IT-Based Management Summary

Group 06

11 12 2018

Forecasting and Probalistic Budgeting: Sales Volume Context (Learning with Case Studies)

Use Case

It is about Candle Manufacturing Inc. company, which produce three different candles like molded candles, solid candles and pulled candles. The company needs to setup the annual sales budget for the next year. To accomplish this, the traditional *time series-based forecasting* methods are applied and two new approaches has been introduced:
multinominal reggression-based
stochastic process-based forecasting of time series
To answer question about sales volumes that can expected for next year, the R software package are used as we can

- perform a traditional time series analysis and use the model for **time series-based forecasting** for the next year
- use timely intervals and perform a **regression-based forecasting**
- read monthly sales from transactional data
- perform a **stochastic process-based forecasting**
- derive year-end forecast in fixed-event form
- budget-forecast deviation with p-value to perform adjustments

Problem Statement

As there is uncertainity in business environment, the future predictation of sales volumes is also uncertain and time series-based approch forecast data with the help of historical data. It is not clear which time-based approch should be used in R to produce unbiased data. The two new approaches that are mentioned above allow additional forecast updating which are not possible in time series-based approches. Updating feature is very important in the fixed-event forecasting[1] context as fixed time period is considered over time leading to a successively reducing of the lead time.

Contribution

Traditional time series analysis covers the range between decomposition approaches, where non-stationary trends and seasonalities are included, and stationary ARIMA approaches. By introducing new modeling techniques in form of the multinomial regression-based and the stochastic processbased forecasting methods the time series modeling repertoire can be extended in an effective and easy to grasp way.

[1] In the management control the fixed-event forecasts are the year-end forecasts

Result
Multinomial regression-based and stochastic processbased forecasting are mathematically defined and statistically calibrated.

Research Methodology

The new artifacts are mathematically modeled and their applications are demonstrated in numerical examples.

Research literature

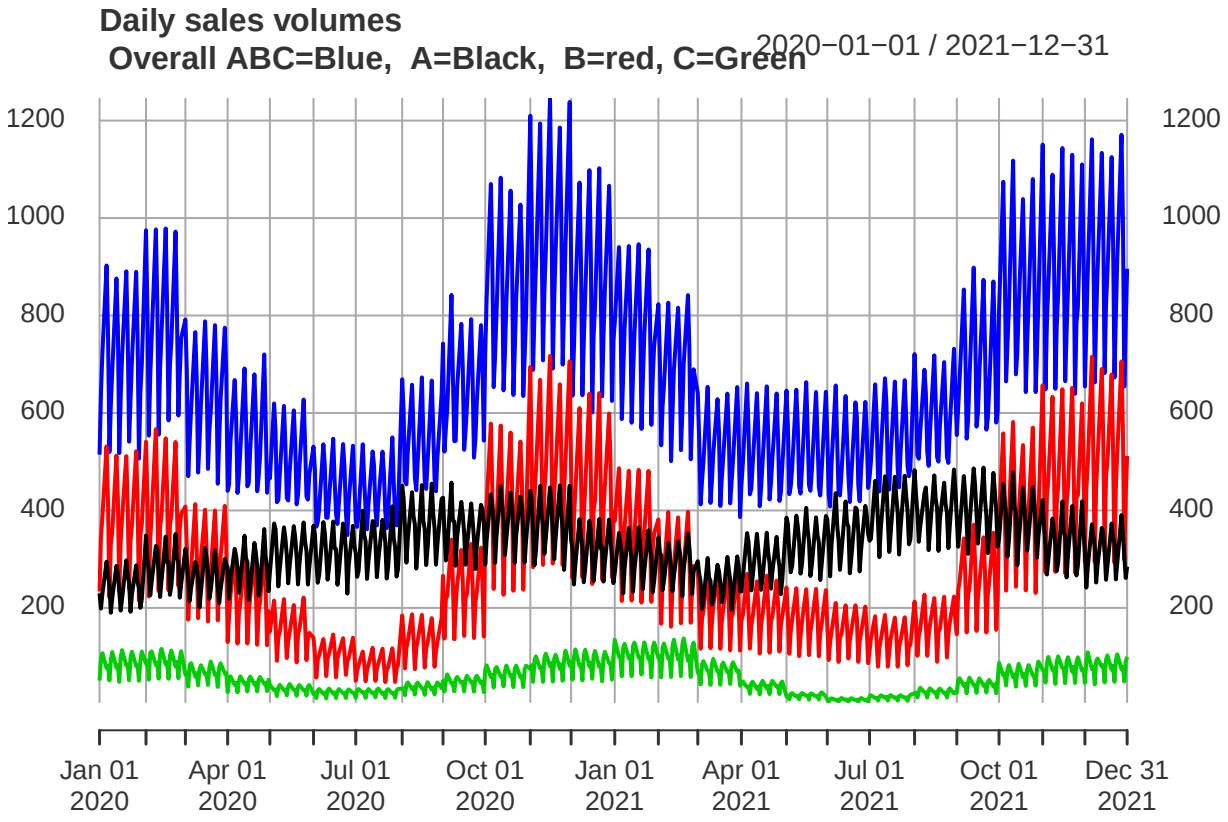
Hyndman Rob/Athanasopoulos George: Forecasting Principles and Practice
Lawrence Michael/O'Connor Marcus: Sales forecasting updates: how good are they in practice?

Implementation

Task 1

Importing and analyzing daily sales data and plotting of xSD01.xts

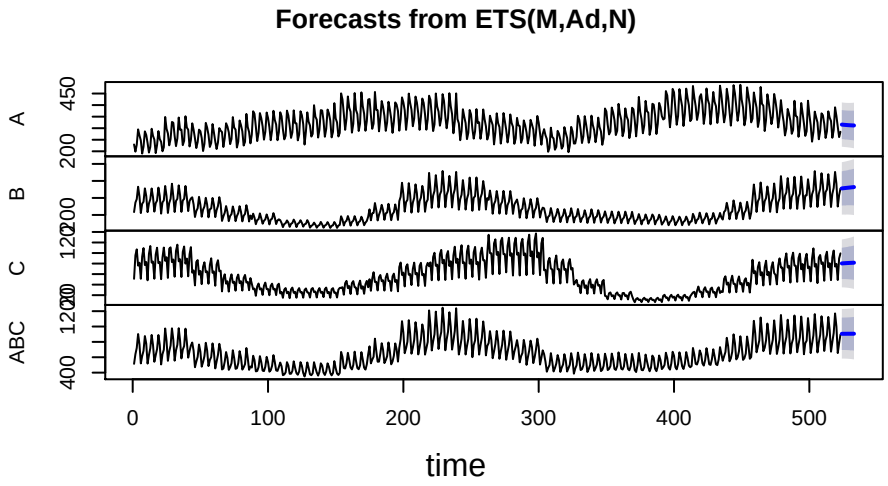
```
library(xts)
## Warning: package 'xts' was built under R version 3.4.4
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 3.4.4
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
xSD01.xts<-readRDS("DataCMI")
str(xSD01.xts)
## Warning in format.POSIXlt(as.POSIXlt(x), ...): unknown timezone 'zone/tz/
## 2018g.1.0/zoneinfo/Europe/Vienna'
## An 'xts' object on 2020-01-01/2021-12-31 containing:
##   Data: num [1:523, 1:4] 230 199 228 295 269 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:4] "A" "B" "C" "ABC"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##  NULL
plot(xSD01.xts, main="Daily sales volumes \n Overall ABC=Blue,  A=Black,  B=red, C=Green")
```



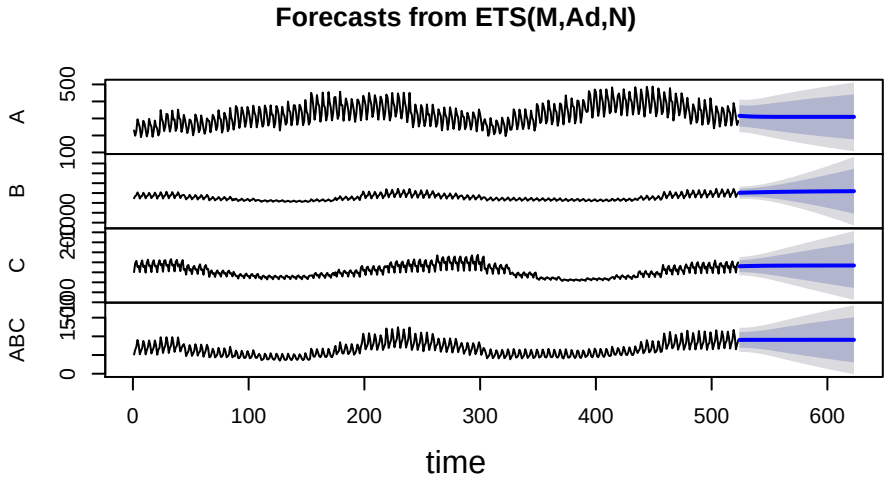
Task 2

Forecasting daily sales volumes by using automated procedures in forecasting packages

```
library(forecast)
## Warning: package 'forecast' was built under R version 3.4.4
plot(forecast(as.ts(xSD01.xts)))
```




```
plot(forecast(as.ts(xSD01.xts),100))
```




```
xSD01.fc<-forecast(as.ts(xSD01.xts),100)
attributes(xSD01.fc)
## $names
## [1] "forecast" "method"
##
## $class
## [1] "mforecast"
```

```
xSD01.fc$method
```

```
##              A              B              C            ABC
## "ETS(M,Ad,N)" "ETS(M,Ad,N)" "ETS(M,Ad,N)" "ETS(M,Ad,N)"
```

```
xSD01.fc$method
```

```
##              A              B              C            ABC
```

```
## "ETS(M,Ad,N)" "ETS(M,Ad,N)" "ETS(M,Ad,N)" "ETS(M,Ad,N)"
```

