Exploration with python

August 21, 2018

1 CDV study - data analysis

In [1]:
```
from pathlib import Path
import pandas as pd
import numpy as np
from datetime import datetime
import time
import matplotlib.pyplot as plt
%matplotlib inline
#%pylab inline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score, GridSearchCV
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
```

In [2]:
```
path_project = Path.home() / Path('Google Drive/Felix')
path_data = path_project / Path("data")
```

In [3]:
```
# loading cdv data
file = path_data / Path("felix.csv")
with Path.open(file, 'rb') as fp:
    cdv = pd.read_csv(fp,  encoding='cp1252',low_memory=False)
```

In [4]:
```
cdv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11131 entries, 0 to 11130
Columns: 354 entries, INTER6 to an_nais
dtypes: float64(47), int64(15), object(292)
memory usage: 30.1+ MB
```

In [5]:
```
# loadind cdv data without format
file = path_data / Path("felix_ssfmt.csv")
with Path.open(file, 'rb') as fp:
    cdv_ssfmt = pd.read_csv(fp,  encoding='cp1252',low_memory=False)
```

```
In [6]: cdv_ssfmt.shape

Out[6]: (11131, 354)
```

1.1 1) Dataset Size and missing values analysis

1.1.1 a) First anlysis regarless of the year of teh study

```
In [7]: print(f"Number of records: {cdv.shape[0]}")
        print(f"Number of variables: {cdv.shape[1]}")

Number of records: 11131
Number of variables: 354
```

```
In [8]: print(f"List of {cdv.shape[1]} variables names:\n")
        print(" ".join(cdv.columns))

List of 354 variables names:

INTER6 INTER ANNEEFUZ ANNEFUZ2 COLLECTE CHAMP POND identifiant SEXE AGE5 PCSENQ8 TYPOSQT DIPL4 A
```

```
In [9]: print(f"Number of lines without missing values : \
        {cdv.dropna().shape[0]} out of {cdv.shape[0]}")

Number of lines without missing values : 0 out of 11131
```

```
In [10]: nb_missing_per_var = np.sum(cdv.isnull())

In [11]: print("Number of missing values per variables :")
         nb_missing_per_var.sort_values(ascending=False).head(50)

Number of missing values per variables :

Out[11]: prescaf     11130
         SEXE_9      11124
         AGE_9       11123
         LIEN_9      11123
         SEXE_8      11107
         LIEN_8      11106
         AGE_8       11106
         SEXE_7      11078
         AGE_7       11077
         LIEN_7      11076
         AUTREAL     10995
         SEXE_6      10920
         AGE_6       10909
```

```
LIEN_6      10906
SEXE_5      10300
LIEN_5      10228
AGE_5       10226
RADIQUOI    10146
RADWHY3      9959
RADWHY9      9959
RADWHY2      9959
RADWHY4      9959
RADWHY1      9959
RADWHY7      9959
RADWHY8      9959
RADWHY5      9959
RADWHY10     9959
RADWHY11     9959
RADWHY12     9959
RADWHY13     9959
RADWHY14     9959
RADWHY6      9959
WHYLIM       9474
REVAUTR      9101
SEXE_4       8800
AGE_4        8525
LIEN_4       8518
med          8232
UDA5         8200
COMMU4       8115
RADI3        8115
COMMU1       8115
COMMU3       8115
RADI1        8115
COMMU5       8115
COMMU6       8115
COMMU7       8115
COMMU8       8115
RADI2        8115
LIMVIAND     8115
dtype: int64
```

```
In [12]: n_complete = len(nb_missing_per_var[nb_missing_per_var == 0])
         n_uncomplete = len(nb_missing_per_var[nb_missing_per_var != 0])
         print(f"Number of variables without missing values :\
         {n_complete} out of {cdv.shape[1]} variable")
         print(f"Number of variables with at least one missing values :\
         {n_uncomplete} out of {cdv.shape[1]} variable")
```

```
Number of variables without missing values :193 out of 354 variable
Number of variables with at least one missing values :161 out of 354 variable
```

In [13]:
```
complete_variables = nb_missing_per_var[nb_missing_per_var == 0].index
uncomplete_variables = nb_missing_per_var[nb_missing_per_var != 0].index
print(f"List of {n_complete} variables without missing values names:\n")
print(" ".join(complete_variables))
print(f"\nList of {n_uncomplete} variables with at least 1 missing value:\n")
print(" ".join(uncomplete_variables))
```

