Dietmar Aumann, Final Project, E185 Big Data Analytics

Big Data Predictions of Stock Prices in R

0. What we want to do

This project aims to introduce a variety of methods for predicting stock prices.
The main tool used is R with packages *forecast* and *fpp,* downloadable on http://cran.r-project.org/web/packages/.

Predicting techniques used and briefly explained are:

- Holt Winters Filtering
- ARIMA model
- Neural Networks
- Pynomial & Linear generalization trends

All of the above are used in a practical demo and finally evaluated on 30 stocks using the **Mean Absolute Error**.

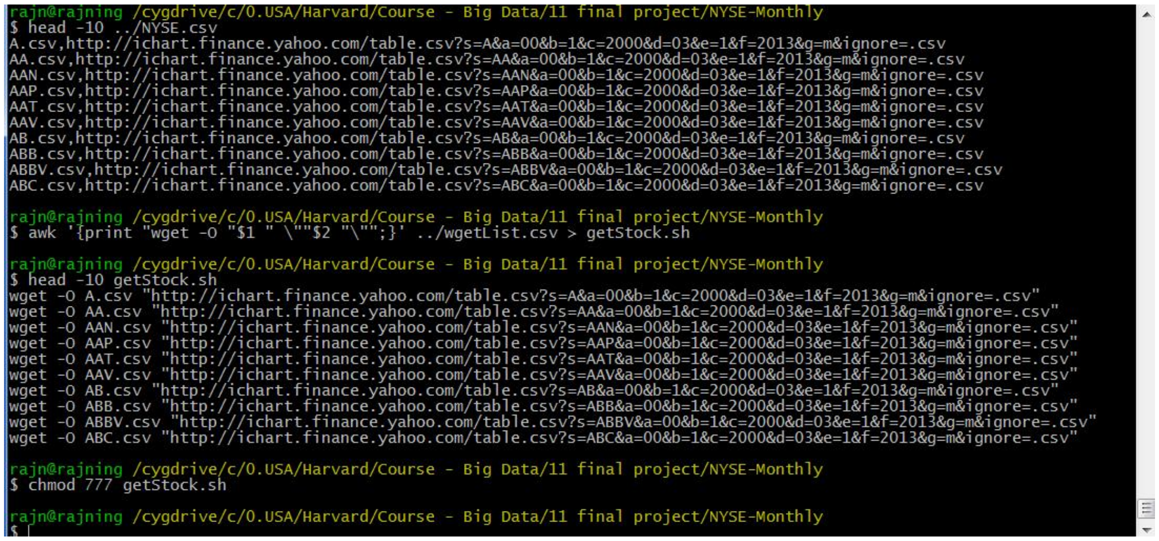
As a prerequisite, an understanding of basic Machine Learning techniques and methods is recommended – if this is not the case, I attached sources for more research if you are interested.

The workflow of this project is the following:

- Acquiring all or some stock prices from http://finance.yahoo.com using a *wget* script
- Creating three functions of the stock price, on which later predictions are based: those are the actual stock graph, as well as two generalized (trend) graphs of it
- Getting familiar with the prediction methods while plotting from attached *SCRIPT.R*
- Computing in total 24 different predictions of the stock prices with attached *findBestPrediction.R*
- Conducting a test evaluation by using 30 stocks and attached *runTestEvaluation.R*
- Looking at the result and setting a future goal (which you can try for yourself if you are interested)

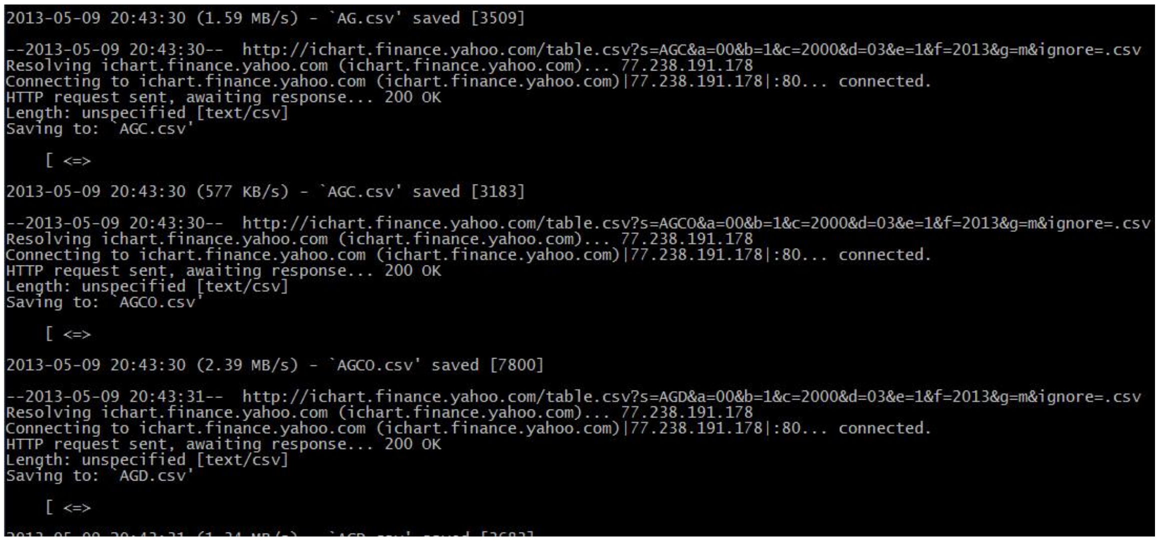
1. Getting the data

Before we can start, we need to get the stock prices in CSV file.
I attached a script (getStock.sh) which will download (wget) all stocks (>3000) on NYSE, monthly prices starting from January 2000.



