None
HW3 Solutions

Note:

1. Only one possible right answer is shown. All possible right answers will be given full credit.
2. Only the final solution is shown, and the details of actual code is not shown.
3. The additional customizations used in the plots are for illustrations only. You are not required to do the additional customizations.
4. You may come to the office hours or the help sessions to discuss the HW solutions.
5. If you find any typos or issues, kindly contact your section instructor.

Table of Contents

Problem-A

```
--------------------------------------------------Problem #A--------------------------------------------------
A-1:
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke	comp
13	0	188.0	bmw	gas	std	four	sedan	rwd	front	101.2	...	164	mpfi	3.31	3.19	
116	0	161.0	peugot	diesel	turbo	four	sedan	rwd	front	107.9	...	152	idi	3.70	3.52	
20	0	81.0	chevrolet	gas	std	four	sedan	fwd	front	94.5	...	90	2bbl	3.03	3.11	
36	0	78.0	honda	gas	std	four	wagon	fwd	front	96.5	...	92	1bbl	2.92	3.41	
39	0	85.0	honda	gas	std	four	sedan	fwd	front	96.5	...	110	1bbl	3.15	3.58	
14	1	NaN	bmw	gas	std	four	sedan	rwd	front	103.5	...	164	mpfi	3.31	3.19	
172	2	134.0	toyota	gas	std	two	convertible	rwd	front	98.4	...	146	mpfi	3.62	3.50	
51	1	104.0	mazda	gas	std	two	hatchback	fwd	front	93.1	...	91	2bbl	3.03	3.15	
183	2	122.0	volkswagen	gas	std	two	sedan	fwd	front	97.3	...	109	mpfi	3.19	3.40	
146	0	89.0	subaru	gas	std	four	wagon	fwd	front	97.0	...	108	2bbl	3.62	2.64	
17	0	NaN	bmw	gas	std	four	sedan	rwd	front	110.0	...	209	mpfi	3.62	3.39	
118	1	119.0	plymouth	gas	std	two	hatchback	fwd	front	93.7	...	90	2bbl	2.97	3.23	
156	0	91.0	toyota	gas	std	four	sedan	fwd	front	95.7	...	98	2bbl	3.19	3.03	
28	-1	110.0	dodge	gas	std	four	wagon	fwd	front	103.3	...	122	2bbl	3.34	3.46	
133	2	104.0	saab	gas	std	four	sedan	fwd	front	99.1	...	121	mpfi	3.54	3.07	

15 rows × 26 columns

```
Random "n" rows form a dataframe can be displayed using df.sample(n) method.
```

```
----------------------------------------------------------------------------------------------------
A-2:

The column and row labels are available from df.columns and df.index methods.
The number of non null elements per column can be obtained from df.count method.
The column headers are: ['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration', 'num-of-doors',
'body-style', 'drive-wheels', 'engine-location', 'wheel-base', 'length', 'width', 'height', 'curb-weight', 'en
gine-type', 'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke', 'compression-ratio', 'horsepow
er', 'peak-rpm', 'city-mpg', 'highway-mpg', 'price']

The row labels are: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52
, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105,
106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127,
128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171,
172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193,
194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204]

