ARE 212 Problem Set 2

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

2024-01-31

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

# 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     trimmed       mad
## year           1 57 1.970000e+03      0.00 1.9700e+03 1.970000e+03      0.00
## country        2 57 4.000000e+00      0.00 4.0000e+00 4.000000e+00      0.00
## co             3 57 3.553200e+02    153.94 4.1300e+02 3.683000e+02    114.16
## type*          4 57 2.900000e+01     16.60 2.9000e+01 2.900000e+01     20.76
## segment        5 57 2.420000e+00      1.29 2.0000e+00 2.340000e+00      1.48
## domestic       6 57 2.600000e-01      0.44 0.0000e+00 2.100000e-01      0.00
## firm           7 57 1.305000e+01     10.25 1.2000e+01 1.232000e+01     11.86
## brand          8 57 1.626000e+01     13.99 1.1000e+01 1.483000e+01     13.34
## loc            9 57 4.420000e+00      1.99 4.0000e+00 4.060000e+00      1.48
## qu            10 57 2.273709e+04  53242.59 3.3870e+03 1.173570e+04   4158.69
## pr            11 57 1.394877e+06 600664.93 1.2650e+06 1.320745e+06 496671.00
## princ         12 57 1.110000e+00      0.48 1.0100e+00 1.050000e+00      0.40
## price         13 57 2.129000e+01      9.17 1.9310e+01 2.016000e+01      7.58
## horsepower    14 57 5.343000e+01     24.54 5.1500e+01 5.181000e+01     25.95
## fuel          15 57 8.700000e+00      2.10 8.6000e+00 8.610000e+00      2.22
## width         16 57 1.599600e+02     11.16 1.5900e+02 1.600600e+02      8.90
## height        17 57 1.422900e+02      5.26 1.4200e+02 1.423300e+02      4.45
## weight        18 57 9.232100e+02    218.48 9.2500e+02 9.160400e+02    229.80
## pop           19 57 5.366000e+07      0.00 5.3660e+07 5.366000e+07      0.00
## ngdp          20 57 6.717800e+13      0.00 6.7178e+13 6.717800e+13      0.00
## ngdpe         21 57 1.099200e+09      0.00 1.0992e+09 1.099200e+09      0.00
## country1      22 57 0.000000e+00      0.00 0.0000e+00 0.000000e+00      0.00
## country2      23 57 0.000000e+00      0.00 0.0000e+00 0.000000e+00      0.00
## country3      24 57 0.000000e+00      0.00 0.0000e+00 0.000000e+00      0.00
## country4      25 57 1.000000e+00      0.00 1.0000e+00 1.000000e+00      0.00
## country5      26 57 0.000000e+00      0.00 0.0000e+00 0.000000e+00      0.00
## yearsquared   27 57 3.880900e+06      0.00 3.8809e+06 3.880900e+06      0.00
## luxury        28 57 5.000000e-02      0.23 0.0000e+00 0.000000e+00      0.00
## logprice      29 57 2.980000e+00      0.40 2.9600e+00 2.960000e+00      0.41
## logqu         30 57 8.590000e+00      1.71 8.1300e+00 8.530000e+00      1.80
## carspc        31 57 0.000000e+00      0.00 0.0000e+00 0.000000e+00      0.00
##                    min         max      range  skew kurtosis       se
## year        1.9700e+03 1.97000e+03       0.00   NaN      NaN     0.00
## country     4.0000e+00 4.00000e+00       0.00   NaN      NaN     0.00
## co          1.5000e+01 5.44000e+02     529.00 -0.78    -0.82    20.39
## type*       1.0000e+00 5.70000e+01      56.00  0.00    -1.26     2.20
## segment     1.0000e+00 5.00000e+00       4.00  0.36    -1.21     0.17
## domestic    0.0000e+00 1.00000e+00       1.00  1.05    -0.92     0.06
## firm        1.0000e+00 3.30000e+01      32.00  0.48    -1.24     1.36
## brand       1.0000e+00 4.60000e+01      45.00  0.78    -0.67     1.85
## loc         1.0000e+00 1.20000e+01      11.00  2.49     6.87     0.26
## qu          3.6800e+02 3.51477e+05  351109.00  4.57    23.68  7052.15
## pr          5.2000e+05 3.30000e+06 2780000.00  1.20     1.22 79560.01
## princ       4.2000e-01 2.64000e+00       2.22  1.20     1.22     0.06
## price       7.9400e+00 5.03700e+01      42.44  1.20     1.22     1.21
## horsepower  1.3000e+01 1.18000e+02     105.00  0.54    -0.32     3.25
## fuel        5.3000e+00 1.50000e+01       9.70  0.59     0.39     0.28
## width       1.3200e+02 1.80500e+02      48.50  0.04    -0.52     1.48
```

