479 HW 1

Caitlin Bolz

2/3/2021

```
library(knitr)
library(readr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggrepel)
library(ggridges)
```

Problems

(1) Ikea Furniture

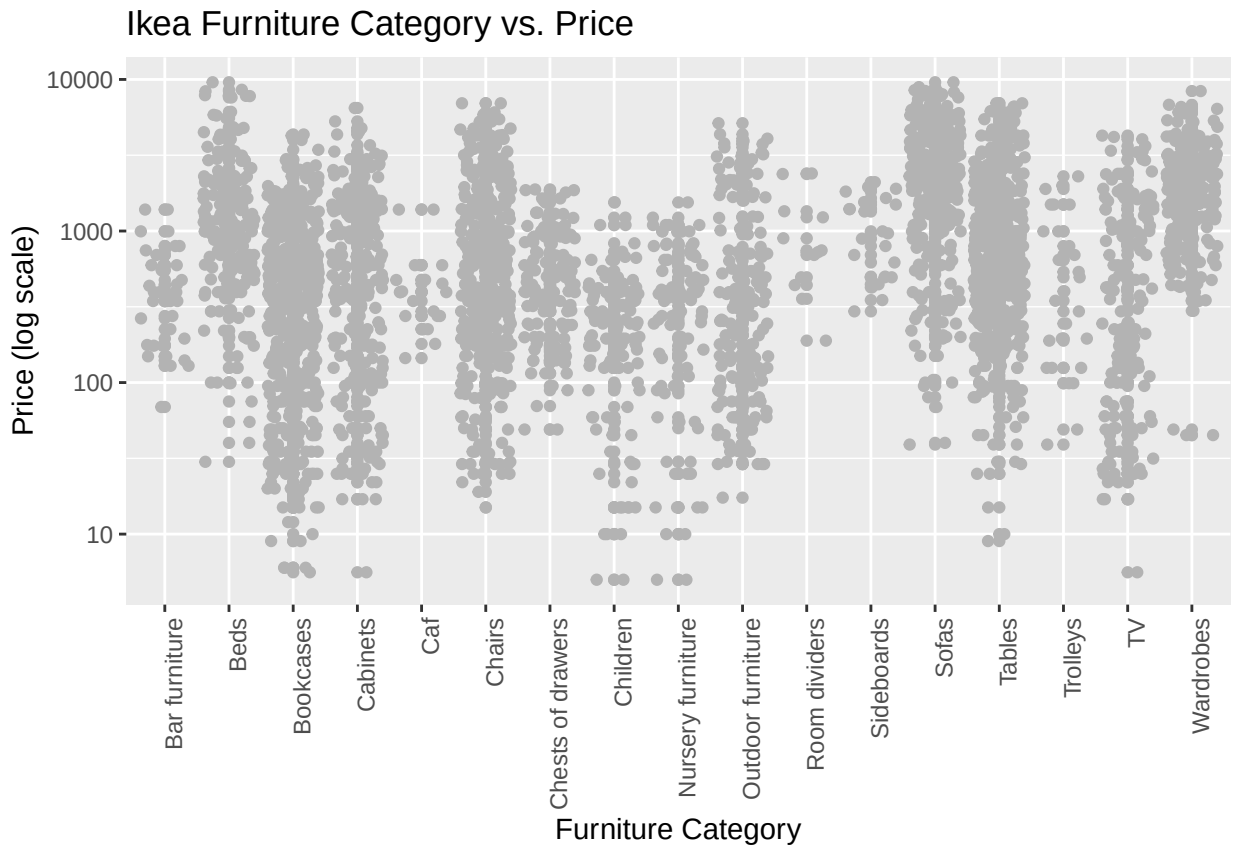
```
ikea <- read_csv("https://uwmadison.box.com/shared/static/iat31h1wjg7abhd2889cput7k264bdzd.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   item_id = col_double(),
##   name = col_character(),
##   category = col_character(),
##   price = col_double(),
##   old_price = col_character(),
##   sellable_online = col_logical(),
##   link = col_character(),
##   other_colors = col_character(),
```

```
##   short_description = col_character(),
##   designer = col_character(),
##   depth = col_double(),
##   height = col_double(),
##   width = col_double()
## )
```

