Baseball

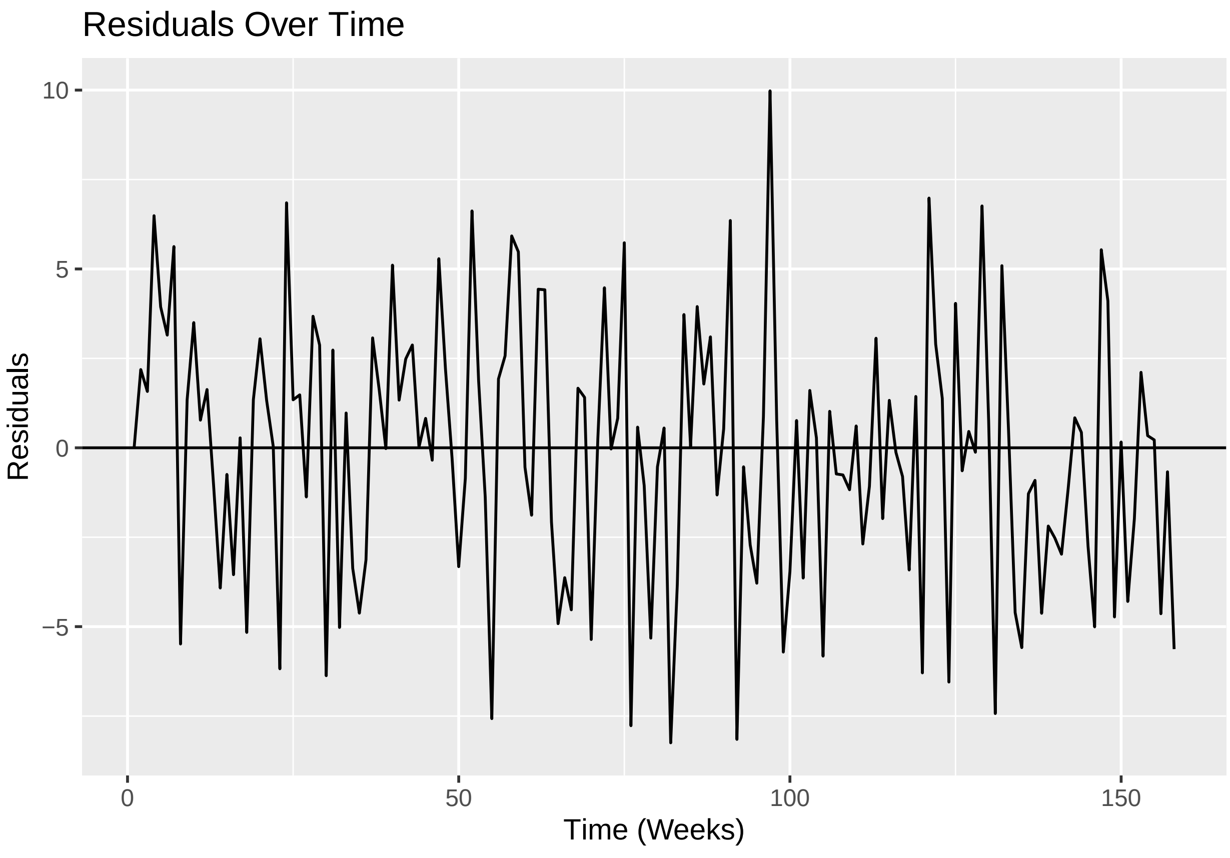
2025-04-11

```r
# Read in the data we processed/created in Python from the pybaseball library
weekly_data <- read.csv("weekly_data_for_r.csv")
```

