# Problem Set 10

## Clustering

### [YOUR NAME]

### Due Date: 2024-04-05

Open `RStudio` and create a new RMarkDown file ( `.Rmd` ) by going to `File -> New File -> R Markdown...`. Accept defaults and save this file as `[LAST NAME]_ps10.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[LAST NAME]_ps10.Rmd` file. Then change the `author: [Your Name]` to your name.

We will be using the `pres_elec.rds` file from the course github page (https://github.com/jbisbee1/DS1000_S2024/raw/main/data/pres_elec.rds).

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is worth 8 total points, plus two extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must have the correct code. In addition, some questions ask you to provide a written response in addition to the code.

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, compiled the completed problem set and upload the PDF file to Brightspace on Friday by midnight. Also note that the TAs and professors will not respond to Campuswire posts after 5PM on Friday, so don't wait until the last minute to get started!

**Good luck!**

*Copy the link to ChatGPT you used here: _____________________

# Question 0

Require `tidyverse` and `tidymodels` , and then load the `pres_elec.rds` (https://github.com/jbisbee1/DS1000_S2024/raw/main/data/pres_elec.rds) data to an object called `dat` .

# Question 1 [2 points]

Describe the data. What is the unit of analysis? What information do the columns provide? What is the period described (i.e., how far back in time does the data go?). Is there any missing data? If so, "where" is it, in terms of both columns and in terms of the observations that have missing data?

```
glimpse(dat)
```

```
## Rows: 40,451
## Columns: 11
## $ year        <dbl> 2008, 2012, 1992, 1988, 1996, 2000, 2004, 2016, 1984, 197…
## $ county      <chr> "abbeville", "abbeville", "abbeville", "abbeville", "abbe…
## $ state_abb   <chr> "SC", "SC", "SC", "SC", "SC", "SC", "SC", "SC", "SC", "SC…
## $ TotalVotes  <dbl> 11001, 10671, 8343, 7401, 7100, 8374, 9925, 10775, 6875, …
## $ RepVotes    <dbl> 6264, 5981, 3317, 3738, 3054, 4450, 5436, 6763, 3798, 326…
## $ DemVotes    <dbl> 4593, 4543, 3968, 3629, 3493, 3766, 4389, 3741, 3051, 134…
## $ RepCandidate <chr> "McCain, John S. III", "Romney, W. Mitt", "Bush, George H…
## $ RepStatus   <chr> "Challenger", "Challenger", "Incumbent", "Challenger", "C…
## $ DemCandidate <chr> "Obama, Barack H.", "Obama, Barack H.", "Clinton, Bill", …
## $ DemStatus   <chr> "Challenger", "Incumbent", "Challenger", "Challenger", "I…
## $ GOP_win     <dbl> 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, …
```

```
summary(dat)
```

```
##       year         county           state_abb          TotalVotes
##  Min.   :1972   Length:40451       Length:40451       Min.   :      0
##  1st Qu.:1984   Class :character   Class :character   1st Qu.:   4420
##  Median :1996   Mode  :character   Mode  :character   Median :   9349
##  Mean   :1996                                         Mean   :  34776
##  3rd Qu.:2008                                         3rd Qu.:  23086
##  Max.   :2020                                         Max.   :4264365
##
##     RepVotes          DemVotes       RepCandidate        RepStatus
##  Min.   :     31   Min.   :      4   Length:40451       Length:40451
##  1st Qu.:   2392   1st Qu.:   1588   Class :character   Class :character
##  Median :   5184   Median :   3623   Mode  :character   Mode  :character
##  Mean   :  16816   Mean   :  16450
##  3rd Qu.:  13020   3rd Qu.:   8961
##  Max.   :1549717   Max.   :3028885
##  NA's   :2         NA's   :2
##  DemCandidate        DemStatus            GOP_win
##  Length:40451       Length:40451       Min.   :0.0000
##  Class :character   Class :character   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :1.0000
##                                        Mean   :0.7335
##                                        3rd Qu.:1.0000
##                                        Max.   :1.0000
##                                        NA's   :2
```

```
dat %>%
  count(year)
```

```
## # A tibble: 13 × 2
##     year     n
##    <dbl> <int>
##  1  1972  3108
##  2  1976  3111
##  3  1980  3111
##  4  1984  3113
##  5  1988  3113
##  6  1992  3113
##  7  1996  3112
##  8  2000  3112
##  9  2004  3112
## 10  2008  3112
## 11  2012  3112
## 12  2016  3111
## 13  2020  3111
```

```
dat %>%
  filter(is.na(RepVotes))
```

```
## # A tibble: 2 × 11
##    year county     state_abb TotalVotes RepVotes DemVotes RepCandidate RepStatus
##   <dbl> <chr>      <chr>          <dbl>    <dbl>    <dbl> <chr>        <chr>
## 1  2016 bedford c… VA                 0       NA       NA Trump, Dona… Challeng…
## 2  2020 norfolk    VA                 0       NA       NA Trump, Dona… Incumbent
## # ℹ 3 more variables: DemCandidate <chr>, DemStatus <chr>, GOP_win <dbl>
```

```
dat %>%
  filter(is.na(DemVotes))
```

