

# Data Wrangling with `dplyr` - Exercises

Lachlan Deer

Uli Bergmann

The goal of this document is to give you some experience using functions from `dplyr` with a little less guidance. Proceed at your own pace, and we will provide solutions later on.

We want to build up some experience with is `dplyr`.

Load it:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

We are going to use a classic dataset from Graduate Econometrics texts that has gasoline consumption and some other variables by country for the 60's and 70's. The data is inside the package `plm` and called `Gasoline`. Load it as follows:

```
data(Gasoline, package = "plm")
```

Use the help to get a sense of what variable names mean in the data.

Now work through the following exercises:

1. Create a dataset `gasoline` which is a tibble
2. Create a subset of that only has data from the 1960s. Do this two ways, once where you don't use piping, `%>%`, and once where you do.
3. Create a subset that contains data from the years ranging from 1969 to 1973.
4. Create a subset that contains data for the years 1969, 1973 and 1977.
5. Create a dataset that contains only the columns `country`, `year`, `lrpmg`.
6. Create a dataset that does not contain the columns `country`, `year`, `lrpmg`.
7. Rename the column `year` to be called `date`.
8. Select all columns that start with "l".
9. Select the columns `country`, `year`, and all columns that contain the letters "car".
10. What does the function `pull()` do? Try it on the column `lrpmg`.
11. Create a grouped data set that groups the data by country.
12. Ungroup the dataset from 10.
13. Find the mean of `lgaspcar` by country. Call that variable `avg_lgaspcar`.
14. Return a dataset that computes the mean of `lgaspcar` for france.
15. Compute the mean, standard deviation, min and max of `lgaspcar` by country.
16. Which country has the highest average gasoline consumption.
17. Return a dataset that returns the countries with the highest and lowest average consumption.

18. Add a variable `count` to the dataset that has the number of times each country appears in the data - Is it balanced?
19. Create a meaningless dataset called `spam` that is the exponential of the sum of `lgaspcar` and `lincomep`. Also check out what happens if you replace `mutate()` with `transmute()`.
20. Create the lead and lag of `lgaspcar` for each row of data. Call the new columns `lead_lgaspcar` and `lag_lgaspcar`.
21. The following countries belong to EU:

```
eu_countries <- c("austria", "belgium", "bulgaria", "croatia", "republic of cyprus",
                  "czech republic", "denmark", "estonia", "finland", "france", "germany",
                  "greece", "hungary", "ireland", "italy", "latvia", "lithuania", "luxembourg",
                  "malta", "netherla", "poland", "portugal", "romania", "slovakia", "slovenia",
                  "spain", "sweden", "u.k.")
```

Create a variable `in_eu` in the gasoline data which takes the value `TRUE` if a country is in the EU. (Note that the case of the string will matter!)

22. Here's a different way to classify countries:

- Mediterranean: france, italy, turkey, greece, spain
- Central Europe: germany, austria, switzerl, belgium, netherla
- Anglosphere: canada, u.s.a. , u.k., ireland
- Nordic: denmark, norway, sweden
- asia: japan

Create a new variable `region` that uses these definitions. (Hint: `case_when()` will likely be your friend here.

23. Notice that in the country names `switzerl` and `netherla` are a little funky. Use the functions `mutate` and `recode` to replace the name with the full country name.
24. Compute the variable `quintile` that computes which quintile of `lgaspcar` each country is. Do this only for 1960. Repeat this to create a variable `decile` with the appropriate definition.
25. Create a variable `high_consumption` that takes the value `TRUE` if `lgaspcar` is higher than the yearly average for a given country.