Pharma sales data analysis and forecasting

Simeen Pawaskar-31031820002

Aishwarya Borkar-31031820007

Neha Chavhan-31031820008

Sanchita Dabre-31031820012

Madhura Jejurkar-31031820049

Karthik Murthy-31031820050

S K Somaiya College (SKSC)

DEPARTMENT OF STATISTICS

131P18E301: Time Series Analysis

MSc GROUP ASSIGNMENT

(2021 – 2022)

https://www.kaggle.com/milanzdravkovic/pharma-sales-data-analysis-and-forecasting

Roll Number	Name	Email ID
31031820002	Simeen Pawaskar	simeen.a@somaiya.edu
31031820007	Aishwarya Borkar	aishwarya.mb@somaiya.edu
31031820008	Neha Chavhan	n.chavhan@somaiya.edu
31031820012	Sanchita Dabre	sanchita.d@somaiya.edu
31031820049	Madhura Jejurkar	madhura.jejurkar@somaiya.edu
31031820050	Karthik Murthy	k.murthy@somaiya.edu

Prof. Mayur More
Professor Incharge

WHY WE SELECTED THIS TOPIC?

Before the finish of task fulfillment, we comprehended that the following most huge part of determining is to record input situations and suppositions considered by partners in driving the figure, as these are utilized to direct asset assignment choices. The forecaster may not be fit for social affair all regions affecting the brand where other utilitarian groups become an integral factor. If it's not too much trouble, note that this is a continuous interaction, changing estimates time-to-time dependent on changing asset allotment.

For a bigger scope, the sales forecasting in drug industry is normally finished by utilizing Naive model, where the forecasted values equivalent qualities in the past period with added variable of development, which is explicitly characterized for various locales, markets, classifications of items, and so forth Although this model fails when the market soaks, overall and for a bigger scope, it has demonstrated as effective.

INTRODUCTION

All things considered, investigation and conjectures on a more limited size, like single wholesaler, drug store chain or even individual drug store, more modest periods like weeks, and so forth, guide vital choices identified with asset and acquirement arranging, imagine a scenario where examinations, profit from venture anticipating, business arranging and others. The principle issue in more limited size time series examinations and gauges are huge vulnerabilities and deals execution exceptionally near irregular, making the forecasts with accuracies above edges as characterized by naive strategies hard to accomplish.

The principle research question we tackle is identified with investigating the attainability of utilization of present day time-series forecasting strategies in drug items deals forecasting on a more limited size. In specific, we benchmark the accuracies accomplished with those techniques against the exhibitions of fundamental Naive, seasonal Naïve and average methods.

Research work behind the paper thinks about 8 time series with various factual elements. Every one of the time-series sums up deals of a gathering of pharmaceutical items. Time-series information are gathered from the point of sale system of a single pharmacy in time of 6 years.

This paper is organized into 4 principle parts. To start with, short hypothetical analysis for time series examination and determining is given to advise the reader on the believability regarding choices made in the execution of this contextual analysis. Then, at that point, research strategy, really an issue impartial time series forecasting pipeline is introduced. Then, the real execution is introduced, by featuring the means made in after the proposed procedure on account of drug items deals information investigation and determining. At long last, the conversation brings the portrayal of genuine outcomes and a few ideas to the outreach group, driven by the aftereffect of the information examination.

OBJECTIVE:

The objective of the research behind the paper was to validate different methods and approaches related to sales time series data preparation, analysis and forecasting, with aim to facilitate recommending sales and marketing strategies based on trend/seasonality effects and forecasting sales of eight different groups of pharmaceutical products with diverse characteristics, such as stationarity, seasonality, amount of residuals and sales data variance.

METHODOLOGY:

DATA PREPARATION AND FEATURE ENGINEERING:

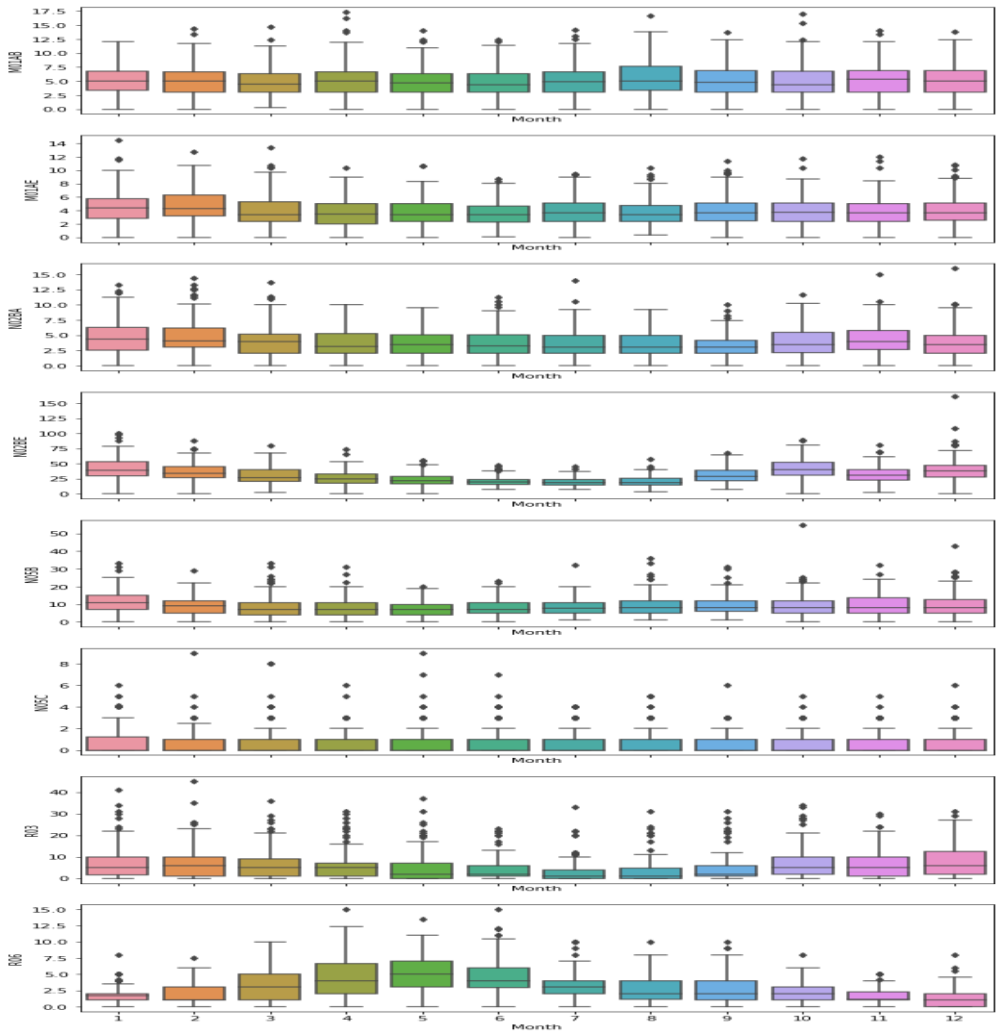
The dataset carries with it 6,00,000 information collected over the period of 6 years ;from 2014 to 2019.it includes date , time, sold batch, pharmaceutical drug name .According to the data collected from pharmacists ,it was decided that topic for analyses and forecasting is going to be drug classes and not any specific medication.therefore,57 drugs were classified into 8 Anatomical Therapeutic Chemical (ATC) Classification System categories:

- M01AB - Anti-inflammatory and antirheumatic products, non-steroids, Acetic acid derivatives and related substances
- M01AE - Anti-inflammatory and antirheumatic products, non-steroids, Propionic acid derivatives
- N02BA - Other analgesics and antipyretics, Salicylic acid and derivatives
- N02BE/B - Other analgesics and antipyretics, Pyrazolones and Anilides
- N05B - Psycholeptics drugs, Anxiolytic drugs
- N05C - Psycholeptics drugs, Hypnotics and sedatives drugs
- R03 - Drugs for obstructive airway diseases
- R06 - Antihistamines for systemic use

SEASONALITY ANALYSIS:

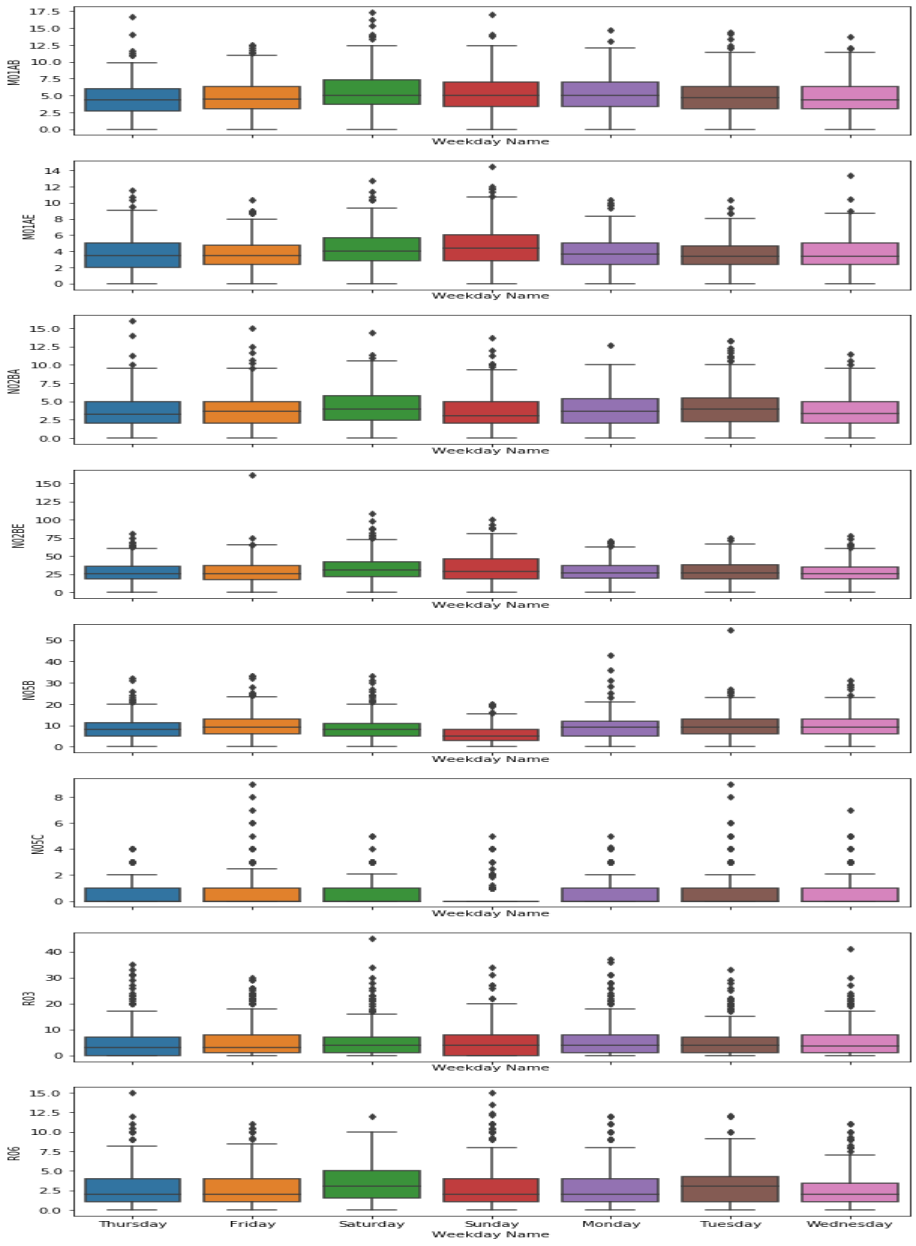
Seasonality is the cycle that keeps on repeating over a period of time. The seasonality analysis can be examined using box plot and we perform this analysis in python.

```
import seaborn as sns
dfatc_daily = pd.read_csv('../input/salesdaily.csv')
fig, axes = plt.subplots(8, 1, figsize=(10, 30), sharex=True)
for name, ax in zip(['M01AB','M01AE','N02BA','N02BE', 'N05B','N05C','R03','R06'], axes):
    sns.boxplot(data=dfatc_daily, x='Month', y=name, ax=ax)
```



As we can see from the above vizualisation ,R03 and N05C has more outliers compare to the others, which gives information that their sales are more challenging to predict.

Now we check seasonality for **weekdays**:



Rolling window means is another visualization that is useful for determining seasonality patterns. Rolling window divide the data into time windows and the input in each window is aggregated into mean,median and sum. rolling windows overlap and "roll" along at the same frequency as the data, so the transformed time series is at the same frequency as the original time series. That indicates that the curve is smoother.

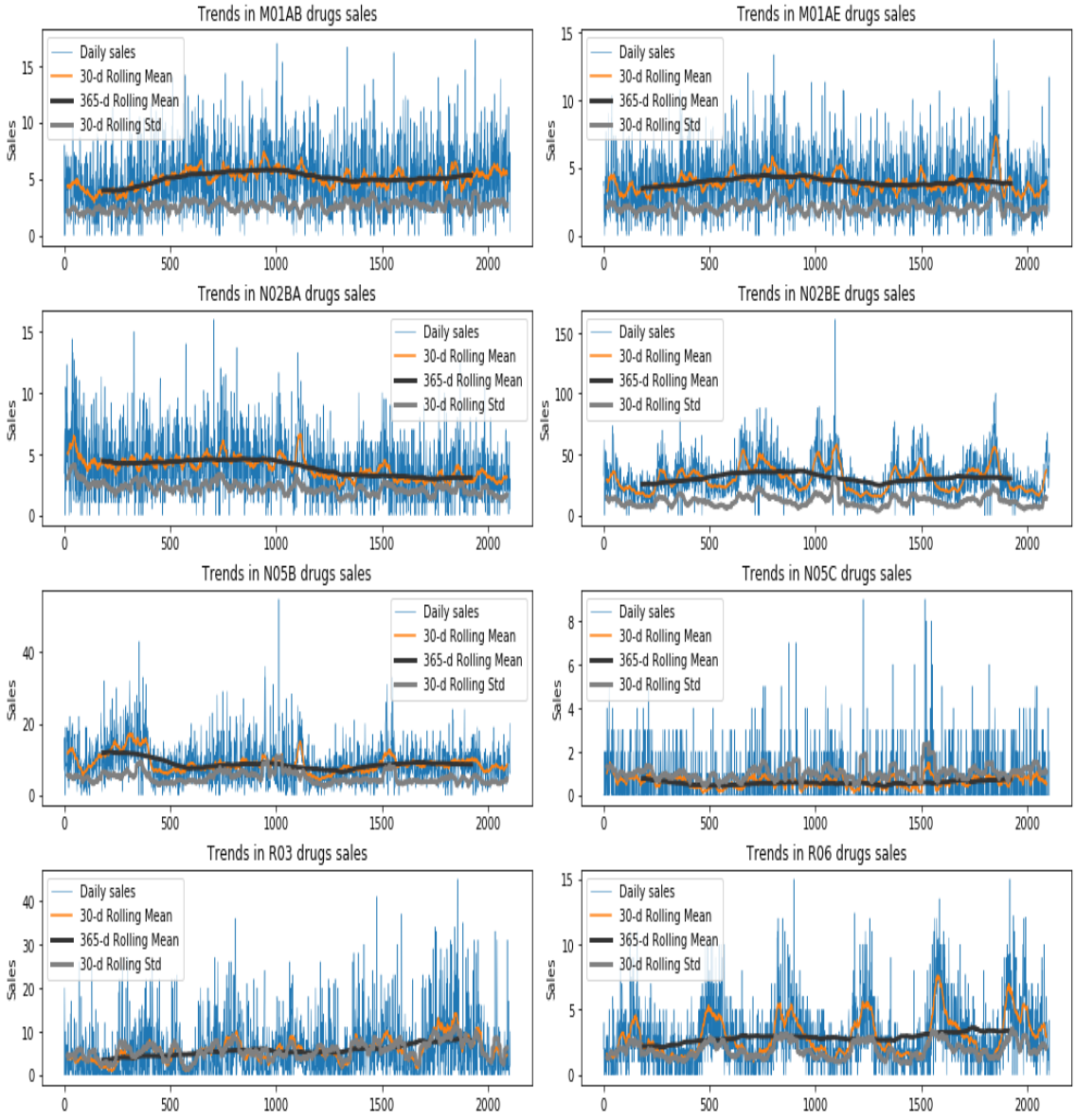
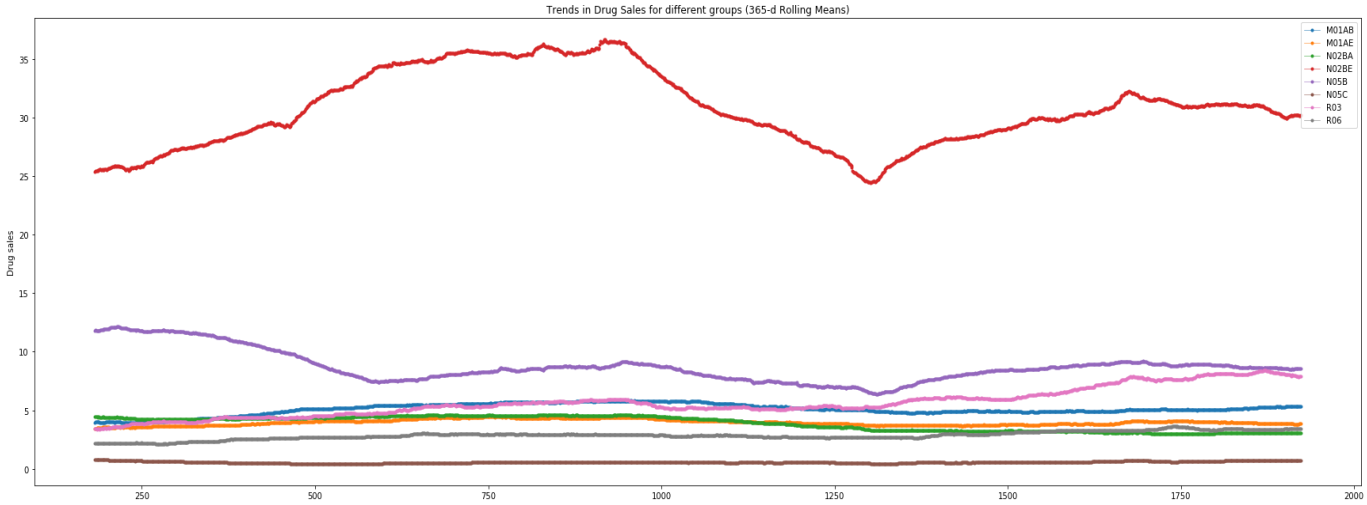


Image below shows trends for each of the drug categories, represented by the 365-d rolling means for each of those categories.



