

# Getting more out of dplyr

sample code

Suzan Baert

**Getting more out of dplyr**

**Code examples**

## The variables of this survey data set

A large mess.... (and this is a really short survey...)

```
glimpse(raw_surveydata)
```

[illegible]

# The variables of this survey data set

(Page2)

```
glimpse(raw_surveydata[18:29])
```

```
## Observations: 126
## Variables: 12
## $ `Specifically about dplyr: can you list below all dplyr functions you can spontaneously think of right
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `How many years have you been using R? Please enter the number of years below. If less than a year, you
## $ `In which continent do you currently live?`
## $ `Which generation do you belong to?`
## $ `Which month do you celebrate your birthday?`
```

# Current variable names

To fit on the screen...

```
raw_surveydata_short <- select(raw_surveydata, 1:4, 18:20, 25:29)
glimpse(raw_surveydata_short)
```

```
## Observations: 126
## Variables: 12
## $ Timestamp
## $ `Thinking about the last month or so, what are the packages you loaded the most when using R? Please an
## $ `I've taken a set of frequently downloaded packages (as per cran logs on Harry Potter's most recent bir
## $ `I've taken a set of frequently downloaded packages (as per cran logs on Harry Potter's most recent bir
## $ `Specifically about dplyr: can you list below all dplyr functions you can spontaneously think of right
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `Below are some quotes taken from brainyquote.com or from my private collection. Can you indicate how m
## $ `How many years have you been using R? Please enter the number of years below. If less than a year, you
## $ `In which continent do you currently live?`
## $ `Which generation do you belong to?`
## $ `Which month do you celebrate your birthday?`
```

# Part 1

## Modifying variable names

# dpplr refresher:

`select()` only keeps the variables you specify.

```
raw_surveydata_short %>%  
  select(`Which generation do you belong to`,  
        `Which month do you celebrate your birthday?`) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 2  
## $ `Which generation do you belong to`      <chr> "Millenials ", "...  
## $ `Which month do you celebrate your birthday?` <chr> "July", "June", ...
```

# dplyr refresher:

You can rename a variable inside your select() statement

```
raw_surveydata_short %>%  
  select(generation = `Which generation do you belong to`,  
         month_born = `Which month do you celebrate your birthday?`) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 2  
## $ generation <chr> "Millenials ", "Generation X ", "Millenials ", "Gen...  
## $ month_born <chr> "July", "June", "June", "January", "July", "March",...
```



**`select_all(data, function_y)`**

select and rename all variable names using function\_y

# Select\_all to modify all variable names with tolower()

```
raw_surveydata_short %>%  
  select_all(tolower) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 12  
## $ timestamp  
## $ `thinking about the last month or so, what are the packages you loaded the most when using r? please an  
## $ `i've taken a set of frequently downloaded packages (as per cran logs on harry potter's most recent bir  
## $ `i've taken a set of frequently downloaded packages (as per cran logs on harry potter's most recent bir  
## $ `specifically about dplyr: can you list below all dplyr functions you can spontaneously think of right  
## $ `below are some quotes taken from brainyquote.com or from my private collection. can you indicate how m  
## $ `below are some quotes taken from brainyquote.com or from my private collection. can you indicate how m  
## $ `below are some quotes taken from brainyquote.com or from my private collection. can you indicate how m  
## $ `how many years have you been using r? please enter the number of years below. if less than a year, you  
## $ `in which continent do you currently live?`  
## $ `which generation do you belong to?`  
## $ `which month do you celebrate your birthday?`
```

