Multivariate Statistics

Assignment 1

Tom Leppens r0716859
Wenhan Cu r0973145
Zeynep Deniz Guvenol r0768138
Thien Nguyen r0774196

Question A

We load the data and rename the variables following the factors. There are concepts measuring well-being that have opposite scaling, but since they are consistent with the concept well-being, we decided to keep them as original and not inverting them. We then center the data, compute the covariance matrix, fit a CFA model with three correlated factors (one for each attitude concept), and assuming each item only has a loading on the concept it aims to measure. We print fit measures, the standardized solution and we compute, for each latent variable, the composite reliability, the average variance extracted and the maximum shared variance with other latent variables.

```
> load("ess.Rdata")
> names(ess)[2:14]<-c("sotru1","sotru2","sotru3","truin1","truin2","truin3","truin4",
+                     "webe1","webe2","webe3","webe4","webe5","webe6")
>
> centered_ess <- ess %>%
+   mutate(across(2:14, ~ . - mean(., na.rm = TRUE)))
> covmat<-cov(centered_ess[-1])

##specify model with 3 correlated factors
cfa1<-'sotru =~NA*+sotru1+sotru2+sotru3
       truin =~NA*truin1+truin2+truin3+truin4
       webe  =~NA*webe1+webe2+webe3+webe4+webe5+webe6
       sotru ~~1*sotru
       truin ~~1*truin
       webe  ~~1*webe'

#fit model on covariance matrix
fitcfa1<-cfa(cfa1,sample.cov=covmat,sample.nobs=4046)

> #standardized solution
> d<-standardizedSolution(fitcfa1)
> d
      lhs op    rhs est.std    se      z pvalue ci.lower ci.upper
1   sotru =~ sotru1   0.684 0.013 52.036      0    0.658    0.709
2   sotru =~ sotru2   0.648 0.013 48.322      0    0.622    0.674
3   sotru =~ sotru3   0.626 0.014 46.031      0    0.600    0.653
4   truin =~ truin1   0.789 0.008 93.956      0    0.773    0.805
5   truin =~ truin2   0.718 0.010 74.774      0    0.699    0.737
6   truin =~ truin3   0.581 0.012 48.194      0    0.557    0.605
7   truin =~ truin4   0.802 0.008 97.758      0    0.786    0.818
8    webe =~  webe1   0.661 0.011 60.710      0    0.640    0.683
9    webe =~  webe2   0.670 0.011 62.343      0    0.649    0.691
10   webe =~  webe3   0.589 0.012 48.379      0    0.565    0.612
11   webe =~  webe4   0.718 0.010 72.725      0    0.699    0.738
12   webe =~  webe5   0.677 0.011 63.729      0    0.656    0.698
13   webe =~  webe6   0.595 0.012 49.291      0    0.571    0.618
14  sotru ~~  sotru   1.000 0.000     NA     NA    1.000    1.000
15  truin ~~  truin   1.000 0.000     NA     NA    1.000    1.000
16   webe ~~   webe   1.000 0.000     NA     NA    1.000    1.000
17 sotru1 ~~ sotru1   0.533 0.018 29.674      0    0.498    0.568
18 sotru2 ~~ sotru2   0.580 0.017 33.355      0    0.546    0.614
19 sotru3 ~~ sotru3   0.608 0.017 35.629      0    0.574    0.641
20 truin1 ~~ truin1   0.377 0.013 28.488      0    0.352    0.403
21 truin2 ~~ truin2   0.485 0.014 35.192      0    0.458    0.512
22 truin3 ~~ truin3   0.662 0.014 47.280      0    0.635    0.690
23 truin4 ~~ truin4   0.357 0.013 27.162      0    0.331    0.383
24  webe1 ~~  webe1   0.562 0.014 39.020      0    0.534    0.591
25  webe2 ~~  webe2   0.551 0.014 38.292      0    0.523    0.579
26  webe3 ~~  webe3   0.654 0.014 45.641      0    0.626    0.682
27  webe4 ~~  webe4   0.484 0.014 34.112      0    0.456    0.512
28  webe5 ~~  webe5   0.542 0.014 37.693      0    0.514    0.570
29  webe6 ~~  webe6   0.646 0.014 45.059      0    0.618    0.675
30  sotru ~~  truin   0.555 0.016 34.183      0    0.524    0.587
31  sotru ~~   webe   0.287 0.020 14.604      0    0.248    0.326
32  truin ~~   webe   0.185 0.018 10.022      0    0.149    0.221
```

```
#print fit measures
> fitmeasures(fitcfa1,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"))
   chisq        df    pvalue       cfi       tli     rmsea      srmr
1526.049    62.000     0.000     0.912     0.889     0.076     0.040

> factorscore<-c("sotru","truin","webe")
> #composite reliability
> reliability<-round(c(compositerel(d[1:3,4]),compositerel(d[4:6,4]),compositerel(d[7:
9,4])),3)
> #average variance extracted
> average_var_extracted<-round(c(mean(d[1:3,4]^2),mean(d[4:6,4]^2),mean(d[7:9,4]^2)),3
)
> #maximum shared variance
> max_shared_var<-round(c(max(d[c(22,23),4]^2),max(d[c(22,24),4]^2),max(d[c(23,24),4]^
2)),3)
> data.frame(factorscore,reliability,average_var_extracted,max_shared_var)

  factorscore reliability average_var_extracted max_shared_var
1       sotru       0.690                 0.427          0.308
2       truin       0.816                 0.530          0.308
3        webe       0.816                 0.427          0.082
```

