Midterm

Yining Qu

2024-04-19

```
library(readtext)
library(SnowballC)
library(tidytext)
```

```
data<-readtext("RedditNews.csv",skip=1)
```

```
## Warning in data.table::fread(input = path, data.table = FALSE, stringsAsFactors
## = FALSE, : Found and resolved improper quoting in first 100 rows. If the fields
## are not quoted (e.g. field separator does not appear within any field), try
## quote="" to avoid this warning.
```

```
## Warning in data.table::fread(input = path, data.table = FALSE, stringsAsFactors
## = FALSE, : Detected 36 column names but the data has 2 columns. Filling rows
## automatically. Set fill=TRUE explicitly to avoid this warning.
```

```
## Warning in data.table::fread(input = path, data.table = FALSE, stringsAsFactors
## = FALSE, : Stopped early on line 103. Expected 36 fields but found 61. Consider
## fill=TRUE and comment.char=. First discarded non-empty line: <<Country ranked
## in bottom third of global equality list, but inauguration raises hopes of
## tackling issue.">>
```

```
date<-data[2] # this is the day of the news

subset<-date=="7/1/16" # let's take a look at news headlines on 7/1/16

data[subset,3] # we have 23 news headlines
```

```
## character(0)
```

```
# Read the DJIA data
dj<-read.csv("DJIA.csv")

head(dj) # Open price, highest, lowest and close price
```

```
##         Date     Open     High      Low    Close    Volume Adj.Close
## 1 2016-07-01 17924.24 18002.38 17916.91 17949.37  82160000  17949.37
## 2 2016-06-30 17712.76 17930.61 17711.80 17929.99 133030000  17929.99
## 3 2016-06-29 17456.02 17704.51 17456.02 17694.68 106380000  17694.68
## 4 2016-06-28 17190.51 17409.72 17190.51 17409.72 112190000  17409.72
## 5 2016-06-27 17355.21 17355.21 17063.08 17140.24 138740000  17140.24
## 6 2016-06-24 17946.63 17946.63 17356.34 17400.75 239000000  17400.75
```

```
ndays<-nrow(dj) # 1989 days

# Read the words
```

```
words<-read.csv("WordsFinal.csv",header=F)

words<-words[,1]

head(words)
```

```
## [1] "ab"      "abandon" "abba"    "abbott"  "abc"     "abduct"
```

```
# Read the word-day pairings

doc_word<-read.table("WordFreqFinal.csv",header=F)

# Create a sparse matrix
library(gamlr)
```

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

```
spm<-sparseMatrix(
        i=doc_word[,1],
        j=doc_word[,2],
        x=doc_word[,3],
        dimnames=list(id=1:ndays,words=words))

dim(spm)
```