```
## height      1.2700e+02 1.55000e+02      28.00 -0.12      0.38      0.70
## weight      5.2000e+02 1.51000e+03     990.00  0.27     -0.50     28.94
## pop         5.3660e+07 5.36600e+07       0.00   NaN       NaN      0.00
## ngdp        6.7178e+13 6.71780e+13       0.00   NaN       NaN      0.00
## ngdpe       1.0992e+09 1.09920e+09       0.00   NaN       NaN      0.00
## country1    0.0000e+00 0.00000e+00       0.00   NaN       NaN      0.00
## country2    0.0000e+00 0.00000e+00       0.00   NaN       NaN      0.00
## country3    0.0000e+00 0.00000e+00       0.00   NaN       NaN      0.00
## country4    1.0000e+00 1.00000e+00       0.00   NaN       NaN      0.00
## country5    0.0000e+00 0.00000e+00       0.00   NaN       NaN      0.00
## yearsquared 3.8809e+06 3.88090e+06       0.00   NaN       NaN      0.00
## luxury      0.0000e+00 1.00000e+00       1.00  3.90     13.46      0.03
## logprice    2.0700e+00 3.92000e+00       1.85  0.24     -0.38      0.05
## logqu       5.9100e+00 1.27700e+01       6.86  0.37     -0.82      0.23
## carspc      0.0000e+00 1.00000e-02       0.01  4.57     23.68      0.00
```

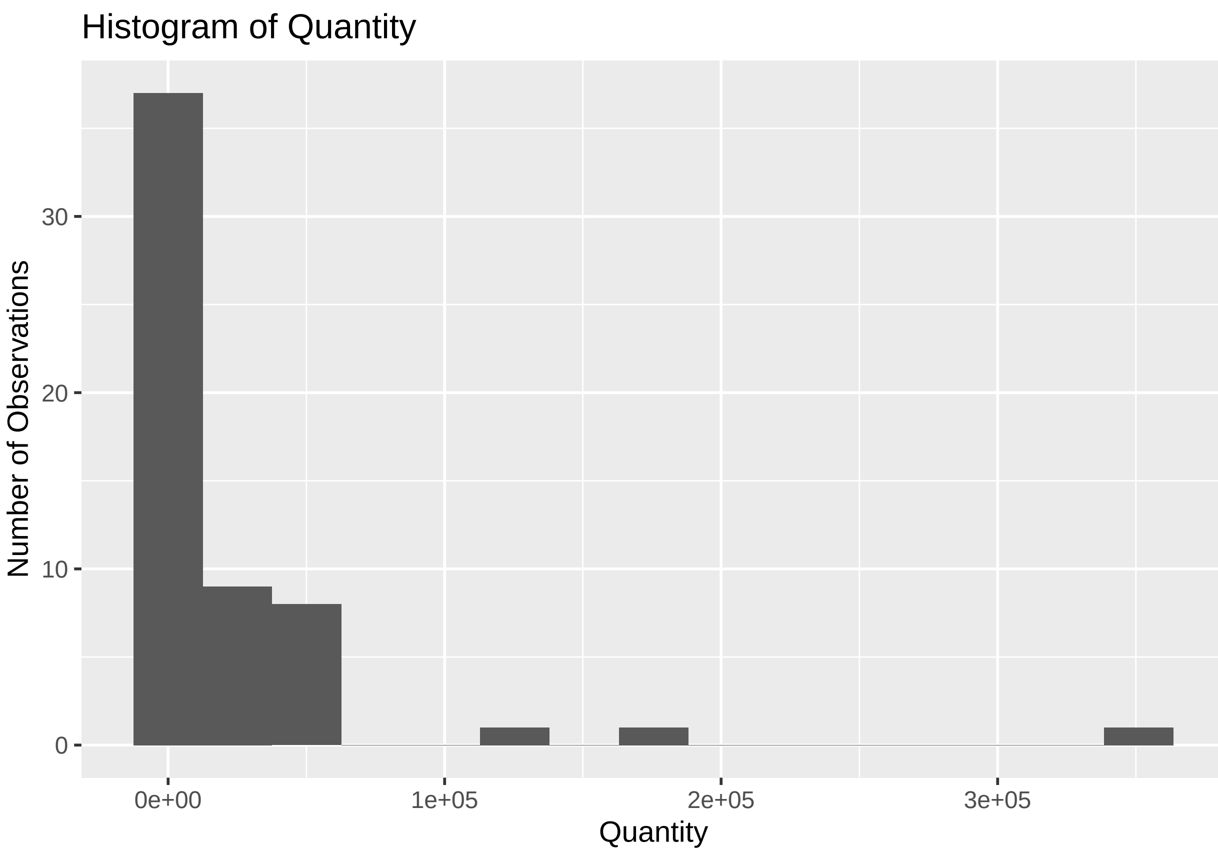
```r
# Create summary table
summary_maker <-
  list("Price" =
         list("min" = ~ min(my_data$price),
              "max" = ~ max(my_data$price),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$price)),
       "Log of Price" =
         list("min" = ~ min(my_data$logprice),
              "max"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

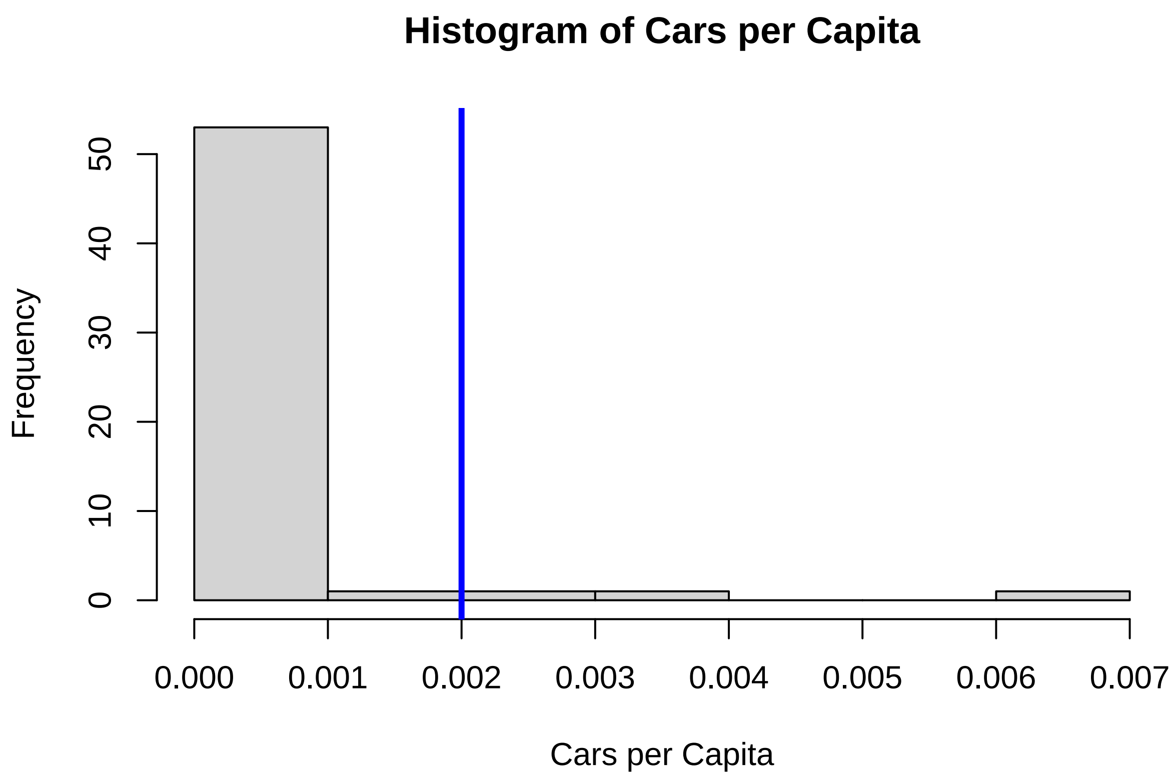
Question 3

```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("Histogram of Quantity"))
```



