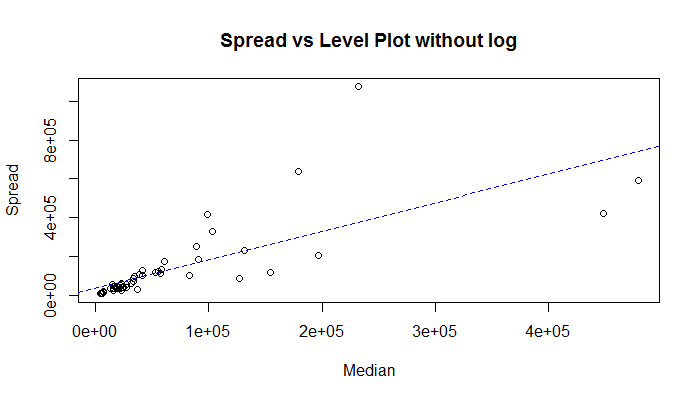
STAT S 670 – Exploratory Data Analysis – Homework #3

Ganesh Nagarajan

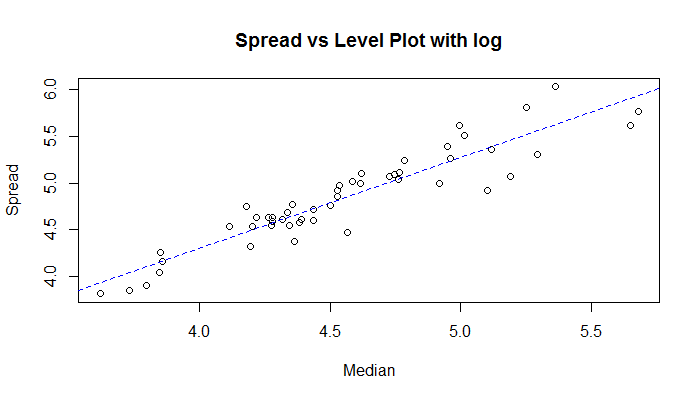
[gnagaraj@indian.edu](mailto:gnagaraj@indian.edu)

Solutions

1. a) Level VS Spread plot without any transformation



Level Vs Spread Plot with log transformation

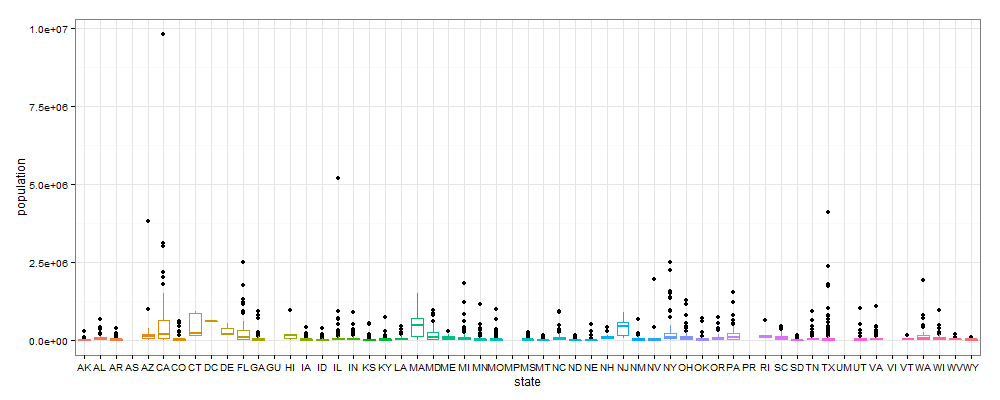


b) From the R code, 1-slope is 0.02806874, slope=0.9719

lm equation log(df) = 0.9719\*log(m)+0.028

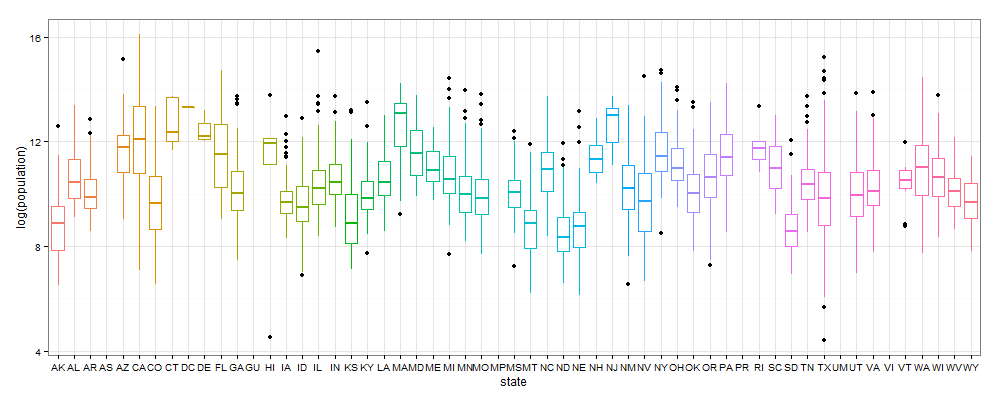
Hence from p formula, the most appropriate transformation would be log transformation.

c) Box plot without log transformation,



It can be seen that there are lot of outliers and outliers distort the interpretation of the box plot.

