Work Injury and Illness - Data Wrangling Exercise

RUG @ HSG

Below, you will find some exercises to apply and improve your data wrangling and visualisation skills. Don't hesitate to ask questions in the Q&A WhatsApp group, we or your fellow students will be happy to answer them!

The packages we loaded for this exercise:

```
library(tidyverse) # For data wrangling
library(tidytext) # For sorting column plots
library(scales) # For formatting axis labels
```

The data set

In these exercises we will be working with data on **Work Injuries and Illnesses in the US** (Click here to download csv). Each row represents an entry for one firm in one year between 2016 and 2021. This dataset could be used to analyse particularly dangerous industries or see if there are time trends in the variables. The data source is the Occupational Safety and Health Administration (OSHA), who publish the yearly data (Link).

We have already downloaded and combined the individual files for you, only keeping the interesting variables. You can download the prepped data from our website (Link). Store it in the same folder as your RScript or Notebook and read in the `osha_combined.csv` file. Download the data dictionary *here*.

```
osha <- read_csv("osha_combined.csv")
```

The solutions can be found at the end of this document.

Task 1: Getting an overview over the data

(a) Try out the function `glimpse()` from the `tidyverse` to get an overview over the data. How many rows are there in the data? How many columns? Get a mental picture of the dataset.

(b) Check for missing values in the data. Are there any problematic columns?

(c) Count the values in the columns `no_injuries_illnesses` and `size`. Check out the data dictionary to understand what these columns mean. (*Hint:* The function `count()` gives you the value counts for the specified variables. Can you additionally try to sort the results?)

(d) Print summary information for all the columns that contain the word "total". (*Hint:* You can use `contains('some word')` within the `select` function.)

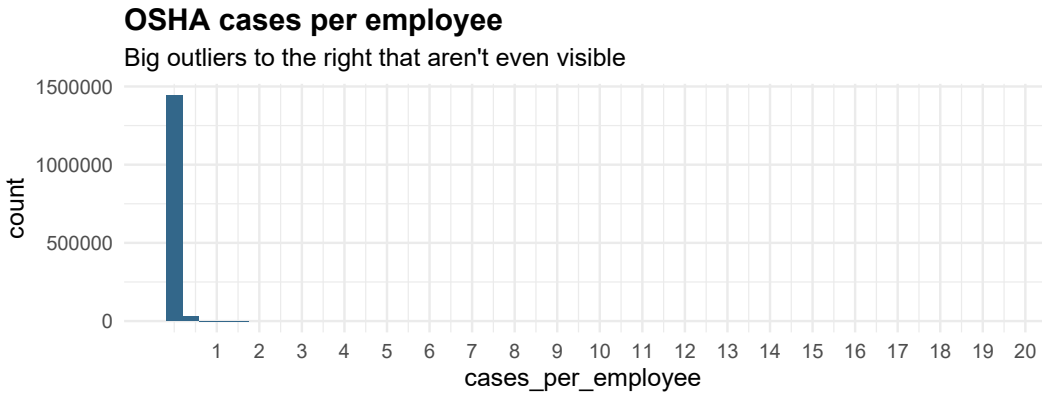
(e) Create a column `total_cases` which sums up `total_deaths`, `total_dafw_cases`, `total_djtr_cases` and `total_other_cases`.

(f) Filter out micro companies (fewer than 10 employees).

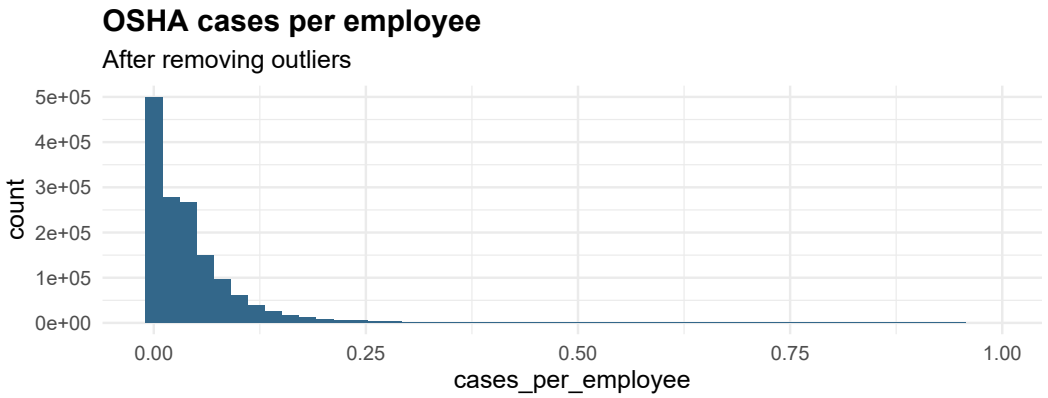
Task 2: Visualising the data

Now that you have an idea about the data, let's make some visualisations. If you feel like you don't know how to do it all yet, just have a look at the solution and try to understand every step (see what happens if you run the code line by line).

