STAT 209
Dimensionality Reduction

August 12, 2021

Colin Reimer Dawson

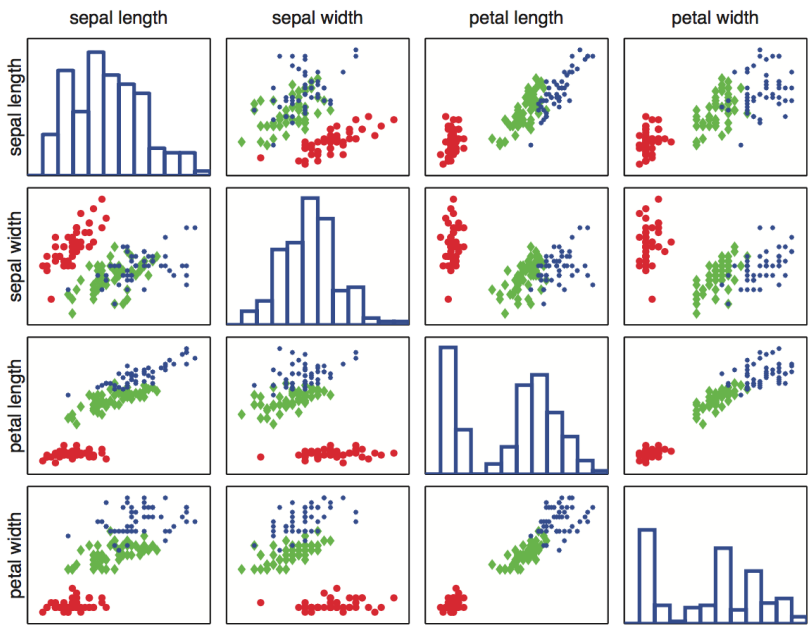
Outline

High Dimensional Data

- Modern datasets often have huge numbers of variables
- E.g., images, biomarker data, measurements at fine-grained time points, social networks, product preferences
- Clustering can be a useful way to find "groups" of similar observations
- However, distance measures have some strange properties in high dimensions
- Can be useful to try to extract a few dimensions that carry most of the "signal"

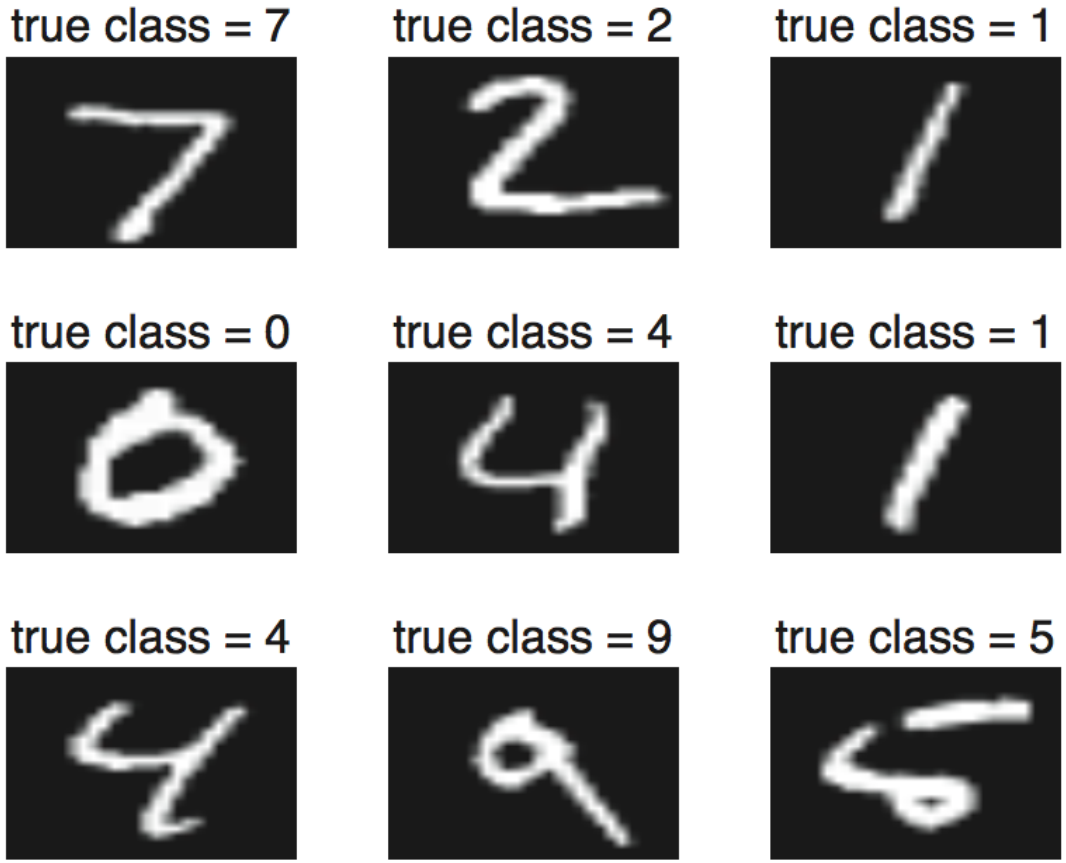
Images Have Many Variables...

