Shanney_BCA1

August 16, 2023

DATA PREPARATION

[1]:

```
#imports and read data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import os

df = pd.read_excel("Project_KL_TL.xlsx", sheet_name="Data")
df_raw = df.copy()
```

[2]:

```
#remove unecessary columns (1)
df = df.drop(df.columns[[0,1,2,7,]], axis=1)

#remove empty value rows of borrow&outstanding column (2)
df = df[df.OS_IDR != 0]
df = df[df.PLAFOND_IDR != 0]
df = df.reset_index(drop=True)

#remove rows with null values in months columns (3)
df = df[df['AVG_9MTHS_AMT_CR'].notna()]
```

[3]:

```
#create new columns/variables
# total number of flags: yellow1 red2 court3 blacklist5
df["total_flags"] = df["REDFLAG_YELLOW"] + df["REDFLAG_RED"]*3 +␣
 ↪df["REDFLAG_INFORMASI"]*5 + df["FLAG_BLACKLIST"]*7

# % of borrowed used (OS_IDR/PLAFOND_IDR)
df["percent_used"] = df["OS_IDR"]/df["PLAFOND_IDR"]
```

[4]:

```
#drop flags
df_withflags = df.copy()
df = df.drop(df.columns[[40,41,42,43]], axis=1)
df
```

```
[4]: KOLEK DEBTOR_CATEGORY   PLAFOND_IDR        OS_IDR  HARI_TUNGGAKAN  \
     0        5          03 SME  1.250000e+09  1.250000e+09             187
     1        4          03 SME  4.850000e+08  4.850000e+08             159
     2        5          03 SME  2.000000e+09  2.000000e+09             678
     3        5          03 SME  3.500000e+09  3.500080e+09             584
     4        5          03 SME  5.250000e+08  5.250000e+08             457
     ..     …             …           …           …             …
     779      5          03 SME  1.600000e+09  1.600000e+09             401
     780      4          03 SME  1.000000e+10  1.000000e+10             128
     781      5          03 SME  6.000000e+08  6.000000e+08             278
     782      5          03 SME  1.500000e+08  1.500350e+08             370
     783      4          03 SME  6.000000e+09  6.000000e+09             126

                CUST_TYPE_CD  AVG_12MTHS_CASA  AVG_12MTHS_DPK  AVG_12MTHS_AMT_DB  \
     0    I                      5.566093e+05    5.566093e+05       2.424547e+06
     1    I                      1.672198e+06    1.672198e+06       8.259598e+06
     2    I                      8.143394e+07    8.143394e+07       1.469788e+08
     3    I                      4.692789e+06    4.692789e+06       1.406454e+07
     4    I                      7.968654e+07    7.968654e+07       1.002586e+07
     ..                  …              …             …                …
     779  O                               NaN             NaN       4.365745e+07
     780  O                               NaN             NaN       1.046247e+08
     781  I                      5.817637e+06    5.817637e+06       1.246017e+07
     782  I                      1.970590e+06    1.970590e+06       6.376608e+06
     783  I                      4.598696e+06    4.598696e+06       5.797037e+08

          AVG_12MTHS_AMT_CR  …  AVG_MUTASI_DB  AVG_MUTASI_CR  FREK_DB  FREK_CR  \
     0         7.020616e+06  …   3.645588e+06   2.752862e+06      7.0     23.0
     1         7.461819e+06  …   2.476667e+06   7.500158e+05      4.0     21.0
     2         2.166592e+08  …   1.287549e+08   1.178752e+08     36.0     19.0
     3         1.436280e+07  …   7.950654e+06   1.566660e+07     45.0     60.0
     4         1.844738e+07  …   9.420308e+06   8.468405e+06     80.0     40.0
     ..                 …  …            …            …      …      …
     779       4.630667e+07  …            NaN            NaN      NaN      NaN
     780       5.201539e+07  …   9.333333e+06   2.169097e+07      3.0      5.0
     781       1.957844e+07  …   1.344000e+07   1.520000e+07      9.0      4.0
     782       1.110582e+07  …   1.266750e+06   5.300000e+06      8.0      6.0
     783       5.480280e+08  …   1.000000e+08   0.000000e+00      2.0      0.0

          FLAG_RESTRU_COV  FLAG_DEFERRED_COV  FLAG_RESTRU_COV_21  \
     0                  1                  1                   1
     1                  1                  1                   1
     2                  0                  0                   0
     3                  1                  1                   1
     4                  1                  1                   1
     ..               …                …                 …
     779                0                  0                   0
```

```
780                0                  0                   0
781                0                  0                   0
782                0                  0                   0
783                1                  1                   1

     FLAG_DEFERRED_COV_21  total_flags  percent_used
0                       1           16      1.000000
1                       1            6      1.000000
2                       0           11      1.000000
3                       1           12      1.000023
4                       1            9      1.000000
..                    …          …           …
779                     0            6      1.000000
780                     0            2      1.000000
781                     0            1      1.000000
782                     0            7      1.000233
783                     1           12      1.000000

[781 rows x 42 columns]
```

[5]:

```
#create new columns/variables
# % decrease in average money entry 12->9->6.... (AVG_12MTHS_AMT_CR) negative␣
 ↪is good
df["12to9"] = (df["AVG_12MTHS_AMT_CR"] - df["AVG_9MTHS_AMT_CR"])/
 ↪df["AVG_12MTHS_AMT_CR"]
df["9to6"] = (df["AVG_9MTHS_AMT_CR"] - df["AVG_6MTHS_AMT_CR"])/
 ↪df["AVG_9MTHS_AMT_CR"]
df["6to3"] = (df["AVG_6MTHS_AMT_CR"] - df["AVG_3MTHS_AMT_CR"])/
 ↪df["AVG_6MTHS_AMT_CR"]
df["3to1"] = (df["AVG_3MTHS_AMT_CR"] - df["AVG_MUTASI_CR"])/
 ↪df["AVG_3MTHS_AMT_CR"]
```

