oRganization

How to make internal R packages a part of your team

Emily Riederer @emilyriederer emily.rbind.io

my
first
day

data access
server connection
proxies, ssh, ssl

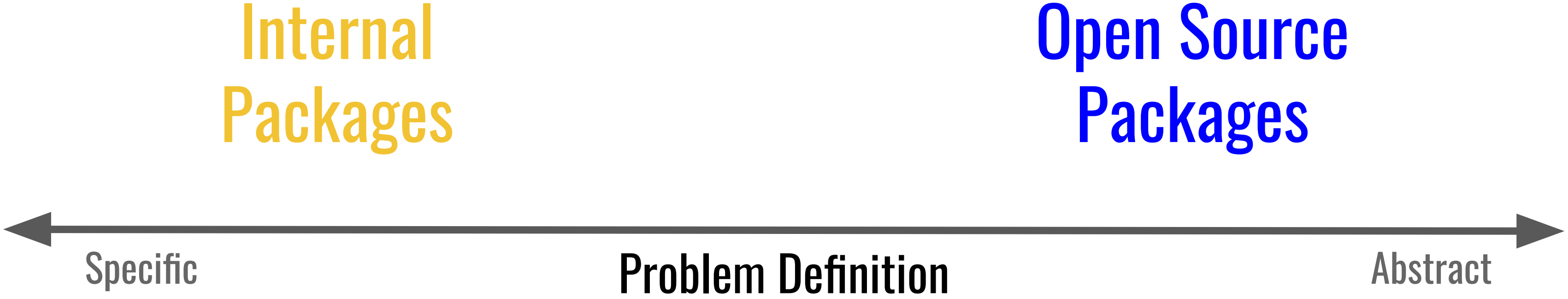
right problems
tribal knowledge
intuition

team norms
meetings
communication

data access
server connection
proxies, ssh, ssl

right problems
tribal knowledge
intuition

team norms
meetings
communication



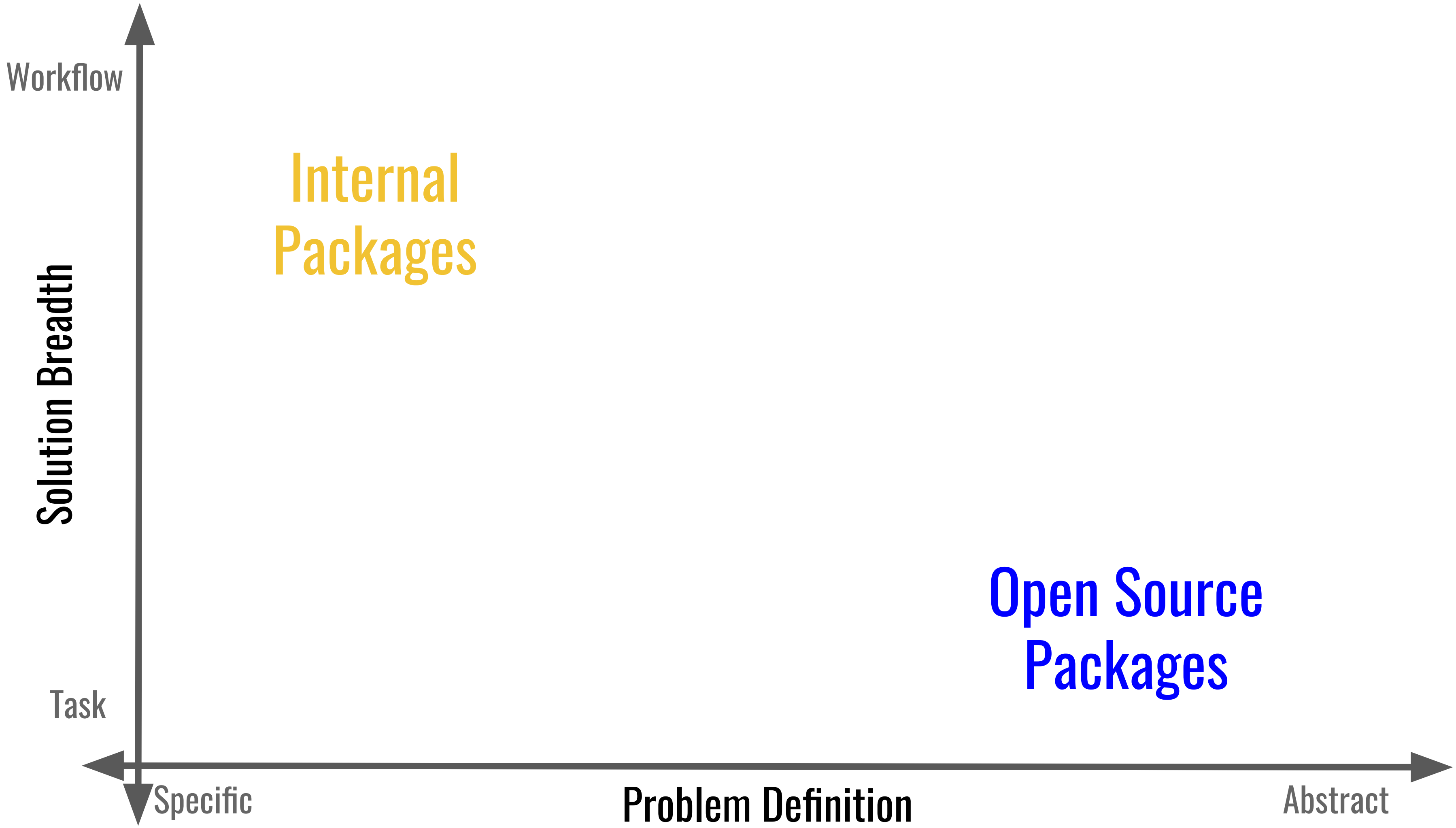
Internal Packages

Open Source Packages

Specific

Problem Definition

Abstract



Workflow

Solution Breadth

Task

Internal Packages

Open Source Packages

Specific

Problem Definition

Abstract

utilities packages

data access
server connection
proxies, ssh, ssl

e.g. abstraction layer for infrastructure

analysis packages

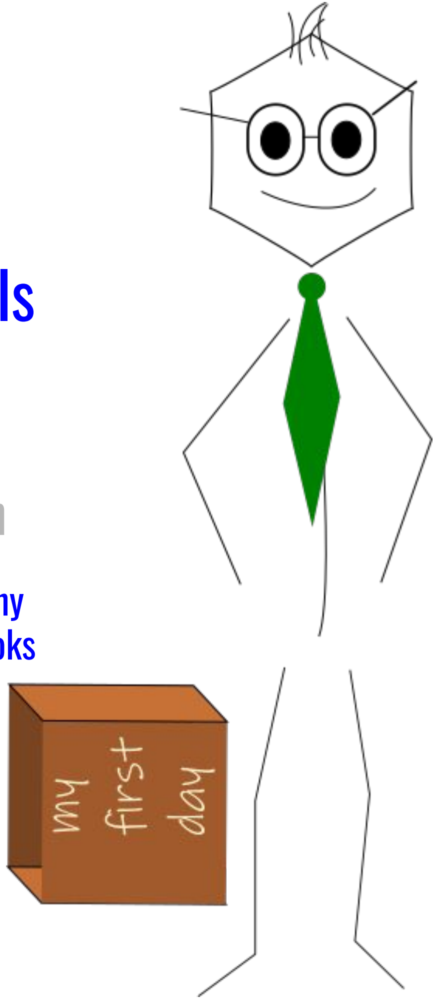
right problems
tribal knowledge
intuition

e.g. curated workflow, tailored function calls, automated result generation

developer tools

team norms
meetings
communication

e.g. color palettes, Shiny modules, linters, git hooks



Jobs-to-be-Done

We

hire a product

to do a

job

that helps us make

progress

towards a goal

Jobs-to-be-Done

We

hire a product

to do a

job

that helps us make

progress

towards a goal

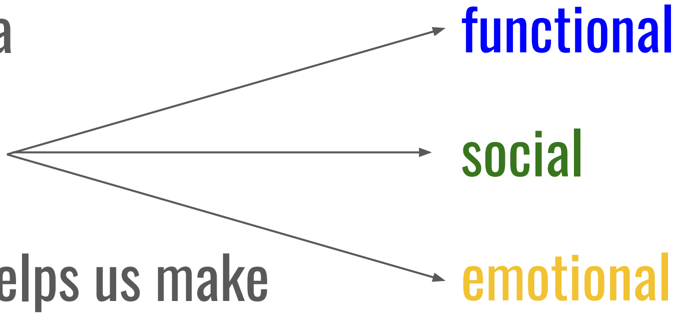
Jobs-to-be-Done

We

hire a product

to do a

job



functional

social

emotional

that helps us make

progress

towards a goal

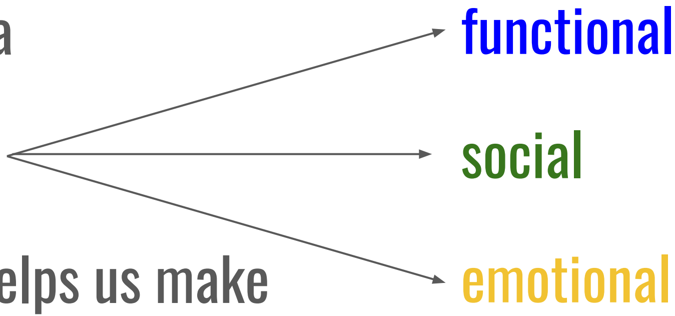
Jobs-to-be-Done

We

hire a product

to do a

job



functional

social

emotional

that helps us make

progress

towards a goal

Let’s

build a team of packages

to do the

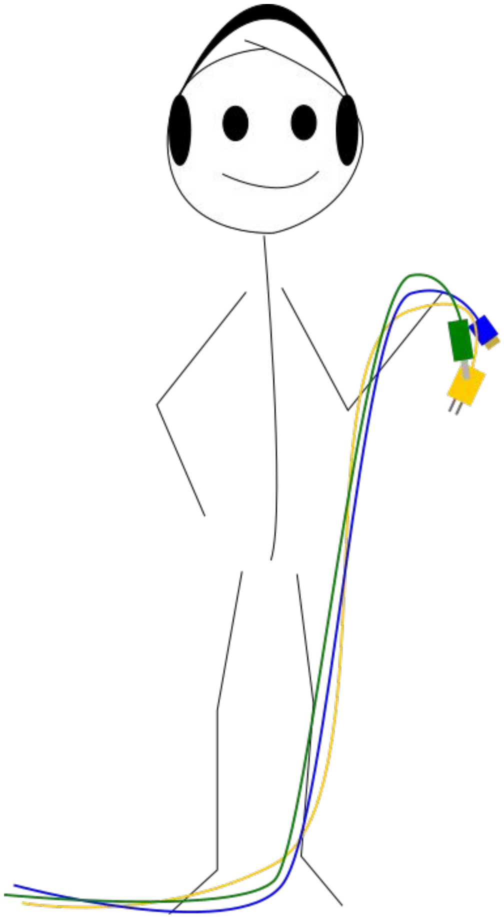
jobs

that helps our org

answer impactful questions

with efficient workflows

The IT Guy

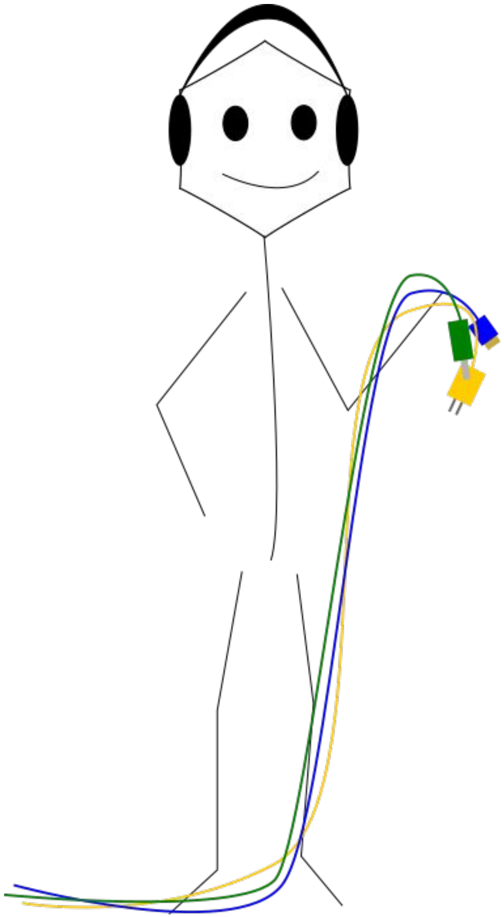


functional handle quirks of infrastructure

social promote or enforce good practices

emotional avoid frustration or stress of time lost

The IT Guy



functional	handle quirks of infrastructure	-> utility functions
social	promote or enforce good practices	-> opinionated design
emotional	avoid frustration or stress of time lost	-> helpful error messages

```r
get_database_conn <- function(username, password) {

conn <-
  DBI::dbConnect(
    drv = odbc::odbc(),
    driver = {driver name},
    server = {server},
    UID = username,
    PWD = password,
    port = {port number}
  )

return(conn)

}
```

