Meetup 12: Hierarchical Clustering

George I. Hagstrom

Week Summary

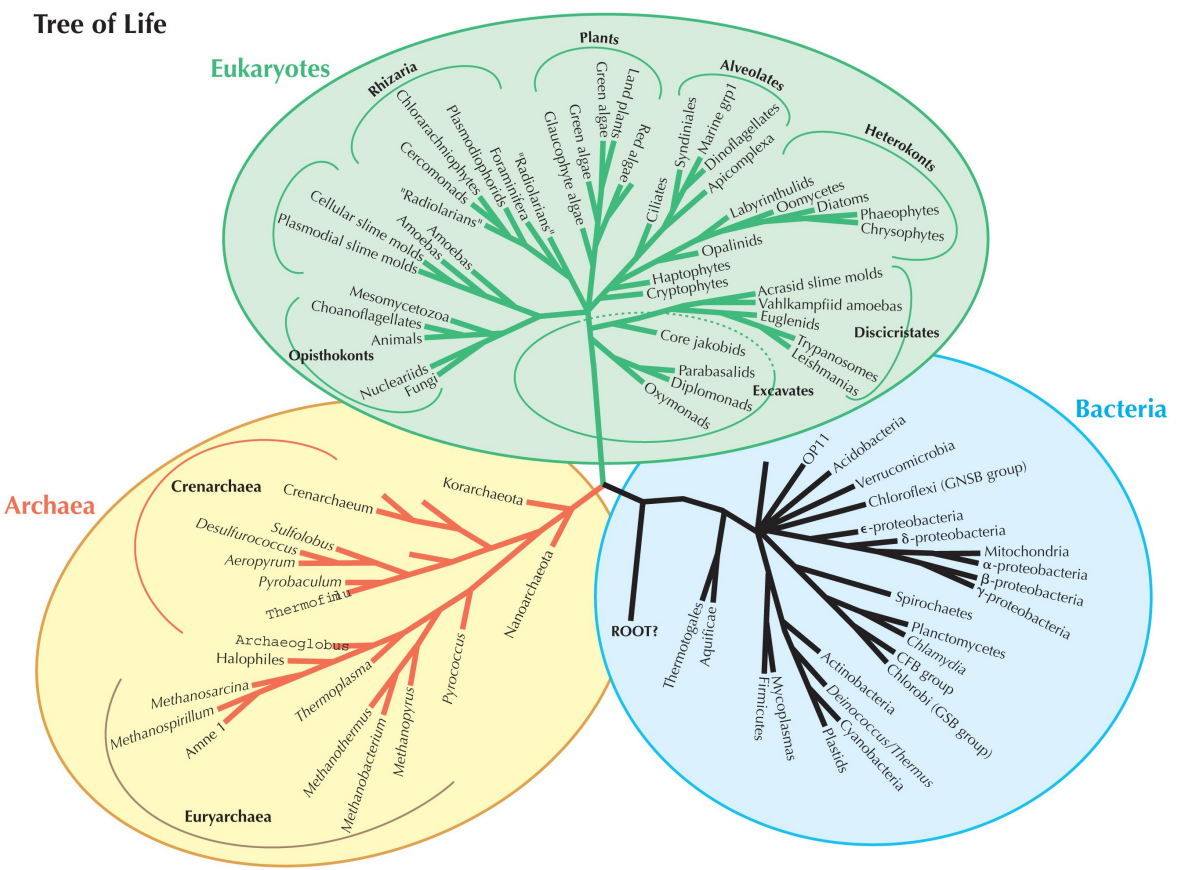
- Entering home stretch
  - This week finish clustering
  - Talk about making plots interactive
- Finish with Shiny Apps

Reading

1. Wikipedia article on Hierarchical Clustering
2. ggdendro introduction and reference
3. tidy_heatmaps documentation
4. Bioinformatics Training and Education Program Tutorial on Hierarchical Clustering
5. For Python: Seaborn Clustermap
6. For Python: Scipy Clusters

Tree of Life

- Lots of data has a hierarchical structure



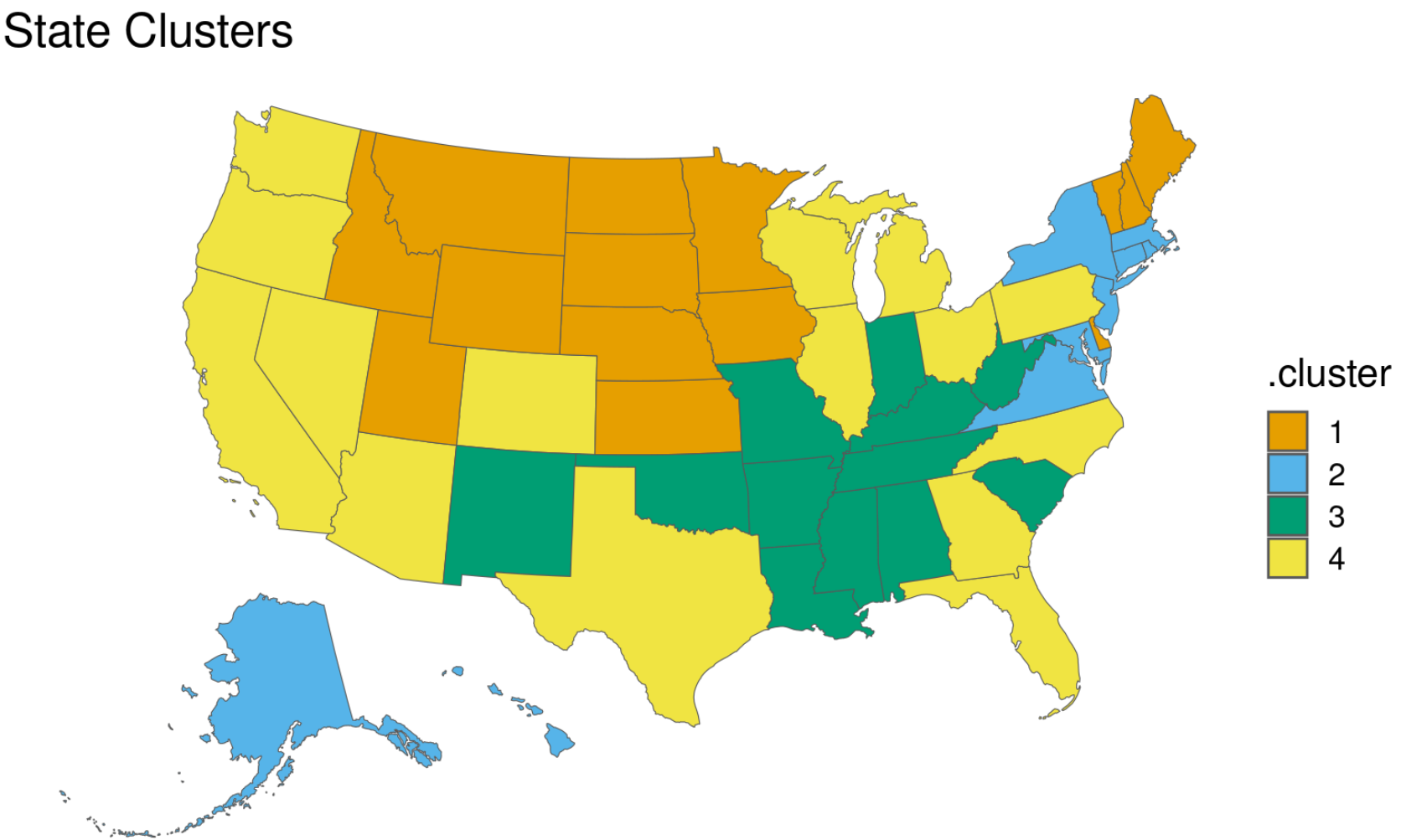
Jonathan Eisen

Statistics of US States

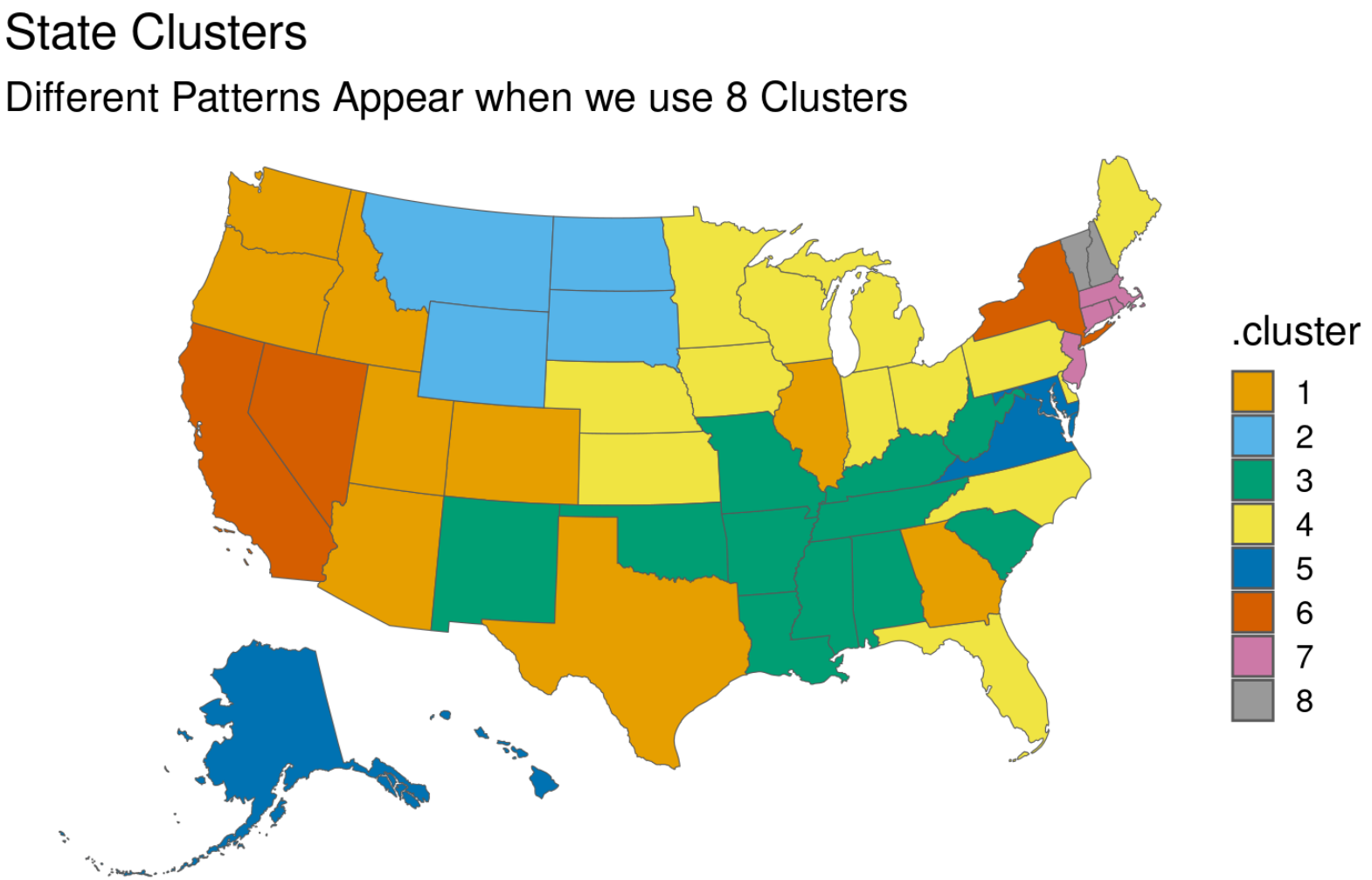
```
  1  US_state_stats |> head(n=15)
```

```
# A tibble: 15 × 20
   state       homeownership multiunit income med_income poverty fed_spend smoke
   <chr>               <dbl>     <dbl>  <dbl>      <dbl>   <dbl>     <dbl> <dbl>
 1 Alabama              71.1      15.5  22984      42081    17.1      11.7   24.8
 2 Alaska               64.7      24.6  30726      66521     9.5      16.8   25
 3 Arizona              67.4      20.7  25680      50448    15.3       9.85  20.4
 4 Arkansas             67.7      15.2  21274      39267    18         9.61  23.5
 5 California           57.4      30.7  29188      60883    13.7       8.89  15.2
 6 Colorado             67.6      25.6  30151      56456    12.2       9.15  19.9
 7 Connecticut          69.2      34.6  36775      67740     9.2      14.8   16.5
 8 Delaware             73.6      17.7  29007      57599    11         8.89  20.7
 9 Florida              69.7      30    26551      47661    13.8       9.62  21.6
10 Georgia              67.2      20.5  25134      49347    15.7       8.88  22.2
11 Hawaii               59.3      39.2  28882      66420     9.6      14.0   17.1
12 Idaho                71        14.9  22518      46423    13.6       8.82  17.9
13 Illinois             69.2      33    28782      55735    12.6       8.52  19.9
14 Indiana              71.5      18.6  24058      47697    13.5       9.45  27.3
15 Iowa                 73.2      18.6  25335      48872    11.6       9.50  20.4
# ℹ 12 more variables: murder <dbl>, robbery <dbl>, agg_assault <dbl>,
#   larceny <dbl>, motor_theft <dbl>, soc_sec <dbl>, nuclear <dbl>, coal <dbl>,
#   tr_deaths <dbl>, tr_deaths_no_alc <dbl>, unempl <dbl>, popdens2010 <dbl>
```

