Gramener Case Study

PARTNERSHIP - SEEDS & GRAMENER JOIN HANDS

SEEDS is a not-for-profit organization that helps make communities resilient through comprehensive interventions in the areas of disaster readiness, response, and rehabilitation.

Gramener has partnered with SEEDS and Microsoft to develop the Sunny Lives AI model under Microsoft's global program 'AI for Humanitarian Action.

This predictive model aids in planning better risk reduction strategies against worsening climate emergencies and disasters.

Gramener Case Study

SAVING ONE HOUSE AT A TIME BEFORE DISASTER STRIK

PROBLEM SOLUTION IMPACT

When disasters like floods and heatwaves occur, warnings and other risk-related information are often vague and not up to date. M

There was a need to localize the risk information down to a neighborhood scale. This would help to mobilize targeted immediate response and build long-term resilience in the vulnerable communities. SEEDS wanted to automate, scale, and code their vast experience accumulated over the decades responding to various disasters at the ground level.

The SEEDS Sunny Lives model is a proactive attempt to address this challenge. The solution can deliver granular risk scores down to house level. This means that authorities can warn the high-risk houses earlier, allowing them to safeguard their property and reduce the damage caused. For instance, residents can take pre-emptive measures such as moving to the top floor of the house or evacuating to a safe site. Such actions reduce the damage, lowering the overall time and cost of recovery.

Gramener Case Study

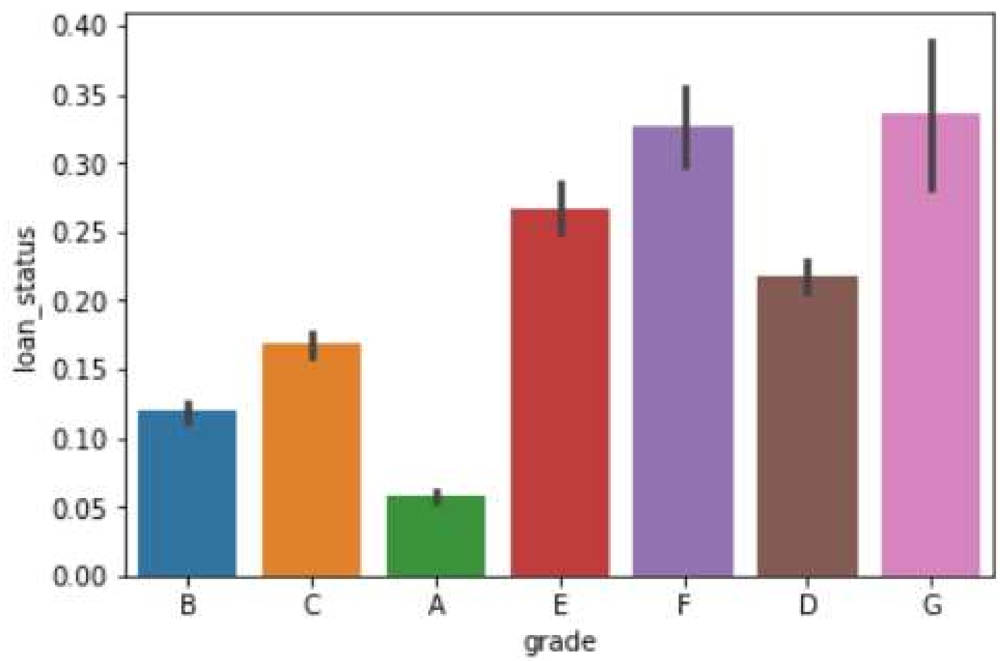
We are going to work with the loan data

LoanStatNew	Description
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
acc_open_past_24mths	Number of trades opened in past 24 months.
addr_state	The state provided by the borrower in the loan application
all_util	Balance to credit limit on all trades
annual_inc	The self-reported annual income provided by the borrower during registration.
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
avg_cur_bal	Average current balance of all accounts
bc_open_to_buy	Total open to buy on revolving bankcards.
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
chargeoff_within_12_mths	Number of charge-offs within 12 months
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
desc	Loan description provided by the borrower
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income
earliest_cr_line	The month the borrower's earliest reported credit line was opened
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
emp_title	The job title supplied by the Borrower when applying for the loan.*
fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
fico_range_low	The lower boundary range the borrower's FICO at loan origination belongs to.
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
grade	LC assigned loan grade
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
id	A unique LC assigned ID for the loan listing.
il_util	Ratio of total current balance to high credit/credit limit on all install acct
initial_list_status	The initial listing status of the loan. Possible values are – W, F

Gramener Case Study

First visualise the average default rates across categorical variables.

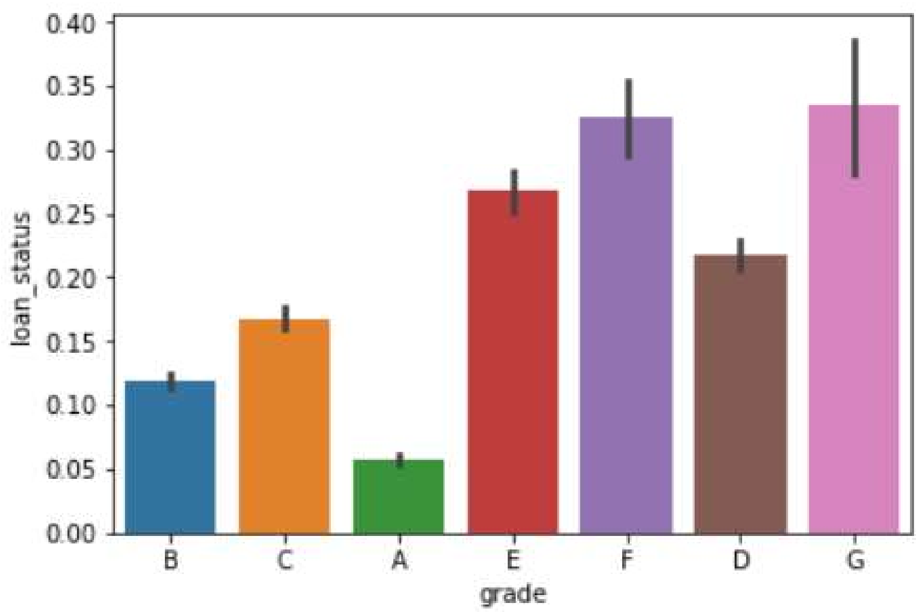
```python
# plotting default rates across grade of the loan
sns.barplot(x='grade', y='loan_status', data=df)
plt.show()
```



Gramener Case Study

compare default rates across grade of loans.