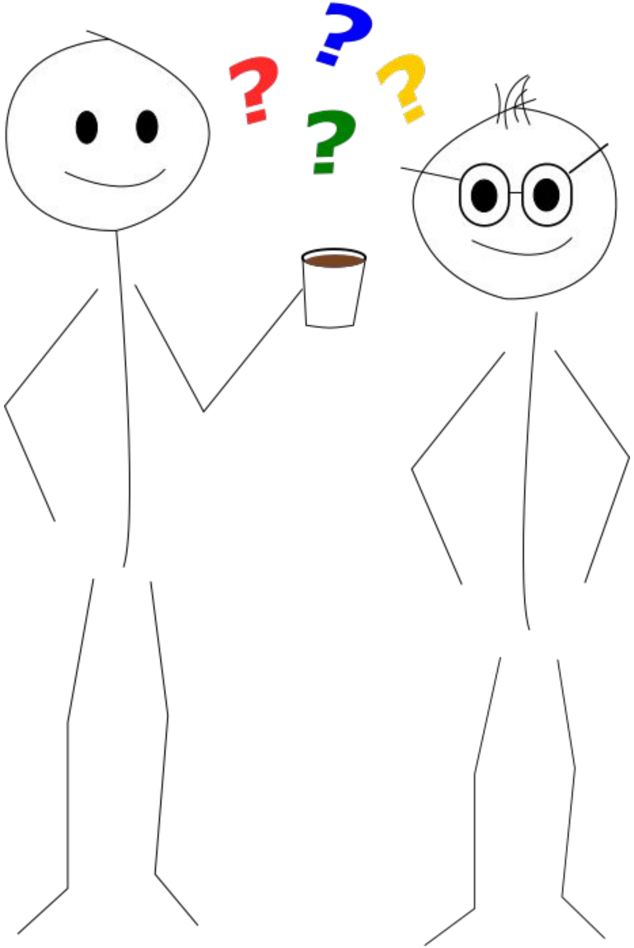
```
get_database_conn <- function(~~username, password~~) {

conn <-
  DBI::dbConnect(
    drv = odbc::odbc(),
    driver = {driver name},
    server = {server},
    UID = Sys.getenv(“DB_USER”) ~~username~~,
    PWD = Sys.getenv(“DB_PASS”) ~~password~~,
    port = {port number}
  )

return(conn)

}
```

```
get_database_conn <- function() {

if (any(Sys.getenv(c(“DB_USER”, “DB_PASS”)) == “”)) {
  stop(
    “DB_USER or DB_PASS environment variables are missing.”,
    “Please read set-up vignette to configure your system.”
  )
}

conn <-
  DBI::dbConnect(
    drv = odbc::odbc(),
    driver = {driver name},
    server = {server},
    UID = Sys.getenv(“DB_USER”),
    PWD = Sys.getenv(“DB_PASS”),
    port = {port number}
  )

return(conn)

}
```

```
get_database_conn <- function() {

if (any(Sys.getenv(c(“DB_USER”, “DB_PASS”)) == “”)) {
  stop(
    “DB_USER or DB_PASS environment variables are missing.”,
    “Please read set-up vignette to configure your system.”
  )
}

conn <-
  DBI::dbConnect(
    drv = odbc::odbc(),
    driver = {driver name},
    server = {server},
    UID = Sys.getenv(“DB_USER”),
    PWD = URLencode(Sys.getenv(“DB_PASS”), reserved = TRUE),
    port = {port number}
  )

return(conn)

}
```

The Junior Analyst

functional	perform work with reasonable assumptions
social	flexible to feedback, trying new things
emotional	builds trust so you can focus on other things



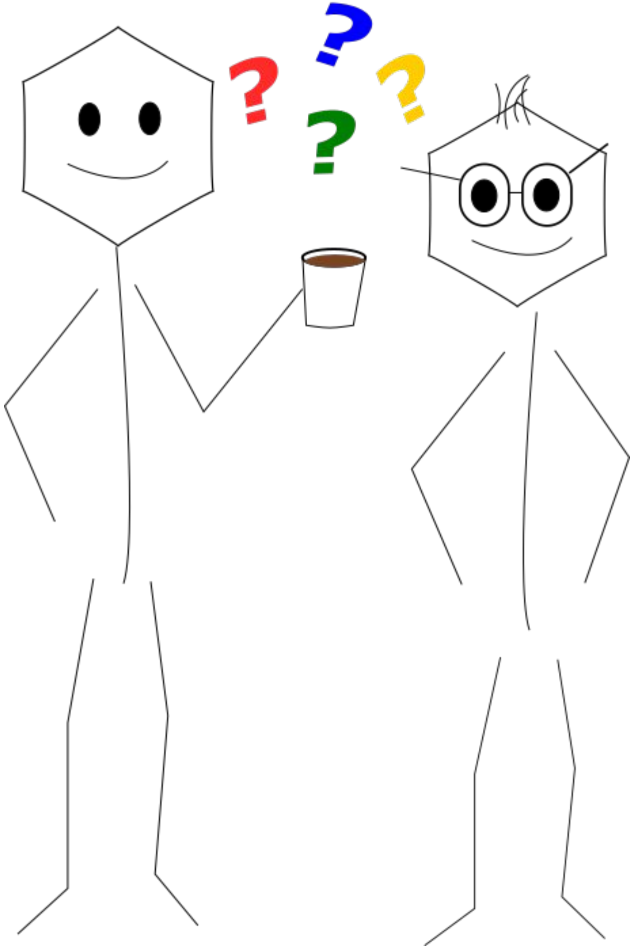
The Junior Analyst

functional perform work with reasonable assumptions

social flexible to feedback, trying new things

emotional builds trust so you can focus on other things

-> default arguments
-> reserved keywords
-> ellipsis

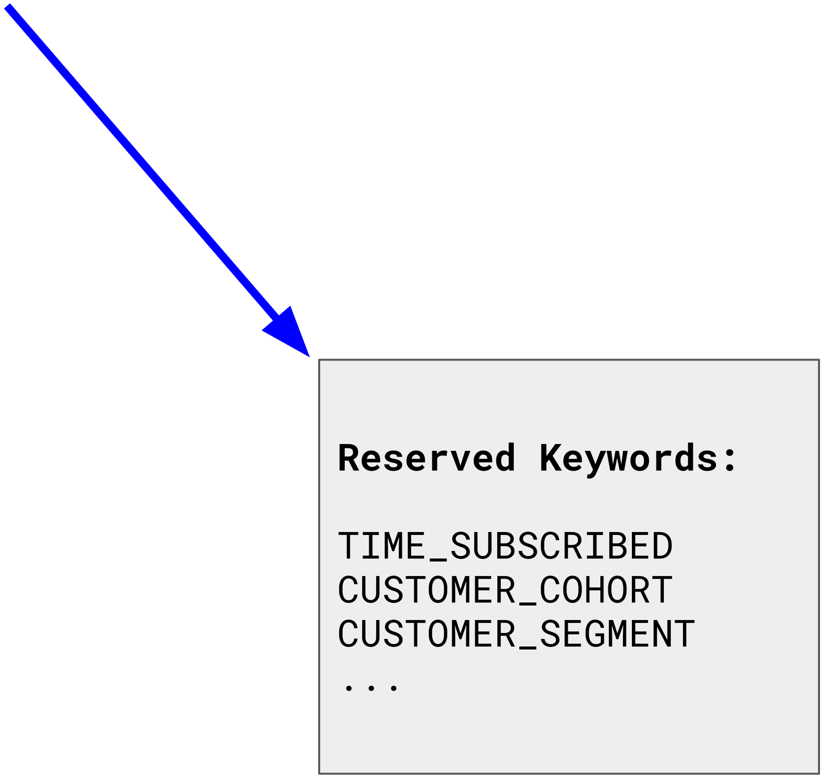


```r
viz_cohort <- function(data, time, metric, group) {

  gg <-
    ggplot(data) +
    aes(x = .data[[time]],
        y = .data[[metric]],
        group = .data[[group]]) +
    geom_line() +
    my_org_theme()

  return(gg)

}
```

```
viz_cohort <- function(data, ~~time~~, metric, group) {

  gg <-
    ggplot(data) +
    aes(x = .data[[“MONTHS_SUBSCRIBED”]],
        y = .data[[metric]],
        group = .data[[group]]) +
    geom_line() +
    my_org_theme()

  return(gg)

}
```

```
viz_cohort <- function(data,
                       metric = “IND_ACTIVE”,
                       time = “MONTHS_SUBSCRIBED”,
                       group = “COHORT”) {

  gg <-
    ggplot(data) +
    aes(x = .data[[time]],
        y = .data[[metric]],
        group = .data[[group]]) +
    geom_line() +
    my_org_theme()

  return(gg)

}
```

```
viz_cohort <- function(data,
                       metric = “IND_ACTIVE”,
                       time = “MONTHS_SUBSCRIBED”,
                       group = “COHORT”) {

  gg <-
    ggplot(data) +
    aes(x = .data[[time]],
        y = .data[[metric]],
        group = .data[[group]]) +
    geom_line() +
    my_org_theme()

  return(gg)

}
```



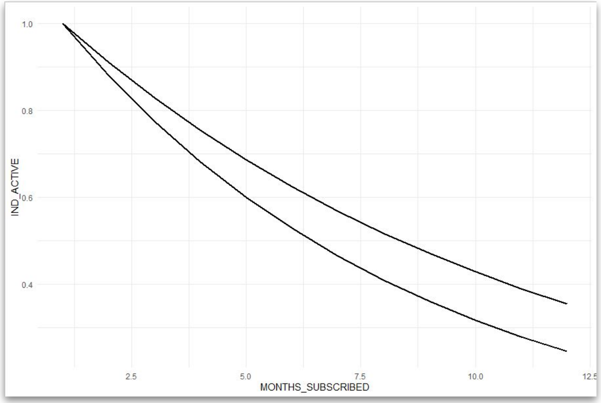
Reserved Keywords:

```
TIME_SUBSCRIBED
CUSTOMER_COHORT
CUSTOMER_SEGMENT
...
```

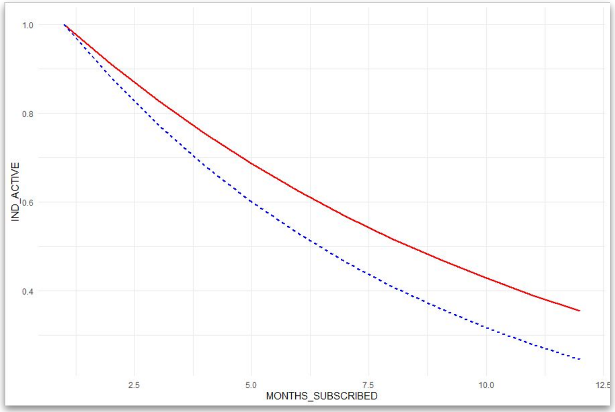
```r
viz_cohort <- function(data,
                       time = “MONTHS_SUBSCRIBED”,
                       metric = “IND_ACTIVE”,
                       group = “COHORT”,
                       ...) {

  gg <-
    ggplot(data) +
    aes(x = .data[[time]],
        y = .data[[metric]],
        group = .data[[group]]) +
    geom_line(aes(...)) +
    my_org_theme()

  return(gg)

}
```

```
viz_cohort <- function(data,
                       time = “MONTHS_SUBSCRIBED”,
                       metric = “IND_ACTIVE”,
                       group = “COHORT”,
                       ...) {

  gg <-
    ggplot(data) +
    aes(x = .data[[time]],
        y = .data[[metric]],
        group = .data[[group]]) +
    geom_line(aes(...)) +
    my_org_theme()

  return(gg)

}
```

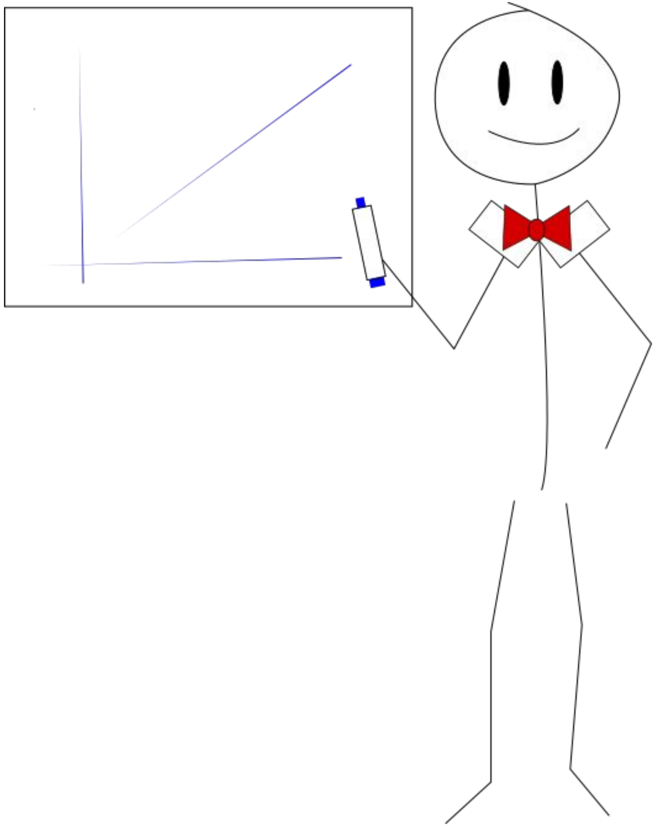
```
> viz_cohort(my_data)
```



```
> viz_cohort(my_data,
             color = COHORT,
             linetype = COHORT)
```