(a) (b) (c)

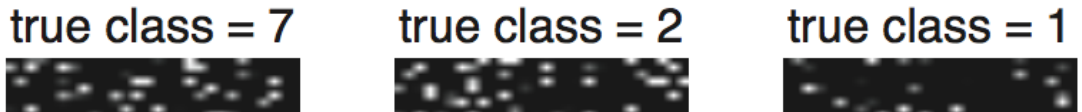


but maybe only a few meaningful "features"

High dimensional inputs



Comprehensible arranged this way...



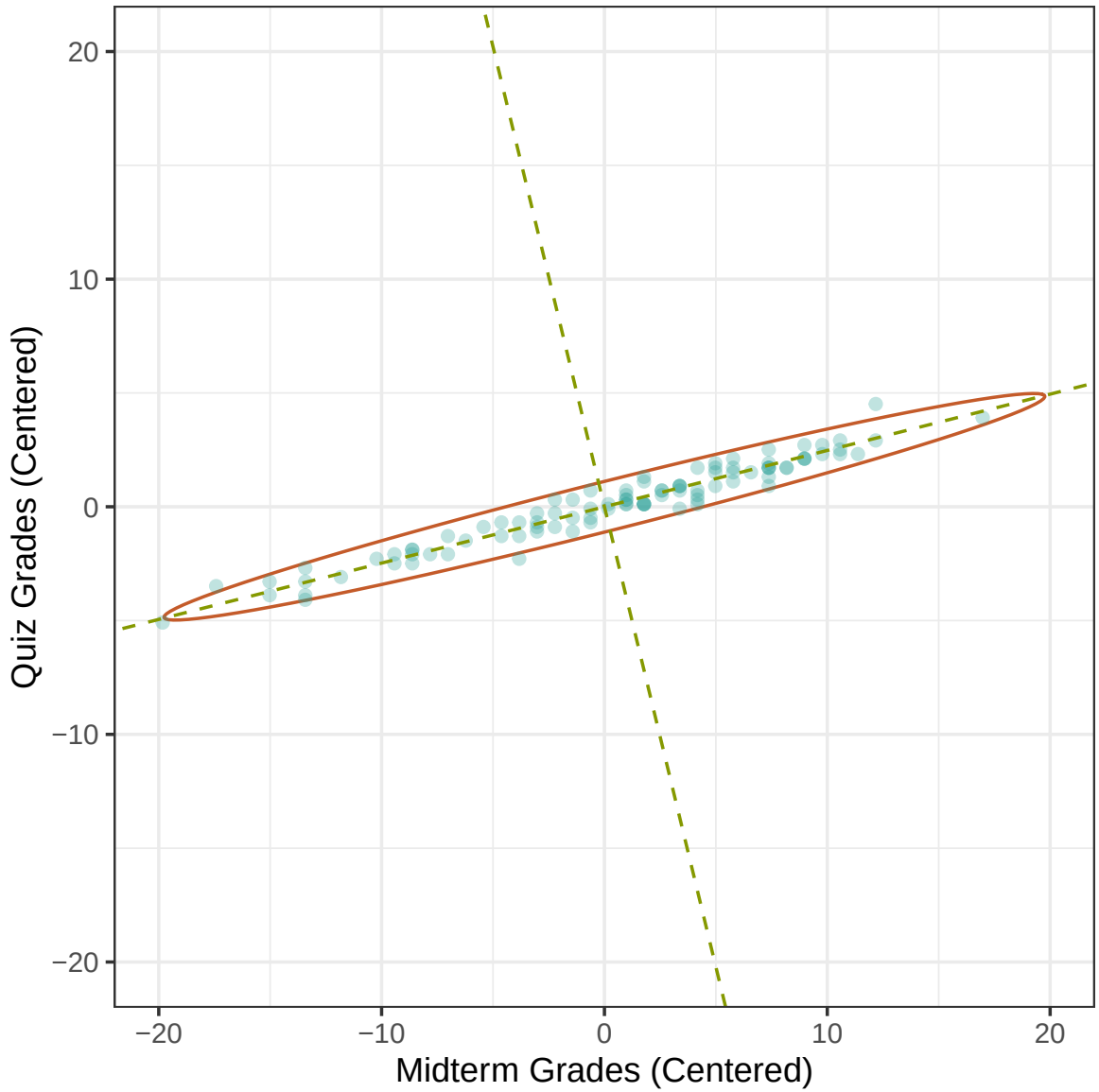
"Eigenfaces"

