

# Problem Set #2

Anaya Hall & Christian Miller

## Part 1: Theory

(for practice only)

## Part 2: Applied - Returns to Scale in Electricity Supply

First, load our OLS function created in Problem Set #1. We're including a built in t-test this time around.

```
ols <- function(data, y_data, X_data, intercept = T, H0 = 0, two_tail = T, alpha = 0.05) {  
  # Function setup ----  
  # Require the 'dplyr' package  
  require(dplyr)  
  # Function to convert tibble, data.frame, or tbl_df to matrix  
  to_matrix <- function(the_df, vars) {  
    # Create a matrix from variables in var  
    new_mat <- the_df %>%  
      #Select the columns given in 'vars'  
      select_(.dots = vars) %>%  
      # Convert to matrix  
      as.matrix()  
    # Return 'new_mat'  
    return(new_mat)  
  }  
  
  # Create dependent and independent variable matrices ----  
  # y matrix  
  y <- to_matrix (the_df = data, vars = y_data)  
  # X matrix  
  X <- to_matrix (the_df = data, vars = X_data)  
  # If 'intercept' is TRUE, then add a column of ones  
  if (intercept == T) {  
    X <- cbind(1,X)  
    colnames(X) <- c("intercept", X_data)  
  }  
  
  # Calculate b, y_hat, and residuals ----  
  b <- solve(t(X) %*% X) %*% t(X) %*% y  
  y_hat <- X %*% b  
  e <- y - y_hat  
  
  # Useful -----  
  n <- nrow(X) # number of observations  
  k <- ncol(X) # number of independent variables  
  dof <- n - k # degrees of freedom
```

```

i <- rep(1,n) # column of ones for demeaning matrix
A <- diag(i) - (1 / n) * i %*% t(i) # demeaning matrix
y_star <- A %*% y # for SST
X_star <- A %*% X # for SSM
SST <- drop(t(y_star) %*% y_star)
SSM <- drop(t(b) %*% t(X_star) %*% X_star %*% b)
SSR <- drop(t(e) %*% e)

# Measures of fit and estimated variance ----
R2uc <- drop((t(y_hat) %*% y_hat)/(t(y) %*% y)) # Uncentered R^2
R2 <- 1 - SSR/SST # Uncentered R^2
R2adj <- 1 - (n-1)/dof * (1 - R2) # Adjusted R^2
AIC <- log(SSR/n) + 2*k/n # AIC
SIC <- log(SSR/n) + k/n*log(n) # SIC
s2 <- SSR/dof # s^2

# Measures of fit table ----
mof_table_df <- data.frame(R2uc, R2, R2adj, SIC, AIC, SSR, s2)
mof_table_col_names <- c("$R^2_\\text{uc}$", "$R^2$",
                        "$R^2_\\text{adj}$",
                        "SIC", "AIC", "SSR", "$s^2$")
mof_table <- mof_table_df %>% knitr::kable(
  row.names = F,
  col.names = mof_table_col_names,
  format.args = list(scientific = F, digits = 4),
  booktabs = T,
  escape = F
)

# t-test----
# Standard error
se <- as.vector(sqrt(s2 * diag(solve(t(X) %*% X))))
# Vector of t_ statistics
t_stats <- (b - H0) / se
# Calculate the p-values
if (two_tail == T) {
  p_values <- pt(q = abs(t_stats), df = dof, lower.tail = F) * 2
} else {
  p_values <- pt(q = abs(t_stats), df = dof, lower.tail = F)
}
# Do we (fail to) reject?
reject <- ifelse(p_values < alpha, reject <- "Reject", reject <- "Fail to Reject")

# Nice table (data.frame) of results
ttest_df <- data.frame(
  # The rows have the coef. names
  effect = rownames(b),
  # Estimated coefficients
  coef = as.vector(b) %>% round(3),
  # Standard errors

```

```

    std_error = as.vector(se) %>% round(3),
    # t statistics
    t_stat = as.vector(t_stats) %>% round(3),
    # p-values
    p_value = as.vector(p_values) %>% round(4),
    # reject null?
    significance = as.character(reject)
  )

  ttest_table <- ttest_df %>% knitr::kable(
    booktabs = T,
    format.args = list(scientific = F),
    escape = F
  )

  # Data frame for exporting for y, y_hat, X, and e vectors ----
  export_df <- data.frame(y, y_hat, e, X) %>% tbl_df()
  colnames(export_df) <- c("y", "y_hat", "e", colnames(X))

  # Return ----
  return(list(n=n, dof=dof, b=b, vars=export_df, R2uc=R2uc, R2=R2,
    R2adj=R2adj, AIC=AIC, SIC=SIC, s2=s2, SST=SST, SSR=SSR,
    mof_table=mof_table, ttest=ttest_table))
}

