Studiengang WI21a / WI21b / WI20a Fach: FUM
HS 2022

Practice Sheet 9: Lösung

1 Classical Risk Analysis of a Model Bank

The task is to perform the (classical) risk analysis lof a model bank as presented in slide series 11. The contracts of the bank's balance sheet are contained in the file `BankBilanzPositionen.csv`.

Remark: Please don't change this file, you risk to introduce formatting errors in the date formats.

In addition, a file named `ZinsSzenarien.csv` with interest rate scenarios is provided.

A third file, named `StaticBankMC_Results.RData`, contains the results of the MC simulation because this simulation is rather long (30min).

The analysis date should be set to `2016-01-02`.

Tasks:

1. Build a balance sheet structure according to the one in the slides, read in the contracts and attach the contracts to the leaf accounts.

2. Create the risk factor environment consisting of a spot rate curve defined by the following rates:

Tenor	3M	1Y	2Y	5Y	7Y	10Y
Rate [%]	-0.28	-0.26	-0.21	0.03	0.20	0.42

   Create a discounting engine that uses this yield curve.

3. Simulate the bank with this environment, define yearly time buckets covering the years from 2016 till 2020, and compute nominal value, market value, liquidity and income.

4. Write a function for carrying out the Monte Carlo simulation with the provided interest rate scenarios.

5. Use this function to carry out a few steps of the simulation with the provided interest rate scenarios.

6. Read-in the file with the results of the full simulation and plot historgrams for the interest rates for the different tenors. Look also at the summary statistics.

7. Extract the distribution of equity from the simulation results (for both, nominal value and market value) and compute VaR95 and ES95.
   Use boxplots to display the time evolution of this distribution over the years.
   Can you explain the different temporal behavior of nominal value and market value?

8. Carry out a similar study for liquidity (both, marginal and cumulative).
9. Carry out a similar study for income.

Solution: Classical Risk Analysis of a Model Bank

1. Balance sheet structure & contraccts:

```
## ----echo=FALSE,results='hide',message=FALSE------------------------------------------------------------------
library(FEMS)
options(warn=-1)
options(width=60)  #  width of console
```

```
## -------------------------------------------------------------------------------------------------------------
# Analysis date:
t0 <- "2016-01-02"
# Create portfolio and import cntracts
ptf <- Portfolio()
ptf.tbl <- read.csv("./R/BankBilanzPositionen.csv",
                    header = TRUE)
import(ptf,source = ptf.tbl)
```

```
R> [1] "20 CTs imported from data.frame based on /var/folders/7_/43hp0rmx5pg00bj9s4574tjw0000gn/T//RtmpoVxHMs/21eb44dbe12.csv
```

```
# Extract attributes for further usage
pars <- ptf.tbl[, c( "BilanzKonto", "ContractID")]
```

```
## -------------------------------------------------------------------------------------------------------------
Bank <- institution("BankA")
temp.node <- FindNode(Bank,"LongTerm")
temp.node$name <- "FixeDarlehen"
Bank$Assets$AddChild("VariableDarlehen")
Bank$Liabilities$AddChild("Interbank")
Bank$Liabilities$AddChild("Kundenkonten")
```

```
## -------------------------------------------------------------------------------------------------------------
fd.id <- subset(pars, BilanzKonto == "FixeDarlehen")$ContractID
addContracts(get(ptf, fd.id), FindNode(Bank$Assets, "FixeDarlehen"))
vd.id <- subset(pars, BilanzKonto == "VariableDarlehen")$ContractID
addContracts(get(ptf, vd.id), FindNode(Bank$Assets, "VariableDarlehen"))
ib.id <- subset(pars, BilanzKonto == "Interbank")$ContractID
addContracts(get(ptf, ib.id), FindNode(Bank$Liabilities, "Interbank"))
kk.id <- subset(pars, BilanzKonto == "Kundenkonti")$ContractID
addContracts(get(ptf, kk.id), FindNode(Bank$Liabilities, "Kundenkonten"))
```

2. Risk factor environment consisting & discounting engine:

```
## -------------------------------------------------------------------------------------------------------------
# Extract label for yield cuve
obj <- unique(ptf.tbl$MarketObjectCodeOfRateReset)
obj <- obj[which(obj != "NULL")] # "NULL" stands for "undefined"
obj
```

```
R> [1] "YC_EA_AAA"
```

```r
# Yield curve
tenors <- c("3M", "1Y", "2Y", "5Y", "7Y", "10Y")
rates <- c(-0.28, -0.26, -0.21, 0.03, 0.20, 0.42)/100
yc <- YieldCurve(label = obj, ReferenceDate = t0,
                 Tenors = tenors, Rates = rates)
# Market environment
rf <- RFConn(yc)
set(ptf, rf) # Connect to portfolio
# Define discounting method
eng <- DcEngine(dc.spread = 0.0, dc.object = yc)
set(eng, rf)
```

