Yahoo's Tumblr Failed Forecasts

Oriental Mix / Data Hustler

6/7/2018

We decided to use a single environment to directly compute the NPV from R in order to avoid having to export the data to excel.

Problems linked to Yahoo's model:

- Estimated number of users in 2022 ~ 800M **UNREALISTIC**
- Financial estimation based on FB data instead of current Yahoo's advertising revenue per user.

Assumptions

- We only focus on the forecast aspect of visitors (Not on the Financial Model itself)

Load initial data

Our initial step is to load the data from our csv. We started by cleaning up the excel file and created two csv containing:

- previous website visits history for WORLD
- previous website visits history for US
- yahoo's forecast leading to 1.2Bn valuation

Additional problems that we noticed on Yahoo's forecasts: * Actual growth rate in the US Vs Rest of World is different. * We would also argue that their model doesn't reflect that trend.

```r
data_monthly_audience <- read.csv("tumblr_worldwide_monthly_direct_audience.csv", header = TRUE)
data_us_audience <- read.csv("tumblr_us_monthly_direct_audience.csv", header = TRUE)
# Load yahoo's forecast to compare valuation models - Not used for predictions
data_yahoo_forecast <-read.csv("model_pop_forecasts.csv", header = FALSE)

ts_world_audience <- ts(data_monthly_audience, start=c(2010, 4), frequency=12)
ts_us_audience <- ts(data_us_audience, start=c(2010, 4), frequency=12)
ts_yahoo_forecast <- ts(data_yahoo_forecast, start=c(2010, 4), frequency=12)

summary(data_monthly_audience)
```

```
##     Uniques              People              Visits
##  Min.   : 25349036   Min.   : 19020118   Min.   : 69837544
##  1st Qu.: 59974119   1st Qu.: 42795006   1st Qu.:193688420
##  Median :112207072   Median : 80234848   Median :348446480
##  Mean   :115313295   Mean   : 83600104   Mean   :333771114
##  3rd Qu.:165071420   3rd Qu.:119131732   3rd Qu.:471823920
##  Max.   :201892656   Max.   :147525568   Max.   :543773824
##
##    Page.Views          Mobile.Web
##  Min.   :1.016e+09   Min.   : 8937000
##  1st Qu.:4.205e+09   1st Qu.:15359716
##  Median :1.254e+10   Median :22140054
```

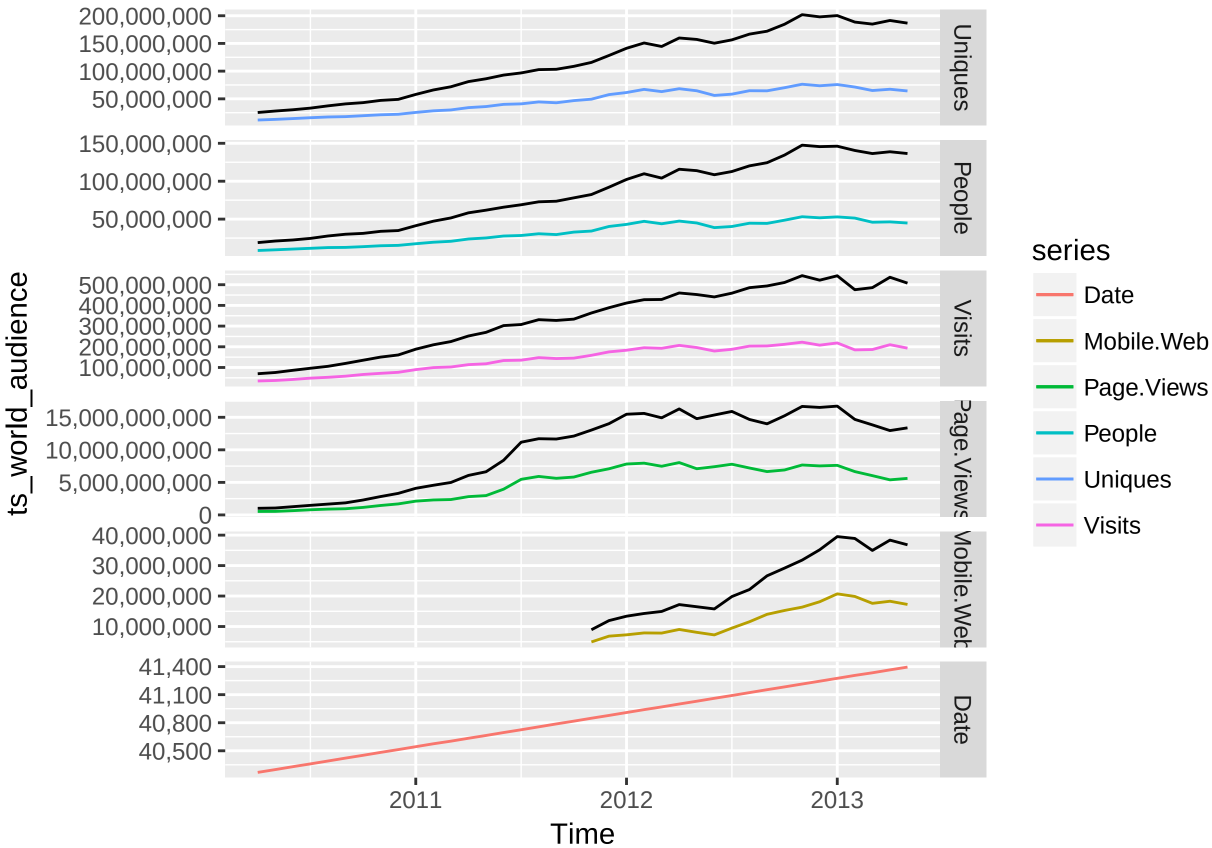
```
## Mean   :1.006e+10   Mean   :24543727
## 3rd Qu.:1.489e+10   3rd Qu.:35082936
## Max.   :1.672e+10   Max.   :39517752
##                     NA's   :19
```

Visualize Data

You can also embed plots, for example:

```
autoplot(ts_world_audience, facets = TRUE) +
  autolayer(ts_us_audience) +
  scale_y_continuous(labels=comma)
```

```
## Warning: Removed 19 rows containing missing values (geom_path).
```



