Tidyverse Create

Dirk Hartog

2023-11-04

Using one or more TidyVerse packages, and any dataset from fivethirtyeight.com or Kaggle, create a programming sample "vignette" that demonstrates how to use one or more of the capabilities of the selected TidyVerse package with your selected dataset.

```
# Load the tidyverse library
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# List all the packages in tidyverse
tidyverse_packages()
```

```
##  [1] "broom"         "conflicted"    "cli"           "dbplyr"
##  [5] "dplyr"         "dtplyr"        "forcats"       "ggplot2"
##  [9] "googledrive"   "googlesheets4" "haven"         "hms"
## [13] "httr"          "jsonlite"      "lubridate"     "magrittr"
## [17] "modelr"        "pillar"        "purrr"         "ragg"
## [21] "readr"         "readxl"        "reprex"        "rlang"
## [25] "rstudioapi"    "rvest"         "stringr"       "tibble"
## [29] "tidyr"         "xml2"          "tidyverse"
```

For this project we will create an example using the googledrive and googlesheets4 packages. These packages are helpful to manage data and files stored in a google drive. We can easily access the data for manipulation and use in R studio for projects.

Reference: https://www.youtube.com/watch?v=Bdvqtb7fsH0

```
#install.packages("googledrive")
#install.packages("googlesheets4")
```

Package: googledrive

```
library(googledrive)
ls("package:googledrive")
```

```
##  [1] "%>%"                   "as_dribble"            "as_id"
##  [4] "as_shared_drive"       "as_team_drive"         "confirm_dribble"
##  [7] "confirm_single_file"   "confirm_some_files"    "do_paginated_request"
## [10] "do_request"            "drive_about"           "drive_api_key"
## [13] "drive_auth"            "drive_auth_config"     "drive_auth_configure"
## [16] "drive_browse"          "drive_cp"              "drive_create"
## [19] "drive_deauth"          "drive_download"        "drive_empty_trash"
## [22] "drive_endpoint"        "drive_endpoints"       "drive_example"
## [25] "drive_example_local"   "drive_example_remote"  "drive_examples_local"
## [28] "drive_examples_remote" "drive_extension"       "drive_fields"
## [31] "drive_find"            "drive_get"             "drive_has_token"
## [34] "drive_link"            "drive_ls"              "drive_mime_type"
## [37] "drive_mkdir"           "drive_mv"              "drive_oauth_app"
## [40] "drive_oauth_client"    "drive_publish"         "drive_put"
## [43] "drive_read_raw"        "drive_read_string"     "drive_rename"
## [46] "drive_reveal"          "drive_rm"              "drive_scopes"
## [49] "drive_share"           "drive_share_anyone"    "drive_token"
## [52] "drive_trash"           "drive_unpublish"       "drive_untrash"
## [55] "drive_update"          "drive_upload"          "drive_user"
## [58] "expose"                "is_dribble"            "is_folder"
## [61] "is_folder_shortcut"    "is_mine"               "is_native"
## [64] "is_parental"           "is_shared_drive"       "is_shortcut"
## [67] "is_team_drive"         "local_drive_quiet"     "no_file"
## [70] "prep_fields"           "request_generate"      "request_make"
## [73] "shared_drive_create"   "shared_drive_find"     "shared_drive_get"
## [76] "shared_drive_rm"       "shared_drive_update"   "shortcut_create"
## [79] "shortcut_resolve"      "single_file"           "some_files"
## [82] "team_drive_create"     "team_drive_find"       "team_drive_get"
## [85] "team_drive_rm"         "team_drive_update"     "with_drive_quiet"
```

1. Our first task will be to download a file from our drive.

Run googledrive::drive_auth() to connect with your google drive. This should open up your google drive asking for your permission to connect to tidyverse

```
drive_auth()
```

```
## ! Using an auto-discovered, cached token.
```

```
##   To suppress this message, modify your code or options to clearly consent to
##   the use of a cached token.
```

```
##   See gargle's "Non-interactive auth" vignette for more details:

##   <https://gargle.r-lib.org/articles/non-interactive-auth.html>

## i The googledrive package is using a cached token for 'diggz84@gmail.com'.
```

