Final Report

Maxim Dokukin

2024-04-19

Data Loading

```
transactions_df <- read.csv(url('https://raw.githubusercontent.com/maxdokukin/Politic
ian-Trades/main/Data/data/transactions_cleaned.csv'))
trades_df <- read.csv(url('https://raw.githubusercontent.com/maxdokukin/Politician-Tr
ades/main/Data/data/trades.csv'))
more_prices_df <- read.csv(url('https://raw.githubusercontent.com/maxdokukin/Politici
an-Trades/main/Data/data/more_prices.csv'))
```

Libraries

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(broom)
```

Data Processing

```
# Calculating stock performance before and after transaction dates

# before and after purchase average annualized rates of change
more_prices_df <- more_prices_df |>
  mutate(
    pd1 = ((Price_.60_days / Price_.90_days - 1) / 30) * 365,
    pd2 = ((Price_.40_days / Price_.60_days - 1) / 20) * 365,
    pd3 = ((Price_.20_days / Price_.40_days - 1) / 20) * 365,
    pd4 = ((Price_.10_days / Price_.20_days - 1) / 10) * 365,
    pd5 = ((Price_0_days / Price_.10_days - 1) / 10) * 365,
    pd_avg = rowMeans(cbind(pd1, pd2, pd3, pd4, pd5), na.rm = TRUE),

    fd1 = ((Price_10_days / Price_0_days - 1) / 10) * 365,
    fd2 = ((Price_20_days / Price_10_days - 1) / 10) * 365,
    fd3 = ((Price_40_days / Price_20_days - 1) / 20) * 365,
    fd4 = ((Price_60_days / Price_40_days - 1) / 20) * 365,
```

```
    fd5 = ((Price_90_days / Price_60_days - 1) / 30) * 365,
    fd_avg = rowMeans(cbind(fd1, fd2, fd3, fd4, fd5), na.rm = TRUE),

    perf_delta = fd_avg - pd_avg
  )

more_prices_df$Transaction.Type[more_prices_df$Transaction.Type == 'RECEIVE'] <- 'BU
Y'
more_prices_df <- more_prices_df[!more_prices_df$Transaction.Type == 'EXCHANGE', ]
more_prices_df <- more_prices_df[!more_prices_df$Party == 'Other', ]

# calculate market figures
market_figure <- function(row) {
  pd_avg <- as.numeric(row[33])
  fd_avg <- as.numeric(row[39])

  if (is.na(pd_avg) || is.na(fd_avg)) {
    return(NA)
  } else if (pd_avg <= 0 && fd_avg <= 0) {
    return('CTS_DOWN')
  } else if (pd_avg >= 0 && fd_avg >= 0) {
    return('CTS_UP')
  } else if (pd_avg > 0 && fd_avg < 0) {
    return('PEAK')
  } else if (pd_avg < 0 && fd_avg > 0) {
    return('DIP')
  }
}

more_prices_df$market_figure <- apply(more_prices_df, 1, market_figure)
more_prices_df <- na.omit(more_prices_df, cols = "market_figure")


# calculate success of transactions
eval_success <- function(row) {
  transaction_type <- row[11]
  market_figure <- row[41]

  if (market_figure == "CTS_DOWN" && transaction_type == "SELL") {
    return("Good Decision")
  } else if (market_figure == "CTS_DOWN" && transaction_type == "BUY") {
    return("Bad Decision")
  } else if (market_figure == "CTS_UP" && transaction_type == "SELL") {
    return("Bad Decision")
  } else if (market_figure == "CTS_UP" && transaction_type == "BUY") {
    return("Good Decision")
  } else if (market_figure == "PEAK" && transaction_type == "SELL") {
```

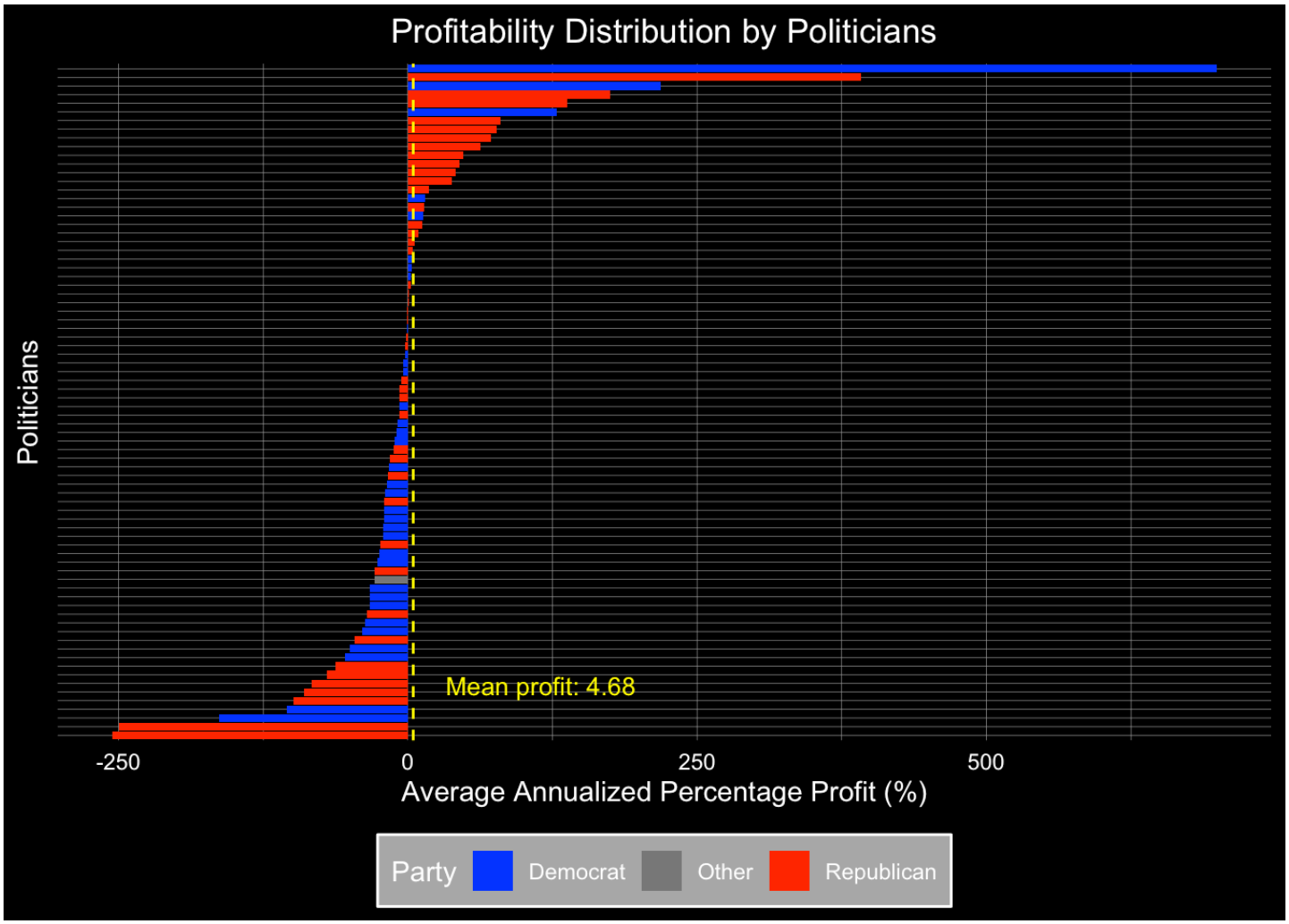
```
    return("Very Good Decision")
  } else if (market_figure == "PEAK" && transaction_type == "BUY") {
    return("Very Bad Decision")
  } else if (market_figure == "DIP" && transaction_type == "SELL") {
    return("Very Bad Decision")
  } else if (market_figure == "DIP" && transaction_type == "BUY") {
    return("Very Good Decision")
  }
}

