CSC100/CSC200 Homework #5: Advanced Transformations

Caroline Hall

Fall 2021 | Data Science for the World | September 24, 2021

Please complete this notebook by filling in the cells provided. When you're done:

1. Remember to put your name in the header at the top of this notebook where it says `author`.
2. Select `Knit (Knit to Word)` from the toolbar menu.
3. Read that file! If any of your lines are too long and get cut off, we won't be able to see them, so break them up into multiple lines and knit again.
4. Save that Word document as a PDF file.
5. Submit BOTH this `.Rmd` file and the **PDF** file you generated to Gradescope. Some questions are autograded and you may improve your score on the tests given by resubmitting your work as many times as you like up to the deadline.
6. **Passing the automatic tests given does not guarantee full credit on any question.** The tests are provided to help catch some common mistakes, but it is *your* responsibility to answer the questions correctly.

If you cannot submit online, come to office hours for assistance. The office hours schedule appears on Blackboard.

This homework assignment is due **October 1 at 3:00PM**. Directly sharing answers is forbidden, but discussing problems with instructors and/or with classmates is encouraged.

Reading:

- Chapter 4 textbook

Run the cell below to prepare the notebook.

REMEMBER to use the *pipe* (`%>%`) operator whenever possible!

Part I: Billboard Top 100. The tibble `billboard` has song rankings for Billboard top 100 in the year 2000. We are told that five measurements (i.e., *variables*) are kept with each song: *artist*, *track*, *date* the song entered the top 100, the *week*, and the *rank* of the song in the corresponding week after it entered.

Let us inspect it:

```
billboard
```

Question 1. It should be evident that, based on the observational unit defined above, this data is *not* tidy. Cite at least two violations of the tidy data guidelines. Explain your reasoning.

The first reason this data isn't tidy is because of all the "NA"s from wk4 on. This violates tidy data rule #3: "Each Value must have its own cell." Similarly, wk76 has all "NA"s in its column. This violates tidy data rule #1: "Each variable forms a column."

Question 2. Using one of the pivot functions we saw in lecture (`pivot_longer` or `pivot_wider`), create a new tibble from `billboard` that has the following columns: `artist`, `track`, `date.entered`, `week`, and `rank`. Assign the new tibble to the name `billboard_tidy`.

```
billboard_tidy <- billboard %>%
  pivot_longer(wk1:wk76, names_to = "week", values_to = "rank")
billboard_tidy
```

```
## # A tibble: 24,092 × 5
##    artist track                   date.entered week   rank
##    <chr>  <chr>                   <date>       <chr> <dbl>
##  1 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk1      87
##  2 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk2      82
##  3 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk3      72
##  4 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk4      77
##  5 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk5      87
##  6 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk6      94
##  7 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk7      99
##  8 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk8      NA
##  9 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk9      NA
## 10 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk10     NA
## # … with 24,082 more rows
```

```
. = ottr::check("tests/billboard_q2.R")
```

```
## All tests passed!
```

Question 3. An immediate concern with `billboard_tidy` (as well as `billboard`) is the presence of missing values in the variable `rank`. Why are these present in the dataset?

These are present in the dataset because the rows with values for "artist", "track", "date.entered", and "week" must still be present when using the "pivot_longer" function, even though some of the corresponding "rank" values are still missing.

Question 4. Let us handle missing values in the variable `rank` by removing those rows from `billboard_tidy` (this strategy should seem reasonable based on your answer to the previous question). Assign the resulting tibble again to the name `billboard_tidy`.

```
billboard_tidy <- billboard_tidy %>%
  drop_na(rank)
billboard_tidy
```

```
## # A tibble: 5,307 × 5
##    artist  track                   date.entered week   rank
##    <chr>   <chr>                   <date>       <chr> <dbl>
##  1 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk1      87
##  2 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk2      82
```

```
##  3 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk3      72
##  4 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk4      77
##  5 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk5      87
##  6 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk6      94
##  7 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk7      99
##  8 2Ge+her The Hardest Part Of ... 2000-09-02   wk1      91
##  9 2Ge+her The Hardest Part Of ... 2000-09-02   wk2      87
## 10 2Ge+her The Hardest Part Of ... 2000-09-02   wk3      92
## # … with 5,297 more rows
```

```
. = ottr::check("tests/billboard_q4.R")
```

```
## All tests passed!
```

Question 5. Which track from 2000 spent the most weeks at #1? Your answer should be expressed as a tibble with a single row and two columns named `track` and `weeks_on_chart`. Assign your tibble to the name `top_track_most_weeks`.

