
#Obtain the scatter plot matrix.  
pairs(y~x1+x4,pch=17,col='red',main='Matrix of scatter plots')



#Create the multiple regression model.  
model1<-lm(y~x1+x4,); model1

##   
## Call:  
## lm(formula = y ~ x1 + x4)  
##   
## Coefficients:  
## (Intercept) x1 x4   
## -110211313.75 8.92 -17.04

#Plot the regression plane on the 3D scatter diagram.  
G<-scatterplot3d(x1,x4, y, pch=17, color='red', main ="Regression Plane")

  
…

#### 3.3.2.2. Hypothesis Tests on the Slope and Intercept

We conducted Hypothesis test on slope and intercept for each variable in two tail test type. Then results are compared with critical value and decision is given about rejected or not …

# FOR GNP DEPENDENT MULTIPLE  
summary(model1); anova(model1)

##   
## Call:  
## lm(formula = y ~ x1 + x4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1628158802 34647222 110187782 110219462 2406578750   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -110211313.75 131108104.95 -0.84 0.41   
## x1 8.92 0.54 16.52 < 0.0000000000000002 \*\*\*  
## x4 -17.04 3.88 -4.39 0.000085 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 676000000 on 39 degrees of freedom  
## Multiple R-squared: 0.978, Adjusted R-squared: 0.977   
## F-statistic: 871 on 2 and 39 DF, p-value: <0.0000000000000002

## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value  
## x1 1 787045294150890029046 787045294150890029046 1723.3  
## x4 1 8786389339684835328 8786389339684835328 19.2  
## Residuals 39 17811644388137398272 456708830465061504   
## Pr(>F)   
## x1 < 0.0000000000000002 \*\*\*  
## x4 0.000085 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#FOR HYPOTHESIS ON HO:B(0,1)=0 AND H1:B(0,1)=1 WITH ALPHA 0.01 F AND T VALUES ARE COMPARISED  
  
 n <- 42  
"T TEST FOR ALPHA: 0.01"

## [1] "T TEST FOR ALPHA: 0.01"

tc <- qt(0.01/2,n-2,lower=F); tc

## [1] 2.704

"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

#for obtaining information about model  
summary(model1.1); anova(model1.1)

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$CE)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1716821872 -141932638 221073922 226161251 2972457859   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -226165666.996 154949927.732 -1.46 0.15   
## data\_22$CE 6.660 0.194 34.40 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 815000000 on 40 degrees of freedom  
## Multiple R-squared: 0.967, Adjusted R-squared: 0.966   
## F-statistic: 1.18e+03 on 1 and 40 DF, p-value: <0.0000000000000002

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$CE 1 787045294150890029046 787045294150890029046 1184  
## Residuals 40 26598033727822225408 664950843195555584   
## Pr(>F)   
## data\_22$CE <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

#for obtaining information about model  
summary(model1.2); anova(model1.2)

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$PE)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1753921283 -189086051 223190424 229625212 3030704428   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -229627474.88 155176339.81 -1.48 0.15   
## data\_22$PE 7.90 0.23 34.37 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 816000000 on 40 degrees of freedom  
## Multiple R-squared: 0.967, Adjusted R-squared: 0.966   
## F-statistic: 1.18e+03 on 1 and 40 DF, p-value: <0.0000000000000002

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$PE 1 786988380344163106806 786988380344163106806 1181  
## Residuals 40 26654947534549295104 666373688363732352   
## Pr(>F)   
## data\_22$PE <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

#for obtaining information about model  
summary(model1.3); anova(model1.3)

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$OCE)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2577727888 68594453 194096010 194158028 2716212037   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -194110949.5 163359824.9 -1.19 0.24   
## data\_22$OCE 42.2 1.3 32.48 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 862000000 on 40 degrees of freedom  
## Multiple R-squared: 0.963, Adjusted R-squared: 0.963   
## F-statistic: 1.05e+03 on 1 and 40 DF, p-value: <0.0000000000000002

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$OCE 1 783911783171596746722 783911783171596746722 1055  
## Residuals 40 29731544707115229184 743288617677880704   
## Pr(>F)   
## data\_22$OCE <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR DEFENCE"

## [1] "FOR DEFENCE"

#for obtaining information about model  
summary(model1.4); anova(model1.4)

##   
## Call:  
## lm(formula = data\_22$GNP ~ data\_22$D)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2129157701 -698222699 159377216 159579281 10950798126   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -159406019.08 366042448.64 -0.44 0.67   
## data\_22$D 44.24 3.22 13.73 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1890000000 on 40 degrees of freedom  
## Multiple R-squared: 0.825, Adjusted R-squared: 0.821   
## F-statistic: 188 on 1 and 40 DF, p-value: <0.0000000000000002

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$D 1 671172057876774387722 671172057876774387722 188  
## Residuals 40 142471270001937809418 3561781750048445440   
## Pr(>F)   
## data\_22$D <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

…

#### 3.3.2.3. Confidence Intervals for the Slope and Intercept

The slope representing the relationship of Y with X, and a 95% confidence interval for the x value were calculated and examined whether it excluded 0. If it does not contain 0, we can conclude that there is a significant linear relationship between X and Y. …

#We can find the 95% CI for the intercept and slopes as follows:  
confint(model1.1, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -539331152.664 86999818.672  
## data\_22$CE 6.269 7.051

confint(model1.2, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -543250556.415 83995606.660  
## data\_22$PE 7.438 8.367

confint(model1.3, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -524273471.43 136051572.50  
## data\_22$OCE 39.54 44.78

confint(model1.4, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -899205403.83 580393365.66  
## data\_22$D 37.73 50.76

#FOR MULTIPLE   
confint(model1)

## 2.5 % 97.5 %  
## (Intercept) -375402487.186 154979859.683  
## x1 7.829 10.014  
## x4 -24.896 -9.181

…

#### 3.3.2.4. Analysis of Sum of Squares and Mean Squares

The sum of squares represents a measure of variation or deviation from the mean. It is calculated as a summation of the squares of the differences from the mean. The calculation of the total sum of squares considers both the sum of squares from the factors and from randomness or error and also we calculated R squared value and table form is established …

n<-42  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

#for Sum of Squares analysis SST,SSR,SSE are Calculated  
xbar.g<- mean(data\_22$GNP)   
sst.ce <- sum((data\_22$GNP-xbar.g)^2); sst.ce

## [1] 813643327878712262646

ssr.ce <- sum((model1.1$fitted.values-xbar.g)^2); ssr.ce

## [1] 787045294150890029046

sse.ce <- sum(model1.1$residuals^2); sse.ce

## [1] 26598033727822225408

f0 <- (ssr.ce/1)/(sse.ce/(n-2)); f0

## [1] 1184

#for Mean Squares MSR,MSE are calculated  
msr.ce <- anova(model1.1)$`Mean Sq`[1]   
mse.ce <- anova(model1.1)$`Mean Sq`[2]  
#R square is calculated  
rsq.ce <- ssr.ce/sst.ce; rsq.ce

## [1] 0.9673

#for table form  
cbind(ssr.ce,sse.ce,sst.ce,msr.ce,mse.ce,rsq.ce)

## ssr.ce sse.ce sst.ce  
## [1,] 787045294150890029046 26598033727822225408 813643327878712262646  
## msr.ce mse.ce rsq.ce  
## [1,] 787045294150890029046 664950843195555584 0.9673

