Final Project

Charlie Zhan_MSQF

4/29/2021

Finance Project: Pricing performance of factors

```
rm(list = ls())
library(cbw)
```

```
## Loading required package: coda
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

```
## Loading required package: roxygen2
```

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

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

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

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

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

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

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

```
## Loading required package: ucminf
```

```
## Loading required package: numDeriv
```

```
## Loading required package: ggplot2
```

```
## Loading required package: RcppArmadillo
```

```
##
## Attaching package: 'cbw'
```

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

1. Select 10 stocks that you are interested in. Find their yahoo symbols. This website can help you find the symbols that yahoo is using http://investexcel.net/all-yahoo-financestock-tickers/. Remember to double check the symbols at yahoo finance.

```
symbols=c('AAPL','BAC','AMZN','T','GOOG','IBM','F','UBS','ACN','EA')
symnames=c('apple','boa','amazon','att','google','ibm','ford','ubs','accentur
e','ea')
```

2. Download monthly premium data for each stock from Jan 2005 to Dec 2018 using the cbw getfinmdat() function. Remember this requires that all 10 stocks you select in step 1 should be available for this time frame.

```
prmdf = getfinmdat(symbols=symbols,symnames=symnames,
                   from = "2004-12-31",
                   to = "2018-12-31")

## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## Warning: ^IRX contains missing values. Some functions will not work if obj
ects
## contain missing values in the middle of the series. Consider using na.omit
(),
## na.approx(), na.fill(), etc to remove or replace them.

## Warning in to.period(datxts, indexAt = "endof", period = "months"): missin
g
## values removed from data
```

3. Load the package czzg and use the data(factor12) as given to find the best factor collection by the Chib, Zeng and Zhao (2020) method. Use a student-t distribution for the errors and let nu = 5 in the model scan.

```
library(czzg)

## Loading required package: future.apply

## Loading required package: future

##
## Attaching package: 'czzg'

## The following objects are masked from 'package:cbw':
##
##     dmvn1, dmvn2, gaussreg_, makesubsets, MCMCregressg, pdfavg, rmvn0,
##     summarymcmc, xpnd
```

```
plan(multisession)
data("factor12")
datdf = factor12
scanls  = CZZtscan(data = datdf,nu = 5)

## starting Chib, Zeng and Zhao (2020) model scan with student-t errors ...
## there are 4095 models in the model space

scandf = scanls$scandf
scandford = scandf[order(scandf$logmarg,decreasing = T),]
Mst = which.max(scandf$logmarg);
scandf[Mst,];

##     Mkt SMB HML RMW CMA MOM IA ROE PEAD FIN MGMT PERF  logmarg
## 905   1   1   0   0   0   0  0   1    1   0    1    0 15089.02

# the best model is Mkt + SMB + ROE + PEAD + MGMT
```

4. Combine the data in factor12 with the data on the 10 stock premiums (this means that at this point you will remove all the rows in factor12 before January 2005).

```
datdfn = datdf[373:540,]
prmdfn = prmdf[,1:10]
prmfactordf = cbind(datdfn,prmdfn)
```

5. Use the factors from the best model to see how many of the 10 stocks can be priced (for each stock you need to fit two models - one with an intercept and one without, as described in the which factors note).

```
xnames = c("Mkt","SMB","ROE","PEAD","MGMT")
ynames = names(prmdfn)
czzfrm = "~Mkt + SMB + ROE + PEAD + MGMT"
J = length(ynames)
marglik1 = rep(0,J)
marglik0 = rep(0,J)
for (j in 1:J) {
  ynamesj = ynames[j]
  frmj = as.formula(paste(ynamesj,czzfrm,sep = ""))
  frmj0 = as.formula(paste(ynamesj,czzfrm,"-1",sep = ""))
  theta1m = MCMCregresst(modelfrm = frmj,
                         data = prmfactordf,
                         nu=5);
  theta0m = MCMCregresst(modelfrm = frmj0,
                         data = prmfactordf,
                         nu=5);
  marglik1[j] = logmarglik(theta1m);
  marglik0[j] = logmarglik(theta0m);
}
diff = marglik1 - marglik0;
ynames[diff > 1.15]

## [1] "prmapple"
```

```
#9 of the 10 stocks can be priced, except Apple
```

Marketing Project: Brand complements and substitutes

```
data("tuna")
dftuna=tuna
```

1. Load the tuna data set. There are seven brands in the data set. For each brand, estimate separate independent student-t models where logsales for each product is regressed on an intercept, the product's log price and display activity. Use the default training sample prior and use log-marginal likelihoods to find the appropriate-degrees of freedom of the student-t distribution on the grid seq(from = 3,to = 6,length.out = 11).

