

Live Session 2: Getting data from unusual sources

Zak Varty

Data from the clipboard

Sometimes you need a fast, hacky way to read data into R. You can read data from the clipboard using the `{datapasta}` package. This is bad in terms of reproducibility but can sometimes be helpful to know about.

The package does well when reading from html and gives a reasonable starting point from pdf.

Tasks

1. Use `{datapasta}` to load the table from `example-table.html` into R as a a tribble (row-wise specification of a tibble).

Solution:

```
# Check the documentation
?datapasta::tribble_paste()
```

- Open html file, highlight and copy table
- When running `tribble_paste()` need to be sure cursor is somewhere sensible (new code block).
- The L's here indicate that these are integer not double.

```
datapasta::tribble_paste()
```

```
example_tibble <- tibble::tribble(
  ~age, ~gender, ~height,
  45L,  "Male",  161L,
```

```

97L, "Female", 157L,
39L, "Female", 150L,
26L, "Female", 150L,
22L, "Male", 159L,
55L, "Female", 147L,
33L, "Female", 155L,
21L, "Male", 165L,
87L, "Male", 152L,
96L, "Male", 164L,
95L, "Male", 159L,
31L, "Female", 150L,
73L, "Male", 168L,
79L, "Female", 166L,
21L, "Male", 186L,
21L, "Female", 152L,
38L, "Male", 186L,
57L, NA, 148L,
73L, "Male", 177L,
84L, "Female", 185L,
22L, "Male", 170L,
83L, "Female", 153L,
64L, "Male", 170L,
57L, "Male", 188L,
65L, "Male", 159L,
20L, "Female", 174L,
58L, NA, 173L,
89L, "Female", 163L,
49L, "Male", 170L,
59L, "Female", 155L
)

```

2. Use `{datapasta}` to load the table of [Wimbledon Champions by Country](#) into R as a data frame. What goes wrong when you try to do the same for the [Women's world record rowing times on water](#) and the [indoor rowing world record times](#)?

Solution:

- `datapasta::df_paste()` works in a similar way
- The output is not quite so pretty because data frames are specified by column rather than by row.

```
datapasta::df_paste()
```

```

data.frame(
  stringsAsFactors = FALSE,
  check.names = FALSE,
  Country = c("Great Britain (GBR)[e][k]",
              "United States (USA)", "Australia (AUS)",
              "Switzerland (SUI)", "France (FRA)", "Serbia (SRB)", "Sweden (SWE)",
              "Germany (GER)[l]", "New Zealand (NZL)", "Spain (ESP)",
              "Croatia (CRO)", "Czechoslovakia (TCH)[m]", "Egypt (EGY)",
              "Netherlands (NED)"),
  Amateur.Era = c(35L, 18L, 15L, 0L, 7L, 0L, 0L,
                  0L, 4L, 1L, 0L, 0L, 1L, 0L),
  Open.Era = c(2L, 15L, 6L, 8L, 0L, 7L, 7L, 4L, 0L, 3L, 1L, 1L, 0L, 1L),
  `All-time` = c(37L, 33L, 21L, 8L, 7L, 7L, 7L,
                 4L, 4L, 4L, 1L, 1L, 1L, 1L),
  First.title = c(1877L, 1920L, 1907L, 2003L,
                  1924L, 2011L, 1976L, 1985L, 1910L, 1966L, 2001L, 1973L,
                  1954L, 1996L),
  Last.title = c(2016L, 2000L, 2002L, 2017L,
                 1946L, 2022L, 1990L, 1991L, 1913L, 2023L, 2001L, 1973L,
                 1954L, 1996L)
)

