Stat 154 HW4

Mitali Yadav (3034158469)

```
install.packages("rpart")
install.packages("rpart.plot")
install.packages("ipred")
install.packages("randomForest")
install.packages("stats")
```

```
library("spls")
```

```
## Warning: package 'spls' was built under R version 4.0.4
```

```
## Sparse Partial Least Squares (SPLS) Regression and
## Classification (version 2.2-3)
```

```
library("plsr")
```

```
## Warning: package 'plsr' was built under R version 4.0.4
```

```
## Be aware that plsr 0.0.1 contains experimental and partly untested code.
## Use cautiously.
```

```
##
## Attaching package: 'plsr'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
## Warning: package 'stringr' was built under R version 4.0.3
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("caret")
```

```
## Warning: package 'caret' was built under R version 4.0.4
```

```
## Loading required package: lattice
```

```
## Registered S3 methods overwritten by 'caret':
##   method         from
##   predict.splsda spls
##   print.splsda   spls
```

```
##
## Attaching package: 'caret'
```

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

```
## The following object is masked from 'package:spls':
##
##     splsda
```

```
library("glmnet")
```

```
## Warning: package 'glmnet' was built under R version 4.0.4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
library("rpart")
```

```
## Warning: package 'rpart' was built under R version 4.0.4
```

```
library("rpart.plot")
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.4
```

```
library("ipred")
```

```
## Warning: package 'ipred' was built under R version 4.0.4
```

```
library("randomForest")
```

```
## Warning: package 'randomForest' was built under R version 4.0.4
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library("stats")
```

Q1

```
sp500 = read.csv("bf4024f5c75fc062.csv", header = T)
sp_vals = read.csv("sp500Data.csv")
head(sp500)
```

```
##   PERMNO     date TICKER      COMNAM BIDLO ASKHI   PRC      VOL OPENPRC NUMTRD
## 1  10104 20070904   ORCL ORACLE CORP 20.20 20.88 20.72 36786550  20.240  71139
## 2  10104 20070905   ORCL ORACLE CORP 20.46 20.85 20.73 30556173  20.490  71880
## 3  10104 20070906   ORCL ORACLE CORP 20.33 20.74 20.54 26450988  20.720  54322
## 4  10104 20070907   ORCL ORACLE CORP 19.97 20.40 20.16 29192043  20.220  61319
## 5  10104 20070910   ORCL ORACLE CORP 19.98 20.46 20.17 25314865  20.340  56737
## 6  10104 20070911   ORCL ORACLE CORP 20.15 20.52 20.46 20869439  20.227  44219
```

```
sp500redc = sp500[,c(2,3,7)]
```

```
#merging the 2 datasets
mega_data = merge(x=sp500redc, y=sp_vals, by.x = "date", by.y = "Calendar.Date", all = FALSE, sort = TRU

#number of unique values in date column
uniq_date = unique(mega_data$date)
```

```
sp500_unstack = unstack(mega_data, PRC~TICKER,)

sp500_matrix = matrix(sp500_unstack)

myMatrix<-matrix(sp500_unstack[[1]],nrow=1342)
vecNames<-names(sp500_unstack[1])
for(k in 2:530){
  if (length(sp500_unstack[[k]])==1342){
    myMatrix<-cbind(myMatrix,sp500_unstack[[k]])
    vecNames<-cbind(vecNames,names(sp500_unstack[k]))
  }
}

my_df = as.data.frame(myMatrix, row.names = vecNames)
names(my_df) = vecNames
my_df["date"] = uniq_date

#merging with sp_vals to get the corresponding sp.index values
my_df = merge(x=my_df, y=sp_vals, by.x = "date", by.y = "Calendar.Date", all=FALSE, suffixes = c("","_2
```

1

Creating a sparse portfolio that replicates the SP500 index.

```
set.seed(100)

#splitting data randomly into train and test set


#pre-processing the data
x_data = as.matrix(my_df[,-c(1,455)])

#finding the best value of lambda for LASSO regression
lambdas = 10^seq(2,-3, by=-0.1)

l1_m_temp = cv.glmnet(x=x_data, y=my_df$SP.500.Level, alpha=1, lambda = lambdas)

optimal_lambda_lasso = l1_m_temp$lambda.min
optimal_lambda_lasso
```

```
## [1] 0.001
```

```
#building the model using the optimal value of lambda
l1_model = glmnet(x_data, my_df$SP.500.Level, alpha=1, lambda = optimal_lambda_lasso)

round(l1_model$beta,2)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##          s0
## A      4.83
## AA     5.41
```

```
## AAPL   0.41
## ABC    0.54
## ABT    1.49
## ACE    1.97
## ACN    0.80
## ADBE   1.49
## ADI    2.66
## ADM    1.60
## ADP    0.30
## ADSK   0.78
## AEE    2.15
## AEP    1.61
## AES    1.52
## AET    0.03
## AFL    0.51
## AGN   -0.22
## AIG    0.29
## AIV   -0.09
## AIZ    0.17
## AKAM   0.10
## ALL   -0.52
## ALTR  -1.26
## ALXN  -0.14
## AMAT   1.55
## AMD    2.54
## AMGN  -0.34
## AMP   -0.14
## AMT   -0.04
## AMZN   0.01
## AN    -0.24
## ANF    0.29
## APA   -0.09
## APC   -0.27
## APD    0.07
## APH    0.41
## APOL   0.10
## ARG   -0.17
## ATI   -0.08
## AVB    0.14
## AVP   -0.05
## AVY    0.13
## AXP    0.14
## AZO    0.01
## BA    -0.02
## BAC    0.52
## BAX    0.48
## BBBY  -0.53
## BBT    0.61
## BBY   -0.08
## BCR    0.01
## BDX   -0.19
## BEN    0.32
## BHI   -0.33
## BIIB  -0.03
```

```
## BK     0.12
## BLK   -0.01
## BLL    0.02
## BMC   -0.20
## BMS   -0.04
## BMY    0.54
## BRCM  -0.19
## BSX   -2.27
## BTU    0.03
## BWA    0.06
## BXP    0.13
## C     -0.33
## CA    -0.95
## CAG   -0.48
## CAH    0.27
## CAM    0.10
## CAT    0.50
## CB    -1.01
## CBG   -0.43
## CCE    0.16
## CCI   -0.28
## CCL    0.10
## CELG   0.32
## CERN   0.18
## CF    -0.05
## CHK   -1.02
## CHRW   0.04
## CI     0.14
## CINF  -0.04
## CL    -0.06
## CLF   -0.02
## CLX    0.26
## CMA   -0.06
## CMCSA  0.74
## CME   -0.02
## CMI    0.02
## CMS   -1.73
## CMT   -0.13
## CNP    0.04
## CNX   -0.08
## COF    0.07
## COG    0.11
## COH    0.19
## COHU  -0.97
## COL    0.47
## COP    0.02
## COST  -0.28
## COV   -0.19
## CPB   -0.20
## CRM    0.00
## CSC    0.13
## CSCO   1.48
## CSX   -0.04
## CTAS  -0.34
```

```
## CTL    0.91
## CTSH   0.09
## CTXS  -0.11
## CVC    0.44
## CVS    0.68
## CVX    0.65
## D      0.12
## DD    -0.77
## DE    -0.17
## DELL  -0.60
## DFS   -0.54
## DGX   -0.47
## DHI   -0.14
## DHR    0.04
## DIS    0.52
## DISCA -0.31
## DLTR  -0.20
## DNB   -0.37
## DNR   -0.02
## DO    -0.01
## DOV    0.14
## DOW   -0.47
## DRI    0.65
## DTE   -0.17
## DTV    0.32
## DUK    0.02
## DVA    0.03
## DVN   -0.09
## EBAY   0.68
## EBIX  -0.27
## ECL    0.28
## ED    -0.54
## EFX    0.07
## EIX   -0.25
## EL     0.15
## EMC   -0.74
## EMN    0.02
## EMR   -0.07
## EOG   -0.06
## EQR   -0.36
## EQT   -0.09
## ESRX  -0.20
## ETFC   0.26
## ETN   -0.14
## ETR    0.48
## EW     0.04
## EXC    0.24
## EXPD  -0.45
## EXPE  -0.06
## F      0.18
## FAST   0.03
## FCX    0.06
## FDO   -0.19
## FDX    0.02
```

```
## FE    -0.61
## FFIV  -0.06
## FIS   -0.11
## FISV   0.69
## FITB   0.57
## FLIR  -0.18
## FLR   -0.09
## FLS   -0.03
## FMC   -0.05
## FOSL  -0.10
## FRX   -0.56
## FSLR  -0.02
## FTI    0.18
## GCI    0.28
## GD    -0.12
## GE     0.28
## GILD   0.06
## GIS   -0.09
## GLW    0.32
## GME    0.23
## GNW    0.43
## GOOG   0.00
## GPC    0.09
## GPS   -0.79
## GRMN  -0.34
## GS     0.06
## GT     0.05
## GWW    0.12
## HAL    0.80
## HAR   -0.21
## HAS   -0.18
## HBAN   0.49
## HCBK   0.10
## HCN   -0.13
## HCP    0.36
## HD    -0.19
## HES   -0.01
## HIG    0.08
## HOG   -0.09
## HON    0.32
## HOT   -0.22
## HP     0.06
## HPQ   -0.31
## HRB    0.47
## HRL   -0.25
## HRS   -0.31
## HSP    0.10
## HST   -0.62
## HSY    0.31
## HUM    0.04
## IBM   -0.06
## ICE   -0.04
## IFF   -0.21
## IGT   -0.16
```

```
## INTC   0.21
## INTU   0.14
## IP     0.68
## IPG   -2.86
## IR    -0.11
## IRM    0.05
## ISRG   0.00
## ITW   -0.10
## IVZ   -1.00
## JBL    .
## JCI    0.07
## JCP    0.91
## JDSU   0.27
## JEC    0.14
## JNJ    0.53
## JNPR  -0.11
## JPM    1.53
## JWN    0.12
## K     -0.15
## KEY   -1.08
## KIM    0.18
## KLAC   0.16
## KMB   -0.31
## KMX    0.02
## KO     0.11
## KR    -0.66
## KSS   -0.10
## KSU   -0.04
## LEG    0.44
## LH    -0.26
## LLL    0.24
## LLTC  -1.73
## LLY    0.14
## LM     0.13
## LMT   -0.01
## LNC   -0.30
## LOW    2.01
## LRCX  -0.01
## LSI   -2.30
## LTD    0.13
## LUK   -0.15
## LUV    0.56
## M      0.05
## MA     0.01
## MAC    0.06
## MAR    0.01
## MAS   -1.14
## MAT   -0.48
## MCD   -0.08
## MCHP   0.56
## MCK   -0.27
## MCO   -0.39
## MDT    0.88
## MET    0.20
```

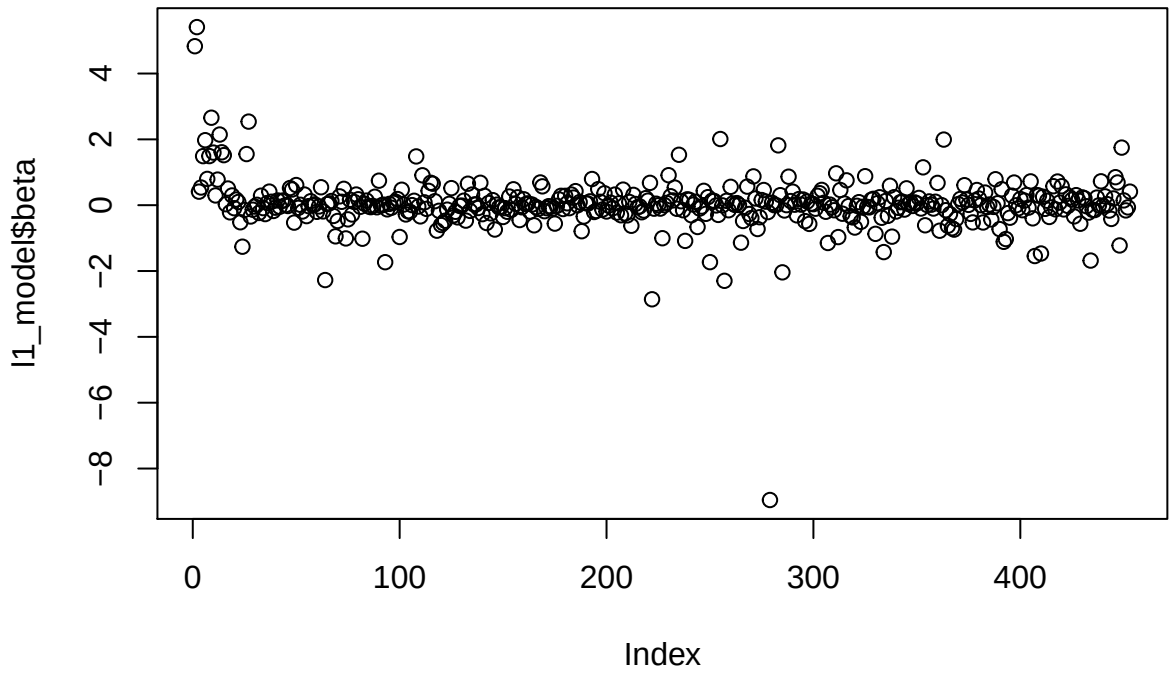
```
## MMC   -0.72
## MMM   -0.37
## MO     0.16
## MOLX   0.46
## MON    0.11
## MOS   -0.21
## MPET  -8.96
## MRK    0.07
## MRO    0.00
## MS     0.00
## MSFT   1.82
## MTB    0.30
## MU    -2.04
## MUR   -0.16
## MWV    0.00
## MYL    0.87
## NBL    0.12
## NBR    0.41
## NDAQ   0.02
## NE    -0.31
## NEM    0.19
## NFLX   0.02
## NFX    0.17
## NI    -0.48
## NKE    0.11
## NOC   -0.57
## NOV    0.05
## NRG    0.03
## NSC   -0.11
## NTAP   0.28
## NTRS   0.36
## NU     0.46
## NUE    0.07
## NVDA  -0.20
## NWL   -1.15
## NYX    0.02
## OI    -0.06
## OKE   -0.15
## OMC    0.97
## ORCL  -0.97
## ORLY   0.46
## OXY   -0.26
## PAYX   0.05
## PBCT   0.75
## PBI   -0.03
## PCAR  -0.33
## PCG   -0.37
## PCL   -0.68
## PCLN  -0.02
## PCP    0.09
## PDCO  -0.51
## PEG   -0.05
## PEP    0.88
## PETM  -0.05
```

```
## PFE   -0.07
## PFG    0.17
## PG     0.19
## PGR   -0.87
## PH     0.06
## PHM    0.24
## PIR   -0.38
## PKI   -1.42
## PLL    0.13
## PNC   -0.32
## PNW    0.59
## POM   -0.96
## PPG    0.24
## PPL   -0.19
## PRGO  -0.08
## PRU    0.09
## PSA    0.11
## PVH   -0.13
## PWR    0.51
## PX     0.04
## PXD    0.08
## QCOM  -0.02
## R      0.08
## RAI    0.05
## RDC    0.22
## REGN  -0.10
## RF     1.15
## RHI   -0.61
## RHT    0.00
## RL     0.11
## ROK    0.02
## ROP   -0.10
## ROST   0.10
## RRC    0.68
## RSG   -0.78
## RTN    0.00
## S      1.99
## SAI   -0.15
## SBUX  -0.63
## SCG   -0.27
## SCHW  -0.70
## SE    -0.74
## SEE   -0.42
## SHW    0.03
## SIAL   0.19
## SJM    0.08
## SLB    0.61
## SLM    0.17
## SNA    0.01
## SNDK  -0.26
## SO    -0.52
## SPG    0.14
## SPLS   0.46
## SRCL   0.20
```

```
## SRE    0.00
## STI   -0.53
## STJ    0.38
## STT   -0.06
## STX    0.02
## SWK   -0.43
## SWN   -0.05
## SWY    0.79
## SYK    0.06
## SYMC  -0.73
## SYY    0.49
## T     -1.11
## TE    -1.03
## TEG   -0.24
## TEL   -0.38
## TER    0.29
## TGT    0.69
## THC    0.00
## TIF   -0.10
## TJX    0.19
## TMK   -0.15
## TMO    0.14
## TROW   0.31
## TRV   -0.07
## TSN    0.72
## TSO   -0.40
## TSS   -1.54
## TWC    0.31
## TWX    0.29
## TXN   -1.47
## TXT    0.25
## TYC   -0.11
## UNH    0.13
## UNM   -0.36
## UNP   -0.07
## UPS    0.58
## URBN  -0.13
## USB    0.72
## UTX    0.08
## VAR    0.58
## VFC   -0.13
## VLO    0.33
## VMC   -0.02
## VNO    0.23
## VRSN   0.17
## VTR   -0.34
## VZ     0.31
## WAG    0.14
## WAT   -0.56
## WDC    0.24
## WEC    0.21
## WFC   -0.03
## WHR   -0.23
## WIN   -1.68
```

```
## WLP   -0.14
## WMB   -0.15
## WMT    0.23
## WPO    0.00
## WU     0.73
## WY    -0.02
## WYN    0.27
## WYNN   0.03
## X     -0.16
## XEL   -0.42
## XL     0.20
## XLNX   0.85
## XOM    0.69
## XRAY  -1.22
## XRX    1.75
## YHOO   0.15
## YUM   -0.16
## ZION  -0.07
## ZMH    0.41
```

```r
plot(l1_model$beta)
```



Here, we have chosen to use the L-1 norm with regression because lasso regression model's loss function is modified to minimize the complexity of the model by putting a restriction on the the sum of the absolute values of the model coefficients.
Since we do not have a test set to test our model on, we cannot evaluate the model quantitatively.

However, one good indicator of how well our model can approximate the SP500 index is the number of predictors that have a coefficient of 0 or close to zero.In the plot above, we can see that majority of the coefficients are near or at 0.
Another way to judge the quality of the model would be to calculate the mse (mean squared error).

2

Building a sparse linear model for the first 60 days

```
#creating the dataset for first 60 days
f60_df = as.data.frame(my_df[1:60,])
xdata_f60 = as.matrix(f60_df[,-c(1,455)])

l1_f60_m = cv.glmnet(xdata_f60, y=f60_df$SP.500.Level, alpha = 1, lambda = lambdas)
optimal_lambda_f60 = l1_f60_m$lambda.min
optimal_lambda_f60
```

```
## [1] 0.001
```

```
#building the model using the optimal value of lambda
l1_f60 = glmnet(xdata_f60, f60_df$SP.500.Level, alpha=1, lambda = optimal_lambda_f60)

round(l1_f60$beta,2)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##          s0
## A     18.72
## AA    16.05
## AAPL  -0.23
## ABC    1.09
## ABT   -0.20
## ACE    2.88
## ACN    1.70
## ADBE   0.41
## ADI    0.70
## ADM    0.37
## ADP    1.24
## ADSK   1.31
## AEE   -3.42
## AEP    0.35
## AES    1.19
## AET   -1.27
## AFL    0.15
## AGN    0.92
## AIG    0.00
## AIV   -0.43
## AIZ   -0.12
## AKAM   0.34
## ALL    0.10
## ALTR  -0.42
## ALXN   0.02
```

```
## AMAT   0.06
## AMD    1.86
## AMGN   0.42
## AMP   -0.96
## AMT   -1.43
## AMZN   0.00
## AN     1.54
## ANF   -0.18
## APA    0.18
## APC   -0.20
## APD   -0.02
## APH    0.05
## APOL   0.01
## ARG   -0.69
## ATI    0.26
## AVB    0.55
## AVP    0.37
## AVY   -0.01
## AXP   -0.02
## AZO   -0.02
## BA     0.19
## BAC   -0.79
## BAX    1.18
## BBBY   0.40
## BBT    0.43
## BBY   -1.54
## BCR   -0.67
## BDX    0.78
## BEN   -0.36
## BHI    0.27
## BIIB  -0.38
## BK     1.03
## BLK   -0.07
## BLL   -0.06
## BMC   -0.67
## BMS    4.47
## BMY    1.24
## BRCM  -0.49
## BSX    0.97
## BTU    0.67
## BWA   -0.15
## BXP    0.15
## C     -0.20
## CA    -3.16
## CAG   -1.64
## CAH    0.34
## CAM    0.01
## CAT   -0.10
## CB    -1.26
## CBG    0.09
## CCE    0.40
## CCI   -0.85
## CCL    0.58
## CELG   0.23
```

```
## CERN  -0.84
## CF     0.03
## CHK    0.88
## CHRW   0.42
## CI     0.15
## CINF   1.05
## CL     0.35
## CLF   -0.13
## CLX    0.26
## CMA    0.31
## CMCSA -0.54
## CME   -0.01
## CMI    0.12
## CMS   -5.53
## CMT   -9.36
## CNP    1.67
## CNX    0.24
## COF    0.37
## COG   -0.15
## COH   -0.17
## COHU  -0.12
## COL    0.46
## COP    0.18
## COST   0.27
## COV    0.86
## CPB   -3.64
## CRM   -0.73
## CSC    0.51
## CSCO  -0.86
## CSX   -0.20
## CTAS  -0.14
## CTL   -0.78
## CTSH   0.05
## CTXS   1.51
## CVC   -0.05
## CVS   -0.23
## CVX   -0.52
## D      0.03
## DD    -0.37
## DE    -0.25
## DELL   0.42
## DFS    0.15
## DGX    0.82
## DHI   -0.18
## DHR    0.55
## DIS   -0.37
## DISCA  0.69
## DLTR  -0.10
## DNB   -0.23
## DNR   -0.31
## DO    -0.17
## DOV   -0.50
## DOW    0.93
## DRI    0.01
```

```
## DTE    0.06
## DTV   -0.45
## DUK    3.15
## DVA   -0.27
## DVN    0.12
## EBAY   0.61
## EBIX  -0.11
## ECL    0.10
## ED     2.01
## EFX   -2.37
## EIX    0.10
## EL     0.59
## EMC    0.01
## EMN   -0.12
## EMR   -0.65
## EOG    0.19
## EQR    0.07
## EQT   -0.43
## ESRX  -0.02
## ETFC  -0.25
## ETN    0.06
## ETR    0.09
## EW    -0.98
## EXC    0.16
## EXPD  -0.45
## EXPE   0.00
## F      1.43
## FAST  -0.21
## FCX   -0.04
## FDO    0.12
## FDX    0.24
## FE     0.59
## FFIV  -0.04
## FIS   -0.60
## FISV   0.39
## FITB   0.57
## FLIR   0.03
## FLR   -0.07
## FLS    0.02
## FMC    0.05
## FOSL   0.84
## FRX   -0.81
## FSLR  -0.06
## FTI    0.22
## GCI   -0.13
## GD     0.11
## GE    -0.13
## GILD  -0.09
## GIS   -2.59
## GLW    1.31
## GME   -0.15
## GNW   -0.15
## GOOG   0.02
## GPC    1.58
```

```
## GPS   -0.80
## GRMN  -0.26
## GS     0.00
## GT    -0.03
## GWW    0.26
## HAL    0.39
## HAR   -0.07
## HAS   -0.37
## HBAN   1.85
## HCBK  -0.98
## HCN   -0.61
## HCP   -0.46
## HD    -0.03
## HES   -0.06
## HIG    0.01
## HOG   -0.54
## HON    0.18
## HOT    0.13
## HP    -0.41
## HPQ   -1.45
## HRB    1.18
## HRL    0.00
## HRS   -0.17
## HSP   -0.63
## HST   -0.16
## HSY   -0.47
## HUM    0.51
## IBM    0.26
## ICE   -0.01
## IFF   -0.88
## IGT   -0.12
## INTC  -0.12
## INTU  -0.43
## IP    -0.93
## IPG    2.32
## IR    -0.26
## IRM   -0.03
## ISRG   0.03
## ITW    0.88
## IVZ    0.44
## JBL    0.48
## JCI   -0.04
## JCP    0.02
## JDSU  -0.91
## JEC   -0.08
## JNJ   -0.63
## JNPR  -0.17
## JPM    0.86
## JWN   -0.21
## K     -0.20
## KEY    0.21
## KIM   -0.02
## KLAC  -0.15
## KMB    1.87
```

```
## KMX   -0.31
## KO    -0.38
## KR    -1.51
## KSS   -0.08
## KSU   -0.03
## LEG    0.53
## LH    -0.05
## LLL    0.08
## LLTC  -0.07
## LLY    0.17
## LM    -0.23
## LMT    0.22
## LNC    0.02
## LOW    0.11
## LRCX   0.15
## LSI   -1.46
## LTD    1.01
## LUK   -0.60
## LUV   -2.03
## M      0.75
## MA     0.01
## MAC    0.00
## MAR    0.05
## MAS    0.27
## MAT    0.96
## MCD   -0.18
## MCHP  -0.33
## MCK   -0.12
## MCO   -0.14
## MDT    0.05
## MET   -0.08
## MMC   -1.40
## MMM   -0.02
## MO     0.31
## MOLX  -1.23
## MON    0.01
## MOS    0.03
## MPET   4.84
## MRK    0.11
## MRO   -1.77
## MS     0.19
## MSFT   0.50
## MTB    0.14
## MU    -1.20
## MUR    0.29
## MWV   -0.76
## MYL    1.79
## NBL    0.15
## NBR    0.87
## NDAQ   0.21
## NE     0.28
## NEM    0.00
## NFLX   0.01
## NFX   -0.63
```

```
## NI     0.22
## NKE    0.39
## NOC    1.29
## NOV    0.02
## NRG    0.02
## NSC   -1.37
## NTAP   0.59
## NTRS  -0.13
## NU    -0.86
## NUE   -0.89
## NVDA  -0.08
## NWL    1.48
## NYX    0.06
## OI    -0.87
## OKE   -0.50
## OMC    0.14
## ORCL   2.67
## ORLY  -1.02
## OXY    0.04
## PAYX  -0.43
## PBCT   1.74
## PBI   -0.02
## PCAR  -0.03
## PCG   -0.37
## PCL    0.01
## PCLN  -0.04
## PCP   -0.12
## PDCO   0.45
## PEG    0.17
## PEP   -0.23
## PETM  -0.09
## PFE    0.51
## PFG   -0.23
## PG     0.28
## PGR    0.91
## PH    -0.04
## PHM    0.05
## PIR   -0.30
## PKI   -1.31
## PLL    0.51
## PNC   -0.13
## PNW    0.22
## POM    0.00
## PPG    0.23
## PPL    0.37
## PRGO   0.16
## PRU   -0.22
## PSA    0.06
## PVH    0.23
## PWR   -0.02
## PX    -0.32
## PXD    0.38
## QCOM  -0.06
## R     -0.42
```

```
## RAI   -0.07
## RDC    0.14
## REGN  -0.55
## RF     0.39
## RHI   -0.07
## RHT   -1.57
## RL    -0.32
## ROK    0.60
## ROP    0.34
## ROST  -0.79
## RRC    0.06
## RSG   -2.34
## RTN    1.60
## S     -0.19
## SAI   -1.06
## SBUX  -0.09
## SCG    0.05
## SCHW  -0.23
## SE     0.20
## SEE   -0.29
## SHW    0.13
## SIAL   0.14
## SJM   -1.35
## SLB    0.08
## SLM    0.43
## SNA    1.18
## SNDK  -0.27
## SO    -0.90
## SPG   -0.11
## SPLS   1.08
## SRCL  -0.45
## SRE   -0.03
## STI    0.04
## STJ    0.14
## STT    0.28
## STX   -3.44
## SWK    0.15
## SWN    0.31
## SWY   -1.34
## SYK   -0.23
## SYMC   0.52
## SYY   -0.31
## T     -0.03
## TE     0.75
## TEG   -0.09
## TEL   -0.72
## TER    0.22
## TGT    0.24
## THC   -0.64
## TIF   -0.01
## TJX    0.06
## TMK    1.19
## TMO   -0.33
## TROW   0.07
```

```
## TRV    0.59
## TSN    0.08
## TSO   -0.05
## TSS   -0.18
## TWC   -0.31
## TWX    0.06
## TXN   -0.15
## TXT   -0.21
## TYC    0.26
## UNH    0.55
## UNM    1.77
## UNP   -0.07
## UPS    0.32
## URBN  -0.65
## USB   -0.75
## UTX   -0.34
## VAR    0.02
## VFC    0.07
## VLO   -0.42
## VMC    0.17
## VNO   -0.05
## VRSN   0.00
## VTR    0.44
## VZ     0.23
## WAG   -0.07
## WAT    0.03
## WDC    0.68
## WEC   -0.74
## WFC   -0.11
## WHR   -0.04
## WIN    1.71
## WLP    1.39
## WMB    0.12
## WMT    0.06
## WPO    0.03
## WU     0.08
## WY    -0.72
## WYN    0.13
## WYNN   0.15
## X     -0.21
## XEL   -0.26
## XL     0.13
## XLNX  -0.42
## XOM    0.00
## XRAY   1.17
## XRX   -0.97
## YHOO  -0.07
## YUM   -0.33
## ZION   0.04
## ZMH    0.04
```

