Practicum 3

Kieran Douglas

2025-12-02

2. Replicating Nerlove's Classic Results on Scale Economies

A.

The purpose of this assignment is to replicate some of the principles of returns-to-scale reported by Nerlove in his 1955 article. His estimated equation was as follows:

$lnC^* = \beta_o + \beta_y ln(y) + \beta_1 lnp_1^* + \beta_2 lnp_2^*$

For part A I will generate the variables necesary for estimating his parameters. includes $lnCP3 = ln(COSTS/PF), lnP13 = ln(PL/PF), lnP23 = ln(PK/PF), lnKWH = ln(KWH)$

I will do this using the mutate function in Tidyverse.

```r
# Load packages and data from the NERLOV file
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(readxl)

nerlov <- read_excel("/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_3_files/

# Clean data and generate new variables per assignment request

clean <- nerlov |>
    mutate(
        LNCP3 = log(COSTS/PF),
        LNP13 = log(PL/PF),
        LNP23 = log(PK/PF),
        LNKWH = log(KWH)
        )

# Preview order of LNKWH

print(clean$LNKWH)
```

```
  [1] 0.6931472 1.0986123 1.3862944 1.3862944 1.6094379 2.1972246 2.3978953
  [8] 2.5649494 2.5649494 3.0910425 3.2188758 3.2188758 3.5553481 3.6635616
 [15] 3.7612001 4.1431347 4.2195077 4.3944492 4.4308168 4.2904594 4.5951199
 [22] 4.6151205 4.7791235 4.7874917 4.8040210 4.8675345 4.9272537 5.0039463
 [29] 5.2781147 5.2832037 5.3423343 5.3659760 5.3936275 5.4553211 5.4595855
 [36] 5.5333895 5.6312118 5.6698809 5.6698809 5.6869754 5.7004436 5.7807435
 [43] 5.8081425 5.8230459 5.8664681 5.8664681 6.0306853 6.0402547 6.1224928
 [50] 6.1820849 6.2461068 6.3099183 6.3332796 6.3385941 6.3835066 6.5087691
 [57] 6.5453497 6.5778614 6.6093492 6.6783421 6.6846117 6.6945621 6.6982681
 [64] 6.7511015 6.7569324 6.8123451 6.8167359 6.8287121 6.8916259 6.8987145
 [71] 6.9077553 7.0012456 7.0112140 7.0192967 7.0228681 7.0361485 7.0527210
 [78] 7.0613344 7.0647590 7.1024994 7.1538338 7.1631724 7.1623975 7.1936858
 [85] 7.2247534 7.2584122 7.2957351 7.3112184 7.3427792 7.4079243 7.4193806
 [92] 7.4854916 7.5126175 7.5137092 7.5164333 7.4882935 7.5590383 7.5652753
 [99] 7.6148054 7.6290039 7.6420444 7.7079615 7.7424020 7.7583335 7.7634464
[106] 7.7693786 7.8042514 7.8066964 7.8268421 7.8359746 7.8539931 7.8659554
[113] 7.9620673 8.0040315 8.0715309 8.0974263 8.1053075 8.1599467 8.1713169
[120] 8.2411762 8.2534880 8.2975435 8.3468793 8.3675324 8.4104985 8.4688429
[127] 8.5711130 8.5722494 8.6425916 8.6448826 8.6688837 8.6995147 8.7191540
[134] 8.7219283 8.8808636 8.9728443 9.0382463 9.0643893 9.0810286 9.1573614
[141] 9.2059307 9.3481003 9.3755158 9.5721322 9.7243011
```

```
# The observations are by size of output, looks good!
```

B.

Now that I have all of the data in line, I will estimate the following model:

$lnC^{*} = \beta_o + \beta_y ln(y) + \beta_1 lnp_1^{*} + \beta_2 lnp_2^{*}$

Running this model I find coefficient estimates that slightly differ from those found by Nerlove, with β_y, β_1 and β_2 estimates coming in at 0.7207, 0.5929, and -0.0074, respectively, with standard errors reporting as 0.0174, 0.2046, and 0.1907.

```
model1 <- lm(data = clean, LNCP3~LNKWH+LNP13+LNP23)
summary(model1)
```

```
Call:
lm(formula = LNCP3 ~ LNKWH + LNP13 + LNP23, data = clean)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.01200 -0.21759 -0.00752  0.16048  1.81922 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.690789   0.884871  -5.301 4.34e-07 ***
LNKWH        0.720688   0.017436  41.334  < 2e-16 ***
LNP13        0.592910   0.204572   2.898  0.00435 ** 
LNP23       -0.007381   0.190736  -0.039  0.96919    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3918 on 141 degrees of freedom
Multiple R-squared:  0.9316,    Adjusted R-squared:  0.9301 
F-statistic:   640 on 3 and 141 DF,  p-value: < 2.2e-16
```

C.

