PA_4_Exploratory

October 26, 2023

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import random as rn
from scipy import stats

from sklearn.metrics import accuracy_score, precision_score, recall_score
```

[2]:
```
RANDOM_SEED = 42
rn.seed(RANDOM_SEED)
```

0.0.1 Load Data

1. Has some missing values, especially in column 'MIS_Status'

[3]:
```
df = pd.concat( pd.read_csv('SBAnational.csv', chunksize = 1000) )
```

[4]:
```
df.columns
```

[4]:
```
Index(['LoanNr_ChkDgt', 'Name', 'City', 'State', 'Zip', 'Bank', 'BankState',
       'NAICS', 'ApprovalDate', 'ApprovalFY', 'Term', 'NoEmp', 'NewExist',
       'CreateJob', 'RetainedJob', 'FranchiseCode', 'UrbanRural', 'RevLineCr',
       'LowDoc', 'ChgOffDate', 'DisbursementDate', 'DisbursementGross',
       'BalanceGross', 'MIS_Status', 'ChgOffPrinGr', 'GrAppv', 'SBA_Appv'],
      dtype='object')
```

[5]:
```
df.isna().sum()
```

[5]:
```
LoanNr_ChkDgt            0
Name                    14
City                    30
State                   14
Zip                      0
Bank                  1559
BankState             1566
NAICS                    0
ApprovalDate             0
```

```
ApprovalFY                0
Term                      0
NoEmp                     0
NewExist                136
CreateJob                 0
RetainedJob               0
FranchiseCode             0
UrbanRural                0
RevLineCr              4528
LowDoc                 2582
ChgOffDate           736465
DisbursementDate       2368
DisbursementGross         0
BalanceGross              0
MIS_Status             1997
ChgOffPrinGr              0
GrAppv                    0
SBA_Appv                  0
dtype: int64
```

[6]:
```
df.describe().transpose()
```

[6]:

	count	mean	std	min	\
LoanNr_ChkDgt	899164.0	4.772612e+09	2.538175e+09	1.000014e+09	
Zip	899164.0	5.380439e+04	3.118416e+04	0.000000e+00	
NAICS	899164.0	3.986610e+05	2.633183e+05	0.000000e+00	
Term	899164.0	1.107731e+02	7.885731e+01	0.000000e+00	
NoEmp	899164.0	1.141135e+01	7.410820e+01	0.000000e+00	
NewExist	899028.0	1.280404e+00	4.517500e-01	0.000000e+00	
CreateJob	899164.0	8.430376e+00	2.366882e+02	0.000000e+00	
RetainedJob	899164.0	1.079726e+01	2.371206e+02	0.000000e+00	
FranchiseCode	899164.0	2.753726e+03	1.275802e+04	0.000000e+00	
UrbanRural	899164.0	7.577483e-01	6.464360e-01	0.000000e+00	

	25%	50%	75%	max
LoanNr_ChkDgt	2.589758e+09	4.361439e+09	6.904627e+09	9.996003e+09
Zip	2.758700e+04	5.541000e+04	8.370400e+04	9.999900e+04
NAICS	2.352100e+05	4.453100e+05	5.617300e+05	9.281200e+05
Term	6.000000e+01	8.400000e+01	1.200000e+02	5.690000e+02
NoEmp	2.000000e+00	4.000000e+00	1.000000e+01	9.999000e+03
NewExist	1.000000e+00	1.000000e+00	2.000000e+00	2.000000e+00
CreateJob	0.000000e+00	0.000000e+00	1.000000e+00	8.800000e+03
RetainedJob	0.000000e+00	1.000000e+00	4.000000e+00	9.500000e+03
FranchiseCode	1.000000e+00	1.000000e+00	1.000000e+00	9.999900e+04
UrbanRural	0.000000e+00	1.000000e+00	1.000000e+00	2.000000e+00

[7]:
```
for col in df.columns:
    print(f'{col} -> {df[col].unique()}')
    print('='*50)
```

```
LoanNr_ChkDgt -> [1000014003 1000024006 1000034009 … 9995613003 9995973006
9996003010]
==================================================
Name -> ['ABC HOBBYCRAFT' 'LANDMARK BAR & GRILLE (THE)' 'WHITLOCK DDS, TODD M.'
 … 'RADCO MANUFACTURING CO.,INC.' 'MARUTAMA HAWAII, INC.'
 'PACIFIC TRADEWINDS FAN & LIGHT']
==================================================
City -> ['EVANSVILLE' 'NEW PARIS' 'BLOOMINGTON' … 'MURFRECSBORO' 'E WENALCHEE'
 'SO. OZONE PARK']
==================================================
State -> ['IN' 'OK' 'FL' 'CT' 'NJ' 'NC' 'IL' 'RI' 'TX' 'VA' 'TN' 'AR' 'MN' 'MO'
 'MA' 'CA' 'SC' 'LA' 'IA' 'OH' 'KY' 'MS' 'NY' 'MD' 'PA' 'OR' 'ME' 'KS'
 'MI' 'AK' 'WA' 'CO' 'MT' 'WY' 'UT' 'NH' 'WV' 'ID' 'AZ' 'NV' 'WI' 'NM'
 'GA' 'ND' 'VT' 'AL' 'NE' 'SD' 'HI' 'DE' 'DC' nan]
==================================================
Zip -> [47711 46526 47401 … 70036 66549 26134]
==================================================
Bank -> ['FIFTH THIRD BANK' '1ST SOURCE BANK' 'GRANT COUNTY STATE BANK' …
 'FIRST ILLINOIS BANK' 'COLORADO COMMUNITY FIRST STATE' 'DEPCO']
==================================================
BankState -> ['OH' 'IN' 'OK' 'FL' 'DE' 'SD' 'AL' 'CT' 'GA' 'OR' 'MN' 'RI' 'NC'
'TX'
 'MD' 'NY' 'TN' 'SC' 'MS' 'MA' 'LA' 'IA' 'VA' 'CA' 'IL' 'KY' 'PA' 'MO'
 'WA' 'MI' 'UT' 'KS' 'WV' 'WI' 'AZ' 'NJ' 'CO' 'ME' 'NH' 'AR' 'ND' 'MT'
 'ID' nan 'WY' 'NM' 'DC' 'NV' 'NE' 'PR' 'HI' 'VT' 'AK' 'GU' 'AN' 'EN' 'VI']
==================================================
NAICS -> [451120 722410 621210 … 315280 922140 221121]
==================================================
ApprovalDate -> ['28-Feb-97' '2-Jun-80' '7-Feb-06' … '24-Feb-97' '25-Feb-97'
'27-Feb-97']
==================================================
ApprovalFY -> [1997 1980 2006 1998 1999 2000 2001 1972 2003 2004 1978 1979 1981
2005
 1962 1982 1965 1966 1983 1973 1984 2007 1985 1986 1987 2008 1988 2009
 1989 1991 1990 1974 2010 1967 2011 1992 1993 2002 2012 2013 1994 2014
 1975 1977 1976 1968 '1994' '2004' '1976' '1976A' '1977' '1975' 1969 1995
 1970 1996 '1995' '1970' 1971 '2005' '1979' '1978' '1981' '1982']
==================================================
Term -> [ 84  60 180 240 120  45 297 162  12 300  87 114 144 126  83 102  80 137
  42  96 167   7  36  37  26 264  72  24   5  54  66 161  71   4  93 288
 108  10  13  90  19  16   3  27 149  41 246  18  57 104  82 298  14  61
 127  58  44  32  85  48  31 112  38  73  47  11 134  15  79  53  39   6
 255  55 133  95  35  59  62  68 123  46  70 138  40  52  25  65  91   1
  74  49 103  77  86  63  56  22   0  97  23  17  69  21  43  89 276  92
```

