# ARE 212 Problem Set 2

Eleanor Adachi, Karla Neri, Anna Cheyette, Stephen Stack, Aline Adayo

2024-02-01

```r
# Comment out after installing
# install.packages("pacman")

options(scipen = 999)

# Load packages
library(pacman)
p_load(tidyverse, haven, readr, knitr, psych, ggplot2,stats4, stargazer,
       magrittr, qwraps2, Jmisc)

# get directory of current file
current_directory <-
  dirname(dirname(rstudioapi::getSourceEditorContext()$path))
```

## Question 1

```r
# Load data
my_data <- read_dta(file.path(current_directory, "data", "pset2_2024.dta"))
head(my_data)
```

```
## # A tibble: 6 x 28
##    year country      co type   segment domestic firm     brand    loc          qu
##   <dbl> <dbl+lbl> <dbl> <chr>  <dbl+l>    <dbl> <dbl+lb> <dbl+lb> <dbl+l>   <dbl>
## 1  1970 4 [Italy]    15 audi ~ 4 [sta~        0 26 [VW]   2 [Aud~ 4 [Ger~   1308
## 2  1970 4 [Italy]    36 citro~ 1 [sub~        0  4 [Fia~  4 [Cit~ 3 [Fra~  14032
## 3  1970 4 [Italy]    64 fiat ~ 1 [sub~        1  4 [Fia~  7 [Fia~ 5 [Ita~ 168548
## 4  1970 4 [Italy]    71 ford ~ 2 [com~        0  5 [For~  8 [For~ 4 [Ger~  50423
## 5  1970 4 [Italy]    77 ford ~ 3 [int~        0  5 [For~  8 [For~ 1 [Bel~    427
## 6  1970 4 [Italy]   100 innoc~ 1 [sub~        1  8 [DeT~ 11 [Inn~ 5 [Ita~  48684
## # i 18 more variables: pr <dbl>, princ <dbl>, price <dbl>, horsepower <dbl>,
## #   fuel <dbl>, width <dbl>, height <dbl>, weight <dbl>, pop <dbl>, ngdp <dbl>,
## #   ngdpe <dbl>, country1 <dbl>, country2 <dbl>, country3 <dbl>,
## #   country4 <dbl>, country5 <dbl>, yearsquared <dbl>, luxury <dbl>
```

```r
# Create new variables
my_data <-
  mutate(my_data,
         logprice=log(price),
         logqu=log(qu),
         carspc=qu/pop)
```

# Question 2

```r
# Get summary statistics for data
describe(my_data)
```

```
##             vars  n              mean        sd            median
## year           1 57           1970.00      0.00           1970.00
## country        2 57              4.00      0.00              4.00
## co             3 57            355.32    153.94            413.00
## type*          4 57             29.00     16.60             29.00
## segment        5 57              2.42      1.29              2.00
## domestic       6 57              0.26      0.44              0.00
## firm           7 57             13.05     10.25             12.00
## brand          8 57             16.26     13.99             11.00
## loc            9 57              4.42      1.99              4.00
## qu            10 57          22737.09  53242.59           3387.00
## pr            11 57        1394877.19 600664.93        1265000.00
## princ         12 57              1.11      0.48              1.01
## price         13 57             21.29      9.17             19.31
## horsepower    14 57             53.43     24.54             51.50
## fuel          15 57              8.70      2.10              8.60
## width         16 57            159.96     11.16            159.00
## height        17 57            142.29      5.26            142.00
## weight        18 57            923.21    218.48            925.00
## pop           19 57       53660000.00      0.00       53660000.00
## ngdp          20 57 67177998712832.00      0.00 67177998712832.00
## ngdpe         21 57     1099200000.00      0.00     1099200000.00
## country1      22 57              0.00      0.00              0.00
## country2      23 57              0.00      0.00              0.00
## country3      24 57              0.00      0.00              0.00
## country4      25 57              1.00      0.00              1.00
## country5      26 57              0.00      0.00              0.00
## yearsquared   27 57        3880900.00      0.00        3880900.00
## luxury        28 57              0.05      0.23              0.00
## logprice      29 57              2.98      0.40              2.96
## logqu         30 57              8.59      1.71              8.13
## carspc        31 57              0.00      0.00              0.00
##                      trimmed       mad               min               max
## year                 1970.00      0.00           1970.00           1970.00
## country                 4.00      0.00              4.00              4.00
## co                    368.30    114.16             15.00            544.00
## type*                  29.00     20.76              1.00             57.00
## segment                 2.34      1.48              1.00              5.00
## domestic                0.21      0.00              0.00              1.00
## firm                   12.32     11.86              1.00             33.00
## brand                  14.83     13.34              1.00             46.00
## loc                     4.06      1.48              1.00             12.00
## qu                  11735.70   4158.69            368.00         351477.00
## pr                1320744.68 496671.00         520000.00        3300000.00
## princ                   1.05      0.40              0.42              2.64
## price                  20.16      7.58              7.94             50.37
## horsepower             51.81     25.95             13.00            118.00
## fuel                    8.61      2.22              5.30             15.00
## width                 160.06      8.90            132.00            180.50
```

```
## height                142.33      4.45             127.00             155.00
## weight                916.04    229.80             520.00            1510.00
## pop              53660000.00      0.00        53660000.00        53660000.00
## ngdp        67177998712832.00      0.00 67177998712832.00 67177998712832.00
## ngdpe           1099200000.00      0.00      1099200000.00      1099200000.00
## country1                0.00      0.00               0.00               0.00
## country2                0.00      0.00               0.00               0.00
## country3                0.00      0.00               0.00               0.00
## country4                1.00      0.00               1.00               1.00
## country5                0.00      0.00               0.00               0.00
## yearsquared       3880900.00      0.00         3880900.00         3880900.00
## luxury                  0.00      0.00               0.00               1.00
## logprice                2.96      0.41               2.07               3.92
## logqu                   8.53      1.80               5.91              12.77
## carspc                  0.00      0.00               0.00               0.01
##                  range  skew kurtosis       se
## year              0.00   NaN      NaN     0.00
## country           0.00   NaN      NaN     0.00
## co              529.00 -0.78    -0.82    20.39
## type*            56.00  0.00    -1.26     2.20
## segment           4.00  0.36    -1.21     0.17
## domestic          1.00  1.05    -0.92     0.06
## firm             32.00  0.48    -1.24     1.36
## brand            45.00  0.78    -0.67     1.85
## loc              11.00  2.49     6.87     0.26
## qu           351109.00  4.57    23.68  7052.15
## pr          2780000.00  1.20     1.22 79560.01
## princ             2.22  1.20     1.22     0.06
## price            42.44  1.20     1.22     1.21
## horsepower      105.00  0.54    -0.32     3.25
## fuel              9.70  0.59     0.39     0.28
## width            48.50  0.04    -0.52     1.48
## height           28.00 -0.12     0.38     0.70
## weight          990.00  0.27    -0.50    28.94
## pop               0.00   NaN      NaN     0.00
## ngdp              0.00   NaN      NaN     0.00
## ngdpe             0.00   NaN      NaN     0.00
## country1          0.00   NaN      NaN     0.00
## country2          0.00   NaN      NaN     0.00
## country3          0.00   NaN      NaN     0.00
## country4          0.00   NaN      NaN     0.00
## country5          0.00   NaN      NaN     0.00
## yearsquared       0.00   NaN      NaN     0.00
## luxury            1.00  3.90    13.46     0.03
## logprice          1.85  0.24    -0.38     0.05
## logqu             6.86  0.37    -0.82     0.23
## carspc            0.01  4.57    23.68     0.00
```

```r
# Create summary table
summary_maker <-
  list("Price" =
         list("min" = ~ min(my_data$price),
              "max" = ~ max(my_data$price),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$price)),
```

