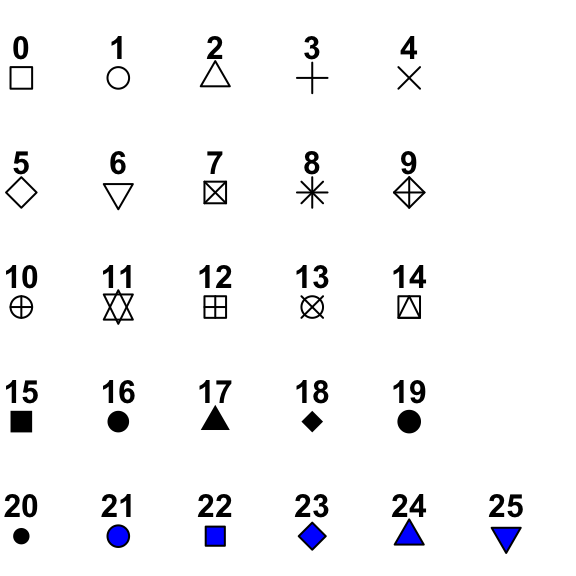
**Day 2 : Advance R programming**

**Session 5: Graphing**

* 1. **ggplot2 basics**

**shape**



**linetype**

**A screenshot of a computer

Description automatically generated**

|  |
| --- |
| E018-ggplot2\_basics.R |
| #---------------------------------------------------------------------------  # ggplot2 basics  #---------------------------------------------------------------------------  library(ggplot2)  library(mosaicData)  #simple scatter plot  ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point()    #scatter plot with various attributes  ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point(color = 'blue', shape = 16, alpha = 0.3, size = 2)    #applying ggplot2 themes  ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point(color = 'blue', shape = 16, alpha = 0.3, size = 2) +  theme\_bw()    #scatter plot and best fit line  ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point(color = 'blue', shape = 16, alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm') +  theme\_bw()    # grouping category variable attribute color  ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +  geom\_point(alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash') +  theme\_bw()    # x-axis and y-axis setting scales  ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +  geom\_point(alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash') +  scale\_x\_continuous(breaks = seq(0,70,10)) +  scale\_y\_continuous(breaks = seq(0,60,5)) +  theme\_bw()    # color manual of grouped category variable  ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +  geom\_point(alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash') +  scale\_x\_continuous(breaks = seq(0,70,10)) +  scale\_y\_continuous(breaks = seq(0,60,5)) +  scale\_color\_manual(values = c('darkgreen','red')) +  theme\_bw()    # subplots according to a categorical variable (facet\_wrap)  ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +  geom\_point(alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash') +  scale\_x\_continuous(breaks = seq(0,70,10)) +  scale\_y\_continuous(breaks = seq(0,60,5)) +  scale\_color\_manual(values = c('darkgreen','red')) +  facet\_wrap(~sector) +  theme\_bw()    # subplots according to a categorical variable (facet\_grid)  ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +  geom\_point(alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash') +  scale\_x\_continuous(breaks = seq(0,70,10)) +  scale\_y\_continuous(breaks = seq(0,60,5)) +  scale\_color\_manual(values = c('darkgreen','red')) +  facet\_grid(union ~ married) +  theme\_bw()    # labels  ggplot(data = CPS85, mapping = aes(x = exper, y = wage, color = sex)) +  geom\_point(alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm', se = FALSE, size = 1.3, linetype = 'longdash') +  scale\_x\_continuous(breaks = seq(0,70,10)) +  scale\_y\_continuous(breaks = seq(0,60,5)) +  scale\_color\_manual(values = c('darkgreen','red'), labels = c('Male', 'Female')) +  facet\_grid(union ~ married) +  labs(title = 'Relationship between wages and experiences',  subtitle = 'Current Population Survey',  caption = "Source: http://mosaic-web.org",  x = "Years of Experience",  y = "Hourly Wage",  color = 'Gender') +  theme\_bw() |

