STOCK PRICE FORECASTING

Aditya Shrish Puranik
Rishika Chaudhary
Vaibhav Sachdeva

Under the Supervision of
Prof. SK Neogy

DATASET

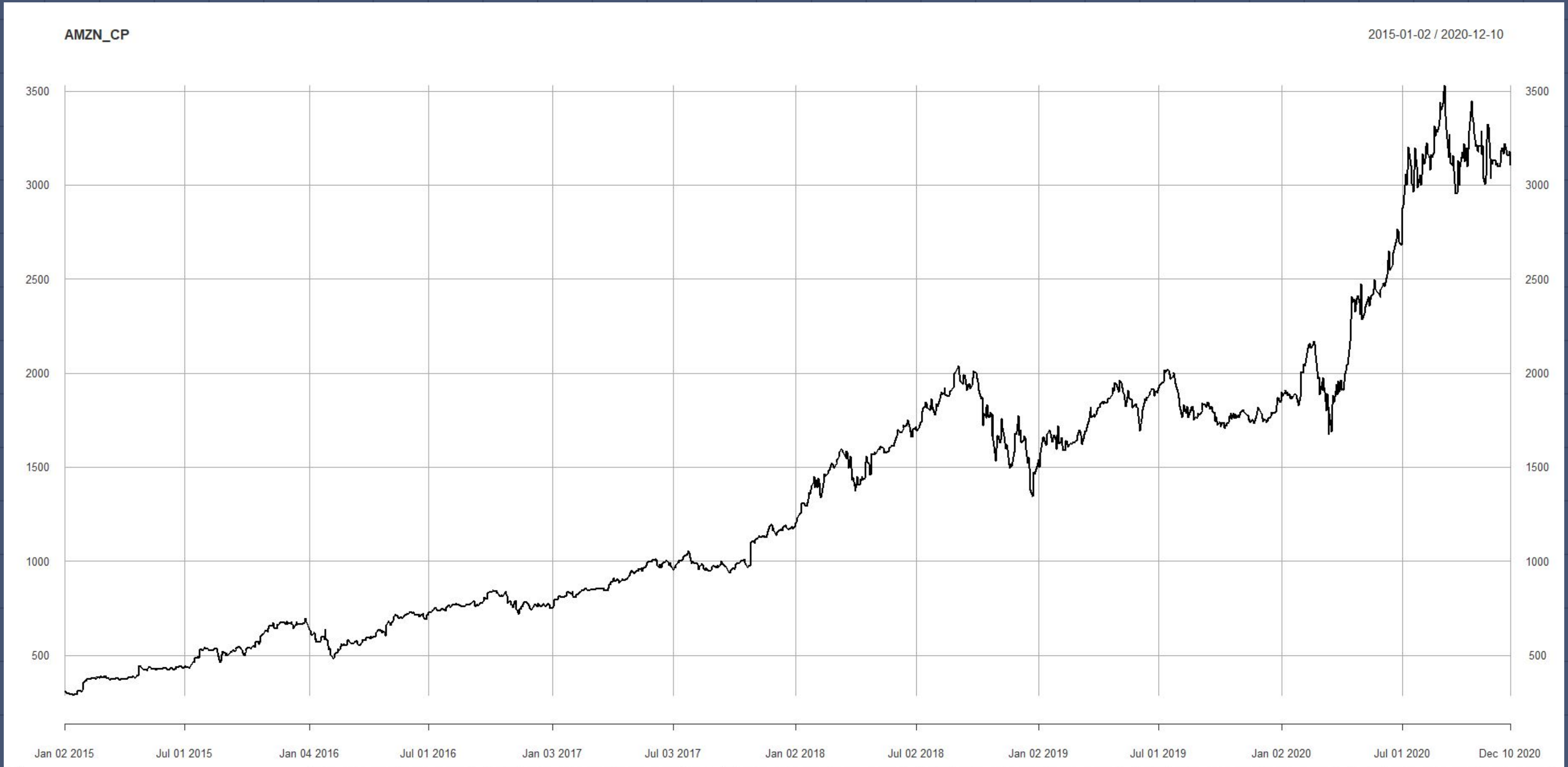
```
> head(AMZN)
```

	AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume	AMZN.Adjusted
2015-01-02	312.58	314.75	306.96	308.52	2783200	308.52
2015-01-05	307.01	308.38	300.85	302.19	2774200	302.19
2015-01-06	302.24	303.00	292.38	295.29	3519000	295.29
2015-01-07	297.50	301.28	295.33	298.42	2640300	298.42
2015-01-08	300.32	303.14	296.11	300.46	3088400	300.46
2015-01-09	301.48	302.87	296.68	296.93	2592400	296.93

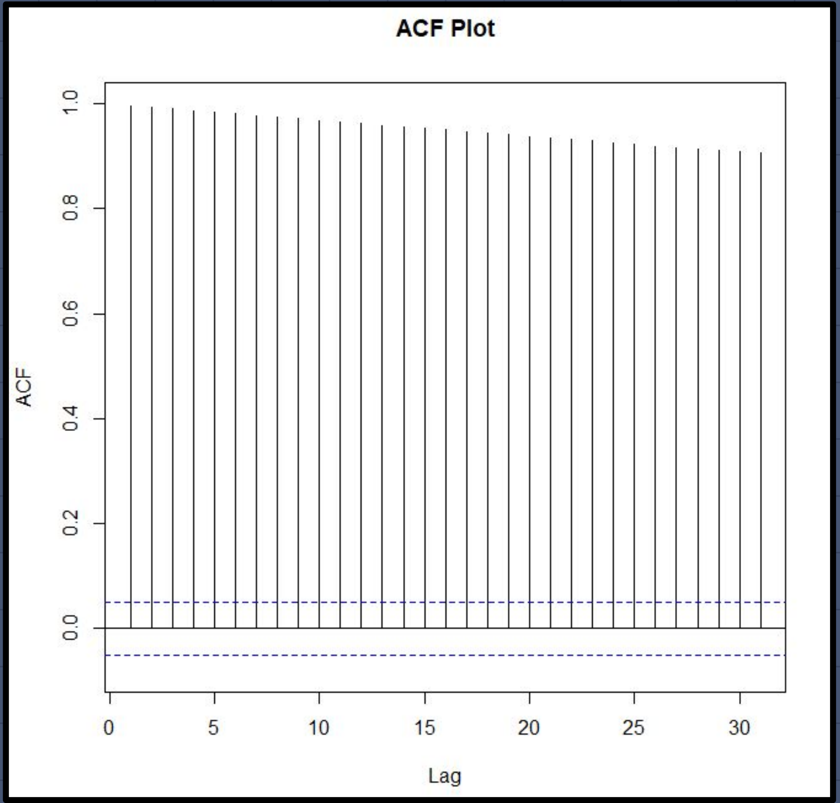
```
> tail(AMZN)
```

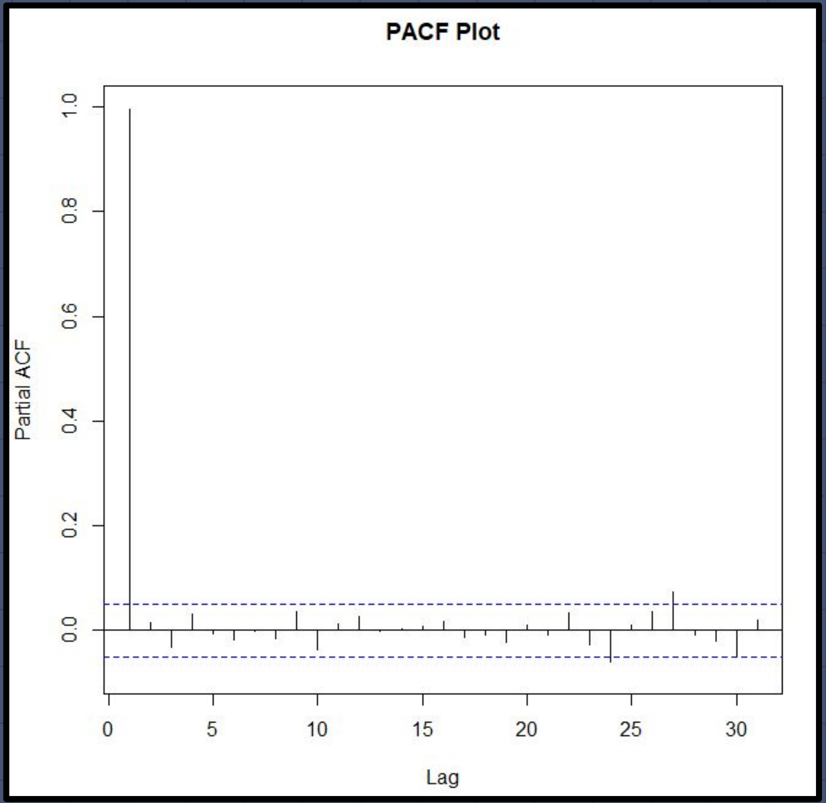
	AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume	AMZN.Adjusted
2020-12-03	3205.46	3228.64	3181.31	3186.73	2892000	3186.73
2020-12-04	3198.21	3198.21	3158.76	3162.58	2913600	3162.58
2020-12-07	3156.48	3180.76	3141.69	3158.00	2751300	3158.00
2020-12-08	3158.90	3184.13	3120.02	3177.29	3286300	3177.29
2020-12-09	3167.89	3174.43	3088.00	3104.20	4100800	3104.20
2020-12-10	3088.99	3142.10	3076.00	3101.49	3023600	3101.49

- Amazon Stock Price data from 02[nd] January 2015 to 10[th] December 2020.
- Closing price of each day would be used for model training and prediction.



