STAT 479 HW2

Instructions

1. Submit your solutions to the exercises below before **February 26 at 11:59pm CST**.
2. Prepare your solutions as Rmarkdown documents. For problems in vega-lite, copy your code into the `.Rmd`, making sure to enclose them within `js` blocks,

````
```js
// your code here
```
````

so that the syntax is properly highlighted, e.g.,

```
// your code here
```

3. For vega-lite problems, include a screenshot of your result.
4. We give example figures below to guide your work. However, view these only as suggestions – we will keep an eye out for improvements over our plots.
5. Include two files in your submission, (a) a pdf of the compiled `.Rmd` file and (b) the original `.Rmd` file.

Rubric

2 problems below will be graded according to,

- Correctness: For plots, the displays meet the required specifications. For conceptual questions, all parts are accurately addressed.
- Attention to detail: Writing is clear and designs are elegant. For example, no superfluous marks are included, all axes are labeled, text is neither too small nor too large.
- Code quality (if applicable): Code is concise but readable, properly formatted and commented.

The remaining problems will be graded for completeness.

Problems

(1) Plant Growth Experiment

This problem will give you practice with tidying a dataset so that it can be easily visualized. The data describe the height of several plants measured every 7 days. The plants have been treated with different amounts of a growth stimulant. The first few rows are printed below – `height.x` denotes the height of the plant on day `x`.

```
library("readr")
plants <- read.csv("https://uwmadison.box.com/shared/static/qg9gwk2ldjdtcmmmiropcunf34ddonya.csv")
```

a. Propose an alternative arrangement of rows and columns that conforms to the tidy data principle.

b. Implement your proposed arrangement from part (a).

c. Using the dataset from (b), create a version of Figure 1, showing the growth of the plants over time according to different treatments.

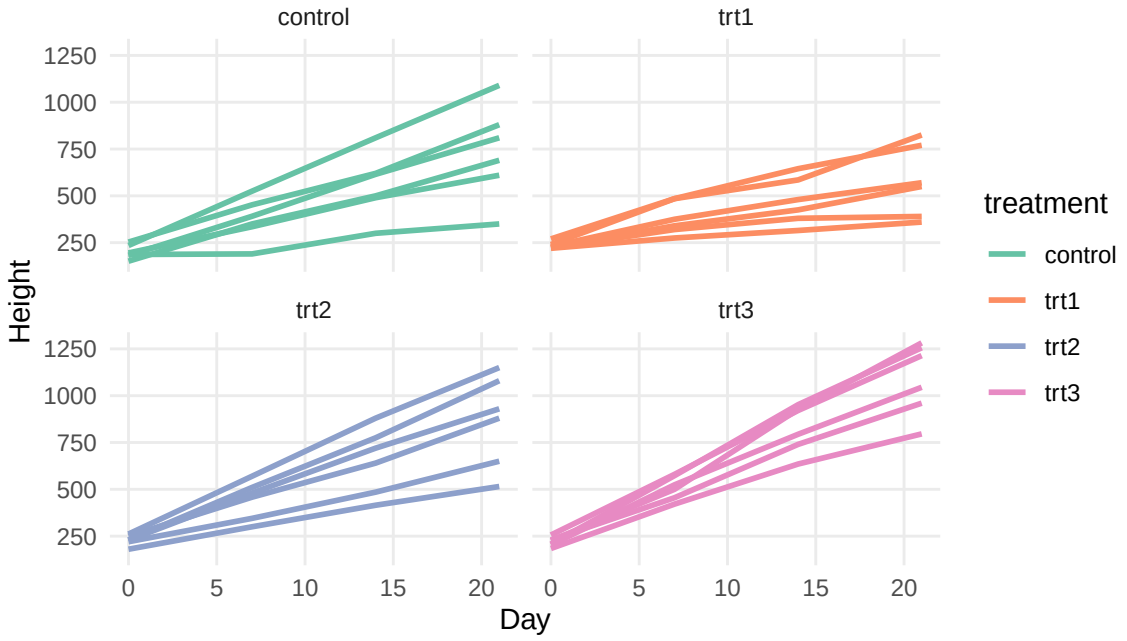

Figure 1: An example result for Problem (1c), showing plant growth according to different treatments.