Building a sparse linear model for the last 60 days

```r
l60_df = as.data.frame(my_df[1283:1342,])
dim(l60_df)
```

```
## [1]  60 455
```

```r
xdata_l60 = as.matrix(l60_df[,-c(1,455)])

l1_l60_m = cv.glmnet(xdata_l60, y=l60_df$SP.500.Level, alpha = 1, lambda = lambdas)
optimal_lambda_l60 = l1_l60_m$lambda.min
optimal_lambda_l60
```

```
## [1] 0.001
```

```r
#building the model using the optimal value of lambda
l1_l60 = glmnet(xdata_l60, l60_df$SP.500.Level, alpha=1, lambda = optimal_lambda_l60)

round(l1_l60$beta,2)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##          s0
## A      6.06
## AA    63.19
## AAPL   0.15
## ABC    1.08
## ABT    0.10
## ACE    1.59
## ACN    0.25
## ADBE  -0.87
## ADI    0.58
## ADM    0.53
## ADP   -0.35
## ADSK  -0.26
## AEE   -0.74
## AEP    0.17
## AES    0.14
## AET    0.43
## AFL   -0.21
## AGN    0.26
## AIG   -0.45
## AIV   -0.69
## AIZ   -0.04
## AKAM  -0.11
## ALL    0.95
## ALTR  -0.44
## ALXN  -0.04
## AMAT   1.04
## AMD   -1.48
## AMGN   0.25
## AMP   -0.17
## AMT    0.20
## AMZN   0.03
## AN     0.25
```

```
## ANF    0.12
## APA    0.04
## APC    0.00
## APD   -0.21
## APH   -0.19
## APOL   .
## ARG    0.09
## ATI   -0.46
## AVB   -0.03
## AVP    1.58
## AVY    0.96
## AXP    0.30
## AZO    0.20
## BA     0.57
## BAC    0.08
## BAX    0.25
## BBBY   0.71
## BBT    0.30
## BBY   -0.39
## BCR    0.28
## BDX    0.08
## BEN    0.18
## BHI   -0.29
## BIIB  -0.14
## BK     0.04
## BLK   -0.02
## BLL   -0.80
## BMC   -1.04
## BMS    0.27
## BMY    0.44
## BRCM   0.64
## BSX    2.44
## BTU    0.10
## BWA   -0.25
## BXP    0.12
## C      0.20
## CA    -0.18
## CAG    0.14
## CAH   -1.42
## CAM    0.14
## CAT   -0.40
## CB    -0.03
## CBG   -0.09
## CCE    1.79
## CCI    0.65
## CCL   -0.58
## CELG  -0.34
## CERN  -0.09
## CF    -0.10
## CHK    0.57
## CHRW   0.00
## CI     0.35
## CINF   0.59
## CL     0.41
```

```
## CLF   -0.34
## CLX   -0.67
## CMA    1.59
## CMCSA  0.72
## CME    0.20
## CMI   -0.17
## CMS    0.95
## CMT   -6.52
## CNP    0.34
## CNX   -0.61
## COF   -0.30
## COG    0.13
## COH    0.10
## COHU  -0.07
## COL   -0.16
## COP    .
## COST   0.00
## COV   -0.46
## CPB    0.04
## CRM   -0.01
## CSC   -0.30
## CSCO   0.25
## CSX   -0.61
## CTAS   0.10
## CTL   -1.77
## CTSH   0.04
## CTXS   0.29
## CVC    0.12
## CVS   -0.12
## CVX   -0.19
## D      0.82
## DD    -0.23
## DE    -0.69
## DELL   1.34
## DFS   -0.66
## DGX   -0.04
## DHI    0.24
## DHR    0.07
## DIS   -0.08
## DISCA -0.20
## DLTR  -0.03
## DNB    0.02
## DNR   -0.71
## DO     0.50
## DOV   -0.04
## DOW   -0.47
## DRI   -0.08
## DTE    0.03
## DTV    0.26
## DUK   -0.50
## DVA    0.07
## DVN    0.00
## EBAY   0.26
## EBIX   0.10
```

```
## ECL    .
## ED    -0.05
## EFX    0.10
## EIX    0.04
## EL     0.27
## EMC   -0.33
## EMN    0.06
## EMR    0.03
## EOG   -0.05
## EQR   -0.02
## EQT    0.20
## ESRX   0.01
## ETFC   0.31
## ETN    0.15
## ETR    0.01
## EW     0.16
## EXC   -0.12
## EXPD   0.05
## EXPE   0.03
## F     -0.77
## FAST  -0.16
## FCX    0.00
## FDO   -0.14
## FDX    1.16
## FE    -0.49
## FFIV  -0.02
## FIS   -0.63
## FISV  -0.38
## FITB  -0.07
## FLIR   0.25
## FLR   -0.09
## FLS    0.01
## FMC   -0.44
## FOSL   0.08
## FRX    0.12
## FSLR   0.06
## FTI   -0.09
## GCI    0.06
## GD     0.06
## GE    -0.74
## GILD   0.37
## GIS   -0.58
## GLW    0.04
## GME   -0.04
## GNW   -0.56
## GOOG   0.02
## GPC    0.38
## GPS    0.29
## GRMN  -0.11
## GS     0.03
## GT    -1.25
## GWW    0.00
## HAL    0.03
## HAR    0.00
```

```
## HAS   -0.06
## HBAN  -0.21
## HCBK  -0.34
## HCN    0.32
## HCP    0.57
## HD    -0.16
## HES   -0.31
## HIG    0.88
## HOG   -0.04
## HON    0.33
## HOT    0.03
## HP     0.02
## HPQ   -0.87
## HRB   -1.12
## HRL    0.36
## HRS    0.36
## HSP    0.04
## HST    0.60
## HSY    0.28
## HUM    0.13
## IBM    0.05
## ICE    0.19
## IFF    0.02
## IGT   -0.24
## INTC  -0.72
## INTU   0.23
## IP    -0.30
## IPG   -0.26
## IR     0.69
## IRM   -0.17
## ISRG  -0.02
## ITW   -0.15
## IVZ   -0.56
## JBL   -0.99
## JCI    0.42
## JCP    0.08
## JDSU  -0.44
## JEC    0.17
## JNJ    0.08
## JNPR   0.01
## JPM    0.42
## JWN   -0.02
## K     -0.08
## KEY   -1.40
## KIM   -1.08
## KLAC  -0.16
## KMB    0.74
## KMX    0.00
## KO     0.07
## KR    -0.48
## KSS   -0.02
## KSU    0.26
## LEG   -0.07
## LH     0.17
```

```
## LLL   -0.31
## LLTC  -0.18
## LLY   -0.55
## LM     0.37
## LMT    0.10
## LNC   -0.31
## LOW   -0.09
## LRCX   0.09
## LSI    2.64
## LTD    0.15
## LUK   -0.29
## LUV    0.62
## M     -0.17
## MA    -0.07
## MAC    0.87
## MAR   -0.48
## MAS   -0.24
## MAT    0.65
## MCD    0.22
## MCHP  -0.13
## MCK   -0.11
## MCO    0.39
## MDT   -0.01
## MET    0.06
## MMC    0.18
## MMM   -0.49
## MO     0.49
## MOLX  -0.06
## MON   -0.33
## MOS    0.03
## MPET  34.61
## MRK   -0.25
## MRO    0.96
## MS    -0.17
## MSFT  -0.13
## MTB   -0.13
## MU     1.33
## MUR   -0.18
## MWV   -0.11
## MYL   -0.36
## NBL    0.13
## NBR    0.69
## NDAQ   1.09
## NE    -0.01
## NEM    0.10
## NFLX  -0.02
## NFX    0.00
## NI    -0.71
## NKE    0.06
## NOC    0.09
## NOV   -0.06
## NRG   -0.16
## NSC    0.12
## NTAP  -0.16
```

```
## NTRS   0.27
## NU    -0.27
## NUE   -1.10
## NVDA  -1.31
## NWL    0.81
## NYX    0.18
## OI     0.44
## OKE   -0.19
## OMC    0.30
## ORCL   0.26
## ORLY  -0.16
## OXY   -0.22
## PAYX   0.67
## PBCT  -4.98
## PBI    0.49
## PCAR  -0.13
## PCG    0.34
## PCL    0.52
## PCLN  -0.01
## PCP    0.08
## PDCO  -1.55
## PEG    0.07
## PEP    0.72
## PETM   0.58
## PFE   -2.04
## PFG   -1.40
## PG     0.05
## PGR    1.19
## PH     0.11
## PHM    0.45
## PIR   -1.85
## PKI   -0.69
## PLL    0.31
## PNC    0.11
## PNW   -0.05
## POM    1.89
## PPG   -0.01
## PPL   -1.30
## PRGO  -0.01
## PRU    0.12
## PSA   -0.11
## PVH    0.09
## PWR    0.15
## PX    -0.20
## PXD   -0.12
## QCOM   0.26
## R     -0.07
## RAI    0.01
## RDC    0.06
## REGN  -0.03
## RF     1.96
## RHI    0.66
## RHT    0.15
## RL     0.17
```

```
## ROK   -0.01
## ROP   -0.17
## ROST  -0.04
## RRC   -0.13
## RSG   -0.10
## RTN    0.18
## S      3.58
## SAI   -1.21
## SBUX  -0.04
## SCG   -0.05
## SCHW   1.01
## SE    -0.18
## SEE   -0.20
## SHW   -0.17
## SIAL  -0.62
## SJM    0.50
## SLB    0.87
## SLM   -3.50
## SNA    0.02
## SNDK  -0.06
## SO    -0.20
## SPG   -0.02
## SPLS  -0.65
## SRCL  -0.36
## SRE   -0.03
## STI   -0.19
## STJ    0.04
## STT    0.18
## STX    0.17
## SWK    0.18
## SWN   -0.43
## SWY   -0.85
## SYK   -0.03
## SYMC   0.62
## SYY   -0.43
## T      0.14
## TE    -0.69
## TEG   -0.03
## TEL   -0.25
## TER    0.14
## TGT    0.83
## THC    0.03
## TIF   -0.26
## TJX   -0.10
## TMK   -0.02
## TMO    0.02
## TROW   0.82
## TRV   -0.35
## TSN   -0.34
## TSO   -0.03
## TSS   -0.71
## TWC    0.09
## TWX   -0.15
## TXN    0.16
```

```
## TXT    1.37
## TYC    1.23
## UNH   -0.33
## UNM   -1.49
## UNP    0.40
## UPS    0.04
## URBN   0.00
## USB   -0.43
## UTX   -0.10
## VAR   -0.07
## VFC   -0.04
## VLO   -0.28
## VMC    0.13
## VNO    0.00
## VRSN  -0.02
## VTR   -0.58
## VZ    -0.11
## WAG   -0.10
## WAT   -0.04
## WDC    0.09
## WEC   -0.04
## WFC    0.14
## WHR    0.10
## WIN    0.20
## WLP    0.07
## WMB    0.13
## WMT    0.08
## WPO    0.06
## WU    -0.03
## WY    -0.92
## WYN   -0.01
## WYNN   0.16
## X     -1.20
## XEL   -0.83
## XL     0.59
## XLNX   0.44
## XOM   -0.08
## XRAY   0.25
## XRX    0.47
## YHOO  -0.14
## YUM    0.02
## ZION   0.73
## ZMH    0.05
```

To check the stability of the portfolios, we will compare the coefficients of the 2 models to see the similarities or differences in the coefficients.

```
v1 = round(l1_f60$beta,4)
v2 = round(l1_l60$beta,4)
v1-v2
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##             s0
## A      12.6584
```

```
## AA    -47.1440
## AAPL   -0.3854
## ABC     0.0125
## ABT    -0.3034
## ACE     1.2840
## ACN     1.4504
## ADBE    1.2855
## ADI     0.1246
## ADM    -0.1614
## ADP     1.5909
## ADSK    1.5763
## AEE    -2.6744
## AEP     0.1837
## AES     1.0596
## AET    -1.6958
## AFL     0.3610
## AGN     0.6540
## AIG     0.4549
## AIV     0.2625
## AIZ    -0.0796
## AKAM    0.4483
## ALL    -0.8506
## ALTR    0.0208
## ALXN    0.0542
## AMAT   -0.9841
## AMD     3.3424
## AMGN    0.1673
## AMP    -0.7909
## AMT    -1.6276
## AMZN   -0.0310
## AN      1.2947
## ANF    -0.2977
## APA     0.1445
## APC    -0.1990
## APD     0.1916
## APH     0.2438
## APOL    0.0131
## ARG    -0.7827
## ATI     0.7185
## AVB     0.5751
## AVP    -1.2086
## AVY    -0.9745
## AXP    -0.3200
## AZO    -0.2140
## BA     -0.3744
## BAC    -0.8730
## BAX     0.9303
## BBBY   -0.3104
## BBT     0.1325
## BBY    -1.1478
## BCR    -0.9514
## BDX     0.7036
## BEN    -0.5424
## BHI     0.5549
```

```
## BIIB   -0.2396
## BK      0.9929
## BLK    -0.0483
## BLL     0.7373
## BMC     0.3678
## BMS     4.1989
## BMY     0.8034
## BRCM   -1.1282
## BSX    -1.4701
## BTU     0.5724
## BWA     0.1050
## BXP     0.0303
## C      -0.3975
## CA     -2.9818
## CAG    -1.7815
## CAH     1.7673
## CAM    -0.1228
## CAT     0.2971
## CB     -1.2368
## CBG     0.1734
## CCE    -1.3899
## CCI    -1.4931
## CCL     1.1673
## CELG    0.5672
## CERN   -0.7517
## CF      0.1304
## CHK     0.3096
## CHRW    0.4255
## CI     -0.2025
## CINF    0.4531
## CL     -0.0658
## CLF     0.2117
## CLX     0.9333
## CMA    -1.2790
## CMCSA  -1.2615
## CME    -0.2149
## CMI     0.2875
## CMS    -6.4788
## CMT    -2.8410
## CNP     1.3371
## CNX     0.8482
## COF     0.6715
## COG    -0.2821
## COH    -0.2741
## COHU   -0.0556
## COL     0.6208
## COP     0.1805
## COST    0.2661
## COV     1.3212
## CPB    -3.6727
## CRM    -0.7192
## CSC     0.8100
## CSCO   -1.1055
## CSX     0.4035
```

```
## CTAS   -0.2404
## CTL     0.9986
## CTSH    0.0101
## CTXS    1.2160
## CVC    -0.1689
## CVS    -0.1075
## CVX    -0.3367
## D      -0.7857
## DD     -0.1328
## DE      0.4475
## DELL   -0.9145
## DFS     0.8115
## DGX     0.8612
## DHI    -0.4246
## DHR     0.4785
## DIS    -0.2870
## DISCA   0.8868
## DLTR   -0.0710
## DNB    -0.2523
## DNR     0.3916
## DO     -0.6722
## DOV    -0.4509
## DOW     1.3984
## DRI     0.0934
## DTE     0.0363
## DTV    -0.7118
## DUK     3.6430
## DVA    -0.3394
## DVN     0.1166
## EBAY    0.3485
## EBIX   -0.2086
## ECL     0.0956
## ED      2.0563
## EFX    -2.4722
## EIX     0.0627
## EL      0.3203
## EMC     0.3375
## EMN    -0.1772
## EMR    -0.6834
## EOG     0.2445
## EQR     0.0970
## EQT    -0.6317
## ESRX   -0.0304
## ETFC   -0.5517
## ETN    -0.0916
## ETR     0.0774
## EW     -1.1389
## EXC     0.2847
## EXPD   -0.5049
## EXPE   -0.0314
## F       2.1946
## FAST   -0.0461
## FCX    -0.0368
## FDO     0.2607
```

```
## FDX    -0.9239
## FE      1.0782
## FFIV   -0.0179
## FIS     0.0264
## FISV    0.7634
## FITB    0.6349
## FLIR   -0.2174
## FLR     0.0208
## FLS     0.0069
## FMC     0.4989
## FOSL    0.7584
## FRX    -0.9298
## FSLR   -0.1152
## FTI     0.3123
## GCI    -0.1868
## GD      0.0564
## GE      0.6129
## GILD   -0.4577
## GIS    -2.0152
## GLW     1.2639
## GME    -0.1138
## GNW     0.4132
## GOOG   -0.0084
## GPC     1.1993
## GPS    -1.0979
## GRMN   -0.1532
## GS     -0.0326
## GT      1.2245
## GWW     0.2630
## HAL     0.3631
## HAR    -0.0787
## HAS    -0.3110
## HBAN    2.0665
## HCBK   -0.6450
## HCN    -0.9260
## HCP    -1.0247
## HD      0.1315
## HES     0.2579
## HIG    -0.8638
## HOG    -0.5016
## HON    -0.1476
## HOT     0.1030
## HP     -0.4327
## HPQ    -0.5860
## HRB     2.3006
## HRL    -0.3647
## HRS    -0.5353
## HSP    -0.6683
## HST    -0.7665
## HSY    -0.7490
## HUM     0.3774
## IBM     0.2158
## ICE    -0.2001
## IFF    -0.9082
```

```
## IGT     0.1198
## INTC    0.6039
## INTU   -0.6554
## IP     -0.6310
## IPG     2.5881
## IR     -0.9414
## IRM     0.1431
## ISRG    0.0434
## ITW     1.0343
## IVZ     0.9949
## JBL     1.4656
## JCI    -0.4596
## JCP    -0.0612
## JDSU   -0.4679
## JEC    -0.2492
## JNJ    -0.7038
## JNPR   -0.1729
## JPM     0.4417
## JWN    -0.1858
## K      -0.1218
## KEY     1.6156
## KIM     1.0644
## KLAC    0.0063
## KMB     1.1249
## KMX    -0.3051
## KO     -0.4508
## KR     -1.0254
## KSS    -0.0561
## KSU    -0.2851
## LEG     0.5975
## LH     -0.2172
## LLL     0.3908
## LLTC    0.1150
## LLY     0.7184
## LM     -0.5987
## LMT     0.1163
## LNC     0.3288
## LOW     0.2044
## LRCX    0.0660
## LSI    -4.1026
## LTD     0.8659
## LUK    -0.3104
## LUV    -2.6517
## M       0.9168
## MA      0.0809
## MAC    -0.8673
## MAR     0.5316
## MAS     0.5013
## MAT     0.3134
## MCD    -0.3929
## MCHP   -0.1963
## MCK    -0.0034
## MCO    -0.5278
## MDT     0.0637
```

```
## MET    -0.1394
## MMC    -1.5815
## MMM     0.4677
## MO     -0.1836
## MOLX   -1.1700
## MON     0.3309
## MOS     0.0069
## MPET  -29.7710
## MRK     0.3654
## MRO    -2.7283
## MS      0.3568
## MSFT    0.6311
## MTB     0.2693
## MU     -2.5389
## MUR     0.4670
## MWV    -0.6546
## MYL     2.1556
## NBL     0.0245
## NBR     0.1862
## NDAQ   -0.8733
## NE      0.2925
## NEM    -0.1016
## NFLX    0.0330
## NFX    -0.6294
## NI      0.9289
## NKE     0.3331
## NOC     1.1978
## NOV     0.0838
## NRG     0.1749
## NSC    -1.4805
## NTAP    0.7517
## NTRS   -0.4016
## NU     -0.5964
## NUE     0.2077
## NVDA    1.2326
## NWL     0.6728
## NYX    -0.1270
## OI     -1.3196
## OKE    -0.3039
## OMC    -0.1687
## ORCL    2.4078
## ORLY   -0.8557
## OXY     0.2576
## PAYX   -1.0985
## PBCT    6.7172
## PBI    -0.5143
## PCAR    0.0977
## PCG    -0.7180
## PCL    -0.5096
## PCLN   -0.0288
## PCP    -0.2019
## PDCO    1.9997
## PEG     0.1020
## PEP    -0.9482
```

```
## PETM   -0.6615
## PFE     2.5420
## PFG     1.1681
## PG      0.2269
## PGR    -0.2794
## PH     -0.1500
## PHM    -0.4026
## PIR     1.5450
## PKI    -0.6201
## PLL     0.2024
## PNC    -0.2320
## PNW     0.2689
## POM    -1.8933
## PPG     0.2406
## PPL     1.6794
## PRGO    0.1685
## PRU    -0.3437
## PSA     0.1781
## PVH     0.1407
## PWR    -0.1676
## PX     -0.1233
## PXD     0.5035
## QCOM   -0.3135
## R      -0.3489
## RAI    -0.0799
## RDC     0.0805
## REGN   -0.5158
## RF     -1.5767
## RHI    -0.7253
## RHT    -1.7159
## RL     -0.4891
## ROK     0.6101
## ROP     0.5049
## ROST   -0.7530
## RRC     0.1880
## RSG    -2.2390
## RTN     1.4240
## S      -3.7724
## SAI     0.1585
## SBUX   -0.0501
## SCG     0.0990
## SCHW   -1.2338
## SE      0.3780
## SEE    -0.0851
## SHW     0.3010
## SIAL    0.7621
## SJM    -1.8458
## SLB    -0.7969
## SLM     3.9263
## SNA     1.1642
## SNDK   -0.2087
## SO     -0.7038
## SPG    -0.0961
## SPLS    1.7274
```

```
## SRCL   -0.0935
## SRE     0.0020
## STI     0.2338
## STJ     0.1017
## STT     0.0980
## STX    -3.6106
## SWK    -0.0336
## SWN     0.7370
## SWY    -0.4864
## SYK    -0.1985
## SYMC   -0.1004
## SYY     0.1234
## T      -0.1705
## TE      1.4455
## TEG    -0.0623
## TEL    -0.4655
## TER     0.0754
## TGT    -0.5912
## THC    -0.6751
## TIF     0.2473
## TJX     0.1574
## TMK     1.2047
## TMO    -0.3506
## TROW   -0.7547
## TRV     0.9409
## TSN     0.4178
## TSO    -0.0184
## TSS     0.5305
## TWC    -0.3948
## TWX     0.2074
## TXN    -0.3140
## TXT    -1.5830
## TYC    -0.9794
## UNH     0.8736
## UNM     3.2592
## UNP    -0.4677
## UPS     0.2815
## URBN   -0.6498
## USB    -0.3181
## UTX    -0.2428
## VAR     0.0872
## VFC     0.1072
## VLO    -0.1440
## VMC     0.0334
## VNO    -0.0547
## VRSN    0.0154
## VTR     1.0278
## VZ      0.3423
## WAG     0.0263
## WAT     0.0700
## WDC     0.5900
## WEC    -0.6994
## WFC    -0.2462
## WHR    -0.1373
```

```
## WIN     1.5095
## WLP     1.3244
## WMB    -0.0046
## WMT    -0.0158
## WPO    -0.0325
## WU      0.1092
## WY      0.1977
## WYN     0.1358
## WYNN   -0.0012
## X       0.9883
## XEL     0.5642
## XL     -0.4631
## XLNX   -0.8563
## XOM     0.0825
## XRAY    0.9175
## XRX    -1.4407
## YHOO    0.0621
## YUM    -0.3495
## ZION   -0.6908
## ZMH    -0.0127
```

Based on v1, v2 and v1-v2 we can see that there is a drastic difference in the 2 model coefficients and for those which are closer to one another, the values are close to zero hence the small difference.
Thus the portfolios are not stable.
In order to ensure that the portfolio changes little over time, we need to add penalties. One such penalty could be the addition of a penalty that is proportional to the sum of the absolute values of the portfolio weights. This would encourage sparse portfolios.
Another penalty to add would be the mean tracking error. $mean\_tracking\_error = Expected\_return - (Risk\_Tolerance * Variance)$

3

In case of the optimization problem, assuming we have N number of companies/securities to invest in.
w1,w2....wN are the weights of the securitites
μ is expected return We can express portfolio return μ_p and risk $\sigma_p(w)$ as:
$\mu_p(w) = w^T \mu$
$\sigma(w) = sqrt(w^T V w)$

If we cannot perform 'shorting' there is an additional constraint of non-negative values:
$x_i >= 0 \; \forall \; i = 1, 2, 3...N$ *or* $x >= 0$
if r_i is the expected return on a stock i
then the final optimization problem is

$\sum_{i=1}^{n} r_i * x_i \; \geq \; r_{min}$ where r_{min} is the minimum expected return.