```

We'll also need a function to do an F-test for this Problem Set.

```

# Joint test function (from Ed's notes). Could also write a more complex functions that takes R and
F_test <- function(data, y_var, X_vars) {
  # Turn data into matrices
  y <- to_matrix(data, y_var)
  X <- to_matrix(data, X_vars)
  # Add intercept
  X <- cbind(1, X)
  # Name the new column "intercept"
  colnames(X) <- c("intercept", X_vars)
  # Calculate n and k for degrees of freedom
  n <- nrow(X)
  k <- ncol(X)
  # J is k-1
  J <- k - 1
  # Create the R matrix: bind a column of zeros
  # onto a J-by-J identity matrix
  R <- cbind(0, diag(J))

  # Estimate coefficients
  b <- b_ols(data, y_var, X_vars)
  # Retrieve OLS residuals
  e <- b$vars$e
  # Retrieve s^2

```

```

s2 <- b$s2 %<>% as.numeric()

# Create the inner matrix  $R(X'X)^{-1}R'$ 
RXXR <- R %*% solve(t(X) %*% X) %*% t(R)
# Calculate the F stat
f_stat <- t(R %*% b) %*% solve(RXXR) %*% (R %*% b) / J / s2
# Calculate the p-value;; why normal and not  $\chi^2$ 
p_value <- pf(q = f_stat, df1 = J, df2 = n-k, lower.tail = F)
# Create a data.frame of the f stat. and p-value
results <- data.frame(
  f_stat = f_stat %>% as.vector(),
  p_value = p_value %>% as.vector())
return(results)
}

```

### Question 1:

Read the data into R. Inspect it. Sort by size (Q (kwh)).

```

nerlove <- readxl::read_excel("nerlove.xls", col_names=T)
summary(nerlove)

```

```

##          TC          Q          PL          PF
## Min.   : 0.082   Min.   :    2   Min.   : 1.450   Min.   :10.30
## 1st Qu.: 2.382   1st Qu.: 279   1st Qu.: 1.760   1st Qu.:21.30
## Median : 6.754   Median : 1109   Median : 2.040   Median :26.90
## Mean   : 12.976   Mean   : 2133   Mean   : 3.208   Mean   :26.18
## 3rd Qu.: 14.132   3rd Qu.: 2507   3rd Qu.: 2.190   3rd Qu.:32.20
## Max.   :139.422   Max.   :16719   Max.   :181.000   Max.   :42.80
##          PK
## Min.   :138.0
## 1st Qu.:162.0
## Median :170.0
## Mean   :174.5
## 3rd Qu.:183.0
## Max.   :233.0

```

```

# Fix typo in 13th row (missing a decimal!)
# DO THIS MORE ELEGANTLY!
nerlove[13, "PL"] <- 1.81

```

```

# nerlove %>%
#   filter(PL > 100) %>%
#   mutate(PL = PL/100)

```

```

nerlove %>% arrange(Q)

