summary: which features to keep for model buiding

also see "variable table"

In [223…
```
## data: creditML
df = pd.read_csv('creditML.csv')
```

In [224…
```
df.head()
```

Out[224…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
0	20000	1	24	1	0	2	0	0
1	120000	1	26	2	0	2	3455	2000
2	90000	0	34	2	0	0	14948	5000
3	50000	0	37	1	0	0	28959	1000
4	50000	0	57	1	0	0	19146	679

import libraries

In [1]:
```
# import libraries

import pandas as pd
import numpy as np

from pandas import Series, DataFrame

import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
```

In [ ]:
```
# Set default matplot figure size
### NameError: name 'pylab' is not defined: pylab.rcParams['figure.figsize'] = (7.0, 7.
plt.rcParams['figure.figsize'] = (7.0, 7.0)

# Set text size
mpl.rcParams['font.size'] = 12
```

In [2]:
```
df = pd.read_csv('creditEDA.csv')
```

In [ ]:
```
## default was                          ## default is
not default    23364                     0    23364
default         6636                     1     6636

## SEX was                               ## SEX is
female    18112                          0    18112
male      11888                          1    11888
```

```
## EDUCATION --4 dummies                          ## education
university         14030                          3    14030
graduate school    10585                          0    10585
high school         4917                          1     4917
other                468                          2      468
```

In [3]:
```
df.columns
```

Out[3]:
```
Index(['credit', 'SEX', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default', 'education',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_other',
       'EDUCATION_university'],
      dtype='object')
```

In [4]:
```
df.head()
```

Out[4]:

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	PAY_AMT3	PAY_AN
0	20000	0	1	24	2	2	-1	-1	-2	-2	...	0	
1	120000	0	2	26	-1	2	0	0	0	2	...	1000	1
2	90000	0	2	34	0	0	0	0	0	0	...	1000	1
3	50000	0	1	37	0	0	0	0	0	0	...	1200	1
4	50000	1	1	57	-1	0	-1	0	0	0	...	10000	9

5 rows × 28 columns

In [5]:
```
df.dtypes
```

Out[5]:
```
credit                       int64
SEX                          int64
MARRIAGE                     int64
AGE                          int64
PAY_0                        int64
PAY_2                        int64
PAY_3                        int64
PAY_4                        int64
PAY_5                        int64
PAY_6                        int64
BILL_AMT1                    int64
BILL_AMT2                    int64
BILL_AMT3                    int64
BILL_AMT4                    int64
BILL_AMT5                    int64
BILL_AMT6                    int64
PAY_AMT1                     int64
PAY_AMT2                     int64
PAY_AMT3                     int64
PAY_AMT4                     int64
PAY_AMT5                     int64
PAY_AMT6                     int64
default                      int64
education                    int64
EDUCATION_graduate school    int64
EDUCATION_high school        int64
```

```
EDUCATION_other             int64
EDUCATION_university        int64
dtype: object
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   credit                     30000 non-null  int64
 1   SEX                        30000 non-null  int64
 2   MARRIAGE                   30000 non-null  int64
 3   AGE                        30000 non-null  int64
 4   PAY_0                      30000 non-null  int64
 5   PAY_2                      30000 non-null  int64
 6   PAY_3                      30000 non-null  int64
 7   PAY_4                      30000 non-null  int64
 8   PAY_5                      30000 non-null  int64
 9   PAY_6                      30000 non-null  int64
 10  BILL_AMT1                  30000 non-null  int64
 11  BILL_AMT2                  30000 non-null  int64
 12  BILL_AMT3                  30000 non-null  int64
 13  BILL_AMT4                  30000 non-null  int64
 14  BILL_AMT5                  30000 non-null  int64
 15  BILL_AMT6                  30000 non-null  int64
 16  PAY_AMT1                   30000 non-null  int64
 17  PAY_AMT2                   30000 non-null  int64
 18  PAY_AMT3                   30000 non-null  int64
 19  PAY_AMT4                   30000 non-null  int64
 20  PAY_AMT5                   30000 non-null  int64
 21  PAY_AMT6                   30000 non-null  int64
 22  default                    30000 non-null  int64
 23  education                  30000 non-null  int64
 24  EDUCATION_graduate school  30000 non-null  int64
 25  EDUCATION_high school      30000 non-null  int64
 26  EDUCATION_other            30000 non-null  int64
 27  EDUCATION_university       30000 non-null  int64
dtypes: int64(28)
memory usage: 6.4 MB
```

In [7]:
```
df.describe()
```

Out[7]:

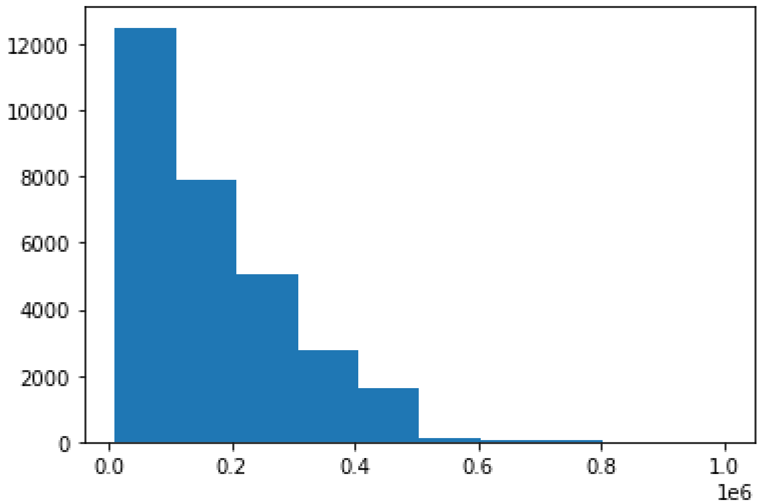
	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PA
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000
mean	167484.322667	0.396267	1.551867	35.485500	-0.016700	-0.133767	-0.166
std	129747.661567	0.489129	0.521970	9.217904	1.123802	1.197186	1.196
min	10000.000000	0.000000	0.000000	21.000000	-2.000000	-2.000000	-2.000
25%	50000.000000	0.000000	1.000000	28.000000	-1.000000	-1.000000	-1.000
50%	140000.000000	0.000000	2.000000	34.000000	0.000000	0.000000	0.000
75%	240000.000000	1.000000	2.000000	41.000000	0.000000	0.000000	0.000
max	1000000.000000	1.000000	3.000000	79.000000	8.000000	8.000000	8.000

8 rows × 28 columns

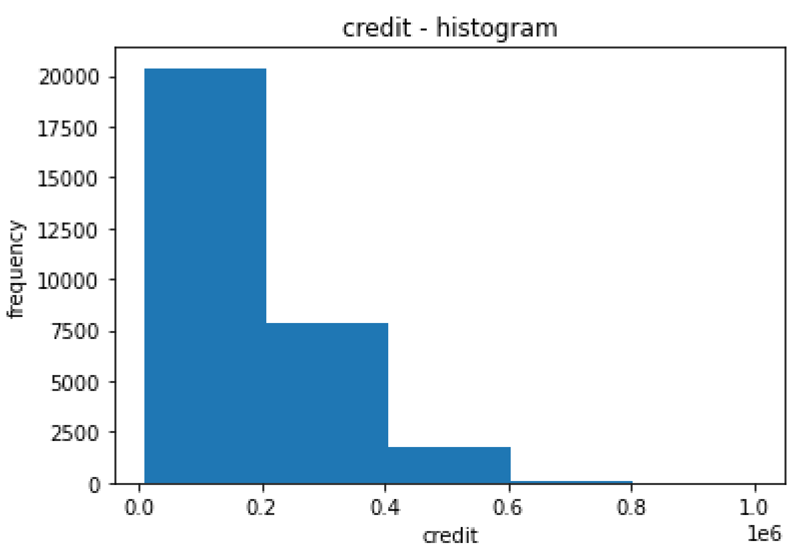
EDA

histogram-- equal numbers of bin

In [8]:
```
plt.hist(df['credit'])
plt.show()
```



In [9]:
```
plt.hist(df['credit'], bins=5)

plt.title('credit - histogram')
plt.xlabel('credit')
plt.ylabel('frequency')

plt.show()
```

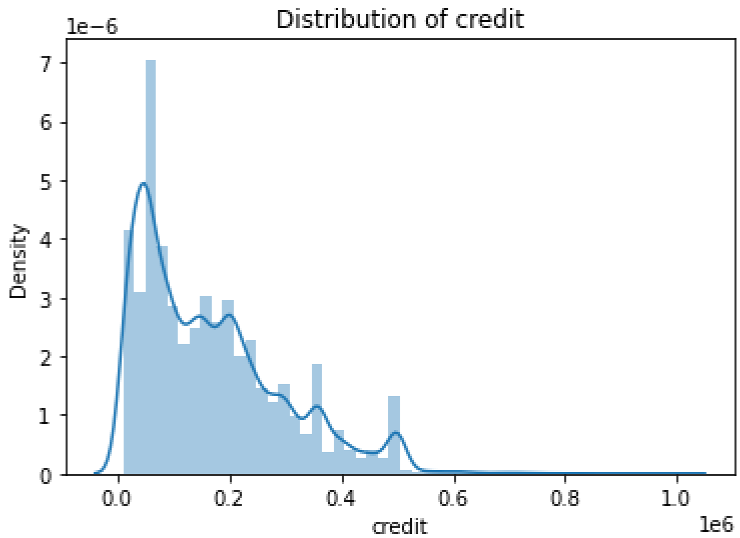


In [77]:
```
df = df['credit'].dropna()
# Drop missing values for the records in which credit is missing
```

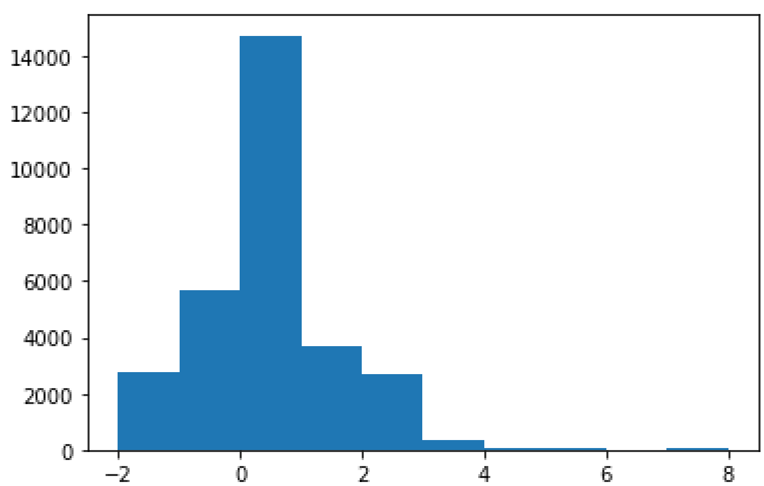
In [78]:
```
df_dist = sns.distplot(df)
df_dist.set_title("Distribution of credit")
```

```
C:\Users\Dongmei\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please a
dapt your code to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

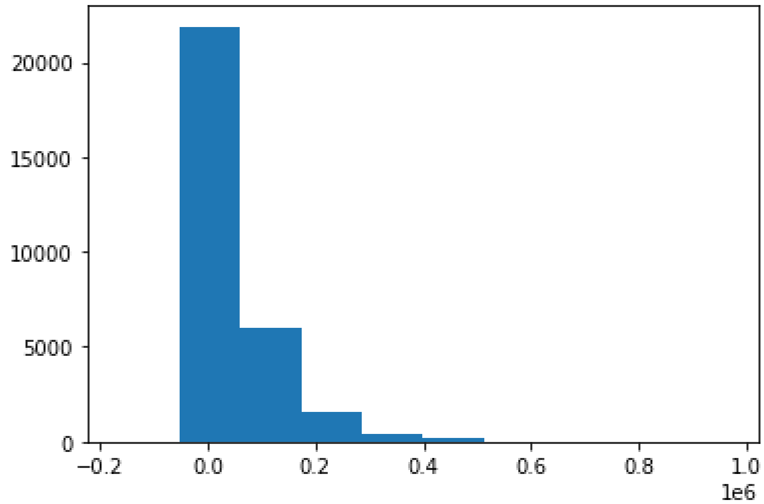
Out[78]: Text(0.5, 1.0, 'Distribution of credit')



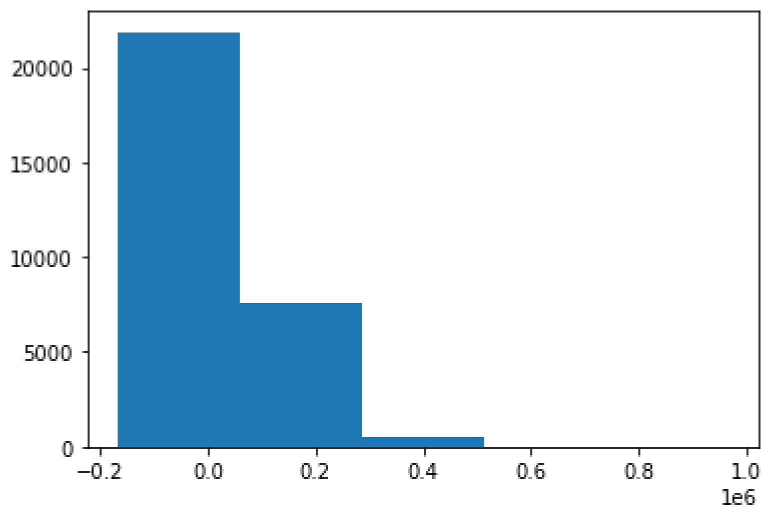
In [10]:
```python
plt.hist(df['PAY_0'])
plt.show()

##-1: Paid in full;
## 0: The use of revolving credit;
## 1 = payment delay for one month;
## 2 = payment delay for two months;
## . . .
## 8 = payment delay for eight months;
## 9 = payment delay for nine months and above.
```



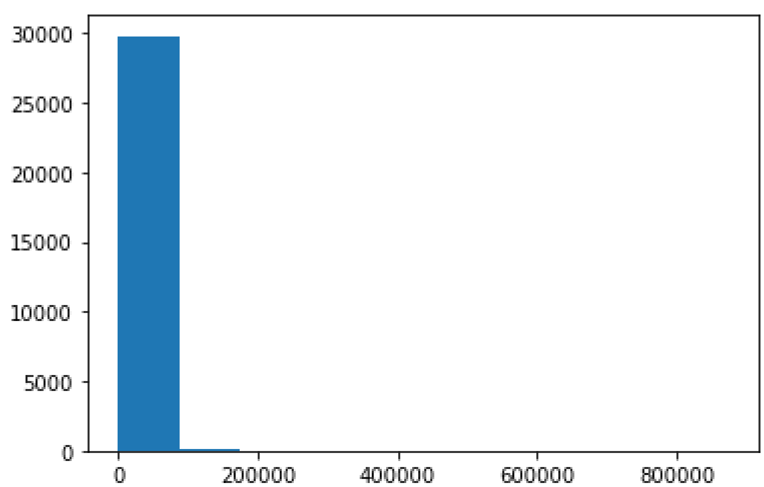
In [12]:
```python
plt.hist(df['BILL_AMT1'])
plt.show()
```



In [11]:
```
plt.hist(df['BILL_AMT1'], bins=5)
plt.show()
```

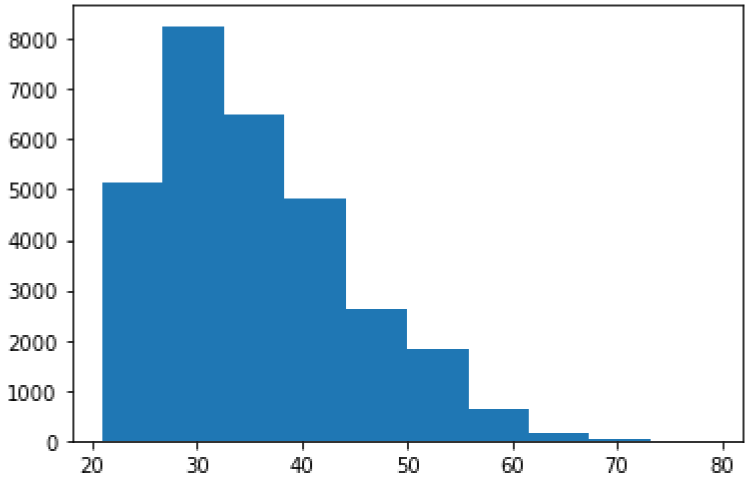


In [13]:
```
plt.hist(df['PAY_AMT1'])
plt.show()
```

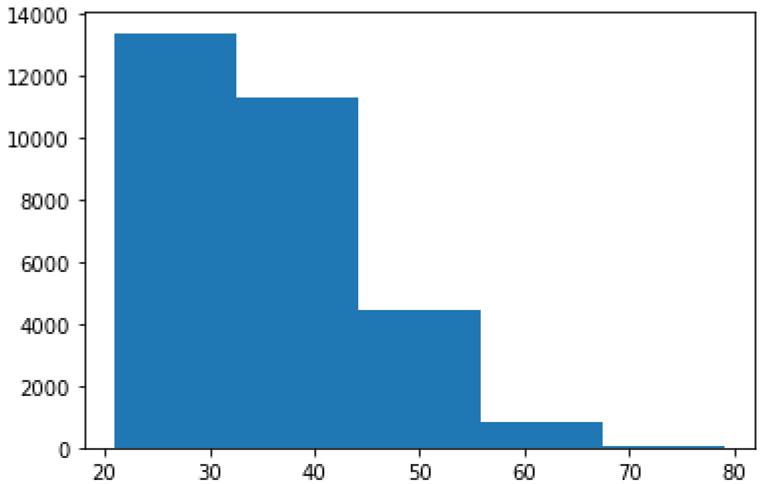


In [ ]:
```
plt.hist(df['PAY_AMT1'], bins=5)
plt.show()
```

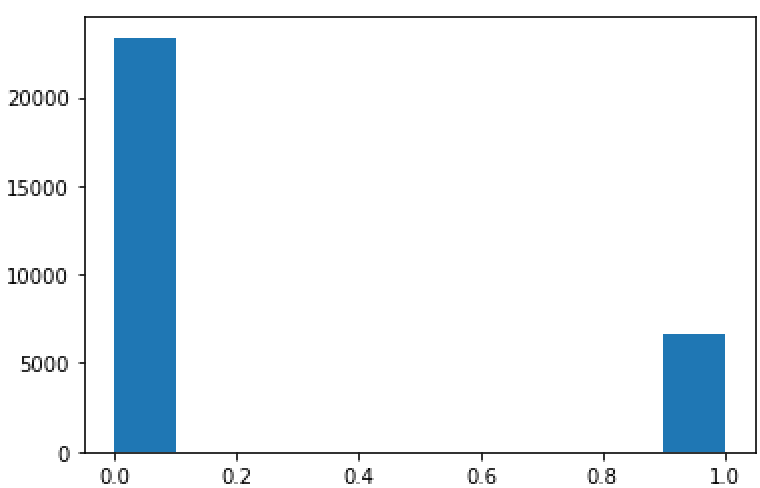
In [14]:
```
plt.hist(df['AGE'])
plt.show()
```



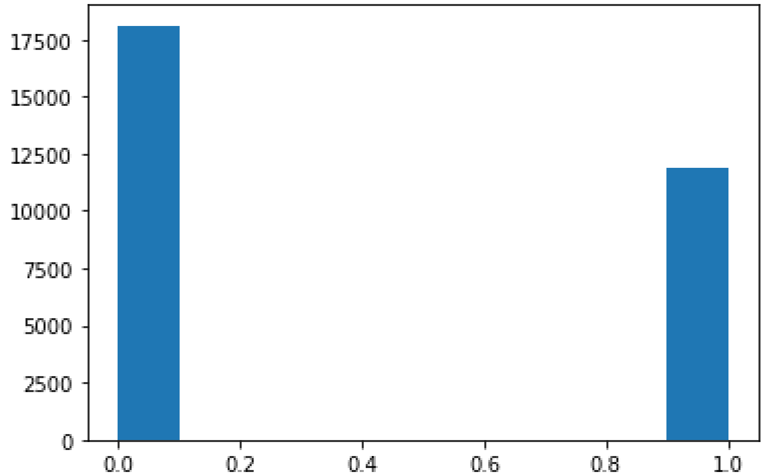
In [15]:
```
plt.hist(df['AGE'], bins=5)
plt.show()
```



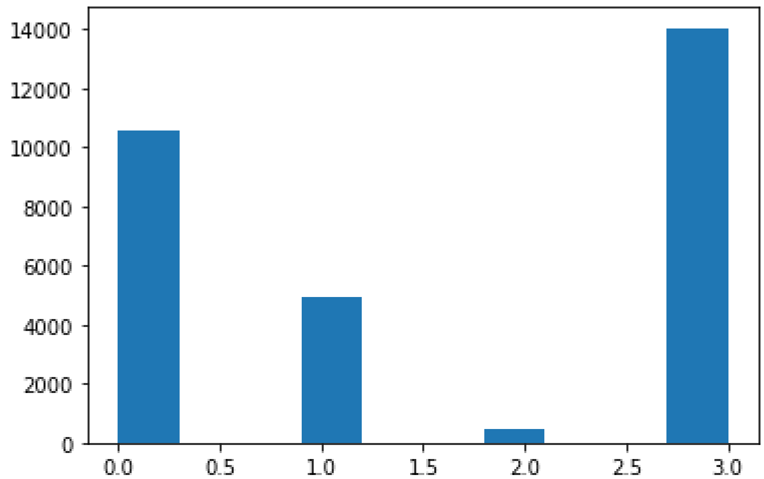
In [16]:
```
plt.hist(df['default'])
plt.show()
```



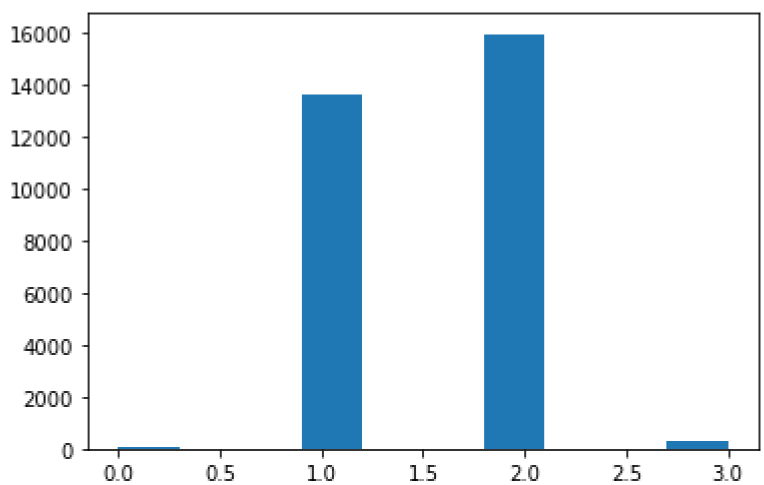
In [17]:
```
plt.hist(df['SEX'])
plt.show()
```



In [18]:
```
plt.hist(df['education'])   ###  0 graduate school 1 high school 2 other 3 univeristy
plt.show()
```



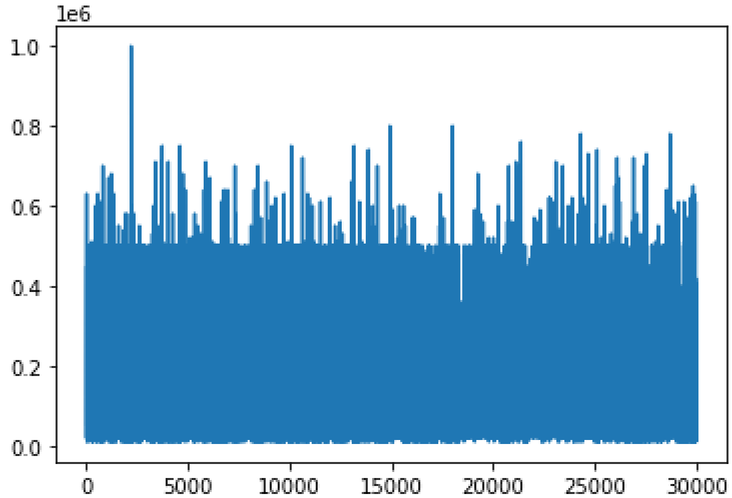
In [19]:
```
plt.hist(df['MARRIAGE'])   ###  0 other 1  married, 2 single 3 divorce
plt.show()
```



line plot

tcredit by features: sex,EDUCATION, default, AGE, marriage, PAY_0, BILL_AMT1, PAY_AMT1

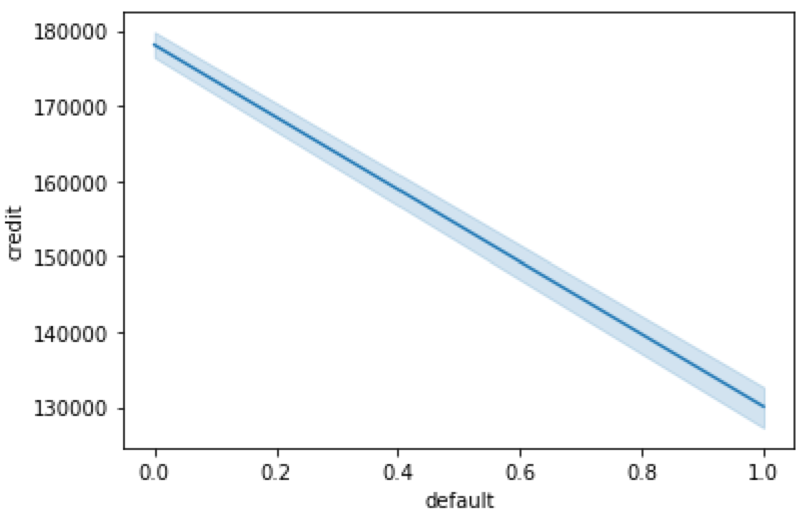
In [20]:
```
plt.plot(df['credit'])
plt.show()
```



In [ ]:
```
## https://seaborn.pydata.org/generated/seaborn.lineplot.html