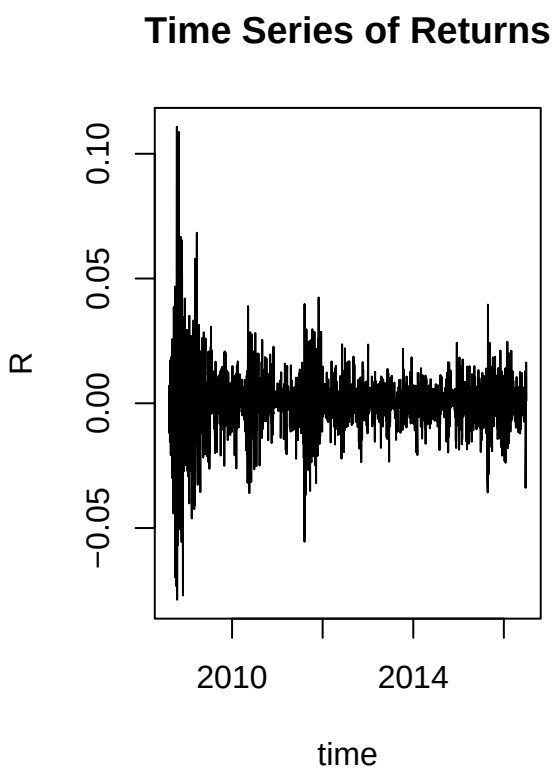
```
## [1] 1989 5271
```

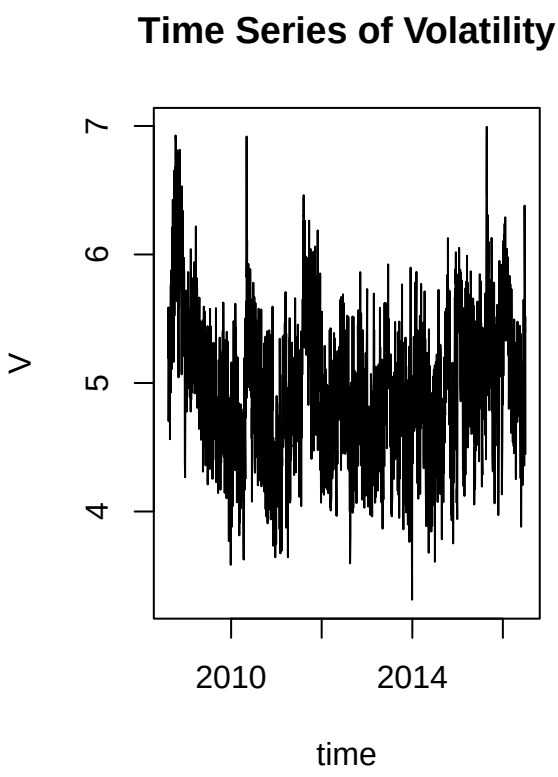
```
# We select only words at occur at least 5 times

cols<-apply(spm,2,sum)

index<-apply(spm,2,sum)>5

spm<-spm[,index]

# and words that do not occur every day

index<-apply(spm,2,sum)<ndays

spm<-spm[,index]

dim(spm) # we end up with 3183 words
```

```
## [1] 1989 3183
```

```
#  *** FDR *** analysis

spm<-spm[-ndays,]

time<-dj[-ndays,1]

time <- as.Date(time, format = "%Y-%m-%d")

# Take returns Rt

par(mfrow=c(1,2))
```

```r
R<-(dj[-ndays,7]-dj[-1,7])/dj[-1,7]

plot(R~time,type="l")
title(main = "Time Series of Returns")

# Take the log of the maximal spread

V<-log(dj[-ndays,3]-dj[-ndays,4])

plot(V~time,type="l")
title(main = "Time Series of Volatility")
```



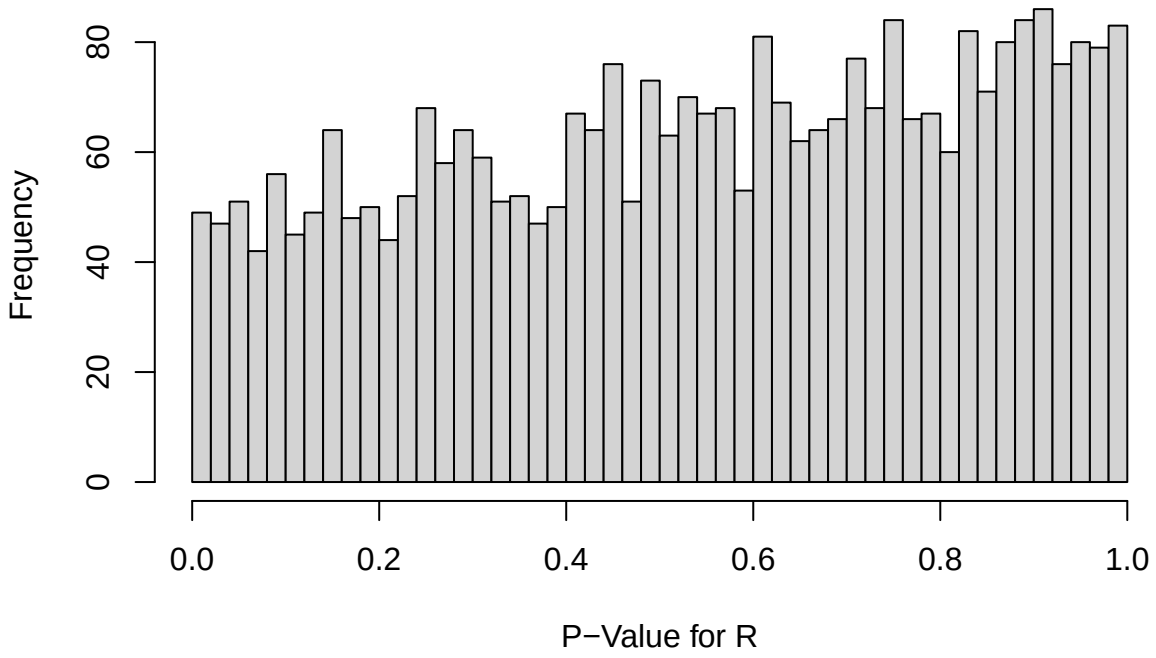



```r
# FDR: we want to pick a few words that correlate with the outcomes (returns and volatility)

# create a dense matrix of word presence

P <- as.data.frame(as.matrix(spm>0))

# we will practice parallel computing now

library(parallel)


margreg <- function(x){
    fit <- lm(Outcome~x)
    sf <- summary(fit)
    return(sf$coef[2,4])
}
```

```
cl <- makeCluster(detectCores())

# pull out stars and export to cores
```