Question 4

```r
# Make a histogram of carspc
histcarspcvertical <- hist(my_data$carspc, main="Histogram of Cars per Capita", xlab="Cars per Capita")
abline(v=0.002, col="blue", lwd=3)
```




```r
histcarspcvertical
```

```
## $breaks
## [1] 0.000 0.001 0.002 0.003 0.004 0.005 0.006 0.007
##
## $counts
## [1] 53  1  1  1  0  0  1
##
## $density
## [1] 929.82456  17.54386  17.54386  17.54386   0.00000   0.00000  17.54386
##
## $mids
## [1] 0.0005 0.0015 0.0025 0.0035 0.0045 0.0055 0.0065
##
## $xname
## [1] "my_data$carspc"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

Question 5

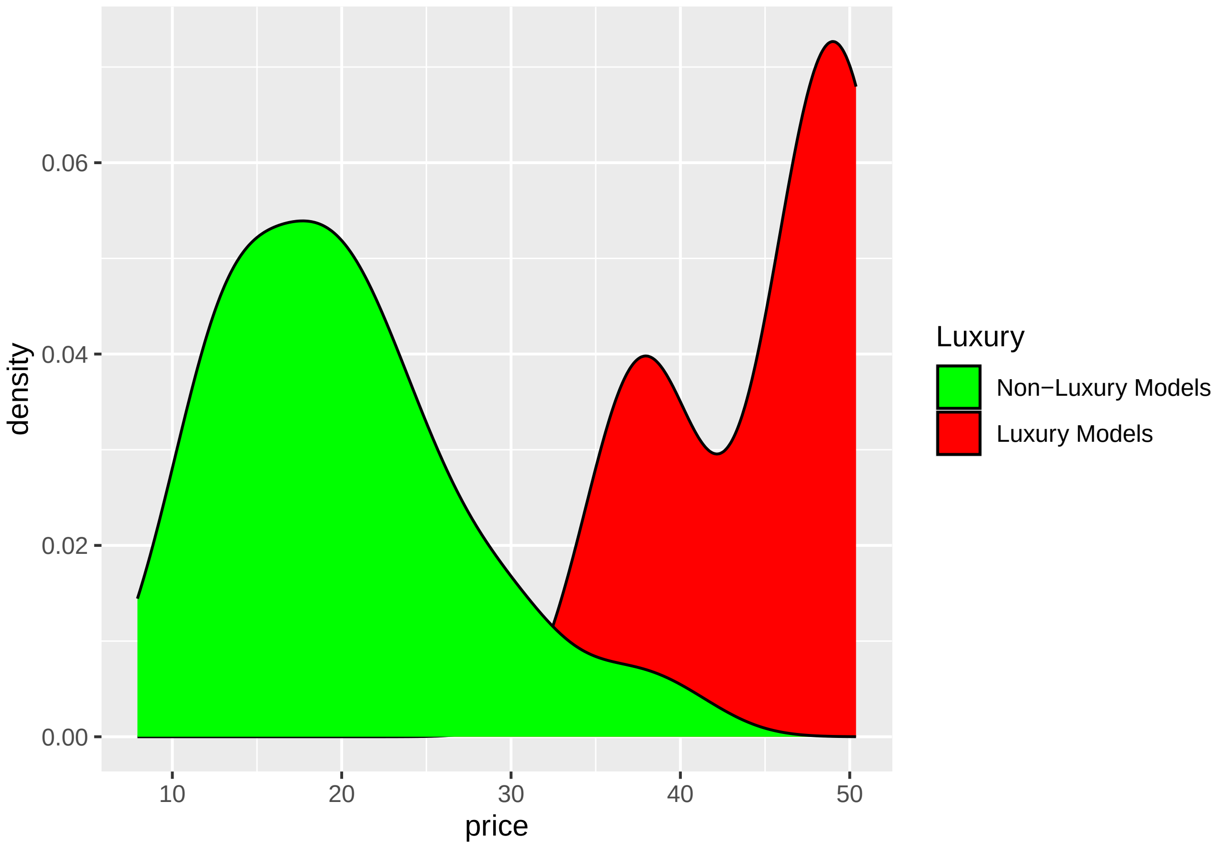
```r
# Make scatter plots of price vs. qu and logprice vs. logqu
scatter <- ggplot(my_data, aes(x=price, y=qu)) + geom_point()
(scatter <- scatter + xlab("Price") + ylab("Quantity") + ggtitle("Scatter Plot of Quantity vs. Price"))
```