### sns.lineplot(data=may_flights, x="year", y="passengers")
### sns.lineplot(data=flights, x="year", y="passengers")
```

In [21]:
```
sns.lineplot(data=df, x="default", y="credit")
```

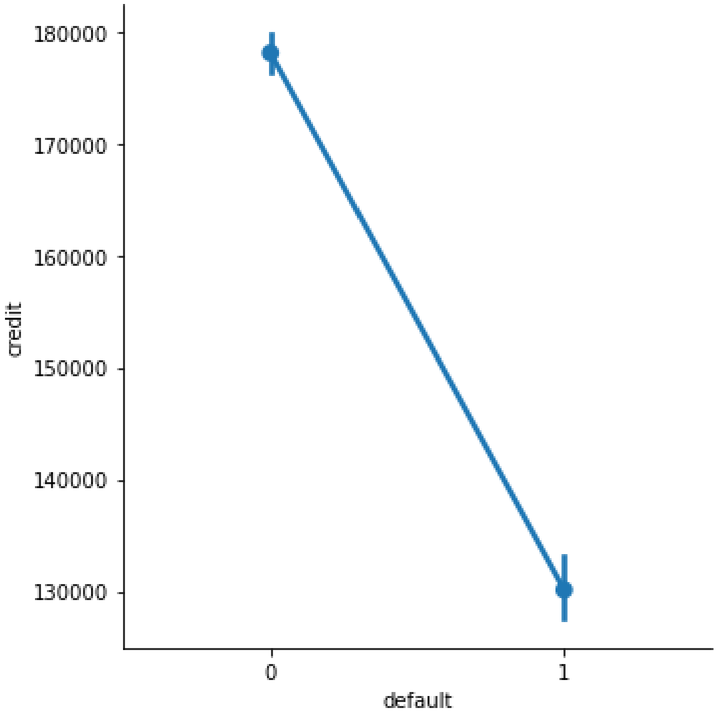
Out[21]: <AxesSubplot:xlabel='default', ylabel='credit'>



In [22]:
```
##
sns.factorplot(x='default', y='credit', data=df)
```

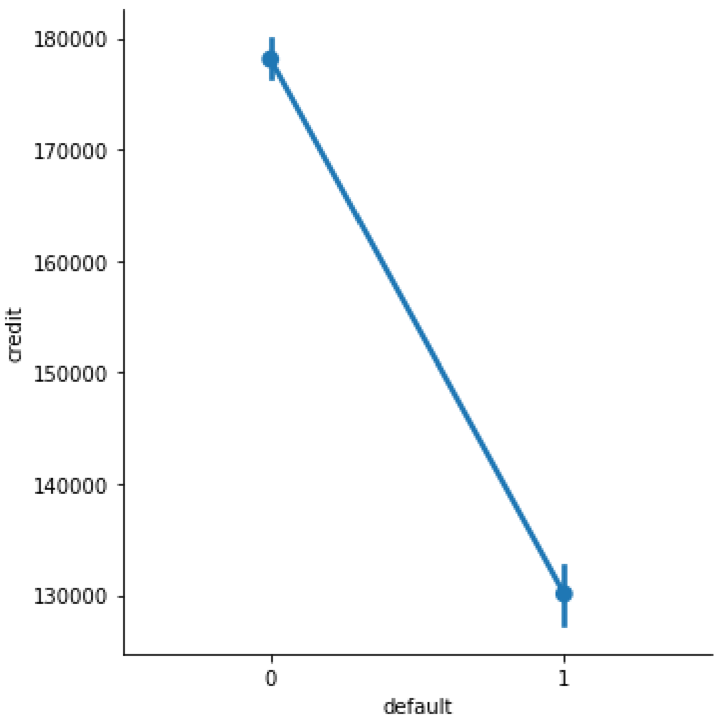
```
C:\Users\Dongmei\anaconda3\lib\site-packages\seaborn\categorical.py:3704: UserWarning: T
he `factorplot` function has been renamed to `catplot`. The original name will be remove
d in a future release. Please update your code. Note that the default `kind` in `factorp
lot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
```

Out[22]: <seaborn.axisgrid.FacetGrid at 0x123071ff550>



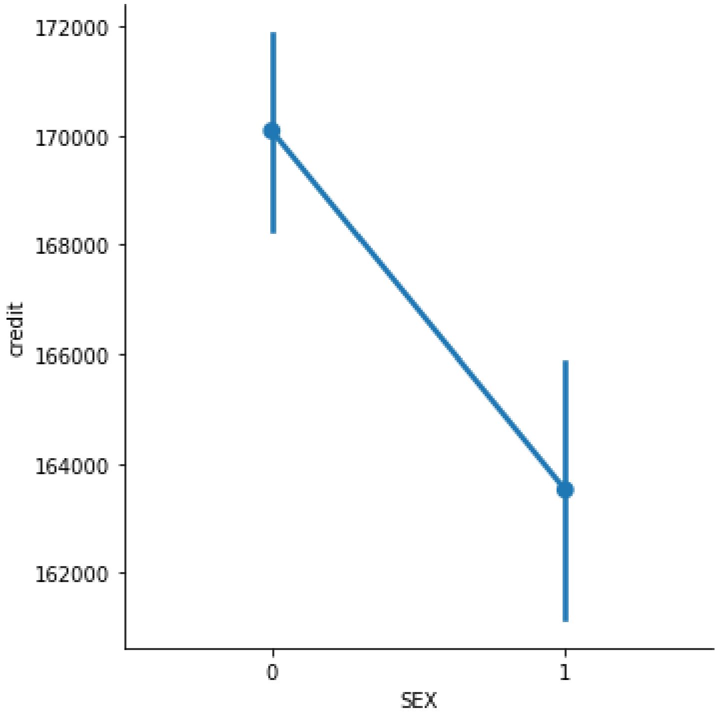
In [23]:
```
sns.catplot(x='default', y='credit', kind='point', data=df)
```

Out[23]: <seaborn.axisgrid.FacetGrid at 0x123071655b0>



In [104…
```
sns.catplot(x='SEX', y='credit', kind='point', data=df)
```

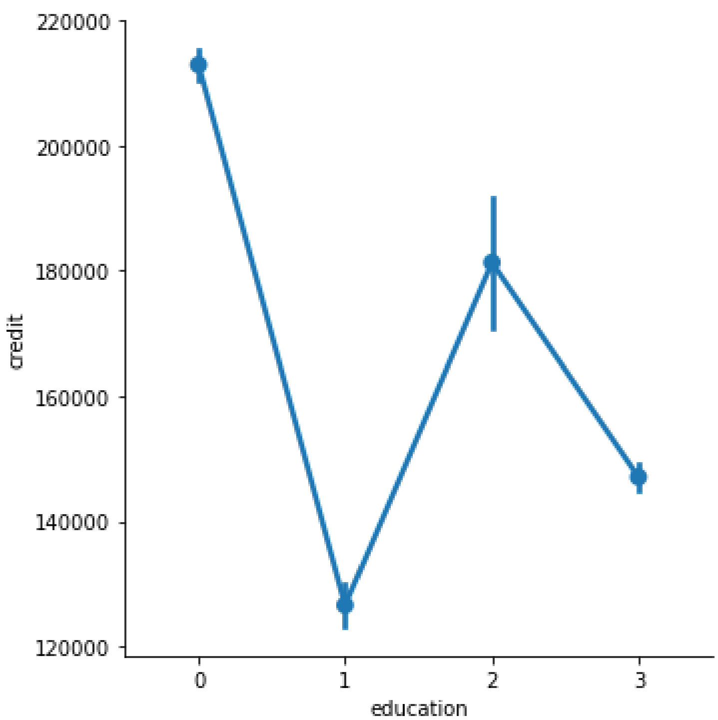
Out[104… <seaborn.axisgrid.FacetGrid at 0x12307545c40>



In [105…

```
sns.catplot(x='education', y='credit', kind='point', data=df)
```

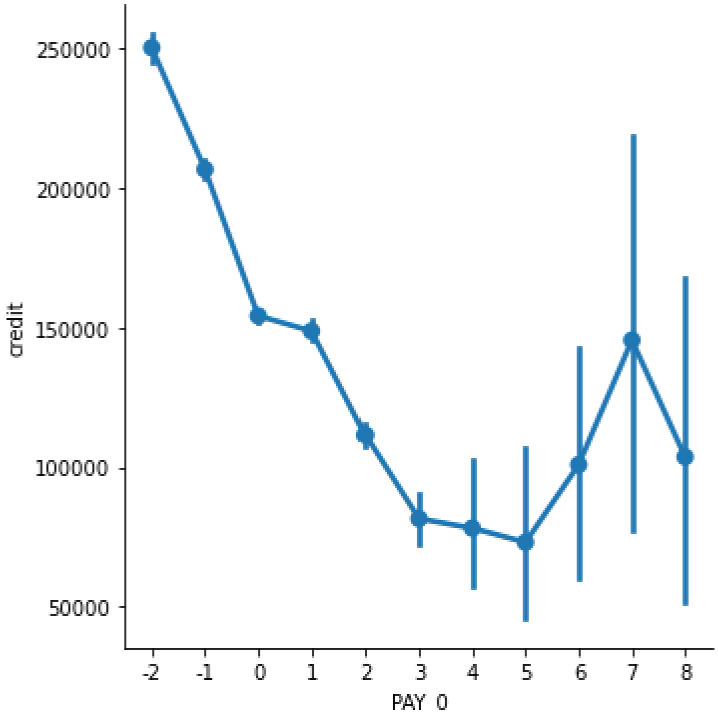
Out[105… <seaborn.axisgrid.FacetGrid at 0x123078aba30>



In [106…

```
sns.catplot(x='PAY_0', y='credit', kind='point', data=df)
```

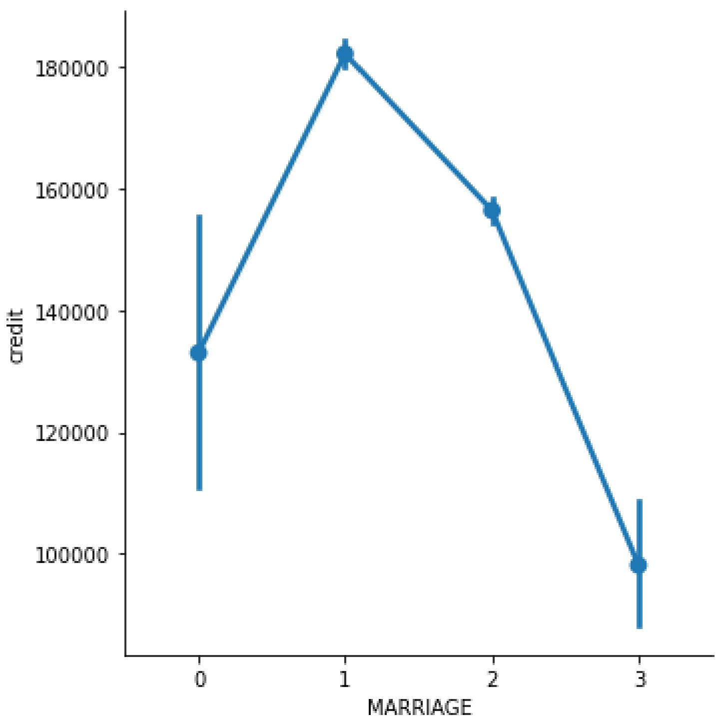
Out[106… <seaborn.axisgrid.FacetGrid at 0x1230c63cdf0>



In [107…

```
sns.catplot(x='MARRIAGE', y='credit', kind='point', data=df)

##  1 = married; 2 = single; 3 = divorce; 0=others).
```

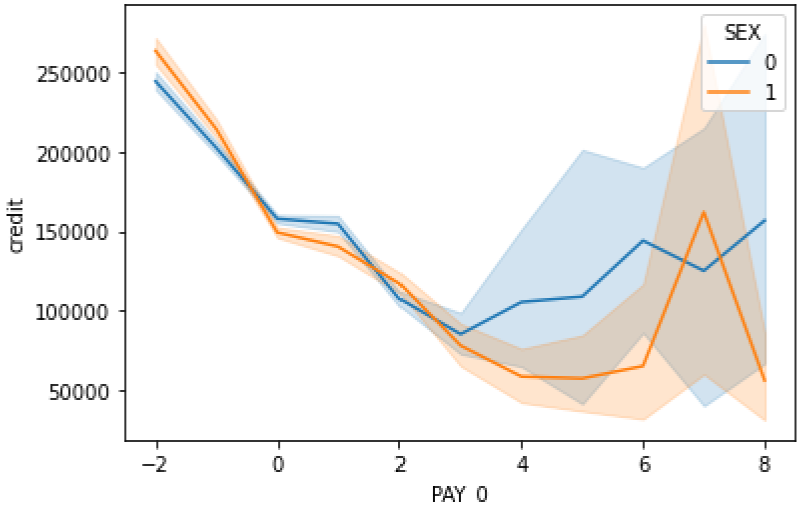
Out[107… <seaborn.axisgrid.FacetGrid at 0x1230c113be0>



In [24]:

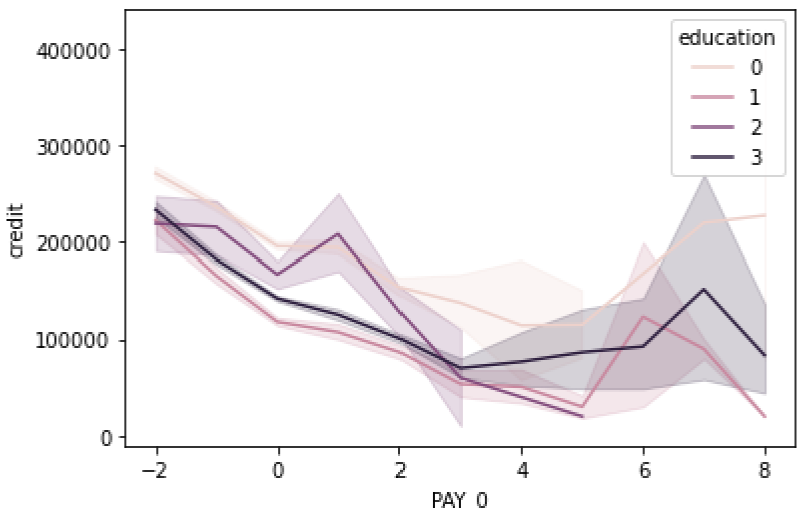
```
sns.lineplot(data=df, x="PAY_0", y="credit", hue="SEX")
```

Out[24]: <AxesSubplot:xlabel='PAY_0', ylabel='credit'>



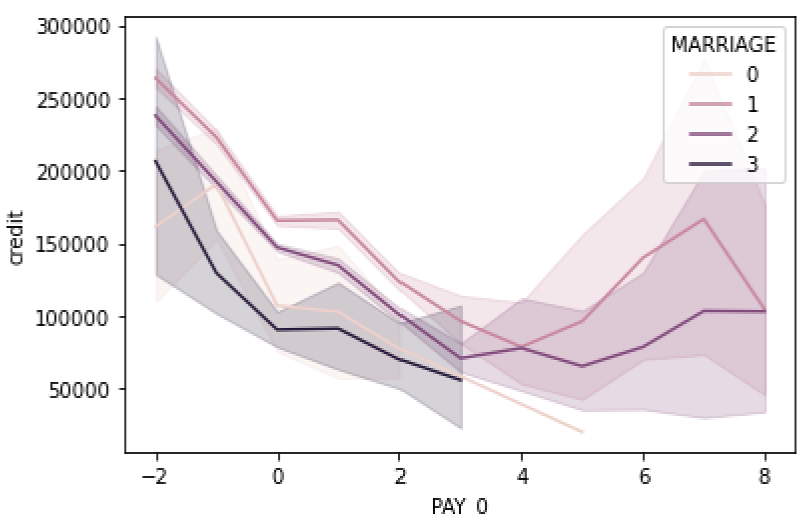
In [25]:
```
sns.lineplot(data=df, x="PAY_0", y="credit", hue="education")
```

Out[25]: <AxesSubplot:xlabel='PAY_0', ylabel='credit'>



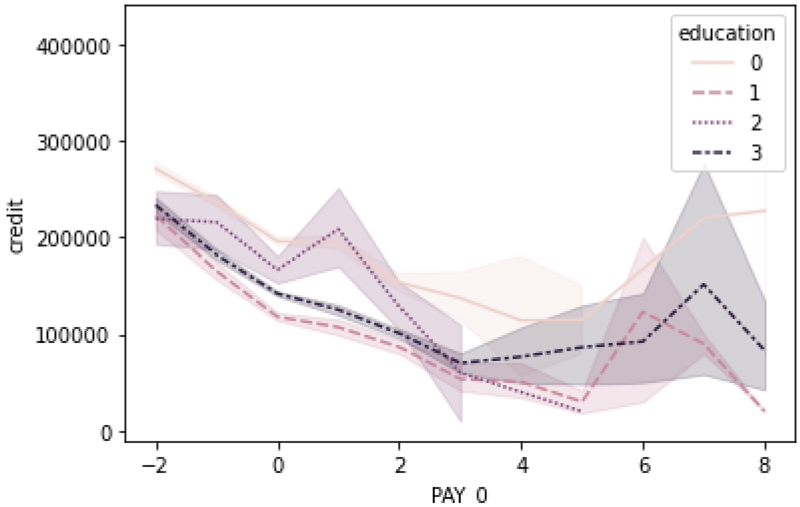
In [26]:
```
sns.lineplot(data=df, x="PAY_0", y="credit", hue="MARRIAGE")
```

Out[26]: <AxesSubplot:xlabel='PAY_0', ylabel='credit'>



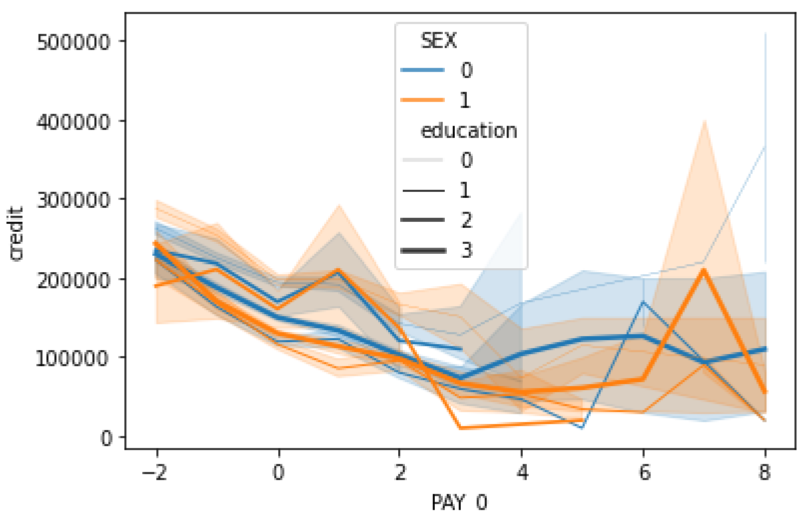
In [27]:
```
sns.lineplot(data=df, x="PAY_0", y="credit", hue="education", style="education")
```

Out[27]: <AxesSubplot:xlabel='PAY_0', ylabel='credit'>



In [28]:
```
sns.lineplot(
    data=df, x="PAY_0", y="credit",
    size="education", hue="SEX",
    sizes=(.25, 2.5)
)
```

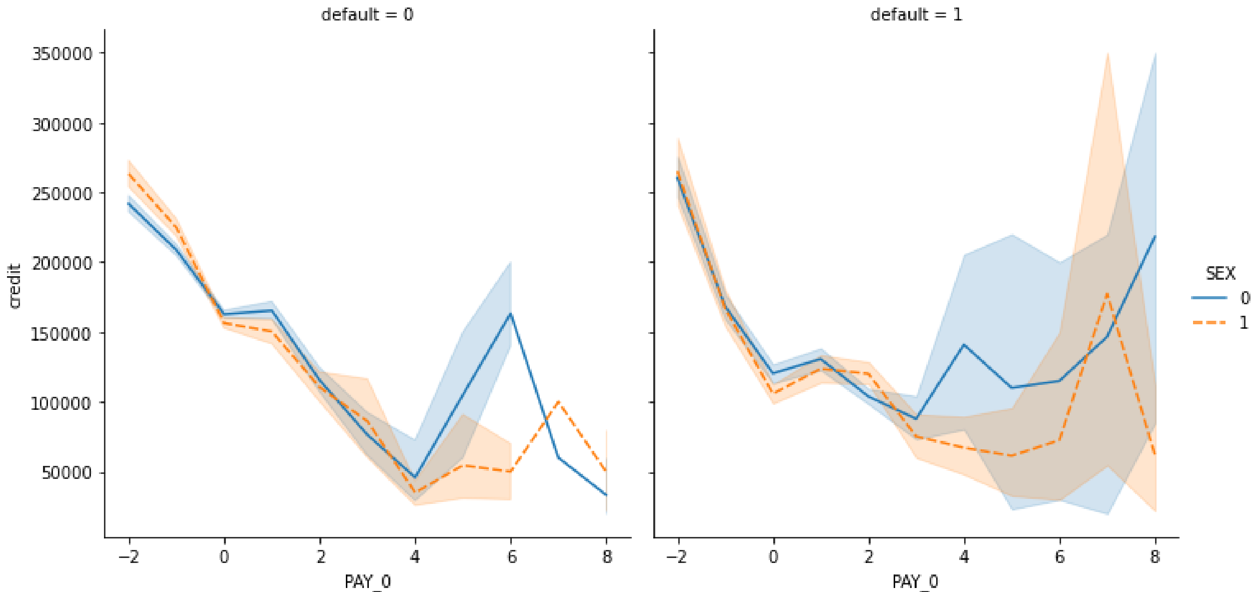
Out[28]: <AxesSubplot:xlabel='PAY_0', ylabel='credit'>



In [ ]:
```
x, y = np.random.normal(size=(2, 5000)).cumsum(axis=1)
sns.lineplot(x=x, y=y, sort=False, lw=1)
```

In [29]:
```
sns.relplot(
    data=df, x="PAY_0", y="credit",
    col="default", hue="SEX", style="SEX",
    kind="line"
)
```

Out[29]: <seaborn.axisgrid.FacetGrid at 0x12306f2d9a0>

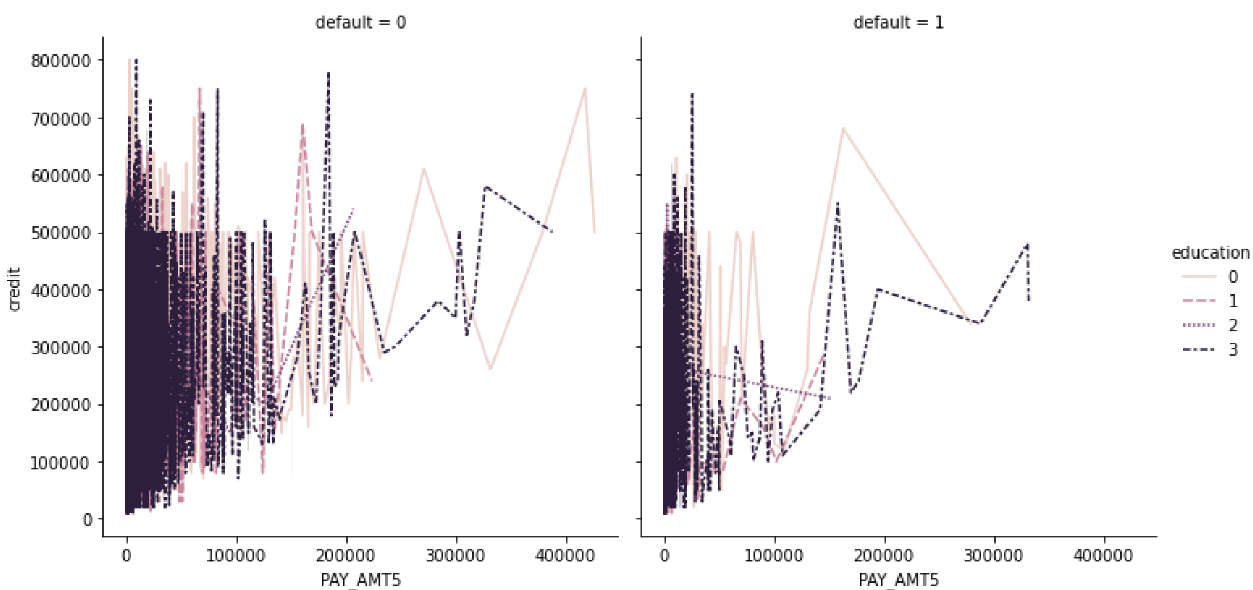


In [30]:
```
df.columns
```

Out[30]:
```
Index(['credit', 'SEX', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default', 'education',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_other',
       'EDUCATION_university'],
      dtype='object')
```

In [32]:
```
sns.relplot(
    data=df, x="PAY_AMT5", y="credit",
    col="default", hue="education", style="education",
    kind="line"
)
```

Out[32]:
```
<seaborn.axisgrid.FacetGrid at 0x12307c47520>
```



In [ ]:
```
#### the above : https://seaborn.pydata.org/generated/seaborn.lineplot.html
```

https://github.com/hoffm386/eda-with-categorical-variables

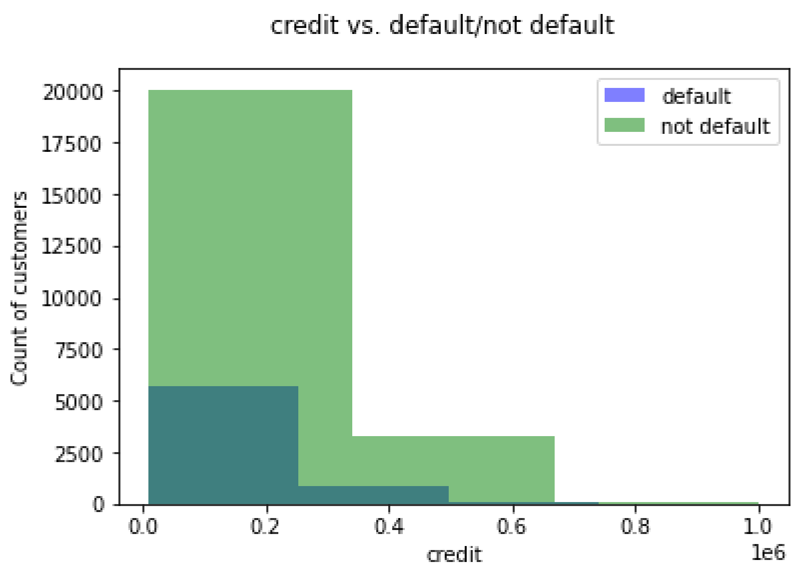
EDA with Categorical Variables

In [33]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
from matplotlib.patches import Patch
from matplotlib.lines import Line2D
import seaborn as sns
```