"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

#for Sum of Squares analysis SST,SSR,SSE are Calculated  
sst.pe <- sum((data\_22$GNP-xbar.g)^2); sst.pe

## [1] 813643327878712262646

ssr.pe <- sum((model1.2$fitted.values-xbar.g)^2); ssr.pe

## [1] 786988380344162975764

sse.pe <- sum(model1.2$residuals^2); sse.pe

## [1] 26654947534549295104

f1 <- (ssr.pe/1)/(sse.pe/(n-2)); f1

## [1] 1181

#for Mean Squares MSR,MSE are calculated  
msr.pe <- anova(model1.2)$`Mean Sq`[1]   
mse.pe <- anova(model1.2)$`Mean Sq`[2]   
  
#R square is calculated  
rsq.pe <- ssr.pe/sst.pe; rsq.pe

## [1] 0.9672

#for table form  
cbind(ssr.pe,sse.pe,sst.pe,msr.pe,mse.pe,rsq.pe)

## ssr.pe sse.pe sst.pe  
## [1,] 786988380344162975764 26654947534549295104 813643327878712262646  
## msr.pe mse.pe rsq.pe  
## [1,] 786988380344163106806 666373688363732352 0.9672

"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

#for Sum of Squares analysis SST,SSR,SSE are Calculated  
sst.oce <- sum((data\_22$GNP-xbar.g)^2); sst.oce

## [1] 813643327878712262646

ssr.oce <- sum((model1.3$fitted.values-xbar.g)^2); ssr.oce

## [1] 783911783171597008986

sse.oce <- sum(model1.3$residuals^2); sse.oce

## [1] 29731544707115229184

f2 <- (ssr.oce/1)/(sse.oce/(n-2)); f2

## [1] 1055

#for Mean Squares MSR,MSE are calculated  
msr.oce <- anova(model1.3)$`Mean Sq`[1]   
mse.oce <- anova(model1.3)$`Mean Sq`[2]   
  
#R square is calculated  
rsq.oce <- ssr.oce/sst.oce; rsq.oce

## [1] 0.9635

#for table form  
cbind(ssr.oce,sse.oce,sst.oce,msr.oce,mse.oce,rsq.oce)

## ssr.oce sse.oce sst.oce  
## [1,] 783911783171597008986 29731544707115229184 813643327878712262646  
## msr.oce mse.oce rsq.oce  
## [1,] 783911783171596746722 743288617677880704 0.9635

"DEFENCE"

## [1] "DEFENCE"

#for Sum of Squares analysis SST,SSR,SSE are Calculated  
sst.d <- sum((data\_22$GNP-xbar.g)^2); sst.d

## [1] 813643327878712262646

ssr.d <- sum((model1.4$fitted.values-xbar.g)^2); ssr.d

## [1] 671172057876774387722

sse.d <- sum(model1.4$residuals^2); sse.d

## [1] 142471270001937809418

f3 <- (ssr.d/1)/(sse.d/(n-2)); f3

## [1] 188.4

#for Mean Squares MSR,MSE are calculated  
msr.d <- anova(model1.4)$`Mean Sq`[1]   
mse.d <- anova(model1.4)$`Mean Sq`[2]   
  
#R square is calculated  
rsq.d <- ssr.d/sst.d; rsq.d

## [1] 0.8249

#for table form  
cbind(ssr.d,sse.d,sst.d,msr.d,mse.d,rsq.d)

## ssr.d sse.d sst.d  
## [1,] 671172057876774387722 142471270001937809418 813643327878712262646  
## msr.d mse.d rsq.d  
## [1,] 671172057876774387722 3561781750048445440 0.8249

#FOR MULTIPLE  
ybarm<-mean(y)  
#Calculate SST,SSR,SSE  
sst <- sum((y-ybarm)^2); sst

## [1] 813643327878712262646

ssr <- sum((model1$fitted.values-ybarm)^2); ssr

## [1] 795831683490574893046

sse <- sum(model1$residuals^2); sse

## [1] 17811644388137398272

#Calculate MSR and MSE.  
msr <- ssr/2; msr

## [1] 397915841745287446528

mse <- sse/(n-3); mse

## [1] 456708830465061504

rsq <- ssr/sst ; rsq

## [1] 0.9781

cbind(ssr,sse,sst,msr,mse,rsq)

## ssr sse sst  
## [1,] 795831683490574893046 17811644388137398272 813643327878712262646  
## msr mse rsq  
## [1,] 397915841745287446528 456708830465061504 0.9781

…

#### 3.3.2.5. Confidence Interval for the Mean Response

A mean response interval is a confidence interval for the mean of all Y’s at a given X value. A prediction interval is a prediction interval for one single Y at a given X value. We predicted x value=1 for y analysis …

#We found a 95% confidence interval about the mean response for each data.  
#For Finding the value of mean oxygen purity when x0=1  
  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

new.dat <- data.frame(x=1)  
predict(model1.1, newdata = new.dat, interval = 'confidence')

## Warning: 'newdata' had 1 row but variables found have 42 rows

## fit lwr upr  
## 1 -226161218 -539326551 87004115  
## 2 -226160495 -539325803 87004813  
## 3 -226156914 -539322100 87008272  
## 4 -226149537 -539314470 87015396  
## 5 -226140389 -539305008 87024230  
## 6 -226136357 -539300838 87028124  
## 7 -226115218 -539278974 87048538  
## 8 -226083046 -539245699 87079607  
## 9 -226031355 -539192236 87129526  
## 10 -225949768 -539107852 87208316  
## 11 -225859539 -539014529 87295452  
## 12 -225660274 -538808434 87487885  
## 13 -225305792 -538441800 87830217  
## 14 -224516456 -537625410 88592498  
## 15 -223500649 -536574793 89573495  
## 16 -221805805 -534821886 91210276  
## 17 -218423880 -531324164 94476404  
## 18 -205934456 -518407826 106538913  
## 19 -184452849 -496194604 127288905  
## 20 -155094754 -465842140 155652632  
## 21 -90470772 -399051996 218110453  
## 22 36049504 -268383290 340482298  
## 23 279664404 -17141717 576470525  
## 24 633794261 347160750 920427773  
## 25 854612979 573728545 1135497414  
## 26 1243296883 971321434 1515272332  
## 27 2001529575 1741984888 2261074262  
## 28 2678564060 2423971638 2933156483  
## 29 3577946652 3320404589 3835488715  
## 30 4718925131 4442739590 4995110672  
## 31 5984513530 5671740847 6297286214  
## 32 6599870201 6264760574 6934979827  
## 33 7122377336 6766514462 7478240211  
## 34 7366356808 7000324508 7732389108  
## 35 7492748423 7121339919 7864156927  
## 36 7764664092 7381457285 8147870898  
## 37 8403426366 7991395299 8815457434  
## 38 9533930846 9067839909 10000021782  
## 39 10204589607 9705018526 10704160688  
## 40 10964914280 10426417190 11503411370  
## 41 12804194733 12168404176 13439985289  
## 42 15362341840 14586483565 16138200115

"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

new.dat <- data.frame(x=1)  
predict(model1.2, newdata = new.dat, interval = 'confidence')

