Sample Work

Jake Blumengarten

2025-02-12

```r
knitr::opts_chunk$set(echo = TRUE)
path <- file.path("/Users/jakeblumengarten/Downloads/Blitz 2025/InjuryData.csv")
path2 <- file.path("/Users/jakeblumengarten/Downloads/Blitz 2025/KickingData.csv")
InjuryData <- read.csv(path)
kickingdata <- read.csv(path2)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

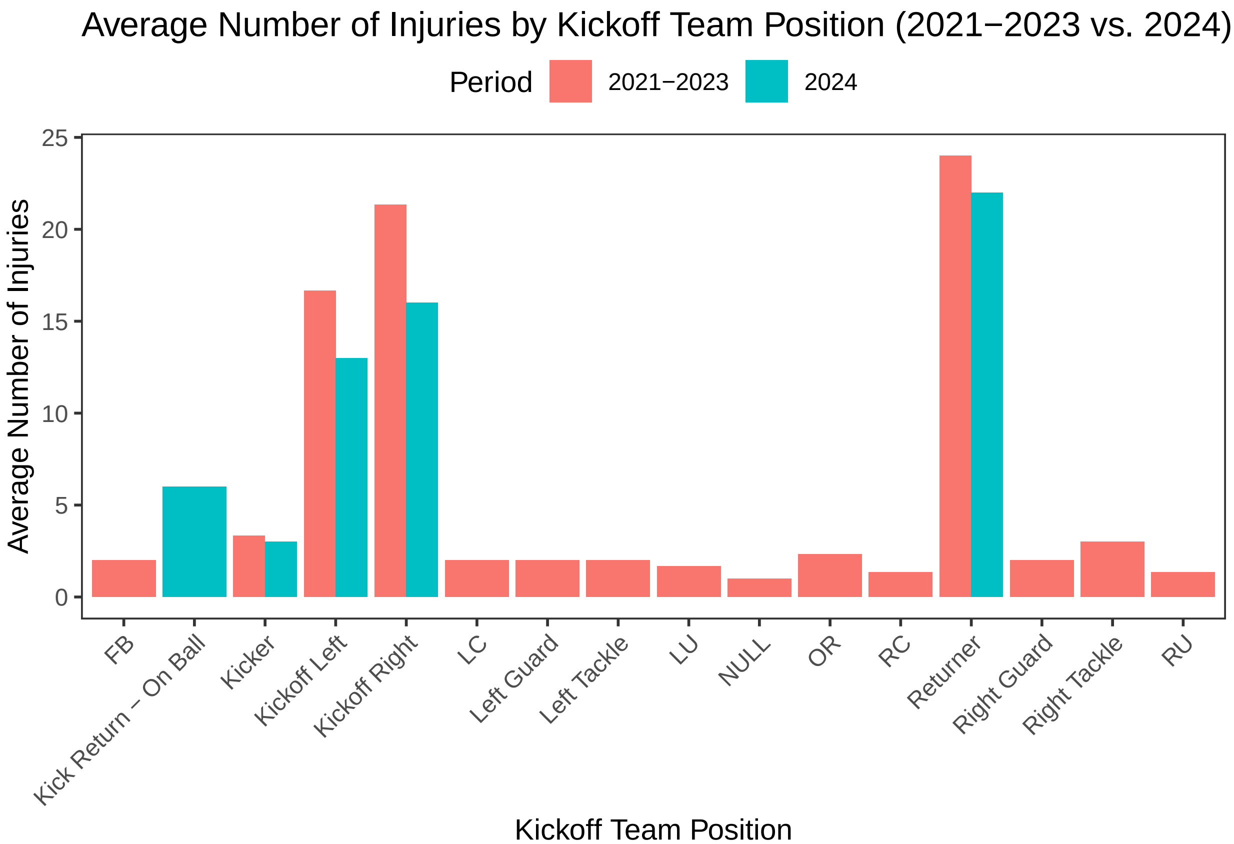
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Differences in Contact Injuries from 2021-2023 vs 2024

```r
InjuryData %>%
  filter(!is.na(Alignment)) %>%
  mutate(Period = ifelse(Season >= 2021 & Season <= 2023, "2021-2023", "2024")) %>%
  group_by(Period, Alignment) %>%
  summarise(avg_injury_count = n() / n_distinct(Season)) %>%
  ggplot(aes(x = Alignment, y = avg_injury_count, fill = Period)) +
  geom_col(position = "dodge") +
  theme_test() +
  labs(title = "Average Number of Injuries by Kickoff Team Position (2021-2023 vs. 2024)",
       x = "Kickoff Team Position",
```