I will now construct a confidence interval for β_y. This ends up being (0.67516030.7662147), which means that using the same procedure on repeated samples, we would expect 99% of intervals to contain the true population coefficient. Based on this, we could determine that at the 99% confidence interval, the null hypothesis that $\beta_y = 1$ is rejected, meaning that with a high degree of certainty we could say that values of 1 are not expected as the true population coefficient under this model. This implies that we would reject the null hypothesis that returns to scale were constant, because we can say with a high degree of certainty that they will be increasing since our confidence interval shows that we would expect 99% of intervals (0.68, 0.77) to contain the true population coefficient. If the interval contained 1 we may say that it is plausible for the model to display constant returns to scale, but since it is <1, we can say with a reasonable degree of confidence that in most all cases a 1% increase in the output will raise cost by less than 1% which implies increasing returns to scale. Computing the point estimate for returns to scale, I find an r of 1.388. Since this is greater than 1, we again confirm observation of increasing returns to scale. This means that if we werer to double inputs, outputs would more than double! Since returns to scale are increasing, in this case we have positive economies of scale. This means that as output increases, cost increases at a decreasing rate, or avarage cost falls.

```r
# Construct a 99% confidence interval using the estimated model
# for the coefficient on LNKWH ($beta_y$)
confint(model1, "LNKWH", level = 0.99)
```

```
          0.5 %    99.5 %
LNKWH 0.6751603 0.7662147
```

```r
# compute the point estimate of returns to scale r,
# where r=1/b_y. It is 1.388
1/0.720688
```

```
[1] 1.387563
```

D.

To calculate the implies estimate of α_2 from part C, I have to multiply the coefficient β_2 by the point estimate for returns to scale, r. This yields $-0.007381 * 1.388 = -0.01024$. This estimate is very close to 0, which makes me think that Nerlove was dissatisfied with his estimate of α_2 since it would imply that the second input (capital) was either not effective or negatively productive which wouldnt really make sense for a necessary input. Additionally, since the estimate is so close to 0, it seems likely hat a confidence interval would contain 0 deeming it non statistically significant which also wouldnt make sense. This finding contradicts what we would expect of Cobb-Douglas, indicating that something weird might be going on with the data we have that is causing it to inaccurately reflect the true production process.

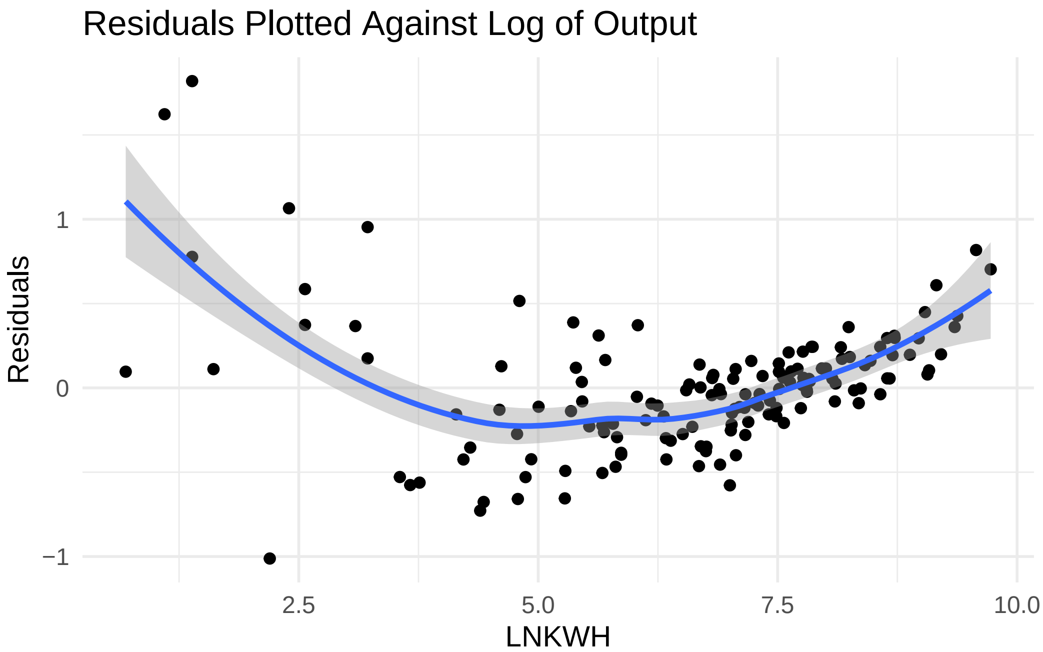
E.

I also find a U-shaped pattern when plotting residuals against LNKWH. I think that this could be a sign of model mispsecification, where the model we are estimating does not adequetly capture the underlying relationship in the data. The implication here is that the true relationship between output and cost is likely nonlinear in some way, and our linearity asusmption is failing. Next, calculating the correlation between the residuals and LNKWH I find a coefficient of effectively 0 (-9.565708e-17). This is not surprising as OLS ensures that residuals are not correlated with regressors in the model by optimizing the fit such that linear association between residuals and regressors is removed. This however is not indicitive of a perfect model, and provides a good example of why its important to check things graphically to ensure that we are getting what we expected.

```r
# create a new row with residuals from the model
clean <- clean |>
    mutate(
        resid = residuals(model1)
    )

# plot against log of output
ggplot(data = clean, mapping = aes(y = resid, x = LNKWH)) +
    geom_point() +
    geom_smooth(method = loess) +
    theme_minimal() +
    labs(title = "Residuals Plotted Against Log of Output", x = "LNKWH", y = "Residuals")
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
# compute sample correlation of residuals with LNKWH across the sample
cor(clean$LNKWH, clean$resid)
```

```
[1] -9.565708e-17
```

3. Assessing Alternative Returns-to-Scale Specifications

A.