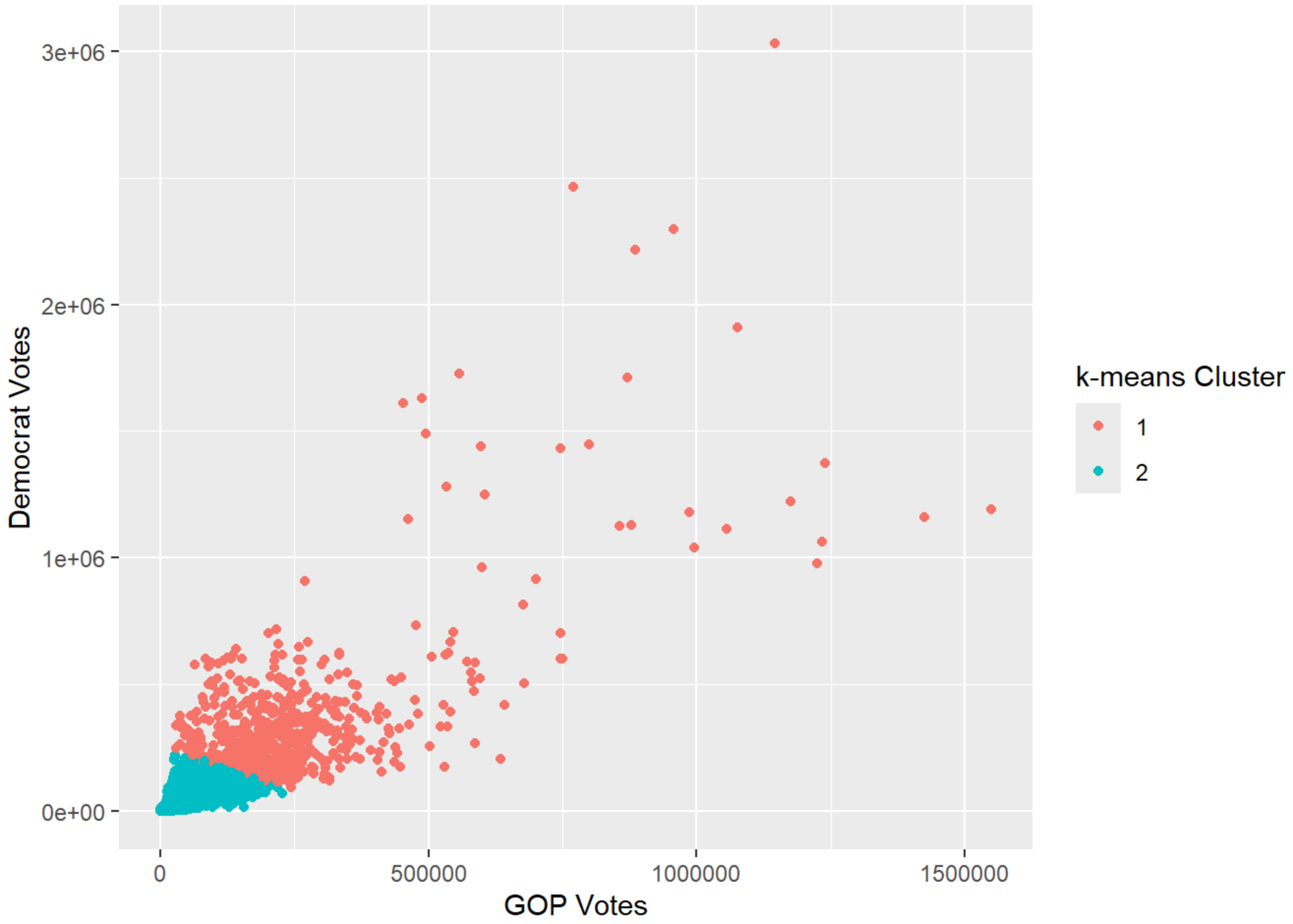
```
## # A tibble: 2 × 11
##    year county     state_abb TotalVotes RepVotes DemVotes RepCandidate RepStatus
##   <dbl> <chr>      <chr>          <dbl>    <dbl>    <dbl> <chr>        <chr>
## 1  2016 bedford c… VA                 0       NA       NA Trump, Dona… Challeng…
## 2  2020 norfolk    VA                 0       NA       NA Trump, Dona… Incumbent
## # ℹ 3 more variables: DemCandidate <chr>, DemStatus <chr>, GOP_win <dbl>
```

