Drake in R

Matthew Pettis

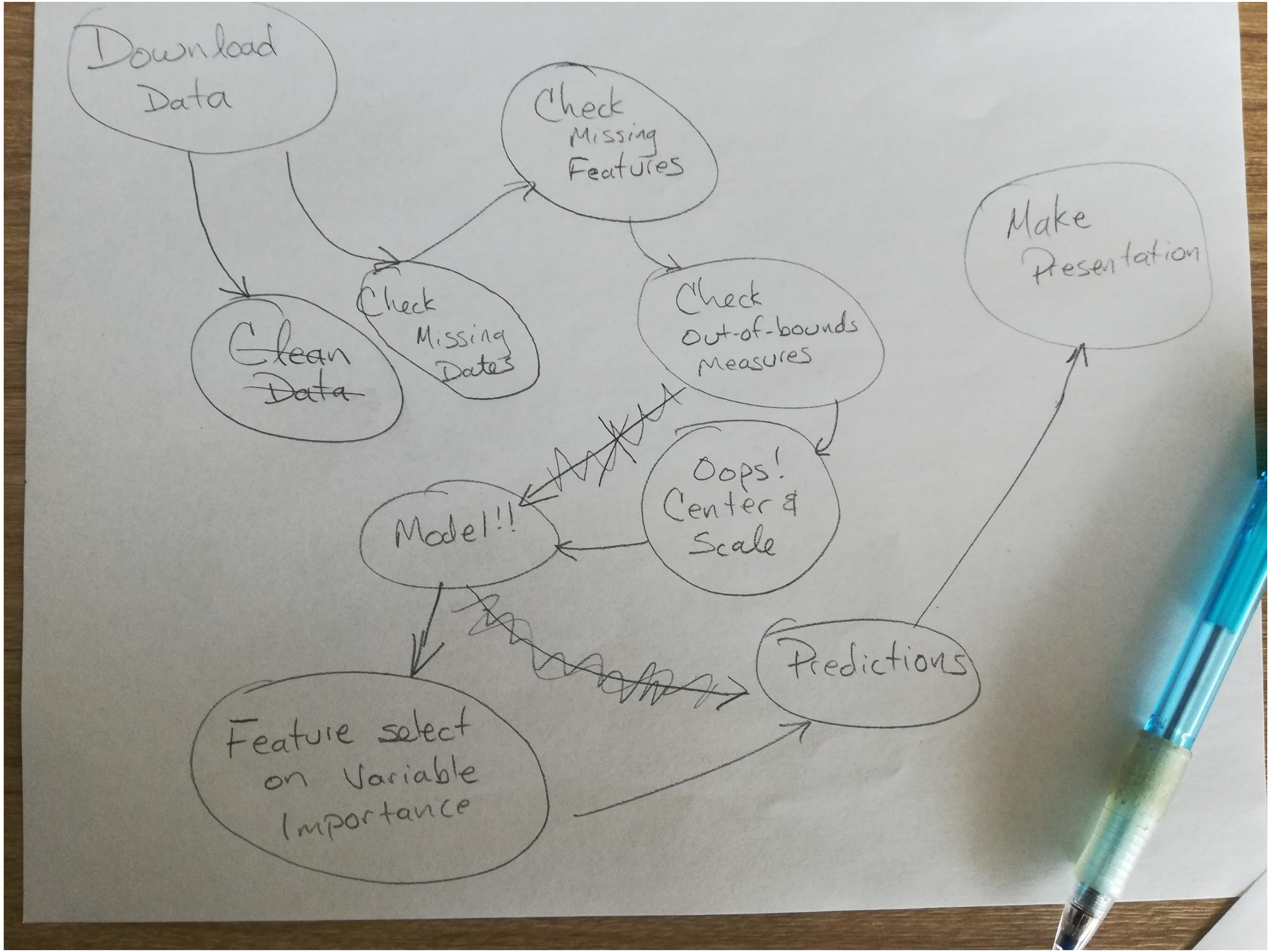
Who am I?

