Time Series Final Assessment
Exercise 6: Forecasting of the USD/EUR Exchange Rate

Kacper Omieliańczyk

1 Exploratory Data Analysis

In this section we explore the USD/EUR exchange-rate data over 2008-2014 to understand its main features, guide our modeling choices, and lay the groundwork for one-step-ahead forecasting. We work in both price (level) space and log-return space.

1.1 Data Description

The data consist of 264 weekly observations of the USD/EUR exchange rate from early 2008 through mid-2014. We denote by p_t the rate on date t. After parsing and cleaning, our working data frame has two columns:

- `Date`: the week's ending date.
- `Price`: the USD/EUR rate p_t.

1.2 Descriptive Statistics of Prices

Table 1 shows, for each calendar year, the sample size, mean, standard deviation, minimum, maximum, skewness and excess kurtosis of the weekly closing rate.

Table 1: Annual summary statistics of USD/EUR (Price)

Year	n	Mean	SD	Min	Max	Skewness	Kurtosis
2009	32	0.694	0.0183	0.666	0.729	0.007	-1.37
2010	52	0.754	0.0351	0.691	0.832	0.302	-0.79
2011	52	0.718	0.0234	0.680	0.766	0.331	-1.06
2012	53	0.778	0.0196	0.748	0.821	0.558	-0.86
2013	52	0.753	0.0145	0.727	0.777	-0.116	-1.14
2014	19	0.728	0.0056	0.720	0.738	0.346	-1.33

Insights.

- The rate dipped during the 2008-2009 crisis (2009 mean 0.694) then climbed through 2012 (mean 0.778).
- Volatility (SD) peaked in 2010 (0.035) and declined thereafter.
- All years exhibit slight negative excess kurtosis (platykurtic) and mild skew.

1.3 Time-Series Plots

Figure 1 shows the weekly closing rate over the full sample.

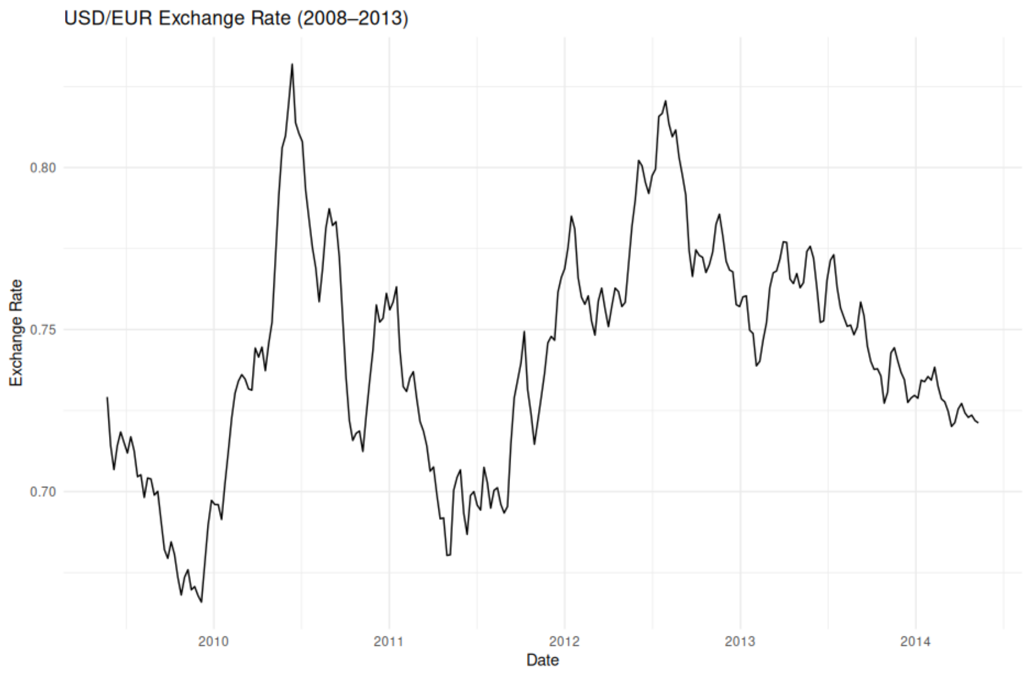

Figure 1: USD/EUR exchange rate, weekly closes (2008-2014).

