STAT 479 HW1

Yifan Zhang

Problems

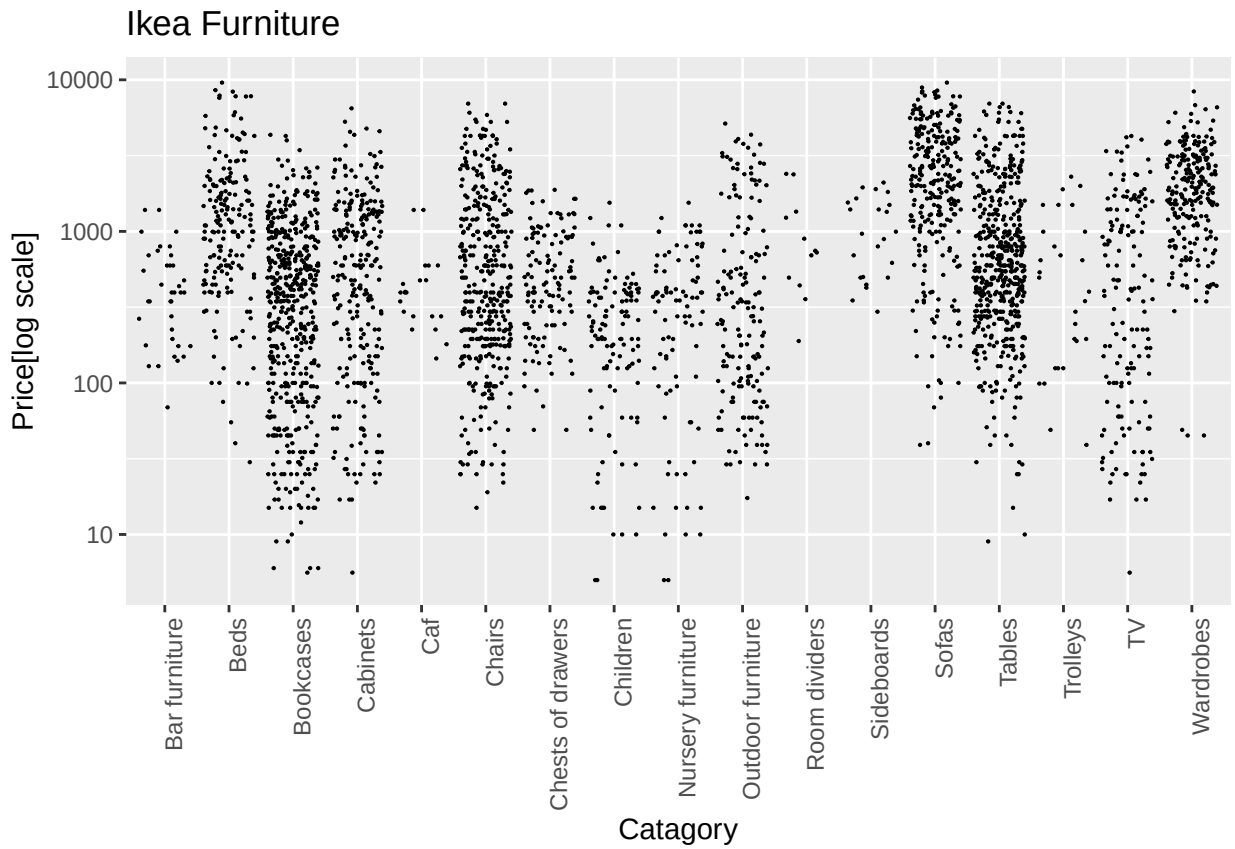
(1) Ikea Furniture

The dataset below shows prices of pieces of Ikea furniture. We will compare prices of different furniture categories and label the (relatively few) articles which cannot be bought online.

```
library("readr")
ikea <- read_csv("https://uwmadison.box.com/shared/static/iat31h1wjg7abhd2889cput7k264bdzd.csv")
```

