

STATS 413 PROBLEM SET 4 SOLUTION

This problem set is due at **noon ET on Oct 7, 2021**. Please upload your solutions to Canvas in two files: a PDF file containing the solutions and a ZIP file containing code that reproduces any computer output in the solutions. You are encouraged to collaborate on problem sets with your classmates, but the final write-up (including any code) **must be your own**.

1. Maximum likelihood estimator of σ^2 . Under the linear model with Gaussian error terms, we saw that the is the maximum likelihood estimator (MLE) of β^* is the OLS estimator $\hat{\beta}$. Show that the MLE of σ^2 is

$$s^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \hat{\beta})^2.$$

Is it an unbiased estimator of σ^2 ?

Solution: $\epsilon \sim N(0, \sigma^2)$ So we have $p(y_i | x_i, \hat{\beta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(y_i - x_i^T \hat{\beta})^2}{2\sigma^2}$

the MLE is $L(\hat{\beta}, s^2) = \log(\prod p(y_i | x_i, \hat{\beta})) = -\frac{n}{2} \log(2\pi) - n \log(s) - \frac{1}{2s^2} \sum (y_i - x_i^T \hat{\beta})^2$

$$\frac{\partial L}{\partial s^2} = 0$$

So $s^2 = \frac{1}{n} \sum (y_i - x_i^T \hat{\beta})^2$

It is biased, since the denominator should be degrees of freedom (n - 1)

2. Returns to scale lab. Download `nerlove.csv` from the course website. It has data on

- TC: total costs (in millions of dollars)
- Q: output (in terawatt hours)
- PL: price of labor
- PF: price of fuel
- PK: price of capital

for 145 utilities companies in 1955. Let $\mathbf{c}_i = \text{TC}_i$, $\mathbf{q}_i = \text{Q}_i$, $\mathbf{p}_{l,i} = \text{PL}_i$, $\mathbf{p}_{f,i} = \text{PF}_i$, $\mathbf{p}_{c,i} = \text{PK}_i$.

(a) Fit the restricted model $\mathbf{y}_i = \mathbf{x}_i^T \beta + \epsilon_i$, where

$$(4.1) \quad \mathbf{y}_i = \log \frac{\mathbf{c}_i}{\mathbf{p}_{c,i}}, \quad \mathbf{x}_i = \left[1 \quad \log \mathbf{q}_i \quad \log \frac{\mathbf{p}_{l,i}}{\mathbf{p}_{c,i}} \quad \log \frac{\mathbf{p}_{f,i}}{\mathbf{p}_{c,i}} \right]^T,$$

to all 145 companies, and plot the residuals. Are there any trends in the residuals? What do they suggest about the degree of returns to scale?

(b) To deal with these issues, Nerlove divided the companies into 5 groups of 29 companies, ordered by output, and fit the restricted model separately to each group. Following Nerlove, fit the restricted model separately to each group. In other words, to each group $j = 1, \dots, 5$, we fit the model

$$\mathbf{y}^{(j)} = \mathbf{X}^{(j)} \beta^{(j)} + \epsilon^{(j)},$$

where $\mathbf{y}^{(j)}, \epsilon^{(j)} \in \mathbf{R}^{29}$ are the vectors of responses and error terms of the companies in the group, $\mathbf{X}^{(j)} \in \mathbf{R}^{29 \times 4}$ is the matrix of features of the companies in the group, and $\beta^{(j)} \in \mathbf{R}^4$ is the vector of regression coefficients of the group. What is the trend in the fitted degree of returns to scale as output increases? What is the trend in the estimated error variances as output increases?

- (c) Consider a model where the regression coefficients are different but the error variance is identical across the 5 groups of companies. In matrix notation, the model is

$$\begin{bmatrix} \mathbf{y}^{(1)} \\ \vdots \\ \mathbf{y}^{(5)} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^{(1)} & & \\ & \ddots & \\ & & \mathbf{X}^{(5)} \end{bmatrix} \begin{bmatrix} \beta^{(1)} \\ \vdots \\ \beta^{(5)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}^{(1)} \\ \vdots \\ \boldsymbol{\epsilon}^{(5)} \end{bmatrix}$$

This is a more general model than the model fit in part (a). Fit this model and compare the fitted coefficients to those from part (b).

Hint: It is possible to encode this model using interactions between five dummy variables that indicate group membership and the features.

- (d) Consider a model where only β_0 and β_1 are different across the 5 groups of companies. Fit this model and compare the fitted coefficients to those from part (a). Are the trends in the fitted degree of returns and estimated error variances present?