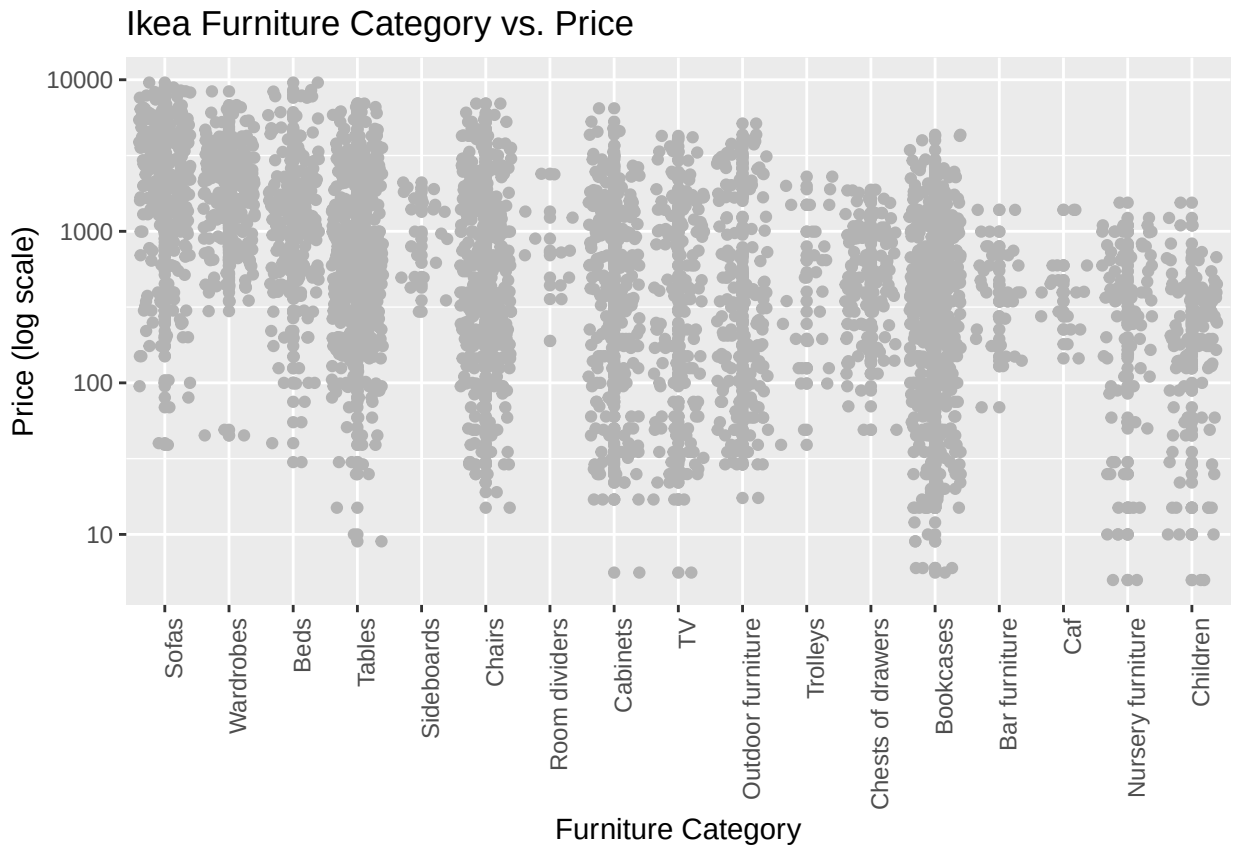
Part A

```
ggplot(ikea) +
  geom_point(aes(x = category, y = price), col = 'gray70') +
  scale_y_log10() +
  geom_jitter(aes(x = category, y = price), col = 'gray70') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    x = "Furniture Category",
    y = "Price (log scale)",
    title = "Ikea Furniture Category vs. Price"
  )
```