```
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 o
```

Question 6

```
# 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")) +
  geom_density(data=datanoluxury, aes(x=price, fill="g")) +
  scale_fill_manual(name="Luxury", values=c("r"="red", "g"="green"),
                    labels=c("r"="Luxury Models", "g"="Non-Luxury Models"))
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
b1
```

```
   [,1]
```

[1,] 591.06

```
# 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
#e <- y1-x%*%b1
SSR_1 <- t(e_1)%*%e_1
```

```r
SSR_1
```

```
         [,1]
[1,] 177542591800
```

```r
# construct demeaner
#i <- c(rep(1,57))
#M0 <- diag(57)-i%*%t(i)*(1/57)
#M0 <- diag(57)-i%*%solve(t(i)%*%i)%*%t(i)
# # demeaned y
# M0y <- M0%*%y1
# # total sum of squares
# SST_1 <- t(M0y)%*%M0y
# SST_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.
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
b2
```

```
         [,1]
[1,] 1.101491e-05
```

```r
# 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
SSR_2
```

```
         [,1]
[1,] 6.165967e-05
```

```r
# construct demeaner
# i <- c(rep(1,57))
#M0 <- diag(length(y))-i%*%t(i)*(1/length(y))
# M0 <- diag(57)-i%*%solve(t(i)%*%i)%*%t(i)
# # demeaned y
# M0y <- M0%*%y2
# total sum of squares
# SST_2 <- t(M0y)%*%M0y
```

```r
# SST_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")
```

Table 1: Effect of Price on Quantity - No Constant

	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

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

Table 2: Effect of Price on Cars per Capity - No Constant

	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 1.1014909×10^{-5} and the R squared is **0.0567026**.

The coefficients 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.

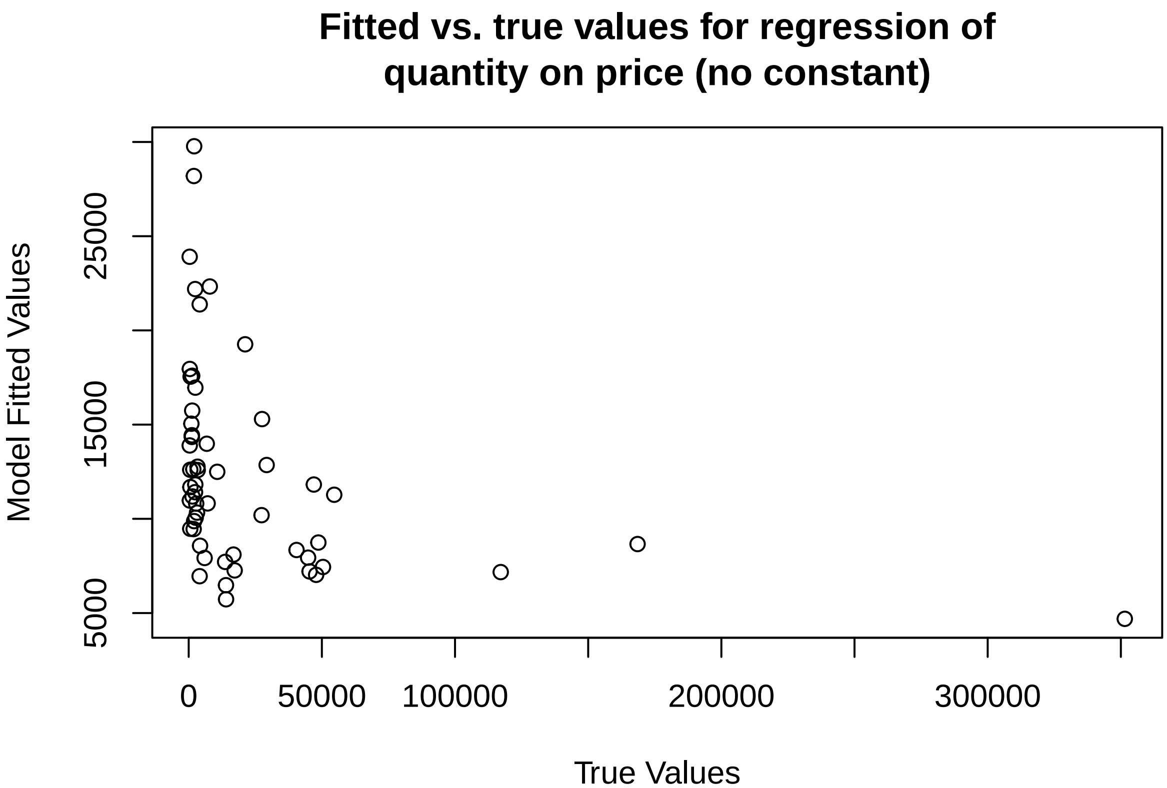
Question 9

- sample size, n = 57
- number of explanatory variables, k = 1
- degrees of freedom, n - k = 56
- estimate of coefficient, b = 591.1

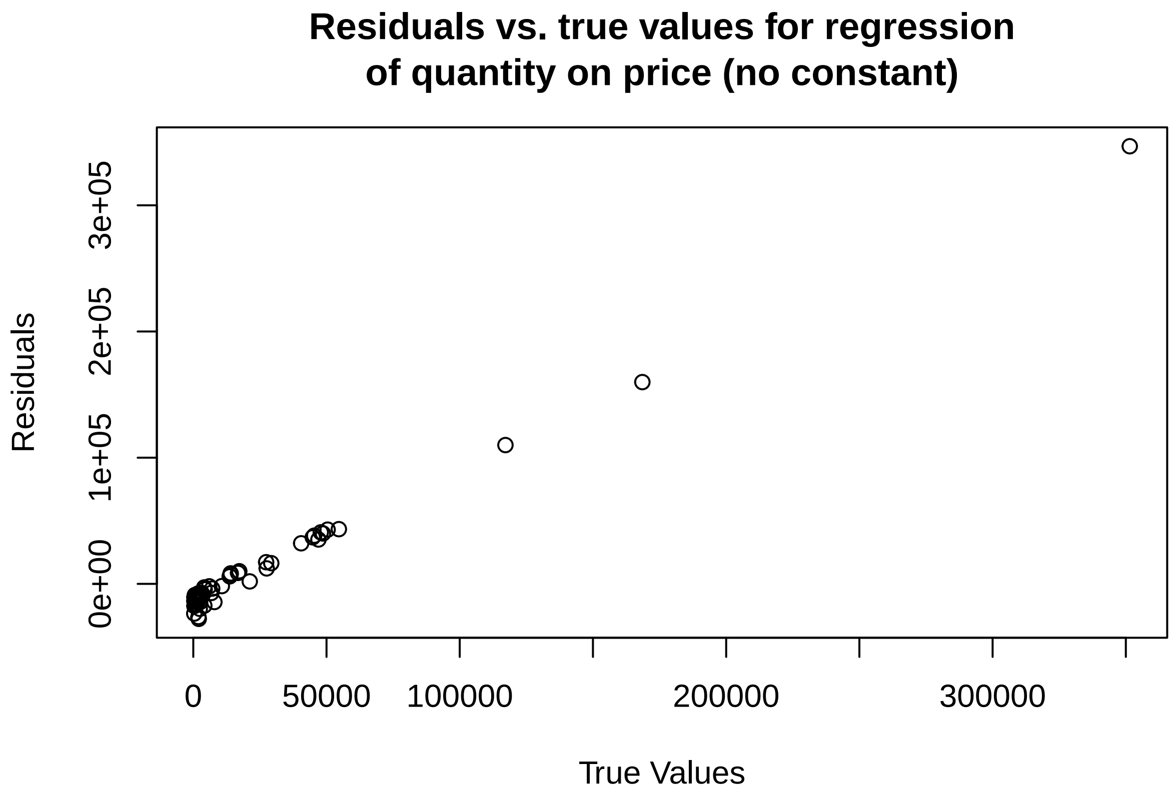
```
# regression of quantity on price
# get degrees of freedom, coefficient, and sample size

