

Draper HW 9

Grade: 96

Matthew Draper

March 13, 2019

Question 1: Heteroskedasticity Simulation

```
## We will test the second kind of heteroskedasticity, where sigma squared is approximat  
ed by a chi-squared distribution.  
set.seed(03132019)  
library(lmtest)
```

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

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

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

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

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

```

a<-1
b<-1
n<-90
numsims<-5
Results<-array(NA,dim=c(numsims,3))
for(i in 1:numsims){
  x<-rnorm(n)
  ErrorVariances<-c(rep(rchisq(1,7),30),rep(rchisq(1,7),30),rep(rchisq(1,7),30))
  y<-a+b*x+rnorm(n,sd=ErrorVariances)
  plot(x,y,pch=19,type="n",ylim=c(-6,8),xlim=c(-3,3))
  points(x,y,col=c(rep(1,4),2),pch=19)
  abline(a,b,lwd=2)
  FitOLS<-lm(y~x)
  abline(FitOLS$coef,lty=2,lwd=2)

  ## We observe that the chi-squared error variances cause estimates of beta hat to diverge from the true value. Our output will display five examples:

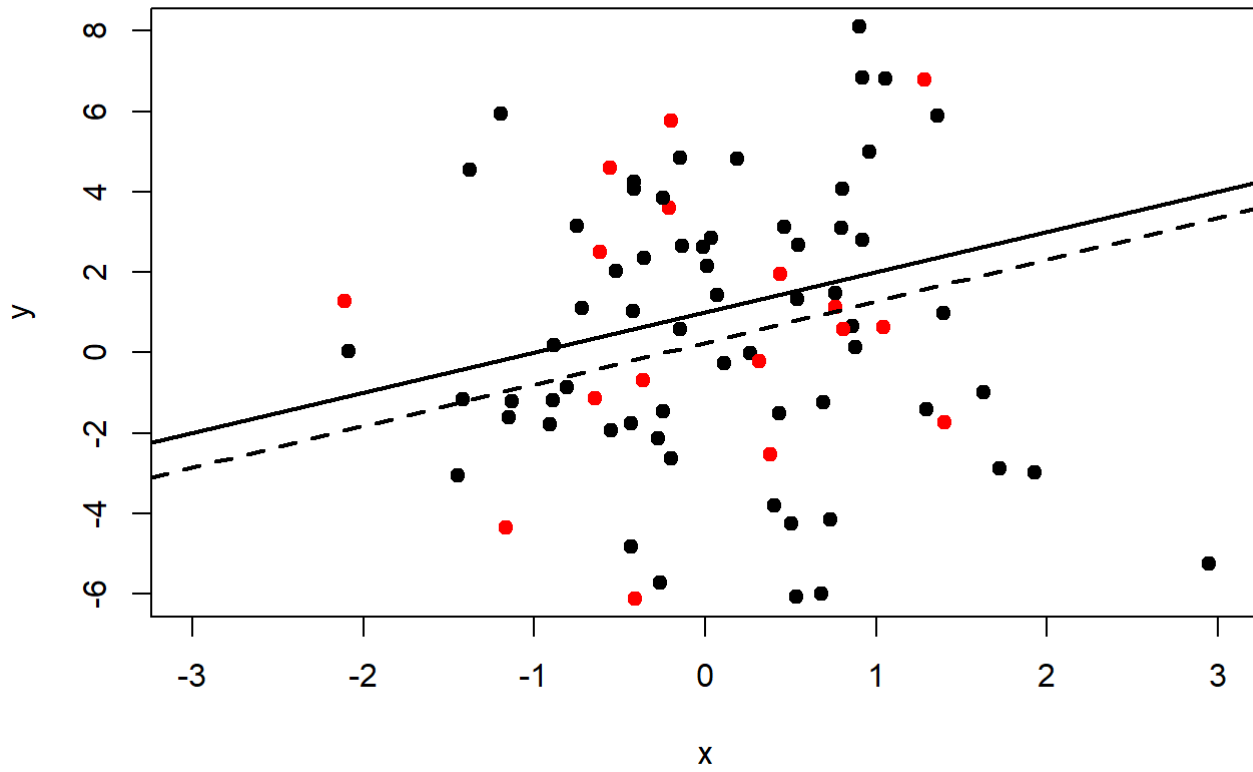
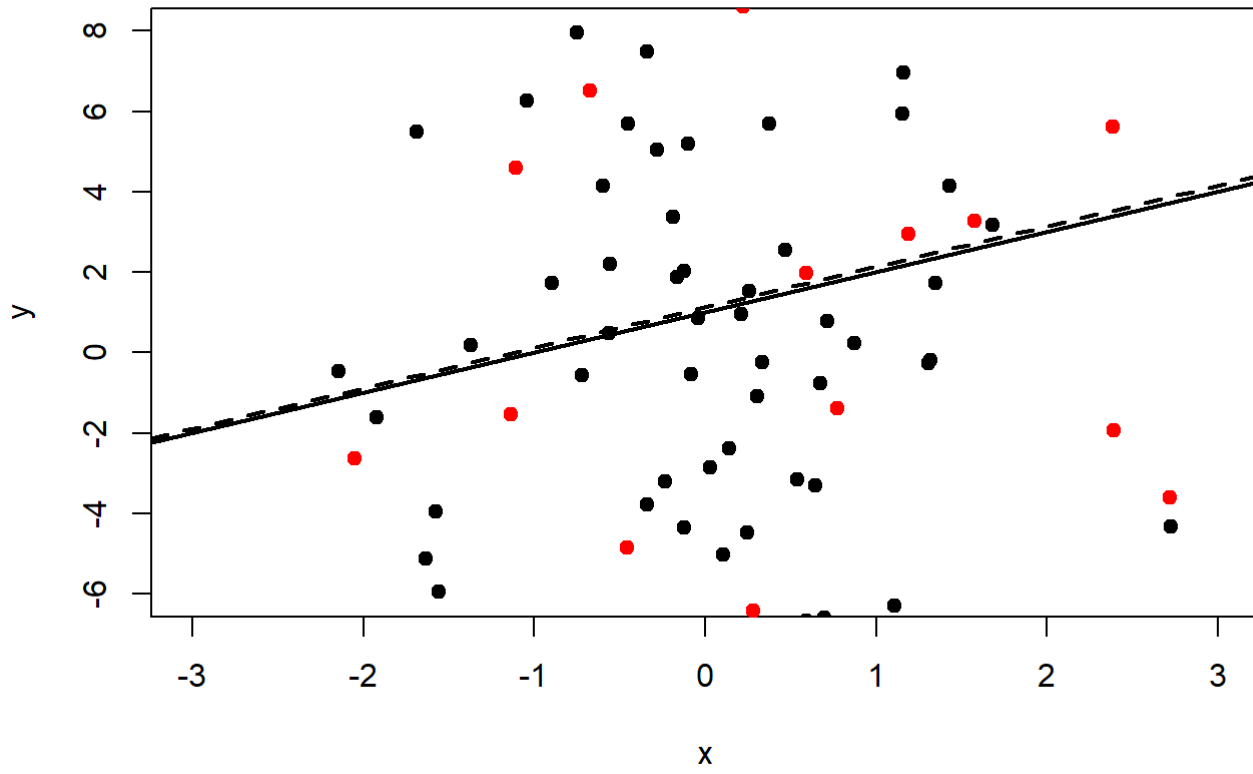
  test<-bptest(FitOLS)
  testres<-test$p.value<=0.05

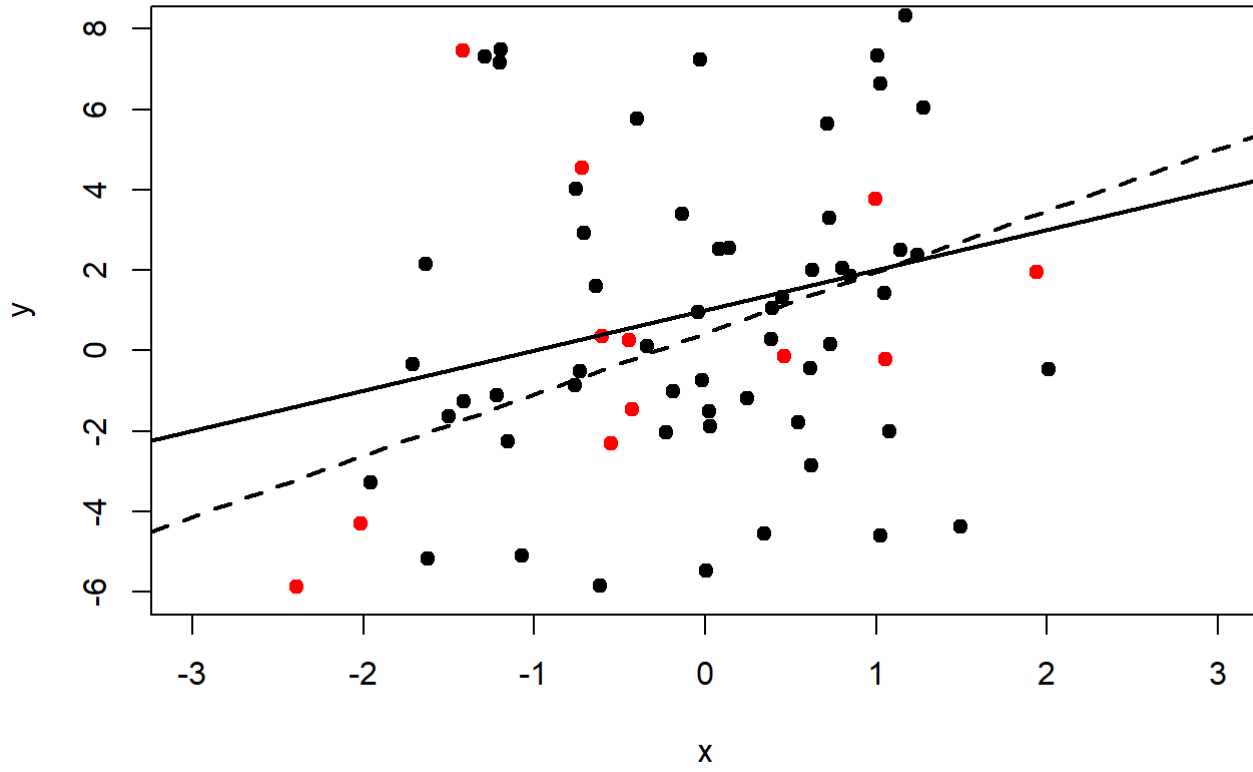
  Results[i,]<-c(FitOLS$coef,testres)
}

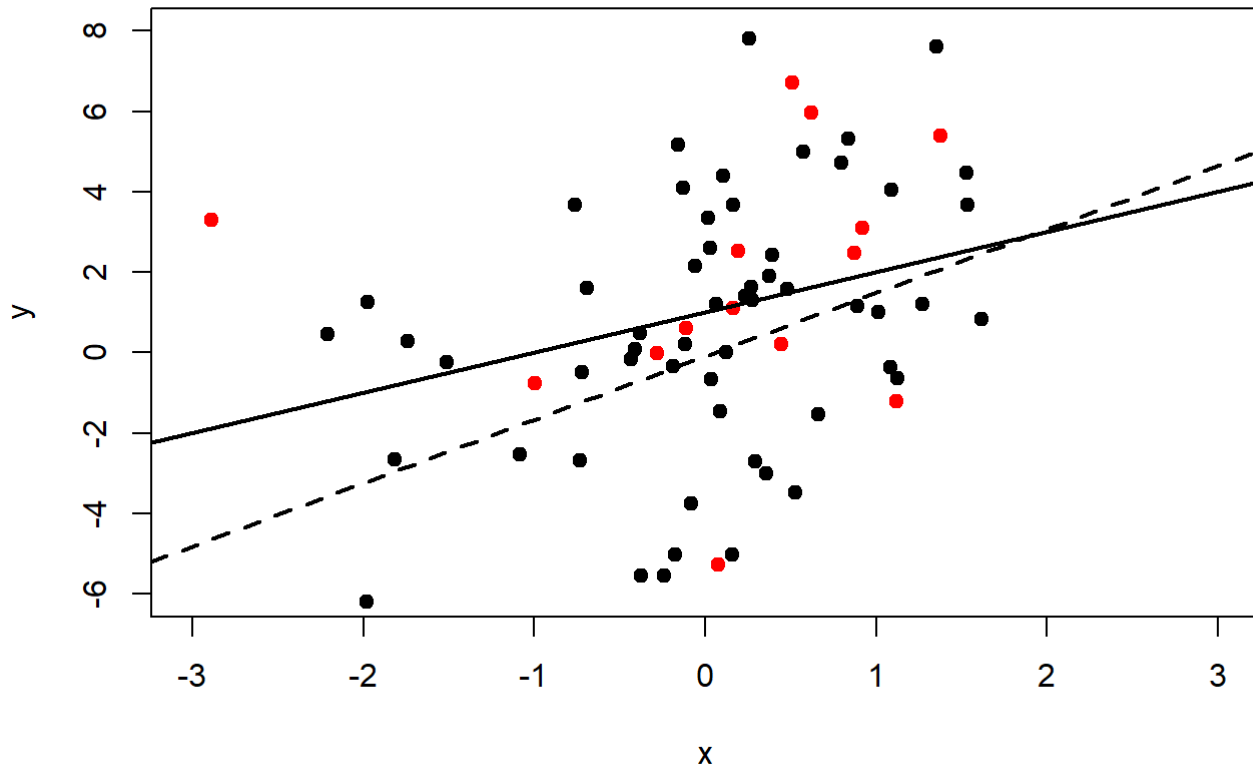
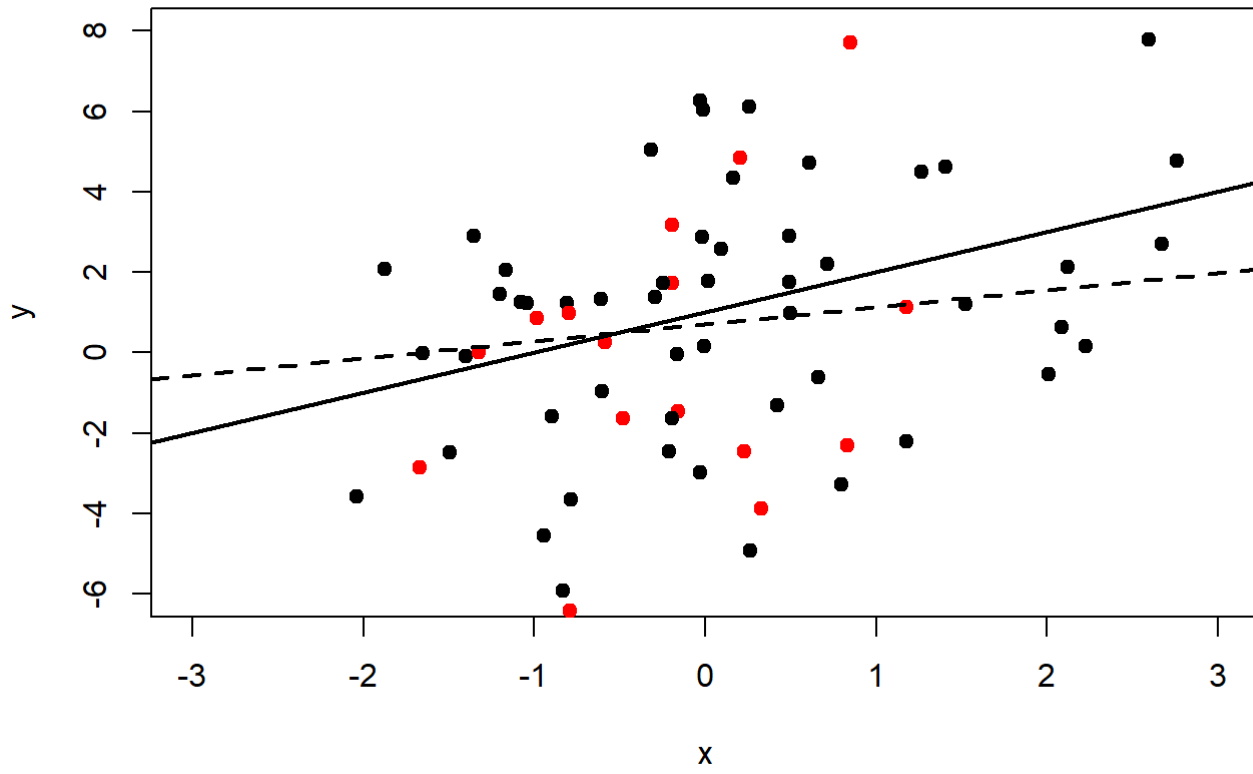
```

