

World fertility analysis

Indrė Blagnytė, Vismantas Tučas

2024-05-23

Chosen data

For this project we chose to use World Fertility Data 2019. The data set consists of age specific fertility rates, total fertility and mean age of childbearing for 201 countries or areas of the world and includes empirical data from civil registration systems, population censuses, and sample surveys that covers the time period from 1950 to 2019. The reports were collected by the United Nations Department of Economic and Social Affairs.

We chose to analyze Total Fertility Rate trends in Europe's and Asia's largest countries from 1950 to 2019. Because of this we focused exclusively on Total Fertility Rates and limited the scope of the data to the largest countries by population in Europe and Asia to compare regional fertility trends (table 1).

Table 1. Countries chosen for the data analysis.

Chosen European countries		Chosen Asian countries	
Russia	Belgium	China	Thailand
Germany	Greece	India	Myanmar
United Kingdom	Czechia	Indonesia	South Korea
France	Portugal	Pakistan	Iraq
Italy	Sweden	Bangladesh	Afghanistan
Spain	Hungary	Japan	Saudi Arabia
Ukraine	Belarus	Philippines	Uzbekistan
Poland	Austria	Vietnam	Malaysia
Romania	Switzerland	Iran	Yemen
Netherlands	Serbia	Turkey	Nepal

The total fertility rate (TF) is calculated using the following formula:

$$TF = \frac{5}{1000} \times \sum_{a=15-49} f_a$$

Where f_a is age-specific fertility expressed in births per 1000 women.

Total fertility describes the mean number of children a woman would have by age 50 if she survived to age 50 and was subject, throughout her life, to the age-specific fertility rates observed in a given year. The measure is expressed as the number of children per woman.

Data processing

We plotted the raw unsmoothed data (Fig. 1). Every line in the plot represents the fertility rates of a single country.

```
# Draw the graph
ggplot(final_data, aes(x = Date, y = Value, group = `Country or Area`, color = `Continent`)) +
  geom_line() + # Draw lines for each country
  scale_color_manual(values = c("Europe" = "blue", "Asia" = "red")) + # Set colors for continents
  labs(title = "Fertility Rates Over Time by Continent",
       x = "Date",
       y = "Fertility Rate",
       color = "Continent") +
  theme_minimal() + # Use a minimal theme
  theme(legend.position = "bottom") # Adjust legend position
```

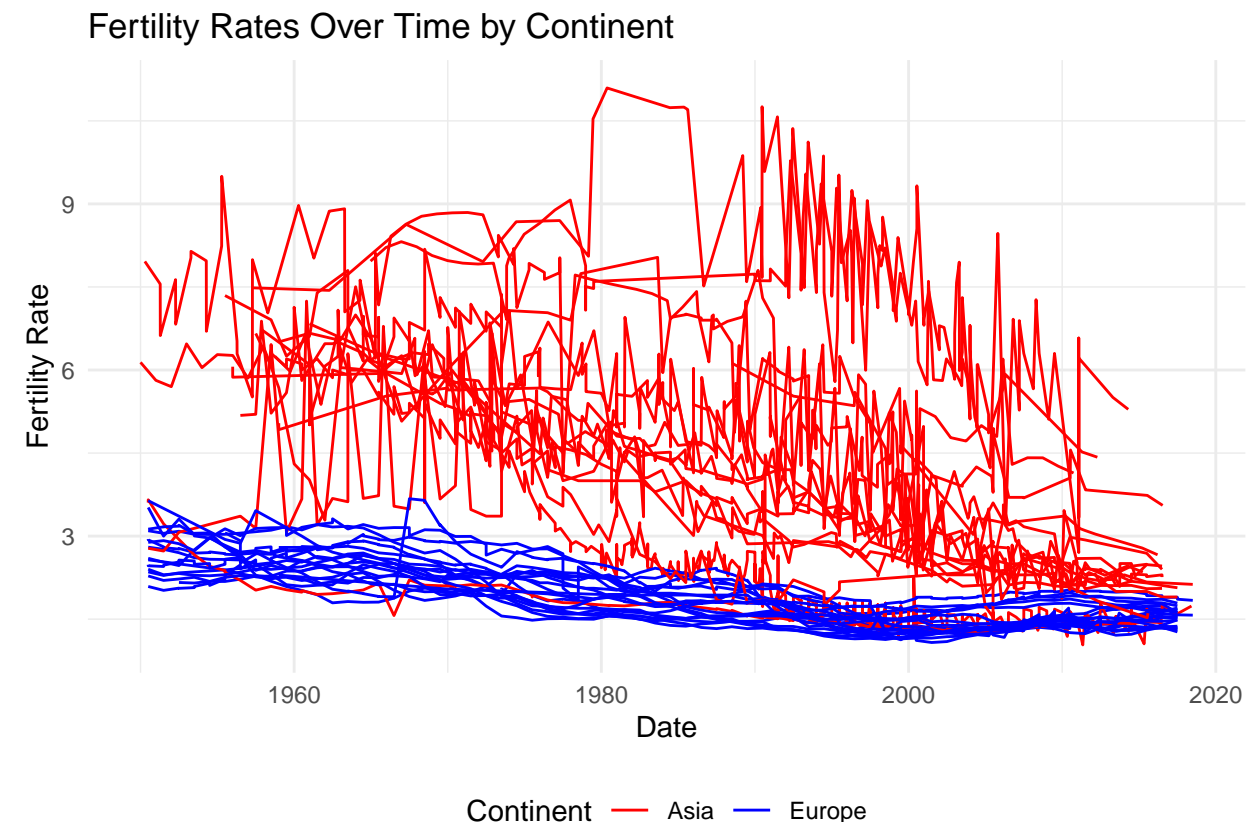


Figure 1: Unsmoothed data.

Because there was a lot of visible noise in the Asian countries' data, we chose to perform a logarithmic transformation (Fig. 2).

```
final_data <- final_data %>%
  mutate(ValueLog = log(Value))

View(final_data)
```

```
# Draw the graph
ggplot(final_data, aes(x = Date, y = ValueLog, group = `Country or Area`, color = `Continent`)) +
  geom_line() + # Draw lines for each country
  scale_color_manual(values = c("Europe" = "blue", "Asia" = "red")) + # Set colors for continents
  labs(title = "Fertility Rates Over Time by Continent",
       x = "Date",
       y = "Fertility Rate",
       color = "Continent") +
  theme_minimal() + # Use a minimal theme
  theme(legend.position = "bottom") # Adjust legend position
```

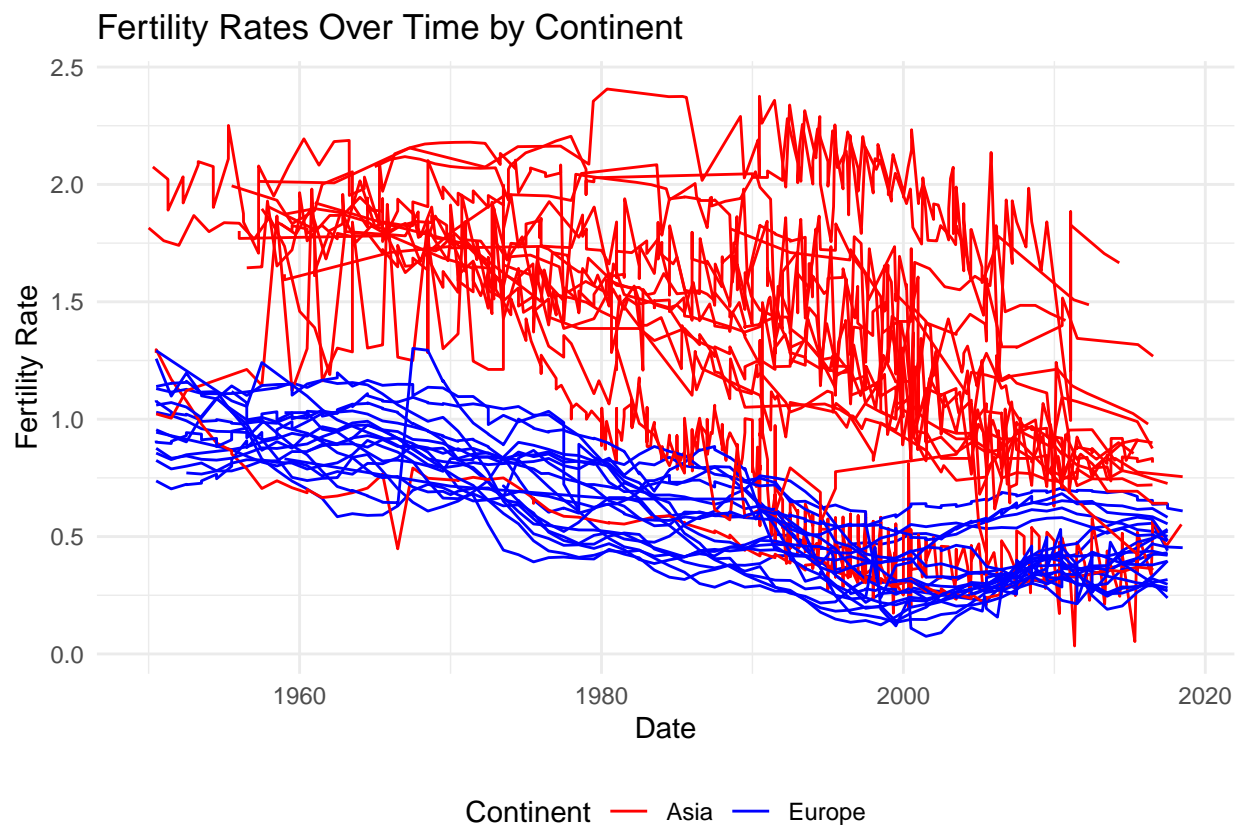


Figure 2: Fertility data after logarithmic transformation.

Smoothing

Due to the nature of our data, the B-Spline basis smoothing method was chosen. After experimenting with different smoothing parameters, we decided on: number of Basis Functions (Nbasis) = 5, order of Basis Functions (Norder) = 4, smoothing Penalty (Lambda) = 1×10^2

```
# Get the range of years
range_of_dates <- range(final_data$Date)
```

```

# Set up the B-spline basis object
nbasis <- 5 # Can experiment, but this one seems to work the best
basis_obj <- create.bspline.basis(rangeval=range_of_dates, nbasis=nbasis, norder=4)

lambda_v = 1e2 # Lambda controls the smoothness

# Initialize an empty list for storing fd objects for each country
country_fds <- list()

# Initialize empty lists for storing smoothed fd objects for European and Asian countries
europe_fds <- list()
asia_fds <- list()

# Loop through unique countries and smooth the data
for(country in unique(final_data$`Country or Area`)) {
  country_data <- subset(final_data, `Country or Area` == country)
  argvals <- country_data$Date
  y <- country_data$ValueLog

  # Smooth the data
  fdParObj <- fdPar(basis_obj, Lfdobj=int2Lfd(2), lambda= lambda_v)
  smooth_res <- smooth.basis(argvals, y, fdParObj)

  # Store the fd object based on continent
  if (country %in% largest_european_countries) {
    europe_fds[[country]] <- smooth_res$fd
  } else {
    asia_fds[[country]] <- smooth_res$fd
  }
}

# Combine the coefficients from each fd object into a single matrix for Europe
europe_coefs <- do.call(cbind, lapply(europe_fds, function(fd) fd$coefs))
europe_basis_obj <- create.bspline.basis(rangeval=range_of_dates, nbasis=nbasis, norder=4)
europe_fd <- fd(europe_coefs, europe_basis_obj)

# Combine the coefficients from each fd object into a single matrix for Asia
asia_coefs <- do.call(cbind, lapply(asia_fds, function(fd) fd$coefs))
asia_basis_obj <- create.bspline.basis(rangeval=range_of_dates, nbasis=nbasis, norder=4)
asia_fd <- fd(asia_coefs, asia_basis_obj)

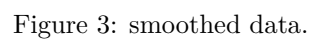
# Correctly set the range for the combined FD creation
combined_basis_obj <- create.bspline.basis(rangeval=range_of_dates, nbasis=nbasis, norder=4)

# Combine the coefficients from European and Asian fd objects
combined_coefs <- cbind(europe_fd$coefs, asia_fd$coefs)

# Use the new, correctly ranged basis object for the combined fd object
combined_fd <- fd(combined_coefs, basisobj = combined_basis_obj)

# Plot the combined functional data object
plot(combined_fd)

```



```
## [1] "done"
```

Mean and Standard Deviation

For the smoothed data, we calculated the mean function (red dotted line), standard deviation (blue dotted line) and the mean function \pm the standard deviation (both purple dotted lines). The red line shows the mean central tendency of all the countries in our dataset, which signifies a downward trend. The blue line shows the variability and dispersion of each countries curves, it seems that in the beginning the data is more spread out, but as years go on, the variability becomes smaller, but towards the year 2005 becomes more variable. Given the purple lines, some countries exceed the the mean function + standard deviation threshold, showing increased variability and uncertainty.

```
plot(combined_fd)
```

```
## [1] "done"
```

```
meanlogprec = mean.fd(combined_fd)
stddevlogprec = std.fd(combined_fd)

lines(meanlogprec, lwd=4, lty=2, col=2)
lines(stddevlogprec, lwd=4, lty=2, col=4)

lines(meanlogprec-stddevlogprec, lwd=4, lty=2, col=6)
lines(meanlogprec+stddevlogprec, lwd=4, lty=2, col=6)
```

2D and 3D covariance surface plot for Asian country data

We then plotted separate covariance surface plots both for Asian and European countries separately to provide more insights in their own respective groups. For Asian data, in both of the covariance surface plots we can see that in the beginning the covariance values of these countries are quite variable, meaning the total fertility rates are not correlated strongly, however, towards the end we can see a peak, signifying a large covariance value, meaning the countries' total fertility rates tend to increase and decrease together.

```
# Calculate the covariance matrix for Asian country data
covariance_matrix_asia <- cov(asia_coefs)

# Plot the 3D covariance surface
persp(covariance_matrix_asia, theta=-45, phi=25, r=3, expand = 0.5,
      ticktype='detailed',
      xlab = "Time", ylab = "Time", zlab = "Covariance",
      main = "3D Covariance for Asian Country Data")
```

```
# Plot the covariance surface in 2d for Asian country data
contour(covariance_matrix_asia,
      ticktype = 'detailed',
      xlab = "Time", ylab = "Time",
      main = "2D Covariance for Asian Country Data")
```

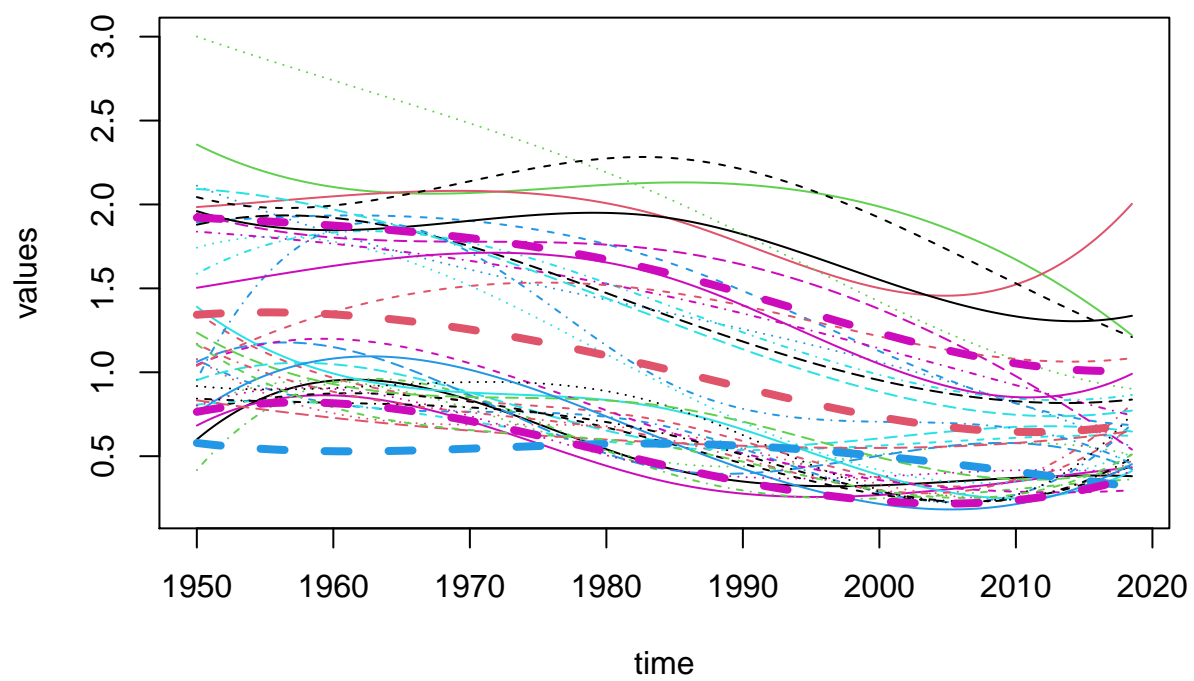


Figure 4: Mean and standard deviation on smoothed data.