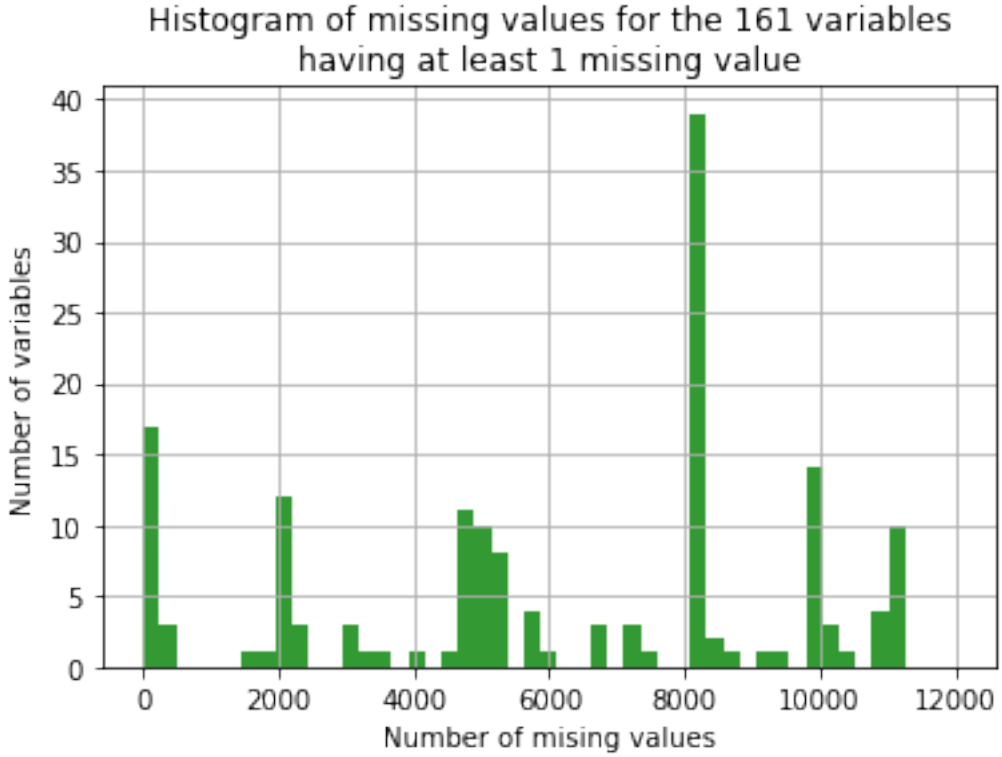
```
List of 193 variables without missing values names:

INTER6 INTER ANNEEFUZ ANNEFUZ2 COLLECTE CHAMP POND SEXE AGE5 PCSENQ8 TYPOSQT DIPL4 AGGLO5 UDA10

List of 161 variables with at least 1 missing value:

identifiant SALCOMP TYPEMPL INTERIM TYPCONT TEMPSTRA nbheures NBHEUR39 NBHEUR35 PREFPALI SALCOMP
```

In [14]:
```
fig=plt.figure()
plt.title("Histogram of missing values for the 161 variables\n\
having at least 1 missing value")
plt.ylabel(u'Number of variables')
plt.xlabel("Number of mising values")
bins = np.linspace(0, 12000,50)
plt.hist(nb_missing_per_var[uncomplete_variables],
         bins, facecolor='g', alpha=0.8)
plt.grid()
```



1.1.2 b) Year of realisation of the study and missing values

Variables **ANNEEFUZ** & **ANNEFUZ2** seems equivalent

According to the authors of the study :

En **2015**, l'enquête a été menée à la fois en face-à-face (**2 000 personnes** interrogées) et aussi online (2 000 personnes également) tous **âgés de 18 ans et plus**, résidant en **France métropolitaine (hors Corse)**. Seuls les 2000 enregistrements correspondants à l'énquête online sont présents dans le dataset.

A partir de 2016, le mode de collecte est passé en ligne et on interroge désormais **3 000 individus âgés de 15 ans et plus en France entière** (France métropolitaine, Corse et DOM-TOM).

```
In [15]: cdv["ANNEEFUZ"].unique()
```

```
Out[15]: array([2015, 2016, 2017, 2018])
```

```
In [16]: cdv["ANNEFUZ2"].unique()
```

```
Out[16]: array(['2015 online', '2016', '2017', '2018'], dtype=object)
```

```
In [17]: nb_enregistrements_anneefuz =  cdv["ANNEEFUZ"].value_counts().sort_values(ascending = F
         print("Number of records per year 'ANNEEFUZ':")
         nb_enregistrements_anneefuz
```

```
Number of records per year 'ANNEEFUZ':
```

```
Out[17]: 2016    3050
         2017    3020
         2018    3016
         2015    2045
         Name: ANNEEFUZ, dtype: int64
```

```
In [18]: nb_enregistrements_annefuz2 =  cdv["ANNEFUZ2"].value_counts()
         print("Number of records per year 'ANNEFUZ2':")
         nb_enregistrements_annefuz2
```

```
Number of records per year 'ANNEFUZ2':
```

```
Out[18]: 2016           3050
         2017           3020
         2018           3016
         2015 online    2045
         Name: ANNEFUZ2, dtype: int64
```

```
In [19]: B =  cdv.ANNEEFUZ.astype(str)
         R = cdv.loc[B != cdv["ANNEFUZ2"],["ANNEEFUZ","ANNEFUZ2"]]
         print(R["ANNEFUZ2"].unique())
         print(R["ANNEEFUZ"].unique())
```

```
['2015 online']
[2015]
```

```
In [20]: # number of missing value per variable for a given year
         na_2015 = np.sum(cdv.loc[cdv["ANNEEFUZ"] == 2015].isnull())
         na_2016 = np.sum(cdv.loc[cdv["ANNEEFUZ"] == 2016].isnull())
         na_2017 = np.sum(cdv.loc[cdv["ANNEEFUZ"] == 2017].isnull())
         na_2018 = np.sum(cdv.loc[cdv["ANNEEFUZ"] == 2018].isnull())
```

```
In [21]: complete_2015 = set(na_2015[na_2015==0].index)
         complete_2016 = set(na_2016[na_2016==0].index)
         complete_2017 = set(na_2017[na_2017==0].index)
         complete_2018 = set(na_2018[na_2018==0].index)
```

```
In [22]: print(f"Number of variable without any missing values in 2015: {len(complete_2015)}")
         print(f"Number of variable without any missing values in 2016: {len(complete_2017)}")
         print(f"Number of variable without any missing values in 2017: {len(complete_2017)}")
         print(f"Number of variable without any missing values in 2018: {len(complete_2018)}")
```

```
Number of variable without any missing values in 2015: 199
Number of variable without any missing values in 2016: 224
Number of variable without any missing values in 2017: 224
Number of variable without any missing values in 2018: 257
```

```
In [23]: missing_2015 = set(na_2015[na_2015==2045].index)
         missing_2016 = set(na_2016[na_2016==3050].index)
         missing_2017 = set(na_2017[na_2017==3020].index)
         missing_2018 = set(na_2018[na_2018==3016].index)
```

```
In [24]: print(f"Number of variable totally missing in 2015: {len(missing_2015)}")
         print(f"Number of variable totally missing in 2016: {len(missing_2016)}")
         print(f"Number of variable totally missing in 2017: {len(missing_2017)}")
         print(f"Number of variable totally missing in 2018: {len(missing_2018)}")
```

```
Number of variable totally missing in 2015: 82
Number of variable totally missing in 2016: 73
Number of variable totally missing in 2017: 56
Number of variable totally missing in 2018: 1
```

```
In [25]: full_scope = set(cdv.columns)
         scope_2015 = full_scope - missing_2015
         scope_2016 = full_scope - missing_2016
         scope_2017 = full_scope - missing_2017
         scope_2018 = full_scope - missing_2018
```

```
In [26]: print(f"Number of variable used 2015: {len(scope_2015)}")
         print(f"Number of variable used 2016: {len(scope_2016)}")
         print(f"Number of variable used 2017: {len(scope_2017)}")
         print(f"Number of variable used 2018: {len(scope_2018)}")
```

```
Number of variable used 2015: 272
Number of variable used 2016: 281
Number of variable used 2017: 298
Number of variable used 2018: 353
```

