

Practicum 2

Kieran Douglas

2. Examining Waugh's 1927 Asparagus Data

A.

We can see that the greatest difference between coeff estimates and those reported in the question are in the number of stalks and the variation in size (nostalks and disp). These end up being an absolute difference of 0.1767 and 0.0697 respectively.

```
# Set up environment and load data
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 become
```

```
library(readr)
library(fixest)
```

```
waugh <- read_table("/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2_files/v
```

```
-- Column specification -----
cols(
```

```

    GREEN = col_double(),
    NOSTALKS = col_double(),
    DISPERSE = col_double(),
    PRICE = col_double()
)

```

Warning: 200 parsing failures.

```

row col expected actual
  1 -- 4 columns 5 columns '/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2
  2 -- 4 columns 5 columns '/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2
  3 -- 4 columns 5 columns '/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2
  4 -- 4 columns 5 columns '/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2
  5 -- 4 columns 5 columns '/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2
... ..
See problems(...) for more details.

```

```

waugh_clean <- waugh |>
  rename(
    green = "GREEN",
    nostalks = "NOSTALKS",
    disp = "DISPERSE",
    price = "PRICE"
  ) |>
  mutate(
    green = as.numeric(green),
    nostalks = as.numeric(nostalks),
    disp = as.numeric(disp),
    price = as.numeric(price)
  )

# Run a MLR
model1 = lm(price ~ green+nostalks+disp, data = waugh_clean)
summary(model1)

```

Call:

```
lm(formula = price ~ green + nostalks + disp, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-56.004	-9.485	-0.122	9.422	49.097

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	40.761264	5.327837	7.651	8.82e-13	***
green	0.137598	0.007099	19.382	< 2e-16	***
nostalks	-1.357256	0.150822	-8.999	< 2e-16	***
disp	-0.345283	0.129656	-2.663	0.00839	**

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.52 on 196 degrees of freedom

Multiple R-squared: 0.7268, Adjusted R-squared: 0.7226

F-statistic: 173.8 on 3 and 196 DF, p-value: < 2.2e-16

```
# compare coefficient estimates
coeffs = coefficients(model1)
original_coeffs = c(green = 0.13826, nostalks = -1.53394, disp = -0.27554)
differences <- coeffs[names(original_coeffs)] - original_coeffs
print(differences)
```

	green	nostalks	disp
	-0.0006617547	0.1766836397	-0.0697428425

We can see that the greatest difference between coeff estimates and those reported in the

B.

It looks like the main issue here is that Waugh has transformed green color on the asparagus variable to inches rather than what I have in the raw data, being hundredths of inches. To fix this I need to rescale the variable green by dividing it by 100. After doing this the difference in means is no longer a problem.

```
means <- c(price_avg = mean(waugh_clean$price), green_avg = mean(waugh_clean$green), nostalks_avg = mean(waugh_clean$nostalks), disp_avg = mean(waugh_clean$disp))
print(means)
```

price_avg	green_avg	nostalks_avg	disp_avg
90.095	588.750	19.555	14.875

```
means_reported <- c(price_avg = 90.095, green_avg = 5.8875, nostalks_avg = 19.555, disp_avg = 14.875)
differences_avg <- means[names(means_reported)] - means_reported
print(differences_avg)
```

price_avg	green_avg	nostalks_avg	disp_avg
0.000000e+00	5.828625e+02	-3.552714e-15	0.000000e+00

```
# It looks like the main issue here is that Waugh has transformed green color on the asparagus
waugh_clean <- waugh_clean |>
  mutate(
    green = green/100
  )
# Recheck mean differences
means_recheck <- c(price_avg = mean(waugh_clean$price), green_avg = mean(waugh_clean$green),
print(means_recheck)
```

price_avg	green_avg	nostalks_avg	disp_avg
90.0950	5.8875	19.5550	14.8750

C.

I notice relative variance in the relative size of the covariances reported. For example, the covariance between price and green according to Waugh is ~3430 while the one I found was 3448 (larger). On the other hand, Waugh found a covairance of ~154 for green and disp while I found one of ~180 (mine was smaller). In other cases though mine was larger like with the cov between green and price. I think that the pattern in differences may relate to the way my price variable is coded relative to Waugh's. This would explain the constant differences in price covariance versus the more consistent findings in ohter categories.