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

# plot fitted (predicted) vs. true (observed) quantities
plot(x = y1, # True values on x-axis
     y = y1_hat, # fitted values on y-axis
     xlab = "True Values",
     ylab = "Model Fitted Values",
     main = str_wrap("Fitted vs. true values for regression of quantity on price (no constant)", 40))
```




```r
# plot residuals vs. true (observed) quantities
plot(x = y1, # True values on x-axis
     y = e, # residuals on y-axis
     xlab = "True Values",
     ylab = "Residuals",
     main = str_wrap("Residuals vs. true values for regression of quantity on price (no constant)", 40)
```




```
# TODO explain why constant variance assumption is or isn't valid ####
```

What do you see in terms of fit and whether the constant variance assumption for the residuals is valid or not? There appears to be a positive linear relationship between the quantity and the residual. Constant variance assumption is not valid because .

Question 10

```
# Regress quantity on price and a constant
# add constant
X10 <- cbind(1, x)
y10 <- my_data$qu
# find coefficient
b10 <- solve(t(X10)%*%X10)%*%t(X10)%*%y10
b10
```

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

```
# 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
#e10 <- y10 - X10%*%b10
SSR <- t(e10)%*%e10
SSR
```

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

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

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

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

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

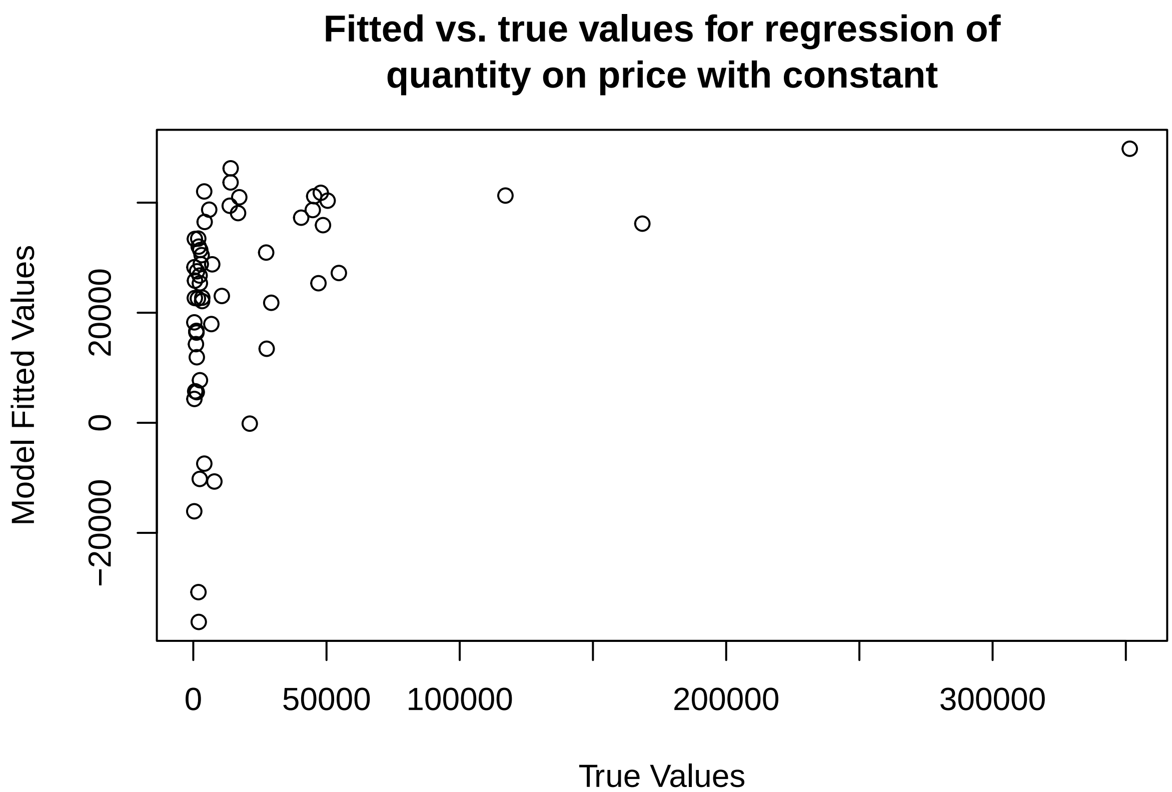
```r
# project estimates of y
y10_hat <- P%*%y10
y10_hat <- X10%*%b10
```

