Exploration with python

August 20, 2018

1 CDV study - data analysis

```
In [2]: from pathlib import Path
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        #%pylab inline
```

```
In [3]: path_project = Path.home() / Path('Google Drive/Felix')
        path_data = path_project / Path("data")
```

```
In [4]: # 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 [5]: cdv.shape
```

```
Out[5]: (11131, 354)
```

```
In [6]: # 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 [7]: cdv_ssfmt.shape
```

```
Out[7]: (11131, 354)
```

1.1 Dataset Size and missing values analysis

```
In [8]: 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 [9]: 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 [10]: print(f"Number of lines without missing values : {cdv.dropna().shape[0]} out of {cdv.sh
```

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

```
In [11]: number_missing_values_per_variable = np.sum(cdv.isnull())
```

```
In [12]: print("Number of missing values per variables :")
         number_missing_values_per_variable
```

```
Number of missing values per variables :
```

```
Out[12]: INTER6            0
         INTER             0
         ANNEEFUZ          0
         ANNEFUZ2          0
         COLLECTE          0
         CHAMP             0
         POND              0
         identifiant    8115
         SEXE              0
         AGE5              0
         PCSENQ8           0
         TYPOSQT           0
         DIPL4             0
         AGGLO5            0
         UDA10             0
         SITUEMP3          0
         AGEDIP2           0
         DPT               0
         COMINSEE          0
         AGGLO9            0
         AGE               0
         DIPLOME           0
         EXERCPRO          0
         SITUEMP           0
         SITUEMP5          0
         SITUEMP6          0
         SALCOMP        5242
         TYPEMPL        5792
```

```
INTERIM         5792
TYPCONT         6055
                ...
AUTREAL        10995
age_OW          8115
UDA5            8200
CSP6            8115
CP              8115
TYPLOG          8115
inseel          2259
inseenum        8115
couple2         8115
cpt             8115
AGE6            2045
PCSRED10        5095
prescaf        11130
refus2          8115
info            8115
med             8232
i               8115
com             8115
type99          5095
AGGLOINS        2045
DEPCOM          5134
pmun            5134
QUOTAAGE        8111
PRIVPUB         5792
interim2        7498
EMP7               0
typcont2        5792
REVTOT6            0
an_enq             0
an_nais            0
Length: 354, dtype: int64
```

In [13]:
```
n_complete = len(number_missing_values_per_variable[number_missing_values_per_variable
n_uncomplete = len(number_missing_values_per_variable[number_missing_values_per_variabl
print(f"Number of variables without missing values : {n_complete} out of {cdv.shape[1]}
print(f"Number of variables with at least one missing values : {n_uncomplete} out of {c
```

```
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 [14]:
```
complete_variables = number_missing_values_per_variable[number_missing_values_per_varia
uncomplete_variables = number_missing_values_per_variable[number_missing_values_per_var
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 values names:\n")
        print(" ".join(uncomplete_variables))
```

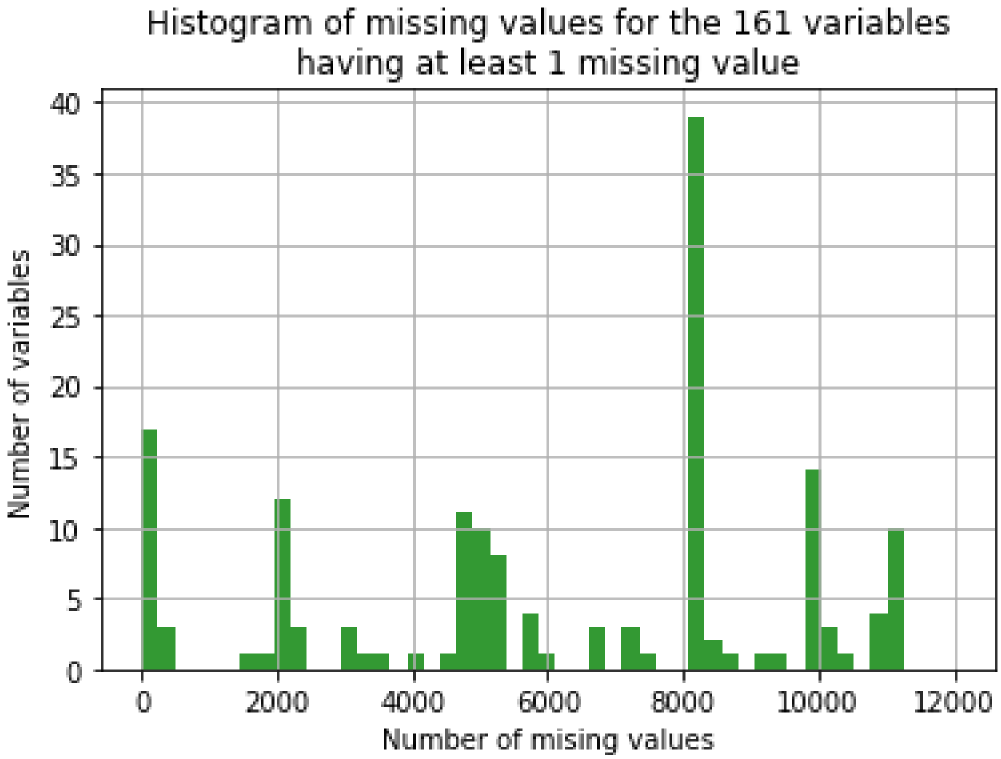
```
List of 193 variables without missing values names:

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

List of 161 variables with at least 1 missing values names:

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

```
In [15]: fig=plt.figure()
         plt.title("Histogram of missing values for the 161 variables\nhaving at least 1 missing
         plt.ylabel(u'Number of variables')
         plt.xlabel("Number of mising values")
         bins = np.linspace(0, 12000,50)
         plt.hist(number_missing_values_per_variable[uncomplete_variables], bins, facecolor='g',
         plt.grid()
```



```
In [ ]: sub_cdv = cdv.loc[:,["NOT_FAMI", "NOT_PROF", "NOT_AMIS",
                            "NOT_COHE", "NOT_POLI", "NOT_LIBR",
                            "NOT_LOG", "NOT_CAD", "ANNEEFUZ", "INTER6"]]
```

```
In [ ]: sub_cdv.shape
```

```
In [ ]: sub_cdv.head()
```

```
In [ ]: cdv1718 = sub_cdv.loc[sub_cdv["ANNEEFUZ"].isin([2017,2018]),:]
```

```
In [ ]: cdv1718.describe()
```

```
In [ ]: np.sum(cdv1718.isnull())
```

```
In [ ]: cdv1718.shape
```

1.1.1 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 [16]: cdv["ANNEEFUZ"].unique()

Out[16]: array([2015, 2016, 2017, 2018])
```

```
In [17]: cdv["ANNEFUZ2"].unique()

Out[17]: array(['2015 online', '2016', '2017', '2018'], dtype=object)
```

```
In [18]: 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[18]: 2016    3050
         2017    3020
         2018    3016
         2015    2045
         Name: ANNEEFUZ, dtype: int64
```

```
In [19]: nb_enregistrements_annefuz2 =  cdv["ANNEFUZ2"].value_counts()
         print("Number of records per year 'ANNEFUZ2':")
         nb_enregistrements_annefuz2

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

```
Out[19]: 2016           3050
         2017           3020
         2018           3016
         2015 online    2045
         Name: ANNEFUZ2, dtype: int64
```

```
In [20]: 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 [21]: # 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 [22]: 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 [23]: 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 [32]: 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 [33]: 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 [34]: 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 [35]: 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 [36]: 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))
```

