Time Series Analysis of UPI Data

[Debaditya Chakraborty](https://twitter.com/bjblazkowiczzz)

2024-06-25

Table of Contents

# Keywords

1. **UPI -** UNIFIED PAYMENT INTERFACE
2. **PSP -** Payment Service Provider.
3. **MSME -** Micro Small and Medium Enterprises.

# **Introduction**

The introduction of UPI has revolutionized the digital space. UPI usage has exponentially increased since its inception in 2016, with its growth outpacing all other modes of digital payments. UPI is an instant, real-time payment network built, owned, and operated by the National Payments Corporation of India (NPCI). This payment system is built as an inter-operable protocol and allows third-party vendors to build apps to provide payments as a service to all customers of participating banks. Due to interoperability, customers with an account in Bank “A” can use a payments app built by PSP “X” to send money from their account in one bank to self or other party accounts of any other bank or PSP participating in UPI via QR codes, mobile numbers, or other identifiers, with instant settlement of payments (NPCI, 2016). UPI is used by multiple stakeholders, including individuals, micro, small, and medium enterprises (MSMEs), and especially smaller merchants. It is easily accessible through mobile devices, provides convenient payment initiation methods, such as users registered mobile numbers, QR codes, etc., and ensures universal interperability between financial institutions. These design choices have helped enhance digital and financial literacy and included the portion of the population that was formerly underserved or unserved by financial institutions.

## Impact of UPI in India’s Economy

In about eight years, India’s indigenously developed UPI, has evolved into the default option to transact—from small ticket purchases at roadside shops to settling utility bills to restaurant bills, to now IPO stock purchases and mutual fund payments.

This transformation, which has now become a global template that many other countries are emulating, is founded on multiple edifices powered by a behavioural change among hundreds of millions.While UPI has made sending and receiving money at the tap of a mobile phone app, the bigger question is how has it added to India’s broader economy? Importantly, what has been the specific incremental contribution of UPI or India’s rapid digitalisation of payments to India’s gross domestic product (GDP).The answer to this is two-fold. One is the opportunity cost. Two is through enabling easier credit-driven spending.

### Ease of Financial Access

UPI has had a profound impact on financial access in India by enhancing the ease and convenience of digital transactions, especially for those who were previously underserved by traditional banking services. Here are several ways UPI has contributed to improving financial access:

1. **Accessibility**: UPI can be accessed through smartphones, making it available to a wide range of individuals, including those in remote areas where traditional banking infrastructure is limited.
2. **Inclusion of Unbanked Population**: UPI has facilitated financial inclusion by allowing unbanked individuals to open a bank account digitally and link it to UPI, enabling them to participate in digital transactions.
3. **Simplified Transactions**: UPI simplifies the process of making payments and transferring money, even for those with limited literacy or familiarity with banking procedures, thus lowering the barrier to entry for digital financial services.

### Digital Lending

Digital lending is one of the sections of fintech in India which has been positively impacted by UPI via streamlining the lending process and enabling more efficient disbursement of loans. UPI facilitates instant fund transfers, allowing digital lenders to disburse loans to borrowers quickly and efficiently. This speed is particularly beneficial for individuals in need of urgent funds, such as during emergencies or for time-sensitive expenses. UPI’s seamless payment interface simplifies the loan repayment process for borrowers. UPI transactions are highly secure, utilizing two-factor authentication and encryption protocols to protect users’ financial information. This enhanced security instills confidence among borrowers,

# **Abstract**

This paper presents a statistical analysis of the growth trajectory of the Unified Payments Interface (UPI) in India and explores its future prospects. UPI, introduced in April 2016 by the National Payments Corporation of India (NPCI), has rapidly transformed the landscape of digital payments in the country. The study begins by examining the evolution of UPI, highlighting key milestones and adoption trends since its inception. Utilizing available data sets, including transaction volumes, transaction amounts, merchant integration , user preference of UPI Apps etc . The paper offers insights into the exponential growth experienced by UPI over the years. Furthermore, it delves into a progressive comparison of UPI with other available online payment systems .All this is done using statistical methods like Descriptive Analysis, Time series analysis , Regression analysis , EDA(Exploratory Data Analysis) etc. The paper also discusses the future growth of UPI in different sectors via statistical methods of estimation. Finally, it concludes with the results about the different factors influencing the growth of UPI and remarks about different findings within the analysis. # Methodology {.unnumbered}

# **Analyzing Monthly UPI Transaction(From 2016 to 2024)**

#### **The Data**

Table 1: **Monthly UPI Transaction** **(Value &Volume)**

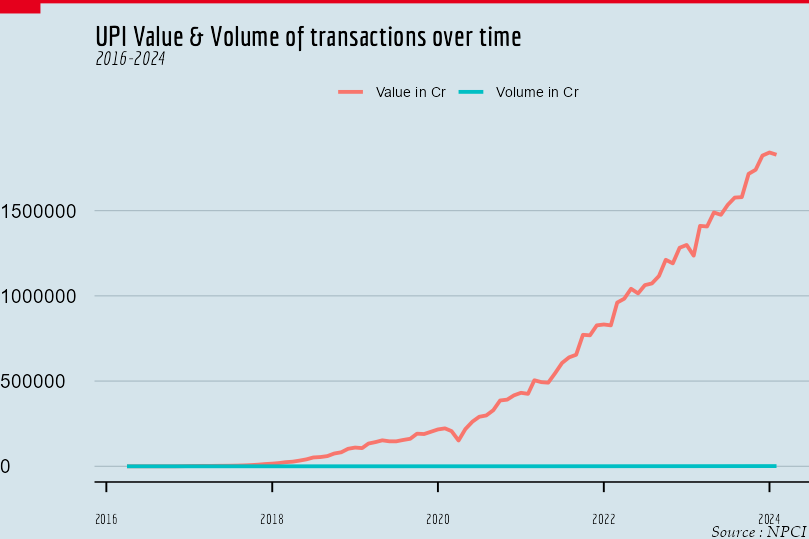
*2016 to 2024*

| Month | No. of Banks live on UPI | Volume(In Mn) | Value(In Cr) | Volume(In Cr) |
| --- | --- | --- | --- | --- |
| 2016-04-01 | 21 | 0 | 0.00 | 0.000 |
| 2016-05-01 | 21 | 0 | 0.00 | 0.000 |
| 2016-06-01 | 21 | 0 | 0.00 | 0.000 |
| 2016-07-01 | 21 | 0.09 | 0.38 | 0.009 |
| 2016-08-01 | 21 | 0.09 | 3.09 | 0.009 |
| 2016-09-01 | 25 | 0.09 | 32.64 | 0.009 |
| 2016-10-01 | 26 | 0.1 | 48.57 | 0.010 |
| 2016-11-01 | 30 | 0.29 | 100.46 | 0.029 |
| 2016-12-01 | 35 | 1.99 | 707.93 | 0.199 |
| 2017-01-01 | 36 | 4.46 | 1696.22 | 0.446 |
| 2017-02-01 | 44 | 4.38 | 1937.71 | 0.438 |
| 2017-03-01 | 44 | 6.37 | 2425.14 | 0.637 |
| 2017-04-01 | 48 | 7.2 | 2271.24 | 0.720 |
| 2017-05-01 | 49 | 9.36 | 2797.07 | 0.936 |
| 2017-06-01 | 52 | 10.35 | 3098.36 | 1.035 |
| *Source*:National Payments Corporation of India (NPCI) | | | | |

Only the first 10 rows are shown for convenience.The volumes have been converted to crore from million.

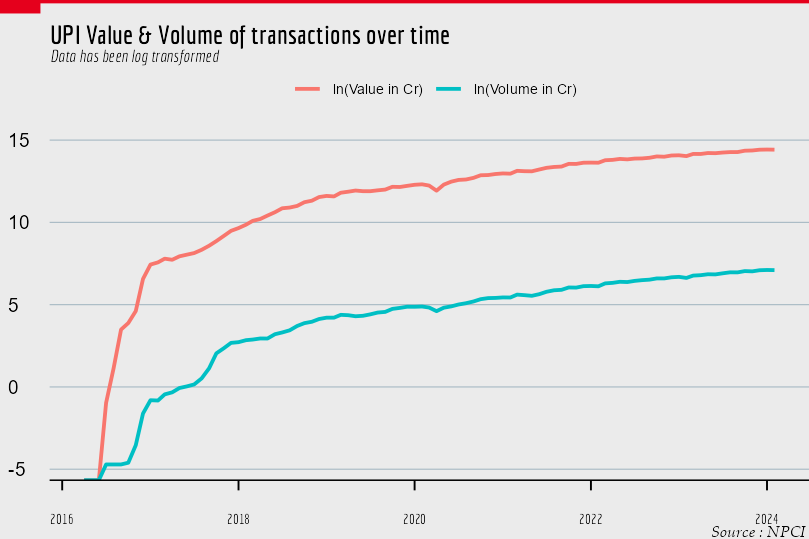
## **Exploratory Analysis**

### Plot showing the growth of Upi Transaction volume and Transaction amount overtime(2016-2024)



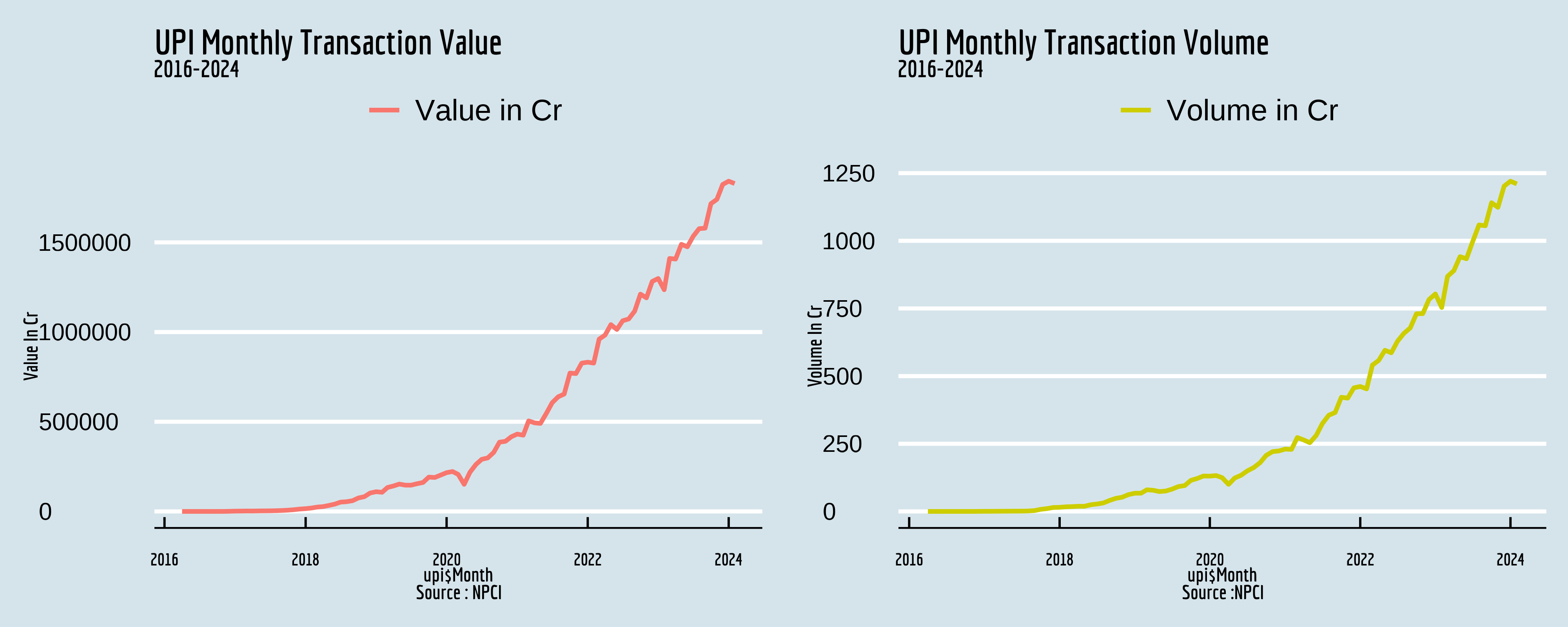
As it can be seen the transaction amounts are much larger than the volume so it is not possible to contain them in the same graph and compare their growth simultaneously. To solve this problem we use the log transformation for both variables to bring them in a comparable range.

### Plot after log transform



Now we can see the growth of both factors simultaneously and as we would’ve guessed the growth of UPI transaction volume and Transaction Amount is nearly identical even though the difference between their values is very high. This is due to the fact that each individual transaction results in some amount of money being transferred which can range from very small to very high values.It is hard to observe the presence of trend and seasonal components from this so we plot value and volume in separate plots.

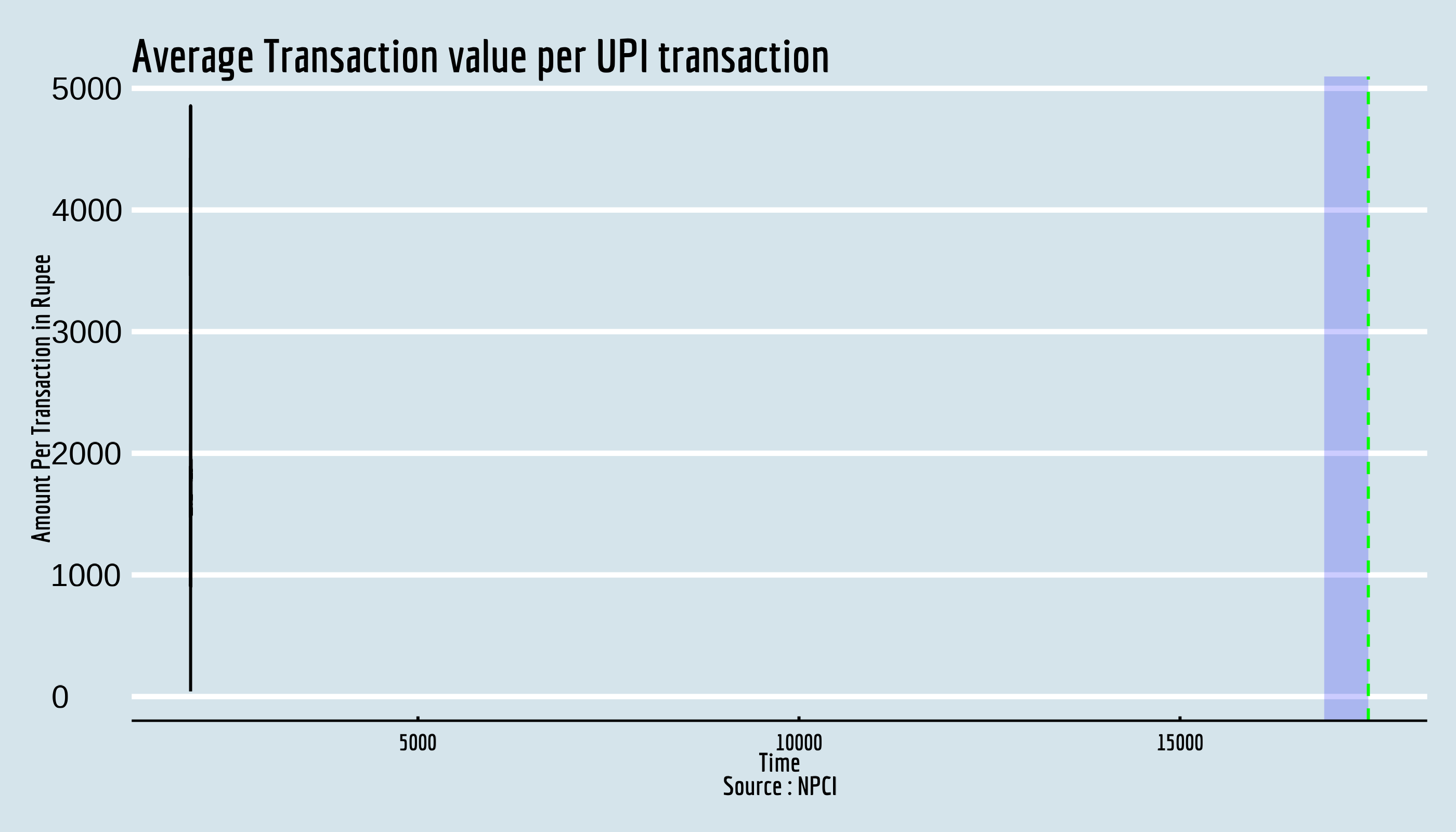
### Plot of Value and Volume of transactions per month.



**Figure** **:** Fig 1.3

The presence of a strong secular trend is visible with some random fluctuations in both of the plots. There isn’t a very strong seasonal effect that can be identified from the plot,what should be noted duly is that the growth of UPI in terms of volume and value of transactions both have not been ~~linear~~ but rather have been very much exponential this information will help us in identifying a proper model for our analysis. We can also see the impact of *COVID-19* as there is a significant change in both metrics(Value & Volume) during the start of *COVID-19* pandemic .We can observe that since post pandemic the increase in value of transaction has been rapid , there might be an underlying effect of *inflation* in this which we may analyze ahead.

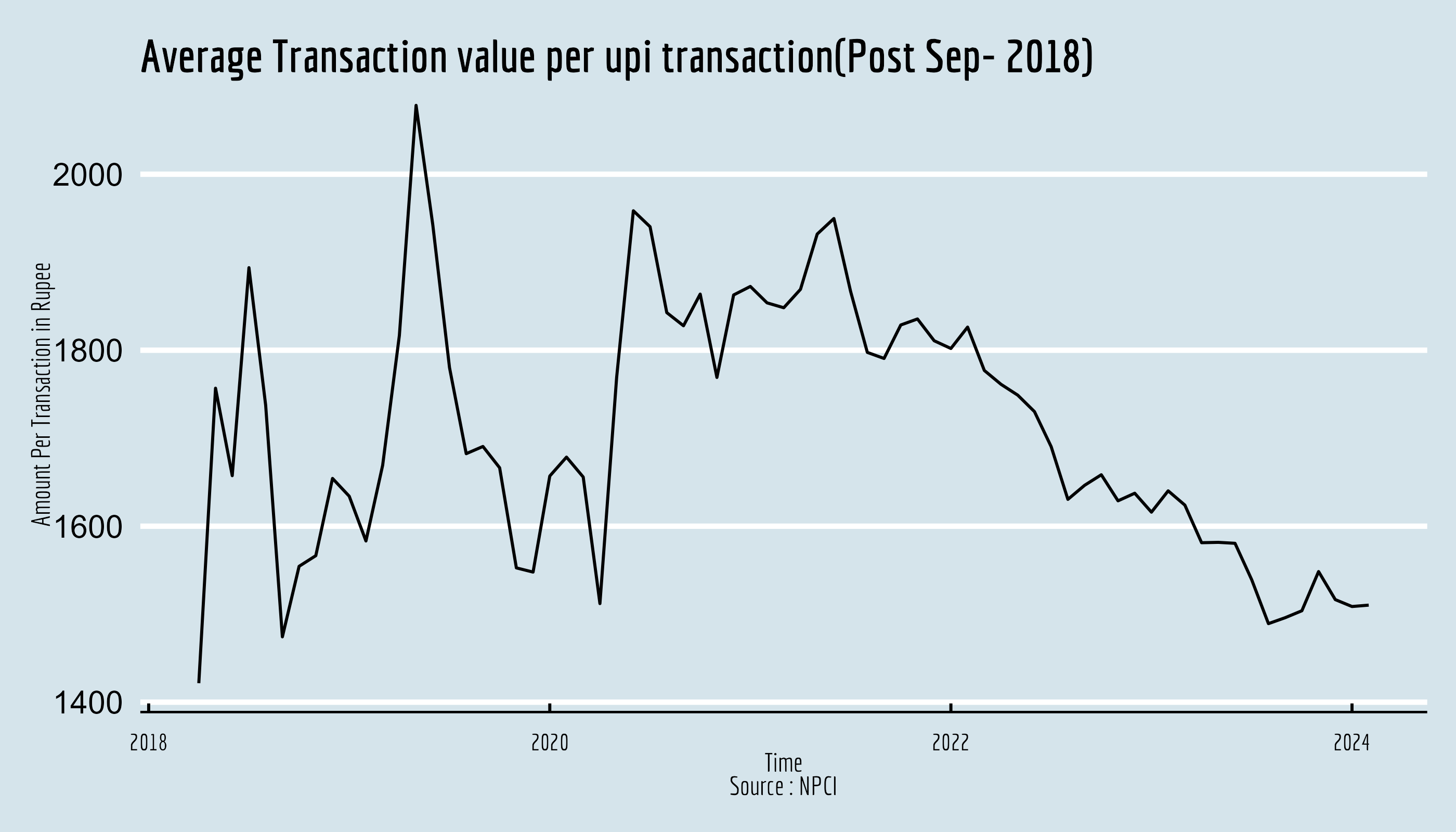
### Plot of Average Transaction value.