```
# I am now going to create a variance covariance matrix similar to Waugh's`
moments_subset <- waugh_clean[, c("price", "green", "nostalks", "disp")]
# to match the layout of the table provided I will modify green to be back to hundredths of a
moments_subset <- moments_subset |>
  mutate(
    green = green*100
  )
# Create the matrix, format it correctly, and print
matrix <- cov(moments_subset)
print(matrix)
```

	price	green	nostalks	disp
price	868.73967	3448.18467	-93.38465	-87.43028
green	3448.18467	24439.38442	-17.09171	-180.05653
nostalks	-93.38465	-17.09171	60.73063	24.92399
disp	-87.43028	-180.05653	24.92399	83.48681

```
matrix_fmt <- formatC(matrix, format="f", digits=1)
matrix_fmt[lower.tri(matrix_fmt)] <- ""
print(noquote(matrix_fmt))
```

	price	green	nostalks	disp
price	868.7	3448.2	-93.4	-87.4
green		24439.4	-17.1	-180.1
nostalks			60.7	24.9
disp				83.5

D.

It is plausible that the principle findings concerning the effects of covariate variation could be affected in terms of the precision and statistical inference of coefficient estimates if there was underestimation or misreporting of variance and covariance among regressors. Despite OLS estimates remaining unbiased under GM, Waugh's under or over-estimated variance and covariance could lead to underestimated standard errors and possibly inaccurate significance. Similarly, cov among the regressors included may inflate the standard errors and lead to reductions in the validity/reliability of estimates. The difference in scaled coefficients between Waugh's and my own analysis do not differ too much, but there are some clear differences. For example, my estimate for the marginal effect of a unit increase in the number of stalks per bunch is about \$0.5 less than Waugh's. I estimate a slightly higher increase in price given a unit increase in green, and a roughly \$0.2 larger decrease in price given a unit increase in size variation. Overall, our estimates are very similar. I find the association between both green and nostalks with price to be highly statistically significant at the $p < 0.001$ level and the association between disp and price to be statistically significant at $0.001 < p < 0.01$ level. All of these observations are ceteris paribus.

```
::: {.cell}
```

```
# run model
coef(model1) * 2.782
```

```
::: {.cell-output .cell-output-stdout}
```

(Intercept)	green	nostalks	disp
113.3978351	0.3827983	-3.7758872	-0.9605769

```
:::
```

```

waugh_scaled <- original_coeffs*2.782
print(waugh_scaled)

```

```

::: {.cell-output .cell-output-stdout}

```

```

      green    nostalks      disp
0.3846393 -4.2674211 -0.7665523

```

```

:::

```

```

# The difference in scaled coefficients between Waugh's and my own analysis do not differ to

```

```

:::

```

E.

To solve this I am going to experiment with a matrix.

```

# my matrix solutions
# Extract covariance matrix of regressors (columns and rows 2 to 4)
xx <- matrix[2:4, 2:4]
# Extract covariance vector of PRICE with regressors (row 1, columns 2 to 4)
xy <- matrix[1, 2:4]
# Compute beta_hat = solve(Sigma_XX) %*% Sigma_XY
beta_hat <- solve(xx) %*% xy
print(beta_hat)

```

```

      [,1]
green    0.1375982
nostalks -1.3572564
disp     -0.3452828

```

```

# Waugh's matrix solutions
vc_mat <- matrix(c(
  1063.64, 3430.89, -100.92, -82.35,
  3430.89, 24317.19, -17.01, -154.54,
  -100.92, -17.01, 61.33, 25.51,
  -82.35, -154.54, 25.51, 83.07
), nrow = 4, byrow = TRUE)

```

