Big Data HW 7

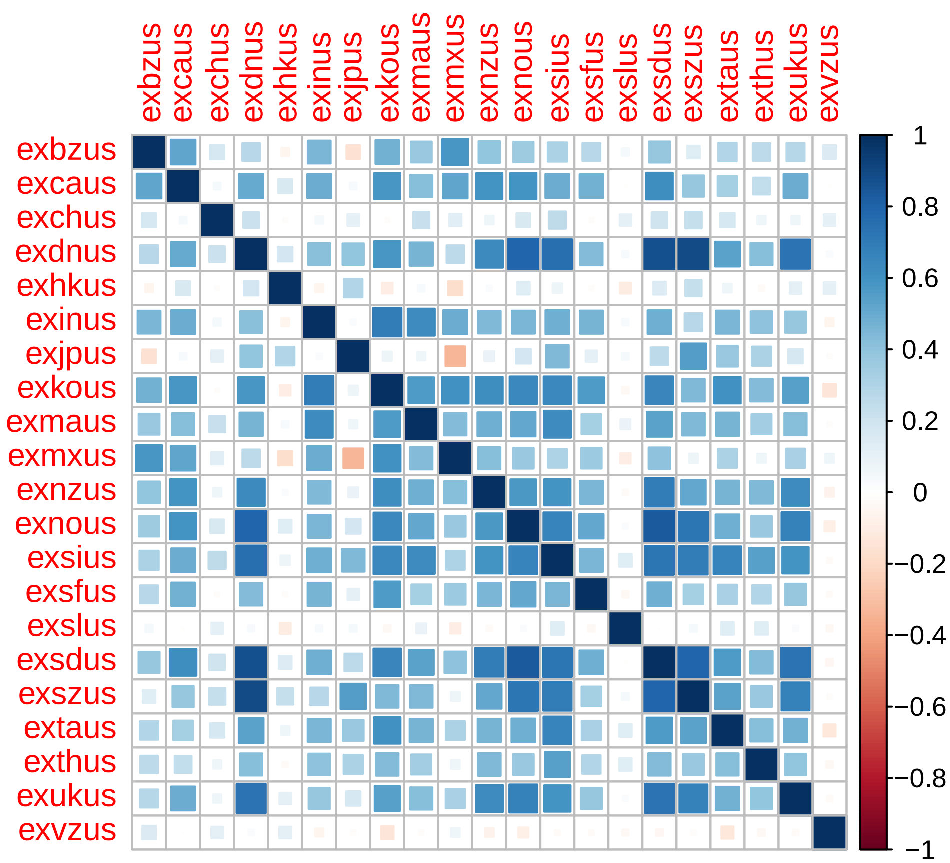
2024-05-14

[1] Discuss correlation amongst dimensions of fx. How does this relate to the applicability of factor modelling?

Factor modeling is for continuous variables when we want to capture the main structure in the data with a few informative features. Factor models imply mixed membership so based on the correlation of our plots we are going to find the topics that we can group our data into to reduce the dimensions.

```
fx = read.csv("FXmonthly.csv")
fx_returns = (fx[2:120,]-fx[1:119,])/(fx[1:119,])
```

```
fx_data <- fx_data |>
  mutate(across(everything(), as.numeric)) |>
  select(!c("exalus", "exeuus"))

fx_returns <- (fx_data[2:120,]-fx_data[1:119,])/(fx_data[1:119,])

fx_correlations <- cor(fx_returns, use = "complete.obs")
corrplot(fx_correlations, method = "square")
```




```
[2] Fit, plot, and interpret principal components.
```

When we plot the variances, we can see there is a big drop in variance from the first to the second PC.

