Time Series Analysis of New Haven Temperature Data

Latherial Calbert

April 25, 2025

1 Introduction

1.1 Motivation and Goal

I wanted to analyze temperature data to understand climate patterns and see if there are any changes over time. For this project, I'm looking at historical temperature data from New Haven, Connecticut, to find patterns, build models, and predict future temperatures. Using time series methods, I can extract insights from how temperatures have changed over the years.

My main goals for this project are:

- Analyze the time series patterns in New Haven temperature data
- Make the data stationary for ARIMA modeling
- Find good ARIMA models through testing
- Compare models and pick the best ones
- Check how accurate the forecasts are
- Predict temperatures for future years

Understanding temperature patterns is useful for climate research, urban planning, energy use forecasting, and farming. This analysis helps understand temperature history in New Haven and forecast future temperature as well.

2 Dataset Description

The dataset used in this analysis is "nhtemp," which contains the mean annual temperatures in degrees Fahrenheit recorded in New Haven, Connecticut, from 1912 to 1971. This dataset is included in the built-in datasets of R, in the library(TSA). The data consists of 60 yearly observations covering six decades of temperature measurements.

The following R code loads the required libraries and the dataset:

```r
library(tseries)
library(TSA)

# Load the New Haven temperature data
data("nhtemp")
```

Listing 1: Loading the libraries and dataset

3 Data Analysis

3.1 Exploratory Data Analysis

We begin by visualizing the time series to understand its basic characteristics and identify any patterns, trends, or seasonality.

```r
# Plot the time series
plot(nhtemp)

# Check for trends
# Plot ACF and PACF of the original series
par(mfrow=c(1,2))
acf(nhtemp, main="ACF of Original Series")
pacf(nhtemp, main="PACF of Original Series")
```

Listing 2: Exploratory data visualization

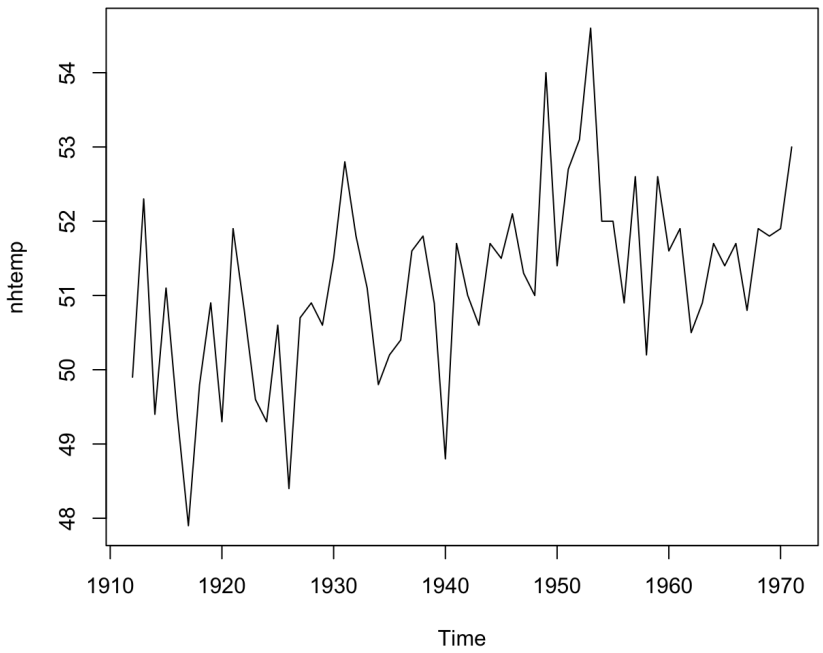

Figure 1: Original time series plot of New Haven temperatures (1912-1971)

