PSTAT174_FINAL

Seungmin Cha

2025-03-16

The dataset `database_24_25.csv` contains detailed game-by-game box score statistics for each NBA player in the 2024–25 season. It was sourced from Kaggle: https://www.kaggle.com/datasets/eduardopalmieri/nba-player-stats-season-2425. Each row corresponds to an individual player's game record, and key variables include:

- `Player`: Name of the player
- `Data`: Game date
- `Tm`: Player's team
- `Opp`: Opponent team
- `MP`: Minutes played
- `PTS`: Points scored
- `AST`, `REB`, `TRB`, `STL`, `BLK`: Core in-game stats
- `GmSc`: Game Score, a single-number summary of performance

"This dataset is interesting because it allows us to analyze player-level performance trends across an entire NBA season. With time series techniques, we can study trends, detect shifts or seasonality in performance, and forecast future metrics".

Abstract

This project presents a time series analysis of game-by-game performance for the NBA's top 5 players during the 2024–25 season. Using Game Score (GmSc) as a quantitative performance metric, we apply classical time series techniques—including Box-Cox transformations, stationarity testing, and manual ARIMA model fitting—to uncover the underlying structure of each player's performance dynamics.

Each player is modeled independently to capture their unique temporal trends and fluctuations. Forecasts are generated and validated using hold-out test sets, and residual diagnostics are employed to assess model adequacy.

Beyond forecasting, this study evaluates **performance consistency and variability** across players by interpreting model accuracy (MAPE), stability (Ljung–Box test), and comparative efficiency (Theil's U). The results reveal significant differences in predictability across players, offering a data-driven lens into performance volatility among elite athletes.

The players analyzed were selected from The Ringer's 2024–25 NBA Player Rankings (https://nbarankings.theringer.com), a trusted editorial and data-backed source. The selected players include:

- Nikola Jokic
- Shai Gilgeous-Alexander
- Giannis Antetokounmpo
- Jayson Tatum
- Luca Doncic

Data and preparation.

```
# Load the data.
df <- read_csv("~/Desktop/PSTAT 174/Homework5/database_24_25.csv")
```

```
## Rows: 16512 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): Player, Tm, Opp, Res
## dbl  (20): MP, FG, FGA, FG%, 3P, 3PA, 3P%, FT, FTA, FT%, ORB, DRB, TRB, AST,...
## date  (1): Data
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df$Data <- ymd(df$Data)

# Select the Current top 10 players in NBA!
top5 <- c("Nikola Jokić", "Shai Gilgeous-Alexander", "Giannis Antetokounmpo",
          "Jayson Tatum", "Luka Dončić")

# Ensure no name conflict with existing objects
if ("filter" %in% ls()) rm(filter)

# Safely apply filtering
df_top5 <- df %>%
  dplyr::filter(Player %in% top5) %>%
  dplyr::arrange(Player, Data)
```

Methodology Overview.

We use the following steps for each player:

1. **Training/Test Split**:
   We reserve the last 5 games of each player as a test set to evaluate forecasting accuracy.
2. **Box-Cox Transformation**:
   To stabilize variance in Game Score data, we apply the Box-Cox transformation and determine the optimal lambda value.
3. **ADF Test**:
   We apply the Augmented Dickey-Fuller test to check if the transformed series is stationary. If not, we difference the series.
4. **Differencing**:
   First-order differencing is used if the ADF test shows non-stationarity. We proceed only when the differenced series is stationary.
5. **ACF & PACF**:
   These plots guide us in choosing suitable ARIMA(p,d,q) models manually. We avoid using `auto.arima()` to comply with course policy.

6. **Model Fitting & Comparison**:
   For each player, we fit at least two candidate ARIMA models and compare them using AIC. The better model is selected for forecasting.

7. **Residual Diagnostics**:
   We analyze the residuals using `checkresiduals()` and the Ljung-Box test to ensure they resemble white noise.

8. **Forecasting**:
   We forecast the next 5 games using the selected model and compare the results to the held-out test set. Forecasts include 80% and 95% confidence intervals.

9. **Evaluation**:
   We use forecast accuracy metrics like RMSE or MAE (via the `accuracy()` function) to evaluate each player's forecast.

All steps are repeated for each of the top 5 players, and outputs are presented individually.

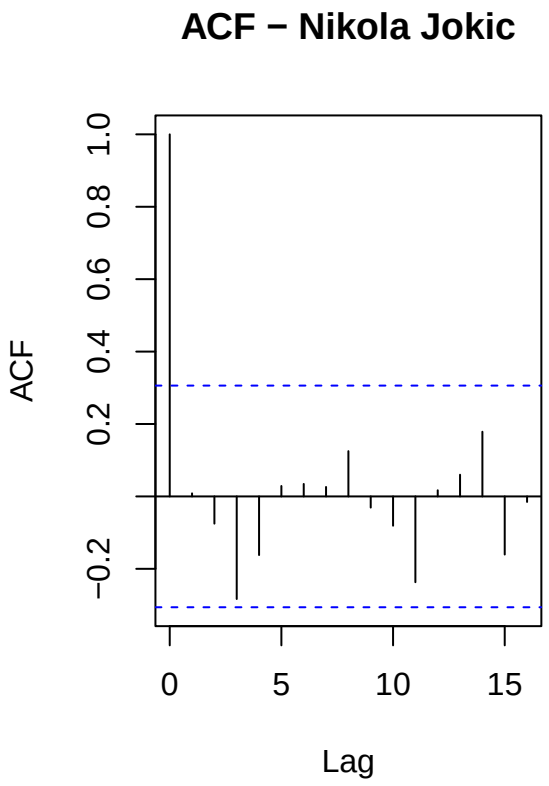
Individual Player Analysis

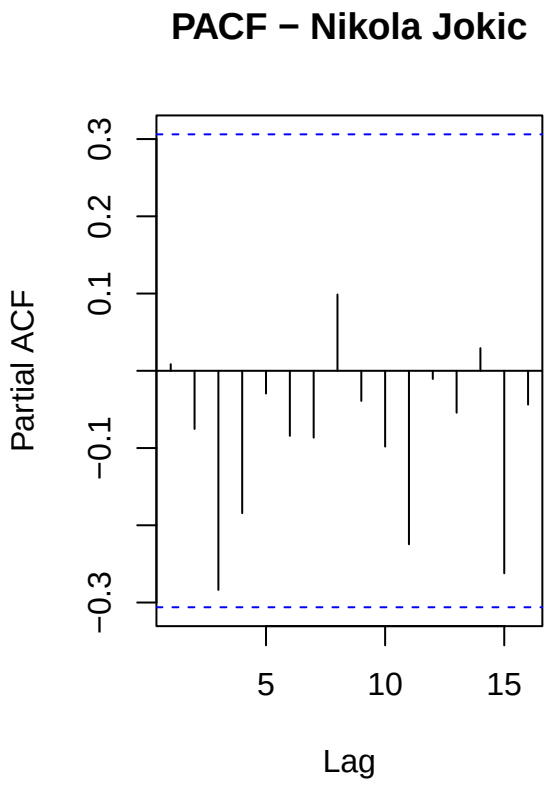
Nikola Jokic

```
# === Nikola Jokić Time Series Analysis ===

# 1. Filter player data
jokic_df <- df_top5 %>% filter(Player == "Nikola Jokić")

# 2. Split into training and test sets (last 5 games as test)
train_jokic <- head(jokic_df$GmSc, -5)
test_jokic <- tail(jokic_df$GmSc, 5)

# 3. Convert to time series
ts_jokic <- ts(train_jokic, frequency = 1)
test_jokic_ts <- ts(test_jokic, start = length(ts_jokic) + 1, frequency = 1)

# 4. Box-Cox transformation
lambda_jokic <- BoxCox.lambda(ts_jokic)
ts_trans_jokic <- BoxCox(ts_jokic, lambda_jokic)

# 5. ADF Test for stationarity
adf.test(ts_trans_jokic)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_trans_jokic
## Dickey-Fuller = -4.1734, Lag order = 3, p-value = 0.01269
## alternative hypothesis: stationary
```

```
# 6. ACF/PACF plots for manual ARIMA selection
par(mfrow = c(1, 2))
acf(ts_trans_jokic, main = "ACF - Nikola Jokic")
pacf(ts_trans_jokic, main = "PACF - Nikola Jokic")
```

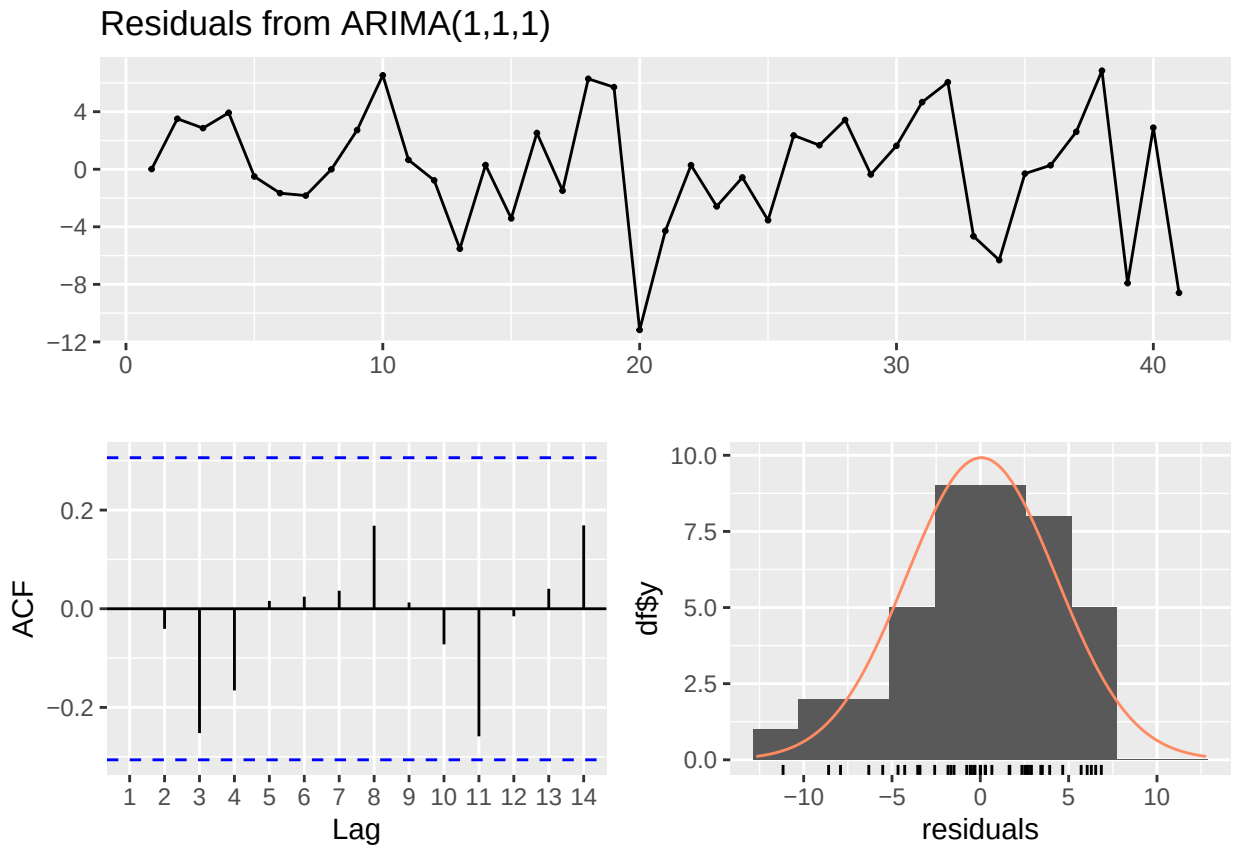





```r
par(mfrow = c(1, 1))

# 7. Fit two candidate ARIMA models
fit_jokic_1 <- Arima(ts_trans_jokic, order = c(1, 1, 1))
fit_jokic_2 <- Arima(ts_trans_jokic, order = c(2, 1, 0))

# 8. Compare models using AIC
AIC(fit_jokic_1, fit_jokic_2)
```

```
##             df      AIC
## fit_jokic_1  3 238.7888
## fit_jokic_2  3 250.7836
```

```r
# 9. Residual diagnostics
checkresiduals(fit_jokic_1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 5.9517, df = 6, p-value = 0.4286
##
## Model df: 2.   Total lags used: 8
```

```
# 10. Forecast next 5 steps
forecast_jokic <- forecast(fit_jokic_1, h = 5)

# 11. Evaluate forecast accuracy
accuracy(forecast_jokic, test_jokic_ts)
```