The Tech Lead

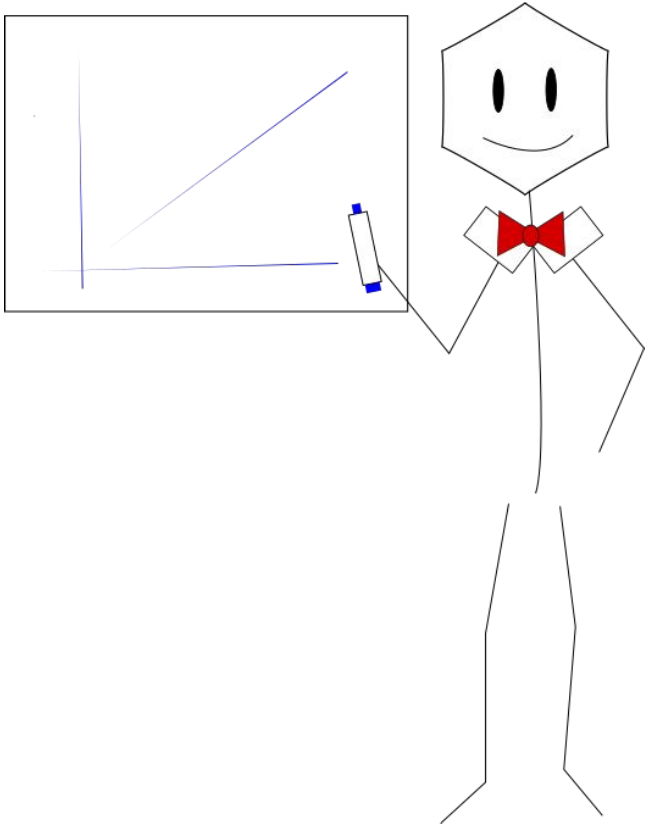


functional coach you through issues & alternatives

social share collected knowledge

emotional inspire you to do your best work

The Tech Lead



functional help navigate common issues & alternatives

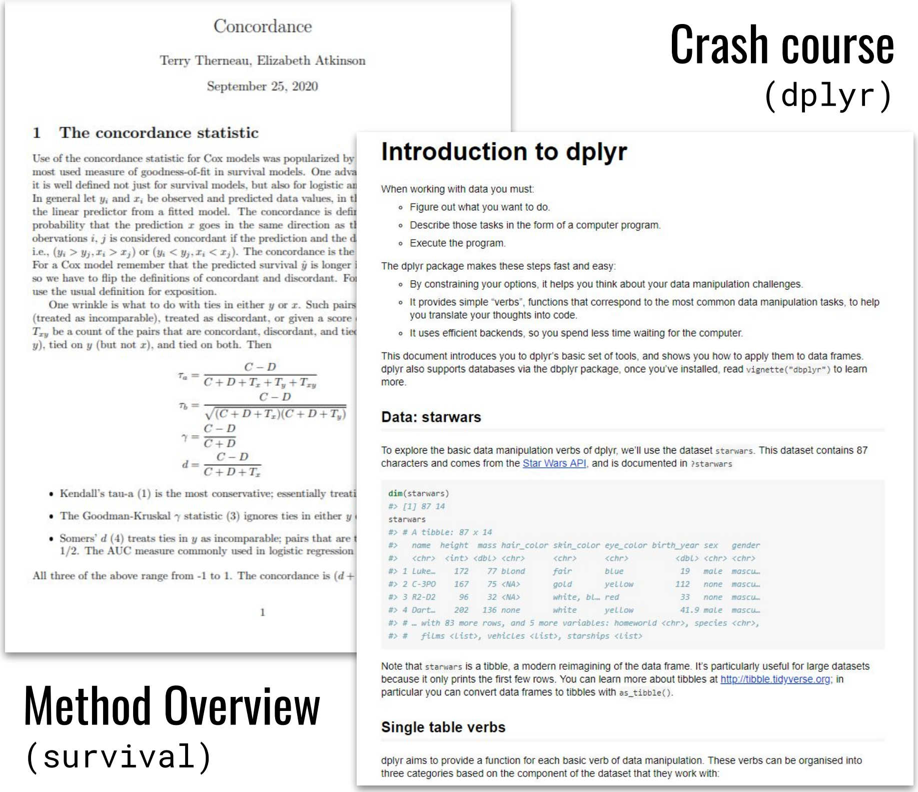
social share collected knowledge

emotional connect to latent community of practice

-> vignettes

-> templates

Vignettes as a time capsule for knowledge transfer



Concordance

Terry Therneau, Elizabeth Atkinson

September 25, 2020

1 The concordance statistic

Use of the concordance statistic for Cox models was popularized by
most used measure of goodness-of-fit in survival models. One adva
it is well defined not just for survival models, but also for logistic ar
In general let y_i and x_i be observed and predicted data values, in tl
the linear predictor from a fitted model. The concordance is defir
probability that the prediction x goes in the same direction as tl
obervations i, j is considered concordant if the prediction and the d
i.e., $(y_i > y_j, x_i > x_j)$ or $(y_i < y_j, x_i < x_j)$. The concordance is the
For a Cox model remember that the predicted survival $\hat{y}$ is longer i
so we have to flip the definitions of concordant and discordant. Fo
use the usual definition for exposition.

One wrinkle is what to do with ties in either y or x. Such pairs
(treated as incomparable), treated as discordant, or given a score
T_{xy} be a count of the pairs that are concordant, discordant, and tie
y), tied on y (but not x), and tied on both. Then

$$\tau_a = \frac{C-D}{C+D+T_x+T_y+T_{xy}}$$
$$\tau_b = \frac{C-D}{\sqrt{(C+D+T_x)(C+D+T_y)}}$$
$$\gamma = \frac{C-D}{C+D}$$
$$d = \frac{C-D}{C+D+T_x}$$

- Kendall's tau-a (1) is the most conservative; essentially treati
- The Goodman-Kruskal γ statistic (3) ignores ties in either y
- Somers' d (4) treats ties in y as incomparable; pairs that are t
  1/2. The AUC measure commonly used in logistic regression

All three of the above range from -1 to 1. The concordance is $(d +$

1

Introduction to dplyr

When working with data you must:

- Figure out what you want to do.
- Describe those tasks in the form of a computer program.
- Execute the program.

The dplyr package makes these steps fast and easy:

- By constraining your options, it helps you think about your data manipulation challenges.
- It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
- It uses efficient backends, so you spend less time waiting for the computer.

This document introduces you to dplyr's basic set of tools, and shows you how to apply them to data frames. dplyr also supports databases via the dbplyr package, once you've installed, read `vignette("dbplyr")` to learn more.

Data: starwars

To explore the basic data manipulation verbs of dplyr, we'll use the dataset `starwars`. This dataset contains 87 characters and comes from the Star Wars API, and is documented in `?starwars`

```
dim(starwars)
#> [1] 87 14
starwars
#> # A tibble: 87 x 14
#>   name  height  mass hair_color skin_color eye_color birth_year sex   gender
#>   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
#> 1 Luke…    172    77 blond      fair       blue            19   male  mascu…
#> 2 C-3PO    167    75 <NA>       gold       yellow         112   none  mascu…
#> 3 R2-D2     96    32 <NA>       white, bl… red             33   none  mascu…
#> 4 Dart…    202   136 none       white      yellow          41.9 male  mascu…
#> # … with 83 more rows, and 5 more variables: homeworld <chr>, species <chr>,
#> #   films <list>, vehicles <list>, starships <list>
```

Note that `starwars` is a tibble, a modern reimagining of the data frame. It's particularly useful for large datasets because it only prints the first few rows. You can learn more about tibbles at http://tibble.tidyverse.org; in particular you can convert data frames to tibbles with `as_tibble()`.

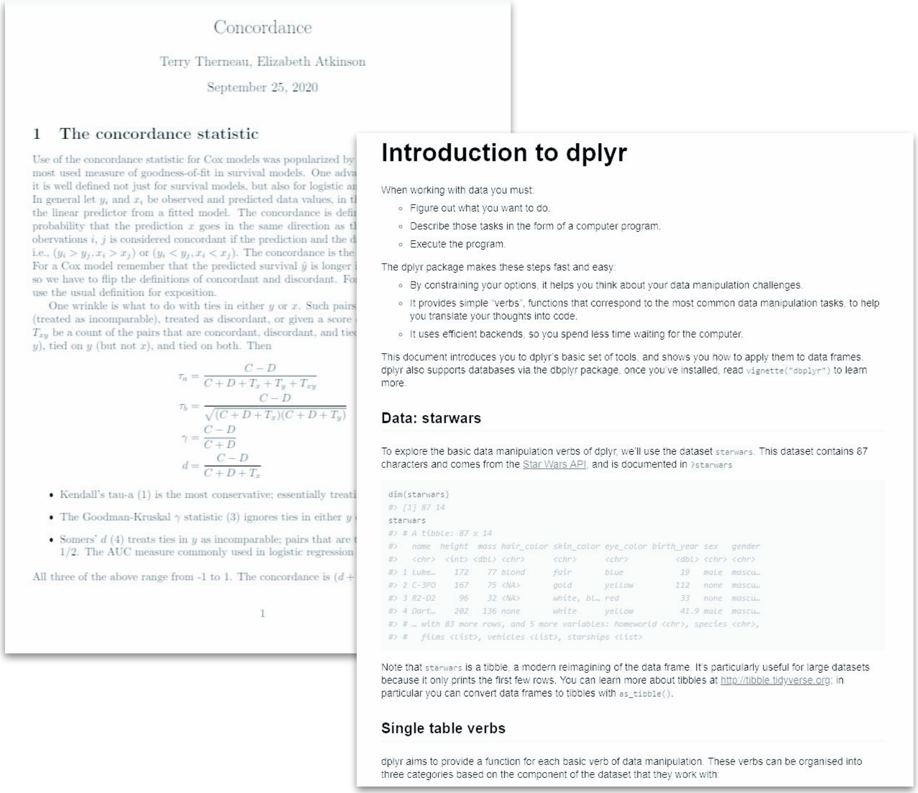
Single table verbs

dplyr aims to provide a function for each basic verb of data manipulation. These verbs can be organised into three categories based on the component of the dataset that they work with:

Crash course
(dplyr)

Method Overview
(survival)

Vignettes as a time capsule for knowledge transfer



Concordance

Terry Therneau, Elizabeth Atkinson

September 25, 2020

1 The concordance statistic

Use of the concordance statistic for Cox models was popularized by
most used measure of goodness-of-fit in survival models. One adva
it is well defined not just for survival models, but also for logistic an
In general let y_i and x_i be observed and predicted data values, in t
the linear predictor from a fitted model. The concordance is defi
probability that the prediction x goes in the same direction as t
obervations i, j is considered concordant if the prediction and the d
i.e., $(y_i > y_j, x_i > x_j)$ or $(y_i < y_j, x_i < x_j)$. The concordance is the
For a Cox model remember that the predicted survival $\hat{y}$ is longer
so we have to flip the definitions of concordant and discordant. Fo
use the usual definition for exposition.

One wrinkle is what to do with ties in either y or x. Such pairs
(treated as incomparable), treated as discordant, or given a score
T_{xy} be a count of the pairs that are concordant, discordant, and tie
y), tied on y (but not x), and tied on both. Then

$$\tau_a = \frac{C-D}{C+D+T_x+T_y+T_{xy}}$$
$$\tau_b = \frac{C-D}{\sqrt{(C+D+T_x)(C+D+T_y)}}$$
$$\gamma = \frac{C-D}{C+D}$$
$$d = \frac{C-D}{C+D+T_x}$$

- Kendall's tau-a (1) is the most conservative; essentially treati
- The Goodman-Kruskal γ statistic (3) ignores ties in either y
- Somers' d (4) treats ties in y as incomparable; pairs that are
  1/2. The AUC measure commonly used in logistic regression

All three of the above range from -1 to 1. The concordance is $(d +$

1

Introduction to dplyr

When working with data you must:

- Figure out what you want to do.
- Describe those tasks in the form of a computer program.
- Execute the program.

