Data transformation with dplyr

And make a plot

Prerna Prasad

```
library(readr)
library(dplyr)
library(ggplot2)
library(ggthemes)
```

Import

In this exercise we use data of the UNICEF/WHO Joint Monitoring Programme (JMP) for Water Supply, Sanitation and Hygiene (WASH). The data is available at https://washdata.org/data and published as an R data package at https://github.com/WASHNote/jmpwashdata/.

The data set `jmp_wld_sanitation_long` is available in the `data` folder of this repository. The data set is in long format and contains the following variables:

- `name`: country name
- `iso3`: ISO3 country code
- `year`: year of observation
- `region_sdg`: SDG region
- `residence`: residence type (national, rural, urban)
- `varname_short`: short variable name (JMP naming convention)
- `varname_long`: long variable name (JMP naming convention)

We use the `read_csv()` function to import the data set into R.

```
sanitation <- read_csv("data/jmp_wld_sanitation_long.csv")
```

Task 1

1. Run all code chunks above.
2. Use the `glimpse()` function to get an overview of the data set.
3. How many variables are in the data set?

```
glimpse(sanitation)
```

```
Rows: 73,710
Columns: 8
$ name          <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanista~
$ iso3          <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", ~
$ year          <dbl> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 20~
$ region_sdg    <chr> "Central and Southern Asia", "Central and Southern Asia"~
$ varname_short <chr> "san_bas", "san_bas", "san_bas", "san_lim", "san_lim", "~
$ varname_long  <chr> "basic sanitation services", "basic sanitation services"~
$ residence     <chr> "national", "rural", "urban", "national", "rural", "urba~
$ percent       <dbl> 21.870802, 19.322798, 30.863719, 5.648528, 3.136148, 14.~
```

Task 2

1. Use the `count()` function with varname_short and varname_long to identify the definitions of the levels in these two variables.

```
sanitation %>%
  count(varname_short,varname_long)
```

```
# A tibble: 5 x 3
  varname_short varname_long                          n
  <chr>         <chr>                             <int>
1 san_bas       basic sanitation services         14742
2 san_lim       limited sanitation services       14742
3 san_od        no sanitation facilities          14742
4 san_sm        safely managed sanitation services 14742
5 san_unimp     unimproved sanitation facilities  14742
```

Task 3

1. Use the `filter()` function to create a subset of the data set that only contains observations:

- for the country you live or work in,
- for the year 2000 and 2020,
- for all variables that are not "safely managed sanitation services".

2. Store the result as a new object in your environment with a name of your choice.

```
india_jmp <- sanitation %>%
  filter(iso3=="IND",
         year %in% c(2000,2020),
         varname_short %in% c("san_bas", "san_lim", "san_od","san_unimp"))

glimpse(india_jmp)
```

```
Rows: 24
Columns: 8
$ name          <chr> "India", "India", "India", "India", "India", "India", "I~
$ iso3          <chr> "IND", "IND", "IND", "IND", "IND", "IND", "IND", "IND", ~
$ year          <dbl> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 20~
$ region_sdg    <chr> "Central and Southern Asia", "Central and Southern Asia"~
$ varname_short <chr> "san_bas", "san_bas", "san_bas", "san_lim", "san_lim", "~
$ varname_long  <chr> "basic sanitation services", "basic sanitation services"~
$ residence     <chr> "national", "rural", "urban", "national", "rural", "urba~
$ percent       <dbl> 15.0148673, 2.2527395, 48.3803556, 5.1485781, 0.5154295,~
```

Task 4

1. Use the `count()` function with the data you created in Task 3 to verify that year 2000 and 2020 remained in the year variable.

```
india_jmp %>%
  count(year)
```

```
# A tibble: 2 x 2
   year     n
  <dbl> <int>
```