```
 183   2 132  34 131   9  78  99 129 216   8  29 289  30 119 228 168 208
  81 147 125  94  51 211  64 111 266  75 306  28 232 117 118 309 303  98
 191 116  76 113 292  88 166 244 176 258 203 231 142  33 157 165  50 210
 294 301 106  20 318 229 204 269 241 178 115 174 192  67 100 141 282 122
 156 153 268 249 238 233 105 263 124 279 140 186 107 190 308 128 243 302
 299 280 223 311 222 202 257 130 101 121 278 272 319 283 221 250 290 199
 252 187 310 304 136 261 196 181 175 195 177 139 110 242 270 277 184 150
 207 358 213 273 357 248 275 164 239 206 215 170 254 217 172 158 218 189
 256 179 262 193 146 155 135 185 307 200 109 265 349 209 236 251 145 152
 169 245 230 285 201 293 171 227 188 225 247 163 143 235 160 148 253 159
 295 198 271 234 237 336 220 291 287 154 219 197 312 284 281 151 267 274
 182 212 305 205 226 194 259 214 173 224 385 313 340 461 296 343 348 286
 260 334 320 315 360 421 342 351 372 314 354 435 316 480 329 387 325 321
 317 352 359 389 333 330 417 404 370 324 398 419 425 414 365 345 481 350
 335 369 341 355 362 322 339 375 323 347 327 505 326 418 328 363 413 361
 356 396 438 382 364 367 374 442 353 527 569 338 366 386 368 428 346 388
 430 443 381 409 445 384 391 511 412 449 403 434 402 423 440 429]
==================================================
NoEmp -> [   4    2    7   14   19   45    1    3   24    5   16   12    6   90
   18    9   20   10    8   50   17   32   31   60   22   40   72   55
   30   25   46   15  214   28   23   11   57   13  112   26   80   42
   65   21   97  100  200  126   48   33   58   38   37   35    0   75
   36   70   66   27 2000   56   34   93   85  150   29   41  290   67
   44   47  119   39  155   54   49   82   95  300   51  120  265  133
   86  160   68   61  220   43   71   98  350   78  233  263   62 7941
   63  210  125  107  450  165  130 9992   77   64  424  257   52  600
  190  142   99   59   73  135   74  109  250   69  500  140  116  260
   96  339   87  110  161   88  105   53  400 2725  605  103   91   81
   94  147  144  175  136  182  156  118  375  345  121  145  180  101
   84   89   76  550  216  108  138  171  117  249  279  208  170   79
 9999  115  104  900   83  153  106  191  270  102  295 3000  230  195
  127  313  205  141  128  199  124  162  305  185  152  129  275  576
  132  196  640  178  750  720  414  315  143  299  217  179  320 1145
 4000  167  114  203  134 8000  111 5511 5921   92  240  280  148  340
  285 1300 1800  680  258  231  173 1980  510 5812  401 7212  192 5084
  480  186  131  146  168 1050  189  343 7216  312 8041  394  307 7389
  174  828  485  266 2100 1000 5947  368 1600 5000 1500  298 6000  221
 4953  360  355  137  325  123  122  344  163  386  530  237  330  188
  387 1003  761  421  113  308  158  310  187  238  235  222 7111 2112
  254 1200  225  228 2400  317  425 8018  430  197  157  176  198  256
 3200  183  277  362  202  289  139  520  154 2151  177  316  224  149
  151  215  395  426  262 6501  169  282 4100  383  172  688  287 1250
  390  318 4847 7231 5149 3900  967 3500  314  602  476  206 1461  274
  408  441  246  336  570  227 1940  735  523  204 1233 3170 1711  255
  351 1451  207  365  454 1550  823  544 1150  294 9000  226 7000  463
  211  370 2900 4005 1900  241  288  194  213  278  332  484  281  322
  650  181  521  218  800  420 3400 1400  700  242 2200 2500  625  495
  358  243  466  354  223  184  479  269  435  342  385  193  232  827
```

```
 2401  273  713  253  261  296 2501 1629 1700  429 2010  455  660  164
 1235  252  376  380  464  245 1100  608  209  247 5555  329  604  456
  166 1280 3089  985 1020  505 1502  234 5200  284  609  259  475  324
 5680 1981  323  251  740  575  396 1030  229 2610  515  328  442  433
 2232  341  306 3732  346  447  850  427  407  782  293  236  356 4685
 7241  363 1005  369  458  267 7999 2020  445 2121 1125 1010 4658  712
  212  271  377 1718 1515  560  404  302  276  248 1015  268 3737  319
 2120  304  512  585  292  808  244 9090 3030  606  840  460  301 2300
 3600  159  525  353 7991 5211 4012 1112 1440  413  410  488 4501 4800
 3100 3334  538 1603 1706 2520  283 1520 2202  357  201 1012  499  423
  635 1073  465 2510 1644 1101  403 4300  382  498  448 3009  685 1340
 2700  367  535  760 1524  309 7007  384  327 1960  540 5013  780  348
  717 8500 7538  405 2005 1382  858 9945 1542 1920 3713]
==================================================
NewExist -> [ 2.  1.  0. nan]
==================================================
CreateJob -> [   0    7   30    5    4    1   20   10    3    8   16   15    6
11
    2   40   55   25   12   21   50    9   13   47   18   17   14   29
   23   35   43   75   22   45   27   65   19   58   48   72   38   28
   24  150  200   82   68   41   80   70   33   97   32   26   34   36
   31  100   56   60   90   77   99   39   44   51  120   85   69   95
   42  160   37   57  600   49 1000   53   54   46   59  163  450  456
 3000  452  451  198   79  454   62  136   64   52  126  180   74  303
   63  386   78   98  455   76  152  221  110   84  153  127 2020  225
  453  125  458  457  174  104   89  320  154  300  102  149 8800  800
  130  235 5199  250  137  500  121  105   96  360  255  140  122  175
 1200   66  112 3500  118  220  115   73   93  151  195   67  138  400
   61  124   91 1711  131  184   83  409 1618 1150   88 1530  157  145
  166  135  210  226  183 3100  252  116   71  129  223   81  569  139
  144 1011  179  214  146  171  141  350   92  101  119  280  123  205
 1229  128  103  189  114  108  158  167   87  186   86  134 1100  750
  206  375  109  433 2140  177  264  168  240 5621  170  169  165  222
  106  148  363 1118  310  164 5085  143  480  256  365  155  190  397
 1027  270   94 2515  162  182 1016  860]
==================================================
RetainedJob -> [   0    7   23    4    6    1    9   20    2    5   19    8    3
10
   24   12   15   11   25   44   17   14   65   28   38   16   42   26
   18   13   50   93   40   37   60   21   30   31   34   35  150   22
   73   41   45  100  180   58   75  165   36  130   29   27  125   99
   46   32  257   43   47   80   70   54   62   33   39  400   55   95
   48  120   71   63   81   52   94   78  160  109   86   77  155   85
   90   64 3225   61   69   66  210  107   97   51   83  112   53   72
   76   87   68  118  138   67   57   56  117  171  229  115  275  153
  300  105  140  135   59   79  200  295  205  206  128  186  137  250
   89   49  131   92  404  110  320  139   82  108   88  104  114  134
  230  102  103   96   98   84  101  220  233   74  267   91 9500  355
```

```
  123  175  550  500  450  170  195  116  305  147  610  187  235  157
  124  127  106  254 4441  277  225  207  111  312  317  173  350  216
  143  430  197  176  145  126  133  256 2200  362  202  148  316 8800
  215  146  185  154  212  141  163  184 5000 3200  132  194  113  161
  172  330  366  190 1300  390 4000  476 3900  967  268  136  602  121
  240  122  162  523  204  159 1711  119  251  152  417  291  544  129
  142  231  189  203  360  213  278  280  484  260  177  281  675  226
  263  700  247  600  245  750  151  270  375  191  182  223 7250  214
  169  342  221  217  232  815  287  285  188 1000 1700  428  660  156
 1500  318  265  167  236  370  310  609  475  322  208  515  259  328
  497  356  255  158  192  166  219  363  274  144  262  315  178  420
  286  585  325  201  710  196  384  237  940  302  371  394 1600 3860
  244  393  410  472  720  168  252  290  297  548  485  183  800  149
  387  298  480  266  164  403  369  498  448  685  535  292  327  911
 3100  540  304 1111  243  199  900  198]
==================================================
FranchiseCode -> [    1     0 15100 …  2899 18701 15930]
==================================================
UrbanRural -> [0 1 2]
==================================================
RevLineCr -> ['N' '0' 'Y' 'T' nan '`' ',' '1' 'C' '3' '2' 'R' '7' 'A' '5' '.'
'4' '-'
 'Q']
==================================================
LowDoc -> ['Y' 'N' 'C' '1' nan 'S' 'R' 'A' '0']
==================================================
ChgOffDate -> [nan '24-Jun-91' '18-Apr-02' … '25-Dec-02' '11-Jul-00'
'9-Oct-98']
==================================================
DisbursementDate -> ['28-Feb-99' '31-May-97' '31-Dec-97' … '21-Jun-97'
'8-May-02'
 '25-Oct-97']
==================================================
DisbursementGross -> ['$60,000.00 ' '$40,000.00 ' '$287,000.00 ' …
'$377,446.00 '
 '$123,770.00 ' '$1,086,300.00 ']
==================================================
BalanceGross -> ['$0.00 ' '$12,750.00 ' '$827,875.00 ' '$25,000.00 ' '$37,100.00
'
 '$43,127.00 ' '$84,617.00 ' '$1,760.00 ' '$115,820.00 ' '$996,262.00 '
 '$395,476.00 ' '$41,509.00 ' '$600.00 ' '$9,111.00 ' '$96,908.00 ']
==================================================
MIS_Status -> ['P I F' 'CHGOFF' nan]
==================================================
ChgOffPrinGr -> ['$0.00 ' '$208,959.00 ' '$14,084.00 ' … '$109,860.00 '
'$140,812.00 '
 '$124,847.00 ']
==================================================
```