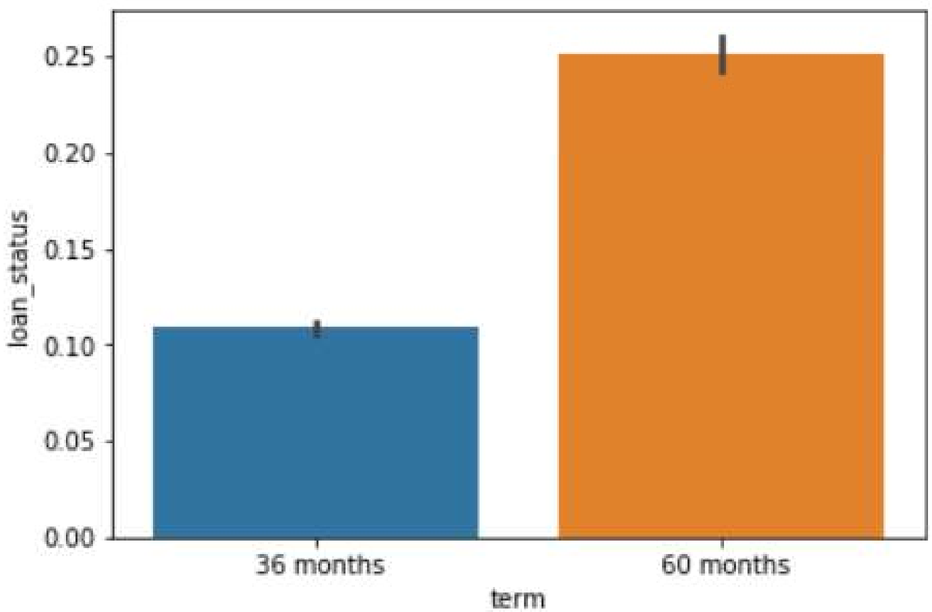
```
# compare default rates across grade of loan
plot_cat('grade')
```



Gramener Case Study

60 months loans default more than 36 months loans

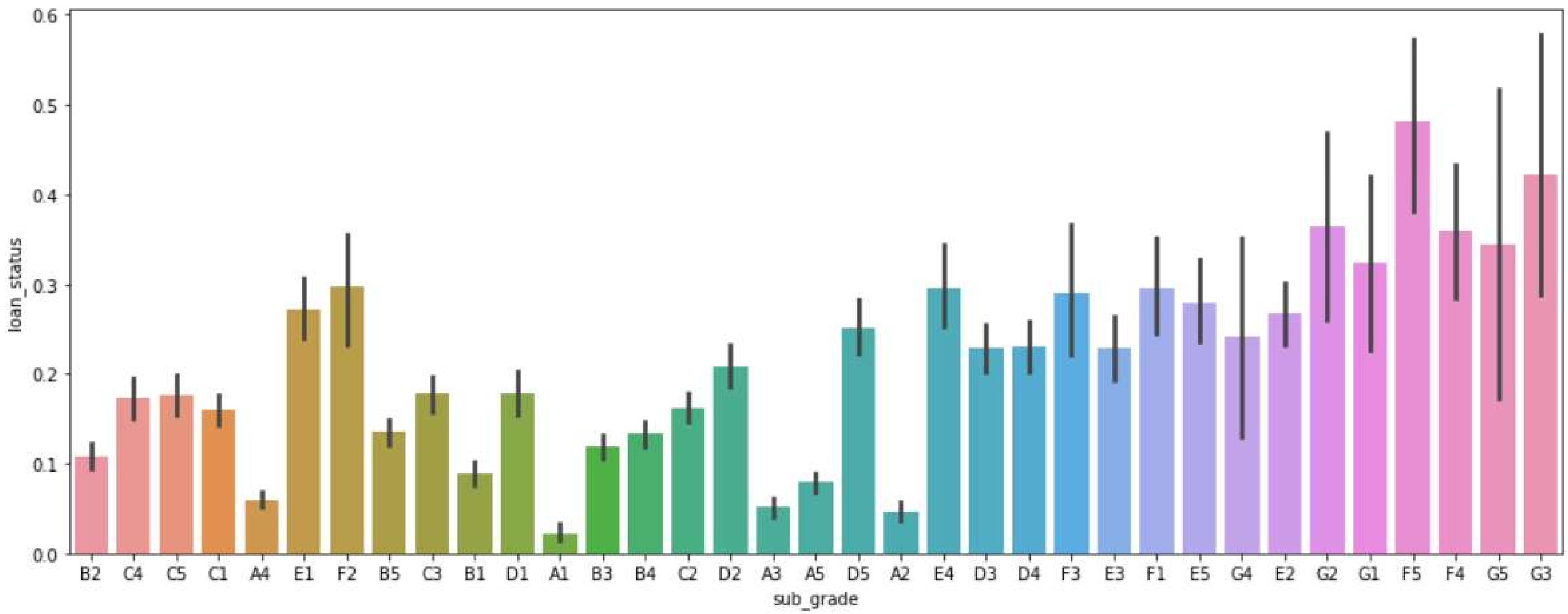
```
# term: 60 months loans default more than 36 months loans
plot_cat('term')
```