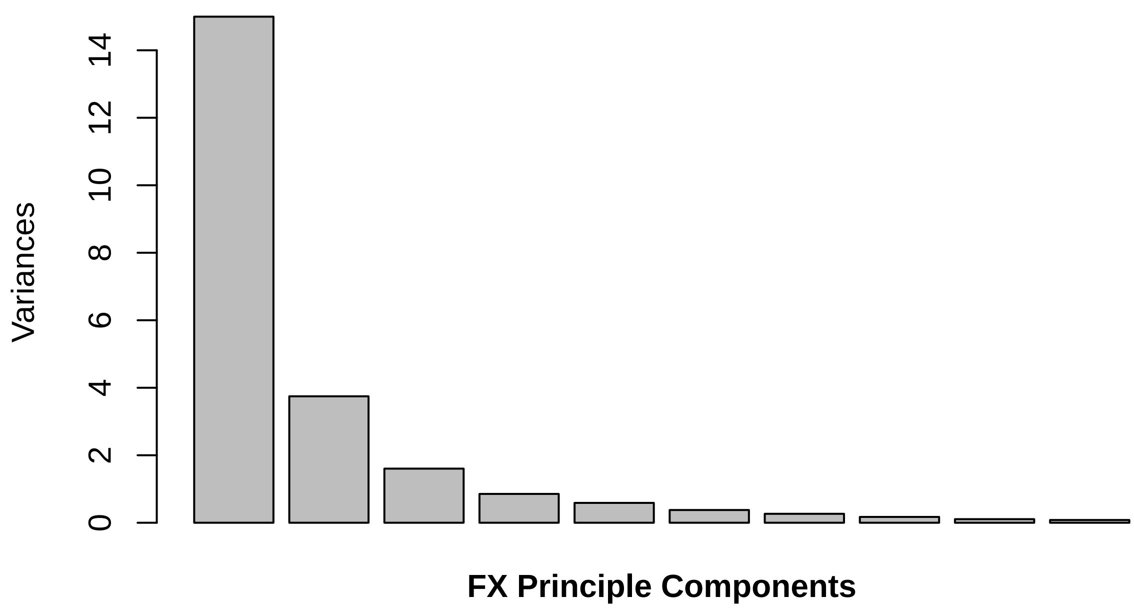
```r
mypca=prcomp(fx, scale=TRUE)
predict(mypca)[,1:2]
```

```
##                 PC1         PC2
## JAN2001 -5.30992704  0.11080332
## FEB2001 -5.54302992  0.21573786
## MAR2001 -6.28922753  0.26426053
## APR2001 -6.77858167  0.39439076
## MAY2001 -6.77837228  0.34979601
## JUN2001 -7.32172190  0.53614426
## JUL2001 -7.47512537  0.54778872
## AUG2001 -6.69789248  0.39513219
## SEP2001 -6.73044763  0.70267979
## OCT2001 -7.04810116  0.79356860
## NOV2001 -7.10071163  0.89530096
## DEC2001 -7.31119617  1.21309789
## JAN2002 -7.50844926  1.22012924
## FEB2002 -7.65167989  1.29951428
## MAR2002 -7.28734971  1.32268678
## APR2002 -6.98235085  1.18201751
## MAY2002 -6.06910442  0.87106629
## JUN2002 -5.21550993  0.61334289
## JUL2002 -4.61814952  0.35944403
## AUG2002 -5.15106630  0.58319335
## SEP2002 -5.28228011  0.49139010
## OCT2002 -5.52381084  0.31946133
## NOV2002 -4.95386060  0.08163468
## DEC2002 -4.65510105 -0.17049637
## JAN2003 -3.74693634 -0.26595806
## FEB2003 -3.61135484 -0.34620171
## MAR2003 -3.63201863 -0.16803325
## APR2003 -3.51115985 -0.37632607
## MAY2003 -2.56079594 -0.72827711
## JUN2003 -2.35198404 -0.81187839
## JUL2003 -2.61444981 -0.88260002
## AUG2003 -2.77044361 -0.79216909
## SEP2003 -2.28423536 -0.67972011
## OCT2003 -1.32209827 -0.34041403
## NOV2003 -1.33086681 -0.61789992
## DEC2003 -0.77564427 -0.79304998
## JAN2004 -0.37899963 -0.94999459
## FEB2004 -0.26676839 -1.19551336
## MAR2004 -0.68681031 -1.39874946
## APR2004 -0.66746616 -1.46634694
## MAY2004 -1.16877338 -1.19266633
## JUN2004 -0.94230847 -1.31792026
## JUL2004 -0.75708608 -1.30266281
## AUG2004 -0.89030944 -1.18957717
## SEP2004 -0.74169935 -1.15005410
```

```
## OCT2004 -0.20411097 -1.19099090
## NOV2004  0.84398597 -1.43956222
## DEC2004  1.37705025 -1.93959816
## JAN2005  1.00857912 -2.15354029
## FEB2005  1.00307429 -2.29980961
## MAR2005  1.27232265 -2.40948346
## APR2005  0.94875125 -2.33535062
## MAY2005  0.85938720 -2.17248704
## JUN2005  0.43806016 -1.81403377
## JUL2005  0.05410007 -1.63980974
## AUG2005  0.50563088 -1.65470492
## SEP2005  0.44572223 -1.47591263
## OCT2005  0.06289019 -1.11303171
## NOV2005 -0.30785434 -0.95534774
## DEC2005 -0.09582303 -1.12277414
## JAN2006  0.65988474 -1.49211572
## FEB2006  0.45268038 -1.50148351
## MAR2006  0.47693843 -1.28127984
## APR2006  0.92024523 -1.22524580
## MAY2006  1.63936945 -1.24052052
## JUN2006  1.11307824 -1.03362485
## JUL2006  0.98074996 -1.15923933
## AUG2006  1.10304025 -1.30280014
## SEP2006  0.91313409 -1.33141908
## OCT2006  0.85075282 -1.34345132
## NOV2006  1.46791940 -1.51985036
## DEC2006  2.01578161 -1.60311707
## JAN2007  1.70649168 -1.82727029
## FEB2007  1.81034115 -1.93337937
## MAR2007  