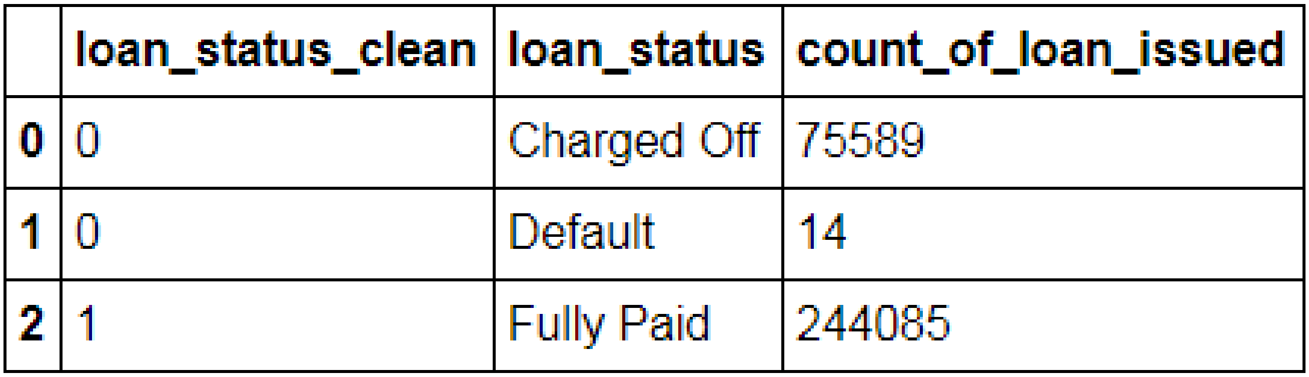
In our attempt to build a predictive model around the Lending Club dataset with the use of Loan related features, it was important to know the relationship of these features to our target variable, that is loan-status. We constructed SQL queries using these input variables and made some important preliminary anaysis below.

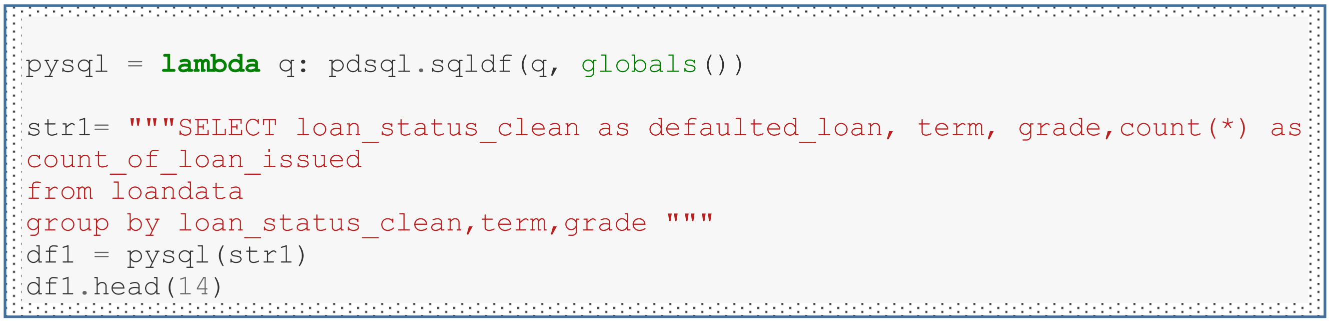
I wanted to see the number of loans issues per loan_status. I wrote below query and took the output in a dataframe df1.

```
pysql = lambda q: pdsql.sqldf(q, globals())
str1= """SELECT loan_status_clean
,loan_status
,count(*) as count_of_loan_issued
from loandata
group by loan_status_clean,loan_status """
df1 = pysql(str1)
df1.head(7)
```



