Introduction

This report utilizes historical data from January 2021 to December 2024 to model and forecast the closing prices of JPMorgan Chase & Co. preferred shares (JPM-PD) for the period of January to December 2025 using seasonal ARIMA models in R. The analysis includes data preprocessing, stationarity testing, model comparison, forecasting, and result visualization.

Data Collection

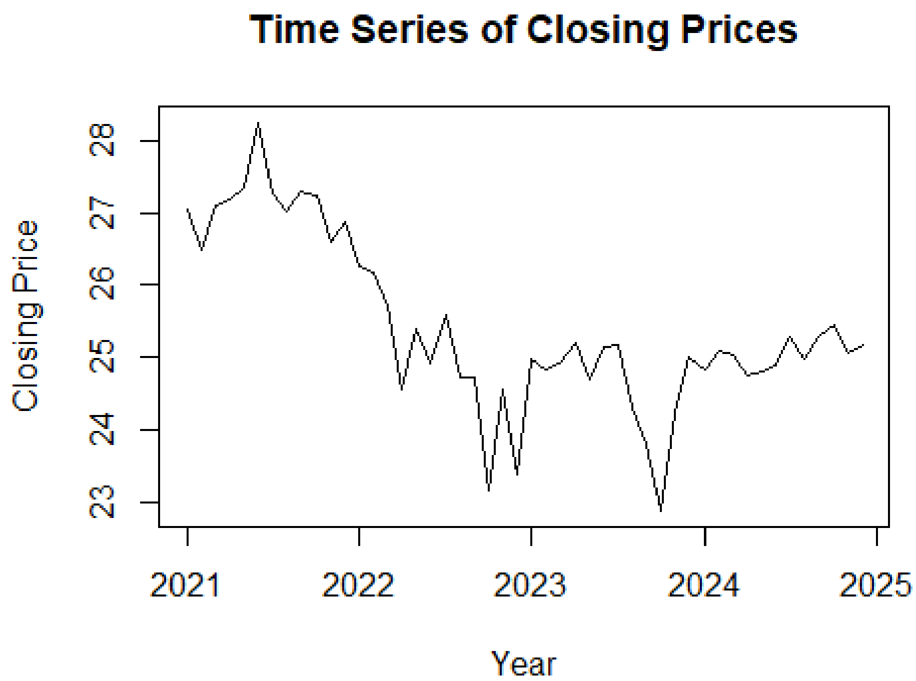
```
library(quantmod)
library(forecast)
library(tseries)
library(TSA)
library(lubridate)
library(dplyr)
library(openxlsx)
library(ggplot2)
```

Retrieve monthly closing price data from Yahoo Finance:

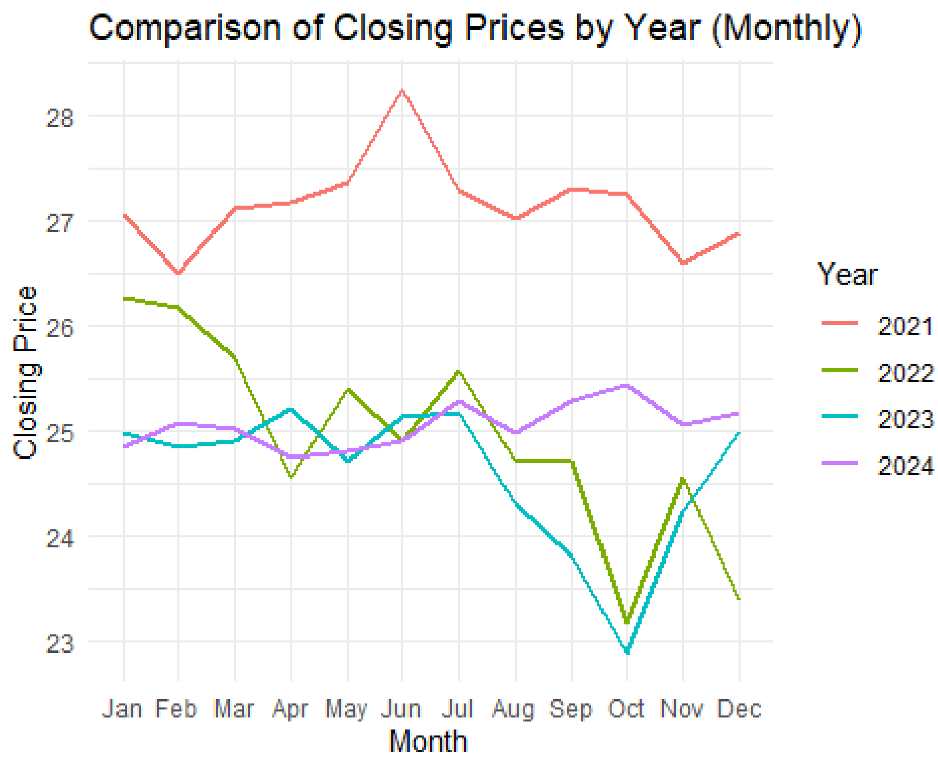
```
getSymbols("JPM-PD", src = "yahoo", from = "2021-01-01", to = "2024-12-31",
periodicity = "monthly")

## [1] "JPM-PD"

data <- `JPM-PD`
close_xts <- Cl(data)
df <- data.frame(Date = index(close_xts), Close = coredata(close_xts))
ts <- ts(df$JPM.PD.Close, start = c(2021, 1), frequency = 12)
```