HINT: To answer this, you should filter your `billboard_tidy` to include only those tracks that are at #1. Then think about how a `group_by()` and `summarize()` could help you. There are a few different ways to get to the answer.

```
top_track_most_weeks <- billboard_tidy %>%
  filter(rank == 1) %>%
  group_by(track) %>%
  summarize(weeks_on_chart = n()) %>%
  arrange(desc(weeks_on_chart)) %>%
  ungroup() %>%
  slice_head()
top_track_most_weeks
```

```
## # A tibble: 1 × 2
##   track                   weeks_on_chart
##   <chr>                            <int>
## 1 Independent Women Pa...             11
```

```
. = ottr::check("tests/billboard_q5.R")
```

```
## All tests passed!
```

Part II: U.S. presidential elections and unemployment. Consider the tibbles `pres_election2008` and `unemp2008` that have been loaded in for you. These datasets give county-level results for presidential elections in the USA from 2008, and the population and unemployment rate of all counties in the US. The data in `unemp2008` was prepared by USDA, Economic Research Service and we selected the relevant unemployment data for 2008.

Let us have a look at the data:

```
pres_election2008
```

```
unemp2008
```

An important variable in both datasets is the FIPS code. FIPS codes are numbers which uniquely identify geographic areas. Every county has a unique five-digit FIPS code. For instance, `12086` is the FIPS code that identifies Miami-Dade, Florida.

Question 1. Suppose that we want to create a new tibble that contains **both** the election results and the unemployment data. More specifically, we would like to add unemployment information to the election data by *joining* `pres_election2008` with `unemp2008`. Assign the resulting tibble to the name `election_unemp2008`.

HINT: What is the key we can use to join these two tables? Note that the column names may be different for the key in each table. For example: we would like to join on the key `student_id` but one table has a column `studentID` and the other `student_id`. In the join function we use, we can say `???_join(tibble_a, tibble_b, by = c("studentID" = "student_id"))`.

```
election_unemp2008 <- pres_election2008 %>%
  left_join(unemp2008, by = c("FIPS" = "FIPS_Code"))
election_unemp2008
```

```
## # A tibble: 6,308 × 11
##     year state   state_po county   FIPS candidate party candidatevotes
totalvotes
##    <dbl> <chr>   <chr>    <chr>   <dbl> <chr>     <chr>          <dbl>
<dbl>
##  1  2008 Alabama AL       Autauga  1001 Barack O… demo…           6093
23641
##  2  2008 Alabama AL       Autauga  1001 John McC… repu…          17403
23641
##  3  2008 Alabama AL       Baldwin  1003 Barack O… demo…          19386
81413
##  4  2008 Alabama AL       Baldwin  1003 John McC… repu…          61271
81413
##  5  2008 Alabama AL       Barbour  1005 Barack O… demo…           5697
11630
##  6  2008 Alabama AL       Barbour  1005 John McC… repu…           5866
11630
##  7  2008 Alabama AL       Bibb     1007 Barack O… demo…           2299
8644
##  8  2008 Alabama AL       Bibb     1007 John McC… repu…           6262
8644
##  9  2008 Alabama AL       Blount   1009 Barack O… demo…           3522
24267
## 10  2008 Alabama AL       Blount   1009 John McC… repu…          20389
24267
## # … with 6,298 more rows, and 2 more variables: State <chr>,
## #   Unemployment_rate_2008 <dbl>
```

```
. = ottr::check("tests/election_q1.R")
```

```
## All tests passed!
```

Question 2. Explain why the join function you selected (e.g., right join, left join, etc.) is appropriate for this problem. Why not choose another join function instead?