The fit measures indicate that the model is rejected by an absolute goodness of fit test, i.e. the fit of the model is significantly lower than for a perfectly fitting model (chisquare= 1526.049, df=62, p<.001). Furthermore, descriptive fit measures also indicate that the model cannot reproduce the observed covariance matrix well: CFI (.912) and TLI (.889) both are lower than .95 and hence do not meet the cutoff of good fit. RMSEA (.076) and SRMR (.04) indicate a good fit as they are below 0.08. Given these results, it can be argued that further modifications to the model are needed.

As can be seen in the standardized solution, all variables have significant and positive standardized loadings. Note that there are only 4 variables having loadings which exceed 0.7. Hence, the square of these loadings i.e. the individual reliabilities are larger than 0.5 only for these 4 variables. This indicates that the other variables do not have sufficient reliability and therefore convergent validity is not satisfied for these other variables in the model. Correlations between the factors (.555, .287, .185) show that they are poorly correlated. Furthermore, divergent validity is satisfied for all latent variables. Using the criterion of Fornell and Lanker to assess divergent validity, it is also confirmed as for each latent variable, the average variance extracted in the observed indicator variables is larger than the maximum variance that is shared with other latent variables except social trust factor since the scores are almost the same.

Finally, we see that composite reliability of the factor scores is not good, but still acceptable as they are .690 and .816.