Notable features:

- A trough near mid-2009 (≈ 0.667), and a peak in late 2010 (≈ 0.83).
- A secondary peak around mid-2012 (≈ 0.82).
- A gradual decline after 2012 toward ≈ 0.72 by mid-2014.

Figure 2 overlays a 30-week moving average to highlight medium-term trends.

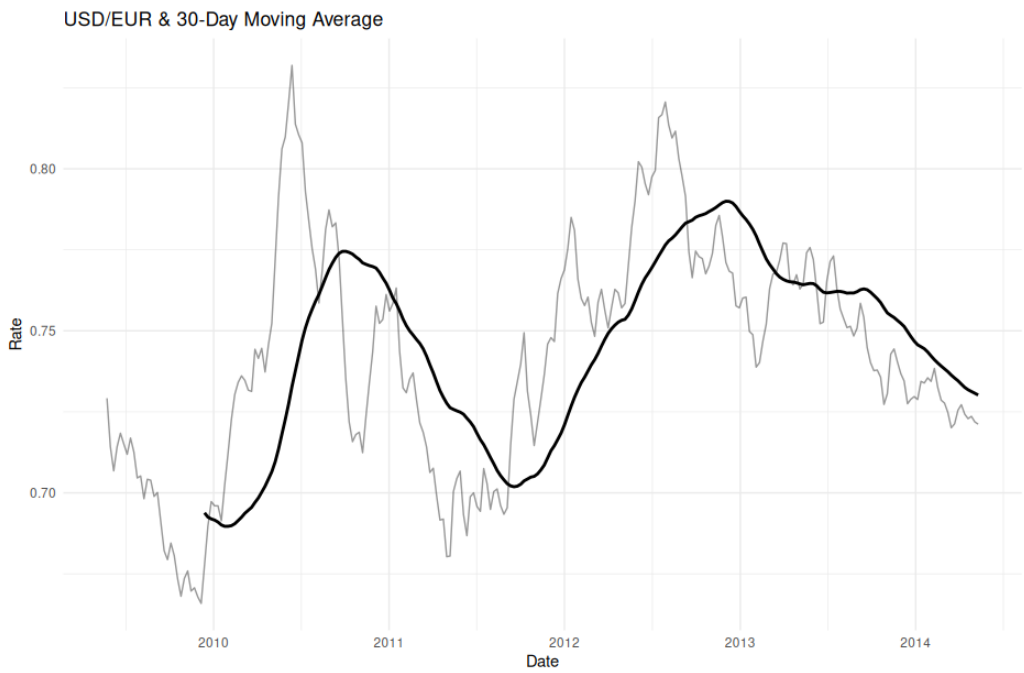

Figure 2: USD/EUR and its 30-week moving average.

1.4 Distribution by Year

To compare variability and location year-by-year, Figure 3 presents boxplots of the weekly rate by calendar year.

