d10

Excellent work!
Grade: 98

Matthew Draper

March 3, 2019

to change: output: pdf_document: keep_tex: yes

```
setwd("C:/Users/mwdra/Google Drive/UCSD/Courses/204B Quantitative Methods/Homework 8")
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 3.5.2
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Ta
bles.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
require(knitr)
```

```
## Loading required package: knitr
```

```
## Warning: package 'knitr' was built under R version 3.5.2
```

```
require(markdown)
```

```
## Loading required package: markdown
```

```
## Warning: package 'markdown' was built under R version 3.5.2
```

```
require(dotwhisker)
```

```
## Loading required package: dotwhisker
```

```
## Warning: package 'dotwhisker' was built under R version 3.5.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
set.seed(030419)
```

Question 1.

We will begin by reviewing last week's regression model.

```
## Reading in data, defining variables of interest:
Tabellini<-read.csv("Tabellini.csv", header=T)
cpi <- Tabellini$CPI
elf <- Tabellini$ELF
pop <- Tabellini$POP
luk <- Tabellini$LEGOR_UK
edu <- Tabellini$EDU
legor_fr <- Tabellini$LEGOR_FR
y <- Tabellini$Y

## H0: Population, education levels, ethnolinguistic fractionalization, GDP and French l
egal origins do not coorrelate with CPI.

reg10 <- lm(cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))
summary(reg10)
```

```
##
## Call:
## lm(formula = cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86176 -0.60012 -0.05559  0.72937  2.28870
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.768e+00  7.955e-01  11.022  < 2e-16 ***
## edu         -2.726e-02  8.232e-03  -3.311 0.001423 **
## legor_fr     8.706e-01  2.255e-01   3.860 0.000236 ***
## y           -2.075e-04  1.676e-05 -12.379  < 2e-16 ***
## elf          4.130e-01  5.339e-01   0.774 0.441594
## pop          1.310e-02  3.433e-03   3.816 0.000274 ***
## elf:pop     -1.748e-02  5.245e-03  -3.333 0.001329 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9865 on 76 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.8456, Adjusted R-squared:  0.8334
## F-statistic: 69.36 on 6 and 76 DF,  p-value: < 2.2e-16
```

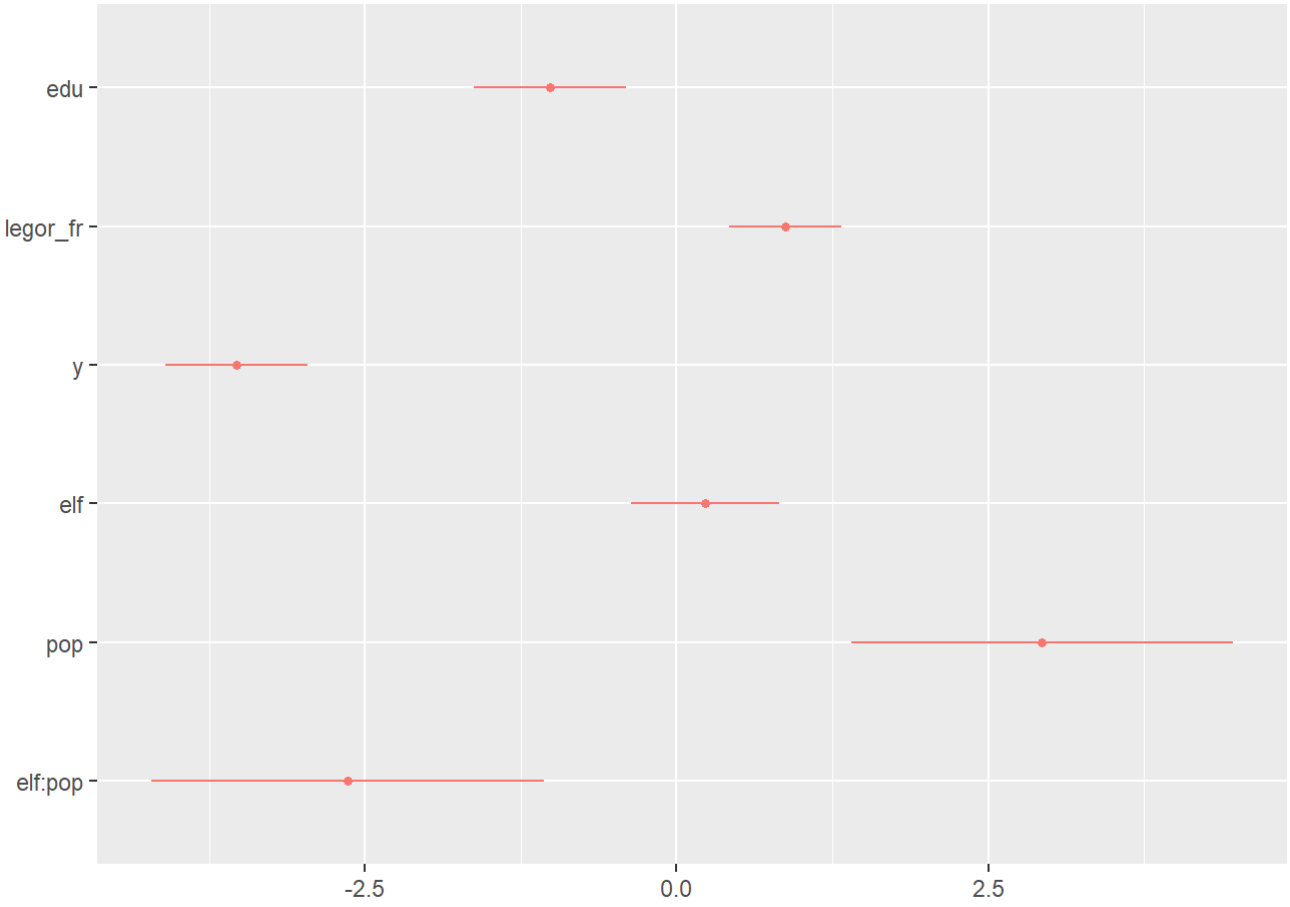
```
stargazer::stargazer(reg10, type = "html",
 title = "Reg10")
```

Reg10

	Dependent variable:
	cpi
edu	-0.027***
	(0.008)
legor_fr	0.871***
	(0.226)
y	-0.0002***
	(0.00002)
elf	0.413
	(0.534)
pop	0.013***
	(0.003)
elf:pop	-0.017***
	(0.005)
Constant	8.768***
	(0.795)
Observations	83
R^2	0.846
Adjusted R^2	0.833
Residual Std. Error	0.987 (df = 76)
F Statistic	69.365*** (df = 6; 76)
Note:	*p<0.1;* ***p<0.05;*** p<0.01

Let's examine the regression coefficients graphically:

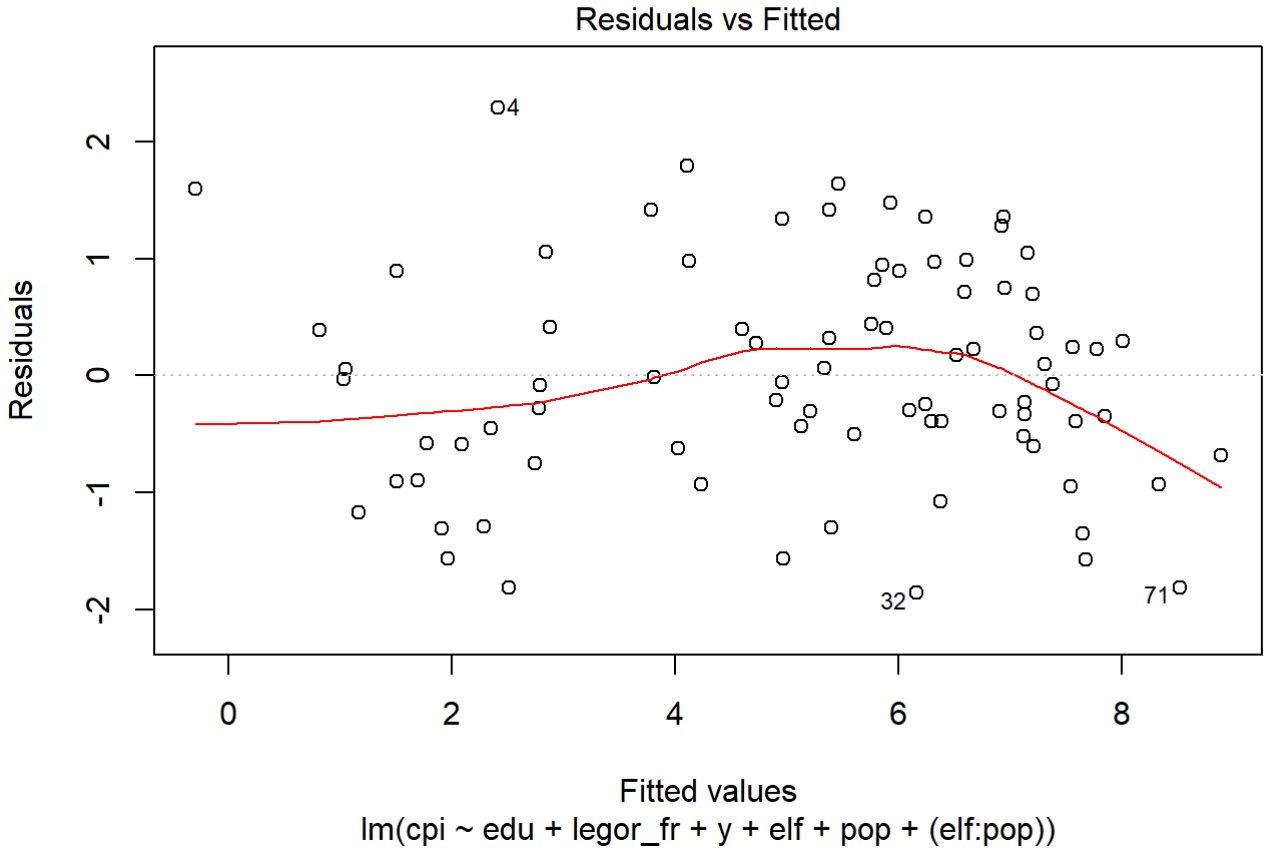
```
dwplot(reg10)
```

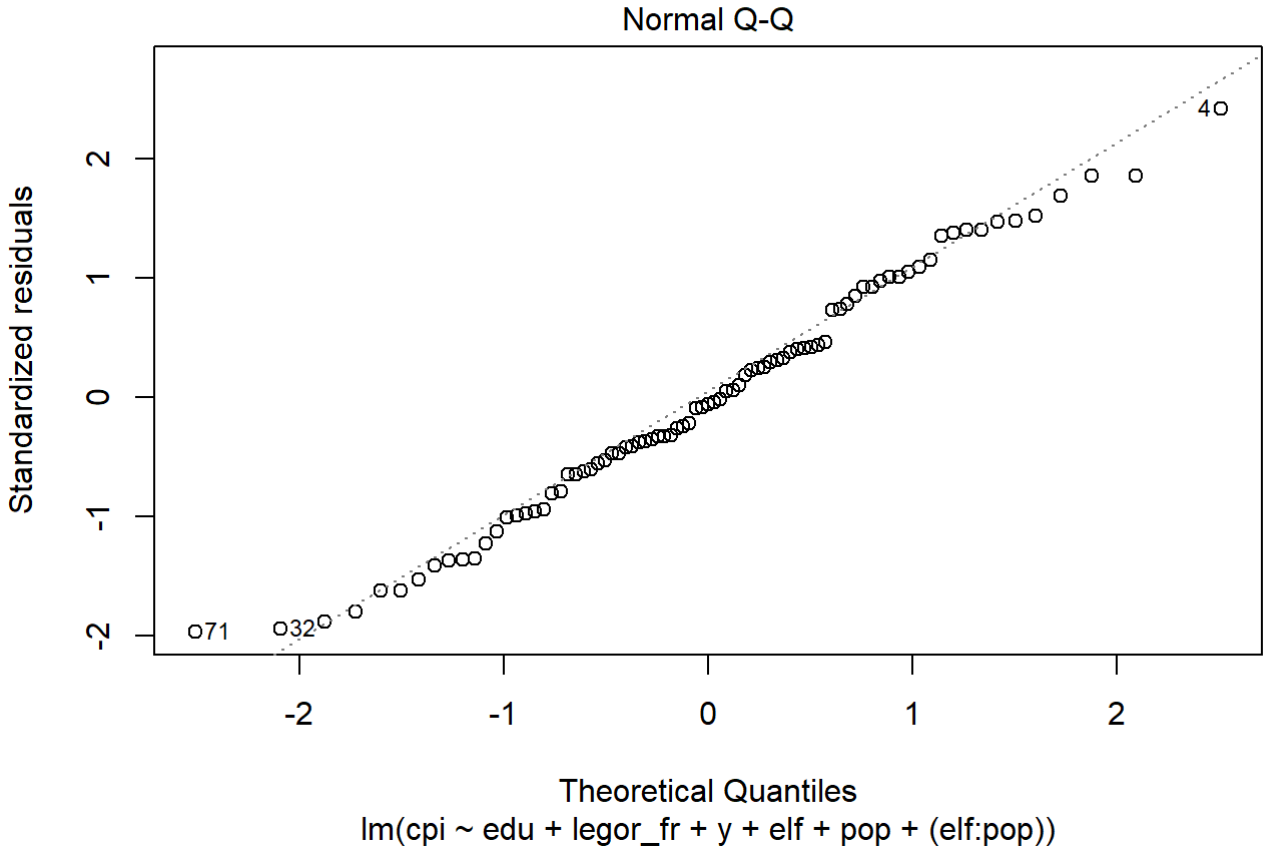


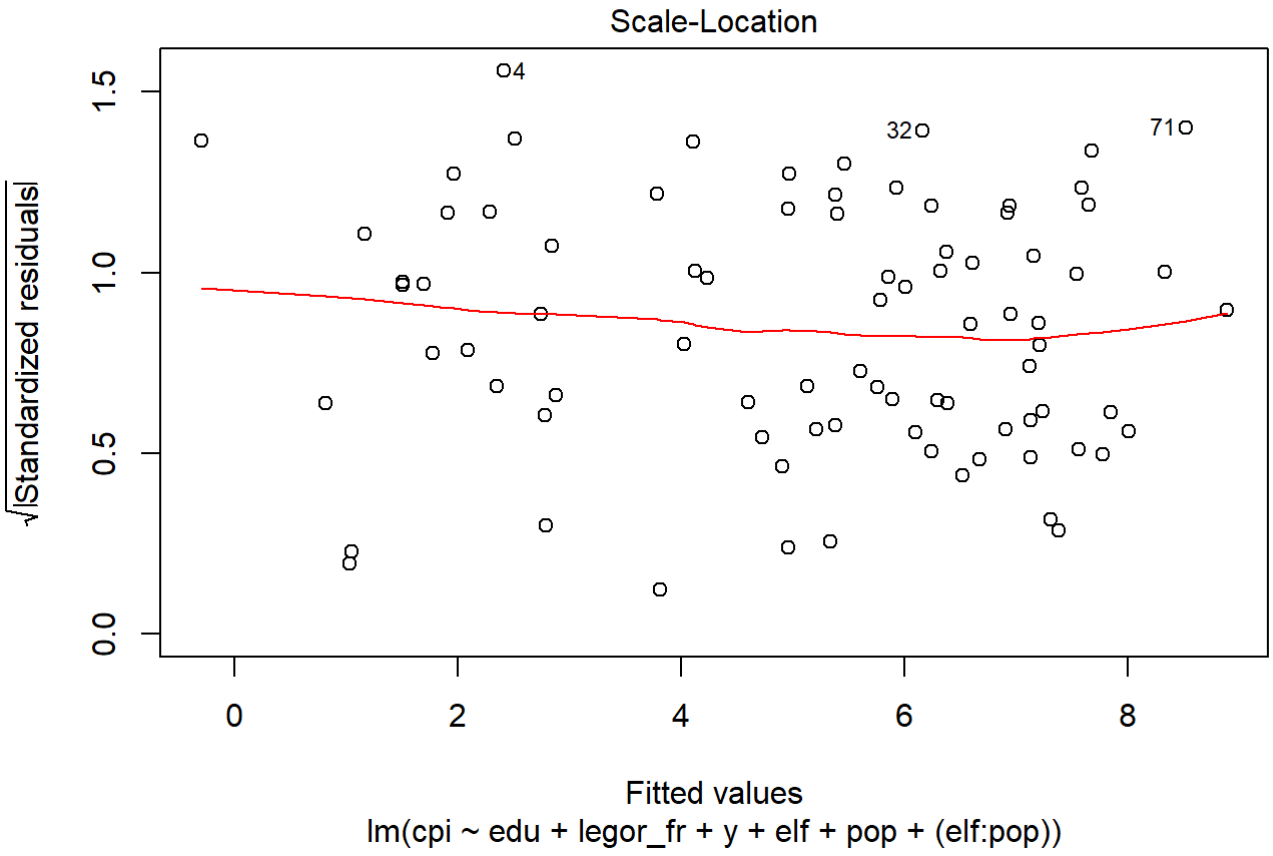
We will now apply the diagnostic tools we learned last week: residual plots, leverages, Studentized residuals, DFBETAs, Cook's distance, and multicollinearity. We want to know whether any of the core regression assumptions (linearity, normality and fixity) are violated.

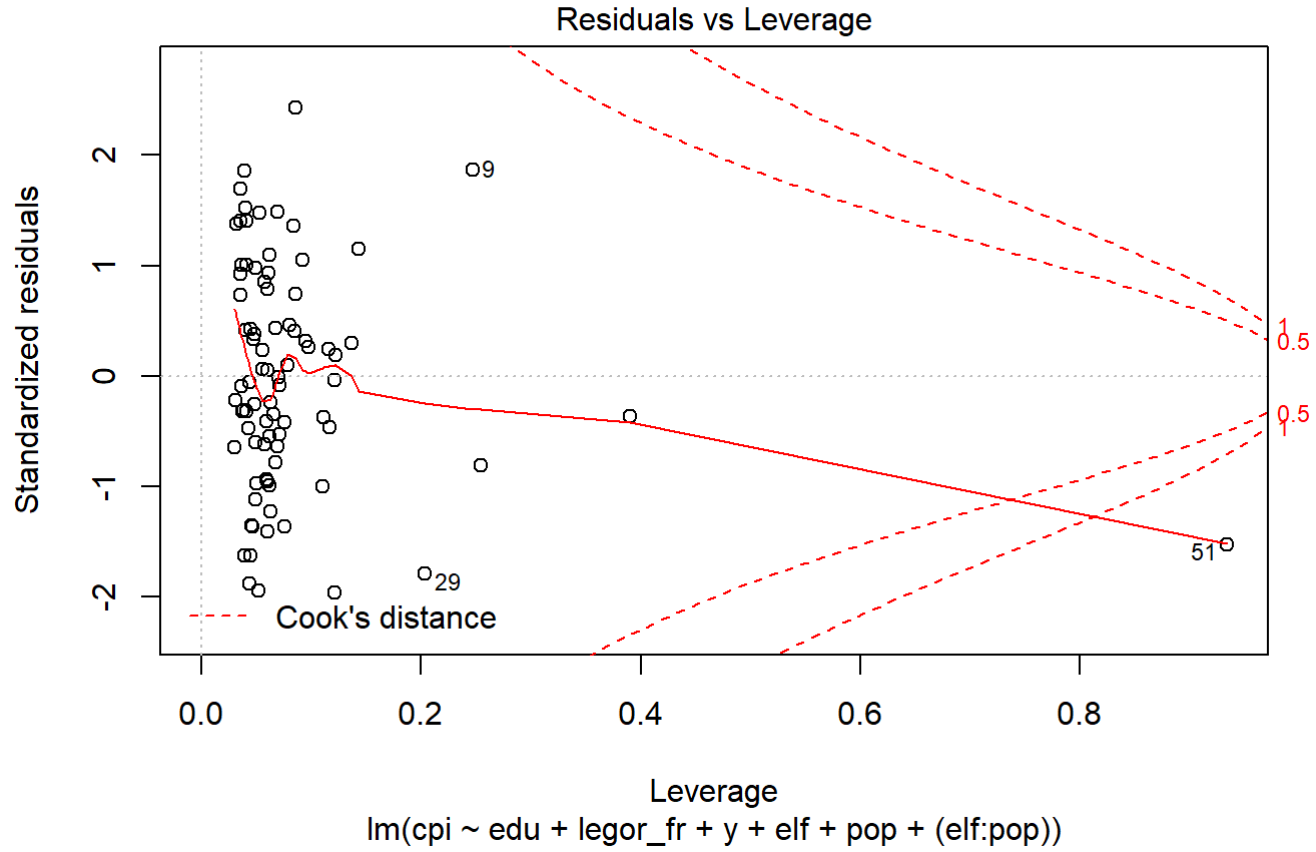
First we will examine the residual plots:

```
plot(reg10)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))
4
32
71


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))
4
32
71


Scale-Location
√|Standardized residuals|
4
32
71
0.0
0.5
1.0
1.5
0
2
4
6
8
Fitted values
lm(cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))


Residuals vs Leverage
Standardized residuals
9
29
51
Cook's distance
1
0.5
0.5
1
-2
-1
0
1
2
0.0
0.2
0.4
0.6
0.8
Leverage
lm(cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))

In the residuals vs. fitted plot, we observe weak linearity at the left and center of the distribution, with a hint of curvature at the right side. In addition, we observe homoskedasticity, as error terms appear to be uncorrelated with x.

The q-q plot verifies the normality of our distribution, with notable outliers at 4, 71 and 32 (Belgium, Senegal and Costa Rica).

The scale-location plot shows an even spread of residuals along the range of predictors, with a nearly-horizontal fitted line, indicating homoskedasticity.

Finally, the residuals vs. leverage plot will assist us in identifying influential cases. Observations beyond the Cook's distance lines are relatively more influential. In our case, observation #51 (India) may be driving our results.

To investigate, we will drop observation #51 from the model and compare the results: **Excellent**

```
Tabellini<-Tabellini[-c(51),]
reg10 <- lm(cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))
summary(reg10)
```

```
##
## Call:
## lm(formula = cpi ~ edu + legor_fr + y + elf + pop + (elf:pop))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86176 -0.60012 -0.05559  0.72937  2.28870
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.768e+00  7.955e-01  11.022  < 2e-16 ***
## edu         -2.726e-02  8.232e-03  -3.311 0.001423 **
## legor_fr     8.706e-01  2.255e-01   3.860 0.000236 ***
## y           -2.075e-04  1.676e-05 -12.379  < 2e-16 ***
## elf          4.130e-01  5.339e-01   0.774 0.441594
## pop          1.310e-02  3.433e-03   3.816 0.000274 ***
## elf:pop     -1.748e-02  5.245e-03  -3.333 0.001329 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9865 on 76 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.8456, Adjusted R-squared:  0.8334
## F-statistic: 69.36 on 6 and 76 DF,  p-value: < 2.2e-16
```

The results are substantially the same without observation #51.

Good

Now we will examine residual plots for each variable:

```
require(car)
```

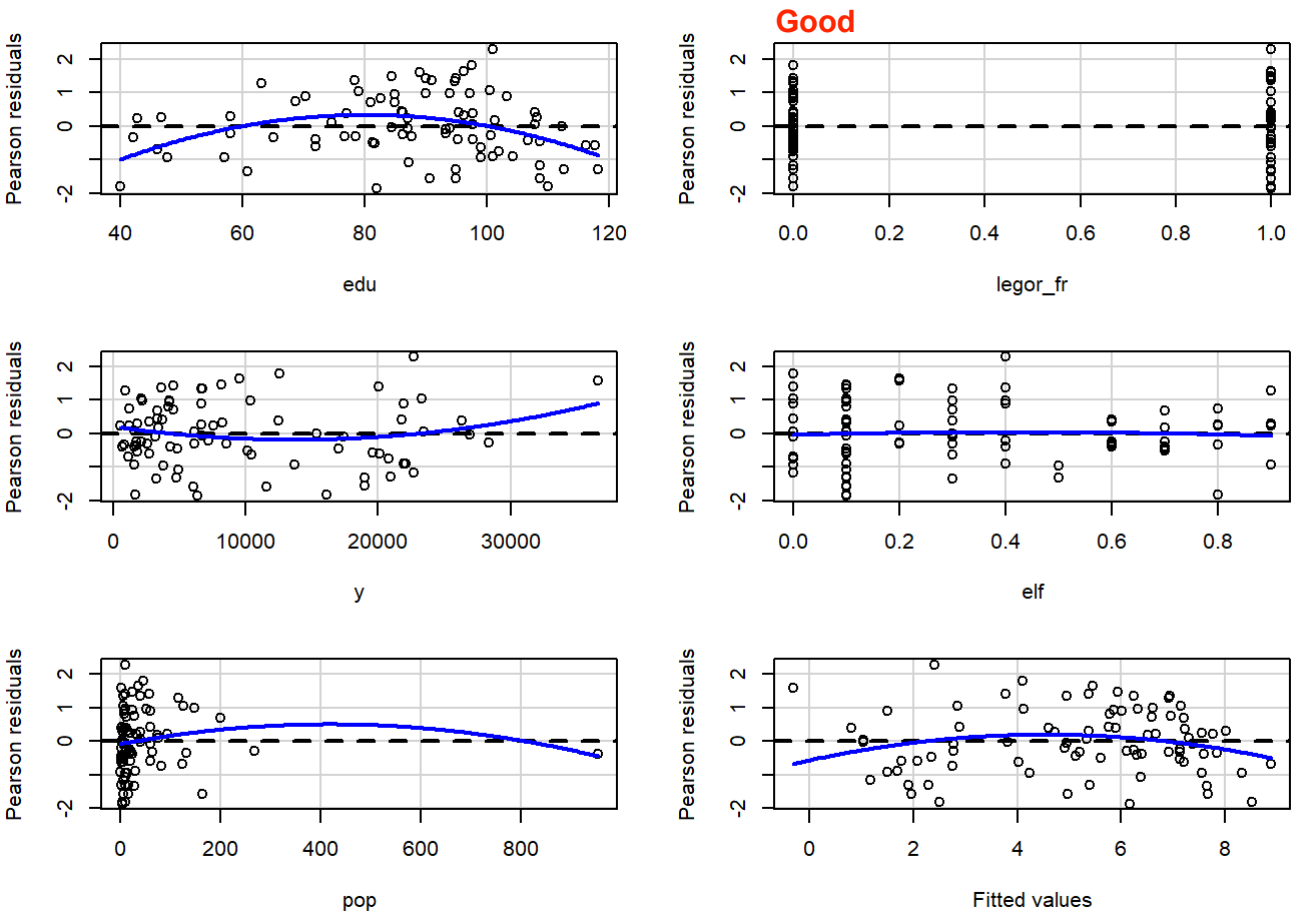
```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 3.5.2
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 3.5.2
```

