Intro

For this problem, I will be using the tractor sales data set from UC analytics. This is a nice, simple data set and we can assume that there may be some other numbers baked into the "sales" variable. Our goal will be to developed methods of forecast, can compare that to actual data. Specifically, the method we will be using decomposition to interpret our time series data, and build a forecasting model on top of this interpretation.

The Data and Objective

In this scenario, we are selling tractors and would like to predict sales for the next 24 months. We have 12 years of previous data given on a monthly timeline. The project can be broken down into the following steps:

1.) Checking pattern of the tractor sales
2.) Identify the components of the series
3.) Propose a model and check its error.

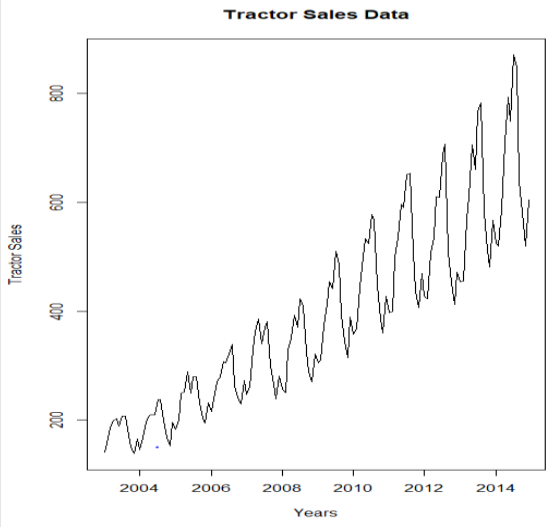
The software used for the problem is R/R studio. In the below code we can see the libraries that are needed for download, the overview of the data, and the dimensions of the data:

```
#general libraries I like to download
library(ggthemes)
library(dplyr)
library(viridis)
library(tidyr)
library(cluster)
library(ggmap)
library(maps)
library(lubridate)
library(reshape)
library(zoo)
library(car)

#specific for problem
library(data.table)
library(ggplot2)
library(fpp2)
library(forecast)
library(stats)
library(tseries)

#download dataset
install.packages("DescTools")
library(DescTools)
tractor<-read.csv('http://ucanalytics.com/blogs/wp-
content/uploads/2015/06/Tractor-Sales.csv')
head(tractor)
> head(tractor)
  Month.Year Number.of.Tractor.Sold
1     Jan-03                    141
2     Feb-03                    157
3     Mar-03                    185
4     Apr-03                    199
5     May-03                    203
6     Jun-03                    189

> dim(tractor)
[1] 144   2

> summary(tractor$Number.of.Tractor.Sold)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  138.0   248.5   369.0   389.7   509.2   871.0
```

Getting the data ready for time series

In order to get started in our time series analysis, we need to transform the data into a time series object. This will help with our plotting.

```
> tractorTS<-ts(tractor[,2],start=c(2003,1),frequency = 12)
> tractorTS
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2003 141 157 185 199 203 189 207 207 171 150 138 165
2004 145 168 197 208 210 209 238 238 199 168 152 196
2005 183 200 249 251 289 249 279 279 232 204 194 232
2006 215 239 270 279 307 305 322 339 263 241 229 272
2007 247 261 330 362 385 340 370 381 299 266 239 281
2008 257 250 329 350 393 370 423 410 326 289 270 321
2009 305 310 374 414 454 441 510 486 393 345 315 389
2010 358 368 444 482 534 524 578 567 447 386 360 428
2011 397 400 498 536 596 591 651 654 509 437 406 470
2012 428 423 507 536 610 609 687 707 509 452 412 472
2013 454 455 568 610 706 661 767 783 583 513 481 567
2014 525 520 587 710 793 749 871 848 640 581 519 605

> plot(tractorTS,xlab="Years",ylab="Tractor Sales",main="Tractor Sales
Data")
```