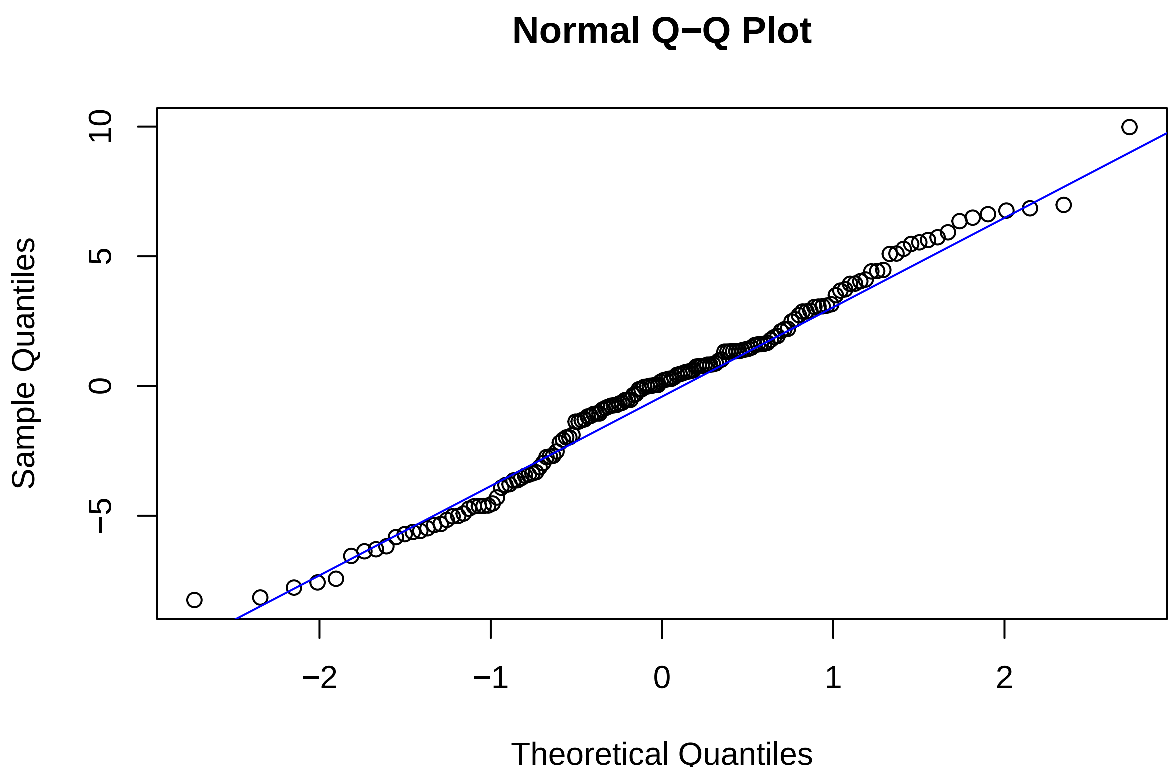
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Baseline ARIMA(1,1,1) AIC: 859.4756
```

```r
autoplot(residuals(model_baseline)) +
  ggtitle("Residuals Over Time") +
  xlab("Time (Weeks)") + ylab("Residuals") +
  geom_hline(yintercept = 0)
```




```r
# Q-Q plot
qqnorm(residuals(model_baseline))
qqline(residuals(model_baseline), col = "blue")
```




```r
# With exog featueres
exog_vars <- c('avg_velocity', 'avg_release_pos_x', 'avg_spin_rate', 'avg_pitch_number',
    'avg_release_extension', 'rest_days', 'zone_rate', 'arm_angle', 'api_break_x_arm')

df_all <- na.omit(weekly_data[, c("K_per_9", exog_vars)])
y_all <- df_all$K_per_9
X_all <- as.matrix(df_all[, exog_vars])

model_exog <- Arima(y_all, order = c(1, 1, 1), xreg = X_all)
cat("SARIMAX AIC:", AIC(model_exog), "\n")
```

```
## SARIMAX AIC: 425.0035
```

```r
summary(model_exog)
```

```
## Series: y_all
## Regression with ARIMA(1,1,1) errors
##
## Coefficients:
##           ar1      ma1  avg_velocity  avg_release_pos_x  avg_spin_rate
##       -0.1833  -1.0000       -0.3406            -1.8307         0.0098
## s.e.   0.1251   0.0383        0.4161             2.2136         0.0059
##       avg_pitch_number  avg_release_extension  rest_days  zone_rate  arm_angle
##                 2.7904                 2.0911     0.4478   -18.0557    -0.1614
## s.e.            2.0859                 3.7557     0.3991     7.8760     0.2349
##       api_break_x_arm
##                6.6355
## s.e.           3.8996
##
## sigma^2 = 10.96:  log likelihood = -200.5
## AIC=425   AICc=429.8   BIC=453.28
```

