library(wooldridge)

library(AER)

library(dplyr)

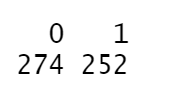
library(stargazer)

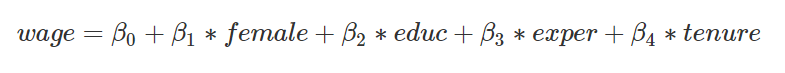
# load wage1 data from wooldridge package

data(wage1, package='wooldridge')

# show the number of men and women in the sample this is in the variable female

table(wage1$female)





# Run the regression that estimates the equation above

# First by using the variable female as a regressor

m1<-lm(wage ~ female+educ+exper+tenure, data=wage1)

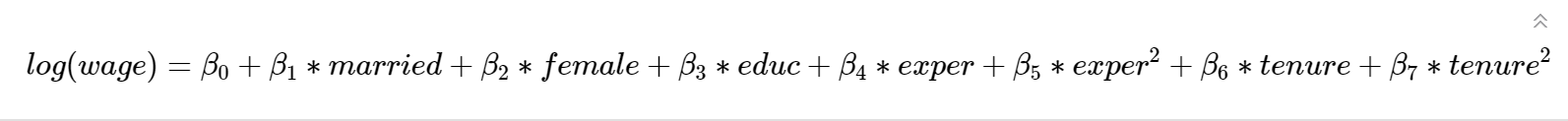
# You can also filter your data and create two separate equations but the most efficient way is to add the subset option inside the lm command data=subset()

m2<-lm(wage ~ educ+exper+tenure, data=subset(wage1, female==0))

# You need to **interact each regressor with the female variable for the models to be the same** when you restrict the sample do it below in model m3

m3<-lm(wage ~ educ\*female+exper\*female+tenure\*female, data=wage1)

* **Dummy variables and arithmetic formulas into a regression**



**Note that the following 2 models are the same**

# First using I() for the interaction and include the variables alone

M1 <- lm(log(wage)~ married + female + educ + exper +I(female\*educ) + I(exper^2) + tenure + I(tenure^2), data=wage1)

# The other option use only the interaction, there is no need to include the variables alone R does it.

M2 <- lm(log(wage)~married + female\*educ + exper + I(exper^2) + tenure + I(tenure^2), data=wage1)

* **Boolean Variable**

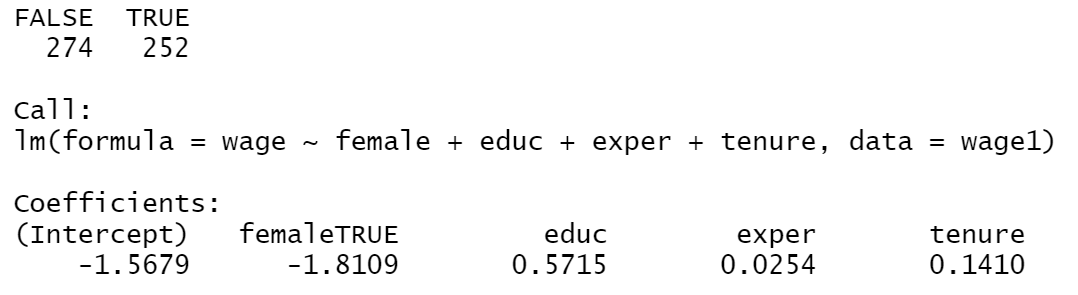
# replace "female" with logical variable using function as.logical()

wage1$female <- as.logical(wage1$female)

table(wage1$female)

# regression with logical variable

lm(wage ~ female+educ+exper+tenure, data=wage1)



* A factor variable is the way R uses to store categorical variables. We transform any categorical variable into a factor `as.factor()`. Factor variables can be directly added to the list of regressors. R implicitly adds \*g-1\* dummy variables if the factor has g outcomes (categories).

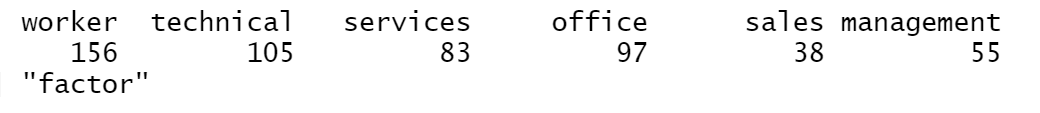
data(CPS1985,package="AER")

# Table of categories and frequencies for factor variable occupation:

table(CPS1985$occupation)

# What type of variable is occupation

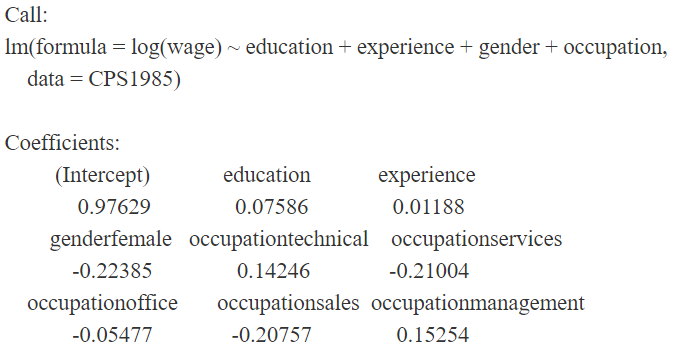
class(CPS1985$occupation)



