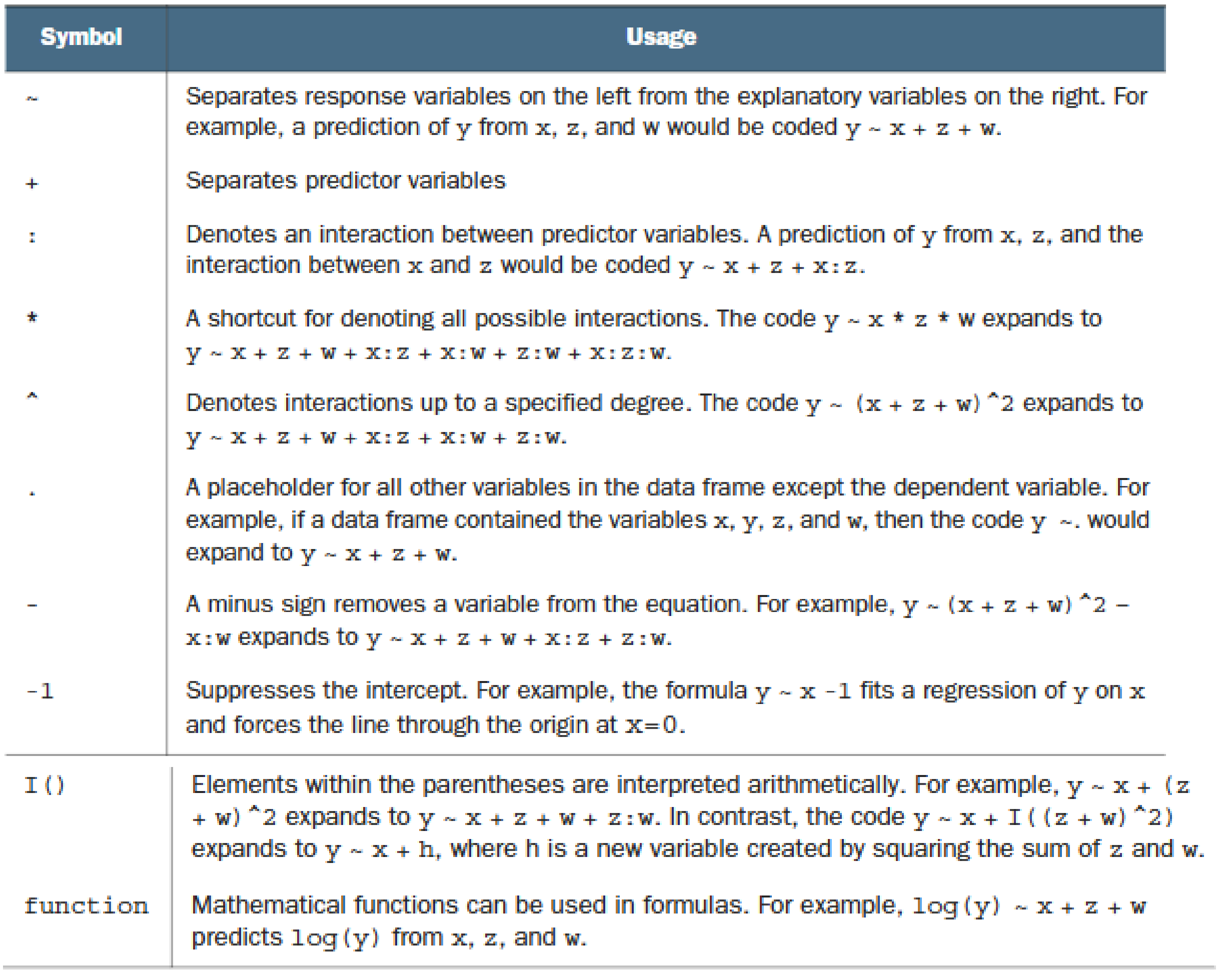
Day 3 : Statistical analysis using R

Session 9: Regression

9.1. Fitting regression models with lm()

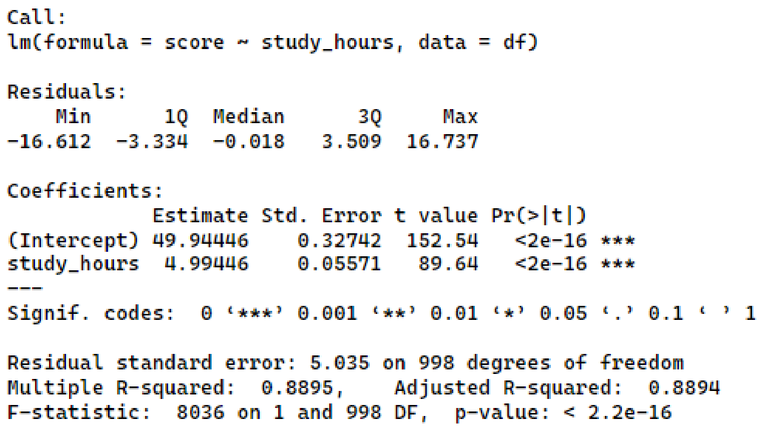
Symbols commonly used in R formulas



Symbol	Usage
~	Separates response variables on the left from the explanatory variables on the right. For example, a prediction of y from x, z, and w would be coded y ~ x + z + w.
+	Separates predictor variables
:	Denotes an interaction between predictor variables. A prediction of y from x, z, and the interaction between x and z would be coded y ~ x + z + x:z.
*	A shortcut for denoting all possible interactions. The code y ~ x * z * w expands to y ~ x + z + w + x:z + x:w + z:w + x:z:w.
^	Denotes interactions up to a specified degree. The code y ~ (x + z + w)^2 expands to y ~ x + z + w + x:z + x:w + z:w.
.	A placeholder for all other variables in the data frame except the dependent variable. For example, if a data frame contained the variables x, y, z, and w, then the code y ~. would expand to y ~ x + z + w.
-	A minus sign removes a variable from the equation. For example, y ~ (x + z + w)^2 – x:w expands to y ~ x + z + w + x:z + z:w.
-1	Suppresses the intercept. For example, the formula y ~ x -1 fits a regression of y on x and forces the line through the origin at x=0.
I()	Elements within the parentheses are interpreted arithmetically. For example, y ~ x + (z + w)^2 expands to y ~ x + z + w + z:w. In contrast, the code y ~ x + I((z + w)^2) expands to y ~ x + h, where h is a new variable created by squaring the sum of z and w.
function	Mathematical functions can be used in formulas. For example, log(y) ~ x + z + w predicts log(y) from x, z, and w.

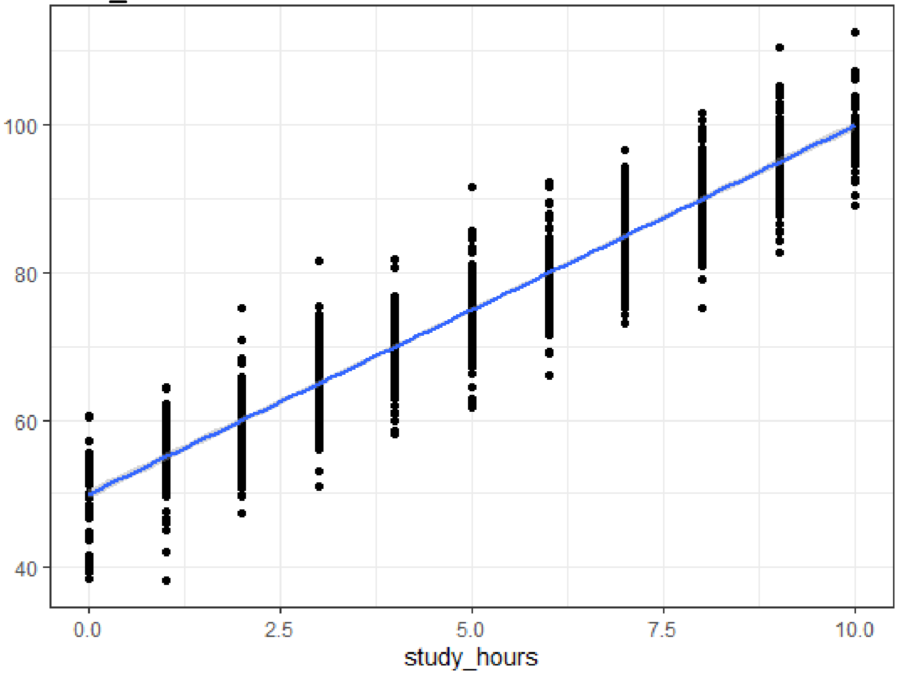
E034-simple_regression.R

```
# Set seed for reproducibility
set.seed(12345)

# Generate 1000 observations
n <- 1000

# Generate study_hours as uniform random numbers between 0 and 10
study_hours <- round(runif(n, min = 0, max = 10))

# Generate score as a linear function of study_hours with noise
score <- 50 + 5 * study_hours + rnorm(n, mean = 0, sd = 5)

# Combine into a data frame
df <- data.frame(study_hours, score)

# Perform linear regression
model <- lm(score ~ study_hours, data = df)

# Summarize the regression results
summary(model)
```



```
Call:
lm(formula = score ~ study_hours, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.612  -3.334  -0.018   3.509  16.737 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 49.94446    0.32742  152.54   <2e-16 ***
study_hours  4.99446    0.05571   89.64   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.035 on 998 degrees of freedom
Multiple R-squared:  0.8895,	Adjusted R-squared:  0.8894 
F-statistic:  8036 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
# visualizing simple regression
library(ggplot2)
ggplot(data = df, aes(x=study_hours, y=score)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  theme_bw()
```

