STAT 479 HW4

Caitlin Bolz

```
library(knitr)
library(readr)
library(ggplot2)
library(janeaustenr)
library(tidytext)
library(dplyr)
library(MASS)
library(tibble)
library(purrr)
library(tidyr)
library(superheat)
library(tidymodels)
library(cluster)
library(topicmodels)
library(forcats)
library(ggrepel)
library(stringr)
library(embed)
```

(1) Polio incidence

```
polio = read_csv("https://uwmadison.box.com/shared/static/nm7yku4y9q7ylvz5kbxya3ouj2njd0x6.csv")
```

Part A

```
polio_pivot = pivot_wider(
  polio,
  names_from = period_start_date,
  values_from = cases)

polio_pivot[is.na(polio_pivot)] = 0

polio_pivot = polio_pivot %>%
  column_to_rownames(var = "state")
```

Part B

```
polio_pivot_log = log(1 + polio_pivot)
cols = c('#f6eff7','#bdc9e1','#67a9cf','#1c9099','#016c59')
superheat(
  polio_pivot_log,
  pretty.order.rows = T,
  left.label.text.size = 7,
  heat.pal = cols,
  heat.lim = c(0,5)
)
```

Part C

Based on the figure 2 below, it appears that the peak amount of new polio cases occurred during the middle of this time period from 1916 to 1922.

```
cols = c('#f6eff7','#bdc9e1','#67a9cf','#1c9099','#016c59')
superheat(
  polio_pivot_log,
  pretty.order.rows = T,
  left.label.text.size = 5,
  heat.pal = cols,
  heat.lim = c(0,5),
  yt = colSums(polio_pivot),
  yt.plot.type = "bar",
  yt.bar.col = "black",
  yt.axis.name = ""
)
```

Part D

- Labeling the x-axis would be useful, because the viewer could see the points in time that this heatmap corresponds to. Also a description of what the 2000 and 4000 for the bar graph means. Additionally, a description of what the 1 to 5 scale at the bottom represents.
- A disadvantage of visualizing a time series as a heatmap is that it is difficult to tell the different in magnitude for each state at a given point in time. If there is a dark teal square for two states, we are only able to say that they both had high values of new cases. We aren't able to easily say which one was higher than the other.
- An advantage of visualizing a time series as a heatmap is that it is easier to compare relative trends over all 52 values on a heatmap, compared to trying to compare 52 line plots to each other.

(2) Silhouette statistics simulation

```
mixture_data = function(start = 0, end = 10, K = 3, n = 100) {
  mu = cbind(seq(start, end, length.out = K), 0)
  map_dfr(1:K, ~ mvrnorm(n, mu[., ], diag(2)) %>% as_tibble())
}
```

