R data validation course
Part 3

by Statistics Iceland, for/with our colleagues from:
Armenia, Azerbaijan, Belarus, Georgia, Kazakhstan, Kyrgyzstan,
Moldova, Tajikistan, Turkmenistan, Ukraine, Uzbekistan

March 2021

Organization

- Lectures (on 22, 24, **26** of March 2021, from 9:00 GMT to 12:00 GMT, via interprefy platform)
- Practical sessions (23 and 25 of March 2021, from 9:00 GMT to 11:00 GMT, via TEAMS):
  - students use, adapt and report, based on two template Rmd-files (provided by the instructors, one for each sesssion) and individual datasets,
    similar to the examples shown at:
    https://github.com/violetacln/revaliew/blob/master/rmd_reports/
- useful sharing at: https://github.com/violetacln/learnRval

R-packages frequently used:

```
# install.packages("tidyverse",
#                  "validate", "validatetools",
#                  "errorlocate", "dcmodify", "simputation",
#                  "DataExplorer", "funModeling",
#                   "tseries", "forecast", "tsfeatures",          #new
#                   "devtools"  , "arules", "EditImputeCont",     #new
#                   "vsgoftest",  "factoextra", "cluster",        #new
#                  "caret", "tidymodels",  ## optional
#                  dependencies=TRUE)
```

Previously

Part1: Reproducible, reviewed, improved

data processing and analysis
with validation, as stages of GSBPM, using R

Part 2: Workflow for *error detection*

based on: expert rules and statistical analysis (advanced validation)

Part 3:

Workflow for *analysis and dissemination of results* with validation, as stages of GSBPM, using *R*

3.1. Output validation with:

- expert rules and machine learning
- comparing data sets

3.2. Error correction methods

- main types: imputations, modifier rules and manual correction
- machine learning for modifier rule discovery
- simultaneous error detection and correction (Bayesian methods): short description

3.3. Dynamic/static reports

3.1. Output validation with:

- expert rules and machine learning
- comparing data sets

3.1.1. expert rules and machine learning, and not only!

- Expert rules: re-applied *after correcting* all errors (see Part 2 and 3.2.)

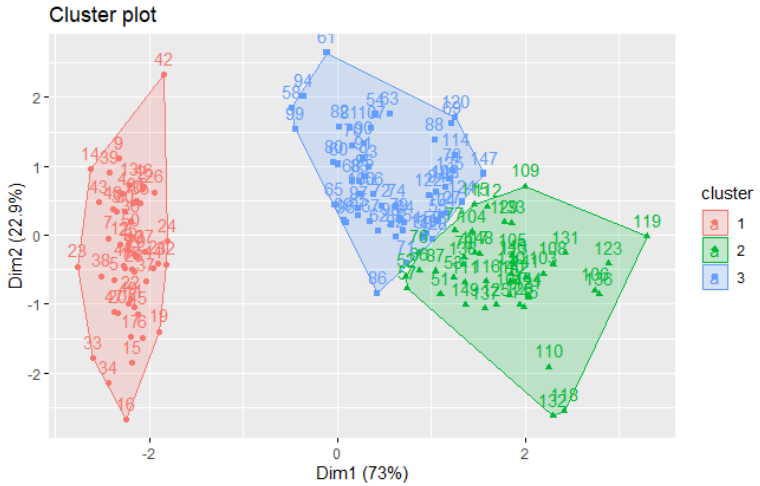
- Identify clusters in the data
- Verify main characteristics of time series
- Identify anomalies in time series
- Verify model performance if any modeling used for output

Identify clusters in the data

```
df <- iris[,1:4]
# find optimal number of clusters
  dff <- scale(df)
  factoextra::fviz_nbclust(dff, kmeans, method = "gap_stat")
# compute and visualise
  set.seed(123)
  km.res <- kmeans(dff, 3, nstart = 25)
# visualize
  factoextra::fviz_cluster(km.res, data = dff,
               ellipse.type = "convex",
               palette = "jco",
               repel = TRUE,
               ggtheme = ggplot2::theme_minimal())
```

... and continue

```
# compare with PAM clustering
  # Compute PAM
  pam.res <- cluster::pam(dff, 3)
  # Visualize
  factoextra::fviz_cluster(pam.res)
```


Cluster plot
Dim2 (22.9%)
Dim1 (73%)
cluster
1
2
3

Verify main characteristics of time series

- uni-variate time series diagnostics

```
# example
tsuniv = ts(rnorm(1000))
 rlist_univ <- list(
        tseries::jarque.bera.test(tsuniv),
        tseries::kpss.test(tsuniv),
        tseries::kpss.test(tsuniv, null = "Trend"),
        tseries::adf.test(tsuniv),
        forecast::Acf(tsuniv),
        forecast::Pacf(tsuniv),
        tsfeatures::tsfeatures(tsuniv)
        )
rlist_univ
```