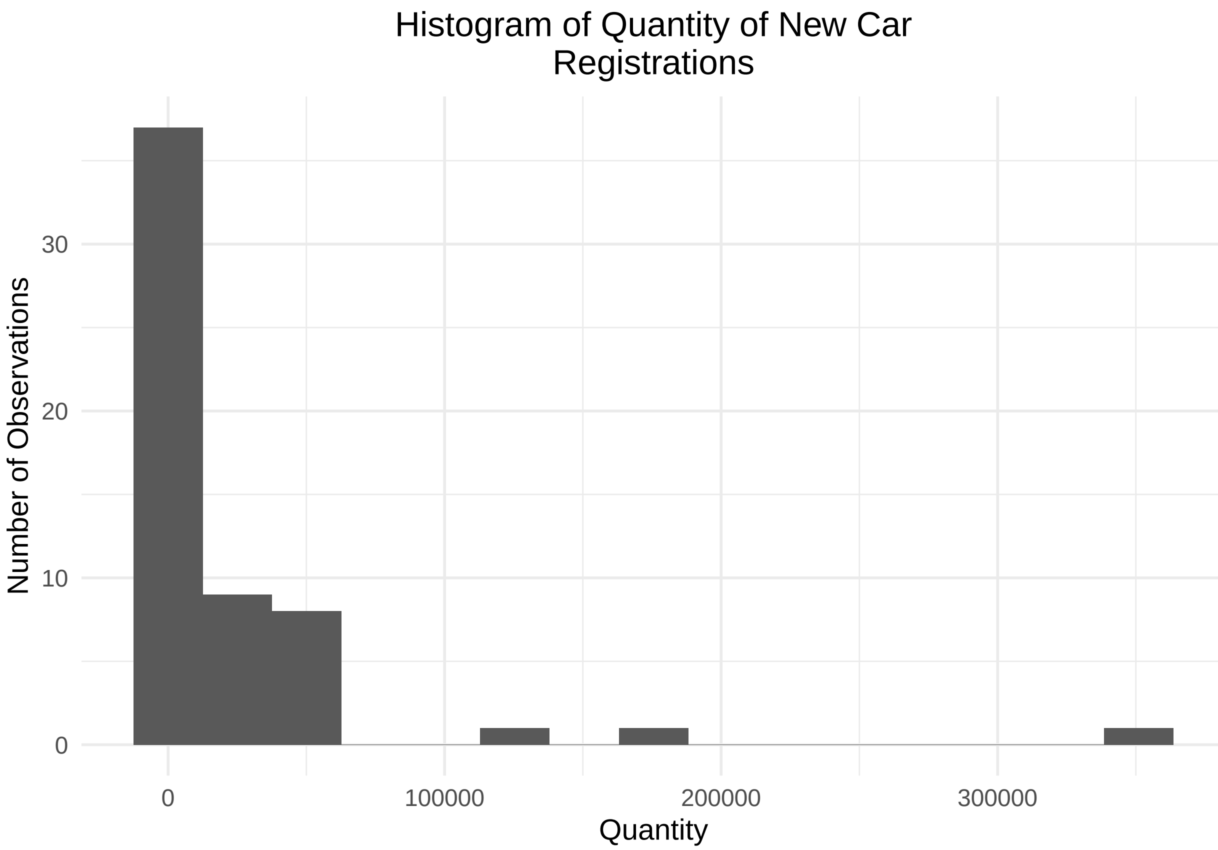
```r
       "Log of Price" =
         list("min" = ~ min(my_data$logprice),
              "max" = ~ max(my_data$logprice),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$logprice)),
       "Quantity" =
         list("min" = ~ min(my_data$qu),
              "max" = ~ max(my_data$qu),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$qu)),
       "Log of Quantity" =
         list("min" = ~ min(my_data$logqu),
              "max" = ~ max(my_data$logqu),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$logqu)))

whole <- summary_table(my_data, summary_maker)
whole
```

| | my_data (N = 57) |
|---|---|
| **Price** | |
| min | 7.93751907348633 |
| max | 50.3727149963379 |
| mean (sd) | 21.29 ± 9.17 |
| **Log of Price** | |
| min | 2.07160076717483 |
| max | 3.91944965936387 |
| mean (sd) | 2.98 ± 0.40 |
| **Quantity** | |
| min | 368 |
| max | 351477 |
| mean (sd) | 22,737.09 ± 53,242.59 |
| **Log of Quantity** | |
| min | 5.90808293816893 |
| max | 12.7698995542371 |
| mean (sd) | 8.59 ± 1.71 |

# Question 3

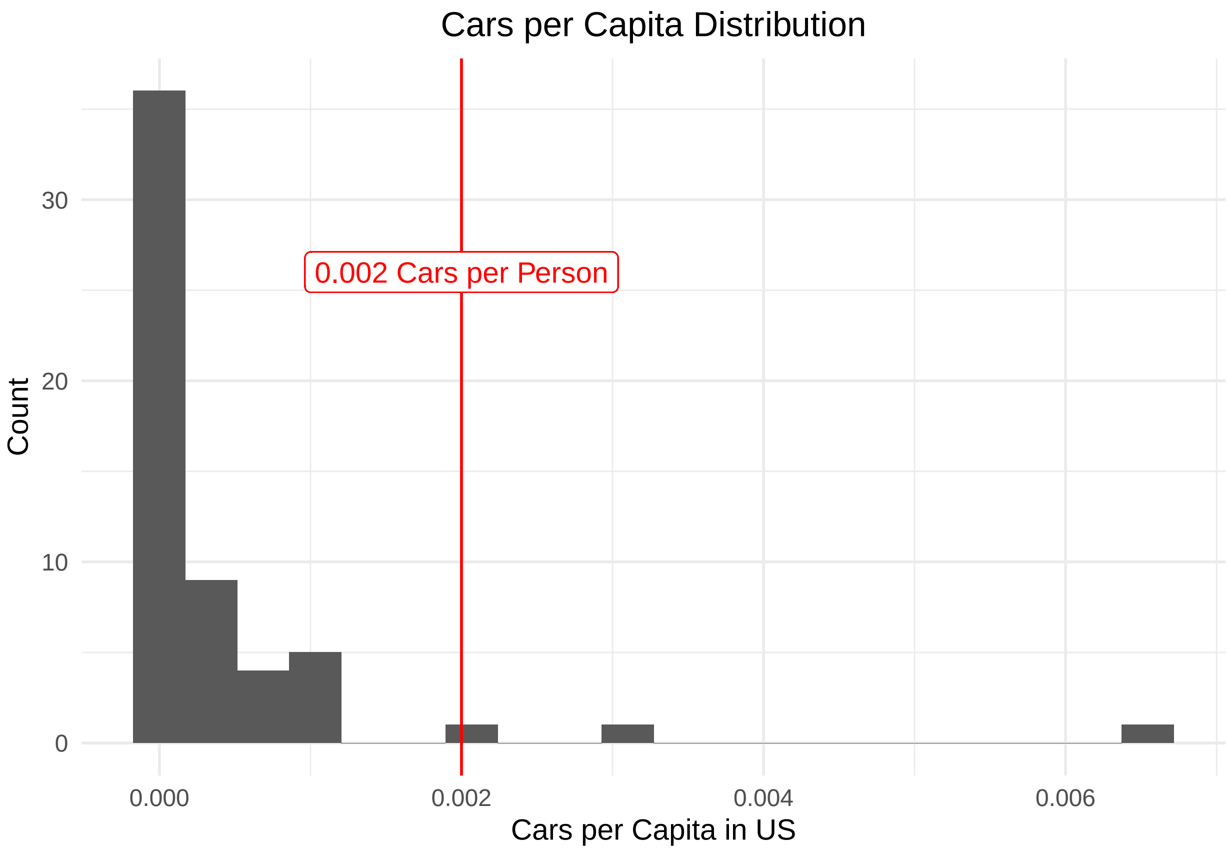
```r
# Make a histogram of qu
histqu <- ggplot(my_data, aes(x=qu)) + geom_histogram(bins=15)
(histqu <- histqu +
    xlab("Quantity") +
    ylab("Number of Observations") +
    ggtitle(str_wrap("Histogram of Quantity of New Car Registrations", 40)) +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5)))
```