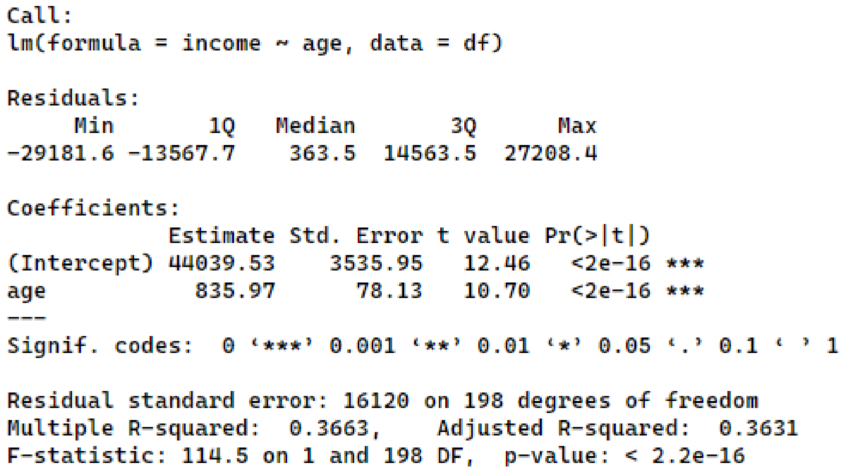


9.2. Multiple regression

E035-multiple_regression.R

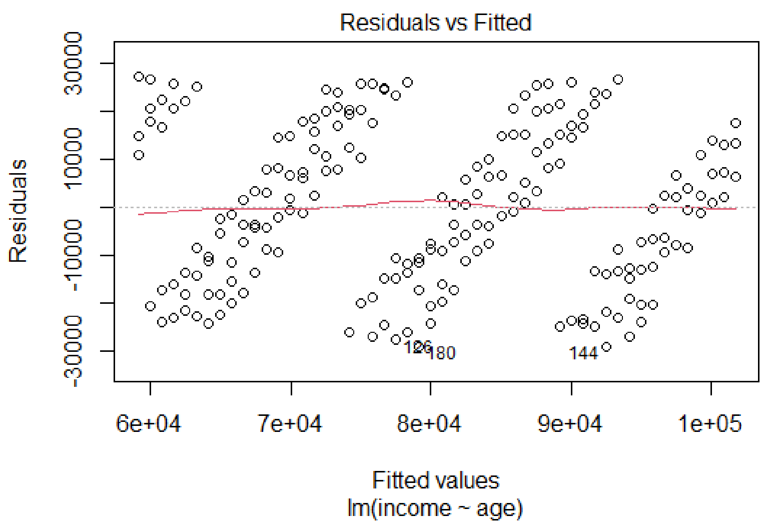
```
# Set seed for reproducibility
set.seed(12345)

# Generate 200 observations
n <- 200

# Generate age variable (cycles from 18 to 69)
age <- (1:n %% 52) + 18

# Generate educ_year variable (cycles from 0 to 17)
educ_year <- (1:n %% 18)

# Generate income variable with a linear relationship to age and educ_year,
plus noise
income <- 20000 + 800 * age + 3000 * educ_year + rnorm(n, mean = 0, sd =
2000)
```

```
# Combine into a data frame
df <- data.frame(age, educ_year, income)

# Regression with omitted variable
model_omitted <- lm(income ~ age, data = df)
summary(model_omitted)
```

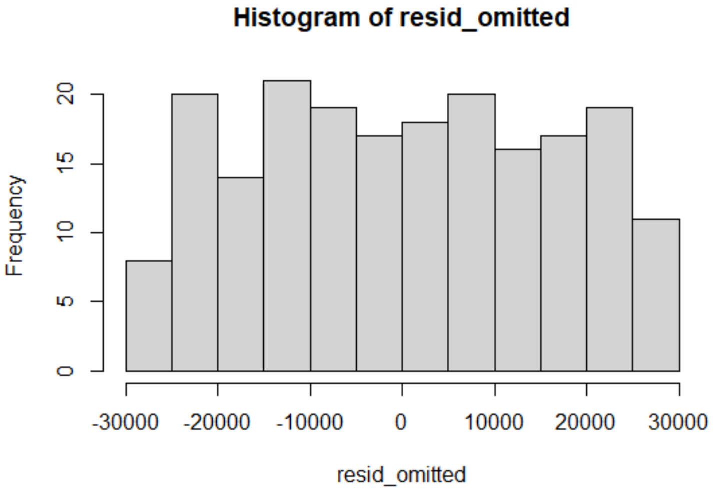


```
Call:
lm(formula = income ~ age, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-29181.6 -13567.7    363.5  14563.5  27208.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 44039.53    3535.95   12.46   <2e-16 ***
age           835.97      78.13   10.70   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16120 on 198 degrees of freedom
Multiple R-squared:  0.3663,	Adjusted R-squared:  0.3631 
F-statistic: 114.5 on 1 and 198 DF,  p-value: < 2.2e-16
```

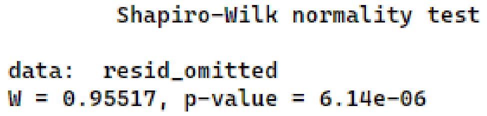
```
# Residual diagnostics for omitted variable model
plot(model_omitted, which = 1)  # Residual vs Fitted plot
```



```
resid_omitted <- residuals(model_omitted)
hist(resid_omitted)
```

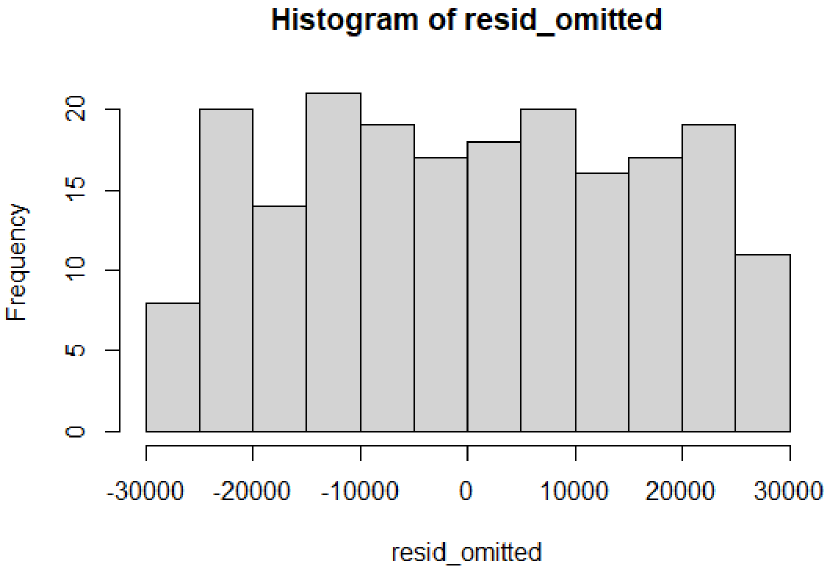


```
shapiro.test(resid_omitted)  # Shapiro-Wilk test for normality [H0: normally distributed]
```

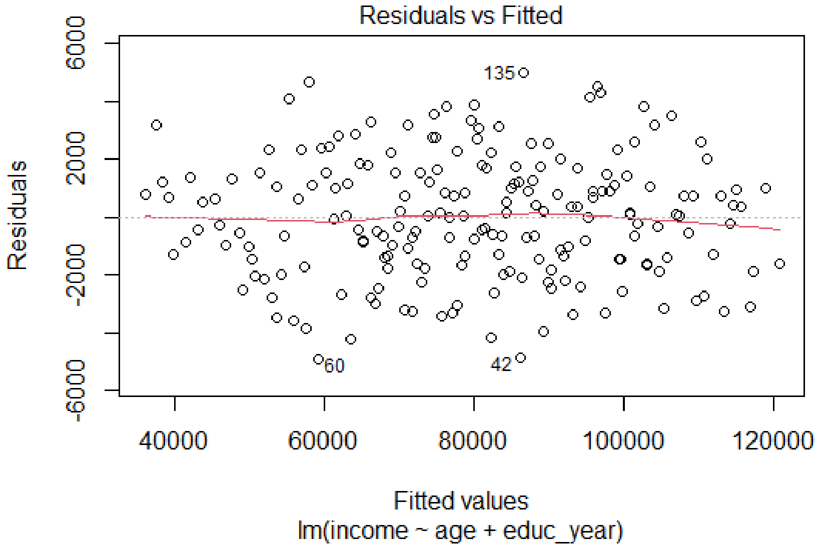


```
        Shapiro-Wilk normality test

data:  resid_omitted
W = 0.95517, p-value = 6.14e-06
```

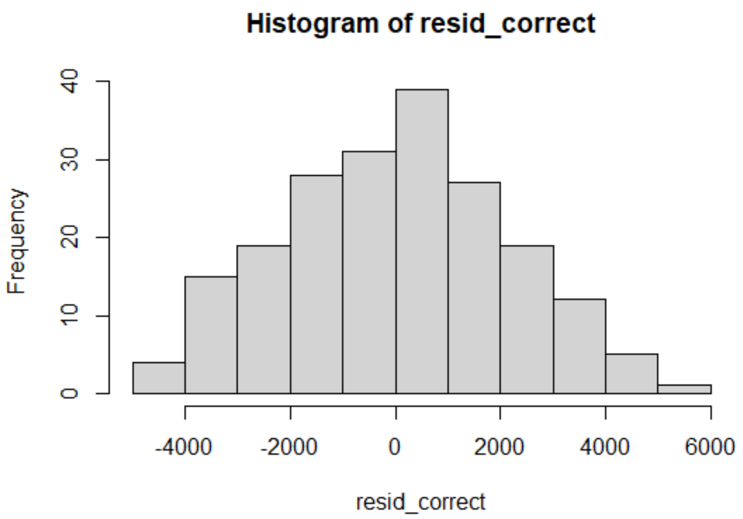
```
#----------------------------------------------------------------------
# Multiple regression with correct specification
model_correct <- lm(income ~ age + educ_year, data = df)
summary(model_correct)
```



```
# Residual diagnostics for correctly specified model
plot(model_correct, which = 1)  # Residual vs Fitted plot
```



```
resid_correct <- residuals(model_correct)
hist(resid_correct)
```