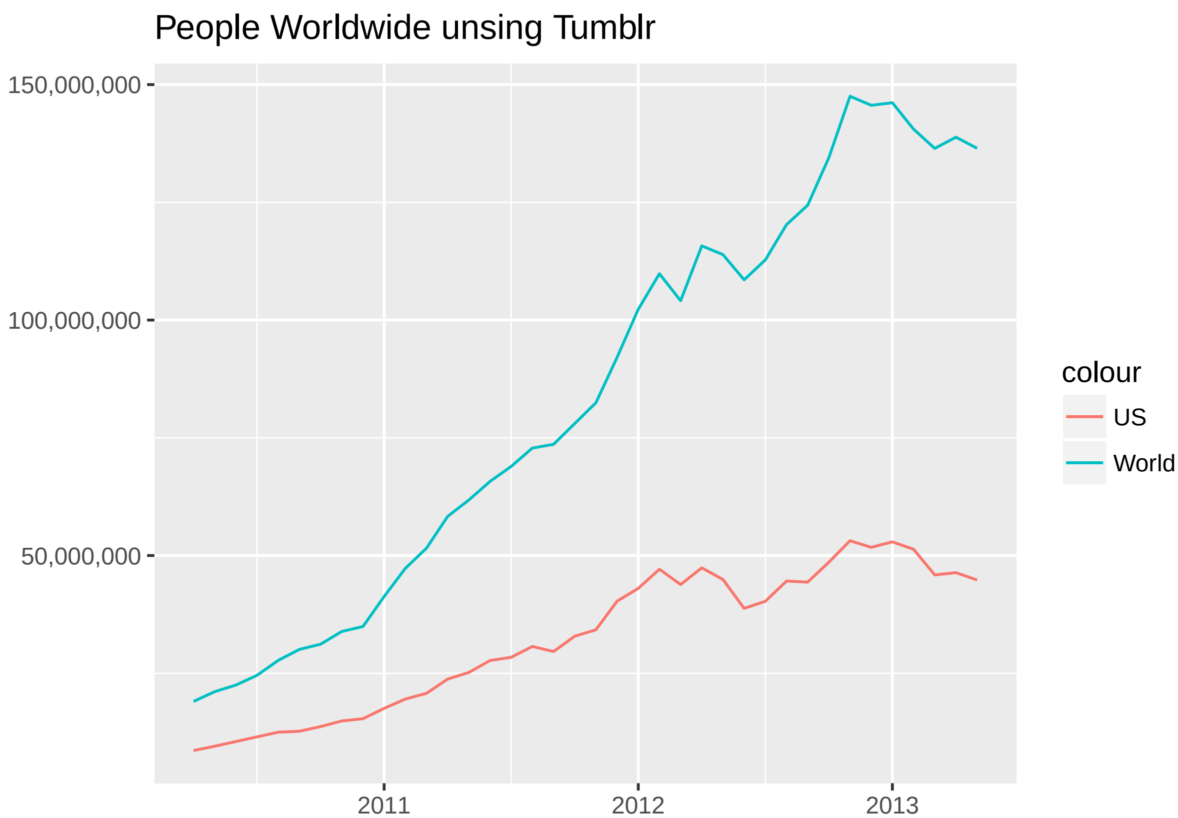
We also look at the mont ** We can see an upward trend and similar trajectory for both World visits and US visits. **

Based on the initial plotting, we cannot find sign of seasonality. Let's investigate.

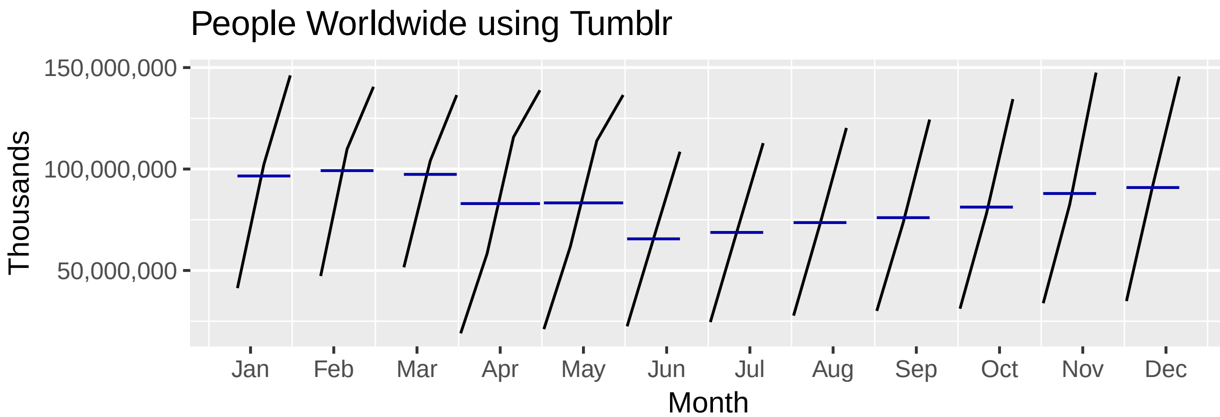
Monthly Breakdown + Timeserie decomposition

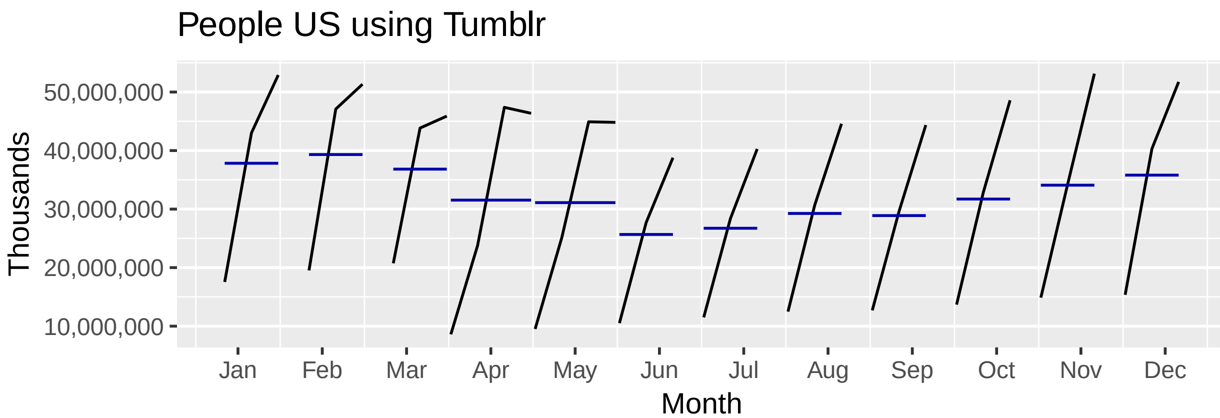
```
ts_world_data<-ts_world_audience[,"People"] # For worldwide data
ts_us_data<-ts_us_audience[,"People"] # For US data

data_world <- data.frame(Y=as.matrix(ts_world_audience), date=time(ts_world_audience))
```

```
data_us <- data.frame(Y=as.matrix(ts_us_audience), date=time(ts_world_audience))
ggplot() +
  geom_line(data = data_world, aes(y=Y.People, x=date, color="World")) +
  geom_line(data = data_us, aes(y=Y.People, x=date, color="US")) +
  scale_y_continuous(labels=comma) +
  ggtitle("People Worldwide unsing Tumblr") + xlab("") + ylab("")
```




```
plot_world_monthly_break <- ggsubseriesplot(ts_world_data) + ggtitle("People Worldwide using Tumblr") +
plot_us_monthly_break <- ggsubseriesplot(ts_us_data) + ggtitle("People US using Tumblr") + xlab("Month"
grid.arrange(grobs=list(plot_world_monthly_break, plot_us_monthly_break), ncol=1)
```

