M2.951_20201_Pràctica2_LoanStatsDataset

December 23, 2020

1 Pràctica 2: Neteja i anàlisi de les dades

Nom i cognoms:

- Joan Oliva Costas
- Ana Cortés Besolí

1.1 Càrrega de dades i exploració del dataset

El dataset escollit és **LoanStats dataset**. Aquest dataset conté dades sobre els préstecs demanats en una empresa de crèdit, LendingClub.

El dataset es pot trobar a un repositori de BigMl de manera oberta.

Les dades representen peticions de préstec que l'empresa ha rebut dels seus clients. En total, conté 48.597 registres i 19 columnes. L'objectiu de les dades és marcar si el préstec és fraudulent o no per tal de poder prendre una decisió sobre la seva concessió.

El dataset també s'adjunta amb el mateix exercici i es pot trobar en el fitxer 'LoanStatsDataset.csv'.

1.1.1 Descripció de les variables de LoanStats

Les columnes del dataset i la seva interpretació és la següent:

- **Total Amount Funded**: volum del préstec demanat.
- **Loan Length**: duració del préstec expressat en mesos.
- **Monthly PAYMENT**: import de la mensualitat a pagar.
- **Debt-To-Income Ratio:** relació deute-ingressos, és la ràtio d'endeutament respecte als ingressos del client.
- **Home Ownership:** Aquest camp ens indica si el client és propietari de la casa on viu, si la té hipotecada o si viu de lloguer. Té 5 possibles valors:
  - Rent: viu de lloguer
  - Mortgage: paga hipoteca a la casa on viu
  - Own: és propietària de la casa on viu
  - Any: té alguna propietat
  - None: no té cap propietat.
- **Monthly Income:** Ingressos mensuals.
- **Approx.Fico Score:** És un número de tres dígits que dóna informació de si un client té risc o no a l'hora de pagar un préstec.
- **Open CREDIT Lines:** Correspon al nombre de préstecs que té en l'actualitat el client.
- **Total CREDIT Lines:** Correspon al nombre total de préstecs que ha tingut el client.

- **Revolving CREDIT Balance:** És el saldo de crèdit del client.
- **Revolving Line Utilization:** És el percentatge del saldo de crèdit utilitzat pel client.
- **Inquiries in the Last 6 Months:** Nombre de sol·licituds que s'han demanat del client per avaluar el risc per aprovar una sol·licitud, o bé perquè el client ha demanat un altre préstec, hipoteca o augment de línia de crèdit per a les targetes. Com més consultes fetes, vol dir que més vegades ha estat avaluat el risc del client suposadament perquè més crèdit ha demanat.
- **Accounts Now Delinquent:** Variable que conté els valors 0 o 1 per determinar si el client té comptes morosos en l'actualitat. 0 (no morós) i 1 (morós).
- **Delinquencies (Last 2 yrs):** Nombre de vegades que el client ha estat morós en els últims dos anys.
- **Months Since Last Delinquency:** mesos des de l'última morositat del client.
- **Public Records On File:** són registres que apareixen a l'informe de crèdit a causa de problemes de pagament, sentències o gravàmens fiscals. Si un client té aquests tipus de registres, s'informa que el client presenta morositat greu.
- **Months Since Last Record:** variable que informa del nombre de mesos des de l'últim registre públic.
- **Employment Length:** Nombre d'anys que el client porta treballant.
- **Status:** Variable que determina si el préstec pot resultar fraudulent o no.

1.1.2 Resum de les dades

En aquesta secció es resumeix de manera breu les dades i els seus principals estadístics.

[3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import matplotlib.patches as mpatches
import seaborn as sns
from scipy.stats import chi2_contingency
from scipy import stats

# Preprocessat i modelatge KNN

# Configuració warnings
# ------------------------------------------------------------------------------
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
# Carreguem les dades en un dataset de pandas i visualitzem alguns estadístics␣
 ↪importants
df = pd.read_csv('LoanStatsDataset.csv', sep=',')
print("5 primeres files del dataset:")
df.head()
```

```
5 primeres files del dataset:
```

[2]:

	Total Amount Funded	Loan Length	Monthly PAYMENT	Debt-To-Income Ratio
0	500	36 months	15.67	0.00
1	500	36 months	15.69	4.27
2	500	36 months	15.75	14.02
3	500	36 months	15.76	2.15
4	500	36 months	15.91	0.00

	Home Ownership	Monthly Income	Approx. Fico Score	Open CREDIT Lines
0	RENT	275.00	732.0	3.0
1	RENT	1500.00	732.0	4.0
2	ANY	8333.33	732.0	4.0
3	RENT	2750.00	732.0	6.0
4	RENT	166.67	695.0	2.0

	Total CREDIT Lines	Revolving CREDIT Balance	Revolving Line Utilization
0	3.0	0.0	0.0
1	4.0	0.0	0.0
2	6.0	56.0	5.6
3	6.0	3461.0	18.6
4	2.0	0.0	0.0

	Inquiries in the Last 6 Months	Accounts Now Delinquent
0	0.0	0.0
1	0.0	0.0
2	1.0	0.0
3	10.0	0.0
4	6.0	0.0

	Delinquencies (Last 2 yrs)	Months Since Last Delinquency
0	0.0	NaN
1	0.0	0.0
2	0.0	NaN
3	0.0	0.0
4	0.0	0.0

	Public Records On File	Months Since Last Record	Employment Length
0	0.0	NaN	1.0
1	0.0	0.0	0.0
2	0.0	NaN	1.0
3	0.0	0.0	2.0
4	0.0	0.0	0.0

	Status
0	Not Delinquent
1	Not Delinquent
2	Not Delinquent
3	Not Delinquent

```
4      Delinquent
```

[3]:
```
#Mostrem la informació del dataset
print("Informació del dataset:")
print(df.info())
```

```
Informació del dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48599 entries, 0 to 48598
Data columns (total 19 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Total Amount Funded             48599 non-null  int64
 1   Loan Length                     48599 non-null  object
 2   Monthly PAYMENT                 48599 non-null  float64
 3   Debt-To-Income Ratio            48599 non-null  float64
 4   Home Ownership                  48599 non-null  object
 5   Monthly Income                  48599 non-null  float64
 6   Approx. Fico Score              48582 non-null  float64
 7   Open CREDIT Lines               48570 non-null  float64
 8   Total CREDIT Lines              48570 non-null  float64
 9   Revolving CREDIT Balance        48570 non-null  float64
 10  Revolving Line Utilization      48498 non-null  float64
 11  Inquiries in the Last 6 Months  48570 non-null  float64
 12  Accounts Now Delinquent         48570 non-null  float64
 13  Delinquencies (Last 2 yrs)      48570 non-null  float64
 14  Months Since Last Delinquency   17443 non-null  float64
 15  Public Records On File          48570 non-null  float64
 16  Months Since Last Record        3826 non-null   float64
 17  Employment Length               47267 non-null  float64
 18  Status                          48599 non-null  object
dtypes: float64(15), int64(1), object(3)
memory usage: 7.0+ MB
None
```

Com es pot observar la lectura del dataset interpreta correctament el tipus de totes les variables, pel que no és necessari aplicar transformacions de tipatge.

1.1.3 Objectiu

L'objectiu d'aquest projecte d'anàlisi de dades és el següent:

- Estudiar com es relacionen les variables del dataset amb la del prèstec, per poder determinar en última instància si aquest **serà fraudulent** o no.

L'empresa que proporciona els prèstecs pot utilitzar aquesta informació per a decidir si acceptar o no un préstec a una determinada petició. Per tant, és important ja que aporta un gran valor de negoci al categoritzar si el préstec serà tornat o no.

Aquest objectiu es pot resoldre a través de l'anàlisi de les variables que formen el dataset, per tal

de construir un model estadístic bàsic que respongui si un préstec pot resultar fraudulent o no.

Per respondre a la pregunta, aquesta serà descomposta en diverses preguntes que la reforcin i que es puguin respondre a través d'eines d'estadística bàsica (contrast d'hipòtesis, correlacions, regressions...).

En els següents apartats s'aplicarà un procés d'anàlisi previ per tal d'observar com aquestes influeixen a la pregunta objectiva, així com aplicar les transformacions necessàries per a aconseguir que l'anàlisi sigui estadísticament correcte.

1.2 Integració i selecció de les dades a analitzar

No es contempla cap integració o fusió de dades, ja que el dataset és contingut en una sola font de dades.

1.2.1 Anàlisi d'integritat

En primer lloc, es realitza una anàlisi d'integritat, per tal de determinar si les variables contenen errors en el format de les seves dades.

Es vol comprovar que les variables de tipus ordinal (aquelles que són numèriques, però representen una relació d'ordre), siguin efectivament nombres sencers completament. És a dir, que no apareguin nombres decimals.

En primer lloc, es comprova que la columna del pagament mensual sí que conté decimals, ja que es tracta d'un import:

[4]:
```
# Funció que determina si una columna té valors decimals o no
def columnHasDecimals(series):
    return series.where(lambda x: x%1 != 0).any()

df[['Monthly PAYMENT']].apply(lambda x: columnHasDecimals(x))
```

[4]:
```
Monthly PAYMENT    True
dtype: bool
```

Un cop s'ha fet la comprovació, es determina si alguna de les columnes que són del tipus float, però que haurien de ser ordinals presenten decimals.

[5]:
```
# Comprovació de les columnes que són float però que haurien de contenir sols␣
 ↪nombres sencers
columnsToCheck = ['Open CREDIT Lines',\
                  'Approx. Fico Score','Total CREDIT Lines','Inquiries in the␣
 ↪Last 6 Months',\
                  'Accounts Now Delinquent','Months Since Last␣
 ↪Delinquency','Delinquencies (Last 2 yrs)',\
                  'Months Since Last Delinquency','Public Records On␣
 ↪File','Months Since Last Record','Employment Length']
df[columnsToCheck].apply(lambda x: columnHasDecimals(x))
```

[5]:
```
Open CREDIT Lines                 False
Approx. Fico Score                False
Total CREDIT Lines                False
Inquiries in the Last 6 Months    False
Accounts Now Delinquent           False
Months Since Last Delinquency     False
Delinquencies (Last 2 yrs)        False
Months Since Last Delinquency     False
Public Records On File            False
Months Since Last Record          False
Employment Length                 False
dtype: bool
```

Cap columna presenta aquest problema. Per tant, les dades són sintàcticament correctes.

1.2.2 Anàlisi estadístic bàsic

En aquesta secció s'analitza la distribució de les dades així com la seva relació amb la variable objectiva. A partir d'aquesta anàlisi es pretén conèixer quines variables són les més importants i quines es poden descartar.

En primer lloc es mostra una taula resum dels estadístics bàsics de les variables que servirà per a enfocar les anàlisis posteriors.

[6]:
```
# Taula resum d'estadístics
df.describe()
```

[6]:
```
       Total Amount Funded  Monthly PAYMENT  Debt-To-Income Ratio  \
count         48599.000000     48599.000000          48599.000000
mean          11178.381757       334.852678             13.396254
std            7355.086203       216.699645              6.661092
min             500.000000        15.670000              0.000000
25%            5500.000000       170.690000              8.320000
50%           10000.000000       291.140000             13.490000
75%           15000.000000       449.320000             18.630000
max           35000.000000      1337.760000             29.990000

       Monthly Income  Approx. Fico Score  Open CREDIT Lines  \
count    48599.000000        48582.000000       48570.000000
mean      5806.618670          715.748137           9.379741
std       5544.183662           35.984118           4.445182
min         -0.080000          650.000000           1.000000
25%       3375.000000          695.000000           6.000000
50%       4916.670000          695.000000           9.000000
75%       6916.335000          732.000000          12.000000
max     500000.000000          790.000000          47.000000

       Total CREDIT Lines  Revolving CREDIT Balance  \
```

count	48570.000000	4.857000e+04
mean	22.121021	1.435322e+04
std	11.459323	2.113321e+04
min	1.000000	0.000000e+00
25%	14.000000	3.949250e+03
50%	20.000000	9.253000e+03
75%	29.000000	1.753575e+04
max	90.000000	1.207359e+06

	Revolving Line Utilization	Inquiries in the Last 6 Months
count	48498.000000	48570.000000
mean	50.108505	1.051822
std	28.156790	1.474013
min	0.000000	0.000000
25%	27.100000	0.000000
50%	51.300000	1.000000
75%	73.500000	2.000000
max	119.000000	33.000000

	Accounts Now Delinquent	Delinquencies (Last 2 yrs)
count	48570.000000	48570.000000
mean	0.000082	0.147601
std	0.009075	0.503296
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	1.000000	13.000000

	Months Since Last Delinquency	Public Records On File
count	17443.000000	48570.000000
mean	35.414321	0.054602
std	22.330053	0.238262
min	0.000000	0.000000
25%	17.000000	0.000000
50%	33.000000	0.000000
75%	52.000000	0.000000
max	120.000000	5.000000

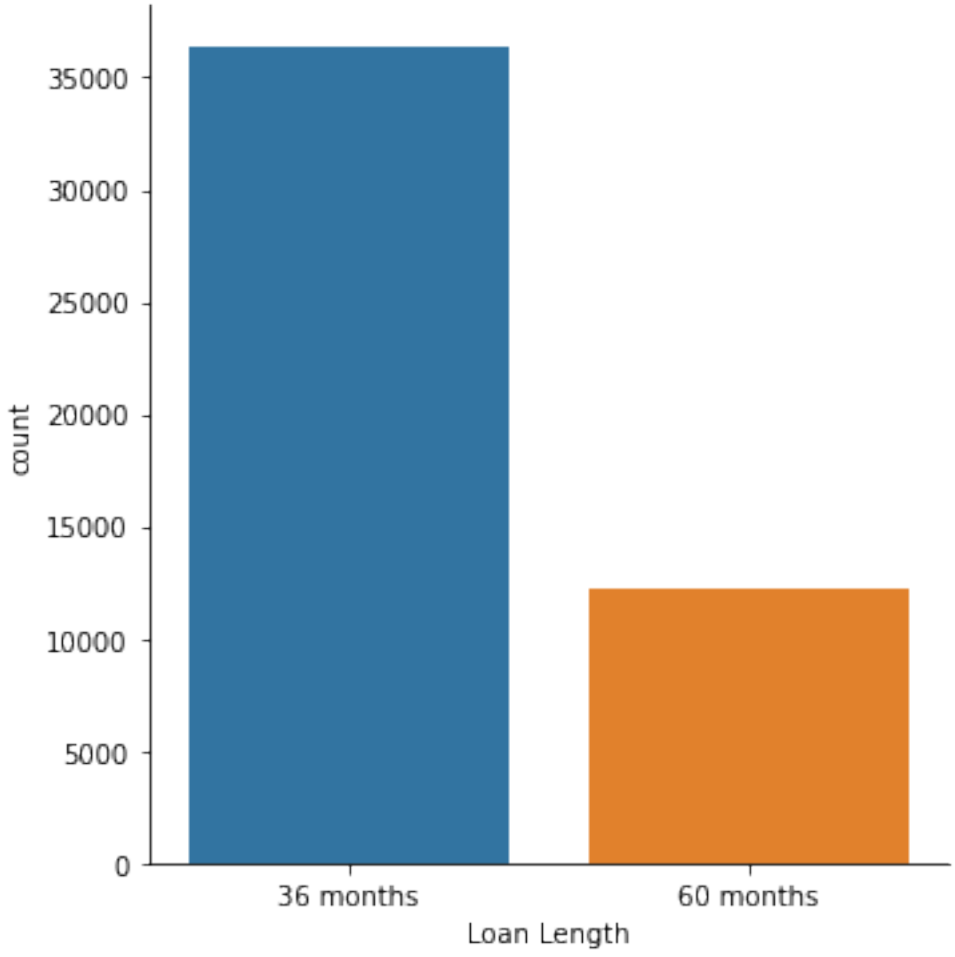
	Months Since Last Record	Employment Length
count	3826.000000	47267.000000
mean	60.824621	4.982927
std	46.871353	3.568941
min	0.000000	0.000000
25%	0.000000	2.000000
50%	86.000000	4.000000
75%	102.000000	9.000000

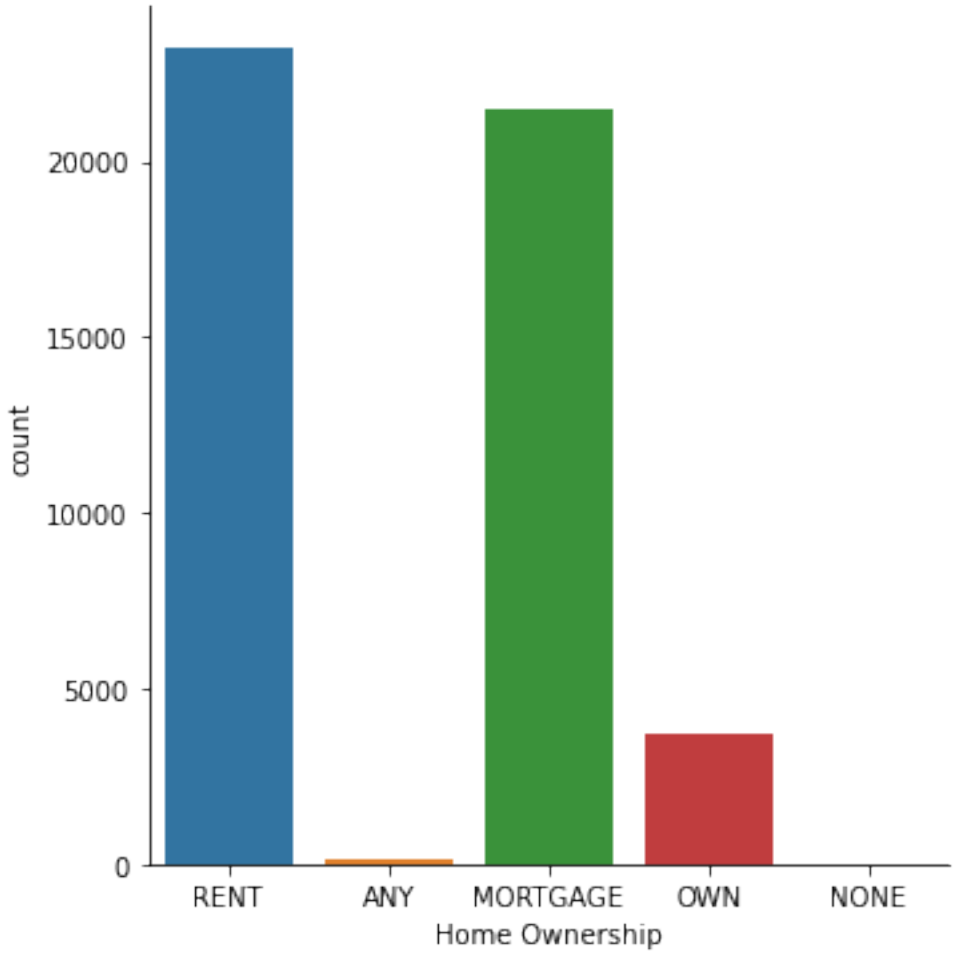
```
max                  129.000000           10.000000
```

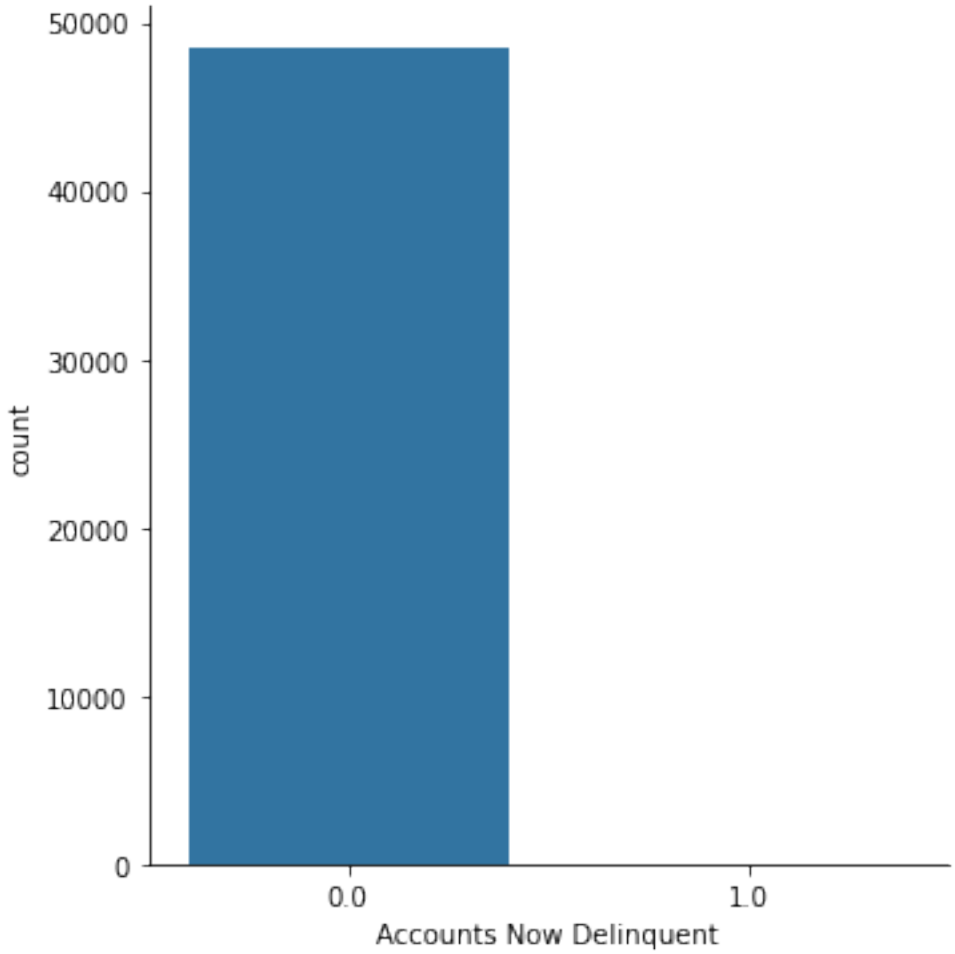
Un fet que es pot remarcar de seguida és que la variable **Accounts Now Delinquent** és categòrica, ja que només conté els valors 0 i 1, encara que ha sigut codificada com a float.

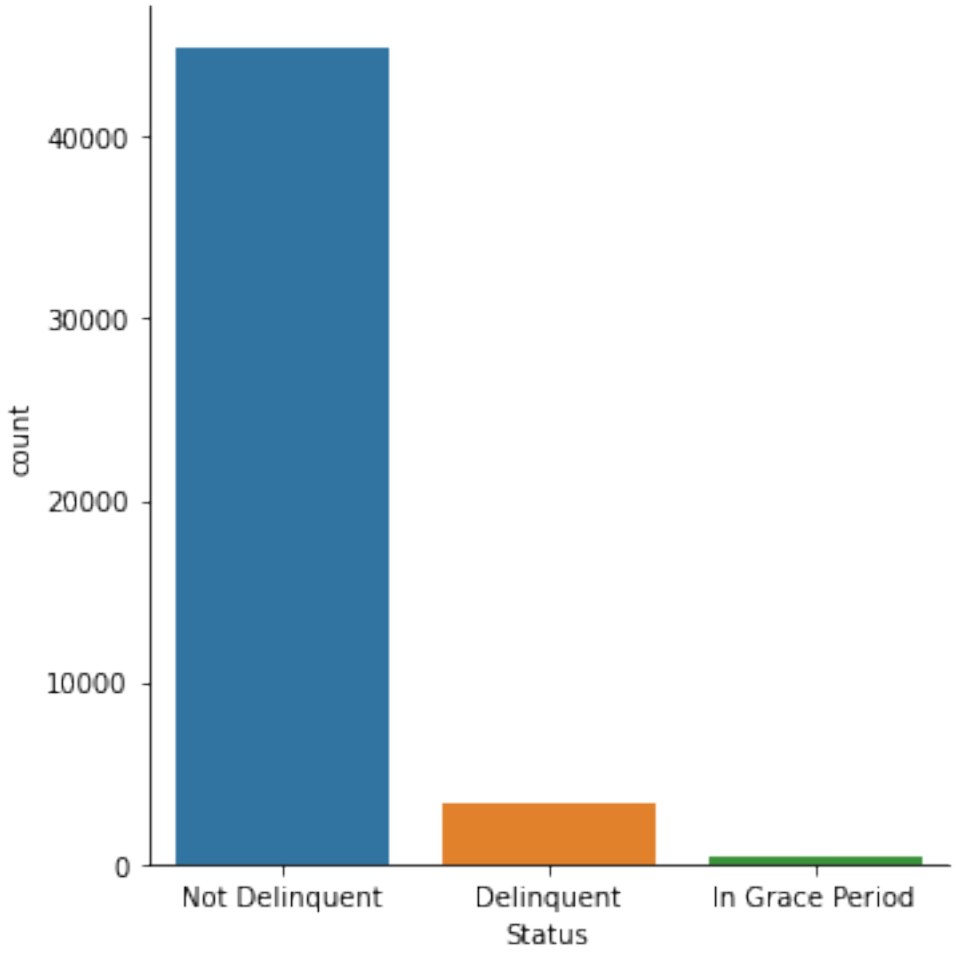
Variables categòriques En segon lloc, s'analitzen les variables categòriques. Interessa conèixer la distribució de les diferents categories i la relació de la variable amb la morositat del préstec. S'observen les freqüències de cada classe:

[7]:
```
sns.catplot(x='Loan Length', data=df, kind='count');
sns.catplot(x='Home Ownership', data=df, kind='count');
sns.catplot(x='Accounts Now Delinquent', data=df, kind='count');
sns.catplot(x='Status', data=df, kind='count');
```




20000
15000
10000
5000
0
count
RENT
ANY
MORTGAGE
OWN
NONE
Home Ownership


50000
40000
30000
20000
10000
0
count
0.0
1.0
Accounts Now Delinquent



Es pot observar que:

1. Els mesos del préstec no presenten cap característica significativa.
2. Els valors ANY i NONE per a la propietat tenen una freqüència molt baixa respecte als altres valors.
3. Pràcticament cap fila de dades presenta la propietat d'accounts now delinquent.
4. Existeix un valor de l'etiqueta del préstec que és període de gràcia, molt poc representatiu.

La categoria de **període de gràcia** és molt poc representativa en el dataset i significa que la decisió sobre si el préstec és o no morós es delega per més endavant.

Per tant, aquesta categoria no aporta informació, ja que pot ser d'un grup o d'un altre. Donada la seva baixa representació, es decideix eliminar totes les files del dataset que tinguin com a etiqueta de préstec aquesta categoria.

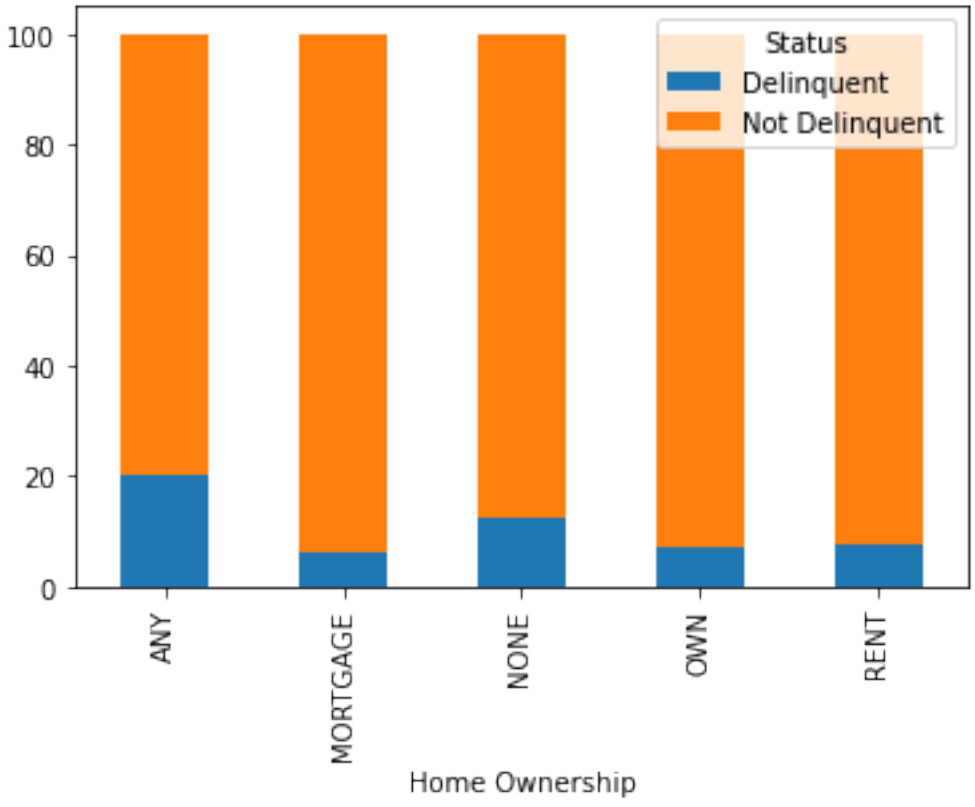
[8]:
```
df_nograce = df[df.Status != 'In Grace Period']
# Veure les noves categories
df_nograce.Status.unique()
```

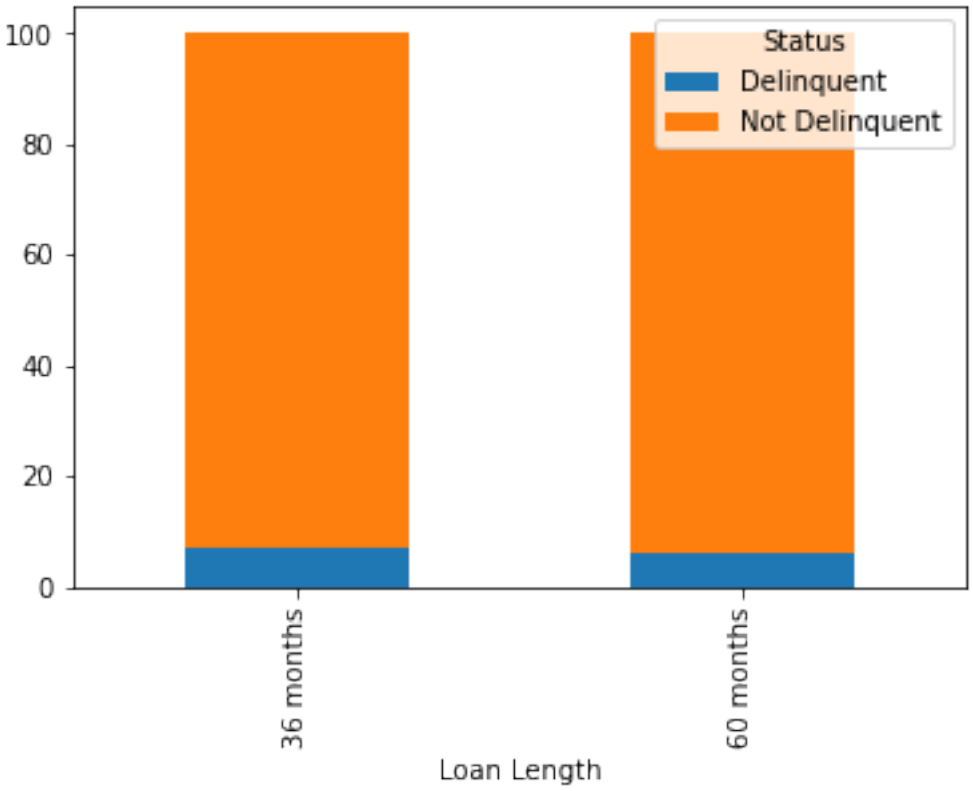
[8]:
```
array(['Not Delinquent', 'Delinquent'], dtype=object)
```

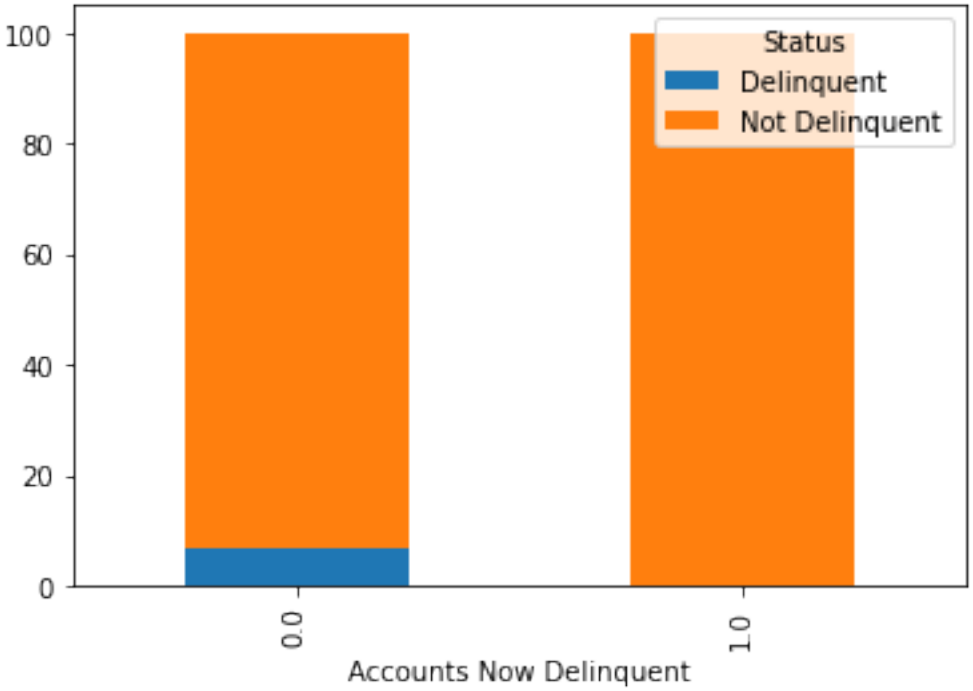
A continuació es visualitzen les possibles correlacions entre les altres variables categòriques i la variable del crèdit. Per fer-ho, s'utilitza un gràfic de barres apilades per percentatge. Aquest gràfic mostra la proporció de les categories de la morositat del crèdit per cada classe de la variable d'interès.

Si existeix una correlació, les proporcions seran diferents.

[9]:
```
# Gràfic de barres apilades utilitzant la taula de contingència
def stackedBarPlot(df, column1, column2):
    cross = pd.crosstab(df[column1], df[column2], margins = False)
    applied = cross.apply(lambda x: x*100/sum(x), axis=1)
    applied.plot(kind="bar", stacked=True)