```
residualPlots(reg10)
```

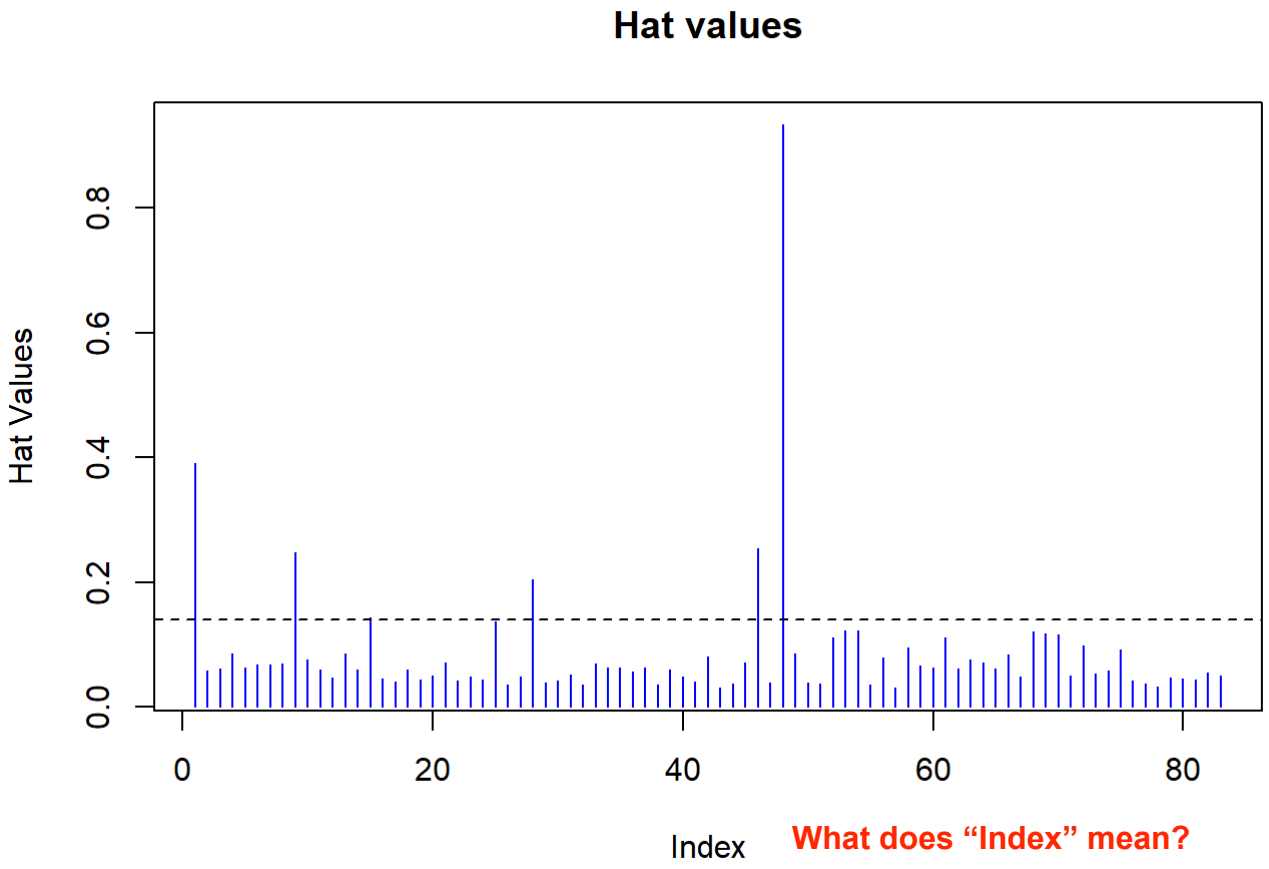



```
##            Test stat Pr(>|Test stat|)
## edu          -3.7535        0.0003418 ***
## legor_fr     -2.1596        0.0340035 *
## y             1.6863        0.0958985 .
## elf          -0.2889        0.7734701
## pop          -1.7867        0.0780220 .
## Tukey test   -2.2512        0.0243713 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ignoring dummy and categorical variables, we nevertheless see curvilinear relationships in four variables. This casts doubt on the linearity assumption. **Good**

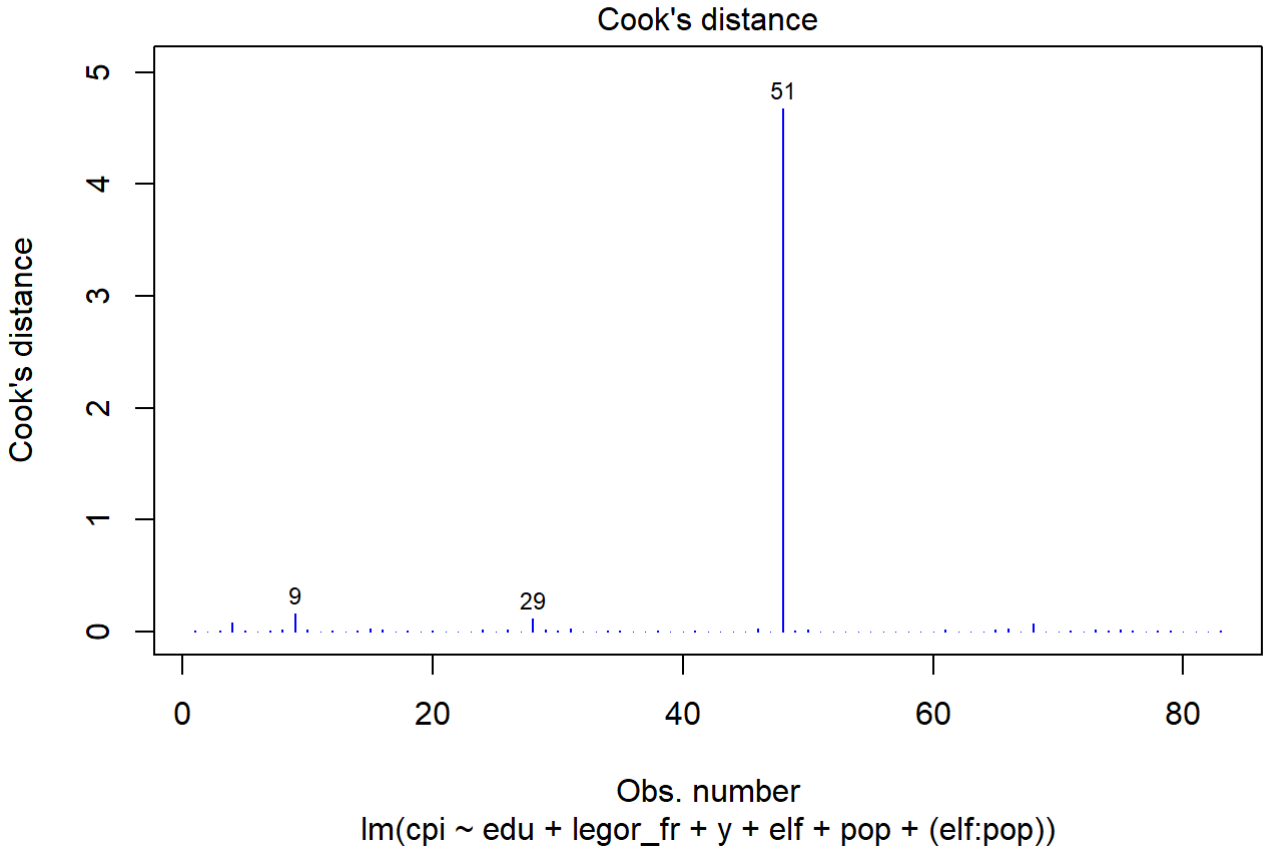
Continuing our investigation, we will look at leverage by graphing the hat values (high-leverage points, outliers in x):

```
plot(hatvalues(reg10), type='h', col="blue", ylab="Hat Values", main="Hat values")
p = 6; n = 86
abline(h = 2*p/n, lty=2) # Threshold for suspects
```



Next we will examine Cook’s distance:

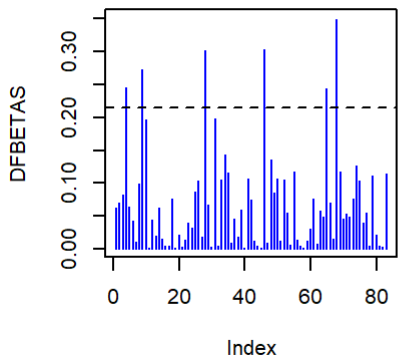
```
plot(reg10, which=4, col="blue")
```

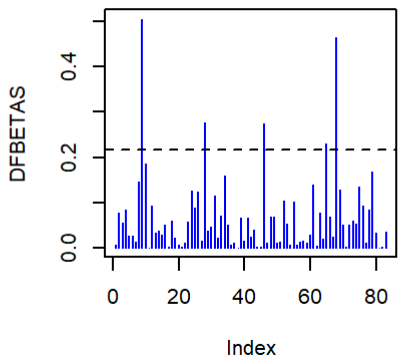


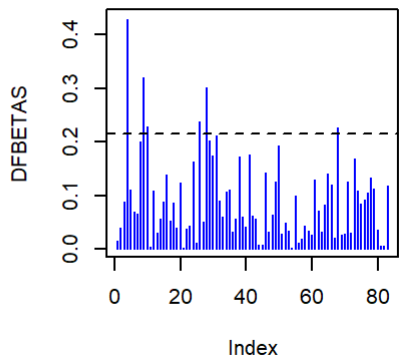
Here, observation #48 (Bangladesh) is the only significant outlier.

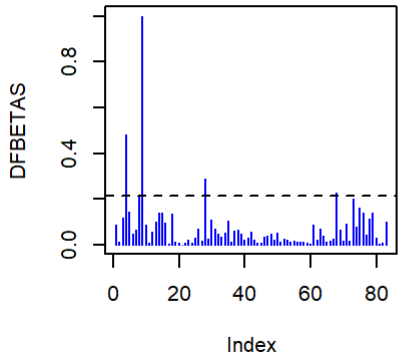
Finally, we will examine the DFBetas:

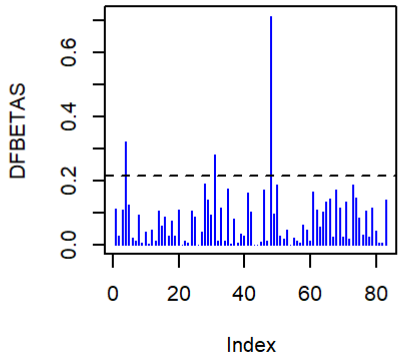
```
par(mfrow=c(2,3))
for (j in 1:7){
  plot(abs((dfbetas(reg10))[,j]), col=4, type='h', ylab='DFBETAS')
  abline(h = 2/sqrt(n), lty=2) # Threshold for suspects
}
```

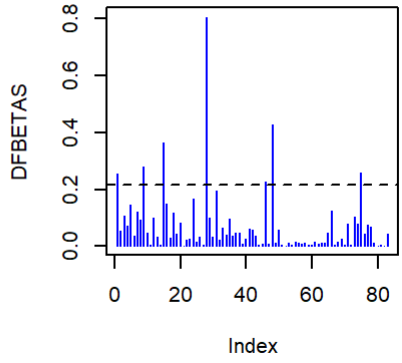


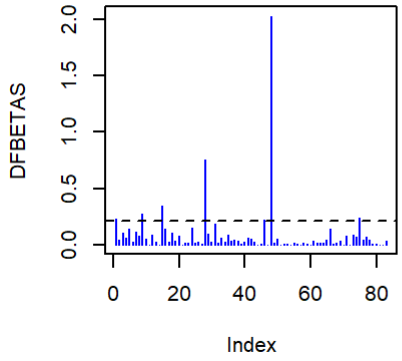












While

Great job interpreting these tests!

individual observations cross the concern threshold on all variables, relatively few of them do, and in addition

different observations have high DFBetas for different variables (rather than the same ones).

Question 2a.