```r
      y = "Average Number of Injuries") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "top")
```

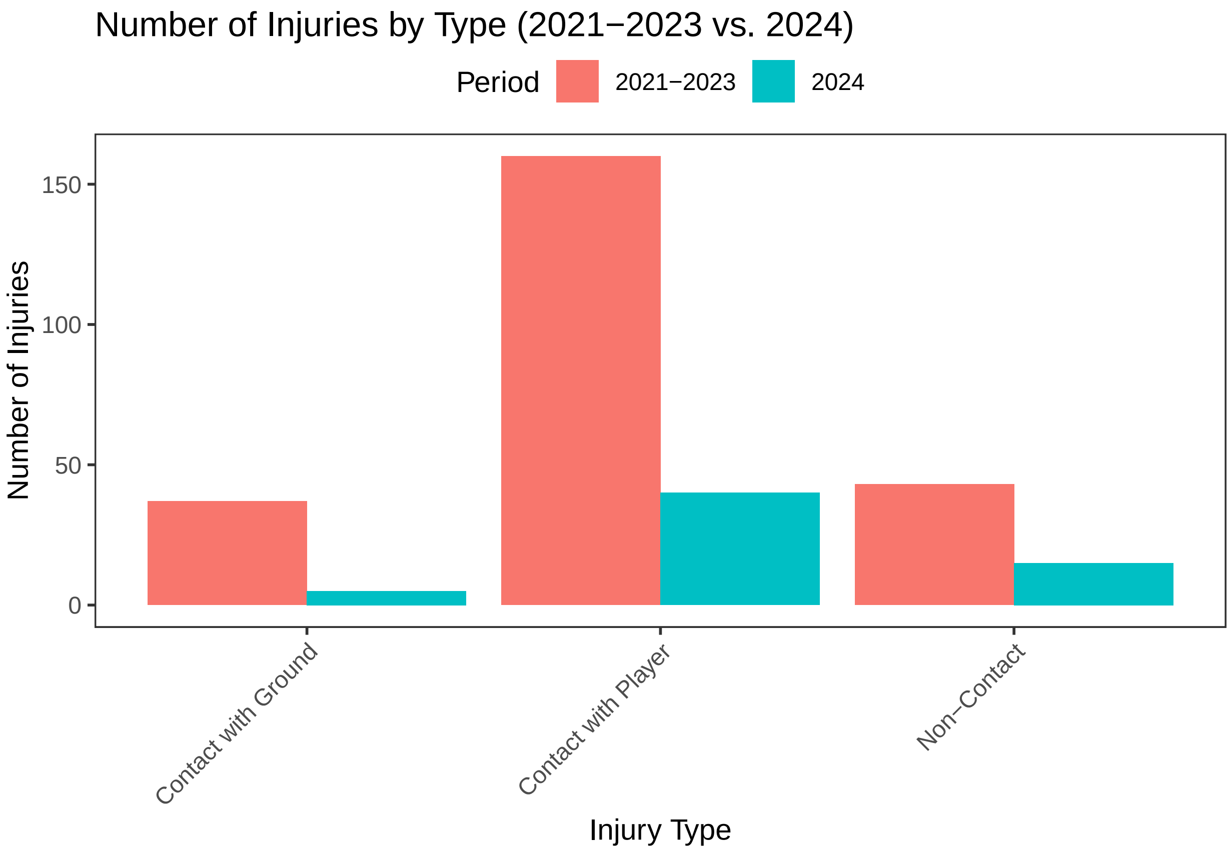
```
## `summarise()` has grouped output by 'Period'. You can override using the
## `.groups` argument.
```



Which position on the kickoff team gets injured the most?

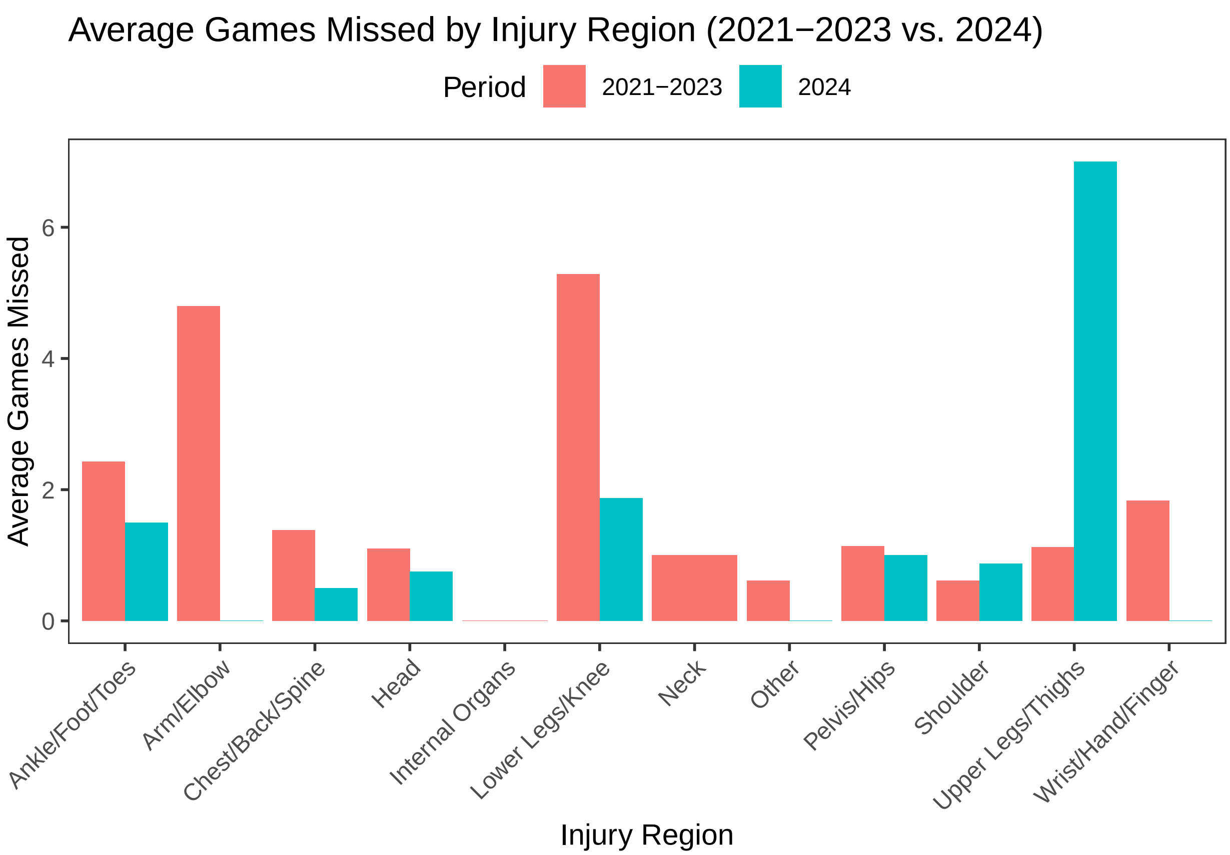
```r
InjuryData %>%
  filter(!is.na(Injury_Type)) %>%
  mutate(Period = ifelse(Season >= 2021 & Season <= 2023, "2021-2023", "2024")) %>%