```
GrAppv -> ['$60,000.00 ' '$40,000.00 ' '$287,000.00 ' … '$12,480.00 '
 '$62,425.00 ' '$1,086,300.00 ']
==================================================
SBA_Appv -> ['$48,000.00 ' '$32,000.00 ' '$215,250.00 ' … '$9,984.00 '
'$34,246.00 '
 '$715,674.00 ']
==================================================
```

[8]:
```
df.dropna(subset=['MIS_Status'], inplace=True)

def remove_spaces(text):
    return text.replace(' ', '')

df['MIS_Status'] = df['MIS_Status'].apply(remove_spaces)

df['MIS_Status'].unique()
```

[8]:
```
array(['PIF', 'CHGOFF'], dtype=object)
```

0.0.2 Convert dates into Pandas date series

[9]:
```
df['OutPut'] = df['MIS_Status'].map({'PIF': 1, 'CHGOFF': 0})
```

[10]:
```
text_col = ['Name','City','State','Bank','BankState','NAICS']
date_col = ['ApprovalDate','DisbursementDate']
numeric_col =␣
 ↪['Term','NoEmp','CreateJob','RetainedJob','DisbursementGross','BalanceGross','ChgOffPrinGr'
```

[11]:
```
for col in date_col:
    df[col] = pd.to_datetime(df[col], format='%d-%b-%y')
```

[12]:
```
for col in numeric_col:
    if df[col].dtype == 'object':
        df[col] = df[col].str.replace('[^\d.]', '', regex=True).astype(float)
```

[13]:
```
df[numeric_col].describe().transpose()
```

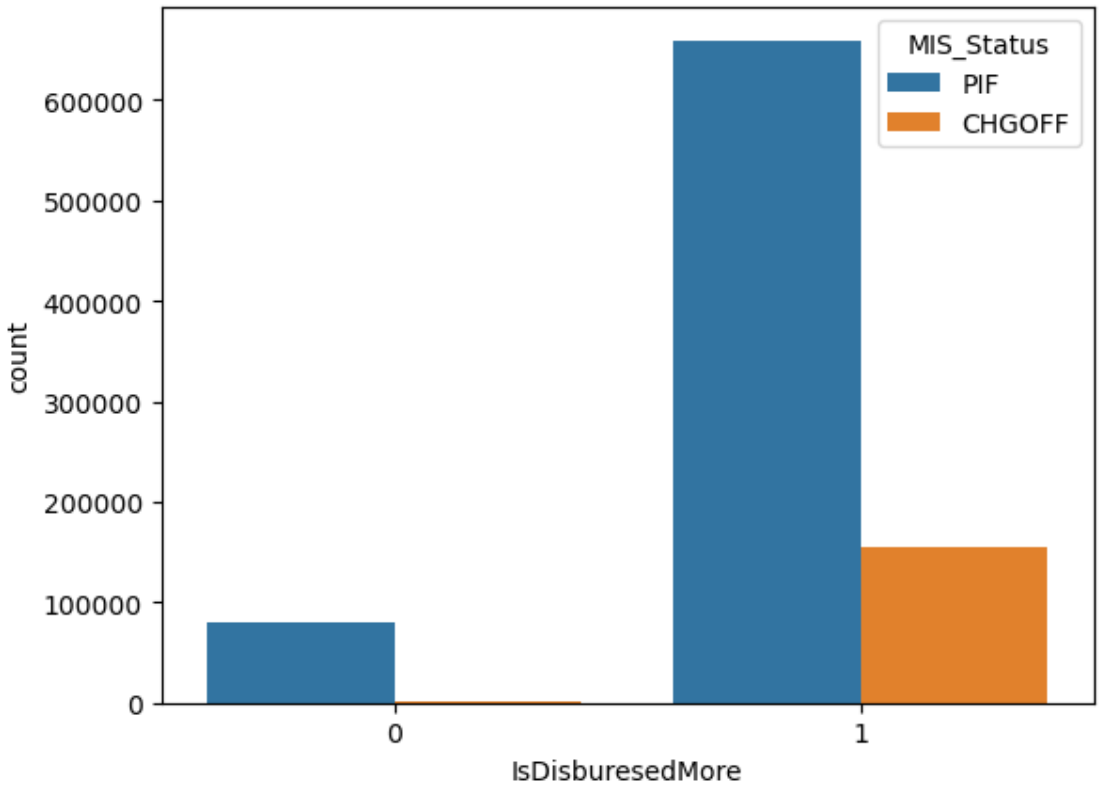
[13]:

	count	mean	std	min	25%	\
Term	897167.0	110.848592	78.900862	0.0	60.0	
NoEmp	897167.0	11.412562	73.793775	0.0	2.0	
CreateJob	897167.0	8.444305	236.950249	0.0	0.0	
RetainedJob	897167.0	10.807308	237.382398	0.0	0.0	
DisbursementGross	897167.0	201598.034681	287806.620570	4000.0	42492.0	
BalanceGross	897167.0	2.996003	1443.766066	0.0	0.0	
ChgOffPrinGr	897167.0	13527.211002	65209.860188	0.0	0.0	
GrAppv	897167.0	193059.516894	283433.114425	1000.0	35000.0	
SBA_Appv	897167.0	149780.698635	228559.979775	500.0	21250.0	

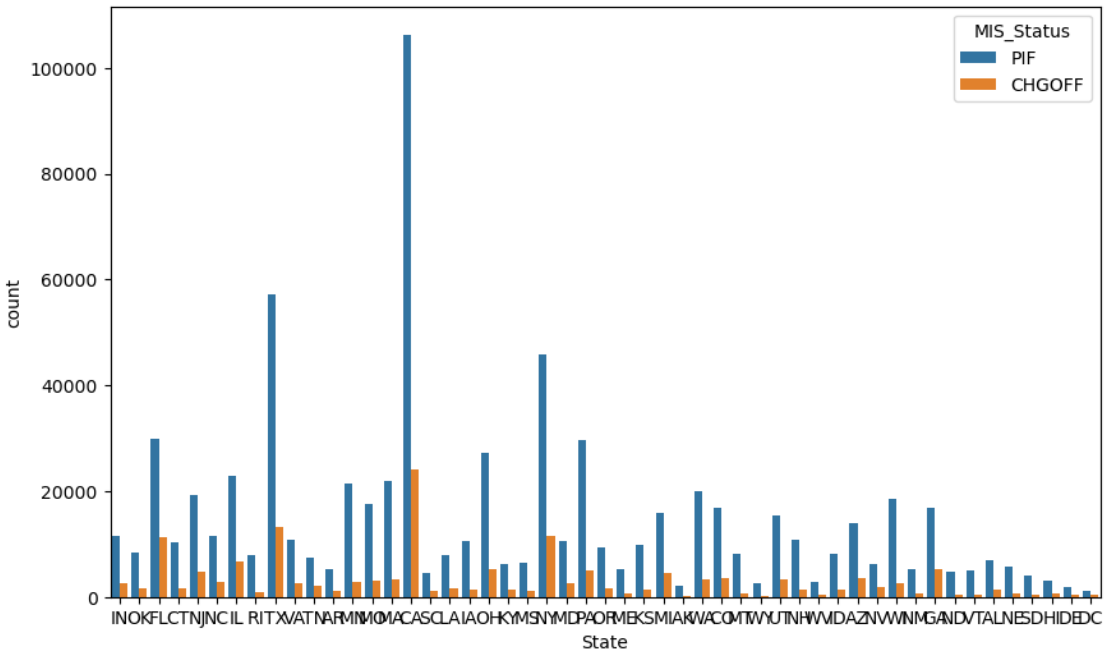
	50%	75%	max
Term	84.0	120.0	569.0
NoEmp	4.0	10.0	9999.0
CreateJob	0.0	1.0	8800.0
RetainedJob	1.0	4.0	9500.0
DisbursementGross	100000.0	239000.0	11446325.0
BalanceGross	0.0	0.0	996262.0
ChgOffPrinGr	0.0	0.0	3512596.0
GrAppv	90000.0	225000.0	5472000.0
SBA_Appv	62050.0	175000.0	5472000.0

[14]:
```
df['IsDisburesedMore'] = df['DisbursementGross'] > df['SBA_Appv']
df['IsDisburesedMore'] = df['IsDisburesedMore'].map({True: 1, False: 0})
df_test = df.sample(frac = 0.25, random_state = RANDOM_SEED)
```

[15]:
```
sns.countplot(data=df, x='IsDisburesedMore', hue='MIS_Status')
plt.show()
```

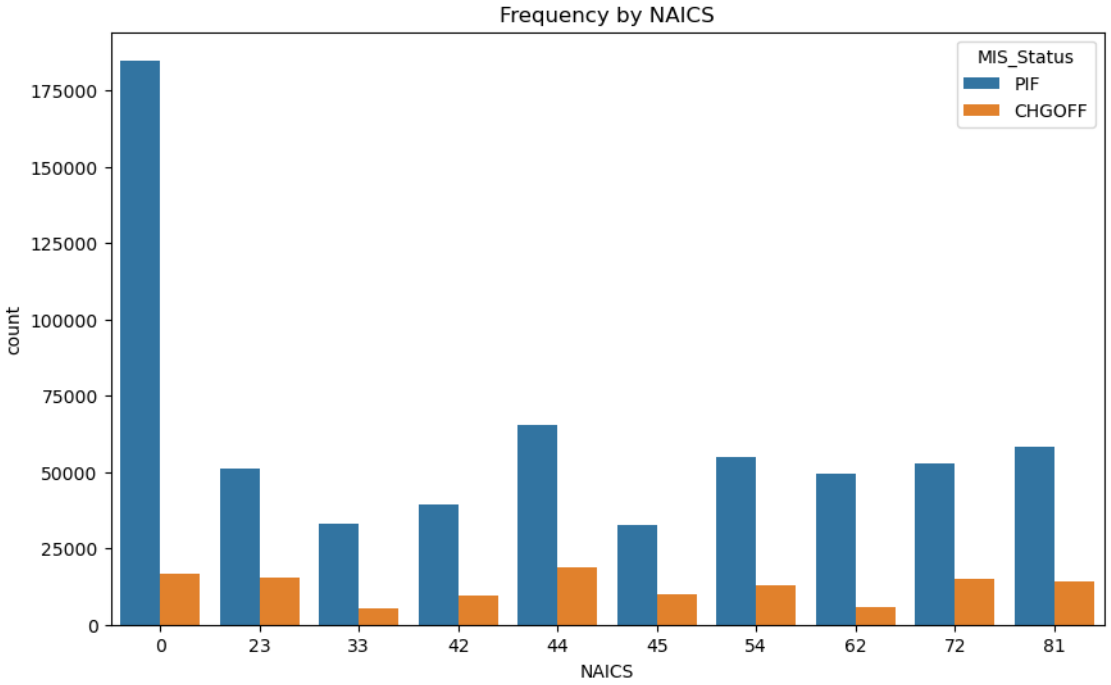


[16]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='State', hue='MIS_Status')
plt.show()
```