# Select\_all to modify all variable names with tolower()

Note that the observations itself are not modified.

```
raw_surveydata_short %>%  
  select_all(tolower)
```

```
## # A tibble: 126 x 12  
##   timestamp `thinking about the~` `i've taken a set ~` `i've taken a set ~`  
##   <chr>      <chr>                <fct>                <fct>  
## 1 2018/08/1~ dplyr, readxl, json~ Never                Never  
## 2 2018/08/1~ tidyverse, broom, l~ Never                Never  
## 3 2018/08/1~ Tidyverse, recipes,~ Usually              About half the time  
## 4 2018/08/1~ <NA>                        Seldom               About half the time  
## 5 2018/08/1~ Ggplot2, reshape2, ~ Never                Seldom  
## 6 2018/08/1~ Tidyverse, here, pl~ Seldom               Never  
## 7 2018/08/1~ Tidyverse, odbc, db~ Usually              Usually  
## 8 2018/08/1~ dplyr, scales, ggre~ Never                Never  
## 9 2018/08/1~ dplyr, ggplot2, kni~ Seldom               Seldom  
## 10 2018/08/1~ Dplyr, purrr, furrr~ Seldom               Seldom  
## # ... with 116 more rows, and 8 more variables: `specifically about dplyr:  
## #   can you list below all dplyr functions you can spontaneously think of  
## #   right now? separate using commas. if you don't use dplyr, leave this  
## #   question blank. and to everyone else, no peeking to documentation or  
## #   scripts please ;)` <chr>, `below are some quotes taken from`
```

# Select\_all to replace spaces with underscores

```
raw_surveydata_short %>%  
  select_all(tolower) %>%  
  select_all(str_replace_all(., " ", "_")) %>%  
  glimpse()
```

Error: .funcs must contain one renaming function, not 12

## Select\_all to replace spaces with underscores

## Making a function on the fly: ~

```
raw_surveydata %>%
  select_all(tolower) %>%
  select_all(~str_replace_all(., " ", "_")) %>%
  glimpse()
```

[illegible]

## Select\_all to replace spaces with underscores

## Alternative way of making a function on the fly: `functools.partial()`

```
raw_surveydata %>%
  select_all(tolower) %>%
  select_all(funs(str_replace_all(., " ", "_"))) %>%
  glimpse()
```

[illegible]

# Clean up of survey names (1)

```
raw_surveydata %>%  
  select_all(~str_replace(., ".+set of frequently.+\\[", "freq_load_package_")) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 29  
## $ Timestamp  
## $ `Thinking about the last month or so, what are the packages you loaded the most when using R? Please an  
## $ `freq_load_package_Rcpp`  
## $ `freq_load_package_stringi`  
## $ `freq_load_package_ggplot2`  
## $ `freq_load_package_glue`  
## $ `freq_load_package_rlang`  
## $ `freq_load_package_stringr`  
## $ `freq_load_package_dplyr`  
## $ `freq_load_package_pillar`  
## $ `freq_load_package_fansi`  
## $ `freq_load_package_tibble`  
## $ `freq_load_package_utf8`  
## $ `freq_load_package_jsonlite`  
## $ `freq_load_package_curl`  
## $ `freq_load_package_purrr`  
## $ `freq_load_package_data.table`
```

## Clean up of survey names (2)

```
surveydata2_cleanvarnames_s <- raw_surveydata_short %>%  
  select_all(~str_replace(., "Thinking .+", "packages_last_month")) %>%  
  select_all(~str_replace(., ".+set of frequently.+\\[", "freq_load_package_")) %>%  
  select_all(~str_replace(., ".+quotes.+\\(", "quote_")) %>%  
  select_all(~str_replace(., ".+years.+using R.+", "years_R_experience")) %>%  
  select_all(~str_replace(., ".+month .+", "month_born")) %>%  
  select_all(~str_replace(., "Specifically .+", "dplyr_functions_mentioned")) %>%  
  select_all(~str_extract(., "continent")) %>%  
  select_all(~str_extract(., "generation")) %>%  
  select_all(~str_replace(., "\\]", "")) %>%  
  select_all(~str_replace(., "\\)", "")) %>%  
  select_all(~str_replace_all(., " ", "")) %>%  
  select(-Timestamp)
```