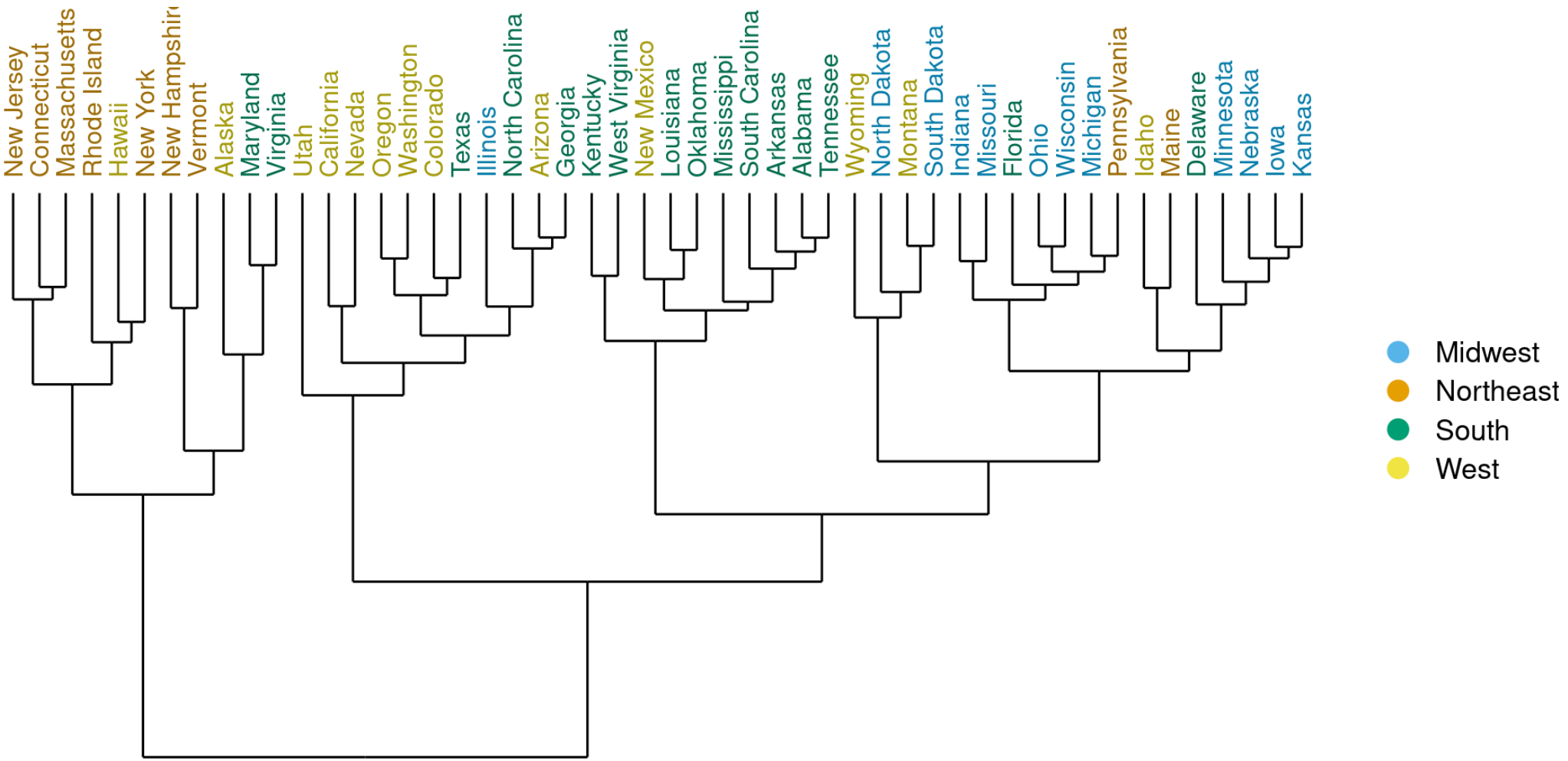
Could Cluster Them



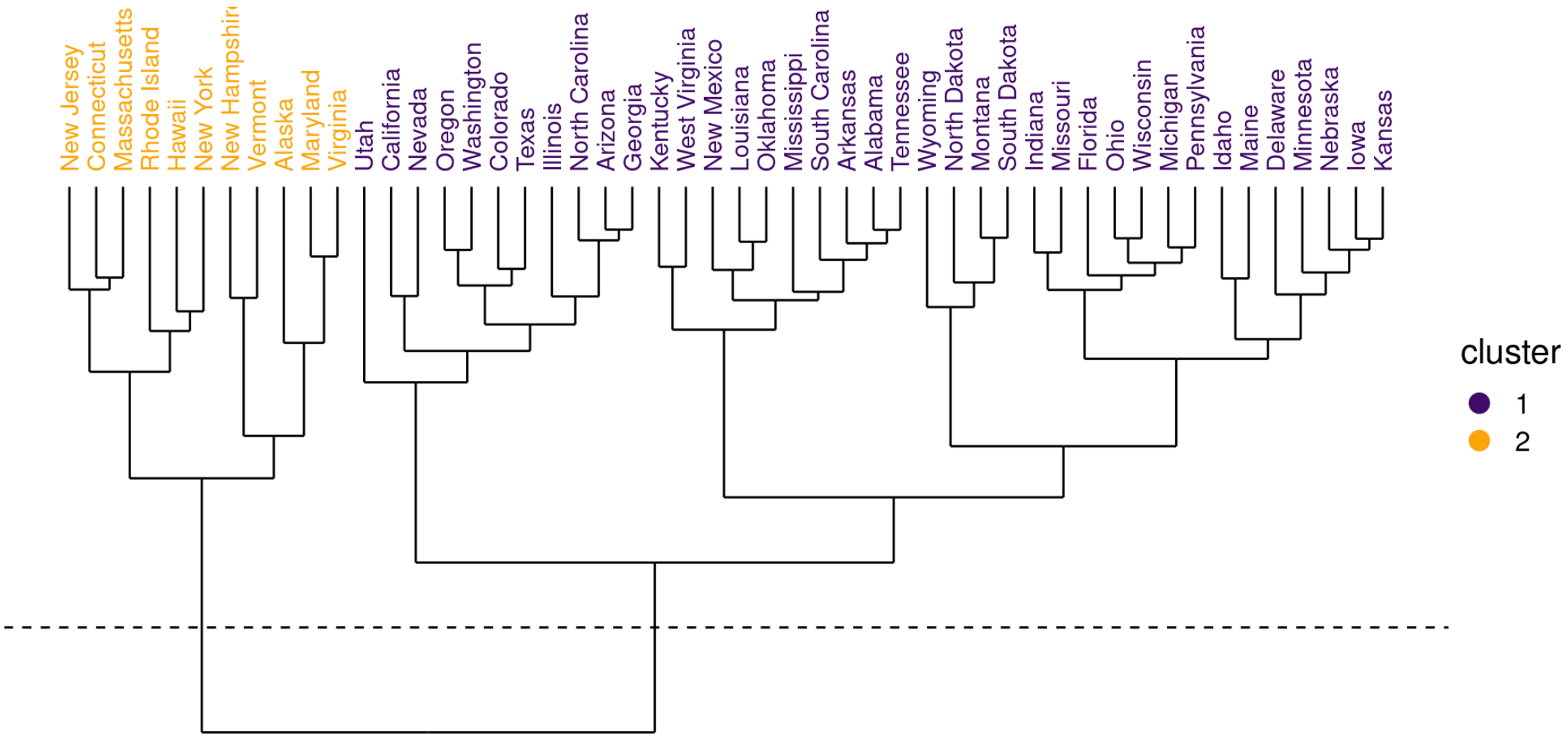
Different Levels and Meaning



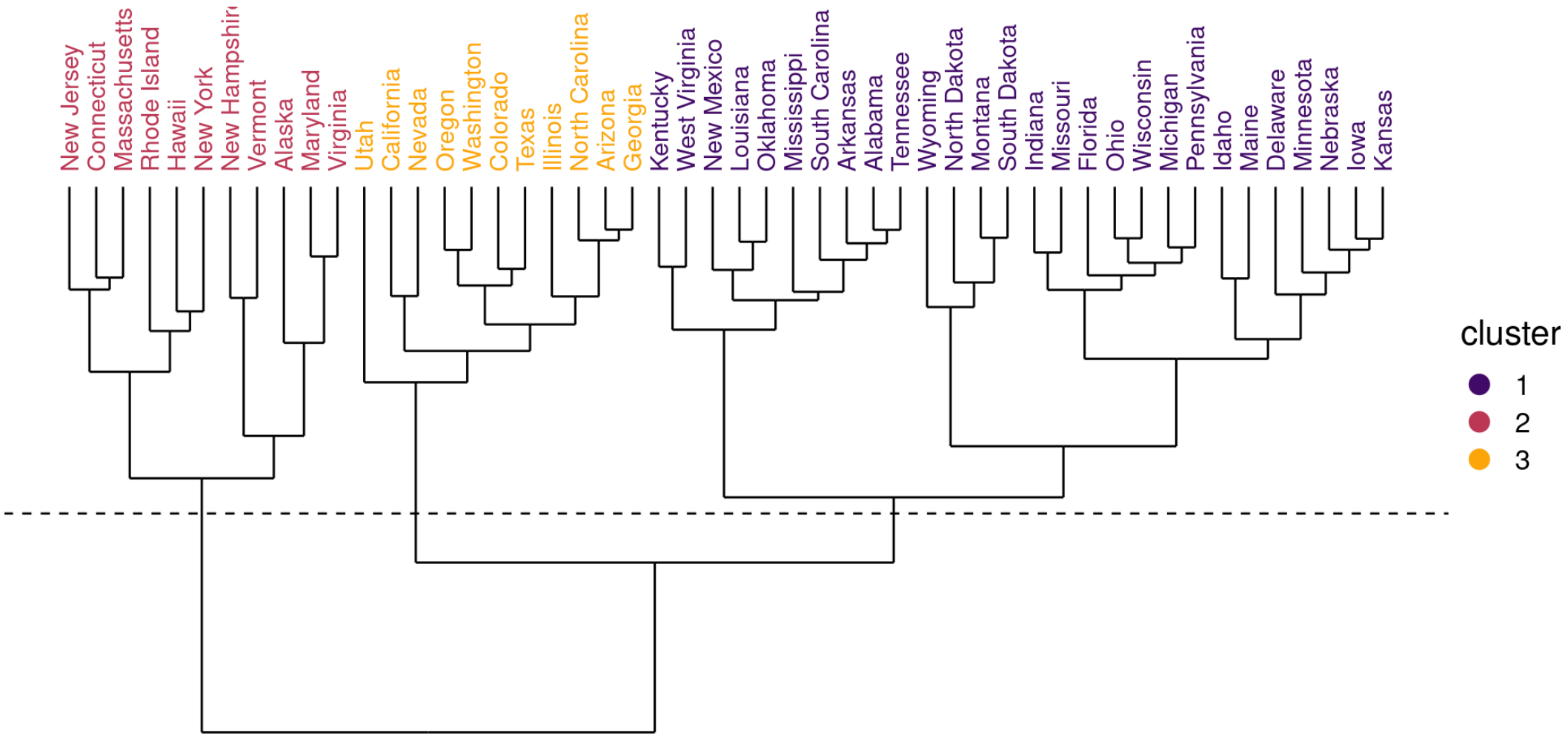
Hierarchical Clustering is Bottom-Up



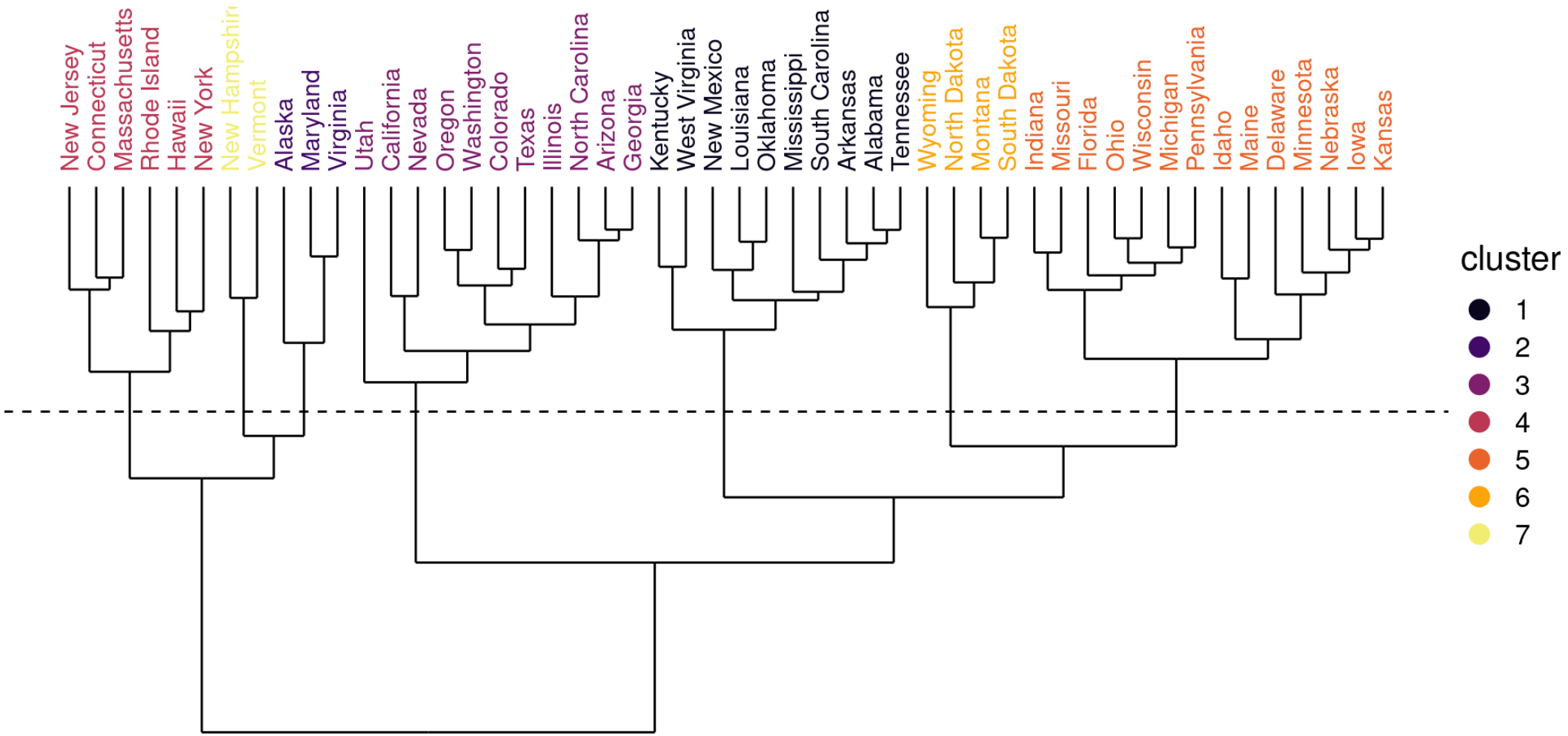
Dendrogram Clusters



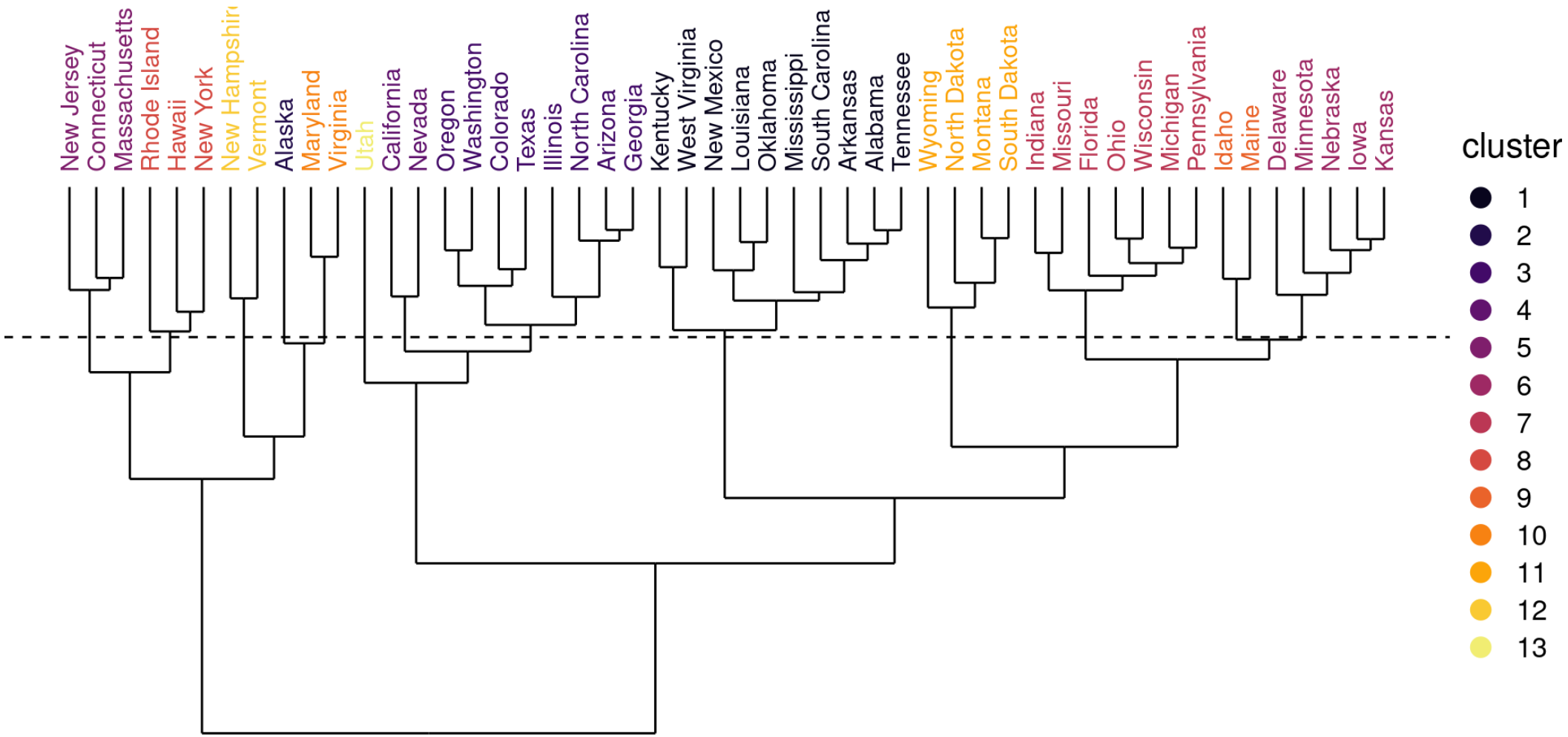
Dendrogram Cuts



Dendrogram Cuts



Dendrogram Cuts



Why Hierarchical Clustering?

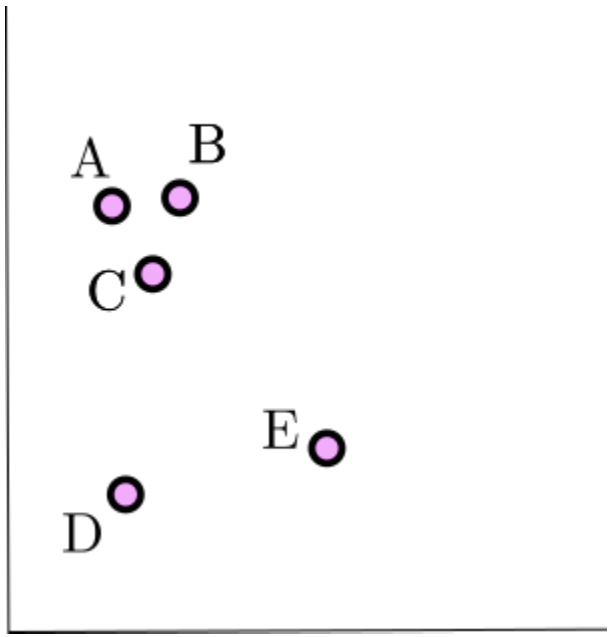
- You care about fine details
  - Can find clusters of small numbers of points
  - Biological Species