3. Simulation and analysis of base scenario:

```r
## ----------------------------------------------------------------------------------------------------------------
by <- timeSequence(t0, by = "1 year", length.out=6)
years <- as.character(2016:2020)
tb <- timeBuckets (by, bucketLabs = years,
        breakLabs=c("16-01-02", "17-01-02", "18-01-02", "19-01-02", "20-01-02",
                    "21-01-02"))
events(Bank, t0, rf, end_date = "2025-12-31")
```

```r
## ----------------------------------------------------------------------------------------------------------------
options(width=100)   #  width of console
value(Bank, tb, type = "nominal")
```

```
R>                              16-01-02 17-01-02 18-01-02 19-01-02 20-01-02 21-01-02
R> 1  BankA                            0        0        0        0        0        0
R> 2   ¦--Assets                   57000    57625    58048    58350    38547    37498
R> 3   ¦   ¦--Current              -2200     8826    12555    15211    22836    23764
R> 4   ¦   ¦--ShortTerm                0        0        0        0        0        0
R> 5   ¦   ¦--FixeDarlehen         29200    19499    16894    15239     8512     7234
R> 6   ¦   °--VariableDarlehen     30000    29300    28600    27900     7200     6500
R> 7   °--Liabilities                  0    -1975    -3579    -5073    -6479    -6654
R> 8       ¦--Debt                     0        0        0        0        0        0
R> 9       ¦--Equity                   0    -1975    -3579    -5073    -6479    -6654
R> 10      ¦--Interbank           -50000   -50000   -50000   -50000   -30000   -30000
R> 11      °--Kundenkonten         -7000    -5650    -4469    -3276    -2069     -844
```

```r
## ----------------------------------------------------------------------------------------------------------------
value(Bank, tb, type = "market", method=eng)
```

```
R>                              16-01-02 17-01-02 18-01-02 19-01-02 20-01-02 21-01-02
R> 1  BankA                            0        0        0        0        0        0
R> 2   ¦--Assets                   -1374    60312    54652    48959    45695    16425
R> 3   ¦   ¦--Current              -1374      823      827      843      840      813
R> 4   ¦   ¦--ShortTerm                0        0        0        0        0        0
R> 5   ¦   ¦--FixeDarlehen             0    22968    20363    16918    14749     7750
R> 6   ¦   °--VariableDarlehen         0    36521    33462    31198    30105     7862
R> 7   ¦--Liabilities               1374      883     4468     8107     9428    16757
R> 8   ¦   ¦--Debt                     0        0        0        0        0        0
R> 9   ¦   ¦--Equity                1374      883     4468     8107     9428    16757
R> 10  ¦   ¦--Interbank                0   -55221   -54441   -53669   -52998   -32323
R> 11  ¦   °--Kundenkonten             0    -5974    -4679    -3398    -2124     -859
R> 12  °--Operations                   0        0        0        0        0        0
R> 13      ¦--Revenues                 0        0        0        0        0        0
R> 14      °--Expenses                 0        0        0        0        0        0
```

```r
## ----------------------------------------------------------------------------------------------------------------
liquidity(Bank, by = tb, type = "marginal", digits = 0)
```

```
R>                             2016  2017  2018   2019  2020
R> 1  BankA                   11031  3744  2652   7595   847
R> 2   ¦--Assets              12978  5721  4724  29748  2810
R> 3   ¦   ¦--Current             0     0     0      0     0
R> 4   ¦   ¦--ShortTerm           0     0     0      0     0
R> 5   ¦   ¦--FixeDarlehen    10672  3470  2450   7451  1679
R> 6   ¦   °--VariableDarlehen  2305  2251  2274  22297  1131
R> 7   ¦--Liabilities         -1946 -1977 -2072 -22153 -1962
R> 8   ¦   ¦--Debt                0     0     0      0     0
R> 9   ¦   ¦--Equity              0     0     0      0     0
R> 10  ¦   ¦--Interbank        -507  -691  -789 -20873  -690
R> 11  ¦   °--Kundenkonten    -1439 -1285 -1283  -1280 -1273
R> 12  °--Operations              0     0     0      0     0
R> 13      ¦--Revenues            0     0     0      0     0
R> 14      °--Expenses            0     0     0      0     0
```

```
## ----------------------------------------------------------------------------------------------------------------
income(Bank, by = tb, type = "marginal", revaluation.gains = FALSE,
       digits = 0)
```

```
R>                            2016 2017 2018 2019 2020
R> 1  BankA                   1975 1605 1494 1405  175
R> 2   ¦--Assets              2571 2401 2373 2351  913
R> 3   ¦   ¦--Current           -5  -15    4   30   82
R> 4   ¦   ¦--ShortTerm          0    0    0    0    0
R> 5   ¦   ¦--FixeDarlehen     971  865  795  724  401
R> 6   ¦   °--VariableDarlehen 1605 1551 1574 1597  431
R> 7   ¦--Liabilities         -596 -796 -879 -945 -738
R> 8   ¦   ¦--Debt               0    0    0    0    0
R> 9   ¦   ¦--Equity             0    0    0    0    0
R> 10  ¦   ¦--Interbank       -507 -691 -789 -873 -690
R> 11  ¦   °--Kundenkonten     -89 -105  -90  -72  -48
R> 12  °--Operations             0    0    0    0    0
R> 13      ¦--Revenues           0    0    0    0    0
R> 14      °--Expenses           0    0    0    0    0
```

4. Function for MC simulation:

```
mc.simulation <- function(bank, t0, end_date, rf, scenarios, by)
{
  yc <- rf[[get(rf, "keys")[1]]]
  rates.old <- yc[["Rates"]] # Store original rates
  results <- list() # empty results list
  # Start with Scenario 0
  events(bank, t0, rf, end_date = end_date)
  value.nom <- value(bank, by, "nominal", digits=0)
  value.npv <- value(bank, by, type = "market", method=eng, digits=0)
  liq <- liquidity(bank, by = tb, type = "marginal", digits = 0)
  inc <- income(bank, by = tb, type = "marginal", revaluation.gains = FALSE, digits = 0)
  results[[1]] <- list(value.nom = value.nom, value.npv = value.npv,
                       liquidity = liq, income = inc)
  print ("Start MC loop")
  for(i in 1:nrow(scenarios)) { # Loop over yield curve scenarios
    print (paste("MC loop no.",i))
    yc[["Rates"]] <- as.numeric(rates.old + scenarios[i,]) # Interest rate shock
    events(bank, t0, rf, end_date = end_date)
    value.nom <- value(bank, by, "nominal", digits = 0)
    value.npv <- value(bank, by, type = "market", method=eng, digits = 0)
    liq <- liquidity(bank, by = tb, type = "marginal", digits = 0)
    inc <- income(bank, by = tb, type = "marginal", revaluation.gains = FALSE, digits = 0)
    results[[i+1]] <- list(value.nom = value.nom, value.npv = value.npv,
                           liquidity = liq, income = inc)
    yc[["Rates"]] <- rates.old  # restore old rates
  }
  return(results)
}
```

5. Use this function to carry out a few steps of the simulation with the provided interest rate scenarios.

```
## ----------------------------------------------------------------------------------------------------------------
# load yield curve scenarios
mc.scenarios = read.table("./R/ZinsSzenarien.csv", sep=";", header=TRUE)
head(mc.scenarios)
```

```
R>   Description     ID   X3M   X1Y   X2Y   X5Y   X7Y  X10Y
R> 1       Shift shock1 0.001 0.001 0.001 0.001 0.001 0.001
R> 2       Shift shock2 0.002 0.002 0.002 0.002 0.002 0.002
R> 3       Shift shock3 0.003 0.003 0.003 0.003 0.003 0.003
R> 4       Shift shock4 0.004 0.004 0.004 0.004 0.004 0.004
R> 5       Shift shock5 0.005 0.005 0.005 0.005 0.005 0.005
R> 6       Shift shock6 0.006 0.006 0.006 0.006 0.006 0.006
```

```
# Drop columns with meta information
mc.scenarios = mc.scenarios[,-c(1,2)]
# Only use the first n lines for the MC simulation
n <- 2
results <- mc.simulation(Bank, t0, end_date = "2025-12-31",
                           rf, scenarios=mc.scenarios[1:n,], tb)
```

```
R> [1] "Start MC loop"
R> [1] "MC loop no. 1"
R> [1] "MC loop no. 2"
```

