Week 3 Code

Ryan Martin

January 17, 2018

3.6

In exercise 2.9 We considered a motel that had discovered that a defective product was used during construction. It took seven months to correct the defects, during which approximately 14 rooms in the 100-unit motel were taken out of service for one month at a time.The data are in `motel.dat`.

a

In the linear regression model $MOTEL\_PCT = \beta_1 + \beta_2 COMP\_PCT + e$, test the null hypothesis $H_0 : \beta_2 \leq 0$ against the alternative hypothesis $H_1 : \beta_2 > 0$ at the $\alpha = .01$ level of significance. Discuss your conclusion. Include in your answer a sketch of the rejection region and a calculation of the p-value.

Solution:

Note, it's a one-sided test and we are rejecting for large, positive β_2. This is a t-test as usual. Test statistic is $t_{test} = \frac{\hat{\beta}_2}{se(\hat{\beta}_2)}$. Compare with the critical value $t_{.99,N-2}$

```
my_wd <- "C:/Users/ryanj/Dropbox/TA/Econ 103/Winter 2018/Data/s4poe_statadata"
my_file <- paste(my_wd, "motel.dta", sep = "/")
library(haven)
dat <- read_stata(my_file)
#View(dat)
reg_out <- lm(data=dat, motel_pct~ comp_pct)
s <- summary(reg_out)
s
```

```
##
## Call:
## lm(formula = motel_pct ~ comp_pct, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.876  -4.909  -1.193   5.312  26.818
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.4000    12.9069   1.658 0.110889
## comp_pct      0.8646     0.2027   4.265 0.000291 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.02 on 23 degrees of freedom
## Multiple R-squared:  0.4417, Adjusted R-squared:  0.4174
## F-statistic: 18.19 on 1 and 23 DF,  p-value: 0.0002906
```

```
t_stat <- coef(s)[2,1]/coef(s)[2,2]
t_stat
```

```
## [1] 4.26536
```

```
t.crit <- qt(.99, nrow(dat)-2)
t.crit
```

```
## [1] 2.499867
```

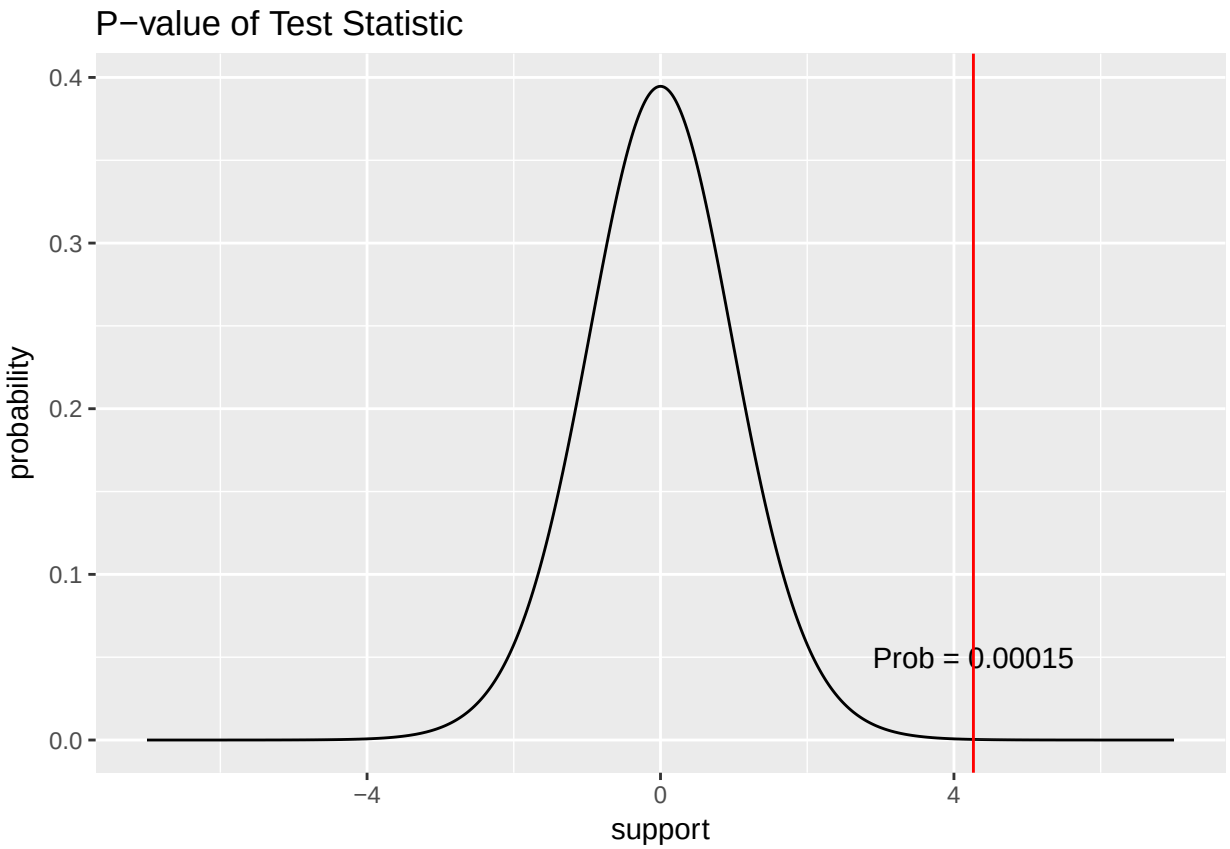
```
t_stat> t.crit #TRUE means reject null
```

```
## [1] TRUE
```

```
########plotting p-value
p_value <- 1 - pt(t_stat, nrow(dat)-2)
p_value
```

```
## [1] 0.0001453107
```

```
#Then use this new data.frame with geom_polygon
support = -700:700/100
plot_data <- as.data.frame(cbind(support,
                  probability = dt(support, nrow(dat)-2)))
shade <- as.data.frame(rbind(c(t_stat,0),
               subset(plot_data, support > t_stat),
               c(plot_data[nrow(plot_data), "support"], 0)))
names(shade) <- c("x","y")
library(ggplot2)
ggplot(data = plot_data, aes(x = support, y= probability))  +
  geom_line() +
  annotate("text", x = t_stat, y = .05, label =
             paste("Prob = ", round(p_value,5), sep = "")) +
  geom_polygon(data = shade, aes(x,y )) +
  ggtitle("P-value of Test Statistic")  +
  geom_vline(xintercept = t_stat, col = 'red'  )
```



b

Consider a linear regression with $y = MOTEL\_PCT$ and $x = RELPRICE$, which is the ratio of the price per room charged by the motel in question relative to its competitors. Test the null hypothesis that there is no relationship between these variables against the alternative that there is an inverse relationship between them, at the $\alpha = .01$ level of significance. Discuss your conclusion. Include in your answer a sketch of the rejection region and a calculation of the p-value. In this exercise follow and **show** all the test procedure steps suggested in Chapter 3.4

Solution: This question could be worded better. An inverse relationship between a and b is $a = 1/b$. An inverse correlation is just a negative correlation. They don't define this in the book, but if you search through the online version for "inverse", in section 4.4.1, they have an example where they use inverse to describe negative correlation. So, that must be what they want. Regress

$$y = \beta_1 + x\beta_2 + e.$$

and test $H_0 : \beta_2 \geq 0$ vs $H_1 : \beta_2 < 0$. Now the problem is just the opposite tail version of a. Our test stat is again $\hat{\beta}_2/se(\hat{\beta}_2)$ and it's T distributed as in a.

We can just copy and paste section a's code after modifying the regression equation and the test side. Now our critical t value is $t_{.01,N-2}$. Note that the p-value changes to the area to the left rather than the area to the right.

```
attach(dat)
reg_out <- lm(motel_pct~ relprice)
```

```r
s <- summary(reg_out)
s
```

```
##
## Call:
## lm(formula = motel_pct ~ relprice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.779  -6.181   4.003   8.842  21.760
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   166.66      43.57   3.825 0.000868 ***
## relprice     -122.12      58.35  -2.093 0.047589 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.52 on 23 degrees of freedom
## Multiple R-squared:   0.16,  Adjusted R-squared:  0.1234
## F-statistic:  4.38 on 1 and 23 DF,  p-value: 0.04759
```

```r
t_stat <- coef(s)[2,1]/coef(s)[2,2]
t_stat
```

```
## [1] -2.092854
```

```r
t.crit <- qt(.01, nrow(dat)-2)
t.crit
```

```
## [1] -2.499867
```

```r
t_stat< t.crit #FALSE means fail to reject null
```