## Question 4

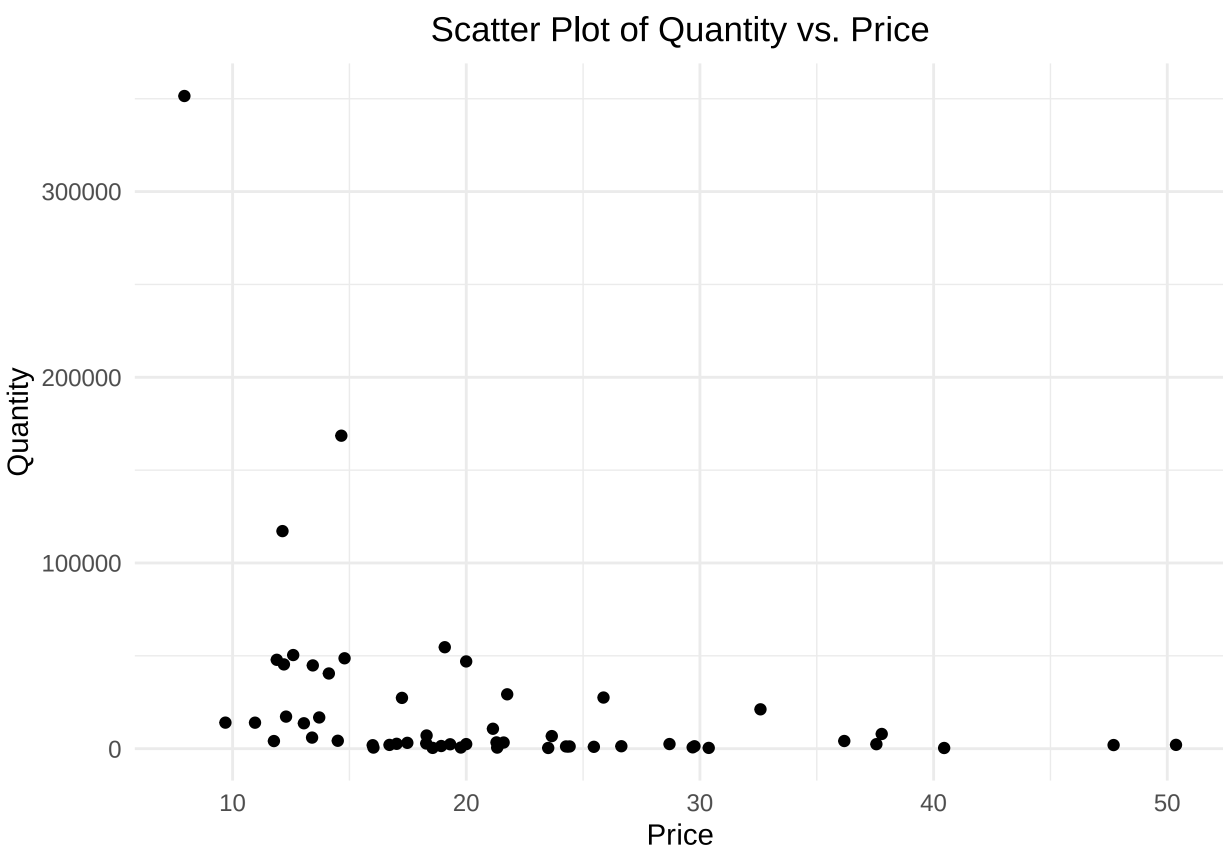
```
# Make a histogram of carspc
histcarspcvertical <-
  ggplot(my_data, aes(carspc)) +
  geom_histogram(bins = 20) +
  geom_vline(xintercept = 0.002, color = "red") +
  geom_label(x = 0.002, y = 26, label = "0.002 Cars per Person", color = "red") +
  labs(title = "Cars per Capita Distribution",
       x = "Cars per Capita in US", y = "Count") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

histcarspcvertical
```

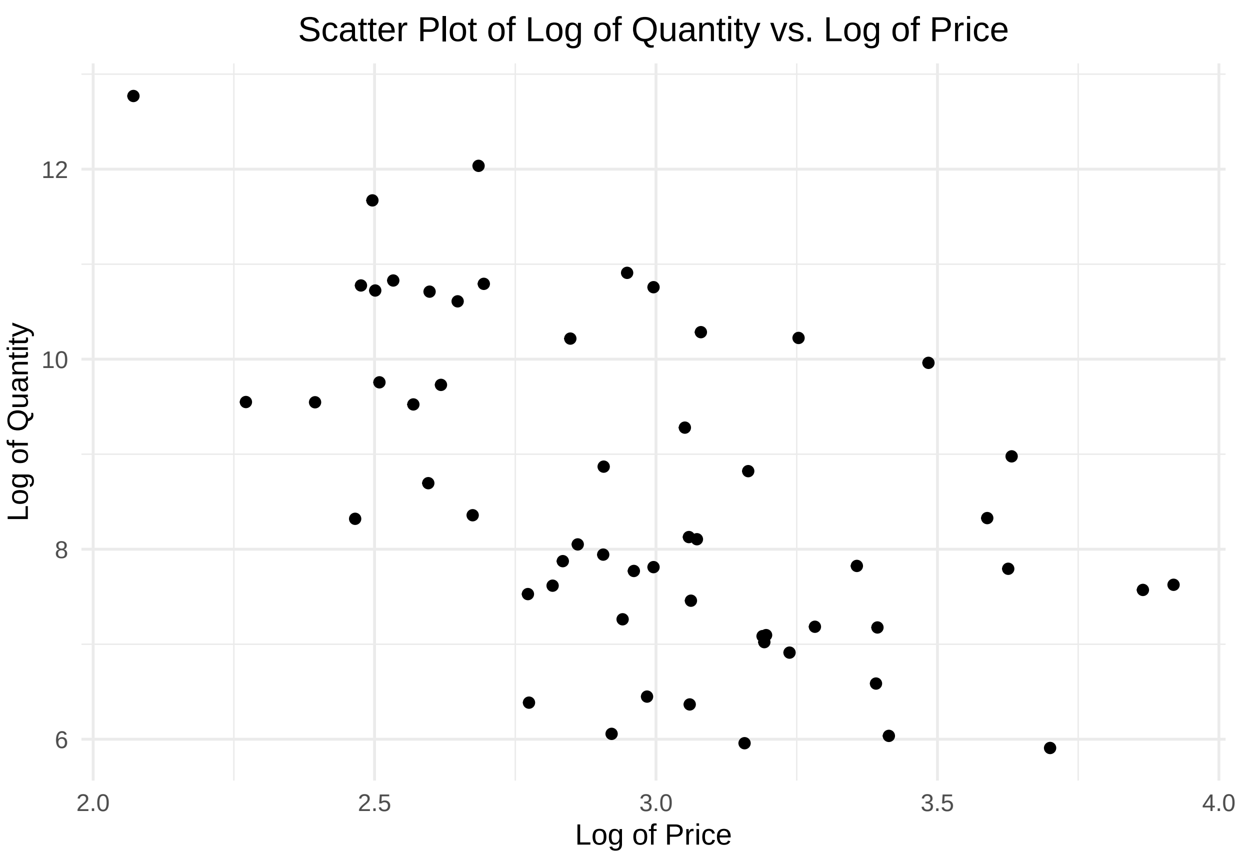



## Question 5

```
# Make scatter plots of price vs. qu and logprice vs. logqu
scatter <- ggplot(my_data, aes(x=price, y=qu)) + geom_point()
(scatter <-
    scatter +
    xlab("Price") +
    ylab("Quantity") +
    ggtitle("Scatter Plot of Quantity vs. Price") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)))
```




```
scatter_logs <- ggplot(my_data, aes(x=logprice, y=logqu)) + geom_point()
(scatter_logs <-
    scatter_logs +
    xlab("Log of Price") +
    ylab("Log of Quantity") +
    ggtitle("Scatter Plot of Log of Quantity vs. Log of Price") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)))
```




## Question 6

```r
# Filter data by luxury
dataluxury <- filter(my_data, luxury==1)
datanoluxury <- filter(my_data, luxury==0)

# Make overlapping histograms for luxury and non-luxury
histprice_luxnolux <-
  ggplot() +
  geom_density(data=dataluxury,
               aes(x=price, fill="r"), alpha = 0.8) +
  geom_density(data=datanoluxury,
               aes(x=price, fill="g"), alpha = 0.8) +
  scale_fill_manual(name="Luxury", values=c("r"="red", "g"="green"),
                    labels=c("r"="Luxury Models", "g"="Non-Luxury Models")) +
  labs(x = "Price", y = "Density",
       title = str_wrap("Density of Price for Luxury vs. Non-Luxury Cars", 40)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

histprice_luxnolux
```


## Question 7

```
# Export data
write.csv(my_data, file="my_data2024.csv")
```

## Question 8

```
# Regress qu on price without constant
x <- my_data$price
y1 <- my_data$qu
# find coefficient
b1 <- solve(t(x) %*% x) %*% t(x) %*% y1

# projection matrix of reg y1 on x
P_1 <- x%*%solve(t(x)%*%x)%*%t(x)
# residual maker of reg y1 on x: M= I - P
M_1 <- diag(57)-P_1
# sum of squared residuals, SSR=e'e
e_1 <- M_1%*%y1
SSR_1 <- t(e_1) %*% e_1


# calculate SST as the sum of the squared values of the dependent
# variable, not relative to its mean bc we do not have a constant.
```

```r
SST_1 <- t(y1) %*% y1

# calculate R squared
Rsquared_1 <- 1-(SSR_1/SST_1)
Rsquared_1
```

```
      [,1]
```

[1,] 0.05670264

```r
# Regress carspc on price without constant
y2 <- my_data$carspc
# find coefficient
b2 <- solve(t(x) %*% x) %*% t(x) %*% y2

# projection matrix of reg y2 on x
P_2 <- x %*% solve(t(x) %*% x) %*% t(x)

# residual maker of reg y2 on x: M= I - P
M_2 <- diag(57)-P_2
# sum of squared residuals, SSR=e'e
e_2 <- M_2%*%y2
SSR_2 <- t(e_2)%*%e_2


# calculate SST as the sum of the squared values of the dependent
# variable, not relative to its mean bc we do not have a constant.
SST_2 <- t(y2) %*% y2

# calculate R squared
Rsquared_2 <- 1-(SSR_2/SST_2)
Rsquared_2
```

```
      [,1]
```

[1,] 0.05670264

```r
# compare coefficients
all.equal(b1, b2)
```

[1] "Mean relative difference: 1"

```r
# compare Rsquared
all.equal(Rsquared_1, Rsquared_2)
```