6. Read-in the file with the results of the full simulation and plot historgrams for the interest rates for the different tenors. Look also at the summary statistics.

```
load(file="./R/StaticBankMC_Results.RData")
results <- resultate
length(results)
```

```
R> [1] 101
```

```
names(results[[1]])
```

```
R> [1] "value.nom" "value.npv" "liquidity" "income"
```

```
mc.rates <- rbind(rep(0,6), mc.scenarios)
rownames(mc.rates) <- c("0", rownames(mc.scenarios))
scenarios.summary <- rbind(
summary(100*mc.rates[,1]),
summary(100*mc.rates[,2]),
summary(100*mc.rates[,3]),
summary(100*mc.rates[,4]),
summary(100*mc.rates[,5]),
summary(100*mc.rates[,6])
)
rownames(scenarios.summary) <- c("3m", "1y", "2yrs", "5yrs", "7yrs", "10yrs")

scenarios.summary
```

```
R>           Min.   1st Qu.  Median      Mean  3rd Qu.     Max.
R> 3m    -0.05910 -0.001914 0.06000 0.3553533 0.500000 2.549592
R> 1y    -0.11820 -0.010500 0.09768 0.3170491 0.500000 1.745352
R> 2yrs  -0.31520 -0.032352 0.20000 0.4069411 0.688352 2.185152
R> 5yrs  -1.39870 -0.210586 0.40000 0.4380613 1.043984 2.130000
R> 7yrs  -2.08820 -0.336444 0.40000 0.4166368 1.060000 3.180000
R> 10yrs -2.72845 -0.496661 0.30000 0.3240222 0.831000 4.155000
```

7. Extract the distribution of equity from the simulation results (for both, nominal value and market value) and compute VaR95 and ES95.
   Use boxplots to display the time evolution of this distribution over the years.

```
Bank

R>                    levelName
R> 1  BankA
R> 2   ¦--Assets
R> 3   ¦   ¦--Current
R> 4   ¦   ¦--ShortTerm
R> 5   ¦   ¦--FixeDarlehen
R> 6   ¦   °--VariableDarlehen
R> 7   ¦--Liabilities
R> 8   ¦   ¦--Debt
R> 9   ¦   ¦--Equity
R> 10  ¦   ¦--Interbank
R> 11  ¦   °--Kundenkonten
R> 12  °--Operations
R> 13      ¦--Revenues
R> 14      °--Expenses

equity.nom <- -sapply(resultate, function(x) as.numeric(x$value.nom[9,]))
rownames(equity.nom) <- colnames(resultate[[1]]$value.nom)
dim(equity.nom)

R> [1]   6 101

equity.nom[,1:10]

R>          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
R> 16-01-02    0    0    0    0    0    0    0    0    0     0
R> 17-01-02 1975 1977 1979 1981 1983 1985 1987 1989 1991  1993
R> 18-01-02 3579 3590 3600 3610 3620 3630 3640 3651 3661  3671
R> 19-01-02 5073 5091 5108 5125 5143 5160 5178 5195 5213  5230
R> 20-01-02 6479 6507 6534 6562 6590 6618 6646 6674 6702  6731
R> 21-01-02 6654 6680 6707 6733 6760 6786 6813 6840 6867  6894

## ---------------------------------------------------------------------------------------------------------------
equity.npv <- -sapply(results, function(x) as.numeric(x$value.npv[9,]))
rownames(equity.npv) <- rownames(equity.nom)
equity.npv[,1:10]

R>            [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
R> 16-01-02  -2200  -2200  -2200  -2200  -2200  -2200  -2200  -2200  -2200  -2200
R> 17-01-02  -5711  -5701  -5691  -5680  -5670  -5659  -5648  -5637  -5626  -5615
R> 18-01-02  -9294  -9292  -9290  -9287  -9285  -9282  -9279  -9276  -9273  -9269
R> 19-01-02 -12949 -12953 -12957 -12960 -12963 -12966 -12969 -12972 -12974 -12977
R> 20-01-02 -14276 -14268 -14260 -14251 -14243 -14234 -14225 -14216 -14207 -14198
R> 21-01-02 -21592 -21599 -21607 -21614 -21621 -21628 -21635 -21642 -21649 -21656

## ----eval=FALSE-----------------------------------------------------------------------------------------------------
## dd <- "17-01-02"
## par(mfrow=c(1,2))
## hist(as.numeric(equity.nom[dd,]), main="Nominal", xlab="Value")
## hist(as.numeric(equity.npv[dd,]), main="Market Oriented", xlab="Value")


## ---------------------------------------------------------------------------------------------------------------
alpha <- 0.95
equity.nom.mean <- rowMeans(equity.nom)
equity.nom.var <- apply(equity.nom, 1, quantile, probs=1-alpha)
equity.nom.es <- equity.nom.var
for (i in 1:6) {
  equity.nom.es[i] <- mean(equity.nom[i, equity.nom[i,]<=equity.nom.var[i]])
}
cbind(equity.nom.mean, equity.nom.var,equity.nom.es)
```