The number of rows are: 205

The number of columns are: 26

The number of null rows for each column are:
symboling            205
normalized-losses    164
make                 205
fuel-type            205
aspiration           205
num-of-doors         203
body-style           205
drive-wheels         205
engine-location      205
wheel-base           205
length               205
width                205
height               205
curb-weight          205
engine-type          205
num-of-cylinders     205
engine-size          205
fuel-system          205
bore                 201
stroke               201
compression-ratio    205
horsepower           203
peak-rpm             203
city-mpg             205
highway-mpg          205
price                201
dtype: int64
```

A-3:

The numerical and non-numerical columns statistical summaries can be obtained from df.describe method.
Set the key-word argument (kwarg) " include='number' " for numeric columns and set the kwarg " include='object' " for non-numeric column.
The statistical summaries for numerical column are:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsep
count	205.000000	164.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	201.000000	201.000000	205.000000	203.0
mean	0.834146	122.000000	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	3.329751	3.255423	10.142537	104.2
std	1.245307	35.442168	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	0.273539	0.316717	3.972040	39.7
min	-2.000000	65.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.0
25%	0.000000	94.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	3.150000	3.110000	8.600000	70.0
50%	1.000000	115.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.0
75%	2.000000	150.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	3.590000	3.410000	9.400000	116.0
max	3.000000	256.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	288.0

The statistical summaries for non-numerical column are:

	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	engine-type	num-of-cylinders	fuel-system
count	205	205	205	203	205	205	205	205	205	205
unique	22	2	2	2	5	3	2	7	7	8
top	toyota	gas	std	four	sedan	fwd	front	ohc	four	mpfi
freq	32	185	168	114	96	120	202	148	159	94

A-4:

To select the rows from a dataframe based on some condition use mask!
The total number of rows corresponding to 'toyota' car will be: 32

Statistical summaries for the numerical columns:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepowe
count	32.000000	31.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.00000
mean	0.562500	110.290323	98.103125	171.934375	65.090625	53.721875	2441.093750	118.812500	3.280000	3.255000	10.340625	92.78125
std	1.216486	40.720342	3.349096	7.517318	1.276426	2.003019	354.510599	27.161925	0.186236	0.217582	3.978641	32.96697
min	-1.000000	65.000000	94.500000	158.700000	63.600000	52.000000	1985.000000	92.000000	3.050000	3.030000	8.700000	56.00000