# Clean up of survey names

```
glimpse(surveydata2_cleanvarnames_s)
```

```
## Observations: 126
## Variables: 11
## $ packages_last_month      <chr> "dplyr, readxl, jsonlite, tidyr, ggp...
## $ freq_load_package_Rcpp   <fct> Never, Never, Usually, Seldom, Never...
## $ freq_load_package_stringi <fct> Never, Never, About half the time, A...
## $ dplyr_functions_mentioned <chr> "mutate, filter, arrange, select, re...
## $ quote_Yoda               <fct> Slightly disagree, Strongly agree, Sl...
## $ quote_EvelynWaugh        <fct> Slightly agree, Agree nor disagree, ...
## $ quote_GeorgeRRMartin     <fct> Strongly agree, Slightly agree, Agree...
## $ years_R_experience        <dbl> NA, NA, 5, 13, 6, 4, 5, 2, 8, 9, 5, ...
## $ continent                <chr> "Europe", "North America", "Europe",...
## $ generation                <chr> "Millenials ", "Generation X ", "Mil...
## $ month_born                <chr> "July", "June", "June", "January", "...
```

**`select_if(data, condition_x, function_y)`**

select and rename all variables that meet condition\_x, and rename using function\_y

# select\_if for selecting columns based on characteristics

Selecting all numeric columns

```
surveydata2_cleanvarnames_s %>%  
  select_if(is.numeric) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 1  
## $ years_R_experience <dbl> NA, NA, 5, 13, 6, 4, 5, 2, 8, 9, 5, 1, 6, 4...
```

# select\_if

To select and modify, add a function:

```
surveydata2_cleanvarnames_s %>%  
  select_if(is.character, toupper) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 5  
## $ PACKAGES_LAST_MONTH      <chr> "dplyr, readxl, jsonlite, tidyr, ggp...  
## $ DPLYR_FUNCTIONS_MENTIONED <chr> "mutate, filter, arrange, select, re...  
## $ CONTINENT                <chr> "Europe", "North America", "Europe",...  
## $ GENERATION               <chr> "Millenials ", "Generation X ", "Mil...  
## $ MONTH_BORN               <chr> "July", "June", "June", "January", "..."
```

# select\_if

To run a function with arguments, make a function on the fly with ~ or funs()

```
surveydata2_cleanvarnames_s %>%  
  select_if(is.factor, ~str_remove(., ".+_" )) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 5  
## $ Rcpp          <fct> Never, Never, Usually, Seldom, Never, Seldom, U...  
## $ stringi       <fct> Never, Never, About half the time, About half t...  
## $ Yoda          <fct> Slightly disagree, Strongly agree, Slightly agree...  
## $ EvelynWaugh   <fct> Slightly agree, Agree nor disagree, Agree nor d...  
## $ GeorgeRRMartin <fct> Strongly agree, Slightly agree, Agree nor disag...
```

# select\_if built-in conditions

To scoped if\_functions, the following can come in handy:

- is.character
- is.factor
- is.ordered
- is.numeric
- is.integer
- is.double
- is.logical
- lubridate::is.Date
- lubridate::is.POSIXt

**`select_at(data, vars, function_y)`**

select and rename all variables mentioned in vars, and rename using function\_y

# select\_at to select variables based on their names

Specifying which columns inside vars()

```
surveydata2_cleanvarnames_s %>%  
  select_at(vars(continent, generation, month_born), toupper) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 3  
## $ CONTINENT <chr> "Europe", "North America", "Europe", "Europe", "Eur...  
## $ GENERATION <chr> "Millenials ", "Generation X ", "Millenials ", "Gen...  
## $ MONTH_BORN <chr> "July", "June", "June", "January", "July", "March",...
```



# Select helper functions

To select based on partial names:

- `starts_with()`
- `ends_with()`
- `contains()`

# Select helper functions

To select based on partial names:

- `starts_with()`
- `ends_with()`
- `contains()`

To select based on regex names:

- `matches()`

# Select helper functions

To select based on partial names:

- `starts_with()`
- `ends_with()`
- `contains()`

To select based on regex names:

- `matches()`

To select based on pre-identified names:

- `one_of()` or `!!`

# Select helper functions

To select based on pre-identified names:

```
demographics <- c("years_R_experience", "continent", "generation", "month_born")
surveydata2_cleanvarnames_s %>%
  select(!!demographics) %>%
  glimpse()
```

```
## Observations: 126
## Variables: 4
## $ years_R_experience <dbl> NA, NA, 5, 13, 6, 4, 5, 2, 8, 9, 5, 1, 6, 4...
## $ continent          <chr> "Europe", "North America", "Europe", "Europ...
## $ generation         <chr> "Millenials ", "Generation X ", "Millenials...
## $ month_born         <chr> "July", "June", "June", "January", "July", ...
```

# Select helper functions

To select based on pre-identified names:

```
demographics <- c("years_R_experience", "continent", "generation", "month_born")
surveydata2_cleanvarnames %>%
  mutate(demographics = 1) %>%
  select(!!demographics) %>%
  glimpse()
```

```
## Observations: 126
## Variables: 4
## $ years_R_experience <dbl> NA, NA, 5, 13, 6, 4, 5, 2, 8, 9, 5, 1, 6, 4...
## $ continent         <chr> "Europe", "North America", "Europe", "Europ...
## $ generation        <chr> "Millenials ", "Generation X ", "Millenials...
## $ month_born        <chr> "July", "June", "June", "January", "July", ...
```

# select\_at with select\_helpers

You can use any of the select helpers inside vars():

```
surveydata2_cleanvarnames %>%  
  select_at(vars(contains("freq")), ~str_remove(., ".+_" )) %>%  
  glimpse()
```

```
## Observations: 126  
## Variables: 15  
## $ Rcpp      <fct> Never, Never, Usually, Seldom, Never, Seldom, Usual...  
## $ stringi    <fct> Never, Never, About half the time, About half the t...  
## $ ggplot2    <fct> Usually, Always, Usually, Usually, Always, Always, ...  
## $ glue       <fct> Never, Seldom, About half the time, Seldom, Never, ...  
## $ rlang      <fct> Seldom, About half the time, About half the time, N...  
## $ stringr    <fct> About half the time, Always, About half the time, U...  
## $ dplyr      <fct> Always, Always, About half the time, Always, Seldom...  
## $ pillar     <fct> Never, Never, Seldom, Never, Never, Never, Seldom, ...  
## $ fansi      <fct> Never, Never, Seldom, Never, Never, Never, Seldom, ...  
## $ tibble     <fct> Never, Always, About half the time, Never, Never, A...  
## $ utf8       <fct> Never, Seldom, Seldom, Never, Never, Never, Seldom,...  
## $ jsonlite   <fct> Seldom, Usually, Seldom, About half the time, Seldo...  
## $ curl       <fct> Never, Seldom, Seldom, Seldom, Never, Seldom, Seldo...  
## $ purrr      <fct> About half the time, Always, About half the time, N...  
## $ data.table <fct> Never, Never, Seldom, Seldom, Never, Seldom, Seldom...
```