- Matt Pettis, Data Scientist at Trane
- Heavy R User

HELP!

CLEAN ALL THE THINGS!

Citytv


Download Data
Check Missing Features
~~Clean Data~~
Check Missing Dates
Check out-of-bounds Measures
Make Presentation
Oops! Center & Scale
Model!!
Predictions
Feature select on Variable Importance

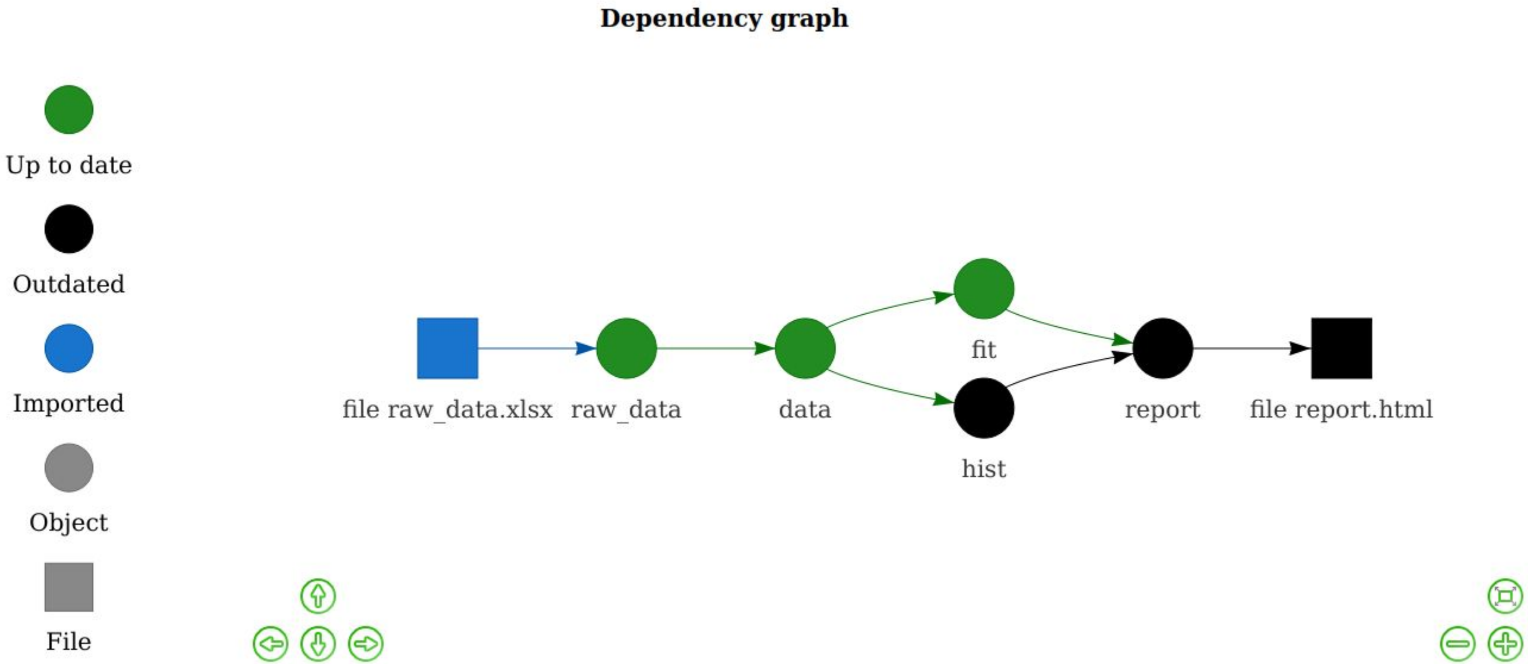
```
irinzn@SPA-TL-1YG9SN2 ~/Documents/personal/tcrug-talks/drake_2019-10-17/example-flow
$ ll
total 0
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 100-query-raw-data.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 200-check-missing-dates.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 210-check-missing-features.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 230-check-out-of-bounds.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 300-test-train-split.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 400-clean-impute.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 410-center-and-scale.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 500-feature-selection.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 600-model-creation.R
-rw-r--r--+ 1 irinzn Domain Users 0 Oct  9 13:58 610-model-evaluation.R
```

