# Lecture 1
# Time Series and Forecasting

# What is “Time Series”?

Data sampled *in order, over time*.

Regular sampling:

- annual: once per year
- monthly: once per month
- quarterly: once every three months
- weekly, daily, etc

Irregular sampling (not covered)

# What is “Time Series”? Example

Monthly time series of gold price (USD/oz):

```
         Jan     Feb     Mar     Apr
2024 2053.25 2048.05 2214.35 2307.00
         May     Jun     Jul     Aug
2024 2348.25 2330.90 2426.30 2513.35
         Sep     Oct     Nov     Dec
2024 2629.95 2734.15 2651.05 2609.10
```

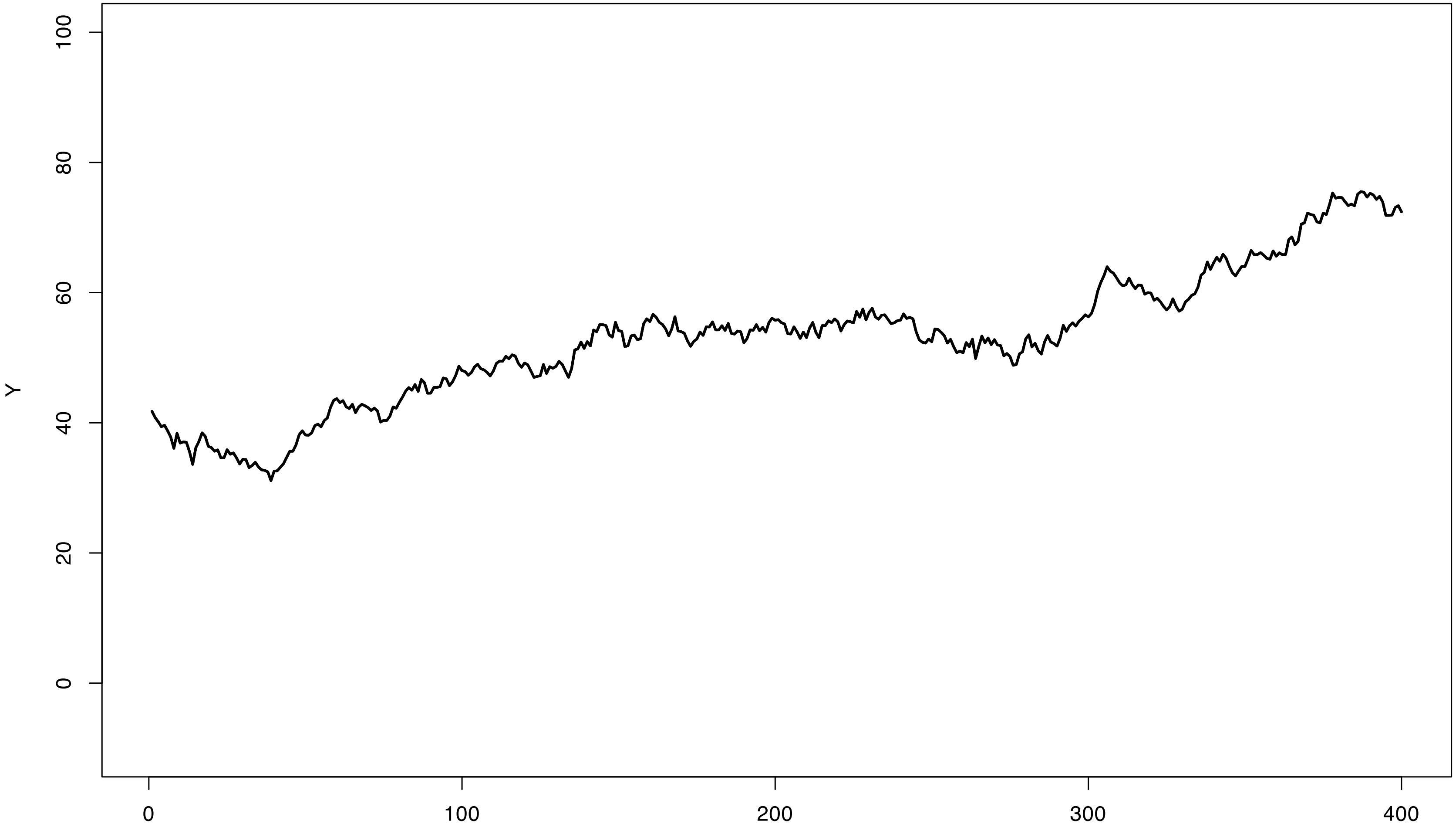
# What is “Time Series”? Example

Time series plot:

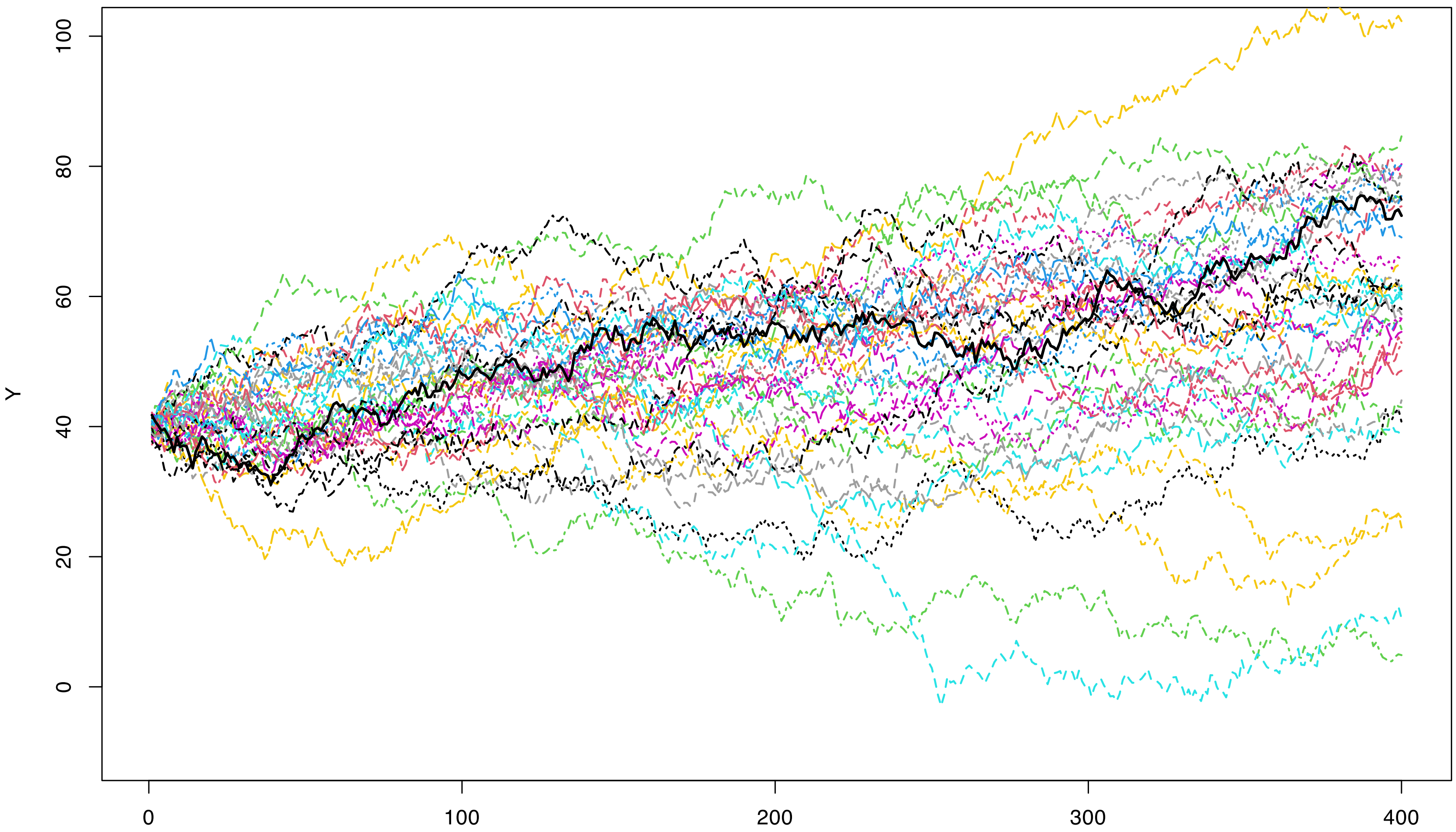
# What is time series “sampling”?

- Concept of a “representative sample” from a “population” is unclear for time series.
- “Sample path”: a time series plot
- “Population” : imagine all possible sample paths that might have occurred if time were re-run repeatedly.

# What is time series "sampling"?



# What is time series “sampling”?



## Dependence over time

Cross section samples often have observations drawn *independently* from a population.

Time series exhibit *dependence* across time.

(Both a challenge and an opportunity!)

## Dependence over time

Cross section samples often have observations drawn *independently* from a population.

Time series exhibit *dependence* across time.