Gramener Case Study

As expected – A1 is better than A2 better than A3 and so on

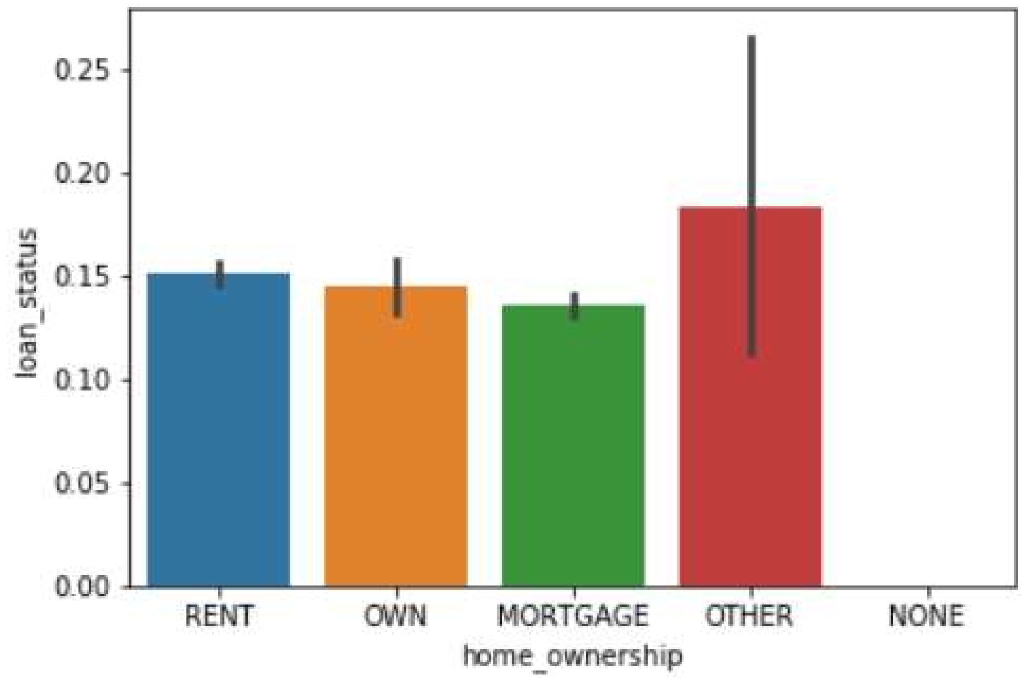
```
# sub-grade: as expected - A1 is better than A2 better than A3 and so on
plt.figure(figsize=(16, 6))
plot_cat('sub_grade')
```



Gramener Case Study

Not a great discriminator

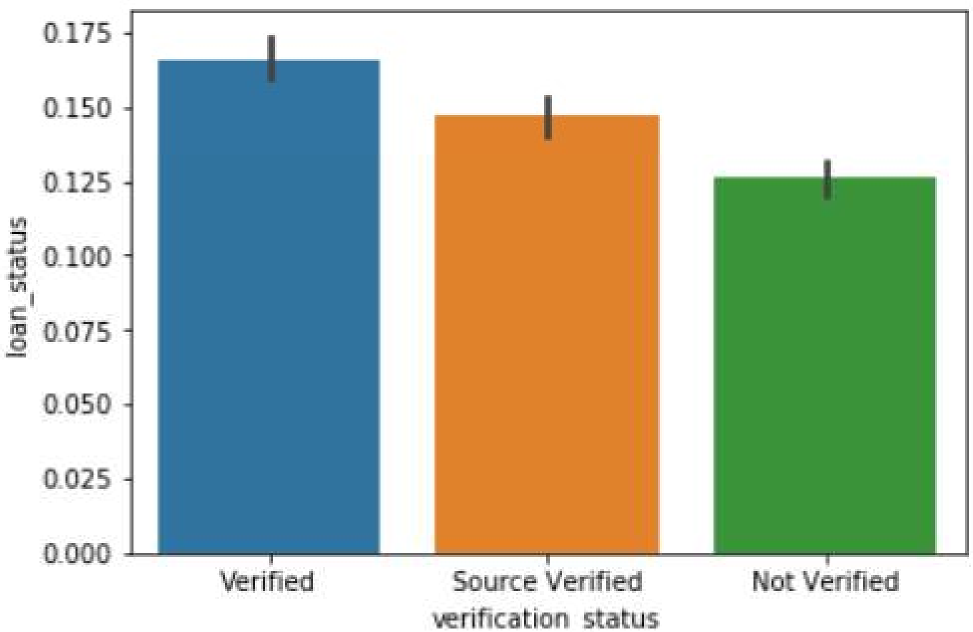
```
# home ownership: not a great discriminator
plot_cat('home_ownership')
```