```

	Country	Amateur.Era	Open.Era	All-time	First.title
1	Great Britain (GBR)[e][k]	35	2	37	1877
2	United States (USA)	18	15	33	1920
3	Australia (AUS)	15	6	21	1907
4	Switzerland (SUI)	0	8	8	2003
5	France (FRA)	7	0	7	1924
6	Serbia (SRB)	0	7	7	2011
7	Sweden (SWE)	0	7	7	1976
8	Germany (GER)[l]	0	4	4	1985
9	New Zealand (NZL)	4	0	4	1910
10	Spain (ESP)	1	3	4	1966
11	Croatia (CRO)	0	1	1	2001
12	Czechoslovakia (TCH)[m]	0	1	1	1973
13	Egypt (EGY)	1	0	1	1954
14	Netherlands (NED)	0	1	1	1996
	Last.title				
1	2016				
2	2000				
3	2002				
4	2017				

5	1946
6	2022
7	1990
8	1991
9	1913
10	2023
11	2001
12	1973
13	1954
14	1996

3. Use `{datapasta}` to load the standard normal distribution table into R. (note this will require some manual editing and does not work so well on Linux.)

Solution:

```
datapasta::tribble_paste()
```

```
tibble::tribble(
  ~z...00..01..02..03..04..05..06..07..08..09,
  "0.0 .0000 .0040 .0080 .0120 .0160 .0199 .0239 .0279 .0319 .0359",
  "0.1 .0398 .0438 .0478 .0517 .0557 .0596 .0636 .0675 .0714 .0753",
  "0.2 .0793 .0832 .0871 .0910 .0948 .0987 .1026 .1064 .1103 .1141",
  "0.3 .1179 .1217 .1255 .1293 .1331 .1368 .1406 .1443 .1480 .1517",
  "0.4 .1554 .1591 .1628 .1664 .1700 .1736 .1772 .1808 .1844 .1879",
  "0.5 .1915 .1950 .1985 .2019 .2054 .2088 .2123 .2157 .2190 .2224",
  "0.6 .2257 .2291 .2324 .2357 .2389 .2422 .2454 .2486 .2517 .2549",
  "0.7 .2580 .2611 .2642 .2673 .2704 .2734 .2764 .2794 .2823 .2852",
  "0.8 .2881 .2910 .2939 .2967 .2995 .3023 .3051 .3078 .3106 .3133",
  "0.9 .3159 .3186 .3212 .3238 .3264 .3289 .3315 .3340 .3365 .3389",
  "1.0 .3413 .3438 .3461 .3485 .3508 .3531 .3554 .3577 .3599 .3621",
  "1.1 .3643 .3665 .3686 .3708 .3729 .3749 .3770 .3790 .3810 .3830",
  "1.2 .3849 .3869 .3888 .3907 .3925 .3944 .3962 .3980 .3997 .4015",
  "1.3 .4032 .4049 .4066 .4082 .4099 .4115 .4131 .4147 .4162 .4177",
  "1.4 .4192 .4207 .4222 .4236 .4251 .4265 .4279 .4292 .4306 .4319",
  "1.5 .4332 .4345 .4357 .4370 .4382 .4394 .4406 .4418 .4429 .4441",
  "1.6 .4452 .4463 .4474 .4484 .4495 .4505 .4515 .4525 .4535 .4545",