```

```

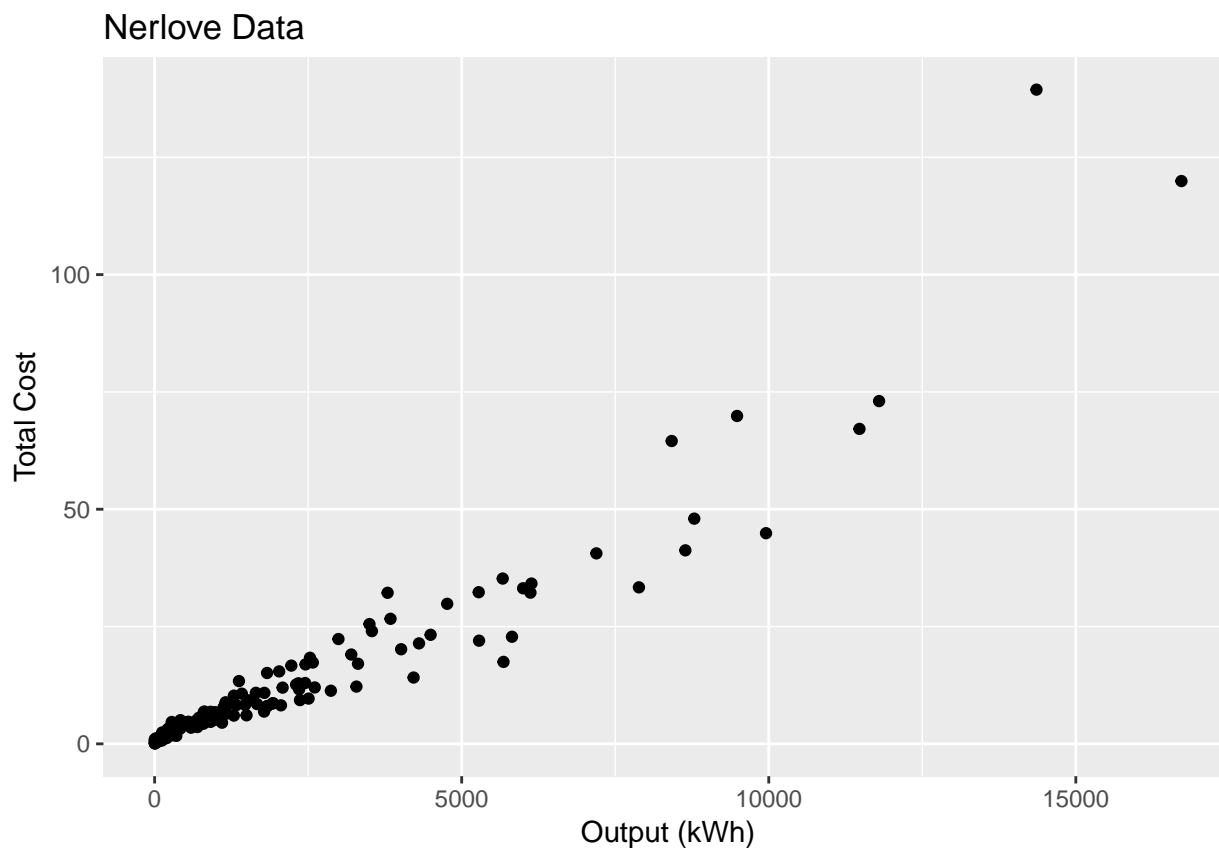
## # A tibble: 145 x 5

```

```
##      TC      Q      PL      PF      PK
##      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 0.0820    2.   2.09  17.9  183.
## 2 0.661     3.   2.05  35.1  174.
## 3 0.990     4.   2.05  35.1  171.
## 4 0.315     4.   1.83  32.2  166.
## 5 0.197     5.   2.12  28.6  233.
## 6 0.0980    9.   2.12  28.6  195.
## 7 0.949    11.   1.98  35.5  206.
## 8 0.675    13.   2.05  35.1  150.
## 9 0.525    13.   2.19  29.1  155.
## 10 0.501   22.   1.72  15.0  188.
## # ... with 135 more rows
```

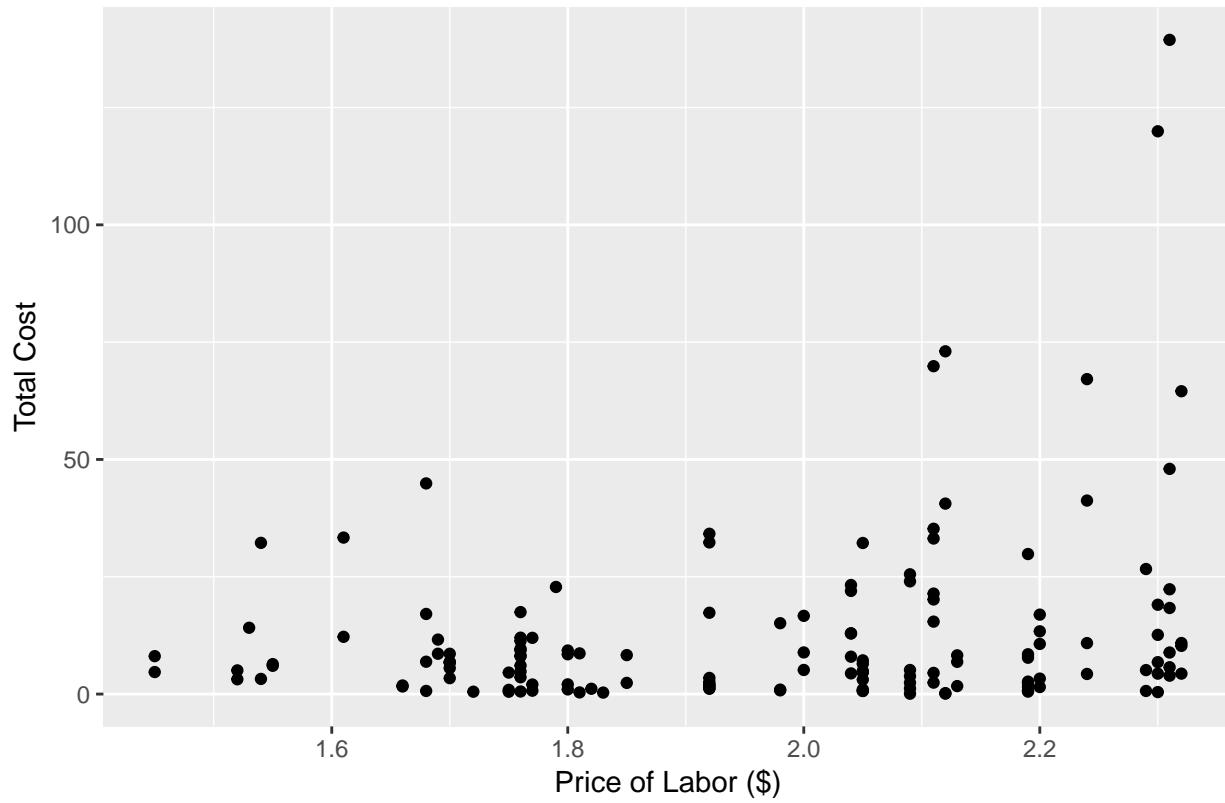
Plot the series and make sure your data are read in correctly.

```
ggplot(nerlove, aes(x=Q, y=TC)) +
  geom_point() +
  labs(title="Nerlove Data", x="Output (kWh)", y="Total Cost")
```



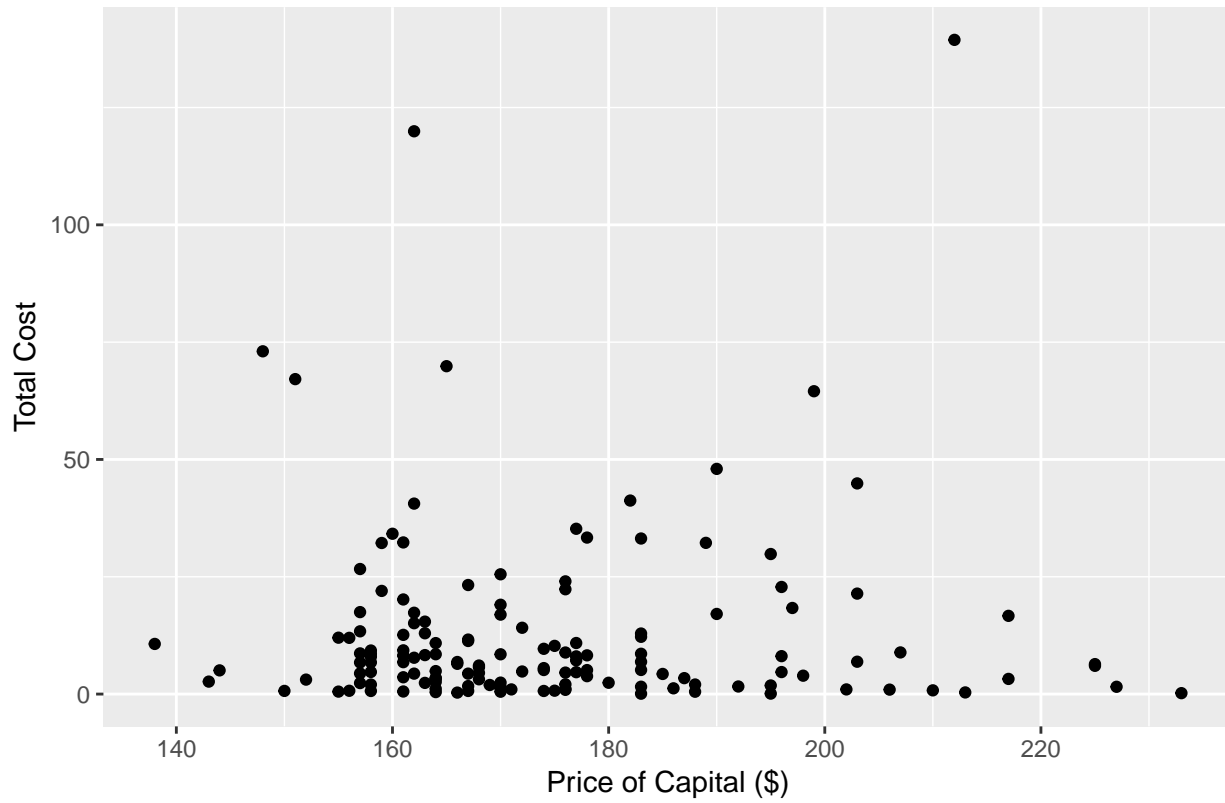
```
ggplot(nerlove, aes(x=PL, y=TC)) +
  geom_point() +
  labs(title="Nerlove Data", x="Price of Labor ($)", y="Total Cost")
```