Gramener Case Study

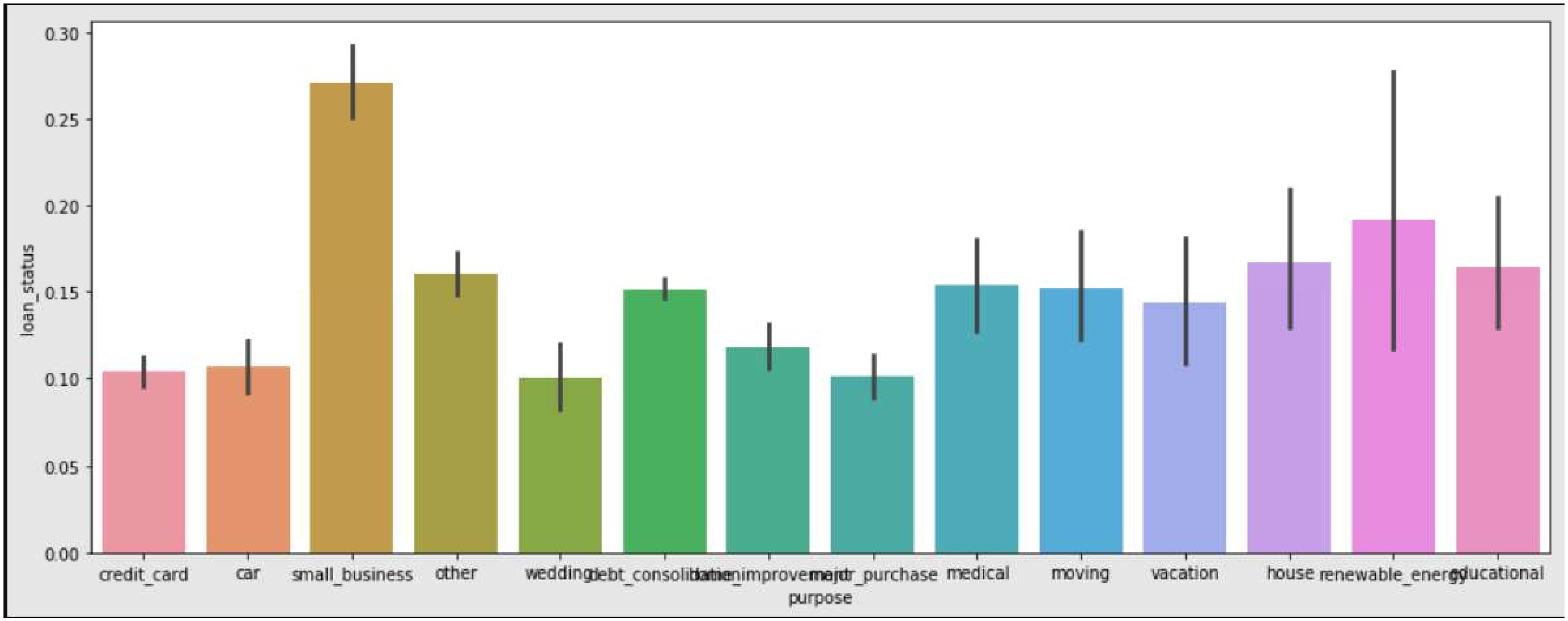
Surprisingly, verified loans default more than not verified.

```
# verification_status: surprisingly, verified loans default more than not verified
plot_cat('verification_status')
```



Gramener Case Study

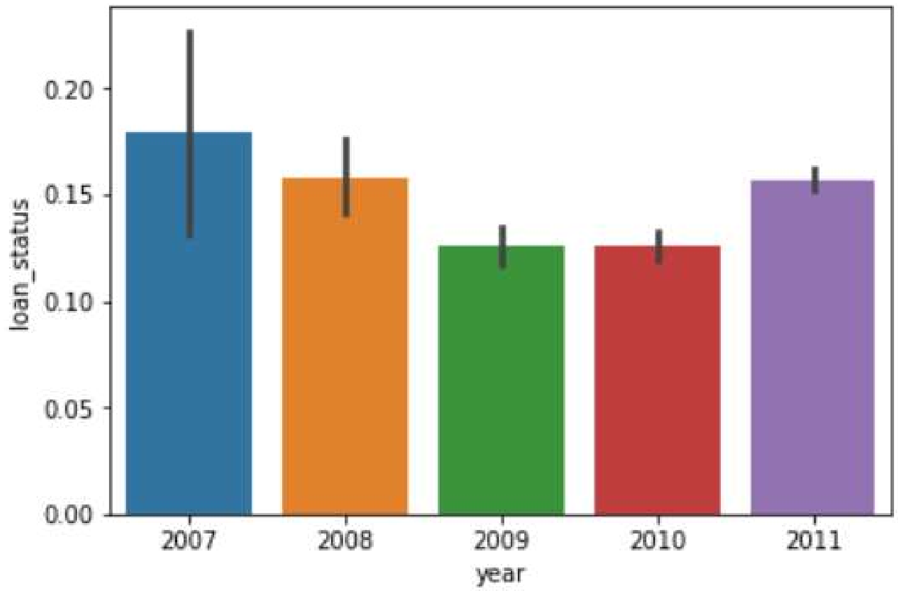
compare default rates across grade of loans.



Gramener Case Study

lets compare the default rates across years

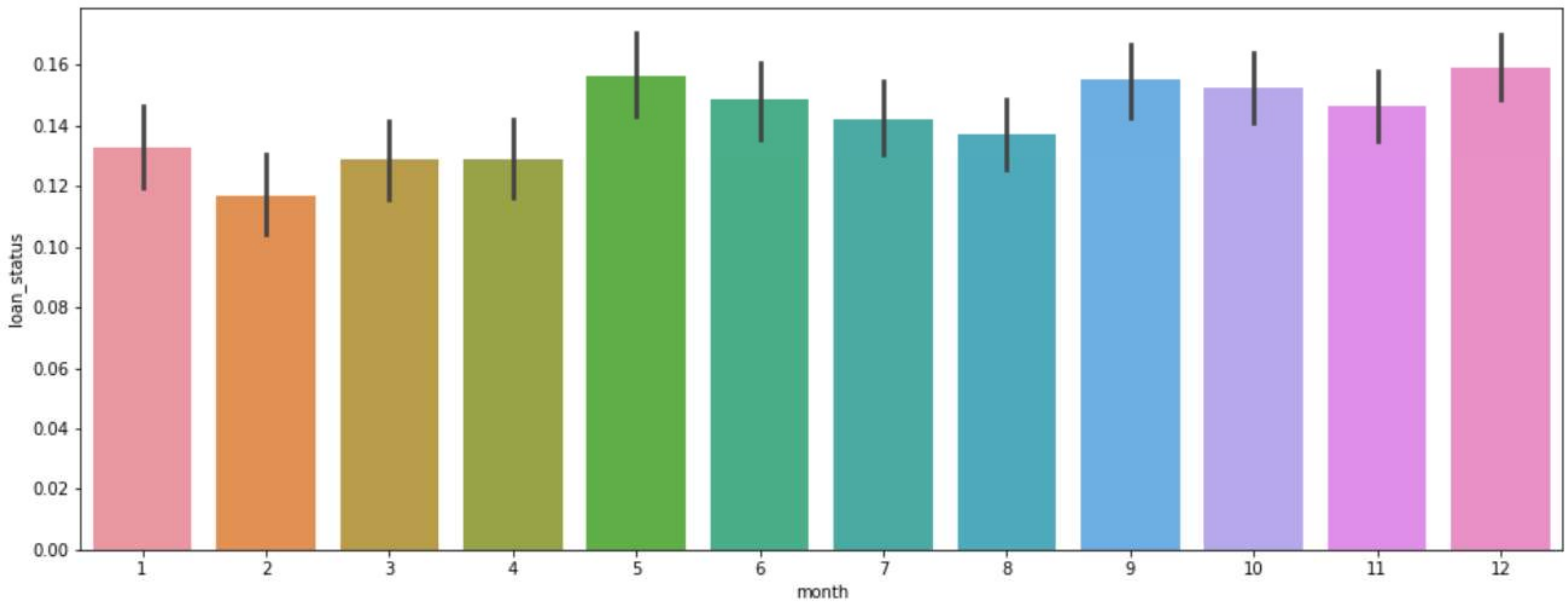
```
# lets compare the default rates across years
# the default rate had suddenly increased in 2011, inspite of reducing from 2008 till 2010
plot_cat('year')
```



