visualization

June 7, 2017

1 Part 1 -- Visualisation

The libraries that we are going to use:

```
In [81]: import pandas as pd
         import matplotlib
         import numpy as np
         import matplotlib.mlab as mlab
         import matplotlib.pyplot as plt
         import matplotlib
         matplotlib.style.use('ggplot')
```

We read our data:

```
In [82]: mydata = pd.read_csv(sep='\t',filepath_or_buffer='train.tsv')

In [83]: mydata.head()
```

Out[83]:

	Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6	\
0	A11	6	A34	A43	1169	A65	
1	A12	48	A32	A43	5951	A61	
2	A14	12	A34	A46	2096	A61	
3	A11	42	A32	A42	7882	A61	
4	A11	24	A33	A40	4870	A61	

	Attribute7	Attribute8	Attribute9	Attribute10	...	Attribute13	\
0	A75	4	A93	A101	...	67	
1	A73	2	A92	A101	...	22	
2	A74	2	A93	A101	...	49	
3	A74	2	A93	A103	...	45	
4	A73	3	A93	A101	...	53	

	Attribute14	Attribute15	Attribute16	Attribute17	Attribute18	Attribute19	\
0	A143	A152	2	A173	1	A192	
1	A143	A152	1	A173	1	A191	
2	A143	A152	1	A172	2	A191	
3	A143	A153	1	A173	2	A191	
4	A143	A153	2	A173	2	A191	

```
   Attribute20 Label     Id
0         A201     1  10101
1         A201     2  10102
2         A201     1  10103
3         A201     1  10104
4         A201     2  10105

[5 rows x 22 columns]
```

```
In [84]: set(mydata['Attribute14'])
```

```
Out[84]: {'A141', 'A142', 'A143'}
```

1.1 Preprocessing

We preprocess our data and we encode the various attributes

```
In [85]: #preprocessing
         proccessedData = mydata #gia kathe attribute an einai categorical to kanoume aplh antis

         #kanoume encode to attribute1
         integer_map1 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute1']))
         for x in set(mydata['Attribute1']): #antikathistoume kathe ena me ton antistoixo arithm
             proccessedData = proccessedData.replace(x, integer_map1[x])
         print(integer_map1)

         #kanoume encode to attribute3
         integer_map3 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute3']))
         for x in set(mydata['Attribute3']): #antikathistoume kathe ena me ton antistoixo arithm
             proccessedData = proccessedData.replace(x, integer_map3[x])
         print(integer_map3)

         #kanoume encode to attribute4
         integer_map4 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute4']))
         for x in set(mydata['Attribute4']): #antikathistoume kathe ena me ton antistoixo arithm
             proccessedData = proccessedData.replace(x, integer_map4[x])
         print(integer_map4)

         #kanoume encode to attribute6
         integer_map6 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute6']))
         for x in set(mydata['Attribute6']): #antikathistoume kathe ena me ton antistoixo arithm
             proccessedData = proccessedData.replace(x, integer_map6[x])
         print(integer_map6)

         #kanoume encode to attribute7
         integer_map7 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute7']))
         for x in set(mydata['Attribute7']): #antikathistoume kathe ena me ton antistoixo arithm
             proccessedData = proccessedData.replace(x, integer_map7[x])
```

```
print(integer_map7)

#kanoume encode to attribute9
integer_map9 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute9']))
for x in set(mydata['Attribute9']): #antikathistoume kathe ena me ton antistoixo arithm
    proccessedData = proccessedData.replace(x, integer_map9[x])
print(integer_map9)

#kanoume encode to attribute10
integer_map10 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute10'
for x in set(mydata['Attribute10']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map10[x])
print(integer_map10)

#kanoume encode to attribute12
integer_map12 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute12'
for x in set(mydata['Attribute12']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map12[x])
print(integer_map12)

#kanoume encode to attribute14
integer_map14 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute14'
for x in set(mydata['Attribute14']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map14[x])
print(integer_map14)

#kanoume encode to attribute15
integer_map15 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute15'
for x in set(mydata['Attribute15']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map15[x])
print(integer_map15)

#kanoume encode to attribute17
integer_map17 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute17'
for x in set(mydata['Attribute17']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map17[x])
print(integer_map17)

#kanoume encode to attribute19
integer_map19 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute19'
for x in set(mydata['Attribute19']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map19[x])
print(integer_map19)

#kanoume encode to attribute20
integer_map20 = dict([(val, i) for i, val in enumerate(set(proccessedData['Attribute20'
for x in set(mydata['Attribute20']): #antikathistoume kathe ena me ton antistoixo arith
    proccessedData = proccessedData.replace(x, integer_map20[x])
```

```
        print(integer_map20)
```

```
{'A11': 0, 'A14': 1, 'A12': 3, 'A13': 2}
{'A31': 3, 'A33': 2, 'A34': 4, 'A30': 1, 'A32': 0}
{'A42': 0, 'A48': 1, 'A43': 5, 'A410': 2, 'A44': 6, 'A40': 3, 'A49': 8, 'A45': 4, 'A46': 9, 'A41
{'A64': 1, 'A63': 0, 'A65': 3, 'A61': 2, 'A62': 4}
{'A74': 0, 'A72': 1, 'A71': 4, 'A75': 2, 'A73': 3}
{'A92': 0, 'A91': 1, 'A94': 2, 'A93': 3}
{'A101': 0, 'A103': 1, 'A102': 2}
{'A124': 0, 'A121': 1, 'A122': 3, 'A123': 2}
{'A141': 0, 'A143': 1, 'A142': 2}
{'A152': 0, 'A151': 1, 'A153': 2}
{'A173': 0, 'A172': 1, 'A171': 2, 'A174': 3}
{'A191': 0, 'A192': 1}
{'A201': 0, 'A202': 1}
```

```
In [86]: set(mydata['Attribute1'])
```

```
Out[86]: {'A11', 'A12', 'A13', 'A14'}
```

```
In [87]: # map attribute's one data to a dictionary
         count = 0;
         mymap = []
         data = set(mydata['Attribute1'])
         print(data)
         print(sorted(data))
         for x in sorted(data):
             mymap.append(count);
             print(x)
             count+=1
         print(mymap)
```

```
{'A11', 'A14', 'A13', 'A12'}
['A11', 'A12', 'A13', 'A14']
A11
A12
A13
A14
[0, 1, 2, 3]
```

```
In [88]: proccessedData.head()
         Bad = proccessedData[proccessedData['Label']==2]
         Good = proccessedData[proccessedData['Label']==1]
```

1.2 Printing our Plots

1.2.1 Categorical Data Plots

```
In [89]: #kanoume ola ta data arithmous gia na mporoume na ftiaksoume to histogram
         Good.head()
```

Out[89]:

	Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6	\
0	0	6	4	5	1169	3	
2	1	12	4	9	2096	2	
3	0	42	0	0	7882	2	
5	1	36	0	9	9055	3	
6	1	24	0	0	2835	0	

	Attribute7	Attribute8	Attribute9	Attribute10	...	Attribute13	\
0	2	4	3	0	...	67	
2	0	2	3	0	...	49	
3	0	2	3	1	...	45	
5	3	2	3	0	...	35	
6	2	3	3	0	...	53	

	Attribute14	Attribute15	Attribute16	Attribute17	Attribute18	\
0	1	0	2	0	1	
2	1	0	1	1	2	
3	1	2	1	0	2	
5	1	2	1	1	2	
6	1	0	1	0	1	

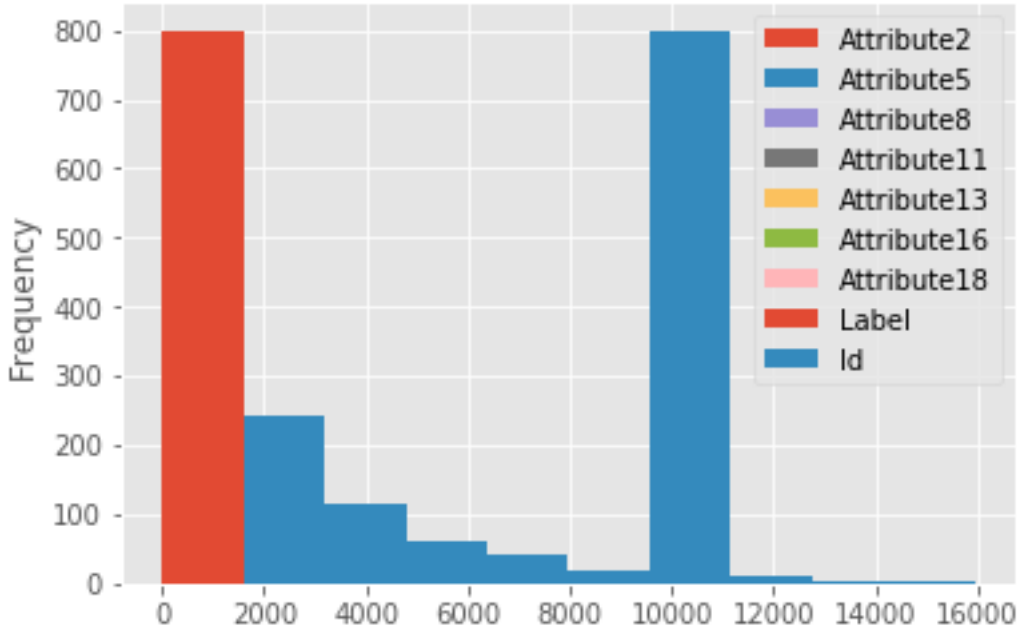
	Attribute19	Attribute20	Label	Id
0	1	0	1	10101
2	0	0	1	10103
3	0	0	1	10104
5	1	0	1	10106
6	0	0	1	10107