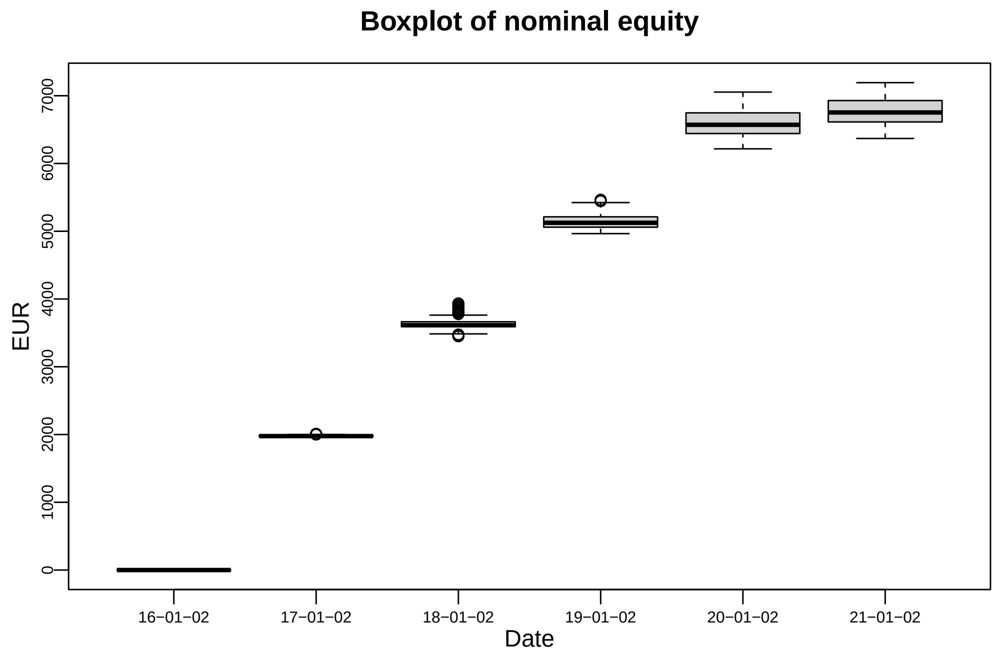
```
R>          equity.nom.mean equity.nom.var equity.nom.es
R> 16-01-02           0.000              0         0.000
R> 17-01-02        1981.386           1974      1973.692
R> 18-01-02        3633.050           3493      3471.667
R> 19-01-02        5150.228           5042      5003.667
R> 20-01-02        6602.436           6395      6306.667
R> 21-01-02        6777.683           6560      6466.000
```

```
## -----------------------------------------------------------------------------------------------------------------------
equity.npv.mean <- rowMeans(equity.npv)
equity.npv.var <- apply(equity.npv, 1, quantile, probs=1-alpha)
equity.npv.es <- equity.npv.var
for (i in 1:6) {
  equity.npv.es[i] <- mean(equity.npv[i, equity.npv[i,]<=equity.npv.var[i]])
}
cbind(equity.npv.mean, equity.npv.var,equity.npv.es)
```

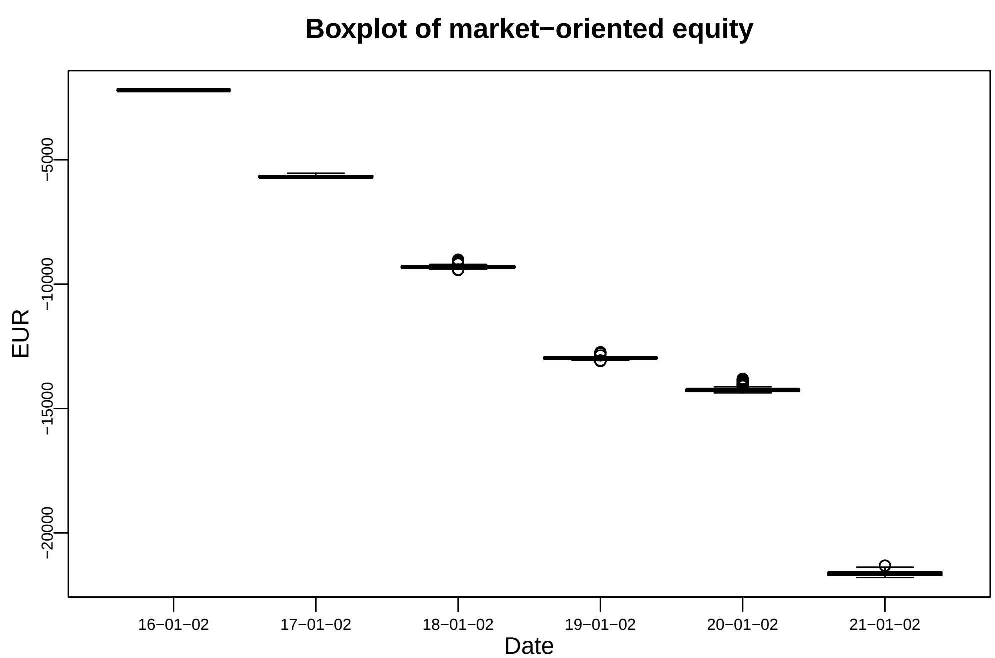
```
R>          equity.npv.mean equity.npv.var equity.npv.es
R> 16-01-02       -2200.000          -2200     -2200.000
R> 17-01-02       -5668.881          -5715     -5715.000
R> 18-01-02       -9298.020          -9397     -9415.167
R> 19-01-02      -12970.030         -13072    -13083.667
R> 20-01-02      -14241.446         -14355    -14363.500
R> 21-01-02      -21636.089         -21775    -21785.167
```

```
## -----------------------------------------------------------------------------------------------------------------------
eq.nom.list = list(
  "16-01-02" = equity.nom[1,],
  "17-01-02" = equity.nom[2,],
  "18-01-02" = equity.nom[3,],
  "19-01-02" = equity.nom[4,],
  "20-01-02" = equity.nom[5,],
  "21-01-02" = equity.nom[6,])

## ----eval=FALSE---------------------------------------------------------------------------------------------------------
## boxplot(eq.nom.list, main="Boxplot of nominal equity",
##         xlab="Date",ylab="EUR")
```

```
## ----echo=FALSE,out.extra = 'height=6.5cm,width=7cm'-----------------------------------------------------------------
par(mfrow=c(1,1))
boxplot(eq.nom.list, main="Boxplot of nominal equity",
        xlab="Date",ylab="EUR")
```