Gramener Case Study

comparing default rates across months: not much variation across months

```python
# comparing default rates across months: not much variation across months
plt.figure(figsize=(16, 6))
plot_cat('month')
```

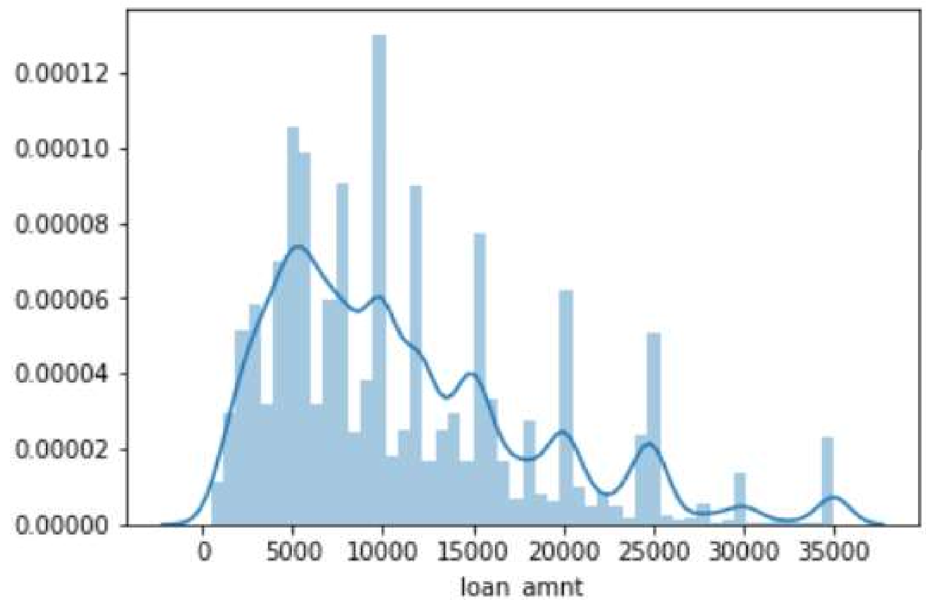


Gramener Case Study

loan amount: the median loan amount is around 10,000.

The easiest way to analyse how default rates vary across continous variables is to bin the variables into discrete categories. Let's bin the loan amount variable into small, medium, high, very high.

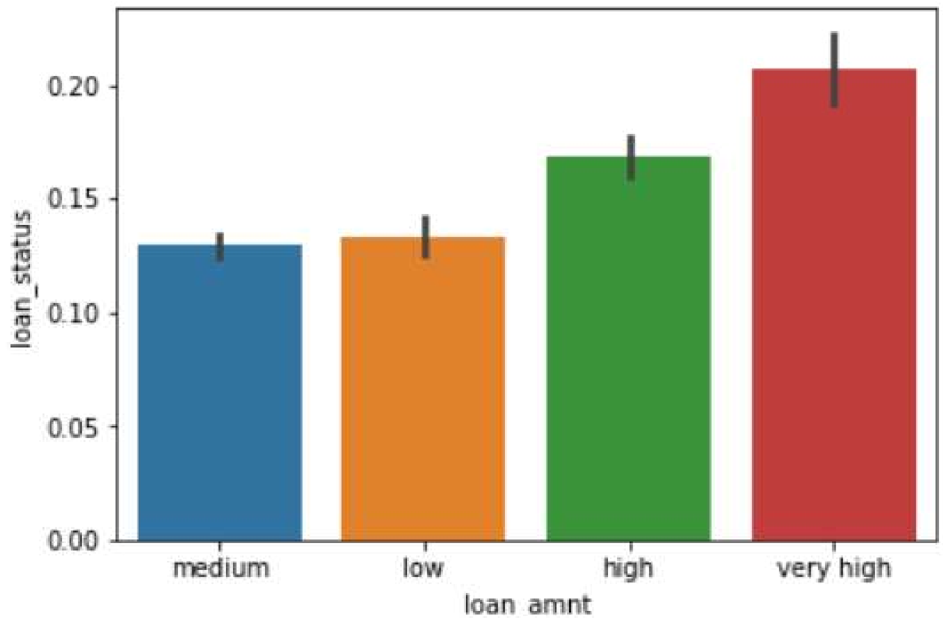
```
# loan amount: the median loan amount is around 10,000
sns.distplot(df['loan_amnt'])
plt.show()
```