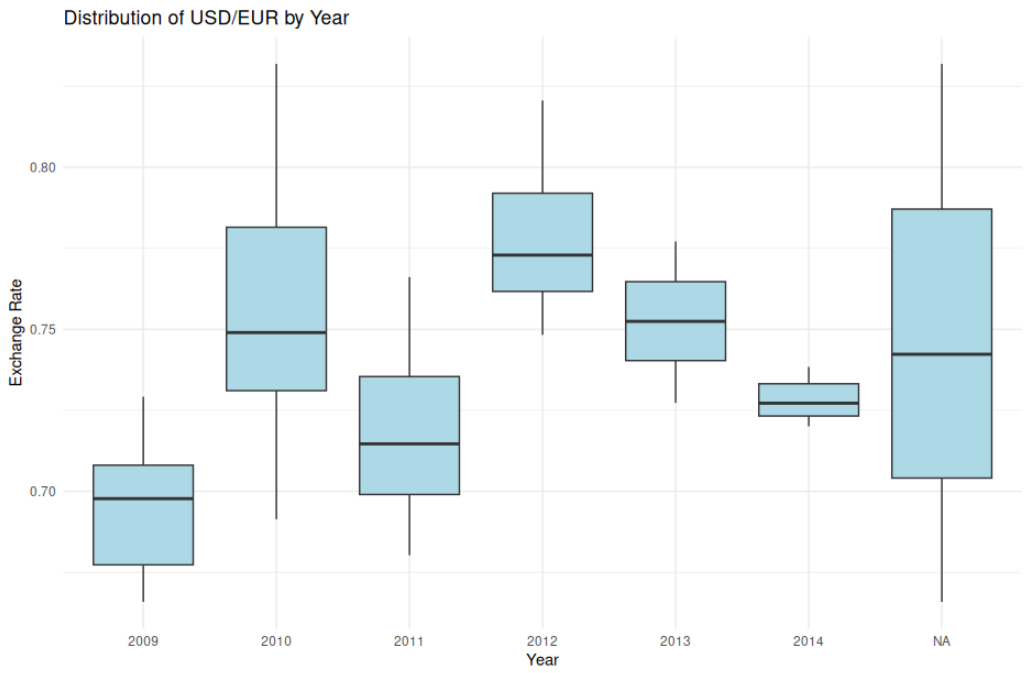

Figure 3: Distribution of USD/EUR by year (2009-2014).

1.5 Log-Return Analysis

We define the log-return $r_t = \ln(p_t) - \ln(p_{t-1})$. Table 2 reports annual summaries.

Table 2: Annual summary statistics of USD/EUR log-returns

Year	Mean	SD	Min	Max
2009	-0.00144	0.00939	-0.0206	0.0177
2010	0.00169	0.0128	-0.0249	0.0255
2011	0.00012	0.0121	-0.0259	0.0288
2012	-0.00022	0.00912	-0.0218	0.0202
2013	-0.00071	0.00791	-0.0147	0.0166
2014	-0.00062	0.00416	-0.0080	0.0077

Figure 4 shows the return series and its distribution.