  group_by(Period, Injury_Type) %>%
  summarise(injury_count = n()) %>%
  ggplot(aes(x = Injury_Type, y = injury_count, fill = Period)) +
  geom_col(position = "dodge") +
  theme_test() +
  labs(title = "Number of Injuries by Type (2021-2023 vs. 2024)",
       x = "Injury Type",
       y = "Number of Injuries") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "top")
```

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



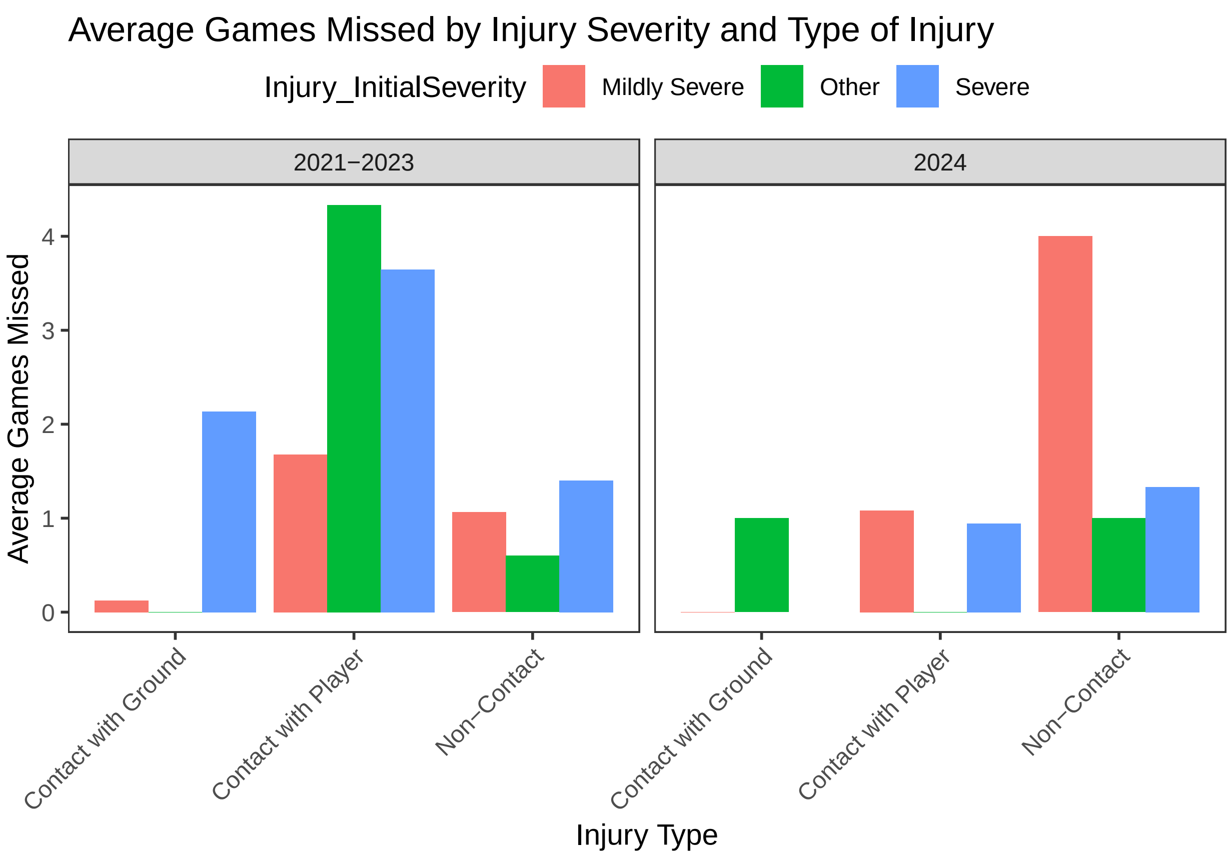
Differences in Contact Injuries from 2021-2023 vs 2024