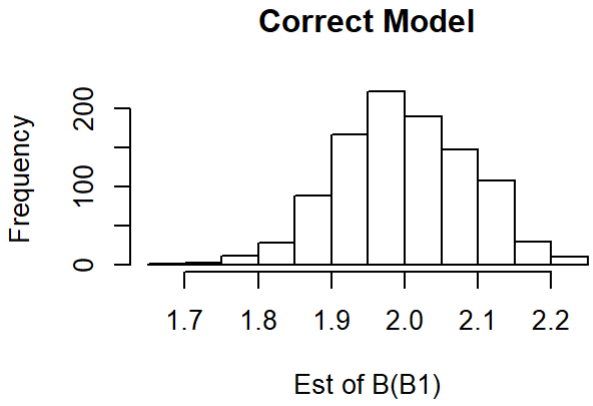
The question asks us what happens to the estimate of b1 when x2 is missing, in two cases: x1 and x2 have positive covariance, and x1 and x2 have negative covariance. We will use the omitted variable simulation from lecture:

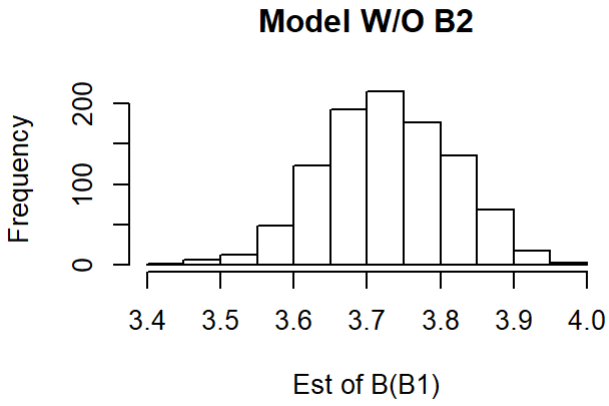
```
omitsim<-function(n=100,numsims=1000,rho1=2){
  par(mfrow=c(2,2))
  B1<-runif(n)*4
  B2<-rnorm(n,mean=100,sd=10)+(B1*rho1) ## The variable rho1 is our measure of correlati
on.
  print(cor(B1,B2))
  res<-array(NA,dim=c(numsims,2))
  for(i in 1:numsims){
    Y<-10+2*B2 + 2*B1+rnorm(n)
    res[i,1]<-lm(Y~B1+B2)$coef[2]
    res[i,2]<-lm(Y~B1)$coef[2]}
  print(lm(Y~B1+B2))
  print(lm(Y~B1))
  hist(res[,1],main="Correct Model",xlab="Est of B(B1)")
  hist(res[,2],main="Model W/O B2",xlab="Est of B(B1)")
  plot(res[,1],res[,2],xlim=c(min(res),max(res)),ylim=c(min(res),max(res)))
  abline(0,1)
}
```

We will now run the omitsim function. B1 and B2 are positively correlated because of the relationship in line 19. Our correlation measure is a positive number, so the simulation will produce a positive covariance, with residuals clustering in the top left corner. When x2 is missing, our overestimate of B1 will increase. As positive covariance increases, the amount by which we overestimate B1 increases when we drop x2.

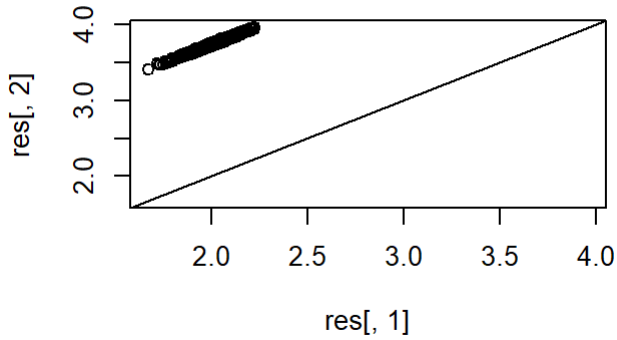
```
omitsim(n=100,numsims=1000,rho1=2)
```

```
## [1] 0.1003581
##
## Call:
## lm(formula = Y ~ B1 + B2)
##
## Coefficients:
## (Intercept)           B1           B2
##       9.968        1.987        2.000
##
##
## Call:
## lm(formula = Y ~ B1)
##
## Coefficients:
## (Intercept)           B1
##     214.552        3.716
```





If we

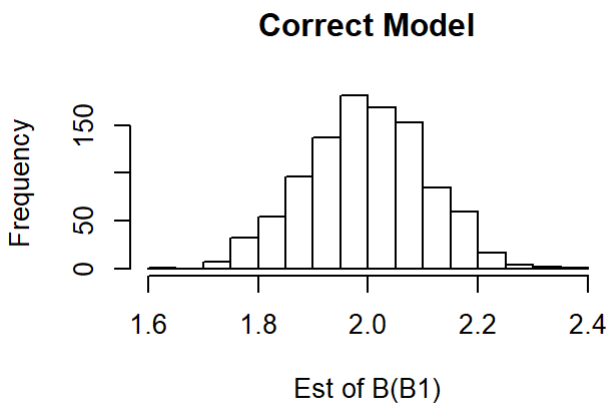


Good

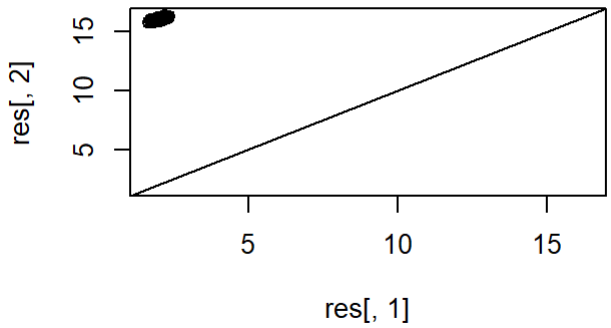
increase the magnitude of rho1, the gap between the models will increase:

```
omitsim(n=100,numsims=1000,rho1=7)
```

```
## [1] 0.621012
##
## Call:
## lm(formula = Y ~ B1 + B2)
##
## Coefficients:
## (Intercept)           B1           B2
##      10.514        2.086        1.993
##
##
## Call:
## lm(formula = Y ~ B1)
##
## Coefficients:
## (Intercept)           B1
##      212.25        16.08
```



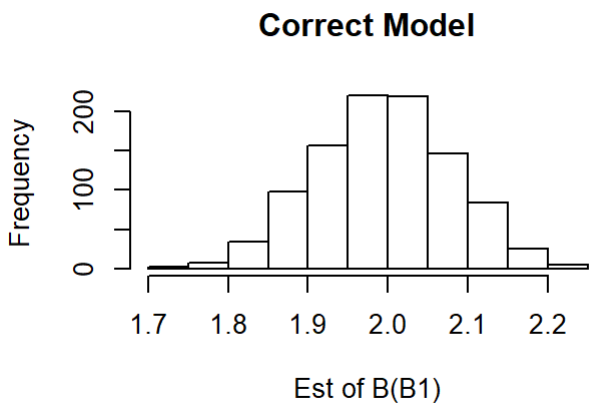
When

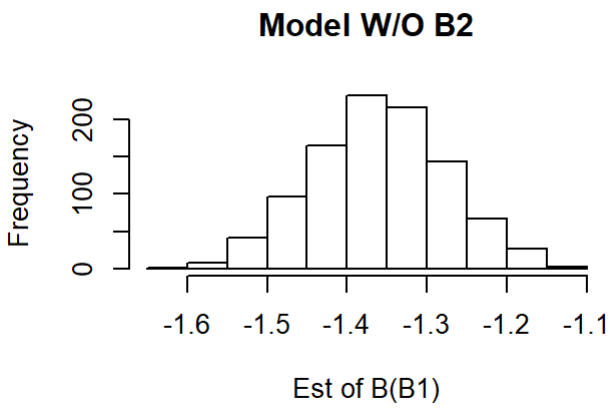


we change the rho1 value to a negative number, this creates a negative covariance between B1 and B2. Residuals will cluster in the bottom right corner. When x2 is missing, our overestimate of B1 will decrease. As negative covariance increases, the amount by which we overestimate B1 increases when we drop x2.

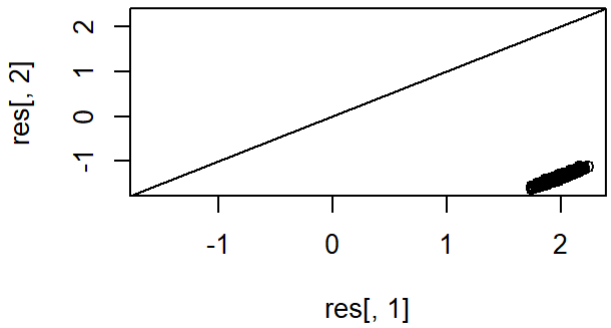
```
omitsim(n=100,numsims=1000,rho1=-2)
```

```
## [1] -0.1890616
##
## Call:
## lm(formula = Y ~ B1 + B2)
##
## Coefficients:
## (Intercept)           B1           B2
##       8.301        2.025        2.016
##
##
## Call:
## lm(formula = Y ~ B1)
##
## Coefficients:
## (Intercept)           B1
##     204.646       -1.357
```





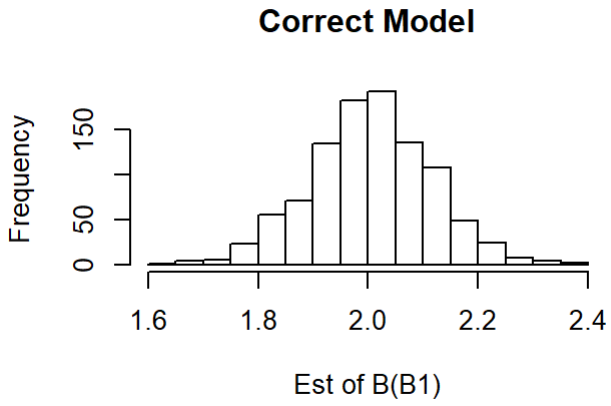
If we

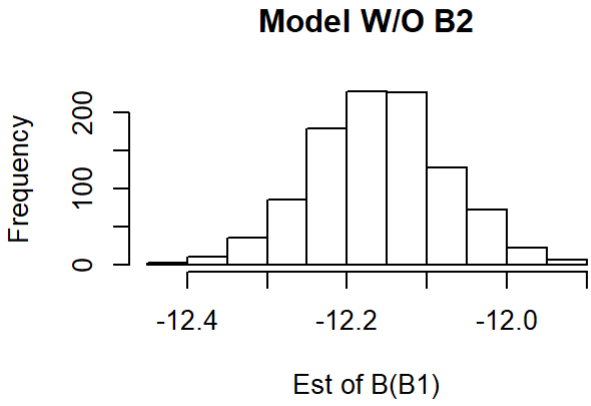


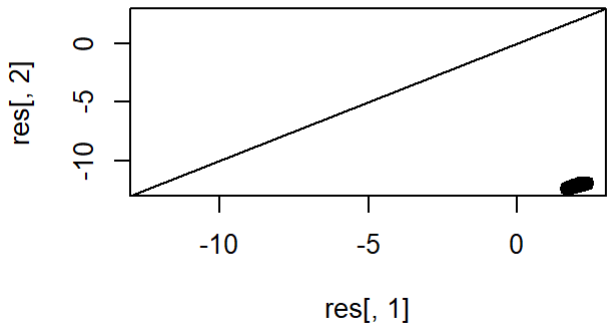
increase the magnitude of rho1, the gap between the models will increase:

```
omitsim(n=100,numsims=1000,rho1=-7)
```

```
## [1] -0.6169707
##
## Call:
## lm(formula = Y ~ B1 + B2)
##
## Coefficients:
## (Intercept)           B1           B2
##      11.430        1.905        1.986
##
##
## Call:
## lm(formula = Y ~ B1)
##
## Coefficients:
## (Intercept)           B1
##      212.89       -12.16
```





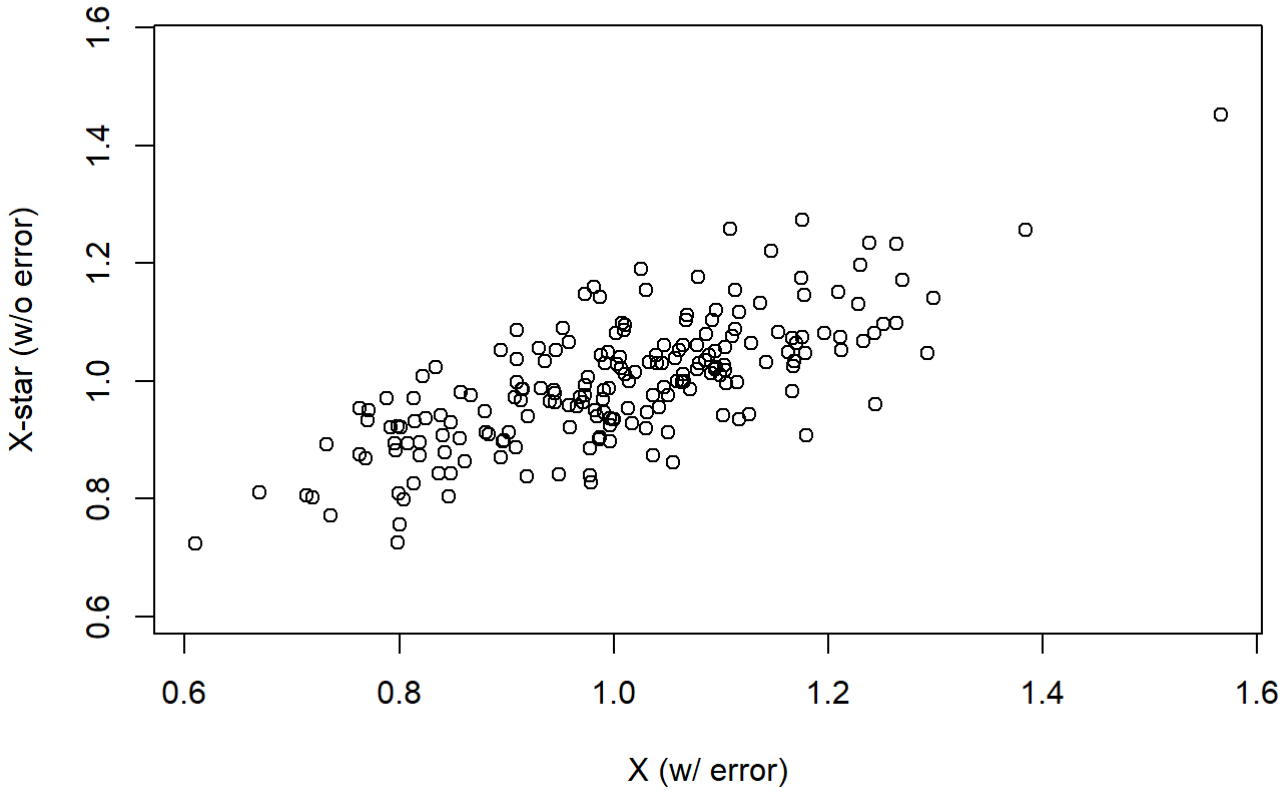


Question 2b.