Nerlove Data



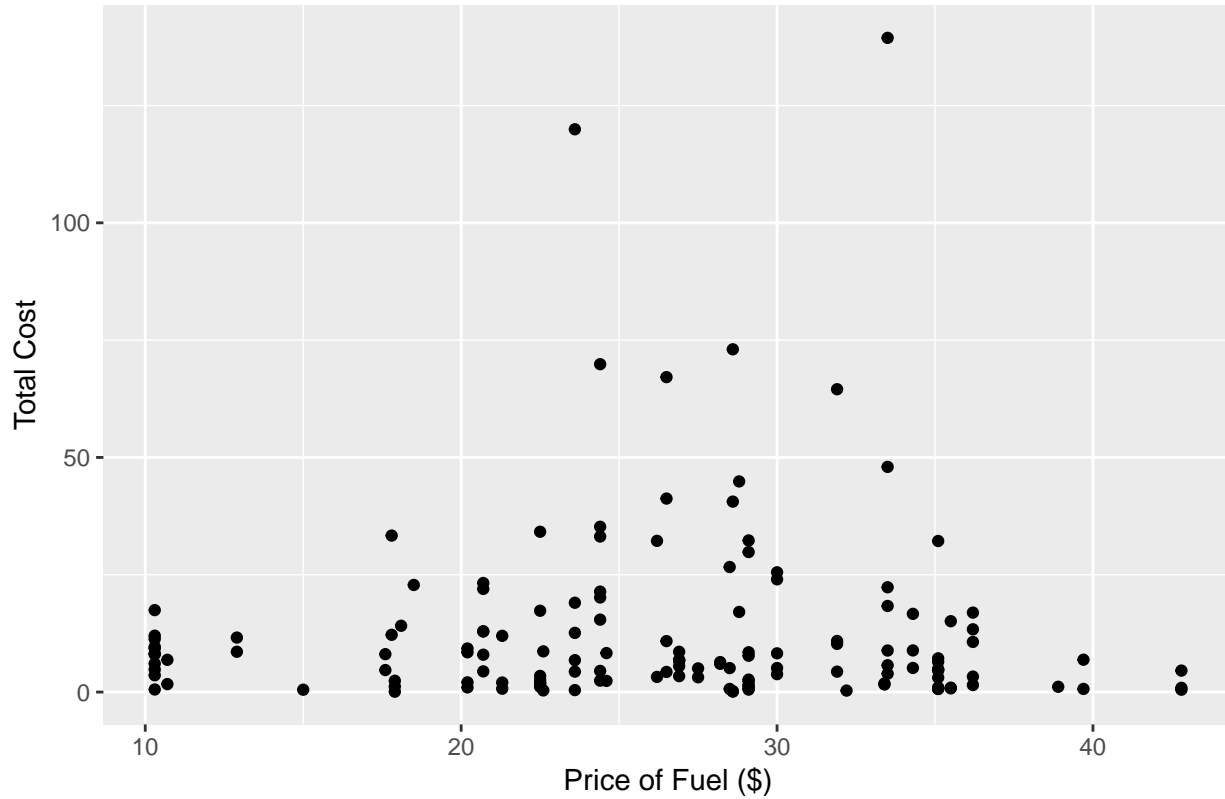
```
ggplot(nerlove, aes(x=PK, y=TC)) +  
  geom_point() +  
  labs(title="Nerlove Data", x="Price of Capital ($)", y="Total Cost")
```

Nerlove Data



```
ggplot(nerlove, aes(x=PF, y=TC)) +  
  geom_point() +  
  labs(title="Nerlove Data", x="Price of Fuel ($)", y="Total Cost")
```

## Nerlove Data



### Question 2:

**Replicate regression I (page 176) in the paper.**

Regression I:

$$\log(TC) - \log(P_F) = \beta_0 + \beta_1 Q + \beta_2 (\log(P_L) - \log(P_F)) + \beta_3 (\log(P_K) - \log(P_F))$$

Equivalent to:

$$\log\left(\frac{TC}{P_F}\right) = \beta_0 + \beta_1 Q + \beta_2 \log\left(\frac{P_L}{P_F}\right) + \beta_3 \log\left(\frac{P_K}{P_F}\right)$$

Where:

$TC$  = total production cost,

$P_L$  = wage rate,

$P_K$  = “price” of capital,

$P_F$  = price of fuel,

$Q$  = output (measured in kWh)