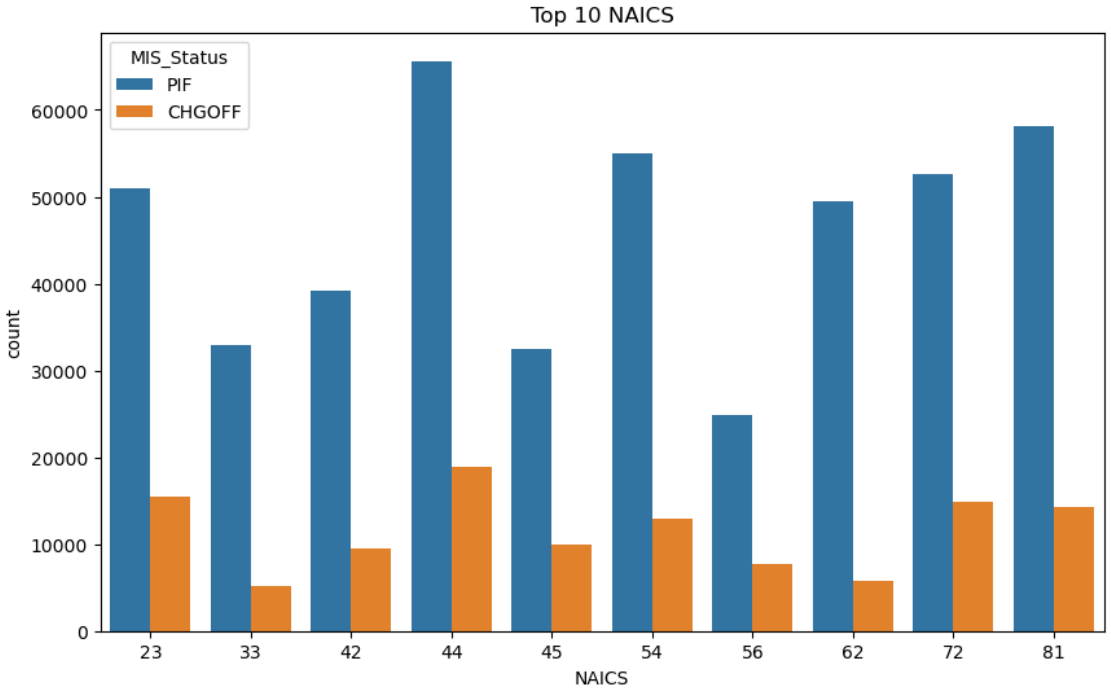


[17]:
```
df['NAICS'] = df['NAICS'] // 10000
```

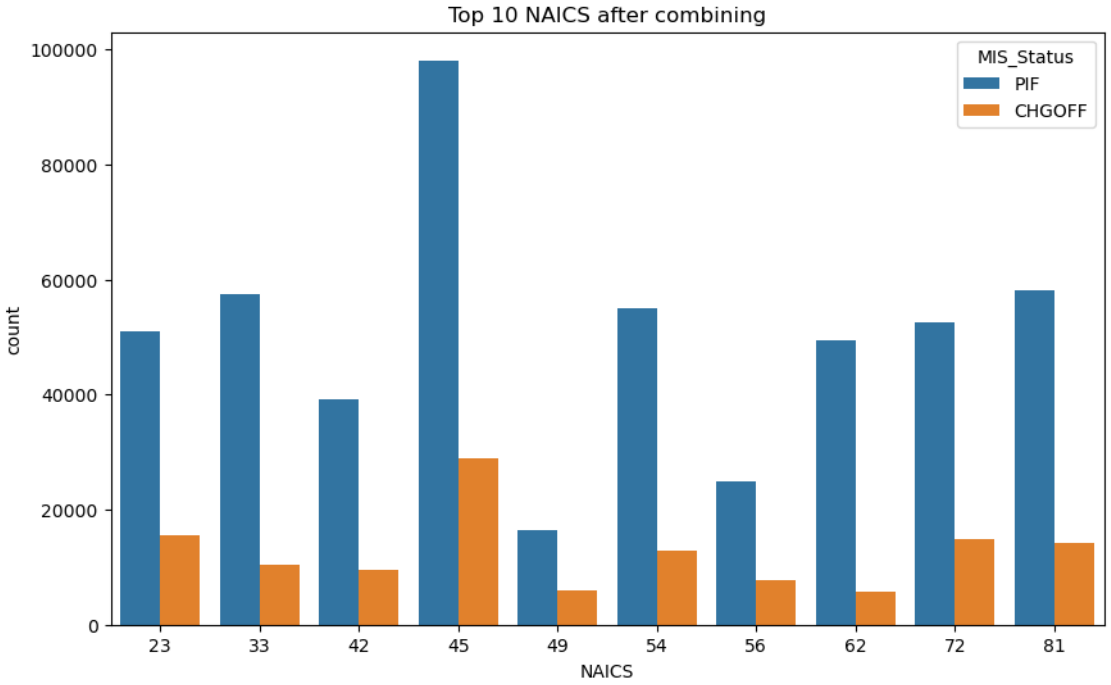
[18]:
```
naics_counts = df['NAICS'].value_counts()
top_10_naics = naics_counts.head(10)

plt.figure(figsize=(10, 6))
sns.countplot(data=df[df['NAICS'].isin(top_10_naics.index)], x='NAICS',␣
↪hue='MIS_Status')
plt.title('Frequency by NAICS')
plt.show()
```



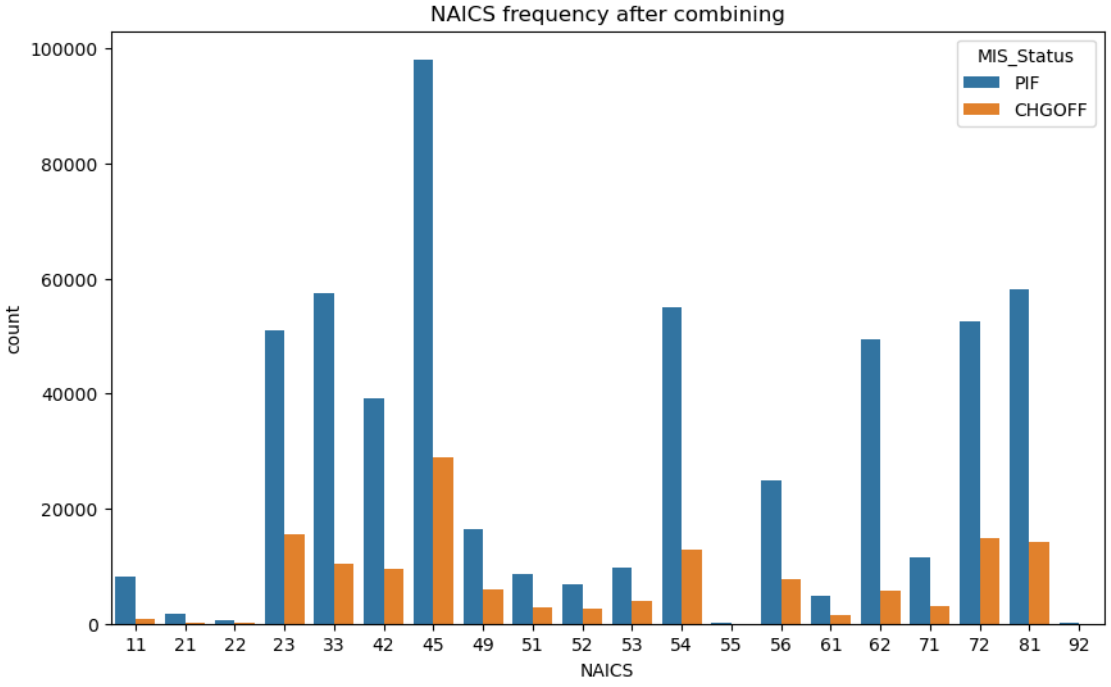
[19]:
```
# After dropping 0 values
df = df[df['NAICS']>0]
naics_counts = df['NAICS'].value_counts()
top_10_naics = naics_counts.head(10)

plt.figure(figsize=(10, 6))
sns.countplot(data=df[df['NAICS'].isin(top_10_naics.index)], x='NAICS',␣
 ↪hue='MIS_Status')
plt.title('Top 10 NAICS')
plt.show()
```



