R for MINNESOTA

Online @ github.com/tidy-mn/cheatsheets

Use this reference to read data, clean column names, make plots, process dates, filter and summarize data, join tables, and more.

PROJECT LAUNCH

New Project

File > New Project > New Directory

New R script

File > New File > R Script

Where am I?

```r
# Show current working directory
getwd()
# Set new working directory
setwd("C:/my-data-folder")
```

Install new packages

```r
install.packages("readr")
library(readr)
```

STORE VALUES

```r
# Use the left-arrow
age <- 7.2

# Text goes in quotes
porg <- "Sunshine"

# Multiple values go inside c()
droids <- c("BB8", "R2D2", "C-3PO")

# Copy an object
my_droids <- droids

# Avoid numbers, spaces, & symbols
3-droids* <- "error_invalid_name"
```

PLOTS

Scatterplot

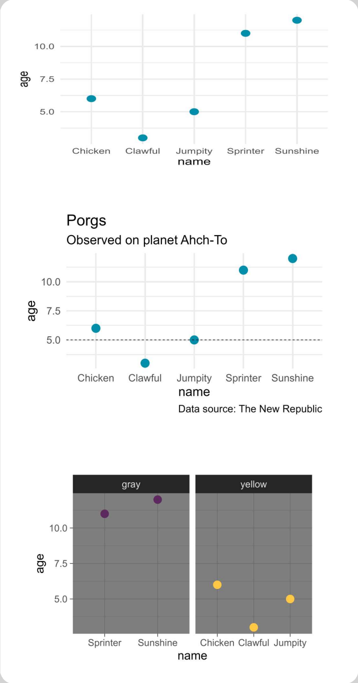
```r
library(ggplot2)

ggplot(porgs, aes(x = name, y = age)) +
  geom_point(size = 8, color = "hotpink")
```

Add titles & lines

```r
ggplot(porgs, aes(x = name, y = age)) +
  geom_point(size = 8, color = "hotpink") +
  geom_hline(yintercept = 5,
             linetype   = "dashed") +
  labs(title    = "Porgs",
       subtitle = "Sampled on planet Ahch-To",
       caption  = "data from New Republic")
```

Facet by group

```r
ggplot(porgs, aes(x = name, y = age)) +
  geom_point(aes(color = color), size = 8) +
  facet_wrap(~color) +
  scale_color_manual(values = c("gray", "yellow")) +
  theme_dark()
```



Porgs

Observed on planet Ahch-To

Data source: The New Republic

READ DATA

Text files (.csv, .txt, .tab)

```r
library(readr)
porgs <- read_csv("txt_file.csv")
```

Excel files (.xlsx, .xls)

```r
library(readxl)
porgs <- read_excel("Excel_file.xlsx")
```

CLEAN NAMES

```r
# Simplify all column names
library(janitor)
porgs <- clean_names(porgs)

# Assign new names manually
library(dplyr)
# Put new name on left: new_name = oldName
rename(porgs, mass_kg = massKG)
```

FILTER

```r
library(dplyr)
# Keep only Porgs older than 3
filter(porgs, age > 3)
# Keep rows with name Jumpity
filter(porgs, name == "Jumpity")
# Keep Porgs named Jumpity OR Chicken
filter(porgs, name %in% c("Jumpity", "Chicken"))
```

SUMMARIZE

```r
library(dplyr)
# Summarize the age for the entire table
summarize(porgs, avg_age = mean(age))
# Summarize the age for each color group
group_by(porgs, color) %>%
    summarize(avg_age = mean(age))
```

DESCRIBE DATA

```r
library(dplyr)
nrow(porgs)
names(porgs)
summary(porgs)
glimpse(porgs)
class(porgs)
# View unique column values
distinct(porgs, age)
## 5 6 11 12 3
```

ADD COLUMNS

```r
library(dplyr)

# Add home planet column
mutate(porgs,
       planet = "Earth")
# Add new calculated column
mutate(porgs,
       growth = height / age)
```

COMPARISONS

Symbol	Comparison
`>`	greater than
`>=`	greater than or equal to
`<`	less than
`<=`	less than or equal to
`==`	equal to
`!=`	NOT equal to
`%in%`	is value X in list: `X %in% c(1,3,5)`
`is.na(...)`	is the value missing?