Synthesis of variable evolution over the period

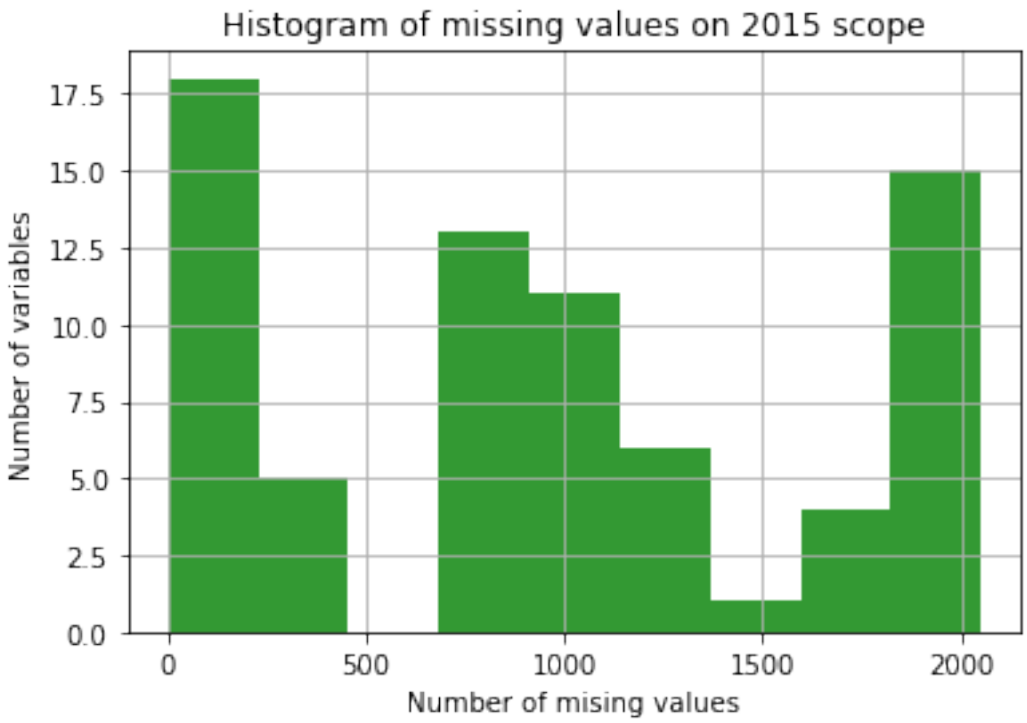
```
In [27]: print(f"2016 vs 2015\n\tNew variable ({len(scope_2016 - scope_2015)}):")
         print(" ".join(scope_2016 - scope_2015))
         print(f"\tVariable dropped ({len(scope_2015 - scope_2016)}):")
         print(" ".join(scope_2015 - scope_2016))
         print(f"\n2017 vs 2016\n\tNew variable ({len(scope_2017 - scope_2016)}):")
         print(" ".join(scope_2017 - scope_2016))
         print(f"\tVariable dropped ({len(scope_2016 - scope_2017)}):")
         print(" ".join(scope_2016 - scope_2017))
         print(f"\n2018 vs 2017\n\tNew variable ({len(scope_2018 - scope_2017)}):")
         print(" ".join(scope_2018 - scope_2017))
         print(f"\tVariable dropped ({len(scope_2017 - scope_2018)}):")
         print(" ".join(scope_2017 - scope_2018))
```

```
2016 vs 2015
        New variable (13):
CONFECOL OPICULT inseel AGE6 CONFBANK PRATCOLL CONFWEB CONFPRES CONFMEFI CONFKEUF AGGLOINS COUPL
        Variable dropped (4):
MONDIAL REVAUON VISITFAM RECEP

2017 vs 2016
        New variable (17):
popinter typodeg NOT_CAD NOT_LOG QUOTAAGE type99 REVAUON DEPCOM poppeud popdense VISITFAM pmun I
        Variable dropped (0):


2018 vs 2017
        New variable (56):
couple2 TYPLOG RADWHY8 ROBOT3 ROBOT1 COMMU1 RESIDALT RADWHY9 RADWHY5 COMMU2 i ADNORDI age_OW RAD
        Variable dropped (1):
QUOTAAGE
```

```
In [28]: nb_missing_per_var_2015 = np.sum(cdv.loc[cdv["ANNEEFUZ"]==2015].isnull())
         fig=plt.figure()
         plt.title("Histogram of missing values on 2015 scope")
         plt.ylabel(u'Number of variables')
         plt.xlabel("Number of mising values")
         bins = np.linspace(0, 2050,10)
         plt.hist(nb_missing_per_var_2015[scope_2015 - complete_2015],
                  bins, facecolor='g', alpha=0.8)
         plt.grid()
```




```
In [29]: nb_missing_per_var_2015[scope_2015 - complete_2015].sort_values(ascending=False)
```

```
Out[29]: LIEN_9      2043
         SEXE_9      2043
         AGE_9       2043
         LIEN_8      2040
         SEXE_8      2040
         AGE_8       2040
         AGE_7       2038
         LIEN_7      2038
         SEXE_7      2038
         LIEN_6      2014
         SEXE_6      2014
         AGE_6       2014
         SEXE_5      1886
         LIEN_5      1886
         AGE_5       1886
         REVAUTR     1715
         LIEN_4      1601
         SEXE_4      1601
         AGE_4       1601
         interim2    1405
         SALCOMPC    1321
         PCSCONJ     1321
```