Lab Solution

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

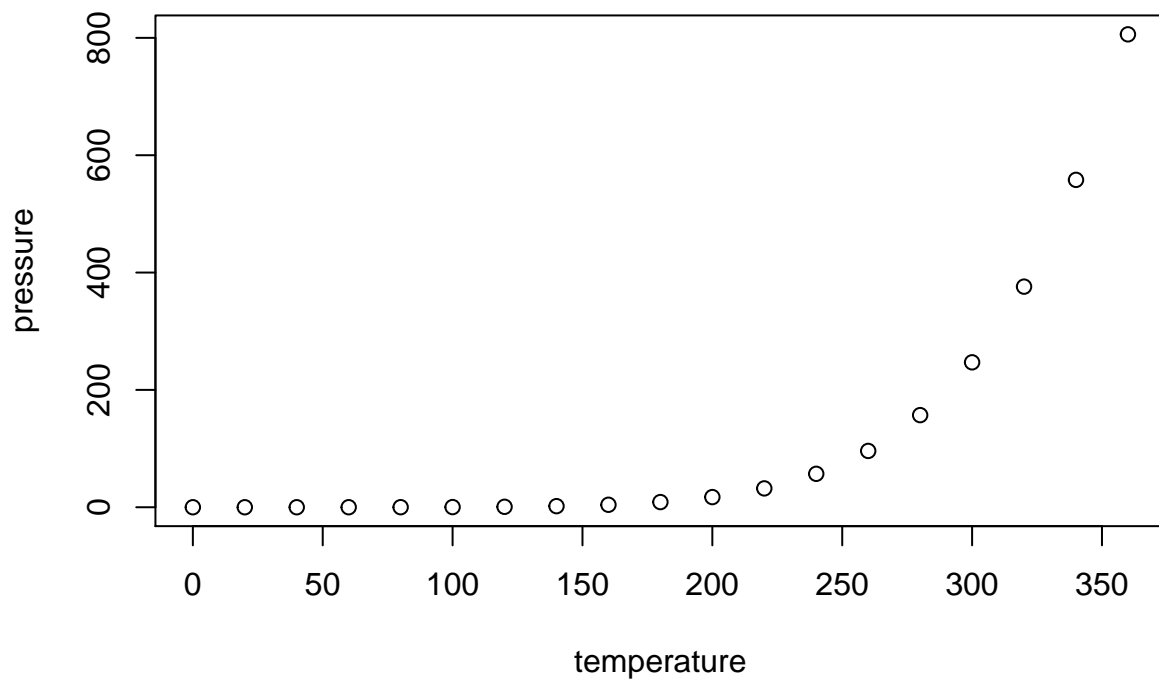
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
## Median :15.0    Median : 36.00
## Mean   :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
## Max.   :25.0    Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

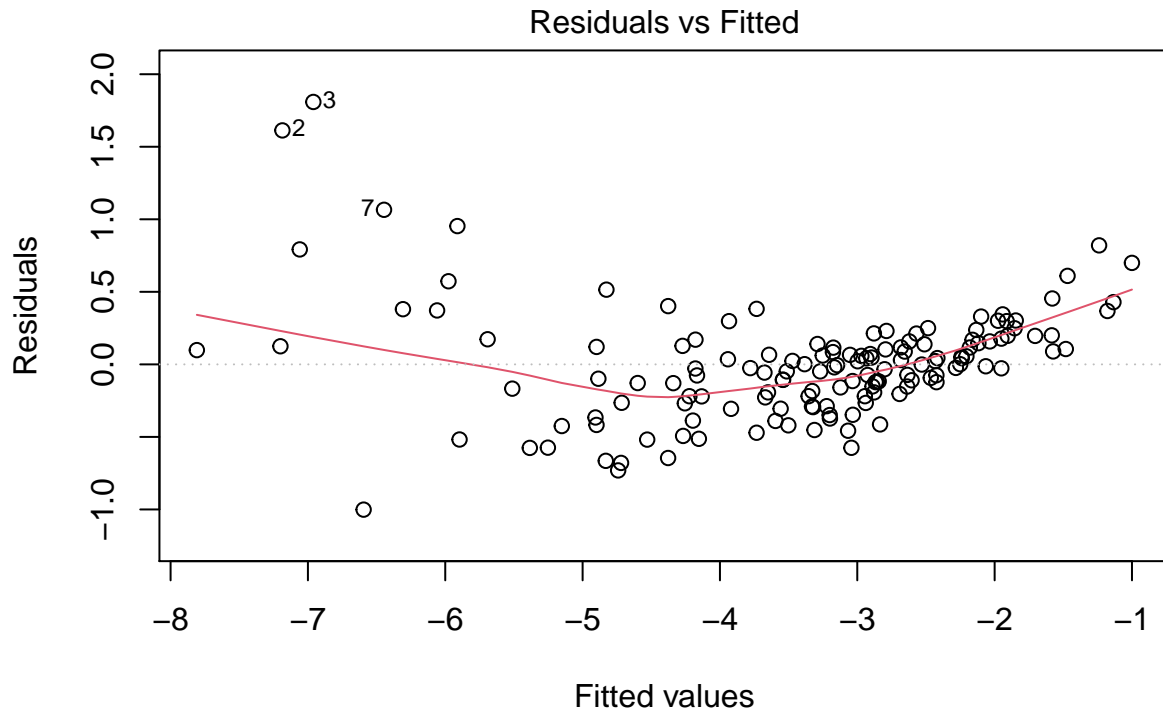
```
library(matlib)
```

```
## Warning: package 'matlib' was built under R version 4.0.5
```

```
data = read.csv('nerlove.csv')
```

```
lm.model = lm(log(TC/PK)~log(Q)+log(PL/PK)+log(PF/PK), data)
```

```
plot(lm.model, which = 1)
```