3D Covariance for Asian Country Data

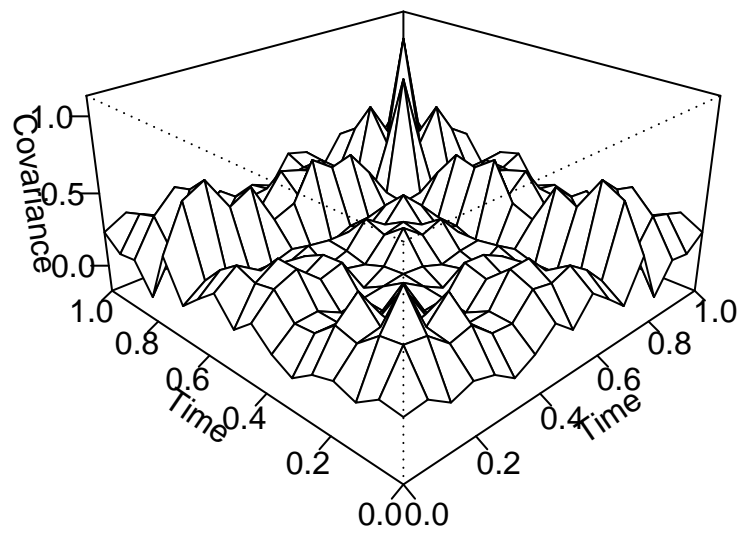


Figure 5: 3D covariance matrix for Asian country data.


```
## Warning in plot.window(xlim, ylim, ...): "ticktype" is not a graphical
## parameter

## Warning in title(...): "ticktype" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ticktype" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ticktype" is not
## a graphical parameter

## Warning in box(...): "ticktype" is not a graphical parameter
```

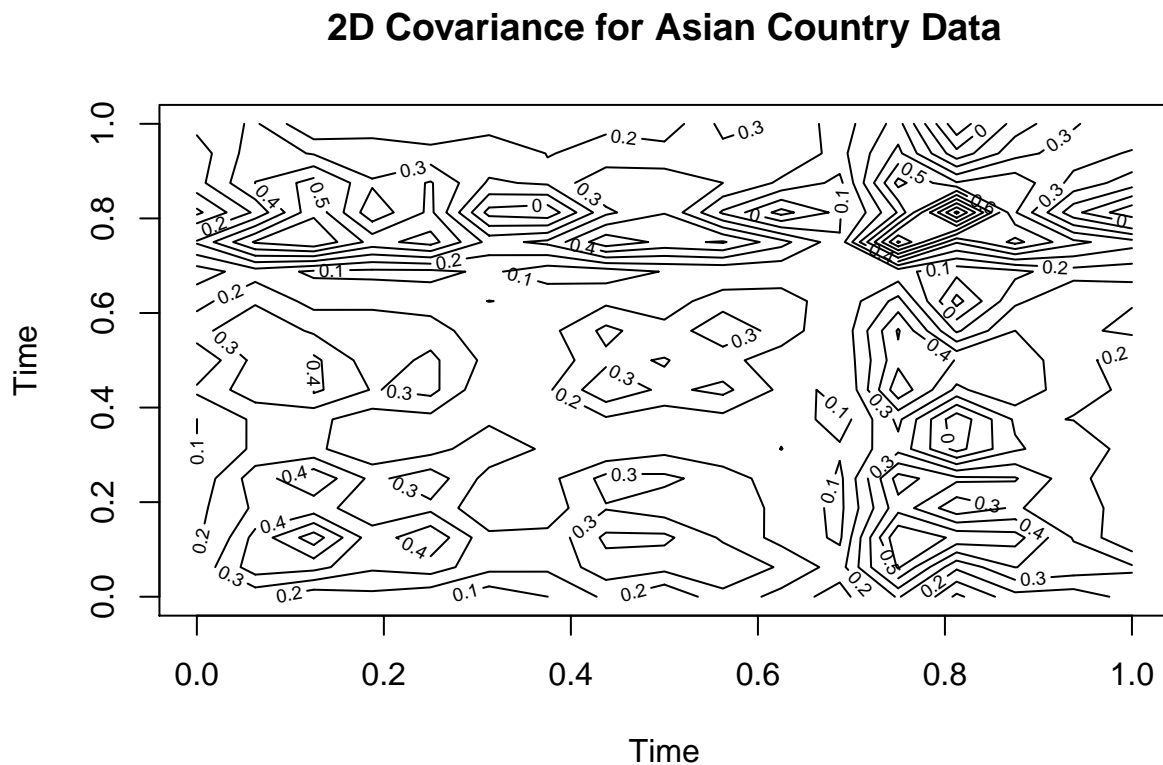


Figure 6: 2D covariance matrix for Asian country data.

2D and 3D covariance surface plot for European country data

Unlike the Asian data, which had a clear peak suggesting a strong correlation in a particular time period, the European data seems to have multiple peaks and valleys, suggesting more complex and variable relationships over time, also it shows quite low covariance values all across the board, further indicating their complex nature.

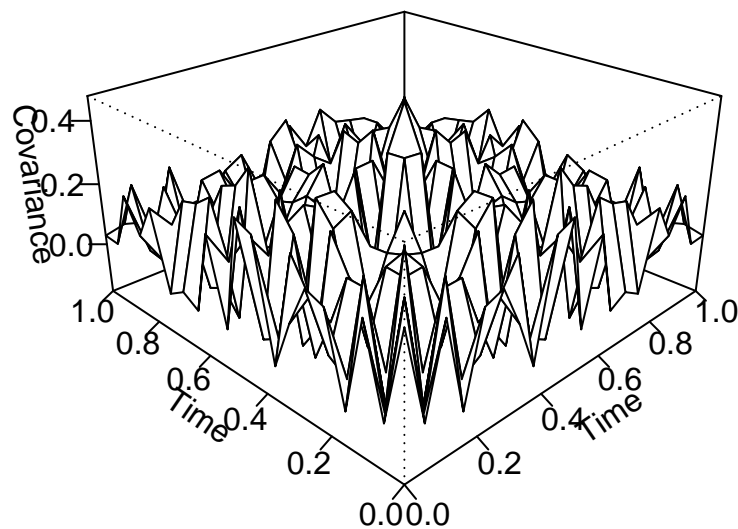
```

# Calculate the covariance matrix for European country data
covariance_matrix_europe <- cov(europe_coefs)

# Plot the 3D covariance surface
persp(covariance_matrix_europe, theta = -45, phi = 25, r = 3, expand = 0.5,
      ticktype = 'detailed',
      xlab = "Time", ylab = "Time", zlab = "Covariance",
      main = "Covariance for European Country Data")

```

Covariance for European Country Data

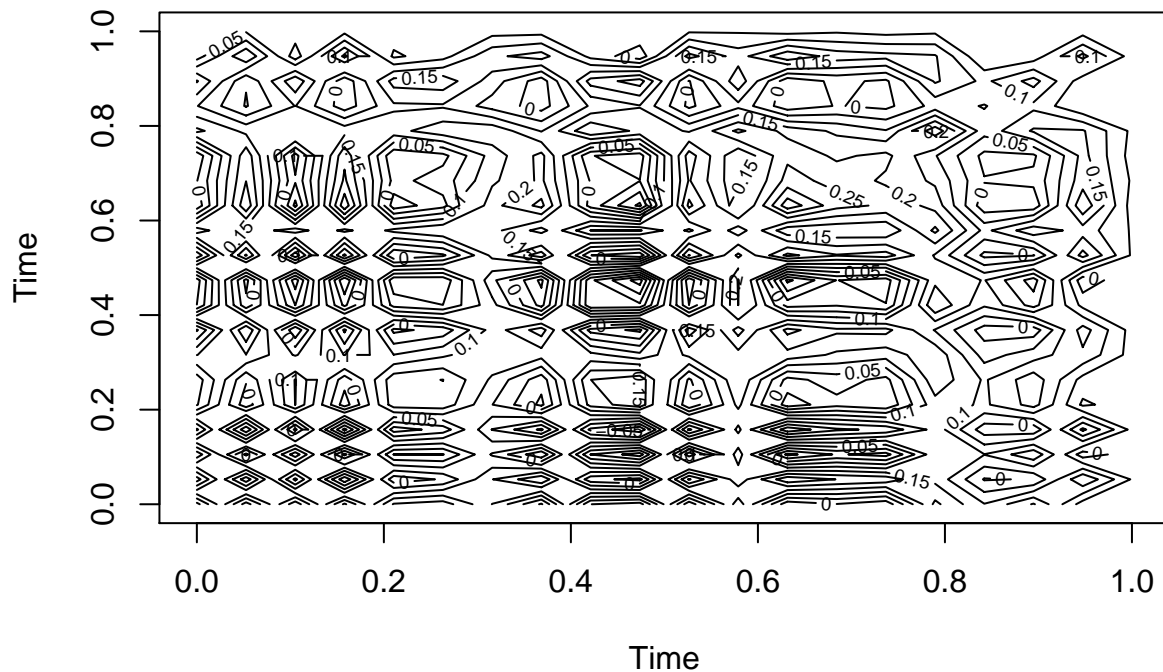


```

# Plot the covariance surface in 2d for European country data
contour(covariance_matrix_europe,
      xlab = "Time", ylab = "Time",
      main = "Covariance for European Country Data")

```

Covariance for European Country Data



Boxplots

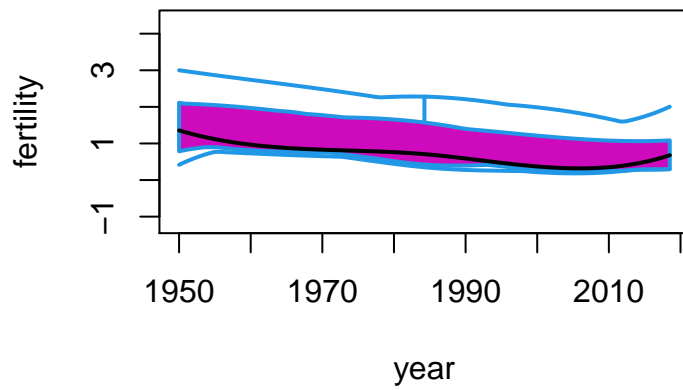
Boxplots were plotted for the combined country data, Asian country data and European country data separately. The combined country boxplot indicates no outliers and an obvious downward trend, with a few countries somewhere around 2011 experiencing an increase in fertility rates.

The Asian country boxplots shows a similar trend with less variability, which is natural, however there are two outliers. The upper outlier is Saudi Arabia and the lower outlier being Japan. Saudi Arabia was a rural country, however became very wealthy after oil discovery, which would affect their populations educational level as well as enabled access to planned parenthood, which explains the fertility rates going down. Japan, however, was and still is a lower outlier due to their demographic situation, they have the largest proportion of senior citizens in the world and are very career driven, thus creating a family is not a priority.

The European country boxplot seems to show a severe downtrend in fertility without any outliers, however in 2005 started to gain momentum for an upward trend.

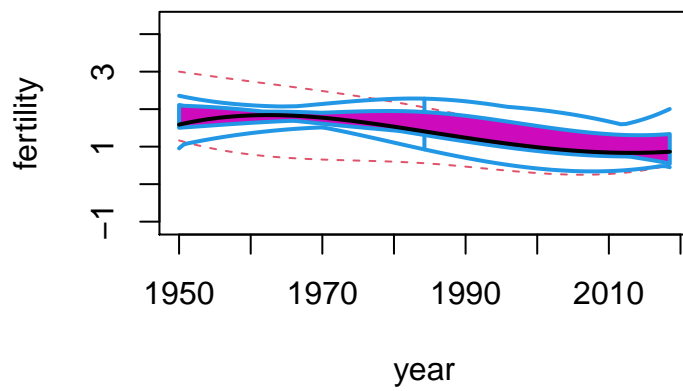
```
boxplot(combined_fd, xlab = "year", ylab = "fertility", main = "Combined fertility")
```

Combined fertility



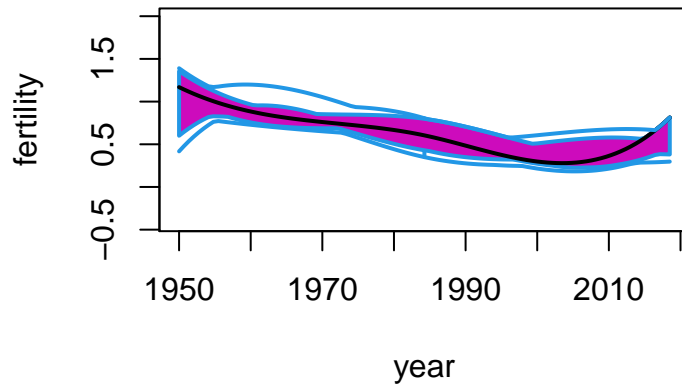
```
asia <- boxplot(asia_fd, xlab = "year", ylab = "fertility", main = "Asian fertility")
```

Asian fertility



```
boxplot(europe_fd, xlab = "year", ylab = "fertility", main = "European fertility")
```

European fertility



```
outliers <- asia$out
print(outliers)
```

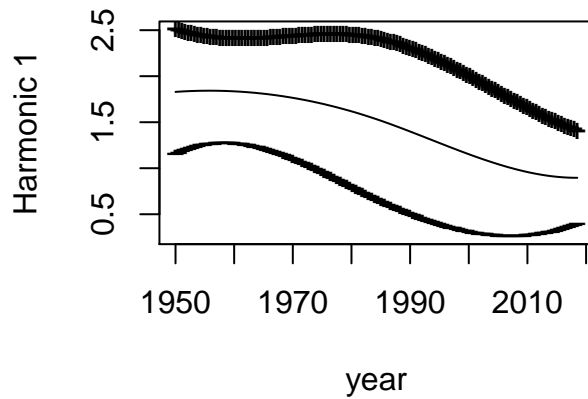
```
## [1] 7 13
```

Functional PCA for asian country data

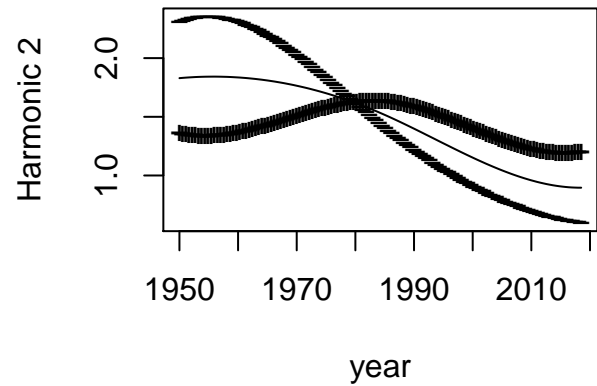
For the Asian country data we used a total of 4 principal components, which explain 99.3% of the variability. Before rotation the majority of variance (81.8 %) is explained by the 1st PCA function which has a positive impact on the mean function throughout the the whole measurement period. The 2nd PCA function which explains 13.5 % of variance has a negative impact on the mean function up until 1980, and a positive impact after. Functions 3 and 4 show more complex tendencies and explain a much smaller portion of variance (2.4 and 1.6 % respectively)

```
# PCA for Asian country dataset
nharm = 4
pcalist = pca.fd(asia_fd, nharm, centerfns = TRUE)
par(cex.main = 0.8)
plot(pcalist, xlab="year")
```