> - The unit of analysis is a county-by-presidential election. We have information on the state, county, and year, along with counts of the total votes for the Republican, Democrat, and in total; as well as information on the candidates themselves, including their name and their “status”. The data covers the period from 1972 to 2020. There are only two missing observations for both the Republican Votes and the Democrat Votes: Bedford City, VA in 2016 and Norfolk County, VA in 2020.

# Question 2 [2 points]

Perform *k*-means analysis on the Republican and Democrat votes with *k* = 2, and then plot the results, coloring the points by cluster assignment. Then predict the `GOP_win` binary outcome as a function of the cluster assignment using a logit regression. Make sure to `factor(cluster)` in the regression. What is the AUC for this model? Finally, use cross validation with an 80-20% split to re-calculate the AUC. Overall, would you say that the *k*-means algorithm helps you predict which counties vote Republican?

```
set.seed(123)
# K-means with k = 2
kRes <- dat %>%
  select(DemVotes,RepVotes) %>%
  drop_na() %>%
  kmeans(centers = 2)

# Plotting the result
dat %>%
  select(DemVotes,RepVotes) %>%
  drop_na() %>%
  mutate(cluster = kRes$cluster) %>%
  ggplot(aes(x = RepVotes,y = DemVotes,color = factor(cluster),group = 1)) +
  geom_point() +
  labs(x = 'GOP Votes',
       y = 'Democrat Votes',
       color = 'k-means Cluster')
```



```
# Create dataset for analysis
toanal <- dat %>%
  select(DemVotes,RepVotes,GOP_win) %>%
  drop_na() %>%
  mutate(cluster = kRes$cluster)

# Estimate logit model
summary(m <- glm(GOP_win ~ factor(cluster),toanal,family = binomial))
```

```
## 
## Call:
## glm(formula = GOP_win ~ factor(cluster), family = binomial, data = toanal)
## 
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)    
## (Intercept)      -1.04002    0.08125   -12.8   <2e-16 ***
## factor(cluster)2  2.10067    0.08206    25.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 46903  on 40448  degrees of freedom
## Residual deviance: 46126  on 40447  degrees of freedom
## AIC: 46130
## 
## Number of Fisher Scoring iterations: 4
```

```
# Calculate AUC
roc_auc(toanal %>%
          mutate(prob_win = predict(m,type = 'response'),
                 truth = factor(GOP_win,levels = c('1','0'))),
        truth,prob_win)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.523
```

```
# Calculate cross-validated result
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(toanal),size = round(nrow(toanal)*.8),replace = F)
  train <- toanal %>% slice(inds)
  test <- toanal %>% slice(-inds)

  summary(m <- glm(GOP_win ~ factor(cluster),train,family = binomial))


  cvRes <- cvRes %>%
    bind_rows(roc_auc(test %>%
                        mutate(prob_win = predict(m,newdata = test,type = 'response'),
                               truth = factor(GOP_win,levels = c('1','0'))),
                      truth,prob_win) %>%
                mutate(cvInd = i))
}

# Cross-validated AUC
cvRes %>%
  summarise(auc = mean(.estimate))
```