```

# names for reference
row_col_names <- c("price","green","nostalks","disp")
dimnames(vc_mat) <- list(row_col_names, row_col_names)
# price-cov covariance vector
Sigma_yX <- vc_mat["price", c("green", "nostalks", "disp")]
# covariate-covariance matrix
Sigma_XX <- vc_mat[c("green", "nostalks", "disp"), c("green", "nostalks", "disp")]
# raw coefficients from matrix
beta_raw <- solve(Sigma_XX, Sigma_yX)
names(beta_raw) <- c("green", "nostalks", "disperse")
# scale
beta_final <- beta_raw * 2.782
beta_final

```

```

      green  nostalks  disperse
0.3847304 -4.1520749 -0.7670883

```

```

# Computing the least squares estimates from Waugh's VC matrix table, we can see that his es

```

3. Exploring Relationships among R^2 , Coefficients of Determination, and Correlation Coefficients

A.

Based on the cor matrix, it is evident that the most highly correlated variables are price and green, price and nostalks, nostalks and disp, and price and disp (ordered from highest to lowest positive and negative correlation). The most orthogonal correlations are between green and nostalks and green and disp because each of their correlations is quite close to zero, thus implying low to no real linear relationship between the variables based on the data we have available.

```

# create cor matrix
order <- c("price", "green", "nostalks", "disp")
cor_matrix <- cor(waugh_clean)
cor_matrix <- cor_matrix[order, order]

# round the matrix
cor_matrix_rounded <- round(cor_matrix, 5)

```

```
# lower triangle with empty strings
cor_matrix_char <- format(cor_matrix_rounded, nsmall = 5)
cor_matrix_char[lower.tri(cor_matrix_char)] <- ""

print(cor_matrix_char, quote = FALSE)
```

	price	green	nostalks	disp
price	1.00000	0.74834	-0.40656	-0.32464
green		1.00000	-0.01403	-0.12605
nostalks			1.00000	0.35003
disp				1.00000

```
# based on the cor matrix, it is evident that the most highly correlated variables are price
```

B.

Comparing the square roots calculated of the R^2 I can see that they are equal except for sign. This is because the correlation coefficient can be negative (between -1 and 1) but the coefficient of determination will always be positive (between 0 and 1) since it's the squared correlation coefficient. If I had run the reverse regressions the R^2 measures would be the same as those from the correct regressions because when dealing with the same two variables, the amount of variation explained in one variable by the other will be constant since again, it is the squared correlation coefficient. Reversing their order has no impact on the amount of variation explained that our R^2 captures.

```
# price model
pricegreenmodel <- lm(data = waugh_clean, price ~ green)
pricenostalksmodel <- lm(data = waugh_clean, price ~ nostalks)
pricedispmode <- lm(data = waugh_clean, price ~ disp)
#price on green
summary(pricegreenmodel)
```

Call:

```
lm(formula = price ~ green, data = waugh_clean)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-64.85	-11.52	1.29	12.55	47.43

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.0275	5.4130	1.298	0.196
green	14.1091	0.8888	15.875	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.6 on 198 degrees of freedom

Multiple R-squared: 0.56, Adjusted R-squared: 0.5578

F-statistic: 252 on 1 and 198 DF, p-value: < 2.2e-16

```
sqrt(0.56)
```

```
[1] 0.7483315
```

```
# price on nostalks  
summary(pricenostalksmodel)
```

Call:

```
lm(formula = price ~ nostalks, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-58.948	-17.317	-3.104	9.538	93.589

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	120.1645	5.1676	23.253	< 2e-16 ***
nostalks	-1.5377	0.2456	-6.262	2.32e-09 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27 on 198 degrees of freedom

Multiple R-squared: 0.1653, Adjusted R-squared: 0.1611

F-statistic: 39.21 on 1 and 198 DF, p-value: 2.32e-09

```
sqrt(0.1653)
```

```
[1] 0.406571
```

```
# price on disp
summary(pricedispmodel)
```

Call:

```
lm(formula = price ~ disp, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-57.153	-19.295	-3.114	15.451	86.272

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	105.6726	3.7826	27.94	< 2e-16 ***
disp	-1.0472	0.2168	-4.83	2.73e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27.95 on 198 degrees of freedom

Multiple R-squared: 0.1054, Adjusted R-squared: 0.1009

F-statistic: 23.33 on 1 and 198 DF, p-value: 2.731e-06

```
sqrt(0.1054)
```

```
[1] 0.3246537
```

C.