**Mapping in individual geom functions, stroing the plot as an R object, and exporting plot**

|  |
| --- |
| E019-ggplot2\_further.R |
| #---------------------------------------------------------------------------  # ggplot2 further  #---------------------------------------------------------------------------  library(ggplot2)  library(mosaicData)  # mapping color = sex in geom\_point instead in ggplot function  ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +  geom\_smooth(method = 'lm') +  theme\_bw()    #mapping aes in geom\_point and geom\_smooth  ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +  geom\_smooth(mapping = aes(linetype = sex, color = sex), method = 'lm') +  theme\_bw()    #storing the plot in an object  myplot <- ggplot(data = CPS85, mapping = aes(x = exper, y = wage)) +  geom\_point(mapping = aes(color = sex), alpha = 0.3, size = 2) +  geom\_smooth(mapping = aes(linetype = sex, color = sex), method = 'lm') +  theme\_bw()  myplot  #Saving the plot  ggsave(filename = 'myplot.jpg')  ggsave(filename = 'myplot.pdf')  ggsave(filename = 'myplot.png')  ggsave(filename = 'myplot.jpg', plot = myplot, units = 'cm', width = 20, height = 16)  ggsave(filename = 'myplot.pdf', plot = myplot, units = 'cm', width = 20, height = 16)  ggsave(filename = 'myplot.png', plot = myplot, units = 'cm', width = 20, height = 16) |

* 1. **Various types of plots**

**Bar charts**

|  |
| --- |
| E020-bar\_chart.R |
| library(ggplot2)  data(Arthritis, package="vcd")  #simple bar chart  table(Arthritis$Improved)    ggplot(Arthritis, aes(x=Improved)) + geom\_bar() +  labs(title="Simple Bar chart",  x="Improvement",  y="Frequency") +  theme\_bw()    #Horizontal bar chart  ggplot(Arthritis, aes(x=Improved)) + geom\_bar() +  labs(title="Simple Bar chart",  x="Improvement",  y="Frequency") +  coord\_flip()+  theme\_bw()    #Stacked bar chart  table(Arthritis$Improved, Arthritis$Treatment)    ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +  geom\_bar(position = 'stack') +  labs(title="Simple Bar chart",  x="Improvement",  y="Frequency") +  theme\_bw()    #Grouped bar chart  ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +  geom\_bar(position = 'dodge') +  labs(title="Simple Bar chart",  x="Improvement",  y="Frequency") +  theme\_bw()    #Filled bar chart  ggplot(Arthritis, aes(x=Treatment, fill = Improved)) +  geom\_bar(position = 'fill') +  labs(title="Simple Bar chart",  x="Improvement",  y="Proportion") +  theme\_bw()    #managing congested labels  ggplot(mpg, aes(x=model)) +  geom\_bar() #produce congested labels    ggplot(mpg, aes(x=model)) +  geom\_bar() +  theme(axis.text.x = element\_text(angle = 90, hjust = 1))    #OR  ggplot(mpg, aes(x=model)) +  geom\_bar() +  coord\_flip() |

**Pie charts**

|  |
| --- |
| E021-pie\_chart.R |
| if(!require(remotes)) install.packages("remotes")  remotes::install\_github("rkabacoff/ggpie")  library(ggplot2)  library(ggpie)  #simple pie chart  ggpie(mpg, class)    # no legend and offset of labels from the pie chart  ggpie(mpg, class, legend=FALSE, offset=1.3,  title="Automobiles by Car Class")    # group wise pie charts  ggpie(mpg, class, year,  legend=FALSE, offset=1.3, title="Car Class by Year") |

**Histograms**

|  |
| --- |
| E022-histogram.R |
| library(ggplot2)  library(dplyr)  data(mpg)  #Simple histogram  ggplot(mpg, aes(x=cty)) +  geom\_histogram() +  theme\_bw()    #Colored histogram with 20 bins  ggplot(mpg, aes(x=hwy)) +  geom\_histogram(bins=20, fill="red", color = 'black') +  theme\_bw()    #Histogram with density cruve  ggplot(mpg, aes(x=hwy, y = ..density..)) +  geom\_histogram(bins=20, fill="red", color = 'black') +  geom\_density(color = 'blue', size = 1.5) +  theme\_bw() |

**Box plots**

A box-and-whiskers plot describes the distribution of a continuous variable by plotting its five-number summary: the minimum, lower quartile (25th percentile), median (50th percentile), upper quartile (75th percentile), and maximum. It can also display observations that may be outliers (values outside the range of Q3 + 1.5 × IQR to Q1 - 1.5 × IQR, where IQR is the interquartile range (Q3 – Q1) defined as the upper quartile minus the lower quartile).