Eg. dependence between events of

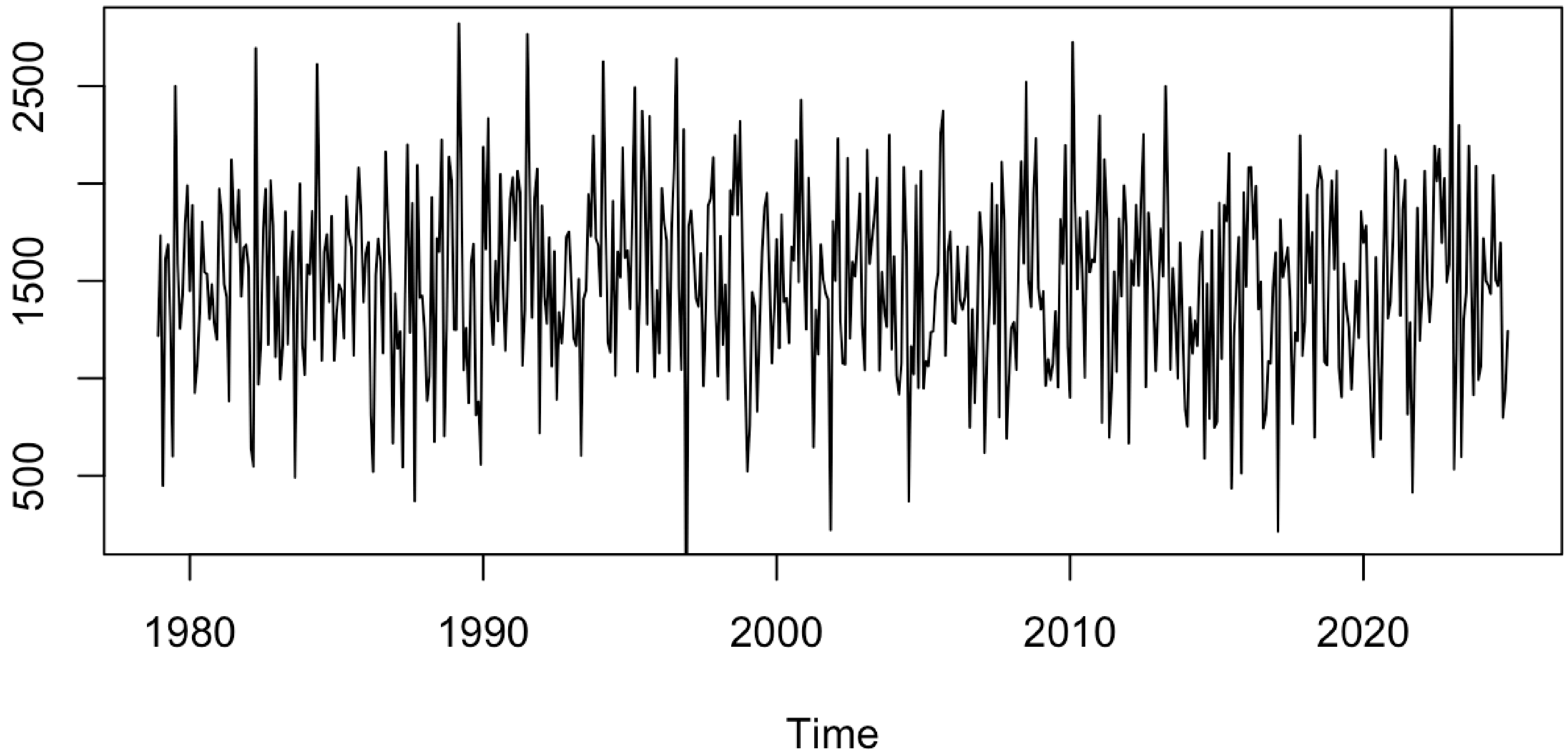
- today and tomorrow
- this month and next month
- this year and next year
- etc

# Dependence over time, example

Dependence can produce a “smooth” plot:

# Lack of Dependence over time, example

A more "random" looking plot:



## Dependence over time

Dependence can induce patterns.

Eg:

- smooth swings
- trend (upwards or downwards) over time
- oscillations
- seasonality
- etc

## Dependence over time

Dependence can induce patterns.

Patterns can be *modelled* to produce forecasts.

*Models* include:

- regression
- ARIMA
- Vector Autoregression
- etc

## Forecasting Steps

- Define forecasting problem
- Collect relevant data
- Estimate plausible models for the data
- Evaluate in-sample fit
- Calculate and evaluate forecasts

# Application: Forecasting Retail Sales

## Example: Retail Sales

- Retail sales are considered a key indicator of the health of an economy.
- Data from Australian Bureau of Statistics, sampled monthly, April 1982 to September 2024.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Date | Food | HouseholdGoods | Clothing | DepartmentStores | Other | Restaurants | Total |
| 2 | Apr-82 | 1162.6 | 592.3 | 359.9 | 460.1 | 479.1 | 342.4 | 3396.4 |
| 3 | May-82 | 1150.9 | 629.6 | 386.6 | 502.6 | 486.1 | 342.1 | 3497.9 |
| 4 | Jun-82 | 1160 | 607.4 | 350.5 | 443.8 | 467.5 | 328.7 | 3357.8 |
| 5 | Jul-82 | 1206.4 | 632.4 | 359.3 | 459.1 | 491.1 | 338.5 | 3486.8 |
| 6 | Aug-82 | 1152.5 | 622.6 | 325.2 | 438.4 | 485.7 | 331.5 | 3355.9 |
| 7 | Sep-82 | 1189.1 | 622 | 346.3 | 465.1 | 489.9 | 341.9 | 3454.3 |
| 8 | Oct-82 | 1247.4 | 637.8 | 354.2 | 452.7 | 500.9 | 358.4 | 3551.5 |
| 9 | Nov-82 | 1280.7 | 717.2 | 403.9 | 522.9 | 531.1 | 374.7 | 3830.5 |
| 10 | Dec-82 | 1483.7 | 1077 | 571.4 | 889.3 | 725.2 | 433.1 | 5179.7 |
| 11 | Jan-83 | 1202.8 | 615.9 | 326.5 | 379.2 | 491.4 | 368.6 | 3384.5 |
| 12 | Feb-83 | 1224.2 | 621.7 | 298.7 | 378 | 499.1 | 348.1 | 3369.8 |
| 13 | Mar-83 | 1332.4 | 719.1 | 377.2 | 472.1 | 538.7 | 365.8 | 3805.3 |
| 14 | Apr-83 | 1270.9 | 633.4 | 412.5 | 503.4 | 493.8 | 351.1 | 3665.1 |
| 15 | May-83 | 1251.7 | 697.9 | 418.5 | 510.6 | 520.8 | 360.5 | 3760 |
| 16 | Jun-83 | 1269.3 | 661.8 | 383 | 462.4 | 507.3 | 347.1 | 3630.8 |
| 17 | Jul-83 | 1292.8 | 656.9 | 384 | 468.3 | 519.9 | 364.5 | 3686.5 |

# The ts object

```r
dt <- read.csv("RetailSales.csv")
print(head(dt[c("Food", "Clothing", "Total"
```

```
    Food Clothing  Total
1 1162.6    359.9 3396.4
2 1150.9    386.6 3497.9
3 1160.0    350.5 3357.8
4 1206.4    359.3 3486.8
5 1152.5    325.2 3355.9
6 1189.1    346.3 3454.3
```

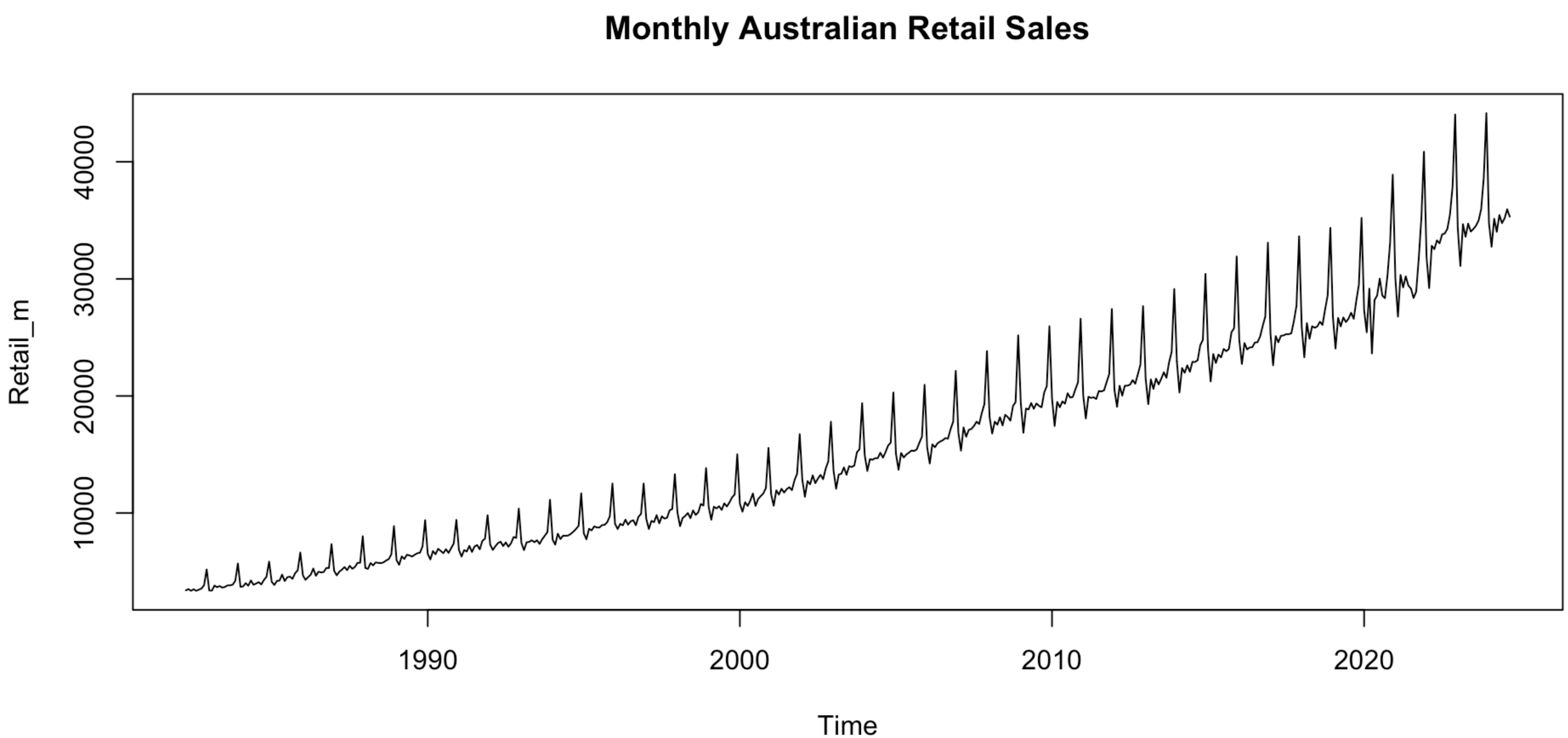
# The ts object

```r
Retail_m <- ts(dt$Total, frequency=12,
               start=c(1982,4), end=c(2024,9))
print(Retail_m)
```

```
        Jan    Feb    Mar    Apr    May
1982                        3396.4 3497.9
1983 3384.5 3369.8 3805.3 3665.1 3760.0
1984 3698.5 3733.2 4010.9 3788.4 4242.3
1985 4130.0 3856.8 4196.9 4228.9 4739.1
1986 4674.9 4294.1 4528.0 4729.0 5261.9
1987 5065.3 4679.8 4993.0 5166.0 5386.3
1988 5304.4 5226.8 5729.7 5527.3 5787.8
```