stackedBarPlot(df_nograce, 'Home Ownership', 'Status')
stackedBarPlot(df_nograce, 'Loan Length', 'Status')
stackedBarPlot(df_nograce, 'Accounts Now Delinquent', 'Status')
```




Status
Delinquent
Not Delinquent
100
80
60
40
20
0
36 months
60 months
Loan Length


Status
Delinquent
Not Delinquent
100
80
60
40
20
0
0.0
1.0
Accounts Now Delinquent

S'observa que:

1. Existeixen variacions de la proporció a l'hora de determinar l'etiqueta del préstec en funció de la propietat del client. Les variacions poden arribar al 10%. Sembla que no tenir propietats o la categoria ANY. Cal mencionar que aquestes dues categories disposen de molt pocs individus, pel que aquestes diferències poden no ser significatives.
2. La longitud del préstec no presenta diferències tan significatives com la variable anterior. Així i tot s'han de tenir present que una diferència petita en un dataset tan gran pot prendre importància.
3. No sembla que l'última variable tingui gens d'importància encara que es presenti una diferència de proporció, ja que la classe 1 a penes presenta files en el dataset.

Per assegurar la jugada, es pot realitzar el test chi-squared d'independència de variables. Si el valor-p és inferior a 0.05 es refusa la hipòtesi nul·la i s'accepta que les variables presenten dependència.

[10]:
```
# Chi-squared tests
def find_p_value(df, column, target_column):
    crosstab = pd.crosstab(df[column], df[target_column])
    stat, p, dof, expected = chi2_contingency(crosstab)
    return {'stat':stat,'p-value':p,'column':column}

print(find_p_value(df_nograce, 'Home Ownership', 'Status'))
print(find_p_value(df_nograce, 'Loan Length', 'Status'))
print(find_p_value(df_nograce, 'Accounts Now Delinquent', 'Status'))
```

```
{'stat': 71.751267195805, 'p-value': 9.6861186528273e-15, 'column': 'Home
Ownership'}
{'stat': 14.898824188140873, 'p-value': 0.0001134336654434595, 'column': 'Loan
Length'}
{'stat': 0.20254490541178188, 'p-value': 0.6526744660590855, 'column': 'Accounts
Now Delinquent'}
```

Efectivament, la variable amb més influència és la propietat del client. També ho és la longitud del préstec però no en tanta mesura.

La tercera variable no rebutja la hipòtesi nul·la i per tant es pot eliminar del dataset, ja que no és rellevant per al problema.

[11]:
```
df_selectedCategorical = df_nograce.drop('Accounts Now Delinquent', axis=1)
```

Variables numèriques A continuació, es realitza el procés d'anàlisi i selecció de les variables numèriques.

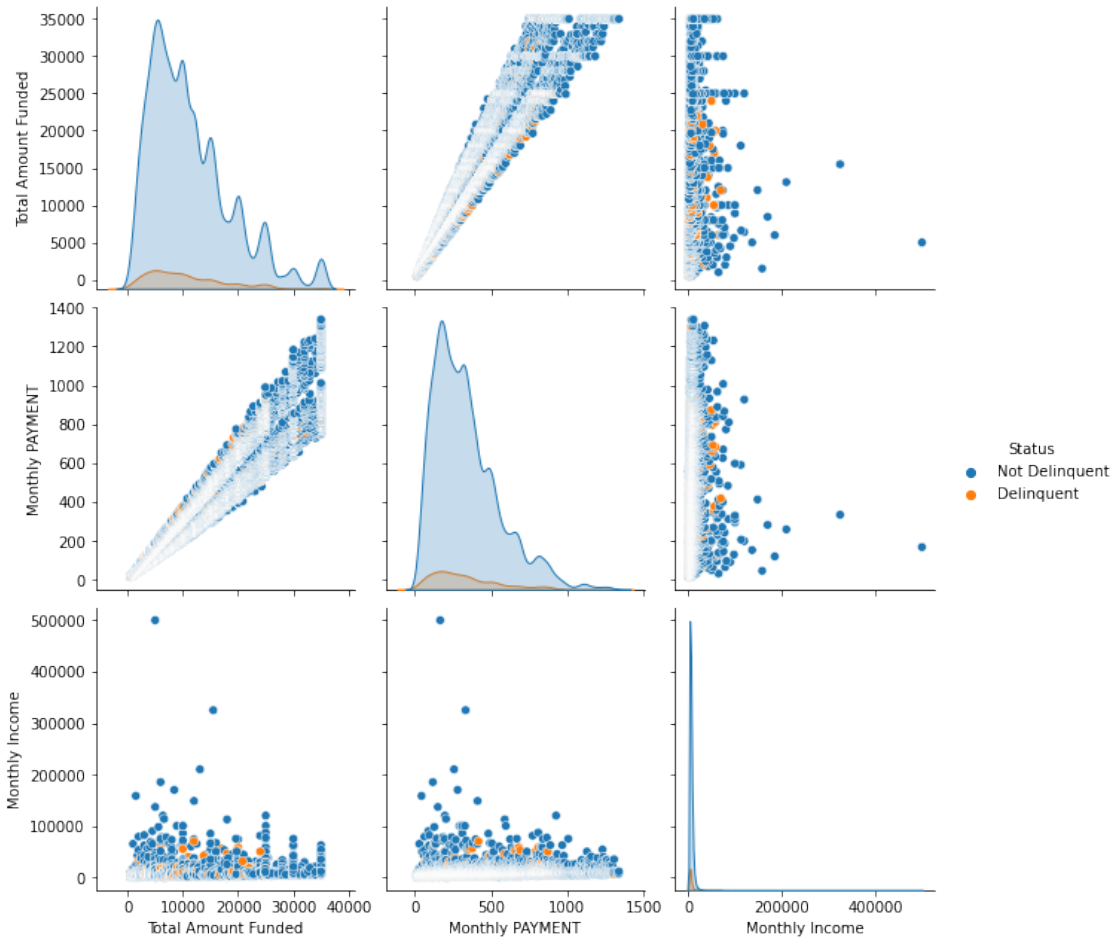
Aquest procés ha d'enfocar a trobar quines variables numèriques presenten una relació amb l'etiqueta del préstec.

En primer lloc, però, interessa conèixer quines variables numèriques són dependents entre elles. Es pot analitzar de forma visual aquesta dependència a través d'un **pair plot**. Aquesta tasca es

realitza per grups per simplificar l'anàlisi.

Addicionalment, s'aprofita per projectar l'etiqueta del préstec d'un color o un altre per tal d'observar si existeix alguna relació evident en els gràfics.

[12]:
```
payments = ['Total Amount Funded','Monthly PAYMENT','Monthly Income','Status']
df_payments = df_selectedCategorical[payments]
sns.pairplot(df_payments, hue="Status", size=3);
```



S'observa que:

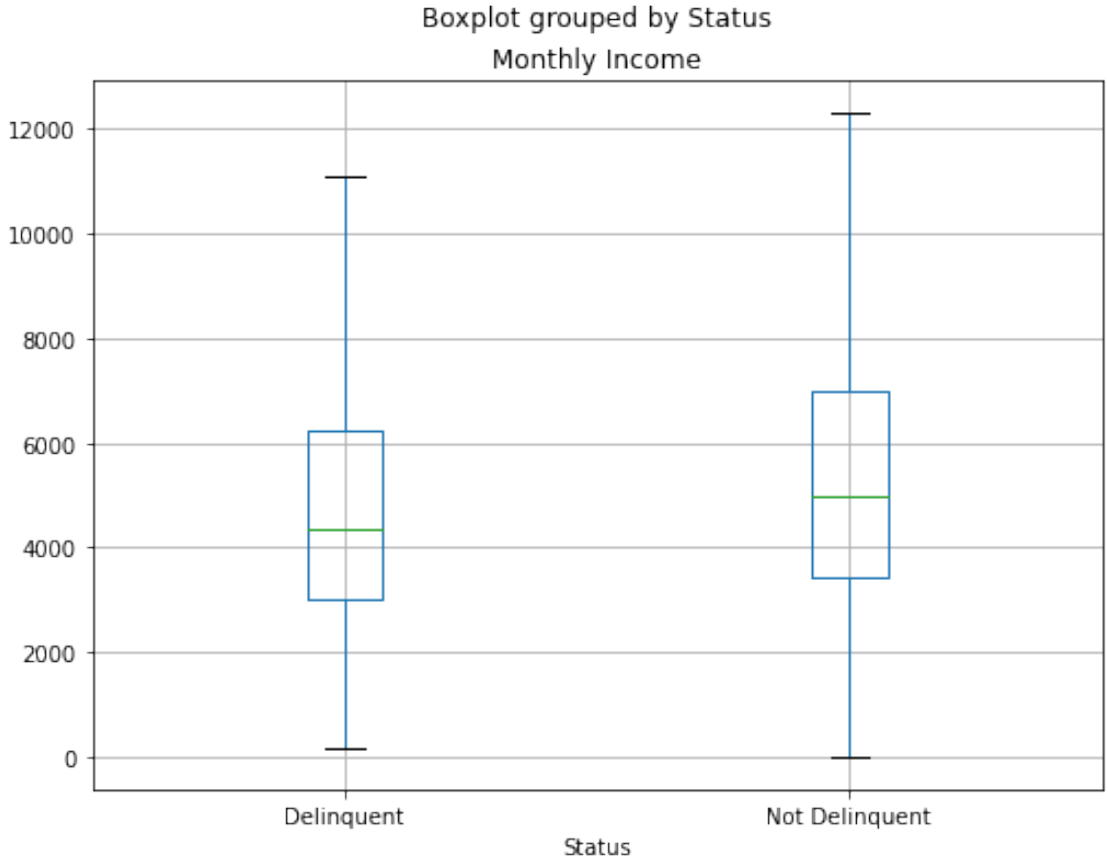
1. No existeix cap relació entre la quantitat demanada i els ingressos de la persona ja que el núvol de punts no té cap forma.
2. Existeix una clara relació lineal entre la quantitat demanada i el pagament mensual. És esperable ja que es deu calcular de forma automàtica.
3. No és evident cap relació entre la categoria del crédit i les variables estudiades, almenys de forma visual és difícil d'interpretar.

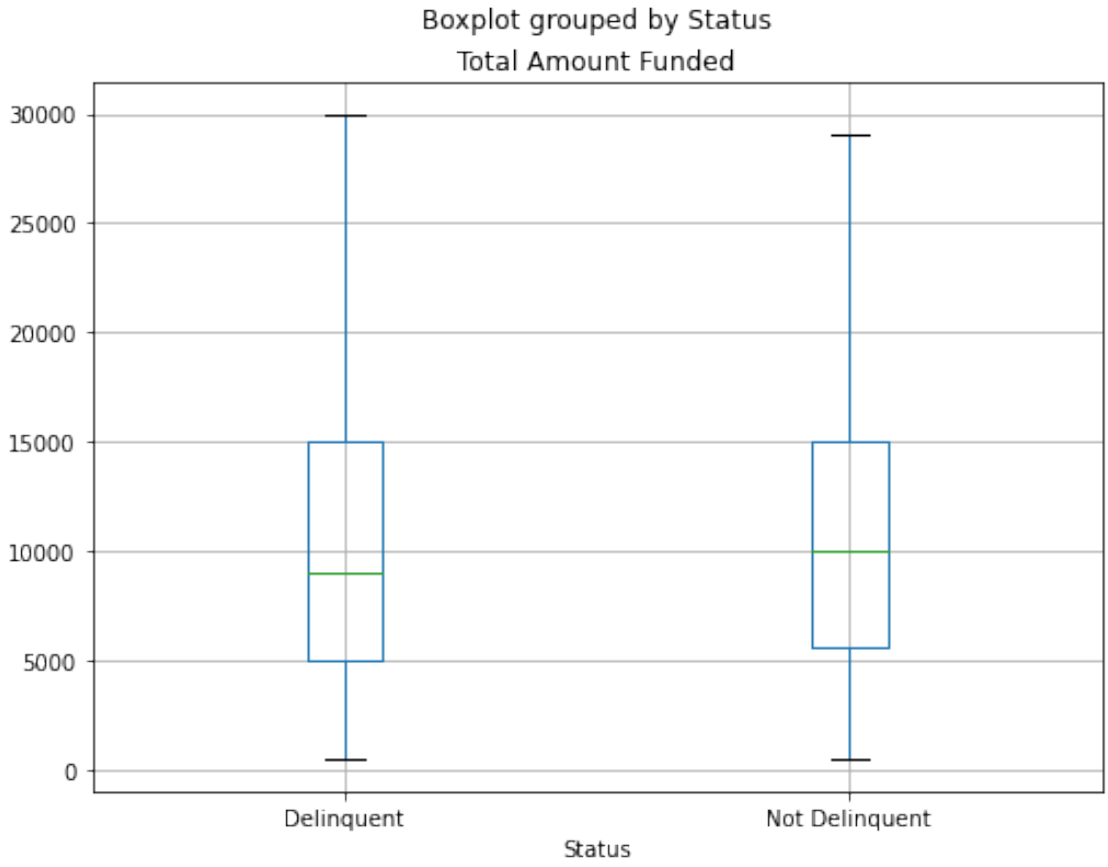
Per tal d'afinar més, es poden projectar els gràfics de caixa segons el tipus de préstec. Per a facilitar la visualització de les dades, s'ignoren els valors atípics.

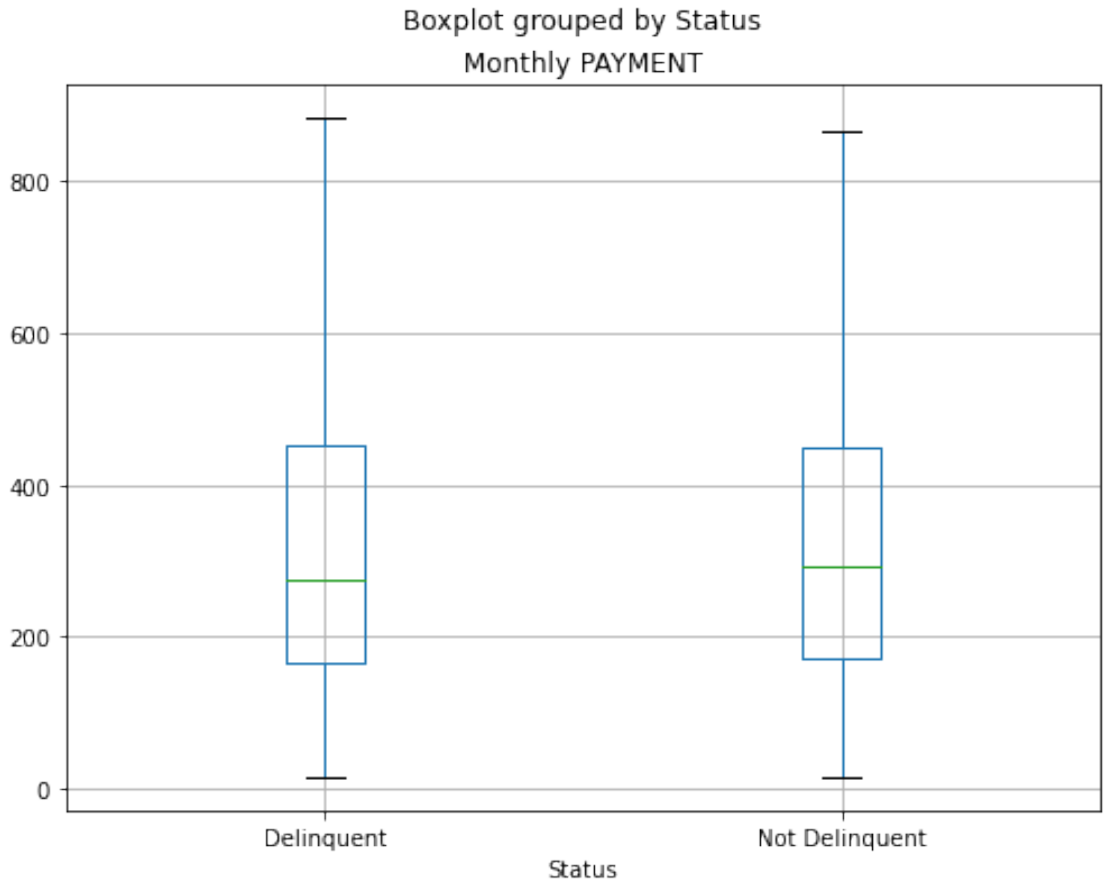
Els diagrames de caixa projecten les distribucions de les variables en funció de cada grup. Si existeix una relació entre la variable numèrica i la categòrica, les distribucions poden canviar significativament de forma (mediana desplaçada, quantils diferents...).

Per contrapartida, si els diagrames de caixa són similars, la distribució és similar i per tant no es pot afirmar que existeixi relació.

[13]:
```
# Mostrar els diagrames de caixa dels ingressos mensuals i del pagament mensual
df_selectedCategorical.boxplot('Monthly Income','Status',showfliers=False,␣
 ↪figsize=(8,6))
df_selectedCategorical.boxplot('Total Amount Funded','Status',showfliers=False,␣
 ↪figsize=(8,6))
df_selectedCategorical.boxplot('Monthly PAYMENT','Status',showfliers=False,␣
 ↪figsize=(8,6))
plt.show()
```




Boxplot grouped by Status
Total Amount Funded
30000
25000
20000
15000
10000
5000
0
Delinquent
Not Delinquent
Status



Així com la diferència entre els quantils Q1 i Q3 és evident en les etiquetes del préstec per als ingressos mensuals, no és tan clar en el pagament mensual.

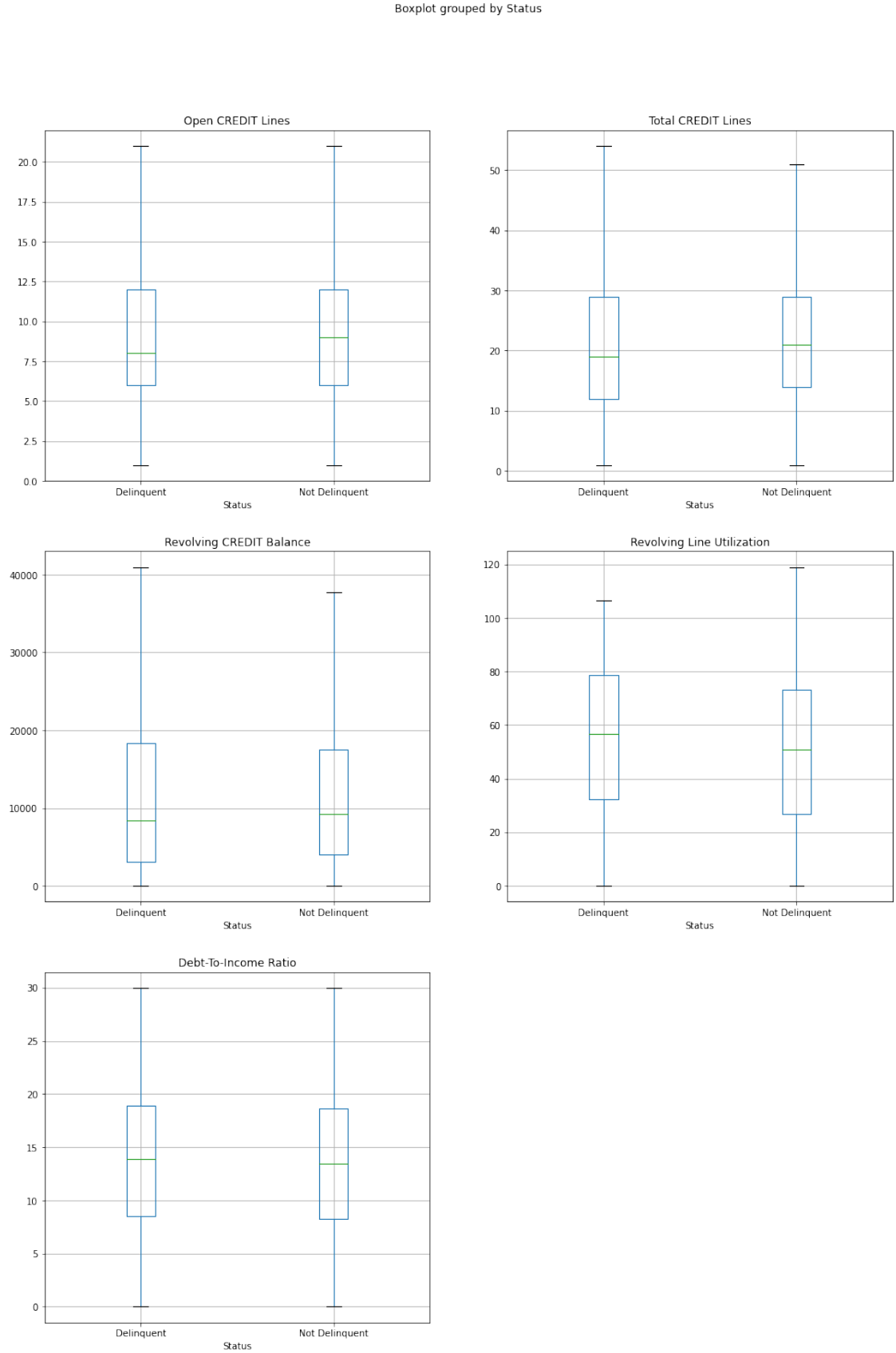
No sembla que el total del prèstec sigui diferent entre les dues categories.

Es conclou que els ingressos mensuals determinen en major importància si el préstec és o no fraudulent que no pas el pagament mensual. (que és dependent amb la quantitat del crèdit demanat). La quantitat demanda pel prèstec no influeix directament en si el préstec resulta fraudulent o no, però sí que determina el pagament mensual dins d'una forquilla de valors.

S'analitzen ara les altres variables numèriques.

[14]:
```
# Mostrar els diagrames de caixa dels ingressos mensuals i del pagament mensual
fig, axes = plt.subplots(3,2, figsize=(16,24))
df_selectedCategorical.boxplot('Open CREDIT␣
 ↪Lines','Status',ax=axes[0][0],showfliers=False, figsize=(8,6))
df_selectedCategorical.boxplot('Total CREDIT␣
 ↪Lines','Status',ax=axes[0][1],showfliers=False, figsize=(8,6))
df_selectedCategorical.boxplot('Revolving CREDIT␣
 ↪Balance','Status',ax=axes[1][0],showfliers=False, figsize=(8,6))
```