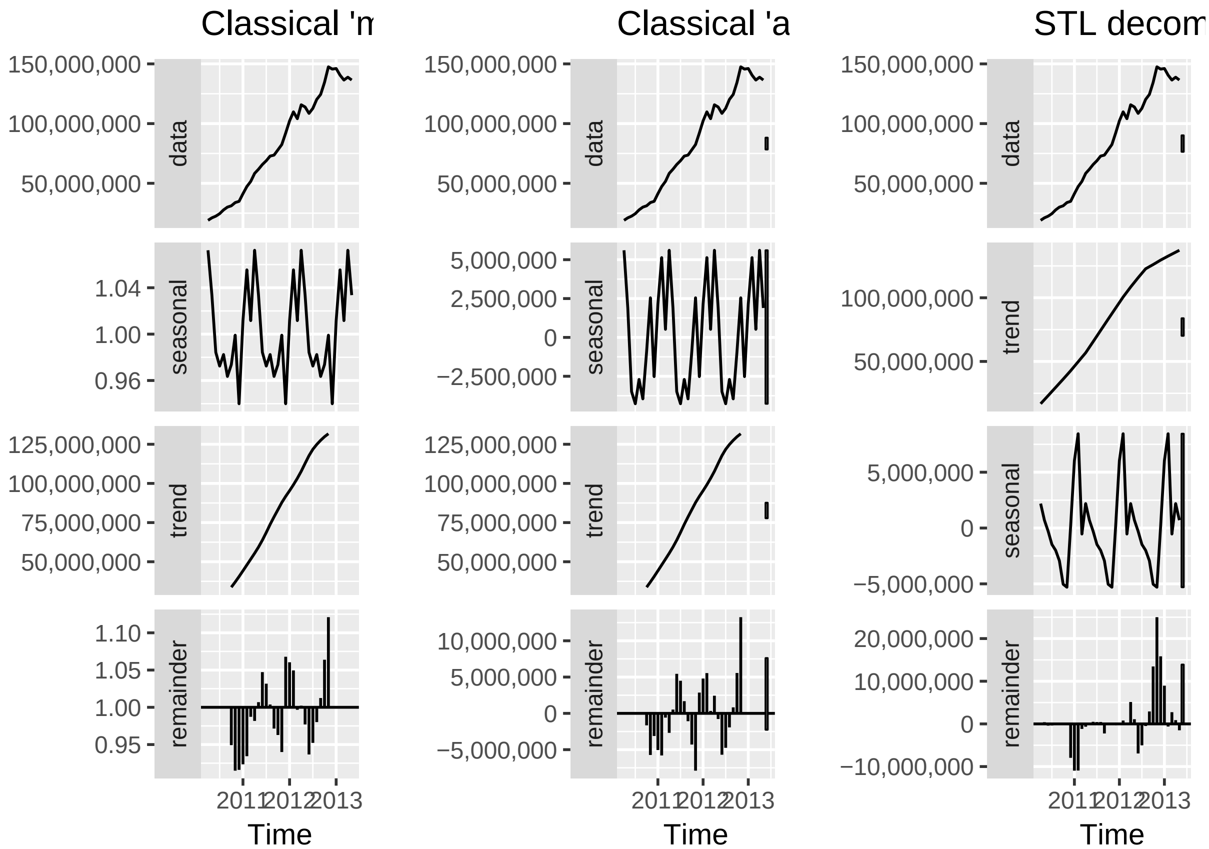




We can see a repetitive pattern across the years and visits seem to always diminush during the summer.

We plot 3 differents decomposition method to have a better overview.

```
fit_classical_additive <- decompose(ts_world_data, type="additive") #decompose using "classical" method
plot_additive <- autoplot(fit_classical_additive) + scale_y_continuous(labels=comma) + ggtitle("Classica
fit_classical_multiplicative <- decompose(ts_world_data, type="multiplicative") #decompose using "class
plot_multiplicative <- autoplot(fit_classical_multiplicative) + scale_y_continuous(labels=comma) + ggti
fit_stl <- stl(ts_world_data, t.window=12, s.window="periodic", robust=TRUE) #decompose using STL (Seas
plot_stl <- autoplot(fit_stl) + scale_y_continuous(labels=comma)  + scale_y_continuous(labels=comma)  +
grid.arrange(grobs=list(plot_multiplicative, plot_additive, plot_stl), ncol=3)
```

