1 TSFA Guide

This guide illustrates the steps for estimating a factor model using as an example the data and process which led to results reported in Gilbert and Meijer (2006). The background theory is reported in Gilbert and Meijer (2005).

Plots and some output from the examples below are omitted to save paper. The functions in the *tsfa* package are made available with

```
> library("tsfa")
```

The code from the vignette that generates this guide can be loaded into an editor with *edit(vignette("tsfa"))*. This uses the default editor, which can be changed using *options()*. In some examples the code runs into the margins and is truncated. (This is a limitation of the current version of RSweave.) If in doubt about the code, please edit the vignette as above or consult its source, which is distributed in the file tsfa/inst/doc/tsfa.Stex. It should also be possible to view the pdf version of the guide for this package with *print(vignette("tsfa"))*.

The data is converted to real per capita data as follows

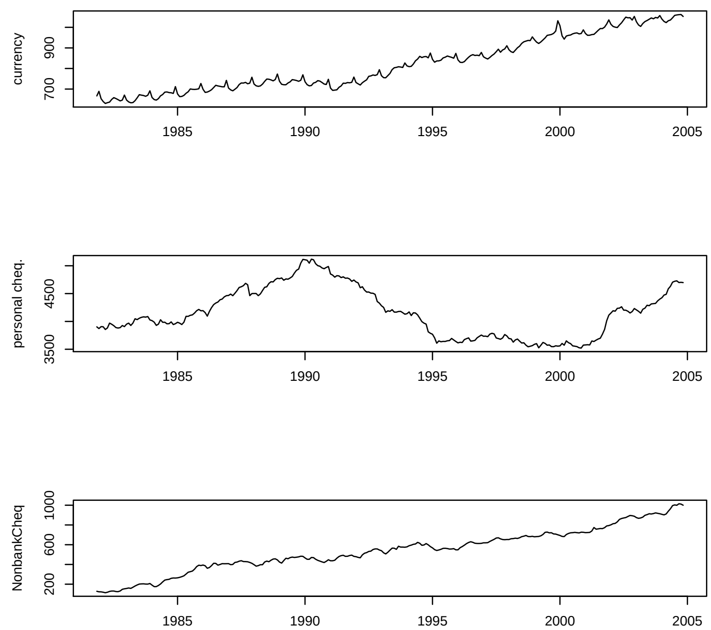
```
> data("CanadianMoneyData.asof.28Jan2005", package = "CDNmoney")
> data("CanadianCreditData.asof.28Jan2005", package = "CDNmoney")
> cpi <- 100 * M1total/M1real
> seriesNames(cpi) <- "CPI"
> popm <- M1total/M1PerCapita
> seriesNames(popm) <- "Population of Canada"
> z <- tframed(tbind(MB2001, MB486 + MB452 + MB453, NonbankCheq,
      MB472 + MB473 + MB487p, MB475, NonbankNonCheq + MB454 + NonbankTerm +
          MB2046 + MB2047 + MB2048 + MB2057 + MB2058 + MB482),
      names = c("currency", "personal cheq.", "NonbankCheq", "N-P demand & notice",
          "N-P term", "Investment"))
> TotalMoney <- tframed(rowSums(z), tframe(z))
> z <- tbind(z, ConsumerCredit, ResidentialMortgage, ShortTermBusinessCredit,
      OtherBusinessCredit)
```

Investment series goes back only to November 1981 and some of the data ends in November 2004, so the data is truncated to that time window.

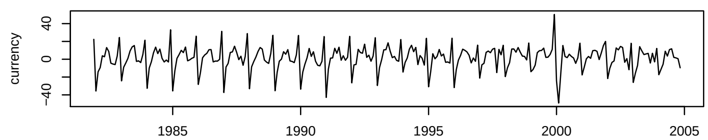
```
> z <- tfwindow(z, start = c(1981, 11), end = c(2004, 11))
> scale <- tfwindow(1e+08/(popm * cpi), tf = tframe(z))
> MBandCredit <- sweep(z, 1, scale, "*")
```

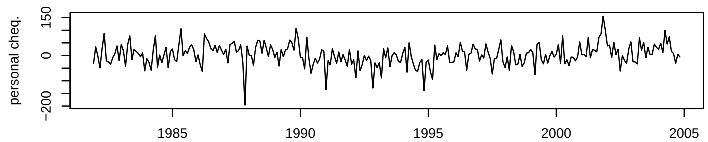
Multiplying by $1e8$ gives real dollars per person. (Credit aggregates, B and MB numbers are in millions, CPI is in percentage points, popm is in units.) Plots to check the data can be generated with

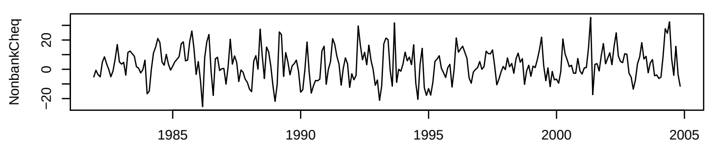
```
> tfplot(MBandCredit, graphs.per.page = 3)
```




```
> tfplot(diff(MBandCredit), graphs.per.page = 3)
```







Various sample statistics are checked with

```
> start(MBandCredit)
```

```
[1] 1981   11
```

```
> end(MBandCredit)
```

```
[1] 2004   11
```

```
> Tobs(MBandCredit)
```

```
[1] 277
```

```
> DX <- diff(MBandCredit, lag = 1)
```

The number of observations is

```
> Tobs(MBandCredit)
```

```
[1] 277
```

The number of series is

```
> nseries(MBandCredit)
```

```
[1] 10
```

The means of differenced series are

```
> colMeans(DX)

                 currency            personal cheq.
                 1.398385                  2.880048
              NonbankCheq       N-P demand & notice
                 3.157783                  9.156883
                 N-P term                Investment
                 4.926683                 48.744141
          Consumer Credit      Residential Mortgage
                12.703423                 29.878367
Short Term Business Credit      Other Business Credit
                -8.066348                 37.283246
```