Rerun one file
Rerun everything
JAKE-CLARK.TUMBLR

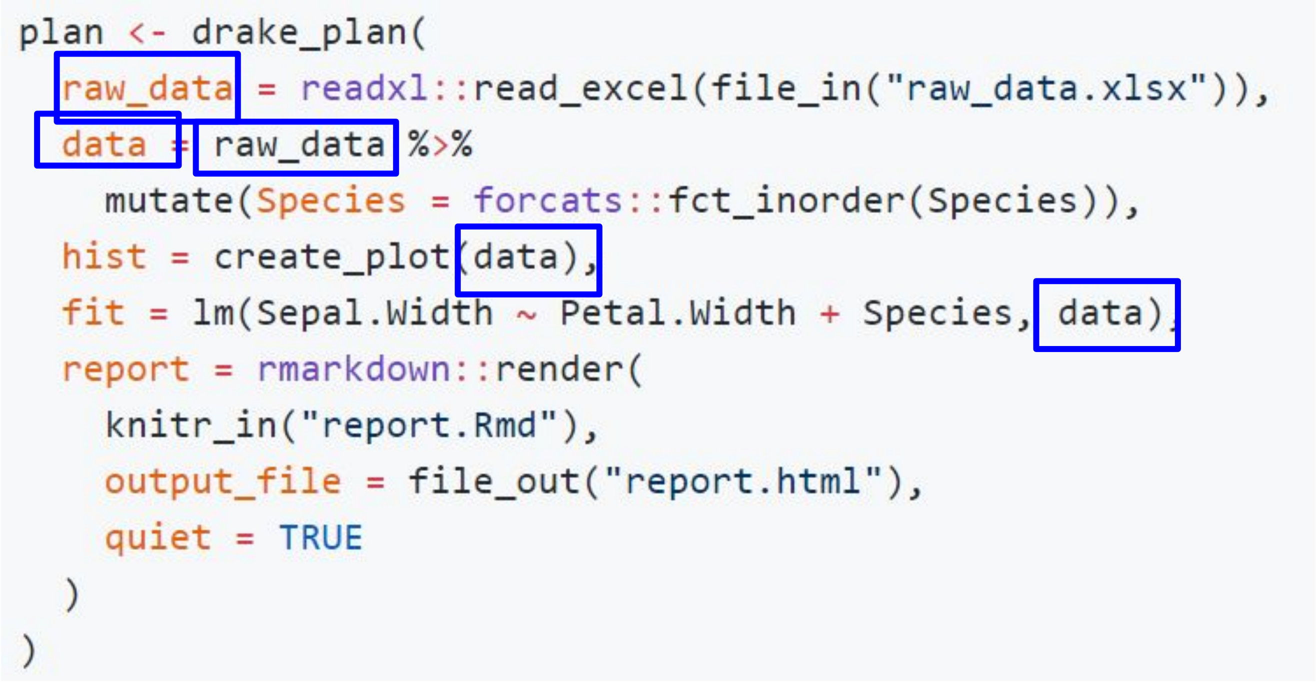
Full vs. Partial Re-runs

- Full runs are usually unnecessary and time-consuming.
- Sometimes they are infeasible.