- multi-variate time series characteristics

see https://robjhyndman.com/hyndsight/tscharacteristics/

```
# example
tsmultiv <- ts(matrix(rnorm(3000),ncol=100),freq=4)
 # all important characteristics of multivar ts in:
  all_characteristics <- tsfeatures::tsfeatures(tsmultiv)
  all_characteristics
```

These include:

frequency, nperiods, seasonal period, trend, spike, linearity, curvature, auto-correlation seasonal strength, peak, trough, entropy, variations in some of these

Identify anomalies in time series

First, install a new package from github

```
##  ***IT IS WORTH IT***
# devtools::install_github("robjhyndman/anomalous")

#### does it work without the Rtools?, see comments at:
### https://cran.r-project.org/bin/windows/Rtools/
### for me, yes

tsmultiv <- ts(matrix(rnorm(3000),ncol=100),freq=4)
library(anomalous)
y <- anomalous::tsmeasures(tsmultiv)
anomalous::biplot.features(y)
anomalous::anomaly(y)
```

Verify model performance if modeling used for output

- For glm models (install lmtest first!)

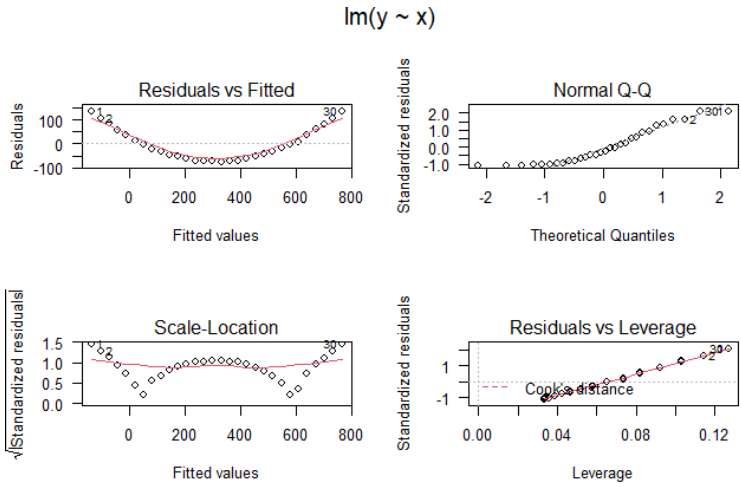
```
x <- c(1:30); y <- x^2+rnorm(30,0,2); model_ex = lm(y~x);
model_a <- model_ex

# model is regression type or glm
  raintesting <- lmtest::raintest(model_a, order.by="mahalanobis")
  print(raintesting)

### J.M. Utts (1982), The Rainbow Test for Lack of Fit in Regression.
### Communications in Statistics -- Theory and Methods 11,2801--2815.
```

```r
#-Durbin-Watson-Test on autocorrelation of disturbances
durbin_watson_testing <- lmtest::dwtest(model_a)

hist(residuals(model_a))

#normality test for residuals of models
jarque_bera_testing <- tseries::jarque.bera.test(residuals(model_a))
print(jarque_bera_testing)
```

```r
##a GOF test based on an entropy measure, see
  ##Justine Lequesne, Philippe Regnault. vsgoftest:
  ##An R Package for Goodness-of-Fit Testing Based on
  ##Kullback-Leibler Divergence, 2018
library("vsgoftest")
vs_goodness_of_fit_testing <- vsgoftest::vs.test(x=residuals(model_a),
                                                 densfun="dnorm")
print(vs_goodness_of_fit_testing)
```

For time series models

```
#portmanteau tests for model residuals, ts models
portmanteau_testing_ts_resid <- Box.test (residuals(model_a),
                                          lag = 1, type="Ljung")
print(portmanteau_testing_ts_resid)
# visualization, model diagnostics
opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
plot(model_a, las = 1)
# visualization, residuals diagnostics
lag.plot(residuals(model_a), lags=12, do.lines=FALSE)
```


lm(y ~ x)
Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

```r
kpss_testing <- tseries::kpss.test(residuals(model_a))
print(kpss_testing)  #------- test of stationarity

adf_testing <- tseries::adf.test(residuals(model_a))
print(adf_testing)  #-------- test of non-stationarity

#visualization: (auto-) correlation
par(mfrow=c(2,1))
#  estimate of autocorr. fct, uni/multi-variate
acf(residuals(model_a))
# estimate of partial autocorr. fct, uni/multi-variate
pacf(residuals(model_a))
```