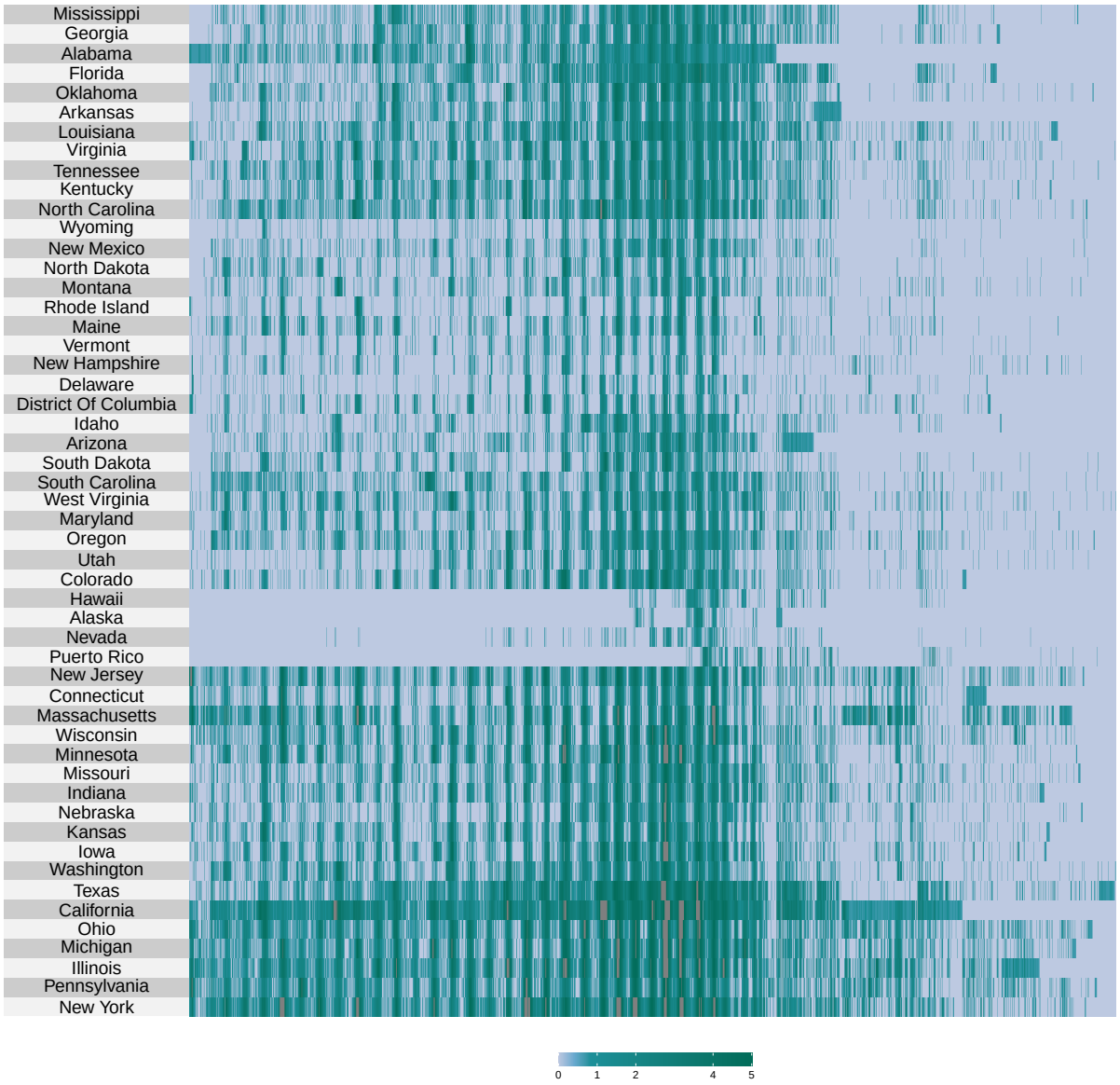

Figure 1: Q1, Part B

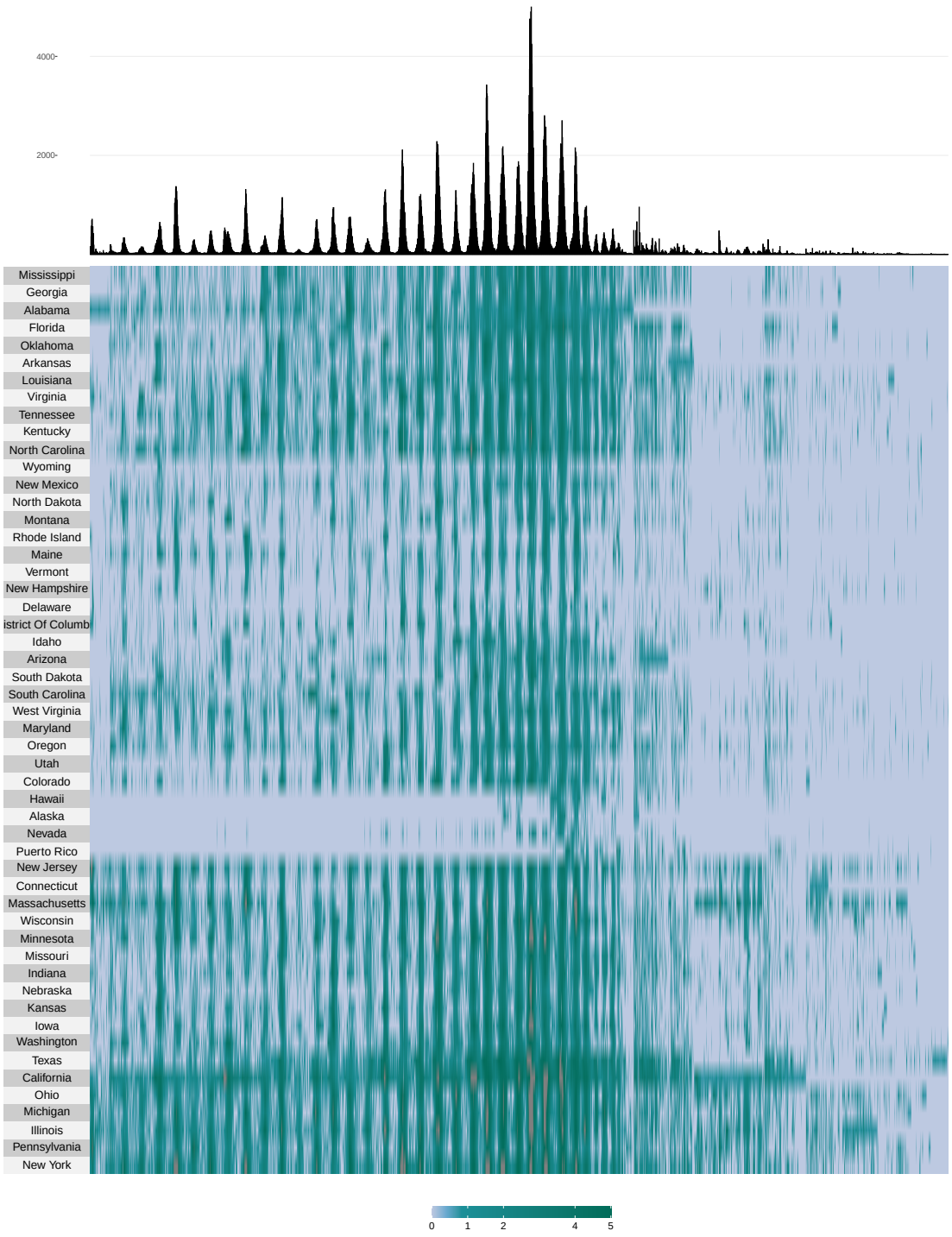

Figure 2: Q1, Part C

Part A

```
dataset10 = mixture_data()
dataset8 = mixture_data(end = 8)
dataset6 = mixture_data(end = 6)
dataset4 = mixture_data(end = 4)
dataset2 = mixture_data(end = 2)
dataset0 = mixture_data(end = 0)
```

