Chlorophyll: group_by - filter - summarize - map

Introduction

In this exercise we handle a data set on phytoplankton. The goal is to identify stations that have long enough measurement series, and plot these. Furthermore we compute correlations between the main variable of interest (chorohpyll) and some a variable that can be a driver (temperature).

Further examples of how to handle data in R can be found in Chapter 5 of *Modern Statistics with R*.

The dataset `Chlorophyll.csv` contains chlorophyll-a, total phosphorus and water temperature for a number of lakes in Sweden[1].

Before starting, make sure you have installed and loaded the packages `tidyverse` and `magrittr`.

```
library(tidyverse)
library(magrittr)
```

Read and subset the phytoplankton dataset

Read the data set using \texttt{read_csv}:

```
chloro <- read_csv("Chlorophyll.csv")
```

To determine the number of lakes in the data we use `n_distinct`:

```
chloro %$%
  n_distinct(Station_name)
```

and an overall summary can be produced by

```
summary(chloro)
```

In this exercise we are interested in understanding the dynamics of chlorophyll-a. In the summary we see that many of the observations are missing (`NA`, meaning *Not Available*). Therefore we start by removing rows with missing values for chlorophyll-a. Depending of the aim of the analysis it is not always necessary or even desirable to remove missing values. Here it simplifies the selection of stations further down.

```
chloro_selected <- chloro %>%
  drop_na(`Chlorophyll-a`)
```

Exercise

1. Check how many lakes are left after removing rows that have missing chlorophyll-a data.

Select stations with more than 20 observations

Now, to make computations we decide to keep stations only if they have at least 20 observations. For this we first need to group by station and then filter out all series with more than 20 observations (`n()>20`).

[1]This data is part of the Swedish environmental monitoring of phytoplankton and water chemistry in lakes and accessible for everyone through SLU's open data, https://miljodata.slu.se/mvm/

```
chloro_selected <- chloro %>%
  drop_na(`Chlorophyll-a`) %>%
  group_by(Station_name) %>%
  filter(n() > 20)
```

Compute correlation coefficients separately for each station

Here we are interested in determining the correlation coefficient between water temperate and chlorophyll-a. To compute correlations for each station we need the `group_by(Station_name)` and the `summarize` together with `cor`:

```
chloro_correlations <- chloro %>%
  drop_na(`Chlorophyll-a`) %>% # Remove rows with missing chlorophyll-a data
  group_by(Station_name) %>% # Perform the next operation for each station
  filter(n() > 20) %>%      # Only keep stations with more than 20 observations
  summarize(cor_WT = cor(`Chlorophyll-a`, Temperature,
                         use = "pairwise.complete.obs"))
```

To see the correlations that were computed we can write

```
chloro_correlations
```

in the Console. Since this is a tibble only the first 10 rows are shown (in R Markdown you will get several pages, i.e you can see all correlations there). To see all correlations you can also find the tibble `cloro_correlations` in the Environment window to the right in RStudio, and double-click it. The same result is produced by

```
View(chloro_correlations)
```

or by directly piping the results into `View`.

```
chloro_correlations <- chloro %>%
  drop_na(`Chlorophyll-a`) %>% # Remove rows with missing chlorophyll-a data
  group_by(Station_name) %>% # Perform the next operation for each station
  filter(n() > 20) %>%      # Only keep stations with more than 20 observations
  summarize(cor_WT = cor(`Chlorophyll-a`, Temperature,
                         use = "pairwise.complete.obs")) %>%
  View()
```

As there are many lakes it can be an advantage to look at the stations with the highest correlations. Correlations are sorted in order of their (absolute) magnitude using `arrange`:

```
chloro_correlations <- chloro %>%
  drop_na(`Chlorophyll-a`) %>% # Remove rows with missing chlorophyll-a data
  group_by(Station_name) %>% # Perform the next operation for each station
  filter(n() > 20) %>%      # Only keep stations with more than 20 observations
  summarize(cor_WT = cor(`Chlorophyll-a`, Temperature,
                         use = "pairwise.complete.obs")) %>%
  arrange(cor_WT %>% abs %>% desc) %>%
  View()
```

Exercise

2. Add correlation between chlorophyll-a and total phosphorus by adding a `cor_TP=...` into the `summarize` statement.

Fit separate linear models to each station using `map`

To fit a linear regression model with Chlorophyll-a as response variable and Temperature as explanatory variable, we can use `lm`:

```
chloro_selected %>%
   lm(`Chlorophyll-a` ~ Temperature, data = .) %>%
  summary
```

If we just want to extract the model's beta coefficients we can do the following:

```
m <- chloro_selected %>%
   lm(`Chlorophyll-a` ~ Temperature, data = .)
m$coef
```