STATIONARITY ANALYSIS:

Stationarity of time-series is the property of displaying constant measurable properties over the long run (for instance, mean, variance, autocorrelation). It very well may be outwardly controlled by plotting rolling statistics (rolling means and fluctuations). In stationary time series, the mean of the series, variance of the series and covariance of the i th term and the (i + m) th term ought not be a function of time.

We can utilize Augmented Dickey-Fuller (ADF) test to actually look at stationarity of the information. Possible values of regression parameters of ADF are::

- c : constant only (default)
- ct : constant and trend
- ctt : constant, and linear and quadratic trend
- nc : no constant, no trend

```
df=pd.read_csv('../input/salesweekly.csv')
from statsmodels.tsa.stattools import adfuller
```

```
for x in ['M01AB','M01AE','N02BA','N02BE', 'N05B','N05C','R03','R06']:
    dftest = adfuller(df[x], regression='ct', autolag='AIC')
    print("ADF test for "+x)
    print("-----------------------------")
    print("Test statistic = {:.3f}".format(dftest[0]))
    print("P-value = {:.3f}".format(dftest[1]))
    print("Critical values :")
    for k, v in dftest[4].items():
        print("\t{}: {} - The data is {} stationary with {}% confidence".format(k
, v, "not" if v<dftest[0] else "", 100-int(k[:-1])))
```

Output:

ADF test for M01AB

Test statistic = -3.700

P-value = 0.022

Critical values :

1%: -3.9897903649837616 - The data is not stationary with 99% confidence

5%: -3.425478311521626 - The data is stationary with 95% confidence

10%: -3.1358607194990724 - The data is stationary with 90% confidence

ADF test for M01AE

Test statistic = -8.717

P-value = 0.000

Critical values :

1%: -3.98926783537037 - The data is stationary with 99% confidence

5%: -3.425226747185185 - The data is stationary with 95% confidence

10%: -3.1357131066666666 - The data is stationary with 90% confidence

ADF test for N02BA

Test statistic = -2.670

P-value = 0.249

Critical values :

1%: -3.990441532957606 - The data is not stationary with 99% confidence

5%: -3.425791763669738 - The data is not stationary with 95% confidence

10%: -3.1360446341572343 - The data is not stationary with 90% confidence

ADF test for N02BE

Test statistic = -4.362

P-value = 0.003

Critical values :

1%: -3.98926783537037 - The data is stationary with 99% confidence

5%: -3.425226747185185 - The data is stationary with 95% confidence

10%: -3.1357131066666666 - The data is stationary with 90% confidence

ADF test for N05B

Test statistic = -3.782

P-value = 0.018

Critical values :

1%: -3.9895792190177497 - The data is not stationary with 99% confidence

5%: -3.4253766620686186 - The data is stationary with 95% confidence

10%: -3.135801074760346 - The data is stationary with 90% confidence

ADF test for N05C

Test statistic = -14.974

P-value = 0.000

Critical values :

1%: -3.9891654349033057 - The data is stationary with 99% confidence

5%: -3.4251774443348975 - The data is stationary with 95% confidence

10%: -3.1356841756951854 - The data is stationary with 90% confidence

ADF test for R03

Test statistic = -3.806

P-value = 0.016

Critical values :

1%: -3.9898970270434053 - The data is not stationary with 99% confidence

5%: -3.4255296586298916 - The data is stationary with 95% confidence

10%: -3.1358908478232332 - The data is stationary with 90% confidence

ADF test for R06

Test statistic = -5.258

P-value = 0.000

Critical values :

1%: -3.9906647072202452 - The data is stationary with 99% confidence

5%: -3.4258991818227846 - The data is stationary with 95% confidence

10%: -3.1361076573601 - The data is stationary with 90% confidence

Augmented Dickey-Fuller (ADF) test have shown that all information, however N02BA (P-value=0.249) in the series were fixed/stationary, with maximum confidence.

```
from statsmodels.tsa.stattools import kpss
warnings.filterwarnings("ignore")
df=pd.read_csv('../input/salesweekly.csv')
for x in ['M01AB','M01AE','N02BA','N02BE', 'N05B','N05C','R0
3','R06']:
    print(" > Is "+x+" data stationary ?")
    dftest = kpss(np.log(df[x]), 'ct')
    print("Test statistic = {:.3f}".format(dftest[0]))
    print("P-value = {:.3f}".format(dftest[1]))
    print("Critical values :")
    for k, v in dftest[3].items():
        print("\t{}: {}".format(k, v))
```

output:

> Is M01AB data stationary ?

Test statistic = 0.285

P-value = **0.010**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is M01AE data stationary ?

Test statistic = 0.242

P-value = **0.010**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is N02BA data stationary ?

Test statistic = 0.147

P-value = **0.049**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is N02BE data stationary ?

Test statistic = 0.076

P-value = **0.100**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is N05B data stationary ?

Test statistic = 0.143

P-value = **0.056**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is N05C data stationary ?

Test statistic = nan

P-value = nan

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is R03 data stationary ?

Test statistic = 0.046

P-value = **0.100**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

> Is R06 data stationary ?

Test statistic = 0.027

P-value = **0.100**

Critical values :

10%: 0.119

5%: 0.146

2.5%: 0.176

1%: 0.216

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test found the trend non-stationarity in N02BE, R03 and R06.

REGULARITY ANALYSIS

For computing regularity and predictability of time series, **Surmised Entropy test** was utilized. For all series, entropy esteems were **higher** than 1 demonstrating low predictability, with highest values for M01AE, M01AB and N02BA.

```
df = pd.read_csv('../input/salesweekly.csv')
def ApEn(U, m, r):
    def _maxdist(x_i, x_j):
        return max([abs(ua - va) for ua, va in zip(x_i, x_j)])
    def _phi(m):
        x = [[U[j] for j in range(i, i + m - 1 + 1)] for i in range(N - m + 1)]
        C = [len([1 for x_j in x if _maxdist(x_i, x_j) <= r]) / (N - m + 1.0) for
x_i in x]
        return (N - m + 1.0)**(-1) * sum(np.log(C))
    N = len(U)
    return abs(_phi(m+1) - _phi(m))

for x in ['M01AB','M01AE','N02BA','N02BE','N05B','N05C','R03','R06']:
    print(x + ': ' + str(ApEn(df[x].values, m=2, r=0.2*np.std(df[x].values))))
```

M01AB: 1.141130089570642

M01AE: 1.166363924596575

N02BA: 1.1370638730125302

N02BE: 1.058024809082593

N05B: 1.074437415034502

N05C: 1.0361887401424648

R03: 1.1847216239035152

R06: 1.031759595747876

AUTOCORRELATION ANALYSIS

Definition of Autocorrelation :

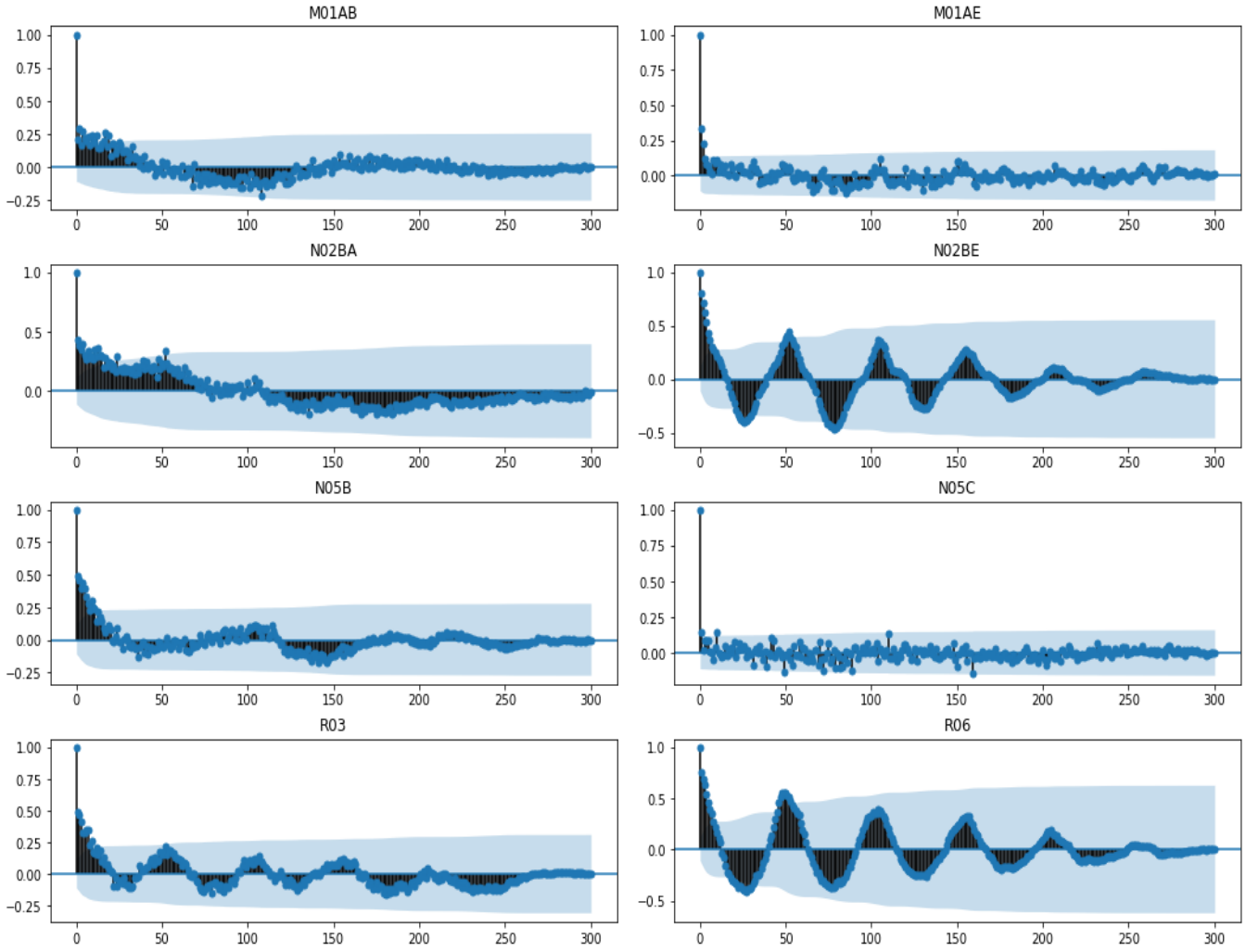
Autocorrelation is outlined as activity that shows associative degree extent to that two or lot of variables can fluctuate together along within the statistic .