Exploratory Data Analysis

```
plot(ts, main = "Time Series of Closing Prices", ylab = "Closing Price", xlab
= "Year")
```




```
df_ts <- data.frame(
  Date = seq(as.Date("2021-01-01"), by = "month", length.out = length(ts)),
  Close = as.numeric(ts)
)
df_ts$Month <- month(df_ts$Date, label = TRUE, abbr = TRUE)
df_ts$Year <- factor(year(df_ts$Date))

ggplot(df_ts, aes(x = Month, y = Close, color = Year, group = Year)) +
  geom_line(size = 1) +
  labs(title = "Comparison of Closing Prices by Year (Monthly)",
       x = "Month", y = "Closing Price") +
  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.
```



Stationarity Test

```
adf.test(ts, alternative = "stationary")

##
##  Augmented Dickey-Fuller Test
##
## data:  ts
## Dickey-Fuller = -1.6093, Lag order = 3, p-value = 0.7302
## alternative hypothesis: stationary
```

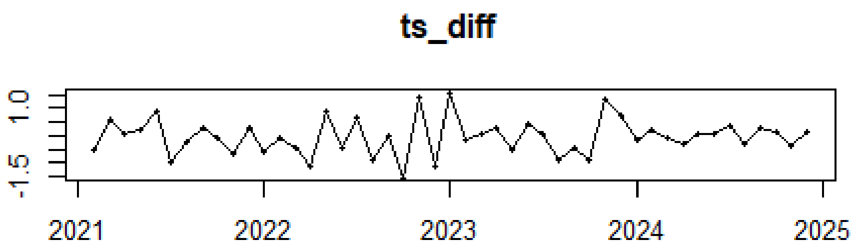
The Augmented Dickey-Fuller test result shows a p-value of 0.7302 (> 0.05), indicating that the time series is non-stationary at level and requires differencing.

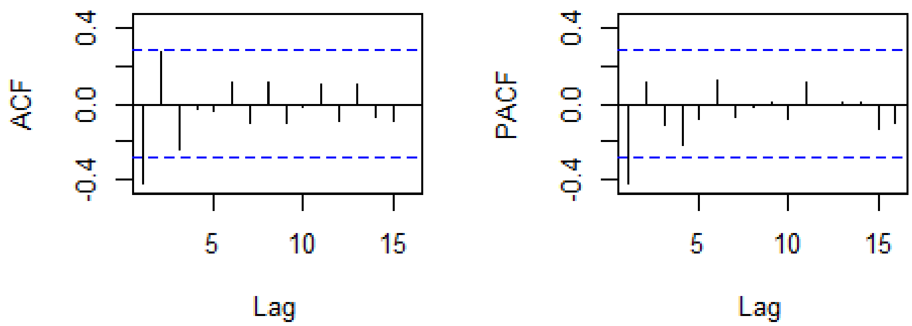
```
ts_diff <- diff(ts, differences = 1)
adf.test(ts_diff, alternative = "stationary")

## Warning in adf.test(ts_diff, alternative = "stationary"): p-value smaller
than
## printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  ts_diff
## Dickey-Fuller = -4.5317, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

After first-order differencing, the Augmented Dickey-Fuller test yields a p-value of 0.01 (< 0.05), indicating that the differenced time series is now stationary and suitable for ARIMA modeling.

ACF, PACF, and EACF

```r
tsdisplay(ts_diff)
```






```r
eacf(ts_diff, ar.max = 8, ma.max = 8)

## AR/MA
##   0 1 2 3 4 5 6 7 8
## 0 x o o o o o o o o
## 1 o o o o o o o o o
## 2 x o o o o o o o o
## 3 x x o o o o o o o
## 4 o x o o o o o o o
## 5 x o o o o o o o o
## 6 x o x o o o o o o
## 7 o o o o o o o o o
## 8 o o o o o o o o o
```

Model Fitting

Based on the patterns observed in the ACF, PACF, and EACF plots of the differenced series, several candidate ARIMA models were specified for comparison. The models include combinations of autoregressive (AR), moving average (MA), and seasonal components to capture the structure suggested by the correlation diagnostics.