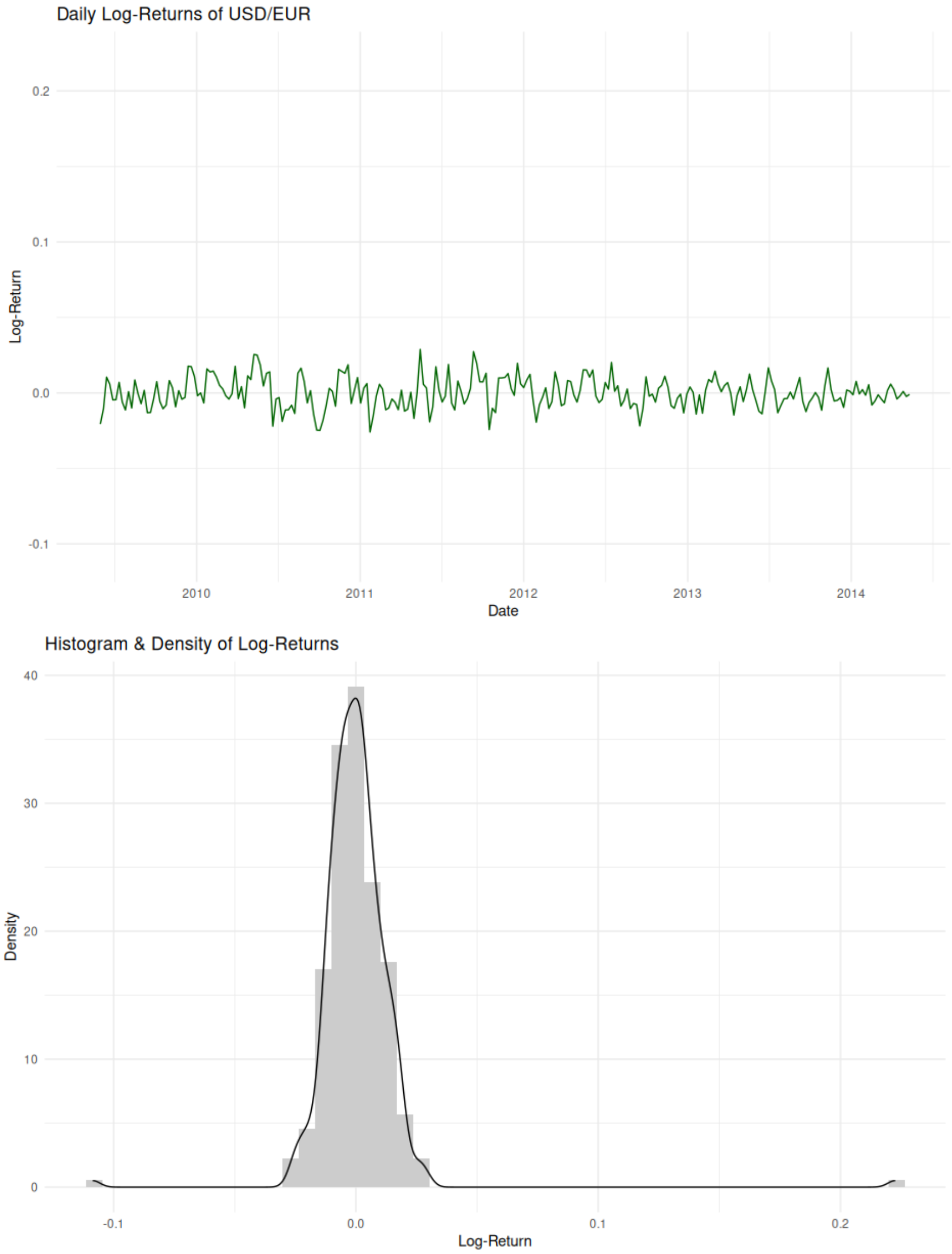

Figure 4: Top: log-returns over time. Bottom: histogram & kernel density.

Insights.

- Returns are tightly centered around zero, with SD decreasing from 2010 onward.
- Distribution is mildly leptokurtic with occasional large moves (—r_t— up to 2–3%).

1.6 Stationarity and Autocorrelation

Augmented Dickey-Fuller tests indicate

$$\text{ADF}(\{p_t\})\ p\text{-value} = 0.338 \quad \text{(non-stationary)}, \quad \text{ADF}(\{r_t\})\ p\text{-value} = 0.01 \quad \text{(stationary)}.$$

Thus we will model in log-return space.

Figure 5 displays ACF and PACF of $\{r_t\}$ up to lag 25.