Interpretation: Planning horizon (plan period) of 100 days already shows exploding uncertainties of ABC, B and C

Task 3

Including additional (predictable) variables into the TS data set

```
xSD01Ext.xts<-NULL # Extracting date information from xts-object
xSD01Ext.xts<-xSD01.xts
xSD01Ext.xts$Year<-as.factor(format(index(xSD01.xts),"%Y"))
xSD01Ext.xts$Quarter<-as.factor(quarters(index(xSD01.xts)))
xSD01Ext.xts$Month<-format(index(xSD01.xts),"%m")
xSD01Ext.xts$wDay<-format(index(xSD01.xts),"%u")
```

Task 4

Performing numeric regressions Numeric regression: Regressing daily sales volumes against numeric variables

```
xSD01ABC.svlm <- NULL # Single variable linear model (svlm)
xSD01ABC.svlm <- lm(ABC~wDay,xSD01Ext.xts)
summary(xSD01ABC.svlm)
```

```
##
## Call:
## lm(formula = ABC ~ wDay, data = xSD01Ext.xts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -285.28 -138.50  -41.32  126.29  492.94
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  802.001     18.156  44.172   <2e-16 ***
## wDay         -50.055      5.466  -9.157   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 176.7 on 521 degrees of freedom
## Multiple R-squared:  0.1386, Adjusted R-squared:  0.137
## F-statistic: 83.85 on 1 and 521 DF,  p-value: < 2.2e-16
```

Task 5

Performing multinomial regressions Multinomial (factor) regression: Regressing daily sales volumes against categorical variables

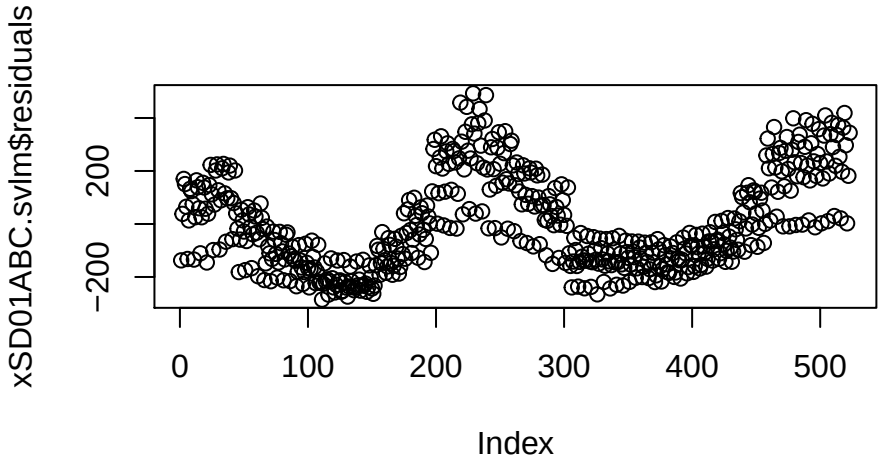
```
xSD01ABC.sflm <- NULL # Single factor linear model (sflm)
xSD01ABC.sflm <- lm(ABC~as.factor(wDay),xSD01Ext.xts)
summary(xSD01ABC.sflm)
```

```
##
## Call:
## lm(formula = ABC ~ as.factor(wDay), data = xSD01Ext.xts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -301.14 -133.00  -25.36  122.85  423.19
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        821.70      15.55  52.844  < 2e-16 ***
## as.factor(wDay)2  -113.65      21.99  -5.168 3.38e-07 ***
## as.factor(wDay)3  -301.69      21.94 -13.752  < 2e-16 ***
## as.factor(wDay)4  -250.92      21.94 -11.438  < 2e-16 ***
## as.factor(wDay)5  -182.33      21.94  -8.311 8.38e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 158.6 on 518 degrees of freedom
## Multiple R-squared:  0.3103, Adjusted R-squared:  0.305
## F-statistic: 58.26 on 4 and 518 DF,  p-value: < 2.2e-16
```

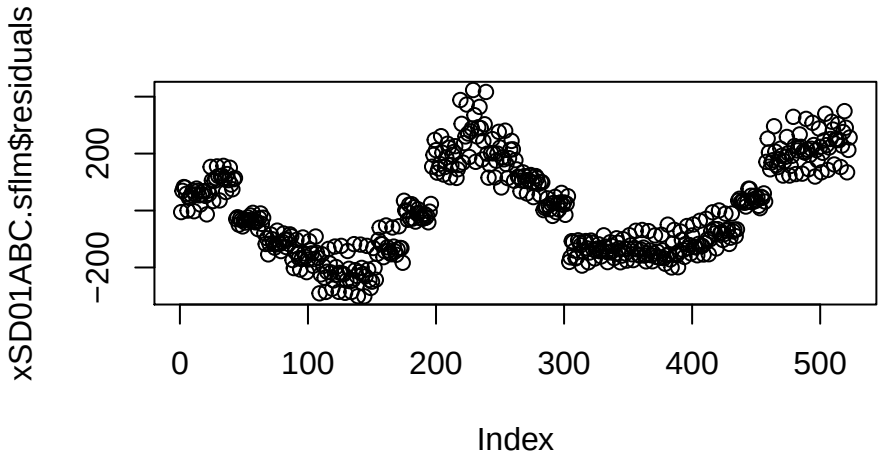
```
attributes(xSD01ABC.sflm)
```

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "contrasts"     "xlevels"       "call"          "terms"
## [13] "model"
##
## $class
## [1] "lm"
```

```
plot(xSD01ABC.svlm$residuals) # Plotting svlm & sflm residuals
```




```
plot(xSD01ABC.sflm$residuals)
```

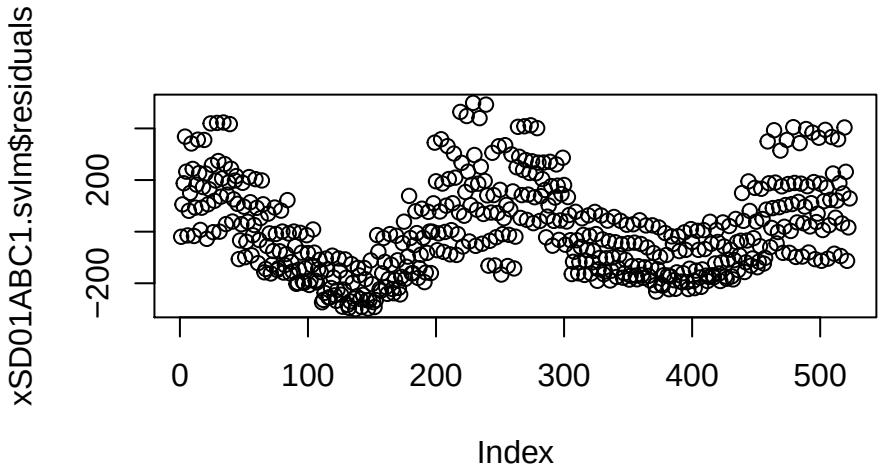


Task 6

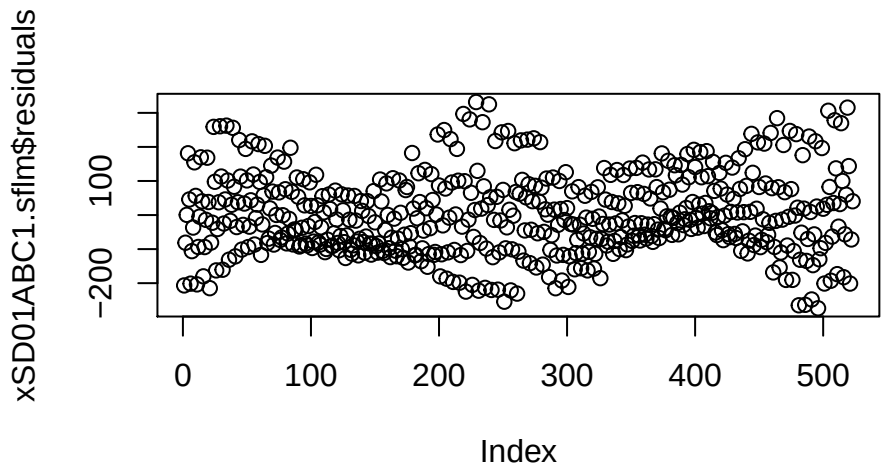
Performing numeric and factor regressions