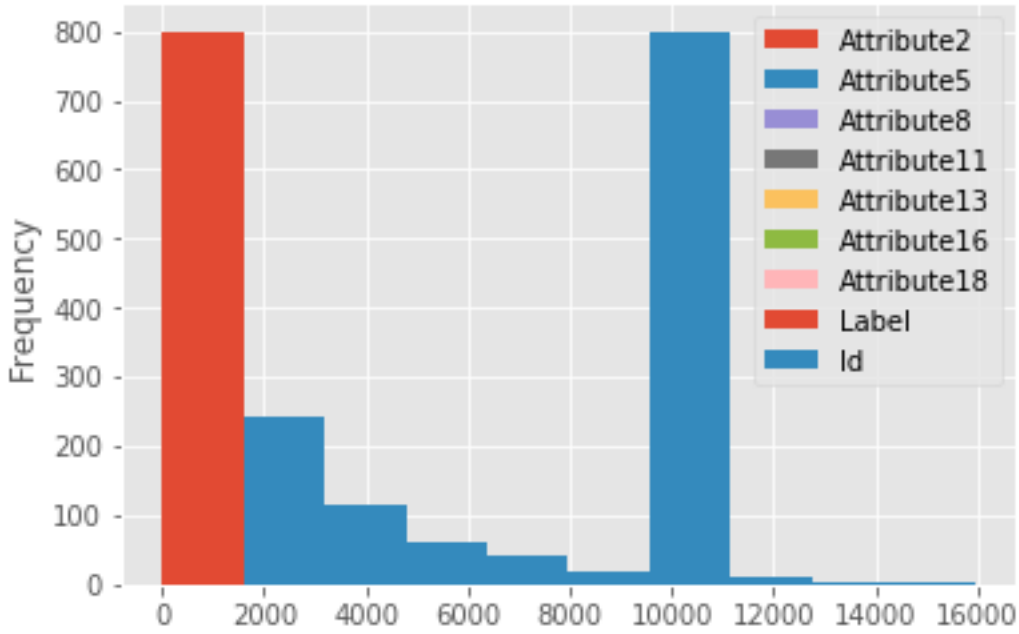
[5 rows x 22 columns]

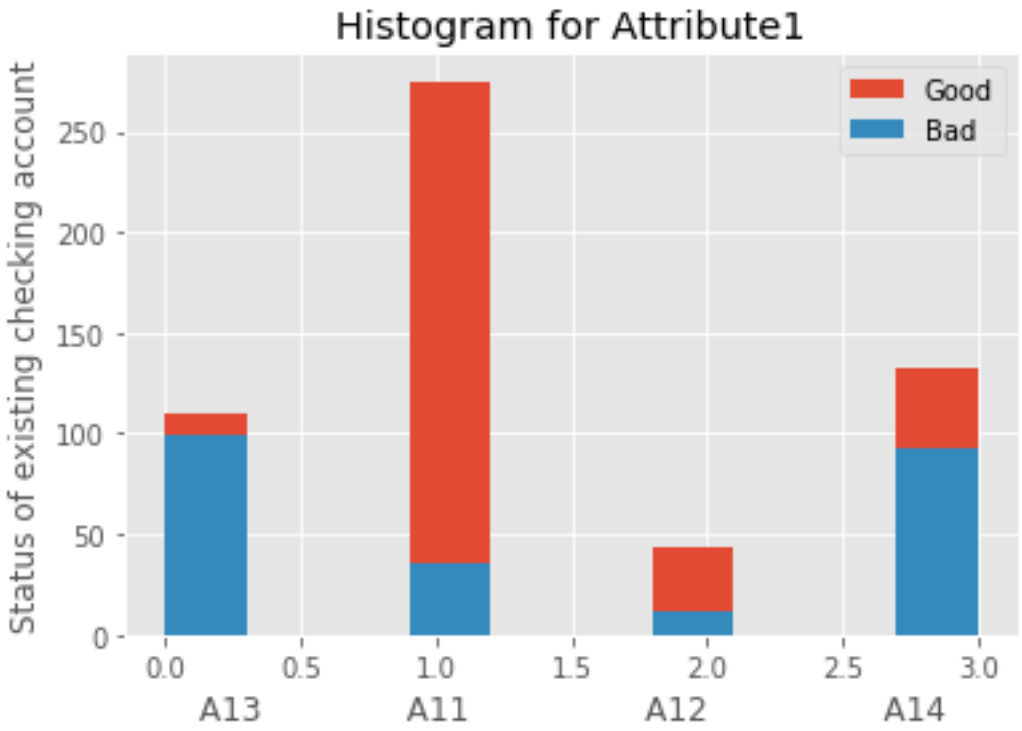
In [ ]:

In [90]:
```
# the histogram of the categorical data of Attribute 1 -- Good and Bad
print("Encoding : ", integer_map1)

df = pd.DataFrame({'Good':Good["Attribute1"],'Bad':Bad["Attribute1"]},columns=['Good','

df.plot.hist()
plt.title("Histogram for Attribute1")
plt.ylabel('Status of existing checking account')
plt.xlabel('A13              A11                A12                A14')
plt.show()
```

```
Encoding :  {'A11': 0, 'A14': 1, 'A12': 3, 'A13': 2}
```

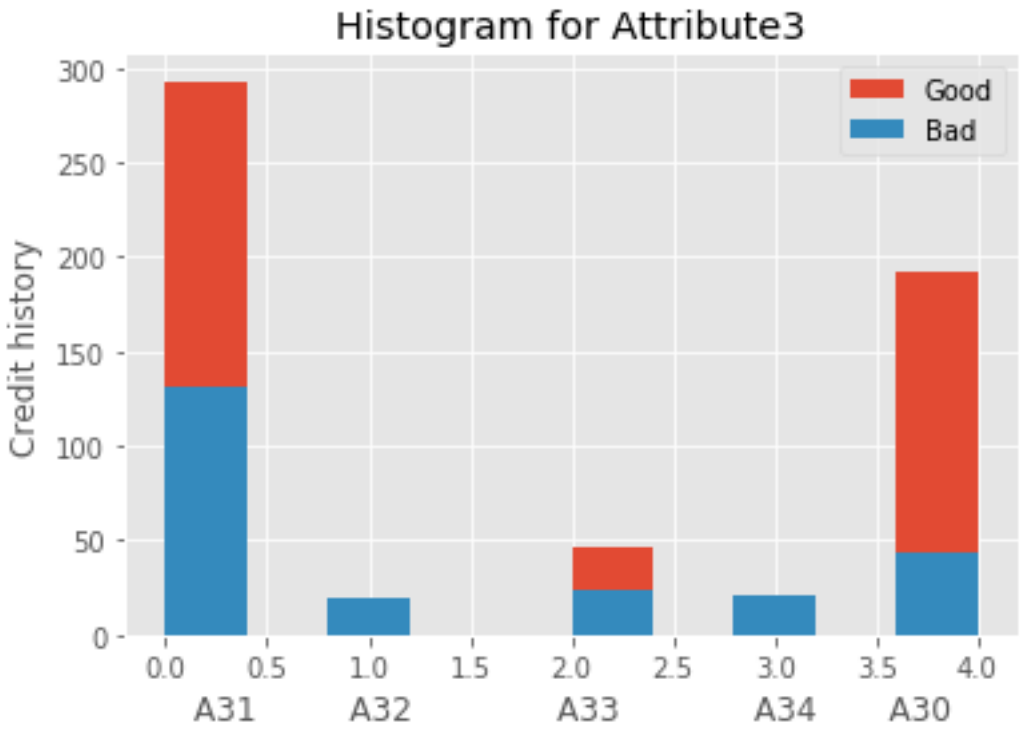

Frequency
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000
14000
16000
Attribute2
Attribute5
Attribute8
Attribute11
Attribute13
Attribute16
Attribute18
Label
Id


Frequency
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000
14000
16000
Attribute2
Attribute5
Attribute8
Attribute11
Attribute13
Attribute16
Attribute18
Label
Id