(2) California Wildfires

In this problem, we will interactively visualize a dataset giving statistics of recent California wildfires. The steps below guide you through the process of building this visualization. Make sure to include your code, a screenshot, and a brief explanation of what you did for each step.

a. [Static version] Plot the day of the year that each fire started against the county within which it was located. Use the size of the mark to encode the number of acres burned in each fire. Sort the counties according to the average latitude of the fires within it. At this point, your figure should look something like Figure 2.

b. [Interactive] Provide a tooltip so that the name of the fire can be identified by hovering over the points. Introduce a slider to interactively highlight selected years. An interactive version is linked in the caption to Figure 2. *Hint*: The conditional encoding examples from Week 3 - 1 and the slider example from Week 1 - 3 may be useful references.

c. What have you learned from this visualization? Is there additional information that is not described by this visualization or dataset that you think would enrich your interpretation?

(3) Pokemon

This problem gives practice in deriving new variables to improve a faceted plot. The data below give attack and defense statistics for Pokemon, along with their types. We will build a visualization to answer the question – how do the different types of Pokemon vary in their attack and defense potential?

```
pokemon <- read_csv("https://uwmadison.box.com/shared/static/hf5cmx3ew3ch0v6t0c2x56838er1lt2c.csv")
```

a. Derive a new column containing the attack-to-defense ratio, defined as $\frac{\text{Attack}}{\text{Defense}}$.

b. For each `type_1` group of Pokemon, compute the median attack-to-defense ratio.

c. Plot the attack vs. defense scores for each Pokemon, faceted by `type_1`. Use the result of (b) to ensure that the panels are sorted from types with highest to lowest attack-to-defense ratio. Your result should look similar to Figure 3.

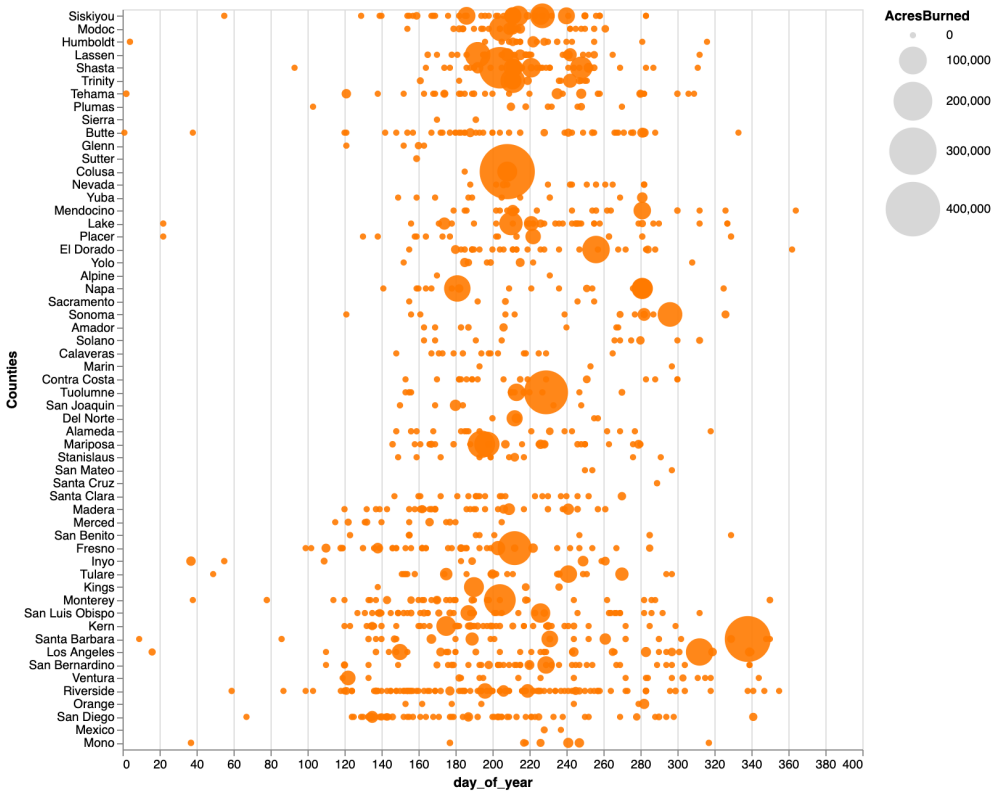

Figure 2: An example result for (2a). An example interactive version for (2b) is visible here.

d. Propose, but do not implement, a visualization of this dataset that makes use of dynamic queries. What questions would the visualization answer? What would be the structure of interaction, and how would the display update when the user provides a cue?