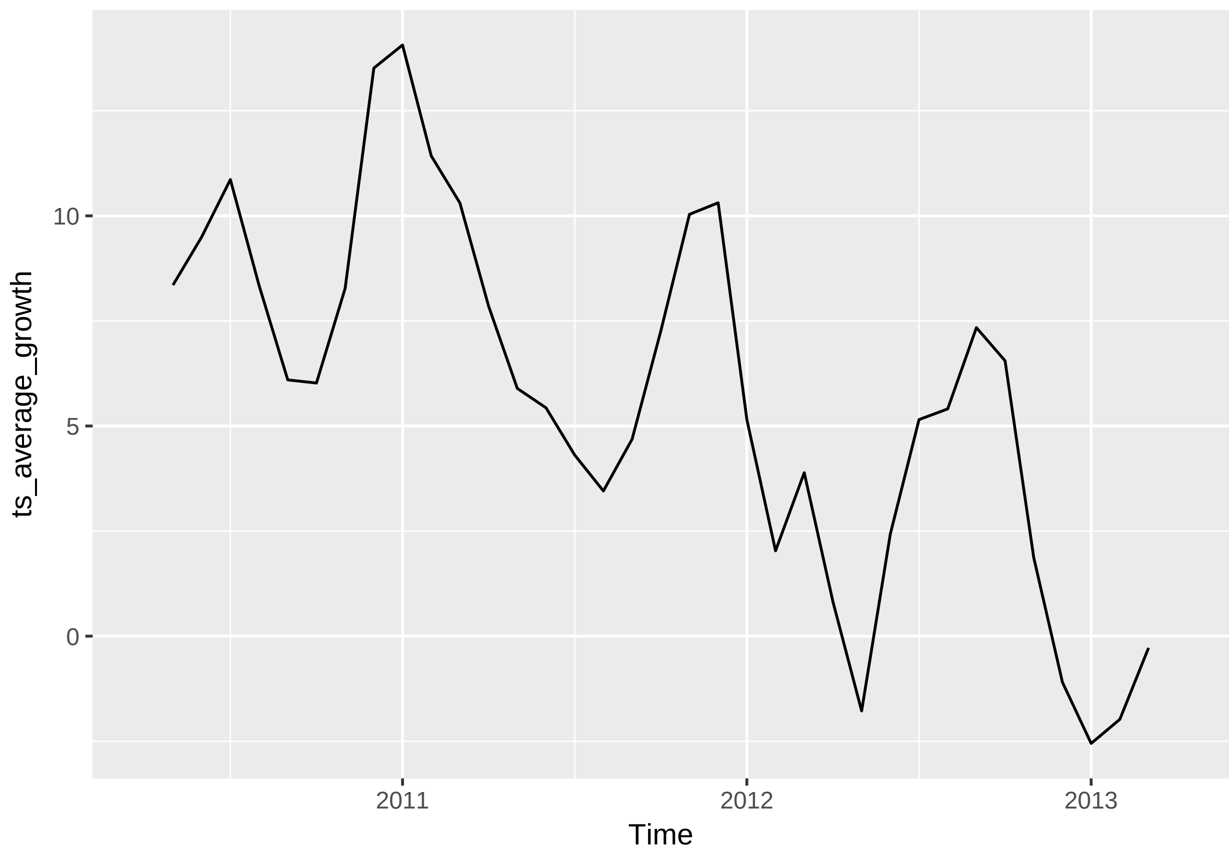


The decomposition and monthly breakdown seem to indicate seasonality

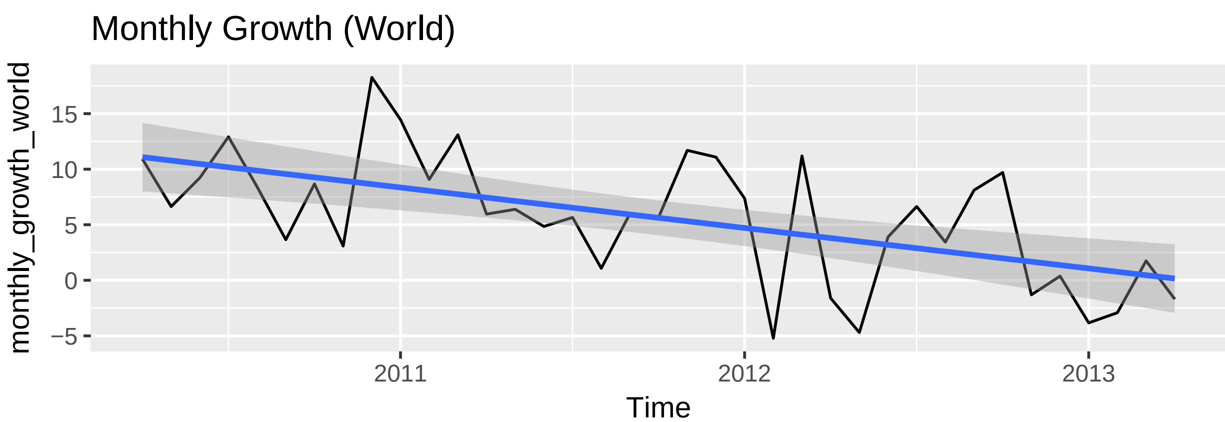
Analyze Monthly Growth

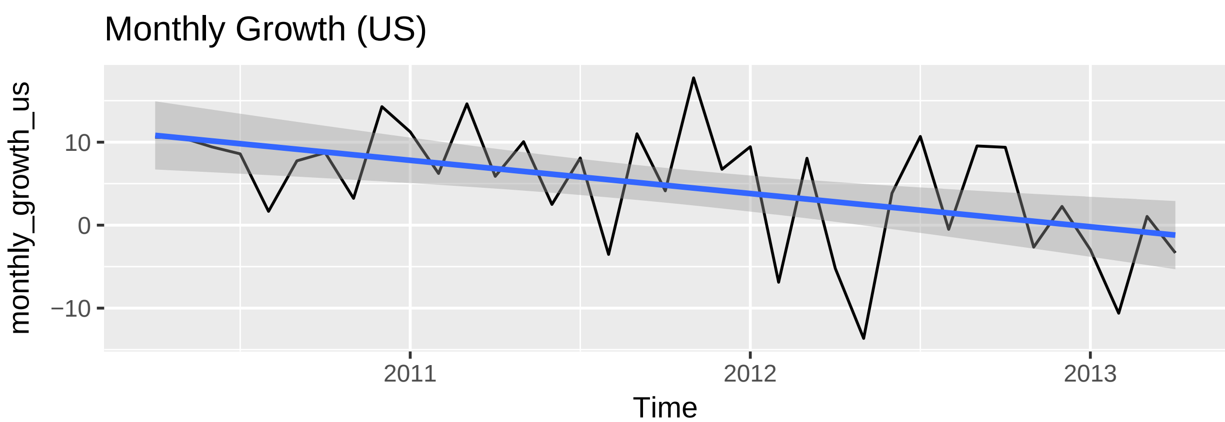
We then looked at the monthly growth in order to compare it with Yahoo's forecast of 1.

```
monthly_growth_world <- ts(data_monthly_audience$People, start = c(2010, 4), frequency = 12)
monthly_growth_world <- (lag(monthly_growth_world)/monthly_growth_world - 1) * 100
monthly_growth_us <- ts(data_us_audience$People, start = c(2010, 4), frequency = 12)
monthly_growth_us <- (lag(monthly_growth_us)/monthly_growth_us - 1) * 100

growth_last_year_world <- window(ts_world_audience, start = c(2012,5), end=c(2013,5))
growth_last_year_world <- (lag(growth_last_year_world)/growth_last_year_world - 1) * 100
growth_last_year_us <- window(ts_us_audience, start = c(2012,5), end=c(2013,5))
growth_last_year_us <- (lag(growth_last_year_us)/growth_last_year_us - 1) * 100

average_world_growth <- mean(monthly_growth_world) # 5.6%
average_us_growth <- mean(monthly_growth_us) # 4.8%
average_lastyear_growth_world <- mean(growth_last_year_world) # 2.2%
average_lastyear_growth_us <- mean(growth_last_year_us) # 1.05%

ts_average_growth <- ma(monthly_growth_world, order=2)
autoplot(ts_average_growth)
```




```
plot_growth_world <- autoplot(monthly_growth_world, ts.geom = 'bar') + ggtitle("Monthly Growth (World)")
plot_growth_us <- autoplot(monthly_growth_us, ts.geom = 'bar') + ggtitle("Monthly Growth (US)") + geom_

grid.arrange(grobs=list(plot_growth_world, plot_growth_us))
```





We can see that yearly growth trend is already 'dampening'

- Average growth World **5.6%** VS Last Year growth **2.2%**
- Average growth US **4.8%** VS Last Year growth **1.05%**

Fitting + Forecasting

We use 115 months horizon (until 2022) and forecast with confidence Intervals of 80% and 95%.

```
horizon <- 7 + 9*12 # 7months in 2013 + 9 years from 2014 to 2022
level<-c(0.8,0.95) # Confidence Interval 80%, 95%
```

We create a lot of different fitting models to test if any of them confirms Yahoo's predictions.

```
fit_drift <- rwf(ts_world_data, drift=TRUE, h=horizon)
# several exponential smoothing models to forecast the increase in the traffic
fit_ZZZ <- ets(ts_world_data, model="ZZZ", damped=FALSE)
fit_ZZZ_damped <- ets(ts_world_data, model="ZZZ", damped=TRUE)
fit_AAN <- ets(ts_world_data, model="AAN", damped=FALSE) # Additive noise, Additive Trend, No seasonali
fit_AAN_damped <- ets(ts_world_data, model="AAN", damped=TRUE) # Additive noise, Additive Trend, No sea
fit_AAZ <- ets(ts_world_data, model="AAZ", damped=FALSE) # Additive Noise, Additive Trend, Automatic Se
fit_AAZ_damped <- ets(ts_world_data, model="AAZ", damped=TRUE) # Additive Noise, Additive Trend, Automa
fit_us_AAZ <- ets(ts_us_data, model="AAZ", damped=FALSE) # Additive Noise, Additive Trend, Automatic Se
fit_us_AAZ_damped <- ets(ts_us_data, model="AAZ", damped=TRUE) # Additive Noise, Additive Trend, Automa

fit_MAZ <- ets(ts_world_data, model="MAZ", damped=FALSE)
fit_MAZ_damped <- ets(ts_world_data, model="MAZ", damped=TRUE)
```

```r
fit_MMN <- ets(ts_world_data, model="MMN", damped=FALSE)
fit_MMN_damped <- ets(ts_world_data, model="MMN", damped=TRUE)
fit_us_MMN <- ets(ts_us_data, model="MMN", damped=FALSE)
fit_us_MMN_damped <- ets(ts_us_data, model="MMN", damped=TRUE)

fit_MMM <- ets(ts_world_data, model="MMM", damped=FALSE)
fit_MMM_damped <- ets(ts_world_data, model="MMM", damped=TRUE)

fit_MMZ <- ets(ts_world_data, model="MMZ", damped=FALSE)
fit_MMZ_damped <- ets(ts_world_data, model="MMZ", damped=TRUE)
fit_TBATS <- tbats(ts_world_data)

fit_arima <- auto.arima(ts_world_data) #automatically fits the ARIMA model (auto-regressive integrated
fit_arima <- auto.arima(ts_world_data, seasonal = TRUE)
```

Based on our previous analysis, we have confirmation that our final model should use multiplicative decomposition.