Part B

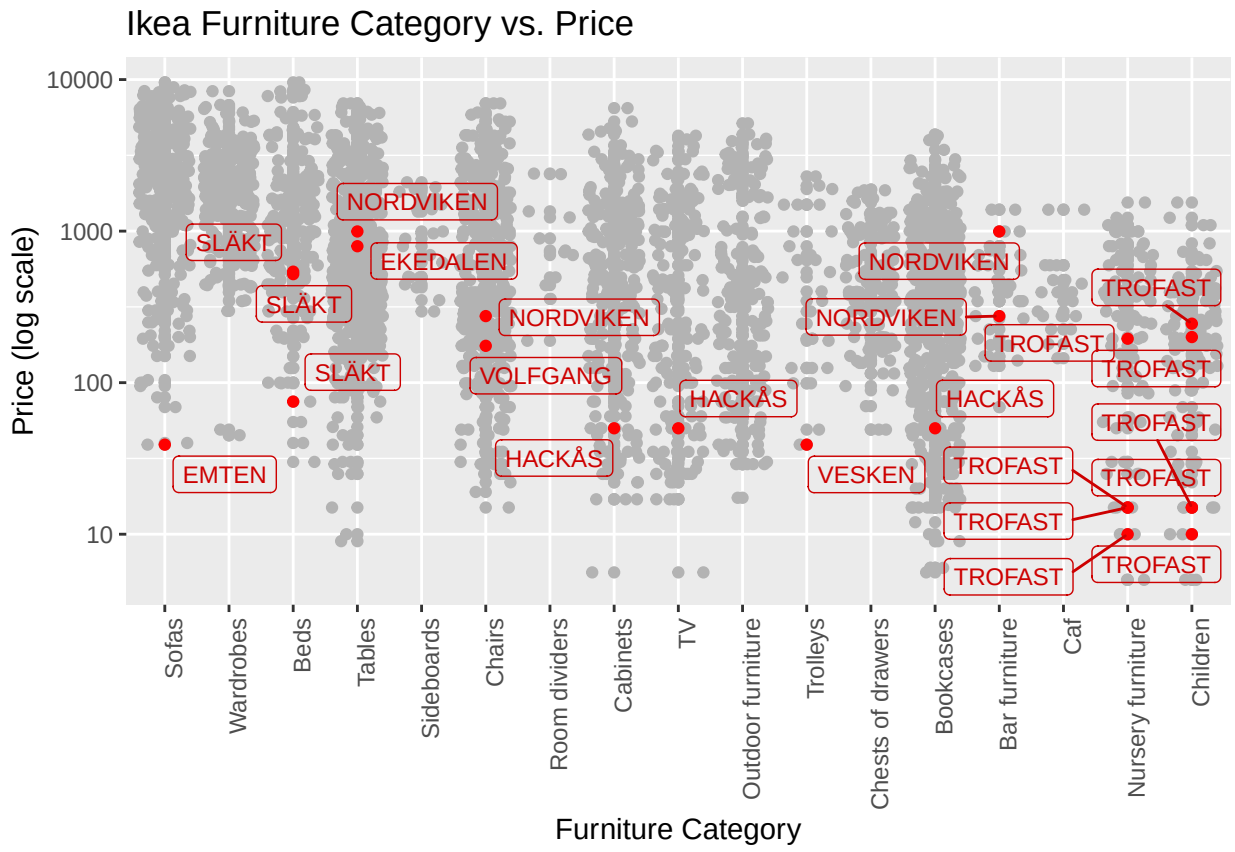
```r
ikea$category = with(ikea, reorder(category, -price, mean))

ggplot(ikea) +
  geom_point(aes(x = category, y = price), col = 'gray70') +
  scale_y_log10() +
  geom_jitter(aes(x = category, y = price), col = 'gray70') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(
    x = "Furniture Category",
    y = "Price (log scale)",
    title = "Ikea Furniture Category vs. Price"
  )
```



Part C

```r
highlight_df <- ikea %>%
            filter(sellable_online == F)
ggplot(ikea) +
  geom_point(aes(x = category, y = price), col = 'gray70') +
  scale_y_log10() +
  geom_jitter(aes(x = category, y = price), col = 'gray70') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_point(data = highlight_df, aes(x = category, y = price), color = 'red') +
```

```
  geom_label_repel(data = highlight_df, aes(x = category, y = price),
                   label = highlight_df$name, col ='red3', size = 3, force = 2,
                   max.overlaps = 20, fill = NA)+
  labs(
    x = "Furniture Category",
    y = "Price (log scale)",
    title = "Ikea Furniture Category vs. Price"
  )
```



