Preprocessing and Data Visualization

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv("loan_data.csv")
```

In [3]:
```
df.head(2)
```

Out[3]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.c
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	5639.9
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	2760.0

In [4]:
```
df.tail(2)
```

Out[4]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.v
9576	0	home_improvement	0.1600	351.58	10.819778	19.18	692	
9577	0	debt_consolidation	0.1392	853.43	11.264464	16.28	732	

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

Out[5]:
```
(9578, 14)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   credit.policy      9578 non-null   int64  
 1   purpose            9578 non-null   object 
 2   int.rate           9578 non-null   float64
 3   installment        9578 non-null   float64
 4   log.annual.inc     9578 non-null   float64
 5   dti                9578 non-null   float64
 6   fico               9578 non-null   int64  
 7   days.with.cr.line  9578 non-null   float64
 8   revol.bal          9578 non-null   int64  
 9   revol.util         9578 non-null   float64
 10  inq.last.6mths     9578 non-null   int64  
 11  delinq.2yrs        9578 non-null   int64  
 12  pub.rec            9578 non-null   int64  
 13  not.fully.paid     9578 non-null   int64  
dtypes: float64(6), int64(7), object(1)
memory usage: 1.0+ MB
```


In [7]:
```
print("missing values count:")
print(df.isnull().sum())
```

```
missing values count:
credit.policy        0
purpose              0
int.rate             0
installment          0
log.annual.inc       0
dti                  0
fico                 0
days.with.cr.line    0
revol.bal            0
revol.util           0
inq.last.6mths       0
delinq.2yrs          0
pub.rec              0
not.fully.paid       0
dtype: int64
```

In [8]:
```
df.shape[0]
```

Out[8]:
```
9578
```

In [9]:
```
print("missing values count:")
print(df.isnull().sum()/df.shape[0]*100)
```

```
missing values count:
credit.policy        0.0
purpose              0.0
int.rate             0.0
installment          0.0
log.annual.inc       0.0
dti                  0.0
fico                 0.0
days.with.cr.line    0.0
revol.bal            0.0
revol.util           0.0
inq.last.6mths       0.0
delinq.2yrs          0.0
pub.rec              0.0
not.fully.paid       0.0
dtype: float64
```

In [10]:
```
print("check the duplicate vales")
print(df.duplicated().sum())
```

```
check the duplicate vales
0
```

In [11]:
```
for i in df.select_dtypes(include='object').columns:
    print(df[i].value_counts())
    print("****"*10)
```