My attempted replication of Nerlove's results was actually far closer than I was expecting. Both the estimates and the standard errors were within a very small distance of each other, and any spread appeared to be relatively unsystematic, varying from being greater than or less than with within and between subsets. I would estimate that any observed discrepencies are probably due to the fact that I used natural logarithms in my transformations while in the reported data from Nerlove common logarithms were used.

```
# generate subsamples of data for each set of 29 rows
ss1 <- clean |>
    filter(
        ORDER <200
    )
```

```r
ss2 <- clean |>
    filter(
        ORDER > 200,
        ORDER < 300
    )
ss3 <- clean |>
    filter(
        ORDER > 300,
        ORDER < 400
    )
ss4 <- clean |>
    filter(
        ORDER >400,
        ORDER <500
    )
ss5 <- clean |>
    filter(
        ORDER > 500
    )
# now I will estimate a model for each of these subsets
# and extract the coefficients to combine in a table
library(broom)

ss1_model <- lm(data = ss1, LNCP3~LNKWH+LNP13+LNP23)
ss2_model <- lm(data = ss2, LNCP3~LNKWH+LNP13+LNP23)
ss3_model <- lm(data = ss3, LNCP3~LNKWH+LNP13+LNP23)
ss4_model <- lm(data = ss4, LNCP3~LNKWH+LNP13+LNP23)
ss5_model <- lm(data = ss5, LNCP3~LNKWH+LNP13+LNP23)

# create a conbined coefficients vector to use in table
coef_combined <- list(ss1_model, ss2_model, ss3_model, ss4_model, ss5_model)
# create and print tables for each subset
coef_tbl <- lapply(seq_along(coef_combined), function(i) {
  tidy(coef_combined[[i]]) %>%
    mutate(coef_combined = paste0("model", i))
})
combined <- bind_rows(coef_tbl)
print(coef_tbl)
```

```
[[1]]
# A tibble: 4 x 6
  term        estimate std.error statistic   p.value coef_combined
```

```
  <chr>          <dbl>     <dbl>     <dbl>     <dbl> <chr>
1 (Intercept)  -3.34      3.15      -1.06  0.298     model1
2 LNKWH         0.400     0.0845     4.74  0.0000731 model1
3 LNP13         0.615     0.729      0.843 0.407     model1
4 LNP23        -0.0814    0.706     -0.115 0.909     model1

[[2]]
# A tibble: 4 x 6
  term        estimate std.error statistic    p.value coef_combined
  <chr>          <dbl>     <dbl>     <dbl>      <dbl> <chr>
1 (Intercept)  -6.49       1.41     -4.59  0.000107   model2
2 LNKWH         0.658      0.116     5.66  0.00000684 model2
3 LNP13         0.0938     0.274     0.342 0.735      model2
4 LNP23         0.378      0.277     1.37  0.184      model2

[[3]]
# A tibble: 4 x 6
  term        estimate std.error statistic   p.value coef_combined
  <chr>          <dbl>     <dbl>     <dbl>     <dbl> <chr>
1 (Intercept)   -7.33      1.69      -4.34 0.000205  model3
2 LNKWH          0.938     0.198      4.74 0.0000732 model3
3 LNP13          0.402     0.199      2.02 0.0546    model3
4 LNP23          0.250     0.187      1.34 0.193     model3

[[4]]
# A tibble: 4 x 6
  term        estimate std.error statistic       p.value coef_combined
  <chr>          <dbl>     <dbl>     <dbl>         <dbl> <chr>
1 (Intercept)  -6.55       1.16     -5.62  0.00000757    model4
2 LNKWH         0.912      0.107     8.48  0.00000000791 model4
3 LNP13         0.507      0.187     2.70  0.0121        model4
4 LNP23         0.0934     0.164     0.569 0.575         model4

[[5]]
# A tibble: 4 x 6
  term        estimate std.error statistic  p.value coef_combined
  <chr>          <dbl>     <dbl>     <dbl>    <dbl> <chr>
1 (Intercept)   -6.71     1.05       -6.42 1.02e- 6 model5
2 LNKWH          1.04     0.0650     16.1  1.09e-14 model5
3 LNP13          0.603    0.197       3.05 5.30e- 3 model5
4 LNP23         -0.289    0.175      -1.66 1.10e- 1 model5
```

B.