```
InjuryData %>%
  filter(!is.na(Injury_Region), !is.na(GamesMissed)) %>%
  mutate(GamesMissed = as.numeric(GamesMissed)) %>%
  mutate(Period = ifelse(Season >= 2021 & Season <= 2023, "2021-2023", "2024")) %>%
  group_by(Period, Injury_Region) %>%
  summarise(avg_GamesMissed = mean(GamesMissed, na.rm = TRUE), .groups = "drop") %>%
  ggplot(aes(x = Injury_Region, y = avg_GamesMissed, fill = Period)) +
  geom_col(position = "dodge") +
  theme_test() +
  labs(title = "Average Games Missed by Injury Region (2021-2023 vs. 2024)",
       x = "Injury Region",
       y = "Average Games Missed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "top")
```



Difference in Contact Injuries from 2021-2023. Vs 2024

```
InjuryData %>%
  filter(!is.na(Injury_Type) & !is.na(Injury_InitialSeverity) & !is.na(GamesMissed)) %>%
  mutate(
    Period = ifelse(Season >= 2021 & Season <= 2023, "2021-2023", "2024"),
    GamesMissed = as.numeric(GamesMissed),  # Ensure GamesMissed is numeric
    Injury_InitialSeverity = case_when(
      grepl("mild|minor", Injury_InitialSeverity, ignore.case = TRUE) ~ "Mildly Severe",
      grepl("severe", Injury_InitialSeverity, ignore.case = TRUE) ~ "Severe",
      grepl("not severe", Injury_InitialSeverity, ignore.case = TRUE) ~ "Not Severe",
      grepl("extreme|extremely", Injury_InitialSeverity, ignore.case = TRUE) ~ "Extremely Severe",
      TRUE ~ "Other"  # Catch all for unexpected cases
    )
  ) %>%
  group_by(Period, Injury_InitialSeverity, Injury_Type) %>%
  summarise(avg_GamesMissed = mean(GamesMissed, na.rm = TRUE), .groups = 'drop') %>%
  ggplot(aes(x = Injury_Type, y = avg_GamesMissed, fill = Injury_InitialSeverity)) +
  geom_col(position = "dodge") +  # Dodge bars to separate injury severity levels
  facet_wrap(~ Period) +  # Facet by period (2021-2023 vs 2024)
  theme_test() +
  labs(title = "Average Games Missed by Injury Severity and Type of Injury",
       x = "Injury Type",
       y = "Average Games Missed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "top")  # Rotate labels fo
```



Is there a significant difference in Mean Contact Injuries from 2021-2023. Vs 2024

```
str(InjuryData$GamesMissed)
```

```
##  chr [1:300] "0" "0" "0" "0" "0" "0" "0" "0" "10" "11" "NULL" "2" "8" "11" ...
```

```
summary(InjuryData$GamesMissed)
```

```
##    Length     Class      Mode
##       300 character character
```

```
InjuryData$GamesMissed <- suppressWarnings(as.numeric(as.character(InjuryData$GamesMissed)))


InjuryData_clean <- InjuryData %>%
  drop_na(GamesMissed)


summary(InjuryData_clean$GamesMissed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   2.186   2.000  41.000
```