```
purpose
debt_consolidation    3957
all_other             2331
credit_card           1262
home_improvement       629
small_business         619
major_purchase         437
educational            343
Name: count, dtype: int64
****************************************
```

In [12]:
```
df.describe().T
```

Out[12]:

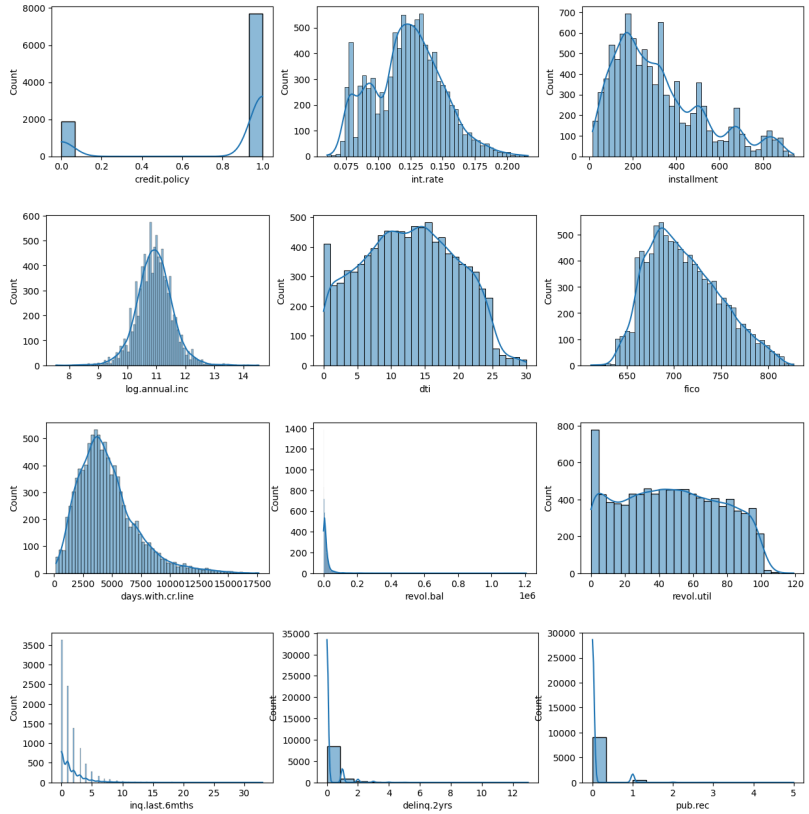
	count	mean	std	min	25%	50%
credit.policy	9578.0	0.804970	0.396245	0.000000	1.000000	1.000000
int.rate	9578.0	0.122640	0.026847	0.060000	0.103900	0.122100
installment	9578.0	319.089413	207.071301	15.670000	163.770000	268.950000
log.annual.inc	9578.0	10.932117	0.614813	7.547502	10.558414	10.928884
dti	9578.0	12.606679	6.883970	0.000000	7.212500	12.665000
fico	9578.0	710.846314	37.970537	612.000000	682.000000	707.000000
days.with.cr.line	9578.0	4560.767197	2496.930377	178.958333	2820.000000	4139.958333
revol.bal	9578.0	16913.963876	33756.189557	0.000000	3187.000000	8596.000000
revol.util	9578.0	46.799236	29.014417	0.000000	22.600000	46.300000
inq.last.6mths	9578.0	1.577469	2.200245	0.000000	0.000000	1.000000
delinq.2yrs	9578.0	0.163708	0.546215	0.000000	0.000000	0.000000
pub.rec	9578.0	0.062122	0.262126	0.000000	0.000000	0.000000
not.fully.paid	9578.0	0.160054	0.366676	0.000000	0.000000	0.000000

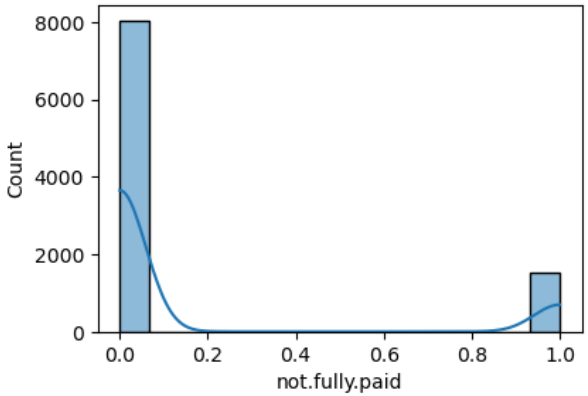
In [13]:
```
df.describe(include="object").T
```

Out[13]:

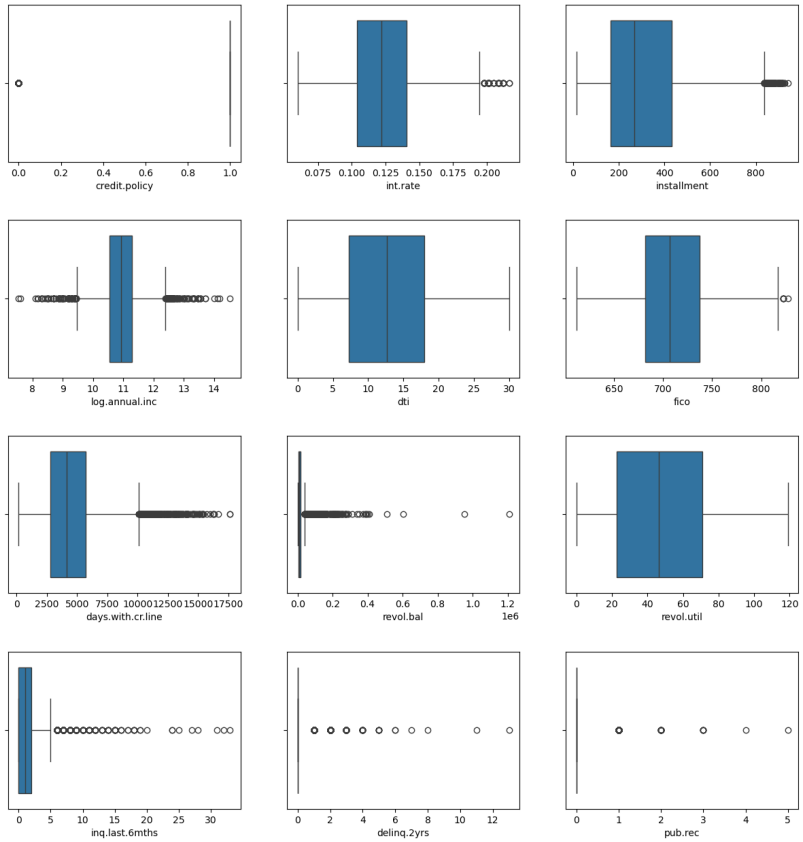
	count	unique	top	freq
purpose	9578	7	debt_consolidation	3957

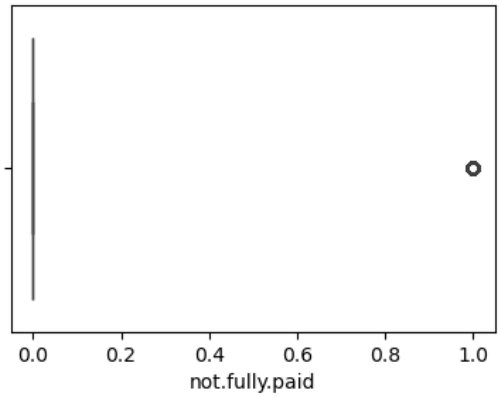
In [14]:
```
import warnings
warnings.filterwarnings("ignore")