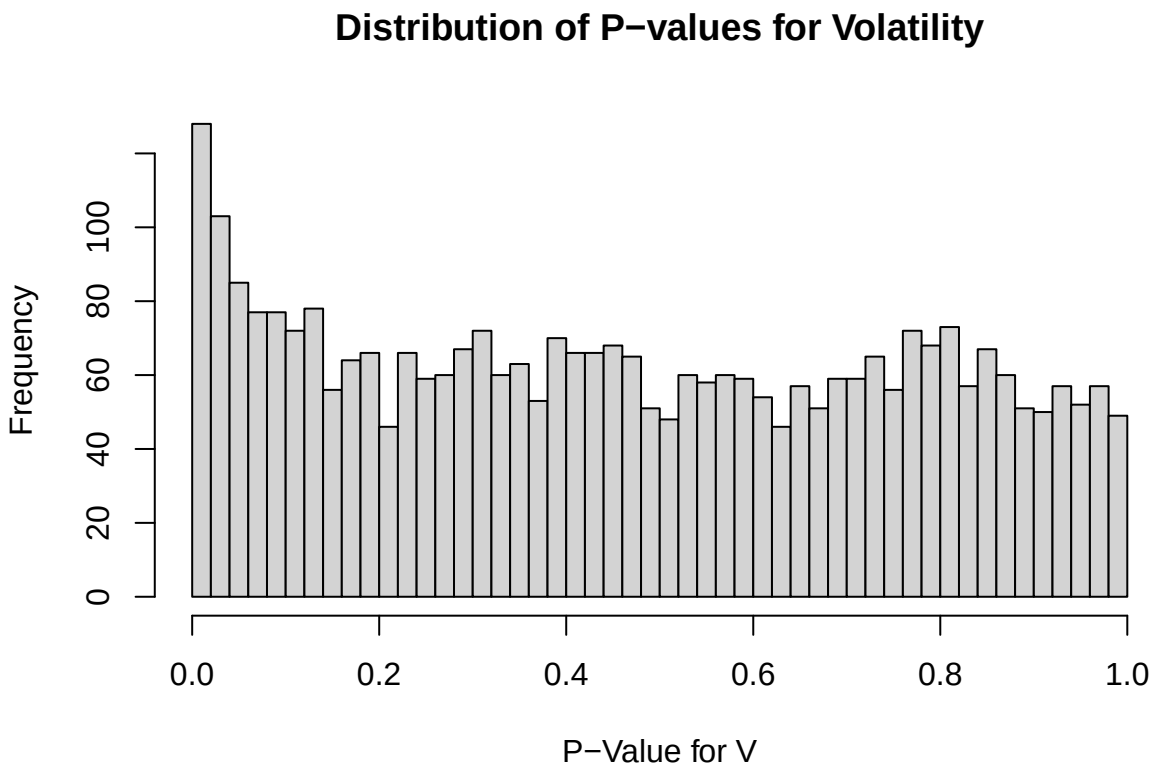
Question 1.1

```
# **** Analysis for Returns ****

Outcome<-R

clusterExport(cl,"Outcome")

# run the regressions in parallel

mrgpvals_R <- unlist(parLapply(cl,P,margreg))

hist(mrgpvals_R,
     main="Distribution of P-values for Returns",
     xlab="P-Value for R",
     ylab="Frequency",
     breaks = 70)
```




```
# **** Repeat for volatility

Outcome<-V

clusterExport(cl,"Outcome")

# run the regressions in parallel
```

```r
mrgpvals_V <- unlist(parLapply(cl,P,margreg))

hist(mrgpvals_V,
     main="Distribution of P-values for Volatility",
     xlab="P-Value for V",
     ylab="Frequency",
     breaks = 70)
```



For Returns: The histogram of p-values for returns (R) displays a uniform distribution, it appears there is no substantial signal that individual words from news headlines can predict stock prices. The lack of a concentration of low p-values suggests that the words do not have a strong and consistent statistical correlation with financial market movements. In an ideal scenario where many words significantly predict returns, we would expect to a higher frequency of small p-values, resulting in a right-skewed distribution with a peak near zero. Since we do not observe this pattern, and the proportion of near-zero pvalues is small, there may not be a strong signal in the words to predict returns.

For Volatility: The histogram of p-values for volatility (V) displays a nearly uniform distribution, it appears there is no substantial signal that individual words from news headlines can predict volatility. However, it's a little bit right-skewed with a spike in the frequency at zero p-value, which means that it has more small p-values compared to returns. It will leads to a slightly stronger signal in predicting volatility.

Question 1.2

```r
source("fdr.R")
q<-0.1

# Computing the alpha value for 10% FDR
alpha_R <- fdr_cut(mrgpvals_R, q=q)
alpha_V <- fdr_cut(mrgpvals_V, q=q)
```

```r
# Count the number of significant words at this level
significant_words_R <- sum(mrgpvals_R<=alpha_R)
significant_words_V <- sum(mrgpvals_V<=alpha_V)

cat("alpha value with 10% FDR for R: ", alpha_R, "\n")
```

```
## alpha value with 10% FDR for R:  1.026222e-05
```

```r
cat("alpha value with 10% FDR for V: ", alpha_V, "\n")
```

```
## alpha value with 10% FDR for V:  0.0003571024
```

```r
cat("Number of significant words for R: ", significant_words_R, "\n")
```

```
## Number of significant words for R:  1
```

```r
cat("Number of significant words for V: ", significant_words_V, "\n")
```

```
## Number of significant words for V:  12
```

Pros: 1. It can easily parse big datasets and do that in parallel. We have basically separated the words from one another and looked how well each one of them associates with the outcome. As a data screening method, FDR is very powerful and computationally feasible. 2. It strikes a balance between identifying significant results and controlling the rate of false positives, which is especially important when running many tests. FDR analysis controls the expected proportion of false positives among the rejected hypotheses, which thereby controls the risk of False discoveries in rejections. 3. FDR typically allows for more discoveries because it's less conservative. 4. FDR control can be more appropriate for exploratory analysis where we're willing to accept a higher risk of false discoveries in order not to miss potentially important findings.FDR is flexible in terms of application, since it can be applied to any model that involves Hypothesis Testing. 5. FDR provides a simple summary of the risk of False discoveries in rejections.