[6]:

```
#create new columns/variables
#IF POSITIVE BAD SIGN, IF NEGATIVE GOOD SIGN
from scipy.stats import linregress
import math

df = df.reset_index(drop=True)
df["slope"] = df["12to9"]
df["intercept"] = df["12to9"]

for i in range(len(df)-1):

    if math.isnan(df["6to3"][i]) or math.isnan(df["3to1"][i]):
        if math.isnan(df["6to3"][i]):
            x_val = [1,2]
            y_val = [df["12to9"][i], df["9to6"][i]]
            slope, intercept, r_value, p_value, std_err =␣
 ↪linregress(x_val,y_val)
            df["slope"][i] = slope
        else:
            x_val = [1,2,3]
            y_val = [df["12to9"][i], df["9to6"][i], df["6to3"][i]]
            slope, intercept, r_value, p_value, std_err =␣
 ↪linregress(x_val,y_val)
            df["slope"][i] = slope
    else:
        if i == 443:
            df["slope"][i] = 0
        else:
            x_val = [1,2,3,4]
            y_val = [df["12to9"][i], df["9to6"][i], df["6to3"][i],␣
 ↪df["3to1"][i]]
            slope, intercept, r_value, p_value, std_err =␣
 ↪linregress(x_val,y_val)
            df["slope"][i] = slope
            df["intercept"][i] = intercept
pd.set_option('display.max_rows', 7)
```

```
/tmp/ipykernel_430/2759182025.py:30: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["slope"][i] = slope
/tmp/ipykernel_430/2759182025.py:31: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["intercept"][i] = intercept
/tmp/ipykernel_430/2759182025.py:17: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["slope"][i] = slope
/tmp/ipykernel_430/2759182025.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["slope"][i] = slope
/tmp/ipykernel_430/2759182025.py:25: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["slope"][i] = 0
```

[7]:

```
pd.set_option('display.max_rows',100)
df.isnull().any()
```

[7]:

```
KOLEK                   False
DEBTOR_CATEGORY         False
PLAFOND_IDR             False
OS_IDR                  False
HARI_TUNGGAKAN          False
CUST_TYPE_CD            False
AVG_12MTHS_CASA          True
AVG_12MTHS_DPK           True
AVG_12MTHS_AMT_DB       False
AVG_12MTHS_AMT_CR       False
AVG_12MTHS_FREK_DB      False
AVG_12MTHS_FREK_CR      False
AVG_9MTHS_CASA           True
AVG_9MTHS_DPK            True
AVG_9MTHS_AMT_DB        False
AVG_9MTHS_AMT_CR        False
AVG_9MTHS_FREK_DB       False
AVG_9MTHS_FREK_CR       False
AVG_6MTHS_CASA           True
AVG_6MTHS_DPK            True
AVG_6MTHS_AMT_DB        False
AVG_6MTHS_AMT_CR        False
AVG_6MTHS_FREK_DB       False
AVG_6MTHS_FREK_CR       False
AVG_3MTHS_CASA           True
AVG_3MTHS_DPK            True
AVG_3MTHS_AMT_DB         True
AVG_3MTHS_AMT_CR         True
AVG_3MTHS_FREK_DB        True
AVG_3MTHS_FREK_CR        True
SALDO_AVG_CASA           True
SALDO_AVG_DPK            True
AVG_MUTASI_DB            True
AVG_MUTASI_CR            True
FREK_DB                  True
FREK_CR                  True
FLAG_RESTRU_COV         False
FLAG_DEFERRED_COV       False
```

```
FLAG_RESTRU_COV_21      False
FLAG_DEFERRED_COV_21    False
total_flags             False
percent_used            False
12to9                    True
9to6                     True
6to3                     True
3to1                     True
slope                    True
intercept                True
dtype: bool
```

[8]:

```
#for null rows in 6to3 and 3to1, use slope to predict
for i in range(len(df)-1):
    if math.isnan(df["3to1"][i]):
        df["3to1"][i] = (df["slope"][i])*4 + df["intercept"][i]
        if math.isnan(df["6to3"][i]):
            df["6to3"][i] = (df["slope"][i])*3 + df["intercept"][i]

#for null rows in avg3 and avg1, use above to calculate
for i in range(len(df)-1):
    if math.isnan(df["AVG_MUTASI_CR"][i]):
        if math.isnan(df["AVG_3MTHS_AMT_CR"][i]):
            df["AVG_3MTHS_AMT_CR"][i] = df["AVG_6MTHS_AMT_CR"][i] -␣
 ↪(df["AVG_6MTHS_AMT_CR"][i])*(df["6to3"][i])
        df["AVG_MUTASI_CR"][i] = df["AVG_3MTHS_AMT_CR"][i] -␣
 ↪(df["AVG_3MTHS_AMT_CR"][i])*(df["3to1"][i])

#create new column 12to3
df["12to3"] = (df["AVG_12MTHS_AMT_CR"] - df["AVG_3MTHS_AMT_CR"])/
 ↪df["AVG_12MTHS_AMT_CR"]
df["12to6DB"] = (df["AVG_12MTHS_AMT_DB"] - df["AVG_6MTHS_AMT_DB"])/
 ↪df["AVG_12MTHS_AMT_DB"]
```

```
/tmp/ipykernel_430/3635947419.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["3to1"][i] = (df["slope"][i])*4 + df["intercept"][i]
/tmp/ipykernel_430/3635947419.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["6to3"][i] = (df["slope"][i])*3 + df["intercept"][i]
/tmp/ipykernel_430/3635947419.py:12: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["AVG_3MTHS_AMT_CR"][i] = df["AVG_6MTHS_AMT_CR"][i] -
(df["AVG_6MTHS_AMT_CR"][i])*(df["6to3"][i])
/tmp/ipykernel_430/3635947419.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["AVG_MUTASI_CR"][i] = df["AVG_3MTHS_AMT_CR"][i] -
(df["AVG_3MTHS_AMT_CR"][i])*(df["3to1"][i])
```