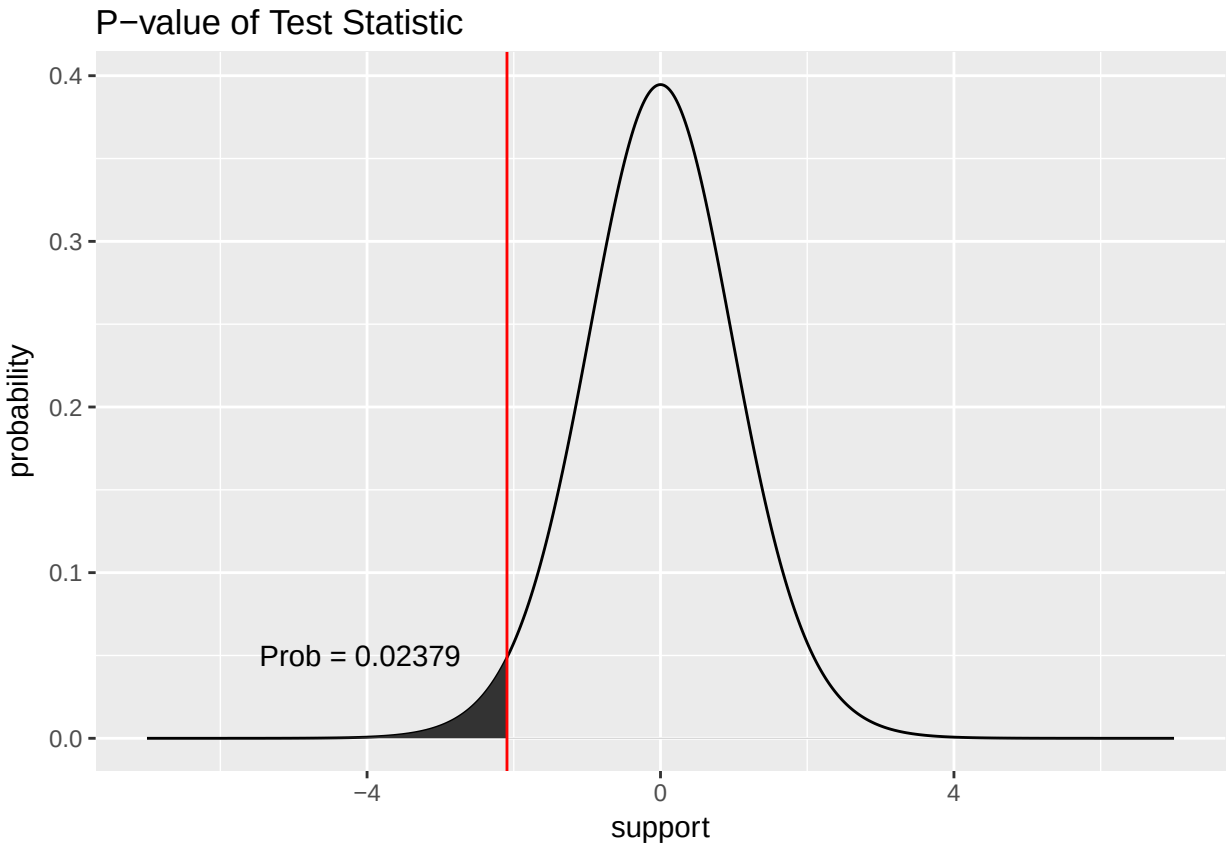
```
## [1] FALSE
```

```r
########plotting p-value
p_value <-  pt(t_stat, nrow(dat)-2) #now p_value just to the left!
      #note that this changed.
p_value
```

```
## [1] 0.02379463
```

```r
#Then use this new data.frame with geom_polygon
support = -700:700/100
plot_data <- as.data.frame(cbind(support,
                  probability = dt(support, nrow(dat)-2)))
#note shade order changed
shade <- as.data.frame(rbind(
               subset(plot_data, support < t_stat),
               c(t_stat,0),
               c(plot_data[nrow(plot_data), "support"], 0)))
names(shade) <- c("x","y")
library(ggplot2)
ggplot(data = plot_data, aes(x = support, y= probability))  +
  geom_line() +
  annotate("text", x = t_stat - 2, y = .05, label =
             paste("Prob = ", round(p_value,5), sep = "")) +
```

```
  geom_polygon(data = shade, aes(x,y )) +
  ggtitle("P-value of Test Statistic")  +
  geom_vline(xintercept = t_stat, col = 'red'  )
```



c

Consider the linear regression $MOTEL\_PCT = \delta_1 + \delta_2 REPAIR + e$, where $REPAIR$ is an indicator variable taking the value 1 during the repair period and 0 otherwise. Test the null hypothesis $H_0 : \delta_2 \geq 0$ against the alternative hypothesis $H_1 : \delta_2 < 0$ at the $\alpha = .05$ significance level. Explain the logic behind stating the null and alternative hypotheses in this way. Discuss your conclusions.

Solution: This one is just like b! reuse the code. You can see already how having a stockpile of code can make future analysis very useful! Even within the same problem set I can reuse code (that is written sufficiently generally), with no more than slight modificatoin, many times!

```
reg_out <- lm(motel_pct ~repair)

s <- summary(reg_out)
s
```

```
##
## Call:
## lm(formula = motel_pct ~ repair)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -26.91 -10.55  -0.55  10.99  16.85
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   79.350      3.154  25.158   <2e-16 ***
## repair       -13.236      5.961  -2.221   0.0365 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 23 degrees of freedom
## Multiple R-squared:  0.1765, Adjusted R-squared:  0.1407
## F-statistic: 4.931 on 1 and 23 DF,  p-value: 0.03651
```

```
t_stat <- coef(s)[2,1]/coef(s)[2,2]
t_stat
```

```
## [1] -2.220528
```

```
t.crit <- qt(.05, nrow(dat)-2) #change to .05 for c
t.crit
```

```
## [1] -1.713872
```

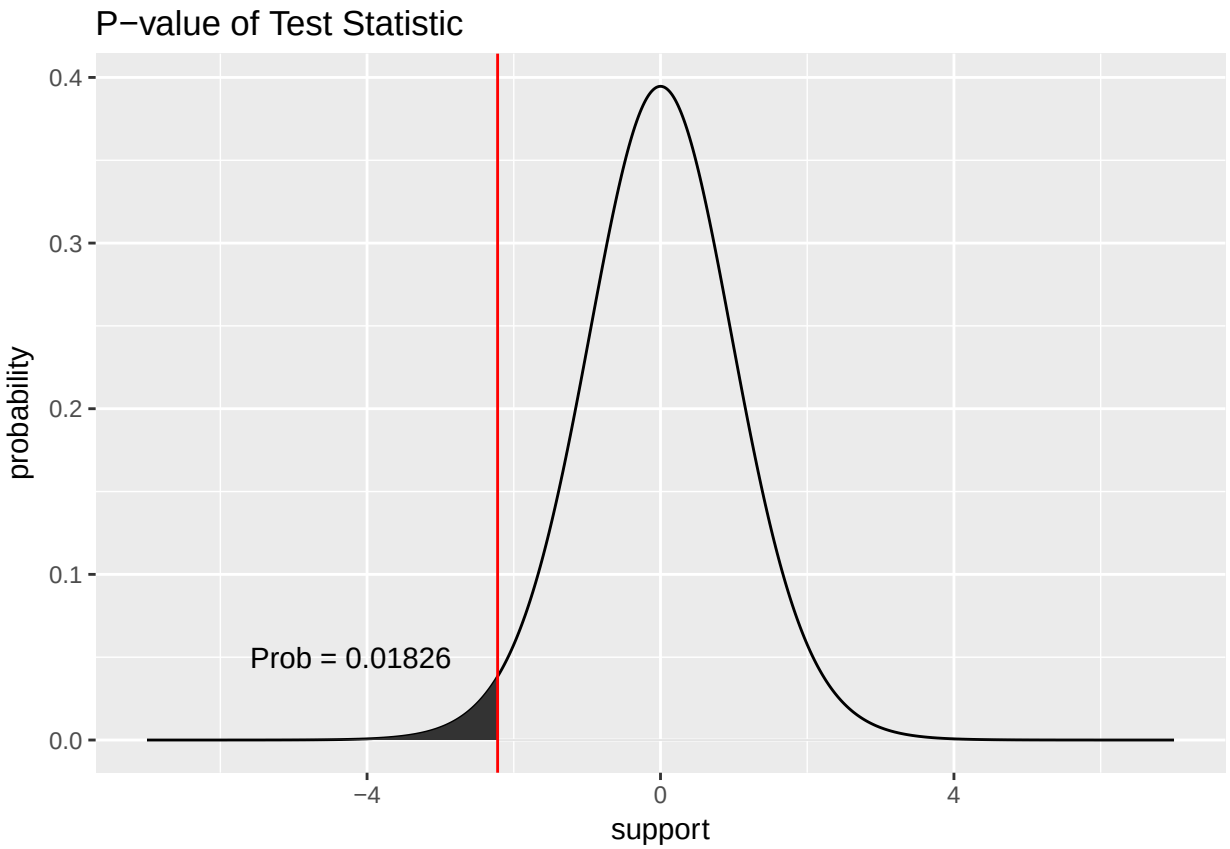
```
t_stat< t.crit #True so reject the null
```

```
## [1] TRUE
```

```
########plotting p-value
p_value <-  pt(t_stat, nrow(dat)-2) #p_value still to the left

p_value
```

```
## [1] 0.0182573
```

```
#Then use this new data.frame with geom_polygon
support = -700:700/100
plot_data <- as.data.frame(cbind(support,
                  probability = dt(support, nrow(dat)-2)))
#note shade order changed
shade <- as.data.frame(rbind(
               subset(plot_data, support < t_stat),
               c(t_stat,0),
               c(plot_data[nrow(plot_data), "support"], 0)))
names(shade) <- c("x","y")
library(ggplot2)
ggplot(data = plot_data, aes(x = support, y= probability))  +
  geom_line() +
  annotate("text", x = t_stat - 2, y = .05, label =
             paste("Prob = ", round(p_value,5), sep = "")) +
  geom_polygon(data = shade, aes(x,y )) +
  ggtitle("P-value of Test Statistic")  +
  geom_vline(xintercept = t_stat, col = 'red'  )
```



