Mobile Vendor Market Share Prediction

5/25/2021

George Bi, Nick Lesh, Akhil Ranjan, Justin Sun

Agenda

1. Business statement
2. EDA & base models
3. Model performance (Exponential Smoothing, ARIMA, ARFIMA, Regression with ARIMA errors)
4. Model selection
5. Future work and conclusion

Business Statement

3.80Billion
smartphone users in the world today

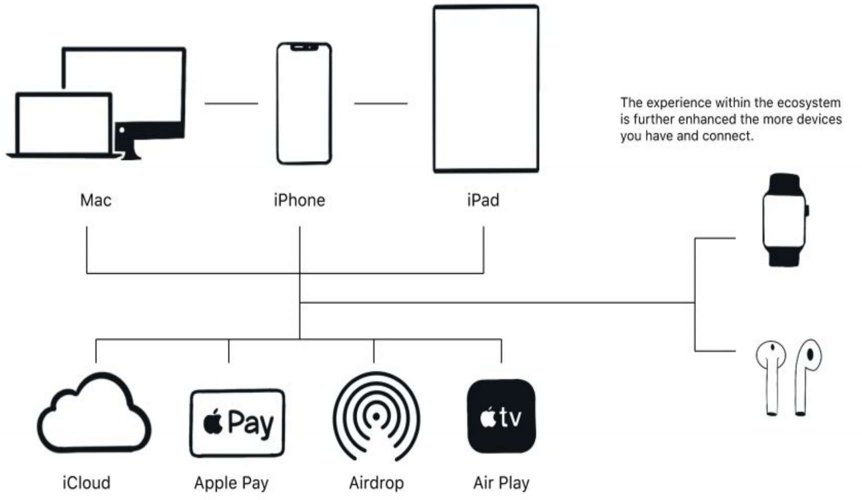
48.33%
of people have smartphones today

4.88Billion
mobile phone users in the world today

62.07%
of people own mobile phones today

By 2025, 72% of all internet users will solely use smartphones to access the web

Ecosystem



They are connected and kept updated through Apple's software.

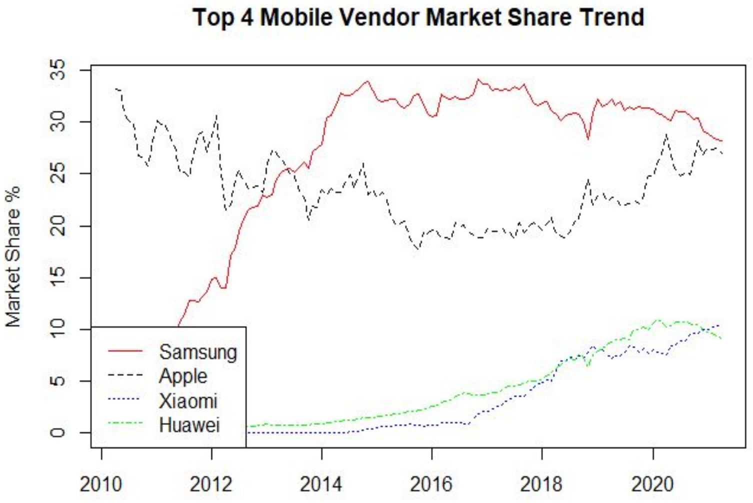
EDA & Base Model

Data:

- Mobile vendor market share data by month from April 2010 to April 2021 - selected top 4 vendors
- Test (April 2010 - April 2020) and train (May 2020 - April 2021) split

Base models:

- For comparison purposes, created the following models for each phone manufacturer:
  - Drift
  - Naive
  - Seasonal Naive
  - Mean
- Of the base models, Naive and Drift performed the best, which we will use for comparison purposes



> driftdf

	Phone	Drift_MAPE	Drift_RMSE	Drift_MAE
1	Samsung	0.07482148	2.799140	2.1642778
2	Apple	0.08026869	2.383487	2.0730000
3	Huawai	0.09283719	1.218398	0.8925773
4	Xaomi	0.14668943	1.482076	1.4234929

> naivedf

	Phone	Naive_MAPE	Naive_RMSE	Naive_MAE
1	Samsung	0.03258067	1.2188588	0.945000
2	Apple	0.08907264	2.5537048	2.309167
3	Huawai	0.04935184	0.5702046	0.485000
4	Xaomi	0.19985332	2.0638677	1.948333

Exponential Smoothing

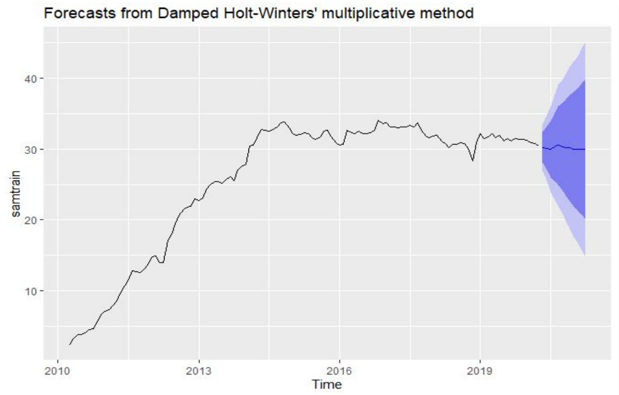
Competing Models

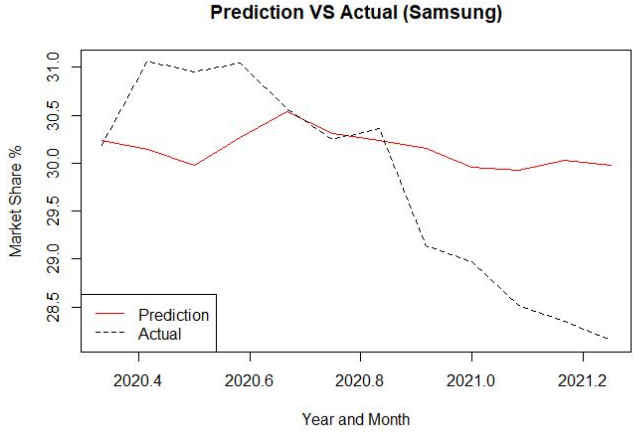
- Holt-Winters(HW) forecasts with additive seasonality and box-cox transformation - Additive
- Holt-Winters forecasts with multiplicative seasonality - Multiplicative
- Holt-Winters forecasts with additive seasonality, box-cox transformation, and damping - Damped Additive
- Holt-Winters forecasts with multiplicative seasonality and damping - Damped Multiplicative
- Results from the ETS() function in R forecast package - ETS

Samsung

	Additive	Multiplicative	Damped Additive	Damped Multiplicative	ETS
ME	-0.0968018	-0.1072325	0.1076610	0.1044767	-0.0322397
RMSE	0.8296303	0.7627181	0.8344323	0.7221548	0.7109405
MAE	0.6361524	0.5766414	0.6363024	0.5046448	0.5074476
MPE	-0.2056028	-0.8724200	1.0416348	0.7301974	-0.9816462
MAPE	3.5382349	3.1102522	3.2295828	2.6988076	3.0333246
MASE	0.1681718	0.1524396	0.1682115	0.1334068	0.1341477
ACF1	0.3160245	0.0698938	0.4397601	0.0715306	0.0906201

RMSE	MAPE
1.023161	0.0281012

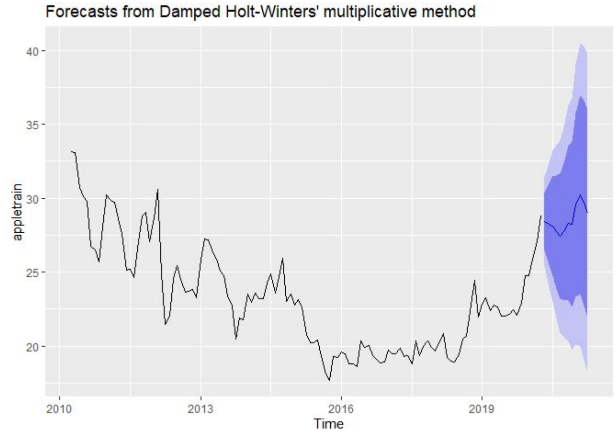


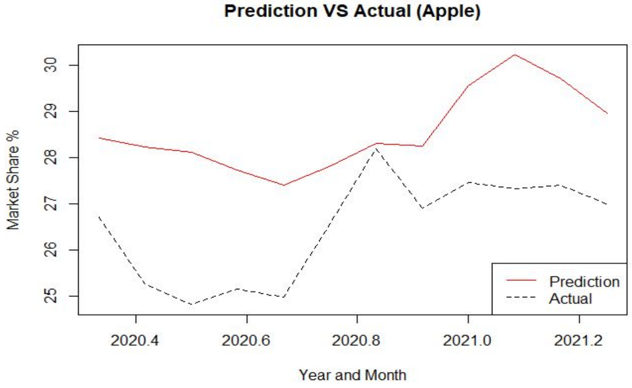


Apple

	Additive	Multiplicative	Damped Additive	Damped Multiplicative	ETS
ME	0.1640988	0.2887990	0.0944054	0.0914446	-0.0360656
RMSE	1.6574629	1.2723657	1.6504858	1.2002414	1.2756479
MAE	1.2202692	0.9250721	1.2858457	0.8674287	0.9089614
MPE	0.4497848	1.0382610	0.1486160	0.2719057	-0.2536703
MAPE	5.1466273	3.8924043	5.3976419	3.6776473	3.8487030
MASE	0.5216871	0.3954850	0.5497222	0.3708414	0.3885974
ACF1	0.6320179	0.1386586	0.4946336	0.0270804	0.0804956