Part B

```
gaussian_kmeans = function(dataset, K){
  x = dataset %>%
    scale()

  kmeans(x, center = K) %>%
    augment(dataset) %>%
    mutate(silhouette = silhouette(as.integer(.cluster), dist(x))[, "sil_width"])
}

kmeans10 = gaussian_kmeans(dataset10, K = 3)
kmeans8 = gaussian_kmeans(dataset8, K = 3)
kmeans6 = gaussian_kmeans(dataset6, K = 3)
kmeans4 = gaussian_kmeans(dataset4, K = 3)
kmeans2 = gaussian_kmeans(dataset2, K = 3)
kmeans0 = gaussian_kmeans(dataset0, K = 3)

kmeans10$id = 10
kmeans8$id = 8
kmeans6$id = 6
kmeans4$id = 4
kmeans2$id = 2
kmeans0$id = 0

kmeans_dataset = rbind(kmeans0, kmeans2, kmeans4, kmeans6, kmeans8, kmeans10)
```

Part C

As you decrease the end value, the range of x for the corresponding graph is also decreased. Additionally, with an end value of 10, you have three seperate distinct groups. As you decrease the end value, the data becomes less clustered and there is not as distinctly seperate groups.

```
ggplot(kmeans_dataset, aes(x = V1, y = V2, col = silhouette)) +
  geom_point() +
  facet_wrap(~ id, ncol = 3) +
  xlab("x") +
  ylab("y") +
  theme(legend.position = "bottom")
```

Figure 3: Q2, Part C

(3) Taxi trips

```
trips_long =
  read_csv("https://uwmadison.box.com/shared/static/098cjaetm8vy0mufq21mc8i9nue2rr2b.csv",
           col_types = cols(TRIP_ID = col_character()))
trips_wide =
  read_csv("https://uwmadison.box.com/shared/static/cv0lij4d9gn3s8m2k98t2ue34oz6sbj5.csv",
           col_types = cols(TRIP_ID = col_character()))
```

Part A

```
trips_long %>%
  filter(TRIP_ID == "1389986517620000047") %>%
  ggplot(aes(x=x, y=y)) +
  geom_point() +
  theme_minimal() +
  labs(
    x = "Longitude",
    y = "Latitude"
  )
```

Part B

If we look at the `wide` format dataset we can interpret latitude, longitude, and trip id through clustering. In comparison, the `long` format dataset has trip id, call type, taxi id, timestamp, day type, trajectory element, latitude, longitude, n rows, date, day of week, year, month, and hour. We are able to examine the same data in `long` as we are in `wide`, but just with more depth. We can choose to cluster by something else than trip id.

Part C

```
dist = trips_wide %>%
  column_to_rownames(var = "TRIP_ID") %>%
  dist()

hclust = hclust(dist)
```

Part D

```
clusters = cutree(hclust, k = 8) %>%
  tibble(TRIP_ID = names(.), cluster = as.factor(.))

trips_long %>%
  left_join(clusters, by = "TRIP_ID") %>%
```