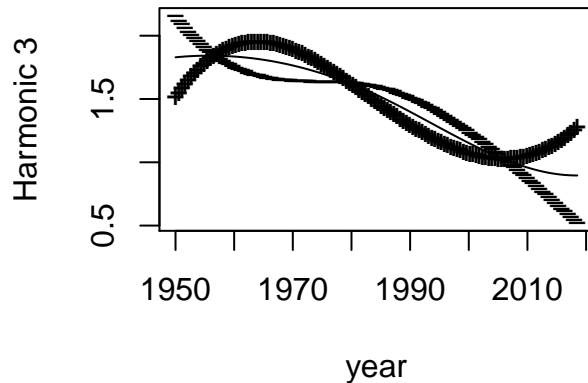
PCA function 1 (Percentage of variability 81.8)



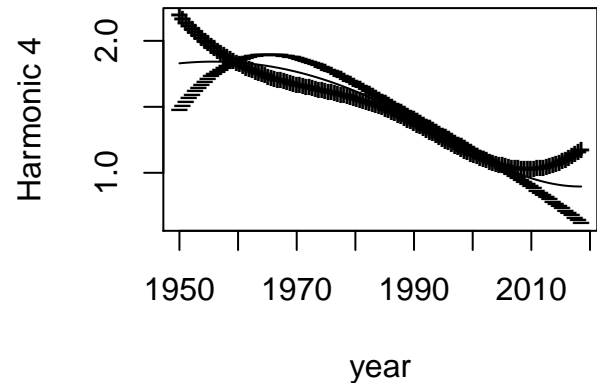
PCA function 2 (Percentage of variability 13.5)



PCA function 3 (Percentage of variability 2.4)



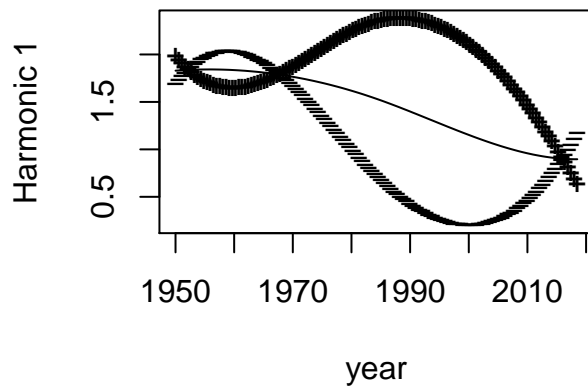
PCA function 4 (Percentage of variability 1.6)



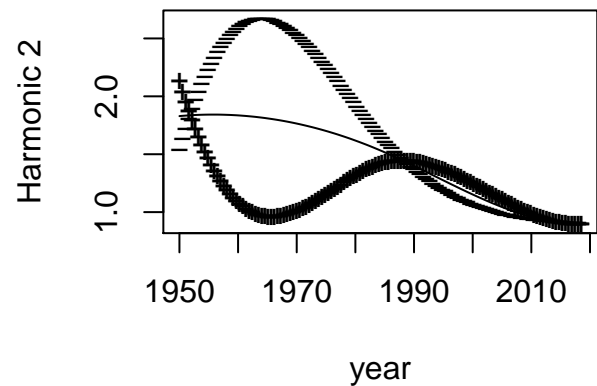
Varimax rotation was used to distribute the variance more evenly across the factors. After rotation the 1st PCA function accounts for 55.3 % of the variance. It has a negative impact on the mean function from around the 1950's to 1970, and positive impact from then to the mid 2010's. The 2nd function explaining 27.4 % mainly has a negative impact on the mean function from 1950's to around 1990. The 3rd function (7.9 %) has a positive impact from the year 2000 and the 4th (8.7 %) positive impact until 1960 and a slight negative impact from 1960 to 1980.

```
# Rotation for Asian data
varmx_as <- varmx.pca.fd(pcalist)
par(cex.main = 0.8)
plot(varmx_as, xlab="year")
```

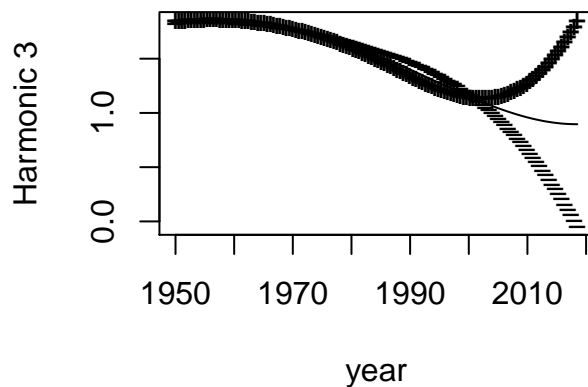
PCA function 1 (Percentage of variability 55.3)



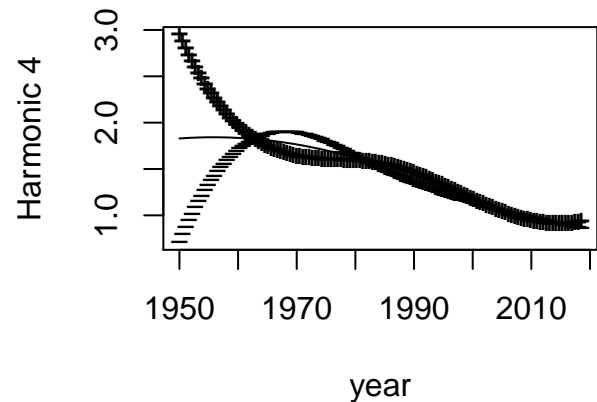
PCA function 2 (Percentage of variability 27.4)



PCA function 3 (Percentage of variability 7.9)



PCA function 4 (Percentage of variability 8.7)

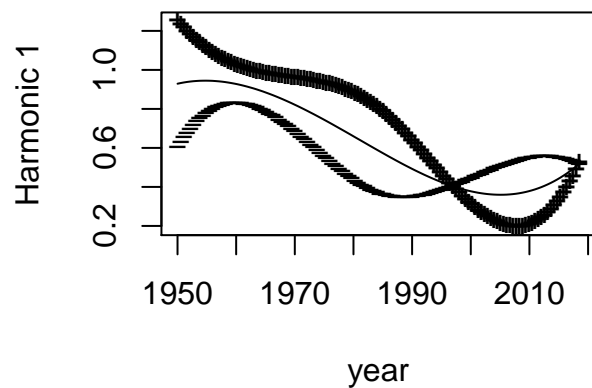


Functional PCA for european country data

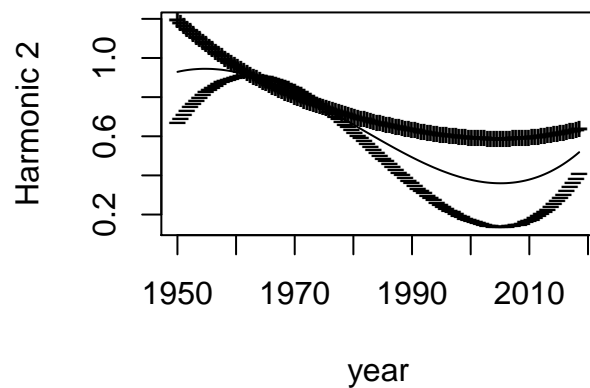
For European data, 5 components were used, which account for >99.9% of the variation. Before rotation the percentage of variance explained was already quite well distributed across functions, however the interpretation was quite difficult. The 1st PCA function which explains 38.4 % of variance has a positive impact on the mean function up until the mid 1990's. The 2nd PCA function (31.9 %) has a positive impact on the mean function up until 1960, and again after 1980. The 3rd function (20.2 %) has a very complex structure. 4th function (6 %) shows negative impact until 1960, 5th (3.5 %) positive from 2010.

```
# PCA for European data
nharm = 5
pcalist = pca.fd(europe_fd, nharm, centerfns = TRUE)
par(cex.main = 0.8)
plot(pcalist, xlab="year")
```

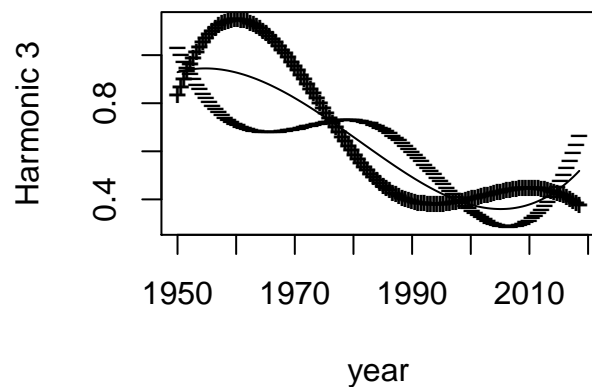
PCA function 1 (Percentage of variability 38.4)



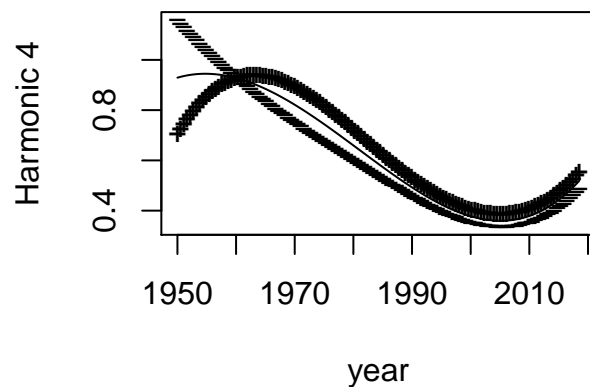
PCA function 2 (Percentage of variability 31.9)



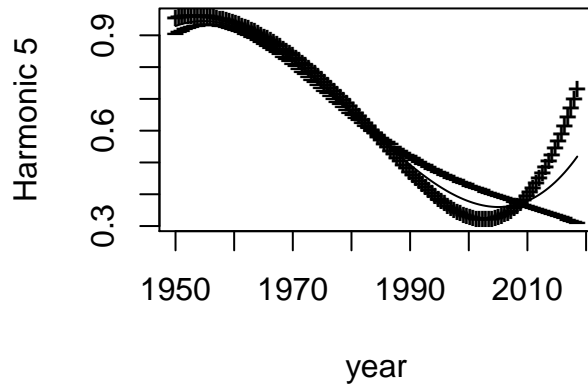
PCA function 3 (Percentage of variability 20.2)



PCA function 4 (Percentage of variability 6)



PCA function 5 (Percentage of variability 3.5)

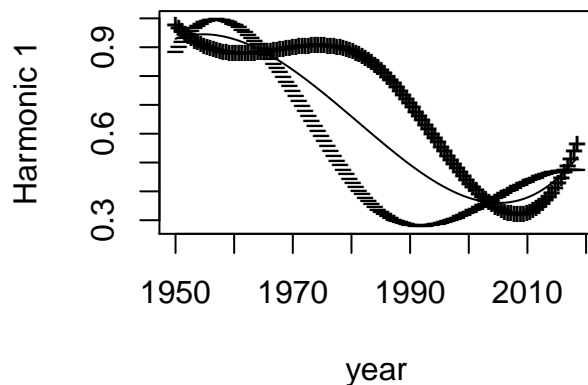


After rotation the 1st PCA function accounts for 25.5 % of the variance and has a complex structure. The 2nd function explaining 29.7 % mainly has a positive impact on the mean function from the mid 1980's to mid 2010's. The 3rd (21.8 %) mainly has a positive impact from 1950's to around 1980, the 4th (16.6 %) - negative impact until 1960, 5th (6.4 %) - positive from mid 2000's.

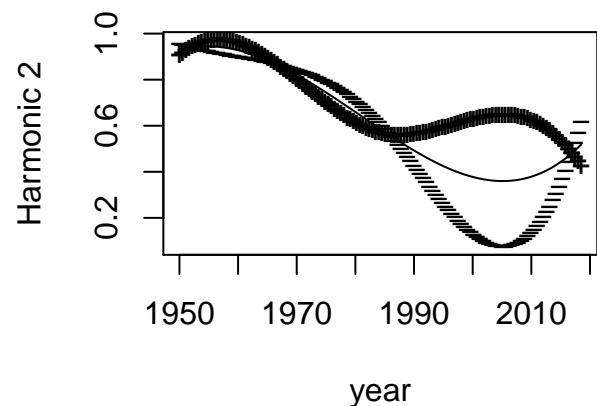
Overall both before and after rotation the PCA functions for European data are quite complex and difficult to interpret.

```
# Rotation for European data
varmx_eu <- varmx.pca.fd(pcalist)
par(cex.main = 0.8)
plot(varmx_eu, xlab="year")
```

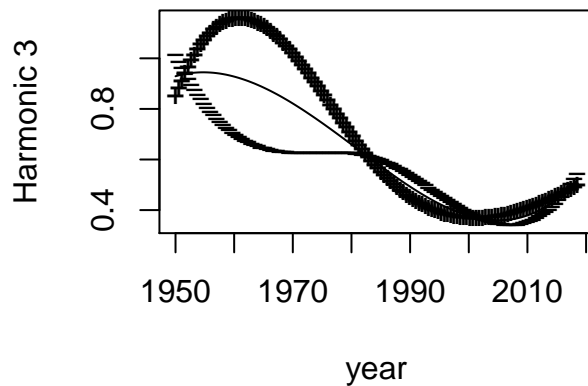
PCA function 1 (Percentage of variability 25.5)



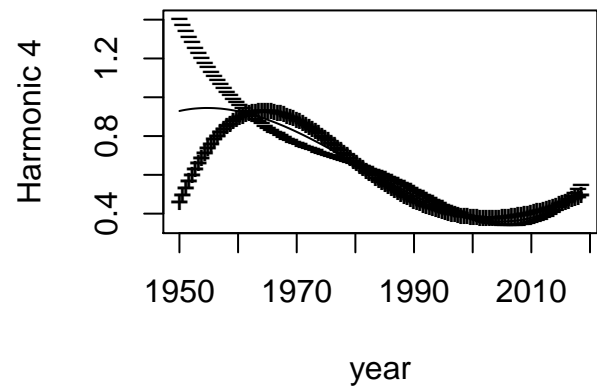
PCA function 2 (Percentage of variability 29.7)



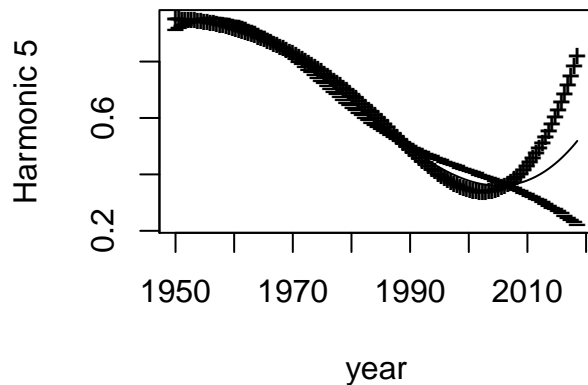
PCA function 3 (Percentage of variability 21.8)



PCA function 4 (Percentage of variability 16.6)



PCA function 5 (Percentage of variability 6.4)



