**Building Models for the data:**

* Technique used to split and process data: **Rolling Origin**

Rolling origin method has two techniques:

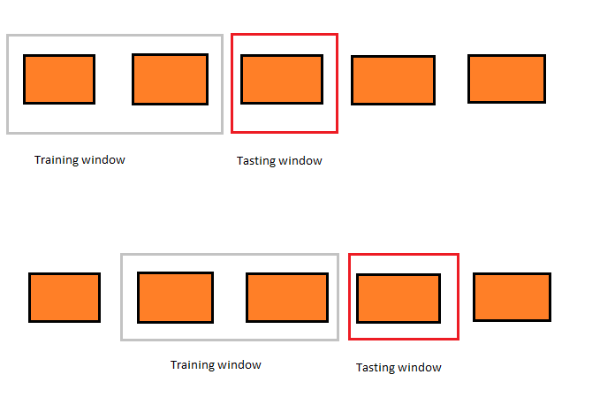
1. Increasing time window method
2. Moving time window.
3. **Increasing time window method:**

A screenshot of a cell phone

Description generated with very high confidence

* In this method, we have split data into train and test sets. We have increased the size of train window step by step. However, testing window size remain constant. We have followed this technique till the train window hits the limit of our data.
* We have divided our data into 4 folds as follows:

1. **Fold 1:** 1977 – 1987 : Train window 1 and 1988 : Test Window 1
2. **Fold 2:** 1977 – 1997 : Train window 2 and 1998 : Test Window 2
3. **Fold 3:** 1977 – 2007 : Train window 3 and 2008 : Test Window 3
4. **Fold 4:** 1977 – 2017 : Train window 4 and 2018 : Test Window 4
5. **Moving window method:**



* In this approach, we keep the size of training and testing window same.
* From above diagram we can see that we considered first two years as training window and next one year as testing window.

In the next fold, we moved training window and testing window to the right keeping size of training and testing window constant.

* Methods and Processes:

The methods and processes that we have used to build the time series models are as follows:

For building the model, we need p and q values. This can be achieved using 3 methods as mentioned below:

1. **Manually finding p and q values:** To build a model we need p and q values. We can determine those values by plotting ACF and PACF plots.
2. **Extended Auto-Correlation Function(EACF):** In this method we give training data as input to generate extended value for p and q.
3. **Automatically finding the p and q values:** This function will generate p and q values automatically and build model accordingly.

We have built models using the above 3 methods for only window 1 for both male and female data as follows:

* **Method 1:** **Manually finding p and q values:**

1. **Male data:**

**A screenshot of a cell phone

Description generated with very high confidence** **A screenshot of a cell phone

Description generated with very high confidence**

ACF Plot PACF plot

From the ACF plot, we can identify the value of q as: **3 and 6**

From the PACF plot, we can identify the value of p as: **3**

Models for the identified p and q values are:

* **AR(3) model:**

**A screenshot of a cell phone

Description generated with very high confidence**

* **MA(3) model MA(6) model:**

**A screenshot of a social media post

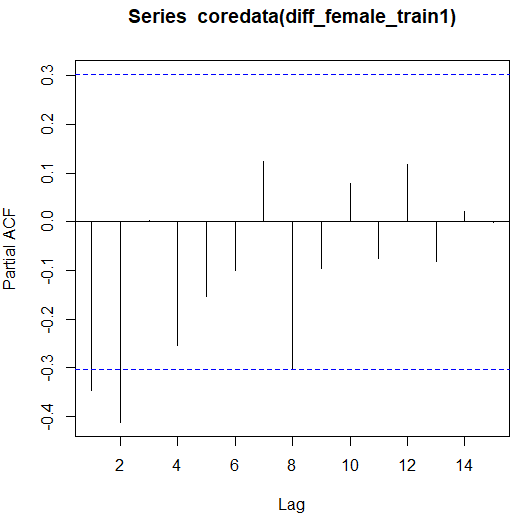
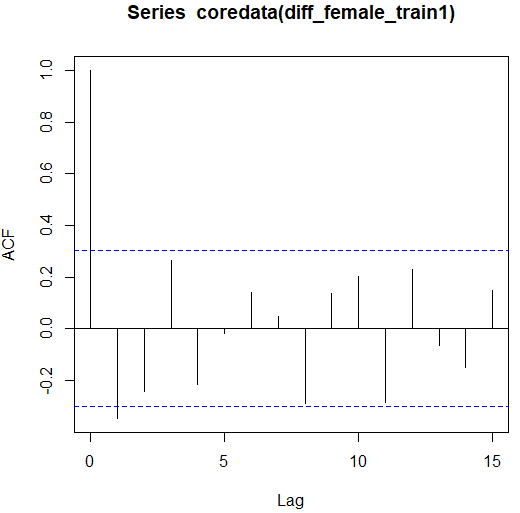
Description generated with very high confidence**