(4) NYC Airbnb Data

In this problem, we'll create a visualization to dynamically query a dataset of Airbnb rentals in Manhattan in 2019. The steps below guide you through the process of building this visualization. Make sure to include your code, a screenshot, and a brief explanation of what you did for each step.

a. Make a scatterplot of locations (Longitude vs. Latitude) for all the rentals, colored in by `room_type`.

b. Make a histogram of the log-prices[1], with stacked colors to distinguish between room types. Vertically concatenate this histogram with the scatterplot from (a). An example of the resulting display is shown in Figure 4.

c. Introduce a selection so that brushing over price ranges highlights the associated rentals on the map. *Hint*: This is similar to the movie-ratings-over-time visualization from Week 3 [2].

d. Comment on the resulting visualization. If you had a friend who was interested in renting an Airbnb in NYC, what would you tell them?

(5) Imputing Housing Data

This problem gives practice visualizing missing-data imputation. We will use another housing-themed dataset, this one describing homes for sale in Melbourne. Notice that the `BuildingArea` and `YearBuilt` variables have many missing values.

[1] We've applied a log-transform to the original rental prices because otherwise the prices are highly skewed.

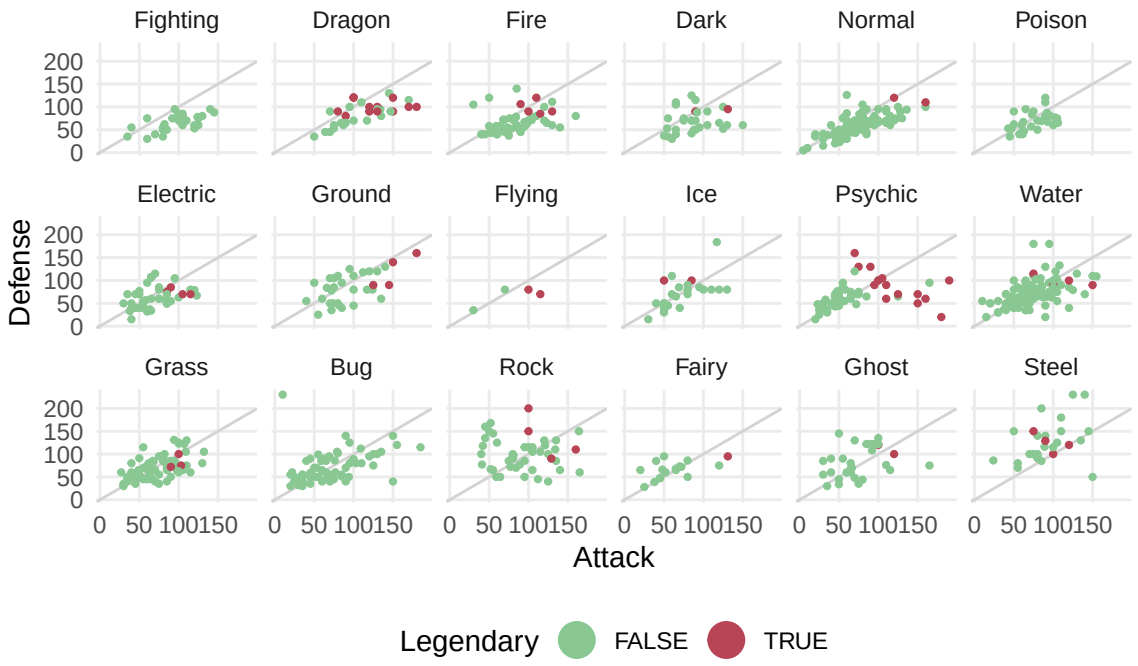

Figure 3: An example result for Problem (3c), showing attack vs. defense statistics for different Pokemon.