Since we are downloading a file from drive, we want to set our local directory to where the file will be kept

```
setwd("/Users/dirkhartog/Desktop/CUNY_MSDS/DATA_607/Tidyverse")
```

Use google::drive_find() and set type to "folder" and use the q parameter to find the folder you want to work with. This will return a table with any folder that matches the pattern in the q parameter

```
drive_find(type = "folder", q = "name contains 'Tidy'")
```

```
## # A dribble: 1 x 3
##   name      id                                drive_resource
##   <chr>     <drv_id>                          <list>
## 1 Tidyverse 1Px9cboC17Qd9a8nulg1yADlYJWL6r_VY <named list [34]>
```

You can copy and past the id of the folder you want to use into google::drive_ls() to find the id of the file you want to upload.

```
drive_ls(path = as_id("1Px9cboC17Qd9a8nulg1yADlYJWL6r_VY"))
```

```
## # A dribble: 3 x 3
##   name                id       drive_resource
##   <chr>               <drv_id> <list>
## 1 Largest-art-museums 1YW__n_~ <named list [36]>
## 2 marvel_movies.csv   178m3yK~ <named list [43]>
## 3 mental_health_data  1QiHsEc~ <named list [36]>
```

Use googledrive::drive_get() and set the parameter path using as_id() to id of the file you want from that folder. You can save this to a variable called target_file that wil be used to download the file.

```
target_file <- drive_get(path = as_id("178m3yKYkiLnggmNLNyBWUS2joVIL7zBO"))
```

Use googledrive::drive_download() to download the csv file from google drive.

- The file parameter is set to our target file from google drive.
- type is set to the type of desired export
- path is set to the path to the local directory on your computer or if we are already pointing to the directory just need to supply the file name we want to save it as.

```
drive_download(file = target_file,
               type = "csv",
               path = "marvel_movies.csv",
               overwrite = TRUE)
```

```
## ! Ignoring `type`. Only consulted for native Google file types.
```

```
##   MIME type of ‘file‘: ’mime_type’.

## File downloaded:

## * ’marvel_movies.csv’ <id: 178m3yKYkiLnggmNLNyBWUS2joVIL7zBO>

## Saved locally as:

## * ’marvel_movies.csv’
```

