Home-Work-2

Naren Suri

September 15, 2015

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Question 1: Lval plot

```
source("lvalprogs.r")
print("used the exponential funciton with just two parameters, a simple one with rate and n")
```

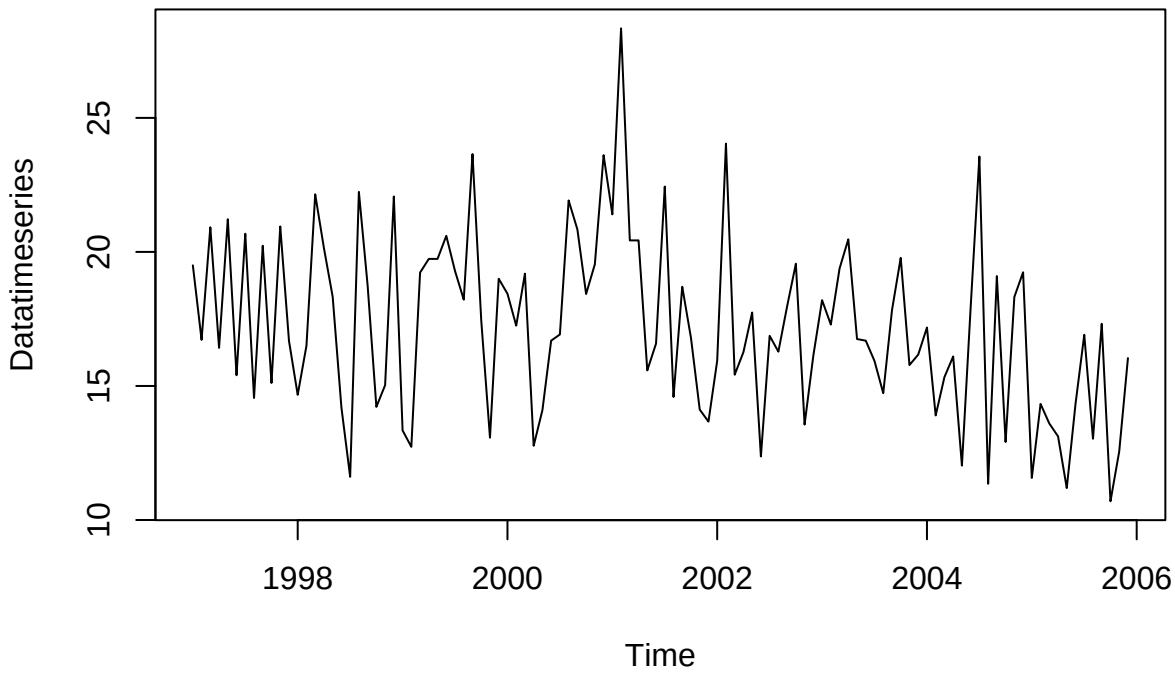
```
## [1] "used the exponential funciton with just two parameters, a simple one with rate and n"
```

```
DataForLeterVal = rexp(1000, 1)
lval(DataForLeterVal)
```

```
##   Depth  Lower  Upper    Mid Spread pseudo-s
## M 500.5 0.7016 0.7016 0.7016 0.0000   0.0000
## F 250.5 0.3092 1.4895 0.8993 1.1803   0.8749
## E 125.5 0.1367 2.2138 1.1753 2.0771   0.9028
## D  63.0 0.0536 2.7409 1.3972 2.6873   0.8758
## C  32.0 0.0200 3.5844 1.8022 3.5644   0.9568
## B  16.5 0.0102 4.5017 2.2560 4.4915   1.0427
## A   8.5 0.0048 5.0729 2.5388 5.0681   1.0482
## Z   4.5 0.0035 5.8679 2.9357 5.8644   1.1023
## Y   2.5 0.0022 5.9727 2.9874 5.9705   1.0345
## X   1.5 0.0013 7.0899 3.5456 7.0886   1.1443
## W   1.0 0.0006 8.1812 4.0909 8.1806   1.2405
```

Question 2:

```
#
setwd("D:/Sem1-DataScience/Exploratory-DA/home-work/HW-2");
TimeSeriesData = scan("D:/Sem1-DataScience/Exploratory-DA/home-work/HW-2/timeseries.dat")
#TimeSeriesData;
Datatimeseries = ts(TimeSeriesData, frequency=12, start=c(1997,1))
plot.ts(Datatimeseries)
```



Stem and Leaf plot there are postive skews, negitive skews, unimodel, bimodel - modality represents the peaks on the curve.

```
library("aplpack", lib.loc="~/R/win-library/3.2")
```

```
## Loading required package: tcltk
```

```
stem.leaf(TimeSeriesData,m=2)
```

```
## 1 | 2: represents 1.2
##  leaf unit: 0.1
##             n: 108
##     1    10. | 7
##     3    11* | 13
##     5    11. | 56
##     7    12* | 03
##    11    12. | 5779
##    15    13* | 0013
##    19    13. | 5669
##    25    14* | 112233
##    29    14. | 5567
##    34    15* | 01344
##    38    15. | 5799
##    45    16* | 0111224
##   (11)   16. | 55666777899
##    52    17* | 12234
```

```
##    47    17. | 7899
##    43    18* | 223344
##    37    18. | 77
##    35    19* | 0112223
##    28    19. | 555777
##    22    20* | 12444
##    17    20. | 66899
##    12    21* | 24
##    10    21. | 9
##     9    22* | 0124
##          22. |
##          23* |
##     5    23. | 566
##     2    24* | 0
## HI: 28.34
```

this has two stems, and mean is 11. Depth of each data point is also illustrated in the left. This has the bimodel nature, since it got two peaks.

```
stem(TimeSeriesData)
```

```
##
##   The decimal point is at the |
##
##   10 | 7
##   11 | 2466
##   12 | 046789
##   13 | 01136679
##   14 | 1122336677
##   15 | 013446899
##   16 | 011233456777788999
##   17 | 23334789
##   18 | 022334478
##   19 | 0122234556778
##   20 | 124456789
##   21 | 0249
##   22 | 1224
##   23 | 667
##   24 | 0
##   25 |
##   26 |
##   27 |
##   28 | 3
```

this has got the most likely distributed curve. However, it has one data point that makes it the right skew.

```
stem.leaf(TimeSeriesData,m=1)
```

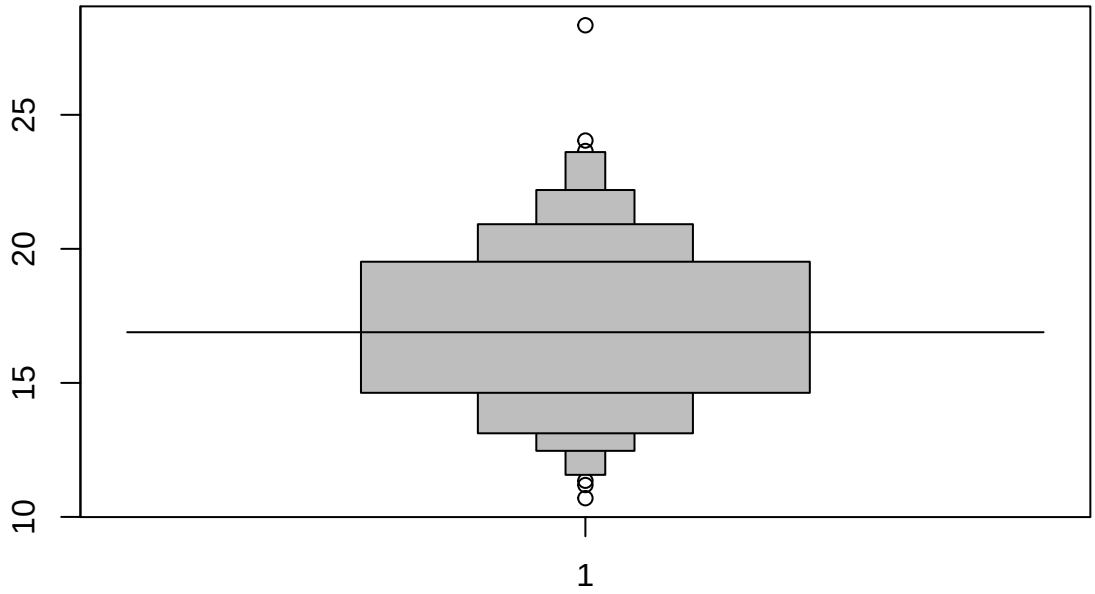
```
## 1 | 2: represents 1.2
##  leaf unit: 0.1
##             n: 108
##    1    10 | 7
##    5    11 | 1356
```

```
##   11    12 | 035779
##   19    13 | 00135669
##   29    14 | 1122335567
##   38    15 | 013445799
##   56    16 | 011122455666777899
##   (9)   17 | 122347899
##   43    18 | 22334477
##   35    19 | 0112223555777
##   22    20 | 1244466899
##   12    21 | 249
##    9    22 | 0124
##    5    23 | 566
##    2    24 | 0
## HI: 28.34
```

it got a mean 9, and the depth of the data points are illustrated to the left. This has the bimodel nature, since it has got two peaks. And the data is right skewd too.

Plot a letter value plot. The letter valu plots gives extra details regarding its tails. which means we can have a clar understanding of the outliers. They are not dense at the tails, but gives good details of the data points those are nar to quartiles and those are outliers.

```
source("lvalprogs.r")
lvplot(TimeSeriesData)
```



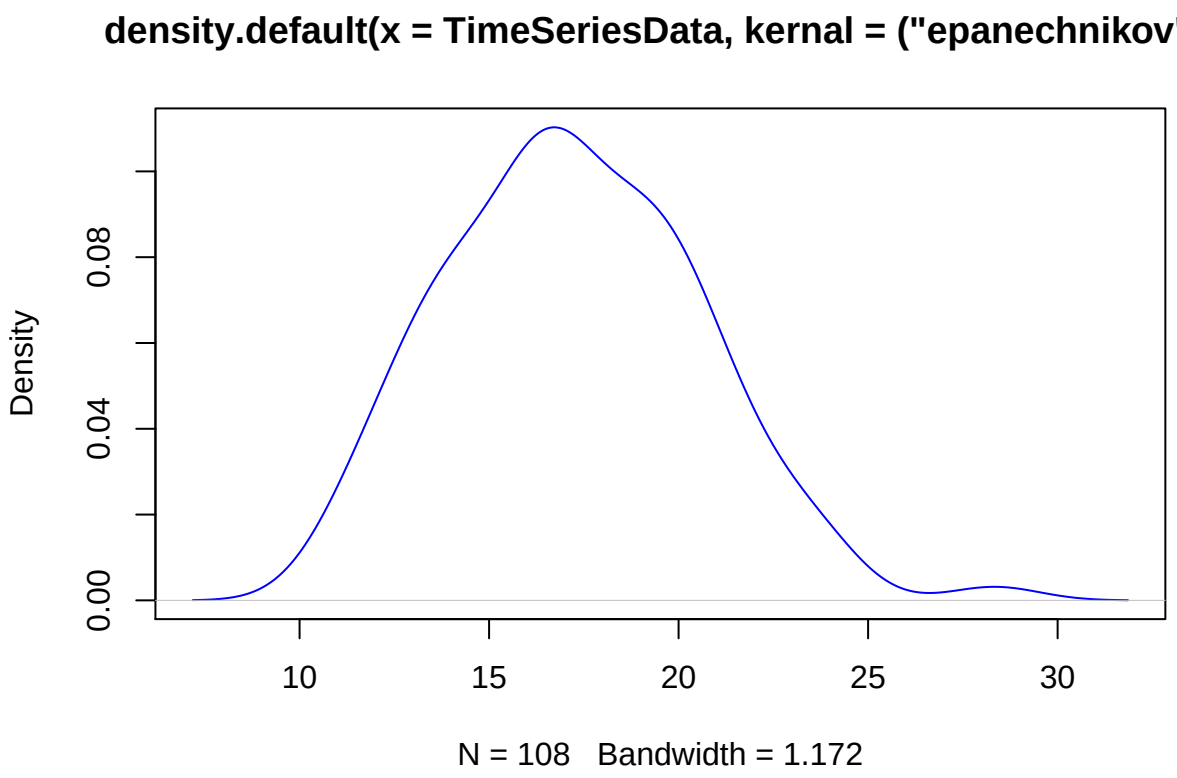
The result cearly shows that there is an outlier in the result obtained. The one value above the 25 is an

outlier. THe other few circles those we see are very near to the quartiles we have, so the outlier of the data is clarly identified with the level plot.

The plot is skewed a little bit to the right. Though its look like a normally distributed graph, its not a normalized one completely. The density plot below confirms the same.

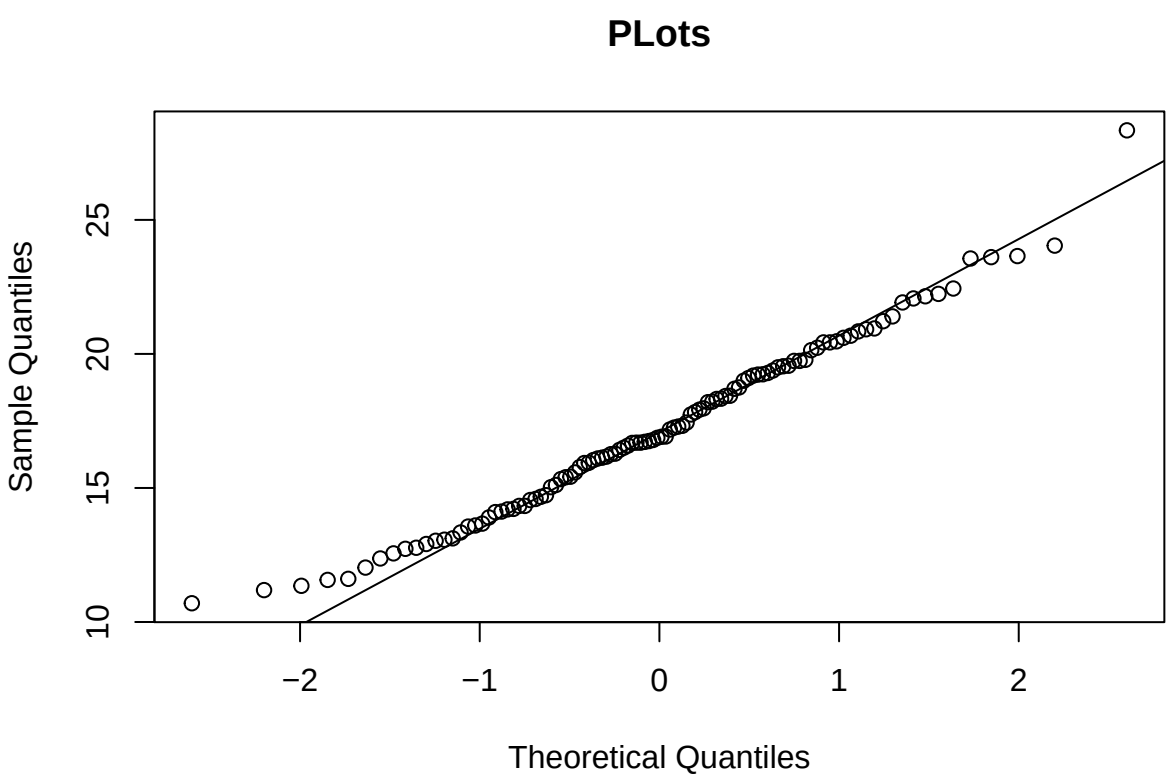
```
plot(density(TimeSeriesData,kernal=("epanechnikov")),col="blue")
```

```
## Warning in density.default(TimeSeriesData, kernal = ("epanechnikov")): non-
## matched further arguments are disregarded
```



The QQ Plots, we draw the qqnorm, which draws the qq plot of the given data and the general normal dstribution to compare how well the given data is normalized.

```
qqnorm(TimeSeriesData, main='PLots')
qqline(TimeSeriesData)
```



The Data is decently Normall Distributed. But as you se there are very few points who are deviated from the line. This states that the given data is Almost and very well Normalized.

Question 3:

The outside cut off talks about the values beyond the 2% of the nrmal distribution. 2%,14%,32% 32%,14%,2% . So the outside cut off value is two-Standard deviations away from the mean. Gausian said that any thing beyond 1.5df+Fu will be considered as the cut off point, and the area from there would be cutoff area. And this signifies the outlier too.

The average outside cutoff given by gausian for a single batch is : 0.4+0.007n

(a) In a single batch of 120 observations?

```
outsideCutoff = 0.4+0.007*120
outsideCutoff
```

```
## [1] 1.24
```

Is The ouside cutoff

(b) In total for two batches of 60?

```
outsideCutoffBatch1 = 0.4+0.007*60

outsideCutoffBatch2 = 0.4+0.007*60

result = outsideCutoffBatch1 + outsideCutoffBatch2

result
```

```
## [1] 1.64
```

is The ouside cutoff for both the batches

(c) In total for batches of 40, 30, 20, 10, 5, 5, 5, and 5?

```
outsideCutoffBatch1 = 0.4+0.007*40
outsideCutoffBatch2 = 0.4+0.007*30
outsideCutoffBatch3 = 0.4+0.007*20
outsideCutoffBatch4 = 0.4+0.007*10
outsideCutoffBatch5 = 0.4+0.007*5
outsideCutoffBatch6 = 0.4+0.007*5
outsideCutoffBatch7 = 0.4+0.007*5
outsideCutoffBatch8 = 0.4+0.007*5
result = outsideCutoffBatch1 + outsideCutoffBatch2+ outsideCutoffBatch3+ outsideCutoffBatch4+ outsideCu
result
```

```
## [1] 4.04
```

is The ouside cutoff for all the batches

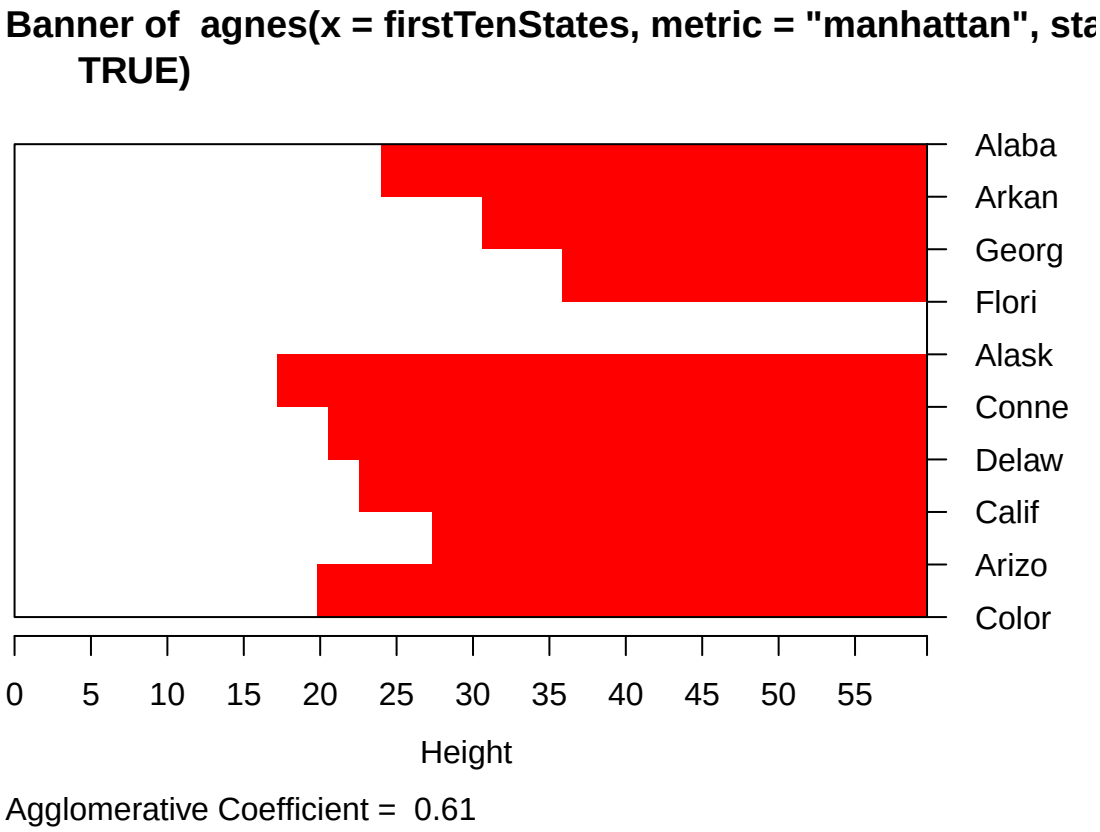
Question 4:

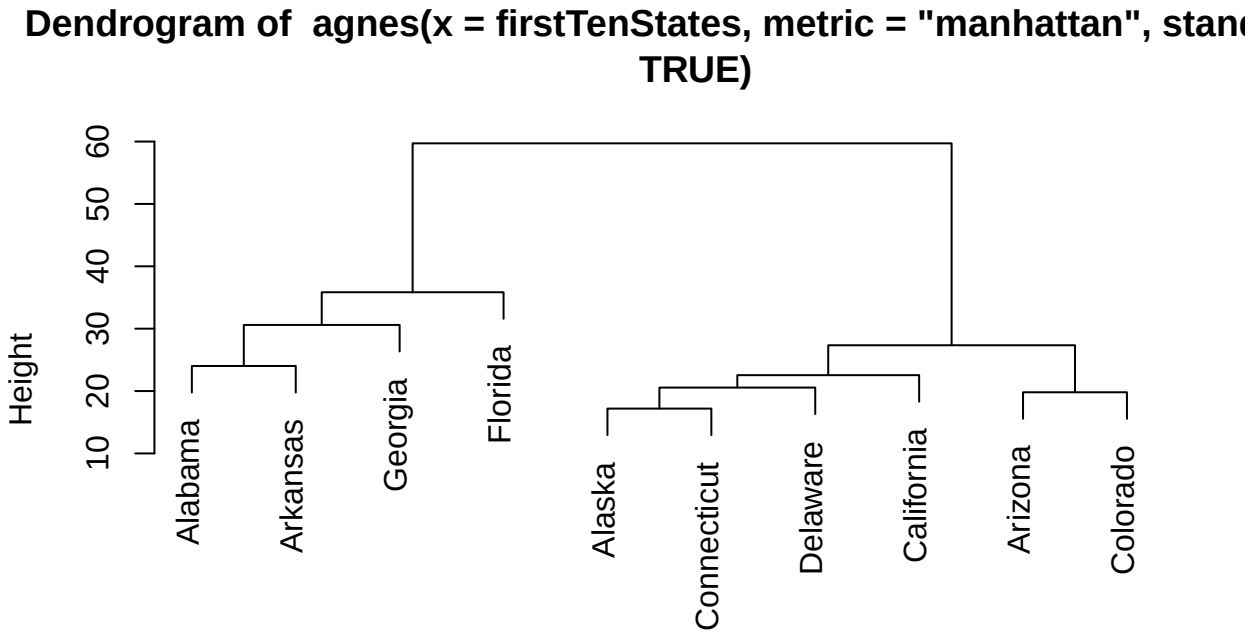
1. Plot the data by year for (1st, 2nd, 3rd, 4th, 5th) 10 states all on one graph (connect the points by lines). Plot a solid horizontal line at 50% for visual comparison (anchoring). Do any lines appear different from the others? Which ones? Compare plots.

```
library("cluster")
library("ggplot2")
library("grid")
library("gridExtra")
library("cowplot")
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

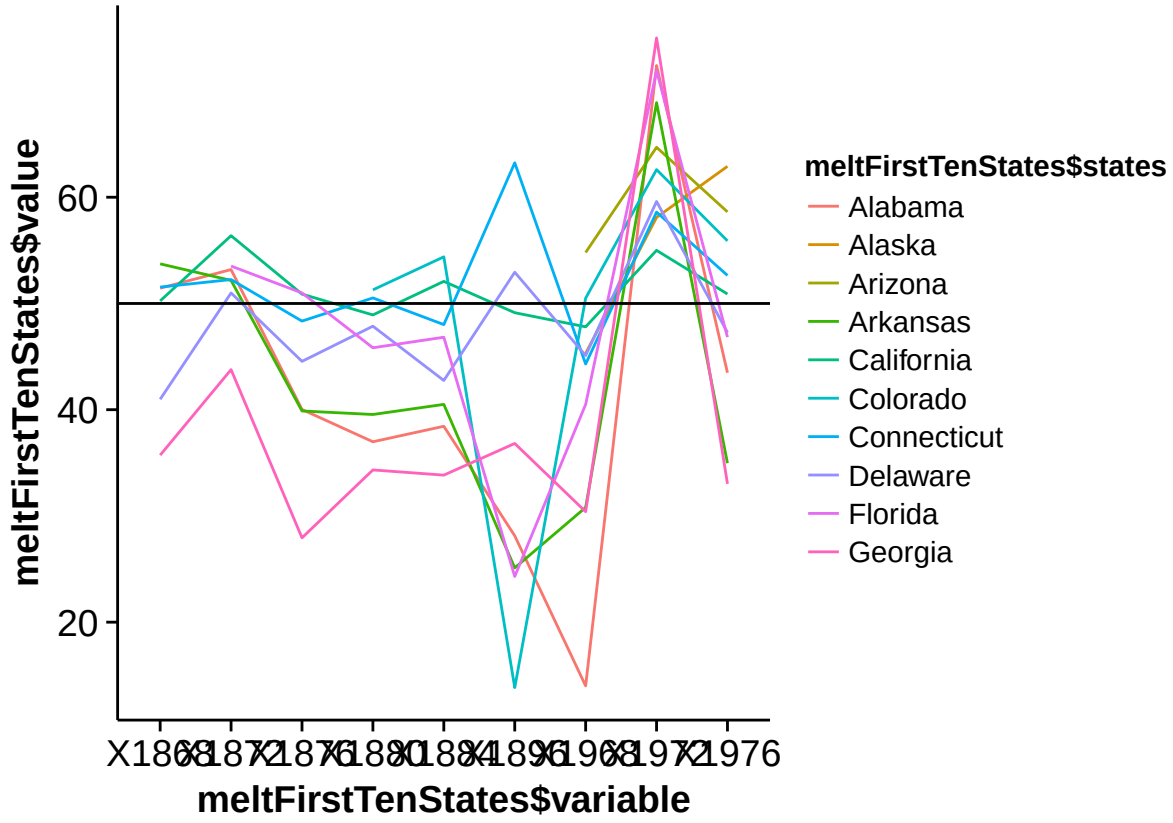
```r
data(votes.repub)
#help(votes.repub)
#votes.repub
firstTenStates = votes.repub[1:10,]
#firstTenStates
# Aglomerative
agglo <- agnes(firstTenStates, metric = "manhattan", stand = TRUE)
#agglo
plot(agglo)
```





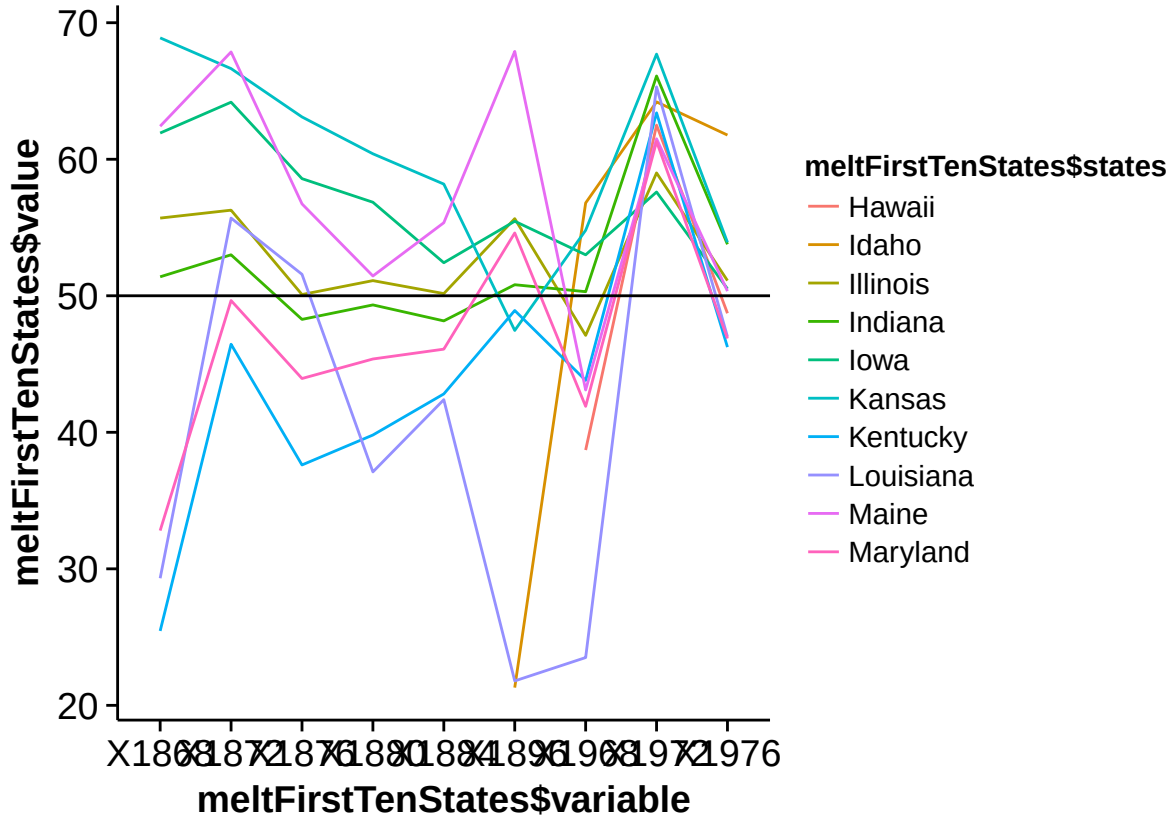

```
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
meltedData <- melt(completeData, id="states")
# but we shall melt data of only first 10 rows to plot the first ten states
firstTenstateRows =  completeData[1:10,c(4:8,11,29:32)]
#firstTenstateRows
meltFirstTenStates = melt(firstTenstateRows,id="states")
#meltFirstTenStates
Plot1 = ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_
ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_line(aes
```

```
## Warning: Removed 16 rows containing missing values (geom_path).
```



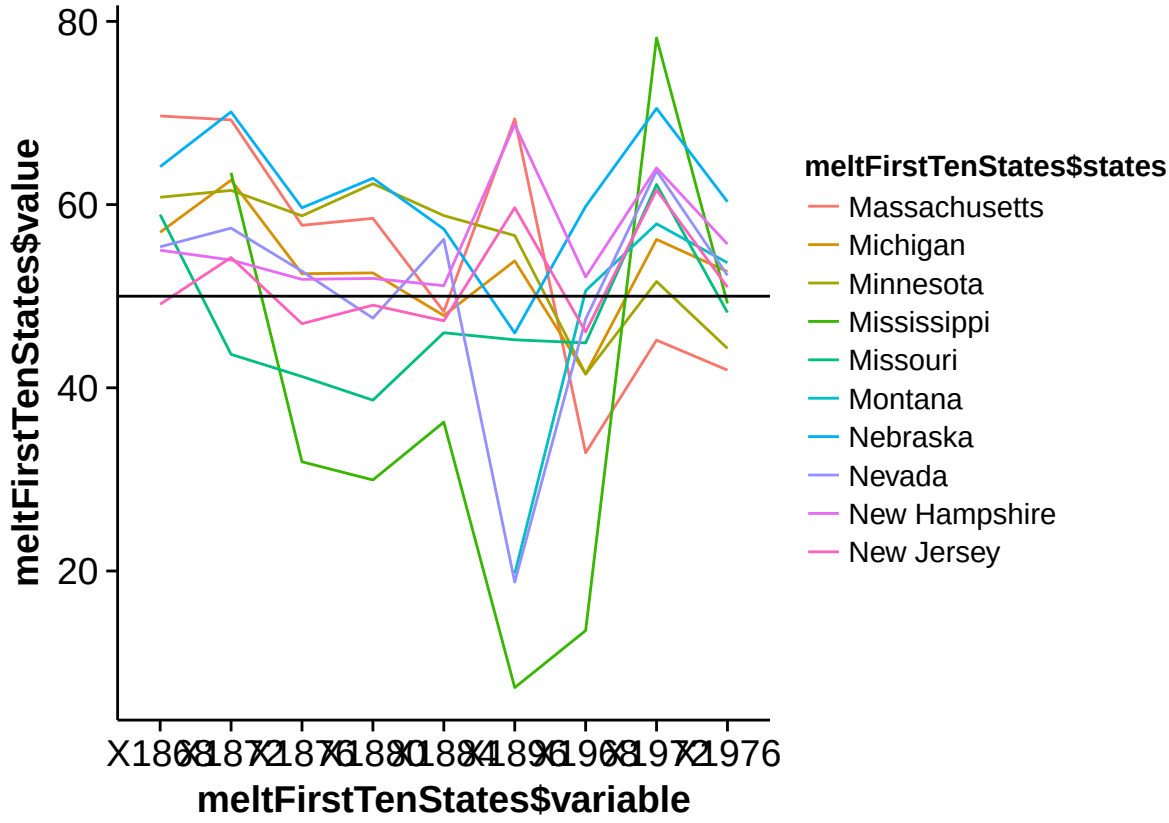
Second Group of states Graphs

```
library("cluster")
data(votes.repub)
#help(votes.repub)
#votes.repub

# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
meltedData <- melt(completeData, id="states")
# but we shall melt data of only first 10 rows to plot the first ten states
firstTenstateRows =  completeData[11:20,c(4:8,11,29:32)]
#firstTenstateRows
meltFirstTenStates = melt(firstTenstateRows,id="states")
#meltFirstTenStates
Plot2 = ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_

ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_line(aes

## Warning: Removed 11 rows containing missing values (geom_path).
```



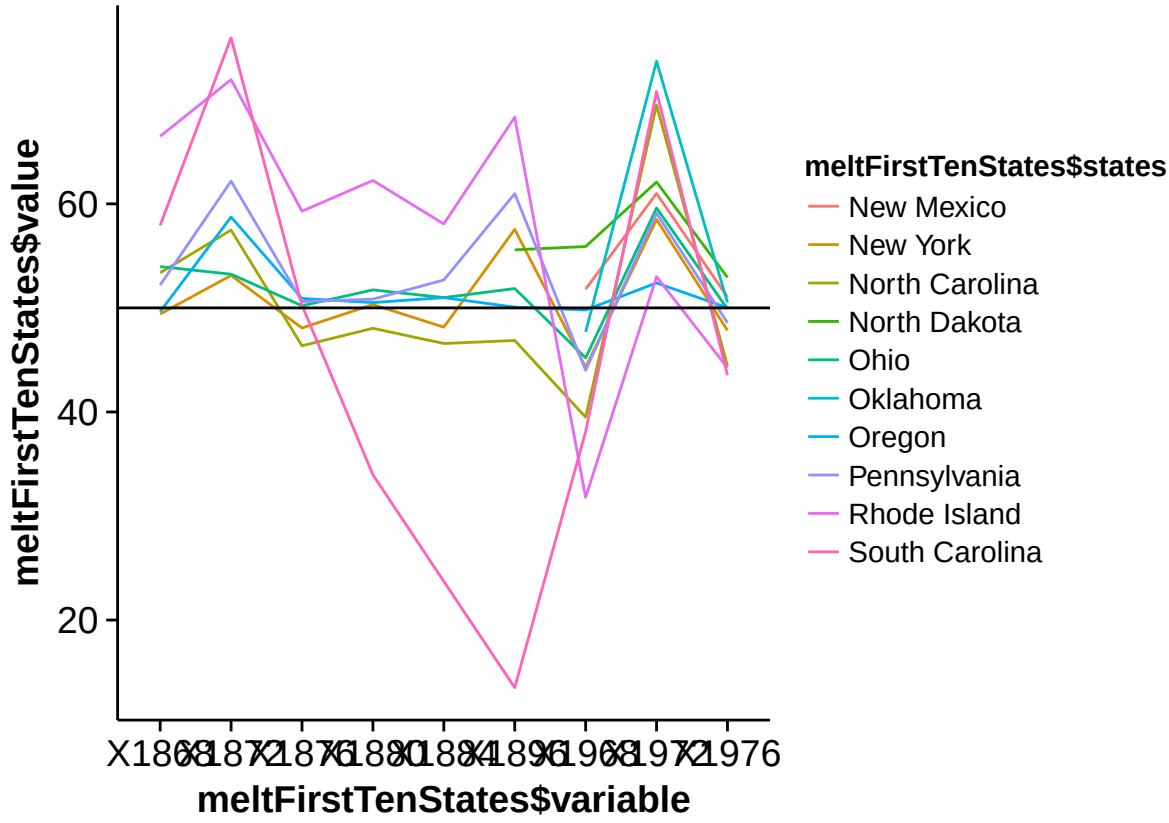
Third Group of states Graphs

```
library("cluster")
data(votes.repub)
#help(votes.repub)
#votes.repub

# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
meltedData <- melt(completeData, id="states")
# but we shall melt data of only first 10 rows to plot the first ten states
firstTenstateRows =  completeData[21:30,c(4:8,11,29:32)]
#firstTenstateRows
meltFirstTenStates = melt(firstTenstateRows,id="states")
#meltFirstTenStates
Plot3 = ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_

ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_line(aes

## Warning: Removed 6 rows containing missing values (geom_path).
```



Fourth Group of states Graphs

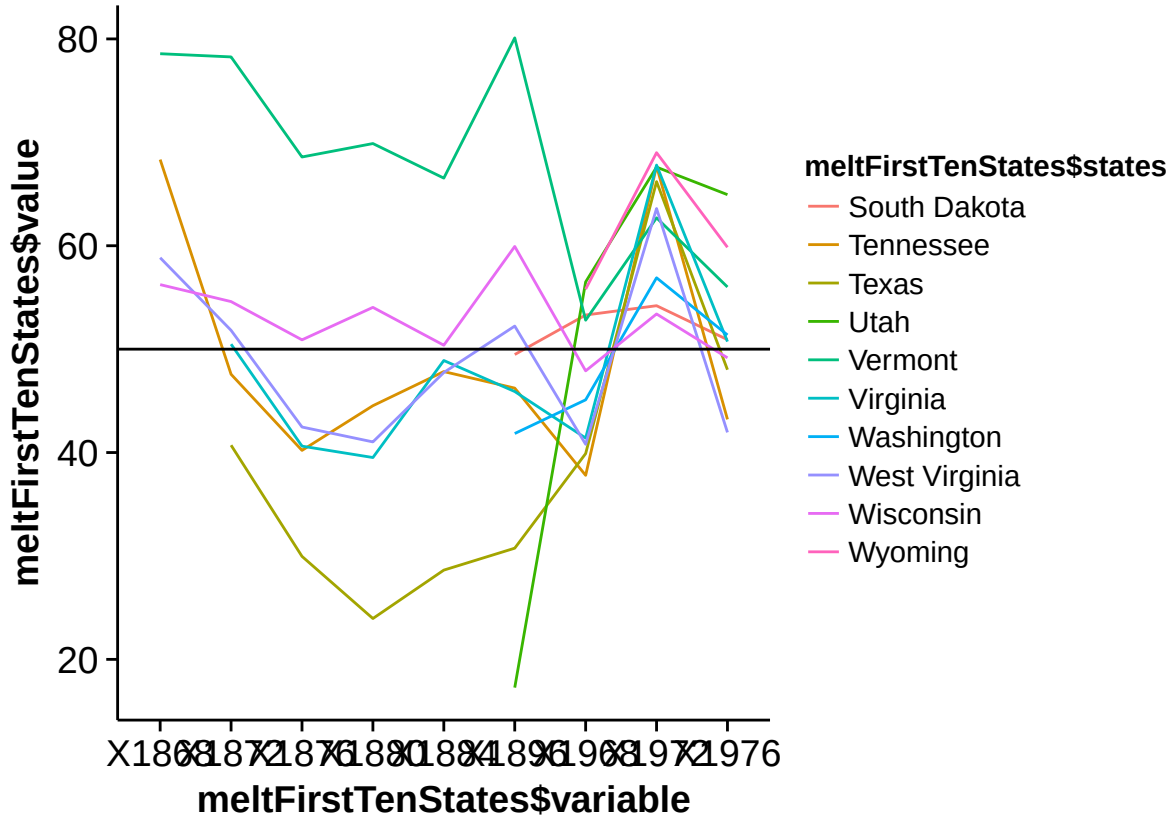
```
library("cluster")
data(votes.repub)
#help(votes.repub)
#votes.repub

# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
meltedData <- melt(completeData, id="states")
# but we shall melt data of only first 10 rows to plot the first ten states
firstTenstateRows =  completeData[31:40,c(4:8,11,29:32)]
#firstTenstateRows
meltFirstTenStates = melt(firstTenstateRows,id="states")
#meltFirstTenStates
Plot4 = ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_

ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_line(aes

## Warning: Removed 17 rows containing missing values (geom_path).
```



Fifth Group of states Graphs

```
library("cluster")
data(votes.repub)
#help(votes.repub)
#votes.repub

# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
# but we shall melt data of only first 10 rows to plot the first ten states
firstTenstateRows =  completeData[41:50,c(4:8,11,29:32)]
#firstTenstateRows
meltFirstTenStates = melt(firstTenstateRows,id="states")
#meltFirstTenStates
Plot5 = ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_

ggplot(meltFirstTenStates,aes(x= meltFirstTenStates$variable, y=meltFirstTenStates$value))+geom_line(aes
```

```
## Warning: Removed 23 rows containing missing values (geom_path).
```



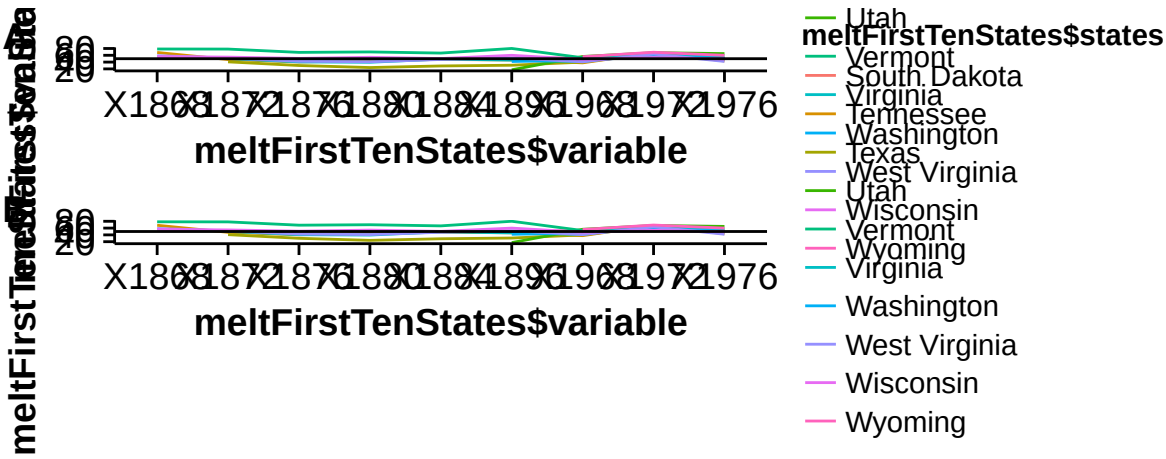

```
print("plotting every graph on a single page using grid is making it clumsy over the pdf. But it looks
```

```
## [1] "plotting every graph on a single page using grid is making it clumsy over the pdf. But it looks
```

```
plot_grid(Plot1, Plot2, labels=c("A", "B"), ncol = 1, nrow = 5)
```

```
## Warning: Removed 23 rows containing missing values (geom_path).
```

```
## Warning: Removed 23 rows containing missing values (geom_path).
```



2. Would a more sensible grouping of states make sense? 1-Northeast: CT DE ME MA NH NJ NY PA RI VT 2-Mid-Atlantic/East-Central: KY MD NC SC TN VA WV 3-South: AL AR FL GA LA MS OK TX 4-Midwest: IL IN IA KS MI MN MO NE OH WI 5-Rockies: CO ID MT ND SD UT WY 6-West: AK AZ CA HI NV NM OR WA Plot 6 groups on 1 page (par (mfrow=c (2,3))), also with abline (h=50) . What do you notice?

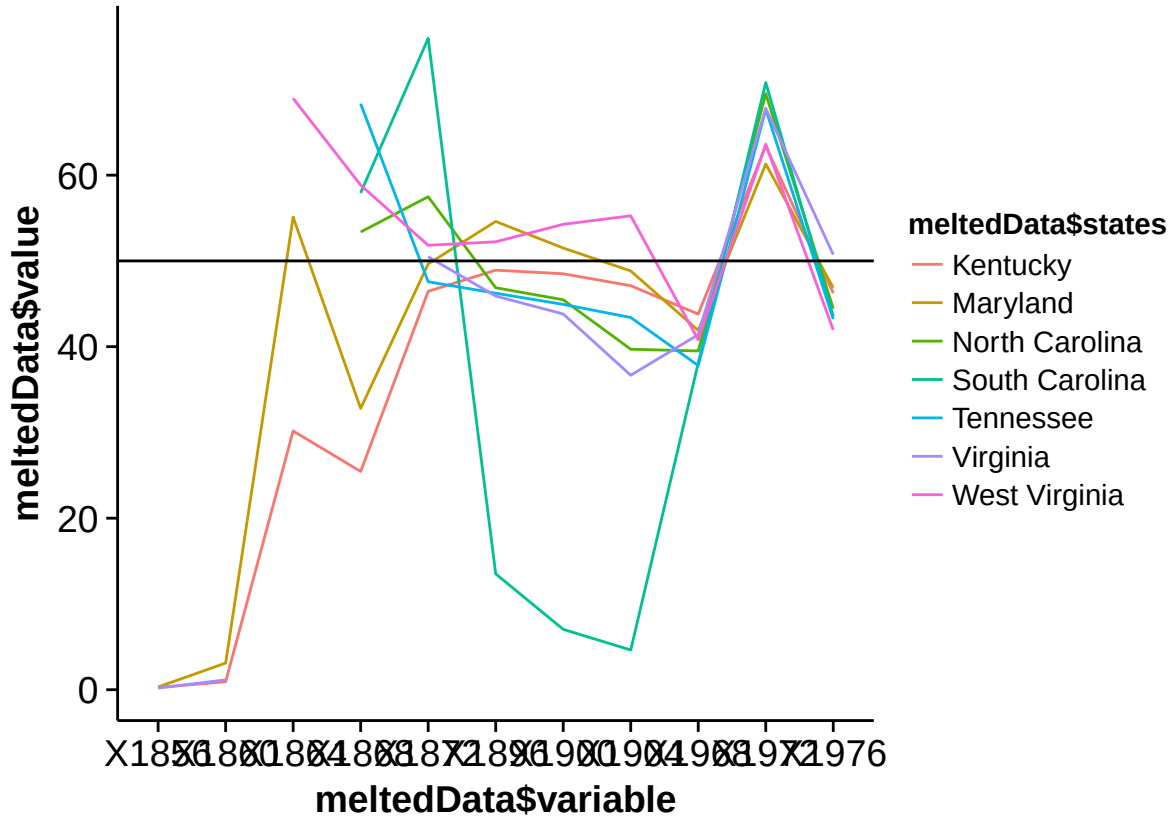
Group by the north east states and then should be drawn their corresponding graphs.

```
northeast = c("Connecticut","Delaware","Maine","Massachusetts","New Hampshire","New Jersey","New York",

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData[,c(1:5,11:13,29:32)]

library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")


North = ggplot(meltedData,aes(x= meltedData$variable, y=meltedData$value,group = meltedData$states, col
```

Second group of states , I am not chaninging the names of the variables, but the intution is that variable names are nothing but the region names. 2-Mid-Atlantic/East-Central: KY MD NC SC TN VA WV

```
northeast = c("Kentucky","Maryland","North Carolina","South Carolina","Tennessee","Virginia","West Virg