(a)

(b)

The faces on the left are weighted combinations of the images on the right

Finding the "Main Direction" of Variation



Finding the "Eigen-features"

Pulling out the perpendicular directions in (Midterm,Quiz) space that align with the ellipse on the scatterplot. **Linear algebra fun fact**: These are the eigenvectors of the covariance matrix

```
directions <- Scores %>%
    select(Midterm, Quiz) %>%
    cov() %>%  # find the covariance matrix
    eigen()    # compute its eigenvalues/vectors

directions %>%
    pluck("vectors") %>%
    round(digits = 2)
```

```
      [,1]  [,2]
[1,] -0.97  0.24
[2,] -0.24 -0.97
```

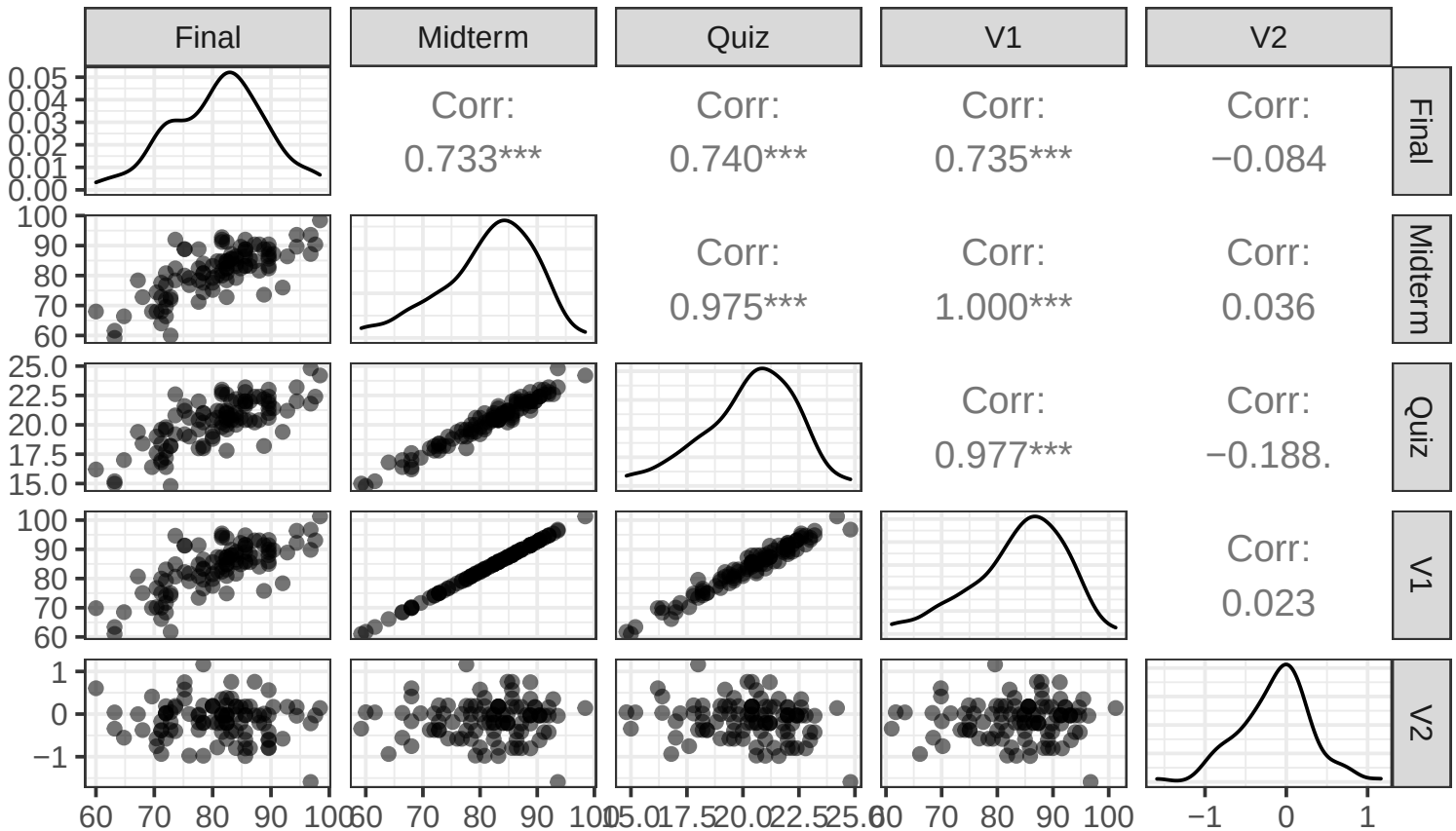
Creating "Eigenscores"

Here we creating two new variables that are a weighted sum and weighted difference of the midterm and quiz score. Weights are chosen based on the eigenvectors so that the new variables are uncorrelated:

```
Scores_augmented <- Scores %>%
  mutate(
    V1 = 0.97 * Midterm + 0.24 * Quiz,
    V2 = 0.24 * Midterm - 0.97 * Quiz)
```