```r
m1 <- Arima(ts, order = c(1, 1, 1), seasonal = c(1, 0, 0))
m2 <- Arima(ts, order = c(1, 1, 0), seasonal = c(1, 0, 0))
m3 <- Arima(ts, order = c(2, 0, 0), seasonal = c(1, 1, 0)) # Selected
m4 <- Arima(ts, order = c(1, 0, 2), seasonal = c(1, 1, 0))
m5 <- Arima(ts, order = c(3, 0, 1), seasonal = c(1, 1, 0))
```

Model Coefficient Analysis

To evaluate and interpret the estimated parameters of each ARIMA model, a custom function `printstatarima()` was created. This function displays:

- Estimated coefficients
- Standard errors (s.e.)
- t-statistics
- p-values (sign.)

The p-values help determine the statistical significance of each parameter in the model. Models with more significant coefficients ($p < 0.05$) are generally preferred.

```r
# Define and use printstatarima to summarize coefficients
printstatarima <- function(x, digits = 4, se = TRUE) {
  if (length(x$coef) > 0) {
    cat("\nCoefficients:\n")
    coef <- round(x$coef, digits = digits)
    if (se && nrow(x$var.coef)) {
      ses <- rep(0, length(coef))
      ses[x$mask] <- round(sqrt(diag(x$var.coef)), digits = digits)
      coef <- matrix(coef, 1, dimnames = list(NULL, names(coef)))
      coef <- rbind(coef, s.e. = ses)
      statt <- coef[1, ] / ses
      pval <- 2 * pt(abs(statt), df = length(x$residuals) - 1, lower.tail =
FALSE)
      coef <- rbind(coef, t = round(statt, digits = digits), sign. =
round(pval, digits = digits))
      coef <- t(coef)
    }
    print.default(coef, print.gap = 2)
  }
}
```

```r
printstatarima(m1)

##
## Coefficients:
##                  s.e.        t   sign.
## ar1   -0.6616  0.2271  -2.9133  0.0055
## ma1    0.3075  0.2881   1.0673  0.2913
## sar1  -0.0476  0.1495  -0.3184  0.7516

printstatarima(m2)

##
## Coefficients:
##                  s.e.        t   sign.
## ar1   -0.4115  0.1329  -3.0963  0.0033
## sar1  -0.0226  0.1444  -0.1565  0.8763

printstatarima(m3)

##
## Coefficients:
##                  s.e.        t   sign.
## ar1    0.5232  0.1533   3.4129  0.0013
## ar2    0.3918  0.1517   2.5827  0.0130
## sar1  -0.5943  0.1510  -3.9358  0.0003

printstatarima(m4)

##
## Coefficients:
##                  s.e.        t   sign.
## ar1    0.9162  0.0796  11.5101  0.0000
## ma1   -0.3476  0.1840  -1.8891  0.0651
## ma2    0.1902  0.1618   1.1755  0.2457
## sar1  -0.5938  0.1539  -3.8583  0.0003

printstatarima(m5)