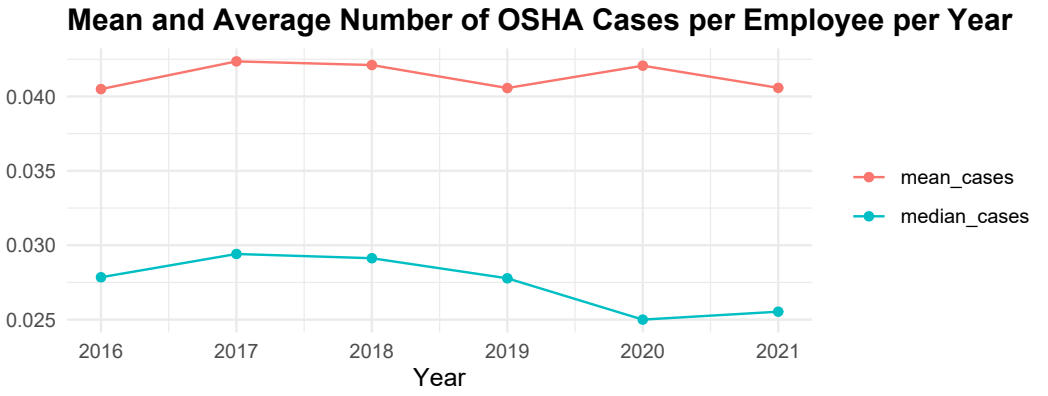
(a) Create two additional columns: `cases_per_employee` as total cases divided by average annual employees, and `deaths_per_employee`as total deaths divided by average annual employees. Plot a histogram of the distribution of cases per employee. Are there any companies that stand out as having particularly high or low injury rates?



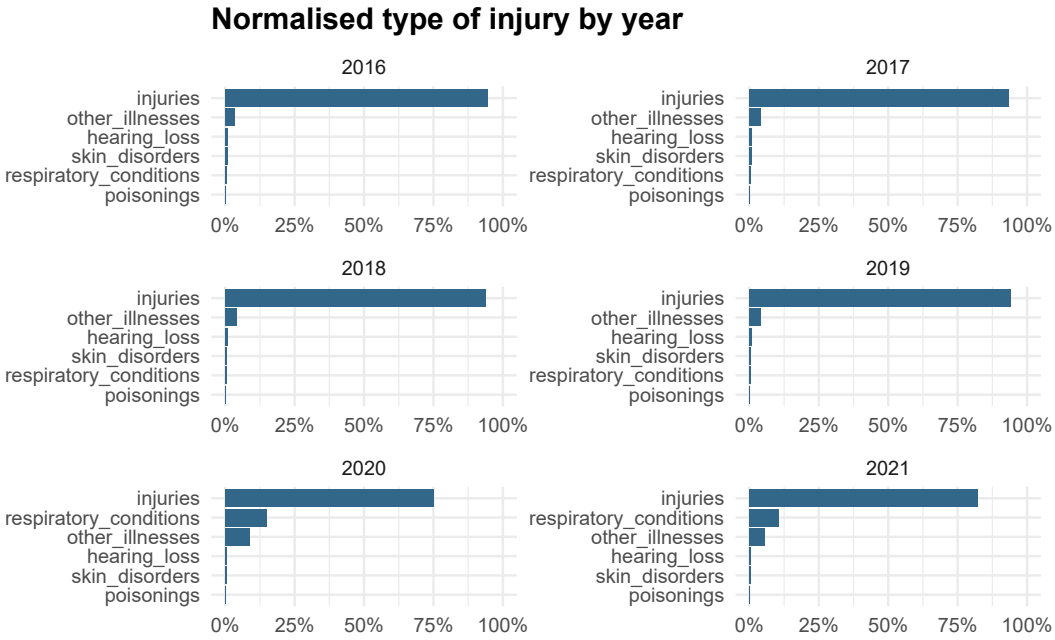
(b) In the previous exercise, we had huge outliers to the right. Count the number of firms with more than one case per employee. There should not be a lot of them, so drop them afterwards. Then plot the previous chart again.