The dplyr package makes these steps fast and easy:

- By constraining your options, it helps you think about your data manipulation challenges.
- It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
- It uses efficient backends, so you spend less time waiting for the computer.

This document introduces you to dplyr's basic set of tools, and shows you how to apply them to data frames. dplyr also supports databases via the dbplyr package, once you've installed, read `vignette("dbplyr")` to learn more.

Data: starwars

To explore the basic data manipulation verbs of dplyr, we'll use the dataset `starwars`. This dataset contains 87 characters and comes from the Star Wars API, and is documented in `?starwars`

```
dim(starwars)
#> [1] 87 14
starwars
#> # A tibble: 87 x 14
#>   name  height  mass hair_color skin_color eye_color birth_year sex   gender
#>   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
#> 1 Luke…    172    77 blond      fair       blue            19   male  mascu…
#> 2 C-3PO    167    75 <NA>       gold       yellow         112   none  mascu…
#> 3 R2-D2     96    32 <NA>       white, bl… red             33   none  mascu…
#> 4 Dart…    202   136 none       white      yellow          41.9 male  mascu…
#> # … with 83 more rows, and 5 more variables: homeworld <chr>, species <chr>,
#> #   films <list>, vehicles <list>, starships <list>
```

Note that `starwars` is a tibble, a modern reimagining of the data frame. It's particularly useful for large datasets because it only prints the first few rows. You can learn more about tibbles at http://tibble.tidyverse.org; in particular you can convert data frames to tibbles with `as_tibble()`.

Single table verbs

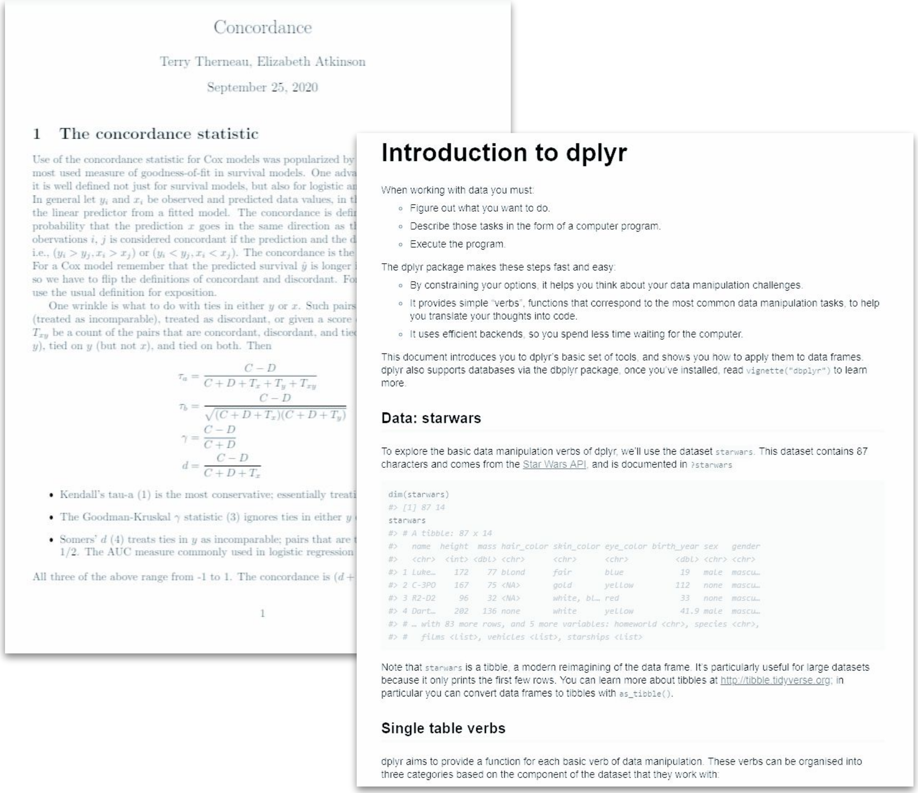
dplyr aims to provide a function for each basic verb of data manipulation. These verbs can be organised into three categories based on the component of the dataset that they work with:

Conceptual Overview

Workflow & Key Questions

Process Documentation

Vignettes as a time capsule for knowledge transfer



Concordance

Terry Therneau, Elizabeth Atkinson

September 25, 2020

1 The concordance statistic

Use of the concordance statistic for Cox models was popularized by
most used measure of goodness-of-fit in survival models. One adva
it is well defined not just for survival models, but also for logistic an
In general let y_i and x_i be observed and predicted data values, in t
the linear predictor from a fitted model. The concordance is defi
probability that the prediction x goes in the same direction as t
obervations i, j is considered concordant if the prediction and the d
i.e., $(y_i > y_j, x_i > x_j)$ or $(y_i < y_j, x_i < x_j)$. The concordance is the
For a Cox model remember that the predicted survival $\hat{y}$ is longer
so we have to flip the definitions of concordant and discordant. Fo
use the usual definition for exposition.

One wrinkle is what to do with ties in either y or x. Such pairs
(treated as incomparable), treated as discordant, or given a score
T_{xy} be a count of the pairs that are concordant, discordant, and tie
y), tied on y (but not x), and tied on both. Then

$$\tau_a = \frac{C-D}{C+D+T_x+T_y+T_{xy}}$$

$$\tau_b = \frac{C-D}{\sqrt{(C+D+T_x)(C+D+T_y)}}$$

$$\gamma = \frac{C-D}{C+D}$$

$$d = \frac{C-D}{C+D+T_x}$$

- Kendall's tau-a (1) is the most conservative; essentially treati
- The Goodman-Kruskal γ statistic (3) ignores ties in either y
- Somers' d (4) treats ties in y as incomparable; pairs that are
  1/2. The AUC measure commonly used in logistic regression

All three of the above range from -1 to 1. The concordance is $(d +$

1

Introduction to dplyr

When working with data you must:

- Figure out what you want to do.
- Describe those tasks in the form of a computer program.
- Execute the program.

The dplyr package makes these steps fast and easy:

- By constraining your options, it helps you think about your data manipulation challenges.
- It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
- It uses efficient backends, so you spend less time waiting for the computer.

This document introduces you to dplyr's basic set of tools, and shows you how to apply them to data frames. dplyr also supports databases via the dbplyr package, once you've installed, read `vignette("dbplyr")` to learn more.

Data: starwars

To explore the basic data manipulation verbs of dplyr, we'll use the dataset `starwars`. This dataset contains 87 characters and comes from the Star Wars API, and is documented in `?starwars`

```
dim(starwars)
#> [1] 87 14
starwars
#> # A tibble: 87 x 14
#>   name  height  mass hair_color skin_color eye_color birth_year sex   gender
#>   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
#> 1 Luke…    172    77 blond      fair       blue            19   male  mascu…
#> 2 C-3PO    167    75 <NA>       gold       yellow         112   none  mascu…
#> 3 R2-D2     96    32 <NA>       white, bl… red             33   none  mascu…
#> 4 Dart…    202   136 none       white      yellow          41.9 male  mascu…
#> # … with 83 more rows, and 5 more variables: homeworld <chr>, species <chr>,
#> #   films <list>, vehicles <list>, starships <list>
```

Note that `starwars` is a tibble, a modern reimagining of the data frame. It's particularly useful for large datasets because it only prints the first few rows. You can learn more about tibbles at http://tibble.tidyverse.org; in particular you can convert data frames to tibbles with `as_tibble()`.

Single table verbs

dplyr aims to provide a function for each basic verb of data manipulation. These verbs can be organised into three categories based on the component of the dataset that they work with:

Conceptual Overview

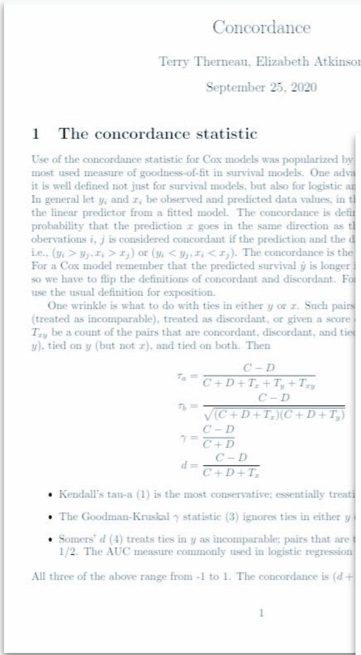
Workflow & Key Questions

Process Documentation

Technical Overview

Methods Comparison

Vignettes as a time capsule for knowledge transfer



Concordance

Terry Therneau, Elizabeth Atkinson

September 25, 2020

1 The concordance statistic

Use of the concordance statistic for Cox models was popularized by
most used measure of goodness-of-fit in survival models. One adva
it is well defined not just for survival models, but also for logistic an
In general let y_i and x_i be observed and predicted data values, in th
the linear predictor from a fitted model. The concordance is defi
probability that the prediction x goes in the same direction as th
obervations i, j is considered concordant if the prediction and the d
i.e., $(y_i > y_j, x_i > x_j)$ or $(y_i < y_j, x_i < x_j)$. The concordance is the
For a Cox model remember that the predicted survival $\hat{y}$ is longer i
so we have to flip the definitions of concordant and discordant. Fo
use the usual definition for exposition.

One wrinkle is what to do with ties in either y or x. Such pairs
(treated as incomparable), treated as discordant, or given a score
T_{xy} be a count of the pairs that are concordant, discordant, and tie
y), tied on y (but not x), and tied on both. Then

$$\tau_a = \frac{C-D}{C+D+T_x+T_y+T_{xy}}$$

$$\tau_b = \frac{C-D}{\sqrt{(C+D+T_x)(C+D+T_y)}}$$

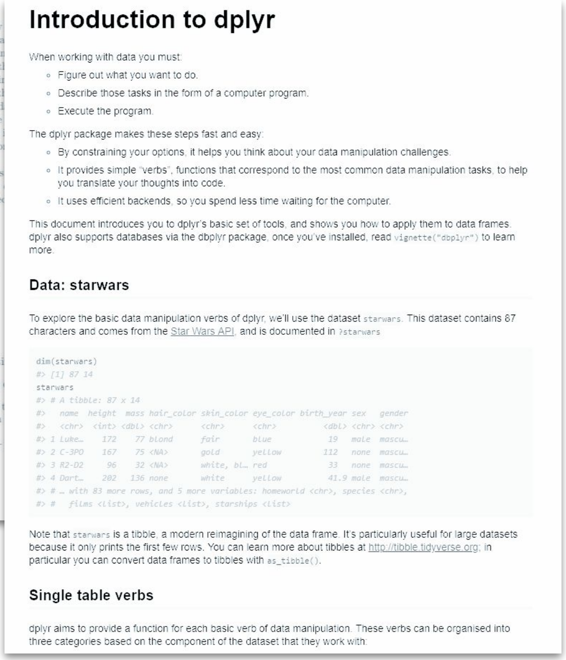
$$\gamma = \frac{C-D}{C+D}$$

$$d = \frac{C-D}{C+D+T_x}$$

- Kendall's tau-a (1) is the most conservative; essentially treati
- The Goodman-Kruskal γ statistic (3) ignores ties in either y
- Somers' d (4) treats ties in y as incomparable; pairs that are t
  1/2. The AUC measure commonly used in logistic regression

All three of the above range from -1 to 1. The concordance is $(d +$

1



Introduction to dplyr

When working with data you must:

- Figure out what you want to do.
- Describe those tasks in the form of a computer program.
- Execute the program.

The dplyr package makes these steps fast and easy:

- By constraining your options, it helps you think about your data manipulation challenges.
- It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
- It uses efficient backends, so you spend less time waiting for the computer.

This document introduces you to dplyr's basic set of tools, and shows you how to apply them to data frames. dplyr also supports databases via the dbplyr package, once you've installed, read `vignette("dbplyr")` to learn more.

Data: starwars

To explore the basic data manipulation verbs of dplyr, we'll use the dataset `starwars`. This dataset contains 87 characters and comes from the Star Wars API, and is documented in `?starwars`

```
dim(starwars)
#> [1] 87 14
starwars
#> # A tibble: 87 x 14
#>   name  height  mass hair_color skin_color eye_color birth_year sex   gender
#>   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
#> 1 Luke…    172    77 blond      fair       blue            19   male  mascu…
#> 2 C-3PO    167    75 <NA>       gold       yellow         112   none  mascu…
#> 3 R2-D2     96    32 <NA>       white, bl… red             33   none  mascu…
#> 4 Dart…    202   136 none       white      yellow          41.9 male  mascu…
#> # … with 83 more rows, and 5 more variables: homeworld <chr>, species <chr>,
#> #   films <list>, vehicles <list>, starships <list>
```

Note that `starwars` is a tibble, a modern reimagining of the data frame. It's particularly useful for large datasets because it only prints the first few rows. You can learn more about tibbles at http://tibble.tidyverse.org; in particular you can convert data frames to tibbles with `as_tibble()`.

