STAT 479 HW2

(1) Plant Growth Experiment

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

Part A

Because the height of the plants is measured in separate columns, I want to pivot height.0, height.7, height.14, and height.21 to all be in one days column. Then also translate the respective values for height at each of those time periods. Instead of having days represented as characters, I alsow want to convert days to an integer.

Part B

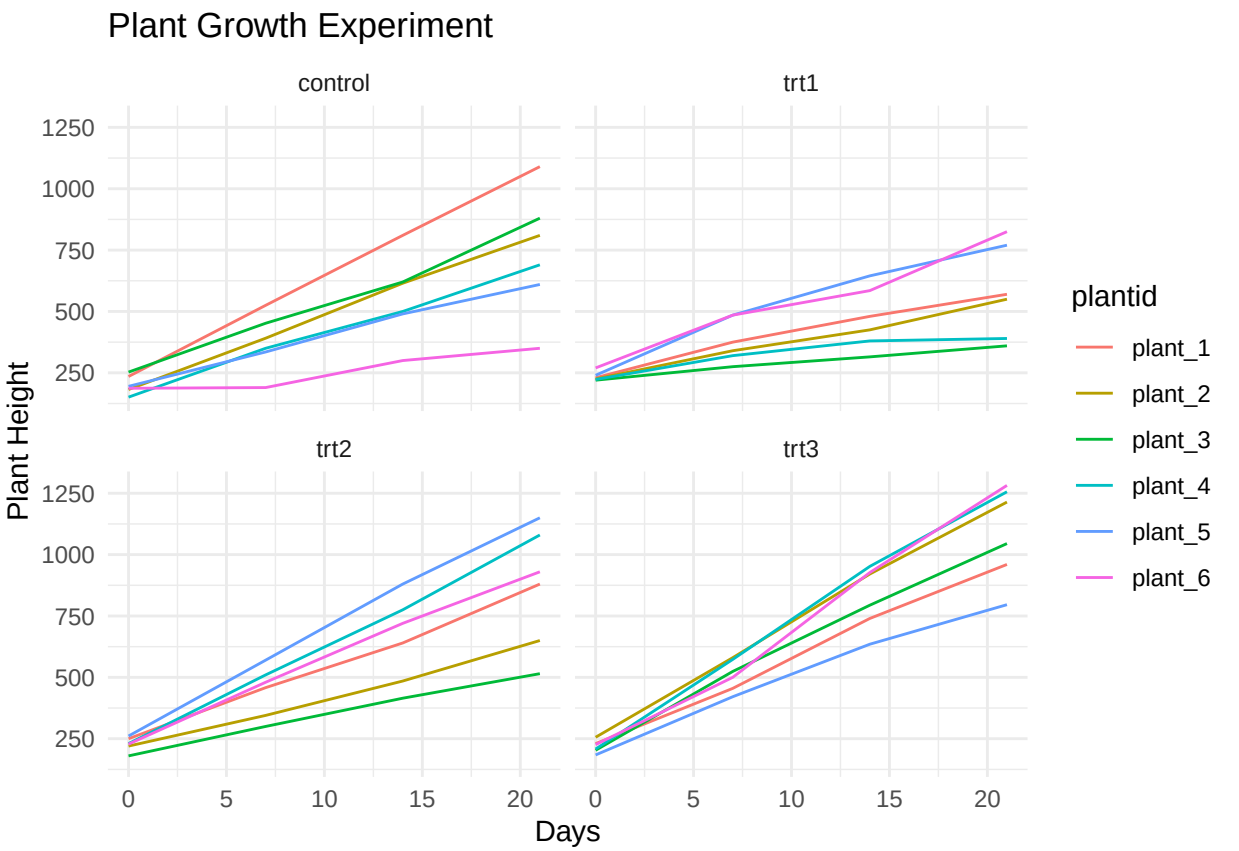
```
plants = plants %>% pivot_longer(c(`height.0`, `height.7`, `height.14`, `height.21`), names_to = "heigh

plants = select(plants, -height)

plants$days = as.integer(plants$days)
```