In generalized Cobb-Douglas form:

$$\beta_1 = \frac{1}{r},$$

$$\beta_2 = \frac{a_L}{r},$$

$$\beta_3 = \frac{a_K}{r}$$

Prepare variables for Regression I.



```

# Create log variables
nerlove %<>% mutate(
  TClog = log(TC),
  Qlog = log(Q),
  PLlog = log(PL),
  PKlog = log(PK),
  PFlog = log(PF)
)

# Create PF scaled variables
nerlove %<>% mutate(
  TCscaled = TClog - PFlog,
  PLscaled = PLlog - PFlog,
  PKscaled = PKlog - PFlog
)

```

Variable names:

$$\log\left(\frac{TC}{P_F}\right) = \text{"TCscaled"}$$

$$\log\left(\frac{PL}{P_F}\right) = \text{"PLscaled"}$$

$$\log\left(\frac{PK}{P_F}\right) = \text{"PKscaled"}$$

```

# Regression I:
# dep var = (log costs - log fuel price) = TCscaled
reg_I <- ols(data = nerlove, y_data = "TCscaled",
             X_data = c("Qlog", "PLscaled", "PKscaled"),
             intercept = T, H0 = 0, alpha = 0.05)

reg_I$ttest

```

effect	coef	std_error	t_stat	p_value	significance
intercept	-2.037	0.384	-5.301	0.0000	Reject
Qlog	0.721	0.017	41.334	0.0000	Reject
PLscaled	0.593	0.205	2.898	0.0044	Reject
PKscaled	-0.007	0.191	-0.039	0.9692	Fail to Reject

```
reg_I$mof_table
```

$R_{uc}^2$	$R^2$	$R_{adj}^2$	SIC	AIC	SSR	$s^2$
0.966	0.9316	0.9301	-3.433	-3.515	4.082	0.02895

Coefficients are pretty close to those in the paper.  $R^2$  matches!

### Question 3:

Conduct the hypothesis test using constant returns to scale ( $\beta_1 = 1$ ) as your null hypothesis.

```
# Re-run regression with null hypothesis H0 = 1
reg_I <- ols(data = nerlove, y_data = "TCscaled",
             X_data = c("Qlog", "PLscaled", "PKscaled"),
             intercept = T, H0 = 1, alpha = 0.05)
```

```
reg_I$ttest
```

effect	coef	std_error	t_stat	p_value	significance
intercept	-2.037	0.384	-7.903	0.0000	Reject
Qlog	0.721	0.017	-16.020	0.0000	Reject
PLscaled	0.593	0.205	-1.990	0.0485	Reject
PKscaled	-0.007	0.191	-5.282	0.0000	Reject

What is the p- value associated with you test statistic? What is your point estimate of returns to scale? Constant? Increasing? Decreasing?

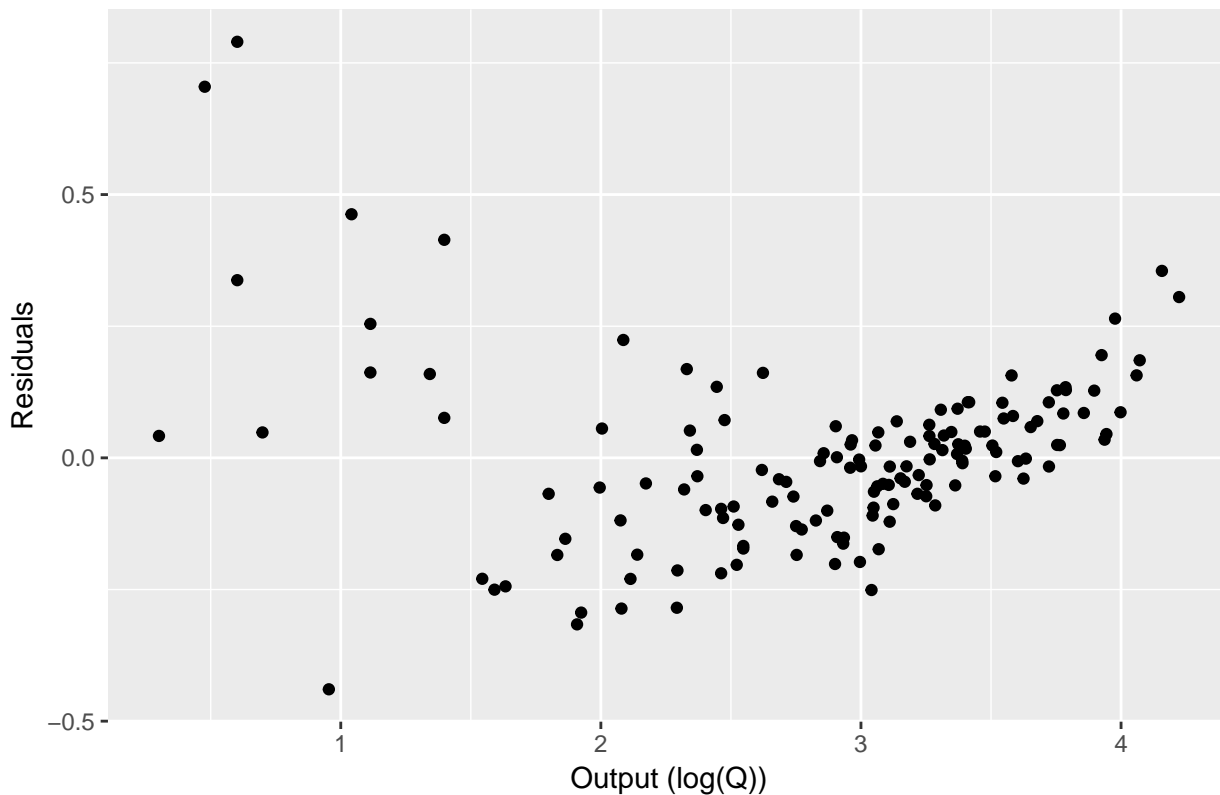
The p-value is 0.000. The point estimate of returns to scale is  $\frac{1}{\beta} = r = 1.3875639$ , hence returns to scale is increasing.