RMSE	MAPE
2.241231	0.0794261

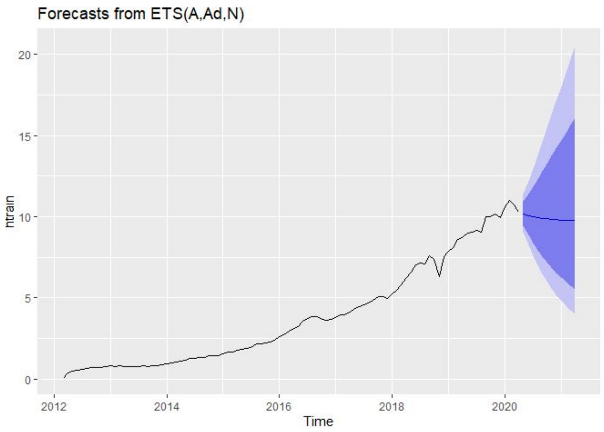


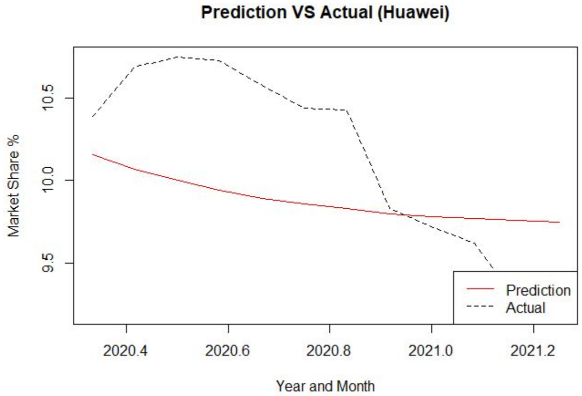


Huawei

	Additive	Multiplicative	Damped Additive	Damped Multiplicative	ETS
ME	-0.0026693	0.0533919	0.0816406	0.0454681	0.0236444
RMSE	0.3304171	0.2401211	0.2901033	0.2283531	0.2672924
MAE	0.2109366	0.1778509	0.1807883	0.1555006	0.1425588
MPE	0.3945302	-3.1026326	1.0926761	-2.2176920	-0.3528384
MAPE	7.7634540	12.3712676	8.1070371	10.5926478	5.0299007
MASE	0.1624769	0.1369922	0.1392548	0.1197765	0.1098079
ACF1	0.0999539	0.4504938	0.1194950	0.3123769	-0.2278620

RMSE	MAPE
0.5304951	0.0450366

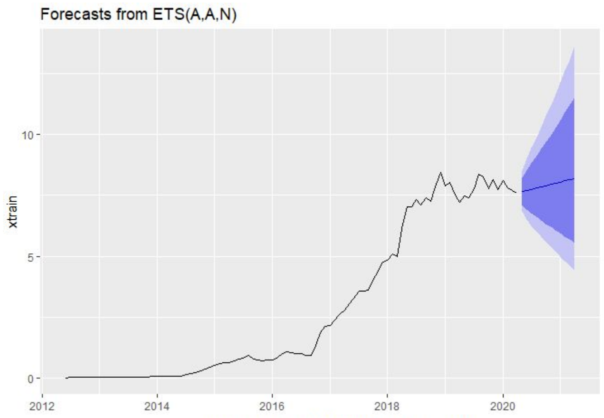


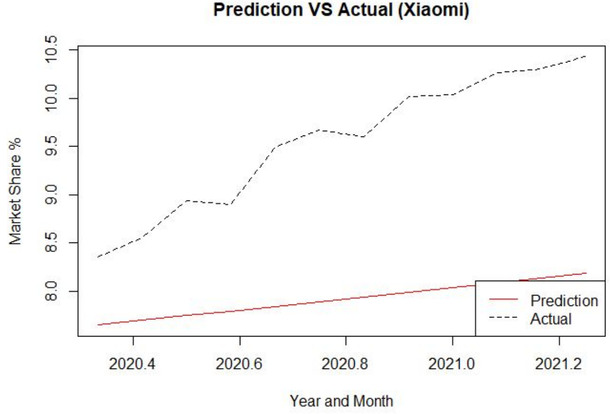


Xiaomi

	Additive	Multiplicative	Damped Additive	Damped Multiplicative	ETS
ME	-0.0320599	-0.0296376	-0.0054650	0.0204482	-0.0328288
RMSE	0.2480692	0.5827801	0.2386730	0.2612921	0.2590589
MAE	0.1479506	0.3605984	0.1355022	0.1659148	0.1527657
MPE	-0.8698816	2.8457707	2.5417860	-29.8512481	-1.4438212
MAPE	12.9650513	20.1121253	10.7331509	41.5312487	11.3110566
MASE	0.1283569	0.3128428	0.1175571	0.1439419	0.1325343
ACF1	0.2954034	0.8103795	0.1540126	0.3634125	0.0132195

RMSE	MAPE
1.710777	0.1676948

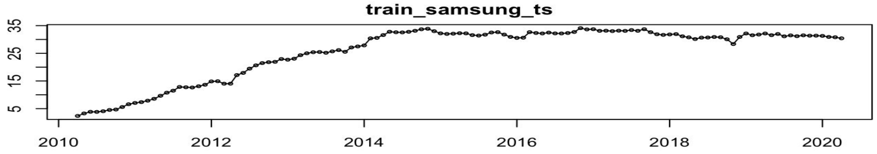




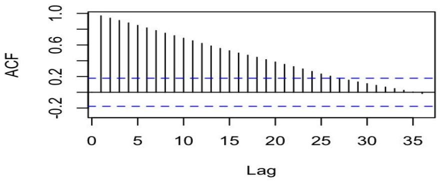
ARIMA MODELS STEPS:

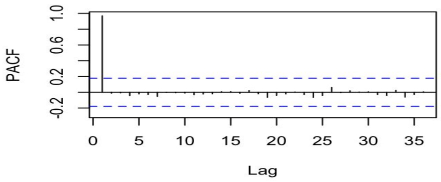
- Trend or Seasonal Pattern
- Box Cox Transformation: lambda
- First & Second Order Differencing
- Check Stationarity - KPSS test
- Fit Arima Models
- Check Residuals: Ljung-Box test
- Forecast
- Performance Measure

ARIMA - Samsung

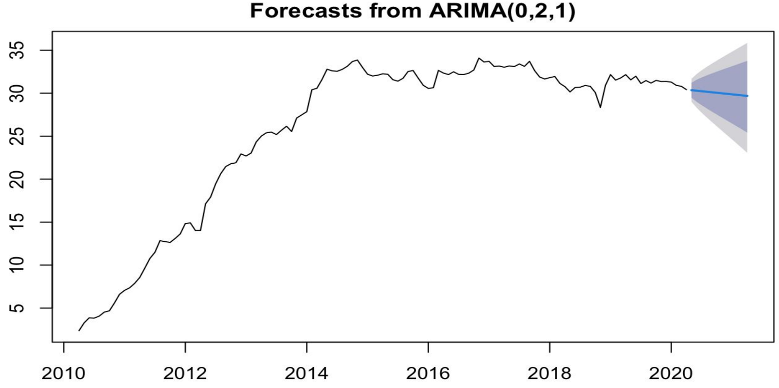


- As we can see , there is no seasonal pattern, but clearly there is a trend inside our dataset.
- The 2nd order difference of the data results in a stationary dataset. After box cox transformation, second order difference is stationary as well.
- ARIMA suggested ARIMA(0, 2, 1)
- White noise from Ljung-Box test






```
p-value greater than printed p-value
        KPSS Test for Level Stationarity

data:  samsung_diff2
KPSS Level = 0.022389, Truncation lag parameter = 4, p-value = 0.1
```

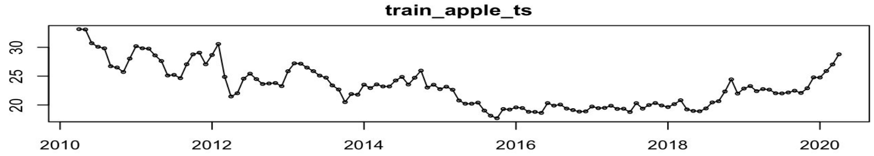


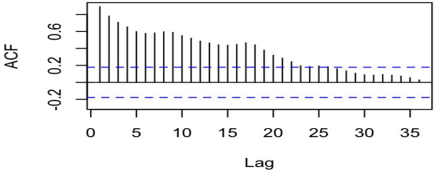

```
Series: train_samsung_ts
ARIMA(0,2,1)
Box Cox transformation: lambda= 1.328032

Coefficients:
          ma1
      -0.9365
s.e.   0.0317

sigma^2 estimated as 4.346:  log likelihood=-256.82
AIC=517.64   AICc=517.75   BIC=523.2

Training set error measures:
                    ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set -0.04085252 0.7069441 0.4991125 -0.1123275 2.358318 0.1319443 0.03256798
```