[9]:

```
#identify correlation
#retain one representative column from the correlated group
correlation_matrix = df.corr().abs()
repetitive_columns = set()
correlated_columns = []

for column in correlation_matrix.columns:
    correlated_columns_list = correlation_matrix.index[
        (correlation_matrix[column] >= 0.80) & (correlation_matrix[column] <␣
 ↪1)].tolist()
    if correlated_columns_list:
        repetitive_columns.update(correlated_columns_list)
        correlated_columns.extend([(column, col, correlation_matrix.loc[column,␣
 ↪col]) for col in correlated_columns_list])

representative_columns = []
for group in correlated_columns:
    representative_columns.append(min(group[:-1], key=lambda col: df[col].
 ↪nunique()))

selected_columns = list(set(df.columns) - repetitive_columns)
print(selected_columns)
print(repetitive_columns)
print("pairs")
for pair in correlated_columns:
    col1, col2, correlation_coefficient = pair
    print(f"{col1} - {col2}: {correlation_coefficient}")
```

```
['CUST_TYPE_CD', 'HARI_TUNGGAKAN', 'KOLEK', 'intercept', 'DEBTOR_CATEGORY',
'9to6', '12to3', 'percent_used', '6to3', 'total_flags', '12to9', '12to6DB']
{'AVG_9MTHS_FREK_DB', 'AVG_MUTASI_DB', 'AVG_6MTHS_FREK_DB', 'AVG_3MTHS_FREK_DB',
'AVG_6MTHS_FREK_CR', 'AVG_12MTHS_FREK_DB', 'SALDO_AVG_DPK',
'AVG_12MTHS_FREK_CR', 'AVG_9MTHS_DPK', 'AVG_3MTHS_AMT_CR', 'SALDO_AVG_CASA',
'AVG_3MTHS_AMT_DB', 'AVG_12MTHS_CASA', 'FLAG_DEFERRED_COV_21',
'AVG_12MTHS_AMT_CR', 'AVG_9MTHS_AMT_CR', 'AVG_3MTHS_FREK_CR', 'AVG_6MTHS_DPK',
'OS_IDR', 'AVG_9MTHS_CASA', 'AVG_6MTHS_AMT_DB', 'PLAFOND_IDR', 'AVG_MUTASI_CR',
'AVG_3MTHS_DPK', 'AVG_9MTHS_AMT_DB', 'slope', 'AVG_6MTHS_AMT_CR',
'AVG_12MTHS_AMT_DB', 'FLAG_RESTRU_COV_21', 'AVG_3MTHS_CASA', 'AVG_12MTHS_DPK',
'FLAG_RESTRU_COV', '3to1', 'FREK_DB', 'FREK_CR', 'AVG_6MTHS_CASA',
'AVG_9MTHS_FREK_CR', 'FLAG_DEFERRED_COV'}
pairs
PLAFOND_IDR - OS_IDR: 0.9675259573874502
OS_IDR - PLAFOND_IDR: 0.9675259573874502
AVG_12MTHS_CASA - AVG_12MTHS_DPK: 0.9493222102815314
AVG_12MTHS_CASA - AVG_9MTHS_CASA: 0.922046293438644
AVG_12MTHS_CASA - AVG_9MTHS_DPK: 0.8261522176906884
AVG_12MTHS_CASA - AVG_6MTHS_CASA: 0.9349285683477018
AVG_12MTHS_CASA - AVG_6MTHS_DPK: 0.8394964680892113
AVG_12MTHS_CASA - AVG_3MTHS_CASA: 0.9349794817389087
AVG_12MTHS_CASA - AVG_3MTHS_DPK: 0.8826528002263704
AVG_12MTHS_CASA - SALDO_AVG_CASA: 0.8731082022551883
AVG_12MTHS_DPK - AVG_12MTHS_CASA: 0.9493222102815314
AVG_12MTHS_DPK - AVG_9MTHS_CASA: 0.8880418710694625
AVG_12MTHS_DPK - AVG_9MTHS_DPK: 0.9277169171272124
AVG_12MTHS_DPK - AVG_6MTHS_CASA: 0.8953570187400136
AVG_12MTHS_DPK - AVG_6MTHS_DPK: 0.9354978120173904
AVG_12MTHS_DPK - AVG_3MTHS_CASA: 0.8866970093695352
AVG_12MTHS_DPK - AVG_3MTHS_DPK: 0.9418253268576017
AVG_12MTHS_DPK - SALDO_AVG_CASA: 0.8351773869474667
AVG_12MTHS_DPK - SALDO_AVG_DPK: 0.8860154482226915
AVG_12MTHS_AMT_DB - AVG_12MTHS_AMT_CR: 0.8934429486300833
AVG_12MTHS_AMT_DB - AVG_9MTHS_AMT_DB: 0.9903833181388703
AVG_12MTHS_AMT_DB - AVG_9MTHS_AMT_CR: 0.8764527891572244
AVG_12MTHS_AMT_DB - AVG_6MTHS_AMT_DB: 0.9800692541843432