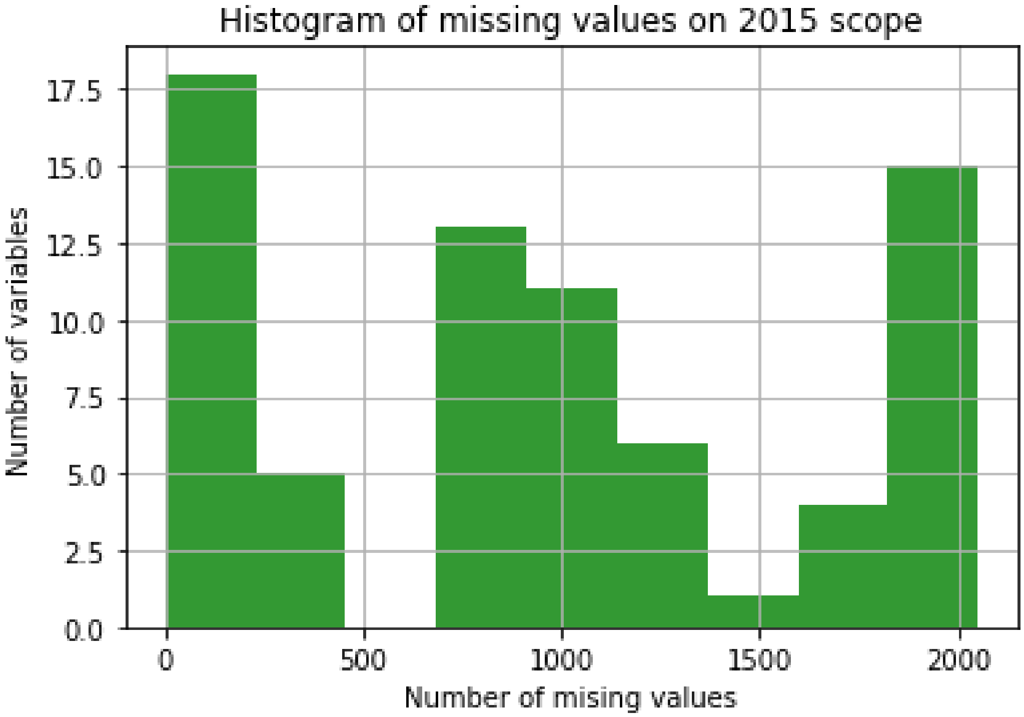
```
2016 vs 2015
        New variable (13):
CONFKEUF AGGLOINS AGE6 CONFPRES STATLOGB CONFBANK inseel COUPLE OPICULT CONFWEB CONFECOL PRATCOL
        Variable dropped (4):
RECEP MONDIAL VISITFAM REVAUON

2017 vs 2016
        New variable (17):
NOT_CAD type99 REVAUON QUOTAAGE DEPCOM NOT_LOG MONDIAL poptrpeu pmun popinter RECEP ISEGO PCSRED
        Variable dropped (0):



2018 vs 2017
        New variable (56):
info i CSP6 couple2 RADI3 RADWHY3 CP RADWHY6 cpt ROBOT2 RADIQUOI UDA5 RADWHY9 RADWHY13 COMMU5 RO
        Variable dropped (1):
QUOTAAGE
```

```
In [66]: number_missing_values_per_variable_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(number_missing_values_per_variable_2015[scope_2015 - complete_2015], bins, fac
         plt.grid()
```