**Figure** **:** Fig 1.4

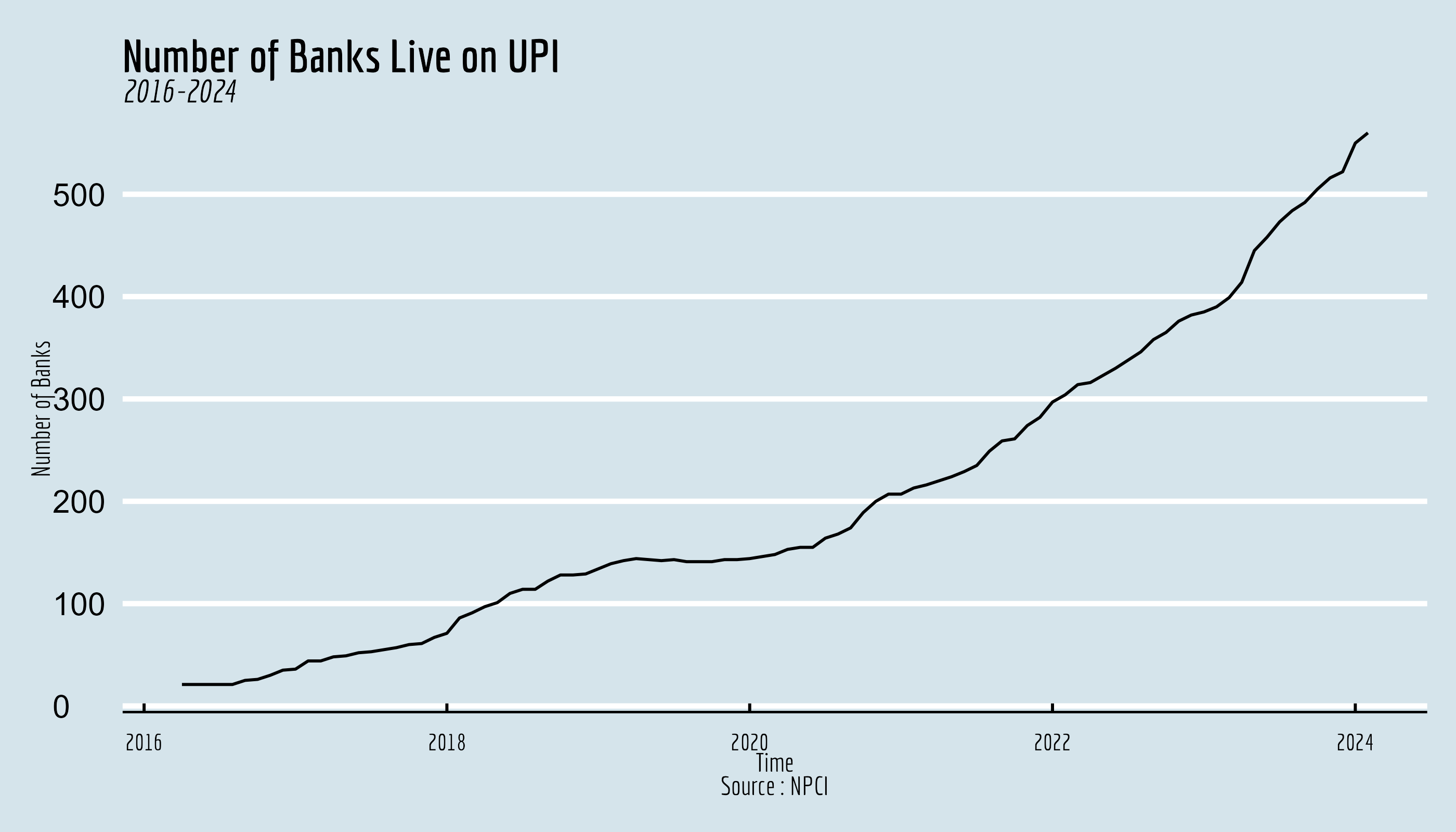
From the plot we can see that the average transaction value between the period of initialization(2016) and upto 2018( *marked by blue area* )has been extremely *high* as during this initial period UPI was used by a selected few individuals. . Overtime with easier access of internet UPI started to become a mainstream method of payment and the average payment value stabilized near around 2019, Right now as the data shows the average transaction value per transaction is *decreasing slowly* this shows as time increases people are using UPI for more and more smaller transactions.

### Plot of Average transaction value post September 2018



In this period we can observe the average transaction value stabilizing around 1300-1500₹ and a slow decline can be seen as well. This time series might be of special interest as it is the only one so far which does not show a strong secular trend and there are no visible seasonal fluctuations as well but it does contain a significant amount of random fluctuation .It will be best to avoid the deterministic procedures for this time series so we will be analyzing this object only using stochastic analysis methods.

### Plot of Number of Banks live in UPI.



Number banks allowing UPI registration is growing rapidly this indicates at the growth of financial inclusion among the population of India the more banks especially regional banks allow UPI registrations the better will be the penetration of digitization of payments throughout the country.

*Table 2*

Summary Statistics for UPI Monthly Metrics

|  | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | NA’s |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Value | 0.00000 | 25597.2250 | 216242.970 | 500797.1573 | 896287.385 | 1841083.970 | 0.000 |
| Volume | 0.00000 | 18.3765 | 130.502 | 303.8076 | 501.140 | 1220.302 | 0.000 |
| Avg. Transaction Val | 42.22222 | 1577.0483 | 1704.861 | 1865.8310 | 1863.037 | 4857.000 | 3.000 |
| Avg. Transaction Val(Post 2018 Sept) | 1421.60406 | 1582.4160 | 1682.418 | 1707.4222 | 1827.031 | 2078.268 | 1421.604 |

## Time Series Analysis of Monthly UPI Metrics.

* [Name Required] - Initially we may want to perform some classical methods such as smoothing procedures like Moving Average or filters to dampen the fluctuations and then proceed to decompose the time series into several components.After this we will move to stochastic models such as *AR*(Auto Regressive), *MA*(Moving Average) and if needed *ARMA*(Auto Regressive Moving Average Process) AND *ARIMA*(Auto Regressive Integrated Moving Average Process) given the condiitons to assume these models hold such as *stationarity* etc. We may also want to define invidual terms in our analysis as we move forward.

**Here we have 4 time series objects**

* val\_time:Value of UPI transactions per month (In Crore,Rupee)
* vol\_time:Volume of UPI transactions per month(In crore)
* per trans:Average transaction amount per UPI transaction in each month(In rupee).
* bank\_num: Total number of registered banks on UPI.

### Analysing Time series with Trend and no Seasonal Variation

From the exploratory analysis we found that that 2 of these time series 1.val\_time 2.vol\_time contained a significant amount of secular positive trend with some underlying random component,there is no visible seasonal fluctuation in these data.

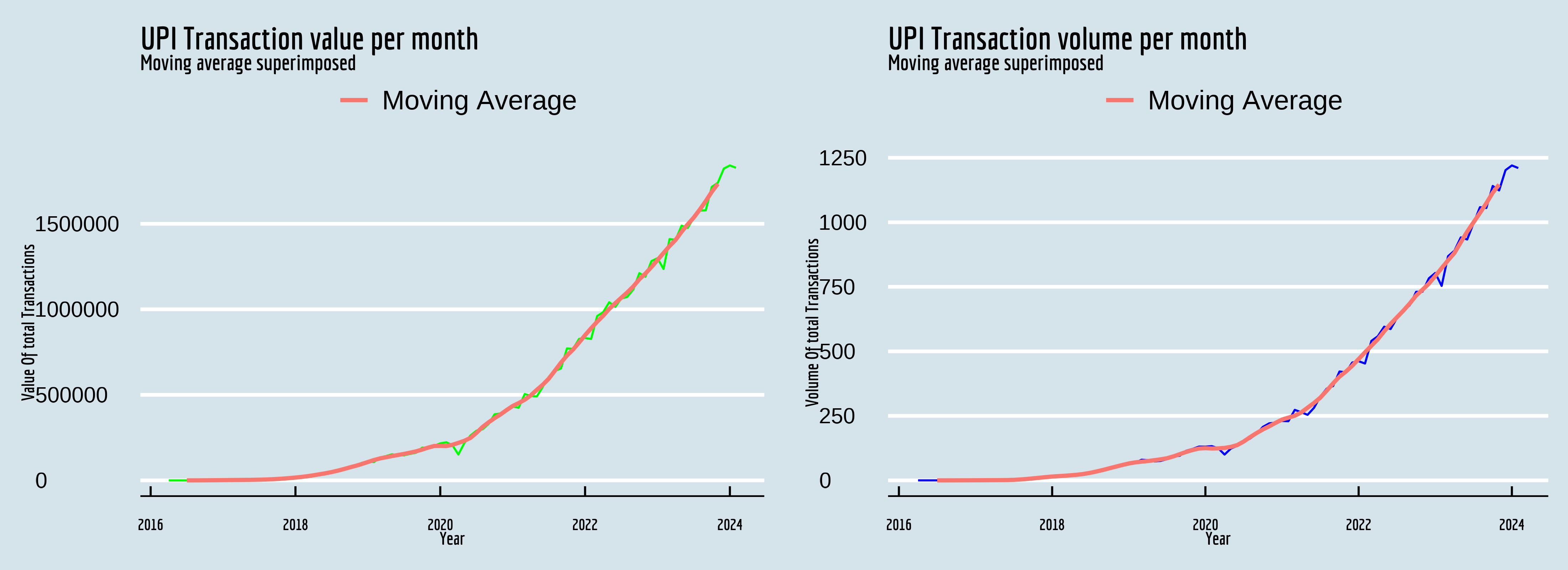
**Trend**- From ([Kendall and Stuart 1966](#ref-Kendall)) “The concept of trend is more difficult to define. Generally, one thinks of it as a smooth broad motion of the system over a long term of years, but” long” in this connexion is a relative term, and what is long for one purpose may be short for another.”  
The simplest type of trend is the familiar ‘linear trend + noise’, for which the observation at time t is a random variable , given by

where α, β are constants and denotes a random error term with zero mean. The mean level at time t is given by

this is sometimes called *‘the trend term’*. Other writers prefer to describe the slope β as the trend, so that trend is the change in the mean level per unit time.The trend in Equation (1) is a deterministic function of time and is sometimes called a *global linear trend*. In practice, this generally provides an unrealistic model, and nowadays there is more emphasis on models that allow for local linear trends.This could be done deterministically, but it is more common to assume that α and β evolve stochastically giving rise to what is called a stochastic trend.So far the models considered have been linear,another possibility, depending on how the data look, is that the trend has a nonlinear form, such as quadratic growth.([Chatfield 2016](#ref-chatfield2016analysis))

* **Filtering**- One of the most used procedure for dealing with a trend is to use a linear filter, which converts one time series, into another , by the linear operation
* where is a set of weights. In order to smooth out local fluctuations and estimate the local mean, we should clearly choose the weights so that , and then the operation is known as **Moving Average.** ([Chatfield 2016](#ref-chatfield2016analysis))

There are many different choices for the weights of the moving average such as Spencer’s 15 Point Moving average weights , Henderson’s Moving average weights etc. We have a relatively smaller dataset so undertaking the end effects we may use the simple moving average with a 6 month order to smooth the data . This can be easily done using the ma() function in stats package .



Both the plots are identical and we can observe that the moving average has successfully removed most of the random fluctuations within the series . If we consider our deterministic model to be where , and represents the original series ,Trend component and the random component respectively then we can consider the moving Average value to be a good representative of the trend component.

* Thus the calculated trend values are-

Table 1: Smoothing of monthly UPI Volume & Value of Transaction

*2016 to 2024*

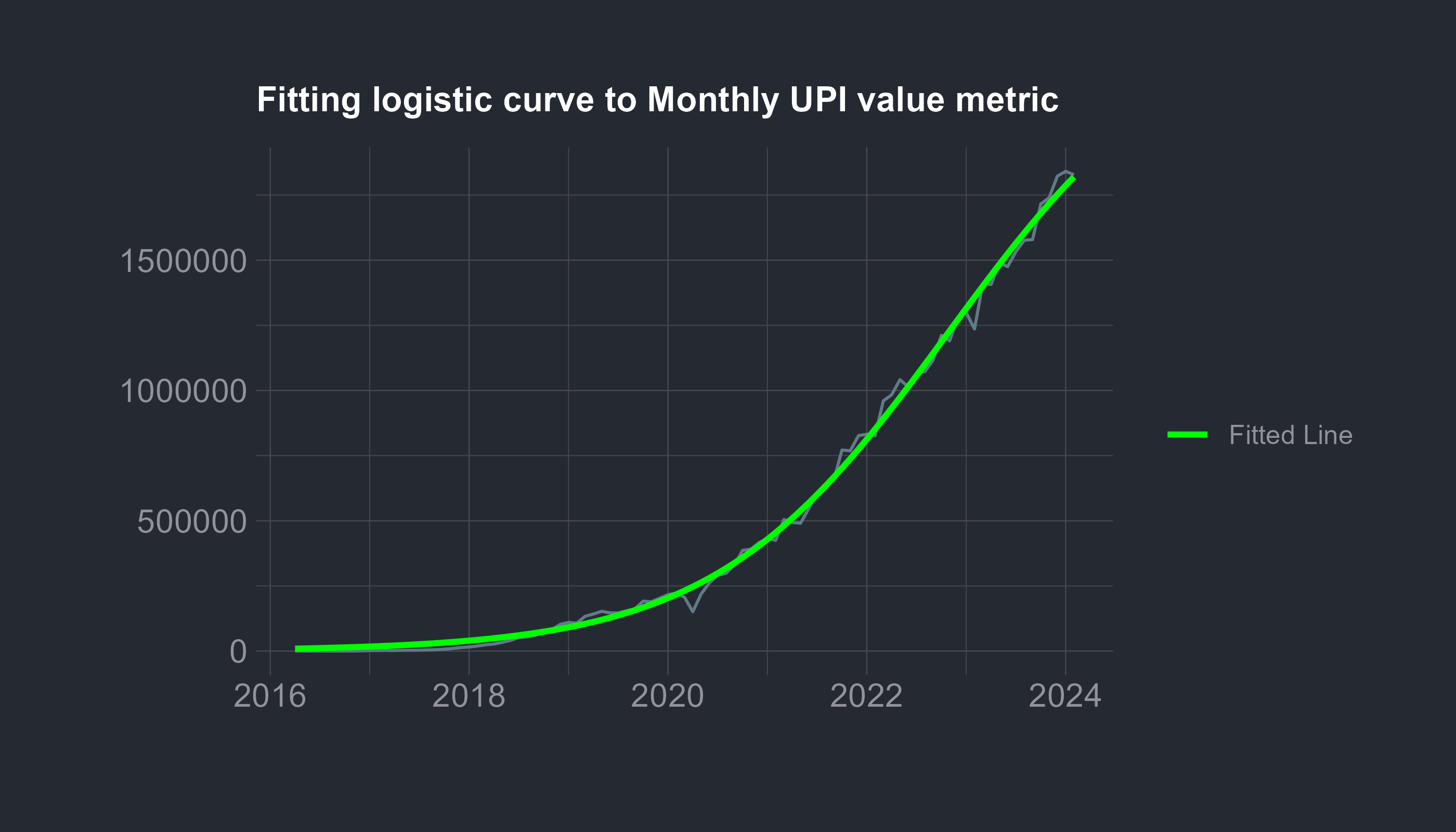
| seq.Date.as.Date..2016.4.1....by....month...length.out...95. | smth\_val | smth\_vol |
| --- | --- | --- |
| 2016-04-01 | NA | NA |
| 2016-05-01 | NA | NA |
| 2016-06-01 | NA | NA |
| 2016-07-01 | 1.006583e+01 | 5.333333e-03 |
| 2016-08-01 | 2.248500e+01 | 8.583333e-03 |
| 2016-09-01 | 8.985083e+01 | 2.758333e-02 |
| 2016-10-01 | 2.901650e+02 | 8.058333e-02 |
| 2016-11-01 | 5.927033e+02 | 1.527500e-01 |
| 2016-12-01 | 9.532967e+02 | 2.408333e-01 |
| 2017-01-01 | 1.337894e+03 | 3.523333e-01 |
| 2017-02-01 | 1.747834e+03 | 4.870833e-01 |
| 2017-03-01 | 2.171754e+03 | 6.323333e-01 |
| 2017-04-01 | 2.513884e+03 | 7.617500e-01 |
| 2017-05-01 | 2.841721e+03 | 9.250000e-01 |
| 2017-06-01 | 3.268352e+03 | 1.233583e+00 |
| 2017-07-01 | 3.908953e+03 | 2.020000e+00 |
| 2017-08-01 | 4.880520e+03 | 3.398500e+00 |
| 2017-09-01 | 6.292865e+03 | 5.323083e+00 |
| 2017-10-01 | 8.145842e+03 | 7.618833e+00 |
| 2017-11-01 | 1.040663e+04 | 1.007550e+01 |
| 2017-12-01 | 1.322466e+04 | 1.258942e+01 |
| 2018-01-01 | 1.645890e+04 | 1.475767e+01 |
| 2018-02-01 | 2.009083e+04 | 1.640417e+01 |
| 2018-03-01 | 2.436408e+04 | 1.794742e+01 |
| 2018-04-01 | 2.969173e+04 | 1.980283e+01 |
| 2018-05-01 | 3.563823e+04 | 2.199067e+01 |
| 2018-06-01 | 4.153396e+04 | 2.506100e+01 |
| 2018-07-01 | 4.850223e+04 | 2.939517e+01 |
| 2018-08-01 | 5.657724e+04 | 3.462633e+01 |
| 2018-09-01 | 6.580261e+04 | 4.053683e+01 |
| 2018-10-01 | 7.579012e+04 | 4.697683e+01 |
| 2018-11-01 | 8.500796e+04 | 5.331992e+01 |
| 2018-12-01 | 9.552048e+04 | 5.961858e+01 |
| 2019-01-01 | 1.072439e+05 | 6.539442e+01 |
| 2019-02-01 | 1.186834e+05 | 6.962800e+01 |
| 2019-03-01 | 1.281991e+05 | 7.248608e+01 |
| 2019-04-01 | 1.349012e+05 | 7.485200e+01 |
| 2019-05-01 | 1.419197e+05 | 7.813283e+01 |
| 2019-06-01 | 1.482334e+05 | 8.146317e+01 |
| 2019-07-01 | 1.546768e+05 | 8.581358e+01 |
| 2019-08-01 | 1.618523e+05 | 9.291192e+01 |
| 2019-09-01 | 1.695801e+05 | 1.015710e+02 |
| 2019-10-01 | 1.800643e+05 | 1.102092e+02 |
| 2019-11-01 | 1.915534e+05 | 1.176265e+02 |
| 2019-12-01 | 2.009715e+05 | 1.234528e+02 |
| 2020-01-01 | 2.013704e+05 | 1.246447e+02 |
| 2020-02-01 | 2.004490e+05 | 1.235359e+02 |
| 2020-03-01 | 2.078221e+05 | 1.239047e+02 |
| 2020-04-01 | 2.189562e+05 | 1.257453e+02 |
| 2020-05-01 | 2.314633e+05 | 1.297910e+02 |
| 2020-06-01 | 2.479930e+05 | 1.368447e+02 |
| 2020-07-01 | 2.777872e+05 | 1.503892e+02 |
| 2020-08-01 | 3.117517e+05 | 1.674541e+02 |
| 2020-09-01 | 3.389974e+05 | 1.830621e+02 |
| 2020-10-01 | 3.635795e+05 | 1.972504e+02 |
| 2020-11-01 | 3.858628e+05 | 2.095791e+02 |
| 2020-12-01 | 4.110806e+05 | 2.229592e+02 |
| 2021-01-01 | 4.346986e+05 | 2.354673e+02 |
| 2021-02-01 | 4.519650e+05 | 2.429572e+02 |
| 2021-03-01 | 4.712014e+05 | 2.504796e+02 |
| 2021-04-01 | 4.967260e+05 | 2.631332e+02 |
| 2021-05-01 | 5.291555e+05 | 2.815311e+02 |
| 2021-06-01 | 5.594488e+05 | 2.997417e+02 |
| 2021-07-01 | 5.950527e+05 | 3.205767e+02 |
| 2021-08-01 | 6.413509e+05 | 3.474476e+02 |
| 2021-09-01 | 6.877903e+05 | 3.758284e+02 |
| 2021-10-01 | 7.298892e+05 | 4.018961e+02 |
| 2021-11-01 | 7.643424e+05 | 4.214067e+02 |
| 2021-12-01 | 8.055054e+05 | 4.441007e+02 |
| 2022-01-01 | 8.486793e+05 | 4.700653e+02 |
| 2022-02-01 | 8.890911e+05 | 4.961747e+02 |
| 2022-03-01 | 9.274760e+05 | 5.217177e+02 |
| 2022-04-01 | 9.623538e+05 | 5.464486e+02 |
| 2022-05-01 | 1.002099e+06 | 5.774768e+02 |
| 2022-06-01 | 1.035583e+06 | 6.060376e+02 |
| 2022-07-01 | 1.067595e+06 | 6.318502e+02 |
| 2022-08-01 | 1.099041e+06 | 6.574887e+02 |
| 2022-09-01 | 1.133770e+06 | 6.851637e+02 |
| 2022-10-01 | 1.175720e+06 | 7.161239e+02 |
| 2022-11-01 | 1.208953e+06 | 7.386541e+02 |
| 2022-12-01 | 1.247041e+06 | 7.624843e+02 |
| 2023-01-01 | 1.287827e+06 | 7.916278e+02 |
| 2023-02-01 | 1.328991e+06 | 8.224483e+02 |
| 2023-03-01 | 1.369988e+06 | 8.525426e+02 |
| 2023-04-01 | 1.405682e+06 | 8.811533e+02 |
| 2023-05-01 | 1.453650e+06 | 9.226448e+02 |
| 2023-06-01 | 1.496098e+06 | 9.636586e+02 |
| 2023-07-01 | 1.535885e+06 | 1.000167e+03 |
| 2023-08-01 | 1.582498e+06 | 1.036257e+03 |
| 2023-09-01 | 1.632338e+06 | 1.073801e+03 |
| 2023-10-01 | 1.686915e+06 | 1.114831e+03 |
| 2023-11-01 | 1.733480e+06 | 1.146123e+03 |
| 2023-12-01 | NA | NA |
| 2024-01-01 | NA | NA |
| 2024-02-01 | NA | NA |
| *Method*: Simple Moving Average | | |

The downside of this method is that it can not be used to make future prediction and also there’s an effect of missing end values due to moving average. We can see that our estimated trend values are very close the original values which falls with our intial assumption that this time series is made up of trend and random error only.