## Part 2

# Transform and summarise

# Extra data wrangling:

```
surveydata3_cleanvartypes <- surveydata2_cleanvarnames %>%  
  separate(packages_last_month, into = paste0("package_mentioned_", 1:12), sep = ",") %>%  
  separate(dplyr_functions_mentioned, into = paste0("dplyr_function_mentioned_", 1:28), sep = ",")  
surveydata3_cleanvartypes_s <- select(surveydata3_cleanvartypes, 1:2, 13, 14, 28:30, 56, 57, 63:66)
```

```
surveydata3_cleanvartypes
```

```
## # A tibble: 126 x 66  
##   package_mentione~ package_mentione~ package_mentione~ package_mentione~  
##   <chr>             <chr>             <chr>             <chr>  
## 1 dplyr             " readxl"          " jsonlite"        " tidyr"  
## 2 tidyverse         " broom"           " lubridate"        " tidytext"  
## 3 Tidyverse         " recipes"         " plotly"           " h2o"  
## 4 <NA>              <NA>              <NA>              <NA>  
## 5 Ggplot2           " reshape2"        " blogdown"         " googlesheets"  
## 6 Tidyverse         " here"            " plotly"           " knitr"  
## 7 Tidyverse         " odbc"            " dbplyr"           " ggmap"  
## 8 dplyr             " scales"          " ggrepel"          " tidyr"  
## 9 dplyr             " ggplot2"         " knitr"            " sf"  
## 10 Dplyr            " purrr"           " furrr"            " ggplot2"  
## # ... with 116 more rows, and 62 more variables:  
## #   package_mentioned_5 <chr>, package_mentioned_6 <chr>,
```



**mutate\_all(data, function\_y)**

**mutate\_if(data, condition\_x, function\_y)**

**mutate\_at(data, vars, function\_y)**

transform observations of multiple/all variables using function\_y

# Scoped mutate functions

```
surveydata3_cleanvartypes_s %>%  
  mutate_all(str_trim) %>%  
  mutate_all(tolower)
```

```
## # A tibble: 126 x 13  
##   package_mentione~ package_mentione~ freq_load_packag~ freq_load_packag~  
##   <chr>             <chr>             <chr>             <chr>  
## 1 dplyr             readxl             never             never  
## 2 tidyverse         broom             never             never  
## 3 tidyverse         recipes           usually           about half the t~  
## 4 <NA>             <NA>             seldom           about half the t~  
## 5 ggplot2           reshape2          never             seldom  
## 6 tidyverse         here             seldom           never  
## 7 tidyverse         odbc             usually           usually  
## 8 dplyr             scales           never             never  
## 9 dplyr             ggplot2          seldom           seldom  
## 10 dplyr            purrr           seldom           seldom  
## # ... with 116 more rows, and 9 more variables:  
## #   dplyr_function_mentioned_1 <chr>, dplyr_function_mentioned_2 <chr>,  
## #   dplyr_function_mentioned_3 <chr>, quote_Yoda <chr>,  
## #   quote_EvelynWaugh <chr>, years_R_experience <chr>, continent <chr>,  
## #   generation <chr>, month_born <chr>
```

# Scoped mutate functions

```
agree_answers <- c("Strongly agree", "Slightly agree")
surveydata3_cleanvartypes %>%
  select(starts_with("quote")) %>% #you don't need this, it's to fit on screen
  mutate_at(vars(starts_with("quote")), ~if_else(. %in% agree_answers, 1, 0))
```

```
## # A tibble: 126 x 7
##   quote_Yoda quote_EvelynWaugh quote_MarkTwain quote_JoeAbercrombie
##   <dbl>         <dbl>         <dbl>         <dbl>
## 1         0         1         0         1
## 2         1         0         0         1
## 3         1         0         1         1
## 4         0         0         1         1
## 5         0         0         1         1
## 6         1         1         0         1
## 7         1         1         0         1
## 8         0         1         1         1
## 9         1         0         0         1
## 10        0         0         1         0
## # ... with 116 more rows, and 3 more variables: quote_DouglasAdams <dbl>,
## #   quote_JimCarrey <dbl>, quote_GeorgeRRMartin <dbl>
```