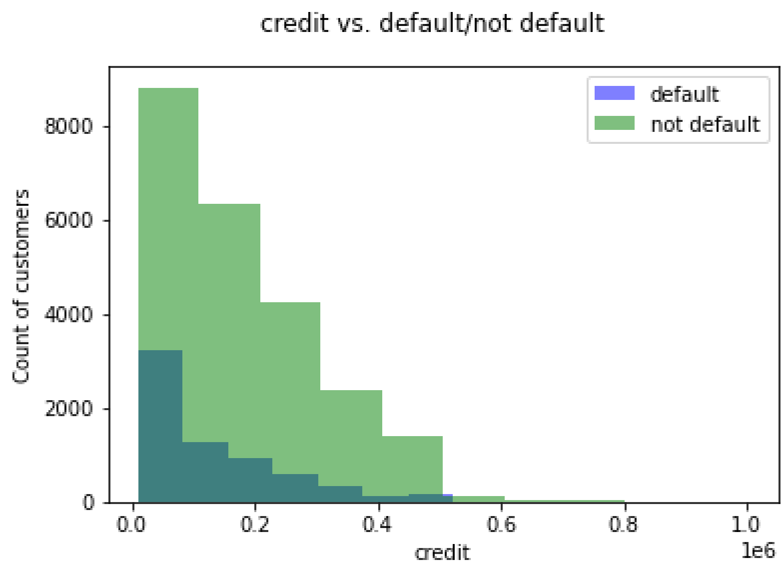
Numeric vs. Categorical EDA

Multiple Histograms---- by default. also by sex, educaiton, marriage?

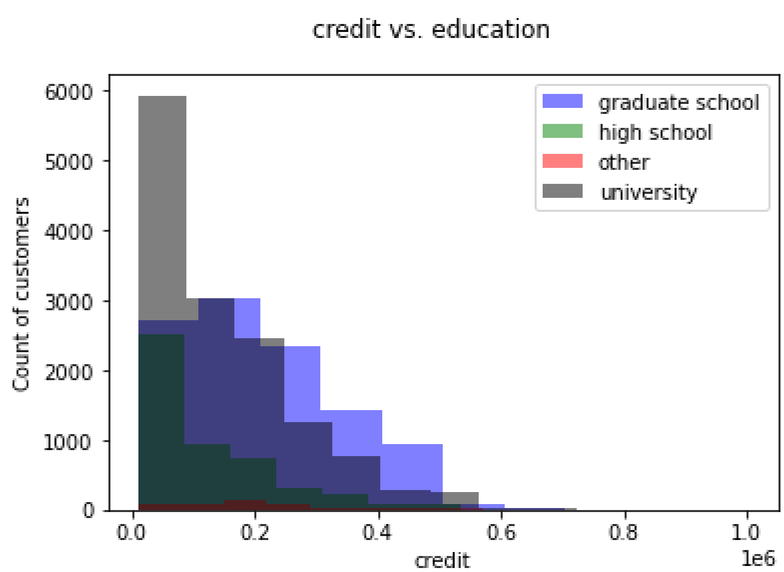
In [34]:
```
fig, ax = plt.subplots()

ax.hist(df[df["default"]==1]["credit"], bins=3, alpha=0.5, color="blue", label="default
ax.hist(df[df["default"]==0]["credit"], bins=3, alpha=0.5, color="green", label="not de

ax.set_xlabel("credit")
ax.set_ylabel("Count of customers")

fig.suptitle("credit vs. default/not default")

ax.legend();
```



In [35]:
```
fig, ax = plt.subplots()

ax.hist(df[df["default"]==1]["credit"], alpha=0.5, color="blue", label="default")
ax.hist(df[df["default"]==0]["credit"], alpha=0.5, color="green", label="not default")

ax.set_xlabel("credit")
ax.set_ylabel("Count of customers")

fig.suptitle("credit vs. default/not default")

ax.legend();
```

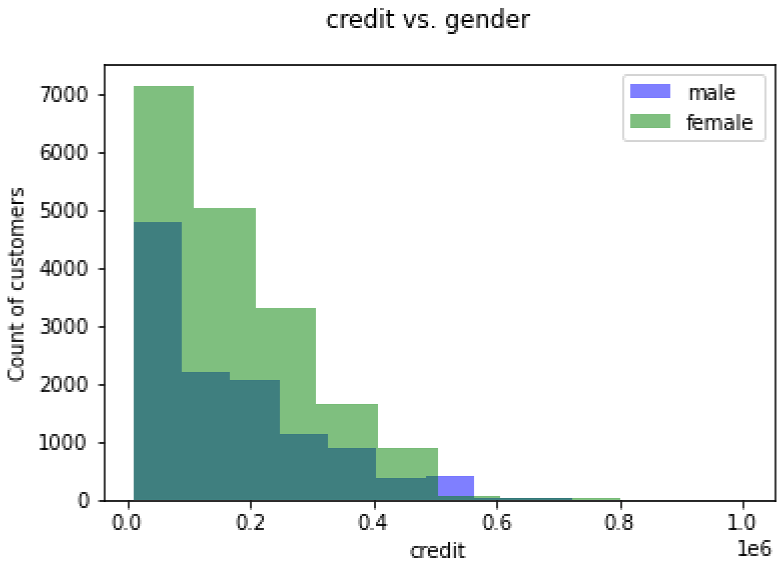


In [36]:
```
fig, ax = plt.subplots()

ax.hist(df[df["education"]==0]["credit"], alpha=0.5, color="blue", label="graduate scho
ax.hist(df[df["education"]==1]["credit"], alpha=0.5, color="green", label="high school"
ax.hist(df[df["education"]==2]["credit"], alpha=0.5, color="red", label="other")
ax.hist(df[df["education"]==3]["credit"], alpha=0.5, color="black", label="university")

ax.set_xlabel("credit")
ax.set_ylabel("Count of customers")

fig.suptitle("credit vs. education")

ax.legend();
```

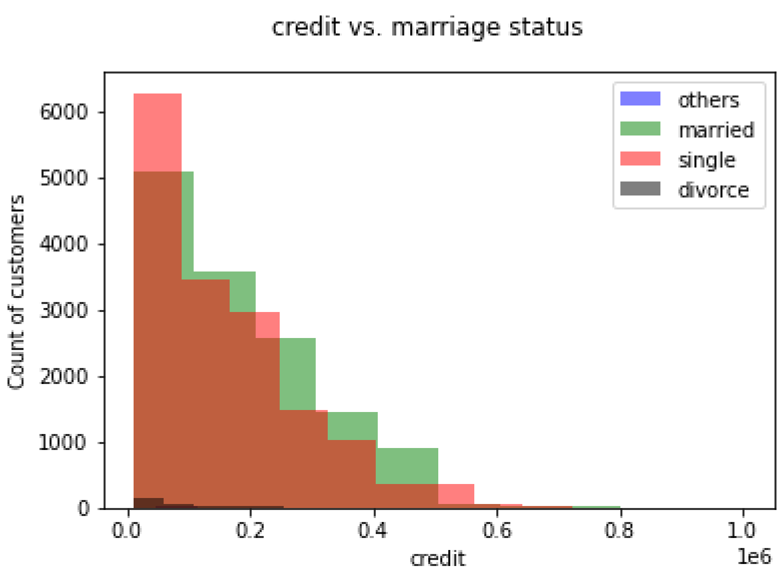


In [57]:
```
fig, ax = plt.subplots()

ax.hist(df[df["SEX"]==1]["credit"], alpha=0.5, color="blue", label="male")
ax.hist(df[df["SEX"]==0]["credit"], alpha=0.5, color="green", label="female")

ax.set_xlabel("credit")
ax.set_ylabel("Count of customers")
```

```
fig.suptitle("credit vs. gender")

ax.legend();
```



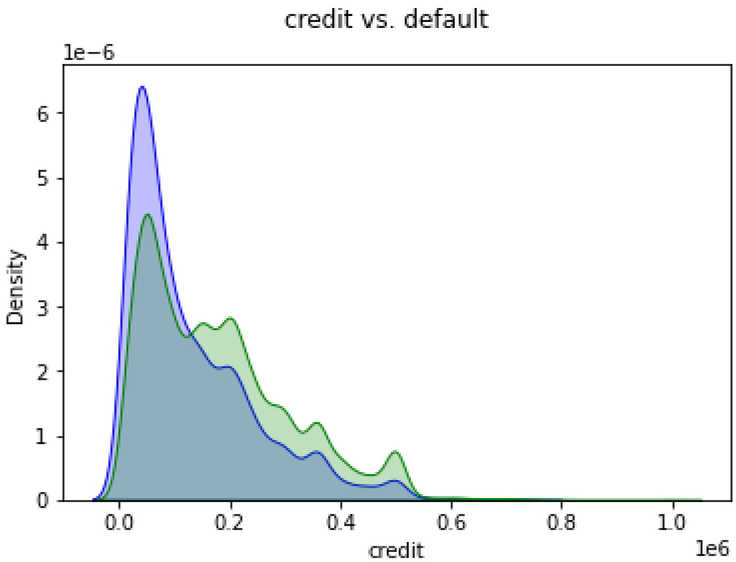
In [58]:
```
fig, ax = plt.subplots()

ax.hist(df[df["MARRIAGE"]==0]["credit"], alpha=0.5, color="blue", label="others")
ax.hist(df[df["MARRIAGE"]==1]["credit"], alpha=0.5, color="green", label="married")
ax.hist(df[df["MARRIAGE"]==2]["credit"], alpha=0.5, color="red", label="single")
ax.hist(df[df["MARRIAGE"]==3]["credit"], alpha=0.5, color="black", label="divorce")

ax.set_xlabel("credit")
ax.set_ylabel("Count of customers")

fig.suptitle("credit vs. marriage status")

ax.legend();
```



Multiple Density Estimate Plots

In [37]:
```
fig, ax = plt.subplots()
```

```
sns.kdeplot(df[df["default"]==1]["credit"], shade=True, color="blue", label="default",
sns.kdeplot(df[df["default"]==0]["credit"], shade=True, color="green", label="not defau

ax.set_xlabel("credit")
ax.set_ylabel("Density")

fig.suptitle("credit vs. default");

## where is my label?
```



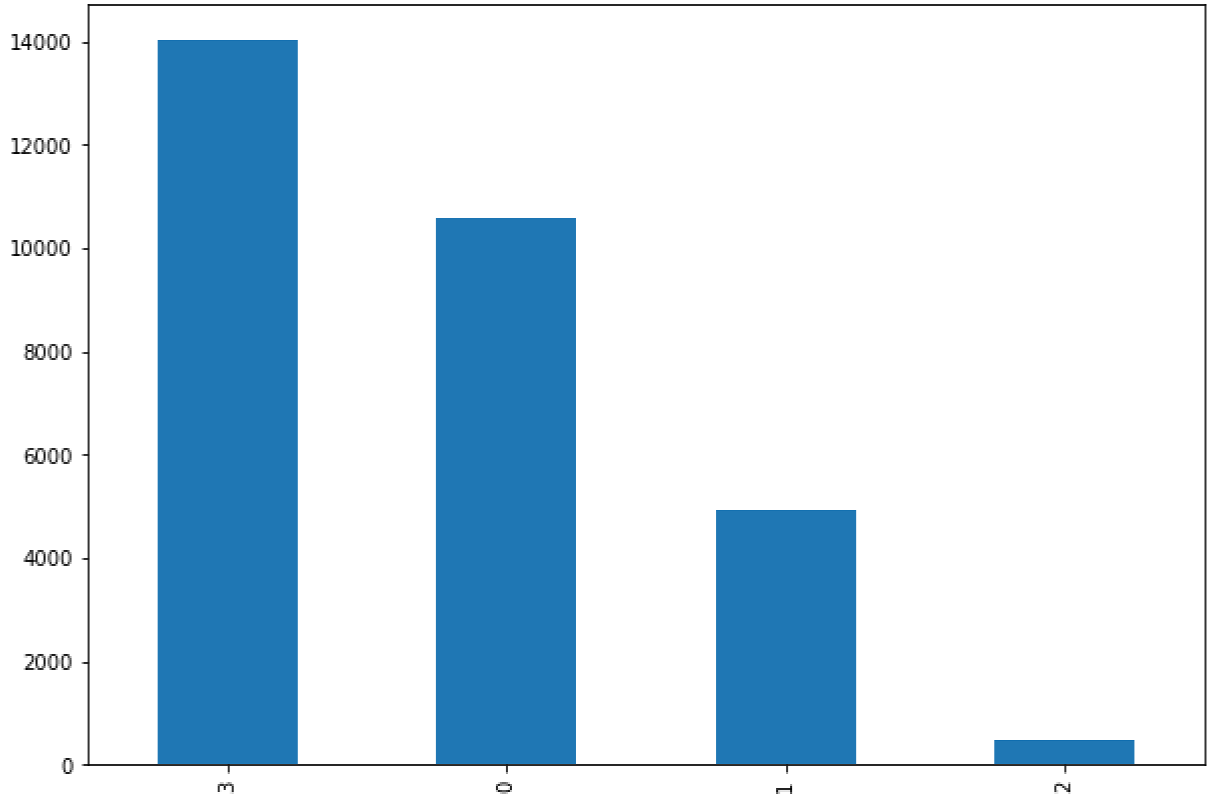
In [38]:
```
df.columns
```

Out[38]:
```
Index(['credit', 'SEX', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default', 'education',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_other',
       'EDUCATION_university'],
      dtype='object')
```

In [39]:
```
df['education'].value_counts()  ## 3 univeristy 0 graduate school  1 high school
```

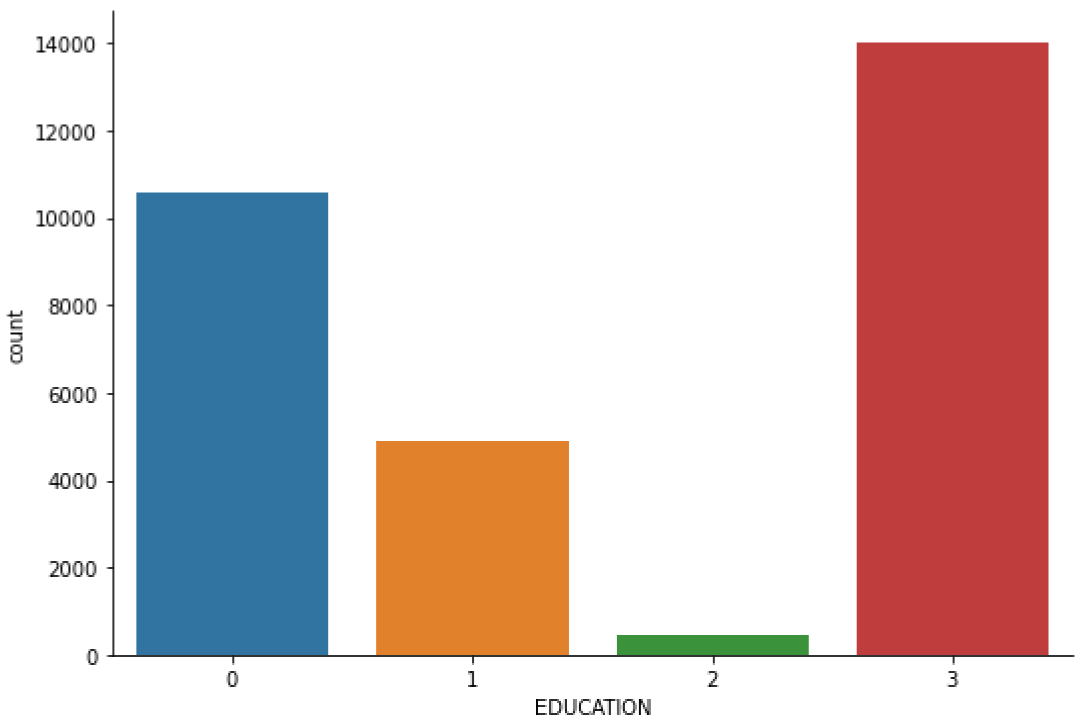
Out[39]:
```
3    14030
0    10585
1     4917
2      468
Name: education, dtype: int64
```

In [40]:
```
ax = df['education'].value_counts().plot(kind='bar', figsize=(10,7))
```

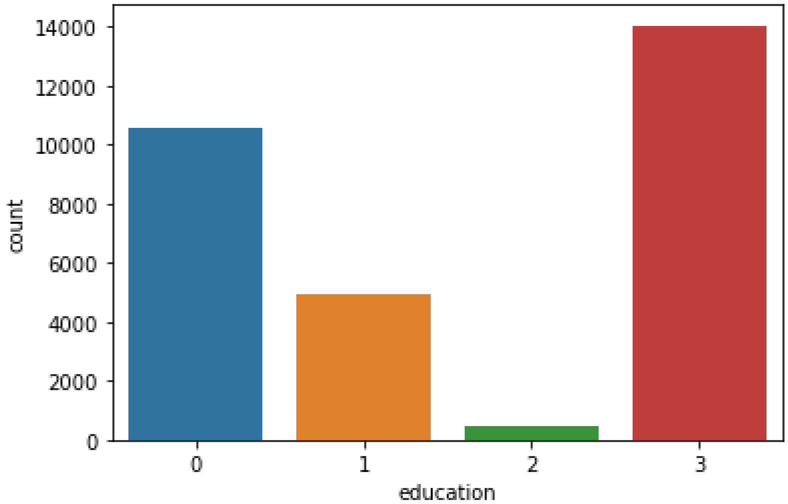


In [41]:
```
fg = sns.catplot(x='education', data=df, kind='count', aspect=1.5)
fg.set_xlabels('EDUCATION')
```

Out[41]:
```
<seaborn.axisgrid.FacetGrid at 0x12306f76310>
```



In [42]:
```
ax = sns.countplot(x="education", data=df)
## https://seaborn.pydata.org/generated/seaborn.countplot.html
```



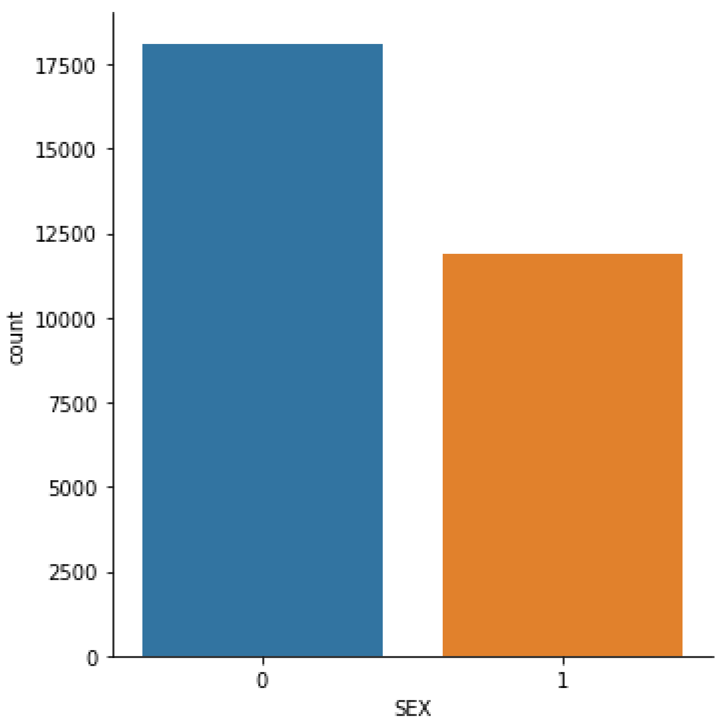
In [43]:
```
df['SEX'].value_counts()
```

Out[43]:
```
0    18112
1    11888
Name: SEX, dtype: int64
```

In [ ]:
```
fg = sns.catplot(x='SEX', data=df, kind='count', aspect=1.5)
fg.set_xlabels('SEX')
```

In [45]:
```
fg = sns.catplot(x='SEX', data=df, kind='count')
fg.set_xlabels('SEX')
```

