MA 575 HW 7

Benjamin Draves

November 7

Exercise 7.1

(a)

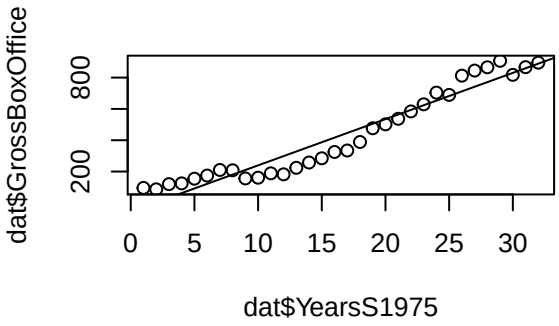
```
#read in data
dat = read.table("~/Desktop/Courses/MA 575/book_data/boxoffice.txt", header = TRUE)

#take a look
str(dat)
```

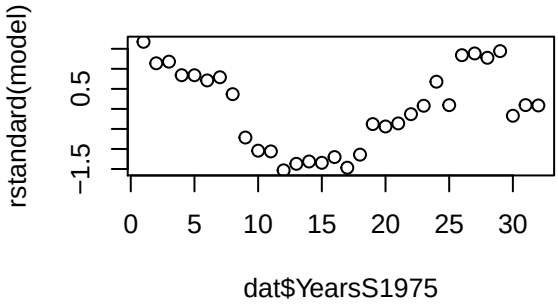
```
## 'data.frame':    32 obs. of  2 variables:
##  $ GrossBoxOffice: num  95.3 86.4 119.4 124.4 154.2 ...
##  $ year          : int  1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 ...
```

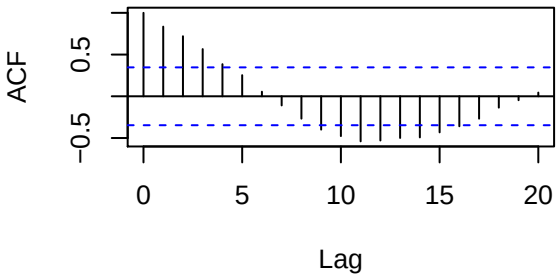
```
#Make YearsS1975
dat$YearsS1975 = dat$year - 1975

#take a loot
str(dat)
```

```
## 'data.frame':    32 obs. of  3 variables:
##  $ GrossBoxOffice: num  95.3 86.4 119.4 124.4 154.2 ...
##  $ year          : int  1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 ...
##  $ YearsS1975    : num  1 2 3 4 5 6 7 8 9 10 ...
```

```
#construct model 9.6
model = lm(GrossBoxOffice ~ YearsS1975, data = dat)

par(mfrow = c(2,2))
plot(dat$YearsS1975, dat$GrossBoxOffice)
abline(model)

plot(dat$YearsS1975, rstandard(model))

acf(model$residuals, lag.max = 20, main = "")
```





