Project

Yongfan Zhao

Dec 10, 2023 (due Dec 10, 2023, 11:59 pm)

Note: This is your project. No discussion with a fellow student is allowed. Honor code is in place.

If you only answer the first two questions, you can only earn up to a maximum of a B+ on the course. If you tackle the optional third question, then that will give you the possibility of getting up to a A-. To have a chance of getting a A or an A+ you must take the in-person final exam.

Question 1: Bayes pricing [40 points]

1. Open Rstudio in your class project folder. Download vandf.RDS from CANVAS to the data sub-folder and read it into the Rstudio. The file vandf has monthly excess return (premium) data on eleven Vanguard ETFs, "VOX" "VCR" "VDC" "VDE" "VFH" "VHT" "VIS" "VGT" "VAW" "VNQ" and "VPU". These are sector-specific ETFs. Check out Yahoo Finance for more details. In addition, the dataset contains prmsp500 (the excess return on the mkt portfolio) and the five Fama-French factors, hml, smb, rmw, cma, and mom. The data runs from Jan 2010 to Dec 2021, but after reading the data use data only up to and including Dec 2020. Use head and tail() to make sure that the data starts at the correct month and ends at the correct month.

```
# write your code here
rm(list = ls());
source("tools.r");
vandf = readRDS(file.path(datapath,"vandf.RDS"))
t1 = as.yearmon("Dec 2020")
dts = as.yearmon(rownames(vandf))
vandf = filter(vandf, dts<= t1)
head(vandf)
```

```
##                prmvox      prmvcr       prmvdc      prmvde       prmvfh
## Jan 2010 -0.077550230 -0.02527158 -0.009631995 -0.04286288 -0.008994286
## Feb 2010  0.011835204  0.05916519  0.030375111  0.02425238  0.035035165
## Mar 2010  0.065496374  0.08397908  0.040924560  0.02767504  0.089565131
## Apr 2010  0.007880549  0.06272150 -0.010585220  0.04463151  0.023618293
## May 2010 -0.044604899 -0.07202009 -0.049796049 -0.11953229 -0.088382148
## Jun 2010 -0.019703511 -0.10062828 -0.026332657 -0.05770016 -0.064557202
##                prmvht      prmvis      prmvgt      prmvaw      prmvnq
## Jan 2010  0.004202854 -0.02156171 -0.08187150 -0.08158095 -0.05524932
## Feb 2010  0.005270234  0.04927768  0.04499957  0.05083245  0.05577288
## Mar 2010  0.036094906  0.08657360  0.06695035  0.08117532  0.10177186
## Apr 2010 -0.027290050  0.04348928  0.02194714  0.01018843  0.07137694
## May 2010 -0.069395089 -0.08953991 -0.08006648 -0.08948745 -0.05347571
## Jun 2010 -0.021360130 -0.06997351 -0.06391469 -0.07813456 -0.05224735
##                prmvpu    prmsp500     hml     smb     rmw     cma     mom
## Jan 2010 -0.044994157 -0.03701592  0.0043  0.0034 -0.0127  0.0046 -0.0540
## Feb 2010 -0.009696771  0.02845535  0.0323  0.0151 -0.0027  0.0143  0.0374
## Mar 2010  0.029799859  0.05870057  0.0221  0.0185 -0.0065  0.0169  0.0376
```

```
## Apr 2010  0.029754626  0.01463420  0.0289  0.0498  0.0069  0.0172  0.0316
## May 2010 -0.058771365 -0.08210504 -0.0244  0.0004  0.0130 -0.0022 -0.0025
## Jun 2010 -0.008255006 -0.05400747 -0.0470 -0.0247 -0.0016 -0.0155 -0.0276
```

```
tail(vandf)
```

```
##                prmvox      prmvcr      prmvdc       prmvde       prmvfh
## Jul 2020  0.073734506  0.09198746  0.06895209 -0.042060853  0.029782491
## Aug 2020  0.080811760  0.12940620  0.04639746 -0.006294627  0.038633921
## Sep 2020 -0.054466490 -0.03341517 -0.01859805 -0.147547585 -0.038308348
## Oct 2020  0.007487659 -0.02351306 -0.02357720 -0.034801760  0.007450671
## Nov 2020  0.122884936  0.14116695  0.07898037  0.280321511  0.163656371
## Dec 2020  0.044736919  0.05233144  0.01957173  0.052455889  0.069209314
##                prmvht      prmvis      prmvgt      prmvaw       prmvnq
## Jul 2020  0.05200268  0.04468424  0.05930424 0.069018632  0.036306799
## Aug 2020  0.02351368  0.08244142  0.11128632 0.044976706  0.004353749
## Sep 2020 -0.01367952 -0.01105759 -0.04874752 0.002927877 -0.026722298
## Oct 2020 -0.02874741 -0.01098462 -0.04328816 0.003346509 -0.030085804
## Nov 2020  0.08829845  0.16823852  0.12523691 0.126902054  0.096679630
## Dec 2020  0.04151481  0.01995178  0.05708690 0.036672729  0.027278018
##                prmvpu    prmsp500     hml     smb     rmw     cma     mom
## Jul 2020  0.072257502  0.05498880 -0.0137 -0.0320  0.0040  0.0096  0.0761
## Aug 2020 -0.025480182  0.06999549 -0.0296 -0.0089  0.0426 -0.0120  0.0055
## Sep 2020  0.005747315 -0.03930715 -0.0268  0.0001 -0.0139 -0.0189  0.0312
## Oct 2020  0.048805902 -0.02773663  0.0422  0.0464 -0.0075 -0.0076 -0.0305
## Nov 2020  0.014183909  0.10747646  0.0213  0.0712 -0.0220  0.0137 -0.1243
## Dec 2020  0.008875154  0.03705895 -0.0150  0.0479 -0.0199 -0.0011 -0.0232
```

