Data Analysis Project 2

Andy Kapoor

April 2024

Question: "Do temporal patterns exist in the gaze fluctuations of a participant while playing different VR games, and do these patterns vary across games?"

Hypotheses:

Null Hypothesis (H0): There are no temporal autocorrelation patterns in gaze fluctuations across different VR games, indicating that gaze movements are random over time within each game.

Alternative Hypothesis (H1): Temporal autocorrelation patterns exist in gaze fluctuations across different VR games, suggesting that gaze movements are not random and may follow specific trends or cycles.

BeatSaber Data Graphs

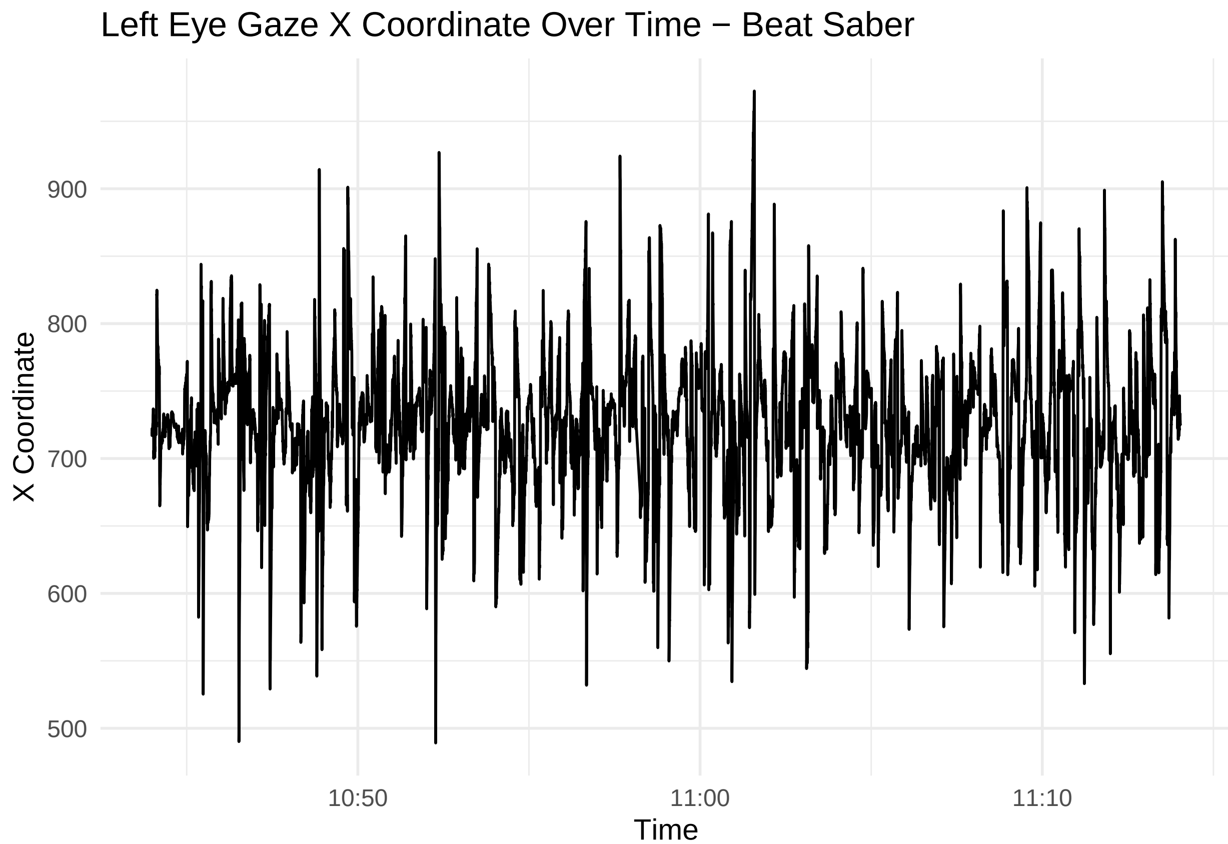
```r
participant1_path <- "/cloud/project/Participant 1"
beat_saber_data <- fread(file.path(participant1_path, "BeatSaber_11-12-2020_GazeDataAll.txt"), header =
```

```
## Warning in fread(file.path(participant1_path,
## "BeatSaber_11-12-2020_GazeDataAll.txt"), : Discarded single-line footer:
## <<637407832442086799, 724.0819 , 816.8118 , 1.948207,2.175821, 709.2674 ,
## 823.0424 , 1.948207,2.165995, 715.7529>>
```

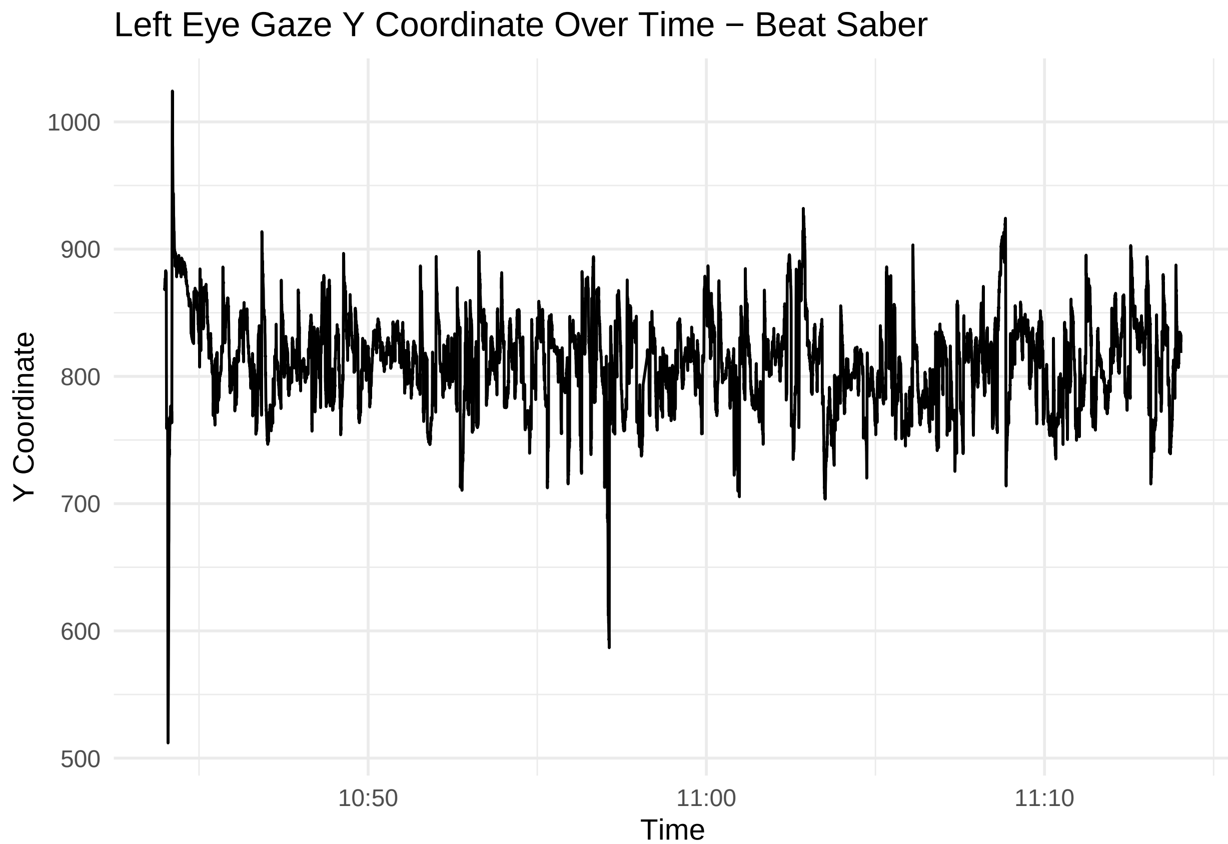
```r
beat_saber_data$Time <- as.numeric(beat_saber_data$Time)
```

```
## Warning in as.double.integer64(beat_saber_data$Time): integer precision lost
## while converting to double
```