  "1.7 .4554 .4564 .4573 .4582 .4591 .4599 .4608 .4616 .4625 .4633",
  "1.8 .4641 .4649 .4656 .4664 .4671 .4678 .4686 .4693 .4699 .4706",
  "1.9 .4713 .4719 .4726 .4732 .4738 .4744 .4750 .4756 .4761 .4767",
  "2.0 .4772 .4778 .4783 .4788 .4793 .4798 .4803 .4808 .4812 .4817",
  "2.1 .4821 .4826 .4830 .4834 .4838 .4842 .4846 .4850 .4854 .4857",
```

```
"2.2 .4861 .4864 .4868 .4871 .4875 .4878 .4881 .4884 .4887 .4890",
"2.3 .4893 .4896 .4898 .4901 .4904 .4906 .4909 .4911 .4913 .4916",
"2.4 .4918 .4920 .4922 .4925 .4927 .4929 .4931 .4932 .4934 .4936",
"2.5 .4938 .4940 .4941 .4943 .4945 .4946 .4948 .4949 .4951 .4952",
"2.6 .4953 .4955 .4956 .4957 .4959 .4960 .4961 .4962 .4963 .4964",
"2.7 .4965 .4966 .4967 .4968 .4969 .4970 .4971 .4972 .4973 .4974",
"2.8 .4974 .4975 .4976 .4977 .4977 .4978 .4979 .4979 .4980 .4981",
"2.9 .4981 .4982 .4982 .4983 .4984 .4984 .4985 .4985 .4986 .4986",
"3.0 .4987 .4987 .4987 .4988 .4988 .4989 .4989 .4989 .4990 .4990",
"3.1 .4990 .4991 .4991 .4991 .4992 .4992 .4992 .4992 .4993 .4993",
"3.2 .4993 .4993 .4994 .4994 .4994 .4994 .4994 .4995 .4995 .4995",
"3.3 .4995 .4995 .4995 .4996 .4996 .4996 .4996 .4996 .4996 .4997",
"3.4 .4997 .4997 .4997 .4997 .4997 .4997 .4997 .4997 .4997 .4998",
"3.5 .4998 .4998 .4998 .4998 .4998 .4998 .4998 .4998 .4998 .4998"
)
```

```
# A tibble: 36 x 1
  z..00..01..02..03..04..05..06..07..08..09
  <chr>
1 0.0 .0000 .0040 .0080 .0120 .0160 .0199 .0239 .0279 .0319 .0359
2 0.1 .0398 .0438 .0478 .0517 .0557 .0596 .0636 .0675 .0714 .0753
3 0.2 .0793 .0832 .0871 .0910 .0948 .0987 .1026 .1064 .1103 .1141
4 0.3 .1179 .1217 .1255 .1293 .1331 .1368 .1406 .1443 .1480 .1517
5 0.4 .1554 .1591 .1628 .1664 .1700 .1736 .1772 .1808 .1844 .1879
6 0.5 .1915 .1950 .1985 .2019 .2054 .2088 .2123 .2157 .2190 .2224
7 0.6 .2257 .2291 .2324 .2357 .2389 .2422 .2454 .2486 .2517 .2549
8 0.7 .2580 .2611 .2642 .2673 .2704 .2734 .2764 .2794 .2823 .2852
9 0.8 .2881 .2910 .2939 .2967 .2995 .3023 .3051 .3078 .3106 .3133
10 0.9 .3159 .3186 .3212 .3238 .3264 .3289 .3315 .3340 .3365 .3389
# i 26 more rows
```

