STOR 664 Team Project Part 2

Aniruddhan, Jack, Jaehyuk and Soumyajouti

Quick Summary of Part 1

Understanding how NFL teams allocate their salary cap spending and draft capital across positions is critical for evaluating organizational strategy and predicting competitive success. Similarly, draft capital is a scarce resource that teams must strategically deploy to build sustainable rosters This team project examines three fundamental questions about NFL resource allocation from 2013 to 2024:

- How are allocations changing over time?
- How different are allocations with respect to teams?
- How do allocations relate to outcomes?

Details about data pre-processing and exploratory data analysis are contained in the part 1 report. In this pdf we only include data analyses and discussion.

Question 1: How are allocations changing over time?

```
library(tidyverse)
library(broom)
library(corrplot)
library(dplyr)
library(glmnet)
library(car)
library(ggplot2)
```

```
cap_year <- read_csv("../data/processed/capital_by_position_year.csv")
cap_year <- cap_year %>%
  mutate(
    position = factor(position),
    year_c   = year - min(year)   # e.g., 0 for 2013, 1 for 2014, ...
  )

glimpse(cap_year)
```

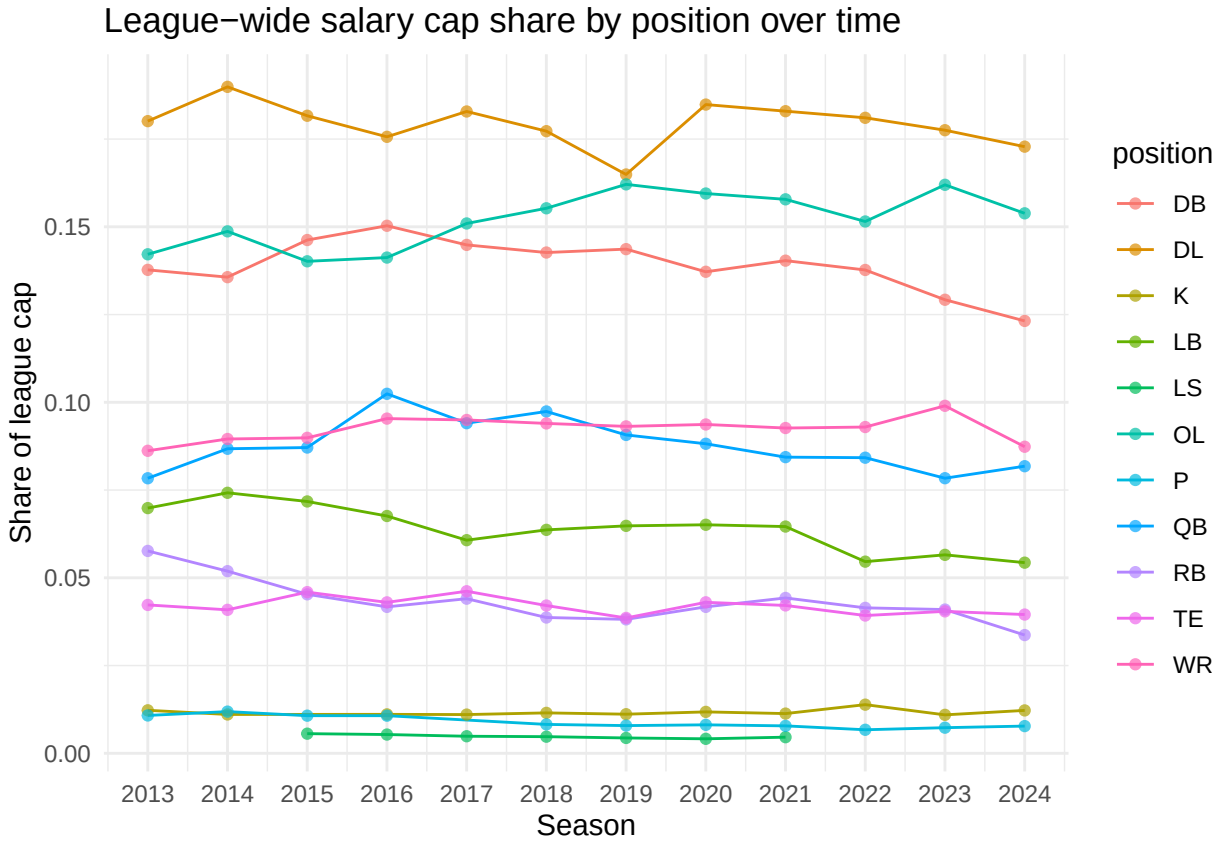
```
## Rows: 125
## Columns: 6
## $ year         <dbl> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 201~
## $ position     <fct> DB, DL, K, LB, OL, P, QB, RB, TE, WR, DB, DL, K, LB, OL, ~
## $ draft_pct_lg <dbl> 0.213171823, 0.200712144, 0.004463677, 0.107628004, 0.191~
## $ cap_pct_team <dbl> 4.4075, 5.7635, 0.3920, 2.2355, 4.5495, 0.3450, 2.5075, 1~
## $ cap_pct_lg   <dbl> 0.13773437, 0.18010938, 0.01225000, 0.06985938, 0.1421718~
## $ year_c       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

```
summary(cap_year)
```

```
##       year         position   draft_pct_lg       cap_pct_team
```

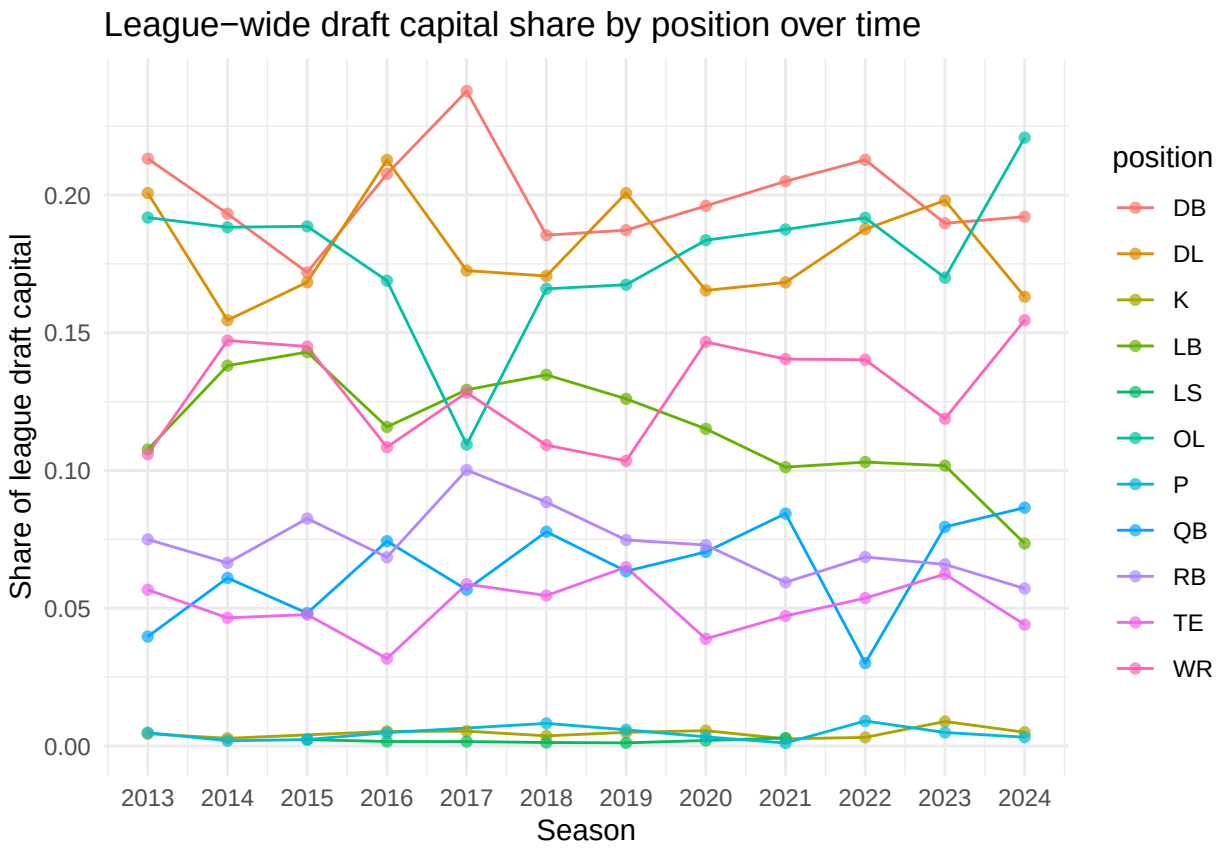
```
##  Min.   :2013   DB     :12   Min.   :0.001096   Min.   :0.132
##  1st Qu.:2016   DL     :12   1st Qu.:0.038910   1st Qu.:1.234
##  Median :2019   LB     :12   Median :0.084365   Median :2.236
##  Mean   :2019   OL     :12   Mean   :0.096000   Mean   :2.524
##  3rd Qu.:2021   QB     :12   3rd Qu.:0.165949   3rd Qu.:4.407
##  Max.   :2024   RB     :12   Max.   :0.237728   Max.   :6.076
##                 (Other):53
##    cap_pct_lg          year_c
##  Min.   :0.004125   Min.   : 0.000
##  1st Qu.:0.038562   1st Qu.: 3.000
##  Median :0.069859   Median : 6.000
##  Mean   :0.078866   Mean   : 5.512
##  3rd Qu.:0.137703   3rd Qu.: 8.000
##  Max.   :0.189875   Max.   :11.000
##
```

```r
ggplot(cap_year, aes(x = year, y = cap_pct_lg,
                     color = position, group = position)) +
  geom_line() +
  geom_point(alpha = 0.7) +
  scale_x_continuous(breaks = sort(unique(cap_year$year))) +
  labs(
    title = "League-wide salary cap share by position over time",
    x = "Season",
    y = "Share of league cap"
  ) +
  theme_minimal()
```



The above figure shows how much of the league's salary cap each position gets from 2013 to 2024. Defensive line (DL) and offensive line (OL) usually get the biggest shares, followed by defensive backs (DB) and wide receivers (WR). Kicker (K), punter (P), and long snapper (LS) always get very small amounts. Most positions do not change very much over time, but there are a few patterns: OL and WR seem to get a little more cap share over the years, while DB and running back (RB) seem to get a little less.

```
ggplot(cap_year, aes(x = year, y = draft_pct_lg,
                     color = position, group = position)) +
  geom_line() +
  geom_point(alpha = 0.7) +
  scale_x_continuous(breaks = sort(unique(cap_year$year))) +
  labs(
    title = "League-wide draft capital share by position over time",
    x = "Season",
    y = "Share of league draft capital"
  ) +
  theme_minimal()
```



The above figure shows a similar plot, but now for how draft picks are used. Compared to cap spending, the draft shares jump around more from year to year. This makes sense because draft choices depend on how strong each draft class is and what teams need that year. Defensive backs (DB), defensive line (DL), running backs (RB), and wide receivers (WR) usually get the most draft capital, while specialists like kicker (K), punter (P), and long snapper (LS) get very little.

```
# Step 4: Full model for cap shares
formula_cap <- cap_pct_lg ~ position * year_c

model_cap <- lm(formula_cap, data = cap_year)
summary(model_cap)
```

```
##
## Call:
## lm(formula = formula_cap, data = cap_year)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0140940 -0.0015583 -0.0000414  0.0027839  0.0131021
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.1457796  0.0024771  58.851  < 2e-16 ***
## positionDL          0.0366861  0.0035032  10.472  < 2e-16 ***
## positionK          -0.1345057  0.0036690 -36.660  < 2e-16 ***
## positionLB         -0.0731284  0.0035032 -20.875  < 2e-16 ***
## positionLS         -0.1399166  0.0052620 -26.590  < 2e-16 ***
## positionOL         -0.0021863  0.0035032  -0.624  0.53395
## positionP          -0.1344763  0.0035681 -37.688  < 2e-16 ***
## positionQB         -0.0546510  0.0035032 -15.600  < 2e-16 ***
## positionRB         -0.0951041  0.0035032 -27.148  < 2e-16 ***
## positionTE         -0.1019557  0.0035032 -29.104  < 2e-16 ***
## positionWR         -0.0553329  0.0035032 -15.795  < 2e-16 ***
## year_c             -0.0012215  0.0003815  -3.202  0.00181 **
## positionDL:year_c   0.0006440  0.0005395   1.194  0.23535
## positionK:year_c    0.0012875  0.0005532   2.327  0.02191 *
## positionLB:year_c  -0.0003547  0.0005395  -0.658  0.51230
## positionLS:year_c   0.0010095  0.0009427   1.071  0.28675
## positionOL:year_c   0.0027690  0.0005395   5.133 1.35e-06 ***
## positionP:year_c    0.0007950  0.0005418   1.467  0.14536
## positionQB:year_c   0.0006186  0.0005395   1.147  0.25417
## positionRB:year_c  -0.0001208  0.0005395  -0.224  0.82322
## positionTE:year_c   0.0008786  0.0005395   1.629  0.10647
## positionWR:year_c   0.0015754  0.0005395   2.920  0.00430 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004562 on 103 degrees of freedom
## Multiple R-squared:  0.9948, Adjusted R-squared:  0.9937
## F-statistic: 937.1 on 21 and 103 DF,  p-value: < 2.2e-16
```

Big regression model

```
# Step 5: Full model for draft shares
formula_draft <- draft_pct_lg ~ position * year_c

model_draft <- lm(formula_draft, data = cap_year)
summary(model_draft)
```

```
##
## Call:
## lm(formula = formula_draft, data = cap_year)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.065835 -0.007925  0.000607  0.007447  0.037809
```

```
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.015e-01  8.512e-03  23.674  < 2e-16 ***
## positionDL         -1.934e-02  1.204e-02  -1.607  0.11115
## positionK          -1.976e-01  1.261e-02 -15.677  < 2e-16 ***
## positionLB         -6.447e-02  1.204e-02  -5.356 5.20e-07 ***
## positionLS         -2.000e-01  1.808e-02 -11.063  < 2e-16 ***
## positionOL         -3.315e-02  1.204e-02  -2.754  0.00696 **
## positionP          -1.976e-01  1.226e-02 -16.118  < 2e-16 ***
## positionQB         -1.493e-01  1.204e-02 -12.403  < 2e-16 ***
## positionRB         -1.197e-01  1.204e-02  -9.944  < 2e-16 ***
## positionTE         -1.523e-01  1.204e-02 -12.654  < 2e-16 ***
## positionWR         -8.124e-02  1.204e-02  -6.749 8.95e-10 ***
## year_c             -3.962e-04  1.311e-03  -0.302  0.76308
## positionDL:year_c   4.033e-05  1.854e-03   0.022  0.98268
## positionK:year_c    5.440e-04  1.901e-03   0.286  0.77533
## positionLB:year_c  -3.470e-03  1.854e-03  -1.872  0.06405 .
## positionLS:year_c   4.699e-04  3.239e-03   0.145  0.88494
## positionOL:year_c   2.119e-03  1.854e-03   1.143  0.25560
## positionP:year_c    5.076e-04  1.862e-03   0.273  0.78566
## positionQB:year_c   2.604e-03  1.854e-03   1.405  0.16308
## positionRB:year_c  -1.142e-03  1.854e-03  -0.616  0.53922
## positionTE:year_c   6.535e-04  1.854e-03   0.353  0.72517
## positionWR:year_c   1.988e-03  1.854e-03   1.073  0.28592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01567 on 103 degrees of freedom
## Multiple R-squared:  0.9598, Adjusted R-squared:  0.9516
## F-statistic: 117.1 on 21 and 103 DF,  p-value: < 2.2e-16
```

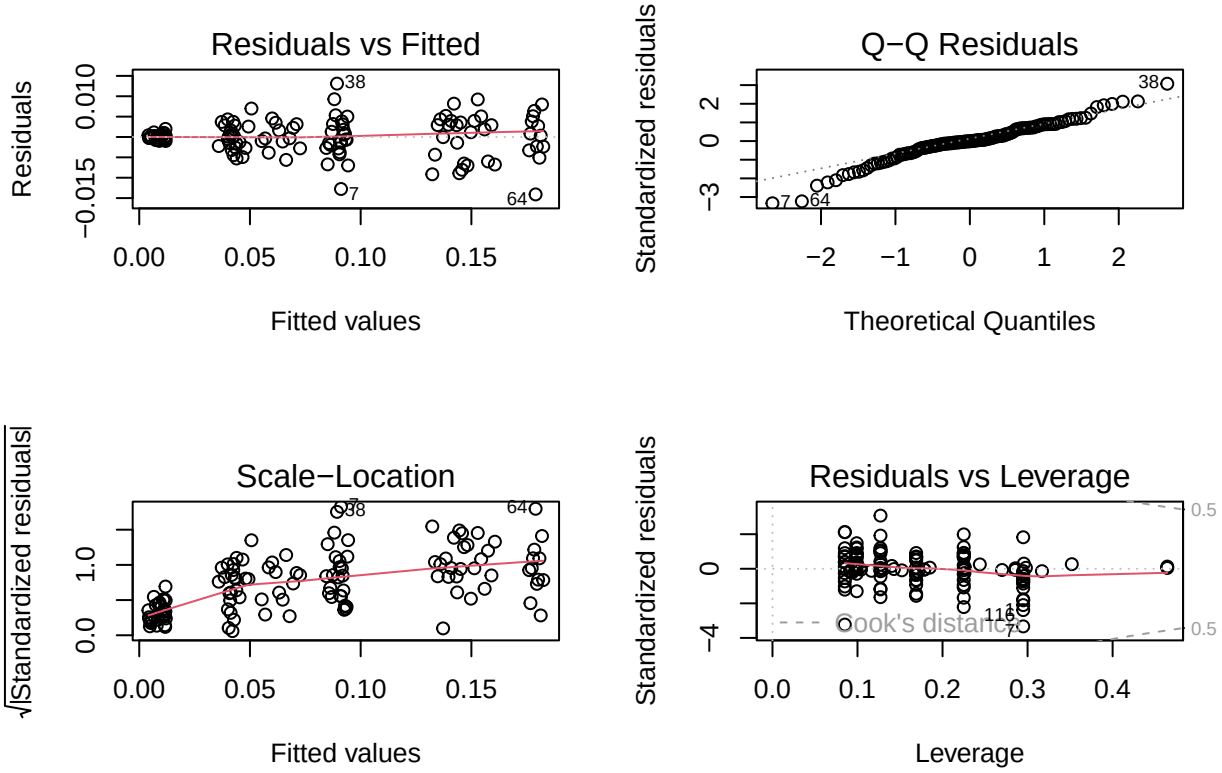
```r
# Per-position simple time trends for draft share
draft_slopes <- cap_year %>%
  group_by(position) %>%
  do(tidy(lm(draft_pct_lg ~ year_c, data = .))) %>%
  ungroup() %>%
  filter(term == "year_c") %>%
  arrange(desc(estimate)) %>%
  mutate(
    slope_pct_points = 100 * estimate
  )

draft_slopes_report <- draft_slopes %>%
  mutate(
    estimate = round(estimate, 5),
    slope_pct_points = round(slope_pct_points, 3),
    p.value = signif(p.value, 3)
  )

draft_slopes_report
```

```
## # A tibble: 11 x 7
##    position term   estimate std.error statistic p.value slope_pct_points
##    <fct>    <chr>     <dbl>     <dbl>     <dbl>   <dbl>            <dbl>
```

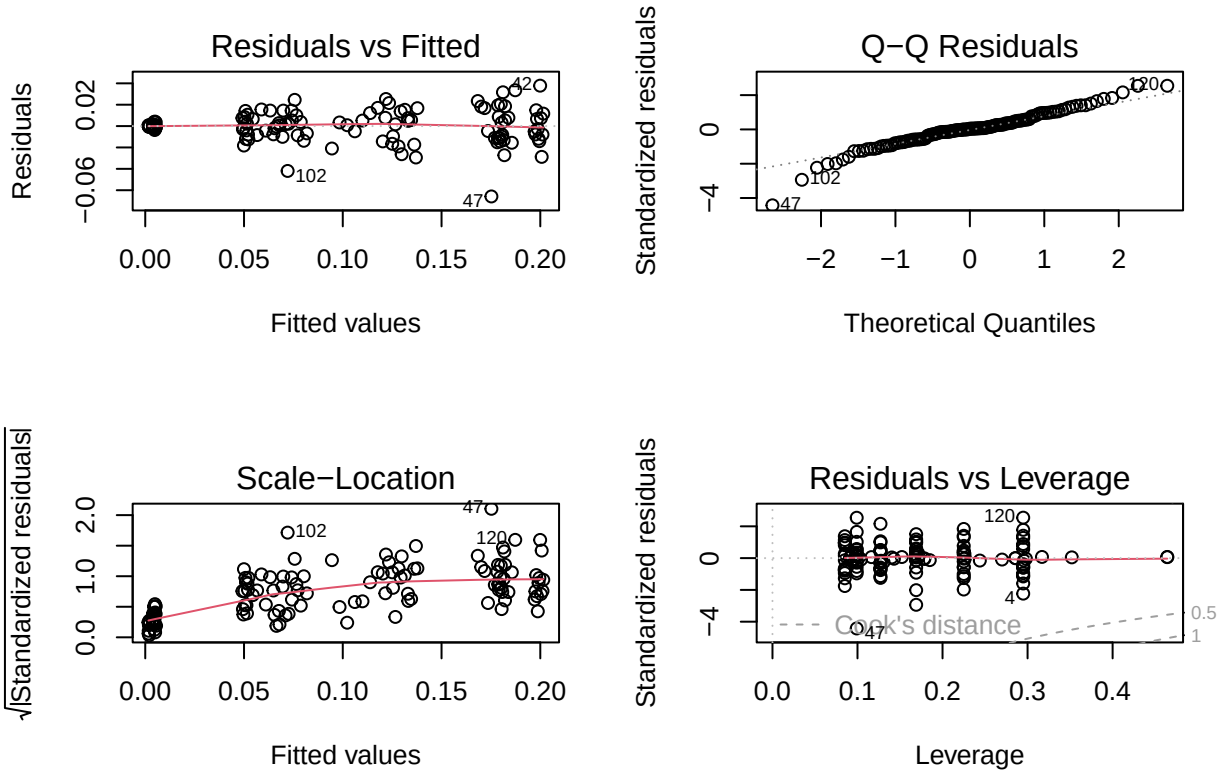
```
##  1 QB       year_c  0.00221  0.00141      1.56   0.149            0.221
##  2 OL       year_c  0.00172  0.00225      0.766  0.461            0.172
##  3 WR       year_c  0.00159  0.00157      1.01   0.336            0.159
##  4 TE       year_c  0.00026  0.000851     0.302  0.769            0.026
##  5 K        year_c  0.00015  0.000155     0.952  0.366            0.015
##  6 P        year_c  0.00011  0.000221     0.505  0.626            0.011
##  7 LS       year_c  0.00007  0.000124     0.594  0.579            0.007
##  8 DL       year_c -0.00036  0.00164     -0.217  0.833           -0.036
##  9 DB       year_c -0.0004   0.00150     -0.264  0.797           -0.04
## 10 RB       year_c -0.00154  0.000956    -1.61   0.139           -0.154
## 11 LB       year_c -0.00387  0.00123     -3.14   0.0105          -0.387
```

Diagnostics of the big regression model

```
# Step 6: Diagnostics for cap model
par(mfrow = c(2, 2))
plot(model_cap)
```




```
par(mfrow = c(1, 1))

# Step 6b: Diagnostics for draft model
par(mfrow = c(2, 2))
plot(model_draft)
```




```
par(mfrow = c(1, 1))
```

The diagnostic plots for both models look mostly fine. In the residuals vs fitted plots, the points are scattered around zero without a clear pattern, so using a straight-line (linear) trend over time seems okay. The Q–Q plots show small deviations from the straight line at the ends, which is not surprising because we are modelling proportions between 0 and 1. The scale–location plots show that the spread of the residuals is a bit larger for bigger fitted values, but not in a serious way. The residuals vs leverage plots show a few points with higher leverage, but nothing too extreme. Overall, the linear regression assumptions seem reasonable for our purpose of describing general time trends.

Discussion 1

We looked at how NFL spending by position has changed over time, using league-wide salary cap shares and draft shares from 2013 to 2024. The plots and regression models show that, for most positions, the shares stay fairly stable from year to year.

For salary cap spending, any changes happen slowly. Defensive backs (DB) lose a small amount of cap share over time, while offensive line (OL) and wide receivers (WR) gain a bit. Specialists such as kicker (K), punter (P), and long snapper (LS) always use only a very small part of the cap, and their trends over time are tiny.

For draft capital, the values move around more from year to year, but clear long-term trends are rare. The only strong pattern we see is that linebackers (LB) get a smaller share of draft picks over time. For most other positions, the estimated slopes are close to zero and not statistically significant, so we do not see a clear increase or decrease.

Overall, our results suggest that the league's spending and drafting by position are mostly stable over this period. The main changes are small: slightly more investment in OL and WR in cap spending, and slightly less investment in LB in the draft. These results give a numerical summary of how positional value has changed slowly in the NFL.

Question 2: How different are allocations with respect to teams?

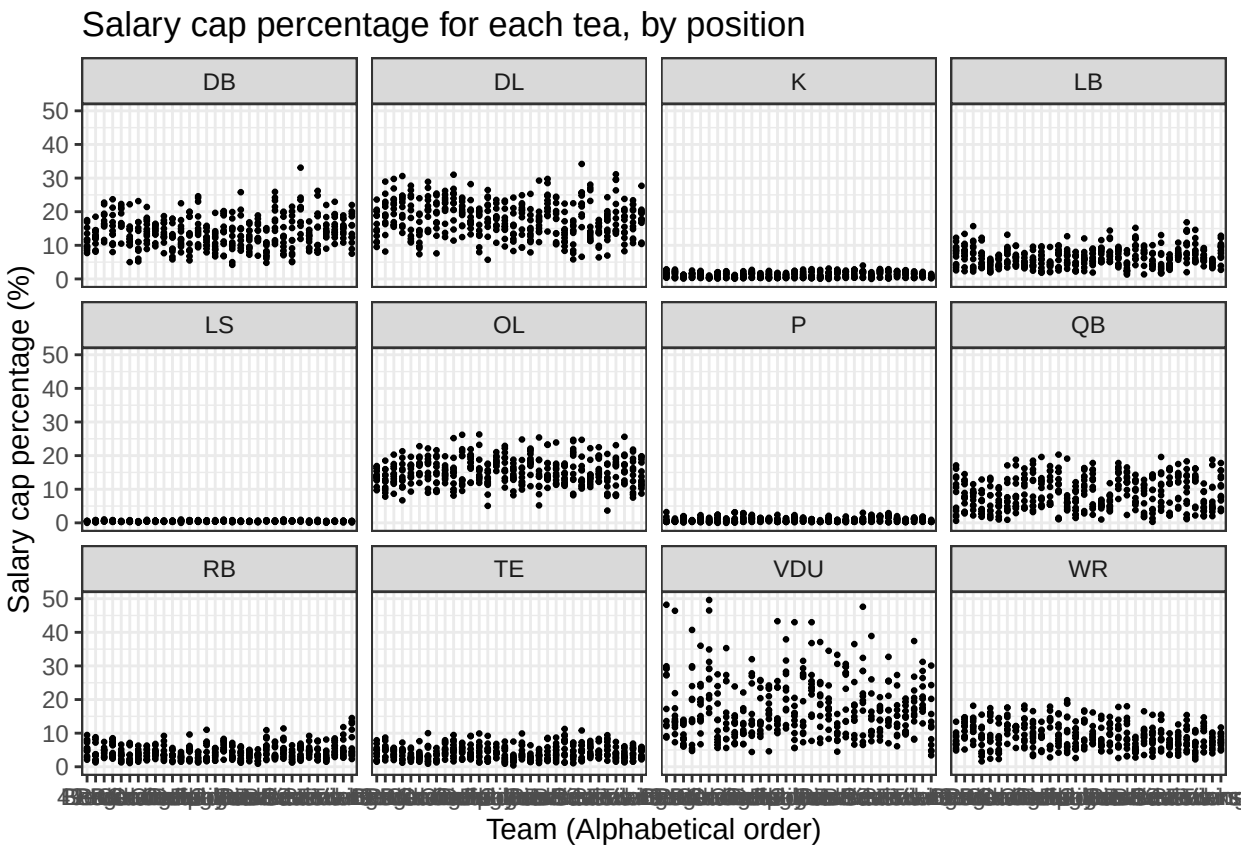
Assumption

Throughout this question, we assume that the capital strategy of each team does not change during the period. Therefore, each year represents an i.i.d. sample from a certain distribution. Additionally, since we perform more than 10 linear regression models, we don't check the linear model assumptions manually but presume they hold.

Exploratory Data Analysis

We begin with visualizing the data with scatter plots. In the below scatter plots, each subplots corresponds to one position. x-axis is team names (alphabetical order), and y-axis is the salary cap percentage (%).

```
data = read.csv('../data/processed/capital_by_position_team_year.csv')
# names(data)
position_list = unique(data$position)
team_list = unique(data$team)

ggplot(data=data) +
    geom_point(mapping = aes(x=team, y=cap_pct_team*100), size=0.5) +
    facet_wrap(~position) +
    theme_bw() +
    ggtitle('Salary cap percentage for each tea, by position') +
    xlab('Team (Alphabetical order)') +
    ylab('Salary cap percentage (%)')
```



As we can see in these plots, we can classify positions by the overall scale of salary cap: K, LS and P occupy

the lowest, LB, RB, and TE are in the middle, and DB, DL, OL, QB and WR are in the highest percentages. For each scatter plot, the distributions appear relatively flat across teams (except for VDU). Nonetheless, small fluctuations are present in several positions, of which we scrutinize by performing hypothesis testing.

Multiple hypotheses testing with Holm-Bonferroni correction

We first divide the dataset by positions.

```
data_by_position = lapply(position_list, function(pos) data[data$position == pos,])
names(data_by_position) = position_list
```

With each sub-dataset, we perform a linear regression model

$$\text{Salary cap}_{\text{pos}} = \beta_0^{\text{pos}} + \sum_{i=1}^{32} \beta_i^{\text{pos}} \mathbf{1}_{i\text{-th Team}},$$

where $\text{pos} \in \{DB, DL, \cdots, WR\}$. Here β_0^{pos} denotes the average among the league. As a result, we perform a multiple hypothesis testing with hypotheses

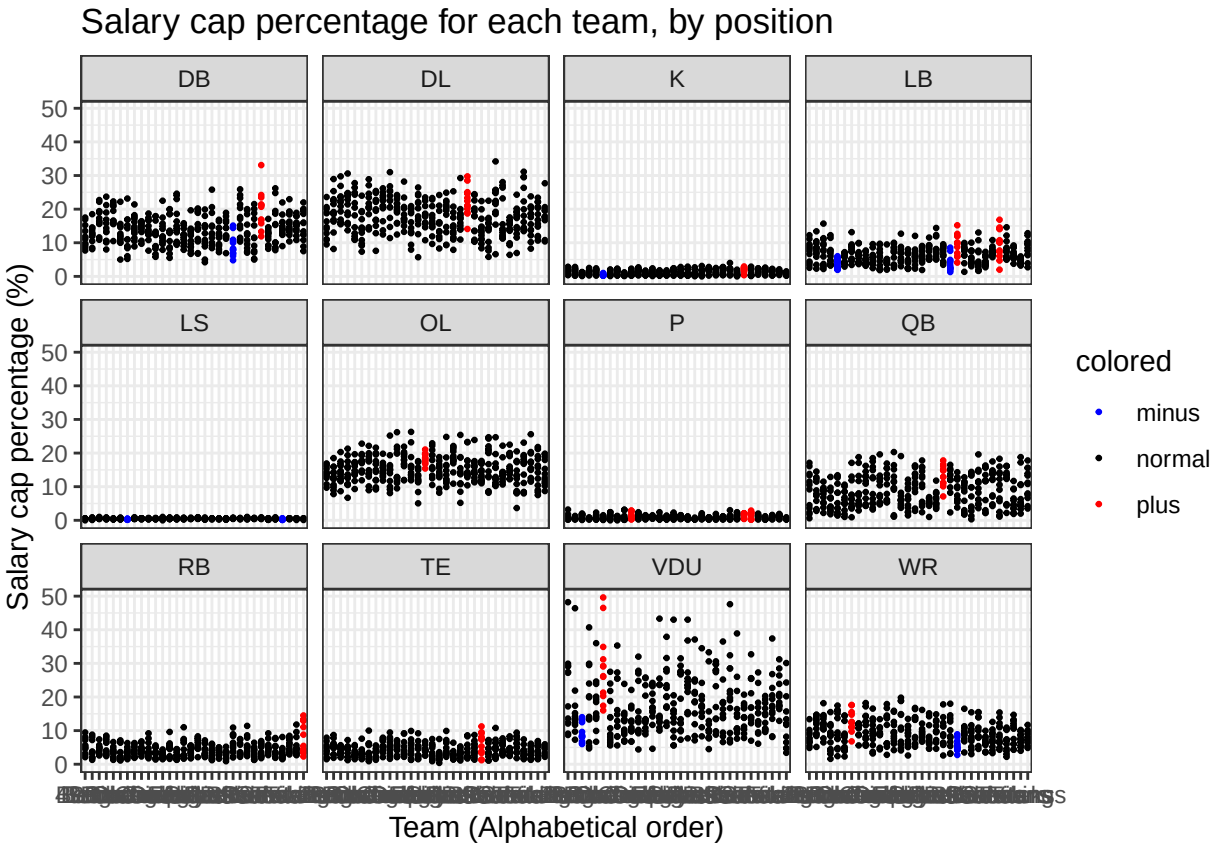
$$H_{0i}^{\text{pos}} : \beta_i^{\text{pos}} = 0 \quad \text{v.s.} \quad H_{1i}^{\text{pos}} : \beta_i^{\text{pos}} \neq 0, \qquad i \in \{1, 2, \cdots, 12\}.$$

We adapt a Holm-Bonferroni correction to control the family-wise error rate (FWER) with $\alpha = 0.05$.

```
lm_by_position = lapply(data_by_position, function(data_pos) {
    beta_zero = mean(data_pos$cap_pct_team)
    n = length(data_pos$cap_pct_team)
    lm(cap_pct_team~team-1, data=data_pos, offset=rep(beta_zero, n))
})

coeff_by_position = lapply(lm_by_position, function(lm_obj) summary(lm_obj)$coefficients)


# Holm-Bonferroni correction with level alpha = 0.05
HB_correction_by_position = lapply(coeff_by_position, function(coeff, alpha = 0.05) {
    pval = coeff[,"Pr(>|t|)"]
    ord = order(pval)
    pval_ordered = pval[ord]

    K = length(pval)
    rejection_rho = alpha / (K + 1 - (1:K))
    k = min(which(pval_ordered > rejection_rho))

    res = coeff[ord[1:(k-1)],]
    if(k<=2) {
        res = t(res)
        rownames(res) = rownames(coeff)[ord[1]]
    }
    return(res)
}, alpha=0.05)

# HB_correction_by_position
```

The following table and figure show the rejected hypotheses for each position. It a team invests significantly more than league avarage, it is marked as red. If a team does significantly lower, it is marked as blue.

Position	Team	Estimate	SE	t-value	$\mathbb{P}(>\|t\|)$
DB	Ravens	0.0609	0.0122	5.0056	8.82×10^{-7}
	Panthers	-0.0476	0.0122	-9.9136	1.09×10^{-4}
DL	Packers	0.0437	0.0142	3.0710	2.30×10^{-3}
K	Ravens	0.0076	0.0021	3.5495	4.39×10^{-4}
	Browns	-0.0073	0.0022	-3.2584	1.23×10^{-3}
LB	Seahawks	0.0349	0.0075	4.6689	4.31×10^{-6}
	Panthers	0.0304	0.0075	4.0661	5.90×10^{-5}
	Packers	-0.0259	0.0075	-3.4639	5.98×10^{-4}
	Broncos	-0.0239	0.0075	-3.2072	1.46×10^{-3}
LS	Buccaneers	-0.0019	0.0005	-4.2384	2.89×10^{-5}
	Steelers	-0.0017	0.0005	-3.6580	2.93×10^{-4}
OL	Eagles	0.0312	0.0113	2.7702	5.90×10^{-3}
P	Saints	0.0061	0.0018	3.4891	5.50×10^{-4}
	Ravens	0.0059	0.0018	3.3452	9.11×10^{-4}
	Chiefs	0.0057	0.0018	3.2271	1.37×10^{-3}
QB	Lions	0.0461	0.0129	3.5742	4.00×10^{-4}
RB	Vikings	0.0302	0.0060	5.0749	6.29×10^{-7}
TE	Eaglesatropts	0.0212	0.0058	3.6562	2.95×10^{-4}
VDU	Browns	0.1153	0.0236	4.8838	1.58×10^{-6}
	Bengals	-0.0786	0.0236	-3.3302	9.60×10^{-4}
WR	Panthers	-0.0371	0.0102	-3.6369	3.17×10^{-4}
	Buccaneers	0.0358	0.0102	3.5068	5.12×10^{-4}



Discussion 2

In each position, 1-4 teams have significantly different salary cap strategies compared to the league average. For example, Detroit Lions invest about 3.02% more of their salary cap in the QB position compared to the league average. Since we control the FWER using Holm-Bonferroni correction, only a few teams are marked as significant, but it is still remarkable that there exists a certain preference and variety among teams' salary cap strategies.

Question 3: How do allocations relate to outcomes?

Objective

The goal of this analysis is to understand how a team's win percentage varies as a function of its financial and draft resource allocation across different positions.

Methodology

Data sources

We combine three datasets describing: 1. Team–Position–Year Spending: for every team, position, and season, this dataset contains (i) Team share of total cap spending on the position (cap_pct_team); (ii) League share of total cap spending represented by that position (cap_pct_lg); (iii) League share of total draft capital invested in the position (draft_pct_lg). 1. Team Win Percentage: Provides each team's seasonal win percentage.

After merging by team and year, each row corresponds to a team–position–year observation.

The data span multiple NFL seasons and provide information on how teams allocate financial resources across positions such as quarterback (QB), running back (RB), offensive line (OL), defensive line (DL), defensive backs (DB), and others. The primary response variable is the team's seasonal win percentage.

Data preparation

The spending dataset is structured at the team–position–year level. Because each team-year has multiple positions (QB, RB, WR, TE, OL, DL, LB, DB, etc.), the raw data is in long format. Linear models require one row per team-year. To analyze how the allocation of spending across positions relates to winning, we performed the following preprocessing steps:

1. Merge spending data with win percentage using team and year as keys.
2. Aggregate spending to create total cap percentage per team-year for exploratory analysis.
3. Reshape the dataset to wide format, where each row corresponds to a team-season and columns represent spending proportions for each position.
4. Standardize variable names and ensure completeness, replacing missing position values with zero for teams that spent nothing at a specific position that year.

This creates three predictors for each position, capturing: 1. Team-Level Spending: How each team allocates cap resources internally. 1. League-Level Spending: How expensive the position is league-wide in a given year. 1. Draft Importance: How much draft capital the league invests in that position.

The response variable is the team's season win percentage.

```
# ------------------------------------------------------------
# 1. Read data
# ------------------------------------------------------------

cap_team <- read_csv("../data/processed/capital_by_position_team_year.csv")
cap_year <- read_csv("../data/processed/capital_by_position_year.csv")
win_pct <- read_csv("../data/processed/win_pct_season.csv")

# Standardize names
cap_team <- cap_team %>% rename(team = team, year = year)
win_pct  <- win_pct %>% rename(team = Team, year = year)

# Merge
df <- cap_team %>%
  left_join(win_pct, by = c("team", "year"))
```

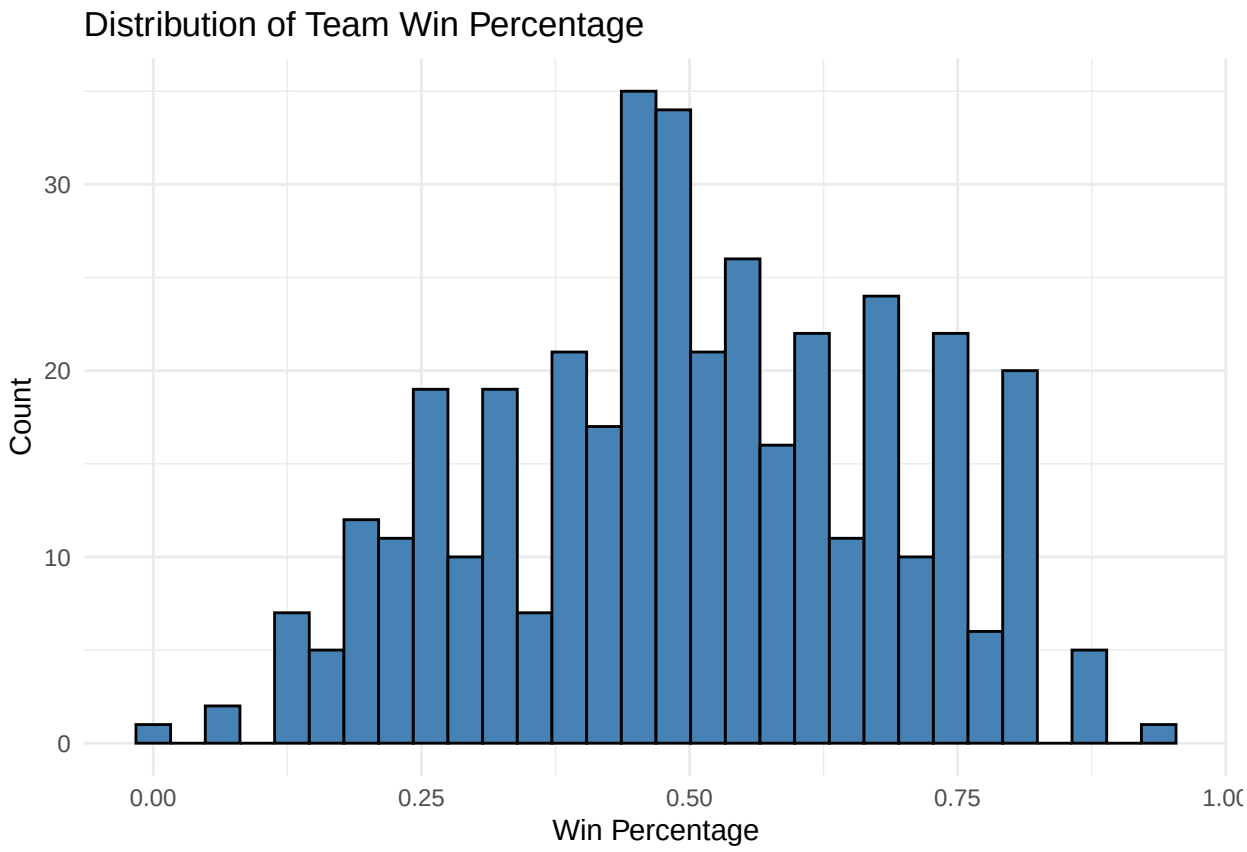
```r
df_wide <- df %>%
  dplyr::select(team, year, position, cap_pct_team, cap_pct_lg, draft_pct_lg, win_pct) %>%
  pivot_wider(
    names_from = position,
    values_from = c(cap_pct_team, cap_pct_lg, draft_pct_lg),
    values_fill = 0
  )
```

```r
# Identify columns NOT to use as predictors
non_predictors <- c("team", "year", "win_pct")

# Identify predictor columns
predictors <- setdiff(names(df_wide), non_predictors)
```

Exploratory Data Analysis

Distribution of win percentage:

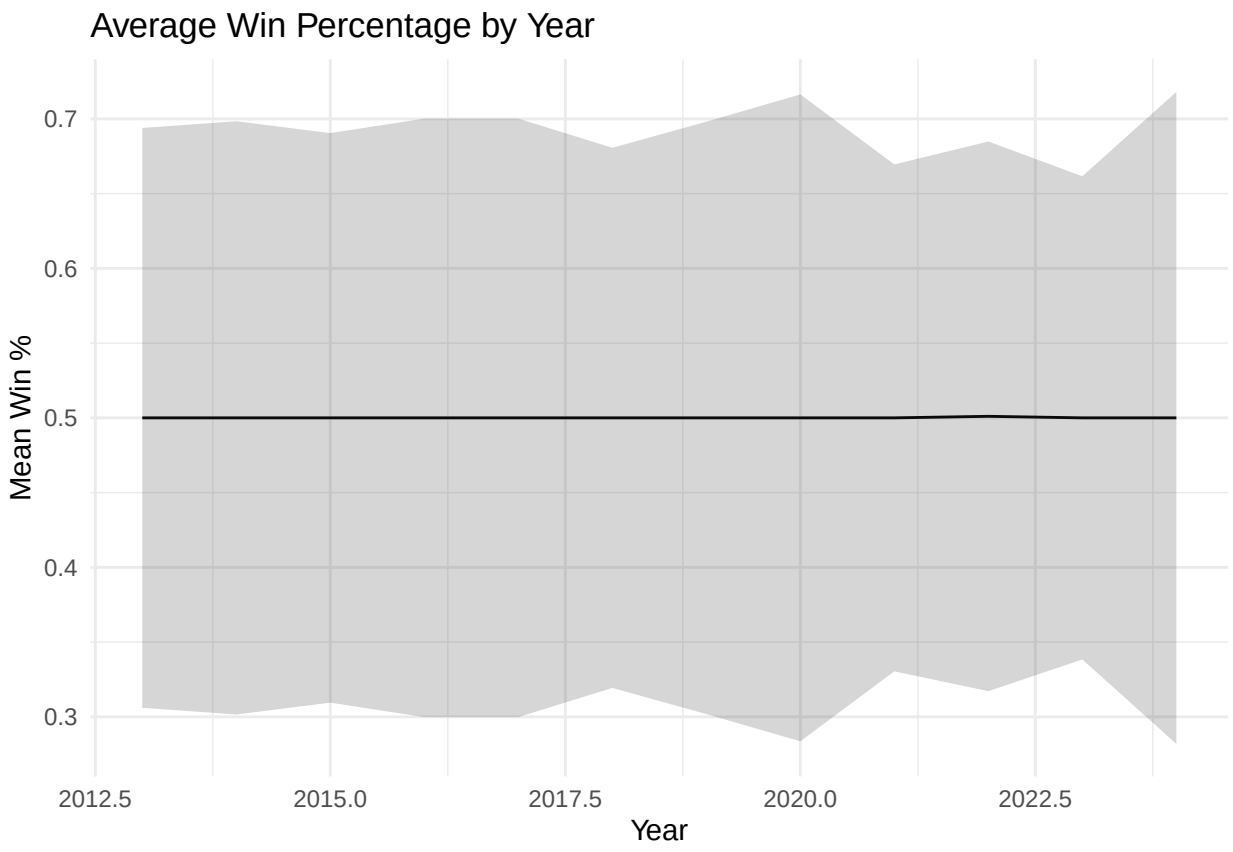
```r
ggplot(df_wide, aes(x = win_pct)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  labs(title = "Distribution of Team Win Percentage",
       x = "Win Percentage", y = "Count") +
  theme_minimal()
```



Win percentage by year:

```r
df_wide %>%
  group_by(year) %>%
```

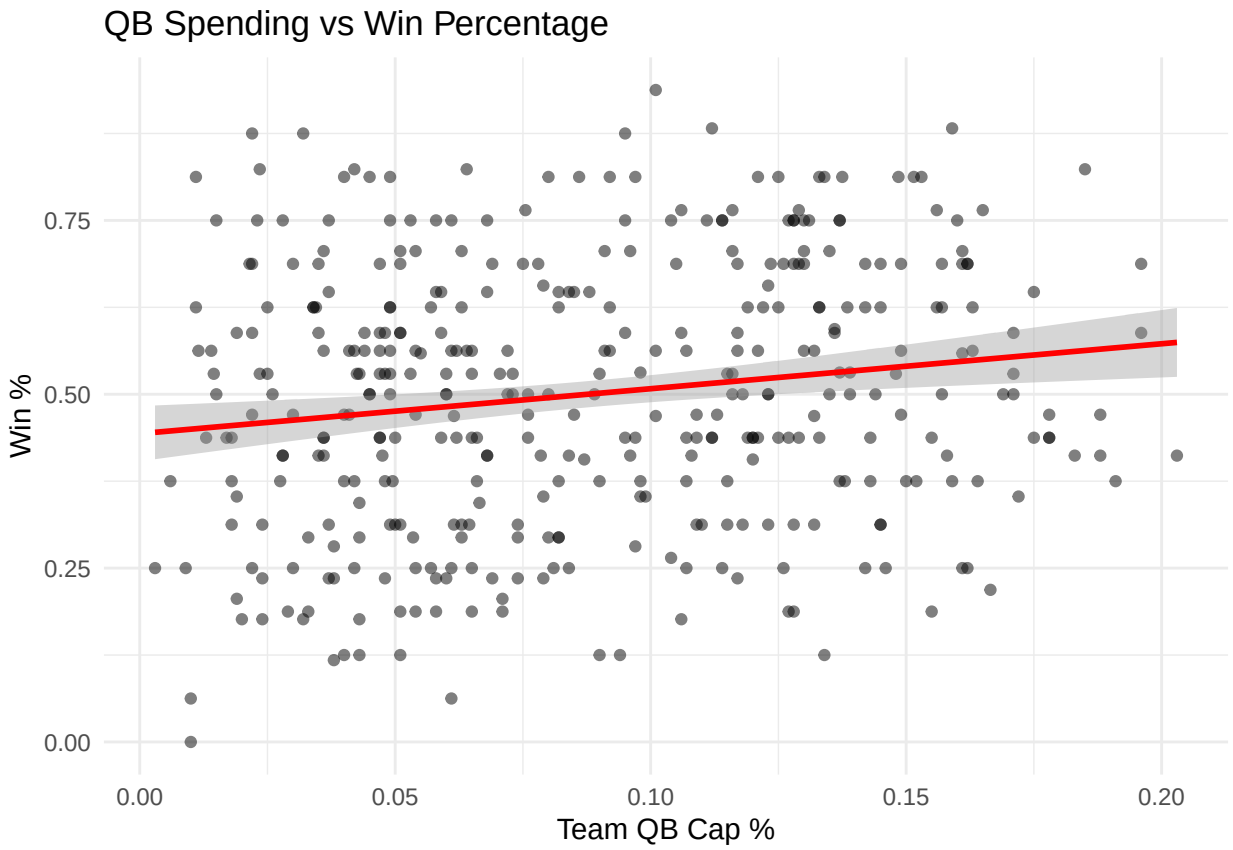
```r
  summarise(mean_win_pct = mean(win_pct, na.rm = TRUE),
            sd_win_pct = sd(win_pct, na.rm = TRUE)) %>%
  ggplot(aes(x = year, y = mean_win_pct)) +
  geom_line() +
  geom_ribbon(aes(ymin = mean_win_pct - sd_win_pct,
                  ymax = mean_win_pct + sd_win_pct), alpha = 0.2) +
  labs(title = "Average Win Percentage by Year",
       x = "Year", y = "Mean Win %") +
  theme_minimal()
```



Relationship between QB spending and winning

```r
ggplot(df_wide, aes(x = cap_pct_team_QB, y = win_pct)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "QB Spending vs Win Percentage",
       x = "Team QB Cap %", y = "Win %") +
  theme_minimal()
```

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



Model fitting

Three modeling approaches were used to evaluate the relationship between positional spending and win percentage:

Full Linear regression model

We start with a full multiple linear regression model including all position-level predictors:

$$\text{win\_pct}_{t,y} = \beta_0 + \sum_p \left[\beta_p^{(T)} \cdot \text{cap\_pct\_team}_{p,t,y} + \beta_p^{(L)} \cdot \text{cap\_pct\_lg}_{p,t,y} + \beta_p^{(D)} \cdot \text{draft\_pct\_lg}_{p,t,y} \right] + \epsilon_{t,y}$$

This model can be used to estimate: 1. Which positions are associated with improved performance when a team invests more. 1. How positional value across the league relates to winning. 1. Whether high-draft positions correspond to on-field success.

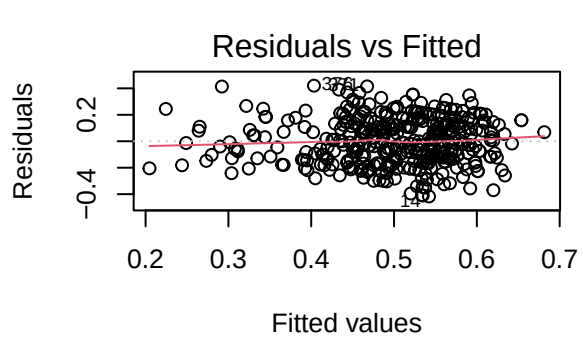
This model is high-dimensional and likely collinear (positions are often interdependent), so further model refinement is used.

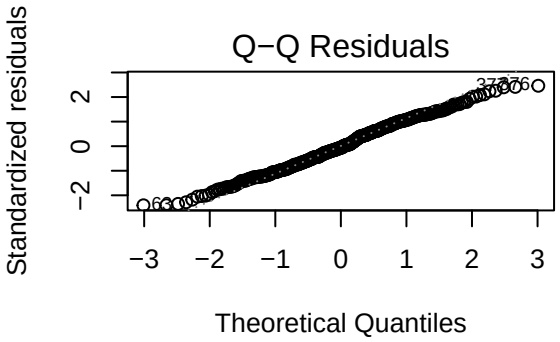
Standard diagnostic checks (residuals vs fitted, Q–Q plot, scale–location plot, leverage plot) are used to assess assumptions of homoscedasticity, normality, and influence.

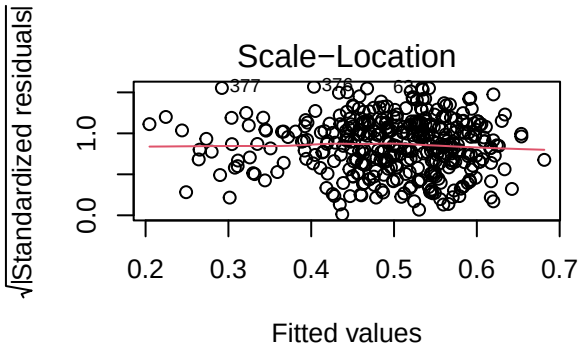
```
formula_lm <- reformulate(termlabels = predictors, response = "win_pct")

model_lm <- lm(formula_lm, data = df_wide)

summary(model_lm)
```

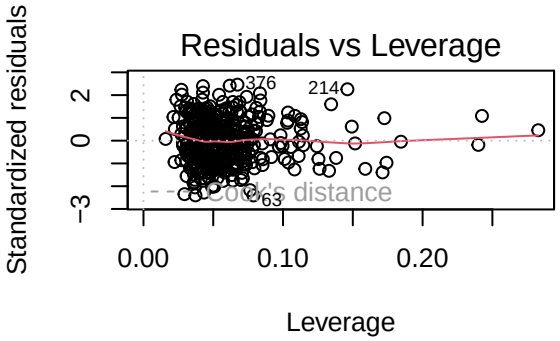
```
##
## Call:
## lm(formula = formula_lm, data = df_wide)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41723 -0.12785 -0.00911  0.13192  0.42021
##
## Coefficients: (14 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.366653   0.235086   1.560  0.11972
## cap_pct_team_DB     0.338741   0.323865   1.046  0.29629
## cap_pct_team_DL     0.330694   0.312422   1.058  0.29054
## cap_pct_team_K      1.642760   1.200824   1.368  0.17215
## cap_pct_team_LB     0.116629   0.401178   0.291  0.77144
## cap_pct_team_LS    -4.650260   5.234148  -0.888  0.37489
## cap_pct_team_OL     0.387456   0.329504   1.176  0.24042
## cap_pct_team_P      1.361874   1.447612   0.941  0.34745
## cap_pct_team_QB     0.446931   0.301088   1.484  0.13858
## cap_pct_team_RB     0.451448   0.453263   0.996  0.31992
## cap_pct_team_TE     0.487095   0.508531   0.958  0.33878
## cap_pct_team_VDU   -0.337378   0.249843  -1.350  0.17775
## cap_pct_team_WR           NA         NA      NA       NA
## cap_pct_lg_DB             NA         NA      NA       NA
## cap_pct_lg_DL             NA         NA      NA       NA
## cap_pct_lg_K              NA         NA      NA       NA
## cap_pct_lg_LB             NA         NA      NA       NA
## cap_pct_lg_LS             NA         NA      NA       NA
## cap_pct_lg_OL             NA         NA      NA       NA
## cap_pct_lg_P              NA         NA      NA       NA
## cap_pct_lg_QB             NA         NA      NA       NA
## cap_pct_lg_RB             NA         NA      NA       NA
## cap_pct_lg_TE             NA         NA      NA       NA
## cap_pct_lg_VDU            NA         NA      NA       NA
## cap_pct_lg_WR             NA         NA      NA       NA
## draft_pct_lg_DB    -4.173516   2.115002  -1.973  0.04922 *
## draft_pct_lg_DL    -0.981657   2.066487  -0.475  0.63505
## draft_pct_lg_K      7.361489  16.317605   0.451  0.65216
## draft_pct_lg_LB     1.424724   2.568658   0.555  0.57947
## draft_pct_lg_LS    53.521926  39.065135   1.370  0.17152
## draft_pct_lg_OL    -0.945120   2.211651  -0.427  0.66939
## draft_pct_lg_P    -18.960624  17.860057  -1.062  0.28912
## draft_pct_lg_QB    -8.051625   2.471122  -3.258  0.00123 **
## draft_pct_lg_RB     0.006443   3.546233   0.002  0.99855
## draft_pct_lg_TE    -4.546276   3.771078  -1.206  0.22878
## draft_pct_lg_VDU          NA         NA      NA       NA
## draft_pct_lg_WR    -3.412935   2.452599  -1.392  0.16491
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.177 on 361 degrees of freedom
## Multiple R-squared:  0.1866, Adjusted R-squared:  0.137
## F-statistic: 3.765 on 22 and 361 DF,  p-value: 5.863e-08
```

```r
# ------------------------------
# Diagnostic Plots for LM
# ------------------------------
library(ggplot2)

# Base R diagnostic plots (simple and standard)
par(mfrow=c(2,2))
plot(model_lm)
```





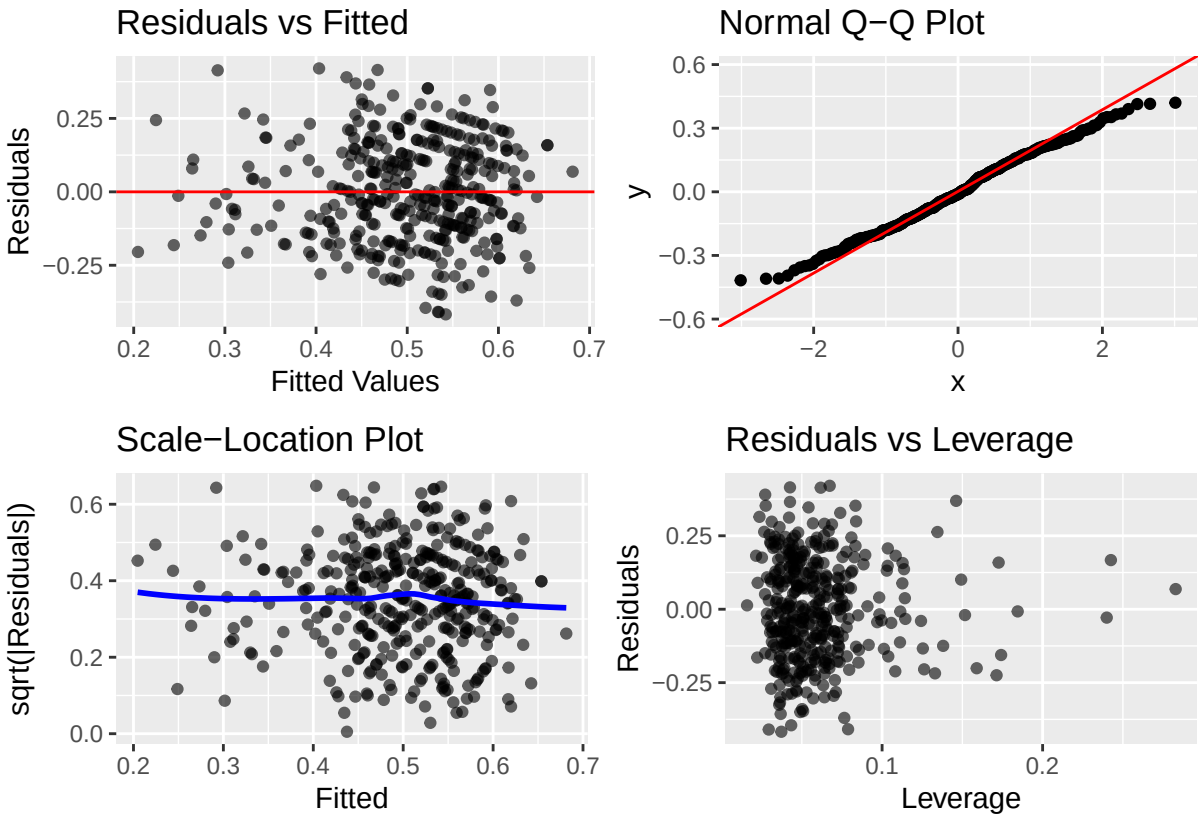





```r
par(mfrow=c(1,1))

# Custom ggplot2 diagnostic plots -------------------------

# Residuals vs Fitted
p1 <- ggplot(data.frame(fitted = fitted(model_lm),
                        residuals = resid(model_lm)),
             aes(fitted, residuals)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title="Residuals vs Fitted",
       x="Fitted Values", y="Residuals")

# QQ plot
p2 <- ggplot(data.frame(sample = resid(model_lm)),
             aes(sample = sample)) +
  stat_qq() +
  stat_qq_line(color="red") +
  labs(title="Normal Q-Q Plot")
```

```r
# Scale-Location Plot
p3 <- ggplot(data.frame(fitted = fitted(model_lm),
                        sqrt_resid = sqrt(abs(resid(model_lm)))),
             aes(fitted, sqrt_resid)) +
  geom_point(alpha=0.6) +
  geom_smooth(method="loess", se=FALSE, color="blue") +
  labs(title="Scale-Location Plot",
       y="sqrt(|Residuals|)", x="Fitted")

# Residuals vs Leverage
lev <- hatvalues(model_lm)
p4 <- ggplot(data.frame(lev = lev, residuals = resid(model_lm)),
             aes(lev, residuals)) +
  geom_point(alpha=0.6) +
  labs(title="Residuals vs Leverage",
       x="Leverage", y="Residuals")

# Display all plots
library(patchwork)
(p1 | p2) / (p3 | p4)
```

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



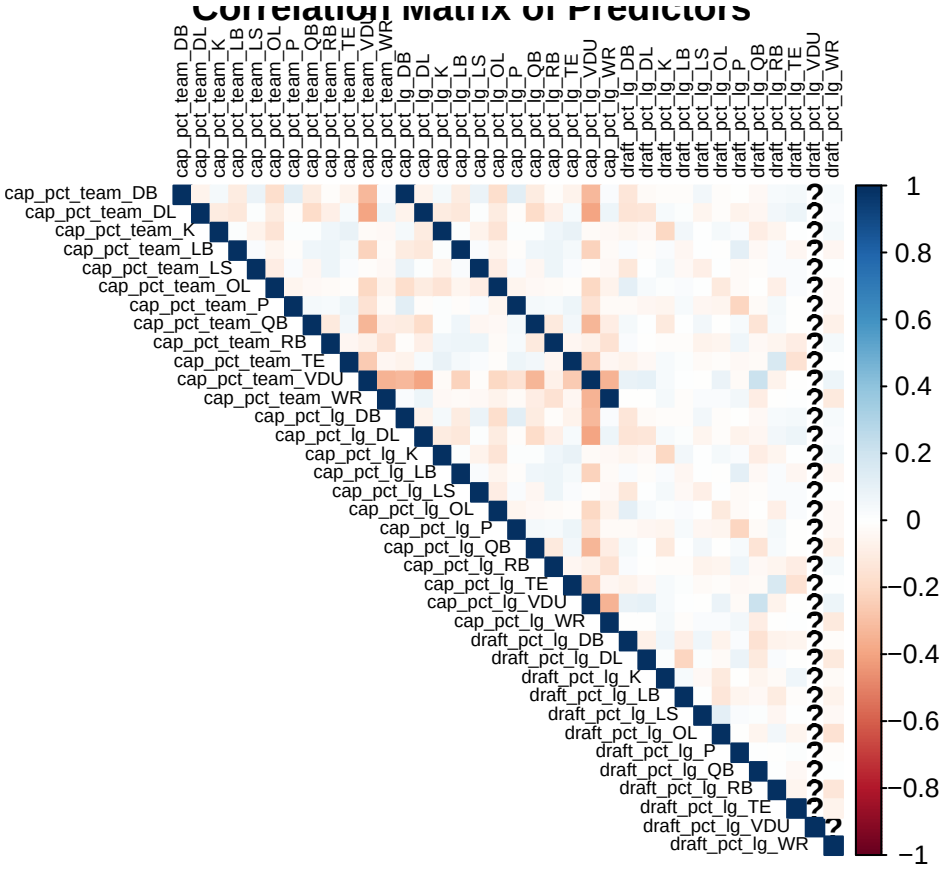
The full linear model suffers from severe multicollinearity (14 undefined coefficients). Among the estimable coefficients: 1. draft_pct_lg_QB has a negative coefficient (-8.05, p=0.001), suggesting that in years when the league invests more draft capital in QBs, team win percentages are lower. This might be because (i) Teams drafting QBs early are often rebuilding (poor teams); (ii) Rookie QBs may hurt win percentage in their first year. 1. draft_pct_lg_DB (-4.17, p=0.049) shows a similar negative pattern. 1. Team-level spending

coefficients are all positive but non-significant, likely due to multicollinearity, as indicated in the plot below.

```
cor_matrix <- cor(df_wide[, predictors], use = "complete.obs")
```

```
## Warning in cor(df_wide[, predictors], use = "complete.obs"): the standard
## deviation is zero
```

```
corrplot(cor_matrix, method = "color", type = "upper",
         tl.cex = 0.6, tl.col = "black",
         title = "Correlation Matrix of Predictors")
```



Stepwise AIC model

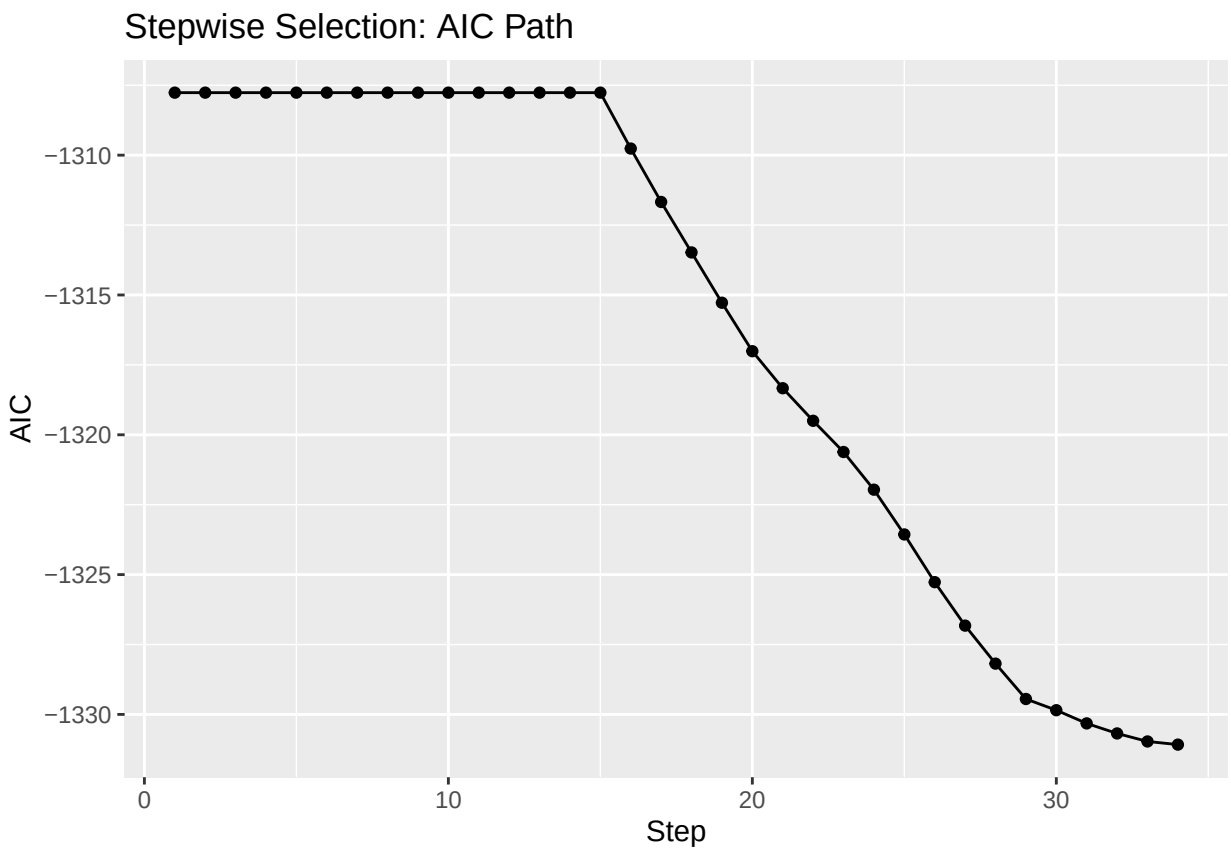
To obtain a more interpretable model, we apply bidirectional stepwise selection using Akaike Information Criterion (AIC). We start from the full model, iteratively add or remove predictors, and stop when AIC cannot be reduced further. This yields a reduced linear model that balances model complexity with predictive accuracy. The resulting model highlights positions and spending metrics most strongly associated with win percentage.

```
step_model <- MASS::stepAIC(model_lm, direction = "both", trace = FALSE)
summary(step_model)
```

```
##
## Call:
## lm(formula = win_pct ~ cap_pct_team_VDU + draft_pct_lg_DB + draft_pct_lg_LS +
##     draft_pct_lg_QB + cap_pct_team_WR, data = df_wide)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.45097 -0.13052 -0.01217  0.13087  0.46606
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.70141    0.03718  18.863  < 2e-16 ***
## cap_pct_team_VDU -0.73072    0.11178  -6.537 2.04e-10 ***
## draft_pct_lg_DB  -3.73591    1.97303  -1.893 0.059056 .
## draft_pct_lg_LS  53.72618   37.94048   1.416 0.157580
## draft_pct_lg_QB  -8.06957    2.34533  -3.441 0.000645 ***
## cap_pct_team_WR  -0.39134    0.25181  -1.554 0.120998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1754 on 378 degrees of freedom
## Multiple R-squared:  0.1637, Adjusted R-squared:  0.1526
## F-statistic: 14.79 on 5 and 378 DF,  p-value: 2.91e-13
```

```r
aic_values <- data.frame(
  Step = seq_len(nrow(step_model$anova)),
  AIC = step_model$anova$AIC
)

ggplot(aic_values, aes(Step, AIC)) +
  geom_line() +
  geom_point() +
  labs(title="Stepwise Selection: AIC Path")
```



The stepwise AIC procedure selected five predictors: 1. cap_pct_team_VDU (-0.73, p<0.001): Strong

negative effect. Teams spending more on VDU have lower win percentages. This likely reflects poor cap management or dead money from released players. 1. draft_pct_lg_QB (-8.07, p<0.001): Confirms the full model finding. League-wide QB draft investment negatively predicts win percentage. 1. draft_pct_lg_DB (-3.74, p=0.059): Marginally significant negative effect for defensive backs draft capital. 1. cap_pct_team_WR (-0.39, p=0.12): Non-significant but retained by AIC. Suggests spending more on WRs may not improve winning. 1. draft_pct_lg_LS (53.73, p=0.16): Large but non-significant. Long snappers represent tiny spending, so high variance is expected.

An R^2 of 0.164 indicates that these five variables explain about 16% of the variance in win percentage.

```
model_comparison <- data.frame(
  Model = c("Full Model", "Stepwise Model"),
  R_squared = c(summary(model_lm)$r.squared,
                summary(step_model)$r.squared),
  Adj_R_squared = c(summary(model_lm)$adj.r.squared,
                    summary(step_model)$adj.r.squared),
  AIC = c(AIC(model_lm), AIC(step_model)),
  BIC = c(BIC(model_lm), BIC(step_model)),
  Num_Predictors = c(22, 5)
)

print(model_comparison)
```

```
##            Model R_squared Adj_R_squared       AIC       BIC Num_Predictors
## 1     Full Model 0.1866112     0.1370418 -216.0195 -121.2041             22
## 2 Stepwise Model 0.1636610     0.1525983 -239.3348 -211.6803              5
```

Elastic net regularized regression

Because the predictor set is large and columns are highly correlated (e.g., teams that spend heavily on secondary positions may also spend heavily on linebackers), we also fit an elastic net model. Elastic net combines L1 (lasso) and L2 (ridge) penalties:

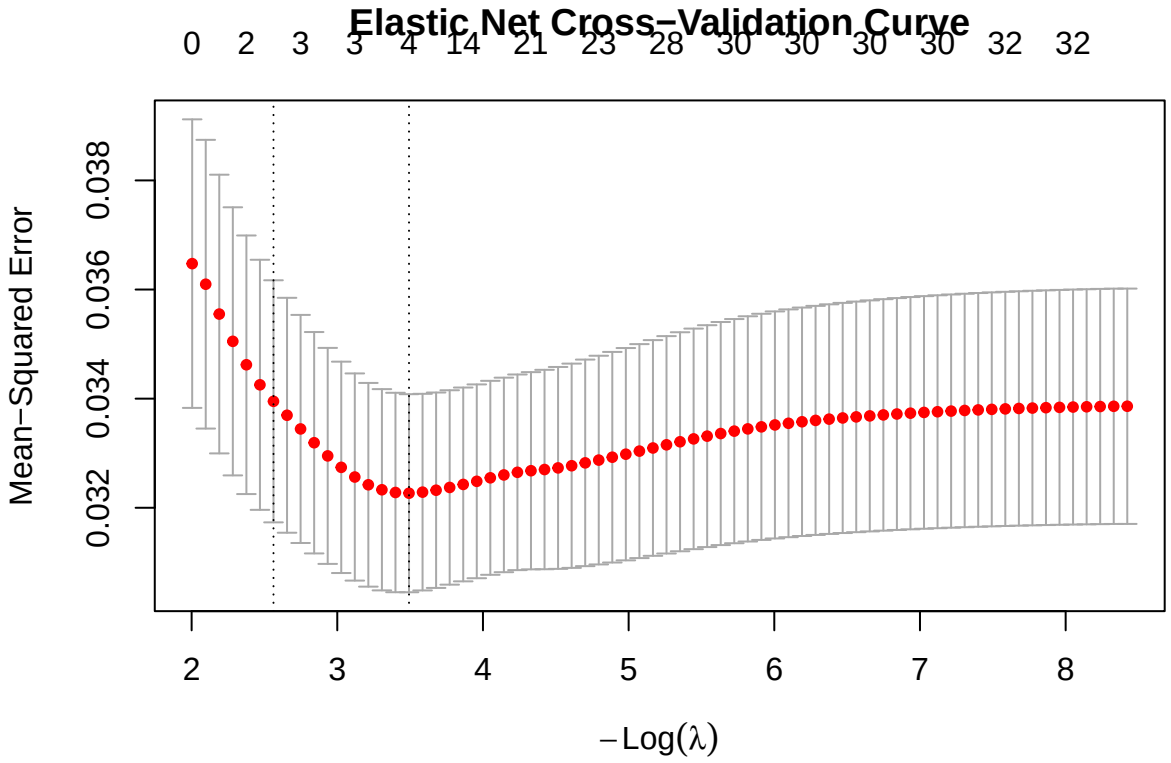
$$\hat{\beta} = \arg\min_{\beta} \left[\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \left(\alpha \|\beta\|_1 + (1-\alpha)\|\beta\|_2^2 \right) \right]$$

We set $\alpha = 0.5$, i.e. give equal weights to lasso and ridge penalties.

This approach automatically (i) Handles multicollinearity; (ii) Performs automatic variable selection; (iii) Shrinks unstable coefficients toward zero; (iv) Stabilizes estimates across correlated position groups.

Further, we use a 10-fold cros validation to select the optimal tuning parameter λ.

```
# Matrix of predictors
X <- df_wide %>% dplyr::select(all_of(predictors)) %>% as.matrix()

# Response vector
y <- df_wide$win_pct

set.seed(123)
cv_mod <- cv.glmnet(X, y, alpha = 0.5)

plot(cv_mod)
title("Elastic Net Cross-Validation Curve")
```




```
best_lambda <- cv_mod$lambda.min
best_lambda
```

```
## [1] 0.0304284
```

```
elastic_mod <- glmnet(X, y, alpha = 0.5, lambda = best_lambda)

print(elastic_mod)
```

```
##
## Call:  glmnet(x = X, y = y, alpha = 0.5, lambda = best_lambda)
##
##   Df  %Dev  Lambda
## 1  4 13.39 0.03043
```

```
coef(elastic_mod)
```

```
## 37 x 1 sparse Matrix of class "dgCMatrix"
##                             s0
## (Intercept)       0.60192584
## cap_pct_team_DB   .
## cap_pct_team_DL   .
## cap_pct_team_K    .
## cap_pct_team_LB   .
## cap_pct_team_LS   .
## cap_pct_team_OL   .
## cap_pct_team_P    .
## cap_pct_team_QB   .
## cap_pct_team_RB   .
## cap_pct_team_TE   .
## cap_pct_team_VDU -0.27143579
```

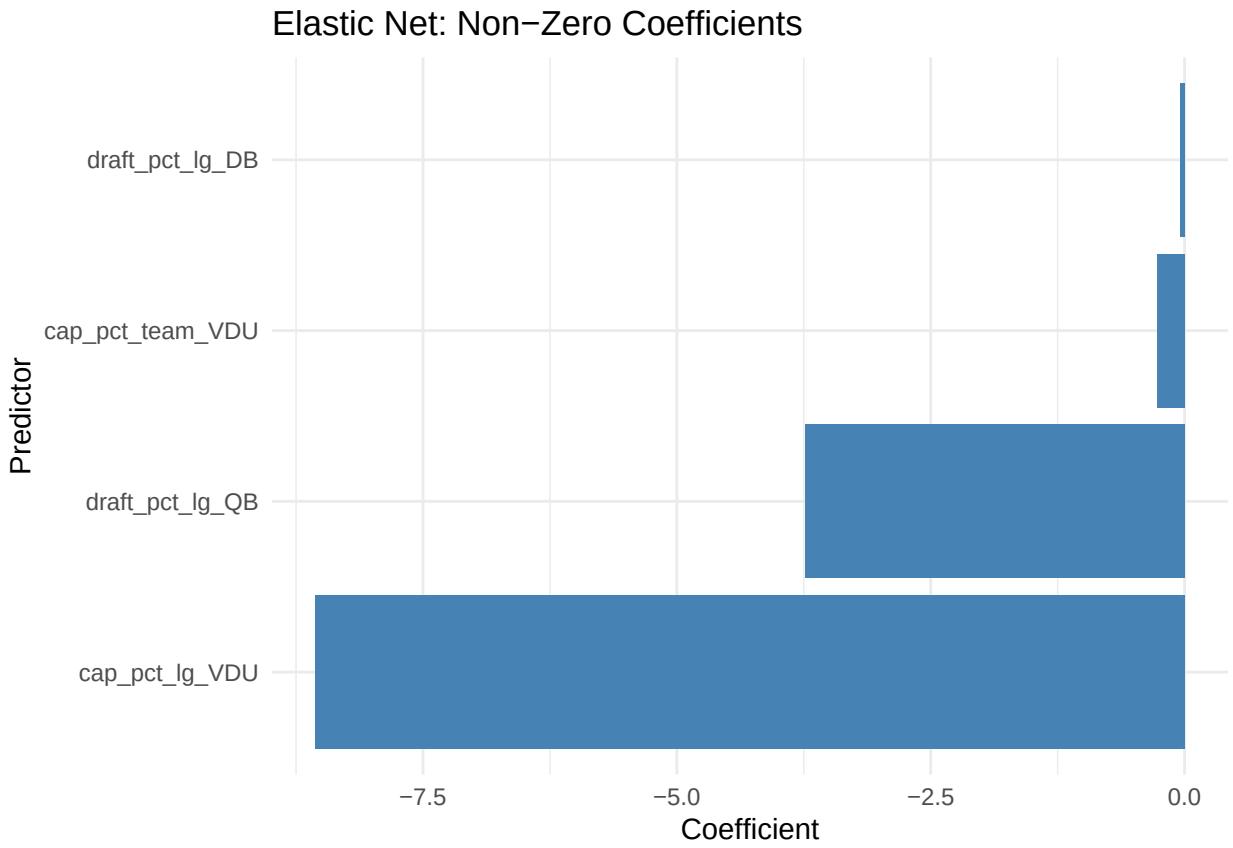
```
## cap_pct_team_WR   .
## cap_pct_lg_DB     .
## cap_pct_lg_DL     .
## cap_pct_lg_K      .
## cap_pct_lg_LB     .
## cap_pct_lg_LS     .
## cap_pct_lg_OL     .
## cap_pct_lg_P      .
## cap_pct_lg_QB     .
## cap_pct_lg_RB     .
## cap_pct_lg_TE     .
## cap_pct_lg_VDU   -8.55877766
## cap_pct_lg_WR     .
## draft_pct_lg_DB  -0.04886692
## draft_pct_lg_DL   .
## draft_pct_lg_K    .
## draft_pct_lg_LB   .
## draft_pct_lg_LS   .
## draft_pct_lg_OL   .
## draft_pct_lg_P    .
## draft_pct_lg_QB  -3.74064112
## draft_pct_lg_RB   .
## draft_pct_lg_TE   .
## draft_pct_lg_VDU  .
## draft_pct_lg_WR   .
```

The elastic net selected only 4 non-zero coefficients (plus intercept), providing the most parsimonious model:

1. cap_pct_team_VDU (-0.27): Negative effect of VDU spending (shrunk from -0.73 in stepwise)
2. cap_pct_lg_VDU (-8.56): League-wide VDU spending also negative
3. draft_pct_lg_DB (-0.05): Very small negative effect
4. draft_pct_lg_QB (-3.74): QB draft capital remains negative (shrunk from -8.07)

The elastic net dramatically shrinks most coefficients to zero, keeping only the most stable predictors. The model explains 13.4% of deviance.

```r
elastic_coefs <- as.matrix(coef(elastic_mod))
elastic_coefs_df <- data.frame(
  term = rownames(elastic_coefs),
  estimate = elastic_coefs[,1]
) %>%
  filter(estimate != 0, term != "(Intercept)")

ggplot(elastic_coefs_df, aes(x = reorder(term, estimate), y = estimate)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Elastic Net: Non-Zero Coefficients",
       x = "Predictor", y = "Coefficient") +
  theme_minimal()
```




```
predictions_df <- data.frame(
  actual = df_wide$win_pct,
  lm_pred = predict(model_lm, df_wide),
  step_pred = predict(step_model, df_wide),
  elastic_pred = as.vector(predict(elastic_mod, newx = X))
)

# Calculate RMSE for each model
rmse <- predictions_df %>%
  summarise(
    RMSE_Full = sqrt(mean((actual - lm_pred)^2)),
    RMSE_Stepwise = sqrt(mean((actual - step_pred)^2)),
    RMSE_Elastic = sqrt(mean((actual - elastic_pred)^2))
  )

print(rmse)
```

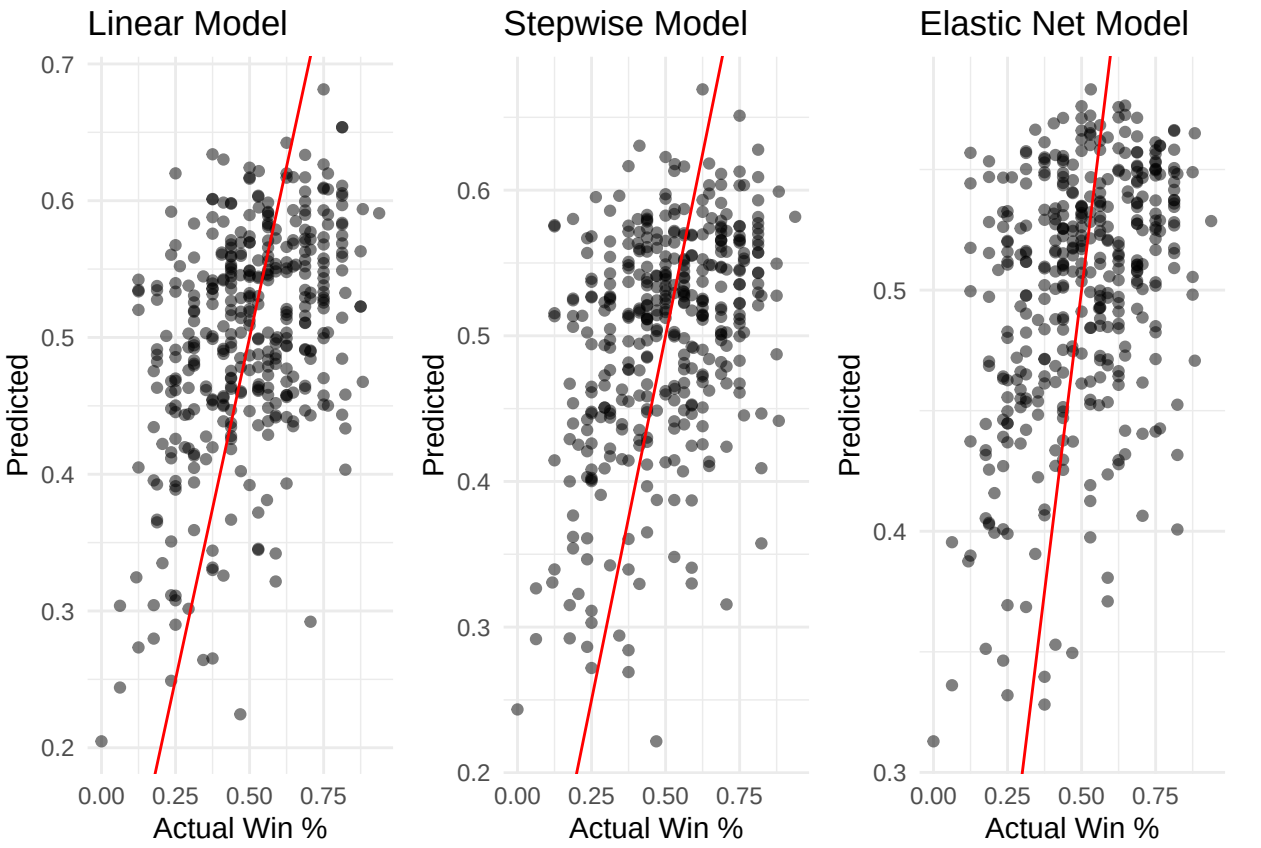
```
##   RMSE_Full RMSE_Stepwise RMSE_Elastic
## 1 0.1715786     0.1739823    0.1770462
```

```
# Scatter plots of predicted vs actual
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
p0 <- ggplot(predictions_df, aes(actual, lm_pred)) +
  geom_point(alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Linear Model", x = "Actual Win %", y = "Predicted") +
  theme_minimal()

p1 <- ggplot(predictions_df, aes(actual, step_pred)) +
  geom_point(alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Stepwise Model", x = "Actual Win %", y = "Predicted") +
  theme_minimal()

p2 <- ggplot(predictions_df, aes(actual, elastic_pred)) +
  geom_point(alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Elastic Net Model", x = "Actual Win %", y = "Predicted") +
  theme_minimal()

grid.arrange(p0, p1, p2, ncol = 3)
```



Discussion 3

This analysis examined the relationship between NFL team resource allocation (salary cap spending and draft capital) across positions and team win percentage over multiple seasons. Three modeling approaches (linear models, stepwise AIC model selection, and elastic net) were employed to handle the high-dimensional, collinear nature of positional spending data.

Model comparison

1. The full linear model suffered from perfect multicollinearity, with 14 undefined coefficients due to spending constraints.
2. Stepwise selection identified 5 predictors, achieving parsimony while maintaining exploratory power (Adj. $R^2 = 0.153$).
3. Elastic net regularization provided the most conservative model with only 4 non-zero coefficients, prioritizing prediction stability over fit.

Key findings:

1. Limited Predictive Power: All models explain only 13-16% of variance in win percentage, suggesting that resource allocation alone is not a strong predictor of team success. Other factors such as coaching, player health, schedule strength, and execution likely dominate winning.
2. Quarterback Draft Capital: The most consistent finding across all models is a negative relationship between league-wide QB draft investment and team win percentage. This result likely reflects that (a) QB-needy teams are often already struggling, and (b) rookie QBs typically don't contribute to winning immediately.
3. VDU Spending as a Red Flag: Teams allocating higher percentages to VDU positions, likely representing dead money or cap penalties, show significantly lower win percentages. This confirms that inefficient cap management hurts performance.
4. Defensive Back Investment: League-wide draft investment in defensive backs shows a negative association with winning, though this effect is less stable than the QB finding.
5. No Significant Positive Associations: Surprisingly, no position showed a strong, significant positive relationship between team spending and winning. This suggests that execution and player quality matter more than raw allocation.

Limitations

1. The analysis did not account for player quality, coaching effectiveness, or injury impacts.
2. Year-to-year carryover effects (e.g., drafted players contributing across multiple seasons) are not captured.

Discussion

Please find each discussion in the end of each analysis.

Acknowledgement

Jack performed the data preparation and exploratory analysis. The analysis of how allocations change over time (Question 1) was completed by Soumyajyoti, the study of how allocations vary with teams (Question 2) was carried out by Jaehyuk, and the analysis linking allocations to outcomes (Question 3) was done by Aniruddhan. Furthermore, ChatGPT was used to assist with coding and the overall structure of this report.

Responses to peer review comments

The issue of multi collinearity raised by Beichen has been addressed in question 3 (predicting outcomes from the given set of predictors). We had 14 columns with perfect multicollinearity, which had undefined coefficients in the full linear model. This was taken care of after using stepwise model selection and elastic net regularization.

Ruocheng's comment on confounders has also been answered in question 3: our best performing model has a predictive power of only about 17%, indicating that factors other than just draft and salary have a more significant role to play in predicting the performance of teams. This includes factors like ability of coach, training strength, etc.

Along the lines of Mark's suggestion, we have analyzed the change of allocations with time using time series methods in the final report.

Section 5.1 of Part 1 of the report talks about combining positions, partly answering David's concern about there being a lot of positions. We avoid the potential double counting by creating a uniform positional mapping (more details in section 5.1).