5 simulations are not nearly enough to diagnose whether beta hat is biased. If you ran around 1000 simulations, you'd see that the beta hats are actually really close to 1. That is because heteroskedasticity does not bias the coefficients, only the standard errors of the coefficients.

-3







8 out of 100 results of the B-P test cause us to reject the null hypothesis that error variance is not a function of the regressors.

Results

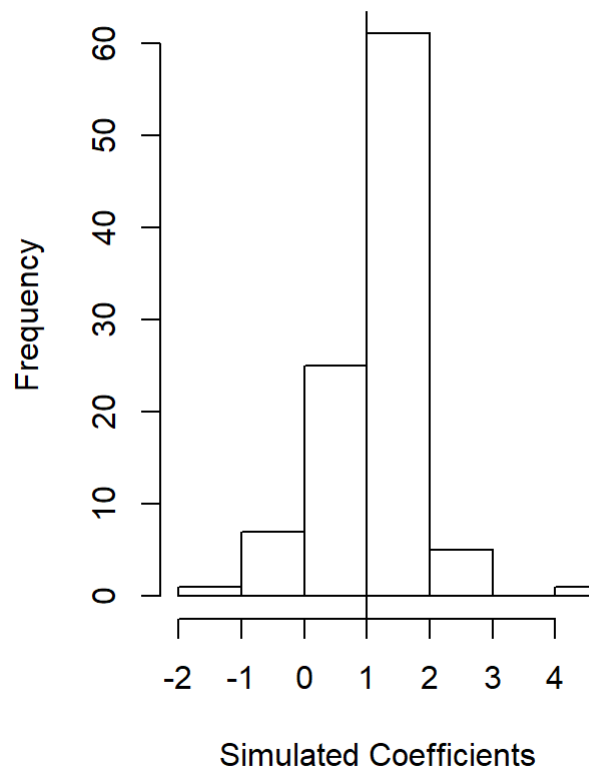
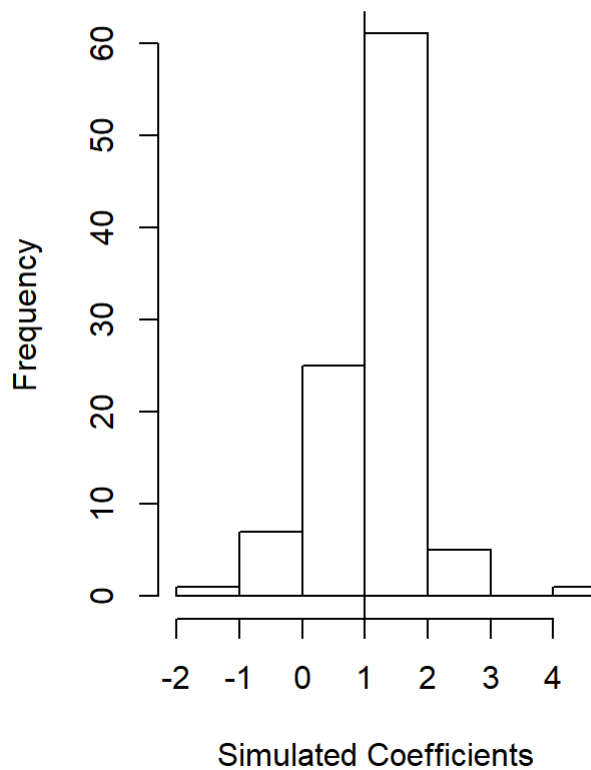
```
##          [,1]          [,2] [,3]
## [1,] -0.147919386 -0.29190691  0
## [2,]  1.215120666  0.56629061  0
## [3,]  1.263593864  1.03143694  0
## [4,]  1.868989580  0.36076084  0
## [5,]  1.498721580  0.65168356  0
## [6,]  1.207949807 -0.24104677  0
## [7,]  1.170508868 -0.11090016  0
## [8,]  1.545406208  0.71853083  1
## [9,]  0.315087854 -1.79978550  0
## [10,] -0.061077370  1.84491484  0
## [11,]  1.514961230  1.54882772  1
## [12,]  1.223350676  2.27334605  0
## [13,]  1.035139212  1.54183404  0
## [14,]  1.375209621  0.84246496  0
## [15,]  1.329625343  0.50282803  0
## [16,]  1.265093902  0.27205048  0
## [17,]  0.858692098  1.08810634  0
## [18,]  1.083011842  1.12086705  0
## [19,]  1.893825110  1.40635091  0
## [20,]  1.457966150  2.03513090  0
## [21,]  0.715566060  0.32561252  0
## [22,]  1.307504377  0.99152719  0
## [23,]  1.222209622  0.70388788  0
## [24,] -0.748967165  2.15513362  0
## [25,]  2.366828767  1.32117306  0
## [26,]  1.255673181 -0.19131812  0
## [27,]  0.750829488  2.22816586  0
## [28,] -0.500491495  2.28938156  0
## [29,]  0.690501157  0.39515867  0
## [30,]  1.185830719  0.94161154  0
## [31,]  0.700082184  1.88088762  0
## [32,]  1.482043678  1.56079175  0
## [33,]  0.917312191  0.97990141  0
## [34,]  1.520584658  1.75974719  0
## [35,] -1.292394338  1.00844808  0
## [36,]  1.743056292  0.65881064  1
## [37,]  0.763987926 -2.03872357  0
## [38,]  0.821763691 -0.50099931  0
## [39,]  1.084122770  1.37763123  0
## [40,]  1.752205883  2.69901660  0
## [41,]  1.811397782  0.34151947  0
## [42,]  1.426886710  0.88278131  0
## [43,]  1.686954218  0.67373347  0
## [44,]  2.179775646 -0.35478976  0
## [45,]  1.831324218  1.54437884  0
## [46,]  1.280582569 -0.05741035  0
## [47,]  1.561520509 -0.26520669  0
## [48,]  2.556744403  1.12012310  0
## [49,]  1.809494303  1.00679133  0
## [50,]  0.607834624  1.12446598  0
## [51,]  0.404477511  1.79660369  0
## [52,]  0.259922684  1.85567136  0
```

##	[53,]	1.118700058	0.58128227	0
##	[54,]	1.990997674	2.24525896	0
##	[55,]	1.583640866	1.64465398	0
##	[56,]	0.404971429	0.72399146	0
##	[57,]	0.992926269	1.38894926	0
##	[58,]	0.724085476	1.93053172	0
##	[59,]	1.010519519	1.09734802	0
##	[60,]	0.007080168	0.63904711	0
##	[61,]	1.936147000	-0.15614416	0
##	[62,]	0.171003049	0.61051858	0
##	[63,]	2.777980986	-0.31569234	0
##	[64,]	0.308273026	0.76048433	0
##	[65,]	1.578269319	-1.20318946	0
##	[66,]	1.688739875	0.67222615	0
##	[67,]	1.023976466	1.70045663	0
##	[68,]	0.712745499	0.77826344	0
##	[69,]	-0.210905261	2.60333797	0
##	[70,]	1.181339066	1.80121690	0
##	[71,]	0.735641307	0.45731996	0
##	[72,]	1.694162253	0.57074063	0
##	[73,]	1.168054592	-0.53095465	0
##	[74,]	1.392255089	-0.02182071	0
##	[75,]	1.334326095	0.98349465	0
##	[76,]	1.580757288	-0.61370023	0
##	[77,]	2.422154234	0.74920893	0
##	[78,]	1.348123664	1.13168289	0
##	[79,]	0.163737446	-0.54081564	1
##	[80,]	-0.402756219	1.17926828	0
##	[81,]	1.049819332	1.42613236	0
##	[82,]	-0.133556136	1.00722831	0
##	[83,]	0.611530537	1.03301576	0
##	[84,]	1.658958643	0.77423146	0
##	[85,]	1.141698357	2.41372220	0
##	[86,]	1.239111892	2.41781474	0
##	[87,]	1.787894887	-0.61041161	0
##	[88,]	1.331388830	0.23888997	0
##	[89,]	1.480328067	1.03830543	0
##	[90,]	1.059898768	0.98134451	0
##	[91,]	1.746043252	1.33748461	0
##	[92,]	1.586453856	3.26926356	0
##	[93,]	0.627546016	1.28771200	0
##	[94,]	1.139226088	1.42555961	0
##	[95,]	0.759477351	1.72089091	0
##	[96,]	0.524011518	2.05505979	0
##	[97,]	1.248836761	1.15281438	0
##	[98,]	1.727462438	2.26209467	0
##	[99,]	1.094872203	1.98363453	0
##	[100,]	4.376282059	0.50941974	0


```

par(mfrow=c(1,2))
xmin<-min(Results)
xmax<-max(Results)
TITLE<-c("OLS", "Just 4 Obs")
XLAB<- "Simulated Coefficients"
  hist(Results[,1],xlim=c(xmin,xmax),main=TITLE[i],xlab=XLAB)
  abline(v=a)
  hist(Results[,1],xlim=c(xmin,xmax),main=TITLE[i],xlab=XLAB)
  abline(v=b)

```



```
mean(Results[,3])
```

```
## [1] 0.04
```

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

```
## Now we will compare the robust standard errors to the ordinary OLS errors. We see that the robust standard errors appear to be smaller on average, with a smaller mean.
head(Results)
```

```
##           [,1]      [,2] [,3]
## [1,] -0.1479194 -0.2919069  0
## [2,]  1.2151207  0.5662906  0
## [3,]  1.2635939  1.0314369  0
## [4,]  1.8689896  0.3607608  0
## [5,]  1.4987216  0.6516836  0
## [6,]  1.2079498 -0.2410468  0
```

```
head(Results2)
```

```
##           [,1]      [,2] [,3]
## [1,] -0.7696980 -0.03354941  0
## [2,]  1.6615873  2.52743755  0
## [3,]  0.5664358  0.52513544  0
## [4,]  1.2906575  0.89393124  0
## [5,]  1.4822079  0.78079184  0
## [6,]  0.5888398  1.98817414  0
```

```
mean(Results)
```