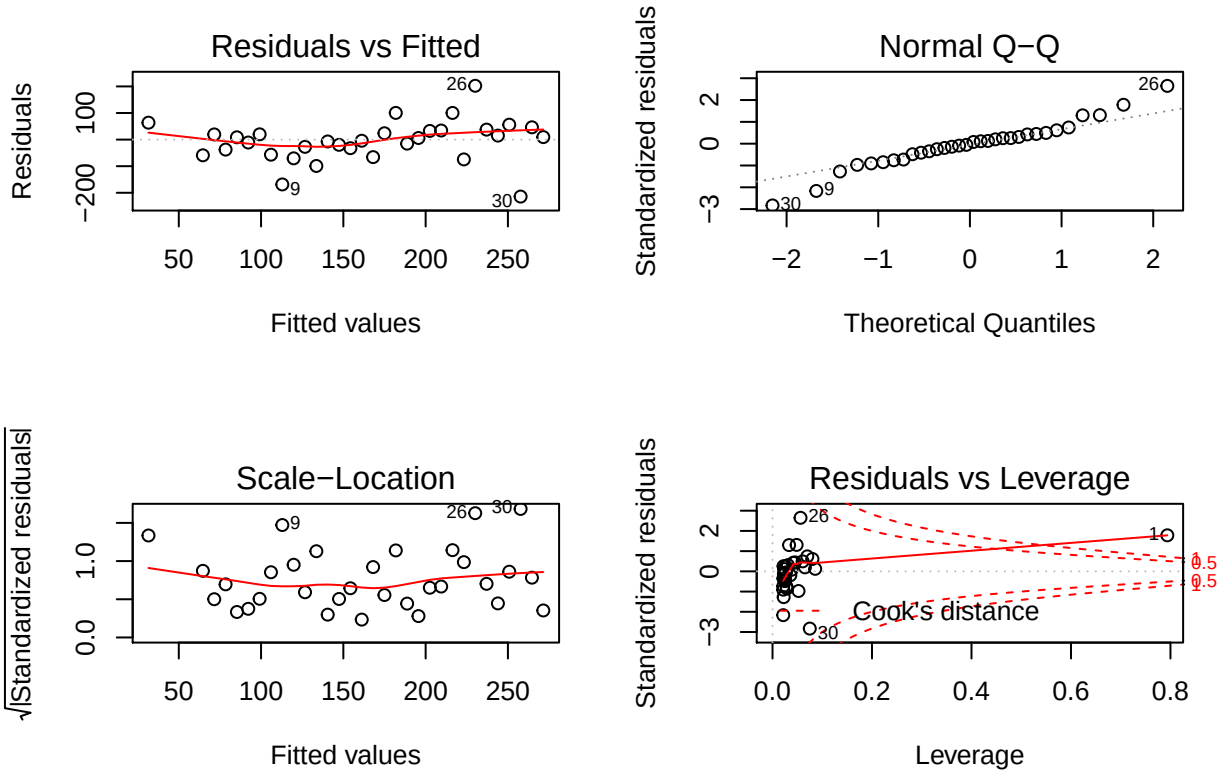



```
library(nlme)
#build AR 1 model with Maximum Likelihood Coefficents
modelgls = gls(GrossBoxOffice ~YearsS1975, data = dat, method = "ML", correlation = corAR1())

#reconstruct xstar model
g <- lm(GrossBoxOffice~YearsS1975,data=dat)
rho <- 0.8782065
x <- model.matrix(g)
Sigma <- diag(length(dat$YearsS1975))
Sigma <- rho^abs(row(Sigma)-col(Sigma))
sm <- chol(Sigma)
smi <- solve(t(sm))
xstar <- smi %*% x
ystar <- smi %*% dat$GrossBoxOffice
m1tls <- lm(ystar ~ xstar-1)
summary(m1tls)
```

```
##
## Call:
## lm(formula = ystar ~ xstar - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -214.235  -42.370    0.902   33.011  202.415
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## xstar(Intercept)     4.514     72.744   0.062    0.951
## xstarYearsS1975     27.075      3.448   7.853 9.17e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.66 on 30 degrees of freedom
## Multiple R-squared:  0.8427, Adjusted R-squared:  0.8322
## F-statistic: 80.37 on 2 and 30 DF,  p-value: 8.919e-13
```

```
#take a look at diagnostics for this model
par(mfrow = c(2,2))
plot(m1tls)
```



(a)

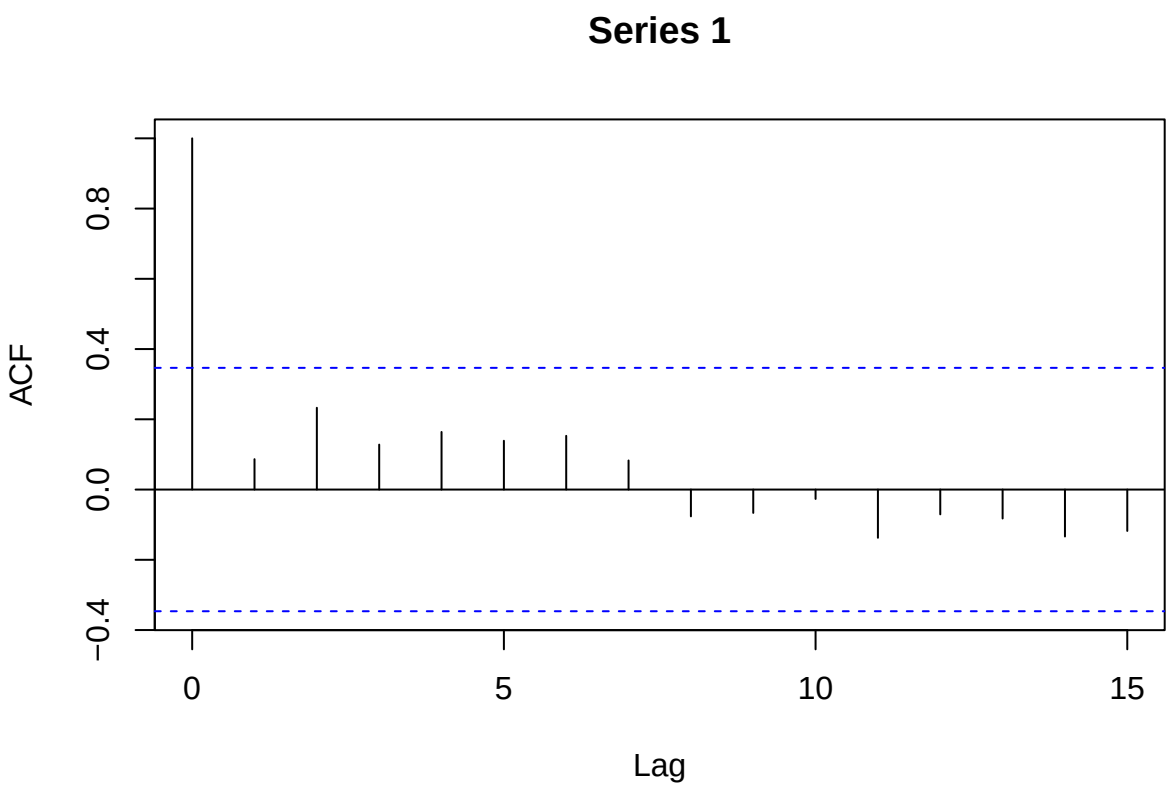
The staffer correctly notes that an OLS fit to this data is insufficient. There is very clearly extra structure in the error they are failing to address. The residuals are positively correlated as evident by the roller coaster patter of the standardized residuals the ACF plot.