Question B

To improve our model, we can use the ‘modificationIndices()’ function to get an idea of which error terms correlation we can add to improve our model.

```
> modificationindices(fitcfa1)

       lhs op     rhs      mi    epc sepc.lv sepc.all sepc.nox

...
sotru3 ~~   webe5   5.836 -0.044  -0.044   -0.046   -0.04691
sotru3 ~~   webe6   1.311 -0.024  -0.024   -0.021   -0.02192
truin1 ~~ truin2  52.242 -0.492  -0.492   -0.210   -0.21093
truin1 ~~ truin3 211.717 -0.854  -0.854   -0.326   -0.32694
truin1 ~~ truin4 559.300  1.707   1.707    0.914    0.91495
```

```
truin1 ~~  webe1   0.319  0.008   0.008    0.011    0.01196
truin1 ~~  webe2   2.238  0.021   0.021    0.030    0.03097
truin1 ~~  webe3   2.379 -0.026  -0.026   -0.030   -0.03098
truin1 ~~  webe4   2.427 -0.025  -0.025   -0.032   -0.03299
truin1 ~~  webe5   1.698  0.022   0.022    0.026    0.026100
truin1 ~~  webe6   0.003  0.001   0.001    0.001    0.001101
truin2 ~~ truin3 478.146  1.275   1.275    0.430    0.430102
truin2 ~~ truin4 168.787 -0.834  -0.834   -0.395   -0.395103
truin2 ~~  webe1   0.000  0.000   0.000    0.000    0.000104
truin2 ~~  webe2   0.671  0.013   0.013    0.015    0.015105
truin2 ~~  webe3   1.245  0.020   0.020    0.020    0.020106
truin2 ~~  webe4   0.004  0.001   0.001    0.001    0.001107
truin2 ~~  webe5   0.006  0.001   0.001    0.001    0.001108
truin2 ~~  webe6   0.196  0.009   0.009    0.008    0.008109
truin3 ~~ truin4  73.909 -0.471  -0.471   -0.199   -0.199110
truin3 ~~  webe1   2.761 -0.028  -0.028   -0.029   -0.029111
truin3 ~~  webe2   0.001  0.000   0.000    0.000    0.000112
truin3 ~~  webe3   1.206  0.021   0.021    0.019    0.019113
truin3 ~~  webe4   3.041  0.032   0.032    0.032    0.032114
truin3 ~~  webe5   0.204  0.009   0.009    0.008    0.008115
truin3 ~~  webe6   0.967  0.021   0.021    0.017    0.017116
truin4 ~~  webe1   1.231  0.015   0.015    0.022    0.022117
truin4 ~~  webe2   0.493 -0.009  -0.009   -0.014   -0.014118
truin4 ~~  webe3   0.022  0.002   0.002    0.003    0.003119
truin4 ~~  webe4   3.942 -0.030  -0.030   -0.041   -0.041120
truin4 ~~  webe5   3.947 -0.031  -0.031   -0.040   -0.040121
truin4 ~~  webe6   0.012 -0.002  -0.002   -0.002   -0.002122
webe1 ~~   webe2 153.517  0.068   0.068    0.255    0.255123
webe1 ~~   webe3  50.754  0.045   0.045    0.137    0.137124
webe1 ~~   webe4  62.404 -0.051  -0.051   -0.173   -0.173125
webe1 ~~   webe5  68.018 -0.054  -0.054   -0.171   -0.171126
webe1 ~~   webe6   5.876 -0.017  -0.017   -0.047   -0.047127
webe2 ~~   webe3 128.133  0.068   0.068    0.219    0.219128
webe2 ~~   webe4  57.439 -0.047  -0.047   -0.168   -0.168129
webe2 ~~   webe5  86.209 -0.058  -0.058   -0.194   -0.194130
webe2 ~~   webe6  23.523 -0.033  -0.033   -0.094   -0.094131
webe3 ~~   webe4 114.875 -0.075  -0.075   -0.219   -0.219132
webe3 ~~   webe5 150.356 -0.088  -0.088   -0.239   -0.239133
webe3 ~~   webe6  33.002  0.045   0.045    0.105    0.105134
webe4 ~~   webe5 612.318  0.182   0.182    0.553    0.553135
webe4 ~~   webe6   0.199  0.004   0.004    0.009    0.009136
webe5 ~~   webe6   3.493  0.015   0.015    0.037    0.037
```

Based on this output we included every error term correlation that lowers the chi square by at least 100, since this is a relatively large improvement in comparison to chisq = 1526.049. This yields the following model:

```
> cfa2<-'sotru =~NA*sotru1+sotru2+sotru3
+        truin =~NA*truin1+truin2+truin3+truin4
+        webe  =~NA*webe1+webe2+webe3+webe4+webe5+webe6
+        sotru ~~1*sotru
+        truin ~~1*truin
+        webe  ~~1*webe
+        truin1 ~~ truin3
+        truin1 ~~ truin4
+        truin2 ~~ truin3
+        truin2 ~~ truin4
+        webe1 ~~ webe2
+        webe2 ~~ webe3
+        webe3 ~~ webe4
+        webe3 ~~ webe5
+        webe4 ~~ webe5'
> #fit model on covariance matrix
> fitcfa2<-cfa(cfa2,sample.cov=covmat,sample.nobs=4046)
> fitmeasures(fitcfa2,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"))
  chisq      df  pvalue     cfi     tli   rmsea    srmr
169.929  53.000   0.000   0.993   0.990   0.023   0.017
```