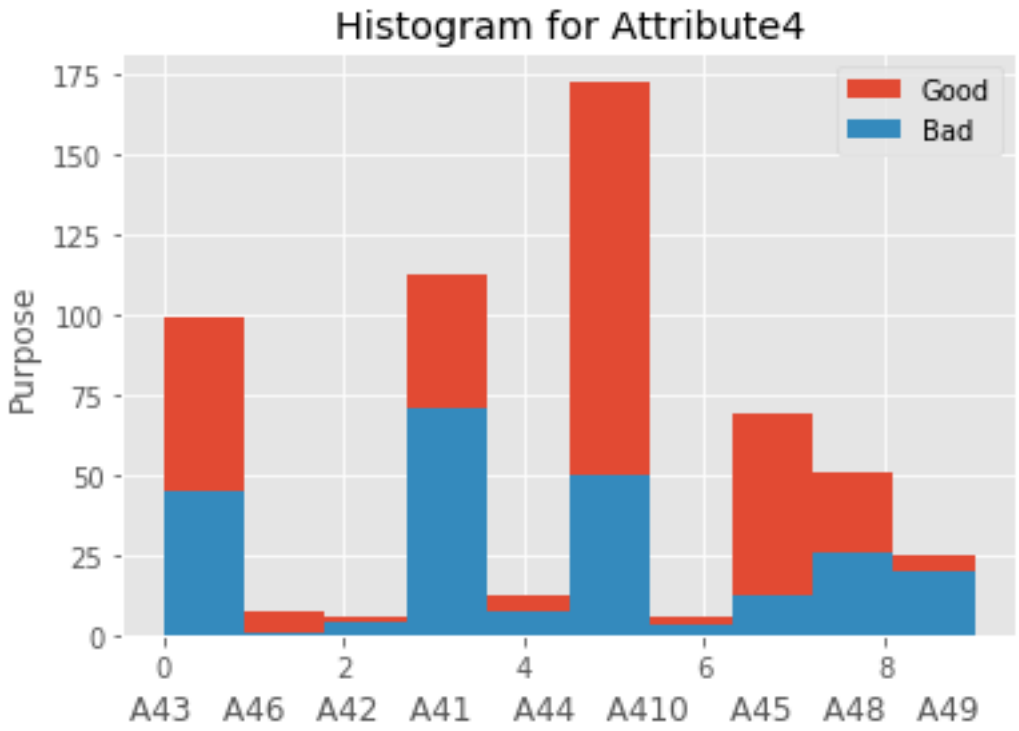

```
In [91]: # the histogram of the categorical data of Attribute 3 -- Good and Bad
         print("Encoding : ", integer_map3)

         df = pd.DataFrame({'Good':Good["Attribute3"],'Bad':Bad["Attribute3"]},columns=['Good','

         df.plot.hist()
         plt.title("Histogram for Attribute3")
         plt.ylabel('Credit history')
         plt.xlabel('A31         A32              A33              A34       A30')
         plt.show()
```

```
Encoding :  {'A31': 3, 'A33': 2, 'A34': 4, 'A30': 1, 'A32': 0}
```



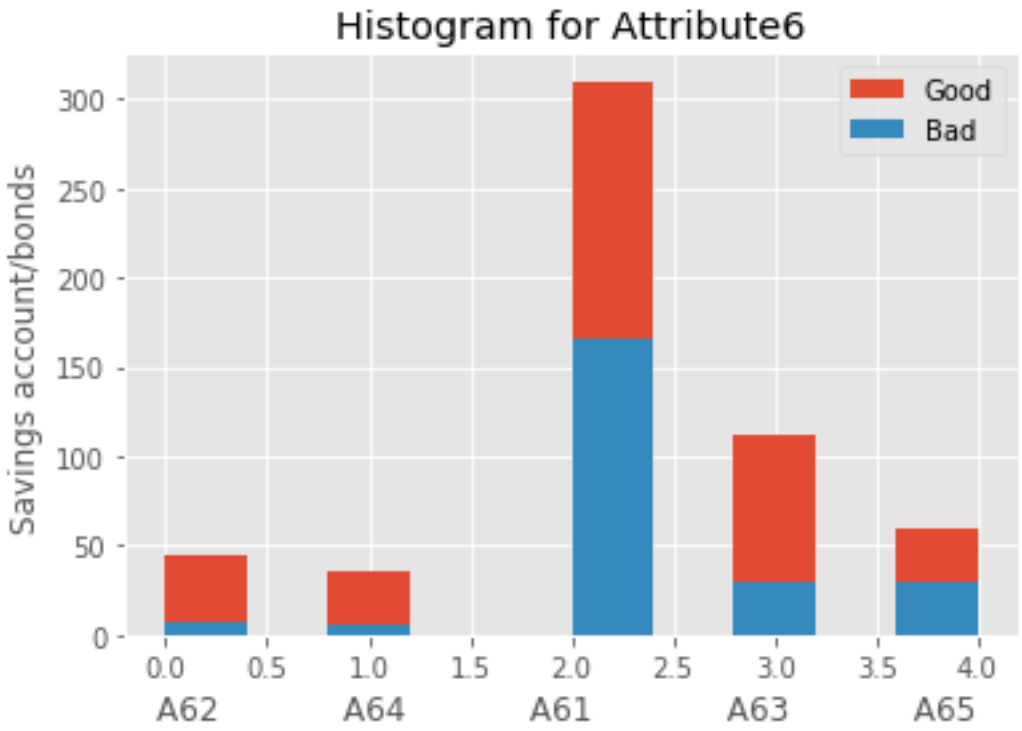
In [92]:
```
# the histogram of the categorical data of Attribute 4 -- Good and Bad
print("Encoding : ", integer_map4)

df = pd.DataFrame({'Good':Good["Attribute4"],'Bad':Bad["Attribute4"]},columns=['Good','

df.plot.hist()
plt.title("Histogram for Attribute4")
plt.ylabel('Purpose')
plt.xlabel('A43   A46   A42   A41   A44   A410   A45   A48   A49   ')
plt.show()
```

```
Encoding :  {'A42': 0, 'A48': 1, 'A43': 5, 'A410': 2, 'A44': 6, 'A40': 3, 'A49': 8, 'A45': 4, 'A
```



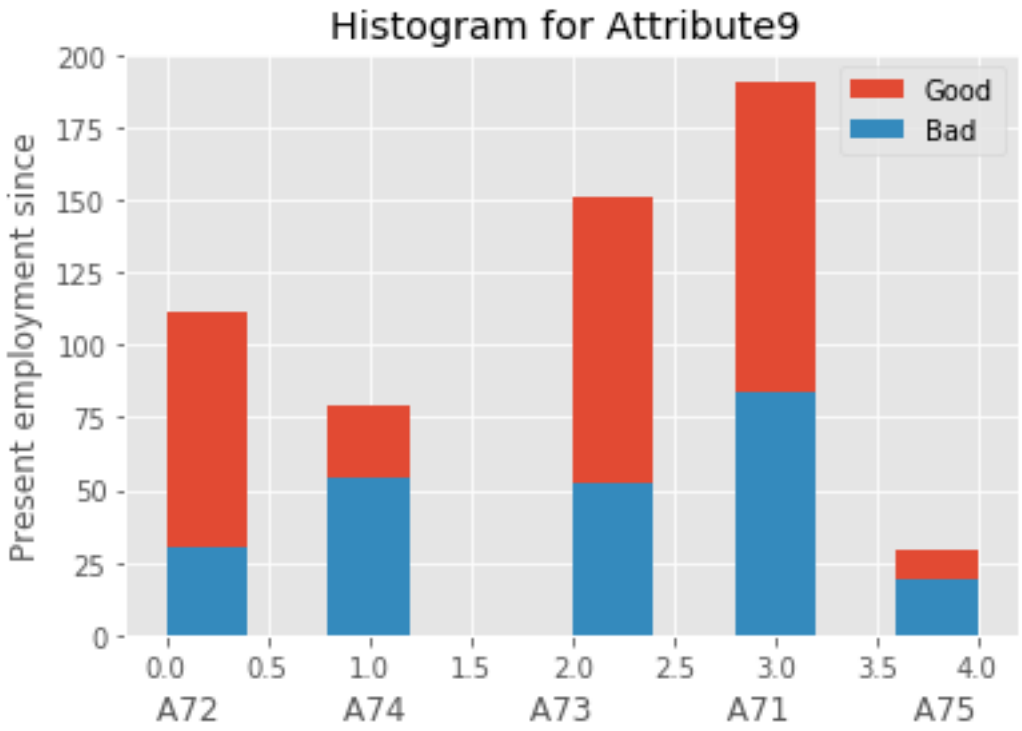

```
In [93]: # the histogram of the categorical data of Attribute 6 -- Good and Bad
         print("Encoding : ", integer_map6)

         df = pd.DataFrame({'Good':Good["Attribute6"],'Bad':Bad["Attribute6"]},columns=['Good','

         df.plot.hist()
         plt.title("Histogram for Attribute6")
         plt.ylabel('Savings account/bonds')
         plt.xlabel('A62            A64            A61            A63            A65 ')
         plt.show()
```

```
Encoding :  {'A64': 1, 'A63': 0, 'A65': 3, 'A61': 2, 'A62': 4}
```