2. Suppose that you were asked to see how many of these 11 ETF excess returns can be priced by a given factor model. How would you use the Bayesian testing approach based on marginal likelihoods to figure this out? Write a function called bayesprice() that takes as inputs anames (the names of the asset premiums), fnames (the names of the factors in the model), data (the name of the data.frame that contains data on the asset premiums and the factors) and trainpct = .20. In the body of the function do the Bayes test of whether the asset premiums are priced at least the 2:1 posterior odds by the factors in the model. The output of the function is a data.frame object with three't columns: anames, diff and priced. The anames column has the names of the 11 ETFs; the diff column has the difference in marginal likelihood values; and the priced column has a TRUE if the asset is priced and FALSE otherwise.

```
#write your code here
bayesprice = function(anames, fnames, data, trainpct = 0.20) {
  results = data.frame(aname = character(), diff = numeric(), priced = logical())

  for (aname in anames) {

    formula_full = as.formula(paste0(aname, " ~ ", paste(fnames, collapse = " + "), " - 1"))
    formula_null = as.formula(paste0(aname, " ~ ", paste(fnames, collapse = " + ")))

    model_full = cbw537::MCMCregressg(modelfrm = formula_full, data = data, trainpct =  trainpct)
    model_null = cbw537::MCMCregressg(modelfrm = formula_null, data = data, trainpct = trainpct)

    # likelihoods
    logmarg_full = cbw537::logmarglik(model_full)
    logmarg_null = cbw537::logmarglik(model_null)

    # check 2:1 posterior
```

```
    diff_logmarg = logmarg_full - logmarg_null
    priced = 1/( 1+ exp(-diff_logmarg)) > 0.67

    # Append results
    results = rbind(results, data.frame(aname = aname, diff = diff_logmarg, priced = priced))
  }

  return(results)
}
```

3. Use the function bayesprice() to see which of the 11 ETFs are priced by the factors in each of the following models: CAPM (prmsp500), FF3 (prmsp500,hml,smb), FF5 (prmsp500,hml,smb,cma,rmw) and FF6 (prmsp500,hml,smb,cma,rmw,mom). Use the at least 2:1 posterior odds criterion to judge if an ETF is priced. And use trainpct = .20. Your final answer should be in four different data.frame objects called pricecapmdf, priceff3df, priceff5df and priceff6df.

```
# write your code here
anames = names(vandf)[1:11]
pricecapmdf = bayesprice(anames, "prmsp500", vandf, 0.20)
priceff3df = bayesprice(anames, c("prmsp500", "hml", "smb"), vandf, 0.20)
priceff5df = bayesprice(anames, c("prmsp500", "hml", "smb", "cma", "rmw"), vandf, 0.20)
priceff6df = bayesprice(anames, c("prmsp500", "hml", "smb", "cma", "rmw", "mom"), vandf, 0.20)
```

