COS-D407. *Scientific Modeling and Model Validation*

Hands-on excercises

Week 7

University of Helsinki, Finland

01.11.2021–15.12.2021

Lecturer: Christina Bohk-Ewald

Source: https://github.com/christina-bohk-ewald/2021-COS-D407-scientific-modeling-and-model-validation

Table of content:

1. Some preparations in R

1.1 Open a new script for week 7 in R (e.g., *week-7.R*) and save it to a folder of your choice (e.g., *course-COS-D407*).

1.2 Create a filepath to this folder from where you would like to load data and to where you would like to save your outcome. For example,

```
the_course_COS_D407_path <- c("C:/course-COS-D407")
```

1.3 You can then set the working directory to this path

```
setwd(the_course_COS_D407_path)
```

2. Load raw data of IFR by age of Levin et al. (2020)

Levin and colleagues (2020) have introduced an exponential model for predicting IFR, in %, by age. As they provide the raw data, you can test how suitable this exponential model (and other models) are to predict IFR by age.

Please go to the preprint of Levin et al. (2020) on medRxiv and download the Supplementary Data Spreadsheets. You can save this file in your course folder and load the respective data in R:

```
require(openxlsx)

## Representative sample:

levin_ifr_rs <- read.xlsx("LevinEtAl2020-spreadsheet.xlsx",sheet = 2,startRow = 1)
levin_ifr_rs[1:2,]
```

```
##     Study AgeGroup Median_Age Population InfectionRate infrate_ci95_low
## 1 Atlanta     0-17        8.5     402362           0.0              0.0
## 2 Atlanta    18-49       33.5     867468           3.3              1.6
##   infrate_ci95_high Deaths        IFR ifr_ci95_low ifr_ci95_high
## 1               1.0      1 0.00000000   0.00000000            NA
## 2               6.4     20 0.06986547   0.03602438     0.1440975
```

```
## Convenience sample:

levin_ifr_cs <- read.xlsx("LevinEtAl2020-spreadsheet.xlsx",sheet = 3,startRow = 1)
levin_ifr_cs[1:2,]
```

```
##     Study AgeGroup Median_Age Population InfectionRate infrate_ci95_low
## 1 Belgium     0-24         12    3228894      6.686021         4.666667
## 2 Belgium    25-44         35    2956684      6.562703         4.666667
##   infrate_ci95_high Deaths    IFR ifr_ci95_low ifr_ci95_high
## 1          9.555556      1 0.0005       0.0004        0.0007
## 2          9.222222     30 0.0170       0.0120        0.0240
```

```
## Comprehensive tracing:

levin_ifr_ct <- read.xlsx("LevinEtAl2020-spreadsheet.xlsx",sheet = 4,startRow = 1)
levin_ifr_ct[1:2,]
```

```
##       Study AgeGroup Median_Age Population Infections Deaths InfectionRate
## 1 Australia     0-39   20.55027   13533252       2800      0    0.06206934
## 2 Australia    40-59   50.00000    6413795       2080      3    0.06486020
##   infrate_ci95_low infrate_ci95_high        IFR ifr_ci95_low ifr_ci95_high
## 1       0.03103467        0.09310401 0.00000000   0.00000000            NA
## 2       0.03891612        0.09729029 0.07211538   0.04807692     0.1201923
```

```
## And putting all raw data together:

levin_ifr <- c(levin_ifr_rs[,"IFR"],levin_ifr_cs[,"IFR"],levin_ifr_ct[,"IFR"])
levin_age <- c(levin_ifr_rs[,"Median_Age"],
            levin_ifr_cs[,"Median_Age"],
            levin_ifr_ct[,"Median_Age"])
```

Brief data description. The data objects *levin_ifr_rs*, *levin_ifr_cs*, and *levin_ifr_ct* contain IFR estimates (in %) in column 7 by median age in column 2. And the data objects *levin_ifr* and *levin_age* contain all raw data (i.e., 134 data points).

3. Find suitable model for predicting IFR by age based on training data and testing data

Remember that it might not be so much about finding the model that fits best to *all* raw data, but rather about finding the model that predicts best values of the IFR by age the model has not seen yet (machine learning; generalization of underlying pattern). That is why we split *all* raw data into training data and testing data in order to find the model that predicts best IFR by age. Following this approach, we use the training data to fit the models and we use the testing data to test how well the models predict IFR by age in a *new* environment.

3.1 Validation set approach

Please see the hands-on exercise for week 6.

3.2 k-fold cross validation

According to the k-fold cross validation, you randomly split *all* raw data into k parts of roughly equal size.

