Time series ARIMA model on tactor sales

Arinze Francis

2022-06-16

```
rm(list=ls())
options(scipen=999,digits=4)
rm
```

```
## function (..., list = character(), pos = -1, envir = as.environment(pos),
##     inherits = FALSE)
## {
##     dots <- match.call(expand.dots = FALSE)$...
##     if (length(dots) && !all(vapply(dots, function(x) is.symbol(x) ||
##         is.character(x), NA, USE.NAMES = FALSE)))
##         stop("... must contain names or character strings")
##     names <- vapply(dots, as.character, "")
##     if (length(names) == 0L)
##         names <- character()
##     list <- .Primitive("c")(list, names)
##     .Internal(remove(list, envir, inherits))
## }
## <bytecode: 0x0000000014cfa2a0>
## <environment: namespace:base>
```

Load R packages

```
library(readxl)
library(tidyquant)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Loading required package: PerformanceAnalytics
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
## Loading required package: quantmod
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## == Need to Learn tidyquant? ====================================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio O
ptimization with tidyquant!
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x dplyr::first()           masks xts::first()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x dplyr::last()            masks xts::last()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(lubridate)
library(xts)
library(quantmod)
library(tseries)
library(zoo)
library(ggplot2)
library(fpp2)
```

```
## Warning: package 'fpp2' was built under R version 4.1.3
```

```
## -- Attaching packages ---------------------------------------------- fpp2 2.4 --
```

```
## v forecast  8.16     v expsmooth 2.3
## v fma       2.4
```

```
## Warning: package 'forecast' was built under R version 4.1.3
```

```
## Warning: package 'fma' was built under R version 4.1.3
```

```
## Warning: package 'expsmooth' was built under R version 4.1.3
```

```
##
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
library(forecast)
```

Dataset loading

```
data <- read.csv("Tractor-Sales.csv")
data
```

```
##     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
## 7       Jul-03                    207
## 8       Aug-03                    207
## 9       Sep-03                    171
## 10      Oct-03                    150
## 11      Nov-03                    138
## 12      Dec-03                    165
## 13      Jan-04                    145
## 14      Feb-04                    168
## 15      Mar-04                    197
## 16      Apr-04                    208
## 17      May-04                    210
## 18      Jun-04                    209
## 19      Jul-04                    238
## 20      Aug-04                    238
## 21      Sep-04                    199
## 22      Oct-04                    168
## 23      Nov-04                    152
## 24      Dec-04                    196
## 25      Jan-05                    183
## 26      Feb-05                    200
## 27      Mar-05                    249
## 28      Apr-05                    251
## 29      May-05                    289
## 30      Jun-05                    249
## 31      Jul-05                    279
## 32      Aug-05                    279
## 33      Sep-05                    232
## 34      Oct-05                    204
## 35      Nov-05                    194
## 36      Dec-05                    232
## 37      Jan-06                    215
## 38      Feb-06                    239
## 39      Mar-06                    270
## 40      Apr-06                    279
## 41      May-06                    307
## 42      Jun-06                    305
## 43      Jul-06                    322
## 44      Aug-06                    339
## 45      Sep-06                    263
## 46      Oct-06                    241
## 47      Nov-06                    229
## 48      Dec-06                    272
## 49      Jan-07                    247
## 50      Feb-07                    261
## 51      Mar-07                    330
```

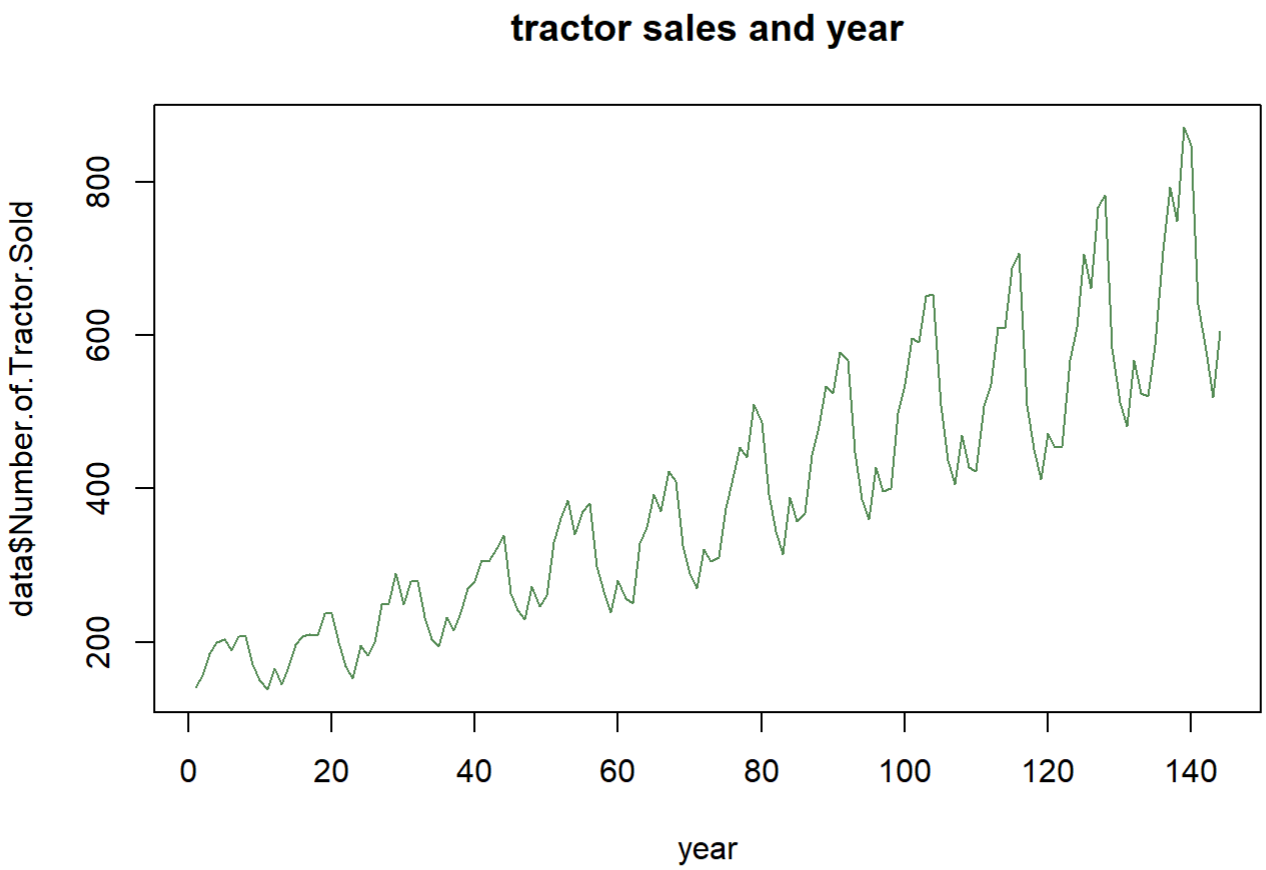
```
## 52     Apr-07                  362
## 53     May-07                  385
## 54     Jun-07                  340
## 55     Jul-07                  370
## 56     Aug-07                  381
## 57     Sep-07                  299
## 58     Oct-07                  266
## 59     Nov-07                  239
## 60     Dec-07                  281
## 61     Jan-08                  257
## 62     Feb-08                  250
## 63     Mar-08                  329
## 64     Apr-08                  350
## 65     May-08                  393
## 66     Jun-08                  370
## 67     Jul-08                  423
## 68     Aug-08                  410
## 69     Sep-08                  326
## 70     Oct-08                  289
## 71     Nov-08                  270
## 72     Dec-08                  321
## 73     Jan-09                  305
## 74     Feb-09                  310
## 75     Mar-09                  374
## 76     Apr-09                  414
## 77     May-09                  454
## 78     Jun-09                  441
## 79     Jul-09                  510
## 80     Aug-09                  486
## 81     Sep-09                  393
## 82     Oct-09                  345
## 83     Nov-09                  315
## 84     Dec-09                  389
## 85     Jan-10                  358
## 86     Feb-10                  368
## 87     Mar-10                  444
## 88     Apr-10                  482
## 89     May-10                  534
## 90     Jun-10                  524
## 91     Jul-10                  578
## 92     Aug-10                  567
## 93     Sep-10                  447
## 94     Oct-10                  386
## 95     Nov-10                  360
## 96     Dec-10                  428
## 97     Jan-11                  397
## 98     Feb-11                  400
## 99     Mar-11                  498
## 100    Apr-11                  536
## 101    May-11                  596
## 102    Jun-11                  591
## 103    Jul-11                  651
```

```
## 104     Aug-11                    654
## 105     Sep-11                    509
## 106     Oct-11                    437
## 107     Nov-11                    406
## 108     Dec-11                    470
## 109     Jan-12                    428
## 110     Feb-12                    423
## 111     Mar-12                    507
## 112     Apr-12                    536
## 113     May-12                    610
## 114     Jun-12                    609
## 115     Jul-12                    687
## 116     Aug-12                    707
## 117     Sep-12                    509
## 118     Oct-12                    452
## 119     Nov-12                    412
## 120     Dec-12                    472
## 121     Jan-13                    454
## 122     Feb-13                    455
## 123     Mar-13                    568
## 124     Apr-13                    610
## 125     May-13                    706
## 126     Jun-13                    661
## 127     Jul-13                    767
## 128     Aug-13                    783
## 129     Sep-13                    583
## 130     Oct-13                    513
## 131     Nov-13                    481
## 132     Dec-13                    567
## 133     Jan-14                    525
## 134     Feb-14                    520
## 135     Mar-14                    587
## 136     Apr-14                    710
## 137     May-14                    793
## 138     Jun-14                    749
## 139     Jul-14                    871
## 140     Aug-14                    848
## 141     Sep-14                    640
## 142     Oct-14                    581
## 143     Nov-14                    519
## 144     Dec-14                    605
```

```
str(data)
```

```
## 'data.frame':    144 obs. of  2 variables:
##  $ Month.Year            : chr  "Jan-03" "Feb-03" "Mar-03" "Apr-03" ...
##  $ Number.of.Tractor.Sold: int  141 157 185 199 203 189 207 207 171 150 ...
```

```
plot(data$Number.of.Tractor.Sold, xlab="year", type ="l", col="palegreen4", main="tractor sales
 and year")
```



Converting to time series class and plotting the time series data

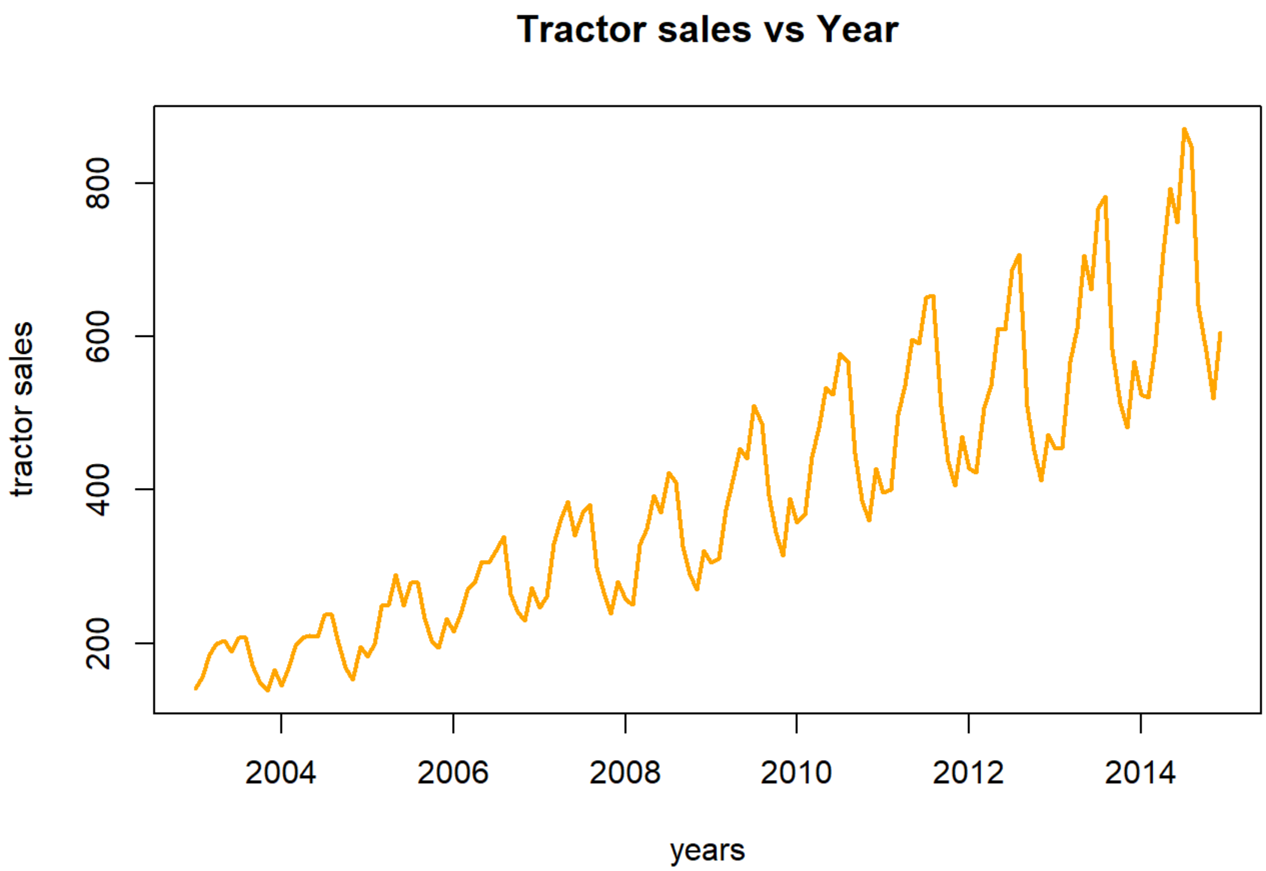
```
data_ts <- ts(data$Number.of.Tractor.Sold, start = c(2003,1), frequency = 12)
data_ts
```

```
##      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
```

```
class(data_ts)
```

```
## [1] "ts"
```

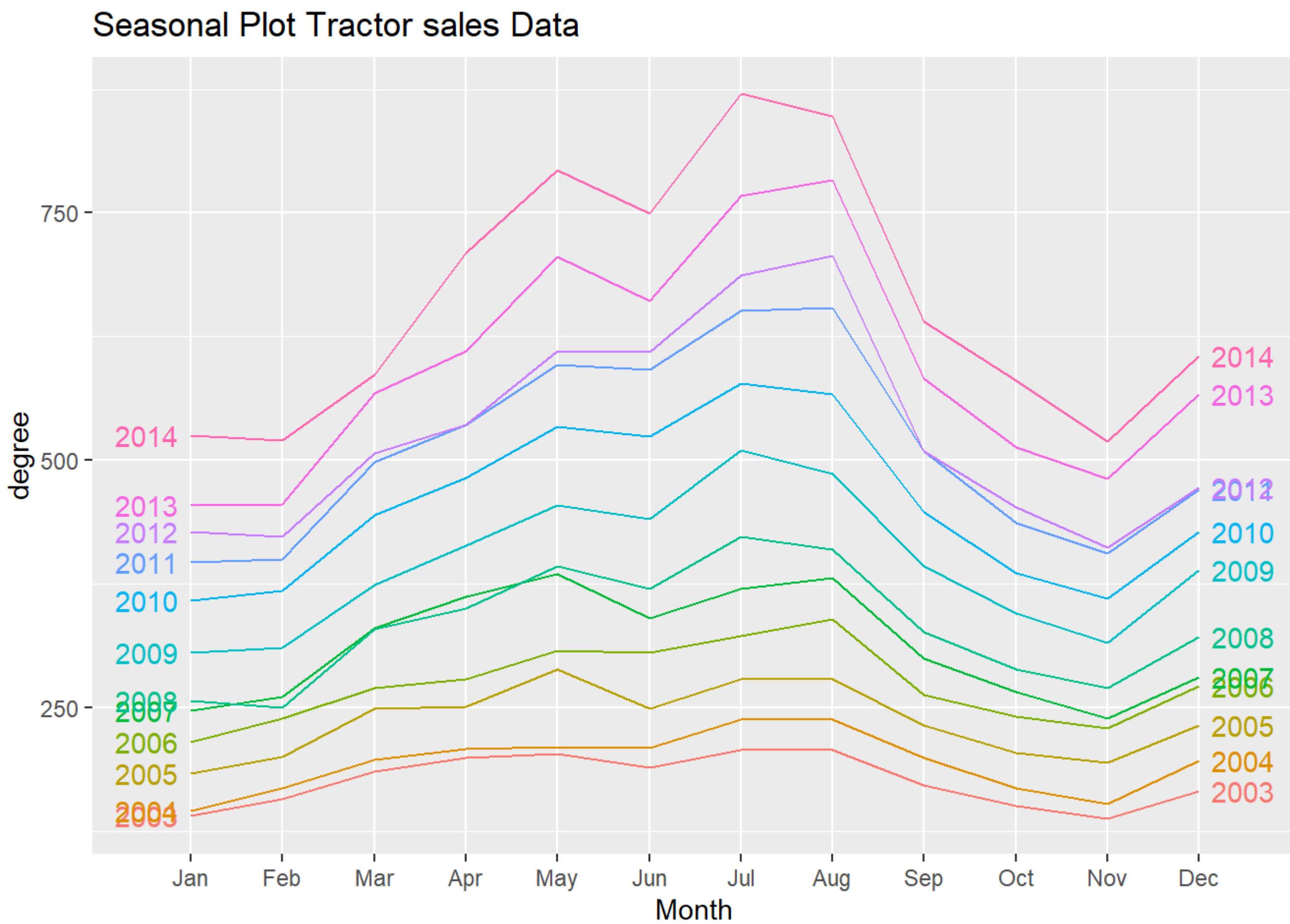
```
plot(data_ts, xlab="years", ylab="tractor sales", main="Tractor sales vs Year",col="orange",type
= "l", lwd=2)
```