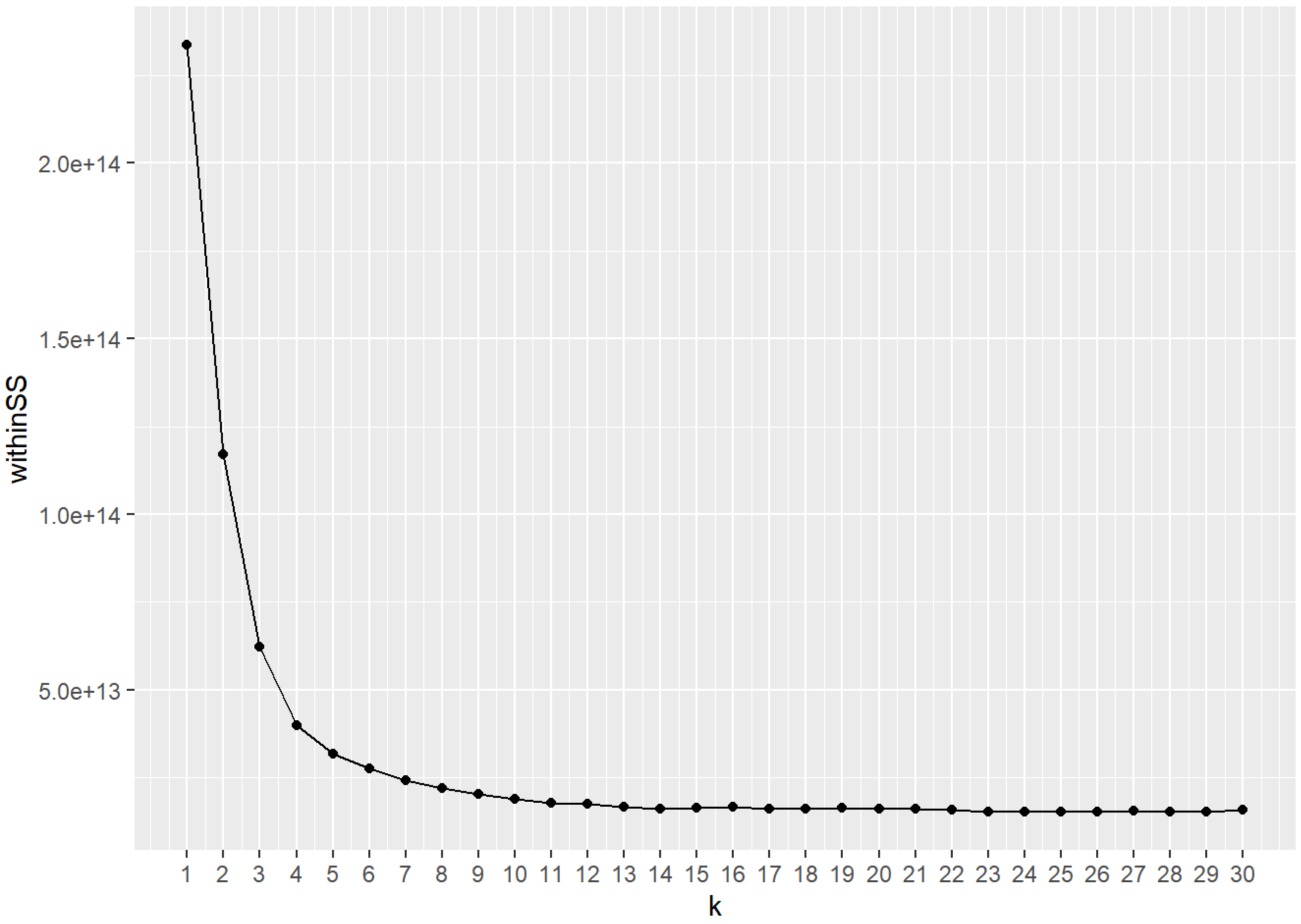
```
## # A tibble: 1 × 1
##     auc
##   <dbl>
## 1 0.524
```

> - The AUC in either the full data or in the cross validated result is a very low 0.52, meaning that this is a very poor model. This means that the *k*-means clustering algorithm’s groups were not helpful in predicting whether a county would elect a Republican or a Democrat.