Let's look at the valuation for each of the forecasts.

```r
valuation(ts_yahoo_forecast, debug=TRUE) #  1230
```

```
## ################ 2013 ##############
##  people Worldwide=145.024025428571
##  Percentage US People=33%
##  usPeoplePerYear=47.8579283914286
##  rowPeoplePerYear=97.1660970371429
##  usRevenuePerUser=13.58
##  rowRevenuePerUser=3.21
##  usUsers=23.9289641957143
##  rowUsers=48.5830485185714
##  Revenue=62.3397858640167
##  Operating Cash Flows = 19.7617121188933
##  Terminal Value = 0
##  Total Cash Flows = 19.7617121188933
## ################ 2014 ##############
##  people Worldwide=167.49760325
##  Percentage US People=31%
##  usPeoplePerYear=52.476999098225
##  rowPeoplePerYear=115.020604151775
##  usRevenuePerUser=13.88
##  rowRevenuePerUser=3.24
##  usUsers=26.2384995491125
##  rowUsers=57.5103020758875
##  Revenue=122.338611659457
##  Operating Cash Flows = 38.7813398960479
##  Terminal Value = 0
##  Total Cash Flows = 38.7813398960479
## ################ 2015 ##############
##  people Worldwide=200.701055916667
##  Percentage US People=30%
##  usPeoplePerYear=59.5279331848833
##  rowPeoplePerYear=141.173122731783
##  usRevenuePerUser=14.18
```

```
## rowRevenuePerUser=3.27
## usUsers=29.7639665924417
## rowUsers=70.5865613658917
## Revenue=145.082467099397
## Operating Cash Flows = 45.991142070509
## Terminal Value = 0
## Total Cash Flows = 45.991142070509
## ################ 2016 ##############
## people Worldwide=240.486508833333
## Percentage US People=28%
## usPeoplePerYear=67.31217382245
## rowPeoplePerYear=173.174335010883
## usRevenuePerUser=14.48
## rowRevenuePerUser=3.3
## usUsers=33.656086911225
## rowUsers=86.5871675054417
## Revenue=171.795064720555
## Operating Cash Flows = 54.4590355164158
## Terminal Value = 0
## Total Cash Flows = 54.4590355164158
## ################ 2017 ##############
## people Worldwide=288.158727833333
## Percentage US People=26%
## usPeoplePerYear=75.8433771657333
## rowPeoplePerYear=212.3153506676
## usRevenuePerUser=14.78
## rowRevenuePerUser=3.33
## usUsers=37.9216885828667
## rowUsers=106.1576753338
## Revenue=203.108359136961
## Operating Cash Flows = 64.3853498464166
## Terminal Value = 0
## Total Cash Flows = 64.3853498464166
## ################ 2018 ##############
## people Worldwide=345.28112525
## Percentage US People=25%
## usPeoplePerYear=85.111797374125
## rowPeoplePerYear=260.169327875875
## usRevenuePerUser=15.08
## rowRevenuePerUser=3.36
## usUsers=42.5558986870625
## rowUsers=130.084663937937
## Revenue=239.739427340527
## Operating Cash Flows = 75.9973984669471
## Terminal Value = 0
## Total Cash Flows = 75.9973984669471
## ################ 2019 ##############
## people Worldwide=413.727032666667
## Percentage US People=23%
## usPeoplePerYear=95.0744721068
## rowPeoplePerYear=318.652560559867
## usRevenuePerUser=15.38
## rowRevenuePerUser=3.39
## usUsers=47.5372360534
```

```
##  rowUsers=159.326280279933
##  Revenue=282.49750681117
##  Operating Cash Flows = 89.551709659141
##  Terminal Value = 0
##  Total Cash Flows = 89.551709659141
## ################ 2020 ##############
##  people Worldwide=495.741137333333
##  Percentage US People=21%
##  usPeoplePerYear=105.642436365733
##  rowPeoplePerYear=390.0987009676
##  usRevenuePerUser=15.68
##  rowRevenuePerUser=3.42
##  usUsers=52.8212181828667
##  rowUsers=195.0493504838
##  Revenue=332.290106613766
##  Operating Cash Flows = 105.335963796564
##  Terminal Value = 0
##  Total Cash Flows = 105.335963796564
## ################ 2021 ##############
##  people Worldwide=594.013095083333
##  Percentage US People=20%
##  usPeoplePerYear=116.664171874367
##  rowPeoplePerYear=477.348923208967
##  usRevenuePerUser=15.98
##  rowRevenuePerUser=3.45
##  usUsers=58.3320859371834
##  rowUsers=238.674461604483
##  Revenue=390.127472402591
##  Operating Cash Flows = 123.670408751621
##  Terminal Value = 0
##  Total Cash Flows = 123.670408751621
## ################ 2022 ##############
##  people Worldwide=711.765739
##  Percentage US People=18%
##  usPeoplePerYear=127.9043032983
##  rowPeoplePerYear=583.8614357017
##  usRevenuePerUser=16.28
##  rowRevenuePerUser=3.48
##  usUsers=63.95215164915
##  rowUsers=291.93071785085
##  Revenue=457.12442821536
##  Operating Cash Flows = 144.908443744269
##  Terminal Value = 2132.22424366567
##  Total Cash Flows = 2277.13268740994
```

```
##      firmValue
## [1,]  1230.118
```

```r
valuation(pred_drift$mean) # 930
```

```
##      firmValue
## [1,]  929.6908
```

```r
valuation(pred_ZZZ$mean) # 703
```

```
##      firmValue
```

```
## [1,]  702.8438
```

```r
valuation(pred_ZZZ_damped$mean) # 351
```

```
##      firmValue
## [1,]  350.6712
```

```r
valuation(pred_AAN$mean) # 1004
```

```
##      firmValue
## [1,]  1004.333
```

```r
valuation(pred_AAN_damped$mean) # 485
```

```
##      firmValue
## [1,]     485.1
```

```r
valuation(pred_AAZ$mean) # 1004
```

```
##      firmValue
## [1,]  1004.333
```

```r
valuation(pred_AAZ_damped$mean) # 485
```

```
##      firmValue
## [1,]     485.1
```

```r
valuationAdvanced(pred_AAZ$mean, pred_us_AAZ$mean, debug=TRUE) # 1338
```

```
## ################ 2013 ##############
##  people Worldwide=151
##  Percentage US People=33%
##  usPeoplePerYear=50
##  rowPeoplePerYear=101
##  usRevenuePerUser=13.58
##  rowRevenuePerUser=3.21
##  Revenue=64.75
##  Operating Cash Flows = 21
##  Terminal Value = 0
##  Total Cash Flows = 21
## ################ 2014 ##############
##  people Worldwide=185
##  Percentage US People=34%
##  usPeoplePerYear=63
##  rowPeoplePerYear=122
##  usRevenuePerUser=13.88
##  rowRevenuePerUser=3.24
##  Revenue=141
##  Operating Cash Flows = 45
##  Terminal Value = 0
##  Total Cash Flows = 45
## ################ 2015 ##############
##  people Worldwide=228
##  Percentage US People=34%
##  usPeoplePerYear=78
##  rowPeoplePerYear=150
##  usRevenuePerUser=14.18
##  rowRevenuePerUser=3.27
```

```
##  Revenue=177
##  Operating Cash Flows = 56
##  Terminal Value = 0
##  Total Cash Flows = 56
## ################ 2016 ##############
##  people Worldwide=270
##  Percentage US People=35%
##  usPeoplePerYear=94
##  rowPeoplePerYear=176
##  usRevenuePerUser=14.48
##  rowRevenuePerUser=3.3
##  Revenue=216
##  Operating Cash Flows = 68
##  Terminal Value = 0
##  Total Cash Flows = 68
## ################ 2017 ##############
##  people Worldwide=313
##  Percentage US People=35%
##  usPeoplePerYear=110
##  rowPeoplePerYear=203
##  usRevenuePerUser=14.78
##  rowRevenuePerUser=3.33
##  Revenue=256
##  Operating Cash Flows = 81
##  Terminal Value = 0
##  Total Cash Flows = 81
## ################ 2018 ##############
##  people Worldwide=356
##  Percentage US People=35%
##  usPeoplePerYear=126
##  rowPeoplePerYear=230
##  usRevenuePerUser=15.08
##  rowRevenuePerUser=3.36
##  Revenue=297
##  Operating Cash Flows = 94
##  Terminal Value = 0
##  Total Cash Flows = 94
## ################ 2019 ##############
##  people Worldwide=399
##  Percentage US People=36%
##  usPeoplePerYear=142
##  rowPeoplePerYear=257
##  usRevenuePerUser=15.38
##  rowRevenuePerUser=3.39
##  Revenue=339
##  Operating Cash Flows = 107
##  Terminal Value = 0
##  Total Cash Flows = 107
## ################ 2020 ##############
##  people Worldwide=442
##  Percentage US People=36%
##  usPeoplePerYear=158
##  rowPeoplePerYear=284
##  usRevenuePerUser=15.68
```

```
##   rowRevenuePerUser=3.42
##   Revenue=383
##   Operating Cash Flows = 121
##   Terminal Value = 0
##   Total Cash Flows = 121
## ################ 2021 ##############
##   people Worldwide=485
##   Percentage US People=36%
##   usPeoplePerYear=174
##   rowPeoplePerYear=311
##   usRevenuePerUser=15.98
##   rowRevenuePerUser=3.45
##   Revenue=428
##   Operating Cash Flows = 136
##   Terminal Value = 0
##   Total Cash Flows = 136
## ################ 2022 ##############
##   people Worldwide=528
##   Percentage US People=36%
##   usPeoplePerYear=189
##   rowPeoplePerYear=339
##   usRevenuePerUser=16.28
##   rowRevenuePerUser=3.48
##   Revenue=473
##   Operating Cash Flows = 150
##   Terminal Value = 2207.14285714286
##   Total Cash Flows = 2357.14285714286

##      firmValue
## [1,]  1325.968
```

```
valuation(pred_MAZ$mean) # 702
```

```
##      firmValue
## [1,]  702.8438
```

```
valuation(pred_MAZ_damped$mean) # 350
```

```
##      firmValue
## [1,]  350.6712
```

```
valuation(pred_MMN$mean) # 546
```

```
##      firmValue
## [1,]  546.3178
```

```
valuation(pred_MMN_damped$mean) # 513
```

```
##      firmValue
## [1,]  512.6945
```

```
valuationAdvanced(pred_MMN$mean, pred_us_MMN$mean) # 515
```

```
##      firmValue
## [1,]  515.8731
```

```
valuationAdvanced(pred_MMN$mean, pred_us_MMN_damped$mean) # 620
```

```
##      firmValue
```

```
## [1,]  620.0876
```

```r
valuation(pred_MMM$mean) # 55.23
```

```
##      firmValue
## [1,]  55.23726
```

```r
valuation(pred_MMM_damped$mean) # 251
```

```
##      firmValue
## [1,]  251.6808
```

```r
valuation(pred_arima$mean) # 515
```

```
##      firmValue
## [1,]  929.6908
```