d

Use the model given in part (c), construct a 95% interval estimate for the parameter δ_2 and give its interpretation. Have we estimated the effect of the repairs on motel occupancy relatively precisely or not? Explain. *Note: Precision and accuracy are not the same thing!*

Solution In symbols the 95% CI will be $\hat{\delta_2} \pm t_{.975,N-2} se(\hat{\delta_2})$. Note that accuracy is how close $\hat{\delta_2}$ is to δ_2, which cannot be known. Whereas, precision is how large $se(\hat{\delta_2})$ is relative to its own size. Since the confidence interval is so large, we can say the effect of repairs on motel occupancy is not estimated precisely. We can only be 95% confidence that the true effect of hotel repairs on motel percent occupancy is between -.91% and -25.6%

Note that this relationship seems like a good candidate to explore nonlinear relationships. We should really do a plot of repair status and motel percent occupancy.

```
#One line/built-in solution
confint(reg_out, level = .95)
```

```
##                 2.5 %     97.5 %
## (Intercept)  72.82533 85.8746722
## repair      -25.56619 -0.9052429
```

```
#by hand
t_crit <- qt(.975, nrow(dat)-2)
coef(s)[2,1] + t_crit*coef(s)[2,2]
```

```
## [1] -0.9052429
```

```
coef(s)[2,1] - t_crit*coef(s)[2,2]
```

```
## [1] -25.56619
```

```
#note they agree
```

e

Consider the linear regression $MOTEL\_PCT - COMP\_PCT$ and $x = REPAIR$, that is

$$MOTEL\_PCT - COMP\_PCT = \gamma_1 + \gamma_2 REPAIR + e.$$

Test the null hypothesis $\gamma_2 = 0$ against the alternative that $\gamma_2 < 0$ at the $\alpha = .01$ level of significance. Discuss the meaning of the test outcome.

Solution: $H_0 : \gamma_2 \geq 0$ vs $H_1 : \gamma_2 < 0$. So we can reuse the code from b or c with slight modification.

We see below that we reject the null. Our p-value is .0009. This means that, given assumptions S1-S5 and that the null is true, we would expect to see this strong a negative coefficient less than .1 percent the time. Thus, we expect that repairs are correlated with occupancy rates below competitors occupancy rates. Note, we cannot make the causal statement from this that "repairs cause lower percent rates". If we wanted to think about this question a little more seriously, we should be thinking about price and other confounders.

```
new_pct <- motel_pct - comp_pct
reg_out <- lm(new_pct ~repair)

s <- summary(reg_out)
s
```

```
##
## Call:
## lm(formula = new_pct ~ repair)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.743  -4.861  -2.261   5.139  23.939
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.861      2.109   7.993 4.35e-08 ***
## repair       -14.118      3.986  -3.542  0.00174 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.949 on 23 degrees of freedom
## Multiple R-squared:  0.3529, Adjusted R-squared:  0.3248
## F-statistic: 12.54 on 1 and 23 DF,  p-value: 0.001742
```

```
t_stat <- coef(s)[2,1]/coef(s)[2,2]
t_stat
```

```
## [1] -3.541672
```

```
t.crit <- qt(.01, nrow(dat)-2)
t.crit
```

```
## [1] -2.499867
```

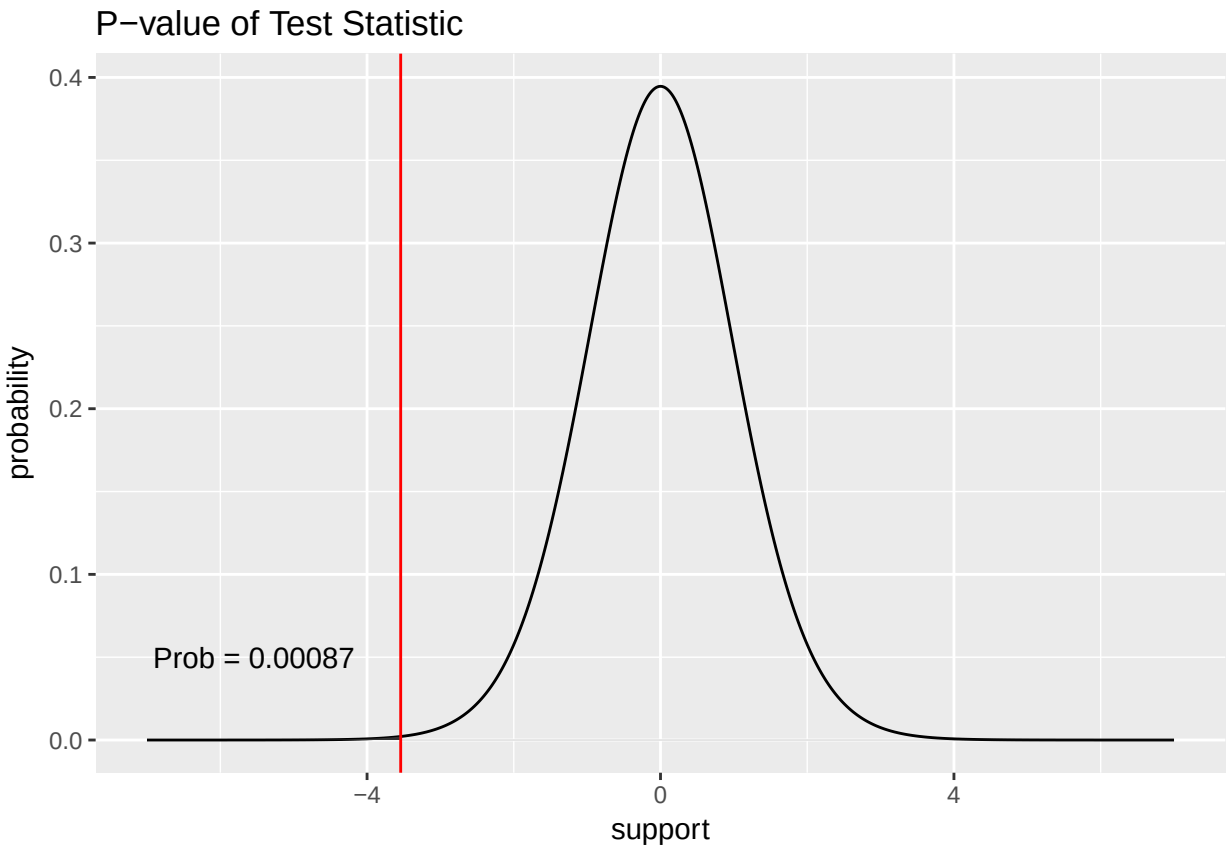
```r
t_stat< t.crit #True so reject the null
```

```
## [1] TRUE
```

```r
########plotting p-value
p_value <-  pt(t_stat, nrow(dat)-2) #p_value still to the left

p_value
```

```
## [1] 0.0008708237
```

```r
#Then use this new data.frame with geom_polygon
support = -700:700/100
plot_data <- as.data.frame(cbind(support,
                  probability = dt(support, nrow(dat)-2)))
#note shade order changed
shade <- as.data.frame(rbind(
               subset(plot_data, support < t_stat),
               c(t_stat,0),
               c(plot_data[nrow(plot_data), "support"], 0)))
names(shade) <- c("x","y")
library(ggplot2)
ggplot(data = plot_data, aes(x = support, y= probability))  +
  geom_line() +
  annotate("text", x = t_stat - 2, y = .05, label =
             paste("Prob = ", round(p_value,5), sep = "")) +
  geom_polygon(data = shade, aes(x,y )) +
  ggtitle("P-value of Test Statistic")  +
  geom_vline(xintercept = t_stat, col = 'red'  )
```



f

Using the model in part (e), construct and discuss the 95% interval estimate of γ_2.