I expect that if I add the regressor nostalks to the regression, the R^2 will increase. This is because we have separately confirmed that variation in nostalks explains some of the variation in price, so if we add it to the regression, the model's explanatory power should increase. Given the correlation between green and nostalks in the table I would expect the change in R^2 when nostalks is added to the regression to be large. My intuition is such that since the two variables are not super correlated BUT are individually correlated with price, they should explain a significant amount of the variation in price when both included as regressors. Running the regression confirms my intuition, as the R^2 is increased by nearly 15 points, representing a significant jump. Further, comparing the R^2 between the model that regressed price on green to the model that regressed price on green AND disp, we can see another increase in the R^2 equating to about 5 points. This represents an interesting jump, despite it being smaller in magnitude to the previous change in R^2 . I think this is consistent with the sample correlation

between green and disp, which is bigger than that between green and nostalks but still not so big as to fail in adding additional explanation to the model. Finally, when comparing the price on nostalks model to the price on nostalks AND disp model, we also observe a jump in R^2 that is about 4 points. This is again smaller than previous jumps and can probably be explained by the higher correlation coefficient between nostalks and disp that comes out to 0.35003, an R that would suggest higher correlation between the two explanatory variables and thus a smaller increase to the R^2 when both included in a model explaining price. This is consistent with the previous findings regarding the effect of adding correlated explanatory variables to a model. I think this is why would prioritize using adjusted R^2 since that measure penalizes for adding less relevant variables.

```
green_nostalks_model <- lm(data = waugh_clean, price~green+nostalks)
summary(green_nostalks_model)
```

Call:

```
lm(formula = price ~ green + nostalks, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-54.507	-9.997	0.746	10.008	50.997

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	36.9431	5.2100	7.091	2.33e-11 ***
green	14.0043	0.7148	19.593	< 2e-16 ***
nostalks	-1.4983	0.1434	-10.449	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.76 on 197 degrees of freedom

Multiple R-squared: 0.7169, Adjusted R-squared: 0.714

F-statistic: 249.5 on 2 and 197 DF, p-value: < 2.2e-16

```
summary(pricegreenmodel)
```

Call:

```
lm(formula = price ~ green, data = waugh_clean)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-64.85	-11.52	1.29	12.55	47.43

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.0275	5.4130	1.298	0.196
green	14.1091	0.8888	15.875	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.6 on 198 degrees of freedom

Multiple R-squared: 0.56, Adjusted R-squared: 0.5578

F-statistic: 252 on 1 and 198 DF, p-value: < 2.2e-16

```
green_disp_model <- lm(data = waugh_clean, price~green+disp)
summary(green_disp_model)
```

Call:

```
lm(formula = price ~ green + disp, data = waugh_clean)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-64.610	-9.949	1.301	10.767	46.273

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	21.5318	5.7871	3.721	0.000259 ***
green	13.5529	0.8414	16.108	< 2e-16 ***
disp	-0.7549	0.1440	-5.244	4.03e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.41 on 197 degrees of freedom

Multiple R-squared: 0.6139, Adjusted R-squared: 0.61

F-statistic: 156.6 on 2 and 197 DF, p-value: < 2.2e-16

```
summary(pricegreenmodel)
```

Call:

```
lm(formula = price ~ green, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-64.85	-11.52	1.29	12.55	47.43

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.0275	5.4130	1.298	0.196
green	14.1091	0.8888	15.875	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.6 on 198 degrees of freedom

Multiple R-squared: 0.56, Adjusted R-squared: 0.5578

F-statistic: 252 on 1 and 198 DF, p-value: < 2.2e-16

```
nostalks_disp_model <- lm(data = waugh_clean, price~nostalks+disp)
summary(nostalks_disp_model)
```