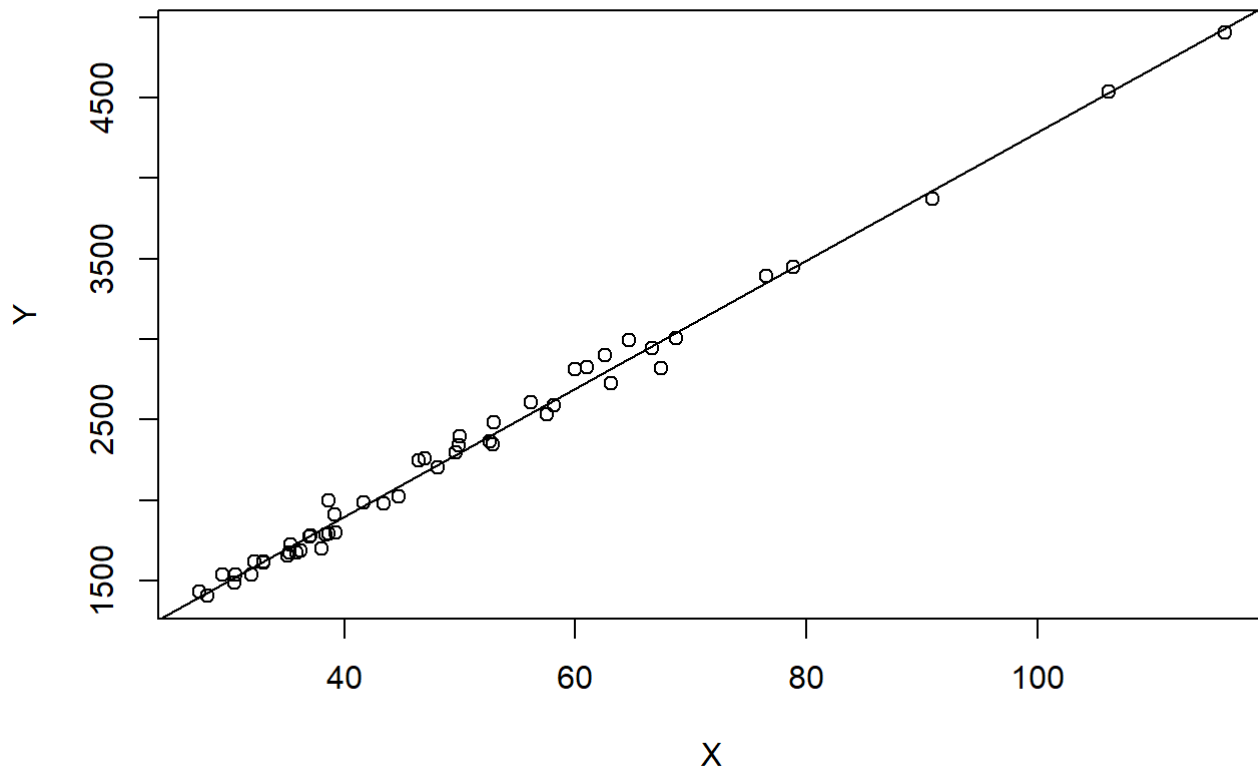
```
## [1] 0.7091004
```

```
mean(Results2)
```

```
## [1] 0.6368702
```

Question 2: Measurement Error Simulation

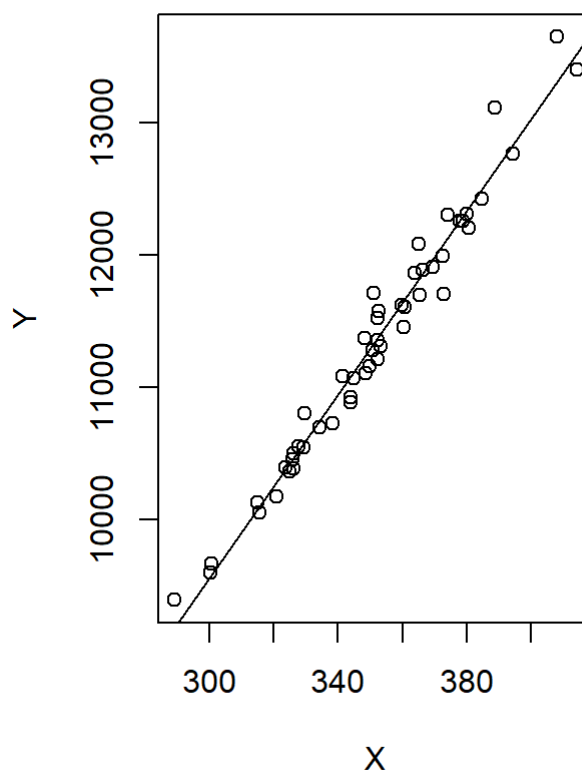
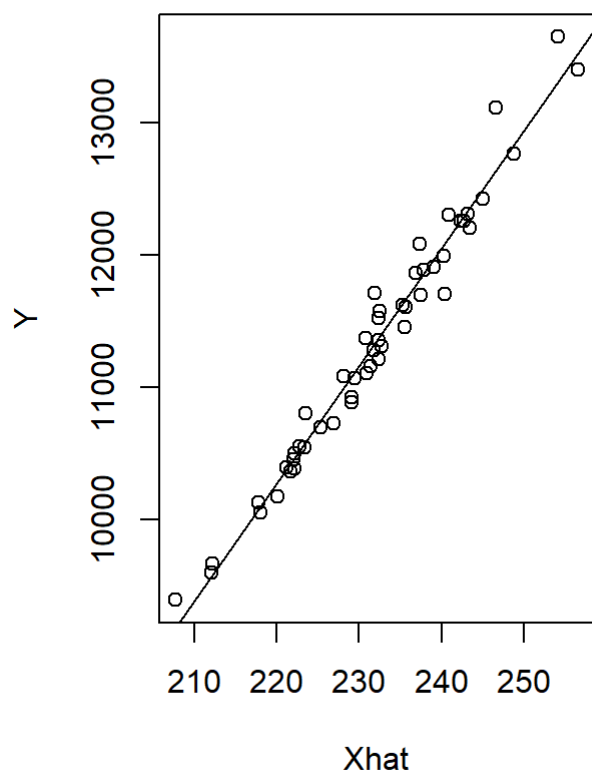
```
n <- 50
Z <- rpois(n, 20)
true.beta <- 30
errors <- rexp(n, 3)
## When X is measured with errors, the data will deviate from the true regression line:
X <- 30 + 50*errors + rnorm(n, 0, 3)
Y <- 600 + true.beta*X + 500*errors + rnorm(n, 0, 50)
plot(X,Y)
abline(lm(Y~X))
```



```
## We will now generate a new variable Z correlated with the true X*:
Z <- true.beta+rpois(n, 200)
X <- 30 + 1.3*Z + 50*errors + rnorm(n, 0, 3)
Y <- 600 + true.beta*X + 500*errors + rnorm(n, 0, 50)
Xhat<-(lm(Z~X)$fitted.values)

## Plotting X-hat against Y, we observe modest variation:
par(mfrow=c(1,2))
plot(Xhat,Y)
abline(lm(Y~Xhat))
plot(X,Y)
abline(lm(Y~X))
```

Rather than plotting each regression line, it might be easier to save just the betas and plot them



```
## We will now regress Y on Xhat and compare to a regression of Y on X:
with.error<-(lm(Y ~ Xhat)$fitted.values)
without.error<-(lm(Y ~ X)$fitted.values)
head(with.error-without.error)
```

```
##           1           2           3           4           5
## 1.637090e-11 -3.637979e-12 -1.818989e-12 0.000000e+00 0.000000e+00
##           6
## 0.000000e+00
```

```
## We observe very small differences. We also observe (as we learned two homeworks ago)
## that random errors tend to flatten the slope.
```

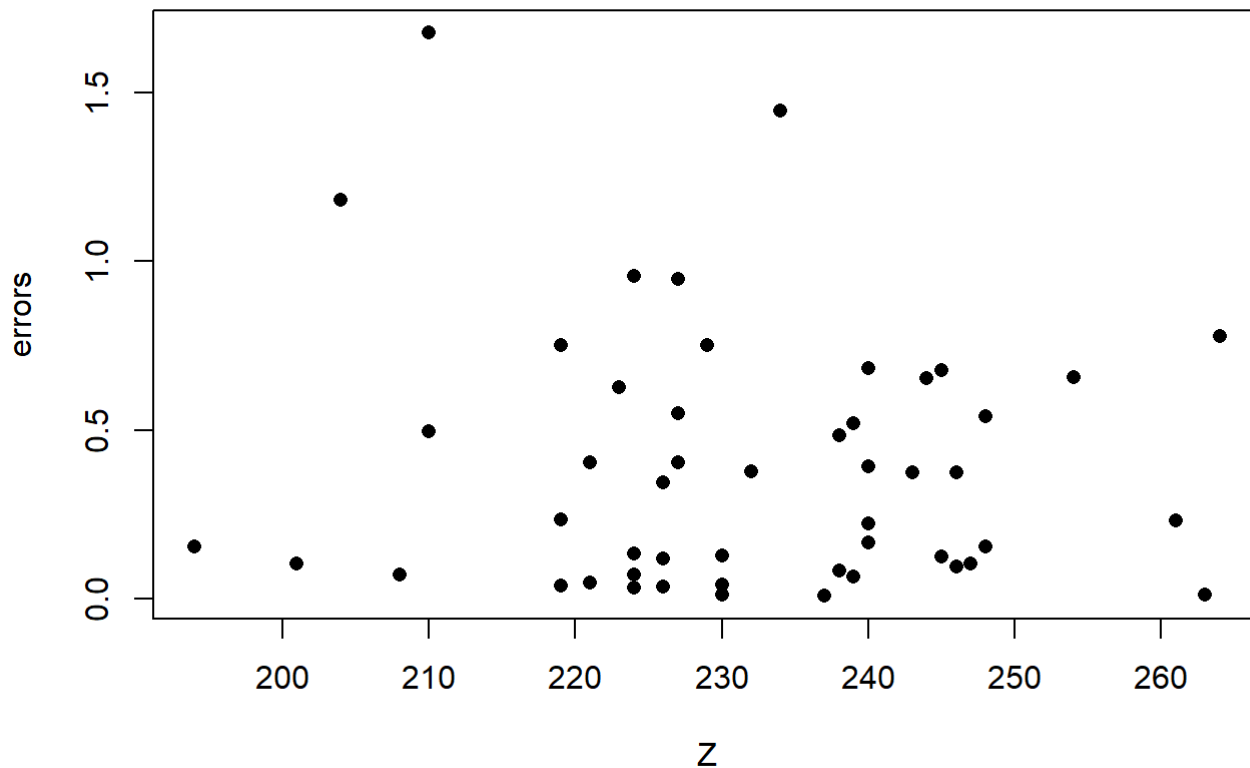
```
## We will now compare slopes and estimated standard errors:
lm(Y ~ Xhat)$coefficients
```

```
## (Intercept)      Xhat
## -9243.46488    88.73463
```

```
lm(Y ~ X)$coefficients
```

```
## (Intercept)      X
## -849.4330      34.6999
```

```
## The slope of Xhat is larger than the slope of X:
plot(Z, errors, pch=16)
```



Question 3: Presentation

Prevailing theory holds that strong institutions prevent corruption (Shleifer and Vishny 1993). We might therefore expect well-trained civil servants and a reliable legal system to reduce corruption.