Hypothesis testing

Two sample pointwise Z test

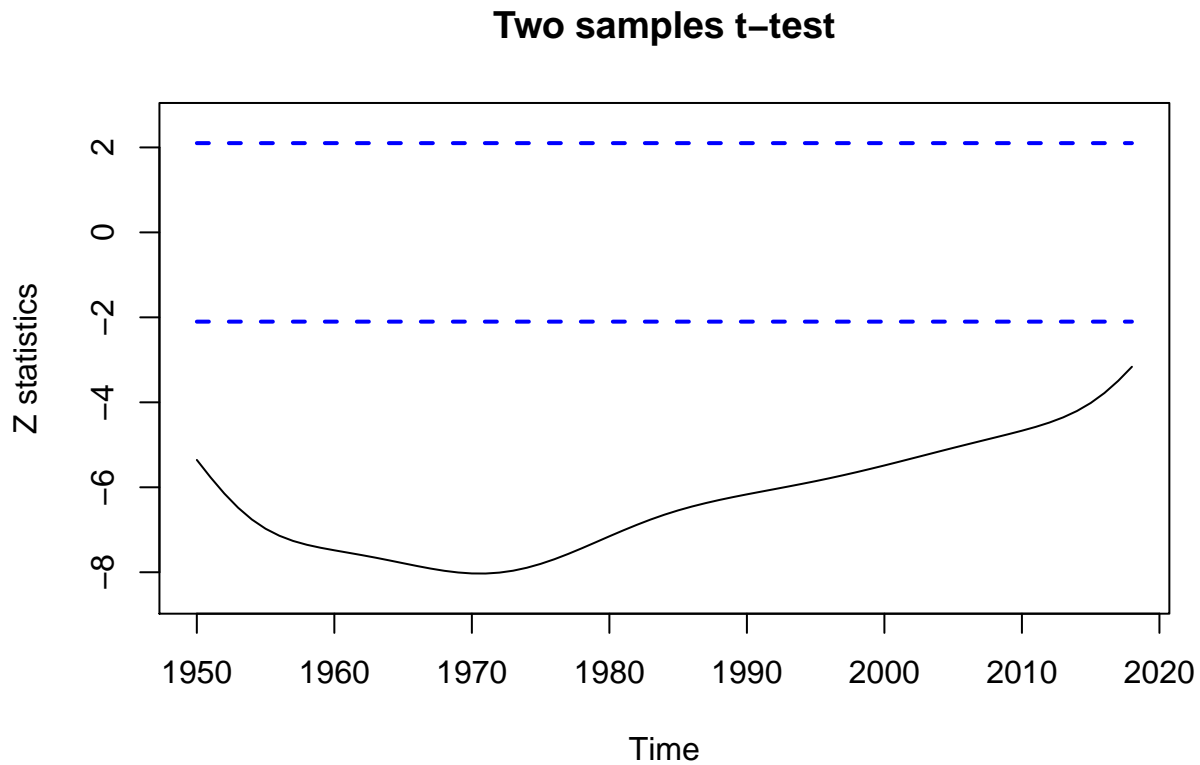
First we will perform a two sample point-wise Z test to compare the point-wise mean functions of Europe and Asia, thus this hypothesis was proposed:

$H_0: \mu(\text{Europe}) = \mu(\text{Asia})$ $H_1: \mu(\text{Europe}) \neq \mu(\text{Asia})$

There was a statistically significant difference between Europe's and Asia's mean fertility throughout the whole reported period. With Europe always showing a lower mean fertility compared to Asia. The difference is increases until the 1970's when the difference is largest and has been shrinking since. The shrinking difference between Europe's and Asians rates could be the result of the growing economy and better living conditions in Asia leading to lower birth rates.

Critical value 2.100922

```
t.seq <- seq(1950.001, 2018.501, by=1)
stat <- Ztwosample(x=europe_fd, y=asia_fd, t.seq = t.seq)
```



```
stat$params
```

```
## $critical.value
## [1] 2.100922
```

Two sample permutation test

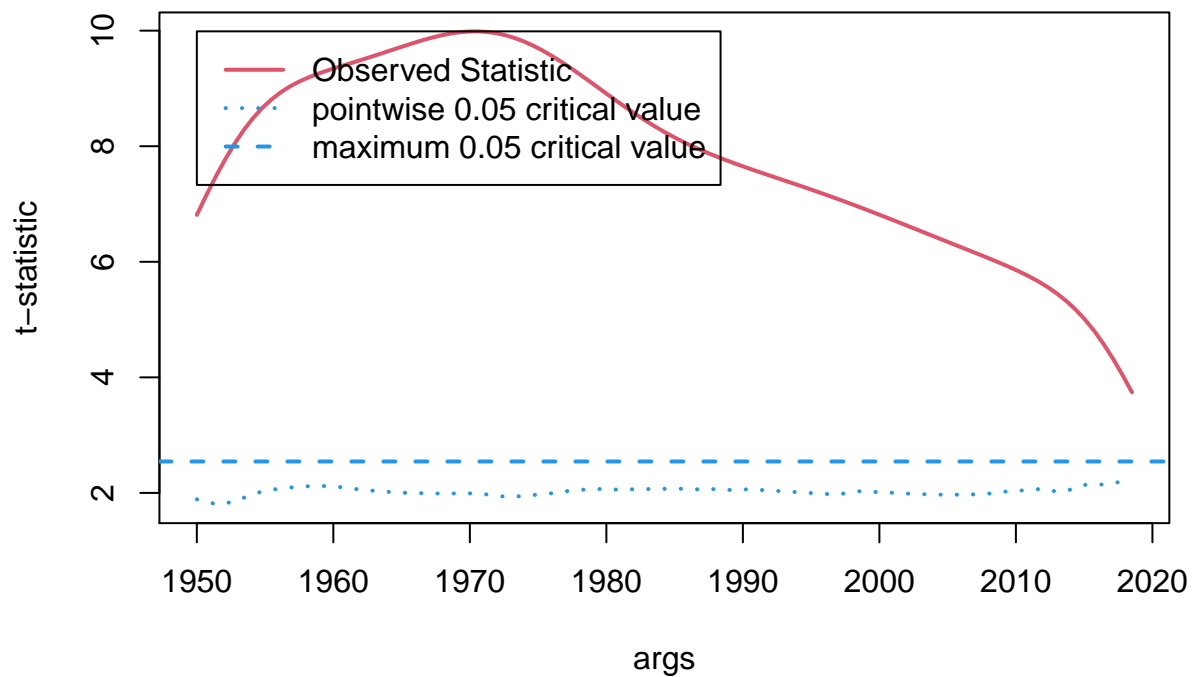
To confirm, we also did a two sample permutation test, two sample L2 statistic and two sample F-statistic for the same hypothesis.

The two sample permutation test shows the same tendency as the Z-test.

P-value was close to 0 (<0.05) - statistically significant difference between the two groups

Critical value 2.365367

```
# Also a two sample permutation test
stat <- tperm.fd(europe_fd, asia_fd)
```



```
# Print only the p-value, q-value, and observed statistic
list(p_value = stat$pval, q_value = stat$qval, T_obs = stat$Tobs)
```

```
## $p_value
## [1] 0
##
## $q_value
##      95%
## 2.544153
##
## $T_obs
## [1] 9.98787
```

Two sample L2 statistic

Performed both naive and bootstrap methods

In both cases p-value was close to 0 (< 0.05)

Statistically significant difference between the two groups

```
stat <- L2.stat.twosample(x=europe_fd, y=asia_fd, t.seq = t.sq, method=1)
stat$pvalue
```

```
## [1] 0
```

```
stat <- L2.stat.twosample(x=europe_fd, y=asia_fd, t.seq = t.sq, method=2, replications=500)
stat$pvalue
```

```
## [1] 0
```

Two sample F-statistic

Performed both naive and bootstrap methods

In both cases p-value was close to 0 (<0.05)

Statistically significant difference between the two groups

```
stat <- F.stat.twosample(x=europe_fd, y=asia_fd, t.seq = t.sq, method=1)
stat$pvalue
```

```
## [1] 0
```

```
stat <- F.stat.twosample(x=europe_fd, y=asia_fd, t.seq = t.sq, method=2, replications=500)
stat$pvalue
```

```
## [1] 0
```

Timeseries modelling and forecasting

Due to the nature of our data, we decided to perform time series modelling and forecasting. After creating functional time series object, we fitted them to separate models - combined, Asian and European data. Their coefficients were visualized.

In the combined data plot, we can see that the beta1 and beta2 coefficients are quite stable for about 20 years, after that it becomes much more variable, beta 1 (first PCA) being more severely varied, possibly due to Asian country data.

In the European data plot, we can see that beta1, beta2 and beta3 are a little bit varied, but not as much as the combined data plot.

In the Asian data plot, beta1 and beta2 are both strongly fluctuated, indicating more variability all across the board.

```
# Get the matrix of combined, European and Asian functional data
total_fertility_m <- eval.fd(t.sq, combined_fd, returnMatrix = TRUE)
eu_fertility_m <- eval.fd(t.sq, europe_fd, returnMatrix = TRUE)
as_fertility_m <- eval.fd(t.sq, asia_fd, returnMatrix = TRUE)

# Create functional time series objects
tf <- fts(x = t.sq, y=total_fertility_m, yname = "TFR", xname = "Years")
```

```
## Warning in fts(x = t.sq, y = total_fertility_m, yname = "TFR", xname =
## "Years"): Please assign column name for the data matrix.
```

```
euf <- fts(x = t.sq, y=eu_fertility_m, yname = "TFR", xname = "Years")
```

```
## Warning in fts(x = t.sq, y = eu_fertility_m, yname = "TFR", xname = "Years"):  
## Please assign column name for the data matrix.
```

```
asf <- fts(x = t.sq, y=as_fertility_m, yname = "TFR", xname = "Years")
```

```
## Warning in fts(x = t.sq, y = as_fertility_m, yname = "TFR", xname = "Years"):  
## Please assign column name for the data matrix.
```

```
# Create functional time series modelling with the most major PCA's  
modell1.ftsm <- ftsm(tf, order=2)
```

```
## Warning in fts(y$x, basis_obj, start = ytsp[1], frequency = ytsp[3], xname =  
## y$xname, : Please assign column name for the data matrix.
```

```
## Warning in fts(y$x, y$y - fits$y, start = ytsp[1], frequency = ytsp[3], :  
## Please assign column name for the data matrix.
```

```
## Warning in fts(y$x, y$y, xname = y$xname, yname = y$yname): Please assign  
## column name for the data matrix.
```

```
summary(modell1.ftsm)
```

```
## Functional time series model  
##  
## Call: ftsm(y = tf, order = 2)  
##  
## Main effects: Mean  
## Order of interaction: 2  
##      y: TFR  
##      x: Years  
##  
## Averages across x:  
##      ME      MSE      MPE      MAPE  
## 0.00000 0.00770 0.01891 0.09130  
##  
## Averages across time:  
##      IE      ISE      IPE      IAPE  
## 0.00001 0.49025 1.25845 6.12905
```

```
model2.ftsm <- ftsm(euf, order=3)
```

```
## Warning in fts(y$x, basis_obj, start = ytsp[1], frequency = ytsp[3], xname =  
## y$xname, : Please assign column name for the data matrix.
```

```
## Warning in fts(y$x, y$y - fits$y, start = ytsp[1], frequency = ytsp[3], :  
## Please assign column name for the data matrix.
```

```
## Warning in fts(y$x, y$y, xname = y$xname, yname = y$yname): Please assign  
## column name for the data matrix.
```

```
summary(model2.ftsm)
```

```
## Functional time series model
##
## Call: ftsm(y = euf, order = 3)
##
## Main effects: Mean
## Order of interaction: 3
##   y: TFR
##   x: Years
##
## Averages across x:
##      ME      MSE      MPE      MAPE
## 0.00000 0.00156 0.00562 0.04851
##
## Averages across time:
##      IE      ISE      IPE      IAPE
## 0.00003 0.09633 0.36123 3.21666
```

```
model3.ftsm <- ftsm(asf, order=2)
```

```
## Warning in fts(y$x, basis_obj, start = ytsp[1], frequency = ytsp[3], xname =
## y$xname, : Please assign column name for the data matrix.
```

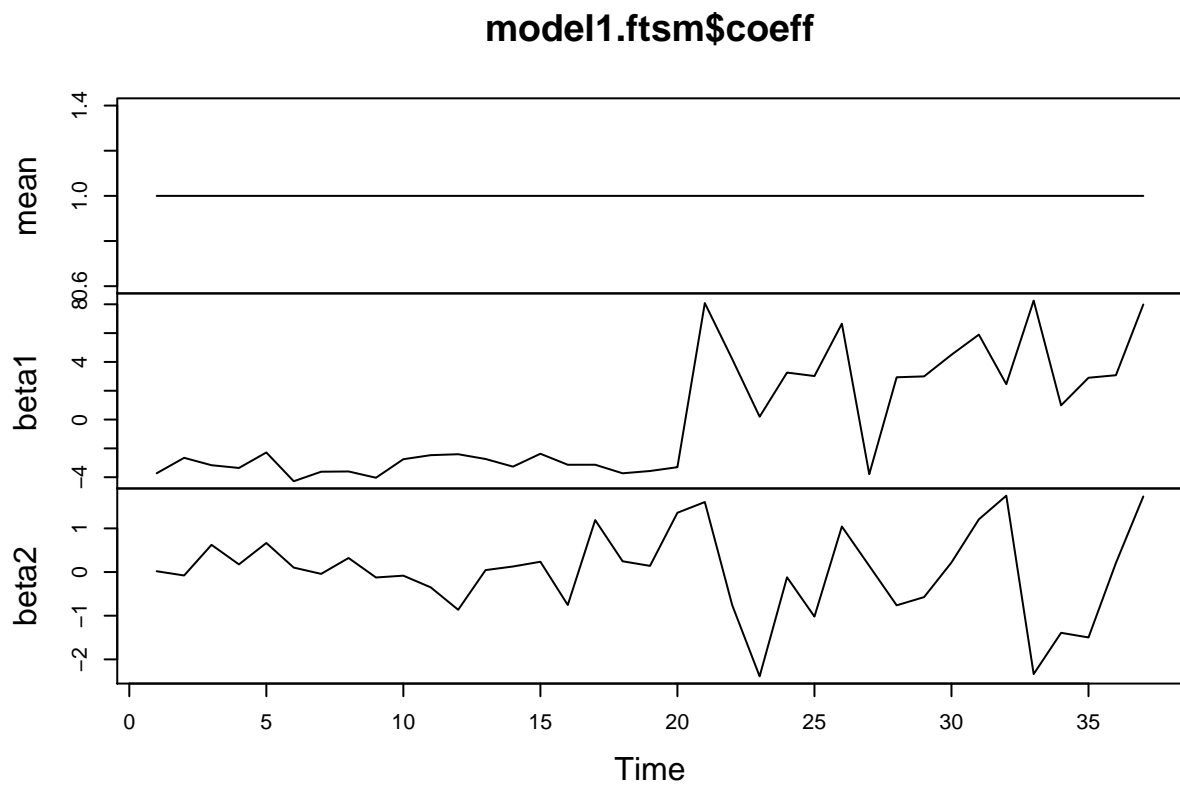
```
## Warning in fts(y$x, y$y - fits$y, start = ytsp[1], frequency = ytsp[3], :
## Please assign column name for the data matrix.
```

```
## Warning in fts(y$x, y$y, xname = y$xname, yname = y$yname): Please assign
## column name for the data matrix.
```