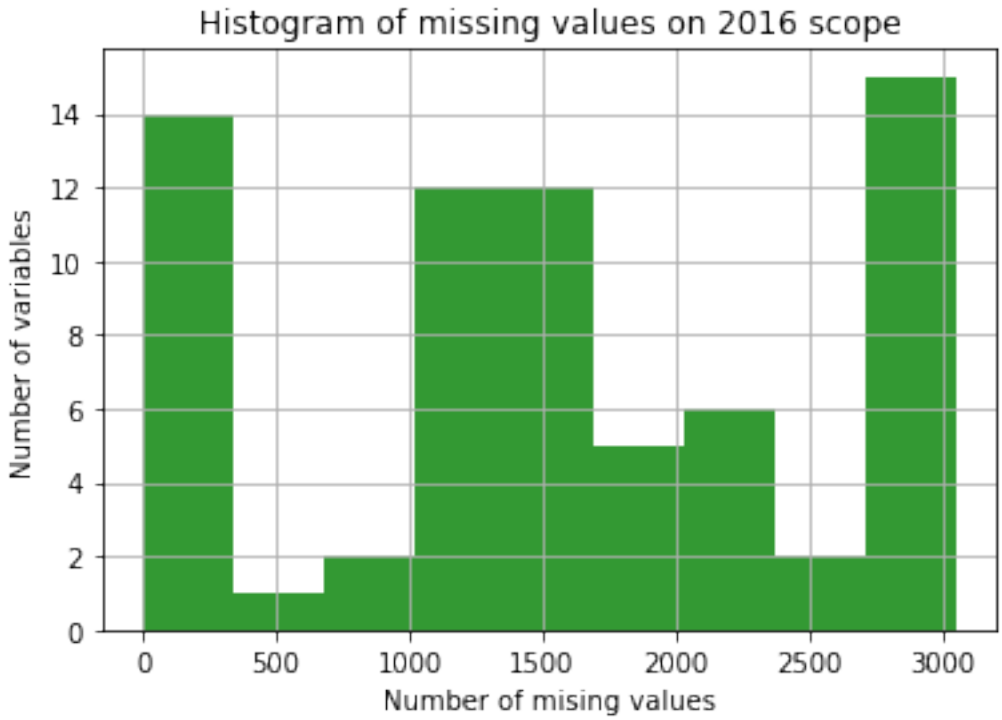
```
SEXE_3      1289
AGE_3       1289
LIEN_3      1289
SALCOMPI    1236
TYPCONT     1089
INTERIM     1063
typcont2    1063
PRIVPUB     1063
            ...
RE_EQUI      794
RE_ALIM      794
RE_HABI      794
RE_LOG       794
ACTCONJ      737
REVCONJ      737
NBENF        687
REVAUON      400
LIEN_2       400
SEXE_2       400
AGE_2        400
PROGRAD      303
statut99     105
zau1999      105
zau2010      103
RURAURBA      93
REVsqt        63
REVUC         63
REVTOT        57
NOT_PROF      25
SENSIENV      24
NOT_AMIS      23
NOT_COHE      23
NOT_POLI      20
NOT_LIBR      19
NOT_FAMI      11
NBPERS         9
NBPIECES       9
SITUFAM        9
NBUC           9
Length: 73, dtype: int64
```