4

Now we create a new column called Returns to add to the sp500 dataset

```
#adding the column
my_df$Returns = ave(my_df$SP.500.Level, FUN = function(x) c(0, diff(x)))
my_df$Returns
```

```
##    [1]    0.00  -17.13    6.26  -25.00   -1.85   19.79    0.07   12.39    0.30
##   [10]   -7.60   43.13    9.25  -10.28    7.00   -8.02   -0.52    8.21    5.96
##   [19]   -4.63   20.29   -0.41   -7.04    3.25   14.75   -5.01   12.57   -2.68
##   [28]   -8.06    7.39  -13.09  -10.18    2.71   -1.16  -39.45    5.70   13.26
##   [37]   -3.71   -1.48   20.88    5.70   -9.96   18.36  -40.94    1.21   -7.48
##   [46]   18.10  -44.65   -0.85  -21.07  -14.52   41.87  -10.47  -19.43    7.59
##   [55]  -25.47    6.43  -22.93   23.93  -33.48   21.01   40.79    0.70   11.42
##   [64]   -8.72   -9.63   22.22   22.33   -2.68   11.30  -38.31    8.94    1.82
##   [73]  -20.46  -22.05    9.08   -1.98    7.12   24.34   11.99    1.21  -21.29
##   [82]    2.12  -10.13  -21.20    0.00  -35.53    4.55  -25.99   18.94   11.20
##   [91]  -19.31   15.23  -35.30   -7.75  -39.95   -8.06  -14.69   28.10   13.47
##  [100]  -21.46   23.36    8.33   -6.49   22.74   16.87  -14.60  -44.18  -10.19
##  [109]   10.46   -5.62    7.84    9.73   18.35  -18.35    1.13   -1.21   11.25
##  [118]  -17.50   10.58   18.69    9.49   -1.27  -12.34  -37.05    0.71   -4.59
##  [127]    6.95  -29.36  -10.97  -20.00   47.28  -11.88    6.71  -27.34  -11.54
##  [136]   54.14  -32.32   31.09   20.37    3.11  -11.86  -15.37  -10.54    7.48
##  [145]   47.48   -2.65    1.78    1.09    2.14   -7.00  -11.05    6.06  -27.72
##  [154]   -4.51    6.11   30.28    0.85   24.77   -2.16  -12.23    3.99    8.89
##  [163]    9.02   -1.47   -5.43   -5.35   23.75    4.56   -6.41   10.77  -25.69
##  [172]    5.11   -9.40   15.30   -0.54    5.62   14.91    1.78    1.28  -13.23
##  [181]  -22.69    3.64  -18.42    9.42    5.49    7.42    2.12  -14.71   -8.02
##  [190]   -0.45   26.85  -43.37    1.08   -3.32  -22.95    4.38   20.16    0.11
##  [199]   -9.21  -13.12    5.02  -24.90    0.07   -3.71    7.68  -38.82   -4.77
##  [208]    1.62    4.91  -23.39    1.38  -10.59   21.39  -29.01    8.70  -13.90
##  [217]  -11.19  -13.39   30.45   14.96    0.36   -0.68   17.00    5.19  -29.65
##  [226]    5.22  -23.39   28.83   21.06  -16.88   -7.07  -11.30   35.87    4.31
##  [235]  -23.12   30.25    9.00  -15.73   -3.76    7.10    5.27  -19.60  -11.91
##  [244]    7.85    3.18   14.48  -25.36    4.67   10.15   19.02  -17.85   -5.25
##  [253]   -2.60  -38.15    5.48   25.48  -43.28    7.53   17.01    2.65  -59.00
##  [262]   20.89  -57.20   50.12   48.57  -47.99  -18.87   -2.35   23.31    3.83
##  [271] -106.62   59.97   -5.30  -46.78  -15.05  -42.34  -60.66  -11.29  -75.02
##  [280]  -10.70  104.13   -5.34  -90.17   38.59   -5.88   44.85  -30.35  -58.27
##  [289]   11.33  -31.34  -27.85   91.59  -10.42   24.00   14.66   -2.45   39.45
##  [298]  -52.98  -47.89   26.11  -11.78  -20.26  -46.65   58.99  -38.00  -22.54
##  [307]    8.37  -52.54  -54.14   47.59   51.78    5.58   30.29    8.56  -80.03
##  [316]   32.60   21.93  -25.52   30.85   33.63  -21.03   10.57  -25.65    6.14
##  [325]  -11.16   44.61   -8.76  -19.14    2.60  -16.25   -8.47    4.99    4.65
##  [334]   -3.38   21.22   12.61   28.55   -4.35    7.25  -28.05    3.08  -19.38
##  [343]  -20.09    1.53  -29.17    1.12    6.38  -44.90   35.02  -12.74    4.45
##  [352]    4.62    9.14   28.38  -28.95  -19.26   -0.44   13.07   -6.28   13.62
##  [361]   22.75    1.29  -42.73    6.58    1.45   -8.35  -37.67   -0.75   -9.48
##  [370]   -8.89  -26.72   29.81   -8.24  -12.07  -17.74  -34.27   -4.49   16.54
##  [379]  -30.32    0.83   -6.85   43.07    1.76   29.38    5.81   -2.66   24.23
##  [388]   16.23  -10.31  -15.50   54.38  -16.80    7.76   18.98  -16.92  -28.41
##  [397]   10.34   13.21   23.30    8.12   -7.02  -19.93    9.61   31.40    2.17
##  [406]  -17.23   10.56   13.24    4.30  -37.21   17.69   -6.53    8.37   14.31
##  [415]   -8.72   -2.35   18.48   -0.83    4.71   29.72   -3.44   15.73  -12.14
##  [424]   21.84  -19.99   -0.89  -24.43    9.15  -10.19   26.83   -1.58   -4.66
##  [433]  -15.14   -1.33   23.33  -17.27   13.77   12.31   23.73    1.87  -12.98
##  [442]   10.70   -2.37   -0.95    3.29   -3.28    5.74    1.32  -22.49  -11.75
##  [451]   -1.26    7.66    2.86  -28.19    2.06    5.84   19.32   -1.36    8.33
##  [460]   -7.91    4.01  -26.91    2.30  -17.69   -1.47    3.12   -3.55   21.92
##  [469]    4.79   26.84    8.06   -0.36   10.75    3.45   -0.51   22.22    2.97
##  [478]    2.92   -2.56   -4.47   11.60    0.73   15.15    3.02   -2.93   -5.64
```

```
##  [487]  13.40  -3.38 -12.75  11.46   6.92  -8.64 -24.36   9.94   6.79
##  [496]  10.91  18.76  -0.56   2.43   0.12   2.86  -2.05  -8.31 -22.58
##  [505]  -3.29   8.49  13.16   8.99   7.98  10.77  -1.41   6.61   3.29
##  [514]  16.13  -3.27   2.81  -3.64   7.00 -10.79 -10.09  -6.40  18.60
##  [523]  -2.37  -3.53 -27.23  -4.64  15.25  14.26   2.86   7.90   6.01
##  [532]   4.70  -3.00  18.83   4.54  -8.88  10.23  -6.85  -9.66  11.51
##  [541] -13.31 -12.65  -3.54 -20.78  23.48 -29.92   6.69   2.53   1.09
##  [550]  20.13   2.67  23.78  -0.07   5.50 -11.27   6.24  15.82   1.02
##  [559]  -0.52 -14.90  -3.52  14.86  -0.59   4.98 -19.14   4.14  13.23
##  [568]   0.38  -9.32   6.06  -2.73 -11.31   4.01   6.40   4.06   7.70
##  [577]  -6.18   1.25 -13.10   6.39  11.58   3.97   2.57   5.89   1.30
##  [586]  -1.58   0.22 -11.32  17.89   3.53   0.62   4.55   3.29   2.00
##  [595] -10.76   9.46   2.78 -12.43  14.20 -12.19 -21.56 -24.72   5.02
##  [604]  -4.61   5.33 -12.97 -10.66  15.32  14.13  -6.04 -34.17   3.08
##  [613]  -9.45  13.78  -2.39  10.34  -2.96  19.36   4.64   7.24   2.42
##  [622]  -1.16 -13.41  10.64  -2.30   1.55  11.22   2.60   0.48   4.18
##  [631]  15.73  -0.20   1.95   5.16   4.63  -0.25   0.52   8.95   6.75
##  [640]  -0.38  -5.93   5.91   8.36  -6.45  -1.99   0.86   6.63   0.05
##  [649]  -3.84   8.67   9.34   2.00  -6.99   3.99   7.93   2.11   0.82
##  [658]  13.35   1.02 -19.54   5.39   9.65  -1.23   2.73   8.61  -5.23
##  [667] -28.34   7.65  15.42 -20.09  15.57 -28.66  -7.70 -37.75 -17.27
##  [676]  48.85  -3.94  15.88 -14.23 -21.76   1.26 -16.14  -5.75 -43.46
##  [685]  16.10 -14.04   0.38  -6.08  35.11 -13.65 -18.70  27.67   4.45
##  [694] -37.95 -14.41  11.53  -6.31  31.15   4.76  -1.97  25.60  -0.62
##  [703]   1.43   1.47  -4.31 -17.89  -3.27 -18.35   3.07  -2.19 -33.33
##  [712] -10.53  -3.34  -4.79   5.48  32.21   9.98   7.71   0.79  16.59
##  [721]  -0.17   1.31 -31.60   6.37  12.23 -13.89  24.08   8.99  12.35
##  [730]  -1.17  -7.71  -4.60   0.07  24.26  -5.40   6.78  -1.43  -4.17
##  [739]   6.15  -6.73 -31.59  -5.86  -4.36   0.13  13.16   1.62 -18.53
##  [748]  -3.94  -4.33 -15.49   3.46  -8.11  17.37 -15.67   0.41  30.96
##  [757]   9.81  14.41 -12.67   7.03   5.31   5.37  12.35  -0.80   3.97
##  [766]  -0.41   0.93  17.12  -2.93  -5.50  -9.45  23.84  -6.51   5.54
##  [775]  -2.97  -3.53   5.04  -9.21  23.72  -0.78  -1.91   7.09   0.17
##  [784]   4.45   8.33  -4.29   2.38   8.52 -18.81  12.27   2.09   2.82
##  [793]   2.54   0.02  -3.19   1.33  -0.52   1.12   9.19   4.39  23.10
##  [802]   4.79  -2.60  -9.85   5.31  -5.17 -14.33  -1.46 -19.41   0.25
##  [811]  18.10   3.04  -1.89 -17.11  17.62  -8.95  -1.64  -7.21  25.52
##  [820]  15.46   3.18  -1.59   0.63   4.53   4.72   7.40   0.06   1.13
##  [829]  -6.36   7.64   1.04   3.17   7.52   4.24  -2.07   0.77   0.97
##  [838]   1.27  -1.90  -0.24  14.23  -1.67   6.36  -2.71  -2.35  -1.75
##  [847]   4.73  11.48  -2.20   9.48   1.78 -13.10  -1.66   3.09   7.49
##  [856]   0.34   5.45   2.91 -23.20   9.78  21.47  -3.56   3.07   3.77
##  [865]   8.18   5.52  -3.69   0.99   7.28   3.17  -4.31   8.31   4.11
##  [874]   2.58 -27.57  -8.04  -1.30  13.78   7.34 -20.89   2.11  22.53
##  [883]  -9.82 -11.02  11.69  -1.80 -24.91   9.17  -7.89 -14.52 -24.99
##  [892]  16.84   5.48  19.18  -4.61   3.77  12.12   4.14  -3.61   9.25
##  [901]   8.82  -2.43   6.58   0.46  -0.24   2.91  -2.03  -5.34  -3.71
##  [910] -10.30   0.25   0.11   5.16 -14.54   7.48  17.74   7.02  -2.13
##  [919]  11.99   8.42   4.82   3.13  -2.39  -4.60  -9.30 -12.22   5.10
##  [928]   6.09  10.87 -15.08   6.57 -10.88  -8.30  -0.49  11.70   2.92
##  [937] -10.33 -15.90  -1.09   4.19   5.22   5.41  14.10 -30.65  -1.61
##  [946] -12.78 -13.99  -1.23  -5.38   9.44 -18.02   0.85  16.04 -22.45
##  [955]   2.22   3.86   6.86  17.16  -8.38  -3.64 -15.05  11.65  16.57
##  [964]  10.74  13.23  19.03  -1.79   1.34  14.00  -9.42 -24.31  -5.85
```

```
##  [973]    4.08   -8.85    7.27  -10.70   21.29   -0.89   17.96    1.22   -7.59
##  [982]   -5.49  -27.05   -4.22   -8.39   -5.34  -32.89    6.29  -60.27   -0.69
##  [991]  -79.92   53.07  -51.77   51.88    6.17   25.68  -11.73    1.13  -53.24
## [1000]  -17.12    0.29   38.53   15.25  -18.33   17.53   33.28    2.84    5.97
## [1009]  -14.47  -30.45   -8.73   33.38  -12.72  -31.67    8.04   10.60   15.81
## [1018]   20.43    6.90  -11.92   -2.00  -35.33  -37.20    6.87   26.52   12.43
## [1027]  -24.32    9.34  -28.98  -32.19   24.72   20.08   20.94   -9.51   39.43
## [1036]    0.65   11.71   -3.59   20.92  -23.72   24.52  -15.50    5.51   22.86
## [1045]   15.94  -25.14   12.95   42.59    0.50  -31.79  -35.02   19.62   23.25
## [1054]   -7.92    7.89   14.80  -46.82   10.59   24.16  -12.07    6.03  -20.90
## [1063]  -20.78   -0.48  -22.67   -4.94  -26.25   -3.12   33.88    2.64   51.77
## [1072]   -2.38   -0.30   12.80    1.39    2.54  -26.66   20.84  -18.72  -10.74
## [1081]  -13.91    3.93    3.91  -14.31   35.95    2.42   10.28   11.33    0.10
## [1090]  -15.79   13.38   -5.42   19.46    0.24    3.76   -3.25    2.89   11.38
## [1099]    0.40    3.02   -6.41    4.58   14.37    6.46    0.88    0.62   -1.35
## [1108]   11.40   -7.62   -2.10   -3.32   -0.60   11.68    1.45   19.36   -0.57
## [1117]    2.72    2.91    1.99   -9.31    9.13   -1.27   -7.27   14.81    3.19
## [1126]    0.98   -4.55    5.80    2.28    1.85    4.59   -6.50    8.41   -4.46
## [1135]   -5.30  -20.97    9.27   13.28    4.96    0.22   24.86   -1.67    8.32
## [1144]    1.57    5.58   -4.23   -2.63  -10.11    4.33   19.40   -3.99   -6.98
## [1153]   -2.26    5.19   10.57   -5.66  -14.42   -0.88  -15.88  -23.61   10.12
## [1162]   18.86  -17.31   -0.69   21.21   -5.64   -8.22    1.61  -11.59    5.03
## [1171]   18.72    9.29    3.38   -5.45    7.91   -3.51  -10.74  -22.47    0.48
## [1180]   -5.86   -9.14    3.41   -4.60  -15.04   -7.69   -5.86  -19.94   -9.64
## [1189]   20.77    0.64    2.23    1.82   -2.86   14.60  -19.10   -2.99  -32.29
## [1198]    0.14    7.32   29.63   -0.14   10.67  -16.73   15.25   -9.30   14.22
## [1207]   13.74    1.94   13.20   -2.29  -30.18    9.51  -21.30    6.27   11.86
## [1216]   -2.81   33.12    3.35    8.51   -6.44  -12.90   -2.22  -10.99   -0.02
## [1225]   -6.69   22.02   -3.14   10.03    9.11    3.73  -13.85  -12.14  -12.21
## [1234]   -0.42   22.13   25.95   -0.67   -5.98   -4.18  -10.14   25.99    3.24
## [1243]    7.12    0.87    0.58    3.07   -1.76   -0.18    1.60    9.98    2.65
## [1252]   -0.03   -4.96    0.32  -11.41    9.05   -0.69   -1.14    1.19  -11.01
## [1261]    7.10   -1.64   -1.50   28.68    5.80   -8.84    4.48    3.00   23.43
## [1270]    5.78   -4.58   -1.87    1.73   -0.79   -0.11   -3.26  -15.30   -8.27
## [1279]   13.83   -6.48    3.82    1.26    5.24   10.41   -0.47   -5.05  -14.40
## [1288]   -8.92    0.28   -4.25   11.54   14.79    5.99   -3.57  -24.15    0.63
## [1297]  -20.71   -4.36    4.22   -1.03    0.22   15.43  -13.39    3.06   11.13
## [1306]  -33.86  -17.02    2.34    0.18   -5.50  -19.04   -2.16    6.55   27.01
## [1315]    0.92    3.22   18.12   -2.86   -7.35   10.99    6.02    0.23   -6.72
## [1324]   -2.41    2.23    4.66    4.13    0.48    9.29    0.64   -9.03   -5.87
## [1333]   16.78   16.43  -10.98    7.88  -13.54   -3.49   -6.83   -1.73  -15.67
## [1342]   23.76
```

```
set.seed(100)
names(my_df)
```

```
##   [1] "date"         "A"            "AA"           "AAPL"         "ABC"
##   [6] "ABT"          "ACE"          "ACN"          "ADBE"         "ADI"
##  [11] "ADM"          "ADP"          "ADSK"         "AEE"          "AEP"
##  [16] "AES"          "AET"          "AFL"          "AGN"          "AIG"
##  [21] "AIV"          "AIZ"          "AKAM"         "ALL"          "ALTR"
##  [26] "ALXN"         "AMAT"         "AMD"          "AMGN"         "AMP"
##  [31] "AMT"          "AMZN"         "AN"           "ANF"          "APA"
##  [36] "APC"          "APD"          "APH"          "APOL"         "ARG"
```

```
##  [41] "ATI"          "AVB"          "AVP"          "AVY"          "AXP"
##  [46] "AZO"          "BA"           "BAC"          "BAX"          "BBBY"
##  [51] "BBT"          "BBY"          "BCR"          "BDX"          "BEN"
##  [56] "BHI"          "BIIB"         "BK"           "BLK"          "BLL"
##  [61] "BMC"          "BMS"          "BMY"          "BRCM"         "BSX"
##  [66] "BTU"          "BWA"          "BXP"          "C"            "CA"
##  [71] "CAG"          "CAH"          "CAM"          "CAT"          "CB"
##  [76] "CBG"          "CCE"          "CCI"          "CCL"          "CELG"
##  [81] "CERN"         "CF"           "CHK"          "CHRW"         "CI"
##  [86] "CINF"         "CL"           "CLF"          "CLX"          "CMA"
##  [91] "CMCSA"        "CME"          "CMI"          "CMS"          "CMT"
##  [96] "CNP"          "CNX"          "COF"          "COG"          "COH"
## [101] "COHU"         "COL"          "COP"          "COST"         "COV"
## [106] "CPB"          "CRM"          "CSC"          "CSCO"         "CSX"
## [111] "CTAS"         "CTL"          "CTSH"         "CTXS"         "CVC"
## [116] "CVS"          "CVX"          "D"            "DD"           "DE"
## [121] "DELL"         "DFS"          "DGX"          "DHI"          "DHR"
## [126] "DIS"          "DISCA"        "DLTR"         "DNB"          "DNR"
## [131] "DO"           "DOV"          "DOW"          "DRI"          "DTE"
## [136] "DTV"          "DUK"          "DVA"          "DVN"          "EBAY"
## [141] "EBIX"         "ECL"          "ED"           "EFX"          "EIX"
## [146] "EL"           "EMC"          "EMN"          "EMR"          "EOG"
## [151] "EQR"          "EQT"          "ESRX"         "ETFC"         "ETN"
## [156] "ETR"          "EW"           "EXC"          "EXPD"         "EXPE"
## [161] "F"            "FAST"         "FCX"          "FDO"          "FDX"
## [166] "FE"           "FFIV"         "FIS"          "FISV"         "FITB"
## [171] "FLIR"         "FLR"          "FLS"          "FMC"          "FOSL"
## [176] "FRX"          "FSLR"         "FTI"          "GCI"          "GD"
## [181] "GE"           "GILD"         "GIS"          "GLW"          "GME"
## [186] "GNW"          "GOOG"         "GPC"          "GPS"          "GRMN"
## [191] "GS"           "GT"           "GWW"          "HAL"          "HAR"
## [196] "HAS"          "HBAN"         "HCBK"         "HCN"          "HCP"
## [201] "HD"           "HES"          "HIG"          "HOG"          "HON"
## [206] "HOT"          "HP"           "HPQ"          "HRB"          "HRL"
## [211] "HRS"          "HSP"          "HST"          "HSY"          "HUM"
## [216] "IBM"          "ICE"          "IFF"          "IGT"          "INTC"
## [221] "INTU"         "IP"           "IPG"          "IR"           "IRM"
## [226] "ISRG"         "ITW"          "IVZ"          "JBL"          "JCI"
## [231] "JCP"          "JDSU"         "JEC"          "JNJ"          "JNPR"
## [236] "JPM"          "JWN"          "K"            "KEY"          "KIM"
## [241] "KLAC"         "KMB"          "KMX"          "KO"           "KR"
## [246] "KSS"          "KSU"          "LEG"          "LH"           "LLL"
## [251] "LLTC"         "LLY"          "LM"           "LMT"          "LNC"
## [256] "LOW"          "LRCX"         "LSI"          "LTD"          "LUK"
## [261] "LUV"          "M"            "MA"           "MAC"          "MAR"
## [266] "MAS"          "MAT"          "MCD"          "MCHP"         "MCK"
## [271] "MCO"          "MDT"          "MET"          "MMC"          "MMM"
## [276] "MO"           "MOLX"         "MON"          "MOS"          "MPET"
## [281] "MRK"          "MRO"          "MS"           "MSFT"         "MTB"
## [286] "MU"           "MUR"          "MWV"          "MYL"          "NBL"
## [291] "NBR"          "NDAQ"         "NE"           "NEM"          "NFLX"
## [296] "NFX"          "NI"           "NKE"          "NOC"          "NOV"
## [301] "NRG"          "NSC"          "NTAP"         "NTRS"         "NU"
## [306] "NUE"          "NVDA"         "NWL"          "NYX"          "OI"
```

```
## [311] "OKE"          "OMC"          "ORCL"         "ORLY"         "OXY"
## [316] "PAYX"         "PBCT"         "PBI"          "PCAR"         "PCG"
## [321] "PCL"          "PCLN"         "PCP"          "PDCO"         "PEG"
## [326] "PEP"          "PETM"         "PFE"          "PFG"          "PG"
## [331] "PGR"          "PH"           "PHM"          "PIR"          "PKI"
## [336] "PLL"          "PNC"          "PNW"          "POM"          "PPG"
## [341] "PPL"          "PRGO"         "PRU"          "PSA"          "PVH"
## [346] "PWR"          "PX"           "PXD"          "QCOM"         "R"
## [351] "RAI"          "RDC"          "REGN"         "RF"           "RHI"
## [356] "RHT"          "RL"           "ROK"          "ROP"          "ROST"
## [361] "RRC"          "RSG"          "RTN"          "S"            "SAI"
## [366] "SBUX"         "SCG"          "SCHW"         "SE"           "SEE"
## [371] "SHW"          "SIAL"         "SJM"          "SLB"          "SLM"
## [376] "SNA"          "SNDK"         "SO"           "SPG"          "SPLS"
## [381] "SRCL"         "SRE"          "STI"          "STJ"          "STT"
## [386] "STX"          "SWK"          "SWN"          "SWY"          "SYK"
## [391] "SYMC"         "SYY"          "T"            "TE"           "TEG"
## [396] "TEL"          "TER"          "TGT"          "THC"          "TIF"
## [401] "TJX"          "TMK"          "TMO"          "TROW"         "TRV"
## [406] "TSN"          "TSO"          "TSS"          "TWC"          "TWX"
## [411] "TXN"          "TXT"          "TYC"          "UNH"          "UNM"
## [416] "UNP"          "UPS"          "URBN"         "USB"          "UTX"
## [421] "VAR"          "VFC"          "VLO"          "VMC"          "VNO"
## [426] "VRSN"         "VTR"          "VZ"           "WAG"          "WAT"
## [431] "WDC"          "WEC"          "WFC"          "WHR"          "WIN"
## [436] "WLP"          "WMB"          "WMT"          "WPO"          "WU"
## [441] "WY"           "WYN"          "WYNN"         "X"            "XEL"
## [446] "XL"           "XLNX"         "XOM"          "XRAY"         "XRX"
## [451] "YHOO"         "YUM"          "ZION"         "ZMH"          "SP.500.Level"
## [456] "Returns"
```

```
#pre-processing the data
x2_data = as.matrix(my_df[,-c(1,455, 456)])

#finding the best value of lambda for LASSO regression
lambdas = 10^seq(2,-3, by=-0.1)

l1_m_temp2 = cv.glmnet(x=x2_data, y=my_df$Returns, alpha=1, lambda = lambdas)

optimal_lambda_lasso2 = l1_m_temp2$lambda.min
optimal_lambda_lasso2
```

```
## [1] 0.01584893
```

```
#building the model using the optimal value of lambda
l1_model2 = glmnet(x2_data, my_df$Returns, alpha=1, lambda = optimal_lambda_lasso2)

round(l1_model2$beta,2)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##           s0
## A      -0.09
## AA     -0.76
```

```
## AAPL   -0.02
## ABC     .
## ABT     0.67
## ACE    -0.16
## ACN     .
## ADBE   -0.61
## ADI     0.45
## ADM     .
## ADP    -2.14
## ADSK    .
## AEE    -1.69
## AEP     0.00
## AES     0.68
## AET     .
## AFL     .
## AGN     .
## AIG     .
## AIV    -1.01
## AIZ     1.01
## AKAM    .
## ALL    -0.44
## ALTR    .
## ALXN    0.38
## AMAT    .
## AMD     .
## AMGN    0.54
## AMP     0.11
## AMT     0.07
## AMZN   -0.14
## AN      .
## ANF    -0.15
## APA     0.07
## APC     .
## APD    -0.22
## APH     0.23
## APOL   -0.21
## ARG     0.48
## ATI     .
## AVB     0.23
## AVP    -1.08
## AVY    -0.90
## AXP     .
## AZO     0.12
## BA     -0.24
## BAC     .
## BAX    -0.56
## BBBY    .
## BBT     0.63
## BBY    -0.38
## BCR     0.36
## BDX     0.03
## BEN     0.38
## BHI    -0.01
## BIIB    .
```

```
## BK      0.59
## BLK     .
## BLL    -0.57
## BMC     0.35
## BMS     1.46
## BMY     2.80
## BRCM    0.25
## BSX     .
## BTU     0.25
## BWA     0.00
## BXP     .
## C       .
## CA      .
## CAG    -2.97
## CAH     0.07
## CAM     .
## CAT     .
## CB      0.34
## CBG     0.99
## CCE     .
## CCI     .
## CCL     0.74
## CELG   -0.30
## CERN   -0.07
## CF      0.16
## CHK    -0.73
## CHRW    0.60
## CI      0.03
## CINF    1.41
## CL      .
## CLF     .
## CLX     1.18
## CMA    -0.15
## CMCSA   .
## CME     0.00
## CMI     .
## CMS    -0.93
## CMT     0.21
## CNP    -1.17
## CNX     0.17
## COF    -0.30
## COG     0.51
## COH     .
## COHU    3.25
## COL    -0.06
## COP     .
## COST    .
## COV    -0.99
## CPB     0.06
## CRM     .
## CSC    -0.12
## CSCO    0.94
## CSX    -0.26
## CTAS    .
```

```
## CTL    -0.59
## CTSH    .
## CTXS    0.27
## CVC    -0.18
## CVS     .
## CVX    -1.17
## D       .
## DD     -2.21
## DE      .
## DELL    0.24
## DFS     .
## DGX    -1.32
## DHI    -1.23
## DHR     0.16
## DIS    -1.11
## DISCA   .
## DLTR    .
## DNB    -0.33
## DNR    -0.32
## DO     -0.65
## DOV    -0.64
## DOW     .
## DRI     .
## DTE     .
## DTV     .
## DUK    -0.61
## DVA    -0.45
## DVN     .
## EBAY    0.07
## EBIX   -0.12
## ECL     1.68
## ED      0.10
## EFX     .
## EIX     .
## EL      .
## EMC     .
## EMN     .
## EMR     1.09
## EOG     0.63
## EQR     1.23
## EQT    -0.48
## ESRX    .
## ETFC    .
## ETN     0.03
## ETR     0.94
## EW      .
## EXC     .
## EXPD    .
## EXPE    0.14
## F      -0.79
## FAST   -0.18
## FCX     .
## FDO     .
## FDX     0.46
```

```
## FE      .
## FFIV   -0.15
## FIS     0.15
## FISV   -0.60
## FITB    3.84
## FLIR    0.29
## FLR     0.02
## FLS    -0.16
## FMC     0.29
## FOSL    .
## FRX     1.58
## FSLR   -0.09
## FTI     0.38
## GCI     .
## GD      .
## GE     -0.63
## GILD    0.28
## GIS     .
## GLW     .
## GME     .
## GNW     .
## GOOG   -0.01
## GPC     .
## GPS     .
## GRMN    .
## GS     -0.32
## GT     -1.74
## GWW    -0.06
## HAL     .
## HAR    -0.28
## HAS    -0.29
## HBAN   -2.79
## HCBK   -1.27
## HCN     .
## HCP     0.41
## HD     -1.04
## HES     0.05
## HIG    -0.16
## HOG    -1.57
## HON    -0.16
## HOT     .
## HP      0.00
## HPQ     .
## HRB     0.39
## HRL     0.66
## HRS     0.06
## HSP     .
## HST     .
## HSY     .
## HUM     .
## IBM     .
## ICE     0.04
## IFF     .
## IGT     .
```

```
## INTC    0.07
## INTU    .
## IP      .
## IPG     3.26
## IR     -0.08
## IRM     .
## ISRG   -0.12
## ITW     .
## IVZ    -0.57
## JBL    -0.26
## JCI     .
## JCP     .
## JDSU   -0.88
## JEC    -0.22
## JNJ     0.05
## JNPR    0.30
## JPM     1.22
## JWN     .
## K       1.43
## KEY    -1.35
## KIM     .
## KLAC    .
## KMB     .
## KMX    -0.38
## KO      0.24
## KR      0.38
## KSS     .
## KSU     .
## LEG    -0.49
## LH      0.22
## LLL     .
## LLTC   -0.62
## LLY    -0.89
## LM      0.24
## LMT     0.20
## LNC     .
## LOW    -0.23
## LRCX   -0.13
## LSI     4.62
## LTD     .
## LUK     .
## LUV     0.92
## M       .
## MA      0.10
## MAC    -0.22
## MAR     0.14
## MAS     .
## MAT     .
## MCD     .
## MCHP    0.32
## MCK     0.31
## MCO    -0.22
## MDT    -0.45
## MET    -0.07
```