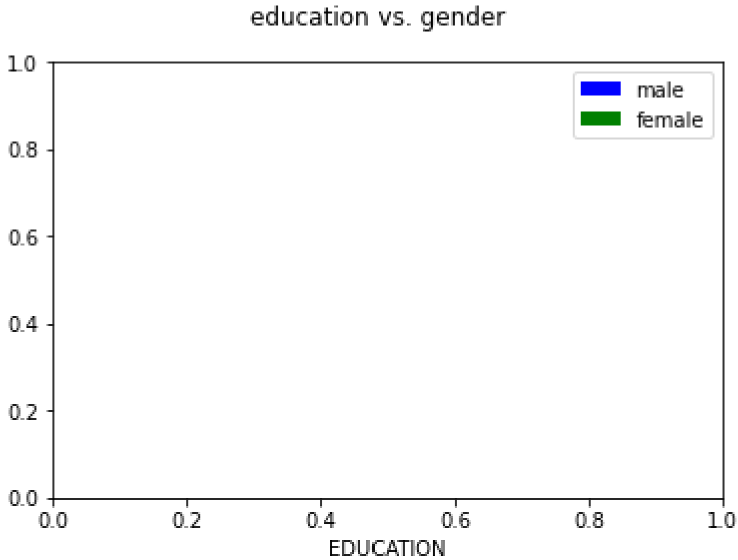
Out[45]:
```
<seaborn.axisgrid.FacetGrid at 0x1230a8adee0>
```



In [46]:
```
fig, ax = plt.subplots()

sns.catplot(x="education", hue="SEX", data=df, kind="count",
            palette={1:"blue", 0:"green"}, ax=ax)
```

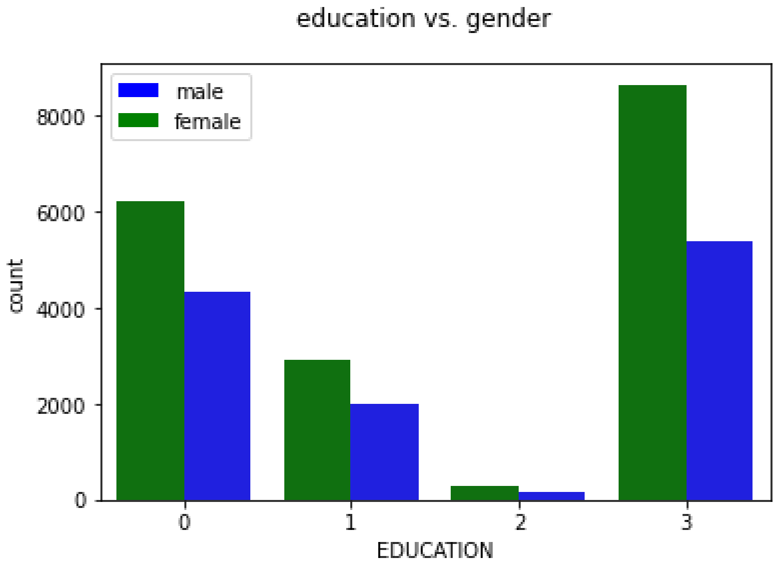
```
plt.close(2) # catplot creates an extra figure we don't need

ax.set_xlabel("EDUCATION")

color_patches = [
    Patch(facecolor="blue", label="male"),
    Patch(facecolor="green", label="female")
]
ax.legend(handles=color_patches)

fig.suptitle("education vs. gender");
```

```
C:\Users\Dongmei\anaconda3\lib\site-packages\seaborn\categorical.py:3762: UserWarning: c
atplot is a figure-level function and does not accept target axes. You may wish to try c
ountplot
  warnings.warn(msg, UserWarning)
```

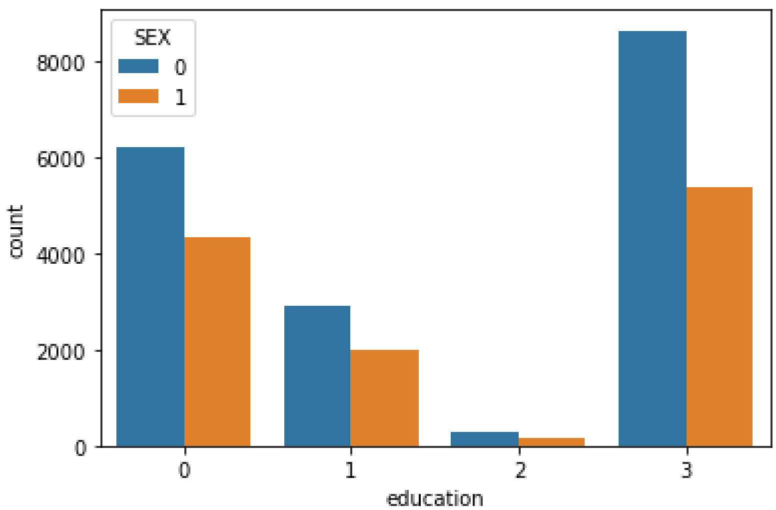


In [48]:

```
fig, ax = plt.subplots()

sns.countplot(x="education", hue="SEX", data=df,
            palette={1:"blue", 0:"green"}, ax=ax)

plt.close(2) # catplot creates an extra figure we don't need

ax.set_xlabel("EDUCATION")

color_patches = [
    Patch(facecolor="blue", label="male"),
    Patch(facecolor="green", label="female")
]
ax.legend(handles=color_patches)

fig.suptitle("education vs. gender");
```



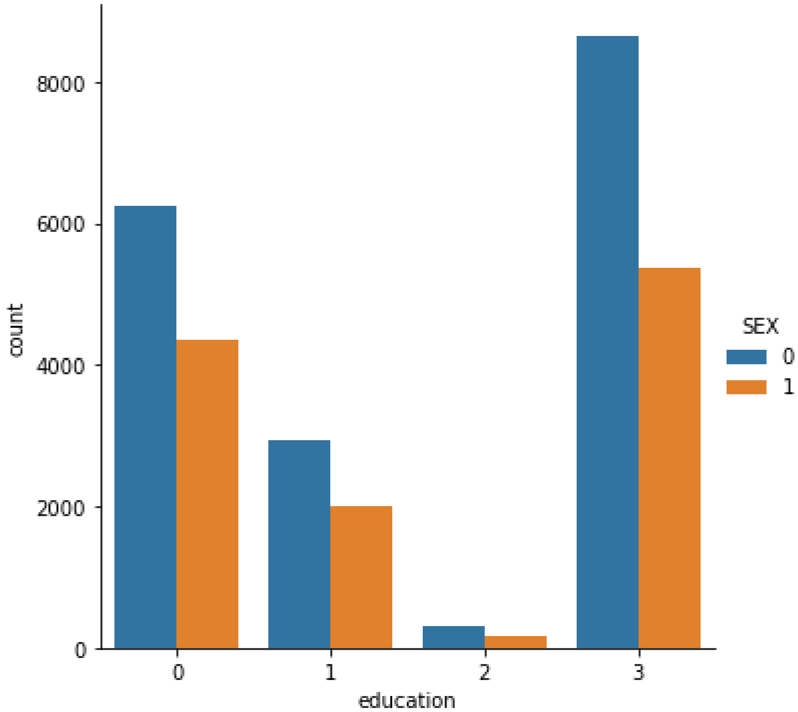
In [49]:
```
ax = sns.countplot(x="education", hue="SEX", data=df)
## https://seaborn.pydata.org/generated/seaborn.countplot.html
```



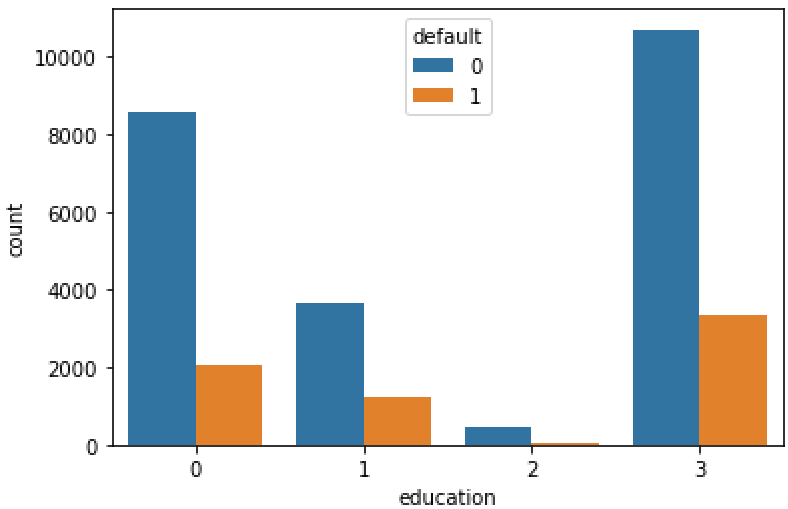
In [ ]:
```
sns.catplot(x= 'education', data=df, kind='count', hue='SEX', aspect=2)
```

In [50]:
```
sns.catplot(x= 'education', data=df, kind='count', hue='SEX')
```

Out[50]: <seaborn.axisgrid.FacetGrid at 0x1230bfbb6a0>



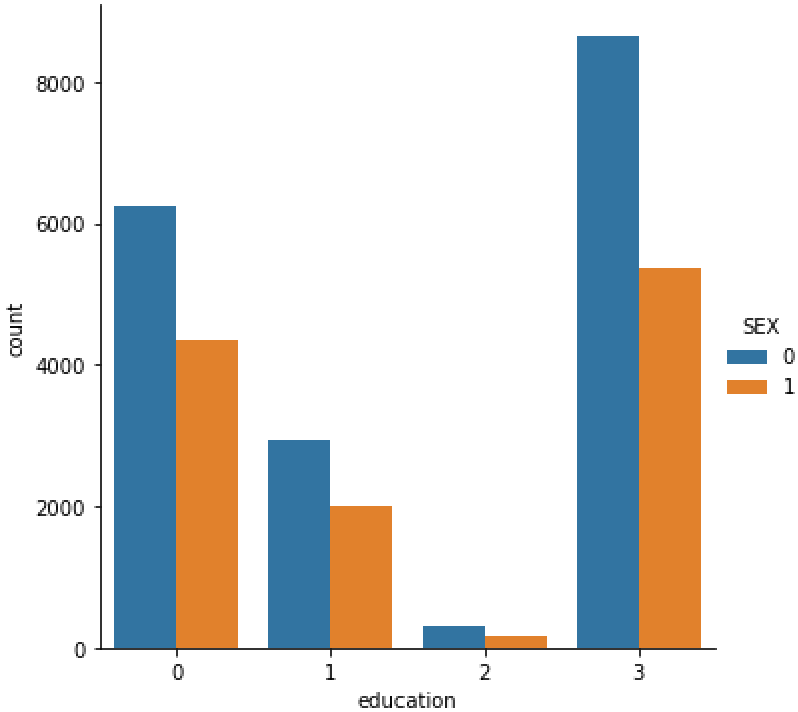
In [66]:
```
ax = sns.countplot(x="education", hue="default", data=df)
```



In [67]:
```
sns.catplot(x="education", hue="SEX", data=df, kind="count", ax=ax)
```

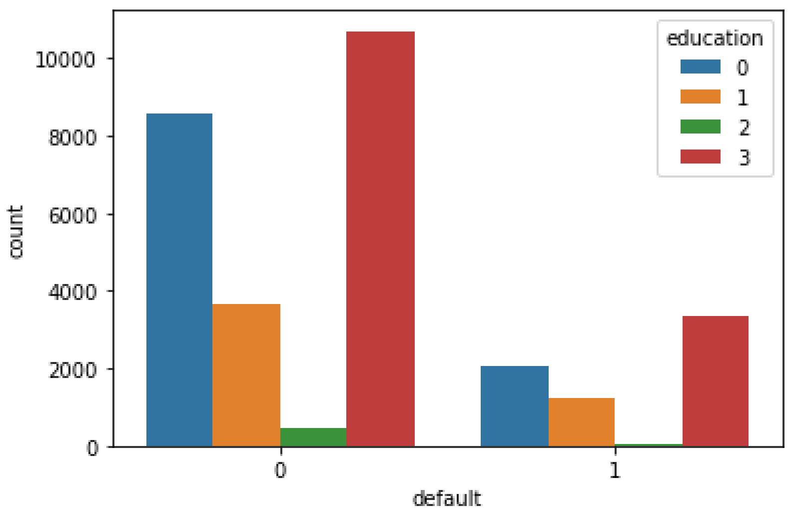
```
C:\Users\Dongmei\anaconda3\lib\site-packages\seaborn\categorical.py:3762: UserWarning: c
atplot is a figure-level function and does not accept target axes. You may wish to try c
ountplot
  warnings.warn(msg, UserWarning)
```

Out[67]: <seaborn.axisgrid.FacetGrid at 0x12307254310>

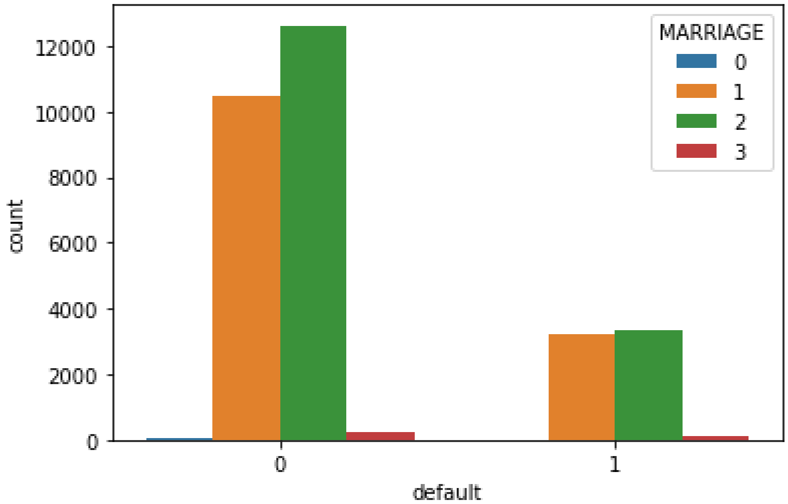



```
In [ ]: ax = sns.countplot(y="EDUCATION", hue="SEX", data=df)
```

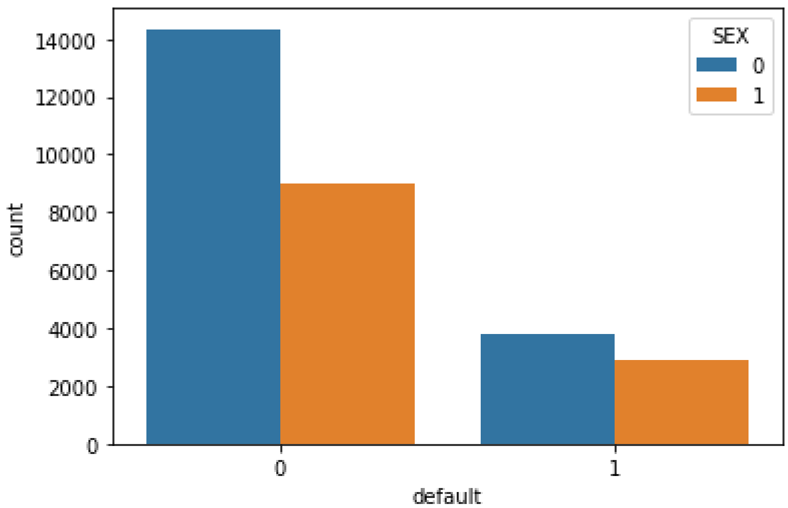
```
In [51]: ax = sns.countplot(x="default", hue="education", data=df)  ### 3 high school
```




```
In [52]: ax = sns.countplot(x="default", hue="MARRIAGE", data=df)
```

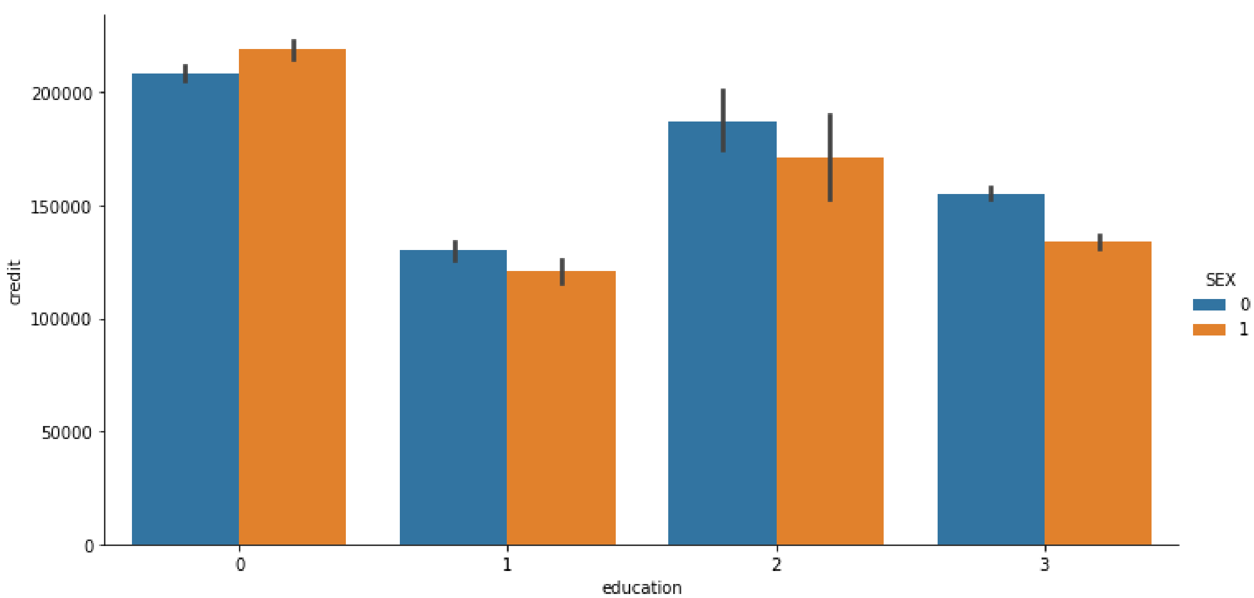


In [53]:
```
ax = sns.countplot(x="default", hue="SEX", data=df)
```



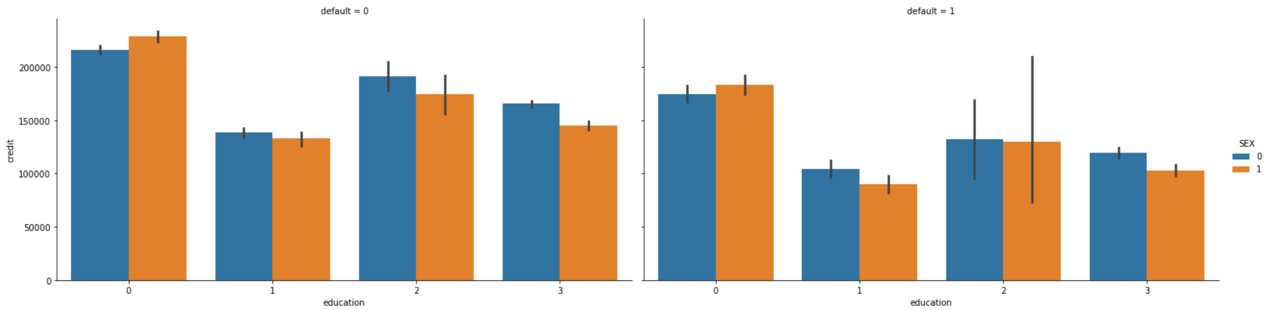
In [60]:
```
sns.catplot(x= 'education', y='credit', data=df, kind='bar', hue='SEX', aspect=2)
```

Out[60]: <seaborn.axisgrid.FacetGrid at 0x1230c596fa0>



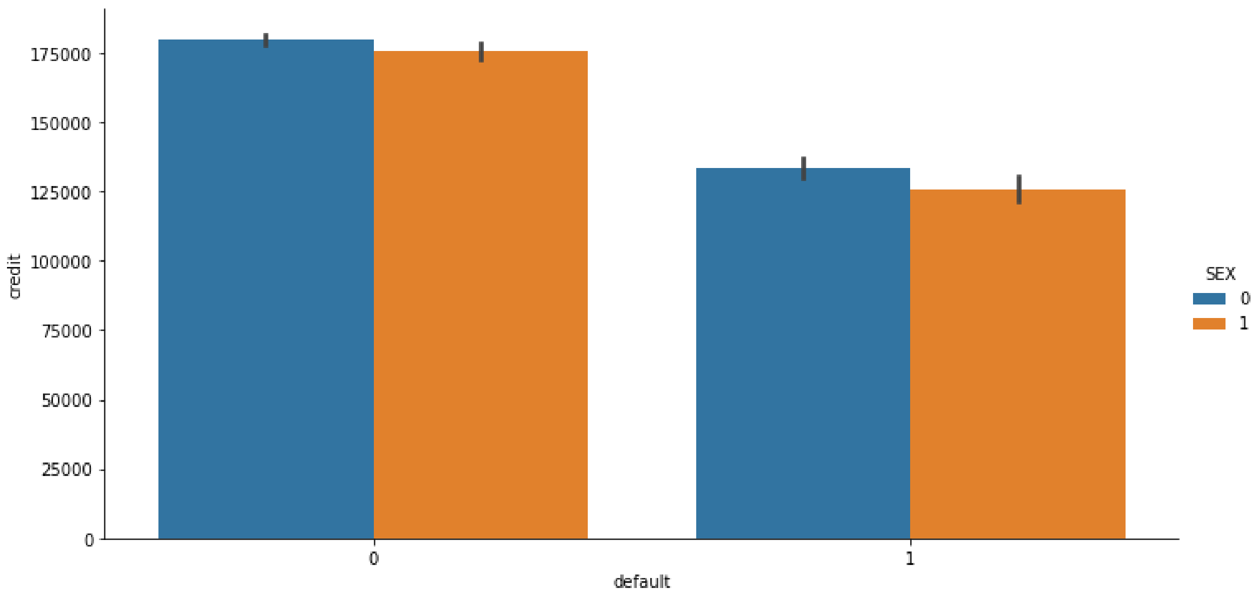
In [61]:
```
sns.catplot(x='education', y='credit', data=df, kind='bar', hue='SEX', col='default', a
```

Out[61]: <seaborn.axisgrid.FacetGrid at 0x1230c4d14c0>



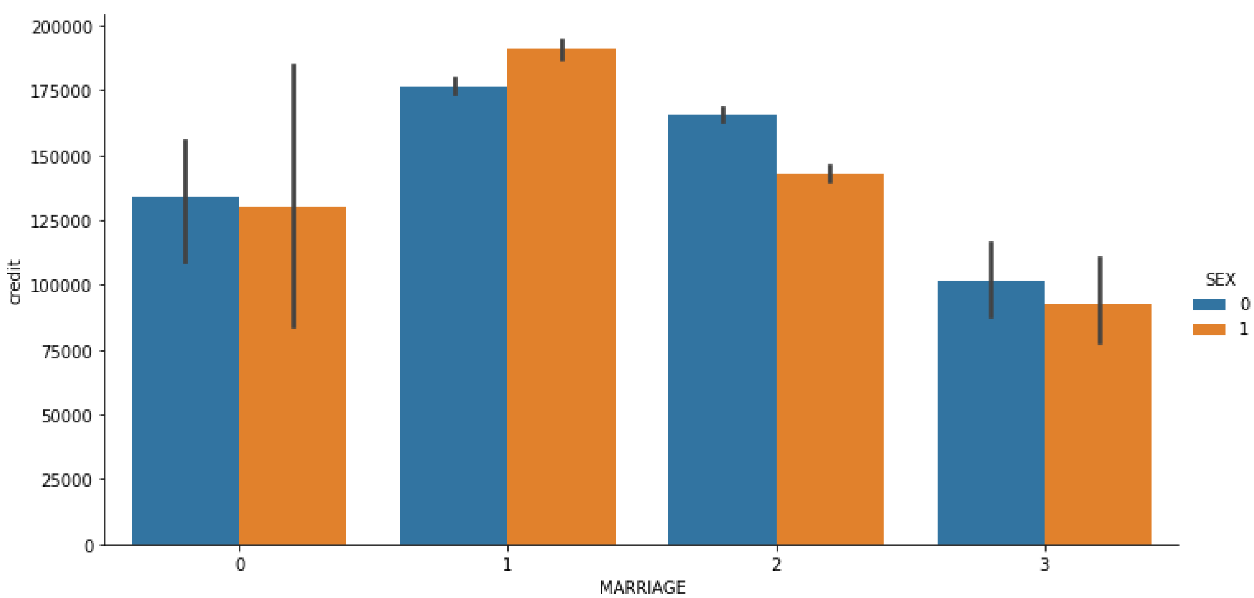
In [62]:
```
sns.catplot(x='default', y='credit', data=df, kind='bar', hue='SEX', aspect=2)
```