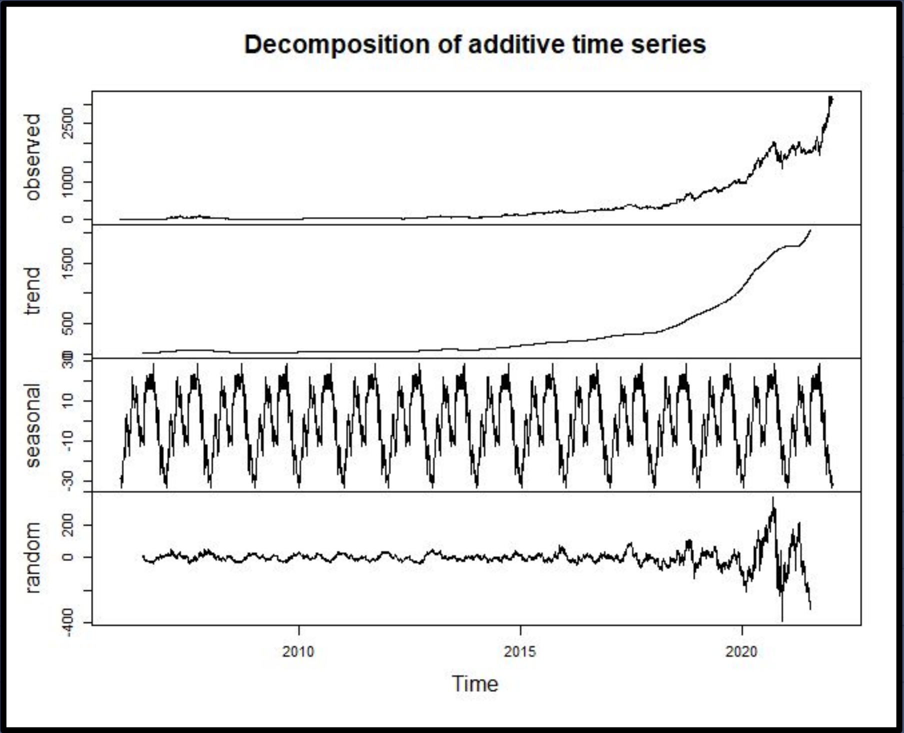
The above graph shows Amazon's stock price trend, where a sharp rise from October 2017 can be observed.