The standard deviations of differenced series are

```
> sqrt(diag(cov(DX)))

                 currency            personal cheq.
                 13.46788                  42.94902
              NonbankCheq       N-P demand & notice
                 10.74835                  46.04252
                 N-P term                Investment
                 71.93040                 116.98518
          Consumer Credit      Residential Mortgage
                 30.00178                  50.32996
Short Term Business Credit      Other Business Credit
                 71.83893                  56.30949
```

2 Checking for the number of factors

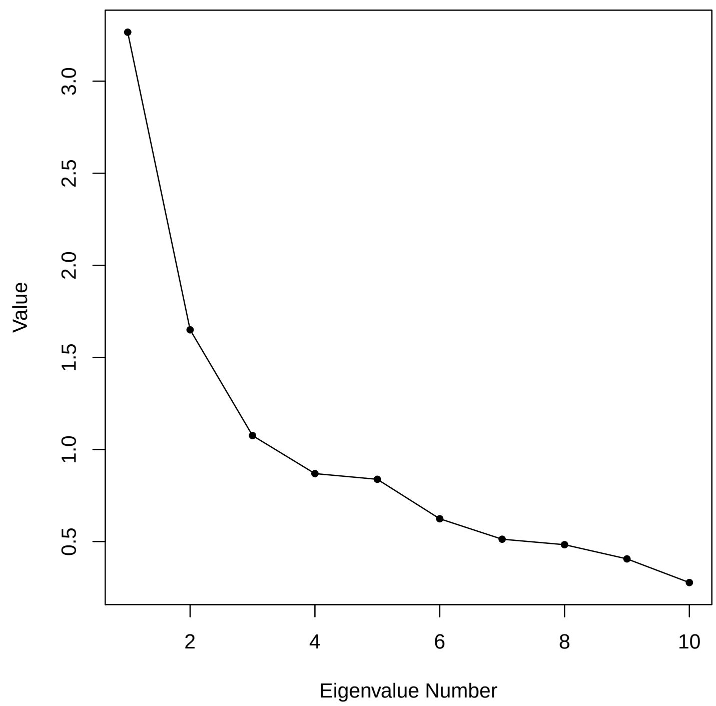
Eigenvalues for the scree plot are

```
> zz <- eigen(cor(diff(MBandCredit, lag = 1)), symmetric = TRUE)[["values"]]
> print(zz)

 [1] 3.2659246 1.6498244 1.0754761 0.8689273 0.8380089 0.6236201 0.5125156
 [8] 0.4828785 0.4056941 0.2771303
```

The scree plot (Figure 1) is generated by

```
> par(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(4.1, 4.1, 0.6, 0.1))
> plot(zz, ylab = "Value", xlab = "Eigenvalue Number", pch = 20:20,
      cex = 1, type = "o")
```



FAfitStats is used to calculate the fit statististics for different numbers of factors (Table 1).

```
> z <- FAfitStats(MBandCredit)
> print(z, digits = 3)

$fitStats
              0        1        2        3        4        5       6 saturated
chisq  7.22e+02 2.52e+02 1.26e+02 3.79e+01  17.8435   8.5095   1.217     0.000
df     4.50e+01 3.50e+01 2.60e+01 1.80e+01  11.0000   5.0000   0.000     0.000
pval  8.88e-123 1.38e-34 3.93e-15 4.01e-03   0.0853   0.1303   0.000        NA
delta  6.77e+02 2.17e+02 1.00e+02 1.99e+01   6.8435   3.5095   1.217     0.000
RMSEA  2.33e-01 1.50e-01 1.18e-01 6.33e-02   0.0475   0.0504     Inf        NA
RNI    0.00e+00 6.80e-01 8.52e-01 9.71e-01   0.9899   0.9948   0.998     1.000
CFI    0.00e+00 6.80e-01 8.52e-01 9.71e-01   0.9899   0.9948   0.998     1.000
MCI    2.94e-01 6.75e-01 8.34e-01 9.65e-01   0.9877   0.9937   0.998     1.000
GFI    5.84e-01 8.33e-01 9.13e-01 9.75e-01   0.9881   0.9941   0.999     1.000
AGFI   4.91e-01 7.37e-01 8.16e-01 9.22e-01   0.9406   0.9347    -Inf     1.000
AIC    6.32e+02 1.82e+02 7.43e+01 1.89e+00  -4.1565  -1.4905   1.217     0.000
CAIC   7.88e+02 3.84e+02 3.18e+02 2.83e+02 309.1411 339.5296 365.339   364.122
SIC    7.78e+02 3.64e+02 2.89e+02 2.46e+02 265.1411 289.5296 310.339   309.122
CAK    2.69e+00 1.06e+00 6.68e-01 4.05e-01   0.3835   0.3932   0.403     0.399
CK     2.69e+00 1.06e+00 6.76e-01 4.17e-01   0.3967   0.4082   0.420     0.415
```

```
$seqfitStats
        0 vs 1   1 vs 2   2 vs 3   3 vs 4 4 vs 5 5 vs 6 6 vs saturated
chisq 4.70e+02 1.25e+02 8.84e+01 20.04248  9.334  7.293           1.22
df    1.00e+01 9.00e+00 8.00e+00  7.00000  6.000  5.000           0.00
pval  1.15e-94 1.08e-22 9.91e-16  0.00548  0.156  0.200           0.00
```

```
> c2withML <- estTSF.ML(MBandCredit, 2)
```