- You want to explore multiple different clustering scales
- You want robustness to outliers

How does it work?

- Needs two basic ingredients:
  1. Distance function between the points
  2. Rule for applying the distance to clusters

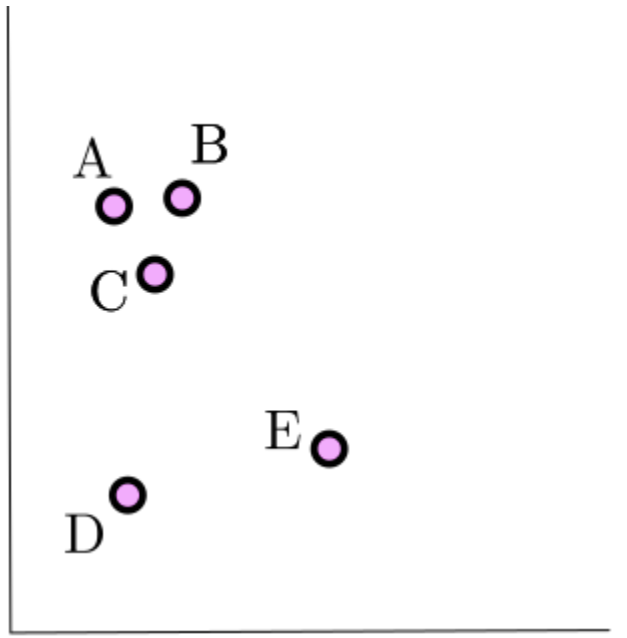
Agglomerative Clustering

- Begin with points



Agglomerative Clustering

- Begin with points
- Distance Matrix

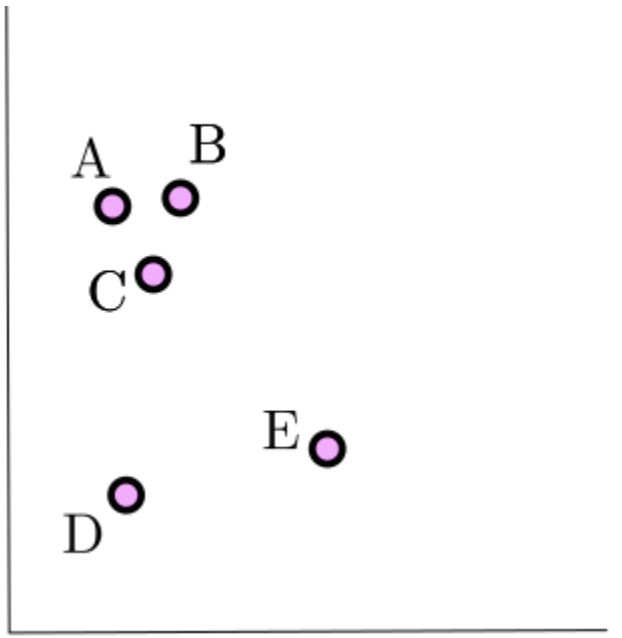


	A	B	C	D	E
A	0	0.2	0.3	1.4	1.8
B	0.2	0	0.3	1.2	1.6
C	0.3	0.3	0	1.4	1.6
D	1.4	1.2	1.4	0	0.8
E	1.8	1.6	1.5	0.8	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest

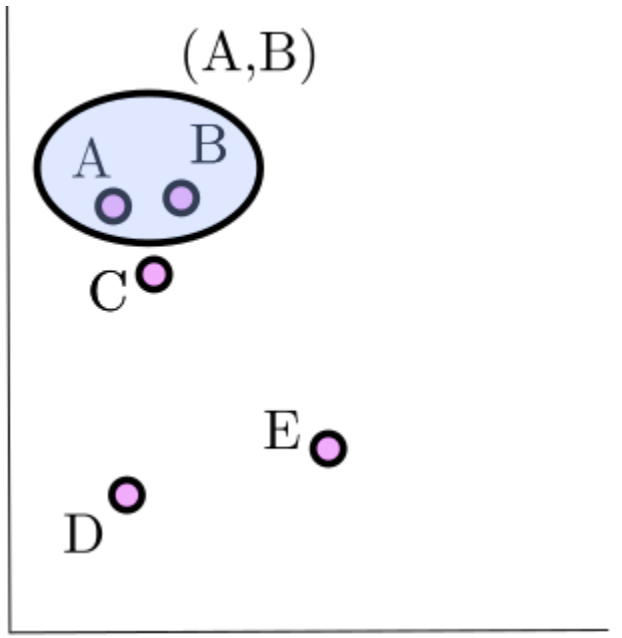


	A	B	C	D	E
A	0	**0.2**	0.3	1.4	1.8
B	**0.2**	0	0.3	1.2	1.6
C	0.3	0.3	0	1.4	1.5
D	1.4	1.2	1.4	0	0.8
E	1.8	1.6	1.5	0.8	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances

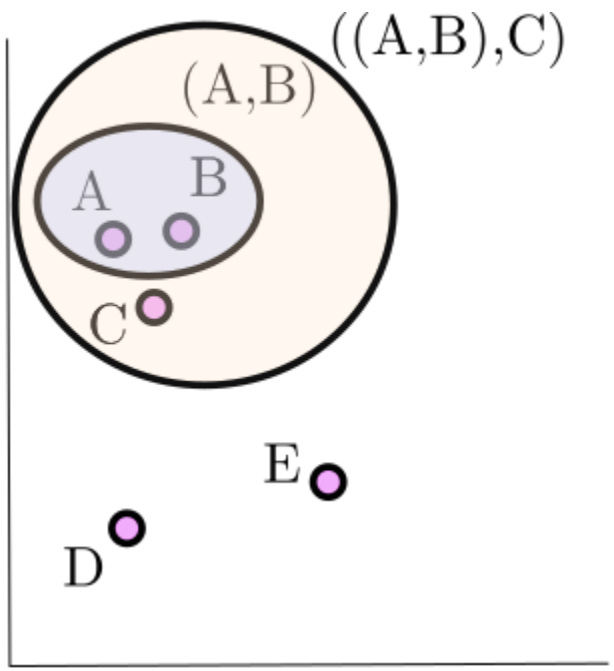


	(A,B)	C	D	E
(A,B)	0	0.3	1.3	1.7
C	0.3	0	1.4	1.5
D	1.3	1.4	0	0.8
E	1.7	1.5	0.8	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances
- Rinse and Repeat

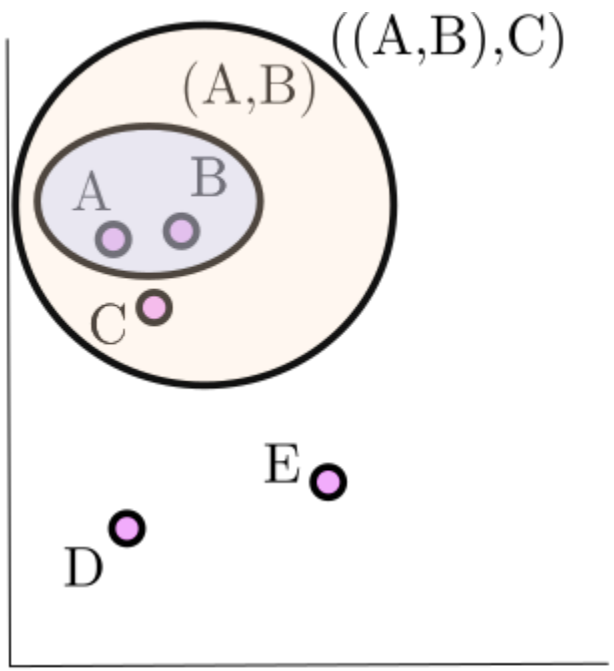


	(A,B)	C	D	E
(A,B)	0	0.3	1.3	1.7
C	0.3	0	1.4	1.5
D	1.3	1.4	0	0.8
E	1.7	1.5	0.8	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances
- Rinse and Repeat

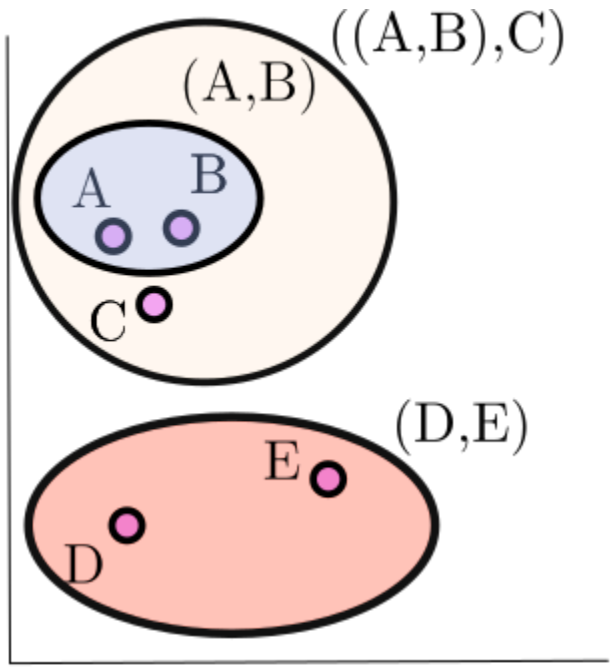


	((A,B),C)	D	E
((A,B),C)	0	1.2	1.6
D	1.2	0	0.8
E	1.6	0.8	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances
- Rinse and Repeat

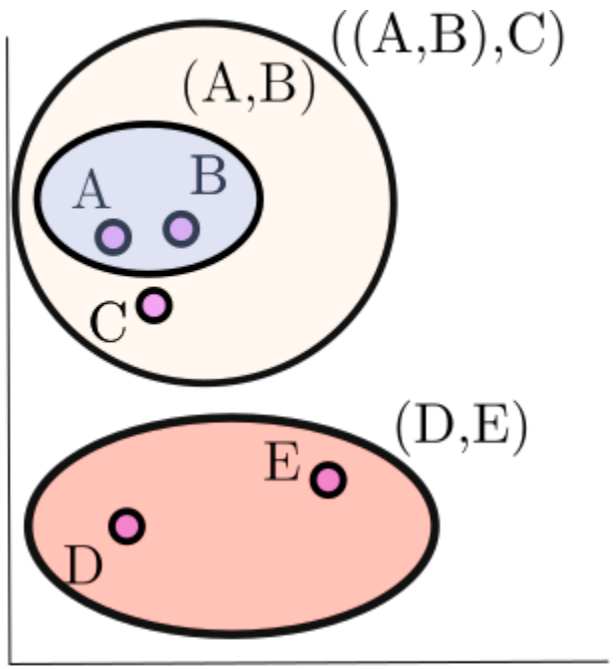


	((A,B),C)	D	E
((A,B),C)	0	1.2	1.6
D	1.2	0	0.8
E	1.6	0.8	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances
- Rinse and Repeat

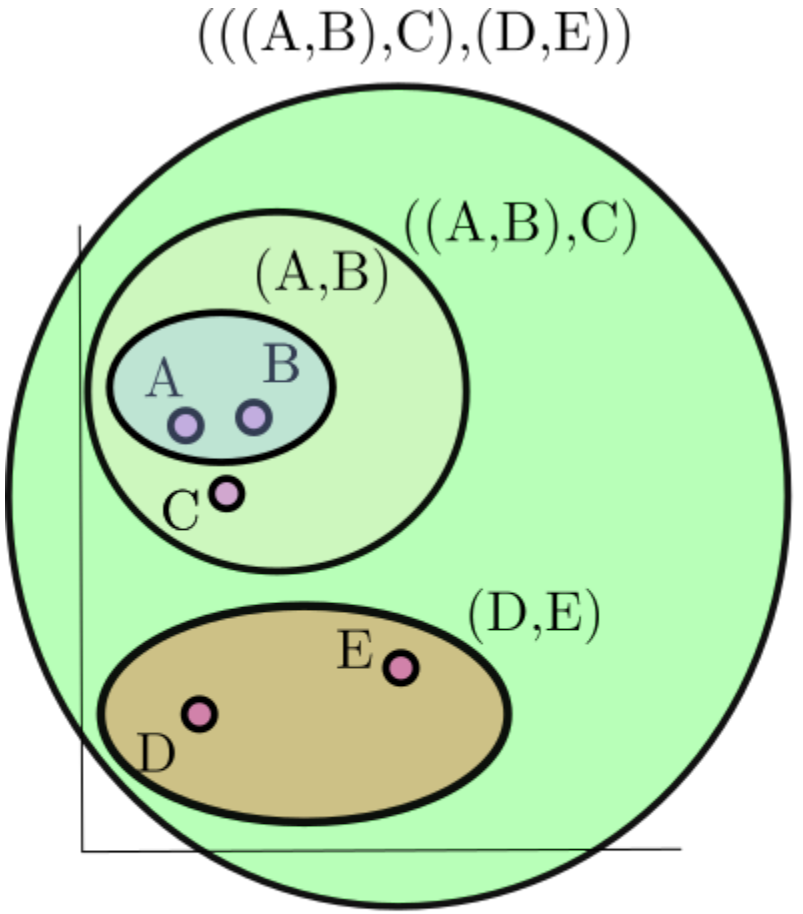


	((A,B),C)	(D,E)
((A,B),C)	0	1.4
(D,E)	1.4	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances
- Rinse and Repeat

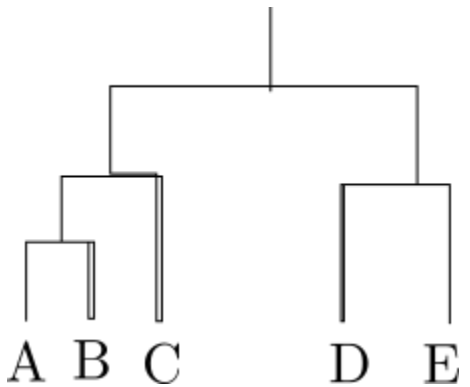


	((A,B),C)	(D,E)
((A,B),C)	0	1.4
(D,E)	1.4	0

Table 1: Distance matrix

Agglomerative Clustering

- Begin with points
- Distance Matrix
- Group Closest
- Recalculate Distances
- Rinse and Repeat
- Form Dendrogram



Distance Metrics

- Euclidean Distance (Most Common)
- Correlation Distance (Common in Bio)
- Manhattan, Maximum (Other Metrics)
- Categorical Metrics:
  - Gower's (Mixed categorical and numerical)
  - Jaccard, Hamming, many more

Linkage Methods

- How to determine the distance between clusters?
- Many on wikipedia
  - Max or Min distance between points
  - UPGMA (unweighted average distance between pairs)
  - Ward's Linkage (minimize increase in within cluster variance)

