NBA Injury Patterns: A Time Series Analysis

An Adhikari, Charlotte Zhuang, Hedavam Solano

2025-01-02

Table of contents

Introduction

Research Motivation

The NBA has seen significant changes in playing style, medical practices, and season structure over the past two decades. Understanding injury patterns is crucial for team management, player health, and league scheduling. This study analyzes injury data from 2001-2023 to identify patterns that could help teams and medical staff better prepare for and possibly prevent injuries.

0.1 Research Questions

Can seasonal trends in NBA injuries be identified and predicted? Do injury rates vary significantly by teams after accounting for yearly trends?

0.2 Data Overview

The original dataset, acquired from Kaggle, comprises NBA player injury records from 1951 to 2023, tracking both injuries(player relinquished) and returns to play(player acquired). For consistency in reporting standards and modern medical practices, we focused on data from 2001 onwards. After filtering for post-2001 period, our dataset contains 33178 injury observations with 5 key variables. Key variables include player name, injury dates, player status, team affiliation and injury description.

1 Data Processing

> ⚠ Warning
>
> The code snippets shown in this section are simplified for illustration. Complete code with detailed data processing steps can be found in the Appendix.

1.1 Initial Data Loading and Cleaning

```
NBA <- read_csv("~/Downloads/NBA Player Injury Stats(1951 - 2023).csv")

NBA_filtered <- NBA %>%
  filter(Year > 2000) %>%
  mutate(is_regular_season = Date < playoff_start | is.na(playoff_start))
```

The data was filtered to include only post-2001 injuries and regular season games.

About this file

All recorded NBA Player injuries from 1951 to 2023

# Index	Date Date of occurence	Team Team Name	Acquired This is the player coming back from an injury	Relinquished This is the player being placed on the IL	Notes Any notes about the reasoning for being placed on IL
0 37.7k	1951-12-24 2023-04-15	Spurs 4% Celtics 4% Other (34486) 92%	[null] 53% Andre Iguodala 0% Other (17574) 47%	[null] 47% (William) Tony Par... 0% Other (19992) 53%	activated from IL 46% placed on IL 17% Other (13902) 37%
0	1951-12-25	Bullets		Don Barksdale	placed on IL
1	1952-12-26	Knicks		Max Zaslofsky	placed on IL with torn side muscle
2	1956-12-29	Knicks		Jim Baechtold	placed on inactive list
3	1959-01-16	Lakers		Elgin Baylor	player refused to play after being denied a room in team's hotel
4	1961-11-26	Lakers		Elgin Baylor	player reported for military duty
5	1962-03-24	Lakers	Elgin Baylor		player given 2-day pass from military duty
6	1962-03-31	Lakers	Elgin Baylor		player given weekend pass from military duty

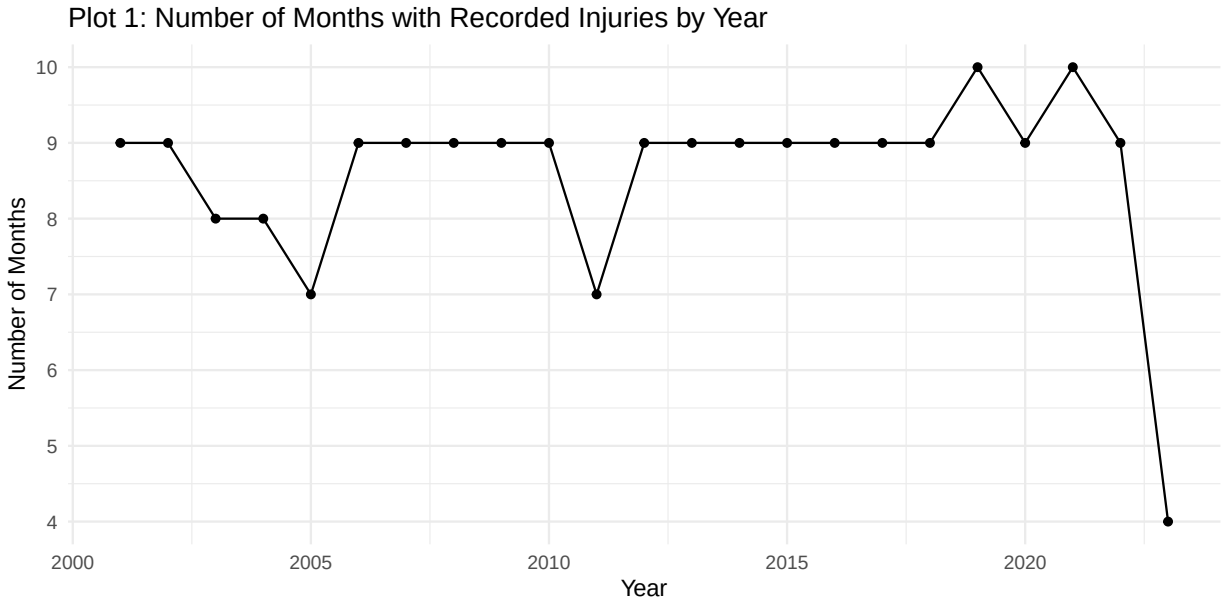
Figure 1: Kaggle Overview

1.2 Missing Month Analysis

Plot 1 shows the varying number of months with recorded injuries across years. The inconsistency in monthly records necessitated our zero imputation approach to create a complete time series.

1.3 Time Series Creation

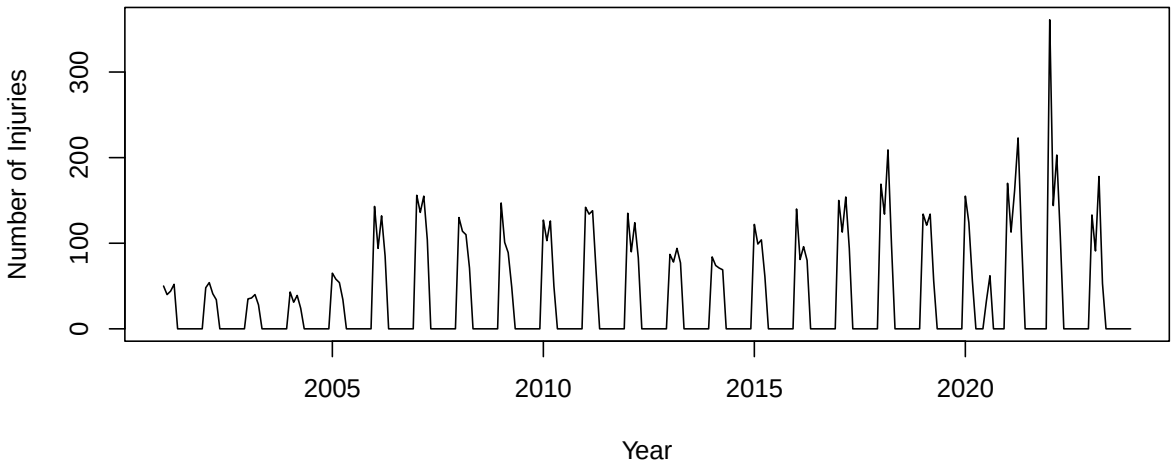
```
NBA_imputed <- NBA_counts %>%
  complete(
    Year = unique(NBA_counts$Year),
    Month = as.character(1:12),
    fill = list(Total_Entries = 0)
  )

NBA_timeseries <- ts(NBA_imputed$Total_Entries,
                     start = c(2001, 1),
```




```
                    frequency = 12)
```

For months with no recorded injuries, zeros were imputed. A missing month in our data indicates either no injuries or no games played. This zero imputation maintains a consistent monthly frequency (12) in our time series, enabling proper seasonal analysis. The time-series object is represented in plot 1.3.