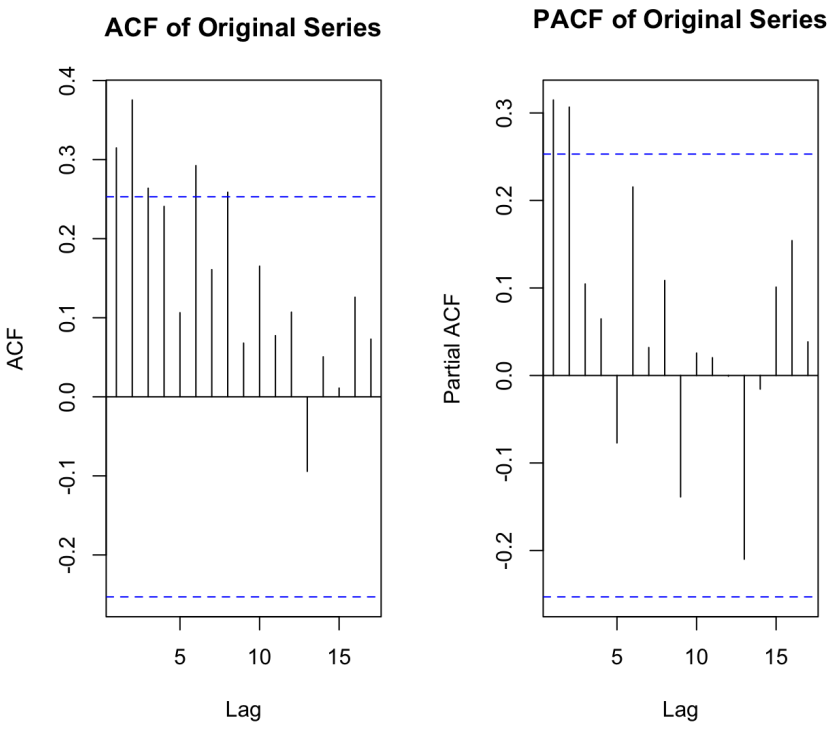

Figure 2: ACF and PACF of original temperature series

Interpretation of Exploratory Plots: Looking at the original time series plot, I can see the yearly temperatures in New Haven from 1912 to 1971. There seems to be a slight upward trend, but it's not very strong. The temperatures bounce around between about 48°F and 55°F. Since these are yearly averages, there's no seasonal patterns to worry about. There are some high and low points throughout the series, but nothing that looks like an extreme outlier. It also looks like the plot displays non-constant variance, suggesting non-stationarity.

Looking at the ACF of the original series, I see significant autocorrelation at several lags with a slow decay pattern, which is typical for non-stationary series.

3.2 Making the Series Stationary

Before I can apply ARIMA modeling, I need to do further checks to see if the series is stationary and make it stationary if it's not. I'll use the Augmented Dickey-Fuller (ADF) test for this.

```
# Test for stationarity using ADF test
adf_test <- adf.test(nhtemp)
print(adf_test)

# Apply first-order differencing
nhtemp_diff <- diff(nhtemp)

# Plot the differenced series
plot(nhtemp_diff)
```

```r

# Check stationarity of the differenced series
adf_test_diff <- adf.test(nhtemp_diff)
print(adf_test_diff)

# Plot ACF and PACF of the differenced series
par(mfrow=c(1,2))
acf(nhtemp_diff, main="ACF of Differenced Series")
pacf(nhtemp_diff, main="PACF of Differenced Series")
```