The staffer decides to fit an AR(1) error structure to the data to address this correlative behavior. Seeing that the ACF is monotone decreasing, AR(1) is probably a good initial error structure to impose. After transformation the variables to have uncorrelated errors, we can use the GLS framework to analyze the model. The staffer fails to do this and thus is concerned with the effect that autocorrelation has on their residual plots even though we know residuals can have non random error even if the correct function is fit. Here we see that the GLS fit has some normality issues with

one very large outlier corresponding to the initial point. This was expected as the first point is always expected to have high leverage. #### (b)

We look to improve the model by transforming the response variable.

```
#model data
mdat = data.frame(YearsS1975 = dat$YearsS1975,lGrossBoxOffice = log(dat$GrossBoxOffice))

#build AR 1 model with Maximum Likelihood Coefficents
modelgls = gls(lGrossBoxOffice ~ YearsS1975, data = mdat, method = "ML", correlation = corAR1()

#construct xstar model
g <- lm(lGrossBoxOffice~YearsS1975,data=mdat)
rho <- 0.7010131
x <- model.matrix(g)
Sigma <- diag(length(mdat$YearsS1975))
Sigma <- rho^abs(row(Sigma)-col(Sigma))
sm <- chol(Sigma)
smi <- solve(t(sm))
xstar <- smi %*% x
ystar <- smi %*% mdat$lGrossBoxOffice

#take a look at acf of ystar
acf(ystar)
```

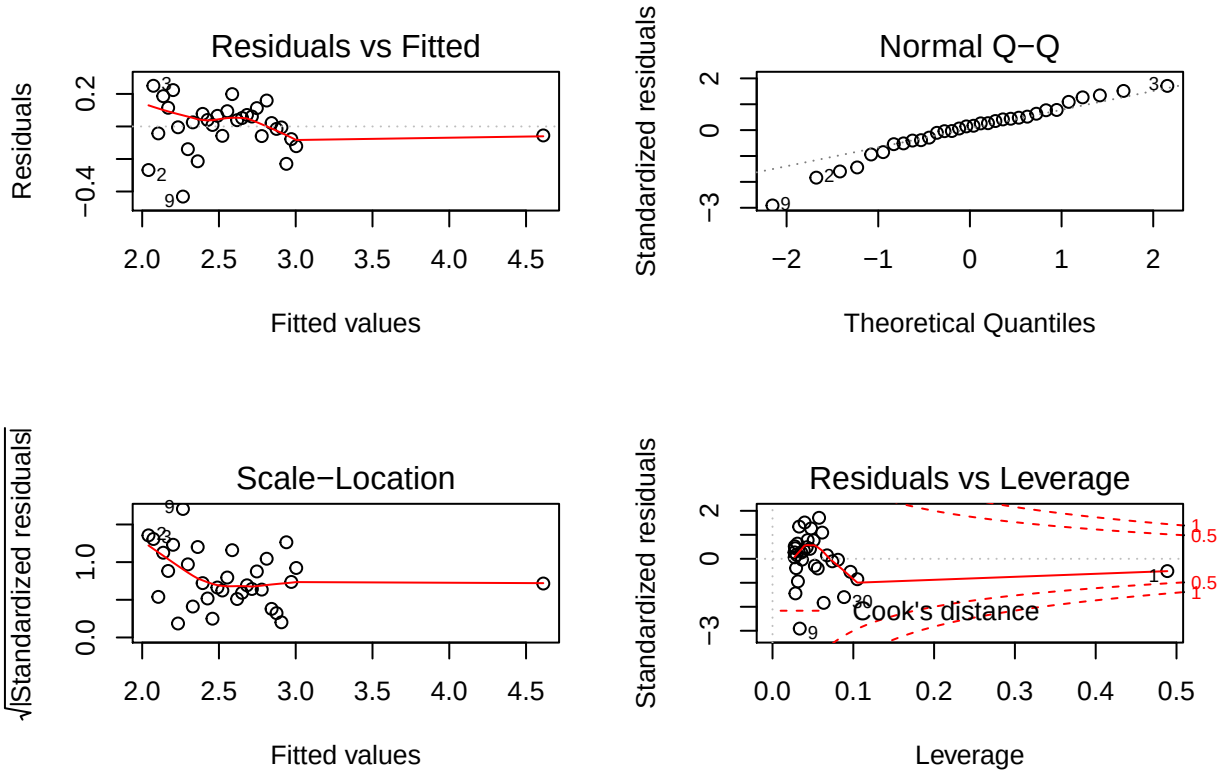



```
#build new model
df = data.frame(Y = ystar[,1],X = xstar[,2], Intercept = xstar[,1])
```

```r
m2tls <- lm(Y ~ X + Intercept -1, data = df)
summary(m2tls)
```

```
##
## Call:
## lm(formula = Y ~ X + Intercept - 1, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43115 -0.05804  0.02298  0.08148  0.24998
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## X         0.076506   0.005615   13.62  2.2e-14 ***
## Intercept 4.535797   0.110031   41.22  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1507 on 30 degrees of freedom
## Multiple R-squared:  0.9969, Adjusted R-squared:  0.9967
## F-statistic:  4866 on 2 and 30 DF,  p-value: < 2.2e-16
```

```r
#take a look at diagnostics for this model
par(mfrow = c(2,2))
plot(m2tls)
```