The sign and order of factors is arbitary. For simulation and estimation comparisons it is useful to put them in the same order when different results are compared. This is done by specifying the *BpermuteTarget* argument in *estTSF.ML*. Other than the sign, this does not affect the estimated values, it only rearranges their order. Here they are arranged so 1=transactions, 2=long term, 3=potential spending, 4=consumer credit , 5=N-P term (which is the order they appear as factors are added). The *BpermuteTarget, z* below was determined by an initial run.

```
> z <- matrix(0, 10, 3)
> z[matrix(c(1, 6, 2, 1:3), 3, 2)] <- c(10, 56, 41)
> c3withML <- estTSF.ML(MBandCredit, 3, BpermuteTarget = z)
> z <- matrix(0, 10, 4)
> z[matrix(c(1, 6, 2, 7, 1:4), 4, 2)] <- c(13, 54, 37, 24)
> c4withML <- estTSF.ML(MBandCredit, 4, BpermuteTarget = z)
> z <- matrix(0, 10, 5)
> z[matrix(c(1, 6, 2, 7, 5, 1:5), 5, 2)] <- c(13, 67, 34, 30, 72)
> c5withML <- estTSF.ML(MBandCredit, 5, BpermuteTarget = z)
```

The standardized loadings for the four factor model (Table 2) are

```
> print(DstandardizedLoadings(c4withML))
```

```
                             Factor 1    Factor 2    Factor 3    Factor 4
currency                   1.01983211  0.09441862 -0.09386876 -0.08001695
personal cheq.            -0.02898840  0.20319639  0.87632955 -0.07894600
NonbankCheq                0.20214387 -0.13884613  0.36792951  0.11966332
N-P demand & notice        0.55851463 -0.00326414  0.20169780  0.24090075
N-P term                   0.02120284  0.10107575 -0.23481508  0.33307726
Investment                 0.15013102  0.45888993 -0.23944032  0.06862474
Consumer Credit            0.01841197  0.10508972  0.10142580  0.79551987
Residential Mortgage       0.21597515  0.49350633  0.22622020  0.14854196
Short Term Business Credit -0.09041731  0.23970572 -0.01376311  0.09703765
Other Business Credit      0.09632589  0.66331904  0.23994908  0.04224045
```

The estimated Φ matrix is

```
> print(c4withML$Phi, digits = 3)
```

```
        Factor1 Factor2 Factor3 Factor4
Factor1   1.000  0.1783  0.3003  0.3400
Factor2   0.178  1.0000 -0.0245  0.5685
Factor3   0.300 -0.0245  1.0000  0.0124
Factor4   0.340  0.5685  0.0124  1.0000
```

Communalities for the 4 factor model are

```
> print(1 - c4withML$stats$uniquenesses)
```

```
                 currency           personal cheq.
               0.97758543               0.77183960
              NonbankCheq      N-P demand & notice
               0.24563691               0.56948987
                 N-P term               Investment
               0.21682570               0.34592814
          Consumer Credit     Residential Mortgage
               0.76283210               0.53132528
Short Term Business Credit    Other Business Credit
               0.08886927               0.57238659
```

Communalities for other models are given by the following, but output is omitted here.

```
> print(1 - c2withML$stats$uniquenesses)
> print(1 - c3withML$stats$uniquenesses)
> print(1 - c5withML$stats$uniquenesses)
```

Loadings for 4 factor model (Table 3) are

```
> print(loadings(c4withML))
```

```
                             Factor 1   Factor 2   Factor 3  Factor 4
currency                   13.7349771  1.2716188  -1.264213 -1.077659
personal cheq.             -1.2450233  8.7270851  37.637493 -3.390653
NonbankCheq                 2.1727138 -1.4923673   3.954637  1.286184
N-P demand & notice        25.7154238 -0.1502893   9.286676 11.091679
N-P term                    1.5251290  7.2704190 -16.890343 23.958381
Investment                 17.5631039 53.6833203 -28.010968  8.028077
Consumer Credit             0.5523918  3.1528783   3.042954 23.867010
Residential Mortgage       10.8700198 24.8381514  11.385653  7.476110
Short Term Business Credit -6.4954824 17.2202022  -0.988727  6.971081
Other Business Credit       5.4240616 37.3511568  13.511410  2.378538
```

Figure 2 is generated by

```
> tfplot(ytoypc(factors(c4withML)), Title = "Factors from 4 factor model (year-to-year growt
      lty = c("solid"), col = c("black"), xlab = c(""), ylab = c("factor 1",
          "factor 2", "factor 3", "factor 4"), par = list(omi = c(0.1,
          0.1, 0.1, 0.1), mar = c(3.1, 4.1, 0.6, 0.1)), reset.screen = TRUE)
```