Listing 3: Testing for stationarity

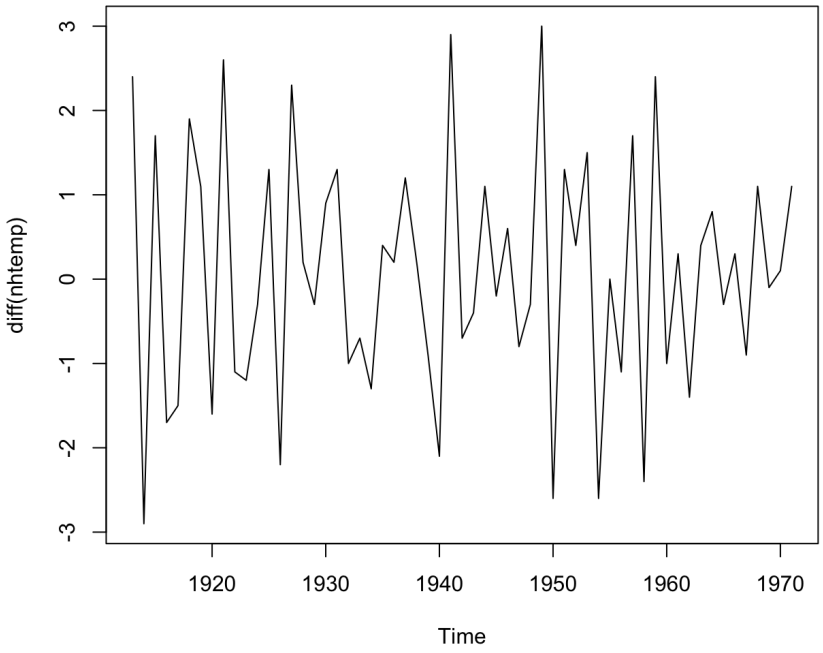

Figure 3: Differenced temperature series

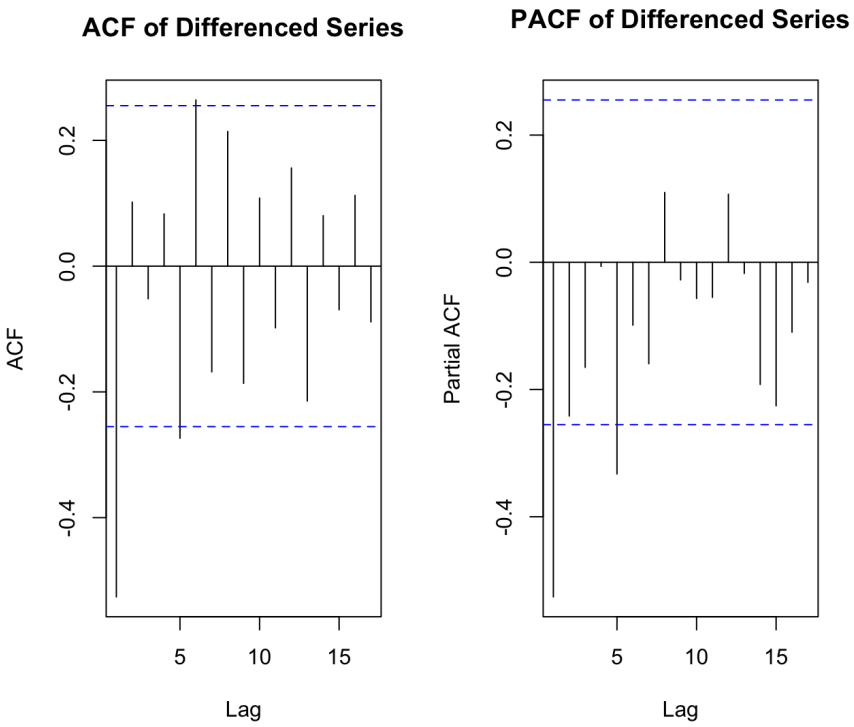

Figure 4: ACF and PACF of differenced temperature series

Interpretation of Stationarity Tests: The ADF test on the original temperature series gives a p-value of 0.08376, which is just above the 0.05 significance level. This suggests the series isn't completely stationary but it's close.

After applying first differencing, the ADF test gives a p-value of 0.01, which is below 0.05. This tells me that the differenced series is now stationary, so taking the first difference was enough to fix the non-stationarity problem.

For the differenced series, the ACF shows significant negative autocorrelation at lag 1, which suggests a possible MA(1) component. The PACF of the differenced series shows a significant spike at lag 1, suggesting a possible AR(1) component.

3.3 Model Specification

Based on the ACF and PACF patterns of the differenced series, I can identify several candidate models:

1. ARIMA(0,1,1): The significant negative spike at lag 1 in the ACF suggests an MA(1) model with differencing.

2. ARIMA(1,1,0): The significant negative spike at lag 1 in the PACF suggests an AR(1) model with differencing.

3. ARIMA(1,1,1): Including both AR(1) and MA(1) components with differencing might capture the pattern better.

4. ARIMA(2,1,0): Adding an AR(2) component might help if there are higher-order effects.

5. ARIMA(0,1,2): Including an MA(2) component might capture additional moving average effects.

I'll fit these models and compare them using diagnostics to find the best ones.