in this code chunk we check that the file is in your local directory and read it into R studio to clean and transform the data. Finally we write the file to our local directory.

```
list.files()
```

```
## [1] "marvel_movies.csv"        "mcu_films.csv"
## [3] "MentalhealthData.csv"     "tidyverse_create_final.R"
## [5] "tidyverse_create.pdf"     "tidyverse_create.R"
## [7] "tidyverse_create.Rmd"
```

```
movies <- read_csv("marvel_movies.csv")
```

```
## Rows: 30 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): film, category, % budget recovered, critics % score, audience % sc...
## dbl  (8): worldwide gross ($m), budget, domestic gross ($m), international g...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(movies)
```

```
## # A tibble: 6 x 19
##   film    category ‘worldwide gross ($m)‘ ‘% budget recovered‘ ‘critics % score‘
##   <chr>   <chr>                    <dbl> <chr>                <chr>
## 1 Ant-Man Ant-Man                    518 398%                 83%
## 2 Ant-Ma~ Ant-Man                    623 479%                 87%
## 3 Avenge~ Avengers                  1395 382%                 76%
## 4 Avenge~ Avengers                  2797 699%                 94%
## 5 Avenge~ Avengers                  2048 683%                 85%
## 6 Black ~ Black P~                  1336 668%                 96%
## # i 14 more variables: ‘audience % score‘ <chr>,
## #   ‘audience vs critics % deviance‘ <chr>, budget <dbl>,
## #   ‘domestic gross ($m)‘ <dbl>, ‘international gross ($m)‘ <dbl>,
## #   ‘opening weekend ($m)‘ <dbl>, ‘second weekend ($m)‘ <dbl>,
## #   ‘1st vs 2nd weekend drop off‘ <chr>, ‘% gross from opening weekend‘ <dbl>,
## #   ‘% gross from domestic‘ <chr>, ‘% gross from international‘ <chr>,
## #   ‘% budget opening weekend‘ <chr>, year <dbl>, source <chr>
```

```r
names(movies) <- c("Film", "Category", "Worldwide_gross_mil", "Budget_recoverd_pct",
                   "Critic_score_pct", "Audience_score_pct", "Audience_critic_diff",
                   "Budget", "Domestic_gross_mil", "International_gross_mil",
                   "Opening_weekend_mil", "Second_weekend_mil", "Weekend_drop",
                   "Pct_gross_opening", "Pct_gross_domestic", "Pct_gross_int",
                   "Pct_budget_opening", "Year", "Source")
library(stringr)

cols_to_use <- c("Budget_recoverd_pct", "Critic_score_pct", "Audience_score_pct",
                 "Audience_critic_diff", "Weekend_drop", "Pct_gross_opening",
                 "Pct_gross_domestic", "Pct_gross_int", "Pct_budget_opening")

movies$Budget_recoverd_pct <- as.double(str_extract(movies$Budget_recoverd_pct, "\\d+"))
movies$Critic_score_pct <- as.double(str_extract(movies$Critic_score_pct, "\\d+"))
movies$Audience_score_pct <- as.double(str_extract(movies$Audience_score_pct, "\\d+"))
movies$Audience_critic_diff <- as.double(str_extract(movies$Audience_critic_diff, "\\d+"))
movies$Weekend_drop <- as.double(str_extract(movies$Weekend_drop, "\\d+"))
movies$Pct_gross_opening <- as.double(str_extract(movies$Pct_gross_opening, "\\d+"))
movies$Pct_gross_domestic <- as.double(str_extract(movies$Pct_gross_domestic, "\\d+"))
movies$Pct_gross_int <- as.double(str_extract(movies$Pct_gross_int, "\\d+"))
movies$Pct_budget_opening <- as.double(str_extract(movies$Pct_budget_opening, "\\d+"))

glimpse(movies)
```

```
## Rows: 30
## Columns: 19
## $ Film                   <chr> "Ant-Man", "Ant-Man & The Wasp", "Avengers: Ag~
## $ Category               <chr> "Ant-Man", "Ant-Man", "Avengers", "Avengers", ~
## $ Worldwide_gross_mil    <dbl> 518, 623, 1395, 2797, 2048, 1336, 855, 379, 37~
## $ Budget_recoverd_pct    <dbl> 398, 479, 382, 699, 683, 668, 342, 190, 264, 4~
## $ Critic_score_pct       <dbl> 83, 87, 76, 94, 85, 96, 84, 79, 79, 90, 90, 79~
## $ Audience_score_pct     <dbl> 85, 80, 82, 90, 91, 79, 94, 80, 75, 89, 92, 45~
## $ Audience_critic_diff   <dbl> 2, 7, 6, 4, 6, 17, 10, 1, 4, 1, 2, 34, 3, 3, 2~
## $ Budget                 <dbl> 130.0, 130.0, 365.0, 400.0, 300.0, 200.0, 250.~
## $ Domestic_gross_mil     <dbl> 180, 216, 459, 858, 678, 700, 453, 183, 176, 4~
## $ International_gross_mil <dbl> 338, 406, 936, 1939, 1369, 636, 401, 196, 193,~
## $ Opening_weekend_mil    <dbl> 57.0, 75.8, 191.0, 357.0, 257.0, 202.0, 181.0,~
## $ Second_weekend_mil     <dbl> 24.0, 29.0, 77.0, 147.0, 114.0, 111.0, 66.0, 2~
## $ Weekend_drop           <dbl> 58, 62, 60, 59, 56, 45, 64, 68, 62, 59, 57, 56~
## $ Pct_gross_opening      <dbl> 31, 35, 41, 41, 38, 28, 48, 43, 36, 43, 36, 35~
## $ Pct_gross_domestic     <dbl> 34, 34, 32, 30, 33, 52, 53, 48, 47, 35, 36, 37~
## $ Pct_gross_int          <dbl> 65, 65, 67, 69, 66, 47, 46, 51, 52, 64, 63, 62~
## $ Pct_budget_opening     <dbl> 43, 58, 52, 89, 85, 101, 72, 40, 46, 71, 55, 8~
## $ Year                   <dbl> 2015, 2018, 2015, 2019, 2018, 2018, 2022, 2021~
## $ Source                 <chr> "https://www.the-numbers.com/movie/Ant-Man#tab~
```

```r
write_csv(movies, "marvel_movies.csv")
```