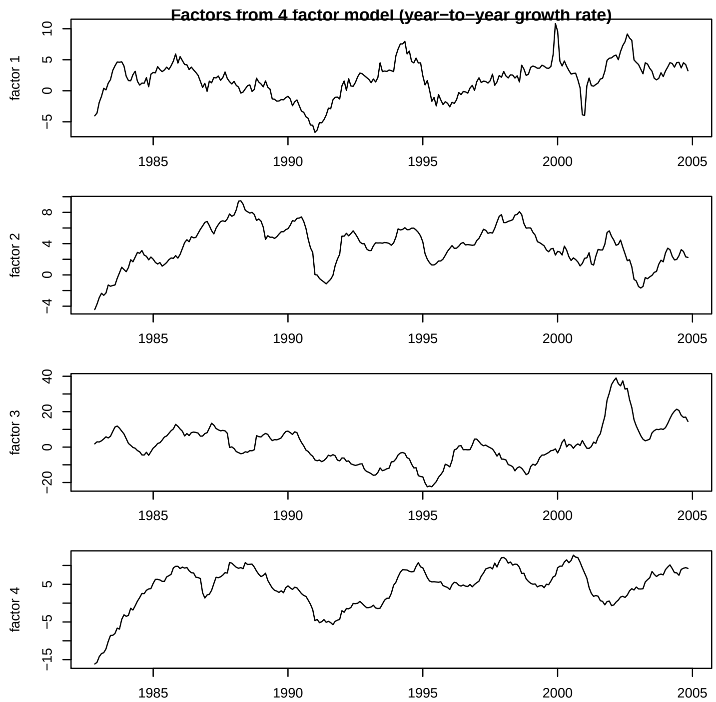
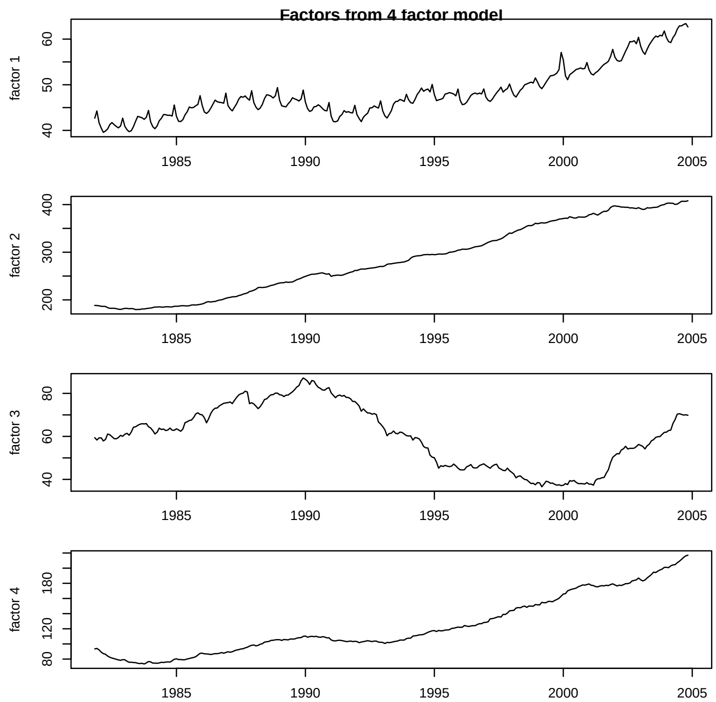


Figure 3 is generated by

```
> tfplot(factors(c4withML), Title = "Factors from 4 factor model",
      lty = c("solid"), col = c("black"), xlab = c(""), ylab = c("factor 1",
          "factor 2", "factor 3", "factor 4"), par = list(omi = c(0.1,
          0.1, 0.1, 0.1), mar = c(3.1, 4.1, 0.6, 0.1)), reset.screen = TRUE)
```



Some points are close to zero and cause problems in growth rate graphics. One solution is to set them to NA, but truncating the graphic works better. Figure 4 is generated by

```
> z <- explained(c4withML)
> tfplot(ytoypc(MBandCredit), ytoypc(z), series = 1:5, graphs.per.page = 5,
      lty = c("solid", "dashed"), col = c("black", "red"), ylab = c("currency",
          "personal cheq.", "NonbankCheq", "N-P demand & notice",
          "N-P term"), ylim = list(NULL, NULL, c(-100, 100), NULL,
          NULL), Title = "Explained money indicator 1-5 (year-to-year growth rate) using 4 f
      par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
          0.6, 0.1)), reset.screen = TRUE)
```

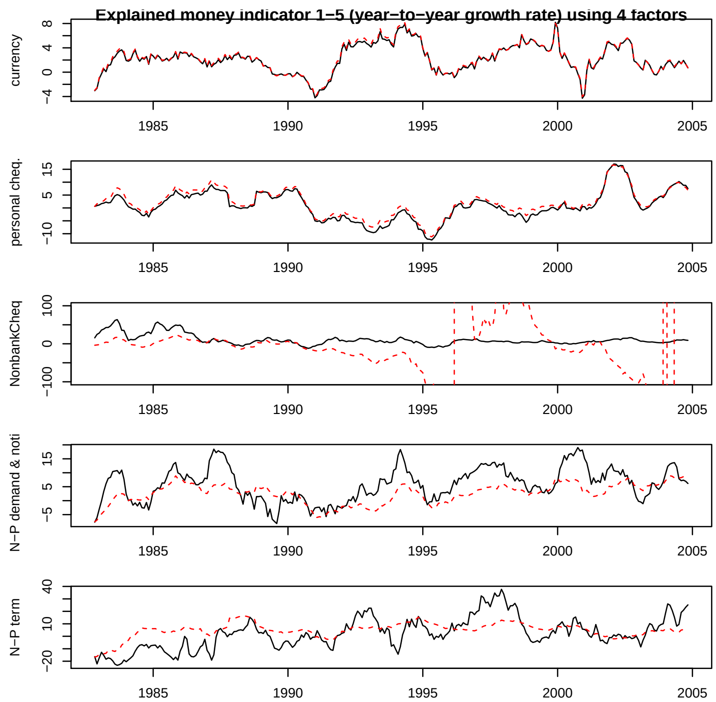


Figure 5 is generated by

```
> tfplot(ytoypc(MBandCredit), ytoypc(explained(c4withML)), series = 6:10,
     graphs.per.page = 5, lty = c("solid", "dashed"), col = c("black",
         "red"), ylab = c("", "", "", "", "", "Investment", "Consumer Credit",
         "Residential Mortgage", "Short Term Business Credit",
         "Other Business Credit"), Title = "Explained money indicator 6-10 (year-to-year gr
     par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
         0.6, 0.1)), reset.screen = TRUE)
```