```
df_selectedCategorical.boxplot('Revolving Line␣
 ↪Utilization','Status',ax=axes[1][1],showfliers=False, figsize=(8,6))
df_selectedCategorical.boxplot('Debt-To-Income␣
 ↪Ratio','Status',ax=axes[2][0],showfliers=False, figsize=(8,6))
axes[2][1].set_axis_off()
plt.show()
```


Boxplot grouped by Status
Open CREDIT Lines
Total CREDIT Lines
Revolving CREDIT Balance
Revolving Line Utilization
Debt-To-Income Ratio
Delinquent
Not Delinquent
Status

Aquestes gràfiques denoten que:

1. La mediana de OPEN Credit lines és més baixa en el cas que el préstec s'etiqueti com a fraudulent. És a dir, és possible que els clients que cometen frau obrin menys línies de crédit. La diferència entre les dues distribucions només existeix en la mediana, però, ja que les caixes són essencialment iguals en els altres quantils.
2. Aquesta diferència també es pot observar a TOTAL Credit lines, que segurament es pugui expressar com a funció de les línies de crèdit obertes.
3. El balanç de les línies del crèdit rotatiu presenta una petita diferència en la distribució segons els grups.
4. L'ús del crèdit rotatiu pel client sí que és bastant significatiu segons els grups, ja que les caixes apareixen desplaçades.
5. La ràtio deute-ingressos no és diferent segons els grups.

Variables Months Since Last Delinquency | Last Record Aquestes dues variables requereixen un tractament especial. Ambdues contenen una gran quantitat de valors perduts que s'han d'interpretar abans d'analitzar la seva relació amb la variable d'interès.

[15]:
```
# Months Since Last Delinquency
df_selectedCategorical['Months Since Last Delinquency'].isnull().value_counts()
```

[15]:
```
True     30922
False    17294
Name: Months Since Last Delinquency, dtype: int64
```

[16]:
```
# Months Since Last Record
df_selectedCategorical['Months Since Last Record'].isnull().value_counts()
```

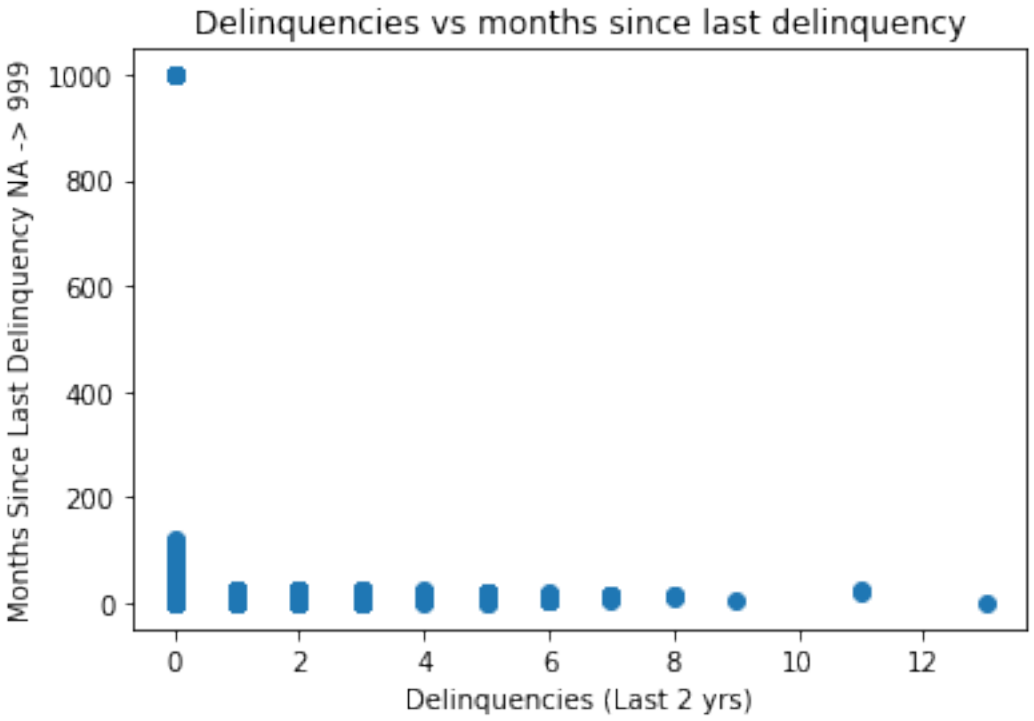
[16]:
```
True     44408
False     3808
Name: Months Since Last Record, dtype: int64
```

Es sospita que els valors perduts d'aquestes variables no són perduts de forma accidental, sinó que es tracten de valors que signifiquen **que no s'ha comés mai l'acció indicada**. És a dir, si el client no ha comés mai cap acte fraudulent i no té cap registre, aquests valors seràn **NA**.

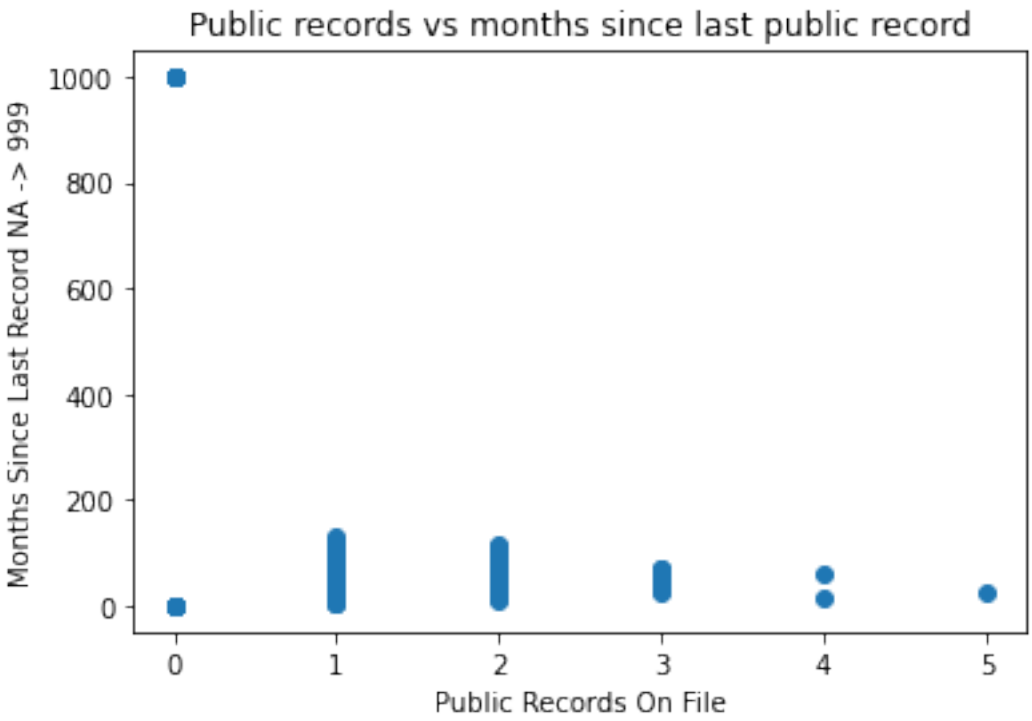
Aquesta teoria es pot comprovar a través de la visualització entre la variable "Delinquencies (Last 2 yrs)" i la de mesos des de l'últim impagament. Per l'altra banda, també es pot comprovar la variable de mesos des de l'últim registre amb "Public Records On File".

[17]:
```
# Crear una nova variable que sigui els mesos des de l'últim impagament sense␣
 ↪els NA, s'imputen els valors 999 per dierenciar-los de la resta
df_selectedCategorical['Months Since Last Delinquency NoNA'] =␣
 ↪df_selectedCategorical['Months Since Last Delinquency'].fillna(999)
plt.scatter(df_selectedCategorical['Delinquencies (Last 2 yrs)'],␣
 ↪df_selectedCategorical['Months Since Last Delinquency NoNA'])
```

```
plt.title("Delinquencies vs months since last delinquency")
plt.xlabel("Delinquencies (Last 2 yrs)")
plt.ylabel("Months Since Last Delinquency NA -> 999")
plt.show()
```



[18]:
```
# Crear una nova variable que sigui els mesos des de l'últim impagament sense␣
 ↪els NA, s'imputen els valors 999 per dierenciar-los de la resta
df_selectedCategorical['Months Since Last Record NoNA'] =␣
 ↪df_selectedCategorical['Months Since Last Record'].fillna(999)
plt.scatter(df_selectedCategorical['Public Records On File'],␣
 ↪df_selectedCategorical['Months Since Last Record NoNA'])
plt.title("Public records vs months since last public record")
plt.xlabel("Public Records On File")
plt.ylabel("Months Since Last Record NA -> 999")
plt.show()
```



[19]:
```
# També es pot comprovar mirant els valors únics que tenen les files filtrades␣
↪pels valors NA
df_selectedCategorical[df_selectedCategorical['Months Since Last Delinquency␣
↪NoNA'] == 999]['Delinquencies (Last 2 yrs)'].value_counts()
```

[19]:
```
0.0    30893
Name: Delinquencies (Last 2 yrs), dtype: int64
```

[20]:
```
df_selectedCategorical[df_selectedCategorical['Months Since Last Record NoNA']␣
↪== 999]['Public Records On File'].value_counts()
```

[20]:
```
0.0    44379
Name: Public Records On File, dtype: int64
```

En definitiva, totes les variables "Months since..." són NULL quan representa que el client no ha comés l'acció explicativa.

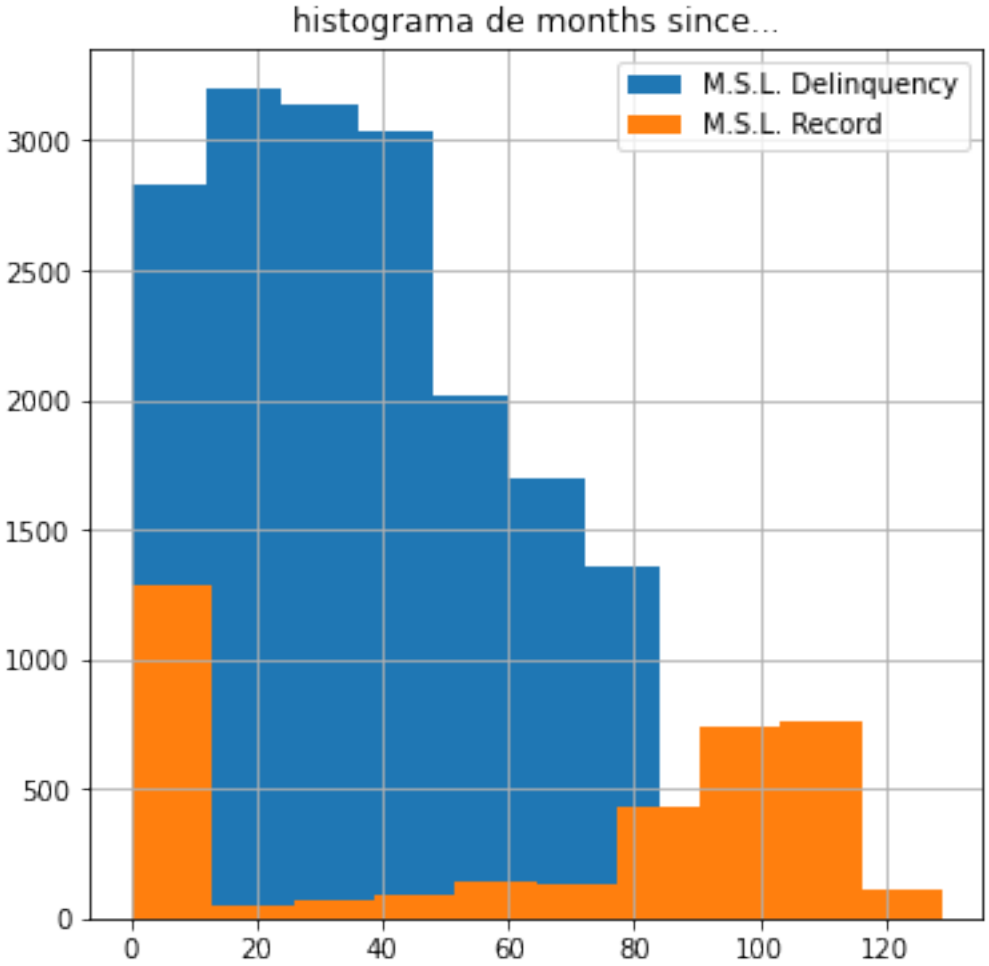
Per tal de poder utilitzar la columna en posteriors anàlisis, s'ha de determinar una estratègia d'imputació. Existeixen les següents variants:

1. Imputar els valors perduts amb un nombre suficientment gran. No és adequat, ja que el nombre pot causar bias.
2. Categoritzar les variables en noves que mantinguin la quantitat més gran d'informació possible i incorporin els valors perduts com a una nova categoria.

S'escull la segona opció. Aquesta serà tractada amb més profunditat a l'etapa d'**imputació de**

valors perduts. De moment, s'imputen els valors de forma intuitiva a partir de la distribució de les variables:

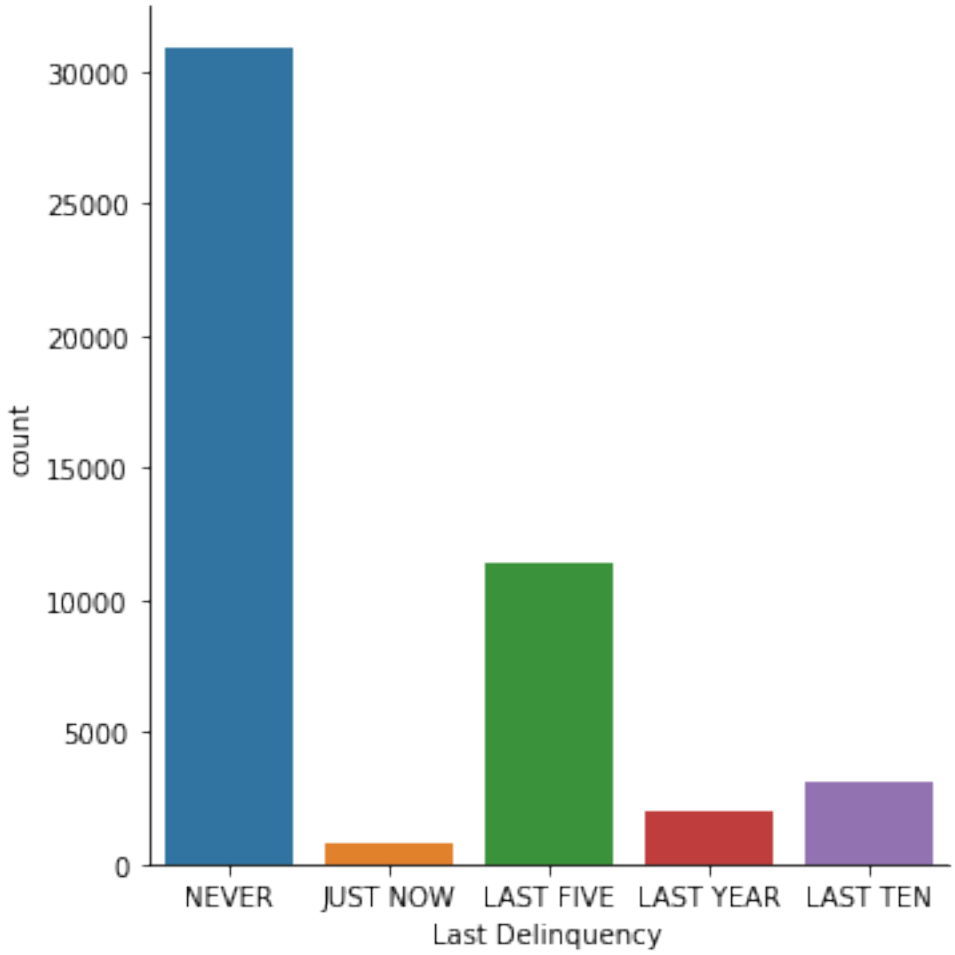
[21]:
```
fig = plt.figure(figsize=(6,6))
df_selectedCategorical['Months Since Last Delinquency'].hist(label="M.S.L.␣
↪Delinquency")
df_selectedCategorical['Months Since Last Record'].hist(label="M.S.L. Record")
plt.legend()
plt.title("histograma de months since...")
plt.show()
```

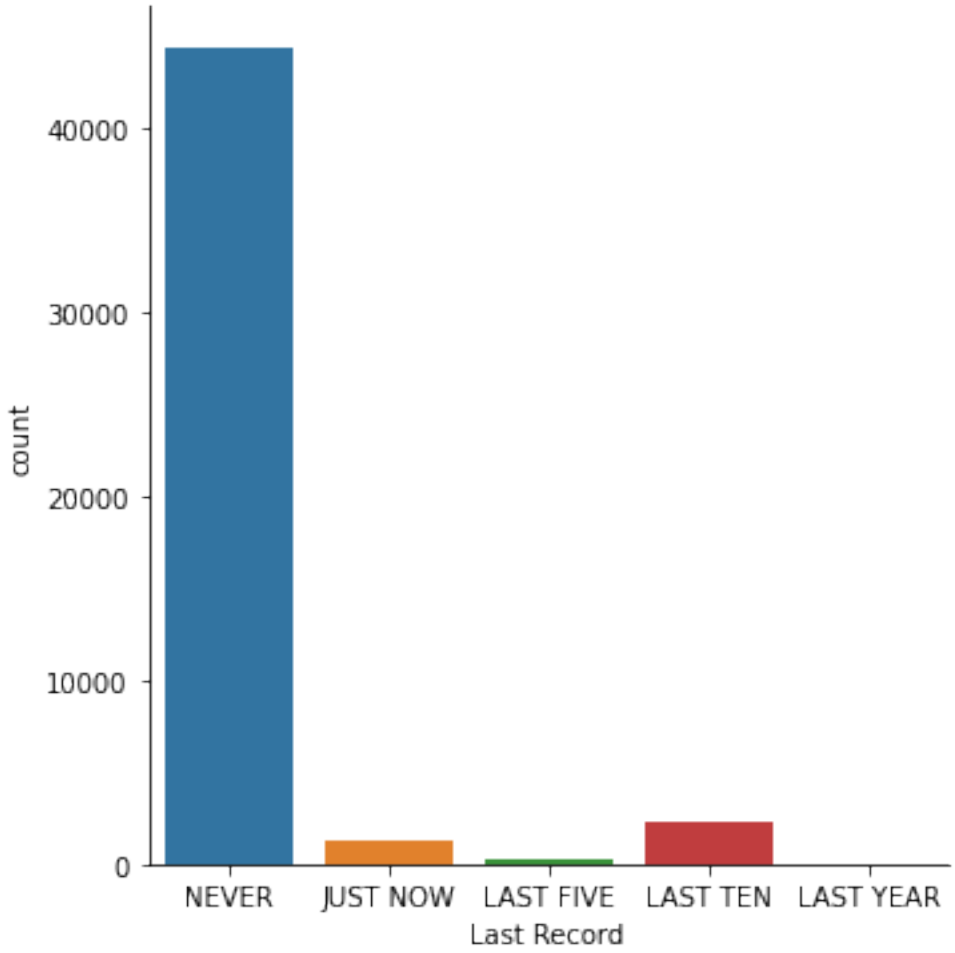


El criteri de les categories és el següent:

1. JUST NOW: 0 mesos.
2. LAST YEAR: 1 a 12 mesos.
3. LAST FIVE: 1 a 5 anys.
4. LAST TEN: 5 a 10 anys.
5. NEVER: No presenta incidència.

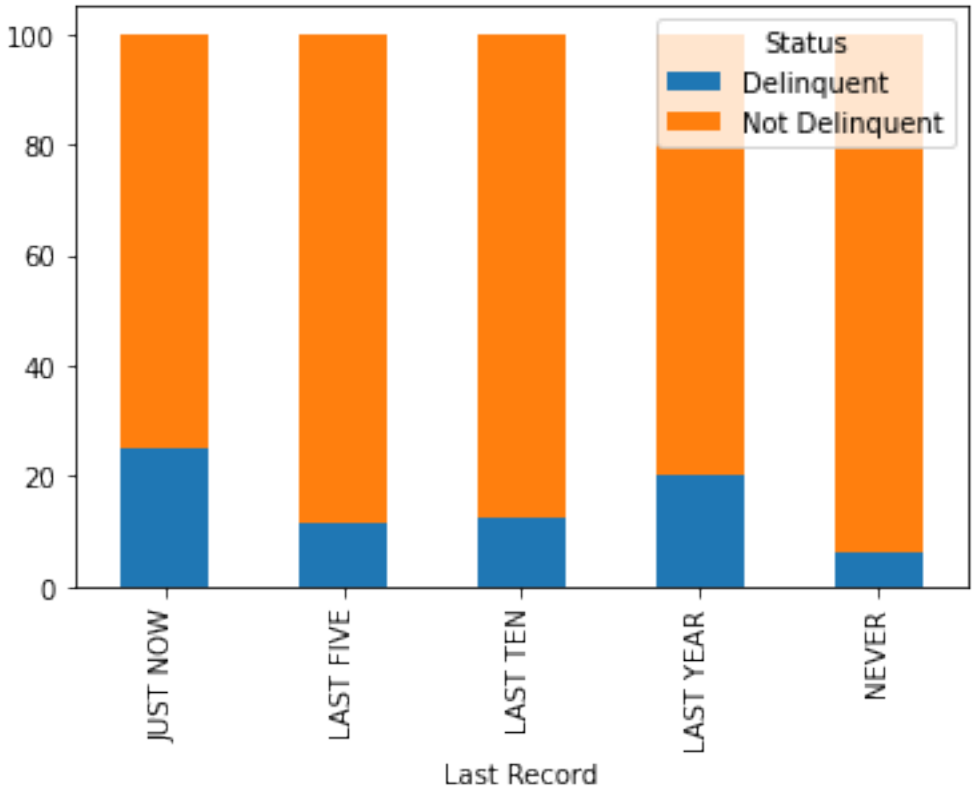
[22]:
```
def mapMonthsSinceValues(x):
    if (x == 0): return 'JUST NOW'
    elif (x > 0 and x < 12): return 'LAST YEAR'
    elif (x >= 12 and x < 60): return 'LAST FIVE'
    elif (x >= 60): return 'LAST TEN'
    else: return 'NEVER'

def inputMissingsMonthsSince(df, column, new_column):
    df[new_column] = df[column].map(mapMonthsSinceValues)

# Imputació dels valors en una nova variable categòrica
inputMissingsMonthsSince(df_selectedCategorical,'Months Since Last Delinquency',␣
 ↪'Last Delinquency')
inputMissingsMonthsSince(df_selectedCategorical,'Months Since Last Record',␣
 ↪'Last Record')

# Veure la distribució de les noves variables
sns.catplot(x='Last Delinquency', data=df_selectedCategorical, kind='count')
sns.catplot(x='Last Record', data=df_selectedCategorical, kind='count')
plt.show()
```

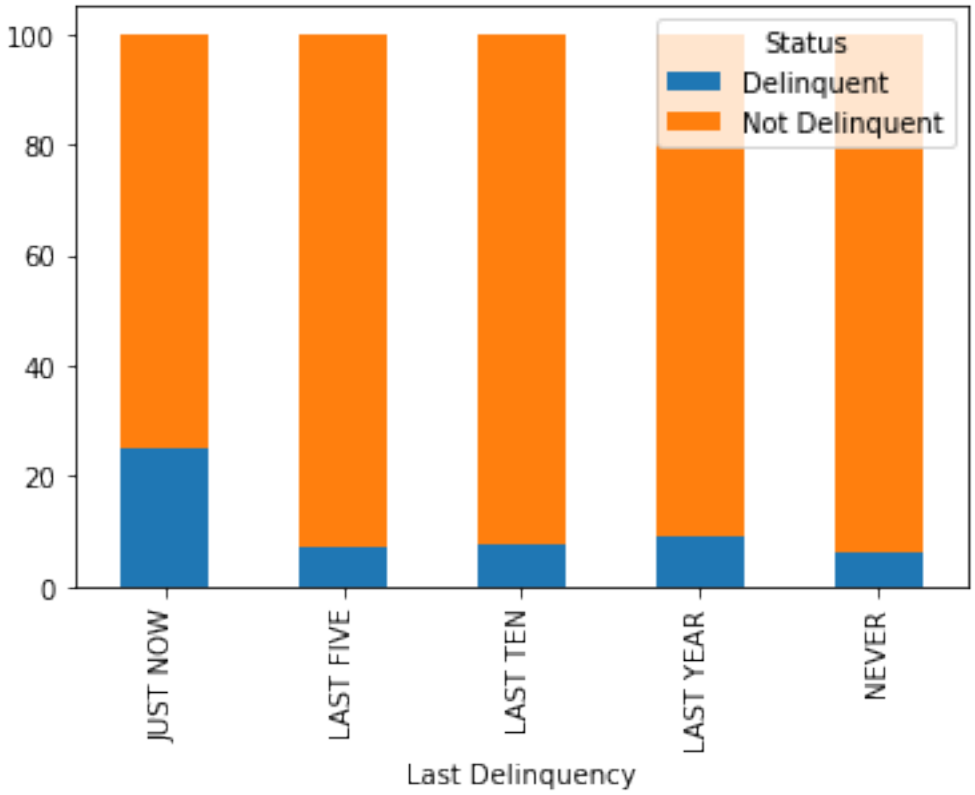

30000
25000
20000
15000
10000
5000
0
count
NEVER
JUST NOW
LAST FIVE
LAST YEAR
LAST TEN
Last Delinquency



També es poden observar les proporcions d'aquestes noves variables i realitzar la prova chi-squared.

[23]:
```
# Gràfic de proporcions segons la categoria
stackedBarPlot(df_selectedCategorical, 'Last Record', 'Status')
stackedBarPlot(df_selectedCategorical, 'Last Delinquency', 'Status')
```


Status
Delinquent
Not Delinquent
100
80
60
40
20
0
JUST NOW
LAST FIVE
LAST TEN
LAST YEAR
NEVER
Last Record



[24]:
```
# Chi-squared test
print(find_p_value(df_selectedCategorical, 'Last Record', 'Status'))
print(find_p_value(df_selectedCategorical, 'Last Delinquency', 'Status'))
```