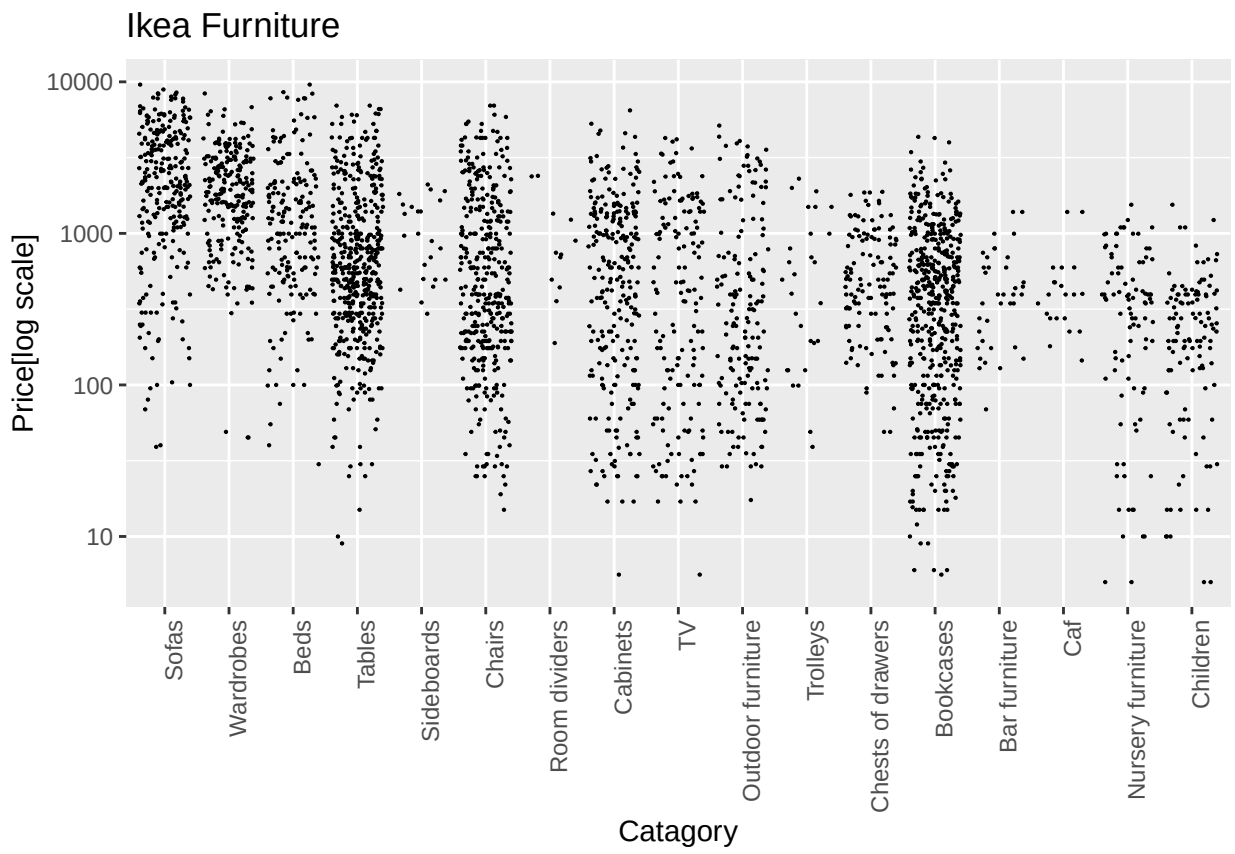
a. Make a plot that shows the relationship between the `category` of furniture and `price` (on a log-scale). Show each `item_id` as a point – do not aggregate to boxplots or ridgelines – but make sure to jitter and adjust the size the points to reduce the amount of overlap. *Hint: use the `geom_jitter` layer.*

```
#first install necessary packages and load the library
library("dplyr")
library("dslabs")
library("ggplot2")
library("ggrepel")
library("scales")
#make the plot
ggplot(ikea, aes(x = category, y = price)) +
  geom_jitter(size = 0.1) +
  scale_y_log10() +
  labs(
    x = "Catagory",
    y = "Price[log scale]",
    title = "Ikea Furniture"
  ) +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1)
  )
```