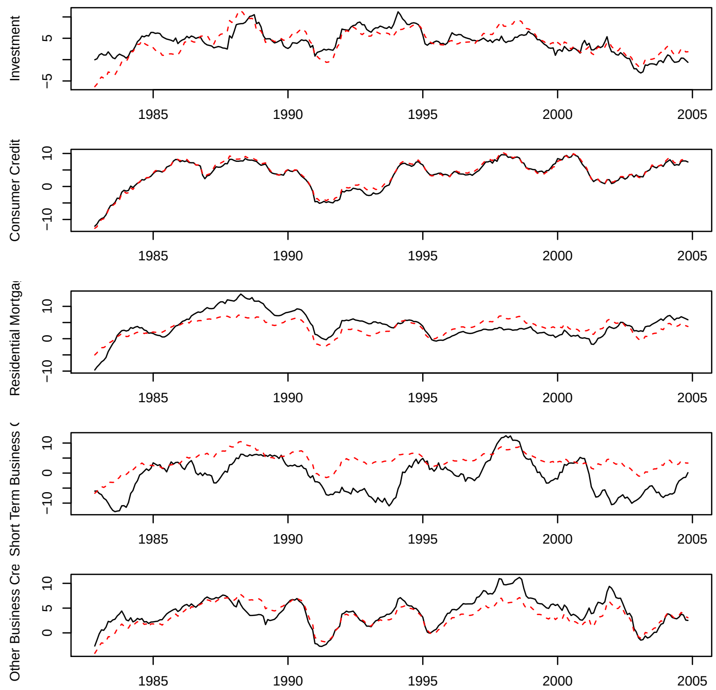


Figure 6 is generated by

```
> tfplot(MBandCredit, explained(c4withML), series = 1:5, graphs.per.page = 5,
      lty = c("solid", "dashed"), col = c("black", "red"), Title = "Explained money indicato
      par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
          0.6, 0.1)), reset.screen = TRUE)
```

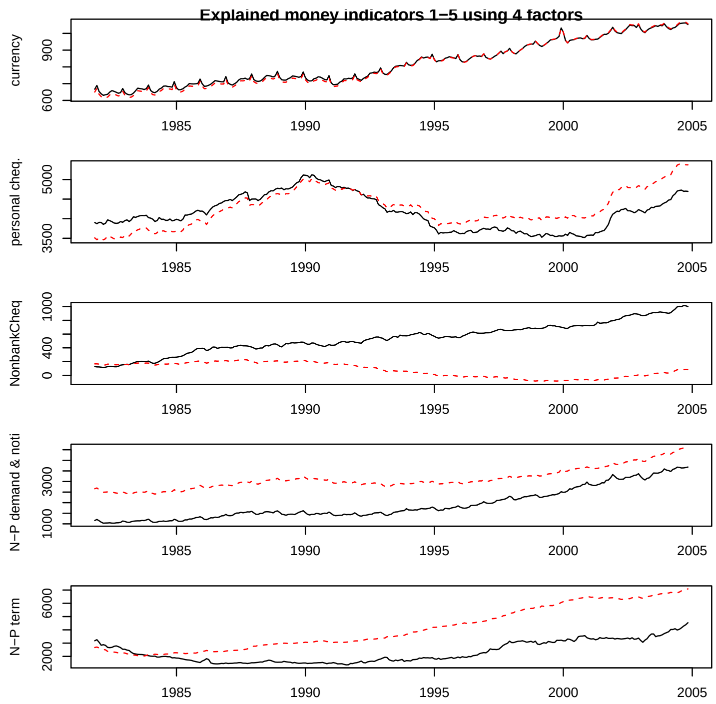
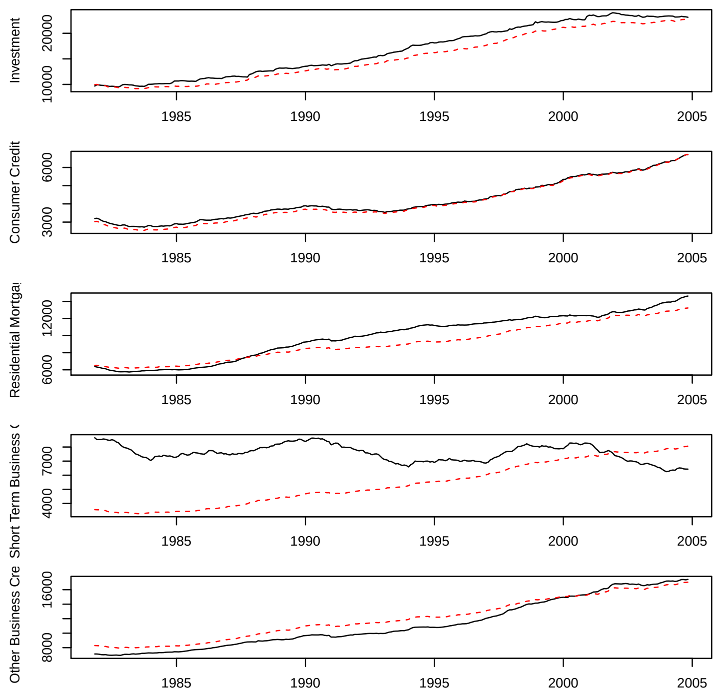