**summarise\_all(data, function\_y)**

**summarise\_if(data, condition\_x, function\_y)**

**summarise\_at(data, vars, function\_y)**

# Scoped summarise functions

```
agree_answers <- c("Strongly agree", "Slightly agree")
survey_quoterecoded <- surveydata3_cleanvartypes %>%
  mutate_at(vars(contains("quote")), ~if_else(. %in% agree_answers, 1, 0))

survey_quoterecoded %>%
  summarise_at(vars(contains("quote")), mean) %>%
  gather(variable, mean)
```

```
## # A tibble: 7 x 2
##   variable      mean
##   <chr>      <dbl>
## 1 quote_Yoda    0.460
## 2 quote_EvelynWaugh 0.325
## 3 quote_MarkTwain  0.579
## 4 quote_JoeAbercrombie 0.706
## 5 quote_DouglasAdams 0.548
## 6 quote_JimCarrey  0.611
## 7 quote_GeorgeRRMartin 0.556
```

# Scoped summarise functions

By default, `mutate_x` and `summarise_x` overwrite the current variables.  
Add a tag inside `funcs()` to change variable name.

```
survey_quoterecoded %>%  
  summarise_at(vars(contains("quote")), funcs(prop_agree = mean)) %>%  
  gather(variable, mean)
```

```
## # A tibble: 7 x 2  
##   variable      mean  
##   <chr>      <dbl>  
## 1 quote_Yoda_prop_agree 0.460  
## 2 quote_EvelynWaugh_prop_agree 0.325  
## 3 quote_MarkTwain_prop_agree 0.579  
## 4 quote_JoeAbercrombie_prop_agree 0.706  
## 5 quote_DouglasAdams_prop_agree 0.548  
## 6 quote_JimCarrey_prop_agree 0.611  
## 7 quote_GeorgeRRMartin_prop_agree 0.556
```

# Scoped summarizing with scoped group\_by

```
breakout <- c("continent", "generation")
survey_quoterecoded %>%
  group_by_at(vars(!breakout)) %>%
  summarise_at(vars(contains("quote")), mean)
```

```
## # A tibble: 11 x 9
## # Groups:   continent [?]
##   continent      generation      quote_Yoda quote_EvelynWau~ quote_MarkTwain
##   <chr>          <chr>          <dbl>          <dbl>          <dbl>
## 1 Africa        "Millenials "          0          0.667          1
## 2 Asia          "Baby Boomers~         1           0           0
## 3 Asia          "Generation X~         0.5         0.5           1
## 4 Europe        "Generation X~         0.6         0.267         0.667
## 5 Europe        "Millenials "          0.346       0.346         0.615
## 6 North Ameri~ "Baby Boomers~         1           1           0
## 7 North Ameri~ "Generation X~         0.481       0.333         0.593
## 8 North Ameri~ "Millenials "          0.513       0.282         0.538
## 9 Oceania       "Generation X~         0.25        0.5           0
## 10 Oceania      "Millenials "          0.333       0.167         0.667
## 11 South Ameri~ "Millenials "          0.5         0.5           0.5
## # ... with 4 more variables: quote_JoeAbercrombie <dbl>,
## #   quote_DouglasAdams <dbl>, quote_JimCarrey <dbl>,
## #   quote_GeorgeRRMartin <dbl>
```