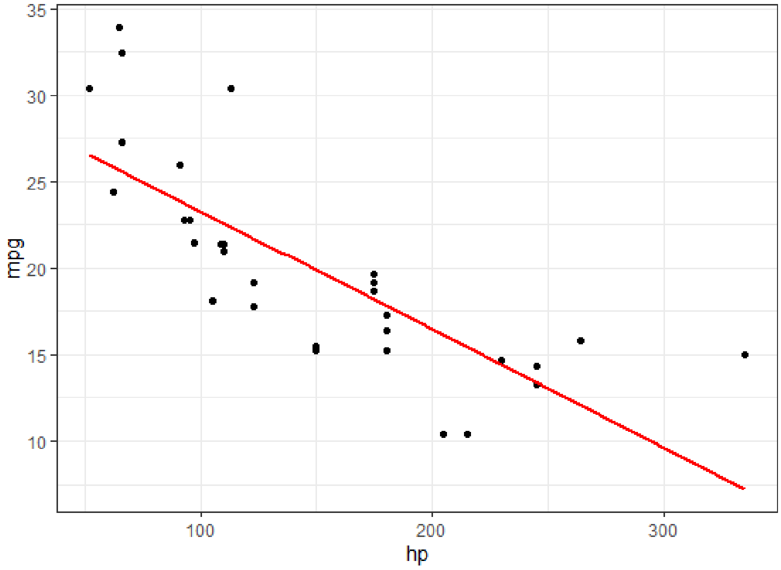


```
shapiro.test(resid_correct)  # Shapiro-Wilk test for normality
        Shapiro-Wilk normality test

data:  resid_correct
W = 0.9928, p-value = 0.4345
```

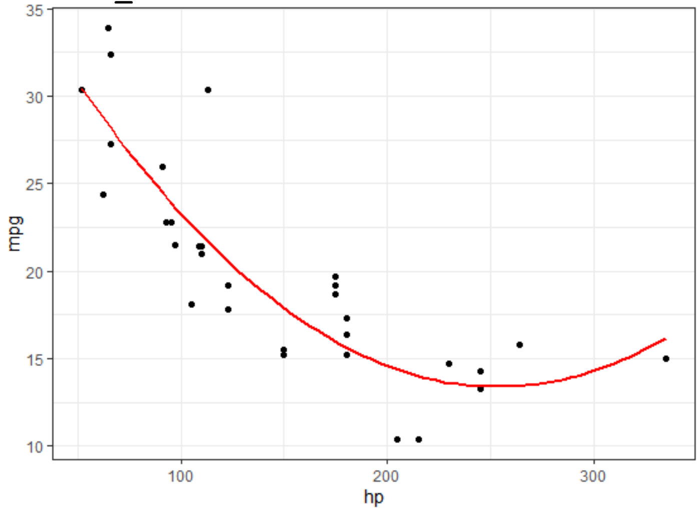
9.3. Polynomial regression

E036-polynomial_regression.R

```
library(ggplot2)

mtcars <- datasets::mtcars

#----------------------------------------
#simple regression
#----------------------------------------
fit <- lm(data = mtcars, formula = mpg ~ hp) # mpg: Miles/(US) gallon, hp:
Gross horsepower
summary(fit) #R-squared : 0.6024, Residual standard error: 3.863
Call:
lm(formula = mpg ~ hp, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7121 -2.1122 -0.8854  1.5819  8.2360 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 30.09886    1.63392  18.421  < 2e-16 ***
hp          -0.06823    0.01012  -6.742 1.79e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.863 on 30 degrees of freedom
Multiple R-squared:  0.6024,	Adjusted R-squared:  0.5892 
F-statistic: 45.46 on 1 and 30 DF,  p-value: 1.788e-07

ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  stat_smooth(method = 'lm', formula = y ~ x, color = 'red', se = FALSE) +
  theme_bw()
```



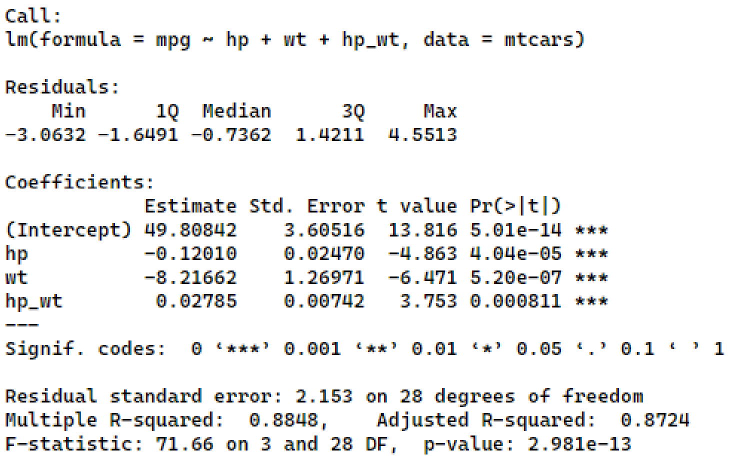
```
#---------------------------------------
#Polynomial regression regression
#---------------------------------------
fit <- lm(data = mtcars, formula = mpg ~ hp + I(hp^2))
summary(fit) #R-squared : 0.7561, Residual standard error: 3.077
Call:
lm(formula = mpg ~ hp + I(hp^2), data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.5512 -1.6027 -0.6977  1.5509  8.7213 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.041e+01  2.741e+00  14.744 5.23e-15 ***
hp          -2.133e-01  3.488e-02  -6.115 1.16e-06 ***
I(hp^2)      4.208e-04  9.844e-05   4.275 0.000189 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.077 on 29 degrees of freedom
Multiple R-squared:  0.7561,	Adjusted R-squared:  0.7393 
F-statistic: 44.95 on 2 and 29 DF,  p-value: 1.301e-09

ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  stat_smooth(method = 'lm', formula = y ~ x + I(x^2), color = 'red', se =
FALSE) +
  theme_bw()
```



9.4. Regression with interaction term

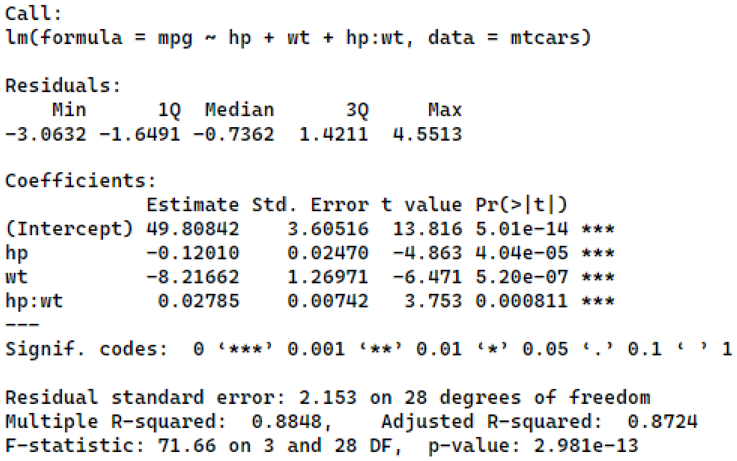
E037-regression_with_interaction.R

```
mtcars <- datasets::mtcars

#generating a new interaction term hp * wt
mtcars$hp_wt <- mtcars$hp * mtcars$wt

fit <- lm(mpg ~ hp + wt + hp_wt, data=mtcars)
summary(fit)
```



```
Call:
lm(formula = mpg ~ hp + wt + hp_wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0632 -1.6491 -0.7362  1.4211  4.5513 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 49.80842    3.60516  13.816 5.01e-14 ***
hp          -0.12010    0.02470  -4.863 4.04e-05 ***
wt          -8.21662    1.26971  -6.471 5.20e-07 ***
hp_wt        0.02785    0.00742   3.753 0.000811 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.153 on 28 degrees of freedom
Multiple R-squared:  0.8848,	Adjusted R-squared:  0.8724 
F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13
```

```
#OR

fit <- lm(mpg ~ hp + wt + hp:wt, data=mtcars)
summary(fit)
```

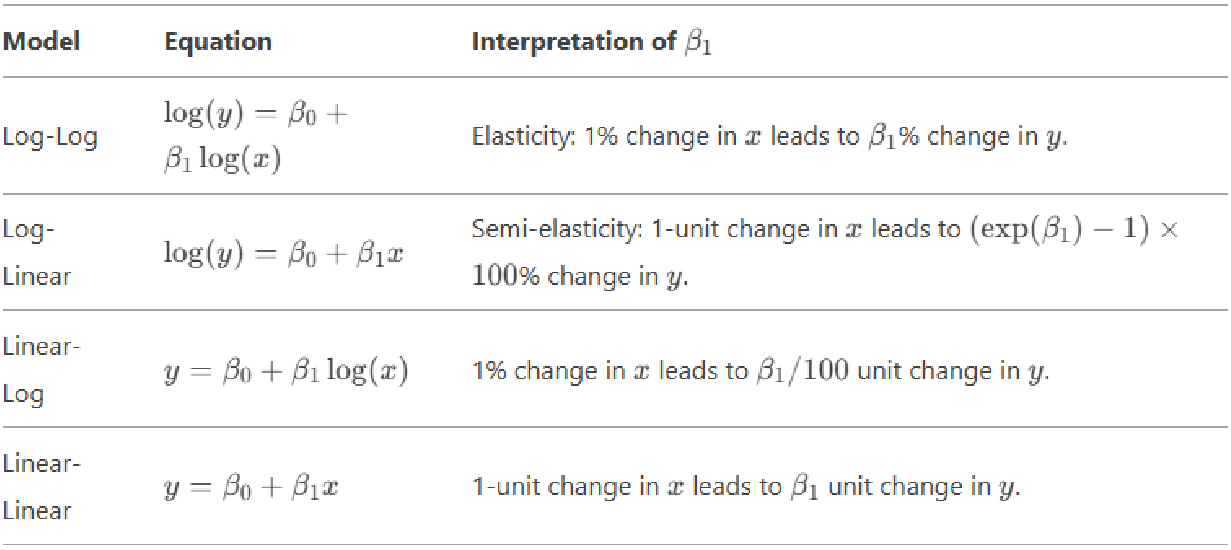


```
Call:
lm(formula = mpg ~ hp + wt + hp:wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0632 -1.6491 -0.7362  1.4211  4.5513 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 49.80842    3.60516  13.816 5.01e-14 ***
hp          -0.12010    0.02470  -4.863 4.04e-05 ***
wt          -8.21662    1.26971  -6.471 5.20e-07 ***
hp:wt        0.02785    0.00742   3.753 0.000811 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.153 on 28 degrees of freedom
Multiple R-squared:  0.8848,	Adjusted R-squared:  0.8724 
F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13
```

```
#* ---------------------------------------------------------------
#* A significant coefficient of interaction term indicates that
#* the relationship between mpg and hp varies by wt. Similarly,
#* the relationship between mpg and wt varies by hp.
#* ---------------------------------------------------------------

# d(mpg)/d(hp) = - 0.12010 + 0.02785 * wt
wt = 1
print(- 0.12010 + 0.02785 * wt) #-0.09225

wt = 2
print(- 0.12010 + 0.02785 * wt) #-0.0644

wt = 3
print(- 0.12010 + 0.02785 * wt) #-0.03655


# d(mpg)/d(wt) = - 8.21662 + 0.02785 * hp
hp = 100
```

```
print(- 8.21662 + 0.02785 * hp) #-5.43162

hp = 150
print(- 8.21662 + 0.02785 * hp) #-4.03912

hp = 200
print(- 8.21662 + 0.02785 * hp) #-2.64662
```