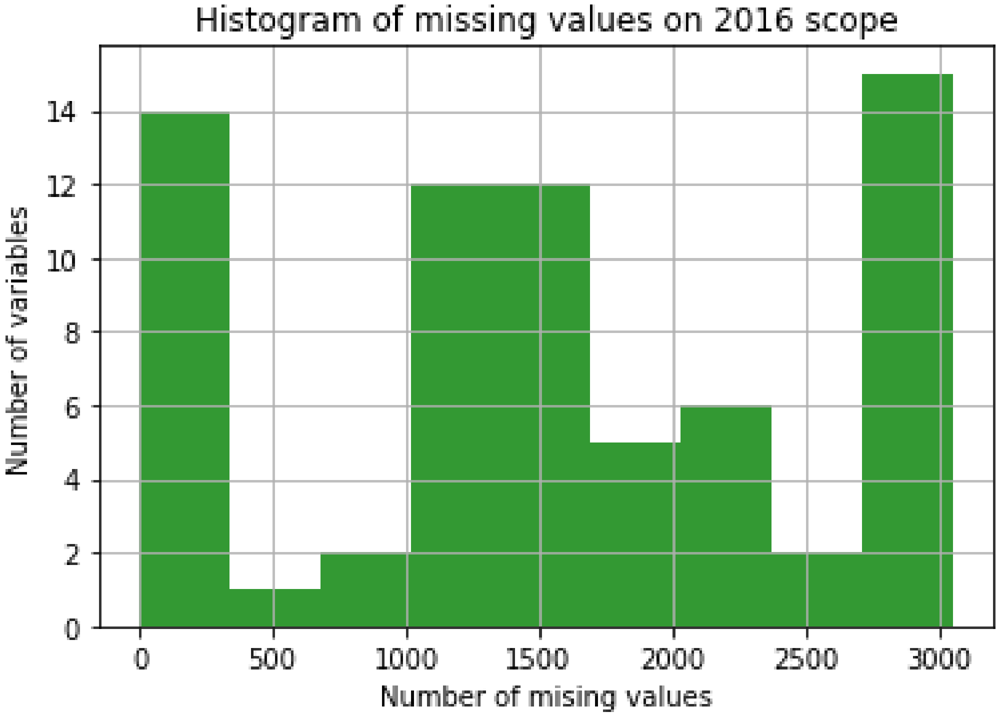
```
In [78]: print("List of variable with more than 75% missing values in 2015:\n")
         l = number_missing_values_per_variable_2015[scope_2015 -
                                                     complete_2015][number_missing_values_per_va
         print(" ".join(l.index))
```

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

REVAUTR SEXE_5 SEXE_8 AGE_7 AGE_8 LIEN_9 AGE_9 LIEN_4 SEXE_4 LIEN_6 SEXE_9 AGE_4 AGE_5 LIEN_8 AG
```

```
In [87]: number_missing_values_per_variable_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(number_missing_values_per_variable_2016[scope_2016 - complete_2016], bins, fac
plt.grid()
```

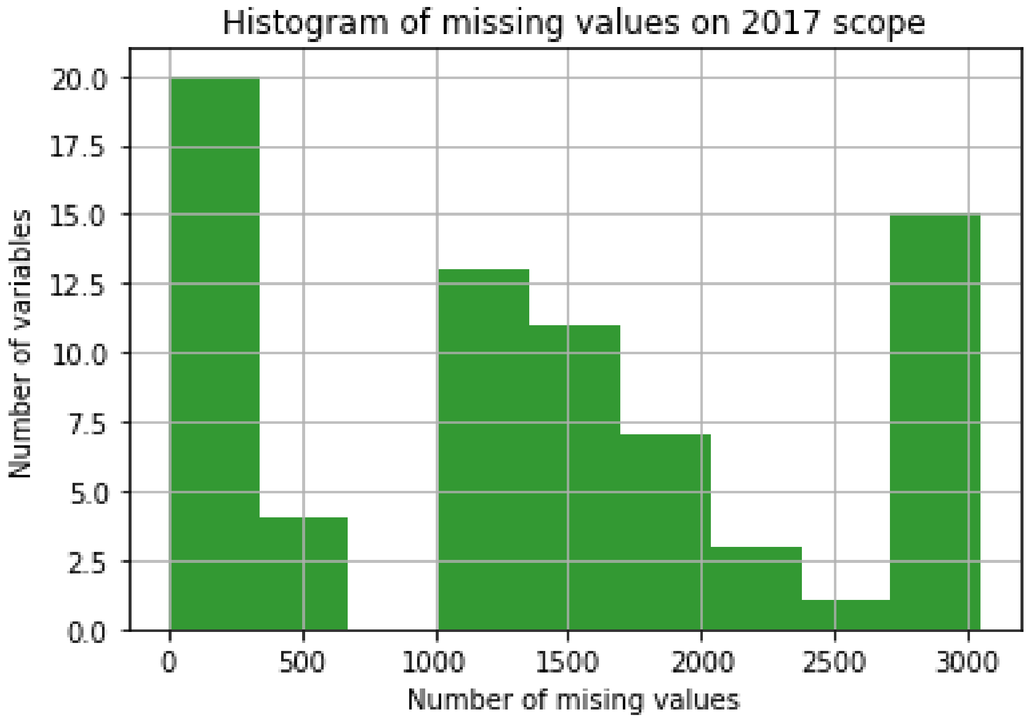