```
 xSD01ABC1.svlm <- NULL # Single linear regression
 xSD01ABC1.svlm <- lm(ABC~Month,xSD01Ext.xts)
 summary(xSD01ABC1.svlm)
##
## Call:
## lm(formula = ABC ~ Month, data = xSD01Ext.xts)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -300.0 -142.9  -24.6  110.2  498.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  513.185     16.444  31.208   <2e-16 ***
## Month         21.165      2.224   9.517   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 175.7 on 521 degrees of freedom
## Multiple R-squared:  0.1481, Adjusted R-squared:  0.1465
## F-statistic: 90.58 on 1 and 521 DF,  p-value: < 2.2e-16

 xSD01ABC1.sflm <- NULL # Single linear factor regression
 xSD01ABC1.sflm <- lm(ABC~as.factor(Month),xSD01Ext.xts)
 summary(xSD01ABC1.sflm)
##
## Call:
## lm(formula = ABC ~ as.factor(Month), data = xSD01Ext.xts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -273.72  -85.22  -13.50   76.41  331.76
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          721.192     18.075  39.900  < 2e-16 ***
## as.factor(Month)2     -5.186     26.193  -0.198   0.8431
## as.factor(Month)3   -149.497     25.420  -5.881 7.37e-09 ***
## as.factor(Month)4   -198.680     25.562  -7.772 4.26e-14 ***
## as.factor(Month)5   -211.938     25.865  -8.194 2.04e-15 ***
## as.factor(Month)6   -245.721     25.562  -9.613  < 2e-16 ***
## as.factor(Month)7   -241.487     25.420  -9.500  < 2e-16 ***
## as.factor(Month)8   -156.229     25.710  -6.077 2.40e-09 ***
## as.factor(Month)9    -61.173     25.562  -2.393   0.0171 *
## as.factor(Month)10   112.040     25.710   4.358 1.59e-05 ***
## as.factor(Month)11   191.931     25.710   7.465 3.61e-13 ***
## as.factor(Month)12   134.171     25.283   5.307 1.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 119.9 on 511 degrees of freedom
## Multiple R-squared:  0.6111, Adjusted R-squared:  0.6027
## F-statistic: 72.98 on 11 and 511 DF,  p-value: < 2.2e-16

 plot(xSD01ABC1.svlm$residuals)
```



```r
plot(xSD01ABC1.sflm$residuals)
```

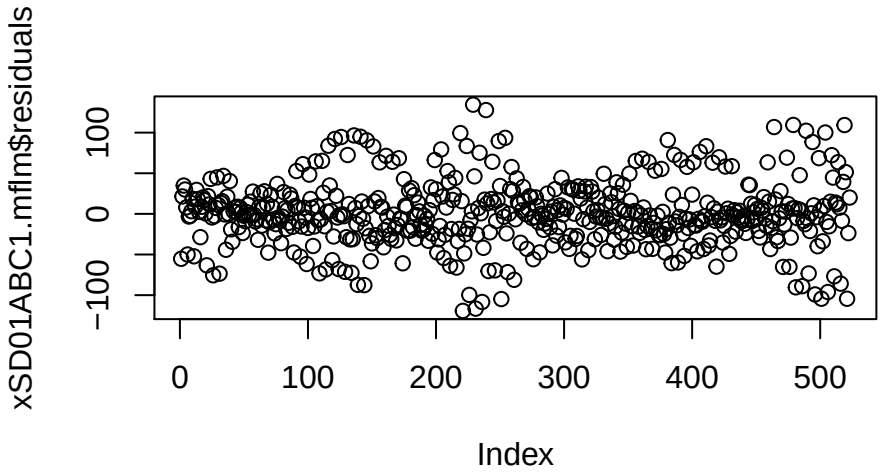


Task 7

Performing multi-factor regressions

```r
xSD01ABC.mflm <- NULL # Multi-factor linear regression
xSD01ABC.mflm <- lm(ABC~as.factor(wDay)+as.factor(Month)+as.factor(Year),xSD01Ext.xts)
summary(xSD01ABC.mflm)
##
## Call:
## lm(formula = ABC ~ as.factor(wDay) + as.factor(Month) + as.factor(Year),
##     data = xSD01Ext.xts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -138.042  -34.790    1.196   34.452  175.543
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          892.622      9.462  94.340  < 2e-16 ***
## as.factor(wDay)2    -114.006      7.277 -15.668  < 2e-16 ***
## as.factor(wDay)3    -303.215      7.265 -41.736  < 2e-16 ***
## as.factor(wDay)4    -253.873      7.268 -34.929  < 2e-16 ***
## as.factor(wDay)5    -185.467      7.268 -25.520  < 2e-16 ***
## as.factor(Month)2    -10.961     11.462  -0.956    0.339
## as.factor(Month)3   -162.627     11.130 -14.612  < 2e-16 ***
## as.factor(Month)4   -197.383     11.185 -17.647  < 2e-16 ***
## as.factor(Month)5   -221.454     11.318 -19.566  < 2e-16 ***
## as.factor(Month)6   -254.996     11.190 -22.788  < 2e-16 ***
## as.factor(Month)7   -239.907     11.122 -21.571  < 2e-16 ***
## as.factor(Month)8   -171.436     11.256 -15.231  < 2e-16 ***
## as.factor(Month)9    -60.378     11.187  -5.397 1.04e-07 ***
## as.factor(Month)10   108.975     11.249   9.688  < 2e-16 ***
## as.factor(Month)11   176.724     11.256  15.700  < 2e-16 ***
## as.factor(Month)12   136.782     11.063  12.363  < 2e-16 ***
## as.factor(Year)2      11.316      4.588   2.466    0.014 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.45 on 506 degrees of freedom
## Multiple R-squared:  0.9263, Adjusted R-squared:  0.924
## F-statistic: 397.4 on 16 and 506 DF,  p-value: < 2.2e-16

 plot(xSD01ABC.mflm$residuals)
```

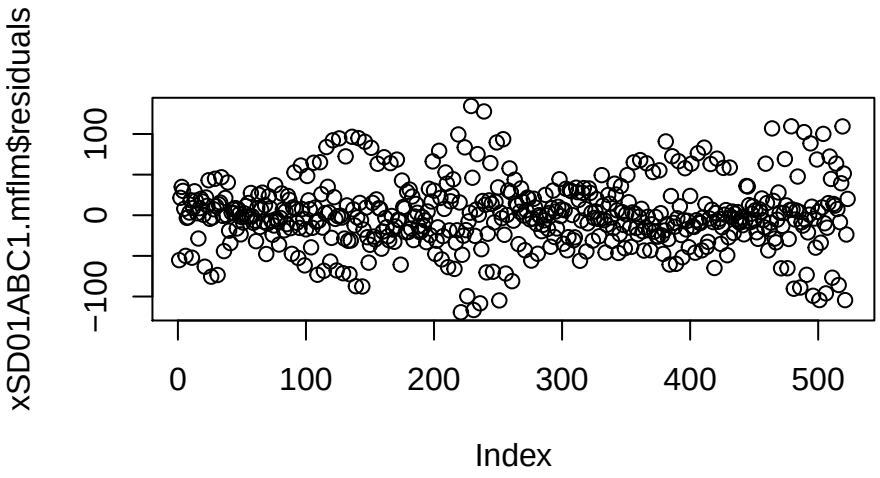


Task 8

Performing multi-factor regressions

```
xSD01ABC1.mflm <- NULL # Multi-factor multinomial linear regression
xSD01ABC1.mflm <- lm(ABC~as.factor(wDay)+as.factor(Month)*as.factor(Year),xSD01Ext.xts)
summary(xSD01ABC1.mflm)
##
## Call:
## lm(formula = ABC ~ as.factor(wDay) + as.factor(Month) * as.factor(Year),
##     data = xSD01Ext.xts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.347  -20.819   -1.485   18.326  134.400
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         872.6171     9.3329  93.499  < 2e-16
## as.factor(wDay)2                   -113.7233     5.6953 -19.968  < 2e-16
## as.factor(wDay)3                   -302.8158     5.6872 -53.245  < 2e-16
## as.factor(wDay)4                   -254.1467     5.6884 -44.678  < 2e-16
## as.factor(wDay)5                   -185.7185     5.6885 -32.648  < 2e-16
## as.factor(Month)2                    59.2000    12.5497   4.717 3.11e-06
## as.factor(Month)3                   -83.1751    12.2457  -6.792 3.18e-11
## as.factor(Month)4                  -157.2783    12.2404 -12.849  < 2e-16
## as.factor(Month)5                  -205.6087    12.3890 -16.596  < 2e-16
## as.factor(Month)6                  -268.4976    12.2457 -21.926  < 2e-16
## as.factor(Month)7                  -264.3774    12.1026 -21.845  < 2e-16
## as.factor(Month)8                  -173.5622    12.3917 -14.006  < 2e-16
```