(2) Penguins

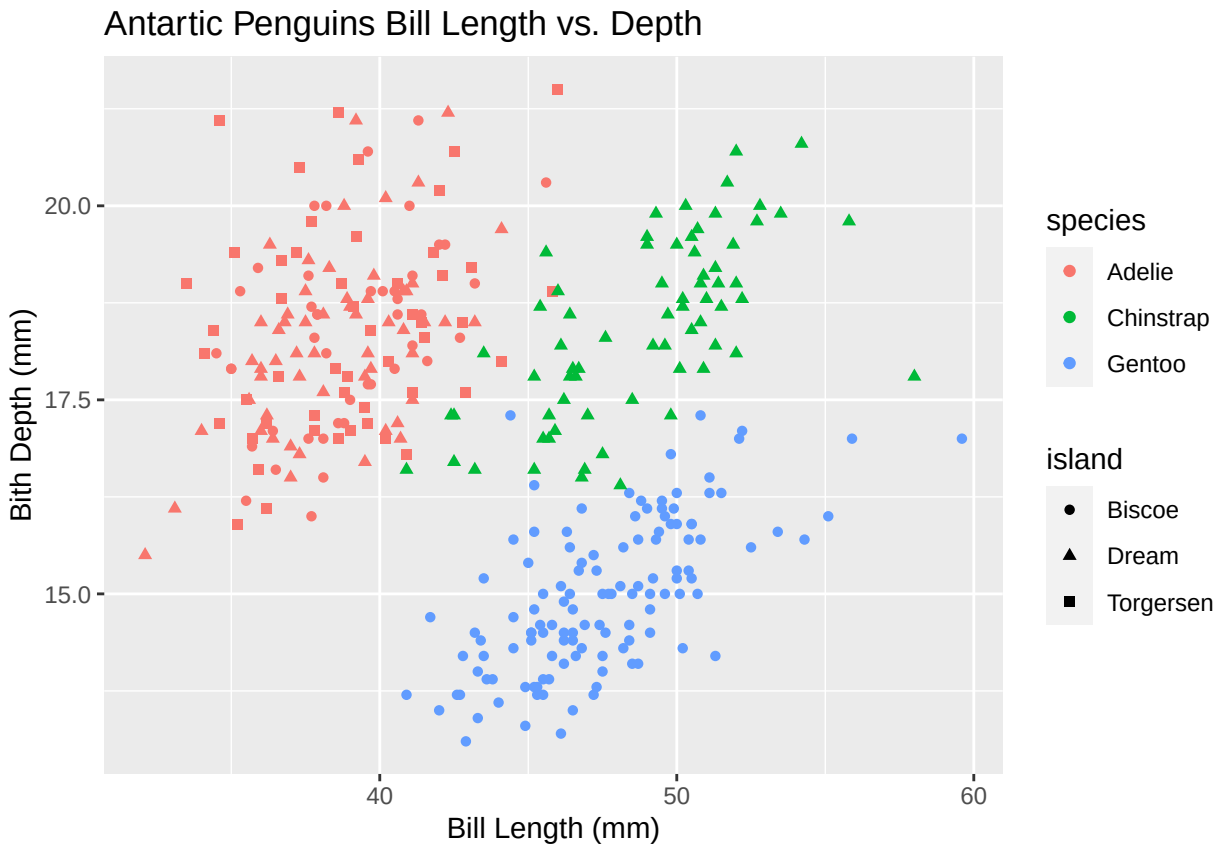
```
penguins <- read_csv("https://uwmadison.box.com/shared/static/ijh7iipc9ect1jf0z8qa2n3j7dgem1gh.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   species = col_character(),
##   island = col_character(),
##   bill_length_mm = col_double(),
##   bill_depth_mm = col_double(),
##   flipper_length_mm = col_double(),
##   body_mass_g = col_double(),
##   sex = col_character(),
```

```
##   year = col_double()
## )
```

```
ggplot(penguins) +
  geom_point(aes(bill_length_mm, bill_depth_mm, col = species,
                 shape = island)) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    x = "Bill Length (mm)",
    y = "Bith Depth (mm)",
    title = "Antartic Penguins Bill Length vs. Depth"
  )
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



(3) 2012 London Olympics