Cons: 1. Because of the bigrams (negations of language), the p values are likely not independent. This is a consequence of basically destroying sentence structure because we take each word individually. 2. Even at a 1% FDR, there's a chance that some of the words deemed significant might be false positives. 3. The original BH procedure assumes that tests are independent, which might not be true in all datasets, potentially leading to an incorrect number of false discoveries. 4. Understanding what FDR control actually means in practice can be complex; it controls the expected proportion of false positives, not the actual number.

Question 1.3

```r
# Identify the 20 smallest p-values for V
p_sorted <- sort(mrgpvals_V)

# Number of tests
n <- length(p_sorted)

# Identify the 20th smallest p-value
p_20 <- p_sorted[20]

# Calculate the FDR value for the 20th smallest p-value
fdr_value_20 <- n * p_20 / 20

expected_false_positives <- fdr_value_20 * 20

cat("FDR value for the 20th smallest p-value:", fdr_value_20, "\n")
```

```
## FDR value for the 20th smallest p-value: 0.1761584
```

```r
cat("Number of discoveries expected to be false: ",expected_false_positives, "\n")
```

```
## Number of discoveries expected to be false:  3.523167
```

With an FDR value of about 0.176 for the 20th smallest p-value, we'd expect roughly 3.5 of the 20 p-values we picked as significant to be false positives. This implies that the p-values aren't completely independent; if they were, the expected number of false discoveries would likely be lower. Also, they may represent words that are more commonly used in contexts related to significant market events, hence more likely to be correlated with market movements. Moreover, we use bigrams (negations of language) and other common word combinations to structure sentences. If we treat these tests as independent, then we are basically destroying sentence structure by taking each word individually.

Question 2.1

```r
# ***** LASSO analysis *****

# First analyze returns

library(gamlr)

lasso1<- gamlr(spm, y=R, lambda.min.ratio=1e-3)

# Get the summary of the LASSO model
lasso_summary1 <- summary(lasso1)
```

```
##
## gaussian gamlr with 3183 inputs and 100 segments.
```

```r
lambda_opt1 <- lasso1$lambda[which.min(lasso_summary1$aicc)]

cat("Optimal lambda (AICc):", lambda_opt1, "\n")
```

```
## Optimal lambda (AICc): 0.0006318906
```

```r
lambda1_index <- which(lasso1$lambda == lambda_opt1)

# Extract the coefficients corresponding to lambda_opt2
coefficients_opt1 <- lasso1$beta[, lambda1_index]

# Count the number of words selected as predictive of returns R
# (excluding the intercept)
num_words_selected <- sum(coefficients_opt1 != 0)
cat("Number of words selected:", num_words_selected, "\n")
```

```
## Number of words selected: 47
```

```r
 # Extract and print the names of the predictors (words) that have non-zero coefficients
 chosen_words <- names(coefficients_opt1)[coefficients_opt1 != 0]
 cat("Words chosen by LASSO:\n", paste(chosen_words, collapse = ", "), "\n")
```

```
## Words chosen by LASSO:
##  atom, bailout, bayer, begin, bom, canada, card, cctv, chart, congo, copyright, damn, did, elect, fa
```

```r
R2_R <- lasso_summary1$r2[lambda1_index]

cat("The in-sample R2 for Returns is: ",R2_R , "\n")
```

```
## The in-sample R2 for Returns is:  0.07200122
```

An in-sample R^2 value of 0.07200122 (about 7.2%) indicates that around 7.2% of the variability in the financial returns is explained by the LASSO model. This value is relatively low, suggesting that the selected words from the headlines do not have a strong predictive power for the financial returns in the sample. Hence, we conclude that there is no enough evidence.

Questions 2.2

```
# **** LASSO Analysis of volatility **** #
lasso2<- gamlr(spm, y=V, lambda.min.ratio=1e-3)

# Get the summary of the LASSO model
lasso_summary2 <- summary(lasso2)
```

```
##
## gaussian gamlr with 3183 inputs and 100 segments.
```

```
lambda_opt2 <- lasso2$lambda[which.min(lasso_summary2$aicc)]

cat("Optimal lambda (AICc):", lambda_opt2, "\n")
```

```
## Optimal lambda (AICc): 0.02601636
```

```
lambda2_index <- which(lasso2$lambda == lambda_opt2)

# Extract the coefficients corresponding to lambda_opt2
coefficients_opt2 <- lasso2$beta[, lambda2_index]

# Count the number of words selected as predictive of returns R
# (excluding the previous and intercept)
num_words_selected2 <- sum(coefficients_opt2[-1] != 0)
cat("Number of words selected:", num_words_selected2, "\n")
```

```
## Number of words selected: 134
```

```
# Extract and print the names of the predictors (words) that have non-zero coefficients
chosen_words2 <- names(coefficients_opt2)[coefficients_opt2 != 0]
cat("Words chosen by LASSO:\n", paste(chosen_words2, collapse = ", "), "\n")
```