A diagram of a diagram

Description automatically generated

|  |
| --- |
| E023-boxplot.R |
| library(ggplot2)  ggplot(mpg, aes(x="", y=cty)) +  geom\_boxplot() +  theme\_bw()    ggplot(mpg, aes(x=factor(cyl), y=cty, fill=factor(year))) +  geom\_boxplot() +  scale\_fill\_manual(values=c("gold", "green")) +  labs(x="Number of Cylinders",  y="Miles Per Gallon",  title="City Mileage by # Cylinders and Year",  fill = "year") |

**Line plots**

|  |
| --- |
| E024-line\_plot.R |
| library(ggplot2)  library(dplyr)  wb\_energy <- read.csv('data/006-wb\_energy.csv')  df <- wb\_energy %>% filter(country %in% c('Nepal', 'India', 'Bangladesh', 'Pakistan'))  ggplot(data = df, mapping = aes(x = year, y = ele\_total, color = country, linetype = country )) +  geom\_line(size = 1.3) +  labs(y = '% of population with access to electricity') +  theme\_bw() +  theme(legend.position = 'bottom') +  scale\_x\_continuous(breaks = seq(1990,2020,2)) +  scale\_y\_continuous(breaks = seq(0,100,10)) |

**Session 6: R programming**

* 1. **Conditional execution**

|  |
| --- |
| E025-conditional\_execution.R |
| age <- 61  if (age <= 20) {  print('Teen')  } else if (age <=60) {  print('Adult')  } else {  print('Old')  } |

* 1. **User-written functions**

|  |
| --- |
| E026-user\_written\_function.R |
| age\_classify <- function(age) {  if (age <= 20) {  age\_type <- 'Teen'  } else if (age <=60) {  age\_type <- 'Adult'  } else {  age\_type <- 'Old'  }  return(age\_type)  }  age\_classify(15)  age\_classify(35)  age\_classify(85) |

* 1. **Looping**

|  |
| --- |
| E027-looping.R |
| #-------------------------------------  # For loop  #-------------------------------------  #finding the sum of squares of 1,2,3,4,5  x <- 0  for (i in c(1,2,3,4,5)) {  x <- x + i^2  }  print(x)  #finding the sum of 1 to 100  x <- 0  for (i in 1:100) {  x <- x + i  }  print(x)  #finding the sum of odd numbers from 1 to 100  x <- 0  for (i in 1:100) {  if (i %% 2 == 1) {  x <- x + i  }  }  print(x)  #-------------------------------------  # While loop  #-------------------------------------  #finding the sum of 1 to 100  x <- 0  i <- 0  while (i <= 100) {  x <- x + i  i <- i + 1  }  print(x)  #finding the sum of odd numbers from 1 to 100  x <- 0  i <- 0  while (i <= 100) {  if (i %% 2 == 1) {  x <- x + i  }  i <- i + 1  }  print(x) |

**Task 5:**

Suppose there is no built-in function in R to calculate mean and standard deviation. Write a user defined functions ***func\_mean*** and ***func\_sd*** to calculate mean and standard deviation of a given vector.

