DataFest R Workshop May 2020

Nathalie Moon (adapted from Sotirios Damouras' 2019 Workshop)

May 28, 2020

PART 1

Task 1.1

Create an R project & associated folder for the workshop

Task 1.2

Download and install the 'tidyverse' library

Task 1.3

Download the "dinesafe.csv" file from the workshop folder and save it in a project subfolder called "data". Then read the data into a data-frame called "dinesafe"

Task 1.4

Use the `glimpse()` function to take a look at the `dinesafe` data frame's structure

Task 1.5

Use the `View()` function to view the `dinesafe` data frame as a spreadsheet

Part 2

Task 2.1

Find all distinct establishment types. Hints: Are values of establishment in different rows or different columns? Which of the `dplyr` functions can you use to remove duplicated values?

Task 2.2

Find all inspections that took place on August 21st, 2018 (i.e. "2018-08-21"). Hints: (1) What variable contains the date of inspections? (2) Which `dplyr` function can you use to keep only observations for establishments that got inspected on this date?

Task 2.3

Find the total # of distinct inspections

Task 2.4

Rank establishment types by total amount fined

Task 2.5

Rank establishment types by average amount fined per establishment. Hints: (1) What variable might you want to group on? (2) How can you calculate the average amount fined in each category?

Task 2.6 (Challenging)

Find the establishment with the highest non-zero total fine amount whithin each establishment type. Hints: (1) Start by calculating the total fine for each establishment (think of what variable to group on to achieve this), then create new groups based on `ESTABLISHMENTTYPE` and use one of the `dplyr` functions to keep only the observations with the highest total in each of these new groups. Note - you'll need to use `ungroup` before regrouping the data into new groups.

PART 3

The file data/establishments.csv contains information on different establishments, in particular its neighborhood.

We will try to match this information with the dinesafe data.

Note that NOT ALL inspected establishments are present.

Task 3.1

Do an inner_join between the dinesafe and establishments tables. Hint: Which variable will you use to do the matching (it should be a variable which is present in both datasets)

Task 3.2

Use inner_join to rank the neighborhoods by the number of "C - Crucial" type infractions in restaurants. Hint: Either before or after joining, you'll need to use the `filter` function to keep only the observations we're interested in here (e.g. restaurants with inspection results of "C - Crucial"). After joining, think about which variable to group by before sorting.

Task 3.3

Find which (distinct) establishments did NOT get matched to a neighborhood. Hint: Use `anti_join`

PART 4

Our goal is to create a publication-quality graph with ggplot2. You will try to reproduce the Gapminder World Poster on World Health: https://www.gapminder.org/downloads/updated-gapminder-world-poster-2015/ using data in the gapminder package

```
# load data
#install.packages("gapminder")
library(gapminder)

# Take a look at the data
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> Afghanistan, Afghanistan, Afghanistan, Afghanistan, Afgha...
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asi...
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 199...
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 4...
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372,...
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.113...
```

Task 4.1

Create a scatter-plot of Life Expectancy (lifeExp) versus GDP-per-capita (gdpPercap), for 2007 data

Task 4.2

On your previous plot, change the x-axis (gdg/cap) to log-scale. Hint: look at the choices for "Scale" geometries on the `ggplot2` cheat sheet

Task 4.3

On your previous plot, change change the size of the points according to the population of each country

Task 4.4

On your previous plot, change the color of the points according to the continent

Task 4.5

On your previous plot, change the scale of each point to range from 1 to 14. Hint: Use the `scale_size(range = )` geometry to specify the range of sizes

Task 4.6

On your previous plot, label to each point with the name of the country; use the `geom_text` function with options `nudge_x = .02`, `alpha = .2` to make the labels readable - play around with values of `nudge_x` and `alpha` to figure out what these are doing!

Task 4.7

Modify the previous plot to make facets for each continent.

Task 4.8

If your x-axis labels are hard to read due to overlapping, you may want to rotate theme. You can use the `theme(axis.text.x=element_text(angle = 90, hjust = 0))` geometry to adjust this; change the values of `angle` and `hjust` to figure out the effect of these two arguments and find a combination that makes the labels easier to read.