Implementing Hierarchical Clustering

- Demonstrate it on a Kaggle Customer Segmentation Dataset

```
1 mall_customers = read_csv("Mall_Customers.csv")
2 mall_customers
```

```
# A tibble: 200 × 5
   CustomerID Gender   Age `Annual Income (k$)` `Spending Score (1-100)`
        <dbl> <chr>  <dbl>                <dbl>                    <dbl>
 1          1 Male      19                   15                       39
 2          2 Male      21                   15                       81
 3          3 Female    20                   16                        6
 4          4 Female    23                   16                       77
 5          5 Female    31                   17                       40
 6          6 Female    22                   17                       76
 7          7 Female    35                   18                        6
 8          8 Female    23                   18                       94
 9          9 Male      64                   19                        3
10         10 Female    30                   19                       72
# ℹ 190 more rows
```

Calculate Distance Matrix

- Ignore Categorical

```
mall_customers |>
  select(-Gender) |>
  column_to_rownames(var = "CustomerID")  |>
  scale()  |>
  dist(method = "euclidean")
```

```
             1          2          3          4          5          6
2   1.63271382
3   1.28047443 2.90546048
4   1.49961180 0.21434013 2.75780621
5   0.86328190 1.74328990 1.53461759 1.54348731
6   1.45080775 0.22002872 2.71475242 0.08985491 1.53575852
7   1.71988710 3.07451056 1.07650230 2.88151502 1.34794551 2.86626187
8   2.15203749 0.53569963 3.41535546 0.66270395 2.16845573 0.70173898
9   3.51342790 4.31534552 3.15404067 4.10357961 2.76396907 4.12759374
10  1.50874907 0.74817170 2.65662924 0.54922364 1.24357994 0.59814119
11  3.57319907 4.19507763 3.38075346 3.98576234 2.76787491 4.01842070
12  2.59491799 1.23024367 3.75978131 1.21523657 2.30387170 1.29040710
13  2.94867334 3.68566467 2.74676800 3.47351925 2.16476168 3.49779854
14  1.52635147 0.32612286 2.76849453 0.16828143 1.52219560 0.18720306
15  1.64631705 2.87788357 1.25607051 2.67766834 1.13610226 2.66794498
```

Calculate Distance Matrix

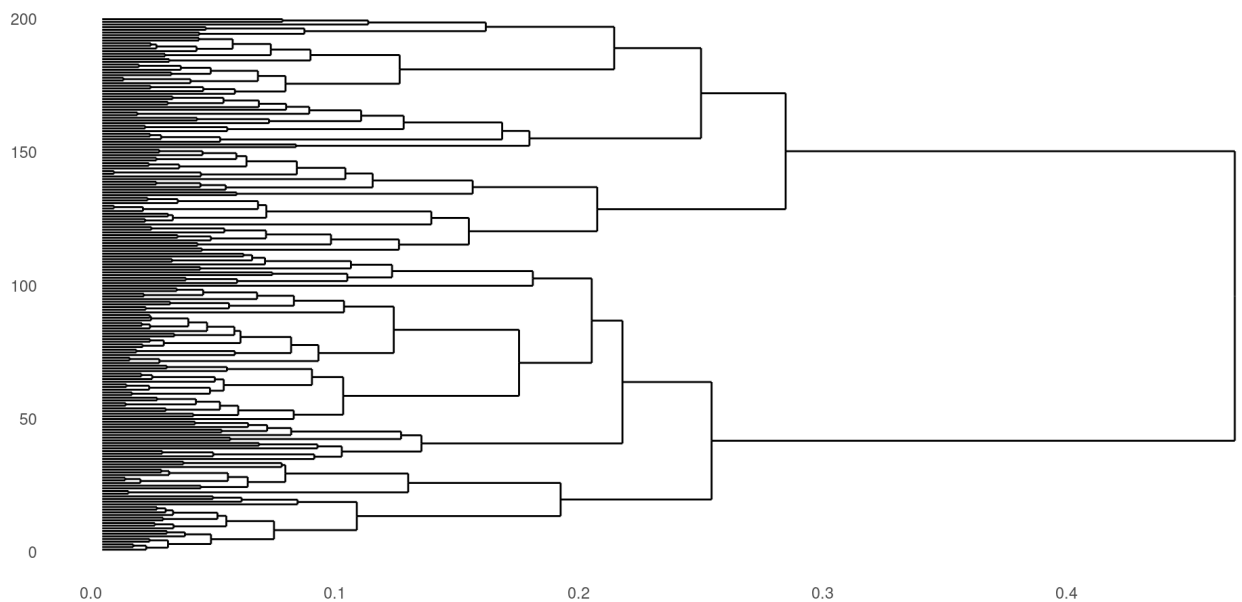
- Keep Categorical

```
# Notice use of `gower` cluster package/daisy function
library(cluster)
mall_dist = mall_customers |> mutate(Gender = as_factor(Gender)) |>
  column_to_rownames(var = "CustomerID")  |>
  daisy(metric = "gower")
```

Create Clusters

```r
library(ggdendro)
mall_clusters = mall_dist |>
  hclust(
  method = "average"
)

mall_clusters |>
  ggdendrogram( rotate = TRUE,labels=FALSE,leaf_labels = FALSE)
```



Tree objects

```
List of 7
 $ merge      : int [1:199, 1:2] -130 -66 -61 -4 -49 -115 -12 -113 -101 -158
...
 $ height     : num [1:199] 0.00481 0.00481 0.0087 0.00941 0.00962 ...
 $ order      : int [1:200] 148 158 160 126 134 140 154 166 176 182 ...
 $ labels     : chr [1:200] "1" "2" "3" "4" ...
 $ method     : chr "average"
 $ call       : language hclust(d = mall_dist, method = "average")
 $ dist.method: NULL
 - attr(*, "class")= chr "hclust"
```

Cut to Assign Clusters

- `cutree` function assigns clusters at a given depth
- `cutree(tree,k= num_groups)`
- `cutree(tree,h=depth_groups)`

```
# A tibble: 200 × 2
   names     x
   <chr> <int>
 1 1         1
 2 2         1
 3 3         2
 4 4         3
 5 5         2
 6 6         3
 7 7         2
 8 8         3
 9 9         4
10 10        3
# ℹ 190 more rows
```

Visualizing Clusters

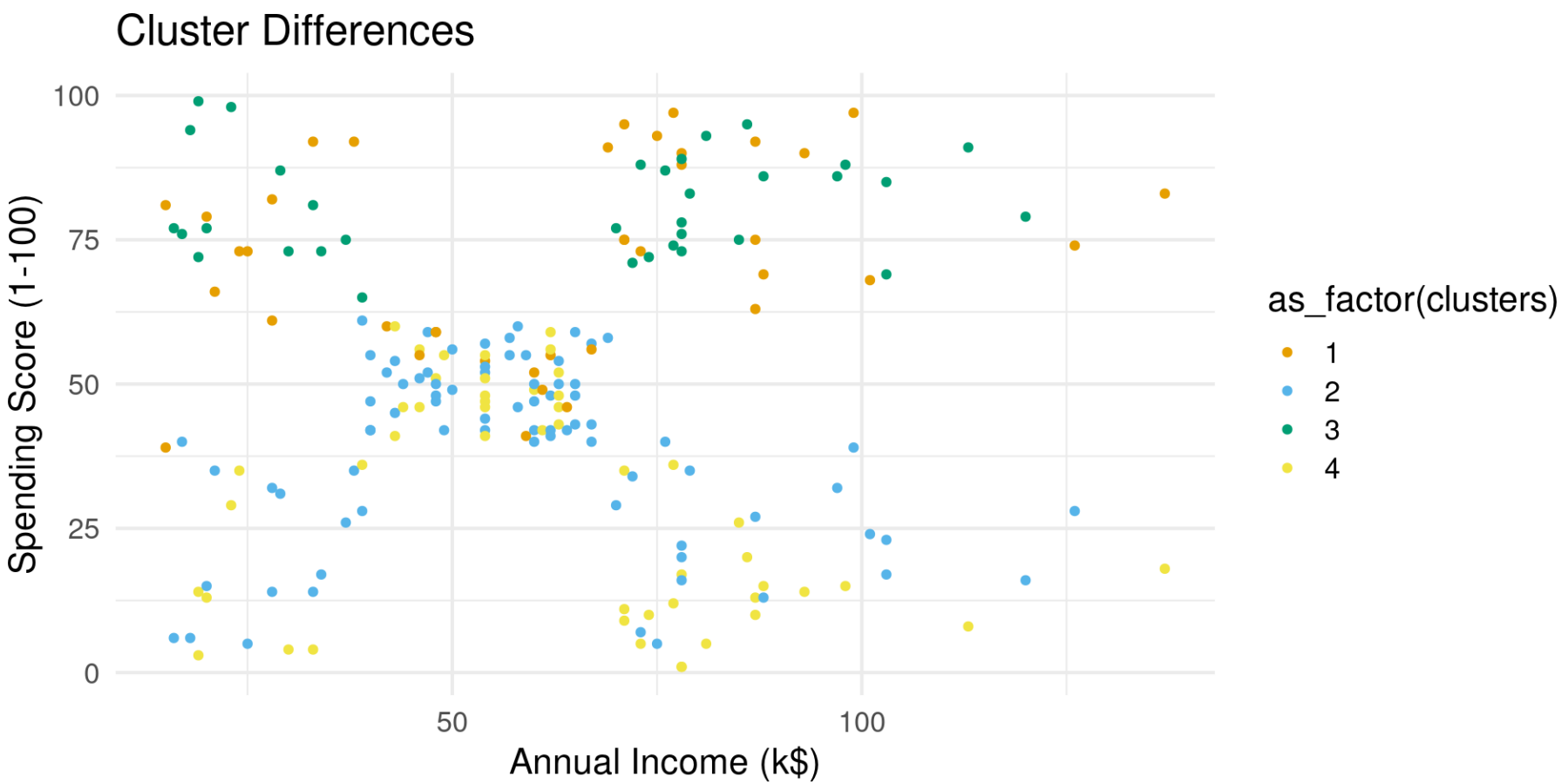
- Join clusters with original data frame

```
mall_customers = mall_customers |> left_join(
  mall_clusters |> cutree(k=4) |> tidy() |> mutate(names = as.numeric(names
  by = c("CustomerID" = "names") )|>
  rename(clusters = x)

mall_customers |> ggplot(aes(x = `Annual Income (k$)`,y=`Spending Score (1-
  geom_point()  +
  scale_color_okabe_ito() +
  scale_fill_okabe_ito()
```

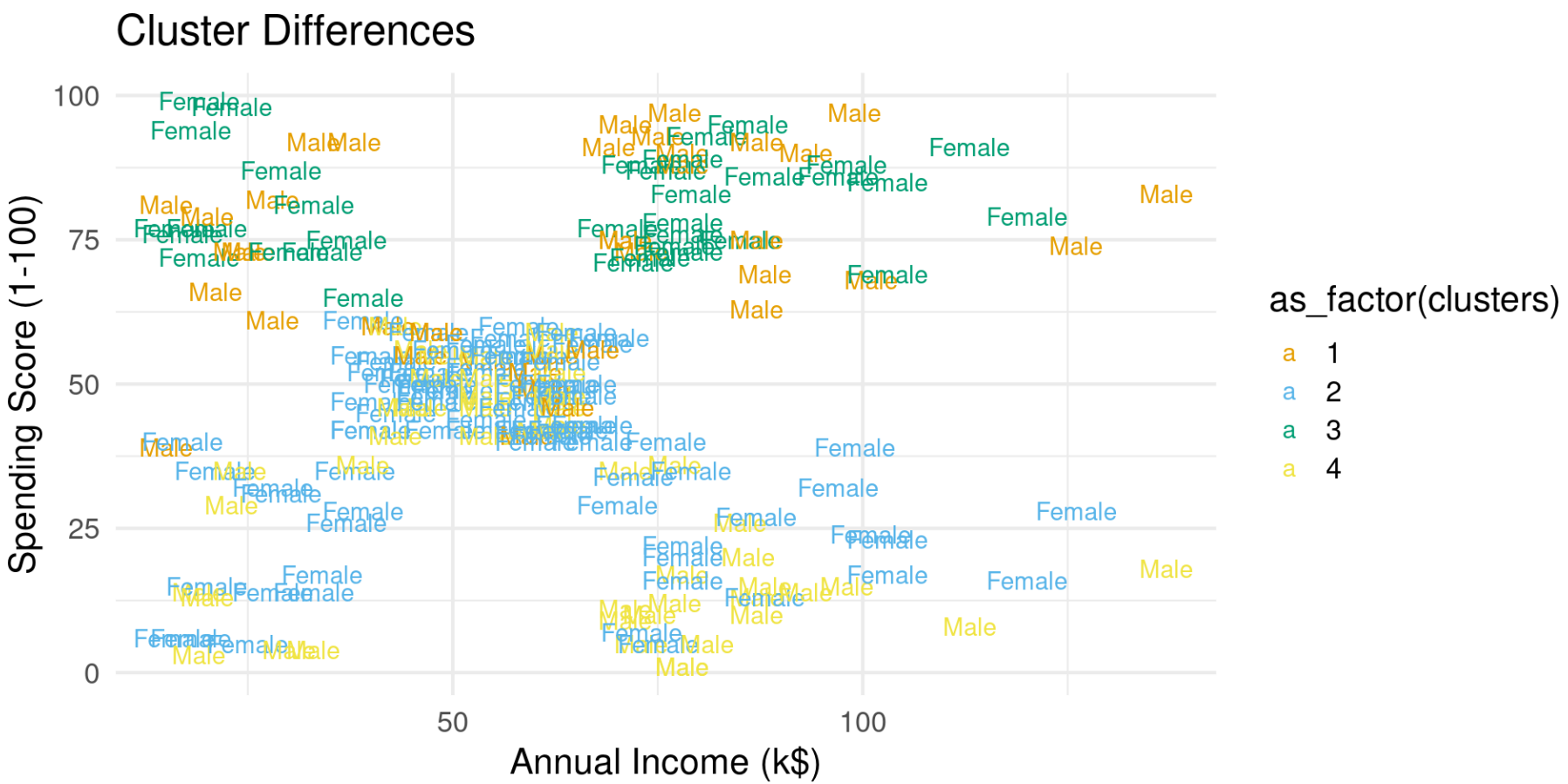
Visualizing Clusters

- Join clusters with original data frame



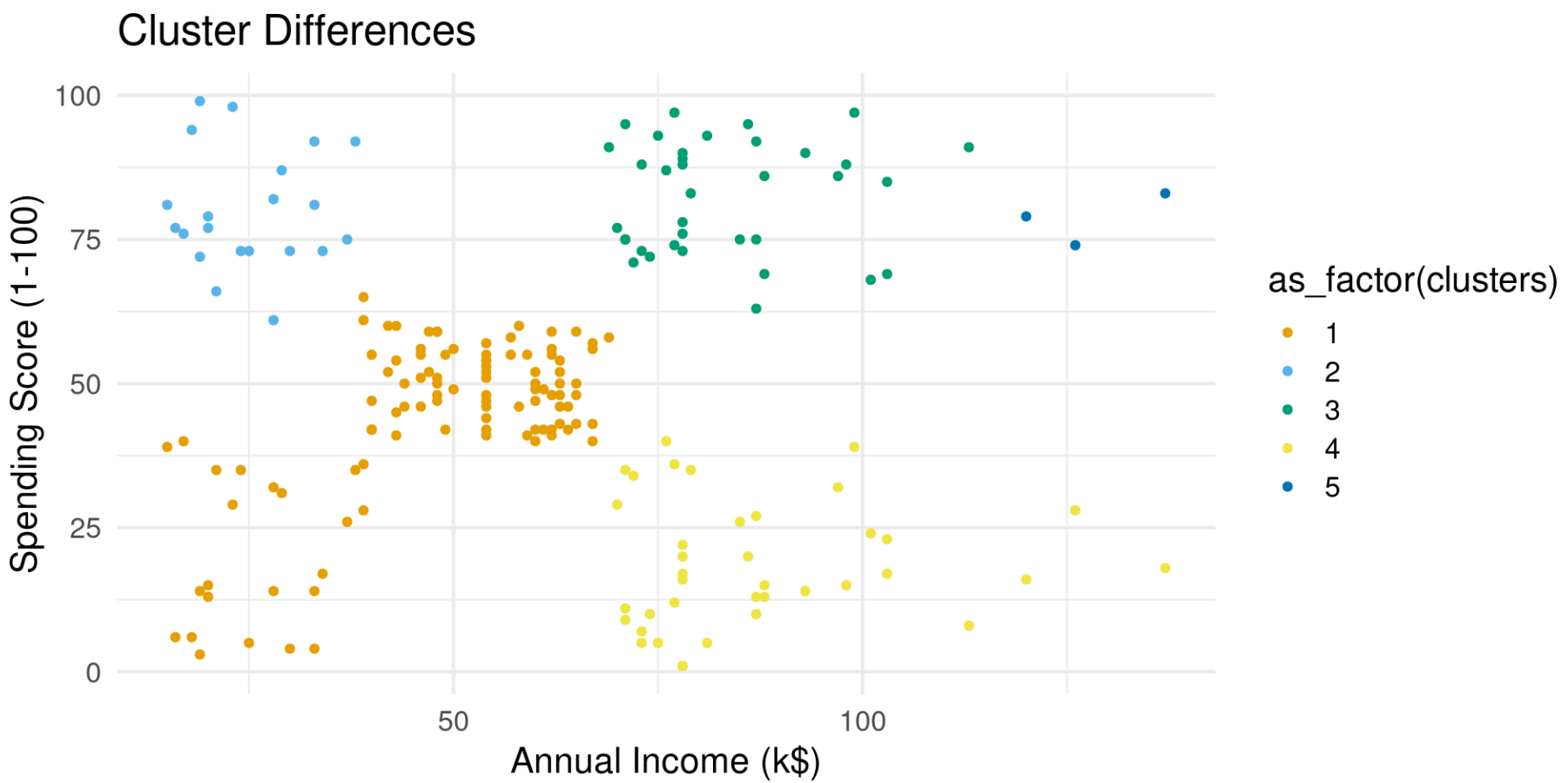
Visualizing Clusters

- Seems like Gender is main differentiator



Try Something Else

- Dropped Gender and Age



Try a Different Linkage

- Use the Ward Linkage Instead

```
mall_clusters = mall_dist |>
  hclust(
  method = "ward"
)

mall_customers = mall_customers |> left_join(
  mall_clusters |> cutree(k=5) |> tidy() |> mutate(names = as.numeric(names
  by = c("CustomerID" = "names") )|>
  rename(clusters = x)

mall_customers |> ggplot(aes(x = `Annual Income (k$)`,y=`Spending Score (1-
  geom_point()  +
  scale_color_okabe_ito() +
  scale_fill_okabe_ito()  +
  theme_minimal(base_size = 16) +
  labs(title = "Cluster Differences")
```

Try a Different Linkage

- Use the Ward Linkage Instead

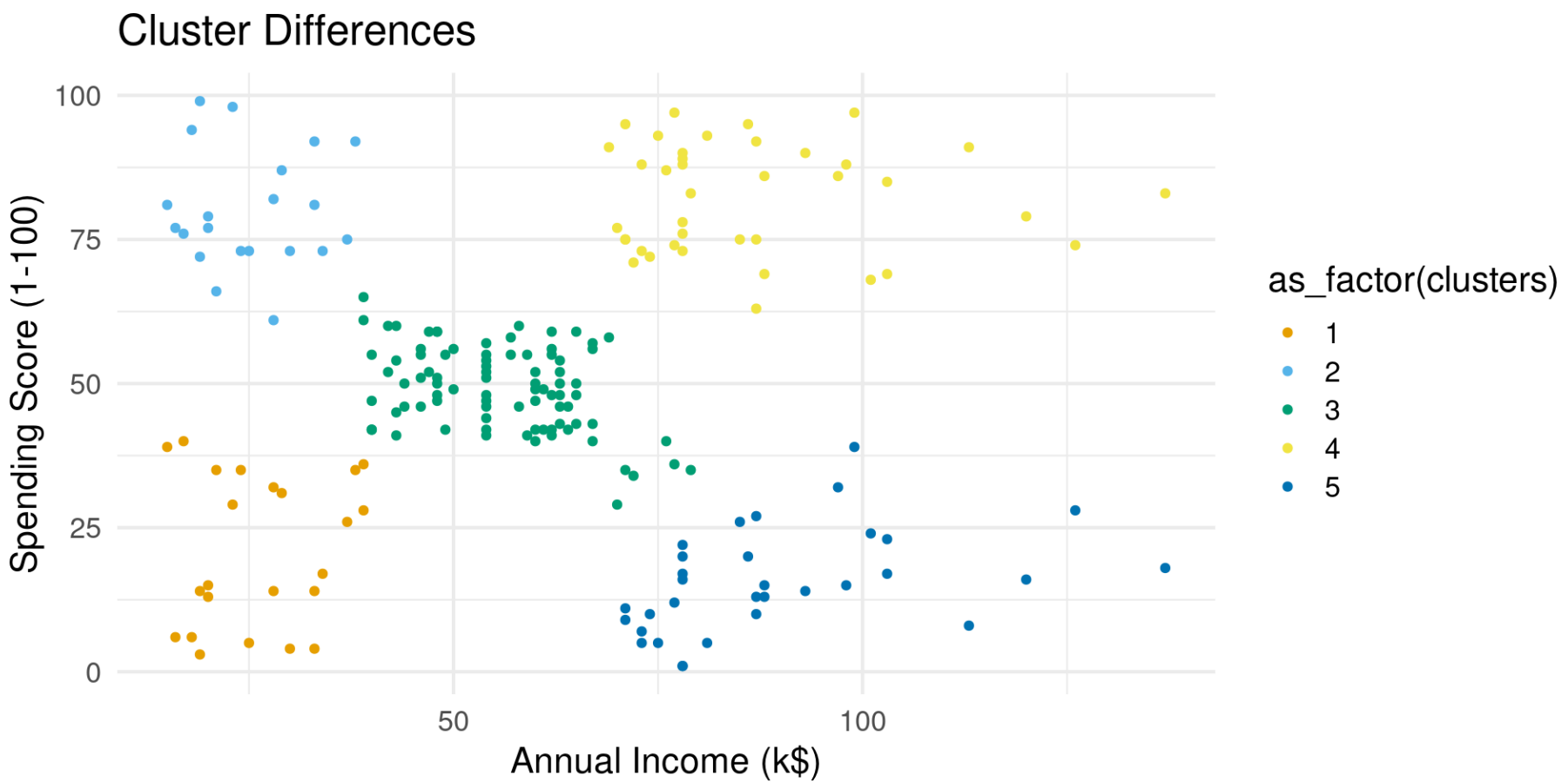
```
mall_clusters = mall_dist |>
  hclust(
  method = "ward"
)

mall_customers = mall_customers |> left_join(
  mall_clusters |> cutree(k=5) |> tidy() |> mutate(names = as.numeric(names
  by = c("CustomerID" = "names") )|>
  rename(clusters = x)

mall_customers |> ggplot(aes(x = `Annual Income (k$)`,y=`Spending Score (1-
  geom_point()  +
  scale_color_okabe_ito() +
  scale_fill_okabe_ito()  +
  theme_minimal(base_size = 16) +
  labs(title = "Cluster Differences")
```