Out[62]: <seaborn.axisgrid.FacetGrid at 0x1230729f820>



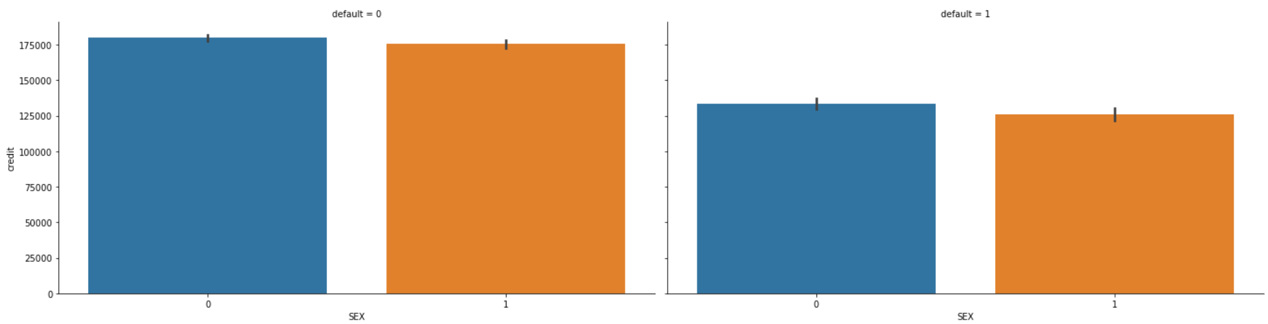
In [63]:
```
sns.catplot(x='MARRIAGE', y='credit', data=df, kind='bar', hue='SEX', aspect=2)
```

Out[63]: <seaborn.axisgrid.FacetGrid at 0x1230c504e80>

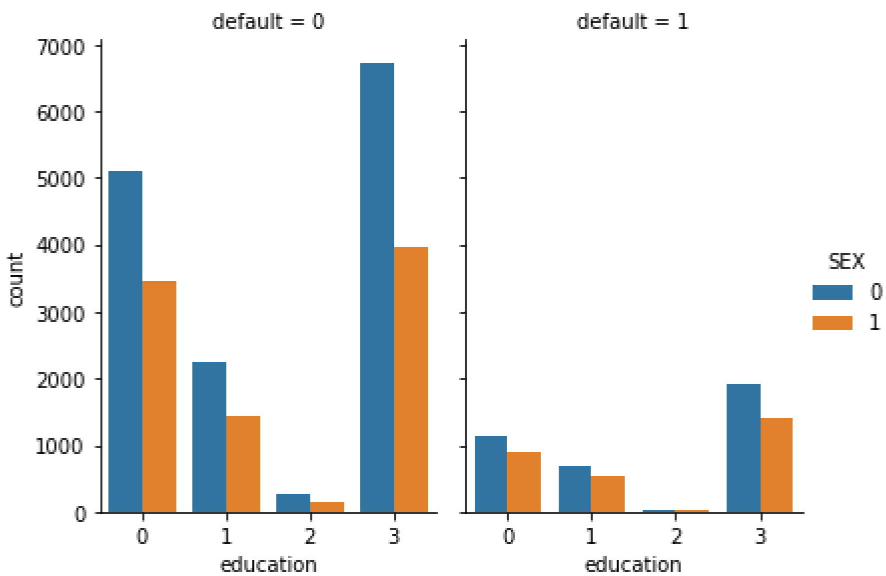


In [64]:
```
sns.catplot(x='SEX', y='credit', data=df, kind='bar', col='default', aspect=2)
```

Out[64]: <seaborn.axisgrid.FacetGrid at 0x12306e4c4c0>



In [69]:
```
g = sns.catplot(x="education", hue="SEX", col="default",
                data=df, kind="count",
                height=4, aspect=.7);
```



In [70]:
```
counts_df = df.groupby(["education", "default"])["SEX"].count().unstack()
counts_df
```

Out[70]:

default	0	1
education		
0	8549	2036
1	3680	1237
2	435	33
3	10700	3330

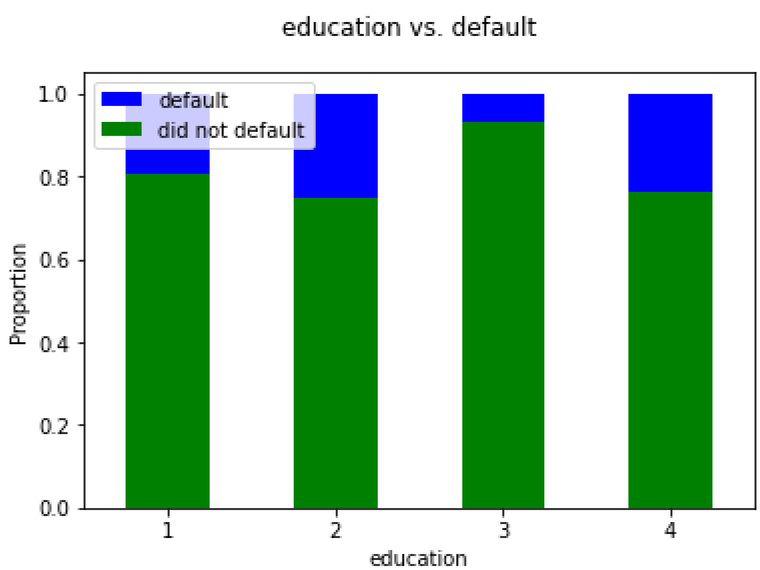
In [71]:
```
default_percents_df = counts_df.T.div(counts_df.T.sum()).T
default_percents_df
```

Out[71]:

default	0	1
education		

default	0	1
education		
0	0.807652	0.192348
1	0.748424	0.251576
2	0.929487	0.070513
3	0.762651	0.237349

In [73]:
```
fig, ax = plt.subplots()

default_percents_df.plot(kind="bar", stacked=True, color=["green", "blue"], ax=ax)

ax.set_xlabel("education")
ax.set_xticklabels([1, 2, 3, 4], rotation=0)
ax.set_ylabel("Proportion")

color_patches = [
    Patch(facecolor="blue", label="default"),
    Patch(facecolor="green", label="did not default")
]
ax.legend(handles=color_patches)

fig.suptitle("education vs. default");
```



In [74]:
```
df.describe()
```

Out[74]:

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PA
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000
mean	167484.322667	0.396267	1.551867	35.485500	-0.016700	-0.133767	-0.166
std	129747.661567	0.489129	0.521970	9.217904	1.123802	1.197186	1.196
min	10000.000000	0.000000	0.000000	21.000000	-2.000000	-2.000000	-2.000
25%	50000.000000	0.000000	1.000000	28.000000	-1.000000	-1.000000	-1.000

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PA
50%	140000.000000	0.000000	2.000000	34.000000	0.000000	0.000000	0.000
75%	240000.000000	1.000000	2.000000	41.000000	0.000000	0.000000	0.000
max	1000000.000000	1.000000	3.000000	79.000000	8.000000	8.000000	8.000

8 rows × 28 columns

In [76]:
```
df.credit.describe()
```

Out[76]:
```
count      30000.000000
mean      167484.322667
std       129747.661567
min        10000.000000
25%        50000.000000
50%       140000.000000
75%       240000.000000
max      1000000.000000
Name: credit, dtype: float64
```

In [75]:
```
print('Average and median credit are %0.f and %0.f, respectively'%(df.credit.mean(),
                                                                      df.credit.med
```

```
Average and median credit are 167484 and 140000, respectively
```

In [86]:
```
df.head()
```

Out[86]:
```
0     20000
1    120000
2     90000
3     50000
4     50000
Name: credit, dtype: int64
```

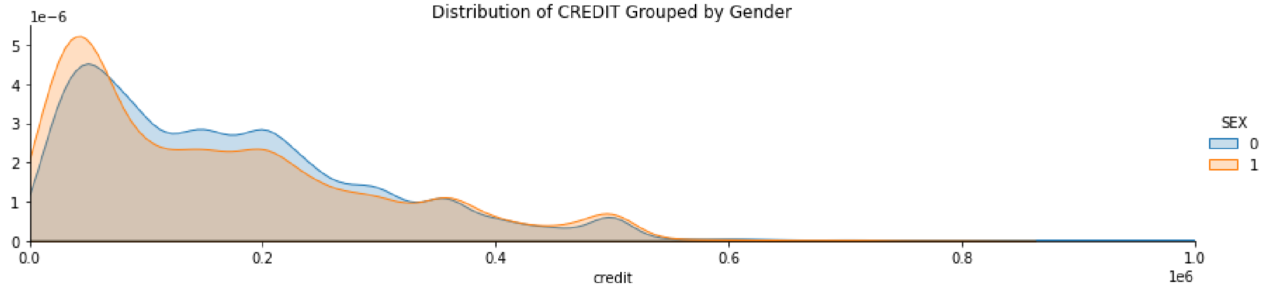
In [87]:
```
df = pd.read_csv('creditEDA.csv')
```

In [88]:
```
df.columns
```

Out[88]:
```
Index(['credit', 'SEX', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default', 'education',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_other',
       'EDUCATION_university'],
      dtype='object')
```

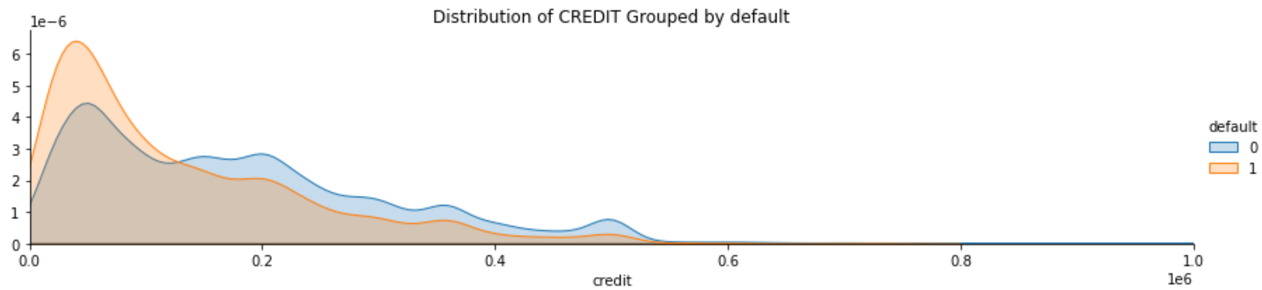
In [89]:
```
fig = sns.FacetGrid(df, hue='SEX', aspect=4)
fig.map(sns.kdeplot, 'credit', shade=True)
oldest = df['credit'].max()
fig.set(xlim=(0,oldest))
fig.set(title='Distribution of CREDIT Grouped by Gender')
fig.add_legend()
```

Out[89]:
```
<seaborn.axisgrid.FacetGrid at 0x12307a1b6a0>
```



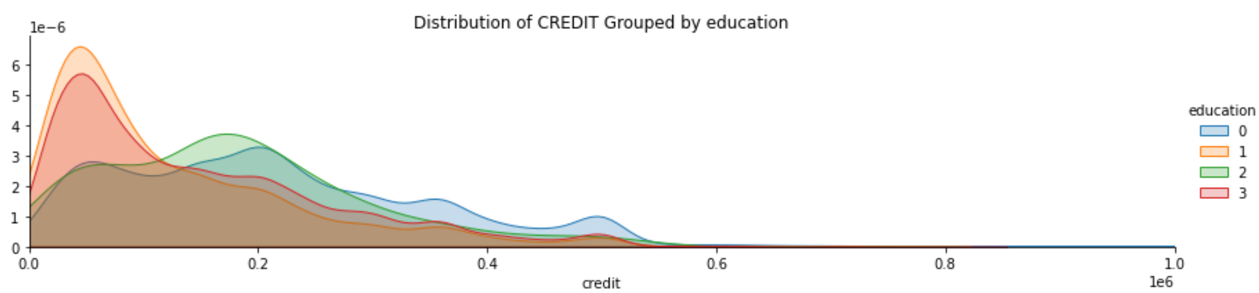
In [90]:
```python
fig = sns.FacetGrid(df, hue='default', aspect=4)
fig.map(sns.kdeplot, 'credit', shade=True)
oldest = df['credit'].max()
fig.set(xlim=(0,oldest))
fig.set(title='Distribution of CREDIT Grouped by default')
fig.add_legend()
```

Out[90]: <seaborn.axisgrid.FacetGrid at 0x1230c21e970>



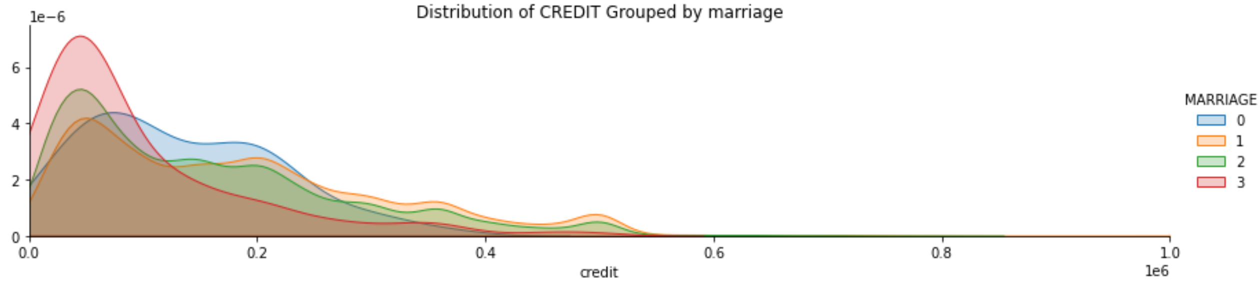
In [91]:
```python
fig = sns.FacetGrid(df, hue='education', aspect=4)
fig.map(sns.kdeplot, 'credit', shade=True)
oldest = df['credit'].max()
fig.set(xlim=(0,oldest))
fig.set(title='Distribution of CREDIT Grouped by education')
fig.add_legend()
```

Out[91]: <seaborn.axisgrid.FacetGrid at 0x12307146b50>



In [94]:
```python
fig = sns.FacetGrid(df, hue='MARRIAGE', aspect=4)
fig.map(sns.kdeplot, 'credit', shade=True)
oldest = df['credit'].max()
fig.set(xlim=(0,oldest))
fig.set(title='Distribution of CREDIT Grouped by marriage')
fig.add_legend()
```

Out[94]: <seaborn.axisgrid.FacetGrid at 0x1230c4f01f0>

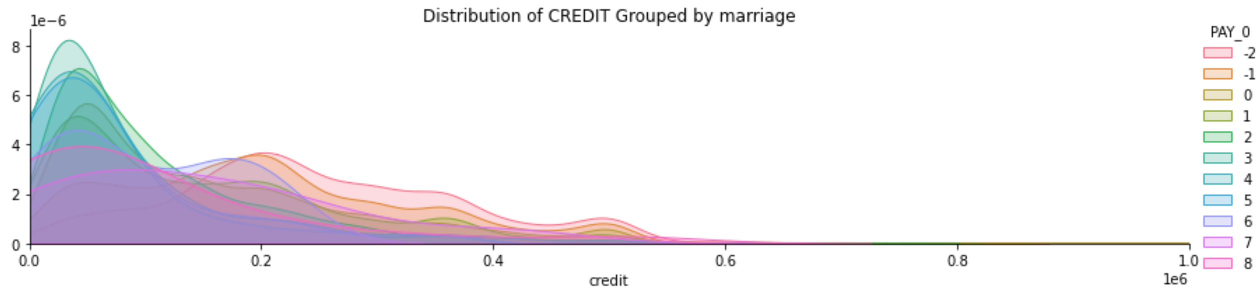


In [100…

```
fig = sns.FacetGrid(df, hue='PAY_0', aspect=4)
fig.map(sns.kdeplot, 'credit', shade=True)
oldest = df['credit'].max()
fig.set(xlim=(0,oldest))
fig.set(title='Distribution of CREDIT Grouped by marriage')
fig.add_legend()
```

Out[100…

```
<seaborn.axisgrid.FacetGrid at 0x123075ea8b0>
```



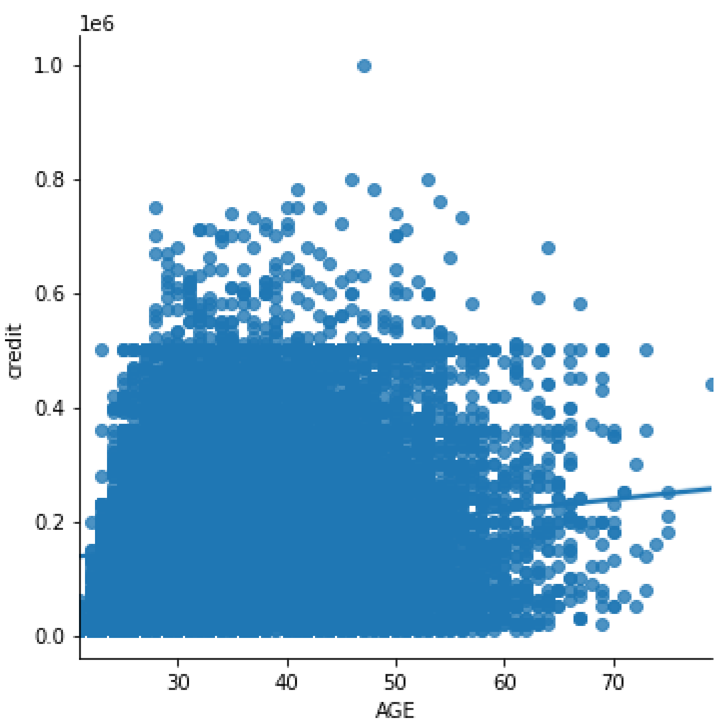
In [95]:

```
sns.lmplot('AGE', 'credit', data=df)
```

```
C:\Users\Dongmei\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variables as keyword args: x, y. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```

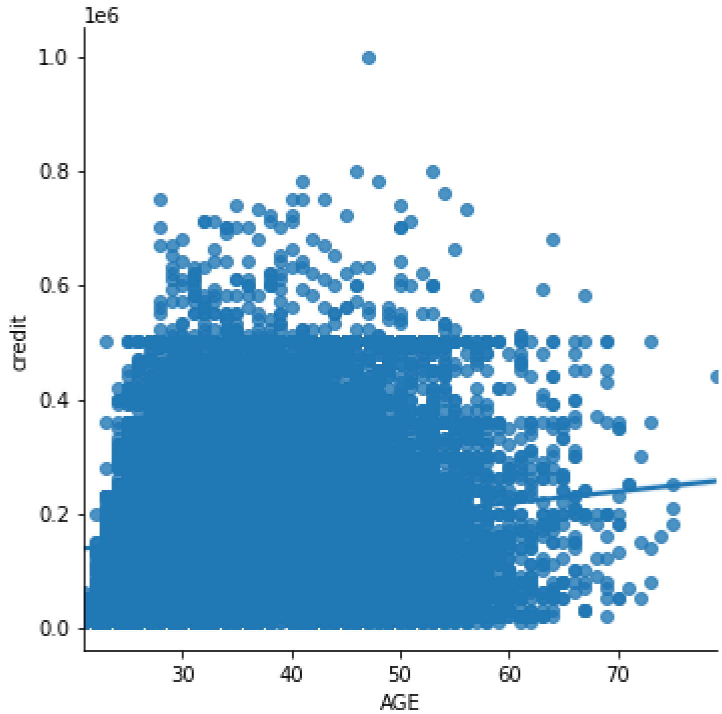
Out[95]:

```
<seaborn.axisgrid.FacetGrid at 0x1230c4ff5b0>
```



In [96]:
```
sns.lmplot(x='AGE', y='credit', data=df)  ## address the warning
```

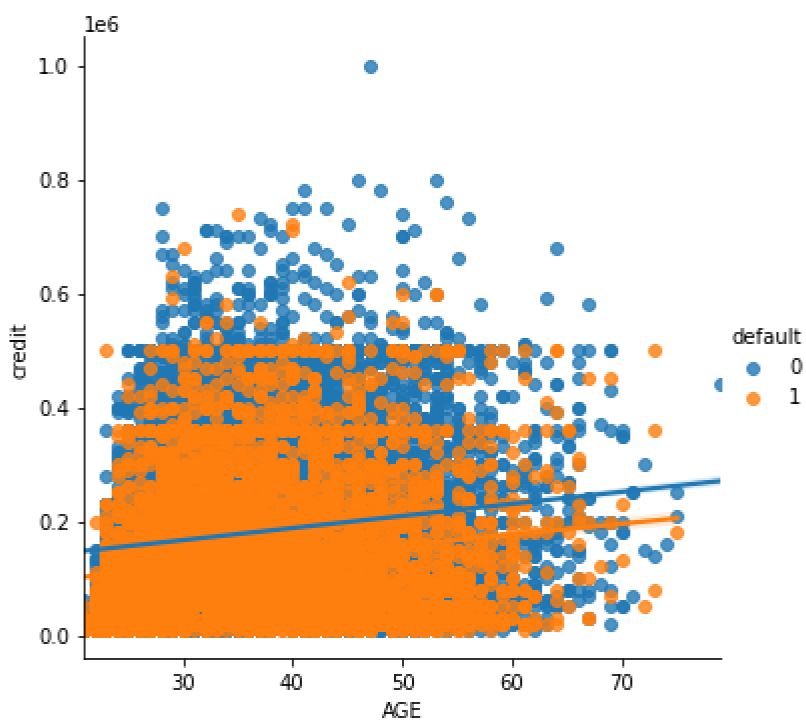
Out[96]: <seaborn.axisgrid.FacetGrid at 0x1230c27d310>



In [97]:
```
sns.lmplot('AGE', 'credit', data=df, hue='default')
```

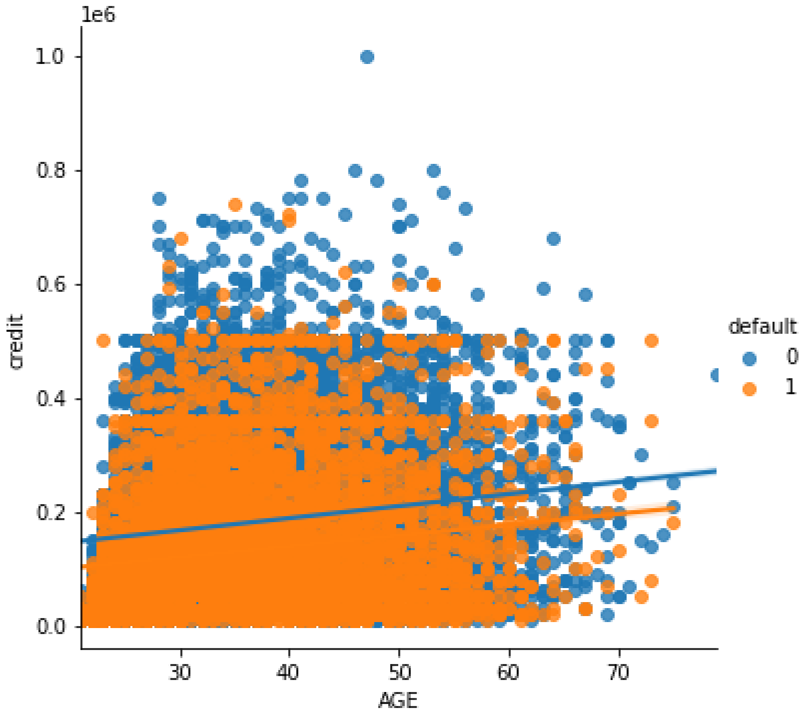
```
C:\Users\Dongmei\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: P
ass the following variables as keyword args: x, y. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```

Out[97]: <seaborn.axisgrid.FacetGrid at 0x12307ab9160>



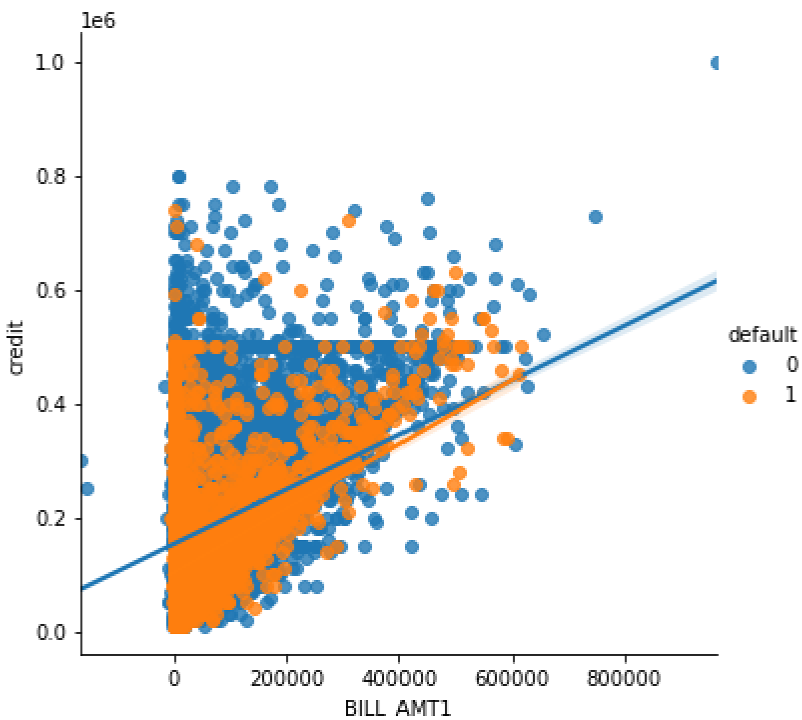
In [98]:
```
sns.lmplot(x='AGE', y='credit', data=df, hue='default')
```