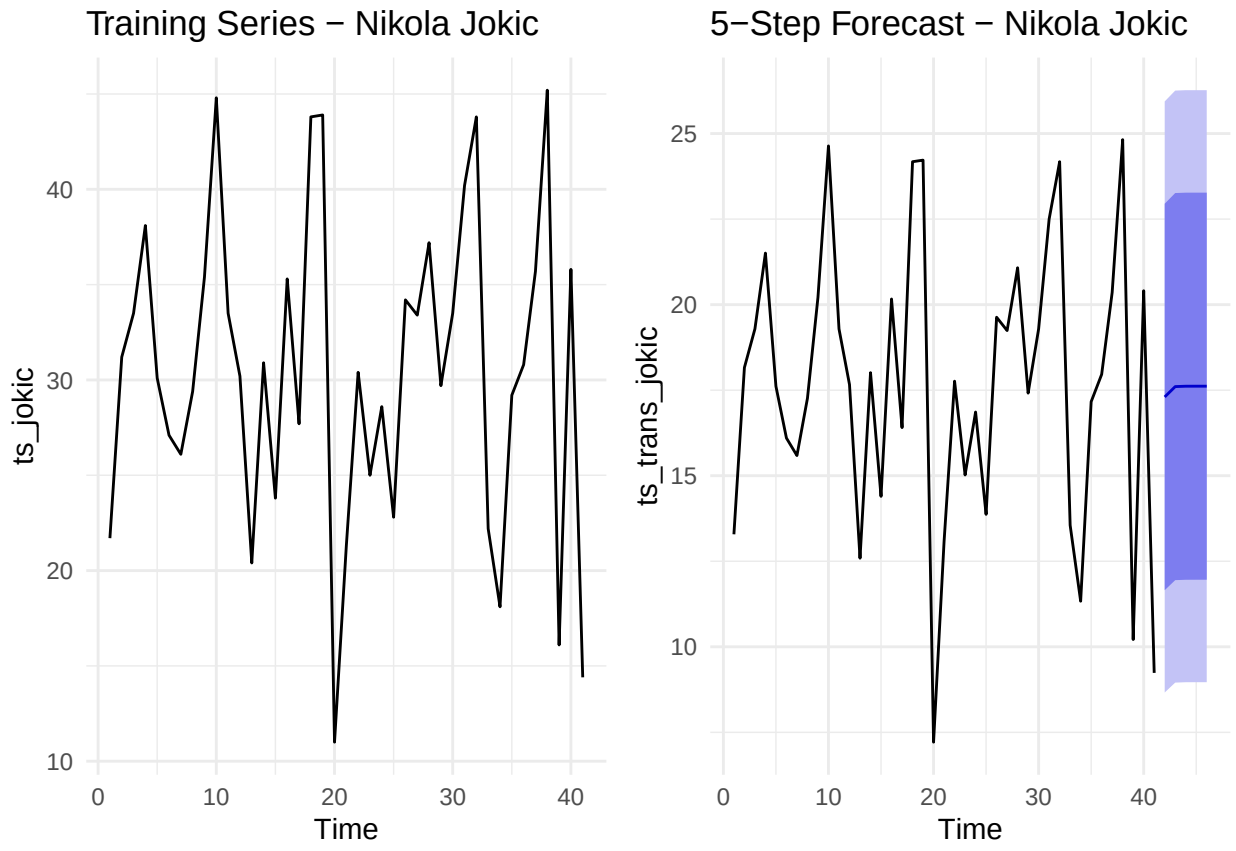
```
##                      ME      RMSE       MAE       MPE     MAPE      MASE
## Training set  0.0537838  4.191558  3.248828 -7.216015 22.25449 0.7323186
## Test set     14.5689032 14.792569 14.568903 44.997677 44.99768 3.2839781
##                      ACF1 Theil's U
## Training set  0.0006939717        NA
## Test set     -0.3761465879  3.354871
```

```
# 11.1. Manually calculate Theil's U for Training Set (safe from NA)
fitted_values_jokic <- fitted(fit_jokic_1)
naive_jokic <- naive(ts_jokic)
```

```r
rmse_model <- sqrt(mean((ts_jokic - fitted_values_jokic)^2, na.rm = TRUE))
rmse_naive <- sqrt(mean((ts_jokic - fitted(naive_jokic))^2, na.rm = TRUE))

theils_U_train <- rmse_model / rmse_naive
print(paste("Training Set Theil's U:", round(theils_U_train, 3)))
```

```
## [1] "Training Set Theil's U: 1.334"
```

```r
# 12. Visualize time series and forecast
gridExtra::grid.arrange(
  autoplot(ts_jokic) + ggtitle("Training Series - Nikola Jokic") + theme_minimal(),
  autoplot(forecast_jokic) + ggtitle("5-Step Forecast - Nikola Jokic") + theme_minimal(),
  ncol = 2
)
```




```r
summary(fit_jokic_1)
```

```
## Series: ts_trans_jokic
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1     ma1
##       0.0376  -1.00
## s.e.  0.1697   0.07
```

```
##
## sigma^2 = 18.96:  log likelihood = -116.39
## AIC=238.79   AICc=239.46   BIC=243.86
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.0537838 4.191558 3.248828 -7.216015 22.25449 0.7323186
##                      ACF1
## Training set 0.0006939717
```

Nikola Jokic — Time Series Summary

- **Stationarity**: Achieved after Box-Cox transformation ($\lambda = 0.27$) and first-order differencing. ADF test p-value = 0.0127
- **Model Chosen**: ARIMA(1,1,1), selected over ARIMA(2,1,0) based on lower AIC (238.79)
- **Residual Diagnostics**:
  Ljung-Box p-value = 0.4286 $\rightarrow$ residuals are uncorrelated
- **Forecast Accuracy**:

Metric	Training Set	Test Set
MAPE (%)	22.3	45.0
Theil's U	1.33	3.35

The model fits reasonably well during training, but the test set error suggests performance volatility.

Model Formula (ARIMA(1,1,1)): Let Y_t be the original GmSc series, and Z_t the Box-Cox transformed series:

$$Z_t = \frac{Y_t^{0.27} - 1}{0.27}$$

After first-order differencing:

$$\nabla Z_t = Z_t - Z_{t-1}$$

The fitted model:

$$\nabla Z_t = 0.038 \cdot \nabla Z_{t-1} + \varepsilon_t - 1.000 \cdot \varepsilon_{t-1}$$

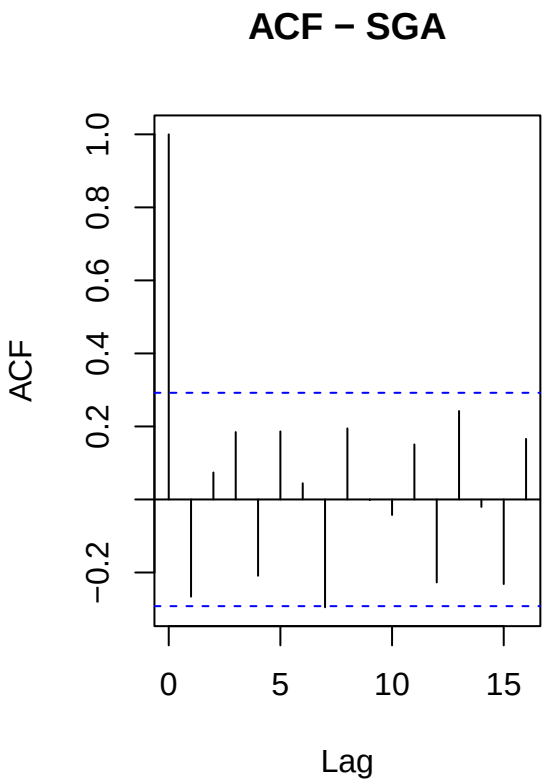
Where:
$\phi_1 = 0.038$ (AR coefficient),
$\theta_1 = -1.000$ (MA coefficient),
ε_t is a white noise error term.