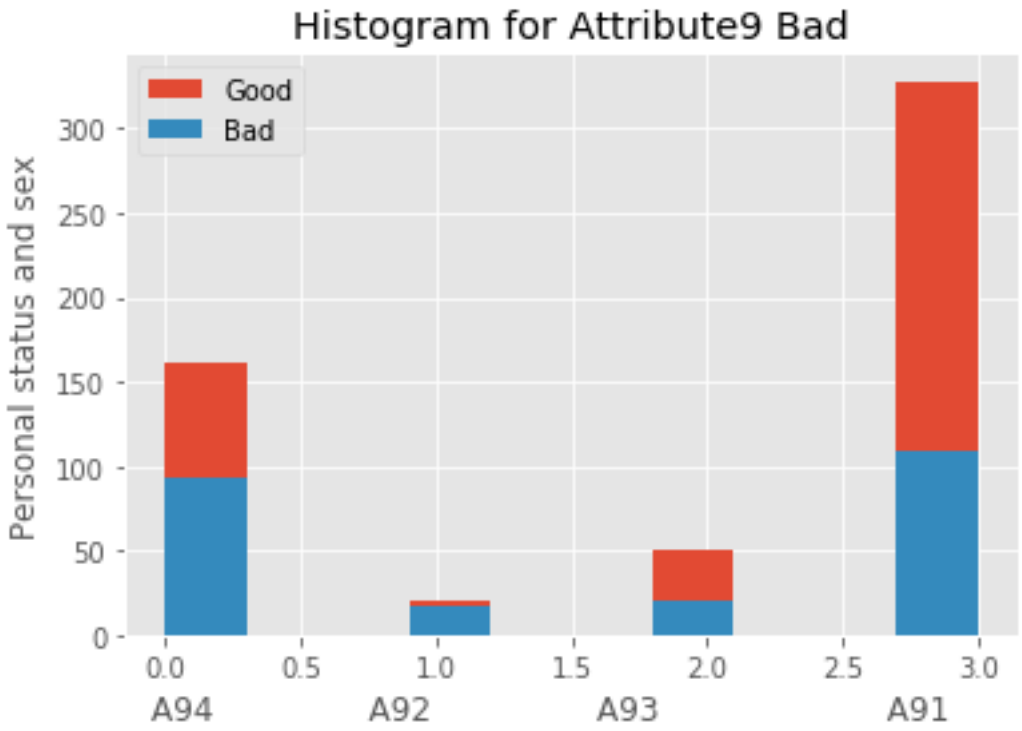



```
In [94]: # the histogram of the categorical data of Attribute 7 -- Good and Bad
         print("Encoding : ", integer_map7)

         df = pd.DataFrame({'Good':Good["Attribute7"],'Bad':Bad["Attribute7"]},columns=['Good','

         df.plot.hist()
         plt.title("Histogram for Attribute9 ")
         plt.ylabel('Present employment since')
         plt.xlabel('A72            A74            A73            A71            A75 ')
         plt.show()
```

```
Encoding :  {'A74': 0, 'A72': 1, 'A71': 4, 'A75': 2, 'A73': 3}
```

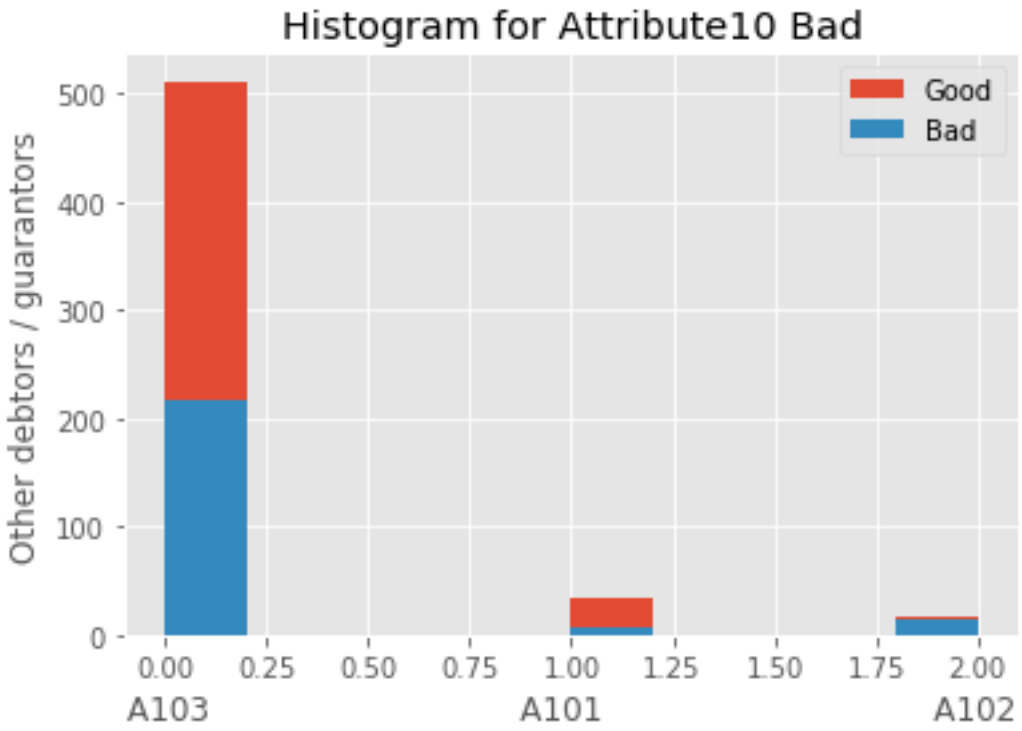



```
In [95]: # the histogram of the categorical data of Attribute 9 -- Good and Bad
         print("Encoding : ", integer_map9)

         df = pd.DataFrame({'Good':Good["Attribute9"],'Bad':Bad["Attribute9"]},columns=['Good','

         df.plot.hist()
         plt.title("Histogram for Attribute9 Bad")
         plt.ylabel('Personal status and sex')
         plt.xlabel('A94              A92              A93                    A91    ')
         plt.show()

Encoding :  {'A92': 0, 'A91': 1, 'A94': 2, 'A93': 3}
```



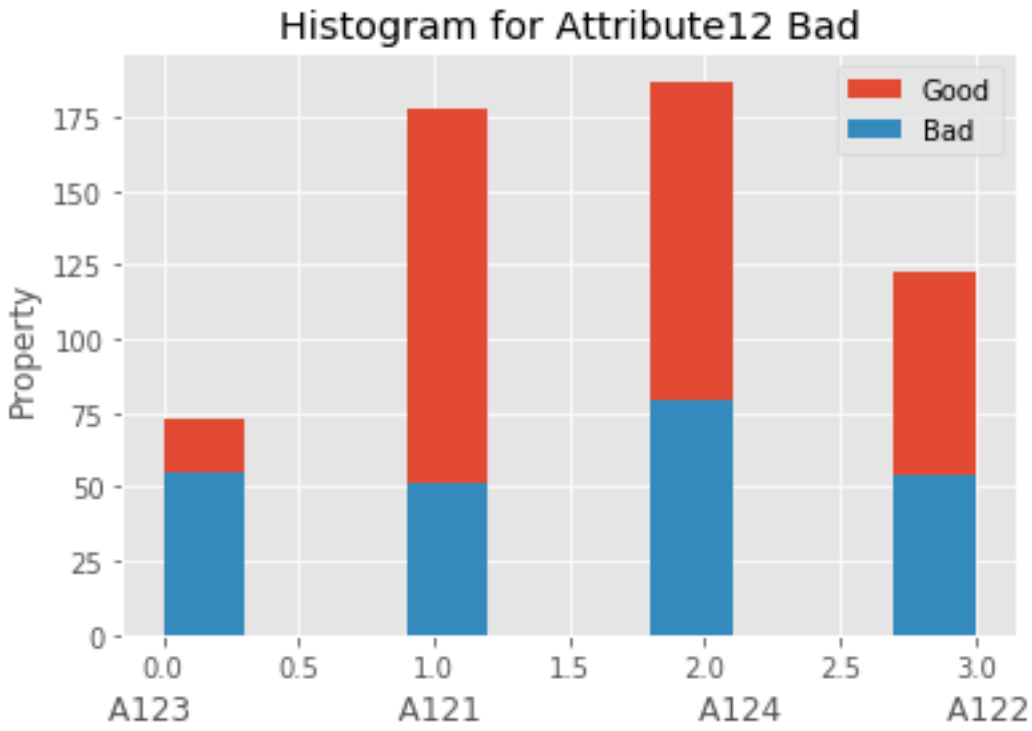
In [96]:
```
# the histogram of the categorical data of Attribute 10 -- Good and Bad
print("Encoding : ", integer_map10)

df = pd.DataFrame({'Good':Good["Attribute10"],'Bad':Bad["Attribute10"]},columns=['Good'

df.plot.hist()
plt.title("Histogram for Attribute10 Bad")
plt.ylabel('Other debtors / guarantors')
plt.xlabel('A103                                A101                                A102'
plt.show()
```

```
Encoding :  {'A101': 0, 'A103': 1, 'A102': 2}
```