The default output from Acrobat Reader interprets this as one character column named `z..00..01..02..03..04..05..06..07..08..09`. This can be easily fixed with a little bit of [multi-cursor magic](#).

```
tibble::tribble(
  ~z , ~`.0.01`, ~`.0.002`, ~`.0.003`, ~`.0.004`, ~`.0.005`, ~`.0.006`, ~`.0.007`, ~`.0.008`, ~`.0.009`,
  0.0, 0.0000, 0.0040, 0.0080, 0.0120, 0.0160, 0.0199, 0.0239, 0.0279, 0.0319, 0.0359,
  0.1, 0.0398, 0.0438, 0.0478, 0.0517, 0.0557, 0.0596, 0.0636, 0.0675, 0.0714, 0.0753,
  0.2, 0.0793, 0.0832, 0.0871, 0.0910, 0.0948, 0.0987, 0.1026, 0.1064, 0.1103, 0.1141,
  0.3, 0.1179, 0.1217, 0.1255, 0.1293, 0.1331, 0.1368, 0.1406, 0.1443, 0.1480, 0.1517,
```

```
0.4, 0.1554, 0.1591, 0.1628, 0.1664, 0.1700, 0.1736, 0.1772, 0.1808, 0.1844, 0.1879,
0.5, 0.1915, 0.1950, 0.1985, 0.2019, 0.2054, 0.2088, 0.2123, 0.2157, 0.2190, 0.2224,
0.6, 0.2257, 0.2291, 0.2324, 0.2357, 0.2389, 0.2422, 0.2454, 0.2486, 0.2517, 0.2549,
0.7, 0.2580, 0.2611, 0.2642, 0.2673, 0.2704, 0.2734, 0.2764, 0.2794, 0.2823, 0.2852,
0.8, 0.2881, 0.2910, 0.2939, 0.2967, 0.2995, 0.3023, 0.3051, 0.3078, 0.3106, 0.3133,
0.9, 0.3159, 0.3186, 0.3212, 0.3238, 0.3264, 0.3289, 0.3315, 0.3340, 0.3365, 0.3389,
1.0, 0.3413, 0.3438, 0.3461, 0.3485, 0.3508, 0.3531, 0.3554, 0.3577, 0.3599, 0.3621,
1.1, 0.3643, 0.3665, 0.3686, 0.3708, 0.3729, 0.3749, 0.3770, 0.3790, 0.3810, 0.3830,
1.2, 0.3849, 0.3869, 0.3888, 0.3907, 0.3925, 0.3944, 0.3962, 0.3980, 0.3997, 0.4015,
1.3, 0.4032, 0.4049, 0.4066, 0.4082, 0.4099, 0.4115, 0.4131, 0.4147, 0.4162, 0.4177,
1.4, 0.4192, 0.4207, 0.4222, 0.4236, 0.4251, 0.4265, 0.4279, 0.4292, 0.4306, 0.4319,
1.5, 0.4332, 0.4345, 0.4357, 0.4370, 0.4382, 0.4394, 0.4406, 0.4418, 0.4429, 0.4441,
1.6, 0.4452, 0.4463, 0.4474, 0.4484, 0.4495, 0.4505, 0.4515, 0.4525, 0.4535, 0.4545,
1.7, 0.4554, 0.4564, 0.4573, 0.4582, 0.4591, 0.4599, 0.4608, 0.4616, 0.4625, 0.4633,
1.8, 0.4641, 0.4649, 0.4656, 0.4664, 0.4671, 0.4678, 0.4686, 0.4693, 0.4699, 0.4706,
1.9, 0.4713, 0.4719, 0.4726, 0.4732, 0.4738, 0.4744, 0.4750, 0.4756, 0.4761, 0.4767,
2.0, 0.4772, 0.4778, 0.4783, 0.4788, 0.4793, 0.4798, 0.4803, 0.4808, 0.4812, 0.4817,
2.1, 0.4821, 0.4826, 0.4830, 0.4834, 0.4838, 0.4842, 0.4846, 0.4850, 0.4854, 0.4857,
2.2, 0.4861, 0.4864, 0.4868, 0.4871, 0.4875, 0.4878, 0.4881, 0.4884, 0.4887, 0.4890,
2.3, 0.4893, 0.4896, 0.4898, 0.4901, 0.4904, 0.4906, 0.4909, 0.4911, 0.4913, 0.4916,
2.4, 0.4918, 0.4920, 0.4922, 0.4925, 0.4927, 0.4929, 0.4931, 0.4932, 0.4934, 0.4936,
2.5, 0.4938, 0.4940, 0.4941, 0.4943, 0.4945, 0.4946, 0.4948, 0.4949, 0.4951, 0.4952,
2.6, 0.4953, 0.4955, 0.4956, 0.4957, 0.4959, 0.4960, 0.4961, 0.4962, 0.4963, 0.4964,
2.7, 0.4965, 0.4966, 0.4967, 0.4968, 0.4969, 0.4970, 0.4971, 0.4972, 0.4973, 0.4974,
2.8, 0.4974, 0.4975, 0.4976, 0.4977, 0.4977, 0.4978, 0.4979, 0.4979, 0.4980, 0.4981,
2.9, 0.4981, 0.4982, 0.4982, 0.4983, 0.4984, 0.4984, 0.4985, 0.4985, 0.4986, 0.4986,
3.0, 0.4987, 0.4987, 0.4987, 0.4988, 0.4988, 0.4989, 0.4989, 0.4989, 0.4990, 0.4990,
3.1, 0.4990, 0.4991, 0.4991, 0.4991, 0.4992, 0.4992, 0.4992, 0.4992, 0.4993, 0.4993,
3.2, 0.4993, 0.4993, 0.4994, 0.4994, 0.4994, 0.4994, 0.4994, 0.4995, 0.4995, 0.4995,
3.3, 0.4995, 0.4995, 0.4995, 0.4996, 0.4996, 0.4996, 0.4996, 0.4996, 0.4996, 0.4997,
3.4, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4998,
3.5, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998
)
```