## Warning: 'newdata' had 1 row but variables found have 42 rows

## fit lwr upr  
## 1 -229623102 -543246034 83999829  
## 2 -229622519 -543245430 84000392  
## 3 -229619639 -543242452 84003173  
## 4 -229611898 -543234444 84010649  
## 5 -229603671 -543225935 84018593  
## 6 -229601619 -543223812 84020575  
## 7 -229579806 -543201250 84041639  
## 8 -229546372 -543166667 84073924  
## 9 -229491312 -543109716 84127093  
## 10 -229408759 -543024328 84206810  
## 11 -229318942 -542931427 84293542  
## 12 -229108554 -542713813 84496704  
## 13 -228773245 -542366988 84820498  
## 14 -227992063 -541558981 85574855  
## 15 -226926731 -540457074 86603613  
## 16 -225219833 -538691592 88251926  
## 17 -221932178 -535291157 91426802  
## 18 -208532690 -521432830 104367451  
## 19 -186087620 -498222120 126046880  
## 20 -156322161 -467447041 154802719  
## 21 -89298907 -398174653 219576839  
## 22 39652532 -264992687 344297752  
## 23 290317835 -6485927 587121596  
## 24 663803260 377671026 949935495  
## 25 857176474 576053838 1138299109  
## 26 1245992650 973781314 1518203987  
## 27 2029124837 1769621871 2288627802  
## 28 2732236858 2477502960 2986970757  
## 29 3588490513 3330574066 3846406959  
## 30 4701687397 4425598913 4977775881  
## 31 5744955263 5439831461 6050079064  
## 32 6616221789 6280121604 6952321974  
## 33 7169201981 6811020493 7527383470  
## 34 7257425895 6895575749 7619276041  
## 35 7589810248 7213816727 7965803768  
## 36 7799614020 7414449634 8184778406  
## 37 8442070683 8027799206 8856342161  
## 38 9570581923 9102172866 10038990979  
## 39 10239228190 9737354990 10741101390  
## 40 11037433524 10494582319 11580284729  
## 41 12772302467 12137531325 13407073608  
## 42 15304095271 14530617782 16077572759

"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

new.dat <- data.frame(x=1)  
predict(model1.3, newdata = new.dat, interval = 'confidence')

## Warning: 'newdata' had 1 row but variables found have 42 rows

## fit lwr upr  
## 1 -194106114 -524268461 136056234  
## 2 -194104647 -524266941 136057648  
## 3 -194097340 -524259371 136064690  
## 4 -194091940 -524253775 136069896  
## 5 -194077917 -524239246 136083412  
## 6 -194063343 -524224145 136097460  
## 7 -194045893 -524206065 136114280  
## 8 -194020593 -524179851 136138666  
## 9 -193987101 -524145149 136170948  
## 10 -193911019 -524066320 136244282  
## 11 -193818984 -523970961 136332992  
## 12 -193679933 -523826887 136467022  
## 13 -193224712 -523355227 136905803  
## 14 -192395323 -522495889 137705243  
## 15 -191648166 -521721758 138425425  
## 16 -190025026 -520040032 139989980  
## 17 -186154956 -516030359 143720447  
## 18 -178575130 -508177448 151027189  
## 19 -162327264 -491345722 166691195  
## 20 -135270601 -463321401 192780198  
## 21 -83728587 -409952136 242494963  
## 22 29269275 -293023786 351562336  
## 23 234207455 -81232260 549647169  
## 24 483531782 175922175 791141388  
## 25 849803722 552599939 1147007505  
## 26 1236084530 948302343 1523866717  
## 27 1858170927 1581815853 2134526001  
## 28 2393129048 2122526274 2663731822  
## 29 3518670181 3246926193 3790414169  
## 30 4802876398 4508705311 5097047485  
## 31 7248983774 6866815831 7631151717  
## 32 6496413229 6145977136 6846849323  
## 33 6854069414 6488957927 7219180901  
## 34 7927915426 7514649506 8341181346  
## 35 6954815382 6585433181 7324197583  
## 36 7556901008 7160885084 7952916932  
## 37 8173164719 7748191719 8598137719  
## 38 9309365755 8827551639 9791179872  
## 39 9987832004 9470443009 10505220999  
## 40 10542748711 9995559840 11089937582  
## 41 12931016174 12250445930 13611586417  
## 42 15618587662 14781751900 16455423425

"FOR DEFENCE"

## [1] "FOR DEFENCE"

new.dat <- data.frame(x=1)  
predict(model1.4, newdata = new.dat, interval = 'confidence')

## Warning: 'newdata' had 1 row but variables found have 42 rows

## fit lwr upr  
## 1 -159399666 -899198484 580399152  
## 2 -159400493 -899199385 580398399  
## 3 -159397219 -899195819 580401381  
## 4 -159391605 -899189704 580406494  
## 5 -159381747 -899178967 580415472  
## 6 -159372487 -899168881 580423906  
## 7 -159350198 -899144603 580444208  
## 8 -159293734 -899083103 580495634  
## 9 -159246815 -899031998 580538369  
## 10 -159185600 -898965323 580594124  
## 11 -159141445 -898917230 580634340  
## 12 -159027090 -898792675 580738495  
## 13 -158850768 -898600626 580899090  
## 14 -158514962 -898234870 581204946  
## 15 -157327697 -896941732 582286337  
## 16 -155682959 -895150368 583784451  
## 17 -153826959 -893128969 585475052  
## 18 -145678885 -884255533 592897764  
## 19 -126334269 -863193773 610525235  
## 20 -104212697 -839117169 630691774  
## 21 31945577 -691132341 755023495  
## 22 327268561 -371447815 1025984937  
## 23 832573619 170977605 1494169632  
## 24 914596292 258427866 1570764719  
## 25 1383352399 754640540 2012064257  
## 26 2229723736 1633309994 2826137478  
## 27 2889742953 2301181468 3478304437  
## 28 3291559183 2699960753 3883157613  
## 29 3386489441 2793313534 3979665347  
## 30 5098602072 4425823759 5771380385  
## 31 5151889376 4475275517 5828503235  
## 32 6336080649 5558527196 7113634102  
## 33 7246700630 6375601808 8117799452  
## 34 7066702857 6214953038 7918452677  
## 35 8688332230 7650797610 9725866850  
## 36 8454172136 7444839831 9463504441  
## 37 8768682860 7721384263 9815981457  
## 38 9906703236 8717081635 11096324837  
## 39 10497005573 9231124749 11762886397  
## 40 11009542906 9676391260 12342694553  
## 41 13342551426 11694204916 14990897935  
## 42 7384001573 6497894628 8270108518

#We can find the 95% CI on the mean response as follows MULTIPLE :  
new.dat <- data.frame(x1=8, x2=15,x3=6,x4=20)  
predict(model1, newdata = new.dat, interval = 'c')

## fit lwr upr  
## 1 -110211583 -375402725 154979559

…

#### 3.3.2.6. Prediction Interval for a New Observation

Prediction intervals tell you where you can expect to see the next data point sampled. … Prediction intervals must account for both the uncertainty in estimating the population mean, plus the random variation of the individual values. So a prediction interval is always wider than a confidence interval.We predicted x value=1 for y analysis …

"FOR MULTIPLE"

## [1] "FOR MULTIPLE"

new.dat <- data.frame(x1=8,x4=20)  
  
predict(model1, interval = 'p')