Gramener Case Study

let's compare the default rates across loan amount type

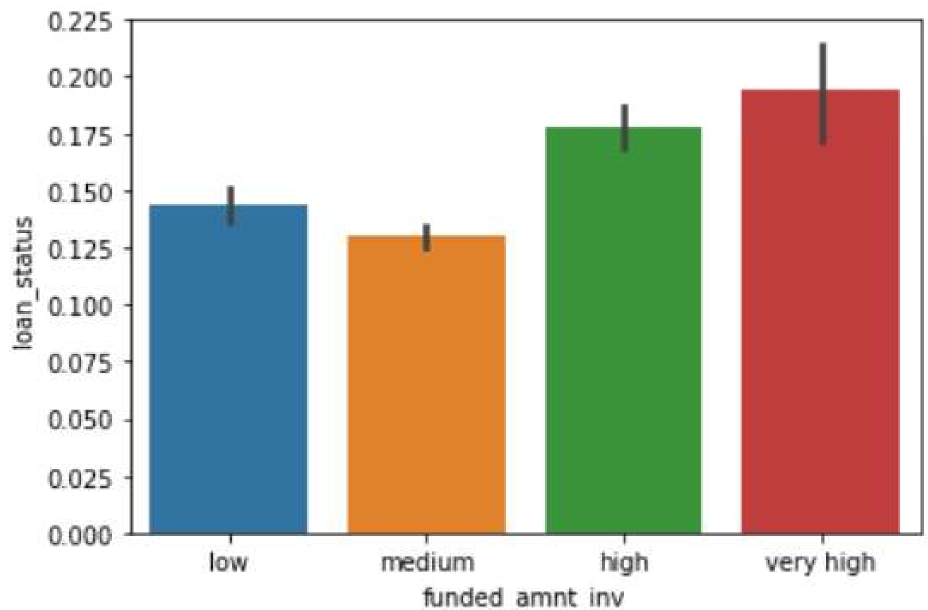
```
# let's compare the default rates across loan amount type
# higher the loan amount, higher the default rate
plot_cat('loan_amnt')
```



Gramener Case Study

Funded amount invested

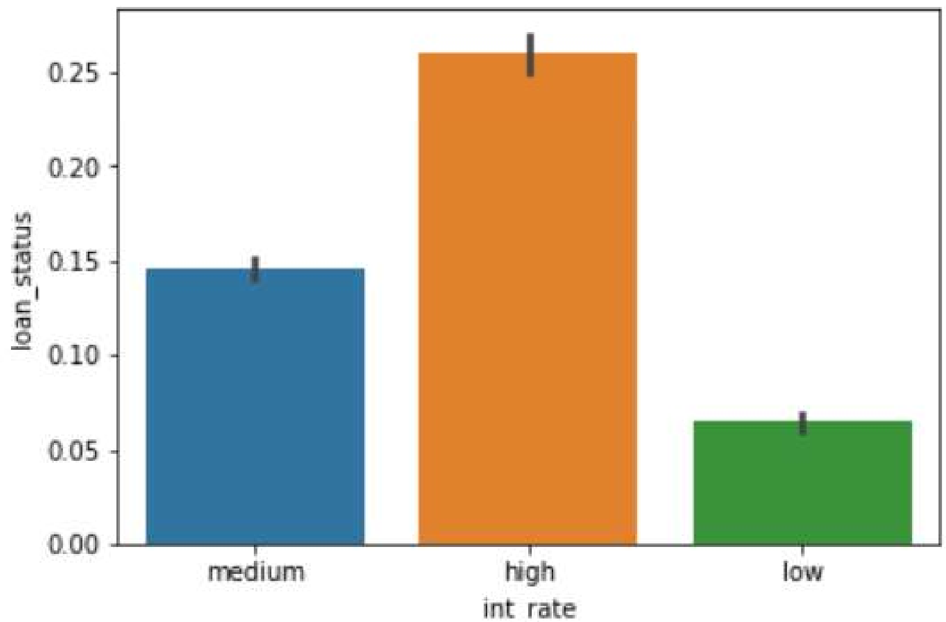
```
# funded amount invested
plot_cat('funded_amnt_inv')
```



Gramener Case Study

Comparing default rates across rates of interest

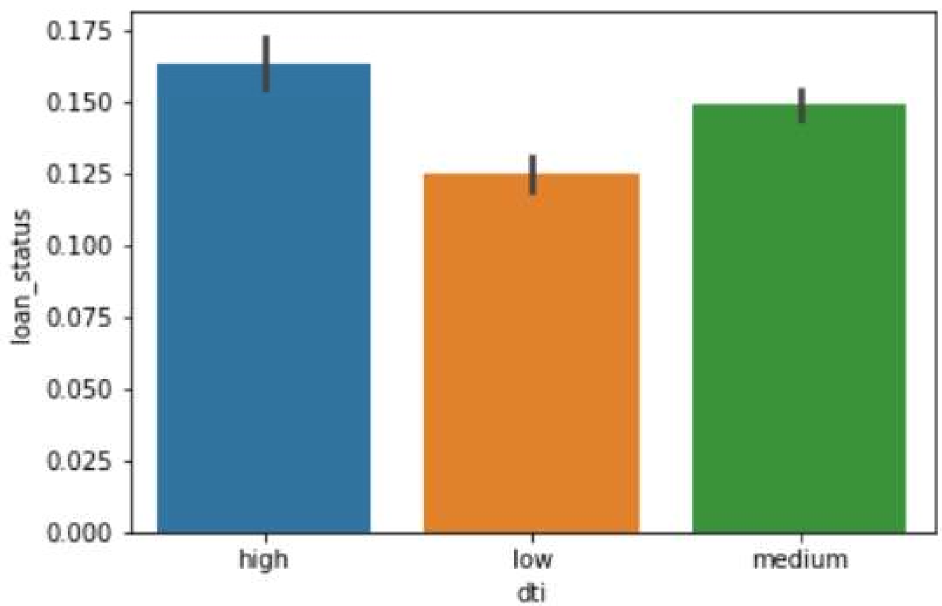
```
# comparing default rates across rates of interest
# high interest rates default more, as expected
plot_cat('int_rate')
```