Plot 1.3: NBA Injuries Over Time (Regular Season Only)

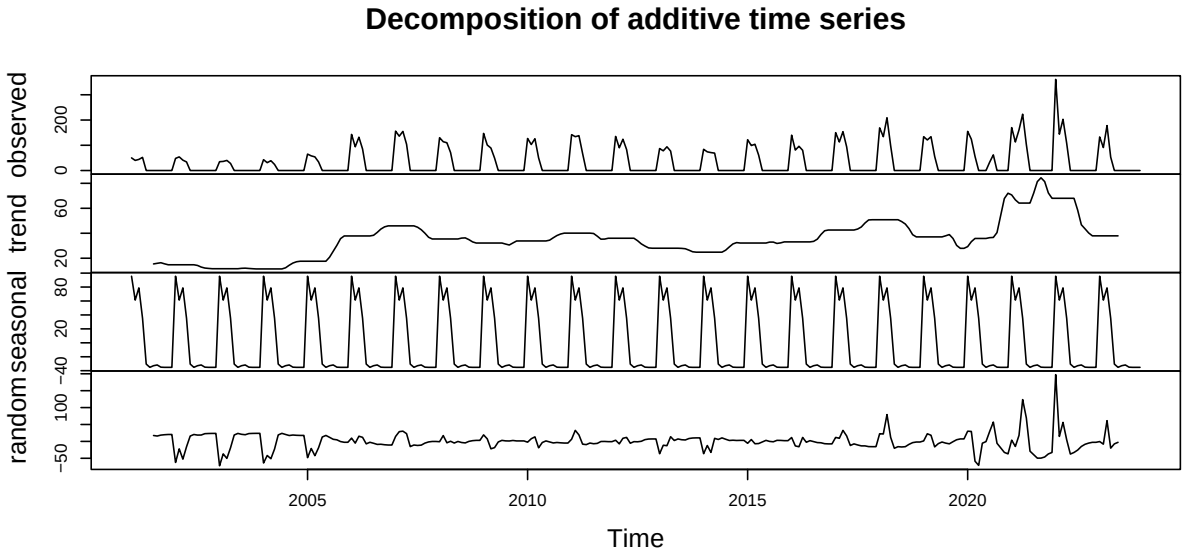


2 Exploratory Analysis

The decomposition of our NBA injury time series reveals three distinct components:

1. **Observed Pattern (top panel):** The observed pattern, shown in the top panel, presents the raw number of injuries over time in the NBA. This pattern demonstrates clear seasonal spikes that occur consistently each year throughout the dataset. We can observe a notable increase in both variability and the height of these peaks after 2020.
2. **Trend Component (second panel):** The trend component, displayed in the second panel, reveals a gradual increase in the baseline number of injuries from 2001 to 2023. Between 2007 and 2017, the trend remained relatively stable, with monthly injury counts averaging between 30 to 40 cases. However, after 2020, there was a sharp increase, with monthly injury counts reaching unprecedented peaks of 60 to 70 cases. This can likely be attributed to the condensed schedule implemented during the COVID-19 pandemic.
3. **Seasonal Component (third panel):** The seasonal component reveals a distinctive "double-peak" pattern within each NBA season. The first and larger peak typically occurs in the early-to-mid season (around December-January), followed by a noticeable dip that coincides with the All-Star break in February. After this break, there's a second, smaller peak as teams make their final push toward the playoffs. This pattern may reflect the season's natural intensity flow: teams start the season at full intensity, take a brief respite during All-Star weekend, followed by the final stretch of regular season games where teams might manage player minutes more carefully as they prepare for potential playoff runs.

This decomposition suggests that NBA injuries follow a predictable seasonal rhythm but have been increasing over time, with particularly notable changes in recent years. The presence of



clear seasonal patterns and an upward trend will be important considerations for our forecasting approach. Before proceeding with time series modeling, we will assess stationarity - a requirement for ARMA models. A stationary time series maintains consistent statistical properties (mean and variance) over time.

2.1 Correlation Analysis

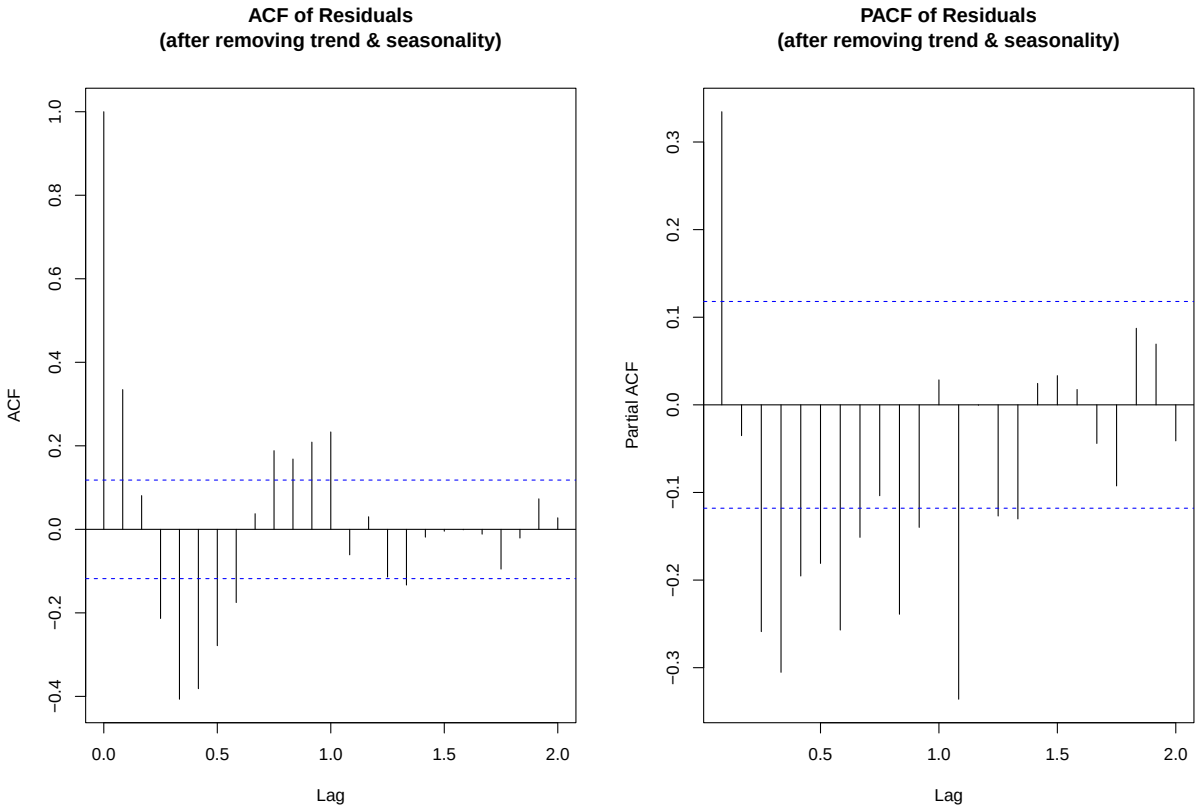
To choose the right time series model for our NBA injury data, we need to understand how injuries in one month might be related to injuries in previous months. We use two special tools for this: ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots.

The ACF plot shows how strongly injury counts in a given month are related to injury counts in previous months. The height of each bar in the ACF plot shows the strength of these relationships at different time gaps (lags).

The PACF plot shows direct relationships between months, filtering out indirect effects. For example, if January affects February, and February affects March, the PACF helps us see if January has any additional direct influence on March beyond its indirect effect through February.

We're looking at these plots for the residuals (what's left after removing trend and seasonal patterns) rather than the original data for an important reason. In our case, we already know injuries increase over time (trend) and spike during certain parts of the season (seasonality). By removing these known patterns first, we can better see any remaining relationships that our model needs to capture.

In the ACF plot (left), there's a significant positive spike at the very beginning (lag 0), which is expected as it represents the correlation of the series with itself. After this, we see a pattern of



few negative correlations in early lags, suggesting that high injury counts tend to be followed by lower counts in the next few months, and vice versa.

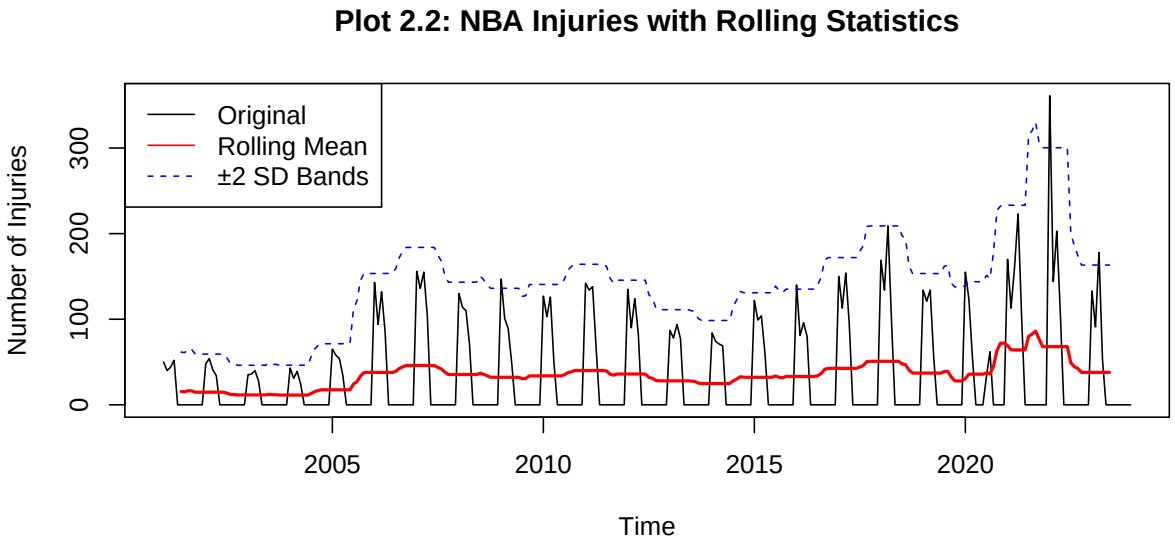
In the PACF plot(right) there are multiple significant negative spikes that extend beyond the blue dotted lines (significance boundaries), particularly in the first few lags up to around lag 1.0. These significant negative correlations suggest that after removing trend and seasonality, there's still a strong pattern of serial correlation, even when accounting for other intervening months.