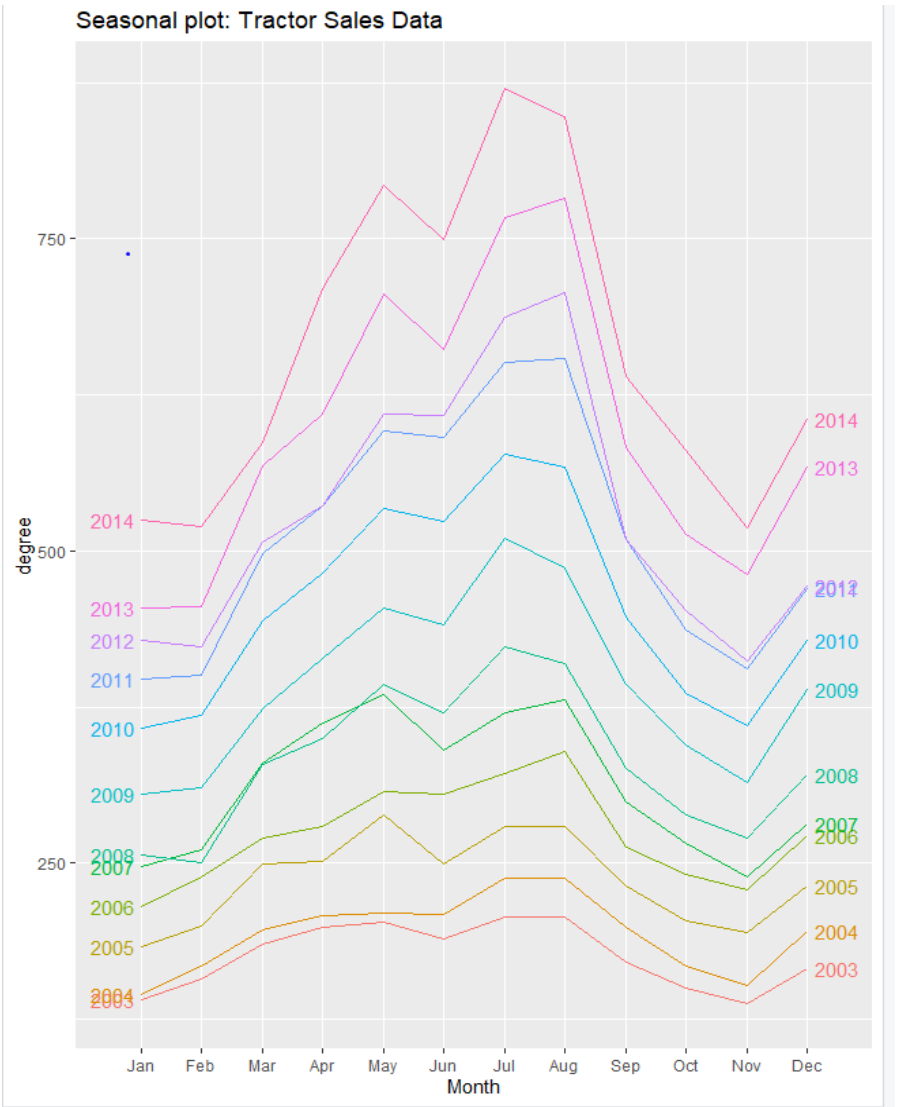


The above plot, shows two things: first – that there is an upward trend, and second – that there is some very apparent seasonality. It is important to point out that there are two types of trends, which are multiplicative and additive. Given that **T = trend, S= Seasonality, and I = randomness** ; additive Trends can be expressed as $Y_t = T_t + S_t + l_t$, while multiplicative trends can be expressed as $Y_t = T_t * S_t * l_t$. Multiplicative trends can be converted to additive trends with the use of log.

Looking at our trend in the tractor sales, we can see that the error increases at time goes on. This is a hint, that in this case, the trend here is multiplicative. We also have seasonality; this means there are fluctuations in a similar manner within the years. We will need to create plots to further understand this.