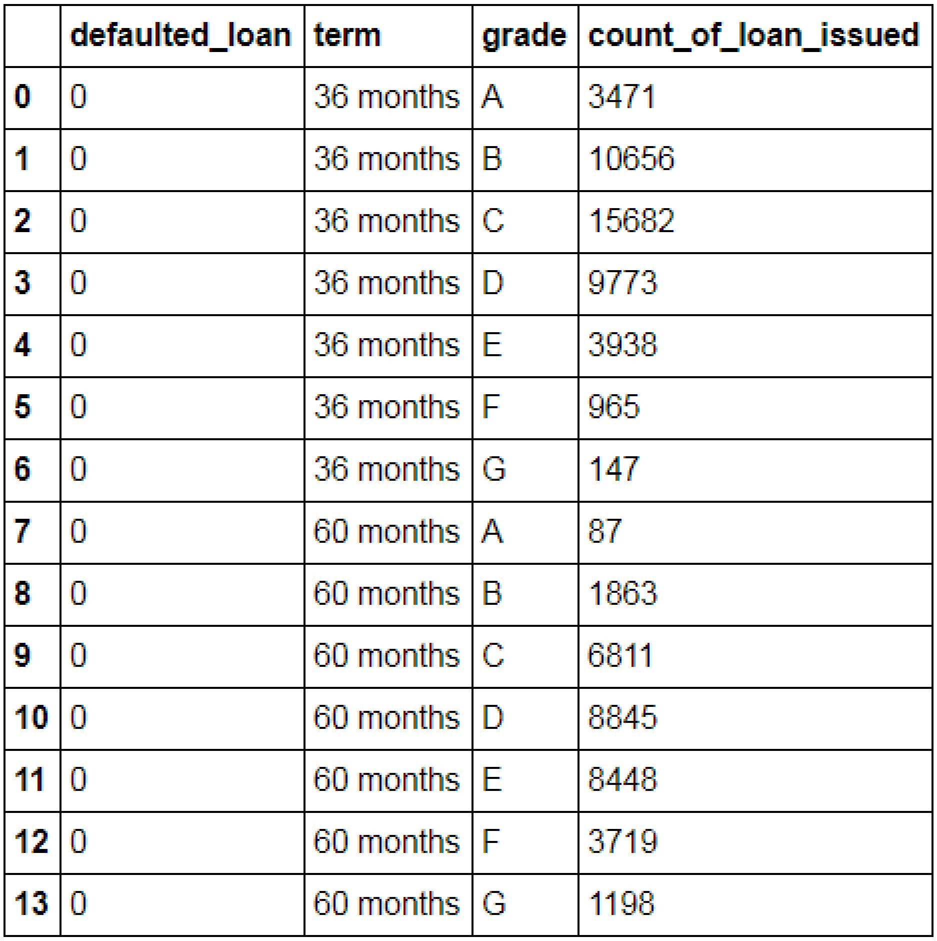
	loan_status_clean	loan_status	count_of_loan_issued
0	0	Charged Off	75589
1	0	Default	14
2	1	Fully Paid	244085

I wanted to see count of loans issues per grade, term and get the results grouped by good(0) and bad(0)loans.



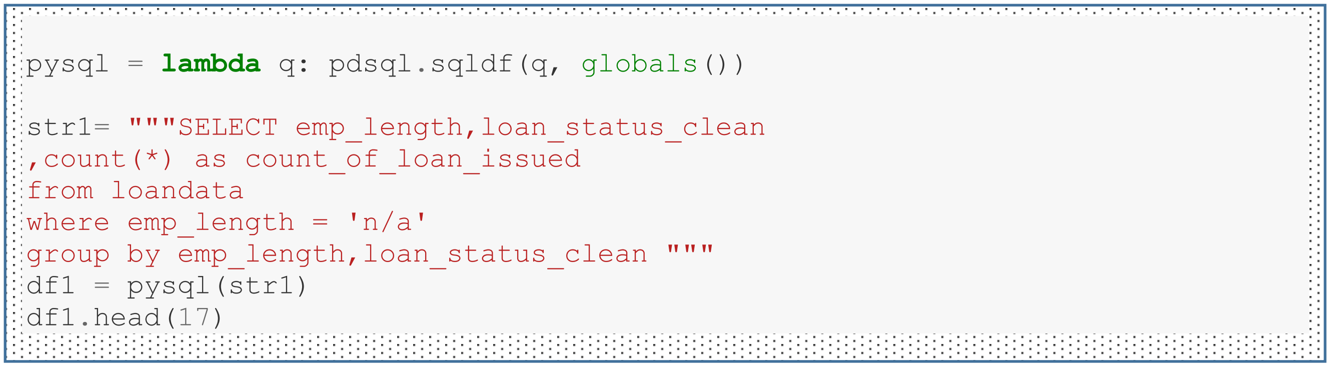
```
pysql = lambda q: pdsql.sqldf(q, globals())

str1= """SELECT loan_status_clean as defaulted_loan, term, grade,count(*) as
count_of_loan_issued
from loandata
group by loan_status_clean,term,grade """
df1 = pysql(str1)
df1.head(14)
```