```
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.3628811 3.049163 2.504712 -12.33635 27.18087 0.6299807
##                     ACF1
## Training set -0.0169155
```

```r
exog_vars <- c("zone_rate", "api_break_x_arm", "avg_spin_rate", "rest_days")

weekly_model_data <- na.omit(weekly_data[, c("K_per_9", exog_vars)])

y <- weekly_model_data$K_per_9
X <- as.matrix(weekly_model_data[, exog_vars])

fit <- auto.arima(
  y,
  xreg = X,
  seasonal = FALSE,
  stepwise = TRUE,
  trace = TRUE
)
```

```
##
##  Regression with ARIMA(2,1,2) errors : Inf
##  Regression with ARIMA(0,1,0) errors : 820.553
##  Regression with ARIMA(1,1,0) errors : 777.2635
##  Regression with ARIMA(0,1,1) errors : Inf
##  Regression with ARIMA(0,1,0) errors : 818.3875
##  Regression with ARIMA(2,1,0) errors : 761.307
##  Regression with ARIMA(3,1,0) errors : 753.1874
##  Regression with ARIMA(4,1,0) errors : 745.7583
##  Regression with ARIMA(5,1,0) errors : 742.6189
##  Regression with ARIMA(5,1,1) errors : Inf
##  Regression with ARIMA(4,1,1) errors : Inf
##  Regression with ARIMA(5,1,0) errors : 740.5673
##  Regression with ARIMA(4,1,0) errors : 743.6621
##  Regression with ARIMA(5,1,1) errors : 731.0005
##  Regression with ARIMA(4,1,1) errors : 728.6424
##  Regression with ARIMA(3,1,1) errors : 726.4599
##  Regression with ARIMA(2,1,1) errors : 724.4144
##  Regression with ARIMA(1,1,1) errors : 722.5691
##  Regression with ARIMA(0,1,1) errors : 722.2775
##  Regression with ARIMA(0,1,2) errors : 722.3252
##  Regression with ARIMA(1,1,0) errors : 775.1145
##  Regression with ARIMA(1,1,2) errors : Inf
##
##  Best model: Regression with ARIMA(0,1,1) errors
```

```r
summary(fit)
```

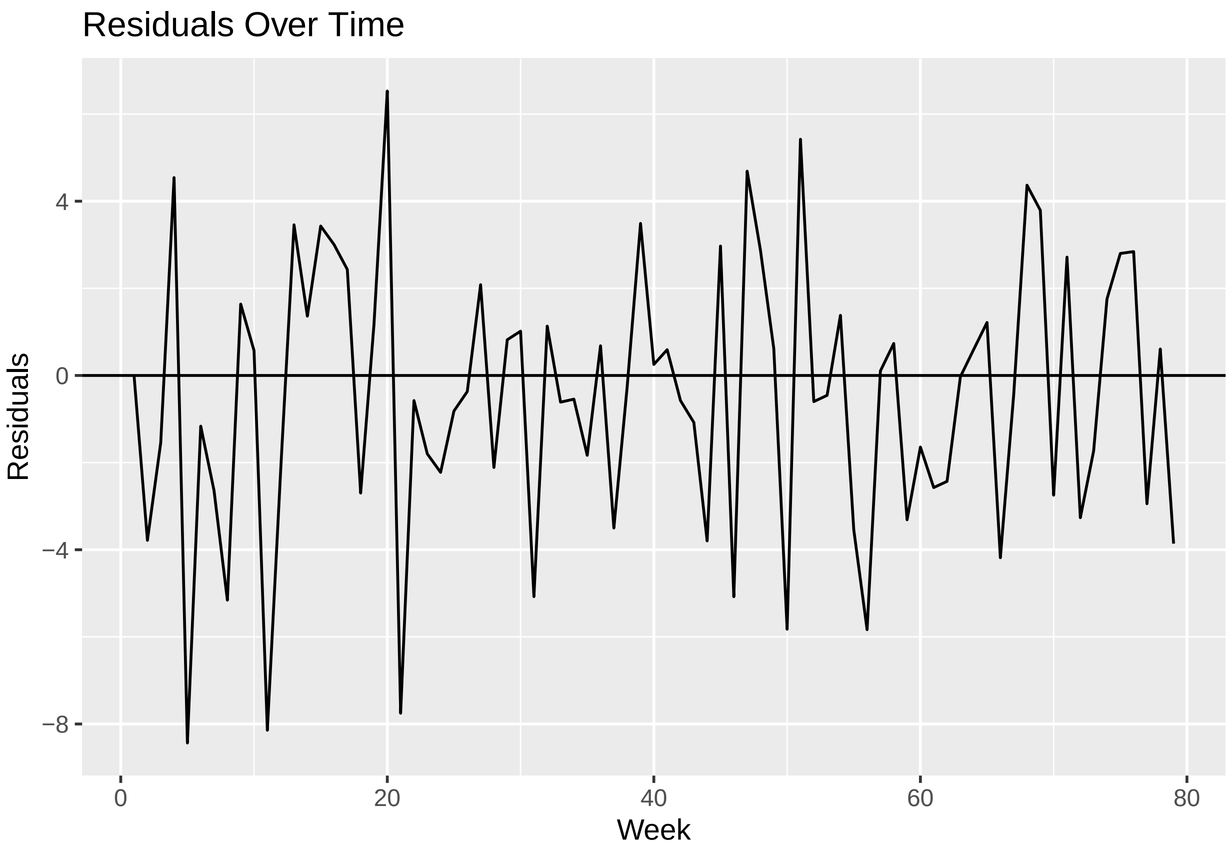
```
## Series: y
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##             ma1  zone_rate  api_break_x_arm  avg_spin_rate  rest_days
```