```
nug = seq(from = 3,to = 6,length.out = 11)
J = 7
xnames = c( "~ LPRICE", "+ NSALE")
ynames = "MOVE"
marglik = rep(0,J)
for (j in 1:J) {
  ynamesj = paste(ynames,j,sep = "")
  xnamesj= paste(xnames,j,sep = "",collapse = "")
  modelj = as.formula(paste(ynamesj,xnamesj,sep = ""))
  outlsj= mapply("MCMCregresst",
                 nu = nug,
                 MoreArgs = list(modelfrm = modelj, data = dftuna),
                 SIMPLIFY = FALSE)
  marglikj = logmarglik(outlsj);
  marglik[j]= marglikj
  A = cbind(nug,t(marglikj))
  indj = which.max(A[,2]); # which model has the largest marg lik
  colnames(A) = c("nu","logmarg")
  rnames = paste("brand",j,sep = "",collapse = "")
  rnames = rep(rnames,times = length(nug))
  rownames(A) = rnames
  print(A[indj,,drop = F])
}
```

```
## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##        nu   logmarg
## brand1  3 -3146.351

## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##        nu  logmarg
## brand2  3 -3267.65

## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##         nu   logmarg
## brand3 3.6 -2331.542
```

```
## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##        nu   logmarg
## brand4  3 -3035.325

## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##        nu   logmarg
## brand5  3 -2273.478

## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##        nu   logmarg
## brand6  6 -2108.064

## Warning in marglik[j] <- marglikj: 被替换的项目不是替换值长度的倍数

##        nu  logmarg
## brand7  3 -2892.07
```

2. Now estimate a SURE student-t model for the seven brands. Again use marginal likelihoods to find the appropriate degrees of freedom on the grid seq(from = 3,to =6,length.out = 11).

```
suremodels=list()
for (j in 1:J) {
  ynamesj = paste(ynames,j,sep = "")
  xnamesj= paste(xnames,j,sep = "",collapse = "")
  suremodels[[j]] = as.formula(paste(ynamesj,xnamesj,sep = ""))
}
sureoutls= mapply("MCMCsuret",
                  nu = nug,
                  MoreArgs = list(modelfrm = suremodels, data = dftuna),
                  SIMPLIFY = FALSE)
suremarglik = logmarglik(sureoutls);
B = cbind(nug,t(suremarglik))
colnames(B) = c("nu","logmarg")
sureind = which.max(B[,2]); # which model has the largest marg lik
surethetatm = sureoutls[[sureind]]
print(B[sureind,,drop = F])

##      nu   logmarg
## [1,]  3 -18852.71
```

3. Do you find that the SURE-t model is an improvement over the independent t-models?

```
sum(marglik)

## [1] -19090.64

logmarglik(surethetatm)

## [1] -18852.71
```

```
#Yes, because SURE-t model has greater marginal likelihood
```

4. From the best fitting SURE-t model, what brands appear to be complements and which appear to be substitutes?

```
summarycorr(surethetatm)

##               MOVE1       MOVE2       MOVE3      MOVE4       MOVE5       MO
VE6
## MOVE1  1.000000000  0.03509488 -0.01296663 0.08697843  0.10229376  0.05560
751
## MOVE2  0.035094879  1.00000000  0.05261763 0.08280791  0.15976886  0.07262
414
## MOVE3 -0.012966629  0.05261763  1.00000000 0.11989816  0.03783104  0.48009
873
## MOVE4  0.086978427  0.08280791  0.11989816 1.00000000  0.11227895  0.04789
851
## MOVE5  0.102293765  0.15976886  0.03783104 0.11227895  1.00000000 -0.04761
021
## MOVE6  0.055607508  0.07262414  0.48009873 0.04789851 -0.04761021  1.00000
000
## MOVE7  0.001440399 -0.09160054 -0.04441525 0.02376912  0.11370513 -0.14980
665
##               MOVE7      MOVE1      MOVE2      MOVE3      MOVE4      MOVE5
## MOVE1  0.001440399 1.00000000 0.06188155 0.06224680 0.06580468 0.06259944
## MOVE2 -0.091600536 0.06188155 1.00000000 0.06110246 0.06064420 0.06094761
## MOVE3 -0.044415251 0.06224680 0.06110246 1.00000000 0.06058495 0.06654685
## MOVE4  0.023769124 0.06580468 0.06064420 0.06058495 1.00000000 0.06166485
## MOVE5  0.113705132 0.06259944 0.06094761 0.06654685 0.06166485 1.00000000
## MOVE6 -0.149806652 0.06498928 0.06197251 0.05180704 0.06351142 0.06675131
## MOVE7  1.000000000 0.06288521 0.06023723 0.06298310 0.06113338 0.06225494
##            MOVE6      MOVE7
## MOVE1 0.06498928 0.06288521
## MOVE2 0.06197251 0.06023723
## MOVE3 0.05180704 0.06298310
## MOVE4 0.06351142 0.06113338
## MOVE5 0.06675131 0.06225494
## MOVE6 1.00000000 0.06396741
## MOVE7 0.06396741 1.00000000

#complements have positive correlation coefficient, while substitutes have ne
gative correlation coefficient
#brand 3 and brand 6 have a correlation of 0.48, so this two appear to be com
plements.
#brand 6 and brand 7 have a correlation of -0.15, so this two appear to be co
mplements.
```