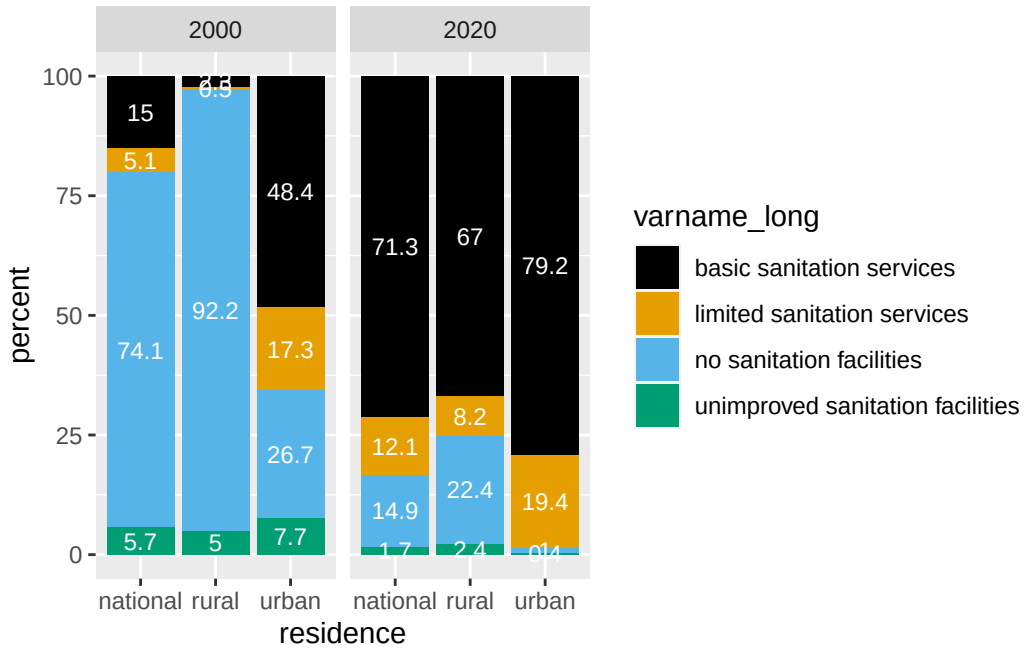
```
1   2000     12
2   2020     12
```

Task 5

1. Use the `ggplot()` function to create a bar plot with `geom_col()` for the data you created in Task 3.
2. Use the `aes()` function to map the `residence` variable to the x-axis, the `percent` variable to the y-axis, and the `varname_long` variable to the fill aesthetic.
3. Use `facet_wrap()` to create a separate plot for each year.
4. Change the fill colors using `scale_fill_colorblind()`.
5. Add labels to the bars by copying the code below this bullet point and adding it to your code for the plot.

```
geom_text(aes(label = round(percent, 1)),
          position = position_stack(vjust = 0.5),
          size = 3,
          color = "white")
```

```
ggplot(india_jmp,
       aes(x=residence, y=percent, fill=varname_long))+
      geom_col()+
  facet_wrap(~year)+
  scale_fill_colorblind()+
  geom_text(aes(label = round(percent, 1)),
            position = position_stack(vjust = 0.5),
            size = 3,
            color = "white")
```



Task 6

If you haven't worked with JMP indicators before, the following questions will be challenging to answer.

1. Look at the plot that you created. What do you notice about the order of the bars / order of the legend?
2. What would you want to change? - There has been a huge improvement from 2000 to 2020 however, in rural areas there is still a considerable percentage (approx 23%) that has no sanitation facilities. I would focus more attention to providing sanitation facilities in these areas but also improving the basic sanitation services to safely managed sanitation services which would mean better access, safe containment, treatment and safe reuse is possible.
3. Why did we remove "safely managed sanitation services" from the data set in Task 3? - I think we are trying to focus on thh problem areas because of which we removed the safely managed sanitation services.