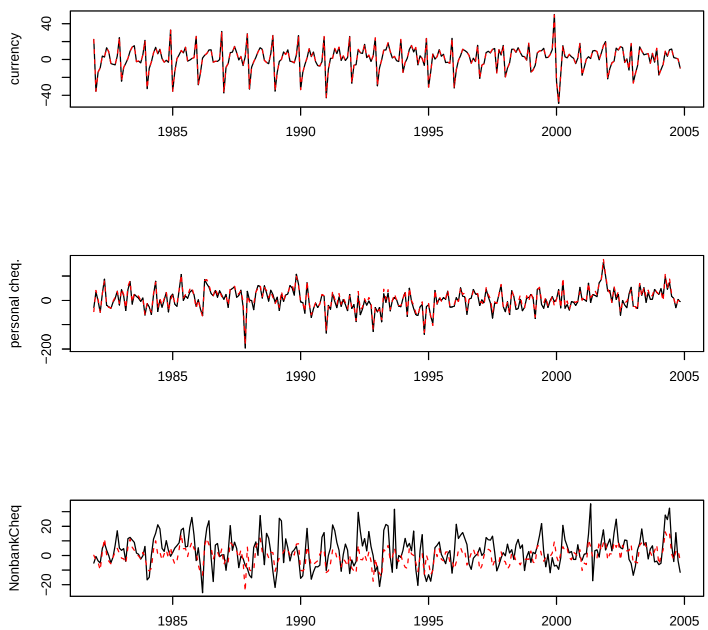
Figure 7 is generated by

```
> tfplot(MBandCredit, explained(c4withML), series = 6:10, graphs.per.page = 5,
     lty = c("solid", "dashed"), col = c("black", "red"), Title = "Explained money indicato
     par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
         0.6, 0.1)), reset.screen = TRUE)
```



The differenced version of the above is not in the paper, but is given by

```
> tfplot(diff(MBandCredit), diff(explained(c4withML)), graphs.per.page = 3)
```



Summary information about the model is calculated with

```
> summary(MBandCredit)
```

3 Two, three and five factors models

Tables of standardized loadings for differenced models, and the estimated Φ are given by the following (output is omitted).

```
> DstandardizedLoadings(c2withML)
> print(c2withML$Phi, digits = 3)
> DstandardizedLoadings(c3withML)
> print(c3withML$Phi, digits = 3)
> print(DstandardizedLoadings(c5withML), digits = 3)
> print(c5withML$Phi, digits = 3)
> DstandardizedLoadings(c4withML)
> print(c2withML$Phi, digits = 3)
```

Figure 8 part 1 is generated by

```
> tfplot(ytoypc(factors(c4withML)), ytoypc(factors(c2withML)),
      ytoypc(factors(c3withML)), ytoypc(factors(c5withML)), series = 1:2,
      xlab = c(""), ylab = c("factor 1", "factor 2"), lty = c("solid",
```

```
    "dotdash", "dashed", "dotted"), col = c("black", "green",
    "red", "blue"), Title = paste("Factors transaction and long ",
    "(year-to-year growth rate) using 2, 3, 4 and 5 factor models",
    sep = ""), par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1,
    4.1, 0.6, 0.1)), reset.screen = TRUE)
```

Figure 8 part 2 is generated by

```
> tfplot(ytoypc(factors(c4withML)), ytoypc(factors(c3withML)),
      ytoypc(factors(c5withML)), series = 3, lty = c("solid", "dashed",
          "dotted"), xlab = c(""), ylab = c("", "", "factor 3"),
      col = c("black", "red", "blue"), Title = paste("Factor near ",
          "(year-to-year growth rate) using 3, 4 and 5 factor models",
          sep = ""), par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1,
          4.1, 0.6, 0.1)), reset.screen = TRUE)
```

4 Rotation method sensitivity

BpermuteTarget just helps put the factors in the same order, and with the same signs as *c4withML*. It does not otherwise affect the estimation or rotation. The oblimin rotation with $\gamma = 0.5$ is given by

```
> c4withMLg0.5 <- estTSF.ML(MBandCredit, 4, BpermuteTarget = loadings(c4withML),
      rotation = "oblimin", rotationArgs = list(gam = 0.5))
> loadings(c4withMLg0.5)

                             Factor 1   Factor 2    Factor 3   Factor 4
currency                    17.204376 -0.2993325  -2.9221503  -4.161355
personal cheq.              -5.888054 11.8489892  40.8218129  -9.350992
NonbankCheq                  2.306075 -3.1917702   3.5808888   2.094587
N-P demand & notice         28.858827 -9.6930351   4.4360441  12.301306
N-P term                    -2.519688  1.6858628 -21.7335744  32.446348
Investment                  14.121979 70.4973501 -30.5725683 -12.535677
Consumer Credit             -4.945051 -4.9008176  -0.7956332  33.697057
Residential Mortgage         6.854980 29.4305518  10.4874854  -1.255816
Short Term Business Credit -12.201107 22.7939544  -0.5132386   4.095379
Other Business Credit       -1.130633 49.8454385  14.9460409 -13.213999

> DstandardizedLoadings(c4withMLg0.5)

                              Factor 1    Factor 2     Factor 3    Factor 4
currency                    1.27743748 -0.02222566 -0.216971802 -0.30898366
personal cheq.             -0.13709404  0.27588499  0.950471411 -0.21772307
NonbankCheq                 0.21455145 -0.29695432  0.333156946  0.19487511
N-P demand & notice         0.62678638 -0.21052354  0.096346674  0.26717270
N-P term                   -0.03502953  0.02343742 -0.302147267  0.45107975
Investment                  0.12071597  0.60261780 -0.261337111 -0.10715611
```

```
Consumer Credit           -0.16482527 -0.16335091 -0.026519536  1.12316872
Residential Mortgage       0.13620079  0.58475219  0.208374622 -0.02495165
Short Term Business Credit -0.16983977  0.31729252 -0.007144296  0.05700780
Other Business Credit     -0.02007891  0.88520494  0.265426680 -0.23466735
```

```
> DstandardizedLoadings(c4withMLg0.5) - DstandardizedLoadings(c4withML)
```

```
                             Factor 1    Factor 2     Factor 3    Factor 4
currency                   0.25760538 -0.11664428 -0.123103042 -0.22896671
personal cheq.            -0.10810564  0.07268860  0.074141866 -0.13877707
NonbankCheq                0.01240759 -0.15810820 -0.034772560  0.07521179
N-P demand & notice        0.06827175 -0.20725940 -0.105351130  0.02627195
N-P term                  -0.05623237 -0.07763833 -0.067332182  0.11800249
Investment                -0.02941505  0.14372787 -0.021896794 -0.17578085
Consumer Credit           -0.18323724 -0.26844063 -0.127945338  0.32764886
Residential Mortgage      -0.07977436  0.09124587 -0.017845579 -0.17349361
Short Term Business Credit -0.07942246  0.07758680  0.006618814 -0.04002985
Other Business Credit     -0.11640480  0.22188590  0.025477600 -0.27690780
```

