Assignment 1: Data Visualization - AMSE

KUZNETCOV, Mark; LAINES MEDINA, Betsy

2025-10-07

1. Install the 'ukelection2019' package by typing: remotes::install_github("kjhealy/ukelection2019") (0.5 pt)

1. Install the 'ukelection2019' package

> *Note: Installation should be done manually in the R console before knitting.*
> *Do not include install.packages() or remotes::install_github() in the RMarkdown file.*

To install it manually (outside this RMarkdown):

```
install.packages("remotes")
remotes::install_github("kjhealy/ukelection2019")
```

---

2. Load the package and inspect the data

```
data(package = "ukelection2019")
head(ukvote2019)
```

```
## # A tibble: 6 x 13
##   cid      constituency electorate party_name candidate votes vote_share_percent
##   <chr>    <chr>             <int> <chr>      <chr>     <int>              <dbl>
## 1 W070000~ Aberavon          50747 Labour     Stephen ~ 17008               53.8
## 2 W070000~ Aberavon          50747 Conservat~ Charlott~  6518               20.6
## 3 W070000~ Aberavon          50747 The Brexi~ Glenda D~  3108                9.8
## 4 W070000~ Aberavon          50747 Plaid Cym~ Nigel Hu~  2711                8.6
## 5 W070000~ Aberavon          50747 Liberal D~ Sheila K~  1072                3.4
## 6 W070000~ Aberavon          50747 Independe~ Captain ~   731                2.3
## # i 6 more variables: vote_share_change <dbl>, total_votes_cast <int>,
## #   vrank <int>, turnout <dbl>, fname <chr>, lname <chr>
```

3. What were the results in the Dover constituency? (0.75 pt)

```
dover_results <- subset(ukvote2019, constituency == "Dover")
dover_results
```

```
## # A tibble: 6 x 13
##   cid      constituency electorate party_name candidate votes vote_share_percent
##   <chr>    <chr>             <int> <chr>      <chr>     <int>              <dbl>
## 1 E140006~ Dover             76355 Conservat~ Natalie ~ 28830               56.9
## 2 E140006~ Dover             76355 Labour     Charlott~ 16552               32.6
## 3 E140006~ Dover             76355 Liberal D~ Simon Do~  2895                5.7
## 4 E140006~ Dover             76355 Green      Beccy Sa~  1371                2.7
## 5 E140006~ Dover             76355 Independe~ Nathan S~   916                1.8
## 6 E140006~ Dover             76355 Women’s E~ Eljai Mo~   137                0.3
## # i 6 more variables: vote_share_change <dbl>, total_votes_cast <int>,
## #   vrank <int>, turnout <dbl>, fname <chr>, lname <chr>
```

4. How many constituencies are there? (0.75 pt)

```
num_constituencies <- length(unique(ukvote2019$constituency))
num_constituencies
```

```
## [1] 650
```

5. How many candidates did each party run in the election as a whole? (0.75 pt)

```
candidates_per_party <- ukvote2019 %>%
  group_by(party_name) %>%
  summarise(num_candidates = n())
candidates_per_party
```

```
## # A tibble: 69 x 2
##    party_name                          num_candidates
##    <chr>                                        <int>
##  1 Advance Together                                 5
##  2 Alliance Party                                  18
##  3 Alliance for Green Socialism                     3
##  4 Animal Welfare Party                             6
##  5 Aontú                                            7
##  6 Ashfield Independents                            1
##  7 Best for Luton                                   1
##  8 Birkenhead Social Justice Party                  1
##  9 British National Party                           1
## 10 Burnley & Padiham Independent Party              1
## # i 59 more rows
```

6. Show the number of candidates fielded by party from highest to lowest. (0.75 pt)

```
candidates_per_party_arranged <- ukvote2019 %>%
  group_by(party_name) %>%
  summarise(num_candidates = n()) %>%
  arrange(desc(num_candidates))
candidates_per_party_arranged
```

```
## # A tibble: 69 x 2
##    party_name                 num_candidates
##    <chr>                               <int>
##  1 Conservative                          636
##  2 Labour                                631
##  3 Liberal Democrat                      611
##  4 Green                                 497
##  5 The Brexit Party                      275
##  6 Independent                           224
##  7 Scottish National Party                59
##  8 UKIP                                   44
##  9 Plaid Cymru                            36
## 10 Christian Peoples Alliance             29
## # i 59 more rows
```

7. Calculate the percentage of all candidates fielded by each party (0.75 pt)

```
candidate_percentages <- ukvote2019 %>%
  group_by(party_name) %>%
  summarise(num_candidates = n()) %>%
  mutate(percentage = 100 * num_candidates / sum(num_candidates)) %>%
  arrange(desc(percentage))
candidate_percentages
```

```
## # A tibble: 69 x 3
##    party_name                 num_candidates percentage
##    <chr>                               <int>      <dbl>
##  1 Conservative                          636     19.2
##  2 Labour                                631     19.0
##  3 Liberal Democrat                      611     18.4
##  4 Green                                 497     15.0
##  5 The Brexit Party                      275      8.28
##  6 Independent                           224      6.75
##  7 Scottish National Party                59      1.78
##  8 UKIP                                   44      1.33
##  9 Plaid Cymru                            36      1.08
## 10 Christian Peoples Alliance             29      0.873
## # i 59 more rows
```

