````
---
title: 'Homework 2: graphics practice'
author: 'Last name, first name'
date: '`r format(Sys.Date(), "%Y, %B %d")`'
output: distill::distill_article
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(
  echo = TRUE,
  message = FALSE,
  error = FALSE,
  warning = FALSE)
```



# Preliminary



For this homework assignment, we'll continue exploring data related to our Citi Bike case study as a way to practice the concepts we've been discussing in class.

In our third discussion, we briefly considered an exploratory visualization of activity and docking station (im)balance, conducted in 2013 by Columbia University's Center for Spatial Research. [https://c4sr.columbia.edu/projects/citibike-rebalancing-study](https://c4sr.columbia.edu/projects/citibike-rebalancing-study).

As practice in understanding encodings, let's review and reconstruct one of the Center's graphics, titled: "CITI BIKE HOURLY ACTIVITY AND BALANCE". You can download and zoom in on a high resolution pdf of the graphic here: [https://c4sr.columbia.edu/sites/default/files/Activity_Matrix_Composite.pdf](https://c4sr.columbia.edu/sites/default/files/Activity_Matrix_Composite.pdf).



# Question 1(a) and 1(b) — data types and visual encodings



What variables and data types have been encoded?



> Write your answer here.



To what visual channels were those variables mapped?



> Write your answer here.
````

Question 2 — coordinate systems

What type of coordinate system was used for this *Activity and Balance* graphic? Explain.

> Write your answer here.

Question 3 — comparing encoded data

From our discussions, we listed several ways we can compare visually-encoded data, from more effective to less effective.

From the Center's *Activity and Balance* graphic, what type(s) of visual comparisons do the encodings enable? Explain.

> Write your answer here.

Question 4 — workflow, tidying and transforming data

Next, we will re-construct the main components of this graphic together. I'll setup most of the code, and you will fill in the needed gaps (I prompt you with a code comment) as your answers.

To get started, we will first load our main library,

```{r}
library(tidyverse)
```

and gather data from the New York City Bike Share data repository: https://ride.citibikenyc.com/system-data. The first time the code chunk below is run, it will download and save the zip file into your subdirectory you previously created called `data`, if the file hasn't already been saved. Then, we read in the `csv` file into an R data frame object we call `df`:

```{r}
savefile <- "data/201909-citibike-tripdata.csv"

if (!file.exists(savefile)) {
  url <- "https://s3.amazonaws.com/tripdata/201909-citibike-tripdata.csv.zip"
```

```
  download.file(url = url, destfile = savefile )
  }

df <- read_csv(savefile)
```

Next, we will *tidy* our data frame by renaming variables.

```{r}
df <- df %>% rename_with(~ gsub(' ', '_', .) )
```

Explore the data frame for missing data. You'll notice that some start and end station names are missing. We cannot reconstruct Columbia University Center for Spatial Research's graphic without these values, so we will filter those `NA` values out of our data frame, keeping in mind that our result is now conditional on the data we still have. We also want to just consider observations with an `end_station_name` that is also used as a `start_station_name`.

```{r}
df <-
  df %>%
  filter(
    if_any(contains('station_name'), ~ !is.na(.)),
    end_station_name %in% start_station_name
  )
```

We need to change the structure of our data so that we can map data values onto the visual encodings used in the Center's graphic.

More specifically, we need to know the number of rides both starting and ending at each station name at each hour of the day, averaged over the number of days in our data set. We'll need to create new variables and pivot some of the data. Specifically, we will create a variable for day of month (`day`) and hour of day (`hour`) from the existing variable `starttime`. Then, we will pivot two variables — `start_station_name` and `end_station_name` into long format, like so:

```{r}
df <-
  df %>%
  mutate(
    day = format(starttime, "%d"),
    hour = format(starttime, "%H")
  ) %>%
  pivot_longer(
    cols = c(start_station_name, end_station_name),
    names_to = "start_end",
    values_to = "station_name"
  ) %>%
  mutate(
    station_name = fct_reorder(station_name, desc(station_name))
  )
```

The pivot results in creating separate observations, from the perspective of a docking station (instead of the perspective of a ride), for both types of events: *a bike parking and a bike leaving*.

Are you starting to see that tidying and transforming data are frequently useful prerequisites to making interesting graphics? Hint, the correct answer is "Yes, and this is awesome!"

> Write your answer here.

Question 5 — transforming data

With the pivoted data frame, we can now group our data by station name and hour, and calculate the averages we'll need to map onto visual variables.

Create new variables `activity` and `balance`, where `activity` holds the average number of rides or observations at each station name each hour and where `balance` hold the average difference between rides beginning at the station and rides ending at the station.