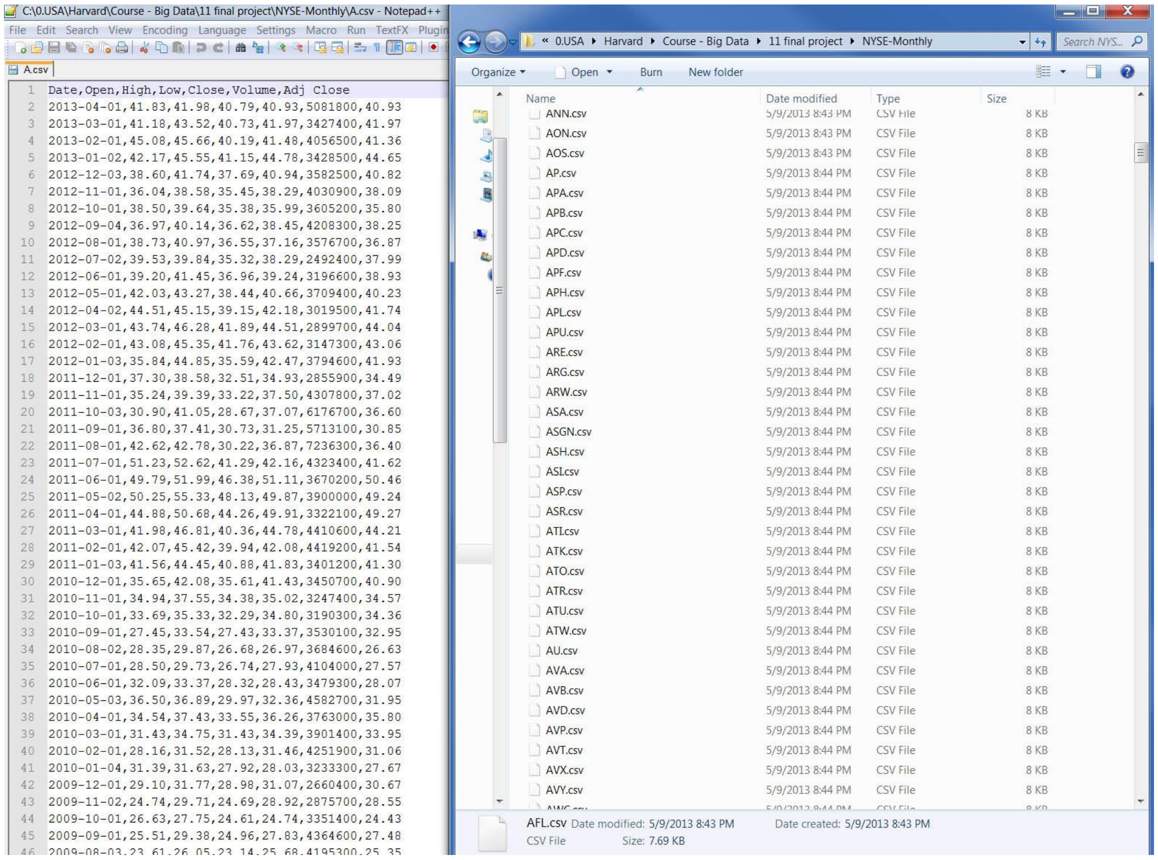
```
rajn@rajning /cygdrive/c/0.USA/Harvard/Course - Big Data/11 final project/NYSE-Monthly
$ head -10 ../NYSE.csv
A.csv,http://ichart.finance.yahoo.com/table.csv?s=A&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
AA.csv,http://ichart.finance.yahoo.com/table.csv?s=AA&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
AAN.csv,http://ichart.finance.yahoo.com/table.csv?s=AAN&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
AAP.csv,http://ichart.finance.yahoo.com/table.csv?s=AAP&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
AAT.csv,http://ichart.finance.yahoo.com/table.csv?s=AAT&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
AAV.csv,http://ichart.finance.yahoo.com/table.csv?s=AAV&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
AB.csv,http://ichart.finance.yahoo.com/table.csv?s=AB&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
ABB.csv,http://ichart.finance.yahoo.com/table.csv?s=ABB&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
ABBV.csv,http://ichart.finance.yahoo.com/table.csv?s=ABBV&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
ABC.csv,http://ichart.finance.yahoo.com/table.csv?s=ABC&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv

rajn@rajning /cygdrive/c/0.USA/Harvard/Course - Big Data/11 final project/NYSE-Monthly
$ awk '{print "wget -O "$1 " \""$2 "\"";}' ../wgetList.csv > getStock.sh

rajn@rajning /cygdrive/c/0.USA/Harvard/Course - Big Data/11 final project/NYSE-Monthly
$ head -10 getStock.sh
wget -O A.csv "http://ichart.finance.yahoo.com/table.csv?s=A&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O AA.csv "http://ichart.finance.yahoo.com/table.csv?s=AA&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O AAN.csv "http://ichart.finance.yahoo.com/table.csv?s=AAN&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O AAP.csv "http://ichart.finance.yahoo.com/table.csv?s=AAP&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O AAT.csv "http://ichart.finance.yahoo.com/table.csv?s=AAT&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O AAV.csv "http://ichart.finance.yahoo.com/table.csv?s=AAV&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O AB.csv "http://ichart.finance.yahoo.com/table.csv?s=AB&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O ABB.csv "http://ichart.finance.yahoo.com/table.csv?s=ABB&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O ABBV.csv "http://ichart.finance.yahoo.com/table.csv?s=ABBV&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"
wget -O ABC.csv "http://ichart.finance.yahoo.com/table.csv?s=ABC&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv"

rajn@rajning /cygdrive/c/0.USA/Harvard/Course - Big Data/11 final project/NYSE-Monthly
$ chmod 777 getStock.sh

rajn@rajning /cygdrive/c/0.USA/Harvard/Course - Big Data/11 final project/NYSE-Monthly
$ 
```

Here is a snippet of how the download process should look like:



```
2013-05-09 20:43:30 (1.59 MB/s) - `AG.csv' saved [3509]

--2013-05-09 20:43:30--  http://ichart.finance.yahoo.com/table.csv?s=AGC&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
Resolving ichart.finance.yahoo.com (ichart.finance.yahoo.com)... 77.238.191.178
Connecting to ichart.finance.yahoo.com (ichart.finance.yahoo.com)|77.238.191.178|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: `AGC.csv'

    [ <=>

2013-05-09 20:43:30 (577 KB/s) - `AGC.csv' saved [3183]

--2013-05-09 20:43:30--  http://ichart.finance.yahoo.com/table.csv?s=AGCO&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
Resolving ichart.finance.yahoo.com (ichart.finance.yahoo.com)... 77.238.191.178
Connecting to ichart.finance.yahoo.com (ichart.finance.yahoo.com)|77.238.191.178|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: `AGCO.csv'

    [ <=>

2013-05-09 20:43:30 (2.39 MB/s) - `AGCO.csv' saved [7800]

--2013-05-09 20:43:31--  http://ichart.finance.yahoo.com/table.csv?s=AGD&a=00&b=1&c=2000&d=03&e=1&f=2013&g=m&ignore=.csv
Resolving ichart.finance.yahoo.com (ichart.finance.yahoo.com)... 77.238.191.178
Connecting to ichart.finance.yahoo.com (ichart.finance.yahoo.com)|77.238.191.178|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: `AGD.csv'

    [ <=>

2013-05-09 20:43:31 (1.24 MB/s) - `AGD.csv' saved [3683]
```