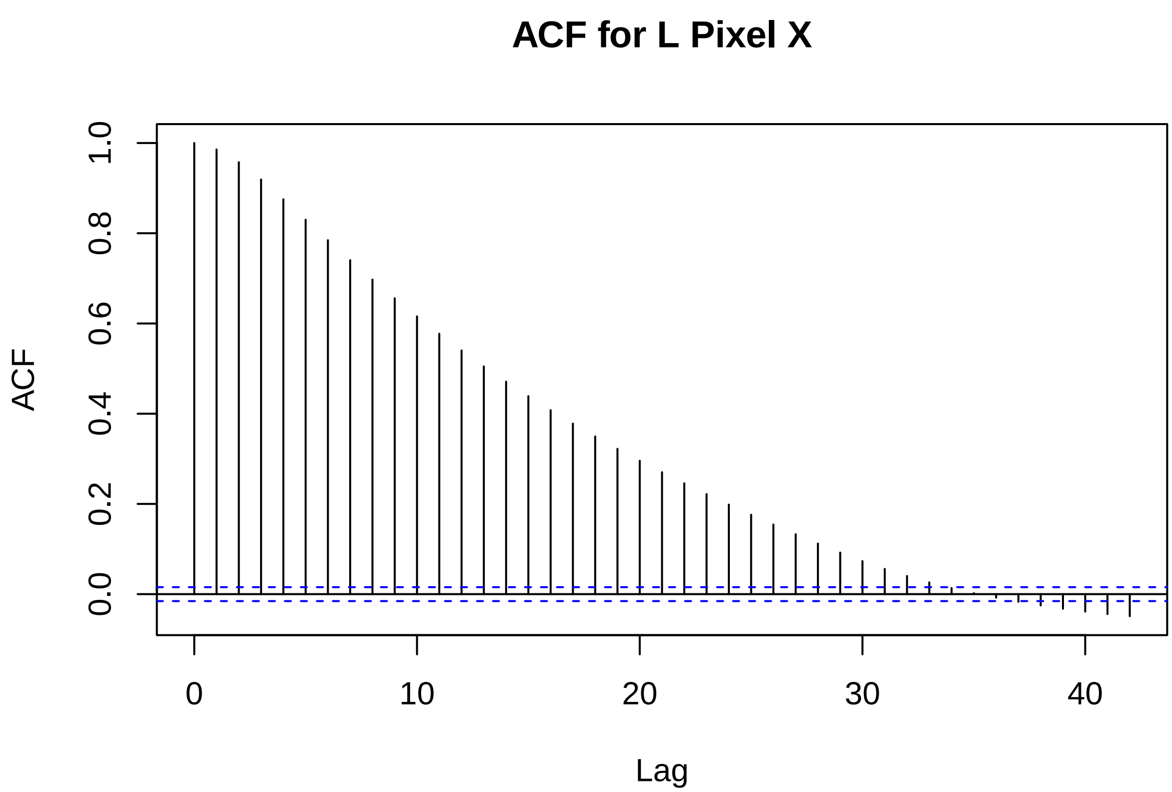
```r
beat_saber_data$Time <- as.POSIXct(beat_saber_data$Time / 1e6, tz = "UTC")

  # Plot the X axis gaze data
ggplot(beat_saber_data, aes(x = Time, y = `L Pixel X`)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Left Eye Gaze X Coordinate Over Time - Beat Saber", x = "Time", y = "X Coordinate")
```



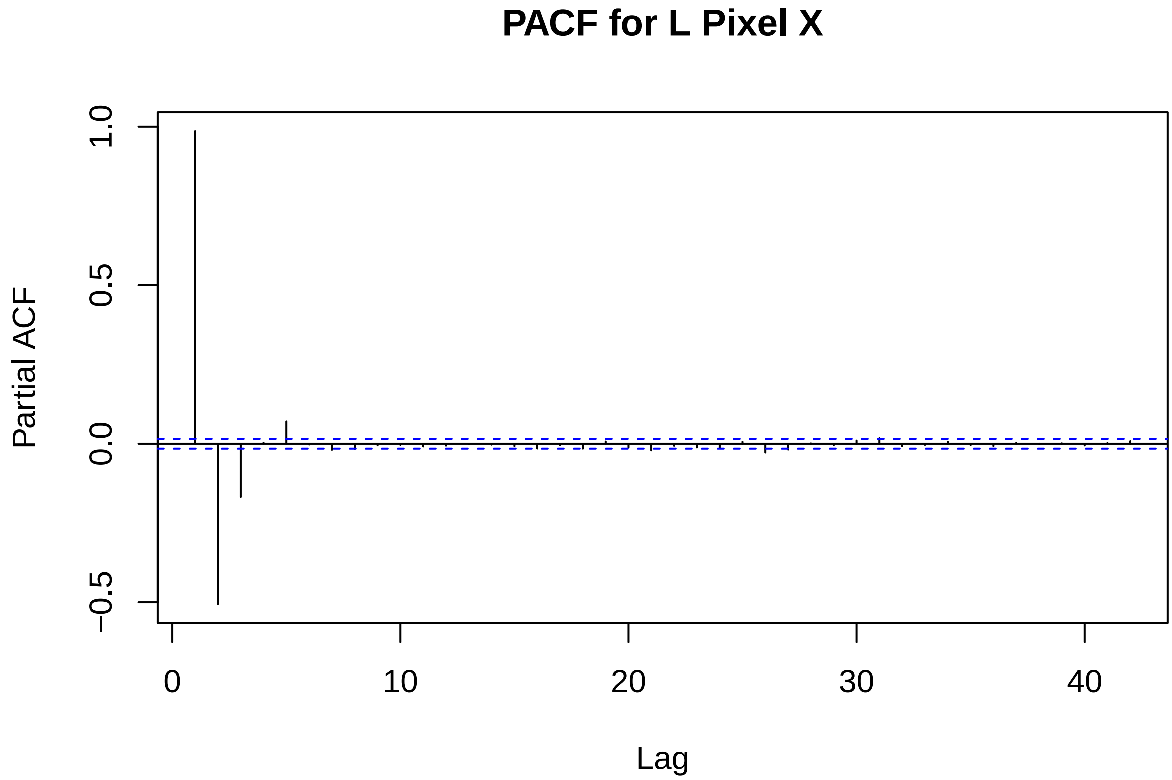

```r
# Plot the y axis gaze data
ggplot(beat_saber_data, aes(x = Time, y = `L Pixel Y`)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Left Eye Gaze Y Coordinate Over Time - Beat Saber", x = "Time", y = "Y Coordinate")
```



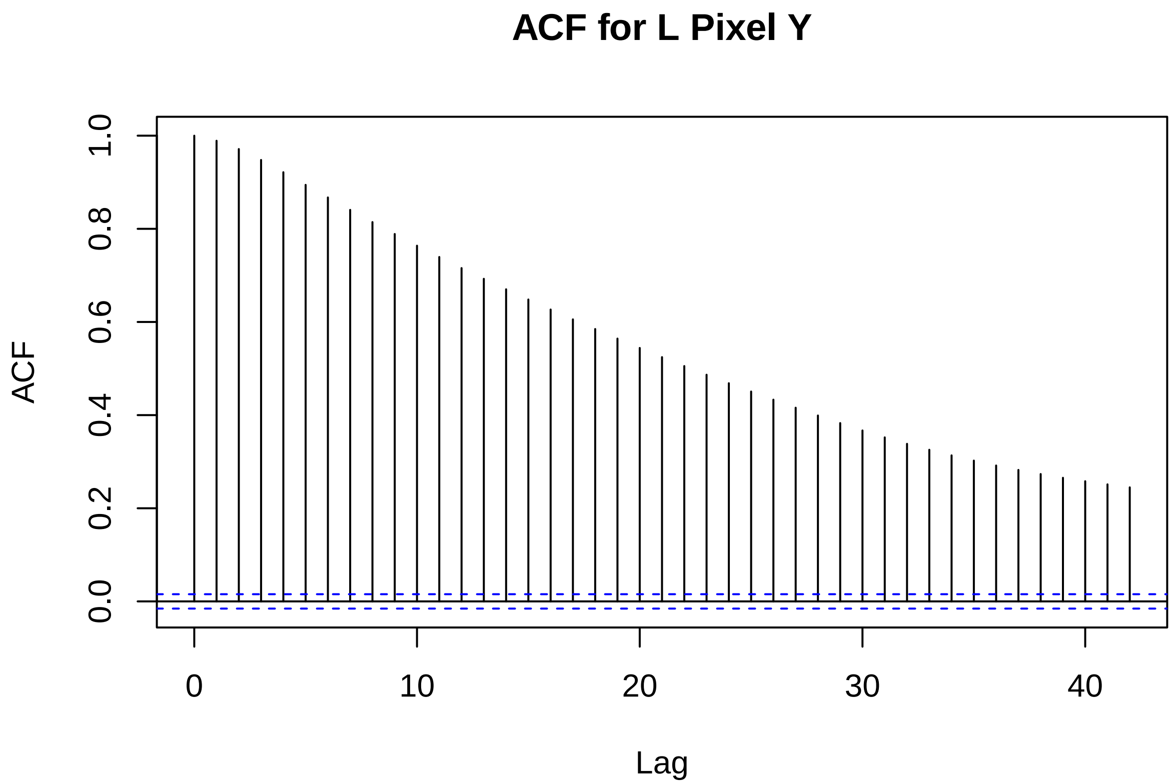

```r
# Autocorrelation Function (ACF)
acf(beat_saber_data$`L Pixel X`, main = "ACF for L Pixel X")
```



