COrruption Risk Indicators in Emergency

Co-funded by the European Union

Statistical codes and user guide developed in R for

"Data quality check, cleaning and pre-processing"

WP3- Deliverable 3.2

Michela Gnaldi – Università di Perugia

Simone Del Sarto – Università di Perugia

Niccolò Salvini – Università Cattolica del Sacro Cuore in Roma

Maria Giovanna Ranalli – Università di Perugia

This document includes two parts: a first part devoted to the R-codes for the reading of the data and a second part including the User Guide, to help the user to apply autonomously the R-codes.

Grant Agreement number: 101038790 — CO.R.E — ISFP-2020-AG-CORRUPT

This document was funded by the European Union's Internal Security Fund — Police. The content of this document represents the views of the author only and is his/her sole responsibility. The European Commission does not accept any responsibility for use that may be made of the information it contains.

1. R-CODES

- **check_dates.R**

```
check_dates <- function(data, call=NULL, deadline=NULL, award=NULL,
start=NULL, end=NULL) {

  #### Function arguments ----
  # - data: dataframe to be checked
  # - call: [character] name of variable about the call for tender
publication date
  # (of type "Date", format=yyyy-mm-dd)
  # - deadline: [character] name of variable about the bid submission
deadline
  # (of type "Date", format=yyyy-mm-dd)
  # - award:[character] name of variable about the award date
  # (of type "Date", format=yyyy-mm-dd)
  # - start: [character] name of variable about the contract start
  # (of type "Date", format=yyyy-mm-dd)
  # - end: [character] name of variable about the contract end
(completion)
  # (of type "Date", format=yyyy-mm-dd)

  #### Function code ----
  # required R packages
  library(lubridate)

  # check the nature of the dates
  if (!is.Date(data[[call]])) stop("\nPlease, check the format of 'call':
it must be 'Date'")
  if (!is.Date(data[[deadline]])) stop("Please, check the format of
'deadline': it must be 'Date'")
  if (!is.Date(data[[award]])) stop("Please, check the format of 'award':
it must be 'Date'")
  if (!is.Date(data[[start]])) stop("Please, check the format of 'start':
it must be 'Date'")
  if (!is.Date(data[[end]])) stop("Please, check the format of 'end': it
must be 'Date'")

  # new variables
  # call >= deadline
  data$datecheck_call_deadline <- 1*(data[[call]] >= data[[deadline]])
  # deadline >= award
  data$datecheck_deadline_award <- 1*(data[[deadline]] >= data[[award]])
  # award >= start
  data$datecheck_award_start <- 1*(data[[award]] >= data[[start]])
  # start >= end
  data$datecheck_start_end <- 1*(data[[start]] > data[[end]])

  toprint <- data %>%
    filter(datecheck_call_deadline == 1 | datecheck_deadline_award == 1 |
             datecheck_award_start == 1 | datecheck_start_end == 1) %>%
    select(cig, call, deadline, award, start, end,
```

```
            datecheck_call_deadline, datecheck_deadline_award,
            datecheck_award_start, datecheck_start_end)

  toprint %>% View

  return(data)

}
```

- **check_amounts.R**

```
check_amounts <- function(data, opening, award, sums_paid, pcut=c(0.05,
0.95)) {

  #### Function arguments ----
  # data: dataframe to be checked
  # opening: [character] name of variable about the opening bid amount
  # award: [character] name of variable about the award amount
  # sums_paid: [character] name of variable about the final amount paid
by the administration

  #### Function code ----

  # new variables
  # award vs. opening
  data$ratio_award_opening <- data[[award]]/data[[opening]]
  q1 <- quantile(data$ratio_award_opening, probs=pcut, na.rm=TRUE)
  # check whether award > opening (ratio > 1) or extremely high distance
between
  # award and opening
  data$amountcheck_award_opening <- 1*(data$ratio_award_opening > 1 |
                                       data$ratio_award_opening <
q1[1])
  # sums_paid vs. award
  data$ratio_sumspaid_award <- data[[sums_paid]]/data[[award]]
  q2 <- quantile(data$ratio_sumspaid_award, probs=pcut, na.rm=TRUE)
  data$amountcheck_sumspaid_award <- 1*(data$ratio_sumspaid_award < q2[1]
|
                                         data$ratio_sumspaid_award >
q2[2])

  # print(q1)
  # print(q2)

  toprint <- data %>%
    filter(!is.na(amountcheck_award_opening) &
!is.na(amountcheck_sumspaid_award)) %>%
    filter(amountcheck_award_opening == 1 | amountcheck_sumspaid_award ==
1) %>%
    select(cig, opening, award, sums_paid,
           ratio_award_opening, amountcheck_award_opening,
           ratio_sumspaid_award, amountcheck_sumspaid_award)

  toprint %>% View

  return(data)

}
```

2. User guide

Introduction

This guide supports the R functions developed for checking and cleaning some of the information contained in the data coming from the Italian National Database of Public Procurement (BDNCP). Since the information related to **dates** and **amount** often contains errors in BDNCP, data quality checks are carried out with respect to two variables.