2.04737570 -1.77681891
## APR2007  2.48589368 -2.28386888
## MAY2007  2.60002207 -2.55800568
## JUN2007  2.66550298 -2.53589739
## JUL2007  3.21577049 -2.74447360
## AUG2007  2.91707485 -2.30970266
## SEP2007  3.44547232 -1.96534078
## OCT2007  4.28564420 -1.95479635
## NOV2007  4.48730398 -2.15490900
## DEC2007  4.17899339 -2.24658672
## JAN2008  4.42049285 -2.00805477
## FEB2008  4.64275972 -1.71559467
## MAR2008  5.31998222 -1.23082549
## APR2008  5.40064439 -1.43599142
## MAY2008  5.02053091 -1.18881683
## JUN2008  4.76605235 -1.20132830
## JUL2008  4.93362941 -1.25750730
## AUG2008  3.94311911 -1.08435395
## SEP2008  3.06915799  0.28432808
## OCT2008  1.52231182  2.96310598
## NOV2008  0.71008519  4.02556483
## DEC2008  1.11061477  4.20566235
## JAN2009  1.10989682  4.46225522
## FEB2009  0.43010578  5.03846343
## MAR2009  0.26930917  5.21412295
```

```
## APR2009  1.16502876  4.07520447
## MAY2009  2.28346241  3.28199146
## JUN2009  2.77508285  2.86051289
## JUL2009  2.81634595  2.97263740
## AUG2009  3.20956768  2.65655925
## SEP2009  3.74073742  2.67953177
## OCT2009  4.32125936  2.36387748
## NOV2009  4.41890364  2.25029706
## DEC2009  4.12906345  2.24658848
## JAN2010  4.53210028  2.17559427
## FEB2010  4.18078645  2.58356949
## MAR2010  4.46306901  2.51397496
## APR2010  4.69530422  2.14755431
## MAY2010  3.74506849  2.67710280
## JUN2010  3.42780766  2.94364180
## JUL2010  4.13500086  3.00539566
## AUG2010  4.52778440  2.86706747
## SEP2010  5.06249712  2.83193115
## OCT2010  5.89652476  2.38668281
## NOV2010  5.83006523  2.47205177
## DEC2010  5.61987672  2.35276499
```

```
plot(mypca, main="")
mtext(side=1, "FX Principle Components",  line=1, font=2)
```




```
loadings <- mypca$rotation
loadings[, 1:2]
```

```
##               PC1         PC2
## exalus -0.2524173  0.06354389
## exbzus -0.1685751 -0.09786507
## excaus -0.2518524  0.02519675
## exchus -0.1947087 -0.29855539
## exdnus -0.2470701  0.04643307
## exhkus -0.1053692 -0.25361892
## exinus -0.1469827  0.35764699
```

```
## exjpus -0.1877235 -0.27820237
## exkous -0.1421975  0.39149084
## exmaus -0.2075953 -0.12801229
## exmxus  0.1711408  0.29287745
## exnzus -0.2371120  0.15003198
## exnous -0.2433417  0.09698880
## exsius -0.2404687 -0.13700676
## exsfus -0.1379518  0.28477287
## exslus  0.2250650  0.16308632
## exsdus -0.2383978  0.15673553
## exszus -0.2485299 -0.04665784
## extaus -0.2019389  0.04937790
## exthus -0.2388732 -0.07672643
## exukus -0.1510349  0.39223034
## exvzus  0.2039186  0.14120376
## exeuus -0.2472840  0.04655334
```

```
loadings[order(abs(loadings[,2]), decreasing = TRUE)[1:5],2]
```

```
##     exukus     exkous     exinus     exchus     exmxus
##  0.3922303  0.3914908  0.3576470 -0.2985554  0.2928774
```