For modeling purposes, these significant PACF values suggest we might need a higher-order AR component in our model than initially thought, as the correlations extend beyond just the immediate next month.

2.2 Visual Inspection of Stationarity

Key Observations from Rolling Statistics

In Plot 3, the rolling average (red line) calculates the mean number of injuries over a 12-month window, which helps us see the underlying trend by smoothing out short-term fluctuations. The blue dashed lines show two standard deviations above and below this average, helping us visualize how much the data typically varies from the mean.



The rolling mean (red line) is relatively stable in the short term, showing only gradual changes until 2020, after which there's a sharp rise. Most of the original data points (black line) fall within the two standard deviation bands (blue dashed lines), suggesting reasonable short-term stability.

The plot reveals that our series is largely stable - the rolling mean maintains a consistent level despite seasonal fluctuations, and most observations fall within the standard deviation bands.

3 Model Development

3.1 Approach Selection

Given our exploratory analysis, we considered both ARMA (Autoregressive Moving Average) and ARIMA (Autoregressive Integrated Moving Average) models for our analysis. ARMA models are used for stationary time series, where statistical properties like mean and variance remain stable over time. ARIMA models extend ARMA by including differencing to handle non-stationary data.

Based on our analysis, we chose to proceed with an ARIMA model for several reasons:

> While our rolling statistics showed some stability, the sharp rise after 2020 and the overall increasing trend suggest our series might not be truly stationary
>
> The PACF plot revealed significant correlations at multiple lags, indicating complex patterns that need to be modeled.
>
> ARIMA is more flexible - if our data turns out to be stationary after all, the ARIMA model will simply use zero differencing (d=0), effectively reducing to an ARMA model. This way, we can let the data guide us to the most appropriate model specification rather than making a strict assumption about stationarity upfront.

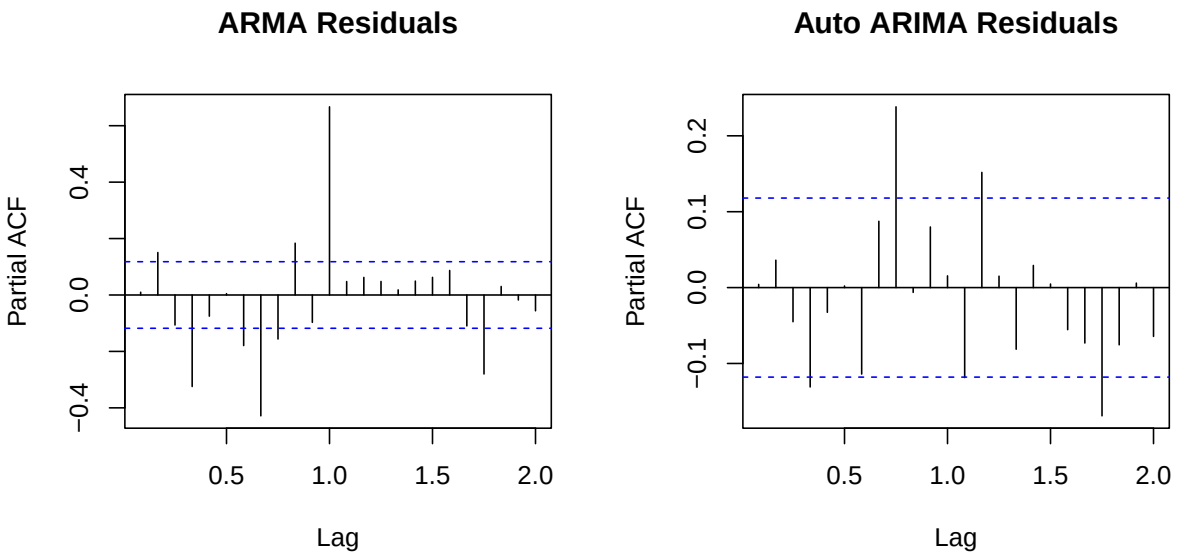
3.2 Model Comparison

To justify our model choice for our NBA injury time series, we compared a simple ARMA(1,1) model with an automatically selected ARIMA model. We'll evaluate these models by examining their residual patterns through PACF plots and assessing how well they capture the observed injury patterns over time. These comparisons will help us understand whether the additional complexity of the ARIMA model is justified by improved model performance.

```
       Model      AIC      BIC
1  ARMA(1,1) 2914.653 2929.135
2 Auto ARIMA 2454.960 2472.840
```

The comparison reveals that both AIC (2452.391 vs 2914.653) and BIC (2480.998 vs 2929.135) are much lower for the Auto ARIMA model - these scores measure how well a model fits the data while penalizing unnecessary complexity, where lower scores indicate better models.

Looking at the PACF plots, we can see that the Auto ARIMA model's residuals show fewer significant spikes beyond the confidence bounds, suggesting it has captured more of the underlying patterns in the data. This is further supported by Plot 3.2, where the Auto ARIMA model's fitted values (red line) track the observed values (black line) more closely than the ARMA model's predictions (blue line), particularly during periods of high variability.



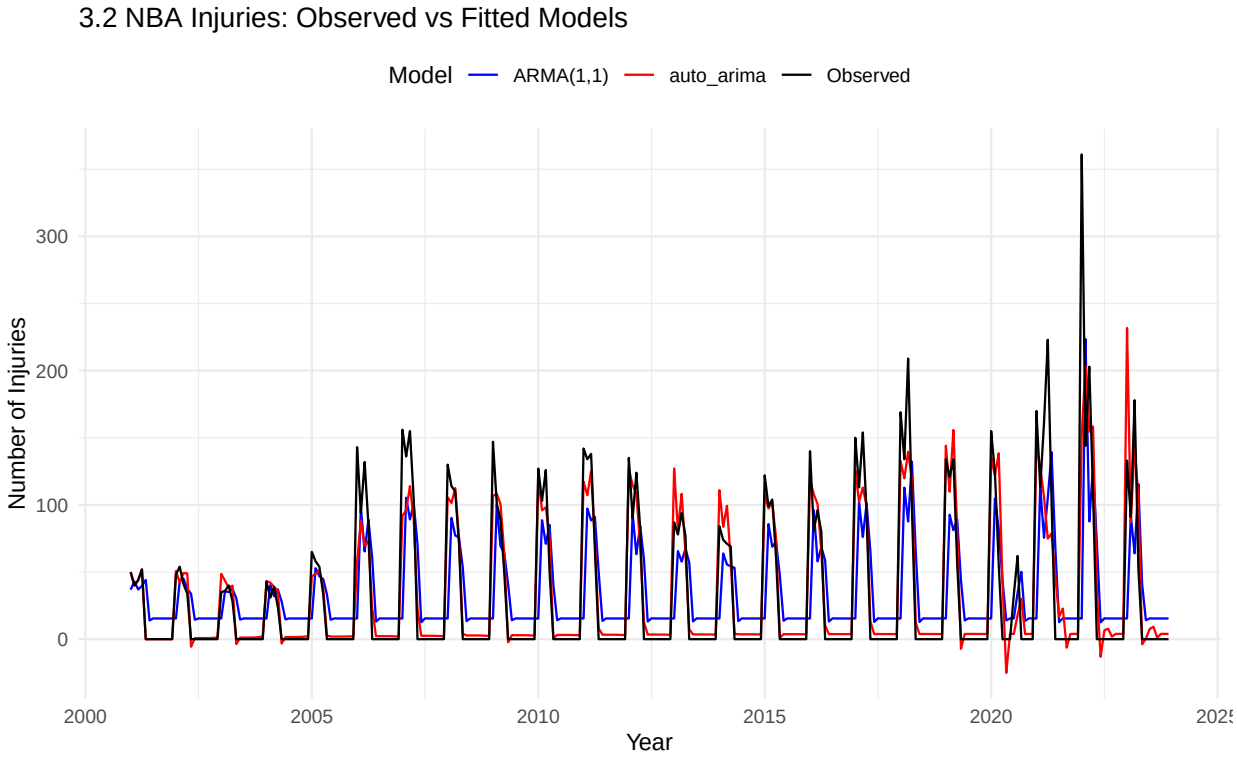
Based on these results, we selected the Auto ARIMA model (ARIMA(1,0,0)(0,1,2)[12]) for our final analysis. This model specification shows that for the non-seasonal part, we need one autoregressive term (1), no differencing (0), and no moving average terms (0). The seasonal component (0,1,2)[12] indicates we need one seasonal difference to handle yearly patterns, and two seasonal moving average terms at lag 12.