Call:

```
lm(formula = price ~ nostalks + disp, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-50.080	-16.153	-3.619	10.968	88.859

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	124.7557	5.2794	23.631	< 2e-16 ***
nostalks	-1.2626	0.2568	-4.917	1.85e-06 ***
disp	-0.6703	0.2190	-3.061	0.00252 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.44 on 197 degrees of freedom

Multiple R-squared: 0.2032, Adjusted R-squared: 0.1951

F-statistic: 25.12 on 2 and 197 DF, p-value: 1.923e-10

```
summary(pricenostalksmodel)
```

Call:

```
lm(formula = price ~ nostalks, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-58.948	-17.317	-3.104	9.538	93.589

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	120.1645	5.1676	23.253	< 2e-16 ***
nostalks	-1.5377	0.2456	-6.262	2.32e-09 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27 on 198 degrees of freedom

Multiple R-squared: 0.1653, Adjusted R-squared: 0.1611

F-statistic: 39.21 on 1 and 198 DF, p-value: 2.32e-09

D.

Here I notice the same trend, where the R^2 for the full regression is 0.7268 while the summed R^2 from the single regressions is 0.8307. I think this is because in this case, the model accounts for overlaps in explanatory power between the regressors that are not even a question in the single models. I think this is because the multiple regression avoids double counting, providing the joint effect on price.

```
fullmodel <- lm(data = waugh_clean, price~nostalks+green+disp)
summary(fullmodel)
```

Call:

```
lm(formula = price ~ nostalks + green + disp, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-56.004	-9.485	-0.122	9.422	49.097

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	40.7613	5.3278	7.651	8.82e-13 ***
nostalks	-1.3573	0.1508	-8.999	< 2e-16 ***
green	13.7598	0.7099	19.382	< 2e-16 ***
disp	-0.3453	0.1297	-2.663	0.00839 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.52 on 196 degrees of freedom

Multiple R-squared: 0.7268, Adjusted R-squared: 0.7226

F-statistic: 173.8 on 3 and 196 DF, p-value: < 2.2e-16

```
# R^2 of 0.7268
```

```
0.56+0.1653+0.1054
```

```
[1] 0.8307
```

```
# sum of individual R^2 = 0.8307
```

E.

Waugh's interpretation of the coefficient of determination is incorrect. His method of adding up individual coefficients of determination is not the right way to do it, as we know the explanatory variables he is including are not orthogonal. Waugh should have first correctly calculated and second explained that the coefficient of determination is the portion of variance in price that is jointly explained by variance in the explanatory variables, and in the case of this model, it is equal to about 0.72 or 72%.

```
# nostalks = 0.14554
```

```
-1.53394 * (-100.92 / 1063.64)
```

```
[1] 0.1455429
```

```
# green = 0.44597
```

```
0.13826 * (3430.89 / 1063.64)
```

```
[1] 0.4459731
```

```
# disp = 0.02133
-0.27554 * (-82.35 / 1063.64)
```

```
[1] 0.02133308
```

F.

Right off the bat I notice that the R^2 calculated are the same! I think this happened because by regressing price on a vector of fitted values that were fitted on a regression on price, I am effectively decomposing price to simply its predicted and residual parts. As for the intercept coefficient of 0 and the slope coefficient of 1, I think this is due to the fact that the vector of fitted values is a linear transformation of the original regressors, meaning that when I go onto regress price on that vector, the best linear fit for it would be a zero intercept and a 1 slope. This is because the fitted values generated are the closest possible linear predictors of the observed data.

```
fullmodel <- lm(data = waugh_clean, price~nostalks+green+disp)
summary(fullmodel)
```

Call:

```
lm(formula = price ~ nostalks + green + disp, data = waugh_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-56.004	-9.485	-0.122	9.422	49.097

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	40.7613	5.3278	7.651	8.82e-13 ***
nostalks	-1.3573	0.1508	-8.999	< 2e-16 ***
green	13.7598	0.7099	19.382	< 2e-16 ***
disp	-0.3453	0.1297	-2.663	0.00839 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.52 on 196 degrees of freedom

Multiple R-squared: 0.7268, Adjusted R-squared: 0.7226

F-statistic: 173.8 on 3 and 196 DF, p-value: < 2.2e-16


```
fitted <- fitted(fullmodel)
fitmodel <- lm(waugh_clean$price~fitted)
summary(fitmodel)
```