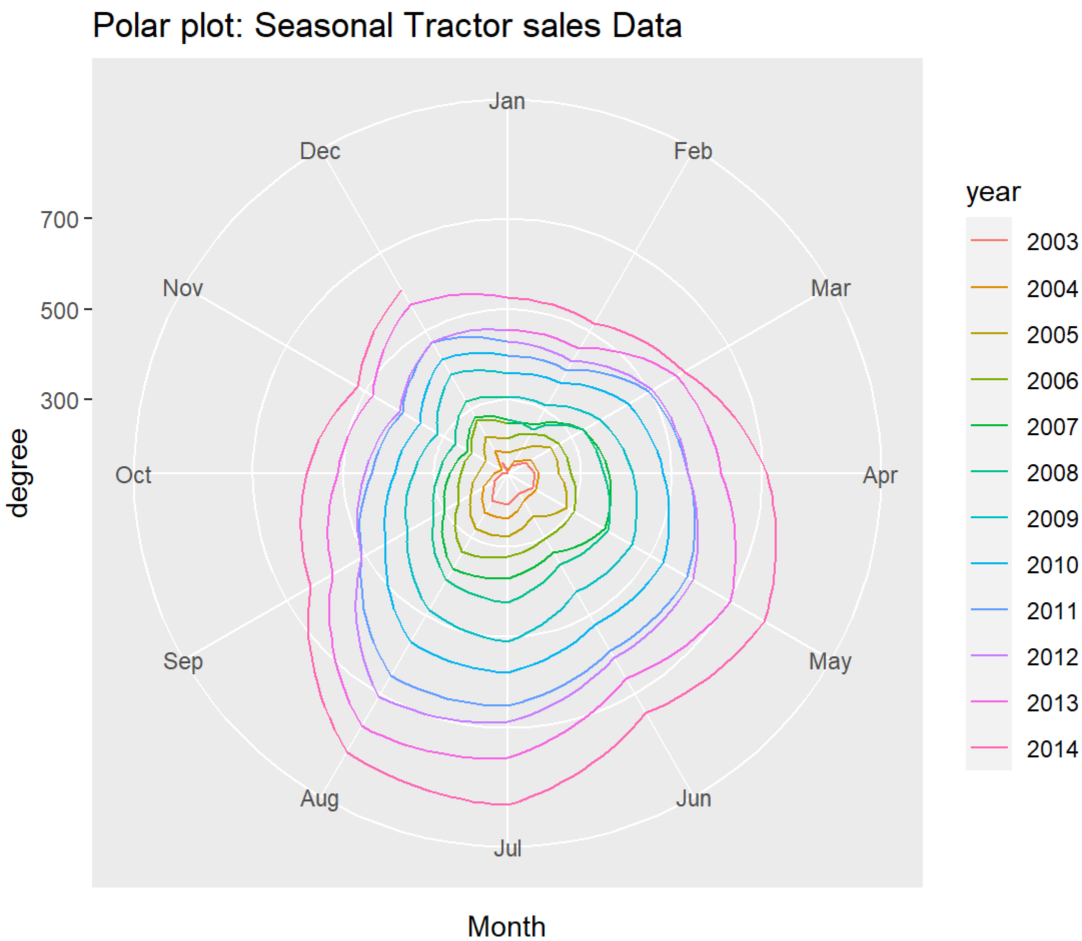

```
# Observation of the plot:
# 1. Values of the data are stored in correct order and no missing data.
# 2. There is an upward trend. On the average, tractor sales is going up. Sales are increasing i
n numbers, implying presence of trend component.
# 3. Intra-year stable fluctuations are indicative of seasonal components. As trend increases, f
luctuations are also increasing. Indicative of multiplicative seasonality.
```

to get the seasonality better

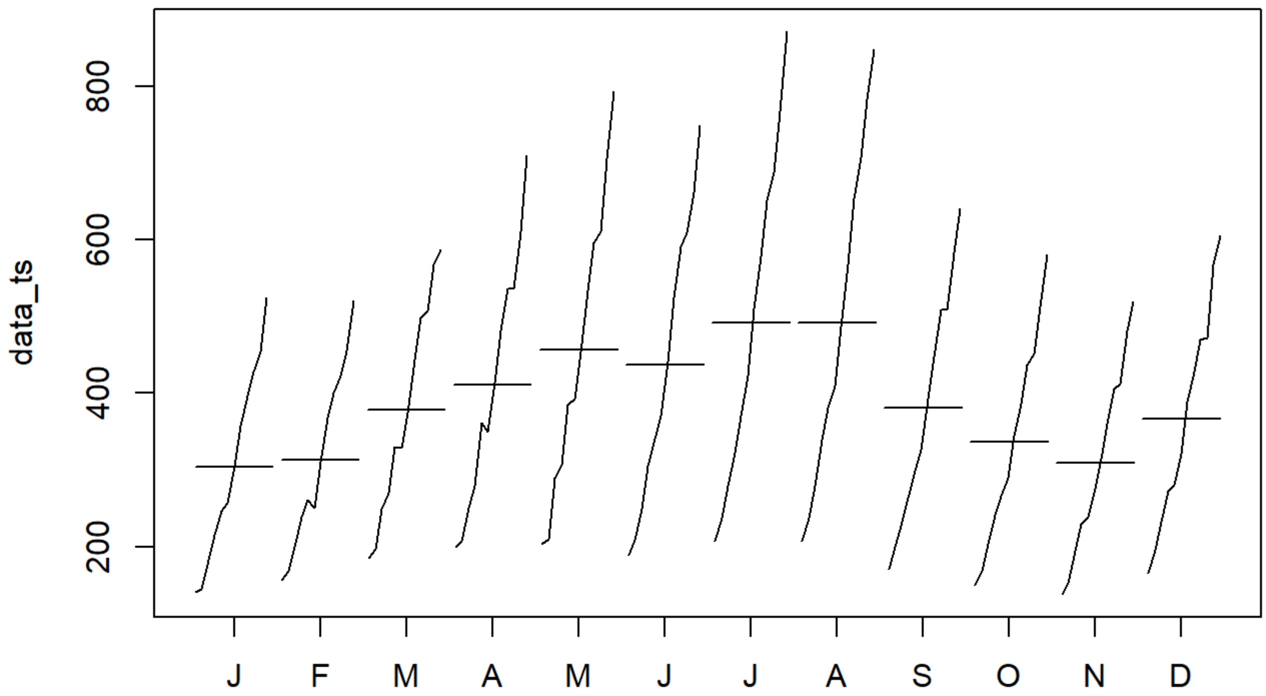
```
ggseasonplot(data_ts, year.labels = T, year.labels.left = T) +ylab("degree") +ggtitle("Seasonal
 Plot Tractor sales Data")
```




```
# Observation:1) as the year goes by, sales increases - meaning trend, 2) There is a common seas
onality pattern across years but not identical ( a bump in April 2014).

ggseasonplot(data_ts, polar = T) +ylab("degree") +ggtitle("Polar plot: Seasonal Tractor sales Da
ta") # a polar visualization
```




```
# Note; if the plot is circular, then there is no seasonality. Again, if it circular but the cen
ter is shifting towards the wrong place, then there is a seasonality pattern the same across the
whole years.

monthplot(data_ts)
```




```
# Average sales are higher in the month of July and August. There were some irregularities in th
e month of April and Febrauary (the bump).
```

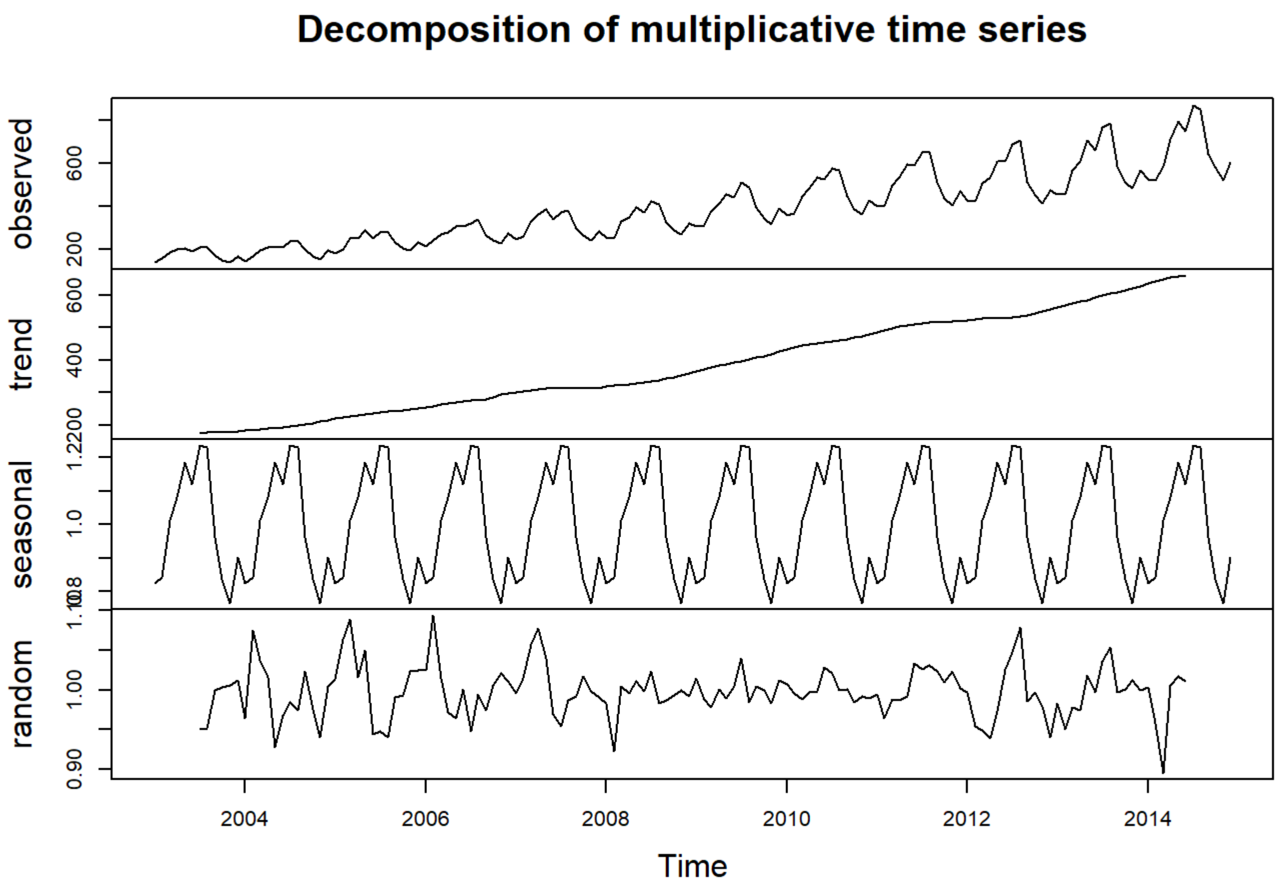
Decomposition of plot: Mutiplicative Seasonal correction/adjustment

```
data_decompose <- decompose(data_ts, type = "multiplicative")
data_decompose
```

```
## $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
## 2003 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2004 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2005 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2006 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2007 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2008 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2009 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2010 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2011 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2012 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2013 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## 2014 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
##         Nov    Dec
## 2003 0.7655 0.9031
## 2004 0.7655 0.9031
## 2005 0.7655 0.9031
## 2006 0.7655 0.9031
## 2007 0.7655 0.9031
## 2008 0.7655 0.9031
## 2009 0.7655 0.9031
## 2010 0.7655 0.9031
## 2011 0.7655 0.9031
## 2012 0.7655 0.9031
## 2013 0.7655 0.9031
## 2014 0.7655 0.9031
## 
## $trend
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 2003    NA    NA    NA    NA    NA    NA 176.2 176.8 177.8 178.6 179.3 180.4
## 2004 182.5 185.1 187.6 189.5 190.8 192.7 195.6 198.5 202.0 206.0 211.0 216.0
## 2005 219.4 222.8 225.9 228.7 232.0 235.2 238.1 241.0 243.5 245.6 247.5 250.6
## 2006 254.7 259.0 262.8 265.6 268.6 271.8 274.8 277.0 280.4 286.4 293.1 297.8
## 2007 301.3 305.0 308.2 310.8 312.2 313.0 313.8 313.8 313.3 312.7 312.6 314.2
## 2008 317.6 321.0 323.4 325.5 327.7 330.7 334.3 338.8 343.2 347.7 353.0 358.5
## 2009 365.0 371.8 377.8 382.9 387.1 391.8 396.9 401.5 406.8 412.6 418.8 425.5
```

```
## 2010 431.8 438.0 443.7 447.6 451.2 454.7 458.0 460.9 464.5 469.0 473.8 479.2
## 2011 485.0 491.7 497.9 502.6 506.7 510.3 513.4 515.6 517.0 517.3 517.9 519.2
## 2012 521.5 525.2 527.4 528.0 528.9 529.2 530.4 532.8 536.7 542.3 549.4 555.6
## 2013 561.1 567.6 573.8 579.5 584.9 591.7 598.6 604.3 607.8 612.8 620.5 627.8
## 2014 635.8 642.9 648.0 653.2 657.6 660.8    NA    NA    NA    NA    NA    NA
##
## $random
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2003     NA     NA     NA     NA     NA     NA 0.9506 0.9518 1.0001 1.0040
## 2004 0.9648 1.0754 1.0374 1.0158 0.9280 0.9676 0.9845 0.9747 1.0241 0.9752
## 2005 1.0132 1.0638 1.0889 1.0155 1.0505 0.9443 0.9481 0.9409 0.9903 0.9931
## 2006 1.0252 1.0935 1.0149 0.9720 0.9638 1.0013 0.9482 0.9949 0.9750 1.0062
## 2007 0.9959 1.0141 1.0575 1.0779 1.0398 0.9690 0.9538 0.9870 0.9921 1.0169
## 2008 0.9827 0.9228 1.0050 0.9952 1.0113 0.9983 1.0236 0.9837 0.9874 0.9936
## 2009 1.0148 0.9880 0.9779 1.0006 0.9890 1.0041 1.0397 0.9840 1.0042 0.9997
## 2010 1.0069 0.9955 0.9885 0.9965 0.9981 1.0281 1.0211 1.0000 1.0004 0.9840
## 2011 0.9941 0.9640 0.9879 0.9869 0.9920 1.0331 1.0259 1.0311 1.0235 1.0099
## 2012 0.9968 0.9544 0.9495 0.9394 0.9726 1.0266 1.0479 1.0786 0.9859 0.9964
## 2013 0.9828 0.9500 0.9777 0.9742 1.0180 0.9966 1.0366 1.0533 0.9971 1.0010
## 2014 1.0029 0.9585 0.8949 1.0060 1.0170 1.0113     NA     NA     NA     NA
##         Nov    Dec
## 2003 1.0054 1.0127
## 2004 0.9408 1.0048
## 2005 1.0239 1.0252
## 2006 1.0207 1.0114
## 2007 0.9988 0.9904
## 2008 0.9993 0.9916
## 2009 0.9826 1.0122
## 2010 0.9925 0.9890
## 2011 1.0240 1.0023
## 2012 0.9796 0.9407
## 2013 1.0125 1.0000
## 2014     NA     NA
##
## $figure
##  [1] 0.8233 0.8439 1.0124 1.0806 1.1858 1.1209 1.2360 1.2301 0.9620 0.8364
## [11] 0.7655 0.9031
##
## $type
## [1] "multiplicative"
##
## attr(,"class")
## [1] "decomposed.ts"
```

```
# On the seasonal part in January for all years, you are going to sell 82% of your annual trend
 (and 18% less) and etc. In July, you sell about 23% more, in May, 18% more.

# On the random part, 2004, Jan was about 4% left than where it should be after accounting for t
rend and seasonality.Jan 2002, about 1% more than my trend and seasonality forecast.

plot(data_decompose)
```




```
# the trend is incresing though there is a flattening in 2007 and 2008, 2011 ans 2012
# The seasonal part is repeating (Note the .008 and 1 unit, maybe multiplicative)
# On random: My unpredictable error is about 10% (o.90). In the future, i don't know what the nu
mber will be, but my best guess is in the middle (1).
```