3.4 Model Estimation

Now I'll fit the ARIMA models to our data:

```r
# Fit ARIMA(0,1,1) - MA(1) with differencing
model_011 <- arima(nhtemp, order=c(0,1,1))
print(model_011)

# Fit ARIMA(1,1,0) - AR(1) with differencing
model_110 <- arima(nhtemp, order=c(1,1,0))
print(model_110)

# Fit ARIMA(1,1,1) - ARMA(1,1) with differencing
model_111 <- arima(nhtemp, order=c(1,1,1))
print(model_111)

# Fit ARIMA(2,1,0) - AR(2) with differencing
model_210 <- arima(nhtemp, order=c(2,1,0))
print(model_210)

# Fit ARIMA(0,1,2) - MA(2) with differencing
model_012 <- arima(nhtemp, order=c(0,1,2))
print(model_012)
```

Listing 4: Model estimation

Table 1: Comparison of ARIMA Model Coefficients with Standard Errors and AIC Values

Model	**AR(1)**	**AR(2)**	**MA(1)**	**MA(2)**	**AIC**
ARIMA(0,1,1)	—	—	-0.7983† (0.0956)	—	185.52
ARIMA(1,1,0)	-0.5467† (0.1109)	—	—	—	193.46
ARIMA(1,1,1)	0.0073 (0.1802)	—	-0.8019† (0.1285)	—	187.52
ARIMA(2,1,0)	-0.6684† (0.1269)	-0.2392 (0.1301)	—	—	192.19
ARIMA(0,1,2)	—	—	-0.7956† (0.1224)	-0.0042 (0.1221)	187.52

† Statistically significant coefficient (coefficient is more than twice its standard error)

Interpretation of Model Estimation: The ARIMA(0,1,1) model has the lowest

AIC value (185.52), balancing model complexity and fit. Looking at coefficient significance (marked with † in the table), we can see:

1. The ARIMA(0,1,1) model has a significant MA(1) term (-0.7983).

2. The ARIMA(1,1,1) model has a significant MA(1) term (-0.8019), but its AR(1) term (0.0073) is not significant, suggesting this parameter is unnecessary.

3. Similarly, the ARIMA(0,1,2) model has a significant MA(1) term (-0.7956), but its MA(2) term (-0.0042) is not significant.

4. These findings clearly favor the ARIMA(0,1,1) model, as it achieves the best fit (lowest AIC) with the fewest parameters. The more complex models (ARIMA(1,1,1) and ARIMA(0,1,2)) have non-significant parameters that add complexity without improving the fit drastically. Following the principle of parsimony, the ARIMA(0,1,1) model seems to be the best choice for further analysis.