```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")
```

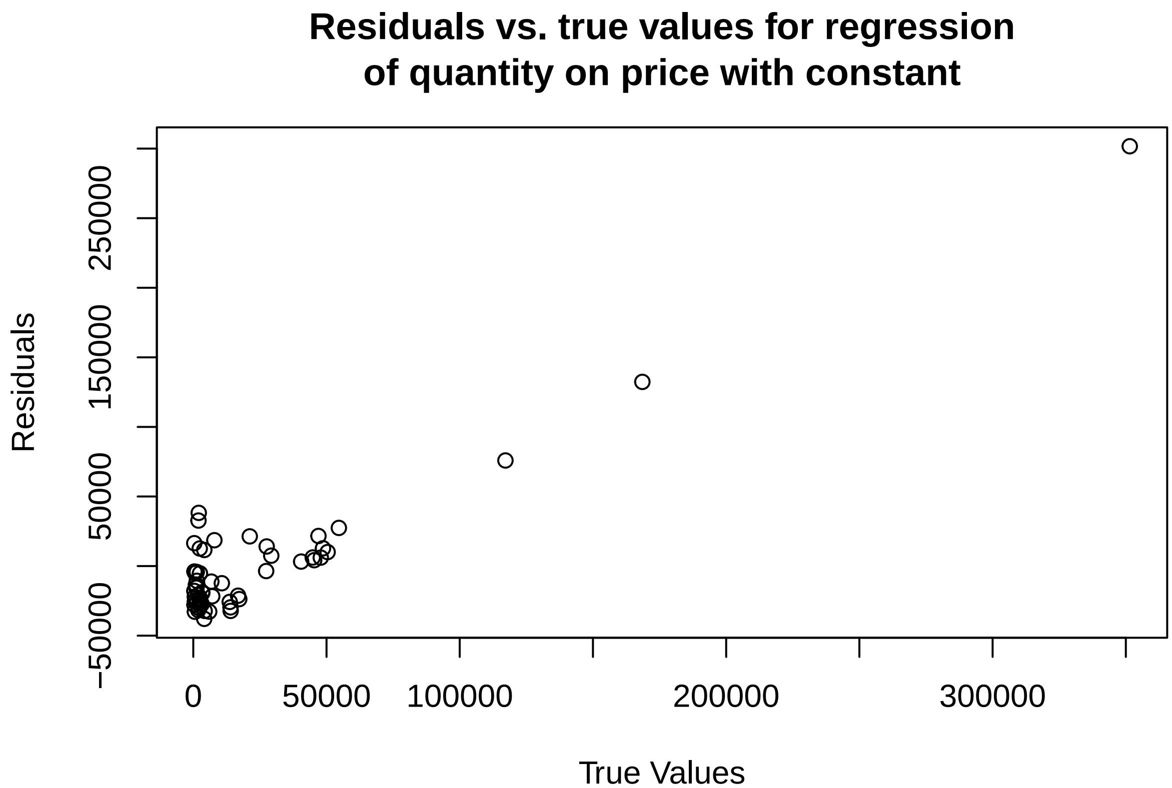
Table 3: Effect of Price on Quantity

	Dependent Variable: Quantity (New Car Registrations)
	qu
	Question 10
Price in Thousands of Euros	$-2{,}026.143^{***}$
	(733.795)
Constant	$65{,}877.820^{***}$
	(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

```r
# plot fitted (predicted) vs. true (observed) quantities
plot(x = y10, # True values on x-axis
     y = y10_hat, # fitted values on y-axis
     xlab = "True Values",
     ylab = "Model Fitted Values",
     main =
       str_wrap("Fitted vs. true values for regression of quantity on price with constant", 40))
```




```r
# plot residuals vs. true (observed) quantities
plot(x = y10, # True values on x-axis
     y = e10, # residuals on y-axis
     xlab = "True Values",
     ylab = "Residuals",
     main = str_wrap("Residuals vs. true values for regression of quantity on price with constant", 40))
```




```
# TODO constant variance assumption valid??? ####
```

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 improved; was negative, now is between 0 and 1. Constant variance assumption is valid because . However, residuals still have a positive linear relationship with quantity.

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))
MO <- diag(57)-i%*%t(i)*(1/57)
# demeaned y--unnecessary??
MOy <- MO%*%y11
# total sum of squares
SST <- t(MOy)%*%MOy
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")
```

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.)