! Model Limitations

While the Auto ARIMA model shows better fit than the simple ARMA model, the PACF plots of residuals still show several significant correlations. This suggests our model, despite incorporating seasonal components and showing good visual fit, doesn't fully capture all temporal dependencies in the NBA injury data. This likely stems from not incorporating external variables like team factors and scheduling changes which could account for some of the remaining autocorrelation. See Section 7.1 for additional limitations.

3.3 ARIMA with an External Regressor

From https://hoopshype.com/salaries/players/, we manually gathered the number of players for each NBA season in our time series. For a given year, we made the assumption that each month there was the same number of players in the league, due to the lack of more granular data. For off-season months, those with 0 injuries, we set the number of players in the league to 0. The inclusion of this external regressor did improve the autocorrelation in the residual series as shown in Plot 3.2 as there are less significant spikes and they pop up at later lags. Perhaps more external regressors could further remedy the issue of independence.

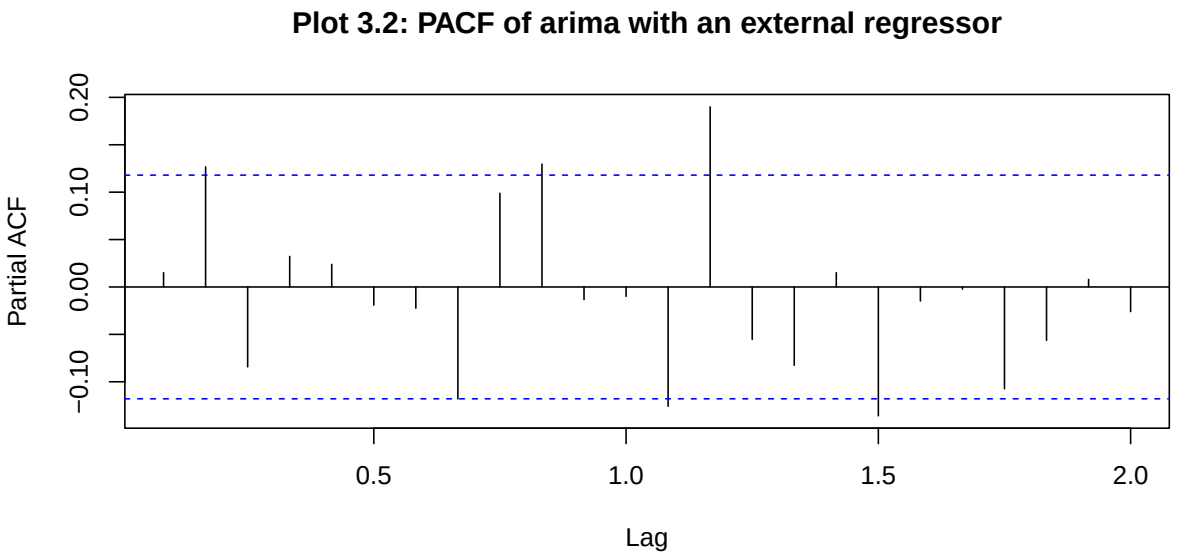


4 Forecasting

Despite the limitations noted above, the Auto ARIMA model remains as it effectively captures the main seasonal patterns and trends in the data, and demonstrates superior predictive ability compared to simpler alternatives.

Looking at the forecasted values (green line) for 2024-2028 in plot 4, our model predicts that NBA injuries will continue to follow the historical seasonal patterns. The model preserves the characteristic "double-peak" pattern we observed in the historical data, with injury rates typically spiking mid-season and near the end of regular season. However, there's a notable limitation in our forecast: it predicts injuries during the off-season months (around July-September), when no regular season games are played. The model's practical application would require adjusting these predictions to account for the actual NBA calendar, particularly by removing or adjusting the off-season predictions that shouldn't show significant injury counts.

The grey shaded area represents the 95% prediction interval, which widens over time, reflecting increasing uncertainty in our long-term predictions.



5 Inference: Team-Specific Injury Trends

Motivation

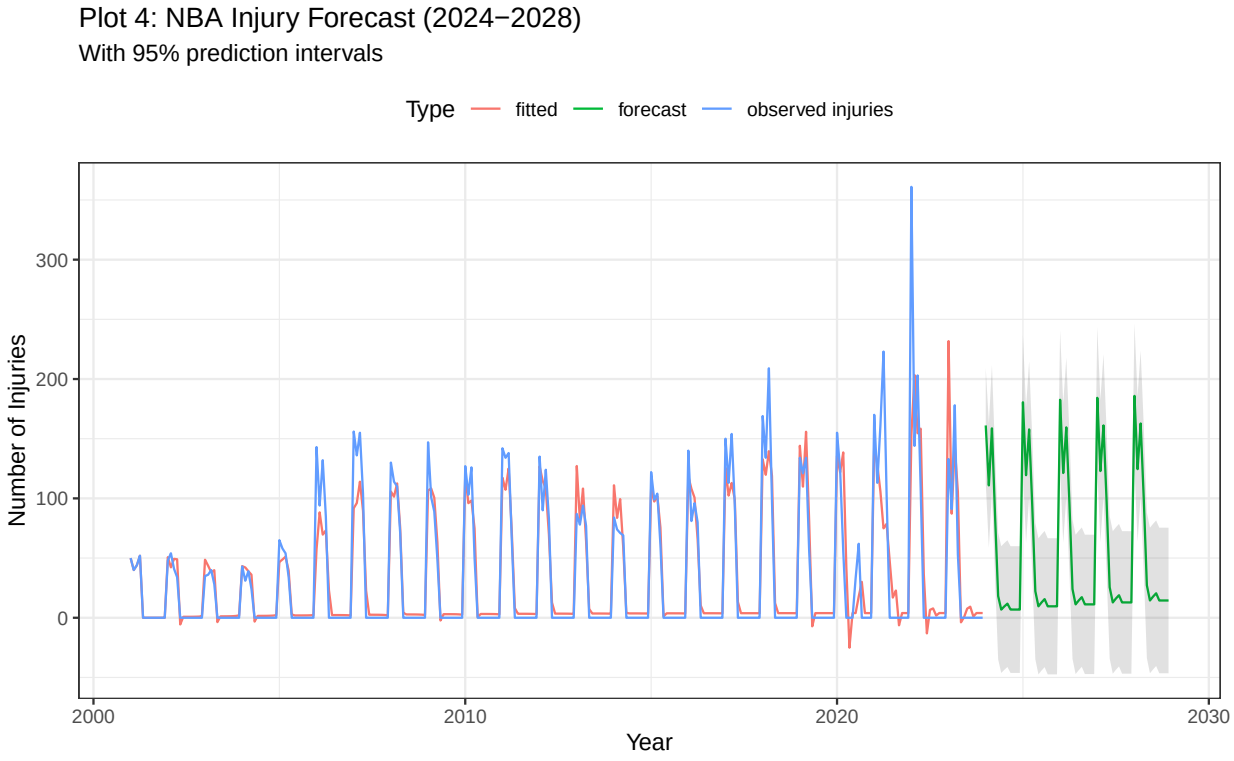
While our time series analysis revealed overall injury patterns, we were also interested in understanding whether some teams were significantly better or worse at managing injuries after accounting for yearly trends.