Out[98]: <seaborn.axisgrid.FacetGrid at 0x123078cad30>



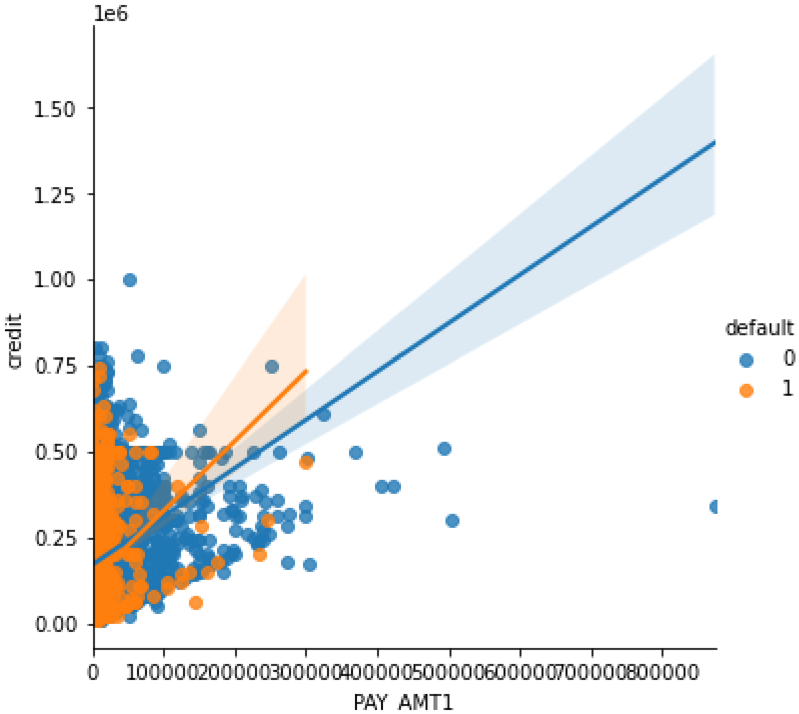
In [101…
```
sns.lmplot(x='BILL_AMT1', y='credit', data=df, hue='default')
```

Out[101…: <seaborn.axisgrid.FacetGrid at 0x1230788c760>



In [102…
```
sns.lmplot(x='PAY_AMT1', y='credit', data=df, hue='default')
```

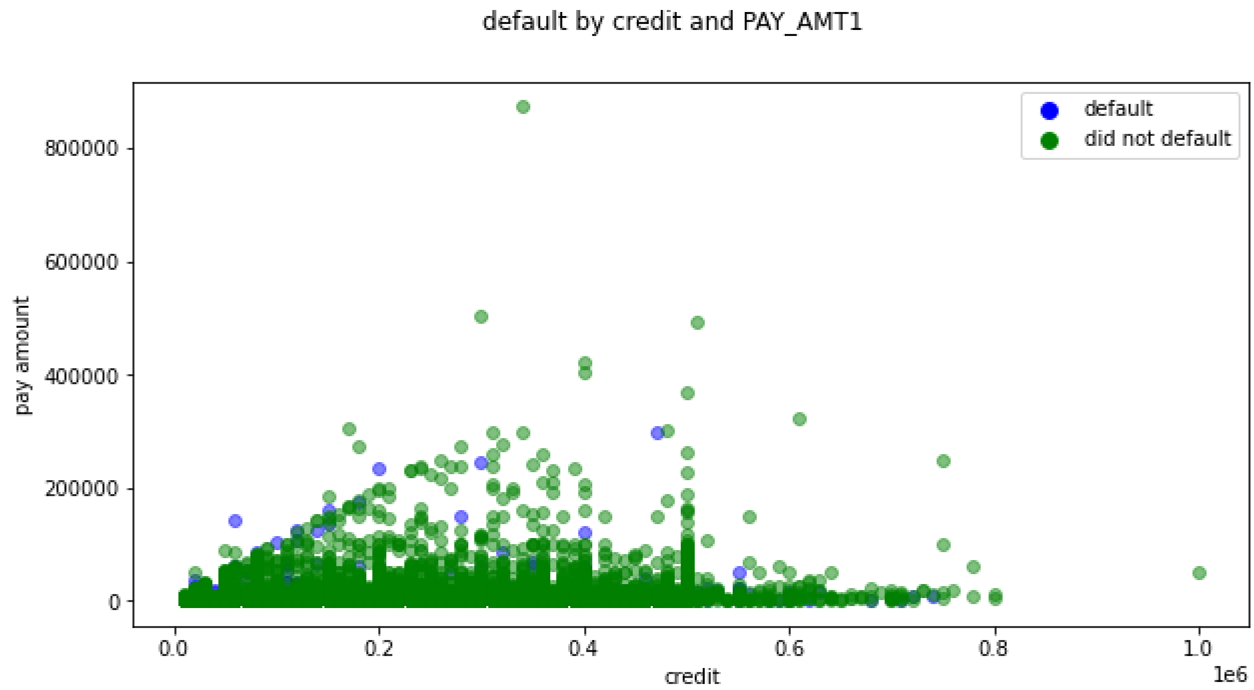
Out[102…: <seaborn.axisgrid.FacetGrid at 0x12307894160>



In [ ]:

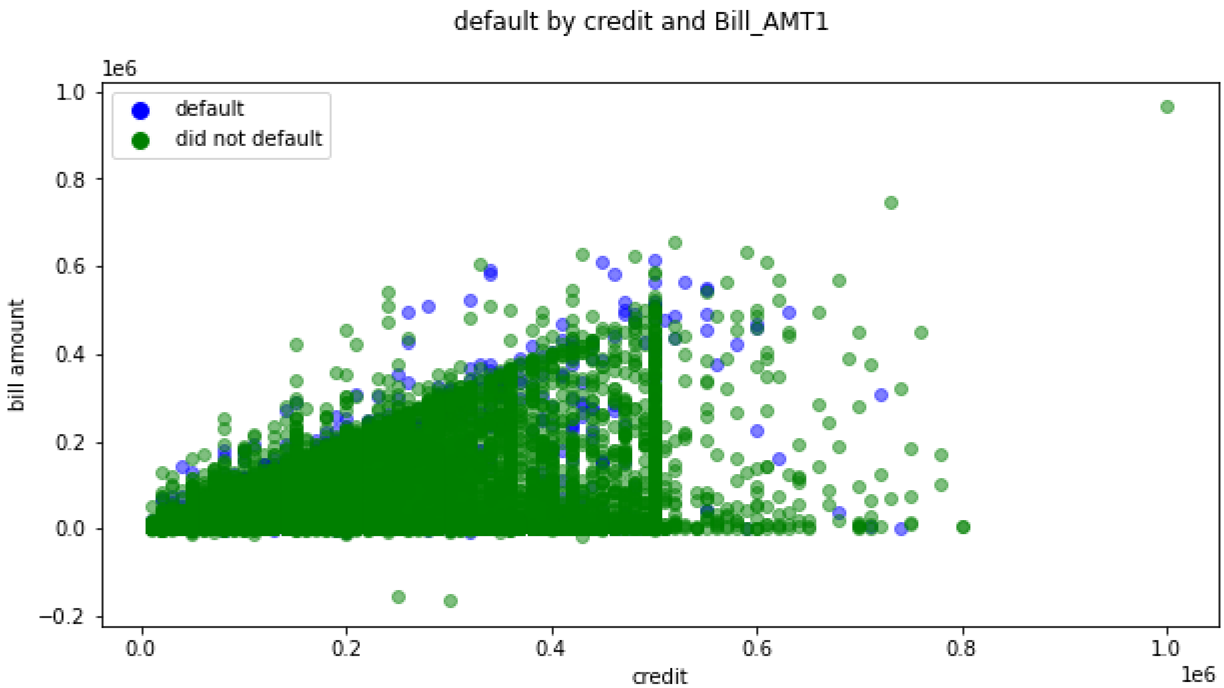
In [109…

```
## Numeric vs. Numeric vs. Categorical EDA

fig, ax = plt.subplots(figsize=(10, 5))

ax.scatter(df[df["default"]==1]["credit"], df[df["default"]==1]["PAY_AMT1"], c="blue",
ax.scatter(df[df["default"]==0]["credit"], df[df["default"]==0]["PAY_AMT1"], c="green",

ax.set_xlabel("credit")
ax.set_ylabel("pay amount")

color_patches = [
    Line2D([0], [0], marker='o', color='w', label='default', markerfacecolor='b', marke
    Line2D([0], [0], marker='o', color='w', label='did not default', markerfacecolor='g
]
ax.legend(handles=color_patches)

fig.suptitle("default by credit and PAY_AMT1");
```

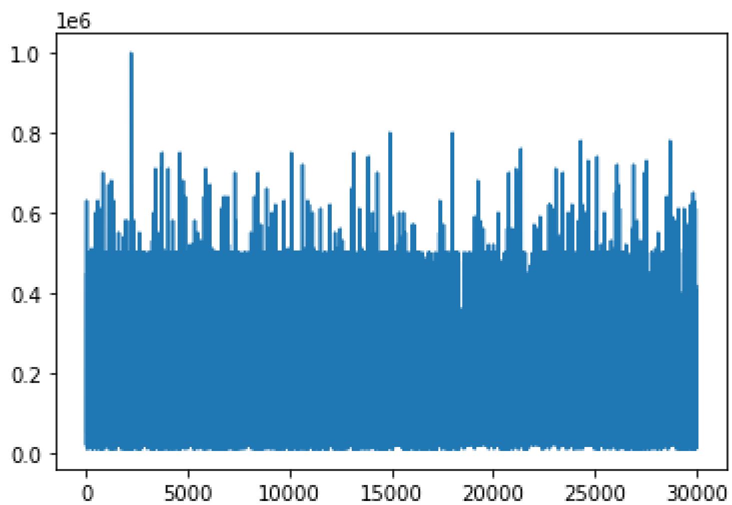


In [110…

```
## Numeric vs. Numeric vs. Categorical EDA

fig, ax = plt.subplots(figsize=(10, 5))

ax.scatter(df[df["default"]==1]["credit"], df[df["default"]==1]["BILL_AMT1"], c="blue",
ax.scatter(df[df["default"]==0]["credit"], df[df["default"]==0]["BILL_AMT1"], c="green"

ax.set_xlabel("credit")
ax.set_ylabel("bill amount")

color_patches = [
    Line2D([0], [0], marker='o', color='w', label='default', markerfacecolor='b', marke
    Line2D([0], [0], marker='o', color='w', label='did not default', markerfacecolor='g
]
ax.legend(handles=color_patches)

fig.suptitle("default by credit and Bill_AMT1");
```

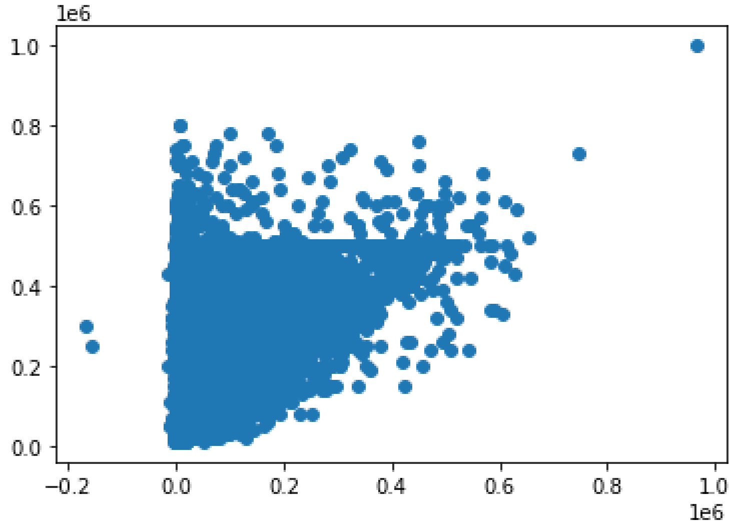


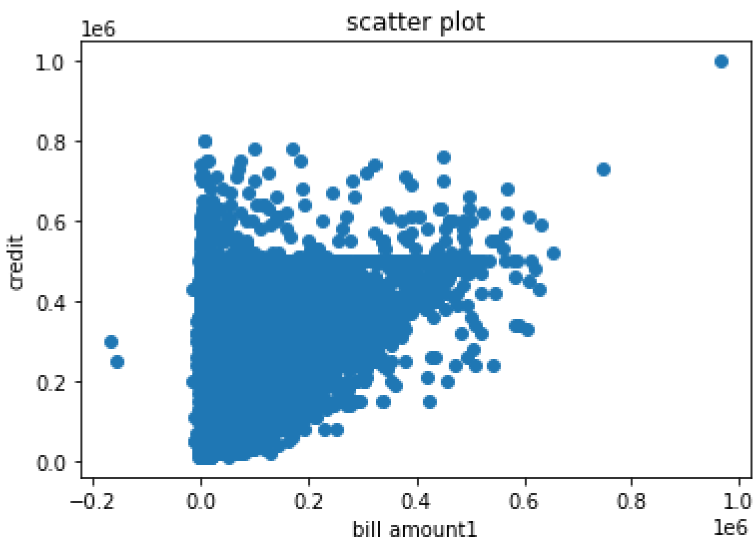
In [111…
```
plt.plot(df['credit'])
plt.show()
```



In [114…
```
x = df['BILL_AMT1']
y = df['credit']
```

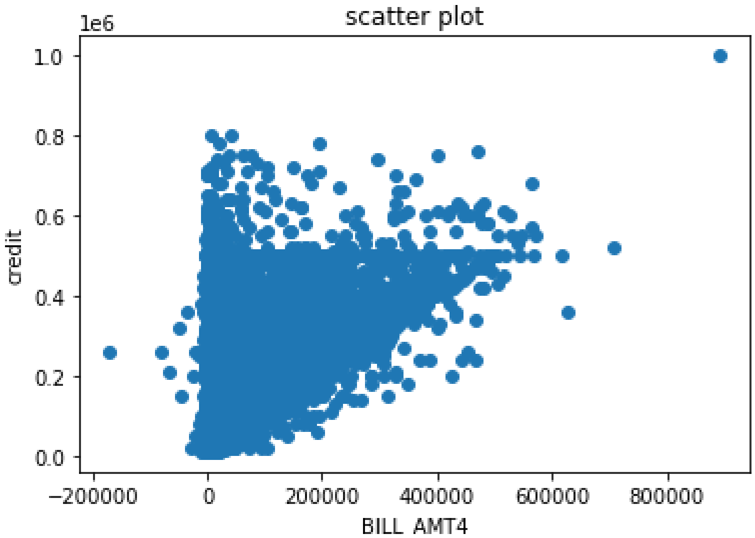
In [115…
```
plt.scatter(x,y)
plt.show()

plt.scatter(x,y, marker='o')
plt.title('scatter plot')
plt.xlabel('bill amount1')
plt.ylabel('credit')
plt.show()
```





In [116…
```
x = df['BILL_AMT4']
y = df['credit']
```

In [117…
```
plt.scatter(x,y, marker='o')
plt.title('scatter plot')
plt.xlabel('BILL_AMT4')
plt.ylabel('credit')
plt.show()
```

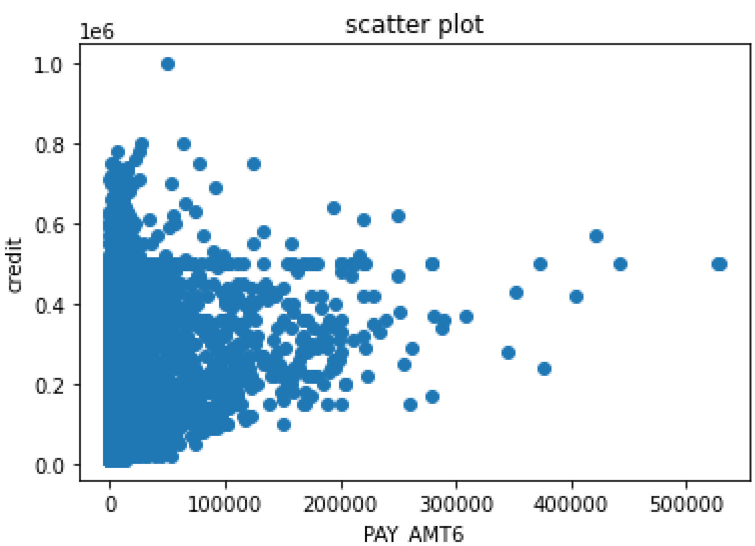


In [119…

```
x = df['PAY_AMT6']
y = df['credit']
```

In [120…

```
plt.scatter(x,y, marker='o')
plt.title('scatter plot')
plt.xlabel('PAY_AMT6')
plt.ylabel('credit')
plt.show()
```



In [121…

```
df.columns
```

Out[121…

```
Index(['credit', 'SEX', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default', 'education',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_other',
       'EDUCATION_university'],
      dtype='object')
```

In [ ]:

```
## Covariance
### Covariance is often used to gauge the linear degree of change between two variables
### This will be very important when studying the impact various features might have on
```