3.1.2. comparing data sets

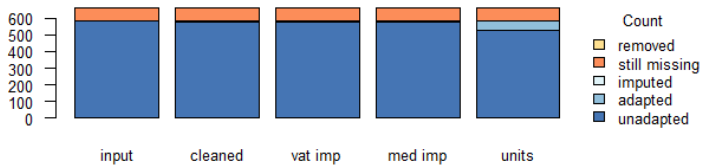
See:
https://cran.r-project.org/web/packages/validate/vignettes/cookbook.html#9_Comparing_data_sets

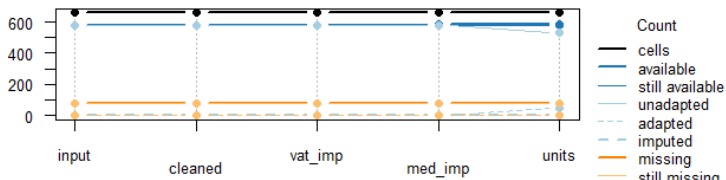
- *cells*(*df*1, *df*2, ..., *compare* = *c*("*to_first*", "*sequential*"))
  - input: many data frames with same dimensions
  - output: array of class *cellComparison*
  - *validate* has two plot methods for this output: *barplot*() and *plot*()
- *compare*(*v*, *df*1, *df*2, ..., *how* = *c*("*to_first*", "*sequential*"))
  - *barplot*() and *plot*() methods available also for the output

Example, *from same handbook*

```
library(validate)
data(SBS2000)
original <- SBS2000
version2 <- original
# make a little change
version2$other.rev <- abs(version2$other.rev)
cells(input = original, cleaned = version2, compare="sequential")
# more versions of data
version3 <- version2
version3$turnover[is.na(version3$turnover)] <-
                        version3$vat[is.na(version3$turnover)]
```

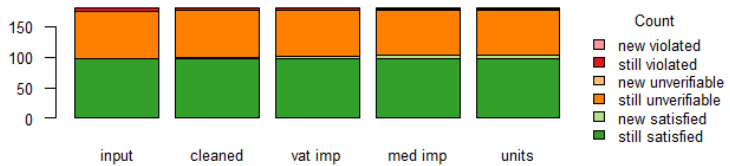
```r
version4 <- version3
version4$turnover[is.na(version4$turnover)] <-
                      median(version4$turnover, na.rm=TRUE)
# from kEUR to EUR
version5 <- version4
version5$staff.costs <- version5$staff.costs * 1000
# check comparisons by cell()
out <- cells(input = original
           , cleaned = version2
           , vat_imp = version3
           , med_imp = version4
           , units   = version5)
par(mfrow=c(2,1))
barplot(out)
plot(out)
```

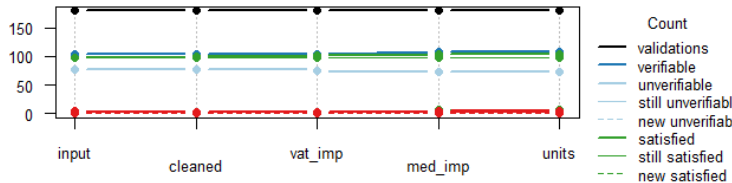

600
500
400
300
200
100
0
input
cleaned
vat imp
med imp
units
Count
removed
still missing
imputed
adapted
unadapted


600
400
200
0
input
cleaned
vat_imp
med_imp
units
Count
cells
available
still available
unadapted
adapted
imputed
missing
still missing

Example of using the *compare*() function, *from same handbook*

```
vrules <- validator(other.rev >= 0, turnover >= 0
                  , turnover + other.rev == total.rev)

comparison <- compare(vrules
                    , input = original, cleaned = version2
                    , vat_imp = version3, med_imp = version4
                    , units   = version5)
comparison
par(mfrow=c(2,1))
barplot(comparison)
plot(comparison)
```


150
100
50
0
input
cleaned
vat imp
med imp
units
Count
new violated
still violated
new unverifiable
still unverifiable
new satisfied
still satisfied


150
100
50
0
input
cleaned
vat_imp
med_imp
units
Count
validations
verifiable
unverifiable
still unverifiable
new unverifiable
satisfied
still satisfied
new satisfied

3.2. Error correction methods

- main types: imputations, modifier rules and manual correction
- machine learning for modifier rule discovery
- simultaneous error detection and correction (Bayesian methods): short description

3.2.1. main types: imputations, modifier rules and manual correction

manual corrections - discusssion:

- when?
- how much?
- practical and theoretical criteria!

modifier rules

- Deterministic type of corrections
- May treat modifying rules exactly like the validation rules:
  - read/write, add metadata, manipulation
- Representation of domain knowledge (or found by *ML* !, see 3.2.2)

Example: when we have domain knowledge

```
#install.packages("dcmodify")
library(dcmodify)
m <- modifier(if (staff.costs < 0) staff.costs <- (-1) * staff.costs)
modified <- modify(retailers, m)
head(modified[3:7], 3)
```

imputations, but first:

Which fields to impute?!