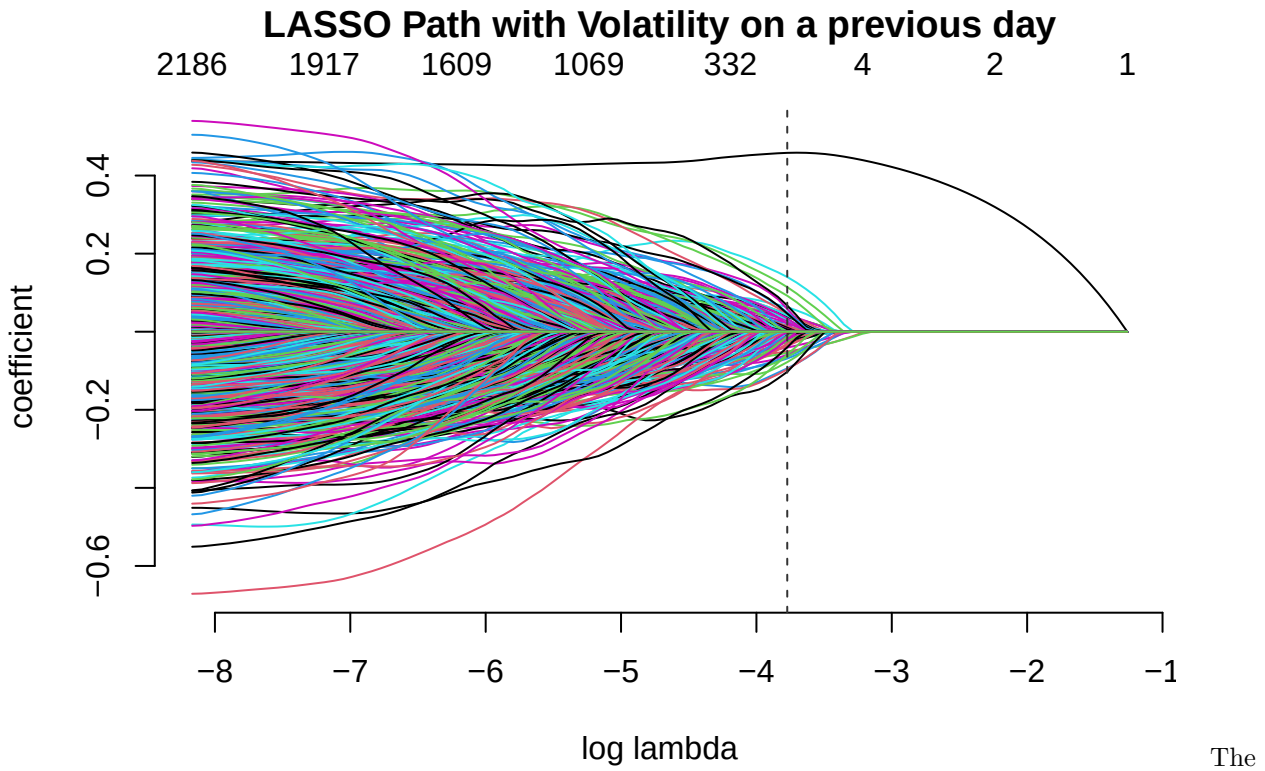
```
## Words chosen by LASSO:
##  abbott, access, almost, american, antiwar, arabia, around, arrest, august, australian, award, ayatol
```

```
R2_V <- lasso_summary2$r2[lambda2_index]
cat("The in-sample R2 for Volatility is: ",R2_V , "\n")
```

```
## The in-sample R2 for Volatility is:  0.1616116
```

```
# let's try to predict future volatility from past volatility, we will add one more predictor-> volatil

Previous<-log(dj[-1,3]-dj[-1,4]) # remove the last return

spm2<-cbind(Previous,spm) # add the previous return to the model matrix

colnames(spm2)[1]<-"previous" # the first column is the previous volatility

lasso3<- gamlr(spm2, y=V, lambda.min.ratio=1e-3)
```

```r
plot(lasso3)
title(main = "LASSO Path with Volatility on a previous day")
```



The LASSO path graph illustrates how the coefficients of predictors in a LASSO regression model vary with respect to log lambda, which is the parameter controlling the penalty's strength on the coefficients. As the value of lambda increases, the penalty intensifies, and more coefficients are compressed towards zero. Beginning from the left side of the graph, where the penalty is weakest (smaller lambda values), many predictors have non-zero coefficients. Moving rightward on the graph, as the penalty grows due to an increase in lambda, fewer predictors retain non-zero coefficients.