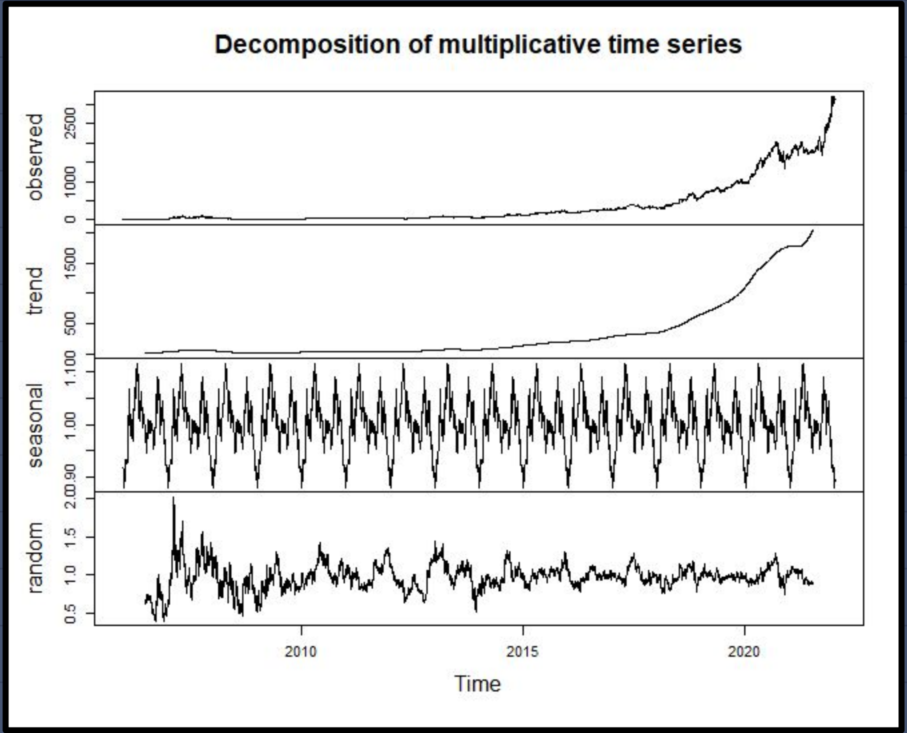




ACF Plot

PACF Plot





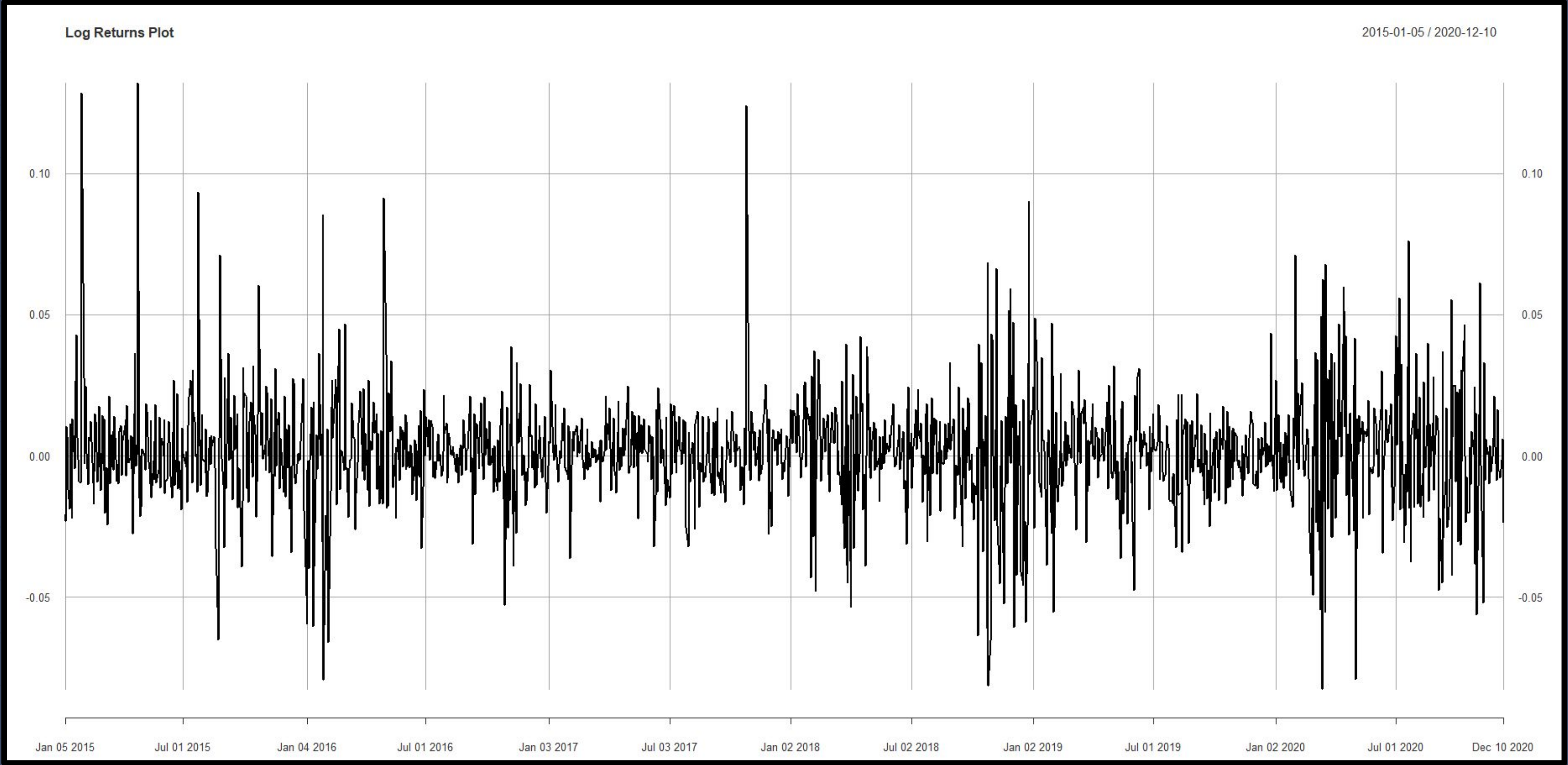
Additive Series Decomposition

Multiclipative Series Decomposition

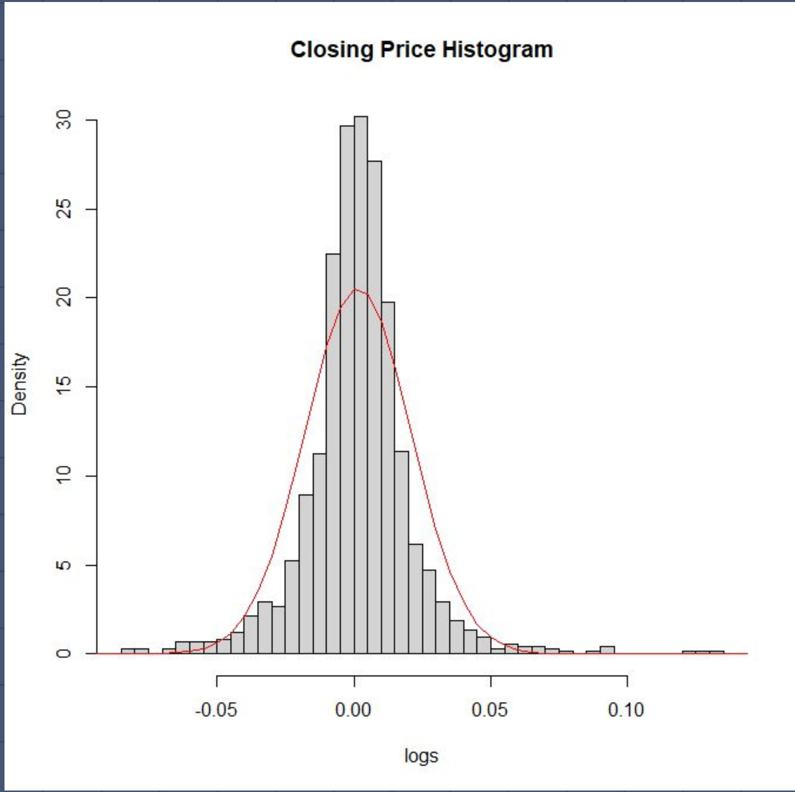
Augmented Dickey Fuller Test

```
> print(adf.test(AMZN_CP))

        Augmented Dickey-Fuller Test

data:  AMZN_CP
Dickey-Fuller = -1.7076, Lag order = 11, p-value = 0.7021
alternative hypothesis: stationary
```

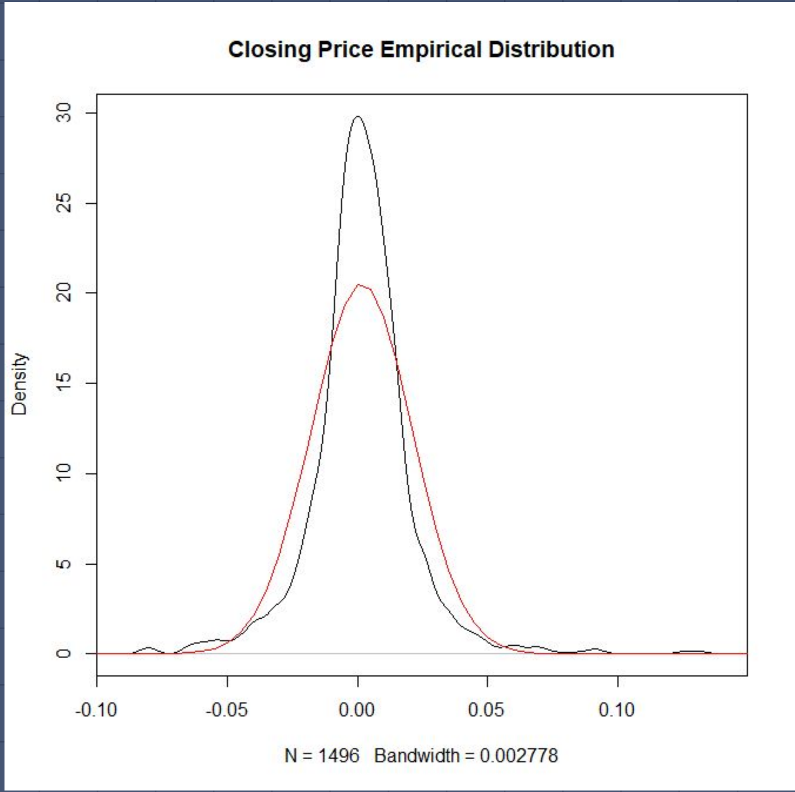
```
> print(adf.test(logs))

        Augmented Dickey-Fuller Test

data:  logs
Dickey-Fuller = -11.629, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```



Log Returns Plot of Amazon Closing Price



Histogram of Amazon Closing Price



Empirical Distribution of Amazon Closing Price

ARIMA MODELING

- ARIMA stands for **Auto-Regressive Integrated Moving Average.**
- Used for forecasting a time series which can be made to be "**stationary**" by differencing.
- ARIMA predictor for linear equation consist of *lags of the dependent* variable and/or *lags of the forecast errors.*
- Parameters of ARIMA model – **AR, I and MA (p, d , q)** where:
    - **p** is the number of autoregressive terms.
    - **d** is the number of non-seasonal differences needed for stationarity.
    - **q** is the number of lagged forecast errors in the prediction equation.

Parameter Optimization - Akaike Information Criterion (AIC)

ARIMA FITTING

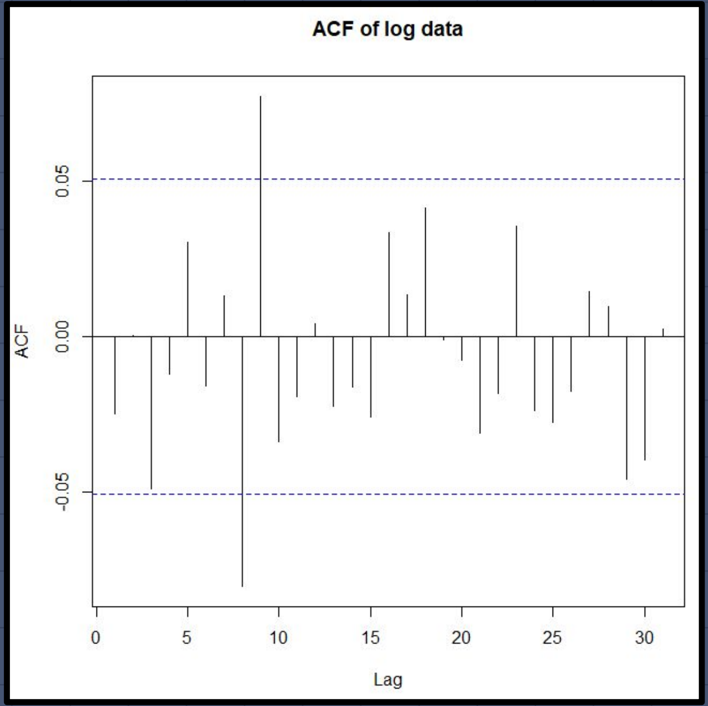
```
> auto.arima(AMZN_CP, seasonal=FALSE)
Series: AMZN_CP
ARIMA(1,1,1) with drift