Splittig data into training and test sets and test the last 2 years

```
data_train <- window(data_ts, start=c(2003,1),end=c(2012,12), freq=12)
data_train
```

```
##      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
```

```
data_test <- window(data_ts, start=c(2013,1), freq=12)
data_test
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 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
```

```
autoplot(data_train, series = "Train") + autolayer(data_test, series = "Test") + ggtitle("Tracto
r train and test set") +xlab("year") +ylab("sales")+guides(colour=guide_legend(title = "Forecas
t"))
```

Data Forecasting Methods using Random walk Drift

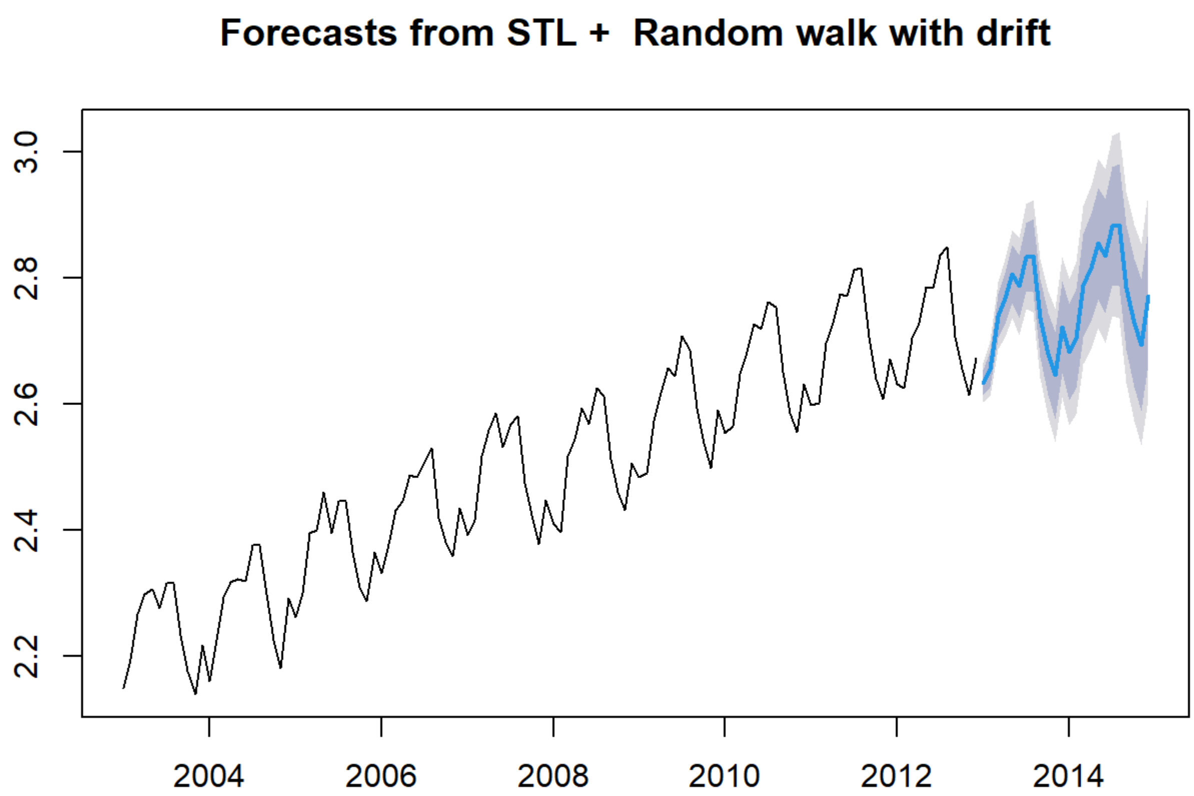
```
# Random Walk drift method forecasts next period value as per the amount of change over time (ca
lled the drift). It evaluates the average change seen in past data.
data_decompose_train_log <-stl(log10(data_train),s.window = "p")
data_decompose_train_log
```

```
##  Call:
##  stl(x = log10(data_train), s.window = "p")
##
## Components
##          seasonal trend   remainder
## Jan 2003 -0.08068 2.238 -0.00762482
## Feb 2003 -0.06278 2.238  0.02057682
## Mar 2003  0.01735 2.239  0.01114276
## Apr 2003  0.04119 2.240  0.01802265
## May 2003  0.07537 2.241 -0.00848435
## Jun 2003  0.05163 2.242 -0.01702980
## Jul 2003  0.09488 2.243 -0.02201629
## Aug 2003  0.09240 2.244 -0.02070614
## Sep 2003 -0.01105 2.245 -0.00140518
## Oct 2003 -0.07161 2.248  0.00017704
## Nov 2003 -0.10978 2.250  0.00007512
## Dec 2003 -0.03692 2.253  0.00105012
## Jan 2004 -0.08068 2.257 -0.01507409
## Feb 2004 -0.06278 2.262  0.02657240
## Mar 2004  0.01735 2.266  0.01120732
## Apr 2004  0.04119 2.270  0.00643020
## May 2004  0.07537 2.275 -0.02813658
## Jun 2004  0.05163 2.280 -0.01195755
## Jul 2004  0.09488 2.286 -0.00425867
## Aug 2004  0.09240 2.293 -0.00908068
## Sep 2004 -0.01105 2.301  0.00933848
## Oct 2004 -0.07161 2.309 -0.01240664
## Nov 2004 -0.10978 2.318 -0.02645047
## Dec 2004 -0.03692 2.326  0.00298629
## Jan 2005 -0.08068 2.334  0.00882716
## Feb 2005 -0.06278 2.341  0.02296058
## Mar 2005  0.01735 2.347  0.03145720
## Apr 2005  0.04119 2.353  0.00499145
## May 2005  0.07537 2.360  0.02592693
## Jun 2005  0.05163 2.365 -0.02079133
## Jul 2005  0.09488 2.371 -0.02039005
## Aug 2005  0.09240 2.376 -0.02295412
## Sep 2005 -0.01105 2.381 -0.00466937
## Oct 2005 -0.07161 2.386 -0.00495206
## Nov 2005 -0.10978 2.391  0.00641781
## Dec 2005 -0.03692 2.397  0.00571394
## Jan 2006 -0.08068 2.402  0.01089588
## Feb 2006 -0.06278 2.408  0.03344572
## Mar 2006  0.01735 2.413  0.00077475
## Apr 2006  0.04119 2.418 -0.01388048
## May 2006  0.07537 2.423 -0.01159048
## Jun 2006  0.05163 2.428  0.00424079
## Jul 2006  0.09488 2.433 -0.02051719
## Aug 2006  0.09240 2.440 -0.00171108
## Sep 2006 -0.01105 2.446 -0.01452771
## Oct 2006 -0.07161 2.453  0.00064845
## Nov 2006 -0.10978 2.460  0.00920494
```

```
## Dec 2006 -0.03692 2.467  0.00449185
## Jan 2007 -0.08068 2.474 -0.00020503
## Feb 2007 -0.06278 2.478  0.00119945
## Mar 2007  0.01735 2.483  0.01830658
## Apr 2007  0.04119 2.486  0.03186048
## May 2007  0.07537 2.488  0.02162340
## Jun 2007  0.05163 2.489 -0.00958070
## Jul 2007  0.09488 2.490 -0.01706399
## Aug 2007  0.09240 2.490 -0.00135576
## Sep 2007 -0.01105 2.489 -0.00265953
## Oct 2007 -0.07161 2.489  0.00712614
## Nov 2007 -0.10978 2.489 -0.00115920
## Dec 2007 -0.03692 2.492 -0.00591540
## Jan 2008 -0.08068 2.494 -0.00313163
## Feb 2008 -0.06278 2.497 -0.03647819
## Mar 2008  0.01735 2.501 -0.00080298
## Apr 2008  0.04119 2.505 -0.00180814
## May 2008  0.07537 2.509  0.01029055
## Jun 2008  0.05163 2.514  0.00252182
## Jul 2008  0.09488 2.519  0.01209874
## Aug 2008  0.09240 2.525 -0.00482868
## Sep 2008 -0.01105 2.531 -0.00680081
## Oct 2008 -0.07161 2.537 -0.00417796
## Nov 2008 -0.10978 2.542 -0.00114625
## Dec 2008 -0.03692 2.548 -0.00494700
## Jan 2009 -0.08068 2.554  0.01052737
## Feb 2009 -0.06278 2.561 -0.00695742
## Mar 2009  0.01735 2.568 -0.01222140
## Apr 2009  0.04119 2.574  0.00156729
## May 2009  0.07537 2.581  0.00093425
## Jun 2009  0.05163 2.587  0.00575808
## Jul 2009  0.09488 2.593  0.01934694
## Aug 2009  0.09240 2.599 -0.00514290
## Sep 2009 -0.01105 2.605  0.00002249
## Oct 2009 -0.07161 2.611 -0.00165600
## Nov 2009 -0.10978 2.617 -0.00864639
## Dec 2009 -0.03692 2.622  0.00465641
## Jan 2010 -0.08068 2.628  0.00687517
## Feb 2010 -0.06278 2.633 -0.00437113
## Mar 2010  0.01735 2.638 -0.00827431
## Apr 2010  0.04119 2.643 -0.00111628
## May 2010  0.07537 2.648  0.00452339
## Jun 2010  0.05163 2.652  0.01610716
## Jul 2010  0.09488 2.656  0.01151363
## Aug 2010  0.09240 2.659  0.00219815
## Sep 2010 -0.01105 2.662 -0.00108304
## Oct 2010 -0.07161 2.666 -0.00768456
## Nov 2010 -0.10978 2.669 -0.00322749
## Dec 2010 -0.03692 2.673 -0.00502416
## Jan 2011 -0.08068 2.677  0.00200622
## Feb 2011 -0.06278 2.682 -0.01759237
## Mar 2011  0.01735 2.687 -0.00751832
```

```
## Apr 2011  0.04119 2.692 -0.00435172
## May 2011  0.07537 2.697  0.00261241
## Jun 2011  0.05163 2.701  0.01882191
## Jul 2011  0.09488 2.705  0.01369987
## Aug 2011  0.09240 2.707  0.01626044
## Sep 2011 -0.01105 2.709  0.00892933
## Oct 2011 -0.07161 2.709  0.00307414
## Nov 2011 -0.10978 2.709  0.00912307
## Dec 2011 -0.03692 2.709 -0.00047696
## Jan 2012 -0.08068 2.710  0.00231869
## Feb 2012 -0.06278 2.711 -0.02201713
## Mar 2012  0.01735 2.712 -0.02480919
## Apr 2012  0.04119 2.714 -0.02646553
## May 2012  0.07537 2.716 -0.00646216
## Jun 2012  0.05163 2.718  0.01489027
## Jul 2012  0.09488 2.720  0.02231087
## Aug 2012  0.09240 2.722  0.03524599
## Sep 2012 -0.01105 2.724 -0.00601816
## Oct 2012 -0.07161 2.726  0.00081446
## Nov 2012 -0.10978 2.728 -0.00339160
## Dec 2012 -0.03692 2.730 -0.01929390
```

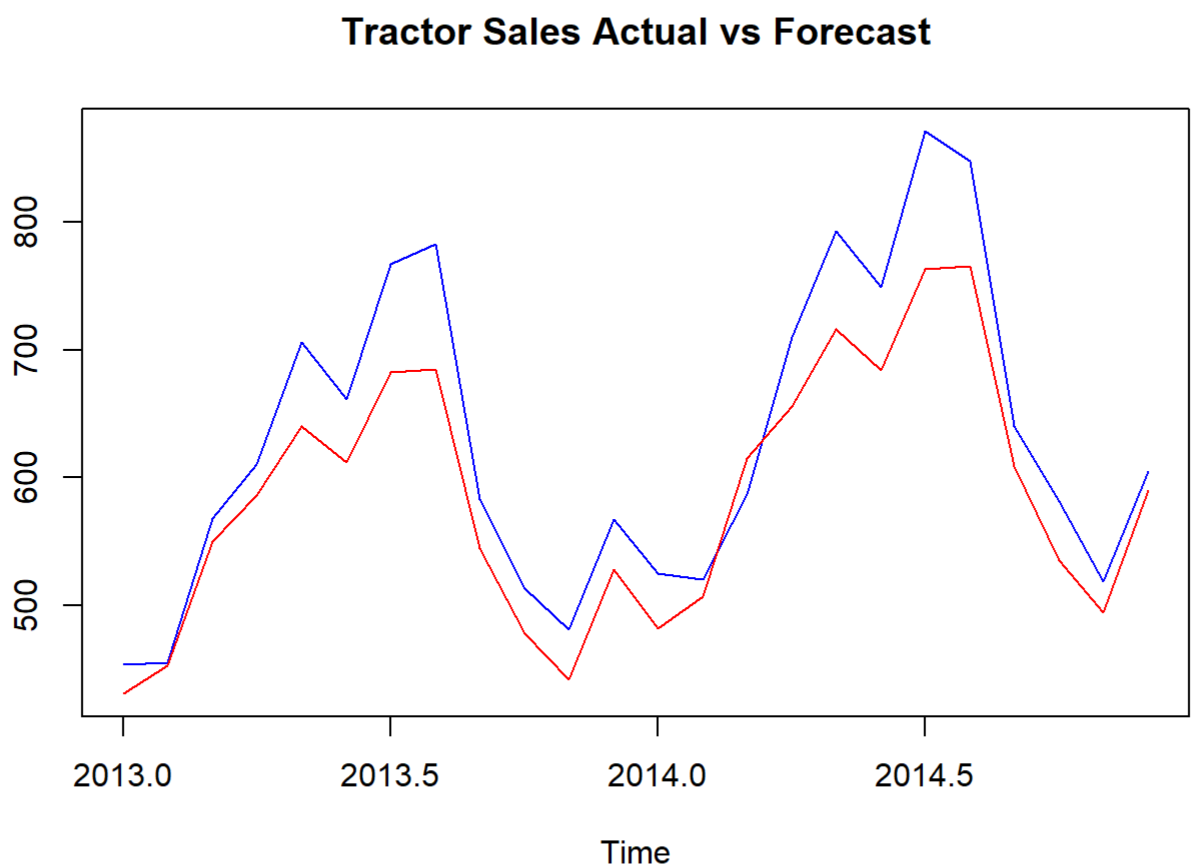
```
# seasonal component is the same, trend is increasing, remainder remains unpredictable)


# Data Forecast with Random walk drift
data_train_stl <- forecast(data_decompose_train_log, method = "rwdrift",h=24) #h=24 means how lo
ng which is 2 years (the test set years), 24months, Lower 80% and higher 95% points etc
plot(data_train_stl) #forecast on the log scale
```




```
## Accuracy Measure using Random walk drift

Vec_2 <- 10^(cbind(log10(data_test), as.data.frame(forecast(data_decompose_train_log, method =
"rwdrift",h=24))[,1]))
Vec_2
```

```
##          log10(data_test)
## Jan 2013              454
## Feb 2013              455
## Mar 2013              568
## Apr 2013              610
## May 2013              706
## Jun 2013              661
## Jul 2013              767
## Aug 2013              783
## Sep 2013              583
## Oct 2013              513
## Nov 2013              481
## Dec 2013              567
## Jan 2014              525
## Feb 2014              520
## Mar 2014              587
## Apr 2014              710
## May 2014              793
## Jun 2014              749
## Jul 2014              871
## Aug 2014              848
## Sep 2014              640
## Oct 2014              581
## Nov 2014              519
## Dec 2014              605
##          as.data.frame(forecast(data_decompose_train_log, method = "rwdrift",
## Jan 2013                                                             430.7
## Feb 2013                                                             453.1
## Mar 2013                                                             550.0
## Apr 2013                                                             586.4
## May 2013                                                             640.4
## Jun 2013                                                             612.0
## Jul 2013                                                             682.4
## Aug 2013                                                             684.9
## Sep 2013                                                             544.7
## Oct 2013                                                             478.3
## Nov 2013                                                             442.1
## Dec 2013                                                             527.8
## Jan 2014                                                             481.6
## Feb 2014                                                             506.6
## Mar 2014                                                             615.0
## Apr 2014                                                             655.7
## May 2014                                                             716.1
## Jun 2014                                                             684.3
## Jul 2014                                                             763.0
## Aug 2014                                                             765.8
## Sep 2014                                                             609.1
## Oct 2014                                                             534.8
## Nov 2014                                                             494.4
## Dec 2014                                                             590.1
```

```
# I am off by 24 units,
# 430 (forecast) +- 1.96 * 53 (RMSE)

ts.plot(Vec_2, col=c("blue", "red"), main = "Tractor Sales Actual vs Forecast")
```