#### Question 4:

Plot residuals against output.

```
ggplot(reg_I$vars, aes(y=e, x=Qlog)) + geom_point() + labs(title="Regression I: Residuals against
```

Regression I: Residuals against Output



What do you notice? What does this potentially tell you from an economic perspective?

*Evidence of heteroskedasticity: residuals seem to track the log output through parabola. We may want to rethink our specification!*

Compute the correlation coefficient of the residuals with output for the entire sample? What does this tell you?

```
# R = cov(xy)/var(x)var(y)
R_I <- (cov(x=reg_I$vars$e, y=reg_I$vars$Qlog)/(var(reg_I$vars$e)*var(reg_I$vars$Qlog))) %>% signif(2)
R_I
```

```
## [1] "7.01e-14"
```

The correlation coefficient is extremely small: 7.01e-14.

### Question 5:

Divide your sample into 5 subgroups of 29 firms each according to the level of output. Estimate the regression model again for each group separately.

```
# Divide sample into 5 subgroups
d <- split(nerlove,rep(1:5,each=29))

# Now we have a list, d, with five dataframes '1' through '5' for our five subgroups.

# Regression IIIA
reg_IIIA <- ols(data = d$`1`, y_data = "TCscaled",
               X_data = c("Qlog", "PLscaled", "PKscaled"),
               intercept = T, H0 = 1, alpha = 0.05)
reg_IIIA$ttest
```

effect	coef	std_error	t_stat	p_value	significance
intercept	-1.452	1.366	-1.795	0.0848	Fail to Reject
Qlog	0.400	0.084	-7.101	0.0000	Reject
PLscaled	0.615	0.729	-0.528	0.6024	Fail to Reject
PKscaled	-0.081	0.706	-1.531	0.1384	Fail to Reject

```
# Regression IIIB
reg_IIIB <- ols(data = d$`2`, y_data = "TCscaled",
               X_data = c("Qlog", "PLscaled", "PKscaled"),
               intercept = T, H0 = 1, alpha = 0.05)
reg_IIIB$ttest
```

effect	coef	std_error	t_stat	p_value	significance
intercept	-2.818	0.614	-6.222	0.0000	Reject
Qlog	0.658	0.116	-2.939	0.0070	Reject
PLscaled	0.094	0.274	-3.304	0.0029	Reject
PKscaled	0.378	0.277	-2.250	0.0335	Reject

# Regression IIIC

```
reg_IIIC <- ols(data = d$`3`, y_data = "TCscaled",
               X_data = c("Qlog", "PLscaled", "PKscaled"),
               intercept = T, H0 = 1, alpha = 0.05)
reg_IIIC$ttest
```

effect	coef	std_error	t_stat	p_value	significance
intercept	-3.185	0.734	-5.705	0.0000	Reject
Qlog	0.938	0.198	-0.312	0.7578	Fail to Reject
PLscaled	0.402	0.199	-2.997	0.0061	Reject
PKscaled	0.250	0.187	-4.010	0.0005	Reject