**vec <- c(3,5,2,3,4,2,5,6,7)**

**mean(vec) # 4.111111**

**sd(vec) # 1.763834**

**func\_mean <- function(vv) {**

**x <- 0**

**count <- 0**

**for (i in vv) {**

**x <- x + i**

**count <- count + 1**

**}**

**x\_bar <- x/count**

**return(x\_bar)**

**}**

**func\_sd <- function(vv) {**

**x\_bar <- func\_mean(vv)**

**x <- 0**

**count <- 0**

**for (i in vv) {**

**x <- x + (x\_bar - i)^2**

**count <- count + 1**

**}**

**x\_sd <- (x/(count-1))^(1/2)**

**return(x\_sd)**

**}**

**func\_mean(vec)**

**func\_sd(vec)**

**Session 7: Webscrapping, duplicates, and missing data**

* 1. **Webscrapping**

|  |
| --- |
| E028-webscrapping.R |
| #-----------------------------------------------------------------  # importing csv data directly from the web  #-----------------------------------------------------------------  df <- read.csv("http://s.anilz.net/wb\_energy")  head(df)  dx <- read.csv("https://data.ny.gov/api/views/d6yy-54nr/rows.csv")  head(dx)  #-----------------------------------------------------------------  # Using rvest package for static website scraping  #-----------------------------------------------------------------  #loading necessary packages  library(rvest) #see https://rvest.tidyverse.org/articles/harvesting-the-web.html for details  library(dplyr)  #loading webpage content  webpage <- read\_html("https://www.sharesansar.com/today-share-price")  #extracting table from the webpage  tables <- html\_table(webpage)  #checking the number of tables available in the webpage  length(tables)  df1 <- tables[[1]]  head(df1)  #filtering upper and lower circuit stock  filtered\_df1 <- df1 %>% filter(`Diff %` > 9 | `Diff %`< -9) %>% arrange(`Diff %`)  filtered\_df1  #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #\*Obtaining Forex information from NRB  #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #loading webpage content  webpage <- read\_html("https://www.nrb.org.np")  #extracting table from the webpage  tables <- html\_table(webpage)  #checking the number of tables available in the webpage  length(tables)  df1 <- tables[[1]]  df2 <- tables[[2]]  df1  df2  #keeping USD and JPY only  filtered\_df1 <- df1 %>% filter(Currency=='USD' | Currency =='JPY')  filtered\_df1 |

**Task 6 :**

Web-scrape the Historical ranking table from <https://en.wikipedia.org/wiki/ICC_Men%27s_T20I_Team_Rankings>

**webpage <- read\_html("https://en.wikipedia.org/wiki/ICC\_Men%27s\_T20I\_Team\_Rankings")**

**tables <- html\_table(webpage)**

**length(tables)**

**df <- tables[[7]]**

* 1. **Finding duplicates**

|  |
| --- |
| E029-duplicates.R |
| library(dplyr)  # Creating a sample data frame  df <- data.frame(  ID = c(1, 2, 3, 3, 4, 5, 4, 3),  Name = c("John", "Jane", "Mark", "Mark", "Luke", "Kate", "Luke", "Mark"),  Age = c(25, 30, 35, 35, 40, 45, 40, 35)  )  df    #showing the duplicated observations  df %>% filter(duplicated(.) == T)    #removing the duplicated observations  df %>% filter(!duplicated(.) == T)    #counting the duplicate observations  df %>% group\_by\_all() %>%  summarise(count = n()) %>%  filter(count > 1)    #identifying duplicate values based on some variables  library(haven)  df <- read\_dta('data/008-nlfs2.dta')  #checking whether there is any duplicate based on selected variables  df %>% count(psu, hhid) %>% filter(n > 1)    #selecting observations that does not have duplicate based on selected variables  df %>% filter(duplicated(psu, hhid) == F) %>% count(psu, hhid) |

* 1. **Finding missing values**

|  |
| --- |
| E030-missing\_values.R |
| library(haven)  library(dplyr)  df <- read\_dta('data/008-nlfs2.dta')  df <- df[c('psu','hhid','q13','q18')]  #Selecting observations with no missing values  df %>% filter(complete.cases(.) == T)  #Selecting observations with missing values  df %>% filter(complete.cases(.) == F) |