```
##       -0.9591    -8.4283          5.0044         0.0065     0.2621
## s.e.   0.0232     6.0320          2.7000         0.0046     0.2305
##
## sigma^2 = 11.9:  log likelihood = -354.81
## AIC=721.62   AICc=722.28   BIC=739
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.5120219 3.372798 2.629556 -14.67702 27.92929 0.6615612
##                    ACF1
## Training set -0.1414145
```

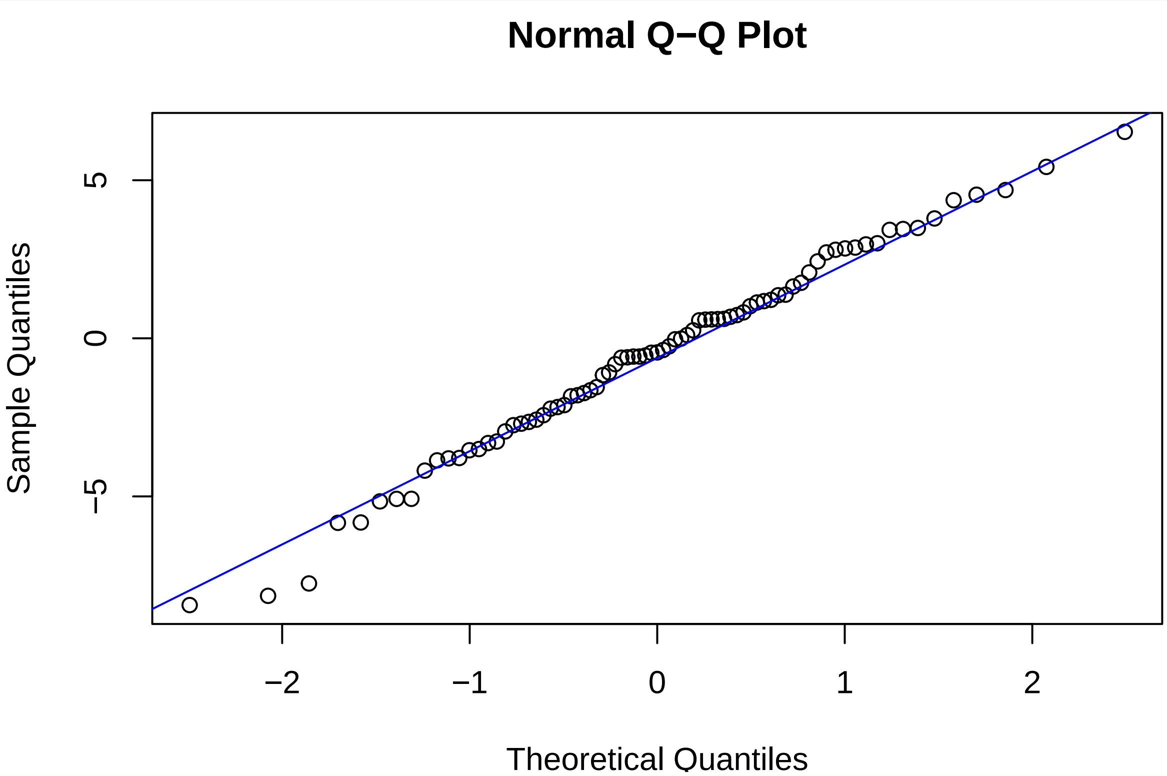
```r
# With the promising features and auto arima best fit

