# Exercise 2.1

## Loading the biofam data set
```r
data(biofam)
```

## Variable names
```r
names(biofam)
```

## Adding an age variable
```r
biofam$age <- 2002 - biofam$birthyr
```

## Distribution
```r
summary(biofam$age)
```

## Distribution of woman age
```r
summary(biofam$age[biofam$sex == "woman"])
```

## Creating the cohort factor
```r
biofam$cohort <- cut(biofam$birthyr, c(1900,1930,1940,1950,1960),
                     right=FALSE)
```

## Frequency table
```r
table(biofam$cohort)
```
### Sequence analysis for social scientists
### Analyzing sequences using dissimilarities
### Matthias Studer, Alexis Gabadinho,
### Gilbert Ritschard, Nicolas S. Müller
### Summer School on Advanced Methods for the
### Analysis of Complex Event History Data,
### Bristol, 28-29 June 2010

#### Exercise 3.1
#### From exercise 2.1
#### Loading the data set and creating the cohort factor
data(biofam)
biofam$cohort <- cut(biofam$birthyr, c(1900,1930,1940,1950,1960),
                   right=FALSE)

#### Getting help
help(biofam)

#### Variables a15 to a30 are in column 10 to 25
names(biofam)

#### Vectors containing state names and long labels
bf.states <- c("Parent", "Left", "Married", "Left/Married", "Child",
                "Left/Child", "Left/Married/Child", "Divorced")

#### Creating the state sequence object
biofam.seq <- seqdef(biofam[,10:25], states=bf.shortlab, labels=bf.states)

#### Printing in STS format: we can use the head() function
head(biofam.seq)

#### Printing in SPS format requires to use explicitly the print() method
#### since we have to pass the format="SPS" argument
print(biofam.seq[1:6,], format="SPS")
## Exercise 3.2

## Sequence index plot
seqIplot(biofam.seq, group=biofam$sex, sortv=biofam$cohort)

## Sequence frequency plot
seqfplot(biofam.seq, group=biofam$cohort)

### Exercise 4.1

### 1. Using biofam data set

---

### Loading TraMineR and the biofam data set
library(TraMineR)
data(biofam)

### Create a cohort factor for later use
biofam$cohort <- cut(biofam$birthyr, c(1900,1930,1940,1950,1960),
                     right=FALSE)
print(summary(biofam$cohort))

### Create the sequence object
bfstates <- c("Parent", "Left", "Married", "Left/Married", "Child",
              "Left/Child", "Left/Married/Child", "Divorced")
bf.shortlab <- c("P","L","M","LM","C","LC", "LMC", "D")
bf.seq <- seqdef(biofam[,10:25], states=bf.shortlab, labels=bfstates)

### Compute the OM distance matrix with substitution costs set according to transition rates
bf.dist <- seqdist(bf.seq, method="OM", indel=1, sm="TRATE")

### 3. Cluster the sequences in 3 groups (using either Ward or PAM).
library(cluster)

### A. Clustering using the "ward" criterion
bf.clusterward <- agnes(bf.dist, diss = T, method="ward")

## Dendrogram
plot(bf.clusterward, ask = F, which.plots = 2)
## Extracting cluster membership

bf.cl3 <- cutree(bf.clusterward, k=3)

### B. Clustering using PAM

bf.pam3 <- pam(bf.dist, k=3, diss=T)

## Plot of the quality of the clustering procedure

plot(bf.pam3)

## Cluster membership is in bf.pam3$clustering

bf.pam3$clustering[1:10]

## Frequency table between Ward and PAM

print(table(bf.cl3, bf.pam3$clustering))

### 4. Explore the clustering solution graphically
### using representative sequences.

seqrplot(bf.seq, dist.matrix=bf.dist, group=bf.cl3, 
etrep=.6, tsim=0.1)

### 5. Name and interpret your clusters.

bf.cl3.factor <- factor(bf.cl3, levels=1:3, 
                       labels=c("Own Household", "Alone", "Parent Household"))

### 6. Fit a logistic regression model for one of
### your cluster using the cohort,
### language (plingu02) and sex covariates.

# Creating a dummy variable

own.household <- bf.cl3==1
alone <- bf.cl3==2
parent.household <- bf.cl3==3

# Fit the model using glm

own.household.reglog <- glm(alone ~ sex + cohort + plingu02, 
                            family=binomial(link=logit), data=biofam)
alone.reglog <- glm(alone ~ sex + cohort + plingu02, family=binomial(link=logit), 
data=biofam)
parent.household.reglog <- glm(alone ~ sex + cohort + plingu02, 
                                family=binomial(link=logit), data=biofam)
## Printing the output of the logistic regression

```r
summary(own.household.reglog)
summary(alone.reglog)
summary(parent.household.reglog)
```

### Exercise 4.2
### Sequence discrepancy analysis

1. Using `bf.dist` as distance matrix

```r
## discrepancy of the whole set of sequence
dissvar(bf.dist)
```

2. Compute the association with the cohort covariate using `dissassoc`.

```r
da <- dissassoc(bf.dist, group=biofam$cohort, R=5000)
print(da)
```

3. Plot the empirical null distribution of $F$

```r
hist(da, col="cyan")
```

4. Interpret the differences graphically using `seqdiff` with all sequences sorted according to the first dimension of an MDS.

```r
## Compute first dimension of an MDS
mds <- cmdscale(bf.dist, k=1)

## Plot the sequences
seq1plot(bf.seq, sortv=mds, group=biofam$cohort)
```

5. Explore the evolution of the association using `seqdiff`.

```r
bf.diff <- seqdiff(bf.seq, group=biofam$cohort)
```

```r
## plot the evolution of the pseudo R2
plot(bf.diff, lwd = 3, col="darkred")
```

## Plotting the evolution of discrepancy

```r
plot(bf.diff, lwd = 3, stat="Variance", legendposition="bottomright")
```

6. Fit a regression tree and plot the results.

```r
```
## Build the tree

dt <- disttree(bf.dist ~ sex + birthyr + plingu02, data=biofam, R = 5000)

print(dt)

## Creating GraphViz file

deqtree2dot(dt, "fg_bfseqtree", seqdata=bf.seq, type="d",
            border=NA, withlegend=FALSE, axes=FALSE, ylab="", yaxis=FALSE)

## Running Graphviz

shell("dot -Tsvg -o fg_bfseqtree.svg fg_bfseqtree.dot")

## Running ImageMagick to convert the output to jpg

shell("convert fg_bfseqtree.svg fg_bfseqtree.jpg")

## Viewing the tree

shell("start fg_bfseqtree.jpg")