Part C

```
ggplot(plants) +
  geom_line(aes(x = days, y = value, col = plantid)) +
  facet_wrap(. ~ treatment, ncol = 2) +
  labs(
    x = "Days",
    y = "Plant Height",
    title = "Plant Growth Experiment"
  )
```



(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 screen shot, 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.

```
fire = read.csv("fires@5.csv")
```

```
sort = fire %>% group_by(Counties) %>% summarise(lat = median(Latitude, na.rm = T))
sor = fire %>% group_by(Counties) %>% summarise(lat = mean(Latitude, na.rm = T))
```

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 data set that you think would enrich your interpretation?

(3) Pokemon

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

Part A

```
pokemon = pokemon %>% mutate(rate = Attack / Defense)
```

Part B

```
pokemon %>% group_by(type_1) %>% summarise(avg_rate = median(rate))
```

```
## # A tibble: 18 x 2
##    type_1   avg_rate
##  * <chr>       <dbl>
##  1 Bug         0.968
##  2 Dark        1.29
##  3 Dragon      1.38
##  4 Electric    1.10
##  5 Fairy       0.956
##  6 Fighting    1.57
##  7 Fire        1.33
##  8 Flying      1.06
##  9 Ghost       0.943
## 10 Grass       0.994
## 11 Ground      1.08
## 12 Ice         1.06
## 13 Normal      1.23
## 14 Poison      1.15
## 15 Psychic     1
## 16 Rock        0.962
## 17 Steel       0.75
## 18 Water       1
```

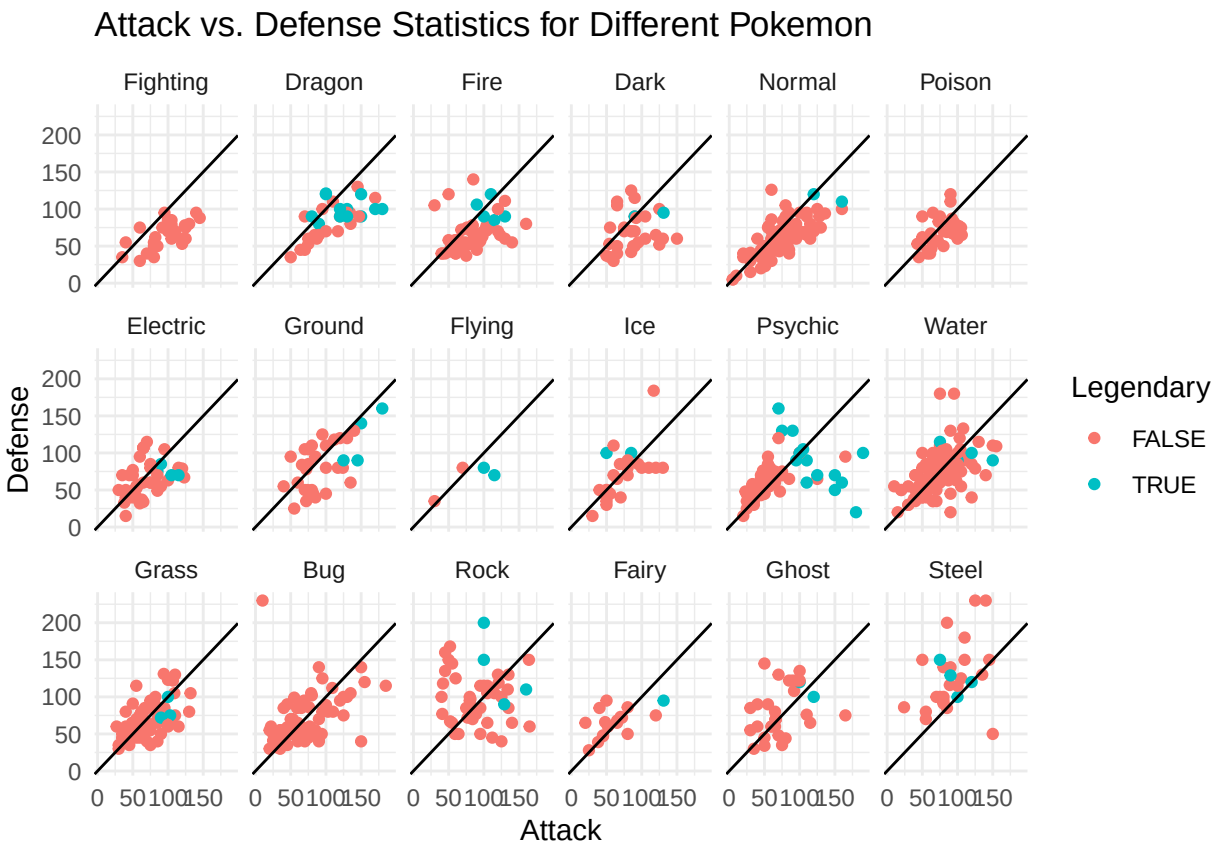
Part C

```
type1_order <- pokemon %>%
  group_by(type_1) %>%
  summarise(avg_value = median(rate)) %>%
  arrange(desc(avg_value)) %>%
  pull(type_1)

pokemon = pokemon %>% mutate(type_1 = factor(type_1, levels = type1_order))

ggplot(pokemon) +
```

```
  geom_point(aes(x = Attack, y = Defense, col = Legendary)) +
  facet_wrap(. ~ type_1, ncol = 6) +
  geom_abline()+
  labs(title = "Attack vs. Defense Statistics for Different Pokemon")
```