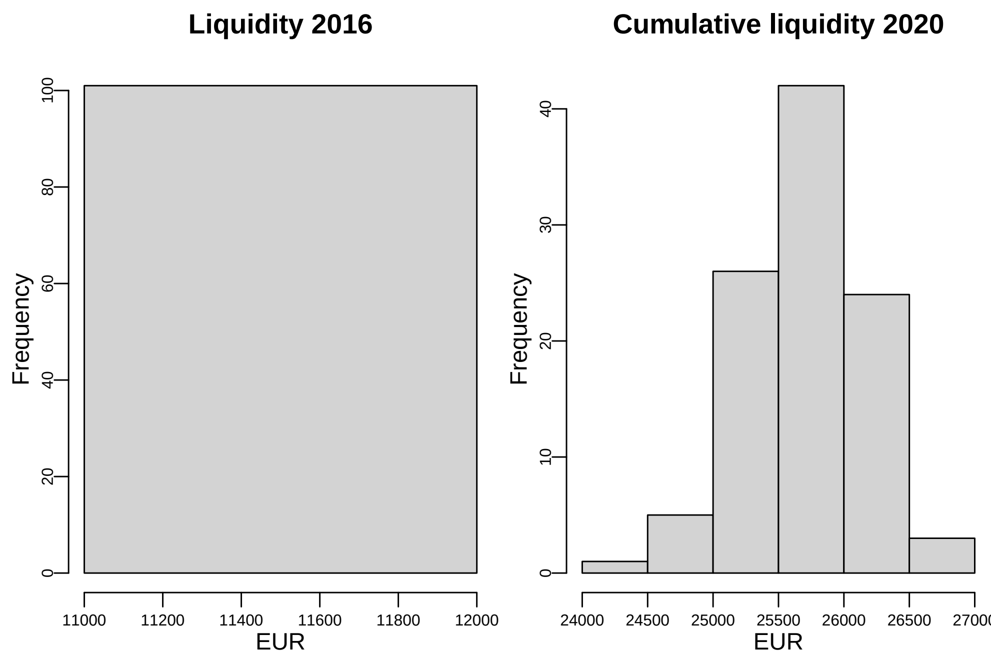



```
## ----echo=FALSE,out.extra = 'height=6.5cm,width=7cm'--------------------------------------------------------------------
eq.npv.list = list(
  "16-01-02" = equity.npv[1,],
  "17-01-02" = equity.npv[2,],
  "18-01-02" = equity.npv[3,],
  "19-01-02" = equity.npv[4,],
  "20-01-02" = equity.npv[5,],
  "21-01-02" = equity.npv[6,])
# boxplot(liq.list, names = as.character(2016:2024), log="y")
boxplot(eq.npv.list, main="Boxplot of market-oriented equity",
        xlab="Date",ylab="EUR")
```

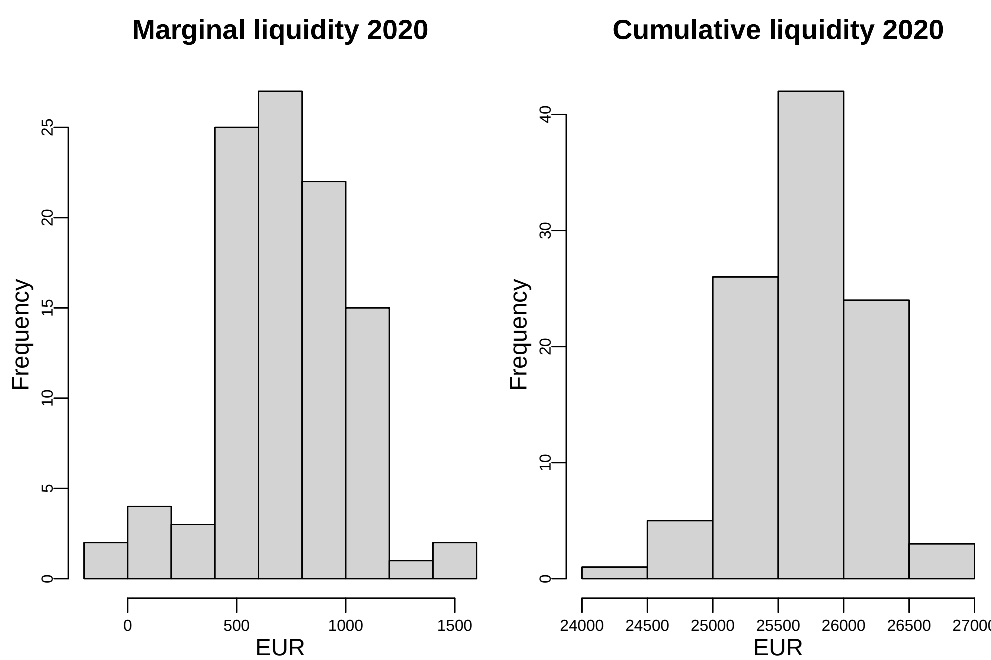


8. Carry out a similar study for liquidity (both, marginal and cumulative).

```
liq <- sapply(results, function(x) as.numeric(x$liquidity[1,]))
rownames(liq) <- 2016:2020

par(mfrow=c(1,2))
hist(liq["2016",],
  main="Liquidity 2016", xlab="EUR")
hist(colSums(liq),
  main="Cumulative liquidity 2020", xlab="EUR")
```




```
par(mfrow=c(1,2))
hist(liq["2020",], main="Marginal liquidity 2020", xlab="EUR")
hist(colSums(liq), main="Cumulative liquidity 2020", xlab="EUR")
```