Example of Autocorrelation in Time Series is If it's rainy these days then it's doubtness to rain tomorrow than it's clear these days then it's doubtness to rain tomorrow than it's clear these days ,associate degree An autocorrelation of +1 represents an ideal correlation, where as associate degree autocorrelation of -1 represents an ideal correlational statistics.

A plot of the autocorrelation of a time series by lag is named the AutoCorrelation Function (ACF). The plot is usually referred to as a correlogram or associate degree correlation plot. Plot shows the lag value on the coordinate axis and the correlation on the y-axis between -1 and 1. Confidence intervals square measure drawn as a cone. By default, this can be set to a ninety fifth confidence interval, suggesting that correlation values outside of this code square measure terribly seeming a correlation

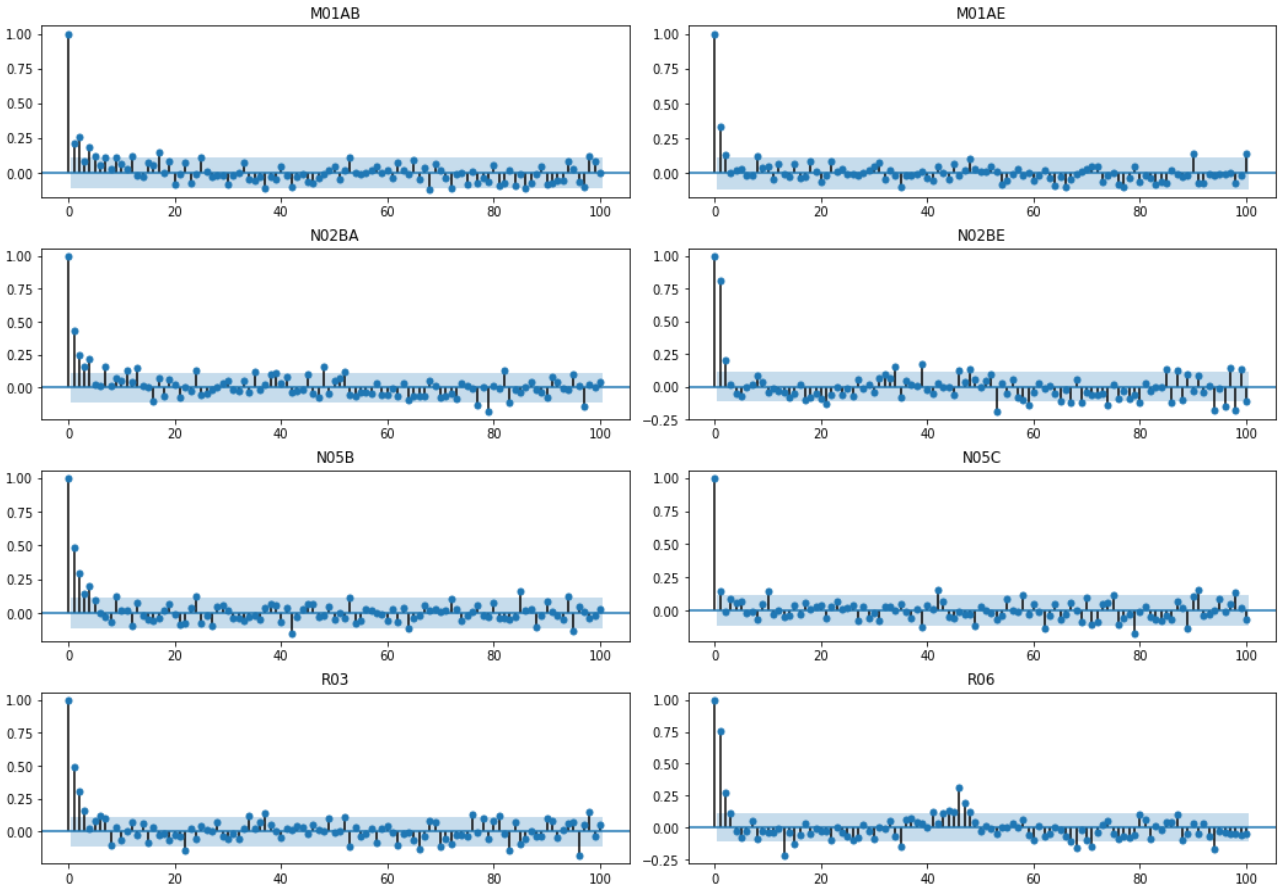
In [12]:

```
from statsmodels.graphics.tsaplots import plot_acf
df = pd.read_csv('../input/salesweekly.csv')
subplotindex=0
numrows=4
numcols=2
fig, ax = plt.subplots(numrows, numcols, figsize=(18,12))
plt.subplots_adjust(wspace=0.1, hspace=0.3)
with plt.rc_context():
    plt.rc("figure", figsize=(18,12))
    for x in ['M01AB','M01AE','N02BA','N02BE', 'N05B','N05C','R03','R06']:
        rowindex=math.floor(subplotindex/numcols)
        colindex=subplotindex-(rowindex*numcols)
        plot_acf(df[x], lags=300, title=x, ax=ax[rowindex,colindex])
        subplotindex=subplotindex+1
```



In general, the "partial" correlation between two variables is that the quantity of correlation between them that isn't explained by their mutual coorelation with a fixed set of alternative variables for instance if we tend to square measure regressing a variable Y on alternative statistics X1,X2,X3 the correlational statistics between Y and X3 is that the quantity of correlation between Y and X3 that's not explained by their common correlations with X1 and X2.

```
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.graphics.tsaplots import plot_acf
df = pd.read_csv('../input/salesweekly.csv')
subplotindex=0
numrows=4
numcols=2
fig, ax = plt.subplots(numrows, numcols, figsize=(18,12))
plt.subplots_adjust(wspace=0.1, hspace=0.3)
with plt.rc_context():
    plt.rc("figure", figsize=(14,6))
    for x in ['M01AB','M01AE','N02BA','N02BE', 'N05B','N05C','R03','R06']:
        rowindex=math.floor(subplotindex/numcols)
        colindex=subplotindex-(rowindex*numcols)
        plot_pacf(df[x], lags=100, title=x, ax=ax[rowindex,colindex])
        subplotindex=subplotindex+1
```



DATA DISTRIBUTION:

Chart with daily sales for various categories of interest is shown below. N02BE and N05B charts, although showing the similar trends, square measure suppresed due to the larger scale which makes the opposite illustrations less eligible

```
dfatch=pd.read_csv('../input/saleshourly.csv')
dfatch['datum']= pd.to_datetime(dfatch['datum'])

grp1=dfatch.groupby(dfatch.datum.dt.hour)['M01AB'].mean()
grp2=dfatch.groupby(dfatch.datum.dt.hour)['M01AE'].mean()
grp3=dfatch.groupby(dfatch.datum.dt.hour)['N02BA'].mean()
grp6=dfatch.groupby(dfatch.datum.dt.hour)['N05C'].mean()
grp7=dfatch.groupby(dfatch.datum.dt.hour)['R03'].mean()
grp8=dfatch.groupby(dfatch.datum.dt.hour)['R06'].mean()

plt.title('Daily average sales')
plt.xlabel('Time of day')
plt.ylabel('Quantity of sale')

grp1.plot(figsize=(8,6))
grp2.plot(figsize=(8,6))
grp3.plot(figsize=(8,6))
grp6.plot(figsize=(8,6))
grp7.plot(figsize=(8,6))
grp8.plot(figsize=(8,6))

plt.legend(['M01AB', 'M01AE', 'N02BA', 'N05C', 'R03', 'R06'], loc='upper left')

plt.show()
```