25%	0.000000	84.000000	95.700000	166.300000	64.000000	52.600000	2161.750000	98.000000	3.190000	3.030000	9.000000	68.00000
50%	0.000000	91.000000	95.700000	169.700000	64.400000	53.000000	2313.000000	110.000000	3.270000	3.350000	9.000000	82.50000
75%	1.250000	134.000000	102.400000	176.200000	66.500000	54.500000	2583.000000	146.000000	3.310000	3.500000	9.300000	116.00000
max	3.000000	197.000000	104.500000	187.800000	67.700000	59.100000	3151.000000	171.000000	3.620000	3.540000	22.500000	161.00000

Statistical summaries for the non-numerical columns:

	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	engine-type	num-of-cylinders	fuel-system
count	32	32	32	32	32	32	32	32	32	32
unique	1	2	2	2	5	3	1	2	2	3
top	toyota	gas	std	four	hatchback	fwd	front	ohc	four	mpfi
freq	32	29	31	18	14	16	32	26	28	16

A-5:

To select the rows from a dataframe based on some condition use mask!
The range of price column for the records that have fuel-type as "gas" and horsepower between 100 and 130 is: 16876.0

A-6:

To select the rows from a dataframe based on some condition use mask!
The proportion of cars having "two" doors and length greater than or equal to 170 is: 0.1902439024390244

A-7:

To select the rows in ascending or descending order use df.sort_values method.
The 15 cars in the data that has the highest price:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke	compr
74	1	NaN	mercedes-benz	gas	std	two	hardtop	rwd	front	112.0	...	304	mpfi	3.80	3.35	
16	0	NaN	bmw	gas	std	two	sedan	rwd	front	103.5	...	209	mpfi	3.62	3.39	
73	0	NaN	mercedes-benz	gas	std	four	sedan	rwd	front	120.9	...	308	mpfi	3.80	3.35	
128	3	NaN	porsche	gas	std	two	convertible	rwd	rear	89.5	...	194	mpfi	3.74	2.90	
17	0	NaN	bmw	gas	std	four	sedan	rwd	front	110.0	...	209	mpfi	3.62	3.39	
49	0	NaN	jaguar	gas	std	two	sedan	rwd	front	102.0	...	326	mpfi	3.54	2.76	
48	0	NaN	jaguar	gas	std	four	sedan	rwd	front	113.0	...	258	mpfi	3.63	4.17	
72	3	142.0	mercedes-benz	gas	std	two	convertible	rwd	front	96.6	...	234	mpfi	3.46	3.10	
71	-1	NaN	mercedes-benz	gas	std	four	sedan	rwd	front	115.6	...	234	mpfi	3.46	3.10	
127	3	NaN	porsche	gas	std	two	hardtop	rwd	rear	89.5	...	194	mpfi	3.74	2.90	
126	3	NaN	porsche	gas	std	two	hardtop	rwd	rear	89.5	...	194	mpfi	3.74	2.90	
47	0	145.0	jaguar	gas	std	four	sedan	rwd	front	113.0	...	258	mpfi	3.63	4.17	
70	-1	93.0	mercedes-benz	diesel	turbo	four	sedan	rwd	front	115.6	...	183	idi	3.58	3.64	
15	0	NaN	bmw	gas	std	four	sedan	rwd	front	103.5	...	209	mpfi	3.62	3.39	
68	-1	93.0	mercedes-benz	diesel	turbo	four	wagon	rwd	front	110.0	...	183	idi	3.58	3.64	

15 rows × 26 columns

Statistical summaries for the numerical columns:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower
count	15.000000	4.000000	15.000000	15.00000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000
mean	0.666667	118.250000	105.613333	190.74000	69.226667	53.886667	3537.933333	233.133333	3.637333	3.343333	10.386667	180.333333