**filter\_all(data, filter\_criteria)**

**filter\_if(data, condition\_x, filter\_criteria)**

**filter\_at(data, vars, filter\_criteria)**



# scoped filter functions:

Use `any_vars()` if condition needs to be true across 1 or more columns ("OR")

```
surveydata3_cleanvartypes %>%  
  filter_all(any_vars(str_detect(., "jsonlite")))
```

```
## # A tibble: 3 x 66  
##   package_mentioned~ package_mentione~ package_mentione~ package_mentione~  
##   <chr>                <chr>                <chr>                <chr>  
## 1 dplyr                " readxl"                " jsonlite"          " tidyr"  
## 2 Tidyverse            " tidytext"              " lubridate"          " plumber"  
## 3 Tidyverse            jsonlite                  ggplot                readxl  
## # ... with 62 more variables: package_mentioned_5 <chr>,  
## #   package_mentioned_6 <chr>, package_mentioned_7 <chr>,  
## #   package_mentioned_8 <chr>, package_mentioned_9 <chr>,  
## #   package_mentioned_10 <chr>, package_mentioned_11 <chr>,  
## #   package_mentioned_12 <chr>, freq_load_package_Rcpp <fct>,  
## #   freq_load_package_stringi <fct>, freq_load_package_ggplot2 <fct>,  
## #   freq_load_package_glue <fct>, freq_load_package_rlang <fct>,  
## #   freq_load_package_stringr <fct>, freq_load_package_dplyr <fct>,  
## #   freq_load_package_pillar <fct>, freq_load_package_fansi <fct>,  
## #   freq_load_package_tibble <fct>, freq_load_package_utf8 <fct>,  
## #   freq_load_package_jsonlite <fct>, freq_load_package_curl <fct>,  
## #   freq_load_package_purrr <fct>, freq_load_package_data.table <fct>,
```

# scoped filter functions:

Use `all_vars()` if condition needs to be true across all columns ("AND")

```
surveydata3_cleanvartypes %>%  
  filter_all(all_vars(str_detect(., "jsonlite")))
```

```
## # A tibble: 0 x 66  
## # ... with 66 variables: package_mentioned_1 <chr>,  
## #   package_mentioned_2 <chr>, package_mentioned_3 <chr>,  
## #   package_mentioned_4 <chr>, package_mentioned_5 <chr>,  
## #   package_mentioned_6 <chr>, package_mentioned_7 <chr>,  
## #   package_mentioned_8 <chr>, package_mentioned_9 <chr>,  
## #   package_mentioned_10 <chr>, package_mentioned_11 <chr>,  
## #   package_mentioned_12 <chr>, freq_load_package_Rcpp <fct>,  
## #   freq_load_package_stringi <fct>, freq_load_package_ggplot2 <fct>,  
## #   freq_load_package_glue <fct>, freq_load_package_rlang <fct>,  
## #   freq_load_package_stringr <fct>, freq_load_package_dplyr <fct>,  
## #   freq_load_package_pillar <fct>, freq_load_package_fansi <fct>,  
## #   freq_load_package_tibble <fct>, freq_load_package_utf8 <fct>,  
## #   freq_load_package_jsonlite <fct>, freq_load_package_curl <fct>,  
## #   freq_load_package_purrr <fct>, freq_load_package_data.table <fct>,  
## #   dplyr_function_mentioned_1 <chr>, dplyr_function_mentioned_2 <chr>,  
## #   dplyr_function_mentioned_3 <chr>, dplyr_function_mentioned_4 <chr>,  
## #   dplyr_function_mentioned_5 <chr>, dplyr_function_mentioned_6 <chr>,
```

# In short:

**dplyr\*\_all: do something to all variables**

**dplyr\*\_if: do something to a set of variables that were selected based on their characteristics**

**dplyr\*\_at: do something to a set of variables that were selected based on their variable names**

Replace \* with select, rename, mutate, transmute, filter, summarise, group\_by, arrange

**Part 6**

**Miscellaneous**

**Unfortunately time's up**

# Unfortunately time's up

Nifty tricks like `na_if()`, `case_when()`, `recode()`, `near()`, `everything()`, `rowwise()`, `add_count()` .... are for another time. Or for your own exploration.

# Resources

- [dplyr documentation!](#)

# Resources

- [dplyr documentation!](#)

## My tutorials

- [bit.ly/dplyr\\_1](#) : selecting columns
- [bit.ly/dplyr\\_2](#) : transforming observations
- [bit.ly/dplyr\\_3](#) : filtering
- [bit.ly/dplyr\\_4](#) : summarising

## Materials from today

- [bit.ly/dplyr\\_slides](#) : all the material for this talk (as of tomorrow evening)



# Thanks!

Twitter: @suzanbaert  
Blog: [bit.ly/suzan\\_blog](http://bit.ly/suzan_blog)