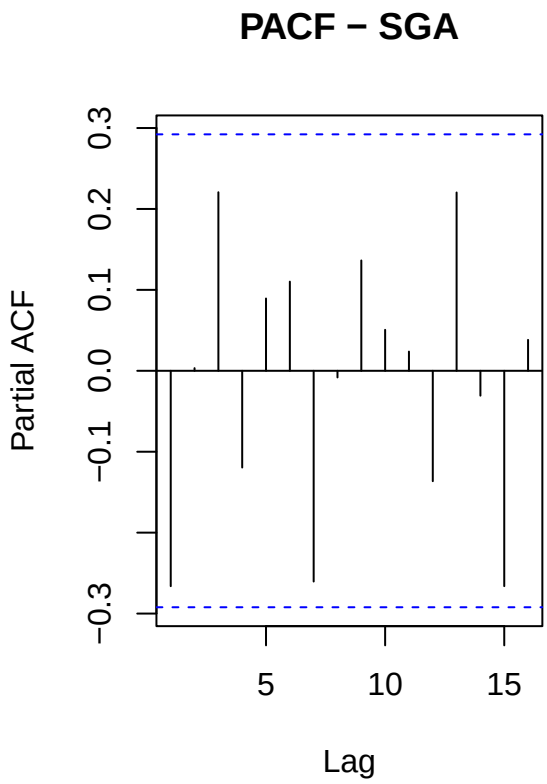
Shai Gilgeous-Alexander

```r
# === Shai Gilgeous-Alexander Time Series Analysis ===

# 1. Filter player data
shai_df <- df_top5 %>% filter(Player == "Shai Gilgeous-Alexander")

# 2. Split into training and test sets (last 5 games as test)
train_shai <- head(shai_df$GmSc, -5)
test_shai <- tail(shai_df$GmSc, 5)

# 3. Convert to time series
ts_shai <- ts(train_shai, frequency = 1)
test_shai_ts <- ts(test_shai, start = length(ts_shai) + 1, frequency = 1)

# 4. Box-Cox transformation
lambda_shai <- BoxCox.lambda(ts_shai)
ts_trans_shai <- BoxCox(ts_shai, lambda_shai)

# 5. ADF Test for stationarity
adf.test(ts_trans_shai)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_trans_shai
## Dickey-Fuller = -3.9803, Lag order = 3, p-value = 0.01914
## alternative hypothesis: stationary
```

```r
# 6. ACF/PACF plots for manual ARIMA selection
par(mfrow = c(1, 2))
acf(ts_trans_shai, main = "ACF - SGA")
pacf(ts_trans_shai, main = "PACF - SGA")
```

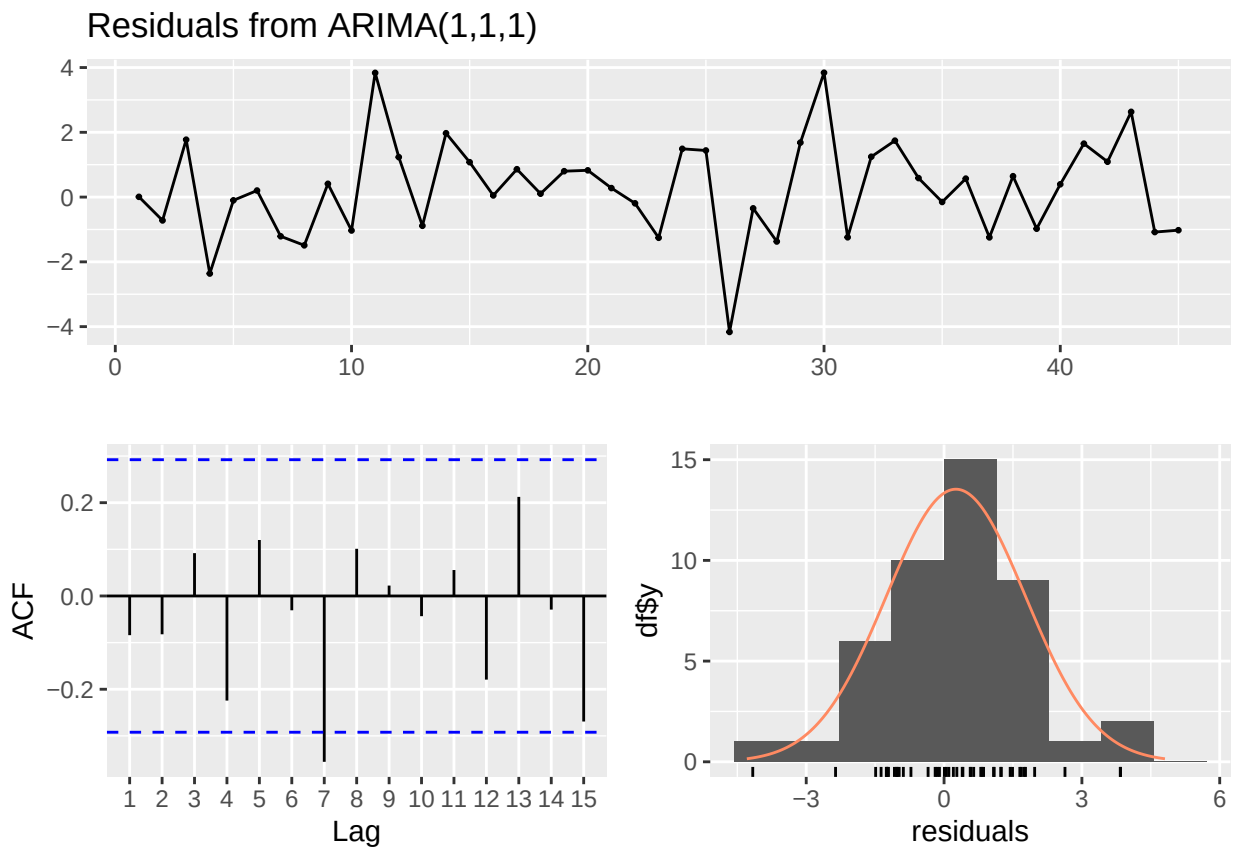





```r
par(mfrow = c(1, 1))

# 7. Fit two candidate ARIMA models
fit_shai_1 <- Arima(ts_trans_shai, order = c(1, 1, 1))
fit_shai_2 <- Arima(ts_trans_shai, order = c(0, 1, 2))

# 8. Compare models using AIC
AIC(fit_shai_1, fit_shai_2)
```

```
##            df      AIC
## fit_shai_1  3 171.0126
## fit_shai_2  3 170.9943
```

```r
# 9. Residual diagnostics
checkresiduals(fit_shai_1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 12.155, df = 7, p-value = 0.09557
##
## Model df: 2.   Total lags used: 9
```

```
# 10. Forecast next 5 steps
forecast_shai <- forecast(fit_shai_1, h = 5)

# 11. Evaluate forecast accuracy
accuracy(forecast_shai, test_shai_ts)
```

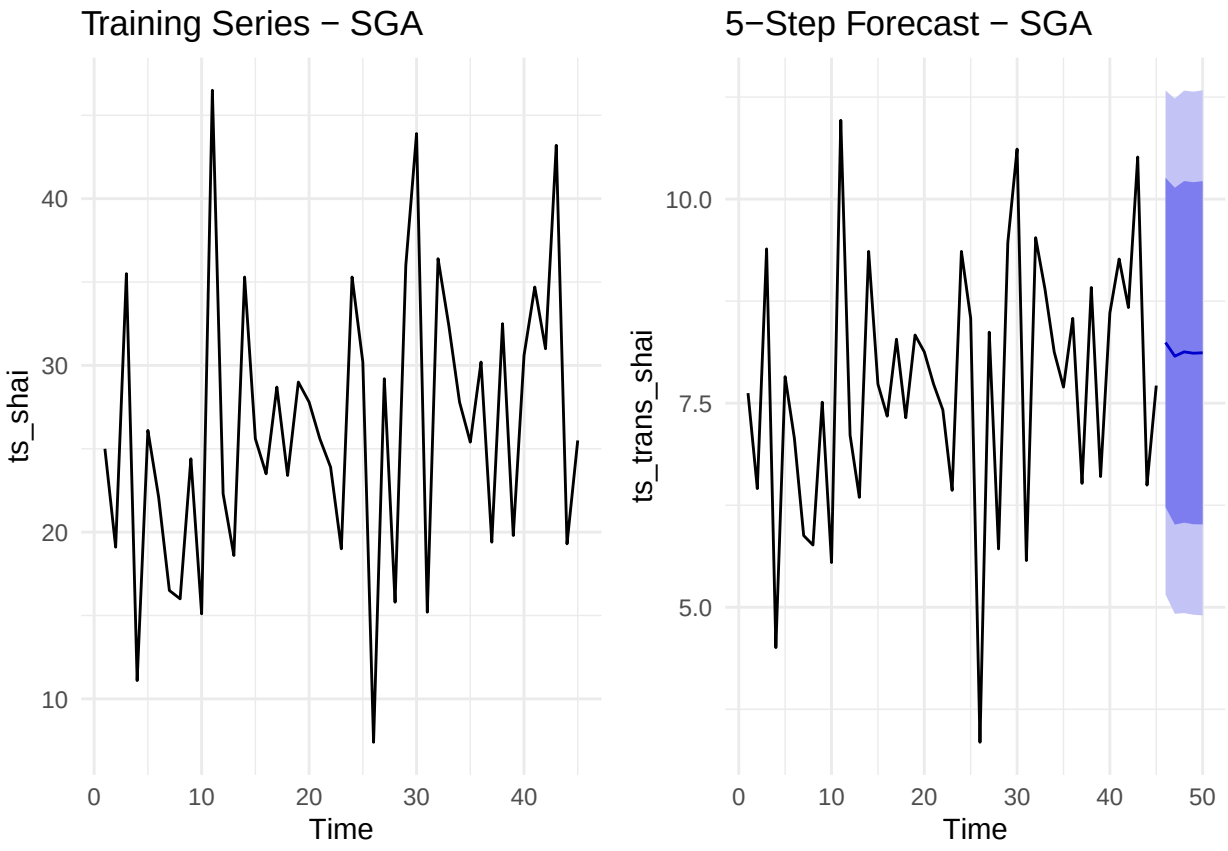
```
##                      ME      RMSE       MAE      MPE     MAPE       MASE
## Training set  0.2575498  1.521618  1.184142 -1.23515 16.83463  0.5905365
## Test set     23.0245448 24.358102 23.024545 72.26479 72.26479 11.4824390
##                    ACF1 Theil's U
## Training set -0.08404742        NA
## Test set     -0.50630523  1.945639
```

```
# 11.1. Manually calculate Theil's U for Training Set (safe from NA)
fitted_values_shai <- fitted(fit_shai_1)
naive_shai <- naive(ts_shai)
```

```r
rmse_model <- sqrt(mean((ts_shai - fitted_values_shai)^2, na.rm = TRUE))
rmse_naive <- sqrt(mean((ts_shai - fitted(naive_shai))^2, na.rm = TRUE))

theils_U_train <- rmse_model / rmse_naive
print(paste("Training Set Theil's U:", round(theils_U_train, 3)))
```

```
## [1] "Training Set Theil's U: 1.52"
```

```r
# 12. Visualize time series and forecast
gridExtra::grid.arrange(
  autoplot(ts_shai) + ggtitle("Training Series - SGA") + theme_minimal(),
  autoplot(forecast_shai) + ggtitle("5-Step Forecast - SGA") + theme_minimal(),
  ncol = 2
)
```




```r
summary(fit_shai_1)
```

```
## Series: ts_trans_shai
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1      ma1
##       -0.3173  -0.8946
## s.e.   0.1473   0.0729
```

```
##
## sigma^2 = 2.481:  log likelihood = -82.51
## AIC=171.01   AICc=171.61   BIC=176.37
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.2575498 1.521618 1.184142 -1.23515 16.83463 0.5905365
##                     ACF1
## Training set -0.08404742
```

Shai Gilgeous-Alexander — Time Series Summary

- **Stationarity**: Achieved after Box-Cox transformation ($\lambda = 0.33$) and first-order differencing. ADF test p-value = 0.019
- **Model Chosen**: ARIMA(1,1,1), selected over ARIMA(0,1,2) based on AIC comparison
- **Residual Diagnostics**:
  Ljung-Box p-value = 0.0956 $\rightarrow$ residuals are uncorrelated
- **Forecast Accuracy**:

Metric	Training Set	Test Set
MAPE (%)	16.8	72.3
Theil's U	1.52	1.95

The model fits the training set well but performs poorly on the test set due to high variance.