# The ts object

```r
Retail_m <- ts(dt$Total, frequency=12,
               start=c(1982,4), end=c(2024,9))
plot(Retail_m, main="Monthly Australian Ret
```



## aggregate : change frequency

```r
Retail_m <- ts(dt$Total, frequency=12,
                        start=c(1982,4), end=c(2024,
Retail_q <- aggregate(Retail_m, FUN=sum,
                               nfrequency=4)
print(Retail_q)
```

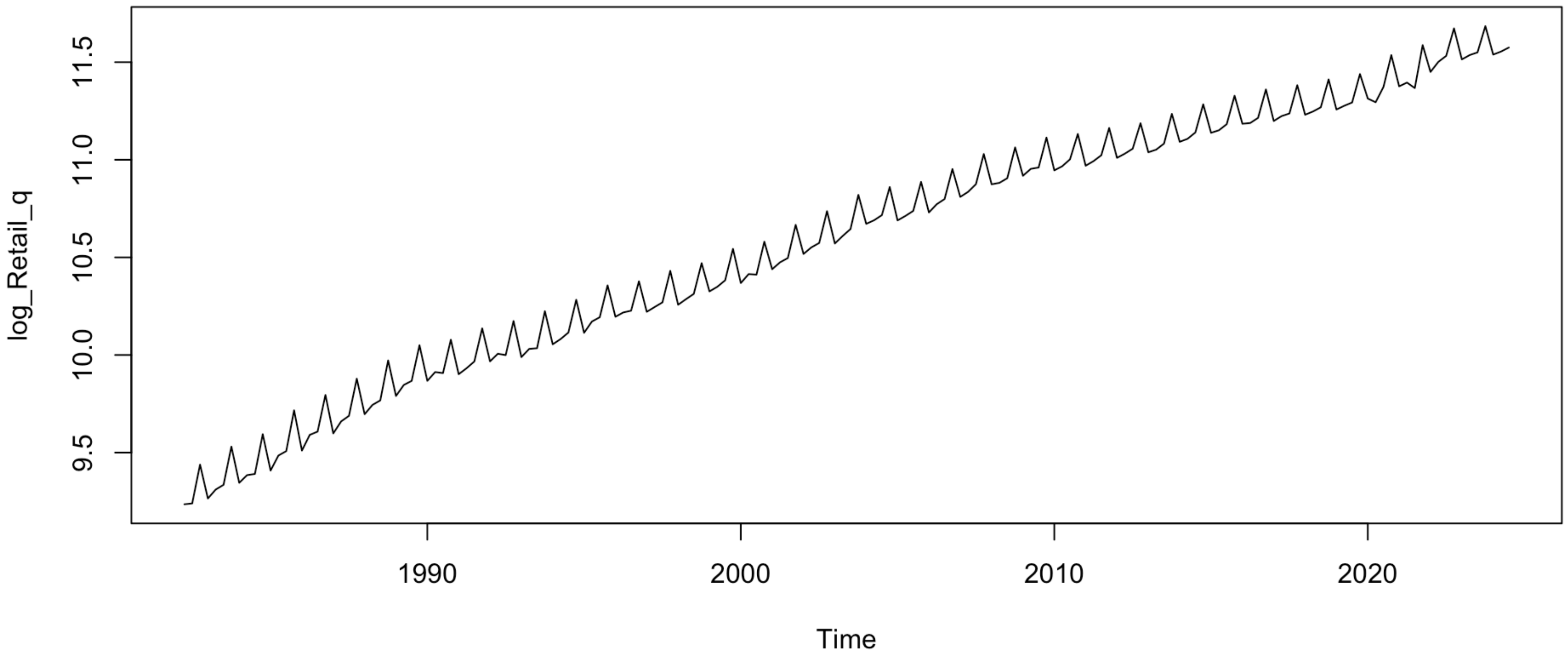
| | Qtr1 | Qtr2 | Qtr3 | Qtr4 |
|---|---|---|---|---|
| 1982 | | 10252.1 | 10297.0 | 12561.7 |
| 1983 | 10559.6 | 11055.9 | 11326.2 | 13774.5 |
| 1984 | 11442.6 | 11902.8 | 11975.5 | 14678.6 |
| 1985 | 12183.7 | 13158.3 | 13461.2 | 16590.9 |
| 1986 | 13497.0 | 14628.7 | 14875.5 | 17948.6 |
| 1987 | 14738.1 | 15671.2 | 16126.4 | 19511.8 |

## Quarters

- Qtr1 : Jan, Feb, Mar (“March quarter”)
- Qtr2 : Apr, May, Jun (“June quarter”)
- Qtr3 : Jul, Aug, Sep (“September quarter”)
- Qtr4 : Oct, Nov, Dec (“December quarter”)

# log : a “stabilising” transformation

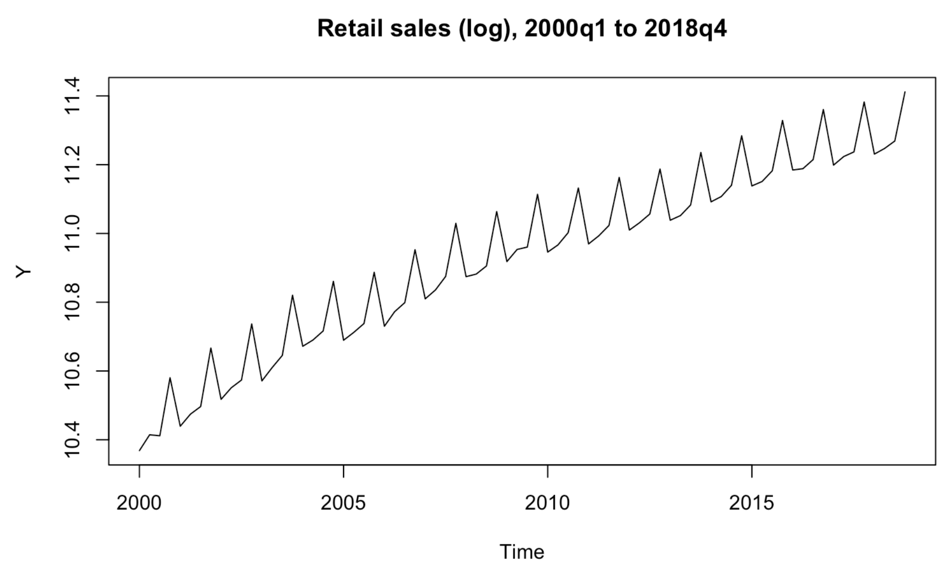
```r
log_Retail_q <- log(Retail_q)
plot(log_Retail_q)
```



# Sample period

We will use observations from 2000 to 2018 to forecast retail sales for 2019.

```r
Y <- window(log_Retail_q, start=c(2000,1),
                          end=c(2018,4))
plot(Y, main="Retail sales (log), 2000q1 to
```



# Sample period

We will use observations from 2000 to 2018 to forecast retail sales for 2019.

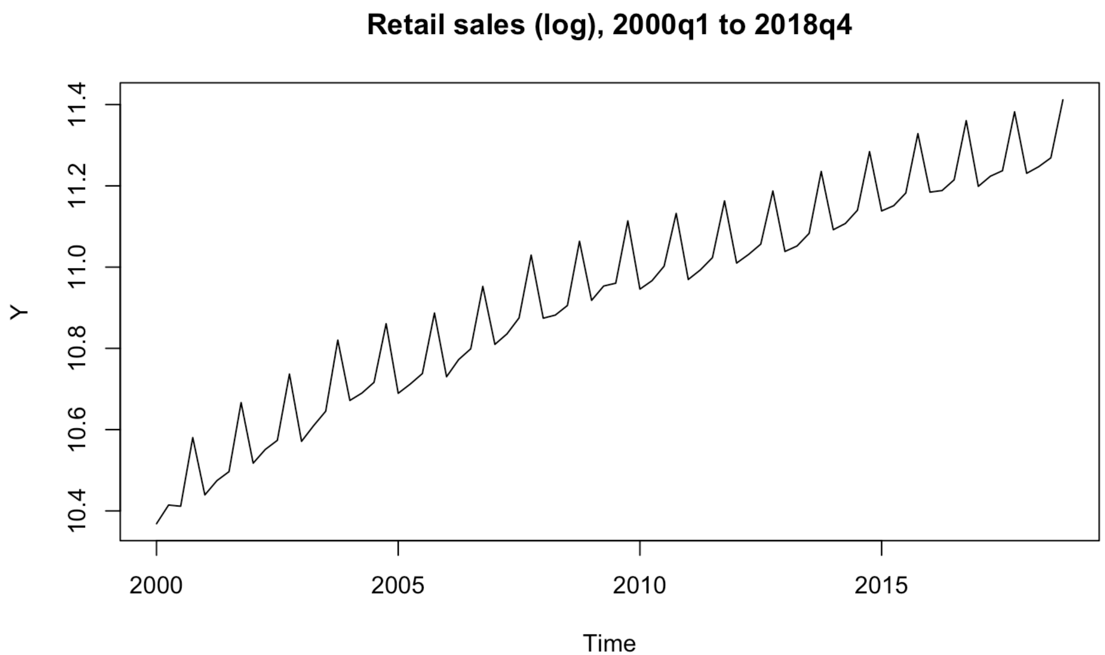
Why?

- Pre-covid time
- 2019 data is available for *forecast evaluation*

## Forecasting Steps

- Define forecasting problem
  Forecast retail sales for the four quarters of 2019 using data available up to end of 2018.
- Collect relevant data
  We have quarterly retail sales for 2000-2018.
- Estimate plausible models for the data
- Evaluate in-sample fit
- Calculate and evaluate forecasts

# Plausible models?




Two obvious features:

- increasing time trend
- seasonal pattern

## Regression Model 1: Time trend

Consider a regression model with a *time trend* :

$$Y_t = \beta_0 + \beta_1 \mathrm{Time}_t + \varepsilon_t$$

where $\mathrm{Time}_t$ is the date of the observation.

$\beta_1 > 0 : Y_t$ increases by $\beta_1$ on average each time.

