FINAL_PROJECT2_AS.R

dieudonneouedraogo

Thu May 5 23:53:48 2016

```
data2 <- read.csv("~/Downloads/cleaned_May012016.csv")
#data<- read.csv("~/Downloads/Alex_Clean_April24.csv")
library(pander)
```

```
## Warning: package 'pander' was built under R version 3.1.3
```

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.1.3
```

```
library(moments)  # ... for Skewness
library(ggplot2)  #...  for Graphics
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
##
## Attaching package: 'ggplot2'
##
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
library(pROC)     #...  for ROC
```

```
## Warning: package 'pROC' was built under R version 3.1.3
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(Matrix)   # ... for matrix operations
library(car)      # ... for ellipse plots
```

```
## Warning: package 'car' was built under R version 3.1.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(stats)    # ... for statistical operations
library(MASS)     # ... for Multivariate Normal Distribution
```

```
## Warning: package 'MASS' was built under R version 3.1.3
```

```
library(graphics) # ... for arrows
library(moments)  # ... for Skewness
require(boot)
```

```
## Loading required package: boot
```

```
## Warning: package 'boot' was built under R version 3.1.3
```

```
##
## Attaching package: 'boot'
##
## The following object is masked from 'package:car':
##
##     logit
##
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(lars)
```

```
## Loaded lars 1.2
##
##
## Attaching package: 'lars'
##
## The following object is masked from 'package:psych':
##
##     error.bars
```

```
library (leaps)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.1.3
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.1.3
```

```
## Loaded glmnet 2.0-3
##
##
## Attaching package: 'glmnet'
##
## The following object is masked from 'package:pROC':
##
##     auc
```

```
#setwd("/Users/alexandersatz/Documents/Cuny/IS621/groupProject/May4")

#data2 <- read.csv("cleaned_May012016.csv", stringsAsFactors=TRUE)
head(data2)
```

```
##   CountryCode         Country_Long TARGET MedianAge  AVG_TEMP PER_CAP_INC
## 1         AFG          Afghanistan   38.3      18.0 12.921455    1932.892
## 2         AGO               Angola   45.1      17.9        NA          NA
## 3         ALB              Albania   27.5      31.6 11.269800   11107.968
## 4         ARE United Arab Emirates   70.9      30.3 26.825609   67674.134
## 5         ARG            Argentina   33.1      31.2        NA          NA
## 6         ARM              Armenia   17.9      33.7  6.374362    8069.723
##   LATITUDE LONGITUDE Avg_Per_Unemp CHGENPCT JDGENPCT ISGENPCT BUGENPCT
## 1       33        65          8.69   0.0003   0.0000   0.9956   0.0001
## 2       NA        NA          6.85   0.8912   0.0000   0.0104   0.0001
## 3       41        20         14.04   0.2144   0.0000   0.6300   0.0000
## 4       24        54          3.69   0.0714   0.0000   0.6748   0.0035
## 5       NA        NA          8.74   0.8515   0.0068   0.0151   0.0002
## 6       40        45         19.90   0.9510   0.0002   0.0003   0.0000
##   ZOGENPCT HIGENPCT NORELPCT OtherRelPCT Prohibited SunniPCT ShiaPCT
## 1    1e-04   0.0003   0.0020      0.0016          1   0.8000  0.1900
## 2    0e+00   0.0000   0.0179      0.0804          0   0.0000  0.0000
## 3    0e+00   0.0000   0.1507      0.0049          0   0.6000  0.0300
## 4    0e+00   0.2225   0.0136      0.0142          0   0.5661  0.1087
## 5    0e+00   0.0000   0.1200      0.0064          0   0.0000  0.0000
## 6    0e+00   0.0000   0.0346      0.0139          0   0.0000  0.0000
```

```
attach(data2)
#pander::pander(describe(data2))
# attach(data2)
# detach(data2)
missingVals <- sapply(data2, function(x) sum(is.na(x)))
pander::pander(missingVals)
```

Table 1: Table continues below

CountryCode	Country_Long	TARGET	MedianAge	AVG_TEMP	PER_CAP_INC
0	0	0	0	29	29

Table 2: Table continues below

LATITUDE	LONGITUDE	Avg_Per_Unemp	CHGENPCT	JDGENPCT	ISGENPCT
29	29	0	1	1	1

Table 3: Table continues below

BUGENPCT	ZOGENPCT	HIGENPCT	NORELPCT	OtherRelPCT	Prohibited
1	1	1	1	1	0

BUGENPCT	ZOGENPCT	HIGENPCT	NORELPCT	OtherRelPCT	Prohibited

SunniPCT	ShiaPCT
0	0

```
pander::pander(names(data2))
```

CountryCode, *Country_Long*, *TARGET*, *MedianAge*, *AVG_TEMP*, *PER_CAP_INC*, *LATITUDE*, *LONGITUDE*, *Avg_Per_Unemp*, *CHGENPCT*, *JDGENPCT*, *ISGENPCT*, *BUGENPCT*, *ZOGENPCT*, *HIGENPCT*, *NORELPCT*, *OtherRelPCT*, *Prohibited*, *SunniPCT* and *ShiaPCT*

```
# data2$TARGET<-as.numeric(TARGET)
# data2$MedianAge<-as.numeric(MedianAge)
# data2$AVG_TEMP<-as.numeric(AVG_TEMP)
# data2$PER_CAP_INC<-as.numeric(PER_CAP_INC)
# data2$LATITUDE<-as.numeric(LATITUDE)
# data2$LONGITUDE<-as.numeric(LONGITUDE)
# data2$Avg_Per_Unemp<-as.numeric(Avg_Per_Unemp)
# data2$CHGENPCT<-as.numeric(CHGENPCT)
# data2$JDGENPCT<-as.numeric(JDGENPCT)
# data2$ISGENPCT<-as.numeric(ISGENPCT)
# data2$BUGENPCT<-as.numeric(BUGENPCT)
# data2$ZOGENPCT<-as.numeric(ZOGENPCT)
# data2$HIGENPCT<-as.numeric(HIGENPCT)
# data2$NORELPCT<-as.numeric(NORELPCT)
#data2$OtherRelPCT<-as.numeric(OtherRelPCT)
#str(data2)

#imputting Train data for missing observations with mean
data2[is.na(data2)] <- mean(data2$AVG_TEMP,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$PER_CAP_INC,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$JDGENPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$CHGENPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$ISGENPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$BUGENPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$ZOGENPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$HIGENPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$NORELPCT,na.rm=TRUE)
data2[is.na(data2)] <- mean(data2$OtherRelPCT,na.rm=TRUE)

corel<-cor(data2[,3:ncol(data2)])
pander::pander(corel)
```

Table 5: Table continues below

	TARGET	MedianAge	AVG_TEMP	PER_CAP_INC	LATITUDE
TARGET	1	0.01216	-0.05745	0.01448	0.03797
MedianAge	0.01216	1	-0.6739	0.5478	0.5859

	TARGET	MedianAge	AVG_TEMP	PER_CAP_INC	LATITUDE
AVG_TEMP	-0.05745	-0.6739	1	-0.3316	-0.66
PER_CAP_INC	0.01448	0.5478	-0.3316	1	0.3998
LATITUDE	0.03797	0.5859	-0.66	0.3998	1
LONGITUDE	-0.01892	-0.008465	-0.07753	-0.02409	-0.06814
Avg_Per_Unemp	0.08905	0.04805	-0.09231	-0.07632	-0.02807
CHGENPCT	0.02828	0.1845	-0.1335	0.06541	0.05614
JDGENPCT	0.02367	0.1327	-0.09318	0.06001	0.0888
ISGENPCT	0.01332	0.03611	-0.03016	0.02645	0.1033
BUGENPCT	0.02743	0.1396	-0.09334	0.05465	0.08828
ZOGENPCT	0.02838	0.132	-0.09366	0.05678	0.08703
HIGENPCT	0.04223	0.1302	-0.08489	0.0621	0.08083
NORELPCT	0.03254	0.1763	-0.1323	0.07945	0.1136
OtherRelPCT	0.02902	0.1143	-0.08065	0.04452	0.0701
Prohibited	-0.1493	-0.1177	0.1042	0.01968	0.03762
SunniPCT	-0.02911	-0.3526	0.228	-0.1245	0.0766
ShiaPCT	-0.05963	-0.08746	0.04911	-0.01909	-0.0007984

Table 6: Table continues below

	LONGITUDE	Avg_Per_Unemp	CHGENPCT	JDGENPCT	ISGENPCT
TARGET	-0.01892	0.08905	0.02828	0.02367	0.01332
MedianAge	-0.008465	0.04805	0.1845	0.1327	0.03611
AVG_TEMP	-0.07753	-0.09231	-0.1335	-0.09318	-0.03016
PER_CAP_INC	-0.02409	-0.07632	0.06541	0.06001	0.02645
LATITUDE	-0.06814	-0.02807	0.05614	0.0888	0.1033
LONGITUDE	1	-0.1005	-0.09638	-0.00706	0.03318
Avg_Per_Unemp	-0.1005	1	-0.009735	-0.01072	0.0256
CHGENPCT	-0.09638	-0.009735	1	0.961	0.8722
JDGENPCT	-0.00706	-0.01072	0.961	1	0.9633
ISGENPCT	0.03318	0.0256	0.8722	0.9633	1
BUGENPCT	0.02599	-0.02989	0.9485	0.9932	0.9553
ZOGENPCT	-0.007662	-0.01021	0.963	0.9991	0.9643
HIGENPCT	0.006183	-0.0192	0.9556	0.996	0.9605
NORELPCT	-0.01088	-0.01432	0.9603	0.9953	0.9527
OtherRelPCT	-0.00403	-0.01292	0.9589	0.9969	0.9602
Prohibited	0.1132	0.03831	-0.1025	-0.018	0.08982
SunniPCT	0.1285	0.1513	-0.2368	-0.04457	0.2002
ShiaPCT	0.08208	0.003734	-0.1037	-0.01978	0.09118

Table 7: Table continues below

	BUGENPCT	ZOGENPCT	HIGENPCT	NORELPCT	OtherRelPCT
TARGET	0.02743	0.02838	0.04223	0.03254	0.02902
MedianAge	0.1396	0.132	0.1302	0.1763	0.1143
AVG_TEMP	-0.09334	-0.09366	-0.08489	-0.1323	-0.08065
PER_CAP_INC	0.05465	0.05678	0.0621	0.07945	0.04452
LATITUDE	0.08828	0.08703	0.08083	0.1136	0.0701

	BUGENPCT	ZOGENPCT	HIGENPCT	NORELPCT	OtherRelPCT
LONGITUDE	0.02599	-0.007662	0.006183	-0.01088	-0.00403
Avg_Per_Unemp	-0.02989	-0.01021	-0.0192	-0.01432	-0.01292
CHGENPCT	0.9485	0.963	0.9556	0.9603	0.9589
JDGENPCT	0.9932	0.9991	0.996	0.9953	0.9969
ISGENPCT	0.9553	0.9643	0.9605	0.9527	0.9602
BUGENPCT	1	0.9941	0.9916	0.9902	0.9923
ZOGENPCT	0.9941	1	0.997	0.9962	0.9979
HIGENPCT	0.9916	0.997	1	0.9924	0.9945
NORELPCT	0.9902	0.9962	0.9924	1	0.9944
OtherRelPCT	0.9923	0.9979	0.9945	0.9944	1
Prohibited	-0.02247	-0.0172	-0.01807	-0.02831	-0.02105
SunniPCT	-0.05608	-0.04399	-0.04855	-0.07202	-0.04996
ShiaPCT	-0.02375	-0.01893	-0.02059	-0.03011	-0.02588

	Prohibited	SunniPCT	ShiaPCT
TARGET	-0.1493	-0.02911	-0.05963
MedianAge	-0.1177	-0.3526	-0.08746
AVG_TEMP	0.1042	0.228	0.04911
PER_CAP_INC	0.01968	-0.1245	-0.01909
LATITUDE	0.03762	0.0766	-0.0007984
LONGITUDE	0.1132	0.1285	0.08208
Avg_Per_Unemp	0.03831	0.1513	0.003734
CHGENPCT	-0.1025	-0.2368	-0.1037
JDGENPCT	-0.018	-0.04457	-0.01978
ISGENPCT	0.08982	0.2002	0.09118
BUGENPCT	-0.02247	-0.05608	-0.02375
ZOGENPCT	-0.0172	-0.04399	-0.01893
HIGENPCT	-0.01807	-0.04855	-0.02059
NORELPCT	-0.02831	-0.07202	-0.03011
OtherRelPCT	-0.02105	-0.04996	-0.02588
Prohibited	1	0.3454	0.2523
SunniPCT	0.3454	1	0.04302
ShiaPCT	0.2523	0.04302	1

```
#Median,Mean,Variance,Standard Deviation
apply(data2[,3:ncol(data2)], 2, function(x) mean(x, na.rm=TRUE))
```

```
##        TARGET     MedianAge      AVG_TEMP   PER_CAP_INC      LATITUDE
##  3.206450e+01  2.889467e+01  1.759250e+01  1.562943e+04  2.005811e+01
##     LONGITUDE Avg_Per_Unemp      CHGENPCT      JDGENPCT      ISGENPCT
##  2.034422e+01  7.979823e+00  6.433699e-01  1.090290e-01  3.668172e-01
##      BUGENPCT      ZOGENPCT      HIGENPCT      NORELPCT   OtherRelPCT
##  1.374042e-01  1.041338e-01  1.298136e-01  1.759817e-01  1.603148e-01
##     Prohibited      SunniPCT       ShiaPCT
##  4.733728e-02  2.186071e-01  3.380710e-02
```

```r
apply(data2[,3:ncol(data2)], 2, function(x) median(x, na.rm=TRUE))
```

```
##        TARGET     MedianAge      AVG_TEMP   PER_CAP_INC      LATITUDE
##       29.8000       27.3000       17.5925     8665.4862       17.5925
##     LONGITUDE Avg_Per_Unemp      CHGENPCT      JDGENPCT      ISGENPCT
##       17.5925        6.9700        0.6920        0.0000        0.0460
##      BUGENPCT      ZOGENPCT      HIGENPCT      NORELPCT   OtherRelPCT
##        0.0000        0.0000        0.0000        0.0249        0.0214
##    Prohibited      SunniPCT       ShiaPCT
##        0.0000        0.0204        0.0000
```

```r
apply(data2[,3:ncol(data2)], 2, function(x) sd(x, na.rm=TRUE))
```

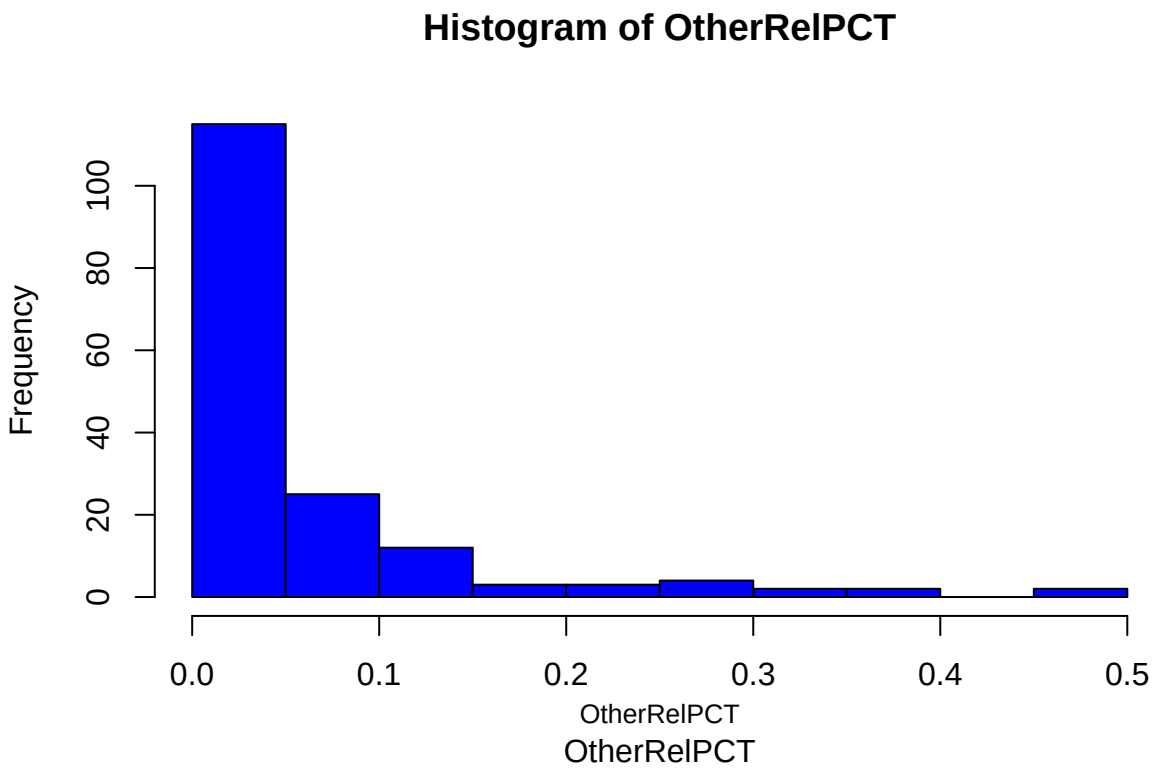
```
##        TARGET     MedianAge      AVG_TEMP   PER_CAP_INC      LATITUDE
##  1.454439e+01  8.562260e+00  7.895389e+00  2.014990e+04  2.307167e+01
##     LONGITUDE Avg_Per_Unemp      CHGENPCT      JDGENPCT      ISGENPCT
##  5.433159e+01  5.989869e+00  1.361984e+00  1.354055e+00  1.382212e+00
##      BUGENPCT      ZOGENPCT      HIGENPCT      NORELPCT   OtherRelPCT
##  1.358680e+00  1.353267e+00  1.355397e+00  1.352892e+00  1.351749e+00
##    Prohibited      SunniPCT       ShiaPCT
##  2.129904e-01  3.319067e-01  1.382015e-01
```