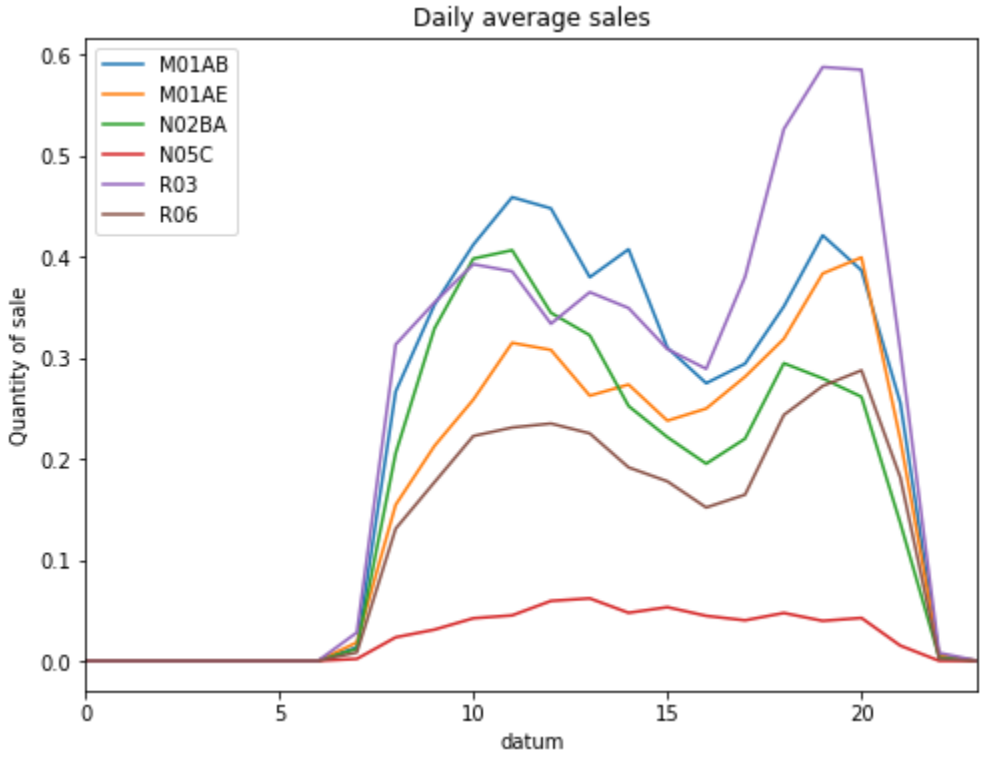
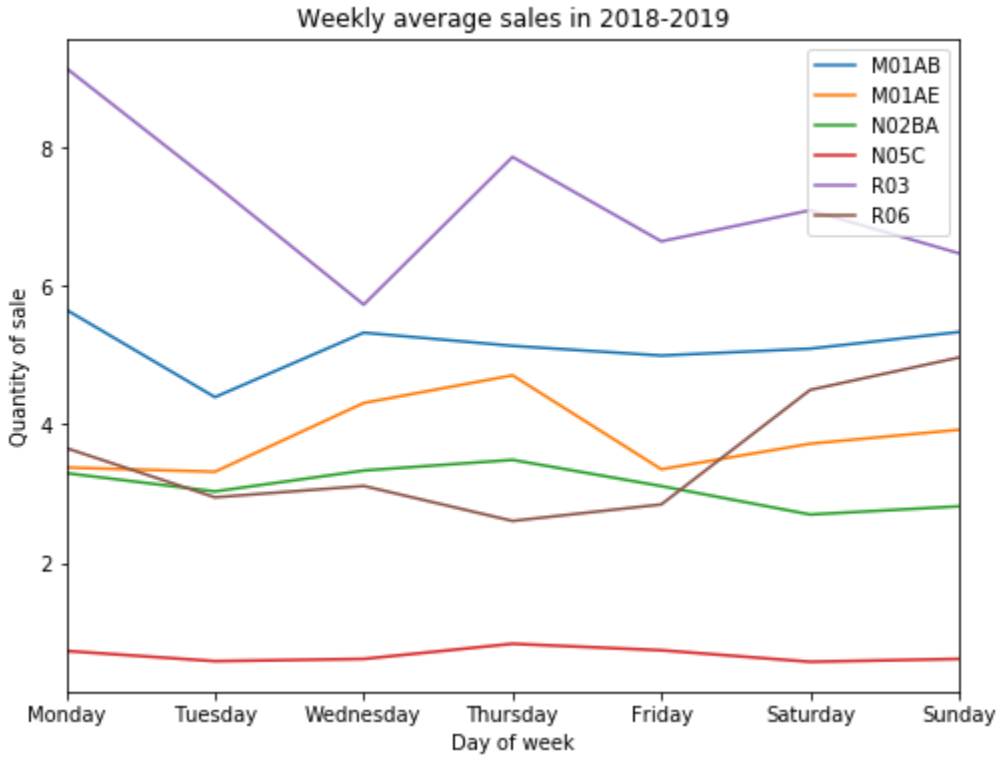


Chart with weekly sales for various categories of interest was shown below. N02BE and N05B charts, although showing the similar trends, square measure suppresed due to the larger scale that makes the opposite illustrations **less eligible**.

```
dfatcw=pd.read_csv('../input/salesdaily.csv')
dfatcw['datum']= pd.to_datetime(dfatcw['datum'])
days = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday', 'Sunday']
plt.rcParams.update({'font.size': 10})

grp1=dfatcw.loc[dfatcw.datum>'2018-01-01'].groupby(dfatch.datum.dt.weekday_name)['M01AB'].me
an().reindex(days)
grp2=dfatcw.loc[dfatcw.datum>'2018-01-01'].groupby(dfatch.datum.dt.weekday_name)['M01AE'].me
an().reindex(days)
grp3=dfatcw.loc[dfatcw.datum>'2018-01-01'].groupby(dfatch.datum.dt.weekday_name)['N02BA'].me
an().reindex(days)
grp6=dfatcw.loc[dfatcw.datum>'2018-01-01'].groupby(dfatch.datum.dt.weekday_name)['N05C'].mea
n().reindex(days)
grp7=dfatcw.loc[dfatcw.datum>'2018-01-01'].groupby(dfatch.datum.dt.weekday_name)['R03'].mean
().reindex(days)
grp8=dfatcw.loc[dfatcw.datum>'2018-01-01'].groupby(dfatch.datum.dt.weekday_name)['R06'].mean
().reindex(days)
```

```
grp1.plot(figsize=(8,6))
grp2.plot(figsize=(8,6))
grp3.plot(figsize=(8,6))
grp6.plot(figsize=(8,6))
grp7.plot(figsize=(8,6))
grp8.plot(figsize=(8,6))

plt.legend(['M01AB', 'M01AE', 'N02BA', 'N05C', 'R03', 'R06'], loc='upper right')
plt.title('Weekly average sales in 2018-2019')
plt.xlabel('Day of week')
plt.ylabel('Quantity of sale')

plt.show()
```



TIME SERIES FORECASTING

Forecasting models were fitted with week by week time-series information with dataset of 302 lines. Three determining techniques were tested: ARIMA, Prophet and LSTM. Train-test split strategy was utilized (52 weeks of test information). MSE was utilized as measurements for looking at the exhibition and furthermore as a misfortune work for LSTM. Mean Absolute Percentage Error (MAPE) is additionally estimated for every one of the situations. To characterize the gauge determining exactness to improve from, three tests were performed. Average method was used as a baseline for long-term forecasting, while Naïve and Seasonal Naïve were used for rolling forecasts.

Forecasting results are approved by utilizing two methodologies: short-term and long-term forecasts. For short-term forecasts (or so-called rolling-forecast)approval walk-forward model validation, train-test split approval is acted in iterative style, where perceptions are added to the preparation dataset after each individual week by week deals expectation, while the model is fitted in every emphasis. Along these lines, during testing, forecast in a timestep t depends on the model which fits the preparation set comprising of perceptions in timesteps (0,t-1), or: f(t) = f(o[0:t-1]). Long-term forecasts approval depends on MSE of one-year estimate, contrasted with actual observations.

All MSE metrics will be put away in a data frame, which will be later use for show of the general outcomes and correlation. Rolling forecasting will be completed by utilizing 5 technique(Naïve, Seasonal Naïve, ARIMA, Auto ARIMA, Prophet). Long-term forecasting will be completed by utilizing 7 strategies (Average, ARIMA, Auto ARIMA, Prophet, Vanilla LSTM, Stacked LSTM, Bidirectional LSTM).

```
resultsRolling={'M01AB':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'M01AE':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N02BA':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N02BE':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N05B':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N05C':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'R03':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'R06':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]}
resultsRollingdf = pd.DataFrame(resultsRolling)
```

```
resultsRollingdf.index = ['Naive MSE', 'Naive MAPE', 'Seasonal Naive MSE', 'Seasonal Naive MAPE',
                          'ARIMA MSE', 'ARIMA MAPE', 'AutoARIMA MSE', 'AutoARIMA MAPE',
                         'Prophet MSE', 'Prophet MAPE']
resultsLongterm={'M01AB':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'M01AE':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N02BA':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N02BE':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N05B':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'N05C':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'R03':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0],
     'R06':[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]}
resultsLongtermdf = pd.DataFrame(resultsLongterm)
resultsLongtermdf.index = ['Average MSE', 'Average MAPE', 'ARIMA MSE', 'ARIMA MAPE',
                           'AutoARIMA MSE', 'AutoARIMA MAPE', 'Prophet MSE', 'Prophet MAPE',
                           'Vanilla LSTM MSE', 'Vanilla LSTM MAPE', 'Stacked LSTM MSE', 'Stacked LSTM MAPE',
                           'Bidirectional LSTM MSE', 'Bidirectional LSTM MAPE']
```

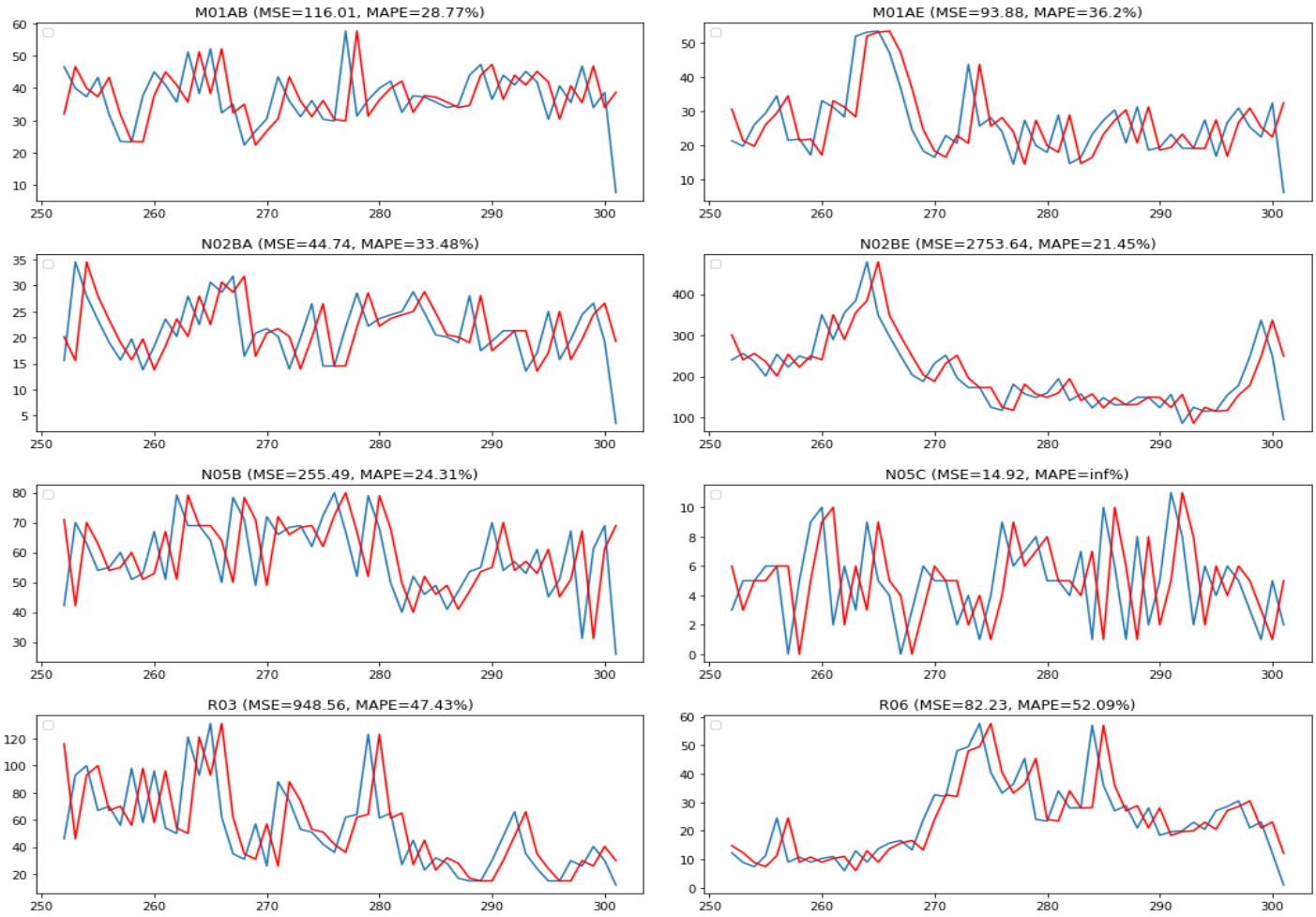
BASELINE FORECASTING ACCURACY:

In this subsection, three summaries of the reference Naïve methods are provided. First, Naïve forecasting was done and results introduced . Second, average method was utilized to figure. At long last, seasonal Naïve forecast was completed for the series that has been found as occasional: N02BE, R03 and R06.

Naïve forecasting:

```
df=pd.read_csv('../input/salesweekly.csv')

subplotindex=0

numrows=4

numcols=2

fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))

plt.subplots_adjust(wspace=0.1, hspace=0.3)
```

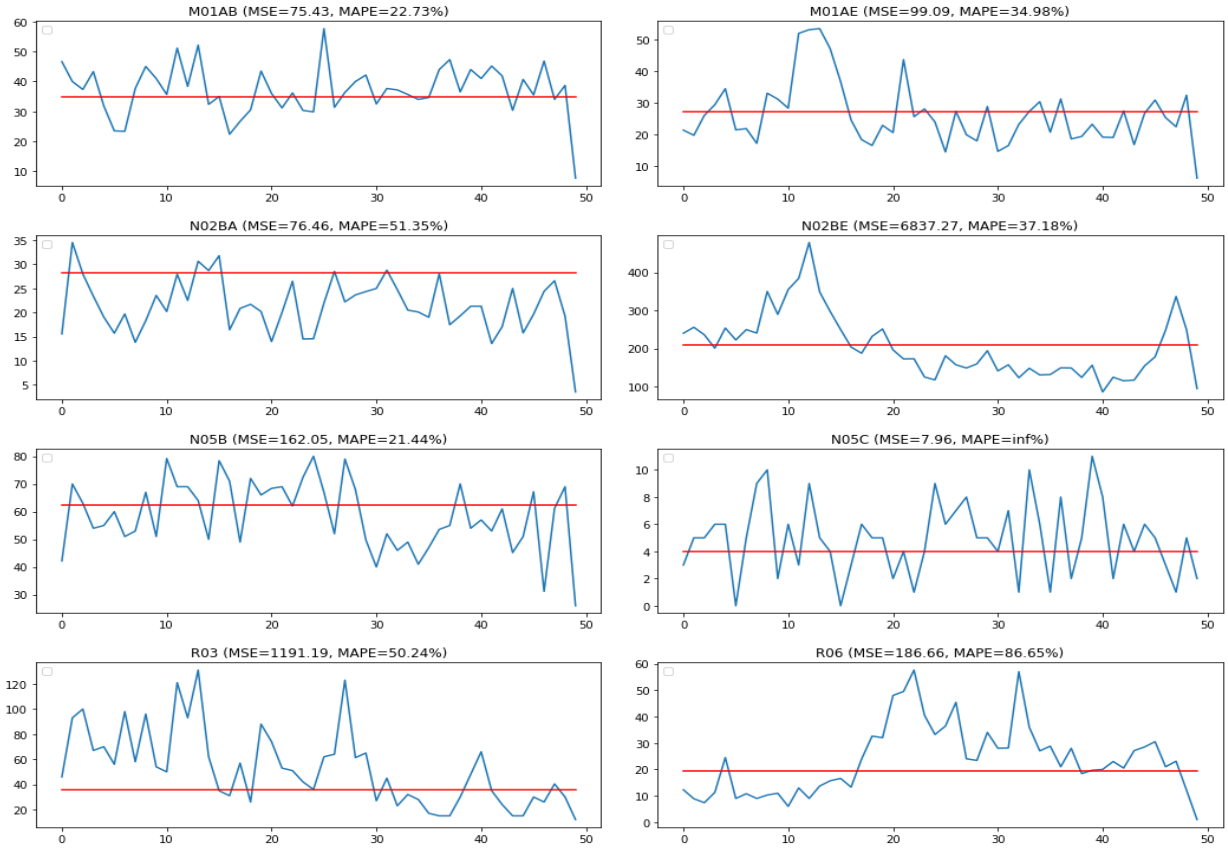
```
for x in ['M01AB','M01AE','N02BA','N02BE','N05B','N05C','R03','R06']:

    rowindex=math.floor(subplotindex/numcols)

    colindex=subplotindex-(rowindex*numcols)

    ds=df[x]

    dataframe = concat([ds.shift(1), ds], axis=1)

    dataframe.columns = ['t+1', 't-1']

    size = len(dataframe)-50

    X=dataframe['t-1']

    Y=dataframe['t+1']

    test, predictions = X[size:len(X)], Y[size:len(Y)]

    error = mean_squared_error(test, predictions)

    perror = mean_absolute_percentage_error(test, predictions)

    resultsRollingdf.loc['Naive MSE',x]=error

    resultsRollingdf.loc['Naive MAPE',x]=perror

    ax[rowindex,colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE='+ s
tr(round(perror,2)) +'%)')

    ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')

    ax[rowindex,colindex].plot(test)

    ax[rowindex,colindex].plot(predictions, color='red')

    subplotindex=subplotindex+1

plt.show()
```



Average method forecasting:

```
  df=pd.read_csv('../input/salesweekly.csv')
subplotindex=0
numrows=4
numcols=2
fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))
plt.subplots_adjust(wspace=0.1, hspace=0.3)
for x in ['M01AB','M01AE','N02BA','N02BE','N05B','N05C','R03','R06']:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
    X=df[x].values
    size = len(X)-50
    test = X[size:len(X)]
    mean = np.mean(X[0:size])
    predictions = np.full(50,mean)
    error = mean_squared_error(test, predictions)
```

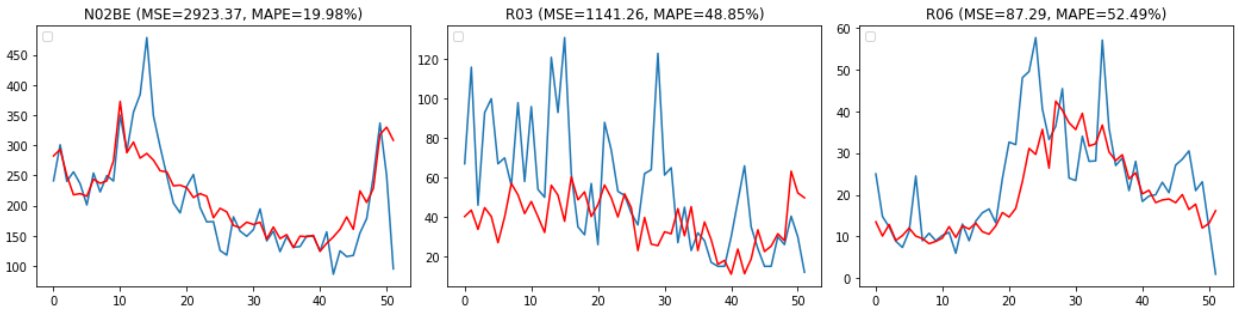
```
    perror = mean_absolute_percentage_error(test, predictions)
    resultsLongtermdf.loc['Average MSE',x]=error
    resultsLongtermdf.loc['Average MAPE',x]=perror
    ax[rowindex,colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE='+ s
tr(round(perror,2)) +'%)')
    ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[rowindex,colindex].plot(test)
    ax[rowindex,colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```



Seasonal Naïve forecasting:

```
df=pd.read_csv('../input/salesweekly.csv')
subplotindex=0
numrows=1
numcols=3
fig, ax = plt.subplots(numrows, numcols, figsize=(18,4))
plt.subplots_adjust(wspace=0.1, hspace=0.3)
for x in ['N02BE','R03','R06']:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
```

```
    X=df[x].values
    size = len(X)-52
    test = X[size:len(X)]
    train = X[0:size]
    predictions=list()
    history = [x for x in train]
    for i in range(len(test)):
        obs=list()
        for y in range(1,5):
            obs.append(train[-(y*52)+i])
        yhat = np.mean(obs)
        predictions.append(yhat)
        history.append(test[i])
    error = mean_squared_error(test, predictions)
    perror = mean_absolute_percentage_error(test, predictions)
    resultsRollingdf.loc['Seasonal Naive MSE',x]=error
    resultsRollingdf.loc['Seasonal Naive MAPE',x]=perror
    ax[colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE='+ str(round(
perror,2)) +'%)')
    ax[colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[colindex].plot(test)
    ax[colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```



ARIMA FORECASTING:-

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a class of model that captures a suite of different standardtemporal structures in time series data.

- First, method **arma_order_select_ic** was used to determine initial p and q parameters. The method computes Akaike's Information Criterion **(AIC)** for many ARIMA models and chooses the best configuration.
- The best-fit model according to AIC is the one that explains the **greatest amount of variation** using the **fewest possible independent variables.**
- **Lower** AIC scores are better.

ARMA(p,q,M01AB) = (3, 3) is the best.
ARMA(p,q,M01AE) = (2, 0) is the best.ARMA(p,q,N02BA) = (5, 4) is the best.ARMA(p,q,N02BE) = (2, 2) is the best.
ARMA(p,q,N05B) = (4, 3) is the best. ARMA(p,q,N05C) = (2, 3) is the best. ARMA(p,q,R03) = (3, 3) is the best.
ARMA(p,q,R06) = (2, 2) is the best.

However, AIC is not used to score accuracy of the forecasting methods inthis research. **Mean squared error** is used instead. For that, reason, grid search optimization method was applied, where different combinations of the hyper-parameters were used to calculate MSE and then, the combination producing the least MSE was chosen as optimal.

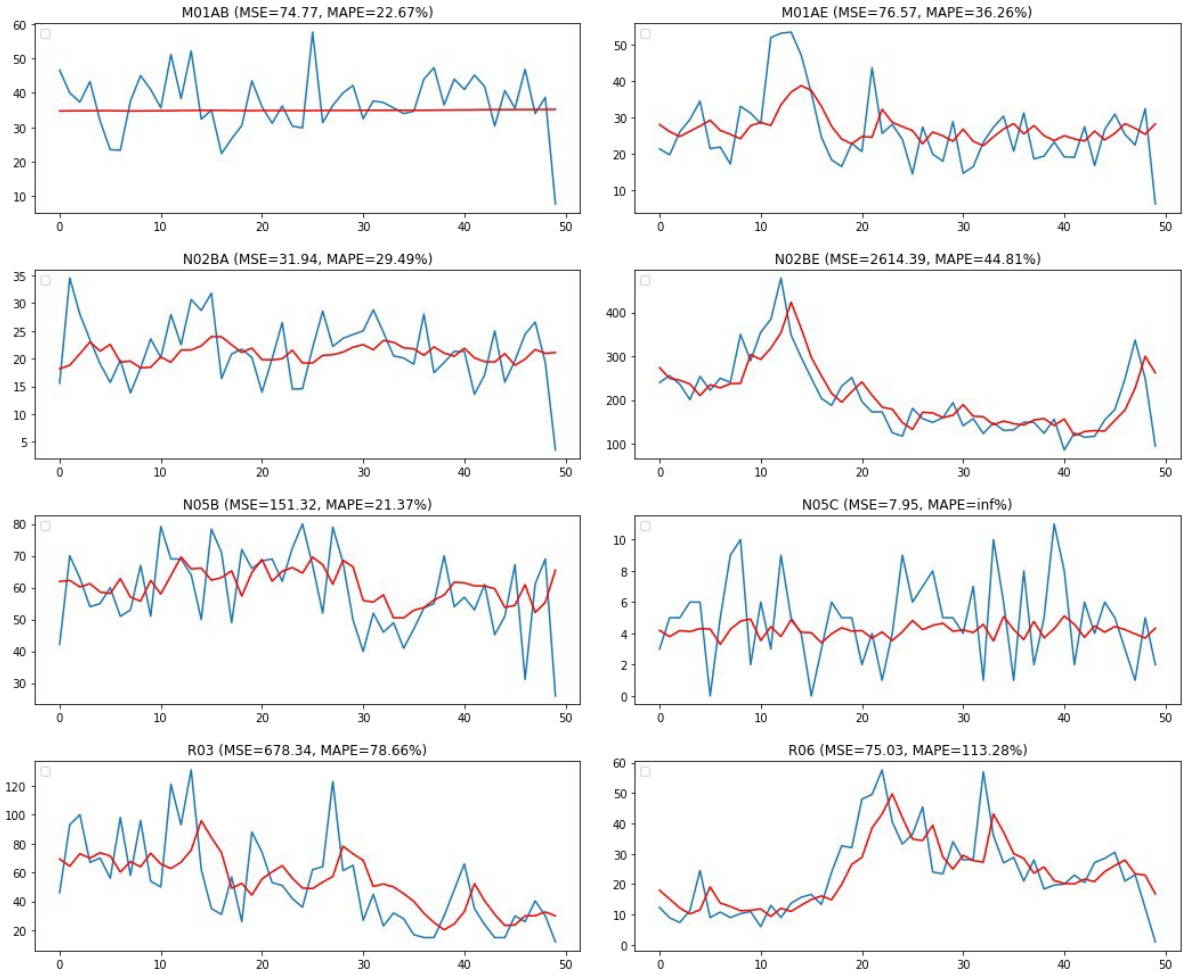
M01AB - Best ARIMA(0, 0, 0) MSE=61.971

M01AE - Best ARIMA(2, 0, 0) MSE=54.293 N02BA - Best ARIMA(5, 1, 1) MSE=29.998 N02BE - Best ARIMA(2, 0, 0) MSE=2080.523N05B - Best ARIMA(0, 0, 5) MSE=195.096 N05C - Best ARIMA(0, 0, 1) MSE=11.383 R03 - Best ARIMA(5, 1, 1) MSE=507.120 R06 - Best

ARIMA(1, 0, 1)
MSE=67.824

MAPE	Interpretation
<10	Highly accurate forecasting
10-20	Good forecasting
20-50	Reasonable forecasting
>50	Inaccurate forecasting

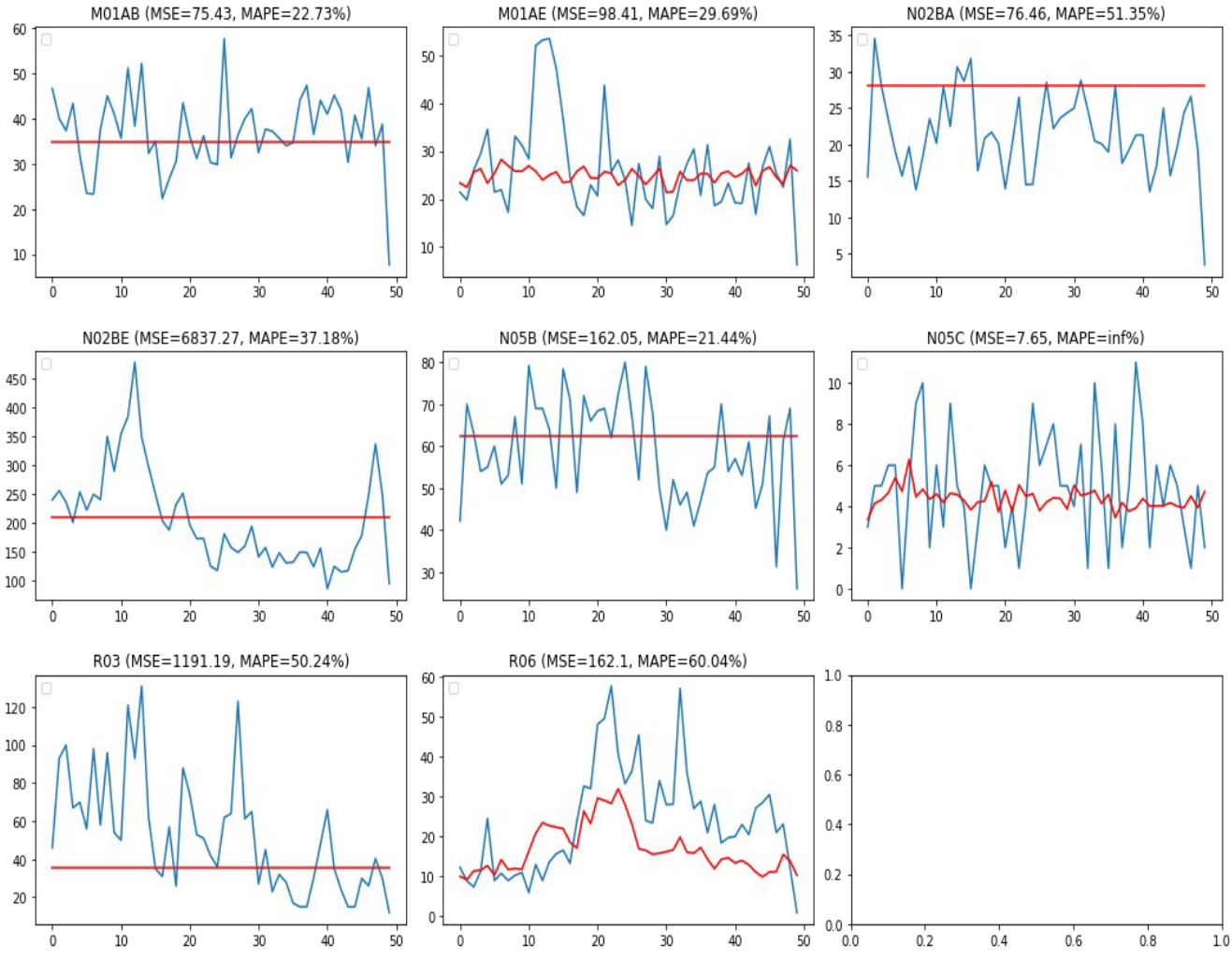
Source: Lewis (1982, p. 40)



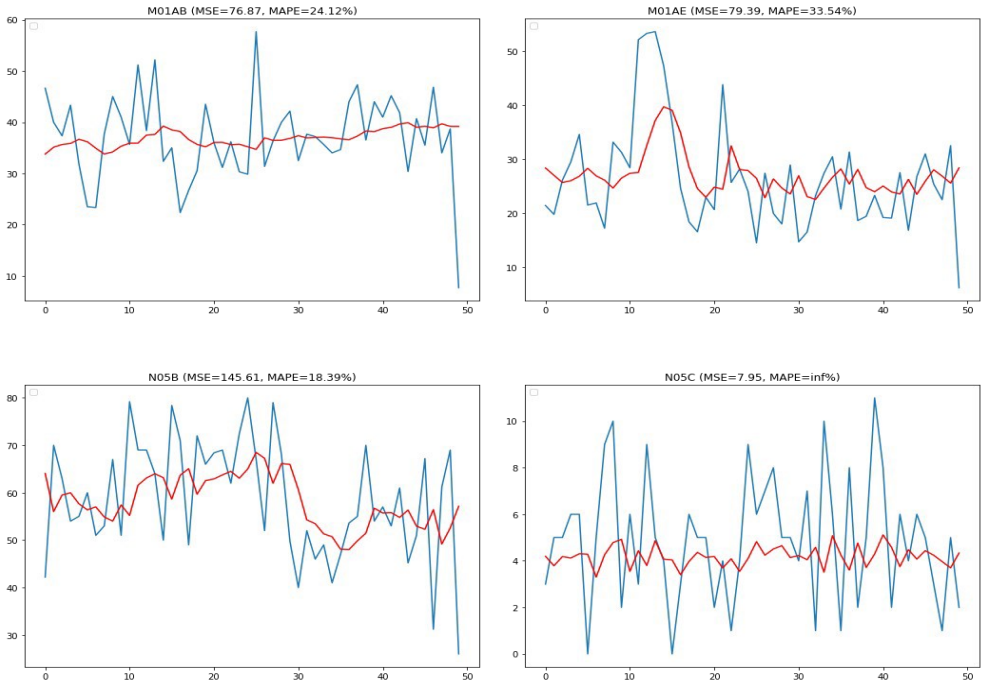
This is an **untrended seasonal** pattern.

Long-term forecasting with ARIMA model

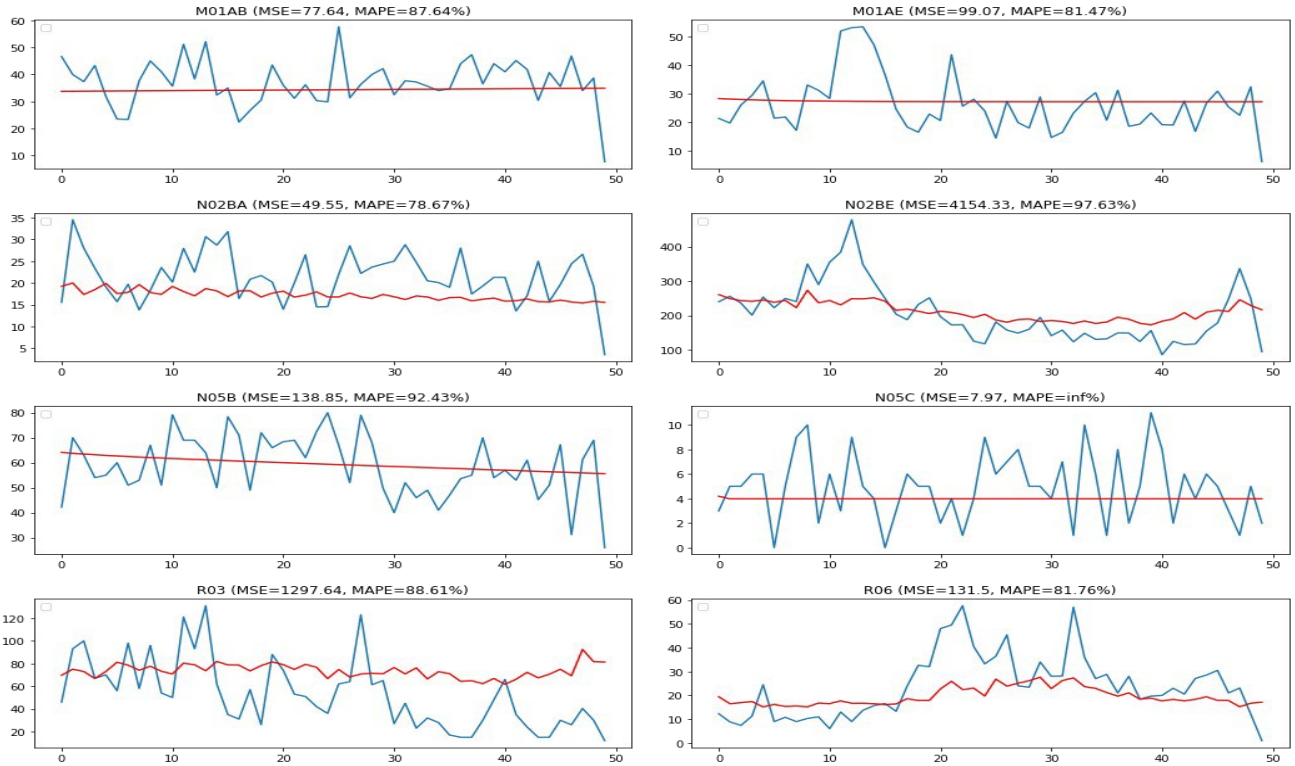
Long-term forecasting is done for a period ranging from six months to five years.



Rolling forecasting with Auto-ARIMA model



Long-term forecasting with Auto-ARIMA model



CONCLUSION:

To conclude , time-series time-series analyses and forecasts have directed conceivably valuable ends and proposals to the drug store. Every day, week after week and yearly irregularity investigation were demonstrated valuable for distinguishing the periods in which extraordinary deals and promoting efforts could be executed, with the exception of N05B and N05C classes of medications which didn't show huge normalities. Conjectures have demonstrated better compared to Innocent techniques and in satisfactory stretches for long haul arranging. Almost certainly, the conjectures could be altogether improved by growing the issue extension to multivariate time series estimating and by including illustrative factors, for example,

- Climate information. Deals of antirheumatic sedates in M01AB and M01AE classifications could be impacted by the progressions of environmental strain. Unexpected decreases in everything classes could be clarified by outrageous climate conditions, like weighty downpour, rainstorms and snowstorms.
- Cost of the medications. Deals spikes might be clarified by the limits, applied in a present moment. Presenting this component might work with consider the possibility that estimating examination of deals execution during advertising efforts including value decreases.
- Dates of the annuity result. Deals spikes are apparent at the dates of state benefits result.
- Public occasions, as non-working days with occasional examples like Sundays are relied upon to disturb day by day deals.

Future work on univariate time series forecasting incorporates expanding the quantity of information, investigating diverse other exactness measurements, streamlining of hyper-boundaries for LSTM models and testing different structures, like CNN LSTM and ConvLSTM. In any case, key enhancements in deals estimating are normal from diminishing the vulnerability of the models by growing to multivariate time series forecasting issue, as clarified previously.

THANK YOU