When we plot the variances, we can see there is a big drop in variance from the first to the second PC; there is also a noticeable drop from PC 2 to PC 3. Each subsequent PC contributes less to the model as seen by the decreasing variance - this plateaus from PC 3. In this case, however, it seems like only the first PC really matters.

The highest PC 1 is the Venezuela-US exchange rate, which is the only positive value. Alternatively, the lowest scoring in PC 1 includes Albania-US, Sweden-US, Denmark-US and EU-US among others. The lower scores are countries/ regions that have strong international currency pricing with the US.

The highest PC 2 is the Mexico-US exchange rate, followed by the US exchange rates with Brazil and India. The lowest scoring in PC 2 includes Japan, EU, Denmark among others.

```
x_fx <- scale(fx_data)
pc_fx <- prcomp(x_fx)
z_fx <- predict(pc_fx)

# Predict is just doing the same thing as the below:
z <- x_fx%*%pc_fx$rotation
all(z==z_fx)
```

```
## [1] FALSE
```

```
# Implies rotations are on scale of standard deviations if scale=TRUE
ROTATION <- pc_fx$rotation
ROTATION <- apply(ROTATION, 2, function(x){round(x,3)})
ROTATION[,(1:2)]
```

```
##          PC1    PC2
## exbzus  0.185 -0.087
## excaus  0.269  0.037
## exchus  0.213 -0.290
## exdnus  0.262  0.055
## exhkus  0.114 -0.252
## exinus  0.155  0.367
## exjpus  0.203 -0.272
## exkous  0.149  0.400
```

```
## exmaus  0.226 -0.116
## exmxus -0.184  0.289
## exnzus  0.250  0.159
## exnous  0.258  0.107
## exsius  0.260 -0.125
## exsfus  0.143  0.290
## exslus -0.242  0.154
## exsdus  0.251  0.166
## exszus  0.264 -0.038
## extaus  0.217  0.061
## exthus  0.257 -0.065
## exukus  0.156  0.399
## exvzus -0.220  0.132
```

```
t(round(mypca$rotation[,1:2],2))
```

```
##     exalus exbzus excaus exchus exdnus exhkus exinus exjpus exkous exmaus
## PC1  -0.25  -0.17  -0.25  -0.19  -0.25  -0.11  -0.15  -0.19  -0.14  -0.21
## PC2   0.06  -0.10   0.03  -0.30   0.05  -0.25   0.36  -0.28   0.39  -0.13
##     exmxus exnzus exnous exsius exsfus exslus exsdus exszus extaus exthus
## PC1   0.17  -0.24  -0.24  -0.24  -0.14   0.23  -0.24  -0.25  -0.20  -0.24
## PC2   0.29   0.15   0.10  -0.14   0.28   0.16   0.16  -0.05   0.05  -0.08
##     exukus exvzus exeuus
## PC1  -0.15   0.20  -0.25
## PC2   0.39   0.14   0.05
```

[3] Regress SP500 returns onto currency movement factors, using both 'glm on first K' and lasso techniques. Use the results to add to your factor interpretation.