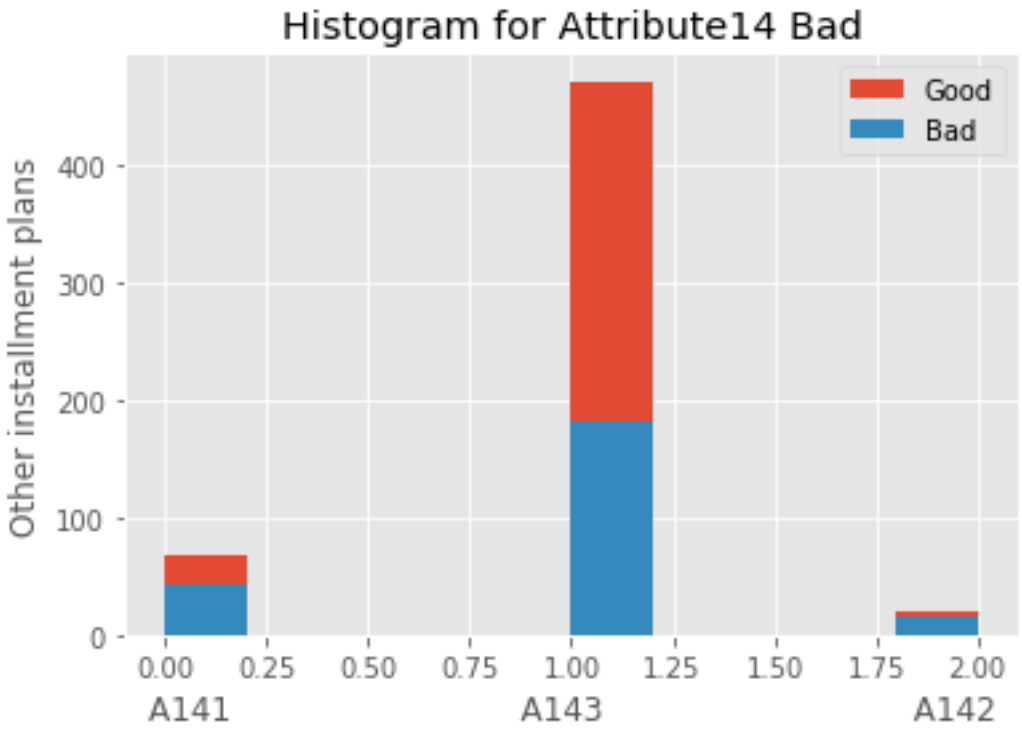
In [97]:
```
# the histogram of the categorical data of Attribute 12 -- Good and Bad
print("Encoding : ", integer_map12)

df = pd.DataFrame({'Good':Good["Attribute12"],'Bad':Bad["Attribute12"]},columns=['Good'

df.plot.hist()
plt.title("Histogram for Attribute12 Bad")
plt.ylabel('Property')
plt.xlabel('A123                    A121                    A124                A122')
plt.show()
```

```
Encoding :  {'A124': 0, 'A121': 1, 'A122': 3, 'A123': 2}
```



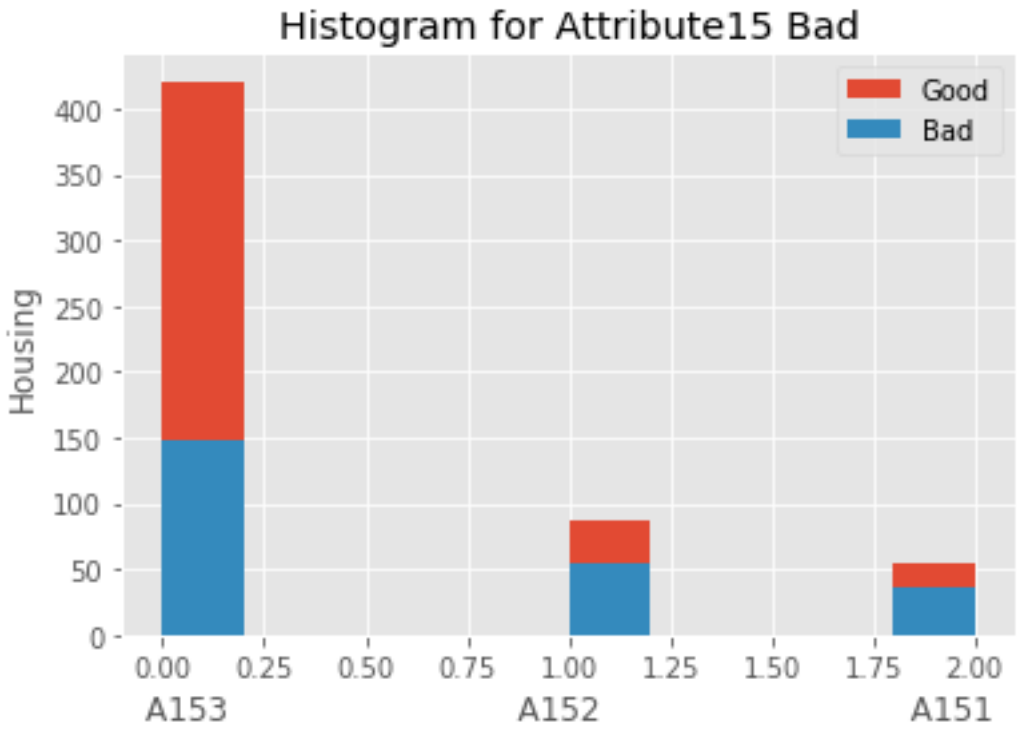
In [98]:
```
# the histogram of the categorical data of Attribute 14 -- Good and Bad
print("Encoding : ", integer_map14)

df = pd.DataFrame({'Good':Good["Attribute14"],'Bad':Bad["Attribute14"]},columns=['Good'

df.plot.hist()
plt.title("Histogram for Attribute14 Bad")
plt.ylabel('Other installment plans')
plt.xlabel('A141                                A143                                A142')
plt.show()
```

```
Encoding :  {'A141': 0, 'A143': 1, 'A142': 2}
```



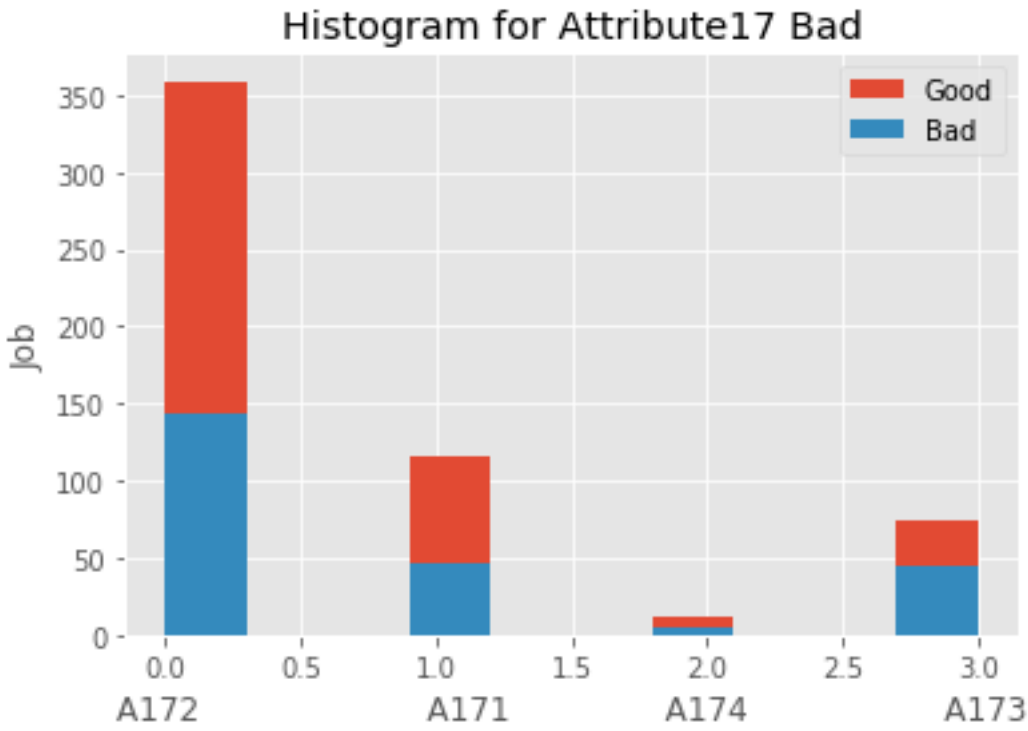

```
In [99]: # the histogram of the categorical data of Attribute 15 -- Good and Bad
         print("Encoding : ", integer_map15)

         df = pd.DataFrame({'Good':Good["Attribute15"],'Bad':Bad["Attribute15"]},columns=['Good'

         df.plot.hist()
         plt.title("Histogram for Attribute15 Bad")
         plt.ylabel('Housing')
         plt.xlabel('A153                                A152                                A151')
         plt.show()
```

```
Encoding :  {'A152': 0, 'A151': 1, 'A153': 2}
```