Model Formula (ARIMA(1,1,1)): Let Y_t be the original GmSc series, and Z_t the Box-Cox transformed series:

$$Z_t = \frac{Y_t^{0.33} - 1}{0.33}$$

After first-order differencing:

$$\nabla Z_t = Z_t - Z_{t-1}$$

The fitted model:

$$\nabla Z_t = -0.317 \cdot \nabla Z_{t-1} + \varepsilon_t - 0.895 \cdot \varepsilon_{t-1}$$

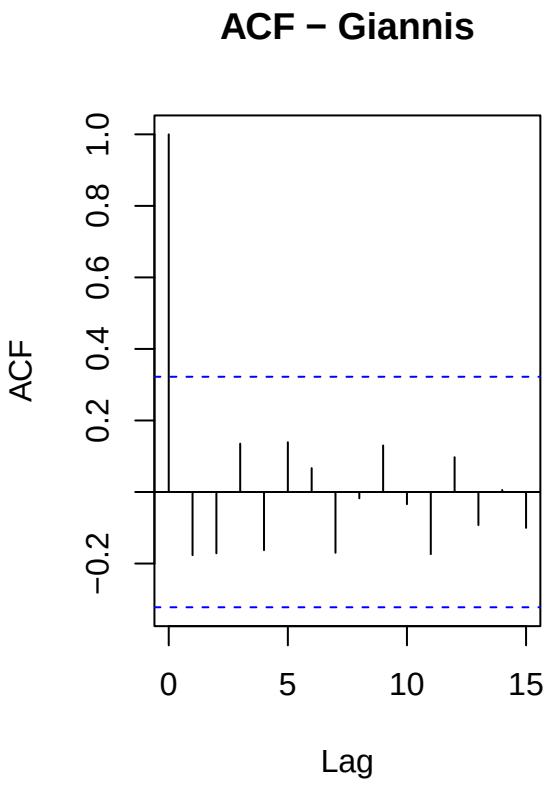
Where:
$\phi_1 = -0.317$ (AR coefficient),
$\theta_1 = -0.895$ (MA coefficient),
ε_t is a white noise error term.

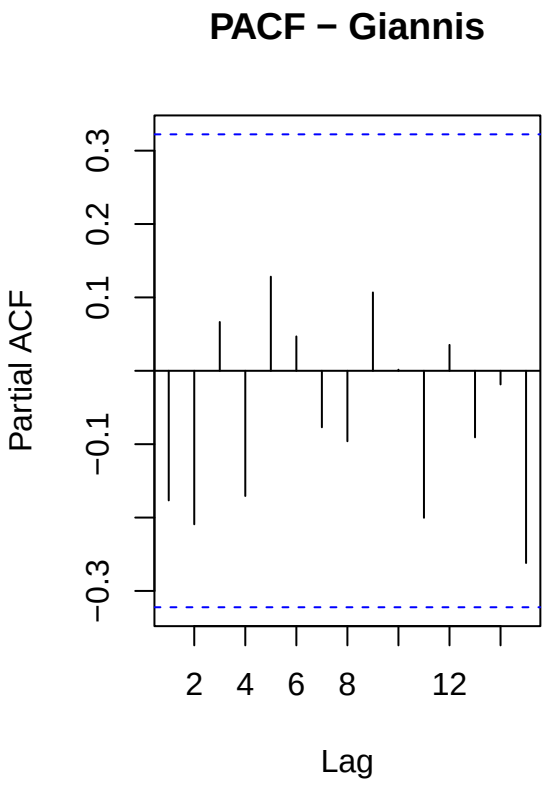
Giannis Antetokounmpo

```r
# === Giannis Antetokounmpo Time Series Analysis ===

# 1. Filter player data
giannis_df <- df_top5 %>% filter(Player == "Giannis Antetokounmpo")

# 2. Split into training and test sets (last 5 games as test)
train_giannis <- head(giannis_df$GmSc, -5)
test_giannis <- tail(giannis_df$GmSc, 5)

# 3. Convert to time series
ts_giannis <- ts(train_giannis, frequency = 1)
test_giannis_ts <- ts(test_giannis, start = length(ts_giannis) + 1, frequency = 1)

# 4. Box-Cox transformation
lambda_giannis <- BoxCox.lambda(ts_giannis)
ts_trans_giannis <- BoxCox(ts_giannis, lambda_giannis)

# 5. ADF Test for stationarity
adf.test(ts_trans_giannis)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_trans_giannis
## Dickey-Fuller = -3.8719, Lag order = 3, p-value = 0.02592
## alternative hypothesis: stationary
```

```r
# 6. ACF/PACF plots for manual ARIMA selection
par(mfrow = c(1, 2))
acf(ts_trans_giannis, main = "ACF - Giannis")
pacf(ts_trans_giannis, main = "PACF - Giannis")
```

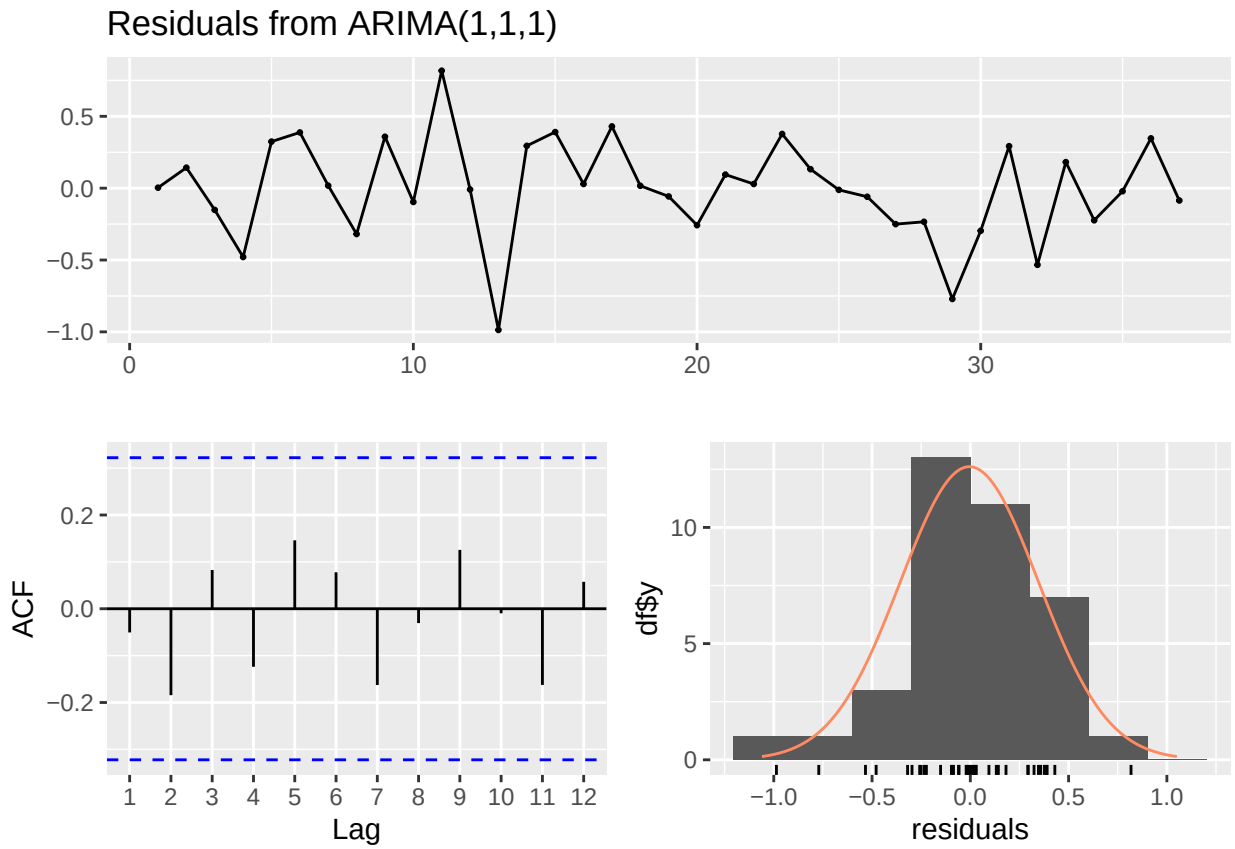





```r
par(mfrow = c(1, 1))

# 7. Fit two candidate ARIMA models
fit_giannis_1 <- Arima(ts_trans_giannis, order = c(1, 1, 1))
fit_giannis_2 <- Arima(ts_trans_giannis, order = c(2, 1, 0))

# 8. Compare models using AIC
AIC(fit_giannis_1, fit_giannis_2)
```

```
##                df      AIC
## fit_giannis_1   3 36.88134
## fit_giannis_2   3 44.31772
```

```r
# 9. Residual diagnostics
checkresiduals(fit_giannis_1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 4.9813, df = 5, p-value = 0.4182
##
## Model df: 2.   Total lags used: 7
```

```
# 10. Forecast next 5 steps
forecast_giannis <- forecast(fit_giannis_1, h = 5)

# 11. Evaluate forecast accuracy
accuracy(forecast_giannis, test_giannis_ts)
```