LASSO Technique

```
library(glmnet)
```

```
## Loaded glmnet 4.1-8
```

```
# Prepare data for lasso regression
x <- as.matrix(fx_returns)
y <- sp_data |>
  select(sp500)

y <- as.matrix(y)

# Fit lasso model
lasso_model <- gamlr(x, y, family="gaussian", lambda.min.ratio=1e-3)

# Pick the lambda
lambda <- lasso_model$lambda[which.min(AICc(lasso_model))]
lambda
```

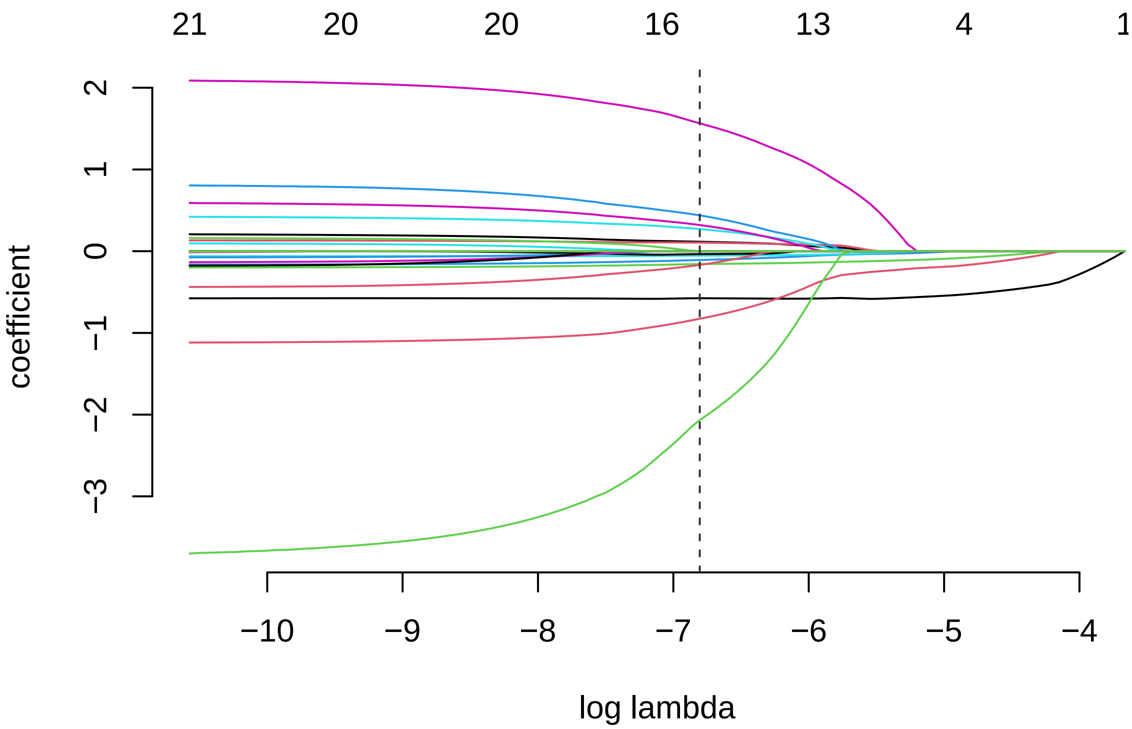
```
##       seg46
## 0.001108196
```

```
# The r-squared
dev <- lasso_model$deviance[which.min(AICc(lasso_model))]
dev0 <- lasso_model$deviance[1]
1-dev/dev0
```

```
##     seg46
```

```
## 0.4808235
```

```
plot(lasso_model)
```