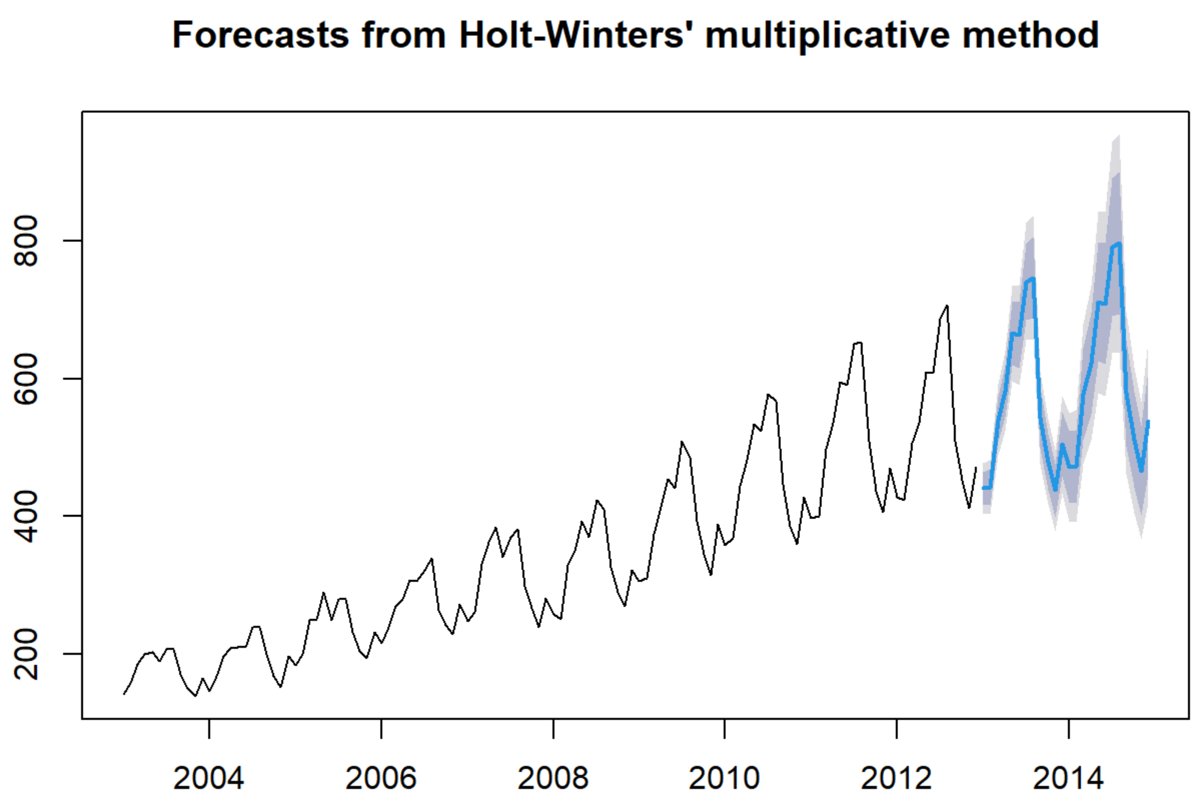

```
#test is blue, forecast is red

# There was slight underprediction. There was something that lifted my sales a little bit above
 the historic trend. Something that was not explained by the trend of the past but I am not that
far off as i have picked up a trend and seasonality.


# how good the forecast is?
RMSE2 <- round(sqrt(sum(((Vec_2[,1]-Vec_2[,2])^2)/length(Vec_2[,1]))),4)# root mean square error
ie standard deviation forecast
MAPE2 <- round(mean(abs(Vec_2[,1]-Vec_2[,2])/Vec_2[,1]),4)# mean absolute percentage


paste("Accuracy measures: RMSE:", RMSE2, "and MAPE:", MAPE2 )
```

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

```
#Interpretion from the ts.plot: I am on average of about 6.9% away from the truth.
# RMSE: whatever i forecast, + or - 53.56 of that, I am above 68% of covering.
```

Data forecasting methods using Holt's Winter

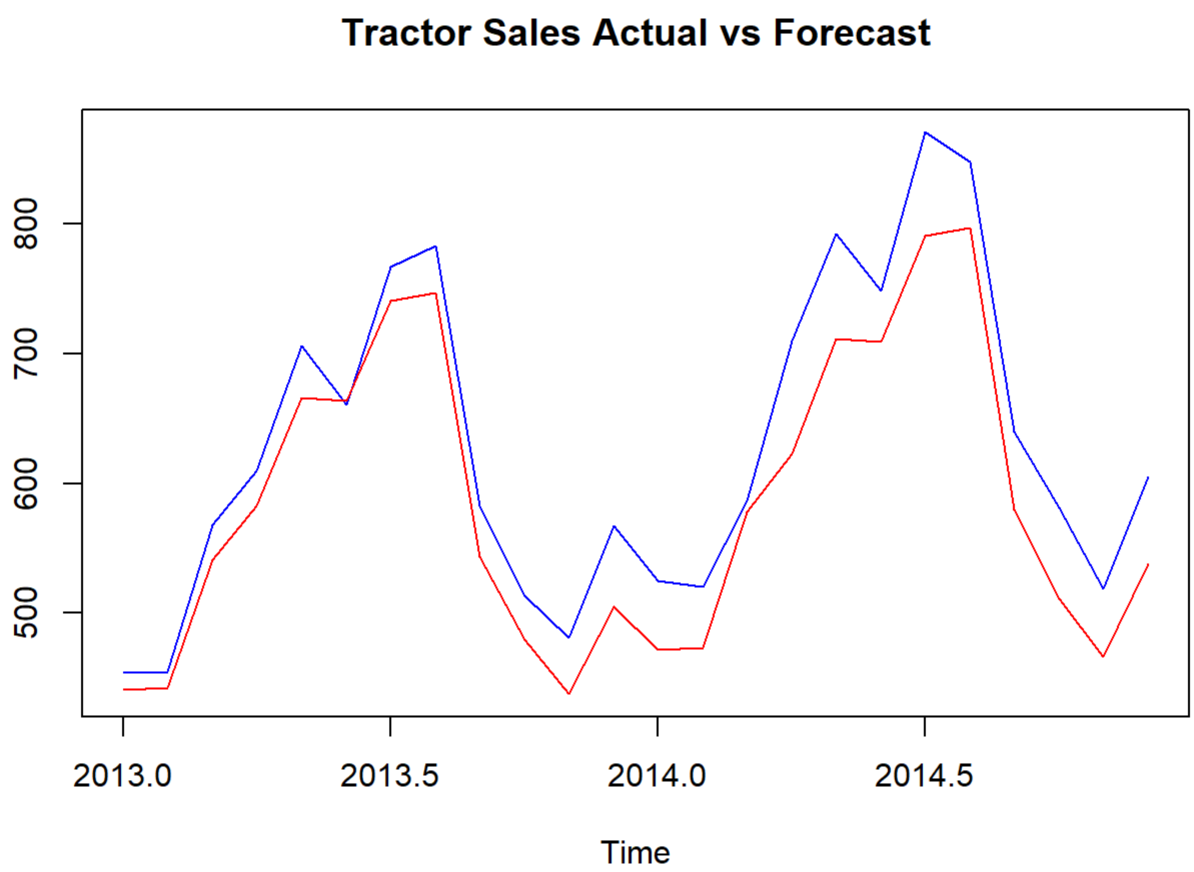
```
data_train_hw <- hw(data_train, seasonal = "multiplicative")


plot(forecast(data_train_hw, h=24))
```




```
# Accuracy measures using HW

vec <- cbind(data_test,as.data.frame(forecast(data_train_hw, h=24))[,1])
vec
```

```
##          data_test as.data.frame(forecast(data_train_hw, h = 24))[, 1]
## Jan 2013       454                                                 440.9
## Feb 2013       455                                                 442.2
## Mar 2013       568                                                 540.7
## Apr 2013       610                                                 582.8
## May 2013       706                                                 666.0
## Jun 2013       661                                                 664.0
## Jul 2013       767                                                 741.2
## Aug 2013       783                                                 747.2
## Sep 2013       583                                                 543.4
## Oct 2013       513                                                 479.5
## Nov 2013       481                                                 437.6
## Dec 2013       567                                                 505.0
## Jan 2014       525                                                 471.7
## Feb 2014       520                                                 472.9
## Mar 2014       587                                                 578.1
## Apr 2014       710                                                 622.8
## May 2014       793                                                 711.5
## Jun 2014       749                                                 709.1
## Jul 2014       871                                                 791.3
## Aug 2014       848                                                 797.4
## Sep 2014       640                                                 579.7
## Oct 2014       581                                                 511.3
## Nov 2014       519                                                 466.5
## Dec 2014       605                                                 538.1
```

```
ts.plot(vec, col=c("blue", "red"), main = "Tractor Sales Actual vs Forecast")
```



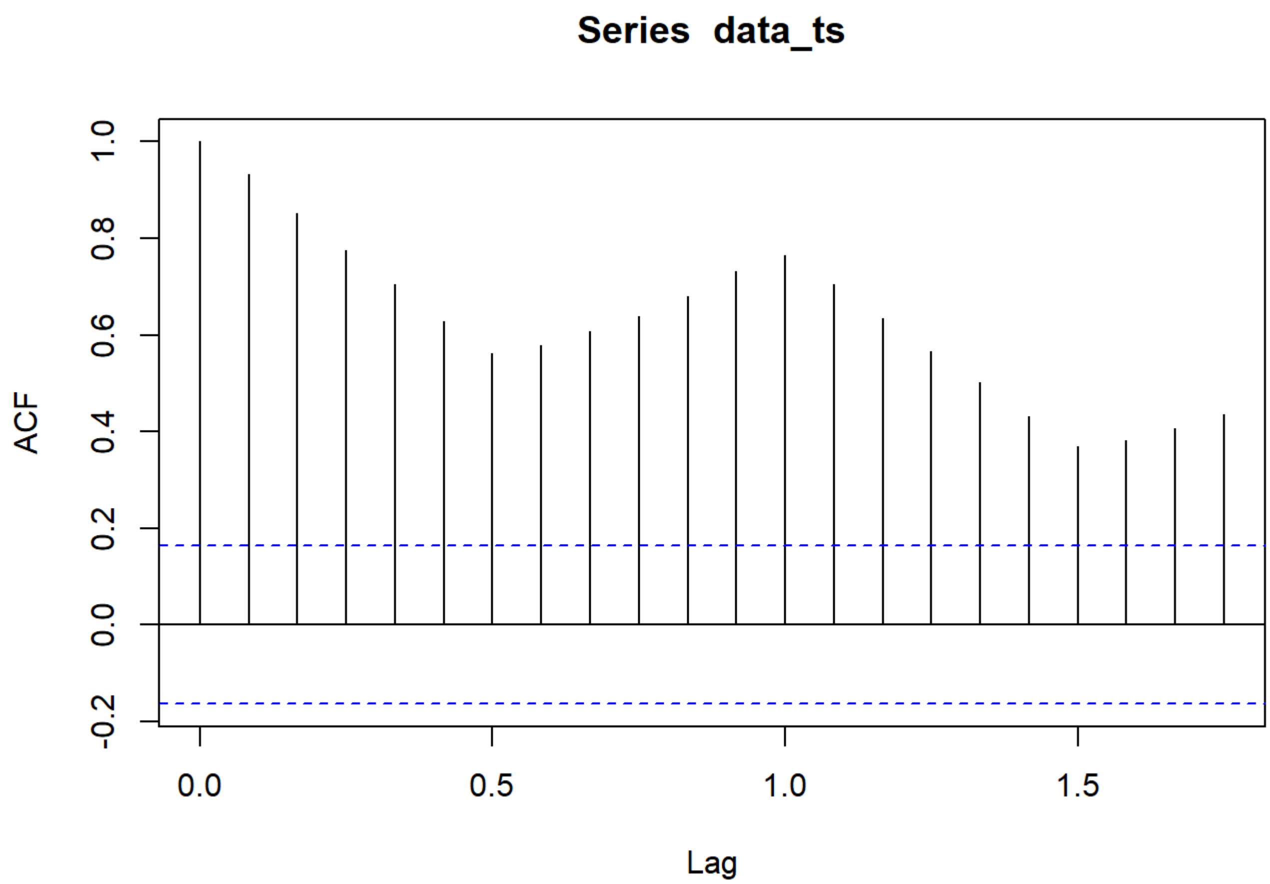

```
# still under predicted.


# how good the forecast is?

RMSE1 <- round(sqrt(sum(((vec[,1]-vec[,2])^2)/length(vec[,1]))),4)
MAPE1 <- round(mean(abs(vec[,1]-vec[,2])/vec[,1]),4)

paste("Accuracy measures: RMSE:", RMSE1, "and MAPE:", MAPE1 )
```

```
## [1] "Accuracy measures: RMSE: 49.7553 and MAPE: 0.0703"
```