Solution Below, our confidence interval is estimated to be [-5.87, -22.36]. This estimate looks more precise than the previous one.

```
confint(reg_out, level = .95)
```

```
##                 2.5 %    97.5 %
## (Intercept)  12.49756 21.224665
## repair      -22.36460 -5.871913
```

4.13

The file `stockton2.dat` contains data on 880 houses sold in Stockton, CA, during mid-2005. Variable descriptions are in the file `stockton2.def`. These data were considered in Exercises 2.12 and 3.11.

As I start the problem, the following table from the text comes in handy:

a

Estimate the log-linear model $\log(PRICE) = \beta_1 + \beta_2 SQFT + e$. Interpret the estimated model parameters. Calculate the slope and elasticity at the sample means, if necessary.

Table 4.1 Some Useful Functions, their Derivatives, Elasticities and Other Interpretation

Name	Function	Slope $= dy/dx$	Elasticity
Linear	$y = \beta_1 + \beta_2 x$	β_2	$\beta_2 \frac{x}{y}$
Quadratic	$y = \beta_1 + \beta_2 x^2$	$2\beta_2 x$	$(2\beta_2 x)\frac{x}{y}$
Cubic	$y = \beta_1 + \beta_2 x^3$	$3\beta_2 x^2$	$(3\beta_2 x^2)\frac{x}{y}$
Log-Log	$\ln(y) = \beta_1 + \beta_2 \ln(x)$	$\beta_2 \frac{y}{x}$	β_2
Log-Linear	$\ln(y) = \beta_1 + \beta_2 x$	$\beta_2 y$	$\beta_2 x$
	or, a 1 unit change in x leads to (approximately) a 100 β_2% change in y		
Linear-Log	$y = \beta_1 + \beta_2 \ln(x)$	$\beta_2 \frac{1}{x}$	$\beta_2 \frac{1}{y}$
	or, a 1% change in x leads to (approximately) a $\beta_2/100$ unit change in y		

Figure 1:

Solution Note, the model is log-linear, so the slope is $\beta_2 price$ and the elasticity is just $\beta_2 sqft$. The interpretation of β_2 is that a 1 unit change in SQFT leads to a $100 \times \hat{\beta}_2$ percent change in price.

```
my_wd <- "C:/Users/ryanj/Dropbox/TA/Econ 103/Winter 2018/Data/s4poe_statadata"
my_file <- paste(my_wd, "stockton2.dta", sep = "/")
library(haven)
dat <- read_stata(my_file)
#View(dat)
attach(dat)
reg_out <- lm(data=dat, I(log(price))~ sqft)
reg_outa <- reg_out
s <- summary(reg_out)

## Elasticity
coef(s)[2,1] * mean(sqft)
```

```
## [1] 0.9606732
```

```
## Slope at average y
coef(s)[2,1] * mean(price)
```

```
## [1] 67.23106
```

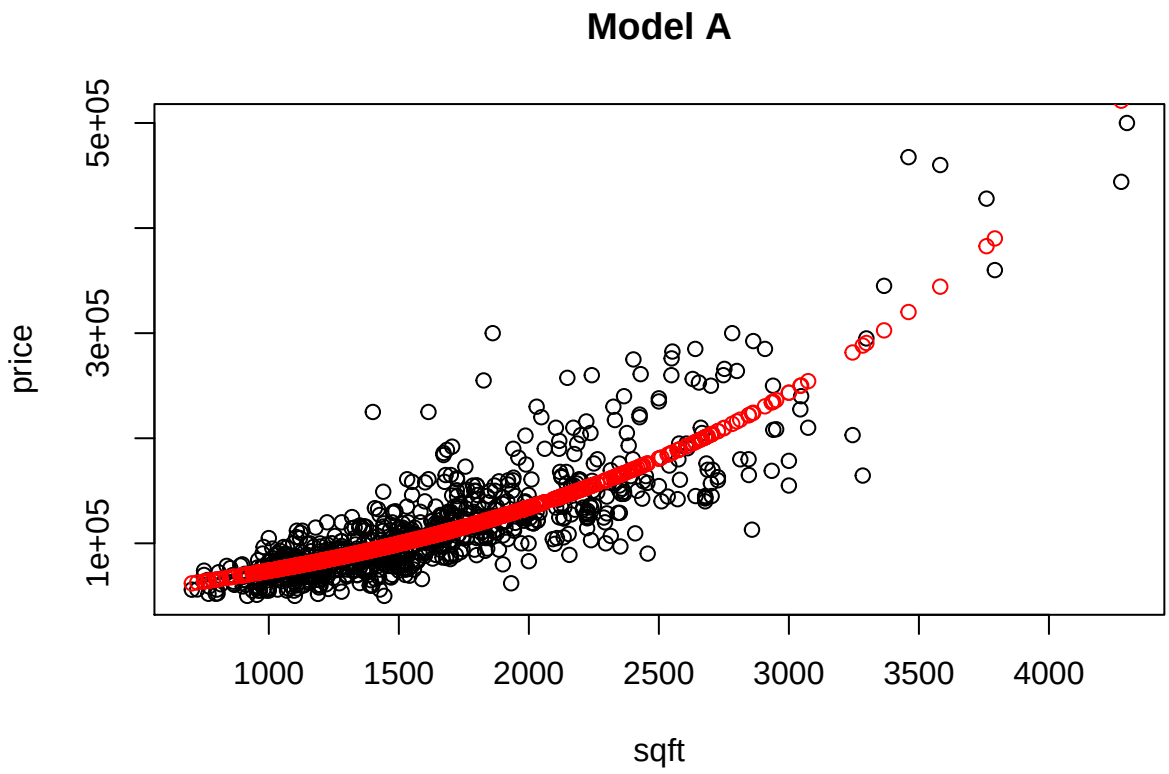
```
r.squareda <- s$r.squared
r.adja <- s$adj.r.squared
y_ca = exp(s$coefficients[1] +
            s$coefficients[2]*sqft  + var(s$residuals)/2)
y.orig <-  exp(s$coefficients[1] +
            s$coefficients[2]*sqft)

cor(price,y.orig)^2
```

```
## [1] 0.7147877
```

```
y.orig.2 <- predict.lm(reg_out)

cor(price,exp(y.orig.2))^2
```

```
## [1] 0.7147877
```

```
res_a <- s$residuals

#special predict,
sqft_spec = 2700
y_spec_ca = exp(s$coefficients[1] +
            s$coefficients[2]*2700  + var(s$residuals)/2)
y_spec_na =  exp(s$coefficients[1] +
            s$coefficients[2]*2700)


plot(sqft,price, main = "Model A")
lines(sqft,y_ca, col ="red", type = "p")
```



b

Estimate the log-log model $\log(PRICE) = \beta_1 + \beta_2 \log(SQFT) + e$. Interpret the estimated parameters. Calculate the slope and elasticity at the sample means, if necessary.

Solution: In this model, β_2 is the elasticity of Price to Sqft. So, a 1% change in SQFT results in a $\hat{\beta}_2$ percent

change in *price*. $\hat{\beta}_1$ is the estimated log of price when $sqft = 1$

```
reg_out <- lm(data=dat, I(log(price))~ I(log(sqft)))
reg_outb <- reg_out
s <- summary(reg_out)
s
```

```
##
## Call:
## lm(formula = I(log(price)) ~ I(log(sqft)), data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75190 -0.13022 -0.01824  0.11806  0.86244
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.17068    0.16551   25.20   <2e-16 ***
## I(log(sqft))  1.00658    0.02254   44.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2083 on 878 degrees of freedom
## Multiple R-squared:  0.6943, Adjusted R-squared:  0.6939
## F-statistic:  1994 on 1 and 878 DF,  p-value: < 2.2e-16
```

