Package Development Day 1

Joyce Robbins

What are packages?

- Collections of code

Where are they stored?

- CRAN
- GitHub
- BioConductor
- Your hard drive

How to install

- from CRAN:

```
install.packages("testthat")
```

Weird stuff

```
install.packages("RLadiesnyc")
```

- from GitHub:

```
devtools::install_github("jtr13/ggformat")
```

```
devtools::install_github("jtr13/ggformat", force = TRUE)
```

Many packages are available both on CRAN and GitHub:
"dev version"

```
devtools::install_github("tidyverse/forcats", force = TRUE)
```

- from your hard drive:

```
devtools::install("~/ggformat")
```

Regardless of the method, the packages are *installed* – see Packages –>

What is a "source" package?

```
fs::dir_tree("~/ggformat")
```

```
## ~/ggformat
## +-- DESCRIPTION
## +-- LICENSE
## +-- NAMESPACE
## +-- R
## |   \-- ggformat.R
## +-- Readme.Rmd
## +-- Readme.md
## +-- ggformat.Rproj
## +-- ggplot2template.png
## +-- inst
## |   \-- rstudio
## |       \-- addins.dcf
## +-- man
## |   \-- FormatCode.Rd
## \-- orderwords.txt
```

```
list.files("~/ggformat")
```

```
##  [1] "DESCRIPTION"         "ggformat.Rproj"      "ggplot2template.png"
##  [4] "inst"                "LICENSE"             "man"
##  [7] "NAMESPACE"           "orderwords.txt"      "R"
## [10] "Readme.md"           "Readme.Rmd"
```

Why use projects?

So you never have to use `getwd()` or `setwd()` again. You're always just in the right place.

Loading vs. installing

Usually we use `library()` to load packages into memory. While developing a package we use `devtools::load_all()`.

- It doesn't show up in the list of packages (or at least the version under development doesn't.)
- The functions are in memory and will disappear if we restart R.

Creating a package

We need some essential files: `DESCRIPTION` and a folder called `R` with `.R` files.

The easier way to create the structure is with `create_package()`:

```
library(devtools)
create_package("~/covidtime")
```

YOUR TURN

See Day1_lab.md

Using the package in another project / script

Before we can use the package elsewhere we need to:

- add function documentation by clicking "Code... Insert Roxygen Skeleton" (or at a minimum include `@export`)
- edit DESCRIPTION
- document with `devtools::document()`
- install the package with `devtools::install()` n

YOUR TURN

See Day1_lab.md

```
knitr::include_graphics("images/installation.png")
```