[1] TRUE

```r
# compare to lm regression
Reg1 <- lm(qu~price-1,my_data)
stargazer(Reg1,
          column.labels = c("Question 8"),
          dep.var.caption = "Dependent Variable: Quantity (New Car Registrations)",
          covariate.labels = "Price in Thousands of Euros",
          header = FALSE,
          title = "Effect of Price on Quantity - No Constant - Regression Using lm() Function")
```

```r
Reg2 <- lm(carspc~price-1,my_data)
stargazer(Reg2,
```

Table 1: Effect of Price on Quantity - No Constant - Regression Using lm() Function

| | Dependent Variable: Quantity (New Car Registrations) |
|---|---|
| | qu |
| | Question 8 |
| Price in Thousands of Euros | 591.060* |
| | (322.152) |
| Observations | 57 |
| $R^2$ | 0.057 |
| Adjusted $R^2$ | 0.040 |
| Residual Std. Error | 56,306.340 (df = 56) |
| F Statistic | 3.366* (df = 1; 56) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

```
          column.labels = c("Question 8"),
          dep.var.caption = "Dependent Variable: Cars per Capita",
          covariate.labels = "Price in Thousands of Euros",
          header = FALSE,
          title = "Effect of Price on Cars per Capity - No Constant - Regression Using lm() Function")
```

Table 2: Effect of Price on Cars per Capity - No Constant - Regression Using lm() Function

| | Dependent Variable: Cars per Capita |
|---|---|
| | carspc |
| | Question 8 |
| Price in Thousands of Euros | 0.00001* |
| | (0.00001) |
| Observations | 57 |
| $R^2$ | 0.057 |
| Adjusted $R^2$ | 0.040 |
| Residual Std. Error | 0.001 (df = 56) |
| F Statistic | 3.366* (df = 1; 56) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

*Report the coefficient on price and compare it to the previous coefficient. Check if they are different in R using all.equal(). Explain your findings.*

The coefficient of quantity regressed on price without a constant is **591.0600136** and the R squared is **0.0567026**.

The coefficient of cars per capita regressed on price without a constant is **0.000011** and the R squared is **0.0567026**.

We are able to find the same coefficients and R-squared values using matrix algebra as with the canned `lm()` function.

The coefficients (b1 and b2) are different but the R-squared values are the same.

Note that finding R-squared without a constant is inherently problematic because it assumes that SST is computed relative to the mean of the dependent variable. For models without an intercept, we replicate what the `lm` function does here by calculating SST as the sum of the squared values of the dependent variable, not relative to its mean. However, this means that this R-squared is no longer a proportion of variation explained.

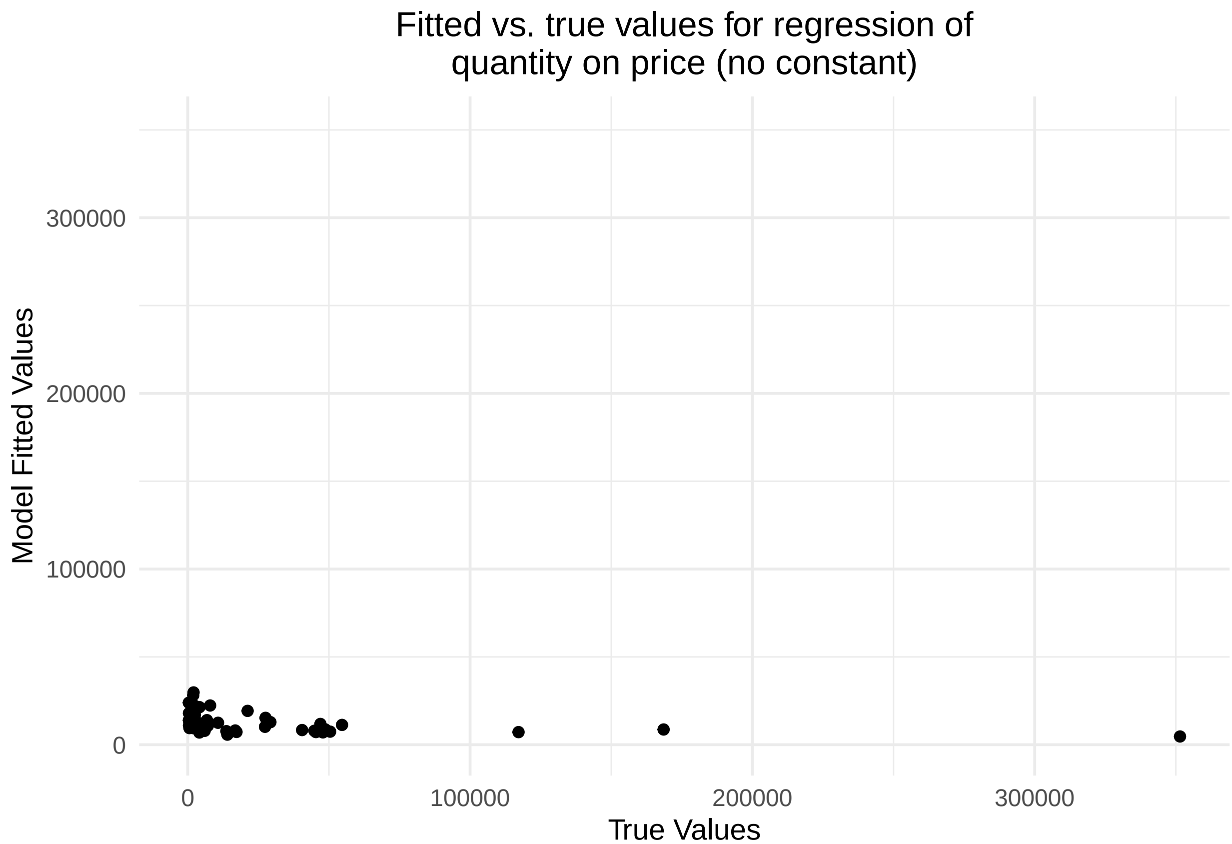
Furthermore, because the R-squared values are the same for regression of $y_1$ on $x$ and regression of $y_2$ on $x$, this suggests that unless $y_1$ and $y_2$ are perfectly correlated (i.e. $y_2$ is linear in $y_1$), this method of calculating R-squared does not depend on the variation in $y$ and only depends on the variation in $x$.