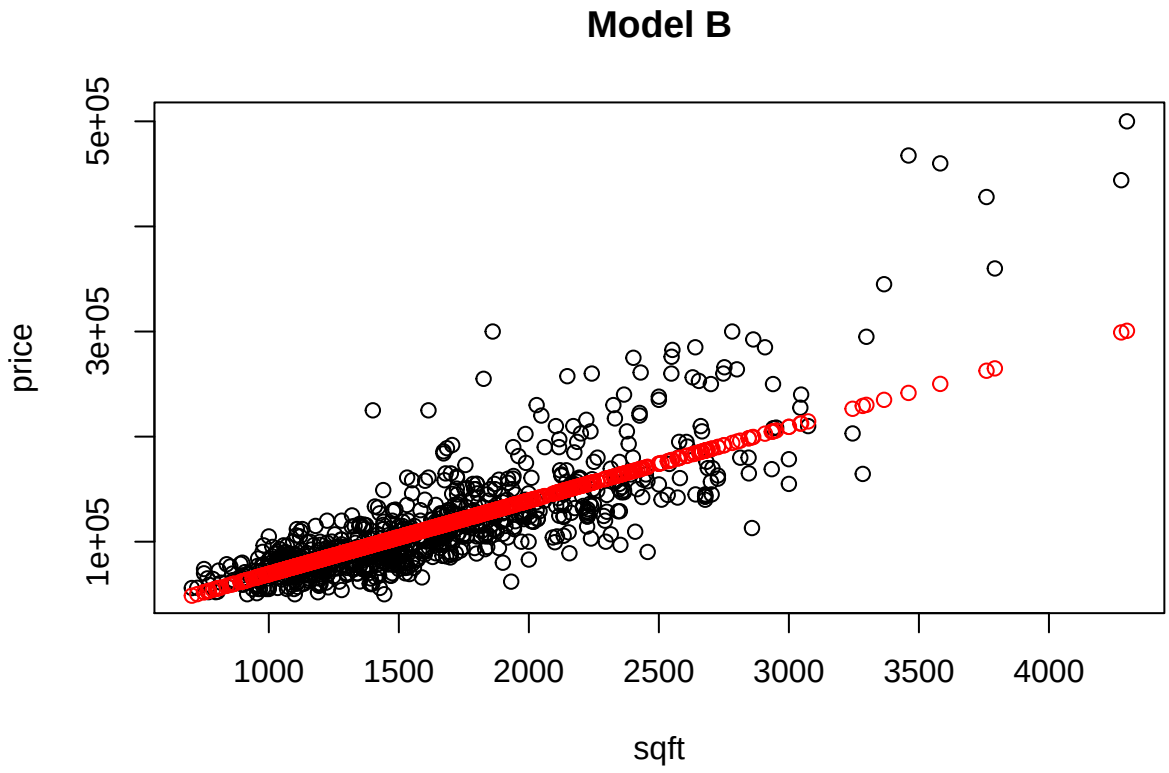
```
## Elasticity
coef(s)[2,1]
```

```
## [1] 1.006582
```

```
## Slope at average y
coef(s)[2,1] * mean(price)/mean(sqft)
```

```
## [1] 70.44389
```

```
r.squaredb <- s$r.squared
r.adjb <- s$adj.r.squared
y_cb = exp(s$coefficients[1] +
            s$coefficients[2]*log(sqft)  + var(s$residuals)/2)

res_b <- s$residuals

#special predict,
y_spec_cb = exp(s$coefficients[1] +
            s$coefficients[2]*log(2700)  + var(s$residuals)/2)
y_spec_nb = exp(s$coefficients[1] +
            s$coefficients[2]*log(2700))

plot(sqft,price, main = "Model B")
lines(sqft,y_cb, col ="red", type = "p")
```



c

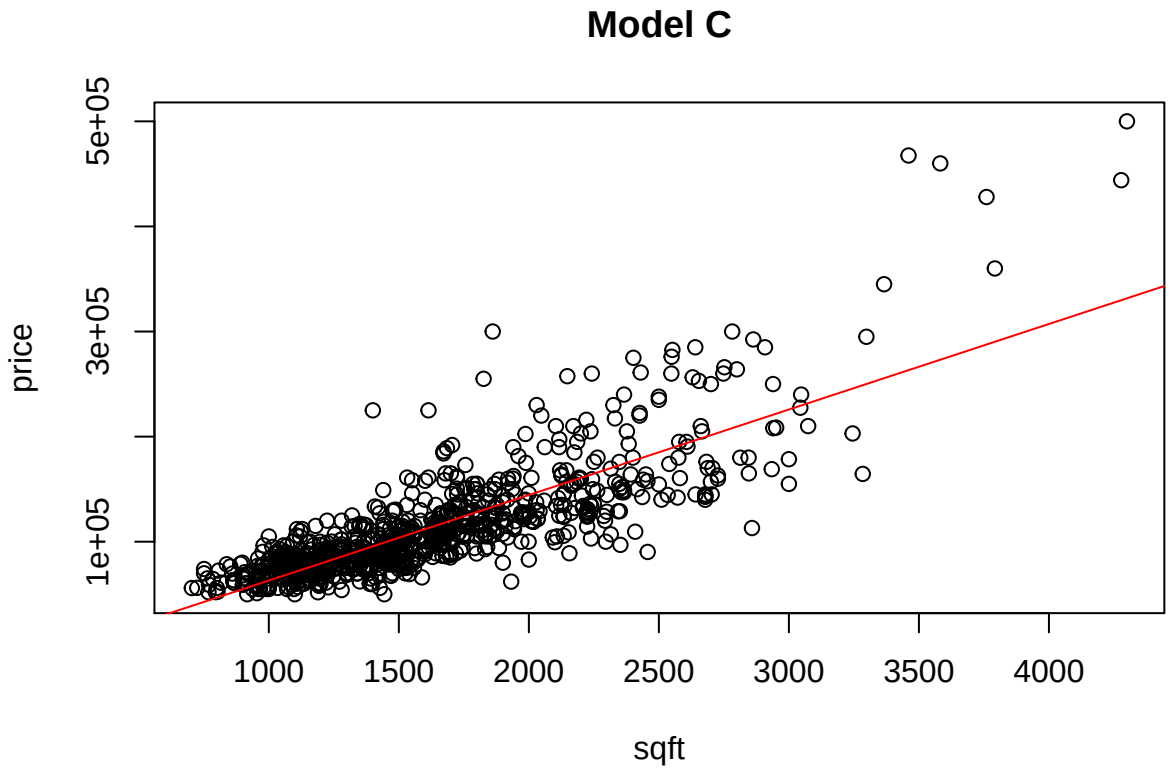
Compare the R^2-value from the linear model $PRICE = \beta_1 + \beta_2 SQFT + e$ to the "generalized" R^2 measure for the models in (a) and (b).

Solution Estimates are similar. See below.

```
#linear model
reg_outc <- lm(data=dat,price ~ sqft)
sc <- summary(reg_outc)
res_c <- sc$residuals #for later
y_spec_c = coef(sc)[1,1] + coef(sc)[2,1] * 2700
sc$r.squared #slightly lower r^2 than the generalized below
```

```
## [1] 0.672113
```

```
plot(sqft, price, main = "Model C")
abline(reg_outc, col ="red")
```




```
#original
r.squareda
```

```
## [1] 0.7094043
```

```
r.squaredb
```

```
## [1] 0.6942774
```

```
#R-determined "adjusted"
r.adja
```

```
## [1] 0.7090733
```

```
r.adjb
```

```
## [1] 0.6939292
```

```
#generalized
cor(y_ca,price)^2
```

```
## [1] 0.7147877
```

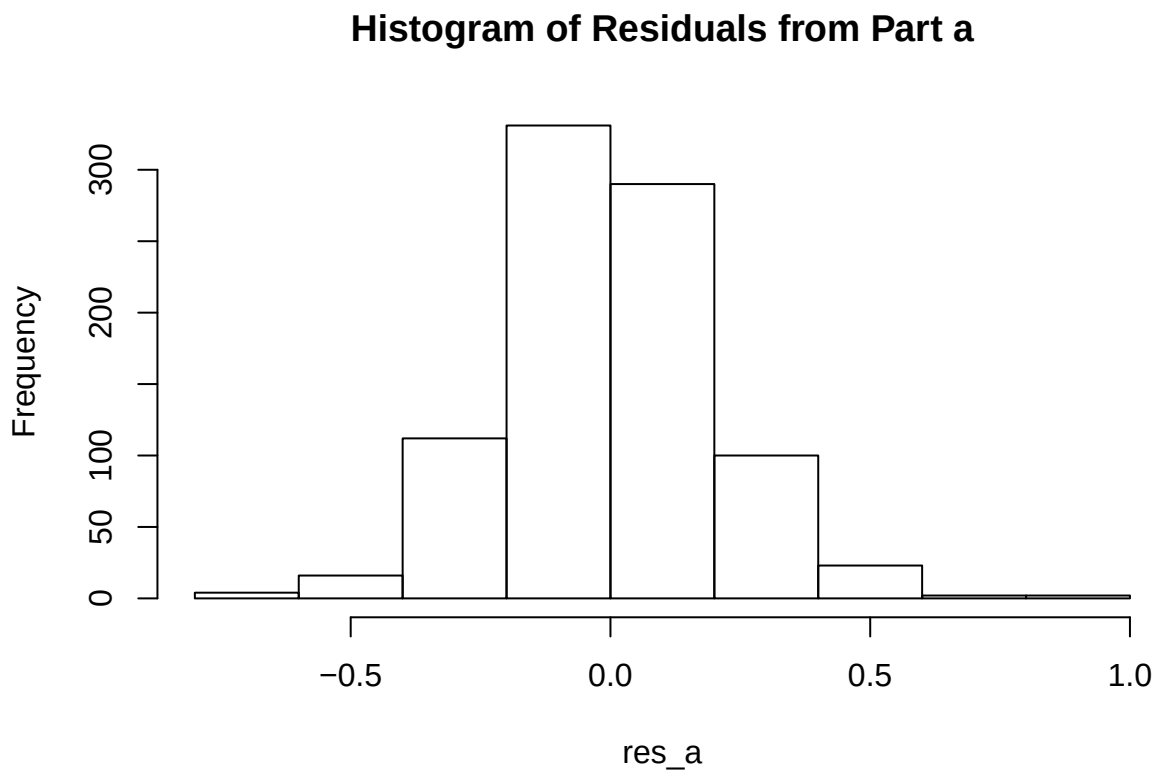
```
cor(y_cb,price)^2
```

```
## [1] 0.6725513
```