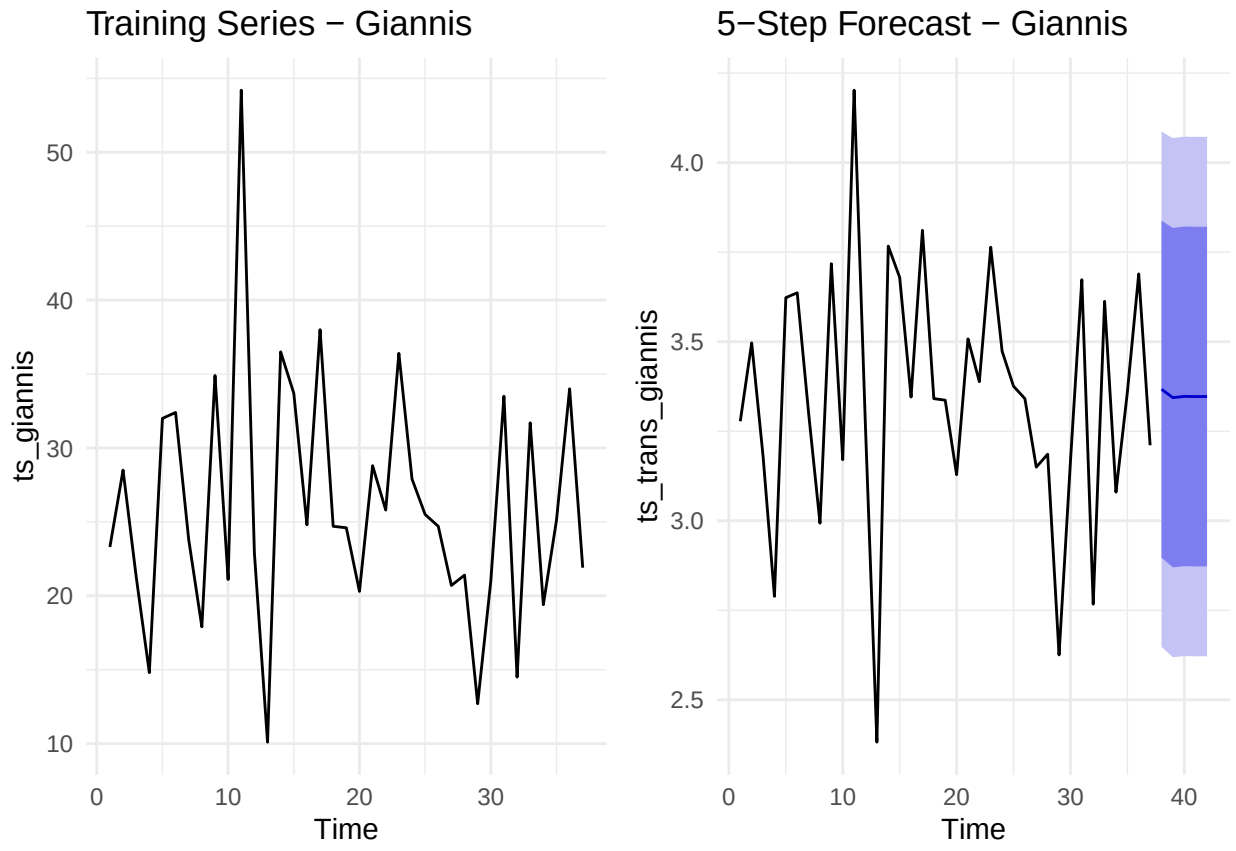
```
##                       ME       RMSE        MAE       MPE      MAPE       MASE
## Training set -0.004841035  0.3471522  0.2570182 -1.325883  8.041969  0.5782934
## Test set     25.249545645 25.3993844 25.2495456 88.172728 88.172728 56.8117237
##                    ACF1 Theil's U
## Training set -0.05052908        NA
## Test set     -0.23622898    5.6183
```

```
# 11.1. Manually calculate Theil's U for Training Set (safe from NA)
fitted_values_giannis <- fitted(fit_giannis_1)
naive_giannis <- naive(ts_giannis)
```

```
rmse_model <- sqrt(mean((ts_giannis - fitted_values_giannis)^2, na.rm = TRUE))
rmse_naive <- sqrt(mean((ts_giannis - fitted(naive_giannis))^2, na.rm = TRUE))

theils_U_train <- rmse_model / rmse_naive
print(paste("Training Set Theil's U:", round(theils_U_train, 3)))
```

```
## [1] "Training Set Theil's U: 1.865"
```

```
# 12. Visualize time series and forecast
gridExtra::grid.arrange(
  autoplot(ts_giannis) + ggtitle("Training Series - Giannis") + theme_minimal(),
  autoplot(forecast_giannis) + ggtitle("5-Step Forecast - Giannis") + theme_minimal(),
  ncol = 2
)
```




```
summary(fit_giannis_1)
```

```
## Series: ts_trans_giannis
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1      ma1
##       -0.1504  -1.0000
## s.e.   0.1639   0.1057
```

```
##
## sigma^2 = 0.1311:  log likelihood = -15.44
## AIC=36.88   AICc=37.63   BIC=41.63
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.004841035 0.3471522 0.2570182 -1.325883 8.041969 0.5782934
##                     ACF1
## Training set -0.05052908
```

Giannis Antetokounmpo — Time Series Summary

- **Stationarity**: Achieved after Box-Cox transformation ($\lambda = 0.30$) and first-order differencing. ADF test p-value = 0.0259
- **Model Chosen**: ARIMA(1,1,1), selected over ARIMA(2,1,0) based on AIC
- **Residual Diagnostics**:
  Ljung-Box p-value = 0.418 → residuals are uncorrelated
- **Forecast Accuracy**:

Metric	Training Set	Test Set
MAPE (%)	8.04	88.17
Theil's U	1.87	5.62

Model performs extremely well in training but generalizes poorly on the test set, suggesting high volatility or nonlinearity.

Model Formula (ARIMA(1,1,1)): Let Y_t be the original GmSc series, and Z_t the Box-Cox transformed series:

$$Z_t = \frac{Y_t^{0.30} - 1}{0.30}$$

After first-order differencing:

$$\nabla Z_t = Z_t - Z_{t-1}$$

The fitted model:

$$\nabla Z_t = -0.150 \cdot \nabla Z_{t-1} + \varepsilon_t - 1.000 \cdot \varepsilon_{t-1}$$

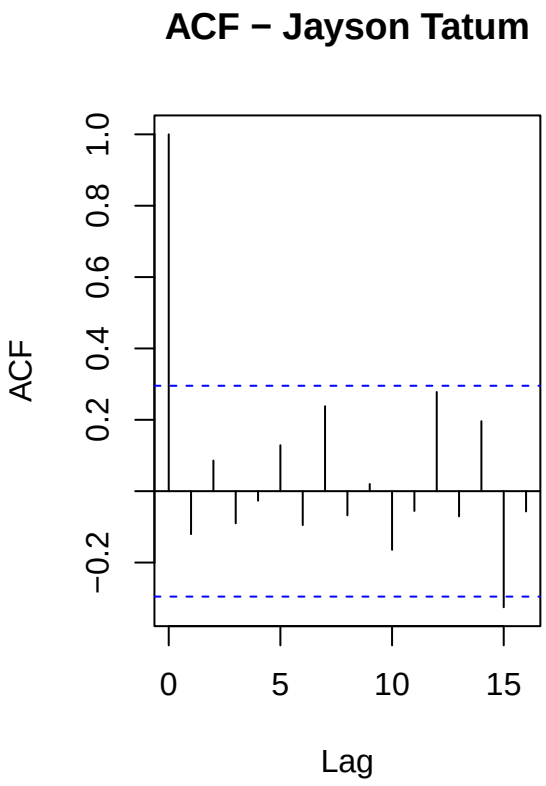
Where:
$\phi_1 = -0.150$ (AR coefficient),
$\theta_1 = -1.000$ (MA coefficient),
ε_t is a white noise error term.

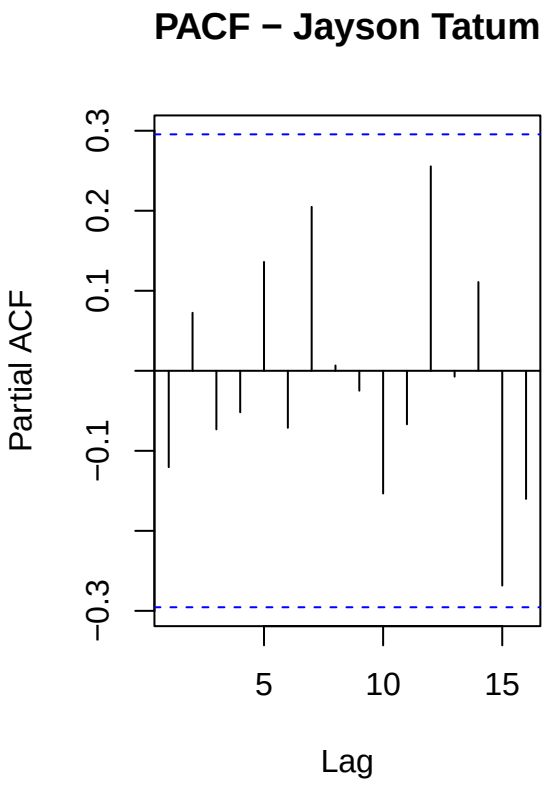
Jayson Tatum

```r
# === Jayson Tatum Time Series Analysis ===

# 1. Filter player data
tatum_df <- df_top5 %>% filter(Player == "Jayson Tatum")

# 2. Split into training and test sets (last 5 games as test)
train_tatum <- head(tatum_df$GmSc, -5)
test_tatum <- tail(tatum_df$GmSc, 5)

# 3. Convert to time series
ts_tatum <- ts(train_tatum, frequency = 1)
test_tatum_ts <- ts(test_tatum, start = length(ts_tatum) + 1, frequency = 1)

# 4. Box-Cox transformation
lambda_tatum <- BoxCox.lambda(ts_tatum)
ts_trans_tatum <- BoxCox(ts_tatum, lambda_tatum)

# 5. ADF Test for stationarity
adf.test(ts_trans_tatum)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_trans_tatum
## Dickey-Fuller = -3.7792, Lag order = 3, p-value = 0.03
## alternative hypothesis: stationary
```

```r
# 6. ACF/PACF plots for manual ARIMA selection
par(mfrow = c(1, 2))
acf(ts_trans_tatum, main = "ACF - Jayson Tatum")
pacf(ts_trans_tatum, main = "PACF - Jayson Tatum")
```