And the end result should look similar to that:



C:\0.USA\Harvard\Course - Big Data\11 final project\NYSE-Monthly\A.csv - Notepad++

```
Date,Open,High,Low,Close,Volume,Adj Close
2013-04-01,41.83,41.98,40.79,40.93,5081800,40.93
2013-03-01,41.18,43.52,40.73,41.97,3427400,41.97
2013-02-01,45.08,45.66,40.19,41.48,4056500,41.36
2013-01-02,42.17,45.55,41.15,44.78,3428500,44.65
2012-12-03,38.60,41.74,37.69,40.94,3582500,40.82
2012-11-01,36.04,38.58,35.45,38.29,4030900,38.09
2012-10-01,38.50,39.64,35.38,35.99,3605200,35.80
2012-09-04,36.97,40.14,36.62,38.45,4208300,38.25
2012-08-01,38.73,40.97,36.55,37.16,3576700,36.87
2012-07-02,39.53,39.84,35.32,38.29,2492400,37.99
2012-06-01,39.20,41.45,36.96,39.24,3196600,38.93
2012-05-01,42.03,43.27,38.44,40.66,3709400,40.23
2012-04-02,44.51,45.15,39.15,42.18,3019500,41.74
2012-03-01,43.74,46.28,41.89,44.51,2899700,44.04
2012-02-01,43.08,45.35,41.76,43.62,3147300,43.06
2012-01-03,35.84,44.85,35.59,42.47,3794600,41.93
2011-12-01,37.30,38.58,32.51,34.93,2855900,34.49
2011-11-01,35.24,39.39,33.22,37.50,4307800,37.02
2011-10-03,30.90,41.05,28.67,37.07,6176700,36.60
2011-09-01,36.80,37.41,30.73,31.25,5713100,30.85
2011-08-01,42.62,42.78,30.22,36.87,7236300,36.40
2011-07-01,51.23,52.62,41.29,42.16,4323400,41.62
2011-06-01,49.79,51.99,46.38,51.11,3670200,50.46
2011-05-02,50.25,55.33,48.13,49.87,3900000,49.24
2011-04-01,44.88,50.68,44.26,49.91,3322100,49.27
2011-03-01,41.98,46.81,40.36,44.78,4410600,44.21
2011-02-01,42.07,45.42,39.94,42.08,4419200,41.54
2011-01-03,41.56,44.45,40.88,41.83,3401200,41.30
2010-12-01,35.65,42.08,35.61,41.43,3450700,40.90
2010-11-01,34.94,37.55,34.38,35.02,3247400,34.57
2010-10-01,33.69,35.33,32.29,34.80,3190300,34.36
2010-09-01,27.45,33.54,27.43,33.37,3530100,32.95
2010-08-02,28.35,29.87,26.68,26.97,3684600,26.63
2010-07-01,28.50,29.73,26.74,27.93,4104000,27.57
2010-06-01,32.09,33.37,28.32,28.43,3479300,28.07
2010-05-03,36.50,36.89,29.97,32.36,4582700,31.95
2010-04-01,34.54,37.43,33.55,36.26,3763000,35.80
2010-03-01,31.43,34.75,31.43,34.39,3901400,33.95
2010-02-01,28.16,31.52,28.13,31.46,4251900,31.06
2010-01-04,31.39,31.63,27.92,28.03,3233300,27.67
2009-12-01,29.10,31.77,28.98,31.07,2660400,30.67
2009-11-02,24.74,29.71,24.69,28.92,2875700,28.55
2009-10-01,26.63,27.75,24.61,24.74,3351400,24.43
2009-09-01,25.51,29.38,24.96,27.83,4364600,27.48
2009-08-03,23.61,26.05,23.14,25.68,4195300,25.35
```

« 0.USA ▸ Harvard ▸ Course - Big Data ▸ 11 final project ▸ NYSE-Monthly

Name	Date modified	Type	Size
ANN.csv	5/9/2013 8:43 PM	CSV File	8 KB
AON.csv	5/9/2013 8:43 PM	CSV File	8 KB
AOS.csv	5/9/2013 8:43 PM	CSV File	8 KB
AP.csv	5/9/2013 8:44 PM	CSV File	8 KB
APA.csv	5/9/2013 8:44 PM	CSV File	8 KB
APB.csv	5/9/2013 8:44 PM	CSV File	8 KB
APC.csv	5/9/2013 8:44 PM	CSV File	8 KB
APD.csv	5/9/2013 8:44 PM	CSV File	8 KB
APF.csv	5/9/2013 8:44 PM	CSV File	8 KB
APH.csv	5/9/2013 8:44 PM	CSV File	8 KB
APL.csv	5/9/2013 8:44 PM	CSV File	8 KB
APU.csv	5/9/2013 8:44 PM	CSV File	8 KB
ARE.csv	5/9/2013 8:44 PM	CSV File	8 KB
ARG.csv	5/9/2013 8:44 PM	CSV File	8 KB
ARW.csv	5/9/2013 8:44 PM	CSV File	8 KB
ASA.csv	5/9/2013 8:44 PM	CSV File	8 KB
ASGN.csv	5/9/2013 8:44 PM	CSV File	8 KB
ASH.csv	5/9/2013 8:44 PM	CSV File	8 KB
ASI.csv	5/9/2013 8:44 PM	CSV File	8 KB
ASP.csv	5/9/2013 8:44 PM	CSV File	8 KB
ASR.csv	5/9/2013 8:44 PM	CSV File	8 KB
ATI.csv	5/9/2013 8:44 PM	CSV File	8 KB
ATK.csv	5/9/2013 8:44 PM	CSV File	8 KB
ATO.csv	5/9/2013 8:44 PM	CSV File	8 KB
ATR.csv	5/9/2013 8:44 PM	CSV File	8 KB
ATU.csv	5/9/2013 8:44 PM	CSV File	8 KB
ATW.csv	5/9/2013 8:44 PM	CSV File	8 KB
AU.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVA.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVB.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVD.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVP.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVT.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVX.csv	5/9/2013 8:44 PM	CSV File	8 KB
AVY.csv	5/9/2013 8:44 PM	CSV File	8 KB

AFL.csv Date modified: 5/9/2013 8:43 PM Date created: 5/9/2013 8:43 PM
CSV File Size: 7.69 KB

File Edit Search View Encoding Language Settings Macro Run TextFX Plugin
A.csv
Organize Open Burn New folder
Search NYS...

2. Trying out different approaches

In order to find a suitable technique for predicting the stock prices accurately, I comprised a pool of 25 prediction techniques which will later compete with each other.