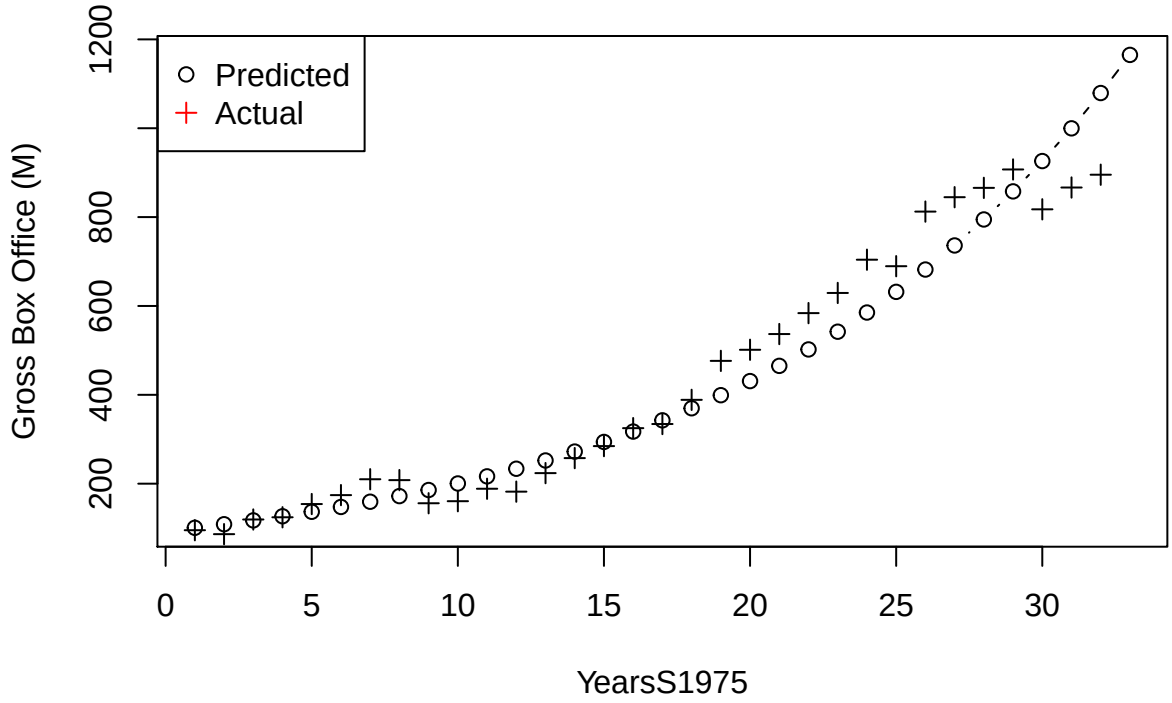


Here again we see we have some issues with the normality asumption and have one large good leverage point refering the first point. The residuals even have some heteroskedastic issues.

(c)

Since $\beta^*_{LS} = \beta_{GLS}$ we can use *modelgls* for our pointwise predection.

```
predx = data.frame(YearsS1975 = 33)
pred = exp(predict(modelgls, predx)[1])


plot(c(dat$YearsS1975,33), c(exp(modelgls$fitted), pred), type = "b", xlab = "YearsS1975", ylab
points(dat$YearsS1975, dat$GrossBoxOffice, pch = 3)

legend("topleft", col = c("black","red"), legend = c("Predicted","Actual"), pch = c(1,3))
```



Our predicted value is given by 1165.035 million which appears very high for this situation. But given the cyclic behavior may be approriate.