In [30]:
```
print("List of variable with more than 75% missing values in 2015:\n")
l = nb_missing_per_var_2015[scope_2015 -
                            complete_2015][nb_missing_per_var_2015 >
                                           0.75*2045]
print(" ".join(l.index))
```

```
List of variable with more than 75% missing values in 2015:
```

```
LIEN_7 SEXE_5 SEXE_4 AGE_9 AGE_6 AGE_8 LIEN_8 LIEN_5 AGE_5 LIEN_6 SEXE_6 SEXE_9 SEXE_7 SEXE_8 RE
```

```
In [31]: nb_missing_per_var_2016 = np.sum(cdv.loc[cdv["ANNEEFUZ"]==2016].isnull())
         fig=plt.figure()
         plt.title("Histogram of missing values on 2016 scope")
         plt.ylabel(u'Number of variables')
         plt.xlabel("Number of mising values")
         bins = np.linspace(0, 3050,10)
         plt.hist(nb_missing_per_var_2016[scope_2016 - complete_2016],
                  bins, facecolor='g', alpha=0.8)
         plt.grid()
```




```
In [32]: nb_missing_per_var_2016[scope_2016
                                  - complete_2016].sort_values(ascending=False)
```

```
Out[32]: SEXE_9      3048
         AGE_9       3047
         LIEN_9      3047
         SEXE_8      3045
         AGE_8       3044
         LIEN_8      3044
         SEXE_7      3040
         AGE_7       3039
```

```
LIEN_7      3038
SEXE_6      3010
AGE_6       2999
LIEN_6      2996
SEXE_5      2881
LIEN_5      2809
AGE_5       2807
SEXE_4      2630
SEXE_3      2388
AGE_4       2355
LIEN_4      2348
REVAUTR     2164
interim2    2120
PCSCONJ     2079
SALCOMPC    2079
AGE_3       1901
LIEN_3      1891
SALCOMPI    1884
SEXE_2      1804
TYPCONT     1737
PRIVPUB     1677
INTERIM     1677
            ...
REVCONJ     1376
ACTCONJ     1340
RE_ALIM     1241
RE_VAC      1241
RE_EQUI     1241
RE_ENF      1241
RE_LOG      1241
RE_WEB      1241
RE_MEDI     1241
RE_HABI     1241
RE_VOIT     1241
RE_TABAL    1241
NBENF       1178
AGE_2        809
LIEN_2       726
PROGRAD      497
inseel        95
REVTOT        74
REVUC         74
REVsqt        74
NOT_PROF      35
NOT_COHE      34
NOT_FAMI      33
NOT_AMIS      32
NOT_POLI      32
```

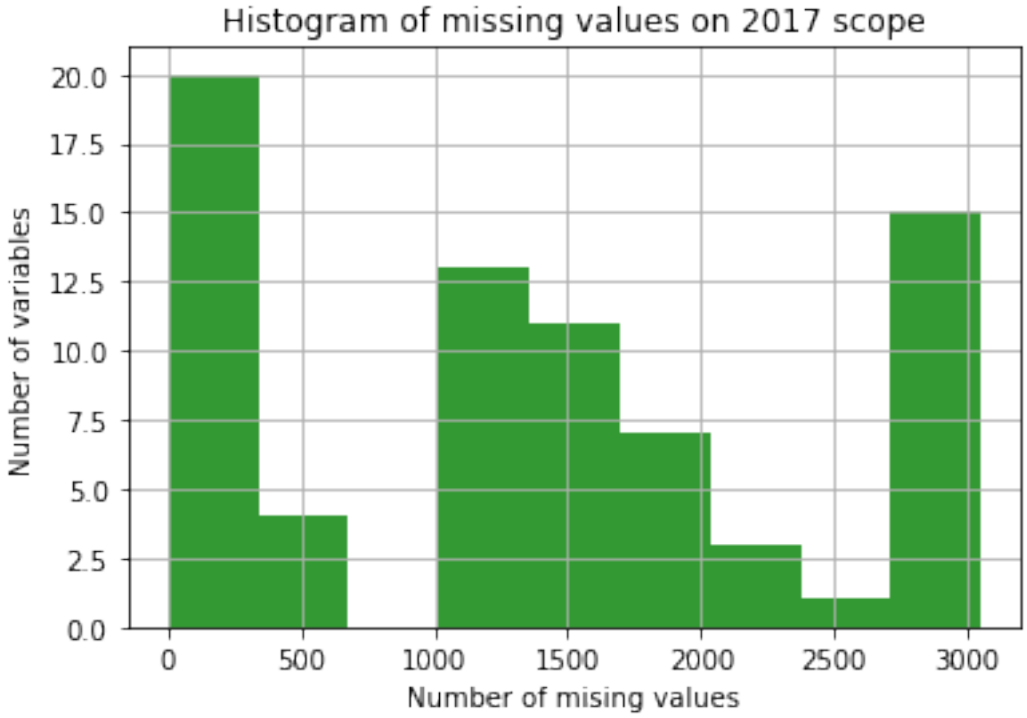
```
NOT_LIBR      29
PRATCOLL      27
zau2010       18
SENSIENV      15
RURAURBA       4
Length: 69, dtype: int64
```