exog_vars_best = c("zone_rate","rest_days", "arm_angle", "avg_spin_rate", "api_break_x_arm")
df_sel <- na.omit(weekly_data[, c("K_per_9", exog_vars_best)])
y_sel <- df_sel$K_per_9
X_sel <- as.matrix(df_sel[, exog_vars_best])

model_sel <- Arima(y_sel, order = c(0,1,1), xreg = X_sel)
summary(model_sel)
```

```
## Series: y_sel
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##           ma1  zone_rate  rest_days  arm_angle  avg_spin_rate  api_break_x_arm
##       -1.0000   -16.9094     0.5164    -0.0916         0.0135           6.5184
## s.e.   0.0549     7.8696     0.3881     0.2406         0.0061           3.0591
##
## sigma^2 = 10.98:  log likelihood = -203.19
## AIC=420.38   AICc=421.98   BIC=436.87
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -0.5251106 3.163433 2.492181 -14.63179 28.05224 0.626829
##                    ACF1
## Training set -0.1562857
```

```r
autoplot(residuals(model_sel)) +
  ggtitle("Residuals Over Time") +
  xlab("Week") + ylab("Residuals") +
  geom_hline(yintercept = 0)
```




```r
# Q-Q plot
qqnorm(residuals(model_sel))
qqline(residuals(model_sel), col = "blue")
```



```r
pred <- fitted(model_sel)

comparison_df <- data.frame(
  Actual_K_per_9 = round(y_sel, 2),
  Predicted_K_per_9 = round(pred, 2)
)

print(head(comparison_df, 20))
```

```
##    Actual_K_per_9 Predicted_K_per_9
## 1           17.55             17.56
## 2           13.14             16.92
## 3           14.14             15.69
## 4           19.06             14.52
## 5            5.68             14.12
## 6           14.09             15.25
## 7           12.19             14.84
## 8            7.94             13.10
## 9           12.60             10.96
## 10          13.50             12.93
## 11           4.70             12.84
## 12           9.00             11.17
## 13          16.20             12.74
## 14          12.46             11.10
## 15          16.50             13.07
## 16          14.40             11.39
## 17          15.88             13.45
## 18          11.57             14.27
## 19          13.50             12.32
## 20          19.29             12.76
```

```r
sd(y_sel)
```

```
## [1] 3.471186
```

```r
sd(pred)
```

```
## [1] 1.87133
```

```r
library("Metrics")
```

```
##
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':
##
##     accuracy
```

```r
exog_vars_best = c("zone_rate", "api_break_x_arm", "avg_pitch_number", "rest_days")
df <- na.omit(weekly_data[, c("K_per_9", exog_vars)])

n <- nrow(df)
train_size <- floor(0.8 * n)
train <- df[1:train_size, ]
test <- df[(train_size + 1):n, ]

y_train <- train$K_per_9
```

```r
xreg_train <- as.matrix(train[, exog_vars])

xreg_test <- as.matrix(test[, exog_vars])
y_test <- test$K_per_9

model <- Arima(y_train, order = c(0, 1, 1), xreg = xreg_train)

forecast_test <- forecast(model, xreg = xreg_test, h = nrow(test))

cat("Train AIC:", AIC(model), "\n")
```

```
## Train AIC: 585.3886
```

```r
pred_test <- forecast_test$mean
rmse_value <- rmse(y_test, pred_test)
mae_value <- mae(y_test, pred_test)