```
## as.factor(Month)9                    -59.2804    12.2431  -4.842 1.72e-06
## as.factor(Month)10                   130.8758    12.2404  10.692  < 2e-16
## as.factor(Month)11                   237.8703    12.3917  19.196  < 2e-16
## as.factor(Month)12                   135.9623    12.1052  11.232  < 2e-16
## as.factor(Year)2                      53.1882    12.3890   4.293 2.12e-05
## as.factor(Month)2:as.factor(Year)2  -142.2478    17.9425  -7.928 1.49e-14
## as.factor(Month)3:as.factor(Year)2  -158.2801    17.4171  -9.088  < 2e-16
## as.factor(Month)4:as.factor(Year)2   -82.1110    17.5105  -4.689 3.55e-06
## as.factor(Month)5:as.factor(Year)2   -33.6024    17.7198  -1.896  0.05850
## as.factor(Month)6:as.factor(Year)2    25.0393    17.5143   1.430  0.15345
## as.factor(Month)7:as.factor(Year)2    49.0720    17.4149   2.818  0.00503
## as.factor(Month)8:as.factor(Year)2     1.3128    17.6145   0.075  0.94062
## as.factor(Month)9:as.factor(Year)2    -4.1531    17.5143  -0.237  0.81265
## as.factor(Month)10:as.factor(Year)2  -45.7773    17.6140  -2.599  0.00963
## as.factor(Month)11:as.factor(Year)2 -122.3568    17.6145  -6.946 1.18e-11
## as.factor(Month)12:as.factor(Year)2   -0.2899    17.3210  -0.017  0.98665
##
## (Intercept)                          ***
## as.factor(wDay)2                     ***
## as.factor(wDay)3                     ***
## as.factor(wDay)4                     ***
## as.factor(wDay)5                     ***
## as.factor(Month)2                    ***
## as.factor(Month)3                    ***
## as.factor(Month)4                    ***
## as.factor(Month)5                    ***
## as.factor(Month)6                    ***
## as.factor(Month)7                    ***
## as.factor(Month)8                    ***
## as.factor(Month)9                    ***
## as.factor(Month)10                   ***
## as.factor(Month)11                   ***
## as.factor(Month)12                   ***
## as.factor(Year)2                     ***
## as.factor(Month)2:as.factor(Year)2   ***
## as.factor(Month)3:as.factor(Year)2   ***
## as.factor(Month)4:as.factor(Year)2   ***
## as.factor(Month)5:as.factor(Year)2   .
## as.factor(Month)6:as.factor(Year)2
## as.factor(Month)7:as.factor(Year)2   **
## as.factor(Month)8:as.factor(Year)2
## as.factor(Month)9:as.factor(Year)2
## as.factor(Month)10:as.factor(Year)2 **
## as.factor(Month)11:as.factor(Year)2 ***
## as.factor(Month)12:as.factor(Year)2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.04 on 495 degrees of freedom
## Multiple R-squared:  0.9559, Adjusted R-squared:  0.9534
## F-statistic: 396.9 on 27 and 495 DF,  p-value: < 2.2e-16

plot(xSD01ABC1.mflm$residuals)
```



Task 9

Multinomial regression-based forecasting the daily sales volumes for the next year

```
PlanPeriod.xts<-NULL # Generating an indexed xts-object
PlanPeriod.xts<-as.xts(seq(as.Date("2022-01-01"),
as.Date("2022-12-31"),by='day'))
PlanPeriod.xts$Year<-c(2)
PlanPeriod.xts$Month<-as.factor(format(index(PlanPeriod.xts),"%m"))
PlanPeriod.xts$wDay<-as.factor(format(index(PlanPeriod.xts),"%u"))
PP.xts<-NULL
PP.xts<-subset(PlanPeriod.xts,!(PlanPeriod.xts$wDay %in% c(6,7)))
```

Task 10

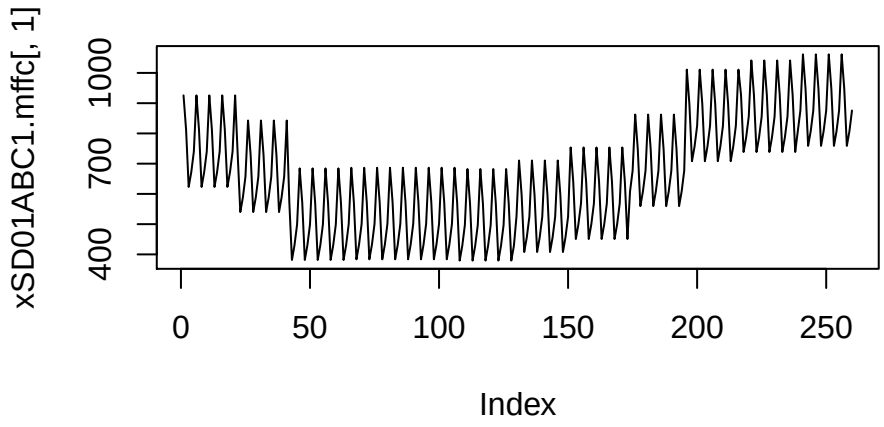
Multi-factor regression-based forecasting

```
xSD01ABC1.mffc<-NULL # Generating predictions
xSD01ABC1.mffc<-predict(xSD01ABC1.mflm,PP.xts,interval = "confidence")
head(xSD01ABC1.mffc)
##                  fit      lwr      upr
## 2022-01-03 925.8052 906.8035 944.8070
## 2022-01-04 812.0819 793.0813 831.0825
## 2022-01-05 622.9894 603.9994 641.9794
## 2022-01-06 671.6585 652.6684 690.6486
## 2022-01-07 740.0867 721.2516 758.9218
## 2022-01-10 925.8052 906.8035 944.8070
  xSD01ABC1.mffc[1,1]
## [1] 925.8052
  xSD01ABC1.mffc[1,2] # lower (lwr) bound
## [1] 906.8035
  xSD01ABC1.mffc[1,3] # upper (upr) bound
## [1] 944.807
  sum(xSD01ABC1.mffc[,1]) # Sum of sales forecasts
## [1] 170750.5
  apply.quarterly(xSD01ABC1.mffc[,1],FUN = sum)# Quarterly forecasts
##                   x
```

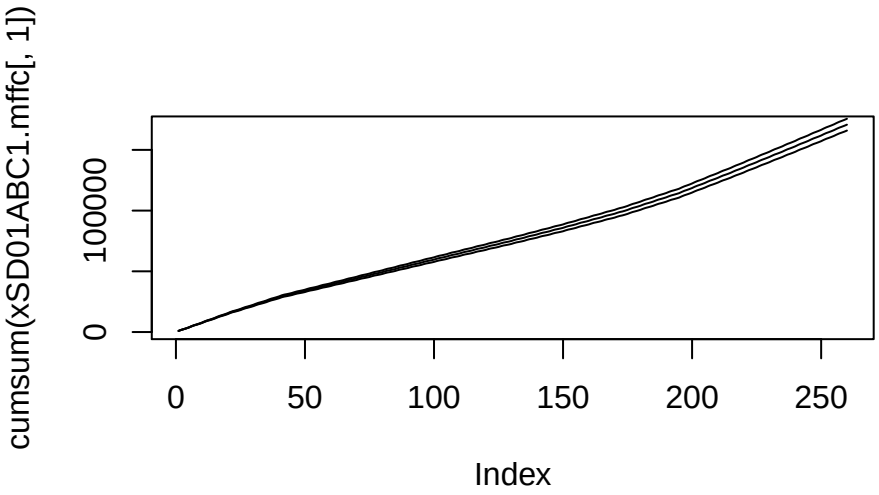
```
## 2022-03-31 41089.57
## 2022-06-30 33398.18
## 2022-09-30 39905.48
## 2022-12-30 56357.24
  apply.monthly(xSD01ABC1.mffc[,1],FUN = sum) # Monthly forecasts
##                   x
## 2022-01-31 16016.29
## 2022-02-28 13429.53
## 2022-03-31 11643.75
## 2022-04-29 10803.40
## 2022-05-31 11565.73
## 2022-06-30 11029.05
## 2022-07-29 11309.16
## 2022-08-31 13489.63
## 2022-09-30 15106.69
## 2022-10-31 17803.36
## 2022-11-30 19066.86
## 2022-12-30 19487.02
```

Historical calibration problem: 1-year sales forecast of 170750.5 is due to the historical calibration similar to last years??? sales volumes

```
plot(xSD01ABC1.mffc[,1],type="l") # Plotting predictions
```

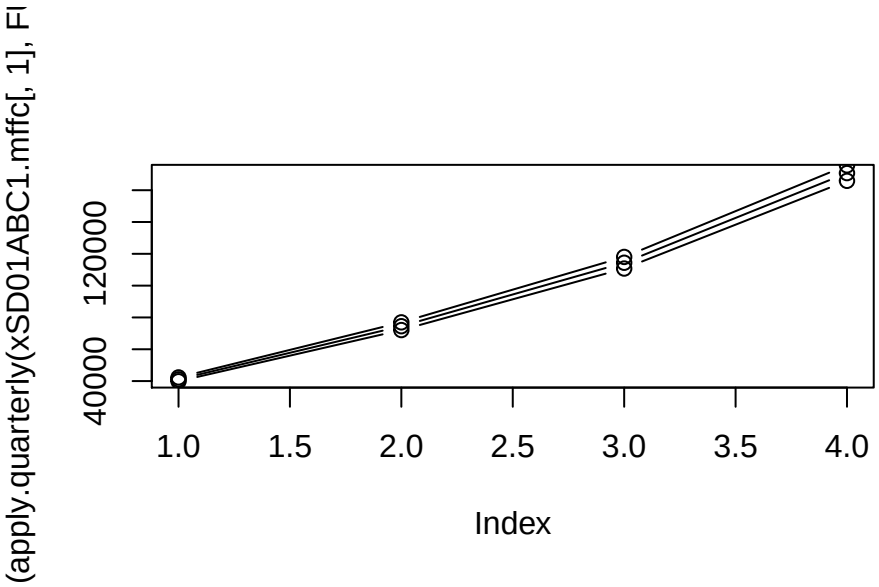