Single table verbs

dplyr aims to provide a function for each basic verb of data manipulation. These verbs can be organised into three categories based on the component of the dataset that they work with:

Conceptual Overview

Workflow & Key Questions

Process Documentation

Technical Overview

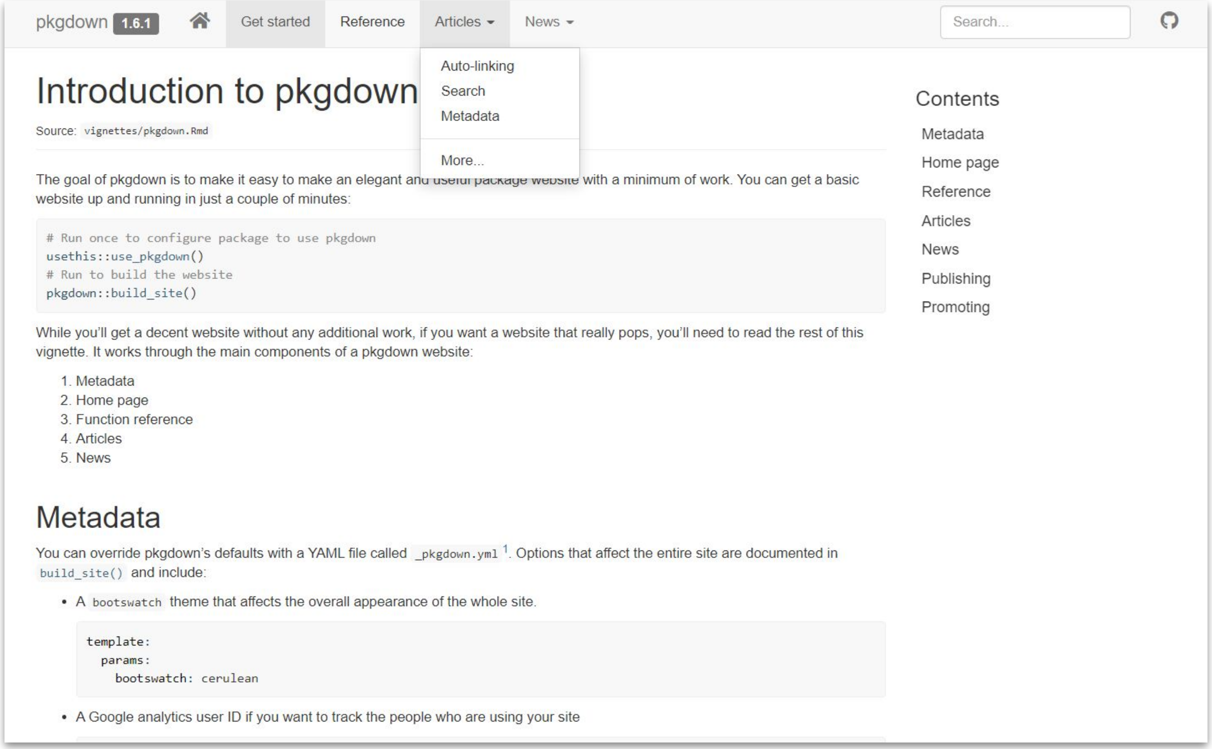
Methods Comparison

Lessons Learned

Past Examples

Expand your reach with pkgdown

```
> pkgdown::build_site()
```



Introduction to pkgdown

Source: `vignettes/pkgdown.Rmd`

The goal of pkgdown is to make it easy to make an elegant and useful package website with a minimum of work. You can get a basic website up and running in just a couple of minutes:

```
# Run once to configure package to use pkgdown
usethis::use_pkgdown()
# Run to build the website
pkgdown::build_site()
```

While you'll get a decent website without any additional work, if you want a website that really pops, you'll need to read the rest of this vignette. It works through the main components of a pkgdown website:

1. Metadata
2. Home page
3. Function reference
4. Articles
5. News

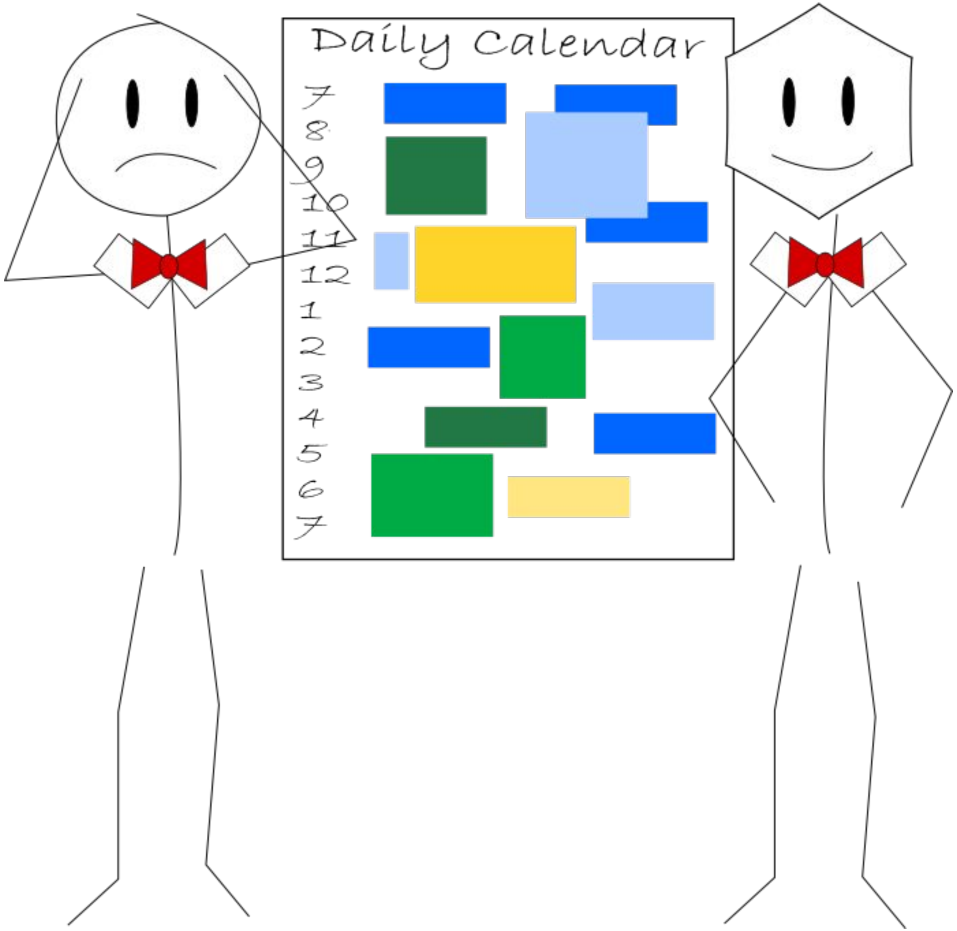
Metadata

You can override pkgdown's defaults with a YAML file called `_pkgdown.yml`[1]. Options that affect the entire site are documented in `build_site()` and include:

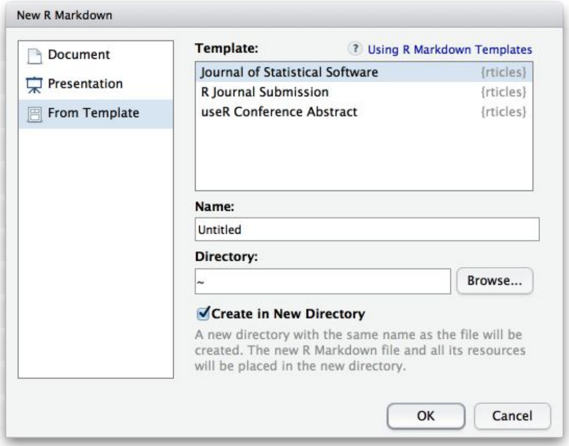
- A `bootswatch` theme that affects the overall appearance of the whole site.

```
template:
  params:
    bootswatch: cerulean
```

- A Google analytics user ID if you want to track the people who are using your site


Daily Calendar
7
8
9
10
11
12
1
2
3
4
5
6
7

Templates as coach



Journal of Statistical Software {rticles}
R Journal Submission {rticles}
useR Conference Abstract {rticles}

Name: Untitled

Directory: ~

☑ Create in New Directory

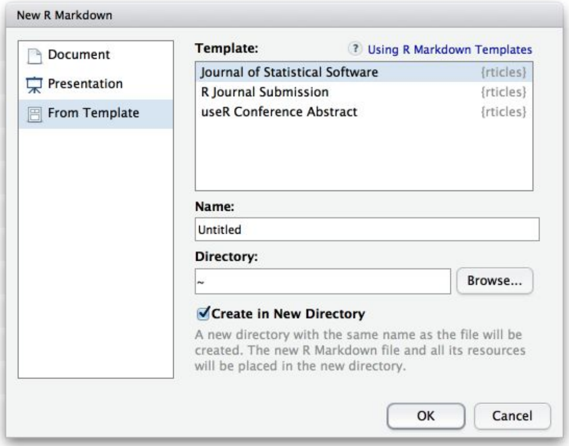
A new directory with the same name as the file will be created. The new R Markdown file and all its resources will be placed in the new directory.

Structure

(flexdashboard)

~~~
---
title: "Untitled"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
---

```{r setup, include=FALSE}
library(flexdashboard)
```

Column {data-width=650}
--------------------------------

### Chart A

```{r}

```

Column {data-width=350}
--------------------------------

### Chart B

```{r}

```
~~~

Templates as coach



New R Markdown

Template:

Journal of Statistical Software	{rticles}
R Journal Submission	{rticles}
useR Conference Abstract	{rticles}

Name: Untitled

Directory: ~

☑ Create in New Directory

A new directory with the same name as the file will be created. The new R Markdown file and all its resources will be placed in the new directory.

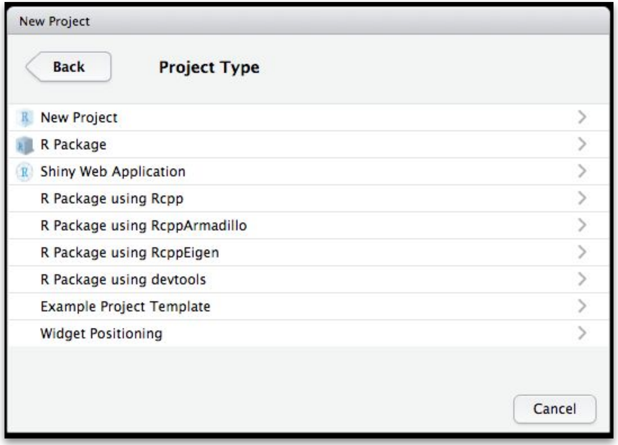
Process walk-through

~~~
---
title: "Data Validation"
output: html_document
---

## Censored Data

Run the following code to visualize how many
observations were censored. Depending on what
you find you will want to...

```{r censored}
~~~

Analysis outline

~~~
---
title: "Final Report"
output: html_document
params:
  month: September
---

## Final Report

TODO: UPDATE COMMENTARY SUMMARIZING TRENDS

```{r dashboard}
~~~

Templates as code reviewer



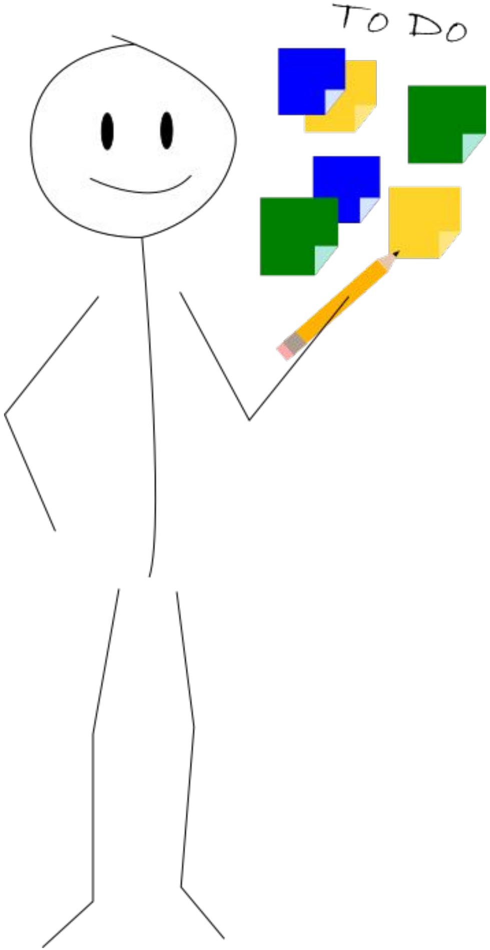
```
scripts/
    data-pull.R
fun/
data/
    sample-data.rds
analysis/
output/
```