# Question 9

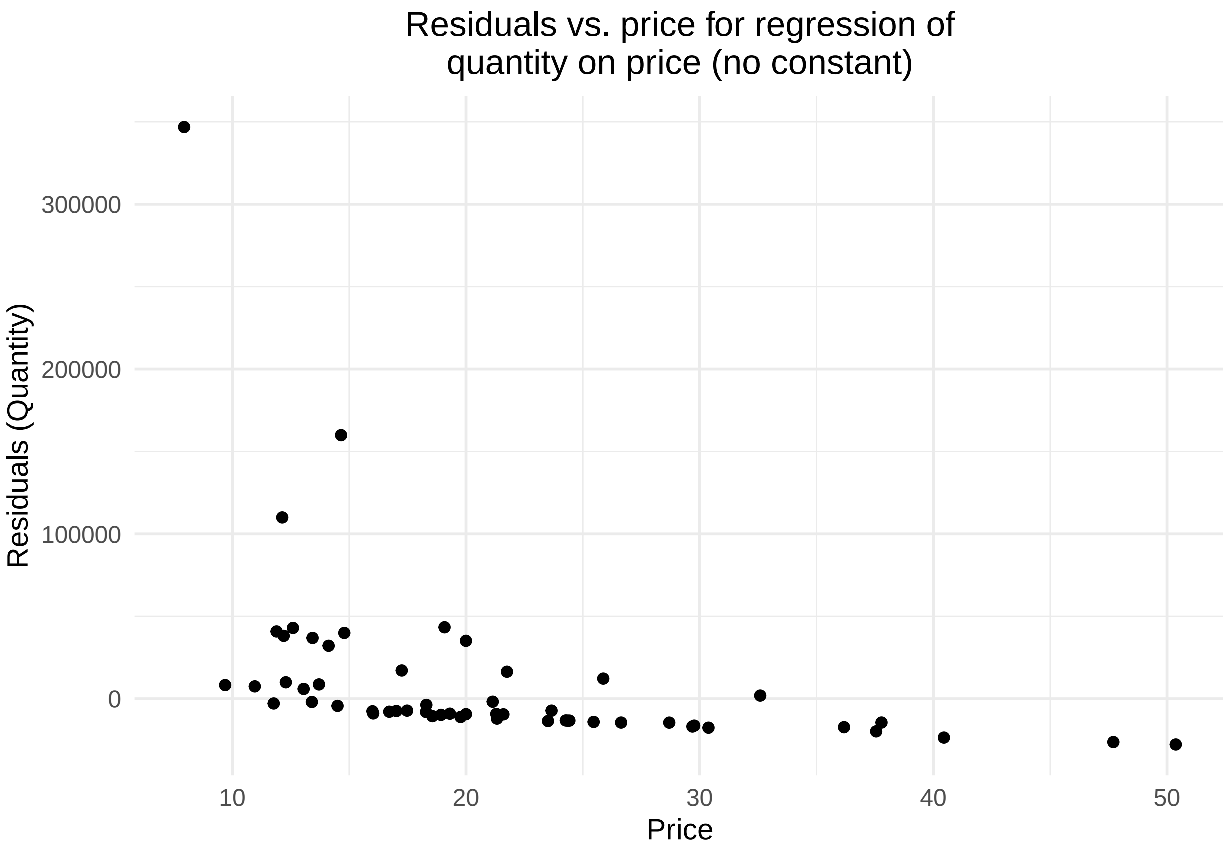
```
# regression of quantity on price
# get degrees of freedom, coefficient, and sample size

# project estimates of y
y1_hat <- P_1%*%y1
# calculate residuals
e <- M_1%*%y1

# plot fitted (predicted) vs. true (observed) quantities
ggplot() +
  # True values on x-axis, fitted values on y-axis
  geom_point(aes(x = y1, y = y1_hat)) +
  labs(x = "True Values",
       y = "Model Fitted Values",
       title = str_wrap(
         "Fitted vs. true values for regression of quantity on price (no constant)",
         40)) +
  ylim(0,max(my_data$qu)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
# plot residuals vs. price
ggplot() +
  # True values on x-axis, residuals on y-axis
  geom_point(aes(x = x, y = e)) +
  labs(x = "Price",
       y = "Residuals (Quantity)",
       title = str_wrap(
         "Residuals vs. price for regression of quantity on price (no constant)",
         40)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




- sample size, n = 57
- number of explanatory variables, k = 1
- degrees of freedom, n - k = 56
- estimate of coefficient, b = 591.0600136

*What do you see in terms of fit and whether the constant variance assumption for the residuals is valid or not?*

In the fitted vs. true values plot, we can see that the points do *not* fall along the 45 degree line. If our model had perfect predictive power, we would see the points falling along the 45-degree line. In our plot, it appears that the model predicts lower values relatively well but significantly underestimates the quantity as the true value increases. In addition, the model's estimates for large values of quantity are extremely poor, although it is possible that these are outliers.

There appears to be a positive linear relationship between the price and the residual.

The key assumption of constant variance (homoscedasticity) in linear regression is that the residuals should be spread randomly around zero, with no clear pattern, and their spread should not change systematically across the range of observed values.

In this plot, the residuals decrease as the price increases, suggesting that the variance of the residuals is not constant — they decrease for higher prices. This pattern of decreasing spread is indicative of heteroscedasticity, which violates the assumption of constant variance in the residuals.

## Question 10

```
# Regress quantity on price and a constant
# add constant
X10 <- cbind(1, x)
```

```r
y10 <- my_data$qu
# find coefficient
b10 <- solve(t(X10)%*%X10)%*%t(X10)%*%y10
b10
```

```
##         [,1]
##    65877.821
## x -2026.143
```

```r
# projection matrix of reg y1 on X
P <- X10%*%solve(t(X10)%*%X10)%*%t(X10)
# residual maker of reg y1 on x: M= I - P
M <- diag(57)-P
# sum of squared residuals, SSR=e'e
e10 <- M%*%y10
SSR <- t(e10)%*%e10

# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y
M0y <- M0%*%y10
# total sum of squares
SST <- t(M0y)%*%M0y

# calculate R squared
Rsquared10 <- 1-(SSR/SST)
Rsquared10
```

```
##           [,1]
## [1,] 0.1217445
```

```r
# project estimates of y
y10_hat <- P%*%y10
y10_hat <- X10%*%b10
```

```r
# check with lm model
Reg10 <- lm(qu~price,my_data)
stargazer(Reg10,
          column.labels = c("Question 10"),
          dep.var.caption = "Dependent Variable: Quantity (New Car Registrations)",
          covariate.labels = "Price in Thousands of Euros",
          header = FALSE,
          title = "Effect of Price on Quantity - Model Using lm Function")
```

```r
# plot fitted (predicted) vs. true (observed) quantities
ggplot() +
  # True values on x-axis, fitted values on y-axis
  geom_point(aes(x = y10, y = y10_hat)) +
  labs(x = "True Values",
       y = "Model Fitted Values",
       title = str_wrap(
         "Fitted vs. true values for regression of quantity on price with constant",
         40)) +
  ylim(NA, max(my_data$qu)) +
  theme_minimal() +
```

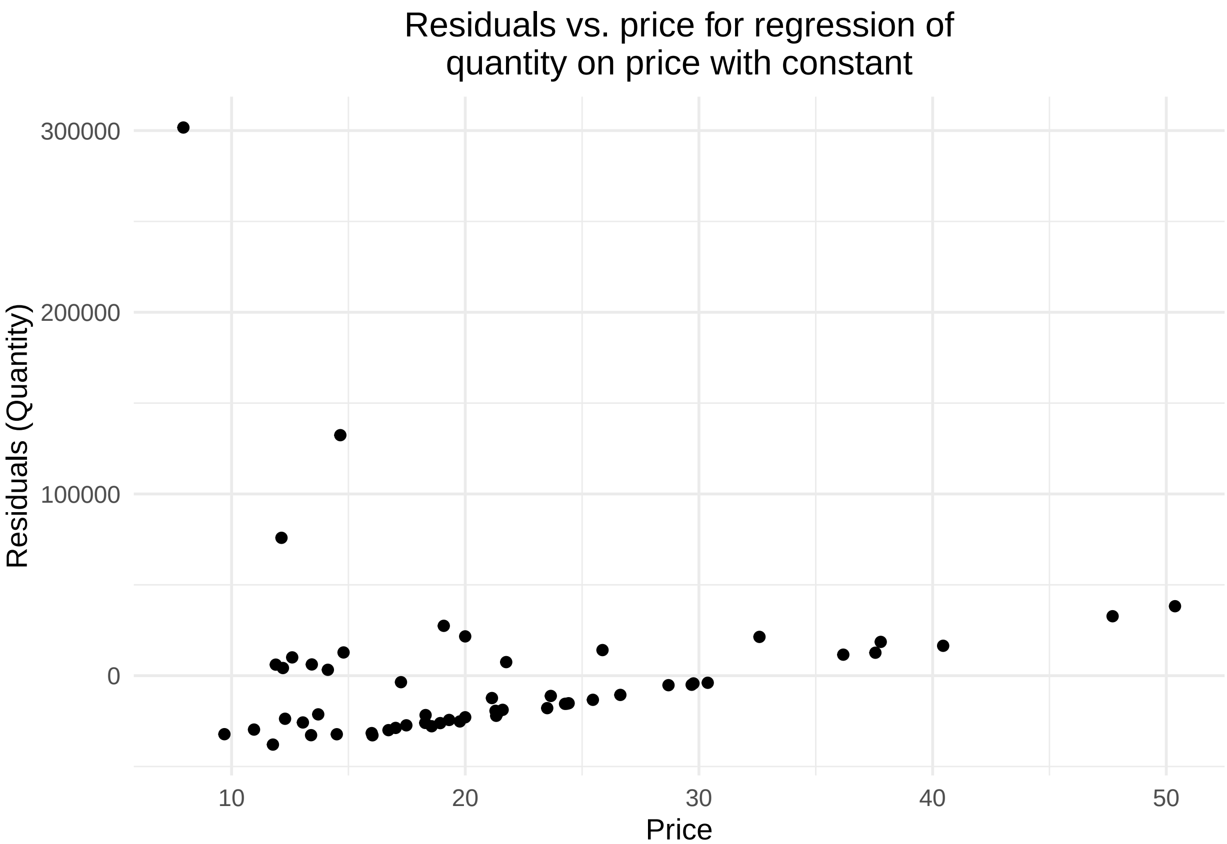
Table 3: Effect of Price on Quantity - Model Using lm Function

| | Dependent Variable: Quantity (New Car Registrations) |
|---|---|
| | qu<br>Question 10 |
| Price in Thousands of Euros | −2,026.143***<br>(733.795) |
| Constant | 65,877.820***<br>(16,987.680) |
| Observations | 57 |
| $R^2$ | 0.122 |
| Adjusted $R^2$ | 0.106 |
| Residual Std. Error | 50,348.000 (df = 55) |
| F Statistic | 7.624*** (df = 1; 55) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

```
  theme(plot.title = element_text(hjust = 0.5))
```


```
# plot residuals vs. price
ggplot() +
  # Price on x-axis, residuals on y-axis
  geom_point(aes(x = x, y = e10)) +
```

```r
  labs(x = "Price",
       y = "Residuals (Quantity)",
       title = str_wrap(
         "Residuals vs. price for regression of quantity on price with constant",
         40)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




*What do you see in terms of fit and whether constant variance assumption for residuals is valid? Has the fit improved or not relative to the question 8 analysis?*

Rsquared has increased compared to question 8.