Hypotheses

>-H0: High bureaucratic quality and rule of law do not affect corruption.

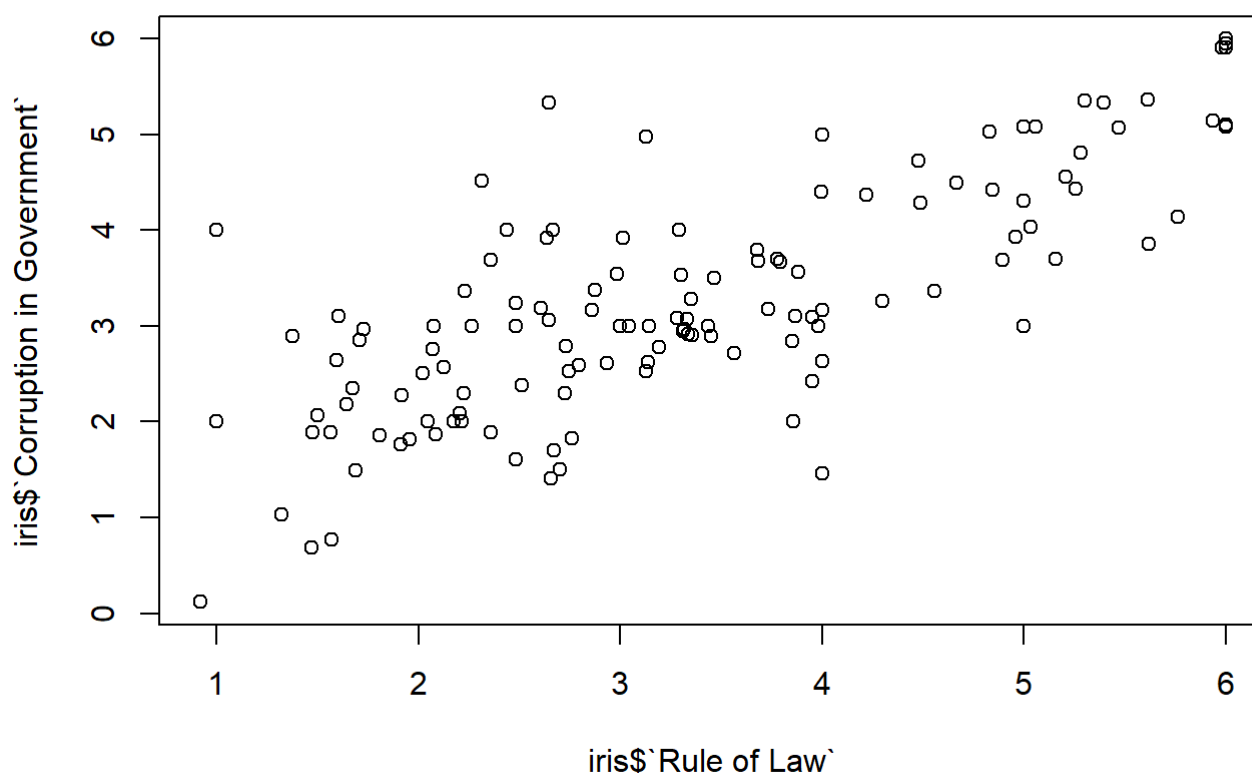
>-HA: High bureaucratic quality and rule of law reduce corruption.

Data

The IRIS dataset was developed in 1993 by Philip Keefer and Stephen Knack in their work for the Institutional Reform and the Informal Sector (IRIS) research and advisory center at the University of Maryland. The dataset is based on data obtained from the International Country Risk Guide (ICRG), and contains scores for six "political risk variables," comprising corruption, the rule of law, bureaucratic quality, ethnic tensions, repudiation of government contracts, and expropriation risk.

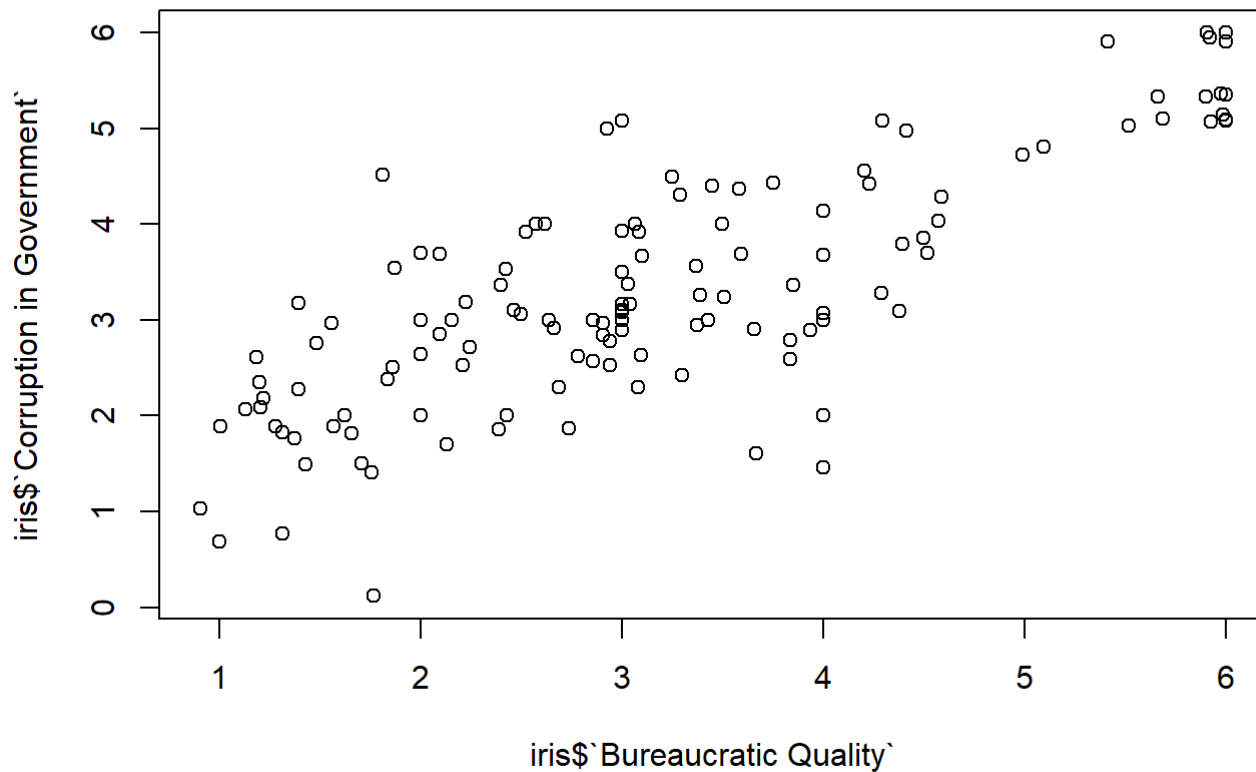
IRIS Dataset

```
iris<-read.csv("IRIS.csv", header=TRUE)
colnames(iris)[colnames(iris)=="i..country"] <- "Country"
colnames(iris)[colnames(iris)=="corruption.in.government"] <- "Corruption in Government"
colnames(iris)[colnames(iris)=="rule.of.law..law.and.order.tradition."] <- "Rule of Law"
colnames(iris)[colnames(iris)=="bureaucratic.quality"] <- "Bureaucratic Quality"
iris<-aggregate(.~Country,iris,mean)
plot(iris$`Rule of Law`,iris$`Corruption in Government`)
```



IRIS Dataset

```
plot(iris$`Bureaucratic Quality`,iris$`Corruption in Government`)
```



The Model

```
Corruption in Government ~ Bureaucratic Quality + Rule of Law + (Bureaucratic Quality :  
Rule of Law)
```