The Project Manager

functional integrates work

social finds common ground

emotional meets you where you are



The Project Manager

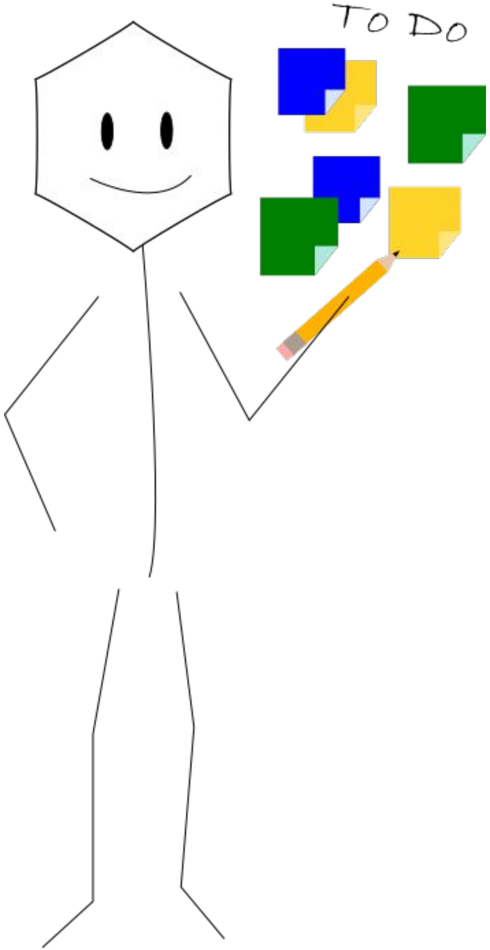
functional integrates work

social finds common ground

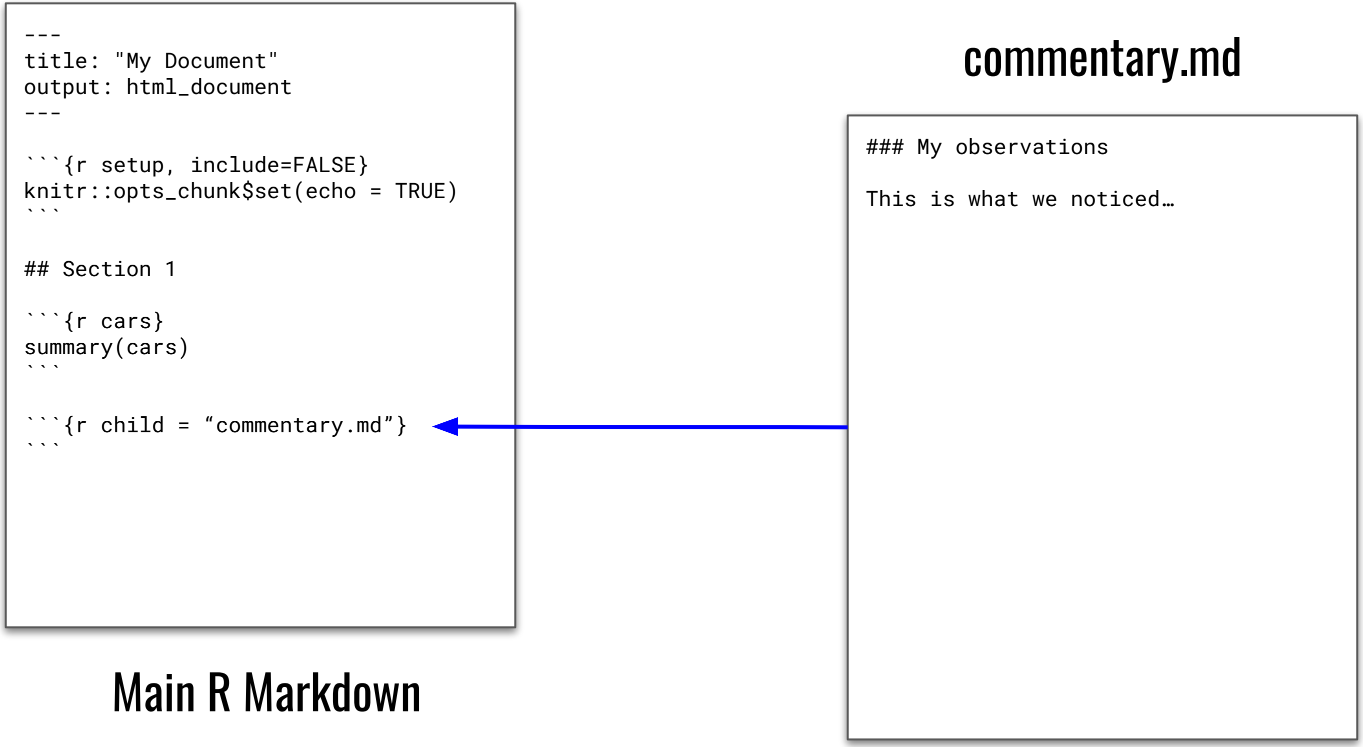
emotional meets you where you are

-> modularized workflow

-> IDE support



Modularization



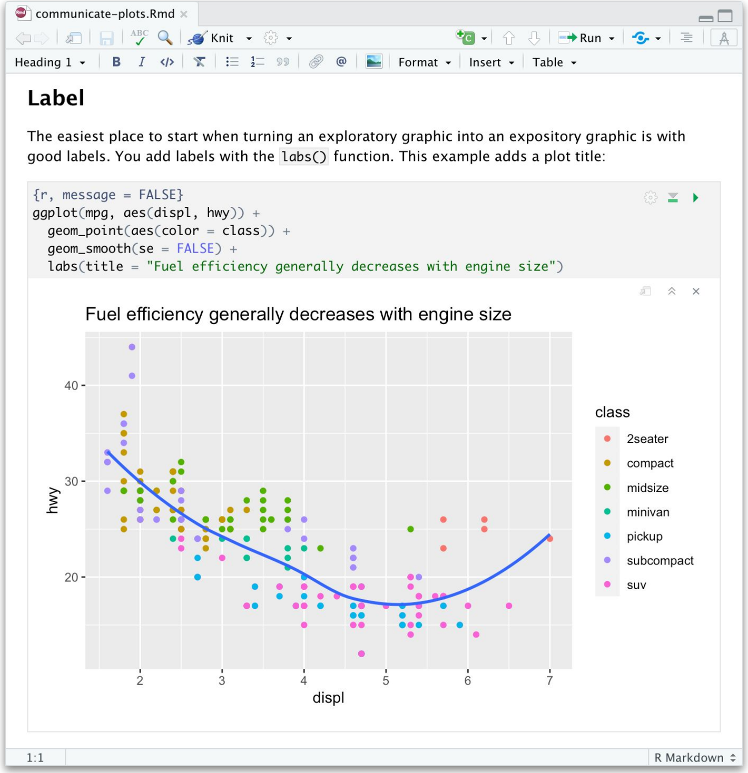
~~~
---
title: "My Document"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## Section 1

```{r cars}
summary(cars)
```

```{r child = “commentary.md”}
```
~~~

Main R Markdown

commentary.md

````
### My observations

This is what we noticed…
````

IDE Support

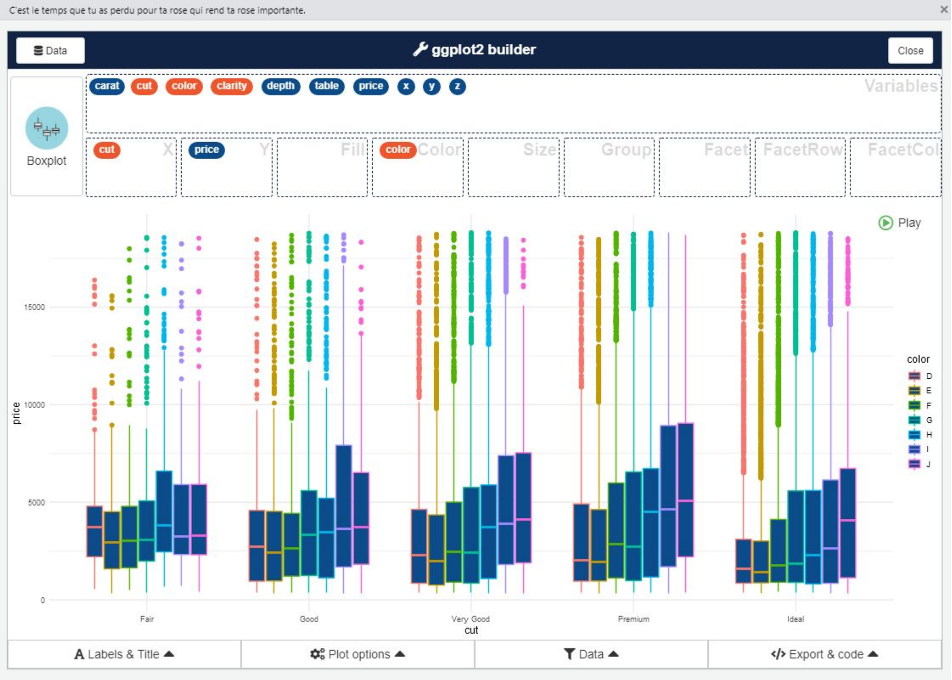


Label

The easiest place to start when turning an exploratory graphic into an expository graphic is with good labels. You add labels with the `labs()` function. This example adds a plot title:

```
{r, message = FALSE}
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  labs(title = "Fuel efficiency generally decreases with engine size")
```

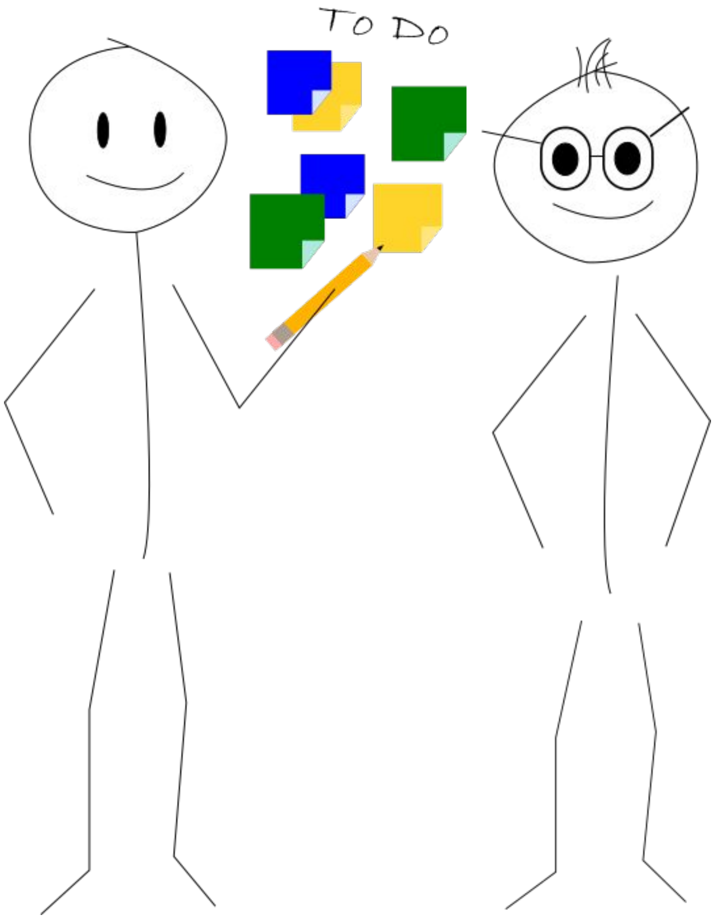
Visual Editor



C'est le temps que tu as perdu pour ta rose qui rend ta rose importante.