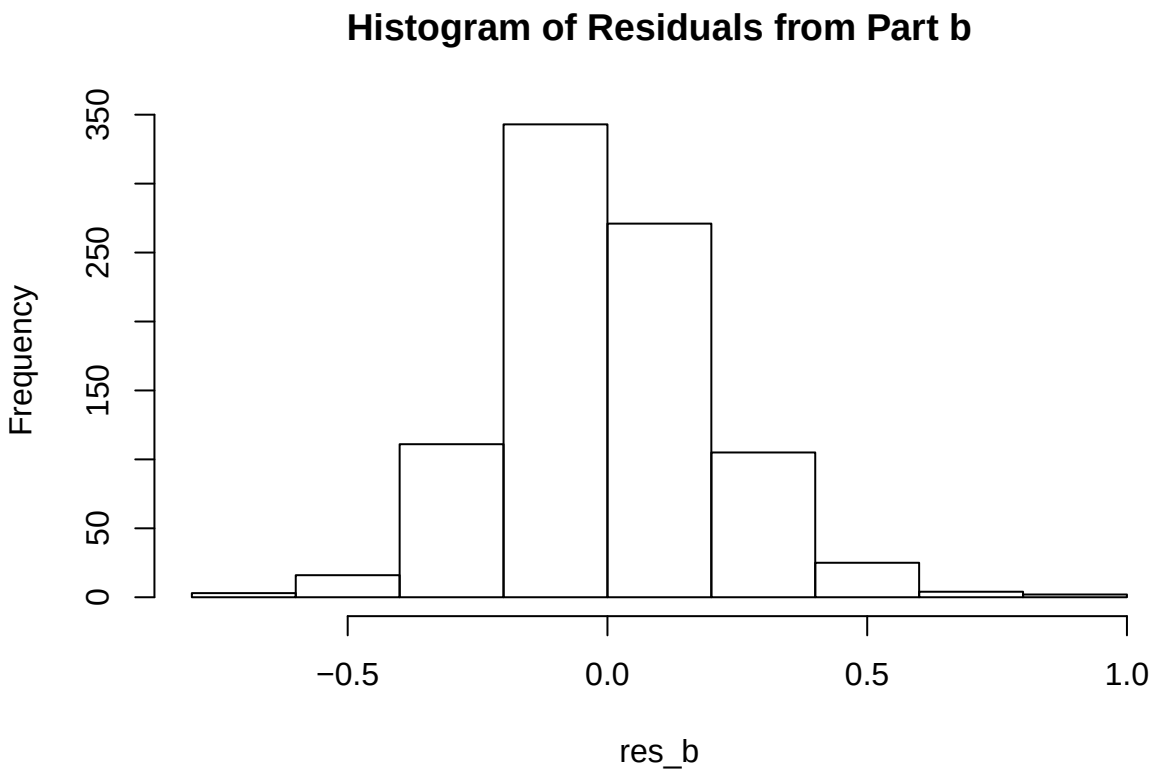
d

Construct histograms of the least squares residuals from each of the models in (a), (b), and (c) and obtain the Jarque-Bera statistics. Based on your observations, do you consider the distributions of the residuals to be compatible with an assumption of normality?

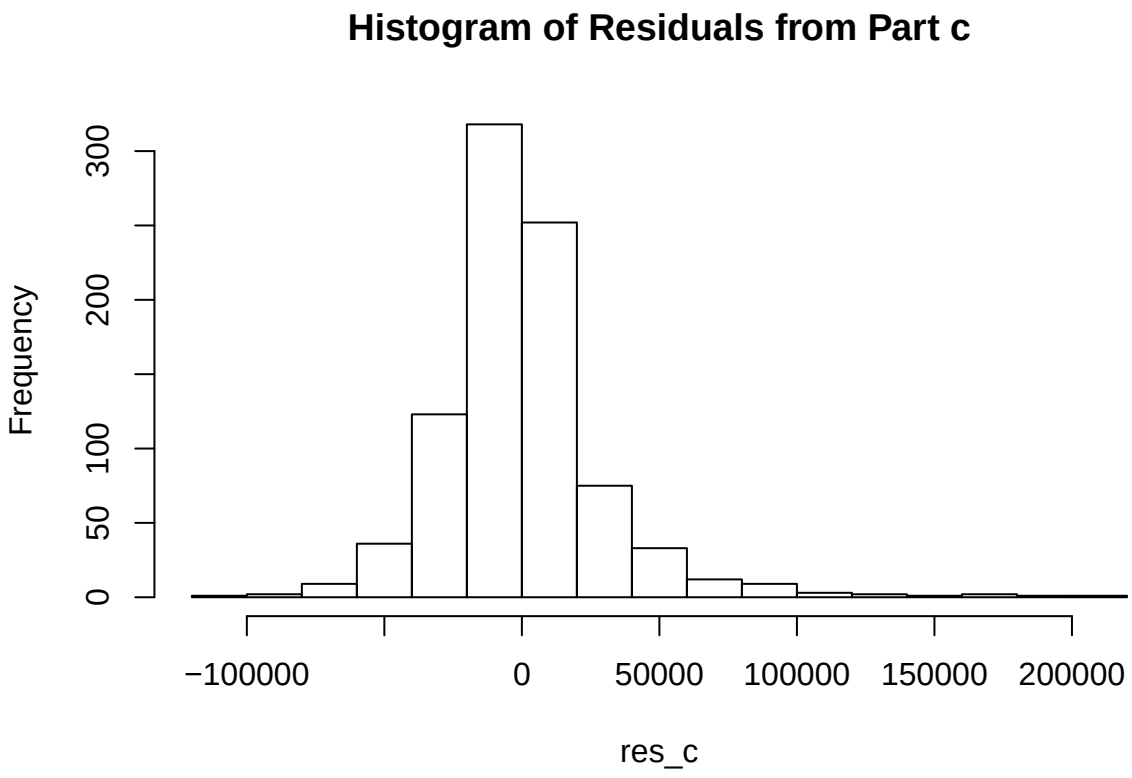
```
hist(res_a,main = "Histogram of Residuals from Part a")
```




```
hist(res_b,main = "Histogram of Residuals from Part b")
```




```
hist(res_c,main = "Histogram of Residuals from Part c")
```




```
#To do jarque-bera, need a package
#install.packages("normtest")
library(normtest)
jb.norm.test(res_a) #reject null, not normal
```

```
##
##  Jarque-Bera test for normality
##
## data:  res_a
## JB = 78.854, p-value < 2.2e-16
```

```
jb.norm.test(res_b) #reject null, not normal
```

```
##
##  Jarque-Bera test for normality
##
## data:  res_b
## JB = 52.744, p-value < 2.2e-16
```

```
jb.norm.test(res_c) #reject null, not normal
```

```
##
##  Jarque-Bera test for normality
##
## data:  res_c
## JB = 2455.9, p-value < 2.2e-16
```

```r
#install.packages("fBasics")
library(fBasics)
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```r
skewness(res_a) #if normal, 0
```

```
## [1] 0.3233787
## attr(,"method")
## [1] "moment"
```

```r
skewness(res_b)
```

```
## [1] 0.3482093
## attr(,"method")
## [1] "moment"
```

```r
skewness(res_c)
```

```
## [1] 1.589347
## attr(,"method")
## [1] "moment"
```

```r
kurtosis(res_a) #if normal, 3
```

```
## [1] 1.305807
## attr(,"method")
## [1] "excess"
```

```r
kurtosis(res_b)
```

```
## [1] 0.9665737
## attr(,"method")
## [1] "excess"
```