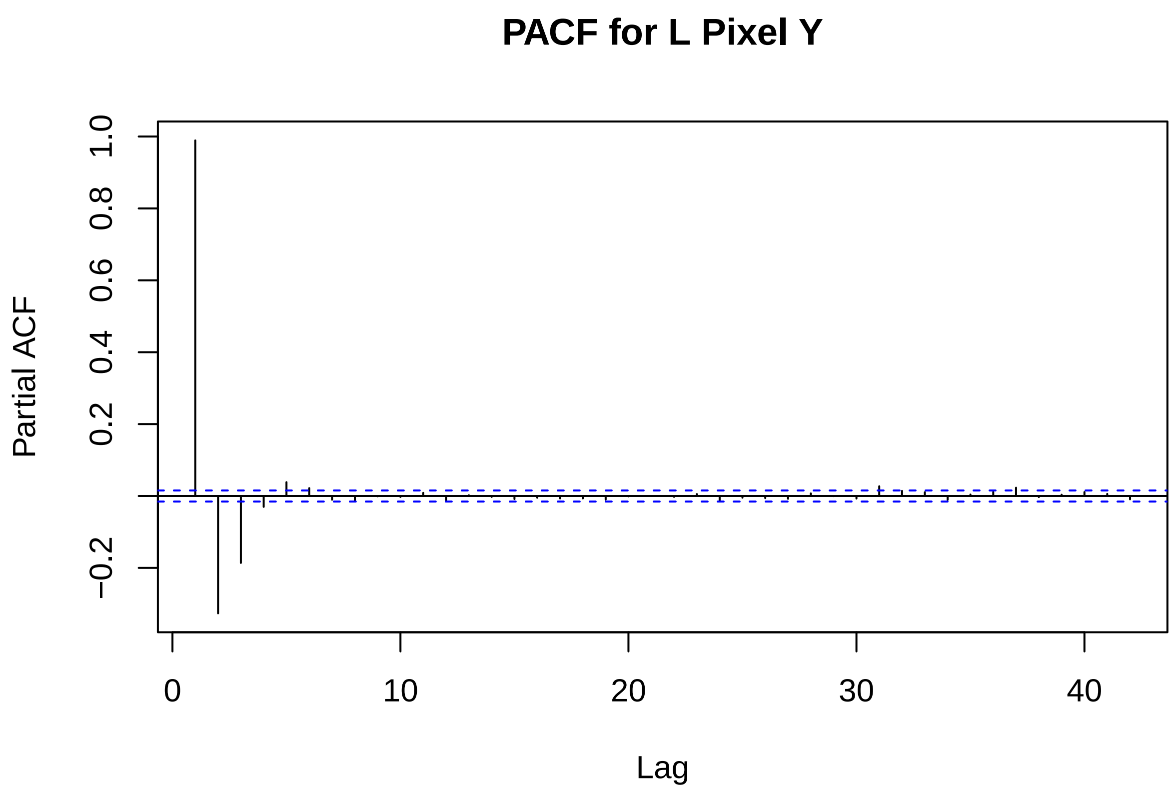
```r
# Partial Autocorrelation Function (PACF)
pacf(beat_saber_data$`L Pixel X`, main = "PACF for L Pixel X")
```




```r
# Autocorrelation Function (ACF)
acf(beat_saber_data$`L Pixel Y`, main = "ACF for L Pixel Y")
```




```r
# Partial Autocorrelation Function (PACF)
pacf(beat_saber_data$`L Pixel Y`, main = "PACF for L Pixel Y")
```



ACF values close to one indicate strong correlation. The gradual decline in the ACF plot suggests a decreasing correlation between the data point and its past values at longer lags.

For the PACF plot of L Pixel X, the first lag is significant, and the rest are not, which suggests that only the immediate past value (one time step back) has a direct and significant effect on the current value, after accounting for the intermediate effects.

Confidence Interval (Dotted Blue Line): This represents the range of values within which we would expect the ACF or PACF values to fall if the series were purely random (white noise). Typically, this is set at the 95% confidence level, meaning we would expect 95% of the values to fall within this range if there were no true autocorrelations in the data. Values Well Outside the CI: When ACF or PACF values extend beyond the confidence interval, it suggests that the observed correlation at that lag is statistically significant and is unlikely to be due to random chance. Values Inside the CI: Conversely, values that lie within the confidence interval are not statistically significant; they could likely be attributed to randomness.

```
BS_adf_test_result_x <- adf.test(beat_saber_data$`L Pixel X`, alternative = "stationary")
```

```
## Warning in adf.test(beat_saber_data$`L Pixel X`, alternative = "stationary"):
## p-value smaller than printed p-value
```

```
print(BS_adf_test_result_x)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  beat_saber_data$`L Pixel X`
## Dickey-Fuller = -18.784, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```
BS_kpss_test_result_x <- kpss.test(beat_saber_data$`L Pixel X`, null = "Trend")
```

```
## Warning in kpss.test(beat_saber_data$`L Pixel X`, null = "Trend"): p-value
## greater than printed p-value
```

```
print(BS_kpss_test_result_x)
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  beat_saber_data$`L Pixel X`
## KPSS Trend = 0.045929, Truncation lag parameter = 14, p-value = 0.1
```

```
print("---------------------------------------------------------------------------------------")
```

```
## [1] "---------------------------------------------------------------------------------------"
```

```
BS_adf_test_result_y <- adf.test(beat_saber_data$`L Pixel Y`, alternative = "stationary")
```

```
## Warning in adf.test(beat_saber_data$`L Pixel Y`, alternative = "stationary"):
## p-value smaller than printed p-value
```

```
print(BS_adf_test_result_y)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  beat_saber_data$`L Pixel Y`
## Dickey-Fuller = -14.528, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```
BS_kpss_test_result_y <- kpss.test(beat_saber_data$`L Pixel Y`, null = "Trend")
```

```
## Warning in kpss.test(beat_saber_data$`L Pixel Y`, null = "Trend"): p-value
```

```
## smaller than printed p-value
```

```r
print(BS_kpss_test_result_y)
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  beat_saber_data$`L Pixel Y`
## KPSS Trend = 0.65954, Truncation lag parameter = 14, p-value = 0.01
```

```r
print("DIFFERENCE DATA BELOW-------------------------------------------------------------------")
```

```
## [1] "DIFFERENCE DATA BELOW-------------------------------------------------------------------"
```

```r
#KPSS results for L Pixel Y are below 0.05, suggesting there are trends we must account for before fitt
  # Differencing the data

beat_saber_data$`L Pixel Y_diff` <- c(NA, diff(beat_saber_data$`L Pixel Y`))

# Running ADF test after removing NAs
BS_adf_test_result_y_diff <- adf.test(na.omit(beat_saber_data$`L Pixel Y_diff`), alternative = "stationa
```

```
## Warning in adf.test(na.omit(beat_saber_data$`L Pixel Y_diff`), alternative =
## "stationary"): p-value smaller than printed p-value
```

```r
print(BS_adf_test_result_y_diff)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(beat_saber_data$`L Pixel Y_diff`)
## Dickey-Fuller = -27.082, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```r
BS_kpss_test_result_y_diff <- kpss.test(na.omit(beat_saber_data$`L Pixel Y_diff`), null = "Trend")
```

```
## Warning in kpss.test(na.omit(beat_saber_data$`L Pixel Y_diff`), null =
## "Trend"): p-value greater than printed p-value
```

```r
print(BS_kpss_test_result_y_diff)
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  na.omit(beat_saber_data$`L Pixel Y_diff`)
## KPSS Trend = 0.00146, Truncation lag parameter = 14, p-value = 0.1
```

```r
BS_initial_model_x <- auto.arima(beat_saber_data$`L Pixel X`, max.p=10, max.d=0, max.q=0, seasonal=FALS
summary(BS_initial_model_x)
```

```
## Series: beat_saber_data$`L Pixel X`
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3      mean
##       1.3995  -0.2567  -0.1678  726.5699
## s.e.  0.0078   0.0133   0.0078    2.3804
##
## sigma^2 = 57.23:  log likelihood = -55627.05
```

```
## AIC=111264.1   AICc=111264.1   BIC=111302.6
##
## Training set error measures:
##                        ME     RMSE      MAE         MPE      MAPE      MASE
## Training set 9.738866e-05 7.564084 4.136636 -0.01088803 0.5777833 0.9635035
##                      ACF1
## Training set 0.0005188375
```