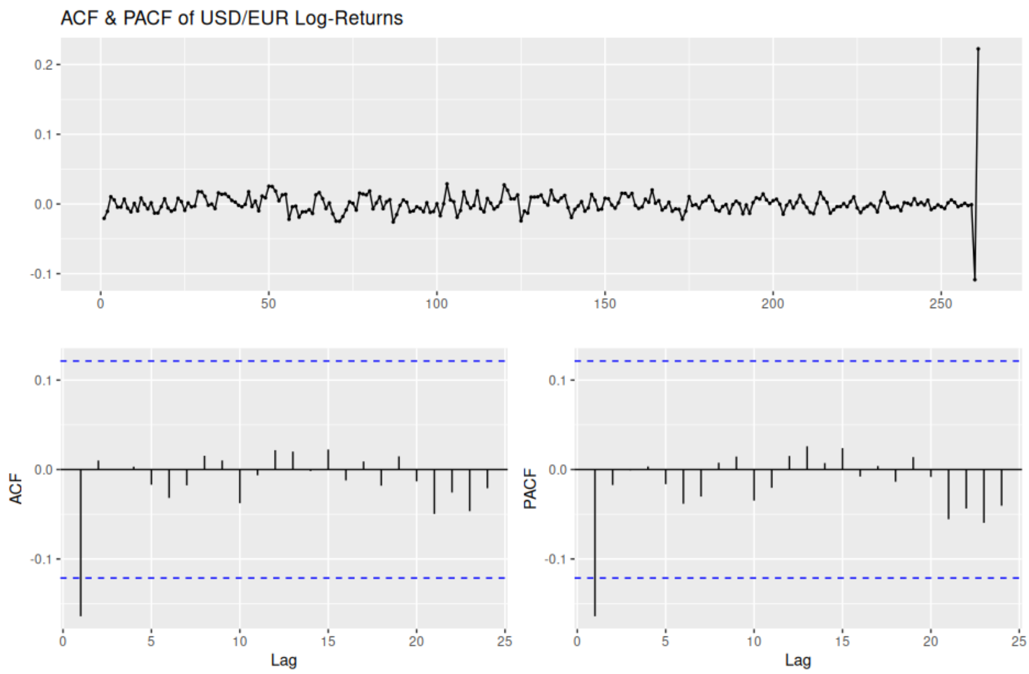

Figure 5: ACF (left) and PACF (right) of USD/EUR log-returns.

The first-order autocorrelation is slightly negative (≈ -0.15), with most higher lags within the 95% bounds.

1.7 Conclusions and Next Steps

Based on the EDA:

- The level series is non-stationary; log-returns are stationary.
- Returns exhibit low autocorrelation beyond lag 1, suggesting an AR(1) or simple EWMA could be competitive.
- Volatility clusters in 2009-2011 but calms thereafter; a simple GARCH(1,1) may yield marginal improvements.
- We will proceed to one-step-ahead forecasts in return space, benchmarked against naive, SES, ARIMA(1,0,1), and (optionally) GARCH-type models, comparing by sum of squared errors.

This completes our exploratory analysis and paves the way for the full forecasting exercise.

1.8 R code

This is the used EDA R code:

```
library(tidyverse)
library(lubridate)
library(forecast)
library(tseries)
library(gridExtra)
library(zoo)
library(e1071)

df <- read_csv("USDEUR.csv", col_types = cols(.default = col_character()))
  %>%
  select(`End Date`, `USD/EUR`) %>%
  rename(
    Date  = `End Date`,
    Price = `USD/EUR`
  ) %>%
```

```r
  mutate(
    Date  = ymd(Date),
    Price = as.double(Price)
  ) %>%
  arrange(Date)

glimpse(df)
head(df)
summary(df$Price)

p1 <- ggplot(df, aes(x = Date, y = Price)) +
  geom_line() +
  labs(title = "USD/EUR␣Exchange␣Rate␣(2008-2013)",
       x = "Date", y = "Exchange␣Rate") +
  theme_minimal()

df <- df %>%
  mutate(Price_MA30 = rollmean(Price, k = 30, fill = NA, align = "right"))

p2 <- ggplot(df, aes(x = Date)) +
  geom_line(aes(y = Price), alpha = 0.4) +
  geom_line(aes(y = Price_MA30), size = 1, linetype = "solid") +
  labs(title = "USD/EUR␣&␣30-Day␣Moving␣Average",
       x = "Date", y = "Rate") +
  theme_minimal()
grid.arrange(p1, p2, ncol = 1)

df_summary <- df %>%
  mutate(Year = year(Date)) %>%
  group_by(Year) %>%
  summarize(
    n        = n(),
    Mean     = mean(Price, na.rm = TRUE),
    SD       = sd(Price, na.rm = TRUE),
    Min      = min(Price, na.rm = TRUE),
    Max      = max(Price, na.rm = TRUE),
    Skewness = skewness(Price, na.rm = TRUE),
    Kurtosis = kurtosis(Price, na.rm = TRUE)
  )
print(df_summary)

p3 <- ggplot(df %>% mutate(Year = factor(year(Date))),
             aes(x = Year, y = Price)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Distribution␣of␣USD/EUR␣by␣Year",
       x = "Year", y = "Exchange␣Rate") +
  theme_minimal()
print(p3)

df <- df %>%
  mutate(
    LogPrice = log(Price),
    Return   = LogPrice - lag(LogPrice)
  )

df_return_summary <- df %>%
  filter(!is.na(Return)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Year) %>%
  summarize(
    MeanRet = mean(Return, na.rm = TRUE),
    SDRet   = sd(Return, na.rm = TRUE),
    MinRet  = min(Return, na.rm = TRUE),
```

```r
    MaxRet   = max(Return, na.rm = TRUE)
  )
print(df_return_summary)

p4 <- ggplot(df, aes(x = Date, y = Return)) +
  geom_line(color = "darkgreen") +
  labs(title = "Daily Log-Returns of USD/EUR",
       x = "Date", y = "Log-Return") +
  theme_minimal()

p5 <- ggplot(df %>% filter(!is.na(Return)), aes(x = Return)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "gray80") +
  geom_density() +
  labs(title = "Histogram & Density of Log-Returns",
       x = "Log-Return", y = "Density") +
  theme_minimal()

print(p5)
grid.arrange(p4, p5, ncol = 1)

adf_price  <- adf.test(na.omit(df$Price), alternative = "stationary")
adf_return <- adf.test(na.omit(df$Return), alternative = "stationary")
print(adf_price)
print(adf_return)

ggtsdisplay(na.omit(df$Return),
            main = "ACF & PACF of USD/EUR Log-Returns")
```