AVG_12MTHS_AMT_DB - AVG_6MTHS_AMT_CR: 0.8651577230823655
AVG_12MTHS_AMT_DB - AVG_3MTHS_AMT_DB: 0.9523610946283111
AVG_12MTHS_AMT_DB - AVG_3MTHS_AMT_CR: 0.8481648486989221
AVG_12MTHS_AMT_CR - AVG_12MTHS_AMT_DB: 0.8934429486300833
AVG_12MTHS_AMT_CR - AVG_9MTHS_AMT_DB: 0.903210848574321
AVG_12MTHS_AMT_CR - AVG_9MTHS_AMT_CR: 0.991611216796697
AVG_12MTHS_AMT_CR - AVG_6MTHS_AMT_DB: 0.8754694815786495
AVG_12MTHS_AMT_CR - AVG_6MTHS_AMT_CR: 0.9680427818087234
AVG_12MTHS_AMT_CR - AVG_3MTHS_AMT_DB: 0.833242902536464
AVG_12MTHS_AMT_CR - AVG_3MTHS_AMT_CR: 0.9037711503858314
AVG_12MTHS_FREK_DB - AVG_9MTHS_FREK_DB: 0.9923001136174481
AVG_12MTHS_FREK_DB - AVG_6MTHS_FREK_DB: 0.9683527602160383
AVG_12MTHS_FREK_DB - AVG_3MTHS_FREK_DB: 0.8098386123760128
AVG_12MTHS_FREK_CR - AVG_9MTHS_FREK_CR: 0.997885360726625
AVG_12MTHS_FREK_CR - AVG_6MTHS_FREK_CR: 0.9863250398002176
AVG_12MTHS_FREK_CR - AVG_3MTHS_FREK_CR: 0.9151082740236894
AVG_12MTHS_FREK_CR - FREK_CR: 0.8849641956401505
```

```
AVG_9MTHS_CASA - AVG_12MTHS_CASA: 0.922046293438644
AVG_9MTHS_CASA - AVG_12MTHS_DPK: 0.8880418710694625
AVG_9MTHS_CASA - AVG_9MTHS_DPK: 0.906920138893867
AVG_9MTHS_CASA - AVG_6MTHS_CASA: 0.9961626056830948
AVG_9MTHS_CASA - AVG_6MTHS_DPK: 0.9063939056819876
AVG_9MTHS_CASA - AVG_3MTHS_CASA: 0.9882943112825219
AVG_9MTHS_CASA - AVG_3MTHS_DPK: 0.9410133261731545
AVG_9MTHS_CASA - SALDO_AVG_CASA: 0.9702868548517564
AVG_9MTHS_CASA - SALDO_AVG_DPK: 0.8946835231154521
AVG_9MTHS_DPK - AVG_12MTHS_CASA: 0.8261522176906884
AVG_9MTHS_DPK - AVG_12MTHS_DPK: 0.9277169171272124
AVG_9MTHS_DPK - AVG_9MTHS_CASA: 0.906920138893867
AVG_9MTHS_DPK - AVG_6MTHS_CASA: 0.8980516696105142
AVG_9MTHS_DPK - AVG_6MTHS_DPK: 0.9951036358983637
AVG_9MTHS_DPK - AVG_3MTHS_CASA: 0.8784640801660845
AVG_9MTHS_DPK - AVG_3MTHS_DPK: 0.9764068315060225
AVG_9MTHS_DPK - SALDO_AVG_CASA: 0.8706462828524563
AVG_9MTHS_DPK - SALDO_AVG_DPK: 0.965944083473074
AVG_9MTHS_AMT_DB - AVG_12MTHS_AMT_DB: 0.9903833181388703
AVG_9MTHS_AMT_DB - AVG_12MTHS_AMT_CR: 0.903210848574321
AVG_9MTHS_AMT_DB - AVG_9MTHS_AMT_CR: 0.8934519616207203
AVG_9MTHS_AMT_DB - AVG_6MTHS_AMT_DB: 0.9902352299514118
AVG_9MTHS_AMT_DB - AVG_6MTHS_AMT_CR: 0.8849083940498239
AVG_9MTHS_AMT_DB - AVG_3MTHS_AMT_DB: 0.9622402702582691
AVG_9MTHS_AMT_DB - AVG_3MTHS_AMT_CR: 0.8660349757172218
AVG_9MTHS_AMT_CR - AVG_12MTHS_AMT_DB: 0.8764527891572244
AVG_9MTHS_AMT_CR - AVG_12MTHS_AMT_CR: 0.991611216796697
AVG_9MTHS_AMT_CR - AVG_9MTHS_AMT_DB: 0.8934519616207203
AVG_9MTHS_AMT_CR - AVG_6MTHS_AMT_DB: 0.8707052699750774
AVG_9MTHS_AMT_CR - AVG_6MTHS_AMT_CR: 0.9834799760855574
AVG_9MTHS_AMT_CR - AVG_3MTHS_AMT_DB: 0.8373152660453125
AVG_9MTHS_AMT_CR - AVG_3MTHS_AMT_CR: 0.9294851282893258
AVG_9MTHS_FREK_DB - AVG_12MTHS_FREK_DB: 0.9923001136174481
AVG_9MTHS_FREK_DB - AVG_6MTHS_FREK_DB: 0.9876854153730271
AVG_9MTHS_FREK_DB - AVG_3MTHS_FREK_DB: 0.8512629858588676
AVG_9MTHS_FREK_CR - AVG_12MTHS_FREK_CR: 0.997885360726625
AVG_9MTHS_FREK_CR - AVG_6MTHS_FREK_CR: 0.9941572478016336
AVG_9MTHS_FREK_CR - AVG_3MTHS_FREK_CR: 0.9355141573721815