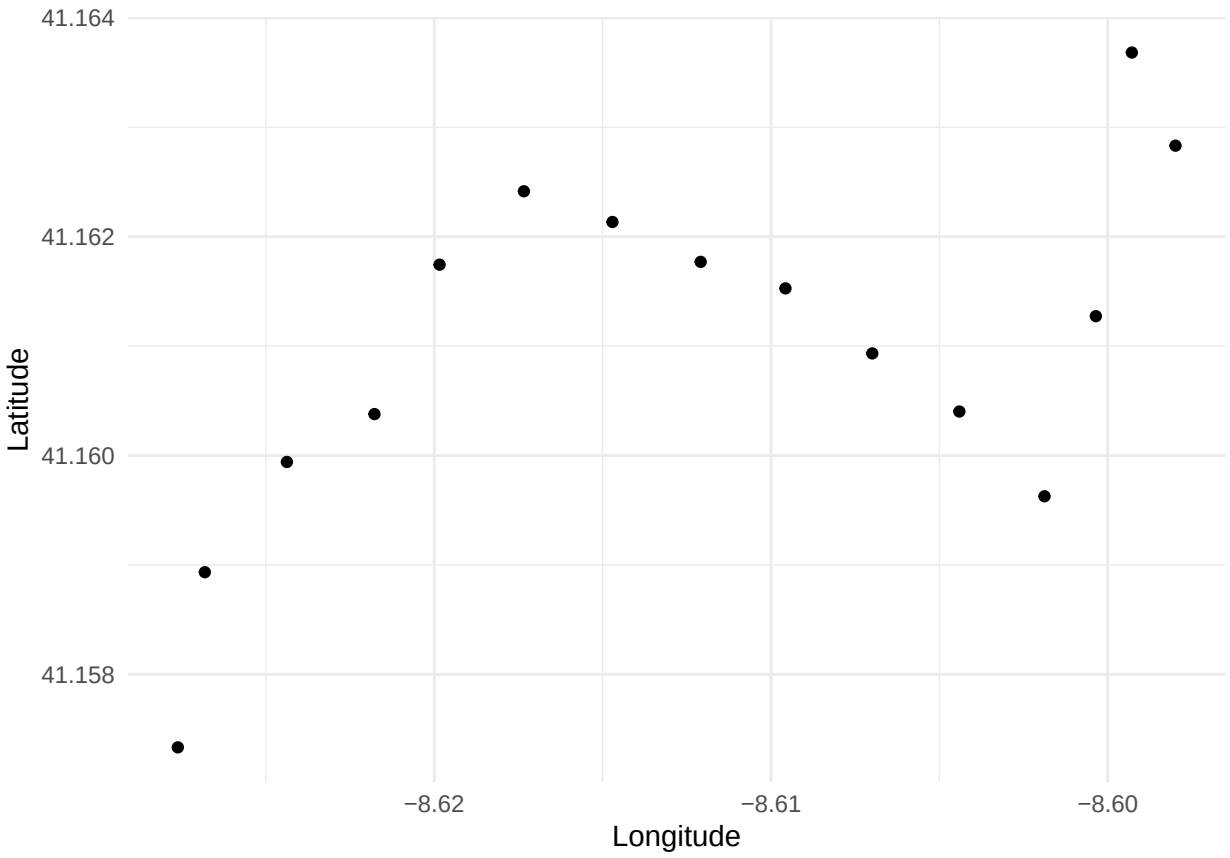

Figure 4: Q3 Part A

```
  ggplot(aes(x = x, y=y, col = cluster))+
  geom_jitter(alpha = 0.1, size = 0.1) +
  theme_bw()+
  theme(
    panel.background = element_rect(fill = "#f7f7f7"),
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
  ) +
  scale_color_brewer(palette = "Set2") +
  coord_fixed()+
  guides(color = guide_legend(override.aes = list(size = 2, alpha = 1))) +
  labs(
    x = "Longitude",
    y = "Latitude",
    color = "Cluster"
  )
```

(4) Food nutrients

```
nutrients =
  read_csv("https://uwmadison.box.com/shared/static/nmgouzobq5367aex45pnbzgkhm7sur63.csv")
```

Part A

```
pca_rep = recipe(~., data = nutrients) %>%
  update_role(name, group, group_lumped, new_role = "id") %>%
  step_normalize(all_predictors()) %>%
  step_pca(all_predictors())

pca_prep = prep(pca_rep)

components = tidy(pca_prep, 2)

scores = juice(pca_prep)

variances = tidy(pca_prep, 2, type = "variance")
```

Part B

I expect foods with lots of water to have low values of PC1 and foods with high calories to have high values of PC1. I expect foods with high values of carbohydrates to have to have low values of PC2.

```
components_new = components %>%
  filter(component %in% str_c("PC", 1:6)) %>%
  mutate(terms = reorder_within(terms, abs(value), component))

ggplot(components_new, aes(value, terms)) +
```