(d)

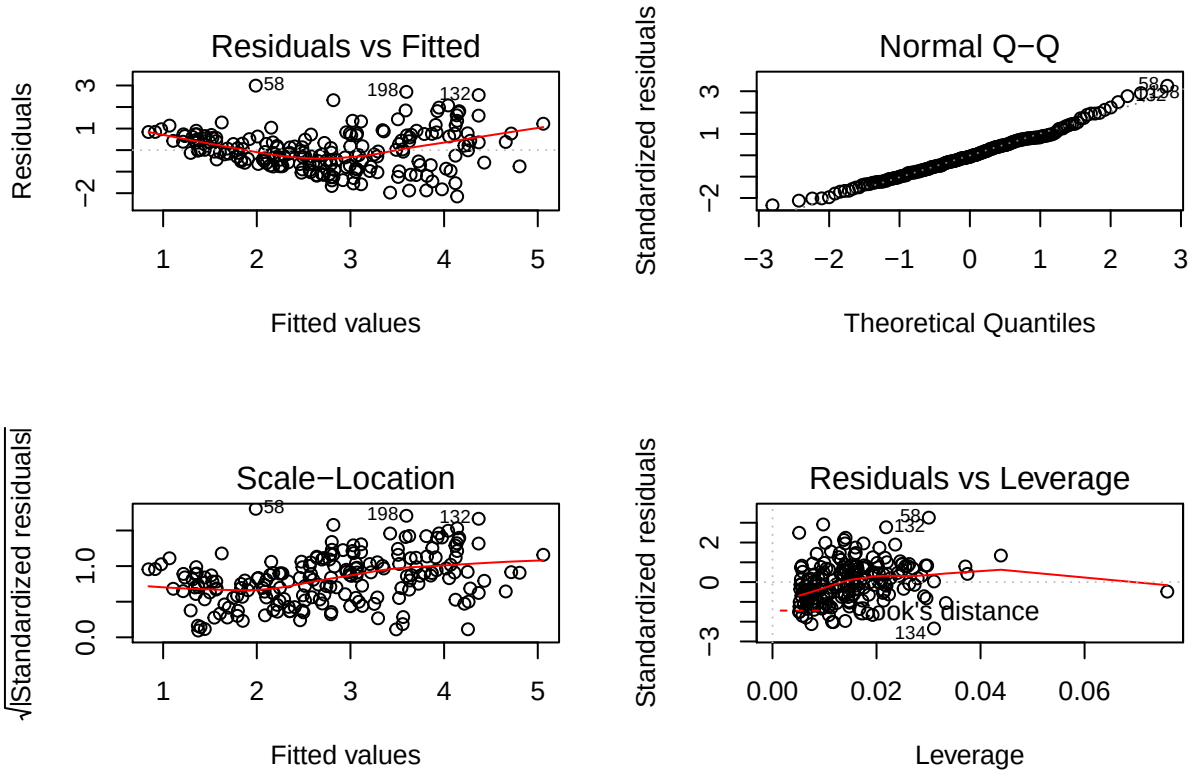
After correcting for the independence of the residuals, the does not appear to be any serious outlier. In fact the year 2,000 does not appear to be noticable at all.

Exercise 7.2

```r
#read in data
un = read.csv("~/Desktop/Courses/MA 575/book_data/UN11.csv")

#take a look
str(un)
```

```
## 'data.frame':    199 obs. of  7 variables:
##  $ X        : Factor w/ 199 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ region   : Factor w/ 8 levels "Africa","Asia",..: 2 4 1 1 3 5 2 3 8 4 ...
##  $ group    : Factor w/ 3 levels "africa","oecd",..: 3 3 1 1 3 3 3 3 2 2 ...
##  $ fertility: num  5.97 1.52 2.14 5.13 2 ...
##  $ ppgdp    : num  499 3677 4473 4322 13750 ...
##  $ lifeExpF : num  49.5 80.4 75 53.2 81.1 ...
##  $ pctUrban : int  23 53 67 59 100 93 64 47 89 68 ...
```

```r
#create model matrix
mun = matrix(ncol = 3, nrow = nrow(un))
mun[,1] = log(un$ppgdp)
mun[,2] = un$pctUrban
mun[,3] = un$fertility
colnames(mun) = c("logppgdp", "pctUrban", "fertility")
mun = data.frame(mun)

#build model
model = lm(fertility~ logppgdp + pctUrban, data = mun)

#plot diagnostics
par(mfrow = c(2,2))
plot(model)
```