```
plot(cumsum(xSD01ABC1.mffc[,1]),type="l")
lines(cumsum(xSD01ABC1.mffc[,2]))
lines(cumsum(xSD01ABC1.mffc[,3]))
```

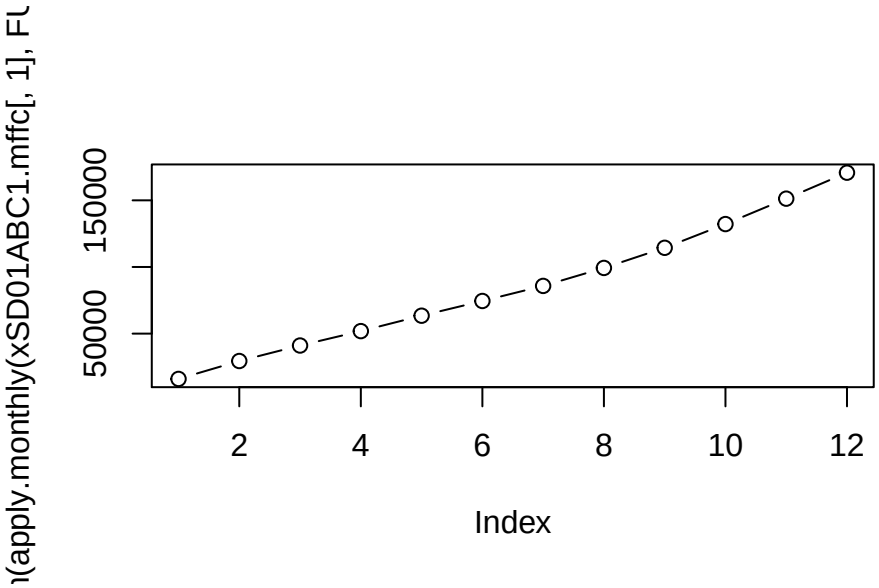


Plotting quaterly and monthly predictions:

```
plot(cumsum(apply.quarterly(xSD01ABC1.mffc[,1],FUN = sum)),type="b")#Q
lines(cumsum(apply.quarterly(xSD01ABC1.mffc[,2],FUN = sum)),type="b")
lines(cumsum(apply.quarterly(xSD01ABC1.mffc[,3],FUN = sum)),type="b")
```



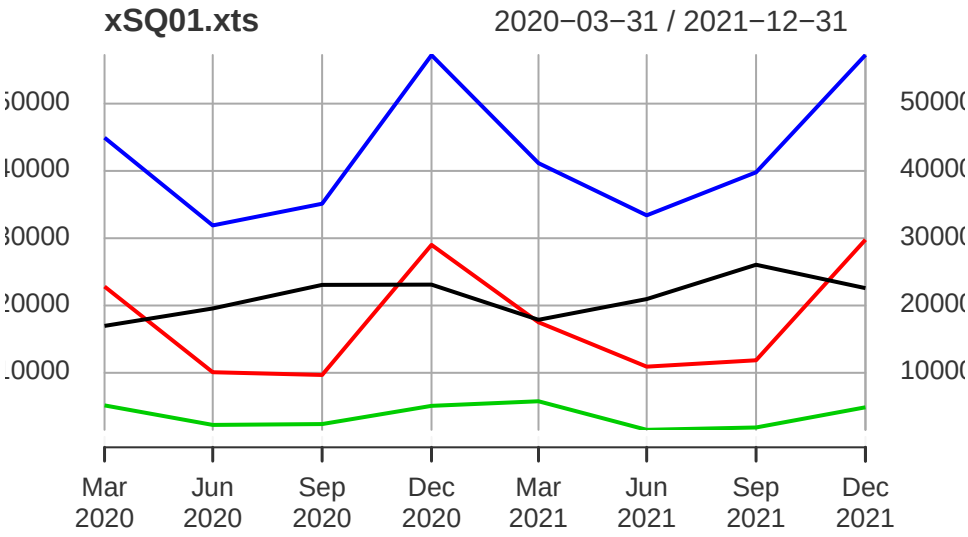

```
plot(cumsum(apply.monthly(xSD01ABC1.mffc[,1],FUN = sum)),type="b") #M
```



Task 11

Quarterly xts-time series xSQ01.xts: Generating lower frequency data (e.g. quarterly sales volumes) via aggregating daily data by using the apply.quarterly() function

```
 xSQ01.xts<-apply.quarterly(xSD01.xts$A,sum) # Quarterly sales volumes
 xSQ01.xts<-cbind(xSQ01.xts,apply.quarterly(xSD01.xts$B,sum),
 apply.quarterly(xSD01.xts$C,sum),apply.quarterly(xSD01.xts$ABC,sum))
 xSQ01.xts
##                  A     B    C   ABC
## 2020-03-31 16968 22818 5155 44941
## 2020-06-30 19560 10087 2255 31902
## 2020-09-30 23066  9679 2382 35127
## 2020-12-31 23104 29013 5090 57207
## 2021-03-31 17876 17512 5770 41158
## 2021-06-30 20957 10915 1526 33398
## 2021-09-30 26057 11862 1875 39794
## 2021-12-31 22581 29779 4873 57233
 apply.yearly(xSD01.xts$ABC,sum)
##                ABC
## 2020-12-31 169177
## 2021-12-31 171583

 plot(xSQ01.xts)
```



Task 12

Regressing the quarterly sales volumes (analogously to the daily sales volumes) w.r.t. predictable calendar information (e.g. quarters) in form of categorical variables

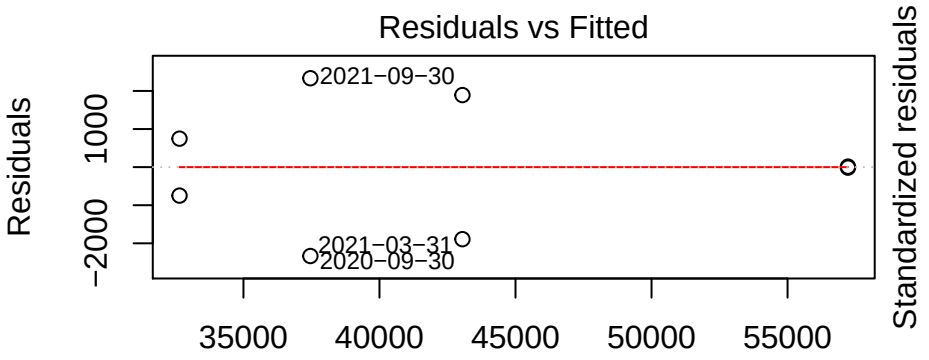
```
 xSQ01Ext.xts<-NULL # Extending the quarterly data base
 xSQ01Ext.xts<-xSQ01.xts
 xSQ01Ext.xts$Quarter<-as.factor(quarters(index(xSQ01.xts)))
 xSQ01Ext.xts
##                A     B    C   ABC Quarter
## 2020-03-31 16968 22818 5155 44941       1
## 2020-06-30 19560 10087 2255 31902       2
## 2020-09-30 23066  9679 2382 35127       3
## 2020-12-31 23104 29013 5090 57207       4
## 2021-03-31 17876 17512 5770 41158       1
## 2021-06-30 20957 10915 1526 33398       2
## 2021-09-30 26057 11862 1875 39794       3
## 2021-12-31 22581 29779 4873 57233       4
xSQ01ABC.sflm <- NULL # Single-factor linear regression
xSQ01ABC.sflm <- lm(ABC~as.factor(Quarter),xSQ01Ext.xts)
summary(xSQ01ABC.sflm)
##
## Call:
## lm(formula = ABC ~ as.factor(Quarter), data = xSQ01Ext.xts)
##
## Residuals:
## 2020-03-31 2020-06-30 2020-09-30 2020-12-31 2021-03-31 2021-06-30
##       1892       -748      -2334        -13      -1892        748
## 2021-09-30 2021-12-31
##       2334         13
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)            43050       1548  27.813 9.94e-06 ***
## as.factor(Quarter)2   -10400       2189  -4.751  0.00896 **
## as.factor(Quarter)3    -5589       2189  -2.553  0.06308 .
```

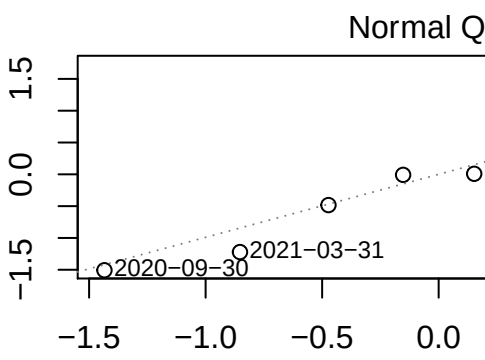
```
## as.factor(Quarter)4    14170       2189   6.474  0.00293 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2189 on 4 degrees of freedom
## Multiple R-squared:  0.9725, Adjusted R-squared:  0.9519
## F-statistic: 47.22 on 3 and 4 DF,  p-value: 0.001401
```

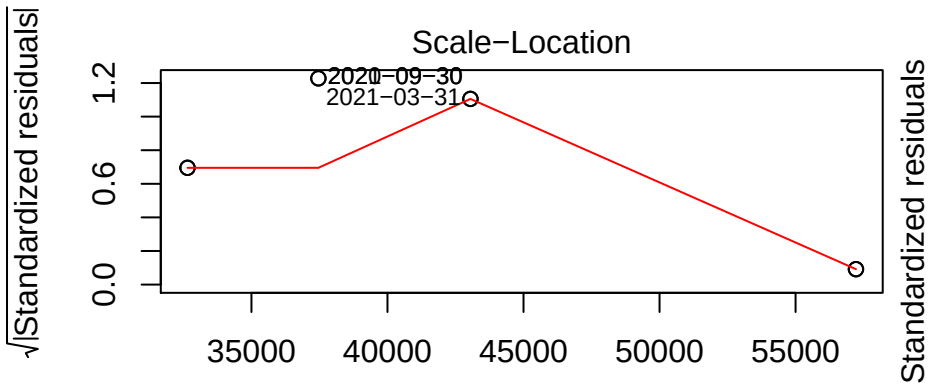
Task 13

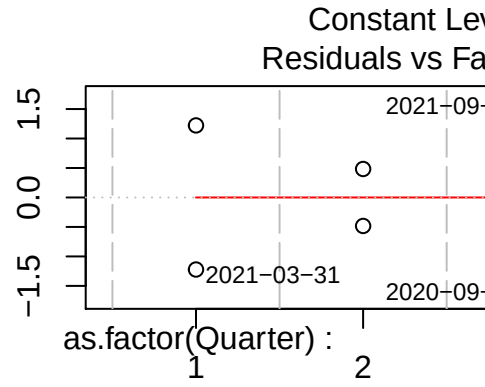
Regressing quarterly sales data without (w/o) intercept term