Summary information is produced with (output omitted)

```
> summary(c4withMLg0.5)
```

Other rotation results are produced by the following, but outputs are omitted.

```
> c4withMLgneg0.5 <- estTSF.ML(MBandCredit, 4, BpermuteTarget = loadings(c4withML),
      rotation = "oblimin", rotationArgs = list(gam = -0.5))
> loadings(c4withMLgneg0.5)
> DstandardizedLoadings(c4withMLgneg0.5)
> DstandardizedLoadings(c4withMLgneg0.5) - DstandardizedLoadings(c4withML)
> summary(c4withMLgneg0.5)
> c4withMLgneg1.0 <- estTSF.ML(MBandCredit, 4, BpermuteTarget = loadings(c4withML),
      rotation = "oblimin", rotationArgs = list(gam = -1))
> loadings(c4withMLgneg1.0)
> DstandardizedLoadings(c4withMLgneg1.0)
> DstandardizedLoadings(c4withMLgneg1.0) - DstandardizedLoadings(c4withML)
> summary(c4withMLgneg1.0)
> c4withMLbQ <- estTSF.ML(MBandCredit, 4, rotation = "bentlerQ",
      BpermuteTarget = loadings(c4withML))
> loadings(c4withMLbQ)
> DstandardizedLoadings(c4withMLbQ)
> DstandardizedLoadings(c4withMLbQ) - DstandardizedLoadings(c4withML)
> summary(c4withMLbQ)
```

Figure 9 is generated by

```
> tfplot(ytoypc(factors(c4withML)), ytoypc(factors(c4withMLg0.5)),
      ytoypc(factors(c4withMLgneg0.5)), ytoypc(factors(c4withMLgneg1.0)),
      ytoypc(factors(c4withMLbQ)), xlab = c(""), ylab = c("factor 1",
          "factor 2", "factor 3", "factor 4"), lty = c("solid",
          "dashed", "dotted", "dotdash", "longdash"), col = c("black",
          "red", "blue", "green", "pink"), Title = paste("Factors from various 4 factor mode
          "\n and oblimin with gam=0 (solid)"), par = list(omi = c(0.1,
          0.1, 0.1, 0.1), mar = c(3.1, 4.1, 0.6, 0.1)), reset.screen = TRUE)
```

Geomin factors 2 and 3 each have one modestly different loading. Factor 2 has personal chequing mixed in with investment and credit. Factor 3 explains on currency, personal chequing, and investment, so the separation is not so interesting. Output is omitted from these.

```
> c4withMLgm <- estTSF.ML(MBandCredit, 4, rotation = "geominQ",
      BpermuteTarget = loadings(c4withML))
> loadings(c4withMLgm)
> DstandardizedLoadings(c4withMLgm)
```

The difference between the estimates can be checked with (output omitted)

```
> DstandardizedLoadings(c4withMLgm) - DstandardizedLoadings(c4withML)
```

The summary of the 4 factor geomin estimate is given by

```
> summary(c4withMLgm)
```

Figure 10 is generated by

```
> tfplot(ytoypc(factors(c4withML)), ytoypc(factors(c4withMLgm)),
      xlab = c(""), ylab = c("factor 1", "factor 2", "factor 3",
          "factor 4"), lty = c("solid", "dashed"), col = c("black",
          "red"), Title = paste("Factors from geomin (dashed) 4 factor model (year-to-year g
          "\n and oblimin with gam=0 (solid)"), par = list(omi = c(0.1,
          0.1, 0.1, 0.1), mar = c(3.1, 4.1, 0.6, 0.1)), reset.screen = TRUE)
> c4withMLnotNorm <- estTSF.ML(MBandCredit, 4, normalize = FALSE,
      BpermuteTarget = loadings(c4withML))
```

There is only a qualitative statement about the next in the paper (outputs omitted).

```
> DstandardizedLoadings(c4withML)
> DstandardizedLoadings(c4withMLnotNorm)
> DstandardizedLoadings(c4withML) - DstandardizedLoadings(c4withMLnotNorm)
```

5 Sensitivity to sample period

BpermuteTarget=loadings(c4withML) is not good enough in some cases. There are difficulties interpreting factors 2 and 3 in here.

```
> z <- matrix(0, 10, 4)
> z[matrix(c(1, 6, 2, 7, 1:4), 4, 2)] <- c(11, 104, 20, 13)
> c4withMLbefore90 <- estTSF.ML(tfwindow(MBandCredit, end = c(1989,
      12)), 4, BpermuteTarget = z)
> c4withMLafter95 <- estTSF.ML(tfwindow(MBandCredit, start = c(1995,
      1)), 4, BpermuteTarget = loadings(c4withML))
> z <- matrix(0, 10, 4)
> z[matrix(c(1, 6, 2, 7, 1:4), 4, 2)] <- c(11, 104, 20, 13)
> c4withMLbefore95 <- estTSF.ML(tfwindow(MBandCredit, end = c(1994,
      12)), 4, BpermuteTarget = z)
> c4withMLafter00 <- estTSF.ML(tfwindow(MBandCredit, start = c(2000,
      1)), 4, BpermuteTarget = loadings(c4withML))
> c4withML90to00 <- estTSF.ML(tfwindow(MBandCredit, start = c(1990,
      1), end = c(2000, 1)), 4, BpermuteTarget = loadings(c4withML))
```

