Manipulating Data with dplyr

Howard Nguyen

2020-04-29

Load libraries

```
library(dplyr)
library(pscl)
#View(presidentialElections)
```

```
votes <- select(presidentialElections, year, demVote)
summary(presidentialElections)
```

```
##     state              demVote           year        south
##  Length:1097        Min.   :10.09   Min.   :1932   Mode :logical
##  Class :character   1st Qu.:40.18   1st Qu.:1952   FALSE:857
##  Mode  :character   Median :47.09   Median :1976   TRUE :240
##                     Mean   :48.36   Mean   :1975
##                     3rd Qu.:54.41   3rd Qu.:1996
##                     Max.   :98.57   Max.   :2016
```

```
# extract columns by name
votes <- presidentialElections[, c("year", "demVote")]
```

```
# select columns in range
select(presidentialElections, state:year)
```

```
## # A tibble: 1,097 x 3
##    state       demVote  year
##    <chr>         <dbl> <int>
##  1 Alabama        84.8  1932
##  2 Arizona        67.0  1932
##  3 Arkansas       86.3  1932
##  4 California     58.4  1932
##  5 Colorado       54.8  1932
##  6 Connecticut    47.4  1932
##  7 Delaware       48.1  1932
##  8 Florida        74.5  1932
##  9 Georgia        91.6  1932
## 10 Idaho          58.7  1932
## # ... with 1,087 more rows
```

```r
# select all columns except "south"
select(presidentialElections, -south)
```

```
## # A tibble: 1,097 x 3
##    state       demVote  year
##    <chr>         <dbl> <int>
##  1 Alabama        84.8  1932
##  2 Arizona        67.0  1932
##  3 Arkansas       86.3  1932
##  4 California     58.4  1932
##  5 Colorado       54.8  1932
##  6 Connecticut    47.4  1932
##  7 Delaware       48.1  1932
##  8 Florida        74.5  1932
##  9 Georgia        91.6  1932
## 10 Idaho          58.7  1932
## # ... with 1,087 more rows
```

Filter

```r
# select all rows from the 2008 election
votes_2008 <- filter(presidentialElections, year == 2008)
```

```r
# select all rows from the 2008 election -> same results as above
votes_2008 <- presidentialElections[presidentialElections$year == 2008, ]
head(votes_2008)
```

```
## # A tibble: 6 x 4
##   state      demVote  year south
##   <chr>        <dbl> <int> <lgl>
## 1 Alabama       38.7  2008 TRUE
## 2 Alaska        37.9  2008 FALSE
## 3 Arizona       44.9  2008 FALSE
## 4 Arkansas      38.9  2008 TRUE
## 5 California    60.9  2008 FALSE
## 6 Colorado      53.7  2008 FALSE
```

```r
# extract the row(s) for the state of CO in 2008
filter(presidentialElections, year == 2008, state == "Colorado")
```

```
## # A tibble: 1 x 4
##   state    demVote  year south
##   <chr>      <dbl> <int> <lgl>
## 1 Colorado    53.7  2008 FALSE
```

Mutate

Add an `other_parties_vote` column that is the percentage of votes for other parties Also add an `abs_vote_difference` column of the absolute difference between percentages Note you can use columns as you create them!"

```r
presidentialElections <- mutate(
  presidentialElections,
  other_parties_vote = 100 - demVote, # other parties is 100% - Democrat %
  abs_vote_difference = abs(demVote - other_parties_vote)
)
head(presidentialElections)
```

```
## # A tibble: 6 x 6
##   state       demVote  year south other_parties_vote abs_vote_difference
##   <chr>         <dbl> <int> <lgl>              <dbl>               <dbl>
## 1 Alabama        84.8  1932 TRUE                15.2                69.5
## 2 Arizona        67.0  1932 FALSE               33.0                34.1
## 3 Arkansas       86.3  1932 TRUE                13.7                72.5
## 4 California     58.4  1932 FALSE               41.6                16.8
## 5 Colorado       54.8  1932 FALSE               45.2                 9.62
## 6 Connecticut    47.4  1932 FALSE               52.6                 5.2
```

Arrange

The arrange() function allows us to sort the rows of data frame by some feature (column value)

```r
# Arrange rows in decreasing order by `year`, then by `demVote` within each `year`
presidentialElections <- arrange(presidentialElections, -year, demVote)
presidentialElections
```

```
## # A tibble: 1,097 x 6
##    state         demVote  year south other_parties_vote abs_vote_difference
##    <chr>           <dbl> <int> <lgl>              <dbl>               <dbl>
##  1 West Virginia    26.2  2016 FALSE               73.8                47.6
##  2 Utah             27.2  2016 FALSE               72.8                45.7
##  3 North Dakota     27.2  2016 FALSE               72.8                45.5
##  4 Idaho            27.5  2016 FALSE               72.5                45.0
##  5 Oklahoma         28.9  2016 FALSE               71.1                42.1
##  6 South Dakota     31.7  2016 FALSE               68.3                36.5
##  7 Kentucky         32.7  2016 FALSE               67.3                34.6
##  8 Arkansas         33.6  2016 TRUE                66.4                32.7
##  9 Nebraska         33.7  2016 FALSE               66.3                32.6
## 10 Alabama          34.4  2016 TRUE                65.6                31.3
## # ... with 1,087 more rows
```

Summarize

This is an aggregation operation (i.e., it will reduce an entire column to a single value—think about taking a sum or average)

```r
# Compute summary statistics for the `presidentialElections` data frame
average_vote <- summarize(
  presidentialElections,
  mean_dem_vote = mean(demVote),
  mean_other_parties = mean(other_parties_vote)
)
average_vote
```

```
## # A tibble: 1 x 2
##   mean_dem_vote mean_other_parties
##           <dbl>              <dbl>
## 1          48.4               51.6
```

```
summary(presidentialElections)
```

```
##     state              demVote           year        south
##  Length:1097        Min.   :10.09   Min.   :1932   Mode :logical
##  Class :character   1st Qu.:40.18   1st Qu.:1952   FALSE:857
##  Mode  :character   Median :47.09   Median :1976   TRUE :240
##                     Mean   :48.36   Mean   :1975
##                     3rd Qu.:54.41   3rd Qu.:1996
##                     Max.   :98.57   Max.   :2016
##  other_parties_vote abs_vote_difference
##  Min.   : 1.43      Min.   : 0.04
##  1st Qu.:45.59      1st Qu.: 7.14
##  Median :52.91      Median :14.98
##  Mean   :51.64      Mean   :19.25
##  3rd Qu.:59.82      3rd Qu.:25.82
##  Max.   :89.91      Max.   :97.14
```

```
# A function that returns the value in a vector furthest from 50
further_from_50 <- function(vec) {
  # subtract 50 from each value
  adjusted_values <- vec - 50
  # return the element with the largest absolute difference from 50
  vec[abs(adjusted_values) == max(abs(adjusted_values))]
}
```

```
# summarize the df, generating a column 'big_landslide"
# that stores the values further from 50%
summarize(
  presidentialElections,
  biggest_landslide = further_from_50(demVote)
)
```

```
## # A tibble: 1 x 1
##   biggest_landslide
##               <dbl>
## 1              98.6
```

Complex Analysis

Performaning sequential operations

Which state had the highest percentage of votes for the Dem Party candidate (BO) in 2008?

```
# use a sequence of steps to find the sate with the highest 2008 demVote %
# 1. filter down to only 2008
votes_2008 <- filter(presidentialElections, year == 2008)
```

```r
# 2. filter down to the state with the highest demVote
most_dem_votes <- filter(votes_2008, demVote == max(demVote))

# 3. select name of the state
most_dem_state <- select(most_dem_votes, state)
head(most_dem_state)
```

```
## # A tibble: 1 x 1
##   state
##   <chr>
## 1 DC
```

```r
# use nested functions to find the sate with the highest 2008 demVote %
most_dem_state <- select(   # 3. select name of the state
  filter(                  # 2. filter down to the highest demVote
    filter(                 # 1. filter down to only 2008 votes
      presidentialElections, # arguments for the step 1 filter
      year == 2008
    ),
    demVote == min(demVote) # second argument for the step 2 - filter
  ),
  state                    # second argument for the step 3 - select
)
head(most_dem_state)
```

```
## # A tibble: 1 x 1
##   state
##   <chr>
## 1 Wyoming
```

```r
# add a new column with percentage calculated
presidentialElections <- presidentialElections %>%
  mutate(percentage_votes = demVote/sum(demVote) * 100)
head(presidentialElections)
```

```
## # A tibble: 6 x 7
##   state         demVote  year south other_parties_vote abs_vote_differ~1 perce~2
##   <chr>           <dbl> <int> <lgl>              <dbl>             <dbl>   <dbl>
## 1 West Virginia    26.2  2016 FALSE               73.8              47.6  0.0493
## 2 Utah             27.2  2016 FALSE               72.8              45.7  0.0512
## 3 North Dakota     27.2  2016 FALSE               72.8              45.5  0.0513
## 4 Idaho            27.5  2016 FALSE               72.5              45.0  0.0518
## 5 Oklahoma         28.9  2016 FALSE               71.1              42.1  0.0545
## 6 South Dakota     31.7  2016 FALSE               68.3              36.5  0.0598
## # ... with abbreviated variable names 1: abs_vote_difference,
## #   2: percentage_votes
```

```r
new_df <- select(presidentialElections, - other_parties_vote, - abs_vote_difference)
```

```r
# sorted asc
new_df[order(new_df$percentage_votes), ]
```

```
## # A tibble: 1,097 x 5
##    state          demVote  year south percentage_votes
##    <chr>            <dbl> <int> <lgl>            <dbl>
##  1 Mississippi       10.1  1948 TRUE            0.0190
##  2 Mississippi       12.9  1964 TRUE            0.0242
##  3 Alabama           18.7  1968 TRUE            0.0353
##  4 Mississippi       19.6  1972 TRUE            0.0370
##  5 Utah              20.6  1980 FALSE           0.0388
##  6 Mississippi       23.0  1968 TRUE            0.0434
##  7 Oklahoma          24    1972 FALSE           0.0452
##  8 South Carolina    24.1  1948 TRUE            0.0455
##  9 Utah              24.6  1992 FALSE           0.0465
## 10 Georgia           24.6  1972 TRUE            0.0465
## # ... with 1,087 more rows
```

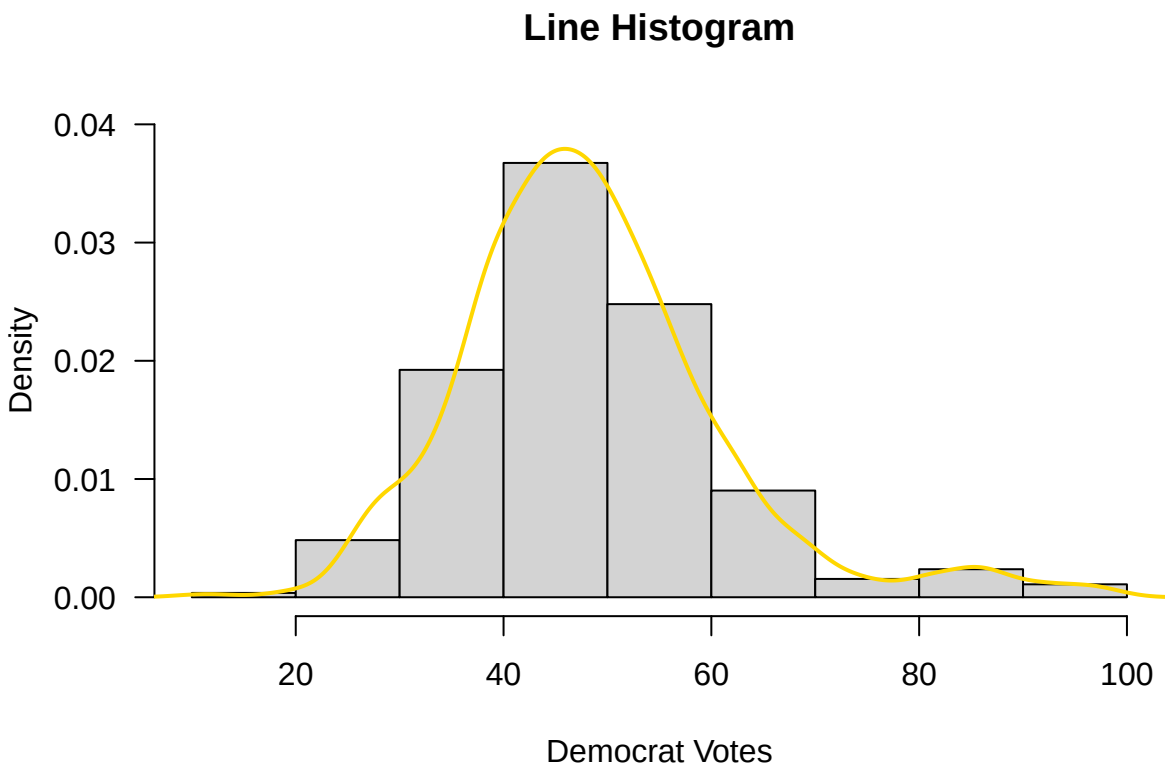
```r
# sorted desc
new_df[order(-new_df$percentage_votes), ]
```

```
## # A tibble: 1,097 x 5
##    state          demVote  year south percentage_votes
##    <chr>            <dbl> <int> <lgl>            <dbl>
##  1 South Carolina    98.6  1936 TRUE             0.186
##  2 South Carolina    98.0  1932 TRUE             0.185
##  3 Mississippi       97.0  1936 TRUE             0.183
##  4 Mississippi       96.0  1932 TRUE             0.181
##  5 Mississippi       95.7  1940 TRUE             0.180
##  6 South Carolina    95.6  1940 TRUE             0.180
##  7 Mississippi       93.6  1944 TRUE             0.176
##  8 Louisiana         92.8  1932 TRUE             0.175
##  9 DC                92.5  2008 FALSE            0.174
## 10 Georgia           91.6  1932 TRUE             0.173
## # ... with 1,087 more rows
```

```r
filter(new_df, year == 2008, state == 'California')
```

```
## # A tibble: 1 x 5
##   state      demVote  year south percentage_votes
##   <chr>        <dbl> <int> <lgl>            <dbl>
## 1 California    60.9  2008 FALSE            0.115
```

```r
# histogram and density
hist(new_df$demVote, freq = FALSE, ylim = c(0, 0.04),
     xlab="Democrat Votes", las = 1, main = "Line Histogram")
lines(density(new_df$demVote), lwd = 2, col = "gold")
```



Time Series

```
library(forecast)
```

```
# create a time series object
ts_data <- ts(data = new_df$demVote, start = c(min(new_df$year)),
              end = c(max(new_df$year)), frequency = 1)
# use the auto.arima() function to identify the best ARIMA model for the data
model <- auto.arima(ts_data)
```

```
# generate forecasts for the next 5 years
forecast_data <- forecast(model, h = 1)

# add the forecasted values as a new column to the original df
df_forecast <- new_df %>%
  mutate(forecast = as.numeric(forecast_data$mean))
```

```
# view the forecasted data
print(df_forecast)
```

```
## # A tibble: 1,097 x 6
##    state         demVote  year south percentage_votes forecast
##    <chr>           <dbl> <int> <lgl>            <dbl>    <dbl>
##  1 West Virginia    26.2  2016 FALSE           0.0493     52.8
##  2 Utah             27.2  2016 FALSE           0.0512     52.8
##  3 North Dakota     27.2  2016 FALSE           0.0513     52.8
```

```
##  4 Idaho            27.5  2016 FALSE           0.0518     52.8
##  5 Oklahoma         28.9  2016 FALSE           0.0545     52.8
##  6 South Dakota     31.7  2016 FALSE           0.0598     52.8
##  7 Kentucky         32.7  2016 FALSE           0.0616     52.8
##  8 Arkansas         33.6  2016 TRUE            0.0634     52.8
##  9 Nebraska         33.7  2016 FALSE           0.0635     52.8
## 10 Alabama          34.4  2016 TRUE            0.0648     52.8
## # ... with 1,087 more rows
```

```
# libraries
library(ggplot2)
library(lubridate)
# install.packages("forecast")
library(forecast)
library(dplyr)
```

```
# assuming your data frame is called `df`
# convert year column to a date object
#head(df_forecast)
#str(df_forecast)
#summary(df_forecast$forecast)
head(new_df)
```

```
## # A tibble: 6 x 5
##   state         demVote  year south percentage_votes
##   <chr>           <dbl> <int> <lgl>            <dbl>
## 1 West Virginia    26.2  2016 FALSE           0.0493
## 2 Utah             27.2  2016 FALSE           0.0512
## 3 North Dakota     27.2  2016 FALSE           0.0513
## 4 Idaho            27.5  2016 FALSE           0.0518
## 5 Oklahoma         28.9  2016 FALSE           0.0545
## 6 South Dakota     31.7  2016 FALSE           0.0598
```

```
str(new_df)
```

```
## tibble [1,097 x 5] (S3: tbl_df/tbl/data.frame)
##  $ state           : chr [1:1097] "West Virginia" "Utah" "North Dakota" "Idaho" ...
##  $ demVote         : num [1:1097] 26.2 27.2 27.2 27.5 28.9 ...
##  $ year            : int [1:1097] 2016 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ south           : logi [1:1097] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ percentage_votes: num [1:1097] 0.0493 0.0512 0.0513 0.0518 0.0545 ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 5
##   .. ..$ X1     : list()
##   .. .. ..- attr(*, "class")= chr [1:2] "collector_integer" "collector"
##   .. ..$ state  : list()
##   .. .. ..- attr(*, "class")= chr [1:2] "collector_character" "collector"
##   .. ..$ demVote: list()
##   .. .. ..- attr(*, "class")= chr [1:2] "collector_double" "collector"
##   .. ..$ year   : list()
##   .. .. ..- attr(*, "class")= chr [1:2] "collector_integer" "collector"
##   .. ..$ south  : list()
```

```
##   .. .. ..- attr(*, "class")= chr [1:2] "collector_logical" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr [1:2] "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

```
df <- new_df
head(df)
```

```
## # A tibble: 6 x 5
##   state         demVote  year south percentage_votes
##   <chr>           <dbl> <int> <lgl>            <dbl>
## 1 West Virginia    26.2  2016 FALSE           0.0493
## 2 Utah             27.2  2016 FALSE           0.0512
## 3 North Dakota     27.2  2016 FALSE           0.0513
## 4 Idaho            27.5  2016 FALSE           0.0518
## 5 Oklahoma         28.9  2016 FALSE           0.0545
## 6 South Dakota     31.7  2016 FALSE           0.0598
```

```
#df_forecast$year <- as.Date(paste0(df_forecast$year, "-01-01"))
```

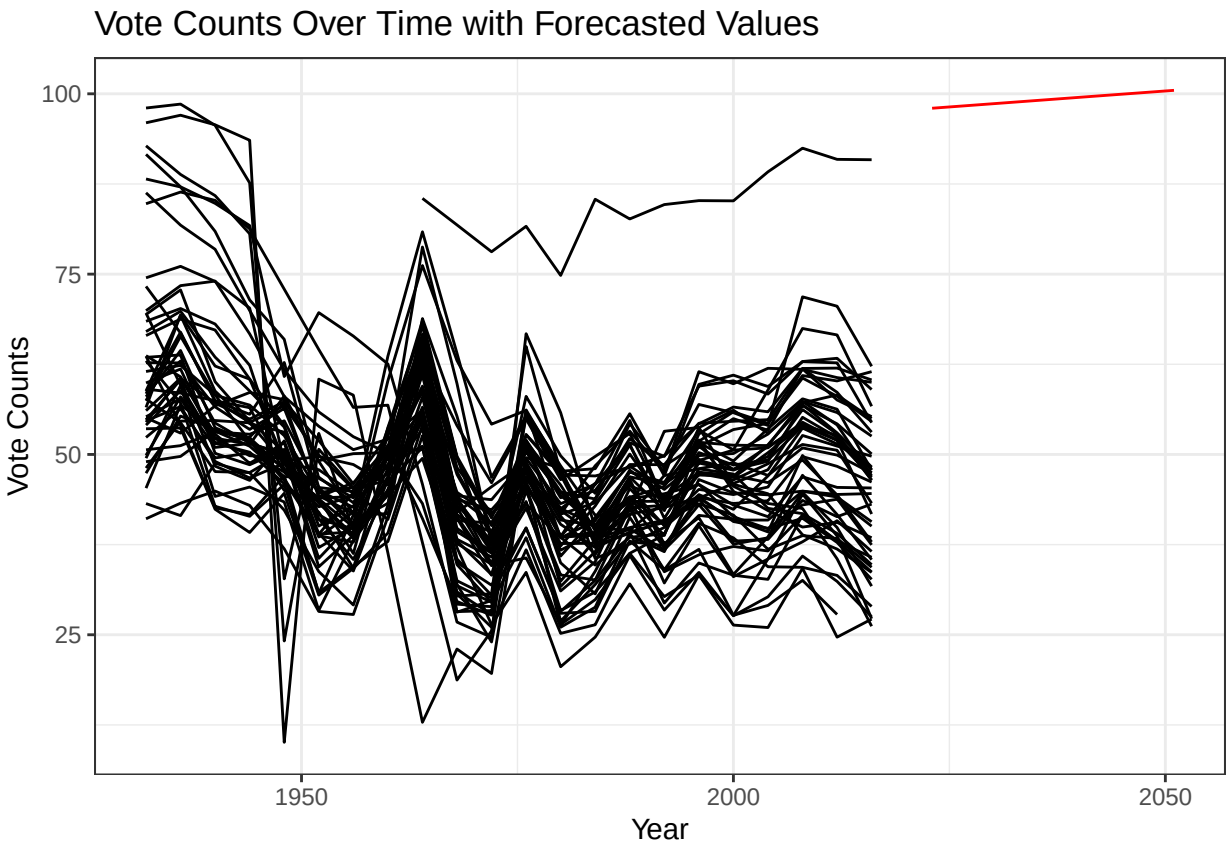
```
# convert year column to a date object
df$year <- as.Date(paste0(df$year, "-01-01"))
```

```
# create time series object with vote_counts column
ts_data <- ts(df$demVote, start = c(year(df$year)[1], 1), frequency = 4)
```

```
# generate forecast values
forecasted_values <- forecast(ts_data, h = 8)
```

```
# create a new data frame for forecasted values
forecast_data <- data.frame(year = seq(as.Date("2023-01-01"), by = "4 years",
                                       length.out = length(forecasted_values$mean)),
                            forecast = forecasted_values$mean)
```

```
# plot the data
ggplot() +
  geom_line(data = df, aes(x = year, y = demVote, group = state)) +
  geom_line(data = forecast_data, aes(x = year, y = forecast), color = "red") +
  xlab("Year") +
  ylab("Vote Counts") +
  ggtitle("Vote Counts Over Time with Forecasted Values") +
  theme_bw()
```


Vote Counts Over Time with Forecasted Values
100
75
50
25
Vote Counts
1950
2000
2050
Year