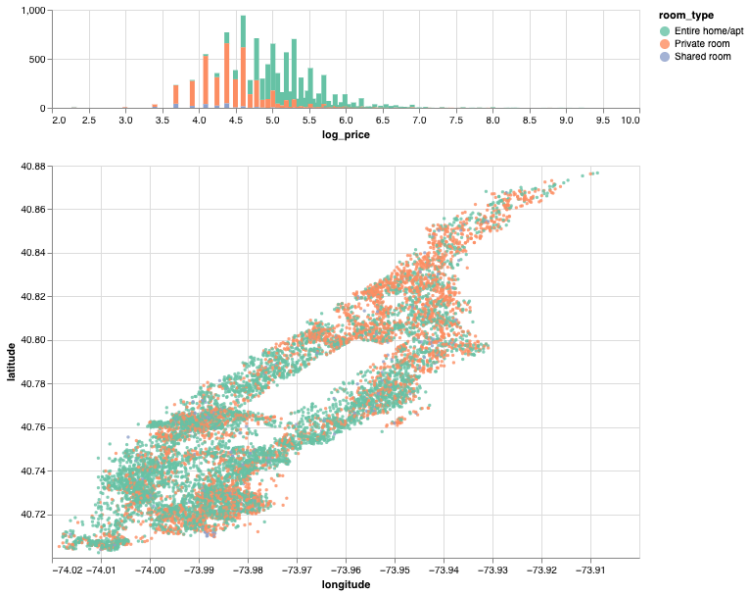

Figure 4: An example of the static visualization for problem (4b). An short video of the result from (4c) can be viewed here.

```
housing <- read_csv("https://uwmadison.box.com/shared/static/h5u176syp4xkret4w89n70efsp1tubex.csv")
```

a. Using `geom_miss_point`, make a plot of `Price` against `BuildingArea`, faceted by the region from which the home is located. Make sure the observations with missing `BuildingArea` values are still displayed somewhere on the plot, as in Figure 5. What does the plot suggest about how you might impute the `BuildingArea` column?

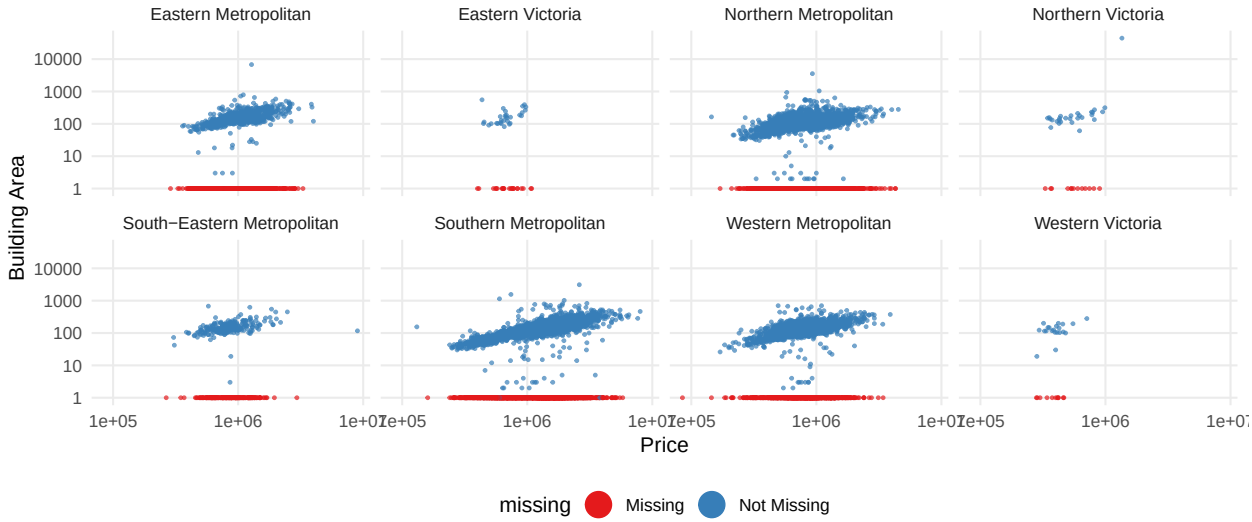

Figure 5: An example result for problem (5c), plotting observed and missing BuildingArea values against Price.

b. Using the `impute_lm` function, impute the `BuildingArea` variable. You may use whichever fully measured columns that you like – using `BuildingArea ~ Price` is a reasonable starting point.

c. Recreate the visualization from part (a), but with the imputed data included. Make sure to distinguish between values that were truly measured and those that were imputed in part (b). Comment on the quality of the results. Do you notice anything unusual about the imputations in Northern Victoria?

Feedback

a. How much time did you spend on this homework?
b. Which problem did you find most valuable?