Homework 5 Assignment

Charlie Marcou, Carrie Mecca, Jasmine Zhang, and Jessie Bustin

We will use congress109 data in package textir. It counts for 1,000 phrases used by each of 529 members of the 109th US congress.

```
library(textir) # to get the data

library(maptpx) # for the topics function

data(congress109) # load the data
```

The counts are in congress109counts. We also have congress109Ideology, a data.frame containing some information about each speaker. These includes some partisan metrics”

party (Republican, Democrat, or Independent) repshare: share of constituents voting for Bush in 2004. Common Scores [cs1,cs2]: basically, the first two principal components of roll-call votes (next week!).

1. Fit K-means to speech text for K in 5,10,15,20,25. Use an IC to choose the K and interpret the selected model.

Based on both AIC and BIC, the optimal K is 5. This means that the simplest model was chosen. However, the R^2 for the model is low, .0311, indicating that only approximately 3.1% of the deviance in x is being explained.

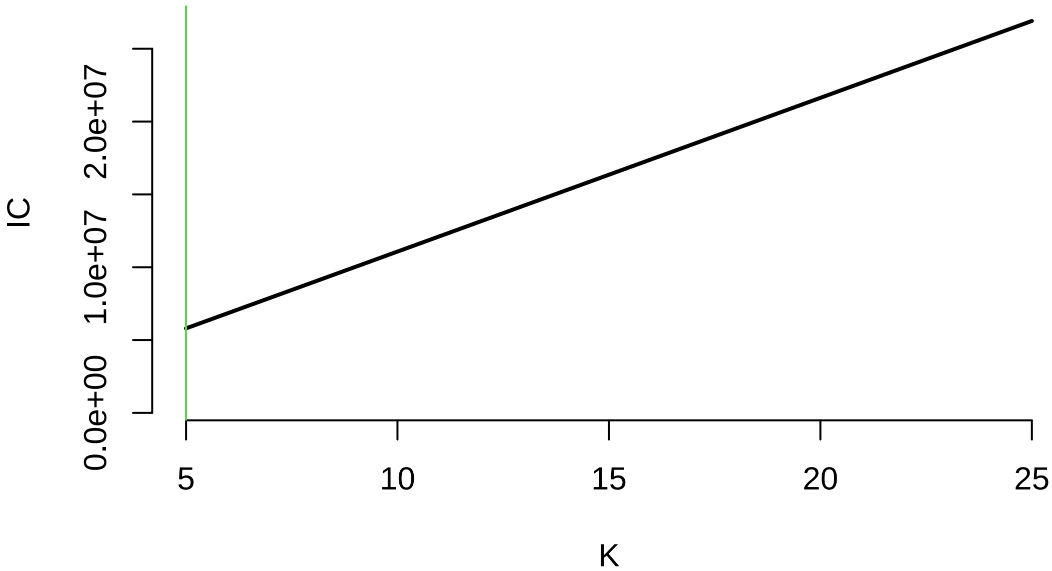
```
set.seed(100)
xcongress <- scale(as.matrix( congress109Counts/rowSums(congress109Counts) ))

k_list <-5*(1:5)
kfit <- lapply(k_list, function(k) kmeans(xcongress,k))

source("kIC.R") ## utility script
kaicc <- sapply(kfit,kIC)
kbic <- sapply(kfit,kIC,"B")

## plot 'em
plot(y=kaicc, x=k_list, xlab="K", ylab="IC",
    ylim=range(c(kaicc,kbic)),
    bty="n", type="l", lwd=2)

abline(v=k_list[which.min(kaicc)], col=4)
abline(v=k_list[which.min(kbic)],col=3)
```




```r
paste0('AIC Optimal k: ', k_list[which.min(kaicc)])
```

```
## [1] "AIC Optimal k: 5"
```

```r
paste0('BIC Optimal k: ',k_list[which.min(kbic)])
```

```
## [1] "BIC Optimal k: 5"
```

```r
##Get r^2
k_index=1
paste0('R^2: ',1 - sum(kfit[[k_index]]$tot.withinss)/kfit[[k_index]]$totss)
```

```
## [1] "R^2: 0.0311163741736133"
```

2. Fit a topic model for the speech counts. Use Bayes factors to choose the number of topics, and interpret your chosen model.