GLM on first K

z = predict(pc_fx)[,1:5] reg = glm(y ~ ., data = as.data.frame(z))

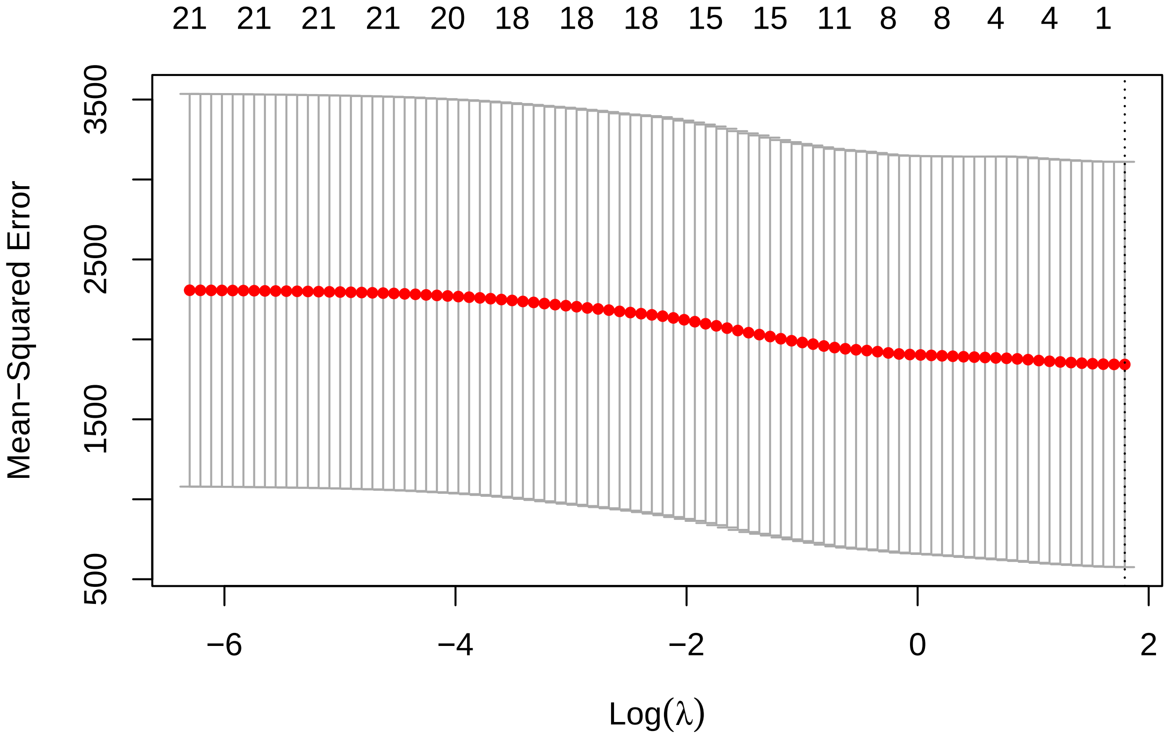
[4] Fit lasso to the original covariates and describe how it differs from PCR here.

LASSO is a dimension reduction method, when we do the principal component regression we are using rotated data. Both of these reduce the dimensions but the PCR is a little harder to interpret since we lose some of the original meaning when we rotate the data to the axis.

```
x <- as.matrix(fx_data)

sp500_ret <- (sp500[2:120,] - sp500[1:119,]) / (sp500[1:119,])
y <- sp500_ret$sp500

valid_indices <- !is.na(y) & !apply(x, 1, anyNA)
```

```
## Warning in !is.na(y) & !apply(x, 1, anyNA): longer object length is not a
## multiple of shorter object length
```

```
x <- x[valid_indices, ]
y <- y[valid_indices]

x <- x[-1, ]
y <- na.omit(y)

original_lasso_mod <- cv.glmnet(x, y, alpha = 1, family = "gaussian")
plot(original_lasso_mod)
```




```
which.min(original_lasso_mod$lambda)
```

```
## [1] 88
```

coef(original_lasso_mod, s = original_lasso_mod$lambda[93])