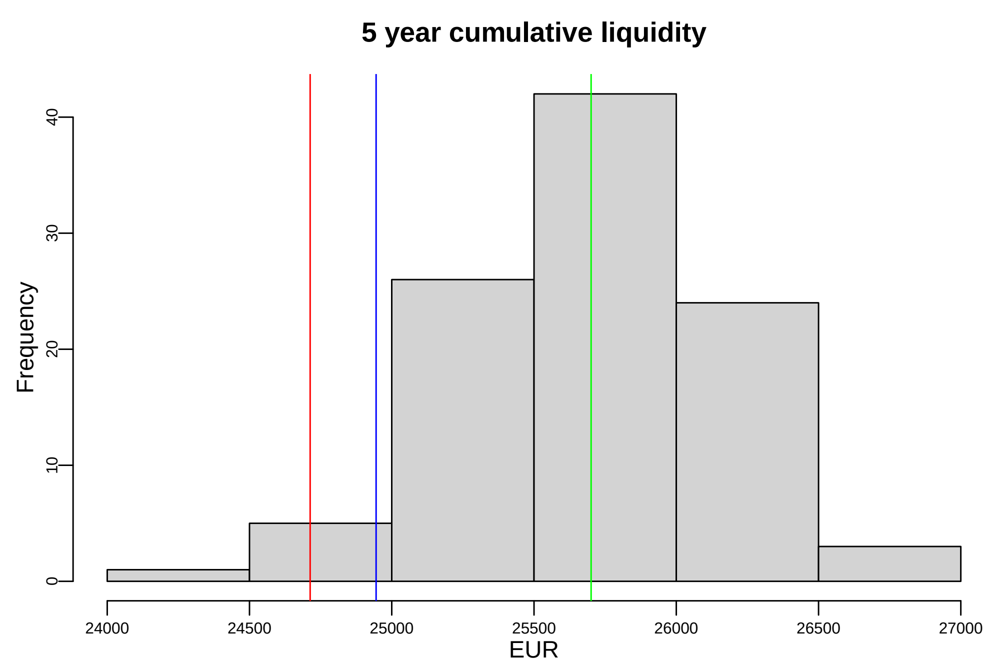
```r
## ----------------------------------------------------------------------------------------------------
liq.cumul <- liq
for (i in 1:101) {
  liq.cumul[,i] <- cumsum(liq[,i])
}
liq.cumul[,1:10]
```

```
R>       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
R> 2016 11031 11031 11031 11031 11031 11031 11031 11031 11031 11031
R> 2017 14775 14775 14775 14776 14776 14776 14777 14777 14777 14777
R> 2018 17427 17422 17417 17413 17407 17402 17398 17393 17388 17383
R> 2019 25022 25012 25001 24992 24981 24971 24961 24951 24941 24930
R> 2020 25869 25833 25797 25762 25725 25690 25654 25618 25582 25546
```

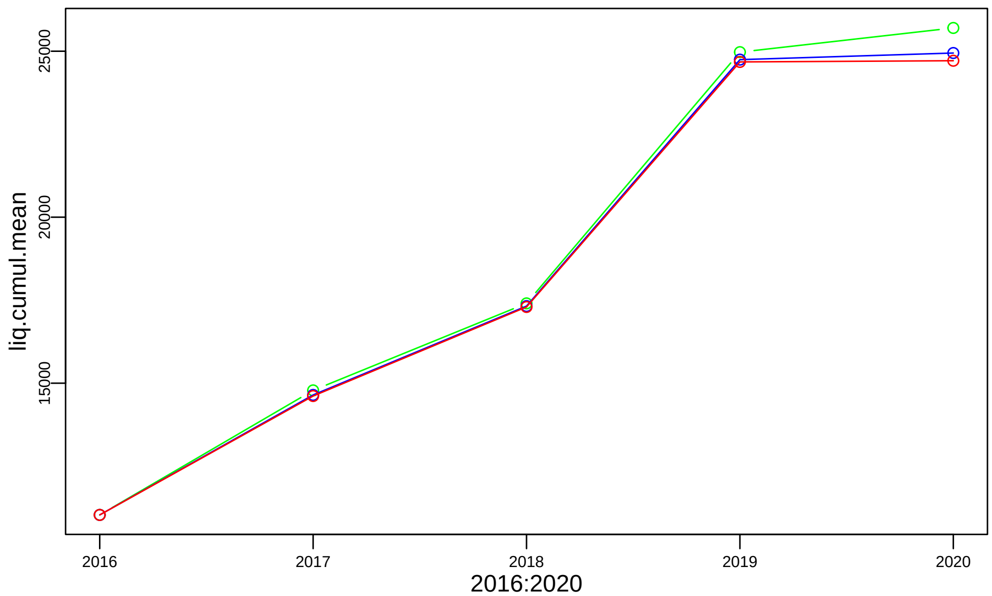
```r
## ----------------------------------------------------------------------------------------------------
liq.cumul.mean <- rowMeans(liq.cumul)
liq.cumul.var <- apply(liq.cumul, 1, quantile, probs=1-alpha)
liq.cumul.es <- liq.cumul.var
for (i in 1:5) {
  liq.cumul.es[i] <- mean(liq.cumul[i, liq.cumul[i,]<=liq.cumul.var[i]])
}
cbind(liq.cumul.mean, liq.cumul.var, liq.cumul.es)
```

```
R>      liq.cumul.mean liq.cumul.var liq.cumul.es
R> 2016       11031.00         11031     11031.00
R> 2017       14782.98         14645     14613.00
R> 2018       17403.55         17322     17295.67
R> 2019       24972.05         24746     24676.67
R> 2020       25700.62         24945     24713.17
```

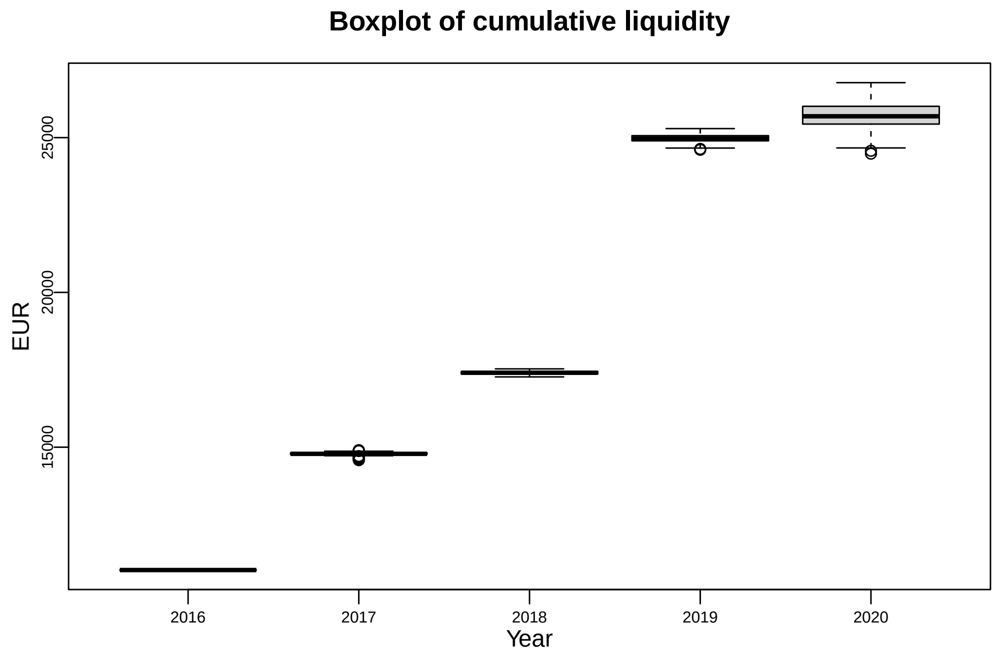
```r
par(mfrow=c(1,1))
dd <- "2020"
hist(liq.cumul[dd,], main="5 year cumulative liquidity", xlab="EUR")
abline(v=liq.cumul.mean[dd],col="green")
abline(v=liq.cumul.var[dd],col="blue")
abline(v=liq.cumul.es[dd],col="red")
```



```r
plot(2016:2020, liq.cumul.mean, type="b", col="green")
lines(2016:2020, liq.cumul.var, col="blue")
points(2016:2020, liq.cumul.var, col="blue")
lines(2016:2020, liq.cumul.es, col="red")
points(2016:2020, liq.cumul.es, col="red")
```