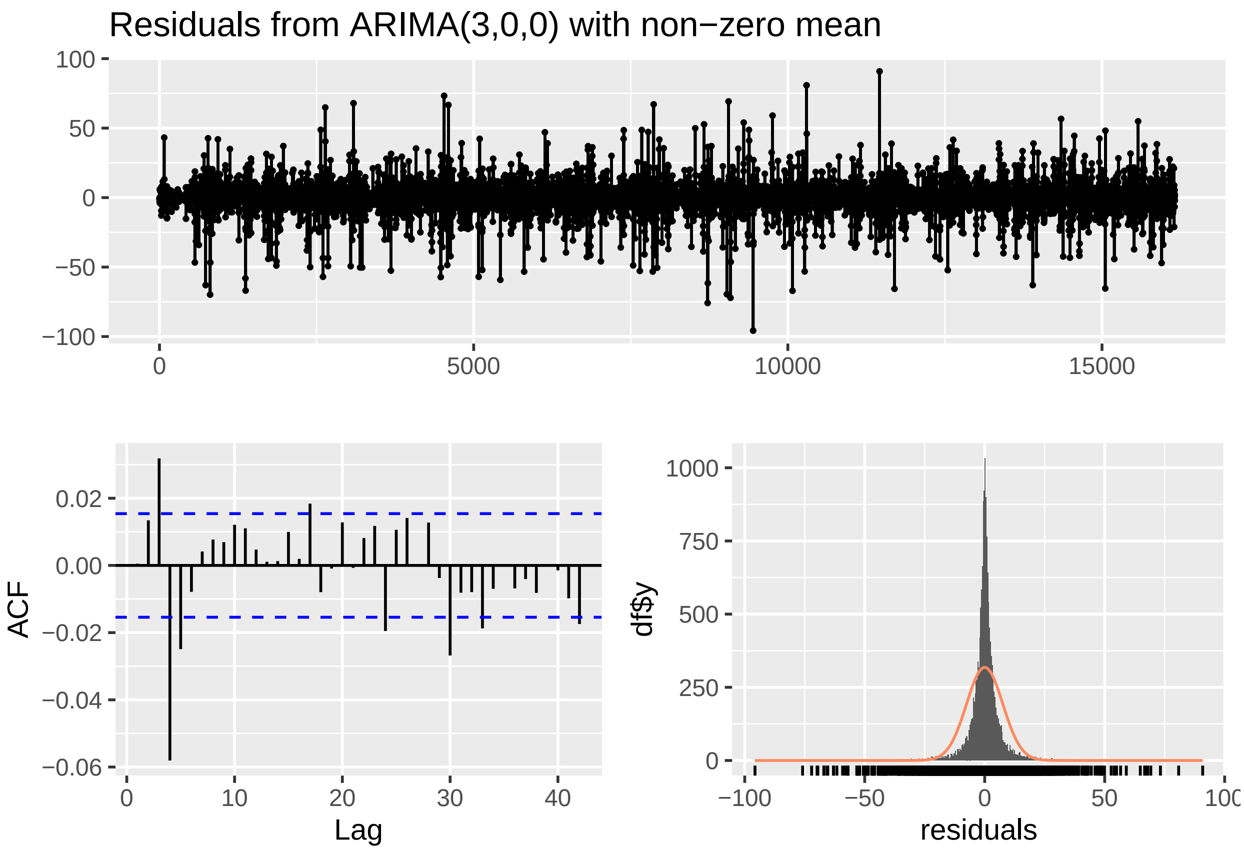
```
print("-------------------------------------------------------------------------------------")
```

```
## [1] "-------------------------------------------------------------------------------------"
```

```
BS_initial_model_y <- auto.arima(beat_saber_data$`L Pixel Y_diff`, max.p=10, max.d=0, max.q=0, seasonal=
summary(BS_initial_model_y)
```

```
## Series: beat_saber_data$`L Pixel Y_diff`
## ARIMA(5,0,0) with zero mean
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ar5
##       0.2617  0.1813  0.0344  -0.0446  -0.0367
## s.e.  0.0079  0.0081  0.0082   0.0081   0.0079
##
## sigma^2 = 25.46:  log likelihood = -49078.58
## AIC=98169.15   AICc=98169.16   BIC=98215.29
##
## Training set error measures:
##                        ME     RMSE      MAE MPE MAPE      MASE          ACF1
## Training set -0.002059612 5.045376 2.920253 NaN  Inf 0.7434936 -0.0001026263
```

```
# Check the residuals
checkresiduals(BS_initial_model_x)
```

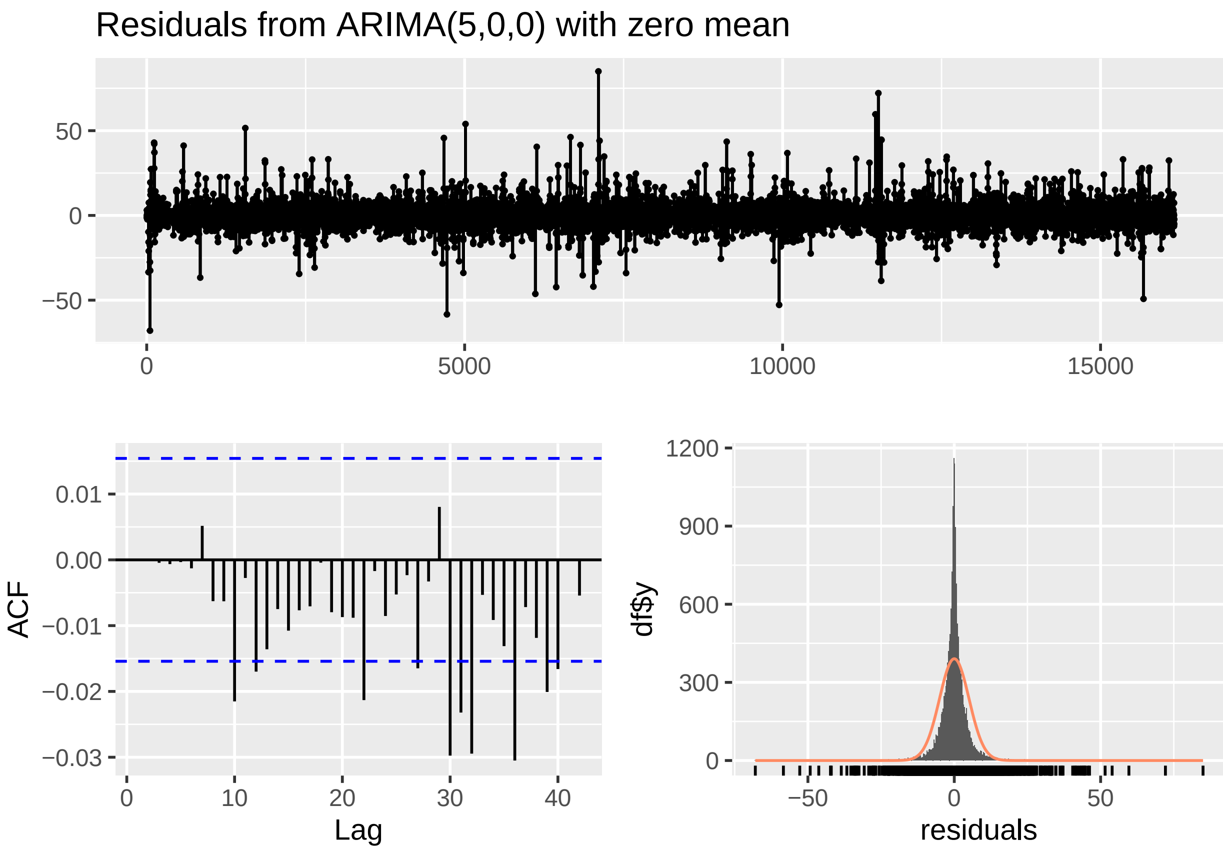



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0) with non-zero mean
## Q* = 89.219, df = 7, p-value = 2.22e-16
##
## Model df: 3.   Total lags used: 10
```

```r
print("------------------------------------------------------------------------------------")
```