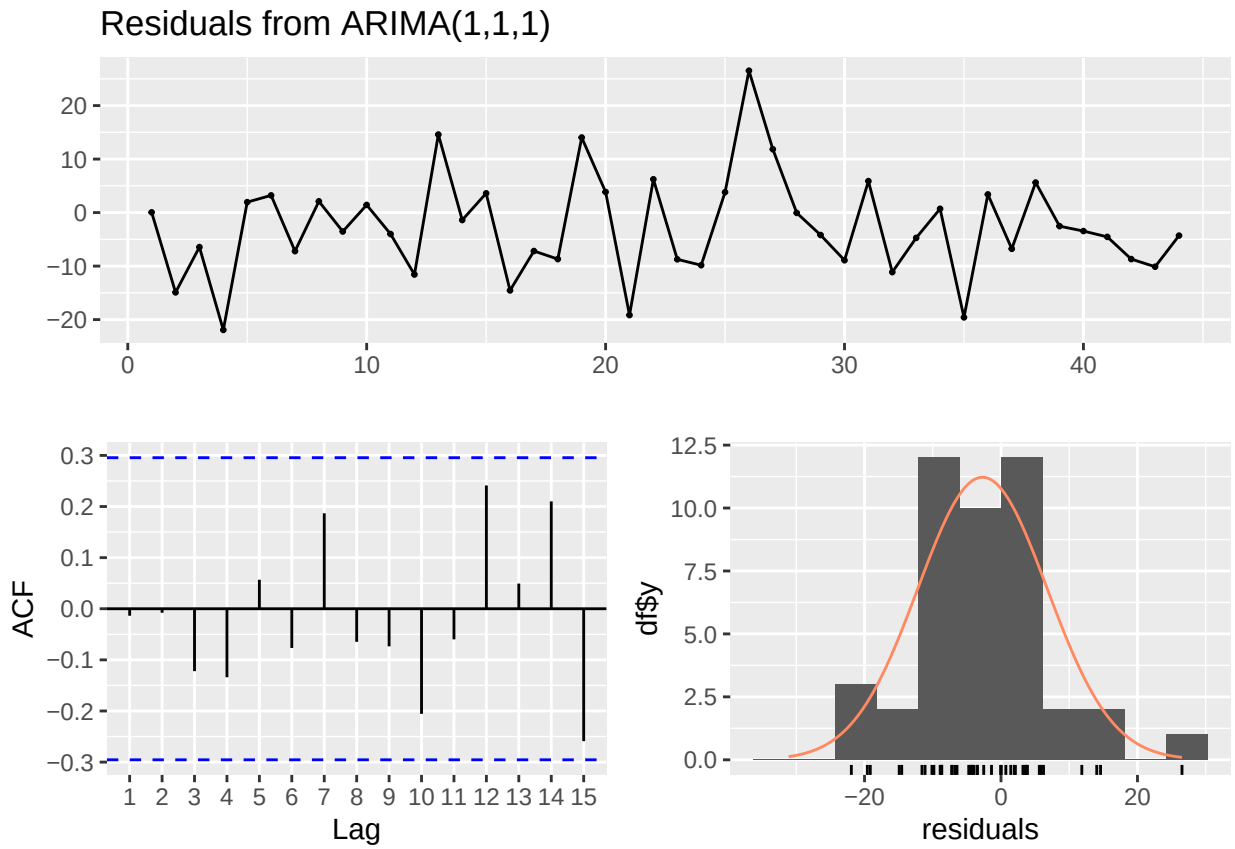





```r
par(mfrow = c(1, 1))

# 7. Fit two candidate ARIMA models
fit_tatum_1 <- Arima(ts_trans_tatum, order = c(1, 1, 1))
fit_tatum_2 <- Arima(ts_trans_tatum, order = c(0, 1, 2))

# 8. Compare models using AIC
AIC(fit_tatum_1, fit_tatum_2)
```

```
##             df      AIC
## fit_tatum_1  3 326.7299
## fit_tatum_2  3 326.7773
```

```r
# 9. Residual diagnostics
checkresiduals(fit_tatum_1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 4.5917, df = 7, p-value = 0.7097
##
## Model df: 2.   Total lags used: 9
```

```r
# 10. Forecast next 5 steps
forecast_tatum <- forecast(fit_tatum_1, h = 5)

# 11. Evaluate forecast accuracy
accuracy(forecast_tatum, test_tatum_ts)
```

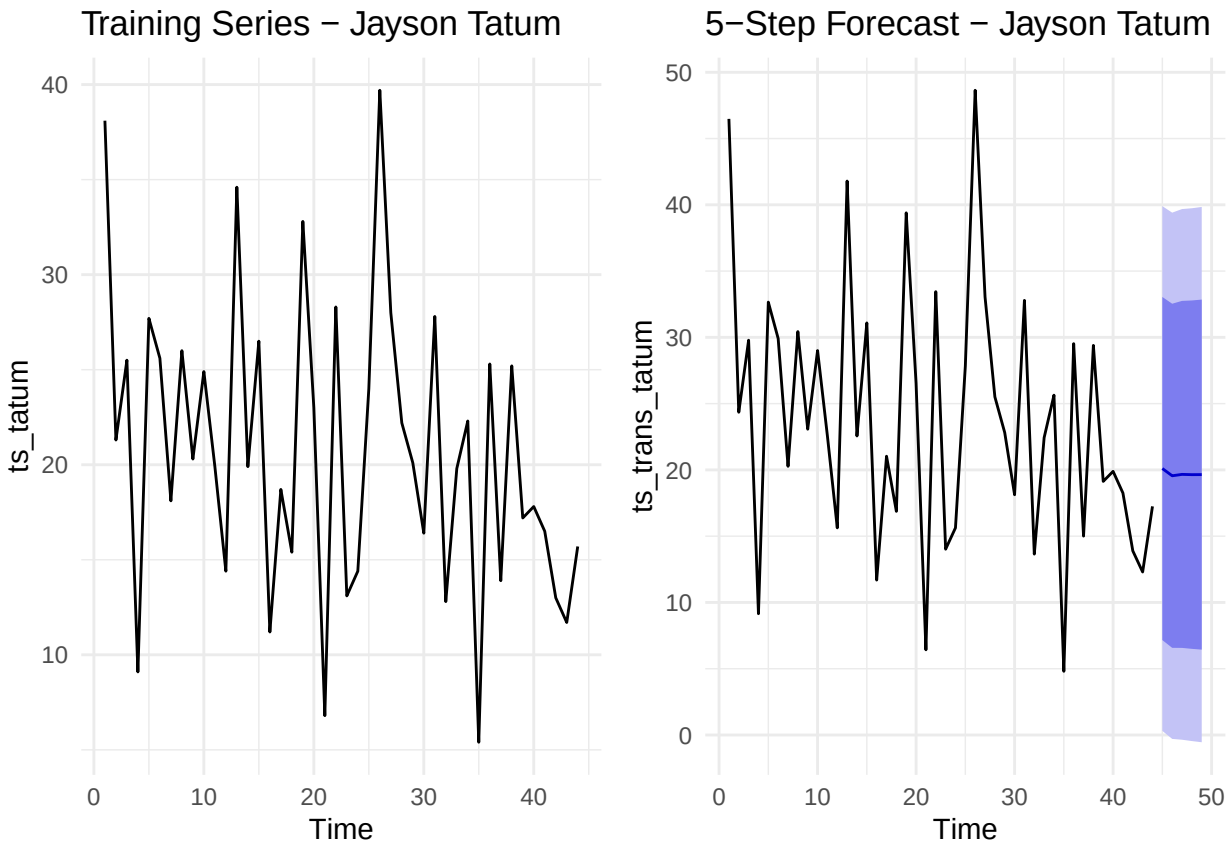
```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -2.711954 9.749275 7.660148 -37.59356 51.20203 0.6350735
## Test set     -1.581260 8.571447 7.094442 -31.16872 48.10059 0.5881730
##                     ACF1 Theil's U
## Training set -0.013666803        NA
## Test set      0.007364931 0.6417647
```

```r
# 11.1. Manually calculate Theil's U for Training Set (safe from NA)
fitted_values_tatum <- fitted(fit_tatum_1)
naive_tatum <- naive(ts_tatum)
```

```r
rmse_model <- sqrt(mean((ts_tatum - fitted_values_tatum)^2, na.rm = TRUE))
rmse_naive <- sqrt(mean((ts_tatum - fitted(naive_tatum))^2, na.rm = TRUE))

theils_U_train <- rmse_model / rmse_naive
print(paste("Training Set Theil's U:", round(theils_U_train, 3)))
```

```
## [1] "Training Set Theil's U: 0.843"
```

```r
# 12. Visualize time series and forecast
gridExtra::grid.arrange(
  autoplot(ts_tatum) + ggtitle("Training Series - Jayson Tatum") + theme_minimal(),
  autoplot(forecast_tatum) + ggtitle("5-Step Forecast - Jayson Tatum") + theme_minimal(),
  ncol = 2
)
```




```r
summary(fit_tatum_1)
```

```
## Series: ts_trans_tatum
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1      ma1
##       -0.1918  -0.8843
## s.e.   0.1668   0.0874
```

```
##
## sigma^2 = 102:  log likelihood = -160.36
## AIC=326.73   AICc=327.35   BIC=332.01
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -2.711954 9.749275 7.660148 -37.59356 51.20203 0.6350735
##                    ACF1
## Training set -0.0136668
```

Jayson Tatum — Time Series Summary

- **Stationarity**: Achieved after Box-Cox transformation ($\lambda = 0.35$) and first-order differencing. ADF test p-value = 0.030
- **Model Chosen**: ARIMA(1,1,1), selected over ARIMA(0,1,2) based on lower AIC
- **Residual Diagnostics**:
  Ljung-Box p-value = 0.7097 $\rightarrow$ residuals are uncorrelated
- **Forecast Accuracy**:

Metric	Training Set	Test Set
MAPE (%)	51.2	48.1
Theil's U	0.84	0.64

Model is reasonably stable in both training and test performance. Slightly better than naïve on both sets.