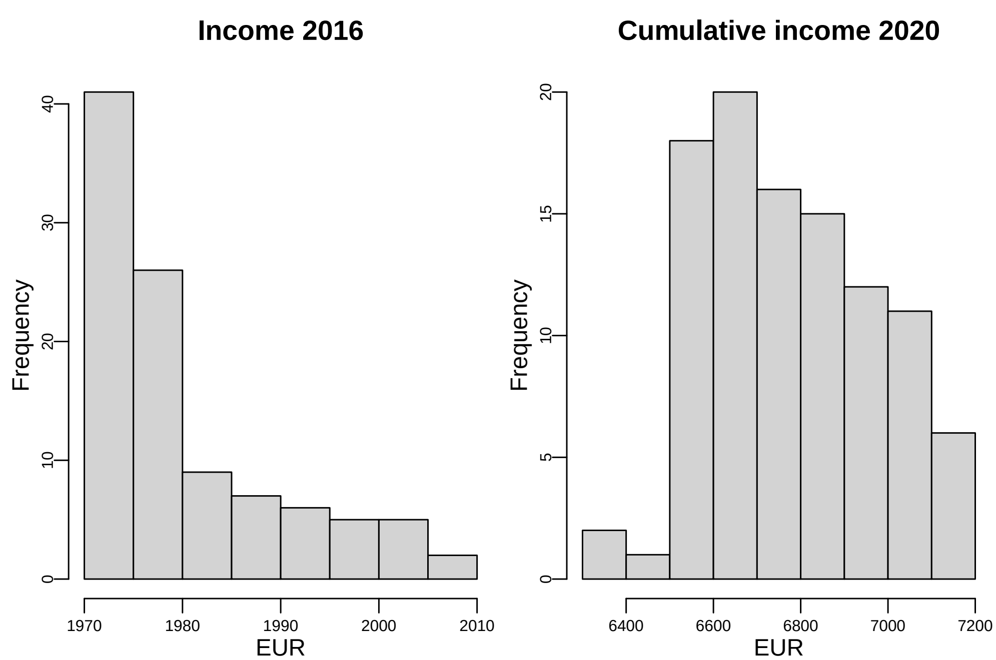



```r
liq.cumul.list = list(
  "2016" = liq.cumul["2016",],
  "2017" = liq.cumul["2017",],
  "2018" = liq.cumul["2018",],
  "2019" = liq.cumul["2019",],
  "2020" = liq.cumul["2020",])

par(mfrow=c(1,1))
boxplot(liq.cumul.list, main="Boxplot of cumulative liquidity",
        xlab="Year",ylab="EUR")
```

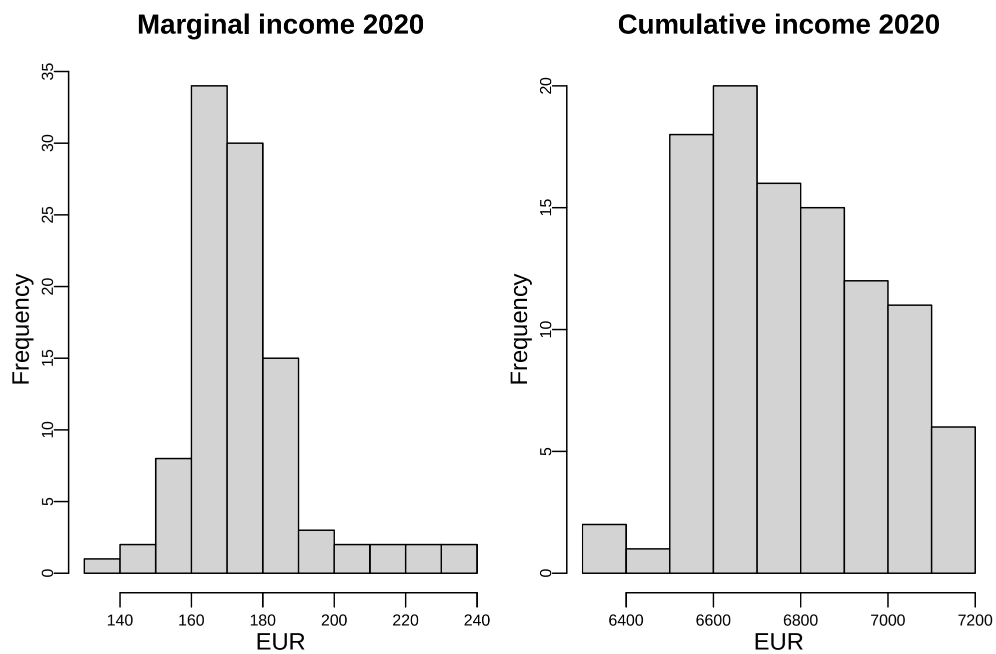


9. Carry out a similar study for income.

```
inc <- sapply(results, function(x) as.numeric(x$income[1,]))
rownames(inc) <- 2016:2020

par(mfrow=c(1,2))
hist(inc["2016",],
  main="Income 2016", xlab="EUR")
hist(colSums(inc),
  main="Cumulative income 2020", xlab="EUR")
```



```
par(mfrow=c(1,2))
hist(inc["2020",], main="Marginal income 2020", xlab="EUR")
hist(colSums(inc), main="Cumulative income 2020", xlab="EUR")
```