```
xSQ01ABC1.sflm <- NULL # Single factor linear model
xSQ01ABC1.sflm <- lm(ABC~0+as.factor(Quarter),xSQ01Ext.xts)
summary(xSQ01ABC1.sflm)
##
## Call:
## lm(formula = ABC ~ 0 + as.factor(Quarter), data = xSQ01Ext.xts)
##
## Residuals:
## 2020-03-31 2020-06-30 2020-09-30 2020-12-31 2021-03-31 2021-06-30
##       1892       -748      -2334        -13      -1892        748
## 2021-09-30 2021-12-31
##       2334         13
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## as.factor(Quarter)1    43050       1548   27.81 9.94e-06 ***
## as.factor(Quarter)2    32650       1548   21.09 2.99e-05 ***
## as.factor(Quarter)3    37460       1548   24.20 1.73e-05 ***
## as.factor(Quarter)4    57220       1548   36.97 3.20e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2189 on 4 degrees of freedom
## Multiple R-squared:  0.9987, Adjusted R-squared:  0.9975
## F-statistic: 792.8 on 4 and 4 DF,  p-value: 4.758e-06
attributes(xSQ01ABC1.sflm)
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "contrasts"     "xlevels"       "call"          "terms"
## [13] "model"
##
## $class
## [1] "lm"
sum(xSQ01ABC1.sflm$coefficients)
## [1] 170380
plot(xSQ01ABC1.sflm)
```










```
xSQ01ABC1.sflm$model
##              ABC as.factor(Quarter)
## 2020-03-31 44941                   1
## 2020-06-30 31902                   2
## 2020-09-30 35127                   3
## 2020-12-31 57207                   4
## 2021-03-31 41158                   1
## 2021-06-30 33398                   2
## 2021-09-30 39794                   3
## 2021-12-31 57233                   4
model.matrix(xSQ01ABC1.sflm)
##            as.factor(Quarter)1 as.factor(Quarter)2 as.factor(Quarter)3
## 2020-03-31                   1                   0                   0
## 2020-06-30                   0                   1                   0
## 2020-09-30                   0                   0                   1
## 2020-12-31                   0                   0                   0
## 2021-03-31                   1                   0                   0
## 2021-06-30                   0                   1                   0
## 2021-09-30                   0                   0                   1
## 2021-12-31                   0                   0                   0
##            as.factor(Quarter)4
## 2020-03-31                   0
## 2020-06-30                   0
```

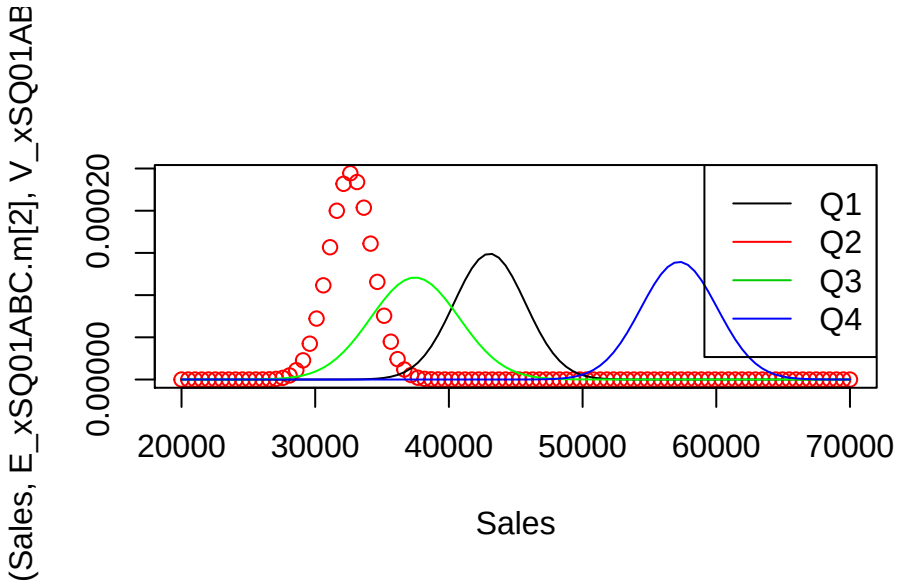
```
## 2020-09-30                   0
## 2020-12-31                   1
## 2021-03-31                   0
## 2021-06-30                   0
## 2021-09-30                   0
## 2021-12-31                   1
## attr(,"assign")
## [1] 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$`as.factor(Quarter)`
## [1] "contr.treatment"
```

Task 14

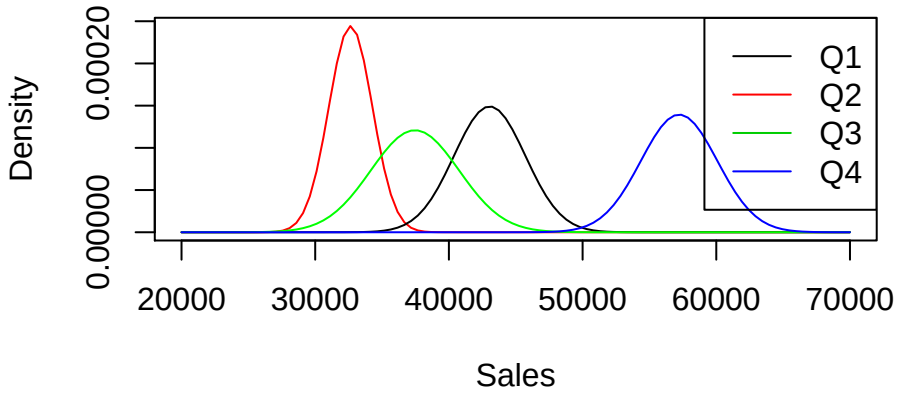
Modeling quarterly sales volumes as Gaussian stochastic processes which are completely specified by the mean vector and the volatility vector in the case of independently distributed sales volumes in the different quarters

```
# Defining and filling of two matrices and an array
 xSQ01ABC_BLY.m<-matrix(coredata(xSQ01.xts[1:4,4]),1,4,byrow=T)
 dimnames(xSQ01ABC_BLY.m)<-list(c("ABC"),c("Q1","Q2","Q3","Q4"))
 xSQ01ABC_LY.m<-matrix(coredata(xSQ01.xts[5:8,4]),1,4,byrow=T)
 dimnames(xSQ01ABC_LY.m)<-list(c("ABC"),c("Q1","Q2","Q3","Q4"))
 xSQ01ABC.a<-array(c(xSQ01ABC_LY.m,xSQ01ABC_BLY.m),dim=c(1,4,2))
 dimnames(xSQ01ABC.a)<-list(c("ABC"),c("Q1","Q2","Q3","Q4"),c("LY","BLY"))
 t(xSQ01ABC.a[1,,])
##        Q1    Q2    Q3    Q4
## LY  41158 33398 39794 57233
## BLY 44941 31902 35127 57207