Add-Ins (e.g. `esquisse`)

Collaboration

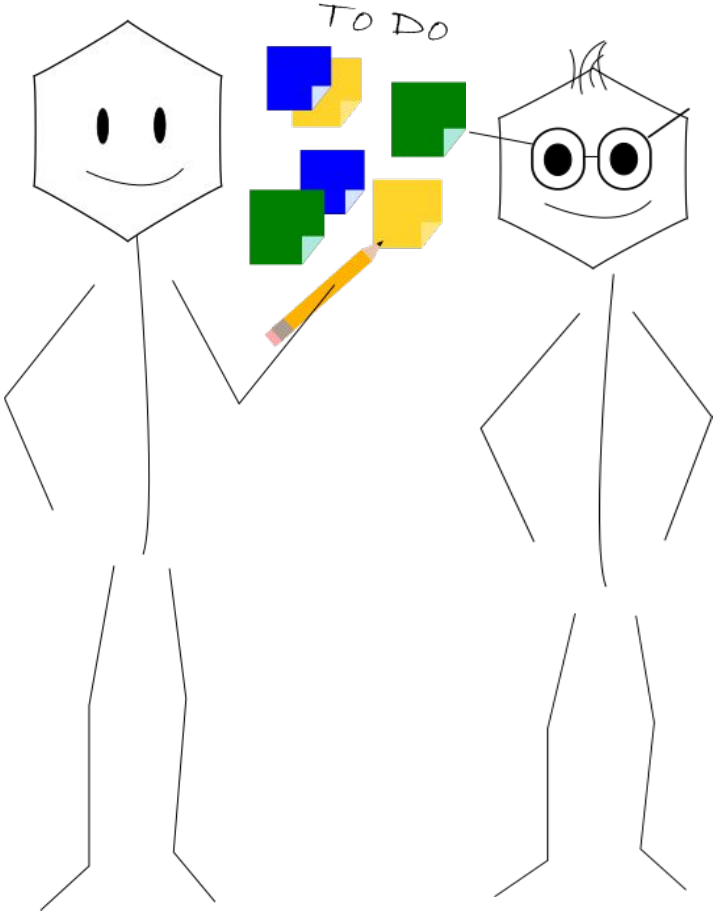


functional clear communication

social keeps promises

emotional confident yet engaged

Collaboration



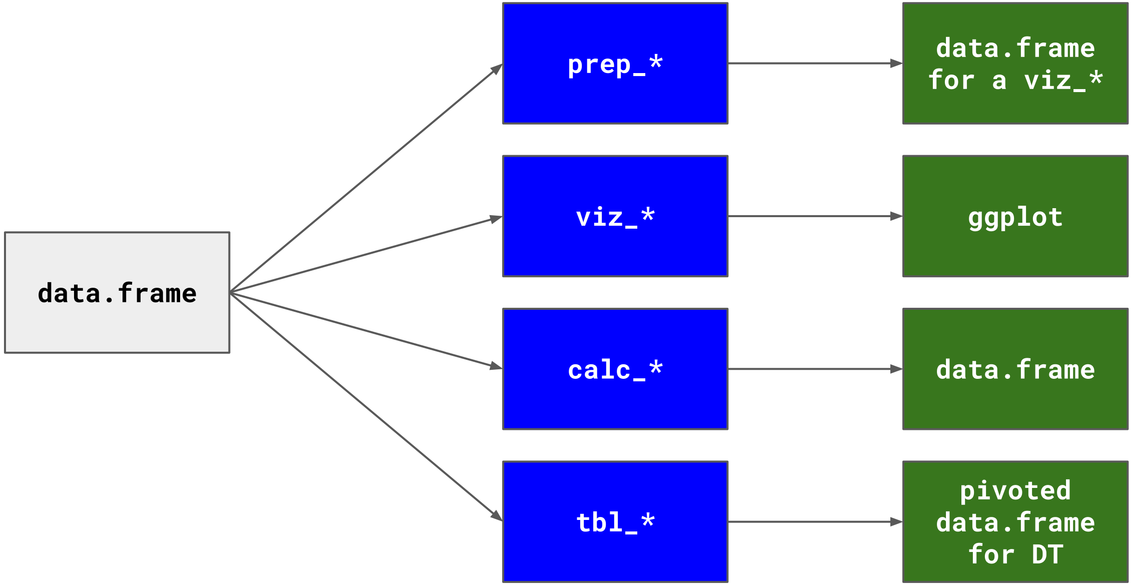
functional clear communication

social keeps promises

emotional confident yet engaged

-> naming
-> scope
-> dependencies
-> testing

Clear communication



Clear ownership

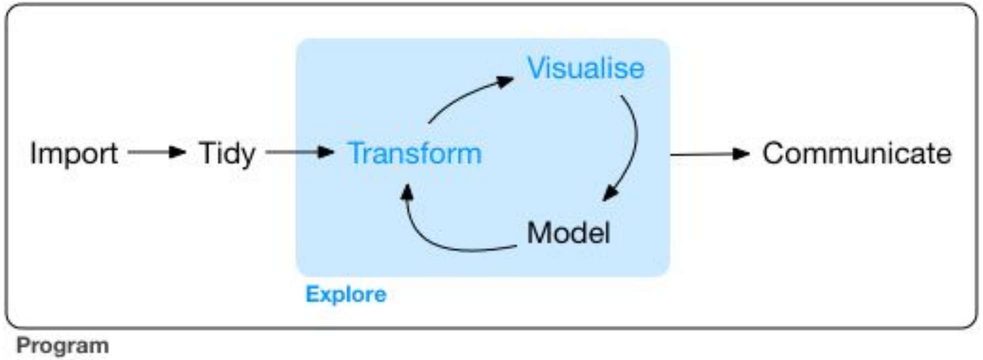


Image from *R for Data Science (Wickham & Grolemund)*

Clear ownership

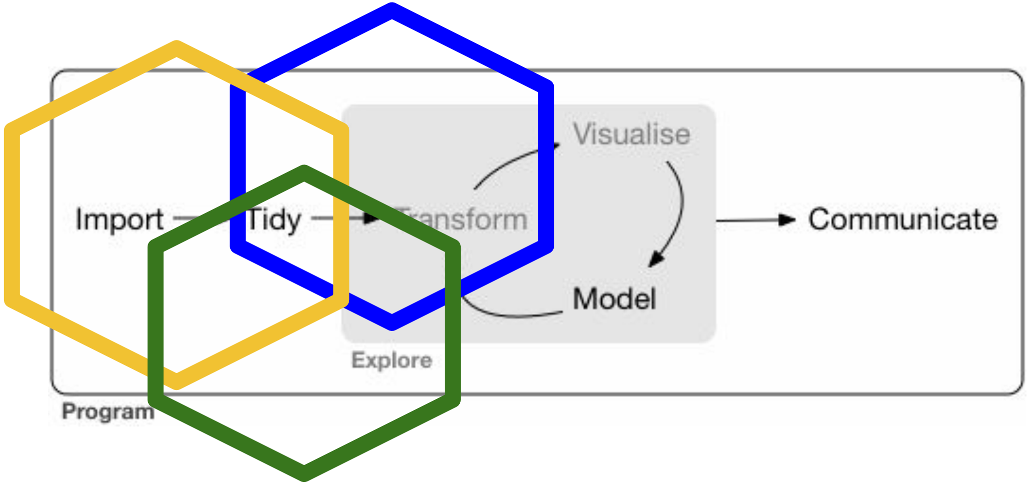


Image from *R for Data Science (Wickham & Grolemund)*

Clear ownership

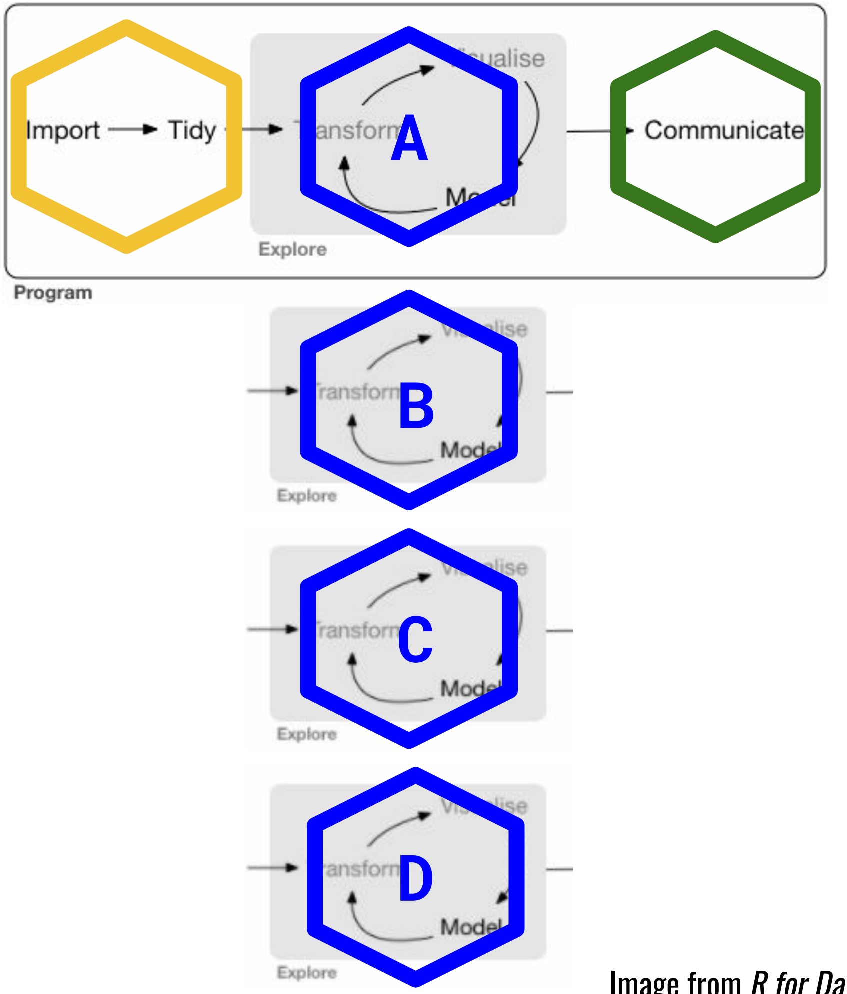


Image from *R for Data Science (Wickham & Grolemund)*

Clear ownership

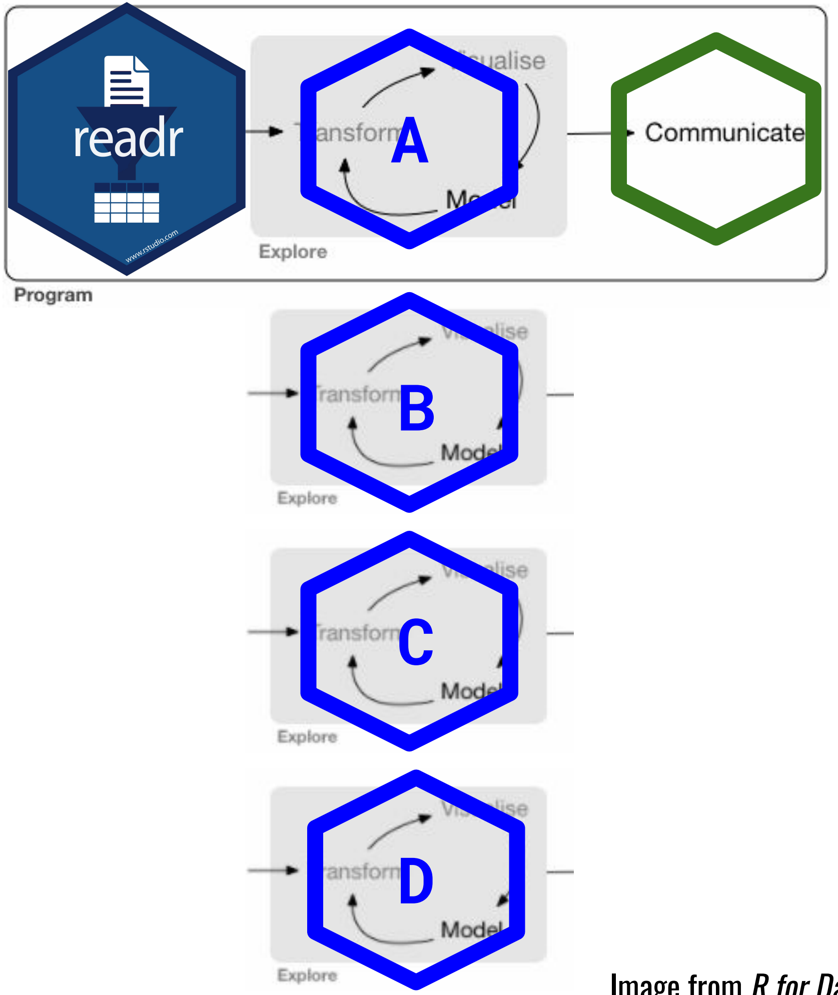
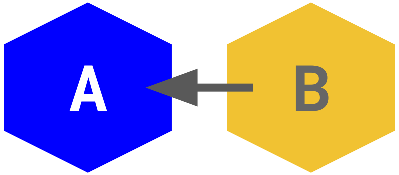


Image from *R for Data Science (Wickham & Grolemund)*

Dependency structures

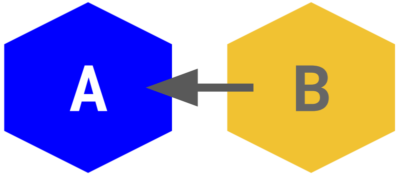


```
a_fx <- function() {...}
```

```
b_fx <- function() {
    ...
    a_fx()
    ...
}
```