In general, when x1 is measured with random error, the errors will tend to flatten the slope of the regression line. This is because the error will tend to disperse the data, resulting in a bigger, less-precise cloud. As random error increases, we therefore expect to see beta hat approach zero. **Good**

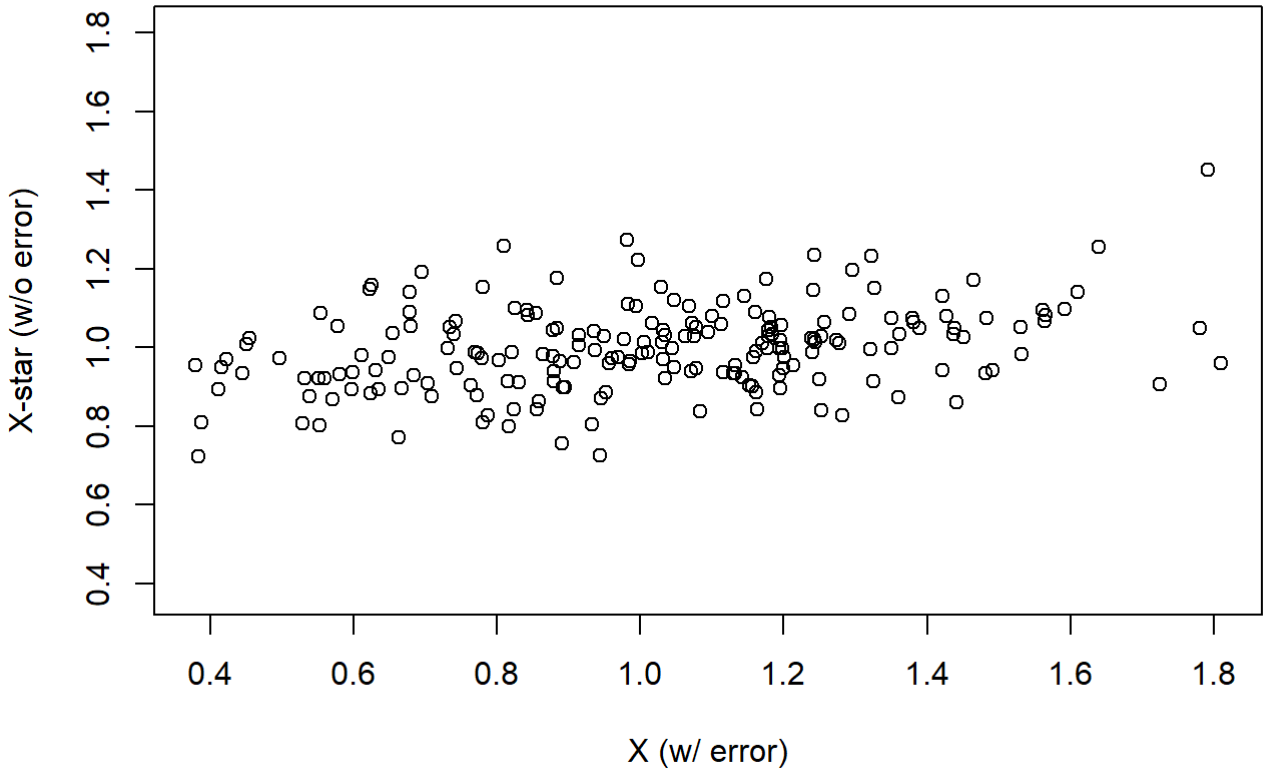
First, we examine the data with a normally distributed error term with a standard deviation of 1. We will graph x (with error) on the x-axis against x-star (true value) on the y-axis. We expect to see dispersion along the x axis.

```
set.seed(030419)
regsim0<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){
  X<-rnorm(n,mean=1)
  res<-array(NA,dim=c(numsims,2))
  for(i in 1:numsims){
    Ystar<-alpha+beta*X+rnorm(n,sd=sereg)
    Y<-Ystar+(1*rnorm(n,sd=1))
    res[i,1]<-lm(Ystar~X)$coef[2]
    res[i,2]<-lm(Y~X)$coef[2]
  }
  plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max
(res)),ylim=c(min(res),max(res)))
}
regsim0()
```



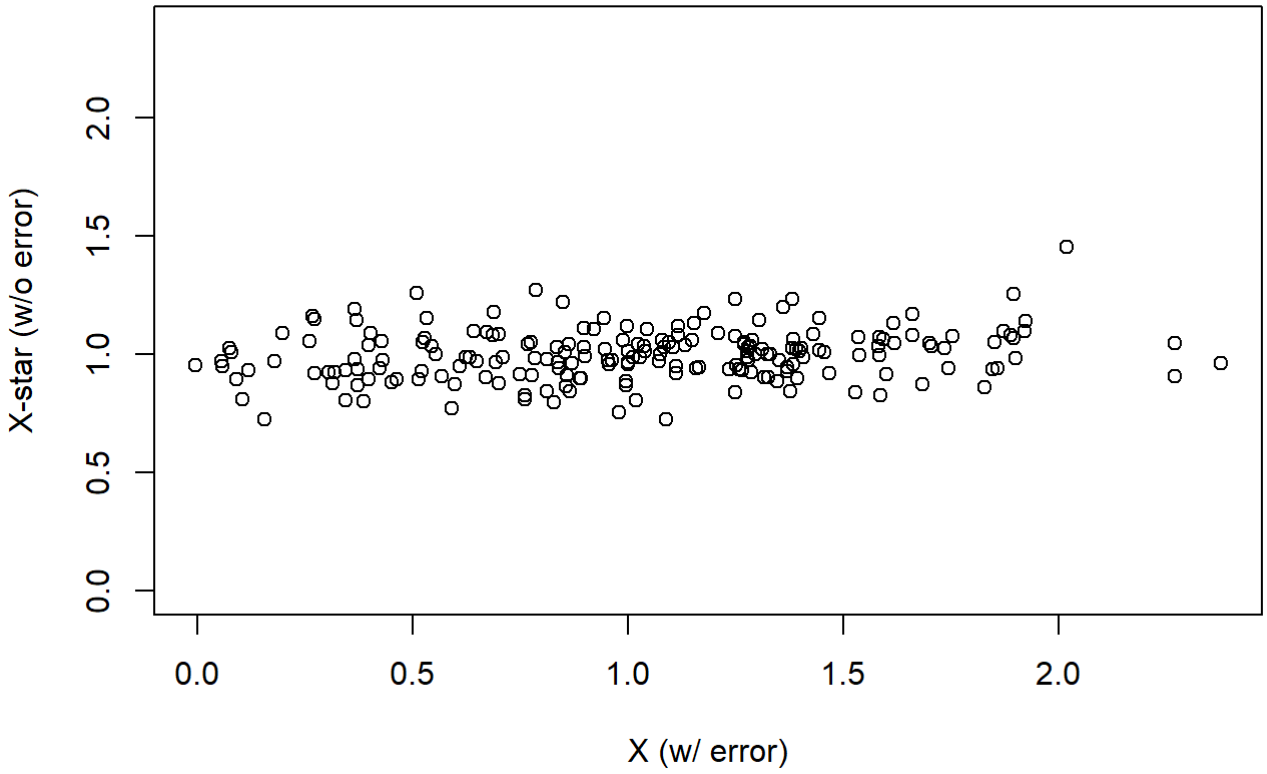
As we increase standard deviation, the distribution flattens:

```
set.seed(030419)
regsim0<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){
  X<-rnorm(n,mean=1)
  res<-array(NA,dim=c(numsims,2))
  for(i in 1:numsims){
    Ystar<-alpha+beta*X+rnorm(n,sd=sereg)
    Y<-Ystar+(3*rnorm(n,sd=1))
    res[i,1]<-lm(Ystar~X)$coef[2]
    res[i,2]<-lm(Y~X)$coef[2]
  }
  plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max
(res)),ylim=c(min(res),max(res)))
}
regsim0()
```

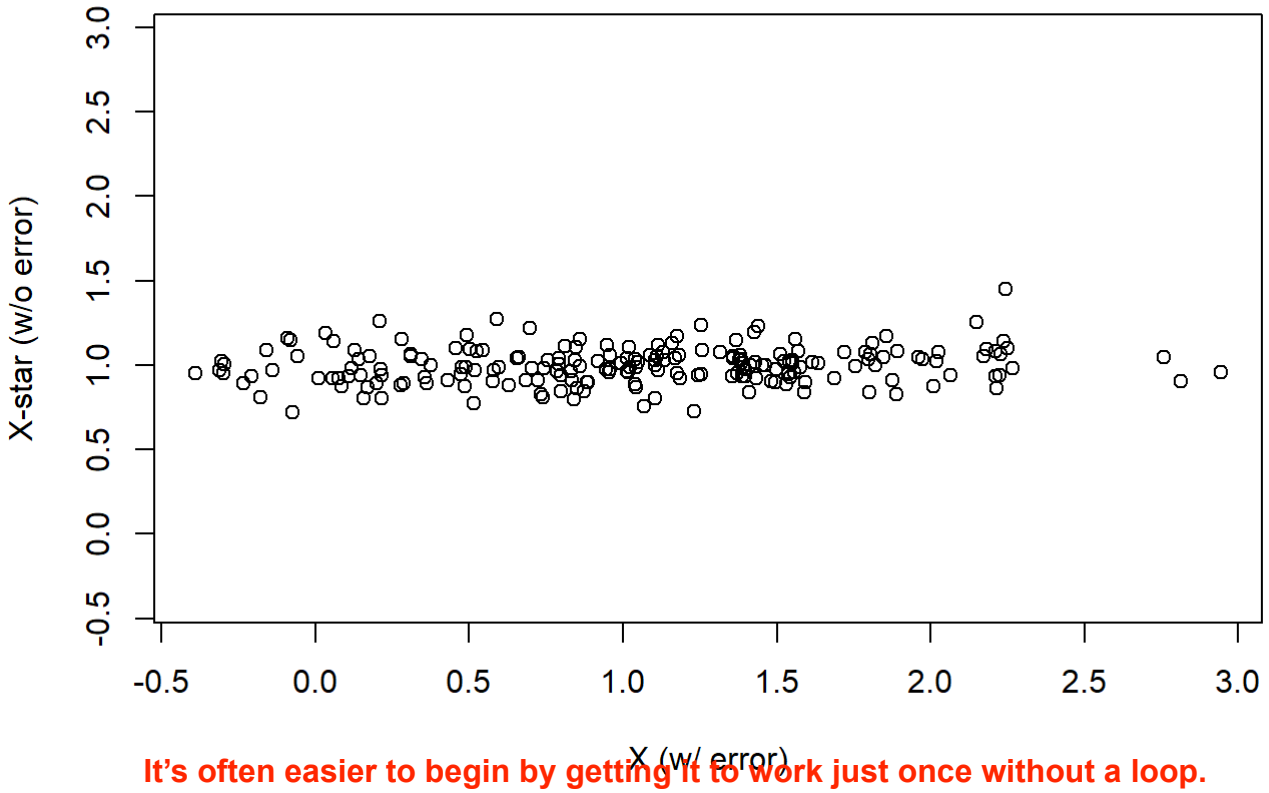


Eventually, the error term dominates the result:

```
set.seed(030419)
regsim0<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){
  X<-rnorm(n,mean=1)
  res<-array(NA,dim=c(numsims,2))
  for(i in 1:numsims){
    Ystar<-alpha+beta*X+rnorm(n,sd=sereg)
    Y<-Ystar+(5*rnorm(n,sd=1))
    res[i,1]<-lm(Ystar~X)$coef[2]
    res[i,2]<-lm(Y~X)$coef[2]
  }
  plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max
(res)),ylim=c(min(res),max(res)))
}
regsim0()
```




```
set.seed(030419)
regsim0<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){
  X<-rnorm(n,mean=1)
  res<-array(NA,dim=c(numsims,2))
  for(i in 1:numsims){
    Ystar<-alpha+beta*X+rnorm(n,sd=sereg)
    Y<-Ystar+(7*rnorm(n,sd=1))
    res[i,1]<-lm(Ystar~X)$coef[2]
    res[i,2]<-lm(Y~X)$coef[2]
  }
  plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max
(res)),ylim=c(min(res),max(res)))
}
regsim0()
```



It's often easier to begin by getting it to work just once without a loop. Then when you know it works, you can copy and paste your code into the loop.