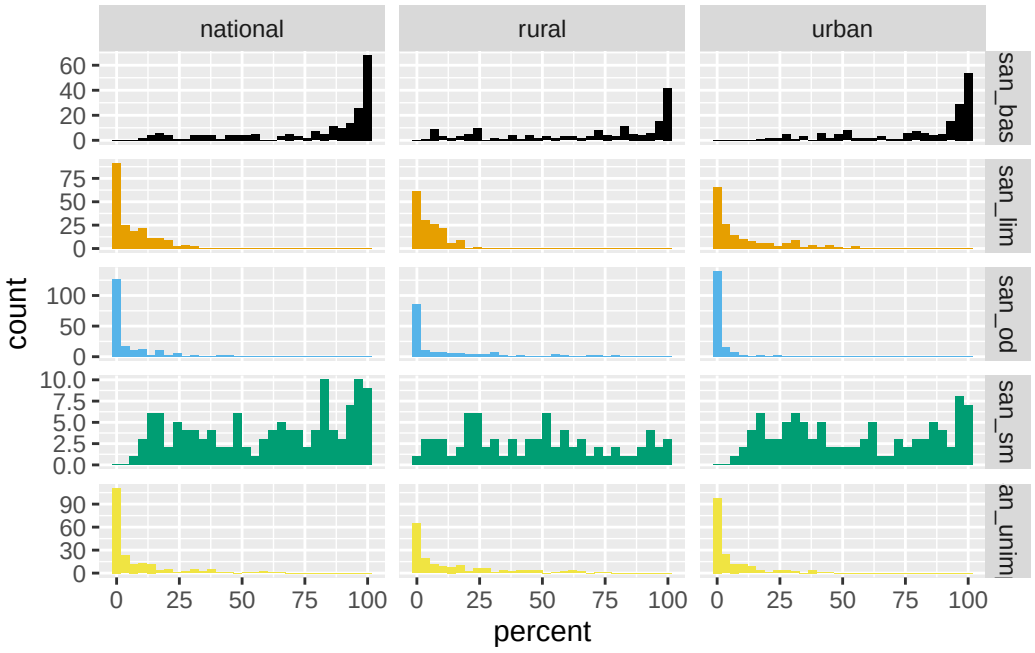
Task 7

1. Run the code in the code chunk below.
2. What do you observe when you look at the code and plot?

I think that most of the data does not follow a normal distribution - so if we need to do any statistical tests then we will have to choose non parametric tests.

```
sanitation_2020 <- sanitation |>
  filter(year == 2020)

ggplot(data = sanitation_2020,
       mapping = aes(x = percent, fill = varname_short)) +
  geom_histogram() +
  facet_grid(varname_short ~ residence, scales = "free_y") +
  scale_fill_colorblind() +
  theme(legend.position = "none")
```



Task 8: Data communication

In the YAML header (between the three dashes at the top of the document)

1. Add your name as the author of this document
2. Render the document and fix any errors

Task 9: Complete assignment

Stage, Commit & Push to GitHub

1. Open the Git pane in RStudio. It's in the top right corner in a separate tab.
2. **Stage** your changes by checking appropriate box next to all files (if you select one file with your mouse, you can then highlight them all with Ctrl + A on your keyboard and check all boxes).
3. Write a meaningful commit message (e.g. "Completed part b of homework assignment 03.) in the **Commit message** box.
4. Click **Commit**. Note that every commit needs to have a commit message associated with it.

Open an issue on GitHub

Once you have ensured that the Quarto document renders without errors and you have pushed all your changes to GitHub, you can complete the assignment by opening an issue on

1. Open github.com in your browser.
2. Navigate to the GitHub organisation for the course.
3. Find the repository md-03-assignments that ends with your GitHub username.
4. Click on the "Issues" tab.
5. Click on the green "New issue" button.
6. In the "Title" field write: "Completed module 3 assignments".
7. In the "Leave a comment" field, tag the course instructors @larnsce @mianzg @sskorik01 and ask some questions, if you like.