Using the subset models I ran in the previous section, I compute point estimates of returns to scale (r=1/b__y) for each of the five subsamples. For subsets 1-5 I find point estimates of returns to scale of 2.500, 1.519, 1.066, 1.096, and 0.961 respectively. These point estimates suggest that as output increases, cost generally decreases suggesting on average positive returns to scale. There are discrepensies that I observe though, as the magnitude at which the returns to scale are increasing are decreasing as output increases. Additionally, in the final subset, we see a switch to what may be either slightly decreasing or constant returns to scale. This indicates that at larger quantities bottlenecks or constraints may be experienced that limit continued gains. I think this suggests an alternative specification that is perhaps broken up by quantity and interpreted in terms of the subsections rather than their aggregate. The implications are that returns to scale may not be consistent in this case (or any really), reflecting the U-shaped residuals plot we observed in the previous section.

```r
# subsample 1
1/0.4
```

```
[1] 2.5
```

```r
#subsample 2
1/0.658
```

```
[1] 1.519757
```

```r
# subsample 3
1/0.938
```

```
[1] 1.066098
```

```r
# subsample 4
1/0.912
```

```
[1] 1.096491
```

```r
#subsample 5
1/1.04
```

```
[1] 0.9615385
```

C.

I think that since the β_y coefficients vary pretty substantially across groups but the $\beta_1 and \beta_2$ coefficients do not as much. Basically, the output causes pretty different cost responses but the responsiveness to different input prices remains relatively stable. Given this, I think that by allowing intercept terms and β_y coefficients to vary while holding the others constant across groups. This should allow for us to better control for the observed heteogeneity, generating a model that is a better specification of our sample given its observed shape in earlier parts. After replicating Nerlove's model by including all of the dummies and their interactions with β_y I was able to achieve almost a perfect copy of his estimates. Comparing mine to Nerlove's, the ones that are the most different are out estimated coefficients on the interaction between the second dummy (those between 200 and 300) in which he estimated 0.651 while I estimated 0.648, and our R^2 which he estimated as 0.95 while I found it to be 0.96. As you can see, these are VERY close to one another and any differences probably have to do again with the fact that I used natural logarithms in transformation while Nerlove used common logarithms. I believe that this also explains why all of my standard errors are slightly smaller than those reported by Nerlove.

```
# I am going to start by creating dummy variables representing each group,
# followed by variables that multiply that dummy by LNKWH to act
# as propper interaction terms and finally rerun the regression
# including our new variables.
# the dummies are called d1-d5
clean <- clean |>
    mutate(
        d1 = ORDER <200,
        d2 = ORDER>=200 & ORDER<300,
        d3 = ORDER>=300 & ORDER<400,
        d4 = ORDER>=400 & ORDER<500,
        d5 = ORDER>=500,
        d1_LNKWH = d1*LNKWH,
        d2_LNKWH = d2*LNKWH,
        d3_LNKWH = d3*LNKWH,
        d4_LNKWH = d4*LNKWH,
        d5_LNKWH = d5*LNKWH
    )
# rerun the model including teh dummy variables
model2 <- lm(data = clean, LNCP3~d1+d2+d3+d4+d5+d1_LNKWH+d2_LNKWH+d3_LNKWH+d4_LNKWH+d5_LNKWH+
summary(model2)
```

```
Call:
```

```
lm(formula = LNCP3 ~ d1 + d2 + d3 + d4 + d5 + d1_LNKWH + d2_LNKWH +
    d3_LNKWH + d4_LNKWH + d5_LNKWH + LNP13 + LNP23, data = clean)

Residuals:
     Min       1Q   Median       3Q      Max
-1.46008 -0.14123  0.00942  0.13061  1.10602

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.08336    1.38018  -5.857 3.51e-08 ***
d1TRUE       3.90353    1.16071   3.363  0.00101 **
d2TRUE       3.03092    1.43979   2.105  0.03716 *
d3TRUE       1.45324    2.36033   0.616  0.53915
d4TRUE       1.35475    2.39642   0.565  0.57281
d5TRUE            NA         NA      NA       NA
d1_LNKWH     0.39688    0.04307   9.214 6.04e-16 ***
d2_LNKWH     0.64816    0.14724   4.402 2.18e-05 ***
d3_LNKWH     0.88478    0.29727   2.976  0.00347 **
d4_LNKWH     0.90874    0.27365   3.321  0.00116 **
d5_LNKWH     1.06274    0.13134   8.091 3.26e-13 ***
LNP13        0.42561    0.16317   2.608  0.01014 *
LNP23        0.10373    0.15221   0.681  0.49675
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3075 on 133 degrees of freedom
Multiple R-squared:  0.9602,    Adjusted R-squared:  0.9569
F-statistic:   292 on 11 and 133 DF,  p-value: < 2.2e-16
```

D.