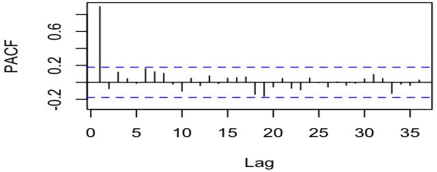
ARIMA - Apple

- As we can see , there is no seasonal pattern, but clearly there is a trend inside our dataset.
- The 2nd order difference of the data results in a stationary dataset. After box cox transformation, both first and second order difference is stationary.
- ARIMA suggested ARIMA(0, 1, 0)
- Ljung-Box test: P-value = 0.6212

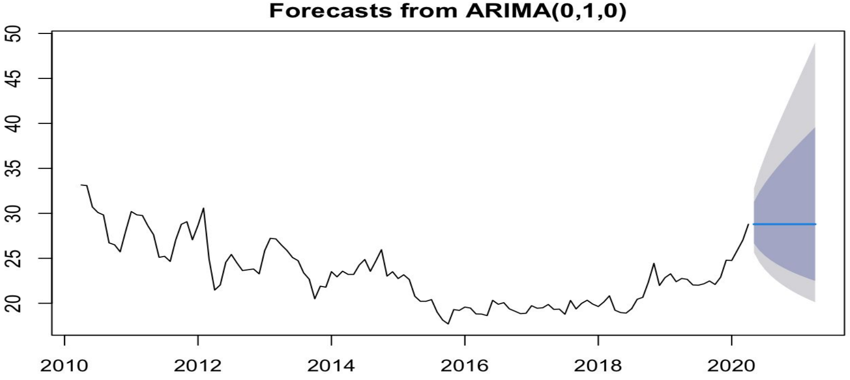
```
Series: train_apple_ts
ARIMA(0,1,0)
Box Cox transformation: lambda= -0.8999268

sigma^2 estimated as 9.166e-06:  log likelihood=525.79
AIC=-1049.58   AICc=-1049.55   BIC=-1046.8

Training set error measures:
                      ME     RMSE       MAE        MPE
Training set -0.02938531 1.277733 0.9155734 -0.2335089
```





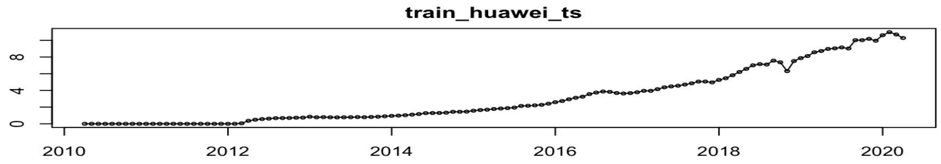


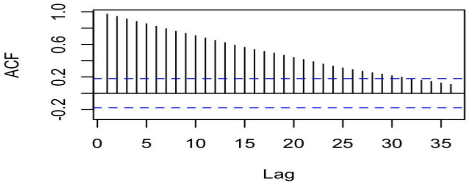

```
p-value greater than printed p-value
        KPSS Test for Level Stationarity

data:  apple_diff2
KPSS Level = 0.030629, Truncation lag parameter = 4, p-value = 0.1
```

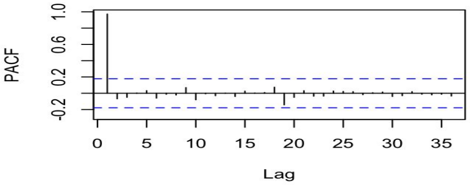


ARIMA - Huawei

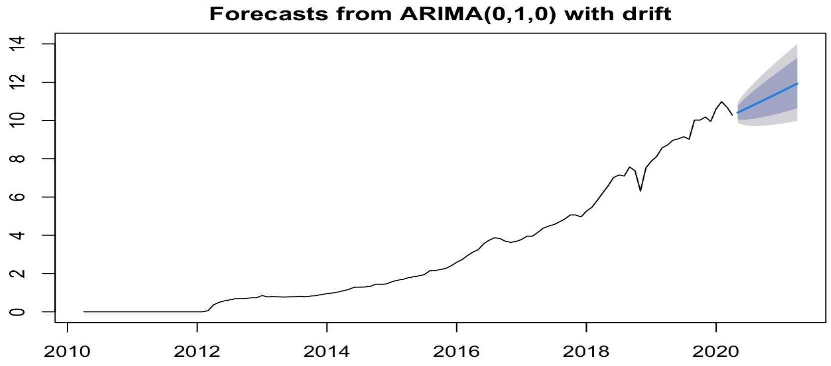
- As we can see , there is no seasonal pattern, but clearly there is a trend inside our dataset.
- The 2nd order difference of the data results in a stationary dataset. After box cox transformation, second order difference is stationary as well.
- ARIMA suggested ARIMA(0, 1, 0) with drift
- White noise from Ljung-Box test








```
p-value greater than printed p-value
          KPSS Test for Level Stationarity

data:  huawei_diff2
KPSS Level = 0.065137, Truncation lag parameter = 4, p-value = 0.1
```

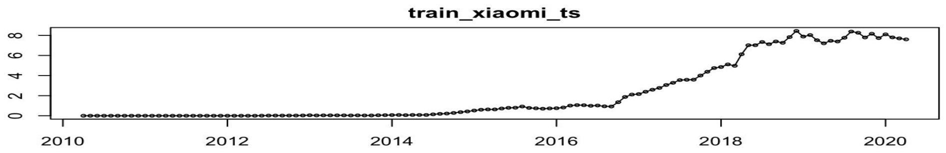


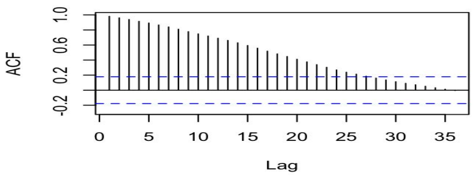

```
Series: train_huawei_ts
ARIMA(0,1,0) with drift
Box Cox transformation: lambda= 0.6416207

Coefficients:
       drift
      0.0579
s.e.  0.0111

sigma^2 estimated as 0.01488:  log likelihood=82.69
AIC=-161.38   AICc=-161.28   BIC=-155.81

Training set error measures:
                     ME      RMSE       MAE  MPE MAPE
Training set 0.01482531 0.2167977 0.1100728 -Inf  Inf
```

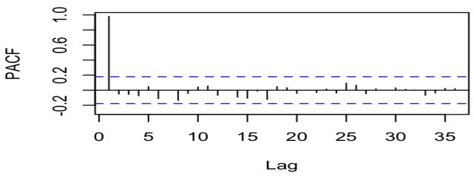
ARIMA - Xiaomi

- As we can see , there is no seasonal pattern, but clearly there is a trend inside our dataset.
- The 2nd order difference of the data results in a stationary dataset. After box cox transformation, both first and second order difference is stationary.
- ARIMA suggested ARIMA(0, 1, 0) with drift
- White noise from Ljung-Box test

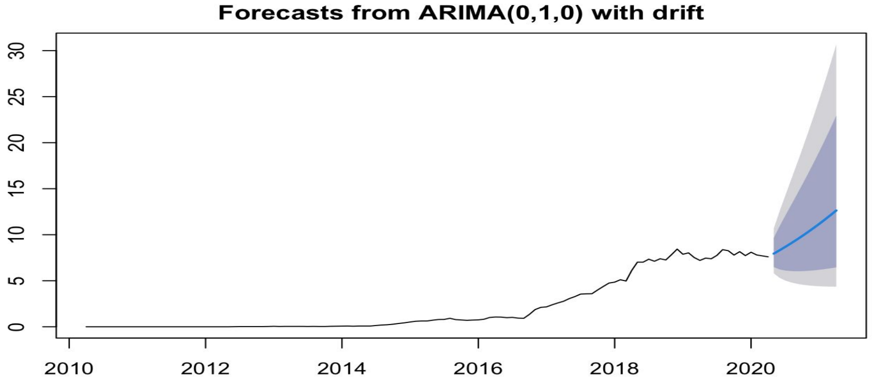
```
Series: train_xiaomi_ts
ARIMA(0,1,0) with drift
Box Cox transformation: lambda= 0.1874663

Coefficients:
       drift
      0.0650
s.e.  0.0208

sigma^2 estimated as 0.05228:  log likelihood=7.3
AIC=-10.6   AICc=-10.5   BIC=-5.03

Training set error measures:
                      ME      RMSE       MAE  MPE MAPE
Training set -0.04365456 0.250541 0.1272362 -Inf  Inf
```








```
p-value greater than printed p-value
        KPSS Test for Level Stationarity

data:  xiaomi_diff2
KPSS Level = 0.035336, Truncation lag parameter = 4, p-value = 0.1
```