```
## [1] "------------------------------------------------------------------------------------"
```

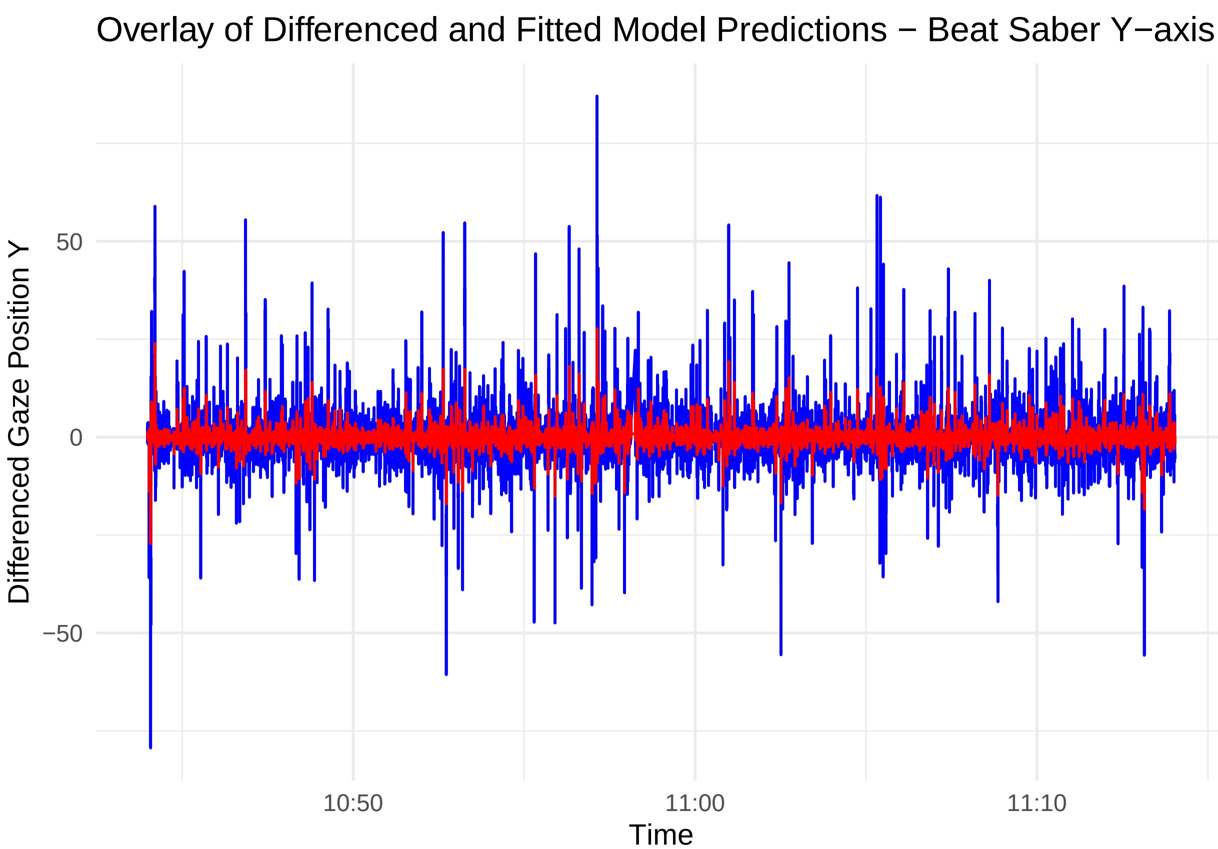
```r
checkresiduals(BS_initial_model_y)
```



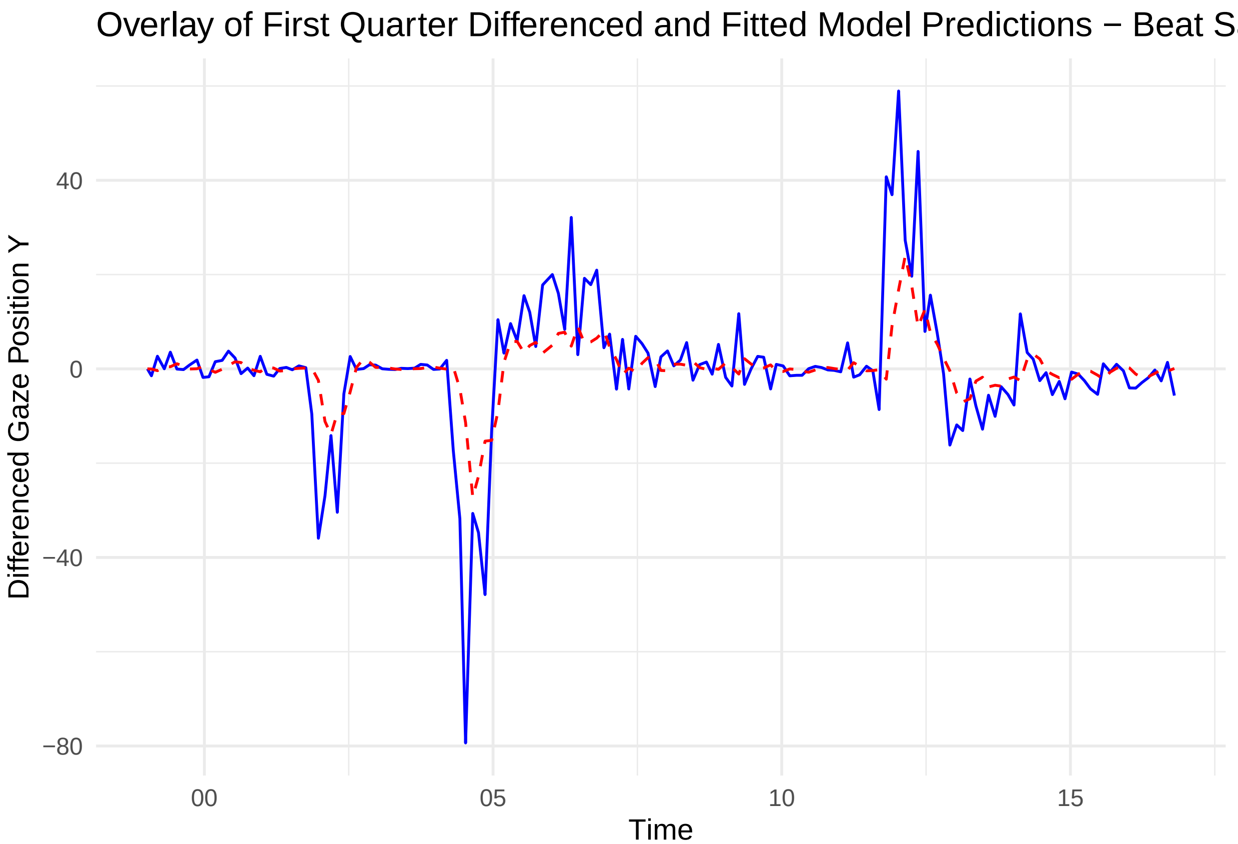

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,0,0) with zero mean
## Q* = 9.2333, df = 5, p-value = 0.1001
##
## Model df: 5.   Total lags used: 10
```

The significant Ljung-Box test tells us there is still some pattern or relationship in the data that has not been accounted for. The model could be improved to better predict the series and reduce forecast error.

```
  # Since beastsaber y data failed KPSS assumption test, it had to be differenced.
    # The plot overlays fitted differenced data with the actual...
    # ...to see how well the predictions align.
plot_data <- data.frame(
  Time = beat_saber_data$Time[-1],
  Differenced_Y = beat_saber_data$`L Pixel Y_diff`[-1],
  Fitted_Y = fitted(BS_initial_model_y)
)

  # Plot the differenced data and the fitted predictions
ggplot(plot_data, aes(x = Time)) +
  geom_line(aes(y = Differenced_Y), colour = "blue") +
  geom_line(aes(y = Fitted_Y), colour = "red") +
  labs(title = "Overlay of Differenced and Fitted Model Predictions - Beat Saber Y-axis",
       x = "Time", y = "Differenced Gaze Position Y") +
  theme_minimal()
```



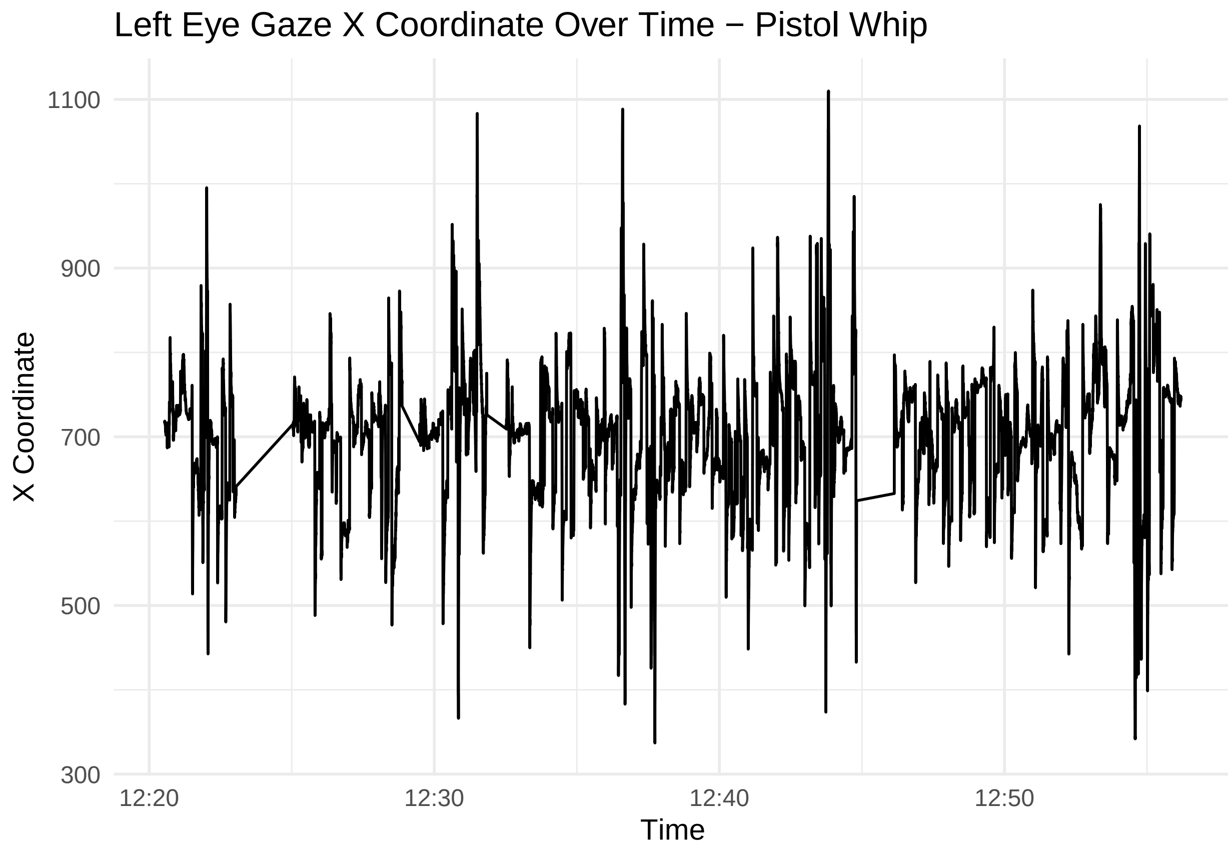

```
num_obs <- nrow(beat_saber_data)
quarter_point <- round(num_obs * 0.01)

  # Subset the data to include only the first 1% of the full plot
plot_data <- data.frame(
  Time = beat_saber_data$Time[2:quarter_point],  # Starting from 2 to avoid NA in the first element
  Differenced_Y = beat_saber_data$`L Pixel Y_diff`[2:quarter_point],
  Fitted_Y = fitted(BS_initial_model_y)[1:(quarter_point-1)]  # Adjusting for the range since fitted va
)

# Plot the first quarter of the differenced data and the fitted predictions
ggplot(plot_data, aes(x = Time)) +
  geom_line(aes(y = Differenced_Y), colour = "blue") +
  geom_line(aes(y = Fitted_Y), colour = "red", linetype = "dashed") +
  labs(title = "Overlay of First Quarter Differenced and Fitted Model Predictions - Beat Saber Y-axis",
       x = "Time", y = "Differenced Gaze Position Y") +
  theme_minimal()
```



```r
library(data.table)
library(ggplot2)

participant1_path <- "/cloud/project/Participant 1"
pistolwhip_data <- fread(file.path(participant1_path, "PistolWhip_11-12-2020_GazeDataAll.txt"),
                         header = TRUE,
                         sep = ",",
                         strip.white = TRUE,
                         fill = TRUE)

pistolwhip_data[] <- lapply(pistolwhip_data, function(x) if (class(x) == "integer64") as.numeric(x) else
```