Enter {drake}

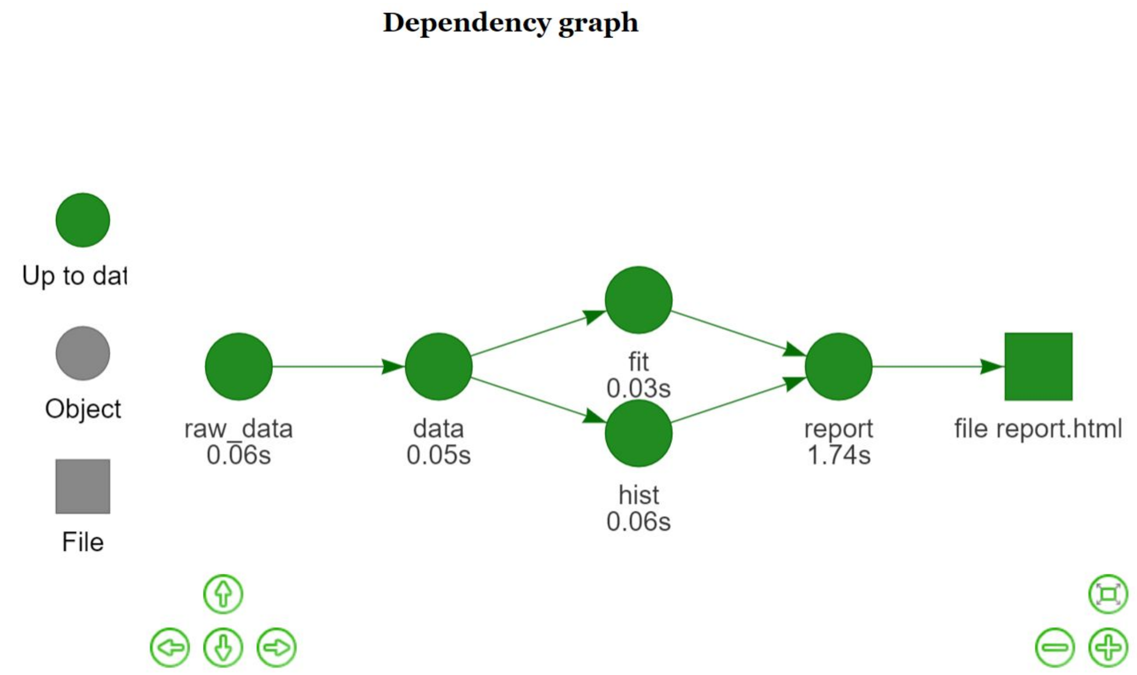

Dependency graph
Up to date
Outdated
Imported
Object
File
file raw_data.xlsx
raw_data
data
fit
hist
report
file report.html