d. Propose, but do not implement, a visualization of this data set 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

```
airbnb = aq.fromCSV( await FileAttachment("airbnb.csv").text() )
```

Part A

I based my code for this question off of the movie-ratings-over-time visualization from Week 3[2]. I knew in the later parts of the question I would need to add a histogram and I wanted my code for part a to be compatible with the other parts, so I started with a histogram and scatter plot. This was to ensure my syntax matched as I went through the various parts of this question. So I actually did part b first, then deleted the histogram so the graph would work for part a.

```
{
  const location = vl.markCircle({size: 10})
    .data(airbnb)
    .encode(
      vl.x().fieldQ('longitude').scale({domain: [-74.02,-73.9]}),
      vl.y().fieldQ('latitude').scale({domain: [40.7,40.9]}),
      vl.color().fieldN('room_type').scale({scheme: 'tableau20'})
    )
    .width(500)
    .height(350);
  return vl.vconcat(location).render();
}
```

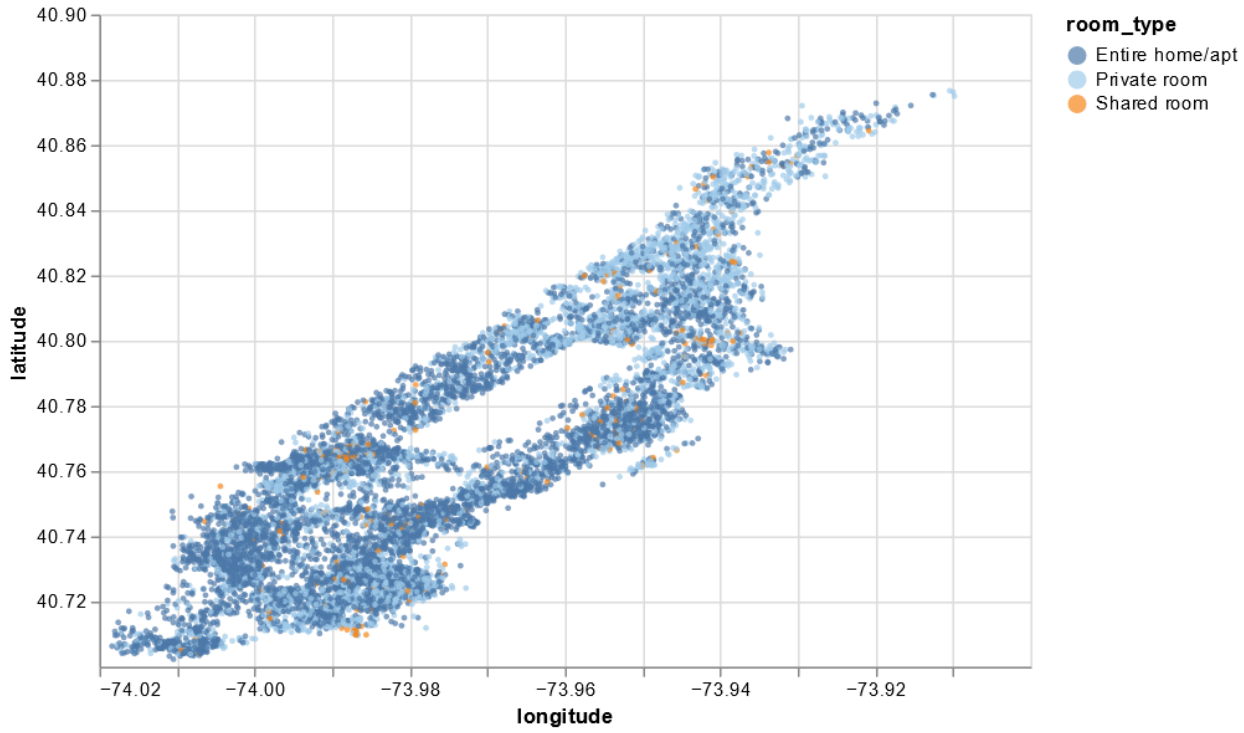


Figure 1: Q4 Part A

Part B

As stated in part a, I started with part b and based if off code from the movie-ratings-over-time visualization from Week 3[2]. I thought I had some big code error initially since it looked like only one point was displaying. After some research and looking at the example graph I found out that I needed to extremely zoom in on the graph, and then it would make sense. I additionally changed the colors for the scale on room type, so it would be easier to distinguish between them.

```
{
  const brush = vl.selectInterval()