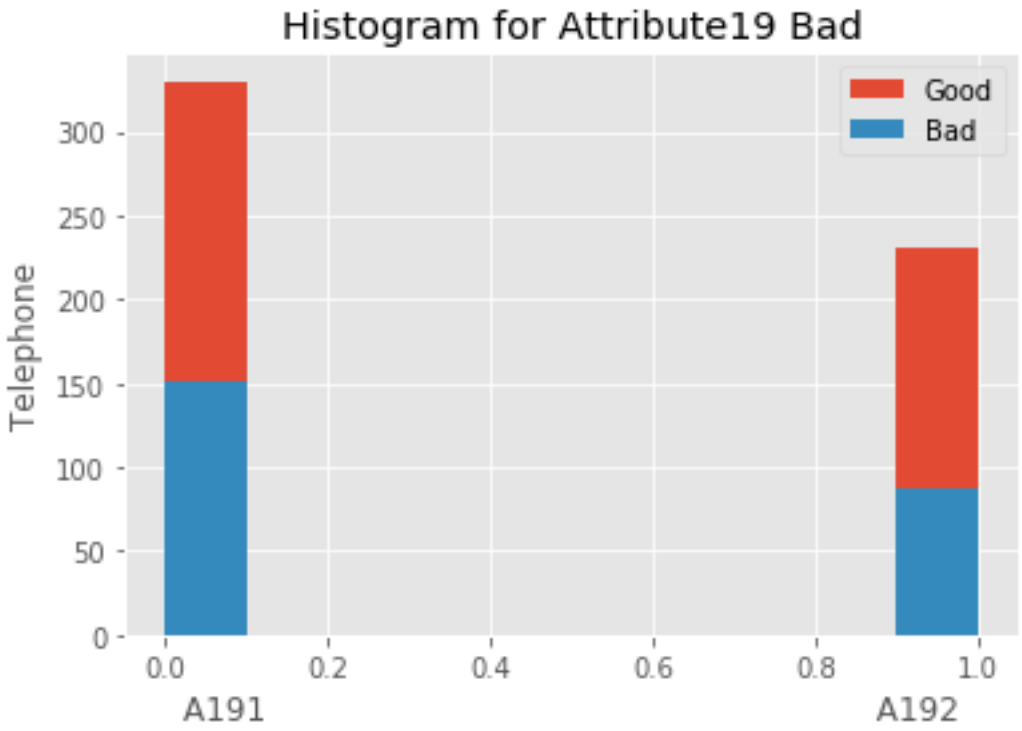



```
In [100]: # the histogram of the categorical data of Attribute 17 -- Good and Bad
          print("Encoding : ", integer_map17)

          df = pd.DataFrame({'Good':Good["Attribute17"],'Bad':Bad["Attribute17"]},columns=['Good

          df.plot.hist()
          plt.title("Histogram for Attribute17 Bad")
          plt.ylabel('Job')
          plt.xlabel('A172                    A171              A174                    A173')
          plt.show()
```

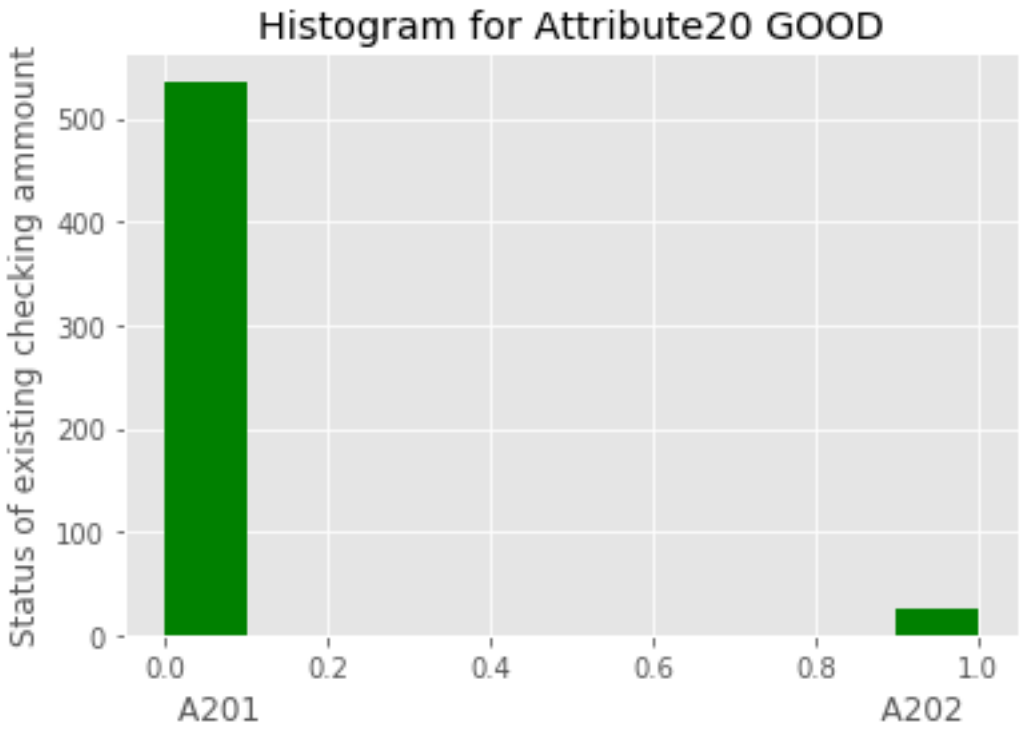
```
Encoding :  {'A173': 0, 'A172': 1, 'A171': 2, 'A174': 3}
```



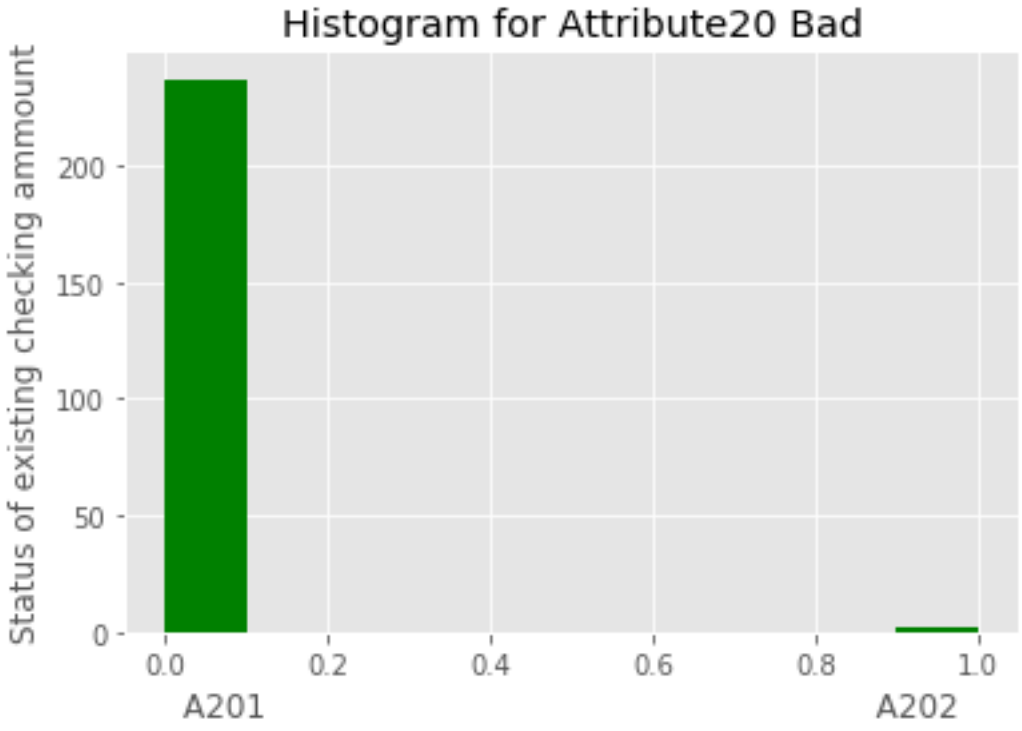

```
In [101]: # the histogram of the categorical data of Attribute 19 -- Good and Bad
          print("Encoding : ", integer_map19)

          df = pd.DataFrame({'Good':Good["Attribute19"],'Bad':Bad["Attribute19"]},columns=['Good

          df.plot.hist()
          plt.title("Histogram for Attribute19 Bad")
          plt.ylabel('Telephone')
          plt.xlabel('A191                                                                    A192')
          plt.show()

Encoding :  {'A191': 0, 'A192': 1}
```




```
In [102]: # the histogram of the categorical data of Attribute 20 -- Good
          print("Encoding : ", integer_map20)

          plt.hist(Good["Attribute20"], facecolor='green')
          plt.title("Histogram for Attribute20 GOOD")
          plt.ylabel('Status of existing checking ammount')
          plt.xlabel('A201                                                    A202')
          plt.show()

          # the histogram of the categorical data of Attribute 20 -- Bad
          print("Encoding : ", integer_map20)

          plt.hist(Bad["Attribute20"], facecolor='green')
          plt.title("Histogram for Attribute20 Bad")
          plt.ylabel('Status of existing checking ammount')
          plt.xlabel('A201                                                    A202')
          plt.show()

Encoding :  {'A201': 0, 'A202': 1}
```