* **Curve Fitting**- While fitting a deterministic function of time as a curve we have to first figure what kind of a function might properly represent our time series. Everett Rogers in his book Diffusion of Innovations([2003](#ref-rogers2003diffusion)) mentions “The logistic function can be used to illustrate the progress of the diffusion of an innovation through its life cycle” ,historically, when new products are introduced there is an intense amount of research and development which leads to dramatic improvements in quality and reductions in cost. This leads to a period of rapid industry growth. Some of the more famous examples are: railroads, incandescent light bulbs, electrification, cars and air travel. Eventually, dramatic improvement and cost reduction opportunities are exhausted, the product or process are in widespread use with few remaining potential new customers, and markets become saturated. UPI is a modern innovation which has revolutionized the way payments are done it may be a good idea to fit a logistic growth curve to the monthly value and volume data for UPI transactions.  
  *The Logistic Function in terms of time is given as*-
* where is the value of the time series at time t and a , b , k are constants.

⬜ There are many different methods to fit a logistic curve to our data most of these include long calculations for our ease we will use the SSlogis() function from stats package along with the nls() function in R, SSlogis() employs a self starting logistic function using the input data(Period of time) and calculates constants k( *Asymptote* ), b( *point of inflexion* ) and a ( *Scaling constant*) , while nls() uses the model given by SSlogis to fit the data using non linear least squares.

Plotting the Calculated model



From the fitted model we can see that our model choice was correct as the data seems to be very close to the fitted line. We can see the estimated values for the constants as -

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| k | 2.436074e+06 | 6.826906e+04 | 35.68342 | 0 |
| b | 7.974670e+01 | 8.953216e-01 | 89.07046 | 0 |
| a | 1.408733e+01 | 3.238176e-01 | 43.50392 | 0 |

from this the calculated equation becomes -

based on the equation the fitted values are

Table 1: Logistic Curve Fit of Monthly UPI transaction Value

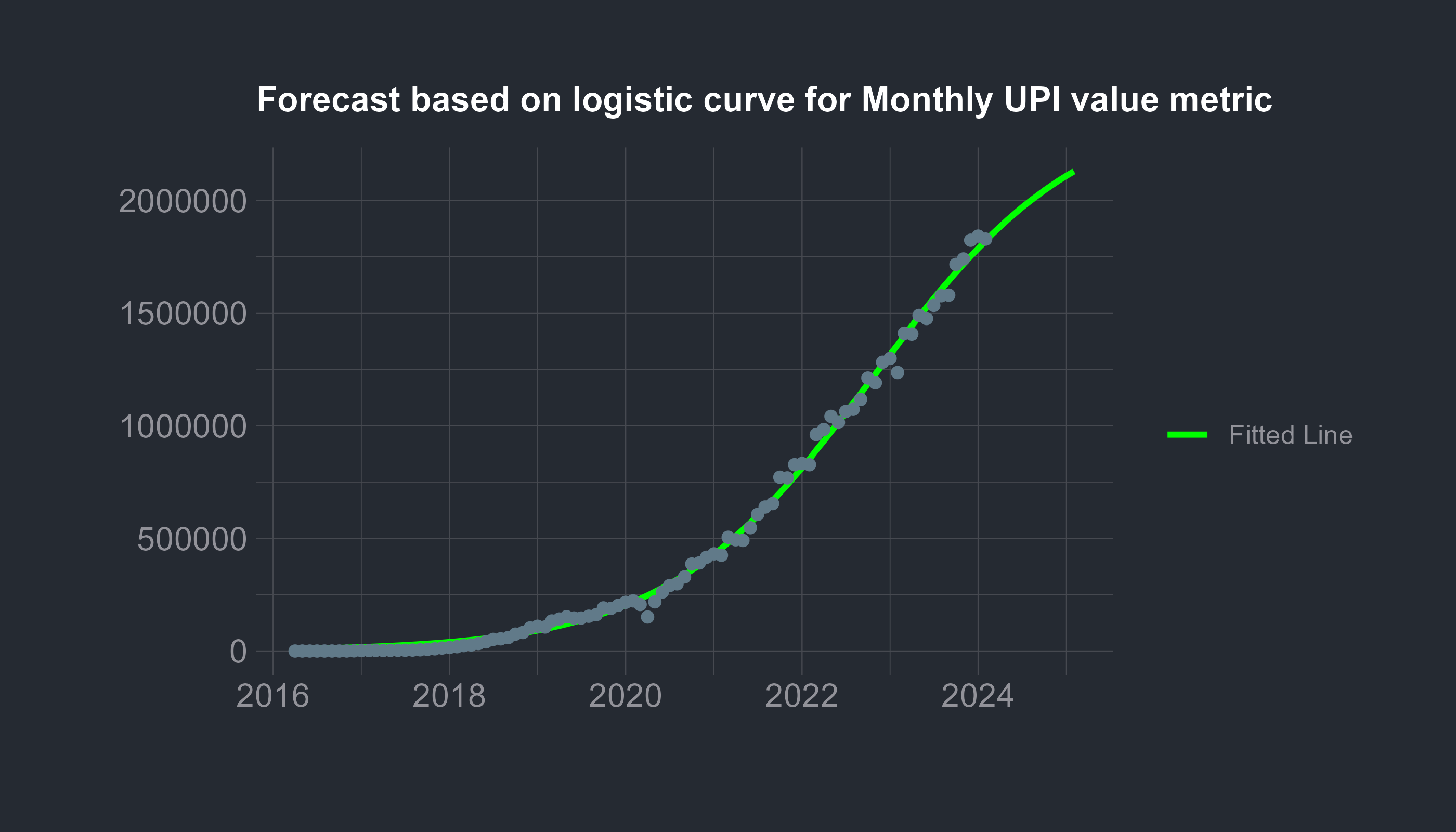
*2016 to 2024*

| Time | Original Value | Fitted Value |
| --- | --- | --- |
| 2016-04-01 | 0.00 | 9065.907 |
| 2016-05-01 | 0.00 | 9730.185 |
| 2016-06-01 | 0.00 | 10442.926 |
| 2016-07-01 | 0.38 | 11207.636 |
| 2016-08-01 | 3.09 | 12028.065 |
| 2016-09-01 | 32.64 | 12908.232 |
| 2016-10-01 | 48.57 | 13852.438 |
| 2016-11-01 | 100.46 | 14865.287 |
| 2016-12-01 | 707.93 | 15951.705 |
| 2017-01-01 | 1696.22 | 17116.961 |
| 2017-02-01 | 1937.71 | 18366.692 |
| 2017-03-01 | 2425.14 | 19706.925 |
| 2017-04-01 | 2271.24 | 21144.100 |
| 2017-05-01 | 2797.07 | 22685.100 |
| 2017-06-01 | 3098.36 | 24337.278 |
| 2017-07-01 | 3411.35 | 26108.484 |
| 2017-08-01 | 4156.62 | 28007.097 |
| 2017-09-01 | 5325.81 | 30042.056 |
| 2017-10-01 | 7057.78 | 32222.894 |
| 2017-11-01 | 9669.33 | 34559.772 |
| 2017-12-01 | 13174.24 | 37063.512 |
| 2018-01-01 | 15571.20 | 39745.638 |
| 2018-02-01 | 19126.20 | 42618.409 |
| 2018-03-01 | 24172.60 | 45694.862 |
| 2018-04-01 | 27021.85 | 48988.846 |
| 2018-05-01 | 33288.51 | 52515.066 |
| 2018-06-01 | 40834.03 | 56289.117 |
| 2018-07-01 | 51843.14 | 60327.531 |
| 2018-08-01 | 54212.26 | 64647.806 |
| 2018-09-01 | 59835.36 | 69268.451 |
| 2018-10-01 | 74978.27 | 74209.017 |
| 2018-11-01 | 82232.21 | 79490.135 |
| 2018-12-01 | 102594.82 | 85133.538 |
| 2019-01-01 | 109932.43 | 91162.096 |
| 2019-02-01 | 106737.12 | 97599.832 |
| 2019-03-01 | 133460.72 | 104471.936 |
| 2019-04-01 | 142034.39 | 111804.777 |
| 2019-05-01 | 152449.29 | 119625.901 |
| 2019-06-01 | 146566.35 | 127964.018 |
| 2019-07-01 | 146386.64 | 136848.980 |
| 2019-08-01 | 154504.89 | 146311.748 |
| 2019-09-01 | 161456.56 | 156384.338 |
| 2019-10-01 | 191359.94 | 167099.754 |
| 2019-11-01 | 189229.09 | 178491.902 |
| 2019-12-01 | 202520.76 | 190595.477 |
| 2020-01-01 | 216242.97 | 203445.834 |
| 2020-02-01 | 222516.95 | 217078.833 |
| 2020-03-01 | 206462.31 | 231530.647 |
| 2020-04-01 | 151140.66 | 246837.551 |
| 2020-05-01 | 218391.60 | 263035.680 |
| 2020-06-01 | 261835.00 | 280160.743 |
| 2020-07-01 | 290537.86 | 298247.717 |
| 2020-08-01 | 298307.61 | 317330.502 |
| 2020-09-01 | 329027.66 | 337441.537 |
| 2020-10-01 | 386106.74 | 358611.397 |
| 2020-11-01 | 390999.15 | 380868.345 |
| 2020-12-01 | 416176.21 | 404237.864 |
| 2021-01-01 | 431181.89 | 428742.169 |
| 2021-02-01 | 425062.76 | 454399.691 |
| 2021-03-01 | 504886.44 | 481224.557 |
| 2021-04-01 | 493663.68 | 509226.067 |
| 2021-05-01 | 490638.65 | 538408.170 |
| 2021-06-01 | 547373.17 | 568768.964 |
| 2021-07-01 | 606281.14 | 600300.220 |
| 2021-08-01 | 639116.95 | 632986.946 |
| 2021-09-01 | 654351.81 | 666807.003 |
| 2021-10-01 | 771444.98 | 701730.797 |
| 2021-11-01 | 768436.11 | 737721.034 |
| 2021-12-01 | 826848.22 | 774732.583 |
| 2022-01-01 | 831993.11 | 812712.437 |
| 2022-02-01 | 826843.00 | 851599.782 |
| 2022-03-01 | 960581.66 | 891326.194 |
| 2022-04-01 | 983302.27 | 931815.955 |
| 2022-05-01 | 1041520.00 | 972986.503 |
| 2022-06-01 | 1014384.00 | 1014748.995 |
| 2022-07-01 | 1062991.00 | 1057009.002 |
| 2022-08-01 | 1072792.68 | 1099667.300 |
| 2022-09-01 | 1116438.10 | 1142620.768 |
| 2022-10-01 | 1211582.51 | 1185763.359 |
| 2022-11-01 | 1190593.39 | 1228987.138 |
| 2022-12-01 | 1282055.01 | 1272183.354 |
| 2023-01-01 | 1298726.62 | 1315243.531 |
| 2023-02-01 | 1235846.62 | 1358060.559 |
| 2023-03-01 | 1410443.01 | 1400529.748 |
| 2023-04-01 | 1407007.55 | 1442549.834 |
| 2023-05-01 | 1489145.44 | 1484023.915 |
| 2023-06-01 | 1475464.27 | 1524860.300 |
| 2023-07-01 | 1533645.20 | 1564973.249 |
| 2023-08-01 | 1576536.56 | 1604283.603 |
| 2023-09-01 | 1579133.18 | 1642719.293 |
| 2023-10-01 | 1715768.34 | 1680215.717 |
| 2023-11-01 | 1739740.61 | 1716716.000 |
| 2023-12-01 | 1822949.42 | 1752171.123 |
| 2024-01-01 | 1841083.97 | 1786539.937 |
| 2024-02-01 | 1827869.33 | 1819789.073 |
| *Source*: NPCI | | |

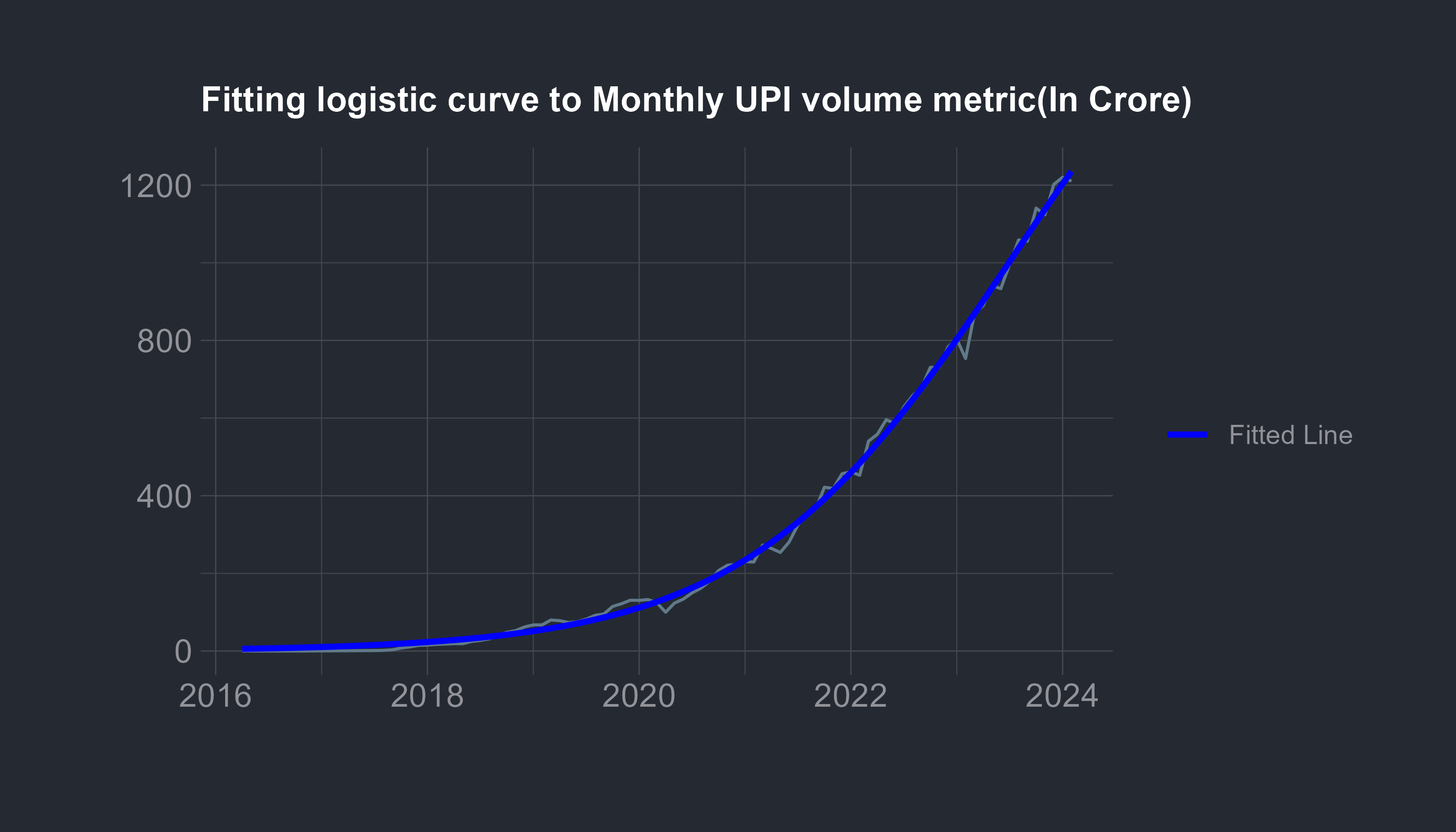
Based on this curve fitting the future estimates for the next 12 months will be

| Month | Predicted Value |
| --- | --- |
| 2024-03-01 | 1851893 |
| 2024-04-01 | 1882832 |
| 2024-05-01 | 1912597 |
| 2024-06-01 | 1941181 |
| 2024-07-01 | 1968585 |
| 2024-08-01 | 1994817 |
| 2024-09-01 | 2019888 |
| 2024-10-01 | 2043815 |
| 2024-11-01 | 2066618 |
| 2024-12-01 | 2088321 |
| 2025-01-01 | 2108951 |
| 2025-02-01 | 2128537 |

A plot with the future estimates looks something like this -



We can apply the similar methods for transaction volume which gives us -



| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| k | 2006.45018 | 81.6561480 | 24.57194 | 0 |
| b | 88.00355 | 1.2140952 | 72.48488 | 0 |
| a | 14.83489 | 0.3258464 | 45.52724 | 0 |