* **ARMA(3,3) and ARMA(3,6) model:**

**A screenshot of a social media post

Description generated with very high confidence**

1. **Female data:**



From the ACF plot, we can identify the value of q as: **1**

From the PACF plot, we can identify the value of p as: **2**

Models for the identified p and q values are:

* **AR(2) model, MA(1) model and ARMA(2,1):**

**A screenshot of a social media post

Description generated with very high confidence**

* **Method 2:** **EACF:**

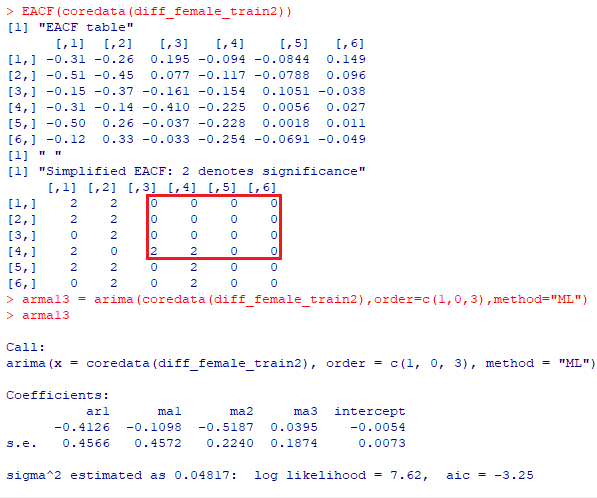
1. **Male:**

A screenshot of a cell phone

Description generated with very high confidence

From the EACF matrix, we can identify that **p = 1 and q = 4**, and built **ARMA(1,4)** model.

1. **Female:**



From the EACF matrix, we can identify that **p = 1 and q = 3**, and built **ARMA(1,3)** model.

* **Method 3: auto.arima():**

In this method we will use auto.arima() function. This function automatically determines the p and q values as well as build a model.

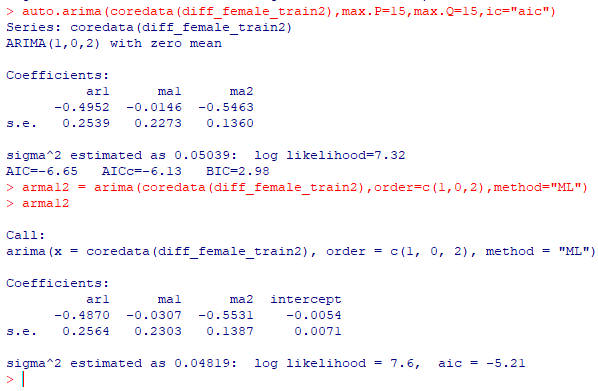
1. **Male:**

A screenshot of a cell phone

Description generated with high confidence

For our training data, auto.arima() function suggests 1 and 2 as p and q values respectively. Hence, we have built a model on these p and q values.

1. **Female:**



For our training data, auto.arima() function suggests 1 and 2 as p and q values respectively. Hence, we have built a model on these p and q values.

Similar to window 1, we have built models using EACF and auto.arima() to build models for each train window. The models available for comparison are as follows:

**Male data:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **SR No.** | **Model** | **AIC** | **Window** | | **1.** | AR(3) | -1200.66 | Window1 | | **2.** | MA(3) | -1192.53 | | **3.** | MA(6) | -1197.68 | | **4.** | ARMA(3,3) | -1196.88 | | **5.** | ARMA(3,6) | -1196.90 | | **6.** | ARMA (1,4) | -1196.74 | | **7.** | ARMA (1,2) | -1200.53 | | **8.** | ARMA (1,4) | -2335.4 | Window2 | | **9.** | ARMA (2,2) | -2338.32 | | **10.** | ARMA (1,4) | -3473.67 | Window3 | | **11.** | ARMA (1,2) | -3476.44 | | **12.** | ARMA (2,4) | -4536.91 | Window4 | | **13.** | ARMA (1,2) | -4538.94 | |
|
|
|
|
|
|
|
|
| **Female data:**   |  |  |  |  | | --- | --- | --- | --- | | **SR No.** | **Model** | **AIC** | **Window** | | **1.** | AR(2) | 18.55 | Window1 | | **2.** | MA(1) | 16.76 | | **3.** | ARMA(2,1) | 19.46 | | **4.** | ARMA(1,3) | -3.25 | | **5.** | ARMA(1,2) | -5.21 | | **6.** | ARMA(1,3) | -3.25 | Window2 |  | ARMA (1,4) | -3473.67 | Window3 | | **7.** | ARMA(1,2) | -5.21 |  | ARMA (1,2) | -3476.44 |  | | **8.** | ARMA(1,3) | -19.13 | Window3 |  | ARMA (2,4) | -4536.91 | Window4 | | **9.** | ARMA(1,2) | -21.12 |  | ARMA (1,2) | -4538.94 |  | | **10.** | ARMA(1,2) | -27.4 | Window4 |  |  |  |  | | **11.** | ARMA(1,2) | -19.65 |  |  |  |  | |