# Calibrating the mean vector (location parameters)
 E_xSQ01ABC.m<-matrix(0,1,4)
 for(j in 1:4) E_xSQ01ABC.m[1,j]<-mean(xSQ01ABC.a[1,j,])
 dimnames(E_xSQ01ABC.m)<-list(c("ABC"),c("Q1","Q2","Q3","Q4"))
 E_xSQ01ABC.m
##          Q1    Q2      Q3    Q4
## ABC 43049.5 32650 37460.5 57220

# Calibrating the volatility vector (dispersion parameters)
 V_xSQ01ABC.m<-matrix(0,1,4)
 for(j in 1:4) V_xSQ01ABC.m[1,j]<-sd(xSQ01ABC.a[1,j,])
 dimnames(V_xSQ01ABC.m)<-list(c("ABC"),c("Q1","Q2","Q3","Q4"))
 V_xSQ01ABC.m
##           Q1       Q2       Q3       Q4
## ABC 2674.985 1057.832 3300.067 18.38478

  CV_xSQ01ABC.m<-V_xSQ01ABC.m/E_xSQ01ABC.m # Coefficient of variation
  CV_xSQ01ABC.m
##             Q1         Q2         Q3           Q4
## ABC 0.06213742 0.03239913 0.08809459 0.0003212998

   CV_xSQ01ABC1.m<-CV_xSQ01ABC.m
```

```
 CV_xSQ01ABC1.m[CV_xSQ01ABC1.m<0.05]<-0.05 # Setting min. CV to 5 %
 CV_xSQ01ABC1.m
##              Q1   Q2         Q3   Q4
## ABC 0.06213742 0.05 0.08809459 0.05

  V_xSQ01ABC1.m<-V_xSQ01ABC.m
 V_xSQ01ABC1.m<-CV_xSQ01ABC1.m*E_xSQ01ABC.m
 V_xSQ01ABC1.m
##            Q1     Q2       Q3   Q4
## ABC 2674.985 1632.5 3300.067 2861

 Sales<-seq(20000, 70000, length=100) # Normal density distributions
 plot(Sales,dnorm(Sales,E_xSQ01ABC.m[2],V_xSQ01ABC1.m[2]),col="red")
 lines(Sales,dnorm(Sales,E_xSQ01ABC.m[1], V_xSQ01ABC1.m[1]),col="black")
 lines(Sales,dnorm(Sales,E_xSQ01ABC.m[3],V_xSQ01ABC1.m[3]),col="green")
 lines(Sales,dnorm(Sales,E_xSQ01ABC.m[4],V_xSQ01ABC1.m[4]),col="blue")
 legend("topright", legend=c("Q1","Q2","Q3","Q4"),col=1:4,lty=1)
```



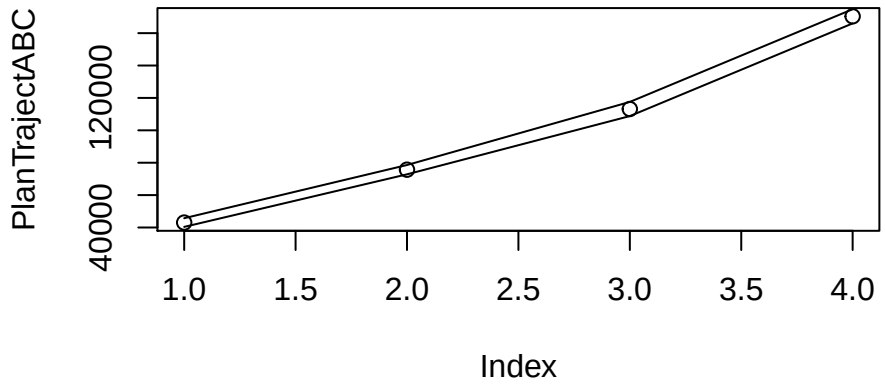

```
  plot(Sales,dnorm(Sales,E_xSQ01ABC.m[2],V_xSQ01ABC1.m[2]), # Mean/Vola
 col="red",type="l",ylab="Density")
 lines(Sales,dnorm(Sales,E_xSQ01ABC.m[1],V_xSQ01ABC1.m[1]),col="black")
 lines(Sales,dnorm(Sales,E_xSQ01ABC.m[3],V_xSQ01ABC1.m[3]),col="green")
 lines(Sales,dnorm(Sales,E_xSQ01ABC.m[4],V_xSQ01ABC1.m[4]),col="blue")
 legend("topright", legend=c("Q1","Q2","Q3","Q4"),col=1:4,lty=1)
```



Task 15

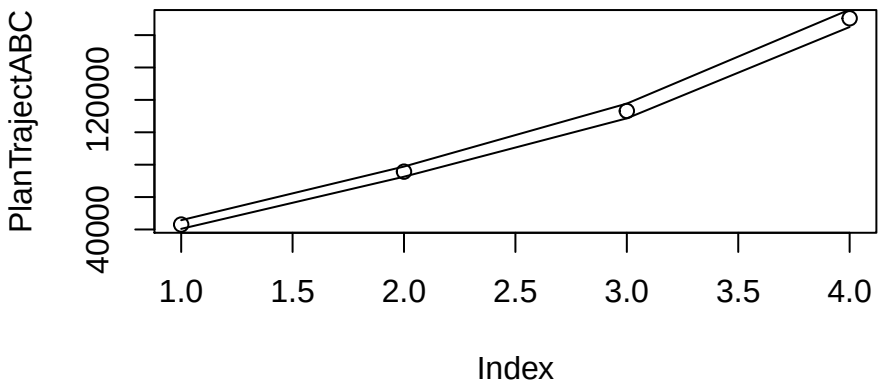
Probabilistic Budgeting: Stochastic process modeling allows calculating the density function of the annual sales volumes (probabilistic budgeting) analytically by temporally aggregating the mean vector elements with the linear form and the volatility vector elements with the quadratic form

```
 PlanTrajectABC<-cumsum(E_xSQ01ABC.m) # Linear temp. aggregation
 PlanTrajectABC
## [1]  43049.5  75699.5 113160.0 170380.0

 plot(PlanTrajectABC)

 BudgetABC<-PlanTrajectABC[4] # Setting the sales budget
 BudgetABC
## [1] 170380


 PlanVolaABC<-sqrt(cumsum((V_xSQ01ABC.m)^2)) # Quadratic temp. aggreg.
 PlanVolaABC
## [1] 2674.985 2876.552 4377.784 4377.823

 lines(PlanTrajectABC+PlanVolaABC)
 lines(PlanTrajectABC-PlanVolaABC)
```

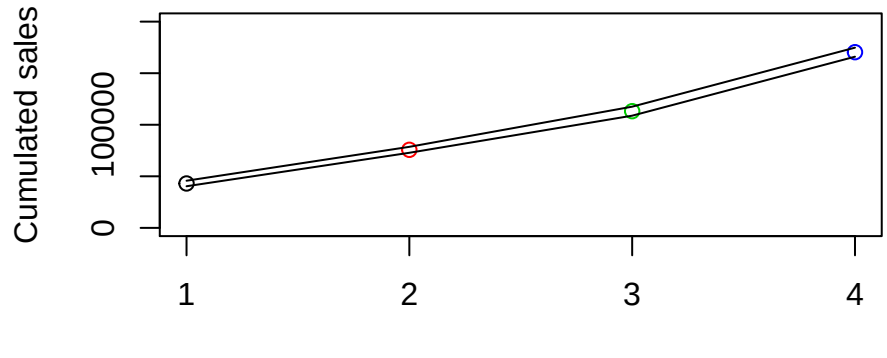