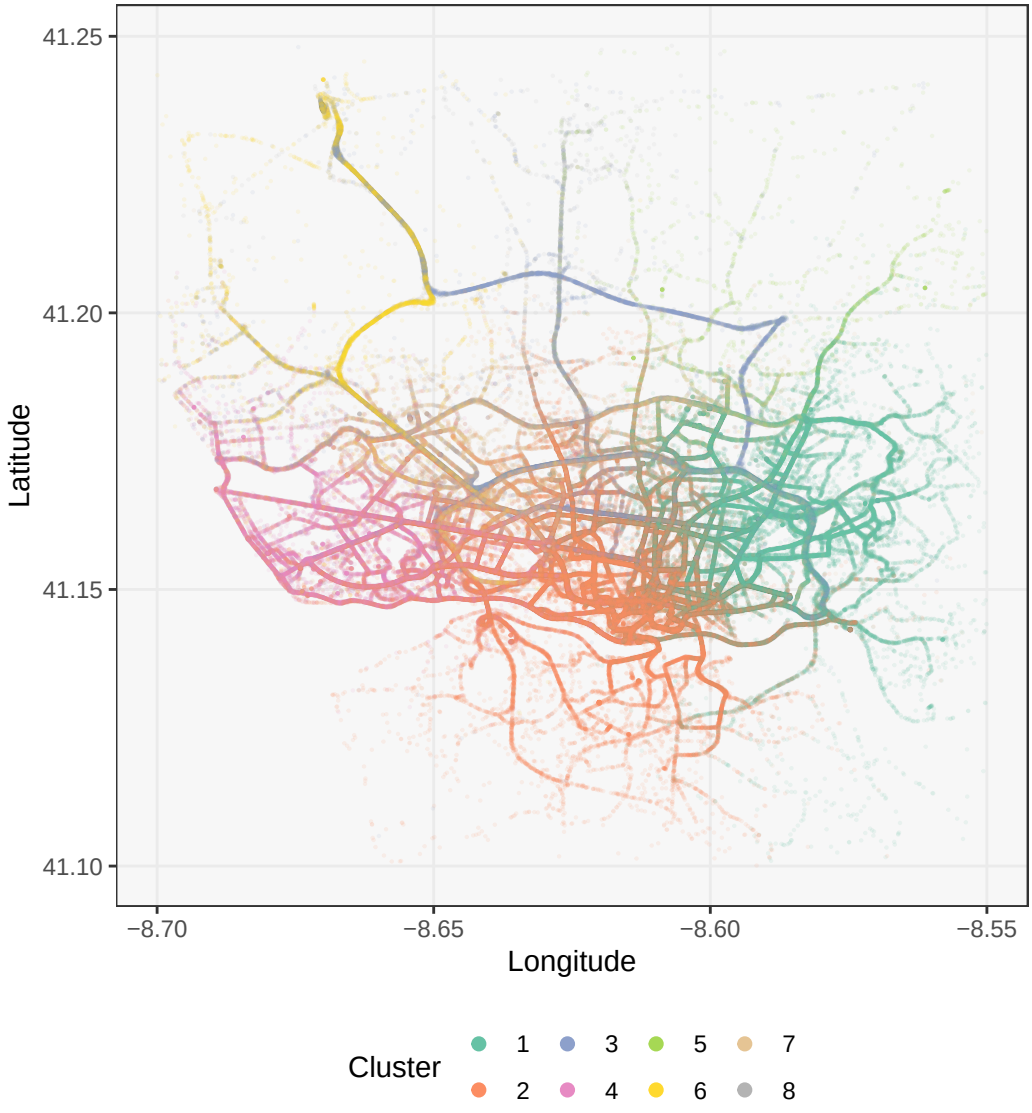

Figure 5: Q3 Part D

```
  geom_col(show.legend = FALSE) +
  facet_wrap(~ component, scales = "free_y") +
  scale_y_reordered() +
  labs(y = NULL) +
  theme(axis.text = element_text(size = 7), strip.text = element_text(size = 8))
```

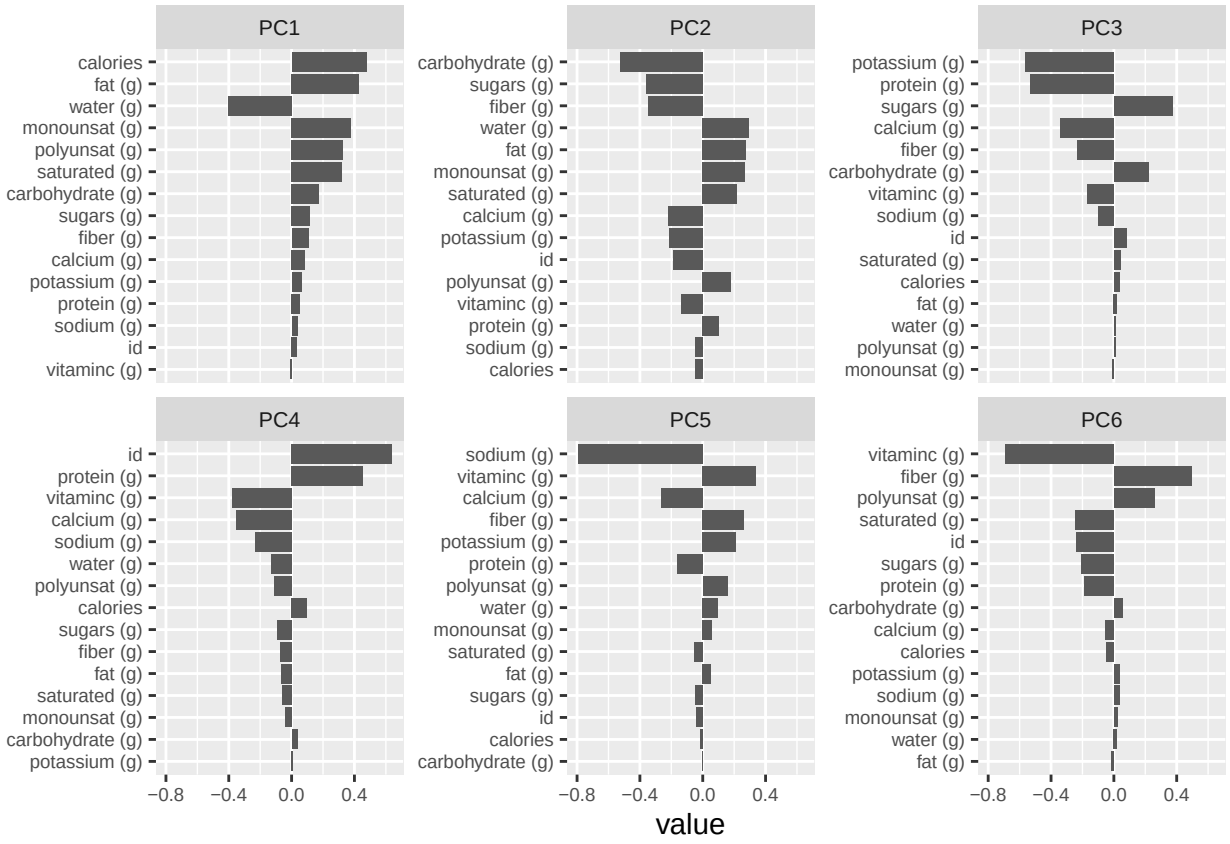

Figure 6: Q4 Part B

Part C

```
group_order = scores %>%
  group_by(group) %>%
  summarise(mean = mean(PC2)) %>%
  arrange(desc(mean)) %>%
  pull(group)

group_order
```

```
##  [1] Fats and Oils                      Sausages and Luncheon Meats
##  [3] Poultry Products                   Pork Products
##  [5] Beef Products                      Lamb, Veal, and Game Products
##  [7] Finfish and Shellfish Products     Dairy and Egg Products
```

```
##  [9] Soups, Sauces, and Gravies        Nut and Seed Products
## [11] Baby Foods                        Ethnic Foods
## [13] Vegetables and Vegetable Products Fast Foods
## [15] Restaurant Foods                  Fruits and Fruit Juices
## [17] Meals, Entrees, and Sidedishes    Legumes and Legume Products
## [19] Beverages                         Baked Products
## [21] Cereal Grains and Pasta           Snacks
## [23] Sweets                            Breakfast Cereals
## [25] Spices and Herbs
## 25 Levels: Baby Foods Baked Products Beef Products ... Vegetables and Vegetable Products
```

```
scores = scores %>% mutate(group = factor(group,  levels = group_order))
```

