Week-3: Code-along

Insert your name here

2023-08-30

I. Code to edit and execute

To be submitted on canvas before attending the tutorial

Loading packages

```r
# Load package tidyverse
```

Assigning values to variables

```r
# Example a.: execute this example
x <- 'A'
x
```

```r
# Complete the code for Example b and execute it
```

```r
# Complete the code for Example c and execute it
```

```r
# Complete the code for Example d and execute it
```

```r
# Complete the code for Example e and execute it
```

```r
# Complete the code for Example f and execute it
```

Checking the type of variables

```r
# Example a.: execute this example
x <- 'A'
typeof(x)
```

```r
# Complete the code for Example b and execute it
# Complete the code for Example c and execute it
```

```r
# Complete the code for Example d and execute it
```

```r
# Complete the code for Example e and execute it
```

```r
# Complete the code for Example f and execute it
```

Need for data types

```r
# import the cat-lovers data from the csv file you downloaded from canvas
```

```r
# Compute the mean of the number of cats: execute this command
mean(cat_lovers$number_of_cats)
```

```r
# Get more information about the mean() command using ? operator
```

```r
# Convert the variable number_of_cats using as.integer()
```

```r
# Display the elements of the column number_of_cats
```

```r
# Display the elements of the column number_of_cats after converting it using as.numeric()
```

Create an empty vector

```r
# Empty vector

# Type of the empty vector
typeof(x)
```

Create vectors of type logical

```r
# Method 1
x<-vector("logical",length=5)
# Display the contents of x
print(x)
# Display the type of x
print(typeof(x))
```

```r
# Method 2
x<-logical(5)
# Display the contents of x
print(x)
# Display the type of x
print(typeof(x))
```

```r
# Method 3
x<-c(TRUE,FALSE,TRUE,FALSE,TRUE)
# Display the contents of x
print(x)
# Display the type of x
print(typeof(x))
```

Create vectors of type character

```r
# Method 1

# Display the contents of x

# Display the type of x
print(typeof(x))
```

```r
# Method 2

# Display the contents of x
print(x)
# Display the type of x
```

```r
# Method 3

# Display the contents of x

# Display the type of x
```

Create vectors of type integer

```r
# Method 1

# Display the contents of x

# Display the type of x
print(typeof(x))
```

```r
# Method 2

# Display the contents of x
print(x)
# Display the type of x
```

```r
# Method 3

# Display the contents of x

# Display the type of x
# Method 4

# Display the contents of x

# Display the type of x
```

```
# Method 5

# Display the contents of x

# Display the type of x
```

Create vectors of type double

```
# Method 1

# Display the contents of x

# Display the type of x
```

```
# Method 2

# Display the contents of x

# Display the type of x
```

```
# Method 3

# Display the contents of x

# Display the type of x
```

Implicit coercion

```


# Create a vector

# Check the type of x
```

```


# Add a character to the vector

# Check the type of x
```

Example 1

```
# Create a vector

# Check the type of x
```

```
# Add a number to the vector

# Check the type of x
```

Example 2

```
# Create a vector

# Check the type of x
```

```
# Add a logical value to the vector

# Check the type of x
```

Example 3

```
# Create a vector

# Check the type of x
```

```
# Add a number to the vector

# Check the type of x
```

Example 4

Explicit coercion

```
# Create a vector

# Check the type of x
# Convert the vector to type character

# Check the type of x
```

Example 1

```
# Create a vector

# Check the type of x
```

```
# Convert the vector to type double

# Check the type of x
```

Example 2

Accessing elements of the vector

```
# Create a vector
x <- c(1,10,9,8,1,3,5)
```

```
# Access one element with index 3
```

```
# Access elements with consecutive indices, 2 to 4: 2,3,4
```

```
# Access elements with non-consecutive indices, 1,3,5
```

```
# Access elements using logical vector
x[c(TRUE,FALSE,FALSE,TRUE,FALSE,FALSE,TRUE)]
```

```
# Access elements using the conditional operator <
```

Examining vectors

```
# Display the length of the vector
print(length(x))
# Display the type of the vector
print(typeof(x))
# Display the structure of the vector
print(str(x))
```

Lists

```r
# Initialise a named list
my_pie = list(type="key lime", diameter=7, is.vegetarian=TRUE)
# display the list
my_pie
```

```r
# Print the names of the list
```

```r
# Retrieve the element named type
```

```r
# Retrieve a truncated list
```

```r
# Retrieve the element named type
```

```r
# Install package
install.packages("openintro")
# Load the package
library(openintro)
# Load package
library(tidyverse)
```

```r
# Catch a glimpse of the data-set: see how the rows are stacked one below another
glimpse(loans_full_schema)
```

```r
# Selecting numeric variables
loans <- loans_full_schema %>% # <-- pipe operator
  select(paid_total, term, interest_rate,
         annual_income,paid_late_fees,debt_to_income)
# View the columns stacked one below another
glimpse(loans)
```

```r
# Selecting categoric variables
loans <- loans_full_schema %>%
  select( ) # type the chosen columns as in the lecture slide
# View the columns stacked one below another
glimpse(loans)
```

Exploring data-sets