# Realistic example

So far, you have seen the basics of manipulating data frames with our cat data; now let's use those skills to digest a more realistic dataset. Let's read in the `gapminder` dataset that we downloaded previously:

```
gapminder <- read.csv("data/gapminder_data.csv", stringsAsFactors = TRUE)
```

## Miscellaneous Tips

- Another type of file you might encounter are tab-separated value files (.tsv). To specify a tab as a separator, use `"\\t"` or `read.delim()`.
- Files can also be downloaded directly from the Internet into a local folder of your choice onto your computer using the `download.file` function. The `read.csv` function can then be executed to read the downloaded file from the download location, for example,

```
download.file("https://raw.githubusercontent.com/swcarpentry/r-novice-gapminder/gh-pages/_episodes_rmd/data/gapminder_
data.csv", destfile = "data/gapminder_data.csv")
gapminder <- read.csv("data/gapminder_data.csv", stringsAsFactors = TRUE)
```

- Alternatively, you can also read in files directly into R from the Internet by replacing the file paths with a web address in `read.csv`. One should note that in doing this no local copy of the csv file is first saved onto your computer. For example,

```
gapminder <- read.csv("https://raw.githubusercontent.com/swcarpentry/r-novice-gapminder/gh-pages/_episodes_rmd/data/ga
pminder_data.csv", stringsAsFactors = TRUE)
```

- You can read directly from excel spreadsheets without converting them to plain text first by using the readxl package.

Let's investigate gapminder a bit; the first thing we should always do is check out what the data looks like with `str`:

```
str(gapminder)
```

```
'data.frame':	1704 obs. of  6 variables:
 $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
 $ pop      : num  8425333 9240934 10267083 11537966 13079460 ...
 $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
 $ gdpPercap: num  779 821 853 836 740 ...
```

An additional method for examining the structure of gapminder is to use the `summary` function. This function can be used on various objects in R. For data frames, `summary` yields a numeric, tabular, or descriptive summary of each column. Factor columns are summarized by the number of items in each level, numeric or integer columns by the descriptive statistics (quartiles and mean), and character columns by its length, class, and mode.

```
summary(gapminder$country)
```

```
             Afghanistan                  Albania                  Algeria
                      12                       12                       12
                  Angola                Argentina                Australia
                      12                       12                       12
                 Austria                  Bahrain               Bangladesh
                      12                       12                       12
                 Belgium                    Benin                  Bolivia
                      12                       12                       12
  Bosnia and Herzegovina                 Botswana                   Brazil
                      12                       12                       12
                Bulgaria             Burkina Faso                  Burundi
                      12                       12                       12
                Cambodia                 Cameroon                   Canada
                      12                       12                       12
Central African Republic                     Chad                    Chile
                      12                       12                       12
                   China                 Colombia                  Comoros
                      12                       12                       12
         Congo Dem. Rep.               Congo Rep.               Costa Rica
                      12                       12                       12
           Cote d'Ivoire                  Croatia                     Cuba
                      12                       12                       12
          Czech Republic                  Denmark                 Djibouti
                      12                       12                       12
      Dominican Republic                  Ecuador                    Egypt
                      12                       12                       12
             El Salvador        Equatorial Guinea                  Eritrea
                      12                       12                       12
                Ethiopia                  Finland                   France
                      12                       12                       12
                   Gabon                   Gambia                  Germany
                      12                       12                       12
                   Ghana                   Greece                Guatemala
                      12                       12                       12
                  Guinea            Guinea-Bissau                    Haiti
                      12                       12                       12
                Honduras           Hong Kong China                  Hungary
                      12                       12                       12
                 Iceland                    India                Indonesia
                      12                       12                       12
                    Iran                     Iraq                  Ireland
                      12                       12                       12
                  Israel                    Italy                  Jamaica
                      12                       12                       12
                   Japan                   Jordan                    Kenya
                      12                       12                       12
         Korea Dem. Rep.               Korea Rep.                   Kuwait
                      12                       12                       12
                 Lebanon                  Lesotho                  Liberia
                      12                       12                       12
                   Libya               Madagascar                   Malawi
                      12                       12                       12
                Malaysia                     Mali               Mauritania
                      12                       12                       12
               Mauritius                   Mexico                 Mongolia
                      12                       12                       12
              Montenegro                  Morocco               Mozambique
                      12                       12                       12
                 Myanmar                  Namibia                    Nepal
                      12                       12                       12
             Netherlands              New Zealand                Nicaragua
                      12                       12                       12
                   Niger                  Nigeria                   Norway
                      12                       12                       12
                    Oman                 Pakistan                   Panama
                      12                       12                       12
                 (Other)
                     516
```