Using Bayes factors we select the topic model with 10 topics. We can visualize these topics by examining word clouds of the most probable words within each topic.

```r
## topic modelling.  Treat counts as actual counts!
## i.e., model them with a multinomial
## we'll use the topics function in maptpx (there are other options out there)

## you need to convert from a Matrix to a `slam' simple_triplet_matrix
## luckily, this is easy.
x <- as.simple_triplet_matrix(congress109Counts)

# to fit, just give it the counts, number of `topics' K, and any other args
tpc <- topics(x,K=10)
```

```
##
## Estimating on a 529 document collection.
## Fitting the 10 topic model.
## log posterior increase: 6696.5, 2146.5, 798.1, 379.5, 132.9, 92.5, 63.8, 41, 11.1, 76.6, 95.7, 12, 1
```

```r
dim(tpc$theta)
```

```
## [1] 1000   10
```

```r
colSums(tpc$theta)
```

```
##  1  2  3  4  5  6  7  8  9 10
##  1  1  1  1  1  1  1  1  1  1
```

```r
dim(tpc$omega)
```

```
## [1] 529  10
```

```r
#rowSums(tpc$omega)

## choosing the number of topics
## If you supply a vector of topic sizes, it uses a Bayes factor to choose
## (BF is like exp(-BIC), so you choose the bigggest BF)
## the algo stops if BF drops twice in a row

tpcs <- topics(x,K=5*(1:5), verb=10) # it chooses 10 topics
```

```
##
## Estimating on a 529 document collection.
## Fit and Bayes Factor Estimation for K = 5 ... 25
## log posterior increase: 5662.2, 4665.3, 3617.2, 2793.6, 2360.1, 2100.1, 1936.1, 1795.8, 1651.8, 1524
## log BF( 5 ) = 59609.03 [  314 steps, disp = 3.71  ]
## log posterior increase: 4937.1, 3253.8, 2486.5, 2287, 1304.4, 798.2, 555.4, 438.6, 356.7, 308.5, 278
## log BF( 10 ) = 76679.4 [  303 steps, disp = 2.84  ]
## log posterior increase: 3175.9, 1771.2, 1367.6, 1028.6, 794.4, 668.6, 552.4, 399.6, 304.5, 247.2, 21
## log BF( 15 ) = 75752.52 [  361 steps, disp = 2.45  ]
## log posterior increase: 2015.7, 1006, 692.9, 534.2, 428.2, 351.7, 281.5, 238.9, 218.2, 205.6, 186.7,
## log BF( 20 ) = 66374.08 [  315 steps, disp = 2.2  ]
```

```r
## interpretation
# summary prints the top `n' words for each topic,
# under ordering by `topic over aggregate' lift:
#the topic word prob over marginal word prob.

summary(tpcs, n=10)
```

```
##
## Top 10 phrases by topic-over-null term lift (and usage %):
##
## [1] 'national.heritage.corridor', 'ryan.white.care', 'violence.sexual.assault', 'white.care.act', 'd
## [2] 'southeast.texa', 'commonly.prescribed.drug', 'ready.mixed.concrete', 'million.illegal.alien', '
## [3] 'near.retirement.age', 'increase.taxe', 'personal.retirement.account', 'medic.liability.reform',
## [4] 'winning.war.iraq', 'near.earth.object', 'troop.bring.home', 'bless.america', 'nunn.lugar.progra
## [5] 'united.airline.employe', 'record.budget.deficit', 'student.loan.cut', 'private.account', 'secur
## [6] 'republic.cypru', 'hate.crime.legislation', 'change.heart.mind', 'driver.education', 'va.health.
## [7] 'hearing.scheduled', 'witness.testify', 'circuit.judge', 'business.meeting', 'judge.alberto.gonz
## [8] 'able.buy.gun', 'western.energy.crisi', 'credit.card.industry', 'caliber.sniper.rifle', 'wild.bi
## [9] 'pluripotent.stem.cel', 'low.cost.reliable', 'national.ad.campaign', 'cel.stem.cel', 'regional.t
## [10] 'american.fre.trade', 'central.american.fre', 'north.american.fre', 'financial.accounting.stand
##
## Log Bayes factor and estimated dispersion, by number of topics:
##
##              5       10       15       20
## logBF 59609.03 76679.40 75752.52 66374.08
## Disp      3.71     2.84     2.45     2.20
```

```
##
## Selected the K = 10 topic model
```

```r
# this will promote rare words that with high in-topic prob
# alternatively, you can look at words ordered by simple in-topic prob
## the topic-term probability matrix is called 'theta',
## and each column is a topic
## we can use these to rank terms by probability within topics

#rownames(tpcs$theta)[order(tpcs$theta[,1], decreasing=TRUE)[1:10]]

#rownames(tpcs$theta)[order(tpcs$theta[,2], decreasing=TRUE)[1:10]]

library(wordcloud)

## we'll size the word proportional to its in-topic probability
## and only show those with > 0.004 omega
## (it will still likely warn that it couldn't fit everything)

par(mfrow=c(1,2))

wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,1], min.freq=0.004, col="maroon")

wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,2], min.freq=0.004, col="navy")
```


```r
wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,3], min.freq=0.004, col="black")

wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,4], min.freq=0.004, col="green")
```

```
wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,5], min.freq=0.004, col="blue")

wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,6], min.freq=0.004, col="navy")
```