4. Which model or models prices the most ETFs?

```
# write your code here
sum(pricecapmdf[3] == TRUE)
```

```
## [1] 6
```

```
sum(priceff3df[3]== TRUE)
```

```
## [1] 5
```

```
sum(priceff5df[3] == TRUE)
```

```
## [1] 5
```

```
sum(priceff6df[3] == TRUE)
```

```
## [1] 6
```

```
cat("CAPM AND FF6DF is better to prices ETFS")
```

```
## CAPM AND FF6DF is better to prices ETFS
```

Question 2: Chib and Zeng [30 points]

1. Given the six factors in vandf, suppose that you wanted to find the best risk factors. How would you use the Chib and Zeng method to figure out the best risk factors? Use trainpct = .20 and call the best risk factors xbest.

```
# write your code here
sixf = vandf[,12:17]
dscandfls = CZZscang(data = sixf, trainpct = .20)
```

```
## starting Chib, Zeng and Zhao (2020) model scan ...
## there are 63 models in the model space
## model scan started ...
```

```
## model scan finished ...
## preparing output ...
```

```
dscandford = dscandfls$scanord
xbest = names(which(dscandford[1,] == 1))
print(xbest)
```

```
## [1] "prmsp500" "hml"
```

2. Now use xbest to see which of the 11 ETFs can be priced by xbest. Call the output data.frame pricexbestdf.

```
# write your code here
pricexbestdf = bayesprice(anames = anames,
                          fnames = xbest,
                          data = vandf)
print(pricexbestdf)
```

```
##     aname        diff priced
## 1  prmvox  1.46892200   TRUE
## 2  prmvcr  0.99768100   TRUE
## 3  prmvdc  1.06303215   TRUE
## 4  prmvde -2.11538775  FALSE
## 5  prmvfh  0.12174496  FALSE
## 6  prmvht  0.50287931  FALSE
## 7  prmvis  1.51806478   TRUE
## 8  prmvgt -0.34854649  FALSE
## 9  prmvaw  1.42616689   TRUE
## 10 prmvnq  1.66153592   TRUE
## 11 prmvpu -0.03992687  FALSE
```

3. How many assets are priced by xbest and how does this compare with the numbers priced by the CAPM, FF3, FF5 and FF6 models?

```
# write your code here
sum(pricexbestdf[3] == TRUE)
```

```
## [1] 6
```

```
cat("xbest has same number of true with CAPM and FF6")
```

```
## xbest has same number of true with CAPM and FF6
```

Question 3: (optional, gives you the chance, but not the certainty, of getting up to an A-) [50 points]

1. (0 points) Load the following data on characteristics. Remember the mve characteristic in these data is in logs. You have to exponentiate it before creating factors.

```
datls = readRDS(file = file.path(datapath,"datcappedls.RDS"));
dts = readRDS(file = file.path(datapath,"dts.RDS"));
chnames = readRDS(file = file.path(datapath,"chnames.RDS"));
chnames = sort(chnames);
kc = length(chnames);
mkt = readRDS(file = file.path(datapath,"MktRf.RDS"))
nT = length(datls);
```

2. The underlying characteristics of the FF5 model are bm for smb, mve for smb, agr for cma and operprof for rmw. Make linear slope factors for these characteristics with the following controls on the RHS