AVG_9MTHS_FREK_CR - FREK_CR: 0.9088126526948385
AVG_6MTHS_CASA - AVG_12MTHS_CASA: 0.9349285683477018
AVG_6MTHS_CASA - AVG_12MTHS_DPK: 0.8953570187400136
AVG_6MTHS_CASA - AVG_9MTHS_CASA: 0.9961626056830948
AVG_6MTHS_CASA - AVG_9MTHS_DPK: 0.8980516696105142
AVG_6MTHS_CASA - AVG_6MTHS_DPK: 0.9051386313121178
AVG_6MTHS_CASA - AVG_3MTHS_CASA: 0.991968737605446
AVG_6MTHS_CASA - AVG_3MTHS_DPK: 0.9379176573068294
AVG_6MTHS_CASA - SALDO_AVG_CASA: 0.9709549565257941
AVG_6MTHS_CASA - SALDO_AVG_DPK: 0.8867351422130509
```

```
AVG_6MTHS_DPK - AVG_12MTHS_CASA: 0.8394964680892113
AVG_6MTHS_DPK - AVG_12MTHS_DPK: 0.9354978120173904
AVG_6MTHS_DPK - AVG_9MTHS_CASA: 0.9063939056819876
AVG_6MTHS_DPK - AVG_9MTHS_DPK: 0.9951036358983637
AVG_6MTHS_DPK - AVG_6MTHS_CASA: 0.9051386313121178
AVG_6MTHS_DPK - AVG_3MTHS_CASA: 0.883116740604042
AVG_6MTHS_DPK - AVG_3MTHS_DPK: 0.9731697949027607
AVG_6MTHS_DPK - SALDO_AVG_CASA: 0.8722242111957933
AVG_6MTHS_DPK - SALDO_AVG_DPK: 0.9582641302351401
AVG_6MTHS_AMT_DB - AVG_12MTHS_AMT_DB: 0.9800692541843432
AVG_6MTHS_AMT_DB - AVG_12MTHS_AMT_CR: 0.8754694815786495
AVG_6MTHS_AMT_DB - AVG_9MTHS_AMT_DB: 0.9902352299514118
AVG_6MTHS_AMT_DB - AVG_9MTHS_AMT_CR: 0.8707052699750774
AVG_6MTHS_AMT_DB - AVG_6MTHS_AMT_CR: 0.8769095172863727
AVG_6MTHS_AMT_DB - AVG_3MTHS_AMT_DB: 0.975307460412532
AVG_6MTHS_AMT_DB - AVG_3MTHS_AMT_CR: 0.8650910604578972
AVG_6MTHS_AMT_CR - AVG_12MTHS_AMT_DB: 0.8651577230823655
AVG_6MTHS_AMT_CR - AVG_12MTHS_AMT_CR: 0.9680427818087234
AVG_6MTHS_AMT_CR - AVG_9MTHS_AMT_DB: 0.8849083940498239
AVG_6MTHS_AMT_CR - AVG_9MTHS_AMT_CR: 0.9834799760855574
AVG_6MTHS_AMT_CR - AVG_6MTHS_AMT_DB: 0.8769095172863727
AVG_6MTHS_AMT_CR - AVG_3MTHS_AMT_DB: 0.8430158534869359
AVG_6MTHS_AMT_CR - AVG_3MTHS_AMT_CR: 0.9533714676515999
AVG_6MTHS_FREK_DB - AVG_12MTHS_FREK_DB: 0.9683527602160383
AVG_6MTHS_FREK_DB - AVG_9MTHS_FREK_DB: 0.9876854153730271
AVG_6MTHS_FREK_DB - AVG_3MTHS_FREK_DB: 0.9117070366960768
AVG_6MTHS_FREK_DB - FREK_DB: 0.8086264594992362
AVG_6MTHS_FREK_CR - AVG_12MTHS_FREK_CR: 0.9863250398002176
AVG_6MTHS_FREK_CR - AVG_9MTHS_FREK_CR: 0.9941572478016336
AVG_6MTHS_FREK_CR - AVG_3MTHS_FREK_CR: 0.9666599719929414
AVG_6MTHS_FREK_CR - FREK_CR: 0.9440903511651497
AVG_3MTHS_CASA - AVG_12MTHS_CASA: 0.9349794817389087
AVG_3MTHS_CASA - AVG_12MTHS_DPK: 0.8866970093695352
AVG_3MTHS_CASA - AVG_9MTHS_CASA: 0.9882943112825219
AVG_3MTHS_CASA - AVG_9MTHS_DPK: 0.8784640801660845
AVG_3MTHS_CASA - AVG_6MTHS_CASA: 0.991968737605446
AVG_3MTHS_CASA - AVG_6MTHS_DPK: 0.883116740604042
AVG_3MTHS_CASA - AVG_3MTHS_DPK: 0.9387803519308997
AVG_3MTHS_CASA - SALDO_AVG_CASA: 0.9793492236798634
AVG_3MTHS_CASA - SALDO_AVG_DPK: 0.8858169340261174
AVG_3MTHS_DPK - AVG_12MTHS_CASA: 0.8826528002263704
AVG_3MTHS_DPK - AVG_12MTHS_DPK: 0.9418253268576017
AVG_3MTHS_DPK - AVG_9MTHS_CASA: 0.9410133261731545
AVG_3MTHS_DPK - AVG_9MTHS_DPK: 0.9764068315060225
AVG_3MTHS_DPK - AVG_6MTHS_CASA: 0.9379176573068294
AVG_3MTHS_DPK - AVG_6MTHS_DPK: 0.9731697949027607
AVG_3MTHS_DPK - AVG_3MTHS_CASA: 0.9387803519308997