    .encodings('x'); // limit selection to x-axis (year) values
  const price = vl.markBar({width: 4})
```

```
    .data(airbnb)
    .select(brush)
    .encode(
      vl.x().fieldQ('log_price').title('Log Price'),
      vl.y().count().title(null),
      vl.color().fieldN('room_type')
    )
    .width(500)
    .height(80);
  const location = vl.markCircle({size: 10})
    .data(airbnb)
    .encode(
      vl.x().fieldQ('longitude').scale({domain: [-74.02,-73.9]}),
      vl.y().fieldQ('latitude').scale({domain: [40.7,40.9]}),
      vl.color().fieldN('room_type').scale({scheme: 'tableau20'}),
    )
    .width(500)
    .height(350);
  return vl.vconcat(price, location).render();
}
```

Part C

Implementing part c was the easiest part of this question, since I only needed one more line of code. In the scatter plot I used vl.opacity() to add a brush that would highlight selected values at 75% opacity and have the rest of the values be 5% opacity.

```
{
  const brush = vl.selectInterval()
    .encodings('x');
  const price = vl.markBar({width: 4})
    .data(airbnb)
    .select(brush)
    .encode(
      vl.x().fieldQ('log_price').title('Log Price'),
      vl.y().count().title(null),
      vl.color().fieldN('room_type')
    )
    .width(500)
    .height(80);
  const location = vl.markCircle({size: 10})
    .data(airbnb)
    .encode(
      vl.x().fieldQ('longitude').scale({domain: [-74.02,-73.9]}),
      vl.y().fieldQ('latitude').scale({domain: [40.7,40.9]}),
      vl.color().fieldN('room_type').scale({scheme: 'tableau20'}),
      vl.opacity().if(brush, vl.value(0.75)).value(0.05)
    )
    .width(500)
    .height(350);
  return vl.vconcat(price, location).render();
}
```