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData[,c(1:5,11:13,29:32)]

library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")


ggplot(meltedData,aes(x= meltedData$variable, y=meltedData$value,group = meltedData$states, colour=melt
```

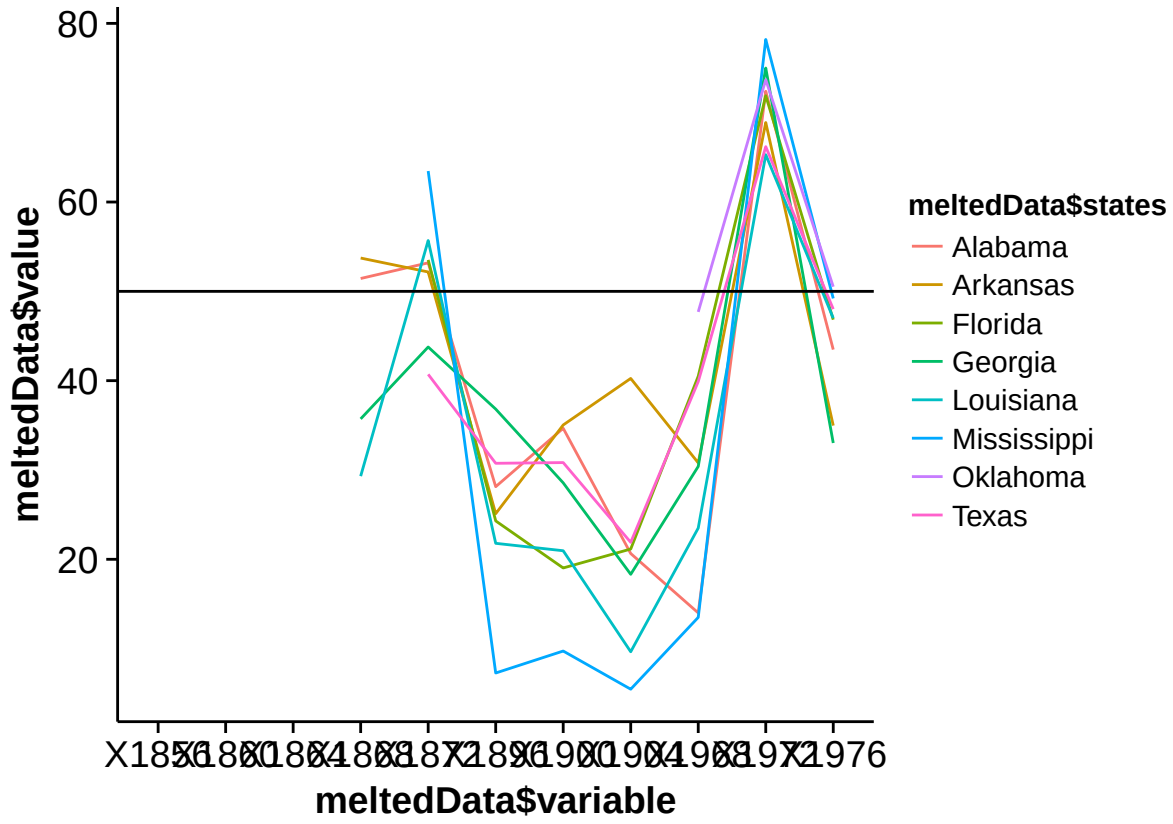
```
## Warning: Removed 11 rows containing missing values (geom_path).
```



3-South: AL AR FL GA LA MS OK TX ‘

```
northeast = c("Alabama","Arkansas","Florida","Georgia","Louisiana","Mississippi","Oklahoma","Texas")

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData[,c(1:5,11:13,29:32)]

library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")

ggplot(meltedData,aes(x= meltedData$variable, y=meltedData$value,group = meltedData$states, colour=melte
```

```
## Warning: Removed 32 rows containing missing values (geom_path).
```

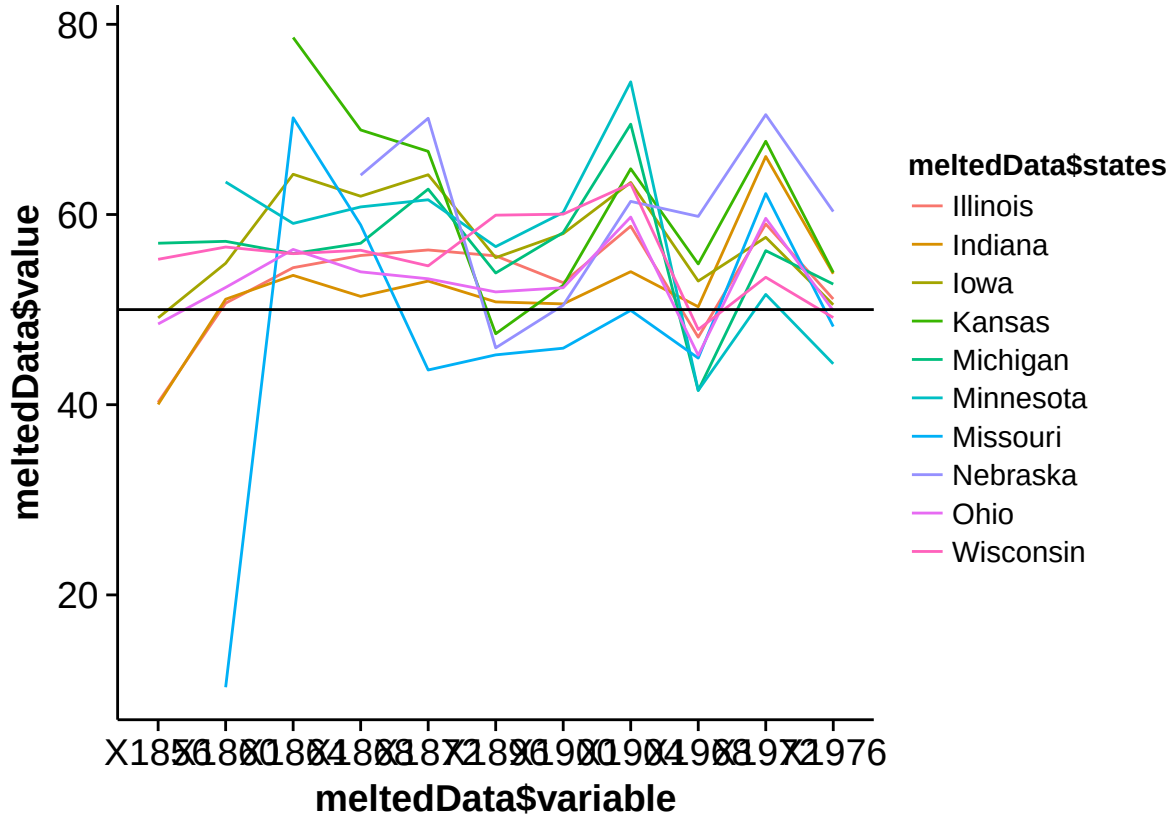


4-Midwest: IL IN IA KS MI MN MO NE OH WI

```
northeast = c("Illinois","Indiana","Iowa","Kansas","Michigan","Minnesota","Missouri","Nebraska","Ohio",

library("cluster")
data(votes.repub)
```

```
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData[,c(1:5,11:13,29:32)]

library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
ggplot(meltedData,aes(x= meltedData$variable, y=meltedData$value,group = meltedData$states, colour=melte
```

```
## Warning: Removed 7 rows containing missing values (geom_path).
```

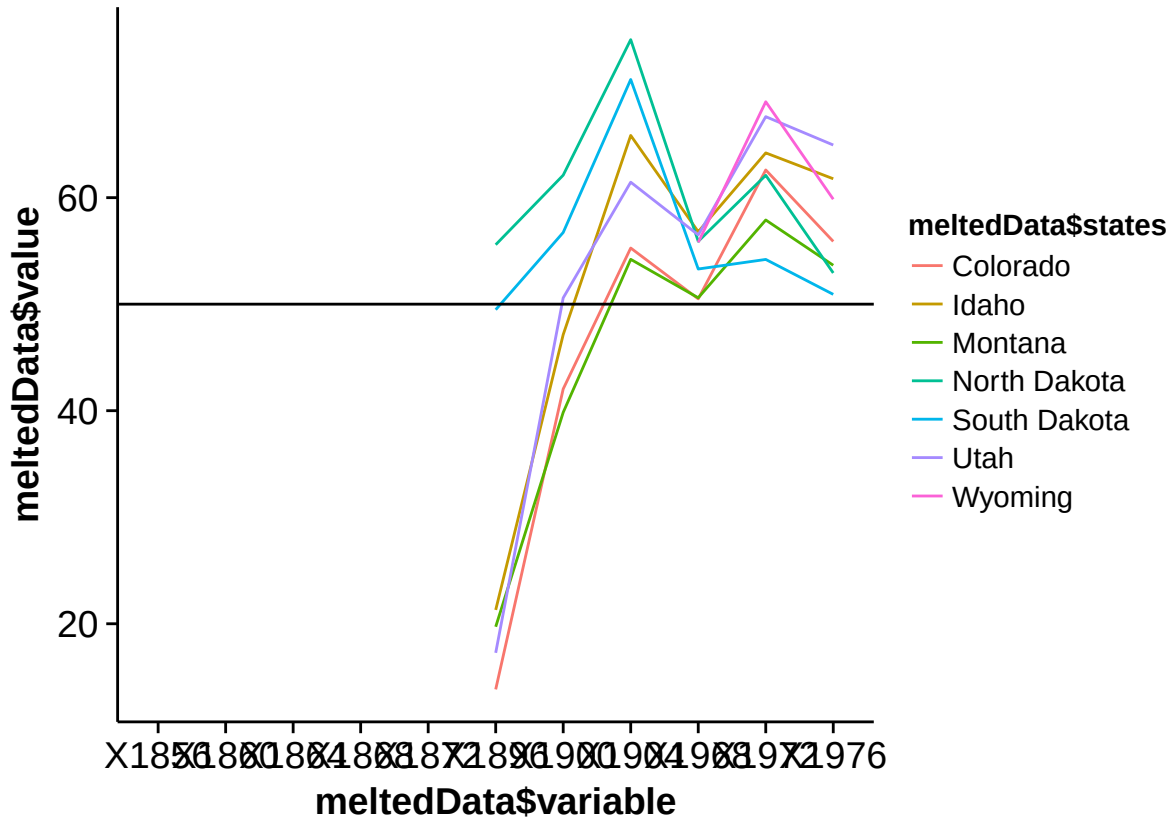


5-Rockies: CO ID MT ND SD UT WY

```
northeast = c("Colorado","Idaho","Montana","North Dakota","South Dakota","Utah","Wyoming")

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
```

```
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData[,c(1:5,11:13,29:32)]

library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
ggplot(meltedData,aes(x= meltedData$variable, y=meltedData$value,group = meltedData$states, colour=melte
```

```
## Warning: Removed 38 rows containing missing values (geom_path).
```

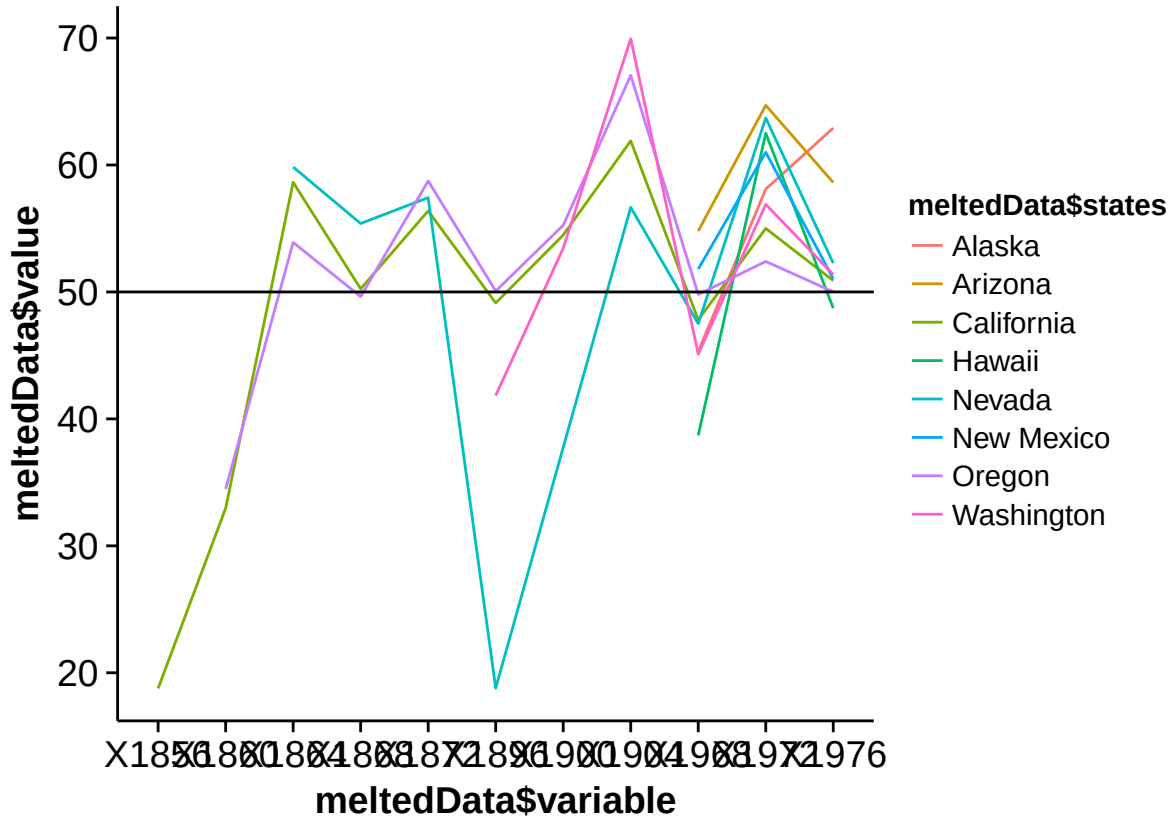


6-West: AK AZ CA HI NV NM OR WA