We will now create a variable Z, correlated with the true x1:

set.seed(030419) regsim0<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){ X<-rnorm(n,mean=1) Z<-X+(.1*(rnorm(n,mean=1))) X1hat<-lm(Z~X) predictions <- predict(X1hat, newdat=X1hat, se.fit=TRUE, interval="confidence", level=0.95) res<-array(NA,dim=c(numsims,2)) for(i in 1:numsims){ Ystar<-alpha+beta*X1hat Y<-Ystar+(1*rnorm(n,sd=1)) res[i,1]<-lm(Ystar~X)\(coef[2] res[i,2]<-lm(Y~X)\)coef[2] } plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max(res)),ylim=c(min(res),max(res))) } regsim0()""

It looks like you are extracting the coefficient from the regression of X on Z. Instead, you want to be extracting the fitted values. If you changed X1hat to X1hat <- lm(Z~X)$fitted.values it would work -2

_ _Attempt #2:

regsim1<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){ X<-rnorm(n,mean=1) Z<-X+(.1*(rnorm(n,mean=1))) res<-array(NA,dim=c(numsims,3)) for(i in 1:numsims){ Y<-alpha+beta*X+rnorm(n,sd=sereg) Xstar<-X+rnorm(n,sd=sigmagamma) res[i,1]<-lm(Z~X)\(coef[2] X1hat<-res[i,1] res[i,2]<-lm(Y~X1hat)\)coef[2] res[i,3]<-lm(Y~Xstar)$coef[2] } plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max(res)),ylim=c(min(res),max(res))) abline(0,1) lines(c(beta,beta),c(min(res),max(res))) lines(c(min(res),max(res)),c(beta,beta)) } regsim1()

Attempt #3: (I'm so close - the trouble is that I can't figure out how to populate X1hat within the loop.)

regsim1<-function(n=100,numsims=200,alpha=1,beta=1,sigmagamma=1,sereg=1){ X<-rnorm(n,mean=1) Z<-X+(.1*(rnorm(n,mean=1))) res<-array(NA,dim=c(numsims,3)) for(i in 1:numsims){ Y<-alpha+beta*X+rnorm(n,sd=sereg) Xstar<-X+rnorm(n,sd=sigmagamma) res[i,1]<-lm(Z~X)\(coef[2] X1hat<-res[i,1] res[i,2]<-lm(Y~X1hat)\)coef[2] res[i,3]<-lm(Y~Xstar)$coef[2] } plot(res[,2],res[,1],xlab="X (w/ error)",ylab="X-star (w/o error)",xlim=c(min(res),max(res)),ylim=c(min(res),max(res))) abline(0,1) lines(c(beta,beta),c(min(res),max(res))) lines(c(min(res),max(res)),c(beta,beta)) } regsim1()

I'm not getting the results I want from these loops, but intuitively we would expect the regression on Z to track X1 to the extent of their correlation, but we will see divergence in the error terms.

Question 3.

I decided to reexamine Tabellini's 2001 corruption dataset from a different angle. I want to know whether the corruption perceptions index (CPI) score affects GDP. The proposed mechanism is that a perceptible increase in corruption-related rents would cause a corresponding deadweight loss to GDP.

H0: An increase (decrease) in CPI has no effect on GDP.
HA: An increase (decrease) in CPI causes an increase (decrease) in GDP.

```
Tabellini<-read.csv("Tabellini.csv", header=T)
y <- Tabellini$Y
cpi<-Tabellini$CPI
reg20<-lm(y ~ cpi)
```

```
stargazer::stargazer(reg20, type = "html",
 title = "lm(y ~ cpi)")
```

lm(y ~ cpi)

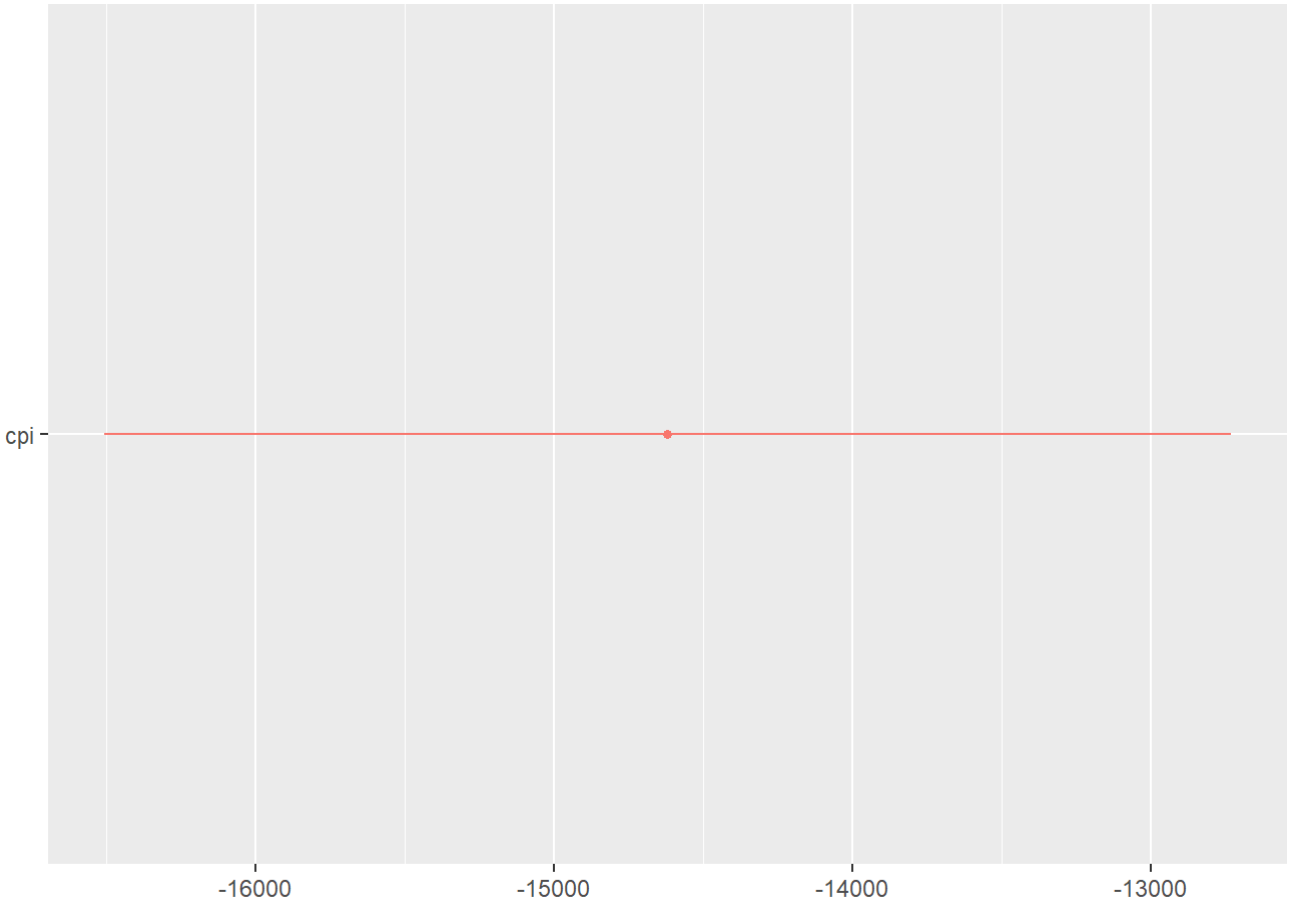
	Dependent variable:
	y
cpi	-3,042.323***
	(197.513)
Constant	25,205.680***
	(1,122.533)
Observations	84
R^2	0.743
Adjusted R^2	0.740
Residual Std. Error	4,323.446 (df = 82)
F Statistic	237.256*** (df = 1; 82)
Note:	*p<0.1;* ***p<0.05;*** p<0.01

P < .05, so we reject the null hypothesis. Corruption is strongly associated with lower GDP.

We might object that there's a lag built in to this model, in that present corruption might affect future gdp, but not present gdp. However, corruption perceptions are themselves composites of corruption perceptions over lifetimes, with perhaps a strong recency bias but at least the potential to encode older information.

Let's examine the regression coefficients graphically:

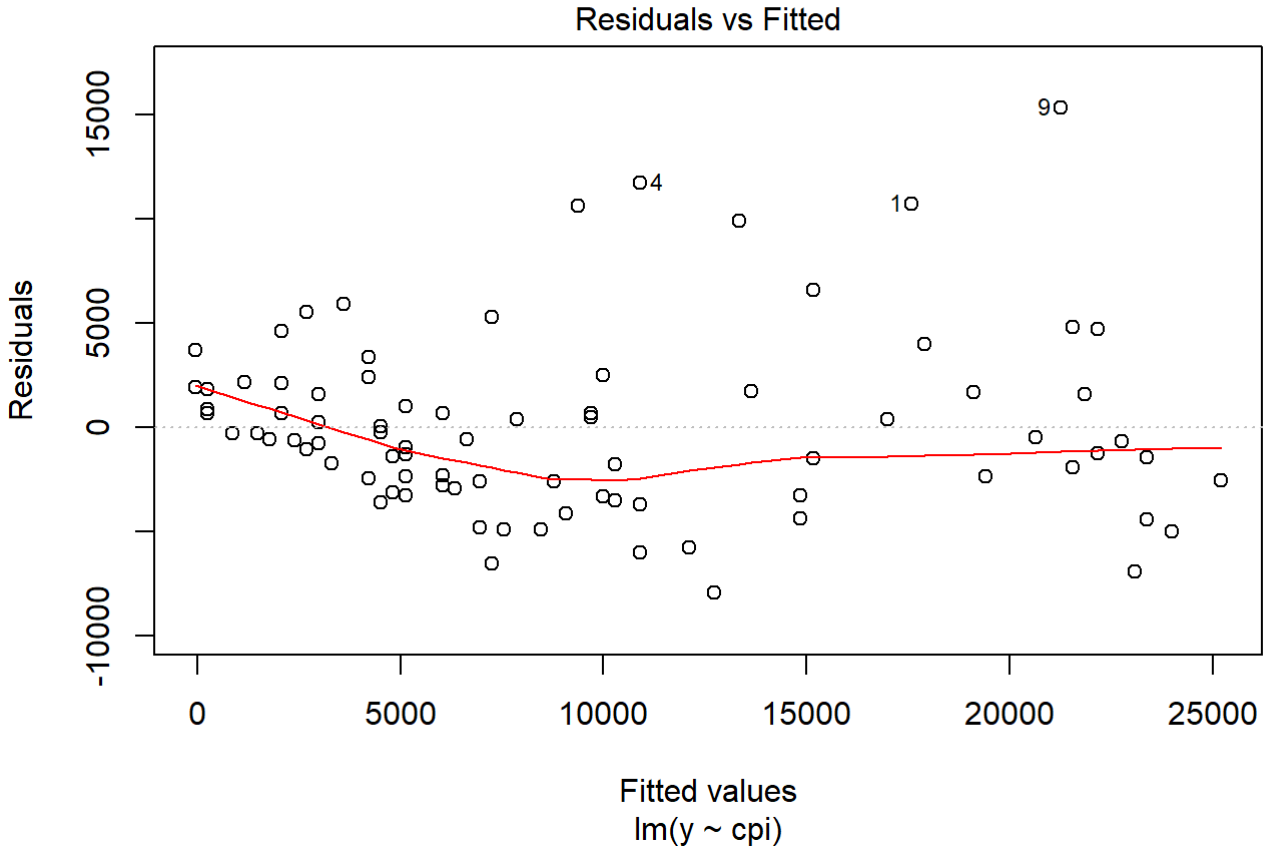
```
dwplot(reg20)
```

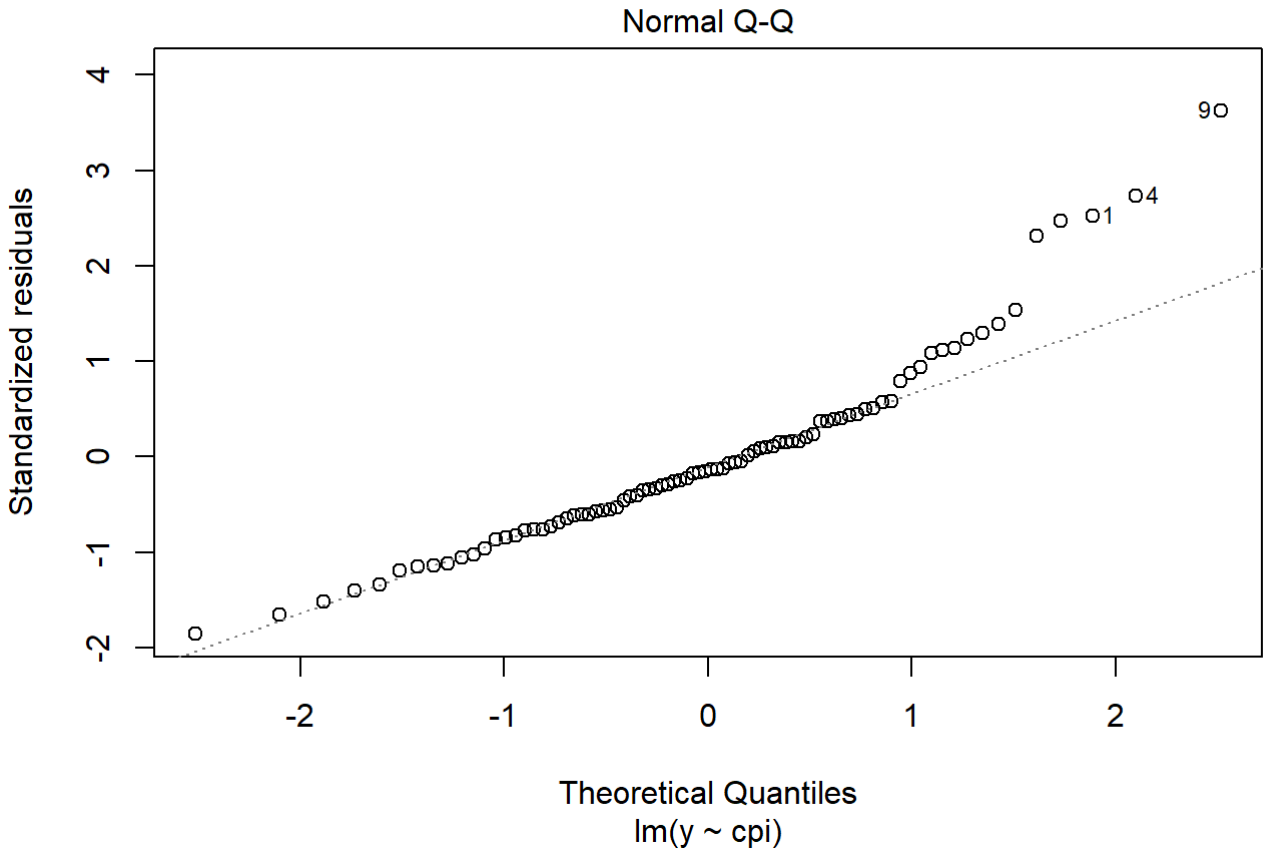


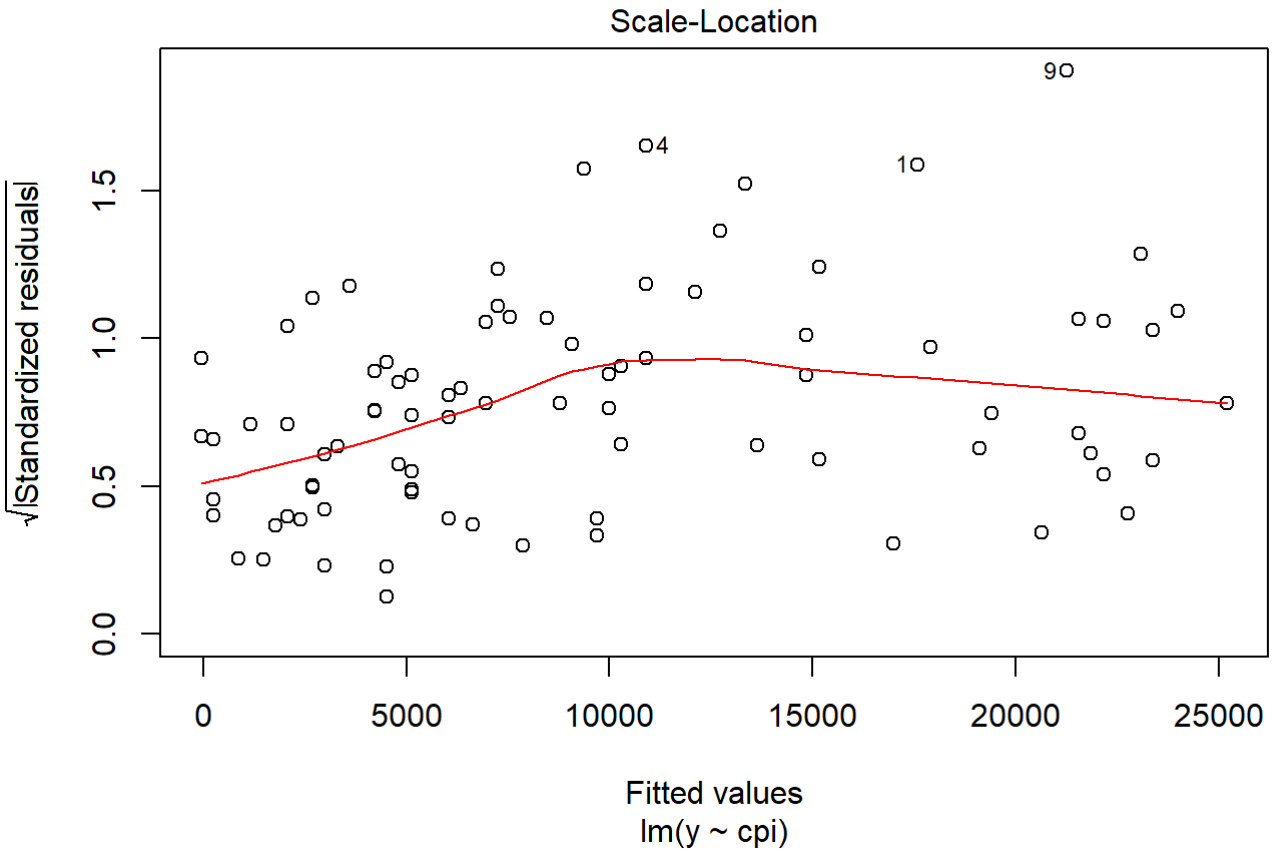
We will now apply the diagnostic tools we learned last week: residual plots, leverages, Studentized residuals, DFBETAs, Cook's distance, and multicollinearity. We want to know whether any of the core regression assumptions (linearity, normality and fixity) are violated.

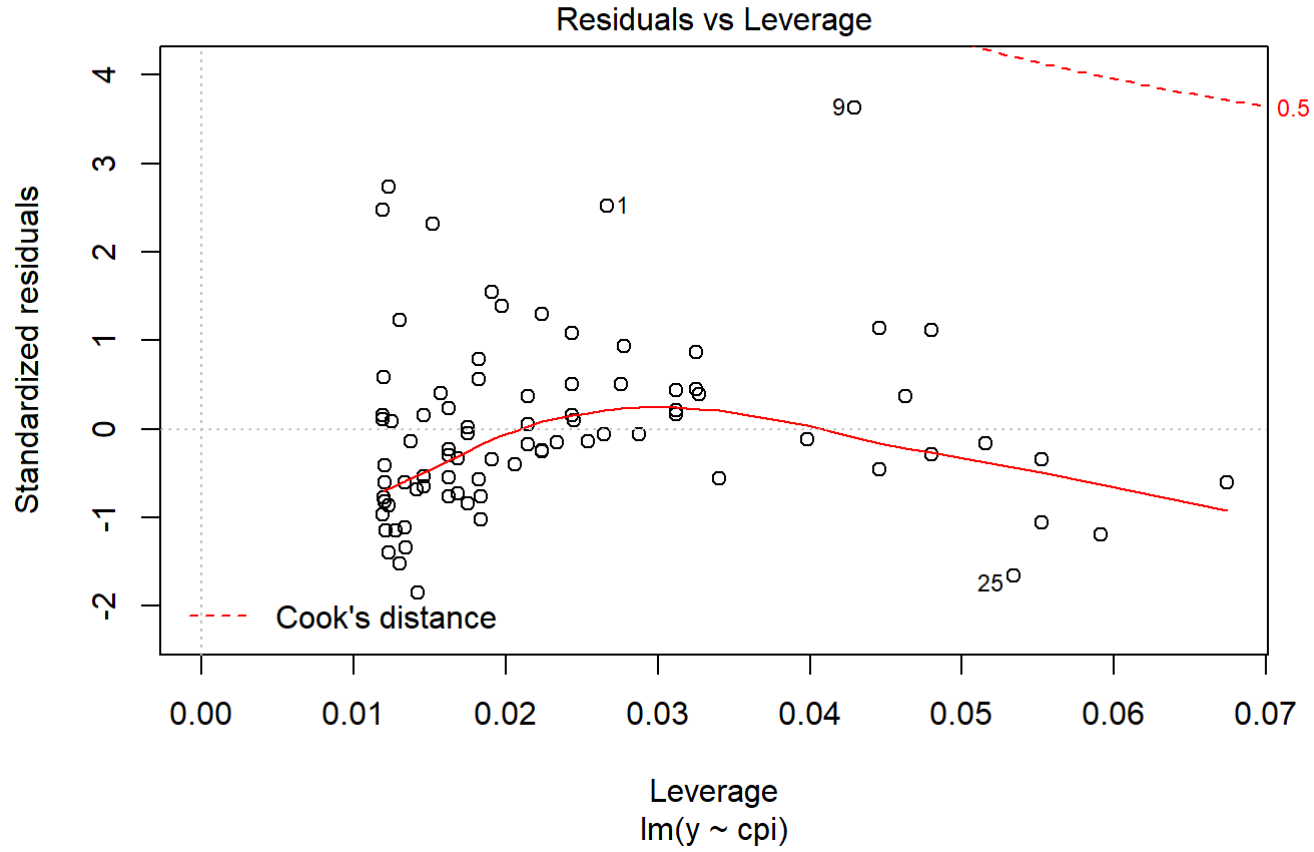
First we will examine the residual plots:

```
plot(reg20)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ cpi)
9
4
1
15000
5000
0
-10000
0
5000
10000
15000
20000
25000


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(y ~ cpi)
9
4
1
4
3
2
1
0
-1
-2
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
5000
10000
15000
20000
25000
Fitted values
lm(y ~ cpi)
9
4
1


Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
3
4
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Leverage
lm(y ~ cpi)
Cook's distance
0.5
9
1
25

In the residuals vs. fitted plot, we observe moderate linearity spread horizontally and fairly evenly, with outliers at 1, 4 and 9 (USA, Belgium, Luxembourg). In addition, we observe homoskedasticity, as error terms appear to be uncorrelated with x.

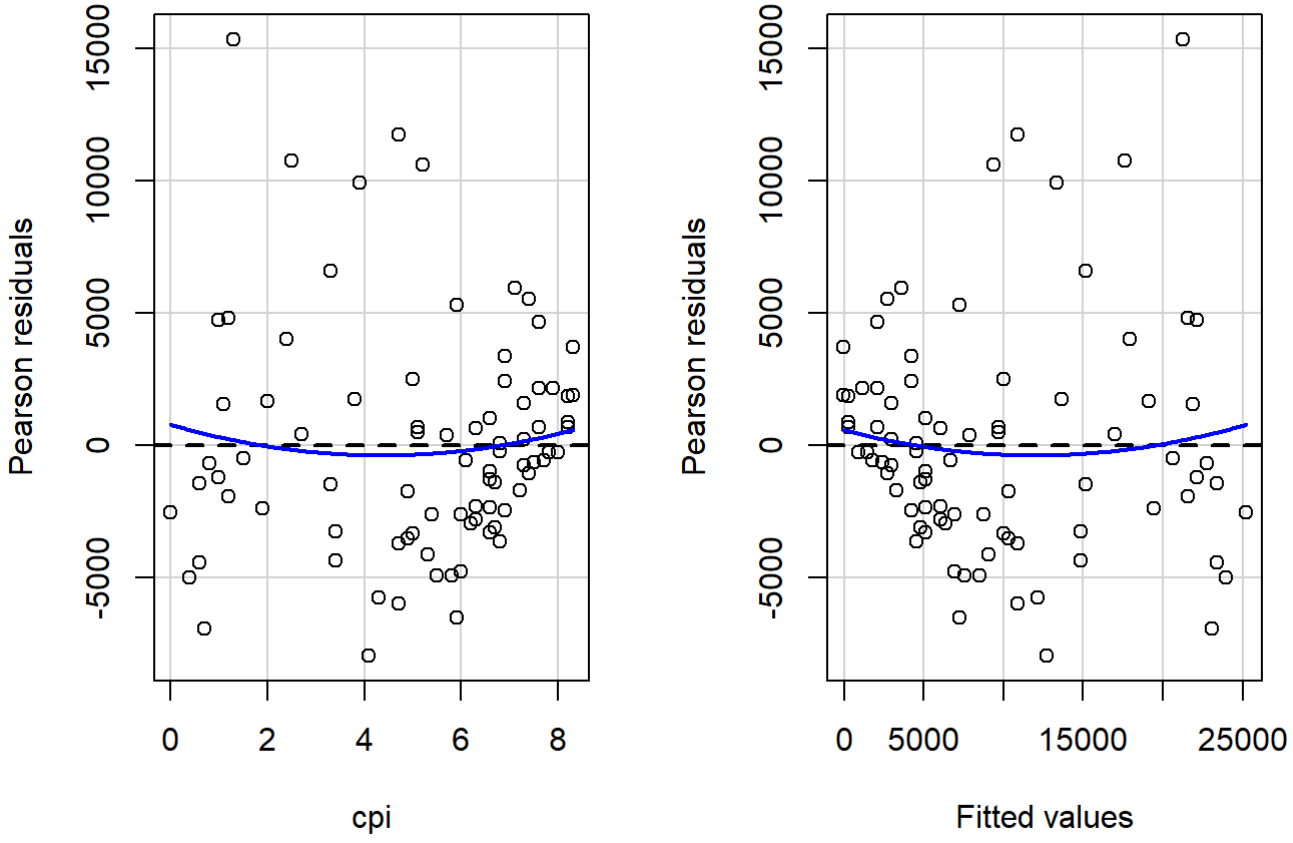
The q-q plot verifies that residuals are normally distributed, with the same notable outliers at 1, 4 and 9 (USA, Belgium, Luxembourg).

The scale-location plot shows an even spread of residuals along the range of predictors, with a nearly-horizontal fitted line, indicating homoskedasticity.

Finally, the residuals vs. leverage plot will assist us in identifying influential cases with high leverage. Observations beyond the Cook's distance lines are relatively more influential. In our case, observations 1, 9 and 25 (USA, Luxembourg and New Zealand) may be driving our results.

Now we will examine residual plots for each variable:

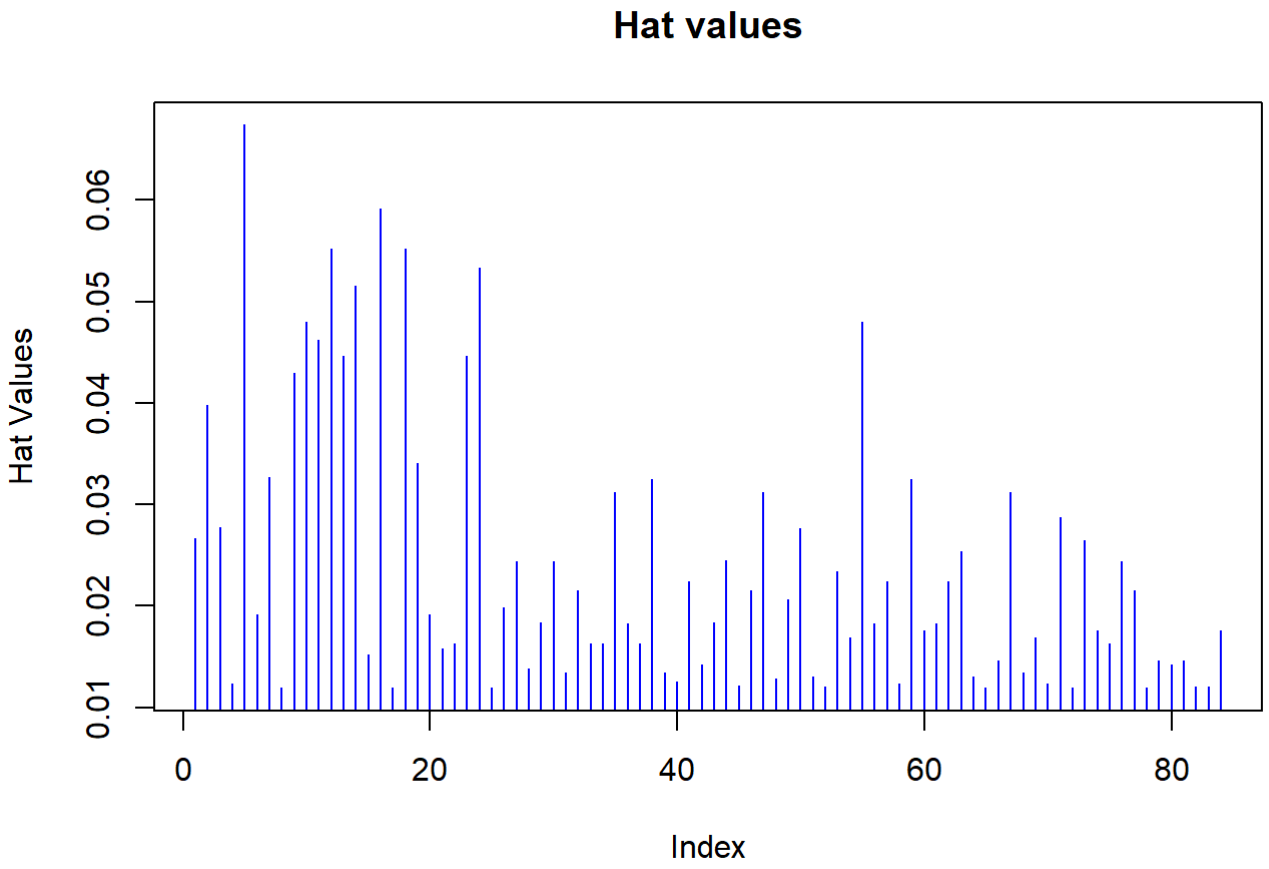
```
residualPlots(reg20)
```