Visualizing Seasonality

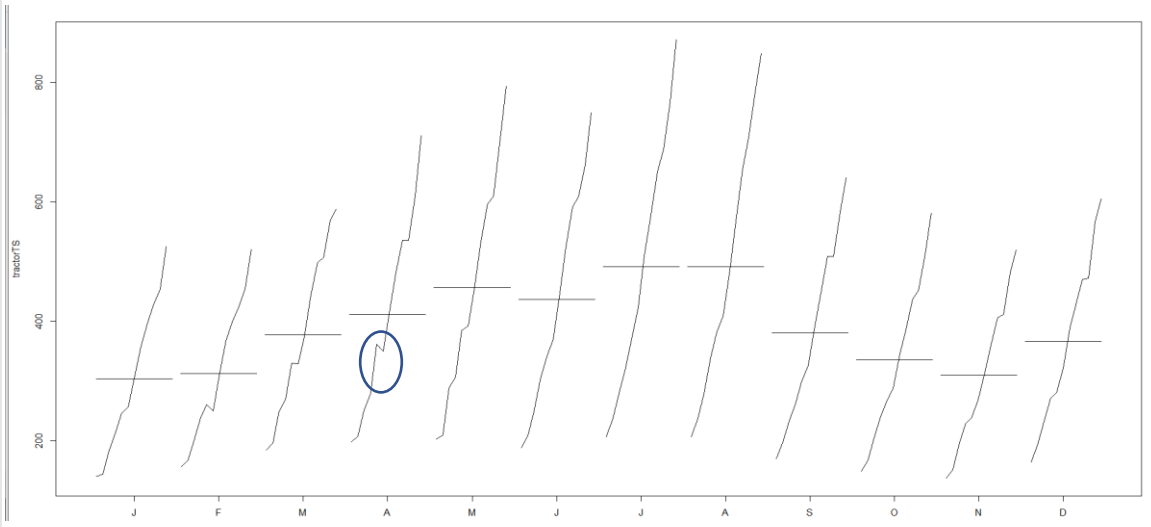
```
> ggseasonplot(tractorTS,year.labels=T,year.labels.left=T)+ylab("degree")
+ggtitle("Seasonal plot: Tractor Sales Data")
```



In the above visualization, we can somewhat see that there is a seasonality spike in the summer months (June to September) of tractor sales. What is more apparent is that as the years go on, there is an increase in sales. This also reinforces that this is multiplicative because as the years go on, the “bump” in sales gets bigger.

We can take an even deeper dive into the monthly trend. With the below visualization

```
> monthplot(tractorTS)
```



In the above plot, we lose all sense of the annual trend in order to get a more granular look the monthly sales data. The horizontal line equals the average, and the size of the line tells of the high and the low point. We can see that July and August have highest monthly average sales. The area where the dark blue circle is shown, is an indication that something occurred during this time. This could be an outside event causing disruption or a deal lost with a large customer.

What can we do with this information?