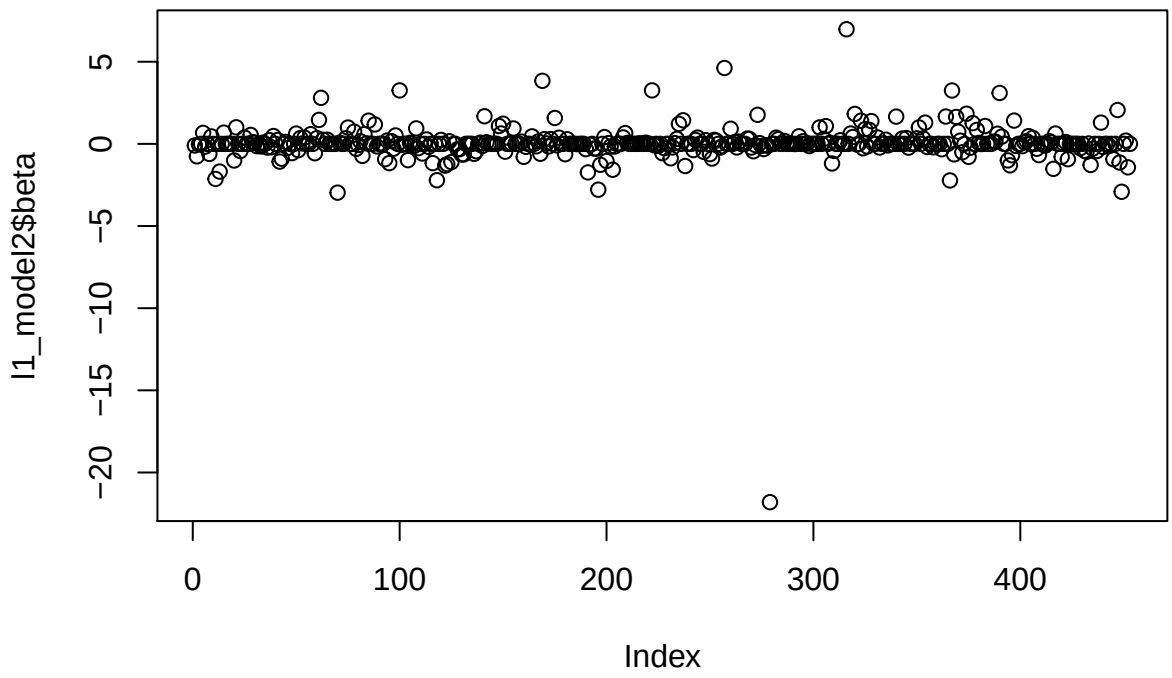
```
## MMC     1.77
## MMM     0.05
## MO     -0.12
## MOLX   -0.31
## MON    -0.08
## MOS    -0.11
## MPET  -21.81
## MRK     .
## MRO     .
## MS      0.38
## MSFT    0.28
## MTB     .
## MU      .
## MUR     0.18
## MWV     .
## MYL     .
## NBL     .
## NBR     .
## NDAQ   -0.01
## NE      .
## NEM     0.46
## NFLX    .
## NFX     0.16
## NI      .
## NKE     0.04
## NOC    -0.13
## NOV     .
## NRG     .
## NSC     .
## NTAP    .
## NTRS    1.00
## NU      .
## NUE     0.30
## NVDA    1.08
## NWL     .
## NYX     0.07
## OI     -1.19
## OKE    -0.43
## OMC     .
## ORCL    .
## ORLY    0.22
## OXY     .
## PAYX    .
## PBCT    6.98
## PBI     .
## PCAR    0.63
## PCG     0.42
## PCL     1.82
## PCLN   -0.04
## PCP     0.00
## PDCO    1.41
## PEG    -0.27
## PEP     0.90
## PETM   -0.16
```

```
## PFE     0.85
## PFG     1.39
## PG      0.01
## PGR     .
## PH      0.39
## PHM    -0.22
## PIR     .
## PKI     0.03
## PLL     0.25
## PNC    -0.09
## PNW     .
## POM     .
## PPG     .
## PPL     1.66
## PRGO    .
## PRU     .
## PSA     0.33
## PVH     .
## PWR     0.35
## PX     -0.23
## PXD     .
## QCOM    0.01
## R       .
## RAI     0.34
## RDC     1.00
## REGN    0.00
## RF      0.33
## RHI     1.29
## RHT    -0.19
## RL      .
## ROK     .
## ROP    -0.21
## ROST   -0.02
## RRC     .
## RSG     .
## RTN    -0.33
## S       .
## SAI     1.66
## SBUX    .
## SCG    -2.23
## SCHW    3.25
## SE     -0.64
## SEE     1.64
## SHW     0.74
## SIAL    0.19
## SJM    -0.51
## SLB     .
## SLM     1.83
## SNA    -0.79
## SNDK   -0.23
## SO      1.26
## SPG     .
## SPLS    0.85
## SRCL    0.06
```

```
## SRE    -0.01
## STI     .
## STJ     1.09
## STT     .
## STX     .
## SWK     .
## SWN     0.13
## SWY     0.13
## SYK     0.62
## SYMC    3.10
## SYY     0.44
## T       .
## TE      .
## TEG    -1.01
## TEL    -1.30
## TER    -0.69
## TGT     1.42
## THC    -0.13
## TIF     .
## TJX     .
## TMK    -0.14
## TMO     0.10
## TROW    0.17
## TRV     0.47
## TSN     .
## TSO     0.36
## TSS     .
## TWC    -0.30
## TWX    -0.68
## TXN     .
## TXT     .
## TYC    -0.11
## UNH     .
## UNM     0.17
## UNP     0.08
## UPS    -1.52
## URBN    0.63
## USB     .
## UTX     .
## VAR    -0.79
## VFC     .
## VLO     0.11
## VMC    -0.93
## VNO     .
## VRSN    .
## VTR     .
## VZ      .
## WAG     .
## WAT    -0.26
## WDC     .
## WEC    -0.42
## WFC    -0.48
## WHR     0.02
## WIN    -1.28
```

```
## WLP    -0.29
## WMB     .
## WMT    -0.43
## WPO     .
## WU      1.30
## WY     -0.19
## WYN     .
## WYNN    .
## X      -0.13
## XEL     .
## XL     -0.93
## XLNX    .
## XOM     2.06
## XRAY   -1.13
## XRX    -2.92
## YHOO    .
## YUM     0.20
## ZION   -1.43
## ZMH     .
```

```
plot(l1_model2$beta)
```



Comparing this plot to the plot created by the coefficients of the model with target variable SP500 index, here, we can see a straight black line at 0, indicating that a large number of values are 0 and the model is more sparse.

```r
#creating the dataset for first 60 days
f60_df2 = as.data.frame(my_df[1:60,])
xdata2_f60 = as.matrix(f60_df2[,-c(1,455,456)])

l1_f60_m2 = cv.glmnet(xdata2_f60, y=f60_df2$Returns, alpha = 1, lambda = lambdas)
optimal_lambda2_f60 = l1_f60_m2$lambda.min
optimal_lambda2_f60
```

```
## [1] 0.6309573
```

```r
#building the model using the optimal value of lambda
l12_f60 = glmnet(xdata2_f60, f60_df2$Returns, alpha=1, lambda = optimal_lambda2_f60)
round(l12_f60$beta,3)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##           s0
## A       .
## AA      .
## AAPL    .
## ABC     .
## ABT     .
## ACE     .
## ACN     .
## ADBE    .
## ADI     .
## ADM     .
## ADP     .
## ADSK    .
## AEE     .
## AEP     .
## AES     .
## AET     .
## AFL     1.415
## AGN     .
## AIG     .
## AIV     .
## AIZ     .
## AKAM    .
## ALL     .
## ALTR    .
## ALXN    .
## AMAT    .
## AMD     .
## AMGN    .
## AMP     .
## AMT     .
## AMZN    .
## AN      .
## ANF     .
## APA     .
## APC     .
## APD     .
## APH     .
```

```
## APOL    .
## ARG     .
## ATI     .
## AVB     .
## AVP     .
## AVY     .
## AXP     .
## AZO     .
## BA      .
## BAC     .
## BAX     .
## BBBY    .
## BBT     .
## BBY     .
## BCR     1.265
## BDX     .
## BEN     .
## BHI     .
## BIIB    .
## BK      .
## BLK     .
## BLL     .
## BMC     .
## BMS     .
## BMY     .
## BRCM    .
## BSX     .
## BTU     .
## BWA     .
## BXP     .
## C       .
## CA      .
## CAG     .
## CAH     .
## CAM     .
## CAT     .
## CB      .
## CBG     .
## CCE     .
## CCI     .
## CCL     .
## CELG    .
## CERN    .
## CF      .
## CHK     .
## CHRW    .
## CI      .
## CINF    .
## CL      .
## CLF     .
## CLX     .
## CMA     .
## CMCSA   .
## CME     .
```

```
## CMI    .
## CMS    .
## CMT    .
## CNP    .
## CNX    .
## COF    .
## COG    .
## COH    .
## COHU   .
## COL    .
## COP    .
## COST   .
## COV    .
## CPB    1.184
## CRM    .
## CSC    .
## CSCO   .
## CSX    .
## CTAS   .
## CTL    0.273
## CTSH   .
## CTXS   .
## CVC    .
## CVS    .
## CVX    .
## D     -0.280
## DD     .
## DE     .
## DELL   .
## DFS    .
## DGX    .
## DHI    .
## DHR    .
## DIS    .
## DISCA  .
## DLTR   .
## DNB    .
## DNR    .
## DO     .
## DOV    .
## DOW    .
## DRI    .
## DTE    .
## DTV    .
## DUK   16.354
## DVA   -0.320
## DVN    .
## EBAY   .
## EBIX   .
## ECL    .
## ED     .
## EFX    .
## EIX    .
## EL     .
```

```
## EMC    .
## EMN    .
## EMR    .
## EOG    .
## EQR    .
## EQT    .
## ESRX   .
## ETFC   .
## ETN    .
## ETR    .
## EW     .
## EXC    .
## EXPD   .
## EXPE   .
## F      .
## FAST   .
## FCX    .
## FDO    .
## FDX    .
## FE     .
## FFIV   .
## FIS    .
## FISV   .
## FITB   .
## FLIR   .
## FLR    .
## FLS    .
## FMC    .
## FOSL   .
## FRX    .
## FSLR   .
## FTI    .
## GCI    .
## GD     .
## GE     .
## GILD   .
## GIS    .
## GLW    .
## GME    .
## GNW    .
## GOOG   .
## GPC    1.411
## GPS    .
## GRMN   .
## GS     .
## GT     .
## GWW    .
## HAL    .
## HAR    .
## HAS    .
## HBAN   .
## HCBK   .
## HCN    .
## HCP    .
```

```
## HD     .
## HES    .
## HIG    .
## HOG   -0.279
## HON    .
## HOT    .
## HP     .
## HPQ    1.627
## HRB    .
## HRL    .
## HRS    .
## HSP    .
## HST    .
## HSY    .
## HUM    .
## IBM    .
## ICE    .
## IFF    .
## IGT    .
## INTC   6.374
## INTU   .
## IP     .
## IPG    .
## IR     .
## IRM    .
## ISRG   .
## ITW    .
## IVZ    .
## JBL    .
## JCI    .
## JCP    .
## JDSU   .
## JEC    .
## JNJ    .
## JNPR   .
## JPM    .
## JWN    .
## K      4.140
## KEY    .
## KIM    .
## KLAC   .
## KMB    8.502
## KMX    3.962
## KO     .
## KR     .
## KSS    .
## KSU    .
## LEG    .
## LH     .
## LLL    .
## LLTC   .
## LLY    .
## LM     .
## LMT    .
```

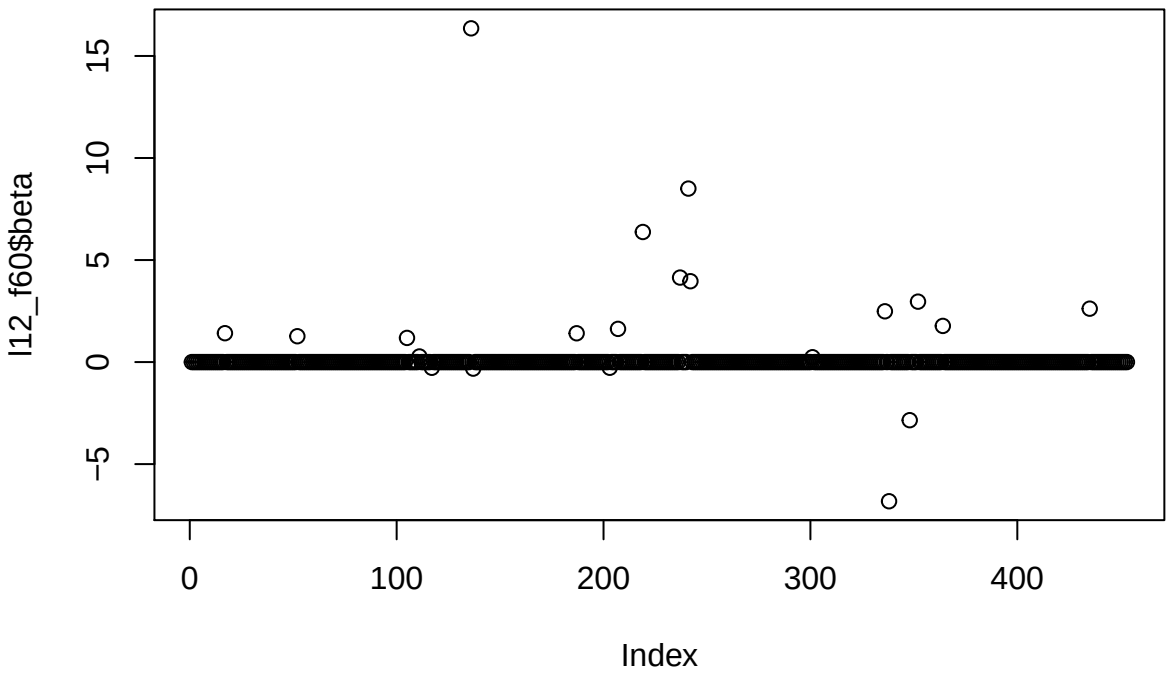
```
## LNC    .
## LOW    .
## LRCX   .
## LSI    .
## LTD    .
## LUK    .
## LUV    .
## M      .
## MA     .
## MAC    .
## MAR    .
## MAS    .
## MAT    .
## MCD    .
## MCHP   .
## MCK    .
## MCO    .
## MDT    .
## MET    .
## MMC    .
## MMM    .
## MO     .
## MOLX   .
## MON    .
## MOS    .
## MPET   .
## MRK    .
## MRO    .
## MS     .
## MSFT   .
## MTB    .
## MU     .
## MUR    .
## MWV    .
## MYL    .
## NBL    .
## NBR    .
## NDAQ   .
## NE     .
## NEM    .
## NFLX   .
## NFX    .
## NI     .
## NKE    .
## NOC    .
## NOV    .
## NRG    .
## NSC    0.238
## NTAP   .
## NTRS   .
## NU     .
## NUE    .
## NVDA   .
## NWL    .
```

```
## NYX    .
## OI     .
## OKE    .
## OMC    .
## ORCL   .
## ORLY   .
## OXY    .
## PAYX   .
## PBCT   .
## PBI    .
## PCAR   .
## PCG    .
## PCL    .
## PCLN   .
## PCP    .
## PDCO   .
## PEG    .
## PEP    .
## PETM   .
## PFE    .
## PFG    .
## PG     .
## PGR    .
## PH     .
## PHM    .
## PIR    .
## PKI    .
## PLL    .
## PNC    2.488
## PNW    .
## POM   -6.820
## PPG    .
## PPL    .
## PRGO   .
## PRU    .
## PSA    .
## PVH    .
## PWR    .
## PX     .
## PXD    .
## QCOM  -2.848
## R      .
## RAI    .
## RDC    .
## REGN   2.961
## RF     .
## RHI    .
## RHT    .
## RL     .
## ROK    .
## ROP    .
## ROST   .
## RRC    .
## RSG    .
```

```
## RTN    .
## S      .
## SAI    1.771
## SBUX   .
## SCG    .
## SCHW   .
## SE     .
## SEE    .
## SHW    .
## SIAL   .
## SJM    .
## SLB    .
## SLM    .
## SNA    .
## SNDK   .
## SO     .
## SPG    .
## SPLS   .
## SRCL   .
## SRE    .
## STI    .
## STJ    .
## STT    .
## STX    .
## SWK    .
## SWN    .
## SWY    .
## SYK    .
## SYMC   .
## SYY    .
## T      .
## TE     .
## TEG    .
## TEL    .
## TER    .
## TGT    .
## THC    .
## TIF    .
## TJX    .
## TMK    .
## TMO    .
## TROW   .
## TRV    .
## TSN    .
## TSO    .
## TSS    .
## TWC    .
## TWX    .
## TXN    .
## TXT    .
## TYC    .
## UNH    .
## UNM    .
## UNP    .
```

```
## UPS    .
## URBN   .
## USB    .
## UTX    .
## VAR    .
## VFC    .
## VLO    .
## VMC    .
## VNO    .
## VRSN   .
## VTR    .
## VZ     .
## WAG    .
## WAT    .
## WDC    .
## WEC    .
## WFC    .
## WHR    .
## WIN    .
## WLP    2.617
## WMB    .
## WMT    .
## WPO    .
## WU     .
## WY     .
## WYN    .
## WYNN   .
## X      .
## XEL    .
## XL     .
## XLNX   .
## XOM    .
## XRAY   .
## XRX    .
## YHOO   .
## YUM    .
## ZION   .
## ZMH    .
```

```
plot(l12_f60$beta)
```




```
#Building a sparse linear model for the last 60 days
l60_df2 = as.data.frame(my_df[1283:1342,])
dim(l60_df2)
```

```
## [1]  60 456
```

```
xdata2_l60 = as.matrix(l60_df2[,-c(1,455,456)])

l1_l60_m2 = cv.glmnet(xdata2_l60, y=l60_df2$Returns, alpha = 1, lambda = lambdas)
optimal_lambda2_l60 = l1_l60_m2$lambda.min
optimal_lambda2_l60
```

```
## [1] 0.01
```

```
#building the model using the optimal value of lambda
l12_l60 = glmnet(xdata2_l60, l60_df2$Returns, alpha=1, lambda = optimal_lambda2_l60)

round(l12_l60$beta, 3)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##            s0
## A       .
## AA      .
## AAPL    .
```

```
## ABC     .
## ABT     .
## ACE     .
## ACN     .
## ADBE    .
## ADI     .
## ADM    -0.441
## ADP     .
## ADSK    .
## AEE     .
## AEP     .
## AES     .
## AET     0.216
## AFL     .
## AGN     .
## AIG     0.364
## AIV     .
## AIZ     .
## AKAM    .
## ALL     .
## ALTR    .
## ALXN    .
## AMAT    .
## AMD    -0.905
## AMGN   -0.232
## AMP     .
## AMT     0.409
## AMZN    .
## AN      .
## ANF     .
## APA     .
## APC     .
## APD     .
## APH     .
## APOL    .
## ARG     0.077
## ATI     .
## AVB     .
## AVP     .
## AVY     .
## AXP     .
## AZO     .
## BA      .
## BAC     .
## BAX     .
## BBBY   -0.104
## BBT     .
## BBY     .
## BCR     .
## BDX     .
## BEN     .
## BHI     .
## BIIB   -0.049
## BK      .
```

```
## BLK     .
## BLL     .
## BMC     .
## BMS     1.334
## BMY     .
## BRCM    .
## BSX     .
## BTU     1.466
## BWA     .
## BXP     .
## C       .
## CA      .
## CAG     .
## CAH     .
## CAM     .
## CAT     .
## CB      .
## CBG     .
## CCE     .
## CCI     .
## CCL     .
## CELG    .
## CERN    .
## CF      0.046
## CHK     .
## CHRW   -1.124
## CI      .
## CINF    .
## CL      .
## CLF     .
## CLX    -0.714
## CMA     .
## CMCSA   .
## CME     .
## CMI     .
## CMS     .
## CMT   -17.314
## CNP     .
## CNX     .
## COF     .
## COG     .
## COH    -0.122
## COHU    .
## COL     .
## COP     .
## COST    1.127
## COV     1.253
## CPB     .
## CRM     .
## CSC     .
## CSCO    .
## CSX     .
## CTAS    0.495
## CTL     3.879
```

```
## CTSH    .
## CTXS    .
## CVC     .
## CVS     .
## CVX     .
## D       .
## DD      .
## DE      1.393
## DELL    .
## DFS     .
## DGX     .
## DHI     .
## DHR     0.057
## DIS    -1.741
## DISCA   0.462
## DLTR    .
## DNB    -1.323
## DNR     .
## DO     -0.167
## DOV     .
## DOW     .
## DRI     .
## DTE     .
## DTV     .
## DUK     .
## DVA     .
## DVN     .
## EBAY    .
## EBIX    .
## ECL     0.550
## ED      .
## EFX     .
## EIX     .
## EL      .
## EMC     .
## EMN     .
## EMR     .
## EOG     .
## EQR     .
## EQT     .
## ESRX    .
## ETFC    .
## ETN     .
## ETR     .
## EW      .
## EXC     .
## EXPD    .
## EXPE    .
## F       .
## FAST    .
## FCX     .
## FDO     0.280
## FDX    -0.377
## FE      .
```

```
## FFIV    .
## FIS     .
## FISV    .
## FITB    .
## FLIR    .
## FLR     .
## FLS     .
## FMC     .
## FOSL    .
## FRX     .
## FSLR    .
## FTI     .
## GCI    -1.143
## GD      .
## GE      .
## GILD    .
## GIS     .
## GLW     .
## GME    -0.846
## GNW     .
## GOOG    .
## GPC     .
## GPS     .
## GRMN    .
## GS      .
## GT      .
## GWW     .
## HAL     .
## HAR     .
## HAS     .
## HBAN    .
## HCBK   -0.065
## HCN    -1.067
## HCP     .
## HD      0.672
## HES     .
## HIG     .
## HOG     .
## HON     .
## HOT     .
## HP      .
## HPQ     7.713
## HRB     0.243
## HRL     .
## HRS     .
## HSP     .
## HST     .
## HSY     .
## HUM     .
## IBM     0.241
## ICE     0.047
## IFF     .
## IGT     .
## INTC    .
```

```
## INTU    .
## IP      .
## IPG     .
## IR      0.032
## IRM     .
## ISRG    .
## ITW    -3.467
## IVZ     .
## JBL     7.936
## JCI     .
## JCP     .
## JDSU    .
## JEC     .
## JNJ     .
## JNPR    .
## JPM     0.440
## JWN     .
## K       .
## KEY     .
## KIM     .
## KLAC    1.004
## KMB     .
## KMX     0.047
## KO      .
## KR      .
## KSS     .
## KSU     .
## LEG     .
## LH      .
## LLL     .
## LLTC    .
## LLY     .
## LM      .
## LMT     .
## LNC     .
## LOW     .
## LRCX    .
## LSI   -14.128
## LTD     0.658
## LUK     .
## LUV     .
## M       1.178
## MA      .
## MAC    -0.910
## MAR     .
## MAS     .
## MAT     .
## MCD    -0.210
## MCHP    .
## MCK     .
## MCO     .
## MDT     .
## MET     .
## MMC    -6.150
```