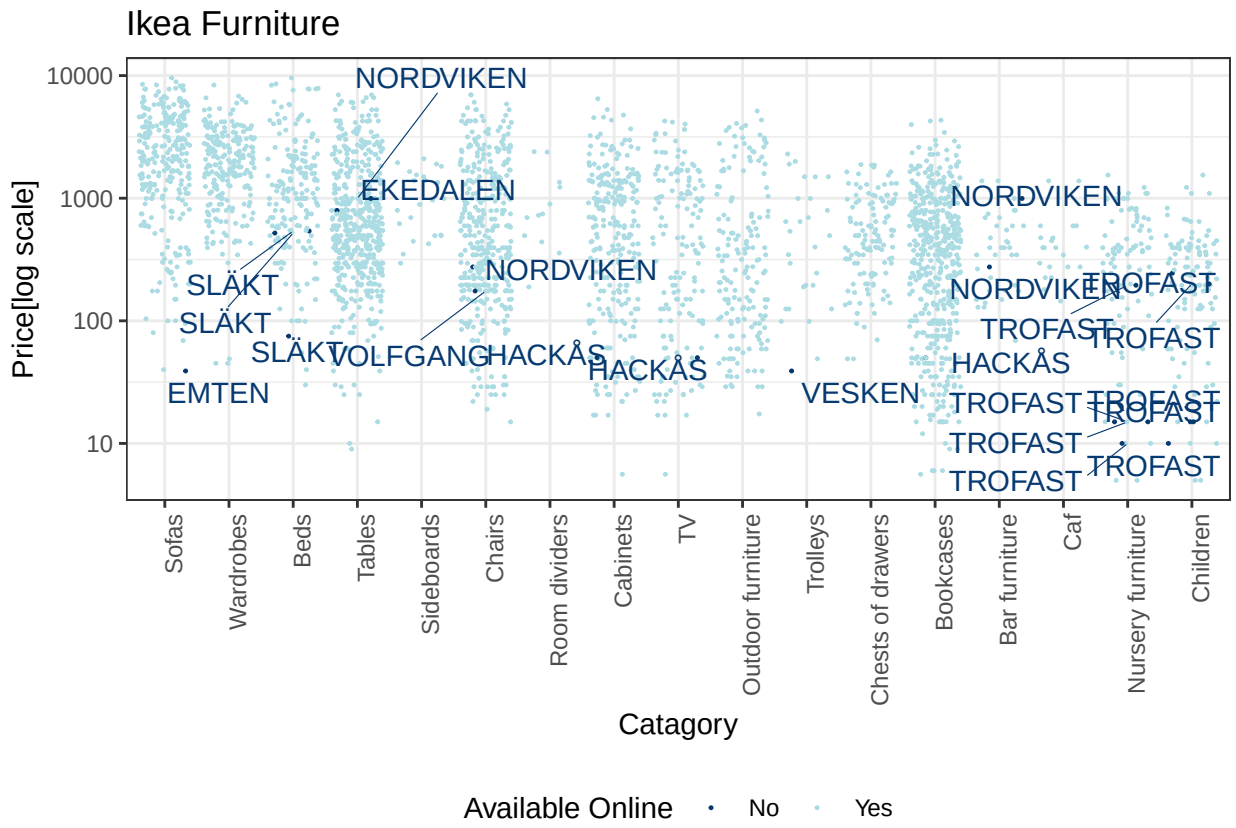
b. Modify the plot in (a) so that categories are sorted from those with highest to lowest average prices.

```
ggplot(ikea, aes(x = reorder(category, -price), y = price)) + #sort categories with reorder() function
  geom_jitter(size = 0.1) +
  scale_y_log10() +
  labs(
    x = "Catagory",
    y = "Price[log scale]",
    title = "Ikea Furniture"
  ) +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1)
  )
```



c. Color points according to whether they can be purchased online. If they cannot be purchased online, add a text label giving the name of that item of furniture.

```
ggplot(ikea, aes(x = reorder(category, -price), y = price)) +
  geom_jitter(size = 0.2, aes(col = sellable_online)) +
  geom_text_repel(
    aes(label = ifelse(sellable_online == FALSE, as.character(name), '')),
    segment.size = 0.2,
    color = "#063970",
    max.overlaps = Inf
    ) +
  scale_y_log10() +
  labs(
    x = "Catagory",
    y = "Price[log scale]",
    color = "Available Online",
    title = "Ikea Furniture"
  ) +
  scale_color_manual(
    values = c("#063970", "#abdbe3"),
    labels = c("No", "Yes")) +
  theme_bw() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 90, hjust = 1)
  )
```