from this the calculated equation becomes -

based on the equation the fitted values are

Table 1: Logistic Curve Fit of Monthly UPI transaction Volume

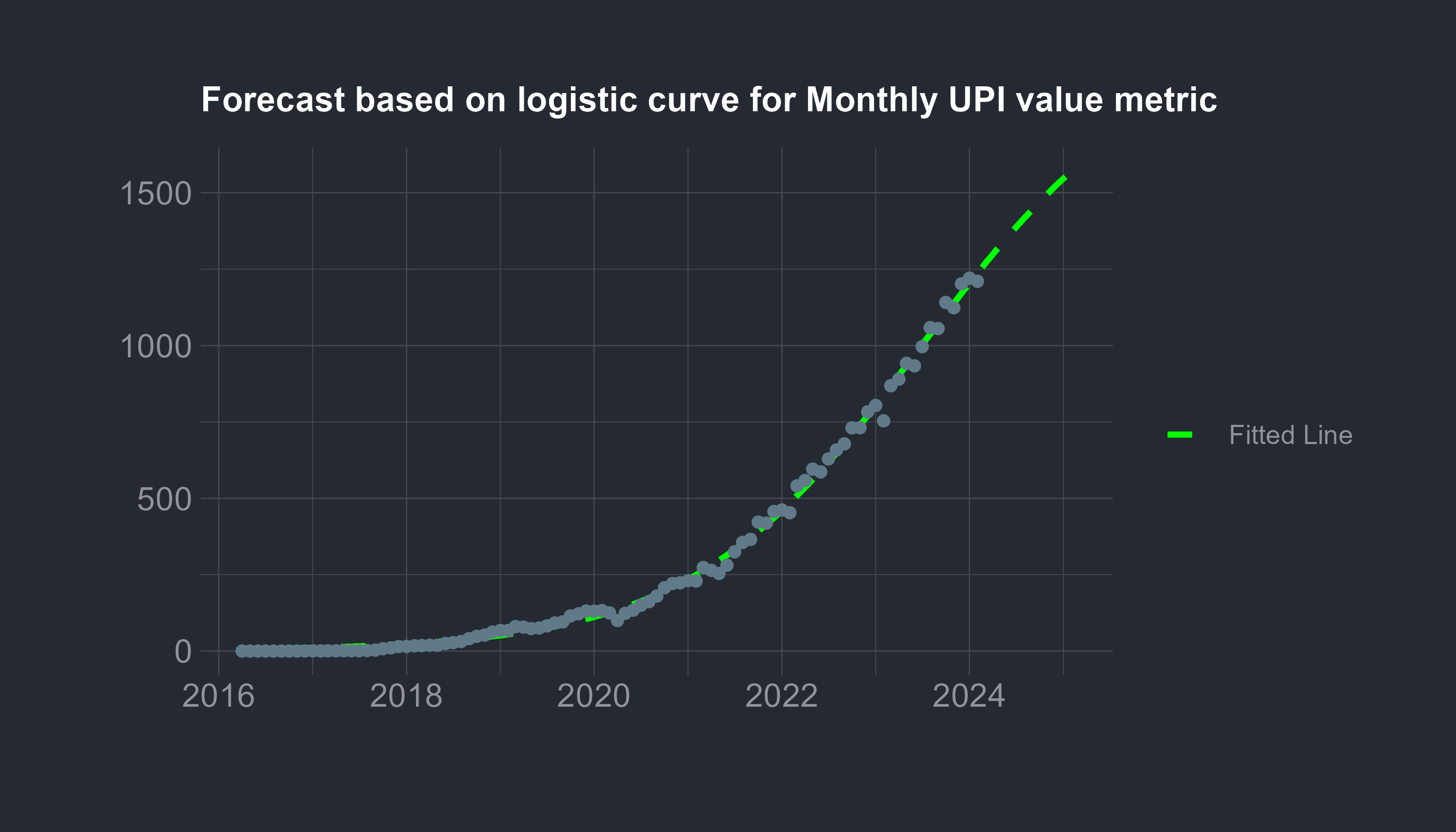
*2016 to 2024*

| Time | Original Volume | Fitted Volume |
| --- | --- | --- |
| 2016-04-01 | 0.000 | 5.677418 |
| 2016-05-01 | 0.000 | 6.072121 |
| 2016-06-01 | 0.000 | 6.494175 |
| 2016-07-01 | 0.009 | 6.945462 |
| 2016-08-01 | 0.009 | 7.427994 |
| 2016-09-01 | 0.009 | 7.943916 |
| 2016-10-01 | 0.010 | 8.495520 |
| 2016-11-01 | 0.029 | 9.085252 |
| 2016-12-01 | 0.199 | 9.715722 |
| 2017-01-01 | 0.446 | 10.389716 |
| 2017-02-01 | 0.438 | 11.110205 |
| 2017-03-01 | 0.637 | 11.880361 |
| 2017-04-01 | 0.720 | 12.703563 |
| 2017-05-01 | 0.936 | 13.583418 |
| 2017-06-01 | 1.035 | 14.523768 |
| 2017-07-01 | 1.163 | 15.528709 |
| 2017-08-01 | 1.680 | 16.602605 |
| 2017-09-01 | 3.098 | 17.750105 |
| 2017-10-01 | 7.696 | 18.976158 |
| 2017-11-01 | 10.502 | 20.286035 |
| 2017-12-01 | 14.564 | 21.685343 |
| 2018-01-01 | 15.183 | 23.180048 |
| 2018-02-01 | 17.140 | 24.776491 |
| 2018-03-01 | 17.805 | 26.481416 |
| 2018-04-01 | 19.008 | 28.301985 |
| 2018-05-01 | 18.948 | 30.245804 |
| 2018-06-01 | 24.637 | 32.320946 |
| 2018-07-01 | 27.375 | 34.535974 |
| 2018-08-01 | 31.202 | 36.899966 |
| 2018-09-01 | 40.587 | 39.422537 |
| 2018-10-01 | 48.236 | 42.113871 |
| 2018-11-01 | 52.494 | 44.984737 |
| 2018-12-01 | 62.017 | 48.046521 |
| 2019-01-01 | 67.275 | 51.311246 |
| 2019-02-01 | 67.419 | 54.791601 |
| 2019-03-01 | 79.954 | 58.500959 |
| 2019-04-01 | 78.179 | 62.453402 |
| 2019-05-01 | 73.354 | 66.663741 |
| 2019-06-01 | 75.454 | 71.147536 |
| 2019-07-01 | 82.229 | 75.921106 |
| 2019-08-01 | 91.835 | 81.001549 |
| 2019-09-01 | 95.502 | 86.406745 |
| 2019-10-01 | 114.836 | 92.155365 |
| 2019-11-01 | 121.877 | 98.266865 |
| 2019-12-01 | 130.840 | 104.761484 |
| 2020-01-01 | 130.502 | 111.660224 |
| 2020-02-01 | 132.569 | 118.984836 |
| 2020-03-01 | 124.684 | 126.757779 |
| 2020-04-01 | 99.957 | 135.002187 |
| 2020-05-01 | 123.450 | 143.741812 |
| 2020-06-01 | 133.693 | 153.000958 |
| 2020-07-01 | 149.736 | 162.804403 |
| 2020-08-01 | 161.883 | 173.177307 |
| 2020-09-01 | 180.014 | 184.145099 |
| 2020-10-01 | 207.162 | 195.733348 |
| 2020-11-01 | 221.023 | 207.967620 |
| 2020-12-01 | 223.416 | 220.873315 |
| 2021-01-01 | 230.273 | 234.475475 |
| 2021-02-01 | 229.290 | 248.798585 |
| 2021-03-01 | 273.168 | 263.866345 |
| 2021-04-01 | 264.106 | 279.701425 |
| 2021-05-01 | 253.957 | 296.325199 |
| 2021-06-01 | 280.751 | 313.757464 |
| 2021-07-01 | 324.782 | 332.016138 |
| 2021-08-01 | 355.555 | 351.116945 |
| 2021-09-01 | 365.430 | 371.073098 |
| 2021-10-01 | 421.865 | 391.894956 |
| 2021-11-01 | 418.648 | 413.589699 |
| 2021-12-01 | 456.630 | 436.160993 |
| 2022-01-01 | 461.715 | 459.608666 |
| 2022-02-01 | 452.749 | 483.928401 |
| 2022-03-01 | 540.565 | 509.111448 |
| 2022-04-01 | 558.305 | 535.144370 |
| 2022-05-01 | 595.520 | 562.008822 |
| 2022-06-01 | 586.275 | 589.681375 |
| 2022-07-01 | 628.840 | 618.133395 |
| 2022-08-01 | 657.963 | 647.330975 |
| 2022-09-01 | 678.080 | 677.234942 |
| 2022-10-01 | 730.542 | 707.800923 |
| 2022-11-01 | 730.945 | 738.979491 |
| 2022-12-01 | 782.949 | 770.716385 |
| 2023-01-01 | 803.689 | 802.952808 |
| 2023-02-01 | 753.476 | 835.625798 |
| 2023-03-01 | 868.530 | 868.668673 |
| 2023-04-01 | 889.814 | 902.011534 |
| 2023-05-01 | 941.519 | 935.581838 |
| 2023-06-01 | 933.506 | 969.305012 |
| 2023-07-01 | 996.461 | 1003.105105 |
| 2023-08-01 | 1058.602 | 1036.905470 |
| 2023-09-01 | 1055.569 | 1070.629458 |
| 2023-10-01 | 1140.879 | 1104.201112 |
| 2023-11-01 | 1123.529 | 1137.545846 |
| 2023-12-01 | 1202.023 | 1170.591099 |
| 2024-01-01 | 1220.302 | 1203.266954 |
| 2024-02-01 | 1210.268 | 1235.506702 |
| *Source*: NPCI | | |

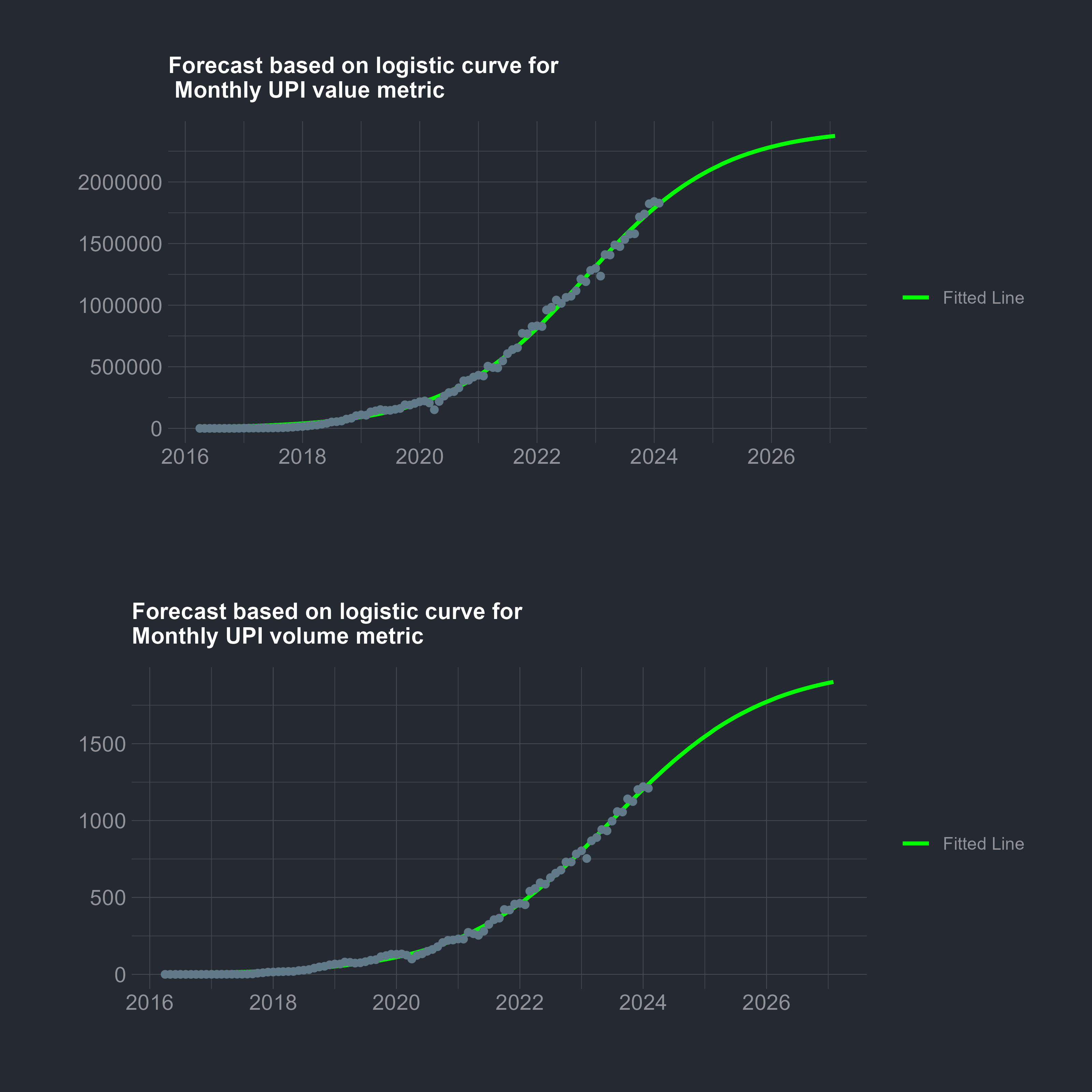
Based on this curve fitting the future estimates for the next 12 months will be

| Month | Predicted Value |
| --- | --- |
| 2024-03-01 | 1267.247 |
| 2024-04-01 | 1298.430 |
| 2024-05-01 | 1329.001 |
| 2024-06-01 | 1358.909 |
| 2024-07-01 | 1388.112 |
| 2024-08-01 | 1416.570 |
| 2024-09-01 | 1444.248 |
| 2024-10-01 | 1471.118 |
| 2024-11-01 | 1497.157 |
| 2024-12-01 | 1522.346 |
| 2025-01-01 | 1546.672 |
| 2025-02-01 | 1570.126 |

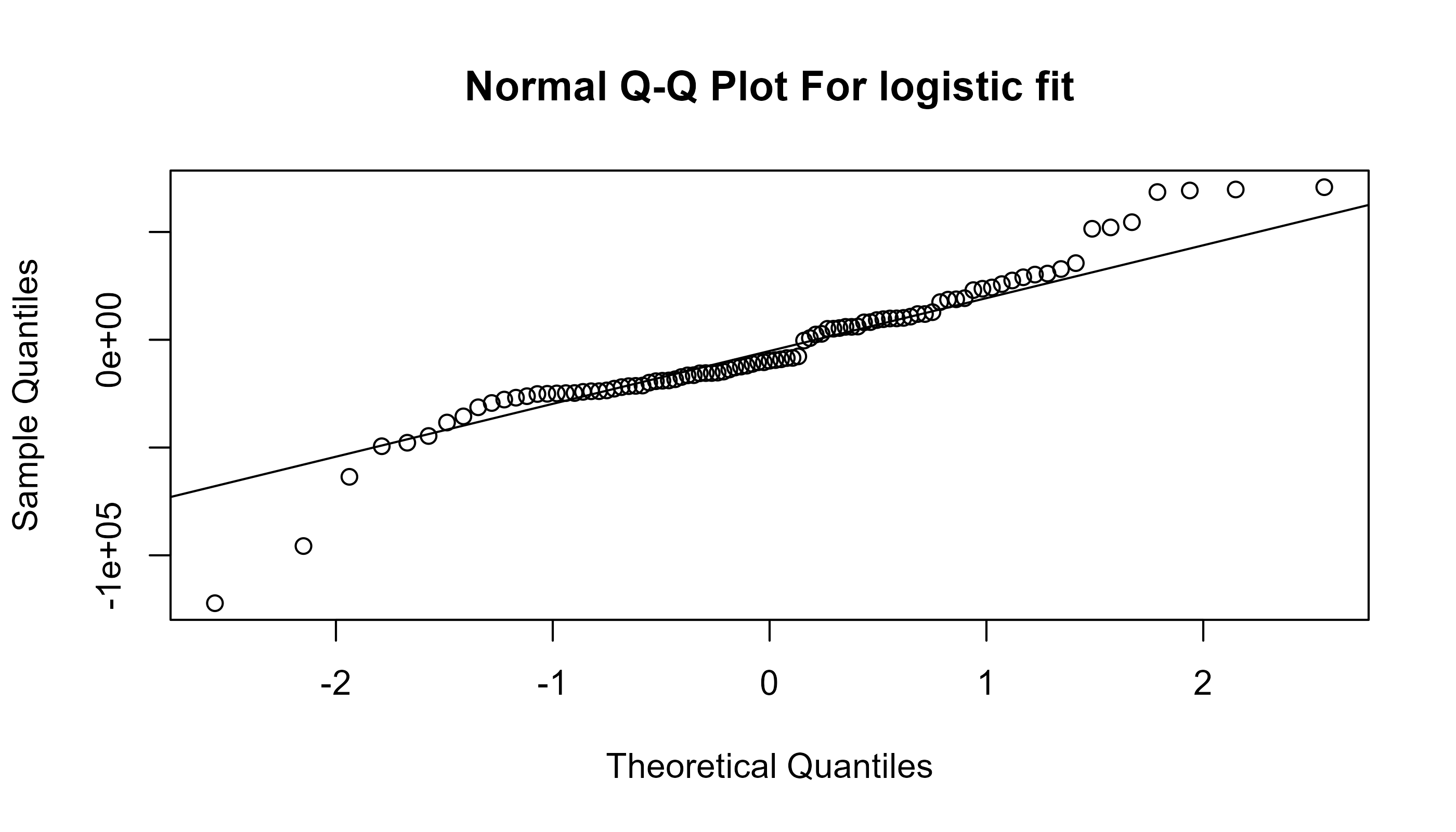
A plot with the future estimates looks something like this -



A long term prediction for both value and volume can be given via a plot as -



Since our assumed model is non linear is not suitable as a model adequacy checker,to overcome this we may check the residuals via a normal qqplot



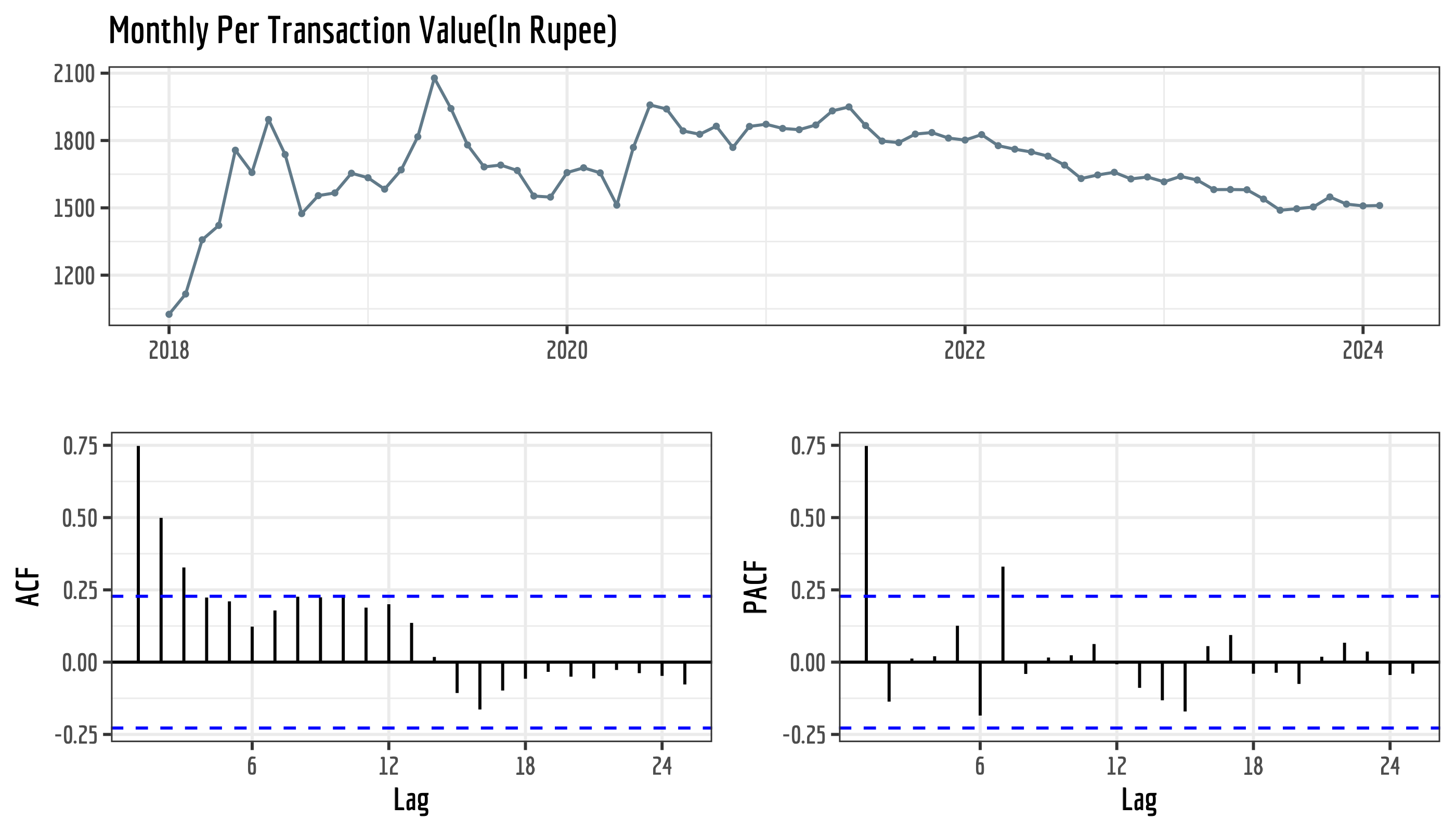
The residuals are very much linear and seem to be close to a standard normal variate.We can conclude that our model fit seems to be decent.

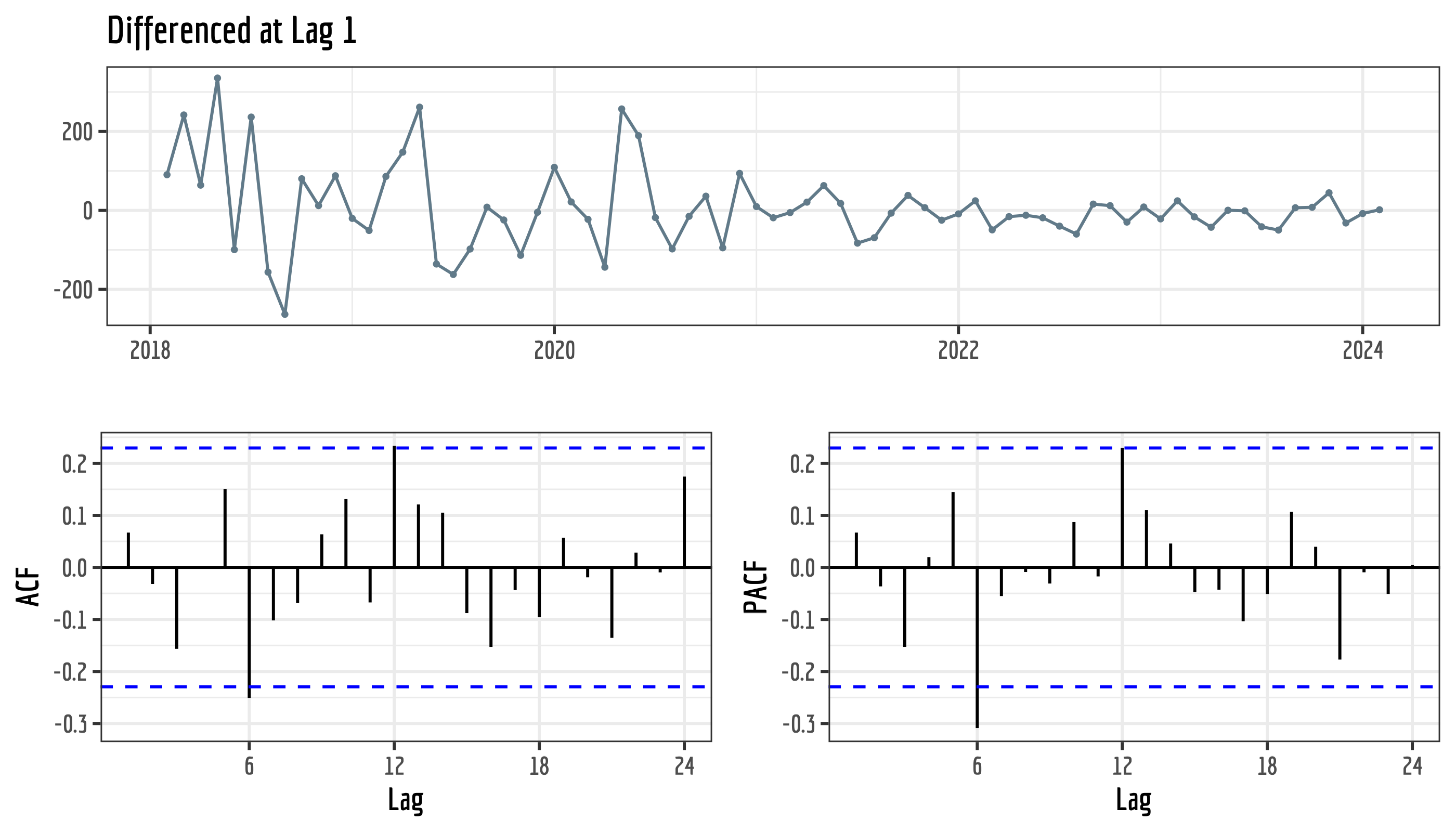
#### Foundings-

* We find that by the year 2025 Volume of monthly transactions will cross 1500 crores and value of monthly transactions will cross 2000000 crores.
* We can also expect that by the year 2027 both of these metrics will start to stabilize, although this is dependent on many other factors which we have not considered in our study these include avaliability of smartphones and fast internet connection for the percentage of population. But never the less this is a expectable figure.

## Stochastic Analysis of monthly Per Transaction Value

Previously From the EDA we found that the monthly per transaction value is decreasing with time , we also found that this data was visually stationary if we ignore the initial instability.For forecasting and analysis of this time series we first plot the time series along with its ACF and PACF.





We can see that the ACF has cut off lag at 1 and the PACF has a significant value in lag 1. Post differencing there are no significant autocorrelation values for the time series. Assuming our time series does not contain any seasonal effect we can define our differenced time series as a random walk model.