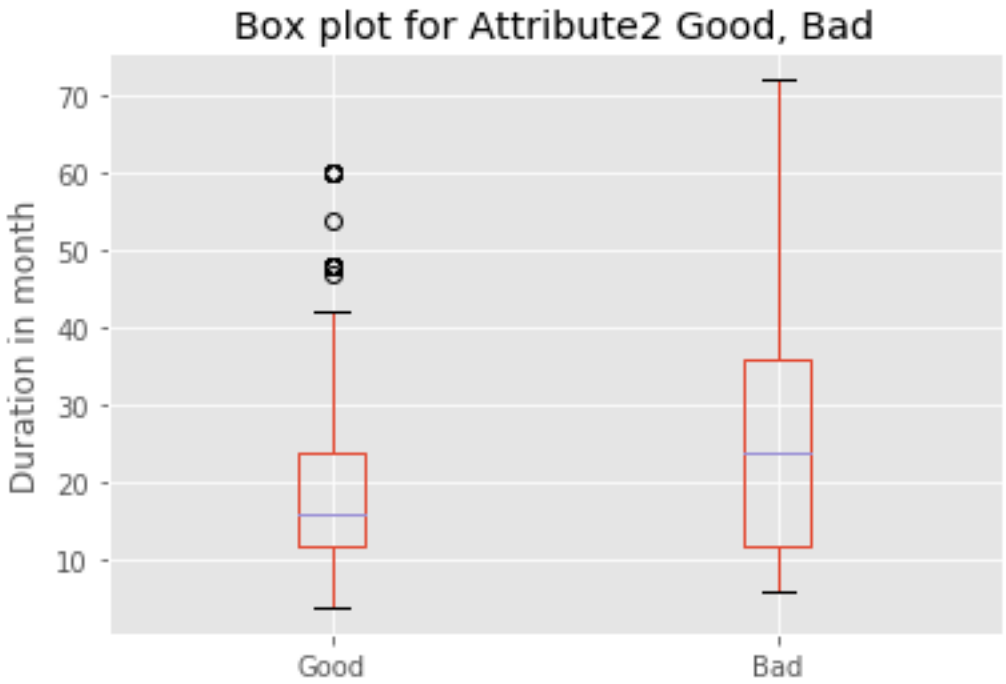



```
Encoding :  {'A201': 0, 'A202': 1}
```

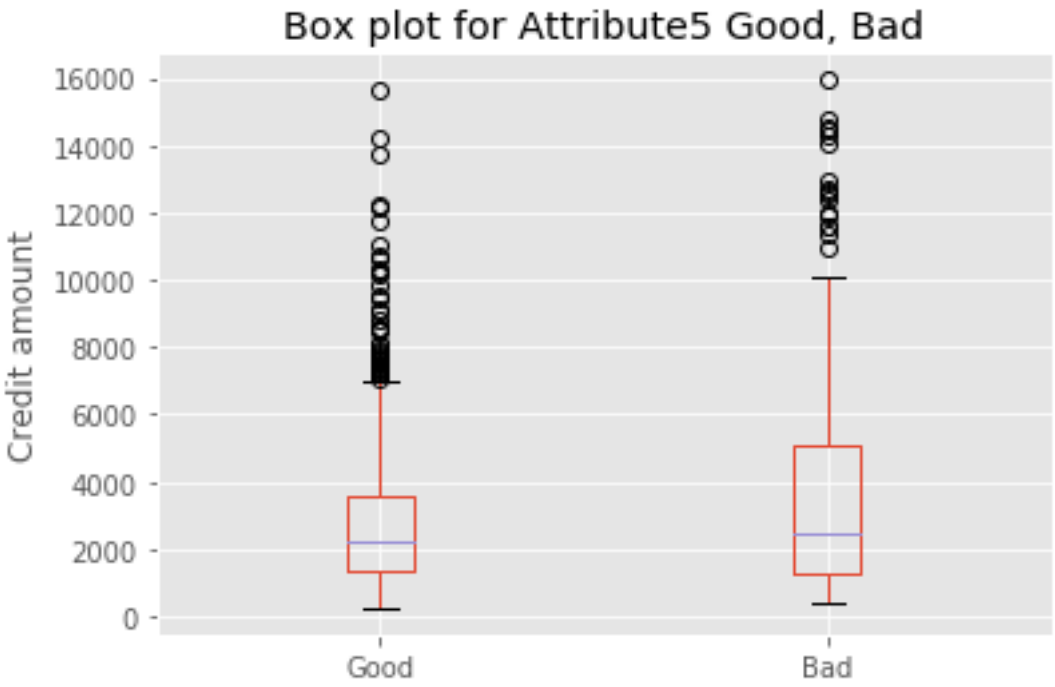


1.2.2 Numerical Data Plots

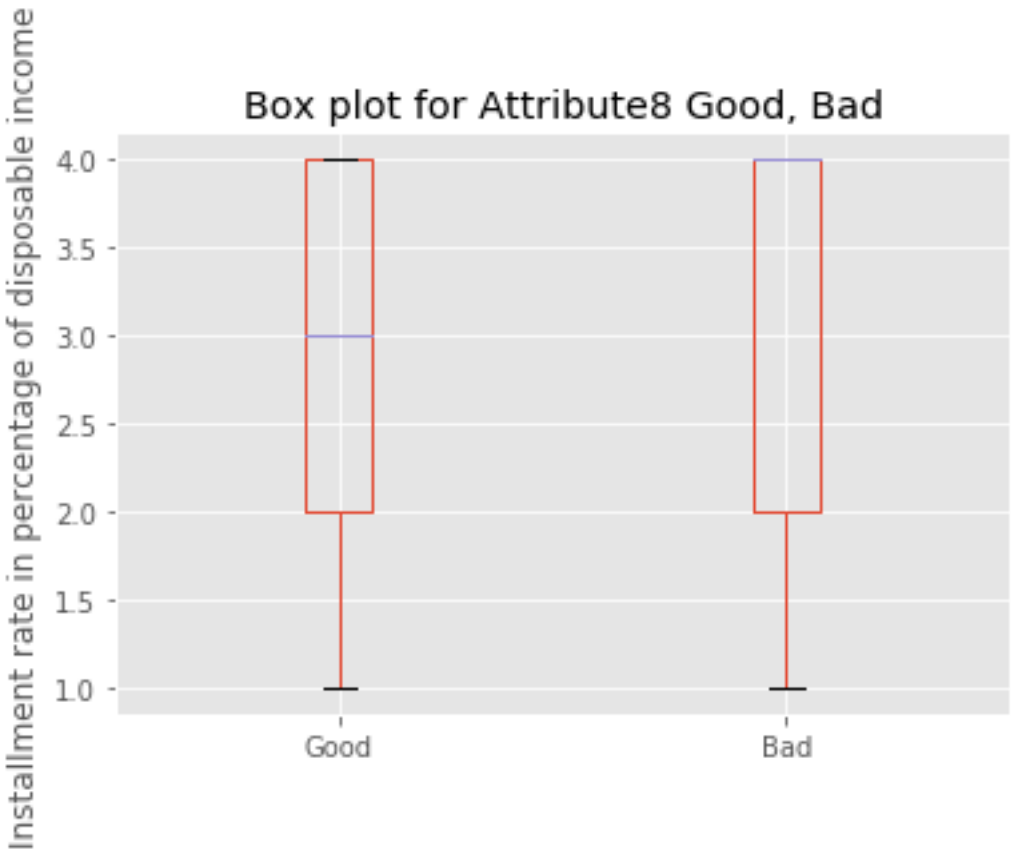
In [103]:
```
# histogram for numerical data of attribute 2
df = pd.DataFrame({'Good': Good['Attribute2'], 'Bad':Bad['Attribute2'] }, columns=['Go
df.plot.box()
plt.title("Box plot for Attribute2 Good, Bad")
plt.ylabel('Duration in month')
plt.show()
```



In [104]:
```
# histogram for numerical data of attribute 5
df = pd.DataFrame({'Good': Good['Attribute5'], 'Bad':Bad['Attribute5'] }, columns=['Go
df.plot.box()
plt.title("Box plot for Attribute5 Good, Bad")
plt.ylabel('Credit amount')
plt.show()
```