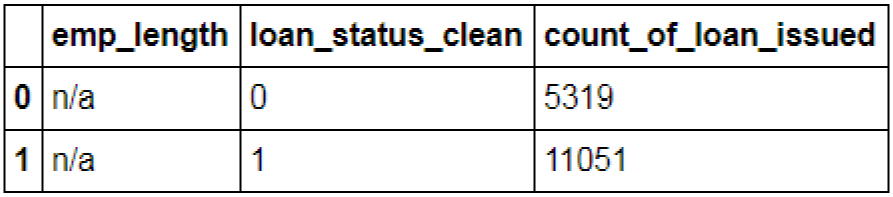
	defaulted_loan	term	grade	count_of_loan_issued
0	0	36 months	A	3471
1	0	36 months	B	10656
2	0	36 months	C	15682
3	0	36 months	D	9773
4	0	36 months	E	3938
5	0	36 months	F	965
6	0	36 months	G	147
7	0	60 months	A	87
8	0	60 months	B	1863
9	0	60 months	C	6811
10	0	60 months	D	8845
11	0	60 months	E	8448
12	0	60 months	F	3719
13	0	60 months	G	1198

I noticed there are quite a bunch of n/a in this column (emp_length). Before cleaning or eliminating them I was interested in seeing it’s spread over data. I wanted to see how many bad loans are being assigned as “n/a”. I constructed below query and I discovered half of the data which are assigned as 0 turned out to be bad loans.



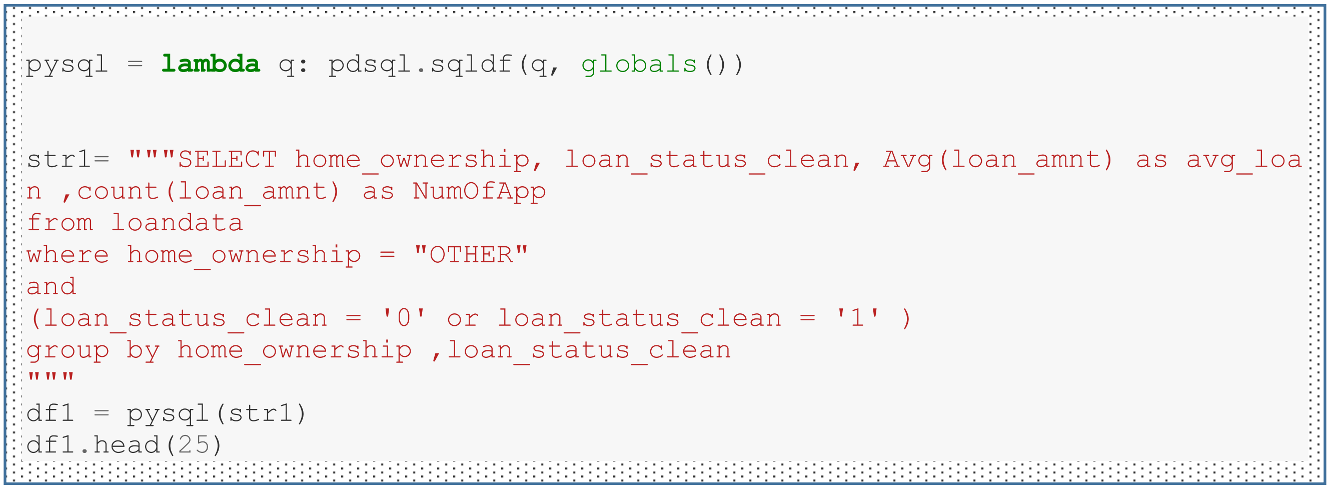
```
pysql = lambda q: pdsql.sqldf(q, globals())

str1= """SELECT emp_length,loan_status_clean
,count(*) as count_of_loan_issued
from loandata
where emp_length = 'n/a'
group by emp_length,loan_status_clean """
df1 = pysql(str1)
df1.head(17)
```



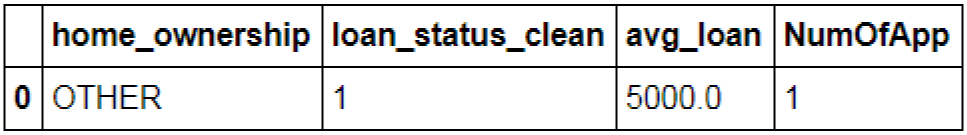
	emp_length	loan_status_clean	count_of_loan_issued
0	n/a	0	5319
1	n/a	1	11051

Next , like above I wanted to see how many records accounted for home_ownership = “Other” and it’s contribution towards bad loans. I found only 1 record in the dataset which was set as “Others” and it was a fully paid loan, hence it was not useful for our analysis and it was okay to eliminate it.



```
pysql = lambda q: pdsql.sqldf(q, globals())


str1= """SELECT home_ownership, loan_status_clean, Avg(loan_amnt) as avg_loa
n ,count(loan_amnt) as NumOfApp
from loandata
where home_ownership = "OTHER"
and
(loan_status_clean = '0' or loan_status_clean = '1' )
group by home_ownership ,loan_status_clean
"""
df1 = pysql(str1)
df1.head(25)
```



	home_ownership	loan_status_clean	avg_loan	NumOfApp
0	OTHER	1	5000.0	1