My title*

My subtitle if needed

First author Another author

November 1, 2024

First sentence. Second sentence. Third sentence. Fourth sentence.

1 Introduction

2 Data

2.1 Overview

2.2 Raw data

```r
names(raw_data)
```

```
 [1] "poll_id"                   "pollster_id"
 [3] "pollster"                  "sponsor_ids"
 [5] "sponsors"                  "display_name"
 [7] "pollster_rating_id"        "pollster_rating_name"
 [9] "numeric_grade"             "pollscore"
[11] "methodology"               "transparency_score"
[13] "state"                     "start_date"
[15] "end_date"                  "sponsor_candidate_id"
[17] "sponsor_candidate"         "sponsor_candidate_party"
[19] "endorsed_candidate_id"     "endorsed_candidate_name"
[21] "endorsed_candidate_party"  "question_id"
[23] "sample_size"               "population"
```

[*] Code and data are available at: https://github.com/RohanAlexander/starter_folder.

```
[25] "subpopulation"                "population_full"
[27] "tracking"                     "created_at"
[29] "notes"                        "url"
[31] "url_article"                  "url_topline"
[33] "url_crosstab"                 "source"
[35] "internal"                     "partisan"
[37] "race_id"                      "cycle"
[39] "office_type"                  "seat_number"
[41] "seat_name"                    "election_date"
[43] "stage"                        "nationwide_batch"
[45] "ranked_choice_reallocated" "ranked_choice_round"
[47] "hypothetical"                 "party"
[49] "answer"                       "candidate_id"
[51] "candidate_name"               "pct"
```

Raw data 52 variable 17133 sample 52 varibles project

```
#    table
del_1 <- c("notes", "url", "url_article", "url_topline", "url_crosstab", "source")
droped_data <- raw_data %>% select(-any_of(del_1))
del_1
```

```
[1] "notes"         "url"           "url_article"   "url_topline"   "url_crosstab"
[6] "source"
```

variables

```
#    table
del_2 <- c("pollster", "sponsors", "display_name", "pollster_rating_name", "sponsor_candidat
           "population_full", "candidate_id", "candidate_name")
droped_data <- droped_data %>% select(-any_of(del_2))
del_2
```

```
[1] "pollster"                "sponsors"
[3] "display_name"            "pollster_rating_name"
[5] "sponsor_candidate"       "endorsed_candidate_name"
[7] "population_full"         "candidate_id"
[9] "candidate_name"
```

variables

```r
#     table
del_3 <- c("office_type", "seat_number", "seat_name")
droped_data <- droped_data %>% select(-any_of(del_3))
del_3
```

```
[1] "office_type" "seat_number" "seat_name"
```

```r
# Identify variables where all values are the same (including NA)
same_value_variables <- names(droped_data)[sapply(droped_data, function(x) length(unique(x))
# Create a data frame with variables and their unique values
same_value_data <- data.frame(Variable = same_value_variables, Value = sapply(same_value_vari
unique(na.omit(droped_data[[var]]))[1]))
# Print the names of variables with all identical values using kable
kable(same_value_data[, 2, drop = FALSE], col.names = c("Variable", "Value"))
```

Variable	Value
endorsed_candidate_id	NA
endorsed_candidate_party	NA
subpopulation	NA
cycle	2024
election_date	11/5/24
stage	general
nationwide_batch	FALSE

```r
del_4 <- c("endorsed_candidate_id", "endorsed_candidate_party", "subpopulation", "cycle", "el
droped_data <- droped_data %>% select(-any_of(del_4))
```

categorical

```r
catego <- c("poll_id", "pollster_id", "sponsor_ids", "pollster_rating_id", "methodology", "s
         "sponsor_candidate_id", "sponsor_candidate_party", "question_id", "population", "tra
         "partisan","race_id", "ranked_choice_reallocated", "ranked_choice_round", "hypotheti
catego
```

```
 [1] "poll_id"                    "pollster_id"
 [3] "sponsor_ids"                "pollster_rating_id"
 [5] "methodology"              "state"
 [7] "sponsor_candidate_id"     "sponsor_candidate_party"
 [9] "question_id"              "population"
[11] "tracking"                 "created_at"
[13] "internal"                 "partisan"
[15] "race_id"                  "ranked_choice_reallocated"
[17] "ranked_choice_round"      "hypothetical"
[19] "party"                    "answer"
```