3.5 Model Diagnostics

Next, I need to check if the residuals from my models behave like white noise and if the residuals are normal. I'll perform several diagnostic tests:

```r
# Model diagnostics for ARIMA(0,1,1)
par(mfrow=c(2,2))
residuals_011 <- residuals(model_011)
plot(residuals_011, main="Residuals - ARIMA(0,1,1)")
acf(residuals_011, main="ACF of Residuals - ARIMA(0,1,1)")
pacf(residuals_011, main="PACF of Residuals - ARIMA(0,1,1)")
qqnorm(residuals_011, main="Normal Q-Q Plot - ARIMA(0,1,1)")
qqline(residuals_011)
LB_test_011 <- LB.test(model_011, lag = 10)
shapiro_test_011 <- shapiro.test(residuals_011)
runs_test_011 <- runs(residuals_011)
print("ARIMA(0,1,1) diagnostics:")
print(LB_test_011)
print(shapiro_test_011)
print(runs_test_011)

# Model diagnostics for ARIMA(1,1,0)
par(mfrow=c(2,2))
residuals_110 <- residuals(model_110)
plot(residuals_110, main="Residuals - ARIMA(1,1,0)")
acf(residuals_110, main="ACF of Residuals - ARIMA(1,1,0)")
pacf(residuals_110, main="PACF of Residuals - ARIMA(1,1,0)")
qqnorm(residuals_110, main="Normal Q-Q Plot - ARIMA(1,1,0)")
qqline(residuals_110)
LB_test_110 <- LB.test(model_110, lag = 10)
shapiro_test_110 <- shapiro.test(residuals_110)
runs_test_110 <- runs(residuals_110)
print("ARIMA(1,1,0) diagnostics:")
print(LB_test_110)
```

```r
print(shapiro_test_110)
print(runs_test_110)

# Model diagnostics for ARIMA(1,1,1)
par(mfrow=c(2,2))
residuals_111 <- residuals(model_111)
plot(residuals_111, main="Residuals - ARIMA(1,1,1)")
acf(residuals_111, main="ACF of Residuals - ARIMA(1,1,1)")
pacf(residuals_111, main="PACF of Residuals - ARIMA(1,1,1)")
qqnorm(residuals_111, main="Normal Q-Q Plot - ARIMA(1,1,1)")
qqline(residuals_111)
LB_test_111 <- LB.test(model_111, lag = 10)
shapiro_test_111 <- shapiro.test(residuals_111)
runs_test_111 <- runs(residuals_111)
print("ARIMA(1,1,1) diagnostics:")
print(LB_test_111)
print(shapiro_test_111)
print(runs_test_111)

# Model diagnostics for ARIMA(2,1,0)
par(mfrow=c(2,2))
residuals_210 <- residuals(model_210)
plot(residuals_210, main="Residuals - ARIMA(2,1,0)")
acf(residuals_210, main="ACF of Residuals - ARIMA(2,1,0)")
pacf(residuals_210, main="PACF of Residuals - ARIMA(2,1,0)")
qqnorm(residuals_210, main="Normal Q-Q Plot - ARIMA(2,1,0)")
qqline(residuals_210)
LB_test_210 <- LB.test(model_210, lag = 10)
shapiro_test_210 <- shapiro.test(residuals_210)
runs_test_210 <- runs(residuals_210)
print("ARIMA(2,1,0) diagnostics:")
print(LB_test_210)
print(shapiro_test_210)
print(runs_test_210)

# Model diagnostics for ARIMA(0,1,2)
par(mfrow=c(2,2))
residuals_012 <- residuals(model_012)
plot(residuals_012, main="Residuals - ARIMA(0,1,2)")
acf(residuals_012, main="ACF of Residuals - ARIMA(0,1,2)")
pacf(residuals_012, main="PACF of Residuals - ARIMA(0,1,2)")
qqnorm(residuals_012, main="Normal Q-Q Plot - ARIMA(0,1,2)")
qqline(residuals_012)
LB_test_012 <- LB.test(model_012, lag = 10)
shapiro_test_012 <- shapiro.test(residuals_012)
runs_test_012 <- runs(residuals_012)
print("ARIMA(0,1,2) diagnostics:")
```

```
print(LB_test_012)
print(shapiro_test_012)
print(runs_test_012)
```