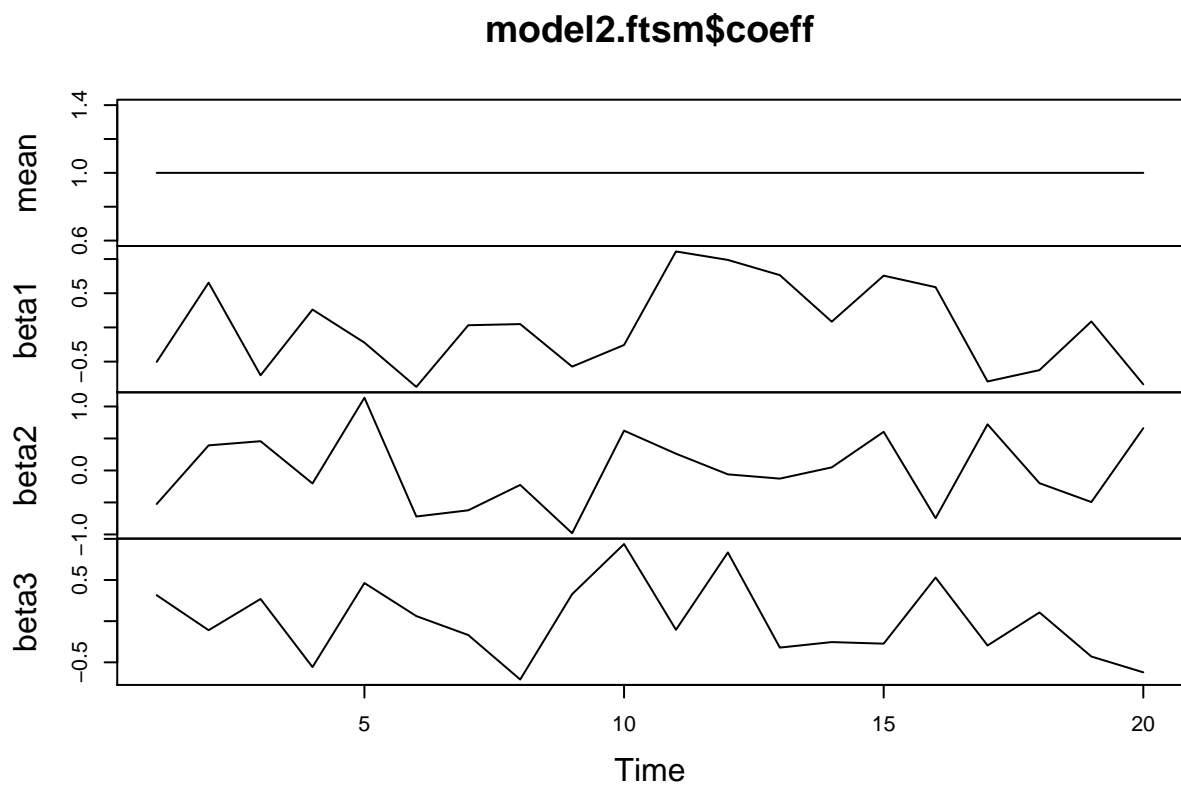
```
summary(model3.ftsm)
```

```
## Functional time series model
##
## Call: ftsm(y = asf, order = 2)
##
## Main effects: Mean
## Order of interaction: 2
##   y: TFR
##   x: Years
##
## Averages across x:
##      ME      MSE      MPE      MAPE
## 0.00000 0.00823 0.00644 0.04894
##
## Averages across time:
##      IE      ISE      IPE      IAPE
## -0.00001 0.51173 0.40742 3.22255
```

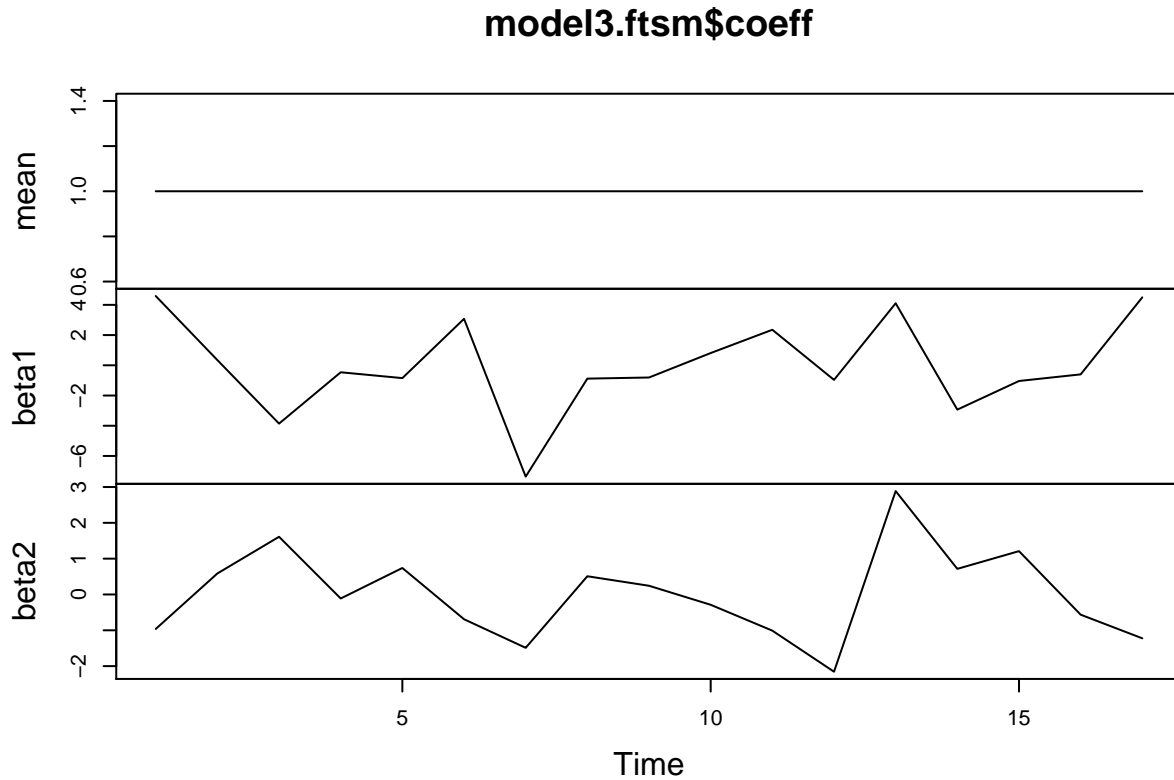
```
plot(model1.ftsm$coeff)
```



```
plot(model2.ftsm$coeff)
```

```
plot(model3.ftsm$coeff)
```



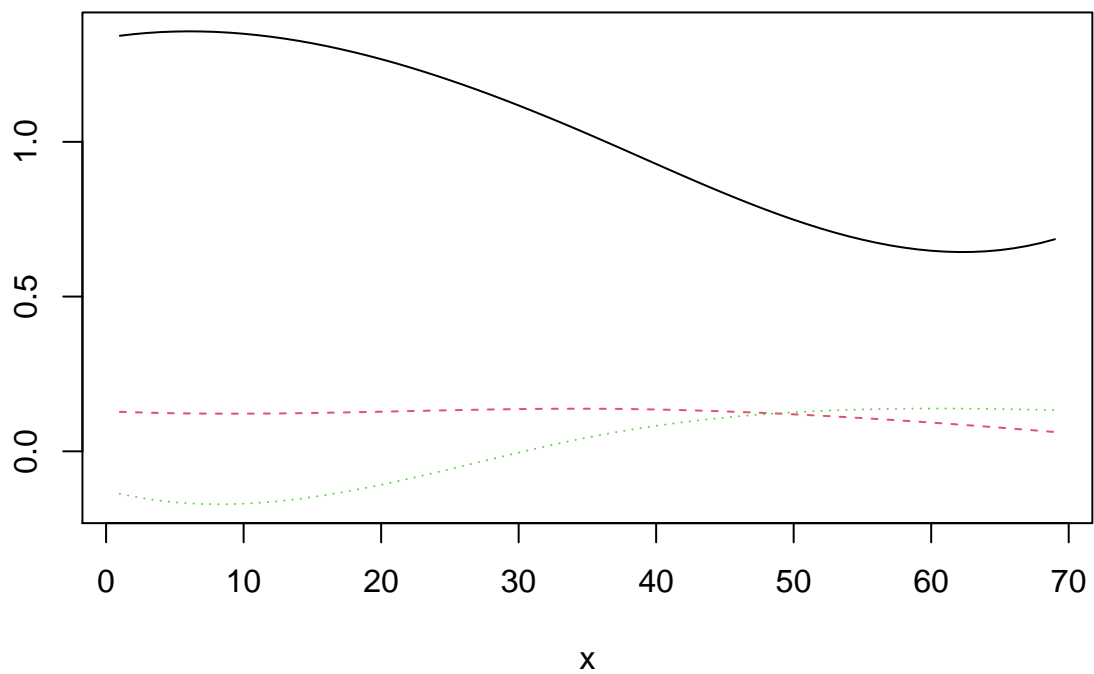
Then, we inspected the basis matrices of all the 3 models to further inspect the variability captured by the PCA's.

The first model for the combined country data shows that the general mean function, indicated by the black line, exhibits a declining trend. The dotted red line, representing Phi1, appears relatively flat across the dataset. However, it is crucial to note that Phi1 captures approximately 91.83% of the dataset's variability, indicating its significant but subtle influence. The green line, Phi2, while only accounting for about 5.41% of the variability, shows minor fluctuations, particularly noticeable after year 20. These fluctuations, though small, highlight additional patterns or deviations not encompassed by Phi1

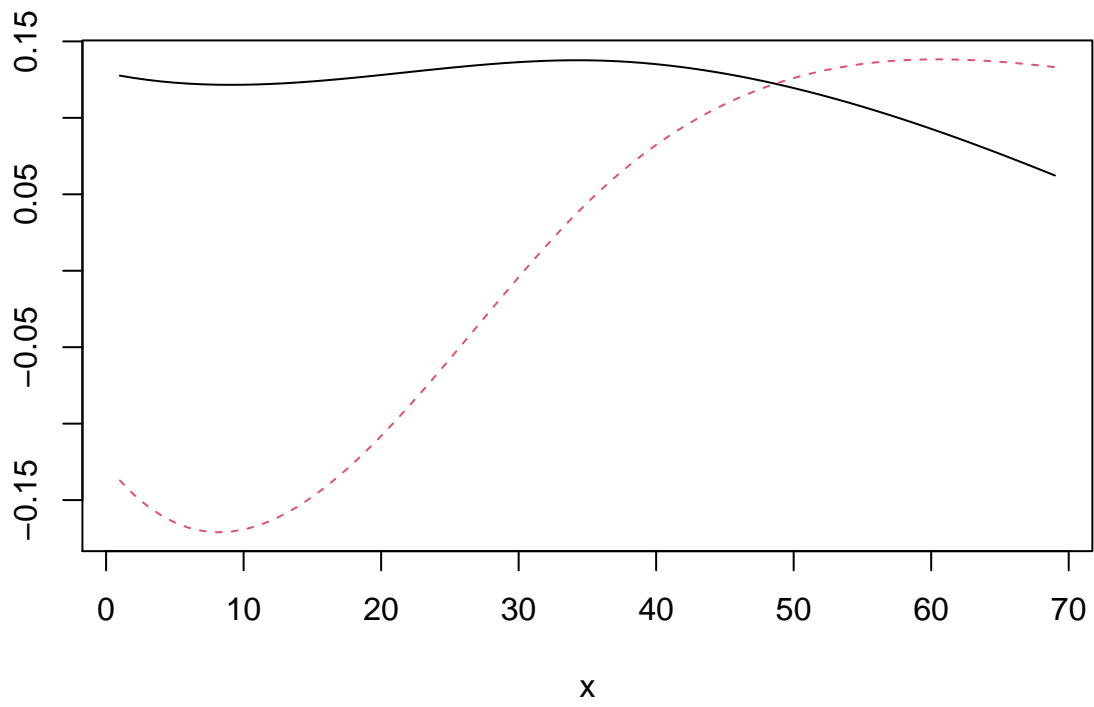
The second model for European country data exhibits a mean function that continues the declining trend observed in the combined country model. Unlike the combined model, all three principal components in this European dataset display noticeable fluctuations, with each component significantly contributing to the data's variability. The variance proportions for these components are 38.76% for the first PCA, 31.97% for the second PCA, and 20.06% for the third PCA. Contrary to the initial observation, the first PCA, not the third, captures the most variability, though all components show significant variance indicating diverse underlying patterns or influences. The third PCA, while the least in terms of variance explained compared to the first two, still shows substantial variability, reflecting more complex dynamics or interactions within the European data set.

The third model for Asian countries displays a consistent declining mean function. Contrary to initial observations, the first principal component (PCA) significantly shapes the dataset, explaining 82% of the variability. The second PCA, accounting for 13.46% of the variance, exhibits notable fluctuations around year 20, highlighting additional, nuanced dynamics not captured by the first PCA.

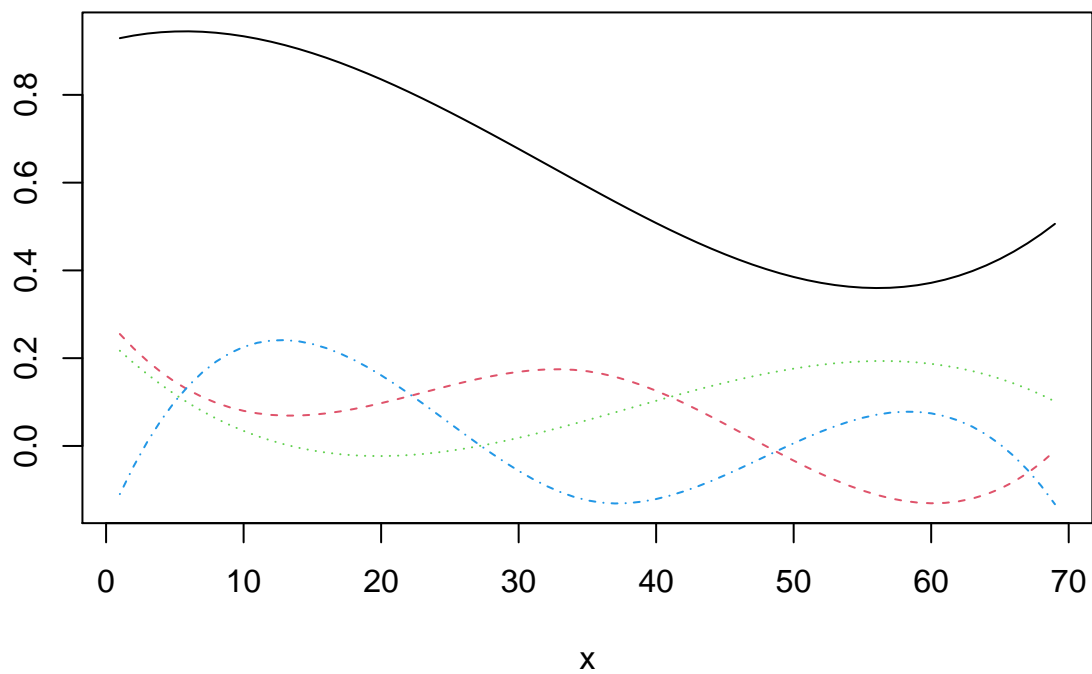
```
matplot(model1.ftsm$basis, type="l")
```