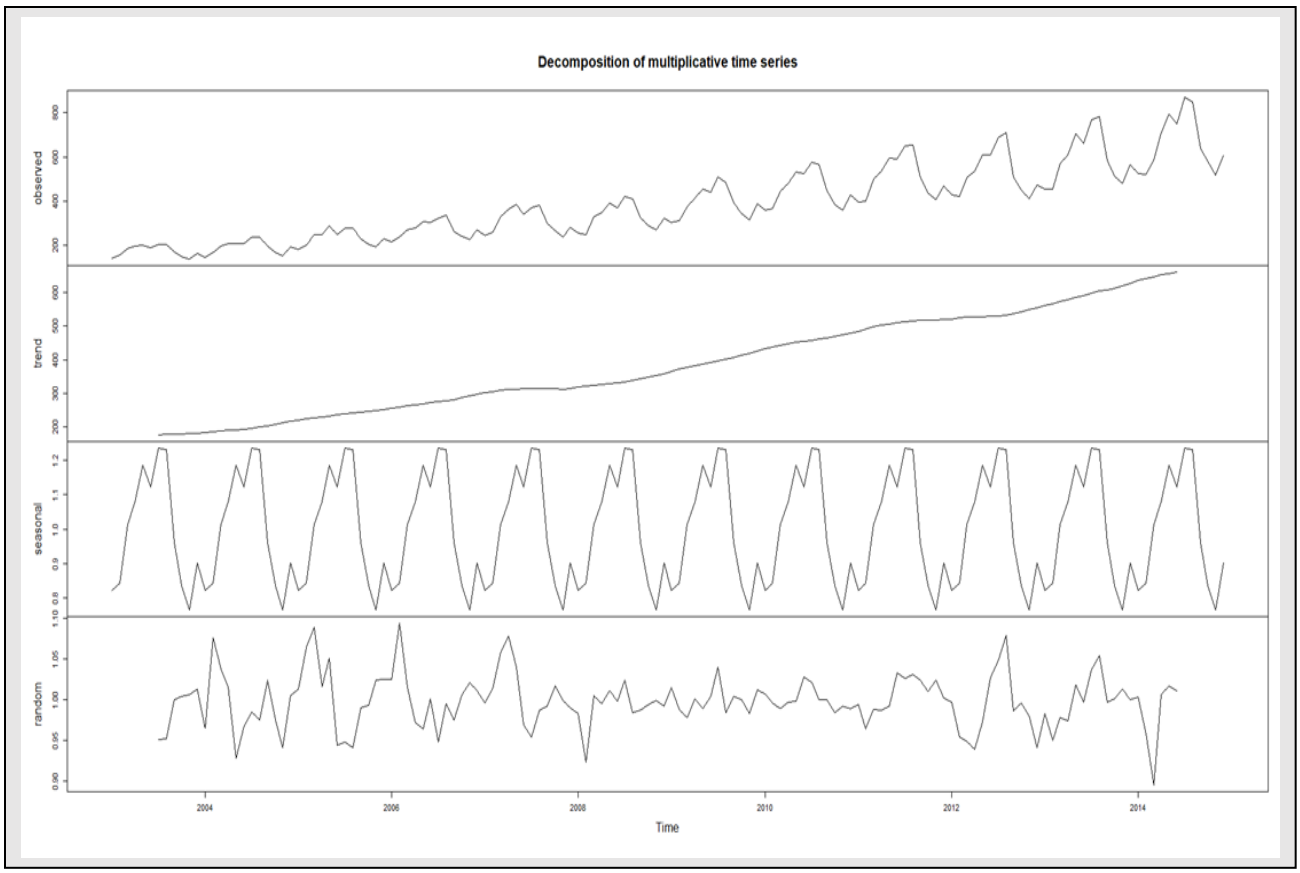
We can now inform our procurement department to increase stock before the orders come in. This can ensure that we don't have stockout cost due to not being able to meet demand in the summer months. We can also run leaner in the down months.