(in addition to these 4 characteristics): CASE 1: no additional controls CASE 2: mom12m as the additional control CASE 3: the remaining six characteristics as additional controls. Please these slope factors in data.frame objects called sf1df, sf2df, and sf3df. Merge each of these data.frames with mkt. Finally, merge each data.frame with vandf. The final data.frames should each cover the period from Jan 2010 to Dec 2020.

```
# write your code here
# use the function from Constructing Factors by siddhartha Chib
calculateslopes = function(data = data,
                           rename = "Re",
                           chnames = chnames,
                           mvename = "mve",
                           mveinlogs = TRUE) {
X = data[,chnames];
if (mveinlogs) {
  X[,mvename] = exp(X[,mvename])
}
Xs = scale(X);
datas = cbind(data[,c("date","PERMNO",rename)],Xs)
rhsfrm = paste0(chnames,collapse = "+");
frm = as.formula(paste0(rename,"~",rhsfrm));
lmout = lm(formula = frm,
           data = datas);
betahat = lmout$coefficients[-1];
return(betahat);
}

datls = datls[as.yearmon(names(datls))>= as.yearmon("Jan 2010") & as.yearmon(names(datls))<= as.yearmon


dts = dts[dts<=as.yearmon("Dec 2020") & dts >= as.yearmon("Jan 2010")]

chnamecase1 = c("bm", "mve", "agr", "operprof")
chnamecase2 = c("bm", "mve", "agr", "operprof","mom12m")

chnamecase3 = names(datls[[1]][-(1:3)])
sf1 = sapply(datls,
             FUN = "calculateslopes",
             chnames = chnamecase1);
sf1 = t(100*sf1);
colnames(sf1) = paste0("sf1.",chnamecase1)
sf1 = xts(sf1,order.by = dts)
sf1 = merge(mkt,sf1,join = "inner")
sf1df = as.data.frame(sf1)
sf1df = cbind(vandf,sf1df)

head(sf1df)
```

```
##                prmvox      prmvcr       prmvdc      prmvde       prmvfh
## Jan 2010 -0.077550230 -0.02527158 -0.009631995 -0.04286288 -0.008994286
## Feb 2010  0.011835204  0.05916519  0.030375111  0.02425238  0.035035165
## Mar 2010  0.065496374  0.08397908  0.040924560  0.02767504  0.089565131
## Apr 2010  0.007880549  0.06272150 -0.010585220  0.04463151  0.023618293
## May 2010 -0.044604899 -0.07202009 -0.049796049 -0.11953229 -0.088382148
## Jun 2010 -0.019703511 -0.10062828 -0.026332657 -0.05770016 -0.064557202
```

```
##                prmvht      prmvis      prmvgt      prmvaw      prmvnq
## Jan 2010  0.004202854 -0.02156171 -0.08187150 -0.08158095 -0.05524932
## Feb 2010  0.005270234  0.04927768  0.04499957  0.05083245  0.05577288
## Mar 2010  0.036094906  0.08657360  0.06695035  0.08117532  0.10177186
## Apr 2010 -0.027290050  0.04348928  0.02194714  0.01018843  0.07137694
## May 2010 -0.069395089 -0.08953991 -0.08006648 -0.08948745 -0.05347571
## Jun 2010 -0.021360130 -0.06997351 -0.06391469 -0.07813456 -0.05224735
##                prmvpu    prmsp500     hml     smb     rmw     cma     mom   Mkt
## Jan 2010 -0.044994157 -0.03701592  0.0043  0.0034 -0.0127  0.0046 -0.0540 -3.36
## Feb 2010 -0.009696771  0.02845535  0.0323  0.0151 -0.0027  0.0143  0.0374  3.40
## Mar 2010  0.029799859  0.05870057  0.0221  0.0185 -0.0065  0.0169  0.0376  6.31
## Apr 2010  0.029754626  0.01463420  0.0289  0.0498  0.0069  0.0172  0.0316  2.00
## May 2010 -0.058771365 -0.08210504 -0.0244  0.0004  0.0130 -0.0022 -0.0025 -7.89
## Jun 2010 -0.008255006 -0.05400747 -0.0470 -0.0247 -0.0016 -0.0155 -0.0276 -5.57
##               sf1.bm    sf1.mve    sf1.agr sf1.operprof
## Jan 2010  1.56717798 -0.4963085 -0.2312201  -0.35763965
## Feb 2010  0.01736248 -0.1787937 -0.5140801   0.45522572
## Mar 2010  1.03890649 -0.4638316 -0.8076209  -0.04270479
## Apr 2010  2.68071547 -1.4562824 -1.0157286  -0.06524303
## May 2010 -1.11329797 -0.1523862  0.1118521   0.53339907
## Jun 2010 -1.28039843  0.4068068  0.9490660   0.13588845
```

```
sf2 = sapply(datls,
             FUN = "calculateslopes",
             chnames = chnamecase2);
sf2 = t(100*sf2);
colnames(sf2) = paste0("sf2.",chnamecase2)
sf2 = xts(sf2,order.by = dts)
sf2 = merge(mkt,sf2,join = "inner")
sf2df = as.data.frame(sf2)
sf2df = cbind(vandf,sf2df)

head(sf2df)
```

```