Gramener Case Study

comparing default rates across debt to income ratio

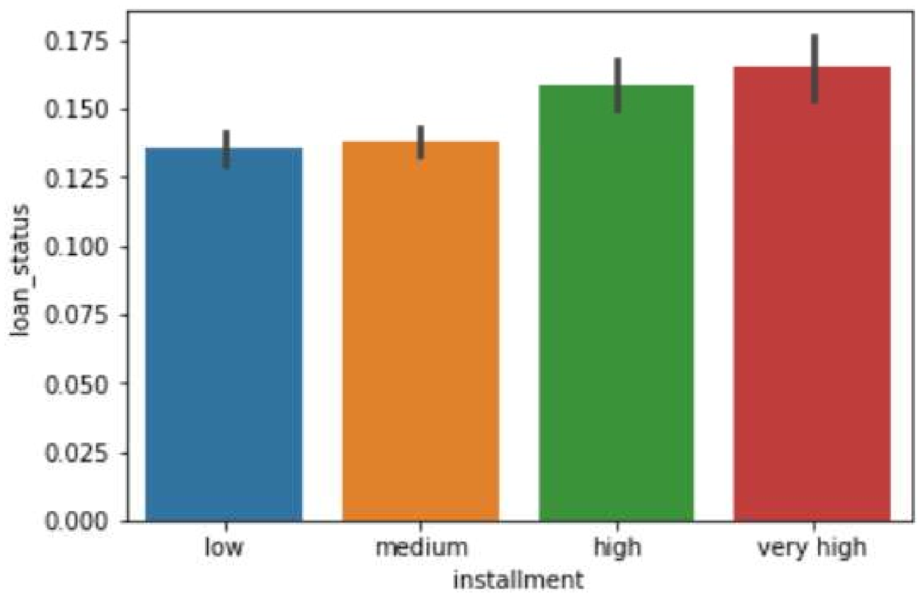
```
# comparing default rates across debt to income ratio
# high dti translates into higher default rates, as expected
plot_cat('dti')
```



Gramener Case Study

comparing default rates across installment

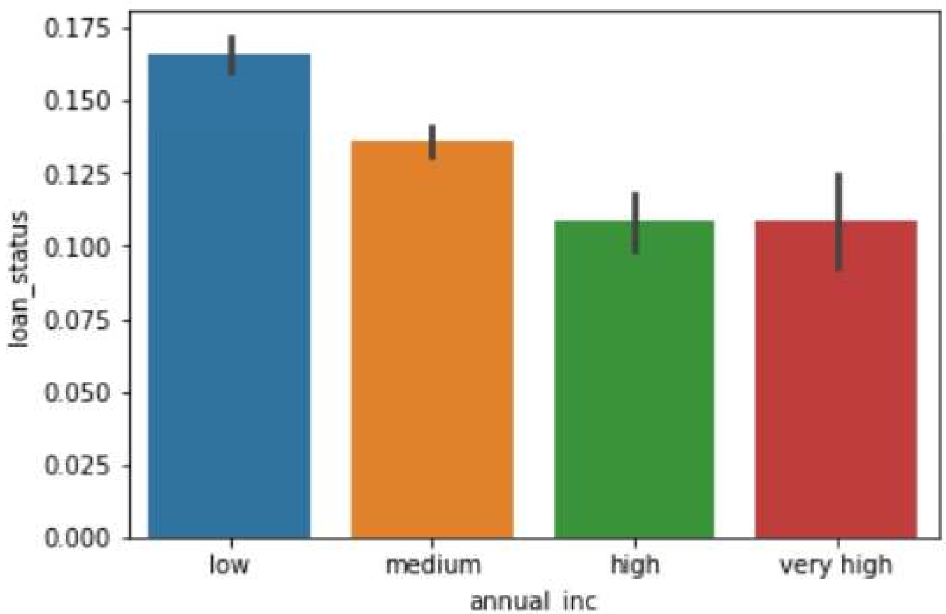
```
# comparing default rates across installment
# the higher the installment amount, the higher the default rate
plot_cat('installment')
```



Gramener Case Study

Annual income and default rate

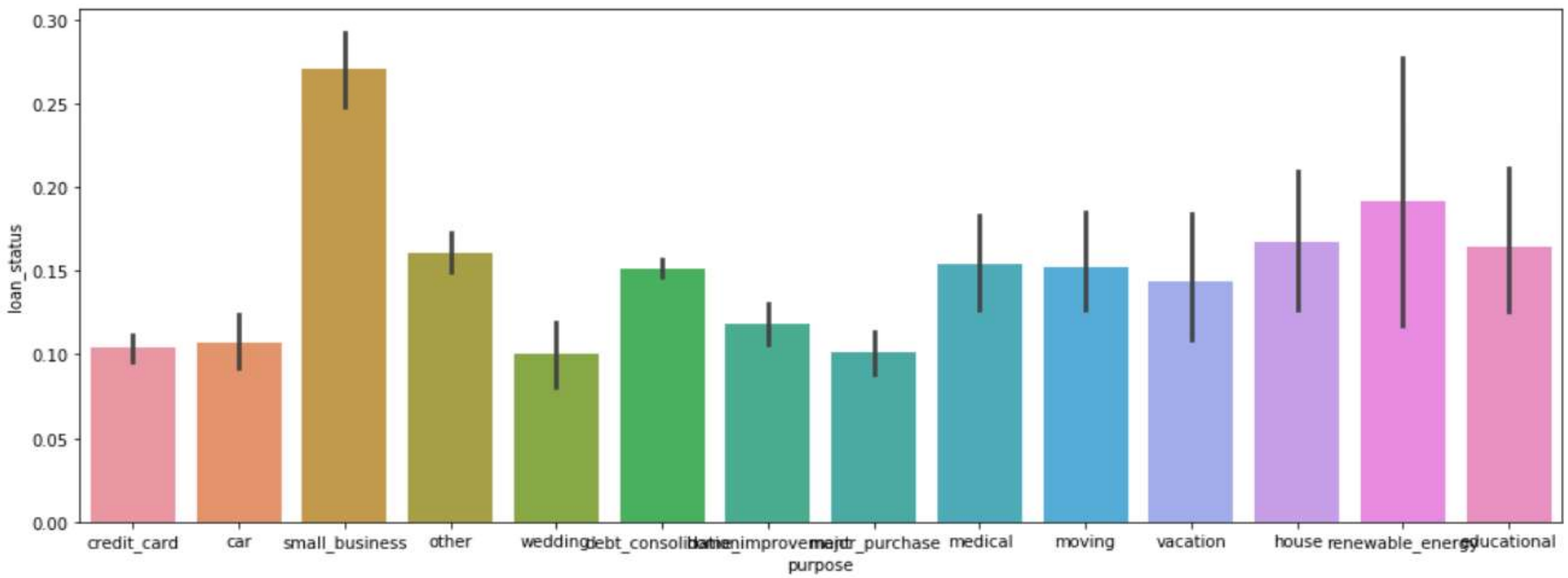
```
# annual income and default rate
# lower the annual income, higher the default rate
plot_cat('annual_inc')
```