```
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() })
```

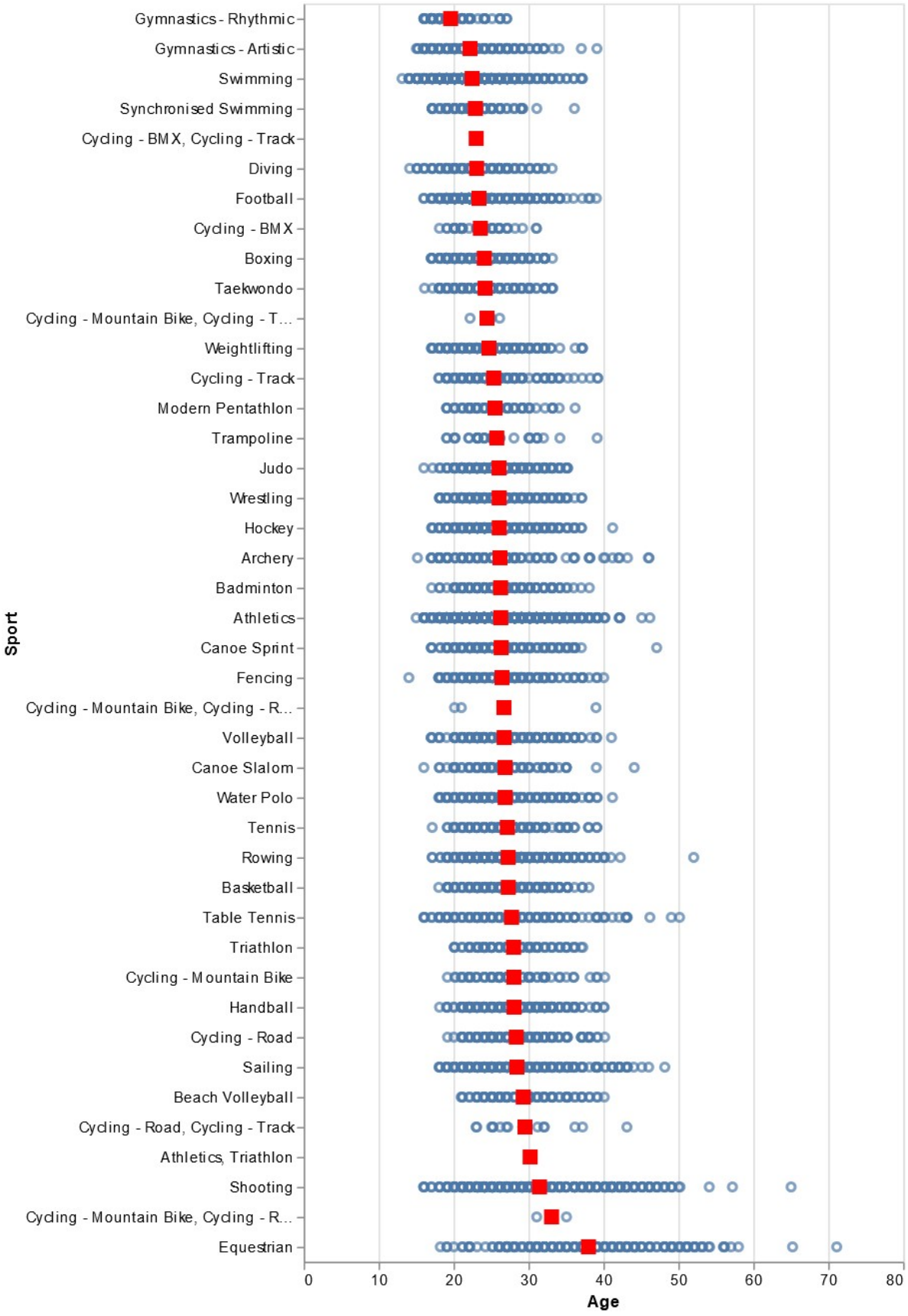


Figure 1: Q3: Olympics Age Visualization

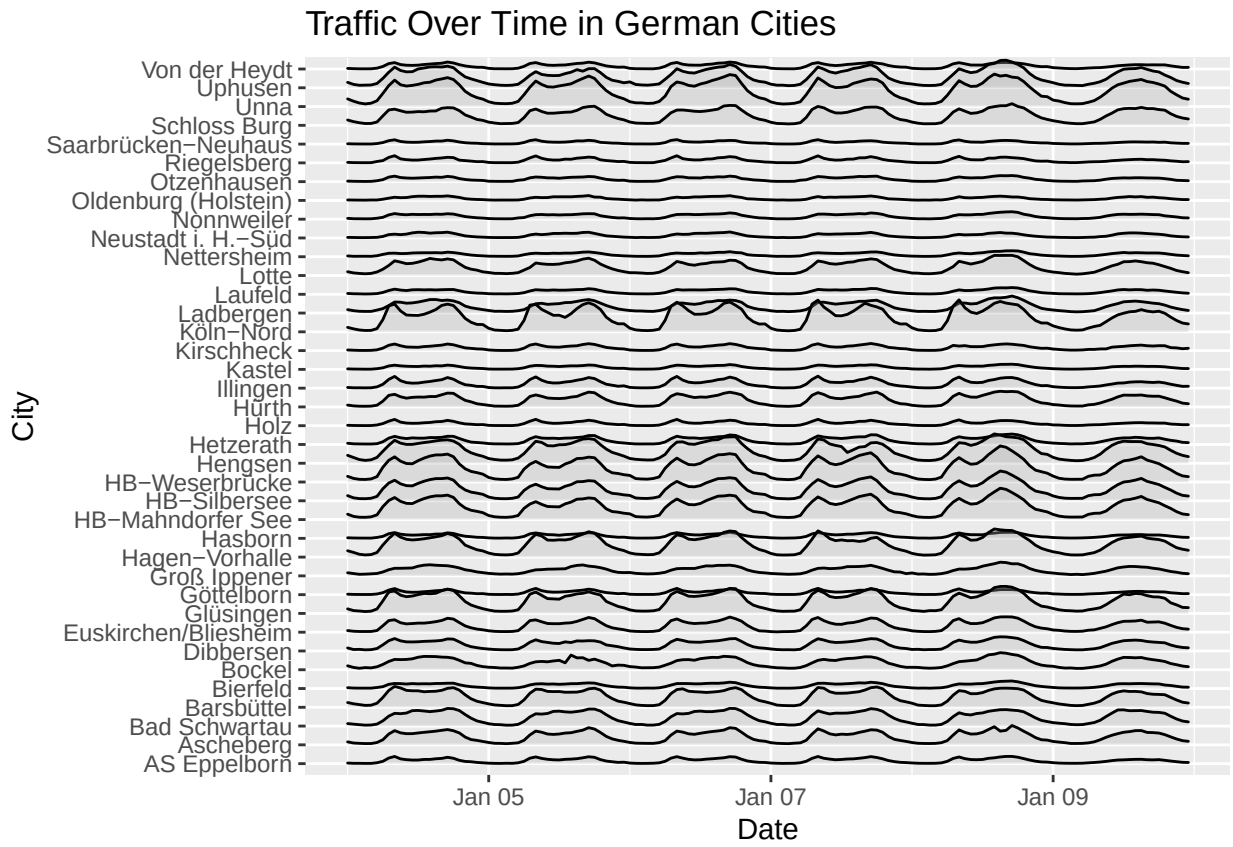
```
viewof layered = vl
  .data(data)
  .layer(
     vl.markPoint()
      .encode(
        vl.x().fieldQ("Age_").title("Age"),
        vl.y().fieldN("Sport").title("Sport").sort({op: "mean", field: "Age_"}),
        vl.tooltip().fieldN("Name"),
       ),
    vl.markSquare({color: "red", size: 100})
      .encode(
        vl.y().fieldN("Sport").title("Sport").sort({op: "mean", field: "Age_"}),
        vl.x().average("Age_"),
        )
    )
.render()
```

(4) Traffic

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

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   name = col_character(),
##   date = col_datetime(format = ""),
##   value = col_double()
## )
```

```
ggplot(traffic, aes(x = date, height = value, y = name)) +
  geom_ridgeline(scale= .4, alpha = .3)+
  labs(
    x = "Date",
    y = "City",
    title = "Traffic Over Time in German Cities"
  )
```