$\beta_1 < 0 : Y_t$ decreases by $\beta_1$ on average each time.

# Regression Model 1: Time trend

time : gives dates for a ts object

```r
Time_t <- time(Y)
print(Time_t)
```

```
        Qtr1    Qtr2    Qtr3    Qtr4
2000 2000.00 2000.25 2000.50 2000.75
2001 2001.00 2001.25 2001.50 2001.75
2002 2002.00 2002.25 2002.50 2002.75
2003 2003.00 2003.25 2003.50 2003.75
2004 2004.00 2004.25 2004.50 2004.75
2005 2005.00 2005.25 2005.50 2005.75
2006 2006.00 2006.25 2006.50 2006.75
```

# Regression Model 1: Time trend

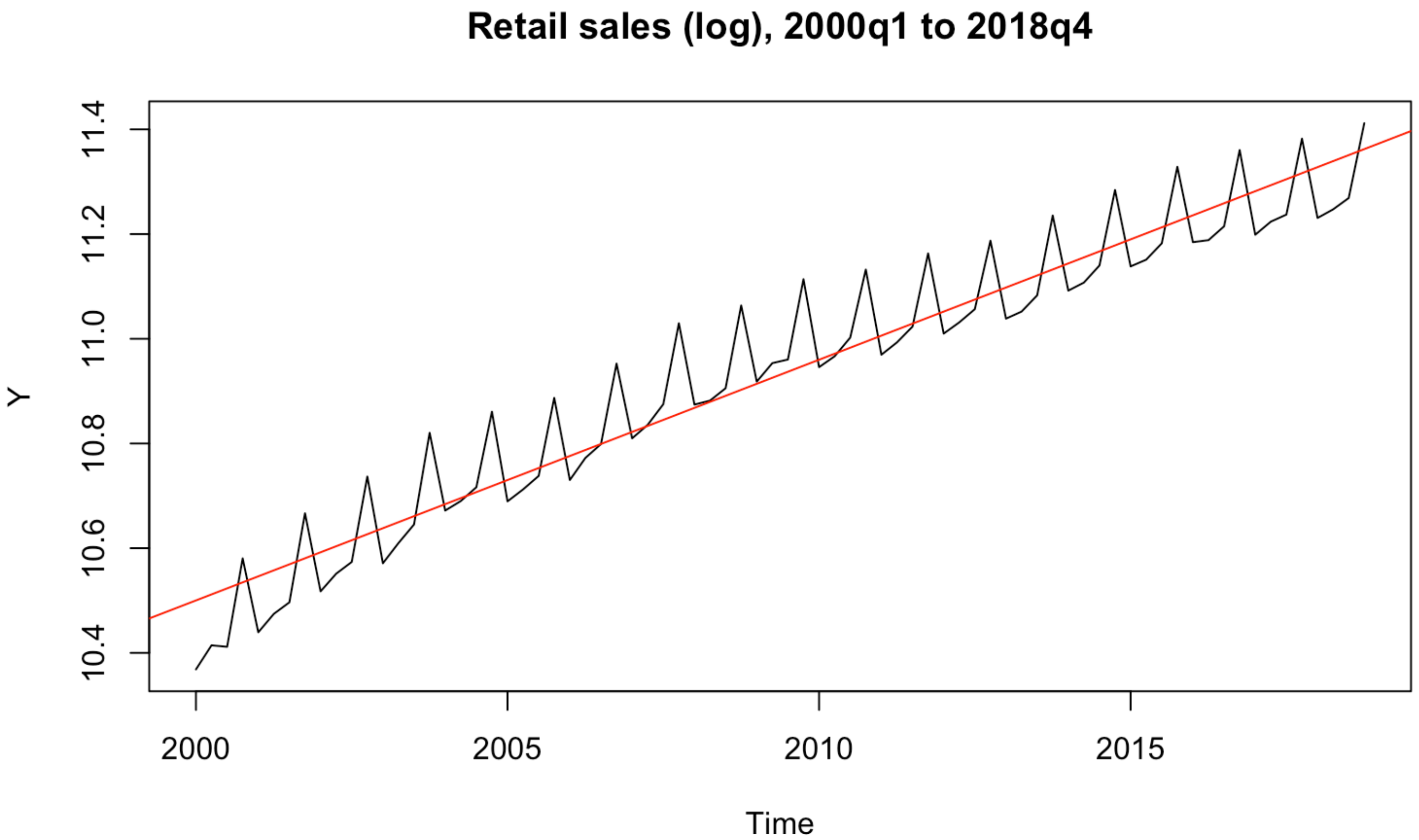
```r
eq <- lm(Y~Time_t)
print(eq)
```

```
Call:
lm(formula = Y ~ Time_t)

Coefficients:
(Intercept)       Time_t
  -81.46907      0.04598
```

$$\widehat{E}(Y_t) = -81.46907 + 0.04598\,\mathrm{Time}_t$$

# Regression Model 1: Time trend

```
plot(Y, main="Retail sales (log), 2000q1 to
abline(eq, col="red")
```



# Regression Model 1: Time trend

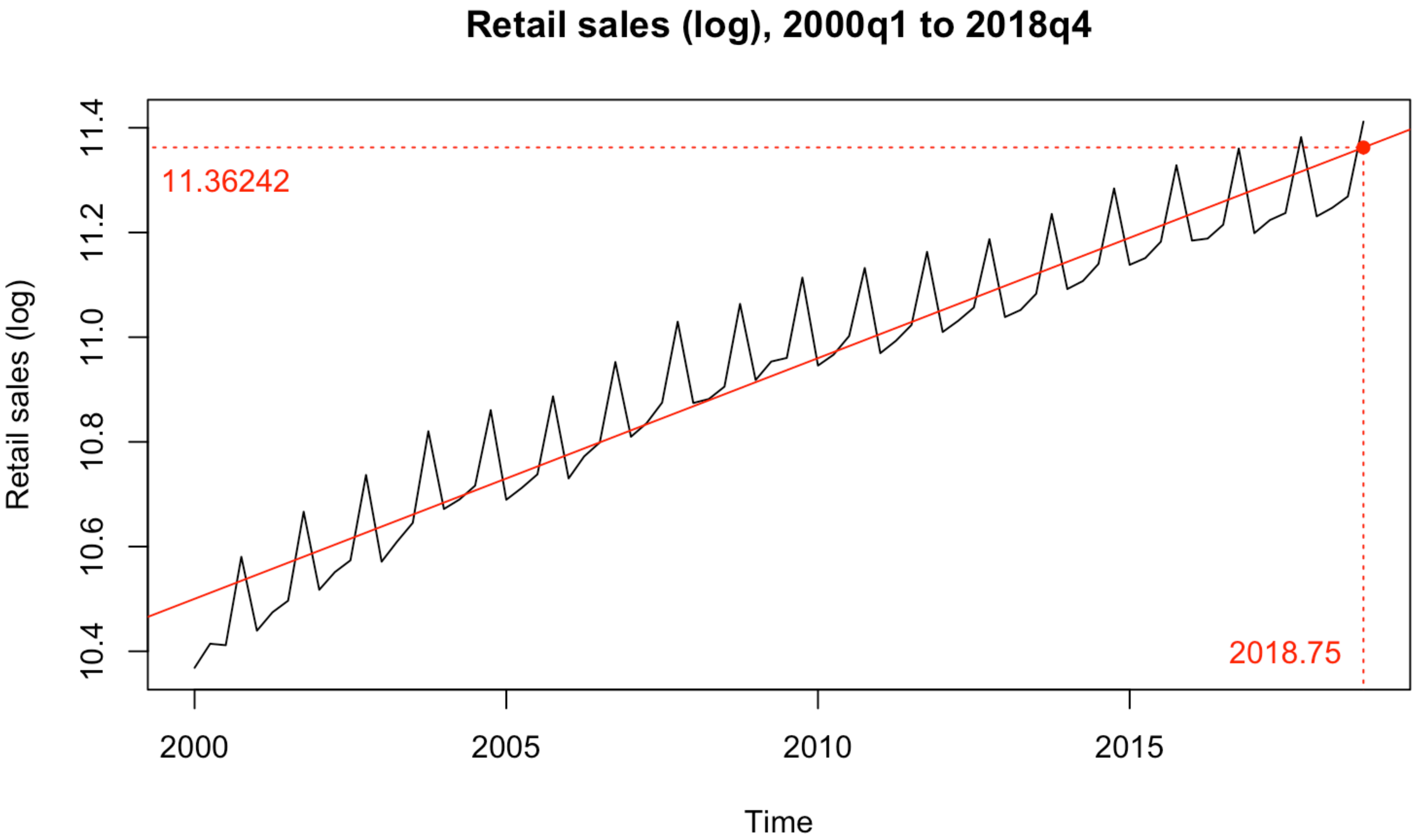
Eg. estimated value of the trend at the end of the sample (i.e. 2018q4 = 2018.75)

$$\widehat{E}(Y_t \mid \mathrm{Time}_t = 2018.75) = \widehat{\beta}_0 + \widehat{\beta}_1 \times 2018.75$$

```
EY2018q4 <- eq$coefficients[1]+
                eq$coefficients[2]*2018.75
cat("Estimated trend at 2018q3: ", EY2018q4
```

```
Estimated trend at 2018q3:  11.36242
```