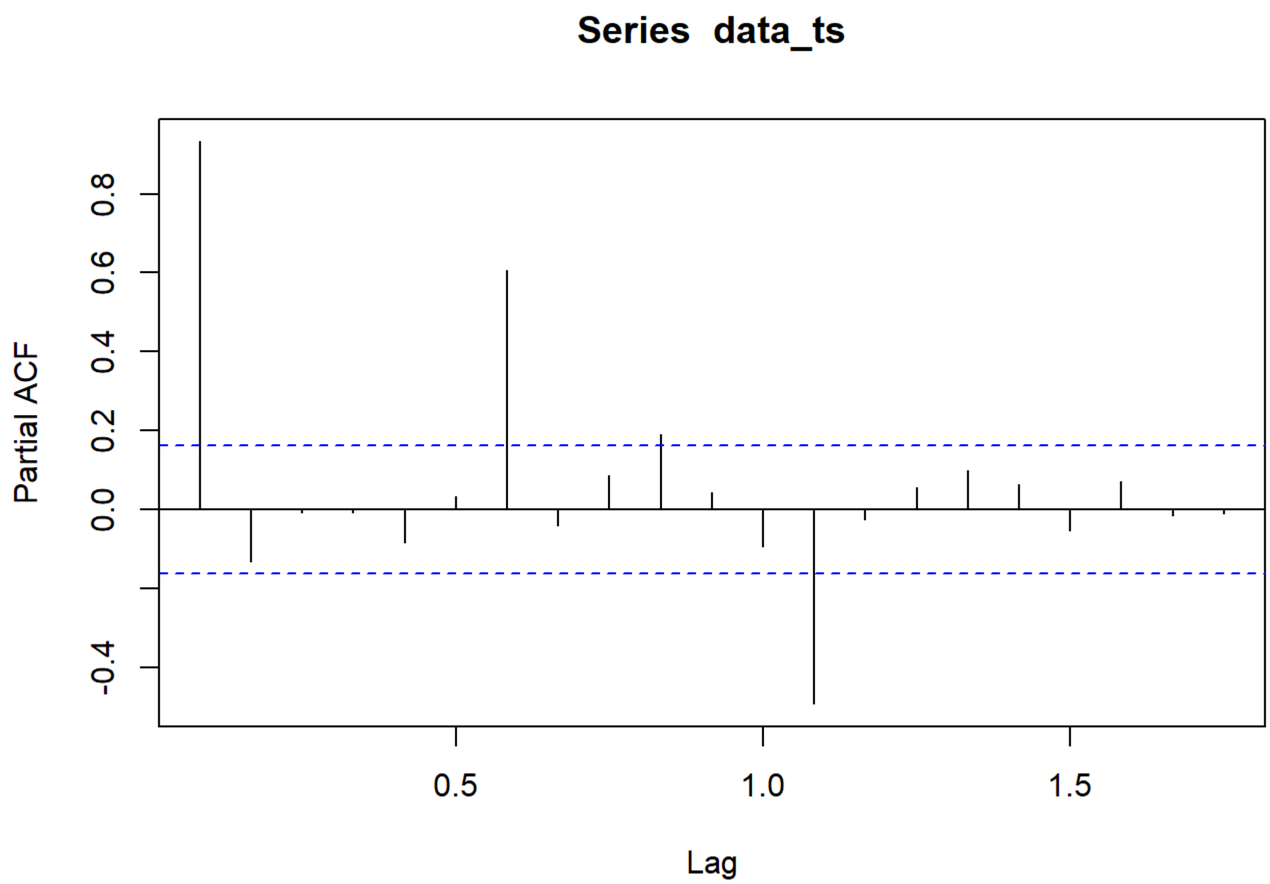
Data Forecasting Using ARIMA methods

To check for stationarity

```
acf(data_ts)
```




```
# it is not stationary (auto correlation because the spikes cross above the blue lines)

pacf(data_ts)
```




```
# partial okay as the spikes are not much

adf.test(data_ts) #p-value should be less than 0.05
```

```
## Warning in adf.test(data_ts): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts
## Dickey-Fuller = -13, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
# Converting non-stationary data to stationary data

new_arima <- auto.arima(data_ts, d=1, D=1,stepwise = F, approximation = F, trace = T) #d=1 means seasonal and trend time series
```

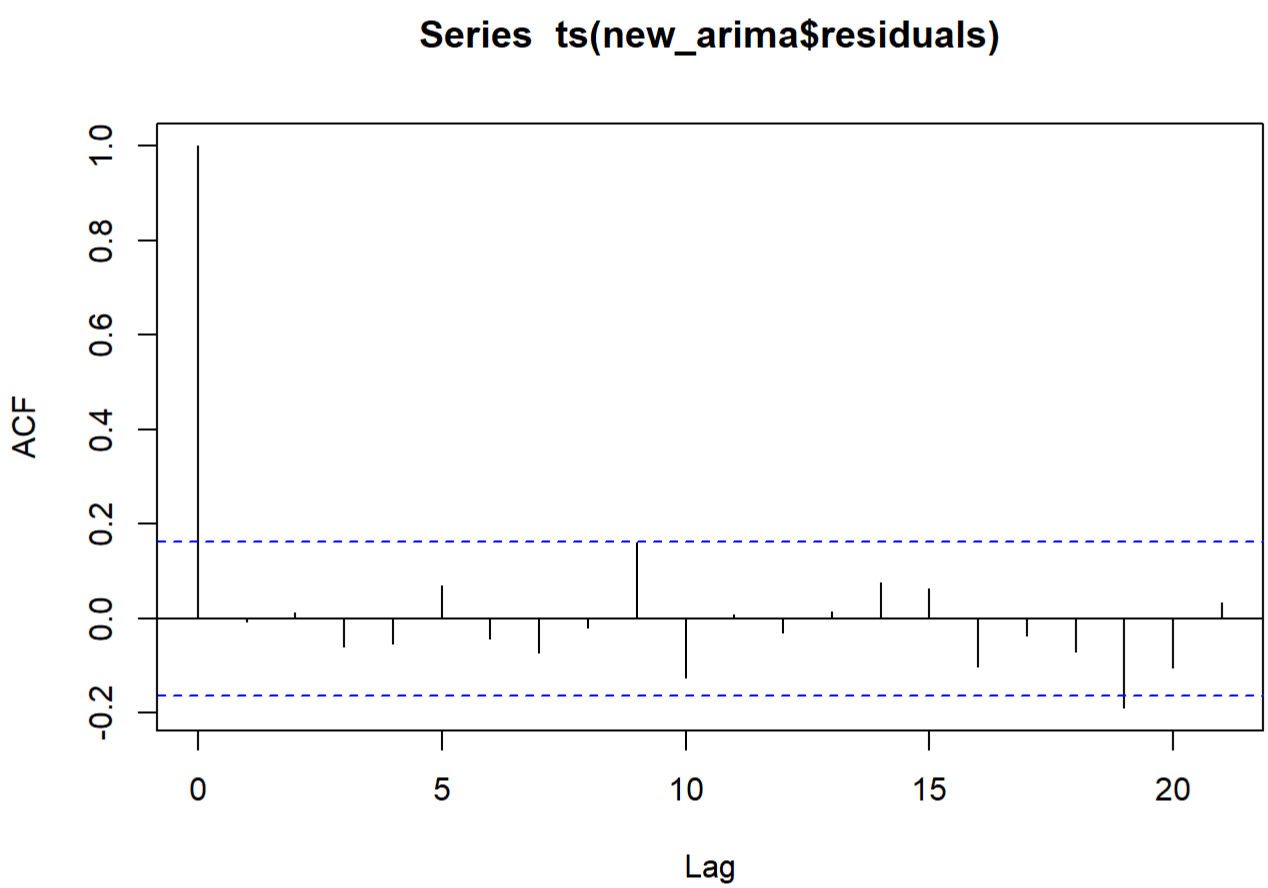
```
##
##  ARIMA(0,1,0)(0,1,0)[12]                    : 1133
##  ARIMA(0,1,0)(0,1,1)[12]                    : 1134
##  ARIMA(0,1,0)(0,1,2)[12]                    : 1135
##  ARIMA(0,1,0)(1,1,0)[12]                    : 1134
##  ARIMA(0,1,0)(1,1,1)[12]                    : 1135
##  ARIMA(0,1,0)(1,1,2)[12]                    : 1137
##  ARIMA(0,1,0)(2,1,0)[12]                    : 1134
##  ARIMA(0,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,0)(2,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(0,1,0)[12]                    : 1123
##  ARIMA(0,1,1)(0,1,1)[12]                    : 1125
##  ARIMA(0,1,1)(0,1,2)[12]                    : 1122
##  ARIMA(0,1,1)(1,1,0)[12]                    : 1124
##  ARIMA(0,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,1)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,1)(2,1,0)[12]                    : 1121
##  ARIMA(0,1,1)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,1)(2,1,2)[12]                    : Inf
##  ARIMA(0,1,2)(0,1,0)[12]                    : 1125
##  ARIMA(0,1,2)(0,1,1)[12]                    : 1127
##  ARIMA(0,1,2)(0,1,2)[12]                    : 1124
##  ARIMA(0,1,2)(1,1,0)[12]                    : 1127
##  ARIMA(0,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,2)(1,1,2)[12]                    : Inf
##  ARIMA(0,1,2)(2,1,0)[12]                    : 1123
##  ARIMA(0,1,2)(2,1,1)[12]                    : Inf
##  ARIMA(0,1,3)(0,1,0)[12]                    : 1124
##  ARIMA(0,1,3)(0,1,1)[12]                    : 1125
##  ARIMA(0,1,3)(0,1,2)[12]                    : 1124
##  ARIMA(0,1,3)(1,1,0)[12]                    : 1125
##  ARIMA(0,1,3)(1,1,1)[12]                    : Inf
##  ARIMA(0,1,3)(2,1,0)[12]                    : 1123
##  ARIMA(0,1,4)(0,1,0)[12]                    : 1125
##  ARIMA(0,1,4)(0,1,1)[12]                    : 1127
##  ARIMA(0,1,4)(1,1,0)[12]                    : 1126
##  ARIMA(0,1,5)(0,1,0)[12]                    : 1127
##  ARIMA(1,1,0)(0,1,0)[12]                    : 1123
##  ARIMA(1,1,0)(0,1,1)[12]                    : 1125
##  ARIMA(1,1,0)(0,1,2)[12]                    : 1122
##  ARIMA(1,1,0)(1,1,0)[12]                    : 1124
##  ARIMA(1,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(1,1,0)(2,1,0)[12]                    : 1121
##  ARIMA(1,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(1,1,0)(2,1,2)[12]                    : Inf
##  ARIMA(1,1,1)(0,1,0)[12]                    : 1125
##  ARIMA(1,1,1)(0,1,1)[12]                    : 1127
##  ARIMA(1,1,1)(0,1,2)[12]                    : 1124
##  ARIMA(1,1,1)(1,1,0)[12]                    : 1126
##  ARIMA(1,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,1)(1,1,2)[12]                    : Inf
```

```
##  ARIMA(1,1,1)(2,1,0)[12]                    : 1123
##  ARIMA(1,1,1)(2,1,1)[12]                    : Inf
##  ARIMA(1,1,2)(0,1,0)[12]                    : 1127
##  ARIMA(1,1,2)(0,1,1)[12]                    : 1129
##  ARIMA(1,1,2)(0,1,2)[12]                    : 1126
##  ARIMA(1,1,2)(1,1,0)[12]                    : Inf
##  ARIMA(1,1,2)(1,1,1)[12]                    : Inf
##  ARIMA(1,1,2)(2,1,0)[12]                    : Inf
##  ARIMA(1,1,3)(0,1,0)[12]                    : 1122
##  ARIMA(1,1,3)(0,1,1)[12]                    : 1123
##  ARIMA(1,1,3)(1,1,0)[12]                    : 1123
##  ARIMA(1,1,4)(0,1,0)[12]                    : Inf
##  ARIMA(2,1,0)(0,1,0)[12]                    : 1125
##  ARIMA(2,1,0)(0,1,1)[12]                    : 1127
##  ARIMA(2,1,0)(0,1,2)[12]                    : 1124
##  ARIMA(2,1,0)(1,1,0)[12]                    : 1127
##  ARIMA(2,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(2,1,0)(1,1,2)[12]                    : Inf
##  ARIMA(2,1,0)(2,1,0)[12]                    : 1124
##  ARIMA(2,1,0)(2,1,1)[12]                    : Inf
##  ARIMA(2,1,1)(0,1,0)[12]                    : 1120
##  ARIMA(2,1,1)(0,1,1)[12]                    : 1122
##  ARIMA(2,1,1)(0,1,2)[12]                    : 1120
##  ARIMA(2,1,1)(1,1,0)[12]                    : 1121
##  ARIMA(2,1,1)(1,1,1)[12]                    : Inf
##  ARIMA(2,1,1)(2,1,0)[12]                    : 1120
##  ARIMA(2,1,2)(0,1,0)[12]                    : 1122
##  ARIMA(2,1,2)(0,1,1)[12]                    : 1123
##  ARIMA(2,1,2)(1,1,0)[12]                    : 1123
##  ARIMA(2,1,3)(0,1,0)[12]                    : 1124
##  ARIMA(3,1,0)(0,1,0)[12]                    : 1127
##  ARIMA(3,1,0)(0,1,1)[12]                    : 1128
##  ARIMA(3,1,0)(0,1,2)[12]                    : 1126
##  ARIMA(3,1,0)(1,1,0)[12]                    : 1128
##  ARIMA(3,1,0)(1,1,1)[12]                    : Inf
##  ARIMA(3,1,0)(2,1,0)[12]                    : 1125
##  ARIMA(3,1,1)(0,1,0)[12]                    : 1122
##  ARIMA(3,1,1)(0,1,1)[12]                    : 1123
##  ARIMA(3,1,1)(1,1,0)[12]                    : 1123
##  ARIMA(3,1,2)(0,1,0)[12]                    : Inf
##  ARIMA(4,1,0)(0,1,0)[12]                    : 1127
##  ARIMA(4,1,0)(0,1,1)[12]                    : 1128
##  ARIMA(4,1,0)(1,1,0)[12]                    : 1128
##  ARIMA(4,1,1)(0,1,0)[12]                    : 1124
##  ARIMA(5,1,0)(0,1,0)[12]                    : 1129
##
##
##
##  Best model: ARIMA(2,1,1)(2,1,0)[12]
```

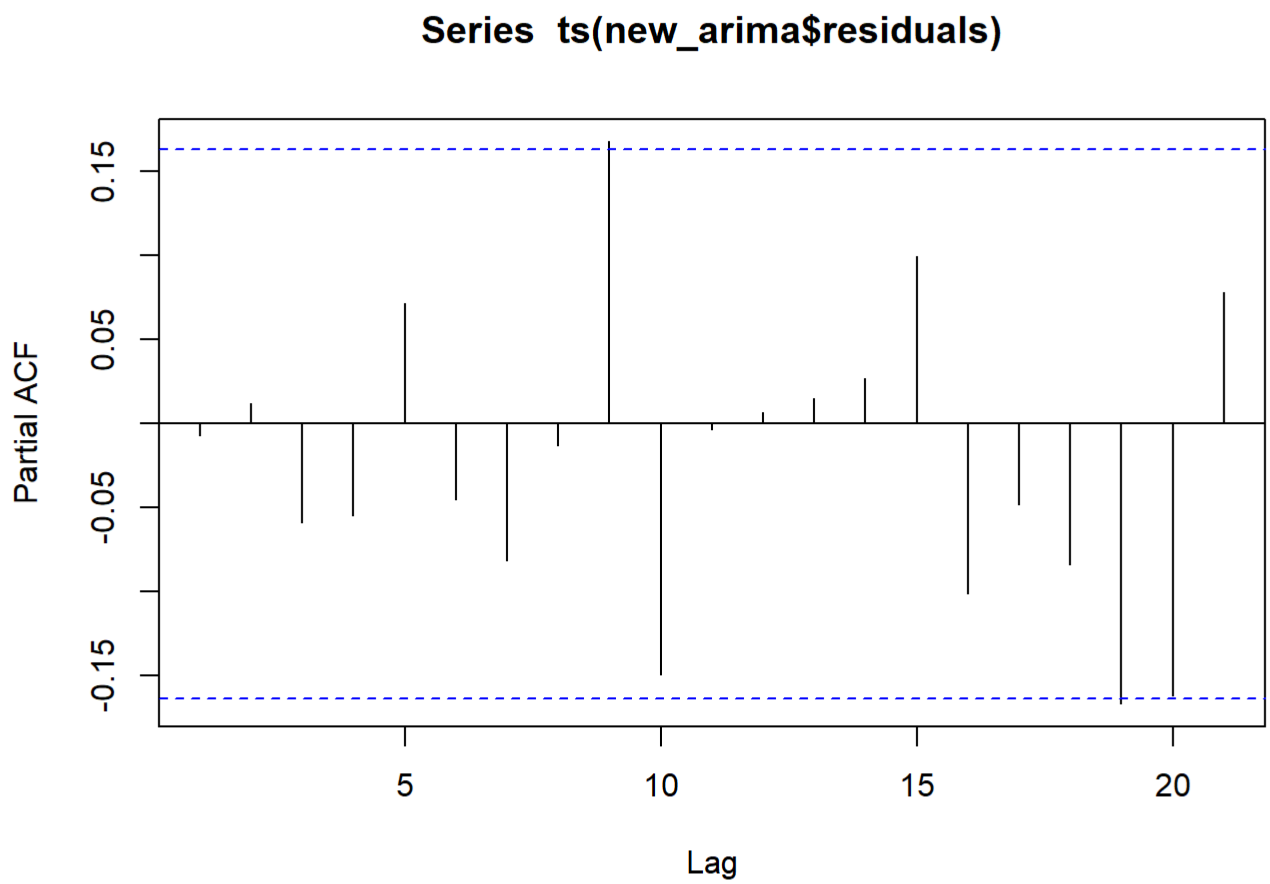
```
new_arima
```

```
## Series: data_ts
## ARIMA(2,1,1)(2,1,0)[12]
##
## Coefficients:
##          ar1    ar2     ma1     sar1    sar2
##        0.573  0.249  -0.985  -0.064  0.219
## s.e.  0.089  0.091   0.028   0.094  0.106
##
## sigma^2 = 280:  log likelihood = -553.4
## AIC=1119   AICc=1120   BIC=1136
```

```
# The best model has the lowest aic

# To check if the new model is stationary

acf(ts(new_arima$residuals))
```




```
pacf(ts(new_arima$residuals))
```




```
# also fine
```

