Q1. Election Returns

Comprehensive results for these Senate races have been compiled by the MIT Election Data and Science lab. We're going to do a bit of cleaning to get this data into a format that we can use for our maps.

a. First, import this data using this link. Download the spreadhseet, and load the election results into R.

b. Our first step to clean this data is removing non-substantive columns. Keep only columns 1-3, 9, 11-12, and 15-16.

```r
dt <- dt[c(1:3,9,11,12,15,16)]
```

c. Next, we're going to remove rows with some incomplete data. Remove any rows that have missing data in the "candidate" or "party" columns.

```r
dt <- dt %>%
  filter(!is.na(candidate) & !is.na(party) &
           candidate != '' & party != '')
```

d. Next, create a new variable, "party3", which recodes the "party" column into "D" for Democrats, "R" for Republicans, and "I" for all other parties. You may want to initialize the column with "I" first, then use the ifelse() function.

```r
dt['party3'] <- 'I'
dt <- dt %>%
  mutate(party3= ifelse(party=='democrat', 'D',
                 ifelse(party=='republican', 'R', 'I')))
```

e. How many Democrats are in this dataset? How many Republicans? How many Independents?

```r
table(dt$party3)
```

```
##
##    D    I    R
##  748 1354  756
```

ANS: 748 democrates, 756 republicans and 1354 Independents in this dataset

f. Now, let's look at the 2-party vote in these data.

i. First, remove the independent candidates from the data.

```r
dt <- dt %>%
  filter(party3!='I')
```

ii. Next, remove all rows where "stage" is not equal to "gen". This ensures that we only get results from the general elections.

```r
dt <- dt %>%
  filter(state != 'gen')
```

g. How many races were contested between more than two candidates? Which state had the most of these races?

```r
dt_agg <- dt %>%
  group_by(year, state) %>%
  summarize(num_candidates = n()) %>%
  mutate(over2_cand = ifelse(num_candidates>2, 1,0))
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

```r
sum(dt_agg$over2_cand)
```

```
## [1] 22
```

```r
# look for the state had the most of these races
dt_agg %>%
  filter(over2_cand==1) %>%
  group_by(state) %>%
  summarise(num_races=n()) %>%
  arrange(-num_races) %>%
  head()
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 6 x 2
##   state       num_races
##   <chr>           <int>
## 1 Louisiana           6
## 2 Georgia             2
## 3 Mississippi         2
## 4 Alabama             1
## 5 Arizona             1
## 6 California          1
```

ANS: 22 races were contested between more than two candidates. Louisiana had the most of these races.

h. Let's take a look at the races from 2012. Create two new dataframes, one that includes only results from Democratic candidates in 2012 and one that only includes results for Republicans.

```
dt2012_d <- dt %>%
  filter(year==2012 & party3=='D')
dt2012_r <- dt %>%
  filter(year==2012 & party3=='R')
```

i. In each dataset, create a new column called "voteprop" that records the proportion of the vote each candidate recieved.

```
dt2012_d <- dt2012_d %>%
  mutate(voteprop=candidatevotes/totalvotes*100)
dt2012_r <- dt2012_r %>%
  mutate(voteprop=candidatevotes/totalvotes*100)
```

j. Join these two datasets together, taking care to preserve all rows. Join on the "state", "state.po", "stage", "totalvotes", and "year" variables. You should end up with a dataset with one row per state.

```
df <- dt2012_d %>%
  full_join(dt2012_r,
            by=c('state', 'state_po', 'stage', 'totalvotes', 'year'),
            suffix = c(".D", ".R"))
```

k. Create a variable "demwin" that records if the Democrat recieved a higher vote share than the Republican.

```
df <- df %>%
  mutate(dewin = ifelse(voteprop.D>voteprop.R, 1, 0))
```

l. Create a variable "demdiff" that records the difference between the Democratic and Republican share of the vote.

```
df <- df %>%
  mutate(demdiff = voteprop.D-voteprop.R)
```

m. Next, we're going to do some analysis to map this data. Load in the state-level mapping data that we worked with in Lesson 9 from the package mapdata.

```
states <- map_data("state")
```

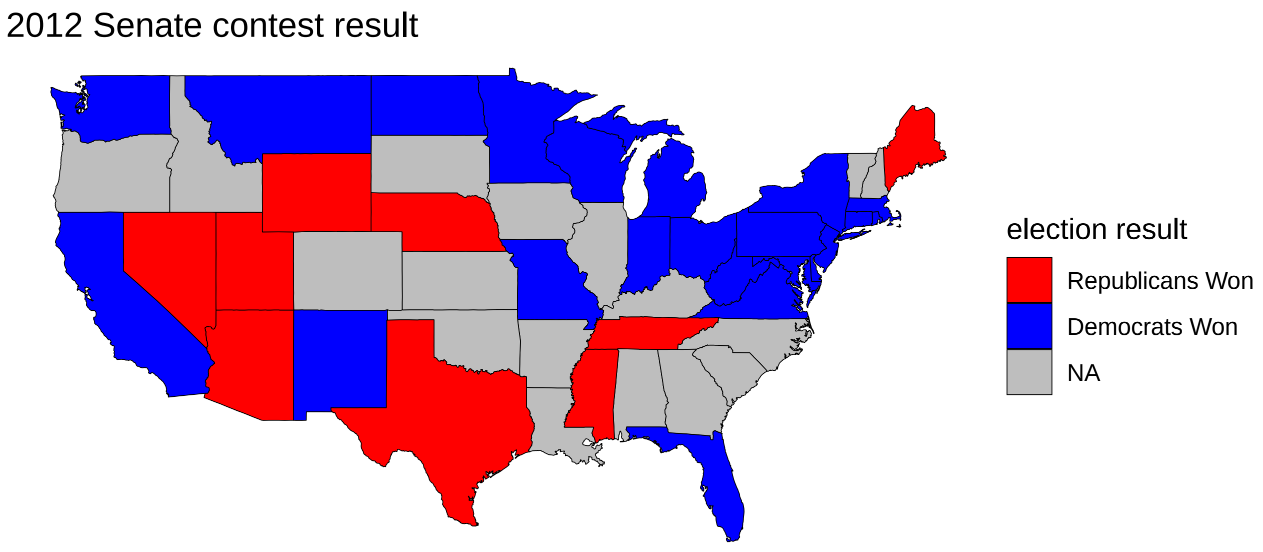
n. Join our 2012 Senate election data to this mapping data. You may need to convert the state names to lowercase.

```r
df <- df %>%
  mutate(state = tolower(state)) %>%
  full_join(states, by=c('state'='region'))
```

o. Create a map that shows the winner of each Senate contest in 2012, with Democrats in blue and Republicans in red. If there was no state contest (or in the case of Vermont, where an Independent won the seat), color the state grey.

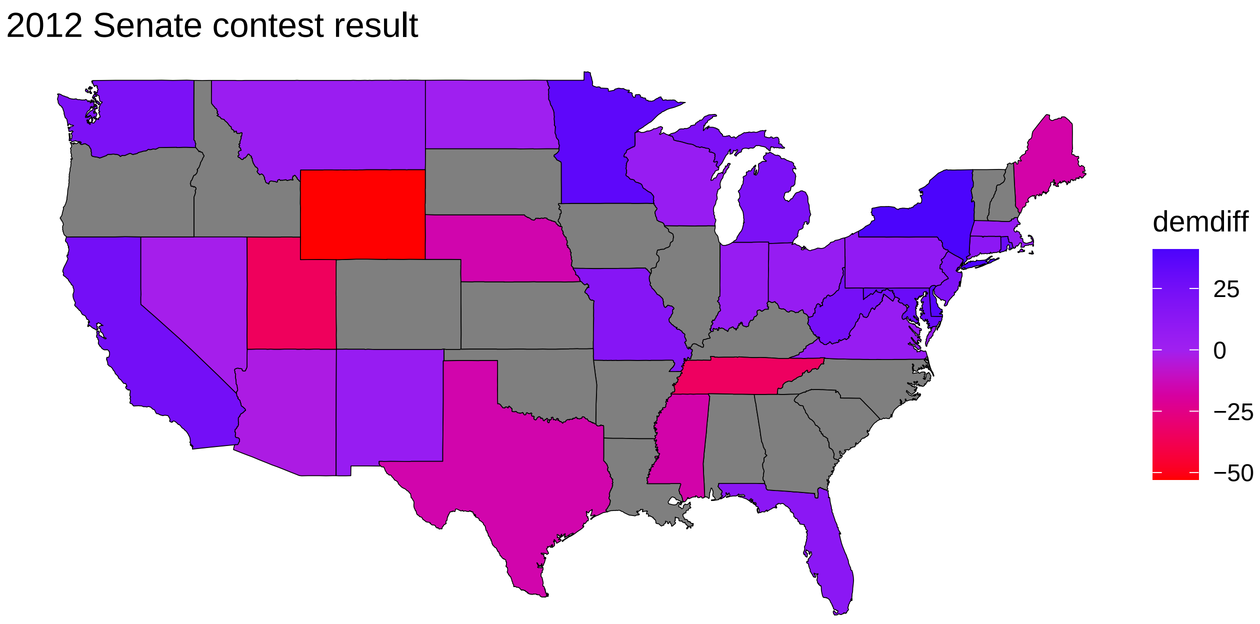
```r
df <- df %>%
  mutate(dewin=replace(dewin, is.na(dewin), 3))

ggplot(data=df) +
  geom_polygon(aes(x=long, y=lat, group=group, fill=as.factor(dewin)), col='black', linetype=1, lwd=0.15
  coord_quickmap() +
  scale_fill_manual(values = c("red", "blue", 'grey'),
                    name="election result",
                    labels = c("Republicans Won", "Democrats Won", 'NA')) +
  theme_void()+
  ggtitle("2012 Senate contest result")
```



i. Create a similar map that shades each state by the Democratic vote difference. If there was no state contest (or in the case of Vermont, where an Independent won the seat), color the state grey.

```r
ggplot(data=df, aes(x=long, y=lat, group=group)) +
  geom_polygon(fill='grey') +
  geom_polygon(aes(fill=demdiff), col='black', lwd=0.15) +
  coord_quickmap() +
  scale_fill_gradient2(
    low="red",
    mid="purple",
    high="blue",
    midpoint = 0) +
  # scale_fill_gradient2(
  #   low="red",
  #   mid="white",
  #   high="blue",
  #   midpoint = 0) +
  theme_void()+
  ggtitle("2012 Senate contest result")
```


2012 Senate contest result
demdiff
25
0
−25
−50