```{r}
df <-
  df %>%
  group_by(station_name, hour, .drop = FALSE) %>%

  summarise(
    activity = # complete this code
    balance  = # complete this code
  ) %>%

  ungroup()
```

Inspect this data frame, and compare with the original imported data frame to understand how each step of the above code changed its structure. Start to consider how we will map these data variables onto the visual variables used in the Center's *Activity and Balance* graphic.

In our third discussion, we considered how to scale data values to map their ranges to the appropriate ranges for each channel of color: hue, chroma (saturation), and luminance. We'll do that next.

Question 6 — scaling data

Complete the code below to properly scale your data variables to the ranges of your visual variables. To get you started, I've written the following code:

```{r}
library(scales)

df <-
  df %>%
  mutate(
    hue = ifelse(balance < 0, 50, 200),
    saturation =
      rescale(
        abs(balance),
        from = # complete this code
        to   = # complete this code
      ),
    luminance =
      rescale(
        activity,
        from = # complete this code
        to   = # complete this code
      )
  )
```

Question 7 — mapping data to visual channels

Finally, we are ready to map our data onto the visual variables. The Center's *Activity and Balance* graphic resembles a so-called *heatmap*.

Use the grammar of graphics to create tiles of information, using the function `geom_tile`. To do that, first review the help file for that function, paying particular attention to the aesthetics you'll need to specify.

Further, to map the individual channels of color, you can use the function `hcl` that's already loaded from `tidyverse`, which works very similarly to (a bit less optimal than) the example I showed you from my R package, `hsluv_hex`. You may also use mine, but that will require you to install it.

I've started the code for you below. Add code where prompted.

```{r}
p <-
  df %>%
  ggplot() +
  scale_fill_identity() +
  geom_tile(
    mapping = aes(
      # complete this code
    ),
    width = 0.95,
    height = 0.95
  ) +
  theme_dark() +
  theme(
```

```
    panel.background = element_blank(),
    panel.grid = element_blank(),
    plot.background = element_rect(fill = "#333333"),
    axis.text.x = element_text(color = "#888888", size = 16 / .pt),
    axis.text.y = element_text(color = "#888888", size =  7 / .pt)
  ) +
  labs(x = "", y = "")

# The next line of code will save the graphic as a pdf onto your working
# directory so that you can separateely open and zoom in while reviewing
it.
ggsave("activity_balance.pdf", plot = p, width = 8, height = 40)

p
```

Question 8 — decoding and interpretation: critical thinking

We've finished roughly reconstructing the Center's Activity and Balance graphic, updated with later data from September 2019, six years after the original graphic but still before the pandemic. We find that the patterns originally described by the Center still show up. Review their description of the Activity and Balance graphic.

Notice that the Center's description of its graphic and data do not, however, discuss whether empty and full docking stations, and rebalancing efforts by Citi Bike, have any effect on the patterns they describe.

How might 1) empty and full docking stations and 2) CitiBike rebalancing bikes affect the visual patterns in our graphic?

> Write your answer here.

Bonus — advanced practice

Citi Bike has monthly data up to the present, though they have changed a few of the variable names in the csv files. Repeat the above importing, tidying, transforming, and visualizing for data last month, September 2021 (post pandemic).

Compare the patterns you see in the graphic for September 2021 with the above graphic for September 2019. Explain how the patterns differ (on a high level), and your best reasoning as to why.

```{r}
# hint, most of the code is identical to the above code
```

```

> Write your answer here.

# Knit and submit

Knit your completed r markdown file (this one) into an html file, name the files `` and `` respectively. Then, submit both files onto canvas.
```