Looking at the fit measurements we see indeed an improvement. CFI and TLI are above 0.95 and RMSEA and SRMR are below 0.08. The chi-square test is still significantly different from the perfectly fitted model (chi-square= 169.929, df=53, p<.001). This is probably due to the large number of observations in the dataset. We can therefore conclude that these additions add enough value to be included in the final model. This is also confirmed with the LR test below (LR= 1356.1, df=9, p<.001).

```
> anova(fitcfa1,fitcfa2)

Chi-Squared Difference Test

        Df    AIC    BIC   Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)    
fitcfa2 53 164265 164504  169.93                                          
fitcfa1 62 165603 165786 1526.05     1356.1 0.19234       9  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comparing the standardized estimates of cfa2 and cfa1. Overall, we can see some differences between the two models but what stands out the most is that only three estimates are above 0.7. This is also reflected in the difference in composite reliability as this is decreased for Trust institution and wellbeing. However, the reliability is above 0.7 for Trust institution and well-being and almost 0.7 social trust.

	cfa2 (est.std)	**cfa1 (est.std)**
sotru =~ sotru1	0.681	0.648
sotru =~ sotru2	0.648	0.626
sotru =~ sotru3	0.629	0.626
truin =~ truin1	0.713	0.789
truin =~ truin2	0.750	0.718
truin =~ truin3	0.628	0.581
truin =~ truin4	0.689	0.802
webe =~ webe1	0.623	0.661
webe =~ webe2	0.613	0.670
webe =~ webe3	0.691	0.589
webe =~ webe4	0.703	0.718
webe =~ webe5	0.657	0.677
webe =~ webe6	0.602	0.595
truin1 ~~ truin3	-0.121	
truin1 ~~ truin4	0.386	
truin2 ~~ truin3	0.200	
truin2 ~~ truin4	0.030	
webe1 ~~ webe2	0.232	
webe2 ~~ webe3	0.096	
webe3 ~~ webe4	-0.266	
webe3 ~~ webe5	-0.270	
webe4 ~~ webe5	0.307	

Factorscore	Composite reliability cfa2	Composite reliability cfa1
sotru	0.690	0.690
truin	0.789	0.816
webe	0.813	0.816

Looking at the correlations between the error terms they are all significant (p<0.001). Overall, the correlations between the different variables are positively correlated except for truin1 (Trust in the country's parliament) and truin3 (Trust in the police) are negatively correlated. This value ranges from 0.030 to 0.386. For well-being, all terms have a positive correlation ranging from 0.096 to 0.307.

Question C

For this question we use the same model from the previous question (cfa2) and transform it to four multi-group structural equation models using sem1 (which lets the coefficient of the regressions range freely) and sem2 (which constraints the coefficients to be equal across groups (countries: FR and GB)).

```
> sem1<-'sotru =~NA*sotru1+sotru2+sotru3

+         truin =~NA*truin1+truin2+truin3+truin4
+         webe  =~NA*webe1+webe2+webe3+webe4+webe5+webe6
+         sotru ~~1*sotru
+         truin ~~1*truin
+         webe  ~~1*webe
+         truin1 ~~ truin3
+         truin1 ~~ truin4
+         truin2 ~~ truin3
+         truin2 ~~ truin4
+         webe1 ~~ webe2
+         webe2 ~~ webe3
+         webe3 ~~ webe4
+         webe3 ~~ webe5
+         webe4 ~~ webe5
+         webe ~ sotru + truin'
>
> sem2<-'sotru =~NA*sotru1+sotru2+sotru3
+         truin =~NA*truin1+truin2+truin3+truin4
+         webe  =~NA*webe1+webe2+webe3+webe4+webe5+webe6
+         sotru ~~1*sotru
+         truin ~~1*truin
+         webe  ~~1*webe
+         truin1 ~~ a*truin3
+         truin1 ~~ b*truin4
+         truin2 ~~ c*truin3
+         truin2 ~~ d*truin4
+         webe1 ~~ e*webe2
+         webe2 ~~ f*webe3
+         webe3 ~~ g*webe4
+         webe3 ~~ h*webe5
+         webe4 ~~ i*webe5
+         webe ~ j*sotru + k*truin'