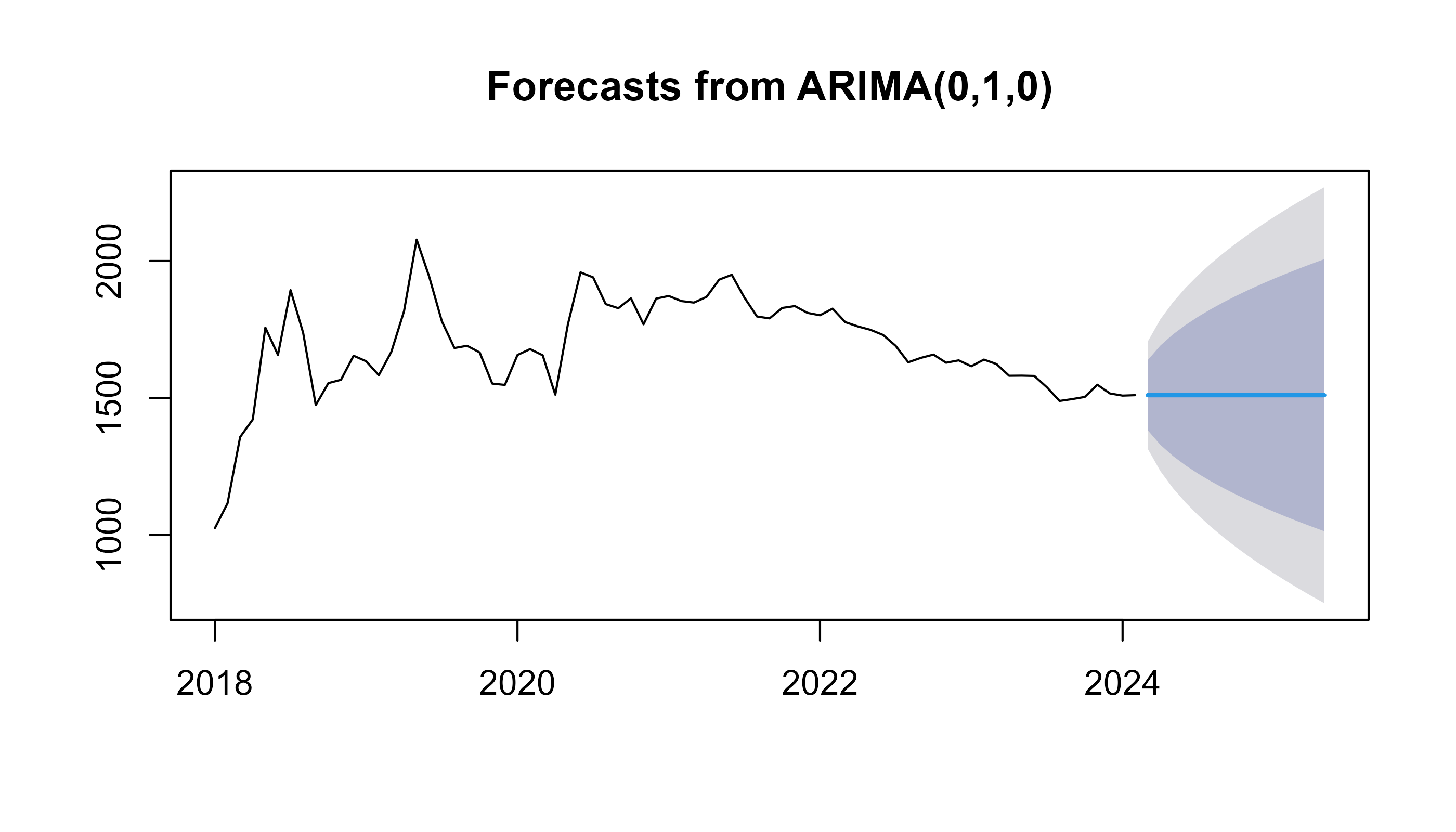
Which in ARIMA terms is written as ARIMA(0,1,0). We now check the auto.arima() output and see if our intuition about the model is correct.

## Series: .   
## ARIMA(0,1,0)   
##   
## sigma^2 = 9998: log likelihood = -439.75  
## AIC=881.51 AICc=881.56 BIC=883.8

Accuracy Measures for ARIMA(0,1,0) fit

|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Training set | 6.564308 | 99.31256 | 65.35913 | 0.3437234 | 3.870788 | 0.3991829 | 0.0661837 |

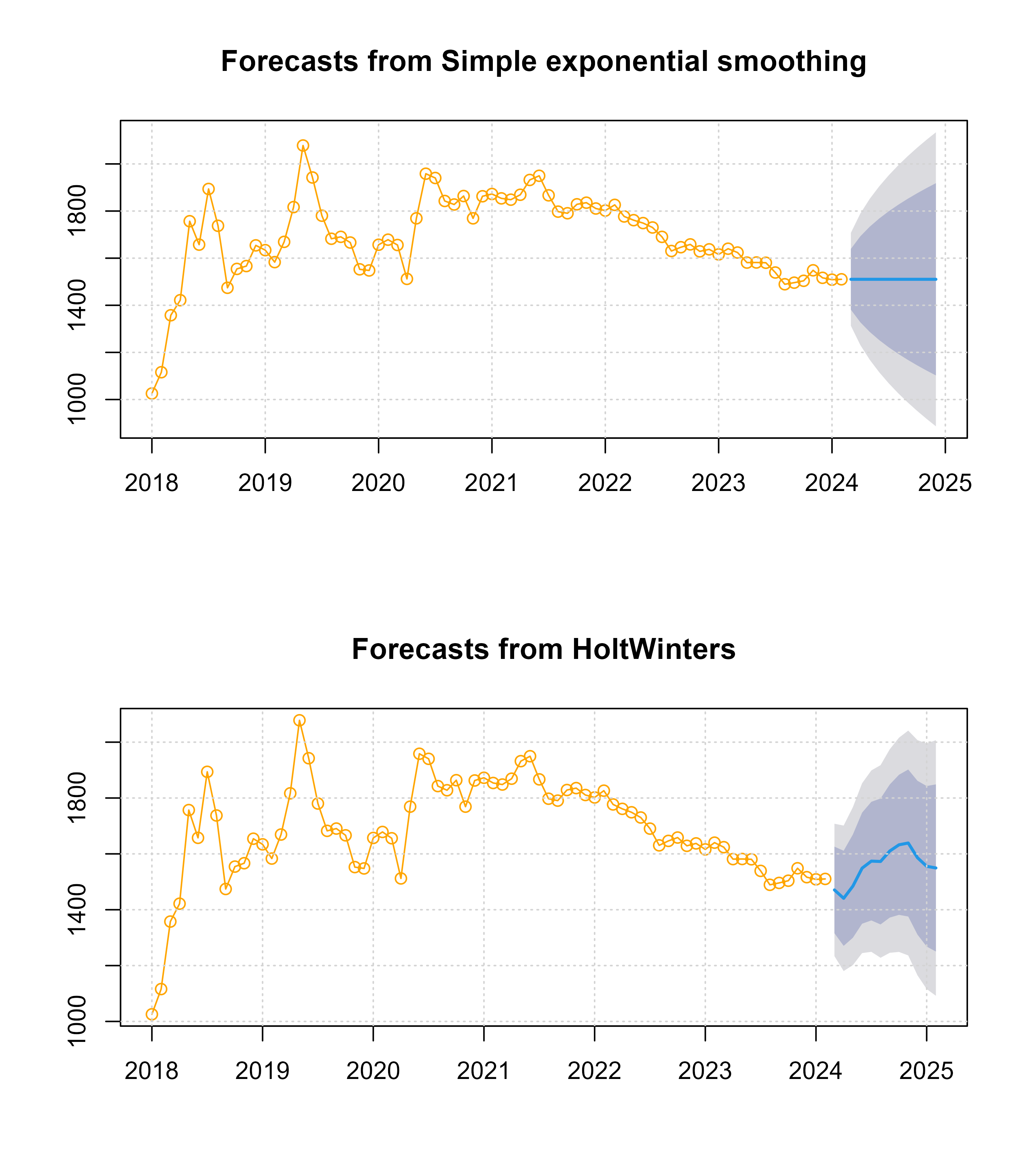
We can see that the MAPE value is less than 10 which usually implies a very good model fit. Also since our data does not contains zero values MAPE is actually a good model accuracy indicator. We now plot the forecast using this ARIMA(0,1,0) model



Forecasts from ARIMA(0,1,0)

|  | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
| --- | --- | --- | --- | --- | --- |
| Mar 2024 | 1510.301 | 1382.158 | 1638.444 | 1314.3236 | 1706.279 |
| Apr 2024 | 1510.301 | 1329.080 | 1691.523 | 1233.1470 | 1787.456 |
| May 2024 | 1510.301 | 1288.351 | 1732.251 | 1170.8579 | 1849.745 |
| Jun 2024 | 1510.301 | 1254.015 | 1766.587 | 1118.3459 | 1902.257 |
| Jul 2024 | 1510.301 | 1223.765 | 1796.838 | 1072.0818 | 1948.521 |
| Aug 2024 | 1510.301 | 1196.416 | 1824.186 | 1030.2559 | 1990.347 |
| Sep 2024 | 1510.301 | 1171.267 | 1849.336 | 991.7930 | 2028.810 |
| Oct 2024 | 1510.301 | 1147.858 | 1872.744 | 955.9926 | 2064.610 |
| Nov 2024 | 1510.301 | 1125.872 | 1894.730 | 922.3682 | 2098.234 |
| Dec 2024 | 1510.301 | 1105.078 | 1915.525 | 890.5654 | 2130.037 |
| Jan 2025 | 1510.301 | 1085.299 | 1935.303 | 860.3168 | 2160.286 |
| Feb 2025 | 1510.301 | 1066.401 | 1954.201 | 831.4146 | 2189.188 |
| Mar 2025 | 1510.301 | 1048.275 | 1972.327 | 803.6936 | 2216.909 |
| Apr 2025 | 1510.301 | 1030.834 | 1989.768 | 777.0199 | 2243.583 |
| May 2025 | 1510.301 | 1014.006 | 2006.597 | 751.2829 | 2269.320 |

we can see that the predicted forecast is constant in fact it is equal to the last observation. This is due to the fact that for random walks only naive predictions are possible since they do not contain any significant pattern as such. We may plot some other forecasts like Simple Exponential Smoothing and Holt-Winters Exponential Smoothing.



As it can be seen SES gives us a naive constant forecast which is the same as the ARIMA forecast. Holt-Winters on the other hand gives us a rather interesting looking prediction, the predicted values are given as -

Forecasts from Holt-Winters Exponential Smoothing

|  | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
| --- | --- | --- | --- | --- | --- |
| Mar 2024 | 1470.844 | 1315.831 | 1625.857 | 1233.773 | 1707.915 |
| Apr 2024 | 1440.896 | 1270.556 | 1611.236 | 1180.383 | 1701.409 |
| May 2024 | 1484.525 | 1299.628 | 1669.421 | 1201.750 | 1767.299 |
| Jun 2024 | 1548.835 | 1349.976 | 1747.693 | 1244.706 | 1852.963 |
| Jul 2024 | 1574.016 | 1361.663 | 1786.369 | 1249.250 | 1898.781 |
| Aug 2024 | 1572.806 | 1347.336 | 1798.276 | 1227.979 | 1917.632 |
| Sep 2024 | 1610.208 | 1371.928 | 1848.487 | 1245.791 | 1974.625 |
| Oct 2024 | 1632.474 | 1381.639 | 1883.309 | 1248.855 | 2016.093 |
| Nov 2024 | 1638.886 | 1375.707 | 1902.065 | 1236.388 | 2041.383 |
| Dec 2024 | 1586.350 | 1311.004 | 1861.697 | 1165.244 | 2007.456 |
| Jan 2025 | 1555.521 | 1268.155 | 1842.886 | 1116.033 | 1995.008 |
| Feb 2025 | 1549.546 | 1250.286 | 1848.805 | 1091.868 | 2007.223 |

Here is the accuracy measure for this model.

Accuracy Measures for Holt-Winters Method

|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Training set | -23.45331 | 122.2484 | 92.50533 | -1.429825 | 5.393558 | 0.5649791 | 0.4701516 |

#### Foundings

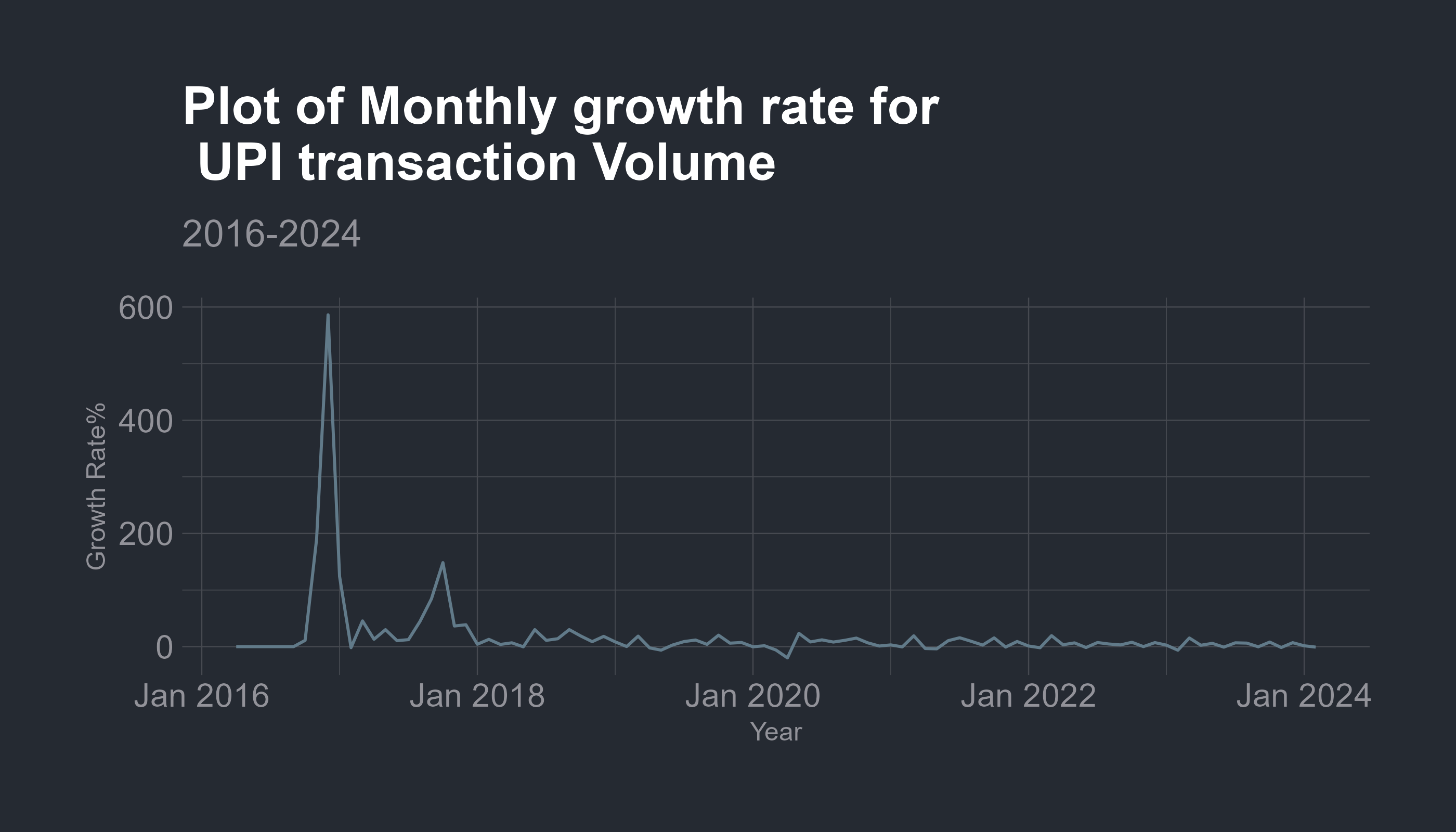
* We have found that per transaction value is decreasing with time, and as we can assume this value will constantly decrease to e certain point after which this will stabilize.This data is very important as it tells us about the user patterns of UPI users and how it has changed through the years. UPI is constantly being used for more lower worth transactions which means it is getting even more implemented to our daily life, it is becoming a decent substitute for cash and is increasing financial inclusivity among people. The ease of UPI transactions also is very helpful towards MSME’s and this is a good opportunity for them to use this information and use UPI specific offers to attract more customers.

## Analyzing monthly growth rate for transaction volume.

We first calculate the monthly growth rate relative to the past month with the help of a simple function₹.

#Calculating growth rate####  
dat<-list()  
#This Function Calculates the growth rate#  
month\_growth<-  
 function(data,returndat)  
 {  
 returndat[1]=0;  
 for(i in 2:length(data)){  
 if(data[i-1]>0)  
 {  
 returndat[i]=(((data[i]-data[i-1])/data[i-1])\*100)  
 }  
 else if(data[i-1]==0)  
 returndat[i]=0  
 }  
 returndat  
 }  
growth<-matrix(month\_growth(as.numeric(data1$`Volume(In Cr)`),dat),ncol=1)  
growth<-data.frame(as.numeric(growth))  
colnames(growth)<-c("GrowthRate")

The function calculates the monthly growth rate relative to the previous month.



Plotting this monthly growth rate we see the initial values are very high so we are not able to properly visualize the change, assuming these to be outlier since it is expected that initial growth rate will be very high we may ignore these values and re-plot the values excluding the first few observations.

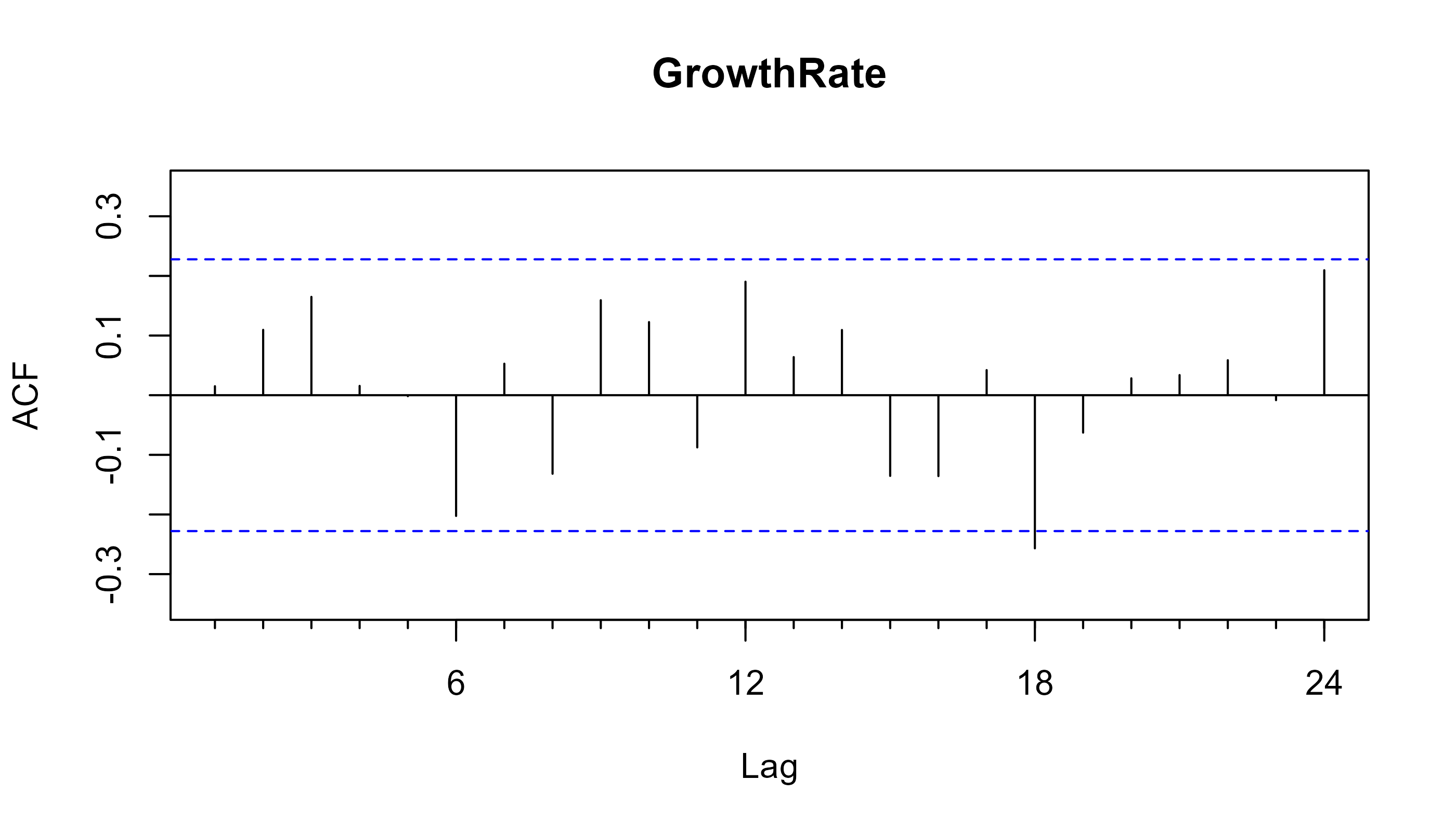


Visually the data looks kind of stationary , to confirm our assumption we may use **Augmented Dickey Fueller** Test to look for unit roots and find if the data is truly stationary or not and also find the lag order. We assume the significance level to be 0.05.