Five main techniques (and variants of those) are each used on the **actual train function** itself and also on two different generalized functions (**trends**).

In the following I will use the attached script SCRIPT.r with the stock price ABC (AmerisourceBergen Corporation) for demonstration

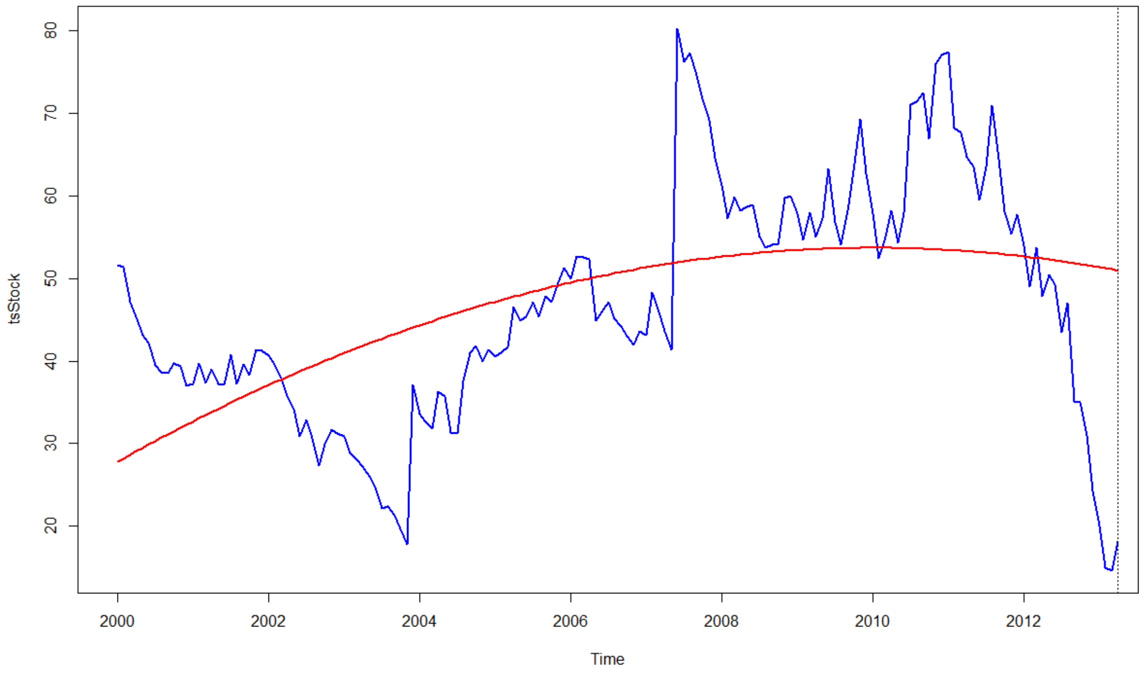
Because the actual functions of stock prices can sometimes be chaotic and misleading for the predictions, I wanted to create two generalizations (trends) of the actual function and also run the prediction techniques on those.

I began by creating a polynomial model of the function by calling:

```
tl = seq(2000,2013,length=length(tsStock))
tl2 = tl^3
polyStock = lm(tsStock ~ tl + tl2)
tsStocktrend1=ts(polyStock$fit,start=c(2000, 1),frequency=12)
```

tsStock is here the time series object of the actual graph.
The function *lm* creates the polynomial model (line 3),
which result is then again converted into a time series object (line4)

The resulting generalized functions can be seen here (red):

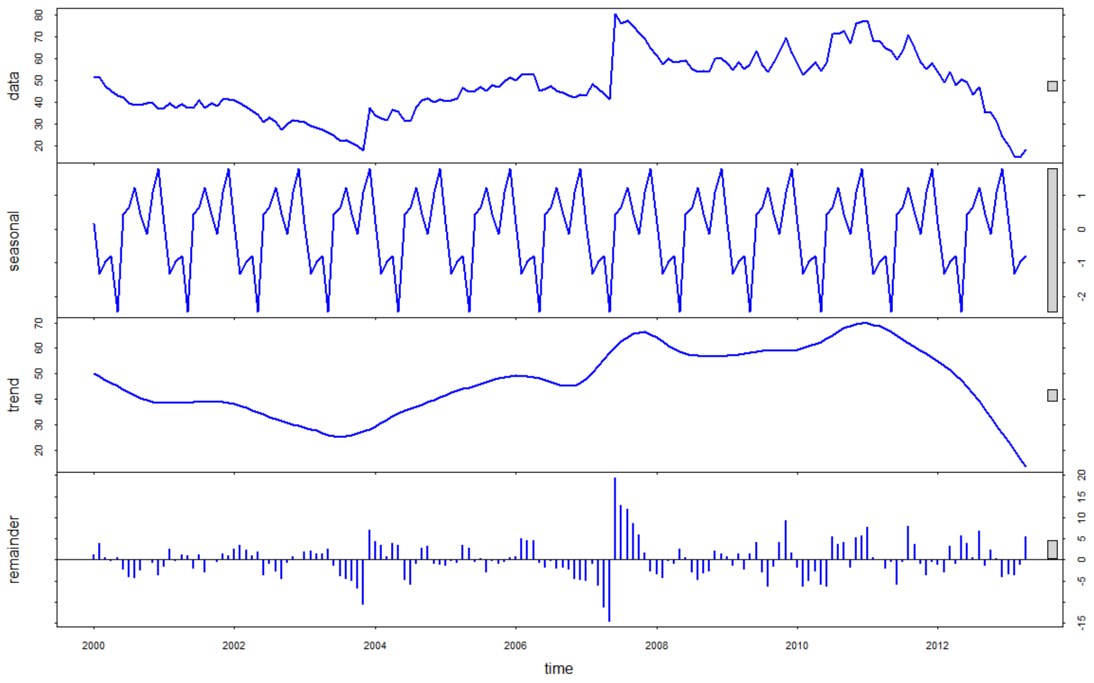


Now that we have one generalized graph, let's create another one based on the *loess window for seasonal extraction* called *STL (**S**easonal,**T**rend, **L**oess)*:

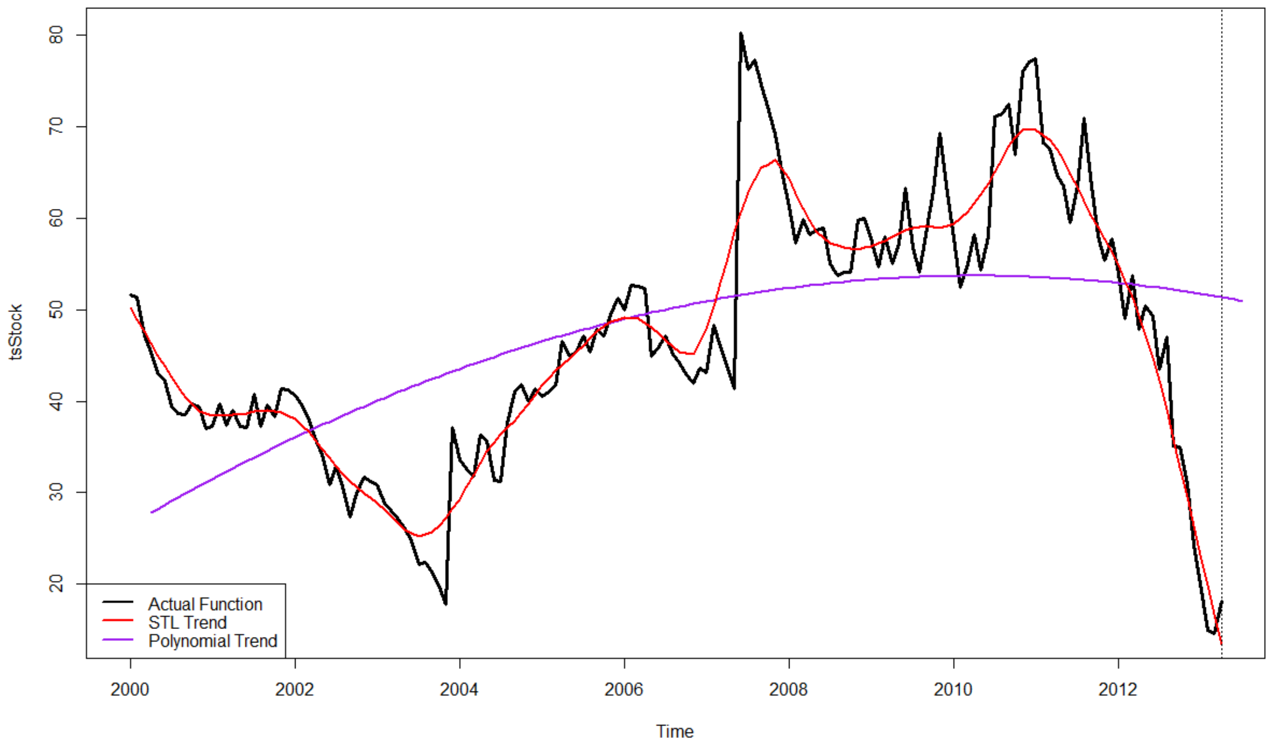
```
stlStock = stl(train,s.window="periodic")
tsStocktrend2 = stlStock$time.series[,2]
```

For more information for the stl function, visit:
http://stat.ethz.ch/R-manual/R-devel/library/stats/html/stl.html

Here you can see the analysis of our stock price graph – we use the **trend component** as our second trend base function (tsStocktrend2).



Let’s have an overview of our base functions, before we begin applying the prediction methods on them:



First we begin by using the most general graph (Polynomial), which is marked purple above.

We compute the predictions using several approaches.

- **Holt Winter Filtering**
  (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/HoltWinters.html)
  In this method the predictions with the minimized squared error are selected
  Here two variants are used, the gamma parameter = false means exponential smoothing is used.

```
HWStock1_ng = HoltWinters(tsStocktrend1,gamma=FALSE)
HWStock1 = HoltWinters(tsStocktrend1)
```

- **Neural Networks**
  (http://www.inside-r.org/packages/cran/forecast/docs/nnetar)
  Here a single hidden layer neural network with type feed-forward is used to predict the future prices

```
NETfit1 <- nnetar(tsStocktrend1)
```

The neural network used in this demonstration averaged 20 networks, each of which consisting of a 2-2-1 network with 9 weights - sigma^2 was estimated as 0.0007721

- **ARIMA model**
  (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/arima.html)
  This approach uses an **A**uto**r**egressive **I**ntegrated **M**oving **A**verage model as a generalization of the ARMA model which are used mainly for non-stationarity function predictions.
  Two variants are used, one determining the parameter automatically and the other with a fixed parameter approach.

```
autofit1 = auto.arima(tsStocktrend1)
fit12 <- arima(tsStocktrend1, order=c(1,0,0), list(order=c(2,1,0), period=12))
```

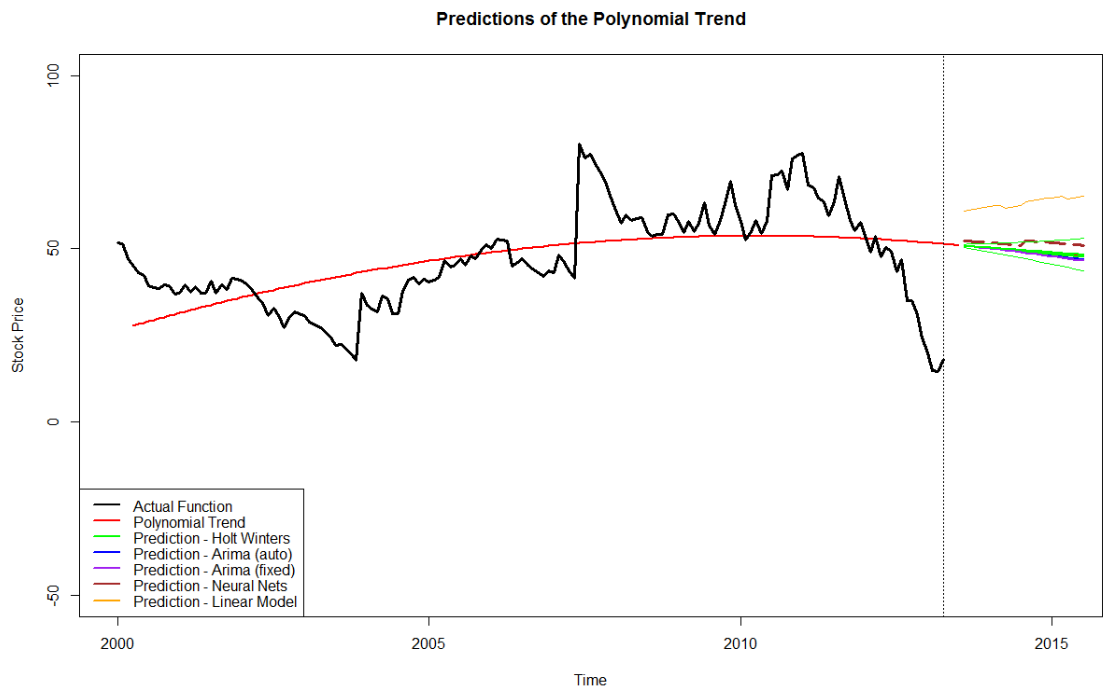
- **Linear Model**
  (http://www.inside-r.org/packages/cran/forecast/docs/tslm)
  This TSLM uses a **l**inear **m**odel as a base and adds **t**rend and **s**eason components to the computation

```
fitl1 <- tslm(tsStocktrend1 ~ trend + season, lambda=0)
```

- **STL Model**
  (http://stat.ethz.ch/R-manual/R-devel/library/stats/html/stl.html)
  The Seasonal Trend Loess model, which was introduced above.