AVG_3MTHS_DPK - SALDO_AVG_CASA: 0.9267795944632143
```

```
AVG_3MTHS_DPK - SALDO_AVG_DPK: 0.9787557160458124
AVG_3MTHS_AMT_DB - AVG_12MTHS_AMT_DB: 0.9523610946283111
AVG_3MTHS_AMT_DB - AVG_12MTHS_AMT_CR: 0.833242902536464
AVG_3MTHS_AMT_DB - AVG_9MTHS_AMT_DB: 0.9622402702582691
AVG_3MTHS_AMT_DB - AVG_9MTHS_AMT_CR: 0.8373152660453125
AVG_3MTHS_AMT_DB - AVG_6MTHS_AMT_DB: 0.975307460412532
AVG_3MTHS_AMT_DB - AVG_6MTHS_AMT_CR: 0.8430158534869359
AVG_3MTHS_AMT_DB - AVG_3MTHS_AMT_CR: 0.8892988999738958
AVG_3MTHS_AMT_CR - AVG_12MTHS_AMT_DB: 0.8481648486989221
AVG_3MTHS_AMT_CR - AVG_12MTHS_AMT_CR: 0.9037711503858314
AVG_3MTHS_AMT_CR - AVG_9MTHS_AMT_DB: 0.8660349757172218
AVG_3MTHS_AMT_CR - AVG_9MTHS_AMT_CR: 0.9294851282893258
AVG_3MTHS_AMT_CR - AVG_6MTHS_AMT_DB: 0.8650910604578972
AVG_3MTHS_AMT_CR - AVG_6MTHS_AMT_CR: 0.9533714676515999
AVG_3MTHS_AMT_CR - AVG_3MTHS_AMT_DB: 0.8892988999738958
AVG_3MTHS_FREK_DB - AVG_12MTHS_FREK_DB: 0.8098386123760128
AVG_3MTHS_FREK_DB - AVG_9MTHS_FREK_DB: 0.8512629858588676
AVG_3MTHS_FREK_DB - AVG_6MTHS_FREK_DB: 0.9117070366960768
AVG_3MTHS_FREK_DB - FREK_DB: 0.9513719378350878
AVG_3MTHS_FREK_CR - AVG_12MTHS_FREK_CR: 0.9151082740236894
AVG_3MTHS_FREK_CR - AVG_9MTHS_FREK_CR: 0.9355141573721815
AVG_3MTHS_FREK_CR - AVG_6MTHS_FREK_CR: 0.9666599719929414
AVG_3MTHS_FREK_CR - FREK_CR: 0.9892064294816267
SALDO_AVG_CASA - AVG_12MTHS_CASA: 0.8731082022551883
SALDO_AVG_CASA - AVG_12MTHS_DPK: 0.8351773869474667
SALDO_AVG_CASA - AVG_9MTHS_CASA: 0.9702868548517564
SALDO_AVG_CASA - AVG_9MTHS_DPK: 0.8706462828524563
SALDO_AVG_CASA - AVG_6MTHS_CASA: 0.9709549565257941
SALDO_AVG_CASA - AVG_6MTHS_DPK: 0.8722242111957933
SALDO_AVG_CASA - AVG_3MTHS_CASA: 0.9793492236798634
SALDO_AVG_CASA - AVG_3MTHS_DPK: 0.9267795944632143
SALDO_AVG_CASA - SALDO_AVG_DPK: 0.9122309371644718
SALDO_AVG_DPK - AVG_12MTHS_DPK: 0.8860154482226915
SALDO_AVG_DPK - AVG_9MTHS_CASA: 0.8946835231154521
SALDO_AVG_DPK - AVG_9MTHS_DPK: 0.965944083473074
SALDO_AVG_DPK - AVG_6MTHS_CASA: 0.8867351422130509
SALDO_AVG_DPK - AVG_6MTHS_DPK: 0.9582641302351401
SALDO_AVG_DPK - AVG_3MTHS_CASA: 0.8858169340261174
SALDO_AVG_DPK - AVG_3MTHS_DPK: 0.9787557160458124
SALDO_AVG_DPK - SALDO_AVG_CASA: 0.9122309371644718
AVG_MUTASI_DB - AVG_MUTASI_CR: 0.8240333116989876
AVG_MUTASI_CR - AVG_MUTASI_DB: 0.8240333116989876
FREK_DB - AVG_6MTHS_FREK_DB: 0.8086264594992362
FREK_DB - AVG_3MTHS_FREK_DB: 0.9513719378350878
FREK_CR - AVG_12MTHS_FREK_CR: 0.8849641956401505
FREK_CR - AVG_9MTHS_FREK_CR: 0.9088126526948385
FREK_CR - AVG_6MTHS_FREK_CR: 0.9440903511651497
FREK_CR - AVG_3MTHS_FREK_CR: 0.9892064294816267
```

```
FLAG_RESTRU_COV - FLAG_RESTRU_COV_21: 0.9383867943568989
FLAG_DEFERRED_COV - FLAG_DEFERRED_COV_21: 0.9122249682345108
FLAG_RESTRU_COV_21 - FLAG_RESTRU_COV: 0.9383867943568989
FLAG_DEFERRED_COV_21 - FLAG_DEFERRED_COV: 0.9122249682345108
3to1 - slope: 0.923757376521067
slope - 3to1: 0.923757376521067
```

```
/tmp/ipykernel_430/1268544356.py:3: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  correlation_matrix = df.corr().abs()
```