We notice some quadratic behavior in the residuals and some newness from the normal Q-Q plot. This suggests that the residuals are not strictly normal and our confidence intervals may not be entirely trustworthy.

```
#build confidence intervals for beta coff.
confint(model)
```

```
##                   2.5 %       97.5 %
## (Intercept)  7.205721498  8.780818268
## logppgdp    -0.741668070 -0.488616865
## pctUrban    -0.008851621  0.007973063
```

(a)

We will use the bootstrap algorithm to build empiracle confidence intervals for these parameters.

```
#Step 1a: Get residuals from non normal model
e = model$residuals

#Step 1b: Set up data structures
n = length(e)
B = 100
boot_coef = matrix(nrow = B, ncol = 3)
colnames(boot_coef) = c("B0", "B1", "B2")

for(b in 1:B){
  #Step 2: Sample with replacement from e
```

```
  estar = sample(e,n, replace = TRUE)

  #Step 3a: Create a boostrap response
  ystar = as.vector(predict(model)) + estar

  #Step 3b: Run regression of boostrap regression on X
  bmodel = lm(ystar ~ mun$logppgdp + mun$pctUrban)

  #Step 3c: Store b* estimates of the coefficents
  boot_coef[b, ] = as.vector(coef(bmodel))
}


#Step 4: Create emprical confidence intervals
quantile(boot_coef[,1], c(.025, .975))
```

```
##     2.5%    97.5%
## 7.333682 8.957156
```

```
quantile(boot_coef[,2], c(.025, .975))
```

```
##       2.5%      97.5%
## -0.7531900 -0.5172589
```

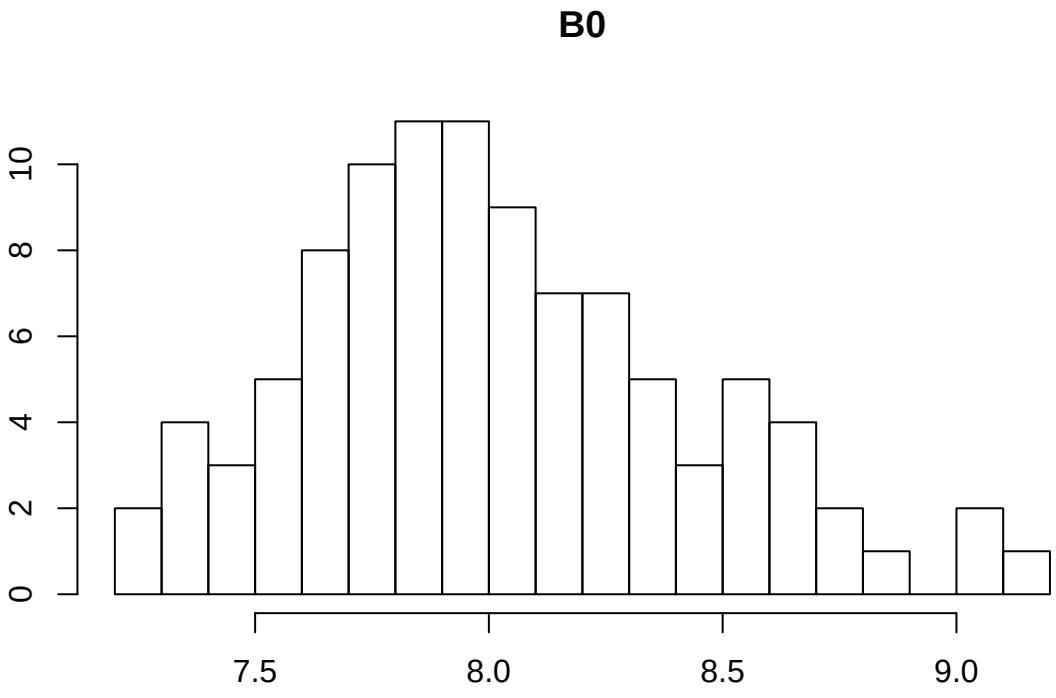
```
quantile(boot_coef[,3], c(.025, .975))
```

```
##         2.5%        97.5%
## -0.007775436  0.007866505
```