(5) Language Learning

Part A

```
data_raw5 = aq.fromCSV(await FileAttachment("language_summary-5.csv").text())
data5 = {
  return data_raw5.derive({Low: d => d.avg_correct - 2 * d.sd_correct / op.sqrt(d.n), High: d => d.avg_
  }
```

Part B

```
{
  // ribbon layer
  const dataMinMax = vl.markArea({opacity: 0.3})
  .data(data5)
  .encode(
    vl.x().fieldQ('Eng_start').title("Age When Started Learning English"),
    vl.y().fieldQ('Low'),
    vl.y2().fieldQ('High'),
    vl.color().fieldN('age_group'),
  );
```

```
  // line layer
  const dataMid = vl.markLine()
    .data(data5)
    .transform(
      vl.calculate('(datum.Low + datum.High) / 2').as('temp_mid')
      )
    .encode(
      vl.x().fieldQ('Eng_start').title("Age When Started Learning English"),
      vl.y().fieldQ('temp_mid').scale({domain: [.6, 1]}).axis({tickCount:4}).title("Test Score"),
      vl.color().fieldN('age_group')
  );

  // overlay
  return vl.layer(dataMinMax, dataMid)
    .data(data5)
    .render();
}
```

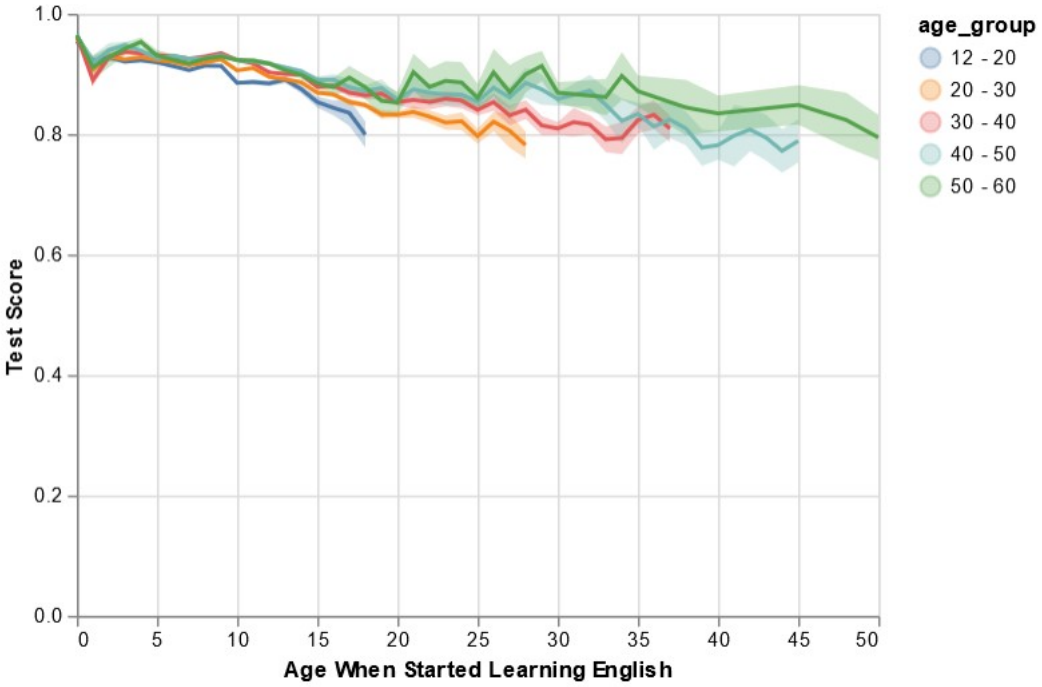


Figure 2: Q5: Ribbion Plot

(6) Deconstruction

Part A

Below are what I think the example columns are. There would be a row associated with each name/ID number 1. Names or ID numbers associated with each ticket recipient 2. Origin city of homeless traveler 3. Final Destination City 4. Average income of origin city 5. Average income of final city 6. Difference in income in two cities