* **Dummies for many categories from a categorical variable**

# Directly using factor variables in regression formula:

lm(log(wage) ~ education+experience+gender+occupation, data=CPS1985)



# *Manually redefine the reference category for gender*

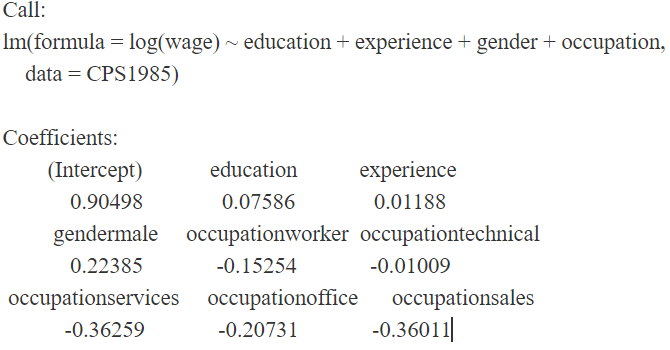
CPS1985$gender <- **relevel**(CPS1985$gender,"female")

# the coefficients for ocupation are now redefine and managment is the reference category

CPS1985$occupation<-**relevel**(CPS1985$occupation,"management")

# Rerun regression with the new based categories:

lm(log(wage) ~ education+experience+gender+occupation, data=CPS1985)



# You can convert a categorical variable into numerical one

CPS1985$occupation<-**as.numeric**(CPS1985$occupation)

* **Numeric variables into categories**

Sometimes we need to make numerical variables into categories because a linear relation with the dependent variable seems implausible or the interpretation is inconvenient. Or we simply want to have a different interpretation.

data(lawsch85, package='wooldridge')

# Define cut points for the rank

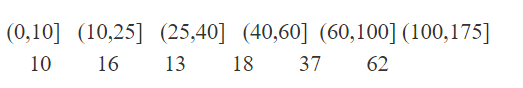
*cutpts* <- c(0,10,25,40,60,100,175)

# Create factor variable containing ranges for the rank use function cut()

lawsch85$rankcat <**- cut**(lawsch85$rank, *cutpts*)

# Display frequencies using table

table(lawsch85$rankcat)



# Choose reference category, we want the last group as the reference category, so we use relevel. Save that in a new variable called rankcat

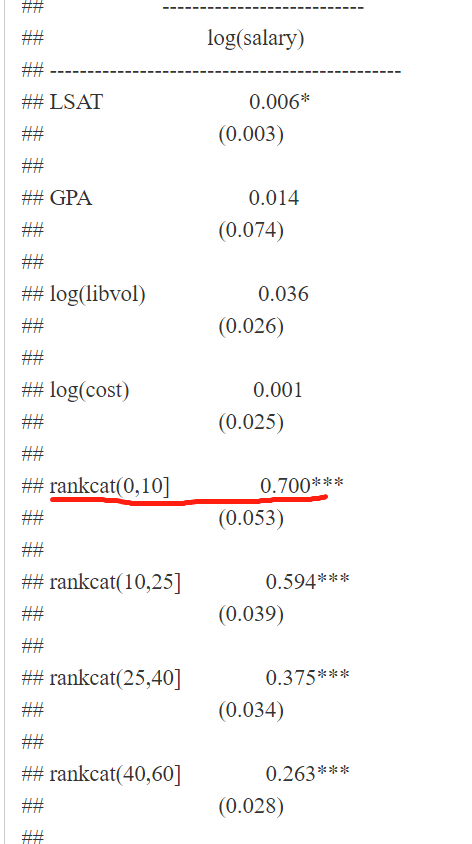
lawsch85$rankcat <- relevel(lawsch85$rankcat,"(100,175]")

# Run regression

res <- lm(log(salary)~ LSAT+GPA+log(libvol)+log(cost)+rankcat, data=lawsch85)

stargazer(res, type="text")

# This regression implies that graduates from the top 10 schools collect a starting salary which is around **70% higher** than those of the schools below rank 100.



* **Categorical dependent variables**

**glm**(formula, **family = binomial(link = "logit")**, data = mydata) **or**

**glm**(formula, **family = binomial(link = "probit")**, data = mydata)

# Your y variable is binary 0 or 1

(1)mylogit <- **glm**(y ~ x1 + x2 + x3, family = **binomial(link = "logit")**,

data = mydata)

or

(2)myprobit <- glm(y ~ x1 + x2 + x3, family = binomial(link = "probit"),

data = mydata)