f = 1
plt.figure(figsize=(15,3))

for i in df.select_dtypes(include="number").columns:
    plt.subplot(1,3,f)
    sns.histplot(data=df,x=i,kde=True)
    if f<3:
        f += 1
    else:
        f = 1
        plt.show()
        plt.figure(figsize=(15,3))
```



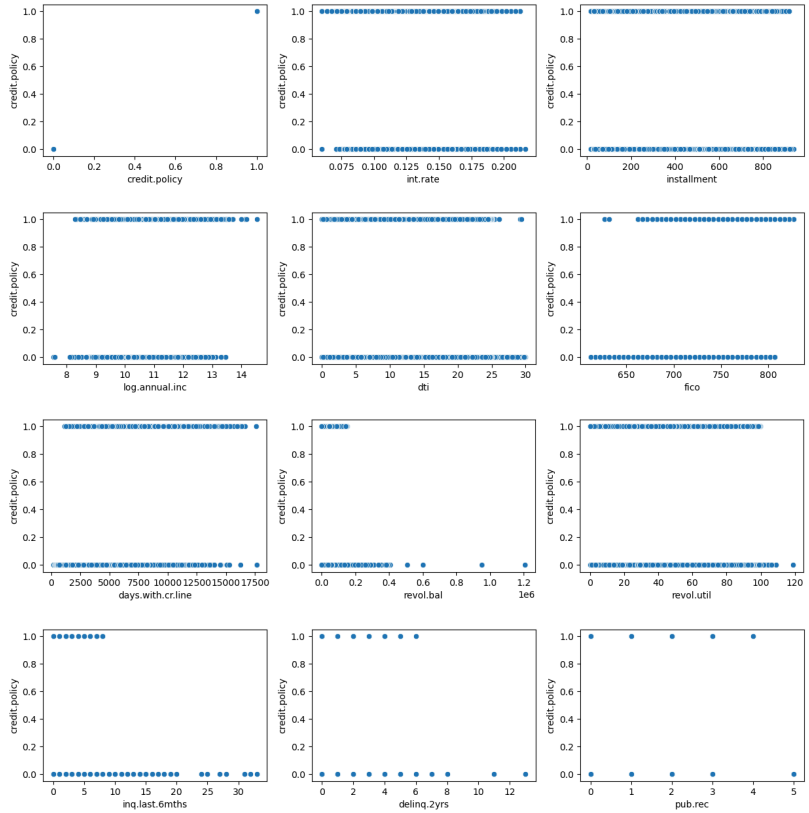


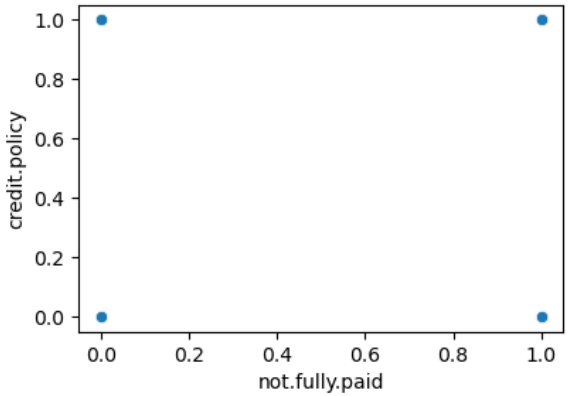
In [15]:
```
f = 1
plt.figure(figsize=(15,3))