```
northeast = c("Alaska","Arizona","California","Hawaii","Nevada","New Mexico","Oregon","Washington")

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData[,c(1:5,11:13,29:32)]

library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
```

```
ggplot(meltedData,aes(x= meltedData$variable, y=meltedData$value,group = meltedData$states, colour=melte
```

```
## Warning: Removed 40 rows containing missing values (geom_path).
```

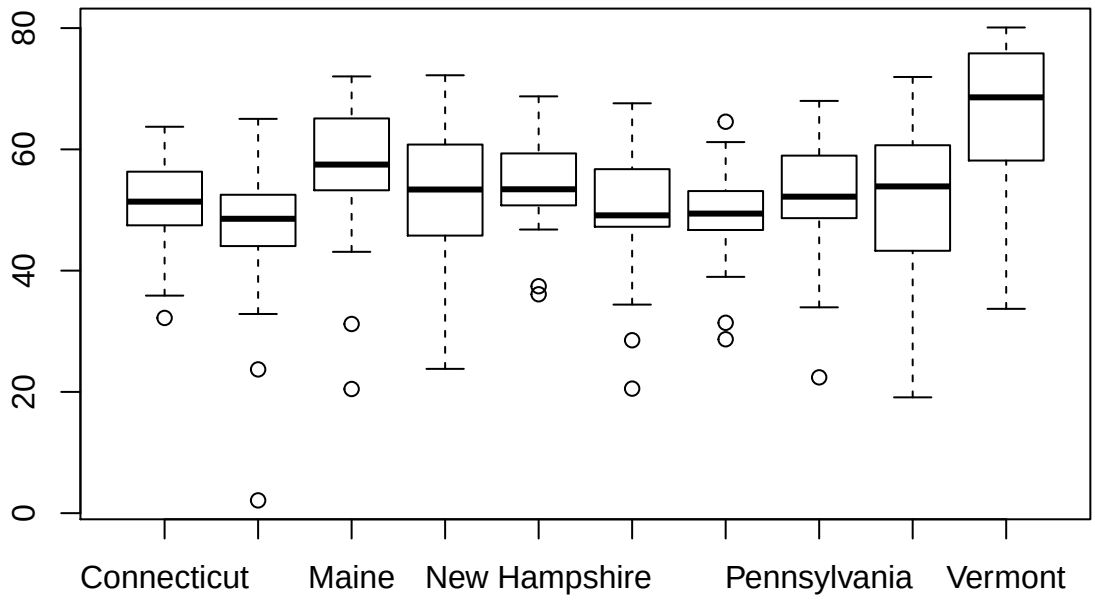



```
library("grid")
library("gridExtra")
library("cowplot")

#grid.arrange(Plot1,Plot2,Plot3,Plot4,Plot5,plot1)
```

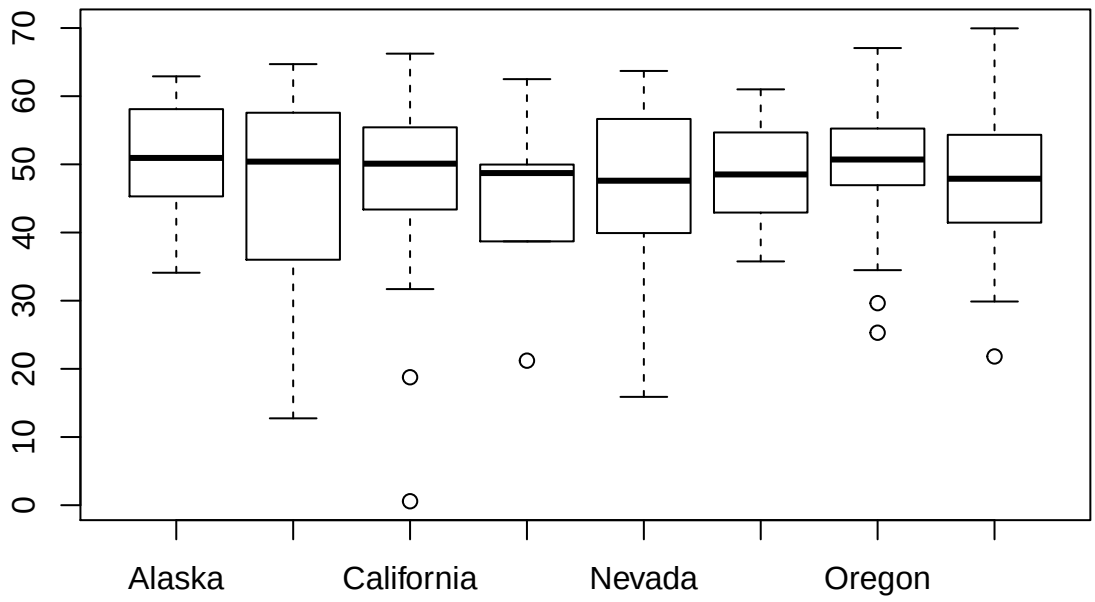
3. Boxplots, QQ-plots of the data (all & by groups) Box plot by each region and see how each state in the region is.

```
par(mfrow=par(mfrow=c(4,2)))
northeast = c("Connecticut","Delaware","Maine","Massachusetts","New Hampshire","New Jersey","New York",

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
```

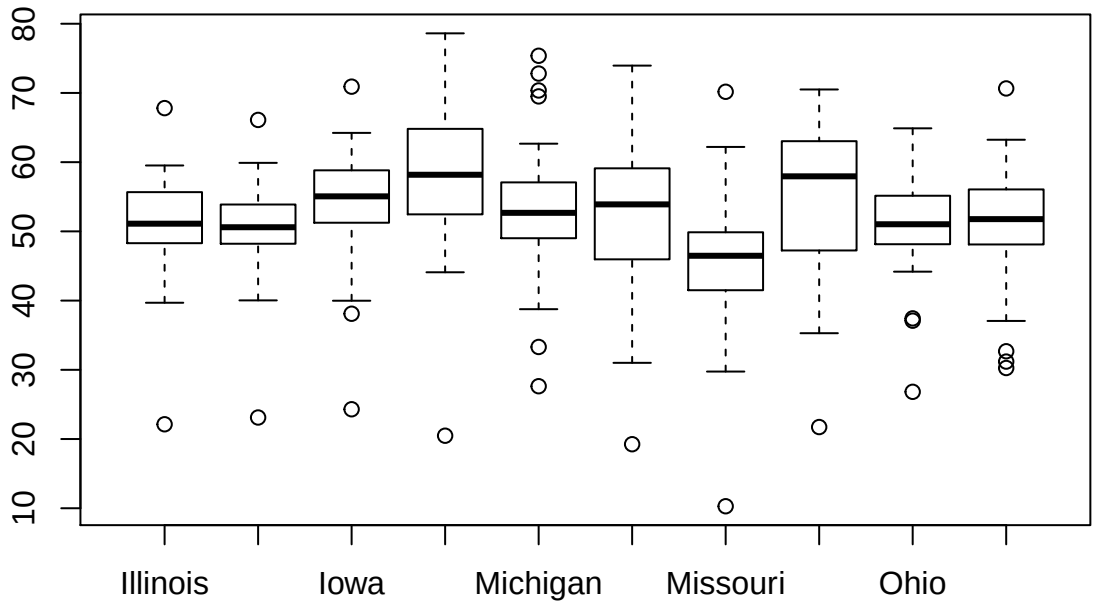
```
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
boxplot(meltedData$value ~ meltedData$states)
```



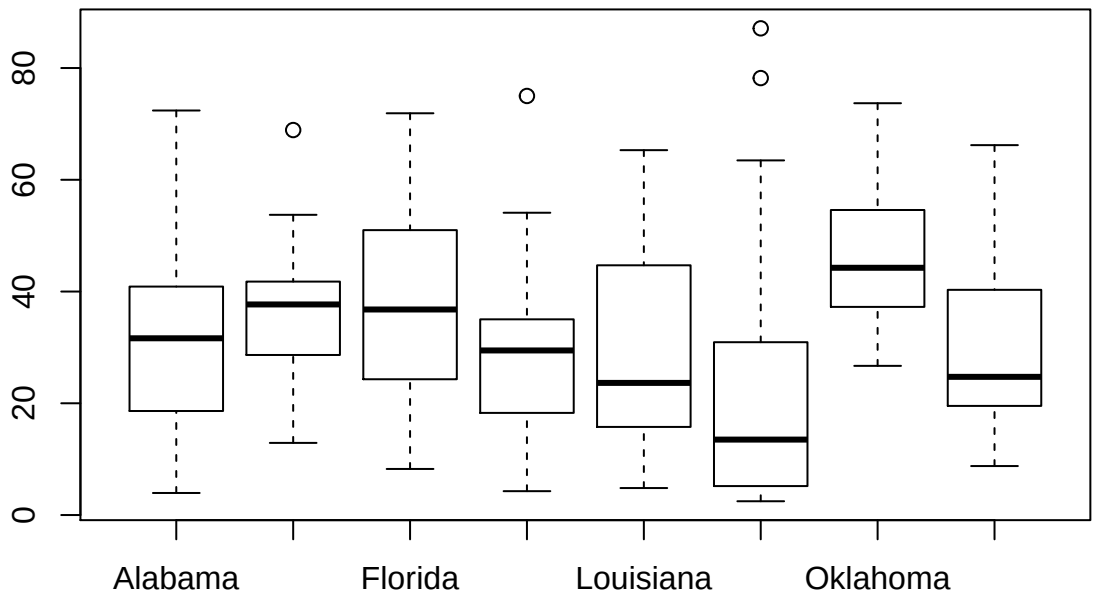

```
northeast = c("Alaska","Arizona","California","Hawaii","Nevada","New Mexico","Oregon","Washington")
library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
boxplot(meltedData$value ~ meltedData$states)
```



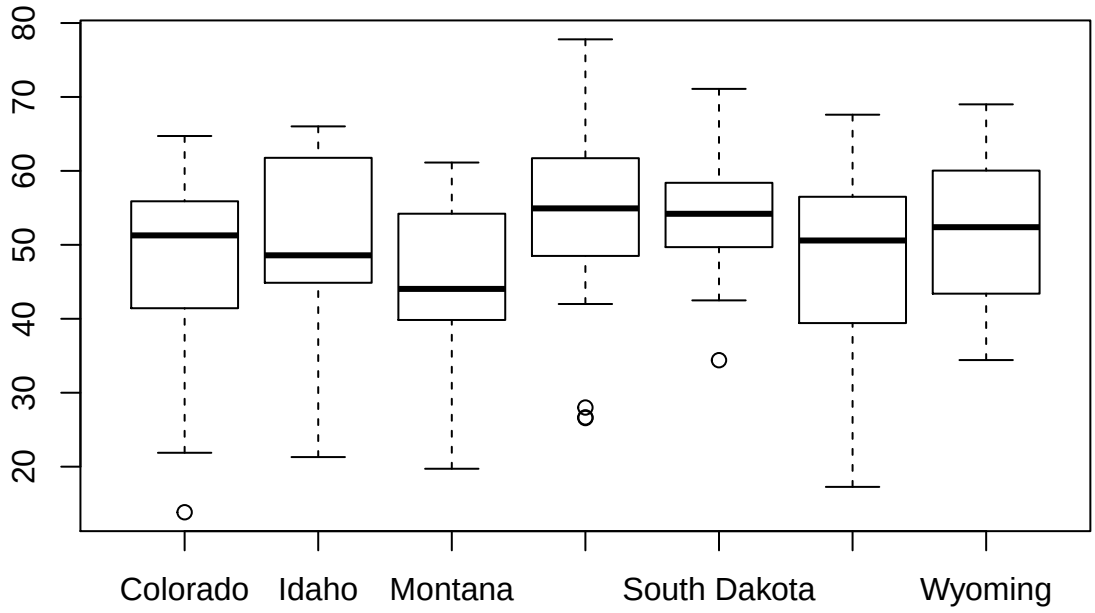