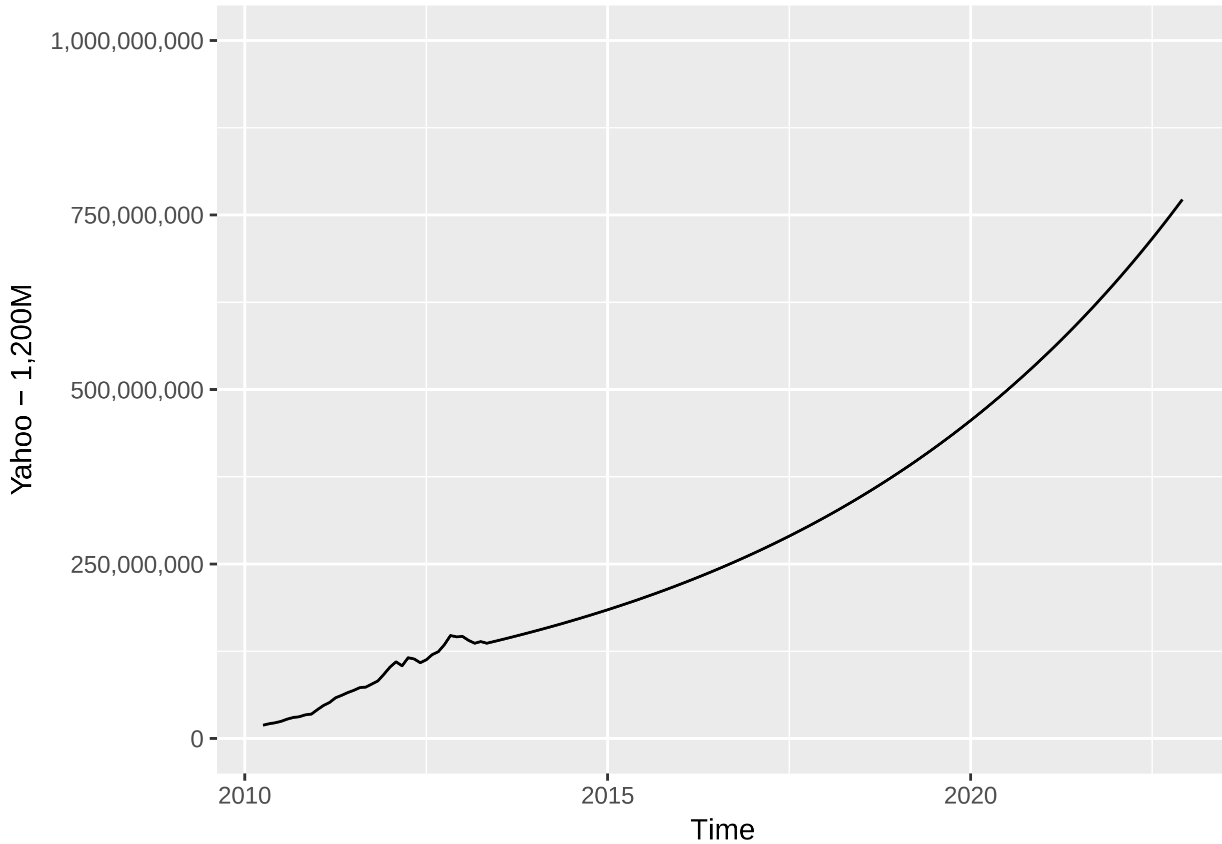
```r
valuation(pred_tbats$mean) # 454
```

```
##      firmValue
## [1,]  454.2923
```