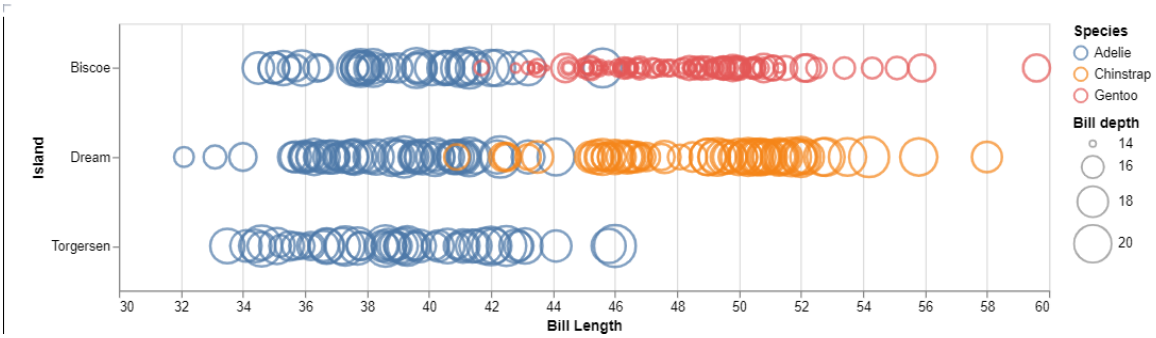
(2) Penguins

The data below measures properties of various antarctic penguins.

```
penguins <- read_csv("https://uwmadison.box.com/shared/static/ijh7iipc9ect1jf0z8qa2n3j7dgem1gh.csv")
```

Using either vega-lite or ggplot2, create a single plot that makes it easy to answer both of these questions,

```
import {vl} from '@vega/vega-lite-api'
import { aq, op } from '@uwdata/arquero'
penguin = aq.fromCSV(await FileAttachment("penguins.csv").text())//use the uploaded file
vl.markPoint({filled: false})
  .data(penguin)
  .encode(
    vl.x().fieldQ("bill_length_mm").scale({domain: [30, 60]}).title("Bill Length"),
    vl.y().fieldN("island").title("Island"),
    vl.color().fieldN("species").title("Species"),
    vl.size().fieldQ("bill_depth_mm").scale({domain: [14, 20], range: [25, 800]}).legend({tickCount: 3})
  )
  .height(200)
  .width(700)
  .render()
```



i) How is bill length related to bill depth within and across species?

From the picture above, overall, the bill length is negative related to bill depth. Within the species, the bill length is positively related to the bill depth.

ii) On which islands are which species found?

Chinstrap is found mostly in Dream. Gentoo is found mostly in Biscoe. Adelie is found in all of these three island.

(Notice that the answer to part (i) is an example of Simpson's paradox!)

(3) 2012 London Olympics

This exercise is similar to the Ikea furniture one, except that it will be interactive. The data at this link describes all participants in the London 2012 Olympics. From an observable notebook, the following code can be used to derive a new variable with a jittered Age variable, which will be useful in part (a).

```
import { vl } from "@vega/vega-lite-api"
import { aq, op } from "@uwdata/arquero"
data_raw = aq.fromCSV(await FileAttachment("All London 2012 athletes - ALL ATHLETES.csv").text())
data = data_raw.derive({Age_: d => d.Age + 0.25 * Math.random() })
```

a. Create a layered display that shows (i) the ages of athletes across sports and (ii) the average age within each sport. Use different marks for participants and for averages. To avoid overplotting, use the jittered `Age_` variable defined in the code block above.

```
age = vl.markSquare()
.data(data)
.encode(
  vl.x().fieldQ('Age_').title("Age"),
  vl.y().fieldN('Sport').title("Sport")
)
mean = vl.markPoint({color: 'red'})
.encode(
  vl.x().average('Age_'),
  vl.y().fieldN('Sport')
)
vl.layer(age,mean)
  .data(data)
  .render()
```

b. Sort the sports from lowest to highest average age. Add a tooltip so that hovering over an athlete shows their name.

```
age = vl.markSquare()
.data(data)
.encode(
```