Function 'check_dates'

Five dates are available in the data, related to the main stages of the procurement process:

1. publication of the call for tenders (*call*);
2. deadline for submitting a bid (*deadline*);
3. award notice (*award*);
4. contract start (*start*);
5. contract end (*end*).

These stages are consecutive, that is,

call < deadline < award < start < end

This function carries out four checks and adds four dummy variables to the data (one for each check), equal to 1 if the check fails and 0 otherwise. Specifically, the function checks whether:

6. call >= deadline (related dummy variable *datecheck_call_deadline*);
7. deadline >= award (*datecheck_deadline_award*);
8. award >= start (*datecheck_award_start*);
9. start >= end (*datecheck_start_end*).

The function requires the following arguments:

- *data*: the dataframe to be checked
- *call*: [character] name of variable about the call for tender publication date (of type 'Date', format=yyyy-mm-dd)
- *deadline*: [character] name of variable about the bid submission deadline (of type 'Date', format=yyyy-mm-dd)
- *award*:[character] name of variable about the award date (of type "Date", format=yyyy-mm-dd)
- *start*: [character] name of variable about the contract start(of type "Date", format=yyyy-mm-dd)
- *end*: [character] name of variable about the contract end (of type "Date", format=yyyy-mm-dd)

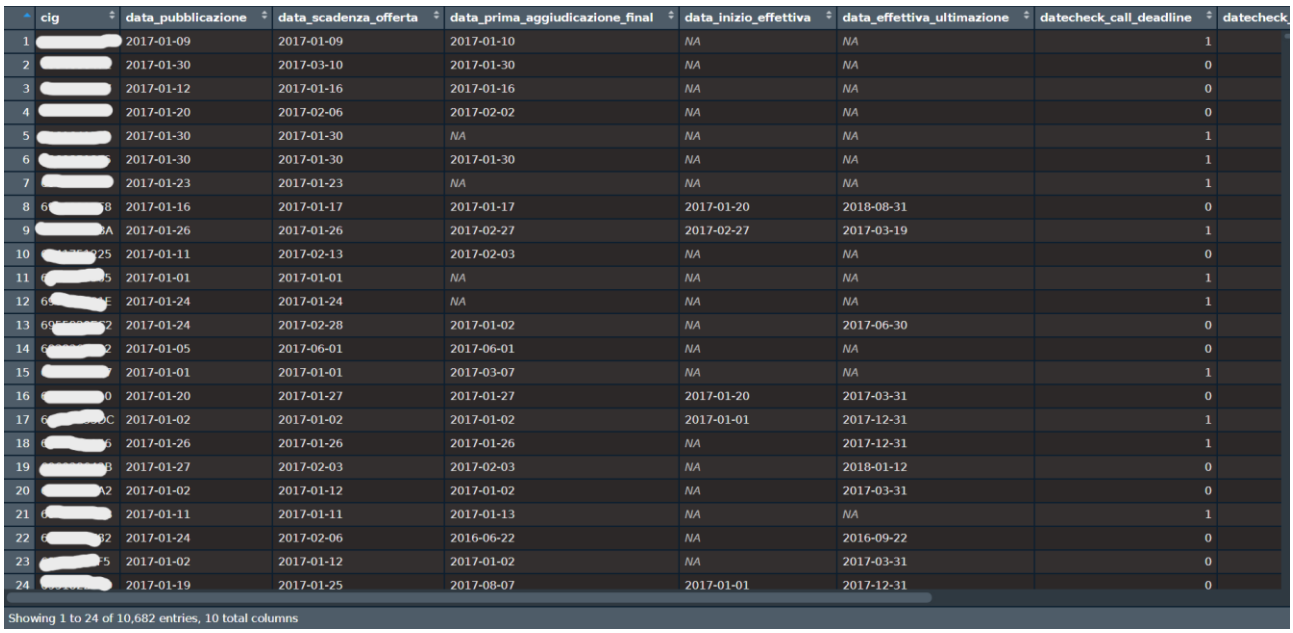
Here is an example on the dataframe 'data_test':

```
source("check_dates.R")
load("data_test.RData")
CALL <- "data_pubblicazione"
DEADLINE <- "data_scadenza_offerta"
AWARD <- "data_prima_aggiudicazione_final"
START <- "data_inizio_effettiva"
END <- "data_effettiva_ultimazione"
data_test2 <- check_dates(data=data_test,
                          call=CALL,
                          deadline=DEADLINE,
                          award=AWARD,
                          start=START,
                          end=END)

## Error in check_dates(data = data_test, call = CALL, deadline = DEADLINE, :
## Please, check the format of 'call': it must be 'Date'
```