**Session 8: Descriptive statistics and hypothesis testing**

* 1. **Descriptive statistics**

|  |
| --- |
| E031-descriptive\_statistics.R |
| data(mtcars) #from datasets package  vars <- c('mpg','cyl','disp')  #basic descriptive statistics from base package  summary(mtcars[vars])    #descriptive statistics from other packages  Hmisc::describe(mtcars[vars])    pastecs::stat.desc(mtcars[vars])    psych::describe(mtcars[vars])    #----------------------------------  # Descriptive statistics by group  #----------------------------------  #grouping by one variable  by(mtcars[vars], #dataset  list(Transmission = mtcars$am), #grouping variable: Transmission (0 = automatic, 1 = manual)  summary) #function    #grouping by multiple variables  by(mtcars[vars],  list(Transmission = mtcars$am, Engine = mtcars$vs), # Transmission (0 = automatic, 1 = manual), Engine (0 = V-shaped, 1 = straight)  summary) |

* 1. **Descriptive statistics**

|  |
| --- |
| E032-frequency\_contingency\_tables.R |
| Arthritis <- vcd::Arthritis  #simple frequency table  mytable <- table(Arthritis$Improved)  mytable    #proportion table  prop.table(mytable)    prop.table(mytable)\*100 #in percentage    #-------------------------------------  # Two-way table  #-------------------------------------  mytable <- xtabs(~ Treatment + Improved, data=Arthritis)  mytable    #calculating sub-total horizontally  margin.table(mytable, 1) # 1 here refers 1st variable i.e. Treatment    #proportion table based on horizontal sub-total  prop.table(mytable, 1) \* 100    # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #calculating sub-total vertically  margin.table(mytable, 2) # 2 here refers 2nd variable i.e. Improved    #proportion table based on vertical sub-total  prop.table(mytable, 2) \* 100    #------------------------------------------------  # Two-way table (add sub-totals and grand totals)  #------------------------------------------------  addmargins(mytable)    addmargins(prop.table(mytable)) \* 100    #proportion addmargins horizontally  addmargins(prop.table(mytable, 1), 2) \* 100    #proportion addmargins vertically  addmargins(prop.table(mytable, 2), 1) \* 100    #------------------------------------------------  # Multidimensional table  #------------------------------------------------  mytable <- xtabs(~ Treatment + Sex + Improved, data=Arthritis)  mytable    #frequency table  ftable(mytable)    #frequency table defining column variables  ftable(mytable, col.vars = c('Sex','Improved'))    #proportion table  prop.table(ftable(mytable, col.vars = c('Sex','Improved'))) \* 100 |

* 1. **The concept of normal distribution**

1. **What is a Normal Distribution?**

* **Shape:** The normal distribution looks like a bell-shaped curve.
* **Symmetry:** It is perfectly symmetrical around the center.

A diagram of a normal distribution

Description automatically generated

1. **Key Characteristics:**

* **Mean (Average):** The center of the curve.
* **Standard Deviation:** Measures the spread of the data.
  + 68.2% of the data falls within 1 standard deviation of the mean.
  + 95.4% falls within 2 standard deviations.
  + 99.7% falls within 3 standard deviations.

1. **Why is it Important?**

* **Natural Occurrences:** Many natural phenomena follow this distribution (e.g., heights, test scores). For example, most students score around the average in a class, fewer scoring very high or very low.
* **Central Limit Theorem:** In large samples, the samples’ mean tend to be normally distributed. ([Video](https://www.youtube.com/shorts/TwctT3Ncm1w))
* **Statistical Inferences:** Helps in making predictions and decisions based on data.
  1. **Hypothesis testing**

1. **What is Hypothesis Testing?**

* Hypothesis testing is a method used to decide whether there is enough evidence to support a particular claim about a population based on a sample of data.
* **Null Hypothesis (**H0**)**: This is the default statement that there is no effect or no difference. It assumes that any observed differences are due to random chance.

Example: "The average age is equal to 20."

* **Alternative Hypothesis (**H1**)**: This is what you want to prove, stating there is an effect or a difference.

Example: "The average age is not equal to 20."

