R cheat sheet W1-5

Tags	Example Tag
UID	202309181414
Created	@September 18, 2023 2:14 PM
Last Edited	@September 18, 2023 3:15 PM
Class tag	
Content tag	nm2207

GGplot

Plot the data

- ggplot(data=starwars)
- ggplot(data=starwars) +
  aes(x=height,y=mass)

adding the dots and relabel axis

- ggplot(data=starwars) + aes(x=height,y=mass) + geom_point() +
  labs(x="Height (cm)",y="Weight (Kg)")

to add a caption and title

- labs(x="Height (cm)",
  y="Weight (Kg)",
  title="Mass versus Height",
  caption="Source: tidyverse/ starwars dataset")

Shiny package

Run an example from the library

runExample("insert_example_name_here")

eg. . 07_widgets

Some operators

1. ? - operator to learn more about an command
2. $ - operator to extract a specific part of data
3. %>% - pipe operator that passes the output of one function as an input to another
4. %in% - operator to check if an element belongs to a data-set
5. library(package_name) - to load an add-on package
6. glimpe(dataset_name) - get a glimpse of your data
7. filter(row_attributes) - subset rows using column values
8. select(col_names) - subset columns using their names or types
9. pull(column_name) - extracts a single column
10. nrow/ncol/dim(dataset_name) - number of rows/columns/dimension of a data-set

Variables

continuous numeric

eg height, temperature

Discrete numeric

eg there is a limit to how many count of a variable

and whole numbers

Non-numeric

categorical

1. character: single or string of words or letters, MUST USE SINGLE QUOTES
2. logical : only true or false option

Ordinal

catergorical can be ordinal.

natural ordering eg like to super dislike, low, medium, high

nominal

no ordering, can be character or logical

eg male, female, blue, green, turtle

Binary

mutually exclusive, so like opposite ends

eg yes, no, 1,0

How to find

- typeof(x), where x is the character or object we want to find

Read Data set

cat_lovers <- read_csv("cat-lovers.csv")

cat_lovers$number_of_cats

Vectors

to make a vector logical or character:

logical: x<-c(TRUE,FALSE,TRUE,FALSE,TRUE). (5 things to show length of vector is 5)

character: x<-c('A','b','r','q')

integer vectors:

x<-c(1,2,3,4,5)

double vectors:

x<-c(1.787,0.63573,2.3890) or x<-double(5). the combined value is 5

Coercion

Implicit

x <- c(TRUE)

typeof(x)

we get logical

when we add 2,

x <- c(x,2)

we get double.

Explicit

x <- c(1L)

we get integer

x <- as.character(x)

its forced into a character vector.

x <- as.character(x)

x <- as.numeric(x)

x<-as.list(x)

To view a vector:

x <- c(1,10,9,8,1,3,5)

to see index number 3

x[3] , we get 9

x[2:4], we see 10,9,8

x[c(1,3,5)], we see the 1,9,1

L3,slide 35 we can also give the numbers TRUE or FALSE, then the numbers labeled TRUE will appear when we click enter

List

x<-list(1,"a",0.289,TRUE)

names(list)

list$type, to retrieve the info of the type of list it is

Data-set

glimpse(loans_full_schema)

select(dataset name, column or variable interested) arrange(dataset,column u want to arrange)

- arrange in increasing order by default.
- arrange(dataset, desc(column))

select + arrange

arrange(dataframe u want to select, how u want to arrange)

eg.

```
arrange( # <-- start with the verb
select(hotels, lead_time), # <-- first argument is the dataframe *
desc(lead_time) # <--- second argument is the how you want arrange
) # i.e. decreasing order of lead_time
```

alternative way:

```
hotels %>% # data frame
select(lead_time) %>% # what u want to arrange
arrange(desc(lead_time)) # how u arrange it
```

Filter

```
hotels %>% # dataset
filter(children >= 1) %>% # conditions of what u want to filter
select(hotel, children) # what column u want
```

- if there is more than 1 condition, just add it in filter argument

```
hotels %>%
filter(children >= 1,hotel == "City Hotel") %>%
select(hotel, children)
```

Slice

choose the rows u want to see

```
hotels %>% slice(1:5)
```

```
hotels %>%
slice(1,3,5) # specific rows not in order
```

```
hotels %>% distinct(hotel) # rows with a unique variable aka have a hotel in their row.
```

Mutate

```
hotels %>%
mutate(little_ones = children + babies) %>%
select(hotel, little_ones,children,babies)
```

data set, what the new row is made of, which rows to select to make new row.

Mutate and Filter

```
hotels %>%
mutate(little_ones = children + babies) %>%
filter(
little_ones >= 1,
hotel == "Resort Hotel"
) %>%
select(hotel, little_ones) # select the new column made
```

Functions

```
function_name <- function(arguments) {
body_of_the_function
return(output)
}
```

print a name:

```
name <- "Kashif"
print(paste0("Hello ", name, "!"))
```

```
"Hello Kashif!”
```

set defaults for arguments

```
calc_sample_mean <- function(sample_size,
our_mean=0,
our_sd=1) {
sample <- rnorm(sample_size,
mean = our_mean,
sd = our_sd)
mean(sample)
}
```

```
calc_sample_mean(sample_size = 10)
```

u cna change defaults in future arguments by refering to them.

eg . calc_sample_mean(10, our_sd = 2)

```
add_two <- function(x) {
x+2
}
```

when u set add_two to a number, x would be assigned that number and u get the output of 6 if add_two is 4

Scoping

global

```
global_var <- 10
```

```
my_function <- function() {
local_var <- 5
global_var <- 20  # This creates a new local variable 'global_var' inside the function.
return(global_var)
}
```

```
result <- my_function()
print(global_var)  # This will print the global variable value (10).
print(result)      # This will print the local variable value (20).
```

local

```
global_var <- 10
```

```
my_function <- function() {
local_var <- 5
return(local_var)
}

result <- my_function()
print(local_var)  # This will result in an error because 'local_var' is not accessible here.
print(result)     # This will print the value of 'local_var' (5).
```

Modify

```
global_var <- 10

my_function <- function() {
global_var <<- 20  # Modifies the global variable 'global_var'.
return(global_var)
}

result <- my_function()
print(global_var)  # This will print the modified global variable value (20).
print(result)      # This will also print the modified global variable value (20).
```

using <← operator

shadowing

create a local variable with global variable name

it will cause the global variable to be overshadowed, the global variable still exists, just hidden lol.