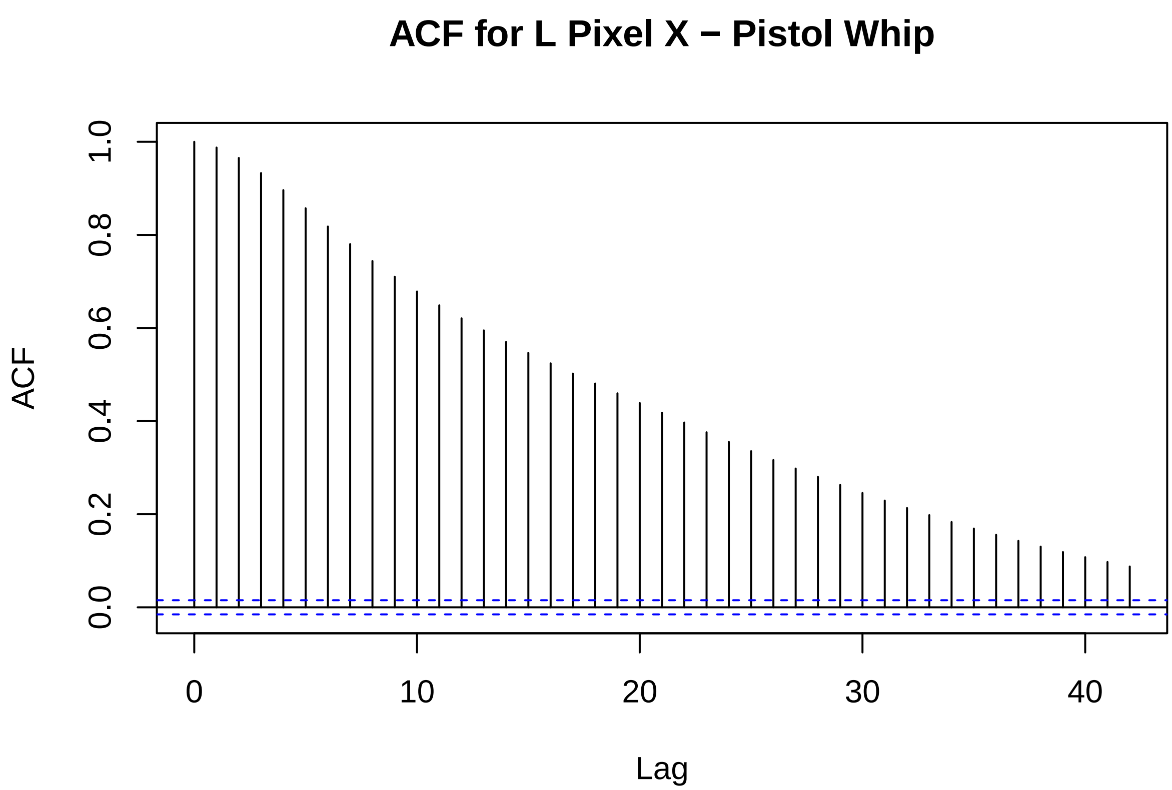
```
## Warning in as.double.integer64(x): integer precision lost while converting to
## double
```

```r
pistolwhip_data$Time <- as.numeric(pistolwhip_data$Time)
pistolwhip_data$Time <- as.POSIXct(pistolwhip_data$Time / 1e6, tz = "UTC")


ggplot(pistolwhip_data, aes(x = Time, y = `L Pixel X`)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Left Eye Gaze X Coordinate Over Time - Pistol Whip", x = "Time", y = "X Coordinate")
```

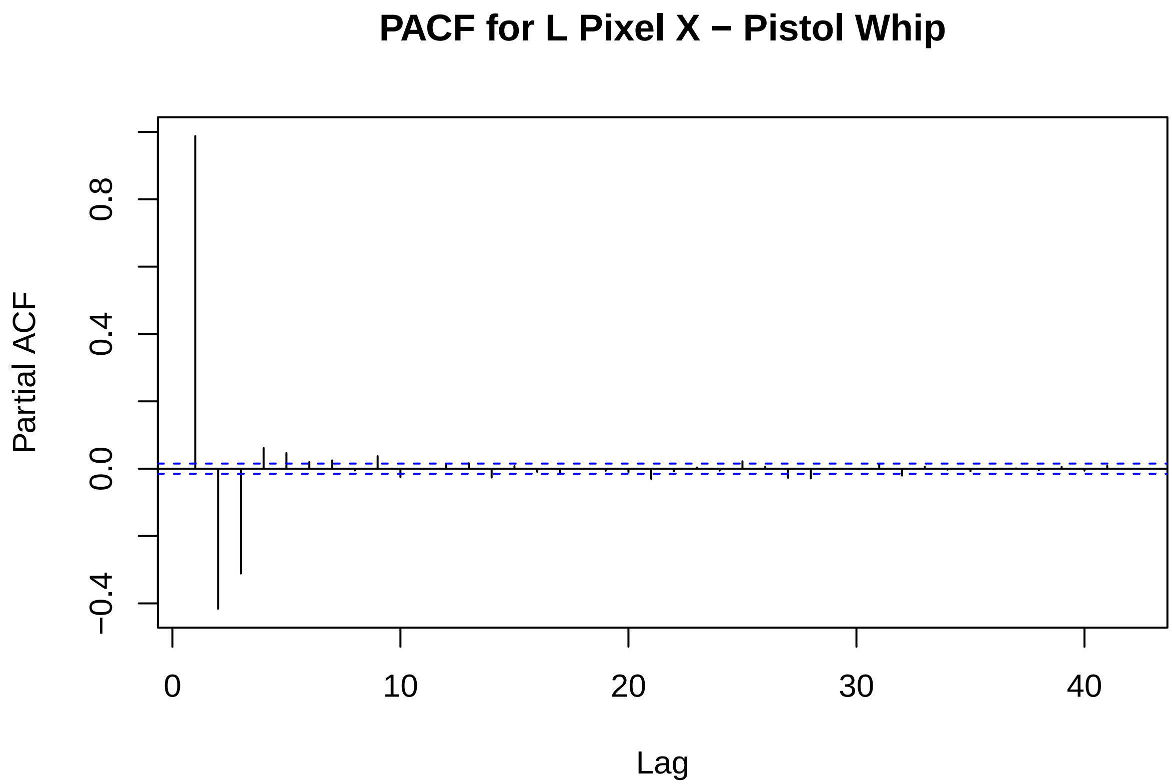



```r
ggplot(pistolwhip_data, aes(x = Time, y = `L Pixel Y`)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Left Eye Gaze Y Coordinate Over Time - Pistol Whip", x = "Time", y = "Y Coordinate")
```