In [122…
```
covMat = df.cov()
print(covMat)
```

```
                                 credit          SEX     MARRIAGE  \
credit                     1.683446e+10 -1571.050630 -7323.669658
SEX                       -1.571051e+03     0.239247     0.008014
MARRIAGE                  -7.323670e+03     0.008014     0.272452
AGE                        1.730767e+05     0.409726    -1.992764
PAY_0                     -3.954593e+04     0.031685     0.011683
PAY_2                     -4.603765e+04     0.041442     0.015122
PAY_3                     -4.443225e+04     0.038694     0.020421
PAY_4                     -4.057181e+04     0.034411     0.020213
PAY_5                     -3.667056e+04     0.030521     0.021074
PAY_6                     -3.509308e+04     0.024754     0.020616
BILL_AMT1                  2.727020e+09  1211.694332  -902.154685
BILL_AMT2                  2.570130e+09  1085.595467  -802.517866
BILL_AMT3                  2.548533e+09   833.207432  -901.679085
BILL_AMT4                  2.453926e+09   688.489572  -783.881599
BILL_AMT5                  2.331481e+09   505.694333  -805.840875
BILL_AMT6                  2.243837e+09   487.430160  -659.223347
PAY_AMT1                   4.195711e+08     1.964266   -51.691615
PAY_AMT2                   5.333504e+08    15.675500   -97.327974
PAY_AMT3                   4.801180e+08    74.034812   -32.546082
PAY_AMT4                   4.131202e+08    17.080110  -103.518204
PAY_AMT5                   4.305657e+08    12.458809    -9.607709
PAY_AMT6                   5.065153e+08    24.051885   -61.623271
default                   -8.267552e+03     0.008113    -0.005273
education                 -3.493060e+04    -0.018208    -0.062974
EDUCATION_graduate school  1.604448e+04     0.005317     0.035451
EDUCATION_high school     -6.709315e+03     0.001385    -0.021418
EDUCATION_other            2.157851e+02    -0.000515    -0.000542
EDUCATION_university      -9.550953e+03    -0.006188    -0.013490

                                     AGE         PAY_0         PAY_2  \
credit                     173076.722569 -39545.930009 -46037.648360
SEX                             0.409726      0.031685      0.041442
MARRIAGE                       -1.992764      0.011683      0.015122
AGE                            84.969755     -0.408639     -0.553408
PAY_0                          -0.408639      1.262930      0.904330
PAY_2                          -0.553408      0.904330      1.433254
PAY_3                          -0.585263      0.772384      1.098371
PAY_4                          -0.535851      0.707972      0.926680
PAY_5                          -0.562245      0.648743      0.844886
PAY_6                          -0.517022      0.613292      0.792320
BILL_AMT1                   38172.933546  15480.304170  20706.614217
BILL_AMT2                   35613.657962  15185.916919  20045.829482
BILL_AMT3                   34334.251320  14011.556537  18609.510991
BILL_AMT4                   30453.108180  12950.248389  17116.298983
BILL_AMT5                   27654.067800  12341.668685  16110.952468
BILL_AMT6                   26137.648547  11844.759724  15642.875812
PAY_AMT1                     3992.041735  -1475.495089  -1600.240756
PAY_AMT2                     4626.861549  -1815.138407  -1627.192336
PAY_AMT3                     4746.824393  -1396.168258  -1178.331282
PAY_AMT4                     3087.324192  -1126.847945   -878.843879
PAY_AMT5                     3218.052172   -999.107730   -678.468530
PAY_AMT6                     3191.903901  -1172.193614   -776.835035
default                         0.053143      0.151499      0.130960
education                      -0.261453      0.184394      0.238080
EDUCATION_graduate school      -0.442349     -0.076644     -0.096806
EDUCATION_high school           0.789120      0.024505      0.028625
EDUCATION_other                 0.010260     -0.003473     -0.004913
EDUCATION_university           -0.357031      0.055612      0.073094

                                   PAY_3         PAY_4         PAY_5  \
credit                     -44432.253315 -40571.811859 -36670.562325
```

```
SEX                            0.038694       0.034411       0.030521
MARRIAGE                       0.020421       0.020213       0.021074
AGE                           -0.585263      -0.535851      -0.562245
PAY_0                          0.772384       0.707972       0.648743
PAY_2                          1.098371       0.926680       0.844886
PAY_3                          1.432492       1.087761       0.931455
PAY_4                          1.087761       1.366885       1.086161
PAY_5                          0.931455       1.086161       1.284114
PAY_6                          0.870815       0.963263       1.064545
BILL_AMT1                  18373.210469   17460.198259   17246.377531
BILL_AMT2                  20214.071495   18790.627741   18301.285286
BILL_AMT3                  18882.491544   19862.999426   19122.663330
BILL_AMT4                  17494.100555   18496.423186   19822.925512
BILL_AMT5                  16382.947539   17265.551898   18586.590324
BILL_AMT6                  15847.089648   16651.586314   17715.690075
PAY_AMT1                      25.668468    -181.295613    -114.281714
PAY_AMT2                   -1841.952825     -52.358166     -83.324487
PAY_AMT3                   -1123.428782   -1425.205189     180.812142
PAY_AMT4                    -863.762183    -796.035739   -1034.961970
PAY_AMT5                    -655.796002    -599.991629    -577.161017
PAY_AMT6                    -763.026041    -552.137338    -463.892613
default                        0.116867       0.105115       0.096020
education                      0.224645       0.209388       0.187787
EDUCATION_graduate school     -0.091629      -0.085144      -0.075112
EDUCATION_high school          0.027674       0.025268       0.020798
EDUCATION_other               -0.005107      -0.004491      -0.004047
EDUCATION_university           0.069062       0.064367       0.058361

                                  PAY_6  ...      PAY_AMT3      PAY_AMT4  \
credit                    -35093.083441  ...  4.801180e+08  4.131202e+08
SEX                            0.024754  ...  7.403481e+01  1.708011e+01
MARRIAGE                       0.020616  ... -3.254608e+01 -1.035182e+02
AGE                           -0.517022  ...  4.746824e+03  3.087324e+03
PAY_0                          0.613292  ... -1.396168e+03 -1.126848e+03
PAY_2                          0.792320  ... -1.178331e+03 -8.788439e+02
PAY_3                          0.870815  ... -1.123429e+03 -8.637622e+02
PAY_4                          0.963263  ... -1.425205e+03 -7.960357e+02
PAY_5                          1.064545  ...  1.808121e+02 -1.034962e+03
PAY_6                          1.322472  ...  1.181210e+02  3.426237e+02
BILL_AMT1                  17560.424872  ...  2.034048e+08  1.826164e+08
BILL_AMT2                  18573.527165  ...  1.888731e+08  1.643518e+08
BILL_AMT3                  19234.422476  ...  1.587478e+08  1.558003e+08
BILL_AMT4                  19705.551629  ...  3.398374e+08  1.312133e+08
BILL_AMT5                  20338.120325  ...  2.700805e+08  2.791830e+08
BILL_AMT6                  19524.880348  ...  2.451233e+08  2.334670e+08
PAY_AMT1                     -28.500666  ...  7.354626e+07  5.178189e+07
PAY_AMT2                    -138.399452  ...  9.929841e+07  6.501168e+07
PAY_AMT3                     118.121022  ...  3.100051e+08  5.966970e+07
PAY_AMT4                     342.623730  ...  5.966970e+07  2.454286e+08
PAY_AMT5                    -815.832688  ...  4.282921e+07  3.634098e+07
PAY_AMT6                    -517.216277  ...  5.093879e+07  4.395747e+07
default                        0.089194  ... -4.110763e+02 -3.695159e+02
education                      0.176146  ... -1.048066e+03 -7.646876e+02
EDUCATION_graduate school     -0.068759  ...  4.710144e+02  3.452550e+02
EDUCATION_high school          0.017612  ... -2.067872e+02 -1.366018e+02
EDUCATION_other               -0.005092  ...  4.859679e+01  2.126372e+00
EDUCATION_university           0.056240  ... -3.128240e+02 -2.107795e+02

                               PAY_AMT5      PAY_AMT6      default  \
credit                     4.305657e+08  5.065153e+08 -8267.551759
SEX                        1.245881e+01  2.405188e+01     0.008113
MARRIAGE                  -9.607709e+00 -6.162327e+01    -0.005273
AGE                        3.218052e+03  3.191904e+03     0.053143
PAY_0                     -9.991077e+02 -1.172194e+03     0.151499
PAY_2                     -6.784685e+02 -7.768350e+02     0.130960
```

```
PAY_3                       -6.557960e+02 -7.630260e+02     0.116867
PAY_4                       -5.999916e+02 -5.521373e+02     0.105115
PAY_5                       -5.771610e+02 -4.638926e+02     0.096020
PAY_6                       -8.158327e+02 -5.172163e+02     0.089194
BILL_AMT1                    1.879091e+08  2.347681e+08  -600.394108
BILL_AMT2                    1.717652e+08  2.204845e+08  -419.289137
BILL_AMT3                    1.904126e+08  2.247817e+08  -405.153680
BILL_AMT4                    1.576892e+08  2.031590e+08  -271.199885
BILL_AMT5                    1.315051e+08  1.774537e+08  -170.597447
BILL_AMT6                    2.799982e+08  1.222761e+08  -132.796294
PAY_AMT1                     3.756893e+07  5.469033e+07  -501.374552
PAY_AMT2                     6.368414e+07  6.456816e+07  -560.210740
PAY_AMT3                     4.282921e+07  5.093879e+07  -411.076284
PAY_AMT4                     3.634098e+07  4.395747e+07  -369.515887
PAY_AMT5                     2.334266e+08  4.207110e+07  -349.562530
PAY_AMT6                     4.207110e+07  3.160383e+08  -392.426415
default                     -3.495625e+02 -3.924264e+02     0.172276
education                   -6.550900e+02 -8.573975e+02     0.022934
EDUCATION_graduate school    3.447913e+02  4.259025e+02    -0.010180
EDUCATION_high school       -1.966421e+02 -2.277882e+02     0.004979
EDUCATION_other              1.400045e+01  3.526644e+01    -0.002351
EDUCATION_university        -1.621496e+02 -2.333807e+02     0.007552

                                education  EDUCATION_graduate school  \
credit                      -34930.604407               16044.482969   
SEX                             -0.018208                   0.005317   
MARRIAGE                        -0.062974                   0.035451   
AGE                             -0.261453                  -0.442349   
PAY_0                            0.184394                  -0.076644   
PAY_2                            0.238080                  -0.096806   
PAY_3                            0.224645                  -0.091629   
PAY_4                            0.209388                  -0.085144   
PAY_5                            0.187787                  -0.075112   
PAY_6                            0.176146                  -0.068759   
BILL_AMT1                     3406.142693                -846.084922   
BILL_AMT2                     3002.625797                -691.567207   
BILL_AMT3                     2225.753773                -434.516602   
BILL_AMT4                     1673.950355                -117.103564   
BILL_AMT5                     1256.256590                  27.342070   
BILL_AMT6                     1556.428952                 -71.869023   
PAY_AMT1                      -930.556016                 394.252555   
PAY_AMT2                     -1216.387136                 488.852307   
PAY_AMT3                     -1048.065641                 471.014411   
PAY_AMT4                      -764.687597                 345.254954   
PAY_AMT5                      -655.090046                 344.791256   
PAY_AMT6                      -857.397465                 425.902474   
default                          0.022934                  -0.010180   
education                        1.881439                  -0.563882   
EDUCATION_graduate school       -0.563882                   0.228350   
EDUCATION_high school           -0.098032                  -0.057831   
EDUCATION_other                  0.006270                  -0.005504   
EDUCATION_university             0.655644                  -0.165014   

                           EDUCATION_high school  EDUCATION_other  \
credit                              -6709.314796       215.785093   
SEX                                     0.001385        -0.000515   
MARRIAGE                               -0.021418        -0.000542   
AGE                                     0.789120         0.010260   
PAY_0                                   0.024505        -0.003473   
PAY_2                                   0.028625        -0.004913   
PAY_3                                   0.027674        -0.005107   
PAY_4                                   0.025268        -0.004491   
PAY_5                                   0.020798        -0.004047   
PAY_6                                   0.017612        -0.005092   
BILL_AMT1                            -599.858730       331.829532   
```

```
BILL_AMT2                      -599.693276       271.462376
BILL_AMT3                      -585.509355       248.814746
BILL_AMT4                      -744.846531       167.053398
BILL_AMT5                      -713.633173        88.983544
BILL_AMT6                      -683.020797        25.219712
PAY_AMT1                       -130.662666         9.123682
PAY_AMT2                       -142.222305        34.274825
PAY_AMT3                       -206.787191        48.596788
PAY_AMT4                       -136.601819         2.126372
PAY_AMT5                       -196.642088        14.000453
PAY_AMT6                       -227.788197        35.266436
default                           0.004979        -0.002351
education                        -0.098032         0.006270
EDUCATION_graduate school        -0.057831        -0.005504
EDUCATION_high school             0.137041        -0.002557
EDUCATION_other                  -0.002557         0.015357
EDUCATION_university             -0.076653        -0.007296

                           EDUCATION_university
credit                             -9550.953266
SEX                                   -0.006188
MARRIAGE                              -0.013490
AGE                                   -0.357031
PAY_0                                  0.055612
PAY_2                                  0.073094
PAY_3                                  0.069062
PAY_4                                  0.064367
PAY_5                                  0.058361
PAY_6                                  0.056240
BILL_AMT1                           1114.114120
BILL_AMT2                           1019.798107
BILL_AMT3                            771.211212
BILL_AMT4                            694.896696
BILL_AMT5                            597.307558
BILL_AMT6                            729.670109
PAY_AMT1                            -272.713571
PAY_AMT2                            -380.904827
PAY_AMT3                            -312.824009
PAY_AMT4                            -210.779507
PAY_AMT5                            -162.149622
PAY_AMT6                            -233.380713
default                                0.007552
education                              0.655644
EDUCATION_graduate school             -0.165014
EDUCATION_high school                 -0.076653
EDUCATION_other                       -0.007296
EDUCATION_university                   0.248963

[28 rows x 28 columns]
```

In [123…

```
df.cov()
```

Out[123…

	credit	SEX	MARRIAGE	AGE	PAY_0	
credit	1.683446e+10	-1571.050630	-7323.669658	173076.722569	-39545.930009	-46037.
SEX	-1.571051e+03	0.239247	0.008014	0.409726	0.031685	0.
MARRIAGE	-7.323670e+03	0.008014	0.272452	-1.992764	0.011683	0.
AGE	1.730767e+05	0.409726	-1.992764	84.969755	-0.408639	-0.
PAY_0	-3.954593e+04	0.031685	0.011683	-0.408639	1.262930	0.

	credit	SEX	MARRIAGE	AGE	PAY_0	
PAY_2	-4.603765e+04	0.041442	0.015122	-0.553408	0.904330	1.
PAY_3	-4.443225e+04	0.038694	0.020421	-0.585263	0.772384	1.
PAY_4	-4.057181e+04	0.034411	0.020213	-0.535851	0.707972	0.
PAY_5	-3.667056e+04	0.030521	0.021074	-0.562245	0.648743	0.
PAY_6	-3.509308e+04	0.024754	0.020616	-0.517022	0.613292	0.
BILL_AMT1	2.727020e+09	1211.694332	-902.154685	38172.933546	15480.304170	20706.
BILL_AMT2	2.570130e+09	1085.595467	-802.517866	35613.657962	15185.916919	20045.
BILL_AMT3	2.548533e+09	833.207432	-901.679085	34334.251320	14011.556537	18609.
BILL_AMT4	2.453926e+09	688.489572	-783.881599	30453.108180	12950.248389	17116.
BILL_AMT5	2.331481e+09	505.694333	-805.840875	27654.067800	12341.668685	16110.
BILL_AMT6	2.243837e+09	487.430160	-659.223347	26137.648547	11844.759724	15642.
PAY_AMT1	4.195711e+08	1.964266	-51.691615	3992.041735	-1475.495089	-1600.
PAY_AMT2	5.333504e+08	15.675500	-97.327974	4626.861549	-1815.138407	-1627.
PAY_AMT3	4.801180e+08	74.034812	-32.546082	4746.824393	-1396.168258	-1178.
PAY_AMT4	4.131202e+08	17.080110	-103.518204	3087.324192	-1126.847945	-878.
PAY_AMT5	4.305657e+08	12.458809	-9.607709	3218.052172	-999.107730	-678.
PAY_AMT6	5.065153e+08	24.051885	-61.623271	3191.903901	-1172.193614	-776.
default	-8.267552e+03	0.008113	-0.005273	0.053143	0.151499	0.
education	-3.493060e+04	-0.018208	-0.062974	-0.261453	0.184394	0.
EDUCATION_graduate school	1.604448e+04	0.005317	0.035451	-0.442349	-0.076644	-0.
EDUCATION_high school	-6.709315e+03	0.001385	-0.021418	0.789120	0.024505	0.
EDUCATION_other	2.157851e+02	-0.000515	-0.000542	0.010260	-0.003473	-0.
EDUCATION_university	-9.550953e+03	-0.006188	-0.013490	-0.357031	0.055612	0.

28 rows × 28 columns

In [ ]:
```
## corrrelation
```

In [ ]:
```
corrMat = df.corr()
print(corrMat)
### from course site
```

In [126…
```
df.corr()
```

Out[126…

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	P
credit	1.000000	-0.024755	-0.108139	0.144713	-0.271214	-0.296382	-0.286123	-0.2
SEX	-0.024755	1.000000	0.031389	0.090874	0.057643	0.070771	0.066096	0.0
MARRIAGE	-0.108139	0.031389	1.000000	-0.414170	0.019917	0.024199	0.032688	0.0
AGE	0.144713	0.090874	-0.414170	1.000000	-0.039447	-0.050148	-0.053048	-0.0
PAY_0	-0.271214	0.057643	0.019917	-0.039447	1.000000	0.672164	0.574245	0.5
PAY_2	-0.296382	0.070771	0.024199	-0.050148	0.672164	1.000000	0.766552	0.6
PAY_3	-0.286123	0.066096	0.032688	-0.053048	0.574245	0.766552	1.000000	0.7
PAY_4	-0.267460	0.060173	0.033122	-0.049722	0.538841	0.662067	0.777359	1.0
PAY_5	-0.249411	0.055064	0.035629	-0.053826	0.509426	0.622780	0.686775	0.8
PAY_6	-0.235195	0.044008	0.034345	-0.048773	0.474553	0.575501	0.632684	0.7
BILL_AMT1	0.285430	0.033642	-0.023472	0.056239	0.187068	0.234887	0.208473	0.2
BILL_AMT2	0.278314	0.031183	-0.021602	0.054283	0.189859	0.235257	0.237295	0.2
BILL_AMT3	0.283236	0.024563	-0.024909	0.053710	0.179785	0.224146	0.227494	0.2
BILL_AMT4	0.293988	0.021880	-0.023344	0.051353	0.179125	0.222237	0.227202	0.2
BILL_AMT5	0.295562	0.017005	-0.025393	0.049345	0.180635	0.221348	0.225145	0.2
BILL_AMT6	0.290389	0.016733	-0.021207	0.047613	0.176980	0.219403	0.222327	0.2
PAY_AMT1	0.195236	0.000242	-0.005979	0.026147	-0.079269	-0.080701	0.001295	-0.0
PAY_AMT2	0.178408	0.001391	-0.008093	0.021785	-0.070101	-0.058990	-0.066793	-0.0
PAY_AMT3	0.210167	0.008597	-0.003541	0.029247	-0.070561	-0.055901	-0.053311	-0.0
PAY_AMT4	0.203242	0.002229	-0.012659	0.021379	-0.064005	-0.046858	-0.046067	-0.0
PAY_AMT5	0.217202	0.001667	-0.001205	0.022850	-0.058190	-0.037093	-0.035863	-0.0
PAY_AMT6	0.219595	0.002766	-0.006641	0.019478	-0.058673	-0.036500	-0.035861	-0.0
default	-0.153520	0.039961	-0.024339	0.013890	0.324794	0.263551	0.235253	0.2
education	-0.196273	-0.027139	-0.087956	-0.020678	0.119623	0.144983	0.136838	0.1
EDUCATION_graduate school	0.258777	0.022750	0.142129	-0.100423	-0.142720	-0.169215	-0.160209	-0.1
EDUCATION_high school	-0.139686	0.007650	-0.110845	0.231252	0.058902	0.064590	0.062461	0.0
EDUCATION_other	0.013420	-0.008498	-0.008386	0.008982	-0.024937	-0.033118	-0.034435	-0.0
EDUCATION_university	-0.147530	-0.025353	-0.051797	-0.077626	0.099177	0.122364	0.115644	0.1

28 rows × 28 columns

In [ ]:

based on correlatin coefficients

select some columns/features for pairplots and regression

after regression, maybe make another round of selection

target: credit

features: default, AGE, MARRIAGE, EDUCATION_graduate school

features: PAY_2, BILL_AMT5, PAY_AMT6

df_S selected

In [127…
```
df_S = df[['credit','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2',
```

In [128…
```
df_S.head()
```

Out[128…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
0	20000	1	24	1	0	2	0	0
1	120000	1	26	2	0	2	3455	2000
2	90000	0	34	2	0	0	14948	5000
3	50000	0	37	1	0	0	28959	1000
4	50000	0	57	1	0	0	19146	679

In [131…
```
df_S.corr()
```

Out[131…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BIL
credit	1.000000	-0.153520	0.144713	-0.108139	0.258777	-0.296382	(
default	-0.153520	1.000000	0.013890	-0.024339	-0.051328	0.263551	-(
AGE	0.144713	0.013890	1.000000	-0.414170	-0.100423	-0.050148	(
MARRIAGE	-0.108139	-0.024339	-0.414170	1.000000	0.142129	0.024199	-(
EDUCATION_graduate school	0.258777	-0.051328	-0.100423	0.142129	1.000000	-0.169215	(
PAY_2	-0.296382	0.263551	-0.050148	0.024199	-0.169215	1.000000	(
BILL_AMT5	0.295562	-0.006760	0.049345	-0.025393	0.000941	0.221348	1
PAY_AMT6	0.219595	-0.053183	0.019478	-0.006641	0.050135	-0.036500	(

In [ ]:
```
## https://seaborn.pydata.org/generated/seaborn.heatmap.html
import numpy as np; np.random.seed(0)
import seaborn as sns; sns.set_theme()
```

```
df_S = np.random.rand(10, 12)
ax = sns.heatmap(df_S)
```

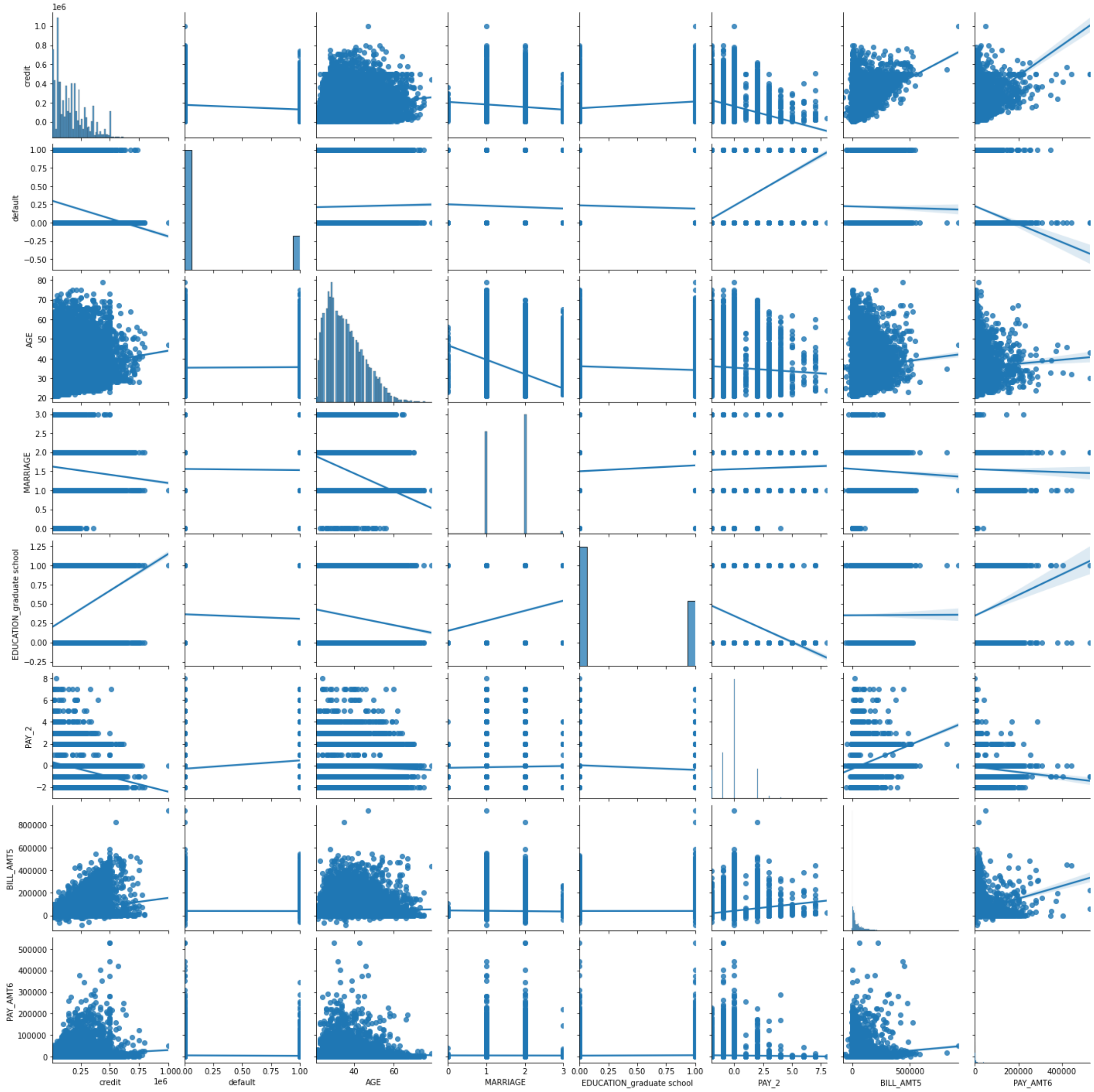
pairplots

In [ ]:
```
### Pairplots in Python
### https://github.com/WillKoehrsen/Data-Analysis/blob/master/pairplots/Pair%20Plots.ip

# Pandas and numpy for data manipulation
import pandas as pd
import numpy as n

# matplotlib for plotting
import matplotlib.pyplot as plt
import matplotlib

# Seaborn for pairplots
import seaborn as sns
```

In [ ]:
```
sns.pairplot(df_S);
```

In [ ]:
```
sns.pairplot(df_S, hue = 'default');
```

In [129…
```
sns.pairplot(df_S, kind='reg');
```



regression

https://gist.github.com/rafiag

In [134...

```
### 1.
# Import libraries
## Basic libs

import pandas as pd
import numpy as np
import warnings

## Building Model
from sklearn import linear_model
from scipy import stats
import statsmodels
import statsmodels.api as sm
```

```
import statsmodels.formula.api as smf
import statsmodels.stats.api as sms
from statsmodels.compat import lzip

## Data Visualization
import seaborn as sns
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

warnings.filterwarnings('ignore')
## plt.rcParams['figure.figsize'] = (7, 7)
## plt.style.use('ggplot')
```

In [141…

```
df.columns
```

Out[141…

```
Index(['credit', 'SEX', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
       'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default', 'education',
       'EDUCATION_graduate school', 'EDUCATION_high school', 'EDUCATION_other',
       'EDUCATION_university'],
      dtype='object')
```

In [142…

```
df_S = df[['credit','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2',
```

In [143…

```
df_S.columns
```

Out[143…

```
Index(['credit', 'default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school',
       'PAY_2', 'BILL_AMT5', 'PAY_AMT6'],
      dtype='object')
```

In [ ]:

```
# Visualize the data using scatter plot and histogram
sns.set_palette('colorblind')
sns.pairplot(data=df_S, height=3)
```

In [146…

```
df_S.head()
```

Out[146…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
0	20000	1	24	1	0	2	0	0
1	120000	1	26	2	0	2	3455	2000
2	90000	0	34	2	0	0	14948	5000
3	50000	0	37	1	0	0	28959	1000
4	50000	0	57	1	0	0	19146	679

In [147…

```
# Set independent and dependent variables
X = df_S[['default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2', 'BILL_AMT
y = df_S['credit']

# Initialize model from sklearn and fit it into our data
regr = linear_model.LinearRegression()
model = regr.fit(X, y)

print('Intercept:', model.intercept_)
print('Coefficients:', model.coef_)
```

```
Intercept: 94769.42275890156
Coefficients: [-1.72921329e+04  1.38762803e+03 -2.06113069e+04  5.93366413e+04
 -3.32335318e+04  7.11394467e-01  1.00209978e+00]
```

y head = 94769 - default*17292 + age*1387 - marriage*20611 + graduateschool*59336 -pay_2*33233 + billant*50.7 + payamt6*1.0

In [158…

```
### Model Validation

X = df_S[['default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2', 'BILL_AMT
X = sm.add_constant(X) # adding a constant

olsmod = sm.OLS(df_S['credit'], X).fit()
print(olsmod.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 credit   R-squared:                       0.308
Model:                            OLS   Adj. R-squared:                  0.308
Method:                 Least Squares   F-statistic:                     1906.
Date:                Fri, 28 May 2021   Prob (F-statistic):               0.00
Time:                        01:46:12   Log-Likelihood:            -3.9025e+05
No. Observations:               30000   AIC:                         7.805e+05
Df Residuals:                   29992   BIC:                         7.806e+05
Df Model:                           7
Covariance Type:            nonrobust
====================================================================================
=====
                               coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------------
-----
const                     9.477e+04   4038.899     23.464      0.000    8.69e+04     1.0
3e+05
default                  -1.729e+04   1562.626    -11.066      0.000   -2.04e+04    -1.4
2e+04
AGE                       1387.6280     74.613     18.598      0.000    1241.384     153
3.872
MARRIAGE                 -2.061e+04   1320.929    -15.604      0.000   -2.32e+04     -1.
8e+04
EDUCATION_graduate school  5.934e+04   1342.881     44.186      0.000    5.67e+04      6.
2e+04
PAY_2                    -3.323e+04    565.902    -58.727      0.000   -3.43e+04    -3.2
1e+04
BILL_AMT5                    0.7114      0.011     66.289      0.000       0.690
0.732
PAY_AMT6                     1.0021      0.036     28.074      0.000       0.932
1.072
==============================================================================
Omnibus:                     4706.972   Durbin-Watson:                   1.947
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7734.171
Skew:                           1.063   Prob(JB):                         0.00
Kurtosis:                       4.291   Cond. No.                     4.90e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
ed.
[2] The condition number is large, 4.9e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [159…