Using the model I estimated in the previous section, I will now use the point estimate returns to scale formula from part B. to compute the implied estimate of returns to scale for each of the five subsamples. From subsamples 1-5 I find point estimate returns to scale of 2.52, 1.54, 1.13, 1.10, and 0.94 respectively. Similar to my findings in part B. we observe increasing returns to scale that decrease with each subsequent output increase. This trend continues until we reach subsample 5 which exhibits decreasing (perhaps constant) returns to scale, which seems to be a common trend as output scales up due to physical constraints. Basically, my estimated scale economies are quite strong when output is very small, but as output grows the benefits diminish. Once we weach the highest reported output point we even notice constand or decreasing average cost per unit meaning that further growth is no longer bringing an advantage as it once did.

```r
# subsample 1
1/0.39688
```

```
[1] 2.519653
```

```r
#subsample 2
1/0.64816
```

```
[1] 1.542829
```

```r
# subsample 3
1/0.88479
```

```
[1] 1.130212
```

```r
# subsample 4
1/0.90874
```

```
[1] 1.100425
```

```r
#subsample 5
1/1.06274
```

```
[1] 0.9409639
```

E.

I will now compute an F-Test to determine whether the constrained model used in part A. fits the data better or worse than the unconstrained model used in part C. Here, my null hypothesis H_0 is that the constraints in part A. create a model that sufficiently explains how cost responds to quantity, while my alternative hypothesis H_1 is that the relaxed model specified in part C. better fits the data in explaining how cost responds to quantity. The F-test in the case of restricted versus unrestricted models comes in the form of $((SSR_{restricted} - SSR_{unrestricted})/q)/(SSR_{unrestricted}/(n - k_{unrestricted})$ where n is the number of observations and k is the number of parameters in the unrestricted model. I will compute this using an ANOVA test in R. From the test, we see that the hypothesis that allowing each group to have its own intercept significantly improves the model fit, rejecting the null. The significantly smaller SSR found in the unrestricted model and the F-stat and its corresponding p-value being VERY close to zero gives my confidence that the unrestircted model does a better job at explaining the sample data. The findings suggest and support the idea that each different group has different intercepts and output elasticities meaning that the effect of output on cost varies by group significantly.

```
# run anova between restricted and unrestricted models
anova(model1, model2)
```

```
Analysis of Variance Table

Model 1: LNCP3 ~ LNKWH + LNP13 + LNP23
Model 2: LNCP3 ~ d1 + d2 + d3 + d4 + d5 + d1_LNKWH + d2_LNKWH + d3_LNKWH +
    d4_LNKWH + d5_LNKWH + LNP13 + LNP23
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1    141 21.640
2    133 12.577  8     9.063 11.98 8.901e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

F.

Running this model gives me estimates that are also quite similar to those of Nerlove. The small differences that exist are likely due to the difference in logs used as noted in previous sections. Similarly, my R^2 is pretty spot on, less than 0.01 away from Nerlove's. Next, I conduct a joint hypothesis test to determine whether returns to scale are constant using a 95% confidence interval. My first null hypothesis holds that returns to scale are nonconstant, that is $\beta_y / = 1 and \beta_{yy} / = 0$, while my other null hypothesis holds that returns to scale are constant, or $\beta_y = 1 and \beta_{yy} = 0$. The joint null hypothesis test yields a massive F-statistic of 252.47 and a p-value that is far below our threshold, meaning we can strongly reject the null hypothesis of constant returns to scale. I think that here the joint hypothesis test gives me a statistical test with greater validity since it evaluates the combined effects of relaxing both restrictions at once which is useful here since the parameters are correlated. The individual t-tests evaluate each parameter seperately which could ignore estimate covariance and lead to inacurate findings. Finally, since the returns to scale in the expanded model vary with the level of output, I compute the implied range of returns to scale estimates using the meadian value of LNY in each of the subsamples. Conputing the r values I find an implied returns of scale for subsets 1-5 of 1.8778161, 1.3498233, 1.1616076, 1.0738059, and 0.9724344 respectively. This reflects earlier findings, though at a slightly lower magnitude towards the higher end, of positive returns to scale diminishing as quantity increases up to the point where it is slightly below constant returns to scale suggesting that further growth in output is no longer bringing the same gains it once did at lower quantities.

```r
# run model3 with the new term
model3 <- lm(LNCP3 ~ LNKWH + I(LNKWH^2) + LNP13 + LNP23, data = clean)
summary(model3)
```

```
Call:
lm(formula = LNCP3 ~ LNKWH + I(LNKWH^2) + LNP13 + LNP23, data = clean)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.38249 -0.13715  0.00814  0.12782  1.13544 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.764648   0.701727  -5.365 3.27e-07 ***
LNKWH        0.152547   0.061860   2.466  0.01487 *  
I(LNKWH^2)   0.050514   0.005364   9.418  < 2e-16 ***
LNP13        0.480586   0.161072   2.984  0.00336 ** 
LNP23        0.074166   0.150016   0.494  0.62181    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3076 on 140 degrees of freedom
Multiple R-squared:  0.9581,	Adjusted R-squared:  0.9569 
F-statistic: 800.7 on 4 and 140 DF,  p-value: < 2.2e-16
```

```r
# run a joint null hypothesis test
library(car)
```

```
Loading required package: carData


Attaching package: 'car'