> # Configural measurement invariance model with country-specific regression
> config1 <- sem(sem1, data = ess, group = "cntry")
> # Configural measurement invariance model with country-specific regression and equal
ity constraints
> config2 <- sem(sem2, data = ess, group = "cntry")
> # Metric measurement invariance model with country-specific regression
> metric1 <- sem(sem1, data = ess, group = "cntry", group.equal="loadings")
> # Metric measurement invariance model with country-specific regression and equality
constraints
> metric2 <- sem(sem2, data = ess, group = "cntry", group.equal="loadings")
```

```
> # Fit measures
> fitconfig1 <- fitmeasures(config1,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fitconfig2 <- fitmeasures(config2,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fitmetric1 <- fitmeasures(metric1,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fitmetric2 <- fitmeasures(metric2,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fit<-rbind(fitconfig1,fitconfig2,fitmetric1,fitmetric2)
> rownames(fit)<-c("config1","config2","metric1","metric2")
> round(fit,3)
          chisq  df pvalue   cfi   tli rmsea  srmr      aic      bic
config1 218.858 106      0 0.993 0.990 0.023 0.020 163211.0 163854.2
config2 259.497 117      0 0.991 0.989 0.025 0.022 163229.6 163803.4
metric1 261.703 119      0 0.991 0.989 0.024 0.027 163227.9 163789.0
metric2 311.430 130      0 0.989 0.987 0.026 0.031 163255.6 163747.4
```

Looking at the fitmeasures we can conclude that the best model to fit the data is config1. This model has the lowest chi-square, this is still different from the perfectly, fitted model, both CFI and TLI are the highest and RMSEA and SRMR are the lowest. Looking at the AIC and BIC again config1 yields the lowest value, indicating this is the best fitting model. ever, the results are remarkably close and overall, the models could be a fit for our data (CFI and TLI >0.95, RMSEA and SRMR< 0.08).
Using The LR test we can conclude that all models are significantly different (p<0.001) except for config2 and metric1 (LR: 2.2057, df=2, p=0.03319). Given the fact that config1 is the best model, we can say that it is best not to constrain the coefficient to be equal across the two countries.

Looking at the standardized solution of config1:

```
> standardizedSolution(config1)
     lhs op    rhs group est.std    se      z pvalue ci.lower ci.upper
1    sotru =~ sotru1     1   0.611 0.021 29.231  0.000    0.570    0.652
2    sotru =~ sotru2     1   0.686 0.020 33.767  0.000    0.646    0.726
3    sotru =~ sotru3     1   0.570 0.021 26.608  0.000    0.528    0.612
4    truin =~ truin1     1   0.688 0.024 28.458  0.000    0.640    0.735
5    truin =~ truin2     1   0.737 0.024 30.183  0.000    0.689    0.785
6    truin =~ truin3     1   0.603 0.028 21.815  0.000    0.549    0.657
7    truin =~ truin4     1   0.666 0.028 24.182  0.000    0.612    0.720
8     webe =~  webe1     1   0.639 0.018 34.716  0.000    0.603    0.675
9     webe =~  webe2     1   0.656 0.020 33.221  0.000    0.617    0.694
10    webe =~  webe3     1   0.692 0.021 32.594  0.000    0.650    0.734
11    webe =~  webe4     1   0.656 0.020 32.239  0.000    0.616    0.696
12    webe =~  webe5     1   0.616 0.021 29.024  0.000    0.575    0.658
13    webe =~  webe6     1   0.608 0.018 33.498  0.000    0.573    0.644
```

Overall, all the correlations with the latent variables are significant however all but one (`truin =~ truin2`) are below 0,7.

Looking at the coefficients of social trust and trust institution we see that in the first group, FR, social trust has a significant effect on well-being while this is not the case for trust institution. In the second group, GB, however, both predictors are significant in explaining well-being but here social trust has a bigger impact than trust institution.