- *errorlocation* package:
  - need: a record may break multiple rules
  - criterion of this package:
    - `find the smallest number of fields to be replaced`
  - most useful functions:
    - `$locate\_errors(df, vrules)$`
    - `$replace\_errors(df, vrules)$ (with NA values)`

Example addapted from https://github.com/data-cleaning/errorlocate

```
library(errorlocate)
rules <- validator( profit == turnover - cost, cost >= 0.6 * turnover
                  , turnover >= 0, cost >= 0)
data <- data.frame(profit=750, cost=125, turnover=200)
  error_locations <- locate_errors(data, rules)
  values(error_locations)
  summary(error_locations)
data_marked_errors <- replace_errors(data, rules)
# faulty data was replaced with NA
print(data_marked_errors)
er <- errors_removed(data_marked_errors)
print(er); summary(er); er$errors
sum(is.na(data))
sum(is.na(data_marked_errors))
```

Example of imputation functions and why *simputations* package

https://cran.r-project.org/web/packages/simputation/vignettes/intro.html

- why: unified and simple calls *impute_modelName*(*data*, *formula*, *options*)

```
library(simputation)
dat <- iris
dat[1:3,1] <- dat[3:7,2] <- dat[8:10,5] <- NA
head(dat,10)
da1 <- impute_lm(dat, Sepal.Length ~ Sepal.Width + Species)
head(da1,3)
da2 <- impute_median(da1, Sepal.Length ~ Species)
head(da2,3)
da3 <- impute_cart(da2, Species ~ .) #decision tree
head(da3,10)
#chaining all these methods is possible, with %>% of magrittr - package
```

- Note1: using the VIM package

- Note2: using MICE, mi, Hmisc, Amelia, missForest
- more packages from the suite: lumberjack, rspa, deductive, ...

3.2.2. machine learning for modifier rule discovery

```
df =datasets::iris #but any other is good
#(citation("arules"))
library(arules)
tdata <- as(df, "transactions")
#method 1 of clustering: eclat algorithm
eclat_res <- inspect(eclat(tdata,
                           parameter = list(supp=0.07, maxlen=15)))
eclat_plot <- itemFrequencyPlot(tdata, topN=10,
                          type="absolute", main="item freguency")
summary_data <- summary(tdata)
eclat_summary <- summary(eclat(tdata,
                          parameter = list(supp=0.07, maxlen=15)))
```

... continue

```
#method 2 of clustering: apriori algorithm
rules <- apriori(tdata)
apriori_summary <- summary(rules)
apriori_res <- inspect(rules)
```

Discuss!

3.2.3. simultaneous error detection and correction (Bayesian methods): short description

https://cran.r-project.org/web/packages/EditImputeCont/index.html

Paper: Kim et al. (2015) doi:10.1080/01621459.2015.1040881

"... Bayesian hierarchical model that includes:

- a flexible joint probability model for the underlying true values of the data with support only on the set of values that satisfy all editing constraints

- a model for latent indicators of the variables that are in error, and

- a model for the reported responses for variables in error."

Example, from
https://rdrr.io/cran/EditImputeCont/man/EditImputeCont-package.html

```
library(EditImputeCont)
## read the toy example data, which has two ratio edits and a balance edit
data(SimpleEx)
data1 = readData(Y.original=SimpleEx$D.obs, ratio=SimpleEx$Ratio.edit,
    range=NULL, balance=SimpleEx$Balance.edit)

## create and initialize the model with 15 DP mixture components
 model1 = createModel(data.obj=data1, K=15)
## Run an iteration of MCMC
model1$Iterate()
dim(model1$Y.edited)
## [1]   1000    4
# Edit-imputed datasets of n=1000 records with p=4 variables
## Please see the example in the demo folder for more details
```

3.3. Dynamic/static reports with RStudio and RMarkdown

hackernoon.com/30-jokes-only-programmers-will-get-a901e1cea549

R_related | Imported | validation_related | gpg_rel | git_rel | popProj

s Only Programmers Will Get by @pavel.malos

9. Where did you two meet?

- Combines:

images, links, tables, LaTeX, and code (R, python, sql, . . .)

- Version control:

with Projects, Github

- Output as:

html, pdf, markdown, Microsoft Word, html-vignettes, web-sites

- Our own take-away examples:

the *RvalPractice1.Rmd* and *RvalPractice2.Rmd* which

you already have used to produce analysis and reports!

- Useful links:

https://rmarkdown.rstudio.com/lesson-1.html

https://rpubs.com/marschmi/RMarkdown

- Interactive! and shiny!
  https://shiny.rstudio.com/gallery/india-blood-banks.html
  with code at
  https://github.com/atmajitg/bloodbanks

Thank you!

Discussion and feedback!