Now, let's say that we want to fit separate linear regression models to each station. To do so, we first `split` the data to create one dataset for each station, and then use `map` to apply `lm` to each of these datasets:

```
chloro_lm <- chloro %>%
  drop_na(`Chlorophyll-a`) %>% # Remove rows with missing chlorophyll-a data
  group_by(Station_name) %>% # Perform the next operation for each station
  filter(n() > 20) %>%       # Only keep stations with more than 20 observations
  split(.$Station_name) %>% # Split the dataset by station name
  map(~lm(`Chlorophyll-a` ~ Temperature, data = .)$coef) # Apply lm to each new dataset
```

To view the coefficients of the fitted models, type:

```
chloro_lm
```

Plot data

To plot the data we can use `ggplot` to create scatter plots. Here we have quite many stations and it might be reasonable to only plot a subgroup, e.g. those with high chlorophyll-a values (in this case: if the mean of chlorophyll at a station is higher than 15 µg/l).

```
chloro_selected <- chloro %>%
  drop_na(`Chlorophyll-a`) %>%
  group_by(Station_name) %>%
  filter(n() > 20) %>%
  filter(mean(`Chlorophyll-a`) > 15)
```

In the following figure we plot chlorophyll-a against water temperature using points. One plot is created for each station by the `facet_wrap` statement. Scales are free, which means that we do not force the limits for x- and y-axes to be the same for all stations.

```
chloro_selected %>%
  ggplot(aes(x = Temperature,
             y = `Chlorophyll-a`)) +
  geom_point() +
  facet_wrap(~Station_name,
             scales = "free")
```

To add regression lines to the plots, we can use `geom_smooth` after `facet_wrap`:

```
chloro_selected %>%
  ggplot(aes(x = Temperature,
             y = `Chlorophyll-a`)) +
  geom_point() +
```

```
  facet_wrap(~Station_name,
             scales = "free") +
  geom_smooth(method = "lm", color = "red")
```

Exercises: Subgroup correlations

The correlations observed before vary strongly from station to station. Part of this could be due to sampling frequency and seasonal variation. Let's check what happens if we choose only summer data, i.e. values measured in July.

3. Select data measured in July and series that have at least 5 observations and compute correlations for these series. Observe that by this you reduce the number of available lakes substantially.
4. It could also be interesting to study only lakes with high chlorophyll-a levels. Start with all data and filter out series with at least 20 observations and a mean chlorophyll-a level of 10. Compute correlation coefficients for these lakes.

Solutions to exercises

1.

```
chloro_selected %$%
  n_distinct(Station_name)
```

2.

```
chloro_correlations <- chloro %>%
  drop_na(`Chlorophyll-a`) %>% # Remove rows with missing chlorophyll-a data
  group_by(Station_name) %>% # Perform the next operation for each station
  filter(n() > 20) %>%      # Only keep stations with more than 20 observations
  summarize(cor_WT = cor(`Chlorophyll-a`, Temperature,
                         use = "pairwise.complete.obs"),
         cor_TP =    cor(`Chlorophyll-a`, `Total phosphorus`,
                         use = "pairwise.complete.obs"))
```

3.

```
chloro_correlations_July <- chloro %>%
  filter(!is.na(`Chlorophyll-a`) & Month == 7) %>%
  group_by(Station_name) %>%
  filter(n() > 5) %>%
  summarize(cor_TP = cor(`Chlorophyll-a`,
                                `Total phosphorus`,
                                use = "pairwise.complete.obs"),
         cor_WT = cor(`Chlorophyll-a`,
                                Temperature,
                                use = "pairwise.complete.obs")) %>%
  arrange(cor_WT)
```

4.

```
chloro_correlations_highchloro <- chloro %>%
  drop_na(`Chlorophyll-a`) %>%
  group_by(Station_name) %>%
  filter(n() > 20 & mean(`Chlorophyll-a`) > 10) %>%
  summarize(cor_TP = cor(`Chlorophyll-a`,
                         `Total phosphorus`,
                         use = "pairwise.complete.obs"),
            cor_WT = cor(`Chlorophyll-a`,
                         Temperature,
                         use = "pairwise.complete.obs")) %>%
  arrange(cor_WT)
```

Bonus: A heat map for correlations

Starting from the results of exercise 4, we restructure the data into one column (examples of how to use `gather` you can find in the instructions of "Glucose: gather - mutate - ifelse"). The plot uses red indicating high positive correlations and blue for high negative correlations.

```
chloro_corr2 <- chloro_correlations_highchloro %>%
  gather(variable, corr, -Station_name)

chloro_corr2 %>%
  ggplot(aes(variable, Station_name)) +
  geom_tile(aes(fill = corr),
            colour = "white") +
  scale_fill_gradient2(low = "darkblue",
                       mid = "white",
                       high = "red")
```