```
> standardizedSolution(config1)
     lhs op    rhs group est.std    se      z pvalue ci.lower ci.upper
26    webe  ~  sotru     1   0.263 0.043  6.191  0.000    0.180    0.347
27    webe  ~  truin     1   0.040 0.042  0.955  0.340   -0.042    0.121
83    webe  ~  sotru     2   0.218 0.036  6.120  0.000    0.148    0.288
84    webe  ~  truin     2   0.070 0.035  1.996  0.046    0.001    0.139
```

```
> # Fit measures
```

```
> fitconfig1 <- fitmeasures(config1,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fitconfig2 <- fitmeasures(config2,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fitmetric1 <- fitmeasures(metric1,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fitmetric2 <- fitmeasures(metric2,c("chisq","df","pvalue","cfi","tli","rmsea","srmr"
,"aic","bic"))
> fit<-rbind(fitconfig1,fitconfig2,fitmetric1,fitmetric2)
> rownames(fit)<-c("config1","config2","metric1","metric2")
> round(fit,3)
          chisq  df pvalue   cfi   tli rmsea  srmr      aic      bic
config1 218.858 106      0 0.993 0.990 0.023 0.020 163211.0 163854.2
config2 259.497 117      0 0.991 0.989 0.025 0.022 163229.6 163803.4
metric1 261.703 119      0 0.991 0.989 0.024 0.027 163227.9 163789.0
metric2 311.430 130      0 0.989 0.987 0.026 0.031 163255.6 163747.4
```

Question D

We load the data with the original variable names, standardize the variables, use the candisc() procedure to conduct canonical correlation analysis and print a summary of the results and compute redundancies.

```
> load("ess.Rdata")
> sess<- ess
> sess[,2:14]<-scale(ess[,2:14],center=TRUE,scale=FALSE)

> cancor.out<-cancor(cbind(fltdpr, fltsd, fltanx, wrhpp, enjlf, fltpcfl)
+ ~ppltrst+ pplfair+ pplhlp+ trstprl+ trstlgl+ trstplc+ trstplt, data=sess)

> summary(cancor.out)
Canonical correlation analysis of:
         7   X  variables:  ppltrst, pplfair, pplhlp, trstprl, trstlgl, trstplc, trst
plt
  with   6   Y  variables:  fltdpr, fltsd, fltanx, wrhpp, enjlf, fltpcfl

      CanR    CanRSQ     Eigen  percent    cum                          scree
1 0.242629 5.887e-02 6.255e-02 77.37503  77.38 ******************************
2 0.110279 1.216e-02 1.231e-02 15.22875  92.60 ******
3 0.063206 3.995e-03 4.011e-03  4.96159  97.57 **
4 0.041142 1.693e-03 1.696e-03  2.09741  99.66 *
5 0.016167 2.614e-04 2.614e-04  0.32339  99.99
6 0.003343 1.118e-05 1.118e-05  0.01383 100.00

Test of H0: The canonical correlations in the
current row and all that follow are zero

      CanR LR test stat approx F numDF denDF   Pr(> F)
1 0.242629      0.92415   7.6396    42 18920 < 2.2e-16 ***
2 0.110279      0.98196   2.4539    30 16138 1.618e-05 ***
3 0.063206      0.99405   1.2056    20 13384    0.2378
4 0.041142      0.99804   0.6617    12 10678    0.7897
5 0.016167      0.99973   0.1834     6  8074    0.9815
6 0.003343      0.99999      NaN     2   NaN       NaN
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> #redundancies
> redu<-redundancy(cancor.out)
> round(redu$Xcan,3)
Xcan1 Xcan2 Xcan3 Xcan4 Xcan5 Xcan6
0.023 0.001 0.001 0.000 0.000 0.000
> round(redu$Ycan,3)
Ycan1 Ycan2 Ycan3 Ycan4 Ycan5 Ycan6
0.030 0.002 0.000 0.000 0.000 0.000