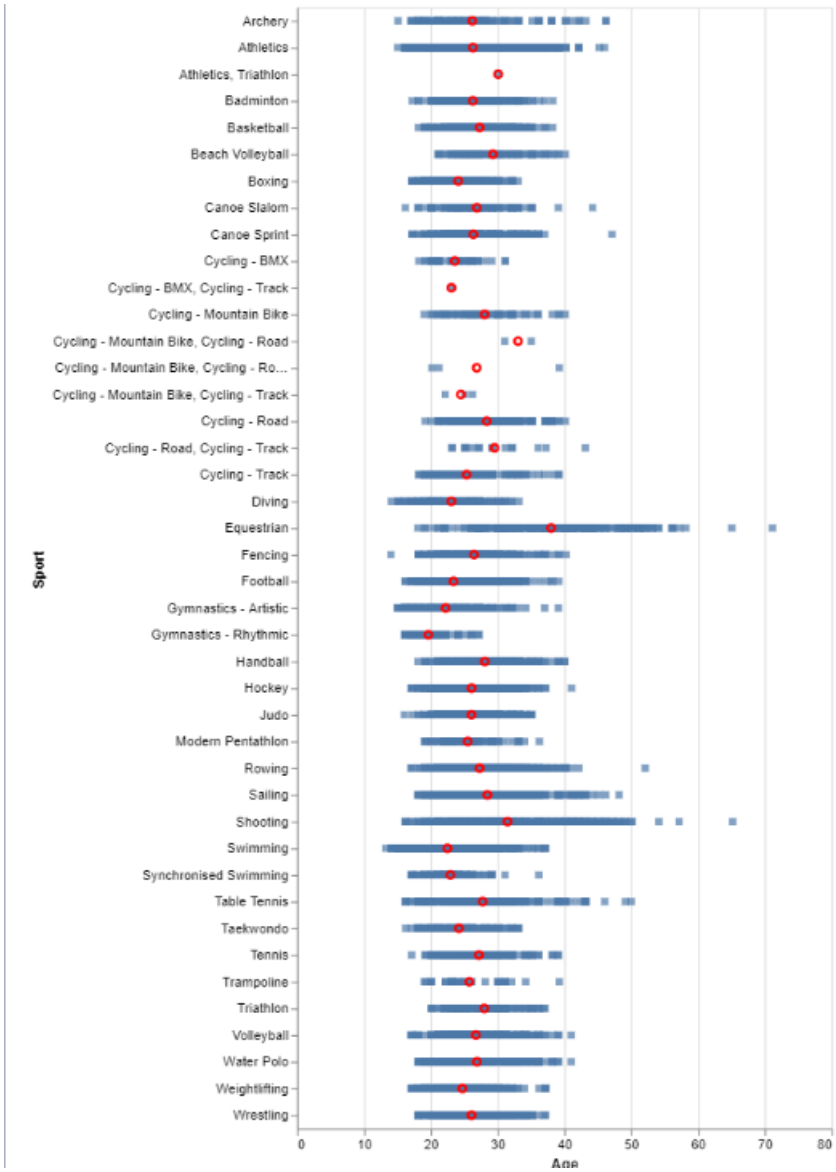


Figure 1: unchanged image

```
  vl.x().fieldQ('Age_').title("Age"),
  vl.y().fieldN('Sport').title("Sport").sort({op:'mean', field:'Age'})
)
mean = vl.markPoint({color: 'red'})
.encode(
  vl.x().average('Age_'),
  vl.y().fieldN('Sport').sort({op:'mean', field:'Age'})
)
vl.layer(age,mean)
  .data(data)
  .render()
```