8. Find the five worst-performing candidates overall, as measured by number of votes. (0.75 pt)

```
worst_candidates <- ukvote2019 %>%
  arrange(votes) %>%
  slice(1:5)
worst_candidates
```

```
## # A tibble: 5 x 13
##   cid      constituency electorate party_name candidate votes vote_share_percent
##   <chr>    <chr>             <int> <chr>      <chr>     <int>              <dbl>
## 1 E140010~ Uxbridge & ~      70369 Independe~ William ~     5                0
## 2 E140010~ Uxbridge & ~      70369 Independe~ Bobby El~     8                0
## 3 E140010~ Uxbridge & ~      70369 Independe~ Norma Bu~    22                0
## 4 E140010~ Uxbridge & ~      70369 Independe~ Yace Yog~    23                0
## 5 E140007~ Kensington        64609 Workers R~ Scott Do~    28                0.1
## # i 6 more variables: vote_share_change <dbl>, total_votes_cast <int>,
## #   vrank <int>, turnout <dbl>, fname <chr>, lname <chr>
```

9. Find the five largest constituencies in the country as measured by votes cast (0.75 pt)

```
largest_constituencies <- ukvote2019 %>%
  group_by(constituency) %>%
  summarise(total_votes_cast = sum(votes, na.rm = TRUE)) %>%
  arrange(desc(total_votes_cast)) %>%
  slice(1:5)
largest_constituencies
```

```
## # A tibble: 5 x 2
##   constituency           total_votes_cast
##   <chr>                             <int>
## 1 Bristol West                      75528
## 2 Isle of Wight                     74442
## 3 Wantage                           67173
## 4 Cambridgeshire South              66929
## 5 Northamptonshire South            66908
```

10. Find the total number of votes cast nationally for the Labour and Conservative parties. (0.75 pt)

```
total_votes_major_parties <- ukvote2019 %>%
  filter(party_name %in% c("Labour", "Conservative")) %>%
  group_by(party_name) %>%
  summarise(total_votes = sum(votes, na.rm = TRUE))
total_votes_major_parties
```

```
## # A tibble: 2 x 2
##   party_name   total_votes
##   <chr>              <int>
## 1 Conservative    13993282
## 2 Labour          10269076
```