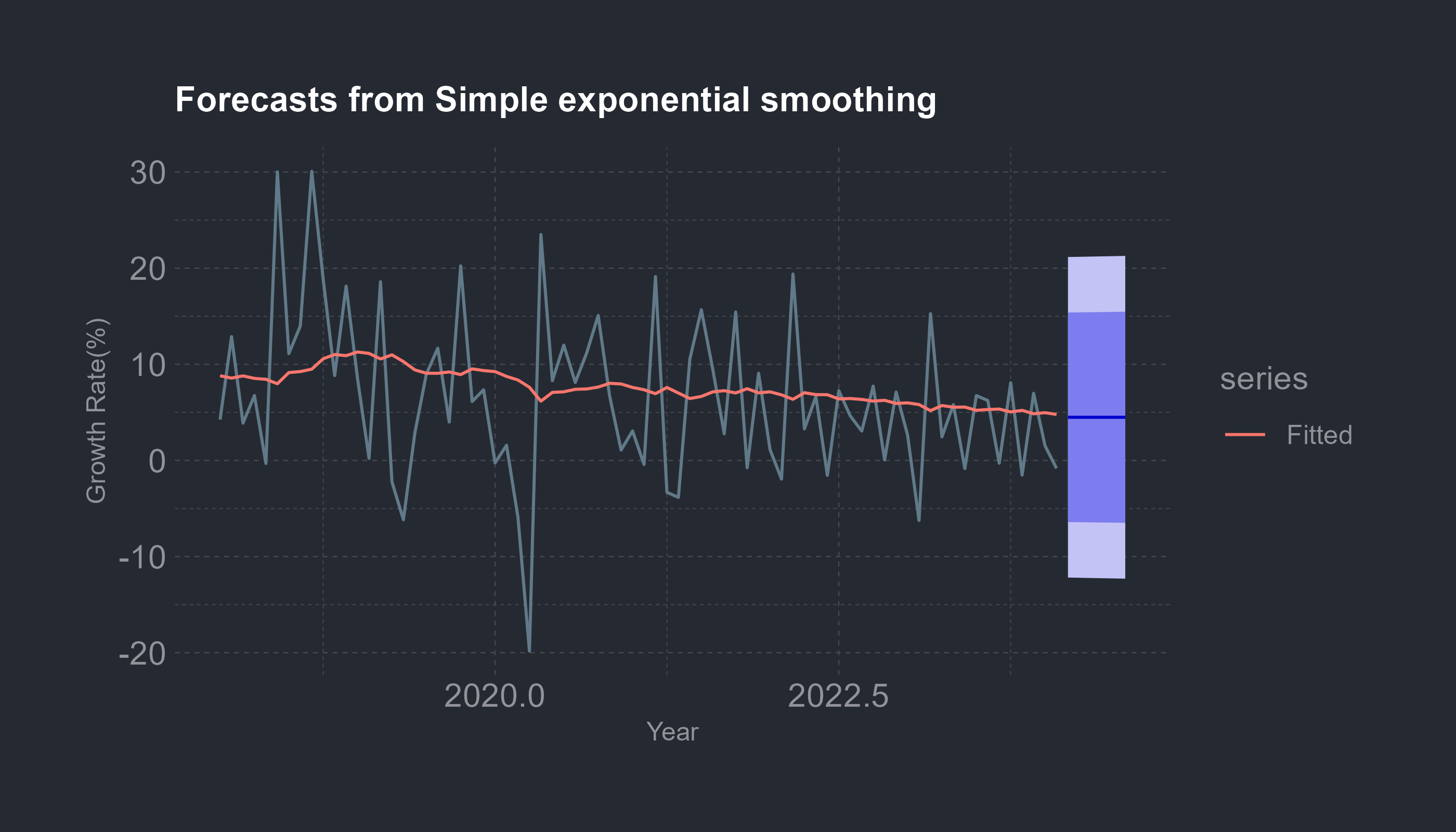
| statistic | p.value | parameter | method | alternative |
| --- | --- | --- | --- | --- |
| -4.070941 | 0.0112874 | 4 | Augmented Dickey-Fuller Test | stationary |

We can observe that the p value for the test is less than our assumed significance level of 0.05 .So we may reject the null hypothesis and conclude that our data is stationary.

We may also analyze the Autocorrelation for this time series.



from the plot we find the that the process acf is identical to a **white noise** process.We can make some short term predictions for the monthly growth rate using a **Simple Exponential Smoothing** forecast we do this using the ses() function in forecast package.



We can see that the forecast model is naive and the fit isn’t very identical. The forecasts for future growth rate are given around 5% positive growth for the next months. This is a simple and naive forecast so it won’t be absolutely perfect.But it does give us some idea.

Accuracy Measure for Holt-Winters

|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Training set | -1.11 | 8.39 | 6.49 | -32 | 503.39 | 0.96 | -0.06 |

We can see that the accuracy measures aren’t very satisfactory as well. We may use the HoltWinters() function to apply the *Holt-Winters Exponential* smoothing for a better forecast.

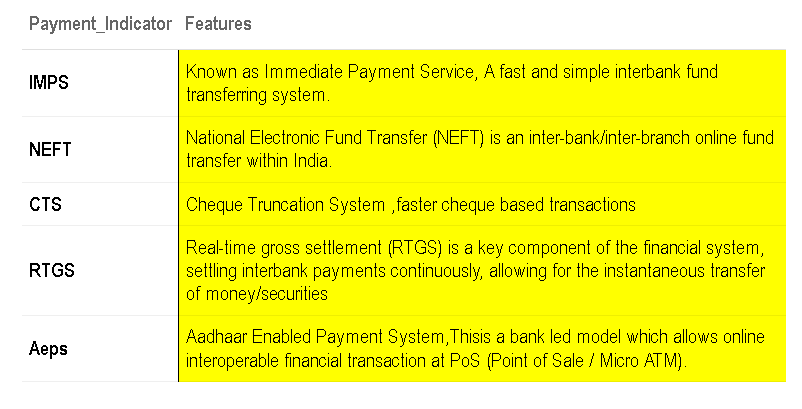


We can see that the forecast lags a little behind the values at certain points within the sample.Outside the sample the prediction are given near 15% for the next month with values dropping and staying around 5% to 8% positive growth.

# Analysing daily UPI transactions(2016-2018)

This data has been collected from RBI daily payment system indicators.This data is daily updated by RBI and is provided in the form of a excel workbook with multiple sheets where each sheet contains data about every months data from 2020 to the most latest data available. The data was in a format with multiple sub-columns within each column ,R isn’t well suited for handling this kind of data so first a power query was run through the excel file to combine multiple sheets into a single sheet . The original file contained more columns and data about other digital payment metrics as well but since these data were added in different intervals of time i had to scrap some of them.Some of the columns contain 0 values these are bank dependent payment methods so they are turned off during bank holidays( some Saturday’s and Sunday’s and other bank holidays ) .

#### Brief idea about the different payment indicators except UPI



## Exploratory Data Analysis

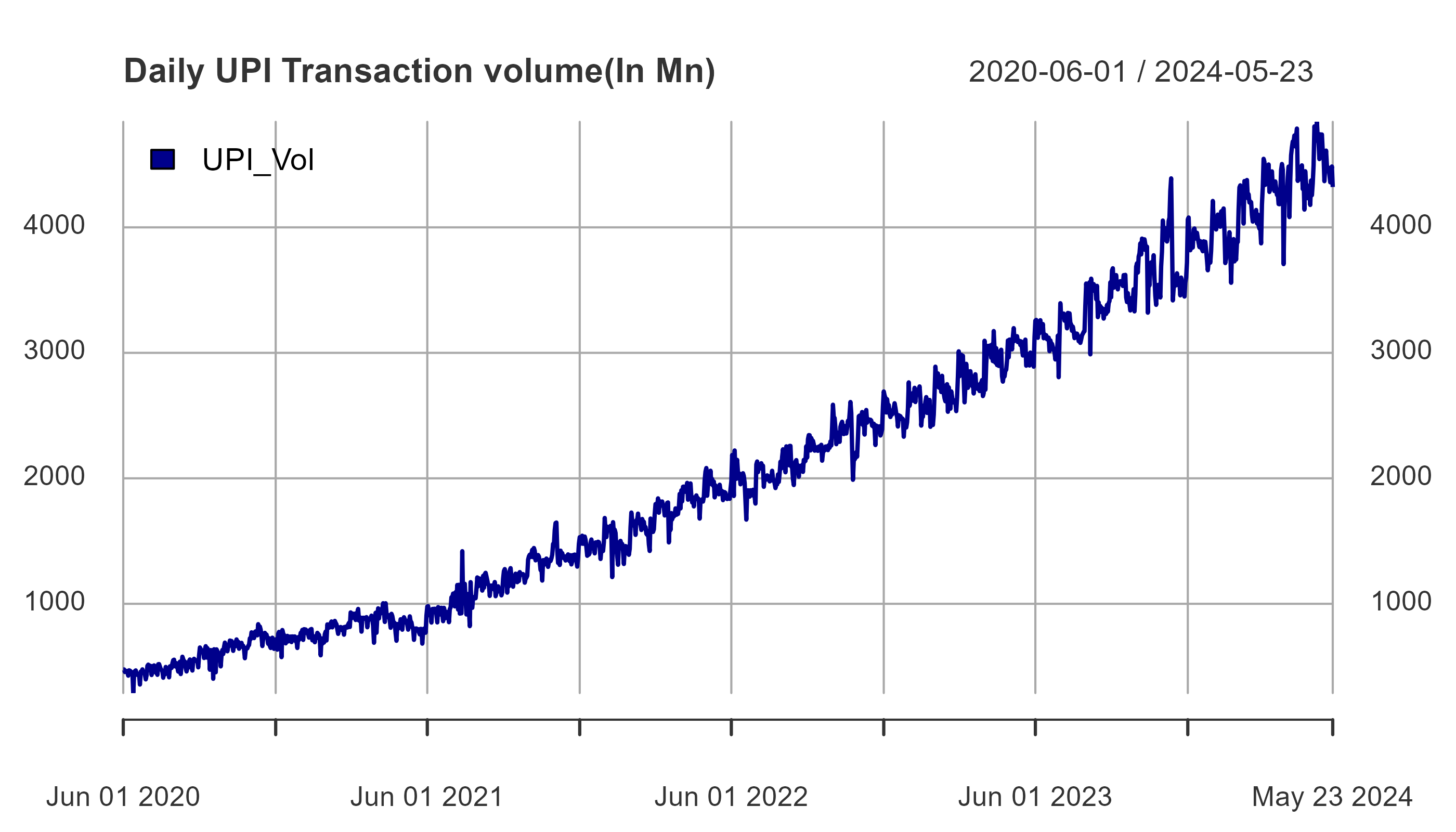
The first few rows of the dataset is –

Table 1: **RBI Daily Setttlement data for digital transactions**

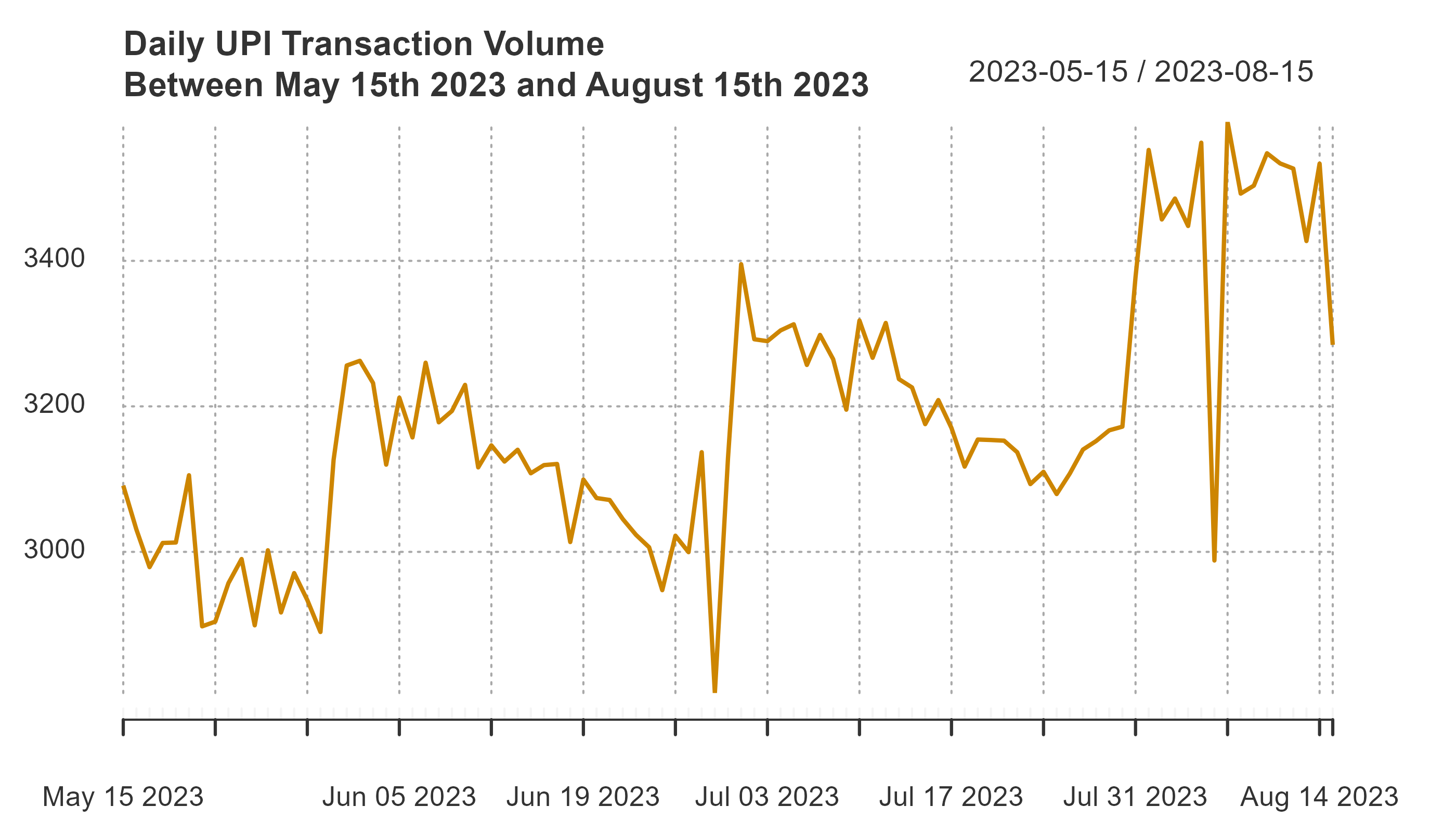
*2020 to 2024*

| Date | RTGS\_Vol | RTGS\_Val | NEFT\_Vol | NEFT\_Val | AePS\_Vol | AePS\_Val | UPI\_Vol | UPI\_Val | IMPS\_Vol | IMPS\_Val | NACH\_Vol | NACH\_Val | NACHDebit\_Vol | NACHDebit\_Val | NETC\_Vol | NETC\_Val | BBPS\_Vol | BBPS\_Val | CTS\_Vol | CTS\_Val |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2020-06-01 | 4.85 | 436,996.69 | 172.11 | 104,275.13 | 0.44 | 7.68 | 476.97 | 10,413.11 | 76.81 | 9,072.55 | 84.86 | 6,326.17 | 20.12 | 2,084.74 | 23.82 | 44.37 | 5.43 | 95.36 | 17.55 | 15,056.24 |
| 2020-06-02 | 4.54 | 361,878.87 | 100.07 | 65,259.02 | 0.44 | 7.67 | 476.78 | 9,951.30 | 72.25 | 8,215.41 | 143.33 | 14,387.49 | 63.13 | 3,613.36 | 24.38 | 46.23 | 5.21 | 81.15 | 18.25 | 16,041.19 |
| 2020-06-03 | 4.30 | 330,632.89 | 100.36 | 62,985.75 | 0.44 | 7.48 | 456.26 | 9,622.38 | 68.15 | 7,653.71 | 40.66 | 2,985.88 | 38.46 | 2,288.95 | 25.36 | 47.21 | 5.24 | 81.36 | 16.76 | 14,536.47 |
| 2020-06-04 | 4.35 | 329,072.45 | 94.66 | 63,148.29 | 0.45 | 7.32 | 463.05 | 9,639.50 | 70.69 | 7,705.89 | 286.04 | 5,113.36 | 37.80 | 3,592.44 | 25.81 | 48.39 | 5.51 | 83.05 | 17.39 | 14,865.92 |
| 2020-06-05 | 4.56 | 365,468.95 | 111.26 | 68,932.72 | 0.48 | 7.32 | 464.79 | 9,539.52 | 73.00 | 7,705.25 | 70.96 | 3,166.10 | 88.48 | 9,558.57 | 25.99 | 48.60 | 5.33 | 85.94 | 18.25 | 15,214.06 |
| 2020-06-06 | 3.79 | 111,618.22 | 77.05 | 60,118.06 | 0.54 | 7.71 | 458.65 | 9,119.20 | 70.35 | 7,076.03 | 55.08 | 3,528.03 | 20.46 | 2,336.25 | 26.74 | 49.48 | 5.71 | 88.65 | 17.56 | 13,210.67 |
| 2020-06-07 | 0.00 | 0.00 | 8.36 | 2,651.29 | 0.44 | 5.58 | 427.26 | 6,875.00 | 54.25 | 4,350.25 | 0.00 | 0.00 | 0.00 | 0.00 | 25.20 | 48.25 | 4.56 | 58.07 | 0.00 | 0.00 |
| 2020-06-08 | 5.33 | 343,217.53 | 121.32 | 68,564.97 | 0.62 | 8.21 | 469.99 | 10,062.17 | 71.30 | 8,143.19 | 97.13 | 3,183.71 | 76.62 | 5,137.65 | 26.14 | 46.35 | 6.14 | 102.27 | 20.45 | 14,905.54 |
| 2020-06-09 | 4.95 | 295,808.24 | 95.19 | 68,888.08 | 0.63 | 8.43 | 466.98 | 9,786.86 | 69.55 | 7,725.25 | 153.41 | 3,759.64 | 14.01 | 1,632.16 | 26.67 | 48.45 | 6.11 | 97.25 | 20.46 | 16,020.33 |
| 2020-06-10 | 4.79 | 332,717.29 | 87.89 | 63,676.19 | 0.69 | 9.11 | 461.58 | 9,839.41 | 69.48 | 7,650.20 | 60.69 | 9,100.73 | 80.53 | 8,699.34 | 28.48 | 51.27 | 6.09 | 100.79 | 21.40 | 16,443.04 |
| 2020-06-11 | 4.68 | 325,252.84 | 79.91 | 69,624.46 | 0.63 | 8.40 | 449.65 | 9,499.97 | 68.71 | 7,424.38 | 149.39 | 7,076.68 | 16.34 | 1,885.18 | 28.09 | 51.39 | 6.31 | 107.87 | 20.20 | 15,962.74 |
| 2020-06-12 | 5.23 | 431,435.81 | 78.75 | 72,097.78 | 0.64 | 8.98 | 453.43 | 9,288.43 | 68.24 | 7,365.12 | 50.51 | 2,389.14 | 21.36 | 2,555.54 | 29.64 | 53.13 | 6.32 | 99.41 | 20.50 | 16,135.23 |
| 2020-06-13 | 0.00 | 0.00 | 16.08 | 5,053.56 | 0.58 | 8.24 | 289.00 | 4,333.74 | 47.25 | 4,517.00 | 0.00 | 0.00 | 0.00 | 0.00 | 28.84 | 52.80 | 5.17 | 75.26 | 0.00 | 0.00 |
| 2020-06-14 | 0.00 | 0.00 | 10.83 | 2,674.32 | 0.39 | 5.53 | 435.87 | 7,238.23 | 57.41 | 4,388.98 | 0.00 | 0.00 | 0.00 | 0.00 | 26.58 | 50.28 | 5.40 | 73.03 | 0.00 | 0.00 |
| 2020-06-15 | 6.74 | 489,520.12 | 95.28 | 88,318.82 | 0.53 | 8.04 | 463.91 | 9,725.94 | 72.58 | 8,165.45 | 93.28 | 5,950.57 | 68.28 | 5,306.86 | 27.51 | 47.91 | 8.27 | 147.37 | 29.00 | 21,170.32 |
| *Source*:Reserve Bank of India(RBI) | | | | | | | | | | | | | | | | | | | | |

We now plot the daily UPI transaction volume



We can see that there is some kind of repeating pattern within the data , this is new to us since the monthly data we worked with earlier did not show any kind of seasonality or cyclic behavior,From further inspection we can see this is due to the fact that the patterns appear within the monthly period, We can zoom to a specific portion of the graph to clearly analyze the pattern.