Interpreting Decomposition

```
TSDecompose<-decompose(tractorTS,type = "multiplicative")


> TSDecompose
$x
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2003 141 157 185 199 203 189 207 207 171 150 138 165
2004 145 168 197 208 210 209 238 238 199 168 152 196
2005 183 200 249 251 289 249 279 279 232 204 194 232
2006 215 239 270 279 307 305 322 339 263 241 229 272
2007 247 261 330 362 385 340 370 381 299 266 239 281
2008 257 250 329 350 393 370 423 410 326 289 270 321
2009 305 310 374 414 454 441 510 486 393 345 315 389
2010 358 368 444 482 534 524 578 567 447 386 360 428
2011 397 400 498 536 596 591 651 654 509 437 406 470
2012 428 423 507 536 610 609 687 707 509 452 412 472
2013 454 455 568 610 706 661 767 783 583 513 481 567
2014 525 520 587 710 793 749 871 848 640 581 519 605

$seasonal
           Jan       Feb       Mar       Apr       May       Jun       Jul       Aug       Sep       Oct       Nov       Dec
2003 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2004 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2005 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2006 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2007 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2008 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2009 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2010 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2011 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2012 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2013 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095
2014 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095

$trend
          Jan       Feb       Mar       Apr       May       Jun       Jul       Aug       Sep       Oct       Nov       Dec
2003       NA        NA        NA        NA        NA        NA 176.1667 176.7917 177.7500 178.6250 179.2917 180.4167
2004 182.5417 185.1250 187.5833 189.5000 190.8333 192.7083 195.5833 198.5000 202.0000 205.9583 211.0417 216.0000
2005 219.3750 222.7917 225.8750 228.7500 232.0000 235.2500 238.0833 241.0417 243.5417 245.5833 247.5000 250.5833
2006 254.7083 259.0000 262.7917 265.6250 268.6250 271.7500 274.7500 277.0000 280.4167 286.3750 293.0833 297.7917
2007 301.2500 305.0000 308.2500 310.7917 312.2500 313.0417 313.8333 313.7917 313.2917 312.7500 312.5833 314.1667
2008 317.6250 321.0417 323.3750 325.4583 327.7083 330.6667 334.3333 338.8333 343.2083 347.7500 352.9583 358.4583
2009 365.0417 371.8333 377.7917 382.9167 387.1250 391.8333 396.8750 401.5000 406.8333 412.5833 418.7500 425.5417
2010 431.8333 438.0417 443.6667 447.6250 451.2083 454.7083 457.9583 460.9167 464.5000 469.0000 473.8333 479.2083
2011 485.0417 491.7083 497.9167 502.6250 506.6667 510.3333 513.3750 515.6250 516.9583 517.3333 517.9167 519.2500
2012 521.5000 525.2083 527.4167 528.0417 528.9167 529.2500 530.4167 532.8333 536.7083 542.3333 549.4167 555.5833
2013 561.0833 567.5833 573.8333 579.4583 584.8750 591.7083 598.6250 604.2917 607.7917 612.7500 620.5417 627.8333
2014 635.8333 642.8750 647.9583 653.1667 657.5833 660.7500        NA        NA        NA        NA        NA        NA

$random
           Jan       Feb       Mar       Apr       May       Jun       Jul       Aug       Sep       Oct       Nov       Dec
2003        NA        NA        NA        NA        NA        NA 0.9506481 0.9518221 1.0000639 1.0039910 1.0054376 1.0126675
2004 0.9647844 1.0754101 1.0373675 1.0157964 0.9280175 0.9675514 0.9845058 0.9746837 1.0241013 0.9752383 0.9408307 1.0047590
2005 1.0131838 1.0638023 1.0889096 1.0154655 1.0505119 0.9442738 0.9480874 0.9409344 0.9902753 0.9931437 1.0239120 1.0251688
2006 1.0252260 1.0935233 1.0148756 0.9720482 0.9637914 1.0012869 0.9481812 0.9948719 0.9749743 1.0061501 1.0206586 1.0113837
2007 0.9958506 1.0140761 1.0574784 1.0779330 1.0397990 0.9689581 0.9538407 0.9870311 0.9921186 1.0168693 0.9987762 0.9903890
2008 0.9827493 0.9228019 1.0049631 0.9952340 1.0113376 0.9982505 1.0236085 0.9836599 0.9874179 0.9935999 0.9992553 0.9915759
2009 1.0148029 0.9879689 0.9778672 1.0005726 0.9889987 1.0040734 1.0396559 0.9840065 1.0041931 0.9997427 0.9826341 1.0122018
2010 1.0069112 0.9955482 0.9885231 0.9965182 0.9980568 1.0280793 1.0211159 1.0000182 1.0003755 0.9840004 0.9924600 0.9889606
2011 0.9941130 0.9640117 0.9879463 0.9869002 0.9920077 1.0331462 1.0259341 1.0310768 1.0235368 1.0099312 1.0240054 1.0022613
2012 0.9968131 0.9544180 0.9495433 0.9393970 0.9725987 1.0265608 1.0478829 1.0786369 0.9858723 0.9964441 0.9795610 0.9407027
2013 0.9827720 0.9499738 0.9777396 0.9742270 1.0179648 0.9966026 1.0366059 1.0533250 0.9971375 1.0009557 1.0125355 0.9999961
2014 1.0028600 0.9585322 0.8948529 1.0059743 1.0169825 1.0112834        NA        NA        NA        NA        NA        NA

$figure
 [1] 0.8233333 0.8438594 1.0123702 1.0805564 1.1857930 1.1209126 1.2360238 1.2301349 0.9619639 0.8364099 0.7655329 0.9031095

$type
[1] "multiplicative"
```