API Example - googlesheets4

You might find yourself in a situation where your data are stored in the cloud, for example in google sheets. In one sense this is good: the data are not lost if your laptop breaks and any changes or updates to the data are tracked.

On the other hand, you have to manually download your data each time you want to update your analysis. APIs can come to our rescue here - in particular the `{googlesheets4}` package,

which acts as a wrapper for [version 4 of the sheets API](#).

You can install the package from CRAN

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

We can then read the gapminder dataset as follows.

```
gapminder_url <- "https://docs.google.com/spreadsheets/d/1U6Cf_qE0hiR9AZqTqS3mbMF3zt2db48Z
googlesheets4::read_sheet(gapminder_url)
```

v Reading from "gapminder".

v Range 'Africa'.

```
# A tibble: 624 x 6
  country continent year lifeExp      pop gdpPercap
  <chr>      <chr>   <dbl>   <dbl>   <dbl>   <dbl>
1 Algeria Africa    1952    43.1  9279525    2449.
2 Algeria Africa    1957    45.7 10270856    3014.
3 Algeria Africa    1962    48.3 11000948    2551.
4 Algeria Africa    1967    51.4 12760499    3247.
5 Algeria Africa    1972    54.5 14760787    4183.
6 Algeria Africa    1977    58.0 17152804    4910.
7 Algeria Africa    1982    61.4 20033753    5745.
8 Algeria Africa    1987    65.8 23254956    5681.
9 Algeria Africa    1992    67.7 26298373    5023.
10 Algeria Africa    1997    69.2 29072015    4797.
# i 614 more rows
```


Note: If you try and run this code, you'll likely have a browser tab asking you to sign in with Google and give permissions. Signing in and granting those permissions will mean that {googlesheets4} is able to do the same things you can with Sheets in the browser (create, read, edit, download, delete). **You do not need to do that for this tutorial.**

If you only want to read public Sheets (as we do in this example), then you can put googlesheets4 into a de-authorized state.

```
googlesheets4::gs4_deauth()
```


This will send an API key that can be used to access public resources for which no Google sign-in is required.

Sign in with Google




Choose an account

to continue to [Tidyverse API Packages](#)



Zak Varty

 Use another account

To continue, Google will share your name, email address, language preference, and profile picture with Tidyverse API Packages. Before using this app, you can review Tidyverse API Packages's [privacy policy](#) and terms of service.


English (United States) ▾

Help


Privacy

Terms


Sign in with Google





Tidyverse API Packages wants access to your Google Account



Select what [Tidyverse API Packages](#) can access

 Associate you with your personal info on Google ☒

 See your primary Google Account email address ☒

 See, edit, create and delete all your Google Sheets spreadsheets. [Learn more](#) ☒

Make sure that you trust Tidyverse API Packages

You may be sharing sensitive info with this site or app. You can always see or remove access in your [Google Account](#).

Learn how Google helps you [share data safely](#).

See Tidyverse API Packages's [privacy policy](#) and Terms of Service.