```
{'stat': 820.9217272731001, 'p-value': 2.2565263992124012e-176, 'column': 'Last
Record'}
{'stat': 479.58627733876153, 'p-value': 1.7410337310816523e-102, 'column': 'Last
Delinquency'}
```

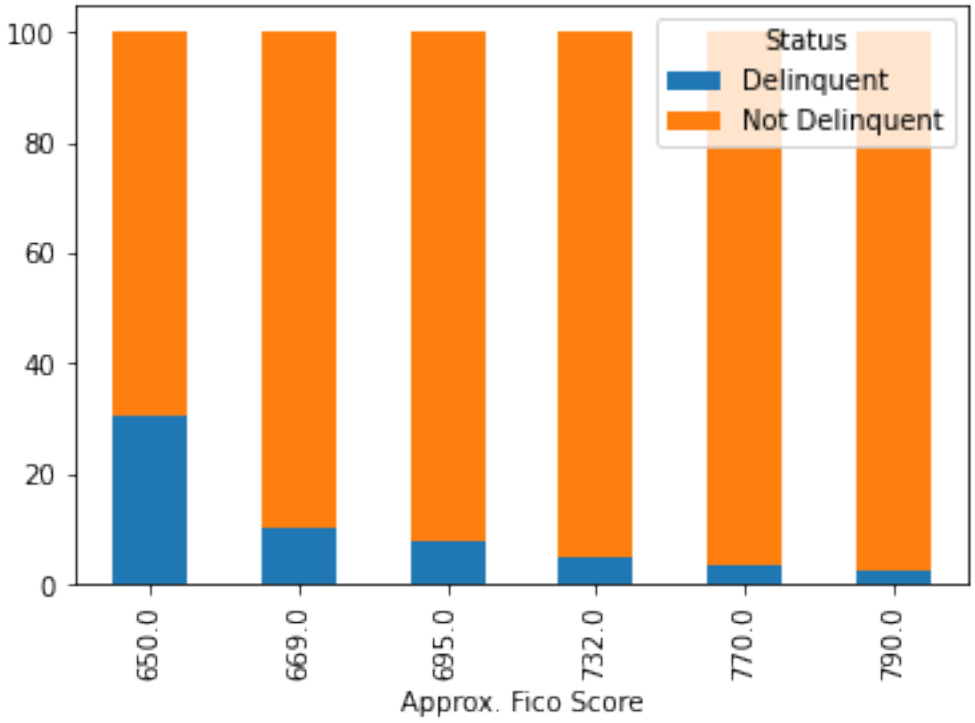
Efectivament s'observa que les dues variables tenen relació amb l'estat del crèdit i que existeix una relació d'ordre en la proporció de frau. Sembla que a **més temps** de l'última infracció, menys es tendeix a classificar el crèdit com a fraudulent.

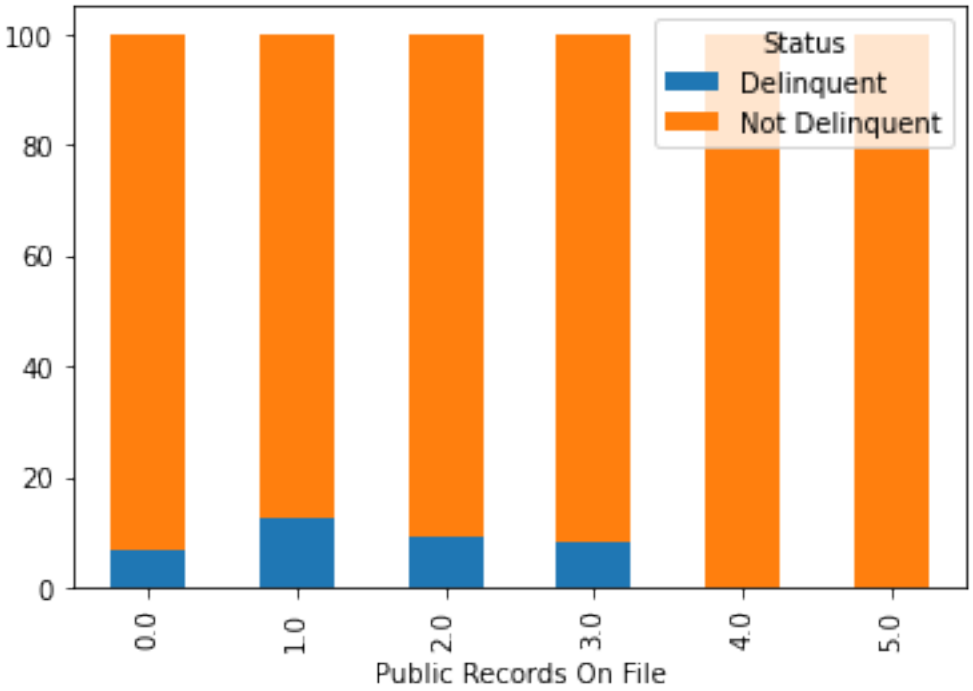
Variables ordinals Tot seguit s'analitzen les variables de tipus ordinal, que són aquelles que es poden interpretar com categòriques amb categories numèriques, i que tenen una relació d'ordre.

L'objectiu d'aquest apartat és poder assegurar quines variables de tipus ordinal influeixen en l'etiqueta del préstec de forma lineal.

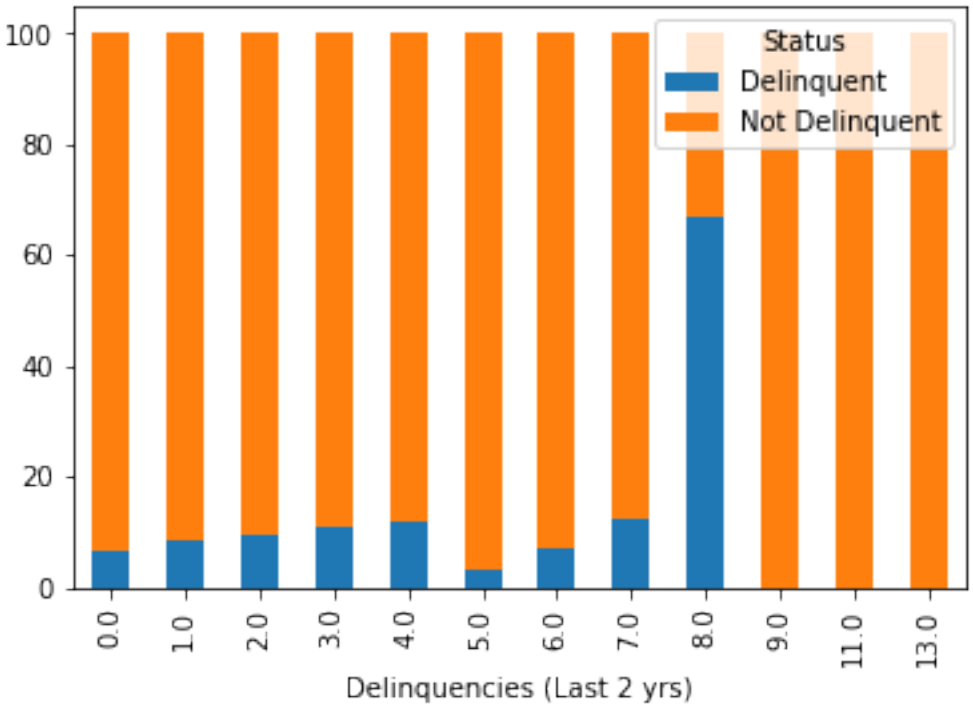
Per visualitzar aquesta informació, es pot utilitzar el mateix gràfic de categories apilades que s'ha fet servir anteriorment.

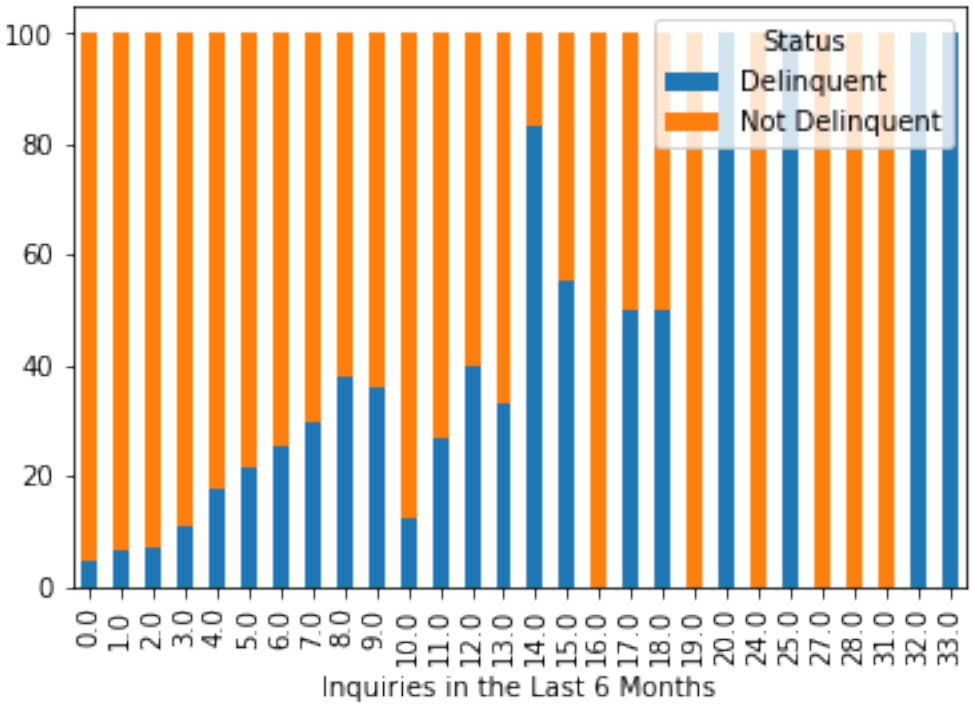
[25]:
```
ordinal = ['Approx. Fico Score','Public Records On File','Employment␣
 ↪Length','Delinquencies (Last 2 yrs)','Inquiries in the Last 6 Months']

for var in ordinal:
    stackedBarPlot(df_selectedCategorical, var, 'Status')

plt.show()
```




Status
Delinquent
Not Delinquent
100
80
60
40
20
0
0.0
1.0
2.0
3.0
4.0
5.0
Public Records On File


Status
Delinquent
Not Delinquent
100
80
60
40
20
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
Employment Length


Status
Delinquent
Not Delinquent
100
80
60
40
20
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
11.0
13.0
Delinquencies (Last 2 yrs)


Status
Delinquent
Not Delinquent
100
80
60
40
20
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
24.0
25.0
27.0
28.0
31.0
32.0
33.0
Inquiries in the Last 6 Months

[26]:
```
# També acompanyat del test chi-squared.
for var in ordinal:
    print(find_p_value(df_selectedCategorical, var, 'Status'))
```

```
{'stat': 827.252472713227, 'p-value': 1.4697988256454803e-176, 'column':
'Approx. Fico Score'}
{'stat': 134.59194258002924, 'p-value': 2.521867075962844e-27, 'column': 'Public
Records On File'}
{'stat': 62.28182921880612, 'p-value': 1.3375717974134529e-09, 'column':
'Employment Length'}
{'stat': 53.165058580299686, 'p-value': 1.6753618066379046e-07, 'column':
'Delinquencies (Last 2 yrs)'}
{'stat': 1446.3503842921966, 'p-value': 2.7795627572500547e-288, 'column':
'Inquiries in the Last 6 Months'}
```

Es conclou que la correlació més forta existeix en la variable de les consultes de crèdit (inquiries), que presenta una clara relació amb la variable de l'etiqueta del frau.

La puntuació de Fico la segueix, però se sospita que aquestes dues variables són altament dependents. Aquesta assumpció es pot comprovar amb la mateixa prova de chi squared.

[27]:
```
find_p_value(df_selectedCategorical, 'Approx. Fico Score','Inquiries in the Last␣
 ↪6 Months')
```

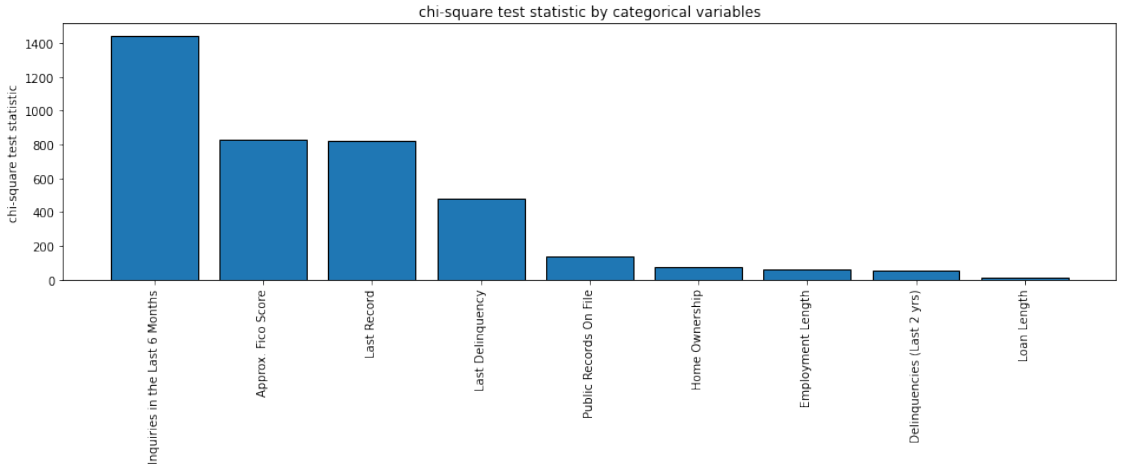
[27]:
```
{'stat': 2836.3995080745863, 'p-value': 0.0, 'column': 'Approx. Fico Score'}
```

Es rebutja la hipòtesi nul·la i s'accepta que són relacionades. El valor de l'estadístic intueix que les dues variables aporten la mateixa informació, en efectes pràctics.

Finalment, es pot realitzar una classificació a tall de resum del grau de relació entre les variables categòriques/ordinals i l'estat del préstec. Es prenen només aquelles variables que han rebutjat la hipòtesi nul·la en el test de chi-squared.

[28]:
```
def find_p_values(df, columns, target_column):
    return [find_p_value(df, column, target_column) for column in columns]

columns = ['Home Ownership','Loan Length','Approx. Fico Score','Inquiries in the␣
 ↪Last 6 Months',\
           'Delinquencies (Last 2 yrs)','Last Delinquency','Public Records On␣
 ↪File',\
           'Last Record','Employment Length']
chi_df = pd.DataFrame(find_p_values(df_selectedCategorical, columns, 'Status'))

chi_df = chi_df.sort_values('stat',ascending=False)

plt.figure(figsize=(16,4))
plt.bar(chi_df['column'],chi_df['stat'],edgecolor='black')
plt.xticks(rotation="vertical")
plt.ylabel("chi-square test statistic")
plt.title("chi-square test statistic by categorical variables")
```

```
plt.show()
```



Resum En aquest apartat s'ha realitzat una anàlisi estadístic bàsic enfocat principalment a la visualització sobre les relacions entre les variables i la variable objectiu, l'etiqueta del crèdit (fraudulent o no). Els resultats obtinguts són els següents:

1. Els valors perduts de les variables "Monts since Last Delinquency | Record" eren en realitat indicadors. Representen quan el client no ha comès cap infracció o obert cap registre. Es decideix categoritzar la variable en una de nova per poder mantenir tota la informació, encara que no sigui numèrica. De moment la imputació s'ha realitzat de forma visual, separant els grups d'interès. Potser es decideix utilitzar un altre mètode d'imputació en apartats posteriors.
2. Existeixen relacions molt fortes entre les variables categòriques/ordinals i la variable d'interès. En ordre d'importància:
3. Inquiries in the last 6 Months.
4. Fico Score.
5. Last Record (nova variable).
6. Last Delinquency (nova variable).
7. Altres variables que superen la prova de dependència, però no són tan fortes com les anteriors són:
8. Public records on file.
9. Home Ownership.
10. Employmentn Length.
11. Delinquencies (Last 2 yrs.)
12. Loan Length
13. També s'ha observat si la distribució de les variables numèriques varia significativament segons la categoria del préstec. Aquelles variables que clarament presenten diferències són:
14. Monthly Income.
15. Total Credit Lines.
16. Revolving Line Utilization.
17. Les altres variables numèriques que no presenten diferències en les distribucions o no són

trivials a simple vista són:

18. Monthly Payment.
19. Open Credit Lines.
20. Revolving Credit Balance.
21. Debt to income ratio.

1.3 Neteja de dades

1.3.1 Elements buits

Els 29 valors buits Si s'analitza l'estructura dels valors buits del dataset, es pot observar que algunes variables tenen exactament el mateix nombre de valors buits (29).

Mitjançant la inspecció d'una taula, es pot veure com les variables comparteixen els mateixos elements per als valors buits. És a dir, totes les 29 files no tenen cap valor imputat per a variables com Open credit lines, revolving credit balance...

[29]:
```
# Inspeccionar les 29 files amb valors faltants per una de les variables que té
↪48570 elements
df_selectedCategorical[df_selectedCategorical['Open CREDIT Lines'].isna()].
↪head(29)
```

[29]:

	Total Amount Funded	Loan Length	Monthly PAYMENT	Debt-To-Income Ratio	\
106	1000	36 months	30.94	1.10	
192	1000	36 months	32.11	10.00	
352	1000	36 months	34.21	16.27	
504	1200	36 months	38.17	3.27	
663	1275	36 months	42.65	10.00	
752	1400	36 months	45.78	8.61	
1545	1900	36 months	61.00	10.00	
1798	2000	36 months	64.50	10.00	
2733	2500	36 months	77.69	10.36	
2969	2525	36 months	80.69	10.00	
3065	2600	36 months	81.94	6.46	
3923	3000	36 months	93.23	0.39	
4156	3000	36 months	95.42	10.00	
4198	3000	36 months	95.86	10.00	
5088	3200	36 months	103.20	10.00	
5883	3500	36 months	113.39	10.00	
6932	3900	36 months	124.62	10.00	
8135	4350	36 months	136.45	4.00	
9891	5000	36 months	155.38	1.00	
10084	5000	36 months	156.11	8.81	
10085	5000	36 months	156.11	5.38	
11211	5000	36 months	164.23	3.51	
11226	5075	36 months	164.42	10.00	
16125	6500	36 months	204.84	4.00	
16510	6500	36 months	208.66	10.00	
16578	6700	36 months	209.18	1.00	

```
16806                6450   36 months           211.85                  10.00
17480                7000   36 months           218.55                   1.00
29696               10500   36 months           344.87                  19.50

      Home Ownership  Monthly Income  Approx. Fico Score  Open CREDIT Lines  \
106             RENT         4166.67                770.0                NaN
192             RENT         1000.00                695.0                NaN
352             RENT         2083.33                650.0                NaN
504             RENT         3000.00                695.0                NaN
663             RENT         3333.33                695.0                NaN
752             RENT         3333.33                669.0                NaN
1545        MORTGAGE         8333.33                695.0                NaN
1798            RENT          500.00                695.0                NaN
2733        MORTGAGE         5552.00                790.0                NaN
2969            RENT         9166.67                695.0                NaN
3065        MORTGAGE          541.67                732.0                NaN
3923        MORTGAGE         6666.67                790.0                NaN
4156            RENT         2916.67                732.0                NaN
4198             OWN         1666.67                695.0                NaN
5088        MORTGAGE        12500.00                695.0                NaN
5883            RENT        15000.00                695.0                NaN
6932            RENT         8166.67                695.0                NaN
8135            RENT        10000.00                732.0                NaN
9891            NONE           -0.08                790.0                NaN
10084       MORTGAGE         5833.33                770.0                NaN
10085       MORTGAGE        25000.00                770.0                NaN
11211           RENT         2333.33                669.0                NaN
11226           RENT         7916.67                695.0                NaN
16125           NONE           -0.08                732.0                NaN
16510           RENT         1666.67                695.0                NaN
16578           NONE           -0.08                790.0                NaN
16806           RENT         2666.67                695.0                NaN
17480           NONE           -0.08                790.0                NaN
29696           RENT         5000.00                732.0                NaN

       Total CREDIT Lines  Revolving CREDIT Balance  ...  \
106                   NaN                       NaN  ...
192                   NaN                       NaN  ...
352                   NaN                       NaN  ...
504                   NaN                       NaN  ...
663                   NaN                       NaN  ...
752                   NaN                       NaN  ...
1545                  NaN                       NaN  ...
1798                  NaN                       NaN  ...
2733                  NaN                       NaN  ...
2969                  NaN                       NaN  ...
3065                  NaN                       NaN  ...
```

```
3923                    NaN                         NaN  ...
4156                    NaN                         NaN  ...
4198                    NaN                         NaN  ...
5088                    NaN                         NaN  ...
5883                    NaN                         NaN  ...
6932                    NaN                         NaN  ...
8135                    NaN                         NaN  ...
9891                    NaN                         NaN  ...
10084                   NaN                         NaN  ...
10085                   NaN                         NaN  ...
11211                   NaN                         NaN  ...
11226                   NaN                         NaN  ...
16125                   NaN                         NaN  ...
16510                   NaN                         NaN  ...
16578                   NaN                         NaN  ...
16806                   NaN                         NaN  ...
17480                   NaN                         NaN  ...
29696                   NaN                         NaN  ...

       Delinquencies (Last 2 yrs)  Months Since Last Delinquency  \
106                           NaN                            NaN
192                           NaN                            NaN
352                           NaN                            NaN
504                           NaN                            NaN
663                           NaN                            NaN
752                           NaN                            NaN
1545                          NaN                            NaN
1798                          NaN                            NaN
2733                          NaN                            NaN
2969                          NaN                            NaN
3065                          NaN                            NaN
3923                          NaN                            NaN
4156                          NaN                            NaN
4198                          NaN                            NaN
5088                          NaN                            NaN
5883                          NaN                            NaN
6932                          NaN                            NaN
8135                          NaN                            NaN
9891                          NaN                            NaN
10084                         NaN                            NaN
10085                         NaN                            NaN
11211                         NaN                            NaN
11226                         NaN                            NaN
16125                         NaN                            NaN
16510                         NaN                            NaN
16578                         NaN                            NaN
16806                         NaN                            NaN
```

```
17480                         NaN                           NaN
29696                         NaN                           NaN

       Public Records On File  Months Since Last Record  Employment Length  \
106                       NaN                       NaN                6.0
192                       NaN                       NaN                0.0
352                       NaN                       NaN               10.0
504                       NaN                       NaN                0.0
663                       NaN                       NaN                1.0
752                       NaN                       NaN                0.0
1545                      NaN                       NaN                1.0
1798                      NaN                       NaN                0.0
2733                      NaN                       NaN                9.0
2969                      NaN                       NaN                0.0
3065                      NaN                       NaN                3.0
3923                      NaN                       NaN                1.0
4156                      NaN                       NaN                0.0
4198                      NaN                       NaN                0.0
5088                      NaN                       NaN                0.0
5883                      NaN                       NaN                0.0
6932                      NaN                       NaN                0.0
8135                      NaN                       NaN                0.0
9891                      NaN                       NaN                0.0
10084                     NaN                       NaN               10.0
10085                     NaN                       NaN               10.0
11211                     NaN                       NaN                0.0
11226                     NaN                       NaN                0.0
16125                     NaN                       NaN                0.0
16510                     NaN                       NaN                0.0
16578                     NaN                       NaN                0.0
16806                     NaN                       NaN                2.0
17480                     NaN                       NaN                0.0
29696                     NaN                       NaN                3.0

              Status  Months Since Last Delinquency NoNA  \
106    Not Delinquent                              999.0
192    Not Delinquent                              999.0
352    Not Delinquent                              999.0
504    Not Delinquent                              999.0
663        Delinquent                              999.0
752    Not Delinquent                              999.0
1545   Not Delinquent                              999.0
1798   Not Delinquent                              999.0
2733   Not Delinquent                              999.0
2969   Not Delinquent                              999.0
3065       Delinquent                              999.0
3923   Not Delinquent                              999.0
```

4156	Not Delinquent	999.0
4198	Not Delinquent	999.0
5088	Not Delinquent	999.0
5883	Not Delinquent	999.0
6932	Not Delinquent	999.0
8135	Not Delinquent	999.0
9891	Not Delinquent	999.0
10084	Not Delinquent	999.0
10085	Not Delinquent	999.0
11211	Not Delinquent	999.0
11226	Not Delinquent	999.0
16125	Not Delinquent	999.0
16510	Delinquent	999.0
16578	Not Delinquent	999.0
16806	Not Delinquent	999.0
17480	Not Delinquent	999.0
29696	Not Delinquent	999.0

	Months Since Last Record NoNA	Last Delinquency	Last Record
106	999.0	NEVER	NEVER
192	999.0	NEVER	NEVER
352	999.0	NEVER	NEVER
504	999.0	NEVER	NEVER
663	999.0	NEVER	NEVER
752	999.0	NEVER	NEVER
1545	999.0	NEVER	NEVER
1798	999.0	NEVER	NEVER
2733	999.0	NEVER	NEVER
2969	999.0	NEVER	NEVER
3065	999.0	NEVER	NEVER
3923	999.0	NEVER	NEVER
4156	999.0	NEVER	NEVER
4198	999.0	NEVER	NEVER
5088	999.0	NEVER	NEVER
5883	999.0	NEVER	NEVER
6932	999.0	NEVER	NEVER
8135	999.0	NEVER	NEVER
9891	999.0	NEVER	NEVER
10084	999.0	NEVER	NEVER
10085	999.0	NEVER	NEVER
11211	999.0	NEVER	NEVER
11226	999.0	NEVER	NEVER
16125	999.0	NEVER	NEVER
16510	999.0	NEVER	NEVER
16578	999.0	NEVER	NEVER
16806	999.0	NEVER	NEVER
17480	999.0	NEVER	NEVER

```
29696                          999.0             NEVER        NEVER

[29 rows x 22 columns]
```

Efectivament totes comparteixen els mateixos índexs. Es pot interpretar com que aquestes files no tenen la informació completa. Per tant, resulta indicat eliminar-les, ja que és un nombre molt baix i no és necessari inferir els valors perduts en funció d'altres columnes, ja que falta molta informació.

[30]:
```
# Netejar les files que tenen masses valors buits
df_noWrongRows = df_selectedCategorical[df_selectedCategorical['Open CREDIT␣
 ↪Lines'].notna()]
```

Fico score i revolving line utilization Existeixen algunes files buides per a la puntuació Fico. El seu nombre és extremadament petit però (17 files). Per tant, es decideix prescindir d'aquestes files ja que no aporten gaire significació envers a les 40000 restants per a l'anàlisi estadístic.

El mateix es pot aplicar a revolving line utilization, que consta d'uns ~70 missings.

[31]:
```
df_removedFicoRevolving = df_noWrongRows[df_noWrongRows['Approx. Fico Score'].
 ↪notna() & \
                                         df_noWrongRows['Revolving Line␣
 ↪Utilization'].notna()]

print(df_removedFicoRevolving['Approx. Fico Score'].isna().value_counts())
print(df_removedFicoRevolving['Revolving Line Utilization'].isna().
 ↪value_counts())
```

```
False    48099
Name: Approx. Fico Score, dtype: int64
False    48099
Name: Revolving Line Utilization, dtype: int64
```

Employment length Existeixen unes 2000 files de la variable Employment Length (anys que el client porta treballant) que són perdudes. Es desconeix si el motiu és o no intencionat.

Es sospita que el significat dels valors perduts pot ser el següent:

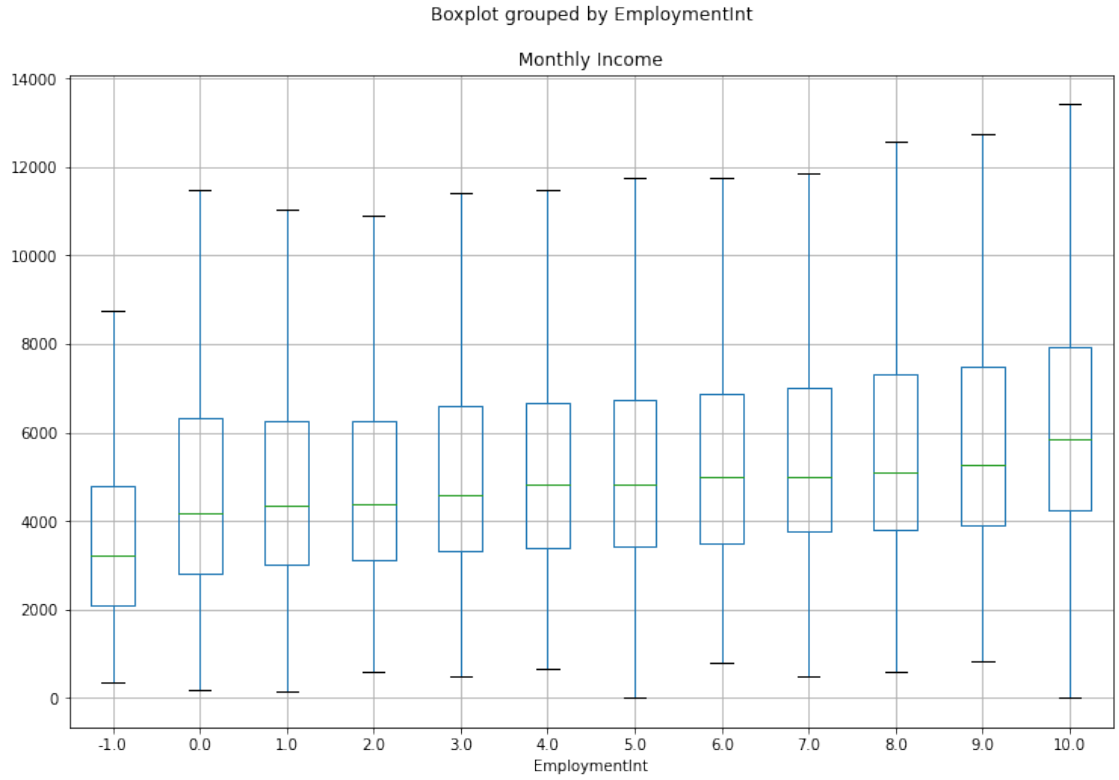
- Que sigui una dada perduda pot codificar un cas especial. Per exemple que el client no té feina.
- Poden ser dades errònies que no han estat imputades.

Es desconeix quina de les dues pot ser la correcta, així que s'ha de recórrer a eines d'anàlisi per poder continuar. En concret, es pot optar per projectar els diagrames de caixa d'una variable numèrica en funció de les categories dels anys que porta el client treballant.

A partir de la visualització es podrà extreure una interpretació.

[32]:
```
# Codificar els nuls del dataset per Employment Length com a -1
df_removedFicoRevolving['EmploymentInt'] = df_removedFicoRevolving['Employment␣
 ↪Length']
```