Notice, that we distinguished this is multiplicative. Also notice $trend is a steadily increasing number, not only year to year but month to month.

We can interpret $random as the % expected sales, and if reached 100% of our goal. For example, in January 2004, we can say that sales are 4% of what we expected, and in January 2005 they are 1% more than expected.



Above is the visual output of the numbers we say before. Notice the steadily increasing trend. Also, notice seasonality along with random and how they can be interpreted from ranges 0-1.x. This helps explain the timing of expected sales or if we met above or below our expected sales.

Forecasting

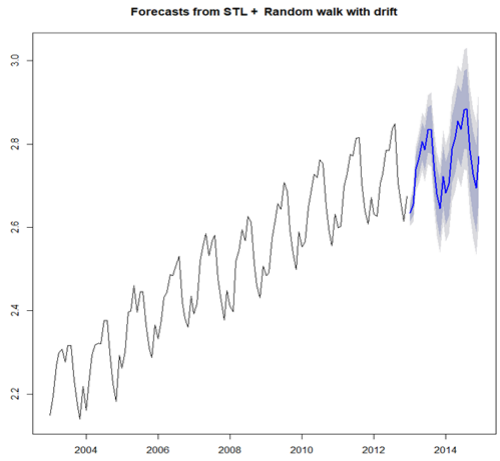
What good is all this information, if we can't present a forecast or see the accuracy of our forecast method? To create a forecast, we will split the data into an 80/20, training/test set. We will be testing the error of our forecast using the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

For our forecast we will be using a method known as **random walk with drift**. This model is under the assumption that at each point in time, the time series data is a random step away from its last recorded position. The mean value of these steps is zero. Any step size that does not have mean != 0, is then considered random walk with drift. That uses the prediction equation:

$$\hat{Y}t = Yt\text{-}1 + \alpha$$

We can simulate this method with the below code, in which we used our log model, and "h" is expressed in our test set of 24 months (20% of the data). Remember that the log expression changes that the multiplicative model to an additive model, and helps us with scaling the data.

```
TSDecompose_train_Log<-stl(log10(TS_Train),s.window='p')
TS_Train_stl<-forecast(TSDecompose_train_Log,method="rwdrift",h=24)
plot(TS_Train_stl)
```



We can see from the above that the forecast is within the "drift". This represents the mean of zero. We can also see that our forecast is not constant, as it picks up the trend and the seasonality. The standard error is represented in the grey portion.