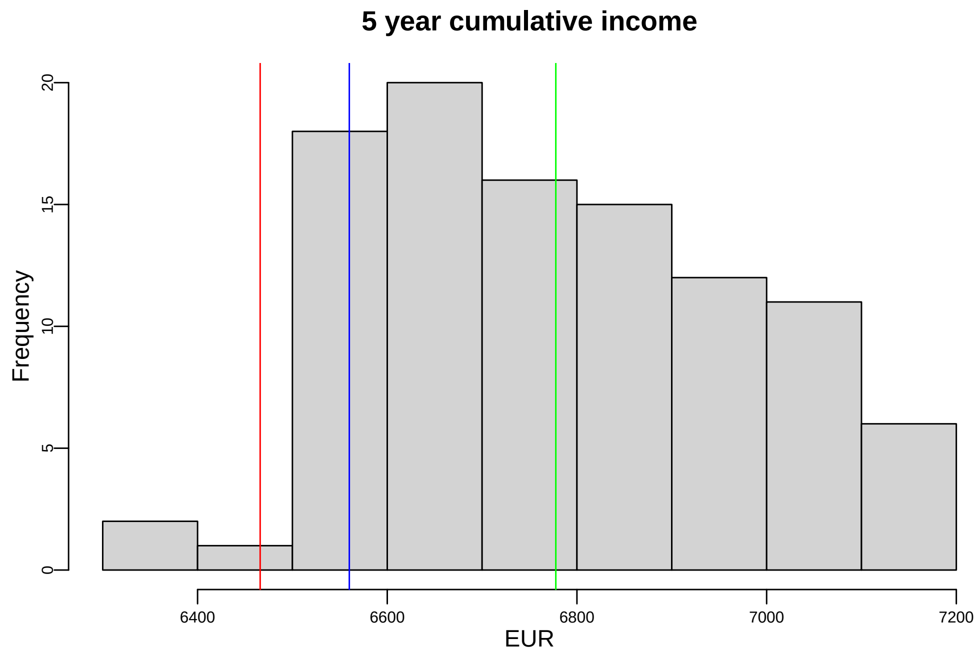

```
inc.cumul <- inc
for (i in 1:101) {
  inc.cumul[,i] <- cumsum(inc[,i])
}
inc.cumul[,1:10]
```

```
R>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
R> 2016 1975 1977 1979 1981 1983 1985 1987 1989 1991  1993
R> 2017 3580 3590 3600 3610 3620 3630 3640 3650 3661  3671
R> 2018 5074 5091 5108 5126 5143 5160 5177 5194 5213  5230
R> 2019 6479 6507 6534 6563 6590 6618 6646 6673 6703  6731
R> 2020 6654 6681 6706 6734 6760 6786 6813 6839 6868  6895
```

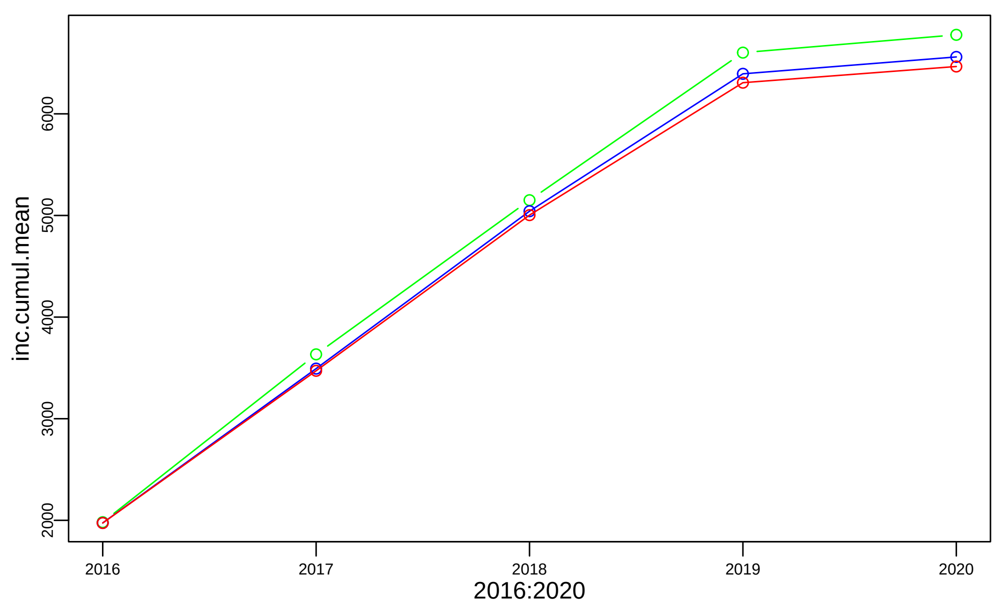
```
inc.cumul.mean <- rowMeans(inc.cumul)
inc.cumul.var <- apply(inc.cumul, 1, quantile, probs=1-alpha)
inc.cumul.es <- inc.cumul.var
for (i in 1:5) {
  inc.cumul.es[i] <- mean(inc.cumul[i, inc.cumul[i,]<=inc.cumul.var[i]])
}
cbind(inc.cumul.mean, inc.cumul.var, inc.cumul.es)
```

```
R>      inc.cumul.mean inc.cumul.var inc.cumul.es
R> 2016       1981.396          1974     1973.667
R> 2017       3633.208          3493     3471.500
R> 2018       5150.287          5042     5003.500
R> 2019       6602.564          6394     6306.667
R> 2020       6777.733          6560     6466.000
```

```
par(mfrow=c(1,1))
dd <- "2020"
hist(inc.cumul[dd,], main="5 year cumulative income", xlab="EUR")
abline(v=inc.cumul.mean[dd],col="green")
abline(v=inc.cumul.var[dd],col="blue")
abline(v=inc.cumul.es[dd],col="red")
```




```r
plot(2016:2020, inc.cumul.mean, type="b", col="green")
lines(2016:2020, inc.cumul.var, col="blue")
points(2016:2020, inc.cumul.var, col="blue")
lines(2016:2020, inc.cumul.es, col="red")
points(2016:2020, inc.cumul.es, col="red")
```




```r
inc.cumul.list = list(
  "2016" = inc.cumul["2016",],
  "2017" = inc.cumul["2017",],
  "2018" = inc.cumul["2018",],
  "2019" = inc.cumul["2019",],
  "2020" = inc.cumul["2020",])
```

```r
par(mfrow=c(1,1))
boxplot(inc.cumul.list, main="Boxplot of cumulative income",
        xlab="Year",ylab="EUR")
```