Part D

The results confirmed some of my guesses from part b. Specifically looking at baked products which is a high carbohydrate food, we can see they have low values of PC2. Fats and oils are very high in calories, and as predicted they have in general, large values of PC1. Fruit and fruit juices are composed primarily of water, and they are shown to have relatively low values of PC1.

```
variances_filtered = variances %>%  filter(terms == "percent variance")

ggplot(scores, aes(PC1, PC2, label = name)) +
  geom_point(alpha = 0.7, size = 1.5) +
  coord_fixed(sqrt(variances_filtered$value[2] / variances_filtered$value[1])) +
  facet_wrap(~ group, nrow = 5) +
  theme(axis.text = element_text(size = 7), strip.text = element_text(size = 8)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0)
```

(5) Modeling topics in *Pride and Prejudice*

```
paragraphs <- read_csv("https://uwmadison.box.com/shared/static/pz1lz301ufhbedzsj9iioee77r95xz4v.csv")
dtm <- paragraphs %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% stop_words$word) %>%
  count(paragraph, word) %>%
  cast_dtm(paragraph, word, n)
```

Part A

```
dtm_LDA = LDA(dtm, k = 6, control = list(seed = 479))
dtm_LDA
```

```
## A LDA_VEM topic model with 6 topics.
```

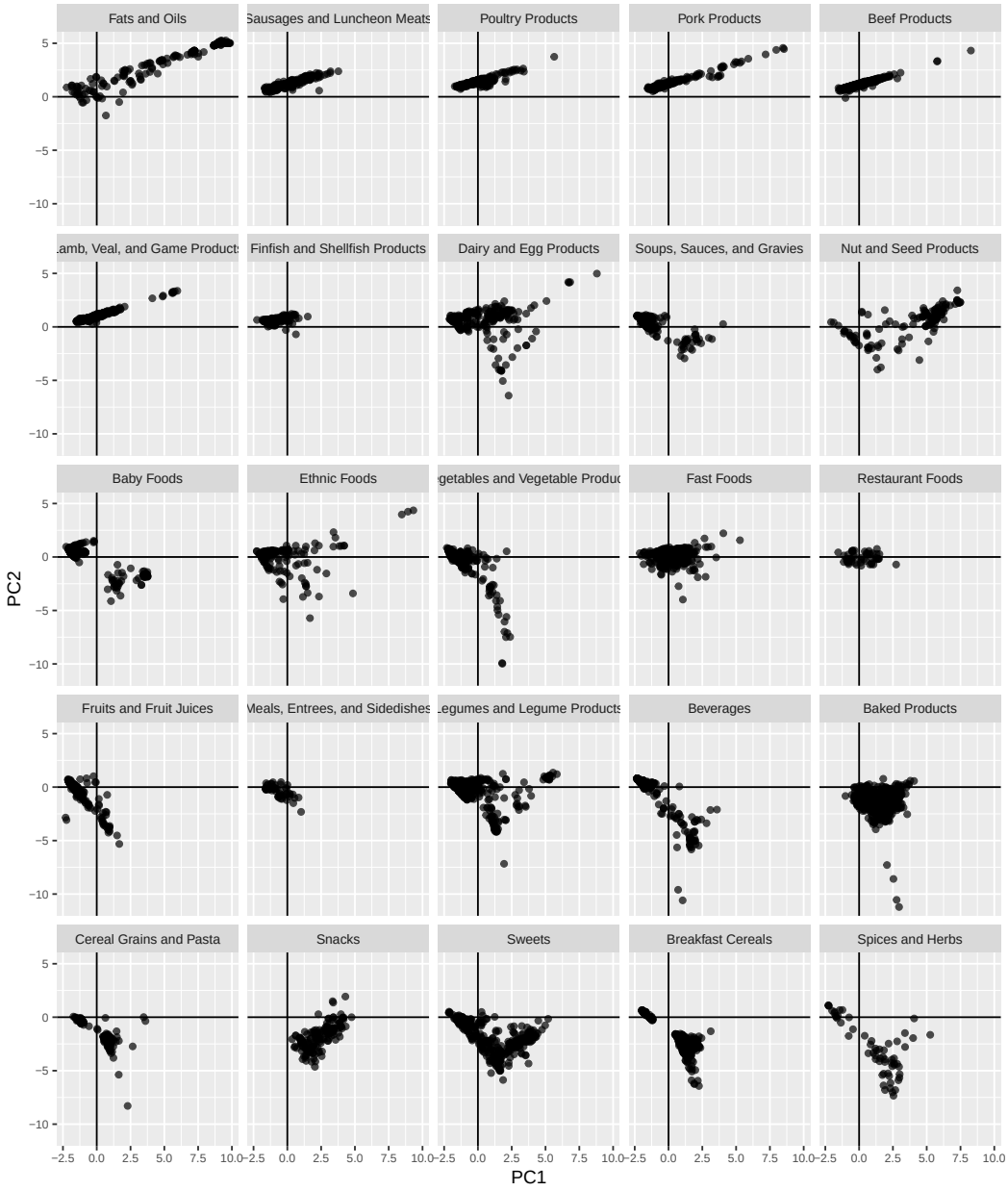

Figure 7: Q4 Part B

```
dtm_topics = tidy(dtm_LDA, matrix = "beta")
dtm_memberships = tidy(dtm_LDA, matrix = "gamma")

dtm_topics %>%
  arrange(topic, -beta)
```

```
## # A tibble: 33,930 x 3
##    topic term         beta
##    <int> <chr>       <dbl>
##  1     1 darcy     0.00967
##  2     1 sister    0.00923
##  3     1 elizabeth 0.00856
##  4     1 wickham   0.00807
##  5     1 hope      0.00709
##  6     1 jane      0.00657
##  7     1 friend    0.00650
##  8     1 bingley   0.00598
##  9     1 marriage  0.00576
## 10     1 feelings  0.00575
## # ... with 33,920 more rows
```

```
dtm_memberships %>%
  arrange(document, topic)
```

```
## # A tibble: 6,552 x 3
##    document topic   gamma
##    <chr>    <int>   <dbl>
##  1 1            1 0.242
##  2 1            2 0.00417
##  3 1            3 0.00417
##  4 1            4 0.00417
##  5 1            5 0.00417
##  6 1            6 0.742
##  7 10           1 0.981
##  8 10           2 0.00388
##  9 10           3 0.00388
## 10 10           4 0.00388
## # ... with 6,542 more rows
```

Part B

```
top_terms = dtm_topics %>%
  group_by(topic) %>%
  slice_max(beta, n = 30) %>%
  mutate(term = reorder_within(term, -beta, topic))

ggplot(top_terms, aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_x") +
  scale_fill_brewer(palette = "Set2") +
```

```
  scale_x_reordered() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(axis.text = element_text(size = 5)) +
  labs(
    x = "Term",
    y = "Beta"
  )
```