##                prmvox      prmvcr       prmvdc      prmvde       prmvfh
## Jan 2010 -0.077550230 -0.02527158 -0.009631995 -0.04286288 -0.008994286
## Feb 2010  0.011835204  0.05916519  0.030375111  0.02425238  0.035035165
## Mar 2010  0.065496374  0.08397908  0.040924560  0.02767504  0.089565131
## Apr 2010  0.007880549  0.06272150 -0.010585220  0.04463151  0.023618293
## May 2010 -0.044604899 -0.07202009 -0.049796049 -0.11953229 -0.088382148
## Jun 2010 -0.019703511 -0.10062828 -0.026332657 -0.05770016 -0.064557202
##                prmvht      prmvis      prmvgt      prmvaw      prmvnq
## Jan 2010  0.004202854 -0.02156171 -0.08187150 -0.08158095 -0.05524932
## Feb 2010  0.005270234  0.04927768  0.04499957  0.05083245  0.05577288
## Mar 2010  0.036094906  0.08657360  0.06695035  0.08117532  0.10177186
## Apr 2010 -0.027290050  0.04348928  0.02194714  0.01018843  0.07137694
## May 2010 -0.069395089 -0.08953991 -0.08006648 -0.08948745 -0.05347571
## Jun 2010 -0.021360130 -0.06997351 -0.06391469 -0.07813456 -0.05224735
##                prmvpu    prmsp500     hml     smb     rmw     cma     mom   Mkt
## Jan 2010 -0.044994157 -0.03701592  0.0043  0.0034 -0.0127  0.0046 -0.0540 -3.36
## Feb 2010 -0.009696771  0.02845535  0.0323  0.0151 -0.0027  0.0143  0.0374  3.40
## Mar 2010  0.029799859  0.05870057  0.0221  0.0185 -0.0065  0.0169  0.0376  6.31
## Apr 2010  0.029754626  0.01463420  0.0289  0.0498  0.0069  0.0172  0.0316  2.00
## May 2010 -0.058771365 -0.08210504 -0.0244  0.0004  0.0130 -0.0022 -0.0025 -7.89
## Jun 2010 -0.008255006 -0.05400747 -0.0470 -0.0247 -0.0016 -0.0155 -0.0276 -5.57
```

```
##              sf2.bm    sf2.mve    sf2.agr sf2.operprof  sf2.mom12m
## Jan 2010  1.8432660 -0.4939234 -0.5076298  -0.20819140 -1.67056597
## Feb 2010 -0.2283406 -0.1638112 -0.2651907   0.30702033  1.61367410
## Mar 2010  1.0947271 -0.4691906 -0.8642594  -0.02321335 -0.32793933
## Apr 2010  2.6223700 -1.4488746 -0.9532901  -0.08516136  0.34706408
## May 2010 -1.1117382 -0.1524759  0.1100738   0.53381371 -0.01068744
## Jun 2010 -1.2057935  0.3948562  0.8689798   0.15703380 -0.45369314
```

```
sf3 = sapply(datls,
             FUN = "calculateslopes",
             chnames = chnamecase3);
sf3 = t(100*sf3);
colnames(sf3) = paste0("sf3.",chnamecase3)
sf3 = xts(sf3,order.by = dts)
sf3 = merge(mkt,sf3,join = "inner")
sf3df = as.data.frame(sf3)
sf3df = cbind(vandf,sf3df)

head(sf3df)
```

```
##                prmvox      prmvcr       prmvdc      prmvde       prmvfh
## Jan 2010 -0.077550230 -0.02527158 -0.009631995 -0.04286288 -0.008994286
## Feb 2010  0.011835204  0.05916519  0.030375111  0.02425238  0.035035165
## Mar 2010  0.065496374  0.08397908  0.040924560  0.02767504  0.089565131
## Apr 2010  0.007880549  0.06272150 -0.010585220  0.04463151  0.023618293
## May 2010 -0.044604899 -0.07202009 -0.049796049 -0.11953229 -0.088382148
## Jun 2010 -0.019703511 -0.10062828 -0.026332657 -0.05770016 -0.064557202
##                prmvht      prmvis      prmvgt      prmvaw      prmvnq
## Jan 2010  0.004202854 -0.02156171 -0.08187150 -0.08158095 -0.05524932
## Feb 2010  0.005270234  0.04927768  0.04499957  0.05083245  0.05577288
## Mar 2010  0.036094906  0.08657360  0.06695035  0.08117532  0.10177186
## Apr 2010 -0.027290050  0.04348928  0.02194714  0.01018843  0.07137694
## May 2010 -0.069395089 -0.08953991 -0.08006648 -0.08948745 -0.05347571
## Jun 2010 -0.021360130 -0.06997351 -0.06391469 -0.07813456 -0.05224735
##                prmvpu    prmsp500     hml     smb     rmw     cma     mom   Mkt
## Jan 2010 -0.044994157 -0.03701592  0.0043  0.0034 -0.0127  0.0046 -0.0540 -3.36
## Feb 2010 -0.009696771  0.02845535  0.0323  0.0151 -0.0027  0.0143  0.0374  3.40
## Mar 2010  0.029799859  0.05870057  0.0221  0.0185 -0.0065  0.0169  0.0376  6.31
## Apr 2010  0.029754626  0.01463420  0.0289  0.0498  0.0069  0.0172  0.0316  2.00
## May 2010 -0.058771365 -0.08210504 -0.0244  0.0004  0.0130 -0.0022 -0.0025 -7.89
## Jun 2010 -0.008255006 -0.05400747 -0.0470 -0.0247 -0.0016 -0.0155 -0.0276 -5.57
##             sf3.agr   sf3.beta      sf3.bm    sf3.cfp sf3.idiovol sf3.operprof
## Jan 2010 -0.4483747 -0.6761343  0.72256866  0.1425619   1.8322645  -0.06901363
## Feb 2010 -0.3224775  1.1076213  0.64207773  0.1547921  -0.9213552   0.22181776
## Mar 2010 -0.6718264  2.1937912  1.04462457  0.2888997   1.5069257   0.03780020
## Apr 2010 -0.7769873  0.9836964  0.87704096  0.2688341   1.8317362   0.14282572
## May 2010 -0.1769829 -1.9072844 -0.51842061 -0.1797226  -1.5018899   0.43473316
## Jun 2010  0.4817709 -2.9013041  0.05037332  0.3943859  -1.4479547  -0.15410038
##             sf3.lev sf3.mom12m  sf3.mom1m     sf3.mve
## Jan 2010  1.5816820 -1.9636730 -1.7092783 -0.29279746
## Feb 2010 -1.5210138  1.3629625 -0.5363618 -0.32776195
## Mar 2010 -0.8689256 -1.7909305 -0.4498968 -0.09056402
## Apr 2010  2.8566499 -0.6210250  0.3080699 -1.08931748
## May 2010 -0.3636484  0.9970352  0.1752052 -0.48816683
## Jun 2010 -1.6690295  0.7104049  1.2146817  0.03281432
```