Coefficients:
          ar1     ma1   drift
      -0.5901  0.5218  1.8603
s.e.   0.1248  0.1303  0.8033

sigma^2 estimated as 1056:  log likelihood=-7329.06
AIC=14666.12   AICc=14666.15   BIC=14687.37
```

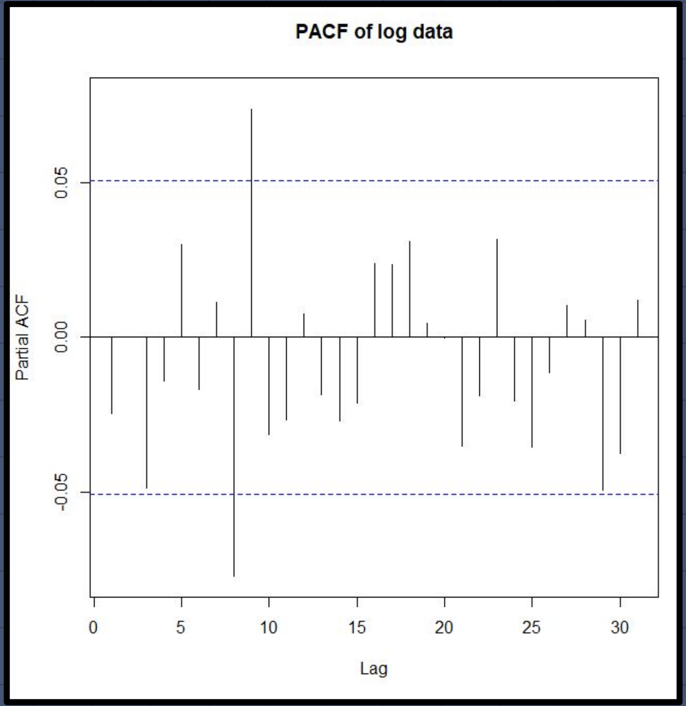
ARIMA with seasonal = FALSE
MAPE = 1.3216

```
> auto.arima(AMZN_CP, lambda = "auto")
Series: AMZN_CP
ARIMA(2,1,2) with drift
Box Cox transformation: lambda= -0.1976275

Coefficients:
          ar1      ar2     ma1     ma2  drift
      -0.3073  -0.8811  0.3201  0.9114  4e-04
s.e.   0.1380   0.1063  0.1240  0.0902  1e-04

sigma^2 estimated as 2.384e-05:  log likelihood=5841.86
AIC=-11671.71   AICc=-11671.66   BIC=-11639.85
```

ARIMA with lambda = auto
MAPE = 1.3169

ARIMA FITTING

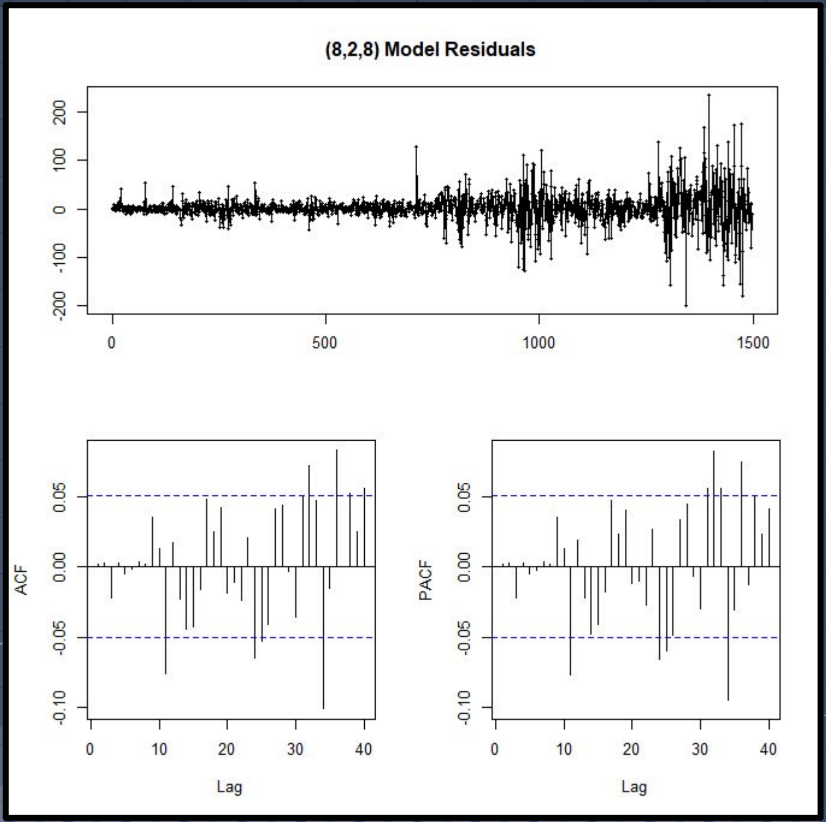


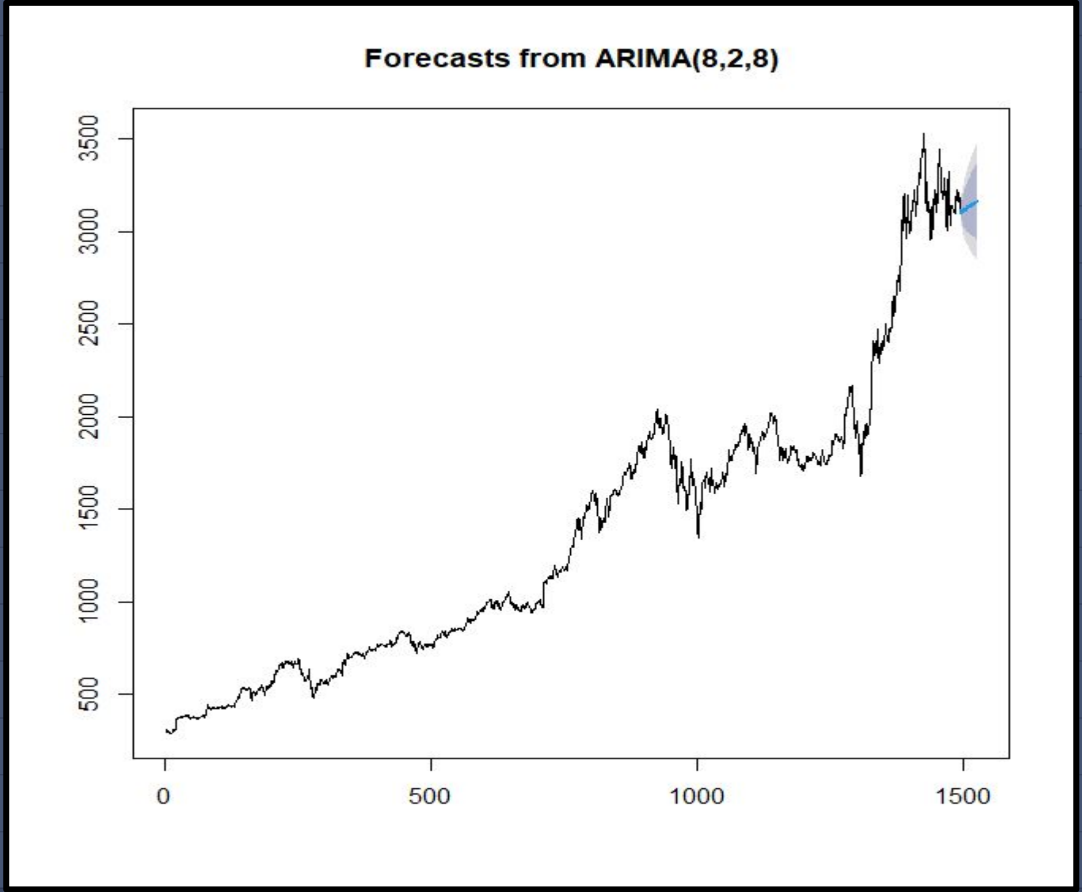


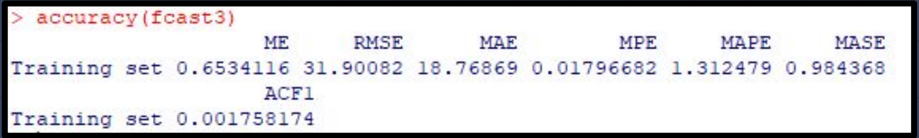
Optimizing the value of p,d,q for ARIMA using ACF and PACF of log data

OPTIMIZED ARIMA MODEL

Order = (8, 2, 8)