```
games_missed_2021_2023 <- InjuryData_clean %>%
  filter(Season %in% c(2021, 2022, 2023)) %>%
  pull(GamesMissed)
```

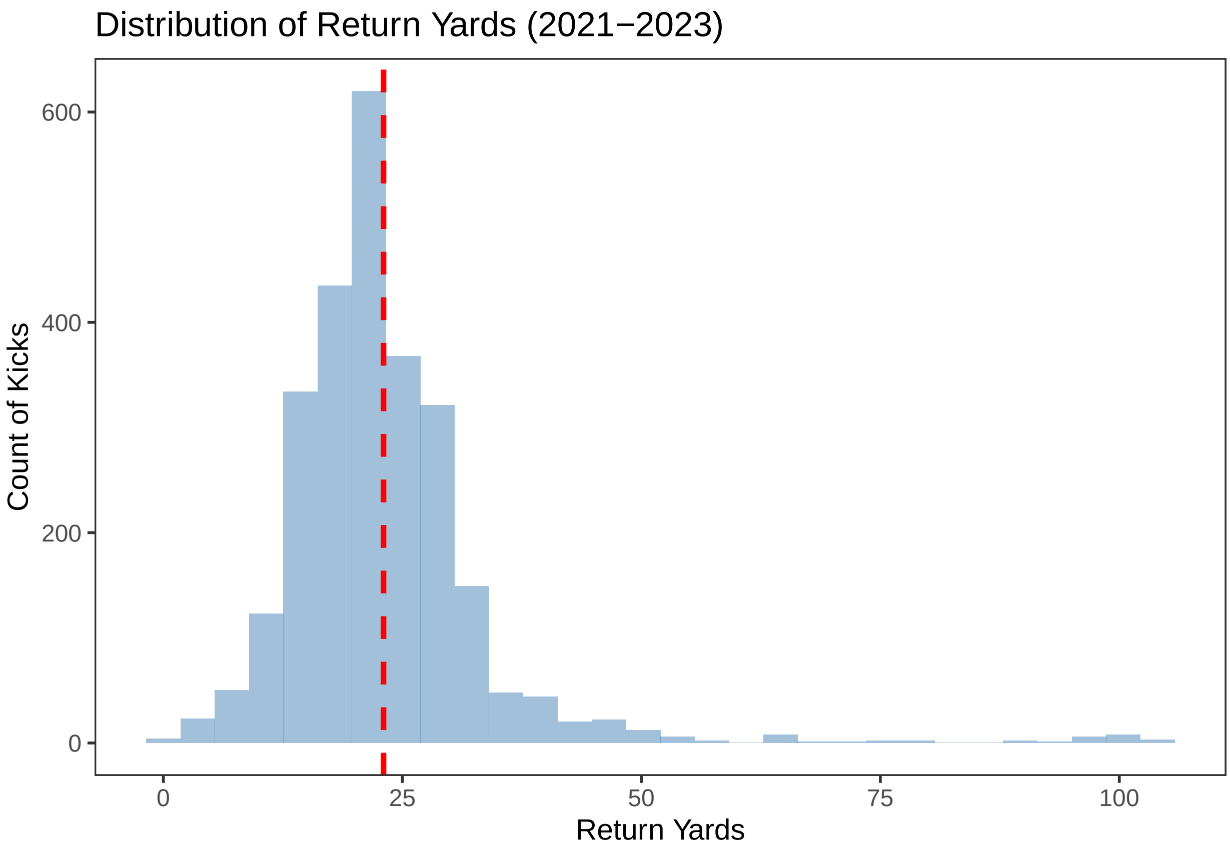
```r
games_missed_2024 <- InjuryData_clean %>%
  filter(Season == 2024) %>%
  pull(GamesMissed)

t_test_result <- t.test(
  games_missed_2021_2023,
  games_missed_2024,
  alternative = "two.sided"
)

print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  games_missed_2021_2023 and games_missed_2024
## t = 2.8524, df = 219.66, p-value = 0.004753
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3836815 2.0991535
## sample estimates:
## mean of x mean of y
##  2.380952  1.139535
```

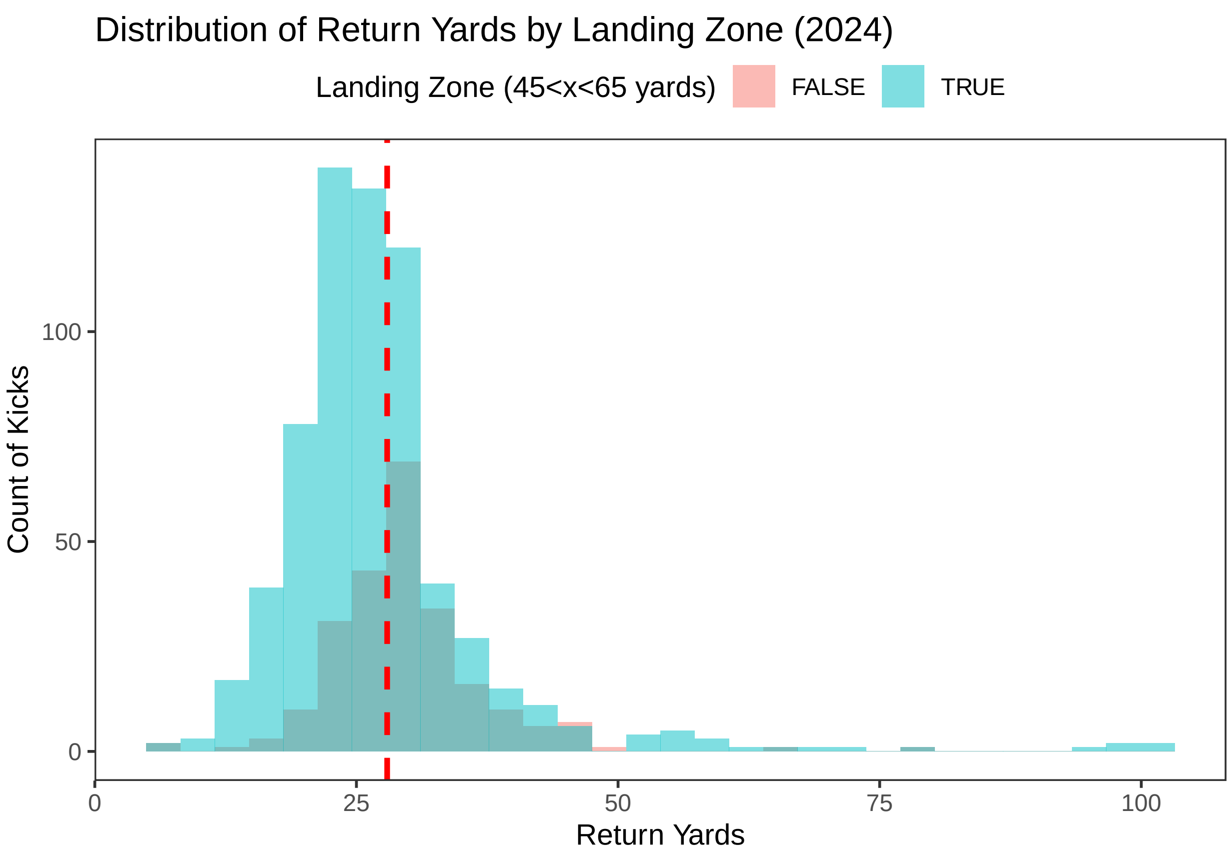
Since the p-value is less than 0.05, we can reject the null hypothesis that the mean number of contact injuries from 2021-2023 and 2024 are equal, and therefore the number of contact injuries in 2024 is significantly different from the mean number of contact injuries in the years 2021-2023.

Distribution of Return Yards Pre Dynamic Kickoff vs. Post Dynamic Kickoff