```
df_removedFicoRevolving['EmploymentInt'] =␣
 ↪df_removedFicoRevolving['EmploymentInt'].fillna(-1)
# Mostra el diagrama de caixa per a cada categoria de employment length
df_removedFicoRevolving.boxplot('Monthly␣
 ↪Income','EmploymentInt',showfliers=False, figsize=(12,8))
plt.show()
```



Efectivament, la distribució dels valors quan la durada de la feina és NULL indica que el salari mitjà és inferior (la interpretació és clara de forma visual sense recórrer a tècniques de contrast).

Per tant, es pot donar per cert que els valors NULS codifiquen que el client no té feina (i per tant té uns ingressos inferiors a altres clients que sí que tenen feina). La font d'ingressos del client en aquest cas és desconeguda. Es sospita que poden ser prestacions socials, però no és rellevant per a aquesta tasca d'anàlisi. Sols resulta interessant conèixer la distribució per intentar classificar la dada d'una forma o una altra.

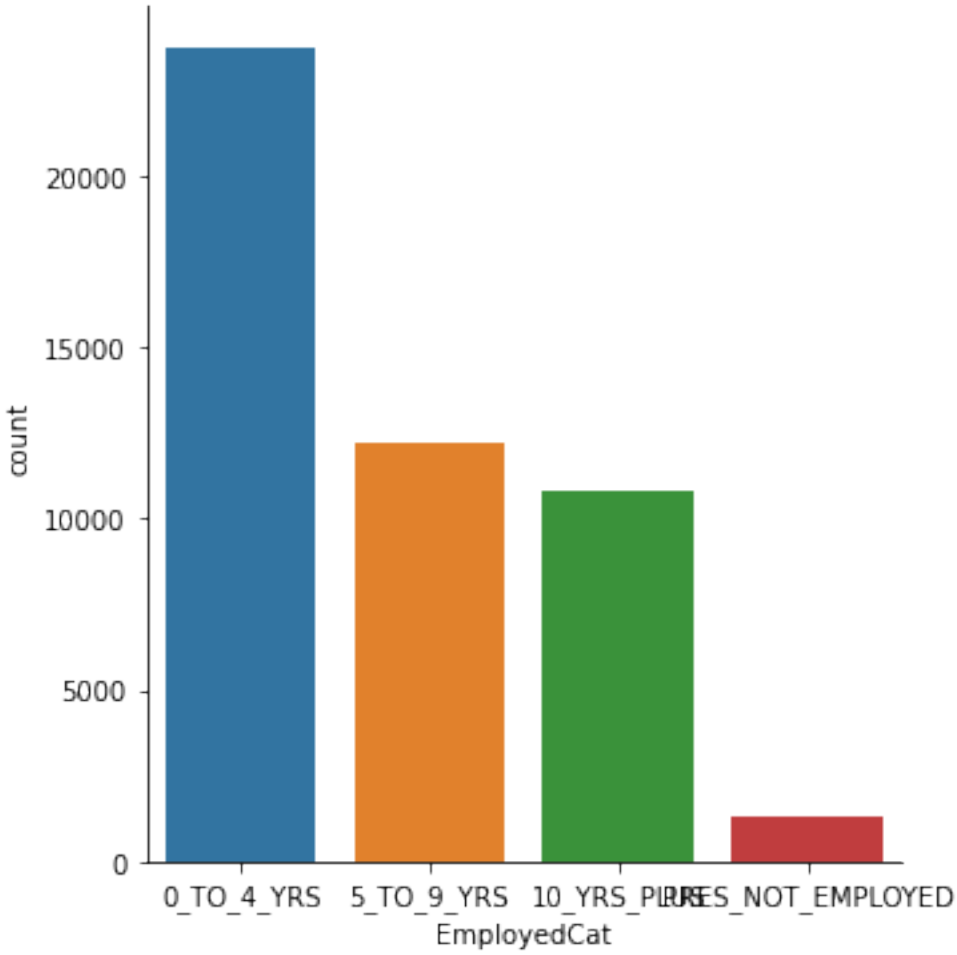
Per tant, es decideix categoritzar la variable en una de nova que incorpori la semàntica d'aquesta nova classe descoberta. Les categories d'aquesta classe seràn les següents:

- PRES_NOT_EMPLOYED: Presumptive not employed. El client mostra la característica d'una mitjana de salari inferior als empleats de 0 o més anys. Pressuntament sense feina i/o rebent prestacions socials o d'algun altre caire.
- 0_TO_4_YRS: Porta treballant de 0 a 4 anys.
- 5_TO_9_YRS: Porta treballant de 5 a 9 anys.

- 10_YRS_PLUS: Porta treballant 10 anys o més.

[33]:
```
def mapEmployment(x):
    if (x == -1):
        return 'PRES_NOT_EMPLOYED'
    elif (x >= 0 and x <= 4):
        return '0_TO_4_YRS'
    elif (x >= 5 and x <= 9):
        return '5_TO_9_YRS'
    elif (x >= 10):
        return '10_YRS_PLUS'

df_removedFicoRevolving['EmployedCat'] =␣
 ↪df_removedFicoRevolving['EmploymentInt'].map(mapEmployment)
```

[34]:
```
# Visualitzar les noves categories
sns.catplot(x='EmployedCat', data=df_removedFicoRevolving, kind='count')
plt.show()
```



[35]:
```
# Consservar nomes les variables transformades i d'interés fins la moment
cols_preserve = ['Total Amount Funded','Monthly PAYMENT',\
                 'Home Ownership','Monthly Income','Approx. Fico Score','Total␣
 ↪CREDIT Lines',\
                 'Revolving CREDIT Balance','Revolving Line Utilization',\
                 'Inquiries in the Last 6 Months','Delinquencies (Last 2␣
 ↪yrs)','Public Records On File',\
                 'Last Delinquency','Last Record','EmployedCat','Status']
df_nomissings = df_removedFicoRevolving[cols_preserve]
print(df_nomissings.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48099 entries, 0 to 48598
Data columns (total 15 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Total Amount Funded             48099 non-null  int64
 1   Monthly PAYMENT                 48099 non-null  float64
 2   Home Ownership                  48099 non-null  object
 3   Monthly Income                  48099 non-null  float64
 4   Approx. Fico Score              48099 non-null  float64
 5   Total CREDIT Lines              48099 non-null  float64
 6   Revolving CREDIT Balance        48099 non-null  float64
 7   Revolving Line Utilization      48099 non-null  float64
 8   Inquiries in the Last 6 Months  48099 non-null  float64
 9   Delinquencies (Last 2 yrs)      48099 non-null  float64
 10  Public Records On File          48099 non-null  float64
 11  Last Delinquency                48099 non-null  object
 12  Last Record                     48099 non-null  object
 13  EmployedCat                     48099 non-null  object
 14  Status                          48099 non-null  object
dtypes: float64(9), int64(1), object(5)
memory usage: 7.1+ MB
None
```

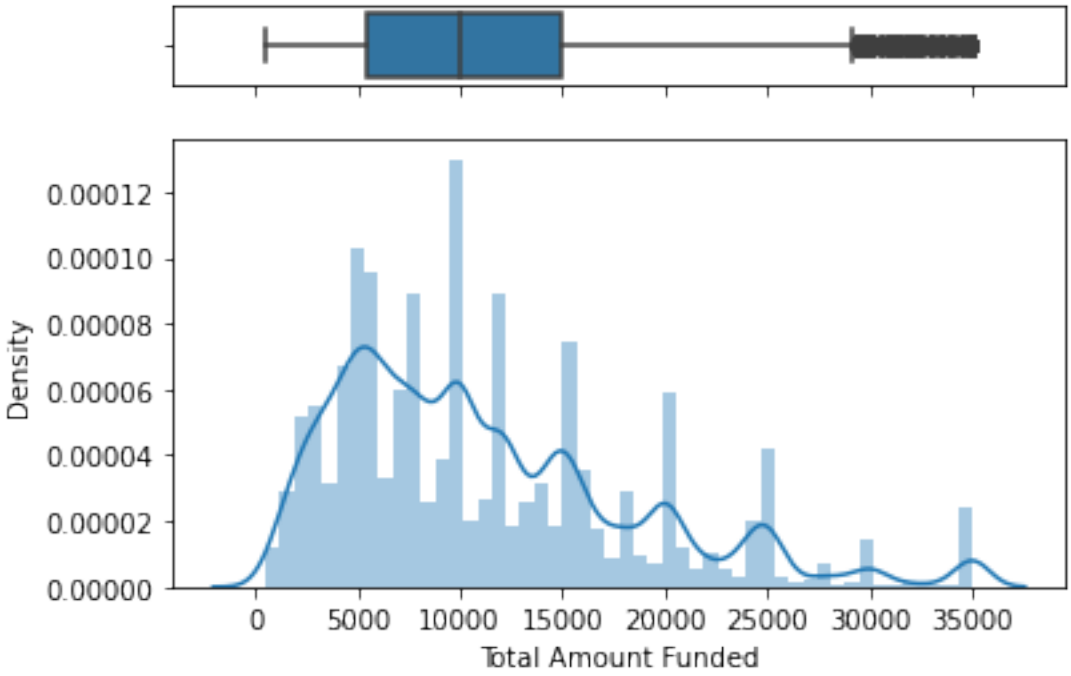
1.3.2 Identificació i tractament dels valors extrems

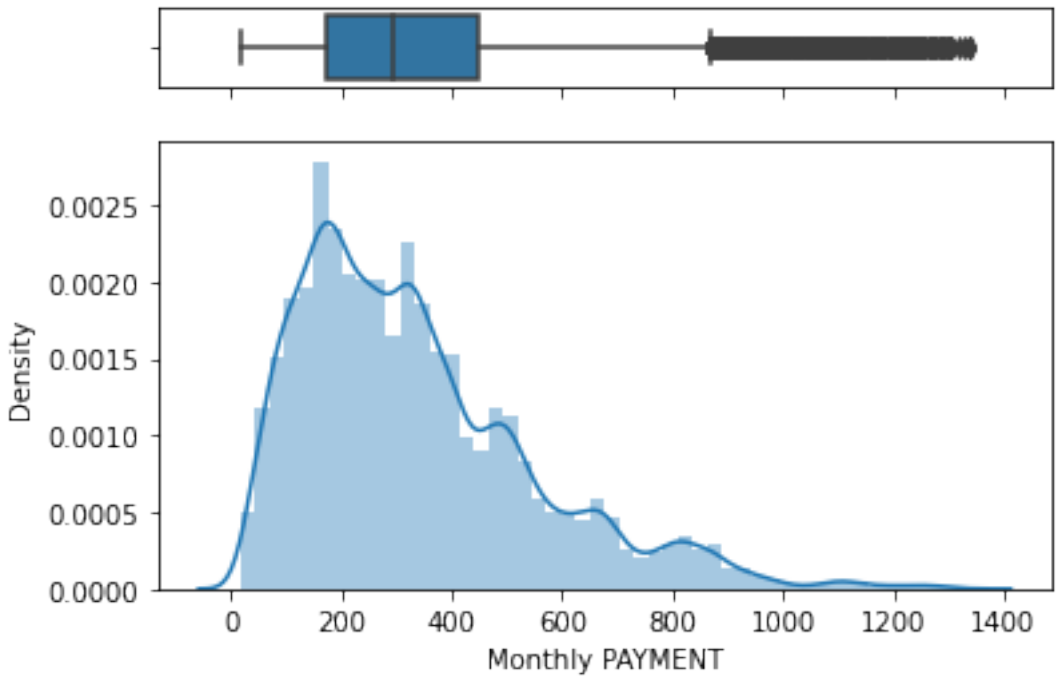
En aquesta secció d'identifiquen i es tracten les ocurrències de valors extrems o outliers. Aquests valors sols apareixen en les variables numèriques i poden comportar bias en les anàlisis estadístiques si no es tracten d'alguna manera.

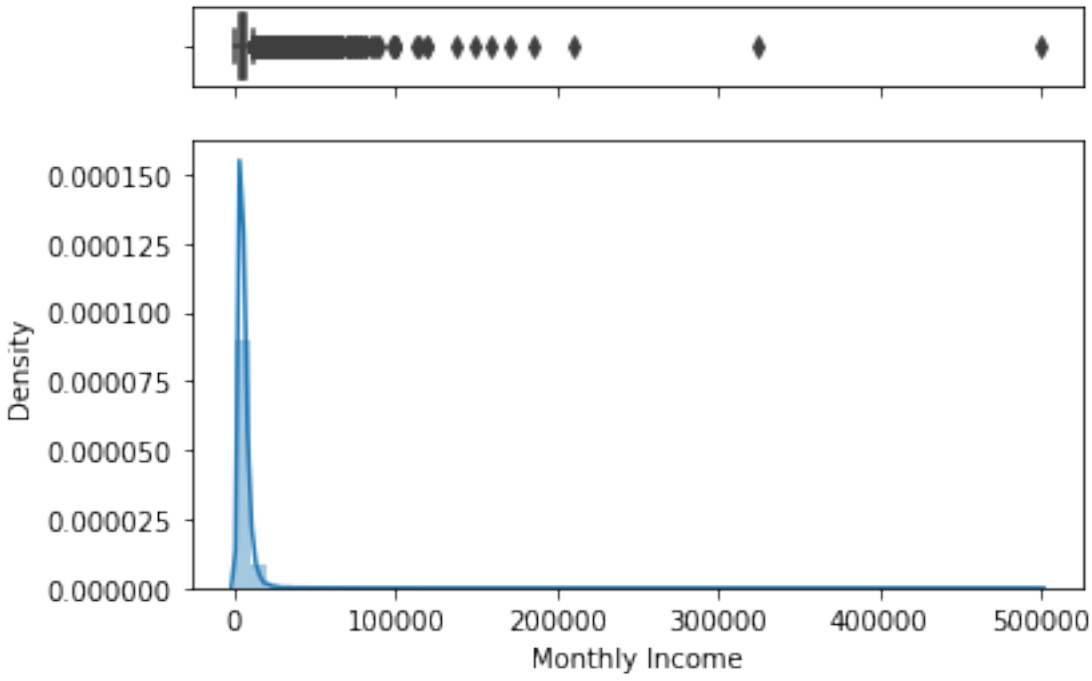
Aquelles variables que són ordinals (categòriques encara que prenguin valors numèriques), no necessiten aquest tractament.

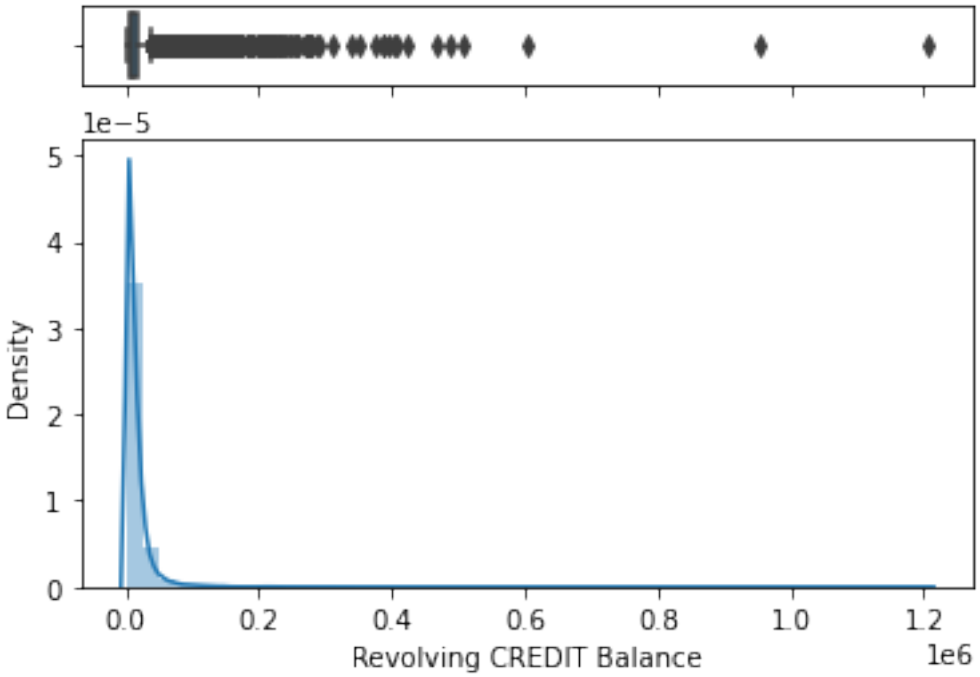
[36]:
```
def plot_hist(df, column):
    # Cut the window in 2 parts
```

```
    f, (ax_box, ax_hist) = plt.subplots(2, sharex=True,␣
 ↪gridspec_kw={"height_ratios": (.15, .85)})

    # Add a graph in each part
    sns.boxplot(df[column], ax=ax_box)
    sns.distplot(df[column], ax=ax_hist)

    # Remove x axis name for the boxplot
    ax_box.set(xlabel='')

def plot_amounts(df):
    # Es comença per la quantitat del crédit, el pagament mensual i el salari␣
 ↪del client
    plot_hist(df, 'Total Amount Funded')
    plot_hist(df, 'Monthly PAYMENT')

    plot_hist(df, 'Monthly Income')
    plot_hist(df, 'Revolving CREDIT Balance')

plot_amounts(df_nomissings)
```




0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Density
0
200
400
600
800
1000
1200
1400
Monthly PAYMENT


0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
Density
0
100000
200000
300000
400000
500000
Monthly Income



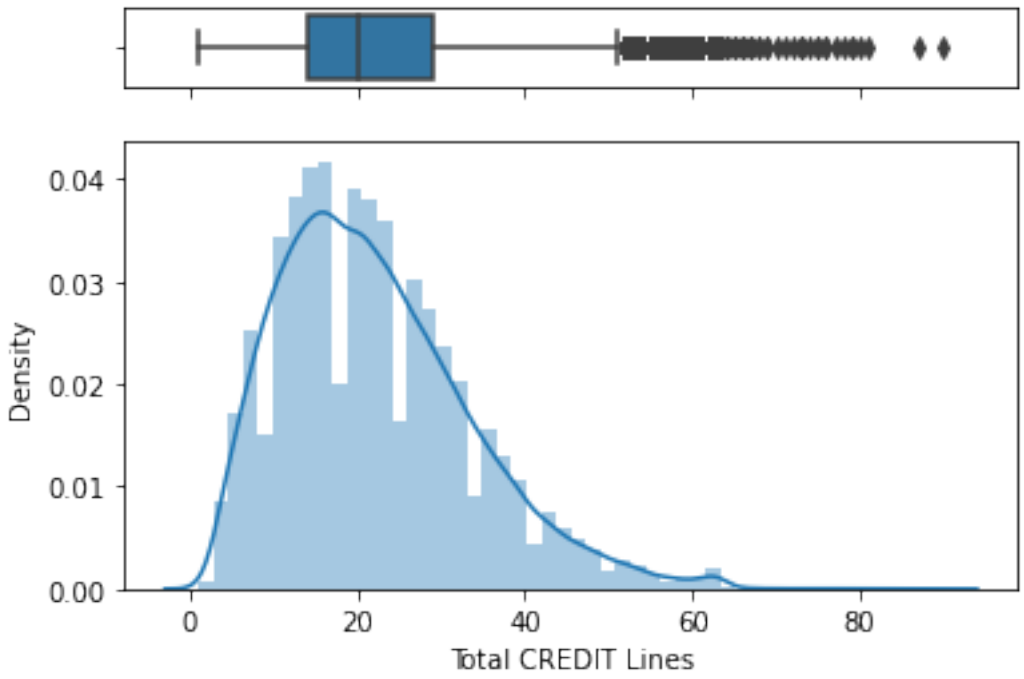
Com es pot observar les distribucions del volum del préstec i el pagament mensual són desplaçades a l'esquerra, provocant que no siguin ben bé normals. Existeixen valors atípics que són els que cab esperar, ja que les variables es distribueixen en funció de la riquesa de la població.

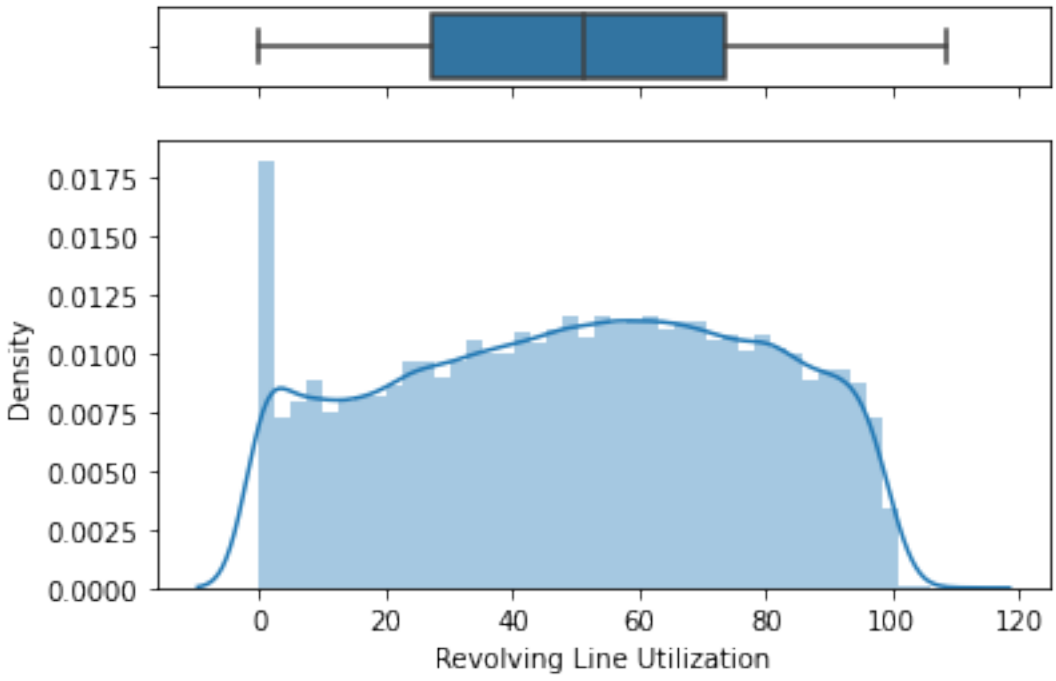
El casos més exagerats són els ingressos mensuals i el balanç de crédit, ja que apareixen molts valors atípics. Aquests valors atípics són representats per la població més rica, que esgarren completament la visualització en el seu favor.

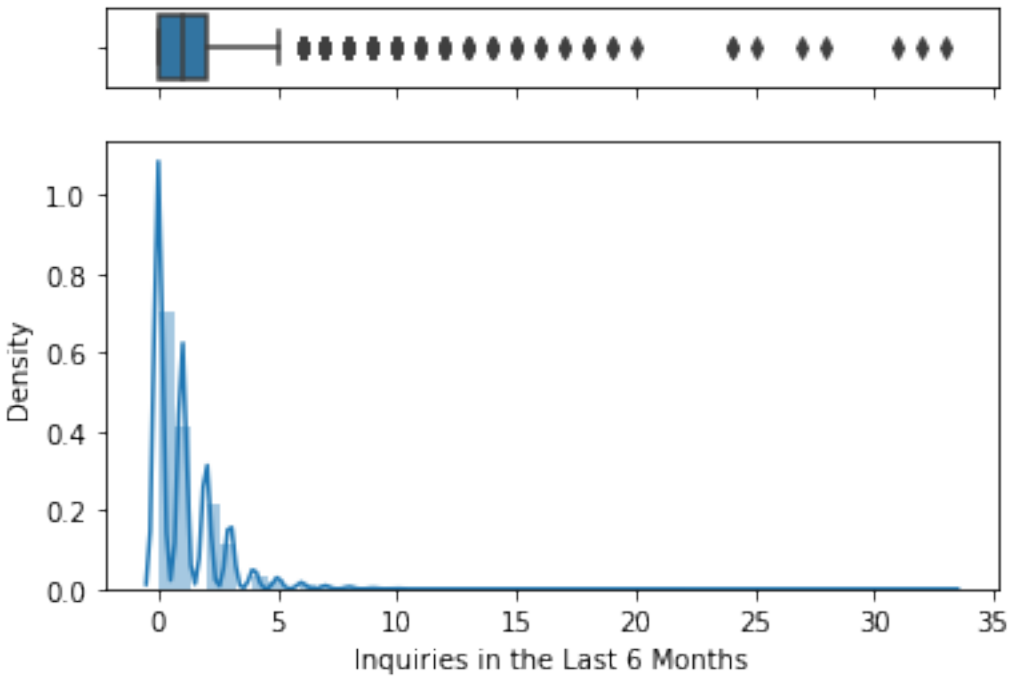
En els tres casos el tractament és el mateix i consisteix a eliminar completament tot punt que representi un outlier. No són dades errònies sinó fruit de la distribució de la riquesa. Poden suposar un bias molt gran de cara a les anàlisis, pel que la seva eliminació és justificada.

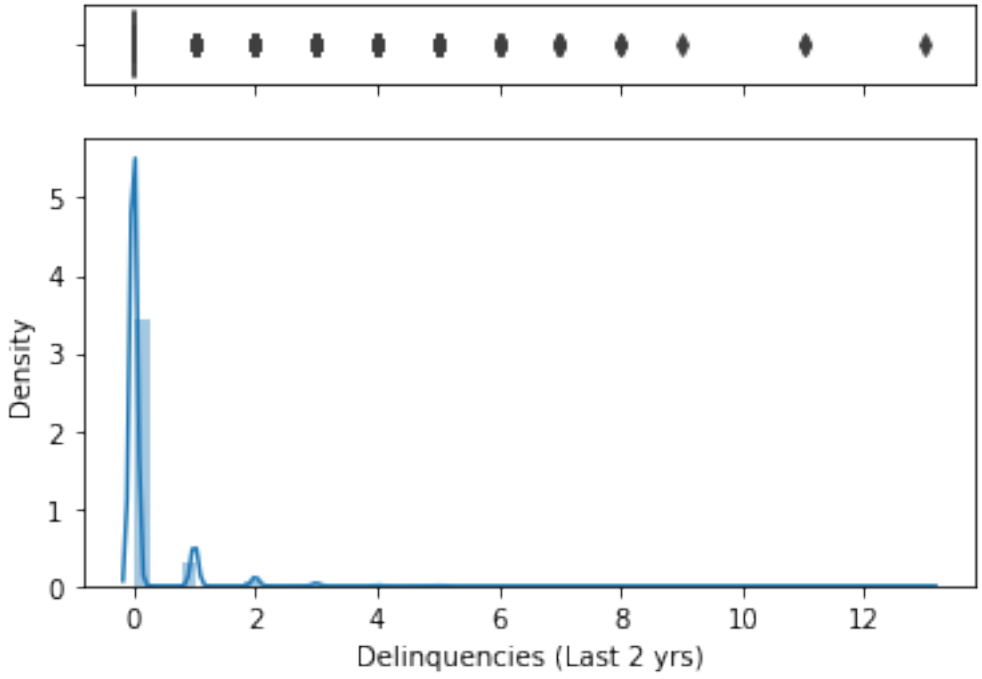
Es continua amb les altres variables numèriques i ordinals d'interés.

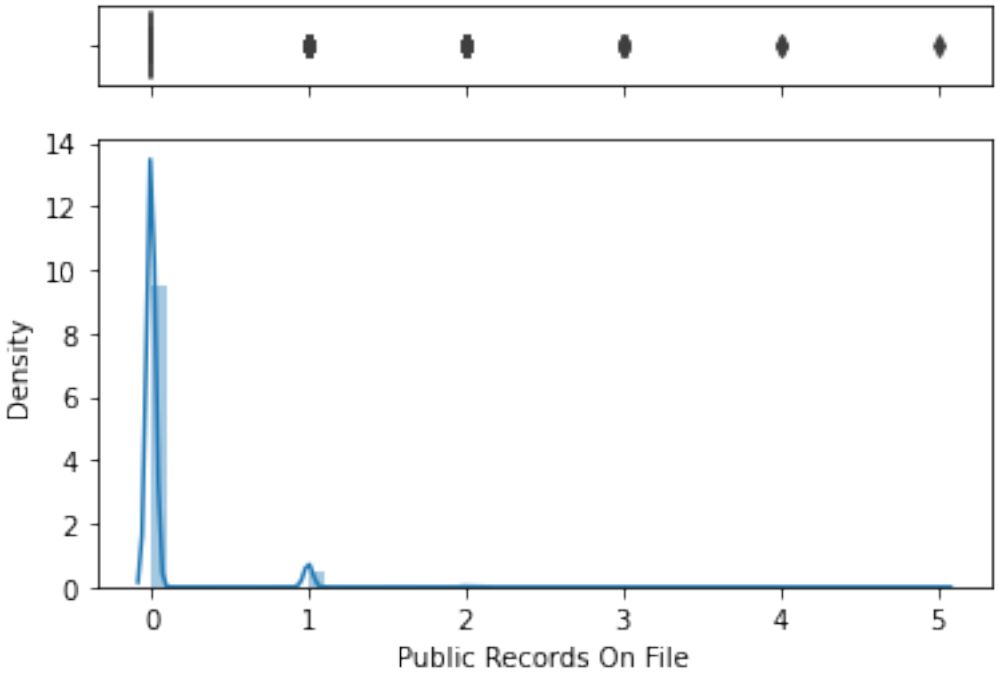
[37]:
```
def plot_misc(df):
    # Altres variables
    plot_hist(df, 'Total CREDIT Lines')
    plot_hist(df, 'Revolving Line Utilization')
    plot_hist(df, 'Inquiries in the Last 6 Months')
    plot_hist(df, 'Delinquencies (Last 2 yrs)')
    plot_hist(df, 'Public Records On File')