The following object is masked from 'package:dplyr':

    recode

The following object is masked from 'package:purrr':

    some
```

```r
linearHypothesis(model3, c("LNKWH = 1", "I(LNKWH^2) = 0"))
```

```
Linear hypothesis test:
LNKWH = 1
I(LNKWH^2) = 0

Model 1: restricted model
Model 2: LNCP3 ~ LNKWH + I(LNKWH^2) + LNP13 + LNP23

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    142 61.027
2    140 13.248  2    47.779 252.47 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# calculate values of returns to scale for all 5 subsamples
# to start I find meadian for LNY for each subsample
medians <- c(
  median(ss1$LNKWH, na.rm = TRUE),
  median(ss2$LNKWH, na.rm = TRUE),
  median(ss3$LNKWH, na.rm = TRUE),
  median(ss4$LNKWH, na.rm = TRUE),
  median(ss5$LNKWH, na.rm = TRUE)
)
print(medians)
```

```
[1] 3.761200 5.823046 7.011214 7.707962 8.668884
```

```r
# now that I have my medians I will plug them into the r equation
r_vals <- c(
    1/(0.152547 + 2 * 0.050514 * 3.761200),
    1/(0.152547 + 2 * 0.050514 * 5.823046),
    1/(0.152547 + 2 * 0.050514 * 7.011214),
    1/(0.152547 + 2 * 0.050514 * 7.707962),
    1/(0.152547 + 2 * 0.050514 * 8.668884)
)
print(r_vals)
```

```
[1] 1.8778161 1.3498233 1.1616076 1.0738059 0.9724344
```

4. Comparing Returns-to-Scale Estimates from 1955 with Updated 1970 Data

A.

To start my analysis, I will construct the necesary variables to estimate Nerlove's equation by OLS. I generate the variables LNC70, LNY70, LNP170, and LNP270. Next I find that the sample mean for KWH70 is 8999.727 while that of KWH is just 2133.083, a pretty substantial difference. This indicates that on average firms are generating significantly larger ammounts of energy in 1970 than they were in 1955. Based on this information, we may expect much smaller returns to scale, potentially even decreasing returns to scale. Out previous model showed that as output scaled up the returns to scale diminished. Since we see such a jump in KWH produced between then and the 1970 data, we may expect to see that trend embodied. There is also the possibility that technological improvements and expansion of productive capacity could offset this expectation however, and we could see similar if not greater retruns to scale in 1970 than we did before.

```
# load new update data
update <- read_excel('/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_3_files/
# clean data and add new generated variables
update_clean <- update |>
    mutate(
        LNC70 = log(COST70/PF70),
        LNY70 = log(KWH70),
        LNP170 = log(PL70/PF70),
        LNP270 = log(PK70/PF70)
    )
# compute sample means ot KWH to compare
mean(update_clean$KWH70)
```

```
[1] 8999.727
```

```
mean(clean$KWH)
```

```
[1] 2133.083
```

B.

I estimate the model LNC70 = 0 + yLNY70 + 1LNP170 + 2×LNP270 + u and find a 99% confidence interval which comes out to (0.7923, 0.8584). Based on these results, I can determine that the null hypothesis that $\beta_y = 1$ (indicating constant returns to scale) can be rejected because the 99% confidence interval is well below 1. The difference between out highest CI boundary is more than a couple standard errors away from 1, giving me strong confidence that given this model we can reject the null. Additionally, we can see that since β_y is well below 1, indicating increasing returns to scale. This is backed by the r value of 1.212 which indicates that as quantity increases, we get more bang for our buck through efficiency gains. Compared to the findings in section 2, this is a bit surprising as I find a larger coefficient on β_y which according to previous results shouldnt be the case since quantity increased so much.

```r
# run model
model4 <- lm(data = update_clean, LNC70~LNY70+LNP170+LNP270)
summary(model4)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + LNP170 + LNP270, data = update_clean)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.49086 -0.13036 -0.01245  0.08125  0.86443 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.09237    0.56393 -14.350   <2e-16 ***
LNY70        0.82533    0.01257  65.677   <2e-16 ***
LNP170       0.14521    0.10903   1.332   0.1861    
LNP270       0.16496    0.09803   1.683   0.0957 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.211 on 95 degrees of freedom
Multiple R-squared:  0.9785,    Adjusted R-squared:  0.9778 
F-statistic:  1441 on 3 and 95 DF,  p-value: < 2.2e-16
```