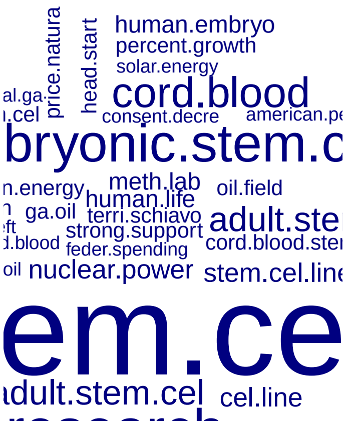


```
wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,7], min.freq=0.004, col="navy")

wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,8], min.freq=0.004, col="navy")
```

```r
wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,9], min.freq=0.004, col="navy")

wordcloud(row.names(tpcs$theta),
    freq=tpcs$theta[,10], min.freq=0.004, col="navy")
```



3. Connect the unsupervised clusters to partisanship.

I tabulate party membership by K-means cluster. Are there any non-partisan topics? I fit topic regressions for each of party and repshare. Compare to regression onto phrase percentages: x<-100*congress109Counts/rowSums(congress109Counts)

We tabulated party membership by K-means cluster. We can see based on the table, clusters 2 and 4 are partisan, containing far more members of one party than the other. Clusters 1,3, and 5 seem to be more bi-partisan and contain a more diverse group of party members.

We then fit topic regressions for party and repshare. We can see that the MSE for both topic regressions and repshare is lower for the topic models compared to models fit with phrase percentages. The topic models have a better performance.

```r
##tabulating topic by party
t<-table(congress109Ideology$party,kfit[[k_index]]$cluster)
t
```

```
##
##       1   2   3   4   5
##   D  31   1   1 128  81
##   I   0   0   0   1   1
##   R  13  14   2   0 256
```

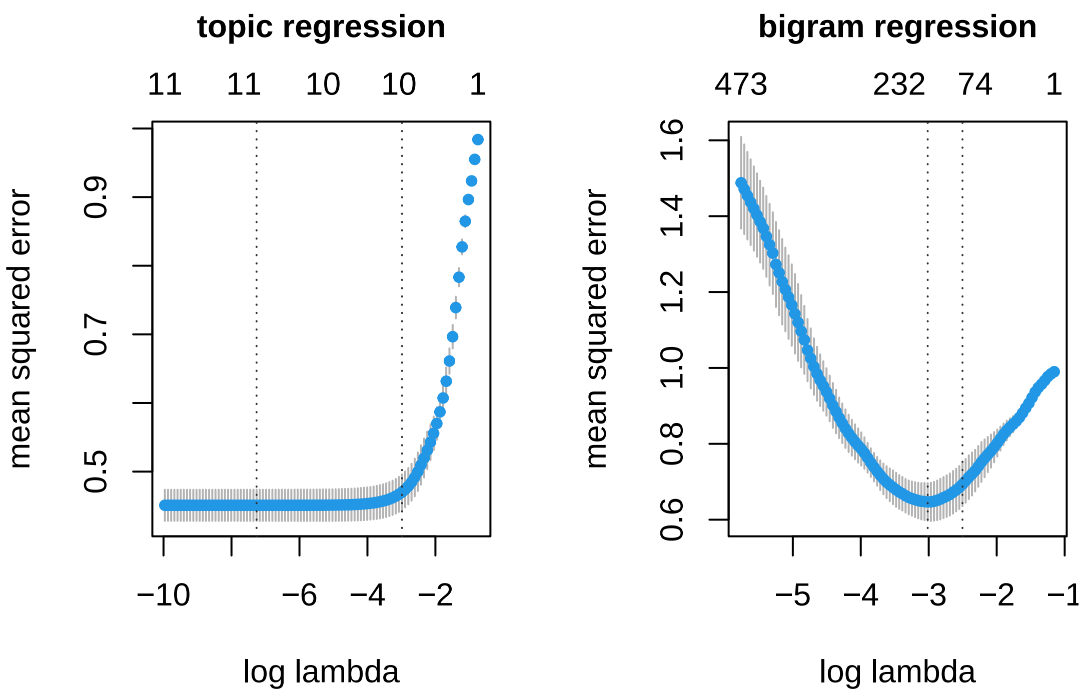
```r
##topic 2,6,7,9,23,25 seem to be highly partisan
##topic 1,5,8,17,22 seem to be bi-partisan

#regress party against topic
party <- congress109Ideology[,"party"]
regtopics.cv <- cv.gamlr(tpcs$omega, party,lambda.min.ratio=10^{-4})

## give it the word %s as inputs
x <- 100*congress109Counts/rowSums(congress109Counts)
regwords.cv <- cv.gamlr(x, party)

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