> #computation redundancies from output
> R2tu<-cancor.out$cancor^2
```

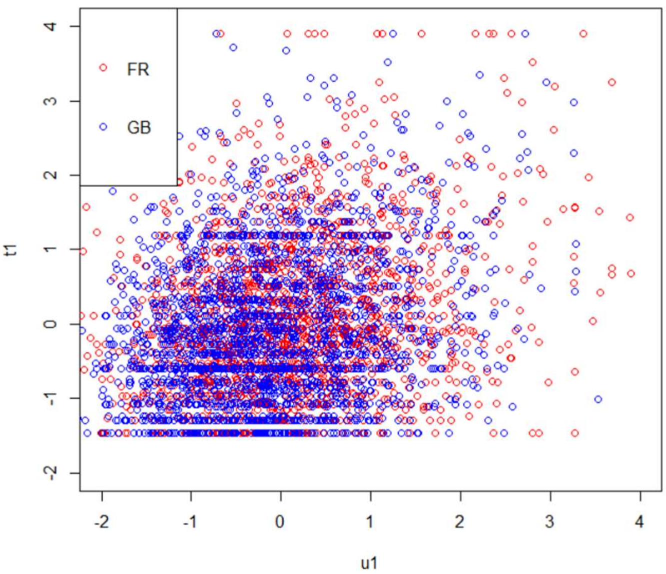
```
> VAFYbyt<-apply(cancor.out$structure$Y.yscores^2,2,sum)/4
> redund<-R2tu*VAFYbyt
> round(cbind(R2tu,VAFYbyt,redund,total=cumsum(redund)),3)
       R2tu VAFYbyt redund total
Ycan1 0.059   0.770  0.045 0.045
Ycan2 0.012   0.192  0.002 0.048
Ycan3 0.004   0.132  0.001 0.048
Ycan4 0.002   0.131  0.000 0.048
Ycan5 0.000   0.154  0.000 0.048
Ycan6 0.000   0.122  0.000 0.048
```

The canonical correlation analysis extracts 6 pairs of canonical variances. Hypotheses tests indicate that only the first two correlations are significant i.e., H0: corr(u3,t3)=0 cannot be rejected at the 5% level (p= 0.2378).

The first canonical correlation equals 0.24. This means that the canonical variate u1 accounts for 5.89% of the variance in the canonical variate t1. The second canonical correlation equals 0.11. This means that the canonical variate u2 accounts for 1.21% of the variance in the canonical variate t2.

Looking at redundancies, we observe that u1 accounts for 3% variance in Y and u2 accounts for 0.2% variance in Y. Since only the first two correlations are significant, we can say that X variables account for 3.2% of variance in the Y variables. The u2 barely contributed. Only a small portion of variance in Y is explained by X.

In addition, we make a scatter plot of the first pair of canonical variates and indicate a different color for observations of each country.



Question E

To assess the validity of the analysis, we used a split-half approach.

```
samplesize<-dim(ess)[1]
train<-ess[seq(2,samplesize,by=2),2:14]
valid<-ess[seq(1,samplesize,by=2),2:14]
train<-as.data.frame(scale(train,center=TRUE,scale=TRUE))
valid<-as.data.frame(scale(valid,center=TRUE,scale=TRUE))
```

```
#conduct CCA on training data
cancor.train<-cancor(cbind(fltdpr, fltsd, fltanx, wrhpp, enjlf, fltpcfl)
~ppltrst+ pplfair+ pplhlp+ trstprl+ trstlgl+ trstplc+ trstplt, data=train)
#summary(cancor.train)

round(cancor.train$structure$X.xscores,3)
round(cancor.train$structure$Y.yscores,3)

#conduct CCA on validation data
cancor.valid<-cancor(cbind(fltdpr, fltsd, fltanx, wrhpp, enjlf, fltpcfl)
~ppltrst+ pplfair+ pplhlp+ trstprl+ trstlgl+ trstplc+ trstplt, data=valid)

#summary(cancor.valid)
round(cancor.valid$structure$X.xscores,3)
round(cancor.valid$structure$Y.yscores,3)