Hence as suggested by the 1-p rule, following is the box plot with log transformation applied.



A clear visual comparison from the box plot with and without transformation supports the effectiveness of the transformation. It can be clearly seen that the box plot with log transformation has lesser outlier effects and better interpretable.

d) Transformation for symmetry table:

Since the transformation of California subset comes under transformation of data from multiple batches, this becomes a problem for transformation of symmetry.

Also, since the transformation is for a single batch, the estimate of p is calculated from the slope of the lm fit line to the x and y axis columns in symmetry table.

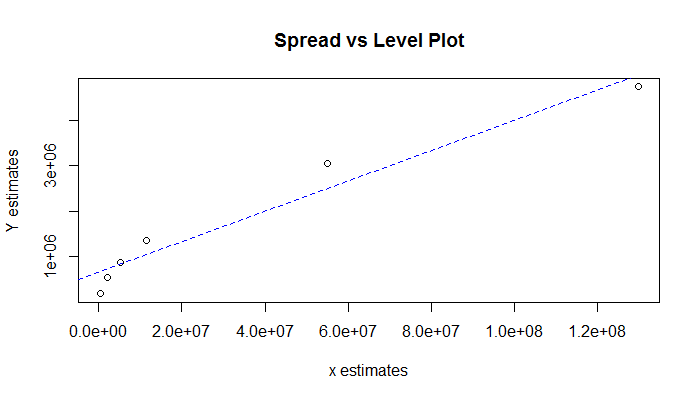
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Depth | XL | XU | Mid Summary | Spread |  |  | P estimate |
| F 15.0 | 45578 | 685306 | 365442 | 639728 | 382440.7 | 186301.5 | 0.5128617 |
| E 8.0 | 20007 | 1418788 | 719397.5 | 1398781 | 2179922.2 | 540257 | 0.7521668 |
| D 4.5 | 13994 | 2112426 | 1063209.8 | 2098432 | 5254066.3 | 884069.2 | 0.8317362 |
| C 2.5 | 6463 | 3052773 | 1529617.8 | 3046310 | 11565752 | 1350477.2 | 0.8832348 |
| B 1.5 | 2207.5 | 6456959 | 3229583.2 | 6454752 | 55043820.9 | 3050442.8 | 0.9445816 |
| A 1.0 | 1175 | 9818605 | 4909890 | 9817430 | 129717941.5 | 4730749.5 | 0.9635305 |

Also from the R code, (Via Linear modeling)

[1] "The power is 0.966537662120274"

[1] "The slope is 0.0334623378797263"

e) The median of the p-estimate is also 0.857485 which suggests that there is no need for any transformation. Also the spread vs level plot for the above table is as follows,



For the entire dataset, generating the p-estimate by considering it to be one single batch,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Depth | XL | XU | Mid Summary | Spread |  |  | P estimate |
| 786.5 | 11104.5 | 66699 | 38901.75 | 55594.5 | 18232.06 | 13044.75 | 0.2845157 |
| 393.5 | 6157 | 157906.5 | 82031.75 | 151749.5 | 172343.66 | 56174.75 | 0.6740539 |
| 197 | 3423 | 321520 | 162471.5 | 318097 | 850058.92 | 136614.5 | 0.8392882 |
| 99 | 2071 | 622263 | 312167 | 620192 | 3444578.75 | 286310 | 0.916881 |
| 50 | 1321 | 919040 | 460180.5 | 917719 | 7719165.86 | 434323.5 | 0.9437344 |
| 25.5 | 813 | 1401948 | 701380.5 | 1401135 | 18314708.22 | 675523.5 | 0.9631158 |
| 13 | 662 | 1951269 | 975965.5 | 1950607 | 35849539.37 | 950108.5 | 0.9734973 |
| 7 | 494 | 2504700 | 1252597 | 2504206 | 59416269.29 | 1226740 | 0.9793535 |
| 4 | 416 | 3817117 | 1908766.5 | 3816701 | 138978803 | 1882909.5 | 0.9864518 |
| 2.5 | 188 | 4643567 | 2321877.5 | 4643379 | 206171486.9 | 2296020.5 | 0.9888635 |
| 1.5 | 86 | 7506640 | 3753363 | 7506554 | 541079576.5 | 3727506 | 0.993111 |
| 1 | 82 | 9818605 | 4909343.5 | 9818523 | 927201316.3 | 4883486.5 | 0.9947331 |