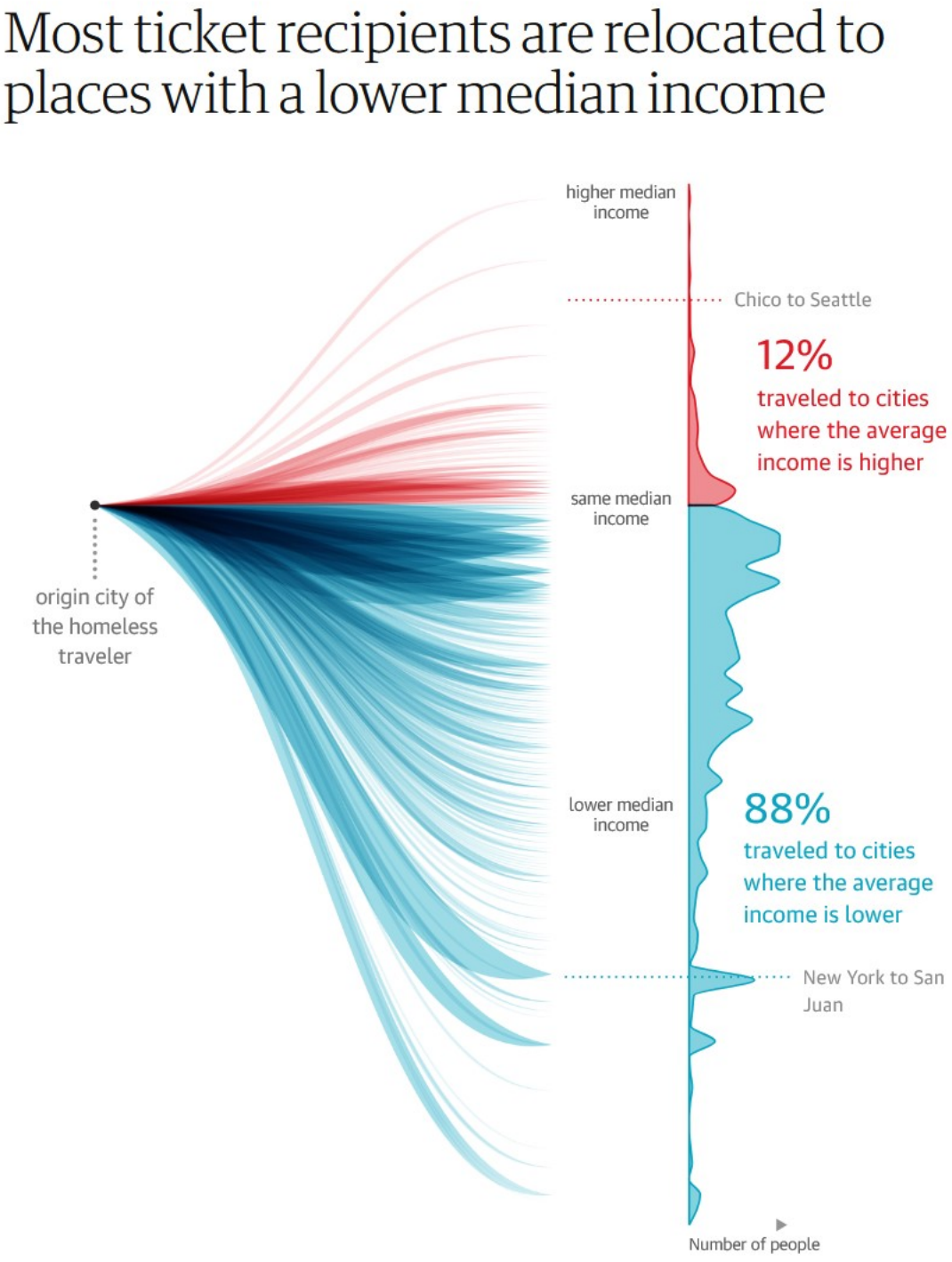


Figure 3: Q6: Article Visualization

Part B

Listed below are the data types for the columns described in Part A 1. Character or integer 2. Character 3. Character 4. Numeric 5. Numeric 6. Numeric

Part C

All of the origin cities start at the same point, giving the illusion that everyone starts in the same city. But really the origin city income is being compared to increase or decrease in the final destination city income. Each individual path represents a mark, since they represent a row. The difference in median income between the cites is encoded in in the lines we see from the origin point to the left. The count for each difference in median income is displayed through a density plot on the far right,.

Part D

Yes. The visualization shows if people went to cities that had a lower or higher median income. In addition, it also shows the density of how many people made that specific increase or decrease in median income.