categorical appendix

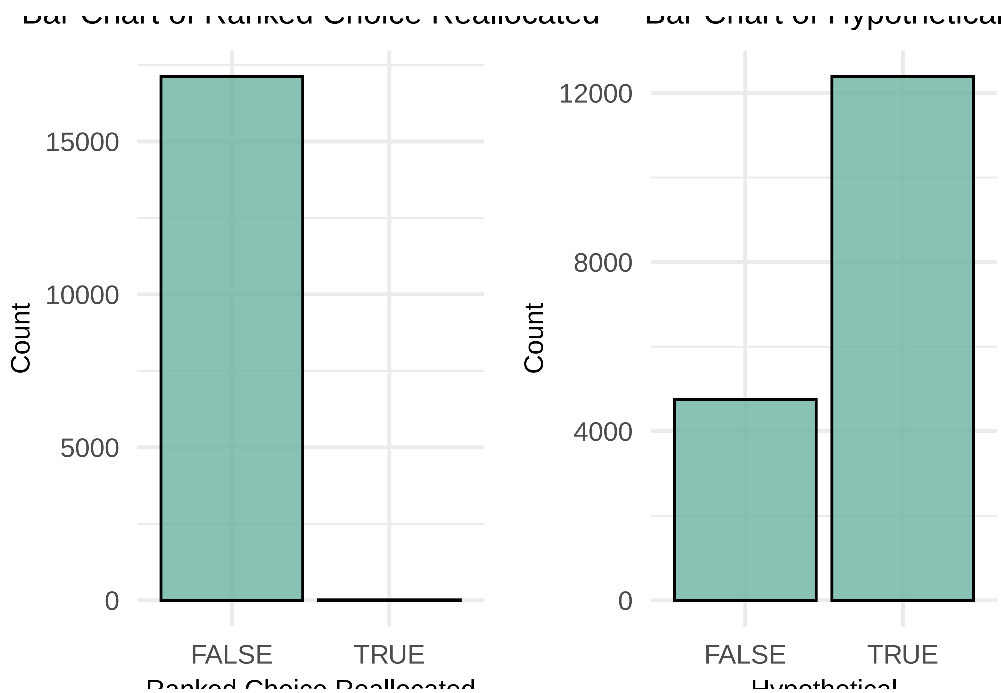
```r
catego_inp <- c("poll_id", "methodology", "population", "ranked_choice_reallocated", "hypothe
catego_inp
```

```
[1] "poll_id"                   "methodology"
[3] "population"                "ranked_choice_reallocated"
[5] "hypothetical"              "answer"
```

3530 poll

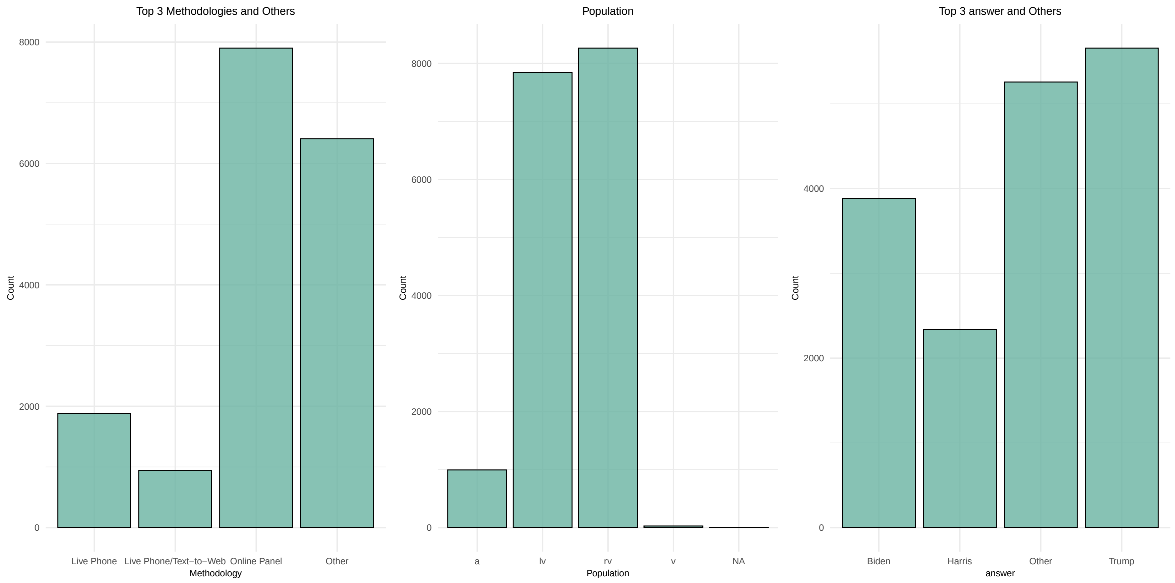
```
[1] 3530
```

```r
plots <- list()
# Create bar charts for 'ranked_choice_reallocated' and 'hypothetical'
plots[["ranked_choice"]] <- ggplot(raw_data, aes(x = ranked_choice_reallocated)) +
  geom_bar(fill = "#69b3a2", color = "black", alpha = 0.8) +
  labs(title = "Bar Chart of Ranked Choice Reallocated", x = "Ranked Choice Reallocated", y =
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10)
  )

plots[["hypothetical"]] <- ggplot(raw_data, aes(x = hypothetical)) +
  geom_bar(fill = "#69b3a2", color = "black", alpha = 0.8) +
  labs(title = "Bar Chart of Hypothetical", x = "Hypothetical", y = "Count") +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10)
  )

# Combine and print the plots using patchwork
combined_plot <- wrap_plots(plots, ncol = 2, nrow = 1, heights = unit(rep(3, 4), "in")) +
  plot_annotation(
    theme = theme(
      plot.title = element_text(hjust = 0.5, size = 14)
    )
  )
# Print the combined plot
print(combined_plot)
```



```
plots <- list()
# Find the top 3 most frequent values in 'methodology' and group others as 'Other'
methodology_counts <- sort(table(raw_data$methodology), decreasing = TRUE)
top_3_methodologies <- names(methodology_counts)[1:3]
raw_data$methodology_grouped <- ifelse(raw_data$methodology %in% top_3_methodologies, raw_da
```

```r
# Create a bar chart for the grouped 'methodology'
plots[["methodology"]] <- ggplot(raw_data, aes(x = methodology_grouped)) +
  geom_bar(fill = "#69b3a2", color = "black", alpha = 0.8) +
  labs(title = "Top 3 Methodologies and Others", x = "Methodology", y = "Count") +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10)
  )
plots[["population"]] <- ggplot(raw_data, aes(x = population)) +
  geom_bar(fill = "#69b3a2", color = "black", alpha = 0.8) +
  labs(title = "Population", x = "Population", y = "Count") +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10)
  )
# Find the top 3 most frequent values in 'answer' and group others as 'Other'
answer_counts <- sort(table(raw_data$answer), decreasing = TRUE)
top_3_answer <- names(answer_counts)[1:3]
raw_data$answer <- ifelse(raw_data$answer %in% top_3_answer, raw_data$answer, "Other")
# Create a bar chart for the grouped 'answer'
plots[["answer"]] <- ggplot(raw_data, aes(x = answer)) +
  geom_bar(fill = "#69b3a2", color = "black", alpha = 0.8) +
  labs(title = "Top 3 answer and Others", x = "answer", y = "Count") +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 10)
  )
combined_plot <- wrap_plots(plots, ncol = 3, nrow = 1, heights = unit(rep(8, 4), "in")) +
  plot_annotation(
    theme = theme(
      plot.title = element_text(hjust = 0.5, size = 14)
    )
  )
# Print the combined plot
print(combined_plot)
```



numerical variables

```
numer <- droped_data %>% select(-any_of(catego))
names(numer)
```

```
[1] "numeric_grade"      "pollscore"           "transparency_score"
[4] "start_date"         "end_date"            "sample_size"
[7] "pct"
```

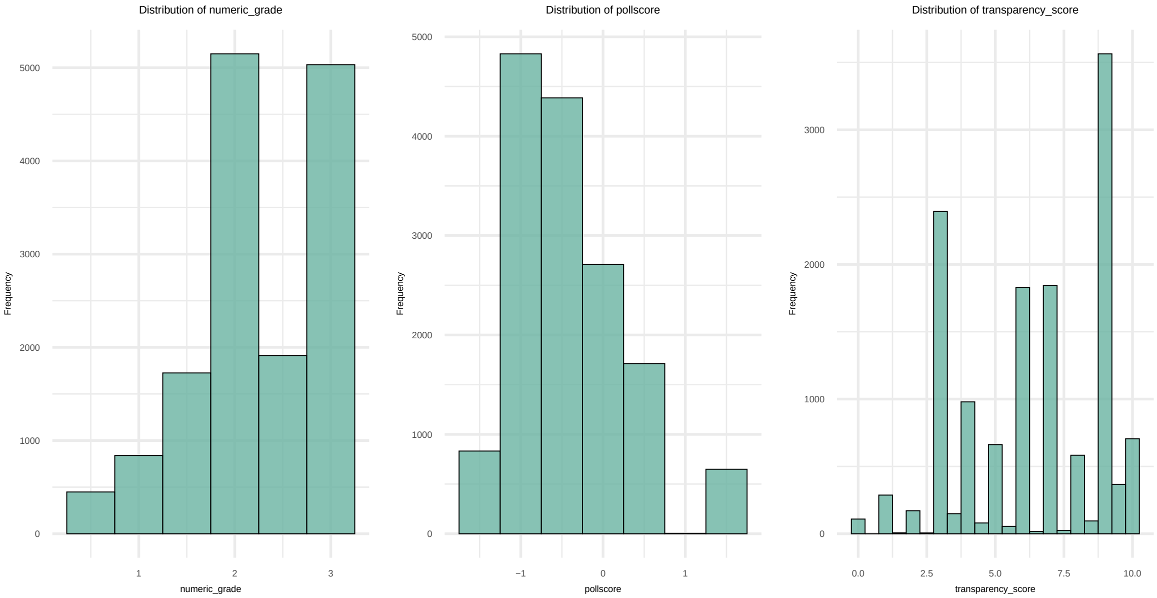


Figure 1: Distribution of numerical varibales

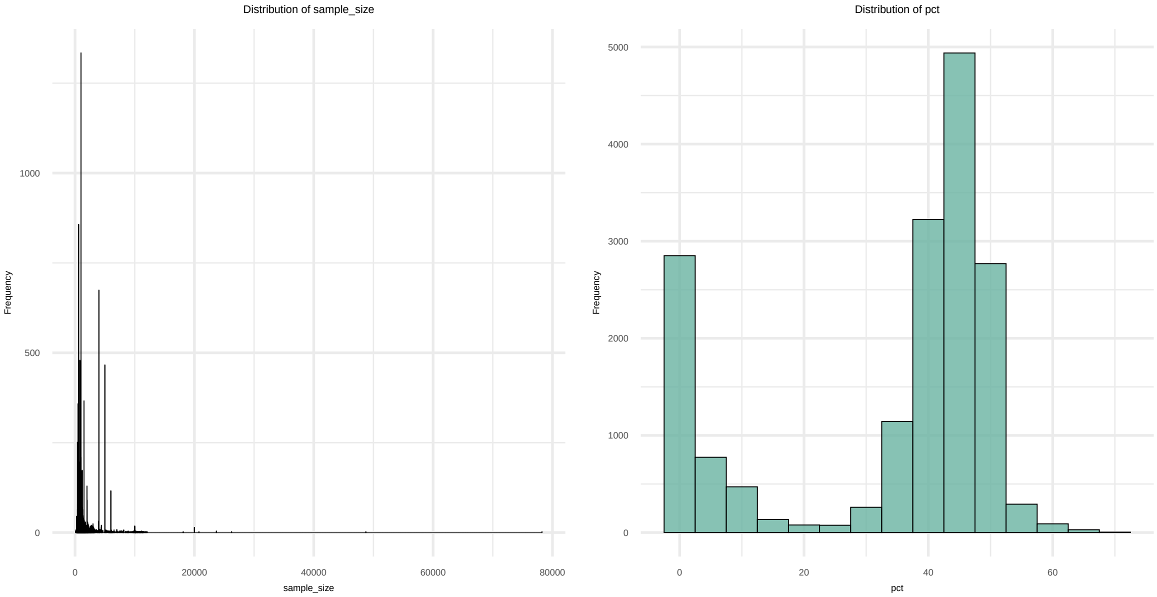


Figure 2: Distribution of numerical varibales

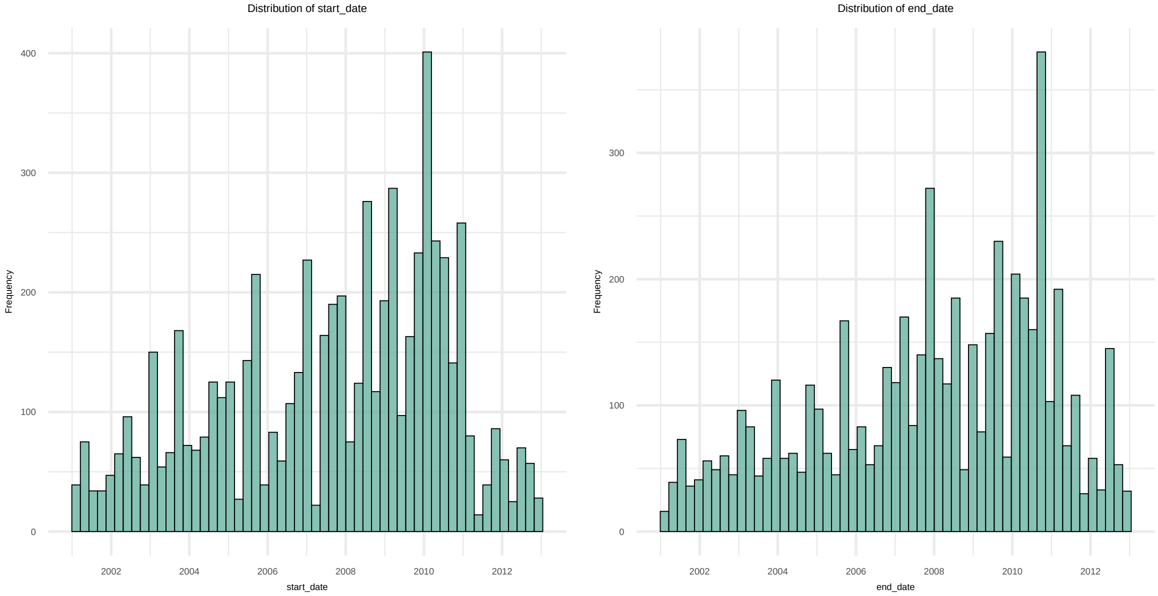


Figure 3: Distribution of date varibales

2.3 Cleaned data

raw data variables project

```
names(droped_data)
```

```
 [1] "poll_id"                    "pollster_id"
 [3] "sponsor_ids"                "pollster_rating_id"
 [5] "numeric_grade"              "pollscore"
 [7] "methodology"                "transparency_score"
 [9] "state"                      "start_date"
[11] "end_date"                   "sponsor_candidate_id"
[13] "sponsor_candidate_party"    "question_id"
[15] "sample_size"                "population"
[17] "tracking"                   "created_at"
[19] "internal"                   "partisan"
[21] "race_id"                    "ranked_choice_reallocated"
[23] "ranked_choice_round"        "hypothetical"
[25] "party"                      "answer"
[27] "pct"
```

NA 40%

```r
na_proportions <- sapply(names(droped_data), function(var) {
  round(mean(is.na(raw_data[[var]])), 2)
})

# Create a data frame with variable names and their NA proportions
na_proportions_data <- data.frame(Variable = names(droped_data), NA_Proportion = na_proporti

# Print the NA proportions using kable
kable(na_proportions_data[, 2, drop = FALSE], col.names = c("NA Proportion"))
```

	NA Proportion
poll_id	0.00
pollster_id	0.00
sponsor_ids	0.52
pollster_rating_id	0.00
numeric_grade	0.12
pollscore	0.12
methodology	0.06
transparency_score	0.19
state	0.46
start_date	0.63
end_date	0.68
sponsor_candidate_id	0.98
sponsor_candidate_party	0.98

	NA Proportion
question_id	0.00
sample_size	0.01
population	0.00
tracking	0.91
created_at	0.00
internal	0.85
partisan	0.92
race_id	0.00
ranked_choice_reallocated	0.00
ranked_choice_round	1.00
hypothetical	0.00
party	0.00
answer	0.00
pct	0.00

```r
# Filter variables with NA proportion greater than 40%
high_na_proportions <- na_proportions_data[na_proportions_data$NA_Proportion > 0.4, ]

# Print the NA proportions greater than 40% using kable
kable(high_na_proportions[, 2, drop = FALSE], col.names = c("Variable", "NA Proportion"))
```

Variable	NA Proportion
sponsor_ids	0.52
state	0.46
start_date	0.63
end_date	0.68
sponsor_candidate_id	0.98
sponsor_candidate_party	0.98
tracking	0.91
internal	0.85
partisan	0.92
ranked_choice_round	1.00

```r
del_5 <- c("sponsor_ids", "state", "sponsor_candidate_id", "sponsor_candidate_party", "tracki
           "internal","partisan","ranked_choice_round")
droped_data <- droped_data %>% select(-any_of(del_5))
names(droped_data)
```

```
 [1] "poll_id"                   "pollster_id"
 [3] "pollster_rating_id"        "numeric_grade"
 [5] "pollscore"                 "methodology"
 [7] "transparency_score"        "start_date"
 [9] "end_date"                  "question_id"
[11] "sample_size"               "population"
[13] "created_at"                "race_id"
[15] "ranked_choice_reallocated" "hypothetical"
[17] "party"                     "answer"
[19] "pct"
```

variables

```
final <- c("poll_id","numeric_grade","pollscore","methodology","transparency_score","sample_s
           "ranked_choice_reallocated", "hypothetical", "answer","pct","start_date", "end_da
final
```

```
 [1] "poll_id"                   "numeric_grade"
 [3] "pollscore"                 "methodology"
 [5] "transparency_score"        "sample_size"
 [7] "population"                "ranked_choice_reallocated"
 [9] "hypothetical"              "answer"
[11] "pct"                       "start_date"
[13] "end_date"
```

Create a new variable called 'duration' (days difference between start_date and end_date) and remove 'start_date' and 'end_date' Group methodology by level Replace NA values - numerical variables with mean, categorical variables with mode Rename pct as score - janitor::clean_names candidate score score *poll

```
# A tibble: 5 x 3
  candidate_name    poll_count avg_weighted_pct
  <chr>                  <int>            <dbl>
1 Donald Trump            5657          252424.
2 Joe Biden               3883          161611.
3 Kamala Harris           2336          109502.
4 Ron DeSantis             466           18823.
5 Robert F. Kennedy       1330           14750.
```

train 70% ,test(30%) analysisdata

```r
kable(head(train_Trump), col.names = names(train_Trump))
```

numeric_grade	pollscore	methodology	transparency_score	sample_size	population	ranked_choice_reallocated	hypothetical	score	duration
2.7	-0.8	level1	6	1373	lv	FALSE	FALSE	50.7	1
2.7	-0.8	level1	6	1373	lv	FALSE	FALSE	50.7	1
2.7	-0.8	level1	6	1005	lv	FALSE	FALSE	51.0	1
2.7	-0.8	level1	6	1212	lv	FALSE	FALSE	48.8	1
2.7	-0.8	level1	6	1212	lv	FALSE	FALSE	50.1	1
2.7	-0.8	level1	6	1136	lv	FALSE	FALSE	49.2	1

2.4 Measurement

2.5 Similar dataset

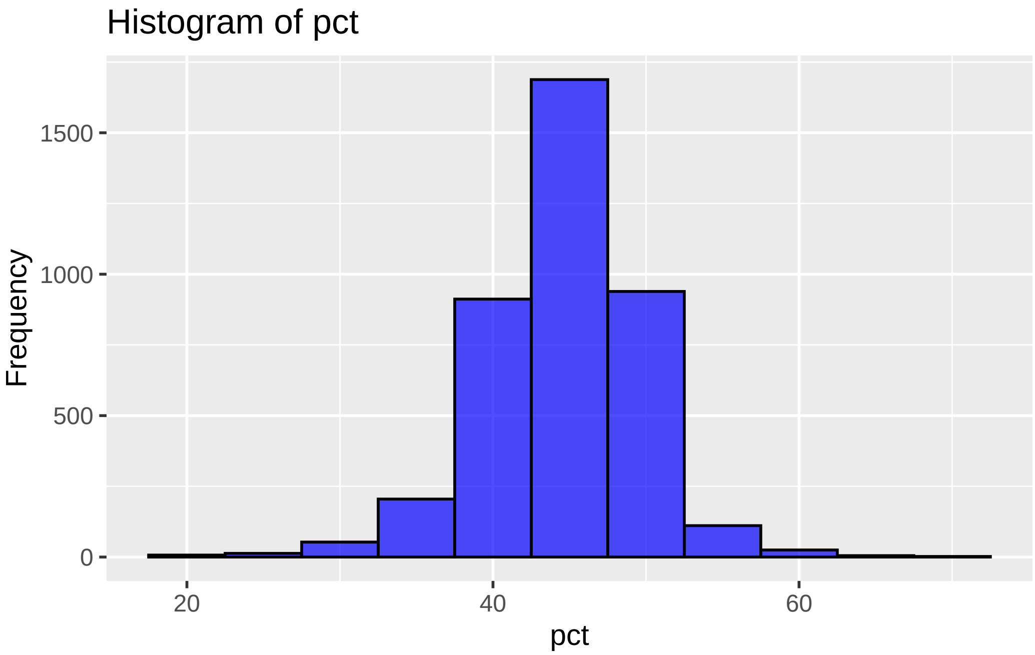
3 Model

3.1 Model set-up

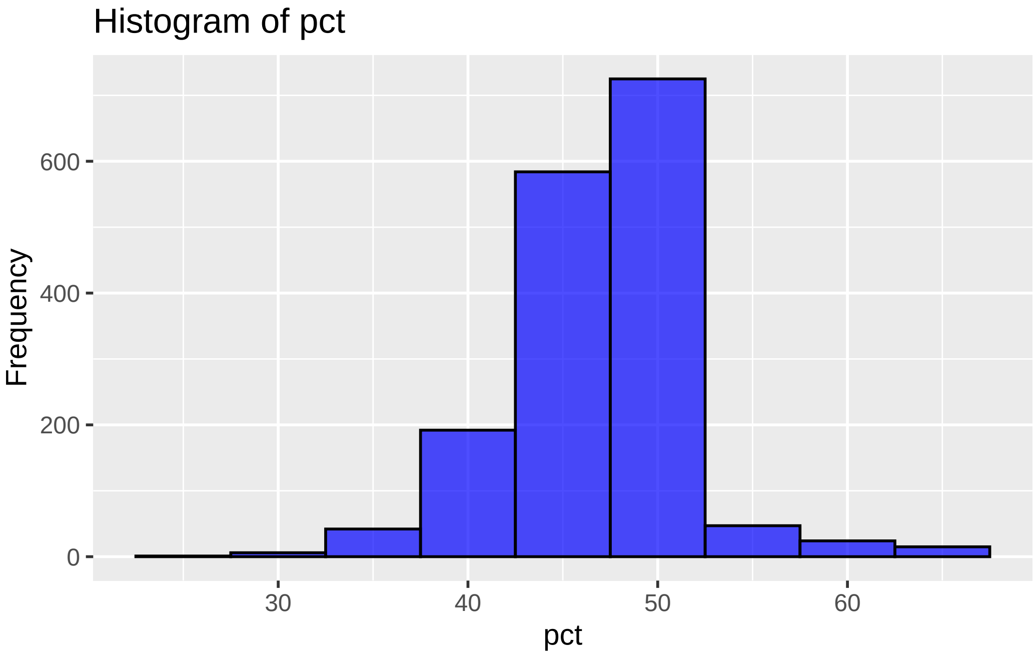
3.1.1 response variable

response variable score score

```r
# Plot a histogram for the 'pct' variable
ggplot(train_Trump, aes(x = score)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of pct", x = "pct", y = "Frequency")
```



```r
ggplot(train_Harris, aes(x = score)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of pct", x = "pct", y = "Frequency")
```

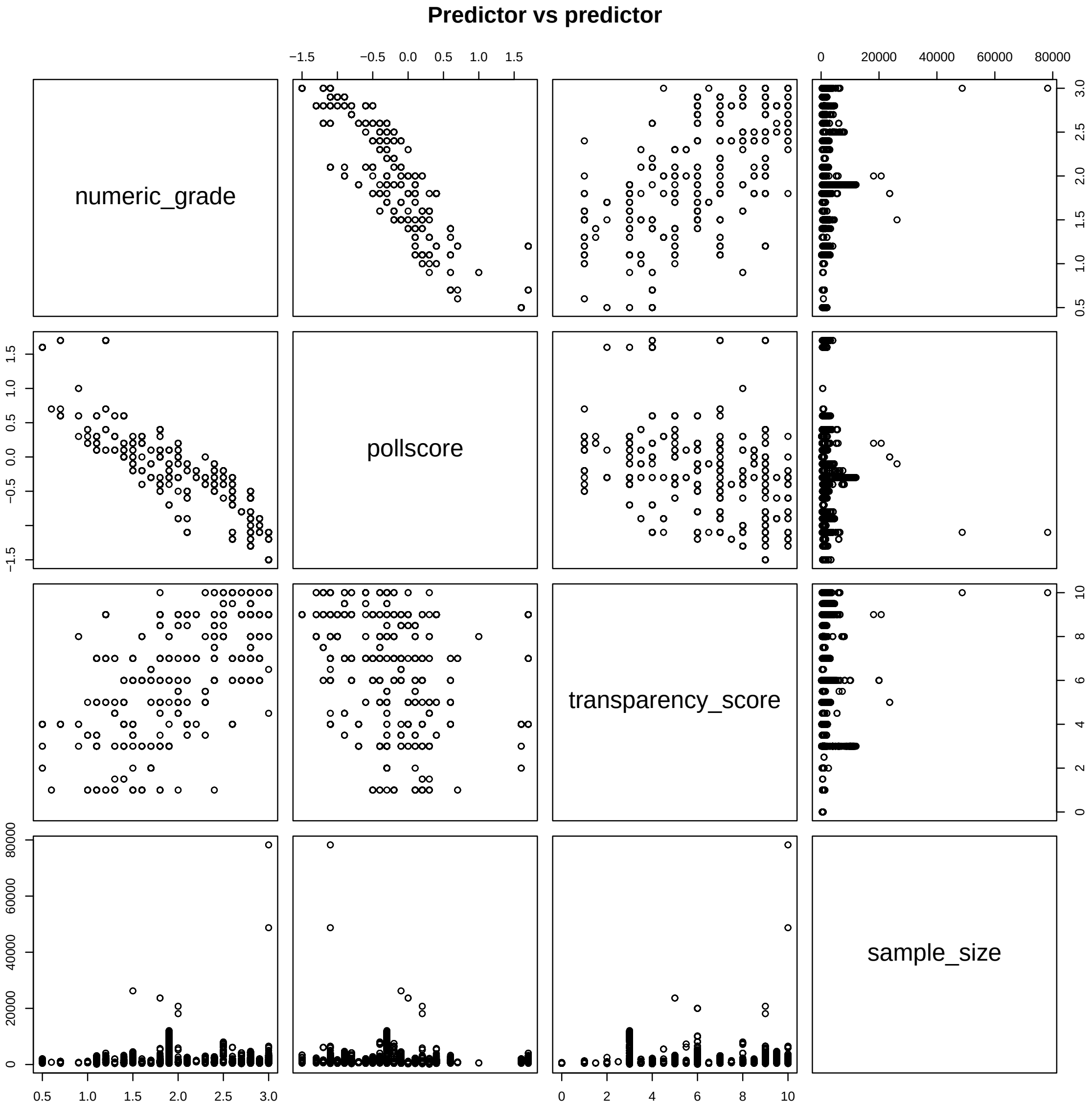


MLR

3.1.2 Predictor

train numerical variable

```
data2 = raw_data %>% select("numeric_grade", "pollscore","transparency_score","sample_size")
pairs( data2 , main = "Predictor vs predictor")
```



pollscore numerical grade numerical grade predictor

```
predictors <- c("pollscore","methodology","transparency_score","sample_size","population",
            "ranked_choice_reallocated", "hypothetical", "duration")
```

3.1.3 alternative models

```
Trump_model_1 <- lm(
  score ~ pollscore + transparency_score + duration + sample_size + population + hypothetical
    methodology, data = train_Trump)
Harris_model_1 <- lm(
  score ~ pollscore + transparency_score + duration + sample_size + population + hypothetical
    methodology, data = train_Harris)
```

```
# Get the summary of the model
Harris_summary <- summary(Harris_model_1)
# Extract coefficients from the summary
coefficients <- Harris_summary$coefficients
# Extract p-values
Harris_p_values <- coefficients[, 4]
# Create a data frame with the results
Harris_results_table <- data.frame(
  Variable = rownames(coefficients),
  P_Value = format(Harris_p_values, scientific = TRUE)
)
Harris_kable <- kable(Harris_results_table[, 2, drop = FALSE], col.names = c("Variable", "P-v
# Get the summary of the model
Trump_summary <- summary(Trump_model_1)
# Extract coefficients from the summary
coefficients <- Trump_summary$coefficients
# Extract p-values
Trump_p_values <- coefficients[, 4]
# Create a data frame with the results
Trump_results_table <- data.frame(
  Variable = rownames(coefficients),
  P_Value = format(Trump_p_values, scientific = TRUE)
)
Trump_kable <- kable(Trump_results_table[, 2, drop = FALSE], col.names = c("Variable", "P-val
```

```r
knitr::kable(
  list(Trump_results_table[, 2, drop = FALSE], Harris_results_table[, 2, drop = FALSE]),
  caption = '    ',
  booktabs = TRUE, valign = 't'
)
```

```r
summary(Trump_model_1)$coefficients[, 4]
```

```
                  (Intercept)                     pollscore 
                 0.000000e+00                  1.923424e-09 
           transparency_score                      duration 
                 2.694363e-03                  3.164509e-04 
                  sample_size                  populationlv 
                 3.889702e-07                  5.329369e-45 
                 populationrv                   populationv 
                 2.162684e-30                  6.301089e-01 
             hypotheticalTRUE ranked_choice_reallocatedTRUE 
                 1.969949e-77                  3.545196e-01 
            methodologylevel2             methodologylevel3 
                 1.460160e-01                  1.095258e-01 
            methodologylevel4 
                 5.411974e-04 
```

```r
summary(Harris_model_1)$coefficients[, 4]
```

```
                  (Intercept)                     pollscore 
                 0.000000e+00                  1.571566e-17 
           transparency_score                      duration 
                 1.374640e-02                  1.782720e-15 
                  sample_size                  populationlv 
                 2.782493e-03                  1.861643e-09 
                 populationrv                   populationv 
                 2.207540e-02                  1.123945e-06 
             hypotheticalTRUE ranked_choice_reallocatedTRUE 
                 3.610001e-09                  6.050213e-01 
            methodologylevel2             methodologylevel3 
                 3.326043e-01                  7.685775e-01 
            methodologylevel4 
                 1.001406e-01 
```

Table 5:

	P_Value
(Intercept)	0.000000e+00
pollscore	1.923424e-09
transparency_score	2.694363e-03
duration	3.164509e-04
sample_size	3.889702e-07
populationlv	5.329369e-45
populationrv	2.162684e-30
populationv	6.301089e-01
hypotheticalTRUE	1.969949e-77
ranked_choice_reallocatedTRUE	3.545196e-01
methodologylevel2	1.460160e-01
methodologylevel3	1.095258e-01
methodologylevel4	5.411974e-04

	P_Value
(Intercept)	0.000000e+00
pollscore	1.571566e-17
transparency_score	1.374640e-02
duration	1.782720e-15
sample_size	2.782493e-03
populationlv	1.861643e-09
populationrv	2.207540e-02
populationv	1.123945e-06
hypotheticalTRUE	3.610001e-09
ranked_choice_reallocatedTRUE	6.050213e-01
methodologylevel2	3.326043e-01
methodologylevel3	7.685775e-01
methodologylevel4	1.001406e-01

methodology ranked_choice_reallocated

```
summary(Trump_model)
```

```

Call:
lm(formula = score ~ pollscore + transparency_score + duration + 
    sample_size + population + hypothetical, data = Trump)

Residuals:
     Min       1Q   Median       3Q      Max 
-26.0485  -1.8860   0.0366   2.1279  23.6316 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)         4.293e+01  4.370e-01  98.253  < 2e-16 ***
pollscore          -6.757e-01  1.175e-01  -5.749 9.66e-09 ***
transparency_score -1.337e-01  3.280e-02  -4.076 4.68e-05 ***
duration            5.821e-02  1.579e-02   3.687  0.00023 ***
sample_size        -1.783e-04  3.003e-05  -5.937 3.16e-09 ***
populationlv        5.059e+00  3.353e-01  15.091  < 2e-16 ***
populationrv        4.106e+00  3.309e-01  12.410  < 2e-16 ***
populationv         2.201e+00  1.463e+00   1.505  0.13239    
hypotheticalTRUE   -2.969e+00  1.600e-01 -18.551  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.502 on 3951 degrees of freedom
Multiple R-squared:  0.1803,	Adjusted R-squared:  0.1787 
F-statistic: 108.7 on 8 and 3951 DF,  p-value: < 2.2e-16
```

```
summary(Harris_model)
```

```

Call:
lm(formula = score ~ pollscore + transparency_score + duration + 
    sample_size + population + hypothetical, data = Harris)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.242  -1.769   0.435   2.008  21.526 
```

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.397e+01  6.566e-01  66.968  < 2e-16 ***
pollscore          -1.438e+00  1.720e-01  -8.364  < 2e-16 ***
transparency_score -1.318e-01  4.341e-02  -3.036 0.002433 **
duration            1.625e-01  2.008e-02   8.092 1.14e-15 ***
sample_size         1.689e-04  5.065e-05   3.334 0.000877 ***
populationlv        3.201e+00  5.247e-01   6.100 1.32e-09 ***
populationrv        1.263e+00  5.437e-01   2.323 0.020316 *
populationv         2.031e+01  4.216e+00   4.817 1.59e-06 ***
hypotheticalTRUE   -1.614e+00  2.686e-01  -6.007 2.32e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.177 on 1627 degrees of freedom
Multiple R-squared:  0.1715,    Adjusted R-squared:  0.1674
F-statistic:  42.1 on 8 and 1627 DF,  p-value: < 2.2e-16
```

3.2 validation

GVIF 1 1.3

```
vif(Trump_model)
```

```
                       GVIF Df GVIF^(1/(2*Df))
pollscore          1.101322  1        1.049439
transparency_score 1.126838  1        1.061526
duration           1.037154  1        1.018408
sample_size        1.051550  1        1.025451
population         1.203265  3        1.031320
hypothetical       1.112077  1        1.054550
```

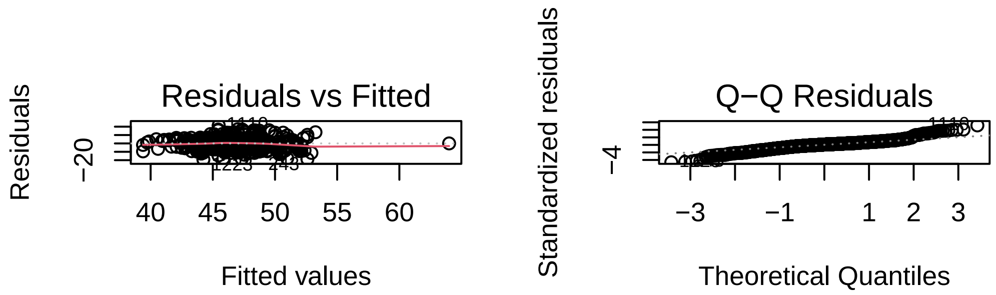
```
vif(Harris_model)
```

```
                       GVIF Df GVIF^(1/(2*Df))
pollscore          1.102432  1        1.049968
transparency_score 1.205052  1        1.097748
duration           1.087927  1        1.043037
sample_size        1.090086  1        1.044072
```

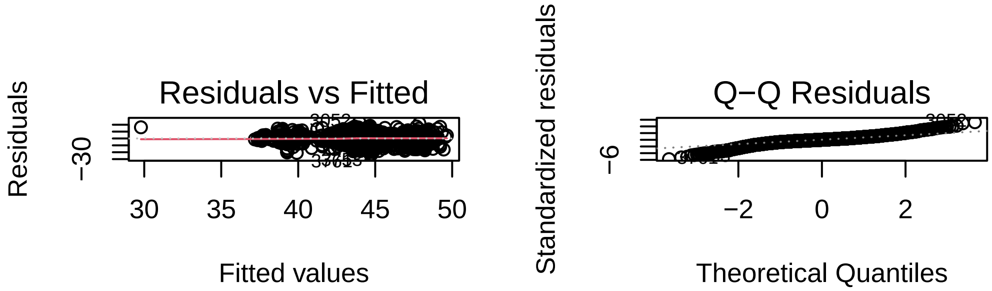
```
population          1.111258  3        1.017738
hypothetical        1.028432  1        1.014116
```

```
par(mfrow=c(2,2))
plot(Harris_model,1)
plot(Harris_model,2)
```

```
Warning: not plotting observations with leverage one:
  746
```



```
par(mfrow=c(2,2))
plot(Trump_model,1)
plot(Trump_model,2)
```



response variable normal numerical variable

MLR