```r
# 2021-2023 data
kickingdata_2021_2023 <- kickingdata %>%
  filter((Season == 2021 | Season == 2022 | Season == 2023) & ReturnYards > 0 & Quarter <= 4)

# Calculate the mean for 2021-2023
mean_2021_2023 <- mean(kickingdata_2021_2023$ReturnYards, na.rm = TRUE)

# Plot for 2021-2023 with mean vertical line
ggplot(kickingdata_2021_2023, aes(x = ReturnYards)) +
  geom_histogram(bins = 30, alpha = 0.5, position = "identity", fill = "steelblue") +
  geom_vline(xintercept = mean_2021_2023, color = "red", linetype = "dashed", size = 1) +
  labs(x = "Return Yards",
       y = "Count of Kicks",
       title = "Distribution of Return Yards (2021-2023)") +
  theme_test()
```

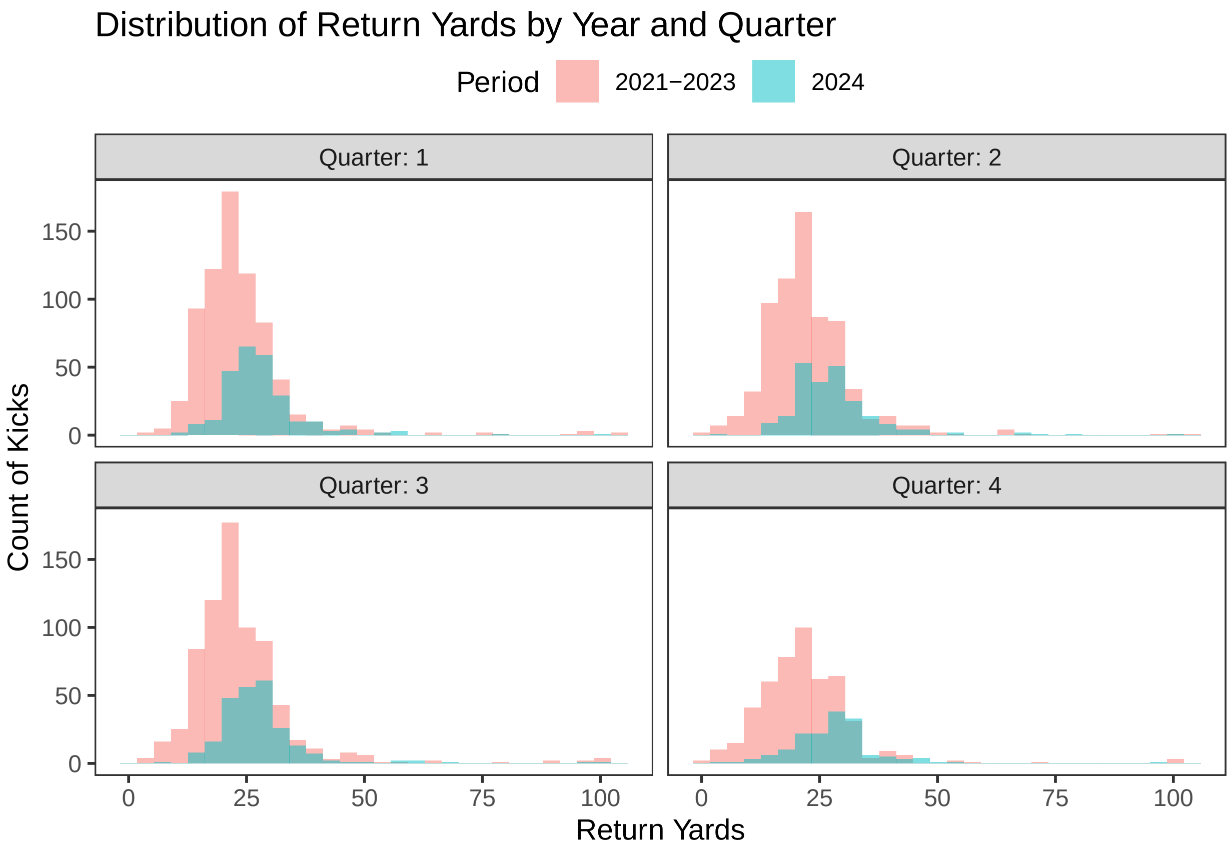



```r
# 2024 data
kickingdata_2024 <- kickingdata %>%
  filter(Season == 2024 & ReturnYards > 0 & Quarter <= 4) %>%
  mutate(landing_zone = between(KickYards, 45, 65),
         pointdiff = KickingTeamScore - ReturningTeamScore)

# Calculate the mean for 2024
mean_2024 <- mean(kickingdata_2024$ReturnYards, na.rm = TRUE)

# Plot for 2024 with mean vertical line
ggplot(kickingdata_2024, aes(x = ReturnYards, fill = factor(landing_zone))) +
  geom_histogram(bins = 30, alpha = 0.5, position = "identity") +
  geom_vline(xintercept = mean_2024, color = "red", linetype = "dashed", size = 1) +
  labs(x = "Return Yards",
       y = "Count of Kicks",
       fill = "Landing Zone (45<x<65 yards)",
       title = "Distribution of Return Yards by Landing Zone (2024)") +
  theme_test() +
  theme(legend.position = "top")
```



Distribution of Return Yards Pre Dynamic Kickoff vs. Post Dynamic Kickoff and Z-test

```
kickingdata2_2021_2023 <- kickingdata %>%
  filter((Season == 2021 | Season == 2022 | Season == 2023) & ReturnYards > 0 & Quarter <= 4) %>%
  mutate(Period = "2021-2023")


kickingdata2_2024 <- kickingdata %>%
  filter(Season == 2024 & ReturnYards > 0 & Quarter <= 4) %>%
  mutate(landing_zone = between(KickYards, 45, 65),
         pointdiff = KickingTeamScore - ReturningTeamScore,
         Period = "2024")


combined_kickingdata <- bind_rows(kickingdata2_2021_2023, kickingdata2_2024)


ggplot(combined_kickingdata, aes(x = ReturnYards, fill = Period)) +
  geom_histogram(bins = 30, alpha = 0.5, position = "identity") +
  facet_wrap(~ Quarter, labeller = label_both) +
  labs(x = "Return Yards",
       y = "Count of Kicks",
       title = "Distribution of Return Yards by Year and Quarter") +
  theme_test() +
  theme(legend.position = "top")
```




```
mean_2024 <- mean(kickingdata_2024$ReturnYards, na.rm = TRUE)
mean_2021_2023 <- mean(kickingdata_2021_2023$ReturnYards, na.rm = TRUE)

sd_2024 <- sd(kickingdata_2024$ReturnYards, na.rm = TRUE)
sd_2021_2023 <- sd(kickingdata_2021_2023$ReturnYards, na.rm = TRUE)


n_2024 <- length(kickingdata_2024$ReturnYards)
n_2021_2023 <- length(kickingdata_2021_2023$ReturnYards)


z_score <- (mean_2024 - mean_2021_2023) / sqrt((sd_2024^2 / n_2024) + (sd_2021_2023^2 / n_2021_2023))


p_value <- 2 * (1 - pnorm(abs(z_score)))


results_table <- data.frame(
  "Statistic" = c("Mean (2024)", "Mean (2021-2023)", "Standard Deviation (2024)", "Standard Deviation (
                  "Z-score", "P-value"),
  "Value" = c(mean_2024, mean_2021_2023, sd_2024, sd_2021_2023, z_score, p_value)
)

print(results_table)
```

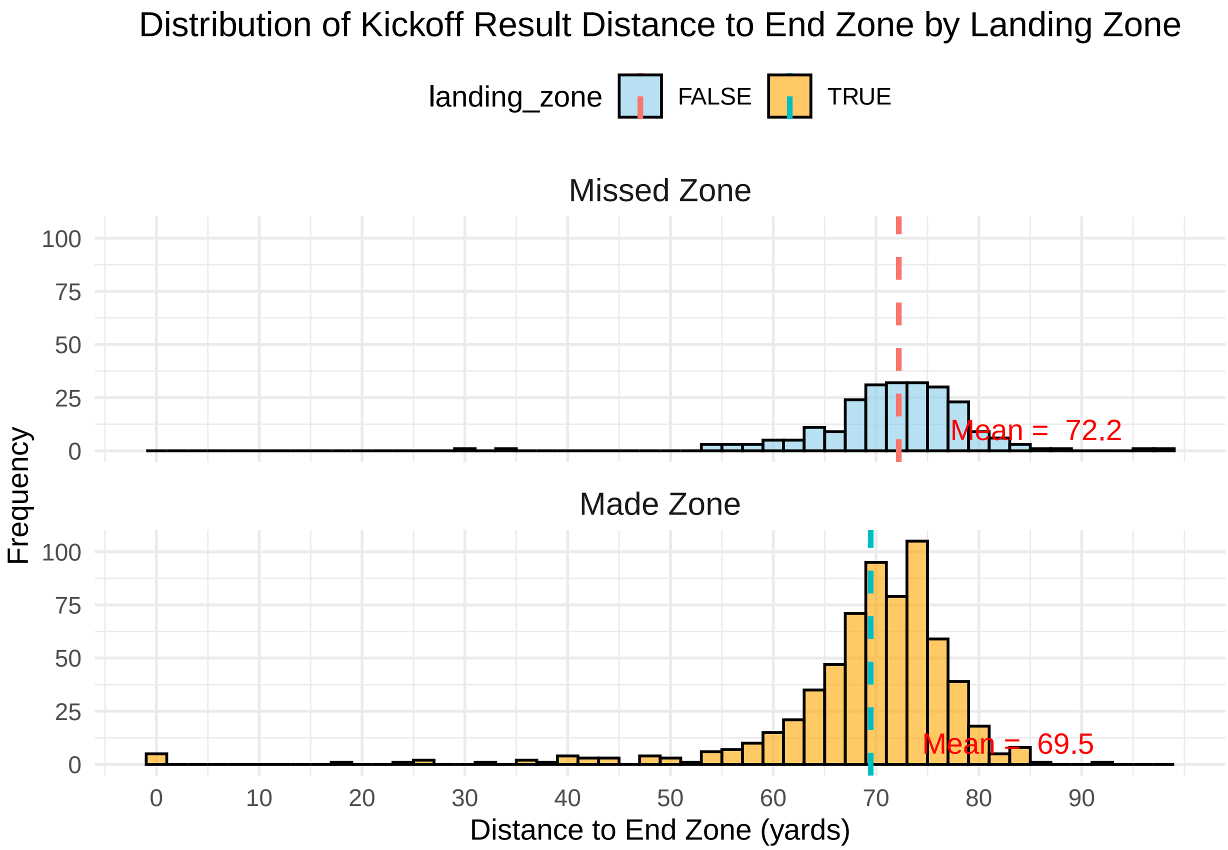
```
##                         Statistic     Value
## 1                     Mean (2024) 27.934685
```

```
## 2              Mean (2021-2023) 23.027916
## 3      Standard Deviation (2024)  9.819579
## 4 Standard Deviation (2021-2023) 10.777803
## 5                        Z-score 12.544121
## 6                        P-value  0.000000
```

Since the p-value is 0, we can reject the null hypothesis that the two sample means are equal, and therefore the distribution of return yards is significantly different from the distribution of return yards in the years 2021-2023.

Does kicking it to the landing zone get your opponent better field position?

```
kickingdata_2024 <- kickingdata %>%
  filter(Season == 2024 & ReturnYards > 0 & Quarter <= 4) %>%
  mutate(landing_zone = between(KickYards, 45,65),
         pointdiff = KickingTeamScore - ReturningTeamScore)

kickingdata_2024 %>%
  ggplot(aes(x = KickoffResultDistToEZ)) +
    geom_histogram(binwidth = 2, aes(fill = landing_zone), color = "black", alpha = 0.6) +
    scale_fill_manual(values = c("skyblue", "orange")) +
    facet_wrap(~ landing_zone, labeller = labeller(landing_zone = c('TRUE' = 'Made Zone', 'FALSE' = 'Mis
    labs(title = "Distribution of Kickoff Result Distance to End Zone by Landing Zone",
         x = "Distance to End Zone (yards)",
         y = "Frequency") +
    scale_x_continuous(breaks = seq(0, max(kickingdata_2024$KickoffResultDistToEZ), by = 10)) +
    geom_vline(data = kickingdata_2024 %>%
                 group_by(landing_zone) %>%
                 summarise(mean_value = mean(KickoffResultDistToEZ, na.rm = TRUE)),
               aes(xintercept = mean_value, color = landing_zone),
               linetype = "dashed", size = 1) +
    geom_text(data = kickingdata_2024 %>%
                group_by(landing_zone) %>%
                summarise(mean_value = mean(KickoffResultDistToEZ, na.rm = TRUE)),
              aes(x = mean_value + 5,
                  y = 10,
                  label = paste("Mean = ", round(mean_value, 1))),
              color = "red", size = 4, angle = 0, hjust = 0) +
    theme_minimal() +
    theme(legend.position = "top", strip.text = element_text(size = 12), plot.title = element_text(hjus
```




```r
# Calculate the mean value for 2024 data
mean_2024 <- mean(kickingdata2_2024$KickoffResultDistToEZ, na.rm = TRUE)

# Calculate the mean value for 2021-2023 data
mean_2021_2023 <- mean(kickingdata2_2021_2023$KickoffResultDistToEZ, na.rm = TRUE)

# t-test between the two means
t_test_result2 <- t.test(kickingdata2_2024$KickoffResultDistToEZ, kickingdata2_2021_2023$KickoffResultDi

# View the result of the t-test
t_test_result2
```

```
##
##  Welch Two Sample t-test
##
## data:  kickingdata2_2024$KickoffResultDistToEZ and kickingdata2_2021_2023$KickoffResultDistToEZ
## t = -10.364, df = 1616.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.726028 -3.221829
## sample estimates:
## mean of x mean of y
##  70.20045  74.17438
```

Similarly to the return yards variable, we can reject the null hypothesis that the two sample means are equal, and therefore the distribution of distance to endzone after the kickoff in 2024 is significantly different from the distribution of distance to endzone after the kickoff in the years 2021-2023. From this conclusion, kicking into the landing zone gives the kicking team an advantage.