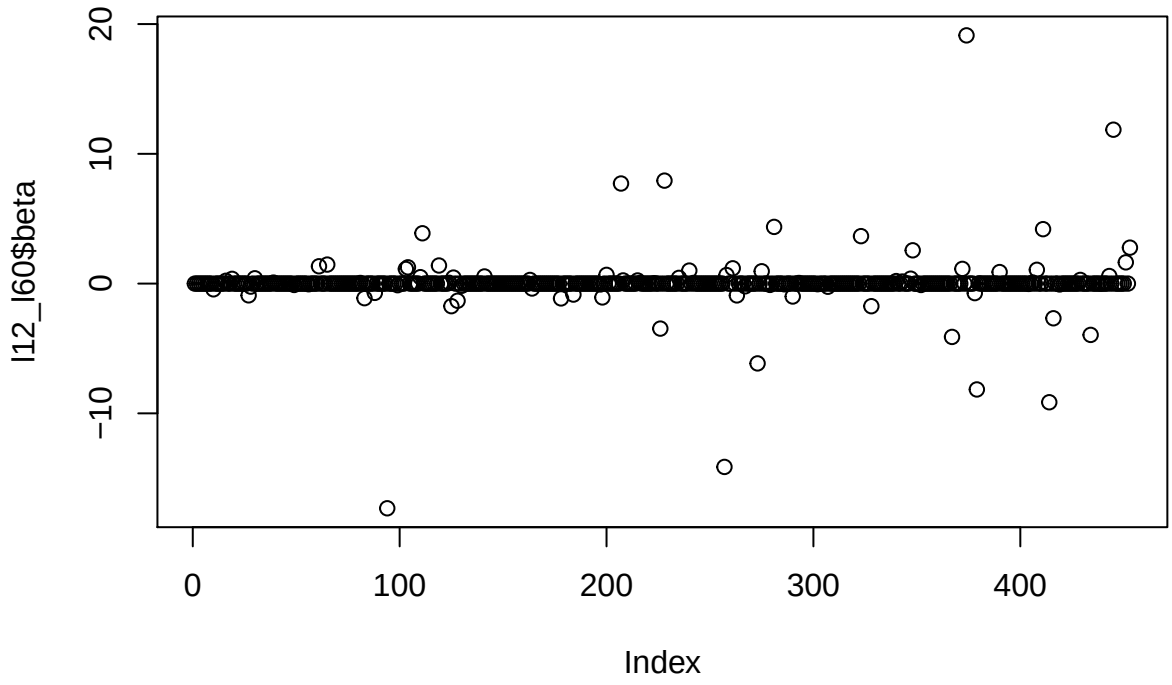
```
## MMM     .
## MO      0.950
## MOLX    .
## MON     .
## MOS     .
## MPET   -0.101
## MRK     .
## MRO     4.368
## MS      .
## MSFT    .
## MTB     .
## MU      .
## MUR    -0.052
## MWV     .
## MYL     .
## NBL     .
## NBR    -0.994
## NDAQ    .
## NE      .
## NEM     0.045
## NFLX    .
## NFX     .
## NI      .
## NKE     .
## NOC     .
## NOV     .
## NRG     .
## NSC     .
## NTAP   -0.045
## NTRS    .
## NU     -0.014
## NUE     .
## NVDA    .
## NWL    -0.241
## NYX     .
## OI      .
## OKE     .
## OMC     .
## ORCL   -0.001
## ORLY    .
## OXY     .
## PAYX    .
## PBCT    .
## PBI     .
## PCAR    .
## PCG     .
## PCL     .
## PCLN    .
## PCP     .
## PDCO    3.657
## PEG     .
## PEP     .
## PETM    .
## PFE     .
```

```
## PFG    -1.742
## PG      .
## PGR     .
## PH      .
## PHM     .
## PIR     .
## PKI     .
## PLL     .
## PNC     .
## PNW     .
## POM     .
## PPG     .
## PPL     0.193
## PRGO    .
## PRU     .
## PSA     0.158
## PVH     .
## PWR     .
## PX      .
## PXD     0.379
## QCOM    2.565
## R      -0.011
## RAI     .
## RDC     .
## REGN   -0.106
## RF      .
## RHI     .
## RHT     .
## RL      .
## ROK     .
## ROP     .
## ROST    .
## RRC     .
## RSG     .
## RTN     .
## S       .
## SAI     .
## SBUX    .
## SCG     .
## SCHW   -4.109
## SE      .
## SEE     .
## SHW     .
## SIAL    .
## SJM     1.134
## SLB     .
## SLM    19.128
## SNA     .
## SNDK    .
## SO      .
## SPG    -0.733
## SPLS   -8.157
## SRCL    0.024
## SRE     .
```

```
## STI     .
## STJ     .
## STT     .
## STX     .
## SWK     .
## SWN     .
## SWY     .
## SYK     .
## SYMC    0.888
## SYY     .
## T       .
## TE      .
## TEG     .
## TEL     .
## TER     .
## TGT     .
## THC     .
## TIF     .
## TJX     0.001
## TMK     .
## TMO     .
## TROW    .
## TRV    -0.024
## TSN     .
## TSO     0.081
## TSS     .
## TWC     1.063
## TWX     .
## TXN     .
## TXT     4.195
## TYC     .
## UNH     .
## UNM    -9.144
## UNP     .
## UPS    -2.672
## URBN    .
## USB     .
## UTX    -0.064
## VAR     .
## VFC     .
## VLO     .
## VMC     .
## VNO     .
## VRSN    .
## VTR     .
## VZ      .
## WAG     .
## WAT     0.283
## WDC     .
## WEC     .
## WFC     .
## WHR     .
## WIN    -3.954
## WLP     .
```

```
## WMB     .
## WMT     .
## WPO     .
## WU     -0.011
## WY      .
## WYN     .
## WYNN    .
## X       0.585
## XEL     .
## XL     11.860
## XLNX    .
## XOM     .
## XRAY    .
## XRX     .
## YHOO    .
## YUM     1.637
## ZION    .
## ZMH     2.771
```

```
plot(l12_l60$beta)
```



In both the above plots which were created similar to part 2 of this question, we see a straight black line at 0 indicating that both these models are even more sparse than the ones we built in part 2. We shall now use them to check the stability of the portfolio.

```
v1new = round(l12_f60$beta,4)
v2new = round(l12_l60$beta,4)
v1new - v2new
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##             s0
## A        .
## AA       .
## AAPL     .
## ABC      .
## ABT      .
## ACE      .
## ACN      .
## ADBE     .
## ADI      .
## ADM      0.4412
## ADP      .
## ADSK     .
## AEE      .
## AEP      .
## AES      .
## AET     -0.2161
## AFL      1.4151
## AGN      .
## AIG     -0.3637
## AIV      .
## AIZ      .
## AKAM     .
## ALL      .
## ALTR     .
## ALXN     .
## AMAT     .
## AMD      0.9047
## AMGN     0.2320
## AMP      .
## AMT     -0.4090
## AMZN     .
## AN       .
## ANF      .
## APA      .
## APC      .
## APD      .
## APH      .
## APOL     .
## ARG     -0.0766
## ATI      .
## AVB      .
## AVP      .
## AVY      .
## AXP      .
## AZO      .
## BA       .
## BAC      .
```

```
## BAX     .
## BBBY    0.1043
## BBT     .
## BBY     .
## BCR     1.2654
## BDX     .
## BEN     .
## BHI     .
## BIIB    0.0485
## BK      .
## BLK     .
## BLL     .
## BMC     .
## BMS    -1.3339
## BMY     .
## BRCM    .
## BSX     .
## BTU    -1.4658
## BWA     .
## BXP     .
## C       .
## CA      .
## CAG     .
## CAH     .
## CAM     .
## CAT     .
## CB      .
## CBG     .
## CCE     .
## CCI     .
## CCL     .
## CELG    .
## CERN    .
## CF     -0.0460
## CHK     .
## CHRW    1.1244
## CI      .
## CINF    .
## CL      .
## CLF     .
## CLX     0.7143
## CMA     .
## CMCSA   .
## CME     .
## CMI     .
## CMS     .
## CMT    17.3141
## CNP     .
## CNX     .
## COF     .
## COG     .
## COH     0.1219
## COHU    .
## COL     .
```

```
## COP     .
## COST   -1.1270
## COV    -1.2535
## CPB     1.1837
## CRM     .
## CSC     .
## CSCO    .
## CSX     .
## CTAS   -0.4955
## CTL    -3.6055
## CTSH    .
## CTXS    .
## CVC     .
## CVS     .
## CVX     .
## D      -0.2797
## DD      .
## DE     -1.3929
## DELL    .
## DFS     .
## DGX     .
## DHI     .
## DHR    -0.0574
## DIS     1.7411
## DISCA  -0.4623
## DLTR    .
## DNB     1.3235
## DNR     .
## DO      0.1674
## DOV     .
## DOW     .
## DRI     .
## DTE     .
## DTV     .
## DUK    16.3539
## DVA    -0.3204
## DVN     .
## EBAY    .
## EBIX    .
## ECL    -0.5498
## ED      .
## EFX     .
## EIX     .
## EL      .
## EMC     .
## EMN     .
## EMR     .
## EOG     .
## EQR     .
## EQT     .
## ESRX    .
## ETFC    .
## ETN     .
## ETR     .
```

```
## EW      .
## EXC     .
## EXPD    .
## EXPE    .
## F       .
## FAST    .
## FCX     .
## FDO    -0.2798
## FDX     0.3774
## FE      .
## FFIV    .
## FIS     .
## FISV    .
## FITB    .
## FLIR    .
## FLR     .
## FLS     .
## FMC     .
## FOSL    .
## FRX     .
## FSLR    .
## FTI     .
## GCI     1.1426
## GD      .
## GE      .
## GILD    .
## GIS     .
## GLW     .
## GME     0.8455
## GNW     .
## GOOG    .
## GPC     1.4109
## GPS     .
## GRMN    .
## GS      .
## GT      .
## GWW     .
## HAL     .
## HAR     .
## HAS     .
## HBAN    .
## HCBK    0.0646
## HCN     1.0668
## HCP     .
## HD     -0.6724
## HES     .
## HIG     .
## HOG    -0.2785
## HON     .
## HOT     .
## HP      .
## HPQ    -6.0866
## HRB    -0.2425
## HRL     .
```

```
## HRS     .
## HSP     .
## HST     .
## HSY     .
## HUM     .
## IBM    -0.2409
## ICE    -0.0470
## IFF     .
## IGT     .
## INTC    6.3741
## INTU    .
## IP      .
## IPG     .
## IR     -0.0322
## IRM     .
## ISRG    .
## ITW     3.4674
## IVZ     .
## JBL    -7.9357
## JCI     .
## JCP     .
## JDSU    .
## JEC     .
## JNJ     .
## JNPR    .
## JPM    -0.4405
## JWN     .
## K       4.1402
## KEY     .
## KIM     .
## KLAC   -1.0039
## KMB     8.5019
## KMX     3.9146
## KO      .
## KR      .
## KSS     .
## KSU     .
## LEG     .
## LH      .
## LLL     .
## LLTC    .
## LLY     .
## LM      .
## LMT     .
## LNC     .
## LOW     .
## LRCX    .
## LSI    14.1282
## LTD    -0.6576
## LUK     .
## LUV     .
## M      -1.1782
## MA      .
## MAC     0.9096
```

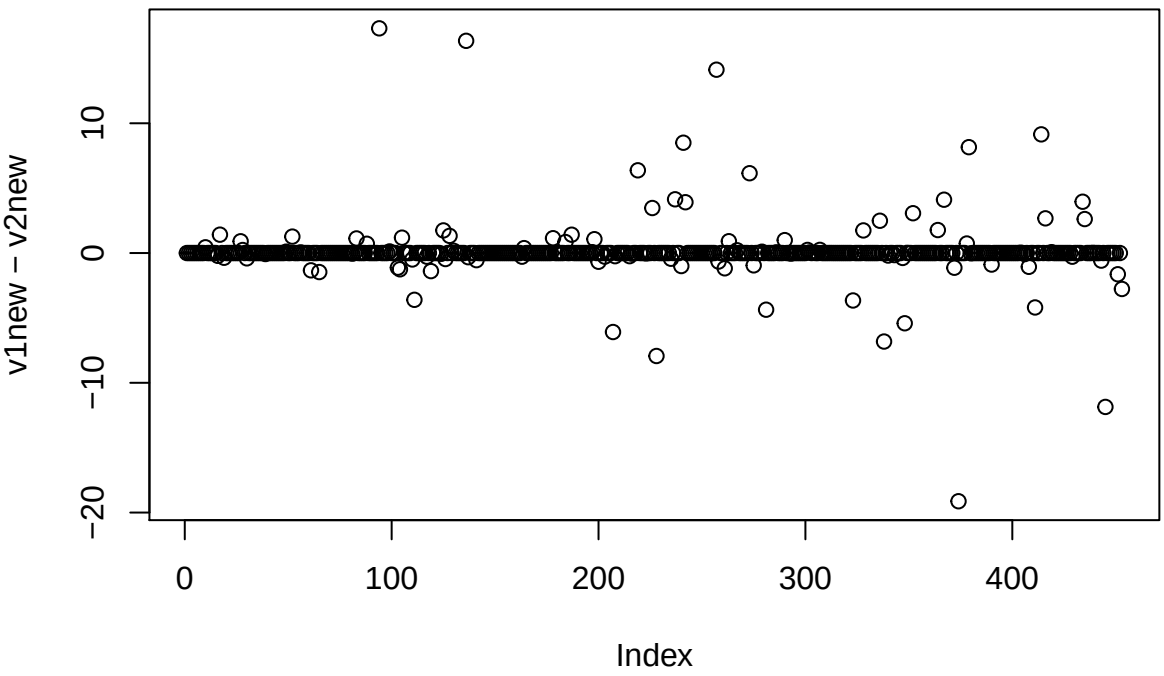
```
## MAR     .
## MAS     .
## MAT     .
## MCD     0.2101
## MCHP    .
## MCK     .
## MCO     .
## MDT     .
## MET     .
## MMC     6.1497
## MMM     .
## MO     -0.9500
## MOLX    .
## MON     .
## MOS     .
## MPET    0.1008
## MRK     .
## MRO    -4.3684
## MS      .
## MSFT    .
## MTB     .
## MU      .
## MUR     0.0519
## MWV     .
## MYL     .
## NBL     .
## NBR     0.9935
## NDAQ    .
## NE      .
## NEM    -0.0449
## NFLX    .
## NFX     .
## NI      .
## NKE     .
## NOC     .
## NOV     .
## NRG     .
## NSC     0.2377
## NTAP    0.0447
## NTRS    .
## NU      0.0145
## NUE     .
## NVDA    .
## NWL     0.2415
## NYX     .
## OI      .
## OKE     .
## OMC     .
## ORCL    0.0014
## ORLY    .
## OXY     .
## PAYX    .
## PBCT    .
## PBI     .
```

```
## PCAR    .
## PCG     .
## PCL     .
## PCLN    .
## PCP     .
## PDCO   -3.6570
## PEG     .
## PEP     .
## PETM    .
## PFE     .
## PFG     1.7415
## PG      .
## PGR     .
## PH      .
## PHM     .
## PIR     .
## PKI     .
## PLL     .
## PNC     2.4880
## PNW     .
## POM    -6.8204
## PPG     .
## PPL    -0.1931
## PRGO    .
## PRU     .
## PSA    -0.1585
## PVH     .
## PWR     .
## PX      .
## PXD    -0.3787
## QCOM   -5.4129
## R       0.0107
## RAI     .
## RDC     .
## REGN    3.0666
## RF      .
## RHI     .
## RHT     .
## RL      .
## ROK     .
## ROP     .
## ROST    .
## RRC     .
## RSG     .
## RTN     .
## S       .
## SAI     1.7709
## SBUX    .
## SCG     .
## SCHW    4.1092
## SE      .
## SEE     .
## SHW     .
## SIAL    .
```

```
## SJM    -1.1336
## SLB     .
## SLM   -19.1277
## SNA     .
## SNDK    .
## SO      .
## SPG     0.7326
## SPLS    8.1573
## SRCL   -0.0236
## SRE     .
## STI     .
## STJ     .
## STT     .
## STX     .
## SWK     .
## SWN     .
## SWY     .
## SYK     .
## SYMC   -0.8878
## SYY     .
## T       .
## TE      .
## TEG     .
## TEL     .
## TER     .
## TGT     .
## THC     .
## TIF     .
## TJX    -0.0006
## TMK     .
## TMO     .
## TROW    .
## TRV     0.0242
## TSN     .
## TSO    -0.0810
## TSS     .
## TWC    -1.0626
## TWX     .
## TXN     .
## TXT    -4.1954
## TYC     .
## UNH     .
## UNM     9.1437
## UNP     .
## UPS     2.6725
## URBN    .
## USB     .
## UTX     0.0642
## VAR     .
## VFC     .
## VLO     .
## VMC     .
## VNO     .
## VRSN    .
```

```
## VTR     .
## VZ      .
## WAG     .
## WAT    -0.2827
## WDC     .
## WEC     .
## WFC     .
## WHR     .
## WIN     3.9542
## WLP     2.6170
## WMB     .
## WMT     .
## WPO     .
## WU      0.0110
## WY      .
## WYN     .
## WYNN    .
## X      -0.5849
## XEL     .
## XL    -11.8597
## XLNX    .
## XOM     .
## XRAY    .
## XRX     .
## YHOO    .
## YUM    -1.6370
## ZION    .
## ZMH    -2.7714
```

```
plot(v1new - v2new)
```



Here, we can see that the plot does have a black straight line indicating a large number of differences are 0 but this could also be due to the fact that the weights for those corresponding predictors was 0 to begin with. As for the non-zero points, there are quite a few of those, which could mean that the portfolio is sparse but it is still not stable.

The potfolio will not be stable.
Although the matrix changes, value of the returns will have a very short range so it would be different from the previous model.
For part 1. the model would be more sparse because values are a lot closer to each other and it is harder to differentiate.
For part 2. the model would be just as unstable as the previous one even though the values have changed.