Tractor Sales Forecasing

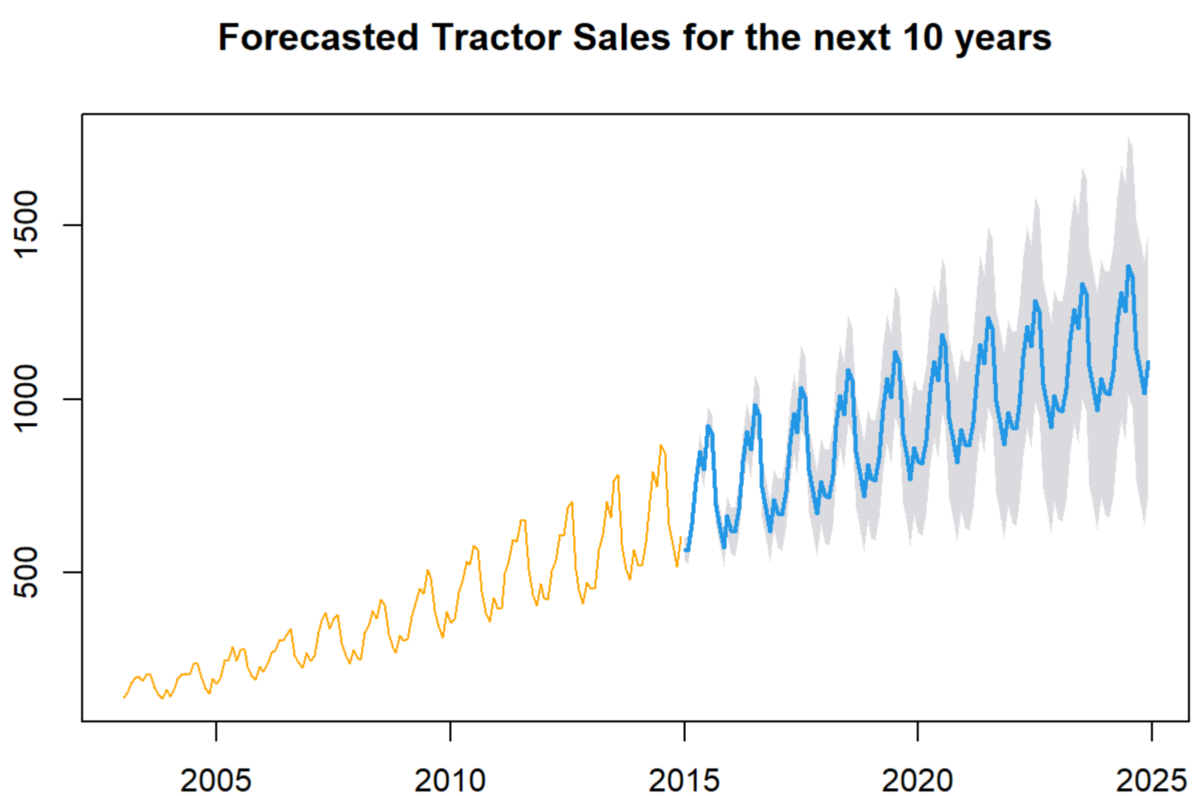
```
data_forecast <- forecast(new_arima, level = c(95), h=10*12)
data_forecast
```

```
##          Point Forecast  Lo 95  Hi 95
## Jan 2015          568.8  536.1  601.6
## Feb 2015          565.6  527.6  603.6
## Mar 2015          641.7  598.9  684.6
## Apr 2015          762.3  716.4  808.2
## May 2015          850.8  802.5  899.1
## Jun 2015          797.1  747.0  847.1
## Jul 2015          924.1  872.7  975.5
## Aug 2015          902.7  850.2  955.2
## Sep 2015          694.7  641.3  748.0
## Oct 2015          632.1  578.0  686.1
## Nov 2015          573.7  519.1  628.3
## Dec 2015          665.4  610.3  720.4
## Jan 2016          623.5  557.1  689.9
## Feb 2016          618.9  548.3  689.5
## Mar 2016          684.3  609.7  758.9
## Apr 2016          822.8  745.4  900.1
## May 2016          908.1  828.4  987.7
## Jun 2016          855.1  773.7  936.5
## Jul 2016          985.3  902.5 1068.2
## Aug 2016          955.2  871.2 1039.2
## Sep 2016          745.5  660.5  830.5
## Oct 2016          685.5  599.8  771.3
## Nov 2016          620.3  533.9  706.8
## Dec 2016          711.6  624.6  798.7
## Jan 2017          671.5  572.3  770.6
## Feb 2017          667.3  563.3  771.2
## Mar 2017          735.4  626.8  843.9
## Apr 2017          872.1  760.2  984.1
## May 2017          958.9  844.2 1073.6
## Jun 2017          903.7  786.8 1020.6
## Jul 2017         1034.8  916.1 1153.5
## Aug 2017         1005.6  885.4 1125.9
## Sep 2017          796.0  674.5  917.5
## Oct 2017          735.1  612.5  857.7
## Nov 2017          671.1  547.6  794.7
## Dec 2017          763.7  639.3  888.1
## Jan 2018          722.2  587.9  856.4
## Feb 2018          717.6  578.9  856.3
## Mar 2018          783.2  640.2  926.1
## Apr 2018          924.0  777.8 1070.2
## May 2018         1010.0  861.1 1158.9
## Jun 2018          955.1  804.0 1106.2
## Jul 2018         1086.9  933.9 1239.9
## Aug 2018         1055.7  901.1 1210.3
## Sep 2018          845.7  689.7 1001.7
## Oct 2018          785.4  628.2  942.6
## Nov 2018          719.9  561.6  878.2
## Dec 2018          812.3  653.0  971.5
## Jan 2019          771.2  602.5  939.9
## Feb 2019          766.8  593.7  939.9
## Mar 2019          833.1  655.7 1010.5
```

```
## Apr 2019         973.3  792.5 1154.0
## May 2019        1059.6  876.0 1243.2
## Jun 2019        1004.2  818.3 1190.2
## Jul 2019        1136.2  948.1 1324.2
## Aug 2019        1105.3  915.5 1295.1
## Sep 2019         895.3  704.0 1086.7
## Oct 2019         834.8  642.1 1027.5
## Nov 2019         769.6  575.7  963.6
## Dec 2019         862.3  667.2 1057.5
## Jan 2020         820.9  617.2 1024.7
## Feb 2020         816.4  608.3 1024.5
## Mar 2020         882.1  669.9 1094.4
## Apr 2020        1023.3  807.6 1238.9
## May 2020        1109.4  890.9 1327.9
## Jun 2020        1054.1  833.2 1275.1
## Jul 2020        1186.2  963.0 1409.3
## Aug 2020        1154.9  929.9 1379.9
## Sep 2020         944.8  718.1 1171.5
## Oct 2020         884.4  656.2 1112.7
## Nov 2020         818.9  589.3 1048.5
## Dec 2020         911.5  680.7 1142.4
## Jan 2021         870.2  631.1 1109.4
## Feb 2021         865.8  622.3 1109.2
## Mar 2021         931.7  684.0 1179.4
## Apr 2021        1072.6  821.5 1323.7
## May 2021        1158.8  904.8 1412.9
## Jun 2021        1103.5  846.8 1360.1
## Jul 2021        1235.5  976.6 1494.5
## Aug 2021        1204.3  943.4 1465.3
## Sep 2021         994.3  731.5 1257.1
## Oct 2021         933.9  669.4 1198.3
## Nov 2021         868.4  602.4 1134.5
## Dec 2021         961.1  693.7 1228.6
## Jan 2022         919.8  644.4 1195.1
## Feb 2022         915.3  635.6 1194.9
## Mar 2022         981.1  697.2 1264.9
## Apr 2022        1122.2  834.8 1409.5
## May 2022        1208.3  917.9 1498.8
## Jun 2022        1153.0  859.9 1446.1
## Jul 2022        1285.1  989.6 1580.6
## Aug 2022        1253.8  956.2 1551.5
## Sep 2022        1043.7  744.1 1343.4
## Oct 2022         983.3  681.9 1284.7
## Nov 2022         917.8  614.8 1220.9
## Dec 2022        1010.5  705.9 1315.2
## Jan 2023         969.2  656.8 1281.5
## Feb 2023         964.7  648.0 1281.4
## Mar 2023        1030.5  709.6 1351.5
## Apr 2023        1171.6  847.1 1496.1
## May 2023        1257.8  930.2 1585.4
## Jun 2023        1202.4  872.0 1532.8
## Jul 2023        1334.5 1001.6 1667.5
```

```
## Aug 2023         1303.3  968.0 1638.5
## Sep 2023         1093.2  755.9 1430.5
## Oct 2023         1032.8  693.5 1372.0
## Nov 2023          967.3  626.2 1308.3
## Dec 2023         1060.0  717.3 1402.7
## Jan 2024         1018.6  668.4 1368.9
## Feb 2024         1014.1  659.5 1368.8
## Mar 2024         1079.9  721.0 1438.9
## Apr 2024         1221.1  858.5 1583.6
## May 2024         1307.2  941.5 1673.0
## Jun 2024         1251.9  883.2 1620.6
## Jul 2024         1384.0 1012.7 1755.3
## Aug 2024         1352.7  979.0 1726.4
## Sep 2024         1142.6  766.7 1518.6
## Oct 2024         1082.2  704.2 1460.2
## Nov 2024         1016.7  636.8 1396.6
## Dec 2024         1109.4  727.7 1491.2
```

```
# the Lo 95 and high 95 is the confidence level, if it is low, it will be 536, if high, it  will be 601. It is safe to go with the minimum.


plot(data_forecast, main = "Forecasted Tractor Sales for the next 10 years", col="orange")
```



```
#  Interpretation: sales will keep growing (a trend) and also captures the seasonality and ARIMA model fits the best according to our end sample statistics and we use to form forecast.
```

Validation of the model

```
Box.test(data_forecast$residuals, lag =20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  data_forecast$residuals
## X-squared = 22, df = 20, p-value = 0.4
```