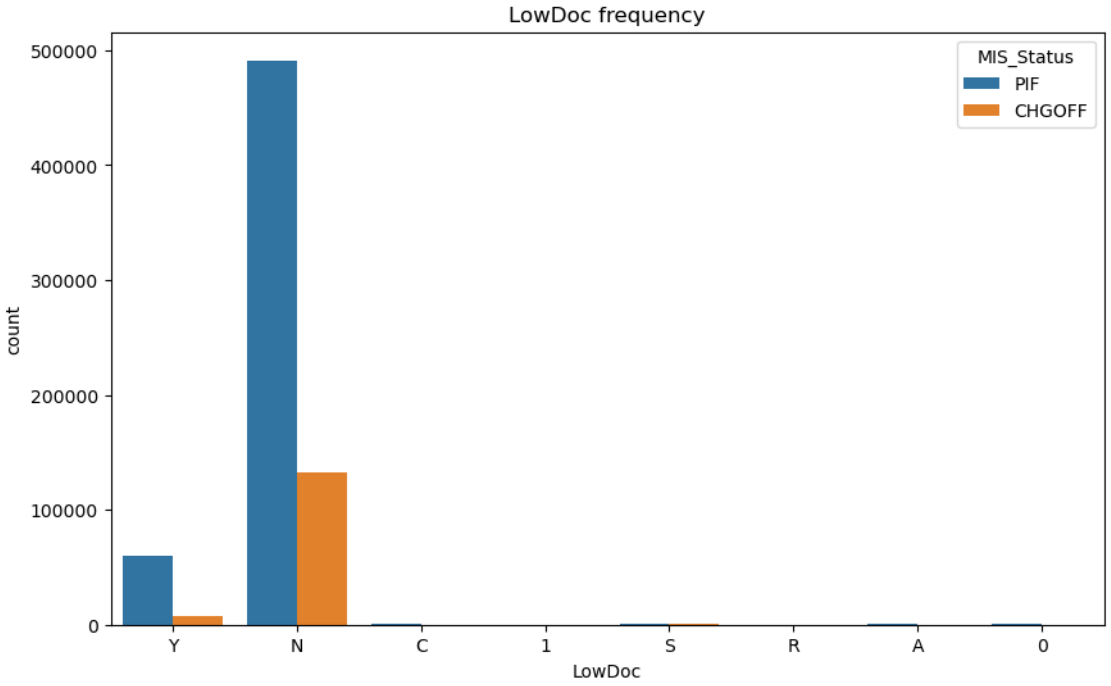
[20]:
```
# After combining the NAICS
df.loc[df['NAICS'].between(31, 33), 'NAICS'] = 33
df.loc[df['NAICS'].between(44, 45), 'NAICS'] = 45
df.loc[df['NAICS'].between(48, 49), 'NAICS'] = 49
naics_counts = df['NAICS'].value_counts()
top_10_naics = naics_counts.head(10)

plt.figure(figsize=(10, 6))
sns.countplot(data=df[df['NAICS'].isin(top_10_naics.index)], x='NAICS',␣
 ↪hue='MIS_Status')
plt.title('Top 10 NAICS after combining')
plt.show()
```



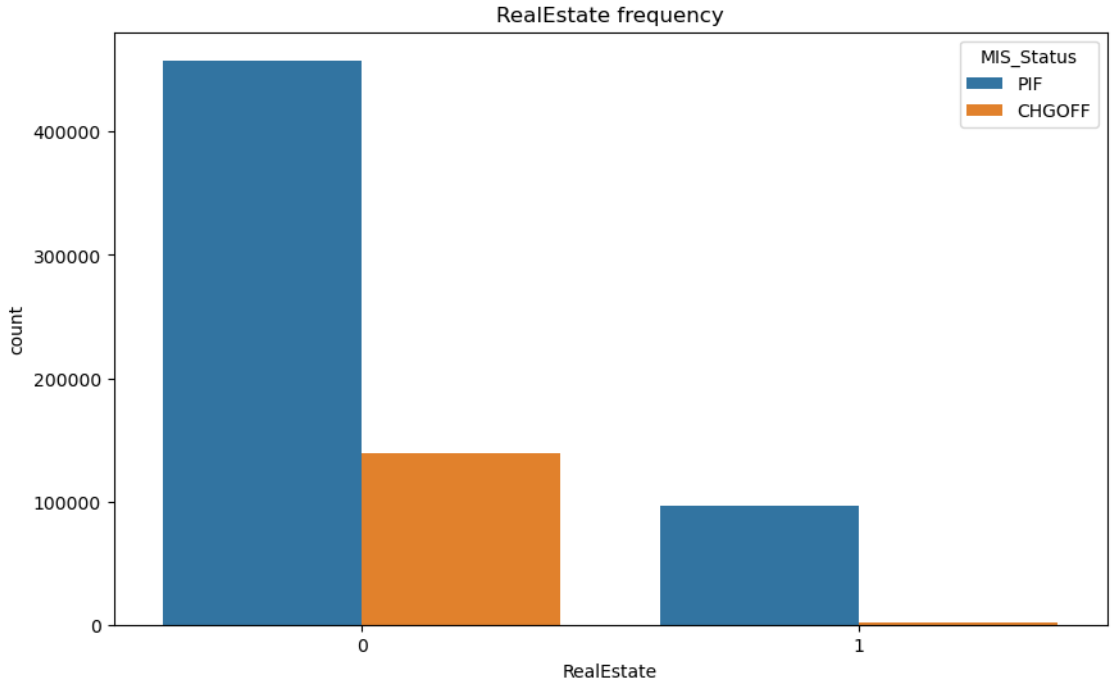
[21]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='NAICS', hue='MIS_Status')
plt.title('NAICS frequency after combining')
plt.show()
```



[22]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='LowDoc', hue='MIS_Status')
plt.title('LowDoc frequency')
plt.show()
```

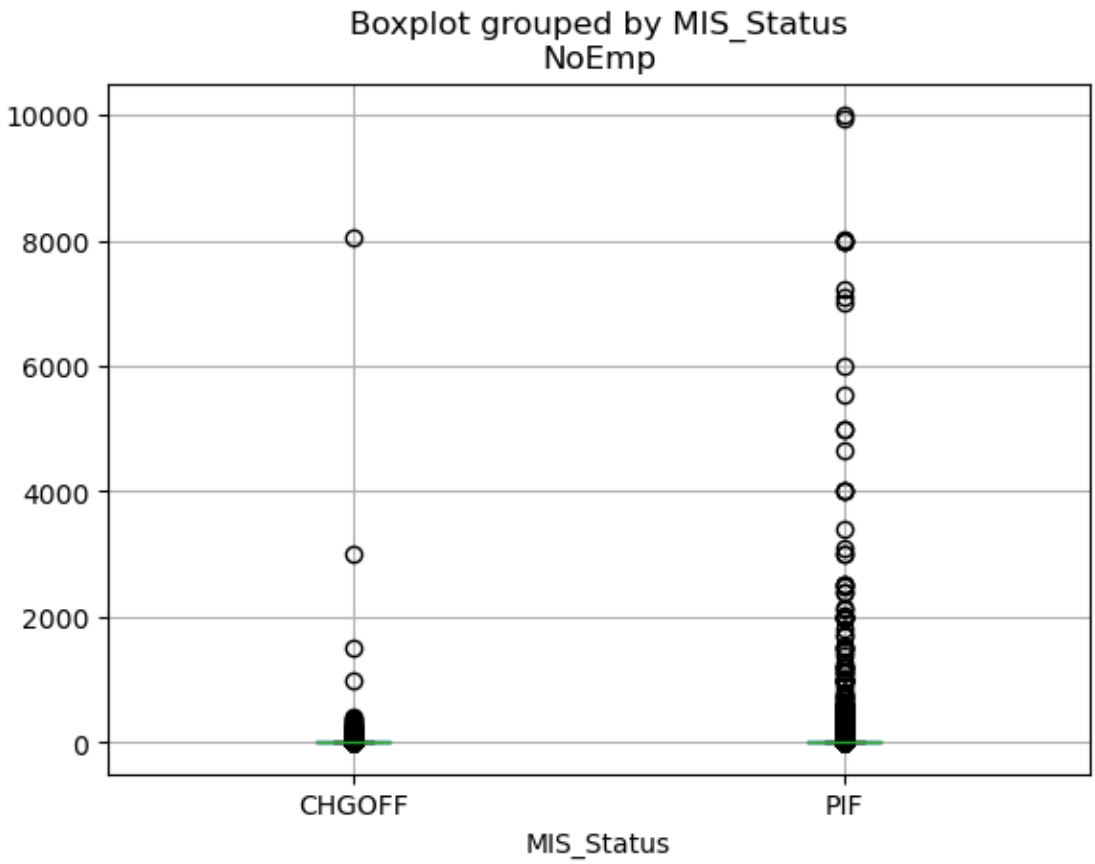


[23]:
```
df['RealEstate'] = df['Term'] >= 240
df['RealEstate'] = df['RealEstate'].map({True: 1, False: 0})
df['Portion'] = df['SBA_Appv']/df['GrAppv']

plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='RealEstate', hue='MIS_Status')
plt.title('RealEstate frequency')
plt.show()
```