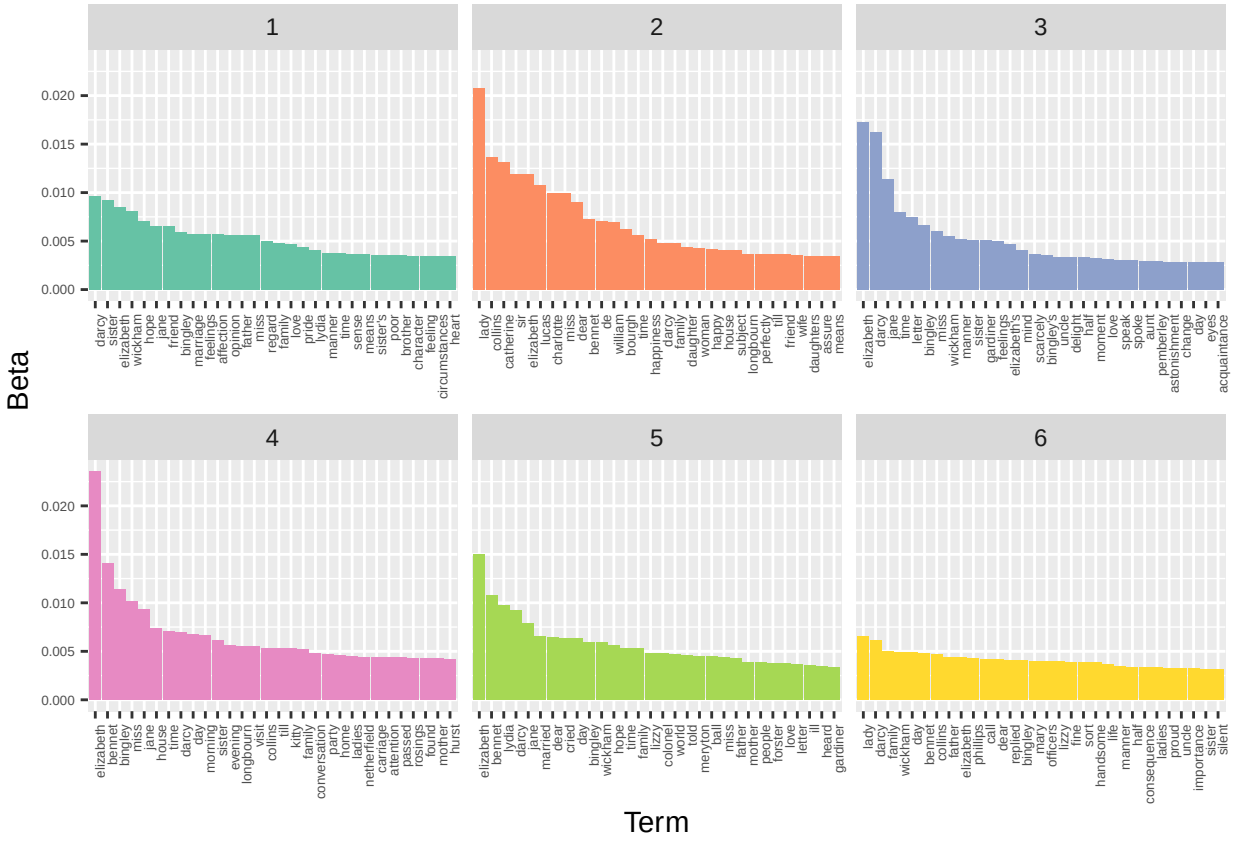

Figure 8: Q5 Part B

Part C

The last code chunk in this section gives the paragraph. I have confirmed that many of the words with high beta values for this topic appear in this paragraph. The first sentence of this paragraph is as follows: * sir william and lady lucas were speedily applied to for their consent; and it was bestowed with a most joyful alacrity.

```
member_filter = dtm_topics %>%
  filter(topic == 2)

dtm_memberships %>%
  filter(topic == 2) %>%
  filter(gamma == max(gamma)) %>%
  arrange(document, topic, gamma)
```

```
## # A tibble: 1 x 3
##   document topic gamma
##   <chr>    <int> <dbl>
## 1 347          2 0.998
```

```
purest_para = paragraphs %>%
  filter(paragraph == 347) %>%
  arrange(paragraph, text)
```

(6) Single-Cell Genomics

```
pbmc = read_csv("https://uwmadison.box.com/shared/static/ai539s30rjsw5ke4vxbjrxjaiihq7edk.csv")
```

Part A

```
umap_recipe = recipe(~., data = pbmc) %>%
  update_role(cell_tag, GNLY, new_role = "id") %>%
  step_umap(all_predictors(), neighbors = 5, learn_rate = 0.1)
umap_prep = prep(umap_recipe)
scores = juice(umap_prep)
```

Part B

```
ggplot(scores, aes(umap_1, umap_2)) +
  geom_point(aes(color = GNLY)) +
  scale_color_viridis_c()
```

Feedback

a. I spent around 15 hours on this homework.
b. I found question 5 about *Pride and Prejudice* to be most valuable.

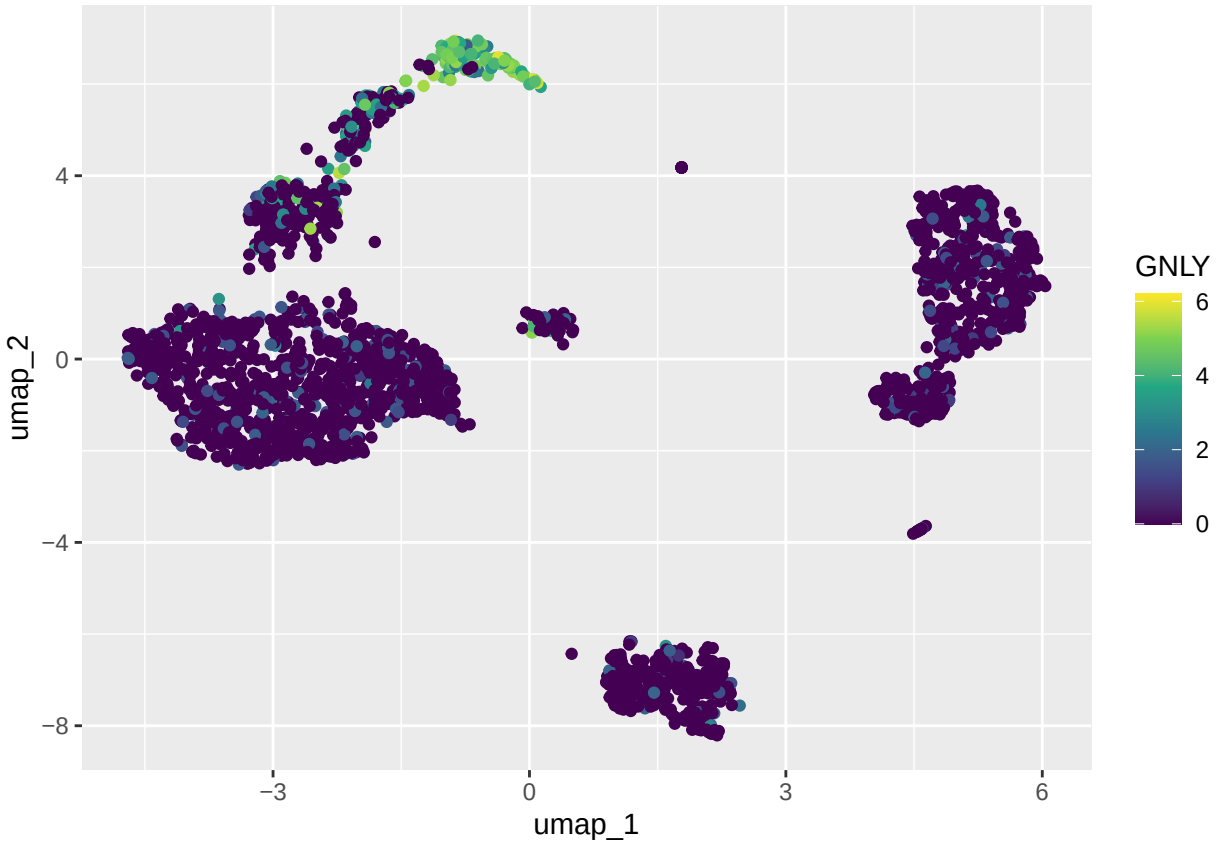

Figure 9: Q6 Part B