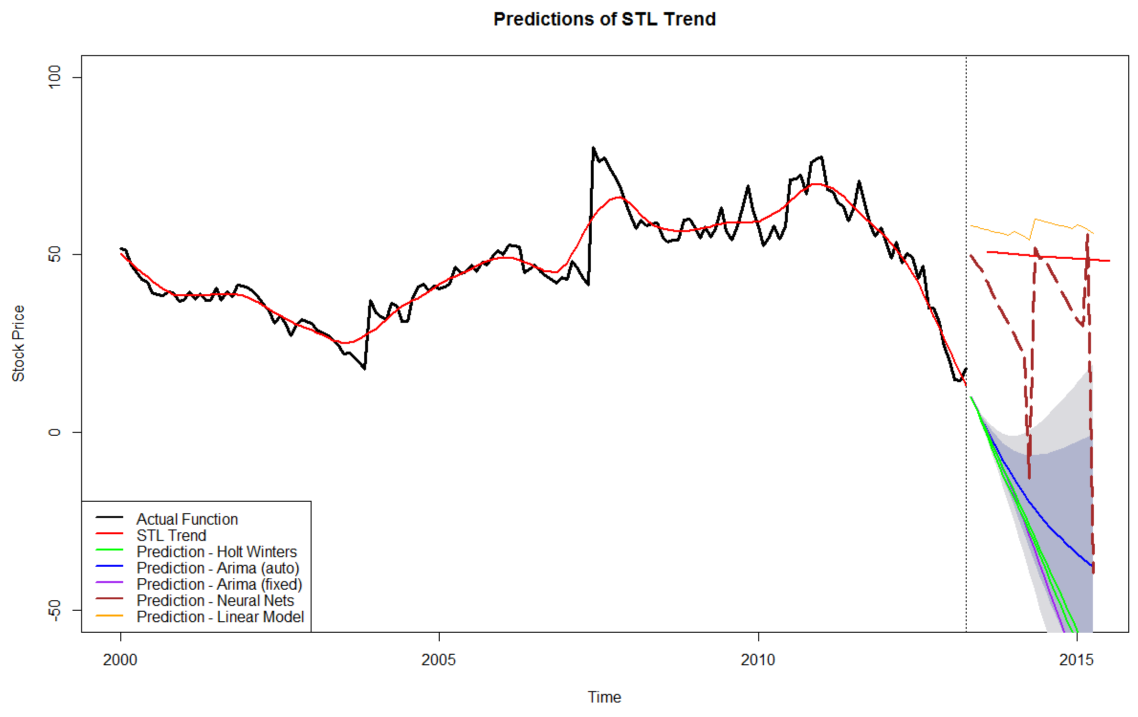
```
stlStock1 = stl(tsStocktrend1,s.window="periodic")
```

Now let’s see the predictions:

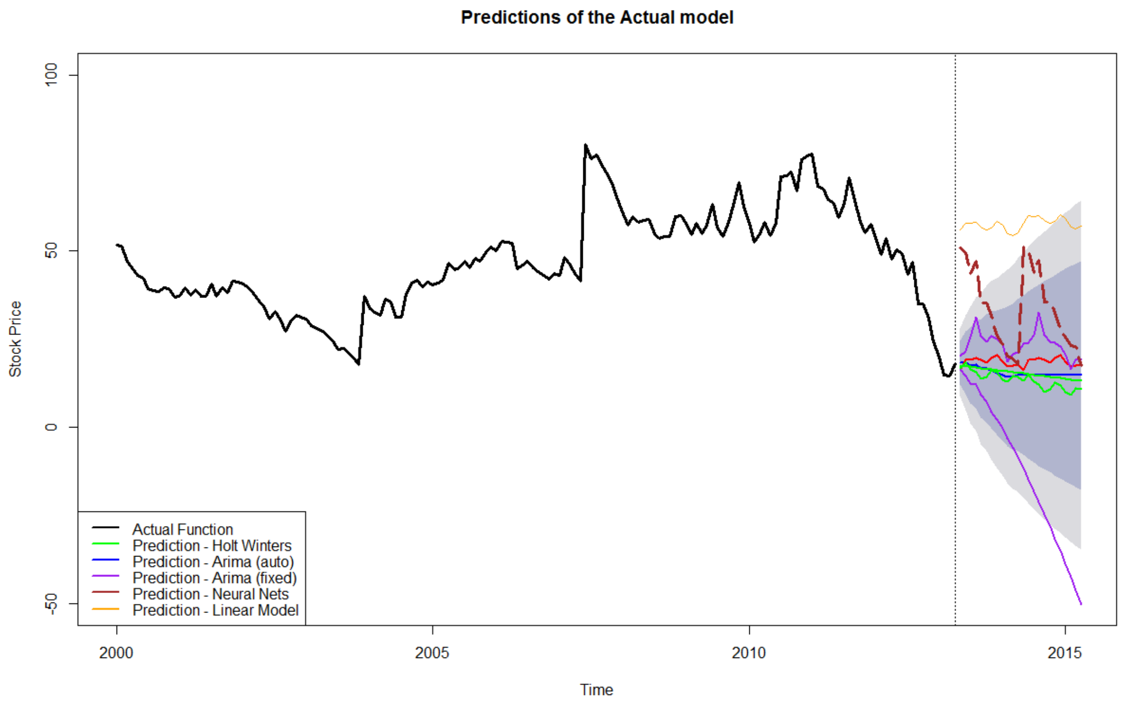


As we can see, the predictions do not vary a lot, because the Polynomial Trend model has not a lot of variance in the Y Axes.