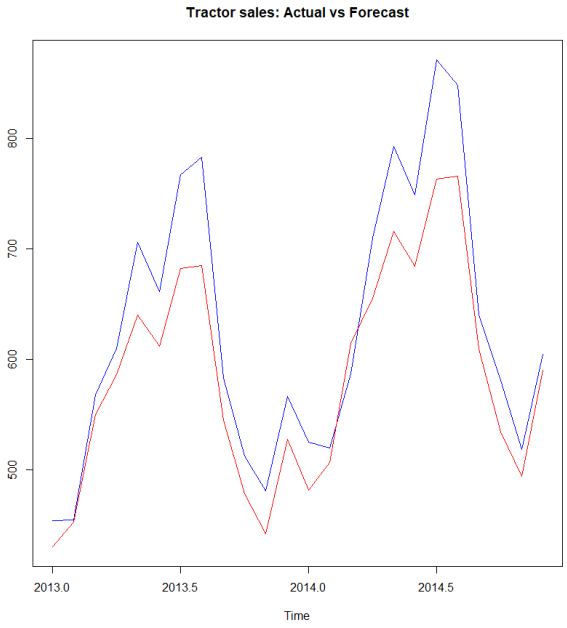
Checking the Forecast

Next, we will compare the forecast of 2013 & 2014 to the actual results of 2013 & 2014. This way we can determine the reliability of our model when applying it to real future dates.

```
Vec2<-10^(cbind(log10(TS_Test),as.data.frame(forecast(TSDecompose_train_Log,method="rwdrift",h=24
))[,1]))ts.plot(Vec2,col=c("blue","red"),main="Tractor sales: Actual vs Forecast")
```

```
Vec2
```

	log10(TS_Test)	as.data.frame(forecast(TSDecompose_train_Log, method = "rwdrift",
Jan 2013	454	430.7478
Feb 2013	455	453.0753
Mar 2013	568	549.9774
Apr 2013	610	586.4370
May 2013	706	640.3927
Jun 2013	661	611.9986
Jul 2013	767	682.3988
Aug 2013	783	684.8525
Sep 2013	583	544.7418
Oct 2013	513	478.2741
Nov 2013	481	442.1196
Dec 2013	567	527.7672
Jan 2014	525	481.6410
Feb 2014	520	506.6065
Mar 2014	587	614.9577
Apr 2014	710	655.7250
May 2014	793	716.0556
Jun 2014	749	684.3067
Jul 2014	871	763.0248
Aug 2014	848	765.7684
Sep 2014	640	609.1035
Oct 2014	581	534.7825
Nov 2014	519	494.3564
Dec 2014	605	590.1233



The “reliability” the forecast is dependent on the what the deciding parties think in an organization, but we can say overall it looks fairly accurate, but with some slight exaggerations during peak periods. In the above visualization, the red represents the actual data, while the blue represents the forecasted.

Forecast errors are the difference between the actual data and the forecast. It is important to note that the error != mistake. Instead, it provides the amount that is not predictable.

RMSE is a scale dependent error measure, which just means everything needs to be on the same scale. Since it squares, the deviations to keep positive and negative deviations from cancelling, it known as a absolute error measure. RMSE is calculated by squaring residuals, getting the average of those residuals, then taking the square root of the result. RMSE can be expressed in this formula:

$$\sqrt{\frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{n}}$$

MAPE is a relative error measure that utilizes absolute values to keep the positive and negative errors from cancelling one another out and the use of relative errors to allow for the ability to compare forecast accuracy between time series. MAPE is expressed in the following formula:

$$\frac{\sum_{t=1}^{n}\left|\frac{(Y_t - \hat{Y}_t)}{Y_t}(100)\right|}{n}$$

```
> RMSE2<-round(sqrt(sum(((Vec2[,1]-Vec2[,2])^2)/length(Vec2[,1]))),4)
> MAPE2<-round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)
> paste("Accuracy Measures:RMSE",RMSE2,"and MAPE",MAPE2)

[1] "Accuracy Measures:RMSE 53.5697 and MAPE 0.0687"
```

In the above output, we can see that our error measures produced about 54 for the RMSE and 6.9% for the MAPE.

What does this mean and how can we benefit from this?

The RMSE represents in the scale of the data, how far off we are in our prediction. So, in this case, we are above 54 tractors too short or too many on average in our prediction. It is up to key decision makers; how tight they want their confidence intervals to be. The surer you want your results to be, the less useful the results become. The MAPE informs us, that on average, we are about 6.9% away from the truth.

In my opinion, this model is useful because it the error is not sufficiently large. That being said. A company with thin margins, may not be willing to be 54 tractors off on their purchase order. It is dependent them to determine what impacts them more; stock outs or carrying cost?