```r
apply(data2[,3:ncol(data2)], 2, function(x) var(x, na.rm=TRUE))
```

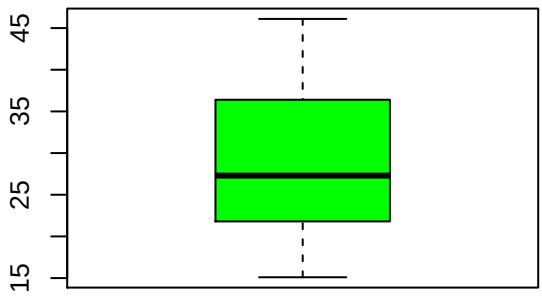
```
##        TARGET     MedianAge      AVG_TEMP   PER_CAP_INC      LATITUDE
##  2.115393e+02  7.331229e+01  6.233717e+01  4.060184e+08  5.323019e+02
##     LONGITUDE Avg_Per_Unemp      CHGENPCT      JDGENPCT      ISGENPCT
##  2.951922e+03  3.587852e+01  1.855000e+00  1.833466e+00  1.910510e+00
##      BUGENPCT      ZOGENPCT      HIGENPCT      NORELPCT   OtherRelPCT
##  1.846012e+00  1.831331e+00  1.837100e+00  1.830317e+00  1.827226e+00
##    Prohibited      SunniPCT       ShiaPCT
##  4.536489e-02  1.101620e-01  1.909966e-02
```

```r
#par(mfrow=c(2,2) , oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)
#pander::pander(describe(TARGET))
#hist(TARGET,col="red")
#mtext("TARGET", side=1, outer=F, line=2, cex=0.8)
#boxplot(TARGET, col="red", pch=19)
#mtext("target", cex=0.8, side=1, line=2)
#hist(PER_CAP_INC,col="blue")
#mtext("Per cap Inc", side=1, outer=F, line=2, cex=0.8)
#hist(MedianAge,col="blue")
#mtext("Median Age", side=1, outer=F, line=2, cex=0.8)
#hist(AVG_TEMP,col="blue")
#mtext("AVG TEMP", side=1, outer=F, line=2, cex=0.8)
#hist(LATITUDE,col="blue")
#mtext("LATITUDE ", side=1, outer=F, line=2, cex=0.8)
#hist(LONGITUDE,col="blue")
#mtext("LONGITUDE", side=1, outer=F, line=2, cex=0.8)
#hist(Avg_Per_Unemp,col="blue")
#mtext("Avg Per Unempl", side=1, outer=F, line=2, cex=0.8)
```

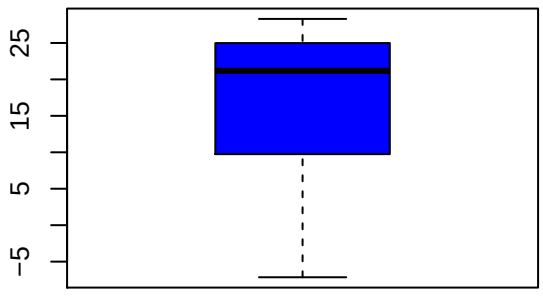
```r
#hist(CHGENPCT,col="blue")
#mtext("CHGENPCT", side=1, outer=F, line=2, cex=0.8)
#hist(JDGENPCT,col="blue")
#mtext("JDGENPCT", side=1, outer=F, line=2, cex=0.8)
#hist(ISGENPCT,col="blue")
#mtext("ISGENPCT", side=1, outer=F, line=2, cex=0.8)
#hist(BUGENPCT,col="blue")
#mtext("BUGENPCT", side=1, outer=F, line=2, cex=0.8)
#hist(ZOGENPCT,col="blue")
#mtext("ZOGENPCT", side=1, outer=F, line=2, cex=0.8)
#hist(HIGENPCT,col="blue")
#mtext("HIGENPCT", side=1, outer=F, line=2, cex=0.8)
#hist(NORELPCT,col="blue")
#mtext("NORELPCT", side=1, outer=F, line=2, cex=0.8)
hist(OtherRelPCT,col="blue")
mtext("OtherRelPCT", side=1, outer=F, line=2, cex=0.8)
```



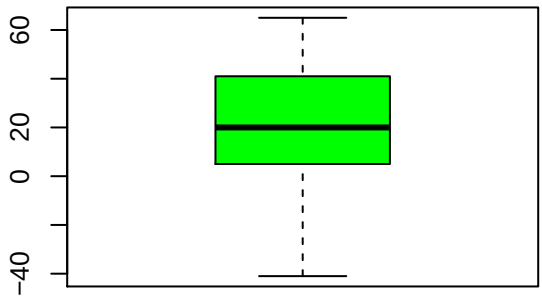

```r
par(mfrow=c(2,2), oma = c(1,1,0,0) + 0.1, mar = c(3,3,1,1) + 0.1)
boxplot(MedianAge, col="green", pch=19)
mtext("Median Age", cex=0.8, side=1, line=2)
boxplot(AVG_TEMP, col="blue", pch=19)
mtext("Average temparature", cex=0.8, side=1, line=2)
boxplot(LATITUDE, col="green", pch=19)
mtext("LATITUDE", cex=0.8, side=1, line=2)
boxplot(LONGITUDE, col="green", pch=19)
mtext("Longitude", cex=0.8, side=1, line=2)
```



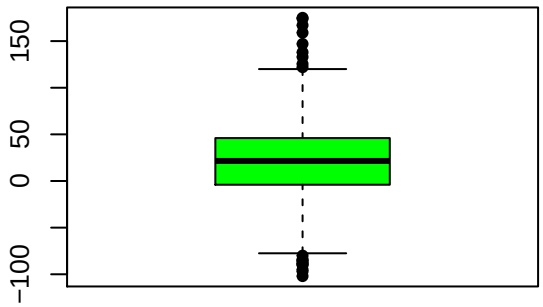
Median Age



Average temparature

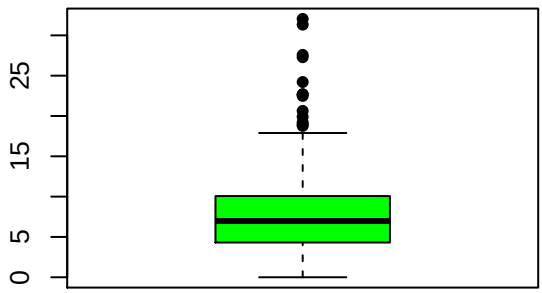


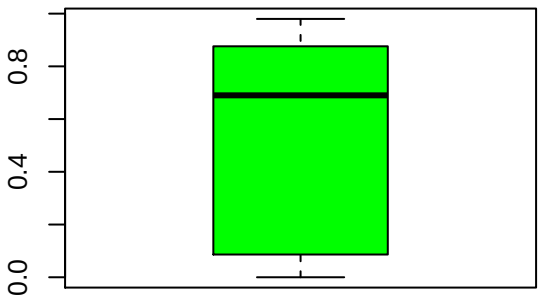
LATITUDE



Longitude

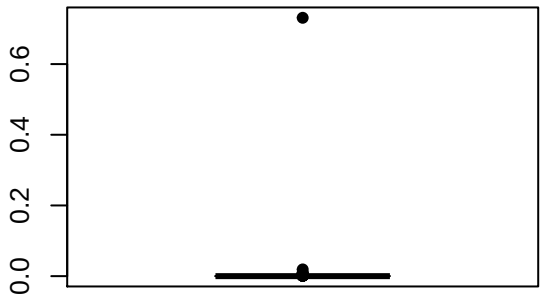
```
boxplot(Avg_Per_Unemp, col="green", pch=19)
mtext("Average per Unemployment", cex=0.8, side=1, line=2)
boxplot(CHGENPCT, col="green", pch=19)
mtext("CHGENPCT", cex=0.8, side=1, line=2)
boxplot(JDGENPCT, col="green", pch=19)
mtext("JDGENPCT", cex=0.8, side=1, line=2)
boxplot(ISGENPCT, col="green", pch=19)
mtext("ISGENPCT", cex=0.8, side=1, line=2)
```

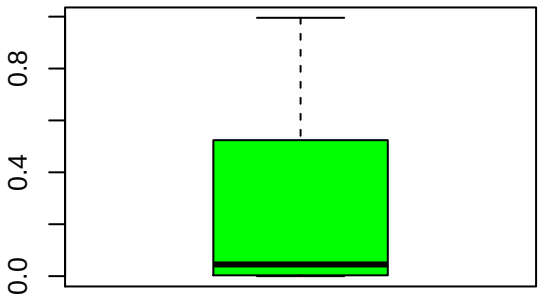




Average per Unemployment

CHGENPCT

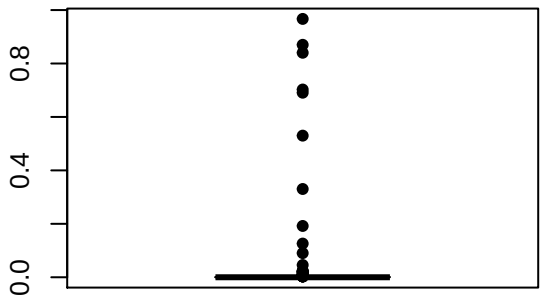




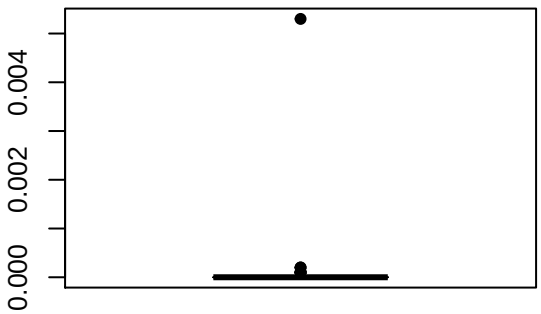
JDGENPCT

ISGENPCT

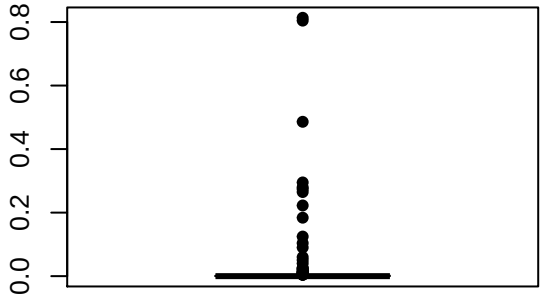
```
boxplot(BUGENPCT, col="green", pch=19)
mtext("BUGENPCT", cex=0.8, side=1, line=2)
boxplot(ZOGENPCT, col="green", pch=19)
mtext("ZOGENPCT", cex=0.8, side=1, line=2)
boxplot(HIGENPCT, col="green", pch=19)
mtext("HIGENPCT", cex=0.8, side=1, line=2)
boxplot(NORELPCT, col="green", pch=19)
mtext("NORELPCT", cex=0.8, side=1, line=2)
```



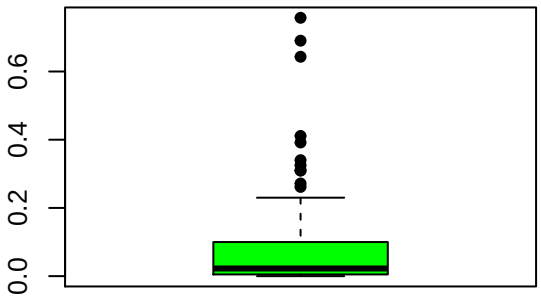
BUGENPCT



ZOGENPCT

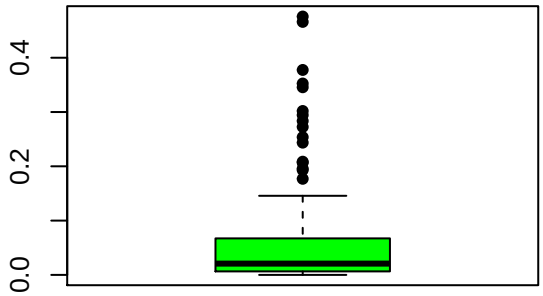


HIGENPCT



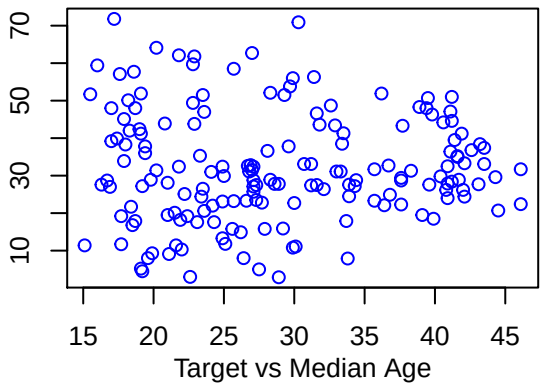
NORELPCT

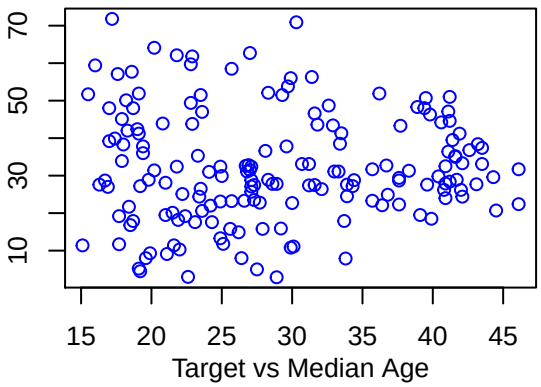
```
boxplot(OtherRelPCT, col="green", pch=19)
mtext("OtherRelPCT", cex=0.8, side=1, line=2)
par(mfrow=c(2,2))
```

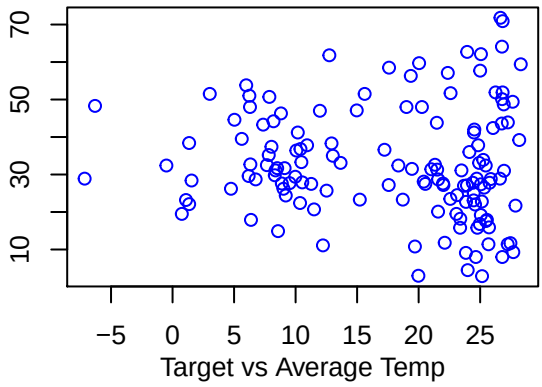


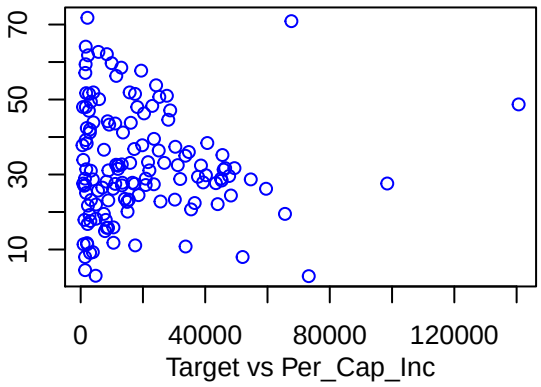
OtherRelPCT