Cancel

Continue

English (United Kingdom) ▾

Help

Privacy

Terms

8

Tasks

1. Use {googlesheets4} to load a copy of our earlier example data set into R from the address provided.

https://docs.google.com/spreadsheets/d/1Xk7HX-EihNMCRaG1bsltAgkZKds_pz1tKIcw7pYEmEU/edit?usp=sharing

```
example_data_url <- "https://docs.google.com/spreadsheets/d/1Xk7HX-EihNMCRaG1bsltAgkZKds_pz1tKIcw7pYEmEU/edit?usp=sharing"
googlesheets4::read_sheet(example_data_url)
```

v Reading from "example-table".

v Range 'plain text version'.

```
# A tibble: 30 x 3
  age gender height
<dbl> <chr>   <dbl>
1    45 Male     161
2    97 Female   157
3    39 Female   150
4    26 Female   150
5    22 Male     159
6    55 Female   147
7    33 Female   155
8    21 Male     165
9    87 Male     152
10   96 Male     164
# i 20 more rows
```

2. Adapt your previous answer to read only the first 5 rows.

```
googlesheets4::read_sheet(example_data_url, n_max = 5)
```

v Reading from "example-table".

v Range 'plain text version'.

```
# A tibble: 5 x 3
  age gender height
<dbl> <chr>   <dbl>
1    45 Male     161
2    97 Female   157
3    39 Female   150
4    26 Female   150
5    22 Male     159
```

1	45	Male	161
2	97	Female	157
3	39	Female	150
4	26	Female	150
5	22	Male	159

3. Adapt your previous answer to read only the first 3 rows of the age and height columns.

We can use the `range` argument to have more detailed control over which rows and columns are read from the spreadsheet, but this only works for contiguous regions. We therefore read the two columns separately and join these together into a single tibble.

```
ages <- googlesheets4::read_sheet(example_data_url, range = "A1:A4")
```

v Reading from "example-table".

v Range 'A1:A4'.

```
heights <- googlesheets4::read_sheet(example_data_url, range = "C1:C4")
```

v Reading from "example-table".

v Range 'C1:C4'.

```
cbind(ages,heights)
```

	age	height
1	45	161
2	97	157
3	39	150

4. Read the second sheet, which has cell formatting. Has this caused any problems? How can you specify which sheet (tab) you want to read?

```
googlesheets4::read_sheet(example_data_url, sheet = 2)
```

v Reading from "example-table".

```
v Range "'with formatting'".
```

```
# A tibble: 30 x 3
  age gender height
  <dbl> <chr>   <dbl>
1    45 Male     161
2    97 Female   157
3    39 Female   150
4    26 Female   150
5    22 Male     159
6    55 Female   147
7    33 Female   155
8    21 Male     165
9    87 Male     152
10   96 Male     164
# i 20 more rows
```

No problems (but the formatting does get ignored, which might be problematic if this formatting was meaningful).

5. Read the indoor rowing records from the third sheet into R.

```
rowing_header <- googlesheets4::read_sheet(example_data_url, range = "Sheet3!B1:H1")
```

```
v Reading from "example-table".
```

```
v Range "'Sheet3'!B1:H1".
```

```
sheets_rowing <- googlesheets4::read_sheet(example_data_url, range = "Sheet3!B3:H14", col_
```

```
v Reading from "example-table".
```

```
v Range "'Sheet3'!B3:H14".
```

```
sheets_rowing
```