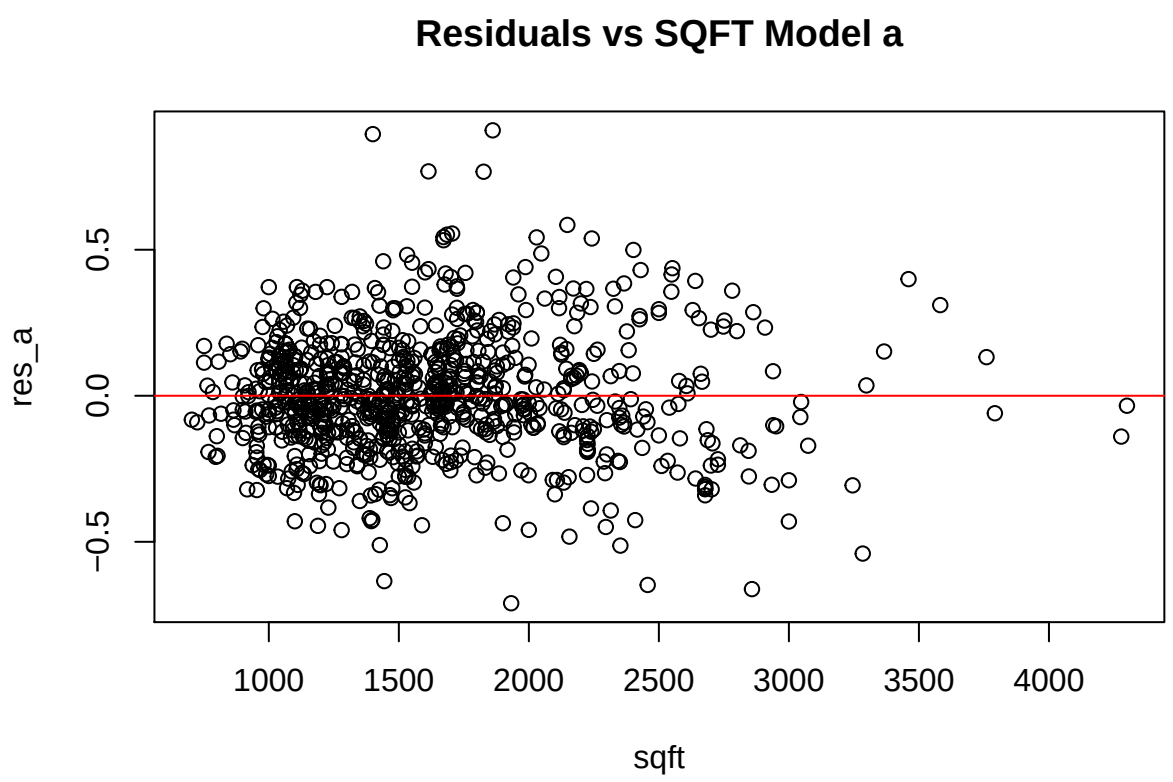
```r
kurtosis(res_c)
```

```
## [1] 7.515277
## attr(,"method")
## [1] "excess"
```

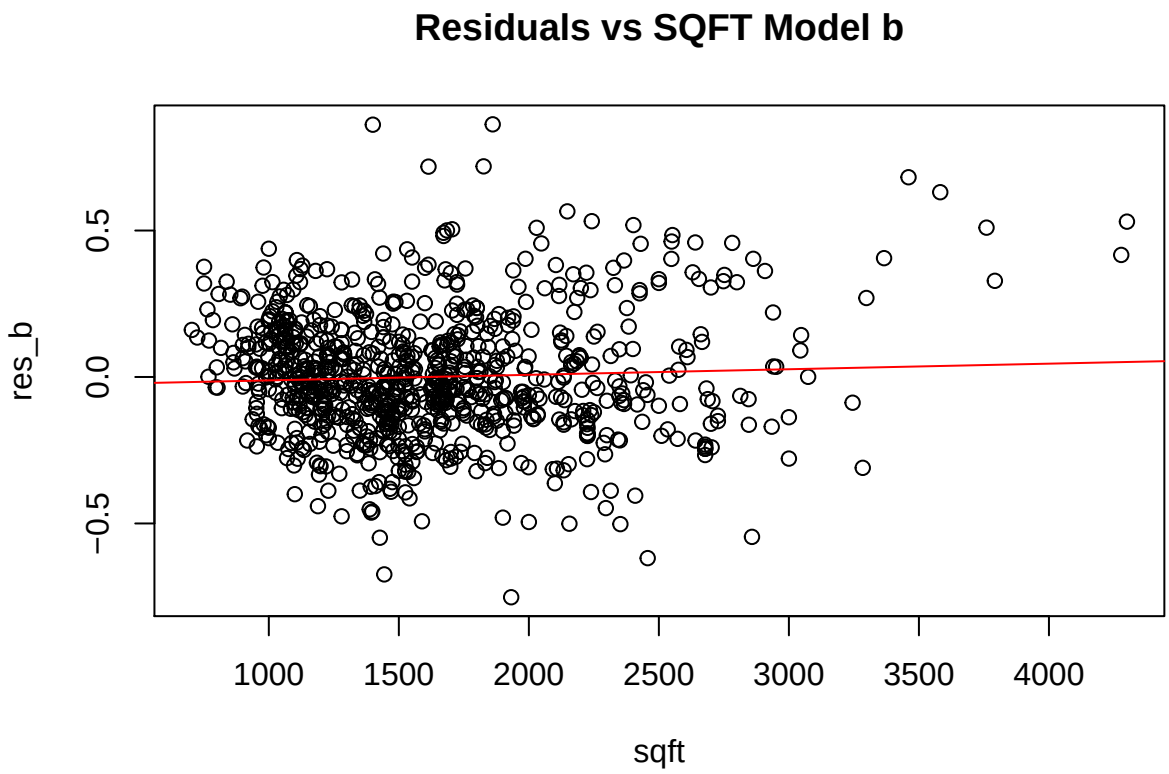
e

For each of the models (a)-(c), plot the least squares residuals against SQFT. Do you observe any patterns?

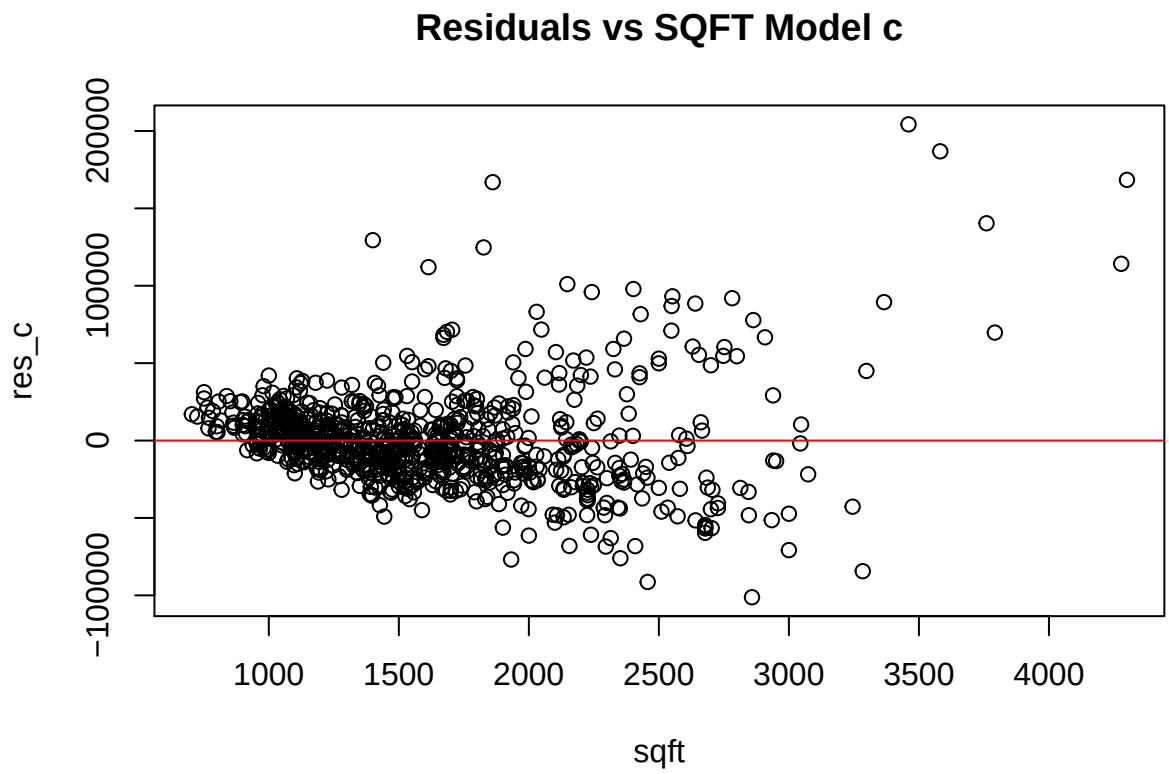
```r
reg1 <- lm(res_a ~sqft)
reg2 <- lm(res_b ~ sqft)
reg3 <- lm(res_c ~ sqft)
plot(sqft, res_a, main = "Residuals vs SQFT Model a")
abline(reg1, col = 'red')
```




```
plot(sqft, res_b, main = "Residuals vs SQFT Model b")
abline(reg2, col = 'red')
```




```
plot(sqft, res_c, main = "Residuals vs SQFT Model c")
abline(reg3, col = 'red')
```



f

For each model in (a)-(c) predict the value of a house with 2700 square feet.