1. **Procedure of hypothesis testing**

* State the null and alternative hypothesis. (e.g. , )
* Collect sample data.
* Calculate sample mean and stadard error ().
* Calculate t-statistics ( ).
* Compare absolute value of t-statistics |t| with critical values for given level of significance (). [1.65 (10% significance level), 1.96 (5%), 2.58 (1%)]

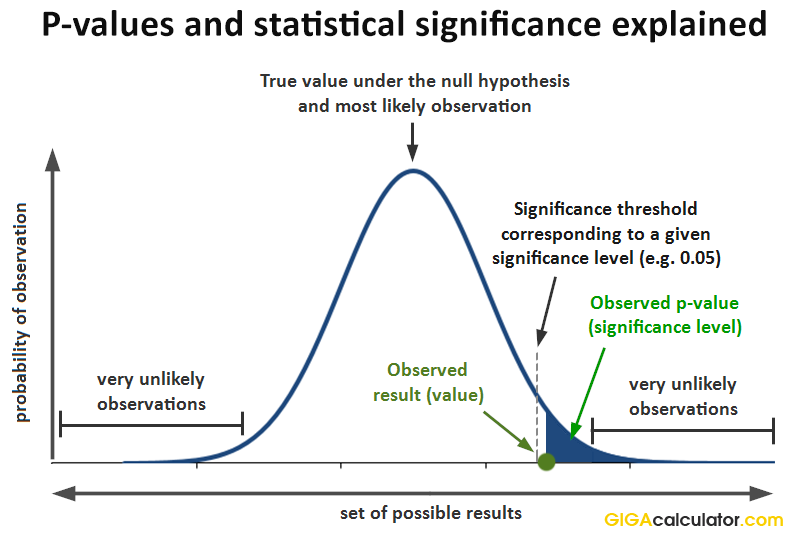
A diagram of a function

Description automatically generated

* Decision: reject null hypothesis if |t| exceeds critical value, otherwise fail to reject null hypothesis.

1. **Hypothesis testing with p-value**

* p-value : probability (area under normal distribution) beyond |t| i.e., probability of rejecting H0 when its true.



* **Decision :** reject null hypothesis if p-value is lower than the significance level, otherwise fail to reject null hypothesis.
* Easier to conduct hypothesis testing with p-value. No need to calculate t-statistics and remember different critical values.

|  |
| --- |
| E033-hypothesis\_testing.R |
| library(dplyr)  # Set seed for reproducibility  set.seed(12345)  # Create a dummy dataset  n <- 1000  group <- rep(0:1, length.out = n)  score <- 50 + group \* 10 + rnorm(n, mean = 0, sd = 10)  # Combine into a data frame  data <- data.frame(group = group, score = score)  # Conducting hypothesis testing (one-sample t-tests)  t.test(data$score, mu = 50) # H0: pop\_mean = 50    t.test(data$score, mu = 55) # H0: pop\_mean = 55    t.test(data$score, mu = 60) # H0: pop\_mean = 60    # --------------------------------------  # Conducting two-sample t-test  # --------------------------------------  data0 <- filter(data, group == 0)  data1 <- filter(data, group == 1)  t.test(data0$score, data1$score)    #OR  t.test(score ~ group, data = data) # H0: pop\_mean\_group1 = pop\_mean\_group2 |

**Task 7:**

Using NMICS6 data (009-hl.sav), conduct a hypothesis test whether average age between male and female is statistically different.

**# Load necessary libraries**

**library(haven) # For reading SPSS files**

**library(dplyr)**

**# Import SPSS file from the URL**

**df <- read\_spss('data/009-hl.sav')**

**# HL6 -> Age, HL4 -> Sex**

**# Summarize age for males (HL4 == 1) and females (HL4 == 2)**

**df\_male <- filter(df, HL4 == 1)**

**df\_female <- filter(df, HL4 == 2)**

**mean(df\_male$HL6)**

**mean(df\_female$HL6)**

**#OR**

**by(df$HL6, df$HL4, summary) #average age : male = 28.26, female = 28.83**

**# Conduct hypothesis testing (two-sample t-test)**

**t.test(HL6 ~ HL4, data = df)**

**A black text on a white background

Description automatically generated**

**# Alternatively, use linear regression**

**regression\_result <- lm(HL6 ~ HL4, data = df)**

**summary(regression\_result)**

**A screenshot of a computer code

Description automatically generated**