This is different, when we look at the **STL Trend predictions**:

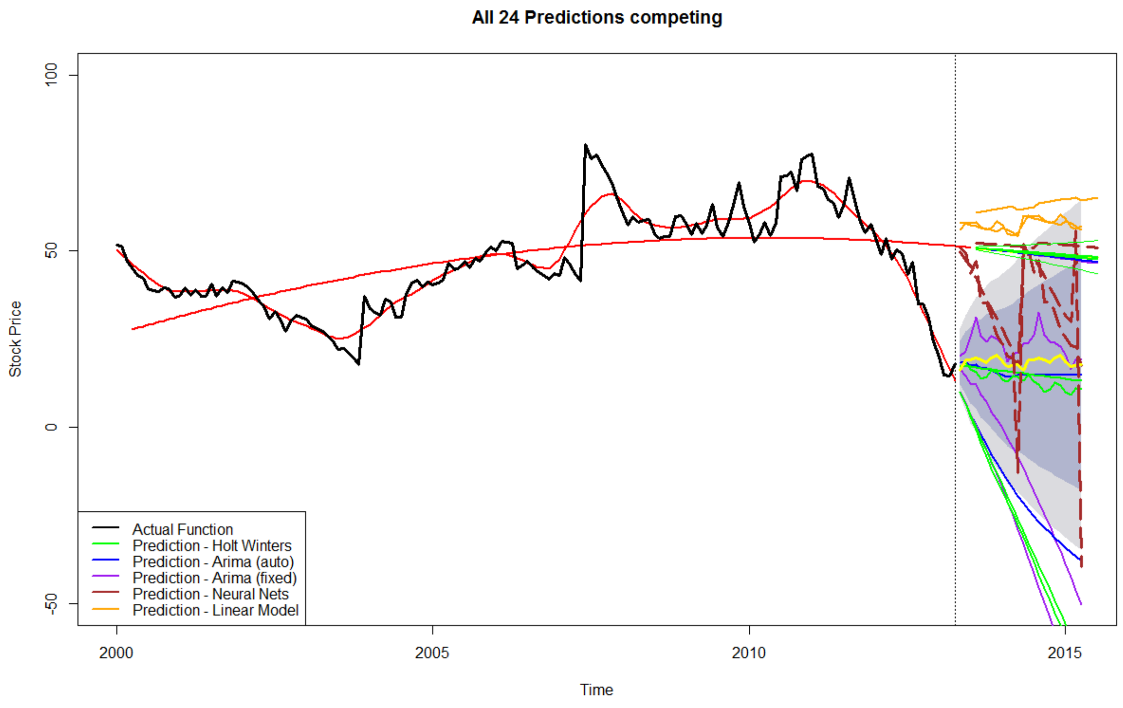


Also the predictions for the Actual Graph vary a lot more:



3. Evaluating the techniques

Before we begin with the evaluation, here are all predictions which are in the race:



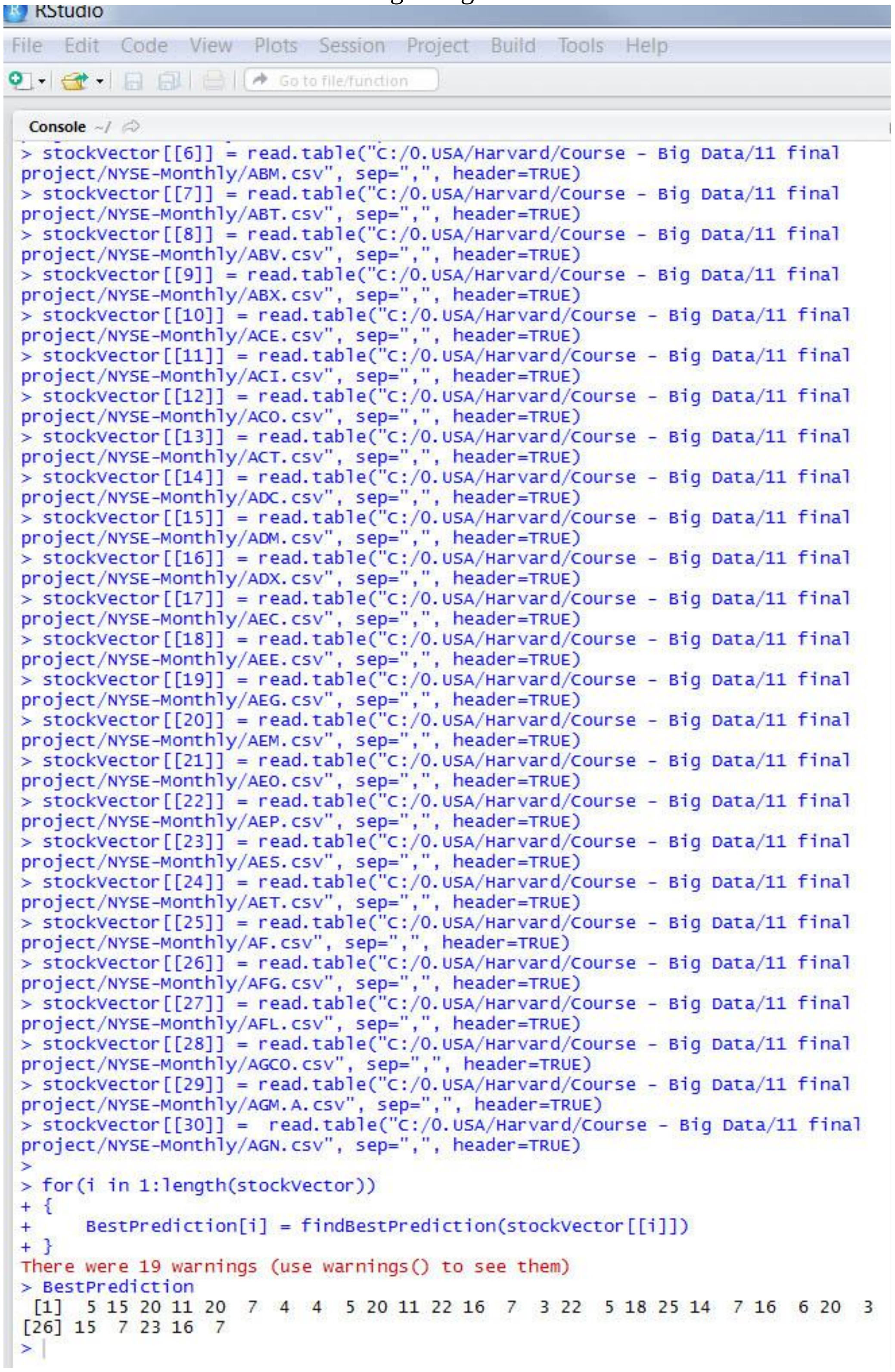
To evaluate this, I wrote a function *findBestPrediction.R*

Please open this attached function, to better understand my explanations.