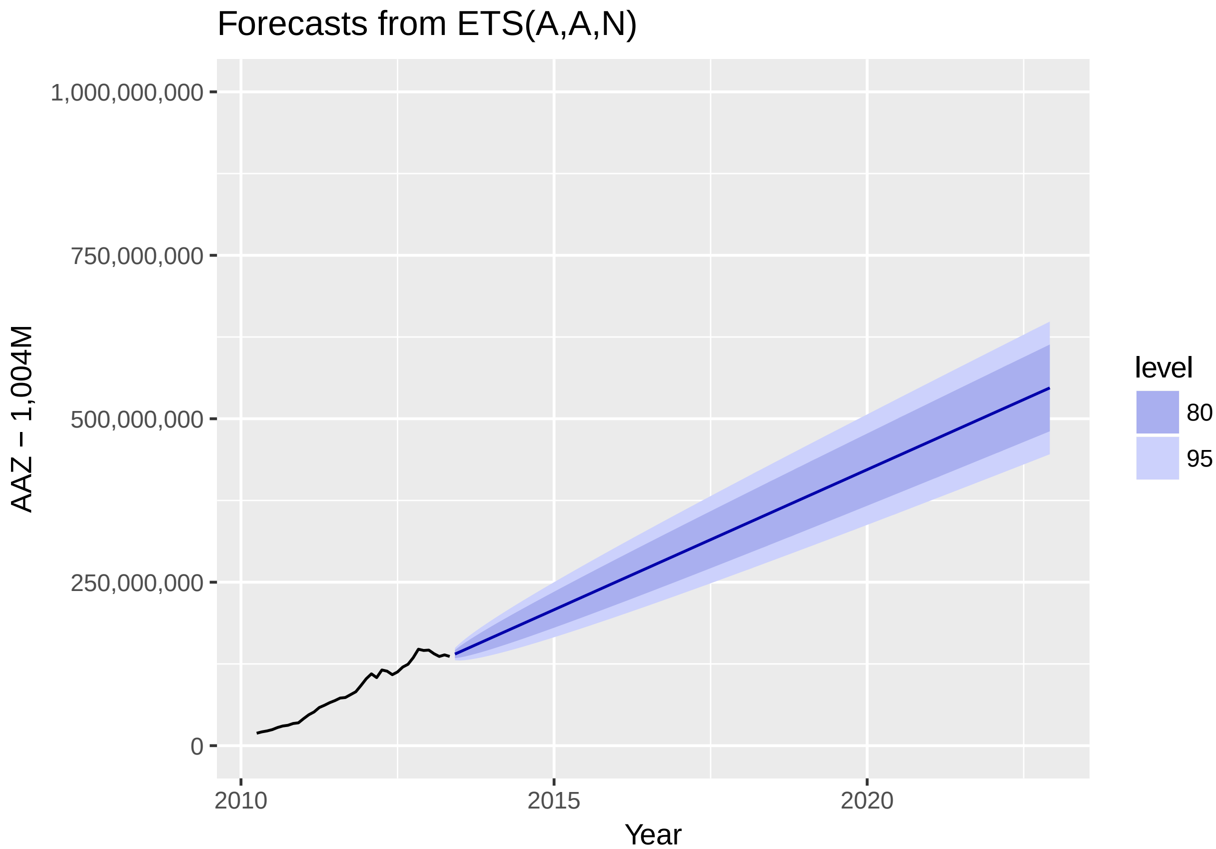
Visualize our Models

We plot our models together using a common scale on Y-axis.

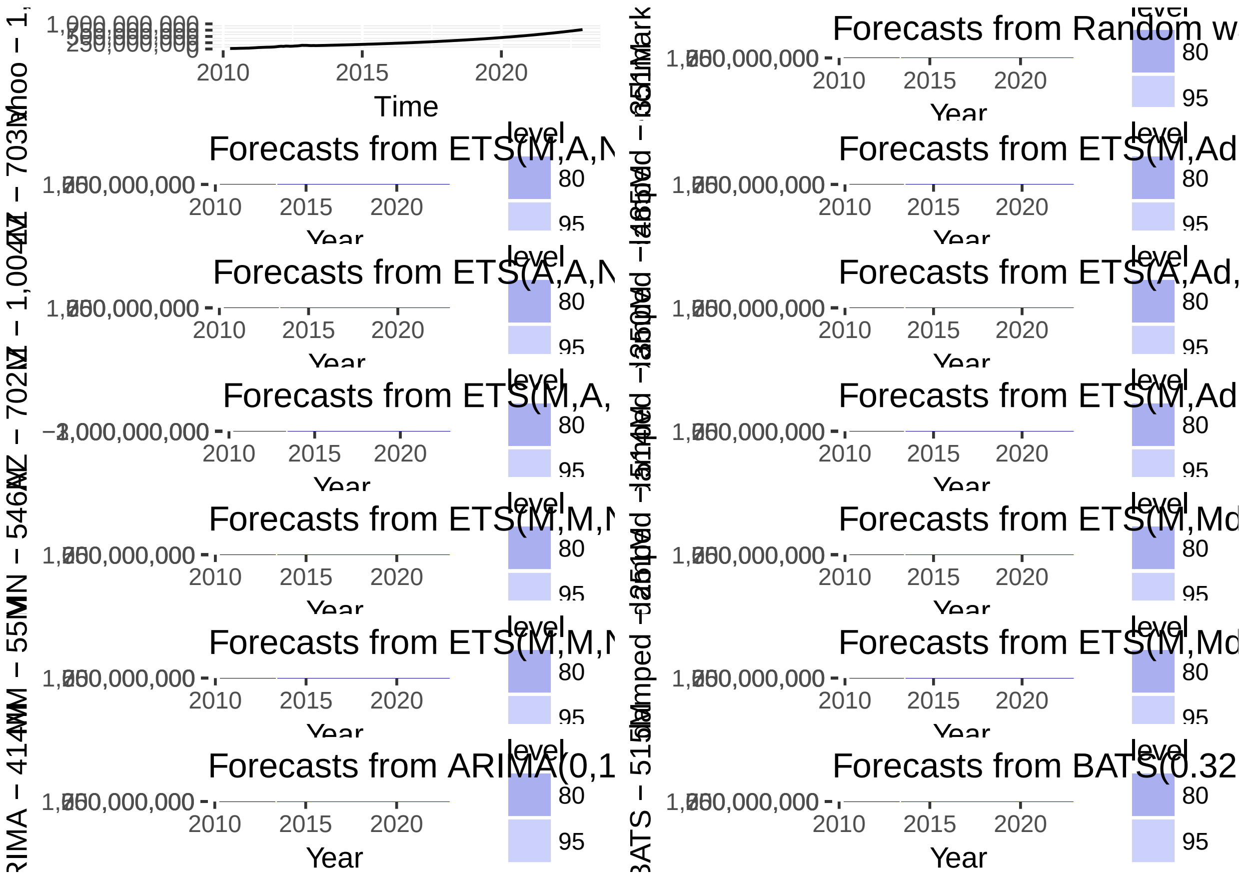
```r
limits<-c(0,1000000000)

ts_yahoo_small <- window(ts_yahoo_forecast, start=c(2013,6))
plot_yahoo <- autoplot(ts_yahoo_forecast, ylab="Yahoo - 1,200M") + scale_y_continuous(labels=comma, limi
plot_yahoo
```




```
plot_drift<-autoplot(fit_drift, xlab="Year", ylab="Drift Benchmark - 930M") + scale_y_continuous(labels=

plot_zzz<-autoplot(pred_ZZZ, xlab="Year", ylab="ZZZ - 703M") + scale_y_continuous(labels=comma, limits =
plot_zzz_damped<-autoplot(pred_ZZZ_damped, xlab="Year", ylab="ZZZ+damped - 351M") + scale_y_continuous(l

plot_AAN<-autoplot(pred_AAN, xlab="Year", ylab="AAN - 1,004M") + scale_y_continuous(labels=comma, limits
plot_AAN_damped<-autoplot(pred_AAN_damped, xlab="Year", ylab="AAN+damped - 485M") + scale_y_continuous(l

plot_AAZ<-autoplot(pred_AAZ, xlab="Year", ylab="AAZ - 1,004M") + scale_y_continuous(labels=comma, limits
plot_AAZ
```




```
plot_AAZ_damped<-autoplot(pred_AAZ_damped, xlab="Year", ylab="AAZ+damped - 485M") + scale_y_continuous(l
plot_MAZ<-autoplot(pred_MAZ, xlab="Year", ylab="MAZ - 702M") + scale_y_continuous(labels=comma)
plot_MAZ_damped<-autoplot(pred_MAZ_damped, xlab="Year", ylab="MAZ+damped - 350M") + scale_y_continuous(l

plot_MMN<-autoplot(pred_MMN, xlab="Year", ylab="MMN - 546M") + scale_y_continuous(labels=comma, limits =
plot_MMN_damped<-autoplot(pred_MMN_damped, xlab="Year", ylab="MMN+damped - 514M") + scale_y_continuous(l

plot_MMM<-autoplot(pred_MMN, xlab="Year", ylab="MMM - 55M") + scale_y_continuous(labels=comma, limits =
plot_MMM_damped<-autoplot(pred_MMN_damped, xlab="Year", ylab="MMM+damped - 251M") + scale_y_continuous(l


plot_tbats <- autoplot(pred_tbats, xlab="Year", ylab="TBATS - 515M") + scale_y_continuous(labels=comma,
plot_arima <- autoplot(pred_arima, xlab="Year", ylab="ARIMA - 414M") + scale_y_continuous(labels=comma,

# Global view of our models
lay <- rbind(c(1,2),
             c(3,4), # zzz
             c(5,6), # aaz
             c(7,8), # maz
             c(9,10), # mmn
             c(11,12), # mmm
             c(13,14) ) # arima, tbats
grid.arrange(grobs=list(plot_yahoo, plot_drift, plot_zzz,plot_zzz_damped, plot_AAZ,plot_AAZ_damped,plot_
```