cat("Test RMSE:", round(rmse_value, 2), "\n")
```

```
## Test RMSE: 3.42
```

```r
cat("Test MAE:", round(mae_value, 2), "\n")
```

```
## Test MAE: 2.77
```

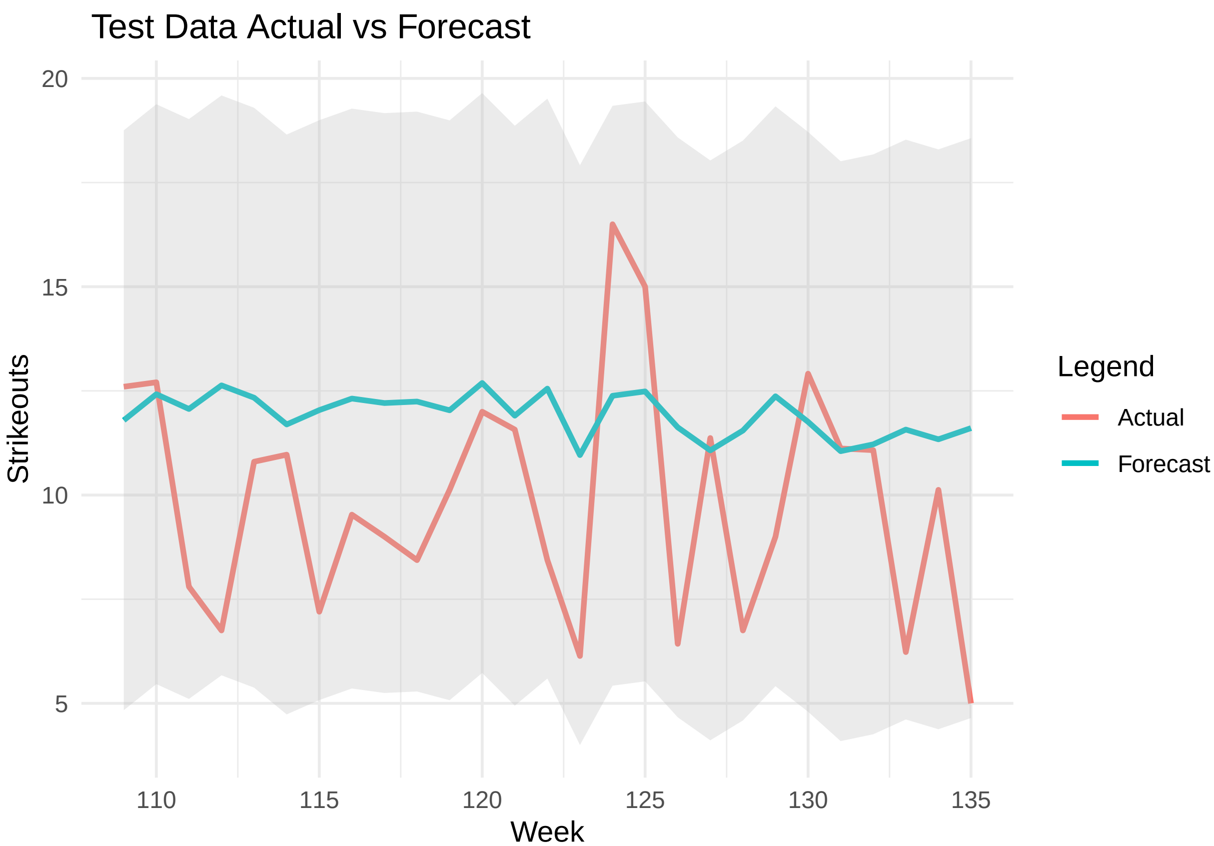
```r
comparison <- data.frame(
  Actual_K_per_9 = round(y_test, 2),
  Predicted_K_per_9 = round(pred_test, 2)
)
print(head(comparison, 10))
```