Gramener Case Study

Small business loans defualt the most, then renewable energy and education

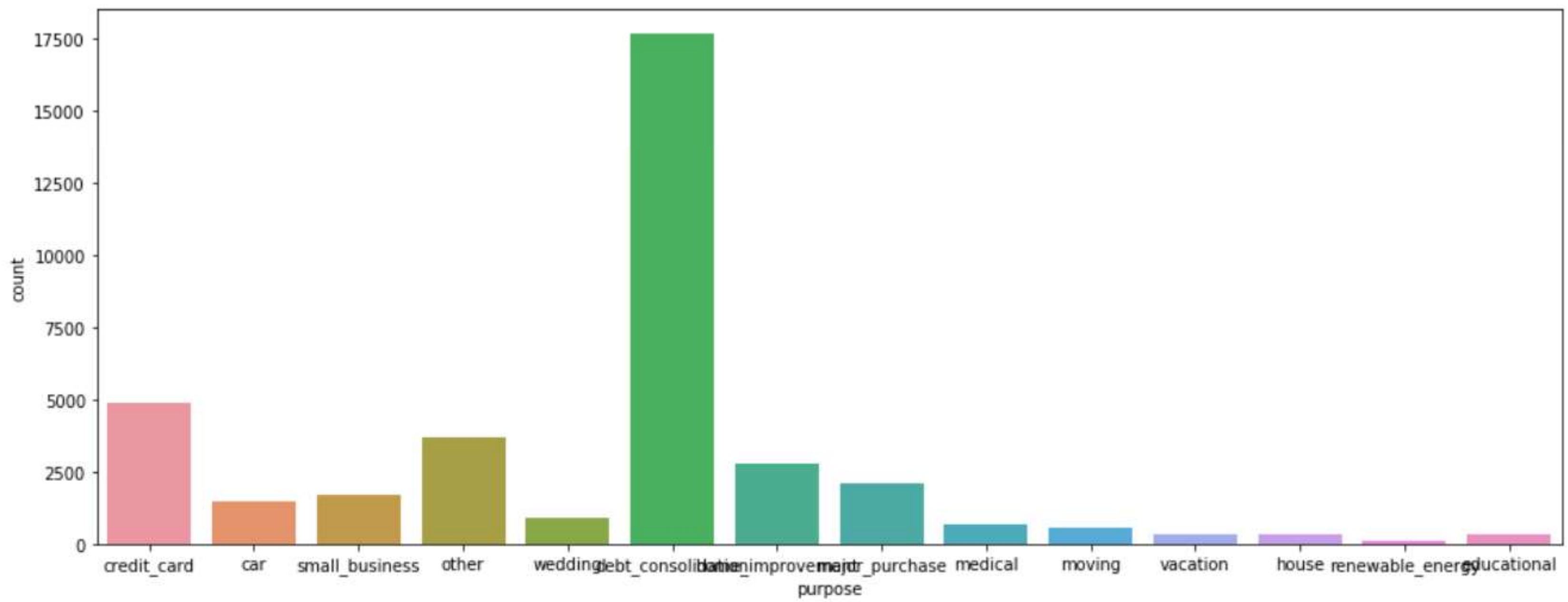
```
# purpose: small business loans defualt the most, then renewable energy and education
plt.figure(figsize=(16, 6))
plot_cat('purpose')
```



Gramener Case Study

Lets first look at the number of loans for each type (purpose) of the loan

```
# lets first look at the number of loans for each type (purpose) of the loan
# most loans are debt consolidation (to repay otehr debts), then credit card, major purchase etc.
plt.figure(figsize=(16, 6))
sns.countplot(x='purpose', data=df)
plt.show()
```



Gramener Case Study

After Analyzing

```
# filtering all the object type variables
df_categorical = df.loc[:, df.dtypes == object]
df_categorical['loan_status'] = df['loan_status']

# Now, for each variable, we can compute the incremental diff in default rates
print([i for i in df.columns])
```

```
['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'term', 'int_rate', 'installment', 'grade', 'sub_grade', 'em
p_title', 'emp_length', 'home_ownership', 'annual_inc', 'verification_status', 'issue_d', 'loan_status', 'pymnt_plan', 'purpos
e', 'dti', 'initial_list_status', 'collections_12_mths_ex_med', 'policy_code', 'acc_now_delinq', 'chargeoff_within_12_mths', 'd
elinq_amnt', 'pub_rec_bankruptcies', 'tax_liens', 'month', 'year']
```

```
/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
# storing the diff of default rates for each column in a dict
d = {key: diff_rate(key)[1]*100 for key in df_categorical.columns if key != 'loan_status'}
print(d)
```

```
{'loan_amnt': 7.000000000000009, 'funded_amnt_inv': 6.0, 'pymnt_plan': 0.0, 'verification_status': 4.0, 'emp_title': 100.0, 'd
ti': 5.0, 'home_ownership': 16.0, 'purpose': 5.0, 'sub_grade': 46.0, 'grade': 27.0, 'funded_amnt': 5.0, 'installment': 3.0, 'in
itial_list_status': 0.0, 'int_rate': 19.0, 'term': 15.0, 'annual_inc': 6.0, 'emp_length': 2.0}
```

THANK YOU

- GAURAV SINGH