In [33]:
```
print("List of variable with more than 75% missing values in 2016:\n")
l = nb_missing_per_var_2016[scope_2016 -
                            complete_2016][nb_missing_per_var_2016 >
                                           0.75*3050]
print(" ".join(l.index))
```

```
List of variable with more than 75% missing values in 2016:

LIEN_7 SEXE_5 SEXE_4 AGE_9 AGE_6 AGE_8 LIEN_8 LIEN_5 AGE_5 LIEN_6 SEXE_6 SEXE_9 SEXE_7 SEXE_8 LI
```

In [34]:
```
nb_missing_per_var_2017 = np.sum(cdv.loc[cdv["ANNEEFUZ"]==2017].isnull())
fig=plt.figure()
plt.title("Histogram of missing values on 2017 scope")
plt.ylabel(u'Number of variables')
plt.xlabel("Number of mising values")
bins = np.linspace(0, 3050,10)
plt.hist(nb_missing_per_var_2017[scope_2017 - complete_2017],
         bins, facecolor='g', alpha=0.8)
plt.grid()
```



```
In [35]: nb_missing_per_var_2017[scope_2017
                                 - complete_2017].sort_values(ascending=False)
```

```
Out[35]: LIEN_9      3018
         SEXE_9      3018
         AGE_9       3018
         SEXE_8      3013
         LIEN_8      3013
         AGE_8       3013
         LIEN_7      2999
         AGE_7       2999
         SEXE_7      2999
         SEXE_6      2946
         AGE_6       2946
         LIEN_6      2946
         LIEN_5      2775
         AGE_5       2775
         SEXE_5      2775
         REVAUTR     2518
         LIEN_4      2278
         AGE_4       2278
         SEXE_4      2278
         interim2    1967
         PCSCONJ     1920
         SALCOMPC    1920
         SALCOMPI    1869
         SEXE_3      1715
         AGE_3       1715
         LIEN_3      1715
         TYPCONT     1590
         INTERIM     1523
         typcont2    1523
         PRIVPUB     1523
                     ...
         RE_EQUI     1327
         RE_ALIM     1327
         RE_HABI     1327
         REVCONJ     1232
         ACTCONJ     1203
         NBENF       1063
         LIEN_2       578
         SEXE_2       578
         AGE_2        578
         PROGRAD      435
         inseel       119
```

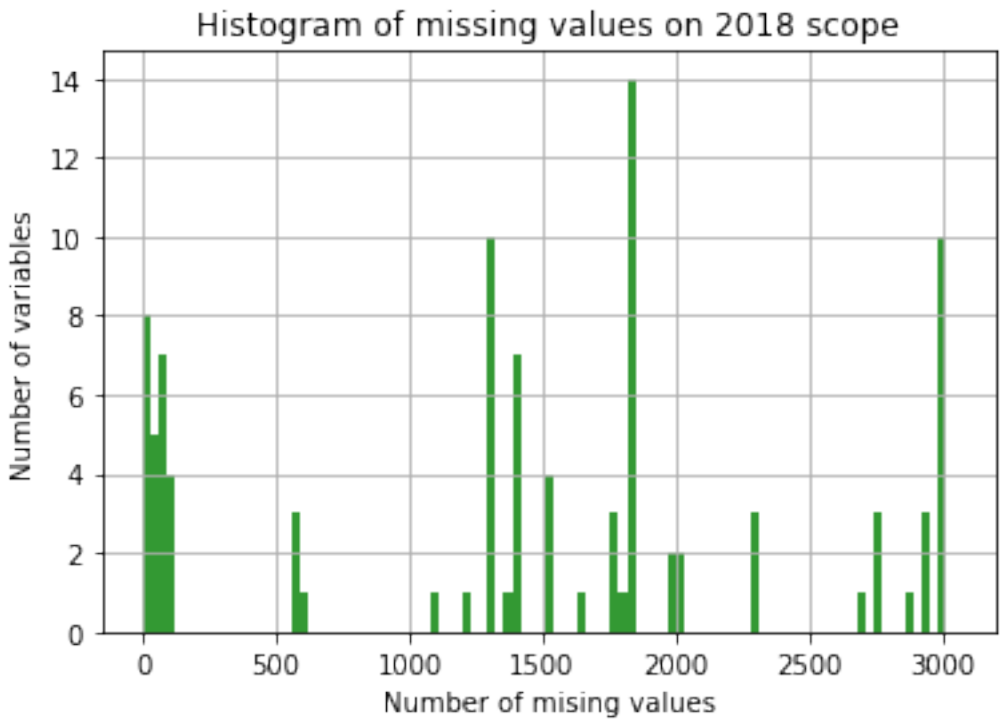
```
REVUC        105
REVTOT       105
REVsqt       105
NOT_PROF      55
NOT_LIBR      52
NOT_POLI      47
NOT_CAD       44
NOT_LOG       43
NOT_FAMI      39
NOT_COHE      39
NOT_AMIS      37
typodeg       24
pmun          24
poptrpeu      24
DEPCOM        24
popinter      24
popdense      24
poppeud       24
SENSIENV      15
Length: 74, dtype: int64
```