```r
# construct confidence interval
confint(model4, "LNY70", level = 0.99)
```

```
          0.5 %    99.5 %
LNY70 0.7923018 0.8583663
```

```r
# calculate the r value
1/0.82533
```

```
[1] 1.211637
```

C.

Now I add a squared term and run a more generalized verison of the previous model to account for potential nonlinearity, a culprate observed in the 1950's data. This model outputs coefficients for $\beta_y, \beta_{yy}, \beta_1, and\beta_2$ of 0.3014, 0.0367, 0.2941, and 0.0507 respectively and an R^2 of 0.99. Next I run a hypothesis test on the null that returns to scale do not vary with output, or that $\beta_{yy} = 0$ using a 95% condifence interval. To do this I compare the t statistic reported in the regression to the respective p value for the coefficient on I(LNY70^2) which, since the p value is VERY close to 0, allows us to reject the null and determine that there is sufficient evidence to suggest that returns to scale vary with output level. Next I test the joint null hypothesis that returns to scale are constant, that is yy $= 0$ and y $= 1$. I find a significantly smaller RSS than in the constricted model, a massive f statistic and a p value that is effectively 0, allowing me to reject the null and determine that the unrestricted specification is a better fit for the sample. This means that there is strong evidence that the relationship between cost and output is nonlinear. These findings are not mutually exclusive, and in fact complement each other nicely. Both the individual t-test and the joint f-test yield results suggesting that the underlying relationship involves some nonlinear element, and that perhaps the restricted specification is inadequate.

```r
# run the updated model
model5 <- lm(data = update_clean, LNC70~LNY70+I(LNY70^2)+LNP170+LNP270)
summary(model5)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + I(LNY70^2) + LNP170 + LNP270, data = update_clean)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.36949 -0.09177  0.00485  0.07526  0.33890 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -7.10760    0.37778 -18.814  < 2e-16 ***
LNY70        0.30138    0.04678   6.443 4.97e-09 ***
```

```
I(LNY70^2)   0.03675    0.00323  11.376  < 2e-16 ***
LNP170       0.29408    0.07229   4.068 9.86e-05 ***
LNP270       0.05073    0.06471   0.784    0.435
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1376 on 94 degrees of freedom
Multiple R-squared:  0.991, Adjusted R-squared:  0.9906
F-statistic:  2574 on 4 and 94 DF,  p-value: < 2.2e-16
```

```r
# conduct hypothesis test
linearHypothesis(model5, c("LNY70 = 1", "I(LNY70^2) = 0"))
```

```
Linear hypothesis test:
LNY70 = 1
I(LNY70^2) = 0

Model 1: restricted model
Model 2: LNC70 ~ LNY70 + I(LNY70^2) + LNP170 + LNP270

  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     96 12.8296
2     94  1.7794  2     11.05 291.88 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

D.

I will now split the 1970 data into 5 subsamples, where the first 4 groups contain 20 rows and the last group contains 19. Next, I estimate by OLS the parameters of the part B. equation seperately for each group. I find β_y coefficients for groups 1-5 of 0.675, 0.779, 1.141, 0.862, and 1.014 respectively. The corresponding returns to scale estimates are 1.481, 1.284, 0.876, 1.160, and 0.986 respectively. Relative to the 1955 data, the retruns to scale estimates are much more moderate and seem to jump around a bit more. They also do not follow as consistent of a trend as we saw in Nerlove's estimation. In groups 1, 2, and 4 I see increasing returns to scale with r-values greater than 1. For grops 3 and 5 I see decreasing (but almost constant) returns to scale. Relative to Nerlove's findings, the 1970 returns to scale estimates I generated suggest a reduction in scale economies over time due to their smaller size relative to those in the 50's. The larger plants exhibit close to constand or slightly decreasing RTS generally with some exceptions, and RTS do not simply trend downwaards in 1970 like was the case in the