```r
plot(regtopics.cv)
mtext("topic regression", font=2, line=2)
plot(regwords.cv)
mtext("bigram regression", font=2, line=2)
```




```r
# min OOS MSE
min(regtopics.cv$cvm)
```

```
## [1] 0.4508901
```

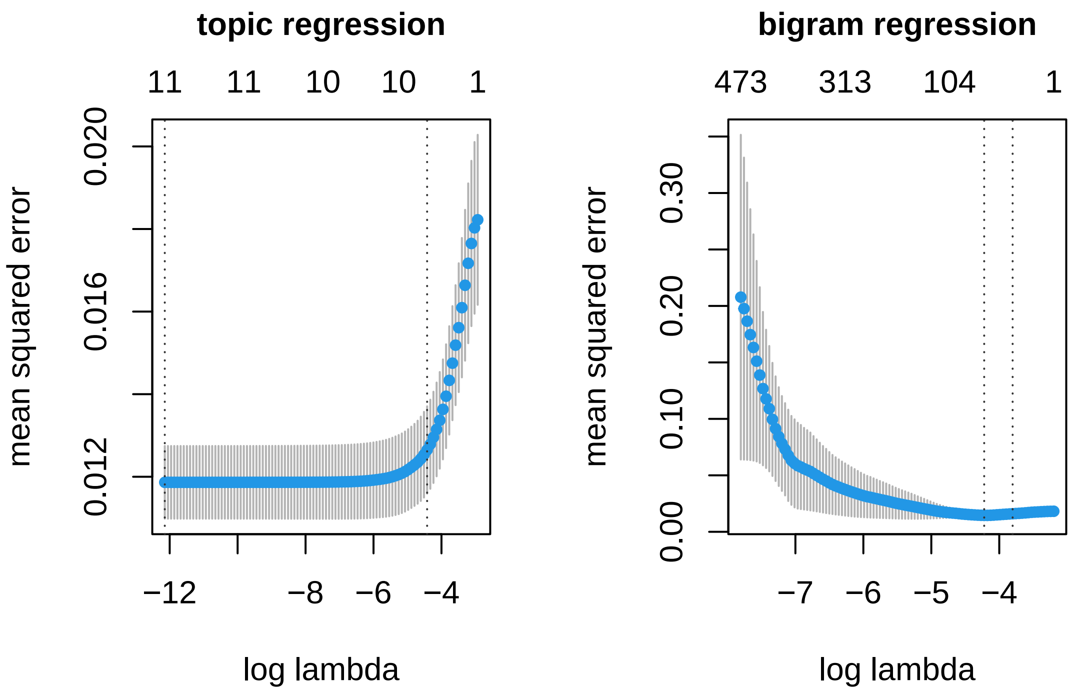
```r
min(regwords.cv$cvm)
```

```
## [1] 0.6467938
```

```r
#regress repshare against topic
repshare <- congress109Ideology[,"repshare"]
regtopics.cv.2 <- cv.gamlr(tpcs$omega, repshare,lambda.min.ratio=10^{-4})

## give it the word %s as inputs
regwords.cv.2 <- cv.gamlr(x, repshare)

par(mfrow=c(1,2))
plot(regtopics.cv.2)
mtext("topic regression", font=2, line=2)
plot(regwords.cv.2)
mtext("bigram regression", font=2, line=2)
```




```
# min OOS MSE
min(regtopics.cv.2$cvm)
```

```
## [1] 0.01186488
```

```
min(regwords.cv.2$cvm)
```

```
## [1] 0.01453615
```