Listing 5: Performing many diagnostic checks

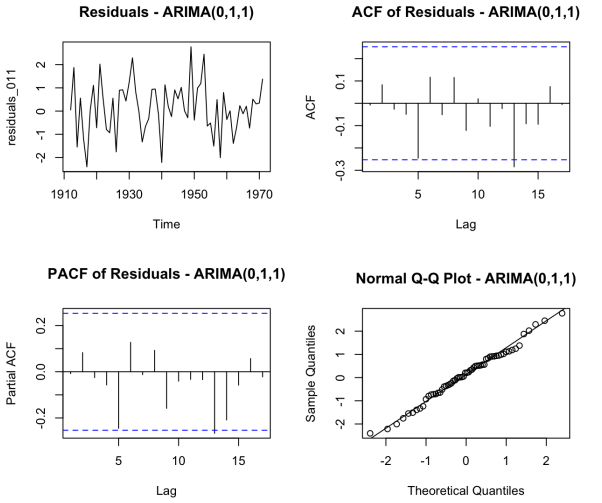

Figure 5: Diagnostic plots for ARIMA(0,1,1) model

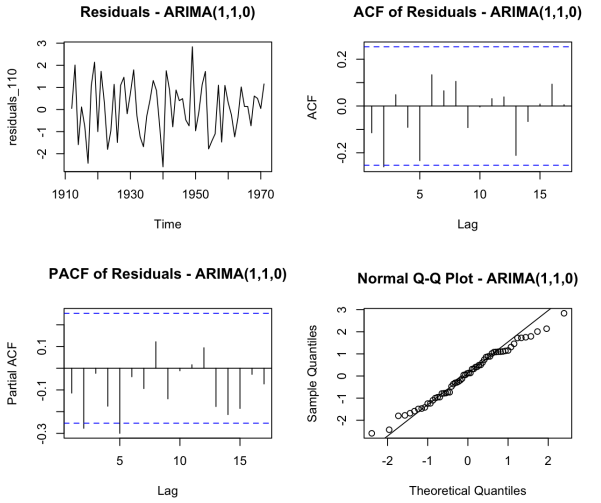

Figure 6: Diagnostic plots for ARIMA(1,1,0) model

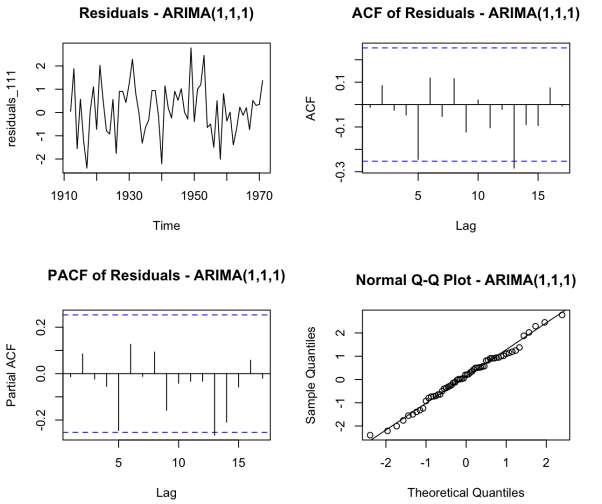

Figure 7: Diagnostic plots for ARIMA(1,1,1) model

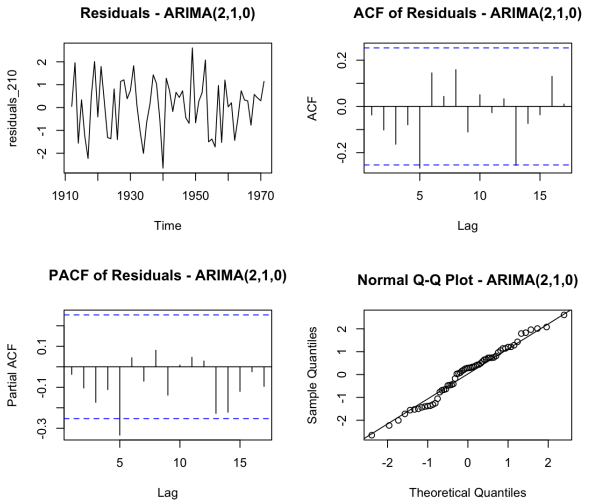

Figure 8: Diagnostic plots for ARIMA(2,1,0) model

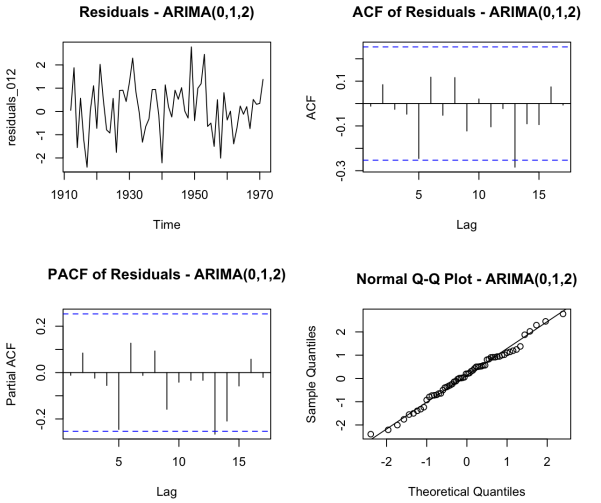

Figure 9: Diagnostic plots for ARIMA(0,1,2) model

Table 2: Residual Diagnostic Test Results for All Models

Test	(0,1,1)	(1,1,0)	(1,1,1)	(2,1,0)	(0,1,2)
Ljung-Box	0.551	0.196	0.444	0.149	0.445
Shapiro-Wilk	0.820	0.717	0.825	0.412	0.824
Runs Test	0.535	0.958	0.535	0.971	0.535

Interpretation of Diagnostic Tests: The table shows that all models pass the diagnostic tests with p-values above 0.05:

1. **Ljung-Box Test:** All p-values are above 0.05, which means there's no significant autocorrelation left in the residuals. The ARIMA(0,1,1), ARIMA(1,1,1), and ARIMA(0,1,2) models have higher p-values (0.551, 0.444, and 0.445), suggesting they've captured the autocorrelation pattern better.