# Regression IIID

```
reg_IIID <- ols(data = d$`4`, y_data = "TCscaled",
               X_data = c("Qlog", "PLscaled", "PKscaled"),
               intercept = T, H0 = 1, alpha = 0.05)
reg_IIID$ttest
```

effect	coef	std_error	t_stat	p_value	significance
intercept	-2.843	0.506	-7.596	0.0000	Reject
Qlog	0.912	0.107	-0.818	0.4210	Fail to Reject
PLscaled	0.507	0.187	-2.630	0.0144	Reject
PKscaled	0.093	0.164	-5.525	0.0000	Reject

# Regression IIIE

```
reg_IIIE <- ols(data = d$`5`, y_data = "TCscaled",
               X_data = c("Qlog", "PLscaled", "PKscaled"),
               intercept = T, H0 = 1, alpha = 0.05)
reg_IIIE$ttest
```

effect	coef	std_error	t_stat	p_value	significance
intercept	-2.916	0.454	-8.618	0.0000	Reject
Qlog	1.044	0.065	0.683	0.5008	Fail to Reject
PLscaled	0.603	0.197	-2.014	0.0549	Fail to Reject
PKscaled	-0.289	0.175	-7.374	0.0000	Reject

## May want to clean this up and run as for loop???

Can you replicate Equations IIIA - IIIE? Calculate the point estimates for returns to scale for each sample. Is there a pattern relating to size of output? Regression | Returns to Scale  
 ———— | ————— IIIA | 2.4981863 IIIB | 1.5194085 IIIC | 1.0657807 IIID | 1.0964381 IIIE | 0.9574966

*Coefficients roughly match the results of the paper!*

## Question 6:

Create "dummy variables" for each industry. Interact them with the output variable to create five "slope coefficients".

```
# create group categorical var
for (i in 1:5) {
  d[[i]] %<>% mutate(gvar=i)
}

# unsplit
df <- rbind(d[[1]], d[[2]], d[[3]], d[[4]], d[[5]])

# create dummies
df %<>%
  mutate(
    g1 = ifelse(gvar==1, 1, 0),
    g2 = ifelse(gvar==2, 1, 0),
    g3 = ifelse(gvar==3, 1, 0),
    g4 = ifelse(gvar==4, 1, 0),
    g5 = ifelse(gvar==5, 1, 0))

df %<>% select(-gvar)

# interact with output variable
df %<>%
  mutate(
    lQ_A = Qlog*g1,
    lQ_B = Qlog*g2,
    lQ_C = Qlog*g3,
    lQ_D = Qlog*g4,
    lQ_E = Qlog*g5)
```

Run a model, letting the intercept and slope coefficient on output differ across plants, but let the remainder of the coefficients be pooled across plants.

*# I'm not sure this is the right model... but I'm gettin similar coeffs!*

```
reg_IV <- ols(data = df, y_data = "TCscaled",
  X_data = c("lQ_A", "lQ_B", "lQ_C", "lQ_D", "lQ_E", "PLscaled", "PKscaled", "g1",
    "g2", "g3", "g4", "g5"),
  intercept = F, H0 = 0, alpha = 0.05)

# Can also omit one of the dummie vars (g) and include an intercept >> get the same coeffs
reg_IV$ttest
```

effect	coef	std_error	t_stat	p_value	significance
lQ_A	0.397	0.043	9.214	0.0000	Reject
lQ_B	0.648	0.147	4.402	0.0000	Reject

effect	coef	std_error	t_stat	p_value	significance
lQ_C	0.885	0.297	2.976	0.0035	Reject
lQ_D	0.909	0.274	3.321	0.0012	Reject
lQ_E	1.063	0.131	8.091	0.0000	Reject
PLscaled	0.426	0.163	2.608	0.0101	Reject
PKscaled	0.104	0.152	0.681	0.4967	Fail to Reject
g1	-1.815	0.305	-5.952	0.0000	Reject
g2	-2.194	0.489	-4.491	0.0000	Reject
g3	-2.879	0.972	-2.963	0.0036	Reject
g4	-2.922	0.966	-3.024	0.0030	Reject
g5	-3.511	0.599	-5.857	0.0000	Reject

```
reg_IV$moftable
```

	$R^2_{uc}$	$R^2$	$R^2_{adj}$	SIC	AIC	SSR	$s^2$
	0.9802	0.9602	0.9569	-3.701	-3.947	2.372	0.01784

Are there any noticeable changes in returns to scale from the previous part?

*yes...*

**Question 7:**

Conduct a statistical test comparing the first model you estimate to the last model you estimated. (Hint: Is one model a restricted version of the other?). Would separate t-test have given you the same results?

**Question 8:**

To see whether returns to scale declined with output, Nerlove tested a nonlinear specification by including  $\ln(y)^2$  as a regressor. Conduct a statistical test you feel is appropriate to test this hypothesis.