```
In [88]: print("List of variable with more than 75% missing values in 2016:\n")
         l = number_missing_values_per_variable_2016[scope_2016 -
                                                     complete_2016][number_missing_values_per_va
         print(" ".join(l.index))
```

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

SEXE_5 SEXE_8 AGE_7 AGE_8 LIEN_9 AGE_9 LIEN_4 SEXE_4 LIEN_6 SEXE_9 AGE_4 AGE_5 LIEN_8 AGE_6 SEXE
```

```
In [89]: number_missing_values_per_variable_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(number_missing_values_per_variable_2017[scope_2017 - complete_2017], bins, fac
         plt.grid()
```

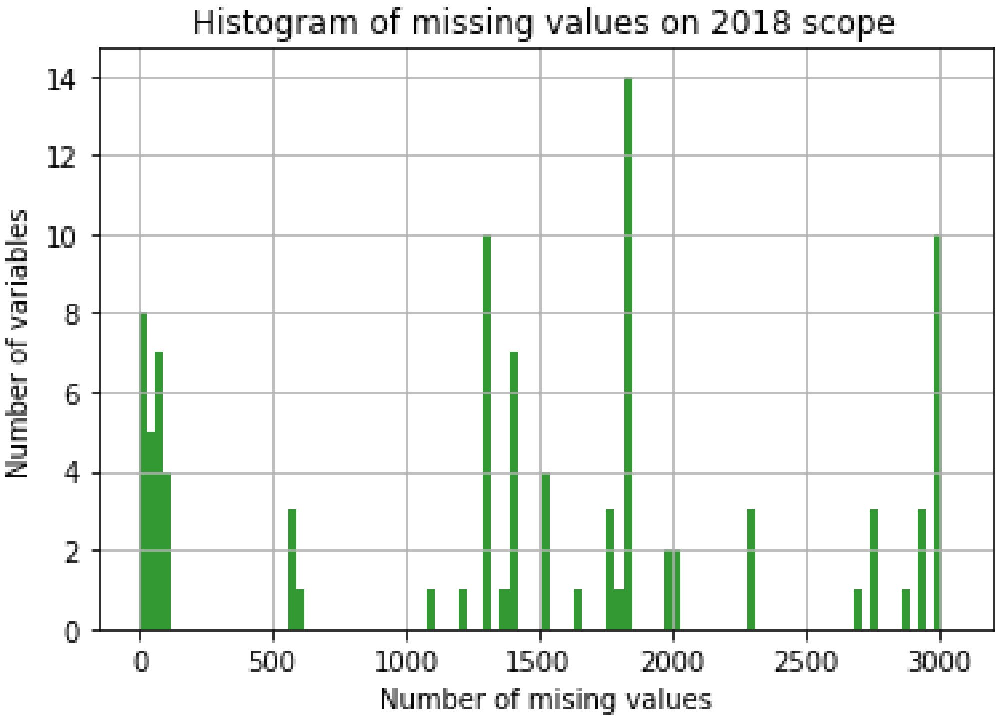


```
In [90]: print("List of variable with more than 75% missing values in 2017:\n")
         l = number_missing_values_per_variable_2017[scope_2017 -
                                                      complete_2017][number_missing_values_per_va
         print(" ".join(l.index))

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

REVAUTR SEXE_5 SEXE_8 AGE_7 AGE_8 LIEN_9 AGE_9 LIEN_6 SEXE_9 AGE_5 AGE_6 LIEN_8 SEXE_6 LIEN_7 LI
```

```
In [93]: number_missing_values_per_variable_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(number_missing_values_per_variable_2018[scope_2018 - complete_2018], bins, fac
         plt.grid()
```



```
In [92]: print("List of variable with more than 75% missing values in 2017:\n")
         l = number_missing_values_per_variable_2018[scope_2018 -
                                                     complete_2018][number_missing_values_per_va
         print(" ".join(l.index))
```

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

AGE_8 SEXE_7 SEXE_4 prescaf SEXE_8 LIEN_6 AGE_5 AGE_6 SEXE_6 LIEN_5 REVAUTR SEXE_5 LIEN_9 AGE_9
```