# Regression Model 1: Time trend



# predict : fitted values

```r
eq1 <- lm(Y~Time_t)
EY_tr_t <- predict(eq1)
print(EY_tr_t)
```

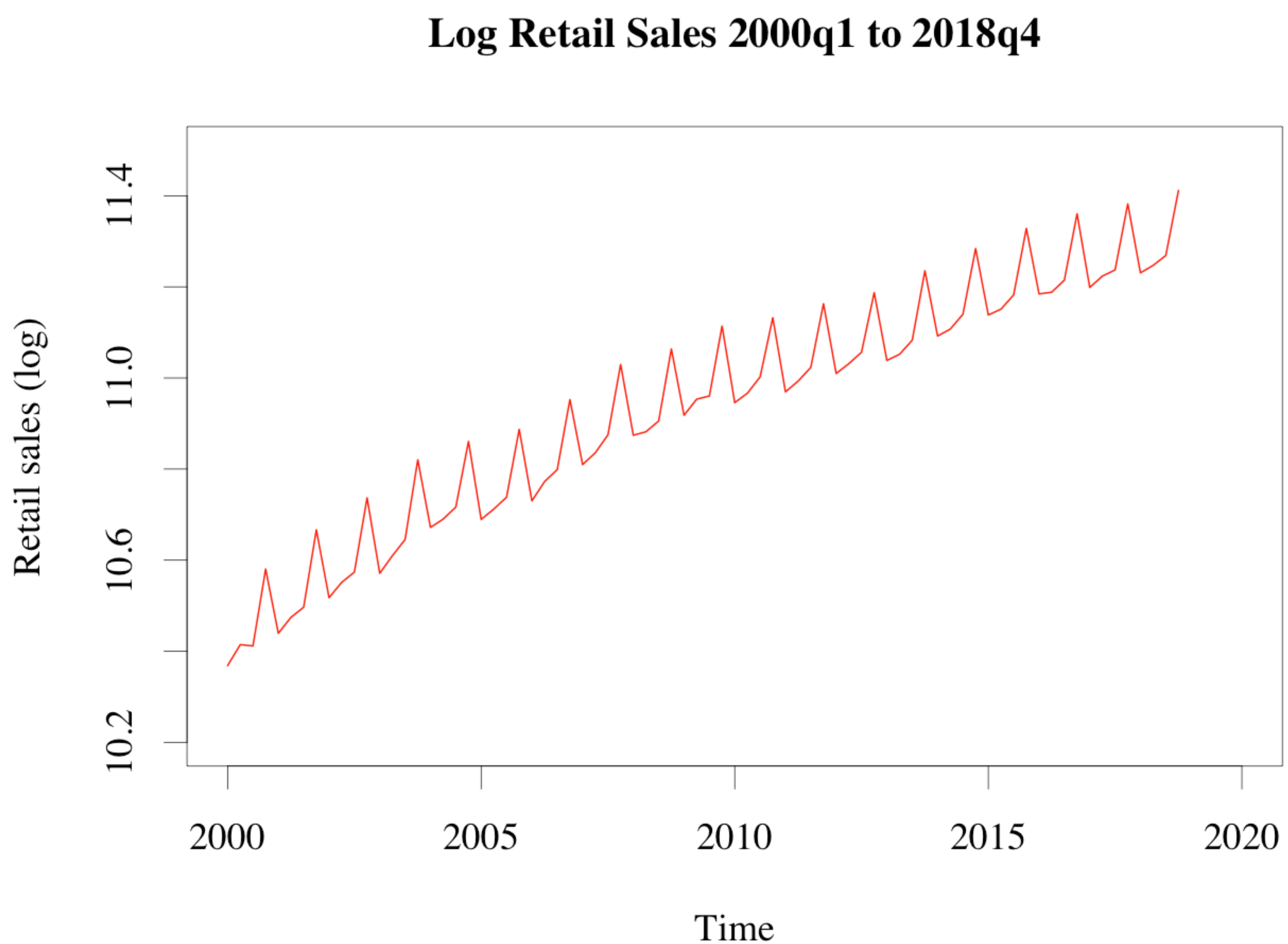
```
       1        2        3        4        5
10.50021 10.51170 10.52320 10.53470 10.54619
       6        7        8        9       10
10.55769 10.56918 10.58068 10.59218 10.60367
      11       12       13       14       15
10.61517 10.62667 10.63816 10.64966 10.66115
      16       17       18       19       20
10.67265 10.68415 10.69564 10.70714 10.71863
```

## predict : fitted values

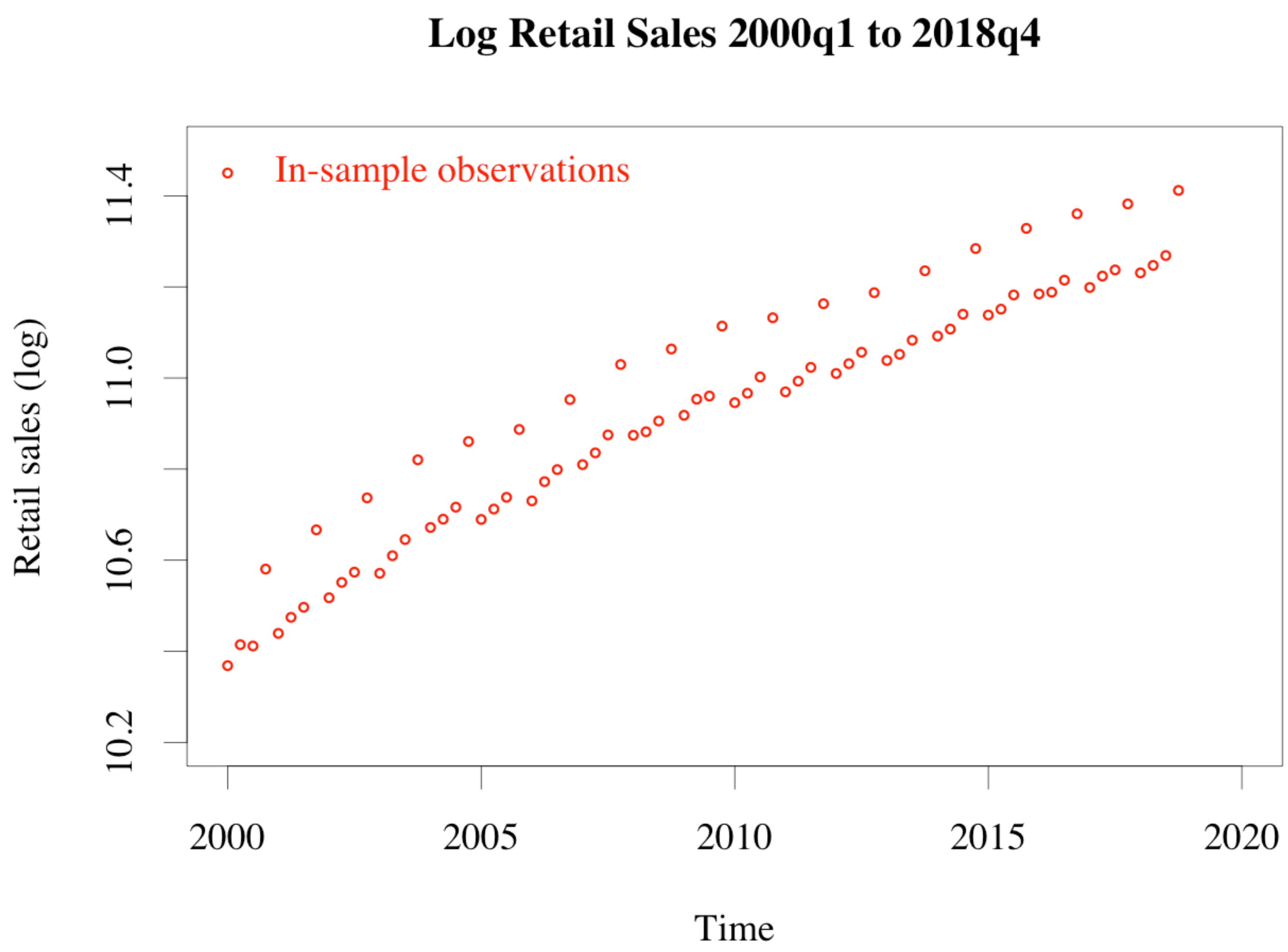
```r
eq1 <- lm(Y~Time_t)
EY_tr_t <- ts(predict(eq1), start=c(2000,1),
                            end=c(2018,4), frequency=4)
print(EY_tr_t)
```

```
         Qtr1     Qtr2     Qtr3     Qtr4
2000 10.50021 10.51170 10.52320 10.53470
2001 10.54619 10.55769 10.56918 10.58068
2002 10.59218 10.60367 10.61517 10.62667
2003 10.63816 10.64966 10.66115 10.67265
2004 10.68415 10.69564 10.70714 10.71863
2005 10.73013 10.74163 10.75312 10.76462
2006 10.77612 10.78761 10.79911 10.81060
```