Repeating the procedure k times, you use each of these k parts once as testing data and the rest of the data as training data. You then fit the six models to the training data before you compare their IFR predictions with the actual IFR values of the testing data using the MSE.

Below you apply a 10-fold cross validation:

```
train_MSE <- matrix(NA,nr=6,nc=10)
test_MSE <- matrix(NA,nr=6,nc=10)

for(run in 1:10){

    ## Random order for using raw data as testing data:

    set.seed(1)
    test_order <- sample(length(levin_ifr),length(levin_ifr))
    unique(test_order)

    ## Determine 10 folds (of equal size) for cross-validation:

    cv_start <- seq(1,length(levin_ifr),round(length(levin_ifr)/10,0))
    cv_end <- seq(round(length(levin_ifr)/10,0),length(levin_ifr),round(length(levin_ifr)/10,0))
    test <- test_order[cv_start[run]:cv_end[run]]
    train <- c(1:length(levin_ifr))[-test]

    ## sort(train)
    ## sort(test)

    train_levin_ifr <- levin_ifr[train]
    train_levin_age <- levin_age[train]
    test_levin_ifr <- levin_ifr[test]
    test_levin_age <- levin_age[test]

    ## Fit models to training data

    model_1 <- lm(train_levin_ifr~train_levin_age)
    model_2 <- lm(train_levin_ifr~I(train_levin_age^2))
    model_3 <- lm(train_levin_ifr~I(train_levin_age^3))
    model_4 <- lm(train_levin_ifr~I(train_levin_age^4))
    model_5 <- lm(log(train_levin_ifr,base=exp(1))~train_levin_age,
subset=train_levin_ifr>0)

    model_6 <- lm(log(train_levin_ifr,base=exp(1))~I(train_levin_age^2),
subset=train_levin_ifr>0)

    ## Calculate and save train MSE

    train_MSE[1,run] <- mean((train_levin_ifr -
predict(model_1,data.frame(train_levin_age=train_levin_age)))^2)

    train_MSE[2,run] <- mean((train_levin_ifr -
predict(model_2,data.frame(train_levin_age=train_levin_age)))^2)

    train_MSE[3,run] <- mean((train_levin_ifr -
predict(model_3,data.frame(train_levin_age=train_levin_age)))^2)

    train_MSE[4,run] <- mean((train_levin_ifr -
predict(model_4,data.frame(train_levin_age=train_levin_age)))^2)

    train_MSE[5,run] <- mean((train_levin_ifr -
exp(predict(model_5,data.frame(train_levin_age=train_levin_age))))^2)

    train_MSE[6,run] <- mean((train_levin_ifr -
exp(predict(model_6,data.frame(train_levin_age=train_levin_age))))^2)

    ## Apply fitted models to testing data and calculate and save test MSE

    test_MSE[1,run] <- mean((test_levin_ifr -
predict(model_1,data.frame(train_levin_age=test_levin_age)))^2)

    test_MSE[2,run] <- mean((test_levin_ifr -
predict(model_2,data.frame(train_levin_age=test_levin_age)))^2)

    test_MSE[3,run] <- mean((test_levin_ifr -
predict(model_3,data.frame(train_levin_age=test_levin_age)))^2)
```

```r
    test_MSE[4,run] <- mean((test_levin_ifr -
predict(model_4,data.frame(train_levin_age=test_levin_age)))^2)

    test_MSE[5,run] <- mean((test_levin_ifr -
exp(predict(model_5,data.frame(train_levin_age=test_levin_age))))^2)

    test_MSE[6,run] <- mean((test_levin_ifr -
exp(predict(model_6,data.frame(train_levin_age=test_levin_age))))^2)

} ## for run
```

The R-code above applies the 10-fold cross validation (run 1 through ten in the for loop). The data objects *train_MSE* and *test_MSE* contain the MSE for each of the six models (rows) in each of the 10 runs (columns) when applied to the training data and the testing data, respectively.

These data allow you to eventually compute the average train MSE and the average test MSE across all ten runs for each of the six models:

```r
rowMeans(train_MSE)
```