The fitted vs. true values plot shows that the data points still do not cluster as closely around a line as would be expected in a well-fitted model. There are also a few negative fitted values, which could be indicative of overfitting.

In the residuals vs. prices plot, the spread of residuals is not constant and there is an upward trend, indicating potential heteroscedasticity. The spread of the residuals decreases as the price increases. In addition, the residuals appear to be centered around an upward-sloping line and not zero. This pattern suggests that the variance of the residuals is not constant.

The inclusion of a constant term does not seem to have notably improved the fit of the model. The points in the fitted vs. true values plot are still not aligning along a line indicating good prediction, and the residuals plot still exhibits a pattern suggesting heteroscedasticity. The presence of extreme values in the true values of quantity could be having a disproportionate effect on the model, which may explain the presence of negative fitted values and large residuals.

## Question 11

```
# Demean quantity
my_data$dmeanqu <- M0%*%my_data$qu

# Demean price and call it
my_data$dmeanprice <- M0%*%my_data$price

# Regress demeaned quantity on demeaned price variable and no constant
x11 <- my_data$dmeanprice
y11 <- my_data$dmeanqu
# find coefficient
b11 <- solve(t(x11)%*%x11)%*%t(x11)%*%y11
b11
```

```
      [,1]
[1,] -2026.143
```

```
# projection matrix, P
P <- x11%*%solve(t(x11)%*%x11)%*%t(x11)
# residual maker, M = I - P
M <- diag(57)-P
# sum of squared residuals, SSR = e'e
e11 <- M%*%y11
SSR <- t(e11)%*%e11
SSR
```

```
         [,1]
[1,] 139420676598
```

```
# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y--unnecessary??
M0y <- M0%*%y11
# total sum of squares
SST <- t(M0y)%*%M0y
SST
```

```
         [,1]
[1,] 158747287373
```

```
# calculate R squared
Rsquared11 <- 1-(SSR/SST)
Rsquared11
```

```
      [,1]
[1,] 0.1217445
```

```
# project estimates of y
y11_hat <- P%*%y11
y11_hat <- x11%*%b11

# compare R-squared
Rsquared10 == Rsquared11
```

```
  [,1]
```

[1,] FALSE

```
# check with lm model
Reg11 <- lm(dmeanqu~dmeanprice,my_data)
stargazer(Reg10, Reg11,
          column.labels = c("Y=Quantity", "Y=Demeaned Quantity"),
          dep.var.caption = "Dependent Variable: Price and Demeaned Price",
          covariate.labels = c("Price", "De-meaned Price"),
          header = FALSE,
          title = "Effect of Price on Quantity Ordinary Least Squares Regression")
```

Table 4: Effect of Price on Quantity Ordinary Least Squares Regression

| | Dependent Variable: Price and Demeaned Price | |
|---|---|---|
| | qu<br>Y=Quantity | dmeanqu<br>Y=Demeaned Quantity |
| | (1) | (2) |
| Price | $-2{,}026.143^{***}$<br>(733.795) | |
| De-meaned Price | | $-2{,}026.143^{***}$<br>(733.795) |
| Constant | $65{,}877.820^{***}$<br>(16,987.680) | 0.000<br>(6,668.756) |
| Observations | 57 | 57 |
| $R^2$ | 0.122 | 0.122 |
| Adjusted $R^2$ | 0.106 | 0.106 |
| Residual Std. Error (df = 55) | 50,348.000 | 50,348.000 |
| F Statistic (df = 1; 55) | $7.624^{***}$ | $7.624^{***}$ |
| *Note:* | | $^{*}$p<0.1; $^{**}$p<0.05; $^{***}$p<0.01 |

*Compare to analysis in question 10. Why do you get this? Explain the theorem behind this briefly.*

We get the same coefficient for qu in 10 and dmeanqu in 11. We also get the same Rsquared for 10 and 11. The coefficients are the same because the slopes in a regression that contains a constant term are obtained by demeaning the other explanatory variables and the dependent variable and then regressing the demeaned dependent on the demeaned explanatory variables. (See Corollary 3.2.2 in Greene.)

# Question 12

```
# Regress quantity on a constant, price, luxury indicator, weight, and fuel efficiency
# add constant
X12 <- cbind(1, my_data$price, my_data$luxury, my_data$weight, my_data$fuel)
y12 <- my_data$qu
# find coefficient
b12 <- solve(t(X12)%*%X12)%*%t(X12)%*%y12
b12
```

```
          [,1]
```

[1,] 118090.25375 [2,] -784.21912 [3,] 41858.87003 [4,] -90.11306 [5,] 268.24678

```r
# projection matrix, P
P <- X12%*%solve(t(X12)%*%X12)%*%t(X12)
# residual maker, M = I - P
M <- diag(57)-P
# calculate residuals
e12 <- M%*%y12
# sum of squared residuals, SSR=e'e
SSR <- t(e12)%*%e12
SSR
```

```
          [,1]
```

[1,] 130998987487

```r
# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y
M0y <- M0%*%y12
# total sum of squares
SST <- t(M0y)%*%M0y
SST
```

```
          [,1]
```

[1,] 158747287373

```r
# calculate R squared
Rsquared12 <- 1-(SSR/SST)
Rsquared12
```

```
       [,1]
```

[1,] 0.1747954

```r
# compare with lm
Reg12 <- lm(qu ~ price + luxury + weight + fuel, my_data)
stargazer(Reg12,
          column.labels = c("Question 12"),
          dep.var.caption = "Dependent Variable: Quantity (New Car Registrations)",
          covariate.labels =
            c("Price in Thousands of Euros",
              "Luxury Indicator",
              "Weight (kg)",
              "Fuel Efficiency (liter/km)"),
          header = FALSE,
          title = "Multivariate Regression - Model Using lm Function")
```

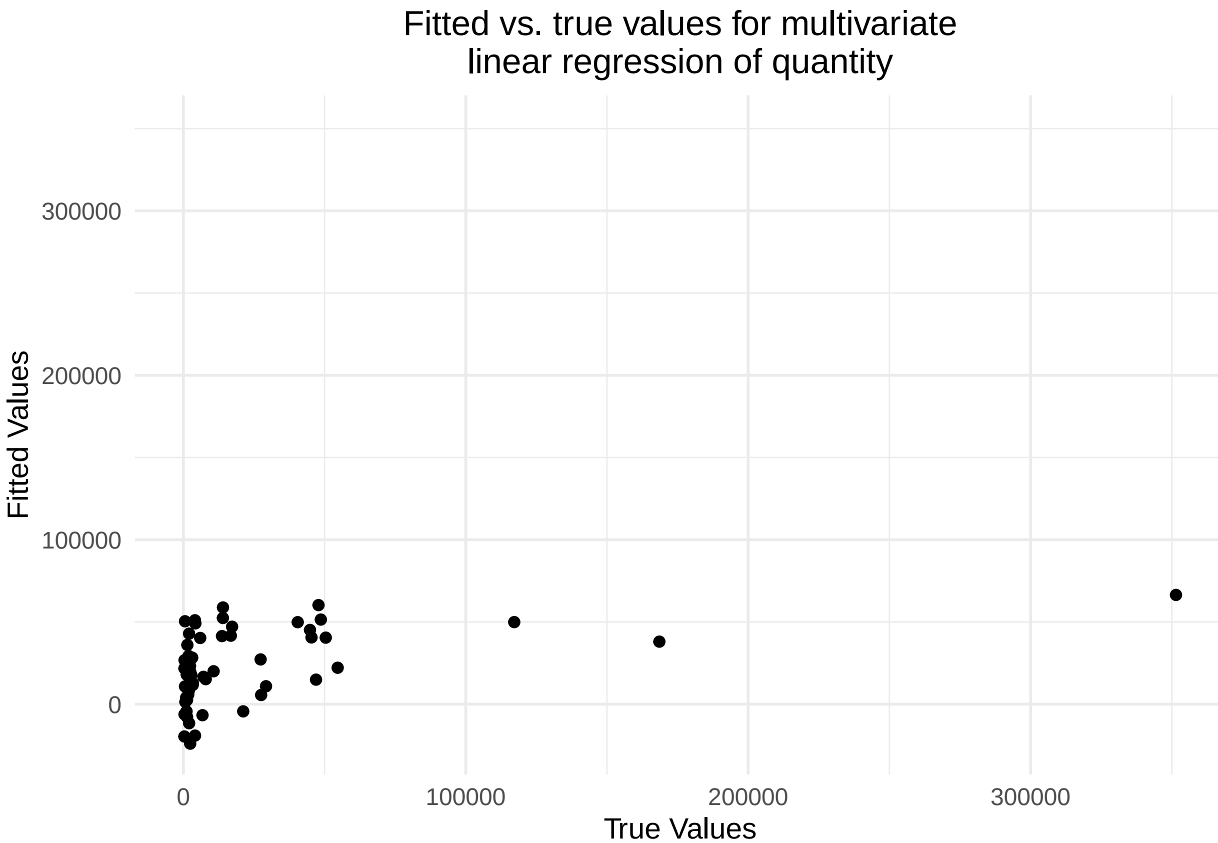
```r
# Generate series of predicted quantity values and plot against quantity
y12_hat <- P%*%y12

ggplot() +
  # True values on x-axis, fitted values on y-axis
  geom_point(aes(x = y12, y = y12_hat)) +
  labs(x = "True Values",
       y = "Fitted Values",
```