# canonical variates calibration set
train.X1<-cancor.train$score$X
train.Y1<-cancor.train$score$Y

# compute canonical variates using data of calibration set and coefficients estimated
on validation set
train.X2<-as.matrix(train[,1:7])%*%cancor.valid$coef$X
train.Y2<-as.matrix(train[,8:13])%*%cancor.valid$coef$Y
```

```
> #R(T,T*) and R(U,U*) for t1,t2,u1,u2
> round(cor(train.Y1,train.Y2)[1:2,1:2],3)
      Ycan1  Ycan2
Ycan1 0.989 -0.111
Ycan2 0.101  0.817
> round(cor(train.X1,train.X2)[1:2,1:2],3)
      Xcan1  Xcan2
Xcan1 0.982 -0.042
Xcan2 0.029  0.514
```

The absolute value of the diagonal elements of R(T,T*) and R(U,U*) represent the reliabilities of the canonical variates for Y and X variables. The reliabilities of t1, t2 equal .989, .817. And the reliabilities of u1, u2 equal .982, .514. In other words, the first pairs of canonical variates have excellent reliability, but the reliability of u2 is unacceptable. The off-diagonal correlations are low.

```
> #R(U,T) and R(U*,T*)
> round(cor(train.X1,train.Y1)[1:2,1:2],3)
      Ycan1 Ycan2
Xcan1 0.253 0.000
Xcan2 0.000 0.129
> round(cor(train.X2,train.Y2)[1:2,1:2],3)
       Ycan1  Ycan2
Xcan1  0.246 -0.028
Xcan2 -0.002  0.044
```

A comparison of R(U*,T*) and R(U,T) shows that R(u1, t1) 0.253 is only a little higher than R(u1*, t1*) 0.246. In other words, overestimation of the first canonical correlation due to the maximization involved is not an issue. Yet, the overestimation in the second canonical correlation is rather large (.129 versus .044).

```
> #R(T*,T*) and R(U*,U*)
> round(cor(train.Y2,train.Y2)[1:2,1:2],3)
       Ycan1  Ycan2
Ycan1  1.000 -0.019
Ycan2 -0.019  1.000
> round(cor(train.X2,train.X2)[1:2,1:2],3)
       Xcan1  Xcan2
Xcan1  1.000 -0.007
Xcan2 -0.007  1.000
```

The off-diagonal elements of R(T*,T*) and R(U*,U*) are close to 0, which indicates that canonical variates of Y variables and of X variables computed on calibration data but based on the coefficients from validation data have as expected correlations that are close to 0 (canonical variates are independent).

Question F

From the redundancies in previous results, we can conclude that the first two pairs of canonical variates have very good reliabilities. The redundancy analysis has shown that u1 accounts for 4.5% of the variance in the Y variables, and that u2 accounts only for an additional 0.2% of the variance in the Y variables. As the second pair of canonical variates is not practically important, we focus for the interpretation on the first pair of canonical variates.

```
> as.matrix(round(cancor.out$structure$X.xscores[,1],3))
          [,1]
ppltrst -0.676
pplfair -0.842
pplhlp  -0.609
trstprl -0.544
trstlgl -0.599
trstplc -0.554
trstplt -0.441
```

These canonical loadings show u1 has negative associations with variables related to social trust and trust in institutions (correlations for ppltrst, pplfair, pplhlp, trstprl, trstlgl, trstplc, trstplt are all negative, with social trust variables have higher correlations). This indicates that people who score lower on u1 have more trust in society and institutions.

```
> as.matrix(round(cancor.out$structure$Y.yscores[,1],3))
          [,1]
fltdpr  -0.740
fltsd   -0.687
fltanx  -0.677
wrhpp   -0.749
enjlf   -0.793
fltpcfl -0.644
```

These canonical loadings show t1 has negative associations with variables related to well-being, including positive emotions and inverted negative emotions (correlations for fltdpr, fltsd, fltanx, wrhpp, enjlf, fltpcfl are all highly negative). This indicates that people who score lower on t1 have more negative emotions.

Hence, the positive correlation between u1 and t1 means that persons who are experiencing more positive emotions and less negative emotions would also have higher trust in institutions and on society.