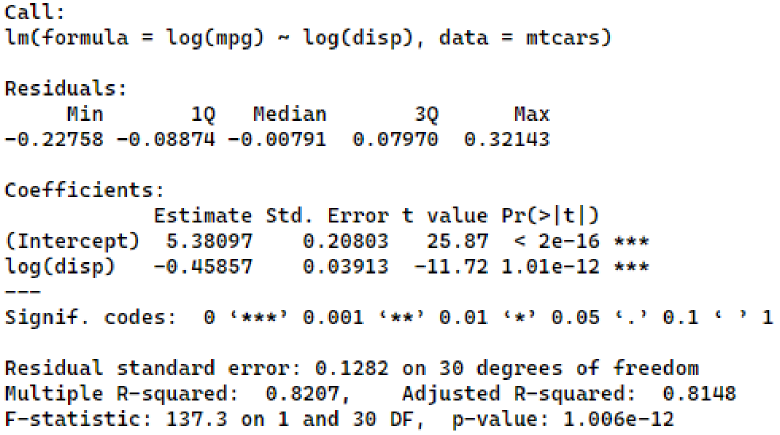
9.5. Logarithmic regression



Model	Equation	Interpretation of β_1
Log-Log	$\log(y) = \beta_0 + \beta_1 \log(x)$	Elasticity: 1% change in x leads to β_1% change in y.
Log-Linear	$\log(y) = \beta_0 + \beta_1 x$	Semi-elasticity: 1-unit change in x leads to $(\exp(\beta_1) - 1) \times 100\%$ change in y.
Linear-Log	$y = \beta_0 + \beta_1 \log(x)$	1% change in x leads to $\beta_1/100$ unit change in y.
Linear-Linear	$y = \beta_0 + \beta_1 x$	1-unit change in x leads to β_1 unit change in y.

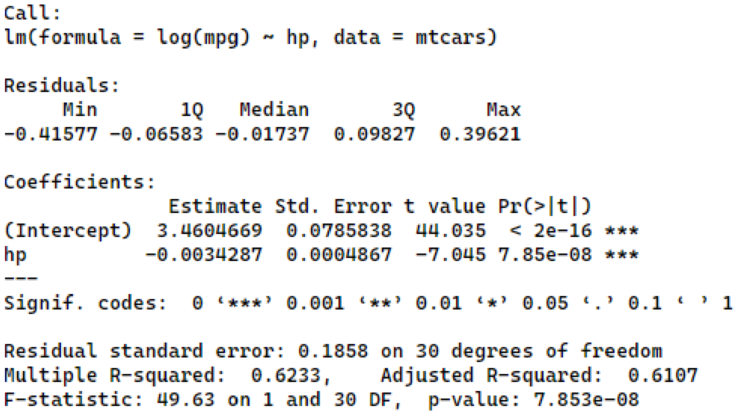
E038-logarithmic_regression.R

```
#-----------------------------------------------------------
# Log-Log Regression
#-----------------------------------------------------------
# Load data
mtcars <- datasets::mtcars

# Log-log regression
model_loglog <- lm(log(mpg) ~ log(disp), data = mtcars)
summary(model_loglog)
```



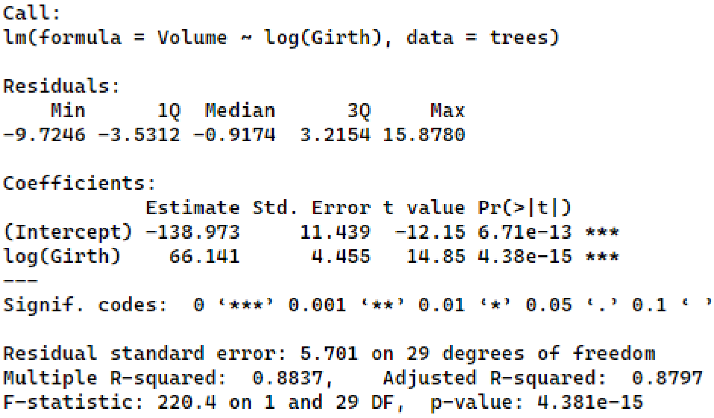
```
Call:
lm(formula = log(mpg) ~ log(disp), data = mtcars)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.22758 -0.08874 -0.00791  0.07970  0.32143 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.38097    0.20803   25.87  < 2e-16 ***
log(disp)   -0.45857    0.03913  -11.72 1.01e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1282 on 30 degrees of freedom
Multiple R-squared:  0.8207,	Adjusted R-squared:  0.8148 
F-statistic: 137.3 on 1 and 30 DF,  p-value: 1.006e-12
```

```
# A 1% increase in Displacement (cu.in.) reduces Miles/(US) gallon by
~0.46%.

#-----------------------------------------------------------
# Log-Linear Regression
#-----------------------------------------------------------
# Log-linear regression
model_loglin <- lm(log(mpg) ~ hp, data = mtcars)
summary(model_loglin)
```



```
Call:
lm(formula = log(mpg) ~ hp, data = mtcars)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.41577 -0.06583 -0.01737  0.09827  0.39621 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.4604669  0.0785838  44.035  < 2e-16 ***
hp          -0.0034287  0.0004867  -7.045 7.85e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1858 on 30 degrees of freedom
Multiple R-squared:  0.6233,	Adjusted R-squared:  0.6107 
F-statistic: 49.63 on 1 and 30 DF,  p-value: 7.853e-08
```

```
# A 1-unit increase in horsepower reduces MPG by ~0.34% (exp(-0.0034287) - 1 ≈ -0.003422829).

#----------------------------------------------------------
# Linear-Log Regression
#----------------------------------------------------------
# Load data
trees <- datasets::trees

# Linear-log regression
model_linlog <- lm(Volume ~ log(Girth), data = trees)
summary(model_linlog)
```



```
Call:
lm(formula = Volume ~ log(Girth), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.7246 -3.5312 -0.9174  3.2154 15.8780 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -138.973     11.439  -12.15 6.71e-13 ***
log(Girth)    66.141      4.455   14.85 4.38e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.701 on 29 degrees of freedom
Multiple R-squared:  0.8837,	Adjusted R-squared:  0.8797 
F-statistic: 220.4 on 1 and 29 DF,  p-value: 4.381e-15
```

```
# A 1% increase in girth increases volume by ~ 0.66 units (66.141 / 100).
```

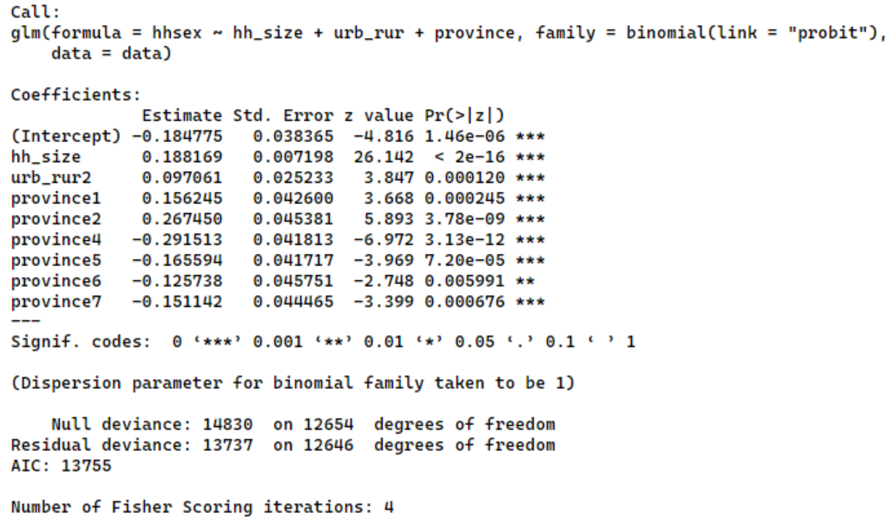
Session 10: Logistic regression

11.1. Logistic regression

Logistic regression is useful when you're predicting a binary outcome from a set of continuous and/or categorical predictor variables.