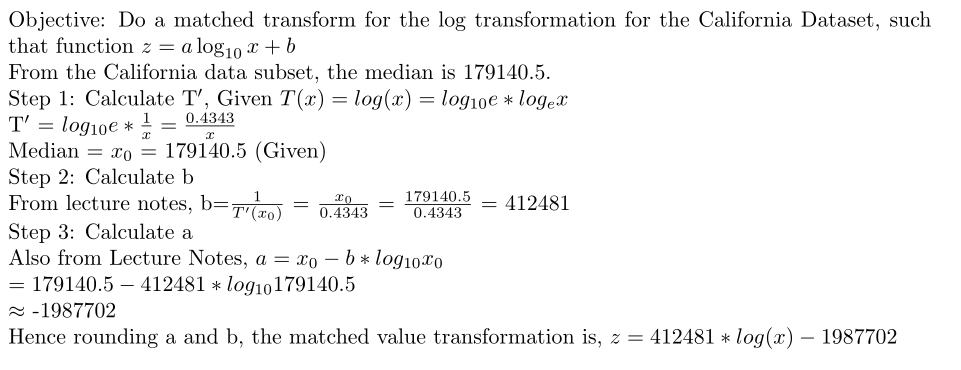
From the above table median p estimate is 0.9683065.

Hence from the above value it can be inferred that when considering the dataset as a set of different batches, the suggested transformation is log transform as shown in 1(a). However when we consider the entire dataset to be in one single batch, the p estimate comes to 0.9683. Hence when rounded to nearest integer, it is one and no transformation is suggested.

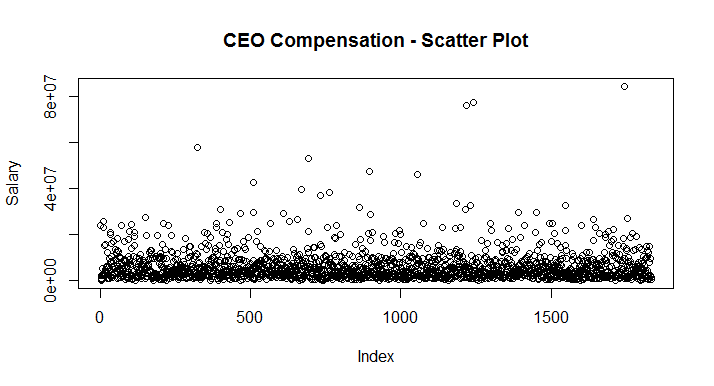
This seems to be intuitive considering the fact that when we separate data into batches, we would have to handle more outliers than when we have one single batch data.

From previous lecture, outliers are given by 0.4+0.07n for one single batch, considering additional 50 batches, outliers become high and requires more stringent transformation techniques.

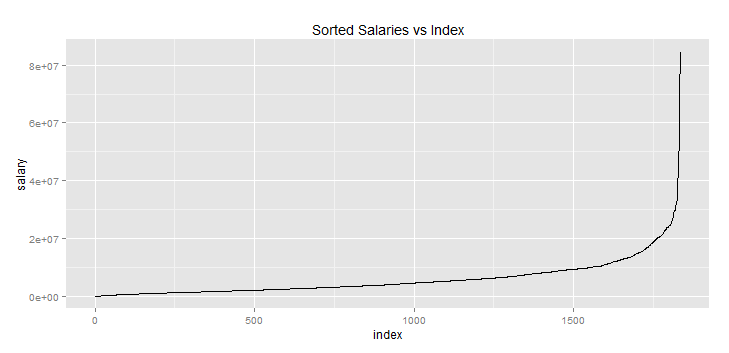
f) Find a and b of the matched transform:



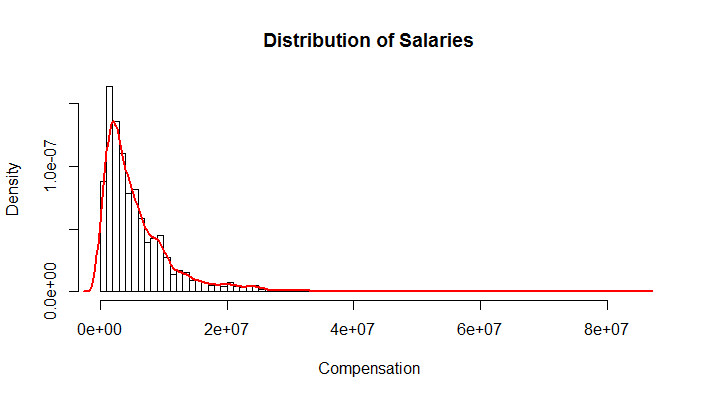
1. CEO Compensation Dataset
2. There are 1835 rows in the dataset. The highest CEO is paid 84515000. The unusual things is that dataset is that,
   1. 8 CEOs draw zero salary
   2. The data is dense in the sorted lower value segments and sparse in the higher end.



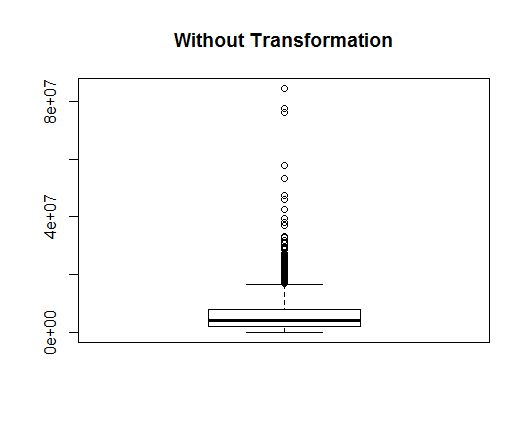
1. Following is the sorted salaries plotted by position, it can be seen that the linear is linear for lower and middle salaries, however it turns exponential!



With respect to distribution, the distribution is skewed right as shown below.



Following with our discussion the data symmetry, following is the box spot

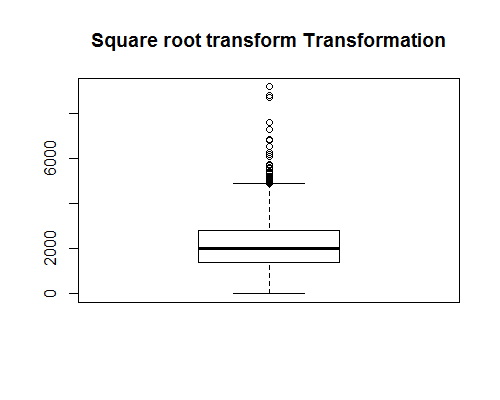


It can be seen that outliers present in the data distort the data representation, hence requires a transformation for readability and further data processing.

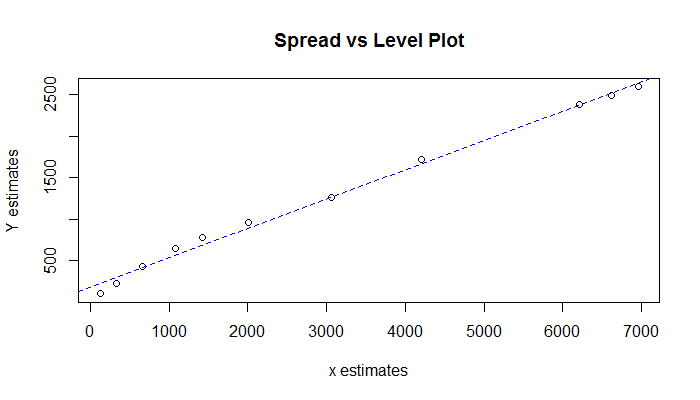
1. Following is the transform for symmetry table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Depth | XL | XU | Mid Summary | Spread |  |  | P estimate |
| F | 1987500 | 7857000 | 4922250 | 5869500 | 1177155 | 911250 | 0.2258876 |
| E | 1219000 | 11190000 | 6204500 | 9971000 | 3698162 | 2193500 | 0.4068675 |
| D | 754500 | 16102500 | 8428500 | 15348000 | 9773695 | 4417500 | 0.5480215 |
| C | 459000 | 21370000 | 10914500 | 20911000 | 19568162 | 6903500 | 0.6472075 |
| B | 279500 | 25320500 | 12800000 | 25041000 | 29170960 | 8789000 | 0.6987072 |
| A | 86000 | 31719000 | 15902500 | 31633000 | 48811948 | 11891500 | 0.7563814 |
| Z | 0 | 42589000 | 21294500 | 42589000 | 93764037 | 17283500 | 0.8156703 |
| Y | 0 | 55502500 | 27751250 | 55502500 | 166259206 | 23740250 | 0.8572094 |
| X | 0 | 76831500 | 38415750 | 76831500 | 331520403 | 34404750 | 0.8962213 |
| W | 0 | 81035500 | 40517750 | 81035500 | 370784201 | 36506750 | 0.9015418 |
| V | 0 | 84515000 | 42257500 | 84515000 | 404947777 | 38246500 | 0.905552 |