plot_misc(df_nomissings)
```


0.04
0.03
0.02
0.01
0.00
Density
0
20
40
60
80
Total CREDIT Lines


0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
0
20
40
60
80
100
120
Revolving Line Utilization


Density
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35
Inquiries in the Last 6 Months


Density
5
4
3
2
1
0
0
2
4
6
8
10
12
Delinquencies (Last 2 yrs)

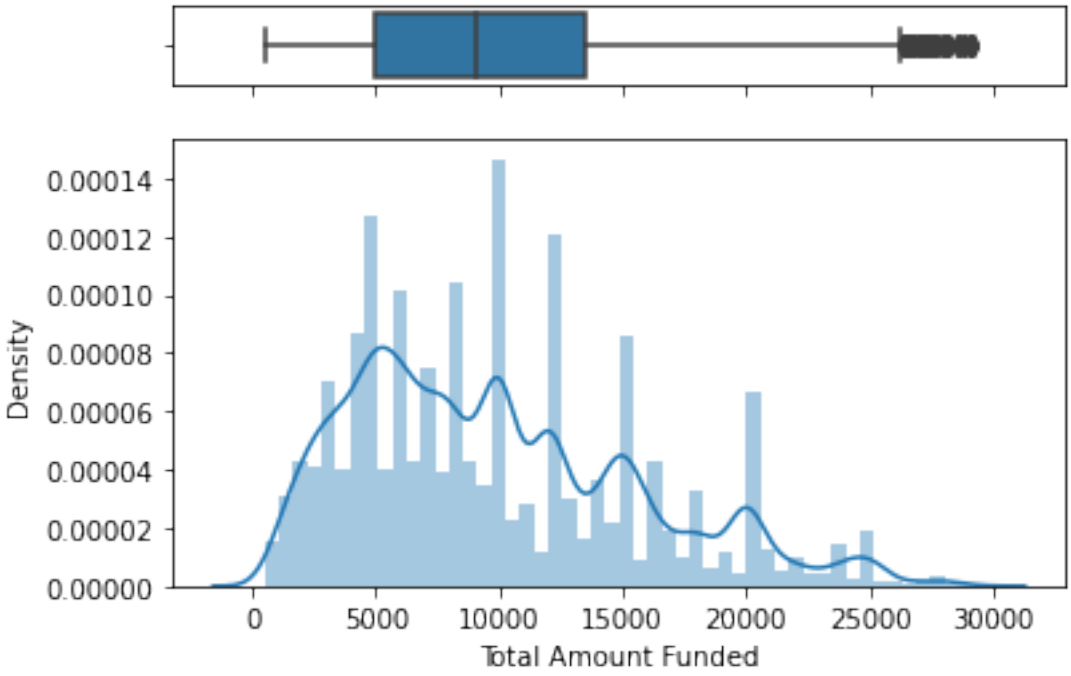


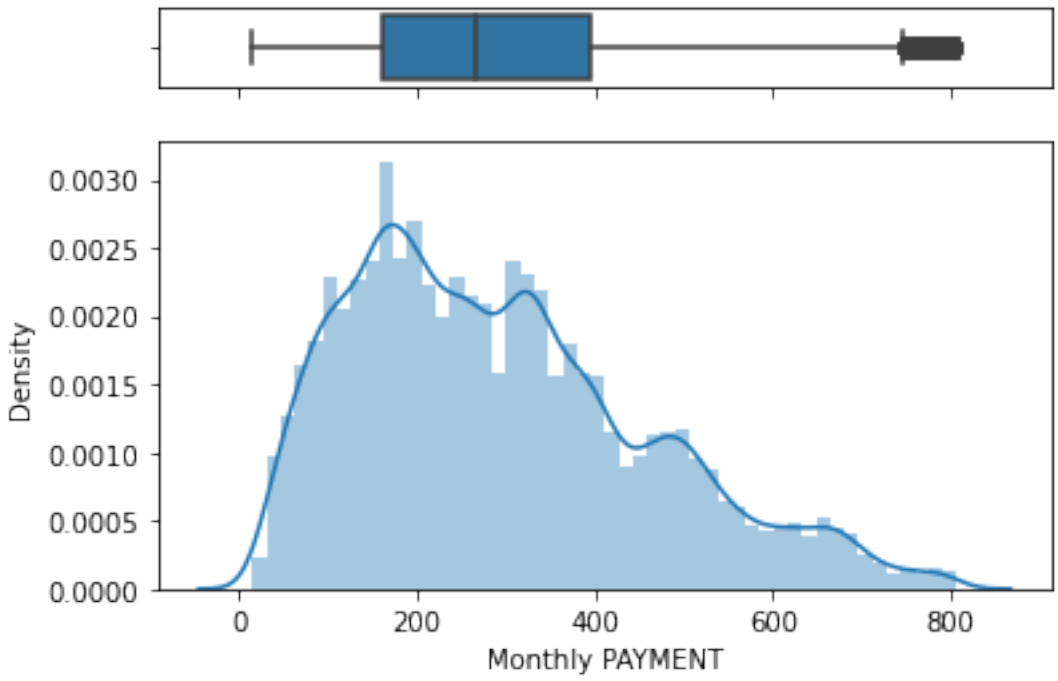
Les conclusions per aquestes variables són:

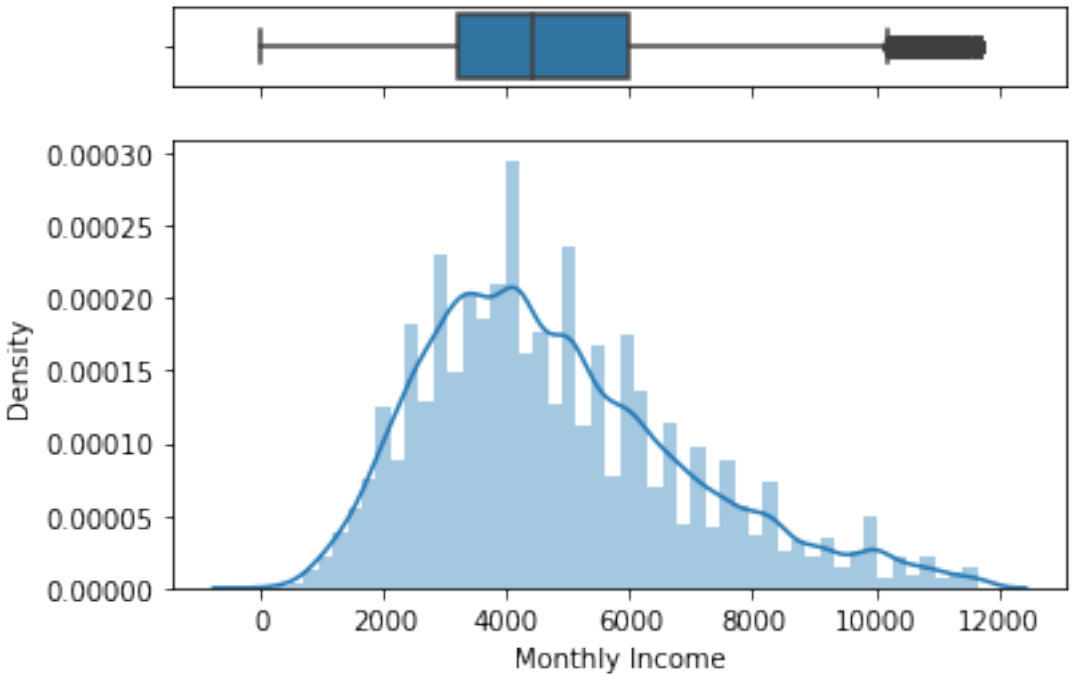
1. Les línies de crèdit totals de l'individu segueixen la mateixa distribució que les variables de quantitats. Els outliers s'eliminaran.
2. La utilització del crèdit (expressat en %), no segueix cap distribució específica ni presenta outliers. Es conserva tal com està.
3. Els outliers de consultes en els últims 6 mesos es poden eliminar, ja que es tracten de casos individuals. Aquests poden o no ser etiquetats com a fraudulents i per tant poden suposar un bias molt alt en les dades. S'han d'eliminar del dataset per tal de garantir que, a major nombre de consultes, major probabilitat de classificar el préstec com a fraudulent.
4. Les variables delinquencies in last 2 years i la de public records on file es poden ignorar, ja que pràcticament tots els valors més grans que 0 són outliers.

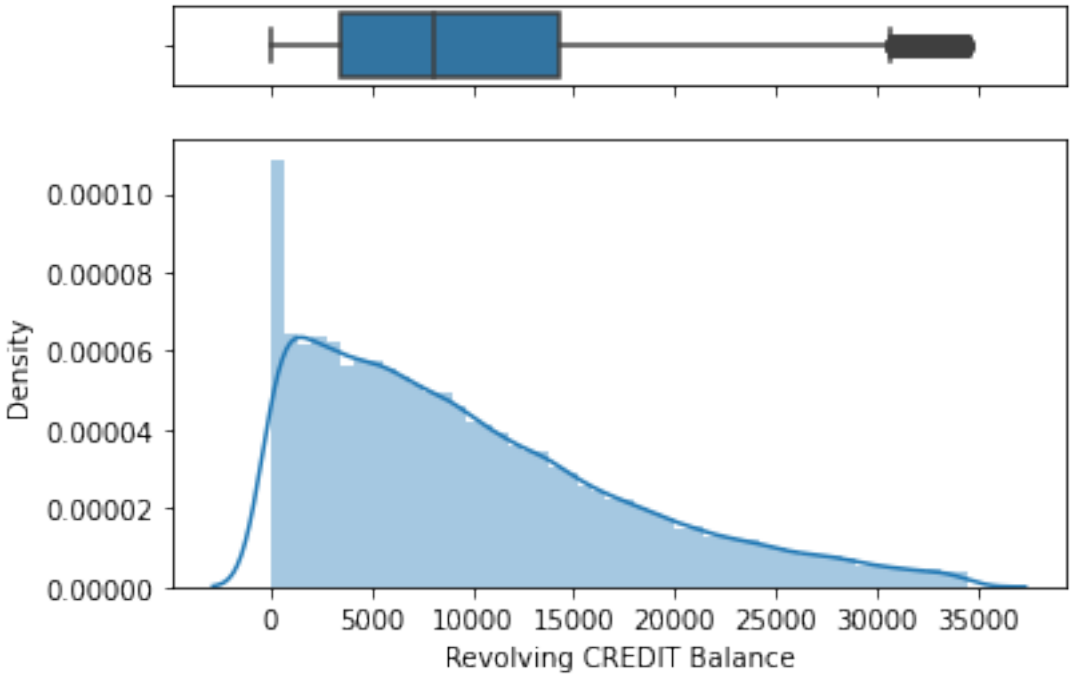
[38]:
```
# Eliminar els outliers
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    return df[(df[column] >= (Q1 - 1.5 * IQR)) & (df[column] <= (Q3 + 1.5 *␣
 ↪IQR))]

def remove_all_outliers(df, columns):
    i = 0
    ret = None
```

```
    for column in columns:
        if (i == 0):
            ret = remove_outliers(df, column)
        else:
            ret = remove_outliers(ret, column)
        i = i + 1
    return ret


# Eliminar outliers
cols = ['Total Amount Funded', 'Monthly PAYMENT', 'Monthly Income', 'Revolving␣
 ↪CREDIT Balance', \
        'Total CREDIT Lines', 'Inquiries in the Last 6 Months']
df_noOutliers = remove_all_outliers(df_nomissings, cols)

cols_drop = ['Delinquencies (Last 2 yrs)','Public Records On File']
# Eliminar variables que no aporten informació al treure els outliers
df_noOutliers = df_noOutliers.drop(labels=cols_drop, axis=1)

# Mostrar les noves distribucions
plot_amounts(df_noOutliers)
```

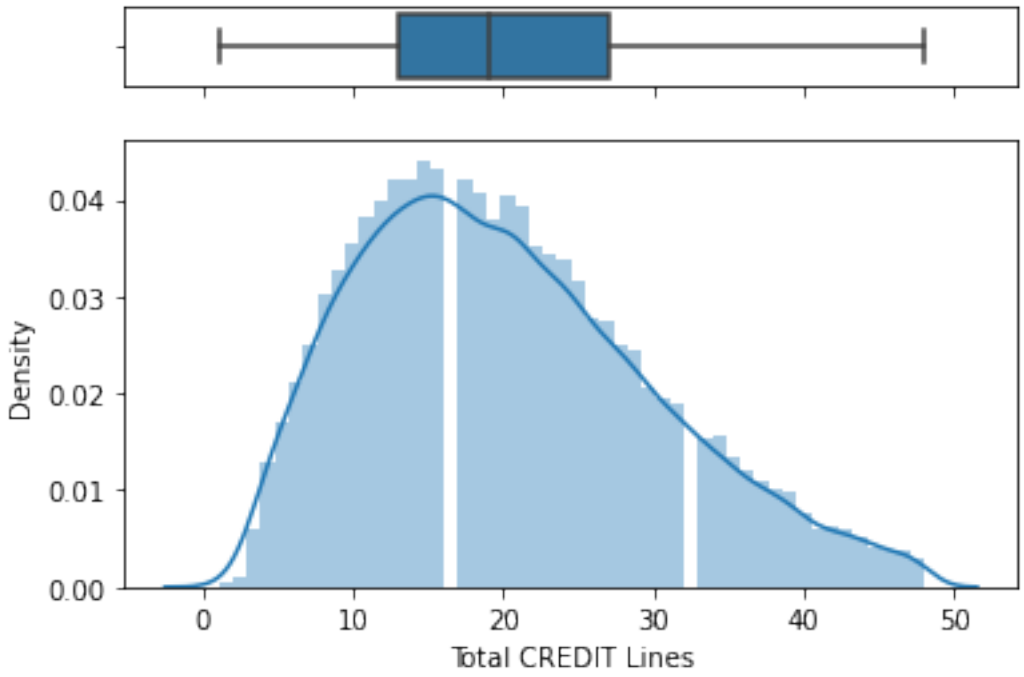


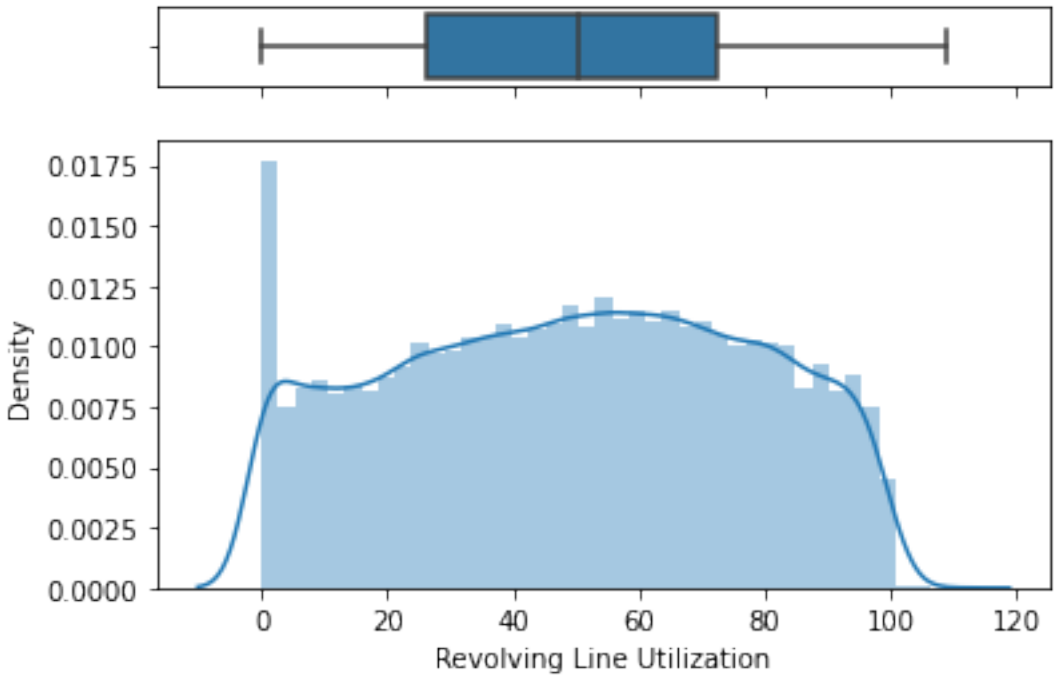

0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Density
0
200
400
600
800
Monthly PAYMENT

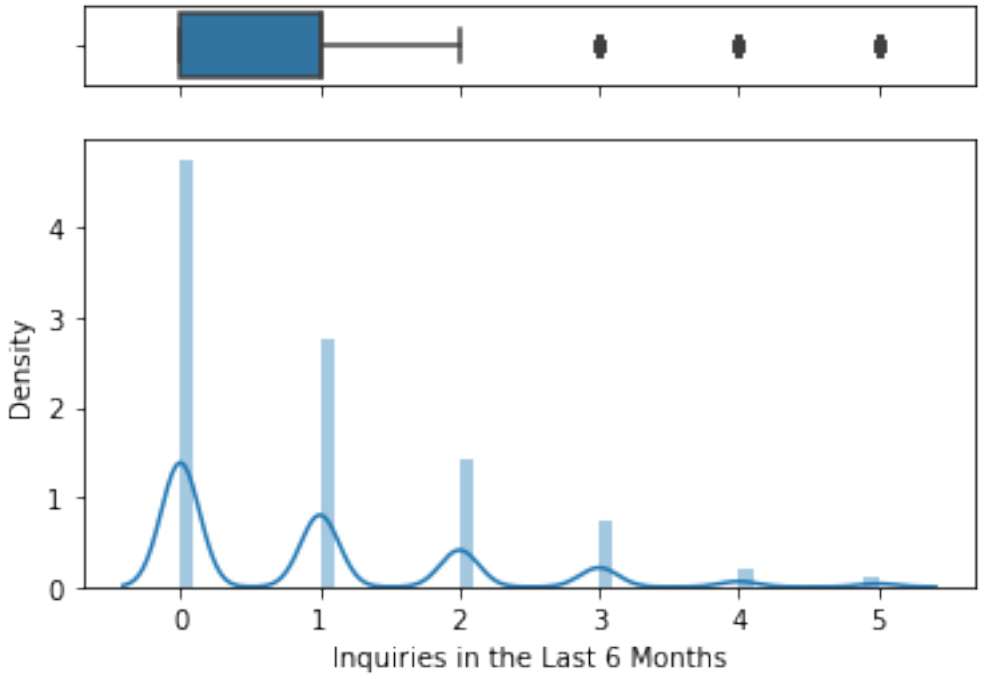

0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
2000
4000
6000
8000
10000
12000
Monthly Income



[39]:
```
plot_hist(df_noOutliers, 'Total CREDIT Lines')
plot_hist(df_noOutliers, 'Revolving Line Utilization')
plot_hist(df_noOutliers, 'Inquiries in the Last 6 Months')
```







Per últim, es corregeix un error de dades detectat de forma manual: existeixen individus amb

ingressos mensuals negatius.

Aparentment, sembla que sigui errors d'imputació o format. Es decideix prescindir d'aquestes files ja que en són molt poques.

[40]:
```
# Mostra les files errònies
df_noOutliers[df_noOutliers['Monthly Income'] <= 0].head()
```

[40]:
```
       Total Amount Funded  Monthly PAYMENT Home Ownership  Monthly Income  \
41733                18000           563.05       MORTGAGE           -0.08

       Approx. Fico Score  Total CREDIT Lines  Revolving CREDIT Balance  \
41733               790.0                38.0                   30769.0

       Revolving Line Utilization  Inquiries in the Last 6 Months  \
41733                        41.4                             1.0

      Last Delinquency Last Record  EmployedCat          Status
41733            NEVER       NEVER  10_YRS_PLUS  Not Delinquent
```

[41]:
```
# Eliminar les files errònies
df_noOutliers = df_noOutliers[df_noOutliers['Monthly Income'] > 0]
# Comprovar que s'hagi eliminat correctament
df_noOutliers[df_noOutliers['Monthly Income'] <= 0]['Monthly Income'].any()
```

[41]:
```
False
```

El dataset resultat després de la fase d'anàlisi i neteja és el següent:

[42]:
```
print(df_noOutliers.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 39724 entries, 0 to 46427
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Total Amount Funded             39724 non-null  int64
 1   Monthly PAYMENT                 39724 non-null  float64
 2   Home Ownership                  39724 non-null  object
 3   Monthly Income                  39724 non-null  float64
 4   Approx. Fico Score              39724 non-null  float64
 5   Total CREDIT Lines              39724 non-null  float64
 6   Revolving CREDIT Balance        39724 non-null  float64
 7   Revolving Line Utilization      39724 non-null  float64
 8   Inquiries in the Last 6 Months  39724 non-null  float64
 9   Last Delinquency                39724 non-null  object
 10  Last Record                     39724 non-null  object
 11  EmployedCat                     39724 non-null  object
 12  Status                          39724 non-null  object
```

```
dtypes: float64(7), int64(1), object(5)
memory usage: 4.2+ MB
None
```

[43]:
```
df_noOutliers.describe()
```

[43]:

	Total Amount Funded	Monthly PAYMENT	Monthly Income
count	39724.000000	39724.000000	39724.000000
mean	9807.487287	293.390298	4786.463751
std	5794.341354	167.030097	2157.087743
min	500.000000	15.670000	158.000000
25%	5000.000000	162.730000	3200.000000
50%	9000.000000	266.600000	4416.670000
75%	13500.000000	395.630000	6000.000000
max	29175.000000	806.570000	11666.670000

	Approx. Fico Score	Total CREDIT Lines	Revolving CREDIT Balance
count	39724.000000	39724.000000	39724.000000
mean	715.203504	20.224927	9795.452824
std	35.888430	9.841389	7835.466294
min	650.000000	1.000000	0.000000
25%	695.000000	13.000000	3477.000000
50%	695.000000	19.000000	8049.000000
75%	732.000000	27.000000	14337.250000
max	790.000000	48.000000	34491.000000

	Revolving Line Utilization	Inquiries in the Last 6 Months
count	39724.000000	39724.000000
mean	49.370617	0.925763
std	28.053437	1.138388
min	0.000000	0.000000
25%	26.400000	0.000000
50%	50.200000	1.000000
75%	72.500000	1.000000
max	108.800000	5.000000

[44]:
```
# Per últim, emmagatzemar el dataset
df_noOutliers.to_csv('LoanStatsDatasetCleaned.csv',sep=',')
```

1.4 Anàlisi de dades

1.4.1 Selecció dels grups

[5]:
```
# Llegir el data set per a la part d'anàlisi avançada
df = pd.read_csv('LoanStatsDatasetCleaned.csv', sep=',')
```

Les anàlisis estadístiques que es volen aplicar són:

1. Comparació de mitjanes del salari segons si el crèdit és fraudulent o no. Trobar si el salari

dels clients que són marcats com fraudulents és inferior als dels que no són fraudulents.
2. Grau de correlació entre totes les variables numèriques. Trobar quines variables numèriques són dependents unes a altres.
3. Realitzar la regressió logística per a calcular si el préstec és o no fraudulent en funció del nombre de consultes, de l'últim registre públic i de l'ús de la línia de crèdit.

Per tant, es requereixen les següents dades:

- Totes les variables numèriques.
- Status.
- Last Record.

1.4.2 Comprovació de la normalitat i l'homogeneïtat de la variància

Comprovació de la normalitat En primer lloc es comprova si la normalitat de les 6 variables numèriques que tenim en el dataset.

Per veure si una variable segueix una distribució normal podem fer-ho de diferents maneres:

1) Primer de tot, es realitza de forma visual, a través d'una comparació de la distribució de la variable "Montly Payment" amb la normal utilitzant la funció distplot.

2) En segon lloc, també es realitza de manera numèrica a través de dos indicadors de normalitat: 1.- Asimetria de la funció 2.- La kurtosis

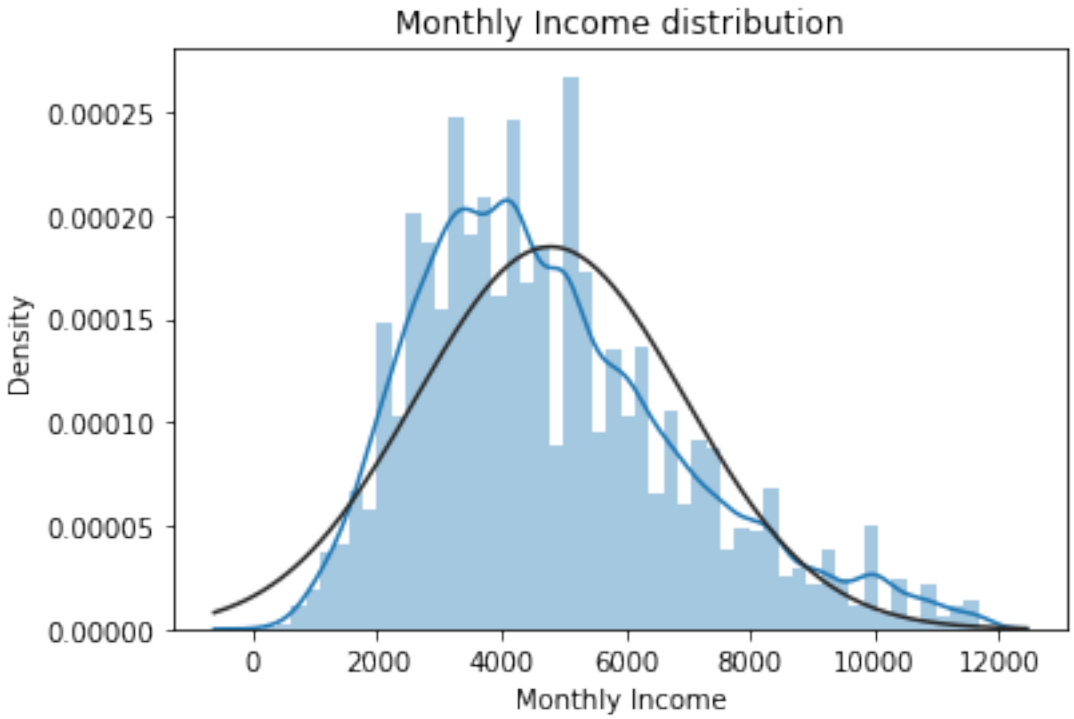
3) Es disposa de diferents tests per veure si una variable segueix una distribució normal, com el de Kolmogorov-Smirnov i el de Shapiro-Wilk, en aquest cas es selecciona el de Shapiro-Wilk, ja que encara que es considera un dels mètodes més potents, funciona per mostres menors de 50 observacions. El dataset disposa de 39.724 registres, per tant s'aplica el test de Kolmogorov-Smirnov. S'assumeix com a hipòtesi nul·la que els ingressos mensuals estan distribuïts normalment, si el p-valor és més petit que el nivell de significació alfa=0,05, es rebutja la hipòtesi nul·la i es conclou que les dades no segueixen una distribució normal.

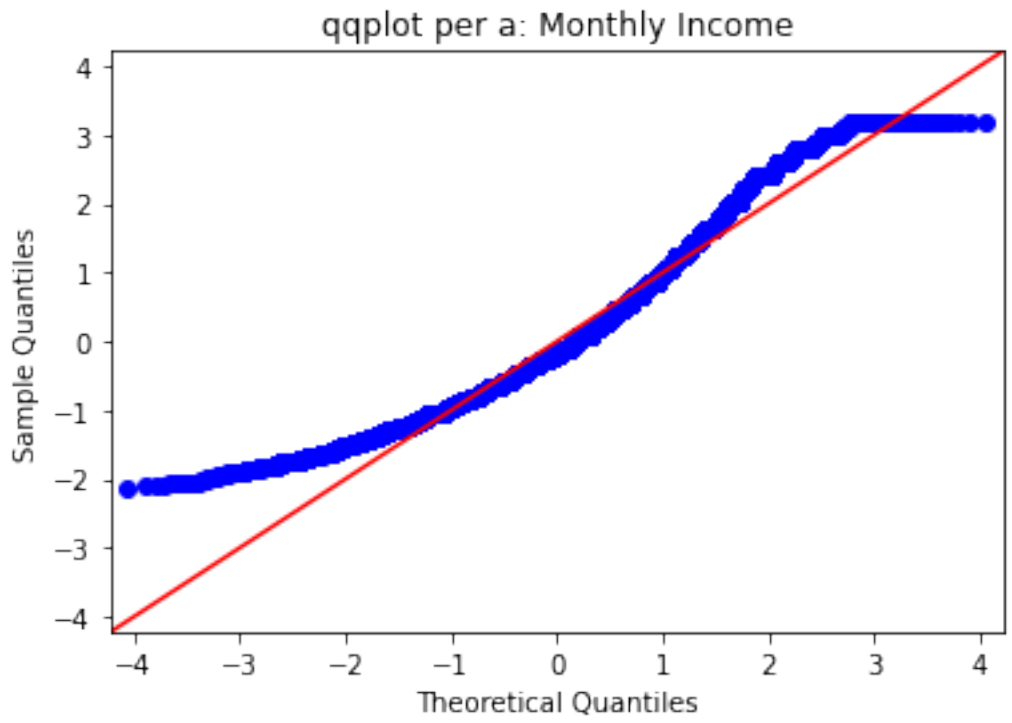
Es realitzen les 3 proves en la variable "Monthly Income" i després en funció del resultat s'escull una de les tres opcions per a fer-ho a totes les variables numèriques del dataset.

[23]:
```
from scipy.stats import norm