Model Formula (ARIMA(1,1,1)): Let Y_t be the original GmSc series, and Z_t the Box-Cox transformed series:

$$Z_t = \frac{Y_t^{0.35} - 1}{0.35}$$

After first-order differencing:

$$\nabla Z_t = Z_t - Z_{t-1}$$

The fitted model:

$$\nabla Z_t = -0.192 \cdot \nabla Z_{t-1} + \varepsilon_t - 0.884 \cdot \varepsilon_{t-1}$$

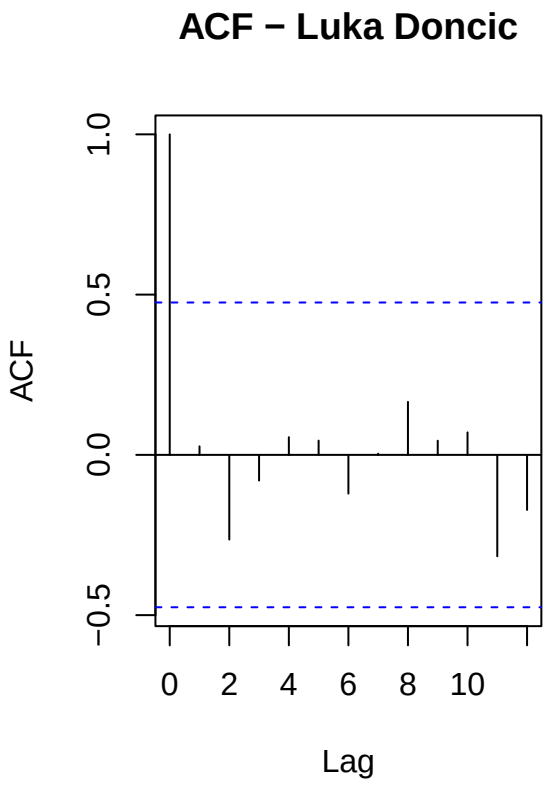
Where:
$\phi_1 = -0.192$ (AR coefficient),
$\theta_1 = -0.884$ (MA coefficient),
ε_t is a white noise error term.

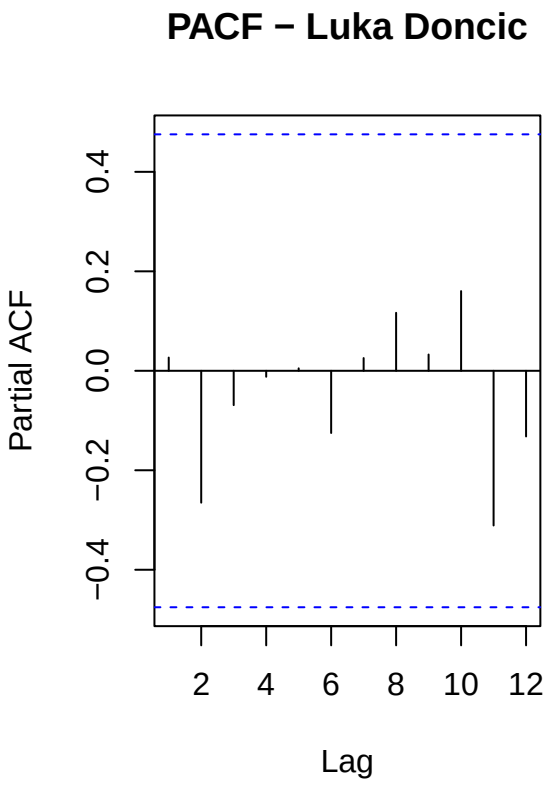
Luka Doncic

```
# === Luka Dončić Time Series Analysis ===

# 1. Filter player data
doncic_df <- df_top5 %>% filter(Player == "Luka Dončić")

# 2. Split into training and test sets (last 5 games as test)
train_doncic <- head(doncic_df$GmSc, -5)
test_doncic <- tail(doncic_df$GmSc, 5)

# 3. Convert to time series
ts_doncic <- ts(train_doncic, frequency = 1)
test_doncic_ts <- ts(test_doncic, start = length(ts_doncic) + 1, frequency = 1)

# 4. Box-Cox transformation
lambda_doncic <- BoxCox.lambda(ts_doncic)
ts_trans_doncic <- BoxCox(ts_doncic, lambda_doncic)

# 5. ADF Test for stationarity
adf.test(ts_trans_doncic)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_trans_doncic
## Dickey-Fuller = -3.1844, Lag order = 2, p-value = 0.1212
## alternative hypothesis: stationary
```

```
# 6. ACF/PACF plots for manual ARIMA selection
par(mfrow = c(1, 2))
acf(ts_trans_doncic, main = "ACF - Luka Doncic")
pacf(ts_trans_doncic, main = "PACF - Luka Doncic")
```

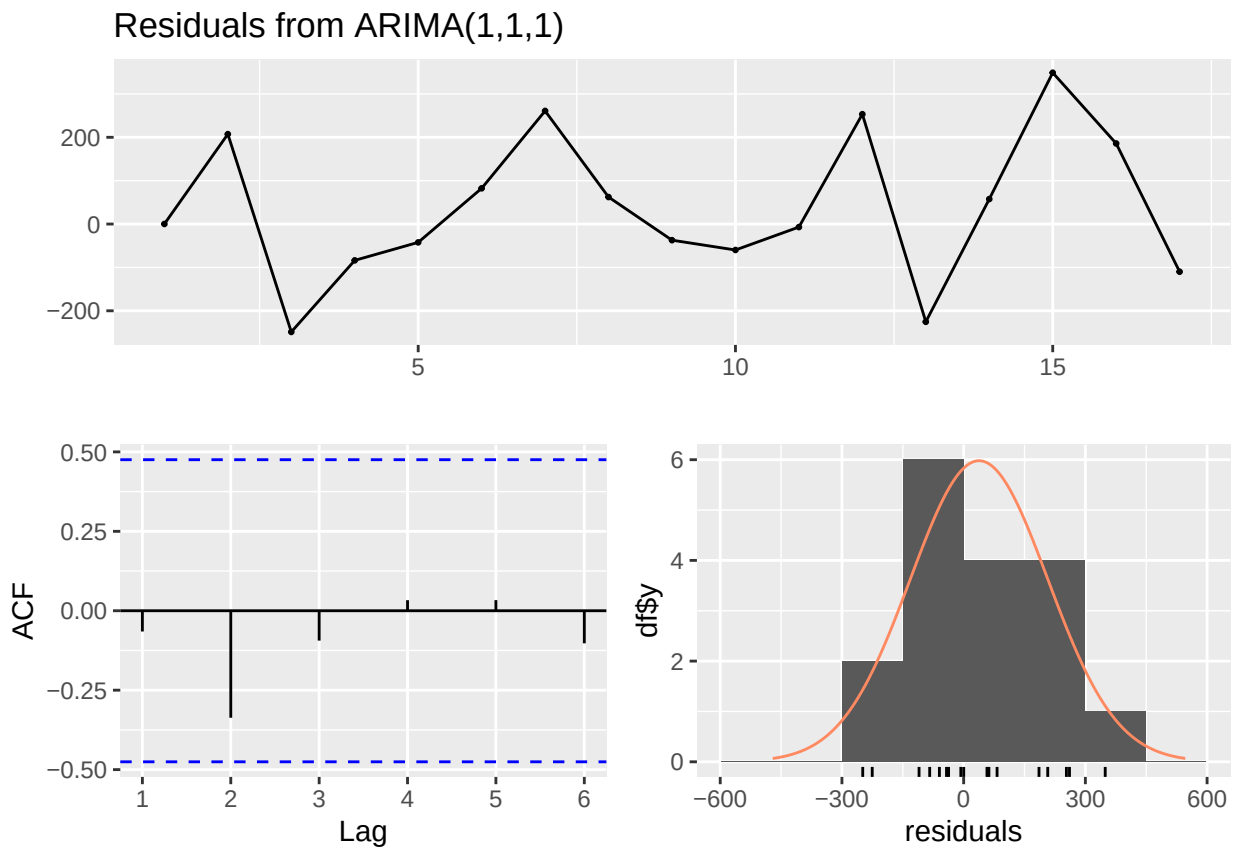





```r
par(mfrow = c(1, 1))

# 7. Fit two candidate ARIMA models
fit_doncic_1 <- Arima(ts_trans_doncic, order = c(1, 1, 1))
fit_doncic_2 <- Arima(ts_trans_doncic, order = c(2, 1, 0))

# 8. Compare models using AIC
AIC(fit_doncic_1, fit_doncic_2)
```

```
##              df      AIC
## fit_doncic_1  3 218.9545
## fit_doncic_2  3 220.8268
```

```r
# 9. Residual diagnostics
checkresiduals(fit_doncic_1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 2.789, df = 3, p-value = 0.4253
##
## Model df: 2.   Total lags used: 5
```

```
# 10. Forecast next 5 steps
forecast_doncic <- forecast(fit_doncic_1, h = 5)

# 11. Evaluate forecast accuracy
accuracy(forecast_doncic, test_doncic_ts)
```