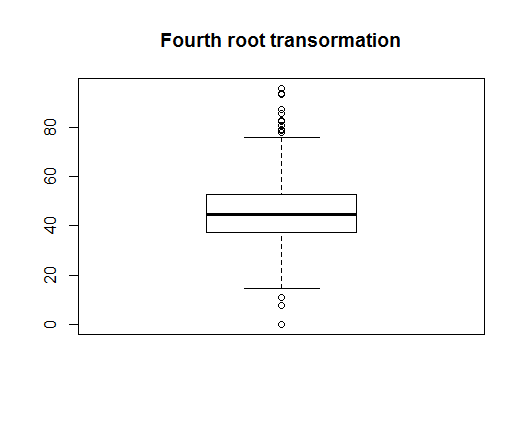
The median of the estimated p values is 0.75, hence plot square root transformation



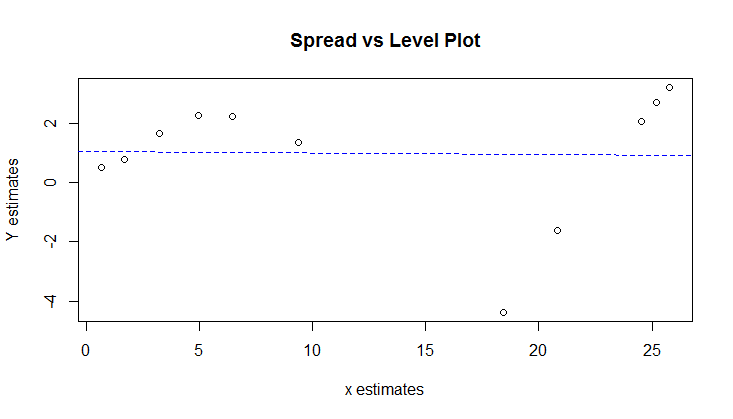
The Symmetry plot is,



Since the diagnostic plot did not give any optimal result, Consider another transform, fourth root transform, the box plot and symmetry plot is as follows,



The symmetry plot is as follows,

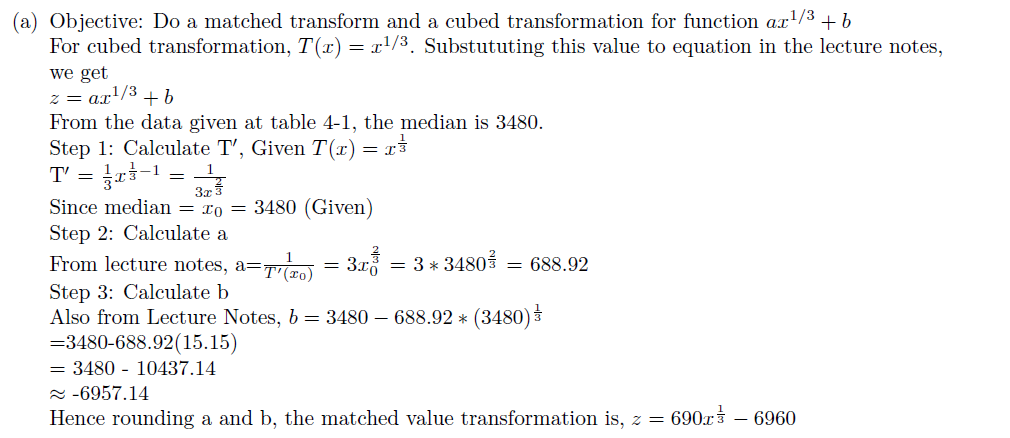


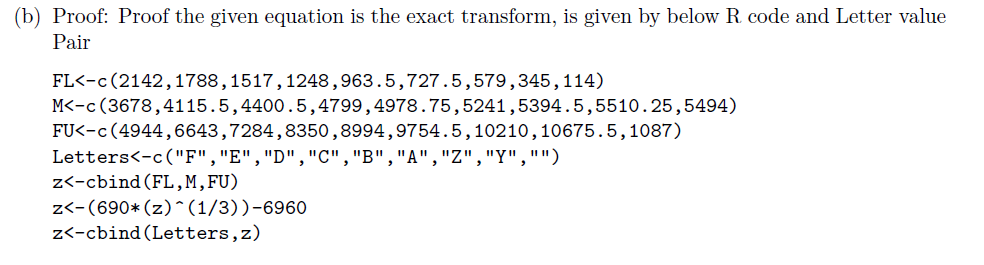
As seen from the above plot, since fourth root creates a straight line slope, as per the discussion in class, is more a proper transformation than the square root transform.