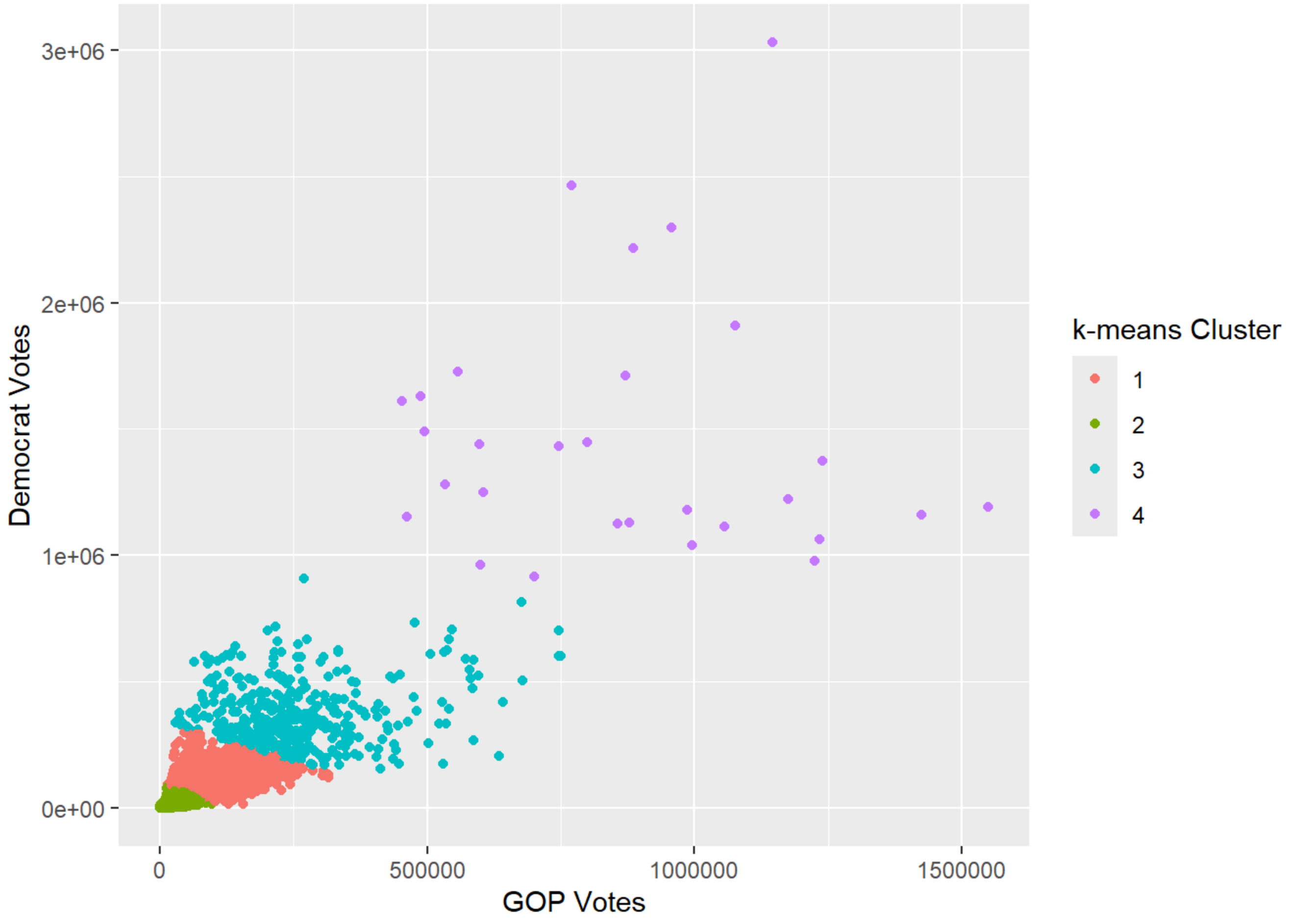
# Question 3 [2 points]

Now create an elbow plot by looping over potential values of *k* from 1 to 30 and plotting the *k* on the x-axis and the total Within Sum of Squares (total WSS) on the y-axis. What value of *k* would you use? Then re-run the preceding analysis with that value of *k* and interpret the results. Does the model improve?

```
# Looking at multiple values of k
kRes <- NULL
for(k in 1:30) {
  # Calculate k-means cluster solution for given value of k
  kResTmp <- dat %>%
  select(DemVotes,RepVotes) %>%
  drop_na() %>%
  kmeans(centers = k)

  # Save result including value of k and the total WSS
  kRes <- data.frame(withinSS = kResTmp$tot.withinss,
             k = k) %>%
    bind_rows(kRes)
}

# Plotting the elbow plot. Looks like k=4 is the elbow?
kRes %>%
  ggplot(aes(x = k,y = withinSS)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = 1:30)
```



```
# Rerunning with k = 4
kRes <- dat %>%
  select(DemVotes,RepVotes) %>%
  drop_na() %>%
  kmeans(centers = 4,nstart = 25)

# Plotting again
dat %>%
  select(DemVotes,RepVotes) %>%
  drop_na() %>%
  mutate(cluster = kRes$cluster) %>%
  ggplot(aes(x = RepVotes,y = DemVotes,color = factor(cluster),group = 1)) +
  geom_point() +
  labs(x = 'GOP Votes',
       y = 'Democrat Votes',
       color = 'k-means Cluster')
```



```
# Create dataset for analysis
toanal <- dat %>%
  select(DemVotes,RepVotes,GOP_win) %>%
  drop_na() %>%
  mutate(cluster = kRes$cluster)

# Estimate logit model
summary(m <- glm(GOP_win ~ factor(cluster),toanal,family = binomial))
```

```
##
## Call:
## glm(formula = GOP_win ~ factor(cluster), family = binomial, data = toanal)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.08831    0.04458  -1.981  0.04759 *
## factor(cluster)2  1.20423    0.04614  26.101  < 2e-16 ***
## factor(cluster)3 -1.18121    0.12554  -9.409  < 2e-16 ***
## factor(cluster)4 -1.74427    0.54036  -3.228  0.00125 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 46903  on 40448  degrees of freedom
## Residual deviance: 45704  on 40445  degrees of freedom
## AIC: 45712
##
## Number of Fisher Scoring iterations: 4
```

```
# Calculate AUC
roc_auc(toanal %>%
          mutate(prob_win = predict(m,type = 'response'),
                 truth = factor(GOP_win,levels = c('1','0'))),
        truth,prob_win)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.548
```

```
# Calculate cross-validated result
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(toanal),size = round(nrow(toanal)*.8),replace = F)
  train <- toanal %>% slice(inds)
  test <- toanal %>% slice(-inds)

  summary(m <- glm(GOP_win ~ factor(cluster),train,family = binomial))



  cvRes <- cvRes %>%
    bind_rows(roc_auc(test %>%
                        mutate(prob_win = predict(m,newdata = test,type = 'response'),
                               truth = factor(GOP_win,levels = c('1','0'))),
                      truth,prob_win) %>%
                mutate(cvInd = i))
}

# Cross-validated AUC
cvRes %>%
  summarise(auc = mean(.estimate))
```