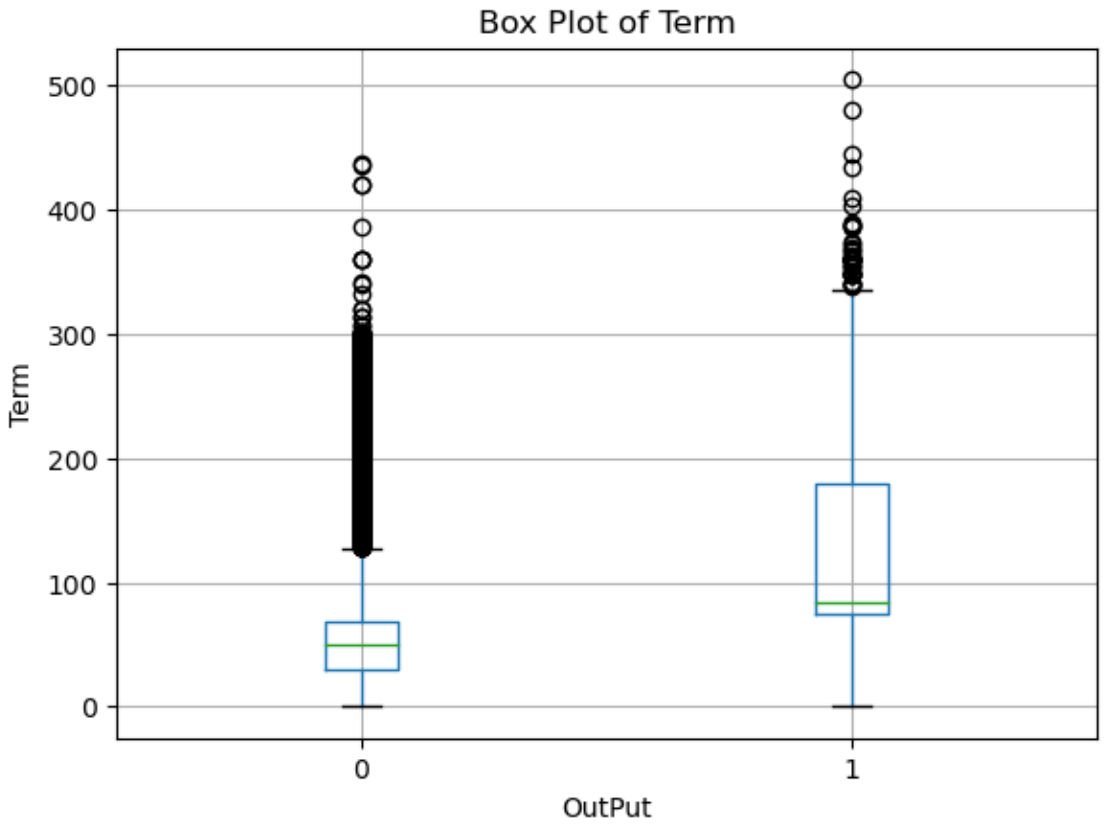
[24]:
```
df_test.boxplot(column='NoEmp', by='MIS_Status')
```

[24]:
```
<Axes: title={'center': 'NoEmp'}, xlabel='MIS_Status'>
```

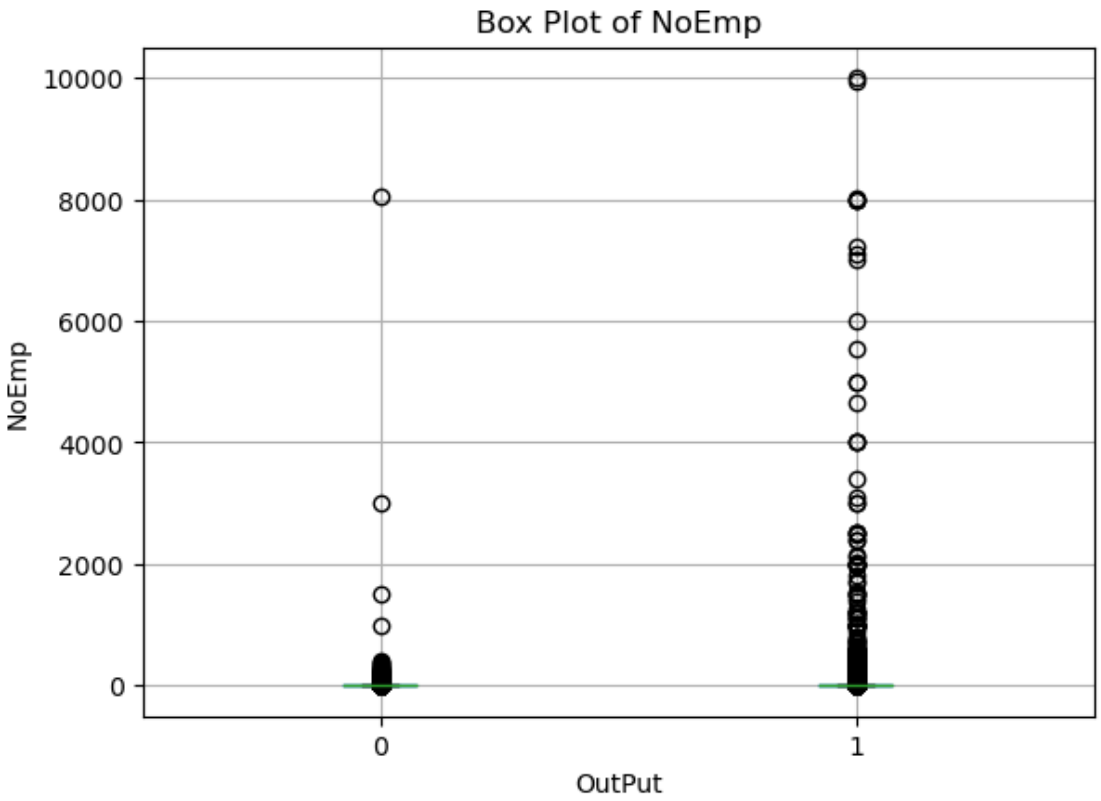


[25]:
```
for col in numeric_col:
    df_test = df_test.dropna(subset=col)
    plt.figure(figsize=(8, 6))
    df_test.boxplot(column=col, by='OutPut')
    plt.title(f'Box Plot of {col}')
    plt.suptitle('')  # Remove the default title
    plt.ylabel(col)
    plt.show();
```

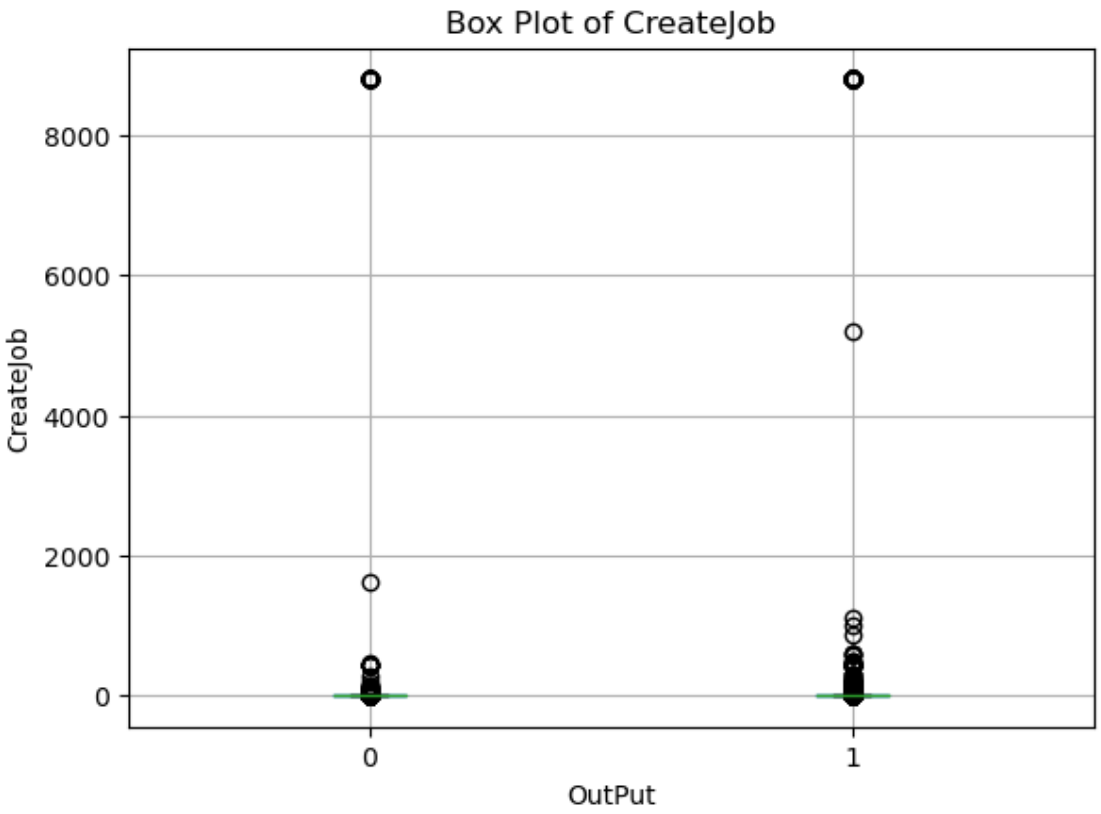
```
<Figure size 800x600 with 0 Axes>
```



