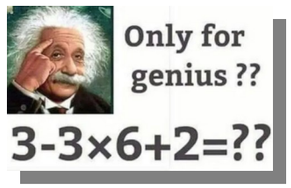
R⁴H₂O : : SESSION ONE CHEAT SHEET

Arithmetic

R is meme proof (applies BODMAS)

```
3 - 3 * 6 + 2
```



Variables

Use meaningful variable names

```
diameter <- 150
nonRevWater <- 10
non_rev_water <- 20
```

Vectors

Create vectors:

```
nonRevWater <- c(1, 3, -1, 0)
day <- 1:4
```

Vector subsets:

```
nonRevWater[2]
nonRevWater[2:3]
```

Math Functions

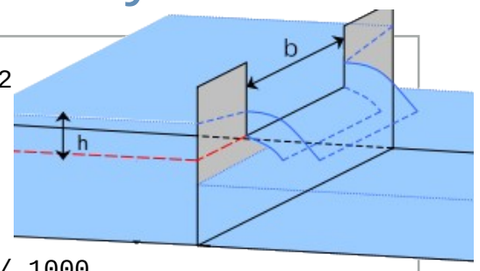
```
sqrt(nonRevWater)
sum(nonRevWater)
prod(nonRevWater)
factorial(nonRevWater)
abs(nonRevWater)
exp(nonRevWater)
log(nonRevWater, base = 10)
```

Extreme outcomes can be `NaN` (Not a Number) or `Inf` (approaching infinity).

Case Study 0

```
Cd <- 0.62
g <- 9.81
b <- 0.5

h <- 100 / 1000
q <- (2 / 3) * Cd * sqrt(2 * g) *
     b * h^(3 / 2)
```



Packages

Packages are libraries of functions and data files to extend R. The CRAN website lists most packages. Initiate libraries every script with:

```
library(tidyverse)
```

Tidyverse is a collection of packages. Click on the hexagonal package logos for more info.

Reading CSV Files

The **readr package** reads delimited files, such as CSV (Comma-Separated Values).

```
library(readr)
gormsey <-
   read_csv("casestudy1/gormsey.csv")
```

Reading Excel Files

The **readxl package** imports Excel files.

```
library(readxl)
read_excel("casestudy1/gormsey.xlsx",
           sheet = 2, skip = 3)
```

Data Frames

Rectangular data (called Tibble in Tidyverse) to store structured data.

Columns: variables

```
gormsey$Result
gormsey[, 6]
gormsey[, "Result"]
```

Rows: observations

```
gormsey[1:10, ]
gormsey[1:10, 4]
```

Counting Data

Count number of rows for each variable

```
length(gormsey$Measure)
unique(gormsey$Measure)
count(gormsey, Measure)
count(gormsey, Measure, Town)
```

Filtering Data

Filter extracts rows that meet logical criteria

Logical operators

==	equal to	|	'or'
!	not equal to	&	'and'

Use the stringr package for wild-cards.

```
turbidity <- filter(gormsey,
          Measure == "Turbidity" &
          Town != "Merton")
```

Statistics

```
turb_results <- turbidity$Result
mean(turb_results)
median(turb_results)
min(turb_results)
max(turb_results)
range(turb_results)
quantile(turb_results,
         probs = 0.95,
         method = 7)
var(turb_results)
sd(turb_results)
```

Grouping

Use **group_by** to create a grouped copy of a table by columns

The **summarise** function acts on each group

```
turb_gr <- group_by(turbidity,
                    Town)
summarise(turb_gr,
          avg = mean(Result),
          max = max(Result))
```

Finding Help

Use **help** function or **?** to read internal help.

Read the detailed cheat sheets on rstudio.com/resources/cheatsheets