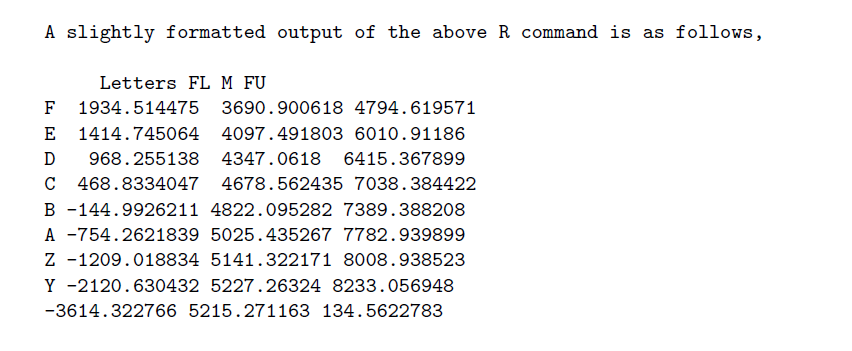
1. Since we cannot delete a data point, we cannot remove irregular values. However as shown above a proper transformation plot hugely increases the readability of the plot.
2. The two transformations are square root transformation and the fourth root transformation are already discussed in section 2.c
3. If one has to choose between the above two, a more apt one would be the fourth root. This can verified by plotting the transform plot for the fourth root of the dataset. We get a straight line, implying the transformation is correct.

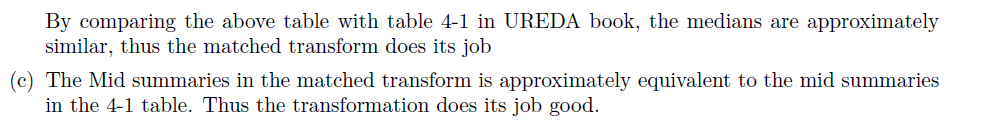
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Depth | Lower | Upper | Mid | Spread | xaxis | yaxis | pestimate |
| F | 459.5 | 37.5471 | 52.9437 | 45.2454 | 15.3965 | 0.6648534 | 0.4933 | 0.2580319 |
| E | 230 | 33.2277 | 57.8372 | 45.5325 | 24.6095 | 1.6984211 | 0.78035 | 0.5405439 |
| D | 115.5 | 29.4723 | 63.3465 | 46.4094 | 33.8741 | 3.2357364 | 1.6573 | 0.4878137 |
| C | 58 | 26.0287 | 67.991 | 47.0098 | 41.9622 | 4.9752536 | 2.25775 | 0.546204 |
| B | 29.5 | 22.9927 | 70.9356 | 46.9641 | 47.943 | 6.47482 | 2.21205 | 0.6583612 |
| A | 15 | 17.1248 | 75.0464 | 46.0856 | 57.9216 | 9.3906896 | 1.3335 | 0.8579976 |
| Z | 8 | 0 | 80.7838 | 40.3919 | 80.7838 | 18.440665 | -4.3602 | 1.2364448 |
| Y | 4.5 | 0 | 86.3004 | 43.1502 | 86.3004 | 20.83149 | -1.6019 | 1.076898 |
| X | 2.5 | 0 | 93.6227 | 46.8114 | 93.6227 | 24.530056 | 2.05925 | 0.916052 |
| W | 1.5 | 0 | 94.8623 | 47.4312 | 94.8623 | 25.215479 | 2.67905 | 0.8937538 |
| V | 1 | 0 | 95.8812 | 47.9406 | 95.8812 | 25.791724 | 3.1885 | 0.8763751 |
|  |  |  |  |  |  |  |  |  |
|  | **Depth** | **Lower** | **Upper** | **Mid** | **Spread** | **xaxis** | **yaxis** | **pestimate** |
| F | 459.5 | 1409.7872 | 2803.033 | 2106.41 | 1393.246 | 123.8372 | 103.6621 | 0.162916 |
| E | 230 | 1104.0833 | 3345.146 | 2224.615 | 2241.062 | 325.7561 | 221.8664 | 0.3189187 |
| D | 115.5 | 868.6192 | 4012.782 | 2440.7 | 3144.163 | 664.8969 | 437.9524 | 0.3413228 |
| C | 58 | 677.4954 | 4622.77 | 2650.133 | 3945.274 | 1076.1223 | 647.3844 | 0.39841 |
| B | 29.5 | 528.6672 | 5031.89 | 2780.278 | 4503.222 | 1416.6302 | 777.5304 | 0.4511409 |
| A | 15 | 293.2576 | 5631.962 | 2962.61 | 5338.705 | 2008.9339 | 959.8619 | 0.5222033 |
| Z | 8 | 0 | 6526.025 | 3263.012 | 6526.025 | 3054.6817 | 1260.2643 | 0.5874319 |
| Y | 4.5 | 0 | 7448.501 | 3724.25 | 7448.501 | 4202.6279 | 1721.5022 | 0.5903748 |
| X | 2.5 | 0 | 8765.261 | 4382.63 | 8765.261 | 6209.2899 | 2379.8822 | 0.6167223 |
| W | 1.5 | 0 | 8999.896 | 4499.948 | 8999.896 | 6612.2996 | 2497.2 | 0.6223402 |
| V | 1 | 0 | 9193.204 | 4596.602 | 9193.204 | 6954.651 | 2593.8539 | 0.6270332 |