Direct Dependency

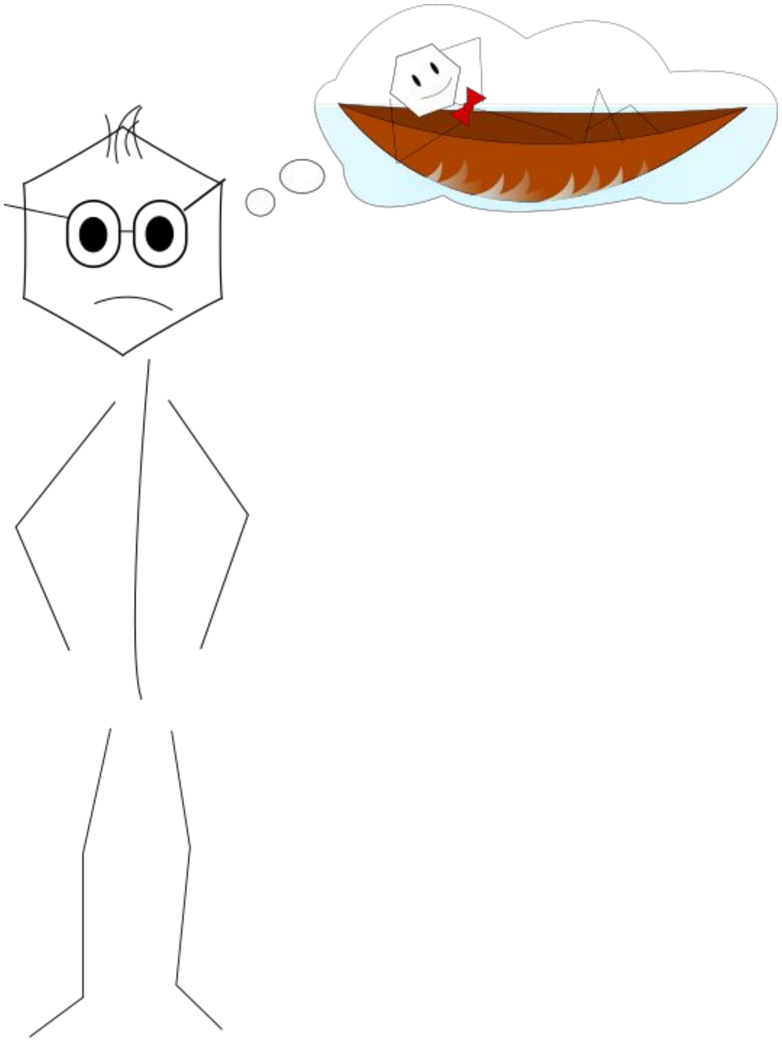
Dependency structures



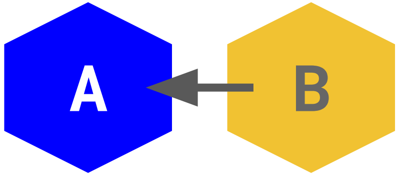
```
a_fx <- function() {...}
```

```
b_fx <- function() {
     ...
     a_fx()
     ...
}
```

Direct Dependency



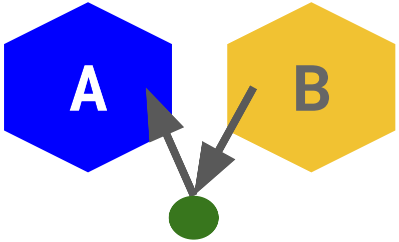
Dependency structures



```
a_fx <- function() {...}
```

```
b_fx <- function() {
     ...
     a_fx()
     ...
}
```

Direct Dependency

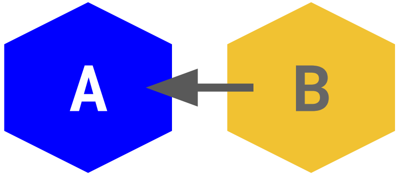


```
a_fx <- function() {...}
```

```
b_fx <- function(a_input) {
     ...
     do_something(a_input)
     ...
}
```

Clean Hand Off

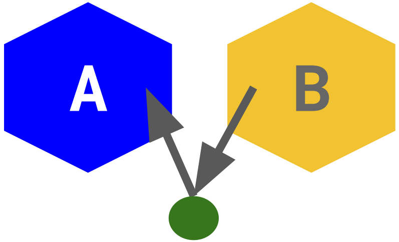
Dependency structures



```
a_fx <- function() {...}
```

```
b_fx <- function() {
    ...
    a_fx()
    ...
}
```

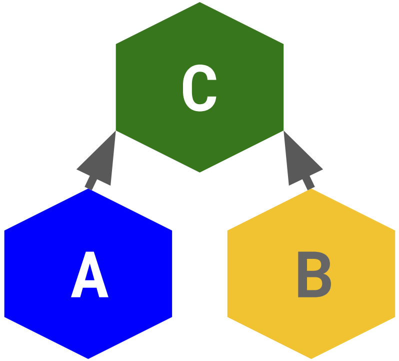
Direct Dependency



```
a_fx <- function() {...}
```

```
b_fx <- function(a_input) {
    ...
    do_something(a_input)
    ...
}
```

Clean Hand Off

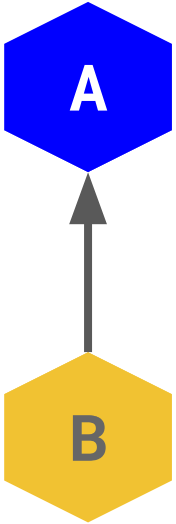


```
b_fx <- function() {
    ...
    c_fx()
    ...
}
```

```
b_fx <- function() {
    ...
    c_fx()
    ...
}
```

Common Parent

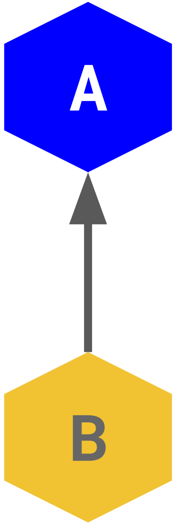
Typical unit test with dependency

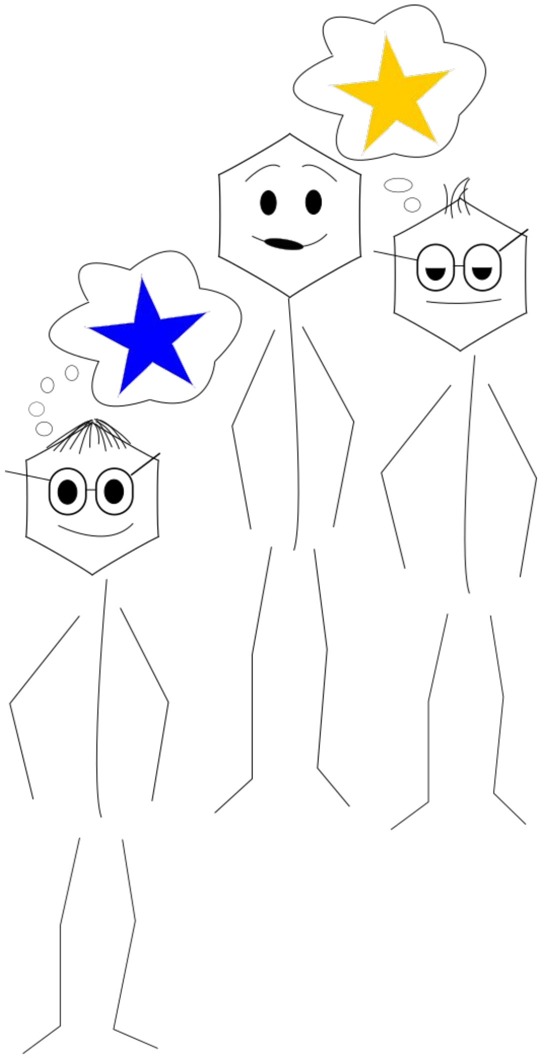


b/tests/testthat/test-pkga.R

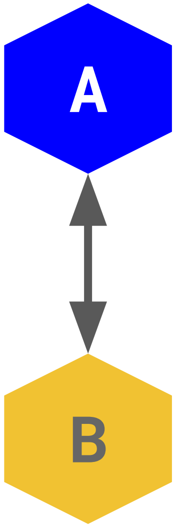
```
test_that(
  “Receives input correctly from a”,
 {
  expect_error(fxb(fxa(1)), NA)
 }
)
```

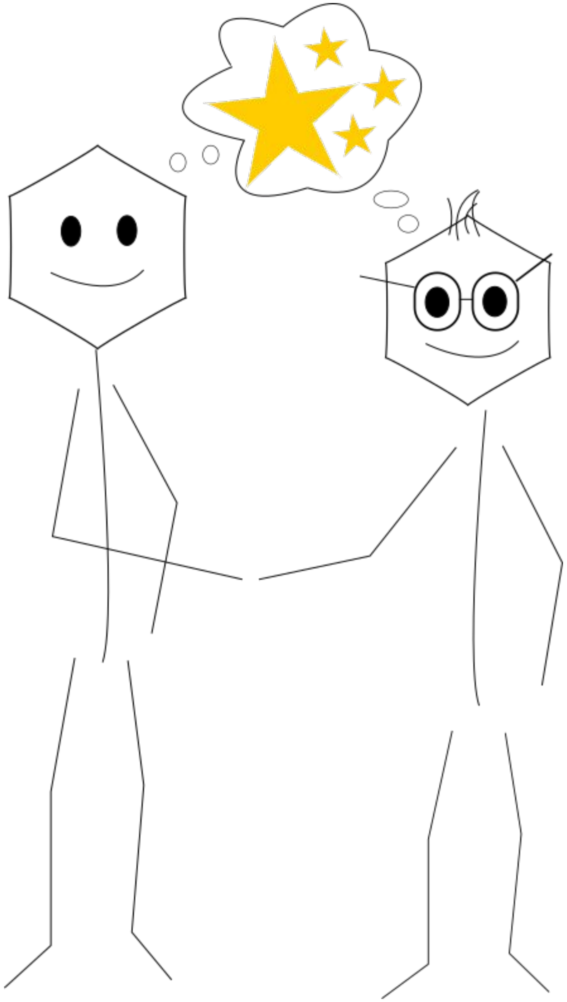
Typical unit test with dependency





Typical unit test with dependency





Integration tests

a/tests/testthat/test-pkgb.R

```
test_that(
  “Preps input correctly for b”,
 {
  expect_error(fxb(fxa(1)), NA)
 }
)
```

b/tests/testthat/test-pkga.R

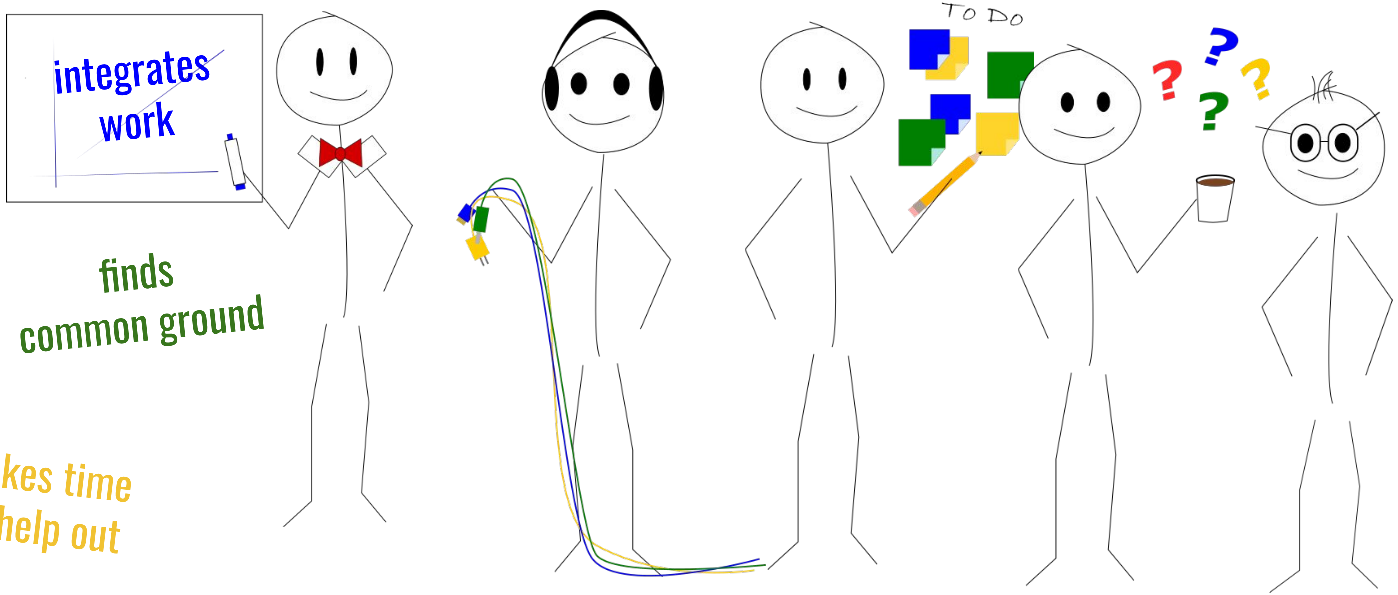
```
test_that(
  “Receives input correctly from a”,
 {
  expect_error(fxb(fxa(1)), NA)
 }
)
```

clear communication

flexible & open to feedback

navigates internal roadblocks

independent, but dependable



integrates work

finds common ground

gives good advice

wise & experienced

makes time to help out

fast & accurate

promotes good practices

trustworthy

reduces frustration & stress

problem-solver