I selected the "left_join" function because I wanted to add Unemployment information to the Presidential Election rather than vice versa where I would have used "right_join".

Let us explore the relationship between candidate votes and unemployment rate for each state.

Question 3. Create a tibble from `election_unemp2008` that contains, **for each state**, only the candidate that received the most amount of votes. Assign the resulting tibble to the name `state_candidate_winner2008`. It should contain three variables: `state`, `candidate`, and `votes`. Here is what the first few rows of `state_candidate_winner2008` looks like:

state	candidate	votes
Alabama	John McCain	1266546
Alaska	John McCain	193841
Arizona	John McCain	1230111
…	…	…

```
state_candidate_winner2008 <- election_unemp2008 %>%
  group_by(state,candidate)%>%
  summarize("votes" = sum(candidatevotes)) %>%
  slice(which.max(votes))
```

```
## `summarise()` has grouped output by 'state'. You can override using the
`.groups` argument.
```

```
state_candidate_winner2008
```

```
## # A tibble: 51 × 3
## # Groups:   state [51]
##    state                candidate      votes
##    <chr>                <chr>          <dbl>
##  1 Alabama              John McCain  1266546
##  2 Alaska               John McCain   193841
##  3 Arizona              John McCain  1230111
##  4 Arkansas             John McCain   638017
##  5 California           Barack Obama 8274473
##  6 Colorado             Barack Obama 1288576
##  7 Connecticut          Barack Obama 1000291
##  8 Delaware             Barack Obama  255459
##  9 District of Columbia Barack Obama  245800
## 10 Florida              Barack Obama 4282366
## # … with 41 more rows
```

```
. = ottr::check("tests/election_q3.R")
```

```
## All tests passed!
```

The following tibble `unemp_by_state2008` gives an average unemployment rate for each state by averaging the unemployment rate over the respective counties.

```
unemp_by_state2008 <- election_unemp2008 %>%
  group_by(state) %>%
  summarize(avg_unemp_rate = mean(Unemployment_rate_2008, na.rm = TRUE))
unemp_by_state2008
```

```
## # A tibble: 51 × 2
##    state                avg_unemp_rate
##    <chr>                         <dbl>
##  1 Alabama                        7.22
##  2 Alaska                         6.4
##  3 Arizona                        7.65
##  4 Arkansas                       6.15
##  5 California                     8.65
##  6 Colorado                       4.62
##  7 Connecticut                    5.51
##  8 Delaware                       5.03
##  9 District of Columbia           6.8
## 10 Florida                        5.86
## # … with 41 more rows
```

Question 4. Create a new tibble that contains **both** the candidate winner voting data and the state-level average unemployment data. More specifically, we would like to add the state-level average unemployment data *to* the winner voting data by joining `state_candidate_winner2008` with `unemp_by_state2008`. Assign the resulting tibble to the name `state_candidate_winner_unemp2008`.

```
state_candidate_winner_unemp2008 <- left_join(state_candidate_winner2008,
unemp_by_state2008, by = "state")
state_candidate_winner_unemp2008
```

```
## # A tibble: 51 × 4
## # Groups:   state [51]
##    state                candidate      votes avg_unemp_rate
##    <chr>                <chr>          <dbl>          <dbl>
##  1 Alabama              John McCain  1266546           7.22
##  2 Alaska               John McCain   193841           6.4
##  3 Arizona              John McCain  1230111           7.65
##  4 Arkansas             John McCain   638017           6.15
##  5 California           Barack Obama 8274473           8.65
##  6 Colorado             Barack Obama 1288576           4.62
##  7 Connecticut          Barack Obama 1000291           5.51
##  8 Delaware             Barack Obama  255459           5.03
##  9 District of Columbia Barack Obama  245800           6.8
## 10 Florida              Barack Obama 4282366           5.86
## # … with 41 more rows
```