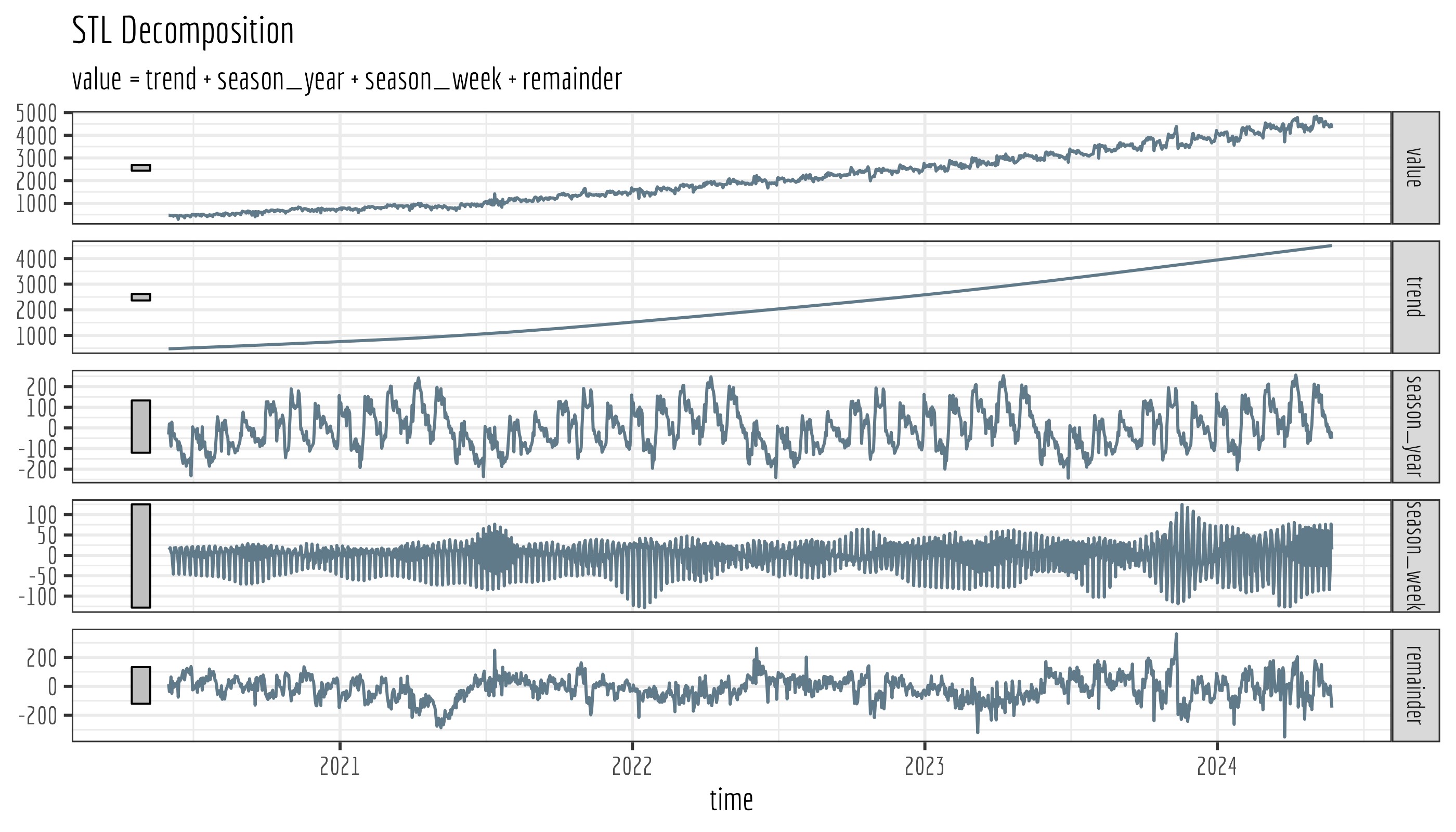


We can see from the plot that at the beginning of each month the transaction volume achieves its maximum value,this value gradually decreases over the month and another peak is acheieved at the start of the next month.

### Decomposition

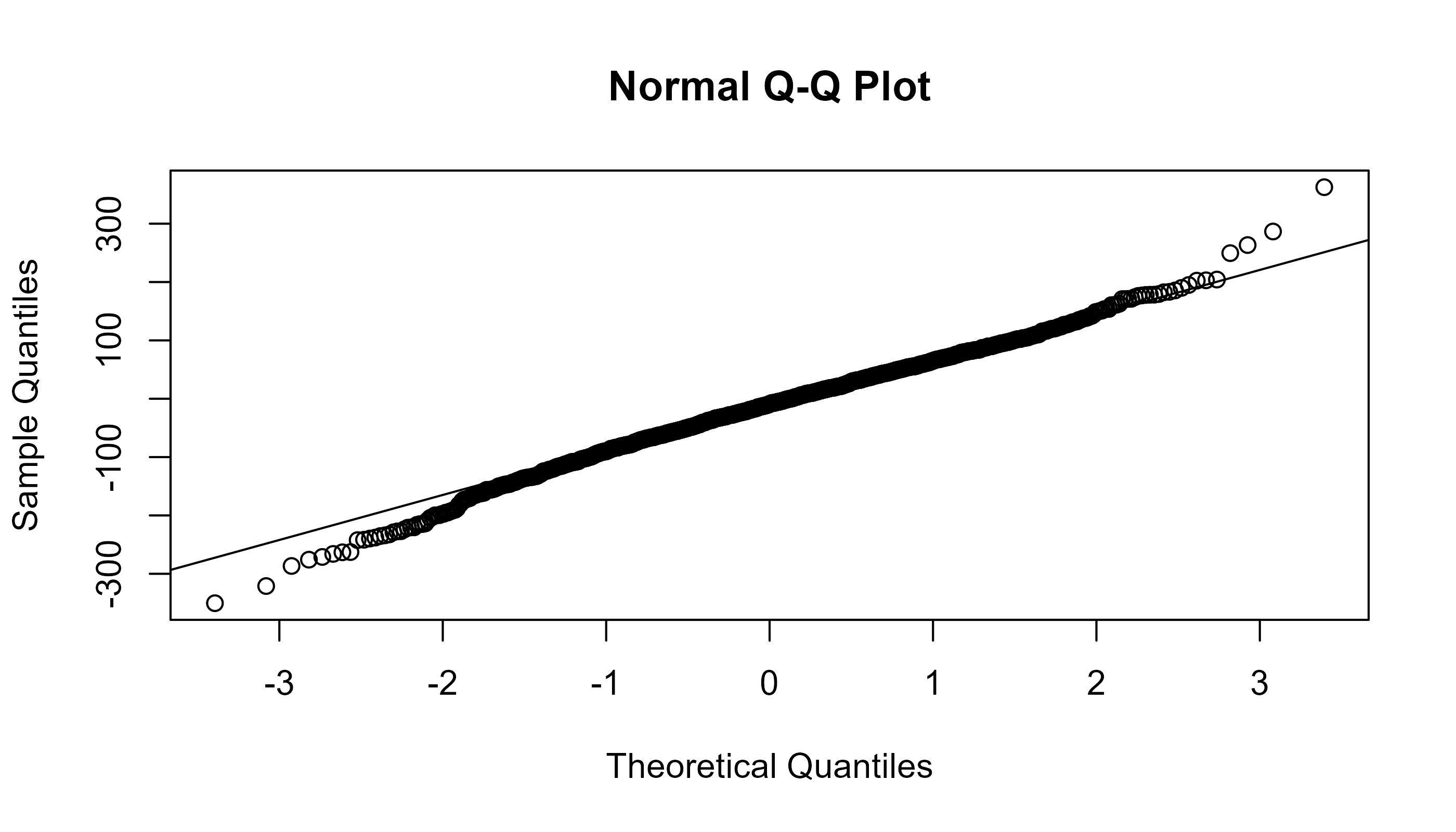
The main problem with decomposing a daily time series is that monthly seasonal patterns are hard to catch since their period of occurrence although is technically seasonal but is irregular patterns since all months don’t have the same number of days. So we may perform a decomposition based on an assumed model of -

where & are monthly and yearly seasonality respectively.  
Since our data is daily so classical decomposition fails since classical decomposition is unable to catch seasonality within the months.To overcome this issue we use the **STL** decomposition method, here STL stands for “Seasonal and Trend decomposition using LOESS(locally estimated scatterplot smoothing)” ,This method was developed by R. B. Cleveland et al. ([Cleveland et al. 1990](#ref-cleveland90)).



##   
## Box-Pierce test  
##   
## data: .  
## X-squared = 703.04, df = 1, p-value < 2.2e-16

We can see from the decomposition plot that the seasonal component is increasing with time,the trend component is fairly smooth and shows a upward growth as we saw in our monthly data.Thely seasonality shows an increasing trend towards the end of the year this can be attributed to increase in festivities during the later part of the Year . The remainders are fairly consistent as can be seen in the qqplot below-



This implies that our selection of the decomposition model is decent.

A sample of The decomposed data is given as -

Table 1: Decomposition of Daily UPI Volume of Transactions

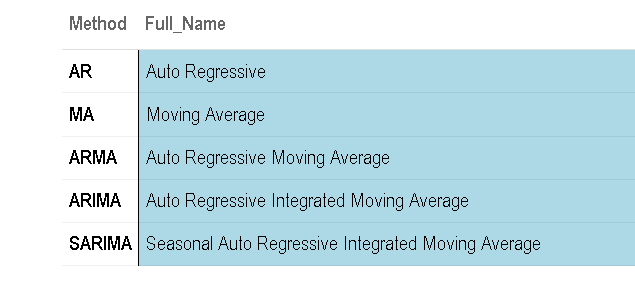
*2021 to 2024*

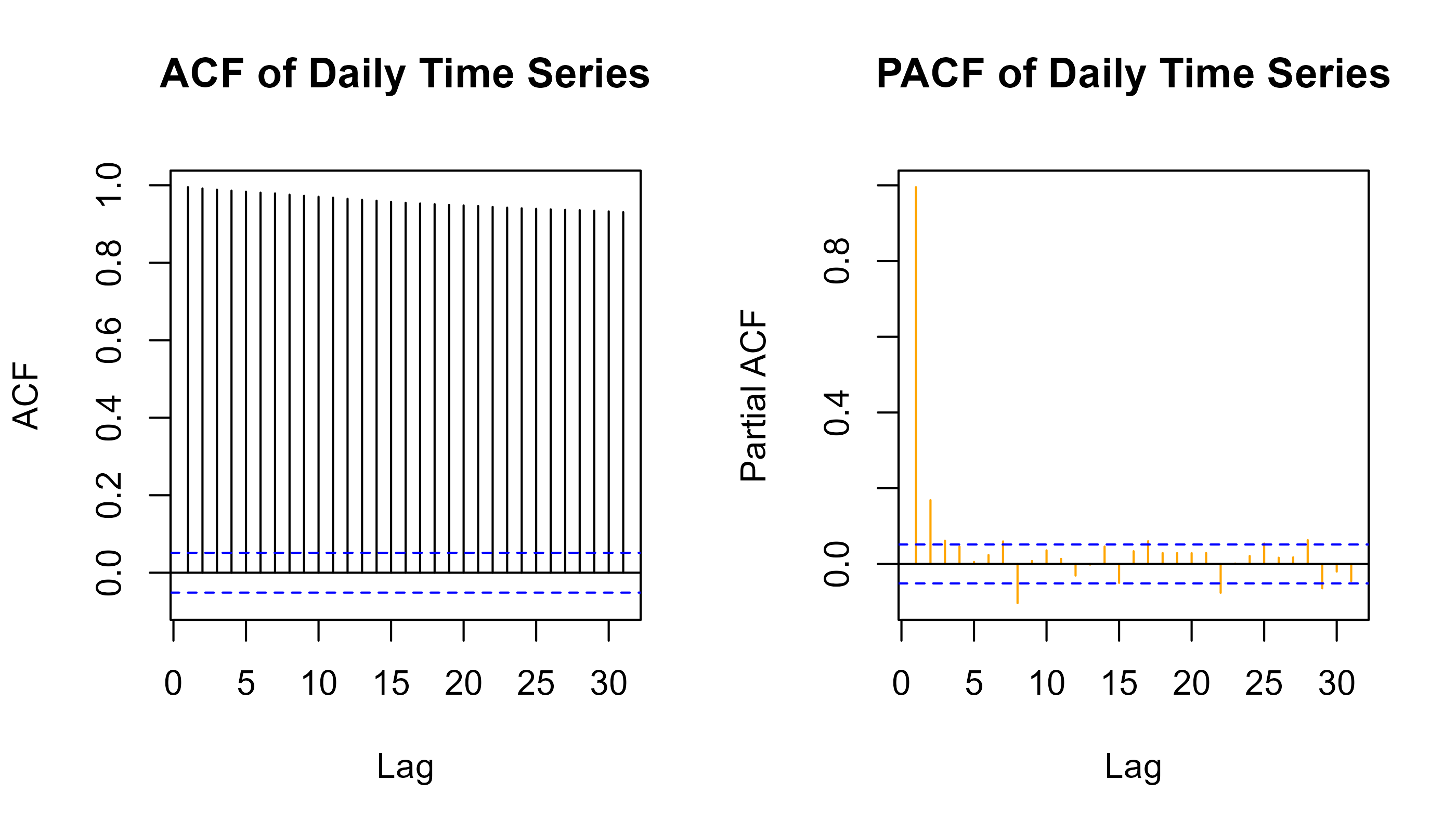
| time | value | trend | season\_week | season\_year | remainder | season\_adjust |
| --- | --- | --- | --- | --- | --- | --- |
| 2022-05-31 | 1912.11 | 1939.688 | -26.2468024 | -24.415914 | 23.0842254 | 1962.773 |
| 2022-06-01 | 1976.68 | 1942.539 | 36.5701591 | -26.491211 | 24.0620075 | 1966.601 |
| 2022-06-02 | 2186.08 | 1945.390 | 33.5294326 | 24.648644 | 182.5123243 | 2127.902 |
| 2022-06-03 | 2050.04 | 1948.240 | 18.8725766 | -36.826319 | 119.7535889 | 2067.994 |
| 2022-06-04 | 1859.27 | 1951.147 | -24.9705283 | -86.088044 | 19.1811365 | 1970.329 |
| 2022-06-05 | 2222.02 | 1954.055 | -29.2858384 | 33.871308 | 263.3798116 | 2217.435 |
| 2022-06-06 | 2020.38 | 1956.962 | -11.2205166 | -32.924176 | 107.5626904 | 2064.525 |
| 2022-06-07 | 1995.90 | 1959.869 | -23.0213909 | -21.720013 | 80.7721193 | 2040.641 |
| 2022-06-08 | 2146.97 | 1962.777 | 36.1509178 | -23.087852 | 171.1303665 | 2133.907 |
| 2022-06-09 | 2042.38 | 1965.684 | 25.1003580 | -50.720591 | 102.3163820 | 2068.000 |
| 2022-06-10 | 2021.75 | 1968.591 | 24.4124308 | -50.903978 | 79.6504136 | 2048.242 |
| 2022-06-11 | 2037.13 | 1971.498 | -18.2753634 | -54.085954 | 137.9929013 | 2109.491 |
| 2022-06-12 | 1952.53 | 1974.406 | -34.7221049 | -71.261036 | 84.1074418 | 2058.513 |
| 2022-06-13 | 2006.68 | 1977.313 | -8.2833114 | -101.681508 | 139.3318368 | 2116.645 |
| 2022-06-14 | 1986.69 | 1980.220 | -20.9357239 | -67.382914 | 94.7883726 | 2075.009 |
| 2022-06-15 | 2040.50 | 1983.128 | 34.6333467 | -81.048142 | 103.7872468 | 2086.915 |
| 2022-06-16 | 2017.36 | 1986.035 | 15.7555931 | -88.596439 | 104.1660151 | 2090.201 |
| 2022-06-17 | 1962.92 | 1988.942 | 29.3483463 | -124.231045 | 68.8605843 | 2057.803 |
| 2022-06-18 | 1818.97 | 1991.849 | -11.8248605 | -153.986270 | -7.0682664 | 1984.781 |
| 2022-06-19 | 1670.82 | 1994.757 | -40.0856004 | -173.495529 | -110.3555509 | 1884.401 |
| 2022-06-20 | 1878.08 | 1997.664 | -3.6487903 | -128.176388 | 12.2412148 | 2009.905 |
| 2022-06-21 | 1904.54 | 2000.571 | -14.3943628 | -143.762310 | 62.1254272 | 2062.697 |
| 2022-06-22 | 1885.50 | 2003.479 | 26.4021110 | -159.030248 | 14.6496077 | 2018.128 |
| 2022-06-23 | 1854.26 | 2006.386 | 5.7329692 | -188.754384 | 30.8956032 | 2037.281 |
| 2022-06-24 | 1904.89 | 2009.293 | 31.7814289 | -176.453232 | 40.2687081 | 2049.562 |
| 2022-06-25 | 1898.02 | 2012.200 | -1.5811876 | -174.078927 | 61.4797372 | 2073.680 |
| 2022-06-26 | 1865.08 | 2015.108 | -46.9054901 | -158.094703 | 54.9725323 | 2070.080 |
| 2022-06-27 | 1908.78 | 2018.015 | 1.5381752 | -161.438950 | 50.6658311 | 2068.681 |
| 2022-06-28 | 1865.29 | 2020.922 | -7.3424042 | -147.372250 | -0.9175727 | 2020.005 |
| 2022-06-29 | 1850.76 | 2023.830 | 18.6599637 | -240.757595 | 49.0281216 | 2072.858 |
| 2022-06-30 | 1798.81 | 2026.737 | -3.7968279 | -133.994350 | -90.1356143 | 1936.601 |
| 2022-07-01 | 2110.38 | 2029.644 | 34.7330293 | 9.093643 | 36.9092529 | 2066.553 |
| 2022-07-02 | 2134.15 | 2032.551 | 9.2075488 | -4.404395 | 96.7954876 | 2129.347 |
| 2022-07-03 | 2049.51 | 2035.459 | -53.2090926 | -16.304091 | 83.5645429 | 2119.023 |
| 2022-07-04 | 2093.17 | 2038.366 | 0.7955550 | -7.728076 | 61.7365973 | 2100.103 |
| 2022-07-05 | 2077.94 | 2041.273 | -3.4892424 | -21.980845 | 62.1368807 | 2103.410 |
| 2022-07-06 | 2093.44 | 2044.180 | 14.6704903 | -51.032131 | 85.6211505 | 2129.802 |
| 2022-07-07 | 2120.26 | 2047.088 | -5.6499260 | -1.577617 | 80.3997697 | 2127.488 |
| 2022-07-08 | 2079.58 | 2049.995 | 26.7446627 | -70.369306 | 73.2095873 | 2123.205 |
| 2022-07-09 | 2100.71 | 2052.902 | 20.3841137 | -34.401726 | 61.8252740 | 2114.728 |
| 2022-07-10 | 1932.44 | 2055.810 | -49.2331872 | -102.169433 | 28.0329987 | 2083.843 |
| 2022-07-11 | 2000.96 | 2058.717 | -1.0069801 | -70.688980 | 13.9390553 | 2072.656 |
| 2022-07-12 | 2011.21 | 2061.624 | -0.5292983 | -96.359251 | 46.4743619 | 2108.099 |
| 2022-07-13 | 1989.50 | 2064.531 | 9.9919384 | 2.434613 | -87.4580214 | 1977.073 |
| 2022-07-14 | 2023.58 | 2067.439 | -8.0481934 | -89.668759 | 53.8581992 | 2121.297 |
| 2022-07-15 | 2019.00 | 2070.346 | 18.2785260 | -94.849472 | 25.2249100 | 2095.571 |
| 2022-07-16 | 2013.56 | 2073.253 | 31.1280097 | -67.936872 | -22.8844567 | 2050.369 |
| 2022-07-17 | 1975.48 | 2076.161 | -45.6360634 | -115.945047 | 60.9005085 | 2137.061 |
| 2022-07-18 | 1980.21 | 2079.068 | -8.1712910 | -145.522092 | 54.8354983 | 2133.903 |
| 2022-07-19 | 1997.15 | 2081.975 | -0.5591309 | -129.684669 | 45.4186315 | 2127.394 |
| 2022-07-20 | 2060.07 | 2084.882 | 13.2363703 | -118.062021 | 80.0132001 | 2164.896 |
| 2022-07-21 | 1995.59 | 2087.790 | 6.5962059 | -166.708375 | 67.9124350 | 2155.702 |
| 2022-07-22 | 1984.87 | 2090.697 | 4.1890264 | -189.146528 | 79.1304844 | 2169.828 |
| 2022-07-23 | 1961.15 | 2093.604 | 33.0966137 | -121.557296 | -43.9936177 | 2049.611 |
| 2022-07-24 | 1922.39 | 2096.512 | -49.7730018 | -174.851530 | 50.5029491 | 2147.015 |
| 2022-07-25 | 1963.87 | 2099.419 | -14.1354722 | -162.273337 | 40.8599438 | 2140.279 |
| 2022-07-26 | 1951.14 | 2102.326 | 0.5231314 | -167.556430 | 15.8471495 | 2118.173 |
| 2022-07-27 | 2030.82 | 2105.233 | 17.4067391 | -146.183957 | 54.3637864 | 2159.597 |
| 2022-07-28 | 1971.07 | 2108.141 | 21.9039079 | -151.509431 | -7.4651918 | 2100.676 |
| 2022-07-29 | 2039.83 | 2111.048 | -9.5496615 | -127.452803 | 65.7844671 | 2176.832 |
| 2022-07-30 | 2132.19 | 2113.955 | 35.1072186 | -70.983128 | 54.1106293 | 2168.066 |
| 2022-07-31 | 2078.18 | 2116.863 | -54.1086064 | -10.380189 | 25.8062322 | 2142.669 |
| 2022-08-01 | 2171.27 | 2119.770 | -16.6005056 | 58.809982 | 9.2906774 | 2129.061 |
| *Source*:Reserve Bank of India(RBI) | | | | | | |

## Stochastic Modelling

So far we have rarely used any stochastic models to analyze our dataset we now move to a more sophisticated analysis and forecast using Stochastic Models like **AR** ,**MA**,**ARMA**,**ARIMA** &**SARIMA** .

#### **Acronyms**

  
We first plot the Auto Correlation function and the Partial Autocorrelation Function to see if we can identify the process.