Listing 1: EDA for USD/EUR

2 One-Step-Ahead Forecasting

In this section we describe our recursive forecasting setup, the five candidate models we evaluated (Random-Walk, SES, Holt's linear, ARIMA, and GARCH(1,1)), and the out-of-sample results on a one-year holdout. We work in log-return space, back-transform to prices, and compare by the sum of squared forecast errors (SSE).

2.1 Forecasting Framework

We choose two consecutive years, 2010 as the *training* (burn-in) period and 2011 as the *testing* (forecasting) period. Let p_t be the observed USD/EUR rate on date t, and

$$r_t \;=\; \ln(p_t) \;-\; \ln(p_{t-1})$$

the log-return. For each week t in 2011 we:

1. Fit each candidate model on all returns $\{r_1, \ldots, r_{t-1}\}$ up to the previous date.
2. Produce a one-step-ahead forecast for the return, $\hat{r}_t$.
3. Back-transform to a price forecast

$$\hat{p}_t \;=\; p_{t-1}\,\exp\!\big(\hat{r}_t\big).$$

We then compute the loss

$$\text{SSE} \;=\; \sum_{t \in \text{test}} (p_t - \hat{p}_t)^2$$

for each model over all 52 weeks in 2011.

2.2 Candidate Models

We considered five methods, all fitted to the log-return series:

Random-Walk (Naive)
$\hat{r}_t = 0$. The standard naive benchmark.

Simple Exponential Smoothing (SES)
An ETS(A, N, N) model on $\{r_t\}$.

Holt's Linear Method
An ETS(A, A, N) model capturing trend in returns.

ARIMA
The ARIMA$(p, 0, q)$ automatically selected by `auto.arima()`.

GARCH(1,1)
A GARCH(1,1) model on $\{r_t\}$ with constant mean, capturing volatility clustering.

2.3 Out-of-Sample Results

Table 3 reports the sum of squared errors for each model on the 2011 holdout:

Table 3: Sum of squared forecast errors (SSE) for 2011

Model	SSE
Random-Walk (RW)	0.00388
SES	0.00451
Holt's Linear	0.00436
ARIMA	**0.00379**
GARCH(1,1)	0.00388

The ARIMA model achieved the lowest SSE, with the GARCH(1,1) model matching the naive Random-Walk benchmark.