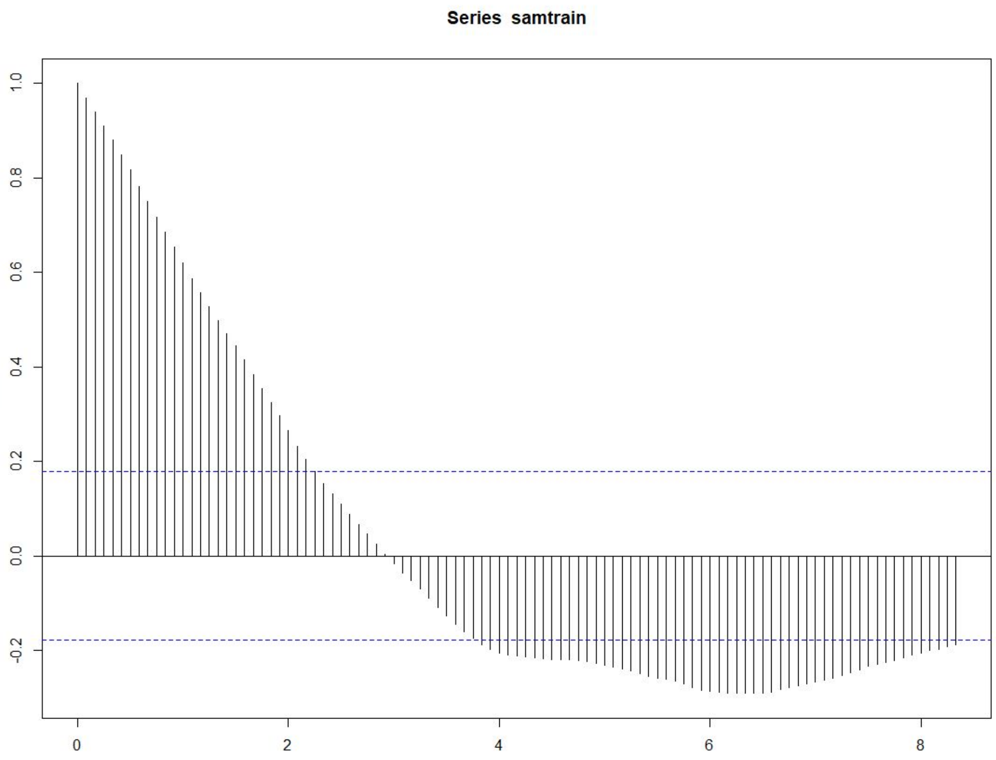
ARIMA - Final Results/Conclusions

- ARIMA models overall performed ok with this dataset.
- Apple and Xiaomi models performed better than Huawai and Samung models, with mean absolute percentage error around 7 and 8%. Huawei performed a little worse with 11% MAPE while Samsung's MAPE is very bad with 78% MAPE.

Phone <chr>	MSE <dbl>	RMSE <dbl>	MAE <dbl>	MAPE <dbl>
Samsung	0.8070207	0.8983433	0.7892179	0.78921790
Apple	6.5214083	2.5537048	2.3091667	0.08907264
Huawei	2.0596719	1.4351557	1.0625636	0.11038630
Xiaomi	1.0597953	1.0294636	0.7709275	0.07729281

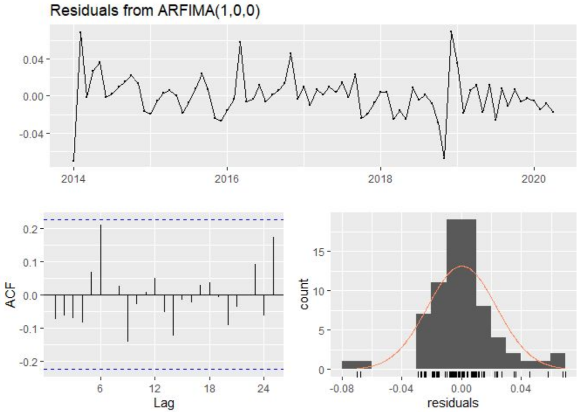
ARFIMA Models

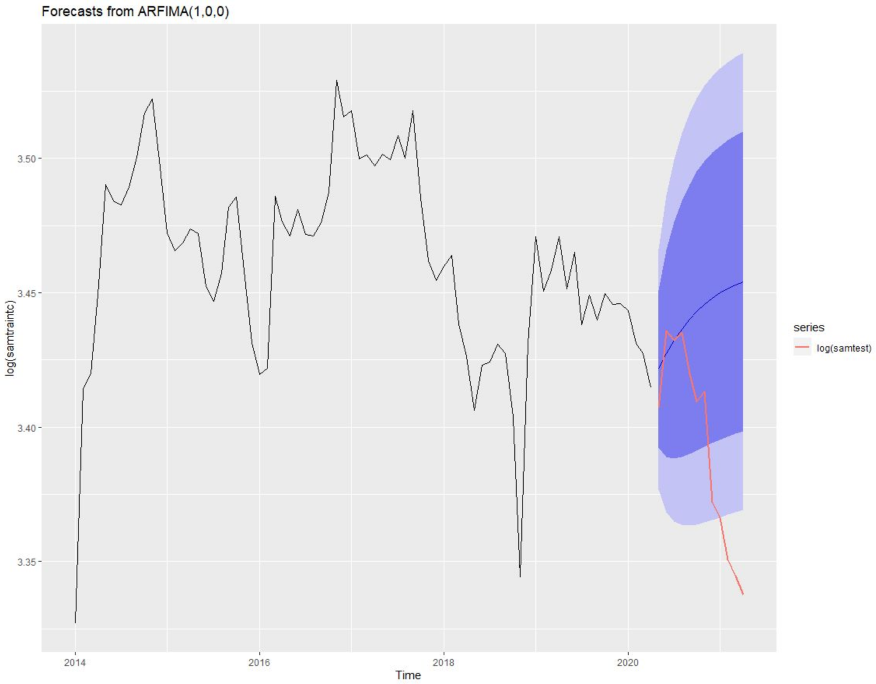
- Some datasets have ‘long memory’ where the ACF is decaying at a slow rate
- ARFIMA models solve for this issue by utilizing fractional differencing
- The d parameter is estimated in the ARFIMA model
  - d = 0, model exhibits short - term memory
  - 0 < d < 0.5, model exhibits long memory
  - d >= 0.5, process is nonstationary
- Fitted multiple ARFIMA models and chose model with lowest AIC score for forecasting purposes



ARFIMA - Samsung

- Samsung gained market share rapidly until 2014, and then market share started to stabilize
- In order to stabilize model, we truncated the dataset to only include 2014 and later data
- Used the logarithmic function to further stabilize the variance
- ARFIMA suggested a model with a (p,d,q) of (1,0,0)

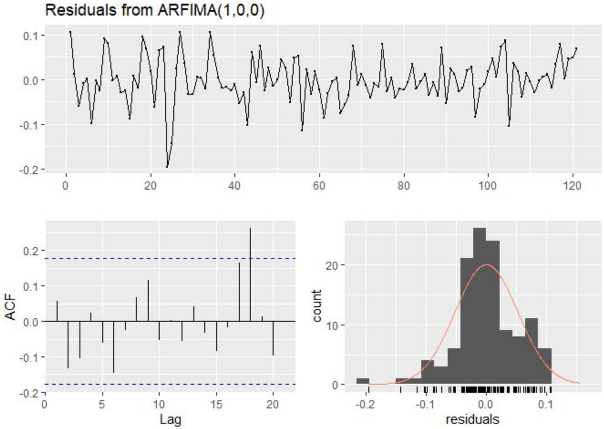


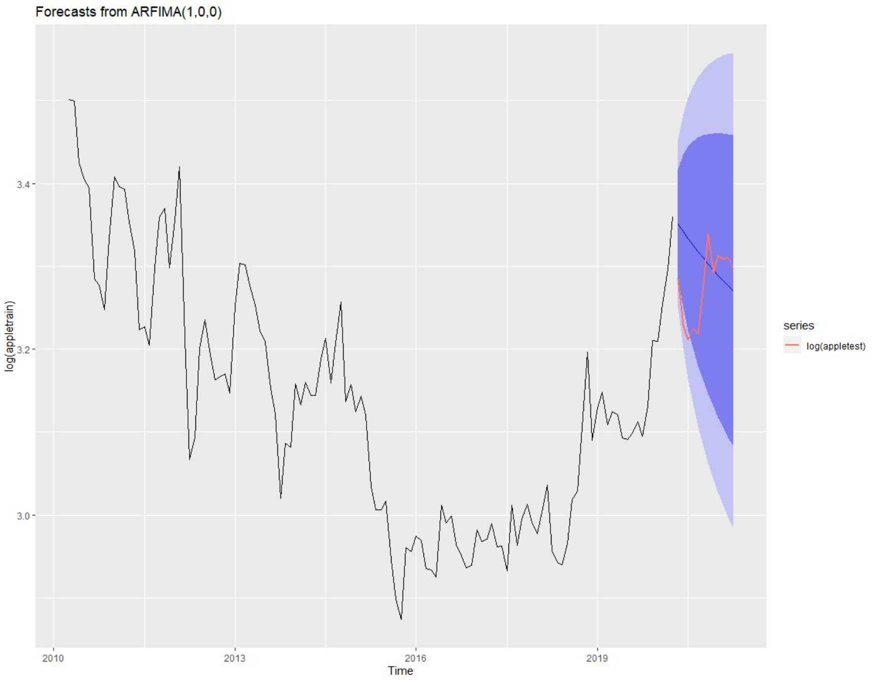



```
## Coefficients:
##          Estimate Std. Error z value Pr(>|z|)
## d      4.583e-05  8.007e-02   0.001        1
## ar.ar1 8.538e-01  2.287e-02  37.335   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## sigma[eps] = 0.02118419
## [d.tol = 0.0001221, M = 100, h = 1.961e-06]
## Log likelihood: 185.1 ==> AIC = -364.2112 [3 deg.freedom]
```