Solution: Recall for the log models, we have two methods of calculating predicted value

```
#The natural estimates for a and b models
y_spec_na
```

```
## [1] 199364
```

```
y_spec_nb
```

```
## [1] 184183.2
```

```
#the corrected estimates for models a and b
y_spec_ca
```

```
## [1] 203511
```

```
y_spec_cb
```

```
## [1] 188216.1
```

```
#the estimate for c
y_spec_c
```

```
## [1] 201364.6
```

g

For each model in (a) - (c) construct a 95% prediction interval for a house with 2700 square feet.

Solution: Interval prediction for log-linear model does NOT use corrected predictors. It uses the natural predictor, $\hat{y_n}$. Recall that if f is the forecast error $f = haty_n - y$, then our prediction interval is$[\exp(\widehat{ln(y)} - t_c se(f)), \exp(\widehat{ln(y)} + t_c se(f))]$.

Finally, note that a prediction interval is not the same as a confidence interval! A confidence interval tries to guess the range of the true parameter given our data. We build confidence intervals around our expected values. A prediction interval tries to find an iterval where most of the true data points will lie. Prediction intervals use the variance of the residuals vs confidence intervals use the variance of the estimates.

```
t_c <- qt(.975, nrow(dat) -2)
#part a
y_spec_na *exp(  - t_c *sd(res_a)) #lower
```

```
## [1] 133870.3
```

```
y_spec_na *exp(  t_c *sd(res_a)) #upper
```

```
## [1] 296899.4
```

```
#part b
y_spec_nb *exp(  - t_c *sd(res_b)) #lower
```

```
## [1] 122417.3
```

```
y_spec_nb *exp(  t_c *sd(res_b)) #upper
```

```
## [1] 277113.3
```

```
#part c
y_spec_c   - t_c *sd(res_c) #lower
```

```
## [1] 142009.6
```

```
y_spec_c +  t_c *sd(res_c) #upper
```

```
## [1] 260719.6
```

```
predict.lm(reg_outc, newdata= data.frame(sqft = 2700),interval = "prediction", level = .95)
```

```
##        fit    lwr      upr
## 1 201364.6 141801 260928.2
```

```
#For comparison
predict.lm(reg_outc, newdata= data.frame(sqft = 2700),interval = "confidence", level = .95)
```

```
##        fit      lwr      upr
## 1 201364.6 196804.8 205924.4
```

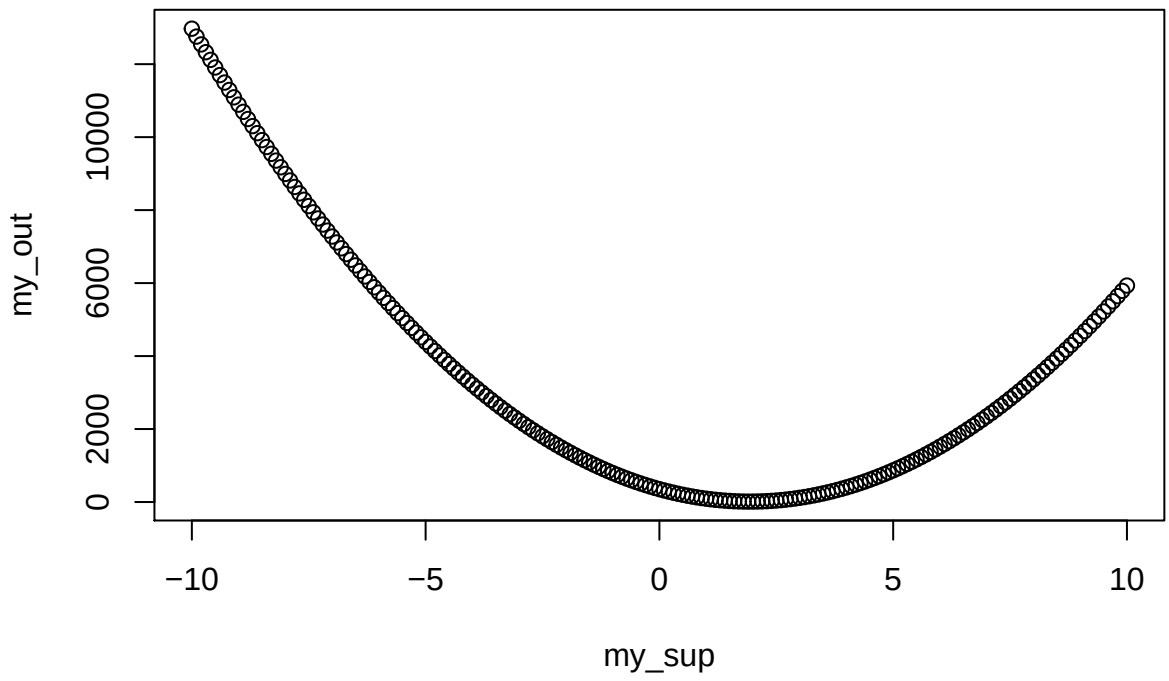
h

Based on your work in this problem, discuss the choice of functional form. Which functional form would you use? Explain.

Solution: I would probably use a, because it captures the right shape for the higher values of sqft.

Homework Help

```
x <- 1:6
y = c(4,6,7,7,9,11)
my_func <- function(b){
  sum( (y - b*x)^2)

}
my_sup =-100:100/10
my_out <- sapply(my_sup, my_func)
plot(my_sup, my_out)
abline()
```

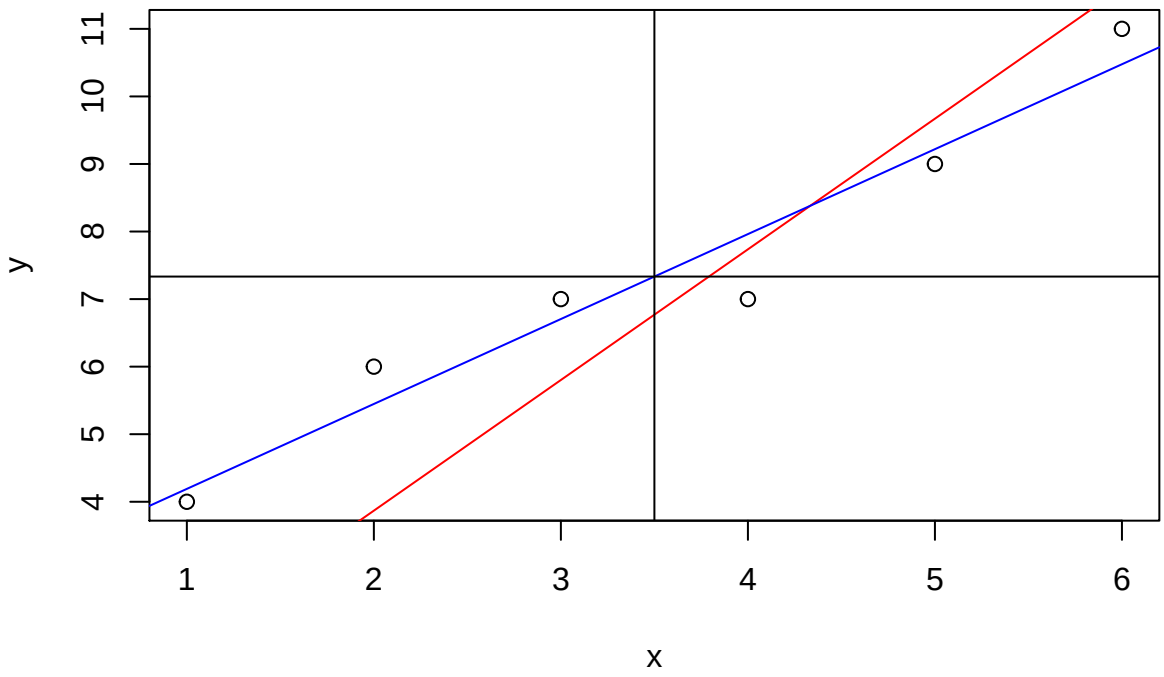



```
sum(y*x)/sum(x^2)
```

```
## [1] 1.934066
```

```
reg_out <- lm(y ~x + 0)
reg_out2 <- lm(y~x)
#plot(c(0,6),c(0,11), col = "white")
#lines(x,y, type = "p")
#abline(h = reg_out$coefficients[1], col='red')
plot(x,y)
abline(reg_out, col = "red")
abline(reg_out2, col = "blue")
abline(v = mean(x))
```

```
abline(h = mean(y))
```




```
reg_out$residuals
```

```
##          1          2          3          4          5          6
##  2.0659341  2.1318681  1.1978022 -0.7362637 -0.6703297 -0.6043956
```

```
sum(reg_out$residuals*x)
```

```
## [1] -2.664535e-15
```

```
sum(reg_out2$residuals*x)
```

```
## [1] -5.551115e-16
```