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

	Dependent Variable: Price and Demeaned Price	
	qu Y=Quantity	dmeanqu Y=Demeaned Quantity
	(1)	(2)
Price	$-2{,}026.143^{***}$	
	(733.795)	
De-meaned Price		$-2{,}026.143^{***}$
		(733.795)
Constant	$65{,}877.820^{***}$	0.000
	(16,987.680)	(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:		$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Question 12

```r
# 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,] 1.30999e+11
```

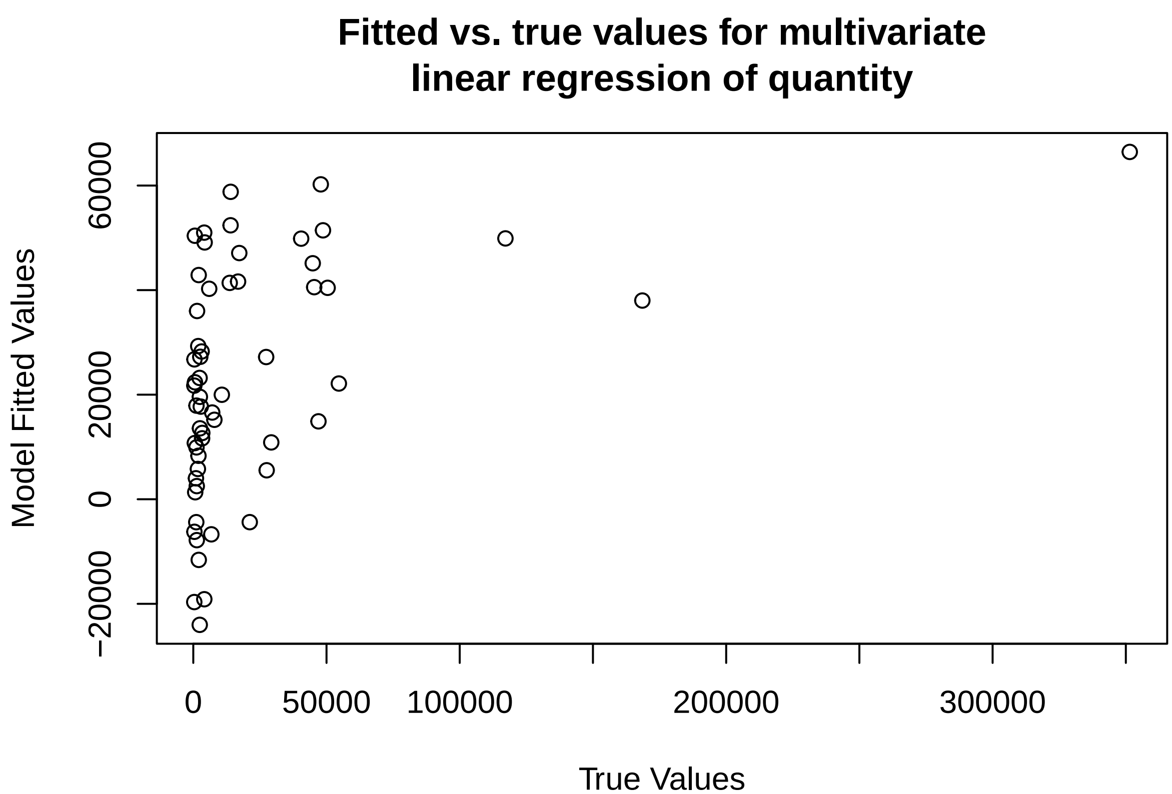
```r
# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y
M0y <- M0%*%y12
# total sum of squares
SST <- t(M0y)%*%M0y
SST
```

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

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

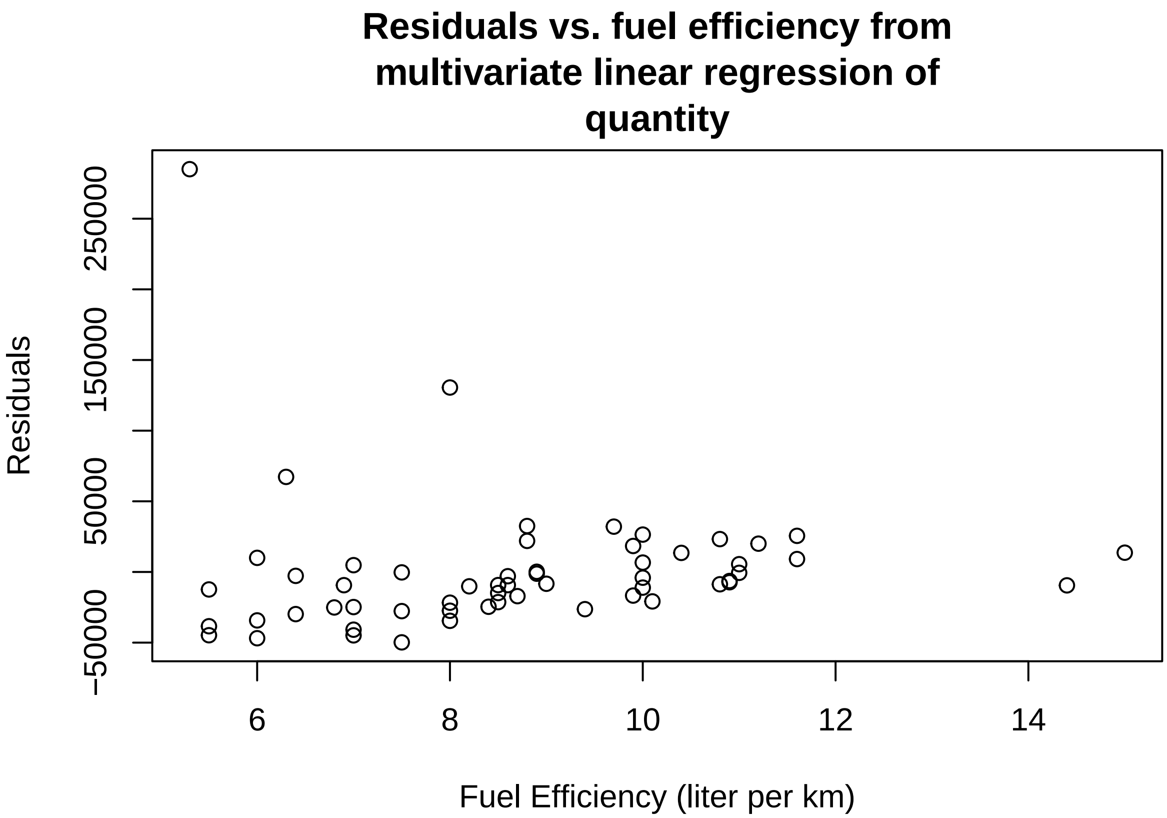
```
##           [,1]
## [1,] 0.1747954
```

```r
# Generate series of predicted quantity values and plot against quantity
y12_hat <- P%*%y12
plot(x = y12, # True values on x-axis
     y = y12_hat, # fitted values on y-axis
     xlab = "True Values",
     ylab = "Model Fitted Values",
     main = str_wrap("Fitted vs. true values for multivariate linear regression of quantity", 40))
```