From the above LV plots, it can be seen that mid summaries of the first LV plot is more stable than the mid summaries of the second LV plot. The first LV plot corresponds to fourth root and second LV plot corresponds to square root. These LV plots clearly indicates as specified above how it stabilizes the mid summaries.









Also, it can be seen that from the table 4-5, matched cube root transform does better than other transforms, especially the mid summaries is near to the true mid summaries.

Attachments: All R programs used for data analysis.

Firstprogram.r

library(noncensus)

data(counties)

findSpread <- function(nList){

sortedInput<-sort(nList)

medianFlg<-(1+length(sortedInput))/2

median <- ifelse(medianFlg==floor(medianFlg),sortedInput[medianFlg],(sortedInput[medianFlg-0.5]+sortedInput[medianFlg+0.5])/2)

#print(median)

flFlg<-(1+floor(medianFlg))/2

fl<-ifelse(flFlg==floor(flFlg),sortedInput[flFlg],(sortedInput[flFlg-0.5]+sortedInput[flFlg+0.5]/2))

#print(fl)

fuFlg<-length(sortedInput)-flFlg+1

fu<-ifelse(fuFlg==floor(fuFlg),sortedInput[fuFlg],(sortedInput[fuFlg-0.5]+sortedInput[fuFlg+0.5]/2))

#print(fu)

return(fu-fl)

}

states <- levels(counties$state)

medianDS <- as.numeric()

spreadDS <- as.numeric()

for (i in states){

inState <- subset(counties,state==i)

inPop <- sort(inState$population[!is.na(inState$population)])

median <- median(inState$population)

medianDS <- c(medianDS,median)

spread <- findSpread(inPop)

spreadDS <- c(spreadDS,spread)

}

statesSpread <- cbind.data.frame(states,medianDS,spreadDS)

statesSpread <- statesSpread[complete.cases(statesSpread),]

statesSpread <- subset(statesSpread,medianDS>0 & spreadDS >0)

statesSpread[statesSpread$states %!in% c("PR","GU","VI"),]

plot(statesSpread$medianDS,statesSpread$spreadDS,main="Spread vs Level Plot without log",xlab = "Median",ylab="Spread")

linear<-lm(statesSpread$spreadDS~statesSpread$medianDS)

abline(linear,lty=2,col="blue")

1-linear$coefficients[2]

plot(log10(statesSpread$medianDS),log10(statesSpread$spreadDS),main="Spread vs Level Plot with log",xlab = "Median",ylab="Spread")

linear<-lm(log10(statesSpread$spreadDS)~log10(statesSpread$medianDS))

abline(linear,lty=2,col="blue")

linear$coefficients[2]

1-linear$coefficients[2]

CA <- subset(counties,state=="CA")

source("lvalprogs.r")

lval(CA$population)

floors<-c(45578.0,20007.0,13994.0,6463.0,2207.5,1175.0)

ceils<-c(685306.0,1418788.0,2112425.5,3052772.5,6456959.0,9818605.0)

y<-ggplot(counties,aes(state,population,color=state))+geom\_boxplot()+theme\_bw()+guides(color=FALSE)

print(y)

z<-ggplot(counties,aes(state,log(population),color=state,guides=FALSE))+geom\_boxplot()+theme\_bw()+guides(color=FALSE)

print(z)

y<-ggplot(CA,aes(state,-(population)^5,guides=FALSE))+geom\_boxplot()+theme\_bw()+guides(color=FALSE)

print(y)

plotbatchtransform.r

plotTransformBatch<-function(inlist){

#inlist<-(CA$population)

source("lvalprogs.r")

tab <- as.data.frame(lval(inlist))

M <- tab$Lower[1];

spreads <- tab[tab$Spread>0,1:5]

xaxis <- as.numeric()

yaxis <- as.numeric()

pestimate <- as.numeric()

for (i in seq(1:length(spreads$Mid))){

xaxis<-c(xaxis,((spreads$Upper[i]-M)^2+(M-spreads$Lower[i])^2)/(4\*M))

yaxis<-c(yaxis,(((spreads$Upper[i]+spreads$Lower[i])/2)-M))

pestimate <- c(pestimate,1-(yaxis[i]/xaxis[i]))

}

spreads["xaxis"]<-xaxis

spreads["yaxis"]<-yaxis

spreads["pestimate"]<-pestimate

spreads<-spreads[complete.cases(spreads),]

print(spreads)

plot(spreads$xaxis,spreads$yaxis,main="Spread vs Level Plot",xlab="x estimates",ylab="Y estimates")

y<-lm(spreads$yaxis~spreads$xaxis)

abline(y,lty=2,col="blue")

print(paste("The power is ",1-coefficients(y)[2]))

print(paste("The slope is ",coefficients(y)[2]))

#return (spreads)

}

Question2.r

library(ggplot2)

setwd("C:/Users/Ganesh/Google Drive/Courses/STAT S 670/Homework 3")

source("plotTransformBatch.R")

CEOCompensation<-read.table("ceo.txt",header = TRUE)

summary(CEOCompensation)

nrow(CEOCompensation)

max(CEOCompensation)

plot(CEOCompensation$TotalCompensation,main="CEO Compensation - Scatter Plot",ylab = "Salary")

plotC<-cbind(sort(CEOCompensation$TotalCompensation),1:nrow(CEOCompensation))

colnames(plotC)<-c("salary","index")

plotC<-as.data.frame(plotC)

qplot(index,salary,data=plotC,geom="line",main = "Sorted Salaries vs Index")

hist(CEOCompensation$TotalCompensation,breaks = 100,freq = F,main = "Distribution of Salaries",xlab = "Compensation")

lines(density(CEOCompensation$TotalCompensation,kernel = "epanechnikov"),lty=1,col="red",lwd=2)

qqnorm(CEOCompensation$TotalCompensation,main="QQ Plot for Salary Distribution")

qqline(CEOCompensation$TotalCompensation)

plotTransformBatch((CEOCompensation$TotalCompensation))

#Box plot without any transfo

boxplot((CEOCompensation$TotalCompensation),main="Without Transformation")

#One Transformation 4th root that equals out

plotTransformBatch((CEOCompensation$TotalCompensation)^0.25)

boxplot((CEOCompensation$TotalCompensation)^0.25,main="Fourth root transormation")

#One Transformation log that equals out

plotTransformBatch((CEOCompensation$TotalCompensation)^0.5)

boxplot((CEOCompensation$TotalCompensation)^0.5,main="Square root transform Transformation")

FL<-c(2142,1788,1517,1248,963.5,727.5,579,345,114)

M<-c(3678,4115.5,4400.5,4799,4978.75,5241,5394.5,5510.25,5494)

FU<-c(4944,6643,7284,8350,8994,9754.5,10210,10675.5,1087)

Letters<-c("F","E","D","C","B","A","Z","Y","")

z<-cbind(FL,M,FU)

z<-(690\*(z)^(1/3))-7000

z<-cbind(Letters,z)

question3.r

#Transformation plots

xlf<-function(xl,xu,median){

return (((xl+xu)/2)-median)

}

ylf<-function(xl,xu,median){

return (((xu-median)^2+(median-xl)^2)/(4\*median))

}

median<-c(179140.5)

letters<-c("F","E","D","C","B","A")

xl<-c(45578.0,20007.0,13994.0,6463.0,2207.5,1175.0)

xu<-c(685306.0,1418788.0,2112425.5,3052772.5,6456959.0,9818605.0)

xlab <- as.numeric()

ylab <- as.numeric()

p <- as.numeric()

for (i in seq(1:length(letters))){

ytemp<-ylf(xl[i],xu[i],median)

xtemp<-xlf(xl[i],xu[i],median)

ylab <- c(ylab,ytemp)

xlab <- c(xlab,xtemp)

p<-c(p,(1-(ytemp/xtemp)))

}

estimator<-cbind(letters,xl,xu,xlab,ylab,p)

print(estimator)