Table 5: Multivariate Regression - Model Using lm Function

| | Dependent Variable: Quantity (New Car Registrations) |
|---|---|
| | qu<br>Question 12 |
| Price in Thousands of Euros | −784.219 |
| | (1,902.925) |
| Luxury Indicator | 41,858.870 |
| | (38,937.710) |
| Weight (kg) | −90.113 |
| | (87.107) |
| Fuel Efficiency (liter/km) | 268.247 |
| | (6,402.192) |
| Constant | 118,090.300*** |
| | (37,751.850) |
| Observations | 57 |
| $R^2$ | 0.175 |
| Adjusted $R^2$ | 0.111 |
| Residual Std. Error | 50,191.750 (df = 52) |
| F Statistic | 2.754** (df = 4; 52) |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

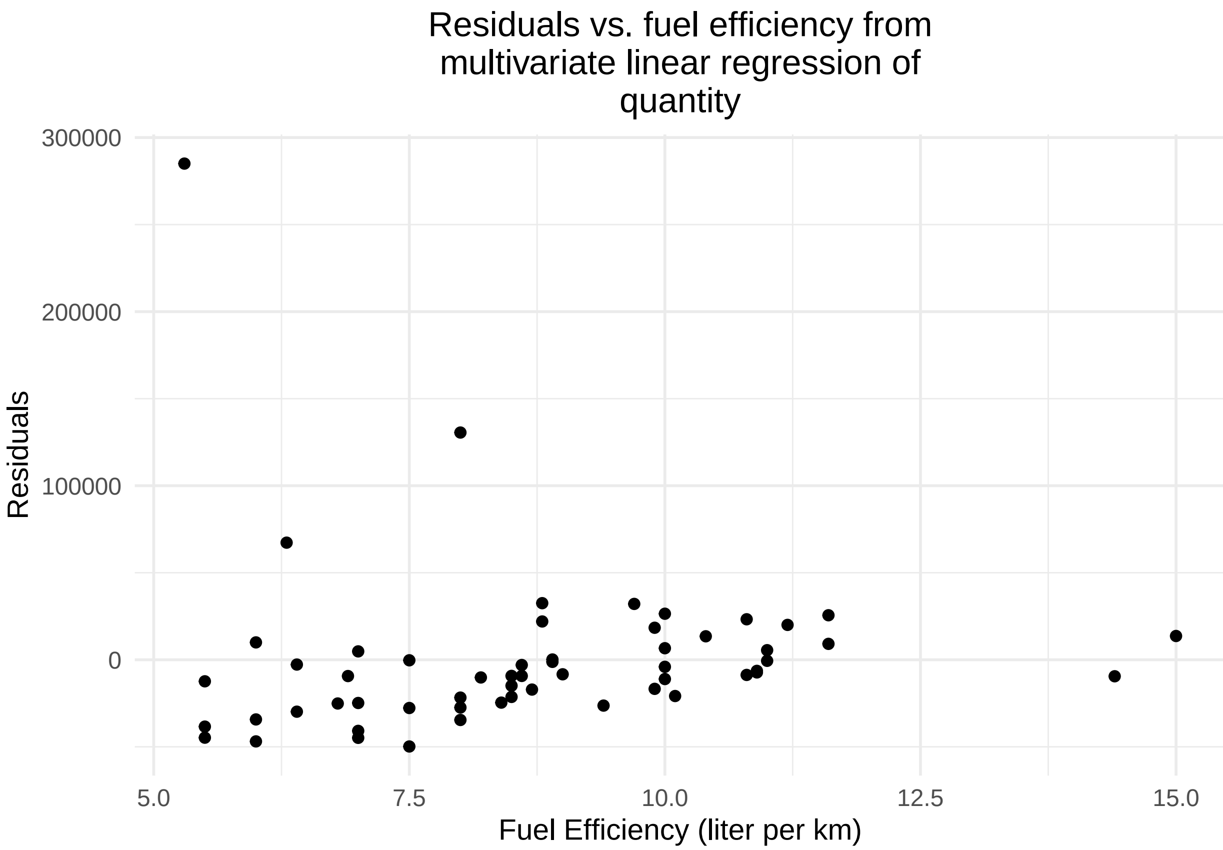
```r
       title = str_wrap(
         "Fitted vs. true values for multivariate linear regression of quantity",
         40)) +
  ylim(NA, max(my_data$qu)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




*What do you see in terms of fit?* The fit is better when there are more explanatory variables included in the model. The R-squared value with more variables is **0.1747954** which is higher than the R-squared value when qu is regressed on only price and a constant, **0.1217445**. However, we also know that R-squared is strictly increasing as we add more predictors; from the lm model output, we can see that the adjusted R-squared has also increased slightly compared to question 10 (0.106 vs 0.111).

From the fitted vs. true plot, we can see that there is a cluster of points around the lower true values, indicating that the model is capable of closely predicting lower quantities. For higher true values of quantity, there are greater discrepancies between the fitted and true values. The model seems to underpredict quantities as the true values rise, indicated by several points lying above the 45-degree line.

```r
# Plot residuals against fuel efficiency
ggplot() +
  # fuel efficiency on x-axis, residuals on y-axis
  geom_point(aes(x = my_data$fuel, y = e12)) +
  labs(x = "Fuel Efficiency (liter per km)",
       y = "Residuals",
       title = str_wrap("Residuals vs. fuel efficiency from multivariate linear regression of quantity"
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




*Is the constant variance assumption for the residuals valid or not?*

The residuals are mostly clustered around the zero line for the majority of the fuel efficiency values. Most of the residuals are evenly spread across the range of fuel efficiency, which suggests that the constant variance assumption (homoscedasticity) might hold. However, the presence of a few points with large residuals could be a cause for concern and warrant further investigation.

## Question 13

```
# Regress quantity on a constant, price, weight, and luxury indicator
X13 <- cbind(1, my_data$price, my_data$weight, my_data$luxury)
y13 <- my_data$qu
# projection matrix, P
P <- X13%*%solve(t(X13)%*%X13)%*%t(X13)
# residual maker, M = I - P
M <- diag(57)-P
# calculate residuals, save as qures
qures <- M%*%y13

# Regress fuel on a constant, price, weight, and luxury indicator
# X13, P and M are the same
y13 <- my_data$fuel
# calculate residuals, save as fuelres
fuelres <- M%*%y13

# Regress qures on fuelres (or Y13 on X13) and no constant
```

```r
x13 = fuelres
y13 = qures
# find coefficient
b13 <- solve(t(x13)%*%x13)%*%t(x13)%*%y13
b13
```

     [,1]

[1,] 268.2468

```r
# projection matrix, P
P <- x13%*%solve(t(x13)%*%x13)%*%t(x13)
# residual maker, M = I - P
M <- diag(57)-P
# calculate residuals
e13 <- M%*%y13
# sum of squared residuals, SSR=e'e
SSR <- t(e13)%*%e13
SSR
```

         [,1]

[1,] 130998987487

```r
# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y
M0y <- M0%*%y13
# total sum of squares
SST <- t(M0y)%*%M0y
SST
```

         [,1]

[1,] 131003410073

```r
# calculate R squared
Rsquared <- 1-(SSR/SST)
Rsquared
```

          [,1]

[1,] 0.00003375932

```r
# compare with lm
Reg13 <- lm(qu ~ price + weight + luxury, my_data)
Reg13_fuel <- lm(fuel ~ price + weight + luxury, my_data)
reg13_res <- lm(qures ~ fuelres - 1)

stargazer(Reg13, Reg13_fuel, reg13_res,
          column.labels = c("Y=Quantity",
                            "Y=Fuel Efficiency",
                            "Y=Quantity Residuals"),
          dep.var.caption = "",
          covariate.labels =
          c("Price in Thousands of Euros",
            "Weight (kg)",
            "Luxury Indicator",
```

```
          "Fuel Residuals"),
        header = FALSE,
        title = "Multivariate Regressions of Quantity, Fuel Efficiency, and Residuals - Model Using lm
```

