IMPORTING JSON DATA

summary:

- Start with nested data in JSON format - from a website → series of operations - get into nice clean rectangle format (just data we need and format we need) - to further look at insight and conclusion we want from our data

# LOAD PACKAGES

- load contribute packages

```
pacman::p_load(pacman, tidyverse, jsonlite)
```

  - **jsonlite**

# GET JSON DATA

- javascript object notation (a little older than XML)
- extract same data as in XML video, but this time with JSON - about 1954 formula 1 races
- http://ergast.com/mrd (explains that can use this info to develop your code)
- file http://ergast.com/api/f1/1954/results/1.json → json data is not in tidy table format - is just a text list
  - hard for humans to read, but easy for computer to read

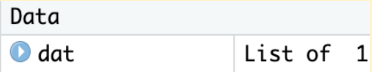
Save info into object called ***dat*** (short for data)

```
dat <- http://ergast.com/api/f1/1954/results/1.json” %>%
    fromJSON() %>%          # put data into list
  print()                   # see raw data
```

- results:

environment - data: dat - list of 1

- messy printout

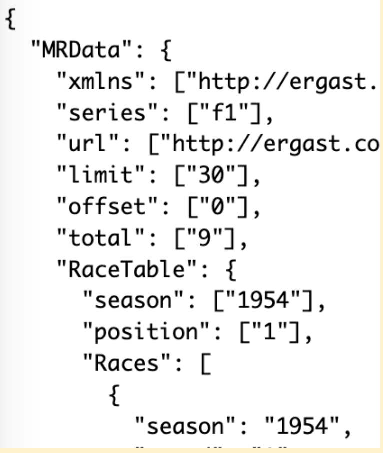


```
Data
dat                 List of 1
```

```
dat %>% toJSON(pretty = T)   # see nested JSON structure
```

- results:

-printout is indented - for different levels of information



```
{
  "MRData": {
    "xmlns": ["http://ergast.
    "series": ["f1"],
    "url": ["http://ergast.co
    "limit": ["30"],
    "offset": ["0"],
    "total": ["9"],
    "RaceTable": {
      "season": ["1954"],
      "position": ["1"],
      "Races": [
        {
          "season": "1954",
```

# LOCATE DATA

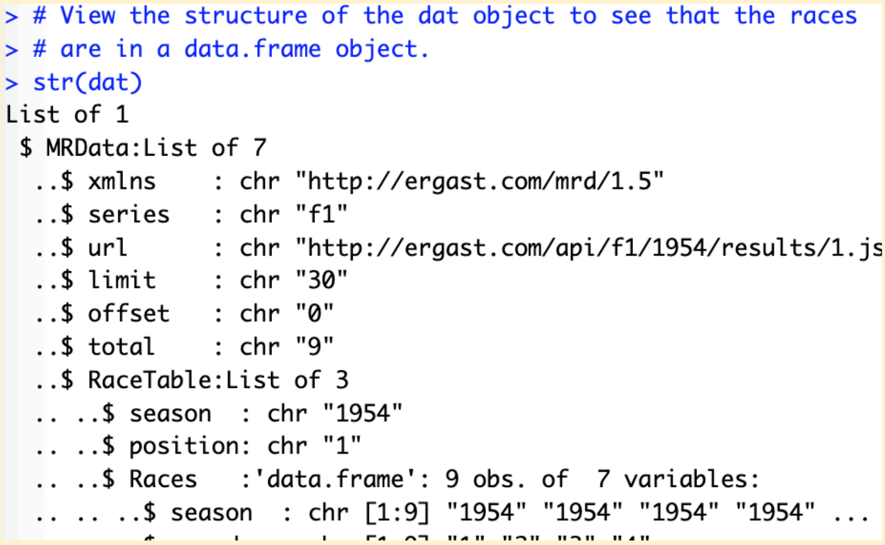
- to find data we need = the race, first and last name, team name

View structure of the `dat` object to see that races are in a dataframe object

```
View structure of the dat object to see that races are in a dataframe object
str(dat)
```