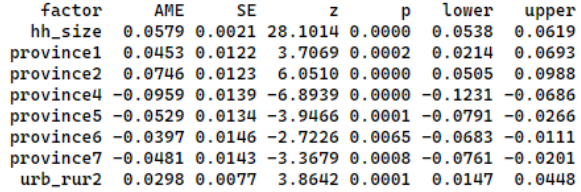
E039-logistic_regression.R

```
# Load necessary libraries
library(haven)  # For reading SPSS files
library(dplyr)    # For data manipulation
library(margins)  # For calculating marginal effects

# Import SPSS file from the URL
data <- read_spss('data/010-hh.sav')

# Dropping missing values in HHSEX
data <- data %>% filter(!is.na(HHSEX))

# Creating new variables
data <- data %>%
  mutate(
    hh_size = HH48,  # HH member size variable
    urb_rur = factor(HH6),  # 1=Urban 2=Rural
    province = factor(HH7),  # Province number
    hhsex = factor(HHSEX) # 1=Male 2=Female
  )

#setting 1=Urban as reference/base
data$urb_rur <- relevel(data$urb_rur, ref = '1')

#setting 2=Female as reference/base
data$hhsex <- relevel(data$hhsex, ref = '2')

#setting province 3 as base category/reference level
data$province <- relevel(data$province, ref = '3')

#-----------------------------------------------
# Running logistic regression
#-----------------------------------------------
logit_model <- glm(hhsex ~ hh_size + urb_rur + province,
                   data = data, family = binomial(link = "logit"))
summary(logit_model)
Call:
glm(formula = hhsex ~ hh_size + urb_rur + province, family = binomial(link = "logit"), 
    data = data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.39332    0.06488  -6.062 1.35e-09 ***
hh_size      0.33308    0.01274  26.145  < 2e-16 ***
urb_rur2     0.16929    0.04273   3.962 7.44e-05 ***
province1    0.24989    0.07272   3.436 0.000590 ***
province2    0.46344    0.07949   5.830 5.53e-09 ***
province4   -0.47811    0.06912  -6.917 4.61e-12 ***
province5   -0.28721    0.06988  -4.110 3.95e-05 ***
province6   -0.22815    0.07692  -2.966 0.003017 ** 
province7   -0.25820    0.07476  -3.454 0.000553 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14830  on 12654  degrees of freedom
Residual deviance: 13722  on 12646  degrees of freedom
AIC: 13740

Number of Fisher Scoring iterations: 4

# Calculating marginal effects for logistic regression
logit_margins <- margins(logit_model)
summary(logit_margins)
    factor     AME     SE       z      p   lower   upper
   hh_size  0.0606 0.0021 28.5476 0.0000  0.0564  0.0647
 province1  0.0424 0.0122  3.4806 0.0005  0.0185  0.0663
 province2  0.0747 0.0123  6.0485 0.0000  0.0505  0.0989
 province4 -0.0936 0.0137 -6.8339 0.0000 -0.1205 -0.0668
 province5 -0.0545 0.0134 -4.0804 0.0000 -0.0806 -0.0283
 province6 -0.0428 0.0146 -2.9299 0.0034 -0.0715 -0.0142
 province7 -0.0487 0.0143 -3.4141 0.0006 -0.0767 -0.0208
  urb_rur2  0.0307 0.0077  3.9854 0.0001  0.0156  0.0457

#-----------------------------------------------
# Running probit regression
#-----------------------------------------------
probit_model <- glm(hhsex ~ hh_size + urb_rur + province,
                    data = data, family = binomial(link = "probit"))
summary(probit_model)
```



```
Call:
glm(formula = hhsex ~ hh_size + urb_rur + province, family = binomial(link = "probit"),
    data = data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.184775   0.038365  -4.816 1.46e-06 ***
hh_size      0.188169   0.007198  26.142  < 2e-16 ***
urb_rur2     0.097061   0.025233   3.847 0.000120 ***
province1    0.156245   0.042600   3.668 0.000245 ***
province2    0.267450   0.045381   5.893 3.78e-09 ***
province4   -0.291513   0.041813  -6.972 3.13e-12 ***
province5   -0.165594   0.041717  -3.969 7.20e-05 ***
province6   -0.125738   0.045751  -2.748 0.005991 **
province7   -0.151142   0.044465  -3.399 0.000676 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14830  on 12654  degrees of freedom
Residual deviance: 13737  on 12646  degrees of freedom
AIC: 13755

Number of Fisher Scoring iterations: 4
```

```
# Calculating marginal effects for probit regression
probit_margins <- margins(probit_model)
summary(probit_margins)
```



```
    factor     AME     SE       z      p   lower   upper
   hh_size  0.0579 0.0021 28.1014 0.0000  0.0538  0.0619
 province1  0.0453 0.0122  3.7069 0.0002  0.0214  0.0693
 province2  0.0746 0.0123  6.0510 0.0000  0.0505  0.0988
 province4 -0.0959 0.0139 -6.8939 0.0000 -0.1231 -0.0686
 province5 -0.0529 0.0134 -3.9466 0.0001 -0.0791 -0.0266
 province6 -0.0397 0.0146 -2.7226 0.0065 -0.0683 -0.0111
 province7 -0.0481 0.0143 -3.3679 0.0008 -0.0761 -0.0201
  urb_rur2  0.0298 0.0077  3.8642 0.0001  0.0147  0.0448
```

Task 8:

Using NMICS6 data (011-Affairs.RData), complete the following tasks.

i. Load the **011-Affairs.RData**

ii. Tabulate the frequency of **affairs** variable from **Affairs** dataframe.

iii. Create a variable **ynaffairs** in **Affairs** dataframe such that the variable takes value 0 if no affairs and 1 if the person is involved in affairs.

iv. Set **ynaffairs** and **rating** variables as factor variables.

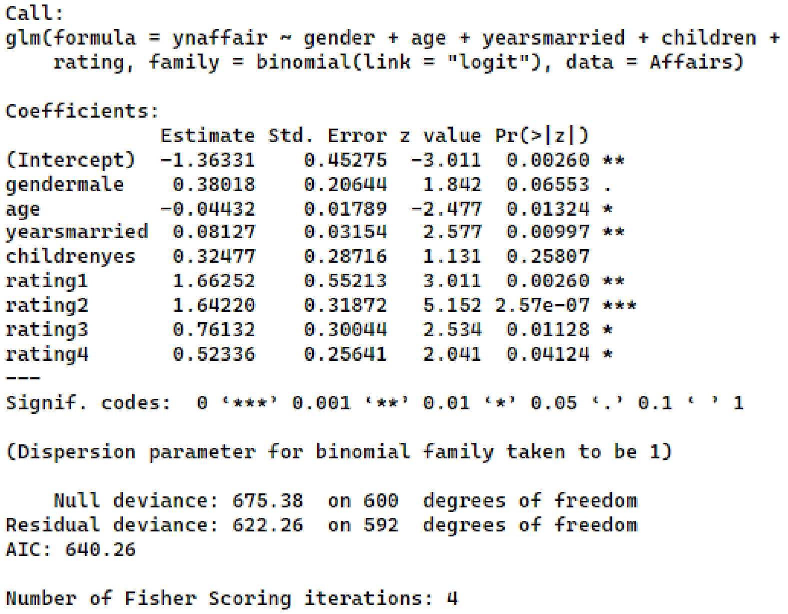
v. Set '0' as reference for **ynaffairs** variable, '5' for **rating**, 'no' for **children**, and 'female' for **gender** variables.

vi. Fit a logistic regression model with **ynaffairs** as dependent variable and **gender, age, yearsmarried, children, rating** as independent variable.

vii. Calculate average marginal effect for each variables using the margins() function.

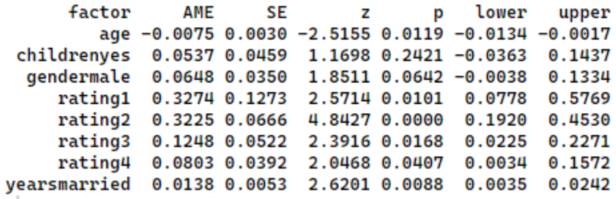
```
library(dplyr)
library(margins)

load('data/011-Affairs.RData')
table(Affairs$affairs)
  0   1   2   3   7  12 
451  34  17  19  42  38 

Affairs <- Affairs %>% mutate(ynaffair = case_when(affairs > 0 ~ 1, TRUE ~ 
0),
                              ynaffair = factor(ynaffair),
                              rating = factor(rating))
table(Affairs$ynaffair)
  0   1 
451 150 

#setting 0 : No-Affairs as base/reference
Affairs$ynaffair <- relevel(Affairs$ynaffair, ref = '0')
```

```
#setting 5 : Very happy as base/reference
# 1 = very unhappy, 2 = somewhat unhappy, 3 = average, 4 = happier than
average, 5 = very happy.
Affairs$rating <- relevel(Affairs$rating, ref = '5')

#setting no children as base/reference
Affairs$children <- relevel(Affairs$children, ref = 'no')

#setting female as base/reference
Affairs$gender <- relevel(Affairs$gender, ref = 'female')

fit <- glm(ynaffair ~ gender
                + age
                + yearsmarried
                + children
                + rating,
                data=Affairs,
                family = binomial(link = "logit"))

summary(fit)
```



```
Call:
glm(formula = ynaffair ~ gender + age + yearsmarried + children +
    rating, family = binomial(link = "logit"), data = Affairs)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.36331    0.45275  -3.011  0.00260 **
gendermale    0.38018    0.20644   1.842  0.06553 .
age          -0.04432    0.01789  -2.477  0.01324 *
yearsmarried  0.08127    0.03154   2.577  0.00997 **
childrenyes   0.32477    0.28716   1.131  0.25807
rating1       1.66252    0.55213   3.011  0.00260 **
rating2       1.64220    0.31872   5.152 2.57e-07 ***
rating3       0.76132    0.30044   2.534  0.01128 *
rating4       0.52336    0.25641   2.041  0.04124 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 675.38  on 600  degrees of freedom
Residual deviance: 622.26  on 592  degrees of freedom
AIC: 640.26

Number of Fisher Scoring iterations: 4
```

```
summary(margins(fit))
```



```
      factor     AME     SE       z      p   lower   upper
         age -0.0075 0.0030 -2.5155 0.0119 -0.0134 -0.0017
 childrenyes  0.0537 0.0459  1.1698 0.2421 -0.0363  0.1437
  gendermale  0.0648 0.0350  1.8511 0.0642 -0.0038  0.1334
     rating1  0.3274 0.1273  2.5714 0.0101  0.0778  0.5769
     rating2  0.3225 0.0666  4.8427 0.0000  0.1920  0.4530
     rating3  0.1248 0.0522  2.3916 0.0168  0.0225  0.2271
     rating4  0.0803 0.0392  2.0468 0.0407  0.0034  0.1572
yearsmarried  0.0138 0.0053  2.6201 0.0088  0.0035  0.0242
```

Session 11: Time-series analysis

11.1. Stationarity concept

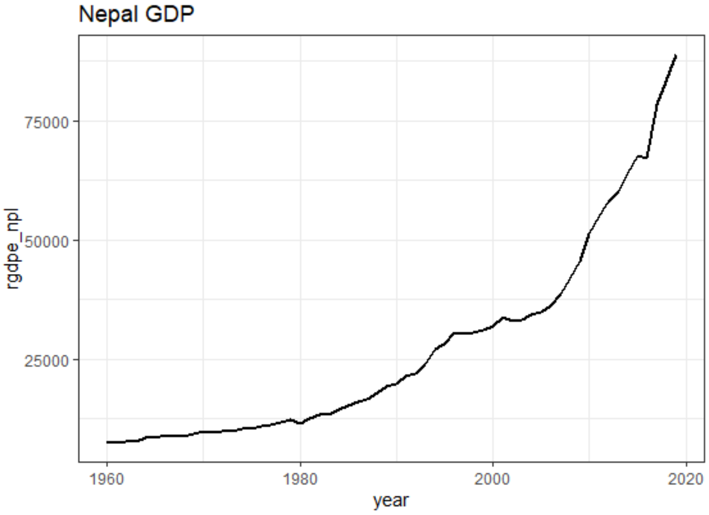
- Stationarity refers to a time series whose statistical properties, such as mean, variance, and autocorrelation, remain constant over time.
- Non-stationary series are prone to spurious relationships.

11.2. Spurious relationships

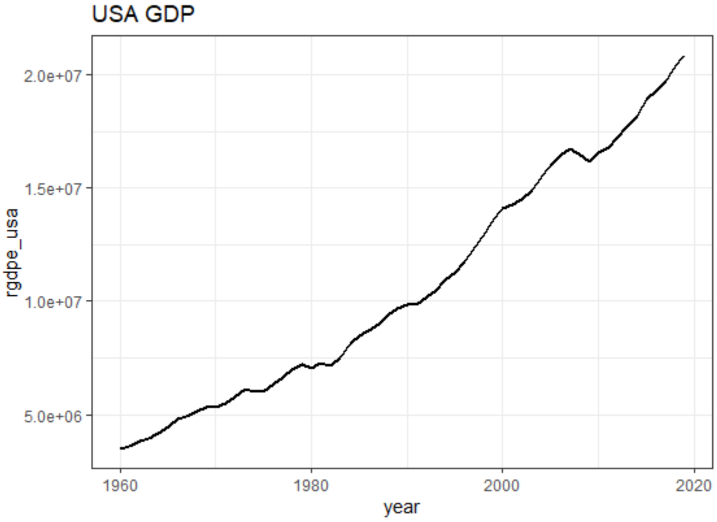
E040-spurious_regression.R

```
library(haven)
library(dplyr)
library(tseries)
library(ggplot2)

df <- read_dta('data/012-pwt1001.dta')
```

```
#keeping real GDP of Nepal from 1960 onwards
npl <- df %>%
  filter(countrycode == 'NPL' & year >= 1960) %>%
  select(year, rgdpe) %>%
  rename(rgdpe_npl = rgdpe)

#keeping real GDP of USA from 1960 onwards
usa <- df %>%
  filter(countrycode == 'USA' & year >= 1960) %>%
  select(year, rgdpe) %>%
  rename(rgdpe_usa = rgdpe)

#joining Nepal and USA data into one dataframe
df_npl_usa <- full_join(npl, usa, by = 'year')

#Visual inspection of stationarity
ggplot() +
  geom_line(data = df_npl_usa, aes(x=year, y=rgdpe_npl), size = 1) +
  labs(title = 'Nepal GDP') +
  theme_bw()
```



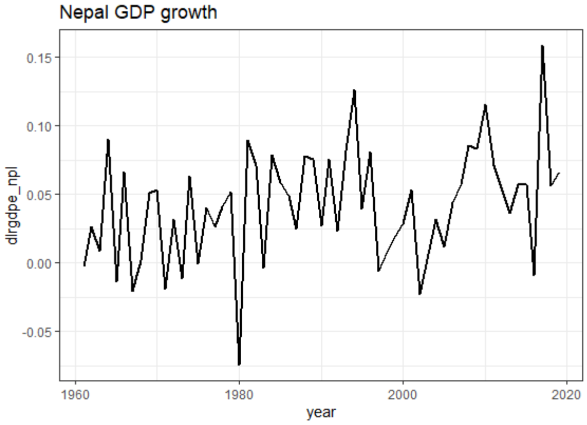
```
ggplot() +
  geom_line(data = df_npl_usa, aes(x=year, y=rgdpe_usa), size = 1) +
  labs(title = 'USA GDP') +
  theme_bw()
```



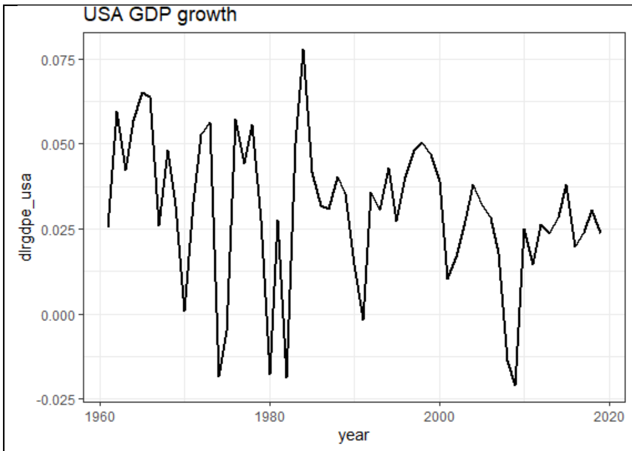
```
#Hypothesis testing of stationarity
adf.test(df_npl_usa$rgdpe_npl)
```

```
	Augmented Dickey-Fuller Test

data:  df_npl_usa$rgdpe_npl
Dickey-Fuller = 1.9811, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary
```

```
adf.test(df_npl_usa$rgdpe_usa)
        Augmented Dickey-Fuller Test

data:  df_npl_usa$rgdpe_usa
Dickey-Fuller = -1.1308, Lag order = 3, p-value = 0.9101
alternative hypothesis: stationary


#Running a regression (Spurious regression observed)
fit <- lm(formula = rgdpe_usa ~ rgdpe_npl ,data = df_npl_usa)
summary(fit)
Call:
lm(formula = rgdpe_usa ~ rgdpe_npl, data = df_npl_usa)

Residuals:
     Min       1Q   Median       3Q      Max 
-3982784 -1070155   -25168   853922  3683084 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4384943.5   369375.8   11.87   <2e-16 ***
rgdpe_npl       230.4       10.7   21.54   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1741000 on 58 degrees of freedom
Multiple R-squared:  0.8888,	Adjusted R-squared:  0.8869 
F-statistic: 463.8 on 1 and 58 DF,  p-value: < 2.2e-16


#-----------------------------------------------------------------
# Making series stationary and repeating the above steps
#-----------------------------------------------------------------
df_npl_usa <- df_npl_usa %>%
  mutate(dlrgdpe_npl = c(NA,diff(log(rgdpe_npl))),
         dlrgdpe_usa = c(NA,diff(log(rgdpe_usa)))) %>%
  na.omit()

#Visual inspection of stationarity
ggplot() +
  geom_line(data = df_npl_usa, aes(x=year, y=dlrgdpe_npl), size = 1) +
  labs(title = 'Nepal GDP growth') +
  theme_bw()
```



```
ggplot() +
  geom_line(data = df_npl_usa, aes(x=year, y=dlrgdpe_usa), size = 1) +
  labs(title = 'USA GDP growth') +
  theme_bw()
```

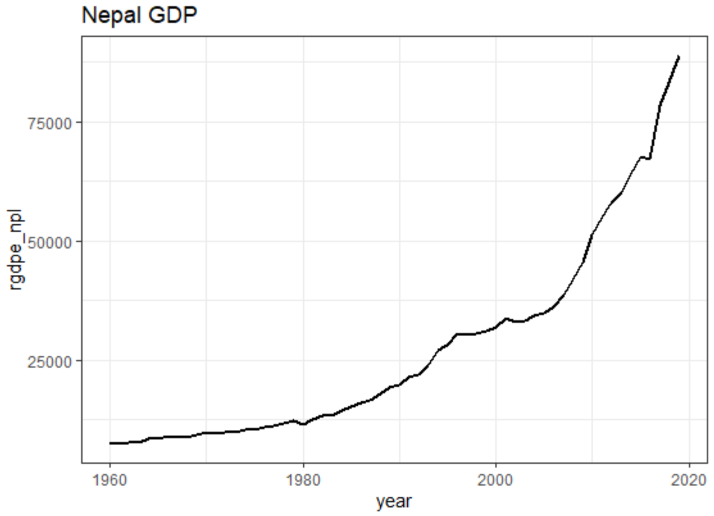


```
#Hypothesis testing of stationarity
adf.test(df_npl_usa$dlrgdpe_npl)
         Augmented Dickey-Fuller Test

data:  df_npl_usa$dlrgdpe_npl
Dickey-Fuller = -3.236, Lag order = 3, p-value = 0.0904
alternative hypothesis: stationary


adf.test(df_npl_usa$dlrgdpe_usa)
         Augmented Dickey-Fuller Test

data:  df_npl_usa$dlrgdpe_usa
Dickey-Fuller = -4.4078, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary

#Running a regression (no spurious regression observed)
fit <- lm(formula = dlrgdpe_usa ~ dlrgdpe_npl ,data = df_npl_usa)
summary(fit)
Call:
lm(formula = dlrgdpe_usa ~ dlrgdpe_npl, data = df_npl_usa)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.052298 -0.006704  0.000871  0.014589  0.048773 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.031797   0.004109   7.738 1.87e-10 ***
dlrgdpe_npl -0.035680   0.070464  -0.506    0.615    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02206 on 57 degrees of freedom
Multiple R-squared:  0.004478,	Adjusted R-squared:  -0.01299 
F-statistic: 0.2564 on 1 and 57 DF,  p-value: 0.6146
```