In [36]:
```
print("List of variable with more than 75% missing values in 2017:\n")
l = nb_missing_per_var_2017[scope_2017 -
                            complete_2017][nb_missing_per_var_2017 >
                                           0.75*3050]
print(" ".join(l.index))
```

```
List of variable with more than 75% missing values in 2017:

LIEN_7 SEXE_5 AGE_9 AGE_6 AGE_8 LIEN_8 LIEN_5 AGE_5 LIEN_6 SEXE_6 SEXE_9 SEXE_7 SEXE_8 REVAUTR A
```

In [37]:
```
nb_missing_per_var_2018 = np.sum(cdv.loc[cdv["ANNEEFUZ"]==2018].isnull())
fig=plt.figure()
plt.title("Histogram of missing values on 2018 scope")
plt.ylabel(u'Number of variables')
plt.xlabel("Number of mising values")
bins = np.linspace(0, 3050,100)
plt.hist(nb_missing_per_var_2018[scope_2018 - complete_2018],
         bins, facecolor='g', alpha=0.8)
plt.grid()
```




```
In [38]: nb_missing_per_var_2018[scope_2018
                                - complete_2018].sort_values(ascending=False)
```

```
Out[38]: LIEN_9      3015
         SEXE_9      3015
         AGE_9       3015
         prescaf     3015
         AGE_8       3009
         SEXE_8      3009
         LIEN_8      3009
         SEXE_7      3001
         AGE_7       3001
         LIEN_7      3001
         AGE_6       2950
         SEXE_6      2950
         LIEN_6      2950
         AUTREAL     2880
         SEXE_5      2758
         AGE_5       2758
         LIEN_5      2758
         REVAUTR     2704
         LIEN_4      2291
         AGE_4       2291
         SEXE_4      2291
```

```
RADIQUOI    2031
interim2    2006
SALCOMPC    1975
PCSCONJ     1975
RADWHY1     1844
RADWHY14    1844
RADWHY8     1844
RADWHY6     1844
RADWHY10    1844
            ...
ACTCONJ     1219
NBENF       1096
PROGRAD      608
LIEN_2       573
AGE_2        573
SEXE_2       573
med          117
REVUC        105
REVTOT       105
REVsqt       105
NOT_POLI      85
UDA5          85
NOT_COHE      81
NOT_LIBR      71
NOT_PROF      70
NOT_CAD       67
NOT_AMIS      65
NOT_LOG       61
ASSOAUTR      59
SENSIENV      54
NOT_FAMI      50
PCSCON7       47
pmun          15
typodeg       15
popdense      15
poptrpeu      15
popinter      15
poppeud       15
DEPCOM        15
RURAURBA       6
Length: 96, dtype: int64
```

```
In [39]: print("List of variable with more than 75% missing values in 2017:\n")
         l = nb_missing_per_var_2018[scope_2018 -
                                     complete_2018][nb_missing_per_var_2018 >
                                                    0.75*3050]
         print(" ".join(l.index))
```

```
List of variable with more than 75% missing values in 2017:
```

```
SEXE_4 LIEN_5 SEXE_7 LIEN_7 AGE_6 AGE_8 LIEN_8 AGE_5 LIEN_6 LIEN_9 SEXE_5 AGE_9 SEXE_9 SEXE_8 LI
```

1.2 Selection and classification of variables

1.2.1 a) Variable to be predicted - "HEUREUX"

```
In [40]: cdv['HEUREUX'].value_counts().sort_values(ascending = False)
```

```
Out[40]: Assez souvent        5423
         Occasionnellement    3665
         Très souvent         1758
         Jamais                203
         [Nsp]                  82
         Name: HEUREUX, dtype: int64
```

1.2.2 b) Variable common to all years

```
In [41]: common_variables = (scope_2015 & scope_2016 & scope_2017 & scope_2018)
         len(common_variables)
```

```
Out[41]: 268
```

```
In [42]: cdv_restricted_common = cdv.loc[:,common_variables]
```

```
In [43]: l = list(common_variables)
         l.sort()
         print("List of variable common to all years")
         print(l)
```

```
List of variable common to all years
['ACM1', 'ACM10', 'ACM11', 'ACM12', 'ACM2', 'ACM3', 'ACM4', 'ACM5', 'ACM6', 'ACM7', 'ACM8', 'ACM
```