The function is divided in mainly **4 steps**:

- Reading in the data and splitting it into train and test data
- Computing all 24 predictions of the various techniques using the three base functions of the train data
- Comparing the predictions with the actual test data, which gives as the error
- Choosing the prediction ID which gave the smallest total error
- This *findBestPrediction.R* is called in another script *runTestEvaluation.R*, which reads in 30 sample stocks, stores them in a vector and iteratively queries *findBestPrediction.R* while saving the result of each in another vector (*BestPrediction*).

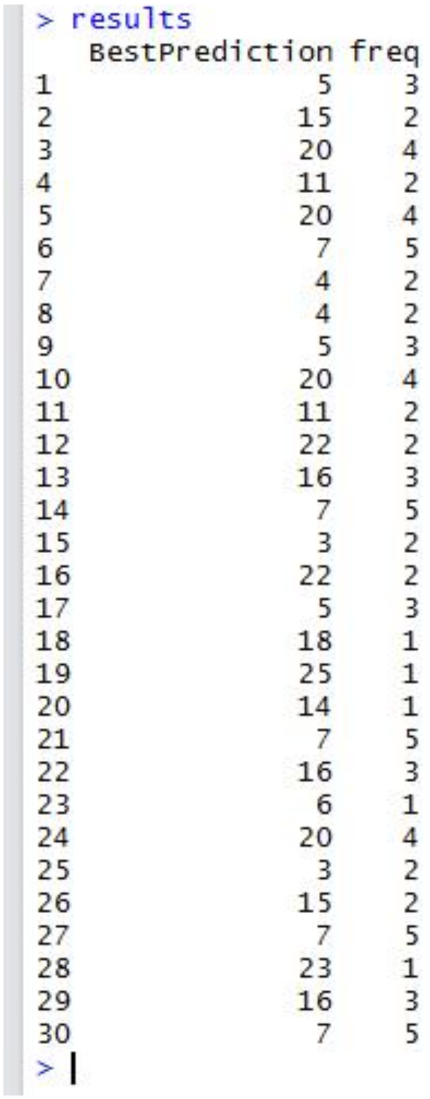
The results of the first 30 stocks beginning with the letter A are shown here:



```
> stockVector[[6]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ABM.csv", sep=",", header=TRUE)
> stockVector[[7]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ABT.csv", sep=",", header=TRUE)
> stockVector[[8]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ABV.csv", sep=",", header=TRUE)
> stockVector[[9]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ABX.csv", sep=",", header=TRUE)
> stockVector[[10]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ACE.csv", sep=",", header=TRUE)
> stockVector[[11]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ACI.csv", sep=",", header=TRUE)
> stockVector[[12]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ACO.csv", sep=",", header=TRUE)
> stockVector[[13]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ACT.csv", sep=",", header=TRUE)
> stockVector[[14]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ADC.csv", sep=",", header=TRUE)
> stockVector[[15]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ADM.csv", sep=",", header=TRUE)
> stockVector[[16]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/ADX.csv", sep=",", header=TRUE)
> stockVector[[17]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AEC.csv", sep=",", header=TRUE)
> stockVector[[18]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AEE.csv", sep=",", header=TRUE)
> stockVector[[19]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AEG.csv", sep=",", header=TRUE)
> stockVector[[20]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AEM.csv", sep=",", header=TRUE)
> stockVector[[21]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AEO.csv", sep=",", header=TRUE)
> stockVector[[22]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AEP.csv", sep=",", header=TRUE)
> stockVector[[23]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AES.csv", sep=",", header=TRUE)
> stockVector[[24]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AET.csv", sep=",", header=TRUE)
> stockVector[[25]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AF.csv", sep=",", header=TRUE)
> stockVector[[26]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AFG.csv", sep=",", header=TRUE)
> stockVector[[27]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AFL.csv", sep=",", header=TRUE)
> stockVector[[28]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AGCO.csv", sep=",", header=TRUE)
> stockVector[[29]] = read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AGM.A.csv", sep=",", header=TRUE)
> stockVector[[30]] =  read.table("C:/0.USA/Harvard/Course - Big Data/11 final
project/NYSE-Monthly/AGN.csv", sep=",", header=TRUE)
>
> for(i in 1:length(stockVector))
+ {
+     BestPrediction[i] = findBestPrediction(stockVector[[i]])
+ }
There were 19 warnings (use warnings() to see them)
> BestPrediction
 [1]  5 15 20 11 20  7  4  4  5 20 11 22 16  7  3 22  5 18 25 14  7 16  6 20  3
[26] 15  7 23 16  7
>
```

After some conversion (see below), we get the frequencies:

```
freq <- ave(rep(1, times=BestPrediction), BestPrediction, FUN=sum)
results = data.frame(BestPrediction,freq)
```



```
> results
   BestPrediction freq
1               5    3
2              15    2
3              20    4
4              11    2
5              20    4
6               7    5
7               4    2
8               4    2
9               5    3
10             20    4
11             11    2
12             22    2
13             16    3
14              7    5
15              3    2
16             22    2
17              5    3
18             18    1
19             25    1
20             14    1
21              7    5
22             16    3
23              6    1
24             20    4
25              3    2
26             15    2
27              7    5
28             23    1
29             16    3
30              7    5
> |
```

Now we have to look up the IDs of the prediction techniques in the *findBestPrediction.R* function and get the resulting ranking from 30 samples:

1. **ARIMA** model with auto parameters (5 times the best)

2. **Linear** model with trend, season components (4 times the best)

3. **Holt-Winters** Filtering (3 times the best)

   Holt-Winters Filtering with exponential smoothing (3 times the best)

4. Discussing the results and Outlook

Because only 30 stocks were used in the evaluation phase, the results are merely hints to which techniques are best suitable for stock predictions.

Using the attached scripts, however, it should not be a problem for you to try it out with 300 or more… ☺

Also it has to be said, that most of the used techniques are heavily dependent on the parameters used – here I also recommend trying out different values and/ or other parameters all together.

A future goal could be to find the top 10 stocks which are worth investing in. Then the evaluation test should be done again with maybe 500 stocks to acquire the most suitable prediction techniques. Afterwards these prediction methods can be used to predict all stocks which have historic data of at least 10 years and a **criterion** for promising trends then should select the most promising stocks. For example the most rapid and constant increase of stock price from the point when the prediction began to the prediction end is feasible.

You are free to use my scripts and methods as a starting point.

Let me know which stocks you found and which predicting method you used. ☺

Dietmar Aumann, 05/09/2013, Malmö (Sweden)