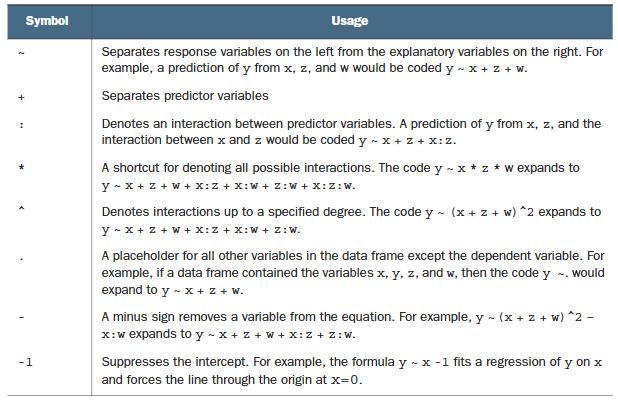
**Day 3 : Statistical analysis using R**

**Session 9: Regression**

* 1. **Fitting regression models with lm()**

Symbols commonly used in R formulas



A close up of text

Description automatically generated

|  |
| --- |
| 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)  A screenshot of a computer code  Description automatically generated  # visualizing simple regression  library(ggplot2)  ggplot(data = df, aes(x=study\_hours, y=score)) +  geom\_point() +  geom\_smooth(method = 'lm') +  theme\_bw() |

* 1. **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)    # 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]    #---------------------------------------------------------------------------  # 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 |

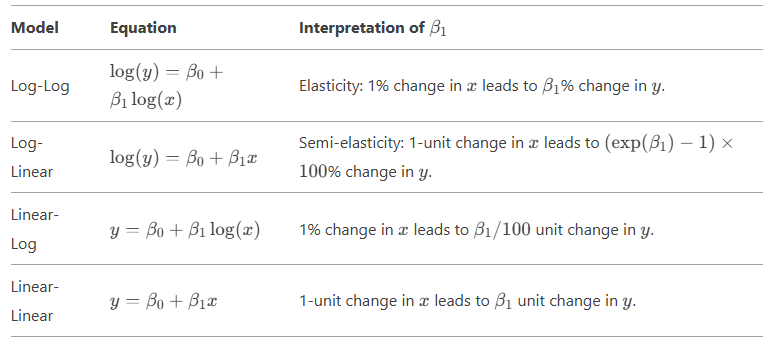
* 1. **Polynomial regression**

|  |
| --- |
| 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    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    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() |

* 1. **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)    #OR  fit <- lm(mpg ~ hp + wt + hp:wt, data=mtcars)  summary(fit)    #\* -------------------------------------------------------------  #\* 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 |

* 1. **Logarithmic regression**



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

**Session 10: Logistic regression**

* 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)    # Calculating marginal effects for logistic regression  logit\_margins <- margins(logit\_model)  summary(logit\_margins)    #-----------------------------------------------  # Running probit regression  #-----------------------------------------------  probit\_model <- glm(hhsex ~ hh\_size + urb\_rur + province,  data = data, family = binomial(link = "probit"))  summary(probit\_model)    # Calculating marginal effects for probit regression  probit\_margins <- margins(probit\_model)  summary(probit\_margins) |

**Task 8:**

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

1. Load the **011-Affairs.RData**
2. Tabulate the frequency of **affairs** variable from **Affairs** dataframe.
3. 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.
4. Set **ynaffairs** and **rating** variables as factor variables.
5. Set ‘0’ as reference for **ynaffairs** variable, ‘5’ for **rating**, ‘no’ for **children**, and ‘female’ for **gender** variables.
6. Fit a logistic regression model with **ynaffairs** as dependent variable and **gender, age, yearsmarried, children, rating** as independent variable.
7. Calculate average marginal effect for each variables using the margins() function.

**library(dplyr)**

**library(margins)**

**load('data/011-Affairs.RData')**

**table(Affairs$affairs)**

****

**Affairs <- Affairs %>% mutate(ynaffair = case\_when(affairs > 0 ~ 1, TRUE ~ 0),**

**ynaffair = factor(ynaffair),**

**rating = factor(rating))**

**table(Affairs$ynaffair)**

****

**#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**

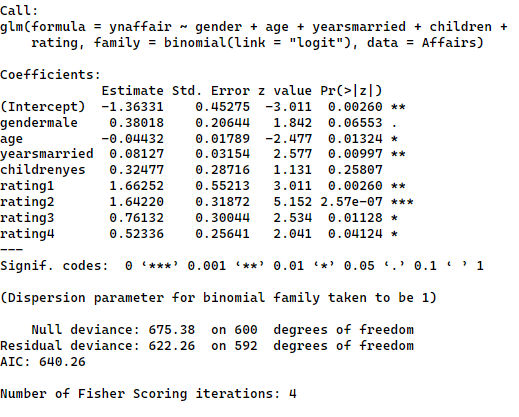
**+ children**

**+ rating,**

**data=Affairs,**

**family = binomial(link = "logit"))**

**summary(fit)**

****

**summary(margins(fit))**

**A table of numbers and letters

Description automatically generated**

**Session 11: Time-series analysis**

* 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.
  1. **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$rgdpe\_npl)    adf.test(df\_npl\_usa$rgdpe\_usa)    #Running a regression (Spurious regression observed)  fit <- lm(formula = rgdpe\_usa ~ rgdpe\_npl ,data = df\_npl\_usa)  summary(fit)    #-------------------------------------------------------------  # 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$dlrgdpe\_npl)    adf.test(df\_npl\_usa$dlrgdpe\_usa)    #Running a regression (no spurious regression observed)  fit <- lm(formula = dlrgdpe\_usa ~ dlrgdpe\_npl ,data = df\_npl\_usa)  summary(fit) |

* 1. **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$rgdpe)    adf.test(df$ccon)    #Running a regression  fit <- lm(formula = rgdpe ~ ccon ,data = df)  summary(fit)    #-------------------------------------------------------------  # 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$dlrgdpe\_npl)    adf.test(df\_npl\_usa$dlrgdpe\_usa)    #Running a regression  fit <- lm(formula = dlrgdpe ~ dlccon ,data = df)  summary(fit) |

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

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