```
print('R2 score:', olsmod.rsquared)
```

```
R2 score: 0.3078377586089446
```

In [160…
```
print('F-statistic:', olsmod.fvalue)
print('Probability of observing value at least as high as F-statistic:', olsmod.f_pvalu
```

```
F-statistic: 1905.5544875152689
Probability of observing value at least as high as F-statistic: 0.0
```

In [161…
```
## Because our f_pvalue is lower than 0.05 we can conclude that our model performs bett
```

In [162…
```
print(olsmod.pvalues)
```

```
const                      1.160412e-120
default                     2.080158e-28
AGE                         9.066584e-77
MARRIAGE                    1.128408e-54
EDUCATION_graduate school   0.000000e+00
PAY_2                       0.000000e+00
BILL_AMT5                   0.000000e+00
PAY_AMT6                   3.350195e-171
dtype: float64
```

In [ ]:
```
## if independent variables have p-value less than 0.05
## it will show that there is sufficient evidence that they affects our credit
```

regression Assumption Testing

In [163…
```
df_S['credit_pred'] = olsmod.predict(X)
df_S['residual'] = olsmod.resid
df_S.head()


### Residual is the difference between the observed value and predicted value from our
##### With statsmodel we can easily get the residual value of our model by simply acces
### .resid attribute of the model and then we can keep it in a new column called 'resid
```

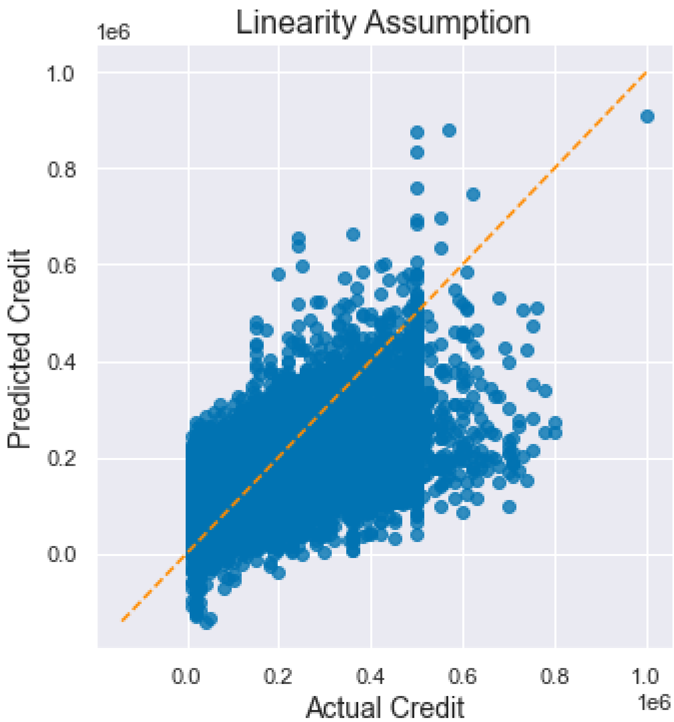
Out[163…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6	credit_p
0	20000	1	24	1	0	2	0	0	23701.992
1	120000	1	26	2	0	2	3455	2000	10328.008
2	90000	0	34	2	0	0	14948	5000	116370.585
3	50000	0	37	1	0	0	28959	1000	147103.725
4	50000	0	57	1	0	0	19146	679	167553.697

In [ ]:

In [165…
```
## Linearity test

# Plotting the observed vs predicted values
sns.lmplot(x='credit', y='credit_pred', data=df_S, fit_reg=False, size=5)

# Plotting the diagonal line
line_coords = np.arange(df_S[['credit', 'credit_pred']].min().min()-10,
                        df_S[['credit', 'credit_pred']].max().max()+10)
plt.plot(line_coords, line_coords,  # X and y points
         color='darkorange', linestyle='--')
```

```
plt.ylabel('Predicted Credit', fontsize=14)
plt.xlabel('Actual Credit', fontsize=14)
plt.title('Linearity Assumption', fontsize=16)
plt.show()
```



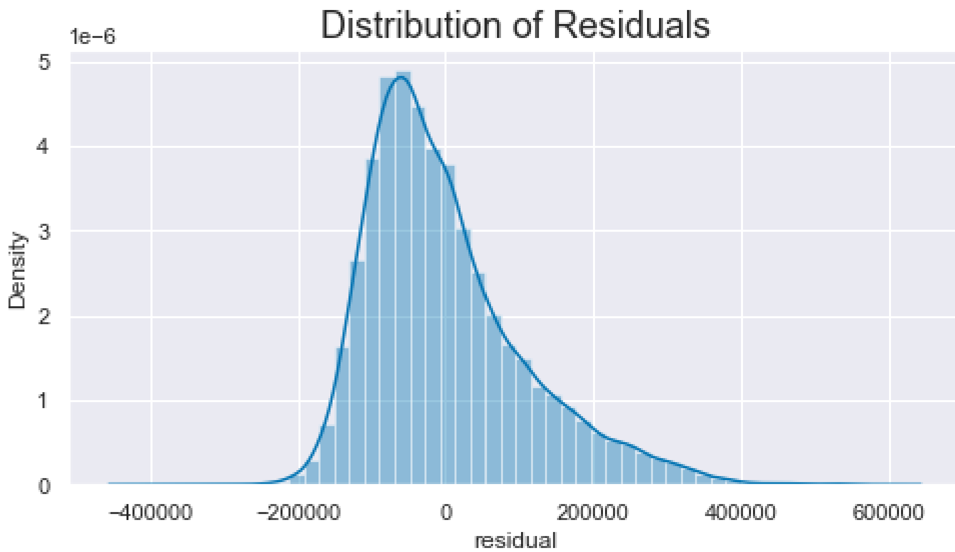
In [ ]:

```
## The scatter plots show residual point sort of evenly spread around the diagonal line
## so we can assume that there is linear relationship between our independent and depen
```

In [166…

```
## Normality test

from statsmodels.stats.diagnostic import normal_ad

# Performing the test on the residuals
p_value = normal_ad(df_S['residual'])[1]
print('p-value from the test Anderson-Darling test below 0.05 generally means non-norma

# Plotting the residuals distribution
plt.subplots(figsize=(8, 4))
plt.title('Distribution of Residuals', fontsize=18)
sns.distplot(df_S['residual'])
plt.show()

# Reporting the normality of the residuals
if p_value < 0.05:
    print('Residuals are not normally distributed')
else:
    print('Residuals are normally distributed')
```

```
p-value from the test Anderson-Darling test below 0.05 generally means non-normal: 0.0
```



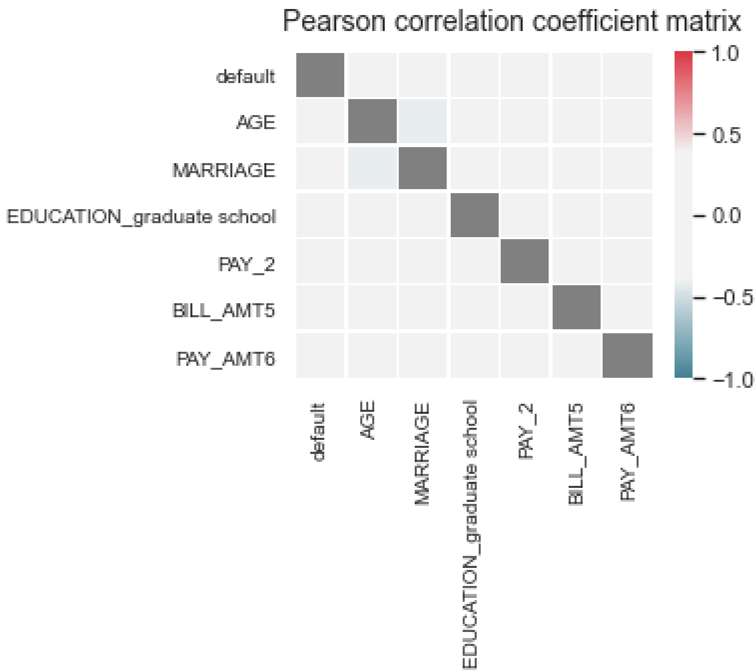

```
Residuals are not normally distributed
```

In [167…

```
## Multicollinearity test


corr = df_S[['default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2', 'BILL_
print('Pearson correlation coefficient matrix of each variables:\n', corr)

# Generate a mask for the diagonal cell
mask = np.zeros_like(corr, dtype=np.bool)
np.fill_diagonal(mask, val=True)

# Initialize matplotlib figure
fig, ax = plt.subplots(figsize=(4, 3))

# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True, sep=100)
cmap.set_bad('grey')

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmin=-1, vmax=1, center=0, linewidths=.5)
fig.suptitle('Pearson correlation coefficient matrix', fontsize=14)
ax.tick_params(axis='both', which='major', labelsize=10)
```

```
Pearson correlation coefficient matrix of each variables:
                              default       AGE  MARRIAGE  \
default                     1.000000  0.013890 -0.024339
AGE                         0.013890  1.000000 -0.414170
MARRIAGE                   -0.024339 -0.414170  1.000000
EDUCATION_graduate school -0.051328 -0.100423  0.142129
PAY_2                       0.263551 -0.050148  0.024199
BILL_AMT5                  -0.006760  0.049345 -0.025393
PAY_AMT6                   -0.053183  0.019478 -0.006641

                            EDUCATION_graduate school     PAY_2  BILL_AMT5  \
default                                     -0.051328  0.263551  -0.006760
AGE                                         -0.100423 -0.050148   0.049345
MARRIAGE                                     0.142129  0.024199  -0.025393
EDUCATION_graduate school                    1.000000 -0.169215   0.000941
PAY_2                                       -0.169215  1.000000   0.221348
BILL_AMT5                                    0.000941  0.221348   1.000000
PAY_AMT6                                     0.050135 -0.036500   0.164184

                            PAY_AMT6
```

```
default                    -0.053183
AGE                         0.019478
MARRIAGE                   -0.006641
EDUCATION_graduate school   0.050135
PAY_2                      -0.036500
BILL_AMT5                   0.164184
PAY_AMT6                    1.000000
```



In [ ]:
```
## almost 0 correlation coefficient, means independent variable are not affecting one o
## and that there is no multicollinearity in our data.
```

In [168…
```
## Autocorrelation

from statsmodels.stats.stattools import durbin_watson

durbinWatson = durbin_watson(df_S['residual'])

print('Durbin-Watson:', durbinWatson)
if durbinWatson < 1.5:
    print('Signs of positive autocorrelation', '\n')
    print('Assumption not satisfied')
elif durbinWatson > 2.5:
    print('Signs of negative autocorrelation', '\n')
    print('Assumption not satisfied')
else:
    print('Little to no autocorrelation', '\n')
    print('Assumption satisfied')
```

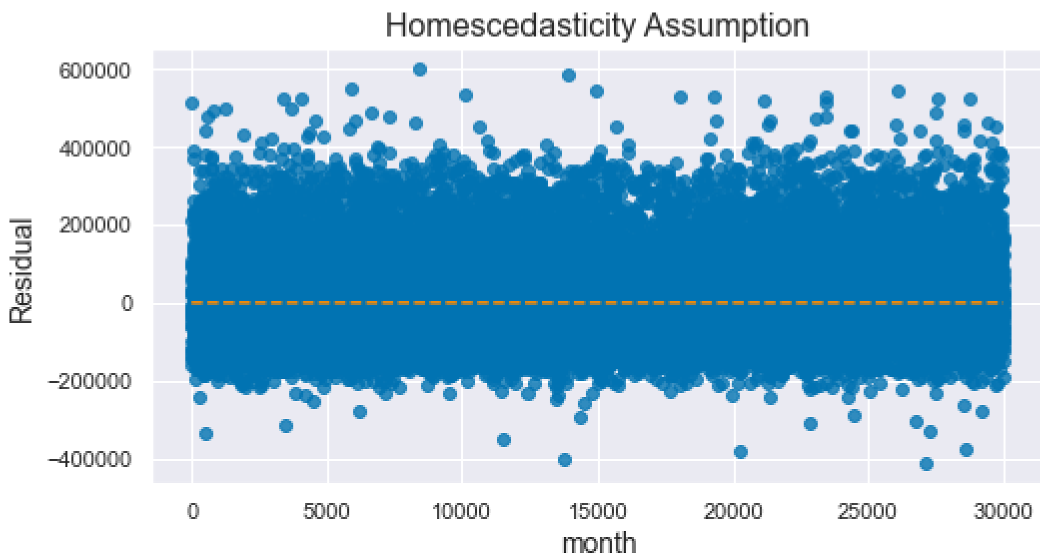
```
Durbin-Watson: 1.9466529704375113
Little to no autocorrelation

Assumption satisfied
```

In [ ]:
```
## Our model got a Durbin-Watson score of about 1.94 which is between 1.5 and 2.5,
## so we can assume that there is no autocorrelation in our residual.
```

In [ ]:
```
## This assumes homoscedasticity, which is the same variance within our error terms.
## Heteroscedasticity, the violation of homoscedasticity, occurs when we don’t have an
## To detect homoscedasticity, we can plot our residual and see if the variance appears
```

In [169…

```
##Homoscedasticity test

# Plotting the residuals
plt.subplots(figsize=(8, 4))
plt.scatter(x=df_S.index, y=df_S.residual, alpha=0.8)
plt.plot(np.repeat(0, len(df_S.index)+2), color='darkorange', linestyle='--')

plt.ylabel('Residual', fontsize=14)
plt.xlabel('month', fontsize=14)
plt.title('Homescedasticity Assumption', fontsize=16)
plt.show()
```



In [ ]:

```
## assume that it satisfied the homoscedasticity assumption.
```

after regression: r squared 0.3 -- not bad

line 158-159

In [170…

```
df_S.head()
```

Out[170…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6	credit_p
0	20000	1	24	1	0	2	0	0	23701.992
1	120000	1	26	2	0	2	3455	2000	10328.008
2	90000	0	34	2	0	0	14948	5000	116370.585
3	50000	0	37	1	0	0	28959	1000	147103.725
4	50000	0	57	1	0	0	19146	679	167553.697

In [175…

```
df_S.head()
```

Out[175…

credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6	credit_p

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6	credit_p
0	20000	1	24	1	0	2	0	0	23701.992
1	120000	1	26	2	0	2	3455	2000	10328.008
2	90000	0	34	2	0	0	14948	5000	116370.585
3	50000	0	37	1	0	0	28959	1000	147103.725
4	50000	0	57	1	0	0	19146	679	167553.697

add variables to see the change of R squared

In [182…

```
df = pd.read_csv('creditEDA.csv')
```

In [183…

```
df.head()
```

Out[183…

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	PAY_AMT3	PAY_AM
0	20000	0	1	24	2	2	-1	-1	-2	-2	...	0	
1	120000	0	2	26	-1	2	0	0	0	2	...	1000	1
2	90000	0	2	34	0	0	0	0	0	0	...	1000	1
3	50000	0	1	37	0	0	0	0	0	0	...	1200	1
4	50000	1	1	57	-1	0	-1	0	0	0	...	10000	9

5 rows × 28 columns

add SEX

In [184…

```
df_S = df[['credit', 'SEX','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', '
```

In [185…

```
df_S.head()
```

Out[185…

	credit	SEX	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
0	20000	0	1	24	1	0	2	0	0
1	120000	0	1	26	2	0	2	3455	2000
2	90000	0	0	34	2	0	0	14948	5000
3	50000	0	0	37	1	0	0	28959	1000
4	50000	1	0	57	1	0	0	19146	679

In [186…

```
X = df_S[['SEX','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2', 'BI
X = sm.add_constant(X) # adding a constant
```

```
olsmod = sm.OLS(df_S['credit'], X).fit()
print(olsmod.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 credit   R-squared:                       0.308
Model:                            OLS   Adj. R-squared:                  0.308
Method:                 Least Squares   F-statistic:                     1670.
Date:                Fri, 28 May 2021   Prob (F-statistic):               0.00
Time:                        02:31:04   Log-Likelihood:            -3.9024e+05
No. Observations:               30000   AIC:                         7.805e+05
Df Residuals:                   29991   BIC:                         7.806e+05
Df Model:                           8
Covariance Type:            nonrobust
====================================================================================
=====
                             coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------------
-----
const                   9.487e+04   4038.063     23.494      0.000     8.7e+04     1.0
3e+05
SEX                    -4939.2710   1287.980     -3.835      0.000   -7463.767    -241
4.775
default                -1.717e+04   1562.619    -10.985      0.000   -2.02e+04    -1.4
1e+04
AGE                     1422.0011     75.132     18.927      0.000    1274.739     156
9.264
MARRIAGE               -2.025e+04   1324.042    -15.292      0.000   -2.28e+04    -1.7
7e+04
EDUCATION_graduate school  5.953e+04   1343.548     44.310      0.000    5.69e+04     6.2
2e+04
PAY_2                  -3.308e+04    567.242    -58.312      0.000   -3.42e+04     -3.
2e+04
BILL_AMT5                  0.7112      0.011     66.284      0.000       0.690
0.732
PAY_AMT6                   1.0026      0.036     28.094      0.000       0.933
1.073
==============================================================================
Omnibus:                     4728.419   Durbin-Watson:                   1.946
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7787.897
Skew:                           1.066   Prob(JB):                         0.00
Kurtosis:                       4.298   Cond. No.                     4.90e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
ed.
[2] The condition number is large, 4.9e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [187…

```
print('R2 score:', olsmod.rsquared)
```

```
R2 score: 0.3081770020939534
```

model 1: without SEX: R2: 0.3078377586089446

model 2: + SEX: R2: 0.3081770020939534

not much change

In [188…

```
df.head()
```

Out[188…

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	PAY_AMT3	PAY_AM
0	20000	0	1	24	2	2	-1	-1	-2	-2	...	0	
1	120000	0	2	26	-1	2	0	0	0	2	...	1000	1
2	90000	0	2	34	0	0	0	0	0	0	...	1000	1
3	50000	0	1	37	0	0	0	0	0	0	...	1200	1
4	50000	1	1	57	-1	0	-1	0	0	0	...	10000	9

5 rows × 28 columns

add two education dummies

In [189…

```
df_S = df[['credit', 'SEX','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', '
           'PAY_2', 'BILL_AMT5', 'PAY_AMT6']]
```

In [190…

```
df_S.head()
```

Out[190…

	credit	SEX	default	AGE	MARRIAGE	EDUCATION_graduate school	EDUCATION_university	EDUCATION_hi sch
0	20000	0	1	24	1	0	1	
1	120000	0	1	26	2	0	1	
2	90000	0	0	34	2	0	1	
3	50000	0	0	37	1	0	1	
4	50000	1	0	57	1	0	1	

In [208…

```
# Set independent and dependent variables
X = df[['SEX','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'EDUCATION_uni
           'PAY_2', 'BILL_AMT5', 'PAY_AMT6']]
y = df['credit']

# Initialize model from sklearn and fit it into our data
regr = linear_model.LinearRegression()
model = regr.fit(X, y)

print('Intercept:', model.intercept_)
print('Coefficients:', model.coef_)
```

```
Intercept: 100620.31169254212
Coefficients: [-5.07633975e+03 -1.69636848e+04  1.64434563e+03 -1.99148083e+04
  4.61235888e+04 -7.56340268e+03 -3.34748934e+04 -3.27799398e+04
  7.03603198e-01  9.94020314e-01]
```

In [209…

```
X = df_S[['SEX','default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'EDUCATION_u
           'PAY_2', 'BILL_AMT5', 'PAY_AMT6']]
X = sm.add_constant(X) # adding a constant
```

```
olsmod = sm.OLS(df_S['credit'], X).fit()
print(olsmod.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                 credit   R-squared:                       0.313
Model:                            OLS   Adj. R-squared:                  0.313
Method:                 Least Squares   F-statistic:                     1366.
Date:                Fri, 28 May 2021   Prob (F-statistic):               0.00
Time:                        23:25:16   Log-Likelihood:            -3.9014e+05
No. Observations:               30000   AIC:                         7.803e+05
Df Residuals:                   29989   BIC:                         7.804e+05
Df Model:                          10                                         
Covariance Type:            nonrobust                                         
==================================================================================
=====
                              coef    std err          t      P>|t|      [0.025      
0.975]
----------------------------------------------------------------------------------
-----
const                    1.006e+05   6386.692     15.755      0.000    8.81e+04     1.1
3e+05
SEX                     -5076.3398   1283.602     -3.955      0.000   -7592.255    -256
0.424
default                 -1.696e+04   1558.319    -10.886      0.000      -2e+04    -1.3
9e+04
AGE                      1644.3456     76.494     21.496      0.000    1494.413     179
4.278
MARRIAGE                -1.991e+04   1319.758    -15.090      0.000   -2.25e+04    -1.7
3e+04
EDUCATION_graduate school  4.612e+04   5088.038      9.065      0.000    3.62e+04     5.6
1e+04
EDUCATION_university    -7563.4027   5066.132     -1.493      0.135   -1.75e+04     236
6.435
EDUCATION_high school   -3.347e+04   5226.536     -6.405      0.000   -4.37e+04    -2.3
2e+04
PAY_2                   -3.278e+04    566.011    -57.914      0.000   -3.39e+04    -3.1
7e+04
BILL_AMT5                   0.7036      0.011     65.721      0.000       0.683       
0.725
PAY_AMT6                    0.9940      0.036     27.944      0.000       0.924       
1.064
==============================================================================
Omnibus:                     4730.222   Durbin-Watson:                   1.951
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7819.340
Skew:                           1.064   Prob(JB):                         0.00
Kurtosis:                       4.315   Cond. No.                     1.20e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
ed.
[2] The condition number is large, 1.2e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [210…

```
print('R2 score:', olsmod.rsquared)   ## previous two models --R2-- 0.31
```

```
R2 score: 0.312963286258l2797
```

model 1: without SEX: R2: 0.3078377586089446

model 2: + SEX: R2: 0.3081770020939534

model 3: + SEX, 'EDUCATION_university', 'EDUCATION_high school': R2: 0.31296328625812797

which features to keep for model buiding -- data for C2T3

use model 1. I think it a parsimonious model

In [213…
```
df = pd.read_csv('creditEDA.csv')
```

In [214…
```
df.head()
```

Out[214…

	credit	SEX	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	PAY_AMT3	PAY_AN
0	20000	0	1	24	2	2	-1	-1	-2	-2	...	0	
1	120000	0	2	26	-1	2	0	0	0	2	...	1000	1
2	90000	0	2	34	0	0	0	0	0	0	...	1000	1
3	50000	0	1	37	0	0	0	0	0	0	...	1200	1
4	50000	1	1	57	-1	0	-1	0	0	0	...	10000	9

5 rows × 28 columns

In [217…
```
df_S = df[['credit', 'default', 'AGE', 'MARRIAGE', 'EDUCATION_graduate school', 'PAY_2'
```

In [218…
```
df_S.head()
```

Out[218…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
0	20000	1	24	1	0	2	0	0
1	120000	1	26	2	0	2	3455	2000
2	90000	0	34	2	0	0	14948	5000
3	50000	0	37	1	0	0	28959	1000
4	50000	0	57	1	0	0	19146	679

In [219…
```
df_S.to_csv('creditML.csv', index = False)
```

In [220…
```
df = pd.read_csv('creditML.csv')
```

In [221…
```
df.head()
```

Out[221…

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
0	20000	1	24	1	0	2	0	0
1	120000	1	26	2	0	2	3455	2000

	credit	default	AGE	MARRIAGE	EDUCATION_graduate school	PAY_2	BILL_AMT5	PAY_AMT6
2	90000	0	34	2	0	0	14948	5000
3	50000	0	37	1	0	0	28959	1000
4	50000	0	57	1	0	0	19146	679

In [ ]:

```

```

In [ ]:

```
## guides
### 1. course site

### 2. the Titanic EDA example: https://github.com/TarekDib03/titanic-EDA/blob/master/T

### 3. Multi-Linear Regression Using Python
### https://medium.com/swlh/multi-linear-regression-using-python-44bd0d10082d
### https://gist.github.com/rafiag

#### 4. Tutorial: Exploratory Data Analysis (EDA) with Categorical Variables |
#### https://github.com/hoffm386/eda-with-categorical-variables
###  https://medium.com/analytics-vidhya/tutorial-exploratory-data-analysis-eda-with-ca

### 5. Pairplots in Python
#### https://github.com/WillKoehrsen/Data-Analysis/blob/master/pairplots/Pair%20Plots.i

### 6. https://seaborn.pydata.org/generated/seaborn.lineplot.html
```