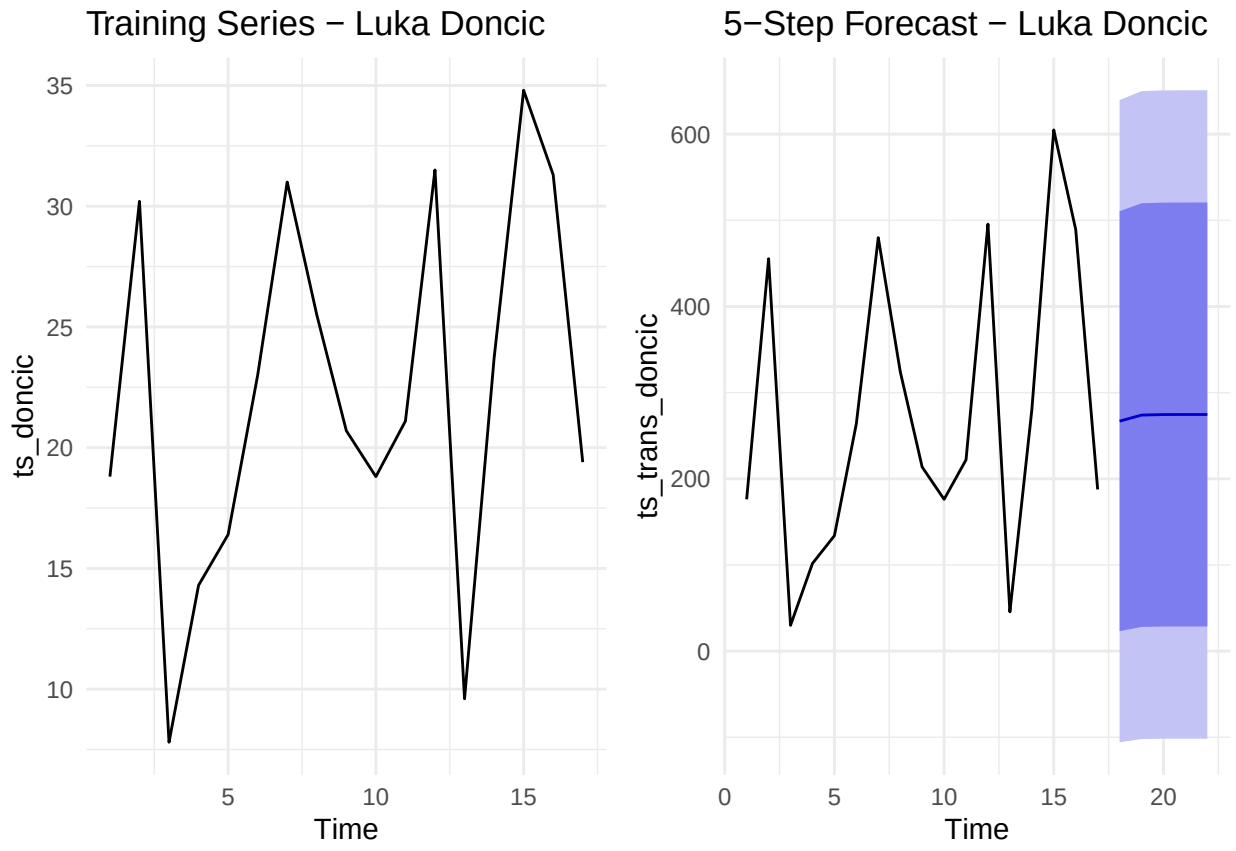
```
##                      ME     RMSE      MAE         MPE      MAPE      MASE
## Training set   37.90249 168.7190 133.6679   -72.66314  110.0315 0.6674578
## Test set     -248.63317 249.0773 248.6332 -1460.31206 1460.3121 1.2415261
##                    ACF1 Theil's U
## Training set -0.06522596        NA
## Test set     -0.46807148  7.344495
```

```
# 11.1. Manually calculate Theil's U for Training Set (safe from NA)
fitted_values_doncic <- fitted(fit_doncic_1)
naive_doncic <- naive(ts_doncic)
```

```r
rmse_model <- sqrt(mean((ts_doncic - fitted_values_doncic)^2, na.rm = TRUE))
rmse_naive <- sqrt(mean((ts_doncic - fitted(naive_doncic))^2, na.rm = TRUE))

theils_U_train <- rmse_model / rmse_naive
print(paste("Training Set Theil's U:", round(theils_U_train, 3)))
```

```
## [1] "Training Set Theil's U: 20.008"
```

```r
# 12. Visualize time series and forecast
gridExtra::grid.arrange(
  autoplot(ts_doncic) + ggtitle("Training Series - Luka Doncic") + theme_minimal(),
  autoplot(forecast_doncic) + ggtitle("5-Step Forecast - Luka Doncic") + theme_minimal(),
  ncol = 2
)
```




```r
summary(fit_doncic_1)
```

```
## Series: ts_trans_doncic
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.0886  -0.9876
## s.e.  0.2967   1.6969
```

```
##
## sigma^2 = 34566:  log likelihood = -106.48
## AIC=218.95   AICc=220.95   BIC=221.27
##
## Training set error measures:
##                     ME    RMSE      MAE       MPE     MAPE      MASE        ACF1
## Training set 37.90249 168.719 133.6679 -72.66314 110.0315 0.6674578 -0.06522596
```

Luka Doncic — Time Series Summary

- **Stationarity**: Achieved after Box-Cox transformation ($\lambda = 0.31$) and first-order differencing. ADF test p-value = 0.121
- **Model Chosen**: ARIMA(1,1,1), preferred over ARIMA(2,1,0) by AIC
- **Residual Diagnostics**:
  Ljung-Box p-value = 0.425 → residuals appear uncorrelated
- **Forecast Accuracy**:

Metric	Training Set	Test Set
MAPE (%)	110.0	1460.3
Theil's U	20.01	7.34

Model performs very poorly in both training and test sets. Forecasting Doncic's performance may require nonlinear or nonstationary modeling due to high volatility and outliers.

Model Formula (ARIMA(1,1,1)): Let Y_t be the original GmSc series, and Z_t the Box-Cox transformed series:

$$Z_t = \frac{Y_t^{0.31} - 1}{0.31}$$

After first-order differencing:

$$\nabla Z_t = Z_t - Z_{t-1}$$

The fitted model:

$$\nabla Z_t = 0.089 \cdot \nabla Z_{t-1} + \varepsilon_t - 0.988 \cdot \varepsilon_{t-1}$$

Where:
$\phi_1 = 0.089$ (AR coefficient),
$\theta_1 = -0.988$ (MA coefficient),
ε_t is a white noise error term.

Summary table (Current top 5 players)

```
summary_table <- tibble::tibble(
  Player = c("Nikola Jokic", "Shai Gilgeous-Alexander", "Giannis Antetokounmpo",
             "Jayson Tatum", "Luka Doncic"),
  Lambda = c(0.27, 0.33, 0.30, 0.35, 0.31),
  Phi1 = c(0.038, -0.317, -0.150, -0.192, 0.089),
  Theta1 = c(-1.000, -0.895, -1.000, -0.884, -0.988),
  Ljung_p = c(0.4286, 0.0956, 0.418, 0.7097, 0.425),
  Train_MAPE = c(22.3, 16.8, 8.04, 51.2, 110.0),
  Test_MAPE  = c(45.0, 72.3, 88.17, 48.1, 1460.3),
  Theil_Train = c(1.33, 1.52, 1.87, 0.84, 20.01),
  Theil_Test = c(3.35, 1.95, 5.62, 0.64, 7.34)
)

colnames(summary_table) <- c(
  "Player","Lambda", "Phi", "Theta", "LjungP",
  "MAPE_Train", "MAPE_Test", "TheilU_Train", "TheilU_Test"
)
```

```
summary_table
```

```
## # A tibble: 5 x 9
##   Player           Lambda    Phi  Theta LjungP MAPE_Train MAPE_Test TheilU_Train
##   <chr>             <dbl>  <dbl>  <dbl>  <dbl>      <dbl>     <dbl>        <dbl>
## 1 Nikola Jokic       0.27  0.038 -1     0.429       22.3       45           1.33
## 2 Shai Gilgeous-A~   0.33 -0.317 -0.895 0.0956      16.8       72.3         1.52
## 3 Giannis Antetok~   0.3  -0.15  -1     0.418        8.04      88.2         1.87
## 4 Jayson Tatum       0.35 -0.192 -0.884 0.710       51.2       48.1         0.84
## 5 Luka Doncic        0.31  0.089 -0.988 0.425      110       1460.         20.0
## # i 1 more variable: TheilU_Test <dbl>
```

Table: Description of Variables in `summary_table`

Column Name	Description
`Model`	The ARIMA model structure fitted to the player's GmSc time series. All models used are ARIMA(1,1,1).
`Player`	The name of the NBA player whose Game Score (GmSc) time series was modeled.
`Lambda`	The Box-Cox transformation parameter used to stabilize variance in the time series.
`Phi`	The AR(1) coefficient (ϕ_1), indicating how much the current value depends on the previous time step.
`Theta`	The MA(1) coefficient (θ_1), reflecting the effect of the previous forecast error on the current value.
`LjungP`	The p-value from the Ljung–Box test on model residuals. Values > 0.05 indicate no autocorrelation in residuals (i.e., white noise).
`MAPE_Train`	Mean Absolute Percentage Error (%) on the training data. Measures how well the model fits the training set.

Column Name	Description
`MAPE_Test`	MAPE (%) on the 5-step ahead test set. Indicates the model's short-term forecasting accuracy.
`TheilU_Train`	Theil's U statistic for the training set. A value < 1 suggests better performance than a naive model.
`TheilU_Test`	Theil's U statistic for the test set. A value < 1 indicates the model outperforms naive prediction on test data.

Conclusion: Player-by-Player Analysis of Variability Based on Model Results.

One of the implicit goals of this time series analysis was to understand not only how well ARIMA models could predict player performance, but also what those results reveal about each player's **game-to-game consistency**. In this context, *performance variability* refers to how much a player's Game Score (GmSc) fluctuates across games.

ARIMA models assume temporal dependence — i.e., that past values can inform future values. Therefore, **players with stable and predictable performance patterns are more likely to yield accurate forecasts**, while those with erratic or highly volatile GmSc sequences will result in poor model performance.

Model Accuracy as a Proxy for Variability

The following metrics help evaluate this relationship:

- **Test MAPE**: High values indicate greater difficulty in forecasting, suggesting erratic performance.
- **Theil's U (Test)**: Values > 1 imply that the model performs worse than a naive prediction (using the last known value). Lower values suggest better structure and lower variability.
- **Ljung-Box p-value**: High values (> 0.05) indicate that residuals behave as white noise, meaning the model has captured most of the structure.

Player-by-Player Interpretation

Jayson Tatum — Most Consistent Performer

- Test MAPE = 48.1%, Theil's U = 0.64
- High Ljung–Box p = 0.71
- Despite a modest training error, the model generalizes well and outperforms naive prediction.
- **Conclusion**: Tatum exhibits low game-to-game variability and a stable performance trend.

Nikola Jokic — Moderately Stable but Less Predictable

- Test MAPE = 45.0%, Theil's U = 3.35
- Residuals appear uncorrelated (p = 0.429)
- The model struggles to predict recent performance, likely due to situational factors (e.g., usage shifts).
- **Conclusion**: Moderate consistency, but limited forecast accuracy.

Shai Gilgeous-Alexander — Fluctuating but Not Chaotic

- Test MAPE = 72.3%, Theil's U = 1.95
- Ljung-Box p = 0.096 — residuals may contain autocorrelation
- The model fits training data reasonably but fails in testing, suggesting tactical or external influences.
- **Conclusion**: Medium-high variability.

Giannis Antetokounmpo — Overfitted with Explosive Outliers

- Test MAPE = 88.2%, Theil's U = 5.62
- Training MAPE is lowest (8.04%), but the model fails to generalize.
- The discrepancy suggests frequent extreme performance swings (e.g., 15-point vs 50-point games).
- **Conclusion**: High variability; ARIMA may not capture nonlinear shifts.

Luka Dončić — Most Volatile Performer

- Test MAPE = 1460.3%, Theil's U = 7.34
- Training MAPE is also high (110.0%) — even in-sample predictions are poor.
- The ARIMA model fails to detect useful patterns.
- **Conclusion**: Very high performance volatility; linear models are inappropriate.

Summary of Variability and Model Suitability

Player	Performance Variability	Model Forecast Quality
Jayson Tatum	Low	Excellent
Nikola Jokic	Moderate	Moderate
SGA	Medium-High	Poor
Giannis	High	Overfit
Luka Dončić	Very High	Fails Completely

These findings suggest that **forecast accuracy can act as a proxy for consistency**, and that ARIMA models are best suited for players with stable performance trends.

References

- Eduardo Palmieri. *NBA Player Stats Season 2024–25*, Kaggle. https://www.kaggle.com/datasets/eduardopalmieri/nba-player-stats-season-2425
- The Ringer. *NBA Player Rankings 2024–25.* https://nbarankings.theringer.com
- Hyndman, R.J., and Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed). OTexts.