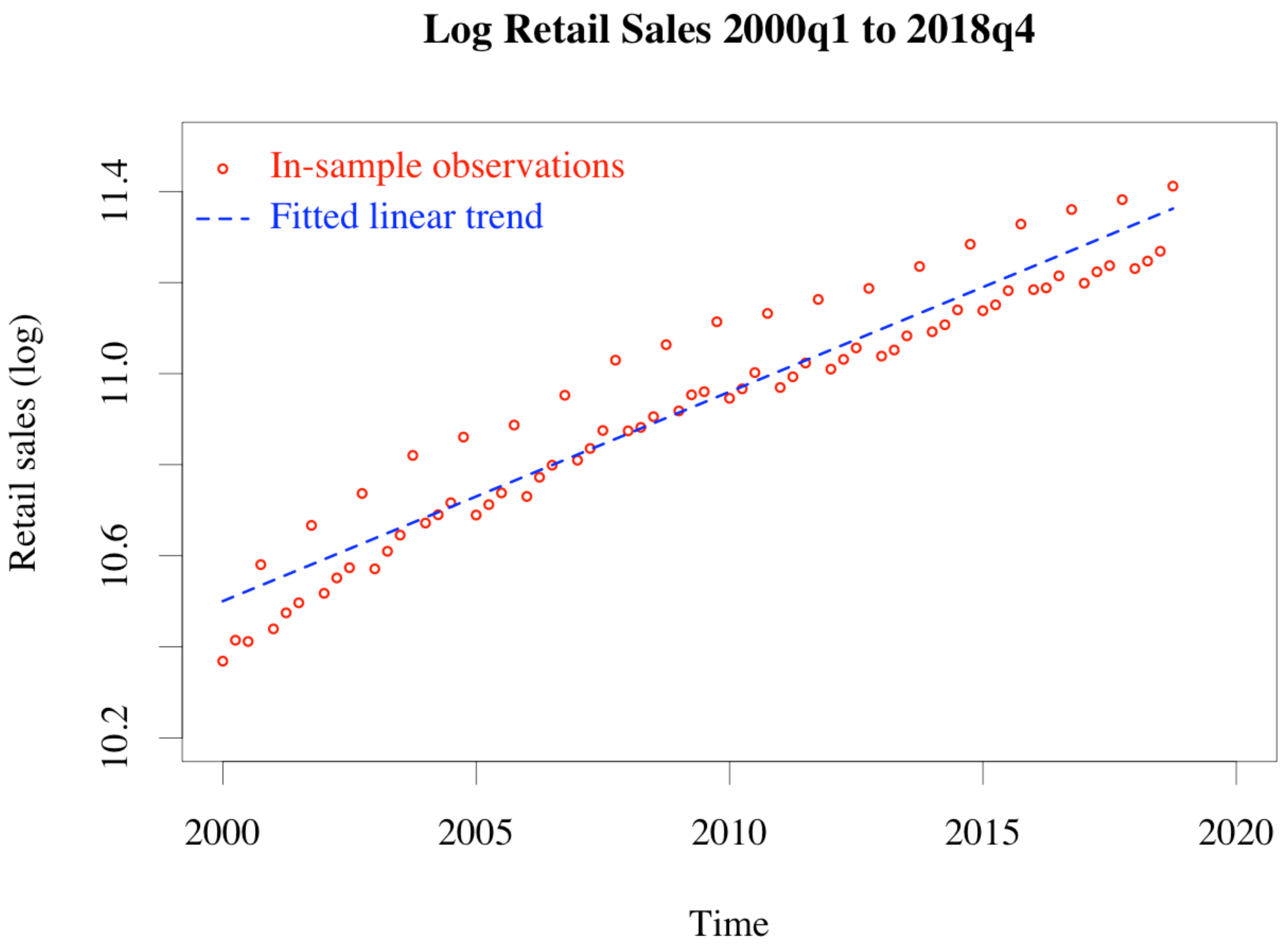
# predict : fitted values



# predict : fitted values



# predict : fitted values



## predict : forecast through 2019

```
eq1 <- lm(Y~Time_t)
Time_2019 <- c(2019,2019.25,2019.5,2019.75)

# Convert to dataframe
X_2019 <- data.frame(Time_t=Time_2019)
```

# predict : forecast through 2019

```
eq1 <- lm(Y~Time_t)
Time_2019 <- c(2019,2019.25,2019.5,2019.75)

# Convert to dataframe
X_2019 <- data.frame(Time_t=Time_2019)
```

Time_t : name of predictor in equation

## predict : forecast through 2019

```r
eq1 <- lm(Y~Time_t)
Time_2019 <- c(2019,2019.25,2019.5,2019.75)

# Convert to dataframe
X_2019 <- data.frame(Time_t=Time_2019)
```

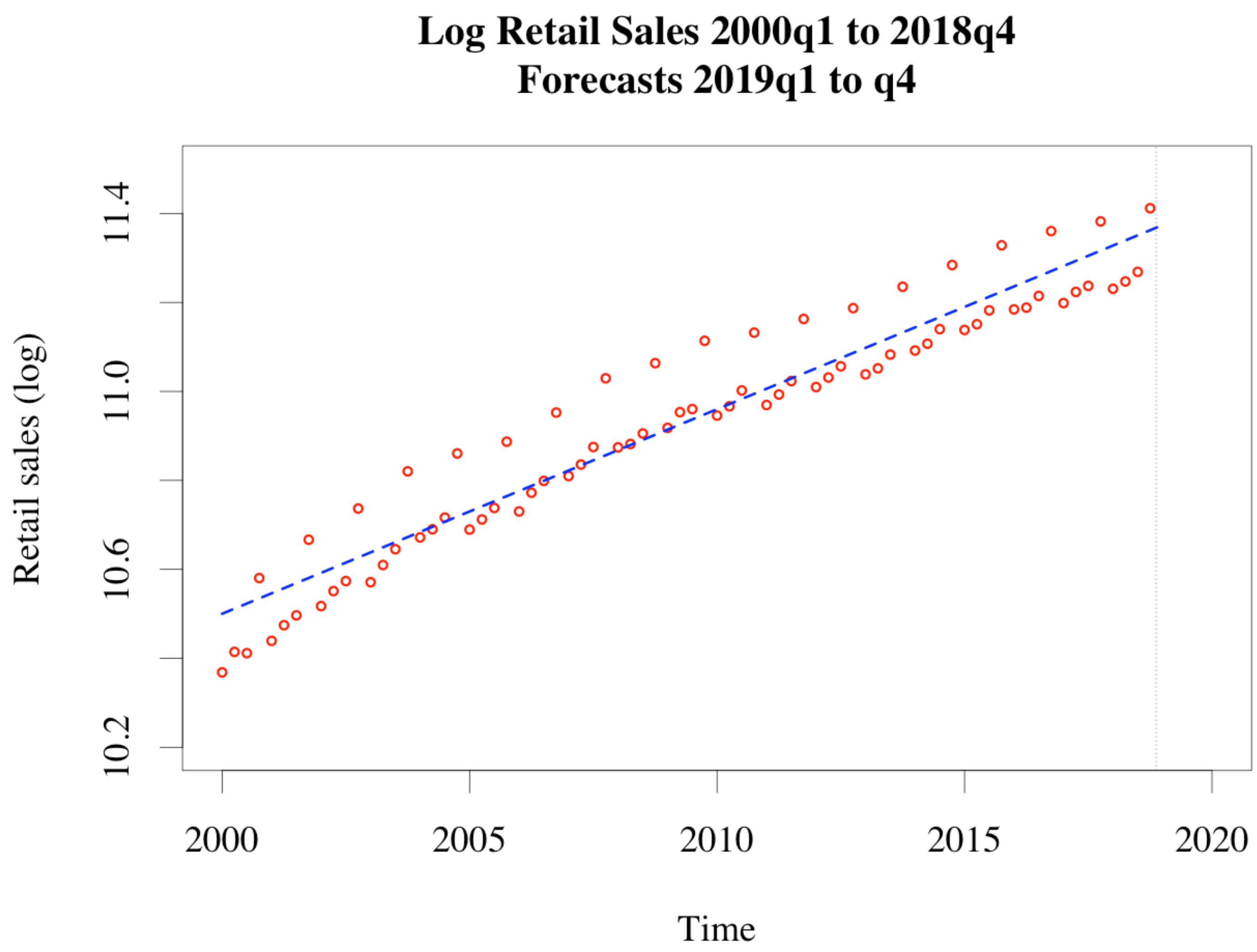
Values at which to predict

## predict : forecast through 2019

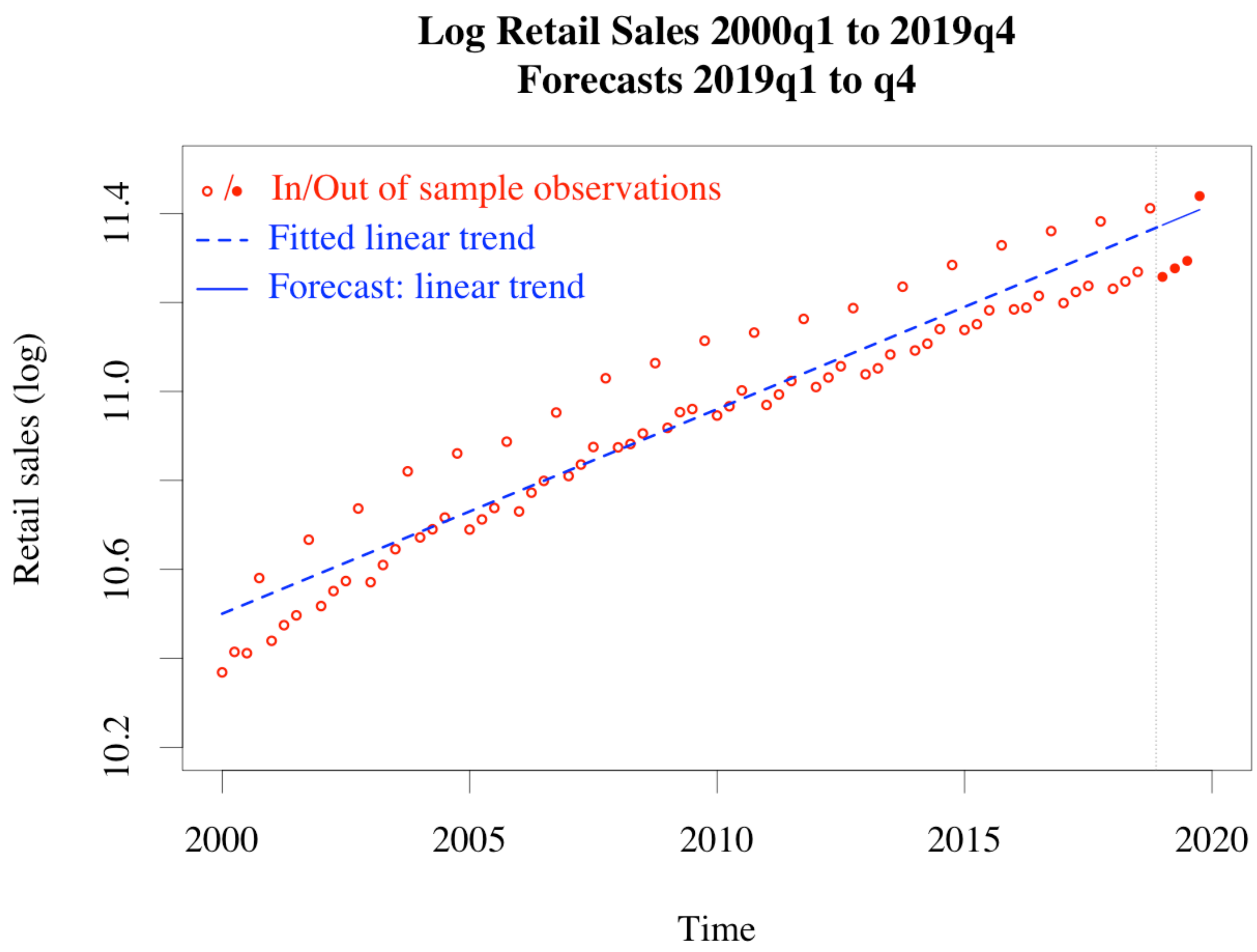
```r
eq1 <- lm(Y~Time_t)
Time_2019 <- c(2019,2019.25,2019.5,2019.75)

# Convert to dataframe
X_2019 <- data.frame(Time_t=Time_2019)

# Forecast 2019 using eq1
EY_tr_2019 <- predict(eq1, newdata=X_2019)
print(EY_tr_2019)
```

```
       1        2        3        4
11.37392 11.38541 11.39691 11.40840
```