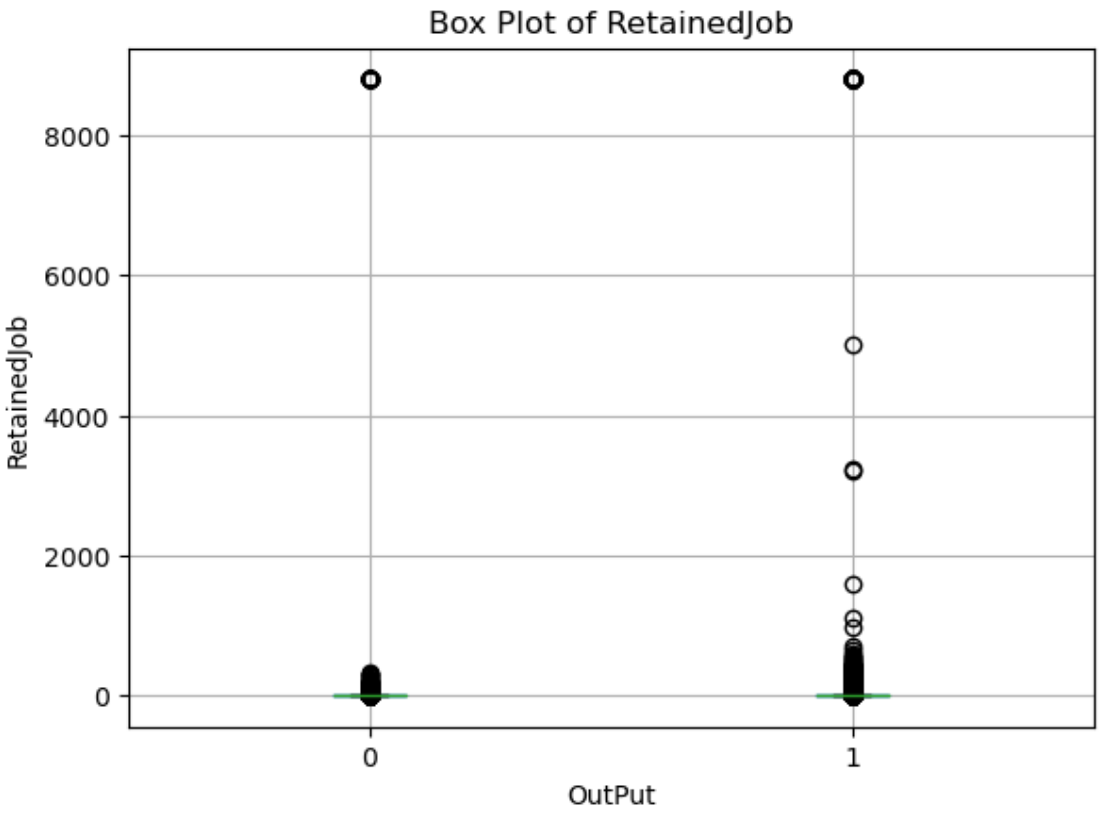

```
<Figure size 800x600 with 0 Axes>
```



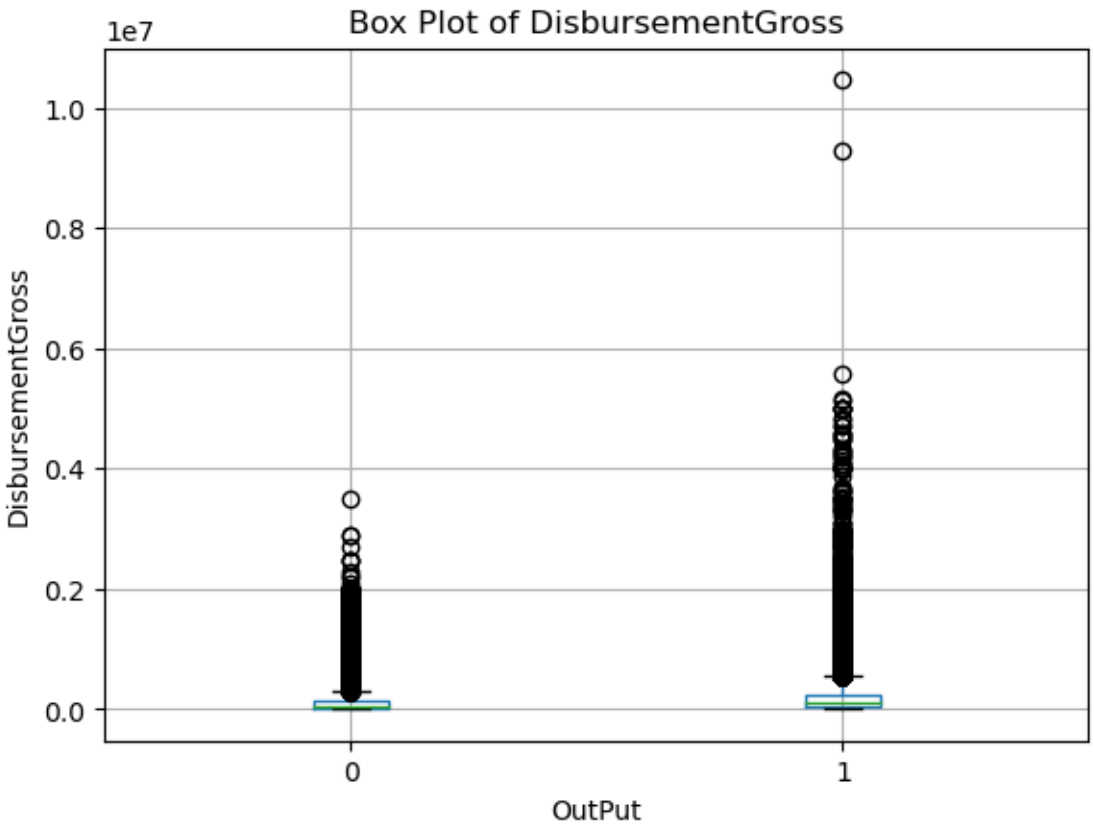

```
<Figure size 800x600 with 0 Axes>
```




```
<Figure size 800x600 with 0 Axes>
```

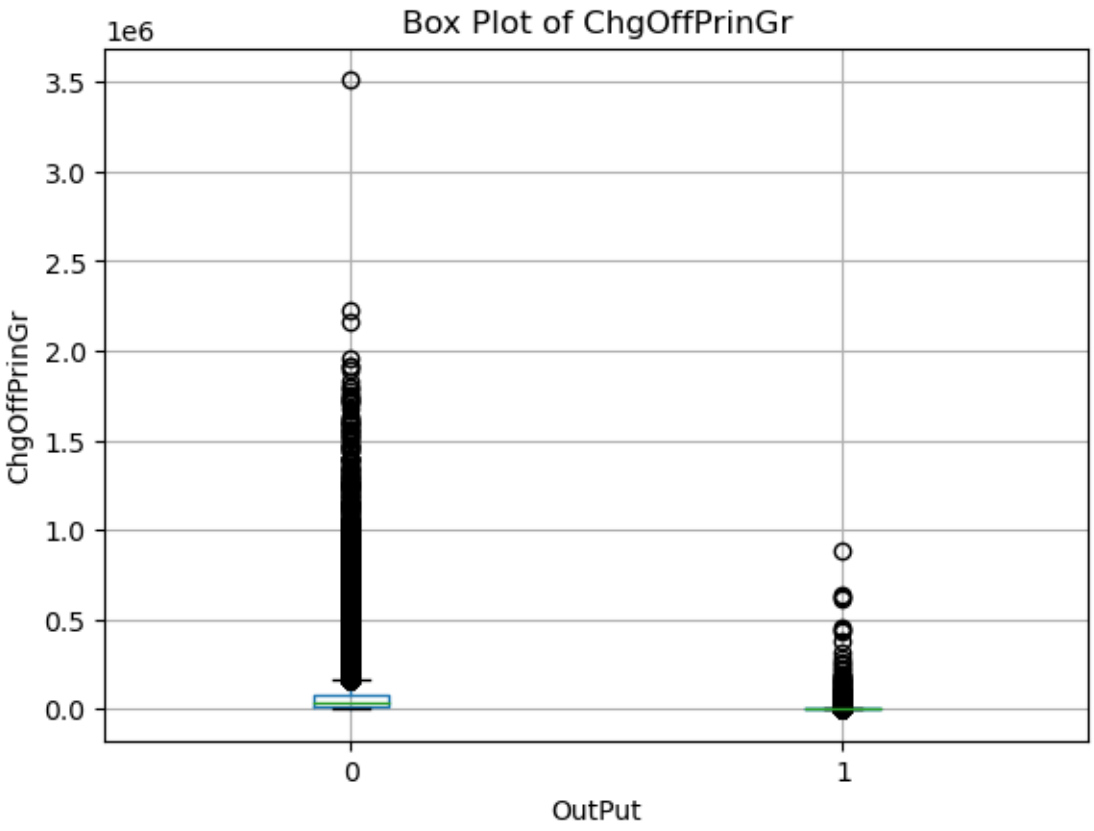



```
<Figure size 800x600 with 0 Axes>
```