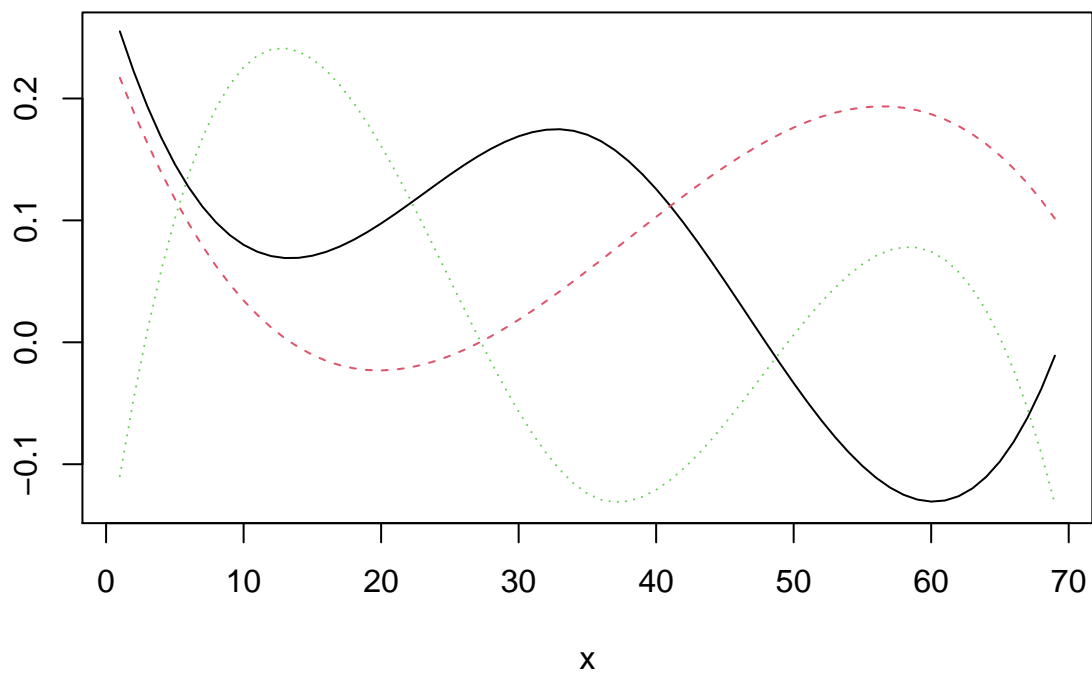
```
matplot(model11.ftsm$basis[, -1], type="l")
```



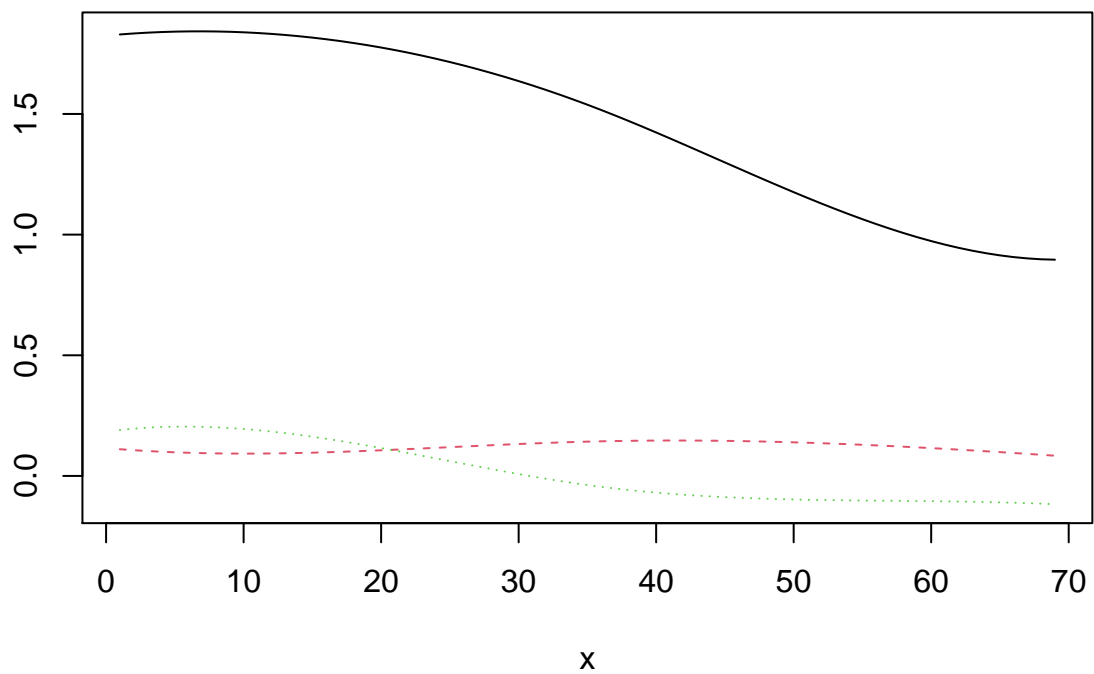
```
matplot(model2.ftsm$basis, type="l")
```



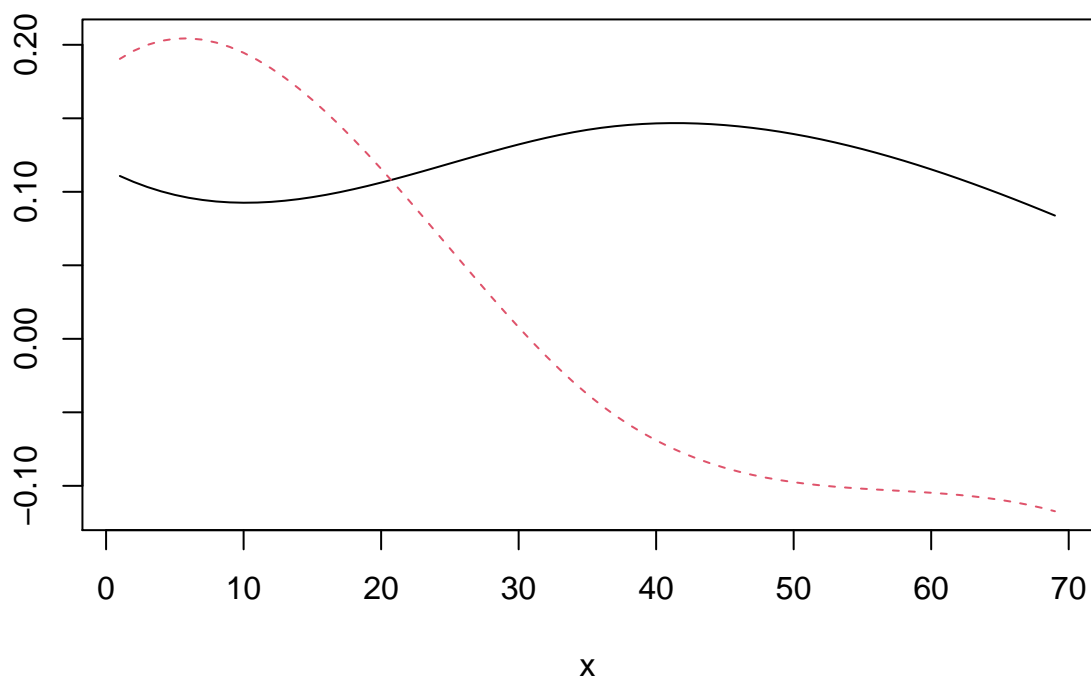
```
matplot(model2.ftsm$basis[, -1], type="l")
```



```
matplot(model3.ftsm$basis, type="l")
```



```
matplot(model3.ftsm$basis[, -1], type="l")
```



```
model1.ftsm$varprop
```

```
## [1] 0.9182699 0.0540801
```

```
model2.ftsm$varprop
```

```
## [1] 0.3875721 0.3196660 0.2006403
```

```
model3.ftsm$varprop
```

```
## [1] 0.8199521 0.1345598
```

Functional timeseries forecasting

We extended our time series modeling by performing multistep forecasts using functional time series models developed from combined, European, and Asian datasets. The models predict future values by leveraging historical trends captured in the data.

Forecasts were generated in a matrix format, with each row representing a base year and each column a future forecast period. For instance, a forecast for the 38th period based on the base year 1950 predicts a value of 2.6, suggesting expected changes in the metric under study.

In the combined dataset, predictions show a stable trend for initial periods, becoming more variable after 20 years. This increased variability, especially in the first PCA component, might reflect influences from Asian country data dynamics.

In the European country dataset, forecasts indicate slight variations across all principal components, showing less variability compared to the combined dataset.

In the Asian country dataset, both principal components exhibit strong fluctuations, indicating a higher level of variability throughout the forecast periods.

These forecasts provide valuable insights for anticipating future trends and preparing strategies accordingly. The variability in predictions, particularly in the Asian dataset, underscores the dynamic nature of the data and potential challenges in forecasting.

```
multistep <- predict(model1.ftsm, h=3)
multistep
```

```
## Functional time series forecasts
##   for 38 to 40
##   using ets method
##   based on
## Functional time series model
##
## Call: ftsm(y = tf, order = 2)
##
## Main effects: Mean
## Order of interaction: 2
##   y: TFR
##   x: Years
```

```
multistep$mean$y
```

```
##           38           39           40
## 1950.001 2.037523 2.074435 2.111347
## 1951.001 2.034479 2.070957 2.107435
## 1952.001 2.031562 2.067674 2.103787
## 1953.001 2.028732 2.064545 2.100357
## 1954.001 2.025953 2.061527 2.097102
## 1955.001 2.023185 2.058579 2.093973
## 1956.001 2.020390 2.055659 2.090928
## 1957.001 2.017530 2.052725 2.087919
## 1958.001 2.014566 2.049734 2.084902
## 1959.001 2.011460 2.046646 2.081832
## 1960.001 2.008173 2.043418 2.078662
## 1961.001 2.004668 2.040008 2.075348
## 1962.001 2.000905 2.036375 2.071845
## 1963.001 1.996846 2.032476 2.068105
## 1964.001 1.992453 2.028269 2.064086
## 1965.001 1.987687 2.023714 2.059740
## 1966.001 1.982510 2.018766 2.055023
## 1967.001 1.976884 2.013386 2.049889
## 1968.001 1.970770 2.007532 2.044293
## 1969.001 1.964128 2.001159 2.038189
## 1970.001 1.956923 1.994228 2.031533
```

```

## 1971.001 1.949115 1.986697 2.024278
## 1972.001 1.940664 1.978521 2.016378
## 1973.001 1.931534 1.969662 2.007791
## 1974.001 1.921686 1.960077 1.998469
## 1975.001 1.911079 1.949722 1.988365
## 1976.001 1.899679 1.938558 1.977438
## 1977.001 1.887445 1.926542 1.965640
## 1978.001 1.874337 1.913630 1.952924
## 1979.001 1.860321 1.899785 1.939248
## 1980.001 1.845355 1.884960 1.924566
## 1981.001 1.829400 1.869115 1.908829
## 1982.001 1.812422 1.852209 1.891997
## 1983.001 1.794379 1.834200 1.874021
## 1984.001 1.775231 1.815043 1.854854
## 1985.001 1.754955 1.794711 1.834467
## 1986.001 1.733595 1.773249 1.812903
## 1987.001 1.711243 1.750750 1.790257
## 1988.001 1.687993 1.727308 1.766623
## 1989.001 1.663934 1.703013 1.742093
## 1990.001 1.639156 1.677958 1.716760
## 1991.001 1.613753 1.652236 1.690719
## 1992.001 1.587814 1.625938 1.664061
## 1993.001 1.561431 1.599156 1.636881
## 1994.001 1.534696 1.571984 1.609272
## 1995.001 1.507699 1.544513 1.581327
## 1996.001 1.480531 1.516835 1.553138
## 1997.001 1.453285 1.489043 1.524801
## 1998.001 1.426050 1.461228 1.496406
## 1999.001 1.398918 1.433483 1.468049
## 2000.001 1.371980 1.405901 1.439821
## 2001.001 1.345328 1.378573 1.411818
## 2002.001 1.319052 1.351591 1.384131
## 2003.001 1.293243 1.325048 1.356852
## 2004.001 1.267995 1.299037 1.330079
## 2005.001 1.243396 1.273649 1.303901
## 2006.001 1.219537 1.248974 1.278411
## 2007.001 1.196513 1.225110 1.253707
## 2008.001 1.174412 1.202145 1.229878
## 2009.001 1.153324 1.180170 1.207016
## 2010.001 1.133345 1.159282 1.185220
## 2011.001 1.114562 1.139570 1.164578
## 2012.001 1.097064 1.121124 1.145183
## 2013.001 1.080951 1.104043 1.127135
## 2014.001 1.066306 1.088413 1.110521
## 2015.001 1.053220 1.074326 1.095431
## 2016.001 1.041794 1.061882 1.081971
## 2017.001 1.032108 1.051165 1.070222
## 2018.001 1.024254 1.042266 1.060278

```

```
multistep$lower$y
```

```

##           38           39           40
## 1950.001 1.4666575 1.5035694 1.5404812
## 1951.001 1.4938166 1.5302946 1.5667727

```

```

## 1952.001 1.5147484 1.5508612 1.5869740
## 1953.001 1.5302622 1.5660748 1.6018874
## 1954.001 1.5412256 1.5767999 1.6123742
## 1955.001 1.5485076 1.5839017 1.6192958
## 1956.001 1.5529250 1.5881936 1.6234622
## 1957.001 1.5552009 1.5903954 1.6255899
## 1958.001 1.5559412 1.5911092 1.6262771
## 1959.001 1.5556245 1.5908102 1.6259958
## 1960.001 1.5546044 1.5898488 1.6250930
## 1961.001 1.5531229 1.5884630 1.6238031
## 1962.001 1.5513210 1.5867907 1.6222603
## 1963.001 1.5492543 1.5848839 1.6205135
## 1964.001 1.5469123 1.5827287 1.6185450
## 1965.001 1.5442282 1.5802546 1.6162810
## 1966.001 1.5410927 1.5773491 1.6136054
## 1967.001 1.5373718 1.5738744 1.6103770
## 1968.001 1.5329120 1.5696738 1.6064354
## 1969.001 1.5275530 1.5645832 1.6016134
## 1970.001 1.5211432 1.5584478 1.5957524
## 1971.001 1.5135417 1.5511231 1.5887045
## 1972.001 1.5046288 1.5424860 1.5803431
## 1973.001 1.4943188 1.5324471 1.5705753
## 1974.001 1.4825548 1.5209462 1.5593375
## 1975.001 1.4693164 1.5079593 1.5466021
## 1976.001 1.4546244 1.4935038 1.5323831
## 1977.001 1.4385293 1.4776267 1.5167240
## 1978.001 1.4211136 1.4604069 1.4997002
## 1979.001 1.4024906 1.4419544 1.4814181
## 1980.001 1.3827876 1.4223929 1.4619982
## 1981.001 1.3621472 1.4018615 1.4415757
## 1982.001 1.3407230 1.3805103 1.4202976
## 1983.001 1.3186613 1.3584822 1.3983032
## 1984.001 1.2961036 1.3359152 1.3757267
## 1985.001 1.2731789 1.3129350 1.3526912
## 1986.001 1.2499681 1.2896224 1.3292766
## 1987.001 1.2265236 1.2660303 1.3055370
## 1988.001 1.2028815 1.2421964 1.2815113
## 1989.001 1.1790616 1.2181412 1.2572207
## 1990.001 1.1550821 1.1938839 1.2326857
## 1991.001 1.1309617 1.1694445 1.2079273
## 1992.001 1.1067192 1.1448426 1.1829660
## 1993.001 1.0823837 1.1201084 1.1578331
## 1994.001 1.0579946 1.0952825 1.1325703
## 1995.001 1.0336019 1.0704156 1.1072292
## 1996.001 1.0092718 1.0455752 1.0818785
## 1997.001 0.9850851 1.0208430 1.0566009
## 1998.001 0.9611383 0.9963166 1.0314948
## 1999.001 0.9375430 0.9721086 1.0066741
## 2000.001 0.9144239 0.9483448 0.9822655
## 2001.001 0.8919195 0.9251646 0.9584096
## 2002.001 0.8701749 0.9027142 0.9352534
## 2003.001 0.8493397 0.8811443 0.9129488
## 2004.001 0.8295673 0.8606092 0.8916511
## 2005.001 0.8110001 0.8412525 0.8715048