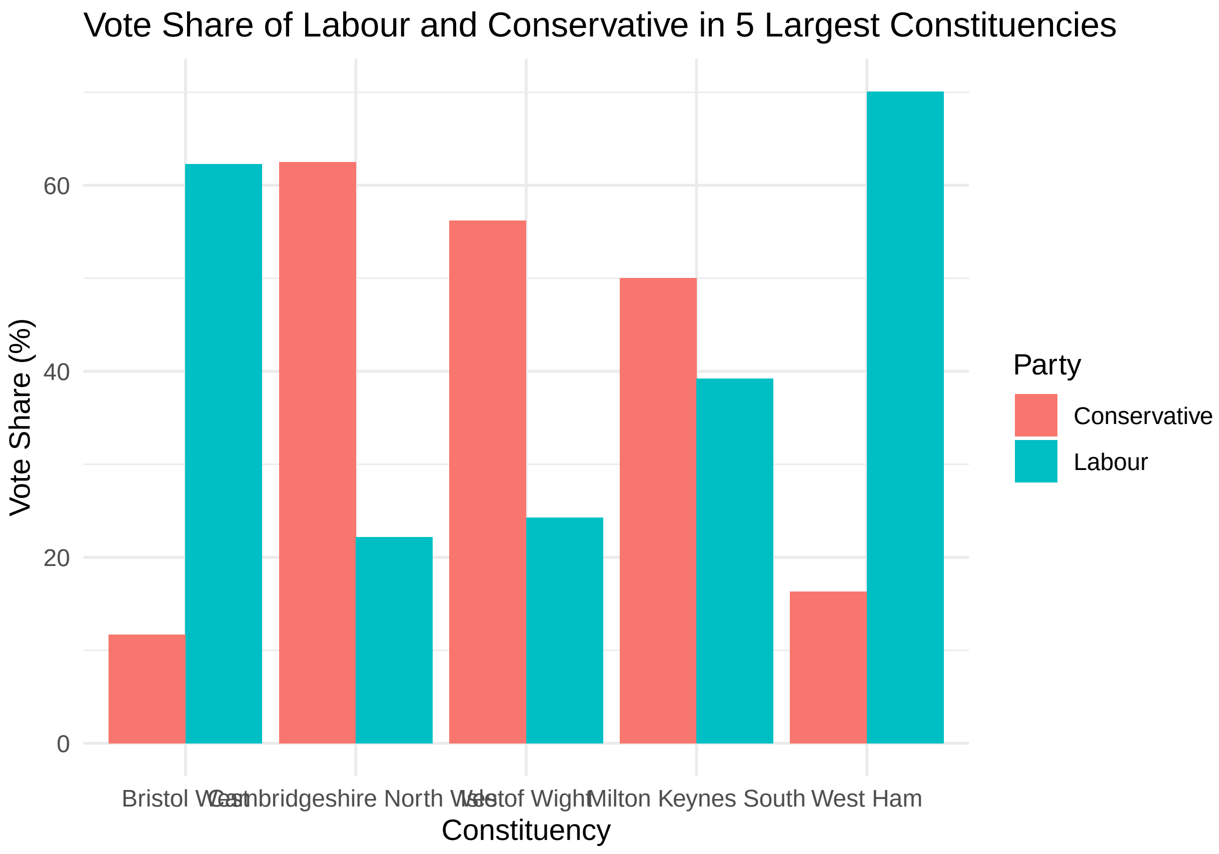
11. Find the ten largest vote-winning candidates in the country, as measured by percentage of vote share in their constituency. (0.75 pt)

```
top_winning_candidates <- ukvote2019 %>%
  filter(vrank == 1) %>%
  arrange(desc(vote_share_percent)) %>%
  slice(1:10)
top_winning_candidates
```

```
## # A tibble: 10 x 13
##    cid     constituency electorate party_name candidate votes vote_share_percent
##    <chr>   <chr>             <int> <chr>      <chr>     <int>              <dbl>
##  1 E14000~ Liverpool W~      62628 Labour     Dan Card~ 34538               84.7
##  2 E14000~ Knowsley          84060 Labour     George H~ 44374               80.8
##  3 E14000~ Bootle            74832 Labour     Peter Do~ 39066               79.4
##  4 E14000~ Birmingham ~      74912 Labour     Shabana ~ 33355               79.2
##  5 E14000~ Birmingham ~      78295 Labour     Liam Byr~ 35397               78.7
##  6 E14000~ Liverpool R~      80310 Labour     Kim John~ 41170               78
##  7 E14000~ Liverpool W~      65640 Labour     Ian Byrne 34117               77.6
##  8 E14000~ Manchester ~      76419 Labour     Afzal Kh~ 34583             77.6
##  9 E14000~ Boston & Sk~      69265 Conservat~ Matt War~ 31963             76.7
## 10 E14000~ Castle Point      69643 Conservat~ Rebecca ~ 33971             76.7
## # i 6 more variables: vote_share_change <dbl>, total_votes_cast <int>,
## #   vrank <int>, turnout <dbl>, fname <chr>, lname <chr>
```

12. Show in one single bar graph the votes share of the Labour and the Conservative parties for each of the 5 biggest constituencies in terms of electors. There should be two bars per constituency. (1.25 pt)

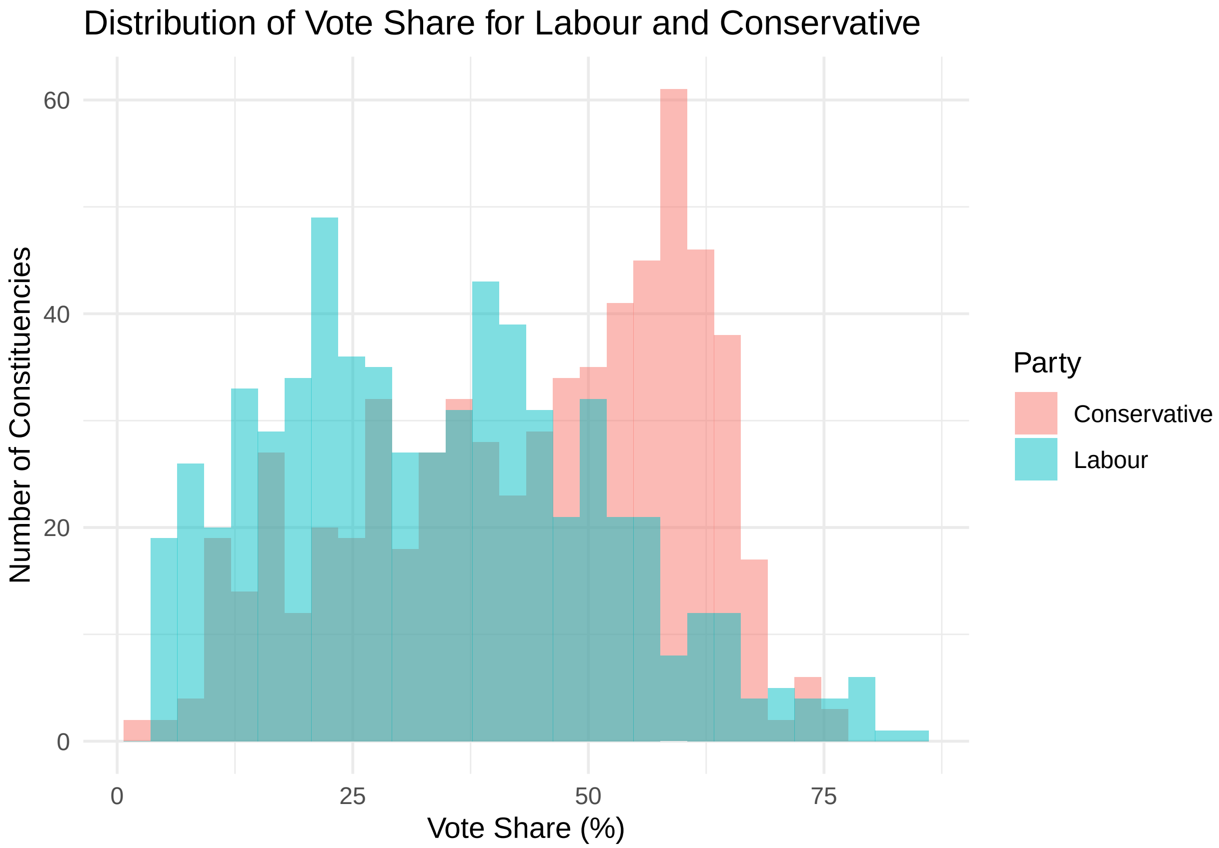
```r
library(ggplot2)
largest_constituencies <- ukvote2019 %>%
  group_by(constituency) %>%
  summarise(electorate = max(electorate)) %>%
  arrange(desc(electorate)) %>%
  slice(1:5)
largest_constituency_names <- largest_constituencies$constituency
plot_data <- ukvote2019 %>%
  filter(constituency %in% largest_constituency_names,
         party_name %in% c("Labour", "Conservative"))
ggplot(plot_data, aes(x = constituency, y = vote_share_percent, fill = party_name)) +
  geom_col(position = "dodge") +  # dodge = side by side
  labs(title = "Vote Share of Labour and Conservative in 5 Largest Constituencies",
       x = "Constituency",
       y = "Vote Share (%)",
       fill = "Party") +
  theme_minimal()
```



```
ggsave(here("plot12.png"), plot = last_plot(), width = 8, height = 6, dpi = 300)
```

13. Build a graph showing how the votes share for the Labour and for the Conservative parties are distributed. (1pt)

```
plot_data_2 <- ukvote2019 %>%
  filter(party_name %in% c("Labour", "Conservative"))
ggplot(plot_data_2, aes(x = vote_share_percent, fill = party_name)) +
  geom_histogram(position = "identity", alpha = 0.5, bins = 30) +
  labs(title = "Distribution of Vote Share for Labour and Conservative",
       x = "Vote Share (%)",
       y = "Number of Constituencies",
       fill = "Party") +
  theme_minimal()
```

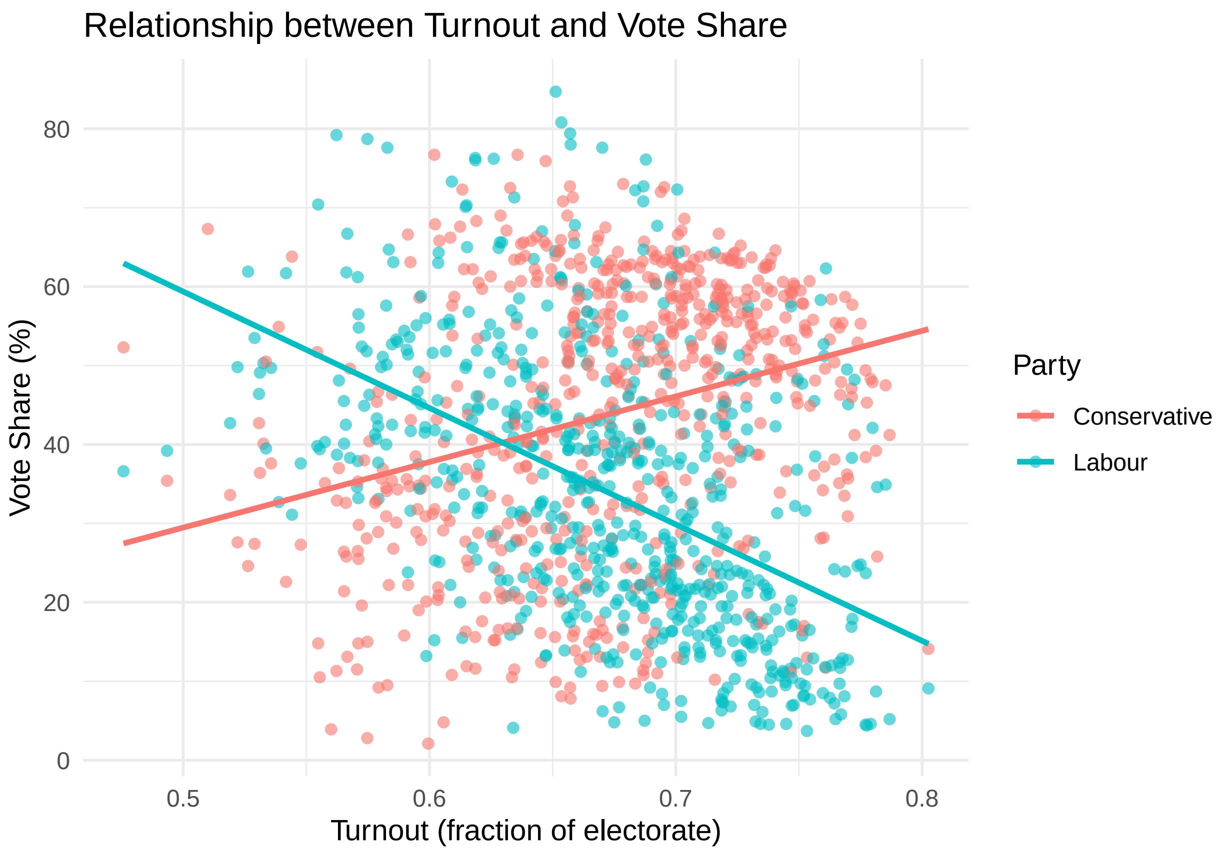


```
ggsave(here("plot13.png"), plot = last_plot(), width = 8, height = 6, dpi = 300)
```

14. Based on a graph, illustrate how turnout is (differentially) correlated with votes share for the Conservative and the Labour party. Briefly comment. (1.25 pt)

```
ggplot(plot_data_2, aes(x = turnout, y = vote_share_percent, color = party_name)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Relationship between Turnout and Vote Share",
       x = "Turnout (fraction of electorate)",
       y = "Vote Share (%)",
       color = "Party") +
  theme_minimal()
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
ggsave(here("plot14.png"), plot = last_plot(), width = 8, height = 6, dpi = 300)
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

more turnout => more conservatives, less turnout => more laborists we can say that mobilization dynamics may favor the conservatives when more of the electorate participates

15. Find the 5 constituencies in which the combined vote share of the Green and UKIP is the highest (Hint: you may need pivoting functions) (1.5 pts)

```
combined = ukvote2019 %>%
  filter(str_detect(party_name, "Green") |
           str_detect(party_name, "UKIP") |
           str_detect(party_name, "UK Independence")) %>%
  group_by(constituency) %>%
  summarise(combined_share = sum(vote_share_percent, na.rm = TRUE)) %>%
  arrange(desc(combined_share)) %>%
  slice(1:5)

combined
```

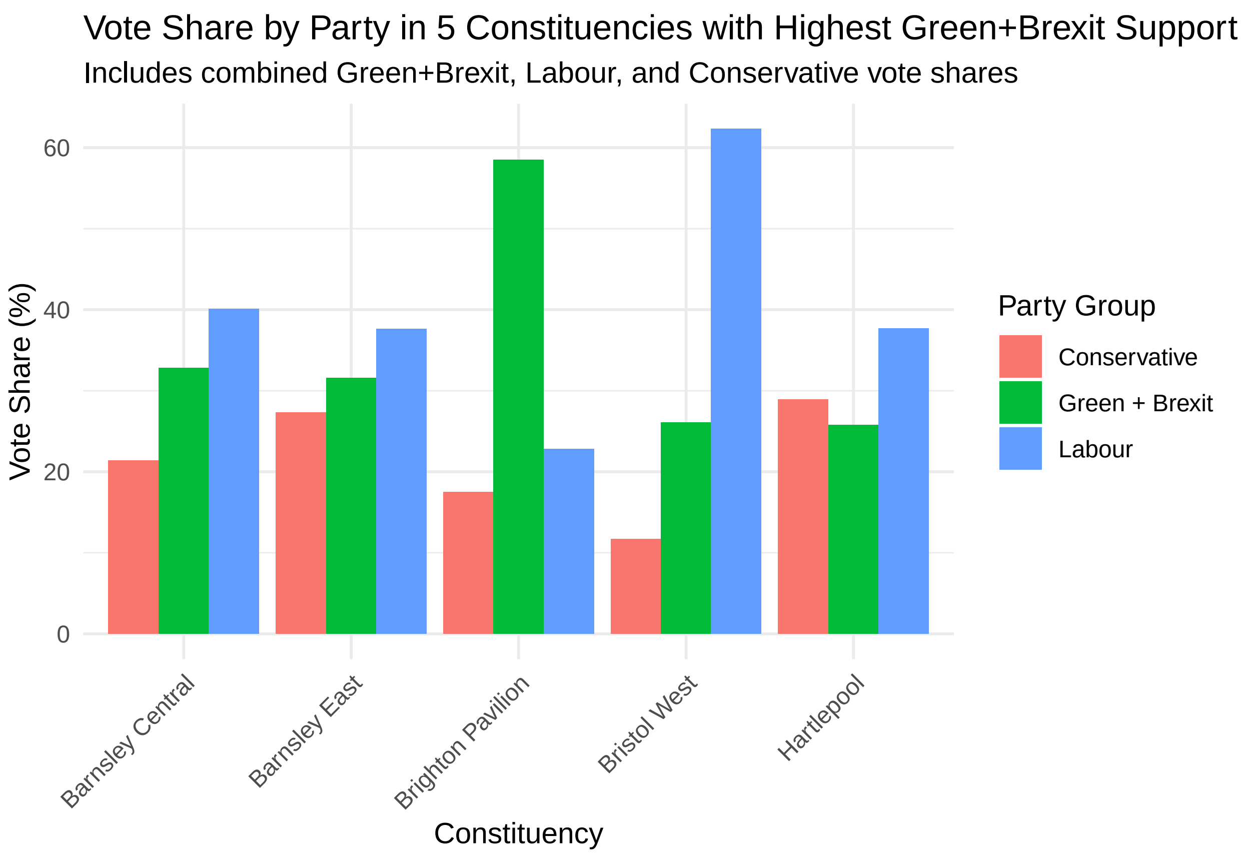
```
## # A tibble: 5 x 2
##   constituency           combined_share
##   <chr>                           <dbl>
## 1 Brighton Pavilion                57.5
## 2 Bristol West                     24.9
## 3 Dulwich & West Norwood           16.6
## 4 Bury St Edmunds                  15.7
## 5 Isle of Wight                    15.2
```

16. Build a graph showing, in each of these 5 constituencies, i) the combined vote share of the Green and Brexit Party, ii) the vote share of the Labour, iii) the vote share of the Convervative. (1.5 pts)

```
top5_constituencies <- ukvote2019 %>%
  filter(str_detect(party_name, "Green") | str_detect(party_name, "Brexit")) %>%
  group_by(constituency) %>%
  summarise(green_brexit_share = sum(vote_share_percent, na.rm = TRUE)) %>%
  arrange(desc(green_brexit_share)) %>%
  slice(1:5) %>%
  pull(constituency)

plot_data <- ukvote2019 %>%
  filter(
    constituency %in% top5_constituencies &
      (str_detect(party_name, "Green") |
         str_detect(party_name, "Brexit") |
         party_name %in% c("Labour", "Conservative"))
  ) %>%
  mutate(group = case_when(
    str_detect(party_name, "Green") | str_detect(party_name, "Brexit") ~ "Green + Brexit",
    TRUE ~ party_name
  )) %>%
  group_by(constituency, group) %>%
  summarise(vote_share = sum(vote_share_percent, na.rm = TRUE)) %>%
  ungroup()
```

```
## ‘summarise()‘ has grouped output by ’constituency’. You can override using the
## ‘.groups‘ argument.
```

```
ggplot(plot_data, aes(x = constituency, y = vote_share, fill = group)) +
  geom_col(position = "dodge") +
  labs(
    title = "Vote Share by Party in 5 Constituencies with Highest Green+Brexit Support",
    subtitle = "Includes combined Green+Brexit, Labour, and Conservative vote shares",
    x = "Constituency",
    y = "Vote Share (%)",
    fill = "Party Group"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
ggsave(here("plot16.png"), plot = last_plot(), width = 8, height = 6, dpi = 300, bg = "white")
```

17. In how many constituencies was the combined vote share of the Green and UKIP less than 2%? (1 pt)

```
ukvote2019 %>%
  filter(str_detect(party_name, "Green") |
           str_detect(party_name, "UKIP") |
           str_detect(party_name, "UK Independence")) %>%
  group_by(constituency) %>%
  summarise(combined = sum(vote_share_percent, na.rm = TRUE)) %>%
  filter(combined < 2) %>%
  summarise(count = n())
```

```
## # A tibble: 1 x 1
##   count
##   <int>
## 1    86
```

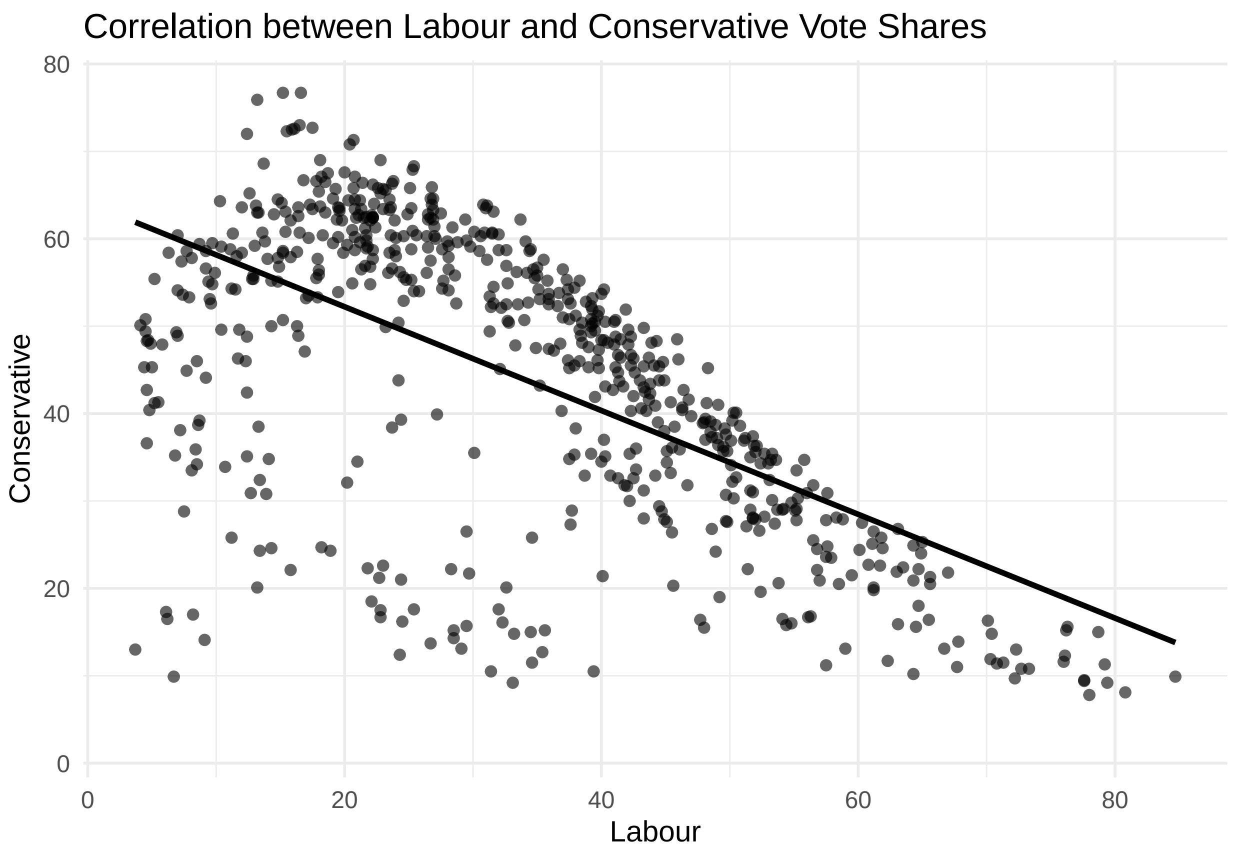
18. Plot the vote share of the Labour party against that of the Conservative party. How are the two correlated? (1.5 pts)

```
shares = ukvote2019 %>%
  filter(str_detect(party_name, "Labour") | str_detect(party_name, "Conservative")) %>%
  select(constituency, party_name, vote_share_percent) %>%
  pivot_wider(names_from = party_name, values_from = vote_share_percent)
colnames(shares)
```

```
## [1] "constituency"                      "Labour"
## [3] "Conservative"                      "Social Democratic & Labour Party"
## [5] "Socialist Labour Party"
```

```
ggplot(shares, aes(x = shares[[2]], y = shares[[3]])) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  labs(title = "Correlation between Labour and Conservative Vote Shares",
       x = names(shares)[2],
       y = names(shares)[3]) +
  theme_minimal()
```

```
## Warning: Use of `shares[[2]]` is discouraged.
## i Use `.data[[2]]` instead.

## Warning: Use of `shares[[3]]` is discouraged.
## i Use `.data[[3]]` instead.

## Warning: Use of `shares[[2]]` is discouraged.
## i Use `.data[[2]]` instead.

## Warning: Use of `shares[[3]]` is discouraged.
## i Use `.data[[3]]` instead.

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 17 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 17 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
cor(shares[[2]], shares[[3]], use = "complete.obs")
```

```
## [1] -0.6207029
```

```
ggsave(here("plot18.png"), plot = last_plot(), width = 8, height = 6, dpi = 300, bg = "white")
```

```
## Warning: Use of ‘shares[[2]]‘ is discouraged.
## i Use ‘.data[[2]]‘ instead.
```

```
## Warning: Use of ‘shares[[3]]‘ is discouraged.
## i Use ‘.data[[3]]‘ instead.
```

```
## Warning: Use of ‘shares[[2]]‘ is discouraged.
## i Use ‘.data[[2]]‘ instead.
```

```
## Warning: Use of ‘shares[[3]]‘ is discouraged.
## i Use ‘.data[[3]]‘ instead.
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 17 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
## Warning: Removed 17 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

interpretation: The plot shows a clear negative correlation between the Labour and Conservative vote shares, as constituencies with a higher Conservative share tend to have a lower Labour share and vice versa.

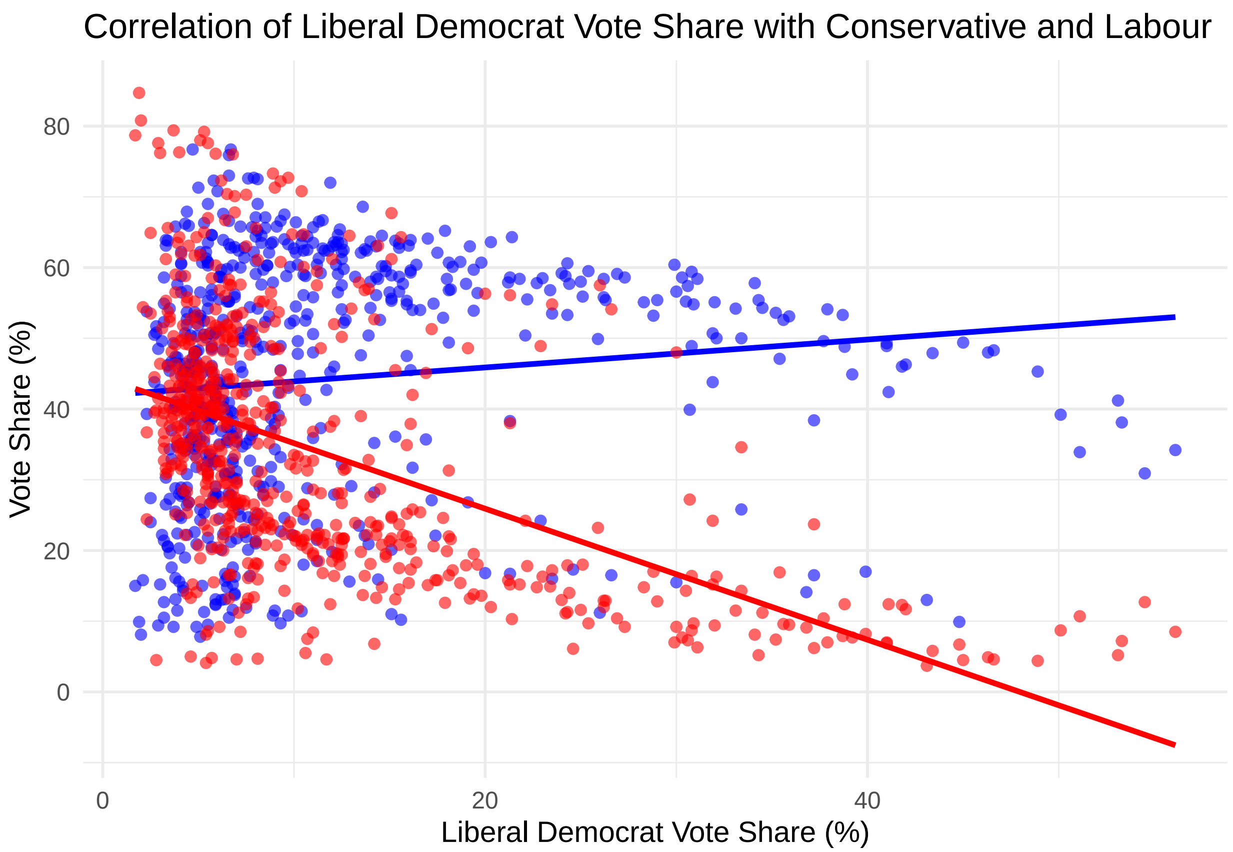
19. Create a new data.frame excluding all constituencies in which the Liberal Democrat received 0% of the votes share. Based on this new data.frame, build a single graph showing

i) the correlation between the votes share of the Liberal Democrat and of the Conservative in blue and ii) the correlation between the votes share of the Liberal Democrat and of the Labour party in red. Add a legend. Briefly comment the resulting graph. (Hint: if a variable name has a space in it, you can refer to it in R using backtick symbols, e.g. ’The Independent‘) (2.5 pts)

```r
shares_ld = ukvote2019 %>%
  filter(str_detect(party_name, "Labour") | str_detect(party_name, "Conservative") | str_detect(party_na
  select(constituency, party_name, vote_share_percent) %>%
  pivot_wider(names_from = party_name, values_from = vote_share_percent) %>%
  filter(`Liberal Democrat` > 0)

ggplot(shares_ld) +
  geom_point(aes(x = `Liberal Democrat`, y = Conservative), color = "blue", alpha = 0.6) +
  geom_smooth(aes(x = `Liberal Democrat`, y = Conservative), method = "lm", se = FALSE, color = "blue")
  geom_point(aes(x = `Liberal Democrat`, y = Labour), color = "red", alpha = 0.6) +
  geom_smooth(aes(x = `Liberal Democrat`, y = Labour), method = "lm", se = FALSE, color = "red") +
  labs(title = "Correlation of Liberal Democrat Vote Share with Conservative and Labour",
       x = "Liberal Democrat Vote Share (%)",
       y = "Vote Share (%)") +
  theme_minimal() +
  scale_color_manual(values = c("Conservative" = "blue", "Labour" = "red")) +
  guides(color = guide_legend(title = "Party"))
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s colour values.
```



```
ggsave(here("plot19.png"), plot = last_plot(), width = 8, height = 6, dpi = 300)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s colour values.
```

The plot indicates a negative correlation between the Liberal Democrat and Labour vote shares, as evidenced by the downward-sloping red trend line. In contrast, the Liberal Democrat vote share shows little correlation with the Conservative vote share, with the blue trend line remaining largely horizontal.

20. Save all your graphs using the “here()” function. (1 pt) Answer: already saved after each question