Call:

```
lm(formula = waugh_clean$price ~ fitted)
```

Residuals:

Min	1Q	Median	3Q	Max
-56.004	-9.485	-0.122	9.422	49.097

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.061e-13	4.075e+00	0.00	1
fitted	1.000e+00	4.357e-02	22.95	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.44 on 198 degrees of freedom

Multiple R-squared: 0.7268, Adjusted R-squared: 0.7254

F-statistic: 526.7 on 1 and 198 DF, p-value: < 2.2e-16

4. Assessing the Stability of the Hedonic Price Equation for the First and Second-Generation Computers

A.

Chow's model: $LN_RENT = 0 + 1LN_MEM + 2LN_MULT + 3LN_ACCESS + u$

```
library("readxl")
chow <- read_excel('/Users/kieran/Documents/MASTERS/METRICS/code/metrics/practicum_2_files/P')

# data cleaning and constructions
chow_clean <- chow |>
  rename(
    obs = "Obs",
    volume = "VOLUME",
    rent = "RENT",
    binary = "BINARY",
```

```

    digits = "DIGITS",
    words = "WORDS",
    add = "ADD",
    mult = "MULT",
    access = "ACCESS",
    year = "YEAR",
    order = "ORDER",
    ibmdum = "IBMDUM"
  ) |>
  mutate(
    ln_rent = log(rent),
    ln_mult = log(mult),
    ln_access = log(access),
    ln_add = log(add),
    mem = words*binary*digits,
    ln_mem = log(mem)
  )
# constructing clope coefficients like in 3.a.
# create cor matrix
order_chow <- c("ln_rent", "ln_mem", "ln_mult", "ln_access")
cor_matrix_chow <- cor(chow_clean)
# cor_matrix_chow <- cor_matrix[order_chow, order_chow]

# round the matrix
cor_matrix_rounded_chow <- round(cor_matrix_chow, 5)

# lower triangle with empty strings
cor_matrix_char_chow <- format(cor_matrix_rounded_chow, nsmall = 5)
cor_matrix_char_chow[lower.tri(cor_matrix_char_chow)] <- ""

print(cor_matrix_char_chow, quote = FALSE)

```

	obs	volume	rent	binary	digits	words	add
obs	1.00000	0.21965	0.03746	-0.30001	0.01232	0.29663	-0.29272
volume		1.00000	-0.13971	0.09402	-0.25418	-0.00153	-0.02353
rent			1.00000	-0.12260	0.37107	0.51318	-0.17710
binary				1.00000	-0.64290	-0.13430	0.18879
digits					1.00000	-0.02091	-0.05012
words						1.00000	-0.12115
add							1.00000
mult							
access							