$$\ln(\log(\text{TC}/\text{PK}) \sim \log(Q) + \log(\text{PL}/\text{PK}) + \log(\text{PF}/\text{PK}))$$

So the residuals are quadratic trends. Indicating that the return to scale is heteroskedasticity

```
group_1 = data[1:29,]
group_2 = data[30:58,]
group_3 = data[59:87,]
group_4 = data[88:116,]
group_5 = data[117:145,]
```

```
lm.model_1 = lm(log(TC/PK)~log(Q)+log(PL/PK)+log(PF/PK), data = group_1)
lm.model_2 = lm(log(TC/PK)~log(Q)+log(PL/PK)+log(PF/PK), data = group_2)
lm.model_3 = lm(log(TC/PK)~log(Q)+log(PL/PK)+log(PF/PK), data = group_3)
lm.model_4 = lm(log(TC/PK)~log(Q)+log(PL/PK)+log(PF/PK), data = group_4)
lm.model_5 = lm(log(TC/PK)~log(Q)+log(PL/PK)+log(PF/PK), data = group_5)
```

```
print(1/lm.model_1$coefficients[2])
```

```
## log(Q)
## 2.48793
```

```
print(1/lm.model_2$coefficients[2])
```

```
## log(Q)
## 1.519019
```

```
print(1/lm.model_3$coefficients[2])
```

```
## log(Q)
## 1.074868
```

```
print(1/lm.model_4$coefficients[2])
```

```
## log(Q)  
## 1.095438
```

```
print(1/lm.model_5$coefficients[2])
```

```
## log(Q)  
## 0.9547541
```

```
print(var(lm.model_1$residuals))
```

```
## [1] 0.3145307
```

```
print(var(lm.model_2$residuals))
```

```
## [1] 0.05257216
```

```
print(var(lm.model_3$residuals))
```

```
## [1] 0.03530706
```

```
print(var(lm.model_4$residuals))
```

```
## [1] 0.01249852
```

```
print(var(lm.model_5$residuals))
```

```
## [1] 0.01962354
```