Scatterplots With Raw and "Eigen" Scores

```
library(GGally)
Scores_augmented %>%
    select(Final, Midterm, Quiz, V1, V2) %>%
    ggpairs(aes(alpha = 0.15))
```



Scottish Parliament Votes

> [C]onsider the Scottish Parliament in 2008. Legislators often vote together in pre-organized blocs, and thus the pattern of "ayes" and "nays" on particular ballots may indicate which members are affiliated (i.e., members of the same political party). To test this idea, you might try clustering the members by their voting record. – MDSR p. 212

Scottish Parliament Votes

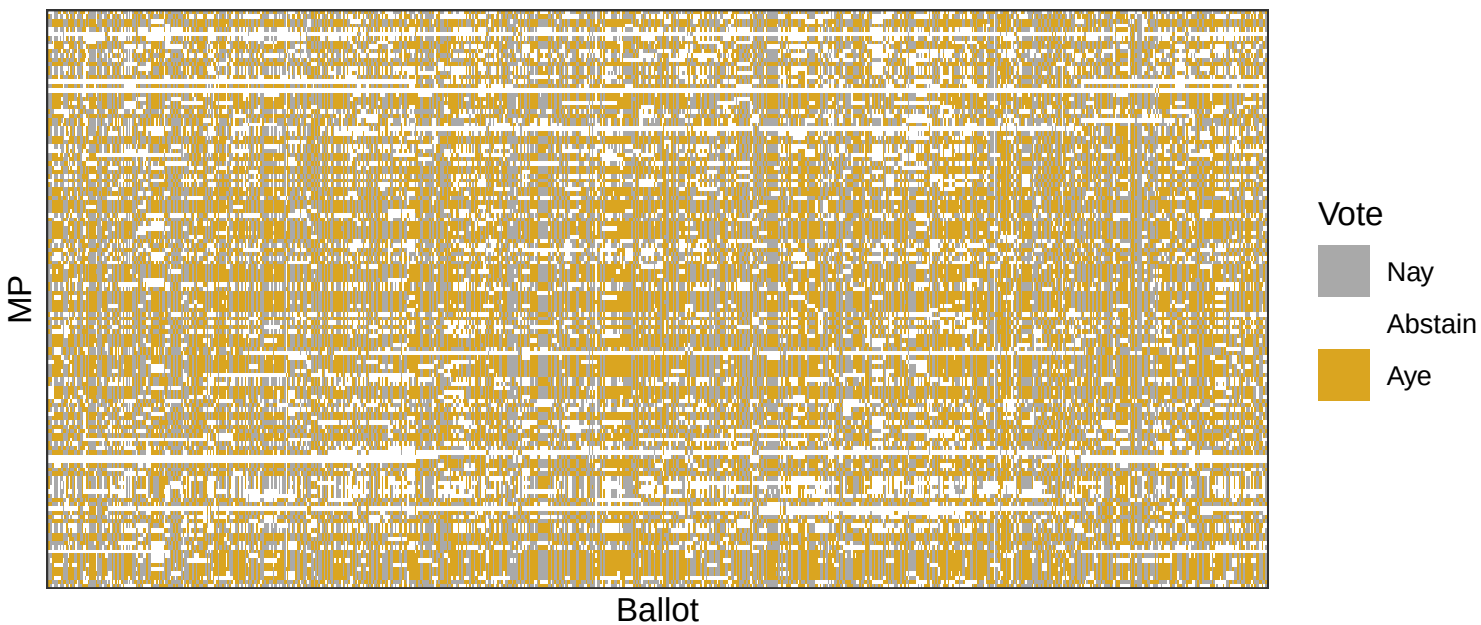
name	S1M-1	S1M-1007.1	S1M-1007.2	S1M-1008
Adam, Brian	1	1	-1	0
Aitken, Bill	1	1	1	-1
Alexander, Ms Wendy	1	-1	-1	1
Baillie, Jackie	1	-1	-1	1
Barrie, Scott	-1	-1	-1	1
Boyack, Sarah	0	-1	-1	1
Brankin, Rhona	0	-1	0	1
Brown, Robert	-1	-1	-1	1
Butler, Bill	0	0	0	0
Campbell, Colin	1	1	-1	0

Table 9.1: Sample voting records data from the Scottish Parliament.

