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.

```
# 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 becor
```

```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 $beta_y$, $beta_1$ and $beta_2$ 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 $beta_y$. 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.

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

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

```
# 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 $alpha_2$ from part C, I have to multiply the coefficient $beta_2$ 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 $alpha_2$ 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.

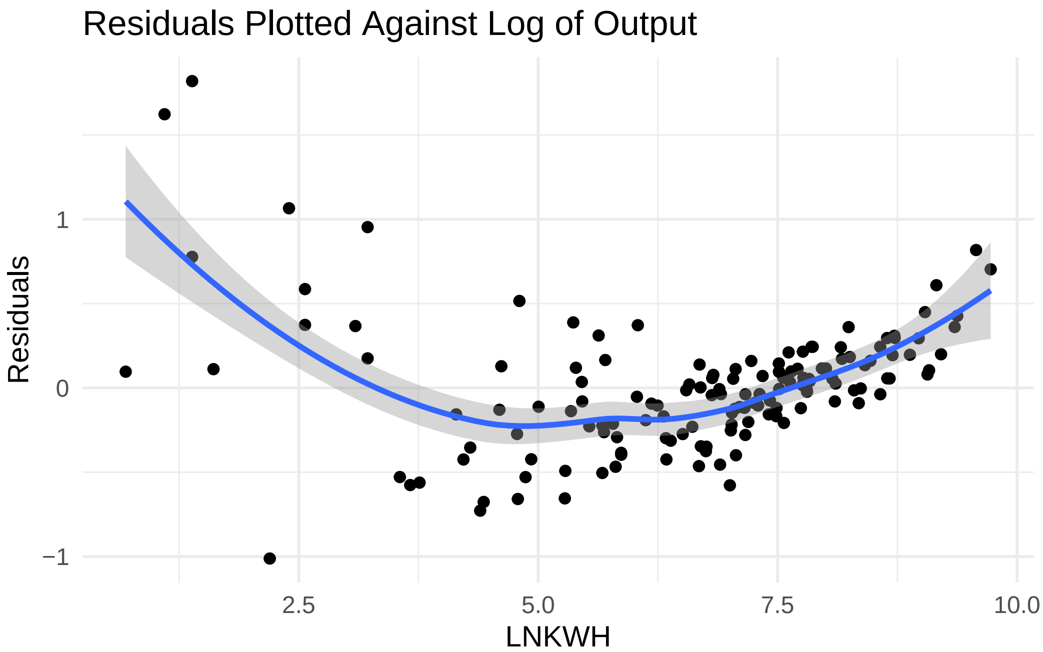
E.

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.

```
# 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
    )
```

```
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 combi
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 $beta_y$ 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 $beta_y$ 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 $beta_y$ 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.

```r
# I am going to start by creating dummy variables representing each group, followed by varia
# 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 constand) returns to scale, which seems to be a common trend as output scales up due to physical constraints. Basicallu, 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 meanign that further growth is no longer bringing an advantage as it once did.

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

```
[1] 2.519653
```

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

```
[1] 1.542829
```

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

```
[1] 1.130212
```

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

```
[1] 1.100425
```

```
#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 $((SSR_{restricted} - SSR_{unrestricted})/q)/(SSR_{unrestricted}/(n - k_{unrestricted})$