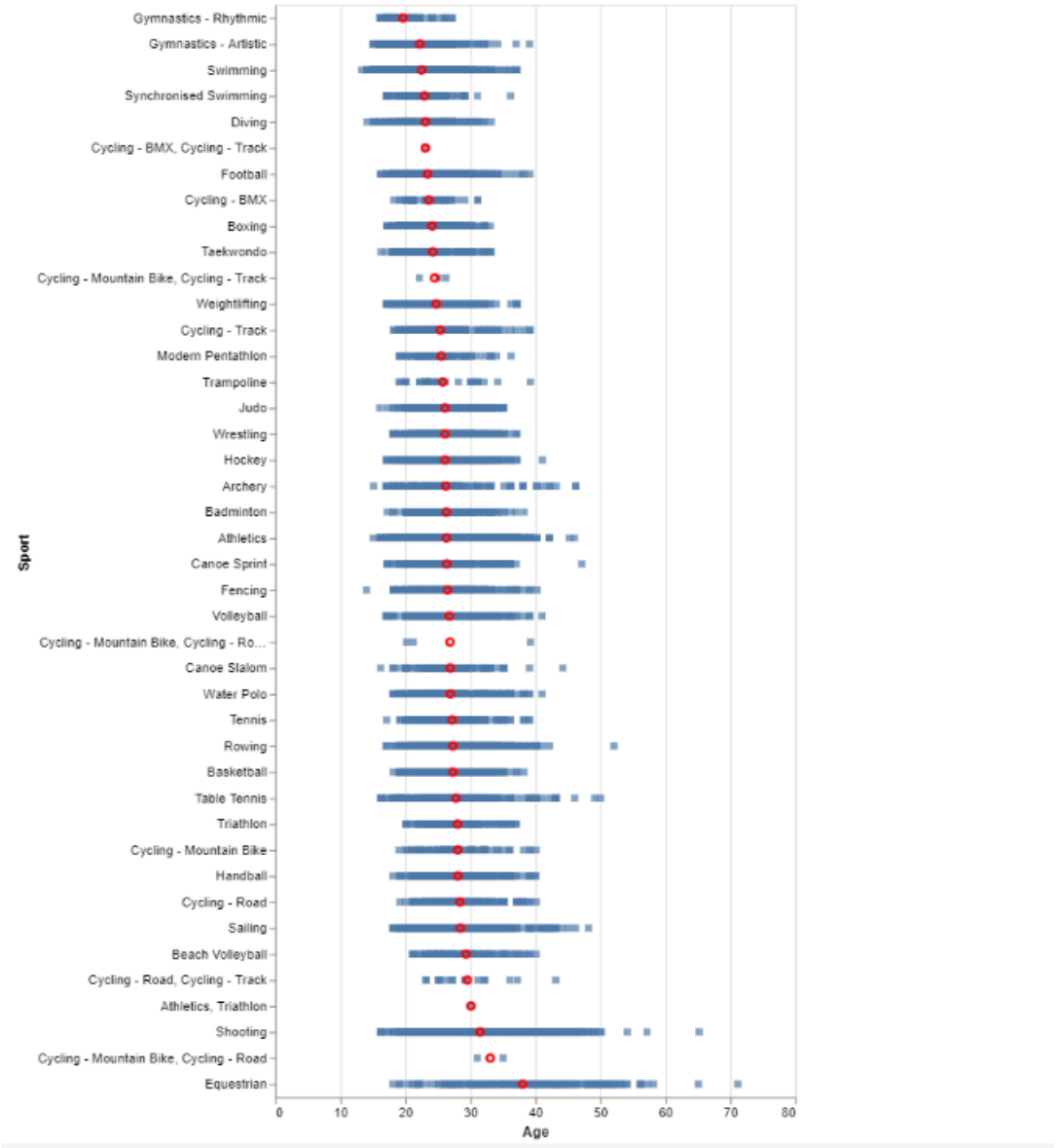


Figure 2: unchanged image

(4) Traffic

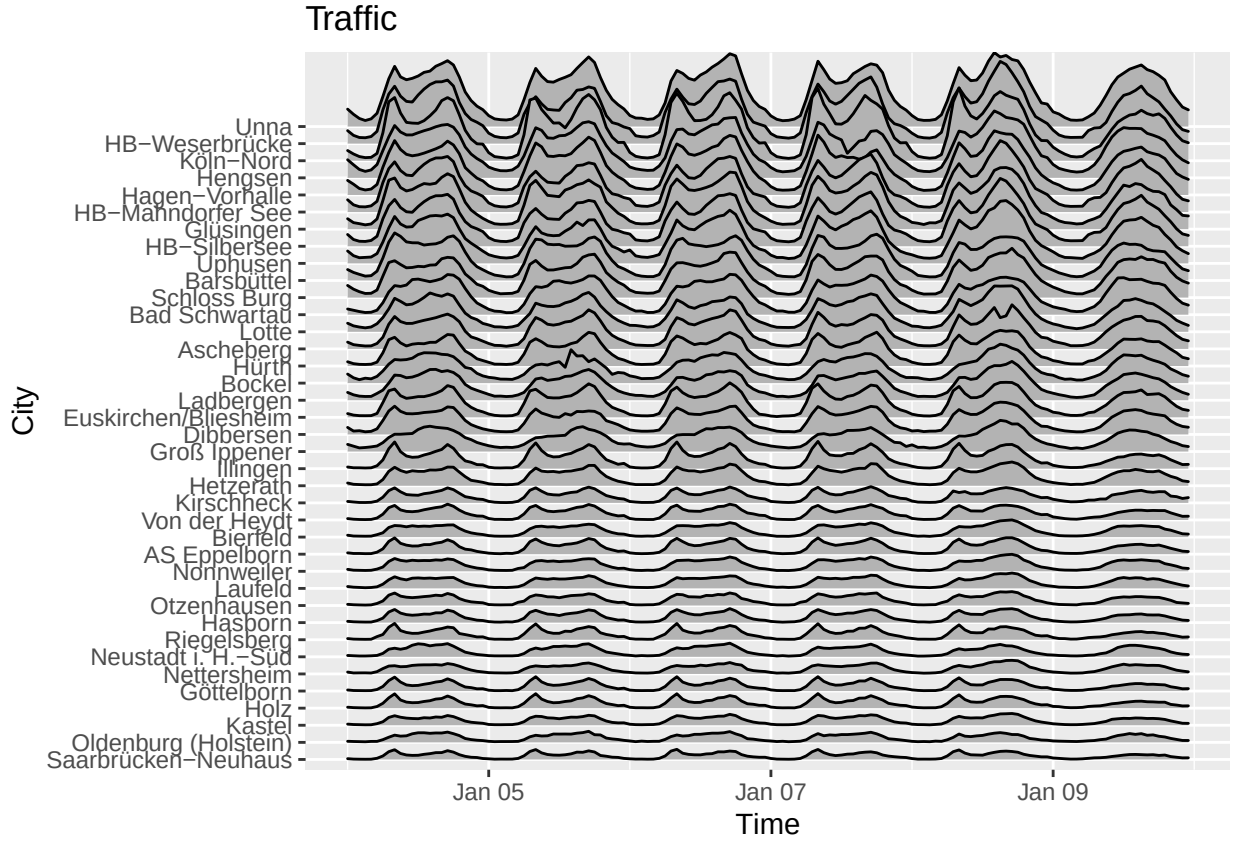
In lecture, we looked at the `geom_density_ridges` function. In this exercise, we will instead use `geom_ridgeline`, which is useful whenever the heights of the ridges have been computed in advance. We will use the traffic data read in below.

```
traffic <- read_csv("https://uwmadison.box.com/shared/static/x0mp3rhhic78vufsxtgrwencchmghbdf.csv")
```