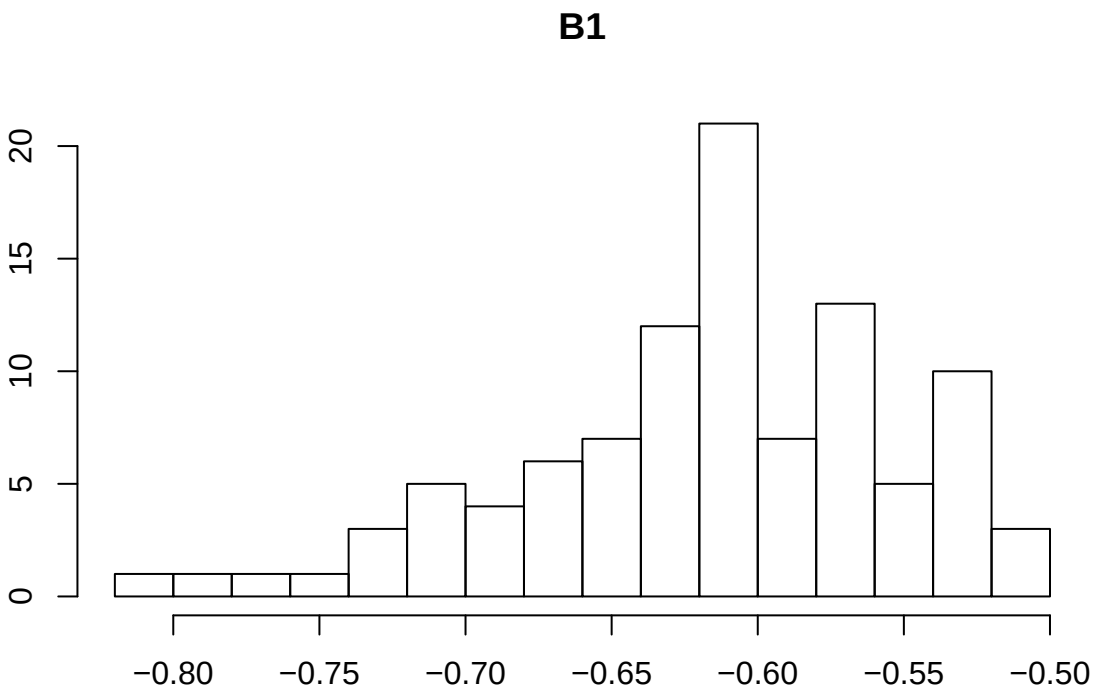
For the intercept, it appears that the empirical CI is shifted slightly to the right and has decreased in length. For logppgdp the left end of the interval has stayed the same while the right side of the interval has shifted to the left. This again corresponds to a smaller CI. Lastly, for pctUrban, the empirical CI has shifted in on both sides of the interval. In all cases, we see our CI decrease in length. If correctly constructed this corresponds to higher power than the normal theory CI.

(b)

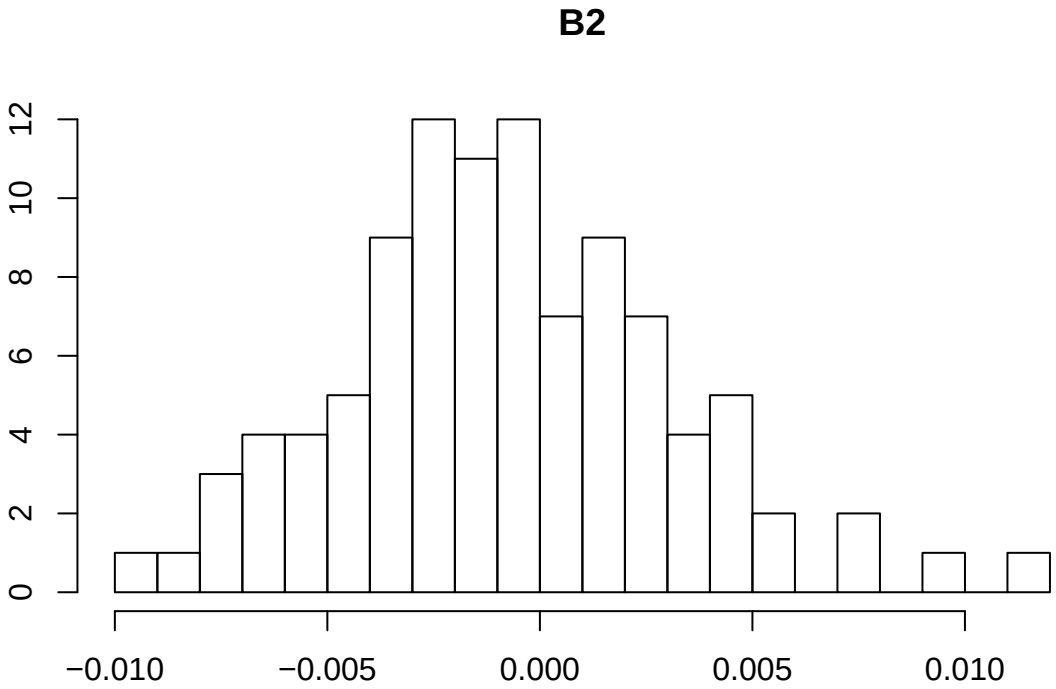
```
#Build histrograms of our empircal sampling distribution

hist(boot_coef[,1], main = "B0", xlab ="", ylab = "", breaks = 15)
```




```
hist(boot_coef[,2], main = "B1", xlab ="", ylab = "", breaks = 15)
```




```
hist(boot_coef[,3], main = "B2", xlab ="", ylab = "", breaks = 15)
```



The β_0 histogram is skewed left with mean clearly not zero. This suggests that $\beta_0 \neq 0$ but is not normally distributed. Thus the CI and test statistics would need more rarely examination. The β_1 histogram is skewed right and highly nonnormal with nonzero mean. We would not be justified in using classical normal theory to analyze this variable. The β_2 plot differs however. The plot appears normal (possible left skew) with mean very close to zero. This could justify using classical normal theory.

Exercise 7.3

Consider the two regression models given below

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{Z}\alpha_{LS} + \mathbf{e}$$

$$\hat{\mathbf{e}}_{Y\sim X} = \hat{\mathbf{e}}_{Z\sim X}\alpha_{AVP} + e^*$$

Our goal is to show $\hat{\alpha}_{LS} = \hat{\alpha}_{AVP}$. We start with $\hat{\alpha}_{AVP}$.

$$\begin{aligned}
\hat{\alpha}_{AVP} &= ((\hat{\mathbf{e}}_{Z\sim X})^T\hat{\mathbf{e}}_{Z\sim X})^{-1}(\hat{\mathbf{e}}_{Z\sim X})^T\hat{\mathbf{e}}_{Y\sim X} \\
&= \left[((I-H_X)Z)^T(I-H_X)Z\right]^{-1}((I-H_X)Z)^T(I-H_X)Y \\
&= \left[Z^T(I-H_X)^T(I-H_X)Z\right]^{-1}Z^T(I-H_X)^T(I-H_X)Y \\
&= \left[Z^T(I-H_X)Z\right]^{-1}Z^T(I-H_X)Y
\end{aligned}$$

Now we consider the $\hat{\alpha}_{LS}$. First we rewrite the first mean function as

$$\mathbf{Y} = \mathbf{W}\Gamma + e$$

where $\mathbf{W} = [\mathbf{X}|\mathbf{Z}]$ and $\Gamma = \begin{bmatrix}\beta \\ \alpha\end{bmatrix}$. The OLS estimates of Γ is given by $\hat{\Gamma} = (W^TW)^{-1}W^TY$. Note that we only have interest in the *last* element of $\hat{\Gamma}$. Thus if any calculation will not aid us in finding this value, we will not include it here. First note that

$$W^TY = \begin{bmatrix}X^TY \\ Z^TY\end{bmatrix}$$

and

$$W^TW = \begin{bmatrix}X^TX & X^TZ \\ Z^TZ & Z^TZ\end{bmatrix}$$

Notice that $\hat{\alpha}_{AVP}$ will only rely on the last row of the inverse of W^TW. Since W^TW is partitioned we can use the same result from quiz 5 to find the last row of the inverse. Using this we see

$$(W^TW)^{-1} = \begin{bmatrix} & \\ -(Z^TZ - Z^TX(X^TX)^{-1}X^TZ)^{-1}Z^TX(X^TX)^{-1} & (Z^TZ - Z^TX(X^TX)^{-1}X^TZ)^{-1}\end{bmatrix}$$

Notice that because Z^T is $1 \times n$ and $(X^TX)^{-1}$ is $(p+1) \times (p+1)$, the left most element is $1 \times (p+1)$. Moreover since Z^T is $1 \times n$ and Z is $n \times 1$ the right most element is 1×1 or simply a scalar. Together, these two expressions are $1 \times (p+2)$ or exactly the last row of $(W^TW)^{-1}$. Now applying this to W^TY we have

$$(W^TW)^{-1}W^TY = \begin{bmatrix} \\ -(Z^TZ - Z^TX(X^TX)^{-1}X^TZ)^{-1}Z^TX(X^TX)^{-1}X^TY + (Z^TZ - Z^TX(X^TX)^{-1}X^TZ)^{-1}Z^TY\end{bmatrix}$$

Focusing on the last row will yield our result

$$\begin{aligned}
\hat{\alpha}_{AVP} &= -(Z^TZ - Z^TX(X^TX)^{-1}X^TZ)^{-1}Z^TX(X^TX)^{-1}X^TY + (Z^TZ - Z^TX(X^TX)^{-1}X^TZ)^{-1}Z^TY \\
&= -(Z^TZ - Z^TH_XZ)^{-1}Z^TH_XY + (Z^TZ - Z^TH_XZ)^{-1}Z^TY \\
&= -(Z^T(Z - H_XZ))^{-1}Z^TH_XY + (Z^T(Z - H_XZ))^{-1}Z^TY \\
&= (Z^T(Z - H_XZ))^{-1}\Big[Z^TY - Z^TH_XY\Big] \\
&= \Big[Z^T(I - H_X)Z\Big]^{-1}Z^T\Big[I - H_X\Big]Y
\end{aligned}$$

Therefore, we see that $\hat{\alpha}_{AVP} = \hat{\alpha}_{LS}$.