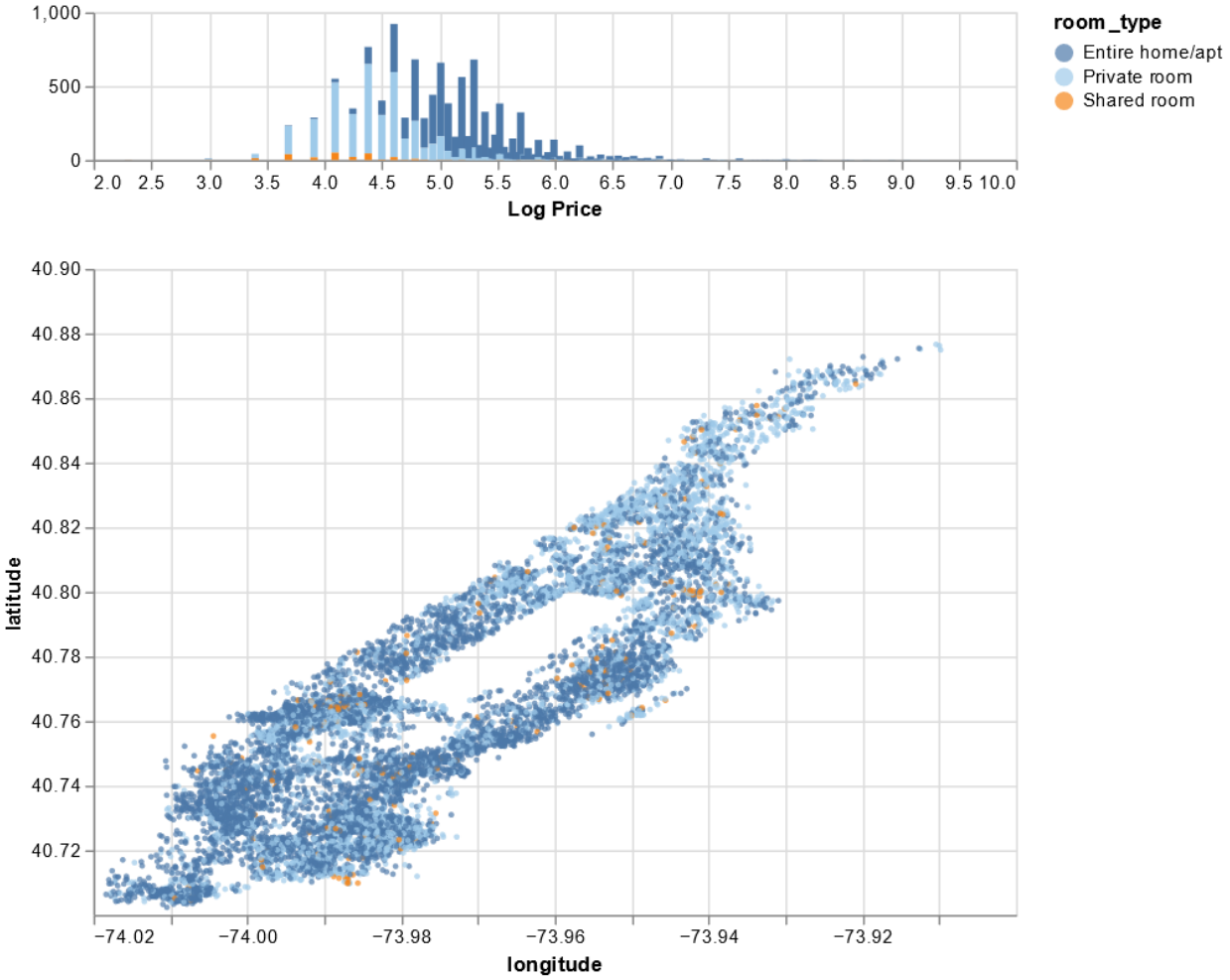


Figure 2: Q4 Part B

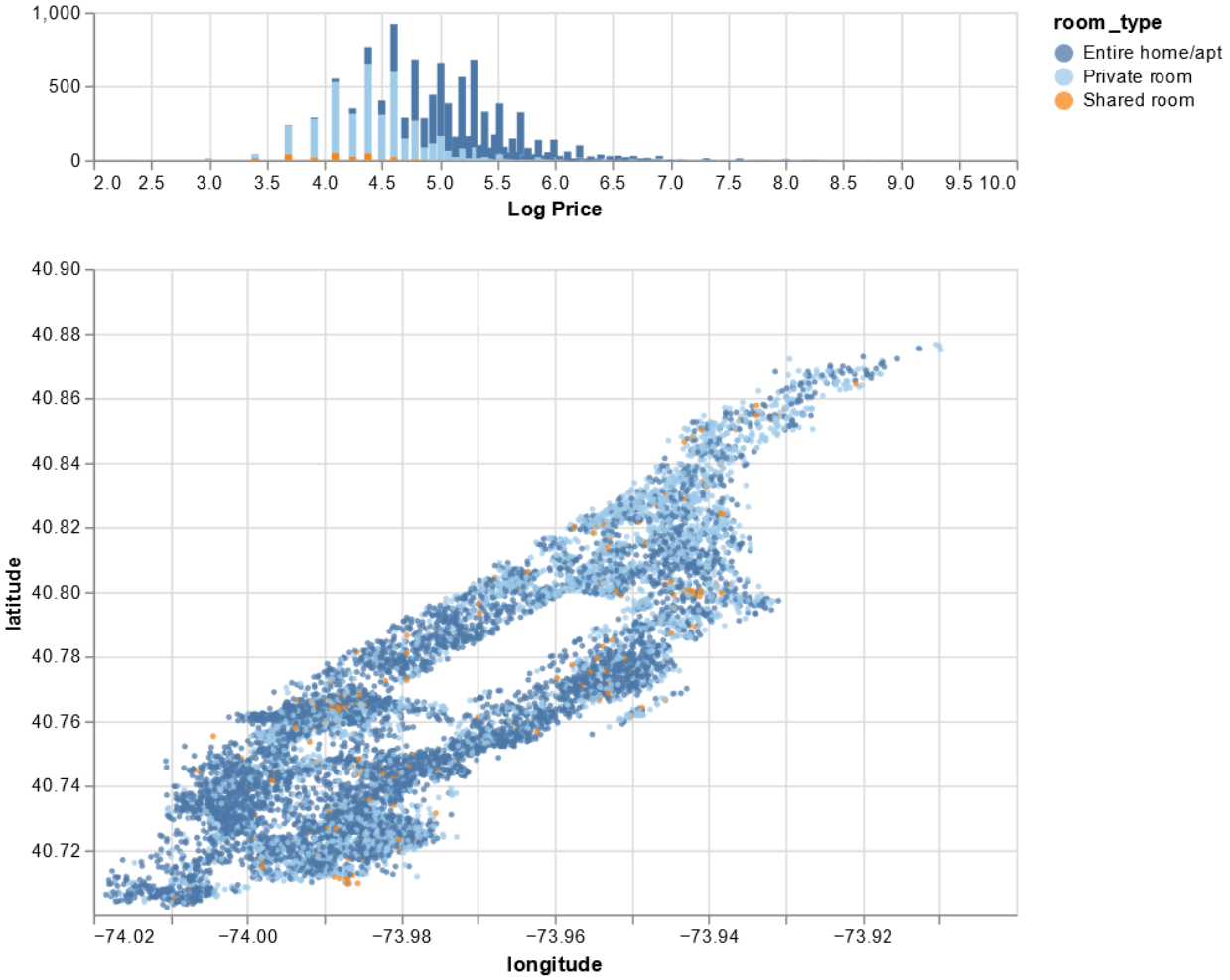


Figure 3: Q4 Part C

Part D

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?

First off, I know New York has some very strict laws and regulations regarding airbnb's and similar services, so I would encourage them to do plenty of research and refer them to this link.

(5) Imputing Housing Data

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

```
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?

b. Using the `impute_lm` package, 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? three hours so far
b. Which problem did you find most valuable?