Results

```
reg1<-lm(`Corruption in Government`~`Bureaucratic Quality`+`Rule of Law`+(`Bureaucratic  
Quality`:`Rule of Law`),data=iris)  
summary(reg1)
```

```
##
## Call:
## lm(formula = `Corruption in Government` ~ `Bureaucratic Quality` +
##     `Rule of Law` + (`Bureaucratic Quality`:`Rule of Law`), data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42106 -0.47090  0.00846  0.41286  2.18607
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.84889    0.39211    2.165  0.03221
## `Bureaucratic Quality`
##      0.37125    0.13374    2.776  0.00631
## `Rule of Law`
##      0.32131    0.12730    2.524  0.01280
## `Bureaucratic Quality`:`Rule of Law`
##      0.01619    0.02958    0.547  0.58502
##
## (Intercept)          *
## `Bureaucratic Quality`
##                   **
## `Rule of Law`
##                   *
## `Bureaucratic Quality`:`Rule of Law`
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6997 on 131 degrees of freedom
## Multiple R-squared:  0.7222, Adjusted R-squared:  0.7159
## F-statistic: 113.5 on 3 and 131 DF,  p-value: < 2.2e-16
```

Robust Standard Errors

```
reg2<-felm(`Corruption in Government`~`Bureaucratic Quality`+`Rule of Law`+(`Bureaucratic
Quality`:`Rule of Law`),data=iris)
summary(reg2)
```



```
##
## Call:
##   felm(formula = `Corruption in Government` ~ `Bureaucratic Quality` + `Rule of
Law` + (`Bureaucratic Quality`:`Rule of Law`), data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42106 -0.47090  0.00846  0.41286  2.18607
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.84889    0.39211    2.165  0.03221
## `Bureaucratic Quality`
##      0.37125    0.13374    2.776  0.00631
## `Rule of Law`
##      0.32131    0.12730    2.524  0.01280
## `Bureaucratic Quality`:`Rule of Law`
##      0.01619    0.02958    0.547  0.58502
##
## (Intercept)          *
## `Bureaucratic Quality`
##          **
## `Rule of Law`
##          *
## `Bureaucratic Quality`:`Rule of Law`
##          *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6997 on 131 degrees of freedom
## Multiple R-squared(full model): 0.7222   Adjusted R-squared: 0.7159
## Multiple R-squared(proj model): 0.7222   Adjusted R-squared: 0.7159
## F-statistic(full model):113.5 on 3 and 131 DF, p-value: < 2.2e-16
## F-statistic(proj model): 113.5 on 3 and 131 DF, p-value: < 2.2e-16
```

The standard errors reported in this table have not been adjusted.

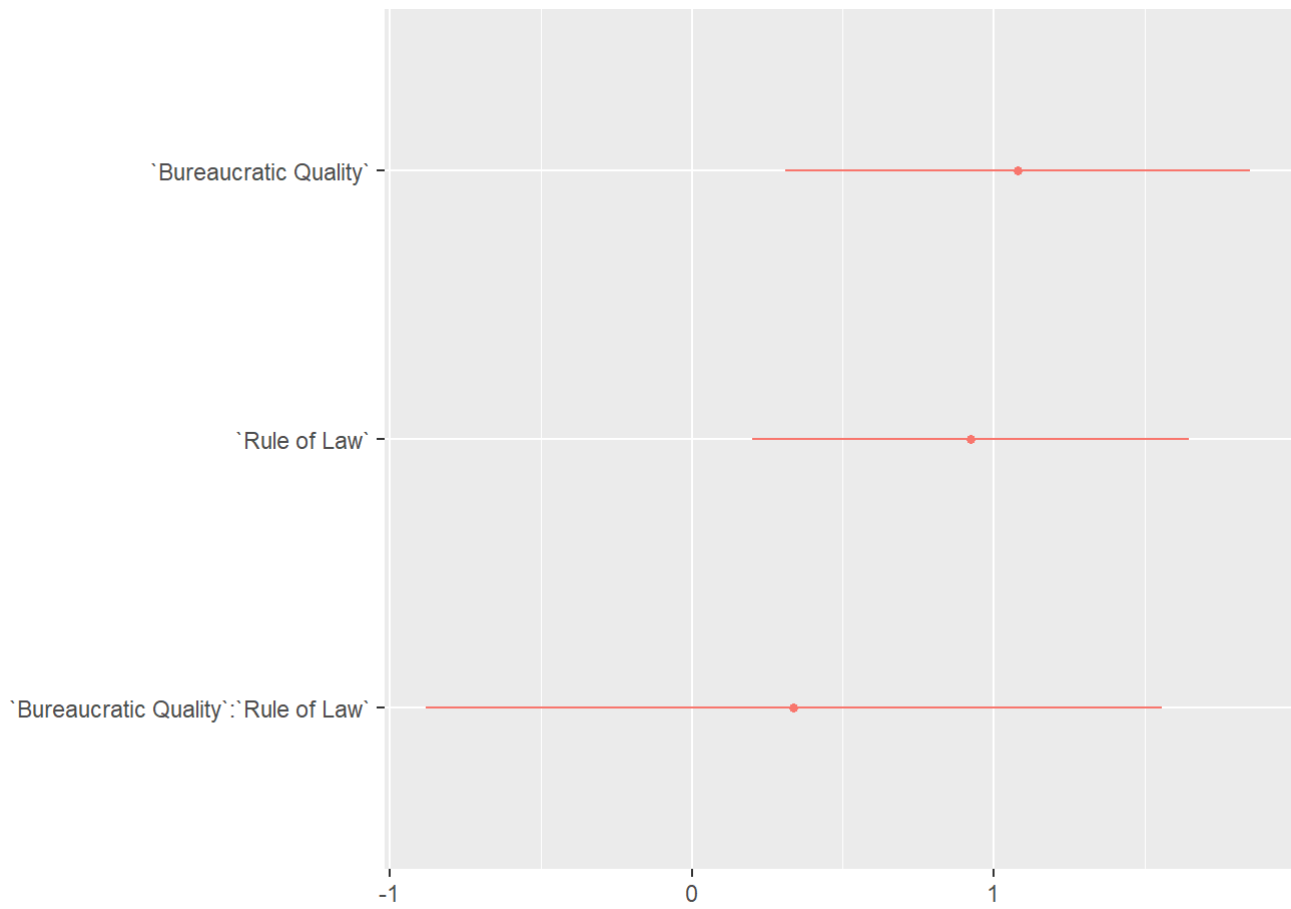
In order to get the robust standard errors, you have to ask for them:

reg2\$rse

-1

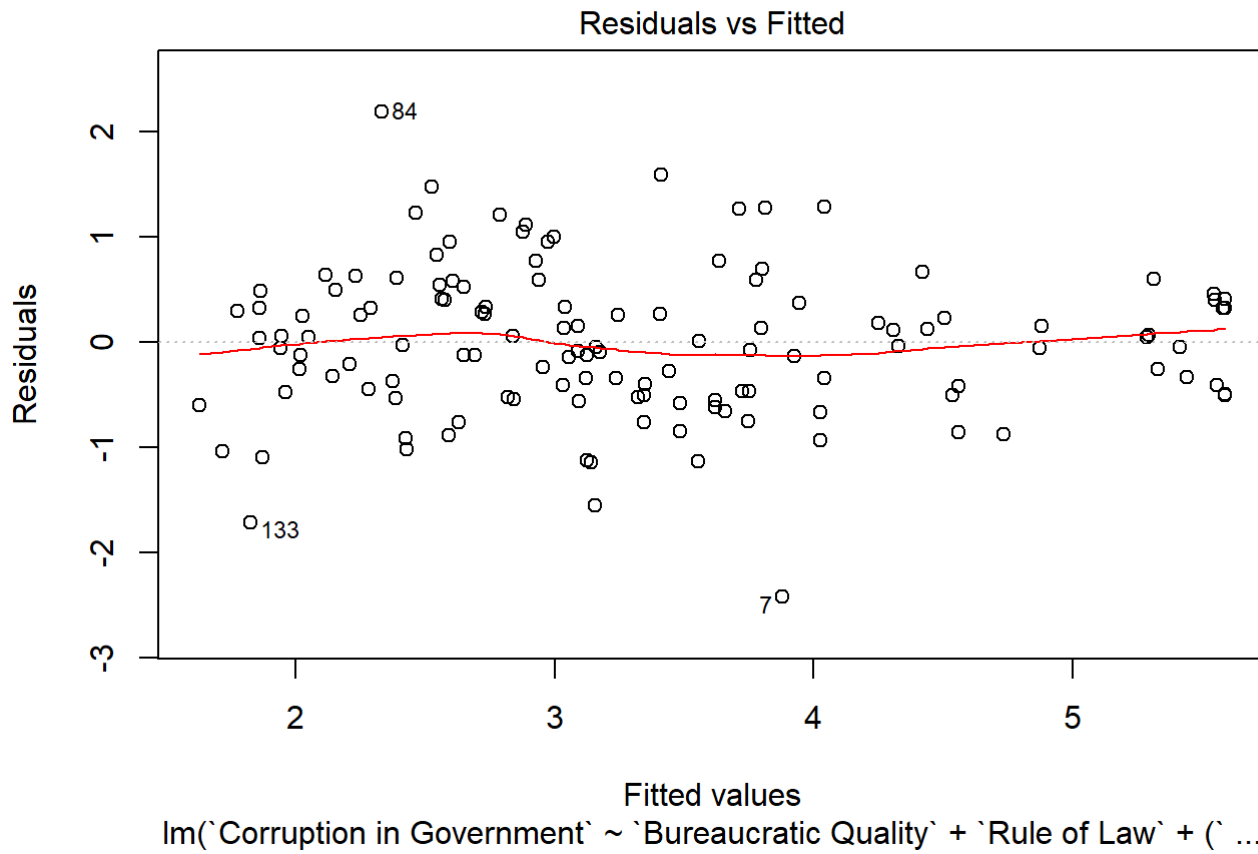
Diagnostics

```
dwplot(reg1)
```



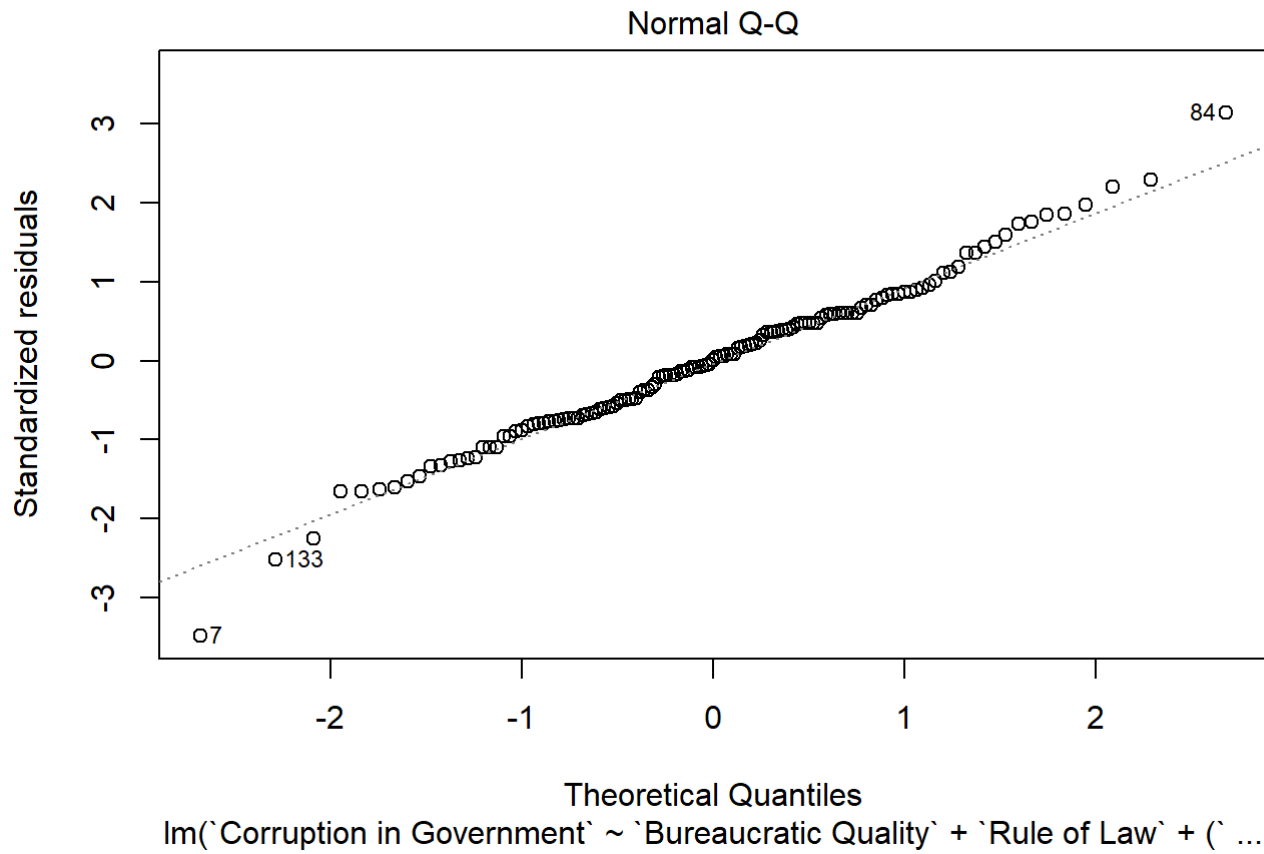
Diagnostics

```
plot(reg1, which=1)
```



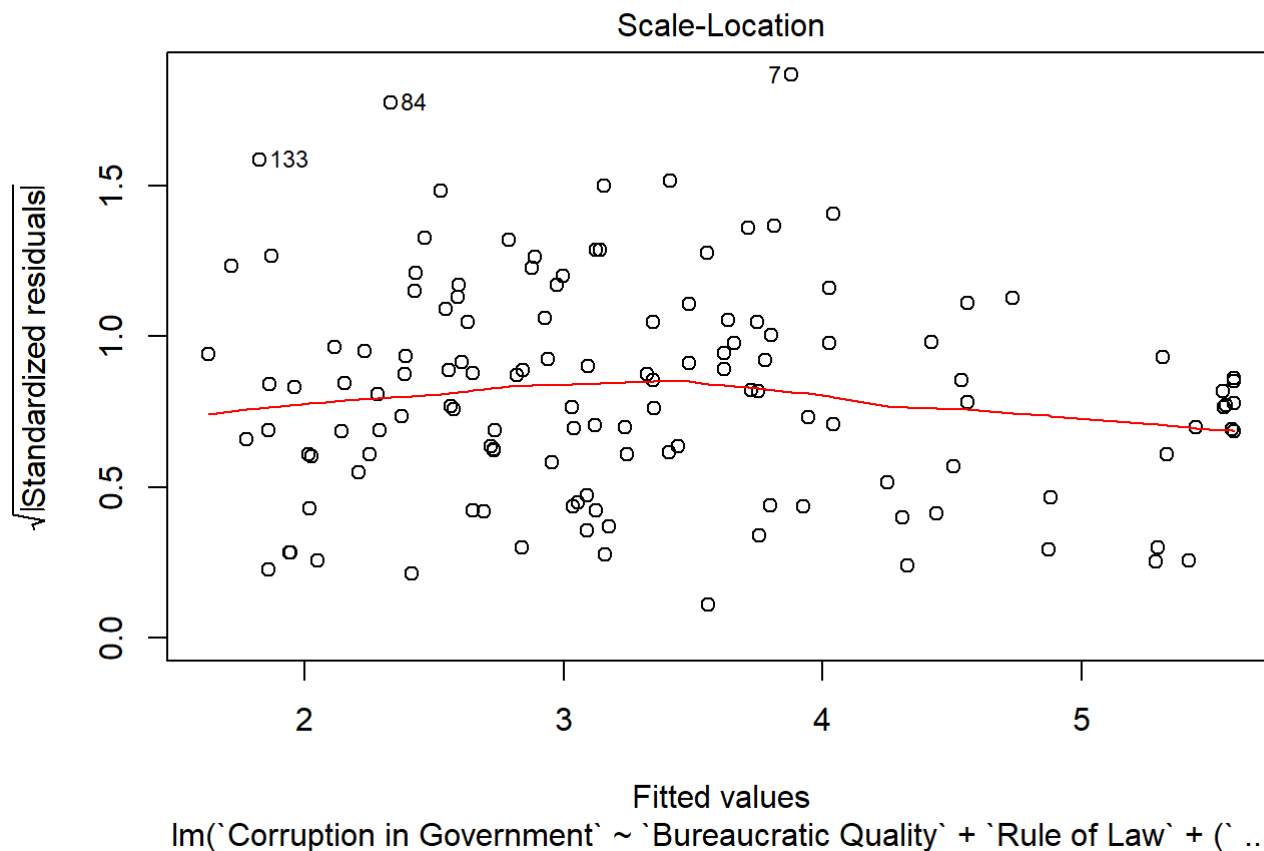
Diagnostics

```
plot(reg1, which=2)
```