2. Re-upload it back to our drive

Use googledrive::drive_get() to target which directory in your google drive you want to upload the file to.

```
td <- drive_get(path = as_id("1Px9cboC17Qd9a8nulg1yADlYJWL6r_VY"))
```

Use google::drive_upload() to upload a file to your google drive. + We need to create a character vector with the file name from the local local directory you wish to upload. + Set the file you will to upload and assign it to media + Set your google drive directory to path. + Set the file name + Set the type to speadsheet that will save it as a google sheet *in this exmaple we will use a different file name to confirm that it uploaded*

```
file <- "marvel_movies.csv"

drive_upload(media = file, path = as_id(td),
             name = "marvel_movies2.csv",
             type = "spreadsheet")
```

```
## Local file:

## * 'marvel_movies.csv'

## Uploaded into Drive file:

## * 'marvel_movies2' <id: 1br3MDDlA37mLOLN59H1bHY4JszuBacJDPdMdvtG33dc>

## With MIME type:

## * 'application/vnd.google-apps.spreadsheet'
```

Finally we can check the drive directory to see if the file was uploaded

```
drive_ls(path = as_id("1Px9cboC17Qd9a8nulg1yADlYJWL6r_VY"))
```

```
## # A dribble: 4 x 3
##   name                id       drive_resource
##   <chr>               <drv_id> <list>
## 1 marvel_movies2      1br3MDD~ <named list [36]>
## 2 Largest-art-museums 1YW__n_~ <named list [36]>
## 3 marvel_movies.csv   178m3yK~ <named list [43]>
## 4 mental_health_data  1QiHsEc~ <named list [36]>
```

Package: googlesheests4

With googlesheets4 we don't need to write the downloaded file to our local directory and can read the file right from our google drive. In this example we will look at reading a **google sheet** (not a csv file) in from google drive, do some data transformation and then upload it back to our drive.

Use googlesheets4::gs4_auth() to connect to our drive

```
library(googlesheets4)
```

```
##
## Attaching package: 'googlesheets4'
```

```
## The following objects are masked from 'package:googledrive':
##
##     request_generate, request_make
```

```
gs4_auth()
```

```
## ! Using an auto-discovered, cached token.

##   To suppress this message, modify your code or options to clearly consent to
##   the use of a cached token.

##   See gargle's "Non-interactive auth" vignette for more details:

##   <https://gargle.r-lib.org/articles/non-interactive-auth.html>

## i The googlesheets4 package is using a cached token for 'diggz84@gmail.com'.
```

Use google::drive_find() and set type to "folder" and use the q parameter to find the folder you want to work with. This will return a table with any folder that matches the pattern in the q parameter

```
drive_find(type = "folder", q = "name contains 'Tidy'")
```

```
## # A dribble: 1 x 3
##   name      id                                drive_resource
##   <chr>     <drv_id>                          <list>
## 1 Tidyverse 1Px9cboC17Qd9a8nulg1yADlYJWL6r_VY <named list [34]>
```

You can copy and past the id of the folder you want to use into google::drive_ls() to find the id of the file you want to upload.

```
drive_ls(path = as_id("1Px9cboC17Qd9a8nulg1yADlYJWL6r_VY"))
```

```
## # A dribble: 4 x 3
##   name                id       drive_resource
##   <chr>               <drv_id> <list>
## 1 marvel_movies2      1br3MDD~ <named list [36]>
## 2 Largest-art-museums 1YW__n_~ <named list [36]>
## 3 marvel_movies.csv   178m3yK~ <named list [43]>
## 4 mental_health_data  1QiHsEc~ <named list [36]>
```

Get the target directory/sheet we want to work with. We will use the googledrive::drive_get() to get the file.

```
file2 <- drive_get(path = as_id("1YW__n_d1OYj9Z7YejQiG-bgUqXS4D6v6zKElB-90iVM"))
```