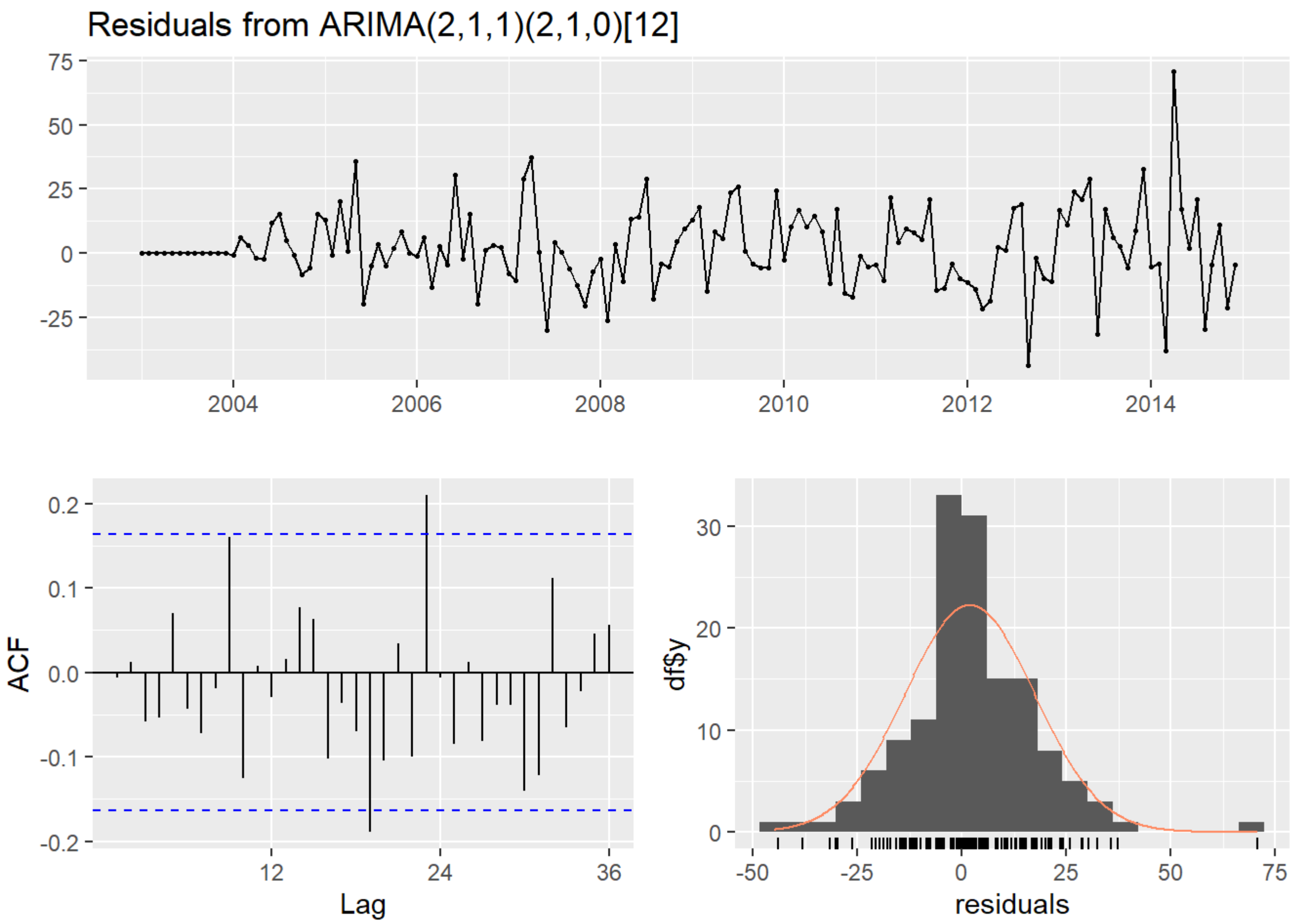
```
# p values less than 0.5, sqrt (sigma^2)

print(summary(data_forecast))
```

```
##
## Forecast method: ARIMA(2,1,1)(2,1,0)[12]
##
## Model Information:
## Series: data_ts
## ARIMA(2,1,1)(2,1,0)[12]
##
## Coefficients:
##         ar1    ar2     ma1    sar1   sar2
##       0.573  0.249  -0.985  -0.064  0.219
## s.e.  0.089  0.091   0.028   0.094  0.106
##
## sigma^2 = 280:  log likelihood = -553.4
## AIC=1119   AICc=1120   BIC=1136
##
## Error measures:
##                ME  RMSE   MAE   MPE  MAPE   MASE      ACF1
## Training set 1.89 15.64 11.23 0.366 2.862 0.2519 -0.006727
##
## Forecasts:
##          Point Forecast  Lo 95  Hi 95
## Jan 2015          568.8  536.1  601.6
## Feb 2015          565.6  527.6  603.6
## Mar 2015          641.7  598.9  684.6
## Apr 2015          762.3  716.4  808.2
## May 2015          850.8  802.5  899.1
## Jun 2015          797.1  747.0  847.1
## Jul 2015          924.1  872.7  975.5
## Aug 2015          902.7  850.2  955.2
## Sep 2015          694.7  641.3  748.0
## Oct 2015          632.1  578.0  686.1
## Nov 2015          573.7  519.1  628.3
## Dec 2015          665.4  610.3  720.4
## Jan 2016          623.5  557.1  689.9
## Feb 2016          618.9  548.3  689.5
## Mar 2016          684.3  609.7  758.9
## Apr 2016          822.8  745.4  900.1
## May 2016          908.1  828.4  987.7
## Jun 2016          855.1  773.7  936.5
## Jul 2016          985.3  902.5 1068.2
## Aug 2016          955.2  871.2 1039.2
## Sep 2016          745.5  660.5  830.5
## Oct 2016          685.5  599.8  771.3
## Nov 2016          620.3  533.9  706.8
## Dec 2016          711.6  624.6  798.7
## Jan 2017          671.5  572.3  770.6
## Feb 2017          667.3  563.3  771.2
## Mar 2017          735.4  626.8  843.9
## Apr 2017          872.1  760.2  984.1
## May 2017          958.9  844.2 1073.6
## Jun 2017          903.7  786.8 1020.6
## Jul 2017         1034.8  916.1 1153.5
```

```
## Aug 2017         1005.6  885.4 1125.9
## Sep 2017          796.0  674.5  917.5
## Oct 2017          735.1  612.5  857.7
## Nov 2017          671.1  547.6  794.7
## Dec 2017          763.7  639.3  888.1
## Jan 2018          722.2  587.9  856.4
## Feb 2018          717.6  578.9  856.3
## Mar 2018          783.2  640.2  926.1
## Apr 2018          924.0  777.8 1070.2
## May 2018         1010.0  861.1 1158.9
## Jun 2018          955.1  804.0 1106.2
## Jul 2018         1086.9  933.9 1239.9
## Aug 2018         1055.7  901.1 1210.3
## Sep 2018          845.7  689.7 1001.7
## Oct 2018          785.4  628.2  942.6
## Nov 2018          719.9  561.6  878.2
## Dec 2018          812.3  653.0  971.5
## Jan 2019          771.2  602.5  939.9
## Feb 2019          766.8  593.7  939.9
## Mar 2019          833.1  655.7 1010.5
## Apr 2019          973.3  792.5 1154.0
## May 2019         1059.6  876.0 1243.2
## Jun 2019         1004.2  818.3 1190.2
## Jul 2019         1136.2  948.1 1324.2
## Aug 2019         1105.3  915.5 1295.1
## Sep 2019          895.3  704.0 1086.7
## Oct 2019          834.8  642.1 1027.5
## Nov 2019          769.6  575.7  963.6
## Dec 2019          862.3  667.2 1057.5
## Jan 2020          820.9  617.2 1024.7
## Feb 2020          816.4  608.3 1024.5
## Mar 2020          882.1  669.9 1094.4
## Apr 2020         1023.3  807.6 1238.9
## May 2020         1109.4  890.9 1327.9
## Jun 2020         1054.1  833.2 1275.1
## Jul 2020         1186.2  963.0 1409.3
## Aug 2020         1154.9  929.9 1379.9
## Sep 2020          944.8  718.1 1171.5
## Oct 2020          884.4  656.2 1112.7
## Nov 2020          818.9  589.3 1048.5
## Dec 2020          911.5  680.7 1142.4
## Jan 2021          870.2  631.1 1109.4
## Feb 2021          865.8  622.3 1109.2
## Mar 2021          931.7  684.0 1179.4
## Apr 2021         1072.6  821.5 1323.7
## May 2021         1158.8  904.8 1412.9
## Jun 2021         1103.5  846.8 1360.1
## Jul 2021         1235.5  976.6 1494.5
## Aug 2021         1204.3  943.4 1465.3
## Sep 2021          994.3  731.5 1257.1
## Oct 2021          933.9  669.4 1198.3
## Nov 2021          868.4  602.4 1134.5
```

```
## Dec 2021          961.1  693.7 1228.6
## Jan 2022          919.8  644.4 1195.1
## Feb 2022          915.3  635.6 1194.9
## Mar 2022          981.1  697.2 1264.9
## Apr 2022         1122.2  834.8 1409.5
## May 2022         1208.3  917.9 1498.8
## Jun 2022         1153.0  859.9 1446.1
## Jul 2022         1285.1  989.6 1580.6
## Aug 2022         1253.8  956.2 1551.5
## Sep 2022         1043.7  744.1 1343.4
## Oct 2022          983.3  681.9 1284.7
## Nov 2022          917.8  614.8 1220.9
## Dec 2022         1010.5  705.9 1315.2
## Jan 2023          969.2  656.8 1281.5
## Feb 2023          964.7  648.0 1281.4
## Mar 2023         1030.5  709.6 1351.5
## Apr 2023         1171.6  847.1 1496.1
## May 2023         1257.8  930.2 1585.4
## Jun 2023         1202.4  872.0 1532.8
## Jul 2023         1334.5 1001.6 1667.5
## Aug 2023         1303.3  968.0 1638.5
## Sep 2023         1093.2  755.9 1430.5
## Oct 2023         1032.8  693.5 1372.0
## Nov 2023          967.3  626.2 1308.3
## Dec 2023         1060.0  717.3 1402.7
## Jan 2024         1018.6  668.4 1368.9
## Feb 2024         1014.1  659.5 1368.8
## Mar 2024         1079.9  721.0 1438.9
## Apr 2024         1221.1  858.5 1583.6
## May 2024         1307.2  941.5 1673.0
## Jun 2024         1251.9  883.2 1620.6
## Jul 2024         1384.0 1012.7 1755.3
## Aug 2024         1352.7  979.0 1726.4
## Sep 2024         1142.6  766.7 1518.6
## Oct 2024         1082.2  704.2 1460.2
## Nov 2024         1016.7  636.8 1396.6
## Dec 2024         1109.4  727.7 1491.2
```

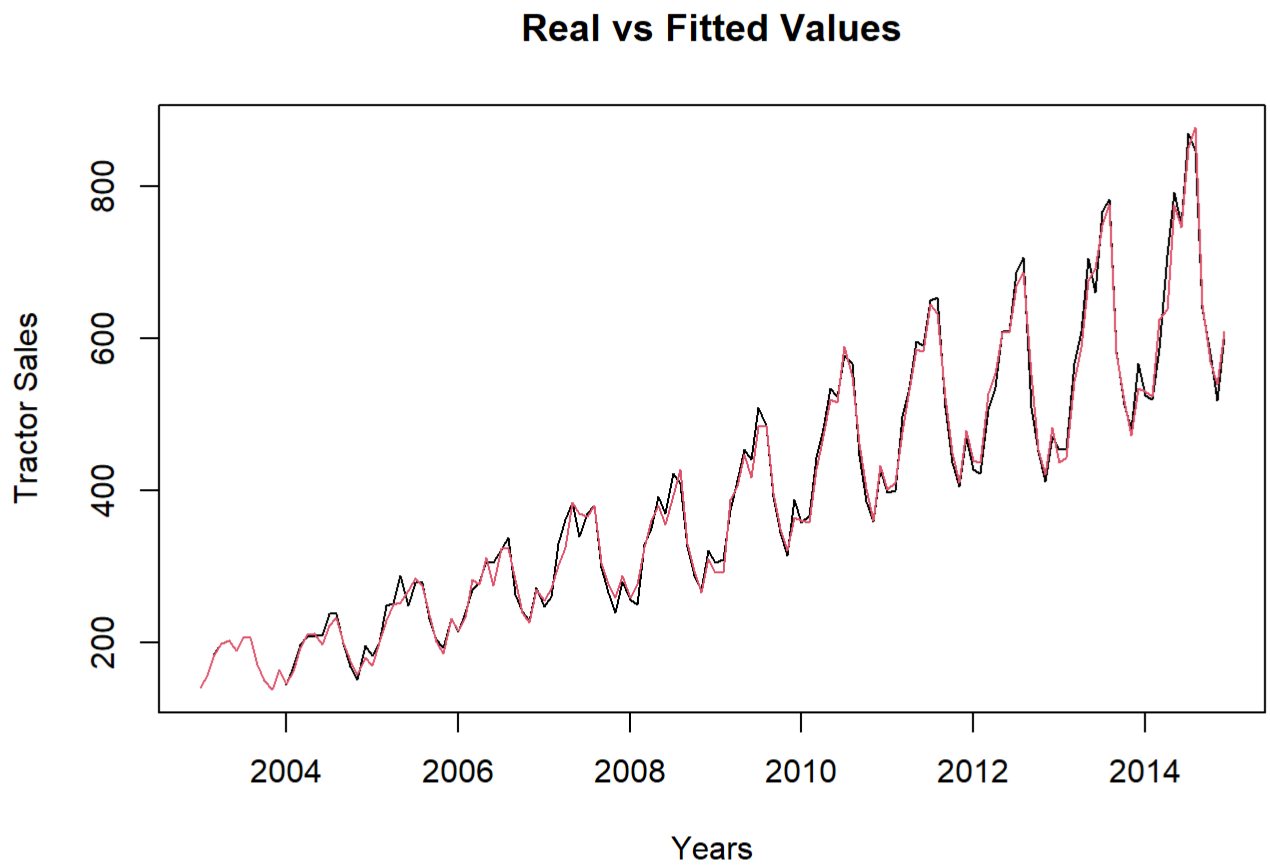
```
checkresiduals(data_forecast)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)(2,1,0)[12]
## Q* = 31, df = 19, p-value = 0.04
##
## Model df: 5.   Total lags used: 24
```

Plotting real vs Fitted Values

```
ts.plot(new_arima$x, new_arima$fitted, col=1:2, gpars = list(xlab = "Years", ylab="Tractor Sale
s", main= "Real vs Fitted Values"))
```



Data Forcast using Seasonal Naive Method

```
data_naive <- snaive(data_ts, level = c(95), h = 10*12)
data_naive
```

```
##          Point Forecast Lo 95  Hi 95
## Jan 2015            525 425.1  624.9
## Feb 2015            520 420.1  619.9
## Mar 2015            587 487.1  686.9
## Apr 2015            710 610.1  809.9
## May 2015            793 693.1  892.9
## Jun 2015            749 649.1  848.9
## Jul 2015            871 771.1  970.9
## Aug 2015            848 748.1  947.9
## Sep 2015            640 540.1  739.9
## Oct 2015            581 481.1  680.9
## Nov 2015            519 419.1  618.9
## Dec 2015            605 505.1  704.9
## Jan 2016            525 383.8  666.2
## Feb 2016            520 378.8  661.2
## Mar 2016            587 445.8  728.2
## Apr 2016            710 568.8  851.2
## May 2016            793 651.8  934.2
## Jun 2016            749 607.8  890.2
## Jul 2016            871 729.8 1012.2
## Aug 2016            848 706.8  989.2
## Sep 2016            640 498.8  781.2
## Oct 2016            581 439.8  722.2
## Nov 2016            519 377.8  660.2
## Dec 2016            605 463.8  746.2
## Jan 2017            525 352.1  697.9
## Feb 2017            520 347.1  692.9
## Mar 2017            587 414.1  759.9
## Apr 2017            710 537.1  882.9
## May 2017            793 620.1  965.9
## Jun 2017            749 576.1  921.9
## Jul 2017            871 698.1 1043.9
## Aug 2017            848 675.1 1020.9
## Sep 2017            640 467.1  812.9
## Oct 2017            581 408.1  753.9
## Nov 2017            519 346.1  691.9
## Dec 2017            605 432.1  777.9
## Jan 2018            525 325.3  724.7
## Feb 2018            520 320.3  719.7
## Mar 2018            587 387.3  786.7
## Apr 2018            710 510.3  909.7
## May 2018            793 593.3  992.7
## Jun 2018            749 549.3  948.7
## Jul 2018            871 671.3 1070.7
## Aug 2018            848 648.3 1047.7
## Sep 2018            640 440.3  839.7
## Oct 2018            581 381.3  780.7
## Nov 2018            519 319.3  718.7
## Dec 2018            605 405.3  804.7
## Jan 2019            525 301.7  748.3
## Feb 2019            520 296.7  743.3
## Mar 2019            587 363.7  810.3
```

```
## Apr 2019            710 486.7  933.3
## May 2019            793 569.7 1016.3
## Jun 2019            749 525.7  972.3
## Jul 2019            871 647.7 1094.3
## Aug 2019            848 624.7 1071.3
## Sep 2019            640 416.7  863.3
## Oct 2019            581 357.7  804.3
## Nov 2019            519 295.7  742.3
## Dec 2019            605 381.7  828.3
## Jan 2020            525 280.4  769.6
## Feb 2020            520 275.4  764.6
## Mar 2020            587 342.4  831.6
## Apr 2020            710 465.4  954.6
## May 2020            793 548.4 1037.6
## Jun 2020            749 504.4  993.6
## Jul 2020            871 626.4 1115.6
## Aug 2020            848 603.4 1092.6
## Sep 2020            640 395.4  884.6
## Oct 2020            581 336.4  825.6
## Nov 2020            519 274.4  763.6
## Dec 2020            605 360.4  849.6
## Jan 2021            525 260.8  789.2
## Feb 2021            520 255.8  784.2
## Mar 2021            587 322.8  851.2
## Apr 2021            710 445.8  974.2
## May 2021            793 528.8 1057.2
## Jun 2021            749 484.8 1013.2
## Jul 2021            871 606.8 1135.2
## Aug 2021            848 583.8 1112.2
## Sep 2021            640 375.8  904.2
## Oct 2021            581 316.8  845.2
## Nov 2021            519 254.8  783.2
## Dec 2021            605 340.8  869.2
## Jan 2022            525 242.6  807.4
## Feb 2022            520 237.6  802.4
## Mar 2022            587 304.6  869.4
## Apr 2022            710 427.6  992.4
## May 2022            793 510.6 1075.4
## Jun 2022            749 466.6 1031.4
## Jul 2022            871 588.6 1153.4
## Aug 2022            848 565.6 1130.4
## Sep 2022            640 357.6  922.4
## Oct 2022            581 298.6  863.4
## Nov 2022            519 236.6  801.4
## Dec 2022            605 322.6  887.4
## Jan 2023            525 225.4  824.6
## Feb 2023            520 220.4  819.6
## Mar 2023            587 287.4  886.6
## Apr 2023            710 410.4 1009.6
## May 2023            793 493.4 1092.6
## Jun 2023            749 449.4 1048.6
## Jul 2023            871 571.4 1170.6
```

```
## Aug 2023           848 548.4 1147.6
## Sep 2023           640 340.4  939.6
## Oct 2023           581 281.4  880.6
## Nov 2023           519 219.4  818.6
## Dec 2023           605 305.4  904.6
## Jan 2024           525 209.2  840.8
## Feb 2024           520 204.2  835.8
## Mar 2024           587 271.2  902.8
## Apr 2024           710 394.2 1025.8
## May 2024           793 477.2 1108.8
## Jun 2024           749 433.2 1064.8
## Jul 2024           871 555.2 1186.8
## Aug 2024           848 532.2 1163.8
## Sep 2024           640 324.2  955.8
## Oct 2024           581 265.2  896.8
## Nov 2024           519 203.2  834.8
## Dec 2024           605 289.2  920.8
```

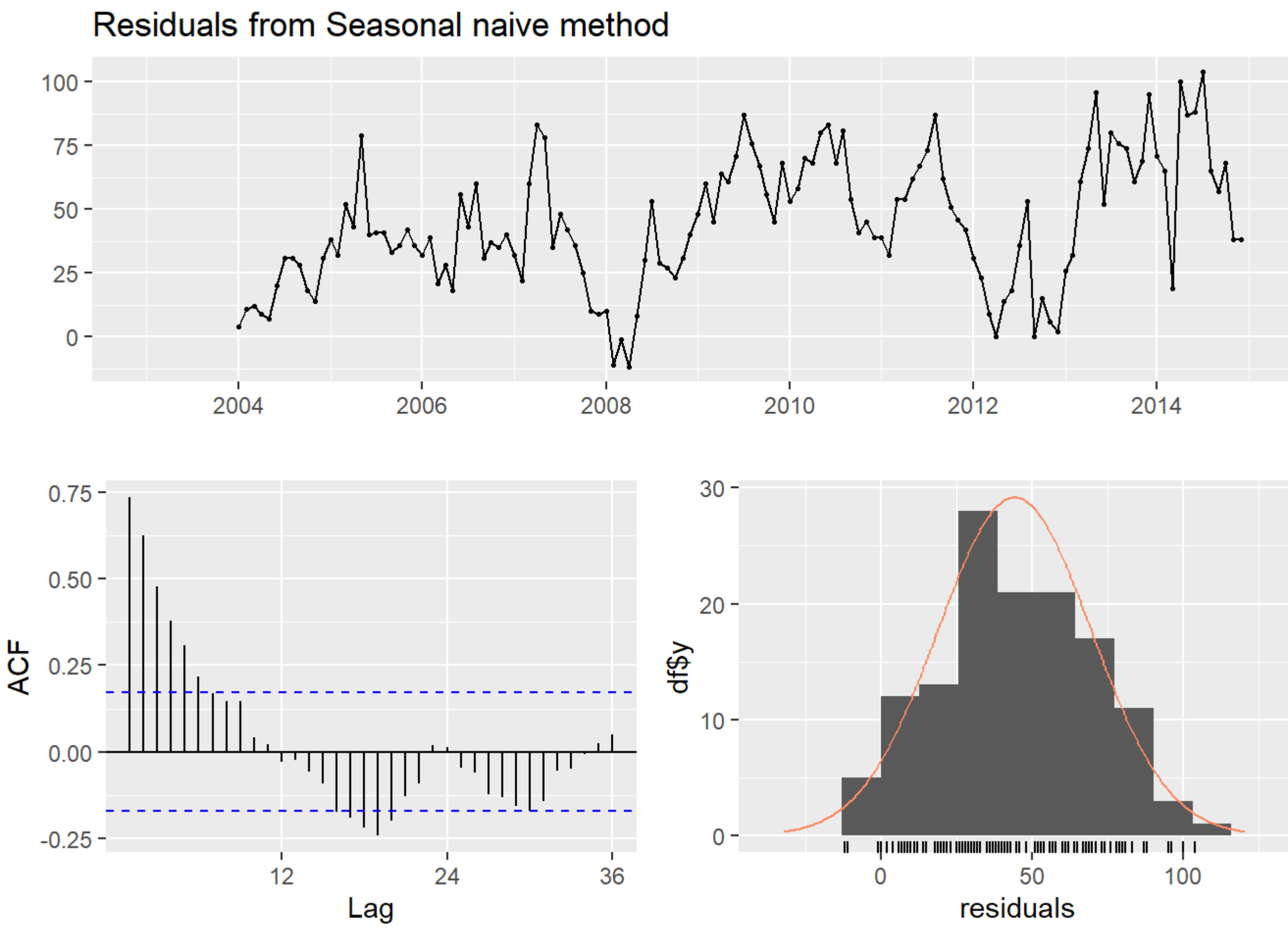
```
print(summary(data_naive))  # resdiual sd : 50.9462
```

```
##
## Forecast method: Seasonal naive method
##
## Model Information:
## Call: snaive(y = data_ts, h = 10 * 12, level = c(95))
##
## Residual sd: 50.9462
##
## Error measures:
##                 ME  RMSE   MAE   MPE  MAPE MASE   ACF1
## Training set 44.21 50.95 44.58 11.13 11.25    1 0.7347
##
## Forecasts:
##          Point Forecast Lo 95  Hi 95
## Jan 2015            525 425.1  624.9
## Feb 2015            520 420.1  619.9
## Mar 2015            587 487.1  686.9
## Apr 2015            710 610.1  809.9
## May 2015            793 693.1  892.9
## Jun 2015            749 649.1  848.9
## Jul 2015            871 771.1  970.9
## Aug 2015            848 748.1  947.9
## Sep 2015            640 540.1  739.9
## Oct 2015            581 481.1  680.9
## Nov 2015            519 419.1  618.9
## Dec 2015            605 505.1  704.9
## Jan 2016            525 383.8  666.2
## Feb 2016            520 378.8  661.2
## Mar 2016            587 445.8  728.2
## Apr 2016            710 568.8  851.2
## May 2016            793 651.8  934.2
## Jun 2016            749 607.8  890.2
## Jul 2016            871 729.8 1012.2
## Aug 2016            848 706.8  989.2
## Sep 2016            640 498.8  781.2
## Oct 2016            581 439.8  722.2
## Nov 2016            519 377.8  660.2
## Dec 2016            605 463.8  746.2
## Jan 2017            525 352.1  697.9
## Feb 2017            520 347.1  692.9
## Mar 2017            587 414.1  759.9
## Apr 2017            710 537.1  882.9
## May 2017            793 620.1  965.9
## Jun 2017            749 576.1  921.9
## Jul 2017            871 698.1 1043.9
## Aug 2017            848 675.1 1020.9
## Sep 2017            640 467.1  812.9
## Oct 2017            581 408.1  753.9
## Nov 2017            519 346.1  691.9
## Dec 2017            605 432.1  777.9
## Jan 2018            525 325.3  724.7
## Feb 2018            520 320.3  719.7
```

```
## Mar 2018            587 387.3  786.7
## Apr 2018            710 510.3  909.7
## May 2018            793 593.3  992.7
## Jun 2018            749 549.3  948.7
## Jul 2018            871 671.3 1070.7
## Aug 2018            848 648.3 1047.7
## Sep 2018            640 440.3  839.7
## Oct 2018            581 381.3  780.7
## Nov 2018            519 319.3  718.7
## Dec 2018            605 405.3  804.7
## Jan 2019            525 301.7  748.3
## Feb 2019            520 296.7  743.3
## Mar 2019            587 363.7  810.3
## Apr 2019            710 486.7  933.3
## May 2019            793 569.7 1016.3
## Jun 2019            749 525.7  972.3
## Jul 2019            871 647.7 1094.3
## Aug 2019            848 624.7 1071.3
## Sep 2019            640 416.7  863.3
## Oct 2019            581 357.7  804.3
## Nov 2019            519 295.7  742.3
## Dec 2019            605 381.7  828.3
## Jan 2020            525 280.4  769.6
## Feb 2020            520 275.4  764.6
## Mar 2020            587 342.4  831.6
## Apr 2020            710 465.4  954.6
## May 2020            793 548.4 1037.6
## Jun 2020            749 504.4  993.6
## Jul 2020            871 626.4 1115.6
## Aug 2020            848 603.4 1092.6
## Sep 2020            640 395.4  884.6
## Oct 2020            581 336.4  825.6
## Nov 2020            519 274.4  763.6
## Dec 2020            605 360.4  849.6
## Jan 2021            525 260.8  789.2
## Feb 2021            520 255.8  784.2
## Mar 2021            587 322.8  851.2
## Apr 2021            710 445.8  974.2
## May 2021            793 528.8 1057.2
## Jun 2021            749 484.8 1013.2
## Jul 2021            871 606.8 1135.2
## Aug 2021            848 583.8 1112.2
## Sep 2021            640 375.8  904.2
## Oct 2021            581 316.8  845.2
## Nov 2021            519 254.8  783.2
## Dec 2021            605 340.8  869.2
## Jan 2022            525 242.6  807.4
## Feb 2022            520 237.6  802.4
## Mar 2022            587 304.6  869.4
## Apr 2022            710 427.6  992.4
## May 2022            793 510.6 1075.4
## Jun 2022            749 466.6 1031.4
```

```
## Jul 2022            871 588.6 1153.4
## Aug 2022            848 565.6 1130.4
## Sep 2022            640 357.6  922.4
## Oct 2022            581 298.6  863.4
## Nov 2022            519 236.6  801.4
## Dec 2022            605 322.6  887.4
## Jan 2023            525 225.4  824.6
## Feb 2023            520 220.4  819.6
## Mar 2023            587 287.4  886.6
## Apr 2023            710 410.4 1009.6
## May 2023            793 493.4 1092.6
## Jun 2023            749 449.4 1048.6
## Jul 2023            871 571.4 1170.6
## Aug 2023            848 548.4 1147.6
## Sep 2023            640 340.4  939.6
## Oct 2023            581 281.4  880.6
## Nov 2023            519 219.4  818.6
## Dec 2023            605 305.4  904.6
## Jan 2024            525 209.2  840.8
## Feb 2024            520 204.2  835.8
## Mar 2024            587 271.2  902.8
## Apr 2024            710 394.2 1025.8
## May 2024            793 477.2 1108.8
## Jun 2024            749 433.2 1064.8
## Jul 2024            871 555.2 1186.8
## Aug 2024            848 532.2 1163.8
## Sep 2024            640 324.2  955.8
## Oct 2024            581 265.2  896.8
## Nov 2024            519 203.2  834.8
## Dec 2024            605 289.2  920.8
```

```
checkresiduals(data_naive)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 247, df = 24, p-value <0.0000000000000002
##
## Model df: 0.   Total lags used: 24
```

Data Forcast using Holt’s winter (Exponential smoothing) Method

```
data_ets <- hw(data_ts, level = c(95),  h=24) #seasonal =c("multiplicative)
data_ets
```

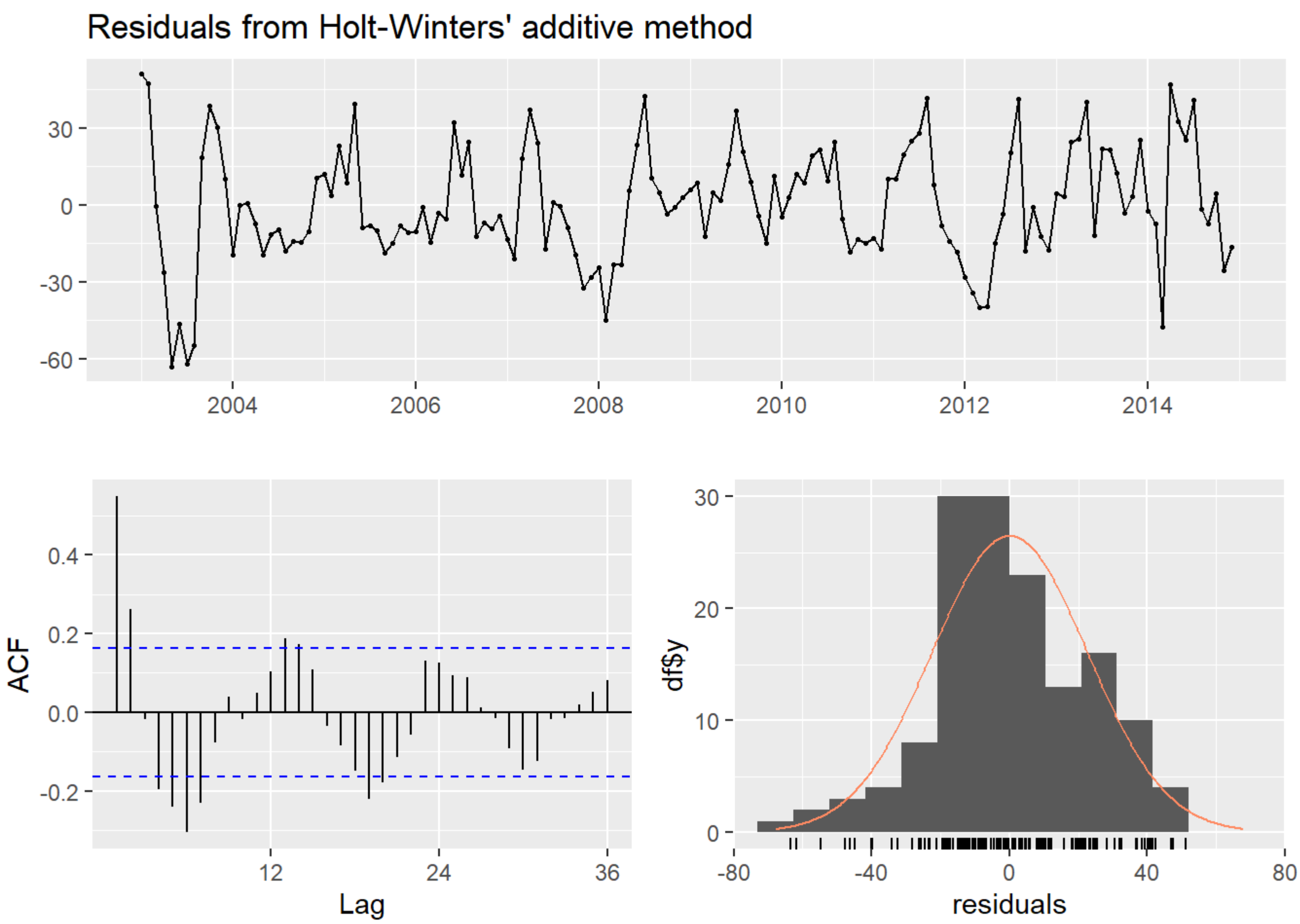
```
##          Point Forecast Lo 95  Hi 95
## Jan 2015          576.7 529.9  623.6
## Feb 2015          573.1 525.5  620.7
## Mar 2015          648.9 600.5  697.3
## Apr 2015          763.2 714.0  812.4
## May 2015          840.2 790.2  890.1
## Jun 2015          791.5 740.8  842.2
## Jul 2015          906.0 854.5  957.4
## Aug 2015          883.3 831.1  935.4
## Sep 2015          676.6 623.7  729.5
## Oct 2015          616.7 563.2  670.3
## Nov 2015          559.5 505.2  613.8
## Dec 2015          648.5 593.5  703.5
## Jan 2016          620.2 547.9  692.5
## Feb 2016          616.6 543.8  689.4
## Mar 2016          692.4 619.1  765.7
## Apr 2016          806.7 732.9  880.5
## May 2016          883.7 809.3  958.0
## Jun 2016          835.0 760.1  909.8
## Jul 2016          949.4 874.1 1024.8
## Aug 2016          926.8 850.9 1002.6
## Sep 2016          720.1 643.7  796.4
## Oct 2016          660.2 583.4  737.1
## Nov 2016          603.0 525.6  680.3
## Dec 2016          692.0 614.1  769.8
```

```
print(summary(data_ets))  # resdiual sd : 23.9
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = data_ts, h = 24, level = c(95))
##
##   Smoothing parameters:
##     alpha = 0.1847
##     beta  = 0.0001
##     gamma = 0.8153
##
##   Initial states:
##     l = 161.2076
##     b = 3.6227
##     s = -37.35 -75.54 -60.95 -12.71 89.86 89.27
##            50.75 73.68 31.35 -4.768 -68.44 -75.16
##
##   sigma:  23.9
##
##  AIC AICc  BIC
## 1647 1652 1697
##
## Error measures:
##                   ME  RMSE   MAE     MPE MAPE  MASE   ACF1
## Training set 0.06784 22.53 17.79 -0.6082 5.68 0.399 0.5484
##
## Forecasts:
##          Point Forecast Lo 95  Hi 95
## Jan 2015          576.7 529.9  623.6
## Feb 2015          573.1 525.5  620.7
## Mar 2015          648.9 600.5  697.3
## Apr 2015          763.2 714.0  812.4
## May 2015          840.2 790.2  890.1
## Jun 2015          791.5 740.8  842.2
## Jul 2015          906.0 854.5  957.4
## Aug 2015          883.3 831.1  935.4
## Sep 2015          676.6 623.7  729.5
## Oct 2015          616.7 563.2  670.3
## Nov 2015          559.5 505.2  613.8
## Dec 2015          648.5 593.5  703.5
## Jan 2016          620.2 547.9  692.5
## Feb 2016          616.6 543.8  689.4
## Mar 2016          692.4 619.1  765.7
## Apr 2016          806.7 732.9  880.5
## May 2016          883.7 809.3  958.0
## Jun 2016          835.0 760.1  909.8
## Jul 2016          949.4 874.1 1024.8
## Aug 2016          926.8 850.9 1002.6
## Sep 2016          720.1 643.7  796.4
```

```
## Oct 2016          660.2 583.4  737.1
## Nov 2016          603.0 525.6  680.3
## Dec 2016          692.0 614.1  769.8
```

```
checkresiduals(data_ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' additive method
## Q* = 134, df = 8, p-value <0.0000000000000002
##
## Model df: 16.   Total lags used: 24
```

Recommendation and Conclusion:

Tractor sales will keep growing upward (a trend) and it also captures the seasonality and ARIMA model fits the best according to our end sample statistics and we use to form forecast.