Example

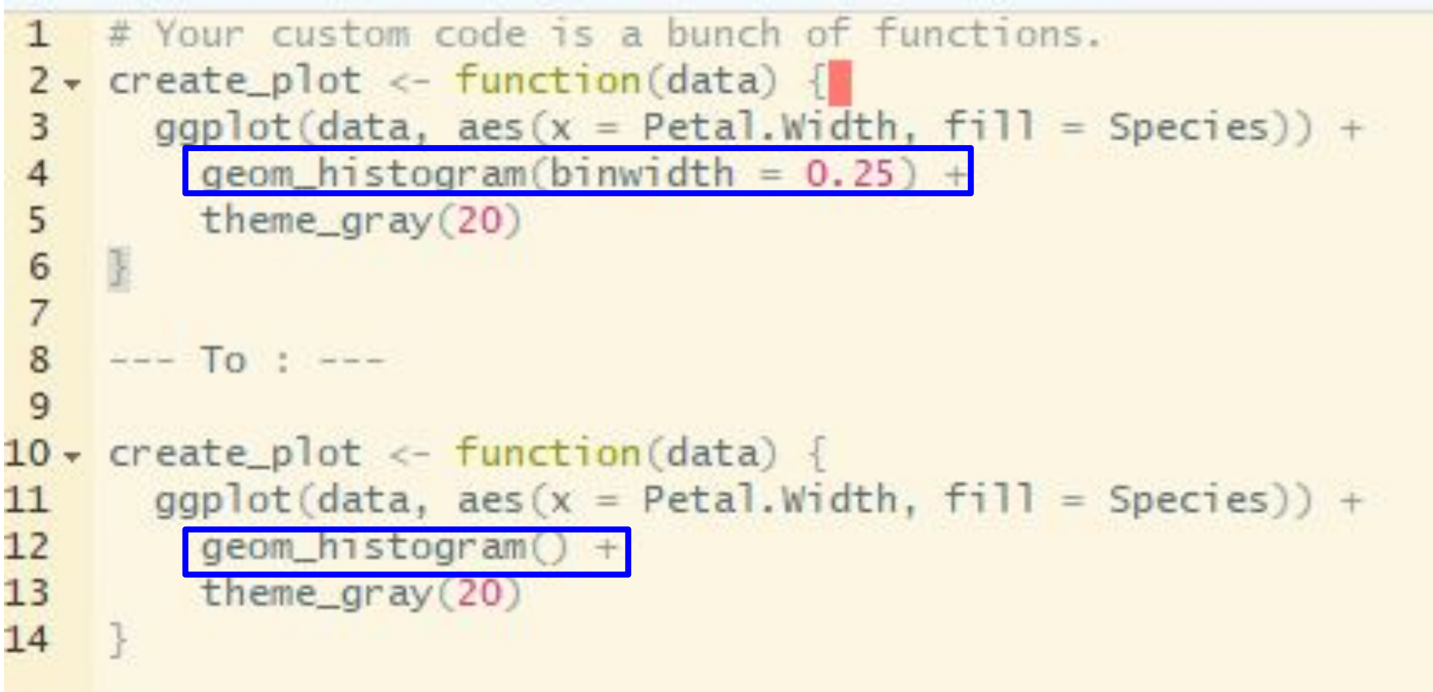


```
plan <- drake_plan(
  raw_data = readxl::read_excel(file_in("raw_data.xlsx")),
  data = raw_data %>%
    mutate(Species = forcats::fct_inorder(Species)),
  hist = create_plot(data),
  fit = lm(Sepal.Width ~ Petal.Width + Species, data),
  report = rmarkdown::render(
    knitr_in("report.Rmd"),
    output_file = file_out("report.html"),
    quiet = TRUE
  )
)
```

After drake build

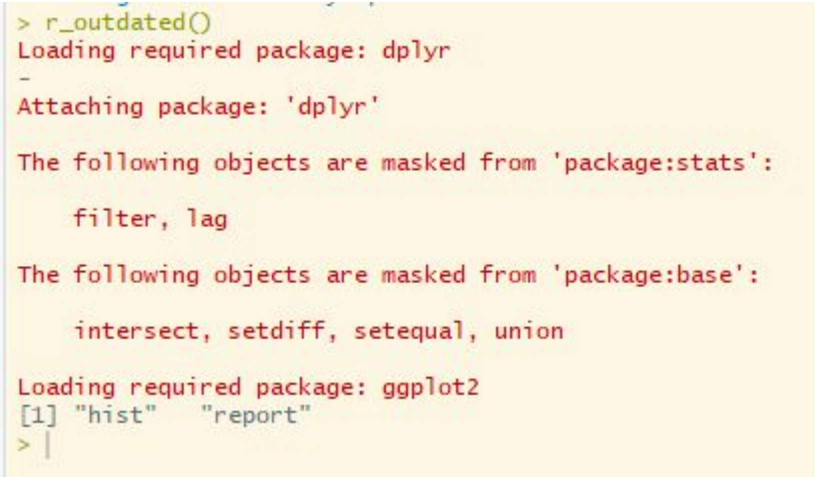


What happens when we alter a function



```
# Your custom code is a bunch of functions.
create_plot <- function(data) {
  ggplot(data, aes(x = Petal.Width, fill = Species)) +
    geom_histogram(binwidth = 0.25) +
    theme_gray(20)
}

--- To : ---

create_plot <- function(data) {
  ggplot(data, aes(x = Petal.Width, fill = Species)) +
    geom_histogram() +
    theme_gray(20)
}
```