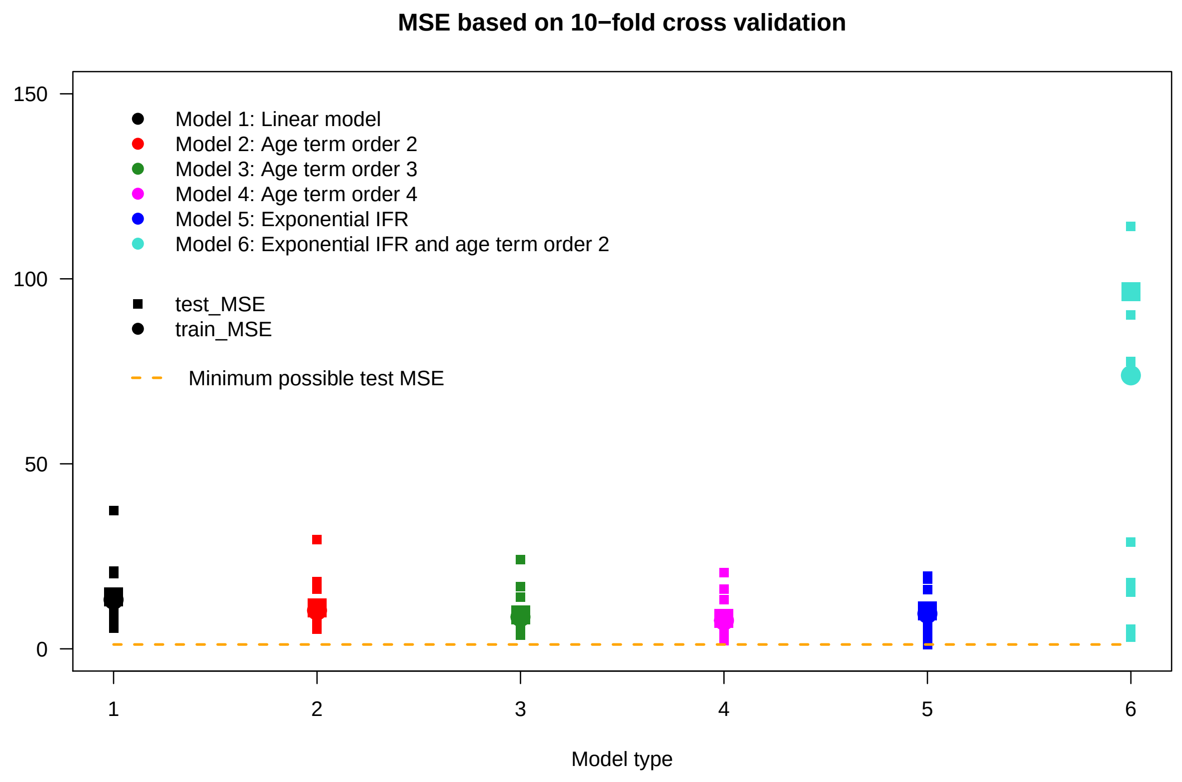
```
## [1] 13.285464 10.382714  8.615378  7.663666  9.514885 73.904353
```

```r
rowMeans(test_MSE)
```

```
## [1] 14.007346 11.061140  9.238990  8.273937 10.271509 96.486483
```

Which model has the smallest / largest testing MSE? Which model has the smallest / largest training MSE? You can also plot your findings:

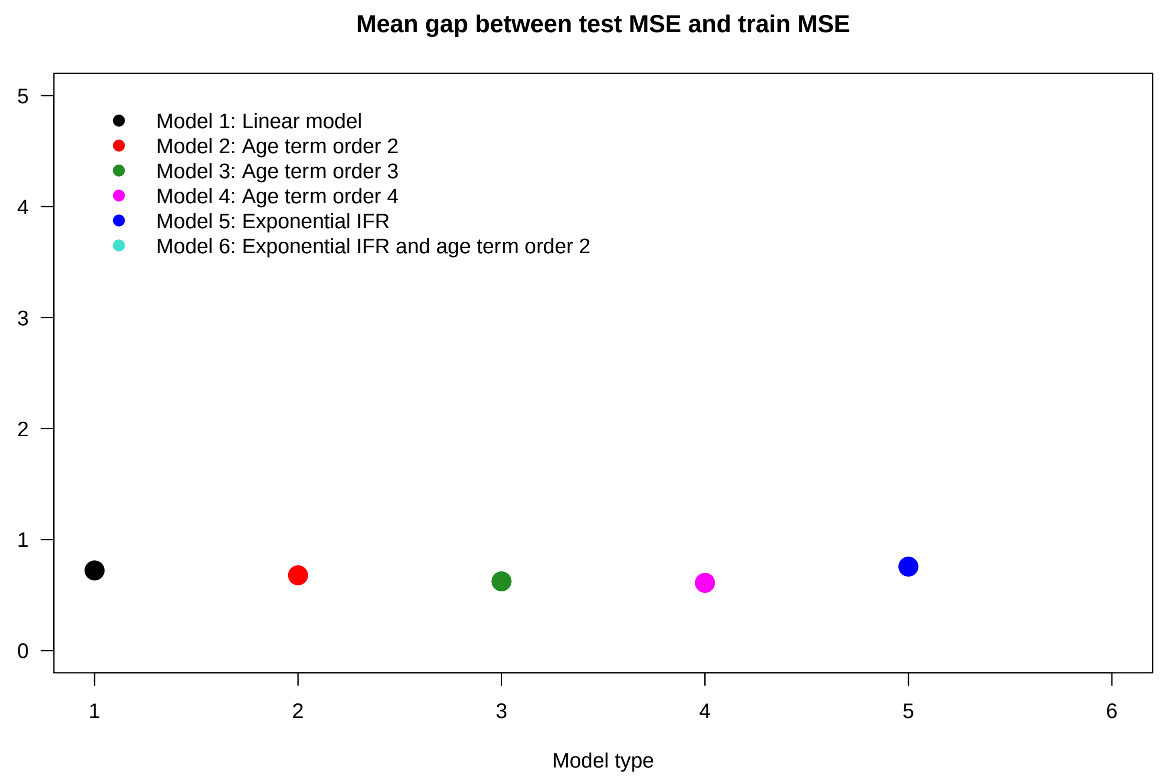
```r
par(fig = c(0,1,0,1), las=1, mai=c(1.4,0.8,0.8,0.4))

plot(1:6,rowMeans(train_MSE),main="MSE based on 10-fold cross validation",
col=c("black","red","forestgreen","magenta","blue","turquoise"),pch=19,cex=2,
xlab="Model type",ylab="",ylim=c(0,150))

points(1:6,rowMeans(test_MSE),
col=c("black","red","forestgreen","magenta","blue","turquoise"),pch=15,cex=2)

for(model in 1:6){
    points(x=rep(model,10),y=test_MSE[model,],
col=c("black","red","forestgreen","magenta","blue","turquoise")[model],pch=15,cex=1)
}

lines(1:6,rep(min(test_MSE),6),col="orange",lty=2,lwd=2)

legend(1,150, c("Model 1: Linear model","Model 2: Age term order 2",
"Model 3: Age term order 3","Model 4: Age term order 4",
"Model 5: Exponential IFR","Model 6: Exponential IFR and age term order 2"),
col=c("black","red","forestgreen","magenta","blue","turquoise"),
pch=rep(19,6),lty=NA,lwd=2,bty="n")

legend(1,100,col=c("black","black"),c("test_MSE","train_MSE"),
lty=NA,pch=c(15,19),lwd=2,bty="n")

legend(1,80,col=c("orange"),c("Minimum possible test MSE"),lty=c(2),lwd=2,bty="n")
```



Something to think about. What do these information tell you about the suitability of each of these models for predicting the IFR by age? What model(s) have comparably low bias and low variance?

To answer this question it might be convenient to zoom into the plot above and it might also be interesting to visualize the gap between average test MSE and average train MSE:

```r
gap <- test_MSE-train_MSE

par(fig = c(0,1,0,1), las=1, mai=c(1.4,0.8,0.8,0.4))

plot(1:6,rowMeans(gap),col=c("black","red","forestgreen","magenta","blue","turquoise"),
pch=19,cex=2,xlab="Model type",ylab="",main="Mean gap between test MSE and train MSE",
ylim=c(0,5))

lines(1:6,rep(min(gap),6),col="orange",lty=2,lwd=2)

legend(1,5, c("Model 1: Linear model","Model 2: Age term order 2",
"Model 3: Age term order 3","Model 4: Age term order 4",
"Model 5: Exponential IFR","Model 6: Exponential IFR and age term order 2"),
col=c("black","red","forestgreen","magenta","blue","turquoise"),
pch=rep(19,6),lty=NA,lwd=2,bty="n")
```



Given these new information, what do you think: what is the most suitable model (of all six models) in order to predict IFR by age?

Something more to think about. Do your findings and conclusions based on the 10-fold cross validation differ from your findings based on the validation set approach (week 6)? What could limitate your findings based on the 10-fold cross validation? Would your findings change if you had done, e.g., a 15-fold cross validation?

3. Time for you to think both creatively and critically about selecting the most suitable method for predicting IFR by age based on raw data provided by Levin and colleagues (2020).

You can consider here the pros and cons of the different procedures adopted to fit and evaluate the models with respect to their ability to accurately predcit IFR by age.

For example, what is the actual purpose of a model that has been designed to predict IFR by age?

What does it mean when a model is fitted to training data instead of to *all* raw data? And what does it mean when a model's predictions are evaluated using testing data instead of *all* raw data?

When selecting the most suitable model (among those six models) for predicting the IFR by age, you may also want to consider the theoretical meaning of the models; e.g., to distinguish between M4 and M5? What other factors may be worthwhile to consider for making this decision?

Which one of the six models would you select to finally predict IFR by age? Why?