```
plot(TARGET~MedianAge,col="blue")
mtext("Target vs Median Age", side=1, outer=F, line=2, cex=0.8)
plot(TARGET~MedianAge,col="blue")
mtext("Target vs Median Age", side=1, outer=F, line=2, cex=0.8)
plot(TARGET~AVG_TEMP,col="blue")
mtext("Target vs Average Temp", side=1, outer=F, line=2, cex=0.8)
plot(TARGET~PER_CAP_INC,col="blue")
mtext("Target vs Per_Cap_Inc", side=1, outer=F, line=2, cex=0.8)
```

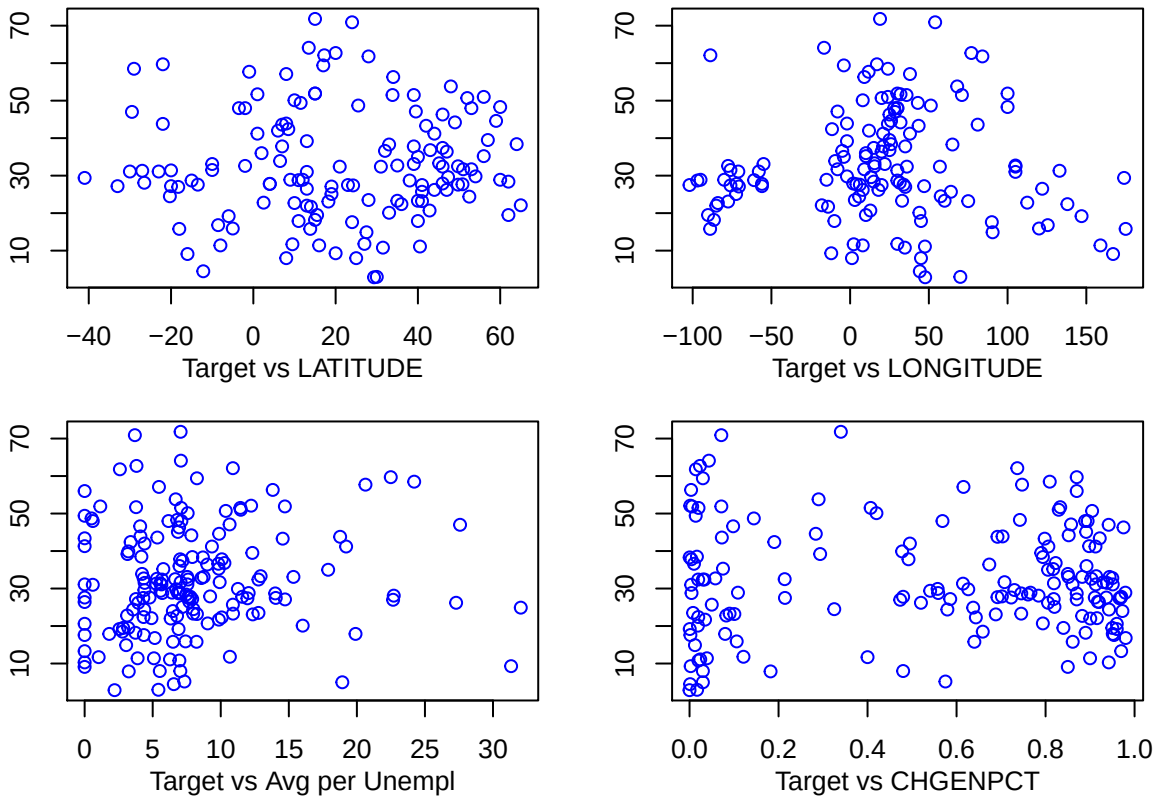




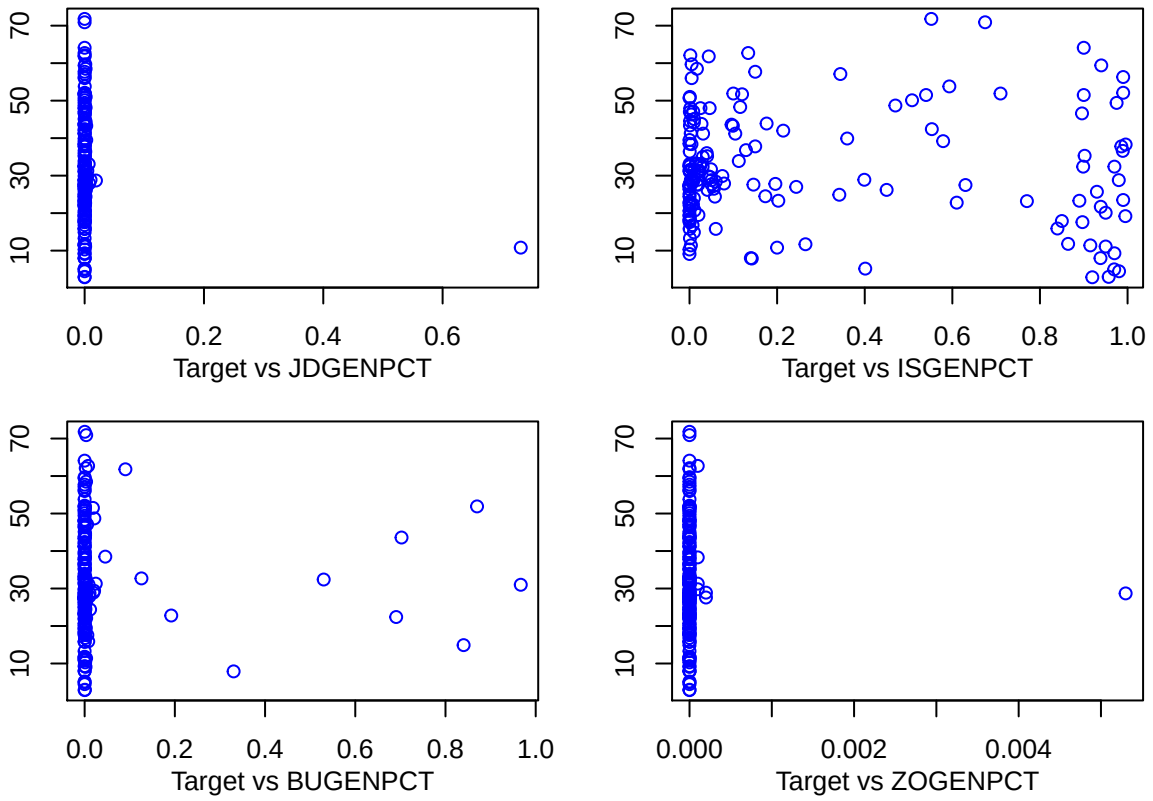





```
plot(TARGET~LATITUDE,col="blue")
mtext("Target vs LATITUDE", side=1, outer=F, line=2, cex=0.8)
plot(TARGET~LONGITUDE,col="blue")
mtext("Target vs LONGITUDE", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~Avg_Per_Unemp,col="blue")
mtext("Target vs Avg per Unempl", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~CHGENPCT,col="blue")
mtext("Target vs CHGENPCT", side=1, outer=F, line=2,cex=0.8)
```




```
plot(TARGET~JDGENPCT,col="blue")
mtext("Target vs JDGENPCT", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~ISGENPCT,col="blue")
mtext("Target vs ISGENPCT", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~BUGENPCT,col="blue")
mtext("Target vs BUGENPCT", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~ZOGENPCT,col="blue")
mtext("Target vs ZOGENPCT", side=1, outer=F, line=2,cex=0.8)
```



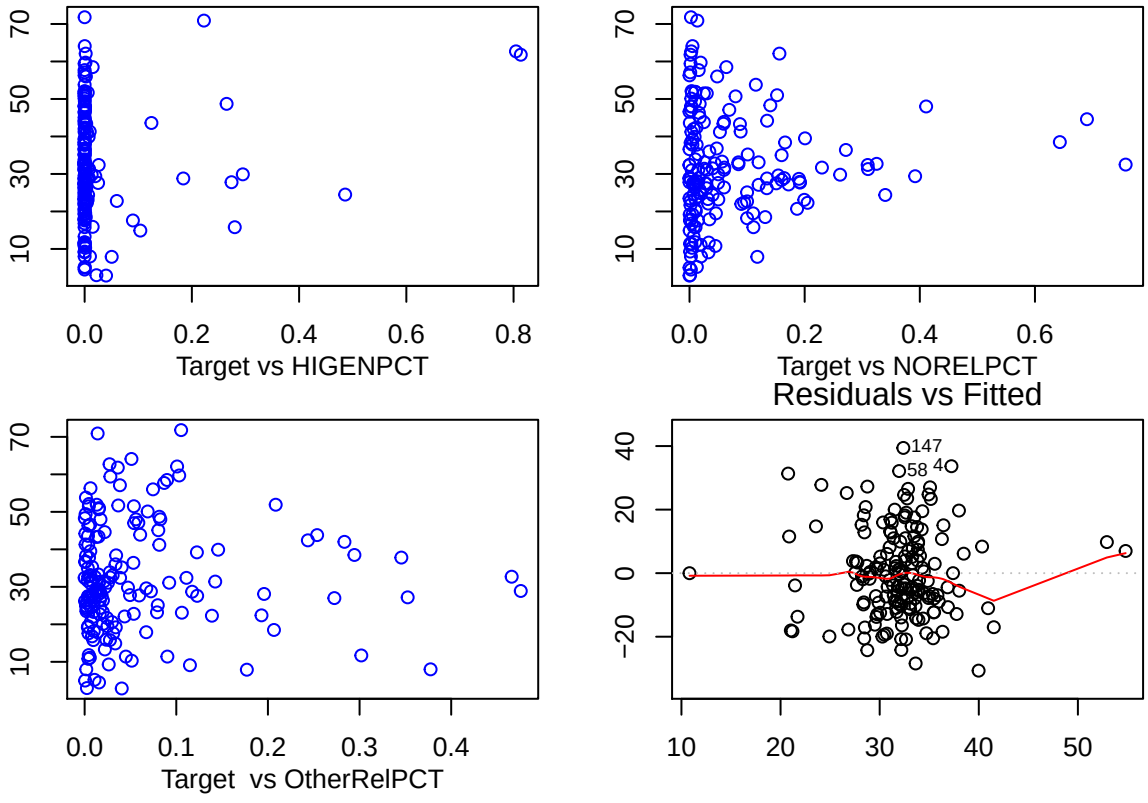

```
plot(TARGET~HIGENPCT,col="blue")
mtext("Target vs HIGENPCT", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~NORELPCT,col="blue")
mtext("Target vs NORELPCT", side=1, outer=F, line=2,cex=0.8)
plot(TARGET~OtherRelPCT,col="blue")
mtext("Target  vs OtherRelPCT", side=1, outer=F, line=2,cex=0.8)

#############################################################################################
## The model with all data and predictors
data2['absLatitude'] <- abs(data2$LATITUDE)
model1<-lm(TARGET~.,data=data2[,3:ncol(data2)])
summary(model1)  ## Not a single predictor is significant, including whether alcohol is legal!
```

```
##
## Call:
## lm(formula = TARGET ~ ., data = data2[, 3:ncol(data2)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.682  -9.629  -1.026   9.426  39.414
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.054e+02  3.622e+02  -0.291   0.7715
## MedianAge     -2.415e-01  2.412e-01  -1.001   0.3184
## AVG_TEMP      -1.988e-01  3.320e-01  -0.599   0.5502
## PER_CAP_INC    4.685e-05  7.836e-05   0.598   0.5509
## LATITUDE       6.483e-02  8.546e-02   0.759   0.4493
```

```
## LONGITUDE     -5.649e-03  2.501e-02  -0.226   0.8216
## Avg_Per_Unemp  3.394e-01  2.152e-01   1.577   0.1169
## CHGENPCT       1.448e+02  3.639e+02   0.398   0.6912
## JDGENPCT       1.167e+02  3.649e+02   0.320   0.7496
## ISGENPCT       1.227e+02  3.643e+02   0.337   0.7367
## BUGENPCT       1.458e+02  3.644e+02   0.400   0.6897
## ZOGENPCT      -9.969e+02  2.528e+03  -0.394   0.6939
## HIGENPCT       1.748e+02  3.644e+02   0.480   0.6322
## NORELPCT       1.547e+02  3.646e+02   0.424   0.6719
## OtherRelPCT    1.460e+02  3.640e+02   0.401   0.6889
## Prohibited    -1.068e+01  5.869e+00  -1.820   0.0708 .
## SunniPCT       2.252e+01  2.440e+01   0.923   0.3575
## ShiaPCT        2.058e+01  2.552e+01   0.807   0.4211
## absLatitude   -8.824e-02  2.009e-01  -0.439   0.6611
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.59 on 150 degrees of freedom
## Multiple R-squared:  0.1013, Adjusted R-squared:  -0.006544
## F-statistic: 0.9393 on 18 and 150 DF,  p-value: 0.5328
```

```
plot(model1)
```

```
## Warning: not plotting observations with leverage one:
##   142
```




```
## Warning: not plotting observations with leverage one:
##   142
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

```
##############################################################################################################
##############################################################################################################
##############################################################################################################
##### below is modeling data after removing those countries with a high ISGENPCT.
summary(data2)
```

```
##   CountryCode       Country_Long     TARGET        MedianAge
##  AFG    :  1   Afghanistan:  1   Min.   : 2.90   Min.   :15.10
##  AGO    :  1   Albania    :  1   1st Qu.:22.80   1st Qu.:21.80
##  ALB    :  1   Algeria    :  1   Median :29.80   Median :27.30
##  ARE    :  1   Angola     :  1   Mean   :32.06   Mean   :28.89
##  ARG    :  1   Argentina  :  1   3rd Qu.:42.00   3rd Qu.:36.40
##  ARM    :  1   Armenia    :  1   Max.   :71.80   Max.   :46.10
##  (Other):163   (Other)    :163
##     AVG_TEMP       PER_CAP_INC          LATITUDE        LONGITUDE
##  Min.   :-7.145   Min.   :    17.59   Min.   :-41.00   Min.   :-102.00
##  1st Qu.:10.934   1st Qu.:  1619.54   1st Qu.:  8.50   1st Qu.:   3.00
##  Median :17.593   Median :  8665.49   Median : 17.59   Median :  17.59
##  Mean   :17.593   Mean   : 15629.43   Mean   : 20.06   Mean   :  20.34
##  3rd Qu.:24.528   3rd Qu.: 22989.58   3rd Qu.: 39.00   3rd Qu.:  38.00
##  Max.   :28.300   Max.   :140649.17   Max.   : 65.00   Max.   : 175.00
##
##  Avg_Per_Unemp      CHGENPCT          JDGENPCT          ISGENPCT
##  Min.   : 0.00   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.: 4.31   1st Qu.: 0.0900   1st Qu.: 0.0000   1st Qu.: 0.0032
##  Median : 6.97   Median : 0.6920   Median : 0.0000   Median : 0.0460
##  Mean   : 7.98   Mean   : 0.6434   Mean   : 0.1090   Mean   : 0.3668
##  3rd Qu.:10.07   3rd Qu.: 0.8822   3rd Qu.: 0.0004   3rd Qu.: 0.5400
##  Max.   :32.04   Max.   :17.5925   Max.   :17.5925   Max.   :17.5925
##
##     BUGENPCT          ZOGENPCT          HIGENPCT          NORELPCT
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0050
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0249
##  Mean   : 0.1374   Mean   : 0.1041   Mean   : 0.1298   Mean   : 0.1760
##  3rd Qu.: 0.0013   3rd Qu.: 0.0000   3rd Qu.: 0.0016   3rd Qu.: 0.1000
##  Max.   :17.5925   Max.   :17.5925   Max.   :17.5925   Max.   :17.5925
##
##   OtherRelPCT        Prohibited         SunniPCT         ShiaPCT
##  Min.   : 0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 0.0064   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 0.0214   Median :0.00000   Median :0.0204   Median :0.00000
##  Mean   : 0.1603   Mean   :0.04734   Mean   :0.2186   Mean   :0.03381
##  3rd Qu.: 0.0673   3rd Qu.:0.00000   3rd Qu.:0.3250   3rd Qu.:0.00520
##  Max.   :17.5925   Max.   :1.00000   Max.   :0.9900   Max.   :0.98060
##
##   absLatitude
##  Min.   : 1.00
##  1st Qu.:15.00
```

```
##  Median :19.00
##  Mean   :25.88
##  3rd Qu.:39.50
##  Max.   :65.00
##
```

```
data3 <- data2[data2[12] <0.8,]  ## data without large islam
data4 <- data2[data2[12] >0.8,]  ## data with only large islam

# function will split the data set into  training and testing sets
splitdf <- function(dataframe, seed=NULL) {
  if (!is.null(seed)) set.seed(seed)
  index <- 1:nrow(dataframe)
  trainindex <- sample(index, trunc(length(index)/2))
  trainset <- dataframe[trainindex, ]
  testset <- dataframe[-trainindex, ]
  list(trainset=trainset,testset=testset)
}

#appling the function
splits <- splitdf(data3, seed=1306)

#Returns two data frames called trainset and testset
str(splits)
```

```
## List of 2
##  $ trainset:'data.frame':    68 obs. of  21 variables:
##   ..$ CountryCode  : Factor w/ 169 levels "AFG","AGO","ALB",..: 130 30 79 13 114 148 61 136 17 38 ..
##   ..$ Country_Long : Factor w/ 169 levels "Afghanistan",..: 125 32 77 24 112 152 59 137 10 57 ...
##   ..$ TARGET       : num [1:68] 48.3 32.7 22.4 39.2 19.5 8 33.1 11.4 27.4 31.7 ...
##   ..$ MedianAge    : num [1:68] 38.9 36.7 46.1 17 39.1 19.6 43.5 21.6 31.2 46.1 ...
##   ..$ AVG_TEMP     : num [1:68] -6.3 6.326 10.363 28.176 0.754 ...
##   ..$ PER_CAP_INC  : num [1:68] 22990 13206 36619 1620 65614 ...
##   ..$ LATITUDE     : num [1:68] 60 35 36 13 62 ...
##   ..$ LONGITUDE    : num [1:68] 100 105 138 -2 10 ...
##   ..$ Avg_Per_Unemp: num [1:68] 6.8 4.34 4.33 3.15 3.21 ...
##   ..$ CHGENPCT     : num [1:68] 0.7423 0.058 0.0196 0.2935 0.8401 ...
##   ..$ JDGENPCT     : num [1:68] 0.0014 0 0 0 0.0002 0 0.0005 0 0.001 0.0015 ...
##   ..$ ISGENPCT     : num [1:68] 0.1157 0.025 0.0015 0.579 0.0204 ...
##   ..$ BUGENPCT     : num [1:68] 0 0.126 0.6907 0 0.0028 ...
##   ..$ ZOGENPCT     : num [1:68] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ HIGENPCT     : num [1:68] 0 0 0.0002 0 0.0011 0 0 0 0 0.0012 ...
##   ..$ NORELPCT     : num [1:68] 0.14 0.325 0.095 0.005 0.111 ...
##   ..$ OtherRelPCT  : num [1:68] 0.0002 0.466 0.193 0.1225 0.0249 ...
##   ..$ Prohibited   : int [1:68] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ SunniPCT     : num [1:68] 0.1 0 0 0.57 0.0204 0.14 0.0212 0 0 0.0355 ...
##   ..$ ShiaPCT      : num [1:68] 0 0 0 0.009 0 0 0.0028 0 0 0.0034 ...
##   ..$ absLatitude  : num [1:68] 60 35 36 13 62 ...
##  $ testset :'data.frame':    68 obs. of  21 variables:
##   ..$ CountryCode  : Factor w/ 169 levels "AFG","AGO","ALB",..: 4 5 7 8 10 11 15 19 20 23 ...
##   ..$ Country_Long : Factor w/ 169 levels "Afghanistan",..: 161 5 7 8 25 15 23 14 16 13 ...
##   ..$ TARGET       : num [1:68] 70.9 33.1 31.3 29.6 48 27.7 36.8 48 62.1 22.3 ...
##   ..$ MedianAge    : num [1:68] 30.3 31.2 38.3 44.3 17 43.1 42.6 39.4 21.8 37.6 ...
##   ..$ AVG_TEMP     : num [1:68] 26.83 17.59 21.51 6.19 20.27 ...
```

```
##   ..$ PER_CAP_INC  : num [1:68] 67674.1 17.6 45925.5 47682.3 769.9 ...
##   ..$ LATITUDE     : num [1:68] 24 17.6 -27 47.3 -3.5 ...
##   ..$ LONGITUDE    : num [1:68] 54 17.6 133 13.3 30 ...
##   ..$ Avg_Per_Unemp: num [1:68] 3.69 8.74 5.25 4.39 7.11 ...
##   ..$ CHGENPCT     : num [1:68] 0.0714 0.8515 0.6145 0.73 0.889 ...
##   ..$ JDGENPCT     : num [1:68] 0 0.0068 0.0045 0.001 0 0.0028 0.0003 0.0013 0 0.0001 ...
##   ..$ ISGENPCT     : num [1:68] 0.6748 0.0151 0.0223 0.0475 0.025 ...
##   ..$ BUGENPCT     : num [1:68] 0.0035 0.0002 0.0247 0 0 0.003 0 0 0.0025 0.0004 ...
##   ..$ ZOGENPCT     : num [1:68] 0e+00 0e+00 1e-04 0e+00 0e+00 0e+00 0e+00 0e+00 0e+00 0e+00 ...
##   ..$ HIGENPCT     : num [1:68] 0.2225 0 0.0084 0 0.0008 ...
##   ..$ NORELPCT     : num [1:68] 0.0136 0.12 0.3101 0.1542 0.0026 ...
##   ..$ OtherRelPCT  : num [1:68] 0.0142 0.0064 0.0154 0.0673 0.0826 ...
##   ..$ Prohibited   : int [1:68] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ SunniPCT     : num [1:68] 0.566 0 0 0 0.024 ...
##   ..$ ShiaPCT      : num [1:68] 0.109 0 0 0 0.001 ...
##   ..$ absLatitude  : num [1:68] 24 17.6 27 47.3 3.5 ...
```

```
# There are 68 observation in the train and 68 observations in the test data frame
lapply(splits,nrow)
```

```
## $trainset
## [1] 68
##
## $testset
## [1] 68
```

```
#view the first couple columns in each data frame
lapply(splits,head)
```

```
## $trainset
##     CountryCode       Country_Long TARGET MedianAge   AVG_TEMP PER_CAP_INC
## 130         RUS Russian Federation   48.3      38.9 -6.3000000   22989.578
## 30          CHN              China   32.7      36.7  6.3256640   13206.384
## 79          JPN              Japan   22.4      46.1 10.3627259   36619.426
## 13          BFA       Burkina Faso   39.2      17.0 28.1763605    1619.541
## 114         NOR             Norway   19.5      39.1  0.7538925   65614.481
## 148         TGO               Togo    8.0      19.6 26.7991114    1428.821
##     LATITUDE LONGITUDE Avg_Per_Unemp CHGENPCT JDGENPCT ISGENPCT BUGENPCT
## 130       60  100.0000          6.80   0.7423   0.0014   0.1157   0.0000
## 30        35  105.0000          4.34   0.0580   0.0000   0.0250   0.1260
## 79        36  138.0000          4.33   0.0196   0.0000   0.0015   0.6907
## 13        13   -2.0000          3.15   0.2935   0.0000   0.5790   0.0000
## 114       62   10.0000          3.21   0.8401   0.0002   0.0204   0.0028
## 148        8    1.1667          7.03   0.4800   0.0000   0.1400   0.0000
##     ZOGENPCT HIGENPCT NORELPCT OtherRelPCT Prohibited SunniPCT ShiaPCT
## 130        0   0.0000   0.1404      0.0002          0   0.1000   0.000
## 30         0   0.0000   0.3250      0.4660          0   0.0000   0.000
## 79         0   0.0002   0.0950      0.1930          0   0.0000   0.000
## 13         0   0.0000   0.0050      0.1225          0   0.5700   0.009
## 114        0   0.0011   0.1105      0.0249          0   0.0204   0.000
## 148        0   0.0000   0.0025      0.3775          0   0.1400   0.000
##     absLatitude
## 130          60
```

```
## 30           35
## 79           36
## 13           13
## 114          62
## 148           8
##
## $testset
##    CountryCode         Country_Long TARGET MedianAge  AVG_TEMP PER_CAP_INC
## 4          ARE United Arab Emirates   70.9      30.3 26.825609  67674.1345
## 5          ARG            Argentina   33.1      31.2 17.592505     17.5925
## 7          AUS            Australia   31.3      38.3 21.506676  45925.4938
## 8          AUT              Austria   29.6      44.3  6.186013  47682.2998
## 10         BDI              Burundi   48.0      17.0 20.266100    769.8822
## 11         BEL              Belgium   27.7      43.1  9.514241  43434.7178
##    LATITUDE LONGITUDE Avg_Per_Unemp CHGENPCT JDGENPCT ISGENPCT BUGENPCT
## 4   24.0000   54.0000          3.69   0.0714   0.0000   0.6748   0.0035
## 5   17.5925   17.5925          8.74   0.8515   0.0068   0.0151   0.0002
## 7  -27.0000  133.0000          5.25   0.6145   0.0045   0.0223   0.0247
## 8   47.3333   13.3333          4.39   0.7300   0.0010   0.0475   0.0000
## 10  -3.5000   30.0000          7.11   0.8890   0.0000   0.0250   0.0000
## 11  50.8333    4.0000          7.70   0.6920   0.0028   0.0500   0.0030
##    ZOGENPCT HIGENPCT NORELPCT OtherRelPCT Prohibited SunniPCT ShiaPCT
## 4     0e+00   0.2225   0.0136      0.0142          0   0.5661  0.1087
## 5     0e+00   0.0000   0.1200      0.0064          0   0.0000  0.0000
## 7     1e-04   0.0084   0.3101      0.0154          0   0.0000  0.0000
## 8     0e+00   0.0000   0.1542      0.0673          0   0.0000  0.0000
## 10    0e+00   0.0008   0.0026      0.0826          0   0.0240  0.0010
## 11    0e+00   0.0007   0.1920      0.0595          0   0.0450  0.0050
##    absLatitude
## 4      24.0000
## 5      17.5925
## 7      27.0000
## 8      47.3333
## 10      3.5000
## 11     50.8333
```

```
# save the training and testing sets as data frames
training <- splits$trainset
testing <- splits$testset



#Regression Model using all 17 predictors from Training Data Set without transformation
mod1<-lm(TARGET~MedianAge+AVG_TEMP+PER_CAP_INC
               +Avg_Per_Unemp+CHGENPCT+JDGENPCT
               +ISGENPCT+BUGENPCT+ZOGENPCT+HIGENPCT
               +NORELPCT+OtherRelPCT+Prohibited
               +SunniPCT+ShiaPCT+LATITUDE+LONGITUDE,data=training)
summary(mod1)  ## medianage, avgtemp, income, higen, and shia are all sign
```

```
##
## Call:
## lm(formula = TARGET ~ MedianAge + AVG_TEMP + PER_CAP_INC + Avg_Per_Unemp +
##     CHGENPCT + JDGENPCT + ISGENPCT + BUGENPCT + ZOGENPCT + HIGENPCT +
```

```
##     NORELPCT + OtherRelPCT + Prohibited + SunniPCT + ShiaPCT +
##     LATITUDE + LONGITUDE, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.5883  -7.0909  -0.3138   7.5599  26.1132
##
## Coefficients: (2 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.405e+01  2.068e+01   2.130   0.0379 *
## MedianAge     -6.360e-01  3.011e-01  -2.113   0.0395 *
## AVG_TEMP      -8.850e-01  3.814e-01  -2.320   0.0243 *
## PER_CAP_INC   -3.559e-05  1.119e-04  -0.318   0.7518
## Avg_Per_Unemp  3.226e-01  2.675e-01   1.206   0.2332
## CHGENPCT       1.914e+01  1.839e+01   1.041   0.3027
## JDGENPCT       4.223e+00  2.435e+01   0.173   0.8630
## ISGENPCT      -1.025e+02  8.197e+01  -1.251   0.2167
## BUGENPCT       3.394e+01  2.153e+01   1.576   0.1210
## ZOGENPCT       2.486e+04  5.509e+04   0.451   0.6536
## HIGENPCT       5.243e+01  2.016e+01   2.601   0.0121 *
## NORELPCT       3.053e+01  2.449e+01   1.247   0.2180
## OtherRelPCT            NA         NA      NA       NA
## Prohibited             NA         NA      NA       NA
## SunniPCT       1.159e+02  7.756e+01   1.495   0.1410
## ShiaPCT        4.531e+02  2.131e+02   2.127   0.0382 *
## LATITUDE      -4.892e-02  1.206e-01  -0.406   0.6866
## LONGITUDE     -4.206e-02  3.497e-02  -1.203   0.2346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.27 on 52 degrees of freedom
## Multiple R-squared:  0.3583, Adjusted R-squared:  0.1732
## F-statistic: 1.936 on 15 and 52 DF,  p-value: 0.04041
```

```
mod.t<-lm(TARGET~MedianAge+AVG_TEMP+PER_CAP_INC
         +Avg_Per_Unemp+CHGENPCT+JDGENPCT
         +ISGENPCT+BUGENPCT+ZOGENPCT+HIGENPCT
         +NORELPCT+OtherRelPCT+Prohibited
         +SunniPCT+ShiaPCT+LATITUDE+LONGITUDE,data=testing)
summary(mod.t)  ## only unemployment is sig
```

```
##
## Call:
## lm(formula = TARGET ~ MedianAge + AVG_TEMP + PER_CAP_INC + Avg_Per_Unemp +
##     CHGENPCT + JDGENPCT + ISGENPCT + BUGENPCT + ZOGENPCT + HIGENPCT +
##     NORELPCT + OtherRelPCT + Prohibited + SunniPCT + ShiaPCT +
##     LATITUDE + LONGITUDE, data = testing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.814  -5.809   0.125   5.443  28.267
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    6.538e+00  2.107e+01   0.310   0.7576
## MedianAge      1.582e-01  3.651e-01   0.433   0.6667
## AVG_TEMP       4.181e-01  3.386e-01   1.235   0.2226
## PER_CAP_INC    1.024e-05  1.126e-04   0.091   0.9278
## Avg_Per_Unemp  7.140e-01  3.155e-01   2.263   0.0279 *
## CHGENPCT       7.642e+00  1.968e+01   0.388   0.6994
## JDGENPCT      -3.700e+02  1.194e+03  -0.310   0.7579
## ISGENPCT      -4.797e+01  9.428e+01  -0.509   0.6131
## BUGENPCT       1.014e+01  2.253e+01   0.450   0.6545
## ZOGENPCT       5.848e+02  4.604e+03   0.127   0.8994
## HIGENPCT       1.135e+01  4.375e+01   0.259   0.7964
## NORELPCT       2.306e+01  3.015e+01   0.765   0.4479
## OtherRelPCT            NA         NA      NA       NA
## Prohibited    -5.768e+00  1.630e+01  -0.354   0.7248
## SunniPCT       8.190e+01  8.849e+01   0.926   0.3590
## ShiaPCT        1.145e+02  1.015e+02   1.127   0.2648
## LATITUDE       7.875e-02  9.731e-02   0.809   0.4221
## LONGITUDE      1.000e-02  3.145e-02   0.318   0.7517
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.68 on 51 degrees of freedom
## Multiple R-squared:  0.2966, Adjusted R-squared:  0.07596
## F-statistic: 1.344 on 16 and 51 DF,  p-value: 0.2082
```

```
### test and training set are too different.  Cannot have a test set.###################################

splitdf <- function(dataframe, seed=NULL) {
  if (!is.null(seed)) set.seed(seed)
  index <- 1:nrow(dataframe)
  trainindex <- sample(index, trunc(length(index)/1))  ## alter so that there is only a training set
  trainset <- dataframe[trainindex, ]
  testset <- dataframe[-trainindex, ]
  list(trainset=trainset,testset=testset)
}

#appling the function
splits <- splitdf(data3, seed=1306)


training <- splits$trainset
mod1<-lm(TARGET~MedianAge+AVG_TEMP+PER_CAP_INC
         +Avg_Per_Unemp+CHGENPCT+JDGENPCT
         +ISGENPCT+BUGENPCT+ZOGENPCT+HIGENPCT
         +NORELPCT+OtherRelPCT+Prohibited
         +SunniPCT+ShiaPCT+LATITUDE+LONGITUDE+absLatitude,data=training)
summary(mod1)  ## avgtemp, higen, and sunni, and shia are sign
```

```
##
## Call:
## lm(formula = TARGET ~ MedianAge + AVG_TEMP + PER_CAP_INC + Avg_Per_Unemp +
##     CHGENPCT + JDGENPCT + ISGENPCT + BUGENPCT + ZOGENPCT + HIGENPCT +
##     NORELPCT + OtherRelPCT + Prohibited + SunniPCT + ShiaPCT +
##     LATITUDE + LONGITUDE + absLatitude, data = training)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.461  -8.086  -0.031   7.353  35.307
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.809e+01  1.569e+01   1.791  0.07588 .
## MedianAge     -2.711e-01  2.309e-01  -1.174  0.24273
## AVG_TEMP      -6.930e-02  3.374e-01  -0.205  0.83761
## PER_CAP_INC    3.206e-05  7.536e-05   0.425  0.67133
## Avg_Per_Unemp  4.601e-01  2.095e-01   2.196  0.03002 *
## CHGENPCT       7.819e+00  1.309e+01   0.597  0.55146
## JDGENPCT      -1.991e+01  2.125e+01  -0.937  0.35062
## ISGENPCT      -9.392e+01  5.357e+01  -1.753  0.08215 .
## BUGENPCT       1.050e+01  1.516e+01   0.693  0.48996
## ZOGENPCT      -8.339e+02  2.532e+03  -0.329  0.74244
## HIGENPCT       3.955e+01  1.564e+01   2.528  0.01279 *
## NORELPCT       2.218e+01  1.843e+01   1.203  0.23125
## OtherRelPCT           NA         NA      NA       NA
## Prohibited     1.338e+01  1.372e+01   0.975  0.33139
## SunniPCT       1.072e+02  5.044e+01   2.126  0.03560 *
## ShiaPCT        2.034e+02  6.746e+01   3.016  0.00314 **
## LATITUDE       5.184e-02  7.706e-02   0.673  0.50242
## LONGITUDE     -4.094e-03  2.277e-02  -0.180  0.85766
## absLatitude   -2.107e-02  1.974e-01  -0.107  0.91516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.68 on 118 degrees of freedom
## Multiple R-squared:  0.2332, Adjusted R-squared:  0.1227
## F-statistic: 2.111 on 17 and 118 DF,  p-value: 0.01047
```

```
coef(mod1)
```

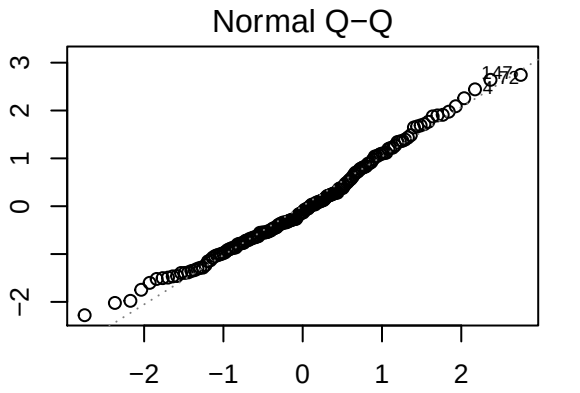
```
##   (Intercept)     MedianAge      AVG_TEMP   PER_CAP_INC Avg_Per_Unemp
##  2.809017e+01 -2.711465e-01 -6.929838e-02  3.205544e-05  4.600928e-01
##      CHGENPCT      JDGENPCT      ISGENPCT      BUGENPCT      ZOGENPCT
##  7.819072e+00 -1.991167e+01 -9.391924e+01  1.049685e+01 -8.338777e+02
##      HIGENPCT      NORELPCT   OtherRelPCT    Prohibited      SunniPCT
##  3.954704e+01  2.218127e+01            NA  1.338144e+01  1.072265e+02
##       ShiaPCT      LATITUDE     LONGITUDE   absLatitude
##  2.034403e+02  5.183928e-02 -4.093555e-03 -2.107073e-02
```

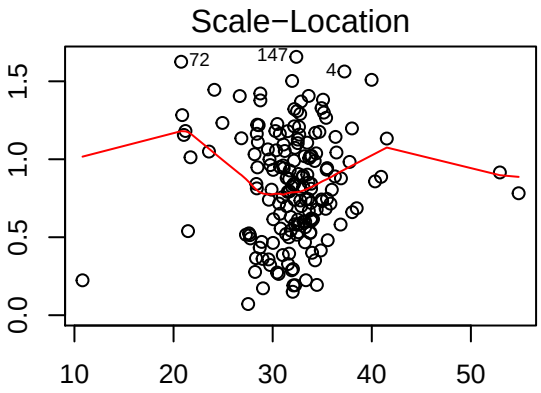
```
confint(mod1)
```

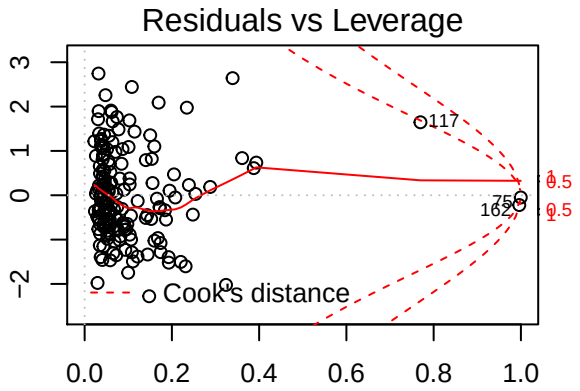
```
##                       2.5 %       97.5 %
## (Intercept)   -2.970719e+00 5.915106e+01
## MedianAge     -7.284830e-01 1.861901e-01
## AVG_TEMP      -7.373945e-01 5.987978e-01
## PER_CAP_INC   -1.171719e-04 1.812828e-04
## Avg_Per_Unemp  4.525766e-02 8.749280e-01
## CHGENPCT      -1.810442e+01 3.374256e+01
```

```
## JDGENPCT       -6.198899e+01 2.216566e+01
## ISGENPCT       -1.999967e+02 1.215825e+01
## BUGENPCT       -1.951793e+01 4.051164e+01
## ZOGENPCT       -5.847096e+03 4.179341e+03
## HIGENPCT        8.569766e+00 7.052432e+01
## NORELPCT       -1.432063e+01 5.868317e+01
## OtherRelPCT               NA           NA
## Prohibited     -1.378740e+01 4.055028e+01
## SunniPCT        7.345058e+00 2.071080e+02
## ShiaPCT         6.985786e+01 3.370228e+02
## LATITUDE       -1.007524e-01 2.044310e-01
## LONGITUDE      -4.919388e-02 4.100677e-02
## absLatitude    -4.118880e-01 3.697466e-01
```

```
par(mfrow = c(3,3))
```








```
plot(mod1);cor(training[,3:ncol(training)])
```

```
## Warning: not plotting observations with leverage one:
##   94
```

```
## Warning: not plotting observations with leverage one:
##   94
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

```
##                    TARGET    MedianAge     AVG_TEMP PER_CAP_INC
## TARGET          1.000000000 -0.050594232  0.008351635  0.06294736
## MedianAge      -0.050594232  1.000000000 -0.685677163  0.55718243
## AVG_TEMP        0.008351635 -0.685677163  1.000000000 -0.37365456
## PER_CAP_INC     0.062947356  0.557182432 -0.373654563  1.00000000
## LATITUDE        0.041285318  0.641071109 -0.690508427  0.42563472
## LONGITUDE       0.039951108  0.005072352 -0.082978348 -0.03557502
## Avg_Per_Unemp   0.157886374  0.091705324 -0.113038170 -0.03986298
## CHGENPCT       -0.166406752  0.048708008 -0.065944068 -0.06158059
## JDGENPCT       -0.143557619  0.010378187  0.017501488  0.08066507
## ISGENPCT        0.232225092 -0.261976786  0.185270661  0.01484391
## BUGENPCT       -0.028992731  0.052398190  0.017346448 -0.03202081
## ZOGENPCT       -0.027807035  0.083789912 -0.117483413  0.16168271
## HIGENPCT        0.210735153 -0.047129466  0.128370596  0.05636770
## NORELPCT        0.006001644  0.492740471 -0.445461689  0.25451223
## OtherRelPCT    -0.035287777 -0.355762902  0.254763494 -0.24173178
## Prohibited      0.119628692 -0.105536703  0.104561463 -0.05250132
## SunniPCT        0.223678167 -0.282997035  0.180761646 -0.02279109
## ShiaPCT         0.268899958 -0.055067006  0.110412535  0.21321461
## absLatitude     0.042174859  0.768176588 -0.888110369  0.55998908
##                   LATITUDE    LONGITUDE Avg_Per_Unemp    CHGENPCT
## TARGET          0.04128532  0.039951108   0.157886374 -0.16640675
## MedianAge       0.64107111  0.005072352   0.091705324  0.04870801
## AVG_TEMP       -0.69050843 -0.082978348  -0.113038170 -0.06594407
## PER_CAP_INC     0.42563472 -0.035575015  -0.039862977 -0.06158059
## LATITUDE        1.00000000 -0.087857321  -0.060599112 -0.07882918
## LONGITUDE      -0.08785732  1.000000000  -0.098077584 -0.36656252
## Avg_Per_Unemp -0.06059911 -0.098077584   1.000000000  0.12018478
## CHGENPCT       -0.07882918 -0.366562522   0.120184779  1.00000000
## JDGENPCT        0.04945197  0.018321336  -0.008760686 -0.18038038
## ISGENPCT        0.02735301  0.145004614   0.061846967 -0.57058485
## BUGENPCT        0.02501858  0.335790654  -0.188857684 -0.51135816
## ZOGENPCT        0.06980610 -0.171798453  -0.016734679  0.02382550
## HIGENPCT       -0.07297453  0.185805939  -0.107748811 -0.37059810
## NORELPCT        0.34882347 -0.011861910  -0.017897685 -0.20227402
## OtherRelPCT    -0.25435619  0.089455008  -0.012983367 -0.37321823
## Prohibited     -0.01443926  0.018441002   0.100940363 -0.16903621
## SunniPCT        0.02083808  0.135060565   0.087614648 -0.53533673
## ShiaPCT         0.04242743  0.100544705  -0.079661008 -0.27570580
## absLatitude     0.73713306  0.055275813   0.177206823  0.01385132
##                  JDGENPCT    ISGENPCT     BUGENPCT     ZOGENPCT
## TARGET         -0.143557619  0.23222509 -0.028992731 -0.027807035
## MedianAge       0.010378187 -0.26197679  0.052398190  0.083789912
## AVG_TEMP        0.017501488  0.18527066  0.017346448 -0.117483413
## PER_CAP_INC     0.080665066  0.01484391 -0.032020810  0.161682712
## LATITUDE        0.049451966  0.02735301  0.025018579  0.069806103
## LONGITUDE       0.018321336  0.14500461  0.335790654 -0.171798453
## Avg_Per_Unemp -0.008760686  0.06184697 -0.188857684 -0.016734679
## CHGENPCT       -0.180380384 -0.57058485 -0.511358158  0.023825502
## JDGENPCT        1.000000000  0.04029133 -0.024306386  0.017379557
## ISGENPCT        0.040291332  1.00000000 -0.048661866 -0.050309492
## BUGENPCT       -0.024306386 -0.04866187  1.000000000 -0.018088432
## ZOGENPCT        0.017379557 -0.05030949 -0.018088432  1.000000000
## HIGENPCT       -0.025079857  0.08139208  0.047136904 -0.008808228
```

```
## NORELPCT      -0.020902875 -0.22871240 -0.038180511  0.073267254
## OtherRelPCT   -0.062329981  0.17689333  0.010232950 -0.052556810
## Prohibited    -0.008428569  0.28922455 -0.021821679 -0.008377960
## SunniPCT       0.040082782  0.98187573 -0.060828890 -0.050209345
## ShiaPCT       -0.023422996  0.49398406 -0.003782054 -0.022122136
## absLatitude    0.036988197 -0.12545963 -0.057102902  0.065920507
##                   HIGENPCT     NORELPCT OtherRelPCT   Prohibited
## TARGET         0.210735153  0.006001644 -0.03528778  0.119628692
## MedianAge     -0.047129466  0.492740471 -0.35576290 -0.105536703
## AVG_TEMP       0.128370596 -0.445461689  0.25476349  0.104561463
## PER_CAP_INC    0.056367702  0.254512230 -0.24173178 -0.052501321
## LATITUDE      -0.072974528  0.348823468 -0.25435619 -0.014439260
## LONGITUDE      0.185805939 -0.011861910  0.08945501  0.018441002
## Avg_Per_Unemp -0.107748811 -0.017897685 -0.01298337  0.100940363
## CHGENPCT      -0.370598100 -0.202274017 -0.37321823 -0.169036208
## JDGENPCT      -0.025079857 -0.020902875 -0.06232998 -0.008428569
## ISGENPCT       0.081392075 -0.228712399  0.17689333  0.289224546
## BUGENPCT       0.047136904 -0.038180511  0.01023295 -0.021821679
## ZOGENPCT      -0.008808228  0.073267254 -0.05255681 -0.008377960
## HIGENPCT       1.000000000 -0.137605210 -0.08853964 -0.022487304
## NORELPCT      -0.137605210  1.000000000 -0.00945223 -0.053136537
## OtherRelPCT   -0.088539636 -0.009452230  1.00000000  0.130042383
## Prohibited    -0.022487304 -0.053136537  0.13004238  1.000000000
## SunniPCT       0.024282659 -0.230347880  0.19286300  0.312599702
## ShiaPCT        0.119398580 -0.110165459 -0.04019814 -0.021086163
## absLatitude   -0.088635136  0.482748891 -0.26900688 -0.055565200
##                  SunniPCT      ShiaPCT absLatitude
## TARGET         0.22367817  0.268899958  0.04217486
## MedianAge     -0.28299703 -0.055067006  0.76817659
## AVG_TEMP       0.18076165  0.110412535 -0.88811037
## PER_CAP_INC   -0.02279109  0.213214608  0.55998908
## LATITUDE       0.02083808  0.042427428  0.73713306
## LONGITUDE      0.13506057  0.100544705  0.05527581
## Avg_Per_Unemp  0.08761465 -0.079661008  0.17720682
## CHGENPCT      -0.53533673 -0.275705803  0.01385132
## JDGENPCT       0.04008278 -0.023422996  0.03698820
## ISGENPCT       0.98187573  0.493984058 -0.12545963
## BUGENPCT      -0.06082889 -0.003782054 -0.05710290
## ZOGENPCT      -0.05020934 -0.022122136  0.06592051
## HIGENPCT       0.02428266  0.119398580 -0.08863514
## NORELPCT      -0.23034788 -0.110165459  0.48274889
## OtherRelPCT    0.19286300 -0.040198138 -0.26900688
## Prohibited     0.31259970 -0.021086163 -0.05556520
## SunniPCT       1.00000000  0.373263676 -0.12489760
## ShiaPCT        0.37326368  1.000000000 -0.02016532
## absLatitude   -0.12489760 -0.020165318  1.00000000
```

```
#Box Cox Transformation
## considering how weak the data is, and the box cox is near 1, I don't think a variable transformation
Ac1=boxcox(glm(TARGET~MedianAge+AVG_TEMP+MedianAge+AVG_TEMP
               +LONGITUDE+Avg_Per_Unemp+CHGENPCT+JDGENPCT+ISGENPCT
               +BUGENPCT+ZOGENPCT+HIGENPCT+PER_CAP_INC+LATITUDE
               +NORELPCT+OtherRelPCT+Prohibited,data=data2),
           plotit = TRUE,family="poisson",MLEQ=TRUE,
```

```
          interp= TRUE, eps = 1/50, xlab = expression(lambda),
          ylab = "log-Likelihood")
 title(main="BoxCox Transformation")
```

```
#Scale and Center variables for Lasso.  TARGET is sqrt
ldata<-data.frame(
TARGET=scale(sqrt(training$TARGET)-mean(sqrt(training$TARGET)))/sd(sqrt(training$TARGET)),
MedianAge=scale(training$MedianAge-mean(training$MedianAge))/sd(training$MedianAge),
AVG_TEMP=scale(training$AVG_TEMP-mean(training$AVG_TEMP))/sd(training$AVG_TEMP),
PER_CAP_INC=scale(training$PER_CAP_INC-mean(training$PER_CAP_INC))/sd(training$PER_CAP_INC),
Avg_Per_Unemp=scale(training$Avg_Per_Unemp-mean(training$Avg_Per_Unemp))/sd(training$Avg_Per_Unemp),
CHGENPCT=scale(training$CHGENPCT-mean(training$CHGENPCT))/sd(training$CHGENPCT),
JDGENPCT=scale(training$JDGENPCT-mean(training$JDGENPCT))/sd(training$JDGENPCT),
ISGENPCT=scale(training$ISGENPCT-mean(training$ISGENPCT))/sd(training$ISGENPCT),
BUGENPCT=scale(training$BUGENPCT-mean(training$BUGENPCT))/sd(training$BUGENPCT),
ZOGENPCT=scale(training$ZOGENPCT-mean(training$ZOGENPCT))/sd(training$ZOGENPCT),
HIGENPCT=scale(training$HIGENPCT-mean(training$HIGENPCT))/sd(training$HIGENPCT),
NORELPCT=scale(training$NORELPCT-mean(training$NORELPCT))/sd(training$NORELPCT),
OtherRelPCT=scale(training$OtherRelPCT-mean(training$OtherRelPCT))/sd(training$OtherRelPCT),
Prohibited=scale(training$Prohibited-mean(training$Prohibited))/sd(training$Prohibited),
SunniPCT=scale(training$SunniPCT-mean(training$SunniPCT))/sd(training$SunniPCT),
ShiaPCT=scale(training$ShiaPCT-mean(training$ShiaPCT))/sd(training$ShiaPCT),
LATITUDE=scale(training$LATITUDE-mean(training$LATITUDE))/sd(training$LATITUDE),
absLatitude=scale(training$absLatitude-mean(training$absLatitude))/sd(training$absLatitude),
LONGITUDE=scale(training$LONGITUDE-mean(training$LONGITUDE))/sd(training$LONGITUDE)
)


#Leaps

regsubsets.out <-regsubsets(TARGET~MedianAge+AVG_TEMP+PER_CAP_INC
                            +Avg_Per_Unemp+CHGENPCT+JDGENPCT
                            +ISGENPCT+BUGENPCT+ZOGENPCT+HIGENPCT
                            +NORELPCT+OtherRelPCT+Prohibited
                            +SunniPCT+ShiaPCT+LATITUDE+LONGITUDE+absLatitude,data=ldata,
                            nbest = 1,       # 1 best model for each number of predictors
                            nvmax = NULL,    # NULL for no limit on number of variables
                            force.in = NULL, force.out = NULL,
                            method = "exhaustive")
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
regsubsets.out
```

```
## Subset selection object
```

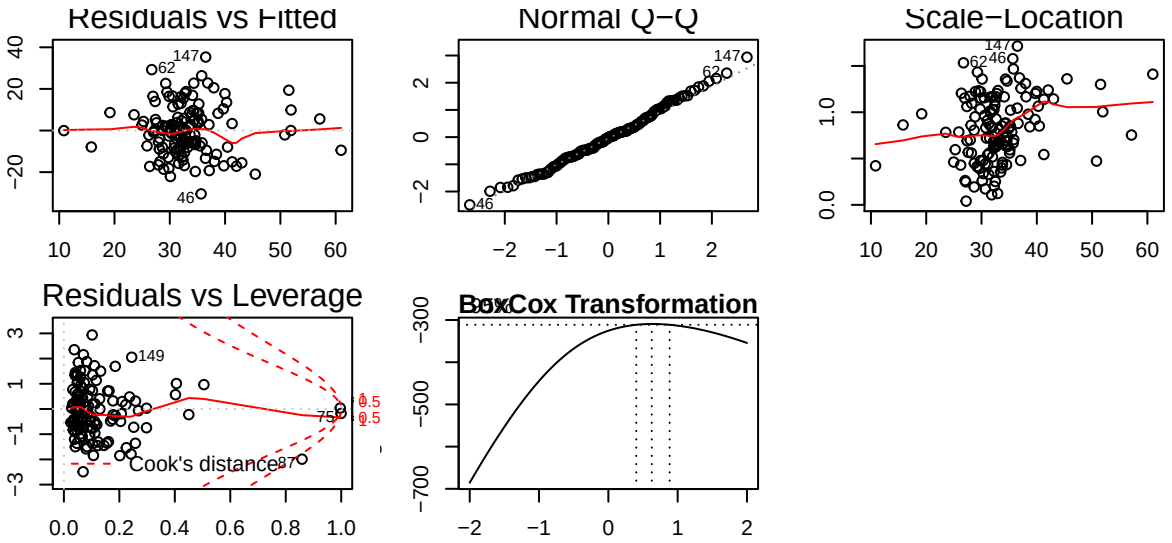
```
## Call: regsubsets.formula(TARGET ~ MedianAge + AVG_TEMP + PER_CAP_INC +
##     Avg_Per_Unemp + CHGENPCT + JDGENPCT + ISGENPCT + BUGENPCT +
##     ZOGENPCT + HIGENPCT + NORELPCT + OtherRelPCT + Prohibited +
##     SunniPCT + ShiaPCT + LATITUDE + LONGITUDE + absLatitude,
##     data = ldata, nbest = 1, nvmax = NULL, force.in = NULL, force.out = NULL,
##     method = "exhaustive")
## 18 Variables  (and intercept)
##               Forced in Forced out
## MedianAge         FALSE      FALSE
## AVG_TEMP          FALSE      FALSE
## PER_CAP_INC       FALSE      FALSE
## Avg_Per_Unemp     FALSE      FALSE
## CHGENPCT          FALSE      FALSE
## JDGENPCT          FALSE      FALSE
## ISGENPCT          FALSE      FALSE
## BUGENPCT          FALSE      FALSE
## ZOGENPCT          FALSE      FALSE
## HIGENPCT          FALSE      FALSE
## NORELPCT          FALSE      FALSE
## Prohibited        FALSE      FALSE
## SunniPCT          FALSE      FALSE
## ShiaPCT           FALSE      FALSE
## LATITUDE          FALSE      FALSE
## LONGITUDE         FALSE      FALSE
## absLatitude       FALSE      FALSE
## OtherRelPCT       FALSE      FALSE
## 1 subsets of each size up to 17
## Selection Algorithm: exhaustive
```

```
summary.out<-summary(regsubsets.out)
as.data.frame(summary.out$outmat)
```

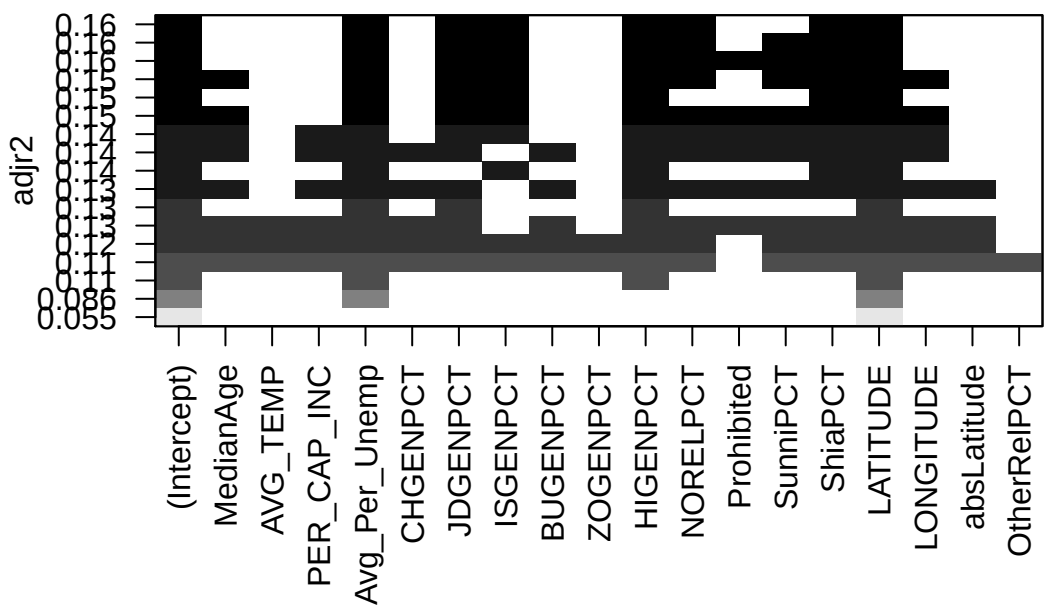
```
##           MedianAge AVG_TEMP PER_CAP_INC Avg_Per_Unemp CHGENPCT JDGENPCT
## 1  ( 1 )                                                                 
## 2  ( 1 )                                             *                   
## 3  ( 1 )                                             *                   
## 4  ( 1 )                                             *                 *
## 5  ( 1 )                                             *                   
## 6  ( 1 )                                             *                 *
## 7  ( 1 )                                             *                 *
## 8  ( 1 )                                             *                 *
## 9  ( 1 )                                             *                 *
## 10  ( 1 )         *                                  *                 *
## 11  ( 1 )         *                                  *                 *
## 12  ( 1 )         *                   *              *                 *
## 13  ( 1 )         *                   *              *        *        *
## 14  ( 1 )         *                   *              *        *        *
## 15  ( 1 )         *         *         *              *        *        *
## 16  ( 1 )         *         *         *              *        *        *
## 17  ( 1 )         *         *         *              *        *        *
##           ISGENPCT BUGENPCT ZOGENPCT HIGENPCT NORELPCT OtherRelPCT
## 1  ( 1 )                                                          
## 2  ( 1 )                                                          
## 3  ( 1 )                                  *                       
```

```
## 4  ( 1 )                                 *
## 5  ( 1 )          *                      *
## 6  ( 1 )          *                      *
## 7  ( 1 )          *                      *         *
## 8  ( 1 )          *                      *         *
## 9  ( 1 )          *                      *         *           *
## 10  ( 1 )         *                      *         *
## 11  ( 1 )         *                      *         *           *
## 12  ( 1 )         *                      *         *           *
## 13  ( 1 )                  *             *         *           *
## 14  ( 1 )                  *             *         *           *
## 15  ( 1 )                  *             *         *           *
## 16  ( 1 )         *        *        *    *         *
## 17  ( 1 )         *        *        *    *         *
##           Prohibited SunniPCT ShiaPCT LATITUDE LONGITUDE absLatitude
## 1  ( 1 )                             *
## 2  ( 1 )                             *
## 3  ( 1 )                             *
## 4  ( 1 )                             *
## 5  ( 1 )                    *        *
## 6  ( 1 )                    *        *
## 7  ( 1 )                    *        *
## 8  ( 1 )           *        *        *
## 9  ( 1 )           *        *        *
## 10  ( 1 )          *        *        *         *
## 11  ( 1 )          *        *        *         *
## 12  ( 1 )          *        *        *         *
## 13  ( 1 )          *        *        *         *
## 14  ( 1 )          *        *        *         *         *
## 15  ( 1 )          *        *        *         *         *
## 16  ( 1 )          *        *        *         *         *
## 17  ( 1 )          *        *        *         *         *           *
```

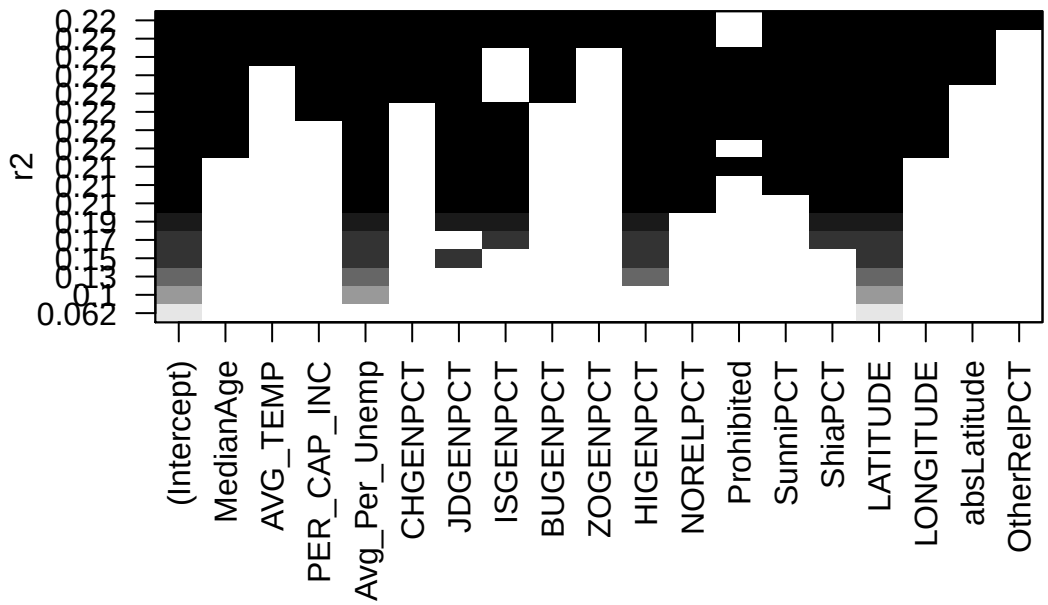
```
par(mfrow = c(1,1))
```



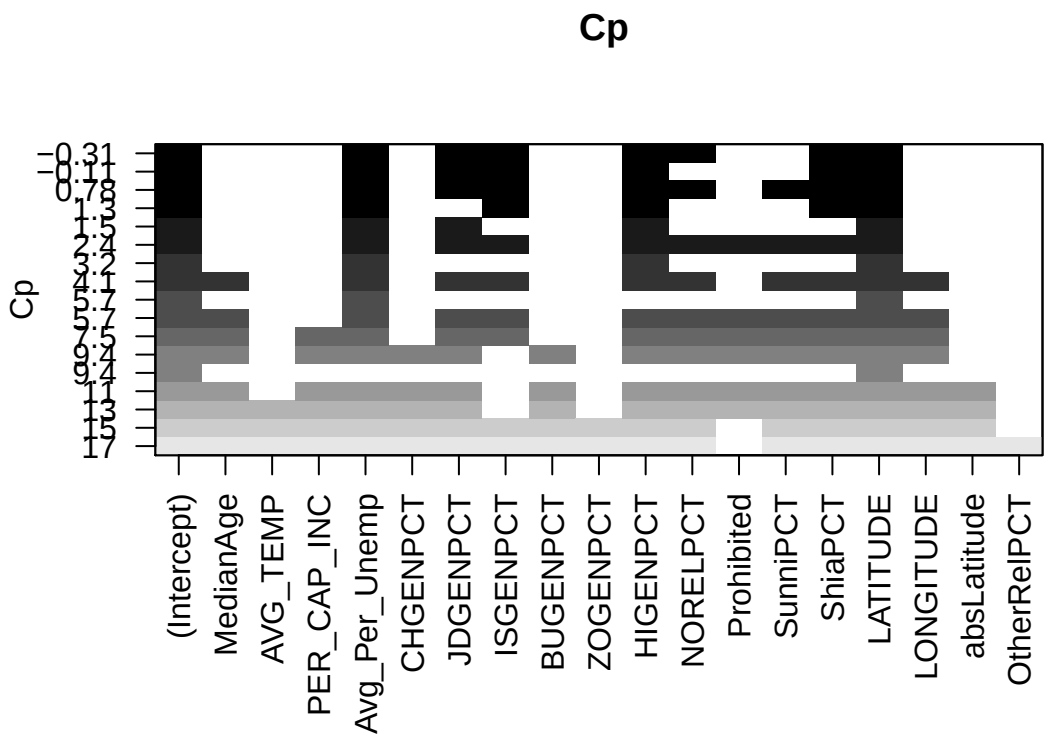
```
plot(regsubsets.out, scale = "adjr2", main = "Adjusted R^2")
```



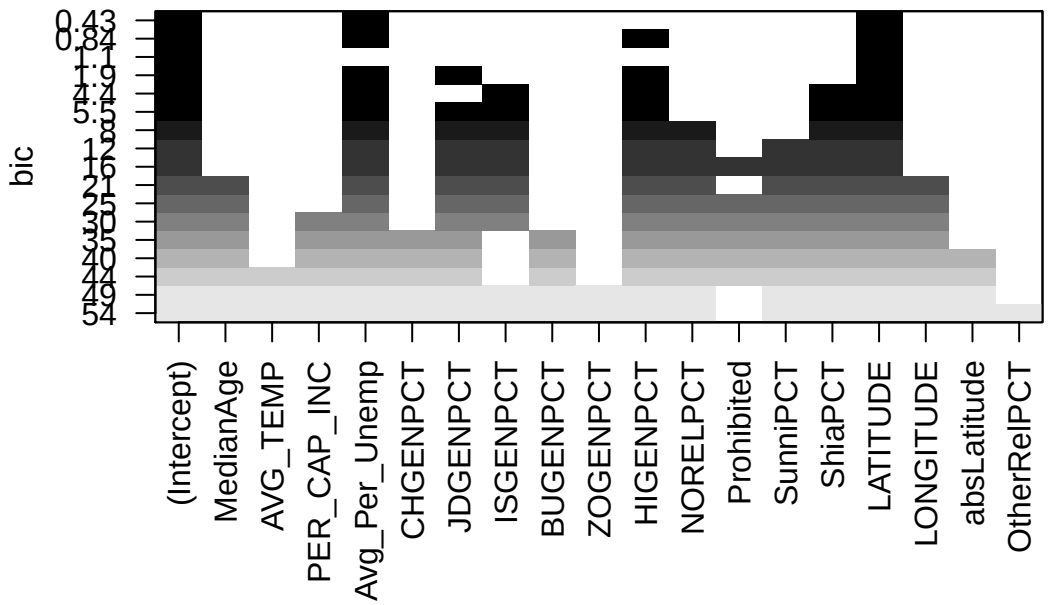

```
plot(regsubsets.out,scale="r2", main = " R2")
```



```
plot(regsubsets.out,scale="Cp", main = " Cp")
```




```
plot(regsubsets.out, scale="bic", main = "BIC")  #we now have 3 predictors, unemployment, HIGENPCT, and
```



```r
coef(regsubsets.out,10)
```

```
##   (Intercept)     MedianAge Avg_Per_Unemp      JDGENPCT      ISGENPCT
## -5.204170e-18 -5.380221e-01  1.042685e+00 -7.837608e-03 -2.141090e-01
##      HIGENPCT      NORELPCT      SunniPCT       ShiaPCT      LATITUDE
##  2.500823e-02  1.356194e-02  2.031164e-01  9.342426e-03  1.780078e+00
##     LONGITUDE
## -1.618399e+00
```

```r
# 10-fold Cross validation with sqrt Validation
set.seed(11)
folds = sample(rep(1:10, length = nrow(data2)))
table(folds)
```

```
## folds
##  1  2  3  4  5  6  7  8  9 10
## 17 17 17 17 17 17 17 17 17 16
```

```r
## Part 3: Apply Best Subset Selection using 10-fold Cross-Validation to select the number
# of predictors and then fit the least squares regression model using the "best" subset.
k <- 10
set.seed(1306)
folds <- sample(1:k, nrow(ldata), replace = TRUE)
cv.errors <- matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))

# Let's write our own predict method
predict.regsubsets <- function(object, newdata, id,...){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars]%*%coefi
}



for (j in 1:k) {
  best.fit <- regsubsets(TARGET ~ ., data = ldata[folds != j, ], nvmax = 10)  ##finds best predictors
  for (i in 1:10) {
    pred <- predict(best.fit, ldata[folds == j, ], id = i)
    cv.errors[j, i] = mean((ldata$TARGET[folds == j] - pred)^2)
  }
}
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 2 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:
```

```r
# This gives us a 10x10 matrix, of which the (i, j)th element corresponds
# to the test MSE for the ith cross-validation fold for the best j-variable model
cv.errors
```

```
##               1         2         3         4         5         6
##  [1,] 0.6997511 0.7178414 0.7193487 0.6551245 0.5570997 0.5562324
##  [2,] 0.6761811 0.6868564 0.6970571 0.6976168 0.6728304 0.6715195
##  [3,] 0.4370095 0.3885493 0.3760518 0.3770928 0.3472307 0.3448071
##  [4,] 0.5504976 0.4911677 0.4851380 0.5284884 0.4909189 0.4982602
##  [5,] 0.4500313 0.4348883 0.5452222 0.5454513 0.5090142 0.5061660
##  [6,] 0.6999574 0.5759728 0.5200616 0.5260679 0.4733135 0.4759919
##  [7,] 0.5149877 0.5929048 0.6817488 0.6685841 0.5996249 0.6244759
##  [8,] 0.7394193 0.7781984 1.2200572 1.4819306 0.6770264 0.6953890
##  [9,] 0.6333326 0.6086818 0.5993763 0.5947010 0.5224415 0.5399328
## [10,] 1.6988969 1.7274295 1.6883767 1.3042750 1.2975110 1.1963220
##               7         8         9        10
##  [1,] 0.5517528 0.5556116 0.5558818 0.6004953
##  [2,] 0.6653877 0.6667475 0.6630084 0.6624858
##  [3,] 0.3492625 0.3574666 0.3663747 0.3662257
##  [4,] 0.7380255 0.8169855 0.8621842 0.8538335
##  [5,] 0.4819794 0.4089062 0.4042577 0.3941133
##  [6,] 0.4811233 0.5158486 0.5239194 0.5334367
##  [7,] 0.6242078 0.5729071 0.5736706 0.5805378
##  [8,] 1.1988864 1.1150852 1.1261856 1.2498830
##  [9,] 0.5334686 0.6009489 0.6128000 0.6196666
## [10,] 1.4693085 1.4704406 1.4785789 1.5502407
```

```
mean.cv.errors <- apply(cv.errors, 2, mean)
mean.cv.errors
```

```
##         1         2         3         4         5         6         7
## 0.7100064 0.7002490 0.7532438 0.7379332 0.6147011 0.6109097 0.7093402
##         8         9        10
## 0.7080948 0.7166861 0.7410918
```

```
which.min(mean.cv.errors)
```

```
## 6
## 6
```

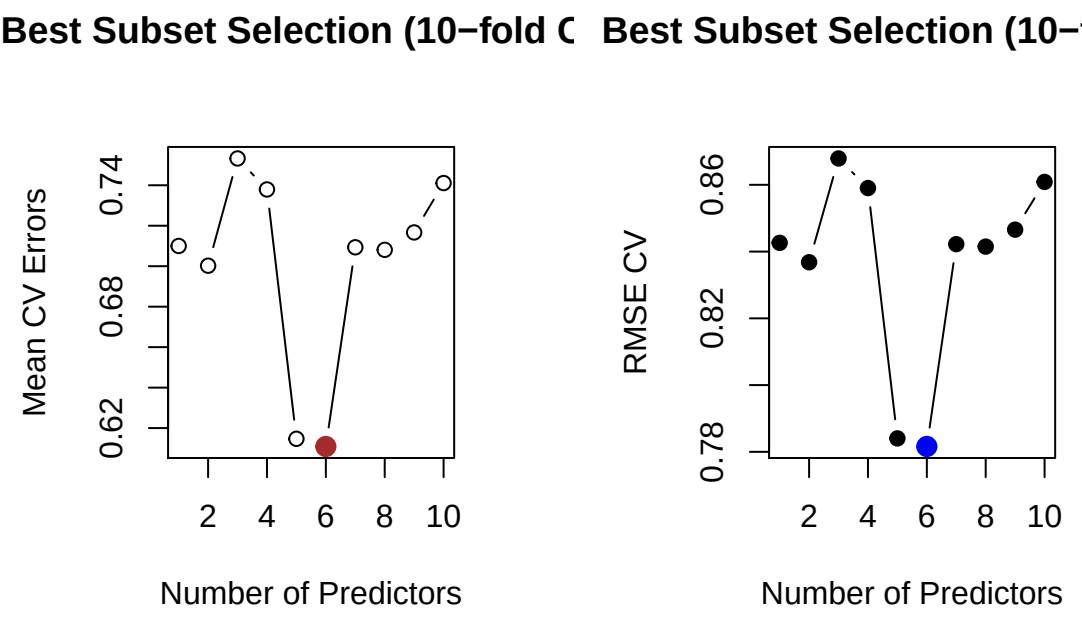
```
mean.cv.errors[6]
```

```
##         6
## 0.6109097
```

```
par(mfrow = c(1,2))
plot(mean.cv.errors, type = 'b', xlab = "Number of Predictors", ylab = "Mean CV Errors",
     main = "Best Subset Selection (10-fold CV)")
points(6, mean.cv.errors[6], col = "brown", cex = 2, pch = 20)

rmse.cv = sqrt(apply(cv.errors, 2, mean))
rmse.cv[6]
```

```
##         6
## 0.7816071
```

```
plot(rmse.cv, pch = 19, type = "b", xlab = "Number of Predictors", ylab = "RMSE CV",
     main = "Best Subset Selection (10-fold CV)")
points(6, rmse.cv[6], col = "blue", cex = 2, pch = 20)
```




```
# The cross-validation selects a 5 or 6-variable model, so we perform best subset
# selection on the training data set to get the best 5-variable model, since it is slightly simpler
reg.best <- regsubsets(TARGET ~ ., data = ldata, nvmax = 10)
```

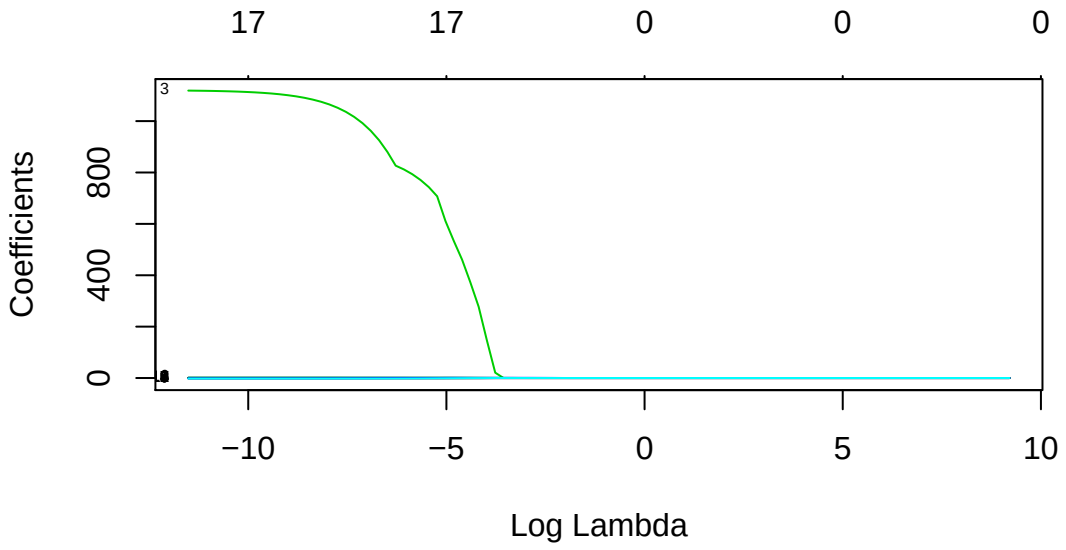
```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

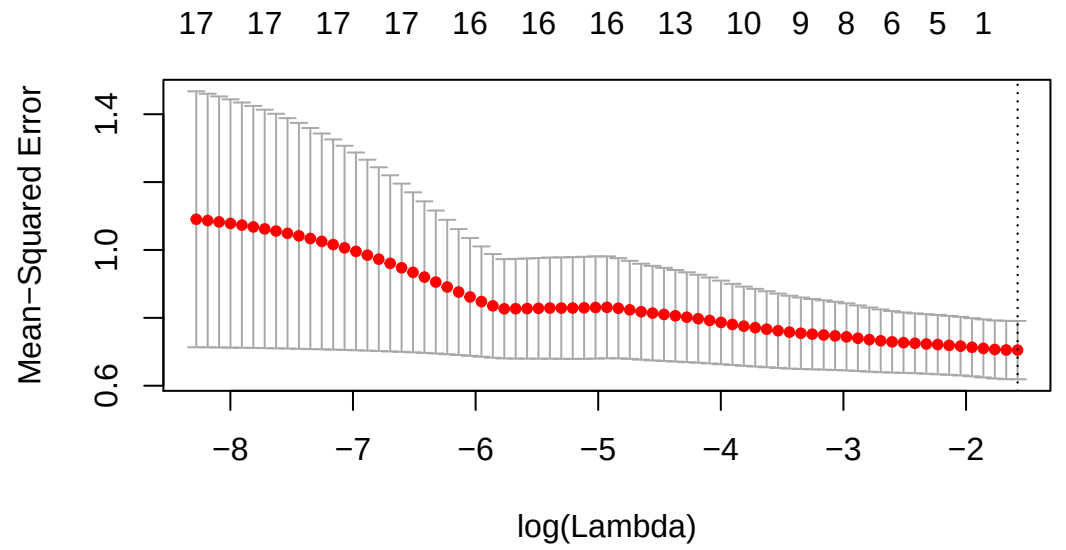
```
coef(reg.best, 5)  #I guess this give the best six
```

```
##   (Intercept) Avg_Per_Unemp      ISGENPCT      HIGENPCT       ShiaPCT
## -5.204170e-18  1.058967e+00  9.162962e-03  1.725234e-02  4.784845e-03
##      LATITUDE
##  1.478791e+00
```

```
## Part 5: Lasso model using 10-fold cross-validation to select that largest
# value of lambda s.t. the CV error is within 1 s.e. of the minimum

x.train <- as.matrix(dplyr::select(ldata, -TARGET))
y.train <- ldata$TARGET


par(mfrow = c(1,1))
grid <- 10^seq(4, -5, length = 100)
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = grid, thresh = 1e-12)
plot(lasso.mod, xvar = "lambda", label = TRUE)
```




```
set.seed(1306)
cv.out <- cv.glmnet(x.train, y.train, alpha = 1)
plot(cv.out)
```




```
bestlam <- cv.out$lambda.min
bestlam                                    # Lambda = 0.2026 (leads to smallest CV error)
```

```
## [1] 0.2060459
```

```
log(bestlam)
```

```
## [1] -1.579656
```

```
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = bestlam)
lasso.coef <- predict(lasso.mod, type = "coefficients", s = bestlam)[1:19,]
lasso.coef[lasso.coef != 0]  ## gives only intercept
```

```
##   (Intercept)
## -1.561251e-17
```

```
largelam <- cv.out$lambda.1se
largelam                                    # Lambda = 4.791278 (largest lambda w/in 1 SE)
```

```
## [1] 0.2060459
```

```
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = largelam)



# Here are the estimated coefficients
lasso.coef <- predict(lasso.mod, type = "coefficients", s = largelam)[1:19,]
lasso.coef[lasso.coef != 0]
```

```
##   (Intercept)
## -1.561251e-17
```

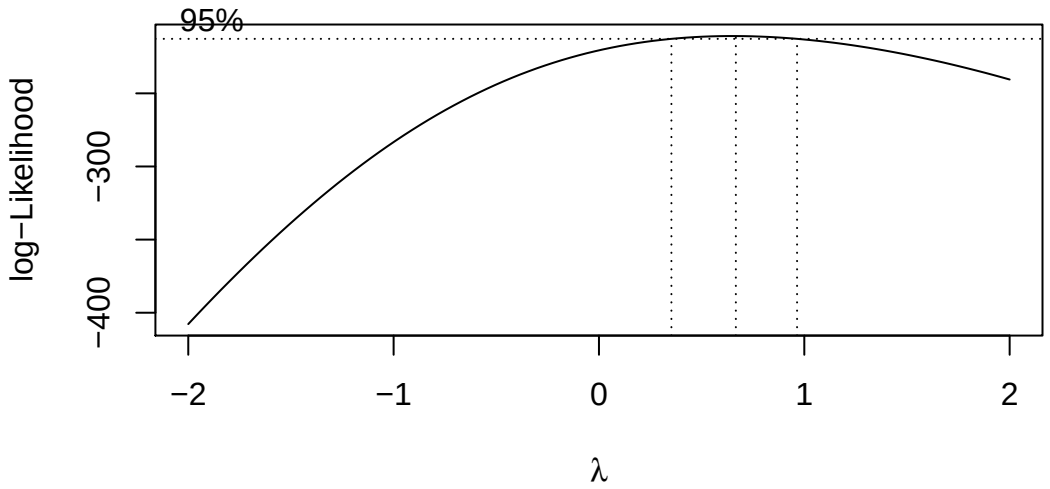
```
#############################################################################
## we use the 5 predictor model taken from best subset 10-fold cross validation
save(training,file="training_NotScaled.Rda")
save(ldata,file="training_Scaled.Rda")
#Then load it with:
#load("data.Rda")
mod1<-lm(sqrt(TARGET)~Avg_Per_Unemp+ISGENPCT+HIGENPCT+ShiaPCT+LATITUDE,data=training)
summary(mod1)  ## R^2 is 0.137.  Only 3 predictors appear significant with all data.
```

```
##
## Call:
## lm(formula = sqrt(TARGET) ~ Avg_Per_Unemp + ISGENPCT + HIGENPCT +
##     ShiaPCT + LATITUDE, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3353 -0.7212 -0.0079  0.8066  2.7057
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.083332   0.189367  26.844   <2e-16 ***
## Avg_Per_Unemp 0.042016   0.016517   2.544   0.0121 *
## ISGENPCT      0.406765   0.629151   0.647   0.5191
## HIGENPCT      1.824796   0.851834   2.142   0.0340 *
## ShiaPCT       9.324349   4.185066   2.228   0.0276 *
## LATITUDE      0.003457   0.003964   0.872   0.3848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.139 on 130 degrees of freedom
## Multiple R-squared:  0.1372, Adjusted R-squared:  0.104
## F-statistic: 4.133 on 5 and 130 DF,  p-value: 0.001622
```

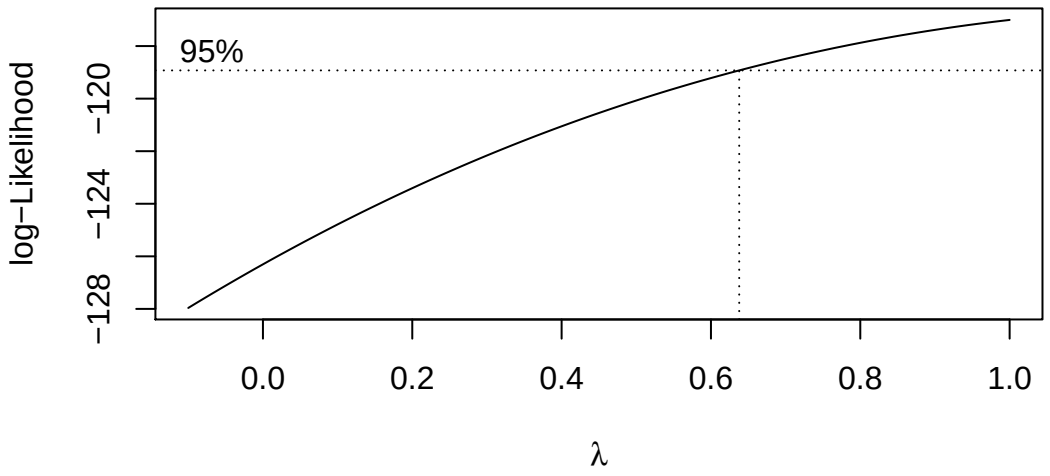
```r
#The model without sqrt tranformation looks just as good.
mod2<-lm((TARGET)~Avg_Per_Unemp+ISGENPCT+HIGENPCT+ShiaPCT+LATITUDE,data=training)
summary(mod2) #R2 = .15, adj R2 = .12
```

```
##
## Call:
## lm(formula = (TARGET) ~ Avg_Per_Unemp + ISGENPCT + HIGENPCT +
##     ShiaPCT + LATITUDE, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.561  -8.864  -1.178   8.691  35.880
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    27.05561    2.10919  12.828   <2e-16 ***
## Avg_Per_Unemp   0.43407    0.18397   2.359   0.0198 *
## ISGENPCT        7.42811    7.00753   1.060   0.2911
## HIGENPCT       23.46109    9.48779   2.473   0.0147 *
## ShiaPCT       103.72644   46.61358   2.225   0.0278 *
## LATITUDE        0.03060    0.04415   0.693   0.4895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.68 on 130 degrees of freedom
## Multiple R-squared:  0.1548, Adjusted R-squared:  0.1223
## F-statistic: 4.761 on 5 and 130 DF,  p-value: 0.0004999
```

```r
#box cox predicts sqrt, but no transformation, model2 looks better
library(MASS)
boxcox(mod2, plotit=T)
```




```r
boxcox(mod1, plotit=T, lambda=seq(-0.1,1,by=0.1))
```



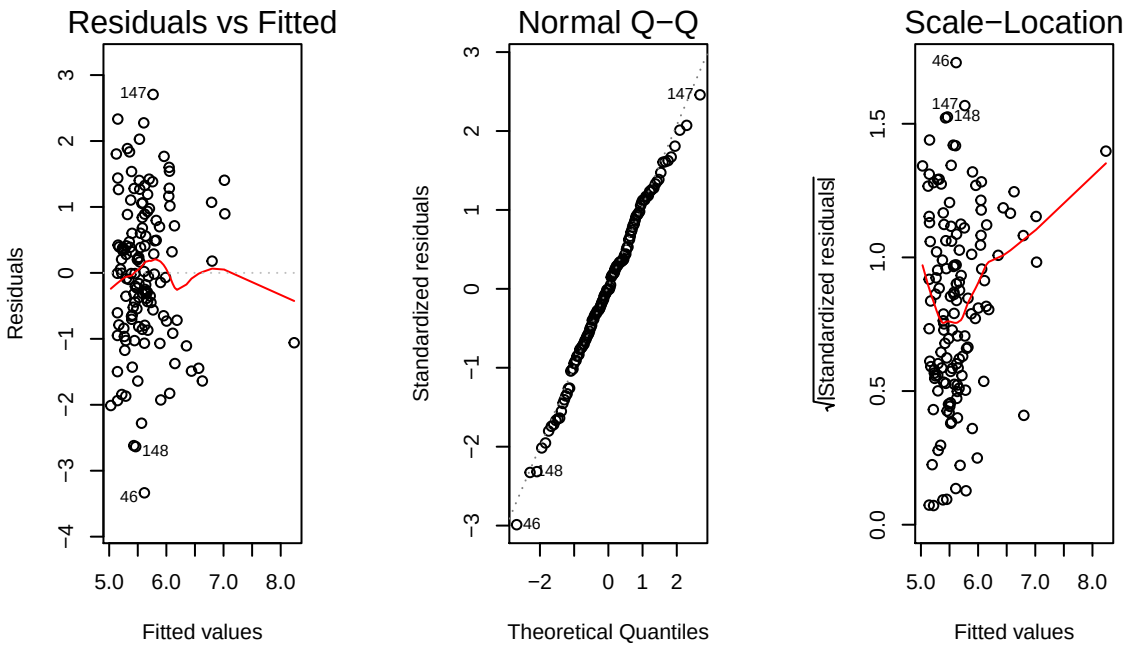

```
coef(mod2)
```

```
##   (Intercept) Avg_Per_Unemp      ISGENPCT      HIGENPCT       ShiaPCT
##   27.05561015    0.43406576    7.42811226   23.46109088  103.72643927
##      LATITUDE
##    0.03060283
```

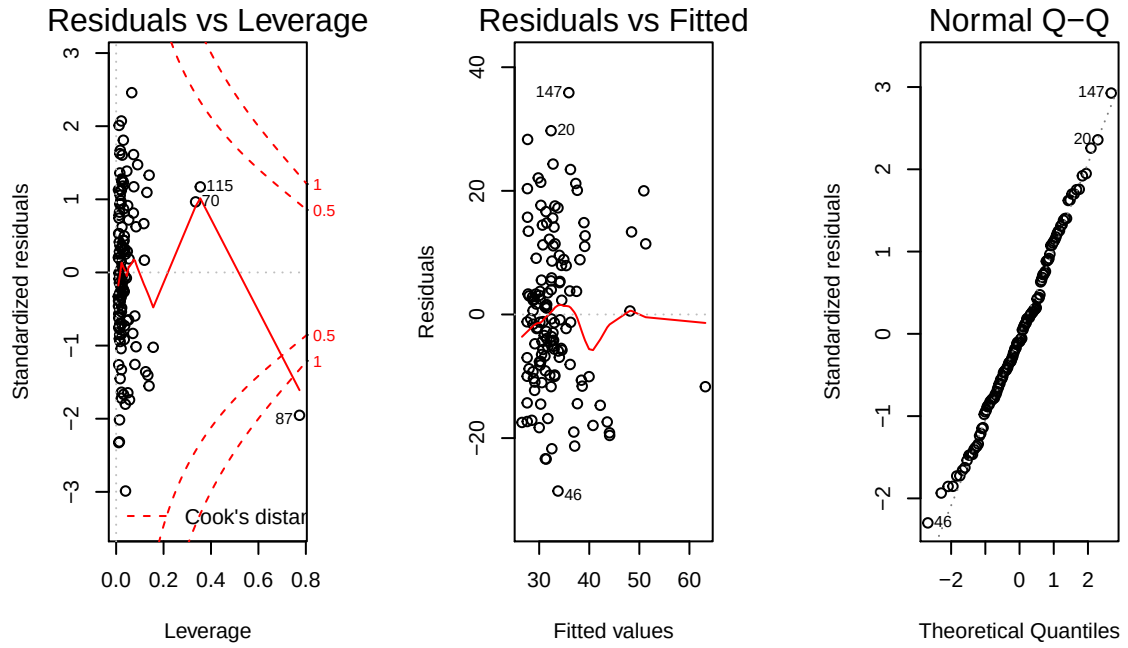
```
confint(mod2)
```

```
##                     2.5 %      97.5 %
## (Intercept)   22.88283405  31.2283863
## Avg_Per_Unemp  0.07010782   0.7980237
## ISGENPCT      -6.43544732  21.2916718
## HIGENPCT       4.69063521  42.2315466
## ShiaPCT       11.50704094 195.9458376
## LATITUDE      -0.05674652   0.1179522
```

```
par(mfrow = c(1,3))
plot(mod1)
```




```
plot(mod2)
```




```
vif(mod2)  #no colinearity
```

```
## Avg_Per_Unemp      ISGENPCT      HIGENPCT       ShiaPCT      LATITUDE
##      1.035951      1.343420      1.033602      1.354141      1.012735
```

```
#############################################################
## we have a working model (mod2)  This can be used
## to make predictions with. Albiet, the model is not very good
```

```
file.e <- read.csv("~/Downloads/RE__project/cities_eval.csv")
#file.e <- read.csv("cities_eval.csv", stringsAsFactors=TRUE)
head(file.e)
```

```
##         TARGET AVG_TEMP PER_CAP_INC LATITUDE LONGITUDE Avg_Per_Unemp
## 1        Miami     20.0       23304       25       -80           5.2
## 2      Detriot     10.0       14100       42       -83          24.0
## 3   Omaha City     10.6       19613       41        96           5.0
## 4 SanFrancisco     14.0       72364       38      -122           8.9
## 5  Albuquerque     13.9       20884       35      -106           7.3
## 6       Boston     11.0       39815       42       -71           7.6
##   CHGENPCT JDGENPCT ISGENPCT BUGENPCT ZOGENPCT HIGENPCT NORELPCT
## 1     0.68     0.09     0.00     0.00        0     0.00     0.24
## 2     0.67     0.02     0.03     0.01        0     0.00     0.57
## 3     0.51     0.00     0.00     0.00        0     0.00     0.49
## 4     0.48     0.03     0.01     0.02        0     0.05     0.35
## 5     0.44     0.00     0.00     0.00        0     0.00     0.56
## 6     0.57     0.04     0.01     0.01        0     0.00     0.33
##   OtherRelPCT MedianAge ShiaPCT
## 1        0.00      37.2   0.000
## 2        0.04      39.1   0.030
## 3        0.00      33.5   0.000
## 4        0.03      38.5   0.005
## 5        0.00      35.3   0.000
## 6        0.02      38.5   0.005
```

```
##Calculate CI intervals for 5-95%
values = 0
high = 0
low = 0
for (x in 1:nrow(file.e)){
  values[x] = (predict(mod2,new=file.e[x,-1],interval="prediction")[1] )
  high[x] = (predict(mod2,new=file.e[x,-1],interval="prediction")[3] )
  low[x] = (predict(mod2,new=file.e[x,-1],interval="prediction")[2] )
}

values
```

```
##  [1] 30.07782 42.09314 30.48065 33.84767 31.29539 32.23274 34.39789
##  [8] 31.71961 31.02369 66.60228 31.23637
```

```
high
```

```
##  [1] 55.31779 68.30498 55.78614 59.13495 56.55462 57.53157 59.70270
##  [8] 57.08642 56.44064 98.33549 56.54388
```

```
low
```

```
##  [1]  4.837855 15.881304  5.175171  8.560395  6.036160  6.933911  9.093080
##  [8]  6.352799  5.606745 34.869072  5.928854
```

```
##NYC is 34.4 with a SE of 12.9
## Standard errors (standard deviations) for the predicted values
SE = c(12.87753571,  13.37338776,   12.91096429 ,12.90167347,   12.88736224,    12.90756633 ,12.9106173
```

```
## ration city/NY


getCIRatio <- function(SDY, SDX, Y, X, T1, COV){
  VY = SDY*SDY
  VX = SDX*SDX
  X2 = X*X
  Y2 = Y*Y
  V = Y/X
  T2 = T1*T1
  Q = 1-T2*VX/X2
  C = V/Q
  SE.R =sqrt(VY-2*Y/X*COV+Y2/X2*VX-T2*VX/X2*(VY-COV^2/VX))/X/Q
  CI1 = C-T1*SE.R
  CI2 = C+T1*SE.R
  l1 = c(CI1, CI2)
  return (l1)
}


## calculate CI for ratios, assuming COV is 0
high.r = 0
low.r = 0
ratios.NY = 0
for (x in 1:length(values)){
  COV = 0
  l1 = getCIRatio(SE[x], 12.9, values[x], 34.4, 1.96, COV)
  high.r[x] = l1[2]
  low.r[x] = l1[1]
  ratios.NY[x] = values[x]/34.4
}

high.r
```

```
##  [1] 3.669356 4.917307 3.711371 4.050402 3.791903 3.887284 4.106720
##  [8] 3.837309 3.768682 7.591885 3.787210
```

```
low.r
```

```
##  [1] 0.1340496 0.4054609 0.1429730 0.2297087 0.1654662 0.1886149 0.2429672
##  [8] 0.1737036 0.1543302 0.8301160 0.1626952
```

```
ratios.NY
```

```
##  [1] 0.8743553 1.2236379 0.8860655 0.9839439 0.9097497 0.9369983 0.9999387
##  [8] 0.9220816 0.9018515 1.9361128 0.9080339
```

```
## calculate CI for ratios, assuming COV is 1
high.r = 0
low.r = 0
ratios.NY = 0
for (x in 1:length(values)){
  COV = 1
  l1 = getCIRatio(SE[x], 12.9, values[x], 34.4, 1.96, COV)
  high.r[x] = l1[2]
  low.r[x] = l1[1]
  ratios.NY[x] = values[x]/34.4
}

high.r
```

```
##  [1] 3.661757 4.908973 3.703742 4.042489 3.784195 3.879501 4.098768
##  [8] 3.829576 3.761016 7.583087 3.779513
```

```
low.r
```

```
##  [1] 0.1416480 0.4137950 0.1506018 0.2376218 0.1731739 0.1963987 0.2509195
##  [8] 0.1814366 0.1619964 0.8389145 0.1703924
```

```
ratios.NY
```

```
##  [1] 0.8743553 1.2236379 0.8860655 0.9839439 0.9097497 0.9369983 0.9999387
##  [8] 0.9220816 0.9018515 1.9361128 0.9080339
```

```
## calculate CI for ratios, assuming COV is -1
high.r = 0
low.r = 0
ratios.NY = 0
for (x in 1:length(values)){
  COV = -1
  l1 = getCIRatio(SE[x], 12.9, values[x], 34.4, 1.96, COV)
  high.r[x] = l1[2]
  low.r[x] = l1[1]
  ratios.NY[x] = values[x]/34.4
}

high.r
```

```
##  [1] 3.676950 4.925633 3.718995 4.058308 3.799605 3.895062 4.114665
##  [8] 3.845036 3.776344 7.600676 3.794902
```

```
low.r
```

```
##  [1] 0.1264556 0.3971354 0.1353487 0.2218023 0.1577638 0.1808369 0.2350219
##  [8] 0.1659760 0.1466689 0.8213257 0.1550032
```

```
ratios.NY
```

```
##  [1] 0.8743553 1.2236379 0.8860655 0.9839439 0.9097497 0.9369983 0.9999387
##  [8] 0.9220816 0.9018515 1.9361128 0.9080339
```

```
## calculate CI for ratios, assuming COV is 0, at 70% confidence
## at 70% confidence ratio, we can state the mumbai drinks more than NYC
##  see http://www.mapsofindia.com/my-india/india/alcohol-consumption-in-india for some info
high.r = 0
low.r = 0
ratios.NY = 0
for (x in 1:length(values)){
  COV = -1
  l1 = getCIRatio(SE[x], 12.9, values[x], 34.4, 1.036, COV)
  high.r[x] = l1[2]
  low.r[x] = l1[1]
  ratios.NY[x] = values[x]/34.4
}

high.r
```

```
##  [1] 1.612364 2.153618 1.630659 1.777704 1.665591 1.707003 1.802117
##  [8] 1.685373 1.655622 3.314536 1.663589
```

```
low.r
```

```
##  [1] 0.4472010 0.7286916 0.4564905 0.5399998 0.4773464 0.5001193 0.5532637
##  [8] 0.4866129 0.4687120 1.2460255 0.4753074
```

```
ratios.NY
```

```
##  [1] 0.8743553 1.2236379 0.8860655 0.9839439 0.9097497 0.9369983 0.9999387
##  [8] 0.9220816 0.9018515 1.9361128 0.9080339
```