5

```
set.seed(100)

#pre-processing the data
x_data = as.matrix(my_df[,-c(1,455,456)])

#finding the best value of lambda for LASSO regression
lambdas = 10^seq(2,-3, by=-0.1)

l2_m_temp = cv.glmnet(x=x_data, y=my_df$SP.500.Level, alpha=0, lambda = lambdas)
```

```
optimal_lambda_ridge = l2_m_temp$lambda.min
optimal_lambda_ridge
```

```
## [1] 0.001
```

```
#building the model using the optimal value of lambda
l2_model = glmnet(x_data, my_df$SP.500.Level, alpha=0, lambda = optimal_lambda_ridge)
#

round(l2_model$beta,2)
```

```
## 453 x 1 sparse Matrix of class "dgCMatrix"
##          s0
## A      5.15
## AA     5.50
## AAPL   0.41
## ABC    0.62
## ABT    1.55
## ACE    2.00
## ACN    0.66
## ADBE   1.51
## ADI    2.54
## ADM    1.72
## ADP    0.20
## ADSK   0.76
## AEE    2.07
## AEP    1.55
## AES    1.39
## AET   -0.06
## AFL    0.53
## AGN   -0.22
## AIG    0.31
## AIV   -0.19
## AIZ    0.20
## AKAM   0.09
## ALL   -0.51
## ALTR  -1.25
## ALXN  -0.13
## AMAT   1.44
## AMD    2.56
## AMGN  -0.31
## AMP   -0.10
## AMT    0.01
## AMZN   0.01
## AN    -0.33
## ANF    0.29
## APA   -0.12
## APC   -0.25
## APD    0.04
## APH    0.45
## APOL   0.11
## ARG   -0.15
## ATI   -0.08
```

```
## AVB    0.15
## AVP   -0.09
## AVY    0.17
## AXP    0.16
## AZO    0.00
## BA    -0.02
## BAC    0.52
## BAX    0.56
## BBBY  -0.54
## BBT    0.66
## BBY   -0.18
## BCR   -0.01
## BDX   -0.23
## BEN    0.34
## BHI   -0.36
## BIIB  -0.06
## BK     0.19
## BLK   -0.01
## BLL    0.00
## BMC   -0.23
## BMS    0.01
## BMY    0.57
## BRCM  -0.15
## BSX   -2.42
## BTU    0.01
## BWA    0.06
## BXP    0.13
## C     -0.33
## CA    -0.92
## CAG   -0.62
## CAH    0.33
## CAM    0.10
## CAT    0.51
## CB    -1.08
## CBG   -0.51
## CCE    0.19
## CCI   -0.30
## CCL    0.16
## CELG   0.35
## CERN   0.18
## CF    -0.04
## CHK   -1.05
## CHRW   0.01
## CI     0.15
## CINF  -0.04
## CL    -0.11
## CLF   -0.02
## CLX    0.27
## CMA   -0.05
## CMCSA  0.87
## CME   -0.02
## CMI    0.00
## CMS   -1.69
## CMT   -0.13
```

```
## CNP    0.22
## CNX   -0.08
## COF    0.08
## COG    0.12
## COH    0.21
## COHU  -1.01
## COL    0.53
## COP    0.01
## COST  -0.35
## COV   -0.19
## CPB   -0.28
## CRM    0.01
## CSC    0.18
## CSCO   1.59
## CSX   -0.05
## CTAS  -0.36
## CTL    0.95
## CTSH   0.10
## CTXS  -0.12
## CVC    0.47
## CVS    0.76
## CVX    0.67
## D      0.08
## DD    -0.83
## DE    -0.18
## DELL  -0.66
## DFS   -0.58
## DGX   -0.43
## DHI   -0.25
## DHR    0.07
## DIS    0.57
## DISCA -0.36
## DLTR  -0.20
## DNB   -0.38
## DNR   -0.05
## DO    -0.02
## DOV    0.21
## DOW   -0.60
## DRI    0.73
## DTE   -0.23
## DTV    0.33
## DUK    0.03
## DVA    0.00
## DVN    0.00
## EBAY   0.72
## EBIX  -0.27
## ECL    0.30
## ED    -0.57
## EFX    0.11
## EIX   -0.23
## EL     0.15
## EMC   -0.87
## EMN    0.01
## EMR   -0.04
```

```
## EOG   -0.07
## EQR   -0.35
## EQT   -0.05
## ESRX  -0.22
## ETFC   0.30
## ETN   -0.17
## ETR    0.51
## EW     0.05
## EXC    0.25
## EXPD  -0.45
## EXPE  -0.10
## F      0.35
## FAST   0.01
## FCX    0.04
## FDO   -0.20
## FDX   -0.01
## FE    -0.63
## FFIV  -0.06
## FIS   -0.09
## FISV   0.76
## FITB   0.62
## FLIR  -0.21
## FLR   -0.10
## FLS   -0.05
## FMC   -0.06
## FOSL  -0.11
## FRX   -0.60
## FSLR  -0.02
## FTI    0.21
## GCI    0.37
## GD    -0.14
## GE     0.33
## GILD   0.01
## GIS   -0.10
## GLW    0.32
## GME    0.27
## GNW    0.34
## GOOG   0.00
## GPC    0.12
## GPS   -0.88
## GRMN  -0.39
## GS     0.07
## GT     0.10
## GWW    0.14
## HAL    0.81
## HAR   -0.22
## HAS   -0.22
## HBAN   0.58
## HCBK   0.13
## HCN   -0.21
## HCP    0.40
## HD    -0.19
## HES   -0.03
## HIG    0.09
```

```
## HOG   -0.08
## HON    0.31
## HOT   -0.24
## HP     0.09
## HPQ   -0.42
## HRB    0.45
## HRL   -0.25
## HRS   -0.28
## HSP    0.16
## HST   -0.70
## HSY    0.29
## HUM    0.02
## IBM   -0.07
## ICE   -0.05
## IFF   -0.21
## IGT   -0.08
## INTC   0.27
## INTU   0.19
## IP     0.61
## IPG   -2.92
## IR    -0.14
## IRM    0.08
## ISRG   0.00
## ITW   -0.13
## IVZ   -0.89
## JBL    0.03
## JCI    0.08
## JCP    1.00
## JDSU   0.25
## JEC    0.12
## JNJ    0.50
## JNPR  -0.17
## JPM    1.55
## JWN    0.07
## K     -0.20
## KEY   -1.13
## KIM    0.23
## KLAC   0.17
## KMB   -0.36
## KMX    0.04
## KO     0.11
## KR    -0.74
## KSS   -0.10
## KSU   -0.02
## LEG    0.49
## LH    -0.27
## LLL    0.24
## LLTC  -1.84
## LLY    0.15
## LM     0.15
## LMT    0.03
## LNC   -0.34
## LOW    2.14
## LRCX   0.04
```

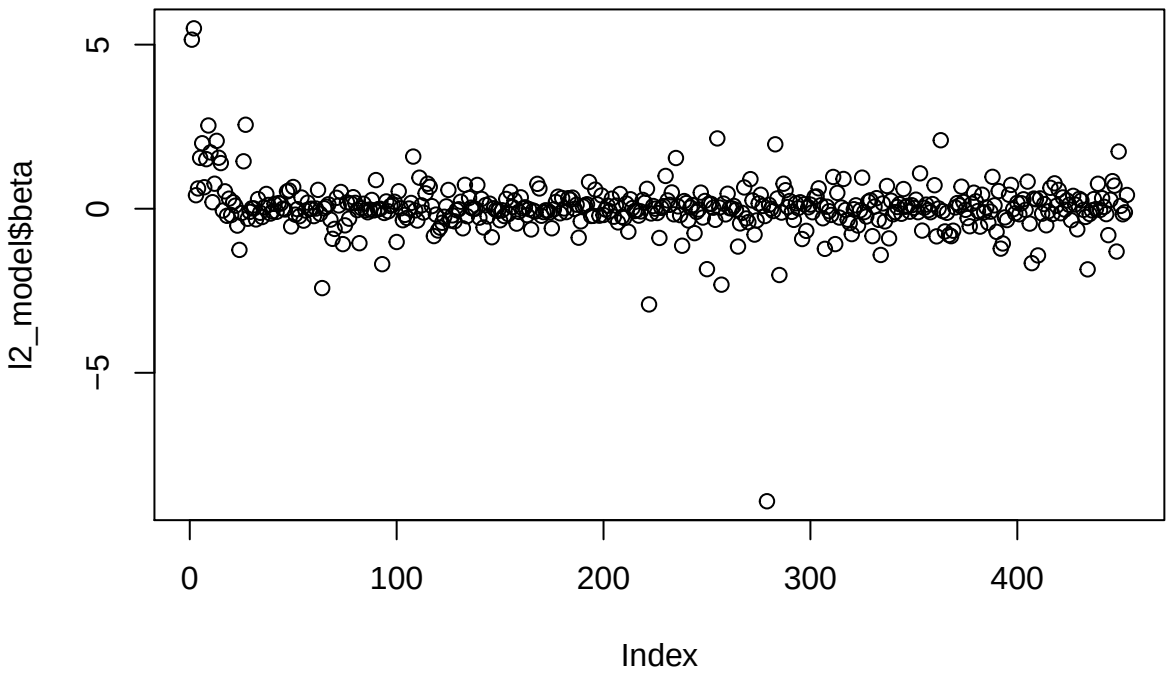
```
## LSI   -2.31
## LTD    0.16
## LUK   -0.17
## LUV    0.46
## M      0.01
## MA     0.01
## MAC    0.08
## MAR   -0.04
## MAS   -1.15
## MAT   -0.45
## MCD   -0.13
## MCHP   0.65
## MCK   -0.30
## MCO   -0.41
## MDT    0.90
## MET    0.25
## MMC   -0.79
## MMM   -0.36
## MO     0.16
## MOLX   0.43
## MON    0.08
## MOS   -0.22
## MPET  -8.91
## MRK    0.12
## MRO    0.00
## MS    -0.07
## MSFT   1.96
## MTB    0.31
## MU    -2.02
## MUR   -0.12
## MWV    0.76
## MYL    0.56
## NBL    0.05
## NBR    0.23
## NDAQ  -0.10
## NE    -0.34
## NEM    0.16
## NFLX   0.02
## NFX    0.20
## NI    -0.92
## NKE    0.13
## NOC   -0.67
## NOV    0.04
## NRG    0.12
## NSC   -0.11
## NTAP   0.36
## NTRS   0.37
## NU     0.62
## NUE    0.10
## NVDA  -0.29
## NWL   -1.22
## NYX    0.06
## OI    -0.08
## OKE   -0.19
```

```
## OMC    0.97
## ORCL  -1.08
## ORLY   0.48
## OXY   -0.27
## PAYX   0.03
## PBCT   0.91
## PBI   -0.04
## PCAR  -0.33
## PCG   -0.44
## PCL   -0.77
## PCLN  -0.02
## PCP    0.10
## PDCO  -0.51
## PEG   -0.05
## PEP    0.95
## PETM  -0.10
## PFE   -0.24
## PFG    0.21
## PG     0.26
## PGR   -0.84
## PH     0.07
## PHM    0.32
## PIR   -0.34
## PKI   -1.41
## PLL    0.16
## PNC   -0.39
## PNW    0.69
## POM   -0.90
## PPG    0.24
## PPL   -0.17
## PRGO  -0.10
## PRU    0.05
## PSA    0.14
## PVH   -0.15
## PWR    0.60
## PX     0.06
## PXD    0.09
## QCOM  -0.09
## R      0.11
## RAI    0.02
## RDC    0.28
## REGN  -0.11
## RF     1.08
## RHI   -0.67
## RHT    0.03
## RL     0.12
## ROK   -0.04
## ROP   -0.11
## ROST   0.14
## RRC    0.72
## RSG   -0.84
## RTN    0.01
## S      2.09
## SAI   -0.07
```

```
## SBUX  -0.69
## SCG   -0.12
## SCHW  -0.80
## SE    -0.84
## SEE   -0.67
## SHW    0.04
## SIAL   0.17
## SJM    0.10
## SLB    0.67
## SLM    0.20
## SNA   -0.03
## SNDK  -0.29
## SO    -0.54
## SPG    0.15
## SPLS   0.49
## SRCL   0.22
## SRE   -0.07
## STI   -0.55
## STJ    0.43
## STT   -0.11
## STX    0.04
## SWK   -0.43
## SWN   -0.09
## SWY    0.97
## SYK    0.10
## SYMC  -0.71
## SYY    0.54
## T     -1.22
## TE    -1.06
## TEG   -0.26
## TEL   -0.35
## TER    0.43
## TGT    0.73
## THC   -0.02
## TIF   -0.14
## TJX    0.18
## TMK   -0.17
## TMO    0.17
## TROW   0.29
## TRV   -0.03
## TSN    0.83
## TSO   -0.45
## TSS   -1.66
## TWC    0.31
## TWX    0.29
## TXN   -1.42
## TXT    0.31
## TYC   -0.15
## UNH    0.14
## UNM   -0.51
## UNP   -0.08
## UPS    0.64
## URBN  -0.16
## USB    0.77
```

```
## UTX    0.09
## VAR    0.57
## VFC   -0.14
## VLO    0.34
## VMC   -0.03
## VNO    0.25
## VRSN   0.12
## VTR   -0.35
## VZ     0.39
## WAG    0.13
## WAT   -0.63
## WDC    0.30
## WEC    0.25
## WFC   -0.03
## WHR   -0.25
## WIN   -1.85
## WLP   -0.16
## WMB   -0.05
## WMT    0.29
## WPO   -0.01
## WU     0.76
## WY    -0.04
## WYN    0.32
## WYNN   0.06
## X     -0.17
## XEL   -0.81
## XL     0.27
## XLNX   0.84
## XOM    0.70
## XRAY  -1.31
## XRX    1.74
## YHOO   0.09
## YUM   -0.17
## ZION  -0.11
## ZMH    0.42
```

```r
plot(l2_model$beta)
```



Since transaction costs are no longer a concern, I would suggest using L2 norm regression to track the sp500 index. Like L1-norm, Ridge regression is also used to reduce overfitting to the data but additionally. Additionally, L2 norm does not equate the coeff of the less relevant predictors to 0, thus it could potentially increase the tracking accuracy of the model.

Since we do not have a test set to test our model on, we cannot evaluate the model quantitatively.
However, one good indicator of how well our model can approximate the SP500 index is the number of predictors that have a coefficient of 0 or close to zero.In the plot above, we can see that majority of the coefficients are near or at 0.
Another way to judge the quality of the model would be to calculate the mse (mean squared error).

Q2

a

```
data("prostate")

x_pros = prostate$x
y_pros = prostate$y
dim(x_pros)
```

```
## [1]  102 6033
```

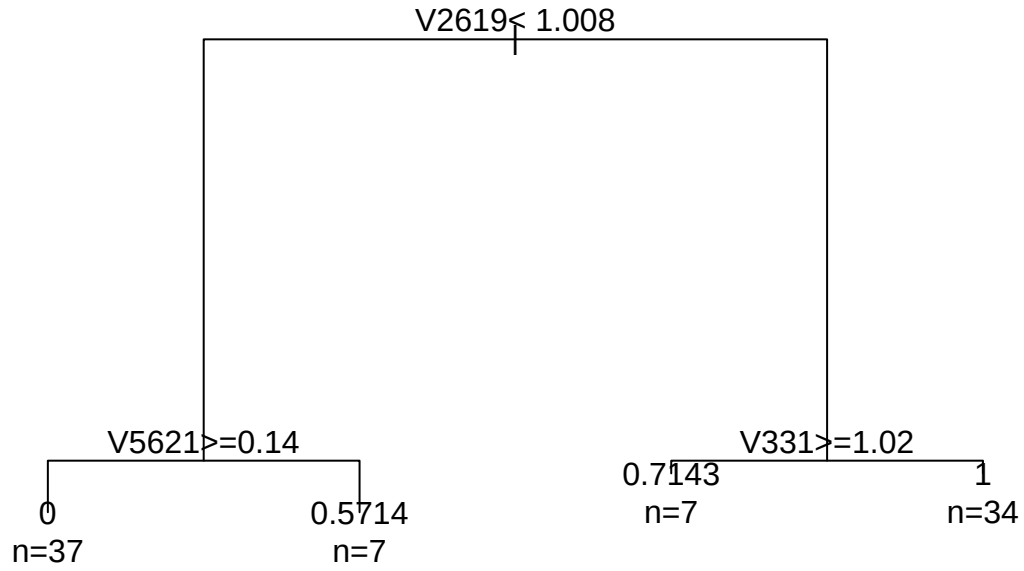
```r
length(y_pros)
```

```
## [1] 102
```

```r
#splitting into training and testing
set.seed(18)
train_pros_idx = sample(seq_len(nrow(prostate$x)), size = 85)

pros_trainX = prostate$x[train_pros_idx,]
pros_testX = prostate$x[-train_pros_idx,]

pros_trainY = prostate$y[train_pros_idx]
pros_testY = prostate$y[-train_pros_idx]


train_cmb = cbind(pros_trainX, pros_trainY)
train_cmb = as.data.frame(train_cmb)
names(train_cmb)[6034] <- "Y"
train_cmb[1:4,6020:6034]
```

```
##        V6020      V6021      V6022    V6023     V6024      V6025      V6026
## 1 -1.3017426 -1.3017426 -0.5468330 2.325471 0.3411372  0.4341927 -1.3017426
## 2  1.1667081 -1.0385988 -0.3083839 2.097410 0.3969704 -1.0385988 -1.0385988
## 3  0.3458277 -1.0796816 -1.0796816 2.329719 0.4069949 -1.0796816 -1.0796816
## 4  1.8630050 -0.9011489 -0.9011489 1.277654 0.6182401 -0.9011489 -0.9011489
##        V6027     V6028      V6029       V6030      V6031      V6032      V6033
## 1  0.3702224 1.3613367 -0.9304124  0.64819964  0.4933691  0.9034402 -1.3017426
## 2 -0.1769623 0.8186555 -0.4068861 -0.35640320 -1.0385988 -0.4600990 -1.0385988
## 3 -1.0796816 0.9373782 -0.2902809 -0.03845725 -1.0796816 -0.3765006 -0.3324107
## 4 -0.6435610 0.3833656  1.3194386 -0.80757387 -0.9011489 -0.9011489 -0.2206206
##   Y
## 1 0
## 2 1
## 3 1
## 4 0
```

```r
rpartModel_pros<-rpart(Y~.,data=train_cmb, method = "anova", control = list(cp = 0.001, xval = 10))
#plot(rpartModel_pros)
#text(rpartModel_pros,pretty=TRUE,family="Helvetica")

opar <- par() # to reset later
par(xpd=TRUE)
plot(rpartModel_pros)
text(rpartModel_pros, use.n = T)
```




```
par <- opar # restore old setting

print(rpartModel_pros)
```

```
## n= 85
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 85 21.247060 0.50588240
##   2) V2619< 1.008322 44  3.636364 0.09090909
##     4) V5621>=0.1399874 37  0.000000 0.00000000 *
##     5) V5621< 0.1399874 7  1.714286 0.57142860 *
##   3) V2619>=1.008322 41  1.902439 0.95121950
##     6) V331>=1.020071 7  1.428571 0.71428570 *
##     7) V331< 1.020071 34  0.000000 1.00000000 *
```

```
pros_testX1 = as.data.frame(pros_testX)
pros_predict<-predict(rpartModel_pros,newdata=pros_testX1)
pros_threshold=.8

is_pros<-(pros_predict>pros_threshold)

actual_pros<-pros_testY
```

```
matched<-(is_pros==actual_pros)
length_Y<-length(actual_pros)

misClassif<-1-sum(matched)/length_Y
misClassif
```

```
## [1] 0.1764706
```

```
accuracy <- 1-misClassif
accuracy
```

```
## [1] 0.8235294
```

```
###Comparing this to other models that we have - sparse logistic regression
#trying out the sparse model (without the lambda)
slg_model_cv = cv.glmnet(pros_trainX, pros_trainY, family="binomial",alpha=1)


#using this cross-validation to find the optimal lambda
optimal_lambda_slgr = slg_model_cv$lambda.min
optimal_lambda_slgr
```

```
## [1] 0.01904922
```

```
#using the optimal value of lambda to build the classifier
slgr_model = glmnet(pros_trainX, pros_trainY, alpha = 1, family = "binomial",
                lambda = optimal_lambda_slgr)

#type="response" gives the probability
yhat_slgr_prob = predict(slgr_model, newx = pros_testX, type="response")
yhat_slgr = as.numeric((yhat_slgr_prob>0.5))
yhat_slgr
```

```
##  [1] 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1
```

```
#testing the accuracy of the model by diving the number of correct predictions by number of rows
corr_pred_slgr = sum(yhat_slgr == pros_testY)
accuracy_slgr = corr_pred_slgr/length(pros_testY)
accuracy_slgr
```

```
## [1] 0.9411765
```

Based on the accuracy of the tree (~0.82) and sparse logistic regression (~0.94), we can say that the sparse logistic regression model does a better job at accurately predicting the Y values.

b.

Using bagging to improve the results of CART trees

```r
set.seed(123)

#train the bagged model
pros_bag1 = bagging(formula = Y ~ ., data = train_cmb, nbagg = 40, coob = TRUE)

pros_bag1
```

```
##
## Bagging regression trees with 40 bootstrap replications
##
## Call: bagging.data.frame(formula = Y ~ ., data = train_cmb, nbagg = 40,
##     coob = TRUE)
##
## Out-of-bag estimate of root mean squared error:  0.2878
```

```r
bag_o_pred = predict(pros_bag1, pros_testX)
bag_o_pred
```

```
##  [1] 0.22408733 0.11747639 0.14908733 0.16694447 0.05229781 0.06658353
##  [7] 0.10229781 0.82487773 0.94220296 0.96720296 0.89630630 0.96006010
## [13] 0.96006010 0.16248018 0.93506010 0.92791725 0.89577439
```

```r
bag_threshold = 0.5

bag_predictions = (bag_o_pred>bag_threshold)
bag_actual = pros_testY
accuracy_bagging = sum(bag_actual == bag_predictions)/length(bag_predictions)
accuracy_bagging
```

```
## [1] 0.8823529
```

Yes, there is notable improvement from the (0.82) to (0.88) compared to the CART results.

c.

using random forests to analyze the dataset

```r
set.seed(71)
#turning target variable into factor
train_cmb$Y = as.factor(train_cmb$Y)
pros_rf <-randomForest(Y ~.,data=train_cmb, ntree=500)
print(pros_rf)
```

```
##
## Call:
##  randomForest(formula = Y ~ ., data = train_cmb, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 77
##
```

```
##         OOB estimate of  error rate: 10.59%
## Confusion matrix:
##    0  1 class.error
## 0 39  3  0.07142857
## 1  6 37  0.13953488
```

```
rf_pred = predict(pros_rf, pros_testX)
rf_pred
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
##  0  0  0  0  0  0  0  0  1  1  1  1  1  0  1  1  1
## Levels: 0 1
```

```
accuracy_rf = sum(rf_pred == pros_testY)/length(pros_testY)
accuracy_rf
```

```
## [1] 0.9411765
```

This has further improved the accuracy of the model. The accuracy has increased from (0.88) to (0.94)

Q3

a.

```
install.packages("FunChisq")
install.packages("Metrics")
```

```
library(FunChisq)
```

```
## Warning: package 'FunChisq' was built under R version 4.0.4
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.0.4
```

```
##
## Attaching package: 'Metrics'
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

Creating the dataframe for linear regression 3(1) + 5(1)

```r
set.seed(124)
col_a = seq(0,120,3)
col_b = seq(0,160,4)
col_c = seq(0,360,9)
Y = col_a*3 - 2.7*col_b + 1.34*col_c + 9

lreg_df = data.frame(col_a, col_b, col_c, Y)
head(lreg_df)
```

```
##   col_a col_b col_c     Y
## 1     0     0     0  9.00
## 2     3     4     9 19.26
## 3     6     8    18 29.52
## 4     9    12    27 39.78
## 5    12    16    36 50.04
## 6    15    20    45 60.30
```

```r
add.noise(lreg_df, 0.1, "house",0)
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]   15   22   33   47
##  [2,]   17   24   34   45
##  [3,]   16   24   42   68
##  [4,]   15   25   51   74
##  [5,]   27   28   70   79
##  [6,]   24   31   75   84
##  [7,]   25   32   79   83
##  [8,]   29   48   83   98
##  [9,]   39   51   83  102
## [10,]   42   49   98  109
## [11,]   44   48  107  128
## [12,]   51   61  112  146
## [13,]   59   55  123  141
## [14,]   45   61  113  134
## [15,]   56   70  132  156
## [16,]   51   81  122  150
## [17,]   68   75  138  177
## [18,]   63   75  162  181
## [19,]   65   74  155  183
## [20,]   80   93  162  183
## [21,]   69   87  165  201
## [22,]   68   89  178  204
## [23,]   68   90  197  209
## [24,]   79  108  191  243
## [25,]   90  104  206  212
## [26,]   88  113  223  220
## [27,]   85  106  221  255
## [28,]   95  112  222  244
## [29,]  103  122  223  260
## [30,]  110  128  229  265
## [31,]  104  120  241  276
## [32,]  122  123  239  297
```

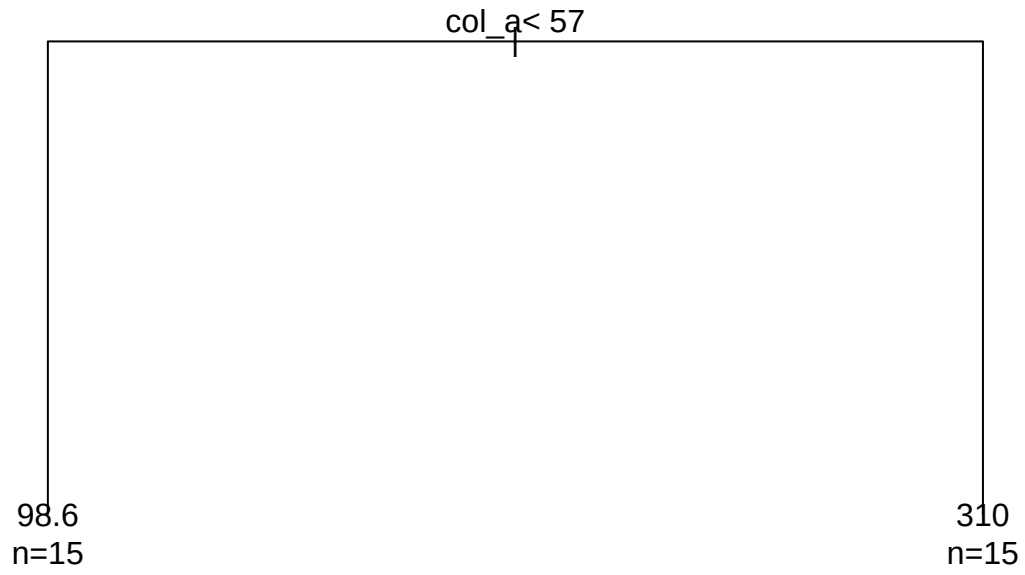
```
## [33,]  115  123  257  290
## [34,]  117  122  256  314
## [35,]  103  137  268  312
## [36,]  108  150  290  306
## [37,]  120  149  256  334
## [38,]  123  155  304  320
## [39,]  120  159  296  341
## [40,]  138  170  279  359
## [41,]  131  178  322  357
```

```
train_pros_idx2 = sample(seq_len(nrow(lreg_df)), size = 30)

lreg_trainX = lreg_df[train_pros_idx2,1:3]
lreg_testX = lreg_df[-train_pros_idx2,1:3]

lreg_trainY = lreg_df$Y[train_pros_idx2]
lreg_testY = lreg_df$Y[-train_pros_idx2]


train_cmb2 = cbind(lreg_trainX, lreg_trainY)
train_cmb2 = as.data.frame(train_cmb2)
train_cmb2
```

```
##    col_a col_b col_c lreg_trainY
## 15    42    56   126      152.64
## 5     12    16    36       50.04
## 18    51    68   153      183.42
## 27    78   104   234      275.76
## 28    81   108   243      286.02
## 19    54    72   162      193.68
## 14    39    52   117      142.38
## 38   111   148   333      388.62
## 2      3     4     9       19.26
## 4      9    12    27       39.78
## 1      0     0     0        9.00
## 16    45    60   135      162.90
## 22    63    84   189      224.46
## 8     21    28    63       80.82
## 41   120   160   360      419.40
## 9     24    32    72       91.08
## 31    90   120   270      316.80
## 32    93   124   279      327.06
## 40   117   156   351      409.14
## 33    96   128   288      337.32
## 30    87   116   261      306.54
## 3      6     8    18       29.52
## 26    75   100   225      265.50
## 24    69    92   207      244.98
## 12    33    44    99      121.86
## 25    72    96   216      255.24
## 21    60    80   180      214.20
## 37   108   144   324      378.36
## 13    36    48   108      132.12
## 7     18    24    54       70.56
```

Fitting CART to this dataset

```
rpartModel_lreg<-rpart(lreg_trainY~.,data=train_cmb2, method = "anova", control = list(cp = 0.001, xval
#plot(rpartModel_pros)
#text(rpartModel_pros,pretty=TRUE,family="Helvetica")

opar <- par() # to reset later
par(xpd=TRUE)
plot(rpartModel_lreg)
text(rpartModel_lreg, use.n = T)
```