Figure 6 overlays the actual USD/EUR rates in 2011 against the ARIMA forecasts:

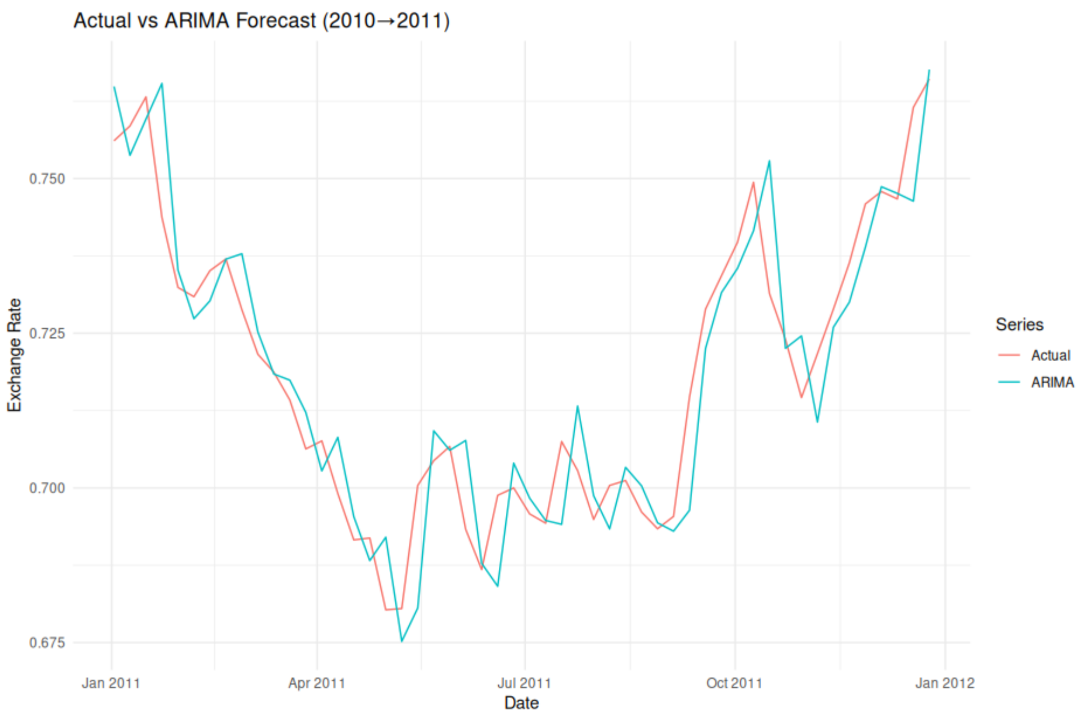

Figure 6: Actual vs. ARIMA one-step-ahead forecasts (training on 2010, testing on 2011).

2.4 Discussion

(a) *Price vs. return space:* The level series p_t is non-stationary (ADF $p \approx 0.34$), whereas the log-returns r_t are stationary (ADF $p \approx 0.01$). Modeling in return space stabilizes variance and satisfies stationarity requirements; forecasts are then back-transformed to price for evaluation.

(b) *Parsimony vs. complexity:* Among the five methods, the low-order ARIMA provided the best out-of-sample accuracy. The GARCH(1,1) model—despite explicitly modeling volatility clustering—offered no improvement over the Random-Walk. This highlights that added complexity (more parameters) does not guarantee better point forecasts.

(c) *Statistical vs. economic significance:* The ARIMA forecast reduced SSE by about 2.3% relative to the Random-Walk. In FX markets, even such a small predictive edge can be operationally valuable, though transaction costs and execution risk may erode gains.

(d) *Robustness and future work:* We might extend the framework by exploring time-varying parameter models, regime-switching, or machine learning methods with engineered features. Incorporating macroeconomic predictors or news-based sentiment could further enhance forecast performance.