Use googlesheets4::read_sheet() that reads the google sheet as a the data frame directly if we want to work the file in R. The function read_sheet() read some of the columns in as a list so we need to do one extra step of unlisting those columns.

```
museumsdf <- read_sheet(file2)
```

```
## v Reading from "Largest-art-museums".
```

```
## v Range 'Largest-art-museums'.
```

```
head(museumsdf)
```

```
## # A tibble: 6 x 6
##   Name               City  Country Gallery space in m2 ~1 Gallery space in sq ~2
##   <chr>              <chr> <chr>   <list>                  <list>
## 1 British Museum     Lond~ United~ <dbl [1]>               <chr [1]>
## 2 Louvre             Paris France  <chr [1]>               <chr [1]>
## 3 State Hermitage M~ St. ~ Russia  <chr [1]>               <chr [1]>
## 4 National Museum o~ Beij~ China   <chr [1]>               <chr [1]>
## 5 Metropolitan Muse~ New ~ United~ <chr [1]>               <chr [1]>
## 6 Museo del Prado    Madr~ Spain   <chr [1]>               <chr [1]>
## # i abbreviated names: 1: 'Gallery space in m2 (sq ft)',
## #   2: 'Gallery space in sq ft'
## # i 1 more variable: 'Year established' <list>
```

```
colnames(museumsdf)
```

```
## [1] "Name"                        "City"
## [3] "Country"                     "Gallery space in m2 (sq ft)"
## [5] "Gallery space in sq ft"      "Year established"
```

```
museumsdf$'Gallery space in m2 (sq ft)' <- NULL

museumsdf$'Gallery space in sq ft' <- str_extract_all(museumsdf$'Gallery space in sq ft', "(\\d+.\\d{3}

museumsdf <- museumsdf %>% unnest_wider(col = 'Gallery space in sq ft', names_sep = ".")

names(museumsdf) <- c("Name", "City", "Country", "Sq_m", "Sq_ft", "Year_est")

glimpse(museumsdf)
```

```
## Rows: 112
## Columns: 6
## $ Name     <chr> "British Museum", "Louvre", "State Hermitage Museum", "Nation~
## $ City     <chr> "London", "Paris", "St. Petersburg", "Beijing", "New York Cit~
## $ Country  <chr> "United Kingdom", "France", "Russia", "China", "United States~
## $ Sq_m     <chr> "92,000", "72,735", "66,842", "65,000", "58,820", "47,700", "~
## $ Sq_ft    <chr> "990,000", "782,910", "719,480", "700,000", "633,100", "513,0~
## $ Year_est <list> 1753, 1792, 1764, 1959, 1870, 1819, 1506, 1872, 1964, 1852, ~
```

We can write the sheet back to google drive using googlesheets4::range_write()

```
range_write(file2, # URL with the file id
            museumsdf, # A data frame
            sheet = NULL, # sting name of the sheet or numerical position
            range = NULL, # cell range
            col_names = TRUE,
            reformat = TRUE)
```

```
## v Editing "Largest-art-museums".
```

```
## v Writing to sheet 'Largest-art-museums'.
```

We can re load our sheet to check that our changes have over written our exisiting sheet

```
file2 <- drive_get(path = as_id("1YW__n_d1OYj9Z7YejQiG-bgUqXS4D6v6zKElB-9OiVM"))

museumsdf <- read_sheet(file2)
```

```
## v Reading from "Largest-art-museums".
```

```
## v Range 'Largest-art-museums'.
```

```
museumsdf$Year_est <- unlist(museumsdf$Year_est)

head(museumsdf)
```

```
## # A tibble: 6 x 6
##   Name                        City           Country        Sq_m   Sq_ft Year_est
##   <chr>                       <chr>          <chr>          <chr>  <chr> <chr>
## 1 British Museum              London         United Kingdom 92,000 990,~ 1753
## 2 Louvre                      Paris          France         72,735 782,~ 1792
## 3 State Hermitage Museum      St. Petersburg Russia         66,842 719,~ 1764
## 4 National Museum of China    Beijing        China          65,000 700,~ 1959
## 5 Metropolitan Museum of Art New York City   United States  58,820 633,~ 1870
## 6 Museo del Prado             Madrid         Spain          47,700 513,~ 1819
```