for i in df.select_dtypes(include="number").columns:
    plt.subplot(1,3,f)
    sns.boxplot(data=df,x=i)
    if f<3:
        f += 1
    else:
        f = 1
        plt.show()
        plt.figure(figsize=(15,3))
```





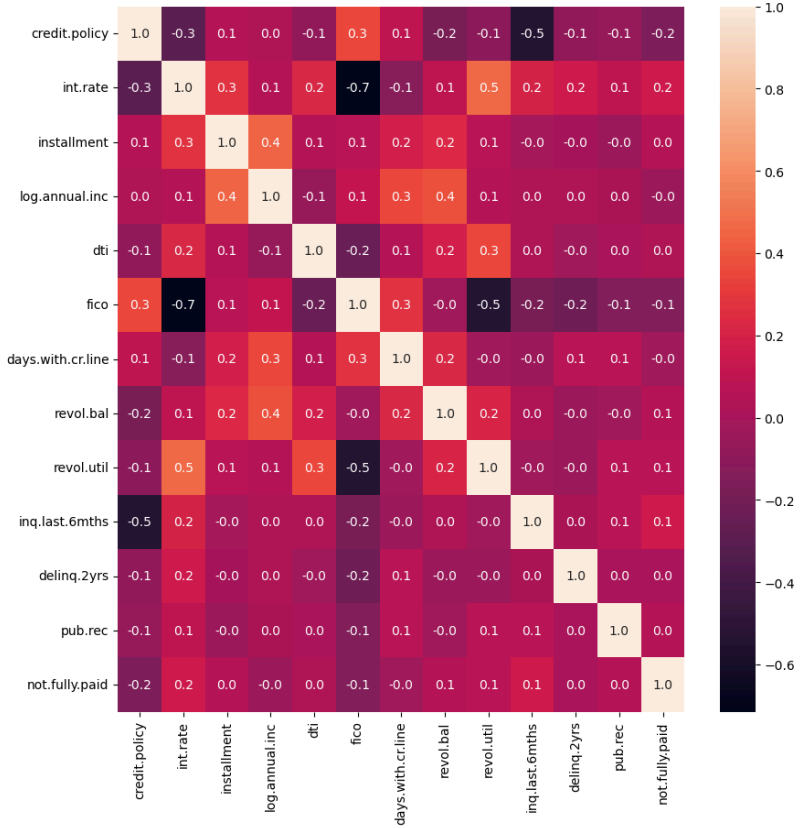
In [16]:
```
f = 1
plt.figure(figsize=(15,3))