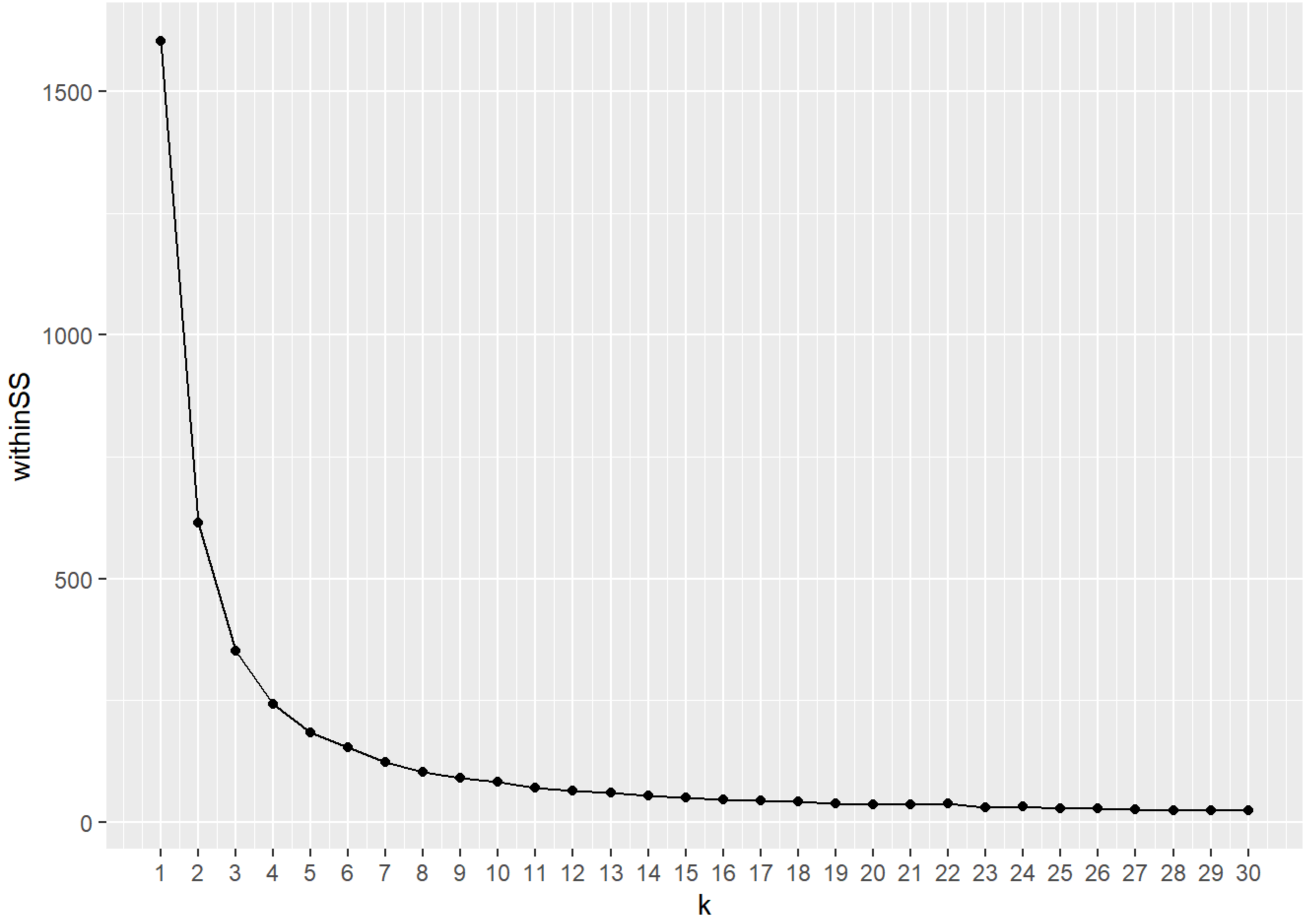
```
## # A tibble: 1 × 1
##     auc
##   <dbl>
## 1 0.547
```

> - Based on the elbow plot, I would choose a value of 4 because this is where the marginal reductions in within sum of squared errors diminish most clearly. In other words, this is roughly the “elbow” of the plot. However, my model is still terrible at predicting which party wins the presidential election.

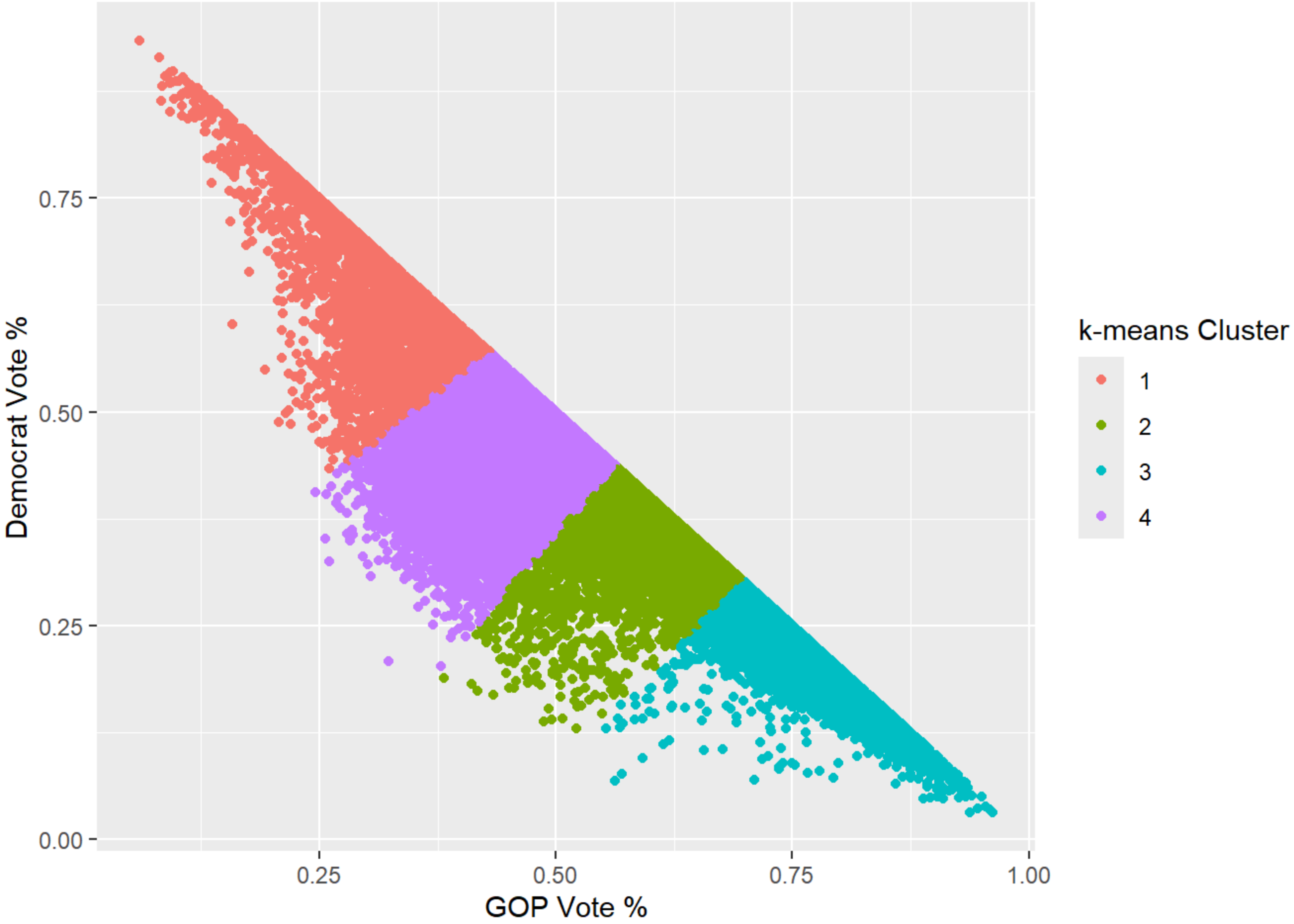
# Question 4 [2 points]