```
> accuracy(fcast3)
                    ME     RMSE      MAE        MPE     MAPE     MASE
Training set 0.6534116 31.90082 18.76869 0.01796682 1.312479 0.984368
                    ACF1
Training set 0.001758174
```

Order = (8, 2, 8)
MAPE = 1.31248

GARCH MODELING

- GARCH stands for Generalized Autoregressive Conditional Heteroskedasticity.
- ARCH model is based on an autoregressive representation of the conditional variance.
- Extension of ARCH model - it incorporates a MA (Moving Average) component together with autoregressive component.
- Includes lag variance terms, along with lag residual errors from a mean process.
- MA component allows to both *model the conditional change in variance over time* and *changes in time-dependent variance*.
- Parameters of GARCH model – **(p, q)** where:
  - "p" - number of lag variances
  - "q" - number of lag residual errors

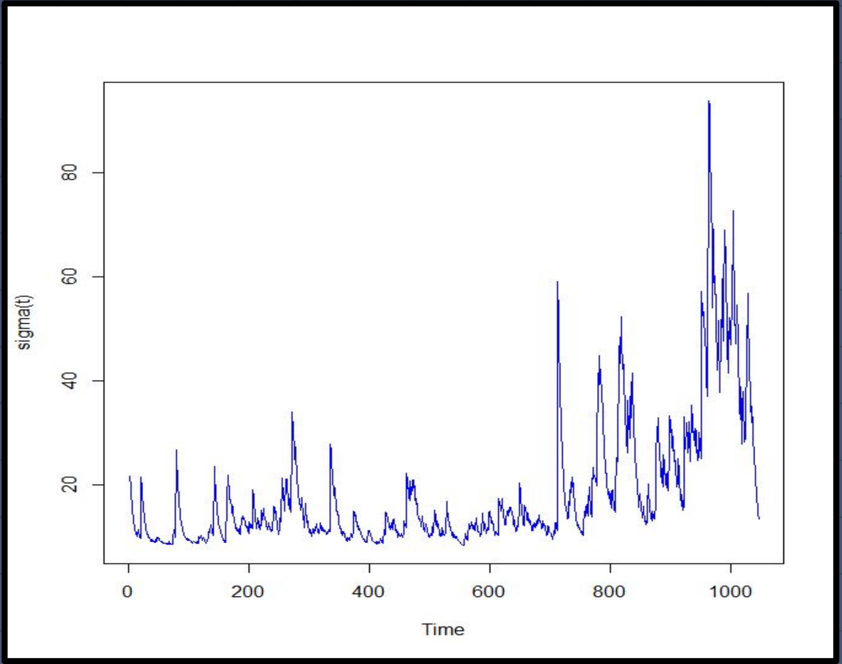
$$\varepsilon_t = \sigma_t \eta_t$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

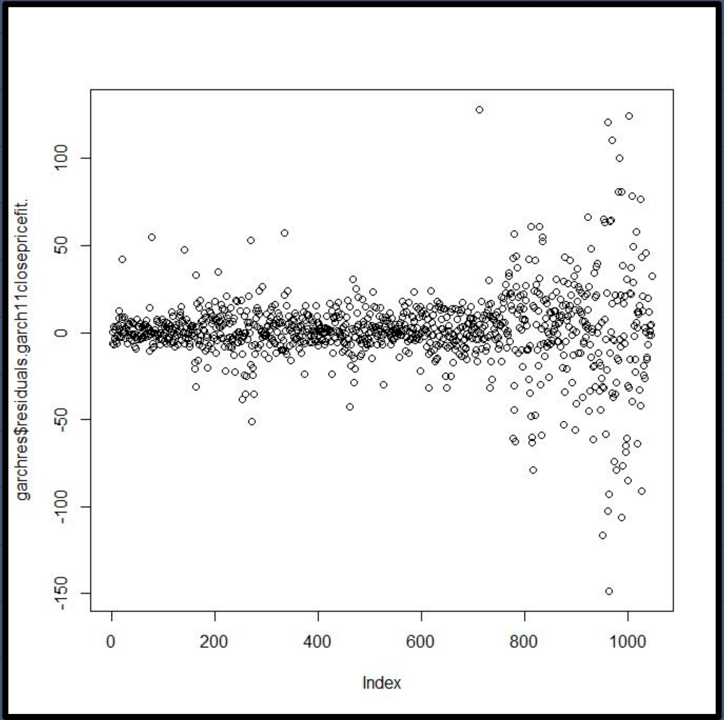
GARCH FORECASTING

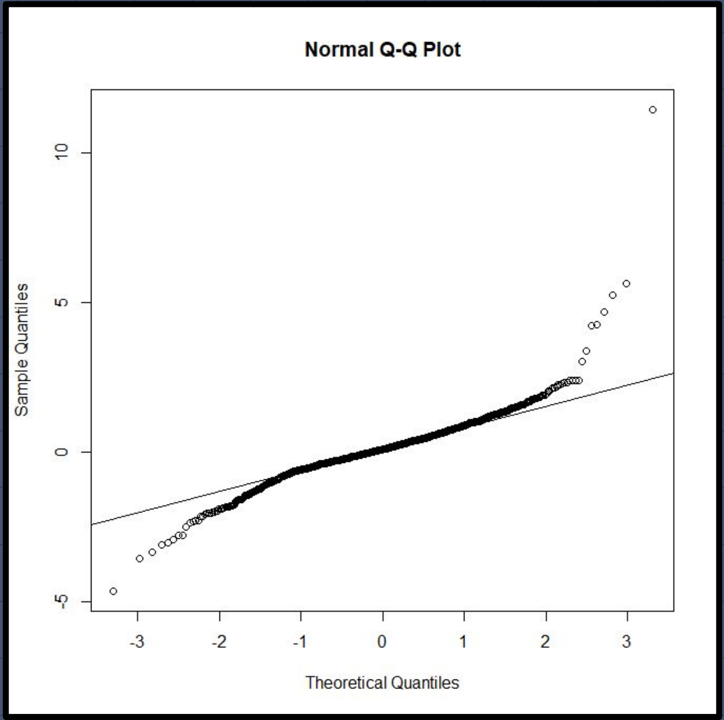
```
*----------------------------------*
*          ARFIMA Model Fit        *
*----------------------------------*
Mean Model        : ARFIMA(2,0,2)
Distribution      : norm