##
## Coefficients:
##                  s.e.        t   sign.
## ar1    0.4363  1.0923   0.3994  0.6914
## ar2    0.5282  0.5818   0.9079  0.3686
## ar3   -0.0714  0.4709  -0.1516  0.8801
## ma1    0.1313  1.0993   0.1194  0.9054
## sar1  -0.5947  0.1551  -3.8343  0.0004
```

Among the five ARIMA models evaluated, Model m3: ARIMA(2,0,0)(1,1,0)[12] was selected as the best fit. This choice was based on the statistical significance of its parameters: all three coefficients—AR(1), AR(2), and SAR(1)—have p-values below 0.05, indicating that each term contributes meaningfully to the model. In contrast, the other models included one or more parameters with high p-values, suggesting less reliable estimation or overfitting.

Therefore, Model m3 is considered both parsimonious and statistically robust for forecasting purposes.

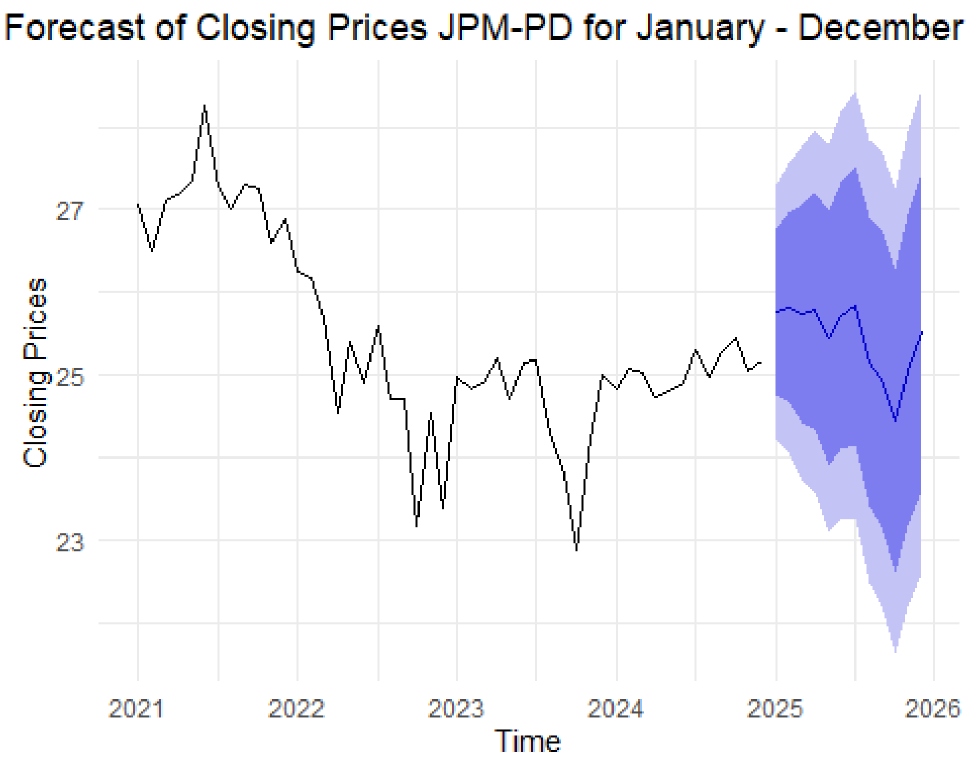
```
summary(m3)

## Series: ts
## ARIMA(2,0,0)(1,1,0)[12]
##
## Coefficients:
##          ar1     ar2     sar1
##       0.5232  0.3918  -0.5943
## s.e.  0.1533  0.1517   0.1510
##
## sigma^2 = 0.6165:  log likelihood = -44.01
## AIC=96.02   AICc=97.31   BIC=102.35
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE     MAPE     MASE
## Training set -0.0383304 0.6510458 0.4577063 -0.1835025 1.853268 0.385527
##                    ACF1
## Training set 0.05809413
```

The training set error metrics indicate acceptable performance: * RMSE: 0.6510 * MAE: 0.4577 * MAPE: 1.85% * MASE: 0.3855

These values suggest that the model has low prediction error and is well-calibrated. Furthermore, the first-lag autocorrelation of residuals (ACF1 = 0.058) is close to zero, implying no significant autocorrelation remains — a sign of model adequacy.

Forecasting

```
forecasted_values <- forecast(m3, h = 12)
autoplot(forecasted_values) +
  ggtitle("Forecast of Closing Prices JPM-PD for January - December 2025") +
  xlab("Time") +
  ylab("Closing Prices") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



Forecast Table

```r
forecast_dates <- seq(tail(index(close_xts), 1) + months(1), by = "month",
length.out = 12)

forecast_table <- data.frame(
  Period = forecast_dates,
  Forecast = as.numeric(forecasted_values$mean),
  Lower95 = as.numeric(forecasted_values$lower[, "95%"]),
  Upper95 = as.numeric(forecasted_values$upper[, "95%"])
)

forecast_table

##        Period Forecast  Lower95  Upper95
## 1  2025-01-01 25.76355 24.22460 27.30249
## 2  2025-02-01 25.81862 24.08173 27.55550
## 3  2025-03-01 25.74501 23.72856 27.76147
## 4  2025-04-01 25.77822 23.58934 27.96709
## 5  2025-05-01 25.44735 23.10034 27.79436
## 6  2025-06-01 25.70982 23.23759 28.18204
## 7  2025-07-01 25.84472 23.26460 28.42484
## 8  2025-08-01 25.16079 22.48982 27.83176
## 9  2025-09-01 24.95982 22.21073 27.70891
## 10 2025-10-01 24.44296 21.62676 27.25916
```

```
## 11 2025-11-01 25.05588 22.18163 27.93013
## 12 2025-12-01 25.52884 22.60427 28.45341
```

Export Results

```r
write.xlsx(forecast_table, file = "JPM-PD - Forecast (2025).xlsx")
```

Conclusion

The ARIMA(2,0,0)(1,1,0)[12] model was selected based on model diagnostics.
The forecasted JPM-PD stock prices for 2025 show a relatively stable pattern, with confidence intervals reflecting moderate uncertainty.