1.2.3 c) variable analysis - link with CDV study

```
In [44]: print(list(cdv.columns))
```

```
['INTER6', 'INTER', 'ANNEEFUZ', 'ANNEFUZ2', 'COLLECTE', 'CHAMP', 'POND', 'identifiant', 'SEXE',
```

```
In [45]: cdv.loc[:,["POND","INTER6",
                    "INTER","COLLECTE",
                    "CHAMP","identifiant"]].head()
```

```
Out[45]:        POND  INTER6  INTER COLLECTE                  CHAMP identifiant
         0  1.313554  373001   3001   Online  18 ans et + métropole         NaN
         1  2.009015  373002   3002   Online  18 ans et + métropole         NaN
         2  0.217607  373003   3003   Online  18 ans et + métropole         NaN
         3  0.539351  373004   3004   Online  18 ans et + métropole         NaN
         4  0.270204  373005   3005   Online  18 ans et + métropole         NaN
```

```
In [46]: cdv["CHAMP"].unique()
```

```
Out[46]: array(['18 ans et + métropole', '15-17 ans + DOM + Corse'], dtype=object)
```

```
In [47]: cdv["COLLECTE"].unique()
```

```
Out[47]: array(['Online'], dtype=object)
```

```
In [48]: # Variables not present in the list ???
         cdv["RURAURBA"].unique()
```

```
Out[48]: array(['PR', 'PU', nan, 'IN'], dtype=object)
```

```
In [49]: cdv["AGGLOINS"].unique()
```

```
Out[49]: array([ nan,   0.,   2.,   1.,   7.,   4.,   8.,   5.,   3.,   6.])
```

```
In [50]: # List of variable explained in the exceel file provided
         liste_explained = {"INTER6","ANNEEFUZ","ANNEEFUZ2","COLLECTE","SEXE",
                            "AGE5","PCSENQ8","TYPOSQT",
                            "DIPL4","AGGLO5","UDA10","SITUEMP3","AGGLO9","AGE",
                            "EXERCPRO","SITUEMP","SITUEMP5",
                            "SITUEMP6","SALCOMP","INTERIM","TYPCONT","TEMPSTRA",
                            "nbheures","NBHEUR39",
                            "NBHEUR35","PREFPALI","SALCOMPI","CHERCHEM","NBCHOM",
                            "STATMAT","ACTCONJ","SALCOMPC",
                            "ENFANTS","NBENF","NBENF6","DIPLOME","FAMILLE",
                            "UNIONGAY","ADOPTGAY","TRAVFEM","NB0003","NB0306",
                            "NB0610","NB1016",
                            "NB1620","NB2099","NBPIECE6","LOGSUFFI","DEPLOG",
                            "DEPLOG3","CADVIE","CADVIE3","SECUR3",
                            "MODCHAUF","TYPCHAUF","TELFIXE","TELMOB","SENSIENV",
                            "TAXENV","HANDICAP","SOUFFTET","SOUFFDOS",
                            "SOUFFNER","SOUFFDEP","SOUFFINS","ETATSAN","NBPERS",
                            "NBPERS5","SEXE_2","SEXE_3",
                            "SEXE_4","SEXE_5","SEXE_6","SEXE_7","SEXE_8","SEXE_9",
                            "AGE_2","AGE_3",
                            "AGE_4","AGE_5","AGE_6","AGE_7","AGE_8","AGE_9",
                            "LIEN_2","LIEN_3","LIEN_4","LIEN_5","LIEN_6",
                            "LIEN_7","LIEN_8","LIEN_9","RESTRICT","NIVPERSO",
                            "NIVFRAN","NIVFRAN4","CDV5","BANQEPA",
                            "BANQVIE","ASSOSPOR","ASSOCULT","ASSOCONF",
                            "ASSOJEUN","ASSOSYND","ASSOENVI",
                            "ASSOPARE","ASSOCONS","ASSOPOLI","ASSOHUMA",
                            "ASSOAUTR","FREQSPOR",
                            "FREQTELE","RAISPAUV","CHOAVANT","CHOVOLON",
                            "OPIRSA","JUSTICE","TRANSFST","PREOCCU1",
                            "PREOCCU2","INQAGRE3","INQALIM","CLASSESO",
                            "HEUREUX","CONFGOUV","revtot7",
                            "NBUC","TYPLOG2","TYPLOG3","AGESEX12",
                            "PCSENQ36","UDA14","zau1999","POND","dpt"}
```

```
In [51]: print(f"Number of variable explained in the exceel file \
         ... :{len(liste_explained)}")
```

```
Number of variable explained in the exceel file :135
```

```
In [52]: columns = set(cdv.columns)
```

```
In [53]: print(f"Variables explained but not present in the dataset :\
         {len(liste_explained - columns)}\n")
         print(" ".join(liste_explained - columns))
```

```
Variables explained but not present in the dataset :4

TYPLOG3 dpt TELFIXE ANNEEFUZ2
```

```
In [54]: print(f"Variables present in the dataset but not explained :\
         {len(columns - liste_explained)}\n")
         print(" ".join(columns - liste_explained))
```

```
Variables present in the dataset but not explained :223

OPICULT typcont2 RE_EQUI TYPLOG ROBOT1 COMMU1 PCSENQ9 pmun RADWHY5 ACM5 NOT_LIBR STATLOG4 ACM11
```

```
In [55]: print(f"Variables present in the dataset for all years but not explained :\
         {len(common_variables - liste_explained)}\n")
         print(" ".join(common_variables - liste_explained))
```

```
Variables present in the dataset for all years but not explained :137

typcont2 RE_EQUI PCSENQ9 ACM5 NOT_LIBR STATLOG4 ORDLIB ACM11 AGEDIP2 PROGRAD NOT_FAMI INQAGRES N
```

1.2.4 d) bottom up...

```
In [56]: cdv["REVENQ"].describe()
```

```
Out[56]: count     11131.000000
         mean      71370.251101
         std      253880.504617
         min           0.000000
         25%        1100.000000
         50%        1800.000000
         75%        2800.000000
         max      999999.000000
         Name: REVENQ, dtype: float64
```