#Visualitzem la distribució de la variable "Monthly Income" comparant amb la␣
 ↪distibució normal.
sns.distplot(df['Monthly Income'], fit = norm)
plt.title("Monthly Income distribution")

#Calculo l'assimetria de la funció
print ("Assimetria de la funció: ", df['Monthly Income'].skew())

#Calculo la kurtosis
print ("Kurtosis: ", df['Monthly Income'].kurt())

#Calculo la prova de Shapiro.Wilk no enva va bé pq la mostra és superior a 50.
#from scipy.stats import shapiro
```

```
#stat, p = shapiro(df['Monthly PAYMENT'])
#print('Estadistics prova Shapiro.Wilk =%.3f, p-value =%.3f' % (stat, p))

def qqplot(variable):
    sm.qqplot(variable, line='45', fit=True)
    plt.title("qqplot per a: " + variable.name)

#Calculo la prova de Kolmogorov-Smirnov
from scipy.stats import kstest
media, desviacion = norm.fit(df['Monthly Income'])
stat, p = kstest(df['Monthly Income'], 'norm', args=(media,desviacion))
print('Estadistics prova Kolmogorov-Smirnov=%.3f, p-value=%.3f' % (stat, p))

qqplot(df['Monthly Income'])
```

```
Assimetria de la funció:  0.7560488417562314
Kurtosis:  0.24983206778454914
Estadistics prova Kolmogorov-Smirnov=0.084, p-value=0.000
```





Les conclusions per la variable "Montly Payment" són:

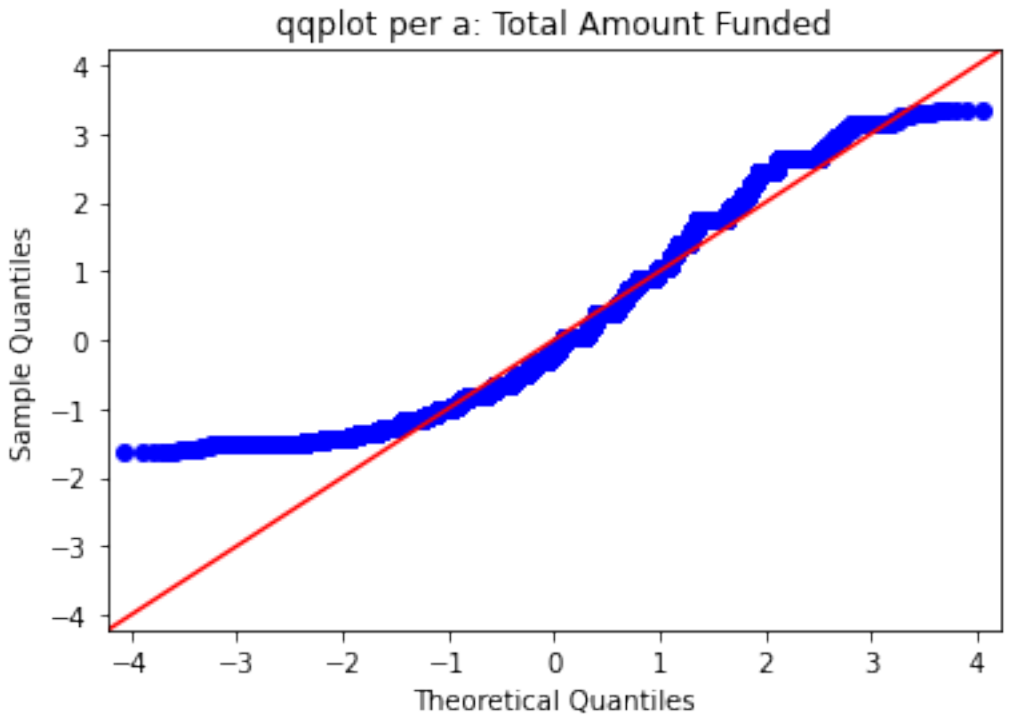
1. Gràficament té una distribució bastant normal, però el gràfic QQ mostra una discrepància respecte a la normal en els extrems de la distribució.
2. El coeficient d'asimetria ha resultat en 0,77. En una distribució normal el coeficient d'asimetria seria proper a 0. En aquest cas és més aviat proper a 1.
3. La funció kurtosis dóna idea de la relació del pic central amb els extrems de la campana de la distribució. Si és proper a 1, el valor serà coherent també la normalitat de la variable, en el nostre cas, és 0,24.
4. L prova de Kolmogorov-Smirnov ens ha resultat en un valor p = 0,0000 inferior al nivell de significació alfa=0,05

D'acord amb les proves realitzades sobre la variable "Montly Payment" es conclou que NO segueix una distribució normal.

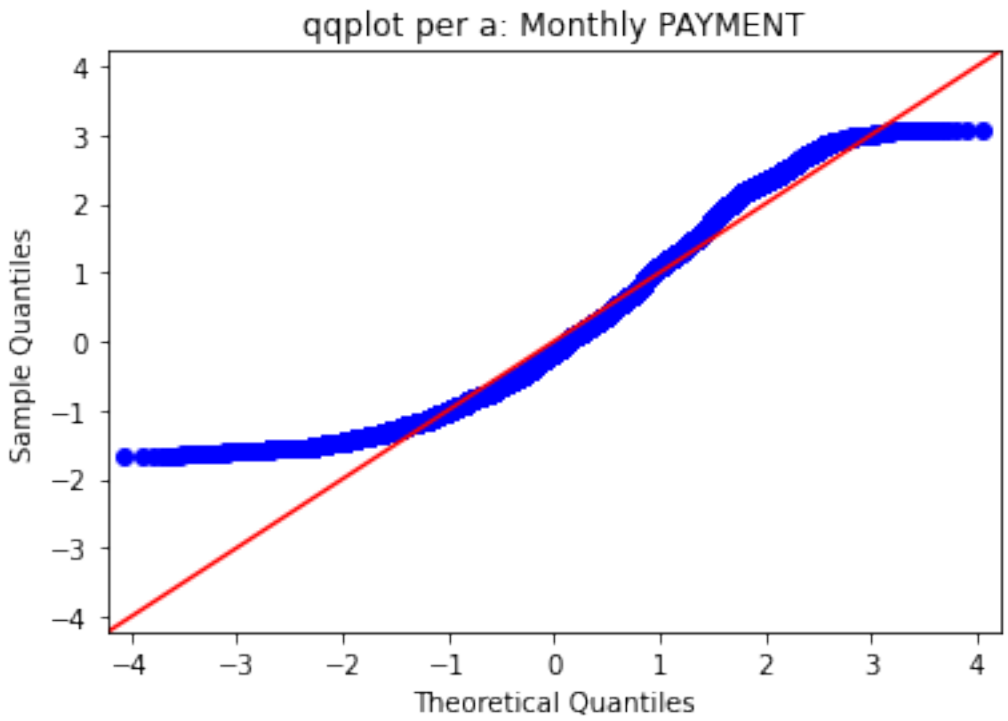
[31]:
```
#Decididim fer la prova de Kolmogorov-Smirnov i el gràfic QQ per les altres 5␣
↪variables numèriques per veure si segueixen o no una distribució normal.

def prova(variable):
    media, desviacion = norm.fit(variable)
    stat, p = kstest(variable, 'norm', args=(media,desviacion))
    print('Variable:', column)
    print("-----------------------------------")
```

```
    print('Estadistics prova Kolmogorov-Smirnov=%.3f, p-value=%.3f' % (stat, p))
    qqplot(variable)
    plt.show()


cols = ['Total Amount Funded', 'Monthly PAYMENT', 'Monthly Income', 'Approx.␣
 ↪Fico Score', 'Revolving CREDIT Balance', \
        'Revolving Line Utilization','Total CREDIT Lines', 'Inquiries in the␣
 ↪Last 6 Months']
for column in cols:

        prova (df[column])
```

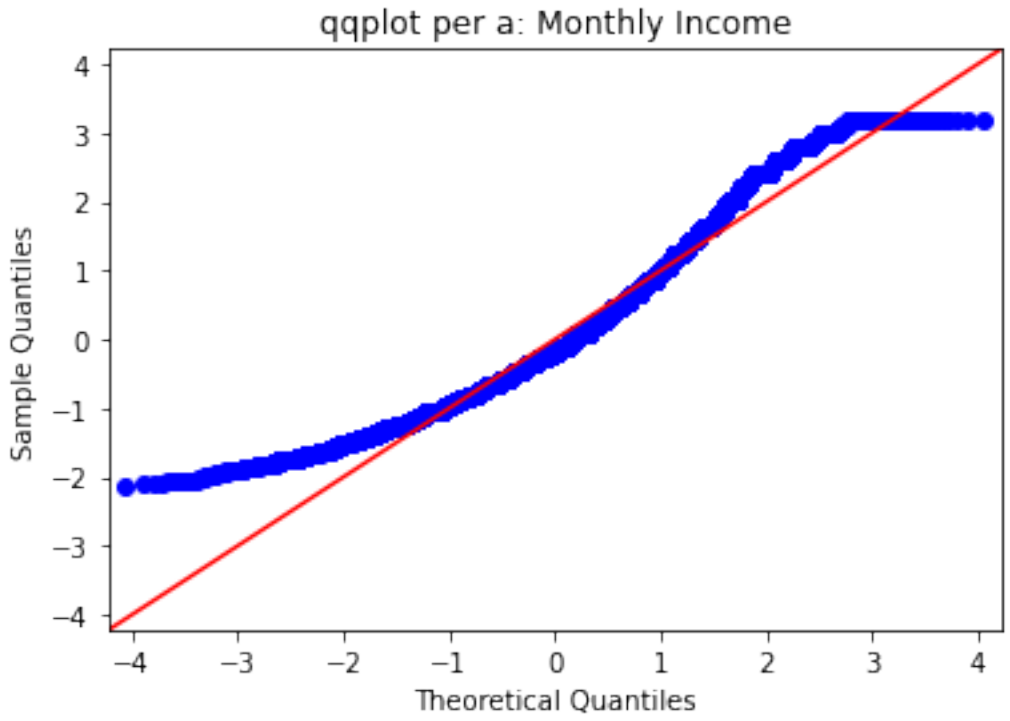
```
Variable: Total Amount Funded
-------------------------------------
Estadistics prova Kolmogorov-Smirnov=0.104, p-value=0.000
```



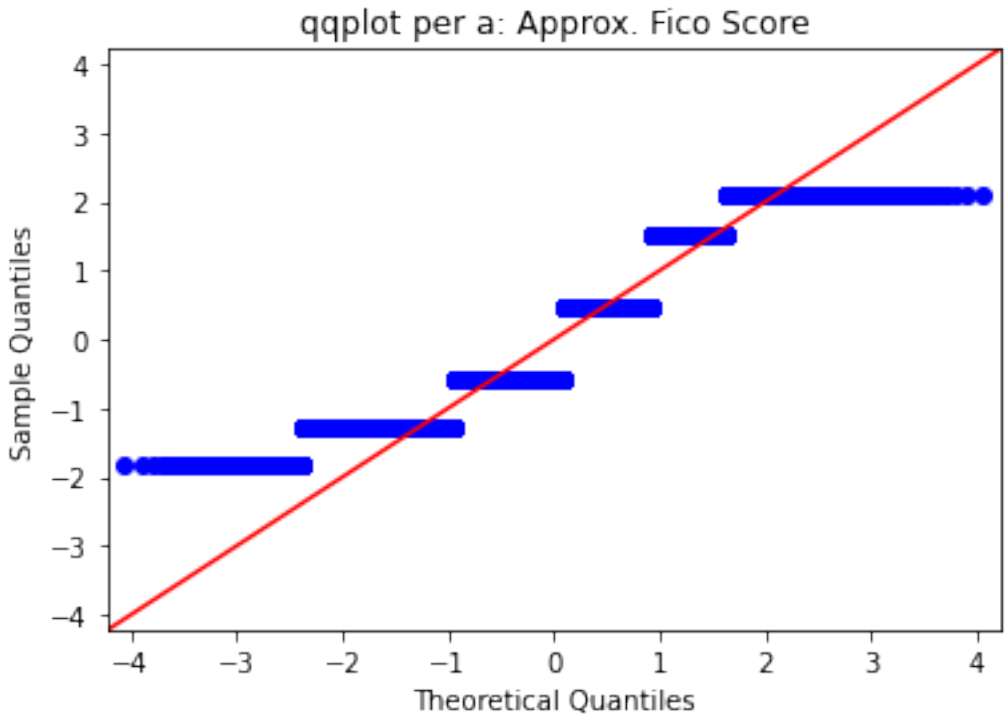

```
Variable: Monthly PAYMENT
-------------------------------------
Estadistics prova Kolmogorov-Smirnov=0.071, p-value=0.000
```



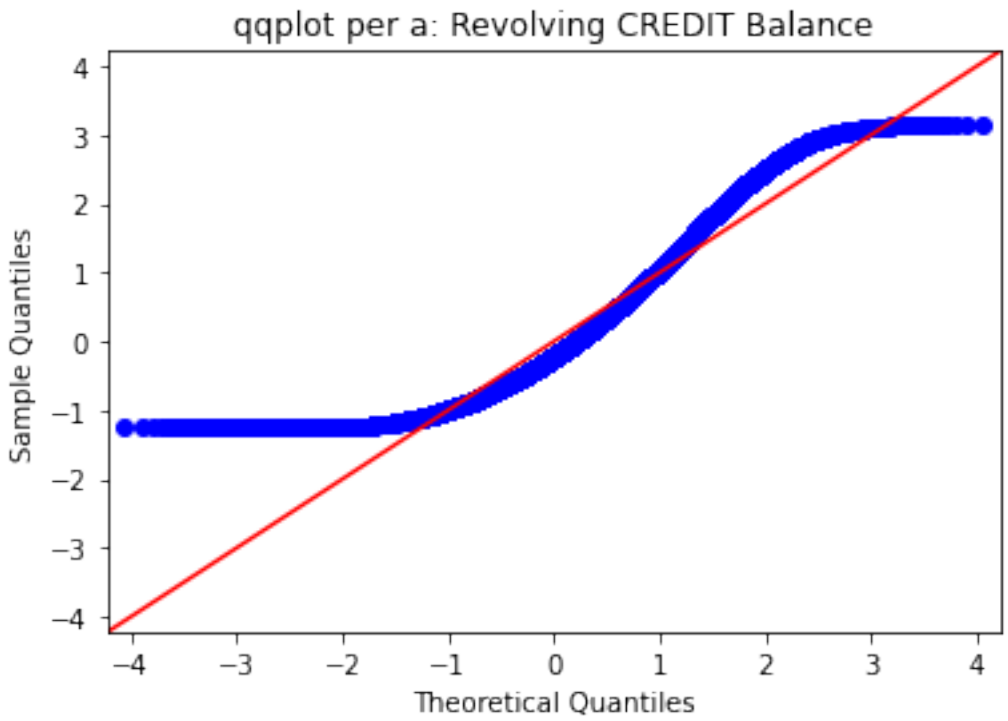

```
Variable: Monthly Income
-----------------------------------
Estadistics prova Kolmogorov-Smirnov=0.084, p-value=0.000
```



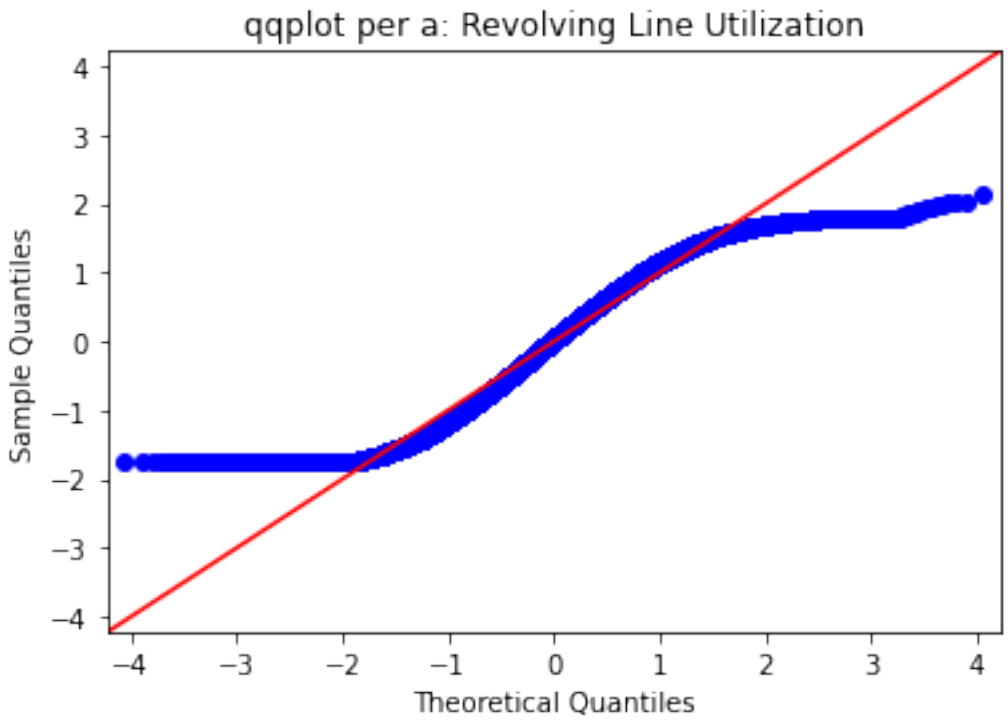

```
Variable: Approx. Fico Score
-----------------------------------
Estadistics prova Kolmogorov-Smirnov=0.249, p-value=0.000
```



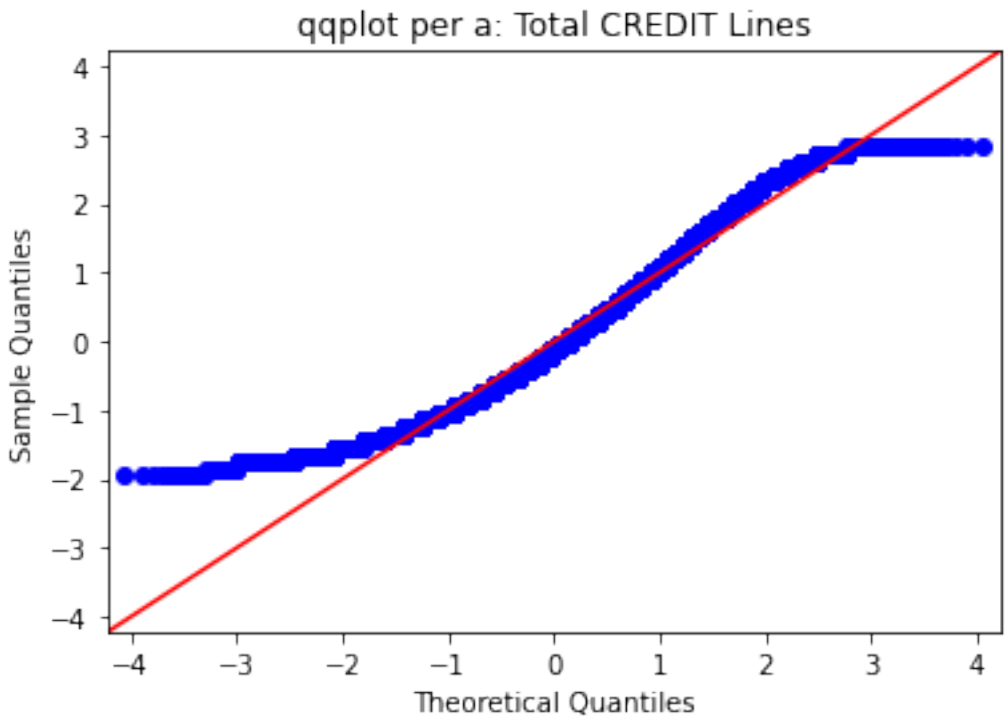

```
Variable: Revolving CREDIT Balance
-----------------------------------
Estadistics prova Kolmogorov-Smirnov=0.106, p-value=0.000
```



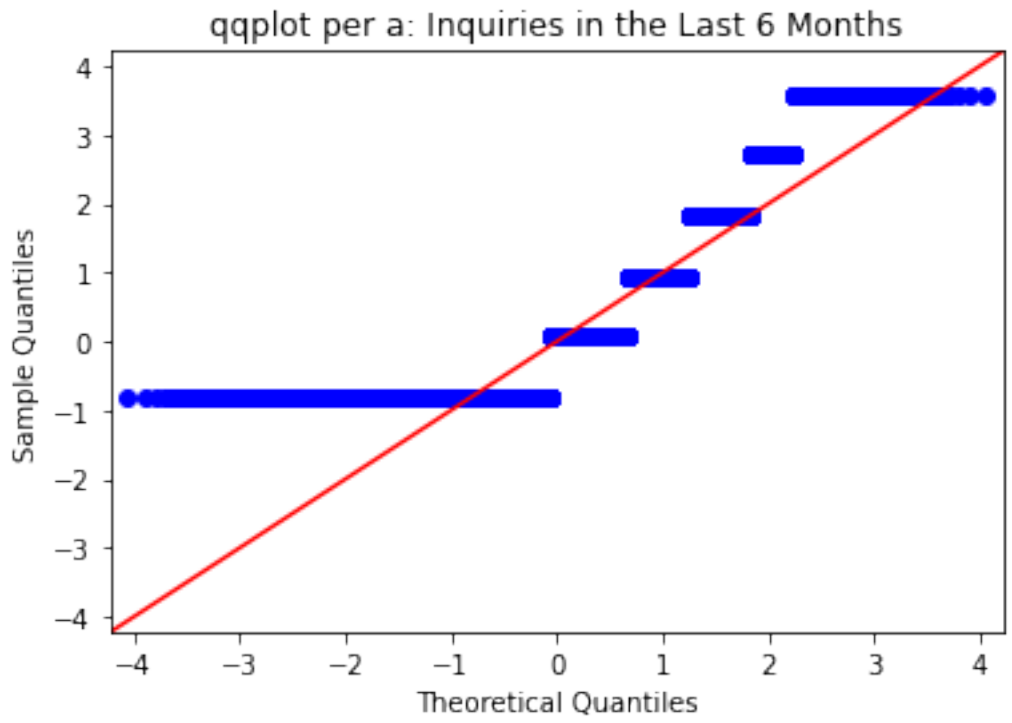

```
Variable: Revolving Line Utilization
-------------------------------------
Estadistics prova Kolmogorov-Smirnov=0.047, p-value=0.000
```




```
Variable: Total CREDIT Lines
-----------------------------------
Estadistics prova Kolmogorov-Smirnov=0.073, p-value=0.000
```




```
Variable: Inquiries in the Last 6 Months
-----------------------------------
Estadistics prova Kolmogorov-Smirnov=0.268, p-value=0.000
```



Les conclusions són:

- totes les variables numèriques tenen els p-values inferiors al nivell de significació alfa=0,05. Es rebutja la hipòtesi nul·la que les distribucions són iguals a la normal.
- Els gràfics qq no denoten normalitat.

Per tant, no es pot assumir que cap variable numèrica segueixi la distribució normal.

Comprovació de l'homogeneïtat Ara es comprova l'homogeneïtat, és a dir, la igualtat de variàncies entre els grups que volem comparar: delinquent i not delinquent, de totes les variables numèriques. Es realitzen els següents tests en funció de la distribució de les variables:

1) Test de Levene, si les dades segueixen una distribució normal.

2) Test de Flingner-Killeen, si no segueixen una distribució normal.

La hipòtesi nul·la assumeix igualtat de variàncies, de manera que si p.valors < 0,05 (nivell de significació) es rebutjarà la hipòtesi nul·la i no hi hauria igualtat entre variàncies per tant la variable no tindrà homogeneïtat. En cas contrari si p-valors fos superior a 0,05, no es pot refusar la hipòtesi nul·la i si tindríem les variàncies iguals.

[34]:
```
from scipy.stats import fligner

#Dividim el nostre conjunt de dades en els 2 grups Delinquent i Not Delinquent.
df_delinquent = df[df['Status'] == 'Delinquent']
df_notdelinquent = df[df['Status'] == 'Not Delinquent']
```