```
 PlanVolaABC1<-sqrt(cumsum((V_xSQ01ABC1.m)^2)) # Volatility adjustment
 PlanVolaABC1
## [1] 2674.985 3133.784 4550.939 5375.534
```

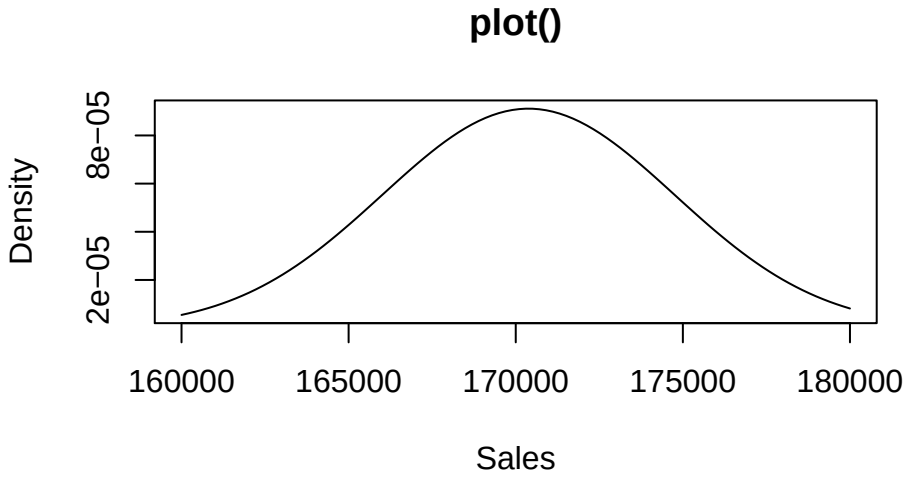
```
plot(PlanTrajectABC)
lines(PlanTrajectABC+PlanVolaABC1)
lines(PlanTrajectABC-PlanVolaABC1)
```



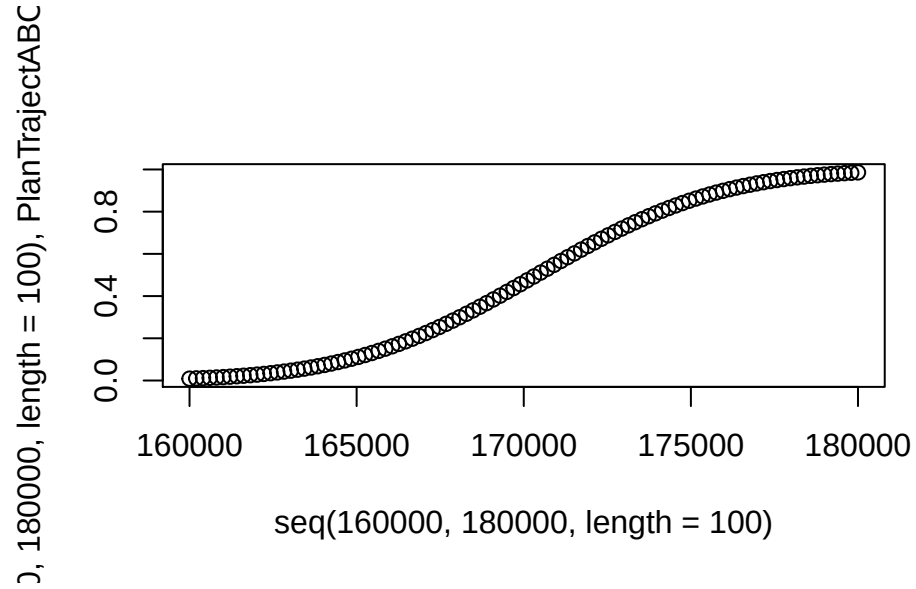

```
# Trajectory and corridor plotting
plot(PlanTrajectABC,col=1:4,xlim=c(1,4),ylim=c(0,200000),xaxt='n',
xlab = "Future Quarters", ylab = "Cumulated sales")
axis(1, at = seq(1, 4, by = 1))
lines(PlanTrajectABC+PlanVolaABC)
lines(PlanTrajectABC-PlanVolaABC)
```



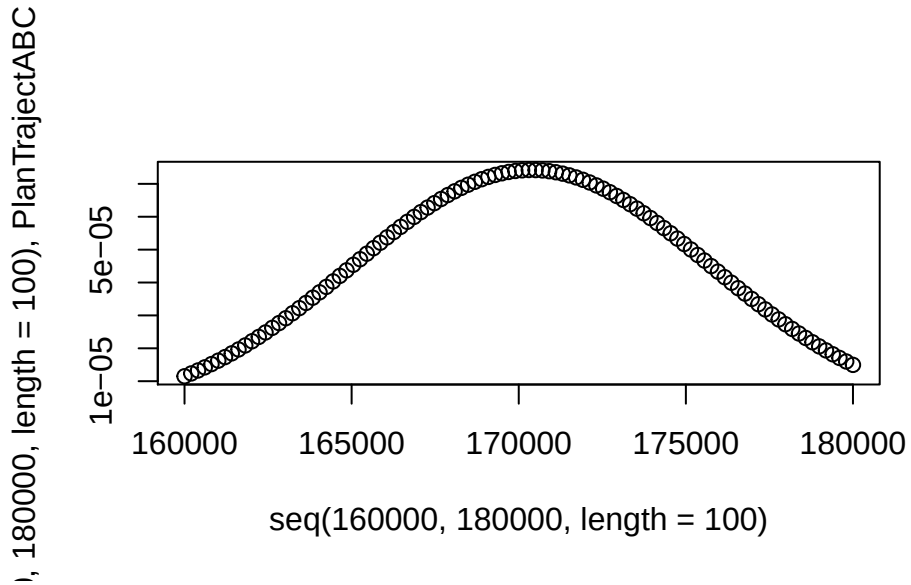

```
# Density function plotting
plot(seq(160000, 180000, length=100), dnorm(seq(160000, 180000,
length=100),PlanTrajectABC[4],PlanVolaABC[4]), type="l",
xlab="Sales", ylab="Density", main="plot()") # Density plot ???ABC
```



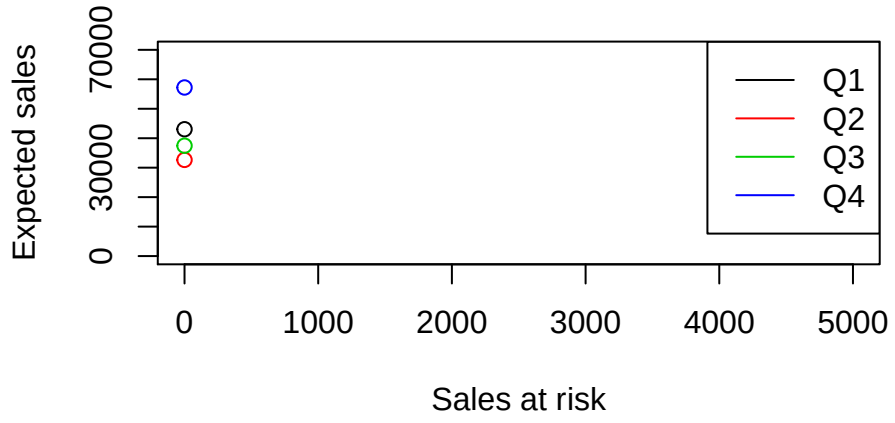

```
 plot(seq(160000, 180000, length=100), pnorm(seq(160000, 180000,
length=100),PlanTrajectABC[4],PlanVolaABC[4])) # Cum.Dens.plot ???ABC
```




```
 plot(seq(160000, 180000, length=100), dnorm(seq(160000, 180000,
length=100),PlanTrajectABC[4],PlanVolaABC1[4])) # Density plot ???ABC1
```




```
# Mean (location) and volatility (dispersion) parameter plotting
 plot(CV_xSQ01ABC1.m,E_xSQ01ABC.m,col=1:4,ylim=c(0,70000),
 xlim=c(0,5000), xlab = "Sales at risk", ylab = "Expected sales")
 legend("topright", legend=c("Q1","Q2","Q3","Q4"),col=1:4,lty=1)
```



Task 16

Year-end forecasting and probabilistic control over time

```
 AA_xSQ01ABC.m<-matrix(0,1,4) # Accumulated Actual
 RE_xSQ01ABC.m<-matrix(0,1,4) # Remaining Expectations
 CF_xSQ01ABC.m<-matrix(0,1,4) # Conditional Forecast
 BFD_xSQ01ABC.m<-matrix(0,1,4) # Budget-Forecast Deviation
 BFDra_xSQ01ABC.m<-matrix(0,1,4) # BFD risk-adjusted
 RV_xSQ01ABC.m<-matrix(0,1,4) # Remaining Volatility
 PrBFDra_xSQ01ABC.m<-matrix(0,1,4) # Probability BFDra
 (AA_xSQ01ABC.m[1]<-c(42234.86)) #Y-end forecast & control after Q1
## [1] 42234.86
 (RE_xSQ01ABC.m[1]<- sum(E_xSQ01ABC.m[2:4]))
## [1] 127330.5
 (CF_xSQ01ABC.m[1]<-AA_xSQ01ABC.m[1]+RE_xSQ01ABC.m[1])
```

```
## [1] 169565.4
 (BFD_xSQ01ABC.m[1]<-CF_xSQ01ABC.m[1]-BudgetABC)
## [1] -814.64
 (RV_xSQ01ABC.m[1]<- sqrt(sum(V_xSQ01ABC.m[2:4]^2)))
## [1] 3465.514
 (BFDra_xSQ01ABC.m[1]<-BFD_xSQ01ABC.m[1]/RV_xSQ01ABC.m[1])
## [1] -0.2350704
 (PrBFDra_xSQ01ABC.m[1]<-pnorm(BFDra_xSQ01ABC.m[1]))
## [1] 0.407077
 (AA_xSQ01ABC.m[2]<-AA_xSQ01ABC.m[1]+c(33852.96))#Y-e.f.&ctr.after Q2
## [1] 76087.82
 (RE_xSQ01ABC.m[2]<- sum(E_xSQ01ABC.m[3:4]))
## [1] 94680.5
 (CF_xSQ01ABC.m[2]<-AA_xSQ01ABC.m[2]+RE_xSQ01ABC.m[2])
## [1] 170768.3
 (BFD_xSQ01ABC.m[2]<-CF_xSQ01ABC.m[2]-BudgetABC)
## [1] 388.32
 (RV_xSQ01ABC.m[2]<- sqrt(sum(V_xSQ01ABC.m[3:4]^2)))
## [1] 3300.119
 (BFDra_xSQ01ABC.m[2]<-BFD_xSQ01ABC.m[2]/RV_xSQ01ABC.m[2])
## [1] 0.1176685
 (PrBFDra_xSQ01ABC.m[2]<-pnorm(BFDra_xSQ01ABC.m[2]))
## [1] 0.5468348
```