Along with the `str` and `summary` functions, we can examine individual columns of the data frame with our `typeof` function:

```
typeof(gapminder$year)
```

```
[1] "integer"
```

```
typeof(gapminder$country)
```

```
[1] "integer"
```

```
str(gapminder$country)
```

```
 Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
```

We can also interrogate the data frame for information about its dimensions; remembering that `str(gapminder)` said there were 1704 observations of 6 variables in gapminder, what do you think the following will produce, and why?

```
length(gapminder)
```

```
[1] 6
```

A fair guess would have been to say that the length of a data frame would be the number of rows it has (1704), but this is not the case; remember, a data frame is a *list of vectors and factors*:

```
typeof(gapminder)
```

```
[1] "list"
```

When `length` gave us 6, it's because gapminder is built out of a list of 6 columns. To get the number of rows and columns in our dataset, try:

```
nrow(gapminder)
```

```
[1] 1704
```

```
ncol(gapminder)
```

```
[1] 6
```

Or, both at once:

```
dim(gapminder)
```

```
[1] 1704    6
```

We'll also likely want to know what the titles of all the columns are, so we can ask for them later:

```
colnames(gapminder)
```

```
[1] "country"   "year"      "pop"       "continent" "lifeExp"   "gdpPercap"
```

At this stage, it's important to ask ourselves if the structure R is reporting matches our intuition or expectations; do the basic data types reported for each column make sense? If not, we need to sort any problems out now before they turn into bad surprises down the road, using what we've learned about how R interprets data, and the importance of *strict consistency* in how we record our data.

Once we're happy that the data types and structures seem reasonable, it's time to start digging into our data proper. Check out the first few lines:

```
head(gapminder)
```

```
      country year      pop continent lifeExp gdpPercap
1 Afghanistan 1952  8425333      Asia  28.801  779.4453
2 Afghanistan 1957  9240934      Asia  30.332  820.8530
3 Afghanistan 1962 10267083      Asia  31.997  853.1007
4 Afghanistan 1967 11537966      Asia  34.020  836.1971
5 Afghanistan 1972 13079460      Asia  36.088  739.9811
6 Afghanistan 1977 14880372      Asia  38.438  786.1134
```

## Challenge 3

It’s good practice to also check the last few lines of your data and some in the middle. How would you do this?

Searching for ones specifically in the middle isn’t too hard, but we could ask for a few lines at random. How would you code this?

## Solution to Challenge 3

To check the last few lines it’s relatively simple as R already has a function for this:

```
tail(gapminder)
tail(gapminder, n = 15)
```

What about a few arbitrary rows just in case something is odd in the middle?

## Tip: There are several ways to achieve this.

The solution here presents one form of using nested functions, i.e. a function passed as an argument to another function. This might sound like a new concept, but you are already using it! Remember my_dataframe[rows, cols] will print to screen your data frame with the number of rows and columns you asked for (although you might have asked for a range or named columns for example). How would you get the last row if you don’t know how many rows your data frame has? R has a function for this. What about getting a (pseudorandom) sample? R also has a function for this.

```
gapminder[sample(nrow(gapminder), 5), ]
```