So return to scale is decreasing and variance of estimated error is decreasing as outputs increase.

```
df = matrix(0, nrow = 145, ncol = 21)
```

```
df[, 1] = log(data$TC/data$PK)
```

```
df[1:29, 2] = rep(1, 29)
```

```
df[1:29, 3] = log(group_1$Q)
```

```
df[1:29, 4] = log(group_1$PL/group_1$PK)
```

```
df[1:29, 5] = log(group_1$PF/group_1$PK)
```

```
df[30:58, 6] = rep(1, 29)
```

```
df[30:58, 7] = log(group_2$Q)
```

```
df[30:58, 8] = log(group_2$PL/group_2$PK)
```

```
df[30:58, 9] = log(group_2$PF/group_2$PK)
```

```
df[59:87, 10] = rep(1, 29)
```

```
df[59:87, 11] = log(group_3$Q)
```

```
df[59:87, 12] = log(group_3$PL/group_3$PK)
```

```
df[59:87, 13] = log(group_3$PF/group_3$PK)
```

```
df[88:116, 14] = rep(1, 29)
```

```
df[88:116, 15] = log(group_4$Q)
```

```
df[88:116, 16] = log(group_4$PL/group_4$PK)
```

```
df[88:116, 17] = log(group_4$PF/group_4$PK)
```

```
df[117:145, 18] = rep(1, 29)
```

```
df[117:145, 19] = log(group_5$Q)
```

```
df[117:145, 20] = log(group_5$PL/group_5$PK)
```

```

df[117:145, 21] = log(group_5$PF/group_5$PK)

lm.model_general = lm(df[,1]~0+df[,-1])

summary(lm.model_general)

##
## Call:
## lm(formula = df[, 1] ~ 0 + df[, -1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44342 -0.11692 -0.00686  0.12871  1.08446
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## df[, -1]1  -3.00535     1.62628  -1.848 0.066967 .
## df[, -1]2   0.40194     0.04422   9.090 1.88e-15 ***
## df[, -1]3   0.70017     0.37957   1.845 0.067454 .
## df[, -1]4   0.44678     0.17002   2.628 0.009668 **
## df[, -1]5  -6.59152     1.81444  -3.633 0.000408 ***
## df[, -1]6   0.65832     0.14988   4.392 2.37e-05 ***
## df[, -1]7   0.07027     0.35080   0.200 0.841558
## df[, -1]8   0.53042     0.19022   2.788 0.006125 **
## df[, -1]9  -7.23763     2.71904  -2.662 0.008793 **
## df[, -1]10  0.93035     0.31239   2.978 0.003484 **
## df[, -1]11  0.40941     0.32863   1.246 0.215164
## df[, -1]12  0.35313     0.24481   1.442 0.151674
## df[, -1]13 -6.33692     3.05961  -2.071 0.040402 *
## df[, -1]14  0.91288     0.27802   3.283 0.001330 **
## df[, -1]15  0.55415     0.49916   1.110 0.269056
## df[, -1]16  0.40241     0.15451   2.605 0.010315 *
## df[, -1]17 -6.66442     2.15125  -3.098 0.002407 **
## df[, -1]18  1.04739     0.13451   7.787 2.26e-12 ***
## df[, -1]19  0.62143     0.40825   1.522 0.130488
## df[, -1]20  0.68142     0.27583   2.470 0.014843 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.312 on 125 degrees of freedom
## Multiple R-squared:  0.9939, Adjusted R-squared:  0.993
## F-statistic: 1024 on 20 and 125 DF,  p-value: < 2.2e-16

```

So the coefficients are similar to the coefficients we get from Part(b)

```

df = matrix(0, nrow = 145, ncol = 13)

df[, 1] = log(data$TC/data$PK)

df[1:29, 2] = rep(1, 29)
df[1:29, 3] = log(group_1$Q)

df[30:58, 4] = rep(1, 29)
df[30:58, 5] = log(group_2$Q)

```

```

df[59:87, 6] = rep(1, 29)
df[59:87, 7] = log(group_3$Q)

df[88:116, 8] = rep(1, 29)
df[88:116, 9] = log(group_4$Q)

df[117:145, 10] = rep(1, 29)
df[117:145, 11] = log(group_5$Q)

df[, 12] = log(data$PL/data$PK)
df[, 13] = log(data$PF/data$PK)

lm.model_new = lm(df[,1]~0+df[,-1])

summary(lm.model_new)

##
## Call:
## lm(formula = df[, 1] ~ 0 + df[, -1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4524 -0.1370  0.0059  0.1328  1.0978
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## df[, -1]1  -4.07552     0.70970  -5.743 6.04e-08 ***
## df[, -1]2   0.39765     0.04298   9.252 4.88e-16 ***
## df[, -1]3  -4.92085     1.13379  -4.340 2.80e-05 ***
## df[, -1]4   0.64395     0.14695   4.382 2.36e-05 ***
## df[, -1]5  -6.46355     2.24171  -2.883  0.00459 **
## df[, -1]6   0.87616     0.29674   2.953  0.00373 **
## df[, -1]7  -6.63637     2.22119  -2.988  0.00335 **
## df[, -1]8   0.91049     0.27300   3.335  0.00111 **
## df[, -1]9  -7.98587     1.37892  -5.791 4.79e-08 ***
## df[, -1]10  1.06413     0.13097   8.125 2.71e-13 ***
## df[, -1]11  0.45123     0.16519   2.732  0.00716 **
## df[, -1]12  0.46730     0.08364   5.587 1.25e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3068 on 133 degrees of freedom
## Multiple R-squared:  0.9938, Adjusted R-squared:  0.9932
## F-statistic: 1764 on 12 and 133 DF,  p-value: < 2.2e-16

```

The trend of fitted degree of returns is also decreasing as outputs increase. However, the estimated error become homoskedasticity