Lab Session 4: Data Cleaning and Analysis in R

Nichole Kulikowski

3/23/2021

Set working directory

We will first set a working directory. Your working directory should be where your plan to save your R code and also where the example datasets have been stored. In the "Files" tab, navigate to this folder. Once you are there, select the "More" dropdown and select "Set As Working Directory".

Install and load R packages

You should have already installed the tidyverse package. Now, you will need to load the package into R. This will allow you to use the functionality of the package.

```
library(tidyverse)
library(lubridate)
```

Load and view data

Load the example Facility ARI ".rds" file and save it as a data frame called "data". An .rds file is an R object and is the best way to store data that will be used in R.

```
data <- readRDS("session4_data/example_multi_facility_ari.rds")
```

View the first six observations in the facility dataset, to get a sense of the format.

```
head(data)
```

Data cleaning: renaming, quick fixes, and missing values

Rename the indicator variables to "ari" and "outpatient" and save as a new data frame. This will make them easier to work with (in R, we generally want to avoid leaving spaces in variables).

```
data %>%
  rename(ari="Number of Acute Respiratory Infections",
         outpatient="Total Outpatient Visits") -> data_new1
```

How many facilities are in the dataset and what are the facility names? Do you notice anything strange?

```
data_new1 %>%
  distinct(facility)
```

Fix the mistake in the facility name. Save as a new data frame and verify that the edit worked.

```
data_new1 %>%
  mutate(facility = case_when(facility == "CLINIC 1" ~ "Clinic 1",
                              TRUE ~ facility)) -> data_new2

data_new2 %>%
  distinct(facility)
```

How many months does each facility have data for?

```
data_new2 %>%
  group_by(facility) %>%
  dplyr::summarize(count=n())
```

What is the average (mean) monthly number of ARI cases in each facility?

```
data_new2 %>%
  group_by(facility) %>%
  dplyr::summarize(mean=mean(ari,na.rm=TRUE))
```

How many months are missing for each facility?

```
data_new2 %>%
  group_by(facility) %>%
  dplyr::summarize(count=n(),
                   missing=sum(is.na(ari)))
```

ACTIVITY: Summary statistics and plotting

1. Now that our dataset is clean, calculate the following summary statistics for each facility: median, minimum, maximum, 25th percentile, and 75th percentile. Remember to specify *na.rm=TRUE* for each summary statistic function (this will remove the empty cells).

```
data_new2 %>%
  group_by(facility) %>%
  dplyr::summarize(median=median(ari,na.rm=TRUE),
                   min=min(ari,na.rm=TRUE),
                   max=max(ari,na.rm=TRUE),
                   q25=quantile(ari,.25,na.rm=TRUE),
                   q75=quantile(ari,.75,na.rm=TRUE))
```

2. Using the ggplot function, plot the number of ARI cases by month from a facility of your choice. **Bonus:** how would you plot all four facilities together?

```
data_new2 %>%
  filter(facility == "Clinic 1") %>%
  ggplot(.,aes(x=date,y=ari)) +
  geom_point() +
  geom_line() +
  theme_bw()
```

3. Create a boxplot for the number of ARI cases from a facility of your choice.

```
data_new2 %>%
  filter(facility == "Clinic 1") %>%
  ggplot(.,aes(x=factor(0),y=ari)) +
  geom_boxplot() +
  theme_bw()
```

More advanced plotting

Plot the number of ARI cases by month for all four facilities using the *facet_wrap* function. Are there any outliers in these plots?

```
data_new2 %>%
  ggplot(.,aes(x=date,y=ari)) +
  geom_point() +
  geom_line() +
  facet_wrap(~facility,scales="free_y") + # tell them how to do this after.
  theme_bw()
```

There is one known data error in the plot: the value for March 2017 in Clinic 2 should be 20 (not 2000). Please fix this value and save as a new data frame.

```
data_new2 %>%
  mutate(ari = case_when(date == as.Date("2017-03-01") & facility == "Clinic 2" ~ 20,
                         TRUE ~ ari)) -> data_new3
```

With the updated dataset, plot the number of ARI cases by month for all four facilities using the *facet_wrap* function again. Did the fix work?

```
data_new3 %>%
  ggplot(.,aes(x=date,y=ari)) +
  geom_point() +
  geom_line() +
  facet_wrap(~facility,scales="free_y") + # tell them how to do this after.
  theme_bw()
```

ACTIVITY: IDENTIFYING OUTLIERS

1. Plot a boxplot for the number of ARI cases overall for each facility. You can either use the *facet_wrap* function or do it all in one plot.

```
data_new3 %>%
  ggplot(.,aes(x=facility,y=ari)) +
  geom_boxplot() +
  theme_bw()
```

2. Use Tukey's rule in the dataset to identify the outlier months in each facility for ARI cases. You should know the facility name, month, and number of ARI cases for each outlier. As a reminder, Tukey's rule defines an outlier point for the number of ARI cases for month t (y_t) as:

$$y_t < Q25 - 1.5 * IQR \text{ or } y_t > Q75 + 1.5 * IQR$$

You will need to compute $Q25$ (25th percentile), $Q75$ (75th percentile), and the interquartile range (IQR), which is defined as the difference between the 75th percentile and 25th percentile ($IQR = Q75 - Q25$). You can do this all in one tidyverse code chunk or you can split it into separate code chunks!

```
data_new3 %>%
  group_by(facility) %>%
  mutate(q25 = quantile(ari,.25,na.rm=TRUE),
         q75 = quantile(ari,.75,na.rm=TRUE)) %>%
  mutate(iqr = q75-q25) %>%
  mutate(flag = (ari < q25-1.5*iqr | ari > q75+1.5*iqr)) %>%
  filter(flag)
```

3. Recreate the number of ARI cases by month for all four facilities plot from earlier in the lab session, but add coloring for the points that were flagged as outliers. You can use the *scale_color_manual()* to change the point colors.

```
data_new3 %>%
  group_by(facility) %>%
  mutate(q25 = quantile(ari,.25,na.rm=TRUE),
         q75 = quantile(ari,.75,na.rm=TRUE)) %>%
  mutate(iqr = q75-q25) %>%
  mutate(flag = (ari < q25-1.5*iqr | ari > q75+1.5*iqr)) %>%
  ggplot(.,aes(x=date,y=ari)) +
  geom_point(aes(color=flag)) +
  geom_line() +
  facet_wrap(~facility,scales="free_y") +
  theme_bw() +
  scale_color_manual(values=c("pink","purple"))
```