Table 6: Multivariate Regressions of Quantity, Fuel Efficiency, and Residuals - Model Using lm Function

| | qu<br>Y=Quantity | fuel<br>Y=Fuel Efficiency | qures<br>Y=Quantity Residuals |
|---|---|---|---|
| | (1) | (2) | (3) |
| Price in Thousands of Euros | −778.786 | 0.020 | |
| | (1,880.537) | (0.041) | |
| Weight (kg) | −88.031 | 0.008*** | |
| | (70.869) | (0.002) | |
| Luxury Indicator | 41,741.010 | −0.439 | |
| | (38,468.490) | (0.833) | |
| Fuel Residuals | | | 268.247 |
| | | | (6,169.306) |
| Constant | 118,393.400*** | 1.130 | |
| | (36,701.280) | (0.795) | |
| Observations | 57 | 57 | 57 |
| $R^2$ | 0.175 | 0.750 | 0.00003 |
| Adjusted $R^2$ | 0.128 | 0.736 | −0.018 |
| Residual Std. Error | 49,716.820 (df = 53) | 1.077 (df = 53) | 48,365.980 (df = 56) |
| F Statistic | 3.741** (df = 3; 53) | 53.089*** (df = 3; 53) | 0.002 (df = 1; 56) |
| *Note:* | | | *p<0.1; **p<0.05; ***p<0.01 |

*Report your findings We wanted to get effect of fuel consumption on quantity, all else constant. To which coefficient of a previous question is the coefficient of fuelres equal to, and why?*

b13 is equal to the coefficient of fuel efficiency in the regression of qu on on a constant, price, luxury indicator, weight, and fuel efficiency.

This is a demonstration of the Frish-Waugh-Lovell Theorem. Let us partition the original $X$ into $X_1$ and $X_2$ where $X_1$ includes the constant, price, luxury, and weight and $X_2$ includes fuel and let $y$ equal quantity. If $X_1$ and $X_2$ are not orthogonal, then $b_2$ is equal to the coefficients obtained when the residuals of regressing $y$ on $x_1$ are regressed on the residuals of regressing $X_2$ on $X_1$.

## Question 14

```
# Repeat regression 12 but now use logqu and logprice and the other variables.
X14 <- cbind(1, my_data$logprice, my_data$luxury, my_data$weight, my_data$fuel)
y14 <- my_data$logqu
# find coefficient
b14 <- solve(t(X14)%*%X14)%*%t(X14)%*%y14
b14
```

```
##                [,1]
## [1,] 18.326734516
## [2,] -4.025965289
## [3,]  1.875205324
## [4,]  0.002041929
## [5,]  0.030354289
```

```r
# projection matrix, P
P <- X14%*%solve(t(X14)%*%X14)%*%t(X14)
# residual maker, M = I - P
M <- diag(57)-P
# calculate residuals
e14 <- M%*%y14
# sum of squared residuals, SSR=e'e
SSR <- t(e14)%*%e14
SSR
```

```
##          [,1]
## [1,] 103.3808
```

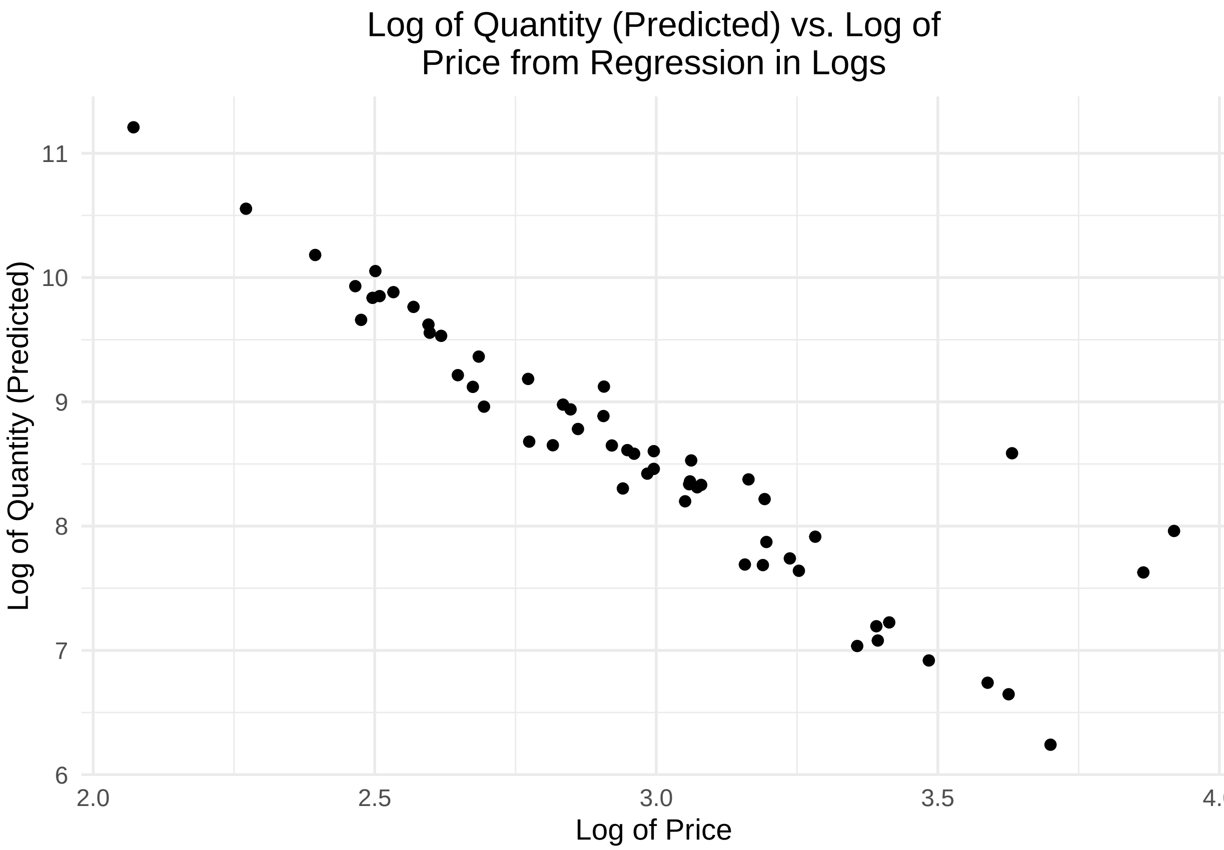
```r
# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y
M0y <- M0%*%y14
# total sum of squares
SST <- t(M0y)%*%M0y
SST
```

```
##          [,1]
## [1,] 163.5365
```

```r
# calculate R squared
Rsquared <- 1-(SSR/SST)
Rsquared
```

```
##           [,1]
## [1,] 0.3678426
```

```r
# Generate series of predicted logqu values and plot against logprice
y14_hat <- P%*%y14
ggplot() +
  # logprice on x-axis, fitted logqu on y-axis
  geom_point(aes(x = my_data$logprice, y = y14_hat)) +
  labs(x = "Log of Price",
       y = "Log of Quantity (Predicted)",
       title = str_wrap("Log of Quantity (Predicted) vs. Log of Price from Regression in Logs", 40)) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




*Call this the Regression in logs. Is the estimated car demand elastic with respect to price?* Yes, demand is elastic with respect to price because the absolute value of the coefficient is greater than 1. A 100% increase in price leads to a >400% decrease in demand.

## Question 15

```
# Set seed equal to 12345.
set.seed("12345")

# Generate two random variables, x and e, of dimension n = 100 such that x, e N(0, 1).
n = 100
x <- rnorm(n, mean=0, sd=1)
e <- rnorm(n, mean=0, sd=1)

# Generate a random variable y according to the data-generating process yi = xi + ei.
y = x + e

# Show that if you regress y on x and a constant,
# then you will get an estimate of the intercept beta0 and the coefficient on x, beta1.
X100 <- cbind(1, x)
# find coefficient
b100 <- solve(t(X100)%*%X100)%*%t(X100)%*%y
b100
```

```
##          [,1]
##   0.02205339
```

```
## x 1.09453503
```

```r
# Increase the sample to 1000, then 10000, and repeat the estimation.
# sample size = 1000
n = 1000
x <- rnorm(n, mean=0, sd=1)
e <- rnorm(n, mean=0, sd=1)
y = x + e
X1000 <- cbind(1, x)
b1000 <- solve(t(X1000)%*%X1000)%*%t(X1000)%*%y
b1000
```

```
##          [,1]
##   -0.03016513
## x  1.03640836
```

```r
# sample size = 10000
n = 10000
x <- rnorm(n, mean=0, sd=1)
e <- rnorm(n, mean=0, sd=1)
y = x + e
X10000 <- cbind(1, x)
b10000 <- solve(t(X10000)%*%X10000)%*%t(X10000)%*%y
b10000
```

```
##          [,1]
##   -0.00171373
## x  1.00645987
```

*What do you see as you increase the sample?* As the sample size increases, beta0 approaches 0 and beta1 approaches 1.