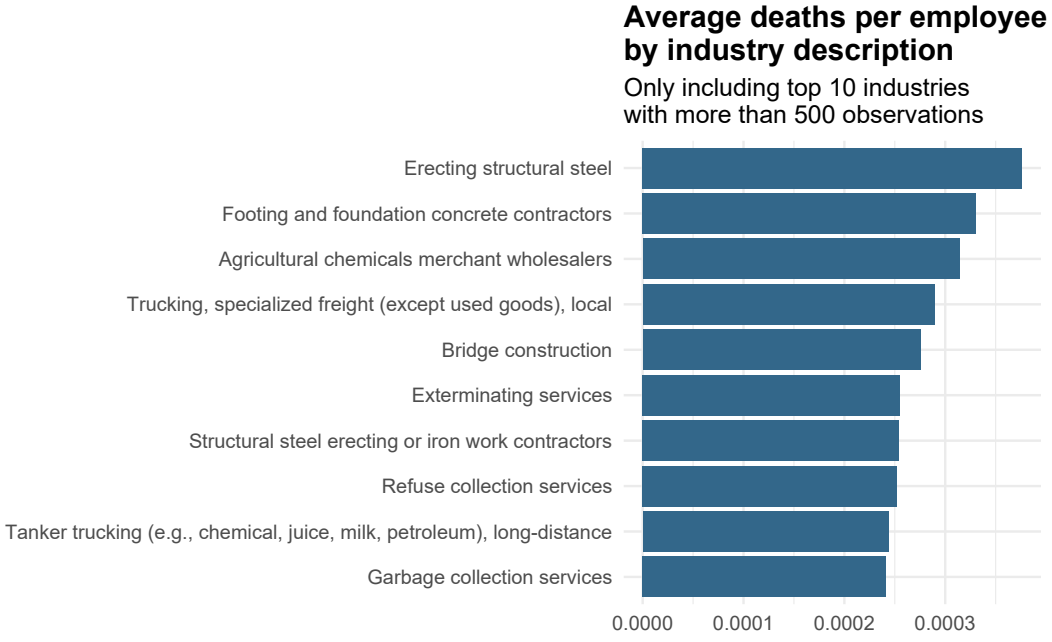
(c) Is the median and average number of cases per employee going up or down over time? Show it in a line plot with different colour for both lines.



(d) Plot the normalised count of each type of injury for each year in a sorted column plot. Bonus: Can you sort the plots within each facet?



(e) Use the `group_by` and `summarise` functions to calculate the average deaths per employee by NAICS code (i.e., industry code). Then, create a horizontal bar chart that shows the top 10 industries (excluding industries that occur less than 500 times in the dataset) with the highest number of cases per employee.