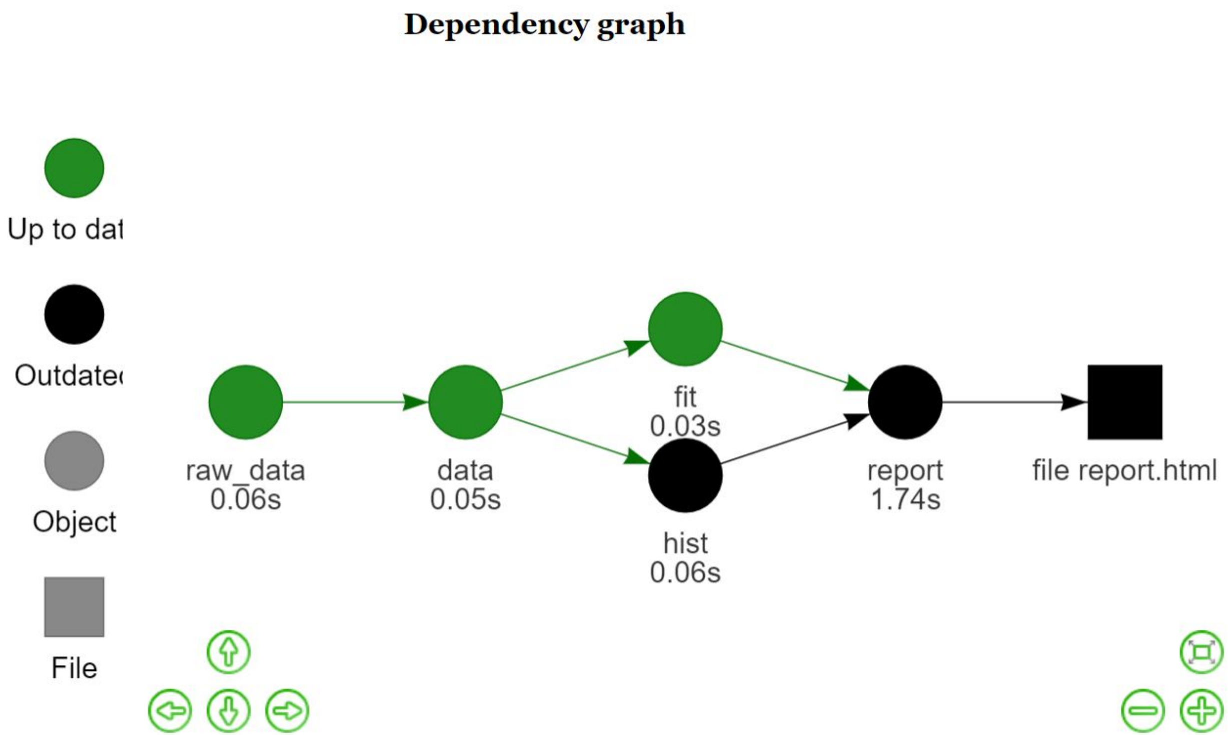
What happens when we alter a function?



```
> r_outdated()
Loading required package: dplyr
-
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Loading required package: ggplot2
[1] "hist"   "report"
> |
```

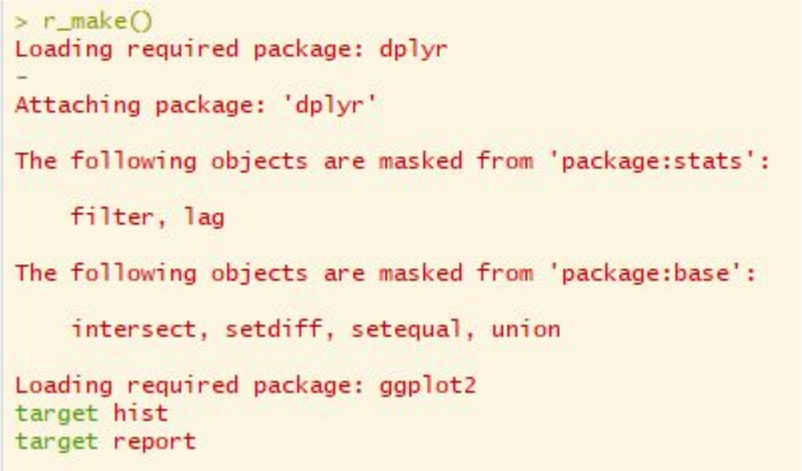
- Running `r_outdated` tells you that because of your change, the listed targets need to be rebuilt.