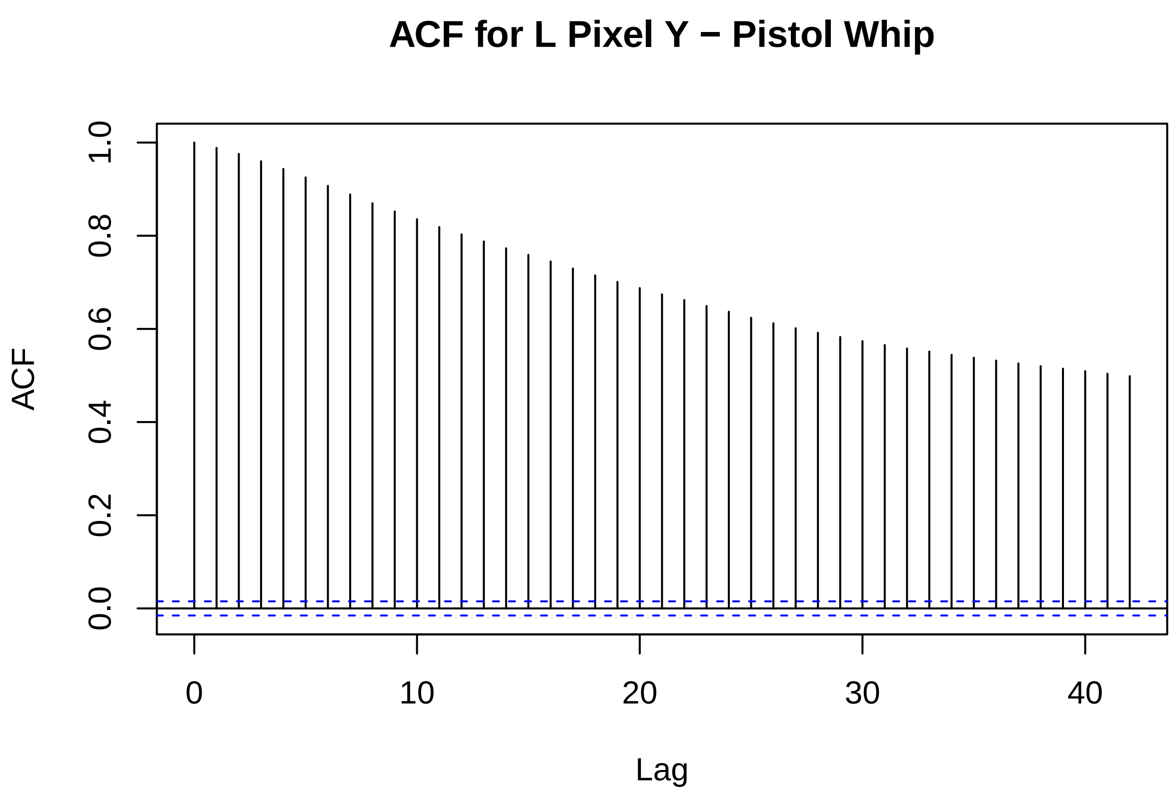

```r
# ACF and PACF for X axis
acf(pistolwhip_data$`L Pixel X`, main = "ACF for L Pixel X - Pistol Whip")
```



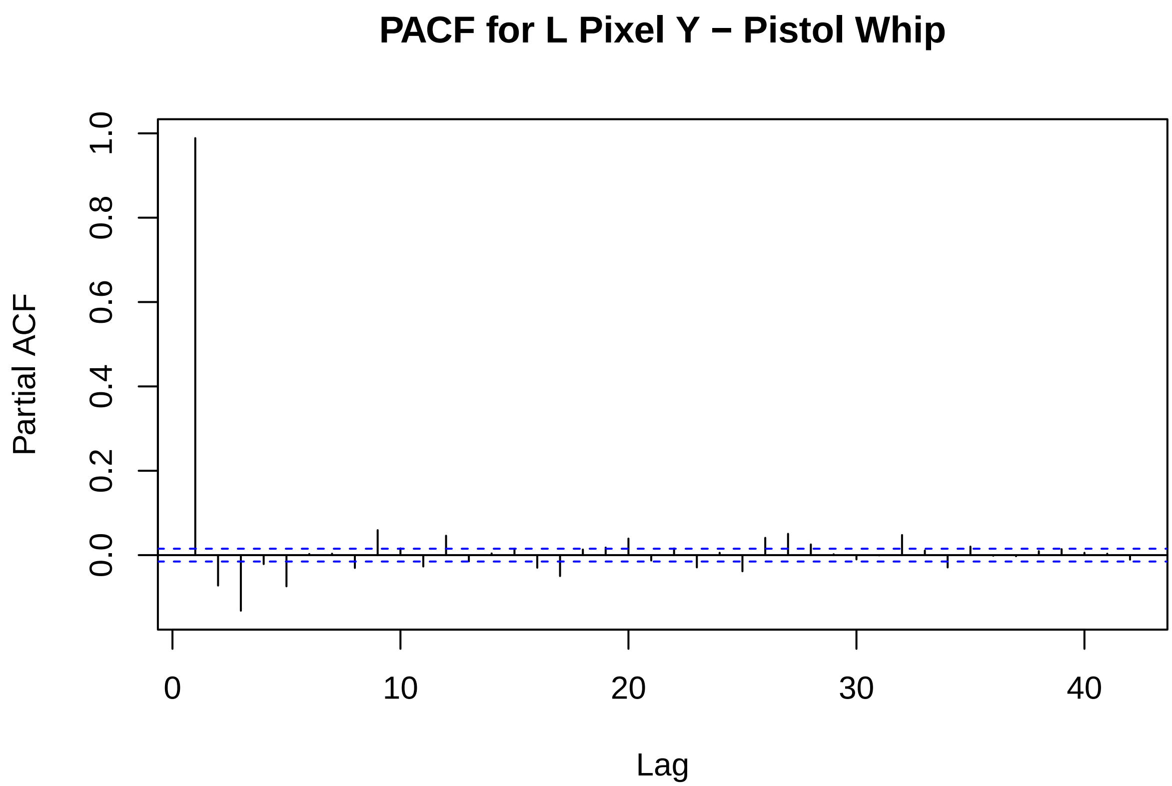
```r
pacf(pistolwhip_data$`L Pixel X`, main = "PACF for L Pixel X - Pistol Whip")
```




```r
# ACF and PACF for Y axis
acf(pistolwhip_data$`L Pixel Y`, main = "ACF for L Pixel Y - Pistol Whip")
```



```r
pacf(pistolwhip_data$`L Pixel Y`, main = "PACF for L Pixel Y - Pistol Whip")
```




```r
# ADF and KPSS tests for X axis
PW_adf_test_result_x <- adf.test(pistolwhip_data$`L Pixel X`, alternative = "stationary")
```

```
## Warning in adf.test(pistolwhip_data$`L Pixel X`, alternative = "stationary"):
## p-value smaller than printed p-value
```

```r
print(PW_adf_test_result_x)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  pistolwhip_data$`L Pixel X`
## Dickey-Fuller = -15.739, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```r
PW_kpss_test_result_x <- kpss.test(pistolwhip_data$`L Pixel X`, null = "Trend")
print(PW_kpss_test_result_x)
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  pistolwhip_data$`L Pixel X`
## KPSS Trend = 0.14361, Truncation lag parameter = 14, p-value = 0.05443
```

```r
print("-----------------------------------------------------------------------------------------")
```

```
## [1] "-----------------------------------------------------------------------------------------"
```

```r
# ADF and KPSS tests for Y axis
PW_adf_test_result_y <- adf.test(pistolwhip_data$`L Pixel Y`, alternative = "stationary")
```

```
## Warning in adf.test(pistolwhip_data$`L Pixel Y`, alternative = "stationary"):
## p-value smaller than printed p-value
```

```r
print(PW_adf_test_result_y)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  pistolwhip_data$`L Pixel Y`
## Dickey-Fuller = -11.313, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```r
PW_kpss_test_result_y <- kpss.test(pistolwhip_data$`L Pixel Y`, null = "Trend")
```

```
## Warning in kpss.test(pistolwhip_data$`L Pixel Y`, null = "Trend"): p-value
## smaller than printed p-value
```

```r
print(PW_kpss_test_result_y)
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  pistolwhip_data$`L Pixel Y`
## KPSS Trend = 0.46192, Truncation lag parameter = 14, p-value = 0.01
```

```r
print("DIFFERENCE DATA BELOW---------------------------------------------------------------------")
```

```
## [1] "DIFFERENCE DATA BELOW---------------------------------------------------------------------"
```

```r
#KPSS results for L Pixel Y are below 0.05, suggesting there are trends we must account for before fitt
  # Differencing the data

pistolwhip_data$`L Pixel Y_diff` <- c(NA, diff(pistolwhip_data$`L Pixel Y`))

# Running ADF test after removing NAs
PW_adf_test_result_y_diff <- adf.test(na.omit(pistolwhip_data$`L Pixel Y_diff`), alternative = "stationa
```

```
## Warning in adf.test(na.omit(pistolwhip_data$`L Pixel Y_diff`), alternative =
## "stationary"): p-value smaller than printed p-value
```

```r
print(PW_adf_test_result_y_diff)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(pistolwhip_data$`L Pixel Y_diff`)
## Dickey-Fuller = -29.332, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```r
PW_kpss_test_result_y_diff <- kpss.test(na.omit(pistolwhip_data$`L Pixel Y_diff`), null = "Trend")
```

```
## Warning in kpss.test(na.omit(pistolwhip_data$`L Pixel Y_diff`), null =
## "Trend"): p-value greater than printed p-value
```

```r
print(PW_kpss_test_result_y_diff)
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  na.omit(pistolwhip_data$`L Pixel Y_diff`)
```

```
## KPSS Trend = 0.0026534, Truncation lag parameter = 14, p-value = 0.1
```

```r
# Fit ARIMA model for X axis
PW_initial_model_x <- auto.arima(pistolwhip_data$`L Pixel X`, max.p=10, max.d=0, max.q=0, seasonal=FALSE
summary(PW_initial_model_x)
```

```
## Series: pistolwhip_data$`L Pixel X`
## ARIMA(7,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3     ar4     ar5      ar6     ar7      mean
##       1.2842  0.0358  -0.3836  0.0109  0.0210  -0.0126  0.0247  702.2567
## s.e.  0.0077  0.0126   0.0126  0.0129  0.0126   0.0126  0.0077    4.1192
##
## sigma^2 = 109:  log likelihood = -62953.09
## AIC=125924.2   AICc=125924.2   BIC=125993.7
##
## Training set error measures:
##                       ME    RMSE      MAE         MPE      MAPE      MASE
## Training set 0.001005457 10.4397 4.752547 -0.02315367 0.7015402 0.9781748
##                      ACF1
## Training set 0.0001744432
```