```

```
## 2006.001 0.7937674 0.8232044 0.8526414
## 2007.001 0.7779808 0.8065776 0.8351744
## 2008.001 0.7637091 0.7914419 0.8191747
## 2009.001 0.7509732 0.7778193 0.8046654
## 2010.001 0.7397333 0.7656709 0.7916084
## 2011.001 0.7298467 0.7548551 0.7798635
## 2012.001 0.7210558 0.7451155 0.7691751
## 2013.001 0.7129633 0.7360554 0.7591476
## 2014.001 0.7049776 0.7270848 0.7491919
## 2015.001 0.6963161 0.7174217 0.7385273
## 2016.001 0.6860174 0.7061059 0.7261943
## 2017.001 0.6729609 0.6920178 0.7110746
## 2018.001 0.6560006 0.6740125 0.6920243
```

```
multistep$upper$y
```

```
##          38          39          40
## 1950.001 2.608389 2.645301 2.682213
## 1951.001 2.575142 2.611620 2.648098
## 1952.001 2.548375 2.584488 2.620600
## 1953.001 2.527202 2.563015 2.598827
## 1954.001 2.510680 2.546255 2.581829
## 1955.001 2.497862 2.533257 2.568651
## 1956.001 2.487856 2.523124 2.558393
## 1957.001 2.479859 2.515054 2.550248
## 1958.001 2.473191 2.508359 2.543527
## 1959.001 2.467296 2.502482 2.537668
## 1960.001 2.461742 2.496987 2.532231
## 1961.001 2.456213 2.491553 2.526893
## 1962.001 2.450490 2.485959 2.521429
## 1963.001 2.444437 2.480067 2.515697
## 1964.001 2.437993 2.473810 2.509626
## 1965.001 2.431147 2.467173 2.503200
## 1966.001 2.423927 2.460184 2.496440
## 1967.001 2.416396 2.452898 2.489401
## 1968.001 2.408628 2.445389 2.482151
## 1969.001 2.400704 2.437734 2.474764
## 1970.001 2.392703 2.430008 2.467313
## 1971.001 2.384689 2.422270 2.459852
## 1972.001 2.376699 2.414556 2.452413
## 1973.001 2.368749 2.406878 2.445006
## 1974.001 2.360817 2.399208 2.437600
## 1975.001 2.352842 2.391485 2.430128
## 1976.001 2.344734 2.383613 2.422493
## 1977.001 2.336360 2.375458 2.414555
## 1978.001 2.327561 2.366854 2.406148
## 1979.001 2.318151 2.357615 2.397079
## 1980.001 2.307922 2.347528 2.387133
## 1981.001 2.296653 2.336368 2.376082
## 1982.001 2.284121 2.323908 2.363696
## 1983.001 2.270096 2.309917 2.349738
## 1984.001 2.254359 2.294171 2.333982
## 1985.001 2.236730 2.276487 2.316243
## 1986.001 2.217221 2.256876 2.296530
```

```
## 1987.001 2.195962 2.235469 2.274976
## 1988.001 2.173105 2.212420 2.251735
## 1989.001 2.148805 2.187885 2.226965
## 1990.001 2.123230 2.162032 2.200834
## 1991.001 2.096545 2.135028 2.173511
## 1992.001 2.068910 2.107033 2.145157
## 1993.001 2.040479 2.078204 2.115929
## 1994.001 2.011398 2.048686 2.085974
## 1995.001 1.981796 2.018610 2.055424
## 1996.001 1.951791 1.988095 2.024398
## 1997.001 1.921484 1.957242 1.993000
## 1998.001 1.890961 1.926139 1.961318
## 1999.001 1.860293 1.894858 1.929424
## 2000.001 1.829535 1.863456 1.897377
## 2001.001 1.798736 1.831981 1.865226
## 2002.001 1.767929 1.800469 1.833008
## 2003.001 1.737147 1.768951 1.800756
## 2004.001 1.706422 1.737464 1.768506
## 2005.001 1.675792 1.706045 1.736297
## 2006.001 1.645307 1.674744 1.704181
## 2007.001 1.615046 1.643643 1.672240
## 2008.001 1.585115 1.612848 1.640581
## 2009.001 1.555674 1.582520 1.609366
## 2010.001 1.526956 1.552894 1.578831
## 2011.001 1.499276 1.524285 1.549293
## 2012.001 1.473072 1.497132 1.521191
## 2013.001 1.448938 1.472031 1.495123
## 2014.001 1.427635 1.449742 1.471849
## 2015.001 1.410124 1.431230 1.452336
## 2016.001 1.397570 1.417658 1.437747
## 2017.001 1.391255 1.410312 1.429369
## 2018.001 1.392508 1.410519 1.428531
```

```
multistep <- predict(model2.ftsm, h=3)
multistep
```

```
## Functional time series forecasts
##   for 21 to 23
##   using ets method
##   based on
## Functional time series model
##
## Call: ftsm(y = euf, order = 3)
##
## Main effects: Mean
## Order of interaction: 3
##   y: TFR
##   x: Years
```

```
multistep$mean$y
```

```
##           21           22           23
## 1950.001 0.9297591 0.9297591 0.9297591
```

```

## 1951.001 0.9354253 0.9354253 0.9354253
## 1952.001 0.9396357 0.9396357 0.9396357
## 1953.001 0.9424378 0.9424378 0.9424378
## 1954.001 0.9438717 0.9438717 0.9438717
## 1955.001 0.9439934 0.9439934 0.9439934
## 1956.001 0.9428501 0.9428501 0.9428501
## 1957.001 0.9404833 0.9404833 0.9404833
## 1958.001 0.9369481 0.9369481 0.9369481
## 1959.001 0.9322911 0.9322911 0.9322911
## 1960.001 0.9265555 0.9265555 0.9265555
## 1961.001 0.9197950 0.9197950 0.9197950
## 1962.001 0.9120562 0.9120562 0.9120562
## 1963.001 0.9033834 0.9033834 0.9033834
## 1964.001 0.8938294 0.8938294 0.8938294
## 1965.001 0.8834407 0.8834407 0.8834407
## 1966.001 0.8722625 0.8722625 0.8722625
## 1967.001 0.8603467 0.8603467 0.8603467
## 1968.001 0.8477398 0.8477398 0.8477398
## 1969.001 0.8344879 0.8344879 0.8344879
## 1970.001 0.8206420 0.8206420 0.8206420
## 1971.001 0.8062488 0.8062488 0.8062488
## 1972.001 0.7913550 0.7913550 0.7913550
## 1973.001 0.7760108 0.7760108 0.7760108
## 1974.001 0.7602630 0.7602630 0.7602630
## 1975.001 0.7441590 0.7441590 0.7441590
## 1976.001 0.7277482 0.7277482 0.7277482
## 1977.001 0.7110777 0.7110777 0.7110777
## 1978.001 0.6941952 0.6941952 0.6941952
## 1979.001 0.6771496 0.6771496 0.6771496
## 1980.001 0.6599881 0.6599881 0.6599881
## 1981.001 0.6427590 0.6427590 0.6427590
## 1982.001 0.6255102 0.6255102 0.6255102
## 1983.001 0.6082896 0.6082896 0.6082896
## 1984.001 0.5911455 0.5911455 0.5911455
## 1985.001 0.5741265 0.5741265 0.5741265
## 1986.001 0.5572929 0.5572929 0.5572929
## 1987.001 0.5407100 0.5407100 0.5407100
## 1988.001 0.5244421 0.5244421 0.5244421
## 1989.001 0.5085555 0.5085555 0.5085555
## 1990.001 0.4931155 0.4931155 0.4931155
## 1991.001 0.4781857 0.4781857 0.4781857
## 1992.001 0.4638335 0.4638335 0.4638335
## 1993.001 0.4501235 0.4501235 0.4501235
## 1994.001 0.4371191 0.4371191 0.4371191
## 1995.001 0.4248886 0.4248886 0.4248886
## 1996.001 0.4134958 0.4134958 0.4134958
## 1997.001 0.4030039 0.4030039 0.4030039
## 1998.001 0.3934823 0.3934823 0.3934823
## 1999.001 0.3849940 0.3849940 0.3849940
## 2000.001 0.3776019 0.3776019 0.3776019
## 2001.001 0.3713765 0.3713765 0.3713765
## 2002.001 0.3663798 0.3663798 0.3663798
## 2003.001 0.3626747 0.3626747 0.3626747
## 2004.001 0.3603329 0.3603329 0.3603329

```

```
## 2005.001 0.3594150 0.3594150 0.3594150
## 2006.001 0.3599839 0.3599839 0.3599839
## 2007.001 0.3621130 0.3621130 0.3621130
## 2008.001 0.3658611 0.3658611 0.3658611
## 2009.001 0.3712914 0.3712914 0.3712914
## 2010.001 0.3784786 0.3784786 0.3784786
## 2011.001 0.3874800 0.3874800 0.3874800
## 2012.001 0.3983588 0.3983588 0.3983588
## 2013.001 0.4111915 0.4111915 0.4111915
## 2014.001 0.4260332 0.4260332 0.4260332
## 2015.001 0.4429477 0.4429477 0.4429477
## 2016.001 0.4620131 0.4620131 0.4620131
## 2017.001 0.4832824 0.4832824 0.4832824
## 2018.001 0.5068199 0.5068199 0.5068199
```

```
multistep$lower$y
```

```
##          21          22          23
## 1950.001 0.6188955 0.6188955 0.6188955
## 1951.001 0.6694905 0.6694905 0.6694905
## 1952.001 0.7109404 0.7109404 0.7109404
## 1953.001 0.7431623 0.7431623 0.7431623
## 1954.001 0.7662652 0.7662652 0.7662652
## 1955.001 0.7807991 0.7807991 0.7807991
## 1956.001 0.7878737 0.7878737 0.7878737
## 1957.001 0.7890577 0.7890577 0.7890577
## 1958.001 0.7860715 0.7860715 0.7860715
## 1959.001 0.7804368 0.7804368 0.7804368
## 1960.001 0.7733151 0.7733151 0.7733151
## 1961.001 0.7655109 0.7655109 0.7655109
## 1962.001 0.7575147 0.7575147 0.7575147
## 1963.001 0.7495833 0.7495833 0.7495833
## 1964.001 0.7418121 0.7418121 0.7418121
## 1965.001 0.7341648 0.7341648 0.7341648
## 1966.001 0.7265076 0.7265076 0.7265076
## 1967.001 0.7186380 0.7186380 0.7186380
## 1968.001 0.7102879 0.7102879 0.7102879
## 1969.001 0.7011444 0.7011444 0.7011444
## 1970.001 0.6908775 0.6908775 0.6908775
## 1971.001 0.6791627 0.6791627 0.6791627
## 1972.001 0.6657289 0.6657289 0.6657289
## 1973.001 0.6504092 0.6504092 0.6504092
## 1974.001 0.6331728 0.6331728 0.6331728
## 1975.001 0.6141438 0.6141438 0.6141438
## 1976.001 0.5935870 0.5935870 0.5935870
## 1977.001 0.5718671 0.5718671 0.5718671
## 1978.001 0.5494077 0.5494077 0.5494077
## 1979.001 0.5266534 0.5266534 0.5266534
## 1980.001 0.5040422 0.5040422 0.5040422
## 1981.001 0.4819921 0.4819921 0.4819921
## 1982.001 0.4608924 0.4608924 0.4608924
## 1983.001 0.4410998 0.4410998 0.4410998
## 1984.001 0.4229352 0.4229352 0.4229352
## 1985.001 0.4066507 0.4066507 0.4066507
```

```
## 1986.001 0.3922181 0.3922181 0.3922181
## 1987.001 0.3794402 0.3794402 0.3794402
## 1988.001 0.3680717 0.3680717 0.3680717
## 1989.001 0.3578245 0.3578245 0.3578245
## 1990.001 0.3483644 0.3483644 0.3483644
## 1991.001 0.3393141 0.3393141 0.3393141
## 1992.001 0.3302695 0.3302695 0.3302695
## 1993.001 0.3208202 0.3208202 0.3208202
## 1994.001 0.3105978 0.3105978 0.3105978
## 1995.001 0.2993449 0.2993449 0.2993449
## 1996.001 0.2869671 0.2869671 0.2869671
## 1997.001 0.2735694 0.2735694 0.2735694
## 1998.001 0.2594502 0.2594502 0.2594502
## 1999.001 0.2450375 0.2450375 0.2450375
## 2000.001 0.2308317 0.2308317 0.2308317
## 2001.001 0.2173578 0.2173578 0.2173578
## 2002.001 0.2051151 0.2051151 0.2051151
## 2003.001 0.1945684 0.1945684 0.1945684
## 2004.001 0.1861497 0.1861497 0.1861497
## 2005.001 0.1802368 0.1802368 0.1802368
## 2006.001 0.1771666 0.1771666 0.1771666
## 2007.001 0.1772411 0.1772411 0.1772411
## 2008.001 0.1806960 0.1806960 0.1806960
## 2009.001 0.1877071 0.1877071 0.1877071
## 2010.001 0.1983738 0.1983738 0.1983738
## 2011.001 0.2126488 0.2126488 0.2126488
## 2012.001 0.2302943 0.2302943 0.2302943
## 2013.001 0.2507769 0.2507769 0.2507769
## 2014.001 0.2730694 0.2730694 0.2730694
## 2015.001 0.2954881 0.2954881 0.2954881
## 2016.001 0.3156328 0.3156328 0.3156328
## 2017.001 0.3307255 0.3307255 0.3307255
## 2018.001 0.3385824 0.3385824 0.3385824
```

```
multistep$upper$y
```

```
##          21          22          23
## 1950.001 1.2406227 1.2406227 1.2406227
## 1951.001 1.2013601 1.2013601 1.2013601
## 1952.001 1.1683311 1.1683311 1.1683311
## 1953.001 1.1417134 1.1417134 1.1417134
## 1954.001 1.1214781 1.1214781 1.1214781
## 1955.001 1.1071877 1.1071877 1.1071877
## 1956.001 1.0978265 1.0978265 1.0978265
## 1957.001 1.0919089 1.0919089 1.0919089
## 1958.001 1.0878247 1.0878247 1.0878247
## 1959.001 1.0841455 1.0841455 1.0841455
## 1960.001 1.0797958 1.0797958 1.0797958
## 1961.001 1.0740790 1.0740790 1.0740790
## 1962.001 1.0665977 1.0665977 1.0665977
## 1963.001 1.0571836 1.0571836 1.0571836
## 1964.001 1.0458467 1.0458467 1.0458467
## 1965.001 1.0327166 1.0327166 1.0327166
## 1966.001 1.0180174 1.0180174 1.0180174
```



```

## 1967.001 1.0020553 1.0020553 1.0020553
## 1968.001 0.9851917 0.9851917 0.9851917
## 1969.001 0.9678315 0.9678315 0.9678315
## 1970.001 0.9504066 0.9504066 0.9504066
## 1971.001 0.9333349 0.9333349 0.9333349
## 1972.001 0.9169811 0.9169811 0.9169811
## 1973.001 0.9016124 0.9016124 0.9016124
## 1974.001 0.8873531 0.8873531 0.8873531
## 1975.001 0.8741742 0.8741742 0.8741742
## 1976.001 0.8619094 0.8619094 0.8619094
## 1977.001 0.8502882 0.8502882 0.8502882
## 1978.001 0.8389828 0.8389828 0.8389828
## 1979.001 0.8276458 0.8276458 0.8276458
## 1980.001 0.8159340 0.8159340 0.8159340
## 1981.001 0.8035258 0.8035258 0.8035258
## 1982.001 0.7901279 0.7901279 0.7901279
## 1983.001 0.7754795 0.7754795 0.7754795
## 1984.001 0.7593557 0.7593557 0.7593557
## 1985.001 0.7416023 0.7416023 0.7416023
## 1986.001 0.7223677 0.7223677 0.7223677
## 1987.001 0.7019798 0.7019798 0.7019798
## 1988.001 0.6808125 0.6808125 0.6808125
## 1989.001 0.6592866 0.6592866 0.6592866
## 1990.001 0.6378665 0.6378665 0.6378665
## 1991.001 0.6170573 0.6170573 0.6170573
## 1992.001 0.5973975 0.5973975 0.5973975
## 1993.001 0.5794268 0.5794268 0.5794268
## 1994.001 0.5636404 0.5636404 0.5636404
## 1995.001 0.5504322 0.5504322 0.5504322
## 1996.001 0.5400244 0.5400244 0.5400244
## 1997.001 0.5324384 0.5324384 0.5324384
## 1998.001 0.5275145 0.5275145 0.5275145
## 1999.001 0.5249505 0.5249505 0.5249505
## 2000.001 0.5243721 0.5243721 0.5243721
## 2001.001 0.5253953 0.5253953 0.5253953
## 2002.001 0.5276445 0.5276445 0.5276445
## 2003.001 0.5307810 0.5307810 0.5307810
## 2004.001 0.5345161 0.5345161 0.5345161
## 2005.001 0.5385932 0.5385932 0.5385932
## 2006.001 0.5428013 0.5428013 0.5428013
## 2007.001 0.5469850 0.5469850 0.5469850
## 2008.001 0.5510262 0.5510262 0.5510262
## 2009.001 0.5548757 0.5548757 0.5548757
## 2010.001 0.5585835 0.5585835 0.5585835
## 2011.001 0.5623111 0.5623111 0.5623111
## 2012.001 0.5664233 0.5664233 0.5664233
## 2013.001 0.5716060 0.5716060 0.5716060
## 2014.001 0.5789970 0.5789970 0.5789970
## 2015.001 0.5904073 0.5904073 0.5904073
## 2016.001 0.6083935 0.6083935 0.6083935
## 2017.001 0.6358394 0.6358394 0.6358394
## 2018.001 0.6750574 0.6750574 0.6750574