2.5 Conclusion

Across all five candidate models, a simple ARIMA on log-returns delivered the lowest SSE and thus the most accurate one-step-ahead forecasts for 2011. Volatility modeling via GARCH(1,1) did not improve point-forecast performance, reinforcing that parsimony often prevails in exchange-rate prediction. Future enhancements could focus on hybrid or regime-aware approaches to capture market dynamics beyond linear autocorrelation.

2.6 R code

This is the used R code for forecasting:

```r
library(tidyverse)
library(lubridate)
library(forecast)
library(rugarch)

train_year <- 2010
test_year  <- 2011

df <- read_csv("USDEUR.csv",
               col_types = cols(.default = col_character())) %>%
  select(`End Date`, `USD/EUR`) %>%
  rename(Date = `End Date`, Price = `USD/EUR`) %>%
  mutate(
    Date  = ymd(Date),
    Price = as.double(Price)
  ) %>%
  arrange(Date) %>%
  mutate(
    LogPrice = log(Price),
    Return   = LogPrice - lag(LogPrice)
  )

df <- df %>% filter(!is.na(Return))

train_end    <- max(df$Date[year(df$Date) == train_year])
test_dates   <- df %>% filter(year(Date) == test_year) %>% pull(Date)
n_test       <- length(test_dates)

fc <- tibble(
  Date   = test_dates,
  Actual = df %>% filter(Date %in% test_dates) %>% pull(Price),
```

```r
  RW      = NA_real_,
  SES     = NA_real_,
  HOLT    = NA_real_,
  ARIMA   = NA_real_,
  GARCH = NA_real_     # if you implement GARCH
)

for(i in seq_along(test_dates)) {
  today    <- test_dates[i]
  hist     <- df %>% filter(Date < today)
  rets     <- hist$Return
  last_pr <- hist$Price %>% last()

  # Random-walk (naive): forecast = 0 return
  rw_ret    <- 0

  # Simple Exponential Smoothing
  ses_ret   <- ses(rets, h = 1)$mean

  # Holt's linear method
  holt_ret <- holt(rets, h = 1)$mean

  # ARIMA via auto.arima()
  arima_ret <- forecast(auto.arima(rets), h = 1)$mean

  #  GARCH(1,1) on returns
  spec     <- ugarchspec(
               mean.model      = list(armaOrder=c(0,0)),
                variance.model = list(garchOrder=c(1,1)),
                distribution.model = "norm"
             )
  fit_g    <- ugarchfit(spec, rets, solver = "hybrid")
  garch_ret <- as.numeric(ugarchforecast(fit_g, n.ahead = 1)@forecast$
      seriesFor)

  ## back-transform to prices
  #    \hat p_t = p_{t-1} * exp(\hat r_t)
  fc$RW[i]     <- last_pr * exp(rw_ret)
  fc$SES[i]    <- last_pr * exp(ses_ret)
  fc$HOLT[i]   <- last_pr * exp(holt_ret)
  fc$ARIMA[i] <- last_pr * exp(arima_ret)
  fc$GARCH[i] <- last_pr * exp(garch_ret)
}

sse <- fc %>%
  summarize(across(RW:GARCH,
                   ~ sum((Actual - .x)^2),
                   .names = "SSE_{col}"))
print(sse)

best_sse_col <- names(sse)[ which.min(unlist(sse)) ]
best_model    <- sub("^SSE_", "", best_sse_col)
message("Best model: ", best_model)

library(ggplot2)
p <- fc %>%
  select(Date, Actual, all_of(best_model)) %>%
  pivot_longer(-Date, names_to = "Series", values_to = "Price") %>%
  ggplot(aes(x = Date, y = Price, color = Series)) +
  geom_line() +
  labs(title = sprintf("Actual vs %s Forecast (%d->%d)",
                       best_model, train_year, test_year),
       x = "Date", y = "Exchange Rate") +
```

```
  theme_minimal()

print(p)
ggsave(sprintf("best_forecast_%s_%d_%d.png",
               tolower(best_model), train_year, test_year),
       plot = p, width = 8, height = 4)
```

Listing 2: Forecast for USD/EUR exchange rate