for i in['credit.policy', 'int.rate', 'installment', 'log.annual.inc', 'dti
       'fico', 'days.with.cr.line', 'revol.bal', 'revol.util',
       'inq.last.6mths', 'delinq.2yrs', 'pub.rec', 'not.fully.paid']:
    plt.subplot(1,3,f)
    sns.scatterplot(data=df,x=i,y='credit.policy')
    if f<3:
        f += 1
    else:
        f = 1
        plt.show()
        plt.figure(figsize=(15,3))
```





In [17]:
```
s=df.select_dtypes(include="number").corr()
plt.figure(figsize=(10,10))
sns.heatmap(s,annot=True,fmt='0.1f')
```

Out[17]: <Axes: >



In [18]:
```
df.isnull().sum()
```

Out[18]:
```
credit.policy        0
purpose              0
int.rate             0
installment          0
log.annual.inc       0
dti                  0
fico                 0
days.with.cr.line    0
revol.bal            0
revol.util           0
inq.last.6mths       0
delinq.2yrs          0
pub.rec              0
not.fully.paid       0
dtype: int64
```

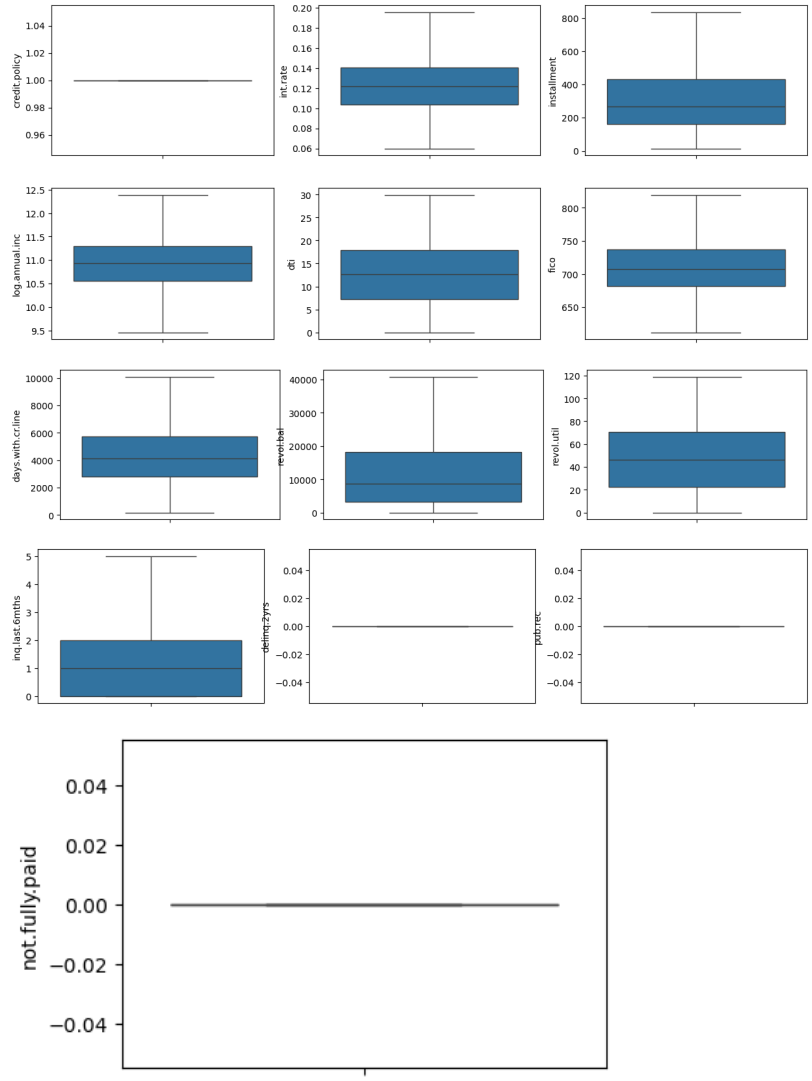
In [19]:
```
def wisker(col):
    q1,q3=np.percentile(col,[25,75])
    iqr=q3-q1
    lw=q1-1.5*iqr
    uw=q3+1.5*iqr
    return lw,uw
```

In [20]:
```
wisker(df['pub.rec'])
```

Out[20]: (0.0, 0.0)

In [21]:
```
for i in ['credit.policy', 'int.rate', 'installment', 'log.annual.inc', 'dt
       'fico', 'days.with.cr.line', 'revol.bal', 'revol.util',
       'inq.last.6mths', 'delinq.2yrs', 'pub.rec', 'not.fully.paid']:# Outl
    lw,uw=wisker(df[i])
    df[i]=np.where(df[i]<lw,lw,df[i])
    df[i]=np.where(df[i]>uw,uw,df[i])
```

In [22]:
```
f = 1
plt.figure(figsize=(15,3))
for i in  ['credit.policy', 'int.rate', 'installment', 'log.annual.inc', 'd
       'fico', 'days.with.cr.line', 'revol.bal', 'revol.util',
       'inq.last.6mths', 'delinq.2yrs', 'pub.rec', 'not.fully.paid']:# Remo
    plt.subplot(1,3,f)
    sns.boxplot(df[i])
    if f<3:
        f += 1
    else:
        f = 1
        plt.show()
        plt.figure(figsize=(15,3))
```



In [23]:
```
df = df.drop_duplicates()
```

In [24]:
```
mydata=pd.get_dummies(data=df,columns=["purpose"],drop_first=True,dtype='in
mydata.head(2)
```

Out[24]:

	credit.policy	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal
0	1.0	0.1189	829.10	11.350407	19.48	737.0	5639.958333	28854.0
1	1.0	0.1071	228.22	11.082143	14.29	707.0	2760.000000	33623.0

In [25]:
```
X=mydata.drop('credit.policy',axis=1)
y=mydata['credit.policy']
```

In [26]:
```
X.head(2)
```

Out[26]:

	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.
0	0.1189	829.10	11.350407	19.48	737.0	5639.958333	28854.0	52.1	
1	0.1071	228.22	11.082143	14.29	707.0	2760.000000	33623.0	76.7	

In [27]:
```
y[:3]
```

Out[27]:
```
0    1.0
1    1.0
2    1.0
Name: credit.policy, dtype: float64
```

In [ ]: