Final Exam Markdown: Group 2

The fall 2020 IST722 final exam has two parts: a statistical analysis you develop in class and a report that you write on your own time. The first part, which you are working on right now, involves using this Markdown file to produce all of the diagnostics, graphs, and statistical output you will need for your report.

The **Knit** button at the top of the code window will generate an html file that includes both content as well as the output of any embedded R code chunks within this document. When you open this html file in a browser, you will be able to save it as a pdf file for submission to Blackboard. The file you submit to Blackboard must contain all of the output that you will use in your report. Any questions you answer with results or graphics not appearing in your PDF will not receive credit.

This initial code block opens the file and produces basic diagnostics. Make sure you have set the working directory and the file name correctly.

```
# First, set the working directory to wherever you have stored your data file.
getwd()
```

```
## [1] "/Users/sijinzhou/Desktop/IST772"
```

```
setwd("/Users/sijinzhou/Desktop/IST772")

# In this block, modify the file name of the Rdata file to match the name of
# the file you downloaded from Blackboard. Change the 1 to whatever number you have
load("datasets9.RData")

# If you can't get that to work, comment out the previous line and try file.choose()
to select
# your data file using a file selection dialog:
#
# load(file.choose())

# Now make sure that the two datasets you will be using are available.
str(usVaccines)
```

```
##  Time-Series [1:38, 1:5] from 1980 to 2017: 83 84 83 84 84 85 88 88 89 81 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "DTP1" "HepB_BD" "Pol3" "Hib3" ...
```

```
dim(usVaccines)
```

```
## [1] 38  5
```

```
str(districts)
```

```
## 'data.frame':    700 obs. of  12 variables:
##  $ DistrictName    : Factor w/ 846 levels "ABC Unified",..: 339 520 618 434 153 72
6 345 331 666 206 ...
##  $ WithoutDTP      : num  2 12 6 21 14 0 20 12 4 3 ...
##  $ WithoutPolio    : num  2 11 4 23 14 0 20 12 4 5 ...
##  $ WithoutMMR      : num  2 10 5 25 14 0 20 13 4 5 ...
##  $ WithoutHepB     : num  2 5 2 10 14 1 20 4 1 1 ...
##  $ PctUpToDate     : num  98 84 94 72 86 99 80 85 94 88 ...
##  $ DistrictComplete: logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ PctBeliefExempt : num  2 2 1 6 14 0 20 2 0 0 ...
##  $ PctChildPoverty : num  51 15 35 20 16 10 13 27 29 35 ...
##  $ PctFamilyPoverty: num  31 6 16 11 5 5 4 12 18 17 ...
##  $ Enrolled        : num  47 2503 337 173 101 ...
##  $ TotalSchools    : num  1 27 5 2 1 9 1 5 38 14 ...
```

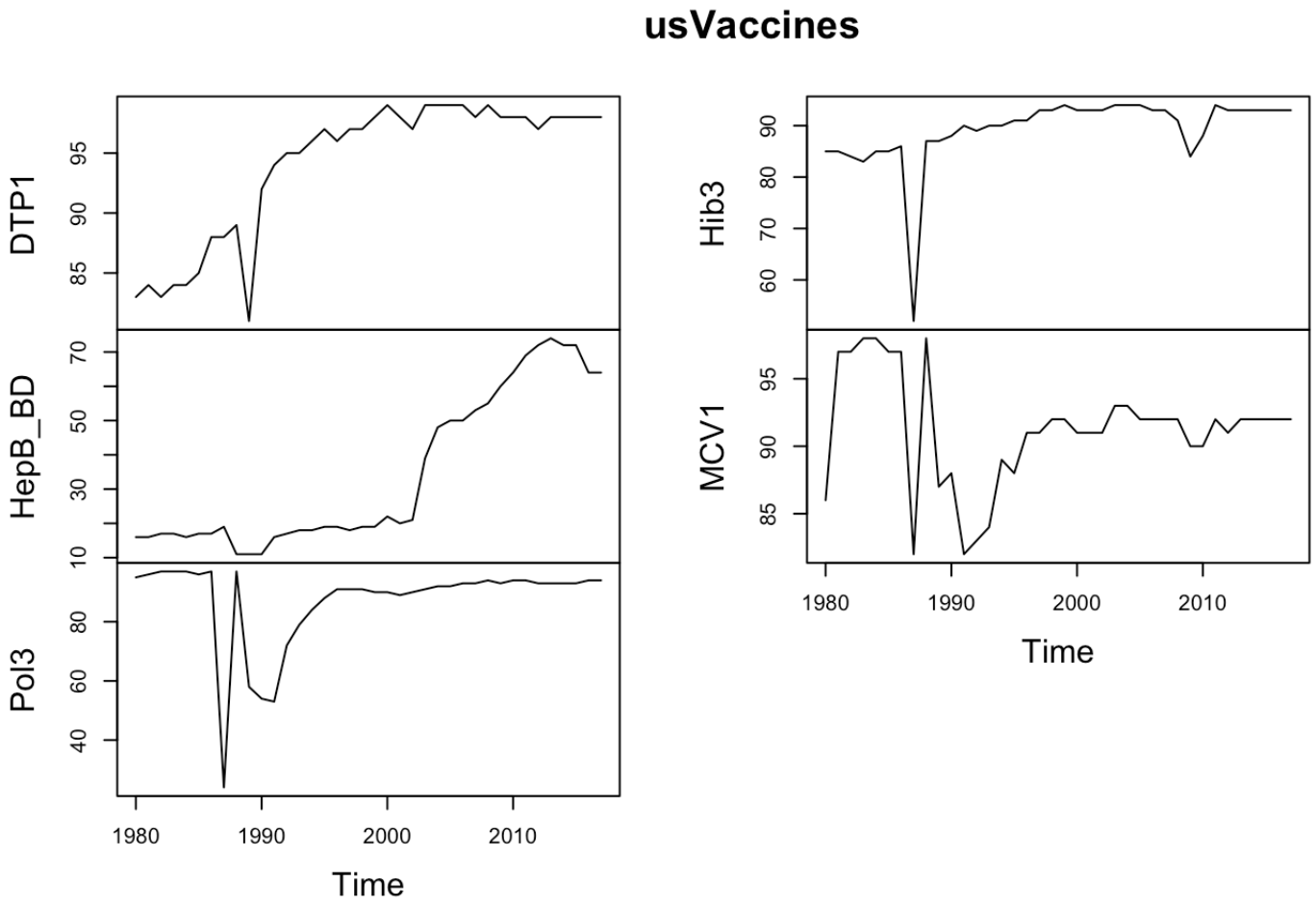
```
dim(districts)
```

```
## [1] 700  12
```

Plotting the Time Series Data

A basic time series plot can be valuable for examining trends, periodicity, and other aspects of a time series. You don't necessarily need to include this plot in your report, particularly because you only care about vaccination levels in the most recent few years.

```
plot(usVaccines)
```

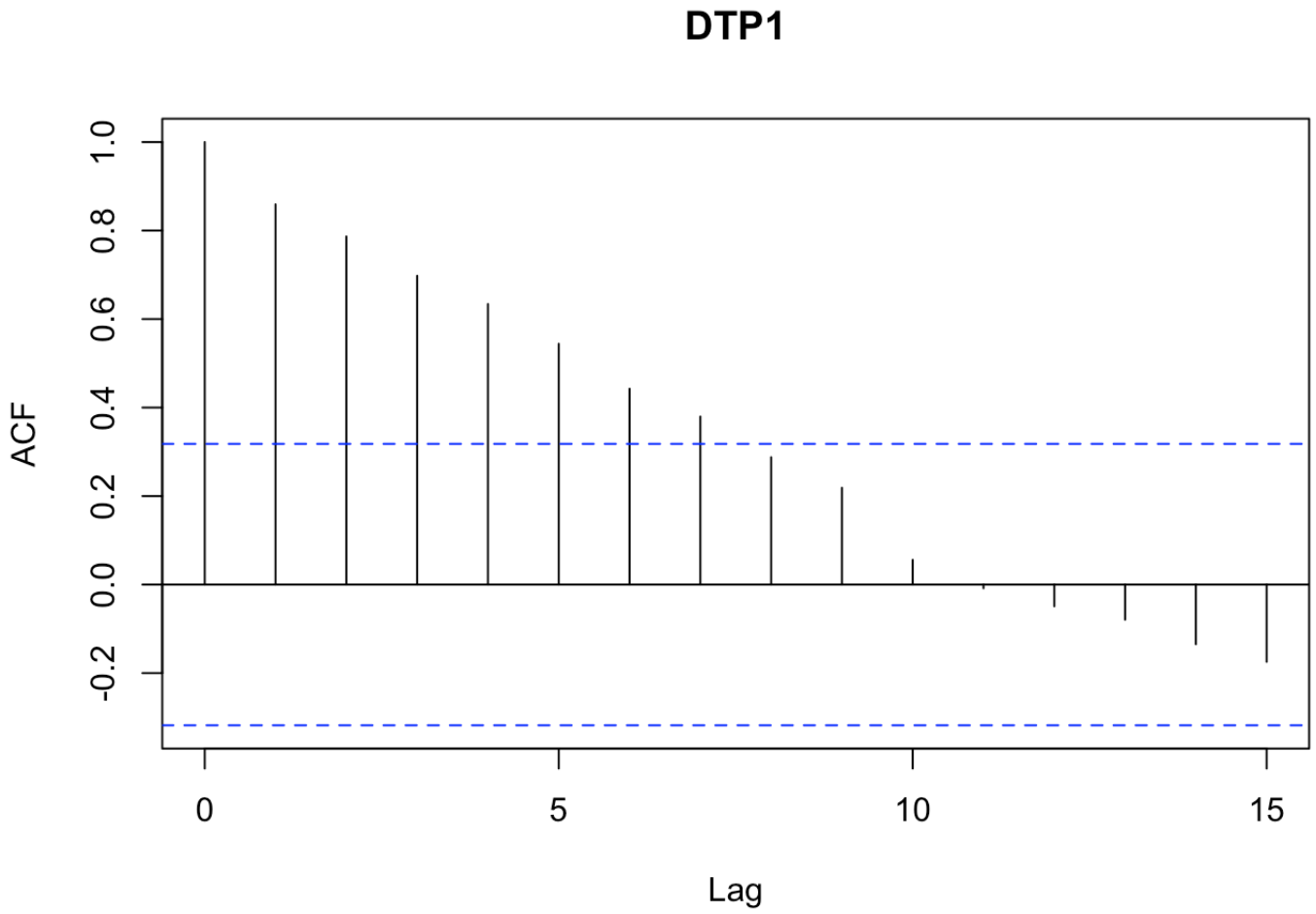


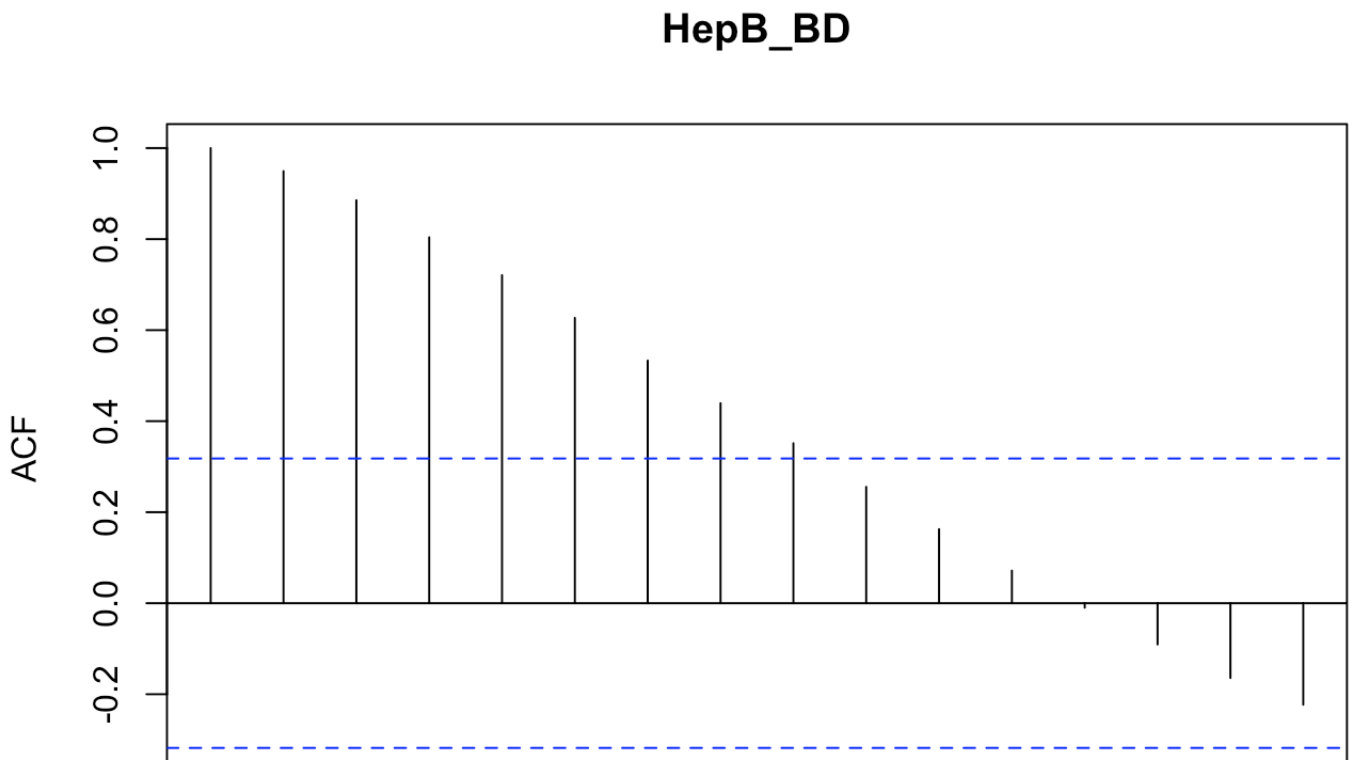

```
# Key to abbreviations (from WHO)
# DTP - Diptheria, Tetanus, Pertussis
# HepB_BD - Hepatitis B, Birth Dose
# Pol3 - Inactivated polio vaccine
# Hib3 - Haemophilus influenza type B vaccine
# MCV - Measles-containing-vaccine first-dose
```

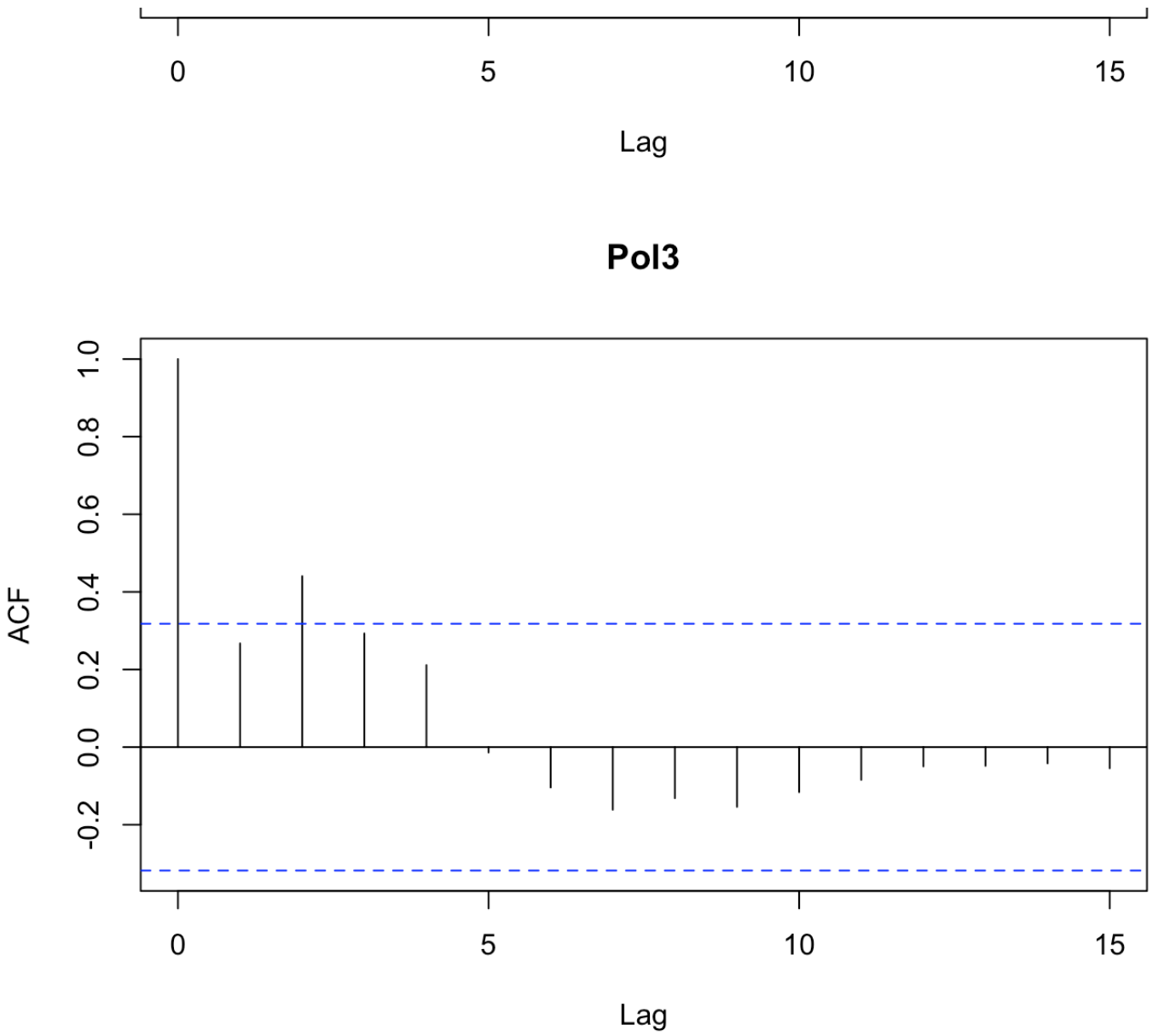
Time Series Autocorrelation

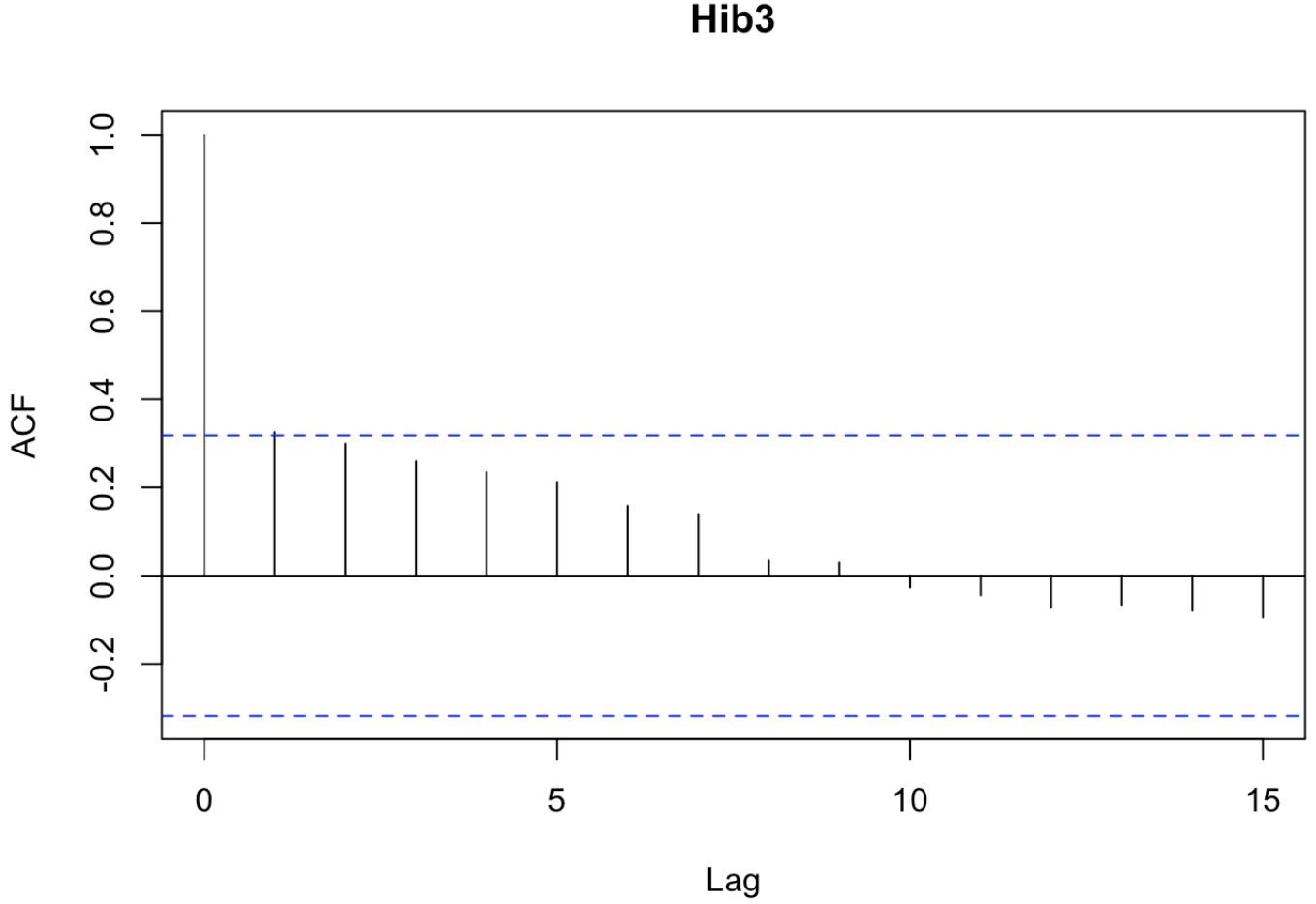
One of the most basic diagnostics for time series is an autocorrelation plot. One of the exam questions asks you to interpret these plots. You need not modify this block of code, as it produces all five of the plots you need.

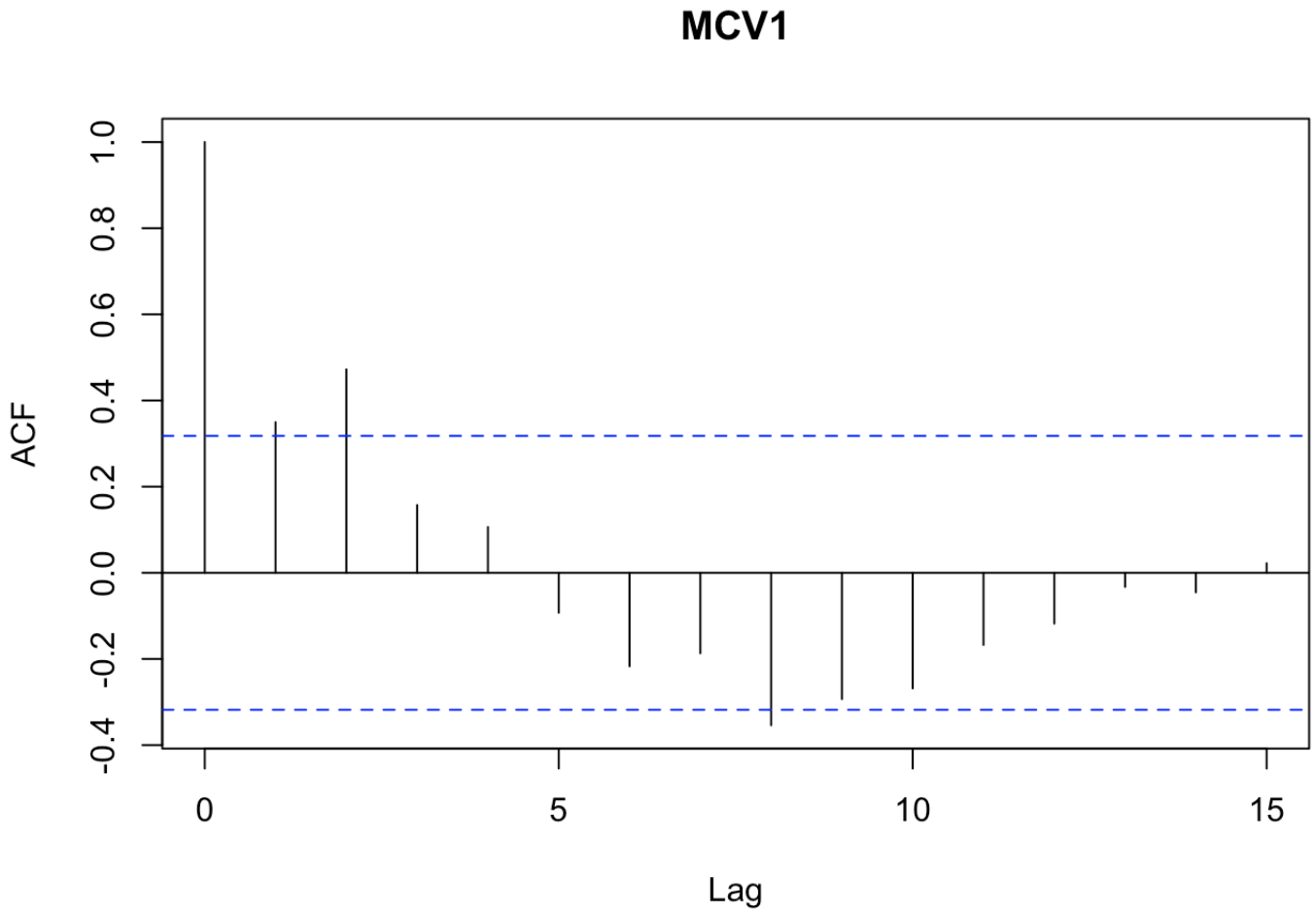
```
# This runs an ACF on each of the five time series
for (i in 1:5) {
  acf(usVaccines[,i], main=attr(usVaccines,"dimnames")[[2]][i])
}
```


DTP1
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
5
10
15
Lag


HepB_BD
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2


0
5
10
15
Lag
Pol3
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
5
10
15
Lag


Hib3
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
5
10
15
Lag



Time Series Changepoint Analysis of Means

This code calculates a changepoint analysis of means that is designed to find the last mean changepoint within each of the five series. As long as you have the changepoint package installed, there should be no need to modify this code.

```
library(changepoint)
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.6.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Successfully loaded changepoint package version 2.2.2
##  NOTE: Predefined penalty values changed in version 2.2.  Previous penalty values
with a postfix 1 i.e. SIC1 are now without i.e. SIC and previous penalties without a
postfix i.e. SIC are now with a postfix 0 i.e. SIC0. See NEWS and help files for furt
her details.
```

```
# This finds the final mean changepoint for each time series
print("Last mean changepoint for each time series:")
```

```
## [1] "Last mean changepoint for each time series:"
```

```
for (i in 1:5) {
  print(attr(usVaccines,"dimnames")[[2]][i])
  print(changepoint::cpt.mean(rev(usVaccines[,i]), method="AMOC")@cpts[1])
}
```

```
## [1] "DTP1"
## [1] 28
## [1] "HepB_BD"
## [1] 14
## [1] "Pol3"
## [1] 23
## [1] "Hib3"
## [1] 30
## [1] "MCV1"
## [1] 31
```

Time Series Subsetting

You will be calibrating your district-by-district vaccine results by comparing to the U.S. mean level of vaccination during recent years. Subset the time series data to only include the most recent years during which all vaccine mean levels were stable. Use the changepoint output from the previous step to make this judgment call. Set the value of startPoint to the most appropriate starting point within the 38 year period of the time series data. Then calculate the mean level for each vaccination during that time period.

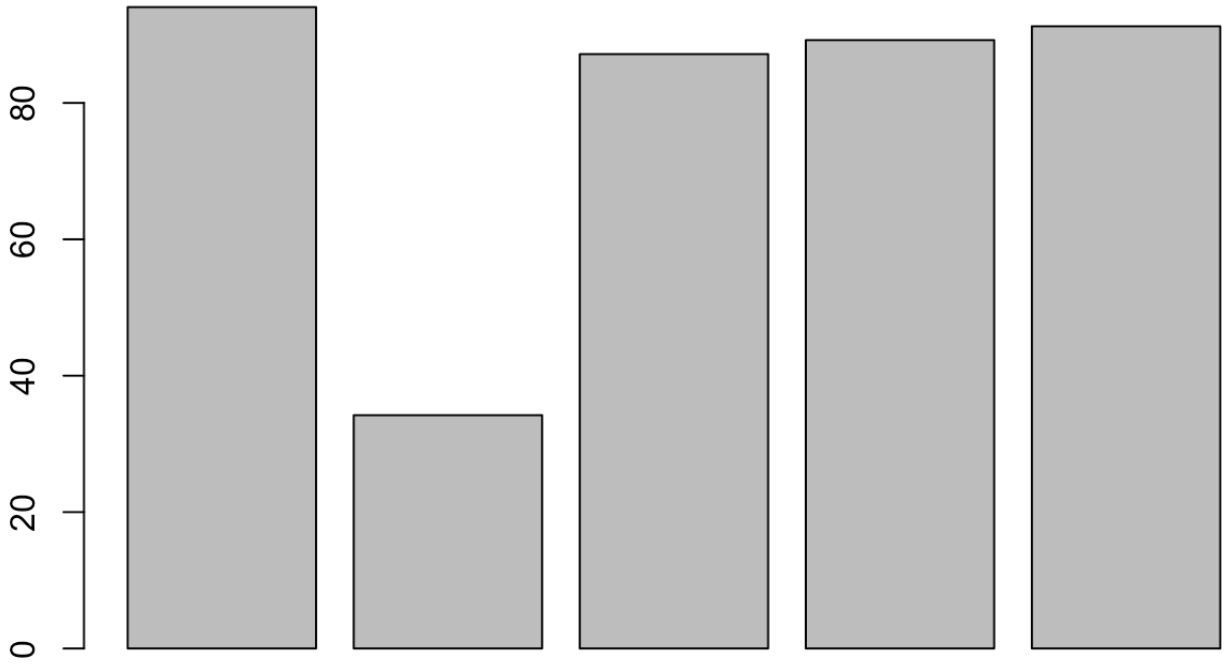
Optionally add a bar plot that puts all of the mean vaccination levels into the same graphic.

```
startPoint <- 1 # Change the starting point for calculating a mean of recent observat
ions only.
# Choose the most reasonable value by looking at the results of the previous code blo
ck.

# Now calculate the mean vaccination level during recent years for
# each type of vaccine.
apply(usVaccines[startPoint:38,], MARGIN=2, FUN=mean)
```

```
##     DTP1  HepB_BD     Pol3     Hib3     MCV1
## 94.05263 34.21053 87.15789 89.21053 91.23684
```

```
# Optionally add a barplot() to display these means.
df <-data.frame(apply(usVaccines[startPoint:38,], MARGIN=2, FUN=mean))
colnames(df)<-'value'
barplot(df$value)
```



That's the end of the analysis of the usVaccines data. Next, you will conduct diagnostics on your main dataset, the district by district results for a random sample of n=700 California school districts. Refer to the exam specification for information on the meaning of each variable in your data set.

Descriptive Statistics for Later Use

Clients expect basic descriptive statistics on the major variables of importance. Descriptives may also be important in guiding your analyses. The code below produces a summary of the data set, but you may also want to compute some other statistics that do not appear in the output of summary().

```
summary(districts)
```

```
##                    DistrictName   WithoutDTP     WithoutPolio
##  ABC Unified              :  1   Min.   : 0.00   Min.   : 0.000
##  Ackerman Charter         :  1   1st Qu.: 3.00   1st Qu.: 3.000
##  Acton-Agua Dulce Unified:  1   Median : 7.00   Median : 6.000
##  Adelanto Elementary      :  1   Mean   :10.21   Mean   : 9.779
##  Alameda Unified          :  1   3rd Qu.:14.00   3rd Qu.:13.000
##  Albany City Unified      :  1   Max.   :77.00   Max.   :77.000
##  (Other)                  :694
##    WithoutMMR     WithoutHepB      PctUpToDate     DistrictComplete
##  Min.   : 0.00   Min.   : 0.000   Min.   : 23.00   Mode :logical
##  1st Qu.: 3.00   1st Qu.: 2.000   1st Qu.: 84.00   FALSE:43
##  Median : 6.00   Median : 4.000   Median : 92.00   TRUE :657
##  Mean   :10.17   Mean   : 7.691   Mean   : 87.87
##  3rd Qu.:14.00   3rd Qu.:10.000   3rd Qu.: 96.00
##  Max.   :77.00   Max.   :77.000   Max.   :100.00
##
##  PctBeliefExempt  PctChildPoverty PctFamilyPoverty    Enrolled
##  Min.   : 0.000   Min.   : 2.00   Min.   : 0.00    Min.   :   10.00
##  1st Qu.: 1.000   1st Qu.:13.00   1st Qu.: 5.00    1st Qu.:   54.75
##  Median : 2.000   Median :21.00   Median : 9.00    Median :  207.50
##  Mean   : 5.621   Mean   :22.21   Mean   :11.39    Mean   :  635.88
##  3rd Qu.: 7.000   3rd Qu.:29.00   3rd Qu.:16.00    3rd Qu.:  686.25
##  Max.   :77.000   Max.   :72.00   Max.   :47.00    Max.   :54238.00
##
##   TotalSchools
##  Min.   :  1.00
##  1st Qu.:  1.00
##  Median :  3.00
##  Mean   :  7.34
##  3rd Qu.:  8.00
##  Max.   :582.00
##
```

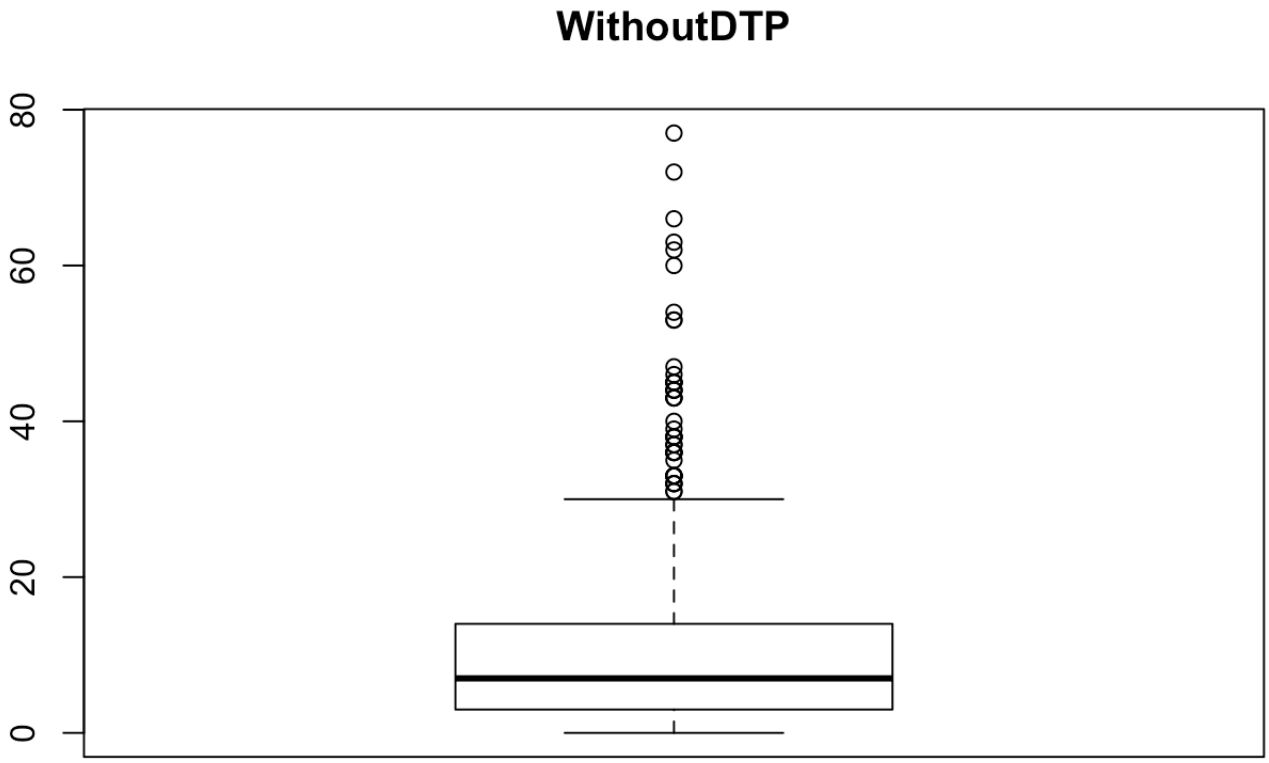
```
# Add any additional descriptive statistics that you want to view now or later.
str(districts)
```

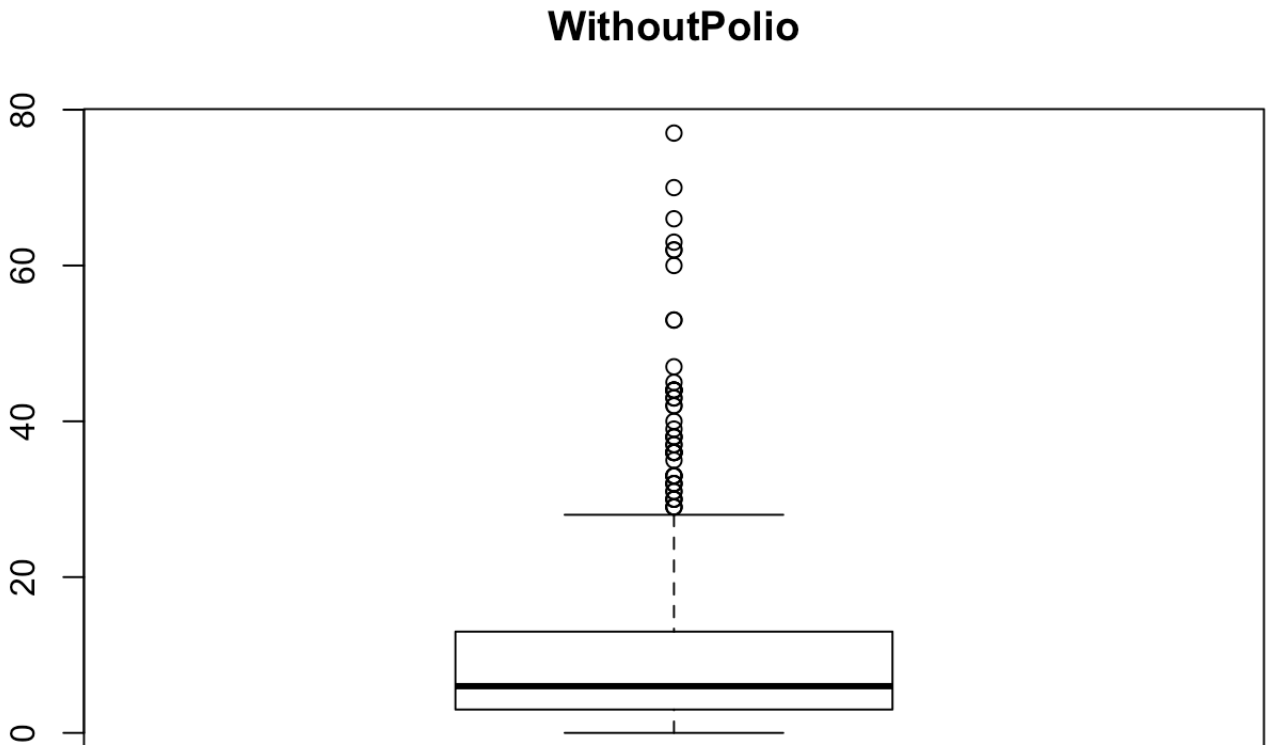
```
## 'data.frame':    700 obs. of  12 variables:
##  $ DistrictName    : Factor w/ 846 levels "ABC Unified",..: 339 520 618 434 153 72
6 345 331 666 206 ...
##  $ WithoutDTP      : num  2 12 6 21 14 0 20 12 4 3 ...
##  $ WithoutPolio    : num  2 11 4 23 14 0 20 12 4 5 ...
##  $ WithoutMMR      : num  2 10 5 25 14 0 20 13 4 5 ...
##  $ WithoutHepB     : num  2 5 2 10 14 1 20 4 1 1 ...
##  $ PctUpToDate     : num  98 84 94 72 86 99 80 85 94 88 ...
##  $ DistrictComplete: logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ PctBeliefExempt : num  2 2 1 6 14 0 20 2 0 0 ...
##  $ PctChildPoverty : num  51 15 35 20 16 10 13 27 29 35 ...
##  $ PctFamilyPoverty: num  31 6 16 11 5 5 4 12 18 17 ...
##  $ Enrolled        : num  47 2503 337 173 101 ...
##  $ TotalSchools    : num  1 27 5 2 1 9 1 5 38 14 ...
```

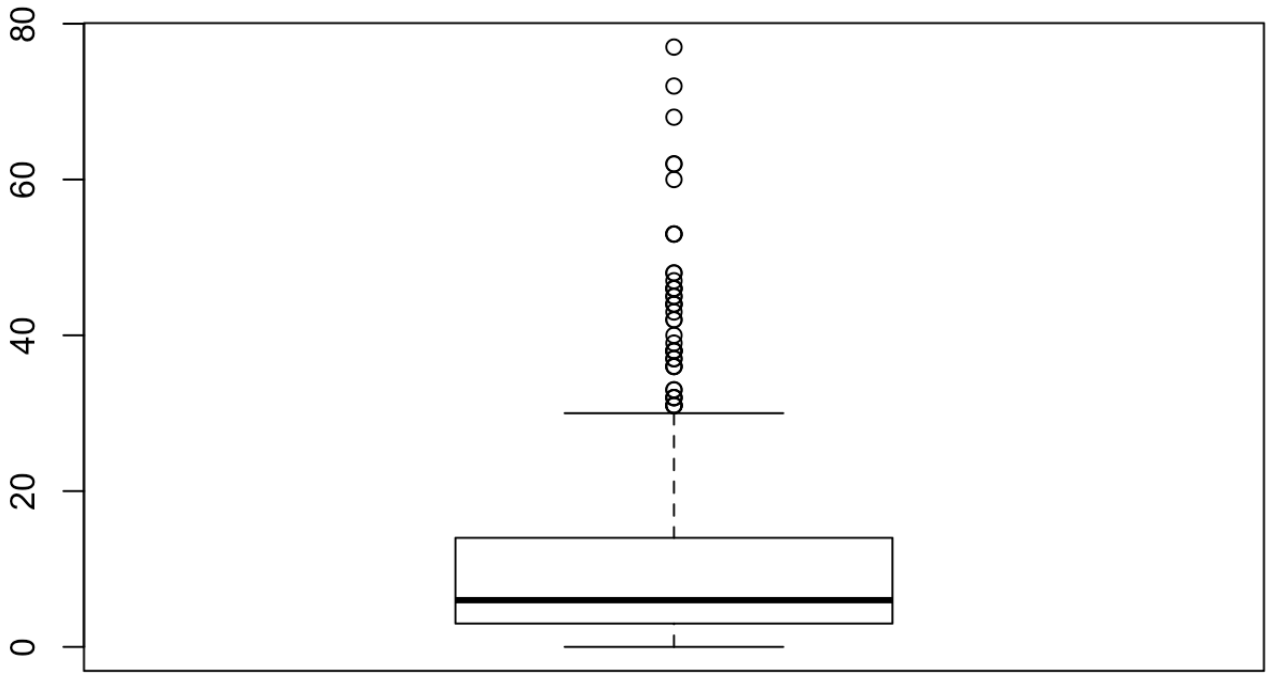
```
dim(districts)
```

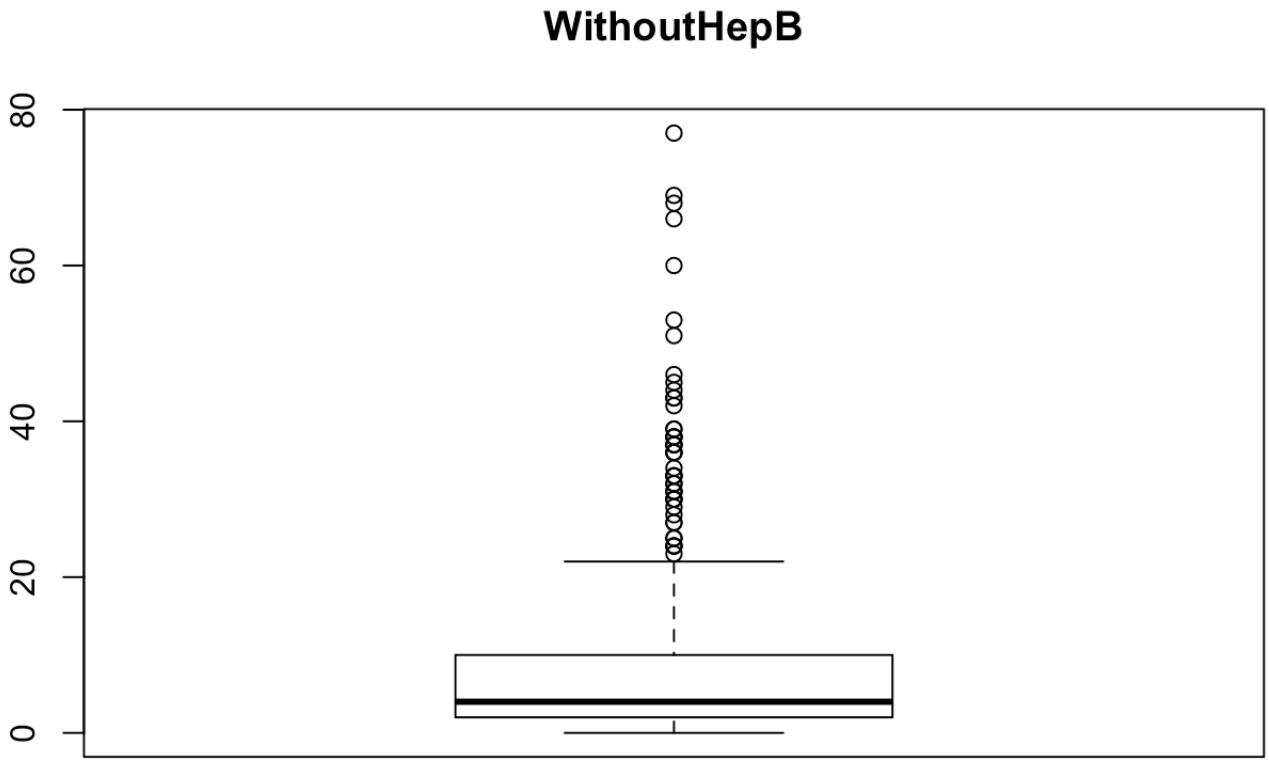
```
## [1] 700  12
```

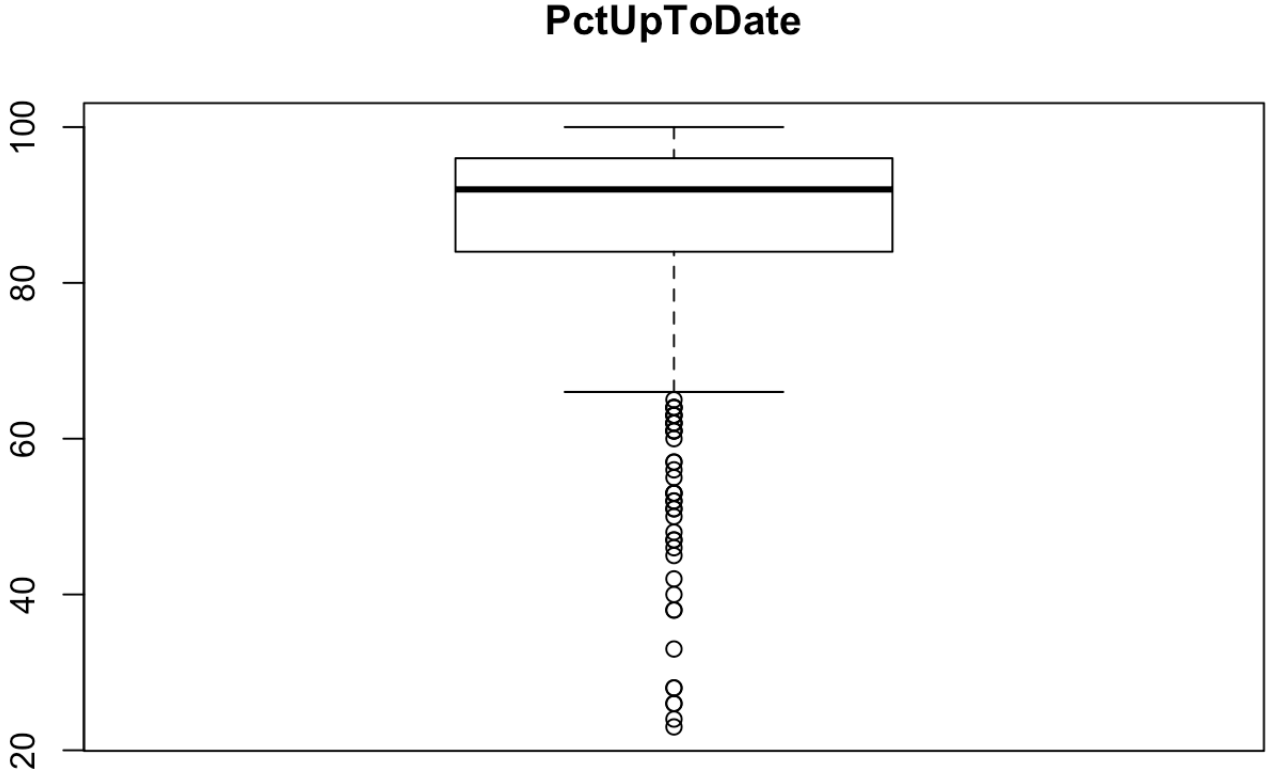
```
sub_districts <- districts[,c(-1,-7)]
for (i in 1:10) {
  boxplot(sub_districts[,i],main=attr(sub_districts,"names")[i])
}
```

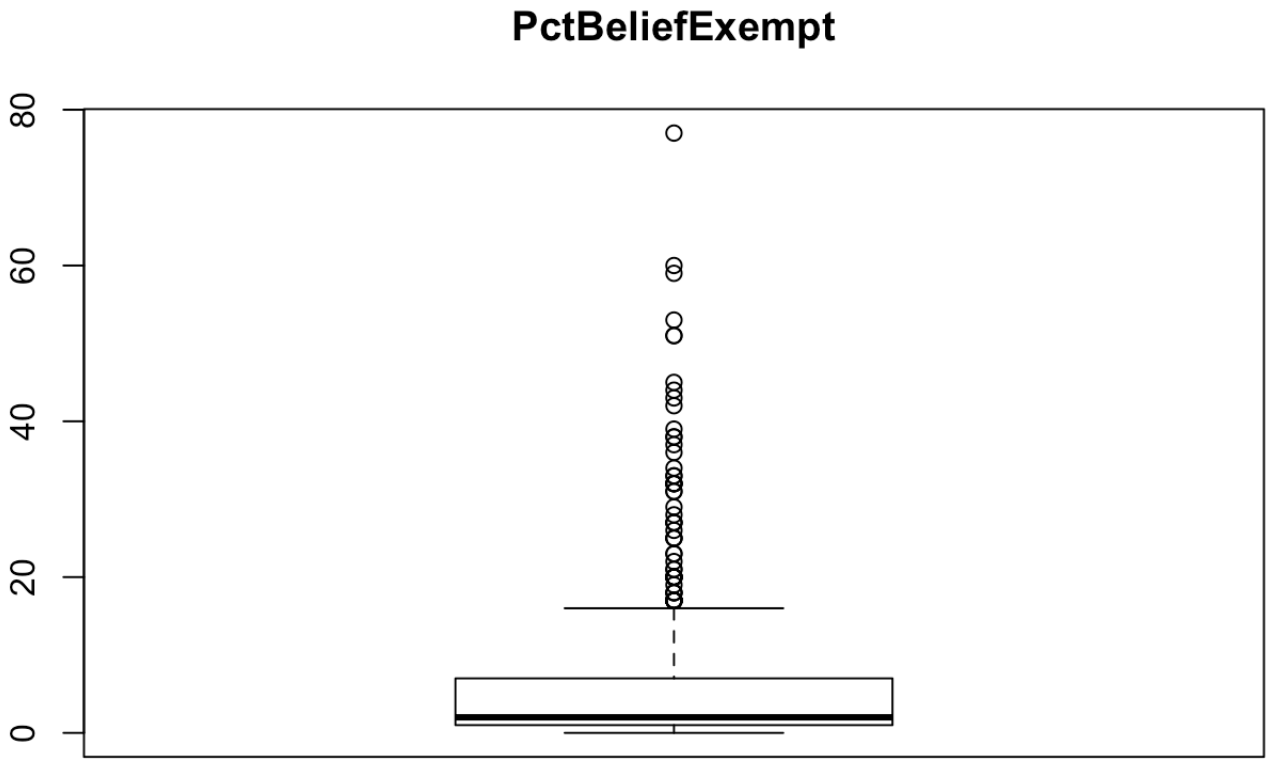

WithoutDTP
0
20
40
60
80

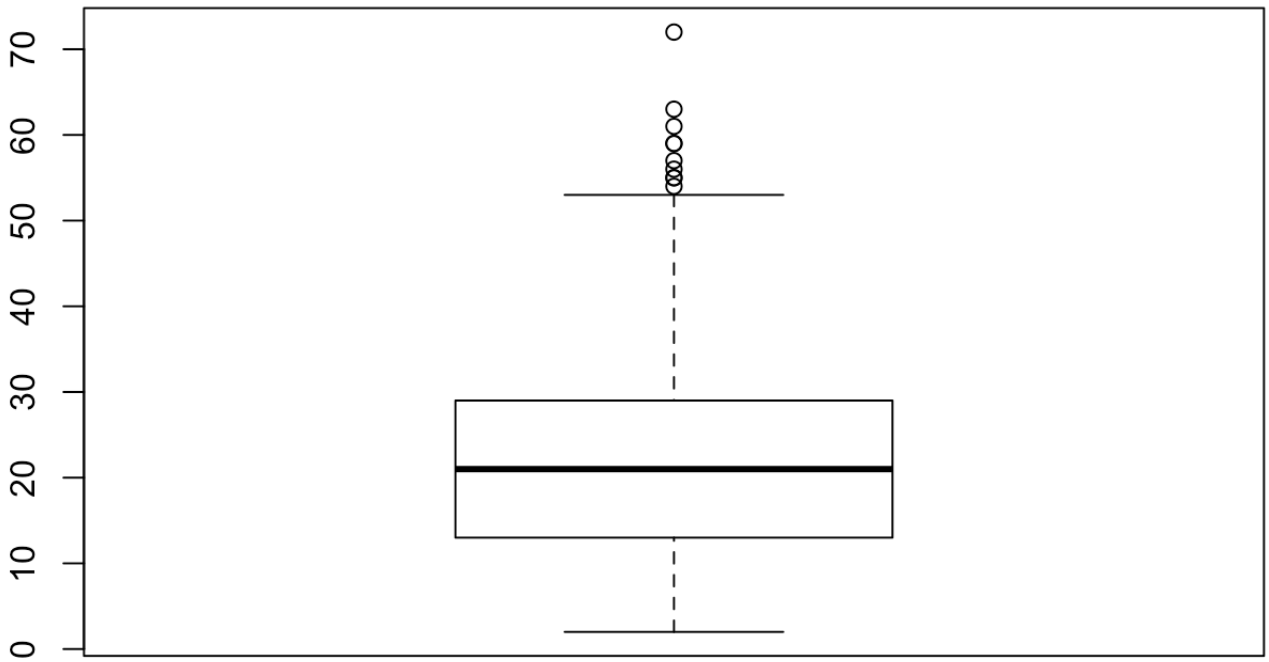

WithoutPolio
0
20
40
60
80

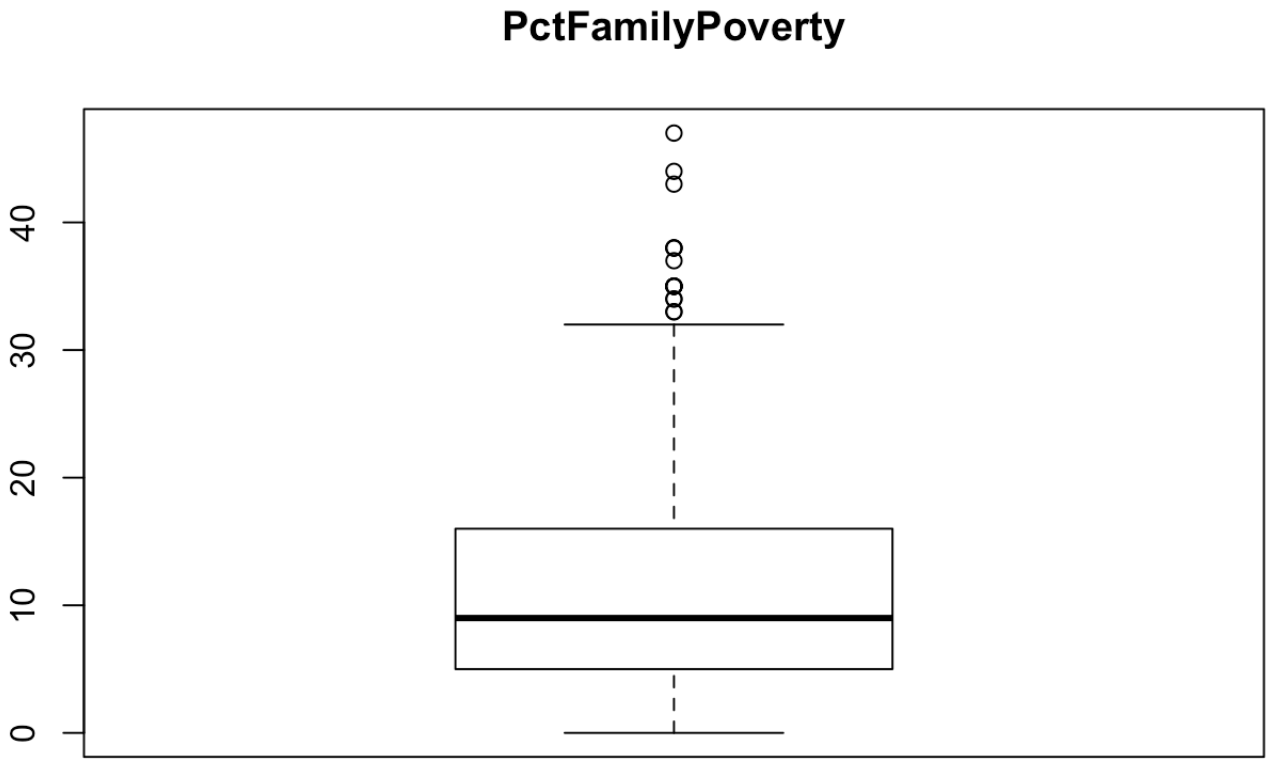

WithoutMMR
0
20
40
60
80

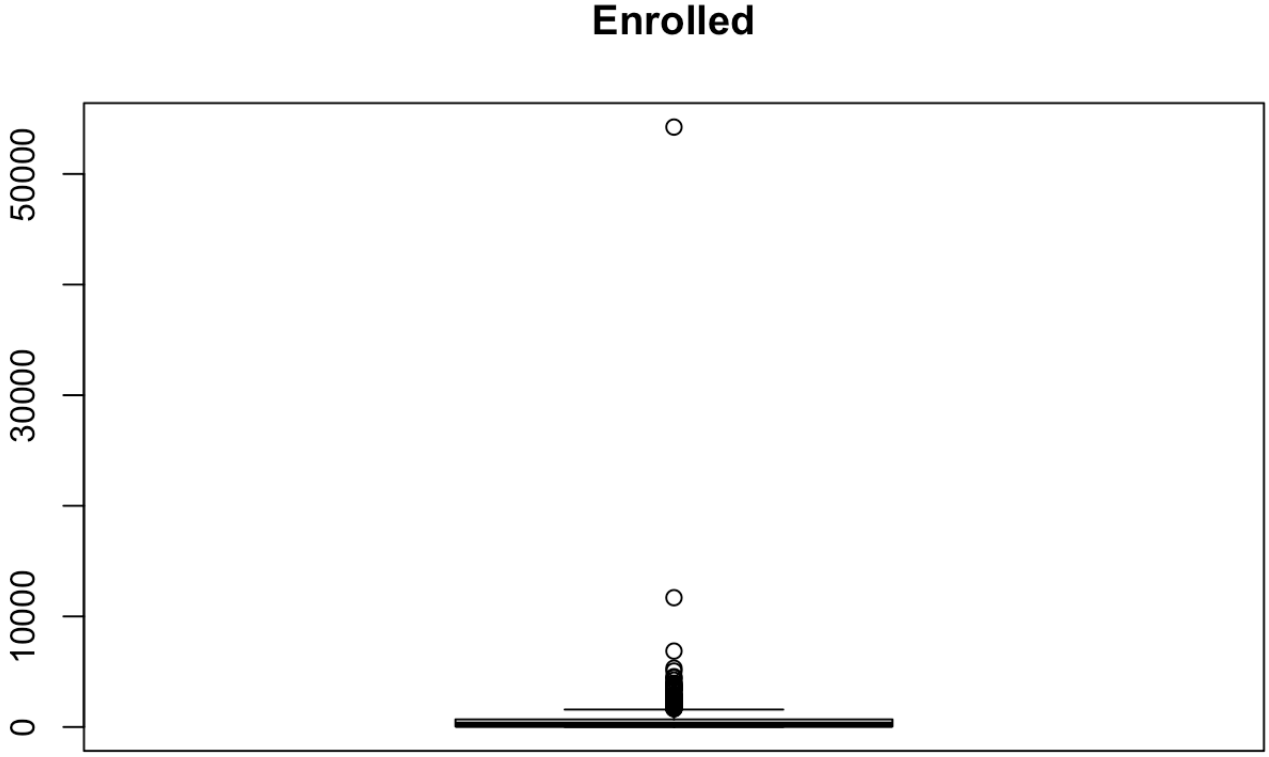

WithoutHepB
0
20
40
60
80

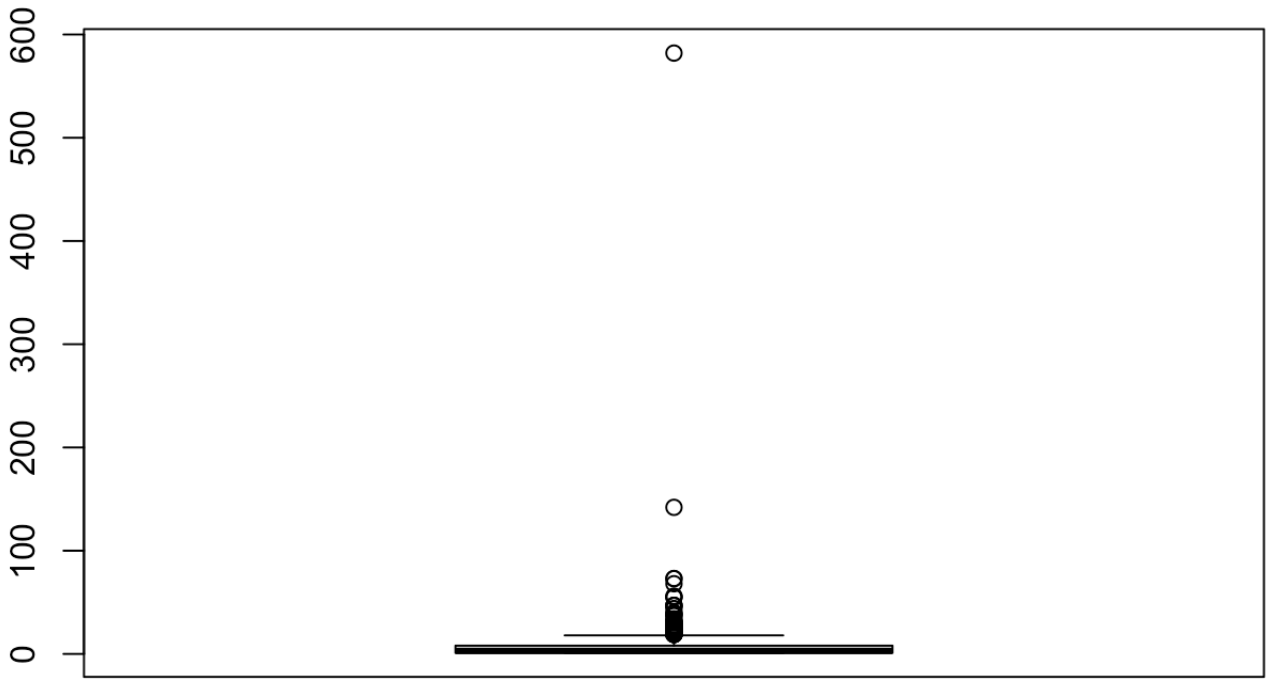

PctUpToDate
100
80
60
40
20


PctBeliefExempt
0
20
40
60
80


PctChildPoverty
0
10
20
30
40
50
60
70


PctFamilyPoverty
0
10
20
30
40


Enrolled
0
10000
30000
50000



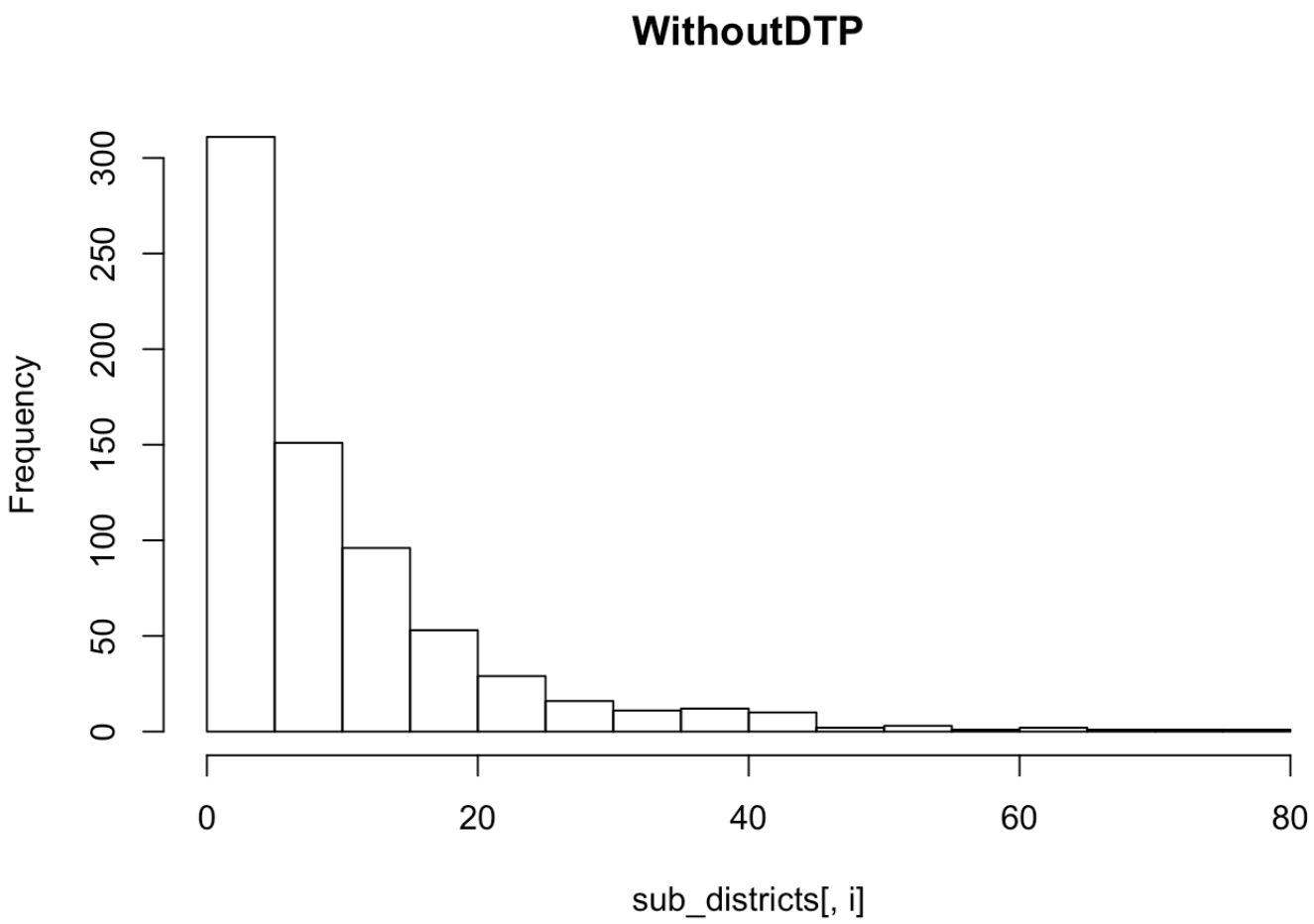

```
# This produces a complete correlation matrix, rounded to two decimal digits
round(cor(districts[,-1]),2)
```

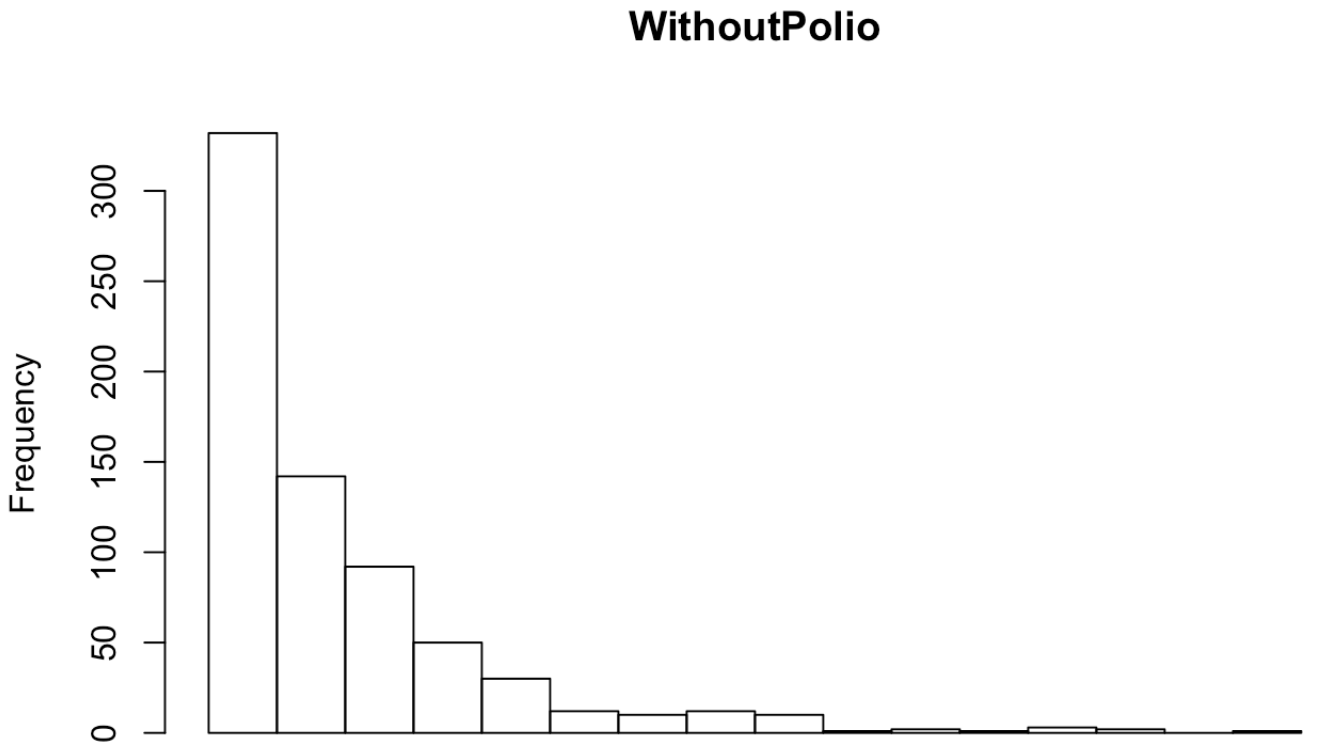
```
##                  WithoutDTP WithoutPolio WithoutMMR WithoutHepB PctUpToDate
## WithoutDTP             1.00         0.98       0.98        0.89       -0.96
## WithoutPolio           0.98         1.00       0.97        0.91       -0.95
## WithoutMMR             0.98         0.97       1.00        0.90       -0.97
## WithoutHepB            0.89         0.91       0.90        1.00       -0.85
## PctUpToDate           -0.96        -0.95      -0.97       -0.85        1.00
## DistrictComplete      -0.06        -0.06      -0.05       -0.02        0.06
## PctBeliefExempt        0.81         0.83       0.80        0.92       -0.73
## PctChildPoverty       -0.21        -0.21      -0.20       -0.22        0.21
## PctFamilyPoverty      -0.25        -0.26      -0.25       -0.27        0.25
## Enrolled              -0.07        -0.07      -0.07       -0.08        0.06
## TotalSchools          -0.06        -0.06      -0.06       -0.07        0.05
##                  DistrictComplete PctBeliefExempt PctChildPoverty
## WithoutDTP                -0.06            0.81           -0.21
## WithoutPolio              -0.06            0.83           -0.21
## WithoutMMR                -0.05            0.80           -0.20
## WithoutHepB               -0.02            0.92           -0.22
## PctUpToDate                0.06           -0.73            0.21
## DistrictComplete           1.00           -0.01           -0.06
## PctBeliefExempt           -0.01            1.00           -0.19
## PctChildPoverty           -0.06           -0.19            1.00
## PctFamilyPoverty          -0.08           -0.25            0.86
## Enrolled                  -0.20           -0.09            0.03
## TotalSchools              -0.22           -0.08            0.02
##                  PctFamilyPoverty Enrolled TotalSchools
## WithoutDTP                -0.25    -0.07        -0.06
## WithoutPolio              -0.26    -0.07        -0.06
## WithoutMMR                -0.25    -0.07        -0.06
## WithoutHepB               -0.27    -0.08        -0.07
## PctUpToDate                0.25     0.06         0.05
## DistrictComplete          -0.08    -0.20        -0.22
## PctBeliefExempt           -0.25    -0.09        -0.08
## PctChildPoverty            0.86     0.03         0.02
## PctFamilyPoverty           1.00     0.04         0.04
## Enrolled                   0.04     1.00         0.99
## TotalSchools               0.04     0.99         1.00
```

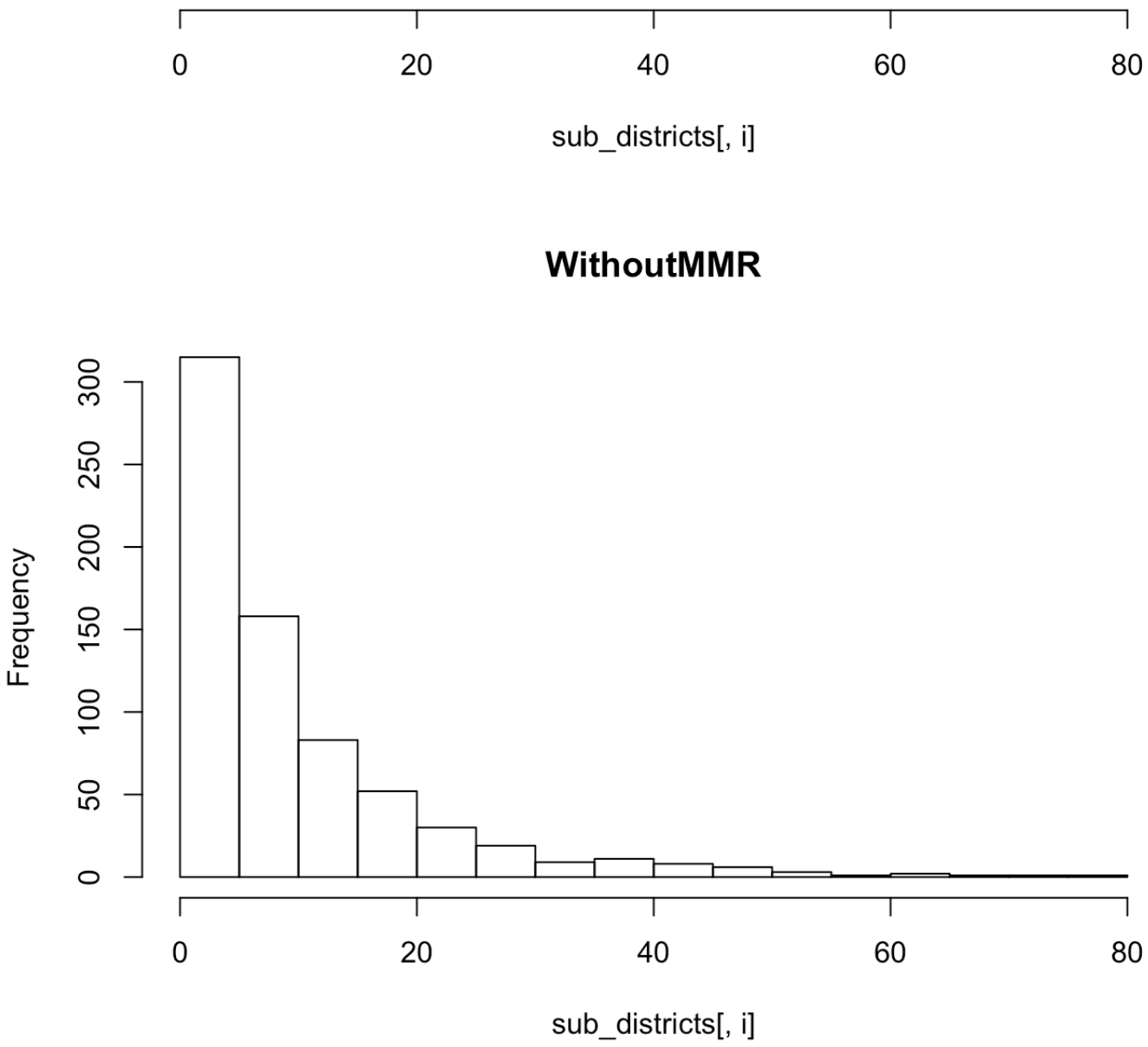
Plotting Histograms as Diagnostics

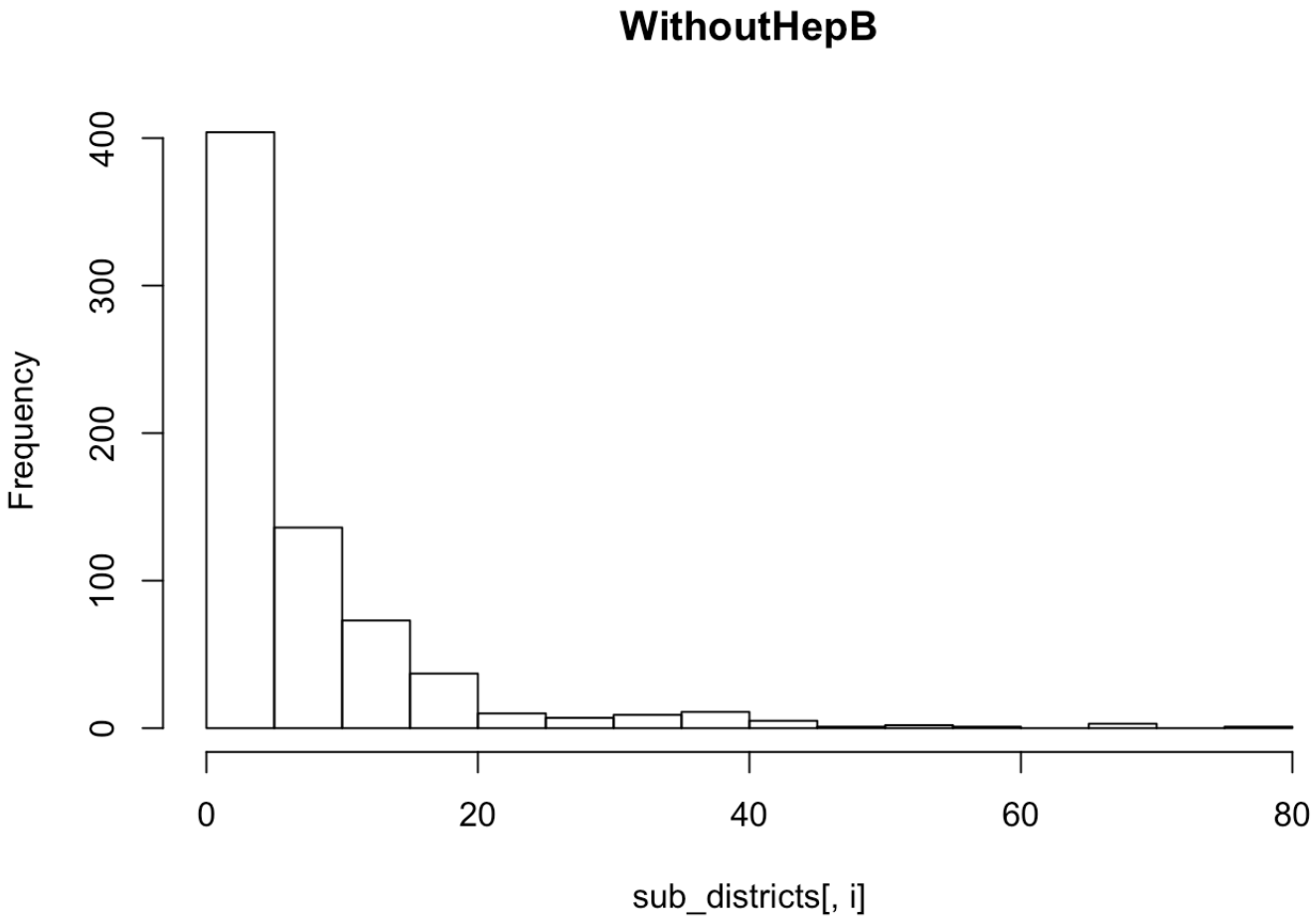
Most reports that you create for clients will NOT benefit from including histograms because histograms are very low level and rarely communicate much of interest. Nonetheless, you should look at them yourself to make sure you understand your data. Add any additional histograms that you wish to view.

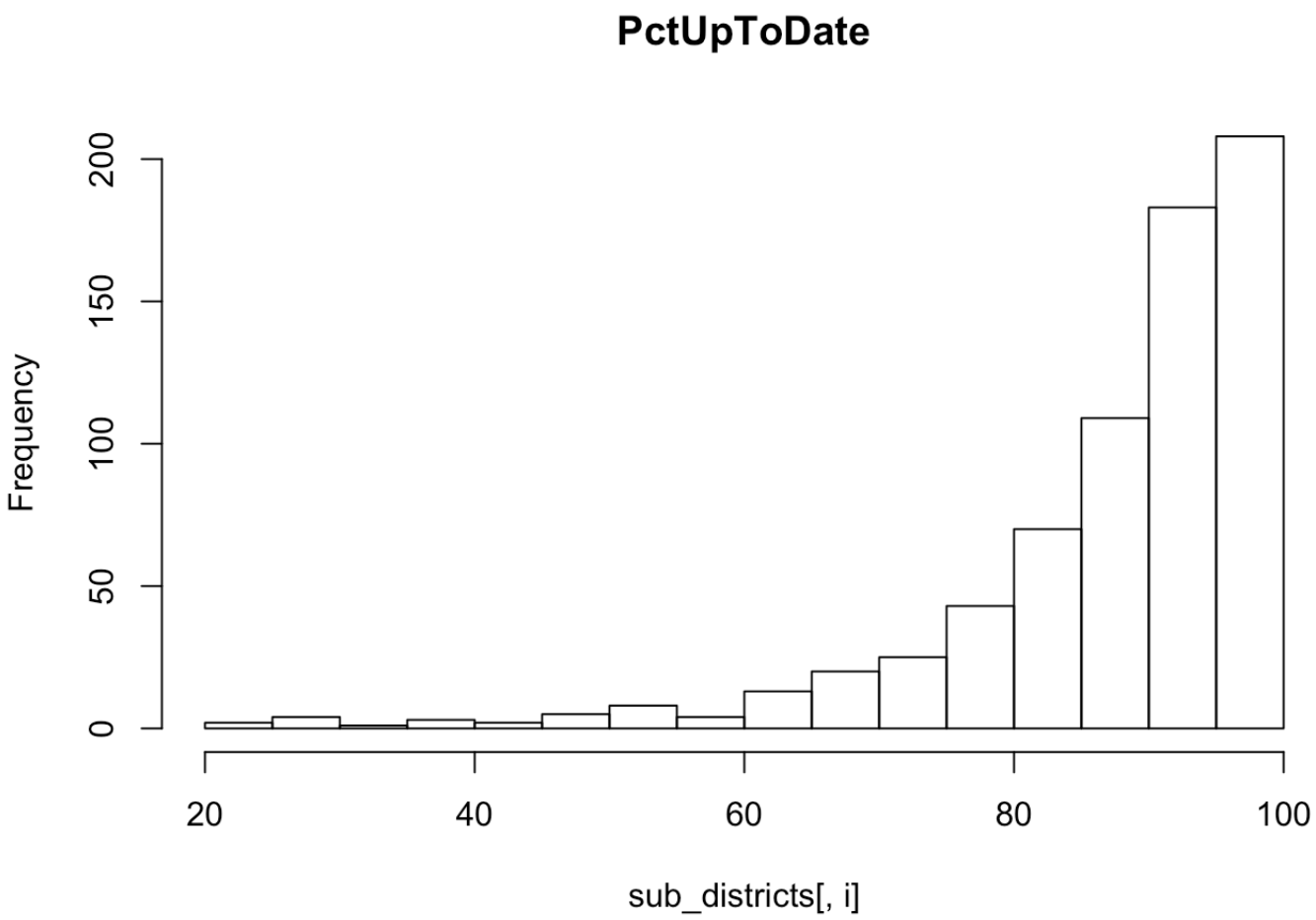
```
# Add any additional histograms that you want to view now or later.
for (i in 1:10) {
  hist(sub_districts[,i],main=attr(sub_districts,"names")[i])
}
```

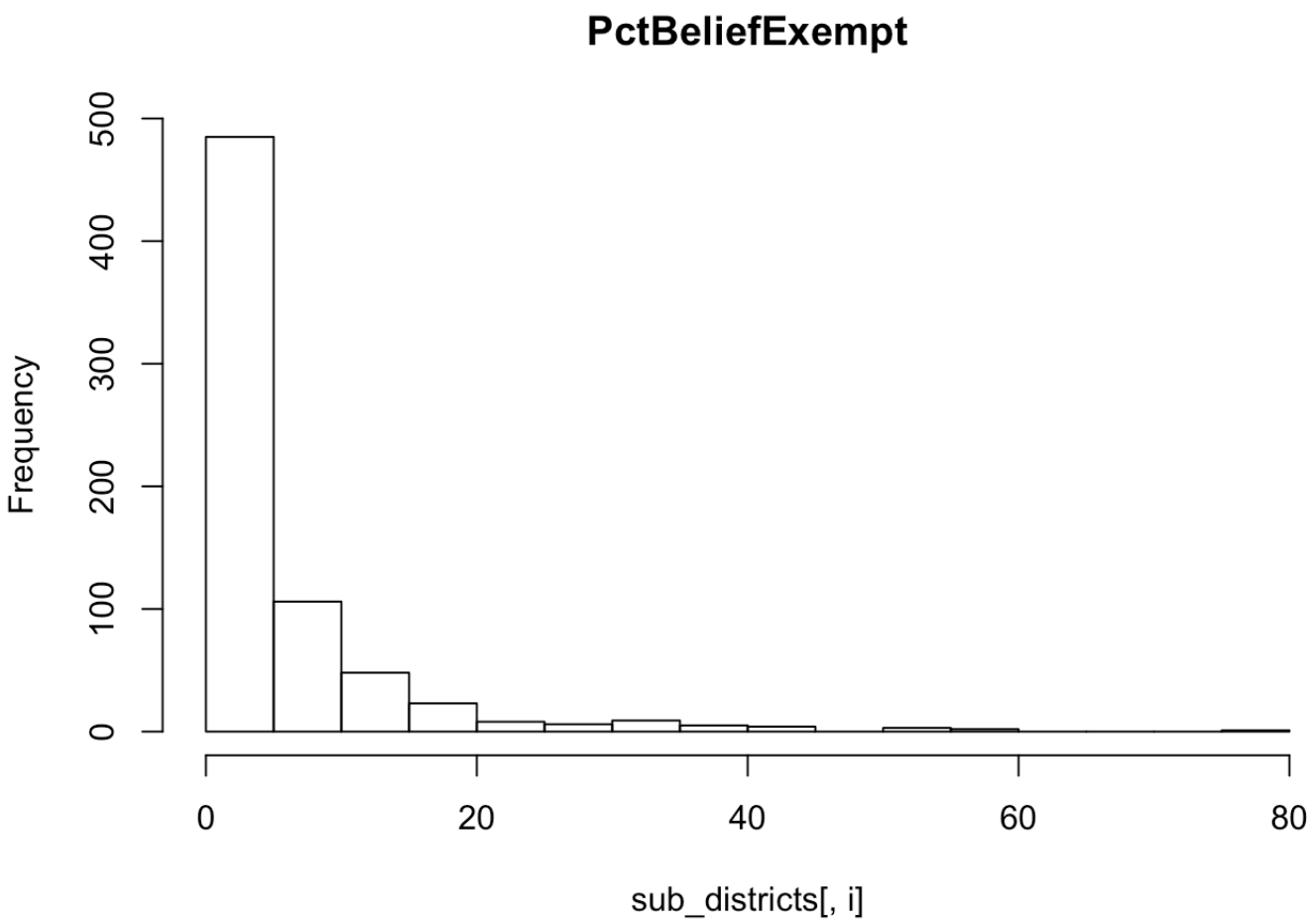

WithoutDTP
Frequency
0
50
100
150
200
250
300
0
20
40
60
80
sub_districts[, i]

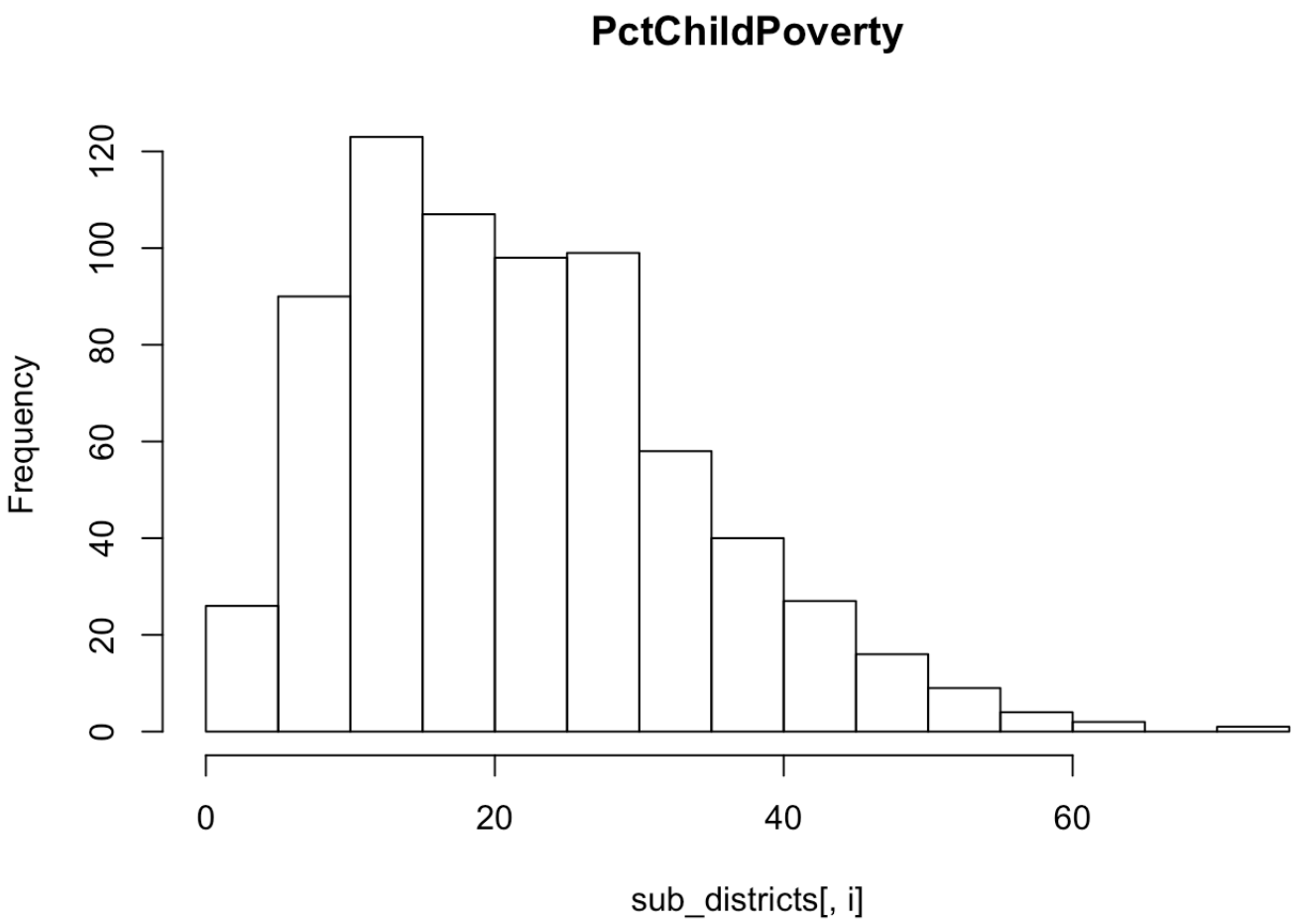

WithoutPolio
Frequency
0
50
100
150
200
250
300

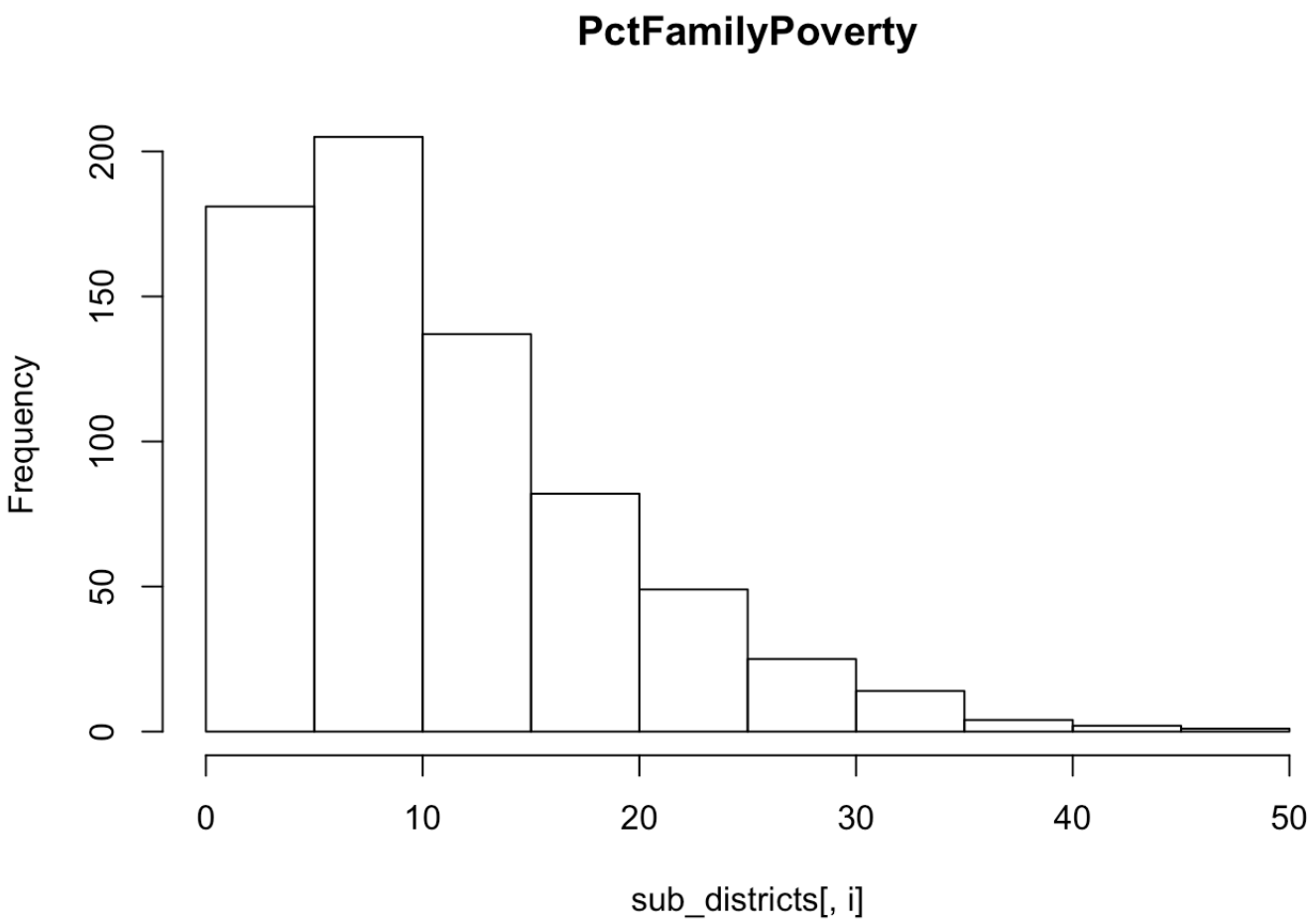

0
20
40
60
80
sub_districts[, i]
WithoutMMR
Frequency
0
50
100
150
200
250
300
0
20
40
60
80
sub_districts[, i]

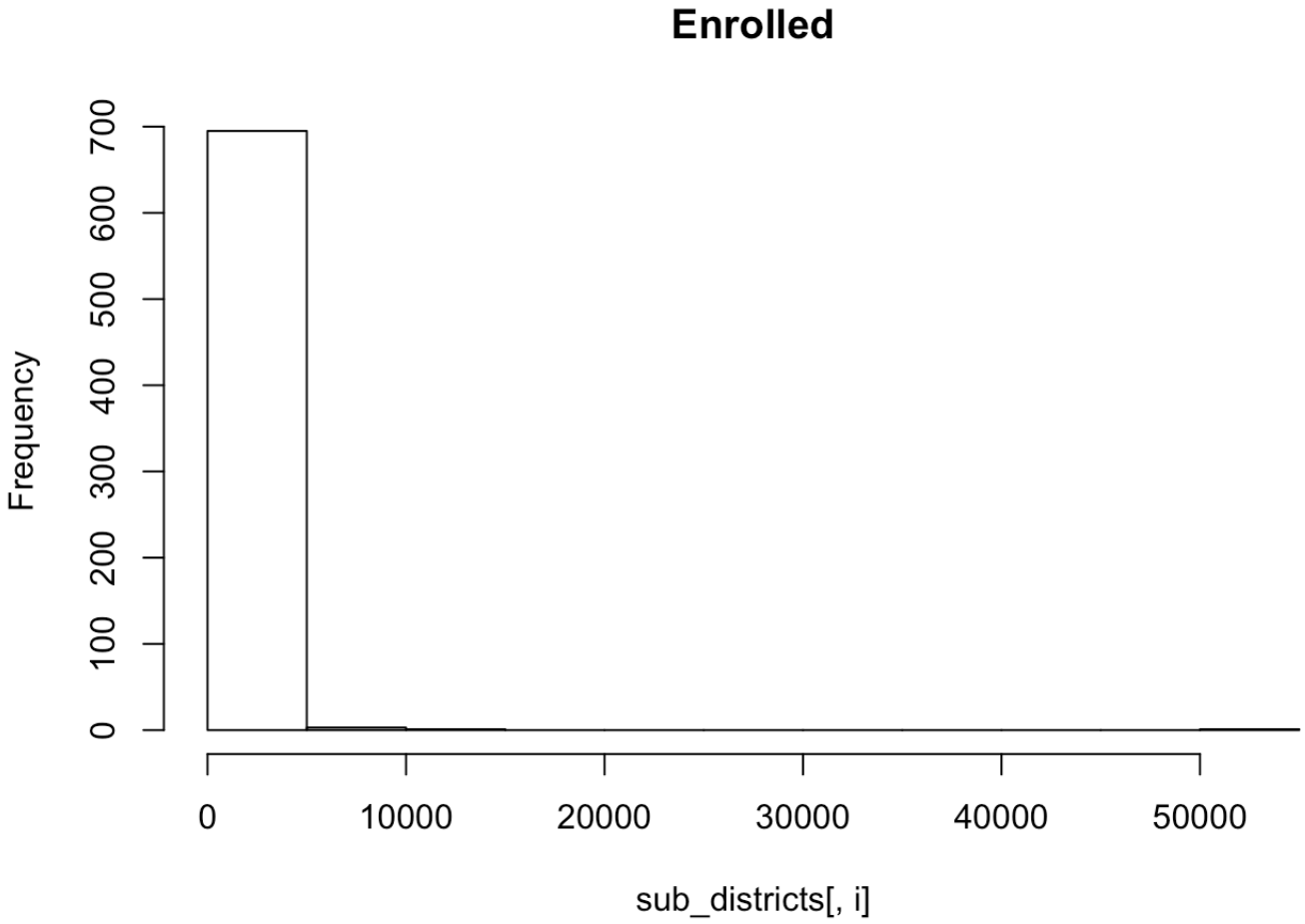

WithoutHepB
Frequency
0
100
200
300
400
0
20
40
60
80
sub_districts[, i]

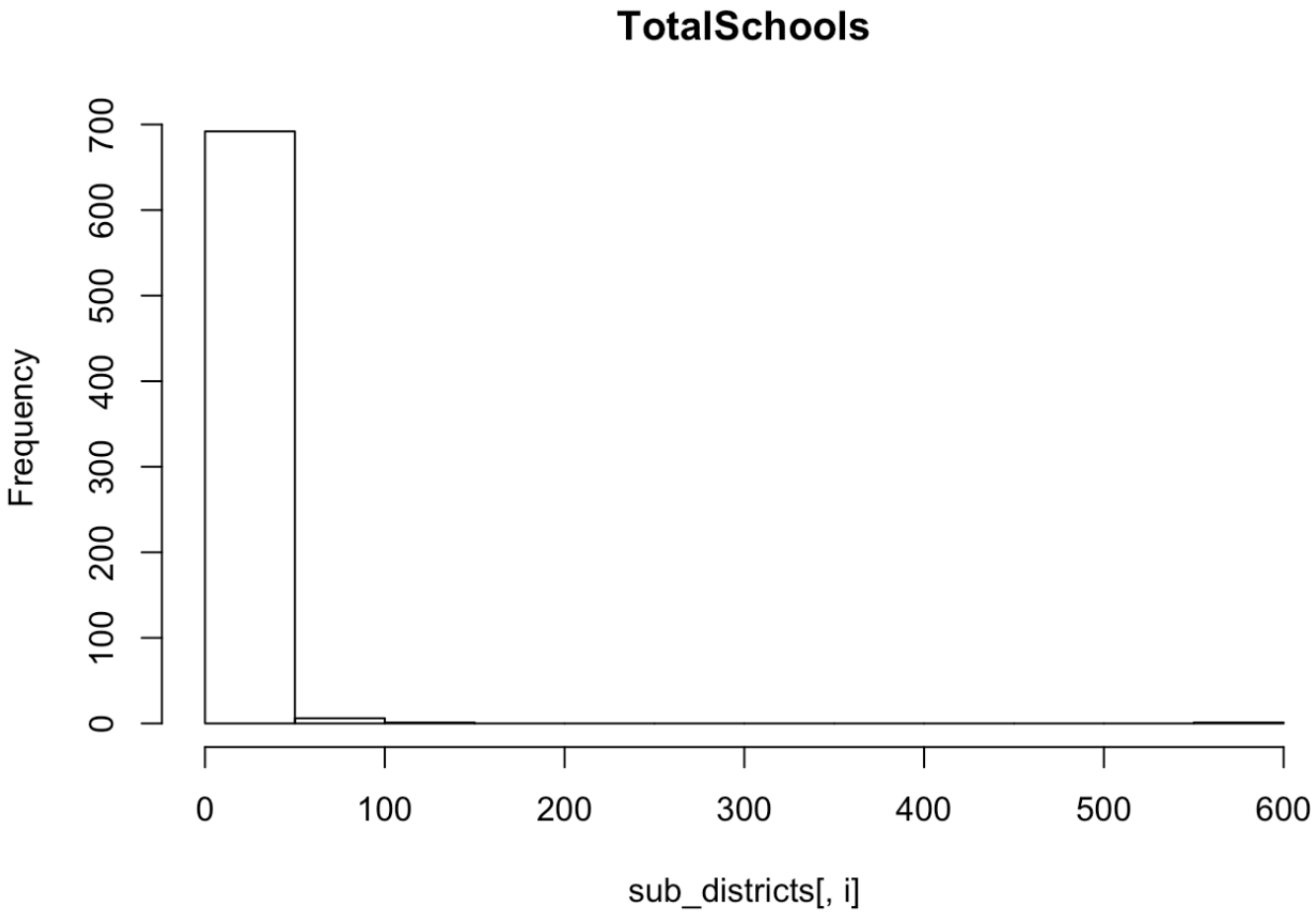

PctUpToDate
Frequency
0
50
100
150
200
20
40
60
80
100
sub_districts[, i]


PctBeliefExempt
Frequency
0
100
200
300
400
500
0
20
40
60
80
sub_districts[, i]


PctChildPoverty
Frequency
0
20
40
60
80
100
120
0
20
40
60
sub_districts[, i]


PctFamilyPoverty
Frequency
0
50
100
150
200
0
10
20
30
40
50
sub_districts[, i]

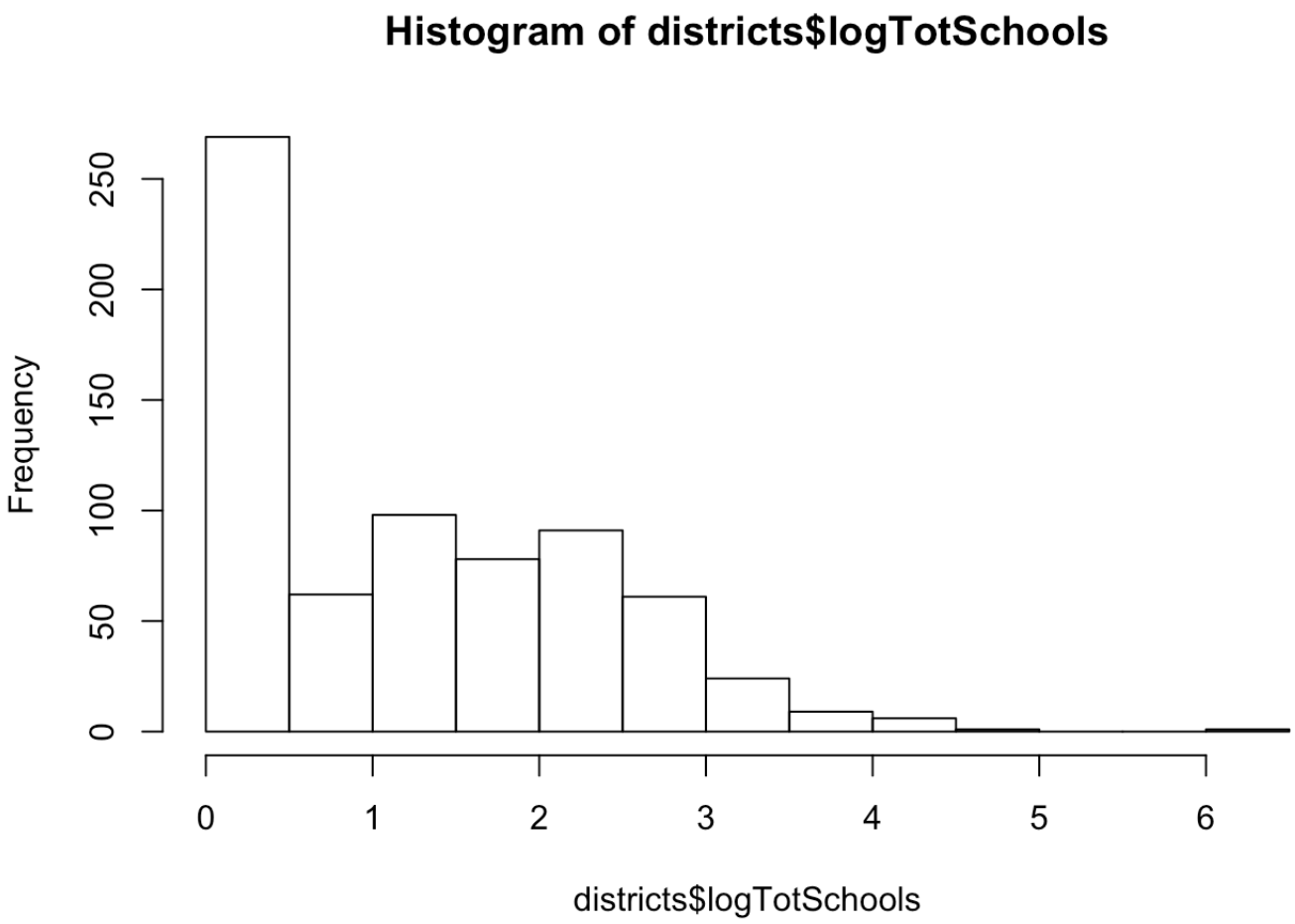

Enrolled
Frequency
0
100
200
300
400
500
600
700
0
10000
20000
30000
40000
50000
sub_districts[, i]



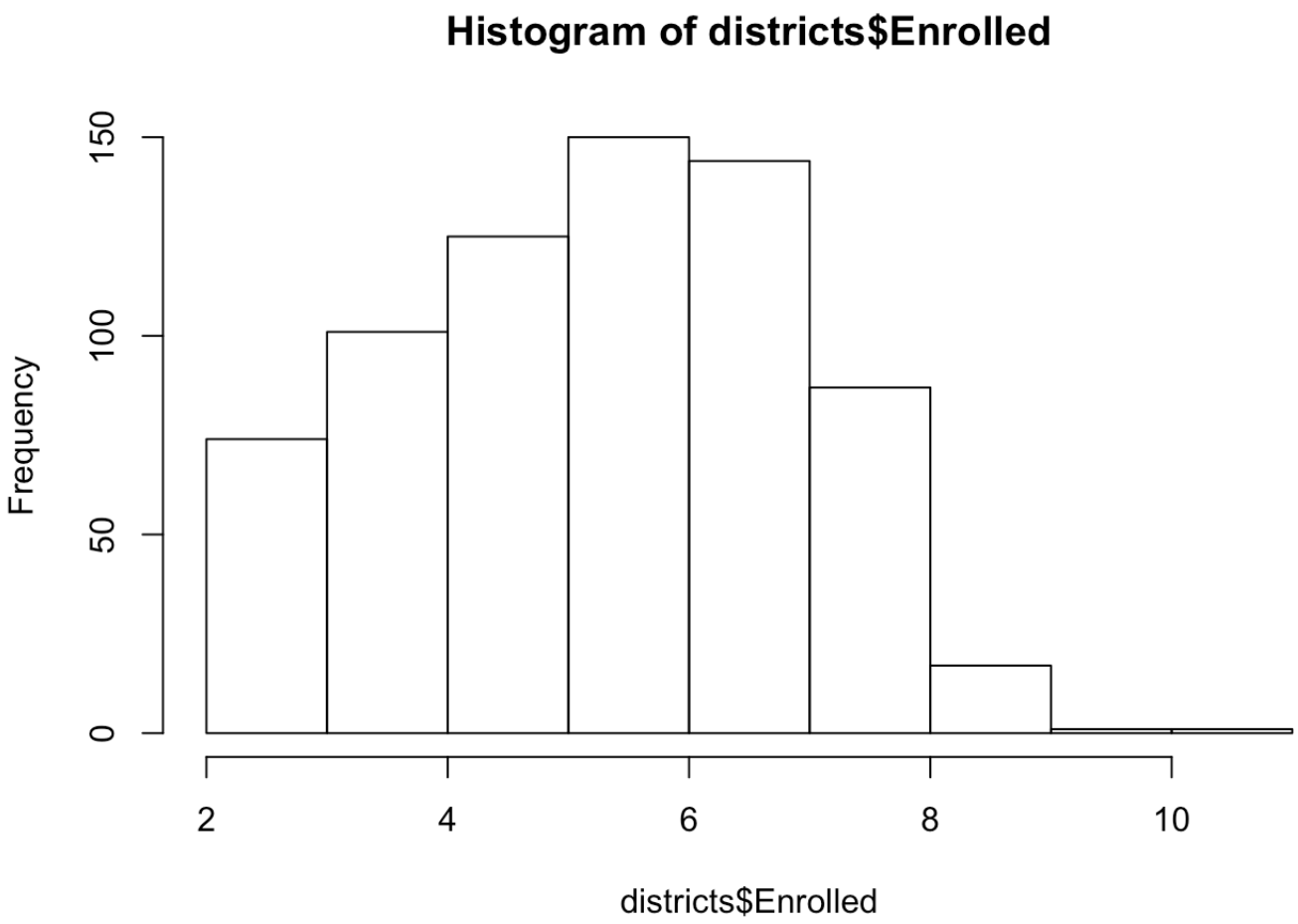
Transforming Variables Prior to Inferential Analysis

Based on the histograms you created above, you may want to make some transformations to your numeric variables to reshape their respective distributions. Remember to add new variables to your dataset, rather than overwriting existing variables.

```
# Here's an example: The total number of schools in each district is generally small
# but there are a few large districts. Try a log() transformation to address this.
districts$logTotSchools <- log(districts$TotalSchools)
hist(districts$logTotSchools)
```




```
# Add any additional transformations that you want to compute and store on your data set.
# Generally speaking, you should inspect the results of each transformation with a histogram
# or some descriptive statistics.
districts$Enrolled <- log(districts$Enrolled )
hist(districts$Enrolled )
```



Inferential Analyses

One very important aspect of doing this final exam is that you need to anticipate what kinds of analytical results will go into your report. Now is a good time to reexamine the exam specification to make sure that you know what analyses you will need in order to complete your report. For example, the following code produces results for a t-test which you may or may not need. Subsequent code blocks create linear regression and logistic regression results. Add any additional t-tests that you think you may need in this block of code.

```
library(BayesFactor)
```

```
## Loading required package: coda
```

```
## Warning: package 'coda' was built under R version 3.6.2
```

```
## Loading required package: Matrix
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard M
orey (richarddmorey@gmail.com).
##
## Type BFManual() to open the manual.
## ************
```

```
# These t-tests compare the mean number of schools per district for those
# districts that completed reporting versus those that did not.
t.test(TotalSchools ~ DistrictComplete, data=districts)
```

```
##
##  Welch Two Sample t-test
##
## data:  TotalSchools by DistrictComplete
## t = 1.6364, df = 42.066, p-value = 0.1092
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.126484 49.101281
## sample estimates:
## mean in group FALSE  mean in group TRUE
##           27.976744            5.989346
```

```
ttestBF(formula=TotalSchools ~ DistrictComplete, data=districts, posterior=FALSE)
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 1957291 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
summary(ttestBF(formula=TotalSchools ~ DistrictComplete, data=districts, posterior=TR
UE, iterations=10000))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                         Mean        SD  Naive SE Time-series SE
## mu                   16.6424    1.8295  0.018295       0.050346
## beta (FALSE - TRUE)  21.2068    3.6795  0.036795       0.100920
## sig2                556.2062   30.0506  0.300506       0.300506
## delta                 0.9002    0.1579  0.001579       0.004235
## g                    40.8326 3231.4270 32.314270      32.314270
##
## 2. Quantiles for each variable:
##
##                         2.5%      25%      50%     75%   97.5%
## mu                   12.9976  15.4393  16.6209  17.863  20.245
## beta (FALSE - TRUE)  14.0270  18.7868  21.1876  23.689  28.431
## sig2                501.0882 535.3892 555.0236 576.236 617.995
## delta                 0.5910   0.7964   0.8998   1.008   1.207
## g                     0.1647   0.4539   0.9347   2.268  27.305
```

```
# Add any additional t-tests that you wish to compute.
aov(TotalSchools ~ DistrictComplete, data=districts)
```

```
## Call:
##    aov(formula = TotalSchools ~ DistrictComplete, data = districts)
##
## Terms:
##                 DistrictComplete Residuals
## Sum of Squares           19511.2  386845.9
## Deg. of Freedom                1       698
##
## Residual standard error: 23.54189
## Estimated effects may be unbalanced
```

The code below produces a linear regression - a one predictor model that predicts the percentage of students who are up to date based on the total student enrollment. This code produces a frequentist result, a Bayes Factor, and a Bayesian estimation result. Add any additional regressions you would like to compute by copying the code and adding new/different predictors.

```
library(BayesFactor)


# These regressions predict the percentage of students who are up to date on vaccines
# using the number of students enrolled in the district.
summary(lm(PctUpToDate ~ Enrolled, data=districts))
```

```
##
## Call:
## lm(formula = PctUpToDate ~ Enrolled, data = districts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.734  -4.175   2.258   7.390  18.627
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  76.3428     1.6088  47.452  < 2e-16 ***
## Enrolled      2.1844     0.2919   7.484 2.18e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.27 on 698 degrees of freedom
## Multiple R-squared:  0.07428,    Adjusted R-squared:  0.07295
## F-statistic: 56.01 on 1 and 698 DF,  p-value: 2.182e-13
```

```
summary(lmBF(PctUpToDate ~ Enrolled, data=districts, posterior=FALSE))
```

```
## Bayes factor analysis
## --------------
## [1] Enrolled : 27078822921 ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
summary(lmBF(PctUpToDate ~ Enrolled, data=districts, posterior=TRUE, iterations=10000
))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean      SD Naive SE Time-series SE
## mu        87.875  0.4646 0.004646       0.004646
## Enrolled   2.156  0.2900 0.002900       0.002942
## sig2     150.986  8.1329 0.081329       0.083485
## g          1.120 16.6537 0.166537       0.166537
##
## 2. Quantiles for each variable:
##
##               2.5%       25%      50%      75%   97.5%
## mu        86.96434  87.56491  87.8751  88.1881  88.786
## Enrolled   1.59054   1.96225   2.1515   2.3509   2.726
## sig2     135.85194 145.36008 150.6652 156.3388 167.935
## g          0.02722   0.07367   0.1468   0.3528   4.428
```

```
# Add any additional regression analyses that you wish to compute
summary(lm(PctUpToDate ~ WithoutDTP+WithoutPolio+WithoutMMR+WithoutHepB, data=districts))
```

```
##
## Call:
## lm(formula = PctUpToDate ~ WithoutDTP + WithoutPolio + WithoutMMR +
##     WithoutHepB, data = districts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.700  -0.392   0.508   1.165  14.488
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  99.14432    0.15532 638.318  < 2e-16 ***
## WithoutDTP   -0.52570    0.06685  -7.864 1.42e-14 ***
## WithoutPolio  0.06483    0.06102   1.062    0.288
## WithoutMMR   -0.76697    0.05271 -14.550  < 2e-16 ***
## WithoutHepB   0.16438    0.02776   5.920 5.05e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.022 on 695 degrees of freedom
## Multiple R-squared:  0.9441, Adjusted R-squared:  0.9438
## F-statistic:  2935 on 4 and 695 DF,  p-value: < 2.2e-16
```

```
summary(lmBF(PctUpToDate ~ WithoutDTP+WithoutPolio+WithoutMMR+WithoutHepB, data=distr
icts, posterior=FALSE))
```

```
## Bayes factor analysis
## --------------
## [1] WithoutDTP + WithoutPolio + WithoutMMR + WithoutHepB : 1.224047e+429 ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
summary(lmBF(PctUpToDate ~WithoutDTP+WithoutPolio+WithoutMMR+WithoutHepB, data=distri
cts, posterior=TRUE, iterations=10000))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean      SD  Naive SE Time-series SE
## mu           87.87089 0.11783 0.0011783      0.0012020
## WithoutDTP   -0.52652 0.07906 0.0007906      0.0008145
## WithoutPolio  0.06612 0.08712 0.0008712      0.0009053
## WithoutMMR   -0.76691 0.05492 0.0005492      0.0005492
## WithoutHepB   0.16439 0.02826 0.0002826      0.0002826
## sig2          9.20022 1.70184 0.0170184      0.0170184
## g             5.65416 7.76935 0.0776935      0.0776935
##
## 2. Quantiles for each variable:
##
##                 2.5%      25%      50%     75%   97.5%
## mu           87.6435 87.79377 87.87087 87.9467 88.0965
## WithoutDTP   -0.6555 -0.57159 -0.52665 -0.4808 -0.3949
## WithoutPolio -0.0522  0.02395  0.06535  0.1065  0.1870
## WithoutMMR   -0.8730 -0.80271 -0.76652 -0.7310 -0.6610
## WithoutHepB   0.1099  0.14556  0.16418  0.1836  0.2192
## sig2          8.2715  8.84069  9.16958  9.5062 10.1851
## g             1.2826  2.55848  3.89703  6.3395 20.3410
```

```
summary(lm(Enrolled ~WithoutDTP+WithoutPolio+WithoutMMR+WithoutHepB+PctBeliefExempt,d
ata=districts))
```

```
##
## Call:
## lm(formula = Enrolled ~ WithoutDTP + WithoutPolio + WithoutMMR +
##     WithoutHepB + PctBeliefExempt, data = districts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3730 -1.0902  0.1622  1.2070  5.5556
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.675550   0.078703  72.114   <2e-16 ***
## WithoutDTP       0.010767   0.033707   0.319    0.749
## WithoutPolio    -0.004616   0.030831  -0.150    0.881
## WithoutMMR      -0.026720   0.027066  -0.987    0.324
## WithoutHepB     -0.020584   0.021212  -0.970    0.332
## PctBeliefExempt -0.005780   0.017418  -0.332    0.740
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.523 on 694 degrees of freedom
## Multiple R-squared:  0.09005,    Adjusted R-squared:  0.08349
## F-statistic: 13.74 on 5 and 694 DF,  p-value: 8.306e-13
```

```
summary(lm(Enrolled ~WithoutDTP+WithoutPolio+WithoutMMR+WithoutHepB+PctUpToDate,data=
districts))
```

```
##
## Call:
## lm(formula = Enrolled ~ WithoutDTP + WithoutPolio + WithoutMMR +
##     WithoutHepB + PctUpToDate, data = districts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3673 -1.0683  0.1569  1.2028  5.5454
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.655506   1.896167   3.510 0.000477 ***
## WithoutDTP    0.005132   0.035142   0.146 0.883927
## WithoutPolio -0.004749   0.030765  -0.154 0.877368
## WithoutMMR   -0.032549   0.030331  -1.073 0.283598
## WithoutHepB  -0.024255   0.014335  -1.692 0.091102 .
## PctUpToDate  -0.009856   0.019109  -0.516 0.606172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.523 on 694 degrees of freedom
## Multiple R-squared:  0.09025,    Adjusted R-squared:  0.0837
## F-statistic: 13.77 on 5 and 694 DF,  p-value: 7.709e-13
```

```
summary(lm(PctUpToDate ~ PctChildPoverty*Enrolled, data=districts))
```

```
##
## Call:
## lm(formula = PctUpToDate ~ PctChildPoverty * Enrolled, data = districts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.483  -3.529   2.259   6.756  24.315
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              65.29264    3.49650  18.674  < 2e-16 ***
## PctChildPoverty           0.45332    0.13398   3.383 0.000756 ***
## Enrolled                  3.26482    0.64925   5.029 6.29e-07 ***
## PctChildPoverty:Enrolled -0.04080    0.02557  -1.596 0.110972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.91 on 696 degrees of freedom
## Multiple R-squared:  0.1308, Adjusted R-squared:  0.127
## F-statistic:  34.9 on 3 and 696 DF,  p-value: < 2.2e-16
```

```
summary(lmBF(PctUpToDate ~ PctChildPoverty*Enrolled, data=districts, posterior=FALSE)
)
```

```
## Bayes factor analysis
## --------------
## [1] PctChildPoverty * Enrolled : 7.56551e+17 ±0.01%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
summary(lmBF(PctUpToDate ~ PctChildPoverty*Enrolled, data=districts, posterior=TRUE,
iterations=10000))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                               Mean      SD  Naive SE Time-series SE
## mu                         87.87174 0.45002 0.0045002      0.0043540
## PctChildPoverty             0.44464 0.13216 0.0013216      0.0013216
## Enrolled                    3.20603 0.64134 0.0064134      0.0061874
## PctChildPoverty.&.Enrolled -0.04022 0.02523 0.0002523      0.0002523
## sig2                      141.99399 7.61845 0.0761845      0.0761845
## g                           0.13578 0.25028 0.0025028      0.0025417
##
## 2. Quantiles for each variable:
##
##                               2.5%       25%       50%       75%     97.5%
## mu                         86.99592  87.56730  87.86752  88.17362 8.876e+01
## PctChildPoverty             0.19165   0.35521   0.44463   0.53470 7.012e-01
## Enrolled                    1.98052   2.77574   3.20693   3.63751 4.450e+00
## PctChildPoverty.&.Enrolled -0.08995  -0.05721  -0.04034  -0.02313 7.917e-03
## sig2                      127.66372 136.71710 141.69596 147.07114 1.575e+02
## g                           0.02425   0.05037   0.08175   0.14058 5.524e-01
```

Finally, the code below produces a logistic regression - a one predictor model that predicts the percentage of students who are up to date based on the total student enrollment. The code produces a frequentist result as well as a Bayesian estimation result. Add any additional regressions you would like to compute by copying the code and adding new/different predictors.

```
library(MCMCpack)
```

```
## Warning: package 'MCMCpack' was built under R version 3.6.2
```

```
## Loading required package: MASS
```

```
## Warning: package 'MASS' was built under R version 3.6.2
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2020 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
# These logistic regressions predict whether a district's reporting is complete
# based on the percentage of students who did not get the DTP vaccine.
glmOut <- glm(DistrictComplete ~ WithoutDTP, family = binomial(), data=districts)
summary(glmOut)
```

```
##
## Call:
## glm(formula = DistrictComplete ~ WithoutDTP, family = binomial(),
##     data = districts)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4471   0.3265   0.3388   0.3614   0.6407
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.94279    0.21469  13.707   <2e-16 ***
## WithoutDTP  -0.01901    0.01149  -1.653   0.0982 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 323.23  on 699  degrees of freedom
## Residual deviance: 320.82  on 698  degrees of freedom
## AIC: 324.82
##
## Number of Fisher Scoring iterations: 5
```

```
exp(coef(glmOut))
```

```
## (Intercept)  WithoutDTP
##  18.9686586   0.9811736
```

```
exp(confint(glmOut))
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %    97.5 %
## (Intercept) 12.6417056 29.400282
## WithoutDTP   0.9605806  1.005434
```

```
glmBayesOut <- MCMCpack::MCMClogit(DistrictComplete ~ WithoutDTP, data=districts)
summary(glmBayesOut)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                 Mean      SD  Naive SE Time-series SE
## (Intercept)  2.94696 0.21699 0.0021699      0.0065483
## WithoutDTP  -0.01759 0.01167 0.0001167      0.0003545
##
## 2. Quantiles for each variable:
##
##                 2.5%      25%      50%       75%    97.5%
## (Intercept)  2.53482  2.79440  2.94486  3.098031 3.373807
## WithoutDTP  -0.03906 -0.02577 -0.01785 -0.009692 0.006221
```

```
summary(exp(glmBayesOut))
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean      SD  Naive SE Time-series SE
## (Intercept) 19.5040 4.31329 0.0431329      0.1316858
## WithoutDTP   0.9826 0.01148 0.0001148      0.0003489
##
## 2. Quantiles for each variable:
##
##                2.5%     25%     50%     75%  97.5%
## (Intercept) 12.6142 16.3528 19.0081 22.1543 29.189
## WithoutDTP   0.9617  0.9746  0.9823  0.9904  1.006
```

```
# Add any additional logistic regression analyses that you wish to compute

glmOut <- glm(DistrictComplete ~ WithoutDTP+WithoutPolio+WithoutMMR+WithoutHepB, fami
ly = binomial(), data=districts)
summary(glmOut)
```

```
##
## Call:
## glm(formula = DistrictComplete ~ WithoutDTP + WithoutPolio +
##     WithoutMMR + WithoutHepB, family = binomial(), data = districts)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4677   0.3189   0.3308   0.3523   0.7817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.96076    0.21896  13.522   <2e-16 ***
## WithoutDTP   -0.10621    0.07775  -1.366    0.172
## WithoutPolio -0.02086    0.06588  -0.317    0.751
## WithoutMMR    0.07619    0.06244   1.220    0.222
## WithoutHepB   0.04323    0.02865   1.509    0.131
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 323.23  on 699  degrees of freedom
## Residual deviance: 316.61  on 695  degrees of freedom
## AIC: 326.61
##
## Number of Fisher Scoring iterations: 5
```

```
exp(coef(glmOut))
```

```
##  (Intercept)   WithoutDTP WithoutPolio   WithoutMMR  WithoutHepB
##   19.3125768    0.8992346    0.9793517    1.0791717    1.0441791
```

```
exp(confint(glmOut))
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %    97.5 %
## (Intercept)  12.7711963 30.205777
## WithoutDTP    0.7773616  1.056891
## WithoutPolio  0.8467484  1.102984
## WithoutMMR    0.9516729  1.216971
## WithoutHepB   0.9855471  1.108083
```

```
glmBayesOut <- MCMCpack::MCMClogit(DistrictComplete ~ WithoutDTP, data=districts)
summary(glmBayesOut)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean      SD  Naive SE Time-series SE
## (Intercept)  2.94696 0.21699 0.0021699      0.0065483
## WithoutDTP  -0.01759 0.01167 0.0001167      0.0003545
##
## 2. Quantiles for each variable:
##
##                2.5%      25%      50%       75%    97.5%
## (Intercept)  2.53482  2.79440  2.94486  3.098031 3.373807
## WithoutDTP  -0.03906 -0.02577 -0.01785 -0.009692 0.006221
```

```
summary(exp(glmBayesOut))
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean      SD  Naive SE Time-series SE
## (Intercept) 19.5040 4.31329 0.0431329      0.1316858
## WithoutDTP   0.9826 0.01148 0.0001148      0.0003489
##
## 2. Quantiles for each variable:
##
##                2.5%     25%     50%     75%  97.5%
## (Intercept) 12.6142 16.3528 19.0081 22.1543 29.189
## WithoutDTP   0.9617  0.9746  0.9823  0.9904  1.006
```

```
glmOut <- glm(DistrictComplete ~ PctUpToDate, family = binomial(), data=districts)
summary(glmOut)
```

```
##
## Call:
## glm(formula = DistrictComplete ~ PctUpToDate, family = binomial(),
##     data = districts)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4528   0.3263   0.3374   0.3606   0.5956
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.24594    0.87481   1.424   0.1544
## PctUpToDate  0.01711    0.01012   1.691   0.0909 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 323.23  on 699  degrees of freedom
## Residual deviance: 320.70  on 698  degrees of freedom
## AIC: 324.7
##
## Number of Fisher Scoring iterations: 5
```

```
exp(coef(glmOut))
```

```
## (Intercept) PctUpToDate
##    3.476198    1.017262
```

```
exp(confint(glmOut))
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.7100548 22.893297
## PctUpToDate 0.9956883  1.036475
```

```
glmBayesOut <- MCMCpack::MCMClogit(DistrictComplete ~ PctUpToDate, data=districts)
summary(glmBayesOut)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean      SD  Naive SE Time-series SE
## (Intercept) 1.36428 0.87856 0.0087856      0.0263862
## PctUpToDate 0.01605 0.01013 0.0001013      0.0003027
##
## 2. Quantiles for each variable:
##
##                  2.5%      25%     50%     75%   97.5%
## (Intercept) -0.260161 0.745377 1.34576 1.94538 3.17326
## PctUpToDate -0.004743 0.009407 0.01607 0.02309 0.03497
```

```
summary(exp(glmBayesOut))
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD  Naive SE Time-series SE
## (Intercept) 5.869 6.94228 0.0694228      0.1949122
## PctUpToDate 1.016 0.01029 0.0001029      0.0003074
##
## 2. Quantiles for each variable:
##
##               2.5%   25%   50%   75%  97.5%
## (Intercept) 0.7709 2.107 3.841 6.996 23.885
## PctUpToDate 0.9953 1.009 1.016 1.023  1.036
```

Knit, PDF, and Submit to Blackboard

Click the **Knit** button to create and inspect your html document. If the knitting process is successful an html file will be submitted to your current working directory. The file will also pop up in a viewer. There is a button on the viewer that says “Open in Browser.” Click this and then save a PDF version of your page from your browser. Submit the PDF version to Blackboard (Week 13 Content).