```
northeast = c("Illinois","Indiana","Iowa","Kansas","Michigan","Minnesota","Missouri","Nebraska","Ohio",
library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
boxplot(meltedData$value ~ meltedData$states)
```



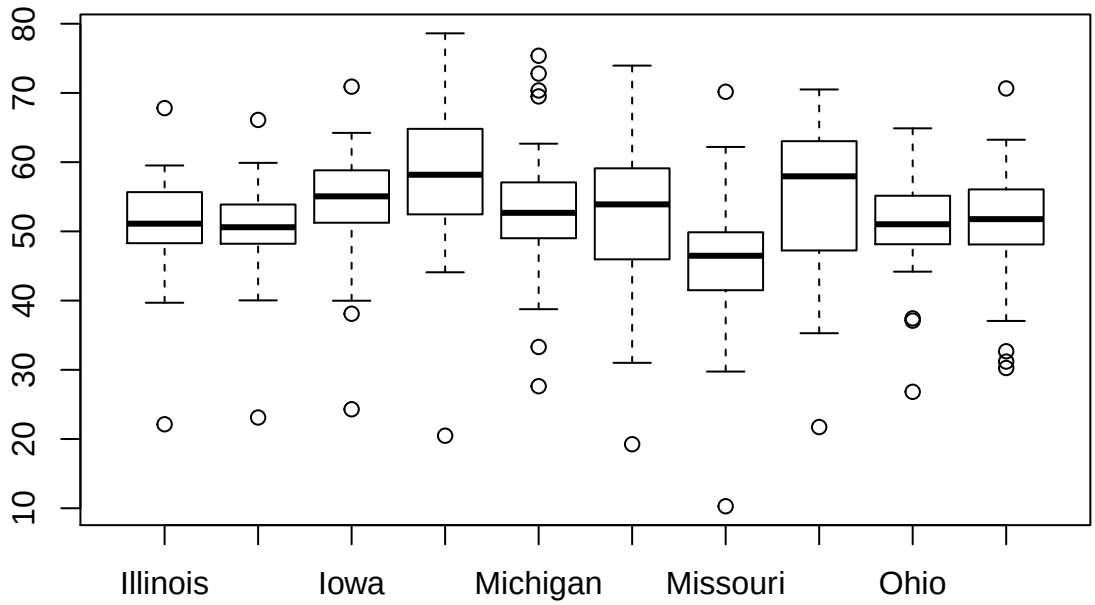

```
northeast = c("Alabama","Arkansas","Florida","Georgia","Louisiana","Mississippi","Oklahoma","Texas")

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
boxplot(meltedData$value ~ meltedData$states)
```




```
northeast = c("Colorado","Idaho","Montana","North Dakota","South Dakota","Utah","Wyoming")
library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
boxplot(meltedData$value ~ meltedData$states)
```

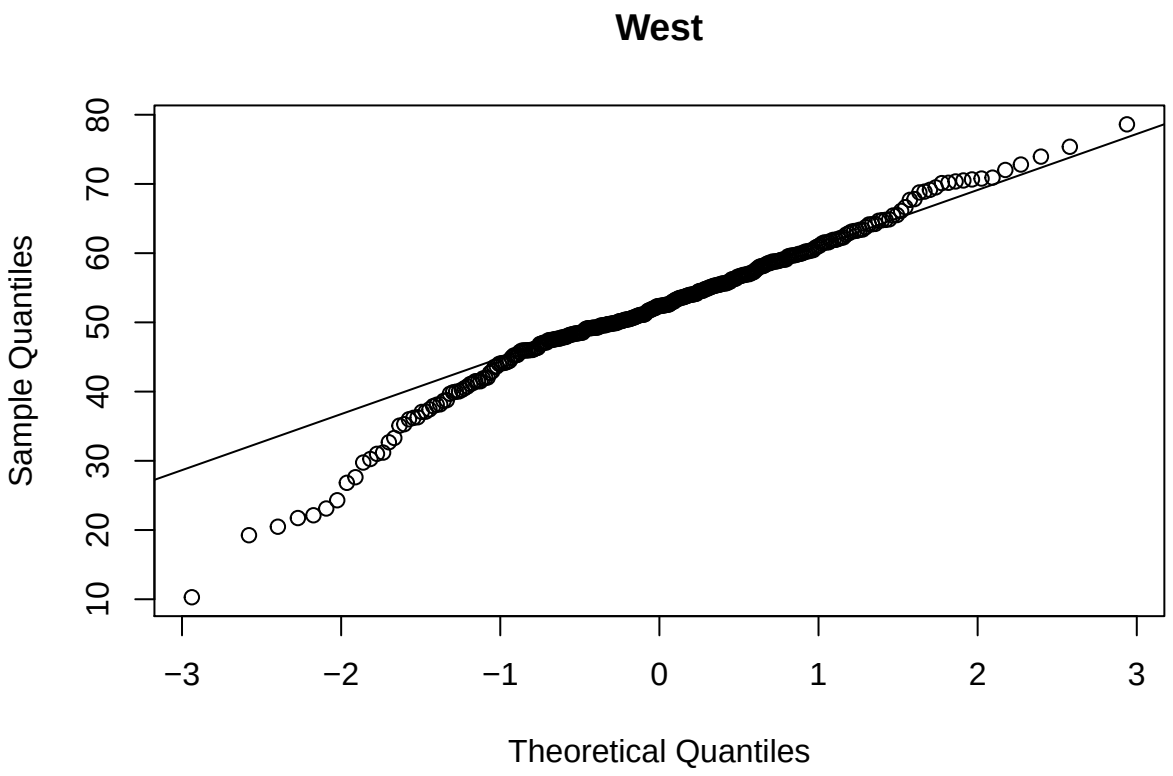



```
northeast = c("Illinois","Indiana","Iowa","Kansas","Michigan","Minnesota","Missouri","Nebraska","Ohio",

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
boxplot(meltedData$value ~ meltedData$states)
```

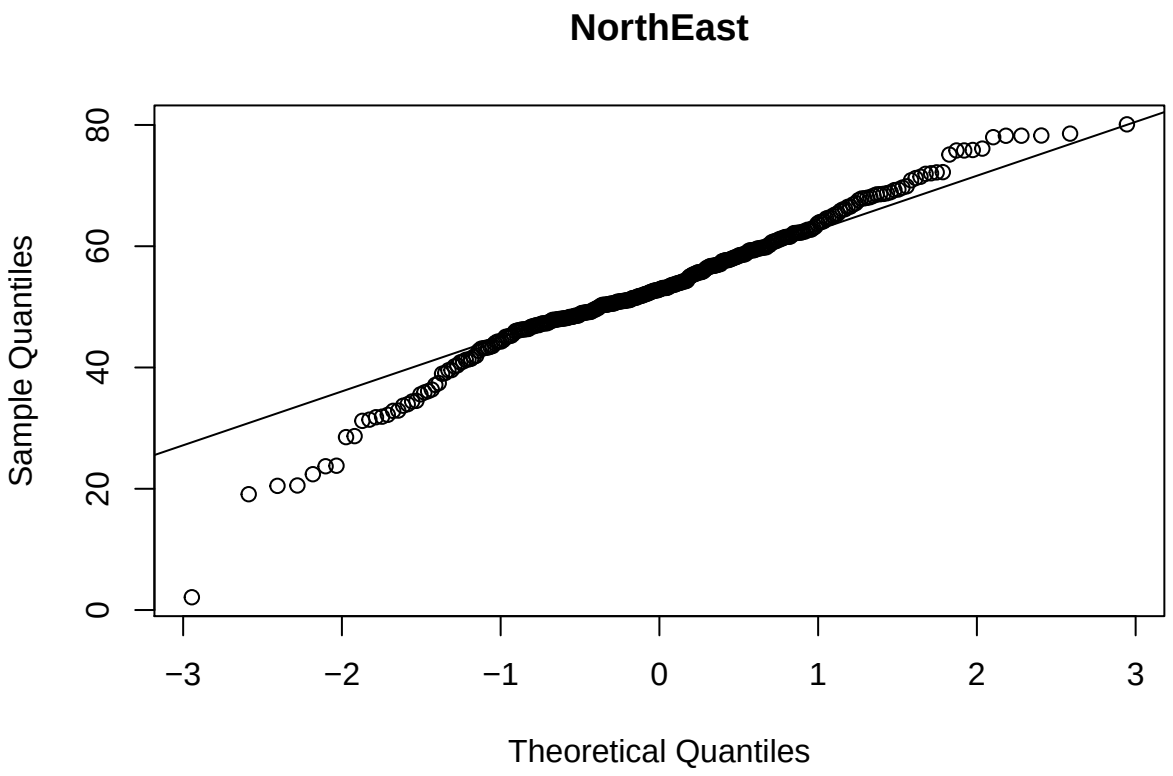


Draw QQ plot for each group and compare how the states are in the each group

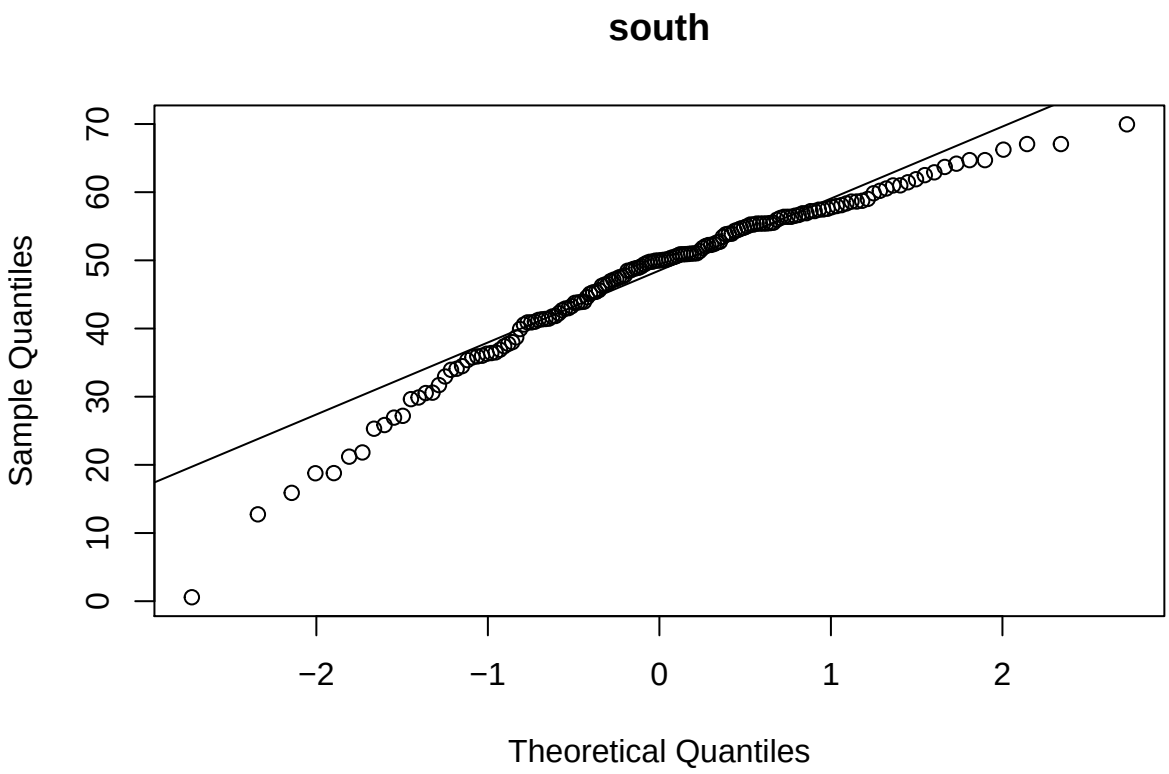
```
northeast = c("Illinois","Indiana","Iowa","Kansas","Michigan","Minnesota","Missouri","Nebraska","Ohio",