What happens when we alter a function?



- Black icons indicate the out-of-date objects that need to be re-created.

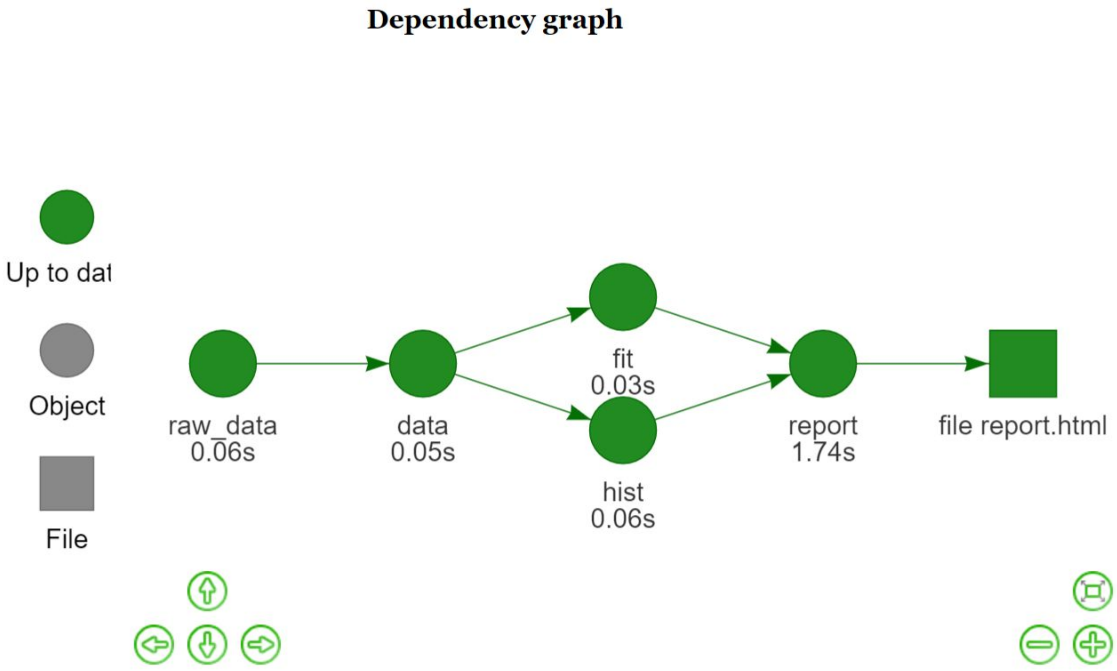
What happens when we alter a function?



```
> r_make()
Loading required package: dplyr
-
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Loading required package: ggplot2
target hist
target report
```

- Re-running `r_make()` runs *just* the necessary functions to propagate the changes.

What happens when we alter a function?



Dependency graph

- Graph is now back up to date.

Resources

- Git repo for drake project: https://github.com/ropensci/drake
- Drake manual: https://ropenscilabs.github.io/drake-manual/
- Similar piepline tools, many languages: https://github.com/pditommaso/awesome-pipeline
- Learn drake repo for self-tutorial: https://github.com/wlandau/learndrake
- Other presentations: https://ropenscilabs.github.io/drake-manual/index.html#presentations

Thank You

Matt Pettis

Email: matthew.pettis@gmail.com

Gitlab: mpettis

Twitter: @mtpettis