more_prices_df$transaction_eval <- apply(more_prices_df, 1, eval_success)
```

```
# create df that contains info by politician names

general_info <- transactions_df |>
  group_by(Politician.Name) |> distinct(Politician.Name, Party, Chamber)

avg_apy_trades <- trades_df |>
  group_by(Politician.Name) |>
  summarise(
    Average_Annualized_Percentage_Profit = mean(Annualized.Percentage.Profit),
    Total_Trades = n(),
    .groups = 'drop'
  )

total_transactions <- transactions_df |>
  group_by(Politician.Name) |>
  summarise(Total_Transactions = n(), .groups = 'drop')

decisions <- more_prices_df |>
  group_by(Politician.Name, transaction_eval) |>
  summarise(Decision_Counts = n(), .groups = 'drop') |>
  pivot_wider(
    names_from = transaction_eval,
    values_from = Decision_Counts,
    values_fill = list(Decision_Counts = 0)
  ) |>
  full_join(total_transactions, by="Politician.Name") |>
  mutate(
    Proportion_of_Very_Bad_Decisions = (`Very Bad Decision` / Total_Transactions) * 1
00,
    Proportion_of_Bad_Decisions = (`Bad Decision` / Total_Transactions) * 100,
    Proportion_of_Good_Decisions = (`Good Decision` / Total_Transactions) * 100,
    Proportion_of_Very_Good_Decisions = (`Very Good Decision` / Total_Transactions) *
100
  )

# very useful for plotting
politician_info_df <- as.data.frame(
  general_info |>
  full_join(avg_apy_trades, by = "Politician.Name") |>
  full_join(decisions, by = "Politician.Name"))

rm(avg_apy_trades, decisions, general_info, total_transactions)
```

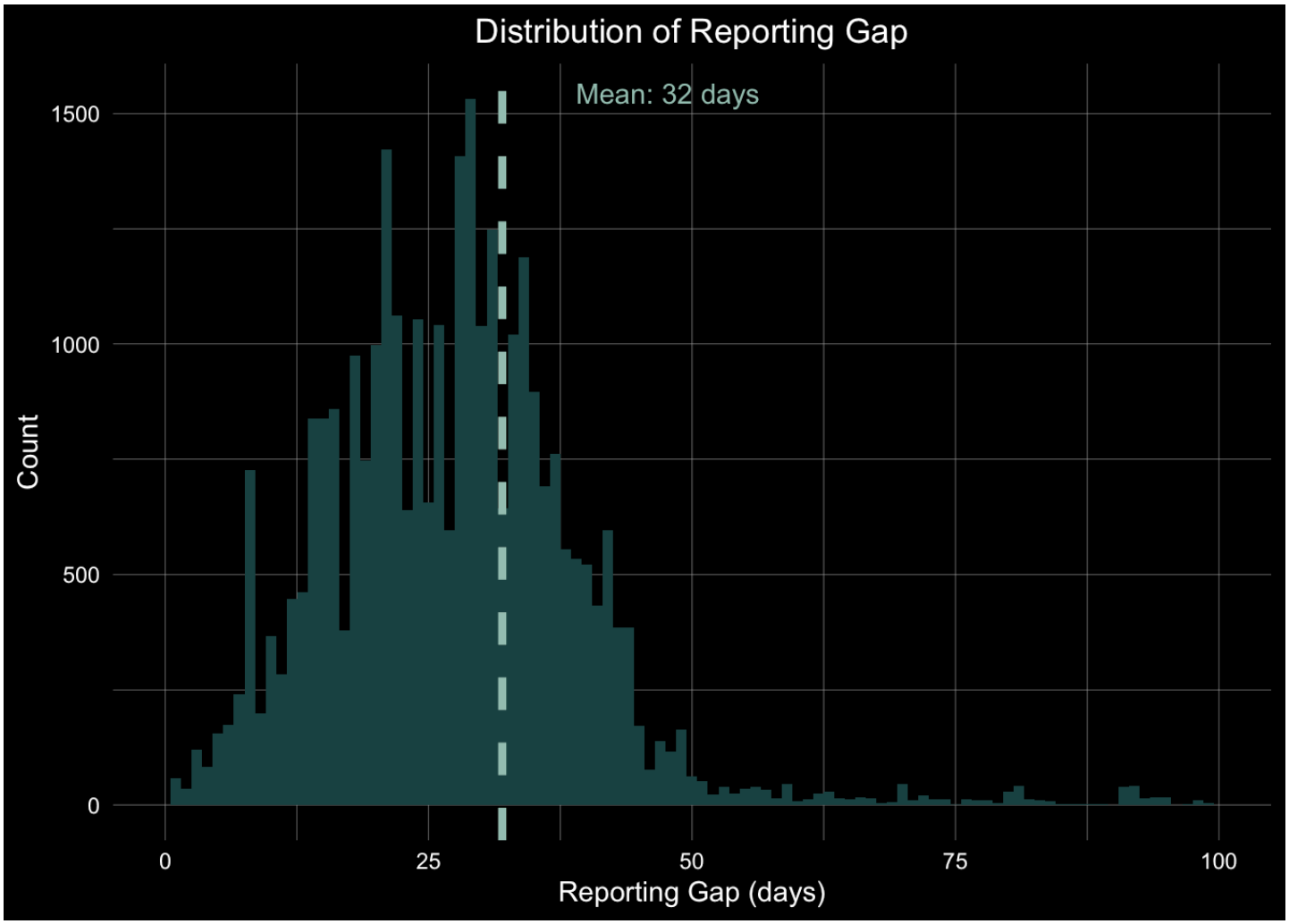
```
# global graphing vars
party_colors <- c(Democrat = "blue", Republican = "red", Other = "#777777")

dark_theme <- function() {
  theme_minimal() +
  theme(plot.background = element_rect(fill = "black"),
        panel.background = element_rect(fill = "black"),
        text = element_text(color = "white"),
        axis.title = element_text(color = "white"),
        axis.text = element_text(color = "white"),
        legend.background = element_rect(fill = "darkgrey", color = "white"),
        legend.position = "bottom",
        panel.grid.major = element_line(color = "darkgray", size = 0.1),
        panel.grid.minor = element_line(color = "darkgray", size = 0.1),
        plot.title = element_text(hjust = 0.5))
}
```

Plots

```
# prepare data
plot_data <- politician_info_df[!is.na(politician_info_df$Average_Annualized_Percenta
ge_Profit), ]
mean_profit <- mean(politician_info_df$Average_Annualized_Percentage_Profit, na.rm =
TRUE)