2. **Shapiro-Wilk Test:** All p-values are above 0.05, confirming the residuals are normally distributed. The ARIMA(0,1,1), ARIMA(1,1,1), and ARIMA(0,1,2) models have higher p-values here too.

3. **Runs Test:** All p-values are well above 0.05, showing there's no pattern in the residuals. Interestingly, the ARIMA(1,1,0) and ARIMA(2,1,0) models have higher p-values for this test.

The diagnostic plots confirm these findings. The residuals look random, and the ACF and PACF of residuals don't show any significant spikes, meaning the models have captured the time series patterns well.

Based on the AIC values and diagnostic tests, I'll focus on ARIMA(0,1,1), ARIMA(1,1,1), and ARIMA(0,1,2) as my best models.

3.6 Forecast Accuracy Comparison – In Sample Predictions

Now I'll compare how well the top three models predict the in-sample data using different accuracy metrics:

```
# Define functions for calculating forecast errors
# Mean Square Relative Error (MSRE) - only squares the numerator
msre <- function(actual, predicted) {
  return(mean((actual - predicted)^2/actual, na.rm=TRUE))
}

# Mean Absolute Percentage Error (MAPE)
mape <- function(actual, predicted) {
  return(mean(abs((actual - predicted)/actual), na.rm=TRUE) *
   100)
}

# Get fitted values for the top three models
fitted_011 <- nhtemp - residuals_011
fitted_012 <- nhtemp - residuals_012
fitted_111 <- nhtemp - residuals_111

```

```r
# Calculate MSRE and MAPE for each model
msre_011 <- msre(nhtemp, fitted_011)
mape_011 <- mape(nhtemp, fitted_011)

msre_012 <- msre(nhtemp, fitted_012)
mape_012 <- mape(nhtemp, fitted_012)

msre_111 <- msre(nhtemp, fitted_111)
mape_111 <- mape(nhtemp, fitted_111)

print(msre_011)
print(mape_011)
print(msre_012)
print(mape_012)
print(msre_111)
print(mape_111)
```

Listing 6: Forecast accuracy comparison

Table 3: Forecast Accuracy Metrics for Top Three Models

Model	MSRE	MAPE (%)
ARIMA(0,1,1)	0.02477252	1.749865
ARIMA(0,1,2)	0.02476881	1.749420
ARIMA(1,1,1)	0.02476739	1.749255

Interpretation of Forecast Accuracy: Looking at the error metrics in the table, all three models have very similar performance. The ARIMA(1,1,1) model has slightly lower MSRE and MAPE values, but the differences are tiny (the MSRE differences are in the 5th decimal place and MAPE differences are in the 4th decimal place). These differences are so small that we should consider the most simple model at this point.

Since the ARIMA(0,1,1) model has the lowest AIC value (185.52 vs 187.52 for the others) and is simpler with fewer parameters, I'll go with it as my best model. In time series analysis, we often prefer simpler models when the performance is similar (this is the principle of parsimony).

3.7 Forecasting Future Values

Finally, I'll use the ARIMA(0,1,1) model to forecast temperatures for the next six years:

```r
# Plot the forecast
plot(model_011, n.ahead=6, type="b", main="ARIMA(0,1,1) Forecast"
    , ylab="Temperature ( F )")

result = plot(model_011,n.ahead=6,type='b')

# Get the forecasted values
forecast <- result$pred
```

```r
print(forecast)

# Create prediction intervals
pred_int=cbind(result$upi, result$lpi)
pred_int
```