In the example, the function returns an error, as the format of some variables is not 'Date' but 'character', as can be noticed below. However, after converting the involved variables in the right format, the function can run:

```
str(data_test[[CALL]])

##  chr [1:50000] "2017-01-02" "2017-01-05" "2017-01-09" "2017-01-30" ...

str(data_test[[DEADLINE]])

##  chr [1:50000] "2017-01-30" "2017-01-15" "2017-01-09" "2017-03-10" ...

data_test[[CALL]] <- as.Date(data_test[[CALL]])
data_test[[DEADLINE]] <- as.Date(data_test[[DEADLINE]])
data_test2 <- check_dates(data=data_test,
                          call=CALL,
                          deadline=DEADLINE,
                          award=AWARD,
                          start=START,
                          end=END)
mean(data_test2$datecheck_call_deadline, na.rm=TRUE)

## [1] 0.05829956

mean(data_test2$datecheck_deadline_award, na.rm=TRUE)

## [1] 0.1795498

mean(data_test2$datecheck_award_start, na.rm=TRUE)

## [1] 0.2993506

mean(data_test2$datecheck_start_end, na.rm=TRUE)

## [1] 0.004457831
```

In addition to creating the four dummy variables, the RStudio dataframe viewer is opened and shows a subset of 'data' for which at least one check failed, containing the contract identifier ('cig') and the related five dates and the four dummy variables about the checks.



	cig	data_pubblicazione	data_scadenza_offerta	data_prima_aggiudicazione_final	data_inizio_effettiva	data_effettiva_ultimazione	datecheck_call_deadline	datecheck
1	[redacted]	2017-01-09	2017-01-09	2017-01-10	NA	NA	1	
2	[redacted]	2017-01-30	2017-03-10	2017-01-30	NA	NA	0	
3	[redacted]	2017-01-12	2017-01-16	2017-01-16	NA	NA	0	
4	[redacted]	2017-01-20	2017-02-06	2017-02-02	NA	NA	0	
5	[redacted]	2017-01-30	2017-01-30	NA	NA	NA	1	
6	[redacted]	2017-01-30	2017-01-30	2017-01-30	NA	NA	1	
7	[redacted]	2017-01-23	2017-01-23	NA	NA	NA	1	
8	[redacted]	2017-01-16	2017-01-17	2017-01-17	2017-01-20	2018-08-31	0	
9	[redacted]	2017-01-26	2017-01-26	2017-02-27	2017-02-27	2017-03-19	1	
10	[redacted]	2017-01-11	2017-02-13	2017-02-03	NA	NA	0	
11	[redacted]	2017-01-01	2017-01-01	NA	NA	NA	1	
12	[redacted]	2017-01-24	2017-01-24	NA	NA	NA	1	
13	[redacted]	2017-01-24	2017-02-28	2017-01-02	NA	2017-06-30	0	
14	[redacted]	2017-01-05	2017-06-01	2017-06-01	NA	NA	0	
15	[redacted]	2017-01-01	2017-01-01	2017-03-07	NA	NA	1	
16	[redacted]	2017-01-20	2017-01-27	2017-01-27	2017-01-20	2017-03-31	0	
17	[redacted]	2017-01-02	2017-01-02	2017-01-02	2017-01-01	2017-12-31	1	
18	[redacted]	2017-01-26	2017-01-26	2017-01-26	NA	2017-12-31	1	
19	[redacted]	2017-01-27	2017-02-03	2017-02-03	NA	2018-01-12	0	
20	[redacted]	2017-01-02	2017-01-12	2017-01-02	NA	2017-03-31	0	
21	[redacted]	2017-01-11	2017-01-11	2017-01-13	NA	NA	1	
22	[redacted]	2017-01-24	2017-02-06	2016-06-22	NA	2016-09-22	0	
23	[redacted]	2017-01-02	2017-01-12	2017-01-02	NA	2017-03-31	0	
24	[redacted]	2017-01-19	2017-01-25	2017-08-07	2017-01-01	2017-12-31	0	

Showing 1 to 24 of 10,682 entries, 10 total columns

Function 'check_amounts'

Three amounts are available in BDNCP, related to the:

10. opening bid (*opening*);
11. award amount (*award*);
12. final amount paid by the administration (*sums_paid*).

Checks on these amounts are carried out by computing the following ratios:

- award/opening;
- sums_paid/award.