R for MINNESOTA

Online @ github.com/tidy-mn/cheatsheets

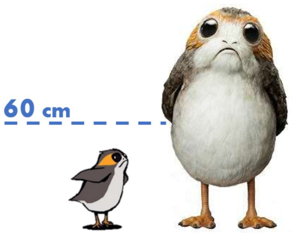
Remember porgs are not for dinner.

SELECT COLUMNS

```
# Keep only 2 columns
select(porgs, id, age)
# Drop the mass column
select(porgs, -mass)
# Put the age column first, but
# keep everything else the same
select(porgs, age, everything())
```

SORT ROWS

```
# Sort by age w/ YOUNGEST on top
arrange(porgs, age)
# Sort by age w/ ELDEST on top
arrange(porgs, desc(age))
# Sort by colors and then by age
arrange(porgs, color, desc(age))
```



IFELSE: YES / NO DECISIONS

Use `ifelse()` to create new values that depend on the value of another column. For example, to only label the porgs with a height over *60 cm* as "tall".

```
# When a porg's height is > 60 cm label it as "tall",
# otherwise label it as "short"
mutate(porgs, label = ifelse(height > 60, "tall", "short"))
```

DATES

Convert text to Date

Function	Order of Input	Output
`mdy()`	Month-Day-Year :: `05-18-2019`	`2019-05-18`

`mdy_hm()`	Month-Day-Year Hour:Minutes ::
	`05-18-2019 8:35`
		`2019-05-18`
		`08:35:00 UTC`

`mdy_hms()`	Month-Day-Year Hour:Mins:Secs ::
	`05-18-2019 8:35:22`
		`2019-05-18`
		`08:35:22 UTC`

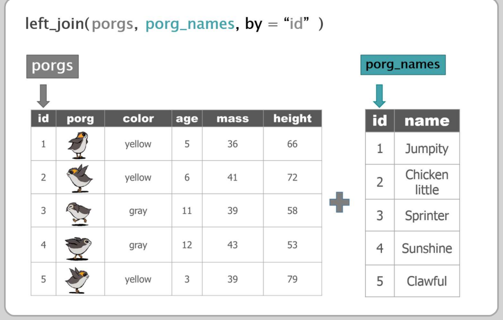
Date parts

Function	Date element
`year()`	Year
`month()`	Month as *1,2,3*
`day()`	Day of the month
`wday()`	Day of the week
`hour()`	Hour of the day *(24hr)*
`tz()`	Time zone

JOIN TABLES

`left_join()` keeps all rows and columns in the left table, and joins rows in the right table with matching IDs.

```
# Table w/ porg ages and heights
porgs

# Table w/ porg names
porg_names

# Join together by id columns
together <- left_join(porgs,
                      porg_names,
                      by = "id")
```



left_join(porgs, porg_names, by = "id")

porgs

id	porg	color	age	mass	height
1		yellow	5	36	66
2		yellow	6	41	72
3		gray	11	39	58
4		gray	12	43	53
5		yellow	3	39	79

porg_names

id	name
1	Jumpity
2	Chicken little
3	Sprinter
4	Sunshine
5	Clawful

SAVE DATA

Data tables

```
library(readr)
# Save data to a CSV text file
write_csv(porgs, "my_porg_data.csv")
```

Plots and images

```
library(ggsave)
# Save the last plot you made
ggsave("most_recent_plot.png")
# Save earlier plot stored to variable
best_plot <- ggplot()
ggsave(best_plot, "best_plot.png")
```

R COMMUNITY

- `#rstats` on 🐦
- ROpenSci.org
- Rweekly.org
- RLadies.org
- R-Bloggers.com
- TidyTuesdays
- **useR** Conferences

HELP!

Online

- Google: `r` or `rstats` + `"question"`
- Stackoverflow.com + `[r]` tag
- TEAMS Channel - Help requests!

From R

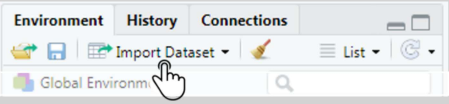
- Go to: `Help > Cheatsheets`
- Type `?` in the *Console*

```
# Function help
?read_csv
# Search help
help.search("boxplot")
```

SHORTCUTS

- Run line: `CTRL` + `ENTER`
- Save script: `CTRL` + `S`
- Tidy code: `highlight` + `CTRL` + `Shift` + `A`

Find data



R Support Crew Barbara.Monaco Kristie.Ellickson Dorian.Kvale Carl.Stenoien Andrea.Borich Derek.Nagel