Figure 11 is generated by

```
> tfplot(ytoypc(factors(c4withML)), ytoypc(factors(c4withMLbefore90)),
     ytoypc(factors(c4withMLbefore95)), ytoypc(factors(c4withMLafter95)),
     ytoypc(factors(c4withMLafter00)), ytoypc(factors(c4withML90to00)),
     xlab = c(""), ylab = c("factor 1", "factor 2", "factor 3",
         "factor 4"), ylim = list(NULL, c(-20, 20), c(-25, 40),
         NULL), graphs.per.page = 4, lty = c("dashed", "dotted",
         "dotdash", "longdash", "dotted", "twodash"), col = c("red",
         "blue", "green", "pink", "violet", "brown"), Title = paste("Factors (year to year
         "ML estimation with quartimin rotation objective", sep = ""),
     par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
         0.6, 0.1)), reset.screen = TRUE)
```

Figure 12 is generated by

```
> tfplot(ytoypc(MBandCredit), ytoypc(explained(c4withML)), ytoypc(explained(c4withMLbefore9(
     ytoypc(explained(c4withMLbefore95)), ytoypc(explained(c4withMLafter95)),
     ytoypc(explained(c4withMLafter00)), ytoypc(explained(c4withML90to00)),
     series = 1:5, graphs.per.page = 5, ylab = c("currency", "personal cheq.",
         "NonbankCheq", "N-P demand & notice", "N-P term"), ylim = list(NULL,
         NULL, c(-70, 70), NULL, c(-70, 70)), lty = c("solid",
         "dashed", "dotted", "dotdash", "longdash", "dotted",
         "twodash"), col = c("black", "red", "blue", "green",
         "pink", "violet", "brown"), Title = paste("Explained money indicators 1-5 (year to
         "using 4 factors, full sample and sub-samples", sep = ""),
     par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
         0.6, 0.1)), reset.screen = TRUE)
```

Figure 13 is generated by

```
> tfplot(ytoypc(MBandCredit), ytoypc(explained(c4withML)), ytoypc(explained(c4withMLbefore9(
     ytoypc(explained(c4withMLbefore95)), ytoypc(explained(c4withMLafter95)),
```

```
    ytoypc(explained(c4withMLafter00)), ytoypc(explained(c4withML90to00)),
    series = 6:10, graphs.per.page = 5, ylab = c("", "", "",
        "", "", "Investment", "Consumer Credit", "Residential Mortgage",
        "Short Term Business Credit", "Other Business Credit"),
    lty = c("solid", "dashed", "dotted", "dotdash", "longdash",
        "dotted", "twodash"), col = c("black", "red", "blue",
        "green", "pink", "violet", "brown"), Title = paste("Explained money indicators 6-1
        "using 4 factors, full sample and sub-samples", sep = ""),
    par = list(omi = c(0.1, 0.1, 0.1, 0.1), mar = c(3.1, 4.1,
        0.6, 0.1)), reset.screen = TRUE)
```

6 Comparison with Aggregates

Compute aggregates M1+ and M2++

```
> M1 <- tfwindow(M1total, start = c(1981, 11), end = c(2004, 11)) *
      scale
> seriesNames(M1) <- "Real per Capita M1"
> z <- tframed(MB2001 + MB486 + MB487p + MB452 + MB452adj + MB472 +
      NonbankCheq)
> M1p <- tfwindow(z, start = c(1981, 11), end = c(2004, 11)) *
      scale
> seriesNames(M1p) <- "Real per Capita M1+"
> M2pp <- tfwindow(M1total + MB472 + MB473 + MB452 + MB453 + MB454 +
      NonbankCheq + NonbankNonCheq + NonbankTerm + +MB2046 + MB2047 +
      MB2048 + MB2057 + MB2058, start = c(1981, 11), end = c(2004,
      11)) * scale
> seriesNames(M2pp) <- "Real per Capita M2++"
```

and put factors on the same scale.

```
> f <- tframed(factors(c4withML)[, 1:2], tf = tframe(factors(c4withML)))
> mnF <- colMeans(f)
> mnM <- colMeans(cbind(M1p, M2pp))
> f <- sweep(f, 2, mnM/mnF, "*")
```

Now compare the transaction factor with M1+ and the long term factor with M2++. Figure 14 is generated by

```
> tfplot(ytoypc(f), ytoypc(cbind(M1, M2pp)), graphs.per.page = 2,
      lty = c("dashed", "solid"), col = c("red", "black"), Title = paste("(year to year grow
          "computed using ordinary ML parameters (dashed)", sep = ""),
      ylab = c("M1+ vs. factor 1", "M2++ vs. factor 2"), par = list(omi = c(0.1,
          0.1, 0.1, 0.1), mar = c(3.1, 4.1, 0.6, 0.1)), reset.screen = TRUE)
```

7 References

Paul D. Gilbert and Erik Meijer, (2006) ”Money and Credit Factors”, Bank of Canada Working Paper 2006-3, Available at `<http://www.bank-banque-canada.ca/en/res/wp/wp(y)_2006.html>`.

Paul D. Gilbert and Erik Meijer, (2005) ”Time Series Factor Analysis with an Application to Measuring Money”, Research Report 05F10, University of Groningen, SOM Research School. Available at <http://som.eldoc.ub.rug.nl/reports/themeF/2005/05F10/>.