```
#Fem el test de Fligner-Killeen, perque la distribució de les variables no␣
 ↪segueix una distribució normal,
def test_fligner(variable_01,variable_02):
    stat_f, p_f = fligner(variable_01, variable_02)
    print('Test de Flinger, estadístic=%.3f, p-value=%.3f' % (stat_f, p_f))

cols = ['Total Amount Funded', 'Monthly PAYMENT', 'Monthly Income', 'Approx.␣
 ↪Fico Score', 'Revolving CREDIT Balance', \
        'Revolving Line Utilization','Total CREDIT Lines', 'Inquiries in the␣
 ↪Last 6 Months']

for column in cols:
        print('Variable:', column)
        test_fligner (df_delinquent[column], df_notdelinquent[column])
```

```
Variable: Total Amount Funded
Test de Flinger, estadístic=0.117, p-value=0.732
Variable: Monthly PAYMENT
Test de Flinger, estadístic=7.612, p-value=0.006
Variable: Monthly Income
Test de Flinger, estadístic=9.790, p-value=0.002
Variable: Approx. Fico Score
Test de Flinger, estadístic=146.883, p-value=0.000
Variable: Revolving CREDIT Balance
Test de Flinger, estadístic=0.821, p-value=0.365
Variable: Revolving Line Utilization
Test de Flinger, estadístic=44.535, p-value=0.000
Variable: Total CREDIT Lines
Test de Flinger, estadístic=20.722, p-value=0.000
Variable: Inquiries in the Last 6 Months
Test de Flinger, estadístic=261.336, p-value=0.000
```

Les conclusions per aquesta prova del test Fligner-Killeen són:

Les variables per les quals s'accepta la hipòtesi nul·la, i per tant les variàncies pels dos grups són iguals perquè p-value és superior al nivell de significació alfa=0,05: * Total Amount Funded amb p-value= 0.732. * Revolving CREDIT Balance amb p-value= 0.365.

Les variables que tenen les variàncies pels dos grups diferents perquè p-value és inferior al nivell de significació alfa=0,05 són:

- Monthly Income amb p-value= 0.002
- Monthly PAYMENT amb p-value= 0.006
- Approx. Fico Score amb p-value= 0.000
- Revolving Line Utilization amb p-value= 0.000
- Total CREDIT Lines amb p-value= 0.000
- Total Inquiries in the Last 6 Months amb p-value= 0.000.

La variable "Monthly Income" no té una distribució normal i tampoc té homogeneïtat, amb aquest

escenari de no normalitat i no homogeneïtat, ja es pot realitzar la primera de les proves estadístiques.

1.4.3 Aplicació de les proves estadístiques

Contrast d'hipòtesi. Els ingressos mensuals per grup dels no morosos (not delinquent) és superior als ingressos mensuals dels morosos (delinquent). Quan no existeix normalitat ni homogeneïtat (la igualtat de variàncies entre els grups que volem comparar delinqüent i no delinqüent) per la variable "Monthly Income", s'ha d'aplicar la prova Wilcoxon (si les dades són dependents) o Mann-Whitney (quan els grups de dades siguin independents).

En aquest cas els grups de dades són independents, per tant es realitza la prova no paramètrica Mann-Whitney.

El contrast d'hipòtesis de dues mostres sobre la diferència de mitjanes dels ingressos mensuals, es pot plantejar com el contrast unilateral següent:

```
H0: μ1(mijana_ingressos_mensuals_delinquent) = μ2(mijana_ingressos_mensuals_notdelinquent)
H1: μ1(mijana_ingressos_mensuals_delinquent) < μ2(mijana_ingressos_mensuals_notdelinquent)
```

[35]:
```
#Calculem estadístic i p-value prova Mann-Whitney.
stat_ch, p_ch = stats.mannwhitneyu(df_delinquent['Monthly␣
 ↪Income'],df_notdelinquent['Monthly Income'], alternative='less')
print('Test de Mann-Whitney, estadístic=%.3f, p-value=%.3f' % (stat_ch, p_ch))
```

```
Test de Mann-Whitney, estadístic=39410973.000, p-value=0.000
```

La conclusió és:

1. el valor de p=0,000 és inferior a alfa=0,05 per tant es rebutja la hipòtesi nul·la que la mitjana del salari mensual dels morosos és igual a la mitjana mensual dels no morosos.
2. s'accepta la hipòtesi alternativa, que la mitjana dels ingressos mensuals de les persones que demanen un préstec i són moroses és inferior a la mitjana dels ingressos mensuals dels quals no són morosos.

Correlació entre variables numèriques En aquesta secció es realitza una correlació N a N entre totes les variables numèriques. Es desitja saber quines es poden arribar a explicar en funció d'altres.

Aquesta prova es realitza amb l'objectiu de poder descartar algunes variables en anàlisis futurs. No és estrictament lligada a l'objectiu principal però pot tenir molta utilitat en el futur. Algunes proves estadístiques i models no són fiables si les variables independents presenten correlació entre elles.

L'associació entre dues variables numèriques es pot expressar com a positiva, negativa, o neutre.

En una associació positiva o negativa, una variable tendeix a crèixer o decrèixer a mesura que l'altre incrementa. En una associació neutre, no es presenta cap relació entre les variables.

Quan les variables no segueixen una distribució gaussiana, es pot optar per realitzar una correlació no paramètrica (rank). En concret, interessa realitzar la prova de Spearman's Rank.

Aquesta prova pren com a hipòtesi nul·la que les dues variables no són correlacionades. La hipòtesi alternativa estipula que sí que ho són en la població.

La prova és similar a una regressió, però no tant restrictiva. En la correlació de Spearman, es captura la tendència de la variable a incrementar o decrementar en funció de l'altre. No necessariament aquesta tendència ha de ser lineal.

Per entrar en detall, s'il·lustra un exemple de la prova de correlació entre el pagament mensual i la quantitat del prèstec.

[7]:
```
from scipy.stats import spearmanr


coef, p = spearmanr(df['Monthly PAYMENT'], df['Total Amount Funded'])
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
    print('Les variables no són correlacionades (no es pot rebutjar H0) p=%.3f'␣
 ↪% p)
else:
    print('Les variables són correlacionades (es rebutja H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: 0.970
Les variables són correlacionades (es rebutja H0) p=0.000
```

Com s'havia intuït anteriorment, el pagament mensual d'un préstec i la seva quantitat final són estrictament relacionades.

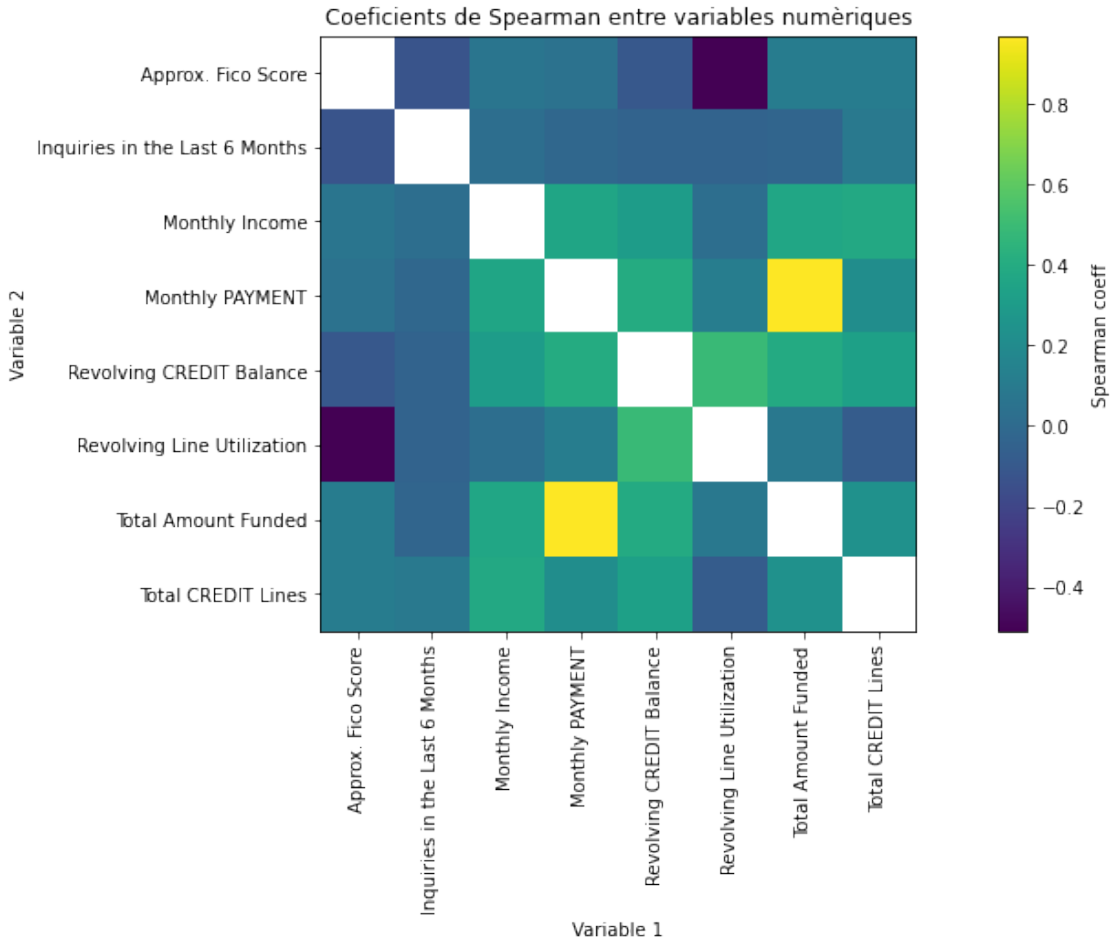
El coeficient indica que a mesura que s'incrementa una, l'altra creix en conseqüència, i pot variar entre -1 i 1. Aquest significa:

- -1: associació molt forta de forma negativa.
- 0: no es presenta associació.
- 1: associació molt forta positiva.

Observem com s'aplica el coeficient de correlació entre totes les variables numèriques.

[16]:
```
def spearmanCorrRank(v1,v2):
    coef, p = spearmanr(v1, v2)
    if p > 0.05:
        return (coef, 0, p)
    else:
        return (coef, 1, p)

cols = ['Total Amount Funded', 'Monthly PAYMENT', 'Monthly Income', 'Approx.␣
 ↪Fico Score', 'Revolving CREDIT Balance', \
        'Revolving Line Utilization','Total CREDIT Lines', 'Inquiries in the␣
 ↪Last 6 Months']
corr = []
for col in cols:
```

```
    for col2 in cols:
        if (col != col2):
            res = spearmanCorrRank(df[col],df[col2])
            corr = corr + [[col, col2, res[0], res[1], res[2]]]

corr_df = pd.DataFrame(corr,␣
 ↪columns=['Variable1','Variable2','Coef','AreRelated','p-value'])

# Mostrar la matriu de coeficients
grid_pivot = corr_df.pivot(index='Variable1',columns='Variable2',values='Coef')
fig = plt.figure(figsize=(16,6))
plt.imshow(grid_pivot)
cbar = plt.colorbar()
cbar.ax.set_ylabel('Spearman coeff')
plt.xticks(range(len(grid_pivot.columns)), grid_pivot.columns.values,␣
 ↪rotation='vertical')
plt.yticks(range(len(grid_pivot.index)), grid_pivot.index.values)
ax = plt.gca()
ax.set_xlabel("Variable 1")
ax.set_ylabel("Variable 2")
plt.title("Coeficients de Spearman entre variables numèriques")
plt.show()
```



S'observen algunes característiques importants:

- La relació més potent es dóna entre el pagament mensual i la quantitat del préstec.
- Existeix una relació negativa entre la puntuació Fico i l'ús del crèdit. S'intueix que a més puntuació Fico, menor percentatge d'ús de la línia de crèdit.
- El balanç del crèdit i el seu ús també semblen estar positivament relacionats, així com el pagament mensual-salari, el pagament mensual-balanç del crèdit.
- El nombre de consultes no sembla relacionat amb cap altra variable.

Per tant, es pot concloure que existeix una relació multivariable entre les dades que representen els imports. Salari mensual, pagament mensual... Els imports varien tots en la mateixa direcció. Molt probablement, amb l'ús de tècniques avançades de selecció de variables (com PCA), es podrien trobar els components principals de totes aquestes variables.

Regressió logística Aquesta pràctica contempla la creació d'un classificador que utilitzi la regressió logística com a model principal.

En la regressió logística, la variable classificadora és binària, fet que és especialment útil en aquest cas, ja que es vol decidir si el préstec resulta fraudulent o no.

La regressió logística utilitza la funció sigmoide com a funció de classificació, on un valor superior a 0,5 significa que es classifica la instància com a la classe "certa" de la variable objectiva.

La regressió divideix les dades en dos grups, independents o dependents. Les variables independents són aquelles que es volen utilitzar per a construir i avaluar el model. Es trien tres variables d'alta importància per tal de construir el model. La tria és per poder visualitzar com varia el criteri de decisió en funció de les variables. Aquestes són:

- Inquiries in the last 6 months.
- Last record.
- Revolving line utilization.

La variable predictora és Status.

Es descarta utilitzar la puntuació Fico ja que és molt dependent amb la variable "Inquiries in the last 6 months". Es recomana no utilitzar variables que presentin correlació en el conjunt de variables independents. Font

A més a més, segons la literatura, no és necessari comprovar la normalitat i la homogeneïtat de la variància en un model de regressió logística, ja que les seves mètriques i mesures són independents d'aquestes propietats. Es pot fer servir en qualsevol distribució. Font

Abans de començar, s'han de tenir aquests aspectes en compte:

- La regressió logística no requereix que les dades siguin normals. Tampoc cal estandarditzar ni normalitzar.
- Sí que es requereix que les variables categòriques es codifiquin en numèriques. Això es pot aconseguir de dues formes:
  1. Assignant a cada possible classe un nombre enter. Té com a principal desavantatge que necessita coneixement sobre quina és la relació d'ordre entre les classes. Altrament el model pot resultar erroni, ja que assigna un coeficient a aquesta variable.
  2. Codificar les classes de manera "dummy". Cada possible classe passa ser una nova variable que pot prendre els valors 0,1 segons si la classe és la donada en aquella instància. Resulta més adequat si no es coneix la relació d'ordre.
- La regressió logística requereix que el dataset sigui equilibrat (mateix nombre d'instàncies en cada classe de la variable a predir). Altrament presenta un gran bias cap a una de les classes.

Per tant, el primer pas es tracta de realitzar un mostreig per tal d'equilibrar el dataset. Es decideix realitzar un upsampling (augmentar el nombre d'instàncies de la classe "delinqüent", que es troba en minoria.

[6]:
```
from sklearn.utils import resample
# Partir els datasets
df_majority = df[df.Status=='Not Delinquent']
df_minority = df[df.Status=='Delinquent']

# Realitzar upsampling
df_minority_upsampled = resample(df_minority, replace=True,␣
 ↪n_samples=len(df_majority),random_state=123)
```

```
# Combinar els datasets
df_upsampled = pd.concat([df_majority, df_minority_upsampled])

# Imprimir el nombre de classes
print(df_upsampled.Status.value_counts())
```

```
Delinquent        37222
Not Delinquent    37222
Name: Status, dtype: int64
```

En segon lloc, es creen les variables dummy a partir de les classes de "Last Record"

[10]:
```
# Codificar la variable categorica de last record com a dummy (múltiples␣
 ↪variables 0,1 segons la categoria)
df_dummy = pd.concat([pd.get_dummies(df_upsampled['Last Record']),␣
 ↪df_upsampled], axis=1)
df_dummy.head(1)
```

[10]:

	JUST NOW	LAST FIVE	LAST TEN	LAST YEAR	NEVER	Unnamed: 0	Total Amount Funded	Monthly PAYMENT	Home Ownership	Monthly Income	Approx. Fico Score	Total CREDIT Lines	Revolving CREDIT Balance	Revolving Line Utilization	Inquiries in the Last 6 Months	Last Delinquency	Last Record	EmployedCat	Status
0	0	0	0	0	1	0	500	15.67	RENT	275.0	732.0	3.0	0.0	0.0	0.0	NEVER	NEVER	0_TO_4_YRS	Not Delinquent

A continuació ja es pot crear el model de regressió logística. Per avaluar el model es realitza una partició de les dades en train i test al 80/20.

El model es crea sobre train i s'avalua sobre el conjunt de test.

[11]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics


# Seleccionar les característiques
feature_cols = ['Inquiries in the Last 6 Months',\
                'JUST NOW','LAST FIVE','LAST TEN','LAST YEAR','NEVER',\
                'Revolving Line Utilization']
```

```
# Crear el dataset de train i de test
X = df_dummy[feature_cols]
y = df_dummy.Status

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20,random_state=0)

# Crear l'objecte del model
logreg = LogisticRegression()

# Ajustar el model
logreg.fit(X_train,y_train)

# Fer les prediccions en test
y_pred = logreg.predict(X_test)

# Extreure la matriu de confusió
labels = np.unique(y_test)
cnf_matrix = metrics.confusion_matrix(y_test, y_pred, labels=labels)
cnf_matrix_df = pd.DataFrame(cnf_matrix, index=labels, columns=labels)
cnf_matrix_df.index.name = 'Real'
cnf_matrix_df.columns.name = 'Predicted'
cnf_matrix_df
```

[11]:

Real \ Predicted	Delinquent	Not Delinquent
Delinquent	3862	3663
Not Delinquent	2323	5041

La matriu de confusió ja deixa veure que els resultats no són excel·lents. En una situació real, la mètrica que més es desitjaria conèixer és el **recall**, que se centra en la tassa de verdaders positius envers la de falsos negatius.

Interessa detectar el màxim de casos de frau i evitar que una persona s'escapi quan en realitat acaba comentent frau. Per tant és la relació entre True Positive i False Negative, que és el **recall**. Etiquetar un client que pagarà com a fraudulent i denegar el préstec no és tan costós com assumir un cas de frau.

[12]:
```
# Imprimir el recall
print("Recall:",metrics.recall_score(y_test, y_pred, average="binary",␣
 ↪pos_label="Delinquent"))
```

```
Recall: 0.5132225913621262
```

Com es pot observar, és un model bastant dolent per a aquesta mètrica. Només detecta el 50% dels casos de delinqüència.

A continuació, s'interpreten els coeficients del model de regressió:

[13]:
```
# Extreure els coeficients de la regressió
coefficients = pd.concat([pd.DataFrame(X.columns),pd.DataFrame(np.
 ↪transpose(logreg.coef_))], axis=1)
coefficients.columns = ['Variable','Coeficient']
coefficients
```

[13]:

	Variable	Coeficient
0	Inquiries in the Last 6 Months	-0.320222
1	JUST NOW	-0.777772
2	LAST FIVE	0.173265
3	LAST TEN	0.070891
4	LAST YEAR	-0.000721
5	NEVER	0.653048
6	Revolving Line Utilization	-0.005898

La contribució de la variable al model de regressió s'explica a partir del coeficient. Com es pot observar, a menor nombre, major serà la probabilitat d'etiquetar la instància com a delinqüent. Un coeficient positiu contribueix a l'acceptació del crèdit.

És per això que quan l'última delinqüència ha sigut recent (JUST NOW=1), el coeficient és -0,78, però si no ha comès cap delinqüència (NEVER=1), es torna a 0,65.

Tot seguit s'intenta esbrinar el mecanisme de decisió que el model utilitza per a determinar la classe d'una instància. Es comença amb una prova simple de dues instàncies:

- Client 1: Una petició de préstec d'un client que té 20 consultes els últims sis mesos, sense cap registre de frau passat, i que realitza una despesa del 80% del seu crèdit.
- Client 2: Un client sense consultes i que utilitza el 20% del seu crèdit.

[74]:
```
# Prediccions
predict_1 = [[20,0,0,0,0,1,80]]
predict_2 = [[0,0,0,0,0,1,20]]
print("Client 1: " + logreg.predict(predict_1))
print("Client 2: " + logreg.predict(predict_2))
```

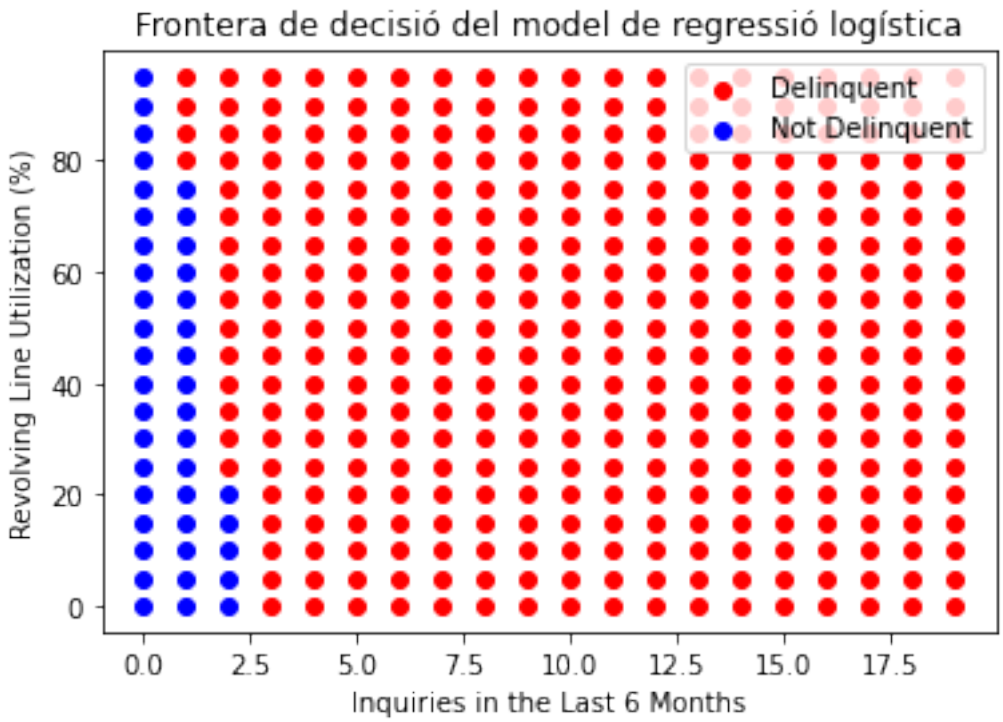
```
['Client 1: Delinquent']
['Client 2: Not Delinquent']
```

Com era d'esperar, el model discrimina correctament la relació detectada en apartats anteriors. A major nombre de consultes i a major consum del crèdit disponible, menys probable que el préstec es concedeixi.

Per últim, es pot observar com varia la predicció de les classes en funció del nombre de consultes i del percentatge de crèdit, fixat el nombre de delinqüències passades a cap.

[111]:
```
# Crear una malla de paràmetres
test_grid = []
for inquiries in range(20):
    for perc in range(0,100,5):
        test_grid = test_grid + [[inquiries,0,0,0,0,1, perc]]
```

```
# Crear el dataframe i evaluar el model
test_df = pd.DataFrame(test_grid, columns=feature_cols)
test_pred = logreg.predict(test_df)

def mapDelinquent(x):
    if (x == 'Not Delinquent'):
        return 0
    else:
        return 1

# Mostrar la frontera de decisió per al model logístic
test_df = pd.concat([test_df, pd.DataFrame(test_pred,columns=['pred'])], axis=1)
test_df['pred_int'] = test_df['pred'].map(mapDelinquent)
test_df_delinquent = test_df[test_df['pred_int'] == 1]
test_df_not_delinquent = test_df[test_df['pred_int'] == 0]
# Pinta el gràfic
plt.scatter(x=test_df_delinquent['Inquiries in the Last 6␣
 ↪Months'],y=test_df_delinquent['Revolving Line␣
 ↪Utilization'],c='red',label="Delinquent")
plt.scatter(x=test_df_not_delinquent['Inquiries in the Last 6␣
 ↪Months'],y=test_df_not_delinquent['Revolving Line␣
 ↪Utilization'],c='blue',label="Not Delinquent")
plt.legend(loc='upper right')
plt.title("Frontera de decisió del model de regressió logística")
plt.xlabel('Inquiries in the Last 6 Months')
plt.ylabel('Revolving Line Utilization (%)')
plt.show()
```



Es pot observar clarament la frontera que el model crea en aquest espai bidimensional. La funció sigmoide delimita la separació entre classificar un préstec com a fraudulent o no. Aquesta funció es basa en un component lineal (ax + by...), per tant és esperable que la frontera creada es pugui projectar en un espai lineal separable.

En definitiva, el model de regressió logística pren els valors de les variables independents i calcula el valor de la funció sigmoide en base els paràmetres del model ajustat. Aquests paràmetres ajusten la importància de cada variable, provocant que la funció resultant sumi fins a arribar al valor de sortida de 0,5, que és la línia que separa una instància com a fraudulenta o no.

Aquest model, aplicat al problema real, no resulta gaire bo. Això és un indicatiu que pot existir una relació no lineal entre les variables independents i la dependent. Possiblement un altre model com els arbres de decisió o mètodes bayesians serien més efectius de cara a aquest problema.

1.5 Conclusions

En aquesta pràctica s'ha processat i analitzat un dataset de préstecs d'un banc de crédit. L'objectiu principal de la pràctica era entendre com es relacionen les variables del model amb la variable que determina si el préstec és fraudulent o no.

Per aconseguir l'objectiu, en primer lloc s'ha requerit un procés d'anàlisi preliminar i neteja de dades en funció dels resultats. Els punts clau d'aquest procés han sigut:

- S'han seleccionat i descartat variables del model en funció de la seva contribució a determinar la fraudulència d'un préstec. Aquestes decisions s'han pres mitjançant eines de suport

visual per a les variables numèriques (com histogrames o diagrames de caixa) i tests simples de correlació per a les variables categòriques.
- S'ha detectat el significat dels valors perduts d'algunes variables. El cas més notable és el de "Months since...", on un valor faltant representava que el client no havia comès mai l'acció concreta que descriu la variable.
- També s'ha donat una interpretació bastant realista del significat dels valors perduts en la variable "Employment Length". On possiblement un valor perdut significa que el client no té feina.
- El procés de neteja ha acabat de complementar els passos anteriors. La poca existència de valors perduts en altres variables s'ha resolt amb l'eliminació de les files amb valors perduts, que no ha superat la centena.
- S'han filtrat els outliers de totes les variables, ja que la seva existència era lligada amb la distribució de la riquesa, i per tant haurien interferit en les proves estadístiques.

Després de realitzar el procés de neteja inicial, ja es podien aplicar les proves estadístiques ideades. En aquesta secció s'han realitzat tres proves estadístiques diferents amb l'objectiu d'augmentar el coneixement sobre el dataset i la relació d'interès.

En primer lloc, s'ha realitzat un contrast d'hipòtesis sobre dos grups. S'ha resolt que la mitjana del salari dels clients amb préstecs catalogats com a fraudulents és inferior a la mitjana dels que no són fraudulents. Per tant, el salari determina en una mesura determinada la categoria del préstec.

En segon lloc, s'ha dut a terme una prova de correlació entre les variables numèriques. D'aquesta prova es conclou que les variables relacionades amb els imports (salari, pagament mensual, volum del préstec, balanç de crèdit, ús del crèdit...) creixen i decreixen en conseqüència. Per exemple, a mesura que augmenta el salari, augmenten totes les altres variables esmentades. Aquest resultat és útil, ja que permet conèixer, de cara a anàlisis posteriors, que aquestes variables d'imports es podrien reduir.

Per últim, s'ha intentat posar a prova un model de regressió logística amb les tres variables més significatives del model. Encara que el classificador presenta un criteri acceptable de decisió, la mètrica del **recall** ha sigut relativament baixa. Per tant, el model no és el més adequat per a resoldre el problema de classificació dels préstecs. Això és un indicatiu que la relació entre la categoria del crèdit (fraudulent o no fraudulent) i les variables independents és no lineal, i per tant, no és indicat utilitzar un classificador lineal com la regressió logística. Altres models ja queden fora de l'àmbit dels mètodes clàssics i per tant no s'han estudiat en aquesta pràctica.

1.6 Taula de contribucions

[8]:
```
row1 = ['Investigació prèvia','ACB, JOC']
row2 = ['Redacció de les respostes','ACB, JOC']
row3 = ['Desenvolupament codi','ACB, JOC']

contribucions = pd.DataFrame([row1,row2,row3],columns=['Contribucions','Firma'])
print(contribucions.to_string(index=False))
```

```
            Contribucions     Firma
      Investigació prèvia  ACB, JOC
Redacció de les respostes  ACB, JOC
     Desenvolupament codi  ACB, JOC
```