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")

qqnorm(meltedData$value, main = "West")
qqline(meltedData$value)
```



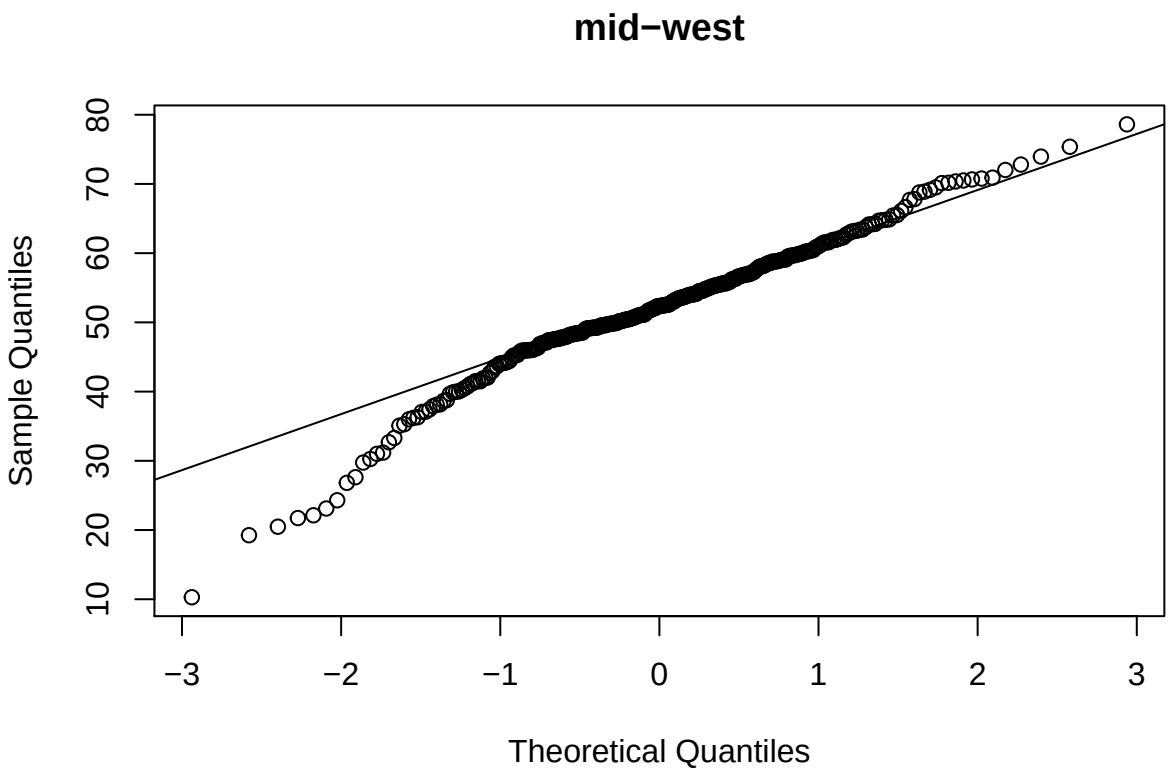

```
northeast = c("Connecticut","Delaware","Maine","Massachusetts","New Hampshire","New Jersey","New York",

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
qqnorm(meltedData$value, main = "NorthEast")
qqline(meltedData$value)
```



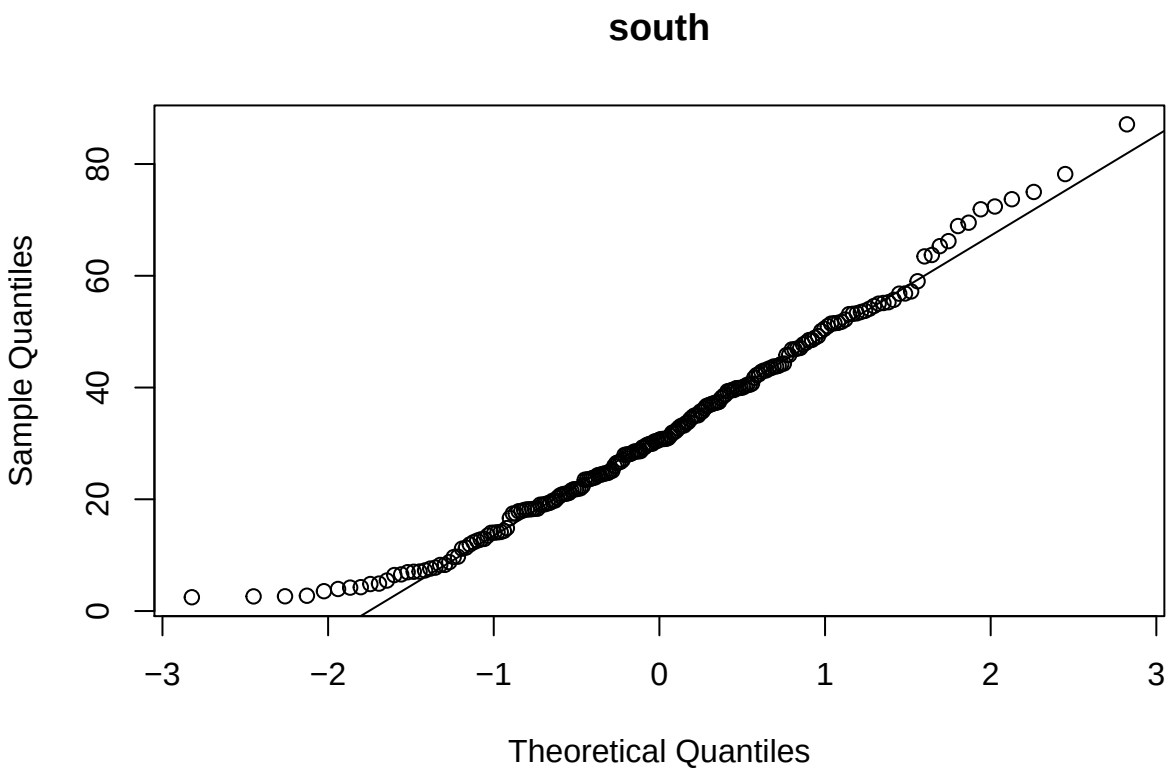

```
northeast = c("Alaska","Arizona","California","Hawaii","Nevada","New Mexico","Oregon","Washington")
library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
qqnorm(meltedData$value, main = "south")
qqline(meltedData$value)
```



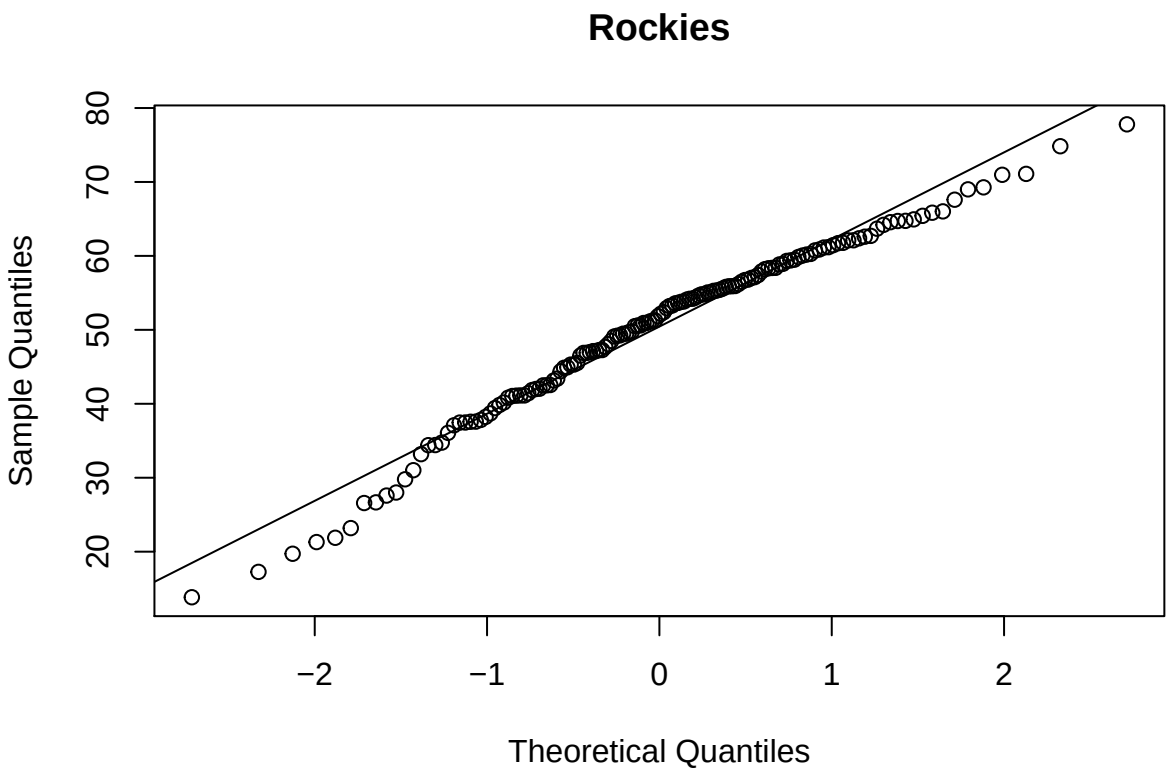

```
northeast = c("Illinois","Indiana","Iowa","Kansas","Michigan","Minnesota","Missouri","Nebraska","Ohio",
library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
qqnorm(meltedData$value, main = "mid-west")
qqline(meltedData$value)
```




```
northeast = c("Alabama","Arkansas","Florida","Georgia","Louisiana","Mississippi","Oklahoma","Texas")

library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
qqnorm(meltedData$value, main = "south")
qqline(meltedData$value)
```




```
northeast = c("Colorado","Idaho","Montana","North Dakota","South Dakota","Utah","Wyoming")
library("cluster")
data(votes.repub)
# line plots to compare the 10 different states
#row.names(votes.repub)
votes.repub$states=row.names(votes.repub)
completeData = votes.repub
library(reshape2)
northeasstData= completeData[northeast,]
TrimNorthEast = northeasstData
library(reshape2)
meltedData <- melt(TrimNorthEast, id="states")
qqnorm(meltedData$value, main = "Rockies")
qqline(meltedData$value)
```


Rockies
Sample Quantiles
20
30
40
50
60
70
80
−2
−1
0
1
2
Theoretical Quantiles

(1) rate = λ

(1) $x_1, x_2 \dots x_n$

$f(x, \lambda) = \lambda \exp(-\lambda x)$

mean $E(x) = \mu = 1/\lambda$

$Var[x] = \sigma^2 = 1/\lambda^2$

median = $x_{0.5} = \frac{\log(2)}{\lambda}$

⇒ two statistics

mean (1) $T_1 = \bar{X} = \frac{\sum_{i=1}^{n} x_i}{n}$

(2) $T_2 = x/\log 2$

(3) $Var(T_1) = \frac{1}{n^2} \sum_{i=1}^{n} var(x_i)$

$\Rightarrow \frac{n \cdot var(x_1)}{n^2} = \frac{\sigma^2}{n} = \frac{1}{n\lambda^2}$

(a) Central limit, what is asymptotic distrib of

$\sqrt{n}\left(\bar{x} - \frac{1}{\lambda}\right)$

The variance will be less than infinity for an Central limit theorem.

⇒ $x_1, x_2, \dots x_n$ is considered IID.

⇒ $\sqrt{n} * \left(\bar{x} - \frac{1}{\lambda}\right)$ ⇒ $\sqrt{n}.(\bar{x} - \mu)$ converges to

$N\mathcal{N}(0, \sigma^2)$

and $\mu = \frac{1}{\lambda}$, which is same as above.

$\sqrt{n} * \left(\bar{x} - \frac{1}{\lambda}\right)$ also converges to $N\mathcal{N}(0,$

(2) Asymptotic distribution of $\sqrt{n}\left(\bar{x} - \frac{\log(2)}{\lambda}\right)$

↳

$$\sqrt{n}\left(\bar{x} - x_{0.5}\right) \Rightarrow N\left(0, \frac{1}{4(f(x_{0.5}))^2}\right)$$

and we are already given that,

$$x_{0.5} = \log(2)/\lambda$$

$$f(0.5) = \lambda \cdot e^{-\lambda \cdot \log 2/\lambda}$$

$$\Rightarrow \lambda/2$$

(a) So, $N\left(0, \frac{1}{4(f(0.5))^2}\right) \Rightarrow N \sim (0, \sigma^2)$ $\because \sigma^2 = 1/\lambda^2$

(3)

Median $= N \sim (0, \sigma^2)$

$$\text{Variance}(T_2) = \left(\frac{1}{\log 2}\right)^2 \cdot \left(\sigma^2/n\right)$$

$\because \text{var}(cT) = c^2 \text{var}(T)$

$$\text{Var}(T_2) = 2.081 \left(\sigma^2/n\right)$$

$$\text{var}(T_2) = 2.08 \frac{\sigma^2}{n}$$

(4)

$$\text{ARE}(T, TT_2) = \text{Var}(T_1) / \text{Var}(T_2)$$

So, from the above information I Boaft did,

$\Rightarrow \left(\frac{\sigma^2}{n}\right) / \left(2.081\left(\frac{\sigma^2}{n}\right)\right)$

$\Rightarrow \frac{1}{2.08} \Rightarrow 0.48$

So, ARE = 48%. Efficient.

---

(5) If we compare the two statistics T_1 & T_2, their variance is to be considered. [The dataset with less variance is better.]

if, $V_1 \lesseqgtr V_2$ then (V1) is better

here $V(T_1) < V(T_2)$ $V(T) \Rightarrow Var(T)$

$Var(T_1) < Var(T_2)$

So, T_1 is better

(6) Just draw the lval Plot.
it's drawn at the first of the assignment.