# plot
ggplot(plot_data,
       aes(y = reorder(Politician.Name, Average_Annualized_Percentage_Profit),
           x = Average_Annualized_Percentage_Profit, fill = Party)) +
  # main data
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = party_colors) +
  # mean line
  geom_vline(xintercept = mean_profit, linetype = "dashed", color = "yellow", size =
0.5) +
  annotate("text", y = mean_profit, x = 0, label = sprintf("Mean profit: %.2f", mean_
profit),
            vjust = -0.5, hjust = -0.2, color = "yellow", size = 3.5) +
  # theme
  dark_theme() +
  theme(axis.text.y = element_blank())+
  # labels
  labs(title = "Profitability Distribution by Politicians",
       x = "Average Annualized Percentage Profit (%)",
       y = "Politicians")
```


Profitability Distribution by Politicians
Politicians
Mean profit: 4.68
-250
0
250
500
Average Annualized Percentage Profit (%)
Party
Democrat
Other
Republican

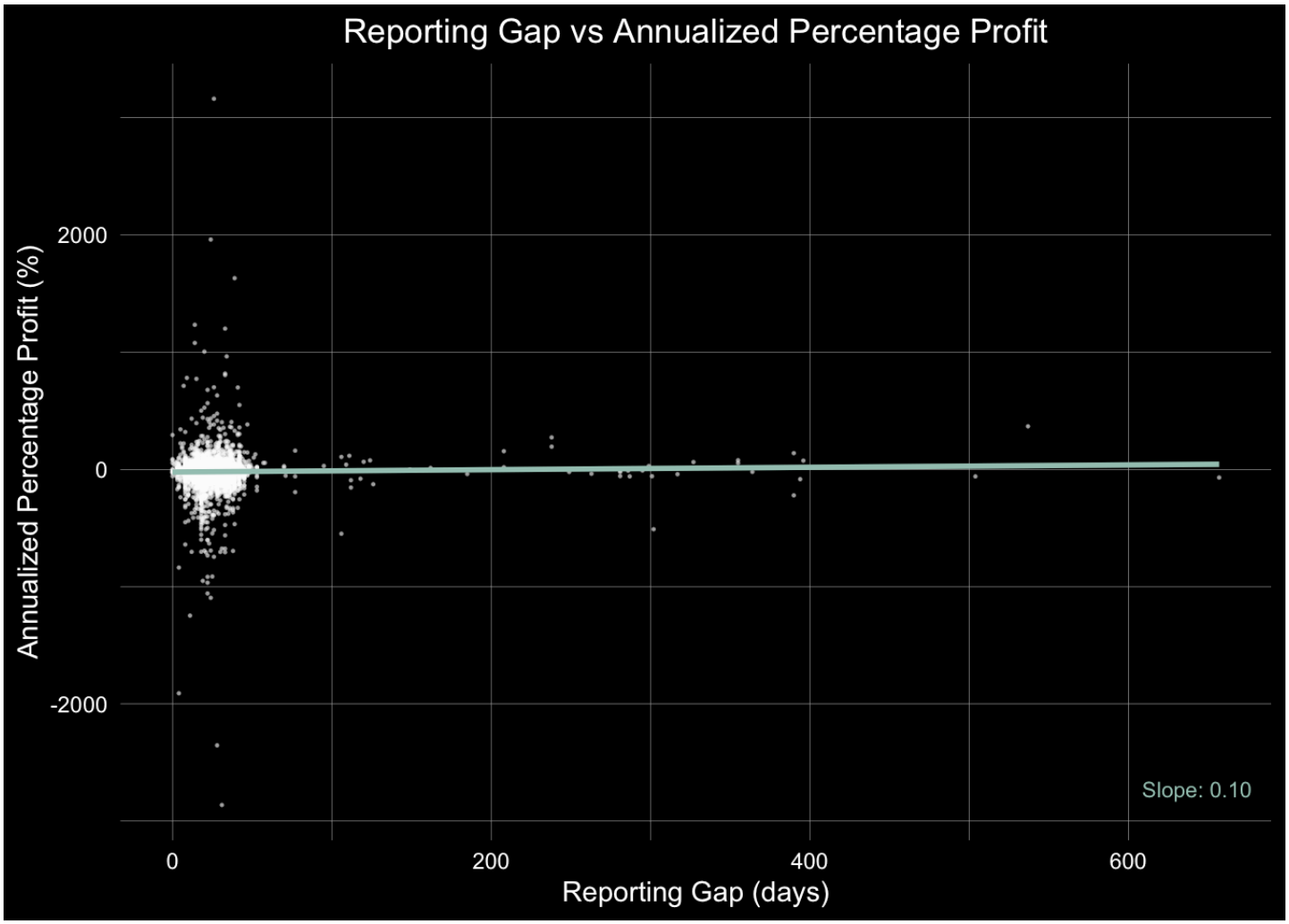
```
# prepare data
plot_data <- transactions_df
mean_gap <- mean(transactions_df$Reporting.Gap, na.rm = TRUE)

# plot
ggplot(plot_data, aes(x = Reporting.Gap)) +
  # main data
  geom_histogram(binwidth = 1, fill = "#1B4242") +
  # mean line
  geom_vline(aes(xintercept = mean_gap), color = "#96BDB0", linetype = "dashed", size
= 1.5) +
  annotate("text", x = mean_gap, y = Inf, label = sprintf("Mean: %.0f days", mean_ga
p),
           vjust = 2, hjust=-0.4, color = "#96BDB0") +
  # theme
  dark_theme() +
  # labels
  labs(title = "Distribution of Reporting Gap",
       x = "Reporting Gap (days)",
       y = "Count") +
  # axes
  xlim(0, 100)
```



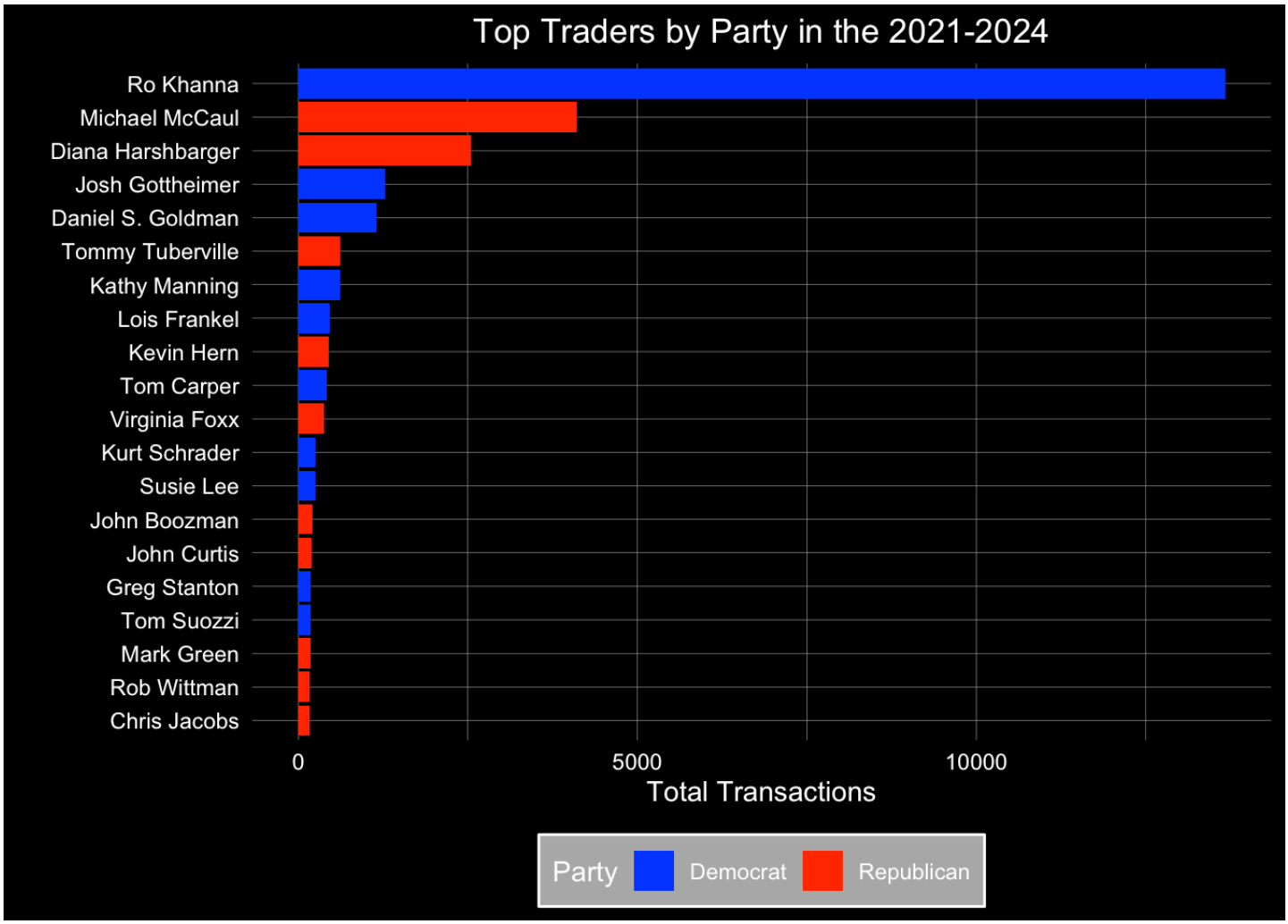

```
# prepare data
plot_data <- trades_df

# model
model <- lm(Annualized.Percentage.Profit ~ Report.Gap, data = plot_data)
summary(model)
```

```
##
## Call:
## lm(formula = Annualized.Percentage.Profit ~ Report.Gap, data = plot_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2843.4   -24.9     9.9    32.7  3181.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -22.76493    2.70443  -8.418   <2e-16 ***
## Report.Gap    0.10136    0.07266   1.395    0.163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 137.4 on 5319 degrees of freedom
## Multiple R-squared:  0.0003658,  Adjusted R-squared:  0.0001778
## F-statistic: 1.946 on 1 and 5319 DF,  p-value: 0.163
```

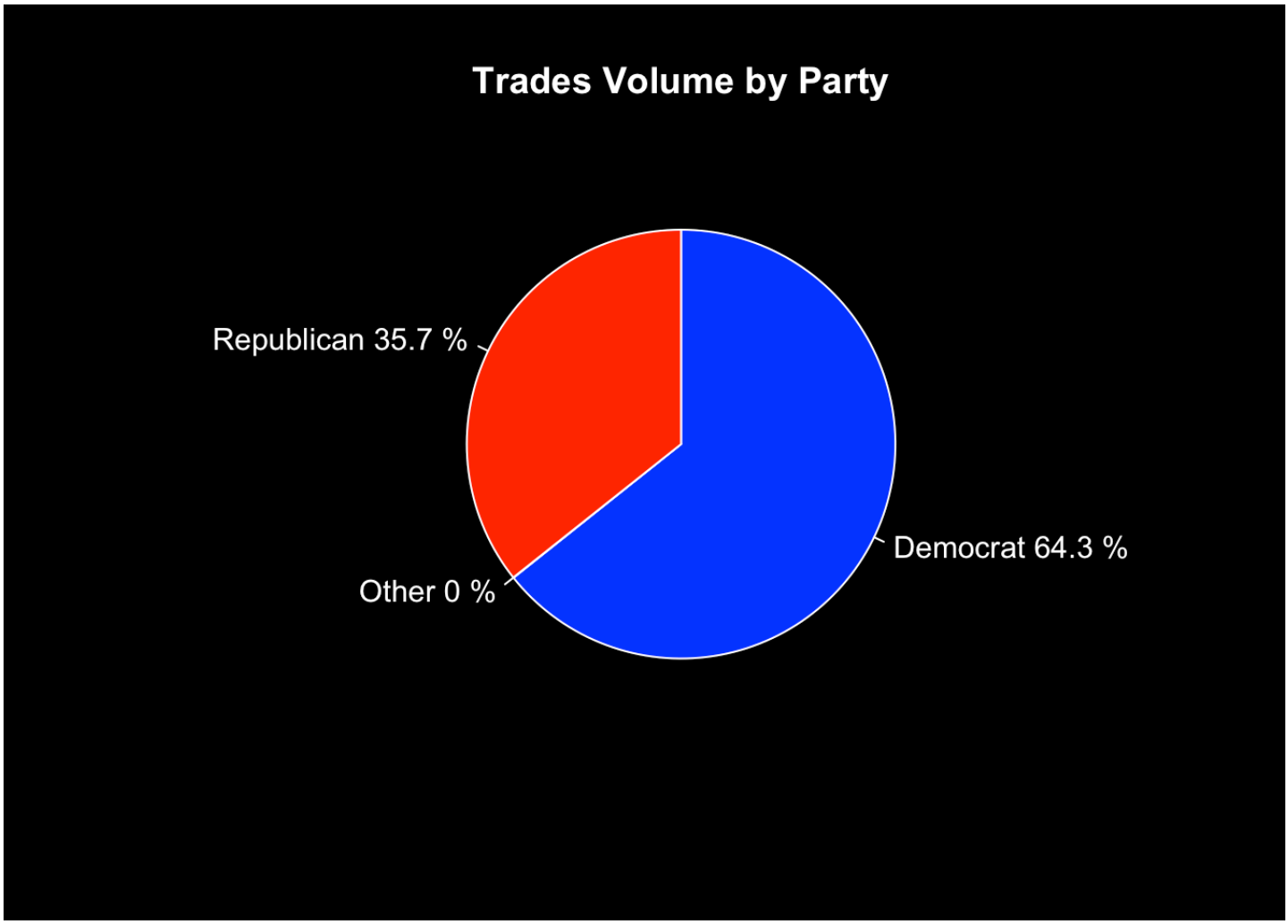
```
slope <- coef(model)["Report.Gap"]

# plot
ggplot(plot_data, aes(x = Report.Gap, y = Annualized.Percentage.Profit)) +
  # main data
  geom_point(alpha = 0.4, size = 0.2, color = "white") +
  # reg line
  geom_smooth(method = "lm", se = FALSE, color = "#96BDB0") +
  annotate("text",
           x = max(trades_df$Report.Gap, na.rm = TRUE),
           y = min(trades_df$Annualized.Percentage.Profit,
           na.rm = TRUE),
           label = sprintf("Slope: %.2f", slope),
           hjust = 0.7,
           vjust = -0.5,
           size = 3,
           color = "#96BDB0") +
  # theme
  dark_theme() +
  # labels
  labs(title = "Reporting Gap vs Annualized Percentage Profit",
       x = "Reporting Gap (days)",
       y = "Annualized Percentage Profit (%)")
```



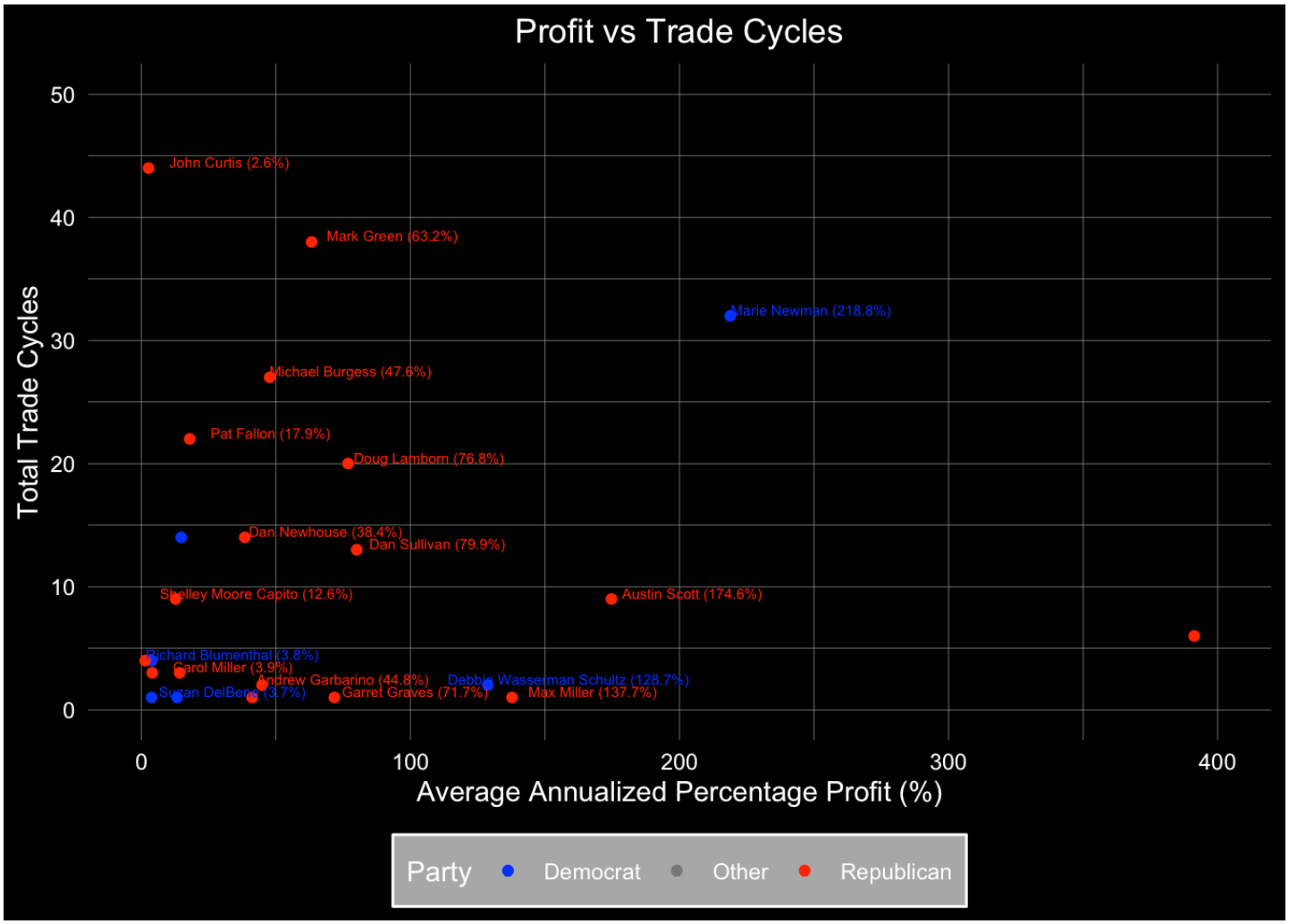

```
# prepare data
plot_data <- politician_info_df |>
  top_n(20, Total_Transactions)

# plot
ggplot(plot_data, aes(y = reorder(Politician.Name, Total_Transactions),
                      x = Total_Transactions,
                      fill = Party)) +
  # main data
  geom_bar(stat = "identity") +
  scale_fill_manual(values = party_colors) +
  # theme
  dark_theme() +
  # labels
  labs(title = "Top Traders by Party in the 2021-2024",
       x = "Total Transactions",
       y = "")
```



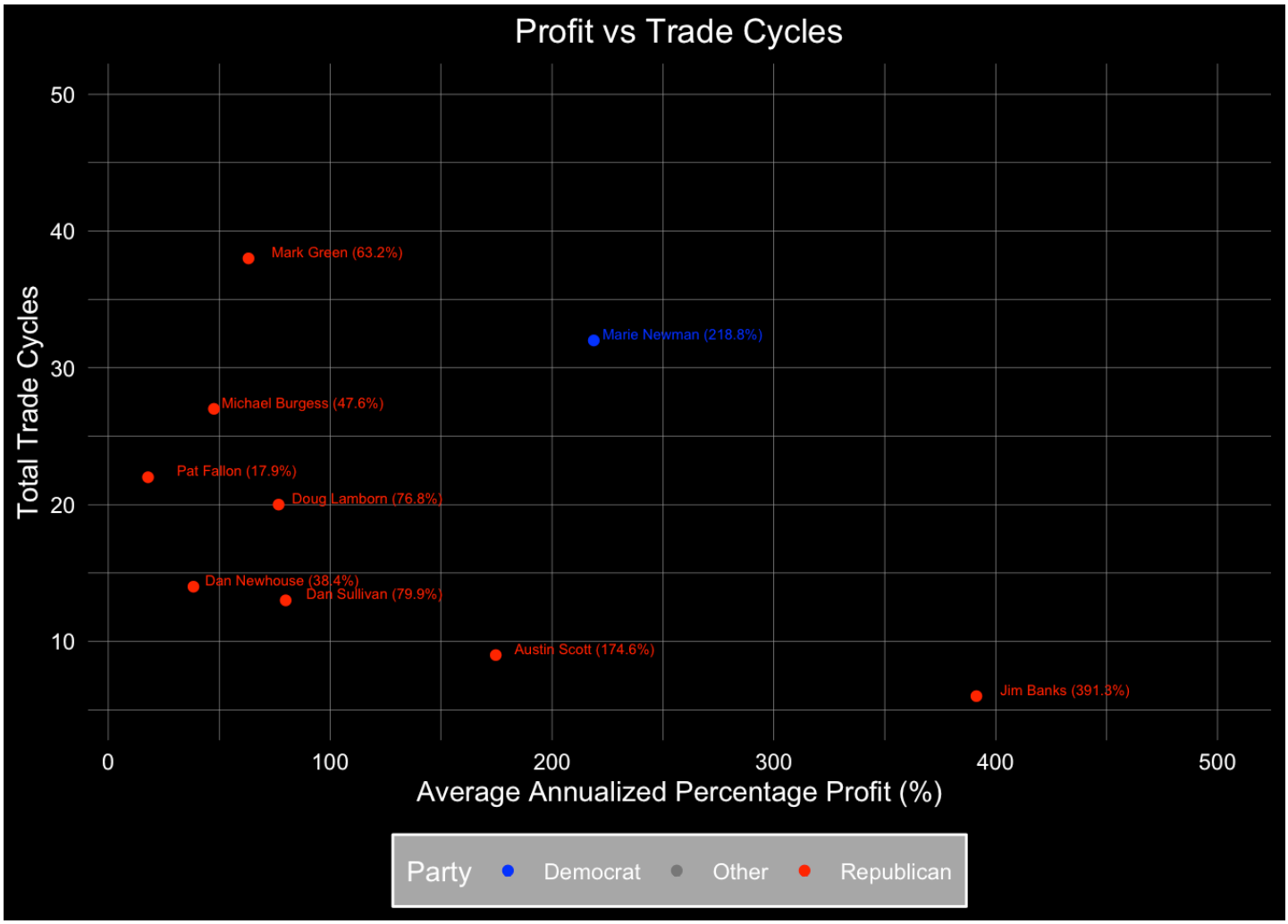

```
# prepare data
plot_data <- transactions_df |>
  count(Party)
percentages <- round(100 * plot_data$n / sum(plot_data$n), 1)
labels <- paste(plot_data$Party, percentages, "%", sep=" ")

# plot
par(bg = "black", col.main = "white", col.lab = "white", col.axis = "white", fg = "wh
ite")
pie(plot_data$n, labels = labels,
    col = c("blue", "gray", "red"),
    main = "Trades Volume by Party",
    init.angle = 90,
    clockwise = TRUE)
```


Trades Volume by Party
Republican 35.7 %
Democrat 64.3 %
Other 0 %

```
# prepare data
plot_data <- politician_info_df |>
  mutate(label = paste(Politician.Name,
                       sprintf("(%.1f%%)",
                       Average_Annualized_Percentage_Profit),
                       sep = " "))

# plot
ggplot(plot_data, aes(x = Average_Annualized_Percentage_Profit,
                          y = Total_Trades, color = Party)) +
  # main data
  geom_point() +
  geom_text(aes(label = label),
            nudge_x = 30,
            nudge_y = 0.5,
            size = 2,
            check_overlap = TRUE) +
  scale_color_manual(values = party_colors) +
  # theme
  dark_theme() +
  # labels
  labs(title = "Profit vs Trade Cycles",
       x = "Average Annualized Percentage Profit (%)",
       y = "Total Trade Cycles") +
  # axes
  xlim(0, 400) +
  ylim(0, 50)
```


Profit vs Trade Cycles
Total Trade Cycles
50
40
30
20
10
0
John Curtis (2.6%)
Mark Green (63.2%)
Marie Newman (218.8%)
Michael Burgess (47.6%)
Pat Fallon (17.9%)
Doug Lamborn (76.8%)
Dan Newhouse (38.4%)
Dan Sullivan (79.9%)
Shelley Moore Capito (12.6%)
Austin Scott (174.6%)
Richard Blumenthal (3.8%)
Carol Miller (3.9%)
Andrew Garbarino (44.8%)
Debbie Wasserman Schultz (128.7%)
Susan DelBene (3.7%)
Garret Graves (71.7%)
Max Miller (137.7%)
0
100
200
300
400
Average Annualized Percentage Profit (%)
Party
Democrat
Other
Republican

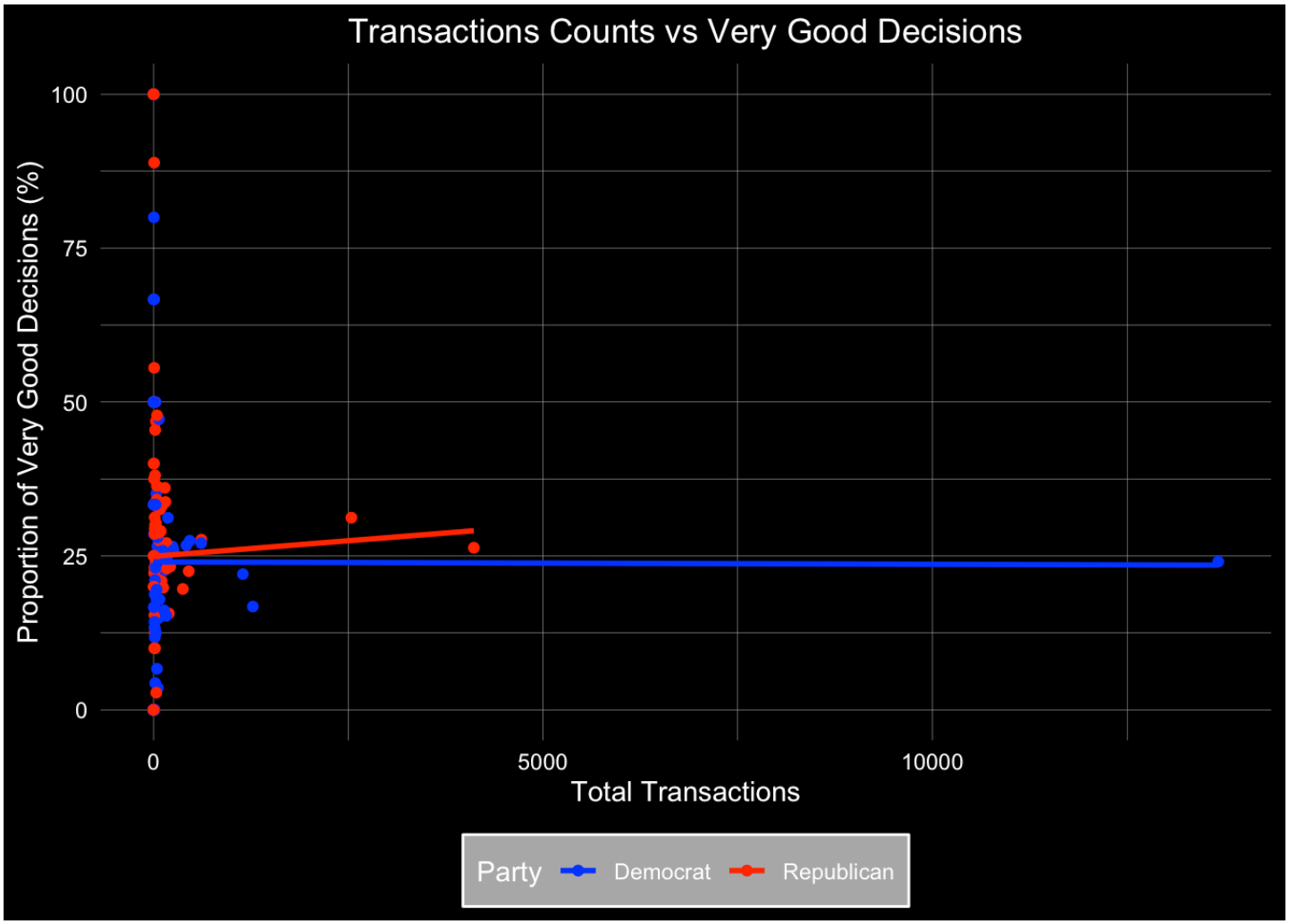
```
# plot
ggplot(plot_data, aes(x = Average_Annualized_Percentage_Profit,
                          y = Total_Trades, color = Party)) +
  # main data
  geom_point() +
  geom_text(aes(label = label),
            nudge_x = 40,
            nudge_y = 0.5,
            size = 2,
            check_overlap = TRUE) +
  scale_color_manual(values = party_colors) +
  # theme
 dark_theme() +
  # labels
  labs(title = "Profit vs Trade Cycles",
       x = "Average Annualized Percentage Profit (%)",
       y = "Total Trade Cycles") +
  # axes
  xlim(15, 500) +
  ylim(5, 50)
```



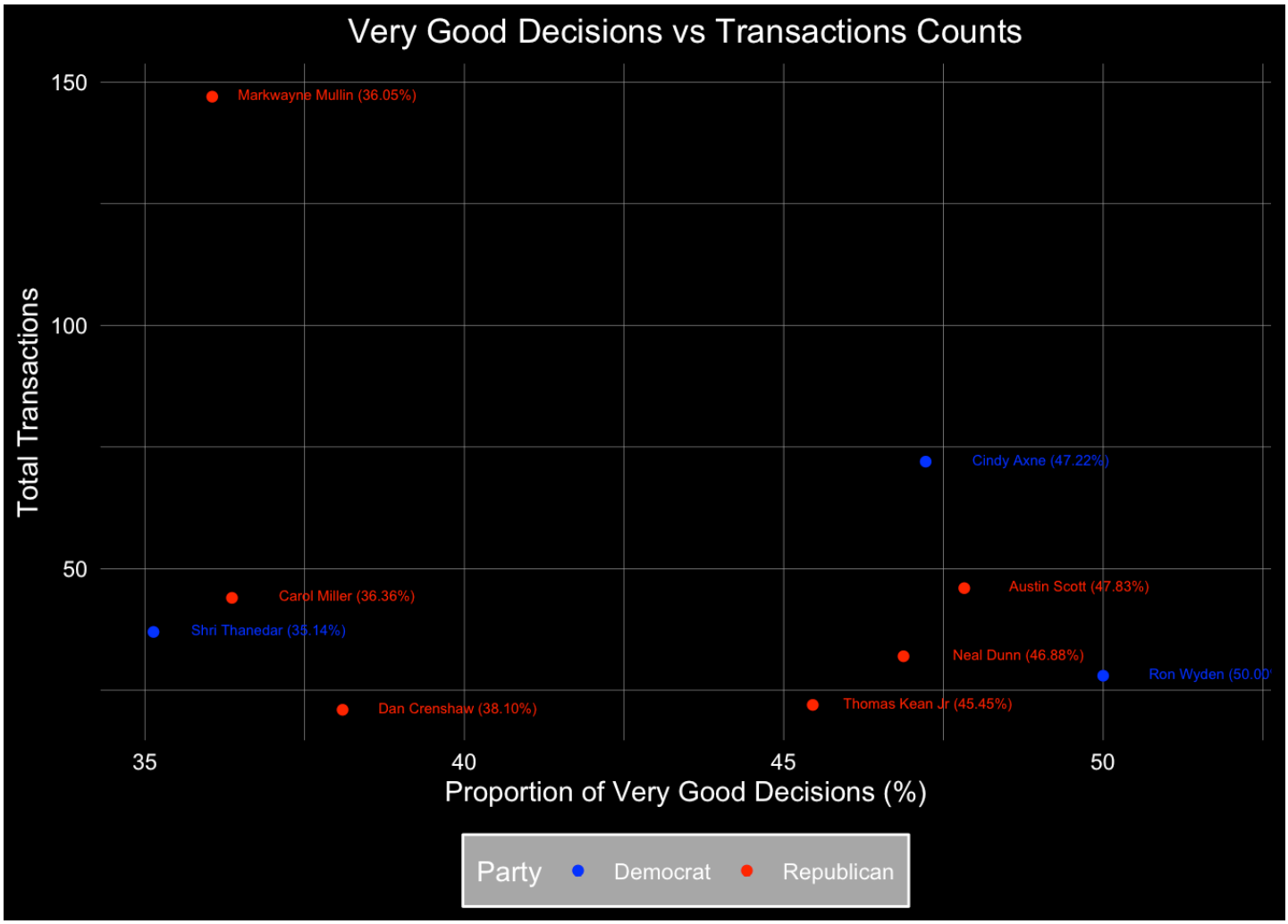

```
# prepare data
plot_data <- politician_info_df[!is.na(politician_info_df$Proportion_of_Very_Good_Dec
isions), ]

# model
models <- plot_data |>
  group_by(Party) |>
  do(tidy(lm(Proportion_of_Very_Good_Decisions ~ Total_Transactions, data = .)))

intercepts <- models |>
  filter(term == "(Intercept)") |>
  select(Party, intercept = estimate)

print(intercepts)
```

```
## # A tibble: 2 × 2
## # Groups:   Party [2]
##   Party      intercept
##   <chr>          <dbl>
## 1 Democrat        24.0
## 2 Republican      24.9
```

```
# plot
ggplot(plot_data, aes(x = Total_Transactions,
                      y = Proportion_of_Very_Good_Decisions, color = Party)) +
  # main data
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  scale_color_manual(values = party_colors) +
  # theme
  dark_theme() +
  # labels
  labs(title = "Transactions Counts vs Very Good Decisions",
       x = "Total Transactions",
       y = "Proportion of Very Good Decisions (%)")
```


Transactions Counts vs Very Good Decisions
Proportion of Very Good Decisions (%)
100
75
50
25
0
0
5000
10000
Total Transactions
Party
Democrat
Republican

```
# prepare data
plot_data <- politician_info_df |>
  filter(Total_Transactions < 500) |>
  filter(Total_Transactions > 10 ) |>
  filter(Proportion_of_Very_Good_Decisions > 35) |>
  mutate(label = sprintf("%s (%.2f%%)", Politician.Name, Proportion_of_Very_Good_Deci
sions))

ggplot(plot_data, aes(y = Total_Transactions,
                      x = Proportion_of_Very_Good_Decisions,
                      color = Party)) +
  # main data
  geom_point() +
  geom_text(aes(label = label),
            nudge_x = 1.8,
            nudge_y = 0.5,
            size = 2,
            check_overlap = TRUE) +
  scale_color_manual(values = party_colors) +  # Define custom colors for the groups
  # theme
  dark_theme() +
  # labels
  labs(title = "Very Good Decisions vs Transactions Counts",
       x = "Proportion of Very Good Decisions (%)",
       y = "Total Transactions")
```




```
  xlim(35, 55)
```

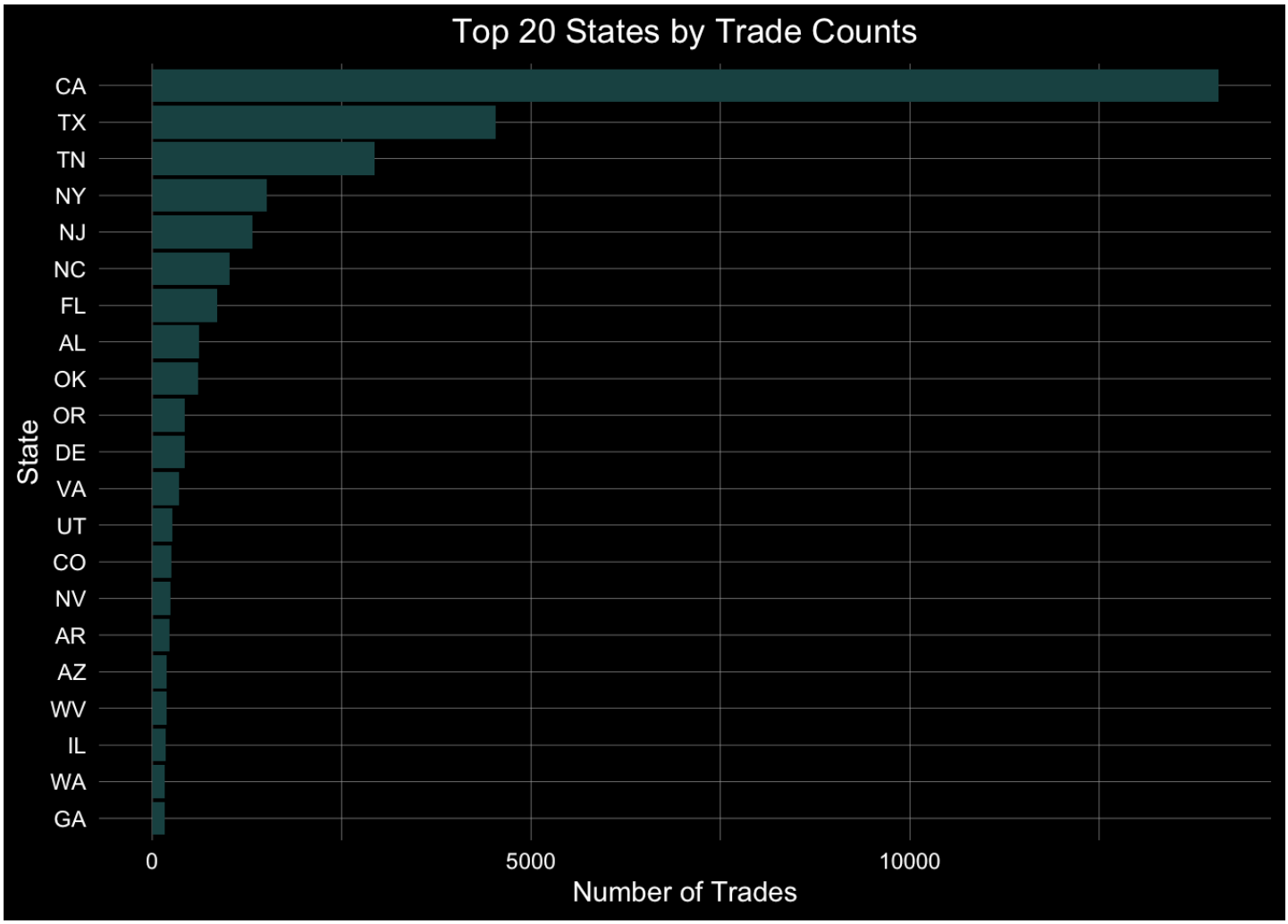
```
## <ScaleContinuousPosition>
##  Range:
##  Limits:   35 --   55
```

```
# binom test on the proportion of very good decisions
tst_data <- politician_info_df[!is.na(politician_info_df$`Very Good Decision`), ]

tst_results <- tst_data |>
  rowwise() |>
  mutate(
    test_result = list(binom.test(x = `Very Good Decision`, n = Total_Transactions, p
= 0.25)),
    p_value = test_result$p.value,
    statistic = test_result$statistic
  ) |>
  ungroup()

sig_results <- tst_results %>%
  filter(p_value < 0.05, Proportion_of_Very_Good_Decisions > 25) |>
  arrange(desc(Proportion_of_Very_Good_Decisions)) |>
  select(`Politician.Name`, Total_Transactions, Proportion_of_Very_Good_Decisions, p_
value)

print(sig_results)
```

```
## # A tibble: 14 × 4
##    Politician.Name    Total_Transactions Proportion_of_Very_Good_Deci…¹  p_value
##    <chr>                           <int>                          <dbl>    <dbl>
##  1 Eric Burlison                       5                          100   9.77e- 4
##  2 Trey Hollingsworth                  9                           88.9 1.07e- 4
##  3 Frank Pallone                       5                           80   1.56e- 2
##  4 Bobby Scott                         9                           66.7 9.99e- 3
##  5 Van Taylor                          9                           55.6 4.89e- 2
##  6 Ron Wyden                          28                           50   4.10e- 3
##  7 Austin Scott                       46                           47.8 9.12e- 4
##  8 Cindy Axne                         72                           47.2 5.24e- 5
##  9 Neal Dunn                          32                           46.9 7.14e- 3
## 10 Thomas Kean Jr                     22                           45.5 4.44e- 2
## 11 Markwayne Mullin                  147                           36.1 2.99e- 3
## 12 Chuck Fleischmann                 154                           33.8 1.52e- 2
## 13 Diana Harshbarger                2541                           31.2 1.84e-12
## 14 Michael McCaul                   4113                           26.3 4.97e- 2
## # ℹ abbreviated name: ¹Proportion_of_Very_Good_Decisions
```

```
write.csv(sig_results, "sus_poltics.csv")
```

```
plot_data <- transactions_df |>
  count(State) |>
  arrange(desc(n)) |>
  slice_max(n, n = 20)

ggplot(plot_data, aes(x = reorder(State, n), y = n)) +
  # main data
  geom_bar(stat = "identity", fill = "#1B4242") +
  coord_flip() +
  # theme
  dark_theme() +
  # labels
  labs(x = "State",
       y = "Number of Trades",
       title = "Top 20 States by Trade Counts")
```



```
plot_data <- transactions_df |>
  drop_na(Issuer.Name) |>
  count(Issuer.Name) |>
  top_n(10, n) |>
  arrange(desc(n))

ggplot(plot_data, aes(x = reorder(Issuer.Name, n), y = n)) +
  # main data
  geom_bar(stat = "identity", fill = "#1B4242") +
  coord_flip() +
  # theme
  dark_theme() +
  # labels
  labs(title = "The Most Popular Stocks",
       x = "Issuer Name",
       y = "Count")
```