ARFIMA - Apple

- Apple started losing market share until 2016, but then started to regain market share
- Used the logarithmic function to further stabilize the variance
- ARFIMA suggested a model with a (p,q,d) of (1,0,0)

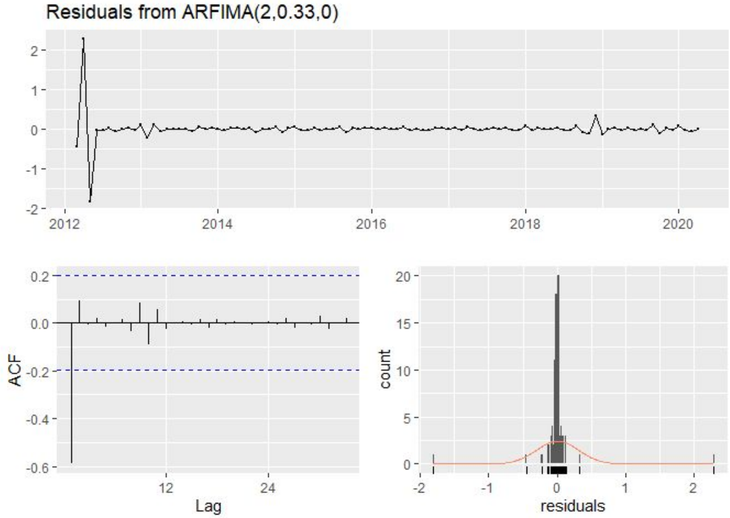


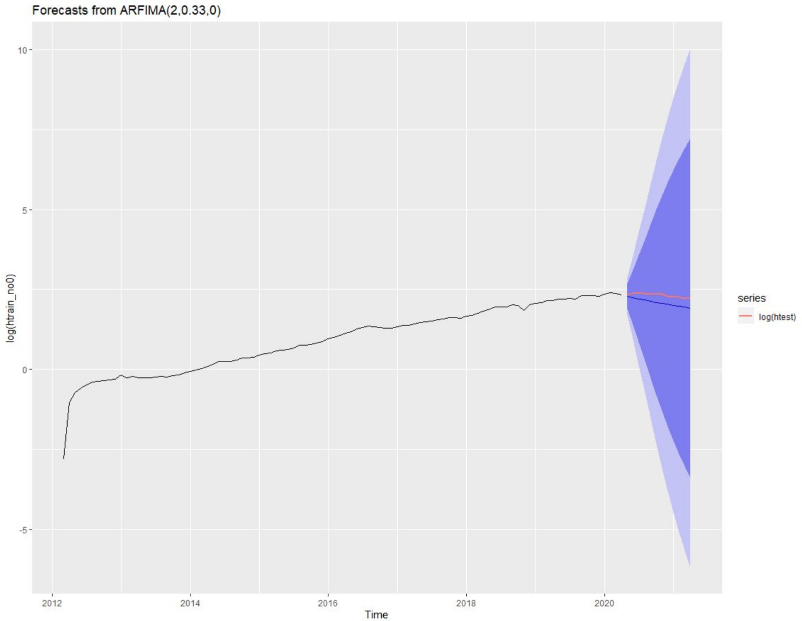



```
Call:
  arfima(y = log(appletrain))

*** Warning during (fdcov) fit: unable to compute correlation matrix; maybe change 'h'

Coefficients:
         Estimate
d           0.005
ar.ar1      0.958
sigma[eps] = 0.05117889
[d.tol = 0.0001221, M = 100, h = 1.991e-06]
Log likelihood:    188 ==> AIC = -369.9537 [3 deg.freedom]
```

ARFIMA - Huawei

- Huawei did not sell phones until March 2012
- Used the logarithmic function to further stabilize the variance
- ARFIMA suggested a model with a (p,d,q) of (2,0.33,0)

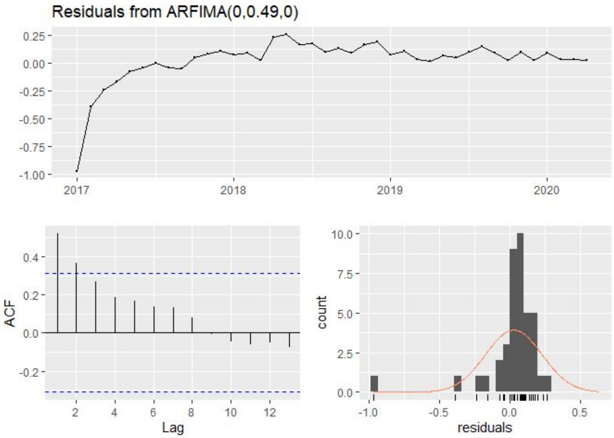


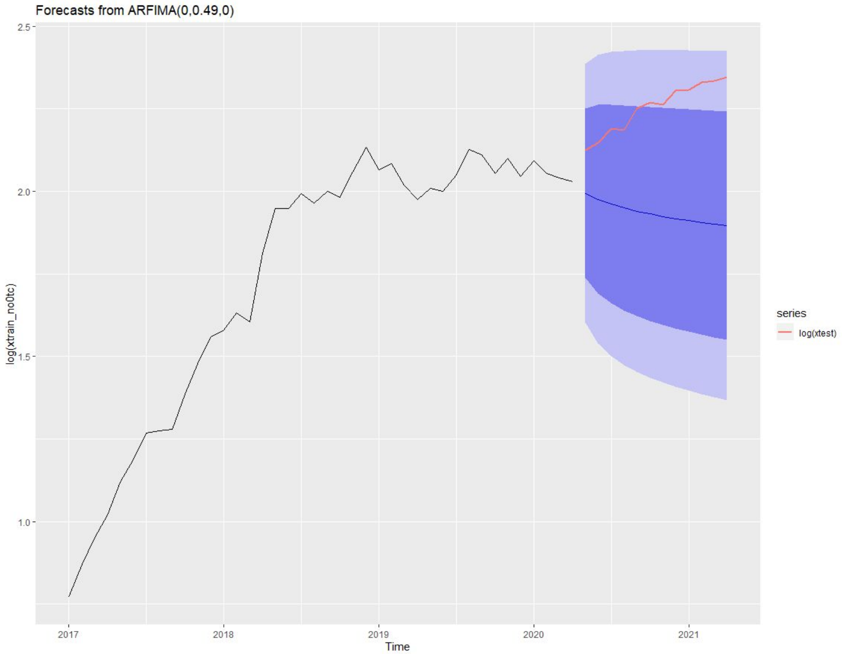



```
Call:
  arfima(y = log(htrain_no0))

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
d        0.3287859  0.0007392  444.77   <2e-16 ***
ar.ar1   1.6187528  0.0147895  109.45   <2e-16 ***
ar.ar2  -0.6387814  0.0202965  -31.47   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
sigma[eps] = 0.0458911
[d.tol = 0.0001221, M = 100, h = 1.723e-06]
Log likelihood: 162.5 ==> AIC = -316.9781 [4 deg.freedom]
```

ARFIMA - Xiaomi

- Xiaomi did not sell phones until June 2012
- Truncated the model further to only include data after January 2017 where growth stabilized
- Used the logarithmic function to further stabilize the variance
- ARFIMA suggested a model with a (p,d,q) of (0,0.49,0)






```
Cal
  arfima(y = log(xtrain_no0tc))

Coefficients:
   Estimate Std. Error z value Pr(>|z|)
d 4.936e-01  1.706e-07 2893222   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
sigma[eps] = 0.1565462
[d.tol = 0.0001221, M = 100, h = 1.717e-07]
Log likelihood: 15.29 ==> AIC = -26.58936 [2 deg.freedom]
```

ARFIMA - Final Results/Conclusions

- ARFIMA models overall performed poorly with this dataset.
- Apple and Samsung models performed better than Huawei and Xiaomi models, however Samsung and Apple models exhibited short memory with d value = 0
- Results and data plots indicate dataset is more suited for a standard ARIMA model
- Ljung Box Results
  - Samsung : 0.5139
  - Apple: 0.39014
  - Huawai: 0
  - Xiaomi: 0.001