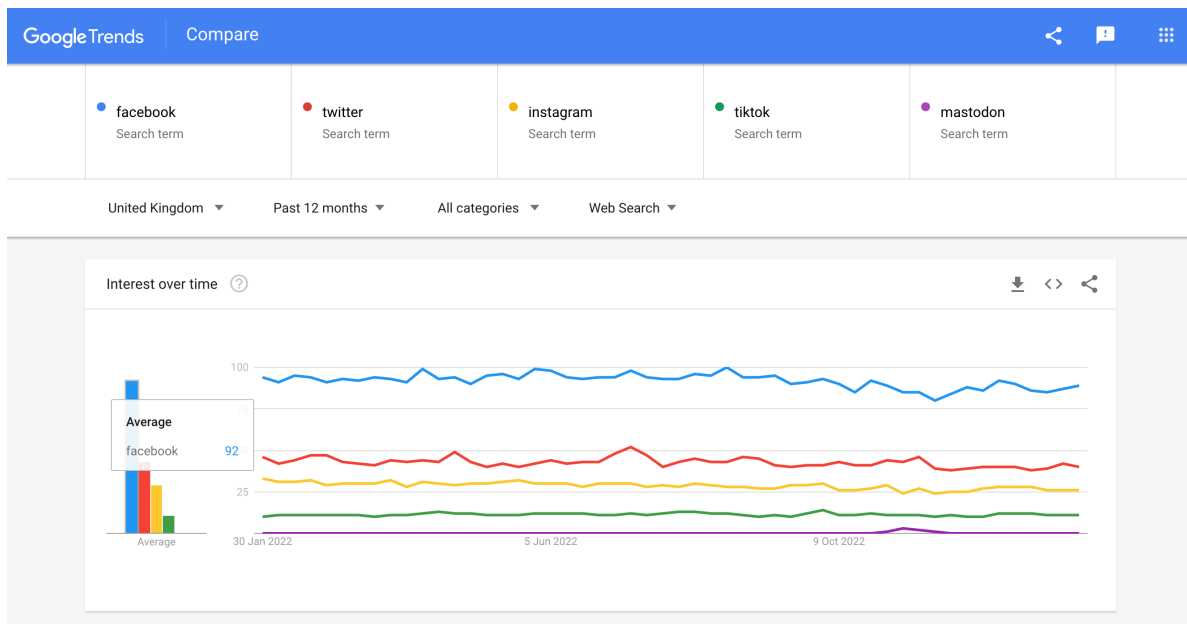
```
# A tibble: 12 x 7
```

	Type	Category	Time	Split	Nation	Year	Name
	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<chr>
1	SIR	Open Men	5:35.8	1:23.9	Australia Australia	2018	Josh~
2	SIR	Lightweight Men	5:56.7	1:29.2	Denmark Denmark	2012	Henr~
3	SIR	Open Women	6:21.1	1:35.3	United States United States	2021	Broo~
4	SIR	Lightweight Women	6:53.8	1:43.4	Canada Canada	2019	Jenn~
5	IRwS	Open Men	5:59.9	NA	Australia Australia	2010	Nick~
6	IRwS	Lightweight Men	6:10.3	NA	Italy Italia	2017	Andr~
7	IRwS	Open Women	6:46.8	NA	United States United States	2006	Tali~
8	IRwS	Lightweight Women	7:29.7	NA	United States United States	2008	Lynn~
9	DIR	Open Men	5:41.8	NA	New Zealand New Zealand	2012	Eric~
10	DIR	Lightweight Men	6:15.6	NA	United Kingdom Great Brita~	2017	Tom ~
11	DIR	Open Women	8:00.6	NA	United States United States	2015	Kare~
12	DIR	Lightweight Women	8:10.4	NA	United States United States	2013	Eliz~

- For examples of how to read private documents or write to Google sheets files see the [documentation](#).
- If you would like to use R to manage your Google drive files more generally, then check out the [googledrive](#) package.

API Example 2 - Google Trends

Go to Google Trends and recreate this time series plot of the search popularity of social media platforms over the last 12 months.



Tasks

Use the {gtrendsR} package to reproducibly recreate this plot.

(**Note:** fast internet connections can mess this up and cause an error because the request rate is too high. Setting `onlyInterest=TRUE` is a way to reduce this issue by making fewer queries to the API. Don't worry if it still throws an error - it just means your computer/internet are too good for the API!)

```
library(gtrendsR)
gt <- gtrendsR::gtrends(
  keyword = c("facebook", "twitter", "instagram", "tiktok", "mastodon"),
  geo = "GB",
  time = "today 12-m",
  onlyInterest = TRUE)
plot(gt)
```