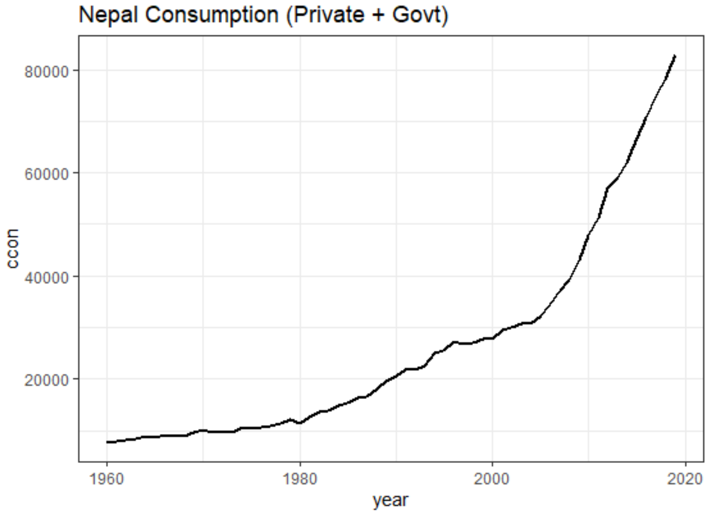
11.3. True relationships

E041-actual_relationship.R

```
library(haven)
library(dplyr)
library(tseries)
library(ggplot2)

df <- read_dta('data/012-pwt1001.dta')

df <- filter(df, countrycode == 'NPL' &  year >= 1960) %>% select(year,
rgdpe, ccon)

#Visual inspection of stationarity
ggplot() +
  geom_line(data = df, aes(x=year, y=rgdpe), size = 1) +
  labs(title = 'Nepal GDP') +
  theme_bw()
```



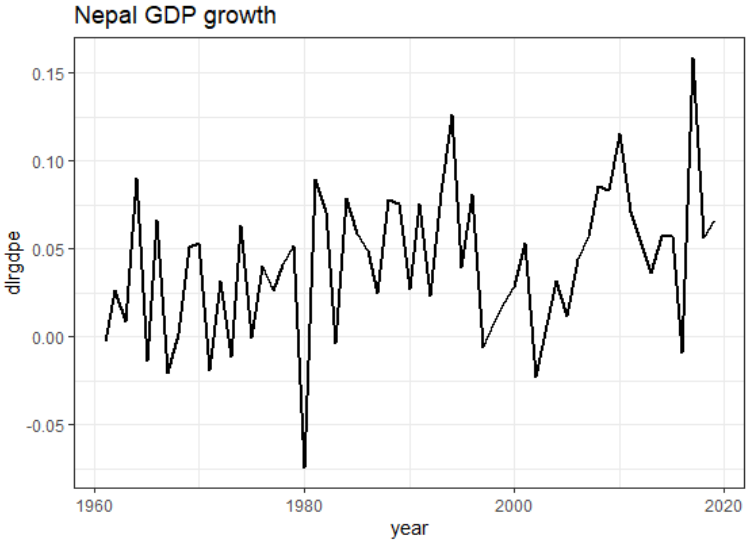
```
ggplot() +
  geom_line(data = df, aes(x=year, y=ccon), size = 1) +
  labs(title = 'Nepal Consumption (Private + Govt)') +
  theme_bw()
```



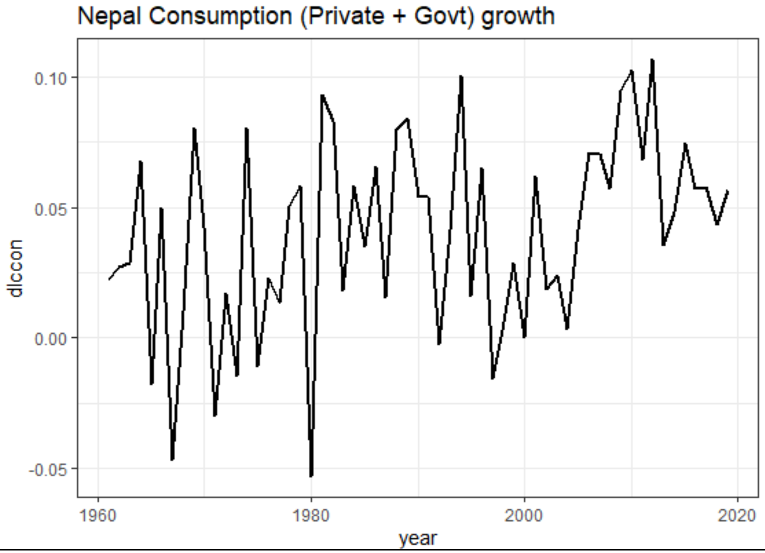
```
#Hypothesis testing of stationarity
adf.test(df$rgdpe)
        Augmented Dickey-Fuller Test

data:  df$rgdpe
Dickey-Fuller = 1.9811, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary


adf.test(df$ccon)
        Augmented Dickey-Fuller Test

data:  df$ccon
Dickey-Fuller = 0.87741, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary


#Running a regression
fit <- lm(formula = rgdpe ~ ccon ,data = df)
summary(fit)
```

```
Call:
lm(formula = rgdpe ~ ccon, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-7032.0  -695.9  -295.4  1007.2  3166.8 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -157.77337  331.29087  -0.476    0.636    
ccon           1.04900    0.01004 104.504   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1554 on 58 degrees of freedom
Multiple R-squared:  0.9947,	Adjusted R-squared:  0.9946 
F-statistic: 1.092e+04 on 1 and 58 DF,  p-value: < 2.2e-16
```

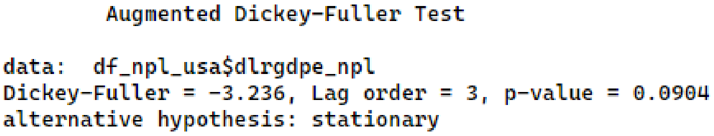
```
#-------------------------------------------------------------
# Making series stationary and repeating the above steps
#-------------------------------------------------------------
df <- df %>%
  mutate(dlrgdpe = c(NA,diff(log(rgdpe))),
         dlccon = c(NA,diff(log(ccon)))) %>%
  na.omit()

#Visual inspection of stationarity
ggplot() +
  geom_line(data = df, aes(x=year, y=dlrgdpe), size = 1) +
  labs(title = 'Nepal GDP growth') +
  theme_bw()
```



```
ggplot() +
  geom_line(data = df, aes(x=year, y=dlccon), size = 1) +
  labs(title = 'Nepal Consumption (Private + Govt) growth') +
  theme_bw()
```

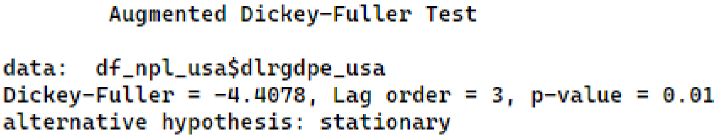


```
#Hypothesis testing of stationarity
adf.test(df_npl_usa$dlrgdpe_npl)
```



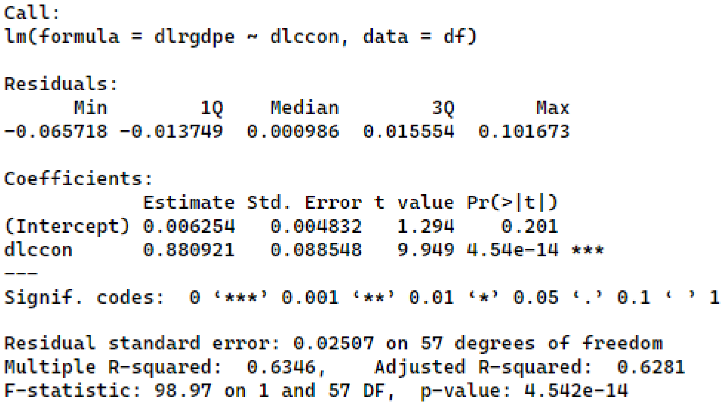
```
	Augmented Dickey-Fuller Test

data:  df_npl_usa$dlrgdpe_npl
Dickey-Fuller = -3.236, Lag order = 3, p-value = 0.0904
alternative hypothesis: stationary
```

```
adf.test(df_npl_usa$dlrgdpe_usa)
```



```
	Augmented Dickey-Fuller Test

data:  df_npl_usa$dlrgdpe_usa
Dickey-Fuller = -4.4078, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

```
#Running a regression
fit <- lm(formula = dlrgdpe ~ dlccon ,data = df)
summary(fit)
```



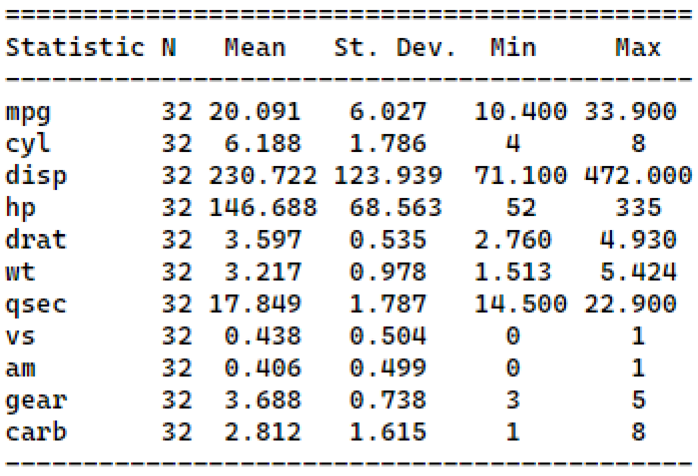
```
Call:
lm(formula = dlrgdpe ~ dlccon, data = df)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.065718 -0.013749  0.000986  0.015554  0.101673 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.006254   0.004832   1.294    0.201    
dlccon      0.880921   0.088548   9.949 4.54e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02507 on 57 degrees of freedom
Multiple R-squared:  0.6346,	Adjusted R-squared:  0.6281 
F-statistic: 98.97 on 1 and 57 DF,  p-value: 4.542e-14
```

Session 12: Stargazer for reporting regression results and the project work

12.1. stargazer

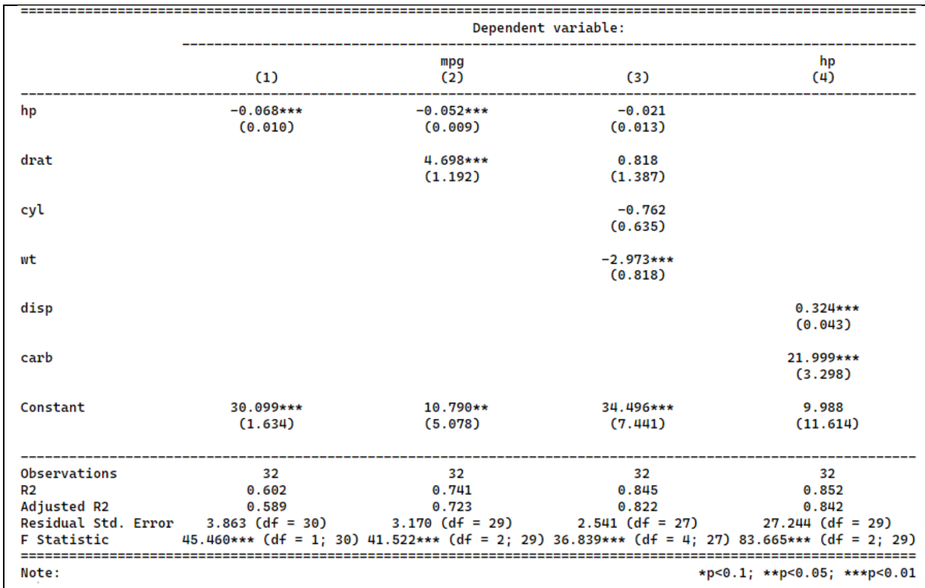
E042-stargazer.R