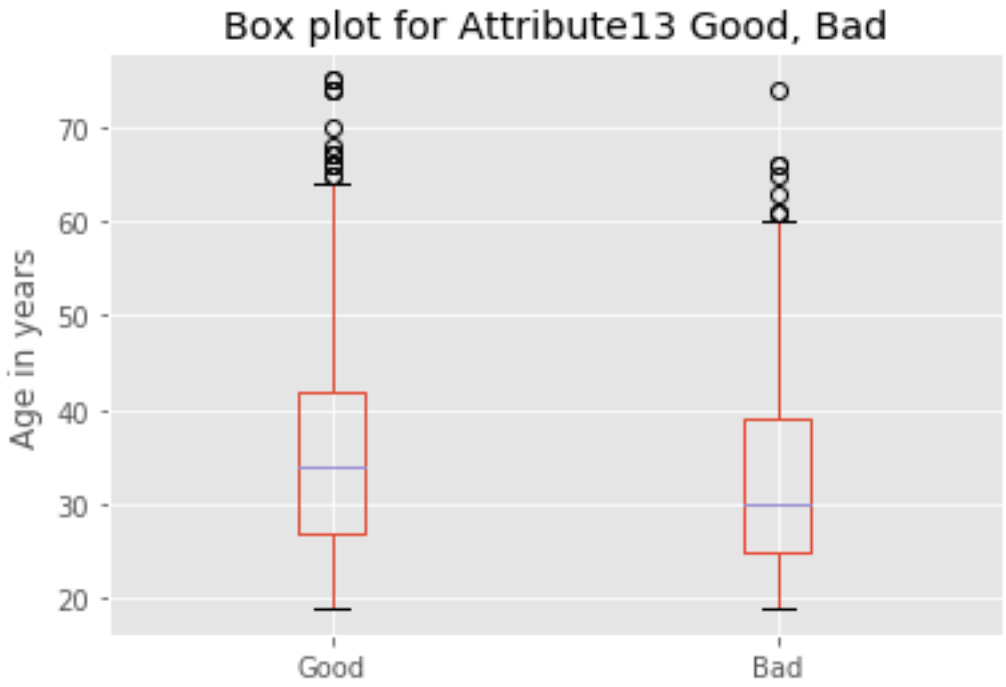


In [105]:
```
# histogram for numerical data of attribute 8
df = pd.DataFrame({'Good': Good['Attribute8'], 'Bad':Bad['Attribute8'] }, columns=['Go
df.plot.box()
plt.title("Box plot for Attribute8 Good, Bad")
plt.ylabel('Installment rate in percentage of disposable income')
plt.show()
```

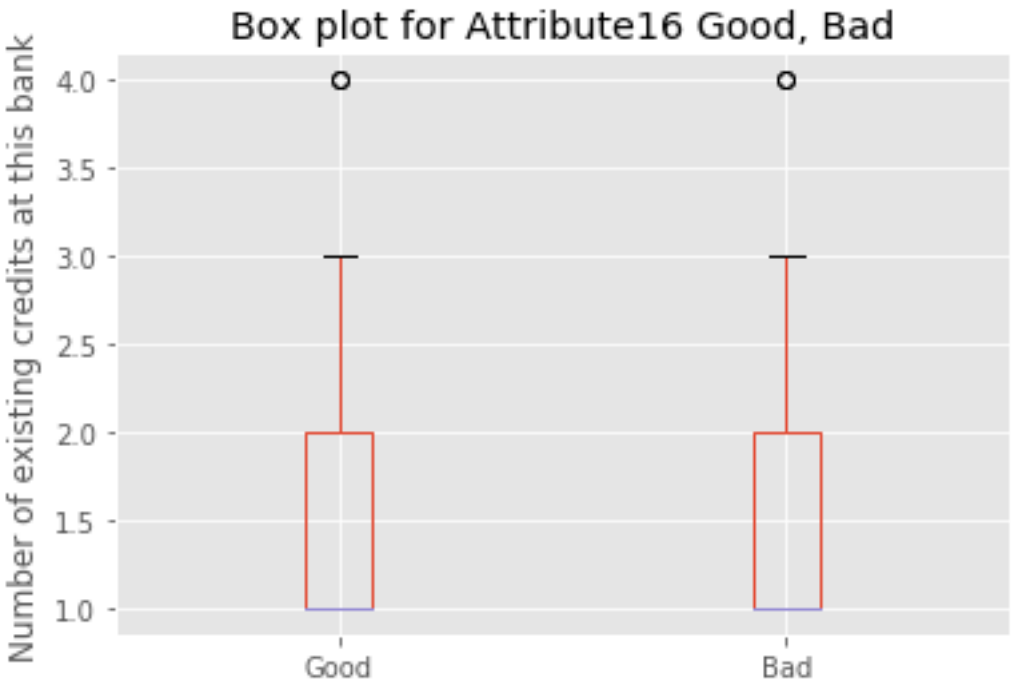


In [106]:
```
# histogram for numerical data of attribute 11
df = pd.DataFrame({'Good': Good['Attribute11'], 'Bad':Bad['Attribute11'] }, columns=['
df.plot.box()
plt.title("Box plot for Attribute11 Good, Bad")
plt.ylabel('Present residence since')
plt.show()
```

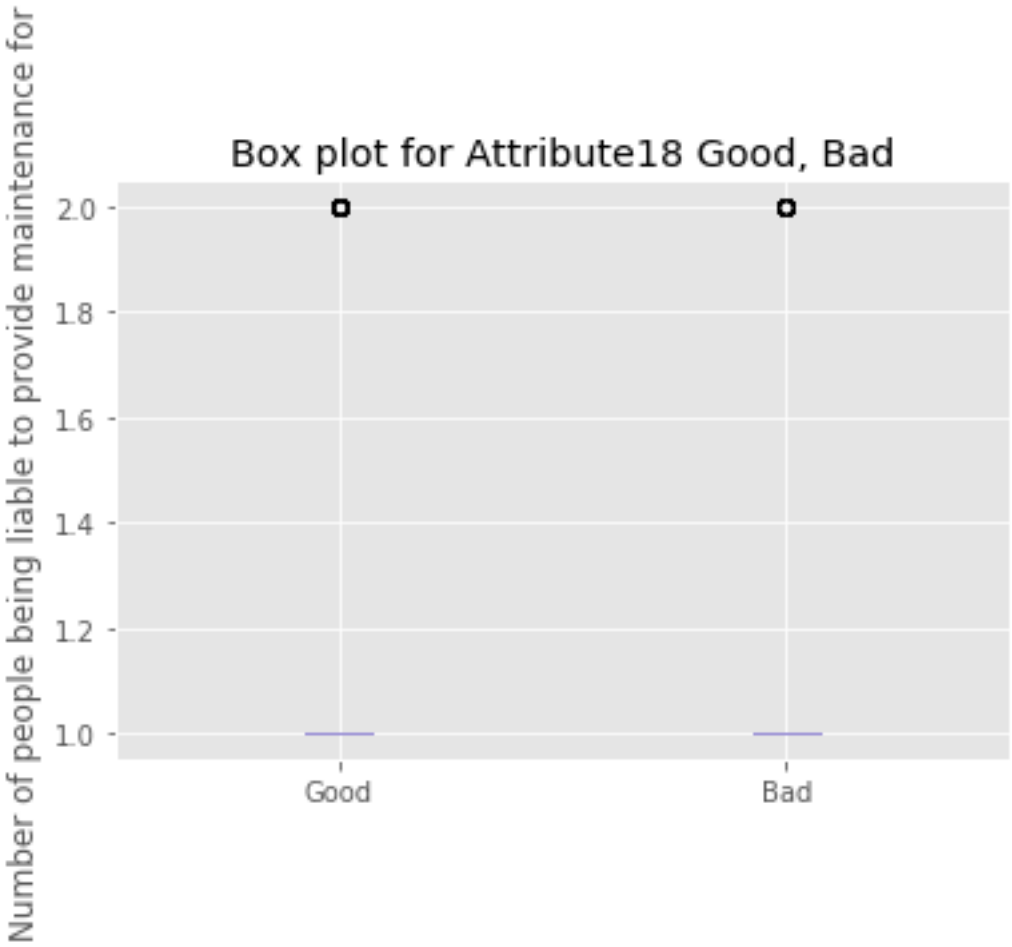
In [107]:
```
# histogram for numerical data of attribute 13
df = pd.DataFrame({'Good': Good['Attribute13'], 'Bad':Bad['Attribute13'] }, columns=['
df.plot.box()
plt.title("Box plot for Attribute13 Good, Bad")
plt.ylabel('Age in years')
plt.show()
```



In [108]:
```
# histogram for numerical data of attribute 16
df = pd.DataFrame({'Good': Good['Attribute16'], 'Bad':Bad['Attribute16'] }, columns=['
df.plot.box()
plt.title("Box plot for Attribute16 Good, Bad")
plt.ylabel('Number of existing credits at this bank')
plt.show()
```




```
In [109]: # histogram for numerical data of attribute 18
          df = pd.DataFrame({'Good': Good['Attribute18'], 'Bad':Bad['Attribute18'] }, columns=['
          df.plot.box()
          plt.title("Box plot for Attribute18 Good, Bad")
          plt.ylabel('Number of people being liable to provide maintenance for')
          plt.show()
```



1.2.3 Conclusions

From the categorical data plots, we can assume that features 1, 4, 12 and 17 will be most useful in the customers classification. This is because they present a satisfactory variance between the amount of "Good" and "Bad" customers that they represent. This means that it is safer to make decisions about whether a customer is "Good" or "Bad" based on them, because the possibility that we are right will be better than 50%. Under the same logic, from the numerical data plots, we assume that features 2 and 5 will be most useful in the customers classification. In both cases, we observe that the plots of all the other attributes data, whether they are numerical or categorical, don't really help us to decide whether someone is "Good" or "Bad" based on them, because the "Good" and the "Bad" plots resemble each other, sometimes a lot.