```
##     Actual_K_per_9 Predicted_K_per_9
## 127          12.60             11.79
## 133          12.71             12.42
## 134           7.80             12.07
## 135           6.75             12.63
## 136          10.80             12.34
## 137          10.97             11.69
## 138           7.20             12.04
## 139           9.53             12.32
## 140           9.00             12.21
## 141           8.44             12.24
```

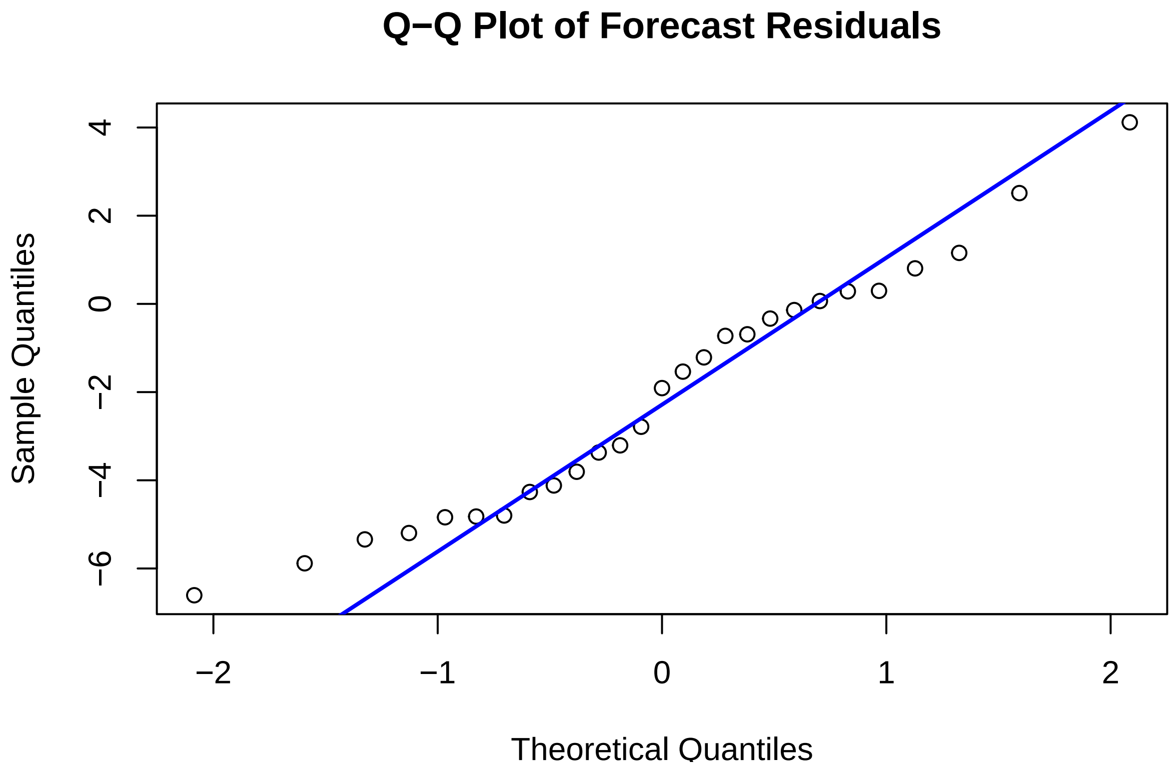
```r
test_index <- seq(train_size + 1, nrow(test) + train_size)

forecast_values <- as.numeric(forecast_test$mean)
lower95 <- forecast_test$lower[, 2]
upper95 <- forecast_test$upper[, 2]

plot_df <- data.frame(
  Week = test_index,
  Actual = y_test,
  Forecast = forecast_values,
  Lower95 = lower95,
  Upper95 = upper95
)
```

```r
ggplot(plot_df, aes(x = Week)) +
  geom_line(aes(y = Actual, color = "Actual"), size = 1) +
  geom_line(aes(y = Forecast, color = "Forecast"), size = 1) +
  geom_ribbon(aes(ymin = Lower95, ymax = Upper95), fill = "grey", alpha = 0.3) +
  labs(title = " Test Data Actual vs Forecast",
       x = "Week",
       y = "Strikeouts",
       color = "Legend") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
forecast_residuals <- y_test - forecast_test$mean

qqnorm(forecast_residuals, main = "Q-Q Plot of Forecast Residuals")
qqline(forecast_residuals, col = "blue", lwd = 2)
```


Q−Q Plot of Forecast Residuals
Sample Quantiles
Theoretical Quantiles
4
2
0
−2
−4
−6
−2
−1
0
1
2