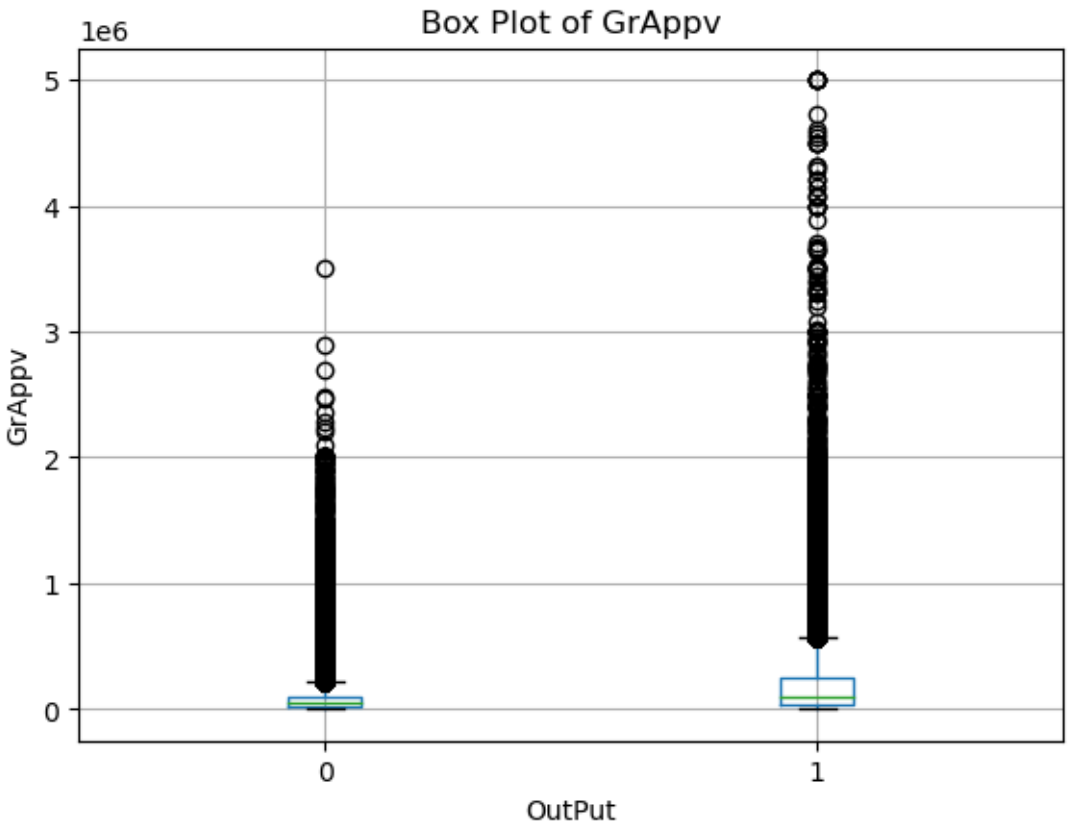



```
<Figure size 800x600 with 0 Axes>
```

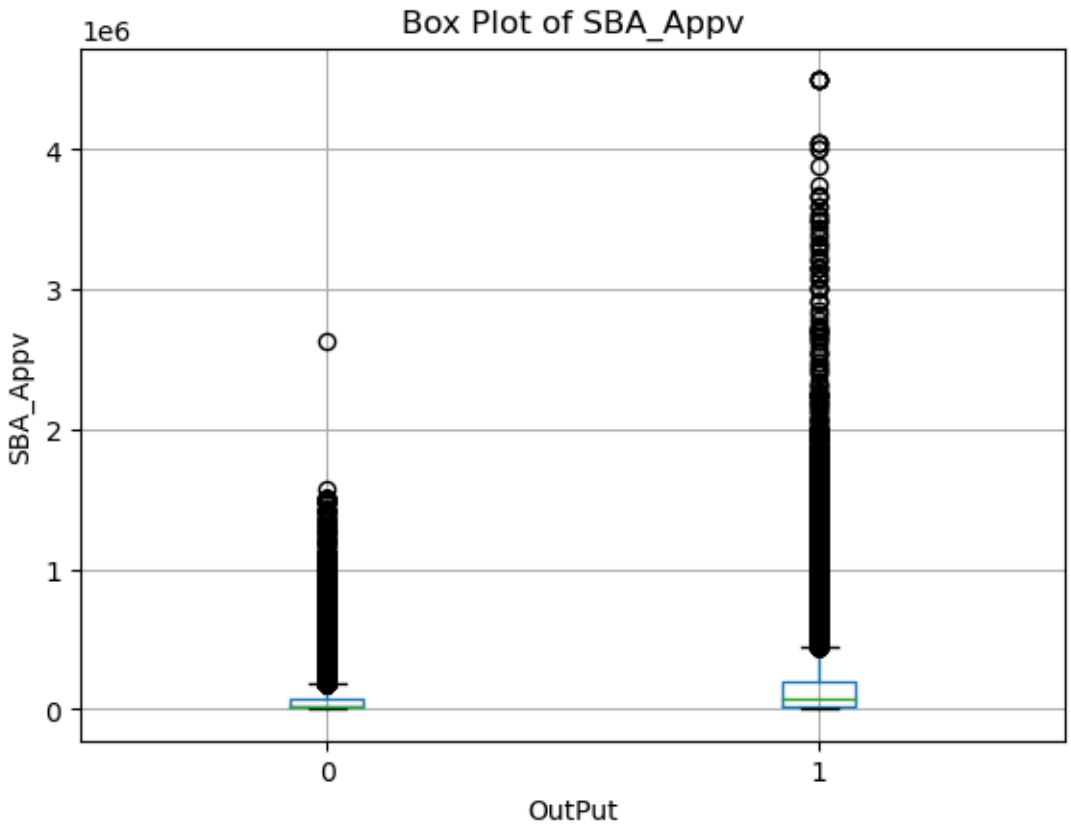

```
<Figure size 800x600 with 0 Axes>
```




```
<Figure size 800x600 with 0 Axes>
```



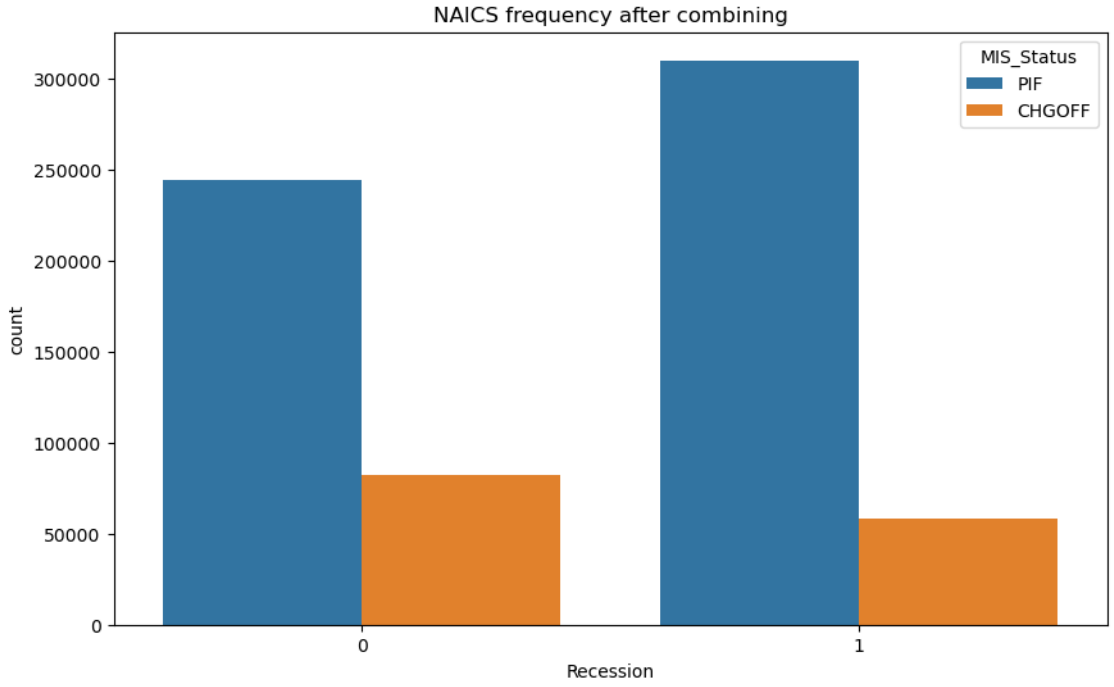

```
<Figure size 800x600 with 0 Axes>
```



[26]:
```
df['UrbanRural'].unique()
```

[26]:
```
array([0, 1, 2])
```

[27]:
```
df['ToDate'] = df['DisbursementDate'] + pd.to_timedelta(df['Term']*30, unit='D')
df['Recession'] = (df['DisbursementDate'] < '2007-12-01') & ( df['ToDate'] >=␣
↪'2009-06-30')
df['Recession'] = df['Recession'].map({True: 1, False: 0})

plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Recession', hue='MIS_Status')
plt.title('NAICS frequency after combining')
plt.show()
```



[28]:
```
# Correlation between Numerical and Categorical
correlation, p_value = stats.pointbiserialr(df['OutPut'],␣
 ↪df['RealEstate']*df['Portion'])

print(f"Point-biserial correlation: {correlation:.2f}")
print(f"P-value: {p_value:.4f}")
```

```
Point-biserial correlation: 0.19
P-value: 0.0000
```

[29]:
```
# Correlation between categorical and categorical

contingency_table = pd.crosstab(df['MIS_Status'], df['NoEmp'])

# Calculate Cramer's V
chi2, _, _, _ = stats.chi2_contingency(contingency_table)
n = np.sum(contingency_table)
min_dim = min(contingency_table.shape)
cramer_v = np.sqrt(chi2 / (n * (min_dim - 1)))

print(f"Cramer's V: {cramer_v}")
```

```
Cramer's V: NoEmp
0         1.262204
1         0.283989
2         0.302298
```

```
3          0.379349
4          0.426364
            …
8000      72.595957
8041     102.666187
9000     102.666187
9992     102.666187
9999     102.666187
Length: 501, dtype: float64
```

[30]:
```
df['NewExist'] >= 2
```

[30]:
```
0           True
1           True
2          False
5          False
7           True
           …
899156     False
899157     False
899159     False
899160     False
899161     False
Name: NewExist, Length: 695500, dtype: bool
```

[31]:
```
df['NewExist'] = df.apply(lambda row: 2 if pd.isna(row['NewExist']) and␣
 ↪row['MIS_Status'] == 'PIF' else row['NewExist'], axis=1)
df['NewExist'] = df.apply(lambda row: 1 if pd.isna(row['NewExist']) and␣
 ↪row['MIS_Status'] == 'CHGOFF' else row['NewExist'], axis=1)
```