Solutions

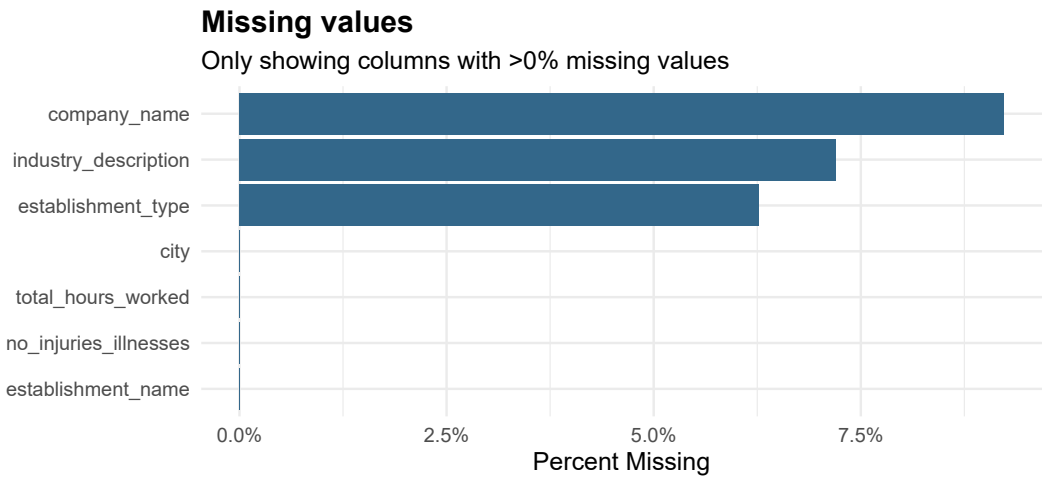
Task 1

(a)

```
glimpse(osha)
```

(b) Check for missing values in the data. Are there any problematic columns?

```
colMeans(is.na(osha)) %>%
  enframe() %>%
  filter(value > 0) %>%
  ggplot(aes(x = value,
             y = name %>% fct_reorder(value))) +
  geom_col(fill = "#33678A") +
  labs(title = "Missing values",
       subtitle = "Only showing columns with >0% missing values",
       x = "Percent Missing", y = NULL) +
  scale_x_continuous(labels = percent_format()) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```



(c)

```
osha %>% count(no_injuries_illnesses, sort = T)
osha %>% count(size, sort = T)
```

(d)

```
osha %>%
  select(contains("total")) %>%
  summary()
```

(e)

```
osha <- osha %>%
  mutate(total_cases = total_deaths + total_dafw_cases +
           total_djtr_cases + total_other_cases)
```

(f)

```
osha <- osha %>%
  filter(annual_average_employees > 10)
```

Task 2

(a)

```
osha <- osha %>%
  mutate(cases_per_employee = total_cases/annual_average_employees,
         deaths_per_employee = total_deaths/annual_average_employees)
```

```
osha %>%
  ggplot(aes(x=cases_per_employee)) +
  geom_histogram(bins = 100, fill = "#33678A") +
  labs(title = "OSHA cases per employee",
       subtitle = "Big outliers to the right that aren't even visible") +
  scale_x_continuous(labels = comma_format(),
                     breaks = 1:20) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```

(b)

Counting cases per employees that are higher than one:

```
osha %>%
  mutate(morethanone = cases_per_employee > 1) %>%
  count(morethanone)

# Alternatively, this is quicker:
# osha %>%
#   count(morethanone = cases_per_employee > 1)
```

Then dropping these observations:

```
osha <- osha %>%
  filter(cases_per_employee < 1)
```

```
osha %>%
  ggplot(aes(x=cases_per_employee)) +
  geom_histogram(bins = 100, fill = "#33678A") +
  labs(title = "OSHA cases per employee",
       subtitle = "After removing outliers") +
  scale_x_continuous(labels = comma_format()) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```

(c)

```
osha %>%
  group_by(year_filing_for) %>%
  summarise(mean_cases = mean(cases_per_employee, na.rm = T),
            median_cases = median(cases_per_employee, na.rm = T)) %>%
```

```
  pivot_longer(-year_filing_for) %>%
  ggplot(aes(x = year_filing_for, y = value, colour = name)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean and Average Number of OSHA Cases per Employee per Year",
       y = NULL, x = "Year", colour = NULL) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```

(d)

```
osha %>%
  select(year_filing_for, total_injuries, total_skin_disorders,
         total_respiratory_conditions, total_poisonings, total_hearing_loss,
         total_other_illnesses) %>%
  pivot_longer(-year_filing_for) %>%
  mutate(name = str_replace(name, "total_", "")) %>%
  count(year_filing_for, name, wt = value) %>%
  group_by(year_filing_for) %>%
  mutate(n = n/sum(n)) %>%
  ungroup() %>%
  ggplot(aes(x = n,
             y = name %>% reorder_within(n, year_filing_for))) +
  geom_col(fill = "#33678A") +
  facet_wrap(~ year_filing_for, scales = "free", ncol = 2) +
  labs(title = "Normalised type of injury by year",
       y = NULL, x = NULL) +
  scale_y_reordered() +
  scale_x_continuous(labels = percent_format(),
                     limits = c(0,1)) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```

(e)

```
osha %>%
  group_by(industry_description) %>%
  summarise(mean_per_employee = mean(deaths_per_employee, na.rm = T),
            n = n()) %>%
  filter(n > 500) %>%
  arrange(desc(mean_per_employee)) %>%
  head(10) %>%
  ggplot(aes(x = mean_per_employee,
             y = factor(industry_description) %>%
               fct_reorder(mean_per_employee))) +
  geom_col(fill = "#33678A") +
  labs(title = "Average deaths per employee\nby industry description",
       subtitle = "Only including top 10 industries\nwith more than 500 observations",
       y = NULL, x = NULL) +
  scale_x_continuous(labels = comma_format()) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```