```
par <- opar # restore old setting

print(rpartModel_lreg)
```

```
## n= 30
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 30 449173.30 204.282
##   2) col_a< 57 15  52942.58  98.604 *
##   3) col_a>=57 15  61195.56 309.960 *
```

```
lreg_pred = predict(rpartModel_lreg, newdata = as.data.frame(lreg_testX))
```

```
#building linear regression model for comparison
lreg_model = lm(lreg_trainY ~., data = train_cmb2)

lreg_model_pred = predict(lreg_model, newdata = as.data.frame(lreg_testX))
```

```
## Warning in predict.lm(lreg_model, newdata = as.data.frame(lreg_testX)):
## prediction from a rank-deficient fit may be misleading
```

```
lreg_model_pred
```

```
##      6     10     11     17     20     23     29     34     35     36     39
##  60.30 101.34 111.60 173.16 203.94 234.72 296.28 347.58 357.84 368.10 398.88
```

```
#using rmse to measure the accuracy of the 2 models

rmse_cart = rmse(lreg_testY, lreg_pred)
rmse_lreg = rmse(lreg_testY, lreg_model_pred)
rmse_cart
```

```
## [1] 59.76119
```

```
rmse_lreg
```

```
## [1] 4.437338e-14
```

Since the values are not categorical, I have used root mean squared error as a measure of fit of the model and comparing the error of 65.64 (CART) to an error « 1 we can see that CART does not seem to do well with data that has an inherently linear relationship
One of the drawbacks is that the model is built based upon the sample without making any inference about the underlying probability distribution so it is not ideal to make any generalizations on the underlying phenomenon based on the results observed.

Q4

CART

```
v1 = seq(-149,150,3)
length(v1)
```

```
## [1] 100
```

```
v2 = seq(-199,200,4)
length(v2)
```

```
## [1] 100
```

```r
v3 = seq(-449,450,9)
length(v3)
```

```
## [1] 100
```

```r
v4 = seq(-349,350,7)
length(v4)
```

```
## [1] 100
```

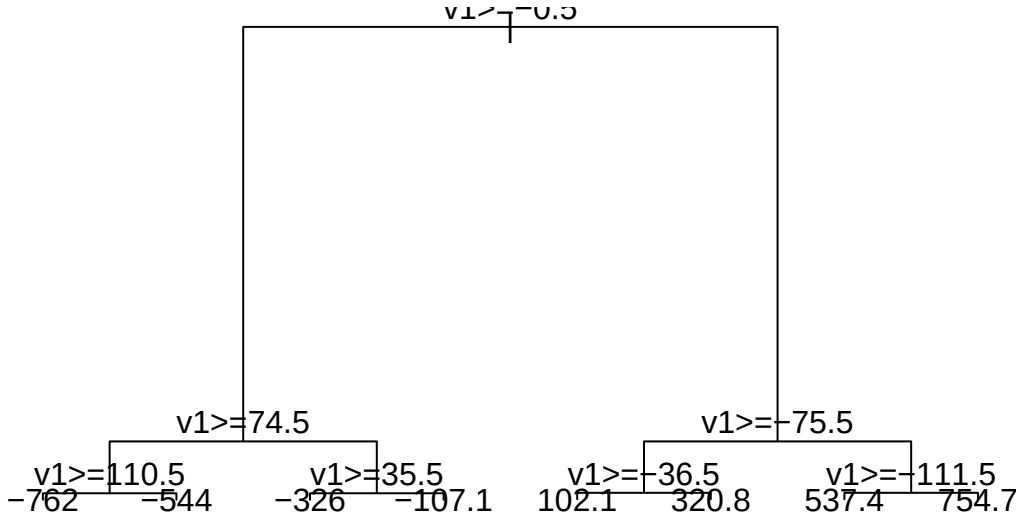
```r
v5 = seq(-249,250, 5)
length(v5)
```

```
## [1] 100
```

```r
Y = v1*3 - 2.7*v2 + 1.34*v3 -4.6*v4 + 0.9*v5 - 9.6

#creating dataframe
sample_df = data.frame(v1,v2,v3,v4,v5,Y)
#adding noise to dataframe
set.seed(9)
sample_df$Y = round(jitter(sample_df$Y), 2)

#fitting a cart tree to this dataset
rpart_sample_df<-rpart(Y~.,data=sample_df, method = "anova", control = list(cp = 0.001))

print(rpart_sample_df)
```

```
## n= 100
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 100 25347840.00   -3.0018
##    2) v1>=-0.5 50  3176937.00 -439.1278
##      4) v1>=74.5 25   396562.90 -657.3592
##        8) v1>=110.5 13    56354.11 -761.9769 *
##        9) v1< 110.5 12    43785.30 -544.0233 *
##      5) v1< 74.5 25   399127.30 -220.8964
##       10) v1>=35.5 13    55052.07 -325.9515 *
##       11) v1< 35.5 12    45167.83 -107.0867 *
##    3) v1< -0.5 50  3150311.00  433.1242
##      6) v1>=-75.5 25   397052.50  215.8460
##       12) v1>=-36.5 12    43459.57  102.0975 *
##       13) v1< -36.5 13    55007.04  320.8446 *
##      7) v1< -75.5 25   392768.20  650.4024
##       14) v1>=-111.5 12    44029.82  537.3617 *
##       15) v1< -111.5 13    53856.62  754.7477 *
```

```
plot(rpart_sample_df)
text(rpart_sample_df, use.n = T, pretty = TRUE)
```



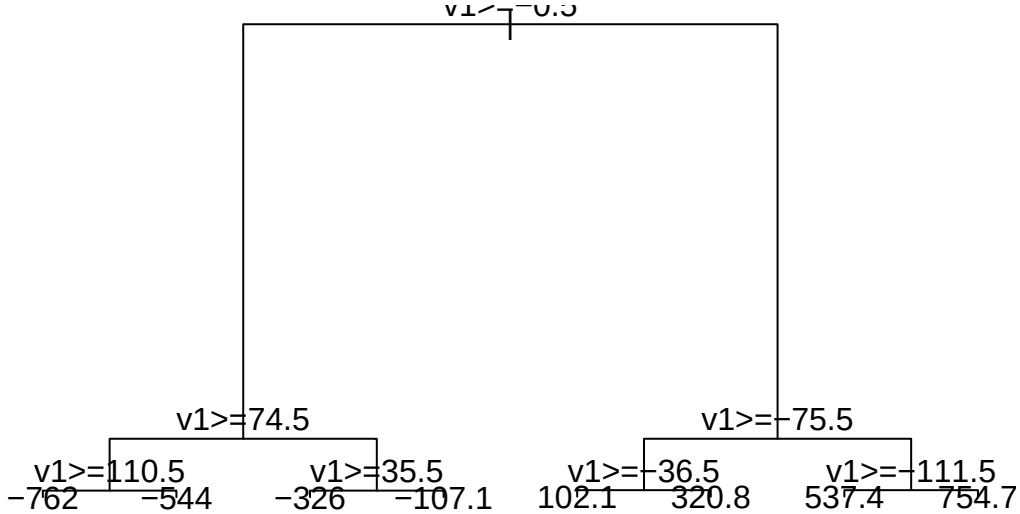
Adding 5 gaussian predictors to the dataset

```
set.seed(56)
for (idx in seq(1,5)) {
  col_name = paste("v",idx+5, sep = "")
  sample_df[col_name] = rnorm(100,0,1)
}

#fitting a cart tree to this dataset
rpart_sample_df5<-rpart(Y~.,data=sample_df, method = "anova", control = list(cp = 0.001))

print(rpart_sample_df5)
```

```
## n= 100
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 100 25347840.00   -3.0018
##    2) v1>=-0.5 50  3176937.00 -439.1278
##      4) v1>=74.5 25   396562.90 -657.3592
##        8) v1>=110.5 13    56354.11 -761.9769 *
```

```
##         9) v1< 110.5 12    43785.30 -544.0233 *
##       5) v1< 74.5 25   399127.30 -220.8964
##        10) v1>=35.5 13    55052.07 -325.9515 *
##        11) v1< 35.5 12    45167.83 -107.0867 *
##     3) v1< -0.5 50  3150311.00  433.1242
##       6) v1>=-75.5 25   397052.50  215.8460
##        12) v1>=-36.5 12    43459.57  102.0975 *
##        13) v1< -36.5 13    55007.04  320.8446 *
##       7) v1< -75.5 25   392768.20  650.4024
##        14) v1>=-111.5 12    44029.82  537.3617 *
##        15) v1< -111.5 13    53856.62  754.7477 *
```

```
plot(rpart_sample_df5)
text(rpart_sample_df5, use.n = T, pretty = TRUE)
```



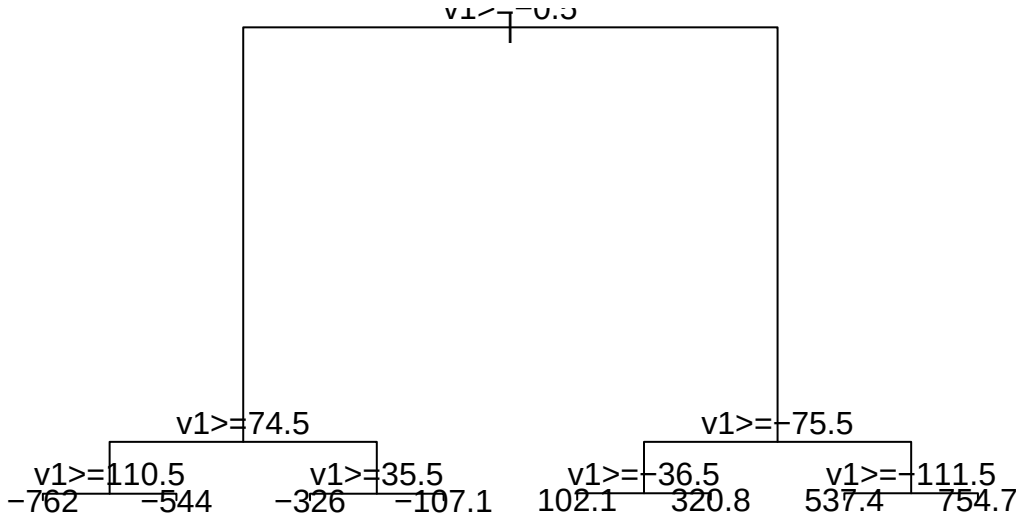
Adding 10 independent predictors

```
set.seed(56)
for (idx in seq(1,10)) {
  col_name = paste("v",idx+10, sep = "")
  sample_df[col_name] = rnorm(100,0,1)
}

#fitting a cart tree to this dataset
rpart_sample_df10<-rpart(Y~.,data=sample_df, method = "anova", control = list(cp = 0.001))
```

```
print(rpart_sample_df10)
```

```
## n= 100
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 100 25347840.00   -3.0018
##    2) v1>=-0.5 50  3176937.00 -439.1278
##      4) v1>=74.5 25   396562.90 -657.3592
##        8) v1>=110.5 13    56354.11 -761.9769 *
##        9) v1< 110.5 12    43785.30 -544.0233 *
##      5) v1< 74.5 25   399127.30 -220.8964
##       10) v1>=35.5 13    55052.07 -325.9515 *
##       11) v1< 35.5 12    45167.83 -107.0867 *
##    3) v1< -0.5 50  3150311.00  433.1242
##      6) v1>=-75.5 25   397052.50  215.8460
##       12) v1>=-36.5 12    43459.57  102.0975 *
##       13) v1< -36.5 13    55007.04  320.8446 *
##      7) v1< -75.5 25   392768.20  650.4024
##       14) v1>=-111.5 12    44029.82  537.3617 *
##       15) v1< -111.5 13    53856.62  754.7477 *
```

```
plot(rpart_sample_df10)
text(rpart_sample_df10, use.n = T, pretty = TRUE)
```



Adding 30 independent predictors

```
set.seed(56)
sample_df
```

```
##       v1   v2   v3   v4   v5       Y            v6           v7           v8
## 1   -149 -199 -449 -349 -249  858.45 -0.2409515599  1.225500610 -0.151917277
## 2   -146 -195 -440 -342 -244  839.66 -0.5004061859  0.624054179  0.342270173
## 3   -143 -191 -431 -335 -239  823.47 -0.3952413568 -0.221553868 -0.189577536
## 4   -140 -187 -422 -328 -234  806.09 -0.7952692429  1.286187643  0.930752541
## 5   -137 -183 -413 -321 -229  790.20  0.5399610863  0.227237240 -1.407887998
## 6   -134 -179 -404 -314 -224  770.65  1.4889312905 -0.148300290 -0.520130610
## 7   -131 -175 -395 -307 -219  754.96  0.4676143976  0.199120803 -0.857104991
## 8   -128 -171 -386 -300 -214  737.37  0.2606337045  0.493100500 -0.107502026
## 9   -125 -167 -377 -293 -209  721.97  0.6830675187  1.064630236  1.403735019
## 10  -122 -163 -368 -286 -204  706.73 -0.0071913952 -0.713674115  1.035841023
## 11  -119 -159 -359 -279 -199  683.34 -0.3106825505 -1.884762538 -1.881548171
## 12  -116 -155 -350 -272 -194  665.16  0.4480623988  0.500294962 -0.179401495
## 13  -113 -151 -341 -265 -189  653.67 -0.0419980916  0.542014622  0.107947877
## 14  -110 -147 -332 -258 -184  632.27 -0.9318072364 -1.660325523 -0.935952073
## 15  -107 -143 -323 -251 -179  616.13 -1.1601490285  0.640381437  0.832328538
## 16  -104 -139 -314 -244 -174  598.74 -0.4956744530  0.750025021  1.405121380
## 17  -101 -135 -305 -237 -169  580.63  0.6271587449  0.260500537  1.505850257
## 18   -98 -131 -296 -230 -164  567.10  0.2551291357  0.306416826 -0.292911997
## 19   -95 -127 -287 -223 -159  545.46  0.9035921626  0.349790902 -1.972907298
## 20   -92 -123 -278 -216 -154  528.92  0.7099322160 -0.004376892 -0.694522138
## 21   -89 -119 -269 -209 -149  514.26  0.7055047398 -1.355443836  0.101042548
## 22   -86 -115 -260 -202 -144  490.84  1.4065252389 -0.459847786 -0.561556149
## 23   -83 -111 -251 -195 -139  475.43  0.0632406289 -0.889954698  0.237249417
## 24   -80 -107 -242 -188 -134  456.59  0.7251589952  0.700283474 -0.427355324
## 25   -77 -103 -233 -181 -129  441.97  0.6098878676 -0.662280220  1.135883677
## 26   -74  -99 -224 -174 -124  427.15  0.9884836849 -0.076941329 -0.185731565
## 27   -71  -95 -215 -167 -119  406.19 -0.9307324103  0.513876206  1.602782145
## 28   -68  -91 -206 -160 -114  388.66  0.0438176040  0.971917821 -0.696076111
## 29   -65  -87 -197 -153 -109  375.20 -0.5523165961  0.081310754 -0.736432829
## 30   -62  -83 -188 -146 -104  356.95 -0.5427445647  0.120031820 -1.197985470
## 31   -59  -79 -179 -139  -99  335.10 -0.2451414305 -0.724977200 -1.525818372
## 32   -56  -75 -170 -132  -94  322.92 -0.5493599268 -1.528978236 -0.556147866
## 33   -53  -71 -161 -125  -89  302.05 -1.2486422728  0.427935637 -1.455263044
## 34   -50  -67 -152 -118  -84  287.54  0.8918525312 -0.550099878  0.441092238
## 35   -47  -63 -143 -111  -79  264.01 -0.8190125625 -0.393093074  1.527695044
## 36   -44  -59 -134 -104  -74  252.51  0.2310438706  1.213474132  0.900190735
## 37   -41  -55 -125  -97  -69  234.63  0.0949280757  0.491639947  0.580241598
## 38   -38  -51 -116  -90  -64  218.07 -0.9952712043  0.893262155  0.876925306
## 39   -35  -47 -107  -83  -59  195.85 -0.0002351228  0.181178421 -0.243254687
## 40   -32  -43  -98  -76  -54  181.91 -0.3887243815  0.517821354  0.405658854
## 41   -29  -39  -89  -69  -49  165.87 -0.9786337533  0.846209190 -0.788761966
## 42   -26  -35  -80  -62  -44  146.81 -1.7305345798 -0.068879602  0.204987677
## 43   -23  -31  -71  -55  -39  128.35 -1.5918192665 -0.422158898 -1.132761426
## 44   -20  -27  -62  -48  -34  108.24 -0.3848600967 -0.017795778 -0.784996013
## 45   -17  -23  -53  -41  -29   92.26 -0.4276127107  0.414811358 -0.840506911
## 46   -14  -19  -44  -34  -24   73.51 -1.0049403786  0.132243342 -0.003867248
## 47   -11  -15  -35  -27  -19   60.30  1.9172156528 -0.892446297 -1.230157805
## 48    -8  -11  -26  -20  -14   41.87 -0.4309173425 -0.813378903  1.566792072
```

```
## 49    -5   -7  -17  -13   -9   21.48  1.0135295191 -0.107416932 -0.408560418
## 50    -2   -3   -8   -6   -4    8.72  0.2875253510 -1.021197796 -0.666296536
## 51     1    1    1    1    1   -8.28  0.9750039211 -1.860732179  0.649650395
## 52     4    5   10    8    6  -28.85  0.4046041601  1.234208463 -0.514176262
## 53     7    9   19   15   11  -45.94 -0.4238476787 -0.022749539 -1.019647250
## 54    10   13   28   22   16  -61.28  0.3214283503 -0.580278337 -0.387591877
## 55    13   17   37   29   21  -83.19  0.0927773068  1.495510224  0.589880355
## 56    16   21   46   36   26  -96.06 -0.3024593154  0.980126688  0.666017732
## 57    19   25   55   43   31 -115.20  0.1768290635 -0.410955338  0.444394486
## 58    22   29   64   50   36 -132.05 -1.2058767766 -0.403685612  1.227167925
## 59    25   33   73   57   41 -150.20 -0.1123782065 -1.991317815 -0.507235088
## 60    28   37   82   64   46 -171.04  0.8591692790 -1.365028151  0.092927941
## 61    31   41   91   71   51 -188.23  0.7014726834 -1.206208074  0.480298167
## 62    34   45  100   78   56 -204.72  0.6735403265 -1.792032949  0.855044614
## 63    37   49  109   85   61 -221.79 -0.5954875971 -0.746268782  1.611478383
## 64    40   53  118   92   66 -241.32  0.7211050366 -0.802174256 -0.242578859
## 65    43   57  127   99   71 -253.42 -0.3366204884  0.858351825  0.145773584
## 66    46   61  136  106   76 -274.92  1.9635013336 -0.827280076  1.218454397
## 67    49   65  145  113   81 -290.25  0.5616664820 -1.373521775  0.875538187
## 68    52   69  154  120   86 -309.31  0.0197743187 -0.786579550  0.837503255
## 69    55   73  163  127   91 -325.49 -0.0663598314 -0.589244136  1.004529176
## 70    58   77  172  134   96 -341.02  0.2150193574 -1.050213630 -0.675635366
## 71    61   81  181  141  101 -362.80 -1.4290332001 -2.053505021 -0.568979976
## 72    64   85  190  148  106 -380.00 -0.2389943340 -1.691014897 -0.993083963
## 73    67   89  199  155  111 -394.33 -0.2509436741 -2.046334252  0.655437948
## 74    70   93  208  162  116 -410.50 -0.0171541648 -0.035687102 -0.038875890
## 75    73   97  217  169  121 -432.22  0.8905019968  1.427050621 -1.471898750
## 76    76  101  226  176  126 -450.23 -2.0707405988  0.818863198 -0.079026293
## 77    79  105  235  183  131 -464.14  0.6083527010 -0.833754932  0.247578463
## 78    82  109  244  190  136 -479.15 -2.7480265840 -2.159072370  1.107401566
## 79    85  113  253  197  141 -499.73 -0.2032720539 -1.052605673  0.852987885
## 80    88  117  262  204  146 -518.29 -0.7836918002 -1.413970626 -0.002678986
## 81    91  121  271  211  151 -536.84 -1.8750320378 -0.347717305 -0.467198858
## 82    94  125  280  218  156 -552.74 -0.9452857415 -1.428297205  1.798616768
## 83    97  129  289  225  161 -572.99  0.1123408749  0.622235746 -1.211704049
## 84   100  133  298  232  166 -586.67 -0.7599140986 -0.097143381  1.061161556
## 85   103  137  307  239  171 -606.46 -1.6048627251 -1.588308990 -1.556682251
## 86   106  141  316  246  176 -622.55  0.2453124866  0.423245707 -0.694246106
## 87   109  145  325  253  181 -638.49  0.9144128310  0.651208604  0.406875942
## 88   112  149  334  260  186 -659.61  1.1851522706 -0.473779681  0.862745457
## 89   115  153  343  267  191 -671.26  1.4513745352 -0.310505772 -0.201441073
## 90   118  157  352  274  196 -689.15  2.0667591379  1.272906319 -0.221122261
## 91   121  161  361  281  201 -709.81 -1.6450098772  0.674254429  0.550772539
## 92   124  165  370  288  206 -728.63  0.4678929348 -1.250755537  0.731823799
## 93   127  169  379  295  211 -746.46 -0.1538968816  0.170048172 -0.536460424
## 94   130  173  388  302  216 -758.45  0.9130619782  1.687402610  0.906890273
## 95   133  177  397  309  221 -781.77  0.0790983618  0.176778145 -1.671393315
## 96   136  181  406  316  226 -794.91 -2.2963872688 -1.340870694 -0.770175166
## 97   139  185  415  323  231 -816.22 -0.2368767829 -0.041389714 -3.144212238
## 98   142  189  424  330  236 -830.99 -1.4536051288 -0.657478081 -1.325449363
## 99   145  193  433  337  241 -850.24  2.0239178292  0.360478004  1.189659073
## 100  148  197  442  344  246 -868.20 -0.3113731257  0.310805262  0.192094489
##                v9         v10           v11          v12          v13
## 1    0.308907245 -0.97657538 -0.2409515599  1.225500610 -0.151917277
```

```
## 2    0.910696318  0.62931097 -0.5004061859  0.624054179  0.342270173
## 3   -0.697892531 -1.38063516 -0.3952413568 -0.221553868 -0.189577536
## 4   -0.309589328 -0.15521143 -0.7952692429  1.286187643  0.930752541
## 5    0.063203173  1.48992757  0.5399610863  0.227237240 -1.407887998
## 6   -0.658719941 -0.71283467  1.4889312905 -0.148300290 -0.520130610
## 7    0.389370620  1.03623308  0.4676143976  0.199120803 -0.857104991
## 8   -1.084839774 -0.51793304  0.2606337045  0.493100500 -0.107502026
## 9   -0.599335214  0.36788184  0.6830675187  1.064630236  1.403735019
## 10  -0.921353246  2.00024695 -0.0071913952 -0.713674115  1.035841023
## 11   0.187187964 -1.69218973 -0.3106825505 -1.884762538 -1.881548171
## 12   0.549441536  0.89528617  0.4480623988  0.500294962 -0.179401495
## 13   1.580358525 -1.22435856 -0.0419980916  0.542014622  0.107947877
## 14   0.030466985  1.23914221 -0.9318072364 -1.660325523 -0.935952073
## 15   0.365891013 -0.89835592 -1.1601490285  0.640381437  0.832328538
## 16   1.140937135  2.11146457 -0.4956744530  0.750025021  1.405121380
## 17  -0.143826630  0.76945379  0.6271587449  0.260500537  1.505850257
## 18  -0.408530269  0.86943599  0.2551291357  0.306416826 -0.292911997
## 19  -0.528601092  0.10231522  0.9035921626  0.349790902 -1.972907298
## 20  -0.721757347  1.45292951  0.7099322160 -0.004376892 -0.694522138
## 21  -1.278245554 -0.19834476  0.7055047398 -1.355443836  0.101042548
## 22  -0.789446896  0.95771040  1.4065252389 -0.459847786 -0.561556149
## 23  -0.540529024  1.96060975  0.0632406289 -0.889954698  0.237249417
## 24  -0.303764499 -0.59394511  0.7251589952  0.700283474 -0.427355324
## 25  -1.709851200 -0.49620749  0.6098878676 -0.662280220  1.135883677
## 26   0.381428762  0.88069614  0.9884836849 -0.076941329 -0.185731565
## 27  -0.743400974  0.08238815 -0.9307324103  0.513876206  1.602782145
## 28   1.051670347  0.08901492  0.0438176040  0.971917821 -0.696076111
## 29  -0.948883248  0.98369357 -0.5523165961  0.081310754 -0.736432829
## 30  -0.962215387  0.72922446 -0.5427445647  0.120031820 -1.197985470
## 31  -0.378957256  1.66294865 -0.2451414305 -0.724977200 -1.525818372
## 32   1.110285961  1.20318191 -0.5493599268 -1.528978236 -0.556147866
## 33  -0.920953696 -0.50349070 -1.2486422728  0.427935637 -1.455263044
## 34   0.577056564  0.81562542  0.8918525312 -0.550099878  0.441092238
## 35  -0.412540541 -0.05382810 -0.8190125625 -0.393093074  1.527695044
## 36  -0.169105194  1.46042030  0.2310438706  1.213474132  0.900190735
## 37   1.196799636 -1.88311767  0.0949280757  0.491639947  0.580241598
## 38   0.947884288 -2.06863640 -0.9952712043  0.893262155  0.876925306
## 39   0.083965188  0.51012029 -0.0002351228  0.181178421 -0.243254687
## 40   0.236276311 -0.85985641 -0.3887243815  0.517821354  0.405658854
## 41  -0.009427075 -0.53006178 -0.9786337533  0.846209190 -0.788761966
## 42   0.466493428 -0.04782237 -1.7305345798 -0.068879602  0.204987677
## 43   0.109433481 -0.38616319 -1.5918192665 -0.422158898 -1.132761426
## 44  -2.580508944 -0.80750326 -0.3848600967 -0.017795778 -0.784996013
## 45  -1.594117537  1.31820007 -0.4276127107  0.414811358 -0.840506911
## 46  -0.842178598 -0.66191086 -1.0049403786  0.132243342 -0.003867248
## 47  -0.354491738  0.68019214  1.9172156528 -0.892446297 -1.230157805
## 48   0.016583518  0.62682430 -0.4309173425 -0.813378903  1.566792072
## 49   0.603034396  0.38043407  1.0135295191 -0.107416932 -0.408560418
## 50  -1.002614347 -0.69491523  0.2875253510 -1.021197796 -0.666296536
## 51  -0.202813836  0.45096921  0.9750039211 -1.860732179  0.649650395
## 52   0.822312344  0.88656034  0.4046041601  1.234208463 -0.514176262
## 53   1.874123685  0.73146772 -0.4238476787 -0.022749539 -1.019647250
## 54   0.506842642 -0.80975566  0.3214283503 -0.580278337 -0.387591877
## 55  -0.224106131 -1.26396514  0.0927773068  1.495510224  0.589880355
```