50's. Instead of the U-shape described in earlier sections, these data take on more of an eratic shape with ups and downs as output increases. I think that this makes sense given what I understand about more modern mature industries, whose economies of scale may diminish but can experience expansionary bumps due to factors like technological advancement that allow them to push past physical limitations that may have been prohibiting.

```
# generate subsets of data
ss170 <- update_clean |>
    filter(
        Obs <= 20
    )
ss270 <- update_clean |>
    filter(
        Obs >20 & Obs <=40
    )
ss370 <- update_clean |>
    filter(
        Obs > 40 & Obs <=60
    )
ss470 <- update_clean |>
    filter(
        Obs>60 & Obs<= 80
    )
ss570 <- update_clean |>
    filter(
        Obs>80
    )

# run all models for individual subsets
model6 <- lm(data = ss170, LNC70~LNY70+LNP170+LNP270)
model7 <- lm(data = ss270, LNC70~LNY70+LNP170+LNP270)
model8 <- lm(data = ss370, LNC70~LNY70+LNP170+LNP270)
model9 <- lm(data = ss470, LNC70~LNY70+LNP170+LNP270)
model10 <- lm(data = ss570, LNC70~LNY70+LNP170+LNP270)
summary(model6)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + LNP170 + LNP270, data = ss170)

Residuals:
     Min       1Q   Median       3Q      Max
```

```
-0.40755 -0.24735  0.05733  0.18731  0.41647

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.30757    1.44588  -5.746 3.01e-05 ***
LNY70        0.67502    0.04339  15.558 4.41e-11 ***
LNP170       0.33866    0.28322   1.196    0.249    
LNP270       0.15756    0.21226   0.742    0.469    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2791 on 16 degrees of freedom
Multiple R-squared:  0.944, Adjusted R-squared:  0.9335 
F-statistic: 89.96 on 3 and 16 DF,  p-value: 3.132e-10
```

```r
summary(model7)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + LNP170 + LNP270, data = ss270)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.144232 -0.064771 -0.001432  0.056232  0.170654 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -9.81753    0.97976 -10.020 2.67e-08 ***
LNY70        0.77898    0.09283   8.391 2.97e-07 ***
LNP170       0.54720    0.15262   3.585  0.00247 ** 
LNP270      -0.15884    0.12571  -1.264  0.22448    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1002 on 16 degrees of freedom
Multiple R-squared:  0.8716,    Adjusted R-squared:  0.8475 
F-statistic:  36.2 on 3 and 16 DF,  p-value: 2.321e-07
```

```r
summary(model8)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + LNP170 + LNP270, data = ss370)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.140904 -0.015412  0.008721  0.043109  0.135465 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -11.31902    0.98669  -11.47 3.94e-09 ***
LNY70         1.14123    0.09158   12.46 1.19e-09 ***
LNP170        0.25401    0.10121    2.51   0.0232 *  
LNP270       -0.02767    0.13199   -0.21   0.8366    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08122 on 16 degrees of freedom
Multiple R-squared:  0.9271,    Adjusted R-squared:  0.9134 
F-statistic: 67.81 on 3 and 16 DF,  p-value: 2.581e-09
```

```r
summary(model9)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + LNP170 + LNP270, data = ss470)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.13686 -0.05853 -0.01678  0.04257  0.24054 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.07959    1.78996  -4.514 0.000353 ***
LNY70        0.86206    0.15575   5.535 4.53e-05 ***
LNP170       0.12665    0.14841   0.853 0.406044    
LNP270      -0.07009    0.12698  -0.552 0.588609    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09769 on 16 degrees of freedom
Multiple R-squared:  0.6949,    Adjusted R-squared:  0.6377 
F-statistic: 12.15 on 3 and 16 DF,  p-value: 0.0002141
```

```r
summary(model10)
```

```
Call:
lm(formula = LNC70 ~ LNY70 + LNP170 + LNP270, data = ss570)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.27966 -0.07606  0.01138  0.07105  0.24334 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -10.88381    1.16563  -9.337 1.22e-07 ***
LNY70         1.01374    0.09048  11.205 1.10e-08 ***
LNP170        0.36433    0.16038   2.272   0.0383 *  
LNP270       -0.02695    0.17873  -0.151   0.8822    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1423 on 15 degrees of freedom
Multiple R-squared:  0.9019,    Adjusted R-squared:  0.8823 
F-statistic: 45.98 on 3 and 15 DF,  p-value: 8.474e-08
```

```r
# calculate returns to scale estimates for each
# ss170
1/0.675
```

```
[1] 1.481481
```

```r
# ss270
1/0.779
```

```
[1] 1.283697
```

```r
# ss370
1/1.141
```

```
[1] 0.8764242
```

```r
# ss470
1/0.862
```

```
[1] 1.160093
```

```r
# ss570
1/1.014
```

```
[1] 0.9861933
```

E.

The Christensen-Greene finding can be evaluated on several fronts, several of which I replicated in previous sections. By exploring for myself the returns to scale values for each subsection and comapring them to those found, by considering the context of the findings, and by relating them back to what makes sense intuitively about constraints, I was able to work through the 1070 findings and conclude in agreement with the finding that most U.S electricity was generated by firms who opperated close to the min of their average cost curves. Comparing the 1970 results to the 1955 results, I find returnes to scale estimates that are far closer t0 1 (constant) implying that firms are seeing less in the way of large cost advantaged from further expansion. This is what we would expect of firms as they grow and increase production due to real world limitations like space, time, technology, and regulation. I also generated a plot similar to that in the first section showing residuals plotted against log of output by obs. As noted in part D. the curve is not as nicely U-shaped as the one generated using Nerlove's data, with most firms clustered around the flat bit where the residual value is approx 0 and the LNY70 is between 7 and 10 (on the higher end). This is the area that is directly before firms may start seeing negative returns to scale, which seems like a perfect place to stop pushing output increases. My estimates allign with the Christensen-Greene finding, suggesting that by the 1970s, most of the economies of scale in the industry has been exploited and plants chose to opperate relatively close to the efficient scale since producing less left cash on the table while producing more wouldnt reduce average cost for most.

```r
# compute graphic showing residuals plotted against log of
# output like in the first section to compare shape.
update_clean <- update_clean |>
    mutate(
        resid = residuals(model5)
    )

# plot against log of output
```

```
ggplot(data = update_clean, mapping = aes(y = resid, x = LNY70, color = Obs)) +
    geom_point() +
    geom_smooth(method = loess) +
    theme_minimal() +
    labs(title = "Residuals Plotted Against Log of Output", x = "LNY70", y = "Residuals")
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: The following aesthetics were dropped during statistical transformation:
colour.
i This can happen when ggplot fails to infer the correct grouping structure in
  the data.
i Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```