Each row is a timepoint of traffic within a city in Germany. Using `geom_ridges`, make a plot of traffic over time, within each of the cities. An example result is shown below.

```
#load the necessary libraries
library("ggridges")

ggplot(traffic) +
  geom_ridgeline(aes(x=date, reorder(name, value, mean), height=value)) +
  labs(
    x = "Time",
    y = "City",
    title = "Traffic"
  )
```



(5) Language Learning

This problem will look at a simplified version of the data from the study *A critical period for second language acquisition: Evidence from 2/3 million English speakers*, which measured the effect of the the age of initial language learning on performance in grammar quizzes. We have downloaded the raw data from the supplementary material and reduced it down to the average and standard deviations of test scores within (initial learning age) $\times$ (current age-group) combinations. We have kept a column `n` showing how many

participants were used to compute the associated statistics. The resulting data are available here.

a) Using the `.derive()` command in arquero, create two new fields, `low` and `high`, giving confidence intervals for the means in each row. That is, derive new variables according to $\hat{x} \pm 2 * \frac{1}{\sqrt{n}}\hat{\sigma}$.

```
import { vl } from "@vega/vega-lite-api"
import { aq,op } from "@uwdata/arquero"
data = aq.fromCSV(await FileAttachment("language_summary-5.csv").text())
language = data.derive({low: d => d.avg_correct - 2*d.sd_correct/op.sqrt(d.n),
                        high: d => d.avg_correct + 2*d.sd_correct/op.sqrt(d.n)})
```

b) Create a `markArea`-based ribbon plot showing confidence intervals for average test scores as a function of starting age. Include a line for the average score within that combination.

```
mean_score = vl.markArea({"opacity": 0.2})
  .data(language)
  .encode(
    vl.x().fieldQ('Eng_start').title("Age when started learning English"),
    vl.y().fieldQ('low').title('Test Score').scale({"zero": false}, {'scheme': 'set2'}),
    vl.y2().fieldQ('high'),
    vl.color().fieldN('age_group')
)

language_mid = language.derive({mid: d => 0.5 * (d.low + d.high)});

mid_score = vl.markLine({"opacity": 1})
.data(language_mid)
.encode(
  vl.x().fieldQ('Eng_start'),
  vl.y().fieldQ('mid'),
  vl.color().fieldN('age_group').scale({'scheme':'set2'}).title("Current Age").legend({orient:'bottom'})
)

vl.layer(mean_score, mid_score)
.data(language_mid)
.render();
```

Interpretation: The test score is negatively related to age when started learning English. The visualization shows that there is a critical second-language studying period, which is about 0-12 years old.

(6) Deconstruction

Take a static screenshot from any of the visualizations in this article, and deconstruct its associated visual encodings.

a) What do you think was the underlying data behind the current view? What where the rows, and what were the columns?

I think the underlying data is those places where homeless relocation programs are concentrated(the starting points of relocations). The rows are cities across the U.S., and the columns may include the homeless rate, the number of homeless relocation pragrams, the number of homeless relocations each year, and the destinations of their relocations.

b) What were the data types of each of the columns?

The homeless rate is double, the number of homeless relocation programs is int, the number of homeless relocations each year is int, and the destination is string.