- The x-axis represents log lambda values. Moving from left to right on the graph, log lambda becomes less negative, which means lambda is increasing and the regularization penalty becomes stronger.
- The y-axis indicates the coefficient values for each predictor as a function of lambda. Each line traces the coefficient of one predictor across the range of lambda values. As we proceed rightward on the graph, the lines representing coefficients trend towards zero, demonstrating that more coefficients are being penalized towards insignificance as the penalty grows.
- The vertical dashed lines correspond to specific lambda values where the number of non-zero coefficients substantially changes, which can be seen in the count provided at the top of the graph.
- The numbers at the top signify the count of non-zero predictors remaining in the model at given lambda levels, which decreases as the penalty strengthens and we move to the right. The black line is likely the path of the intercept term, which is not penalized by LASSO and therefore does not shrink toward zero with increasing lambda.

```r
# Get the summary of the LASSO model
lasso_summary3 <- summary(lasso3)
```

```
##
## gaussian gamlr with 3184 inputs and 100 segments.
```

```r
lambda_opt3 <- lasso3$lambda[which.min(lasso_summary3$aicc)]
```

```r
cat("Optimal lambda (AICc):", lambda_opt3, "\n")
```

```
## Optimal lambda (AICc): 0.02300037
```

```r
lambda3_index <- which(lasso3$lambda == lambda_opt3)

# Extract the coefficients corresponding to lambda_opt2
coefficients_opt3 <- lasso3$beta[, lambda3_index]

# Count the number of words selected as predictive of returns R
# (excluding the intercept and Volatility)
num_words_selected3 <- sum(coefficients_opt3[-(1)] != 0)
cat("Number of words selected:", num_words_selected3, "\n")
```

```
## Number of words selected: 97
```

```r
R2_V2 <- lasso_summary3$r2[lambda3_index]

cat("The in-sample R2 for Volatility with Volatility on a previous day is: ",R2_V2 , "\n")
```

```
## The in-sample R2 for Volatility with Volatility on a previous day is:  0.331585
```

```r
# Find 10 strongest coefficients
effects <- coefficients_opt3
top_10_coef <- names(sort(abs(effects), decreasing = TRUE)[1:10])
print("Top 10 strongest coefficients:")
```

```
## [1] "Top 10 strongest coefficients:"
```

```r
print(top_10_coef)
```

```
##  [1] "previous"   "shed"       "fusion"     "unleash"    "shake"
##  [6] "joe"        "republican" "payout"     "direct"     "pioneer"
```

```r
coef_terr <- coefficients_opt3["terrorist"]
coef_vt1 <- coefficients_opt3["previous"]

cat("The coefficient for the word 'terrorist' is", coef_terr, "\n")
```

```
## The coefficient for the word 'terrorist' is 0.01783894
```

```r
cat("The coefficient for Vt-1 is", coef_vt1, "\n")
```

```
## The coefficient for Vt-1 is 0.4580944
```

The coefficient for the word 'terrorist' is 0.01783894. This means that, when holding all other variables constant, the volatility today increases by 0.01783894 if the word terrorist appears 1 more time in the headlines today.

The coefficient for Vt-1 is 0.4580944. This means that, holding all other variables constant, the volatility today increases by 0.4580944 unit if yesterday's volatility increases by 1 unit.

Question 2.3

```r
# Bootstrap to obtain s.e. of 1.s.e. chosen lambda

# We apply bootstrap to approximate
# the sampling distribution of lambda
# selected by AICc
```

```
# export the data to the clusters

Outcome<-V

clusterExport(cl,"spm2")
clusterExport(cl,"V")

# run 100 bootstrap resample fits

boot_function <- function(ib){

    require(gamlr)

    fit <- gamlr(spm2[ib,],y=V[ib], lambda.min.ratio=1e-3)

    fit$lambda[which.min(AICc(fit))]
}

boots <- 100

n <- nrow(spm2)

resamp <- as.data.frame(
            matrix(sample(1:n,boots*n,replace=TRUE),
            ncol=boots))

lambda_samp <- unlist(parLapply(cl,resamp,boot_function))


set.seed(41201)
# Sequential bootstrap resampling using a for loop
for (i in 1:boots) {
   # Sample indices with replacement
   ib <- sample(1:n, n, replace = TRUE)
   # Call the bootstrap function and store the result
   lambda_samp[i] <- boot_function(ib)
 }
```

```
# Calculate the standard error of the lambda estimates
lambda_se <- sd(lambda_samp)

# Number of bootstrap samples
 n <- length(lambda_samp)

 # Calculate the mean of the lambda samples
 mean_lambda <- mean(lambda_samp)

# Find the t-distribution critical value for 95% confidence interval # Degrees of freedom = n - 1
t_critical <- qt(c(0.025, 0.975), df = n - 1)

 # Calculate the 95% confidence interval for lambda using the t-distribution
 ci_lambda <- mean_lambda + t_critical * lambda_se
 cat("95% Confidence Interval for lambda:", ci_lambda, "\n")
```

```
## 95% Confidence Interval for lambda: 0.0003386807 0.001397175
```

Question 3.1

```
# High-dimensional Covariate Adjustment

d <- Previous # this is the treatment

# marginal effect of past on present volatility

summary(glm(V~d))
```

```
##
## Call:
## glm(formula = V ~ d)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.42168    0.09620   25.17   <2e-16 ***
## d            0.51195    0.01926   26.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2262754)
##
##     Null deviance: 609.23  on 1987  degrees of freedom
## Residual deviance: 449.38  on 1986  degrees of freedom
## AIC: 2691.5
##
## Number of Fisher Scoring iterations: 2
```

The regression analysis shows that yesterday's volatility (Vt-1) has a significant positive effect on today's volatility (V), with a coefficient of 0.51195 and a standard error of 0.01926. The t-value is substantially large at 26.58, and the p-value is less than 2e-16, indicating a highly significant effect of d on V. This indicates that higher volatility one day is associated with an increase in volatility the next day. The strong significance suggests Vt-1 is an important predictor for V. This high magnitude of the coefficient, together with its statistical significance, implies a strong correlation.

```
# we want to isolate the effect of d from external influences. We saw that words can explain some of th

# Stage 1 LASSO: fit a model for d on x

treat <- gamlr(spm,d,lambda.min.ratio=1e-4)

## Predict d based on the model "treat" and x
dhat <- predict(treat, spm, type="response")

## IS R^2
R2 <- cor(drop(dhat),d)^2

cat("The In-Sample R-squared value is", R2,".\n")
```

```
## The In-Sample R-squared value is 0.3648263 .
```

The model predict d from x, and we get dhat, which represents the overlapped between treatment(d) and x. The in-sample R-squared value of 0.3648263 indicates a moderate small degree of correlation between

treatment(d) and dhat, so the model does not fits the data very well. The treatment effect (d) can be partially predicted from the variables included in x. The remaining d part for treatment effect is relatively large (correlated part between x and treatment(d) is moderately small). IS R2 the moderate level of in-sample R-squared also means x and treatment(d) is overlapped a little. The fit between x and treatment(d) is moderately tight. We do not expect a large degree of confounding. The degree of confounding is moderately small because the overlapped part between d and x is relatively small. There is some information in d independent of x because there is only 36.48% of the variability in the d is explained by the model, about 63.5% (= 1 - 36.5%) of the variance in d is not explained by x.

Question 3.2

```
# Stage 2 LASSO: fit a model for V using d, dhat and x

causal <- gamlr(cbind(d,dhat,spm),V,free=2,lmr=1e-4)

## Coefficient of treatment
causald <- coef(causal)["d",]
cat("The effect of d on Volatility is",causald, ".\n")
```

```
## The effect of d on Volatility is 0.3602978 .
```

The coefficient of 0.3602978 for the treatment variable `d` (which represents the previous day's volatility, `Vt-1`) suggests that according to the AICc, `Vt-1` has a causal effect on today's volatility. The positive coefficient indicates that higher values of `Vt-1` are associated with an increase in the expected value of today's volatility `Vt`, with all other variables held constant. Specifically, a 1 unit increase in previous day's volatility is associated with a 0.36 unit increase in today's volatility. This finding points towards a significant influence of past volatility on current market behavior, highlighting the persistence of volatility over time.

```
## naive lasso

naive <- gamlr(cbind(d,spm),V)

## Coefficient for 'd
naived <- coef(naive)["d",]

cat("The effect of d on volatility from a straight (naive) lasso is",naived, "\n")
```

```
## The effect of d on volatility from a straight (naive) lasso is 0.4574218
```

The causal (double) LASSO model suggests that Vt-1 (yesterday's volatility) has a significant effect on today's volatility, with a coefficient of 0.3602978. When compared to the effect obtained from the naive LASSO, which is 0.4574218, the causal LASSO's coefficient is smaller. This difference indicates that the naive LASSO may have overestimated the effect of Vt-1 due to not accounting for the potential confounding effect of the control variables included in the causal LASSO. After removing the confounding effect of headlines words, the causal effect becomes smaller.

Question 3.3

We cannot conclusively assert causality based solely on the information provided because:

There may be other confounding variables not included in the specified model (x), which could lead to an inaccurate estimation of the treatment effect. The statistical significance of the treatment effect has not been established. This can be assessed by examining p-values or constructing confidence intervals to determine if they include zero. However, while statistical significance is necessary, it alone does not confirm causality. The data set used may not be comprehensive enough, necessitating validation of the findings across additional data sets. The assumptions underlying linear regression must be rigorously verified. This

includes checking for linearity, the absence of perfect multicollinearity, homoscedasticity, and the normality of the residuals. In summary, relying solely on the double Lasso model does not provide sufficient basis for claiming causality. Although double Lasso can help mitigate some issues in identifying causal relationships by adjusting for potential confounders, establishing causality requires thorough theoretical and empirical validation, significance testing, and scrutiny of the model's assumptions and robustness.