Listing 7: Forecasting future values

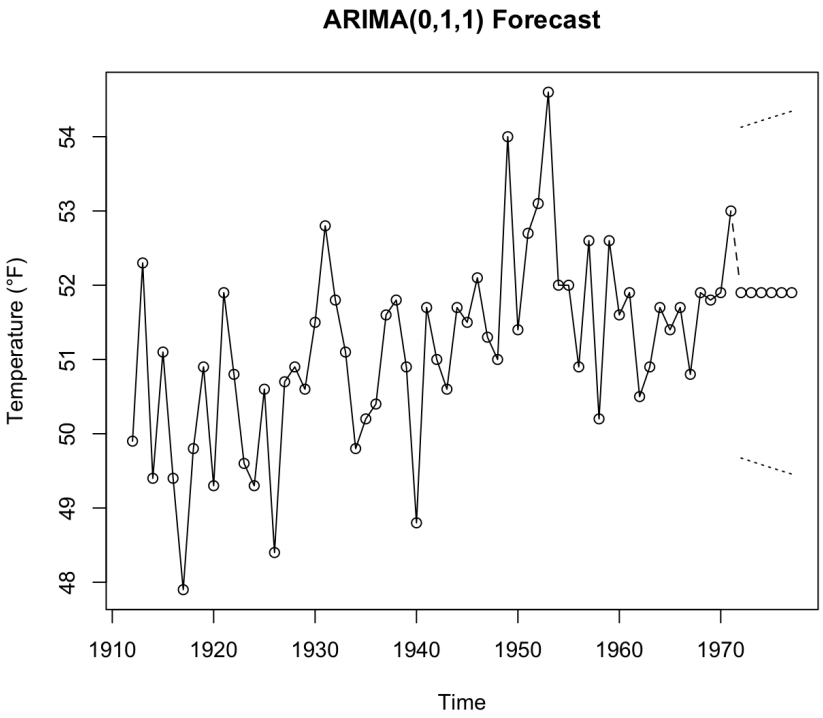

Figure 10: Temperature forecast for 1972–1977 using ARIMA(0,1,1) model

Table 4: Forecast Values and 95% Prediction Intervals

Year	Forecast (°F)	Lower PI (°F)	Upper PI (°F)
1972	51.90008	49.67310	54.12705
1973	51.90008	49.62824	54.17191
1974	51.90008	49.58425	54.21591
1975	51.90008	49.54107	54.25908
1976	51.90008	49.49868	54.30148
1977	51.90008	49.45702	54.34314

Interpretation of Forecasts: The forecast shows a constant temperature of 51.90°F for all years from 1972 to 1977. This makes sense because ARIMA models with differencing tend to give constant forecasts when the series is stationary.

The prediction intervals get wider as we go further into the future, which shows increasing uncertainty. For 1972, the 95% interval is about 49.67°F to 54.13°F, and by 1977, it widens to about 49.46°F to 54.34°F. This is normal because it's harder to predict accurately further

into the future.

Based on the historical data, we can expect New Haven's temperatures to stay around 51.90°F in the future, with natural variations within these prediction intervals.

4 Concluding Remarks

In this project, I analyzed temperature data from New Haven, Connecticut from 1912 to 1971 using the methodology from our course materials and textbook. This approach involves making the series stationary, identifying possible models, estimating parameters, and selecting the best model through diagnostic checking.

4.1 Summary of Findings

I found that the New Haven temperature series had slight non-stationarity, which I fixed with first differencing. After comparing different ARIMA models, I chose the ARIMA(0,1,1) model as the best one based on its low AIC value, simplicity, and good diagnostic test results. While the ARIMA(1,1,1) and ARIMA(0,1,2) models had slightly better forecast accuracy, the differences were so small that they didn't justify the extra complexity.

The ARIMA(0,1,1) model passed all diagnostic tests, with residuals that looked like white noise and showed no significant patterns. This model forecasts a stable temperature of about 51.90°F for 1972-1977, with prediction intervals widening over time due to increasing uncertainty.

The equation for the ARIMA(0,1,1) model is:

$$Y_t = Y_{t-1} + \varepsilon_t - 0.7983\varepsilon_{t-1} \tag{1}$$

where Y_t is the temperature at time t and ε_t is the error term.

4.2 Limitations and Future Work

This study has some limitations:

1. The dataset only includes yearly averages, so I couldn't analyze seasonal patterns that would show up in monthly or daily data.

2. The data ends in 1971, and climate patterns might have changed significantly since then.

3. I only looked at time patterns without considering other factors that might affect temperature, like urbanization or CO2 levels.

For future research, I could:

1. Include more recent temperature data to see if patterns have changed over time.
2. Use monthly or daily data to capture seasonal patterns.
3. Try multivariate models that include related climate variables.
4. Investigate climate change signals by comparing historical data with recent data.
5. Try more advanced models like seasonal ARIMA or GARCH models.

Despite these limitations, this analysis provides a good understanding of historical temperature patterns in New Haven and shows how to apply time series analysis to climate data.