Accordingly, checks are performed as follows:

- *award vs. opening*. The former should not be greater than the latter (except for specific cases); the former should not be **much lower** (defined through a specific quantile of the distribution of the corresponding ratio) than the latter. Hence, the checks are
  - award/opening > 1;
  - award/opening < q, where q is a specific quantile (in the lower tail) of the distribution of award/opening.
- *sums_paid vs. award*. These two amounts should not differ **too much** (again, defined through specific quantiles of the distribution of the corresponding ratio), hence the check is: sums_paid/award < q_l or sums_paid/award > q_u, where q_l and q_u are suitable quantiles of of the distribution of sums_paid/award (in the lower and upper tail, respectively).

In addition to the above ratios, this function also adds a dummy variable for each check, equal to 1 if the check fails and 0 otherwise.

The function requires the following arguments:

- *data*: the dataframe to be checked
- *opening*: [character] name of variable measuring the opening bid amount
- *award*: [character] name of variable measuring the award amount
- *sums_paid*:[character] name of variable measuring the final amount paid by the administration
- *pcut*: [numeric vector] two quantiles (in terms of probabilities in the tails) for identifying extreme cases (e.g., 0.05 and 0.95)

Here is an example on the dataframe 'data_test':

```
source("check_amounts.R")
load("data_test.RData")
OPENING <- "importo_lotto"
AWARD <- "importo_aggiudicazione"
SUMS_PAID <- "imp_finale"
data_test2 <- check_amounts(data=data_test,
                            opening=OPENING,
```

```
                              award=AWARD,
                              sums_paid=SUMS_PAID)
summary(data_test2$ratio_award_opening)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##     0.000     0.736     0.918     1.282     1.000 6954.867     13164

summary(data_test2$ratio_sumspaid_award)

##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.00    0.95    1.00     Inf    1.04     Inf   41345

mean(data_test2$amountcheck_award_opening, na.rm=TRUE)

## [1] 0.08032903

mean(data_test2$amountcheck_sumspaid_award, na.rm=TRUE)

## [1] 0.05002889
```

In addition to creating the four variables above, the RStudio dataframe viewer is opened and shows a subset of 'data' for which at least one check failed, containing the contract identifier ('cig'), the related amounts and the four variables just created.

	cig	importo_lotto	importo_aggiudicazione	imp_finale	ratio_award_opening	amountcheck_award_opening	ratio_sumspaid_award	amountcheck_sumspaid_award
1	[redacted]	100000.00	88480.000	149661.13	0.8848000000	0	1.69146847	1
2	[redacted]	120000.00	122986.690	122836.73	1.0248890833	1	0.99878068	0
3	[redacted]	291000.00	291000.000	460493.70	1.0000000000	0	1.58245258	1
4	[redacted]	83500.00	77807.000	172290.32	0.9318203593	0	2.21432930	1
5	[redacted]	48808.00	49811.810	0.00	1.0205665055	1	0.00000000	0
6	[redacted]	940847.66	799149.423	1360664.81	0.8493930069	0	1.70264130	1
7	[redacted]	325000.00	295714.450	412227.65	0.9098906154	0	1.39400577	1
8	[redacted]	53417.03	42936.824	62326.03	0.8038040303	0	1.45157523	1
9	[redacted]	40261.00	37712.519	87995.83	0.9367010010	0	2.33333207	1
10	[redacted]	60000.00	83390.000	0.00	1.3898333333	1	0.00000000	0
11	[redacted]	350000.00	294814.670	441851.31	0.8423276286	0	1.49874262	1
12	[redacted]	150782.98	170561.020	175742.73	1.1311689157	1	1.03038039	0
13	[redacted]	75000.00	56926.000	100848.49	0.7590133333	0	1.77157169	1
14	[redacted]	61703.48	43480.600	61228.85	0.7046701418	0	1.40818779	1
15	[redacted]	91000.00	67839.260	109780.03	0.7454863736	0	1.61823743	1
16	[redacted]	208000.00	77000.000	0.00	0.3701923077	1	0.00000000	0
17	[redacted]	59556.47	59556.490	59556.49	1.0000003358	1	1.00000000	0
18	[redacted]	43909.15	32119.543	58221.21	0.7314999949	0	1.81264129	1
19	[redacted]	2052000.00	904750.840	903594.18	0.4409117154	1	0.99872157	0
20	[redacted]	82000.00	82000.000	111491.00	1.0000000000	0	1.35964634	1
21	[redacted]	60000.00	63800.000	39950.50	1.0633333333	1	0.62618339	0
22	[redacted]	73600.00	50673.647	73448.34	0.6885006386	0	1.44943852	1
23	[redacted]	49366.00	54074.000	49326.66	1.0953692825	1	0.91220661	0
24	[redacted]	47664.76	47664.760	88491.75	1.0000000000	0	1.85654454	1

Showing 1 to 24 of 791 entries, 8 total columns