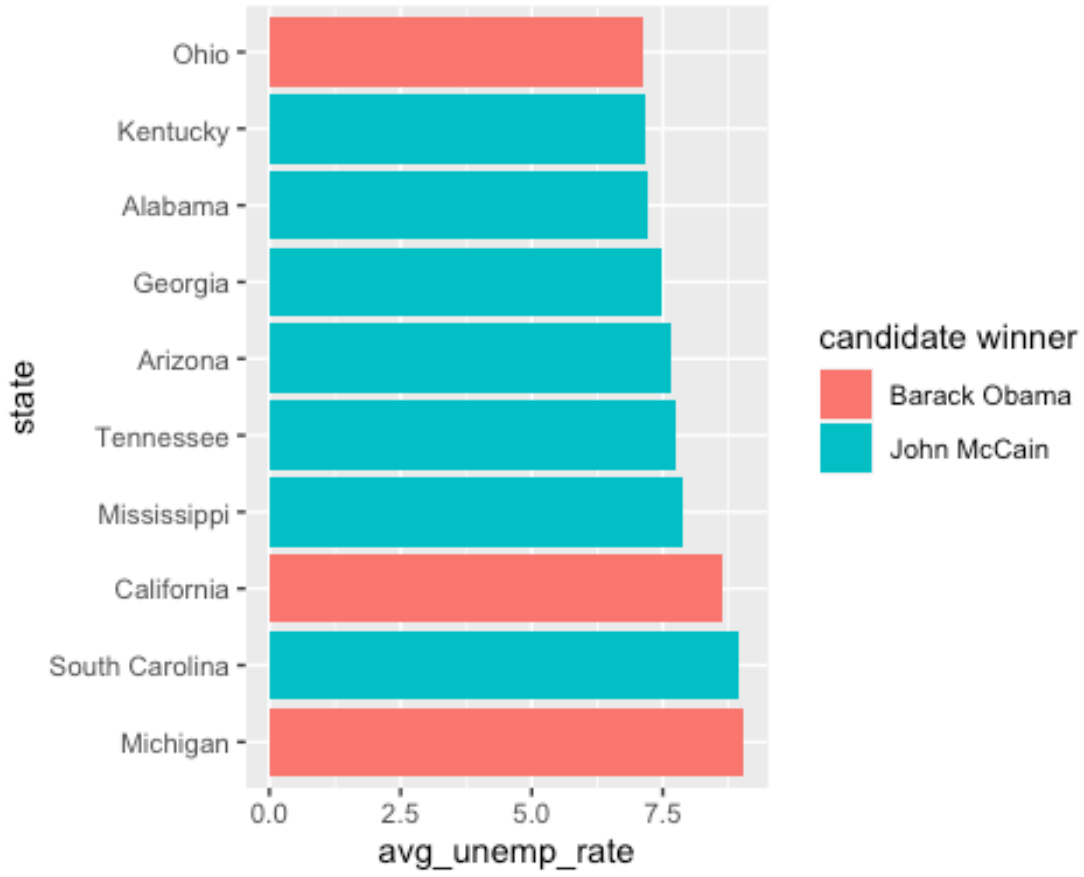
```
. = ottr::check("tests/election_q4.R")
```

```
## All tests passed!
```

The following cell produces a bar chart visualization from the tibble you just made. Do **NOT** worry about understanding any of the following code (visualization is next week :-). If there are any issues running the cell, that means there is something off with your `state_candidate_winner_unemp2008` tibble!

```
top_10 <- state_candidate_winner_unemp2008 %>%
  ungroup() %>%
  slice_max(avg_unemp_rate, n = 10)
state_order <- top_10 %>% pull(state)

ggplot(top_10) +
  geom_bar(aes(x = state, y = avg_unemp_rate, fill = candidate), stat =
"identity") +
  coord_flip() +
  scale_x_discrete(limits = state_order) +
  labs(fill = "candidate winner")
```



Question 5. Does the bar chart reveal any possible associations between average unemployment rate and the candidate that received the most votes in the top 10 states with the highest average unemployment rates?

Based on this bar chart, it does seem like there is an association between average unemployment rate and the candidate that received the most votes. It seems like this candidate was John McCain because of these 10 states he received 7/10, and Barack Obama only received 3/10. This may suggest that John McCain was the better candidate when it came to helping with unemployment.