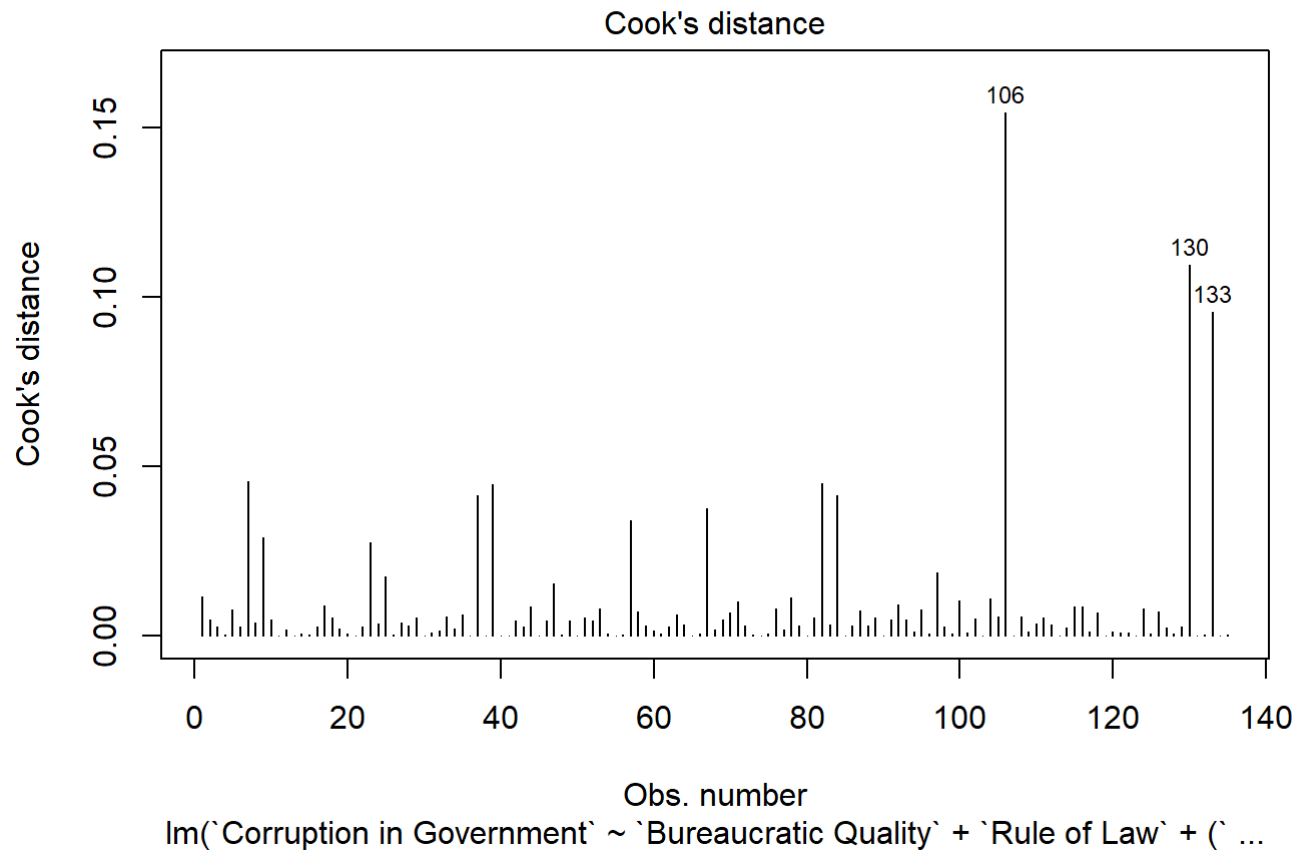
Diagnostics

```
plot(reg1, which=3)
```



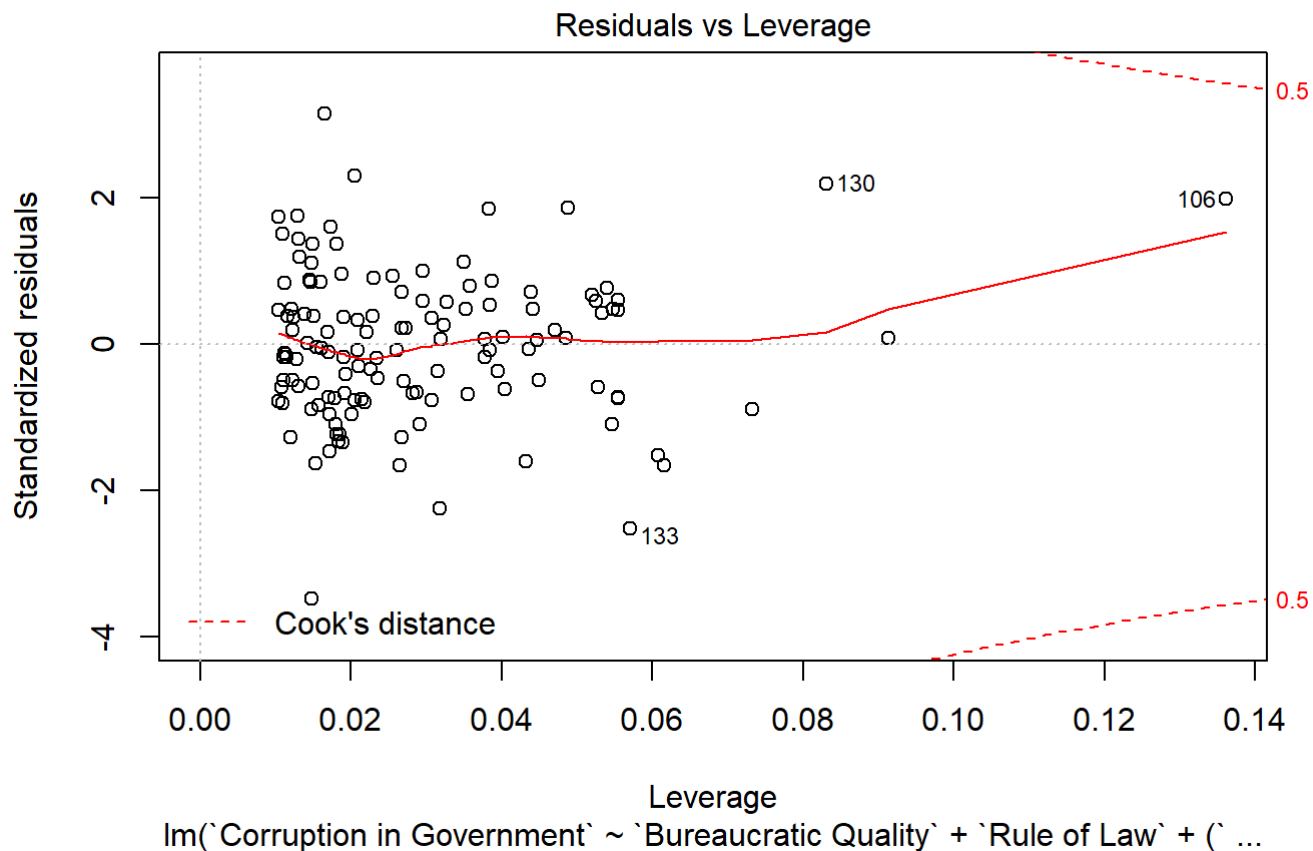
Diagnostics

```
plot(reg1, which=4)
```



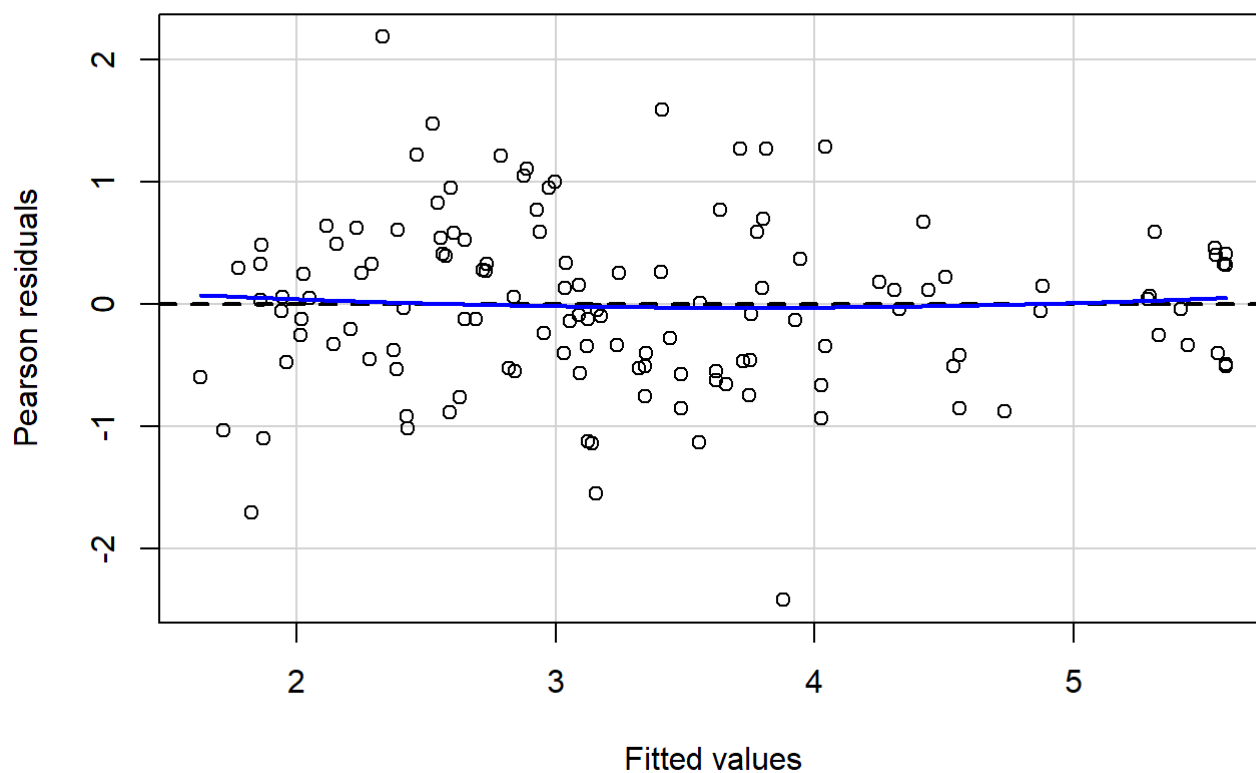
Diagnostics

```
plot(reg1, which=5)
```



Diagnostics

```
residualPlots(reg1)
```



```
##          Test stat Pr(>|Test stat|)
## Tukey test    2.8438      0.004457 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Works Cited

Andrei Shleifer and Robert W. Vishny The Quarterly Journal of Economics Vol. 108, No. 3 (Aug., 1993), pp. 599-617