```

```
multistep <- predict(model3.ftsm, h=3)
multistep
```

```
## Functional time series forecasts
##   for 18 to 20
##   using ets method
##   based on
## Functional time series model
##
## Call: ftsm(y = asf, order = 2)
##
## Main effects: Mean
## Order of interaction: 2
##   y: TFR
##   x: Years
```

```
multistep$mean$y
```

```
##           18           19           20
## 1950.001 1.8295551 1.8295551 1.8295551
## 1951.001 1.8336226 1.8336226 1.8336226
## 1952.001 1.8369186 1.8369186 1.8369186
## 1953.001 1.8394435 1.8394435 1.8394435
## 1954.001 1.8411938 1.8411938 1.8411938
## 1955.001 1.8421747 1.8421747 1.8421747
## 1956.001 1.8423866 1.8423866 1.8423866
## 1957.001 1.8418261 1.8418261 1.8418261
## 1958.001 1.8404985 1.8404985 1.8404985
## 1959.001 1.8384038 1.8384038 1.8384038
## 1960.001 1.8355392 1.8355392 1.8355392
## 1961.001 1.8319096 1.8319096 1.8319096
## 1962.001 1.8275149 1.8275149 1.8275149
## 1963.001 1.8223527 1.8223527 1.8223527
## 1964.001 1.8164277 1.8164277 1.8164277
## 1965.001 1.8097396 1.8097396 1.8097396
## 1966.001 1.8022864 1.8022864 1.8022864
## 1967.001 1.7940725 1.7940725 1.7940725
## 1968.001 1.7850975 1.7850975 1.7850975
## 1969.001 1.7753598 1.7753598 1.7753598
## 1970.001 1.7648637 1.7648637 1.7648637
## 1971.001 1.7536083 1.7536083 1.7536083
## 1972.001 1.7415927 1.7415927 1.7415927
## 1973.001 1.7288208 1.7288208 1.7288208
## 1974.001 1.7152917 1.7152917 1.7152917
## 1975.001 1.7010048 1.7010048 1.7010048
## 1976.001 1.6859637 1.6859637 1.6859637
## 1977.001 1.6701674 1.6701674 1.6701674
## 1978.001 1.6536156 1.6536156 1.6536156
## 1979.001 1.6363119 1.6363119 1.6363119
## 1980.001 1.6182550 1.6182550 1.6182550
## 1981.001 1.5994449 1.5994449 1.5994449
## 1982.001 1.5798852 1.5798852 1.5798852
```

```
## 1983.001 1.5595741 1.5595741 1.5595741
## 1984.001 1.5385124 1.5385124 1.5385124
## 1985.001 1.5167088 1.5167088 1.5167088
## 1986.001 1.4942165 1.4942165 1.4942165
## 1987.001 1.4711166 1.4711166 1.4711166
## 1988.001 1.4474918 1.4474918 1.4474918
## 1989.001 1.4234214 1.4234214 1.4234214
## 1990.001 1.3989869 1.3989869 1.3989869
## 1991.001 1.3742701 1.3742701 1.3742701
## 1992.001 1.3493512 1.3493512 1.3493512
## 1993.001 1.3243116 1.3243116 1.3243116
## 1994.001 1.2992325 1.2992325 1.2992325
## 1995.001 1.2741947 1.2741947 1.2741947
## 1996.001 1.2492796 1.2492796 1.2492796
## 1997.001 1.2245675 1.2245675 1.2245675
## 1998.001 1.2001406 1.2001406 1.2001406
## 1999.001 1.1760795 1.1760795 1.1760795
## 2000.001 1.1524640 1.1524640 1.1524640
## 2001.001 1.1293777 1.1293777 1.1293777
## 2002.001 1.1069000 1.1069000 1.1069000
## 2003.001 1.0851108 1.0851108 1.0851108
## 2004.001 1.0640946 1.0640946 1.0640946
## 2005.001 1.0439300 1.0439300 1.0439300
## 2006.001 1.0246964 1.0246964 1.0246964
## 2007.001 1.0064801 1.0064801 1.0064801
## 2008.001 0.9893581 0.9893581 0.9893581
## 2009.001 0.9734097 0.9734097 0.9734097
## 2010.001 0.9587229 0.9587229 0.9587229
## 2011.001 0.9453730 0.9453730 0.9453730
## 2012.001 0.9334394 0.9334394 0.9334394
## 2013.001 0.9230118 0.9230118 0.9230118
## 2014.001 0.9141635 0.9141635 0.9141635
## 2015.001 0.9069742 0.9069742 0.9069742
## 2016.001 0.9015354 0.9015354 0.9015354
## 2017.001 0.8979183 0.8979183 0.8979183
## 2018.001 0.8962028 0.8962028 0.8962028
```

```
multistep$lower$y
```

```
##           18           19           20
## 1950.001 1.2149792 1.2149792 1.2149792
## 1951.001 1.2515729 1.2515729 1.2515729
## 1952.001 1.2811558 1.2811558 1.2811558
## 1953.001 1.3044992 1.3044992 1.3044992
## 1954.001 1.3224201 1.3224201 1.3224201
## 1955.001 1.3357484 1.3357484 1.3357484
## 1956.001 1.3452682 1.3452682 1.3452682
## 1957.001 1.3516827 1.3516827 1.3516827
## 1958.001 1.3556031 1.3556031 1.3556030
## 1959.001 1.3575227 1.3575227 1.3575227
## 1960.001 1.3578191 1.3578191 1.3578191
## 1961.001 1.3567709 1.3567709 1.3567709
## 1962.001 1.3545563 1.3545563 1.3545563
## 1963.001 1.3512684 1.3512684 1.3512684
```

```

## 1964.001 1.3469393 1.3469393 1.3469393
## 1965.001 1.3415419 1.3415419 1.3415419
## 1966.001 1.3350063 1.3350063 1.3350063
## 1967.001 1.3272421 1.3272421 1.3272421
## 1968.001 1.3181386 1.3181386 1.3181386
## 1969.001 1.3075814 1.3075814 1.3075814
## 1970.001 1.2954700 1.2954700 1.2954700
## 1971.001 1.2817169 1.2817169 1.2817169
## 1972.001 1.2662604 1.2662604 1.2662604
## 1973.001 1.2490768 1.2490768 1.2490768
## 1974.001 1.2301736 1.2301736 1.2301736
## 1975.001 1.2095968 1.2095968 1.2095968
## 1976.001 1.1874342 1.1874342 1.1874342
## 1977.001 1.1638033 1.1638033 1.1638033
## 1978.001 1.1388531 1.1388531 1.1388531
## 1979.001 1.1127614 1.1127614 1.1127614
## 1980.001 1.0857211 1.0857211 1.0857211
## 1981.001 1.0579400 1.0579400 1.0579400
## 1982.001 1.0296374 1.0296374 1.0296374
## 1983.001 1.0010308 1.0010308 1.0010308
## 1984.001 0.9723375 0.9723375 0.9723375
## 1985.001 0.9437651 0.9437651 0.9437651
## 1986.001 0.9154521 0.9154521 0.9154521
## 1987.001 0.8874950 0.8874950 0.8874950
## 1988.001 0.8599752 0.8599752 0.8599752
## 1989.001 0.8329601 0.8329601 0.8329601
## 1990.001 0.8065116 0.8065116 0.8065116
## 1991.001 0.7806875 0.7806875 0.7806875
## 1992.001 0.7555415 0.7555415 0.7555415
## 1993.001 0.7311282 0.7311282 0.7311281
## 1994.001 0.7075024 0.7075024 0.7075024
## 1995.001 0.6847206 0.6847206 0.6847206
## 1996.001 0.6628412 0.6628412 0.6628412
## 1997.001 0.6419240 0.6419240 0.6419240
## 1998.001 0.6220320 0.6220320 0.6220320
## 1999.001 0.6032273 0.6032273 0.6032273
## 2000.001 0.5855706 0.5855706 0.5855706
## 2001.001 0.5691238 0.5691238 0.5691238
## 2002.001 0.5539395 0.5539395 0.5539395
## 2003.001 0.5400621 0.5400621 0.5400621
## 2004.001 0.5275282 0.5275282 0.5275282
## 2005.001 0.5163501 0.5163501 0.5163501
## 2006.001 0.5065167 0.5065167 0.5065167
## 2007.001 0.4979908 0.4979908 0.4979908
## 2008.001 0.4906840 0.4906840 0.4906840
## 2009.001 0.4844569 0.4844569 0.4844569
## 2010.001 0.4791130 0.4791130 0.4791130
## 2011.001 0.4743643 0.4743643 0.4743643
## 2012.001 0.4698370 0.4698370 0.4698370
## 2013.001 0.4650694 0.4650694 0.4650694
## 2014.001 0.4594862 0.4594862 0.4594862
## 2015.001 0.4524351 0.4524351 0.4524351
## 2016.001 0.4432230 0.4432230 0.4432230
## 2017.001 0.4311321 0.4311321 0.4311321

```

2018.001 0.4155084 0.4155084 0.4155084

multistep\$upper\$y

##		18	19	20
##	1950.001	2.444131	2.444131	2.444131
##	1951.001	2.415672	2.415672	2.415672
##	1952.001	2.392681	2.392681	2.392681
##	1953.001	2.374388	2.374388	2.374388
##	1954.001	2.359967	2.359967	2.359967
##	1955.001	2.348601	2.348601	2.348601
##	1956.001	2.339505	2.339505	2.339505
##	1957.001	2.331969	2.331969	2.331969
##	1958.001	2.325394	2.325394	2.325394
##	1959.001	2.319285	2.319285	2.319285
##	1960.001	2.313259	2.313259	2.313259
##	1961.001	2.307048	2.307048	2.307048
##	1962.001	2.300474	2.300474	2.300474
##	1963.001	2.293437	2.293437	2.293437
##	1964.001	2.285916	2.285916	2.285916
##	1965.001	2.277937	2.277937	2.277937
##	1966.001	2.269566	2.269566	2.269566
##	1967.001	2.260903	2.260903	2.260903
##	1968.001	2.252056	2.252056	2.252056
##	1969.001	2.243138	2.243138	2.243138
##	1970.001	2.234257	2.234257	2.234257
##	1971.001	2.225500	2.225500	2.225500
##	1972.001	2.216925	2.216925	2.216925
##	1973.001	2.208565	2.208565	2.208565
##	1974.001	2.200410	2.200410	2.200410
##	1975.001	2.192413	2.192413	2.192413
##	1976.001	2.184493	2.184493	2.184493
##	1977.001	2.176531	2.176531	2.176531
##	1978.001	2.168378	2.168378	2.168378
##	1979.001	2.159862	2.159862	2.159862
##	1980.001	2.150789	2.150789	2.150789
##	1981.001	2.140950	2.140950	2.140950
##	1982.001	2.130133	2.130133	2.130133
##	1983.001	2.118117	2.118117	2.118117
##	1984.001	2.104687	2.104687	2.104687
##	1985.001	2.089653	2.089653	2.089653
##	1986.001	2.072981	2.072981	2.072981
##	1987.001	2.054738	2.054738	2.054738
##	1988.001	2.035008	2.035008	2.035008
##	1989.001	2.013883	2.013883	2.013883
##	1990.001	1.991462	1.991462	1.991462
##	1991.001	1.967853	1.967853	1.967853
##	1992.001	1.943161	1.943161	1.943161
##	1993.001	1.917495	1.917495	1.917495
##	1994.001	1.890963	1.890963	1.890963
##	1995.001	1.863669	1.863669	1.863669
##	1996.001	1.835718	1.835718	1.835718
##	1997.001	1.807211	1.807211	1.807211
##	1998.001	1.778249	1.778249	1.778249

```
## 1999.001 1.748932 1.748932 1.748932
## 2000.001 1.719357 1.719357 1.719357
## 2001.001 1.689632 1.689632 1.689632
## 2002.001 1.659861 1.659861 1.659861
## 2003.001 1.630159 1.630159 1.630159
## 2004.001 1.600661 1.600661 1.600661
## 2005.001 1.571510 1.571510 1.571510
## 2006.001 1.542876 1.542876 1.542876
## 2007.001 1.514969 1.514969 1.514969
## 2008.001 1.488032 1.488032 1.488032
## 2009.001 1.462362 1.462362 1.462362
## 2010.001 1.438333 1.438333 1.438333
## 2011.001 1.416382 1.416382 1.416382
## 2012.001 1.397042 1.397042 1.397042
## 2013.001 1.380954 1.380954 1.380954
## 2014.001 1.368841 1.368841 1.368841
## 2015.001 1.361513 1.361513 1.361513
## 2016.001 1.359848 1.359848 1.359848
## 2017.001 1.364704 1.364704 1.364704
## 2018.001 1.376897 1.376897 1.376897
```

Stationarity test

Lastly, we conducted Monte Carlo tests to assess the stationarity of functional time series for the three datasets. The null hypothesis for each test was that the series is stationary.

Combined country data - p-value is 0.045, thus we reject the null hypothesis, indicating potentially non-stationarity of the data.

European country data - p-value is 0.482, thus we cannot reject the null hypothesis, indicating stationary data.

Asian country data - p-value is 0.122, thus we cannot reject the null hypothesis, indicating stationary data.

```
# Stationarity
T_stationary(tf$y)
```

```
##
## Monte Carlo test of stationarity of a functional time series
##
## null hypothesis: the series is stationary
##
## p-value = 0.057
## N (number of functions) = 37
## number of MC replications = 1000
```

```
T_stationary(euf$y)
```

```
##
## Monte Carlo test of stationarity of a functional time series
##
## null hypothesis: the series is stationary
##
```

```
## p-value = 0.496
## N (number of functions) = 20
## number of MC replications = 1000
```

```
T_stationary(asf$y)
```

```
##
## Monte Carlo test of stationarity of a functional time series
##
## null hypothesis: the series is stationary
##
## p-value = 0.119
## N (number of functions) = 17
## number of MC replications = 1000
```