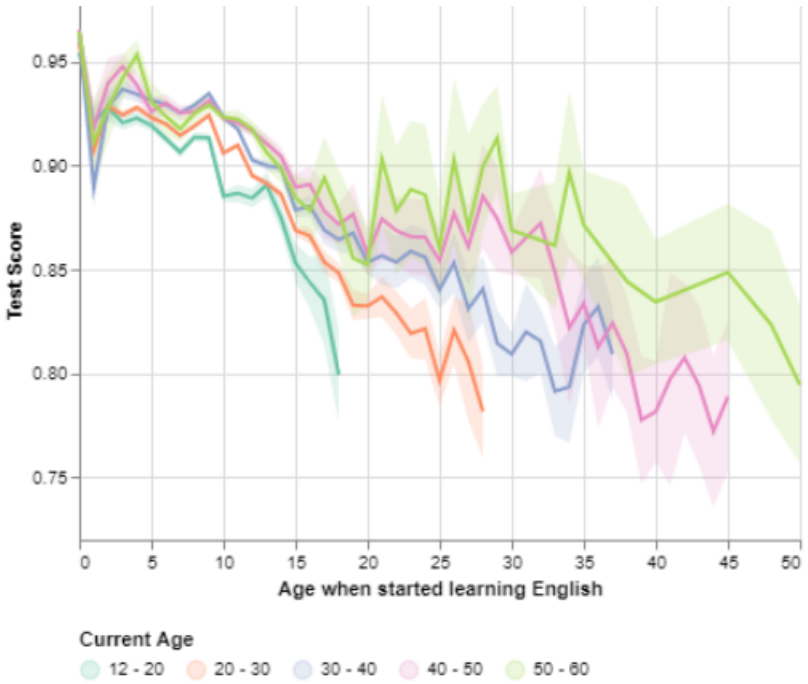


Figure 3: unchanged image

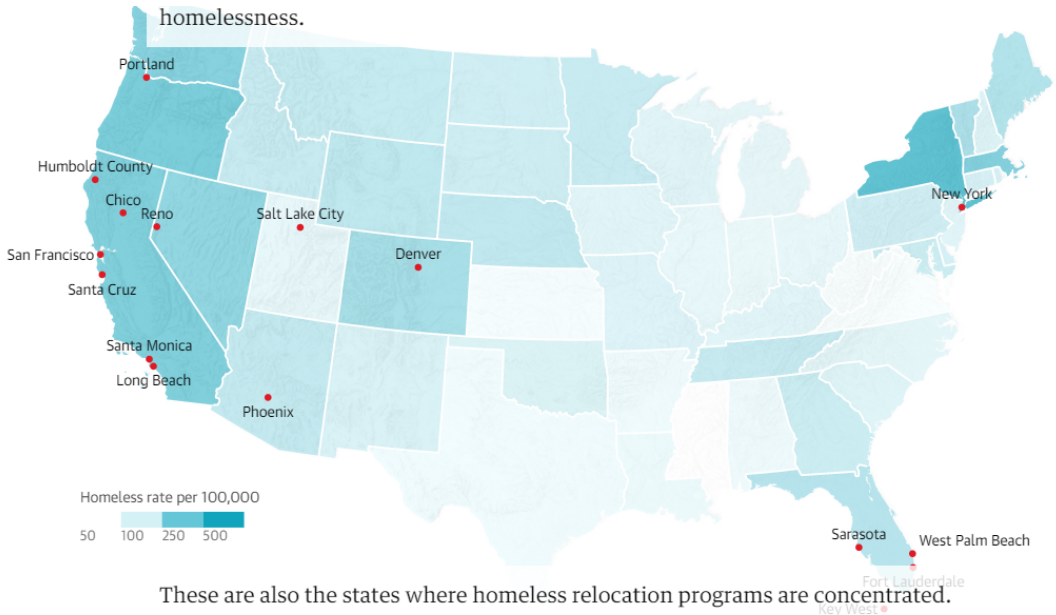


Figure 4: unchanged image

c) What encodings were used? How are properties of marks on the page derived from the underlying abstract data?

Encodings may include point marks and area marks, which are underlied by color and labels. The red point marks seem very obvious under the blue area marks, which is derived from those places with the most concentrated homeless relocation programs. Deep blue area marks are more obvious than light blue ones, which is derived from the homeless rates.

c) Is multi-view composition being used? If so, how?

Yes. There are multiple layers in this graph, including the map(area marks) and the cities(point marks). Maybe include the texts(article contents) over the graph.