5.1 Modeling Approaches

5.1.1 Ordinary Least Squares (OLS)

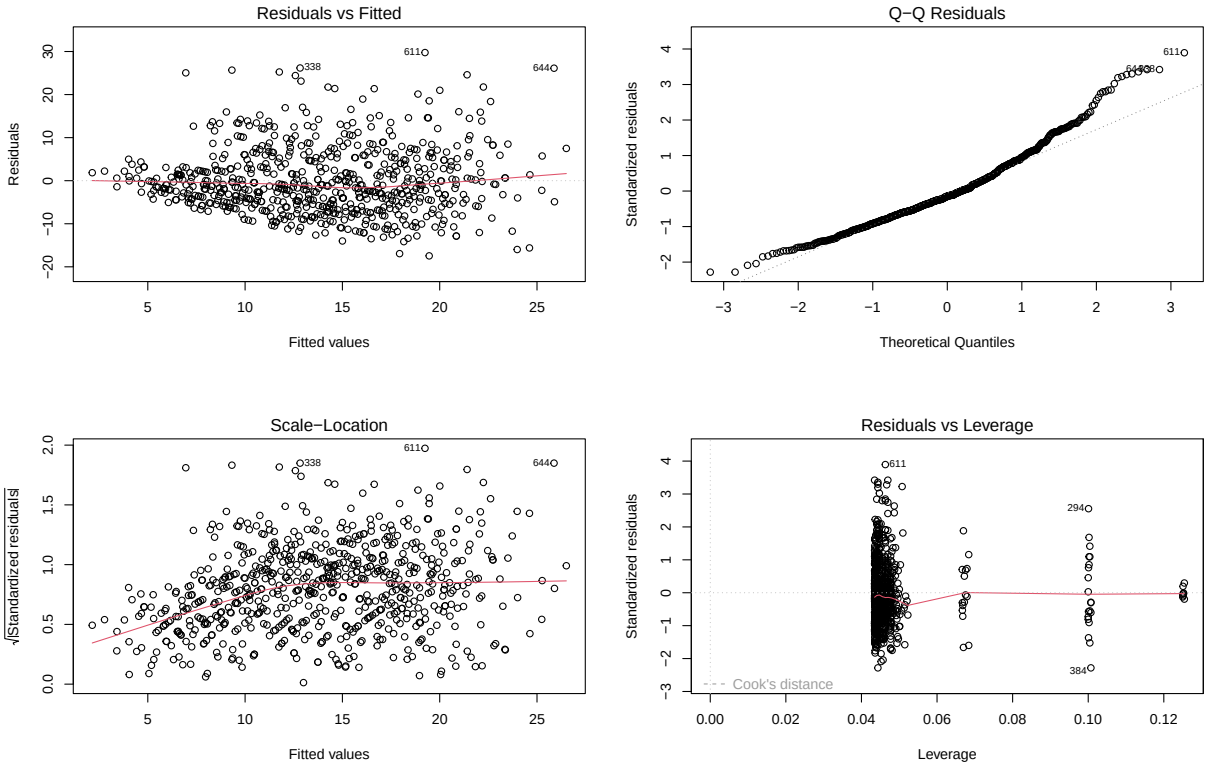
We fit an OLS model first that uses the total number of injuries by year for each team as the response variable and then year and team as explanatory variables. We excluded interaction term because believe overall trends in injury management will differ significantly by team. In theory, the NBA's medical policies would be applied uniformly even if some teams are overall better/worse or just unlucky with injuries. Additionally, excluding an interaction term allows for better interpretability.

Shifting our focus to our assessment of the assumptions of regression, we see that linearity is satisified as the LOESS line on the Residuals vs. Fitted model is fairly straight suggesting a lack of patterns in the residuals. However, constant variance might be violated as the errors for smaller fitted values are pretty small compared to the error associated with larger fitted values. The slanted start to the LOESS line on the Scale-Location plot highlights this issue. Normality is also questionable as the points towards the end of our reference line are slightly curved and pretty far



way from it. Lastly, independence is violated as we see an abundance on significant spikes on the ACF/PACF plots for the residuals.

Given these violations of constant variance and independence assumptions, we turned to Generalized Least Squares (GLS), which can account for correlated error structures. To determine the appropriate error structure, we first used auto.arima() to identify the optimal ARMA(p,q) specification.

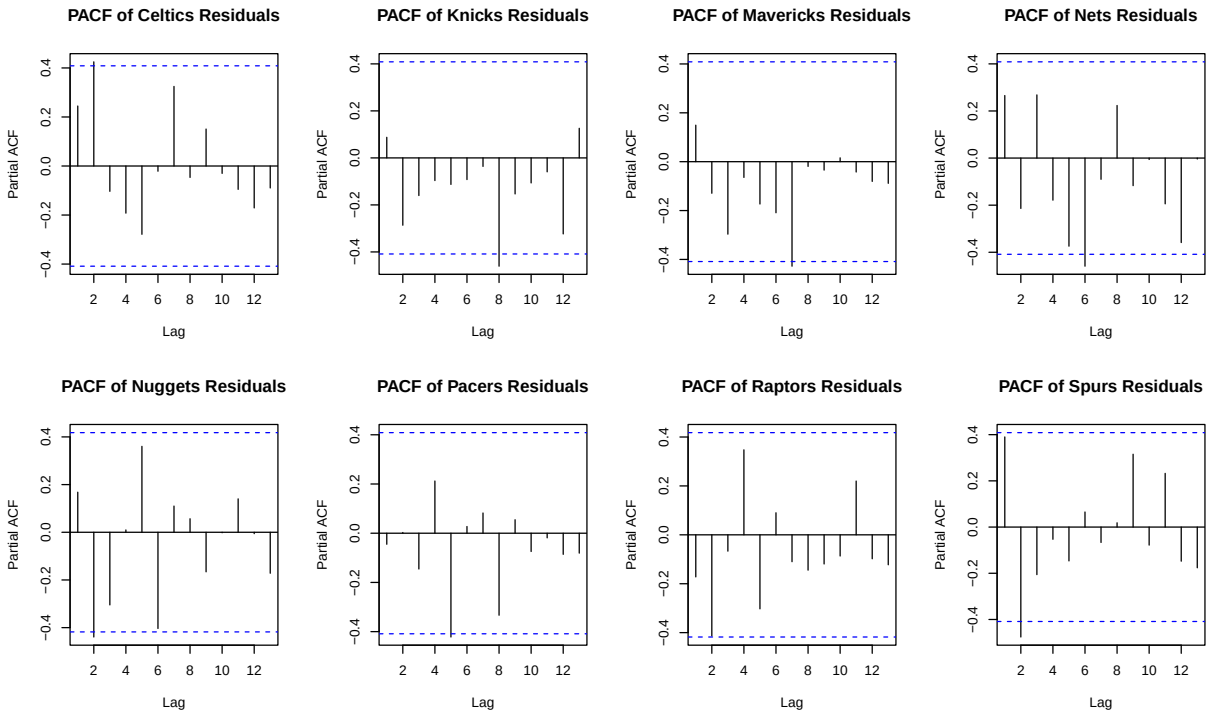


5.1.2 ARIMA

```
xreg <- cbind(NBA_imputed$Year, factor(NBA_imputed$Team))
arima_fit <- auto.arima(NBA_imputed$Total_Entries, xreg = xreg)
cat("ARIMA", paste0(arima_fit$arma[c(1,6,2)], collapse=","))
```

```
ARIMA 2,0,3
```

The auto.arima function identified an ARIMA(2,0,3) model as the best fit for our data with external regressors. This specification indicates that the model uses two auto-regressive terms (p=2), no differencing (d=0), and three moving average terms (q=3). The PACF plot of the residuals (Plot 5.1.4) shows no significant spikes outside the confidence bounds, suggesting that this model adequately captures the temporal dependencies in our data. This lack of significant autocorrelation in the residuals indicates that the ARIMA(2,0,3) specification is appropriate for modeling the error structure in our subsequent GLS analysis.



5.1.3 Generalized Least Squares

Using the ARMA(2,3) process obtained by ARIMA, we fit a GLS model that considers autocorrelation to be global over time, maintaining our assumption that the injury trends should behave similarly for all teams.

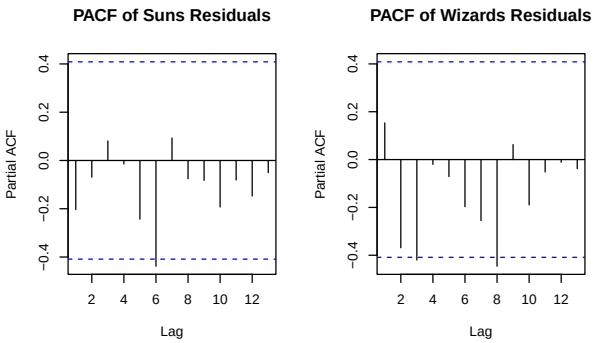
```
Denom. DF: 646
            numDF   F-value p-value
(Intercept)     1 286.60800  <.0001
Year            1  25.89896  <.0001
Team           31   3.09787  <.0001
```

With the included correlation structure, our GLS model does end up satisfying the independence assumption, with just one possibly significant spike coming at a very late lag.

We set up an anova test, with the null hypothesis being that the the number of injuries is not affected by the team after accounting for yearly trends. Using 0.05 as the significance level, the Team term is significant.

So, we can reject our null hypothesis, meaning for these teams we observe injury rates that differ from the general norm, which cannot be attributed to chance.

A closer look at the Celtics, Heat, and Spurs, some of the franchises that saw great success from 2000-2023, shows that they found ways to win despite being more affected by injuries. Based



on the 95% Confidence Intervals for these teams that have p-values below the significance level, for any given year, being on the Heat/Celtics is associated with approximately 1-9 more injuries. This range jumps up to 5-13 for being on the Spurs. It seems like these teams found a way to win despite their injuries. Alternatively, their players may have had lots of minor injuries or mostly their rotation players got injured as opposed to key starters, or their injuries may have come in years they were not in title contention.

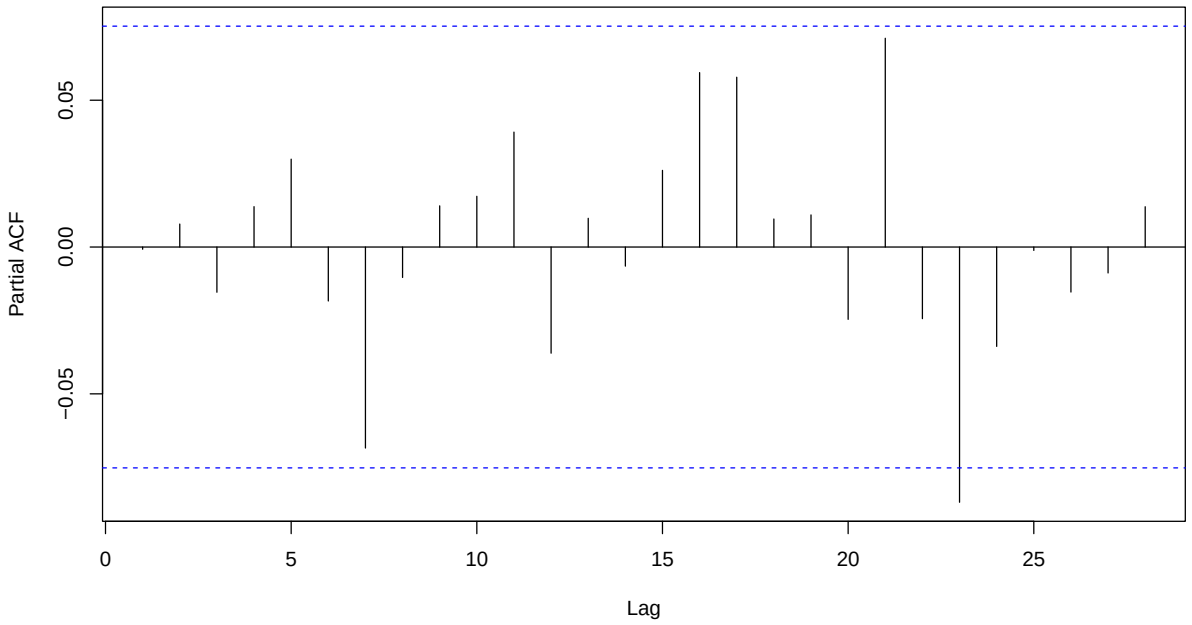
Interestingly, the confidence intervals for any of the teams with significant p-values are never negative. So, it seems like no team was exceptionally lucky/capable of preventing injuries more than other teams.

6 Additional Inference: Recovery Times

Another interesting aspect to examine is the recovery time and whether it has changed over time. Below is our recovery time analysis:

Note that the yearly recovery time is used because recovery periods often span multiple months, making monthly averages less accurate. It also smooths out fluctuations caused by some months having very few injuries or unusually short or long recovery times, providing more general trends. From plot 6, it appears that the yearly recovery time before 2005 was significantly higher than after 2005, possibly because of less advanced medical treatments and more physical play, which led to longer and more severe injuries. After 2005, recovery times decreased, likely due to improvements

Plot 5.1.2: PACF of ARIMA(2,0,3) Residuals with External Regressors



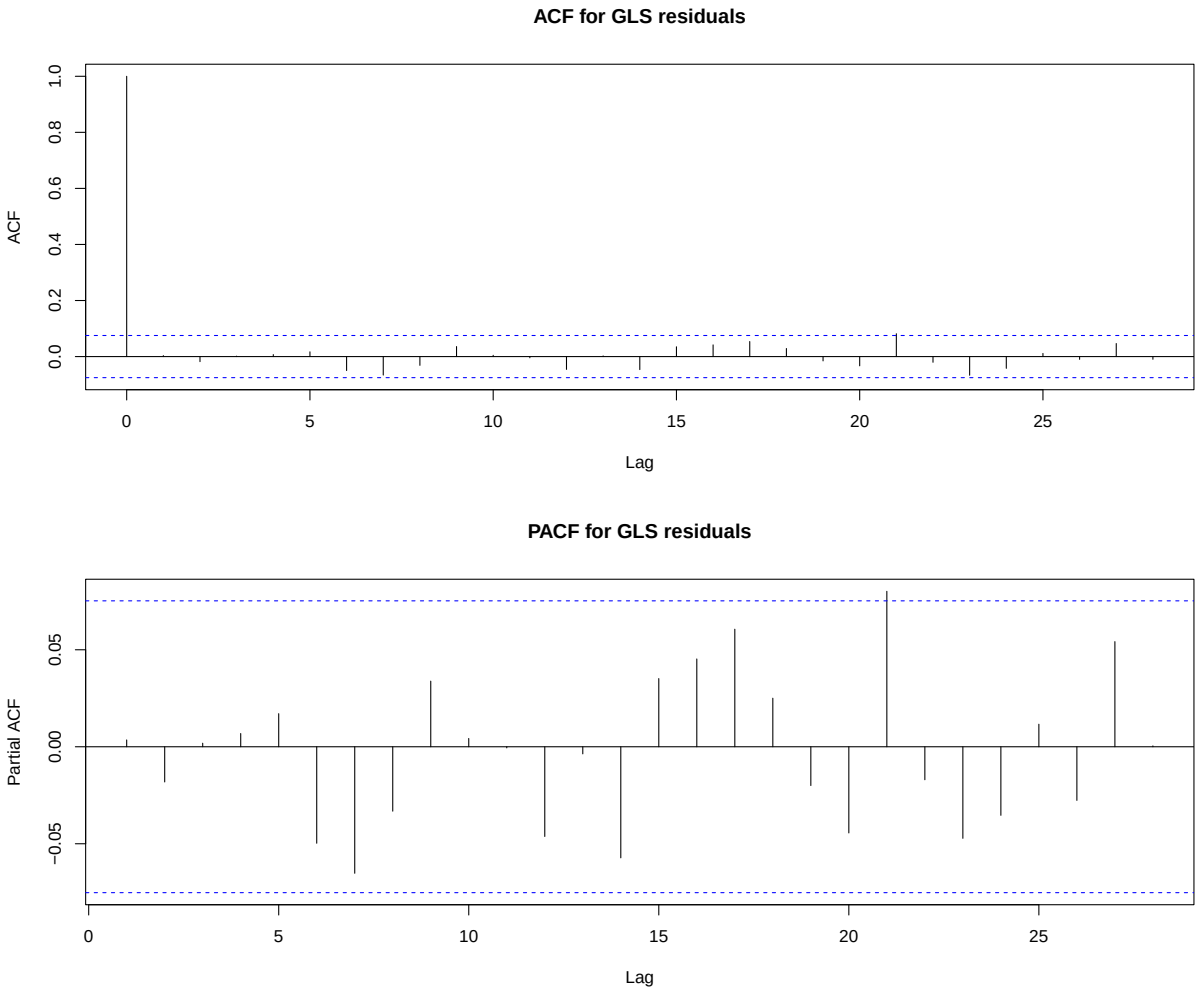
in sports medicine and training. The slight increase in 2020 could be due to COVID-19 disruptions, which affected the season schedule and may have prolonged recovery for some players.

7 Conclusions

Our time series analysis of NBA injuries from 2001 to 2023 revealed several significant patterns. First, we identified a distinctive "double-peak" seasonal pattern within each NBA season, with injury rates typically spiking mid-season (December-January) and again near the end of regular season, with a consistent dip during the All-Star break. Second, we observed a gradual increase in baseline injury rates over time, with a particularly sharp rise after 2020 during the COVID-19 pandemic.

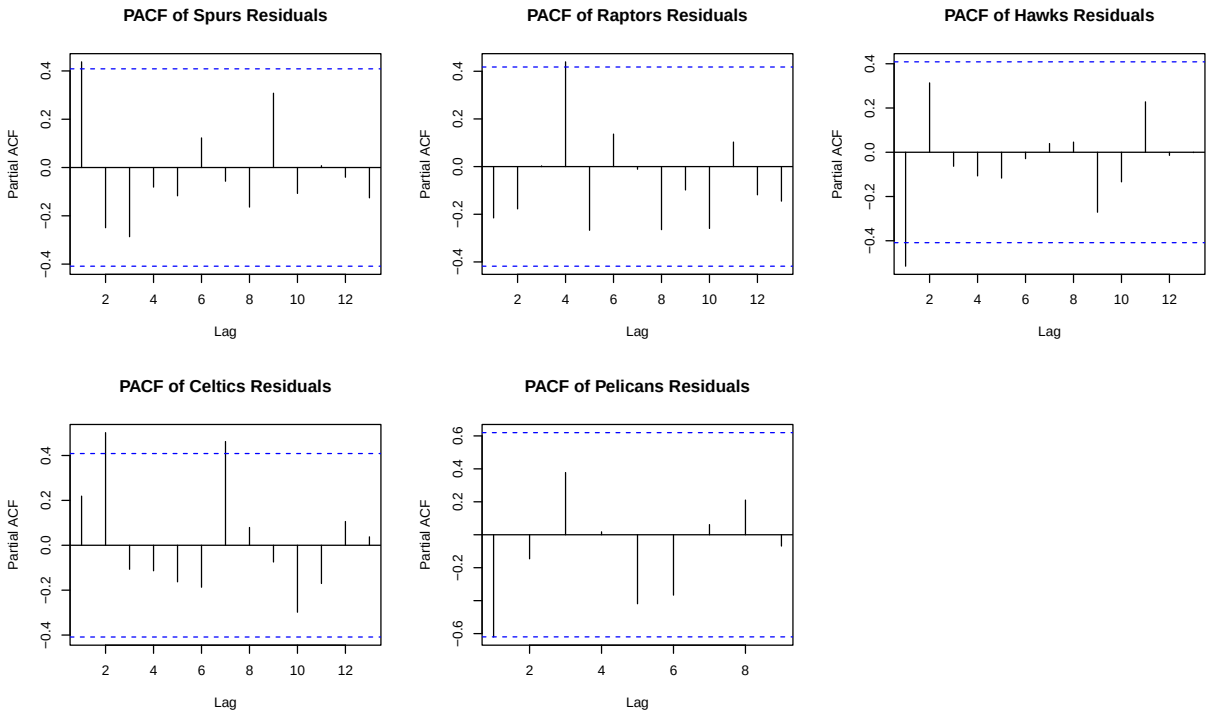
Using an ARIMA(1,0,0)(0,1,2)[12] model, we were able to capture both these seasonal patterns and the overall trend. Our model's forecasts suggest these patterns will likely continue through 2024-2028, though with increasing uncertainty over time. Additionally, our analysis of recovery times showed a notable improvement after 2005, likely due to advancements in sports medicine and training practices.

On the inference front, we tried to determine if injury rates differed by teams after accounting for yearly trends. Our simpler OLS modeling violated assumptions of regression, so we pivoted to a GLS model that accounted autocorrelation structures using auto.arima's() estimated ARMA model. After accounting for yearly trends, our t-test revealed that a few teams had significant



higher injury rates than others, but, surprisingly, none had significant lower injury rates. The assumptions for regression associated with this model were arguably satisfied, possibly validating our observed results.

These findings have practical implications for team management and medical staff. The consistent seasonal patterns suggest opportunities for proactive injury prevention, particularly during high-risk periods of December-January. However, the model's limitations, including its inability to account for remaining serial correlation, indicate that additional factors beyond temporal patterns influence injury rates in the NBA.



8 Limitations and Future Work

8.1 Study Limitations

8.1.1 Data Limitations:

Completeness:

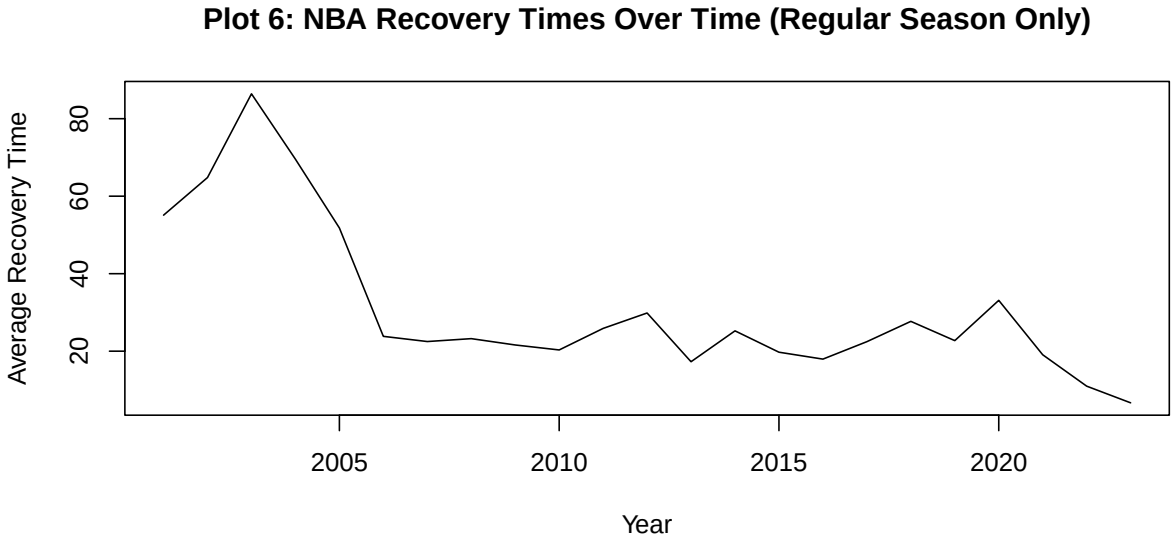
A limitation of the dataset is that certain months show no reported injuries, likely due to periods with few or no matches. Since these months vary across years, excluding them could introduce bias. Including them, however, results in gaps that may affect trend representation in the time series analysis. Additionally, the data only spans a limited time period since we have very limited data before 2000, which could restrict long-term trend analysis.

Bias:

The dataset may be subject to biases, such as underreporting of minor injuries, as teams might prioritize documenting only significant injuries. Additionally, variations in injury reporting standards across teams or seasons could lead to inconsistencies, affecting the reliability of comparisons over time or between teams.

External Factors:

The dataset does not include several external factors that could influence injury rates, such as training frequency. These factors may impact injury trends as well.



8.1.2 Model Limitations

Inference:

For inference, we explored Ordinary Least Squares (OLS) and Generalized Least Squares (GLS) methods as part of our analysis. However, the assumptions of constant variance and normality in the linear model may not have been fully satisfied, which affects the reliability of these methods.

Forecasting:

For forecasting, despite using the auto.arima model, the PACF and ACF plots indicate leftover serial autocorrelation in the residuals. This suggests that the model may not fully capture the underlying structure of the data. One potential reason for this is that we are fitting the raw time series data without incorporating additional relevant regressors, such as team-specific factors, training regimens, or changes in scheduling. These missing variables could account for some of the remaining autocorrelation, and their inclusion might improve model fit and forecast accuracy. Therefore, the inability to account for these external influences is a limitation of our current model.

8.1.3 Other Considerations

Contextual Factors:

Factors like player workload, travel schedules, and changes in game style could impact injury rates, but these were not addressed in the analysis. Including these factors might provide additional insights into injury trends. But it should also be noted that it is very hard to gather and quantify those factors as well.

Generalizability:

The findings may not generalize beyond the NBA or the specific dataset used. Additionally, we excluded the playoffs due to their increased variability, which would be difficult to account for and could skew injury trends.

8.2 Future Work

Model Enhancement

The model can be enhanced by incorporating key external variables such as game schedules, travel distance, and rest days into the ARIMA framework. The analysis could also be expanded by developing separate models for different injury types to improve prediction accuracy and extending the scope to include playoff data with appropriate seasonal adjustments.

Technical Enhancement

From a technical perspective, implementing advanced machine learning models like Random Forests and XGBoost would provide valuable comparisons with our ARIMA results.

Additional Research

Additional research directions could investigate injury patterns by player position, analyze the relationship between injury rates and team performance, and examine how specific medical protocols affect recovery times. These analyses would provide more nuanced insights for injury prevention.

9 Appendix

9.1 Data Processing Steps

```
#Section 1
NBA <- read_csv("~/Downloads/NBA Player Injury Stats(1951 - 2023).csv")

# Create a dataframe with playoff start dates
playoff_dates <- tribble(
  ~Year, ~playoff_start,
  2001, "2001-04-21",
  2002, "2002-04-20",
  2003, "2003-04-19",
  2004, "2004-04-17",
  2005, "2005-04-23",
  2006, "2006-04-22",
  2007, "2007-04-21",
  2008, "2008-04-19",
  2009, "2009-04-18",
  2010, "2010-04-17",
  2011, "2011-04-16",
  2012, "2012-04-28",
  2013, "2013-04-20",
  2014, "2014-04-19",
  2015, "2015-04-18",
  2016, "2016-04-16",
  2017, "2017-04-15",
  2018, "2018-04-14",
  2019, "2019-04-13",
  2020, "2020-08-17",
  2021, "2021-05-22",
  2022, "2022-04-16",
  2023, "2023-04-15"
) %>%
  mutate(playoff_start = as.Date(playoff_start))

# Process NBA data
NBA <- NBA %>%
  mutate(Date = as.Date(Date)) %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  filter(Year > 2000) %>%
  left_join(playoff_dates, by = "Year") %>%
  mutate(is_regular_season = Date < playoff_start | is.na(playoff_start))
```

```r
# Check months per year
months_per_year <- NBA %>%
  group_by(Year) %>%
  summarise(Months_of_Data = n_distinct(Month))

# Remove returning players and filter for regular season only
NBA_injuries_only <- NBA %>%
  filter(is.na(Relinquished)) %>%
  filter(is_regular_season)  # Only include regular season injuries

# Count injuries by year and month
NBA_counts <- NBA_injuries_only %>%
  group_by(Year, Month) %>%
  summarise(
    Total_Entries = n(),
    .groups = 'drop'
  ) %>%
  mutate(Month = as.character(Month)) %>%
  arrange(Year, Month)

# Create complete time series with zeros for missing values
NBA_imputed <- NBA_counts %>%
  complete(
    Year = unique(NBA_counts$Year),
    Month = as.character(1:12),
    fill = list(Total_Entries = 0)
  ) %>%
  arrange(Year, as.integer(Month))

# Create time series object
NBA_timeseries <- ts(NBA_imputed$Total_Entries,
                     start = c(2001, 1),
                     frequency = 12)

# Plot the time series
plot(NBA_timeseries,
     main = "NBA Injuries Over Time (Regular Season Only)",
     ylab = "Number of Injuries",
     xlab = "Year")

NBA_decomp <- decompose(NBA_timeseries)
plot(NBA_decomp)
```

9.2 Model Building & Forecasting

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

residuals <- NBA_decomp$random

# ACF and PACF of residuals
acf(residuals,
    main="ACF of Residuals\n(after removing trend & seasonality)",
    na.action=na.pass)
pacf(residuals,
     main="PACF of Residuals\n(after removing trend & seasonality)",
     na.action=na.pass)

par(mfrow=c(1,1))

library(zoo)
window_size <- 12  # One year window

# Calculate rolling mean and standard deviation
rolling_mean <- rollmean(NBA_timeseries, k=window_size, fill=NA)
rolling_sd <- rollapply(NBA_timeseries, width=window_size, FUN=sd, fill=NA)

# Plot them together
plot(NBA_timeseries,
     main="Plot 2.2: NBA Injuries with Rolling Statistics",
     ylab="Number of Injuries",
     type="l")
lines(rolling_mean, col="red", lwd=2)
lines(rolling_mean + 2*rolling_sd, col="blue", lty=2)
lines(rolling_mean - 2*rolling_sd, col="blue", lty=2)
legend("topleft",
       c("Original", "Rolling Mean", "±2 SD Bands"),
       col=c("black", "red", "blue"),
       lty=c(1,1,2))

library(forecast)
library(tseries)
library(lmtest)

arma_model <- Arima(NBA_timeseries, order=c(1,0,1))

# Fit auto.arima model
```

```r
auto_model <- auto.arima(NBA_timeseries)

# Compare models using AIC
models_comparison <- data.frame(
  Model = c("ARMA(1,1)", "Auto ARIMA"),
  AIC = c(arma_model$aic, auto_model$aic),
  BIC = c(arma_model$bic, auto_model$bic)
)

par(mfrow=c(1,2))
pacf(residuals(arma_model), main = "ARMA Residuals")
pacf(residuals(auto_model), main = "Auto ARIMA Residuals")
par(mfrow=c(1,1))

plot_data <- data.frame(
  date = time(NBA_timeseries),
  observed = as.numeric(NBA_timeseries),
  arma = as.numeric(arma_model$fitted),
  auto_arima = as.numeric(auto_model$fitted)
)

# Convert to long format for ggplot
plot_data_long <- plot_data %>%
  pivot_longer(cols = c(observed, arma, auto_arima),
               names_to = "model",
               values_to = "value")

# Create plot
ggplot(plot_data_long, aes(x = date, y = value, color = model)) +
  geom_line() +
  labs(title = "3.2 NBA Injuries: Observed vs Fitted Models",
       x = "Year",
       y = "Number of Injuries",
       color = "Model") +
  scale_color_manual(values = c("observed" = "black",
                                "arma" = "blue",
                                "auto_arima" = "red"),
                     labels = c("observed" = "Observed",
                                "arma" = "ARMA(1,1)",
                                "auto_arima" = paste("Auto ARIMA:", arimaorder(auto_model)[1:3]))) +
  theme_minimal() +
  theme(legend.position = "top")
```

```
auto_forecast <- forecast(auto_model, h = 5*12, level = 95)

obs_df <- NBA_imputed %>%
  mutate(time = c(time(NBA_timeseries)), fitted = fitted(auto_model)) %>%
  dplyr::select(time, Total_Entries, fitted) %>%
  rename(`observed injuries` = Total_Entries) %>%
  pivot_longer(-time)

forecast_df <- tibble(
  time = c(time(auto_forecast$mean)),
  value = c(auto_forecast$mean),
  lwr = c(auto_forecast$lower),
  upr = c(auto_forecast$upper),
  name = "forecast"
)

# Forecast plot
ggplot() +
  geom_line(data = obs_df, aes(x = time, y = value, col = name)) +
  geom_line(data = forecast_df, aes(x = time, y = value, col = name)) +
  geom_ribbon(data = forecast_df, aes(x = time, ymin = lwr, ymax = upr), alpha = 0.15) +
  theme_bw() +
  labs(title = "Plot 4: NBA Injury Forecast (2024-2028)",
       subtitle = "With 95% prediction intervals",
       x = "Year",
       y = "Number of Injuries",
       color = "Type") +
  theme(legend.position = "top")
```

9.3 Inference

```
NBA_counts <- NBA_injuries_only %>%
  group_by(Year, Team) %>%
  summarise(
    Total_Entries = n(),
    .groups = 'drop'
  ) %>%
  mutate(Team = factor(Team)) %>%
  arrange(Year)

NBA_imputed <- NBA_counts %>%
  complete(
```

```
    Year = unique(NBA_counts$Year),
    fill = list(Total_Entries = 0)
  ) %>%
  arrange(Year)

## OLS
# display setup
par(mfrow = c(2, 2))

# model
ols_fit <- lm(Total_Entries ~ Year + factor(Team), data = NBA_imputed)

# linearity, constant variance, normality assumptions & outliers
plot(ols_fit)

# independence assumptions
par(mfrow = c(1, 2))
pacf(resid(ols_fit), main = "ACF for OLS residuals")
pacf(resid(ols_fit), main = "PACF for OLS residuals")

## ARIMA inference
library(forecast)

# external regressors of team, year
xreg <- cbind(NBA_imputed$Year, factor(NBA_imputed$Team))

arima_fit <- auto.arima(NBA_imputed$Total_Entries, xreg = xreg)

# find the ARMA(p,q) processes
summary(arima_fit)

# serial autocorrelation
pacf(resid(arima_fit))

## GLS inference
gls_fit <- gls(Total_Entries ~ Year + Team,
                 correlation = corARMA(p = 2, q = 3, form = ~ 1 | Year),
                 data = NBA_imputed,
                 method = "ML")

acf(resid(gls_fit, type = "normalized"), main = "ACF for GLS residuals")
pacf(resid(gls_fit, type = "normalized"), main = "PACF for GLS residuals")
```

```
# analyze results
summary(gls_fit)
confint(gls_fit)
```

9.4 Additional Analysis

```
# only use recovery data in regular season
NBA_regular <- NBA %>%
  filter(is_regular_season == TRUE)

# separate relinquished and acquired data
relinquished <- NBA_regular %>%
  filter(!is.na(Relinquished)) %>%
  select(player = Relinquished, Date, Team) %>%
  mutate(status = "relinquished")

acquired <- NBA_regular %>%
  filter(!is.na(Acquired)) %>%
  select(player = Acquired, Date, Team) %>%
  mutate(status = "acquired")

# combine the two datasets
events <- rbind(relinquished, acquired) %>%
  arrange(player, Date)

# calculate recovery times
recovery_times <- events %>%
  group_by(player) %>%
  mutate(next_status = lead(status),
         next_date = lead(Date)) %>%
  filter(status == "relinquished" & next_status == "acquired") %>%
  mutate(recovery_time_days = as.numeric(next_date - Date)) %>%
  ungroup() %>%
  select(player, injury_date = Date, recovery_date = next_date, recovery_time_days)

# arrange recovery time based on injury times
recovery_times <- recovery_times %>%
  arrange(injury_date) %>%
  mutate(start_year = year(injury_date),
         start_month = month(injury_date))

# yearly recovery time
```

```r
recovery_times_yr <- recovery_times %>%
  group_by(start_year) %>%
  summarize(avg_recovery_time = mean(recovery_time_days))

# time series based on yearly recovery time
recovery_time_yr_ts <- ts(
  recovery_times_yr$avg_recovery_time,
  start = c(2001, 1),
  freq = 1
)

plot(recovery_time_yr_ts,
     main = "NBA Recovery Times Over Time (Regular Season Only)",
     ylab = "Average Recovery Time",
     xlab = "Year")
```

9.5 Session Information

```
## Session Information

R version: R version 4.4.1 (2024-06-14)

Operating System: aarch64-apple-darwin20

### Packages Used:

* tidyverse (2.0.0)
* forecast (8.23.0)
* nlme (3.1.164)
* tseries (0.10.58)
* zoo (1.8.12)
* ggplot2 (3.5.1)
* lmtest (0.9.40)
```