Optimal Parameters
------------------------------------
         Estimate  Std. Error     t value Pr(>|t|)
mu     309.285920    0.001663 1.8603e+05        0
ar1      0.000766    0.000001 5.1719e+02        0
ar2      0.979537    0.000048 2.0423e+04        0
ma1      1.108109    0.000078 1.4221e+04        0
ma2      0.101367    0.000000 1.3421e+07        0
sigma   22.973637    0.043979 5.2238e+02        0
```

ARFIMA Fitting



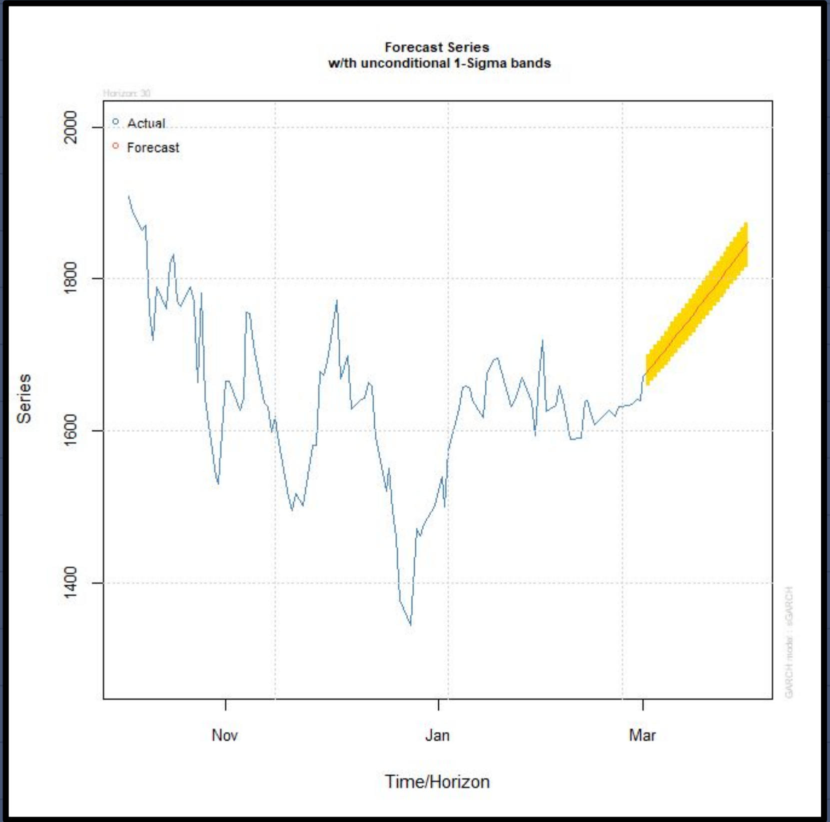
Conditional Volatility Graph





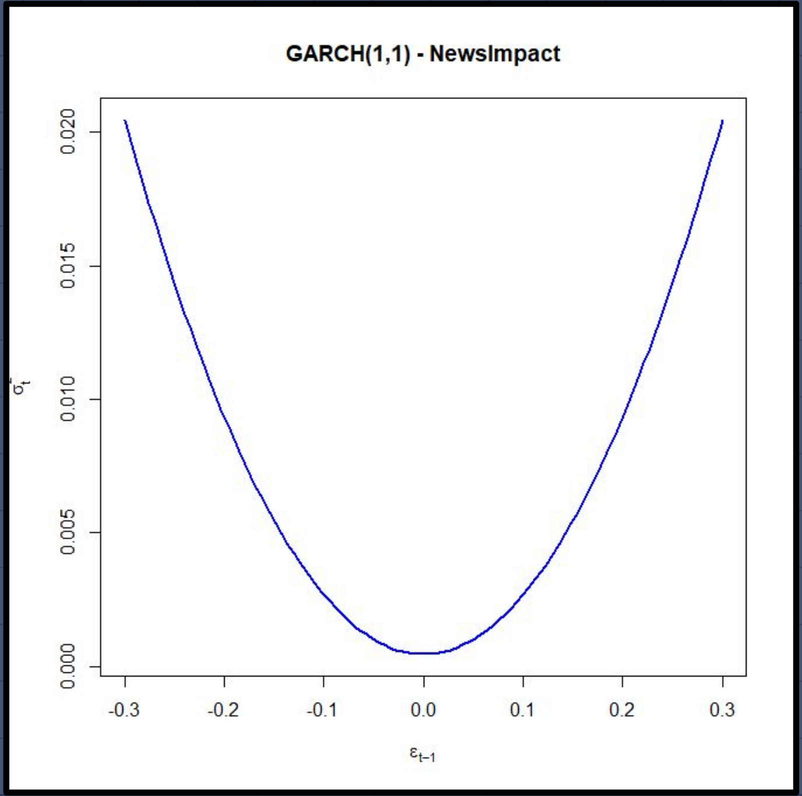
Normal Residual Plot

Normal QQ Plot



	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	1.909139	32.58867	19.05021	0.1383544	1.318285	0.9991381

MAPE = 1.3182



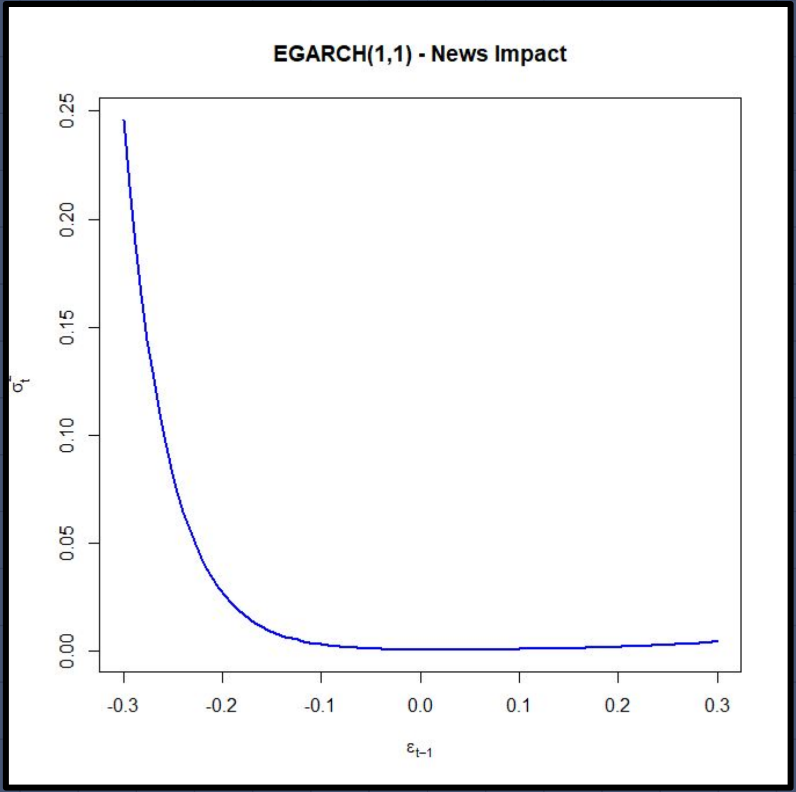
GARCH News Impact Plot

E-GARCH

- ARCH and GARCH do not capture one of the most important feature of data - leverage or asymmetric effect.
- This effect occurs when an unexpected drop in price (bad news), increases price volatility, more than unexpected increase in price (good news) of similar magnitude.
- This does away with the symmetry constraint on the conditional variance of the past value.
- The asymmetric effects can be captured by exponential GARCH or EGARCH.

$$\varepsilon_t = \sigma_t \eta_t$$

$$\log \sigma_t^2 = \omega + \sum_{i=1}^{q} (\alpha_i \eta_{t-i} + \gamma(|\eta_{t-i}| - E|\eta_{t-i}|)) + \sum_{j=1}^{q} \beta_j \log \sigma_{t-j}^2$$



E-GARCH News Impact Plot

T-GARCH

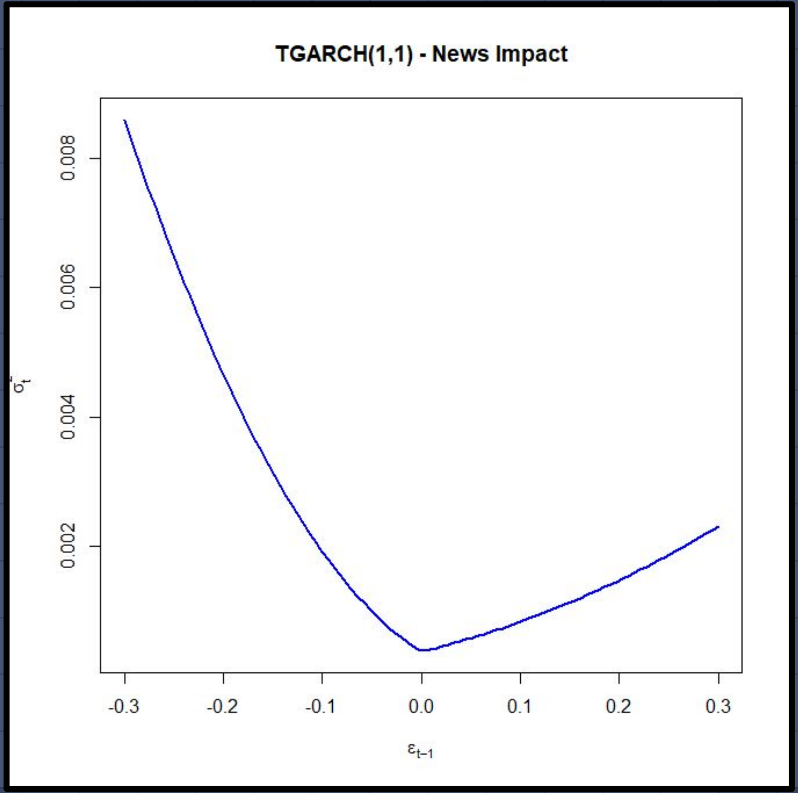
- The threshold GARCH or TGARCH is also used to deal with leverage effects.
- It involves an explicit distinction of model parameters above or below a certain threshold.

$$\varepsilon_t = \sigma_t \eta_t$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} (\alpha_i + \gamma_i I_{t-i}) \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

where

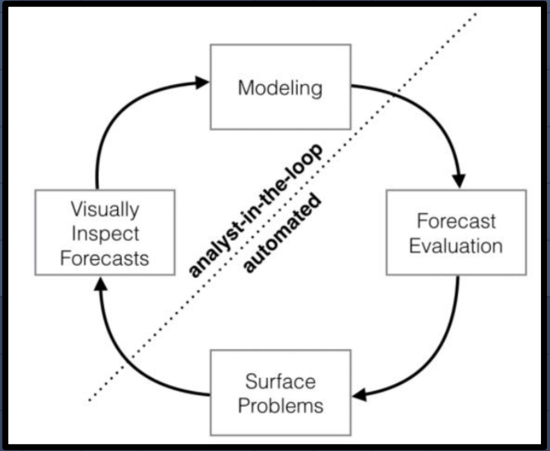
$$I_{t-i} = \begin{cases} 1 & \text{if } \varepsilon_{t-i} < 0 \\ 0 & \text{if } \varepsilon_{t-i} > 0 \end{cases}$$



T-GARCH News Impact Plot

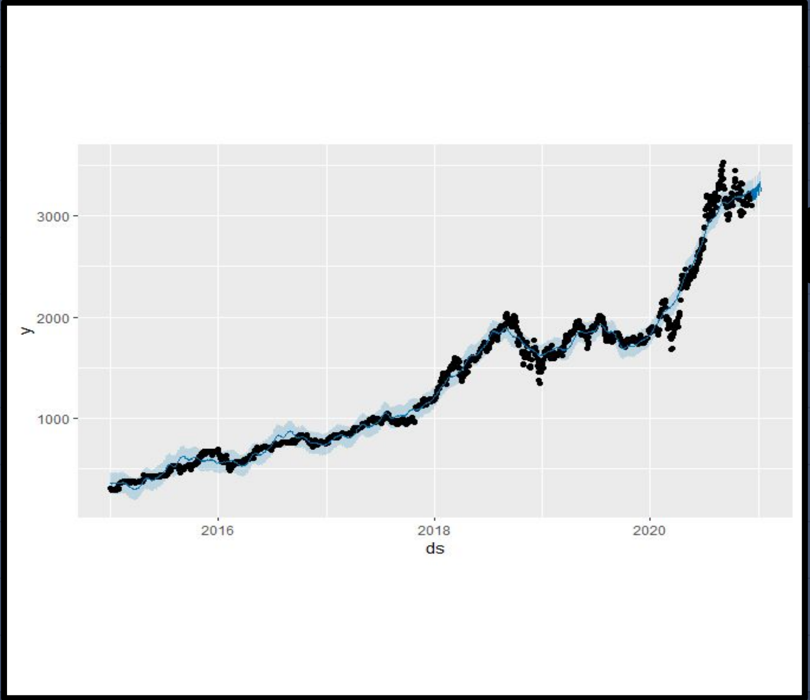
PROPHET MODELING

- Helps in shaping business decisions by following statistical approach.
- Developed by Facebook's Core Data Science team for business forecasting.
- Idea behind : By fitting the trend component very flexibly, we more accurately model seasonality.
- We use a very flexible regression model (like curve-fitting) instead traditional time series - gives us modeling flexibility, easier model fitting, gracefully handle missing data.



$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

PROPHET FORECASTING



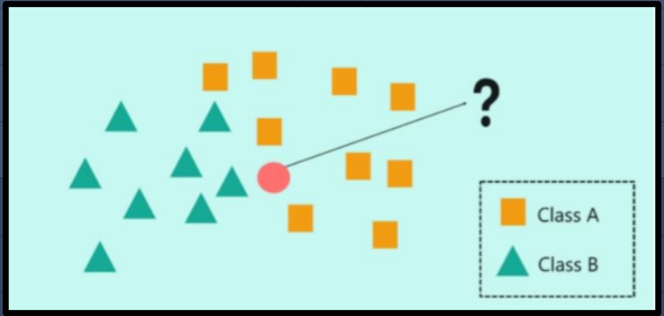
Prophet Forecast Plot

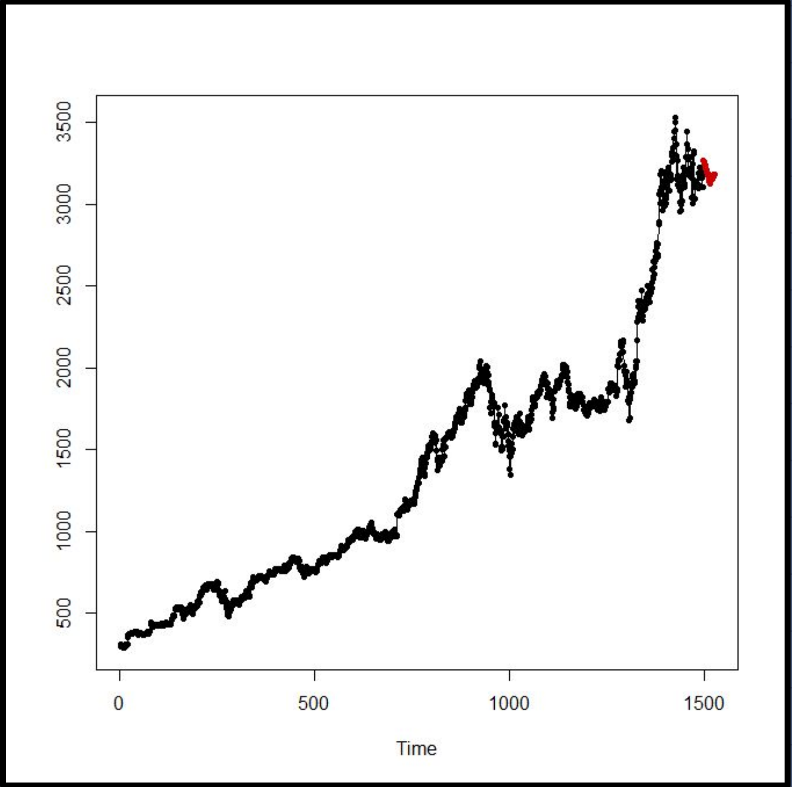
	ME	RMSE	MAE	MPE	MAPE
Test set	-0.003406316	82.43328	60.35623	-0.2890445	5.387829

MAPE = 5.3878

KNN MODELING

- Popular algorithm used in classification and regression problems.
- A collection of samples, each consisting of a vector of features and its associated class or numeric value, is stored.
- Given a new sample, KNN finds its k most similar examples (known as nearest neighbors) according to a distance metric and predicts its class according to the majority class of nearest neighbors.
- In regression, an aggregation of target values associated with its nearest neighbors is predicted.
- R package for KNN - tsfknn - used for univariate time series forecasting.





RMSE	MAE	MAPE
88.037987	76.574837	2.447051

MAPE = 2.4470

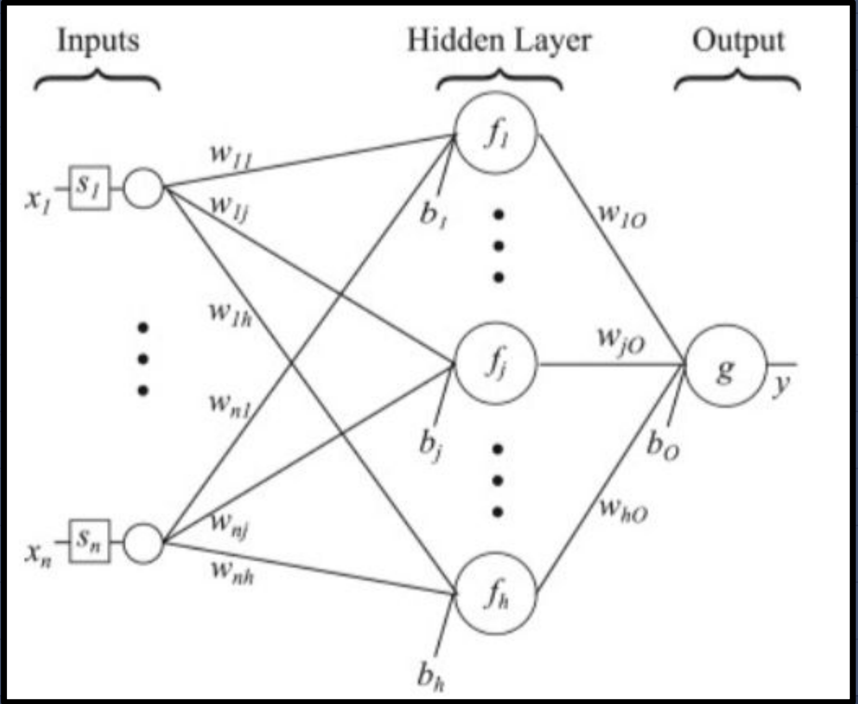
KNN Forecast Plot

NEURAL NETWORK MODELING

- A neural network can be defined as a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.

- Neural Networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer', which is the final output.

The nnetar function in the forecast package fits a single hidden layer neural network model to a timeseries. The function model approach is to use lagged values of the time series as input data, reaching to a non-linear autoregressive model.



Single Layer Neural Network

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))}$$

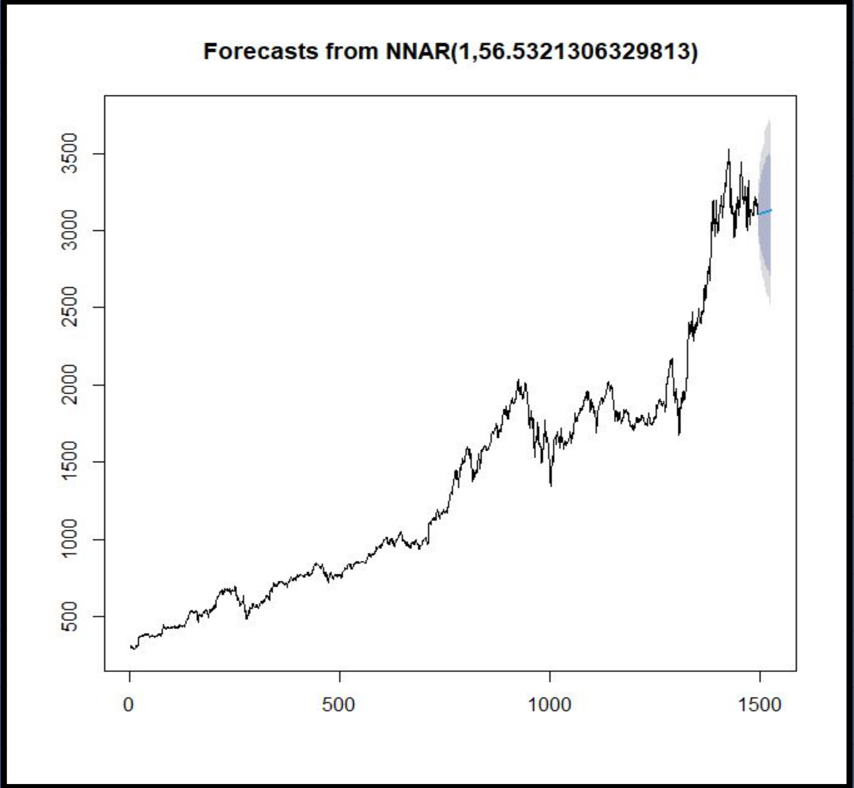
$$N_i = number.of.input.neurons.$$

$$N_o = number.of.output.neurons.$$

$$N_s = number.of.train.samples$$

$$\alpha = 1.5^{-10}$$

Parameters



	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.3412222	32.3966	18.90085	-0.013197	1.31211	0.9912993	-0.06934085

MAPE = 1.3121

Neural Network Forecast

ETS MODELING

- ETS (Error, Trend, Seasonal) method is an approach used for forecasting univariate time series.
- It focuses on the trend and seasonality in the data and thus defines how these unobserved components (error, trend and seasonality) change over time.
- The flexibility of the ETS model lies in its ability to capture trend and seasonal components of different traits.
- By considering variations in the combinations of the trend and seasonal components, nine exponential smoothing methods are possible.
- Each method is labelled by a pair of letters (T,S) defining the type of ‘Trend’ and ‘Seasonal’ components.

Trend Component	Seasonal Component		
	N	A	M
	(None)	(Additive)	(Multiplicative)
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)

- For each method there exist two models: one with additive errors and one with multiplicative errors.
- To distinguish between a model with additive errors and one with multiplicative errors (and also to distinguish the models from the methods), we add a third letter to the classification.
- The possibilities for each component are: Error = {A,M} , Trend = {N,A,Ad} and Seasonal={N,A,M}.

- A great advantage of the ETS statistical framework is that information criteria can be used for model selection.
- The AIC, AICc and BIC can be used here to determine which of the ETS models is most appropriate for a given time series.

$$\mathrm{AIC} = -2\log(L) + 2k$$

Akaike Information Criterion

L - likelihood of the model
k - total number of parameters and initial states

$$\mathrm{AIC_c} = \mathrm{AIC} + \frac{2k(k+1)}{T-k-1}$$

Akaike Information Criterion Corrected

$$\frac{2k(k+1)}{T-k-1}$$

Bias correction

$$\mathrm{BIC} = \mathrm{AIC} + k[\log(T) - 2]$$

Bayesian Information Criteria

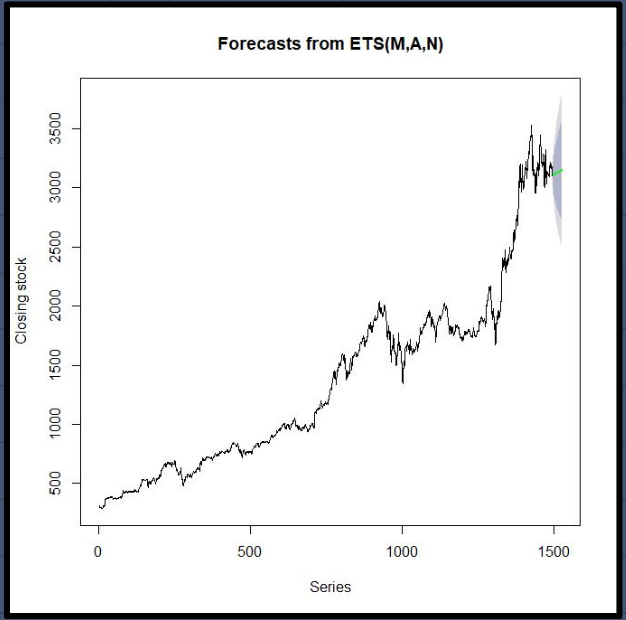
T - Number of Observations

ETS FORECASTING

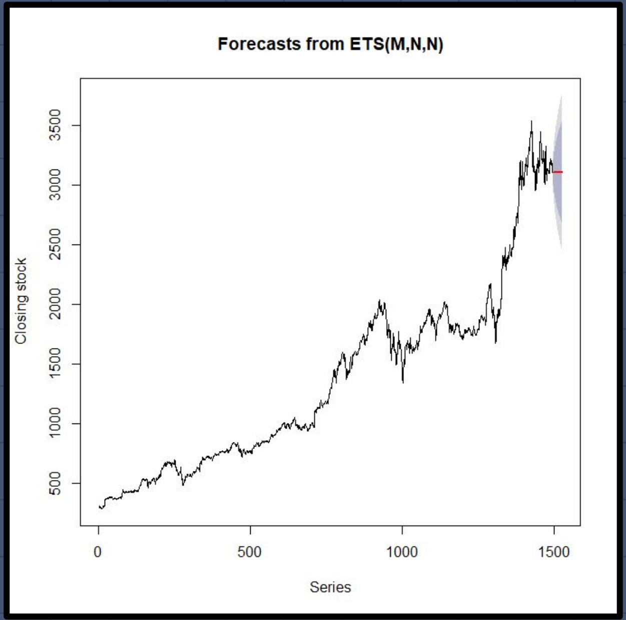
```
> auto.amzn.aic = ets(AMZN$Close,model="ZZZ",ic="aic")
> auto.amzn.aic$method
[1] "ETS(M,A,N)"
> auto.amzn.bic = ets(AMZN$Close,model="ZZZ",ic="bic")
> auto.amzn.bic$method
[1] "ETS(M,N,N)"
> auto.amzn.aic.damped = ets(AMZN$Close,model="ZZZ",damped = TRUE, ic="aic")
> auto.amzn.aic.damped$method
[1] "ETS(M,Ad,N)"
> auto.amzn.bic.damped = ets(AMZN$Close,model="ZZZ",damped = TRUE, ic="bic")
> auto.amzn.bic.damped$method
[1] "ETS(M,Ad,N)"
```

ETS package in R determines the optimum model by itself taking AIC, AICc and BIC into consideration.

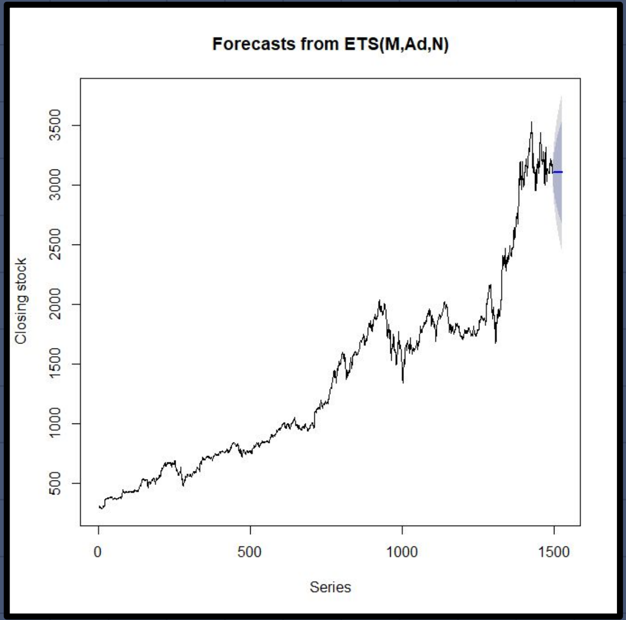
Three optimum models that can be used for this dataset are - ETS(M,A,N), ETS(M,Ad,N) and ETS(M,N,N)



ETS(M, A, N)



ETS(M, N, N)



ETS(M, Ad, N)

ETS(M, A, N)

```
Training set error measures:
                    ME     RMSE      MAE          MPE     MAPE      MASE
Training set 0.5463344 32.52234 18.97161 -0.004824417 1.316074 0.9950157
                    ACF1
Training set -0.03296453
```

MAPE = 1.3160

ETS(M, N, N)

```
Training set error measures:
                   ME     RMSE      MAE       MPE     MAPE      MASE
Training set 1.909139 32.58867 19.05021 0.1383544 1.318285 0.9991381
                    ACF1
Training set -0.04575169
```

MAPE = 1.3183

ETS(M, Ad, N)

```
Training set error measures:
                   ME     RMSE      MAE       MPE     MAPE     MASE       ACF1
Training set 1.816439 32.58462 19.04411 0.1281414 1.318435 0.998818 -0.0446543
```

MAPE = 1.3184

Best Model - ETS(M,A,N)

RESULTS

Model	MAPE
ARIMA	1.3125
GARCH	1.3182
PROPHET	5.3878
KNN	2.4470
NEURAL NETWORKS (BEST)	1.3121
ETS	1.3161

THANK YOU