#### Interpretation of ACF & PACF

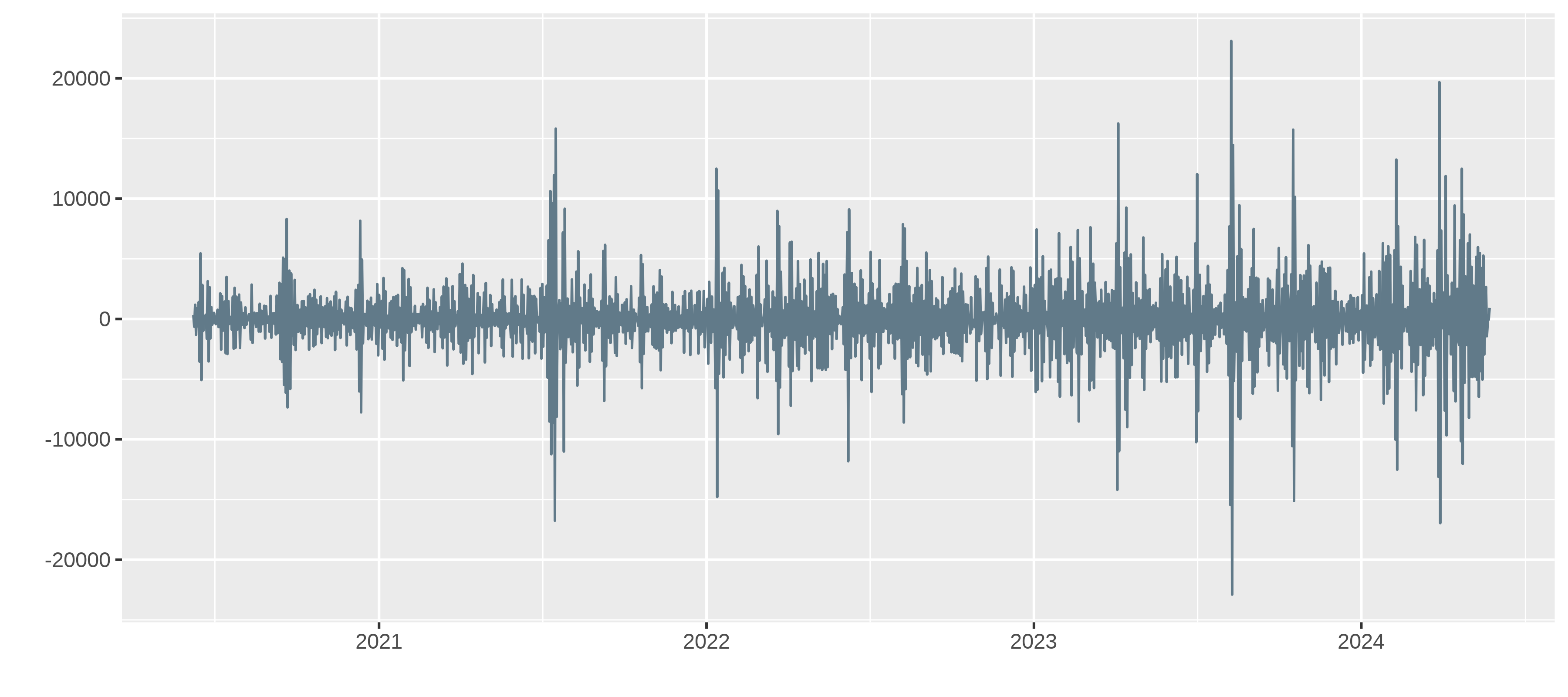
1. **ACF**- The ACF shows significant autocorrelations at all lags, slowly decreasing. This pattern is characteristic of a non-stationary series, typically one that might be differenced to achieve stationarity.
2. **PACF** - The PACF has a significant spike at lag 1 and then cuts off quickly, with the subsequent lags being mostly within the confidence intervals. This suggests that the time series might follow an autoregressive process of order 1 - .  
   Using the auto.arima() function from the forecast package we can find the optimal ARIMA model based on the lowest AIC values.This function is based on the Hyndman-Khandakar algorithm.([Hyndman and Khandakar 2008](#ref-JSSv027i03))

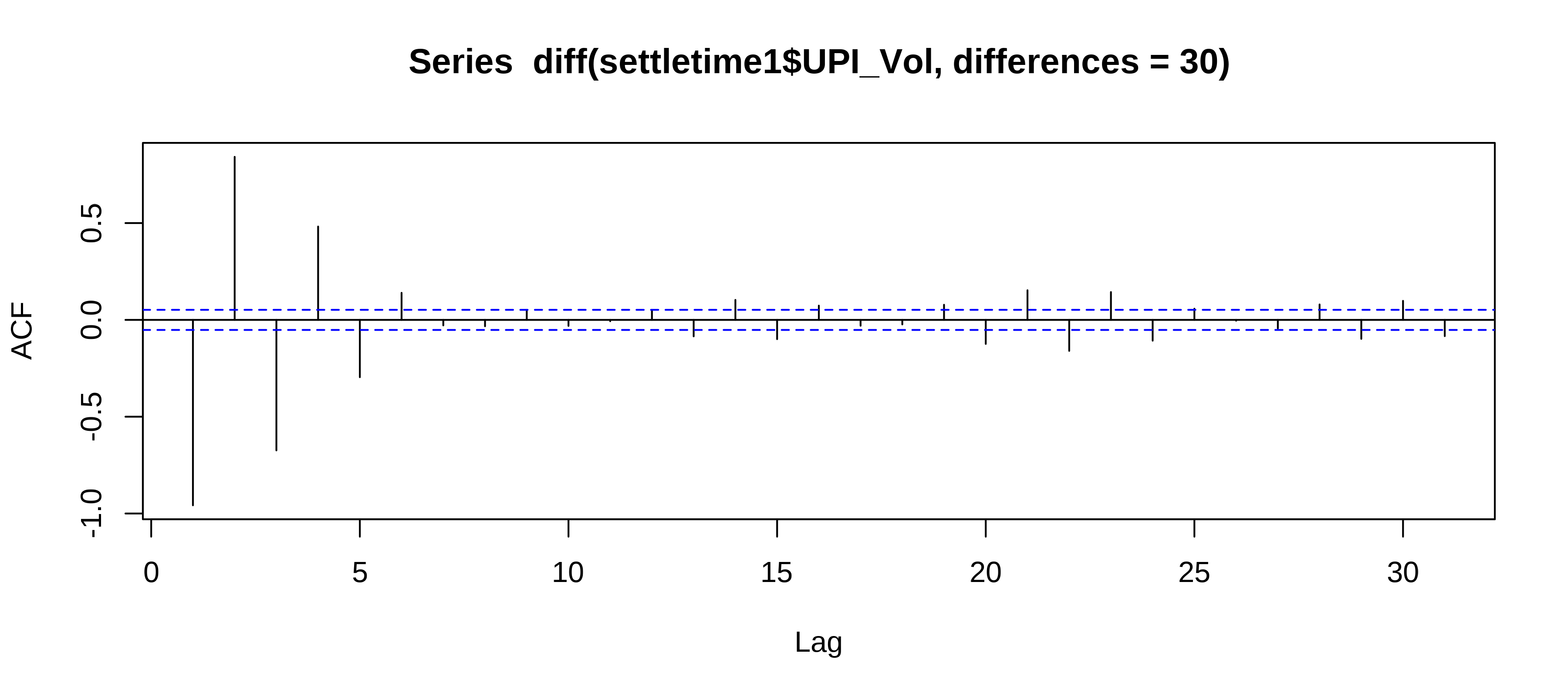
### ARIMA Modeling -

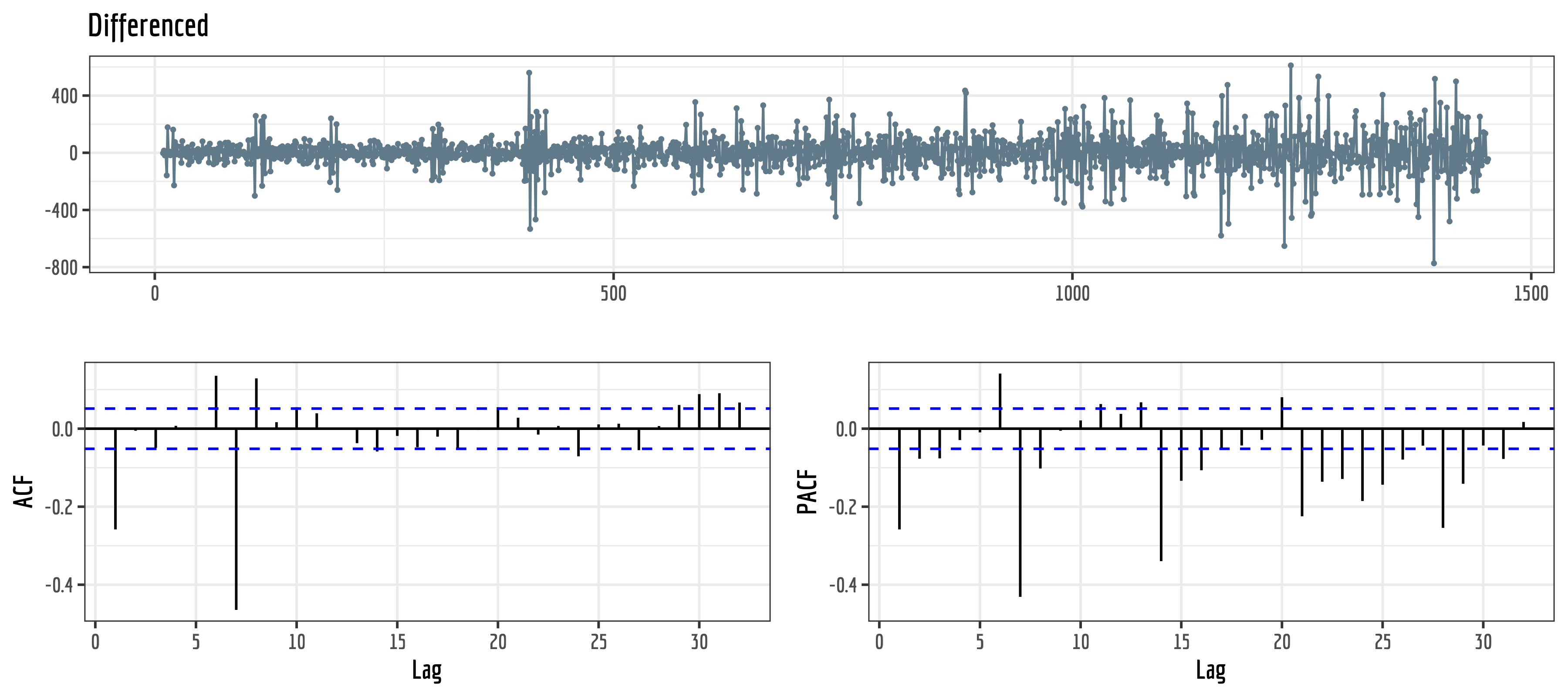
We first fit the data to the best possible ARIMA model using the auto.arima() function The Summary of the fitted model is given as -

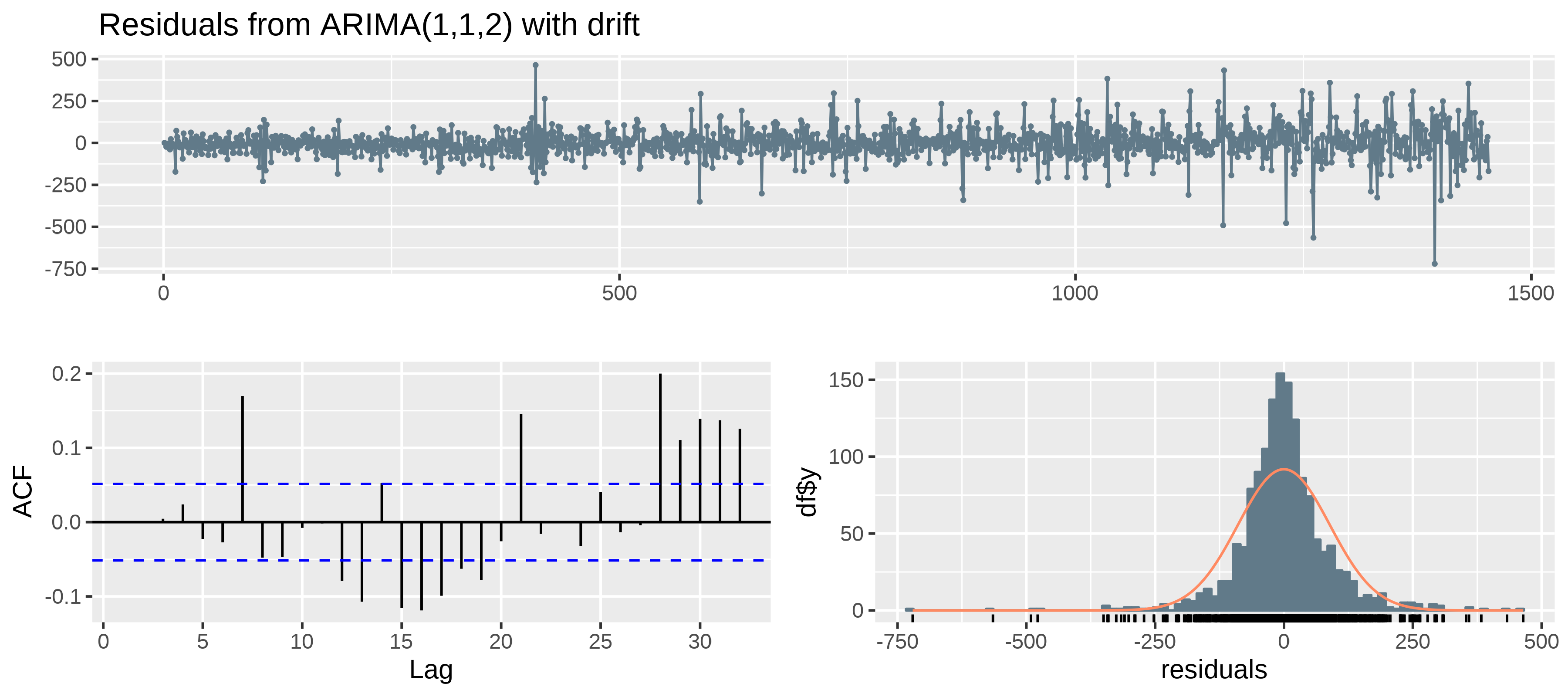
## Series: UPI Volume of Daily Transactions   
## ARIMA(1,1,2) with drift   
##   
## Coefficients:  
## ar1 ma1 ma2 drift  
## 0.7930 -1.2040 0.2273 2.8383  
## s.e. 0.0267 0.0394 0.0369 0.2677  
##   
## sigma^2 = 7953: log likelihood = -8579.26  
## AIC=17168.52 AICc=17168.56 BIC=17194.92  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.2706182 89.02712 59.52952 -0.653771 3.613357 0.9234873  
## ACF1  
## Training set -0.0004667928

We can see that our chosen model is ARIMA(1,1,2)

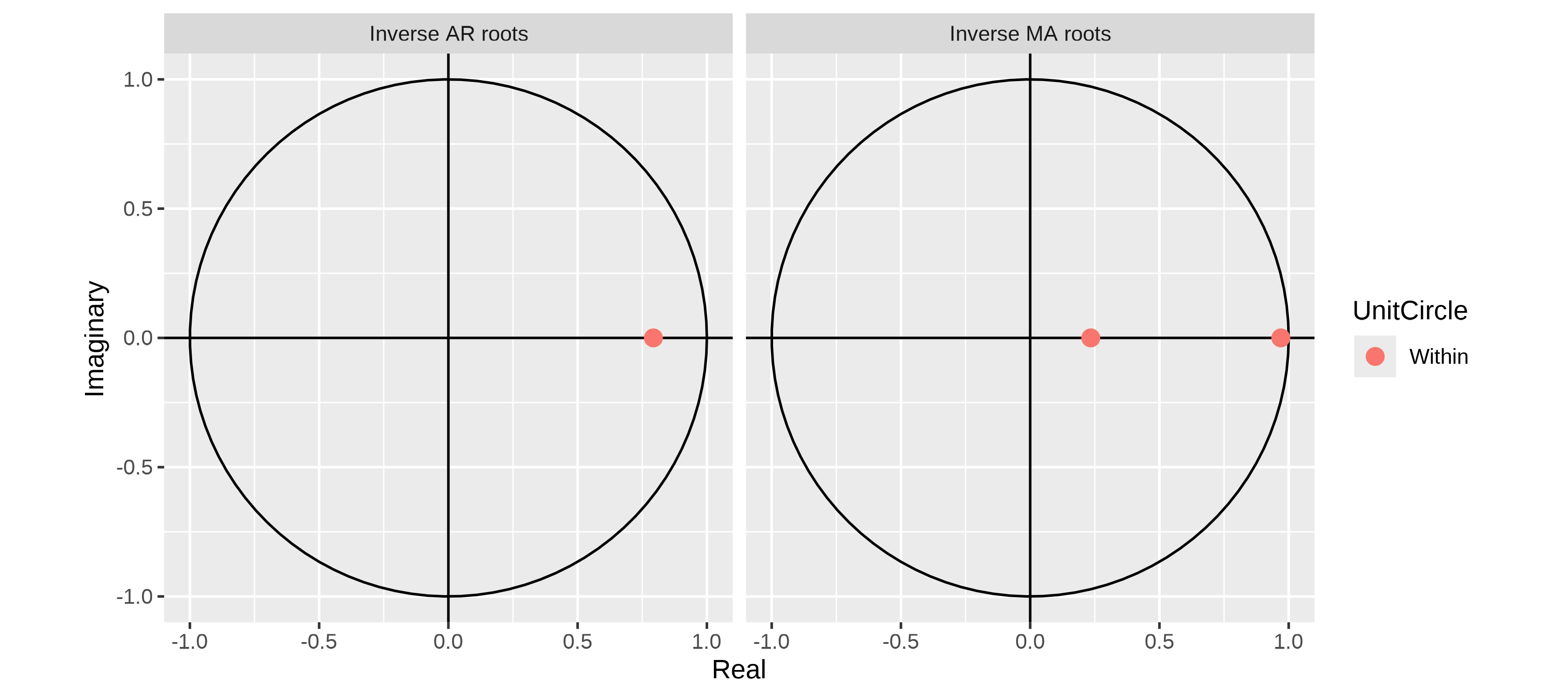


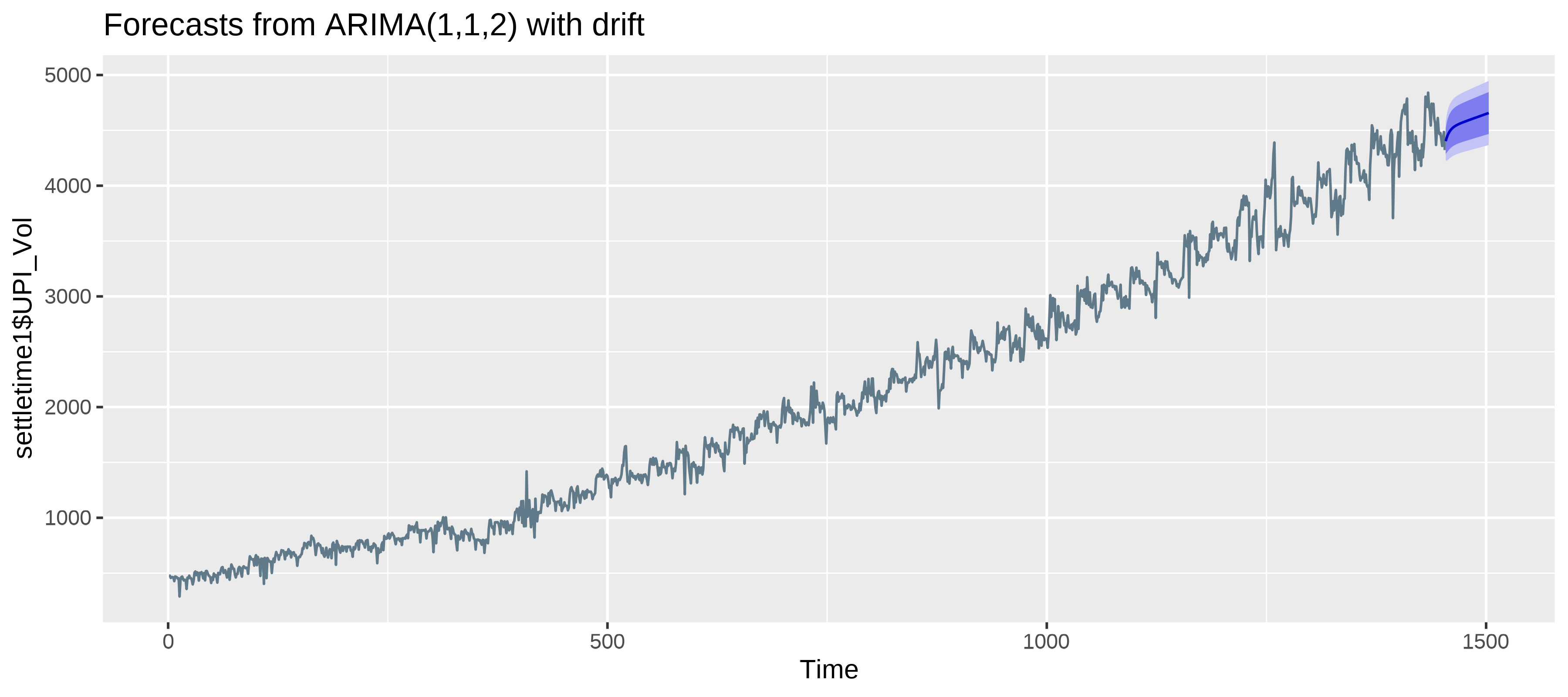






##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,2) with drift  
## Q\* = 51.451, df = 7, p-value = 7.489e-09  
##   
## Model df: 3. Total lags used: 10





# References

Chatfield, C. 2016. *The Analysis of Time Series: An Introduction, Sixth Edition*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press. <https://books.google.co.in/books?id=qKzyAbdaDFAC>.

Cleveland, Robert B., William S. Cleveland, Jean E. McRae, and Irma Terpenning. 1990. “STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion).” *Journal of Official Statistics* 6: 3–73.

Hyndman, Rob J., and Yeasmin Khandakar. 2008. “Automatic Time Series Forecasting: The Forecast Package for r.” *Journal of Statistical Software* 27 (3): 1–22. <https://doi.org/10.18637/jss.v027.i03>.

Kendall, M. G., and A. Stuart. 1966. *The Advanced Theory of Statistics: Vol. 3, Design and Analysis, and Time-Series*. v. 3. Griffin. <https://books.google.co.in/books?id=lvlgzwEACAAJ>.

Rogers, E. M. 2003. *Diffusion of Innovations, 5th Edition*. Free Press. <https://books.google.co.in/books?id=9U1K5LjUOwEC>.