```
## 56  -1.736027733  1.72775288 -0.3024593154  0.980126688  0.666017732
## 57  -1.296152363 -1.04788841  0.1768290635 -0.410955338  0.444394486
## 58   0.680977385 -0.50800264 -1.2058767766 -0.403685612  1.227167925
## 59  -1.825812912  0.44296594 -0.1123782065 -1.991317815 -0.507235088
## 60   1.581648556  0.78862019  0.8591692790 -1.365028151  0.092927941
## 61   2.345600043  1.17162210  0.7014726834 -1.206208074  0.480298167
## 62   0.382472333 -0.28240976  0.6735403265 -1.792032949  0.855044614
## 63   0.151146580 -0.32927476 -0.5954875971 -0.746268782  1.611478383
## 64  -2.657492147 -0.50776335  0.7211050366 -0.802174256 -0.242578859
## 65  -0.110559486  1.16245991 -0.3366204884  0.858351825  0.145773584
## 66  -0.548902399 -0.68952695  1.9635013336 -0.827280076  1.218454397
## 67   0.032510956  0.46596900  0.5616664820 -1.373521775  0.875538187
## 68  -2.879262213  0.19401464  0.0197743187 -0.786579550  0.837503255
## 69   0.283366472 -0.57714044 -0.0663598314 -0.589244136  1.004529176
## 70   0.760596959  1.39244080  0.2150193574 -1.050213630 -0.675635366
## 71   2.089988694  0.23249390 -1.4290332001 -2.053505021 -0.568979976
## 72  -0.605564312  0.61484855 -0.2389943340 -1.691014897 -0.993083963
## 73  -0.921891881 -0.31882298 -0.2509436741 -2.046334252  0.655437948
## 74  -1.443878276  0.59653263 -0.0171541648 -0.035687102 -0.038875890
## 75  -0.532238494 -1.18593318  0.8905019968  1.427050621 -1.471898750
## 76  -0.051033669  1.14951324 -2.0707405988  0.818863198 -0.079026293
## 77   1.811139938 -1.00148344  0.6083527010 -0.833754932  0.247578463
## 78  -0.265895560  0.90142179 -2.7480265840 -2.159072370  1.107401566
## 79   1.217405418 -0.53171019 -0.2032720539 -1.052605673  0.852987885
## 80   0.155645784 -0.50821555 -0.7836918002 -1.413970626 -0.002678986
## 81  -0.292227960 -2.04755991 -1.8750320378 -0.347717305 -0.467198858
## 82   1.908786597  0.73014030 -0.9452857415 -1.428297205  1.798616768
## 83   0.350322036 -0.03997010  0.1123408749  0.622235746 -1.211704049
## 84  -1.409239904 -1.15661047 -0.7599140986 -0.097143381  1.061161556
## 85  -0.457949088  0.42682296 -1.6048627251 -1.588308990 -1.556682251
## 86  -0.691714472  0.01690350  0.2453124866  0.423245707 -0.694246106
## 87  -0.420044098 -0.67778045  0.9144128310  0.651208604  0.406875942
## 88   1.152880468 -0.50063730  1.1851522706 -0.473779681  0.862745457
## 89  -0.577366539  1.88629643  1.4513745352 -0.310505772 -0.201441073
## 90   1.148851200 -0.57815130  2.0667591379  1.272906319 -0.221122261
## 91   1.910297812  0.43071219 -1.6450098772  0.674254429  0.550772539
## 92  -0.661433879 -0.89943626  0.4678929348 -1.250755537  0.731823799
## 93  -1.098227562  1.01898824 -0.1538968816  0.170048172 -0.536460424
## 94   1.006376198 -0.24059421  0.9130619782  1.687402610  0.906890273
## 95  -1.176160327  0.04025772  0.0790983618  0.176778145 -1.671393315
## 96   1.073807209 -0.28077186 -2.2963872688 -1.340870694 -0.770175166
## 97  -0.139976353 -1.31923767 -0.2368767829 -0.041389714 -3.144212238
## 98  -0.980809015  0.98625171 -1.4536051288 -0.657478081 -1.325449363
## 99  -0.587730725 -0.14173754  2.0239178292  0.360478004  1.189659073
## 100  1.580727485 -1.14141871 -0.3113731257  0.310805262  0.192094489
##              v14         v15          v16          v17          v18         v19
## 1    0.308907245 -0.97657538  0.371322971  1.374518727  1.125986558  1.52341401
## 2    0.910696318  0.62931097 -0.385597037 -0.610583573  1.067737543  1.01061823
## 3   -0.697892531 -1.38063516  0.033232137  0.429448887  1.099938604 -0.66225721
## 4   -0.309589328 -0.15521143 -0.137740060  0.124838992  0.120314448  0.84507154
## 5    0.063203173  1.48992757 -0.412405224 -0.008280392  1.048625332  0.40734906
## 6   -0.658719941 -0.71283467 -2.050450344  1.772809753  0.102074275 -0.25770566
## 7    0.389370620  1.03623308  0.095676517 -0.363628032  1.435121002  0.72125707
## 8   -1.084839774 -0.51793304 -0.898302429 -2.313902783 -1.153972252 -0.12073385
```

```
## 9   -0.599335214  0.36788184  0.533248176 -0.645610269  0.585419511  1.31143416
## 10  -0.921353246  2.00024695 -0.323059805 -0.754752157 -0.730407235 -1.98839089
## 11   0.187187964 -1.69218973  0.451605787  1.783407338  0.440672772 -0.57464169
## 12   0.549441536  0.89528617 -1.116689366 -1.121212931  0.104950092 -0.07195544
## 13   1.580358525 -1.22435856 -0.497570089 -2.296368087 -1.491365580  1.85077823
## 14   0.030466985  1.23914221 -0.167226995  0.400147223  0.317295774  0.72361326
## 15   0.365891013 -0.89835592  0.261847632 -1.088831137  0.390570567  0.31371838
## 16   1.140937135  2.11146457  0.726311139 -0.153838576  0.436700079  1.31205530
## 17  -0.143826630  0.76945379 -0.435048746  0.043446115 -0.729316443 -1.18504352
## 18  -0.408530269  0.86943599  1.069397268 -0.090271724 -0.181655291 -0.96020034
## 19  -0.528601092  0.10231522 -0.095228108 -0.057725904 -0.631213107  0.73654793
## 20  -0.721757347  1.45292951 -0.967647396  0.283007580  0.181293880 -0.13725519
## 21  -1.278245554 -0.19834476  0.629133002 -1.694875827  0.312117368  0.44426864
## 22  -0.789446896  0.95771040  1.202087097 -0.470249860 -1.936323823  0.58447892
## 23  -0.540529024  1.96060975 -1.161260221 -0.070843520  0.375953216 -0.33871028
## 24  -0.303764499 -0.59394511  0.969894271 -0.662742469  0.522362665  1.47308830
## 25  -1.709851200 -0.49620749  2.159204760 -0.904092290 -1.388453804 -1.96928400
## 26   0.381428762  0.88069614  0.445061471  0.444958458  0.245671889 -0.06676718
## 27  -0.743400974  0.08238815  0.845377513 -0.888848307  0.351147431 -0.20611624
## 28   1.051670347  0.08901492  0.975726176  0.598100876  1.028668965  0.91297230
## 29  -0.948883248  0.98369357 -0.199168203  0.970739591  0.881899348 -2.28155809
## 30  -0.962215387  0.72922446  0.727222792 -1.548908107 -0.084076654  1.59342196
## 31  -0.378957256  1.66294865 -0.640889987 -0.769049228  1.359508655 -0.66582656
## 32   1.110285961  1.20318191 -0.969725098 -0.659881669 -0.305391108  0.26876490
## 33  -0.920953696 -0.50349070  0.663311549  0.857963684 -0.040234060 -1.18585010
## 34   0.577056564  0.81562542 -0.932663552  0.435844652  0.895409285 -0.32099524
## 35  -0.412540541 -0.05382810  0.254705797 -0.972150257 -0.413509166  1.33329236
## 36  -0.169105194  1.46042030 -0.107905454  1.538499823 -1.334964433 -0.26505995
## 37   1.196799636 -1.88311767  0.802575909  0.499006633 -1.551567370 -0.22345429
## 38   0.947884288 -2.06863640  0.883218171 -0.479102328  0.458886518 -0.19287438
## 39   0.083965188  0.51012029 -0.023945230  0.531086113  0.268222811  2.42342067
## 40   0.236276311 -0.85985641 -1.182232538 -0.347615953 -0.625761475  2.01466353
## 41  -0.009427075 -0.53006178 -0.002745356 -1.771301091 -0.771282932 -0.55413946
## 42   0.466493428 -0.04782237 -0.061155111  0.872536361  0.930678683 -0.82612227
## 43   0.109433481 -0.38616319 -0.225933380 -0.813804052 -0.536140823 -1.44492453
## 44  -2.580508944 -0.80750326  1.312782139 -0.563105562  0.550545405 -0.94888064
## 45  -1.594117537  1.31820007 -0.778300549 -0.763506782 -0.421490170 -0.44465744
## 46  -0.842178598 -0.66191086 -1.524461094  0.709858699 -0.612872821 -1.04645716
## 47  -0.354491738  0.68019214  0.122604830 -2.422051130  0.507648201  1.24259772
## 48   0.016583518  0.62682430 -0.261363145 -0.669338097 -0.588075864 -1.03743623
## 49   0.603034396  0.38043407 -2.285255928  1.071145957 -0.110670768 -2.30524063
## 50  -1.002614347 -0.69491523 -0.339091846 -1.366327935 -0.127239282 -0.06536753
## 51  -0.202813836  0.45096921 -1.206516261 -0.656474145 -0.912197609  0.70435098
## 52   0.822312344  0.88656034  1.245012885  0.178493093 -0.121474465 -0.62820580
## 53   1.874123685  0.73146772  0.328483472  0.472708992  0.342029555 -1.17774615
## 54   0.506842642 -0.80975566 -0.090549243 -0.471644812  0.278262615  0.84823904
## 55  -0.224106131 -1.26396514 -0.032769632 -0.443401406  0.692396117 -0.60872853
## 56  -1.736027733  1.72775288 -1.292265590 -0.170615484 -0.945813991 -1.07077018
## 57  -1.296152363 -1.04788841 -0.387190507  0.289433935 -0.543213328  0.11156425
## 58   0.680977385 -0.50800264 -1.936310132  0.871471958 -0.365784513  0.04928616
## 59  -1.825812912  0.44296594  2.197434403 -0.785227174  1.320611951 -0.43086465
## 60   1.581648556  0.78862019  0.748031477  0.282918992 -0.750055980 -0.26299008
## 61   2.345600043  1.17162210  1.117409299 -1.711917271  0.965298732  1.20135641
## 62   0.382472333 -0.28240976  1.397457783  1.323600119  0.951007352  0.38066627
```

```
## 63   0.151146580 -0.32927476 -1.574758848 -1.154897151 -1.575019750  0.45090980
## 64  -2.657492147 -0.50776335  0.668627146 -0.422387221 -0.422909880 -0.28309290
## 65  -0.110559486  1.16245991 -0.337177869  0.858504552 -0.215791048 -0.12951872
## 66  -0.548902399 -0.68952695  0.133811603  0.094096668 -0.619264969 -0.21818040
## 67   0.032510956  0.46596900  0.334169846  0.102673642 -2.533033192 -2.20945583
## 68  -2.879262213  0.19401464  2.823777603 -1.421565811 -1.280676218  0.40461516
## 69   0.283366472 -0.57714044 -0.652351597  0.561458895  0.675747212 -0.43757841
## 70   0.760596959  1.39244080  1.697732074  0.208733455 -0.007115385 -0.40232104
## 71   2.089988694  0.23249390  1.054172536  0.139400942  1.342783177 -1.62867716
## 72  -0.605564312  0.61484855 -0.071227624  0.135384202  0.573889114  2.32979955
## 73  -0.921891881 -0.31882298 -0.148186875 -2.354353487  0.264211958  0.01678168
## 74  -1.443878276  0.59653263 -2.505075152 -0.616218973 -0.838712488  0.03824958
## 75  -0.532238494 -1.18593318 -1.736212693  0.779512450 -0.406030351 -0.66474275
## 76  -0.051033669  1.14951324 -1.303164095 -0.208676350  1.468507471 -0.41387690
## 77   1.811139938 -1.00148344 -0.224549598  0.176633046  1.376628165  0.64611043
## 78  -0.265895560  0.90142179  1.487927911  0.341457607  0.842518127  0.59622246
## 79   1.217405418 -0.53171019  0.995888722  0.005962110 -0.343035155 -0.71130352
## 80   0.155645784 -0.50821555  0.797963664  1.667155076  1.948627910 -0.22128107
## 81  -0.292227960 -2.04755991  1.223704321  0.135906605  1.437041869  0.40778193
## 82   1.908786597  0.73014030  1.611360319  0.413852046 -0.209160440 -1.57551638
## 83   0.350322036 -0.03997010  1.747123827  1.226089168 -0.823015669  0.30102146
## 84  -1.409239904 -1.15661047  0.811916781  0.320661261  1.221048983 -1.60628253
## 85  -0.457949088  0.42682296  0.122574343 -0.454543737 -1.469168379  1.50766609
## 86  -0.691714472  0.01690350  0.219572825  0.350824973  0.575302270  1.35289328
## 87  -0.420044098 -0.67778045 -0.052548757 -0.547121061  1.147067366 -0.24849580
## 88   1.152880468 -0.50063730  0.081019150 -0.295693772  0.823136619 -1.34032716
## 89  -0.577366539  1.88629643  1.009865488 -0.450516208 -0.882492158  0.55314126
## 90   1.148851200 -0.57815130  1.216006041  0.653354485  0.545927091  1.57449941
## 91   1.910297812  0.43071219 -0.438571178  1.405193075  1.552129717 -1.37673828
## 92  -0.661433879 -0.89943626  1.869908853  1.111518645 -0.014729909 -1.56460764
## 93  -1.098227562  1.01898824  0.678568013 -1.149125463 -1.038874265 -0.86772866
## 94   1.006376198 -0.24059421  0.194223461  1.449704380 -1.323818487  1.48724242
## 95  -1.176160327  0.04025772  0.121598425  0.496221035 -2.445040248  0.69470362
## 96   1.073807209 -0.28077186 -0.164572086 -2.546257132 -1.037117173  0.71261441
## 97  -0.139976353 -1.31923767  1.878729312  0.053095065  0.126884746 -1.00965918
## 98  -0.980809015  0.98625171  0.565665828 -0.637742916  0.038087221 -0.19517335
## 99  -0.587730725 -0.14173754 -2.291268984 -0.630882920  0.670844212  0.60063609
## 100  1.580727485 -1.14141871 -1.786869972  0.598716727  0.649074816 -1.21778267
##             v20
## 1    0.60119279
## 2   -0.29255727
## 3    0.65632798
## 4   -0.84987245
## 5   -0.55640111
## 6   -1.11400577
## 7    0.12501484
## 8   -0.01511479
## 9   -0.92393233
## 10   1.76324961
## 11   0.67211684
## 12   1.12707822
## 13  -2.74376371
## 14   0.58774501
## 15   1.02260430
```

```
## 16  -0.74661533
## 17  -1.73707114
## 18   1.10002616
## 19   0.63045423
## 20   0.80704730
## 21   0.10454101
## 22   0.66598849
## 23  -1.55205881
## 24   0.38698014
## 25  -1.09198736
## 26  -0.35514498
## 27   1.35643386
## 28  -0.56074834
## 29  -0.02702128
## 30  -1.70355925
## 31   0.81981562
## 32   1.13760885
## 33   0.27502764
## 34   1.05189272
## 35   0.26109482
## 36  -1.75242771
## 37  -1.68551102
## 38   0.51641853
## 39   1.15318918
## 40  -0.01587602
## 41  -1.34650356
## 42   1.04473701
## 43   1.74851847
## 44  -0.25983707
## 45   1.19425894
## 46  -0.16042303
## 47  -1.28377911
## 48   1.55852148
## 49   1.29194036
## 50   0.57955686
## 51  -0.78597194
## 52  -1.04001395
## 53  -0.51897998
## 54  -0.85454470
## 55  -0.13921144
## 56  -0.33416157
## 57   0.08839520
## 58  -2.41423399
## 59   0.58268568
## 60   0.05693550
## 61  -0.02903951
## 62   0.34837574
## 63   2.05998028
## 64   0.23870739
## 65   0.30410978
## 66  -0.89085698
## 67   1.37159330
## 68   0.73673346
## 69  -1.18737808
```

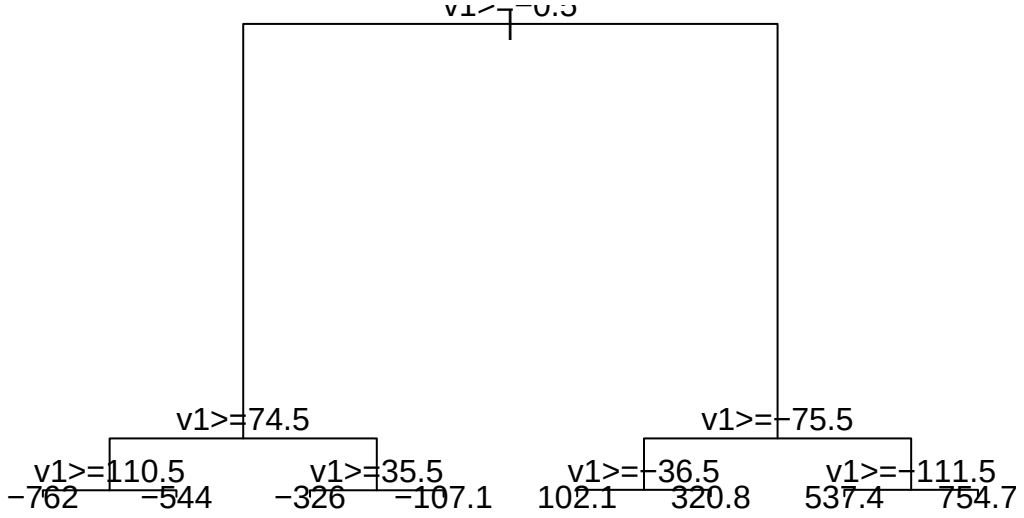
```
## 70   0.85701664
## 71   0.62589569
## 72  -0.09264993
## 73   0.27110814
## 74   0.95329207
## 75  -0.37433197
## 76   0.03953770
## 77  -0.58954118
## 78   0.69059779
## 79   0.34442738
## 80   0.40693288
## 81  -0.84136624
## 82   1.62760446
## 83   1.58292021
## 84   2.54787311
## 85  -0.78540874
## 86   0.34602644
## 87  -0.59564773
## 88  -0.76388424
## 89   0.72964515
## 90  -1.01058316
## 91  -1.49767040
## 92  -1.50003332
## 93   0.73694563
## 94   0.41383390
## 95  -0.17665928
## 96   1.01577224
## 97  -0.55973972
## 98  -0.68607963
## 99   0.87325418
## 100  0.06876842
```

```
for (idx in seq(1,30)) {
  col_name = paste("v",idx+21, sep = "")
  sample_df[col_name] = rnorm(100,0,1)
}

#fitting a cart tree to this dataset
rpart_sample_df30<-rpart(Y~.,data=sample_df, method = "anova", control = list(cp = 0.001))

print(rpart_sample_df30)
```

```
## n= 100
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 100 25347840.00   -3.0018
##    2) v1>=-0.5 50  3176937.00 -439.1278
##      4) v1>=74.5 25   396562.90 -657.3592
##        8) v1>=110.5 13    56354.11 -761.9769 *
##        9) v1< 110.5 12    43785.30 -544.0233 *
##      5) v1< 74.5 25   399127.30 -220.8964
##       10) v1>=35.5 13    55052.07 -325.9515 *
```

```
##       11) v1< 35.5 12    45167.83 -107.0867 *
##    3) v1< -0.5 50  3150311.00  433.1242
##      6) v1>=-75.5 25   397052.50  215.8460
##       12) v1>=-36.5 12    43459.57  102.0975 *
##       13) v1< -36.5 13    55007.04  320.8446 *
##      7) v1< -75.5 25   392768.20  650.4024
##       14) v1>=-111.5 12    44029.82  537.3617 *
##       15) v1< -111.5 13    53856.62  754.7477 *
```

```
plot(rpart_sample_df30)
text(rpart_sample_df30, use.n = T, pretty = TRUE)
```



Adding the independent predictors does not affect the tree. While we are not sure if it picks the "right" variables, it does pick the variables picked by the original tree and increasing the number of independent predictors does not change that. This tells us that CART is a useful technique to get a general idea of which columns have more influence on the target variables.

Random Forest

```
#using the same dataset from earlier
sample_df2 = sample_df[,c("v1","v2","v3","v4","v5", "Y")]


#fitting a cart tree to this dataset
```

```r
set.seed(10)
rf_sample_df<-randomForest(Y~.,data=sample_df2,ntree=400)

print(rf_sample_df)
```

```
##
## Call:
##  randomForest(formula = Y ~ ., data = sample_df2, ntree = 400)
##                Type of random forest: regression
##                      Number of trees: 400
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 190.6144
##                     % Var explained: 99.92
```

Adding 5 Gaussian predictors

```r
set.seed(10)
for (idx in seq(1,5)) {
  col_name = paste("v",idx+5, sep = "")
  sample_df2[col_name] = rnorm(100,0,1)
}

#refitting the model
rf_sample_df5 = randomForest(Y ~., data = sample_df2, ntree = 400)

print(rf_sample_df5)
```

```
##
## Call:
##  randomForest(formula = Y ~ ., data = sample_df2, ntree = 400)
##                Type of random forest: regression
##                      Number of trees: 400
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 333.2116
##                     % Var explained: 99.87
```

Adding 10 Gaussian Predictors

```r
set.seed(10)
for (idx in seq(1,10)) {
  col_name = paste("v",idx+10, sep = "")
  sample_df2[col_name] = rnorm(100,0,1)
}

#refitting the model
rf_sample_df10 = randomForest(Y ~., data = sample_df2, ntree = 400)

print(rf_sample_df10)
```

```
##
## Call:
##  randomForest(formula = Y ~ ., data = sample_df2, ntree = 400)
##                Type of random forest: regression
##                      Number of trees: 400
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 565.4339
##                     % Var explained: 99.78
```

Adding 30 Gaussian Predictors

```
set.seed(10)
for (idx in seq(1,30)) {
  col_name = paste("v",idx+20, sep = "")
  sample_df2[col_name] = rnorm(100,0,1)
}

#refitting the model
rf_sample_df30 = randomForest(Y ~., data = sample_df2, ntree = 400)

print(rf_sample_df30)
```

```
##
## Call:
##  randomForest(formula = Y ~ ., data = sample_df2, ntree = 400)
##                Type of random forest: regression
##                      Number of trees: 400
## No. of variables tried at each split: 16
##
##           Mean of squared residuals: 963.5006
##                     % Var explained: 99.62
```

While CART was able to select the right predictors and did not use any of the Gaussian predictors, the random forest model did not do the same. This is shown by the exponential increase in mean squared residuals: The value grew from 190 to 333, to 565 and 963 as gaussian predictors were added to the model.

Q5

In the case of the CART model, the tree is built such that every leaf in the tree has constant regression. This means that the dependent variable is constant in each leaf.
An alternative proposition is to apply piecewise linear function to each leaf in the tree (including the root.) The algorithm is as follows:
Ncurr - current node being referred to
1. At the very beginning the root node is the Ncurr and since no split has taken place we use the full dataset.
2. On the entire dataset, we create a linear regression model at node Ncurr.
3. Calculate the R^2 of the linear model at Ncurr.
3.1 If R^2 > pre-decided threshold theta, we can call it a leaf and move to step 5
3.2 If not, continue to step 4
4. This is the repeating step where we must perform n random decisions and select the one with the highest

possible R^2
4.1 We start y randomly selecting an independent variable Xi and a random threshold theta(i)
4.2 We split the dataset into Ncurr1, Ncurr2 based on the threshold Xi < theta(i)
4.3 Next, we create a linear regression model on Ncurr1 and Ncurr2 and calculate their R^2s (r1 and r2) as well as the difference in the R^2 values (diff_r2)
4.4 For the Ncurr node, we select the independent variable Xi and threshold theta(i) for which diff_r2 is maximum and Ncurr gives rise to 2 new subnodes.
5. At this point, we can say that Ncurr is a leaf (no more sub-nodes). So we travel back up the tree to see if there are any subnodes remaining to be processed and repeat step 2 onwards. If all the nodes have been process, we can end the algorithm.

Here, step 3 is becomes the exit condition, where we can end the algorithm if:
1. Ncurr's depth is greater than the max_depth of the tree
2. Ncurr's dataset is smaller than threshold for dataset size.