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**.

```r
# Plot residuals against fuel efficiency
plot(x = my_data$fuel, # fuel efficiency on x-axis
     y = e12, # residuals on y-axis
     xlab = "Fuel Efficiency (liter per km)",
     ylab = "Residuals",
     main = str_wrap("Residuals vs. fuel efficiency from multivariate linear regression of quantity", 40
```




```r
# TODO is the constant variance assumption for the residuals valid or not? ####
```

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

Question 13

```r
# 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
```

```r
fuelres <- M%*%y13

# Regress qures on fuelres (or Y13 on X13) and no constant
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,] 1.30999e+11
```

```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,] 3.375932e-05
```

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

```r
# 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)
```

```r
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
```

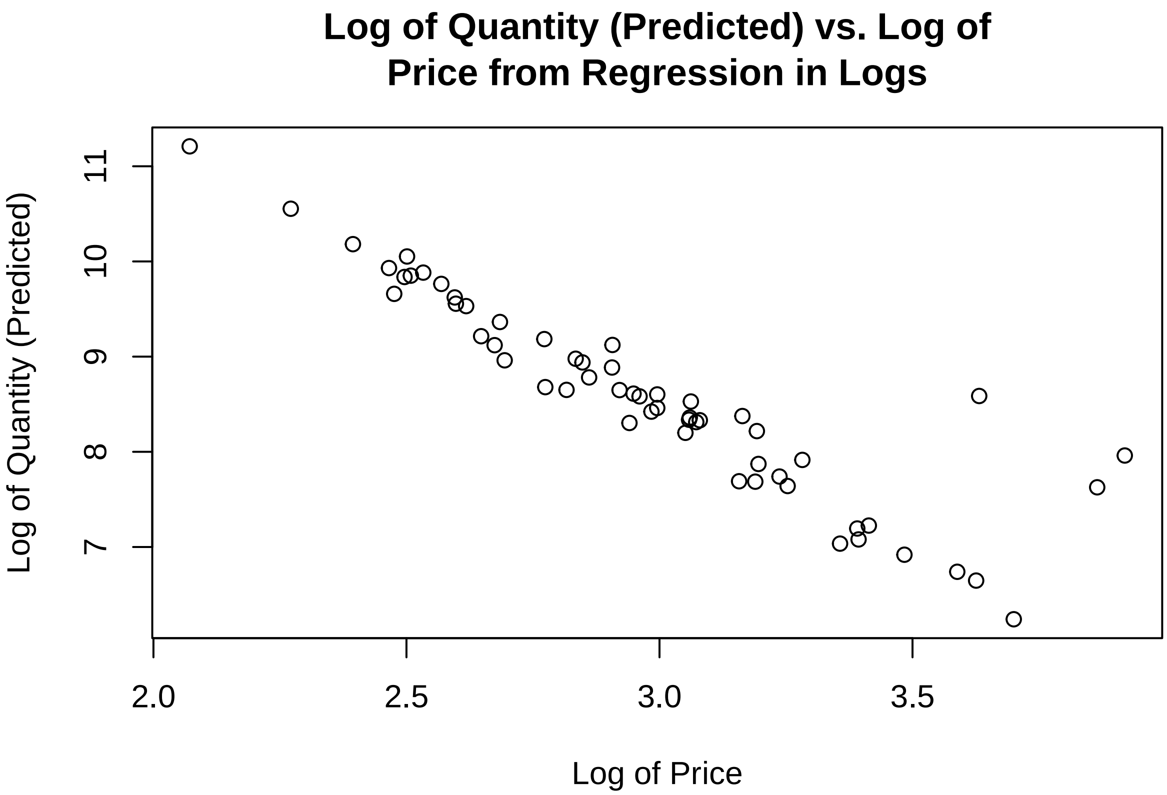
```r
# construct demeaner
i <- c(rep(1,57))
M0 <- diag(57)-i%*%t(i)*(1/57)
# demeaned y
M0y <- M0%*%y14
# total sum of squares
SST <- t(M0y)%*%M0y
SST
```

```
##          [,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
plot(x = my_data$logprice, # logprice
     y = y14_hat, # fitted logqu values on y-axis
     xlab = "Log of Price",
     ylab = "Log of Quantity (Predicted)",
     main = str_wrap("Log of Quantity (Predicted) vs. Log of Price from Regression in Logs", 40))
```



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.