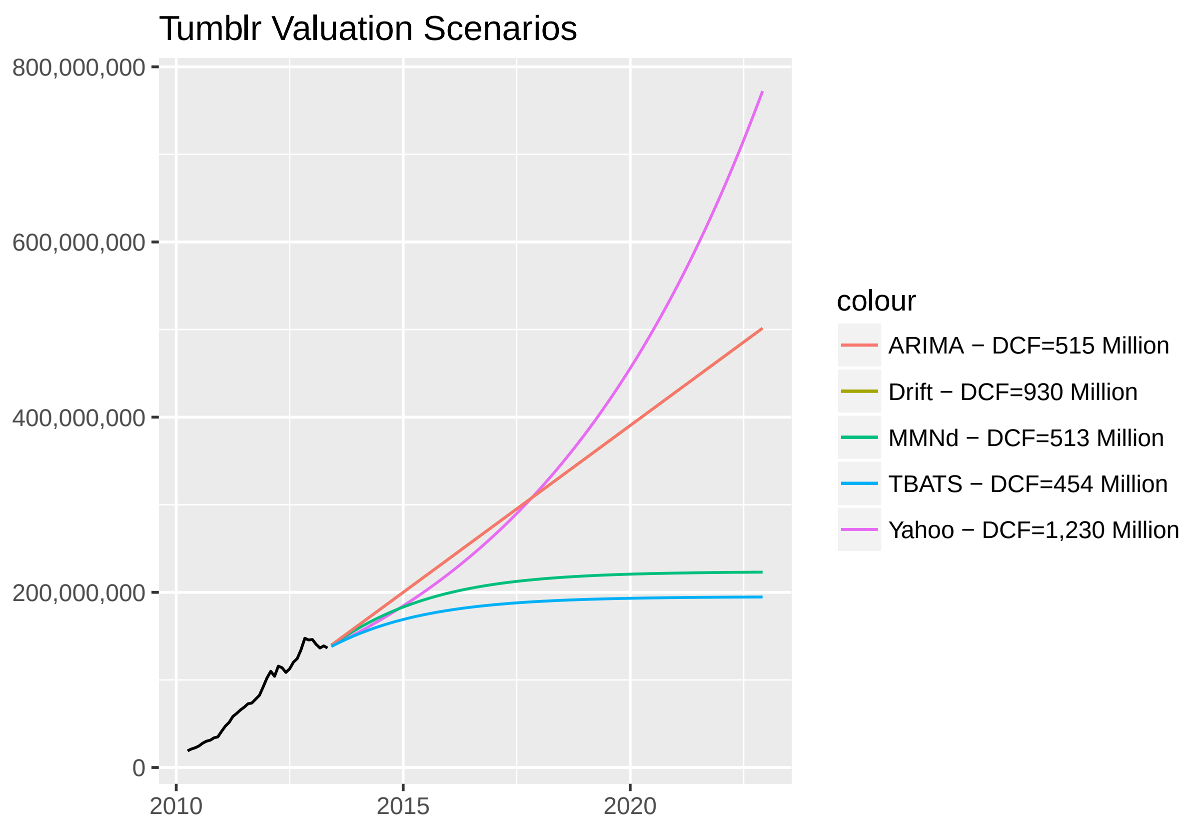


Summary forecasts

We select the few models that we believe fit best the visits and compare them to Yahoo's predictions.

```r
# Arrange the data for plotting
data <- data.frame(Date=time(ts_world_audience), People=data_monthly_audience$People)
ts_yahoo_small <- window(ts_yahoo_forecast, start=c(2013,6))
yahoo <- data.frame(Date=time(ts_yahoo_small), People=ts_yahoo_small[,1])
drift <- data.frame(Date=time(pred_drift$mean), People=fit_drift$mean)

mmn_damped <- data.frame(Date=time(pred_MMN_damped$mean), People=pred_MMN_damped$mean)
arima = data.frame(Date=time(pred_arima$mean), People=pred_arima$mean)
tbats = data.frame(Date=time(pred_tbats$mean), People=pred_tbats$mean)

ggplot() +
  geom_line(data = data, aes(y=People, x=Date)) +
  geom_line(data=yahoo, aes(y=People, x=Date, color="Yahoo - DCF=1,230 Million")) +
  geom_line(data=drift, aes(y=People, x=Date, color="Drift - DCF=930 Million")) +
  geom_line(data=mmn_damped, aes(y=People, x=Date, color="MMNd - DCF=513 Million")) +
  geom_line(data=arima, aes(y=People, x=Date, color="ARIMA - DCF=515 Million")) +
  geom_line(data=tbats, aes(y=People, x=Date, color="TBATS - DCF=454 Million")) +
  scale_y_continuous(labels=comma) +
  ggtitle("Tumblr Valuation Scenarios") + xlab("") + ylab("")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



We conclude that Yahoo overpaid for Tumblr using inaccurate forecasts

We would instead recommend using “MMMd” model for a valuation of **513M**.