Figure: Source: MDSR p. 212

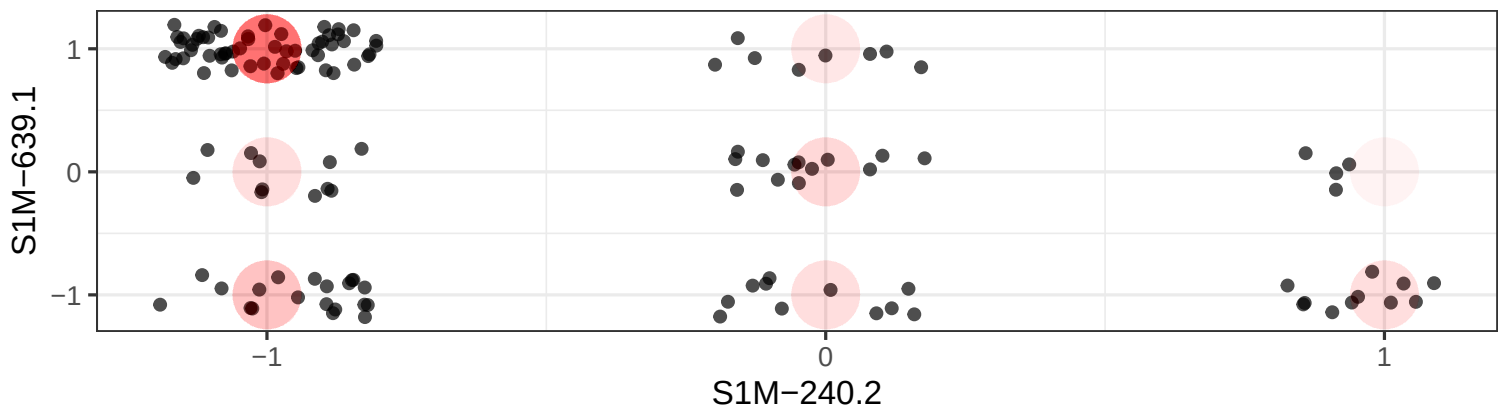
Visualizing All of the Votes

```
library(mdsr)
Votes %>%
    mutate(Vote = factor(vote, labels = c("Nay","Abstain","Aye"))) %>%
    ggplot(aes(x = bill, y = name, fill = Vote)) +
      geom_tile() +
      scale_fill_manual(values = c("darkgray", "white", "goldenrod")) +
      scale_x_discrete(name = "Ballot", breaks = NULL, labels = NULL) +
      scale_y_discrete(name = "MP", breaks = NULL, labels = NULL)
```



Visualizing Two Randomly Selected Votes

```
Votes %>%
    filter(bill %in% c("S1M-240.2", "S1M-639.1")) %>%
    pivot_wider(names_from = bill, values_from = vote) %>%
    ggplot(aes(x = `S1M-240.2`, y = `S1M-639.1`)) +
        geom_point(
            alpha    = 0.7,
            position = position_jitter(width = 0.2, height = 0.2)) +
        geom_point(alpha = 0.01, size = 10, color = "red" ) +
        scale_x_continuous(breaks = c(-1,0,1), labels = c(-1,0,1)) +
        scale_y_continuous(breaks = c(-1,0,1), labels = c(-1,0,1))
```



Are there eight clusters of MPs?

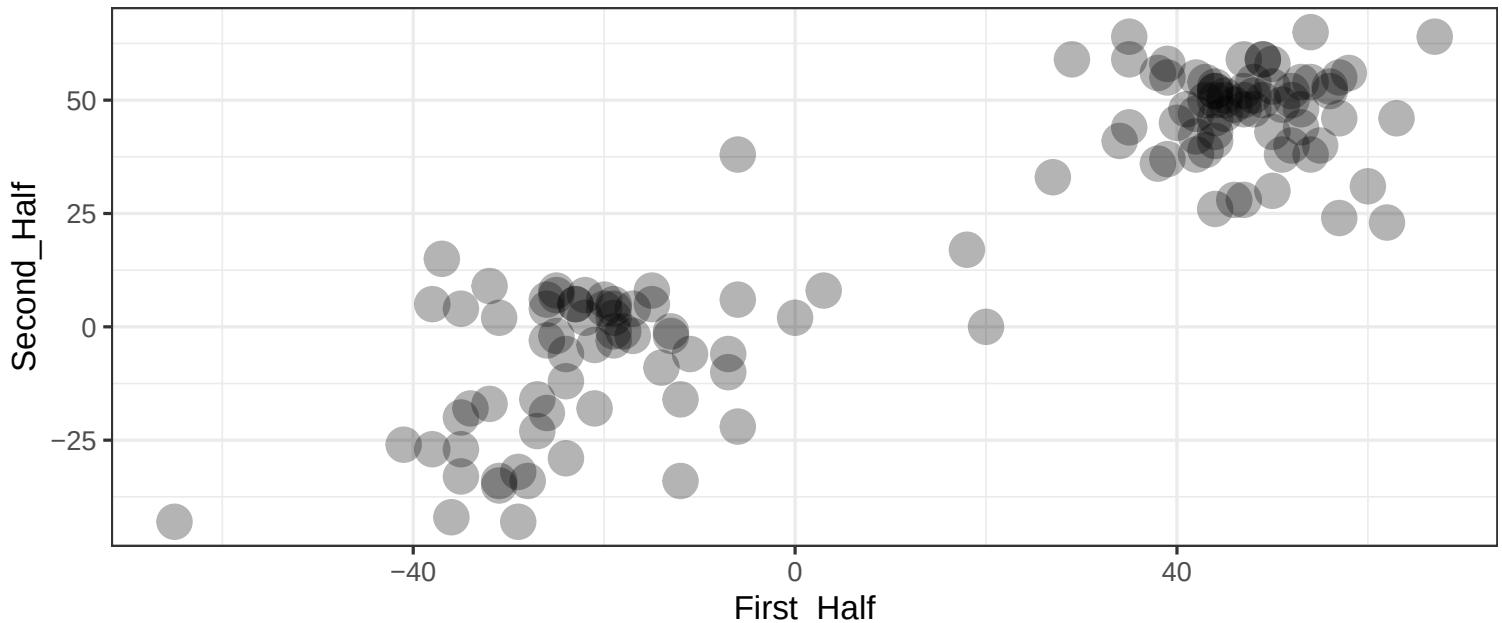
Two Arbitrary Aggregate Features

```
Votes_by_half <- Votes %>%
    mutate(
        set_num = bill %>% factor() %>% as.numeric(),
        set     = ifelse(
            set_num < median(set_num),
            "First_Half", "Second_Half")) %>%
    group_by(name, set) %>%
    summarise(Ayes = sum(vote)) %>%
    pivot_wider(names_from = set, values_from = Ayes)
Votes_by_half %>%
    head(n = 5)
```

```
# A tibble: 5 x 3
# Groups:   name [5]
  name                First_Half Second_Half
  <chr>                    <int>       <int>
1 Adam, Brian                -25          -2
2 Aitken, Bill               -32         -17
3 Alexander, Ms Wendy         35          59
4 Baillie, Jackie             43          50
5 Barrie, Scott               48          54
```

Visualizing these Features

```
Votes_by_half %>%
    ggplot(aes(x = First_Half, y = Second_Half)) +
    geom_point(alpha = 0.3, size = 5)
```



Maybe Two Clusters?

A More Principled Approach

- Instead of arbitrarily splitting the votes into "first half" and "second half", we can extract some "high signal" aggregate features using linear algebra
- **Singular Value Decomposition** (SVD) takes a matrix and finds linear combinations of columns (variables) that account for a high degree of variability in the observations

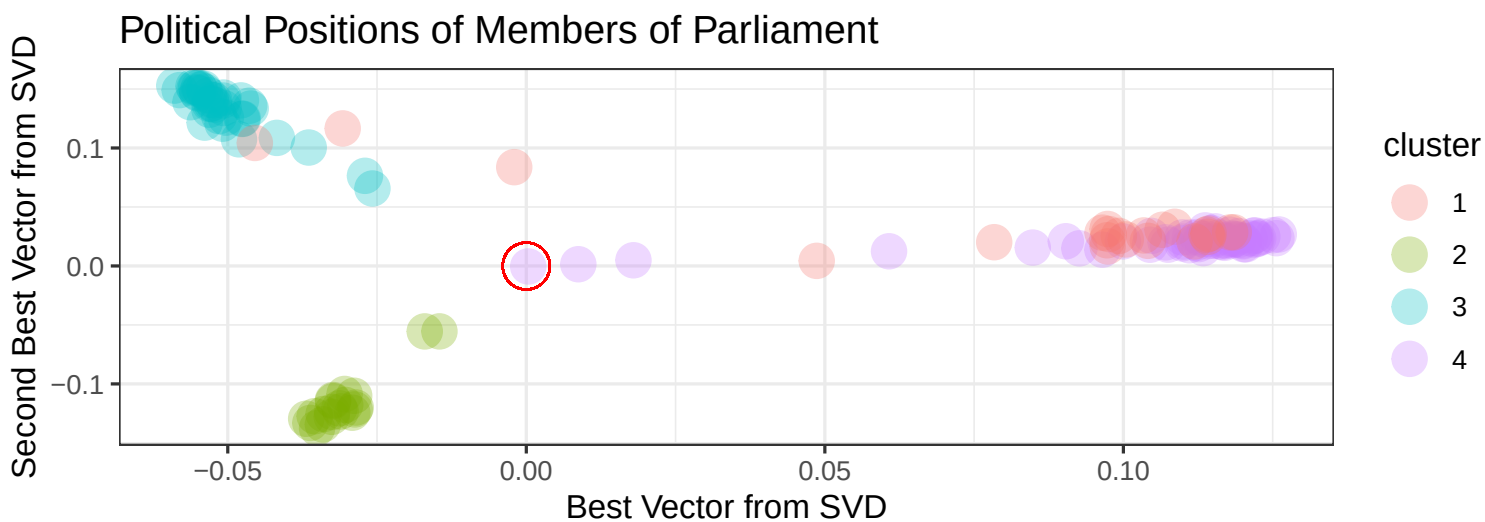
Finding the Singular Value Decomposition

```
Votes_wide <- Votes %>%
    pivot_wider(names_from = bill, values_from = vote)
Vote_SVD <- Votes_wide %>%
    select(-name) %>%
    svd()
MPs_SVD <- Vote_SVD %>%
    pluck("u") %>%
    as_tibble() %>%
    select(1:5)
Votes_wide %>%
    select(name) %>%
    bind_cols(MPs_SVD) %>%
    head(n = 10)
```

```
# A tibble: 10 x 6
   name                          V1      V2     V3       V4       V5
   <chr>                      <dbl>   <dbl>  <dbl>    <dbl>    <dbl>
 1 Canavan, Dennis          -0.0307  0.117  0.0266 -0.302    0.280
 2 Aitken, Bill             -0.0341 -0.134  0.212  -0.00495  0.00737
 3 Davidson, Mr David       -0.0324 -0.115  0.187  -0.0341  -0.0333
 4 Douglas Hamilton, Lord James -0.0350 -0.137  0.216  -0.00276 -0.00291
 5 Fergusson, Alex          -0.0325 -0.124  0.202  -0.0117   0.00612
```

Clusters in 5D SVD space

```
clusters <- MPs_SVD %>%
    kmeans(centers = 4, nstart = 100)
MPs_SVD <- MPs_SVD %>%
    mutate(cluster = clusters %>% pluck("cluster") %>% factor())
MPs_SVD %>%
    ggplot(aes(x = V1, y = V2)) +
    geom_point(aes(x = 0, y = 0), color = "red", shape = 1, size = 7) +
    geom_point(aes(color = cluster), size = 5, alpha = 0.3) +
    xlab("Best Vector from SVD") + ylab("Second Best Vector from SVD") +
    ggtitle("Political Positions of Members of Parliament")
```



Do Clusters Align With Party?

```
library(mosaic)
MPs_SVD <- MPs_SVD %>%
    mutate(name = pull(Votes_wide, name)) %>%
    left_join(Parties, by = "name")
MPs_SVD %>% tally(party ~ cluster, data = .)
```

	cluster			
party	1	2	3	4
Member for Falkirk West	1	0	0	0
Scottish Conservative and Unionist Party	0	20	0	0
Scottish Green Party	1	0	0	0
Scottish Labour	3	0	0	55
Scottish Liberal Democrats	16	0	0	1
Scottish National Party	0	0	36	0
Scottish Socialist Party	1	0	0	0

- Clusters contain natural political coalitions:
  - Conservatives, Labour, SNP, Misc. Left-leaning Parties
- Note that neither SVD or K-means had access to party labels

Ballots as Cases

```
Ballots_SVD <- Vote_SVD %>%
    pluck("v") %>%
    as_tibble() %>%
    select(1:5)
Votes %>%
    select(bill) %>%
    distinct() %>%
    bind_cols(Ballots_SVD) %>%
    mutate(across(-bill, round, 3)) %>%
    head(n = 5)
```

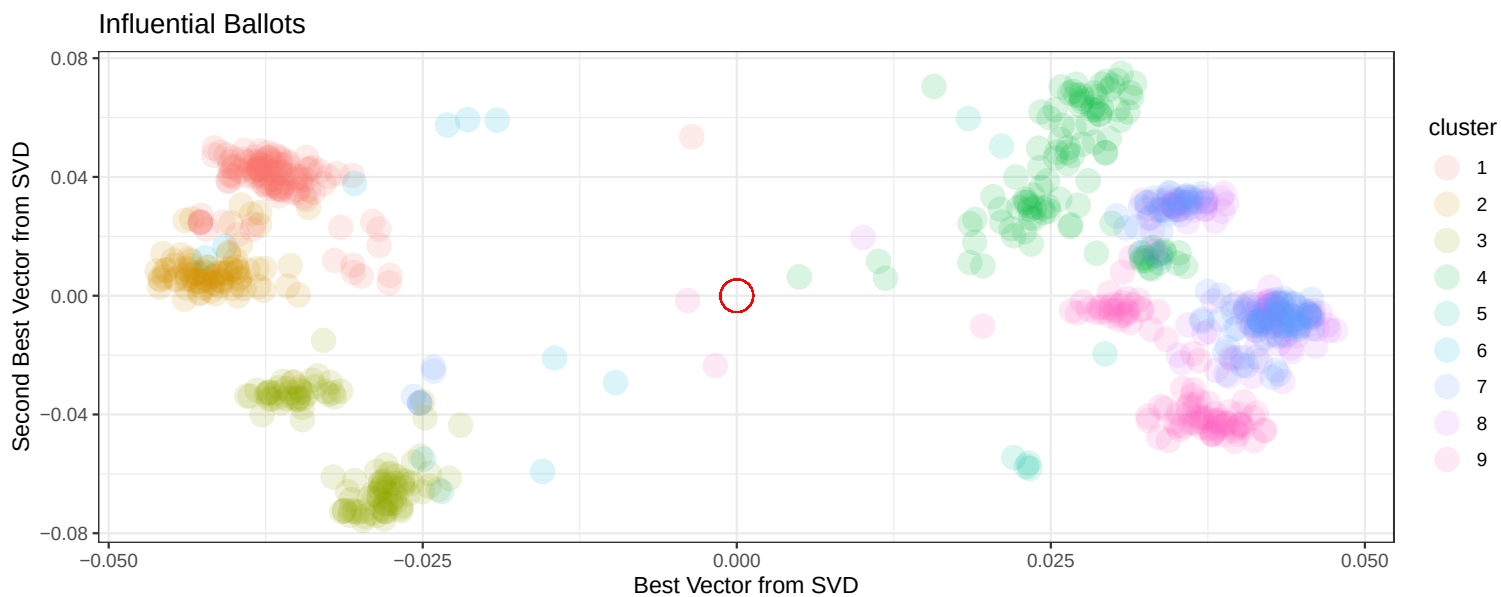
	bill	V1	V2	V3	V4	V5
1	S1M-1	-0.004	-0.002	0.050	-0.073	0.014
2	S1M-4.1	-0.046	0.008	0.041	-0.016	0.071
3	S1M-4.3	-0.043	0.026	0.004	0.003	0.056
4	S1M-4	0.044	-0.019	-0.015	0.001	-0.065
5	S1M-5	0.044	-0.018	-0.021	-0.009	-0.057

Clustering Ballots

```
Ballot_clusters <- Ballots_SVD %>%
    kmeans(centers = 9, nstart = 1000)
Ballots_SVD <- Ballots_SVD %>%
    mutate(
       bill    = Votes %>% pull("bill") %>% levels(),
       cluster = Ballot_clusters %>% pluck("cluster") %>% factor())
```

Clusters of Ballots

```
Ballots_SVD %>%
    ggplot(aes(x = V1, y = V2)) +
        geom_point(aes(x = 0, y = 0), color = "red", shape = 1, size = 7) +
        geom_point(size = 5, alpha = 0.15, aes(color = cluster)) +
        xlab("Best Vector from SVD") +
        ylab("Second Best Vector from SVD") +
        ggtitle("Influential Ballots")
```



Reconstructing Votes by Ballot

```
Votes_SVD <- Votes %>%
    mutate(Vote = factor(vote, labels = c("Nay", "Abstain", "Aye"))) %>%
    inner_join(Ballots_SVD, by = "bill") %>%
    inner_join(MPs_SVD, by = "name", suffix = c("_ballot", "_MP"))
Votes_SVD %>%
    select(-vote) %>%
    mutate(across(V1_ballot:V5_ballot, round, 3)) %>%
    mutate(across(V1_MP:V5_MP, round, 3)) %>%
    slice_sample(n = 5)
```

```
         bill             name    Vote V1_ballot V2_ballot V3_ballot
1 S1M-3879.1 Steel, Sir David Abstain    -0.036    -0.033     0.010
2  S1M-461.1    Paterson, Gil     Aye    -0.042     0.010     0.040
3   S1M-1555      Scott, John     Aye     0.040    -0.003    -0.035
4  S1M-131.1 Macdonald, Lewis     Nay     0.027    -0.006     0.057
5 S1M-3477.2   Murray, Elaine     Nay     0.027     0.067     0.015
  V4_ballot V5_ballot cluster_ballot  V1_MP  V2_MP V3_MP  V4_MP  V5_MP
1     0.026    -0.042              3  0.000  0.000 0.001 -0.002 -0.001
2    -0.037    -0.035              2 -0.054  0.148 0.091  0.014 -0.013
3     0.025    -0.048              8 -0.029 -0.110 0.180  0.028 -0.046
4    -0.025    -0.031              9  0.119  0.023 0.037  0.078  0.026
5    -0.013     0.021              4  0.115  0.021 0.033  0.026  0.067
```

All Members/Votes, Sorted by 1st SVD Feature

```
Votes_SVD %>%
    ggplot(aes(x = reorder(bill, V1_ballot), y = reorder(name, V1_MP),
               fill = Vote)) +
    geom_tile() + xlab("Ballot") + ylab("Member of Parliament") +
    scale_fill_manual(values = c("darkgray", "white", "goldenrod")) +
    scale_x_discrete(breaks = NULL, labels = NULL) +
    scale_y_discrete(breaks = NULL, labels = NULL)
```