ARFIMA Results

Phone	MAPE	RMSE	MAE
Samsung	0.0518846	1.963128	1.499242
Apple	0.0589439	1.842285	1.520067
Huawai	0.1975295	2.058276	1.982254
Xaomi	0.7961826	7.647637	7.614589

Naive Base Model Results

Phone	Naive_MAPE	Naive_RMSE	Naive_MAE
Samsung	0.0325807	1.2188588	0.945000
Apple	0.0890726	2.5537048	2.309167
Huawai	0.0493518	0.5702046	0.485000
Xaomi	0.1998533	2.0638677	1.948333

Regression with ARIMA errors

To forecast the regression model with ARIMA errors, we will be forecasting the two parts of the equation:

- The regression part of the model
- The ARIMA part of the model

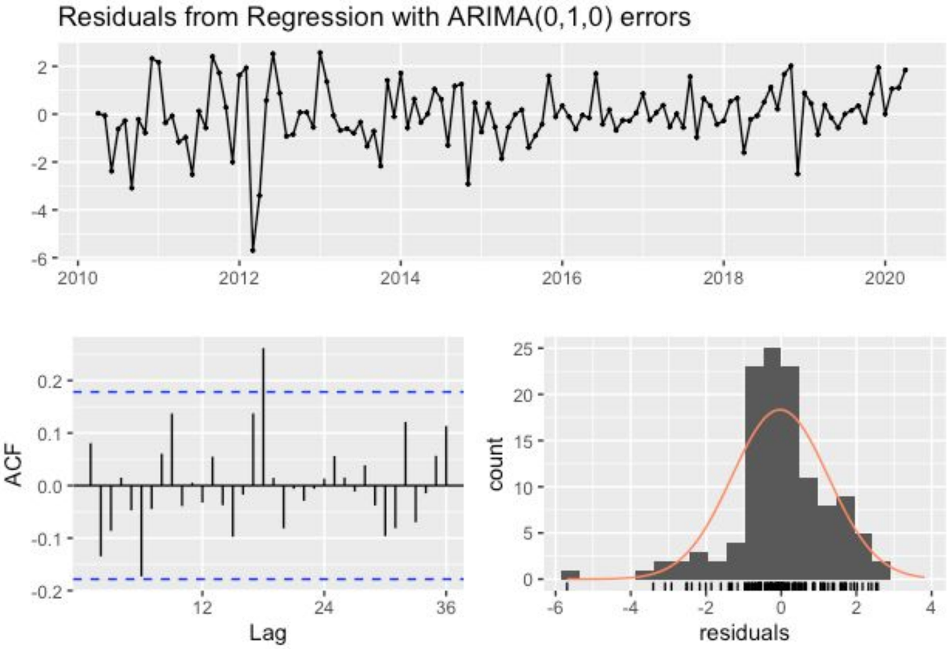
and combine the results.

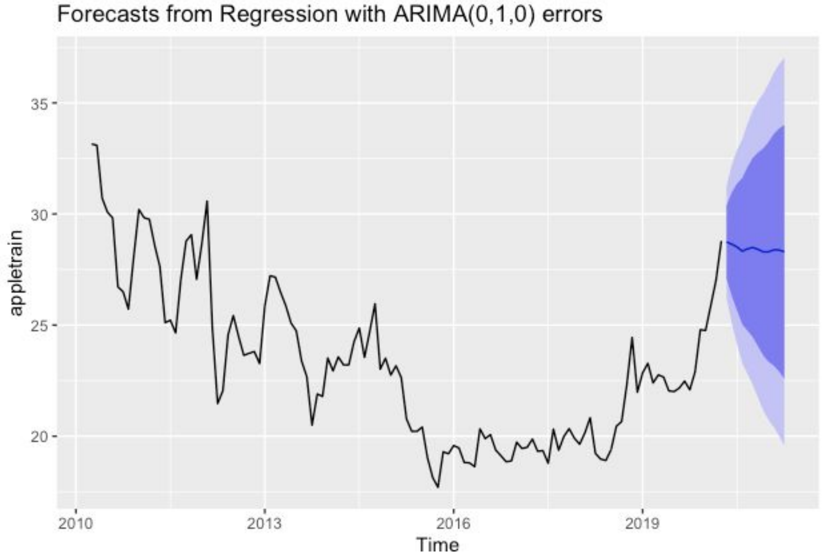
For xreg part of the model, we will be using the **interest over time from Google Trends** and **stocks** (Samsung and Apple)

Regression with ARIMA errors - Apple stocks

- In the ARIMA part of the model, we will be using Apple stocks as xreg
- Tried to fit ARIMA models:

Model	AICc
Regression with ARIMA(2,1,2) errors	413.6789
Regression with ARIMA(0,1,0) errors	406.06
Regression with ARIMA(1,1,0) errors	409.3955
Regression with ARIMA(0,1,1) errors	409.091
Regression with ARIMA(0,1,0) errors	404.0327





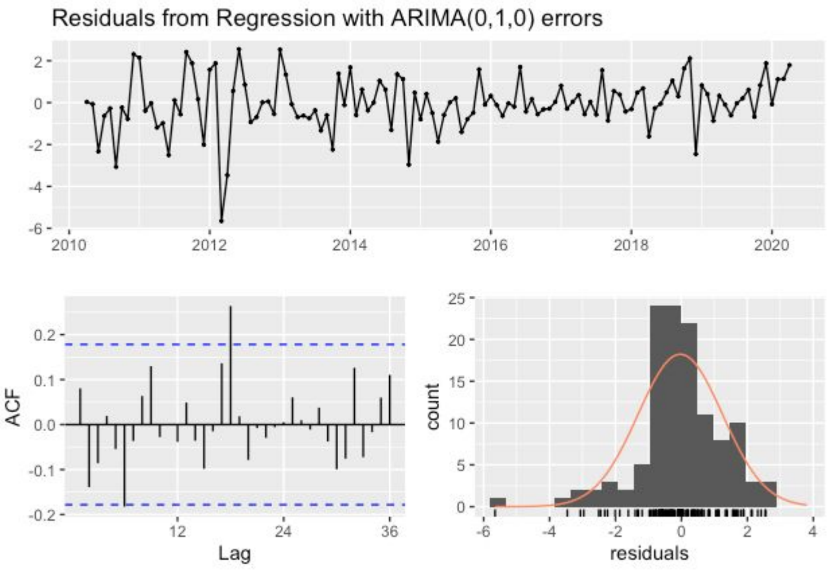

```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,0) errors
Q* = 27.584, df = 23, p-value = 0.2319

Model df: 1.   Total lags used: 24
```

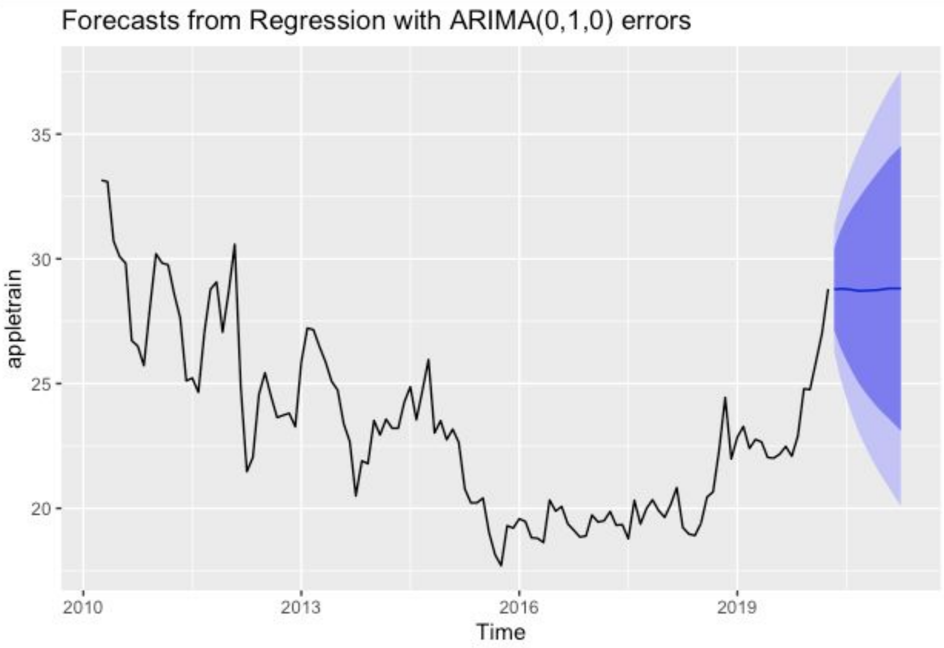
MAPE	MAE	RMSE
7.567 %	1.95313	2.26219

Regression with ARIMA errors - Apple Interest over time

- In the ARIMA part of the model, we will be using Apple's Interest over time from Google Trends as xreg
- Tried to fit ARIMA models:

Model	AICc
Regression with ARIMA(2,1,2) errors	413.3203
Regression with ARIMA(0,1,0) errors	405.8281
Regression with ARIMA(1,1,0) errors	409.1159
Regression with ARIMA(0,1,1) errors	408.7968
Regression with ARIMA(0,1,0) errors	403.8174





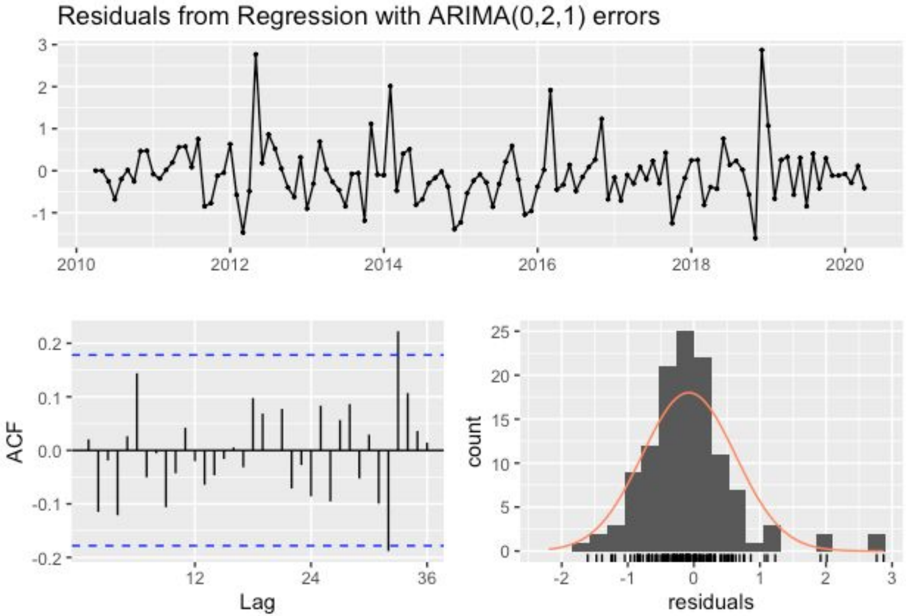

```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,0) errors
Q* = 27.878, df = 23, p-value = 0.2204

Model df: 1.   Total lags used: 24
```

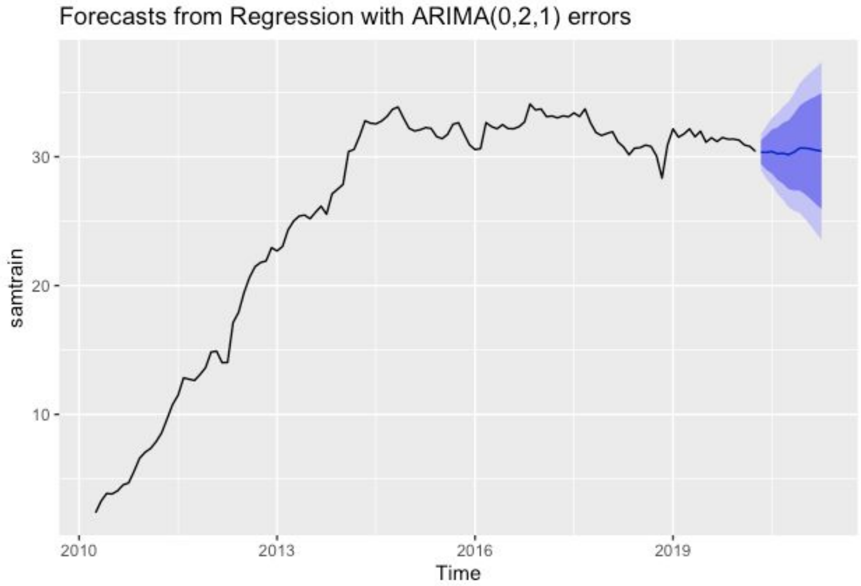
MAPE	MAE	RMSE
8.8332%	2.28981	2.53441

Regression with ARIMA errors - Samsung stocks

- In the ARIMA part of the model, we will be using Samsung stocks as xreg
- Tried to fit ARIMA models:

Model	AICc
Regression with ARIMA(1,2,2) errors	268.8425
Regression with ARIMA(0,2,0) errors	331.7893
Regression with ARIMA(1,2,0) errors	312.1221
Regression with ARIMA(0,2,1) errors	265.566
Regression with ARIMA(0,2,2) errors	267.4893





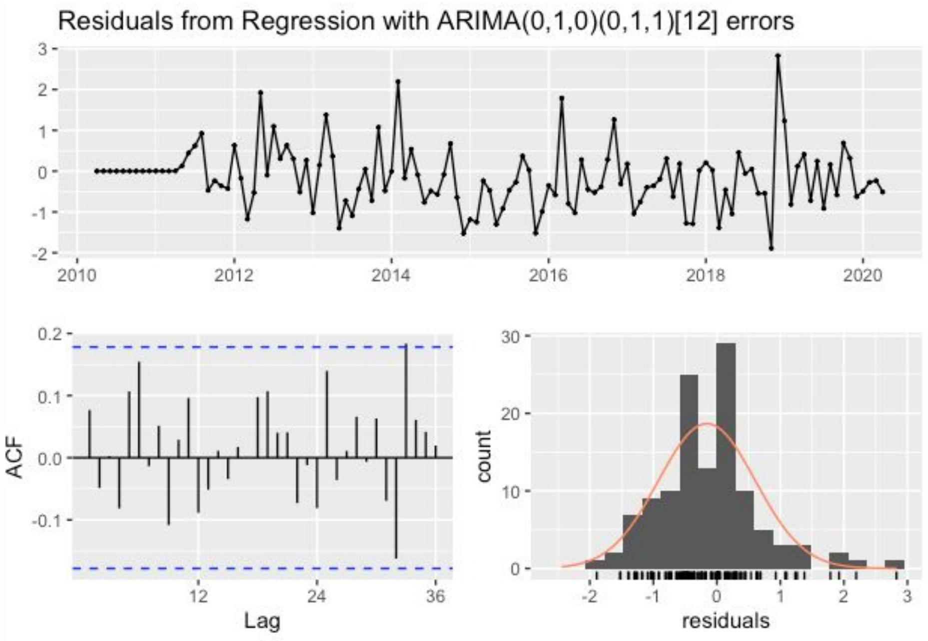

```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,2,1) errors
Q* = 14.827, df = 22, p-value = 0.8695

Model df: 2.   Total lags used: 24
```

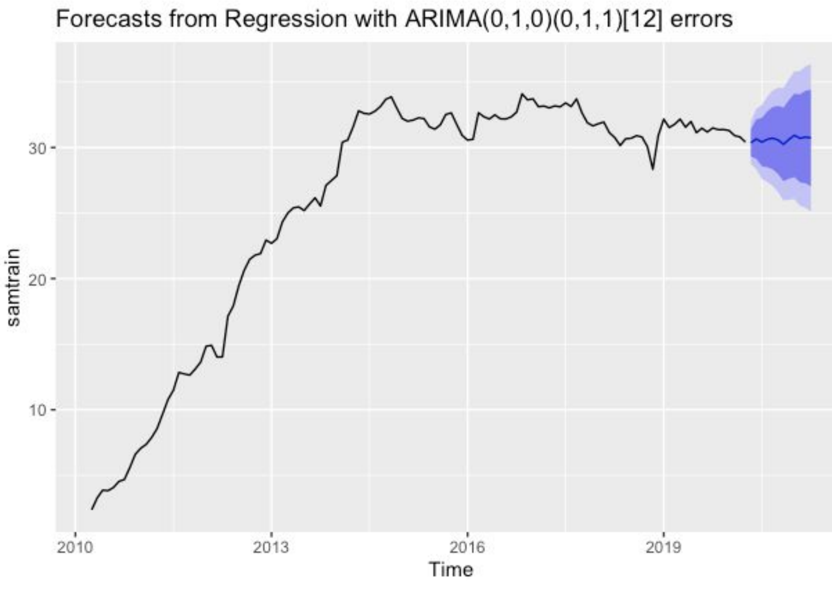
MAPE	MAE	RMSE
3.564 %	1.03488	1.32681

Regression with ARIMA errors - Samsung Interest over time

- In the ARIMA part of the model, we will be using Samsung's Interest over time from Google Trends as xreg
- Tried to fit ARIMA models:

Model	**AICc**
Regression with ARIMA((0,1,0)(1,1,0)	290.0806
Regression with ARIMA(0,1,0)(0,1,0)	313.7806
Regression with ARIMA(1,1,0)(1,1,0)	291.7907
Regression with ARIMA((0,1,1)(0,1,1)	278.477
Regression with ARIMA(0,1,0)(1,1,1)	280.0724