3. Now use the data in sf1df, sf2df and sf3df to price the 11 ETFs with the FF5 slope factors. Call the three output data.frame objects price1df, price2df and price3df.

```
# write your code here
chnamecase4 = c("prmsp500","hml","smb","cma","rmw")

price1df = bayesprice(anames = anames,
                      fnames = chnamecase4,
                      data = sf1df)

head(price1df)
```

```
##    aname       diff priced
## 1 prmvox  1.7499502   TRUE
## 2 prmvcr  0.4254395  FALSE
## 3 prmvdc  1.3053064   TRUE
## 4 prmvde -2.3998642  FALSE
## 5 prmvfh -1.2994267  FALSE
## 6 prmvht  0.6260827  FALSE
```

```
price2df = bayesprice(anames = anames,
                      fnames = chnamecase4,
                      data = sf2df)
price3df = bayesprice(anames = anames,
                      fnames = chnamecase4,
                      data = sf3df)
```

4. How many assets are priced by the different sets of slope factors and how does this compare with the number priced in priceff5df above?

```
# write your code here

sum(price1df[3] == TRUE)
```

```
## [1] 5
```

```
sum(price2df[3] == TRUE)
```

```
## [1] 5
```

```
sum(price3df[3] == TRUE)
```

```
## [1] 5
```