```
mtcars <- datasets::mtcars

model1 <- lm(mpg ~ hp, data = mtcars)
model2 <- lm(mpg ~ hp + drat, data = mtcars)
model3 <- lm(mpg ~ hp + drat + cyl + wt, data = mtcars)
model4 <- lm(hp ~ disp + carb, data = mtcars)

library(stargazer)

#descriptive statistics table
stargazer(mtcars, type = 'text')
```



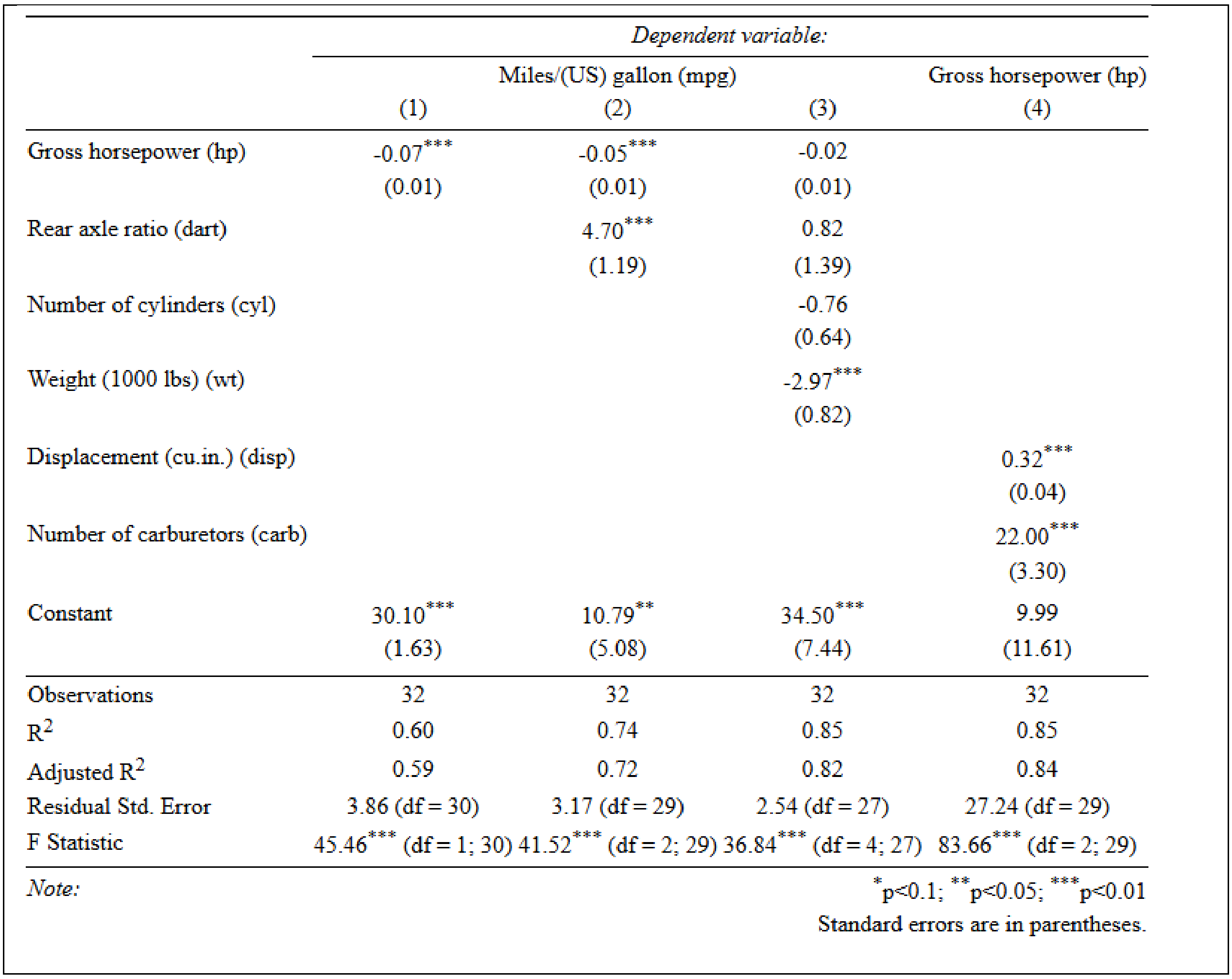
```
============================================
Statistic N   Mean   St. Dev.  Min     Max  
--------------------------------------------
mpg       32 20.091   6.027   10.400 33.900 
cyl       32  6.188   1.786     4       8   
disp      32 230.722 123.939  71.100 472.000
hp        32 146.688  68.563    52     335  
drat      32  3.597   0.535   2.760   4.930 
wt        32  3.217   0.978   1.513   5.424 
qsec      32 17.849   1.787   14.500 22.900 
vs        32  0.438   0.504     0       1   
am        32  0.406   0.499     0       1   
gear      32  3.688   0.738     3       5   
carb      32  2.812   1.615     1       8   
--------------------------------------------
```

```
#displaying regression models results in a single table
stargazer(model1, model2, model3, model4, type = "text")
```



```
===============================================================================================================
                                                        Dependent variable:
                    -------------------------------------------------------------------------------------------
                                                     mpg                                           hp
                              (1)                    (2)                    (3)                    (4)
---------------------------------------------------------------------------------------------------------------
hp                         -0.068***              -0.052***              -0.021
                            (0.010)                (0.009)                (0.013)

drat                                               4.698***               0.818
                                                   (1.192)                (1.387)

cyl                                                                      -0.762
                                                                          (0.635)

wt                                                                      -2.973***
                                                                          (0.818)

disp                                                                                             0.324***
                                                                                                 (0.043)

carb                                                                                            21.999***
                                                                                                 (3.298)

Constant                   30.099***              10.790**               34.496***               9.988
                            (1.634)                (5.078)                (7.441)                (11.614)

---------------------------------------------------------------------------------------------------------------
Observations                  32                     32                     32                     32
R2                           0.602                  0.741                  0.845                  0.852
Adjusted R2                  0.589                  0.723                  0.822                  0.842
Residual Std. Error    3.863 (df = 30)        3.170 (df = 29)        2.541 (df = 27)       27.244 (df = 29)
F Statistic         45.460*** (df = 1; 30) 41.522*** (df = 2; 29) 36.839*** (df = 4; 27) 83.665*** (df = 2; 29)
===============================================================================================================
Note:                                                                               *p<0.1; **p<0.05; ***p<0.01
```

```
#defining the covariate and variable labels
stargazer(model1, model2, model3, model4, type = "text",
          digits = 2,
          covariate.labels = c('Gross horsepower (hp)',
                               'Rear axle ratio (dart)',
                               'Number of cylinders (cyl)',
                               'Weight (1000 lbs) (wt)',
                               'Displacement (cu.in.) (disp)',
                               'Number of carburetors (carb)'),
          dep.var.labels = c("Miles/(US) gallon (mpg)", "Gross horsepower
(hp)"),
          notes = "Standard errors are in parentheses.")

#export and save the result as html
stargazer(model1, model2, model3, model4, type = "html", out =
'model_results.html',
          digits = 2,
          covariate.labels = c('Gross horsepower (hp)',
                               'Rear axle ratio (dart)',
                               'Number of cylinders (cyl)',
                               'Weight (1000 lbs) (wt)',
                               'Displacement (cu.in.) (disp)',
                               'Number of carburetors (carb)'),
          dep.var.labels = c("Miles/(US) gallon (mpg)", "Gross horsepower
(hp)"),
          notes = "Standard errors are in parentheses.")
```



	Dependent variable:			
		Miles/(US) gallon (mpg)		Gross horsepower (hp)
	(1)	(2)	(3)	(4)
Gross horsepower (hp)	-0.07^{***}	-0.05^{***}	-0.02	
	(0.01)	(0.01)	(0.01)	
Rear axle ratio (dart)		4.70^{***}	0.82	
		(1.19)	(1.39)	
Number of cylinders (cyl)			-0.76	
			(0.64)	
Weight (1000 lbs) (wt)			-2.97^{***}	
			(0.82)	
Displacement (cu.in.) (disp)				0.32^{***}
				(0.04)
Number of carburetors (carb)				22.00^{***}
				(3.30)
Constant	30.10^{***}	10.79^{**}	34.50^{***}	9.99
	(1.63)	(5.08)	(7.44)	(11.61)
Observations	32	32	32	32
R^2	0.60	0.74	0.85	0.85
Adjusted R^2	0.59	0.72	0.82	0.84
Residual Std. Error	3.86 (df = 30)	3.17 (df = 29)	2.54 (df = 27)	27.24 (df = 29)
F Statistic	45.46^{***} (df = 1; 30)	41.52^{***} (df = 2; 29)	36.84^{***} (df = 4; 27)	83.66^{***} (df = 2; 29)
Note:				*p<0.1; **p<0.05; ***p<0.01 Standard errors are in parentheses.