Try a Different Linkage

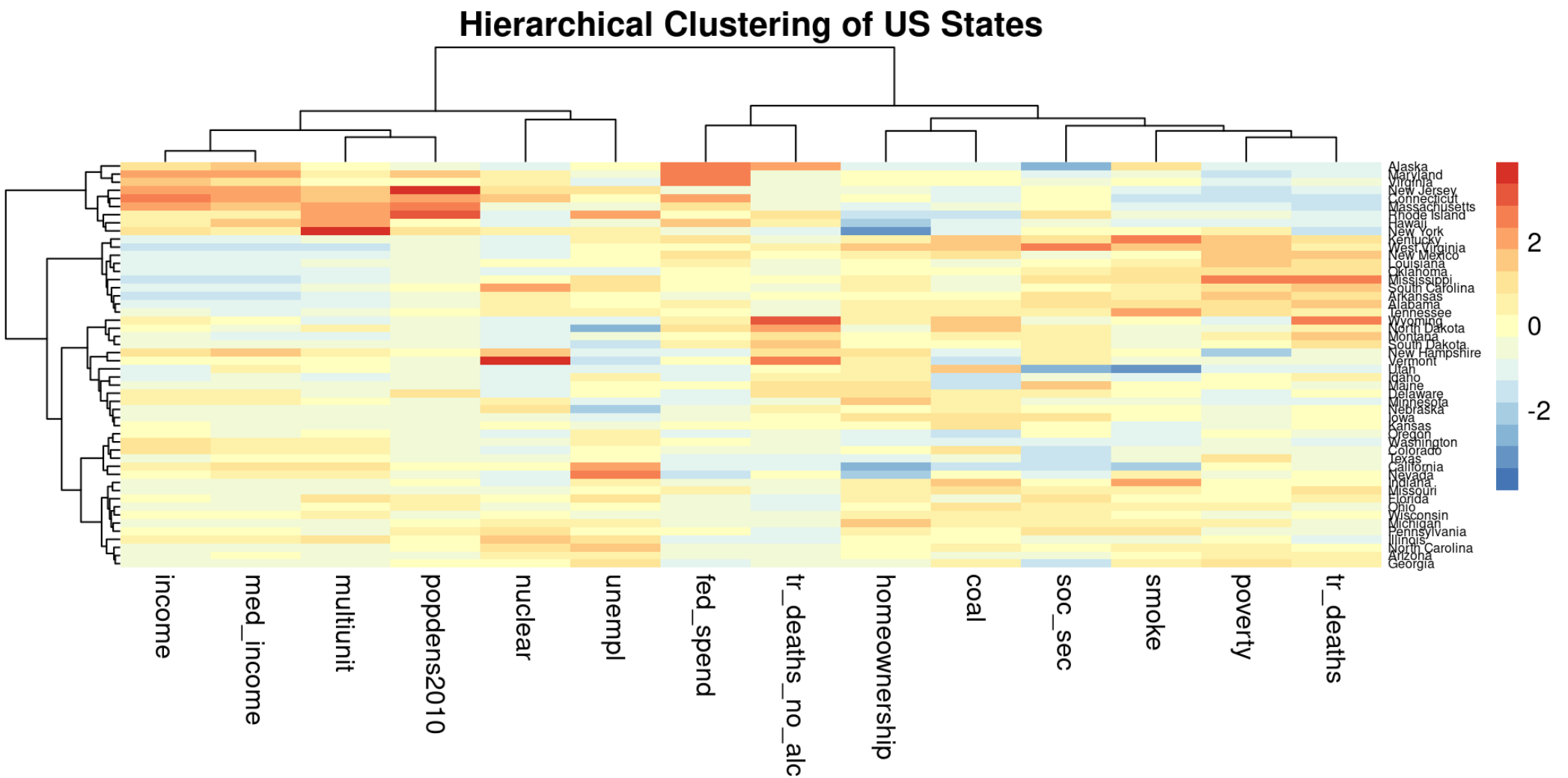
- Use the Ward Linkage Instead



Interpreting Clusters

- Similar procedure as to k-means
  - Summarize/average over the clusters
  - Plot distributions over the clusters
- Take advantage of Hierarchical Structure using Heatmaps

Clustered Heatmaps



Clustered Heatmaps

- `tidyheatmaps` package
  - `pheatmap` frontend
  - Others have similar names
- Not `ggplot2` compatible

Clustered Heatmaps

- Need long-format data

```
  1  US_state_stats
```

```
# A tibble: 50 × 15
   state       homeownership multiunit income med_income poverty fed_spend smoke
   <chr>               <dbl>     <dbl>  <dbl>      <dbl>   <dbl>     <dbl> <dbl>
 1 Alabama              71.1      15.5  22984      42081    17.1      11.7   24.8
 2 Alaska               64.7      24.6  30726      66521     9.5      16.8   25
 3 Arizona              67.4      20.7  25680      50448    15.3       9.85  20.4
 4 Arkansas             67.7      15.2  21274      39267    18         9.61  23.5
 5 California           57.4      30.7  29188      60883    13.7       8.89  15.2
 6 Colorado             67.6      25.6  30151      56456    12.2       9.15  19.9
 7 Connecticut          69.2      34.6  36775      67740     9.2      14.8   16.5
 8 Delaware             73.6      17.7  29007      57599    11         8.89  20.7
 9 Florida              69.7      30    26551      47661    13.8       9.62  21.6
10 Georgia              67.2      20.5  25134      49347    15.7       8.88  22.2
# ℹ 40 more rows
# ℹ 7 more variables: soc_sec <dbl>, nuclear <dbl>, coal <dbl>,
#   tr_deaths <dbl>, tr_deaths_no_alc <dbl>, unempl <dbl>, popdens2010 <dbl>
```

Clustered Heatmaps

- Need long-format data

```
US_long = US_state_stats |> pivot_longer(
  names_to = "variable",
  cols=homeownership:popdens2010,
  values_to = "value")
US_long
```

```
# A tibble: 700 × 3
   state   variable        value
   <chr>   <chr>           <dbl>
 1 Alabama homeownership    71.1
 2 Alabama multiunit        15.5
 3 Alabama income        22984
 4 Alabama med_income    42081
 5 Alabama poverty          17.1
 6 Alabama fed_spend        11.7
 7 Alabama smoke            24.8
 8 Alabama soc_sec          18.5
 9 Alabama nuclear          16.3
10 Alabama coal             56.6
# ℹ 690 more rows
```

Clustered Heatmaps

- `tidy_heatmap` computes clusters for you, but takes a ton of arguments

```
library(tidyheatmaps)

US_long |>
  tidy_heatmap(
                rows = state,
                columns = variable,
                values = value,
                cluster_rows = TRUE,
                cluster_cols = TRUE,
                scale = "column",
                fontsize_row = 6,
                fontsize_col=12,
                clustering_method = "ward",
                main = "Hierarchical Clustering of US States",
                fontsize = 12)
```

Clustered Heatmaps

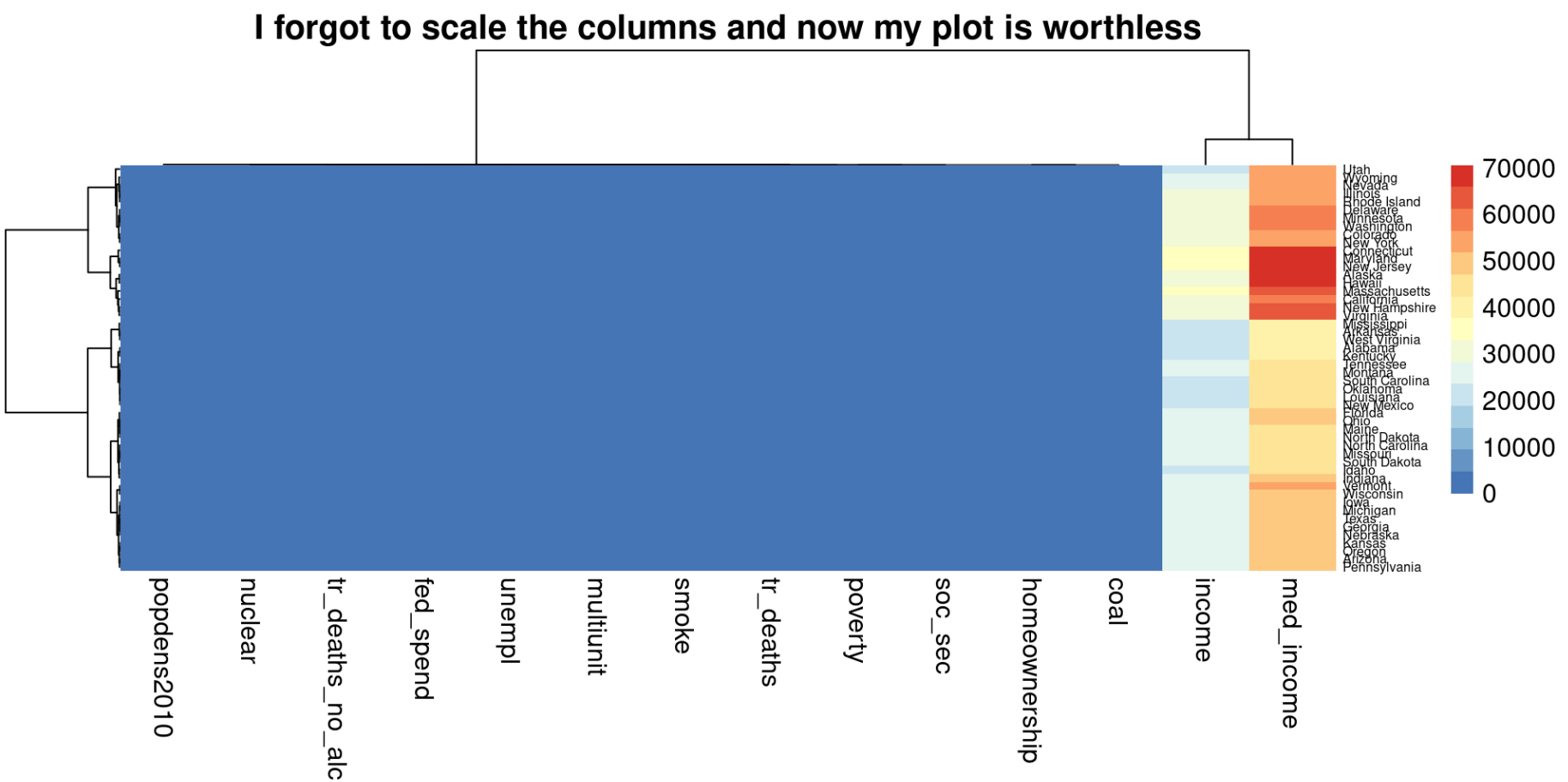
- `scale = column/row is very important`

```
library(tidyheatmaps)

US_long |>
  tidy_heatmap(
                rows = state,
                columns = variable,
                values = value,
                cluster_rows = TRUE,
                cluster_cols = TRUE,
                scale = "column",
                fontsize_row = 6,
                fontsize_col=12,
                clustering_method = "ward",
                main = "Hierarchical Clustering of US States",
                fontsize = 12)
```