Further Analysis 1

I want to investigate whether the previous day's return has causal effect on today's return. First of all, we want to isolate the effect of d from external influences. We already know that headlines words have little predicting power for return, so we probably want to investigate whether volatility has prediciting power for return.

```
summary(glm(R~V))
```

```
##
## Call:
## glm(formula = R ~ V)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.009210   0.002552   3.609 0.000315 ***
## V           -0.001797   0.000511  -3.516 0.000448 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.0001591109)
##
##     Null deviance: 0.31796  on 1987  degrees of freedom
## Residual deviance: 0.31599  on 1986  degrees of freedom
## AIC: -11741
##
## Number of Fisher Scoring iterations: 2
```

The output from the regression model summary indicates that the coefficient for V (today's volatility) is significant in predicting R (today's return). The coefficient value is negative (-0.001797), and the p-value associated with this coefficient is very small (0.000448), which is below any conventional significance level (like 0.05, 0.01, etc.). This statistically significant negative coefficient suggests that as volatility increases, returns tend to decrease, which could reflect a risk-averse sentiment in the market—investors may expect higher returns to compensate for higher risk (volatility).

We want to isolate the effect of yesterday's return on today's return, controlling for yesterday's volatility. We should explore a marginal regression (just a regression of Rt on Rt−1) to see if there is any correlation.

```
R_prev <- R[-1]  # generate yesterday's return

R_r <- R[-length(R)] # remove the last row to match the length

d_r <- R_prev # this is the treatment

# marginal effect of past on present return

summary(glm(R_r~d_r))
```

```
##
## Call:
```

```
## glm(formula = R_r ~ d_r)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0003221  0.0002825   1.140    0.254
## d_r         -0.1033322  0.0223244  -4.629 3.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.0001584643)
##
##     Null deviance: 0.31795  on 1986  degrees of freedom
## Residual deviance: 0.31455  on 1985  degrees of freedom
## AIC: -11743
##
## Number of Fisher Scoring iterations: 2
```

The regression analysis shows that yesterday's return (Rt-1) has a significant negative effect on today's return (Rt), with a coefficient of -0.1033322 and a standard error of 0.0223244. The t-value is large at -4.629, and the p-value is less than 3.92e-06, indicating a highly significant effect of d on V. This indicates that higher return one day is associated with an decrease in return the next day. The strong significance suggests Rt-1 is an important predictor for R. This high magnitude of the coefficient, together with its statistical significance, implies a strong correlation.

Now, we predict d from x (yesterday's volatility)

```
# remove yesterday vol's last row to match the length
Previous_r <- Previous[-length(Previous)]

# Convert the vector Previous_r to a matrix with one column
spm3 <- matrix(Previous_r, ncol = 1)

# Stage 1 LASSO: fit a model for d on x

model_r <- gamlr(spm3,d_r,lambda.min.ratio=1e-4)

dhat_r <- predict(model_r, newdata = spm3, type = "response")

R2_r <- cor(drop(dhat_r), d_r)^2

cat("The In-Sample R-squared value is", R2_r,".\n")
```

```
## The In-Sample R-squared value is 0.006183657 .
```

The R^2 = 0.006183657 indicates that approximated 0.62% of the variance in d is explained by x. This indicates a not enough degree of correlation between treatment(d) and x, which also means that the remaining d part for treatment effect that is isolated from x is still large. Hence, x and treatment(d) do not overlap, we do not expect a large degree of confounding. Moreover, the R^2 also implies that about 99.38% (= 1 - 0.62%) of the variance in d is not explained by x, which suggests the large portion of existance of information in d independent of x.

```
# Stage 2 LASSO: fit a model for R using d, dhat and x
causal_r <- gamlr(cbind(d_r,dhat_r,spm3), R_r, family='gaussian',free=2) ## Coefficient of treatment
```

```
## 'as(<dgeMatrix>, "dgCMatrix")' is deprecated.
## Use 'as(., "CsparseMatrix")' instead.
## See help("Deprecated") and help("Matrix-deprecated").
```

```
causald_r <- coef(causal_r)["d_r",]
cat("The effect of d on today's return is",causald_r, ".\n")
```

```
## The effect of d on today's return is -0.103477 .
```

```
naive_r <- gamlr(cbind(d_r,spm3), R_r , family='gaussian')

## Coefficient for 'd'

naived_r <- coef(naive_r)["d_r",]

cat("The effect of d on today's return from a naive lasso is", naived_r, "\n")
```

```
## The effect of d on today's return from a naive lasso is -0.08935086
```

Comapre causal (double) LASSO with the naive LASSO, they generate similar treatment coefficient (-0.10 and -0.09). This suggests that the relationship between d and x is not significantly confounded, since a similar coefficient is generated after removing the effect of x that are correlated with d.

The negative coefficient of -0.103477 implies that for every one-unit increase in the previous day's return, the current day's return is expected to decrease by 0.103477 units on average. This model suggests a mean-reversion effect: if returns were high on the previous day, they tend to be lower the next day, or vice versa.

Although the model suggests a significant relationship, it does not necessarily imply causation. Other factors not accounted for in the model could be driving the relationship. This model appears to be quite simple, only including the previous day's return as a predictor. In reality, return dynamics are influenced by a multitude of factors. A more comprehensive model might include additional variables that could affect returns, such as market sentiment, economic news, or technical indicators.