```
##             Test stat Pr(>|Test stat|)
## cpi             0.648           0.5188
## Tukey test      0.648           0.5170
```

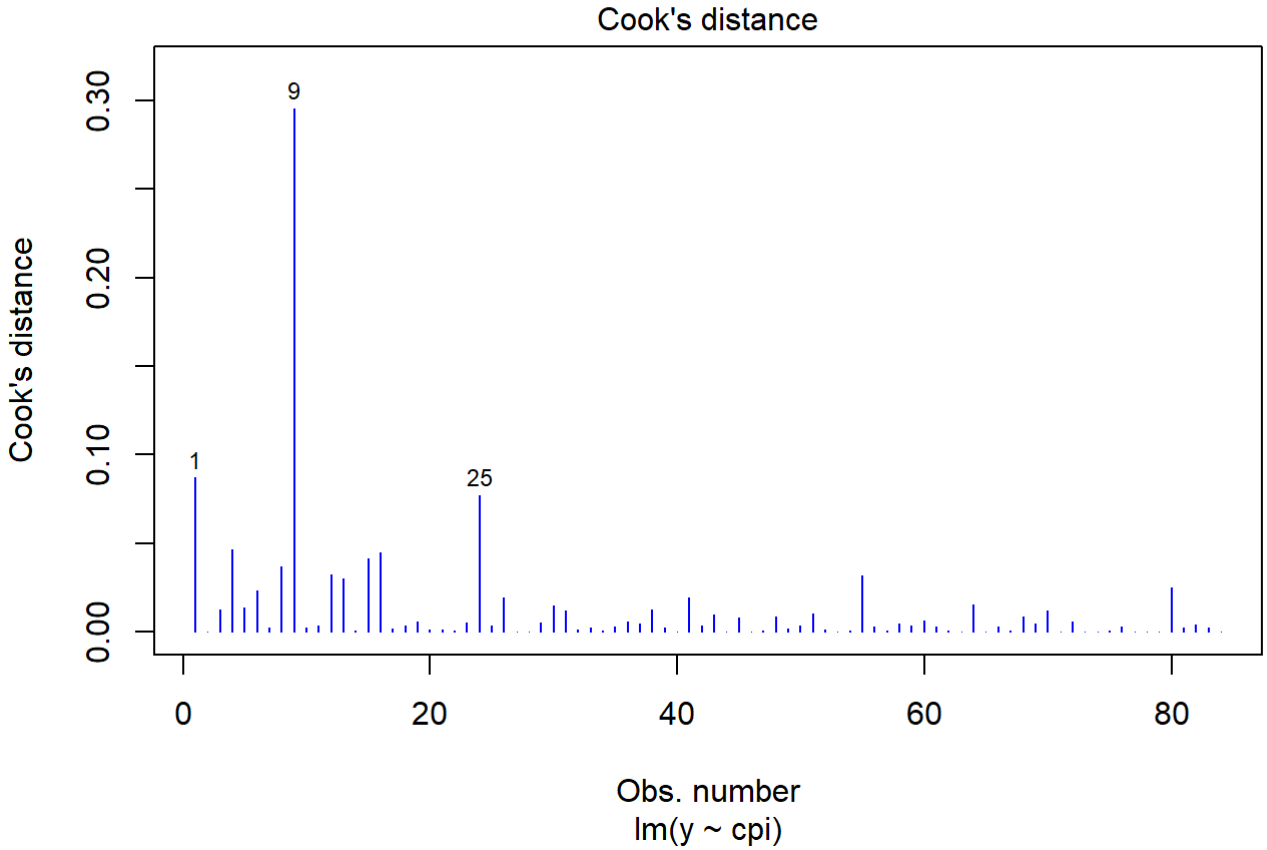
Now we will look at leverage by graphing the hat values (high-leverage points, outliers in x):

```
plot(hatvalues(reg20), type='h', col="blue", ylab="Hat Values", main="Hat values")
p = 6; n = 86
abline(h = 2*p/n, lty=2) # Threshold for suspects
```



Next we will examine Cook's distance:

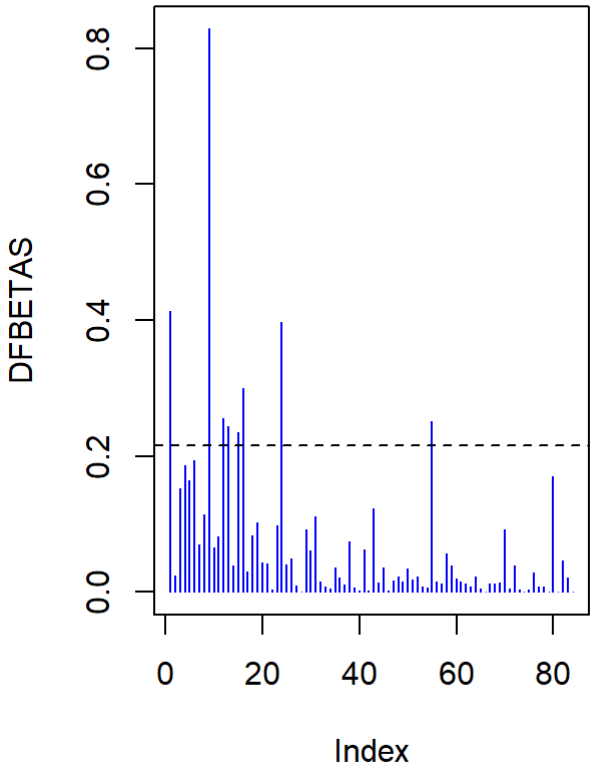
```
plot(reg20, which=4, col="blue")
```

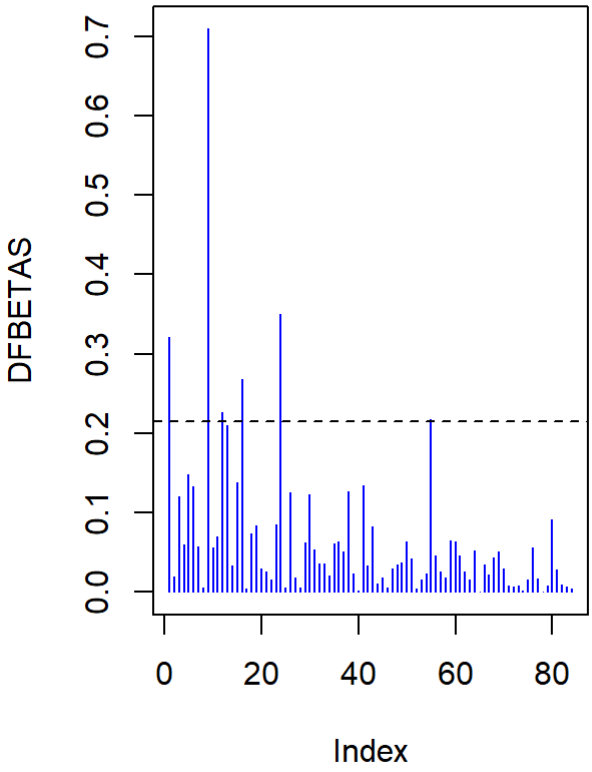


Here, observations 1, 9 and 25 (USA, Luxembourg and New Zealand) are significant outliers.

Finally, we will examine the DFBetas:

```
par(mfrow=c(1,2))
for (j in 1:2){
  plot(abs((dfbetas(reg20))[,j]), col=4, type='h', ylab='DFBETAS')
  abline(h = 2/sqrt(n), lty=2) # Threshold for suspects
}
```


DFBETAS
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
Index


DFBETAS
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
Index