- results:

-indented structure - easier to find things we are looking for



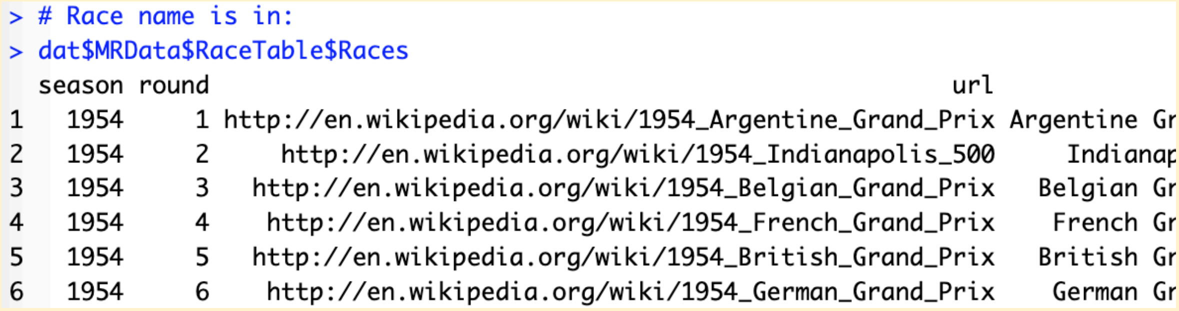
```
> # View the structure of the dat object to see that the races
> # are in a data.frame object.
> str(dat)
List of 1
 $ MRData:List of 7
  ..$ xmlns     : chr "http://ergast.com/mrd/1.5"
  ..$ series    : chr "f1"
  ..$ url       : chr "http://ergast.com/api/f1/1954/results/1.js
  ..$ limit     : chr "30"
  ..$ offset    : chr "0"
  ..$ total     : chr "9"
  ..$ RaceTable:List of 3
  .. ..$ season  : chr "1954"
  .. ..$ position: chr "1"
  .. ..$ Races   :'data.frame': 9 obs. of  7 variables:
  .. .. ..$ season   : chr [1:9] "1954" "1954" "1954" "1954" ...
```

Race name is in

```
dat$MRData$RaceTable$Races
```

- results:

-info about races - still pretty complex



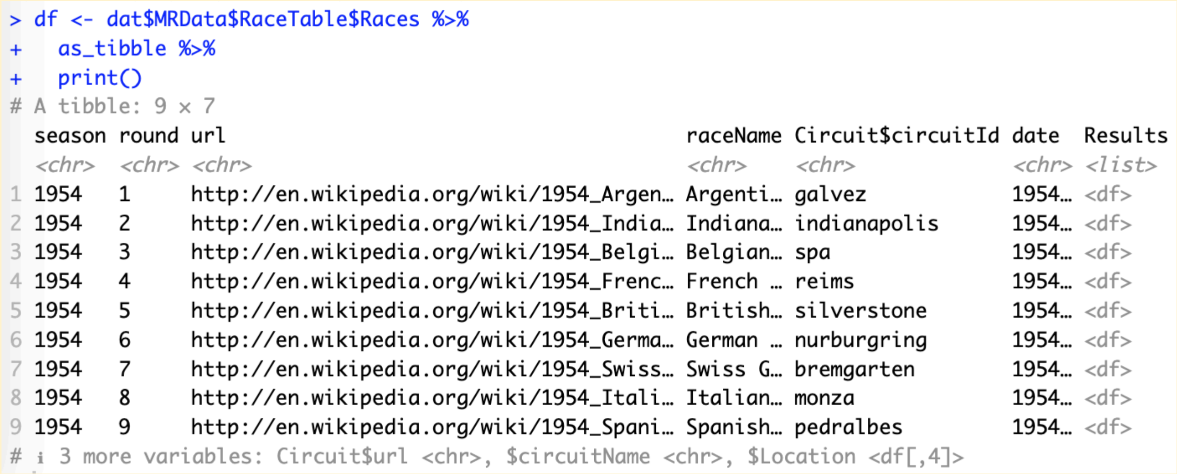
```
> # Race name is in:
> dat$MRData$RaceTable$Races
  season round                                               url
1   1954     1 http://en.wikipedia.org/wiki/1954_Argentine_Grand_Prix Argentine Gr
2   1954     2     http://en.wikipedia.org/wiki/1954_Indianapolis_500     Indianap
3   1954     3   http://en.wikipedia.org/wiki/1954_Belgian_Grand_Prix   Belgian Gr
4   1954     4    http://en.wikipedia.org/wiki/1954_French_Grand_Prix    French Gr
5   1954     5   http://en.wikipedia.org/wiki/1954_British_Grand_Prix   British Gr
6   1954     6    http://en.wikipedia.org/wiki/1954_German_Grand_Prix    German Gr
```