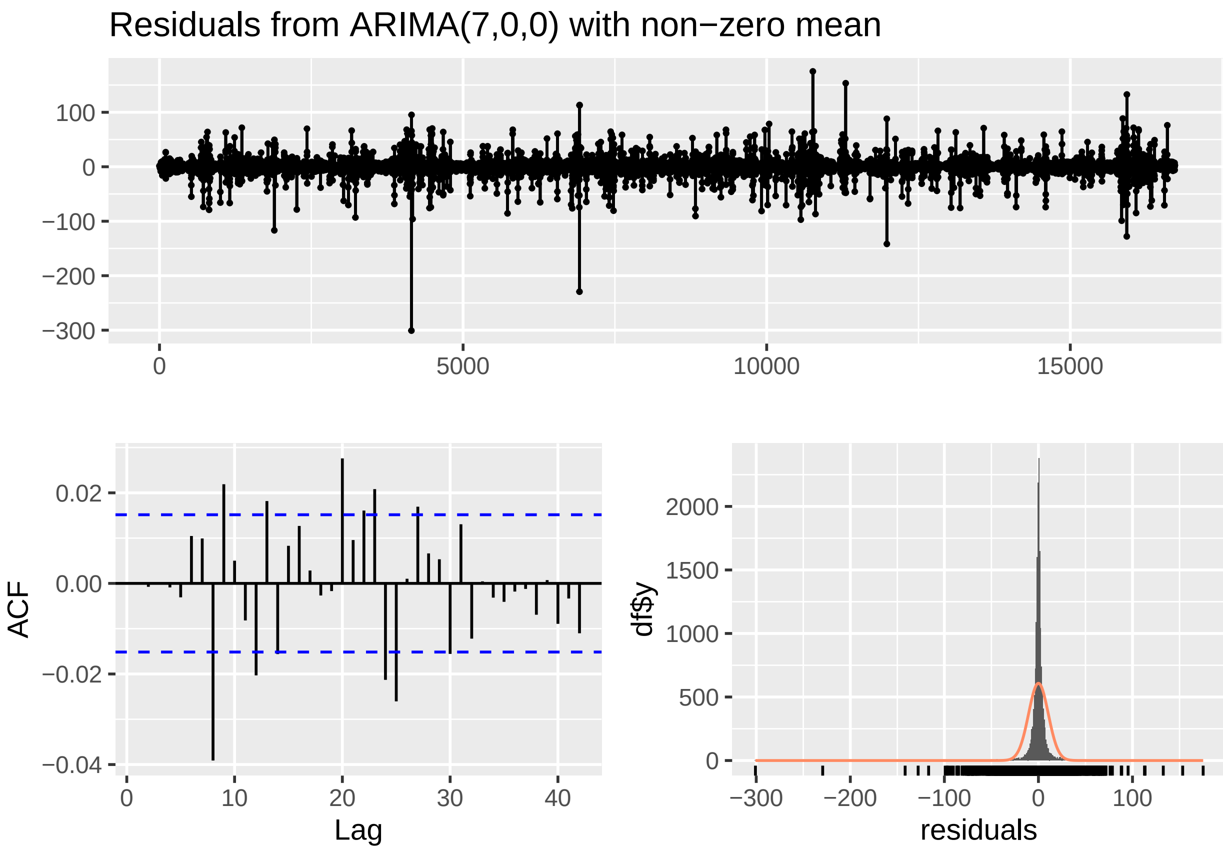
```r
print("---------------------------------------------------------------------------------------")
```

```
## [1] "---------------------------------------------------------------------------------------"
```

```r
# Fit ARIMA model for Y axis
PW_initial_model_y <- auto.arima(pistolwhip_data$`L Pixel Y_diff`, max.p=10, max.d=0, max.q=0, seasonal=
summary(PW_initial_model_y)
```

```
## Series: pistolwhip_data$`L Pixel Y_diff`
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1     ar2
##       0.0585  0.1253
## s.e.  0.0077  0.0077
##
## sigma^2 = 129.5:  log likelihood = -64384.87
## AIC=128775.7   AICc=128775.8   BIC=128798.9
##
## Training set error measures:
##                      ME     RMSE      MAE MPE MAPE      MASE         ACF1
## Training set 0.00871982 11.37707 4.144423 NaN  Inf 0.6834317 -0.001618065
```

```r
checkresiduals(PW_initial_model_x)
```

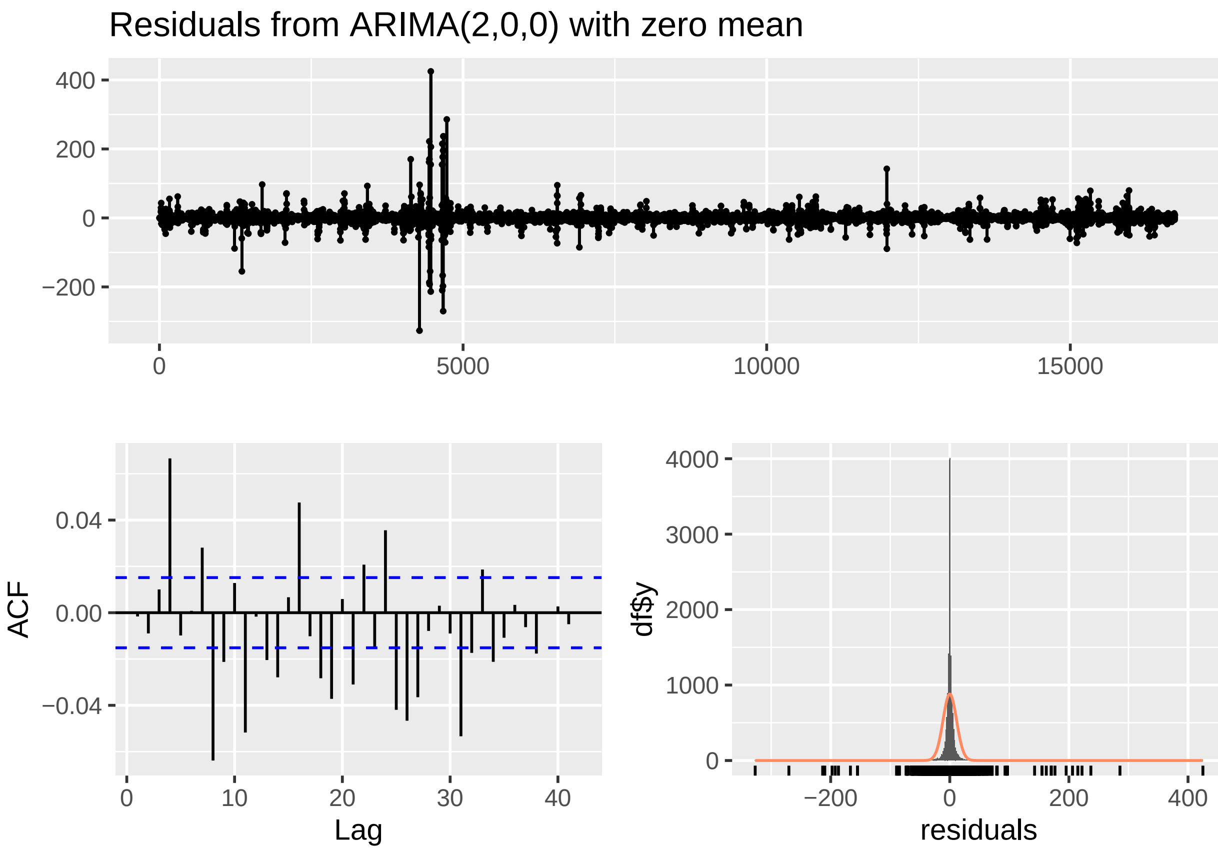



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(7,0,0) with non-zero mean
## Q* = 37.689, df = 3, p-value = 3.289e-08
##
## Model df: 7.   Total lags used: 10
```

```r
print("---------------------------------------------------------------------------------------")
```

```
## [1] "---------------------------------------------------------------------------------------"
```

```r
checkresiduals(PW_initial_model_y)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0) with zero mean
## Q* = 170.62, df = 8, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 10
```