## Warning in stats::predict.lm(object, ...): predictions on current data refer to \_future\_ responses

## fit lwr upr  
## 1 -110207801 -1502633487 1282217885  
## 2 -110206514 -1502632206 1282219179  
## 3 -110202977 -1502628650 1282222695  
## 4 -110195257 -1502620897 1282230382  
## 5 -110186799 -1502612378 1282238780  
## 6 -110184965 -1502610483 1282240554  
## 7 -110165231 -1502590613 1282260150  
## 8 -110143879 -1502568898 1282281140  
## 9 -110092705 -1502517441 1282332031  
## 10 -110006986 -1502431363 1282417390  
## 11 -109903122 -1502327261 1282521018  
## 12 -109680231 -1502103732 1282743270  
## 13 -109273277 -1501695827 1283149274  
## 14 -108345221 -1500766046 1284075605  
## 15 -107441701 -1499855151 1284971748  
## 16 -105804736 -1498208180 1286598707  
## 17 -101989147 -1494382383 1290404090  
## 18 -88396500 -1480743495 1303950494  
## 19 -67070076 -1459301462 1325161311  
## 20 -36261897 -1428366512 1355842719  
## 21 -2129827 -1393420217 1389160564  
## 22 53620129 -1336468190 1443708449  
## 23 185360005 -1204250256 1574970265  
## 24 628157998 -759303438 2015619434  
## 25 743436185 -644161207 2131033576  
## 26 938156970 -454398444 2330712383  
## 27 1699686941 308917487 3090456395  
## 28 2451884030 1064786299 3838981762  
## 29 3620122593 2236457080 5003788107  
## 30 4489191255 3099147734 5879234776  
## 31 6164030545 4770232052 7557829038  
## 32 6532297517 5137038585 7927556449  
## 33 6881542310 5478696637 8284387982  
## 34 7277693738 5876849941 8678537536  
## 35 6822485479 5387578695 8257392262  
## 36 7276918717 5855604234 8698233199  
## 37 8011470001 6590912495 9432027507  
## 38 9087601733 7652215547 10522987919  
## 39 9758668417 8315582415 11201754418  
## 40 10579798854 9130816864 12028780843  
## 41 12145182429 10648866079 13641498780  
## 42 17866784992 15965078729 19768491256

#KISA KOD#  
  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

new.dat <- data.frame(x=1)  
predict(model1.1, newdata = new.dat, interval = 'prediction')

## Warning: 'newdata' had 1 row but variables found have 42 rows

## fit lwr upr  
## 1 -226161218 -1903727207 1451404770  
## 2 -226160495 -1903726479 1451405489  
## 3 -226156914 -1903722875 1451409047  
## 4 -226149537 -1903715451 1451416377  
## 5 -226140389 -1903706244 1451425466  
## 6 -226136357 -1903702186 1451429472  
## 7 -226115218 -1903680912 1451450476  
## 8 -226083046 -1903648534 1451482442  
## 9 -226031355 -1903596512 1451533802  
## 10 -225949768 -1903514403 1451614867  
## 11 -225859539 -1903423596 1451704519  
## 12 -225660274 -1903223057 1451902508  
## 13 -225305792 -1902866306 1452254723  
## 14 -224516456 -1902071920 1453039009  
## 15 -223500649 -1901049617 1454048318  
## 16 -221805805 -1899343938 1455732328  
## 17 -218423880 -1895940410 1459092650  
## 18 -205934456 -1883371408 1471502495  
## 19 -184452849 -1861753670 1492847971  
## 20 -155094754 -1832211046 1522021538  
## 21 -90470772 -1767187054 1586245511  
## 22 36049504 -1639908265 1712007274  
## 23 279664404 -1394924797 1954253605  
## 24 633794261 -1039021894 2306610417  
## 25 854612979 -817227678 2526453637  
## 26 1243296883 -427070072 2913663838  
## 27 2001529575 333141561 3669917589  
## 28 2678564060 1010939276 4346188845  
## 29 3577946652 1909869006 5246024298  
## 30 4718925131 3047867510 6389982752  
## 31 5984513530 4307020797 7662006264  
## 32 6599870201 4918069504 8281670897  
## 33 7122377336 5436318765 8808435908  
## 34 7366356808 5678122593 9054591023  
## 35 7492748423 5803340421 9182156425  
## 36 7764664092 6072623142 9456705042  
## 37 8403426366 6704625401 10102227332  
## 38 9533930846 7821214909 11246646782  
## 39 10204589607 8482461161 11926718053  
## 40 10964914280 9231093575 12698734984  
## 41 12804194733 11037733652 14570655814  
## 42 15362341840 13540773510 17183910171

"FOR DEFENCE"

## [1] "FOR DEFENCE"

new.dat <- data.frame(x=1)  
predict(model1.4, newdata = new.dat, interval = 'prediction')

## Warning: 'newdata' had 1 row but variables found have 42 rows

## fit lwr upr  
## 1 -159399666 -4044792194 3725992862  
## 2 -159400493 -4044793035 3725992049  
## 3 -159397219 -4044789706 3725995267  
## 4 -159391605 -4044783996 3726000787  
## 5 -159381747 -4044773971 3726010476  
## 6 -159372487 -4044764554 3726019579  
## 7 -159350198 -4044741885 3726041490  
## 8 -159293734 -4044684463 3726096995  
## 9 -159246815 -4044636747 3726143118  
## 10 -159185600 -4044574492 3726203293  
## 11 -159141445 -4044529588 3726246697  
## 12 -159027090 -4044413290 3726359111  
## 13 -158850768 -4044233974 3726532439  
## 14 -158514962 -4043892466 3726862542  
## 15 -157327697 -4042685046 3728029651  
## 16 -155682959 -4041012398 3729646481  
## 17 -153826959 -4039124923 3731471005  
## 18 -145678885 -4030838891 3739481121  
## 19 -126334269 -4011168208 3758499670  
## 20 -104212697 -3988676288 3780250893  
## 21 31945577 -3850297909 3914189063  
## 22 327268561 -3550511394 4205048516  
## 23 832573619 -3038690011 4703837248  
## 24 914596292 -2955743462 4784936046  
## 25 1383352399 -2482427136 5249131933  
## 26 2229723736 -1630934518 6090381991  
## 27 2889742953 -969710043 6749195948  
## 28 3291559183 -568358110 7151476476  
## 29 3386489441 -473669942 7246648824  
## 30 5098602072 1225411725 8971792420  
## 31 5151889376 1278030948 9025747804  
## 32 6336080649 2443322992 10228838306  
## 33 7246700630 3334184053 11159217207  
## 34 7066702857 3158448700 10974957015  
## 35 8688332230 4735428121 12641236339  
## 36 8454172136 4508576552 12399767719  
## 37 8768682860 4813204740 12724160980  
## 38 9906703236 5911184201 13902222271  
## 39 10497005573 6478121764 14515889381  
## 40 11009542906 6968965457 15050120355  
## 41 13342551426 9187311478 17497791373  
## 42 7384001573 3468116191 11299886955

…

#### 3.3.2.7. Analysis of Variance

A one-way ANOVA is used for three or more groups of data, to gain information about the relationship between the dependent and independent variables.In general, variances tests assess the variability of the data in multiple groups to determine whether they are different. F tests are anaylzed for ths purpose …

"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

anova(model1.1)

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$CE 1 787045294150890029046 787045294150890029046 1184  
## Residuals 40 26598033727822225408 664950843195555584   
## Pr(>F)   
## data\_22$CE <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

anova(model1.2)

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$PE 1 786988380344163106806 786988380344163106806 1181  
## Residuals 40 26654947534549295104 666373688363732352   
## Pr(>F)   
## data\_22$PE <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

anova(model1.3)

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$OCE 1 783911783171596746722 783911783171596746722 1055  
## Residuals 40 29731544707115229184 743288617677880704   
## Pr(>F)   
## data\_22$OCE <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR DEFENCE"

## [1] "FOR DEFENCE"

anova(model1.4)

## Analysis of Variance Table  
##   
## Response: data\_22$GNP  
## Df Sum Sq Mean Sq F value  
## data\_22$D 1 671172057876774387722 671172057876774387722 188  
## Residuals 40 142471270001937809418 3561781750048445440   
## Pr(>F)   
## data\_22$D <0.0000000000000002 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

"FOR MULTIPLE"

## [1] "FOR MULTIPLE"

summary(model1); anova(model1)

##   
## Call:  
## lm(formula = y ~ x1 + x4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1628158802 34647222 110187782 110219462 2406578750   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -110211313.75 131108104.95 -0.84 0.41   
## x1 8.92 0.54 16.52 < 0.0000000000000002 \*\*\*  
## x4 -17.04 3.88 -4.39 0.000085 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 676000000 on 39 degrees of freedom  
## Multiple R-squared: 0.978, Adjusted R-squared: 0.977   
## F-statistic: 871 on 2 and 39 DF, p-value: <0.0000000000000002

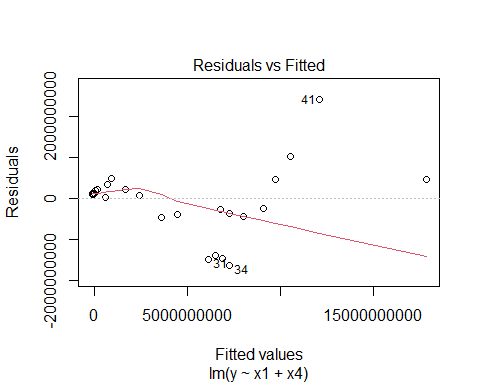
## Analysis of Variance Table  
##   
## Response: y  
## Df Sum Sq Mean Sq F value  
## x1 1 787045294150890029046 787045294150890029046 1723.3  
## x4 1 8786389339684835328 8786389339684835328 19.2  
## Residuals 39 17811644388137398272 456708830465061504   
## Pr(>F)   
## x1 < 0.0000000000000002 \*\*\*  
## x4 0.000085 \*\*\*  
## Residuals   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

…

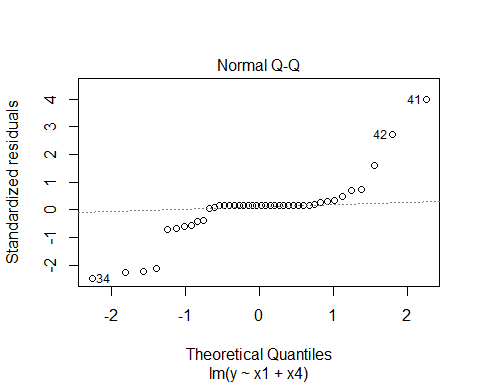
#### 3.3.2.8. Adequacy of the Regression Model

This model quantifies the percentage of the original uncertainty in the data that is explained by the straight line model.Residuals versus Fitted plot and the Normal Q-Q plot was shown. When look at Residuals versus Fitted, it is understood that model is inadequacy and linear relationship could not cover the information in the GNP variable. In addition, it is observed that normality cannot be achieved from the Normal Q-Q plot and there are also some outliers. They are put on report and interpreted …

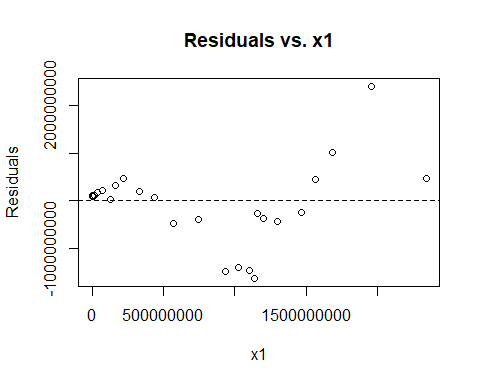
plot(model1,1)



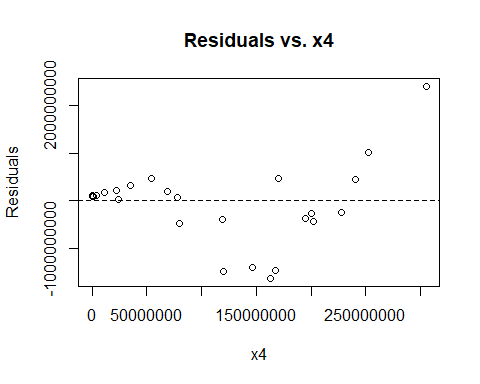
plot(model1,2)



#We can also plot residuals vs. x1 and residuals vs. x4 as follows:  
plot(x1, resid(model1), ylab="Residuals", xlab="x1", main="Residuals vs. x1")   
abline(0, 0, lty=2)



plot(x4, resid(model1), ylab="Residuals", xlab="x4", main="Residuals vs. x4")   
abline(0, 0, lty=2)

  
…

#### 3.3.2.9. Residuals Analysis

Residuals analysis was done in the Adequacy of the Regression Model part. Residual analysis graphs such as Residual versus x1(Current Expenditures), Residuals versus x4(Defence) and also interpretation about the Residuals analysis can be seen in the Adequacy of the Regression Model part …

"ABOVE "

## [1] "ABOVE "

…

## 3.4. Trend Based Forecasting

Forecast analysis for Budget Expenditures has been made. Since the data is annual, it cannot be observed any seasonality. Only trend-based forecasting can be applied. Quadratic Trend and Linear Trends are applied and comparision of two are interpreted with R squared. All of Data Quadratic method is more useful than linear trend because of adjusted R squared are high …

require(forecast)

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

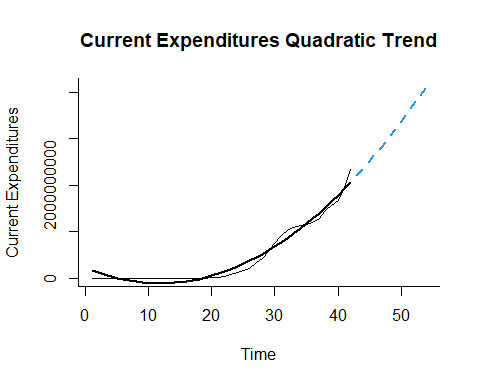
#Write your codes here. Their results will appear after knitting the markdown file.  
#Quadratic Trend  
"FOR CURRENT EXPENDITURES"

## [1] "FOR CURRENT EXPENDITURES"

ce.ts <- ts(data\_22$CE, start = c(1), end = c(42), freq = 1)  
ce.lm <- tslm(ce.ts ~ trend + I(trend^2))  
summary(ce.lm)

##   
## Call:  
## tslm(formula = ce.ts ~ trend + I(trend^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -166088625 -89048609 -5932131 80386057 276099850   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 219385166 51784597 4.24 0.00013 \*\*\*  
## trend -55667315 5554495 -10.02 0.0000000000024 \*\*\*  
## I(trend^2) 2371442 125261 18.93 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 107000000 on 39 degrees of freedom  
## Multiple R-squared: 0.975, Adjusted R-squared: 0.974   
## F-statistic: 762 on 2 and 39 DF, p-value: <0.0000000000000002

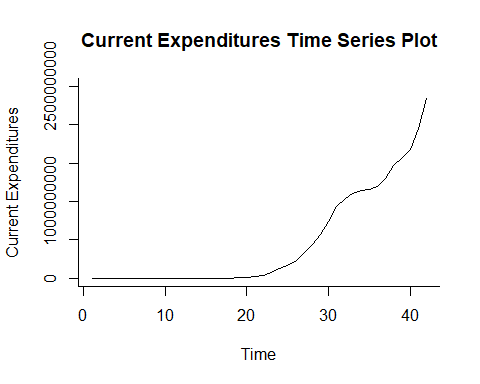
ce.lm.pred <- forecast(ce.lm, h = 12, level = 0)  
plot(ce.lm.pred, ylab = "Current Expenditures", xlab = "Time", bty = "l",  
 main = "Current Expenditures Quadratic Trend", flty = 2)  
lines(ce.lm$fitted, lwd = 2)



"LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

## [1] "LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

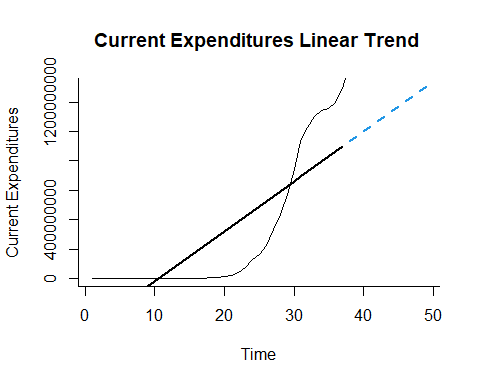
ce.ts <- ts(data\_22$CE, start = c(1), end = c(42))  
plot(ce.ts, xlab = "Time", ylab = "Current Expenditures", main="Current Expenditures Time Series Plot",ylim = c(500, 2500000000), bty = "l")



cetrain.ts <- window(ce.ts, start = 1, end = 37)  
cevalid.ts <- window(ce.ts, start = 38, end = 42)  
  
cetrain.lm <- tslm(cetrain.ts ~ trend)  
summary(cetrain.lm)

##   
## Call:  
## tslm(formula = cetrain.ts ~ trend)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -349415711 -224318073 -14681611 237722006 396898995   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -359333293 85243321 -4.22 0.00017 \*\*\*  
## trend 34005534 3911232 8.69 0.00000000029 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 254000000 on 35 degrees of freedom  
## Multiple R-squared: 0.684, Adjusted R-squared: 0.674   
## F-statistic: 75.6 on 1 and 35 DF, p-value: 0.000000000289

cetrain.lm.pred <- forecast(cetrain.lm, h = 12, level = 0)  
plot(cetrain.lm.pred, ylab = "Current Expenditures", xlab = "Time", bty = "l",  
 main = "Current Expenditures Linear Trend ", flty = 2)  
lines(cetrain.lm$fitted, lwd = 2)  
lines(ce.ts)



accuracy(cetrain.lm.pred$mean,cevalid.ts)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 800987695 845168379 800987695 43.3 43.3 0.3321 3.778

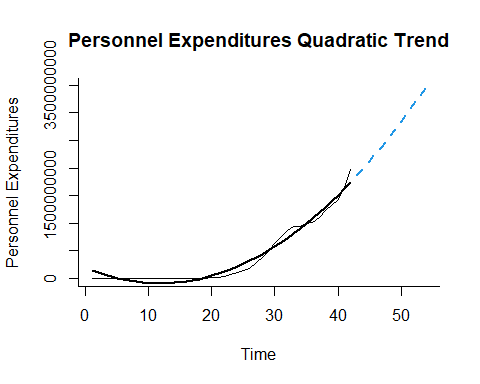
"FOR PERSONNEL EXPENDITURES"

## [1] "FOR PERSONNEL EXPENDITURES"

pe.ts <- ts(data\_22$PE, start = c(1), end = c(42), freq = 1)  
pe.lm <- tslm(pe.ts ~ trend + I(trend^2))  
summary(pe.lm)

##   
## Call:  
## tslm(formula = pe.ts ~ trend + I(trend^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -139983425 -71281555 -1405392 67621806 224446987   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 184900043 42354437 4.37 0.000090450713 \*\*\*  
## trend -46915335 4543002 -10.33 0.000000000001 \*\*\*  
## I(trend^2) 1999271 102451 19.51 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 87200000 on 39 degrees of freedom  
## Multiple R-squared: 0.976, Adjusted R-squared: 0.975   
## F-statistic: 810 on 2 and 39 DF, p-value: <0.0000000000000002

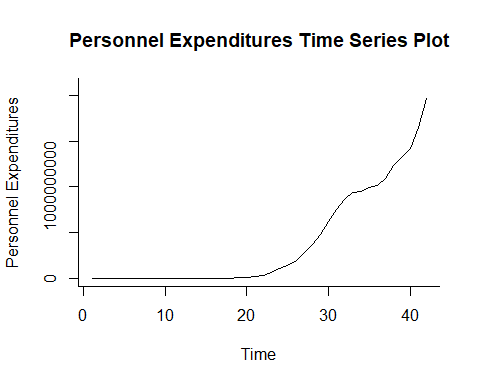
pe.lm.pred <- forecast(pe.lm, h = 12, level = 0)  
plot(pe.lm.pred, ylab = "Personnel Expenditures", xlab = "Time", bty = "l",  
 main = "Personnel Expenditures Quadratic Trend", flty = 2)  
lines(pe.lm$fitted, lwd = 2)



"LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

## [1] "LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

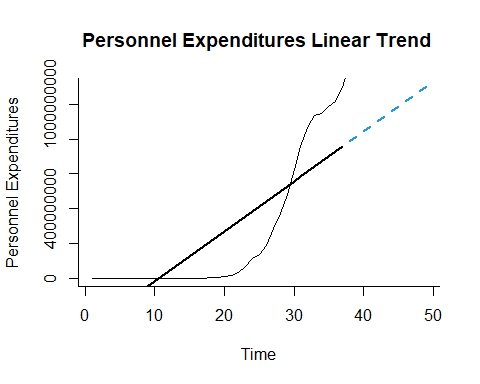
pe.ts <- ts(data\_22$PE, start = c(1), end = c(42))  
plot(pe.ts, xlab = "Time", ylab = "Personnel Expenditures", main="Personnel Expenditures Time Series Plot",ylim = c(500, 2100000000), bty = "l")



petrain.ts <- window(pe.ts, start = 1, end = 37)  
pevalid.ts <- window(pe.ts, start = 38, end = 42)  
  
petrain.lm <- tslm(petrain.ts ~ trend)  
summary(petrain.lm)

##   
## Call:  
## tslm(formula = petrain.ts ~ trend)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -293880164 -185519193 -12470346 188230375 339197940   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -302935713 71630623 -4.23 0.00016 \*\*\*  
## trend 28676827 3286638 8.73 0.00000000026 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 213000000 on 35 degrees of freedom  
## Multiple R-squared: 0.685, Adjusted R-squared: 0.676   
## F-statistic: 76.1 on 1 and 35 DF, p-value: 0.000000000265

petrain.lm.pred <- forecast(petrain.lm, h = 12, level = 0)  
plot(petrain.lm.pred, ylab = "Personnel Expenditures", xlab = "Time", bty = "l",  
 main = "Personnel Expenditures Linear Trend ", flty = 2)  
lines(petrain.lm$fitted, lwd = 2)  
lines(pe.ts)



accuracy(petrain.lm.pred$mean,pevalid.ts)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 676142782 711630563 676142782 43.38 43.38 0.3298 3.858

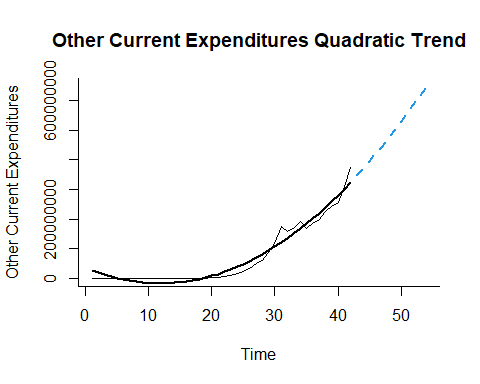
"FOR OTHER CURRENT EXPENDITURES"

## [1] "FOR OTHER CURRENT EXPENDITURES"

oce.ts <- ts(data\_22$OCE, start = c(1), end = c(42), freq = 1)  
oce.lm <- tslm(oce.ts ~ trend + I(trend^2))  
summary(oce.lm)

##   
## Call:  
## tslm(formula = oce.ts ~ trend + I(trend^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -26105200 -17855663 -913152 13883393 55714053   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 34485123 9928649 3.47 0.0013 \*\*   
## trend -8751980 1064962 -8.22 0.00000000049 \*\*\*  
## I(trend^2) 372172 24016 15.50 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 20400000 on 39 degrees of freedom  
## Multiple R-squared: 0.963, Adjusted R-squared: 0.961   
## F-statistic: 509 on 2 and 39 DF, p-value: <0.0000000000000002

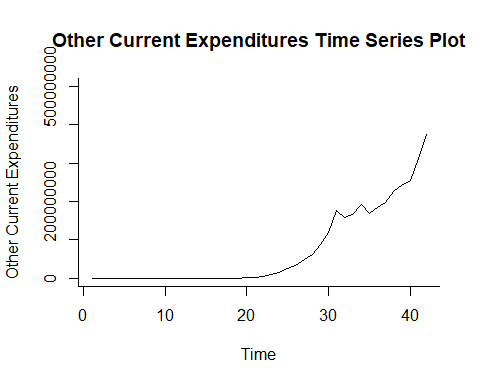
oce.lm.pred <- forecast(oce.lm, h = 12, level = 0)  
plot(oce.lm.pred, ylab = "Other Current Expenditures", xlab = "Time", bty = "l",  
 main = "Other Current Expenditures Quadratic Trend", flty = 2)  
lines(oce.lm$fitted, lwd = 2)



"LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

## [1] "LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

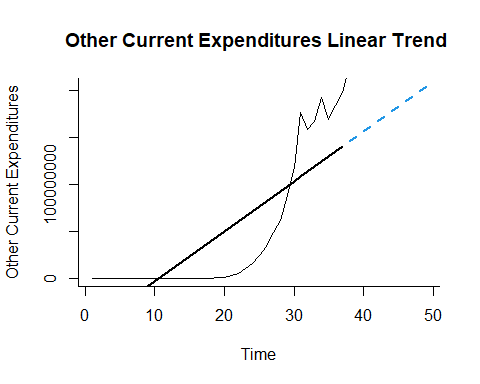
oce.ts <- ts(data\_22$OCE, start = c(1), end = c(42))  
plot(oce.ts, xlab = "Time", ylab = "Other Current Expenditures", main="Other Current Expenditures Time Series Plot",ylim = c(100, 500000000), bty = "l")



ocetrain.ts <- window(oce.ts, start = 1, end = 37)  
ocevalid.ts <- window(oce.ts, start = 38, end = 42)  
  
ocetrain.lm <- tslm(ocetrain.ts ~ trend)  
summary(ocetrain.lm)

##   
## Call:  
## tslm(formula = ocetrain.ts ~ trend)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -56003271 -38798880 -2211265 39460119 67870042   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -56397580 13809229 -4.08 0.00024 \*\*\*  
## trend 5328706 633611 8.41 0.00000000064 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 41200000 on 35 degrees of freedom  
## Multiple R-squared: 0.669, Adjusted R-squared: 0.66   
## F-statistic: 70.7 on 1 and 35 DF, p-value: 0.00000000064

ocetrain.lm.pred <- forecast(ocetrain.lm, h = 12, level = 0)  
plot(ocetrain.lm.pred, ylab = "Other Current Expenditures", xlab = "Time", bty = "l",  
 main = "Other Current Expenditures Linear Trend ", flty = 2)  
lines(ocetrain.lm$fitted, lwd = 2)  
lines(oce.ts)



accuracy(ocetrain.lm.pred$mean,ocevalid.ts)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 124844912 133708414 124844912 42.85 42.85 0.3387 3.344

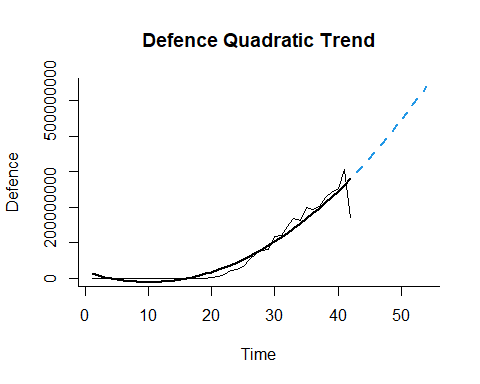
"FOR DEFENCE"

## [1] "FOR DEFENCE"

d.ts <- ts(data\_22$D, start = c(1), end = c(42), freq = 1)  
d.lm <- tslm(d.ts ~ trend + I(trend^2))  
summary(d.lm)

##   
## Call:  
## tslm(formula = d.ts ~ trend + I(trend^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -110086618 -8071169 4307412 10228024 42508025   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17971342 11161067 1.61 0.12   
## trend -5696346 1197153 -4.76 0.00002677447220 \*\*\*  
## I(trend^2) 284502 26997 10.54 0.00000000000057 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23000000 on 39 degrees of freedom  
## Multiple R-squared: 0.94, Adjusted R-squared: 0.937   
## F-statistic: 305 on 2 and 39 DF, p-value: <0.0000000000000002

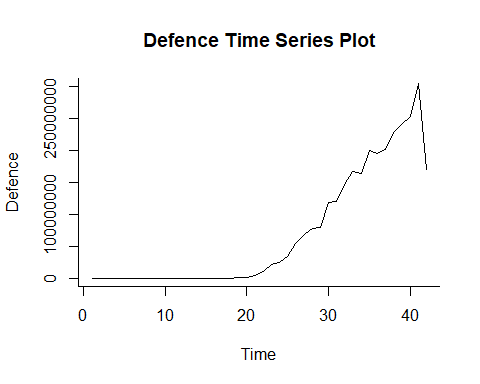
d.lm.pred <- forecast(d.lm, h = 12, level = 0)  
plot(d.lm.pred, ylab = "Defence", xlab = "Time", bty = "l",main = "Defence Quadratic Trend", flty = 2)  
lines(d.lm$fitted, lwd = 2)



"LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

## [1] "LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

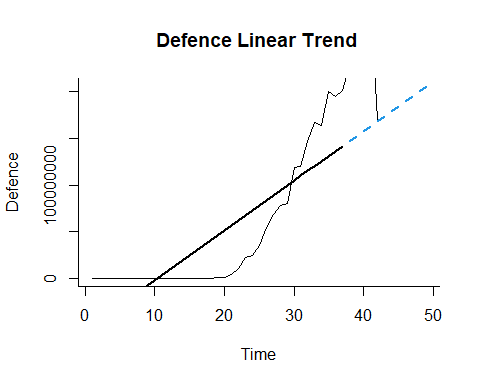
d.ts <- ts(data\_22$D, start = c(1), end = c(42))  
plot(d.ts, xlab = "Time", ylab = "Defence", main="Defence Time Series Plot",ylim = c(100, 300000000), bty = "l")



dtrain.ts <- window(d.ts, start = 1, end = 37)  
dvalid.ts <- window(d.ts, start = 38, end = 42)  
  
dtrain.lm <- tslm(dtrain.ts ~ trend)  
summary(dtrain.lm)

##   
## Call:  
## tslm(formula = dtrain.ts ~ trend)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -52106422 -29692341 -3115581 31741139 68914646   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -55519287 12510460 -4.44 0.000086478335 \*\*\*  
## trend 5330986 574019 9.29 0.000000000057 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 37300000 on 35 degrees of freedom  
## Multiple R-squared: 0.711, Adjusted R-squared: 0.703   
## F-statistic: 86.3 on 1 and 35 DF, p-value: 0.0000000000566

dtrain.lm.pred <- forecast(dtrain.lm, h = 12, level = 0)  
plot(dtrain.lm.pred, ylab = "Defence", xlab = "Time", bty = "l",  
 main = "Defence Linear Trend ", flty = 2)  
lines(dtrain.lm$fitted, lwd = 2)  
lines(d.ts)



accuracy(dtrain.lm.pred$mean,dvalid.ts)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 81579451 93249849 81579451 31.49 31.49 -0.3855 1.597

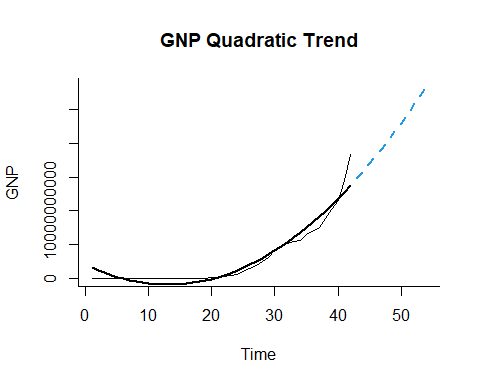
"FOR GNP"

## [1] "FOR GNP"

g.ts <- ts(data\_22$GNP, start = c(1), end = c(42), freq = 1)  
g.lm <- tslm(g.ts ~ trend + I(trend^2))  
summary(g.lm)

##   
## Call:  
## tslm(formula = g.ts ~ trend + I(trend^2))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1581748970 -580246715 -188428931 632559395 4568965913   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2020432936 530384790 3.81 0.00048 \*\*\*  
## trend -456200462 56889884 -8.02 0.00000000089 \*\*\*  
## I(trend^2) 17520306 1282940 13.66 < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1090000000 on 39 degrees of freedom  
## Multiple R-squared: 0.943, Adjusted R-squared: 0.94   
## F-statistic: 322 on 2 and 39 DF, p-value: <0.0000000000000002

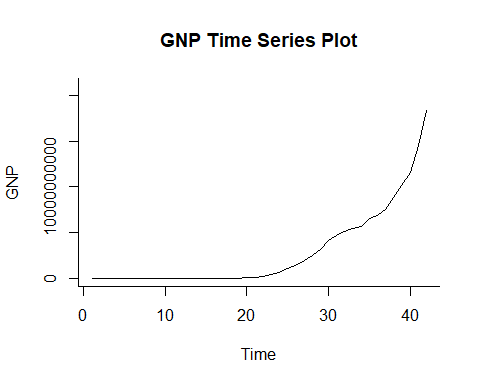
g.lm.pred <- forecast(g.lm, h = 12, level = 0)  
plot(g.lm.pred, ylab = "GNP", xlab = "Time", bty = "l",main = "GNP Quadratic Trend", flty = 2)  
lines(g.lm$fitted, lwd = 2)



"LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

## [1] "LINEAR TREND WITH ASSUMPTION ABOUT TRAINING AND VALIDATION"

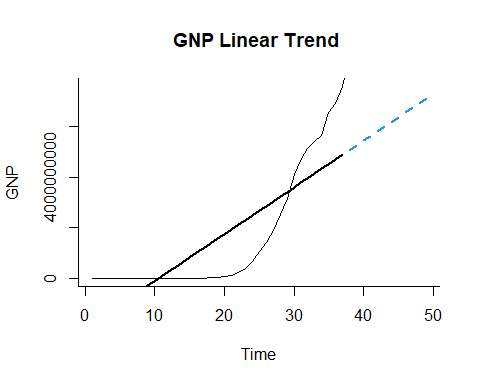
g.ts <- ts(data\_22$GNP, start = c(1), end = c(42))  
plot(g.ts, xlab = "Time", ylab = "GNP", main="GNP Time Series Plot",ylim = c(3500, 21000000000), bty = "l")



gtrain.ts <- window(g.ts, start = 1, end = 37)  
gvalid.ts <- window(g.ts, start = 38, end = 42)  
  
gtrain.lm <- tslm(gtrain.ts ~ trend)  
summary(gtrain.lm)

##   
## Call:  
## tslm(formula = gtrain.ts ~ trend)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1888938877 -1181124492 -79833674 1170411078 2698194439   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1951105935 455950685 -4.28 0.00014 \*\*\*  
## trend 184657156 20920452 8.83 0.0000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1360000000 on 35 degrees of freedom  
## Multiple R-squared: 0.69, Adjusted R-squared: 0.681   
## F-statistic: 77.9 on 1 and 35 DF, p-value: 0.0000000002

gtrain.lm.pred <- forecast(gtrain.lm, h = 12, level = 0)  
plot(gtrain.lm.pred, ylab = "GNP", xlab = "Time", bty = "l",  
 main = "GNP Linear Trend ", flty = 2)  
lines(gtrain.lm$fitted, lwd = 2)  
lines(g.ts)



accuracy(gtrain.lm.pred$mean,gvalid.ts)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 7273447880 7920995491 7273447880 54.83 54.83 0.3564 3.395

…