Create tibble

```
df <- dat$MRData$RaceTable$Races %>%
    as.tibble() %>%
    print()
```

- results:

-environment - data: df - 9 obs of 7 variables



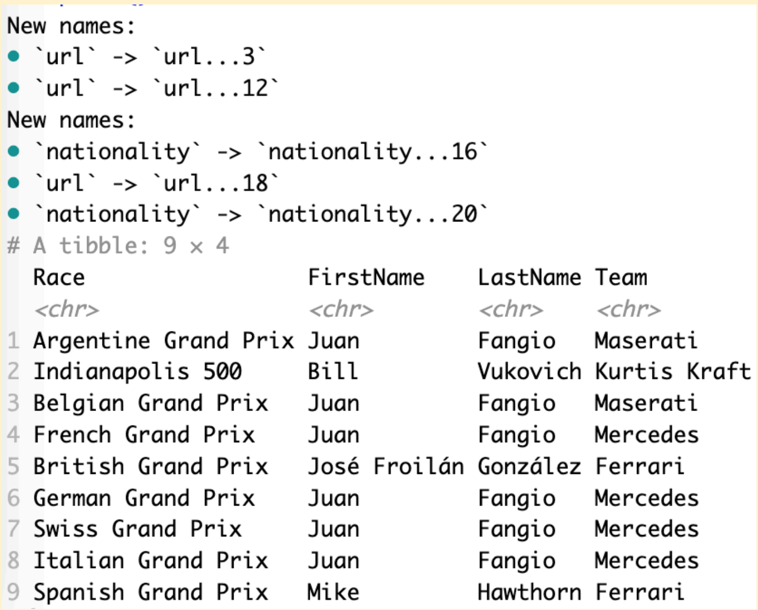
```
> df <- dat$MRData$RaceTable$Races %>%
+   as_tibble %>%
+   print()
# A tibble: 9 × 7
  season round url                                      raceName Circuit$circuitId date  Results
  <chr>  <chr> <chr>                                    <chr>    <chr>             <chr> <list>
1 1954   1     http://en.wikipedia.org/wiki/1954_Argen… Argenti… galvez            1954… <df>
2 1954   2     http://en.wikipedia.org/wiki/1954_India… Indiana… indianapolis      1954… <df>
3 1954   3     http://en.wikipedia.org/wiki/1954_Belgi… Belgian… spa               1954… <df>
4 1954   4     http://en.wikipedia.org/wiki/1954_Frenc… French … reims             1954… <df>
5 1954   5     http://en.wikipedia.org/wiki/1954_Briti… British… silverstone       1954… <df>
6 1954   6     http://en.wikipedia.org/wiki/1954_Germa… German … nurburgring       1954… <df>
7 1954   7     http://en.wikipedia.org/wiki/1954_Swiss… Swiss G… bremgarten        1954… <df>
8 1954   8     http://en.wikipedia.org/wiki/1954_Itali… Italian… monza             1954… <df>
9 1954   9     http://en.wikipedia.org/wiki/1954_Spani… Spanish… pedralbes         1954… <df>
# ℹ 3 more variables: Circuit$url <chr>, $circuitName <chr>, $Location <df[,4]>
```