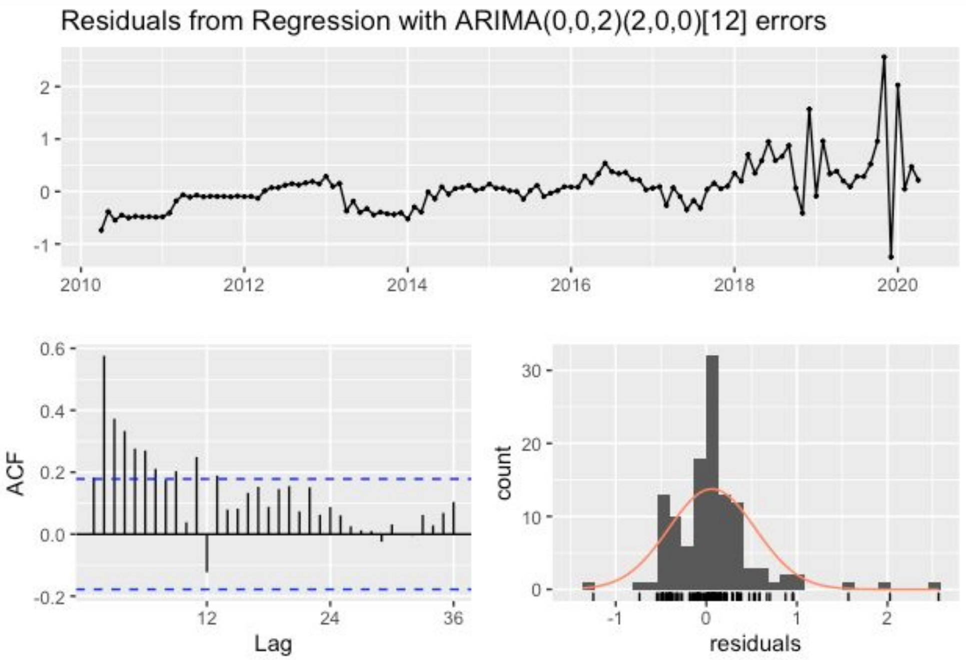

```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,0)(0,1,1)[12] errors
Q* = 16.789, df = 22, p-value = 0.7749

Model df: 2.   Total lags used: 24
```

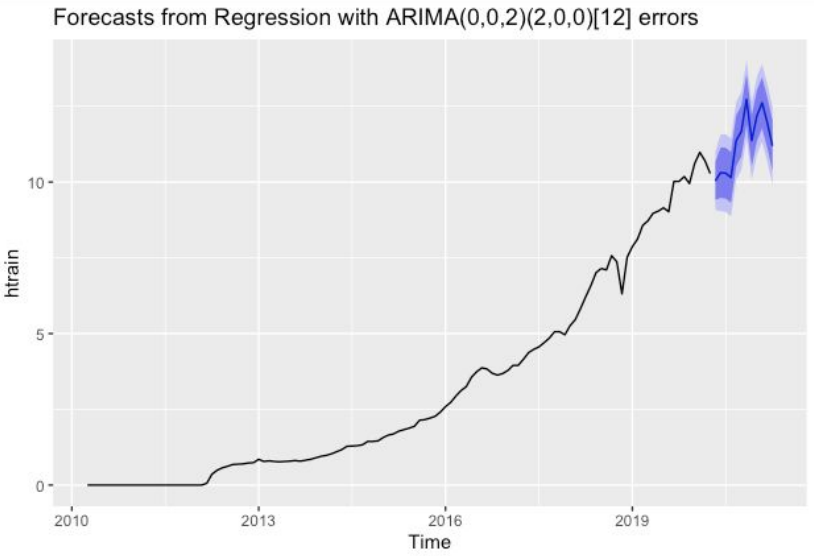
MAPE	MAE	RMSE
3.6820%	1.06413	1.41996

Regression with ARIMA errors - Huawei Interest over time

- In the ARIMA part of the model, we will be using Huawei's Interest over time from Google Trends as xreg
- Tried to fit ARIMA models:

Model	AICc
Regression with ARIMA(0,0,0)	377.818
Regression with ARIMA(0,0,1)(0,0,1)	286.6336
Regression with ARIMA(0,0,2) errors	255.6123
Regression with ARIMA(0,0,2)(2,0,0)	230.2831
Regression with ARIMA(0,0,1)(2,0,0)	285.9782





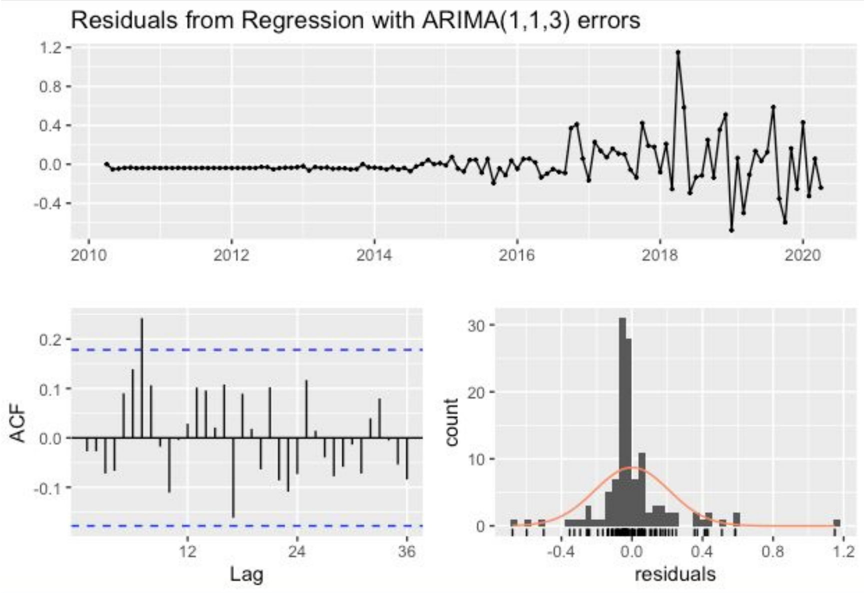

```
	Ljung-Box test

data:  Residuals from Regression with ARIMA(0,0,2)(2,0,0)[12] errors
Q* = 149.1, df = 18, p-value < 2.2e-16

Model df: 6.   Total lags used: 24
```

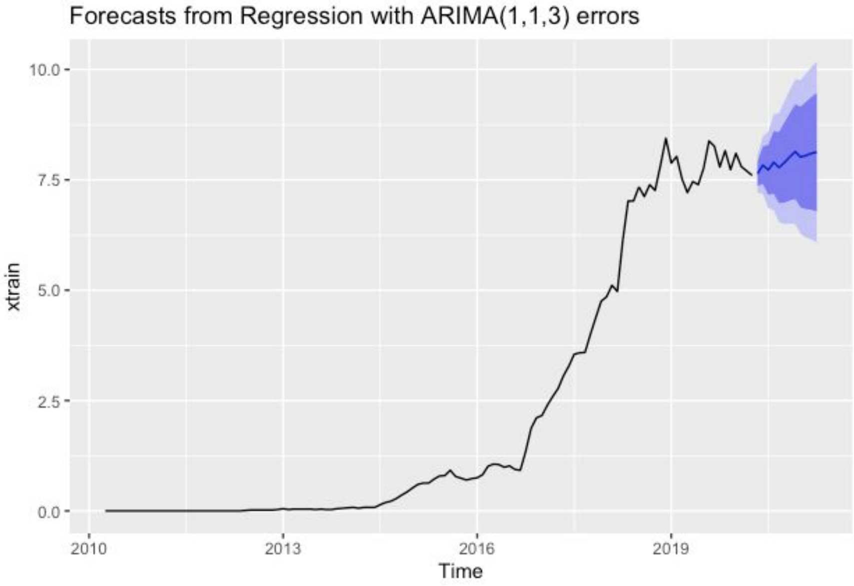
MAPE	MAE	RMSE
15.069%	1.48149	1.75348

Regression with ARIMA errors - Xiaomi Interest over time

- In the ARIMA part of the model, we will be using Xiaomi's Interest over time from Google Trends as xreg
- Tried to fit ARIMA models:

Model	AICc
Regression with ARIMA(0,1,0)	-13.13368
Regression with ARIMA(1,1,0)(1,0,0)	-9.291821
Regression with ARIMA(2,1,2)	-14.20796
Regression with ARIMA(3,1,2)	-11.92446
Regression with ARIMA(1,1,3) errors	-15.47369






```
      Ljung-Box test

data:  Residuals from Regression with ARIMA(1,1,3) errors
Q* = 31.2, df = 18, p-value = 0.02729

Model df: 6.   Total lags used: 24
```

MAPE	MAE	RMSE
16.57%	1.61387	1.70444

Model Selection

	Model	MAPE	RMSE
Samsung	Exponential Smoothing	2.81%	1.023
Apple	ARIMA(1,0,0) lambda=0	5.65%	1.957
Huawei	Exponential Smoothing	4.50%	0.530
Xiaomi	ARIMA(0, 1, 0) with drift	7.73%	1.029

Next Steps/Future Work

- Continue to obtain new data for fitting our best model
- Utilize cross validation such as sliding or expanding windows to ensure our models are a good fit and are not being over or underfit
- Train additional models such as Neural Networks, ARCH, GARCH, etc. to see if results can further be improved

Thank you!

Questions?