Re-do the preceding analysis except instead of using total votes, calculate the percent vote share for Democrats and Republicans in each county. Then identify the optimal value of *k* using the elbow plot visualization, use this as your value of *k* for the clustering solution and again plot the results. Now use a logit regression to predict `GOP_win` as a function of the cluster membership for each county and calculate the AUC using the same cross validation method. Does your answer change?

```r
# Wrangle data to get DEM and REP vote shares
dat <- dat %>%
  mutate(DemShare = DemVotes / TotalVotes,
         RepShare = RepVotes / TotalVotes)

# Re-calculate elbow plot
kRes <- NULL
for(k in 1:30) {
  kResTmp <- dat %>%
  select(DemShare,RepShare) %>% # Using vote shares instead of total votes
  drop_na() %>%
  kmeans(centers = k)

  kRes <- data.frame(withinSS = kResTmp$tot.withinss,
             k = k) %>%
    bind_rows(kRes)
}

# Plotting the elbow plot
kRes %>%
  ggplot(aes(x = k,y = withinSS)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = 1:30)
```



```
# Rerunning with k = 4
kRes <- dat %>%
  select(DemShare,RepShare) %>%
  drop_na() %>%
  kmeans(centers = 4,nstart = 25)

dat %>%
  select(DemShare,RepShare) %>%
  drop_na() %>%
  mutate(cluster = kRes$cluster) %>%
  ggplot(aes(x = RepShare,y = DemShare,color = factor(cluster),group = 1)) +
  geom_point() +
  labs(x = 'GOP Vote %',
       y = 'Democrat Vote %',
       color = 'k-means Cluster')
```



```
toanal <- dat %>%
  select(DemShare,RepShare,GOP_win) %>%
  drop_na() %>%
  mutate(cluster = kRes$cluster)

summary(m <- glm(GOP_win ~ factor(cluster),toanal,family = binomial))
```

```
##
## Call:
## glm(formula = GOP_win ~ factor(cluster), family = binomial, data = toanal)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -20.57     249.87  -0.082    0.934
## factor(cluster)2    41.13     291.99   0.141    0.888
## factor(cluster)3    41.13     314.03   0.131    0.896
## factor(cluster)4    20.79     249.87   0.083    0.934
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 46903  on 40448  degrees of freedom
## Residual deviance: 17789  on 40445  degrees of freedom
## AIC: 17797
##
## Number of Fisher Scoring iterations: 19
```

```
roc_auc(toanal %>%
          mutate(prob_win = predict(m,type = 'response'),
                 truth = factor(GOP_win,levels = c('1','0'))),
        truth,prob_win)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.935
```

```
cvRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(toanal),size = round(nrow(toanal)*.8),replace = F)
  train <- toanal %>% slice(inds)
  test <- toanal %>% slice(-inds)

  summary(m <- glm(GOP_win ~ factor(cluster),train,family = binomial))


  cvRes <- cvRes %>%
    bind_rows(roc_auc(test %>%
                        mutate(prob_win = predict(m,newdata = test,type = 'response'),
                               truth = factor(GOP_win,levels = c('1','0'))),
                      truth,prob_win) %>%
                mutate(cvInd = i))
}

cvRes %>%
  summarise(auc = mean(.estimate))
```

```
## # A tibble: 1 × 1
##     auc
##   <dbl>
## 1 0.935
```

- The model’s performance improves dramatically up to an AUC of 0.935! This is clearly a much better predictor to use!

# Extra Credit [2 points]

Provide an explanation for why the *k*-means results are so much more helpful when run using vote shares instead of total votes.

- The total votes measures are bad for predicting who wins the presidency because they divide counties by overall size, not politics. In other words, one cluster is created for larger counties who record more votes for both Democrat and Republican candidates simply because they are more populous, and another cluster is created for smaller counties. While potentially interesting for other analyses, these clusters are not helpful for predicting vote choice because they capture county size, not the political make-up.