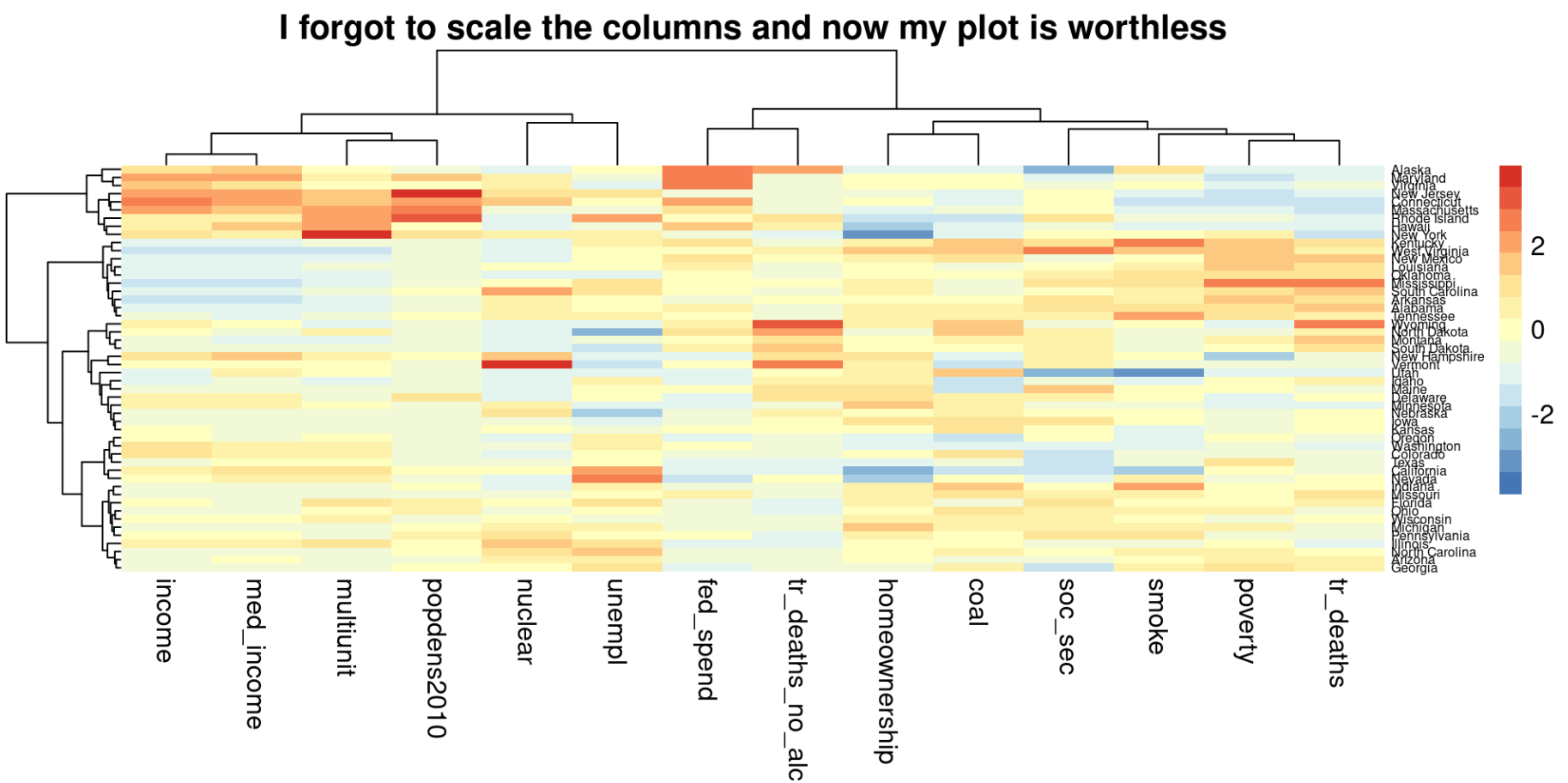
Clustered Heatmaps

- `scale = column/row is very important`



Clustered Heatmaps

- `scale = column/row is very important`



Annotations

- Say we want to annotate the clusters with a categorical variable
- Add US-Regions to data frame

```
US_state_aug = US_state_stats |>
  left_join(US_regions) |>
  mutate(state = state_abr) |>
  select(-state_abr) |>
  pivot_longer(
    names_to = "variable",
    cols=homeownership:popdens2010,
    values_to = "value")
```

Annotations

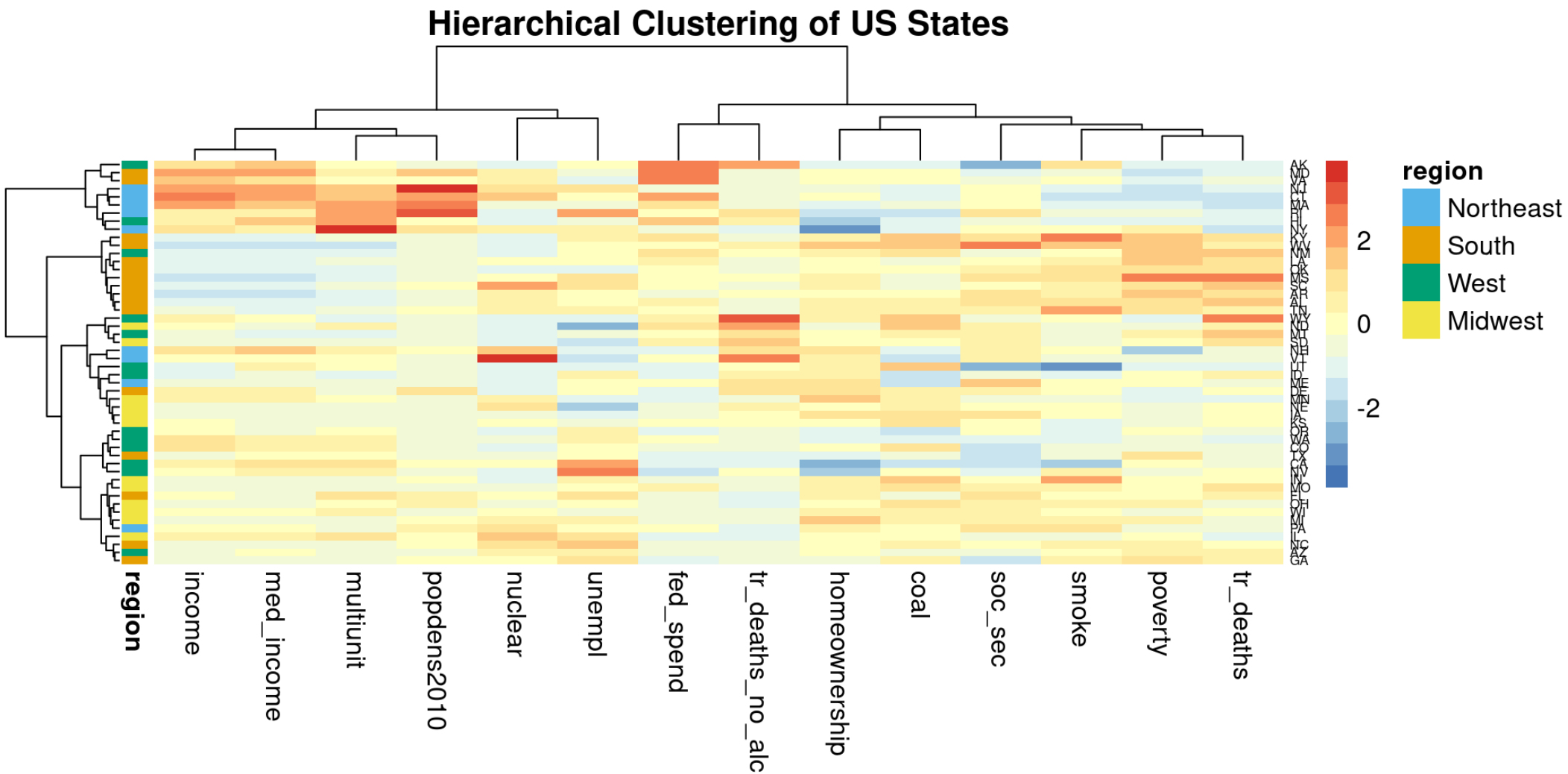
- Say we want to annotate the clusters with a categorical variable
- Add US-Regions to data frame

```
# A tibble: 700 × 5
   state region division           variable        value
   <chr> <chr>  <chr>              <chr>           <dbl>
 1 AL    South  East South Central homeownership    71.1
 2 AL    South  East South Central multiunit        15.5
 3 AL    South  East South Central income         22984
 4 AL    South  East South Central med_income     42081
 5 AL    South  East South Central poverty          17.1
 6 AL    South  East South Central fed_spend        11.7
 7 AL    South  East South Central smoke            24.8
 8 AL    South  East South Central soc_sec         18.5
 9 AL    South  East South Central nuclear         16.3
10 AL    South  East South Central coal            56.6
# ℹ 690 more rows
```

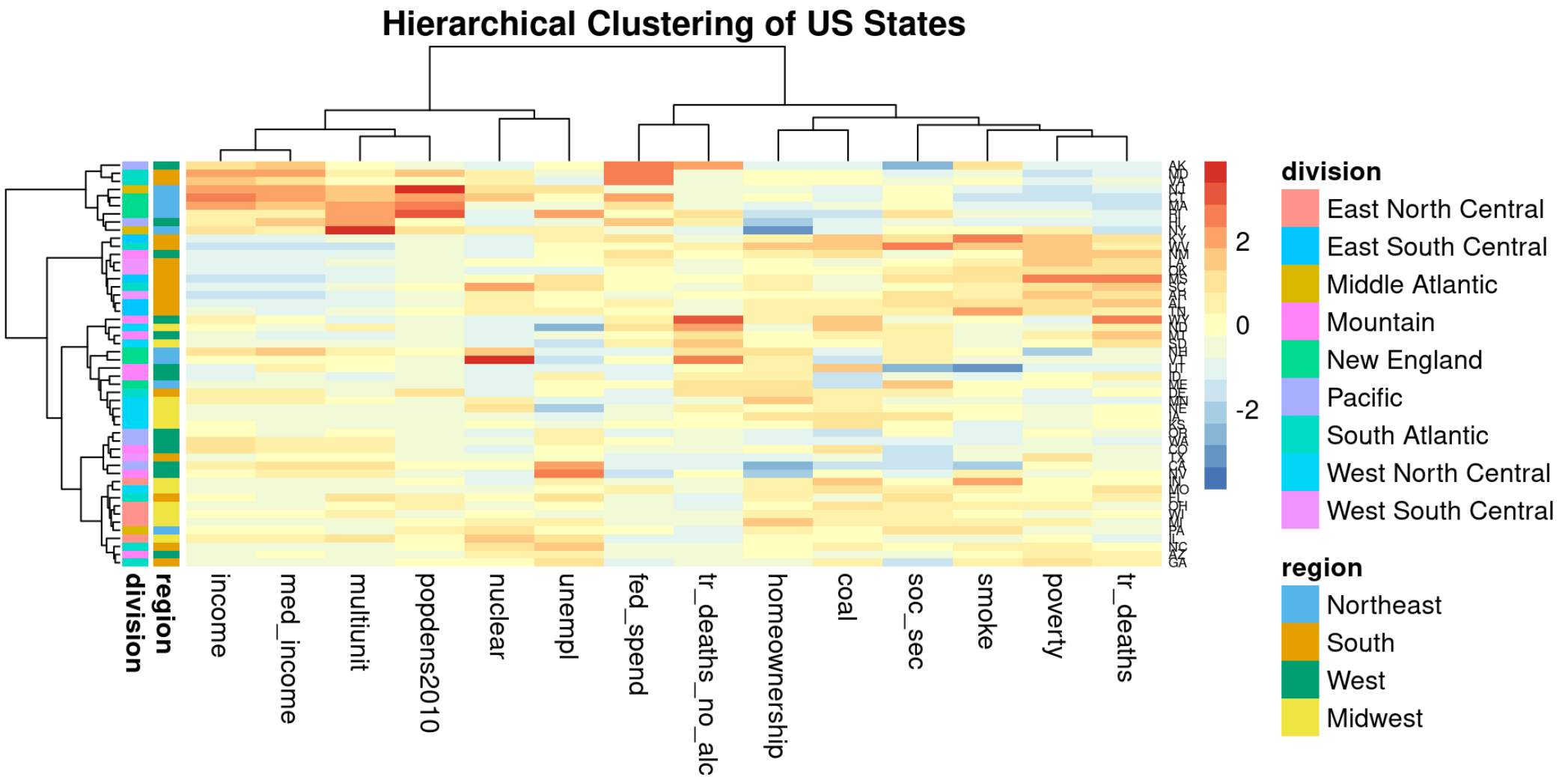
Annotations

```
US_state_aug |>
  tidy_heatmap(
                rows = state,
                columns = variable,
                values = value,
                cluster_rows = TRUE,
                cluster_cols = TRUE,
                scale = "column",
                fontsize_row = 6,
                fontsize_col=12,
                clustering_method = "ward",
                main = "Hierarchical Clustering of US States",
                fontsize = 12,
                annotation_row = region,
                annotation_colors =  list(  # specify colors for annotation
      region = c("Northeast" = "#56B4E9", "South" = "#E69F00", "West" = "#0
    ))
```

Annotations



Multiple Annotations



Downsides of Hierarchical Clustering

- Computationally very challenging
  - Naive methods scale like $O(n^3)$
  - $O(n^2)$ for memory
  - Need to become a pro….
- Bad with many features/high dimensions
- Bad if data is nonlinearly shaped

Thanks!