- problem: has more info than need - dn need url info
- soln: unnest data and select variables
  - use **names_repair**
    - bc some of the nested data frames have the same variable names, and need to distinguish them

```
df %<>%
    unnest_wider(Results) %%
    unnest_wider(Driver, names_repair = “unique”) %>%
    unnest_wider(Constructor, names-repair = “unique”) %>%
    select(
       Race = racename,          # get race name
       FirstName = givenName, # get first name
       LastName = familyName   # get last name
       Team = name               # get team name
       ) %>%
    print()                      # show data
```

- results:

-small dataframe
-has everything that want (except not all are Grand Prix races



```
New names:
● `url` -> `url...3`
● `url` -> `url...12`
New names:
● `nationality` -> `nationality...16`
● `url` -> `url...18`
● `nationality` -> `nationality...20`
# A tibble: 9 x 4
  Race                 FirstName    LastName Team
  <chr>                <chr>        <chr>    <chr>
1 Argentine Grand Prix Juan         Fangio   Maserati
2 Indianapolis 500     Bill         Vukovich Kurtis Kraft
3 Belgian Grand Prix   Juan         Fangio   Maserati
4 French Grand Prix    Juan         Fangio   Mercedes
5 British Grand Prix   José Froilán González Ferrari
6 German Grand Prix    Juan         Fangio   Mercedes
7 Swiss Grand Prix     Juan         Fangio   Mercedes
8 Italian Grand Prix   Juan         Fangio   Mercedes
9 Spanish Grand Prix   Mike         Hawthorn Ferrari
```

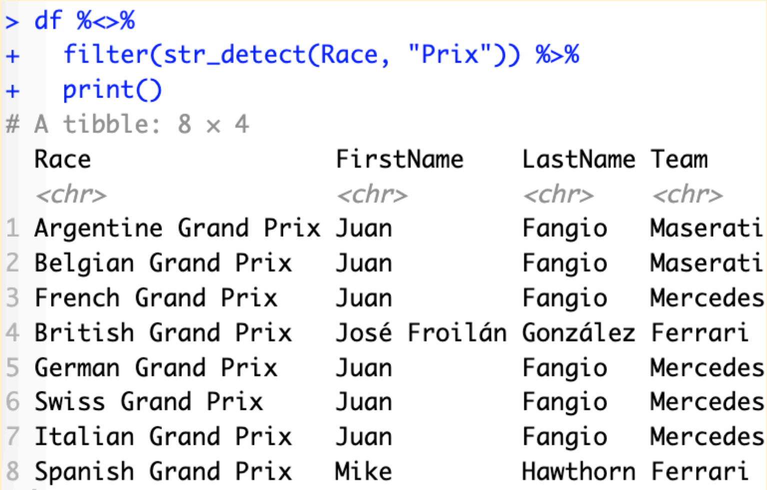
# FILTER AND PRINT DATA

Filter cases - select just Grand Prix

```
df %>%
    filter(str_detect(Race, “Prix”)) %>%
    print()
```

- results:

-see that juan fongio won 6 of the races, even for 2 dift teams - which explains why he is a legend in the early history of auto racing



```
> df %<>%
+   filter(str_detect(Race, "Prix")) %>%
+   print()
# A tibble: 8 × 4
  Race                 FirstName    LastName Team
  <chr>                <chr>        <chr>    <chr>
1 Argentine Grand Prix Juan         Fangio   Maserati
2 Belgian Grand Prix   Juan         Fangio   Maserati
3 French Grand Prix    Juan         Fangio   Mercedes
4 British Grand Prix   José Froilán González Ferrari
5 German Grand Prix    Juan         Fangio   Mercedes
6 Swiss Grand Prix     Juan         Fangio   Mercedes
7 Italian Grand Prix   Juan         Fangio   Mercedes
8 Spanish Grand Prix   Mike         Hawthorn Ferrari
```