year							
order							
ibmdum							
ln_rent							
ln_mult							
ln_access							
ln_add							
mem							
ln_mem							
	mult	access	year	order	ibmdum	ln_rent	ln_mult
obs	-0.32512	-0.53380	0.98968	1.00000	-0.01696	0.04020	-0.67477
volume	-0.02577	-0.13205	0.21132	0.21965	0.18270	-0.16712	0.04120
rent	-0.18803	-0.30584	0.04531	0.03746	0.35135	0.78977	-0.48891
binary	0.18506	0.05802	-0.30849	-0.30001	0.22482	-0.02370	0.40788
digits	-0.04427	0.06432	0.02960	0.01232	-0.20199	0.16764	-0.20710
words	-0.12831	-0.21607	0.27597	0.29663	0.29011	0.45336	-0.37587
add	0.99334	0.49600	-0.30844	-0.29272	-0.13647	-0.33007	0.51617
mult	1.00000	0.52413	-0.34138	-0.32512	-0.14532	-0.34507	0.53977
access		1.00000	-0.54836	-0.53380	-0.26138	-0.46524	0.72318
year			1.00000	0.98968	-0.02983	0.04176	-0.68294
order				1.00000	-0.01696	0.04020	-0.67477
ibmdum					1.00000	0.33318	-0.12620
ln_rent						1.00000	-0.58474
ln_mult							1.00000
ln_access							
ln_add							
mem							
ln_mem							
	ln_access	ln_add	mem	ln_mem			
obs	-0.70016	-0.62843	0.22658	0.34817			
volume	-0.12977	-0.00407	-0.07879	-0.12772			
rent	-0.40202	-0.46438	0.80777	0.65657			
binary	0.15272	0.39932	-0.19094	-0.20511			
digits	-0.00773	-0.17664	0.36405	0.21697			
words	-0.33875	-0.35789	0.62968	0.52719			
add	0.42748	0.57882	-0.11318	-0.44234			
mult	0.45680	0.59353	-0.12088	-0.46409			
access	0.86526	0.74038	-0.18763	-0.43530			
year	-0.71031	-0.64090	0.22517	0.34420			
order	-0.70016	-0.62843	0.22658	0.34817			
ibmdum	-0.21527	-0.13584	0.14798	0.17801			
ln_rent	-0.55393	-0.59442	0.51569	0.84689			
ln_mult	0.87557	0.96936	-0.41591	-0.65735			

```
ln_access  1.00000  0.88256 -0.30229 -0.57526
ln_add      1.00000 -0.37522 -0.66034
mem          1.00000  0.60344
ln_mem          1.00000
```

```
# now its time to run the constrained and unconstrained models
#constrained data and model
constr_chow <- chow_clean |>
  filter(year %in% c(60, 61, 62, 63, 64, 65))

constrained_model <- lm(data = constr_chow, ln_rent~factor(year)+ln_mem+ln_mult+ln_access)
summary(constrained_model)
```

Call:

```
lm(formula = ln_rent ~ factor(year) + ln_mem + ln_mult + ln_access,
    data = constr_chow)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-0.95543 -0.26930  0.00555  0.21484  0.83803
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.10446    0.31494  -0.332  0.741092
factor(year)61 -0.13980    0.16647  -0.840  0.403770
factor(year)62 -0.48911    0.17377  -2.815  0.006272 **
factor(year)63 -0.59385    0.16610  -3.575  0.000625 ***
factor(year)64 -0.92482    0.16630  -5.561  4.17e-07 ***
factor(year)65 -1.16317    0.16611  -7.003  1.03e-09 ***
ln_mem         0.57933    0.03539  16.369  < 2e-16 ***
ln_mult       -0.06537    0.02841  -2.301  0.024265 *
ln_access     -0.14060    0.02933  -4.794  8.38e-06 ***
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.3842 on 73 degrees of freedom

Multiple R-squared: 0.9084, Adjusted R-squared: 0.8984

F-statistic: 90.52 on 8 and 73 DF, p-value: < 2.2e-16

```

# unconstrained model
by_year <- constr_chow %>%
  group_split(year) %>%
  map(~ lm(ln_rent ~ ln_mem + ln_mult + ln_access, data = constr_chow))

# sums of squares
SSR_R <- sum(resid(constrained_model)^2)
SSR_U <- sum(map_dbl(by_year, ~ sum(resid(.x)^2)))

# now to run the f-test like chow
n <- nrow(constr_chow)
p <- 3L
g <- n_distinct(constr_chow$year)
q <- p * (g - 1)
K_U <- g * (p + 1)
df2 <- n - K_U

# F-statistic and p-value
fstat <- ((SSR_R - SSR_U) / q) / (SSR_U / df2)
pval <- pf(fstat, q, df2, lower.tail = FALSE)

list(
  f = fstat,
  df1 = q,
  df2 = df2,
  p_value = pval,
  SSR_restricted = SSR_R,
  SSR_unrestricted = SSR_U,
  n = n,
  g = g
)

```

```

$f
[1] -3.54138

```

```

$df1
[1] 15

```

```

$df2
[1] 58

```

```

$p_value

```

```
[1] 1
```

```
$SSR_restricted
```

```
[1] 10.77318
```

```
$SSR_unrestricted
```

```
[1] 128.0602
```

```
$n
```

```
[1] 82
```

```
$g
```

```
[1] 6
```