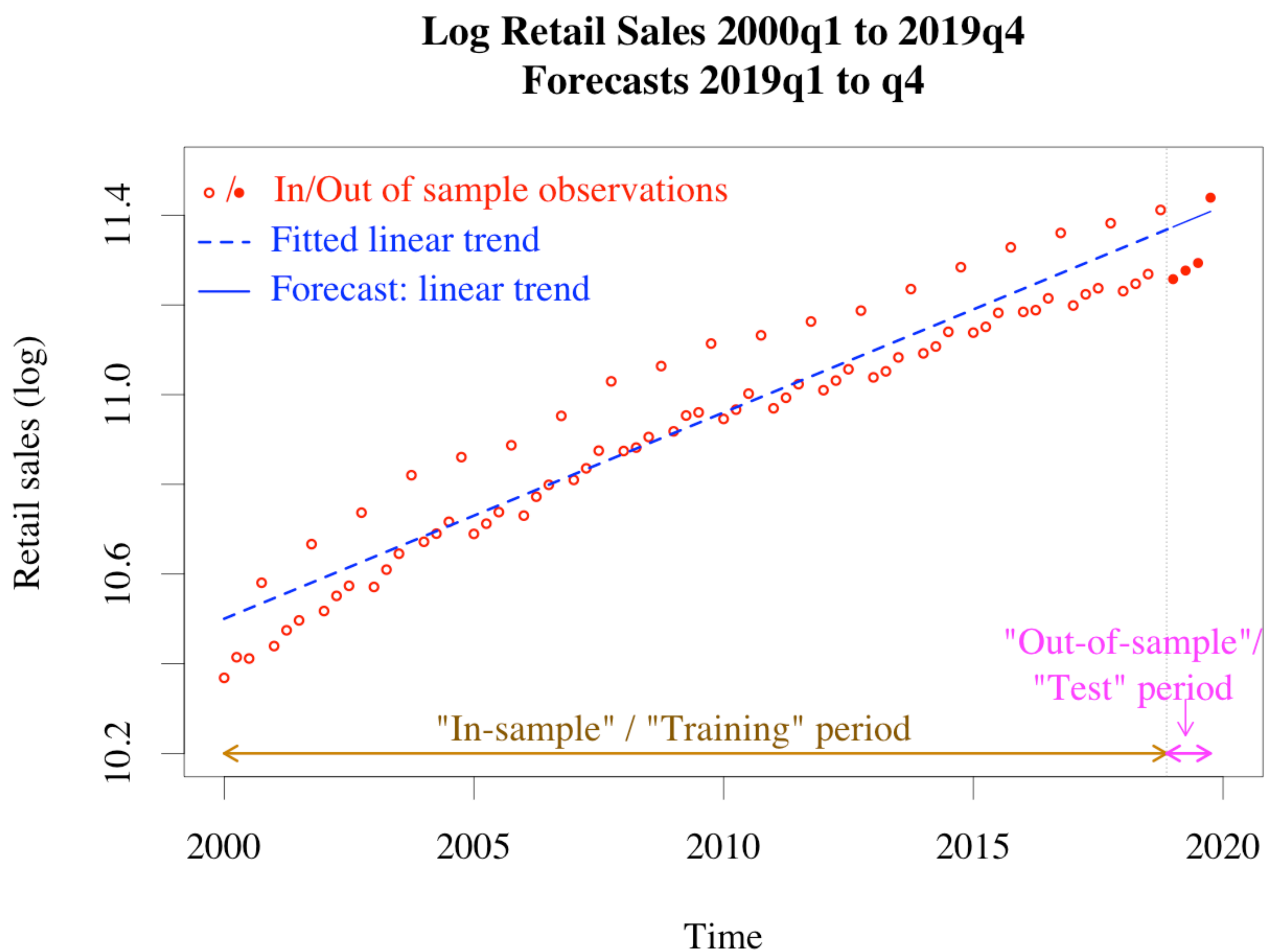
# predict : forecast through 2019



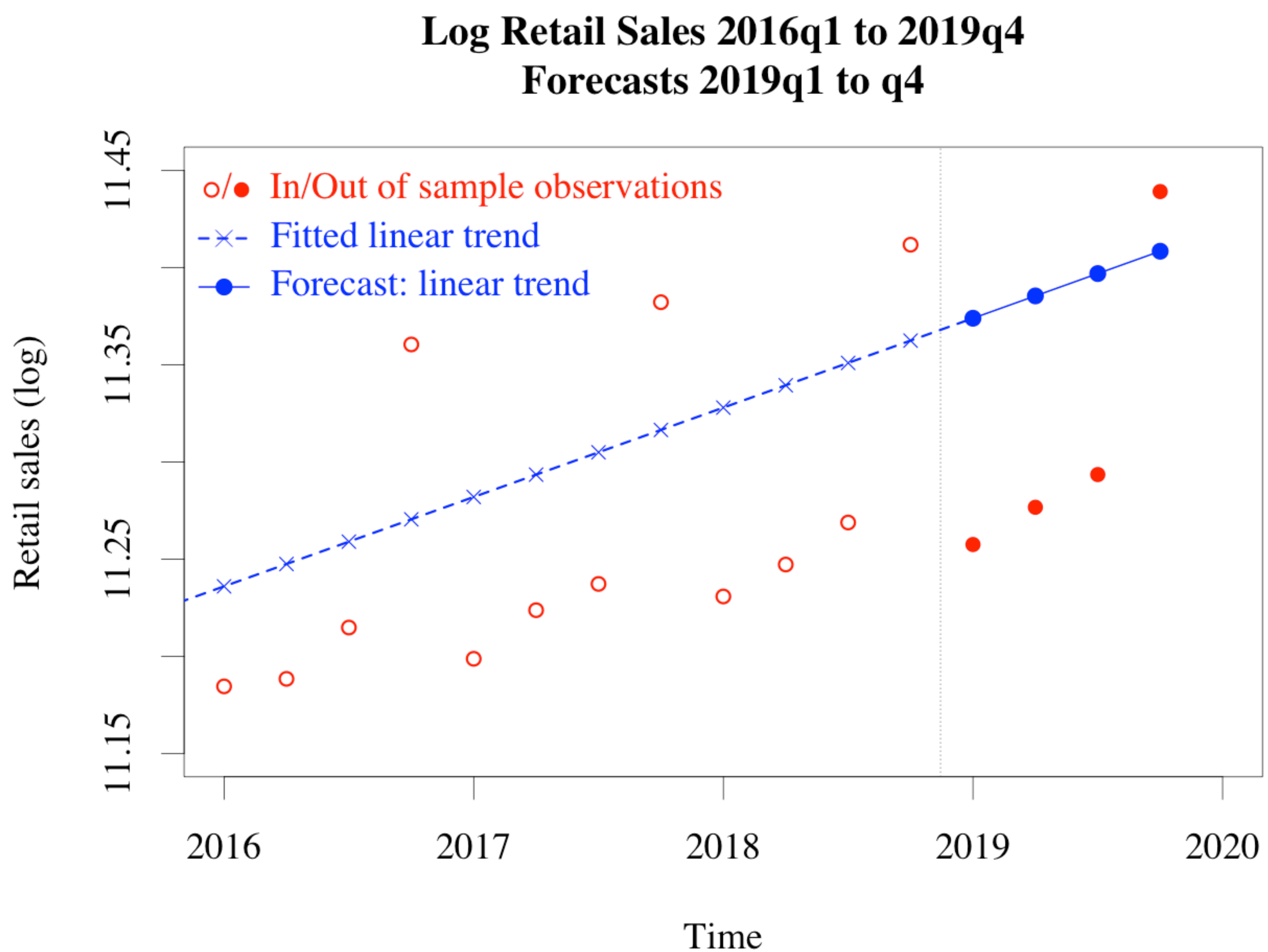
# predict : forecast through 2019



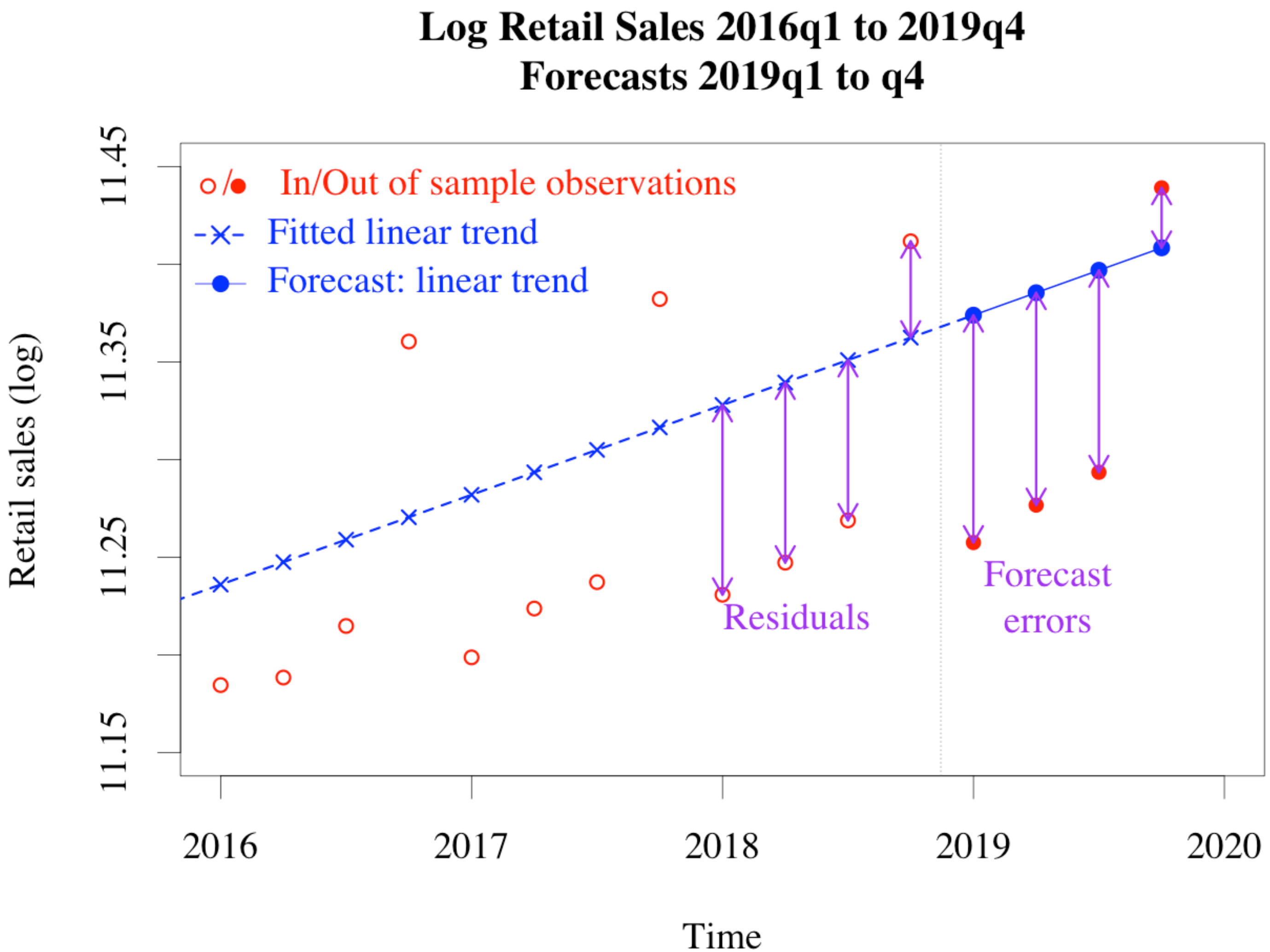
# predict : forecast through 2019



# predict : forecast through 2019



# Residuals and Forecast Errors



## Residuals and Forecast Errors

In-sample :

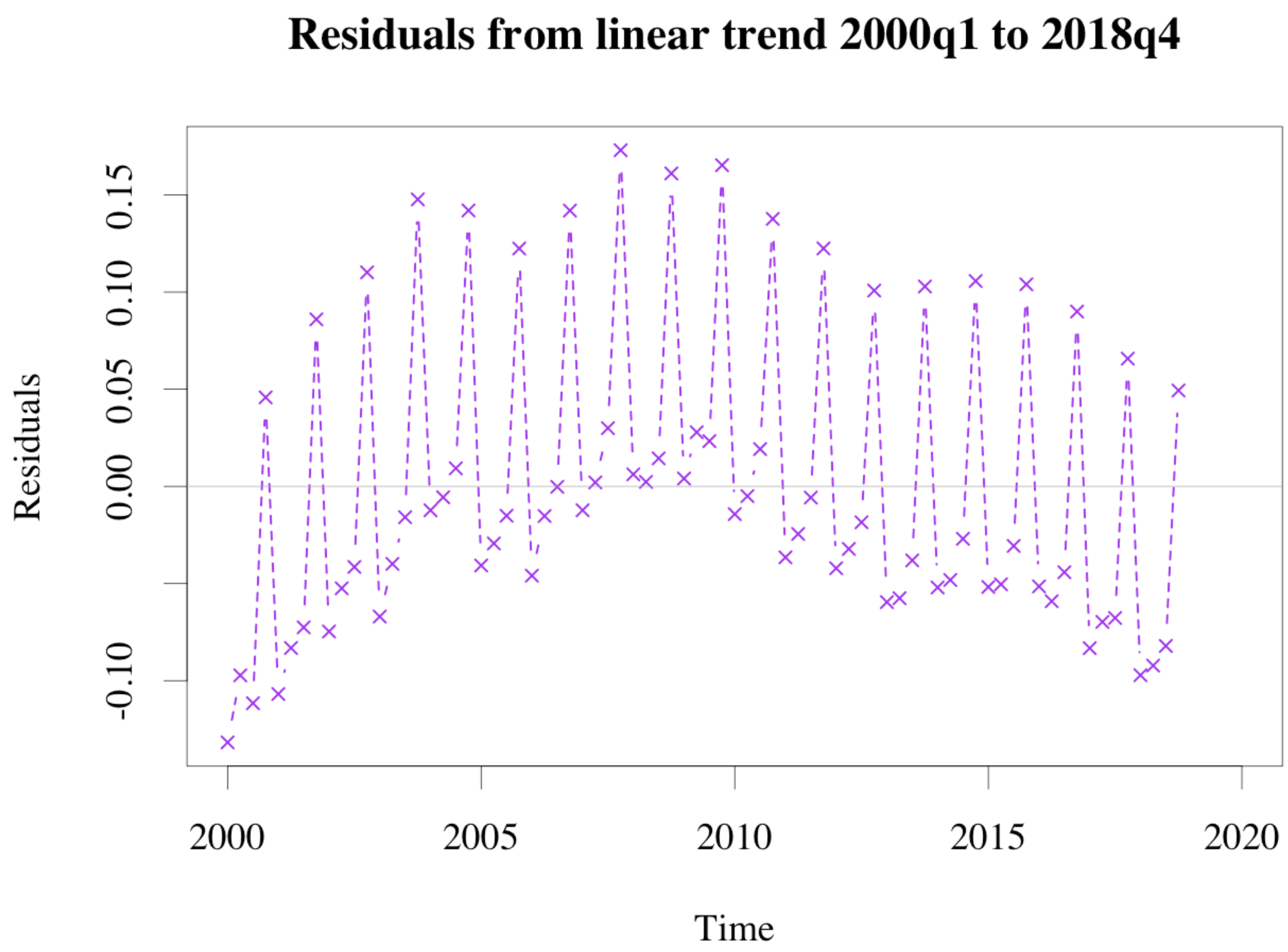
$$\text{Residual}_t = \text{Actual}_t - \text{Fitted}_t$$

Out-of-sample :

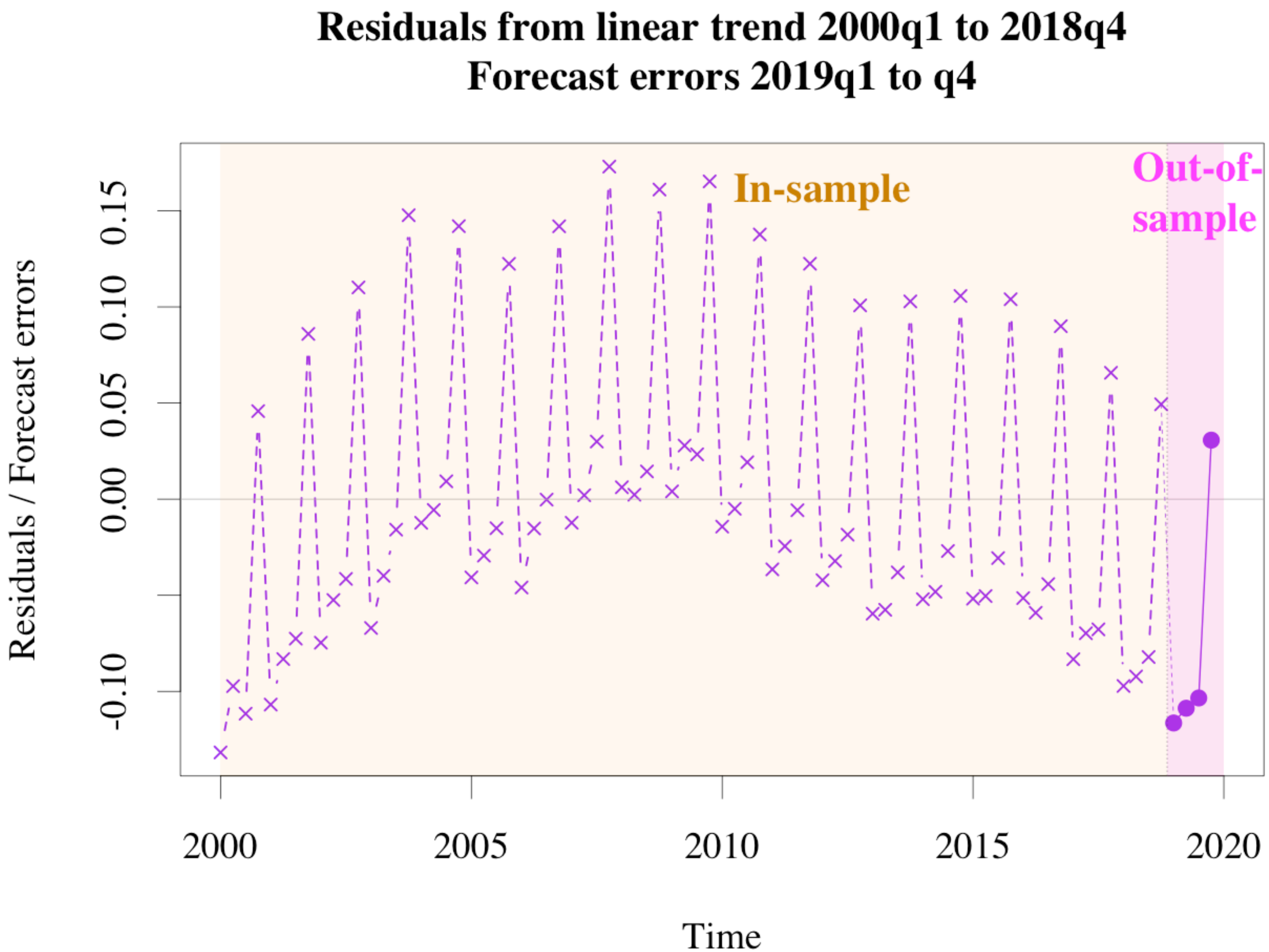
$$\text{ForecastError}_t = \text{Actual}_t - \text{Forecast}_t$$

Patterns in residuals and/or forecast errors can suggest model mis-specification.

# Residuals and Forecast Errors



# Residuals and Forecast Errors



## Residuals and Forecast Errors

In-sample :

$$\text{Residual}_t = \text{Actual}_t - \text{Fitted}_t$$

Out-of-sample :

$$\text{ForecastError}_t = \text{Actual}_t - \text{Forecast}_t$$

Residuals and forecast errors show *seasonality*.

## The Path Ahead

- Formalise the concept of “forecasting”.
- Develop a range of models for forecasting.
- Methods for model selection, evaluation and interpretation.