[10]:

```
#create new columns. -1 is deprove, 0 stay the same, 1 improve
df["rest_change"] = df["FLAG_RESTRU_COV_21"] - df["FLAG_RESTRU_COV"]
df["def_change"] = df["FLAG_DEFERRED_COV_21"] - df["FLAG_DEFERRED_COV"]
df["restdef_change"] = df["rest_change"] + df["def_change"]
```

[11]:

```
#remove columns from correlation analysis
df_dropped = df[['KOLEK', 'HARI_TUNGGAKAN', 'total_flags','percent_used',␣
 ↪'slope', '12to3', '12to6DB', 'restdef_change']]
```

[12]:

```
df_dropped
```

[12]:

```
     KOLEK  HARI_TUNGGAKAN  total_flags  percent_used     slope     12to3  \
0        5             187           16      1.000000  0.049647  0.553371
1        4             159            6      1.000000  0.185455  0.518953
2        5             678           11      1.000000  0.061662  0.093380
3        5             584           12      1.000023  0.040082 -0.238241
4        5             457            9      1.000000  0.166746 -1.100974
..     …             …          …           …       …       …
776      5             401            6      1.000000  0.165955  0.713864
777      4             128            2      1.000000  0.101690 -0.151916
778      5             278            1      1.000000  0.059484 -0.352962
779      5             370            7      1.000233 -0.013076  0.441755
780      4             126           12      1.000000 -0.265374  0.996959

      12to6DB  restdef_change
0    0.308853               0
1    0.729586               0
2   -0.130110               0
3   -0.063072               0
4    0.307290               0
..        …             …
776 -0.291876               0
777  0.043406               0
778 -0.017117               0
779  0.459780               0
```

```
780 -0.862965               0

[781 rows x 8 columns]
```

[19]:
```
#convert string columns to integers
#df_con = df_dropped.copy()
#df_con['DEBTOR_CATEGORY'] = np.where(df_con['DEBTOR_CATEGORY'] == '03 SME', 1,0)
#df_con['CUST_TYPE_CD'] = np.where(df_con['CUST_TYPE_CD'] == 'O', 1,0)
#df_con = df_con.fillna(0)
```

[13]:
```
#nan_rows  = df_dropped[df_dropped.isna().any(axis=1)]
#nan_rows
```

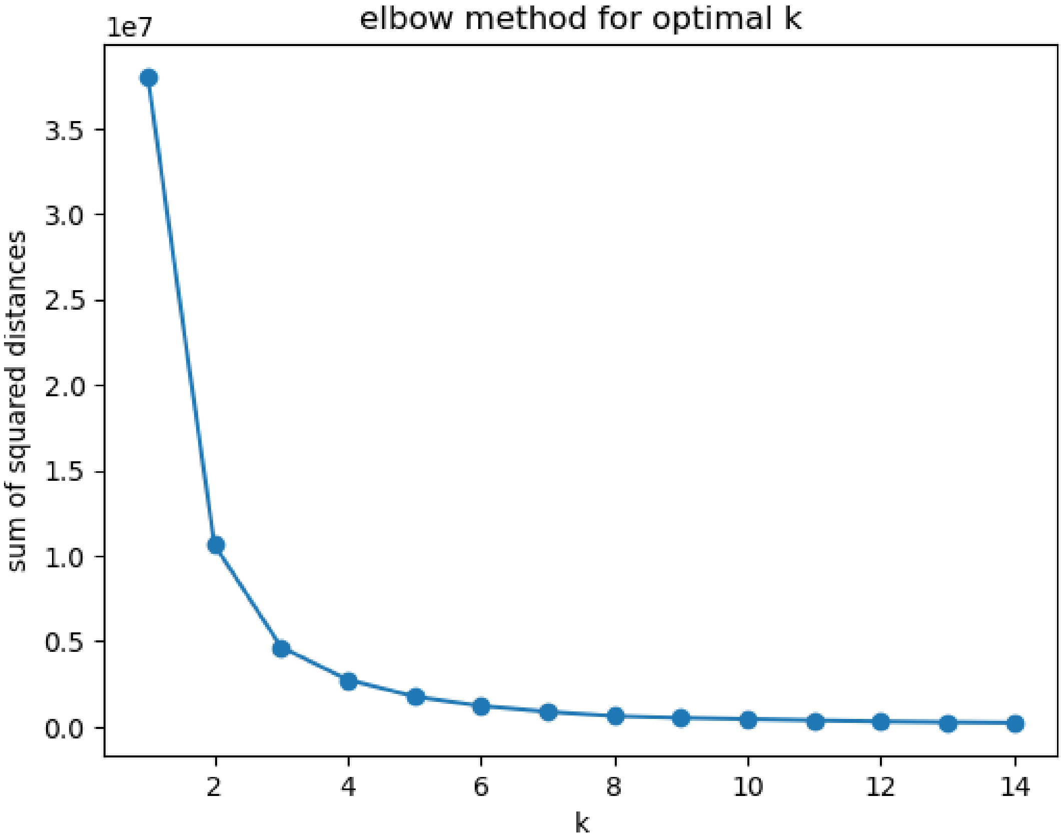
[14]:
```
df_dropped = df_dropped.drop(159)
df_dropped = df_dropped.drop(762)
df_dropped = df_dropped.drop(258)
df_dropped = df_dropped.drop(426)
df_dropped = df_dropped.drop(650)
df_dropped = df_dropped.drop(699)
df_dropped = df_dropped.drop(714)
df_dropped = df_dropped.drop(443)
df_dropped = df_dropped.drop(445)
df_dropped = df_dropped.drop(547)
df_dropped = df_dropped.drop(554)
```

K-MEANS CLUSTERING

[15]:
```
#k-means
from sklearn.cluster import KMeans

sum_of_squared_distances = []
K = range(1,15)

for k in K:
    km = KMeans(n_clusters=k, init='k-means++')
    km = km.fit(df_dropped)
    sum_of_squared_distances.append(km.inertia_)

#plot results
plt.plot(K, sum_of_squared_distances, marker='o')
plt.xlabel('k')
plt.ylabel('sum of squared distances')
plt.title('elbow method for optimal k')
plt.show()
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
```

```
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
```



[16]:

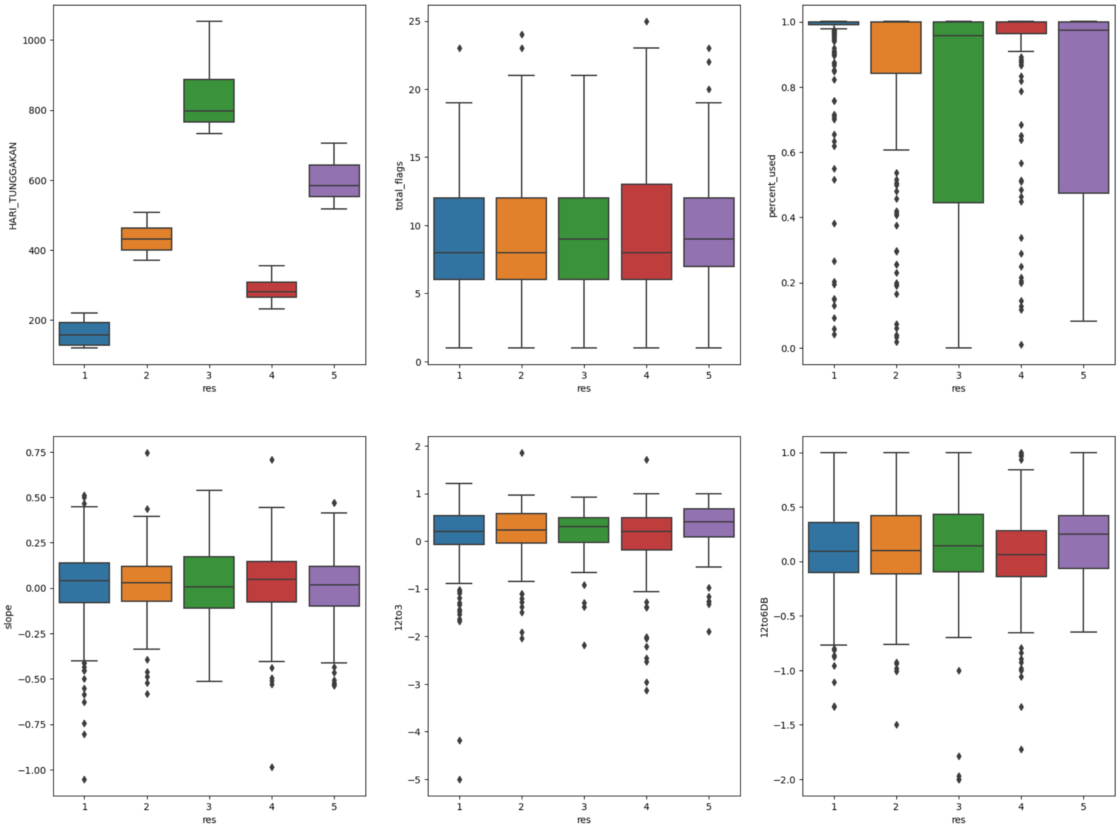
```python
km2 = KMeans(n_clusters=5, init='random', max_iter = 100, random_state = 0)
km2 = km2.fit(df_dropped)
label2 = km2.predict(df_dropped)

df_res2 = df_dropped.copy()
df_res2['res'] = [i+1 for i in label2]

check2 = df_res2.copy()
print(check2.groupby(["res"]).agg(len)["KOLEK"])

fig, axes = plt.subplots(2,3, figsize=(20,15))
#fig.subtitle('box plot for each cluster')

sns.boxplot(
    x = 'res',
    y = 'HARI_TUNGGAKAN',
    data = df_res2,
    ax = axes[0,0]
)

sns.boxplot(
    x = 'res',
    y = 'total_flags',
    data = df_res2,
    ax = axes[0,1]
)

sns.boxplot(
    x = 'res',
    y = 'percent_used',
    data = df_res2,
    ax = axes[0,2]
)

sns.boxplot(
    x = 'res',
    y = 'slope',
    data = df_res2,
    ax = axes[1,0]
)

sns.boxplot(
    x = 'res',
    y = '12to3',
    data = df_res2,
    ax = axes[1,1]
)
```

```
sns.boxplot(
    x = 'res',
    y = '12to6DB',
    data = df_res2,
    ax = axes[1,2]
)
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
```

```
res
1    268
2    147
3     79
4    142
5    134
Name: KOLEK, dtype: int64
```

[16]:
```
<Axes: xlabel='res', ylabel='12to6DB'>
```



- cluster 1:
  - 5th hari tunggakan
  - 5th total flags
  - 1st percent used
  - slope ?
  - 3rd 12to3
  - 4th 12to6DB
- cluster 2:
  - 3rd hari tunggakan
  - 3rd total flags
  - 3rd percent used
  - slope ?
  - 4th 12to3
  - 3rd 12to6DB
- cluster 3:
  - 1st hari tunggakan
  - 2nd total flags
  - 5th percent used
  - slope ?
  - 2nd 12to3
  - 2nd 12to6DB
- cluster 4:
  - 4th hari tunggakan
  - 1st total flags
  - 2nd percent used
  - slope ?
  - 5th 12to3
  - 5th 12to6DB
- cluster 5:
  - 2nd hari tunggakan
  - 4th total flags
  - 4th percent used
  - slope ?
  - 1st 12to3
  - 1st 12to6DB

```
[17]: df_res2.groupby(['res']).agg('mean').reset_index()
```

```
[17]:    res     KOLEK  HARI_TUNGGAKAN  total_flags  percent_used     slope  \
      0    1  4.533582      167.313433     8.917910      0.944938  0.020513
      1    2  5.000000      432.482993     9.238095      0.852411  0.026835
      2    3  5.000000      835.924051     8.708861      0.733289  0.003174
      3    4  5.000000      289.556338     8.711268      0.897492  0.023079
      4    5  5.000000      594.067164     9.514925      0.746047 -0.002813
      12to3   12to6DB  restdef_change
0  0.139745  0.116008       -0.052239
1  0.168448  0.103781        0.068027
2  0.180601  0.079909        0.000000
3  0.022212  0.055786        0.063380
4  0.311223  0.219895        0.074627
```

```
[ ]:
```