std	1.543033	29.181901	10.363733	13.10326	2.598754	2.925227	458.216647	47.702750	0.108395	0.430758	4.615636	34.572216
min	-1.000000	93.000000	89.500000	168.90000	65.000000	47.800000	2756.000000	183.000000	3.460000	2.760000	8.000000	123.000000
25%	0.000000	93.000000	99.300000	184.65000	67.400000	51.600000	3305.000000	194.000000	3.580000	3.000000	8.000000	165.500000
50%	0.000000	117.500000	110.000000	193.80000	70.300000	53.700000	3715.000000	209.000000	3.620000	3.350000	8.300000	182.000000
75%	2.000000	142.750000	113.000000	199.60000	71.300000	56.300000	3835.000000	258.000000	3.740000	3.515000	9.500000	195.500000
max	3.000000	145.000000	120.900000	208.10000	72.000000	58.700000	4066.000000	326.000000	3.800000	4.170000	21.500000	262.000000

A-8:

To apply custom (or lambda) functions on every element of a column (or more than one column), use df.apply or df.applymap methods.

The dataframe's columns after modifications will be:

	aspiration	length	width	height
0	standard	169	64	49
1	standard	169	64	49
2	standard	171	66	52
3	standard	177	66	54
4	standard	177	66	54

A-9:

To apply custom (or lambda) functions on every element of a column (or more than one column), use df.apply or df.applymap methods.

A new column in a dataframe can be created by assigning df["new column name"]= the new column values.

The dataframe's columns after modifications will be:

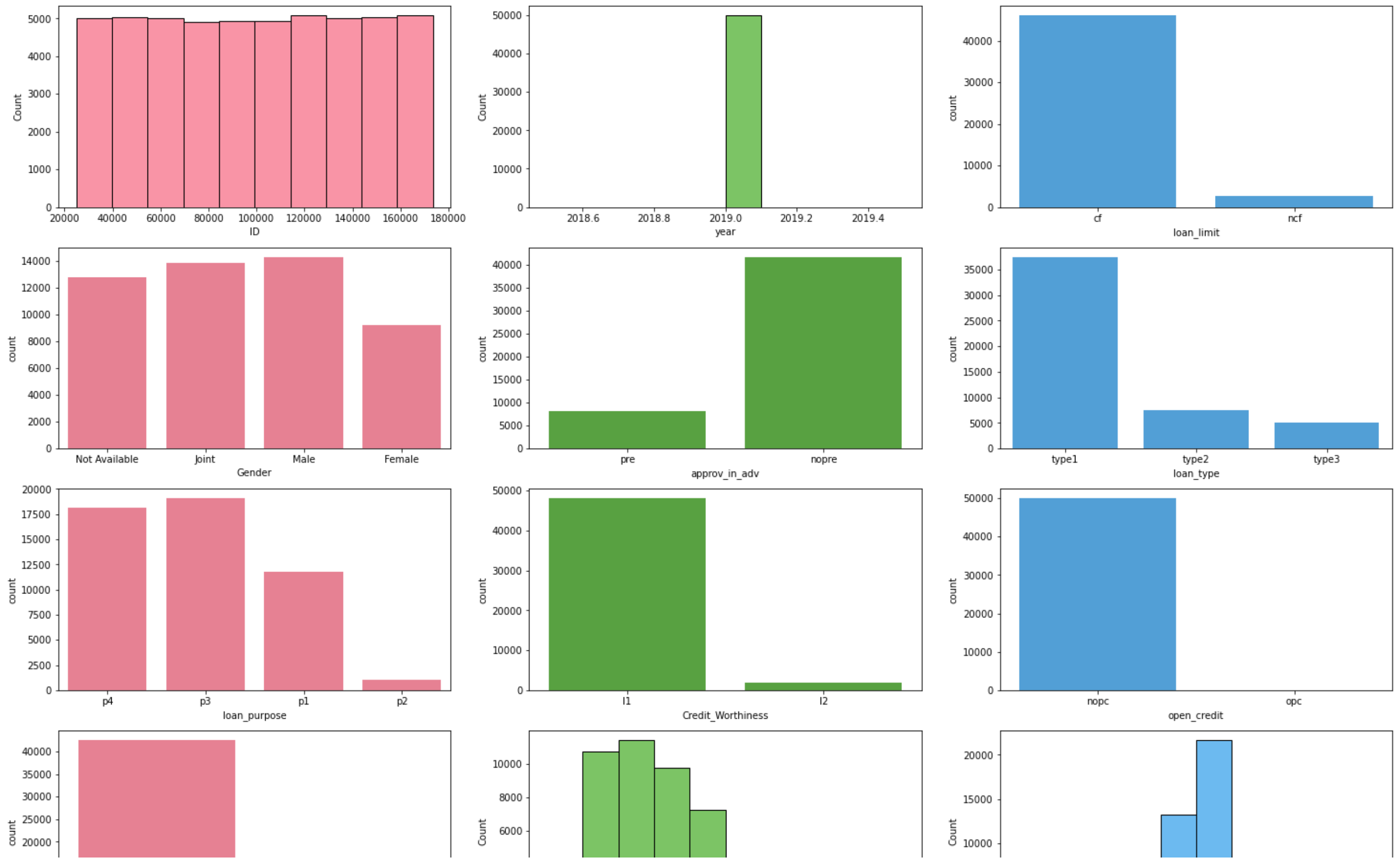
	city-mpg	city-kpl	highway-mpg	highway-kpl
0	21	8.925	27	11.475
1	21	8.925	27	11.475
2	19	8.075	26	11.050
3	24	10.200	30	12.750
4	18	7.650	22	9.350

~-~

Problem-B

--Problem #B--

B-1: The histograms for all numeric and nonnumeric columns are as follows (for numeric columns 10 bins are taken).

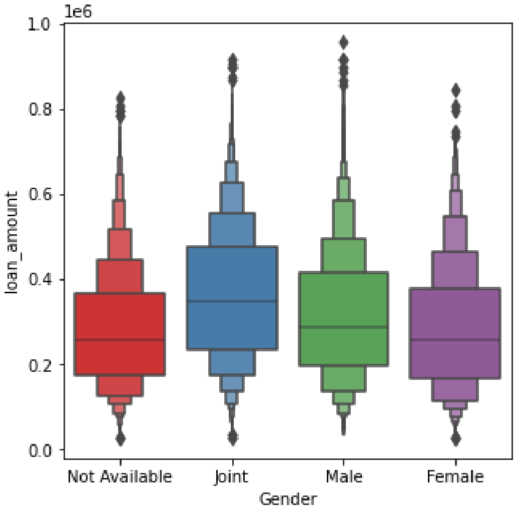



nob/c
b/c
business_or_commercial
loan_amount
1e6
rate_of_interest
Count
Interest_rate_spread
Upfront_charges
term
count
not_neg
neg_amm
Neg_ammortization
not_int
int_only
interest_only
not_lpsm
lpsm
lump_sum_payment
property_value
sb
mh
construction_type
pr
ir
sr
occupancy_type
home
land
Secured_by
1U
2U
4U
3U
total_units
income
EXP
CRIF
CIB
EQUI
credit_type
Credit_Score
co-applicant_credit_type
25-34
35-44
55-64
65-74
45-54
>74
<25
age
to_inst
not_inst
submission_of_application
LTV
south
North
central
North-East
Region
direct
Indriect
Security_Type
Status
dtir1

B-2:
Loan amount is right skewed. Majority of loans are less than 500,000 (0.5*1e6)
Rate of interest appears normally distributed, and most loans has interest rate between 3.5% to 4.5%
There are 3 loan types (type1, type2 and type3). Type1 is the most common type, followed by type2, and then followed by type3.
The data is slightly right skewed. Most properties have values between 200,000 and 400,000. The frequency starts decreasing above 400,000 and only few properties have a value greater than 800,000. The maximum value in this dataset is 1,000,000

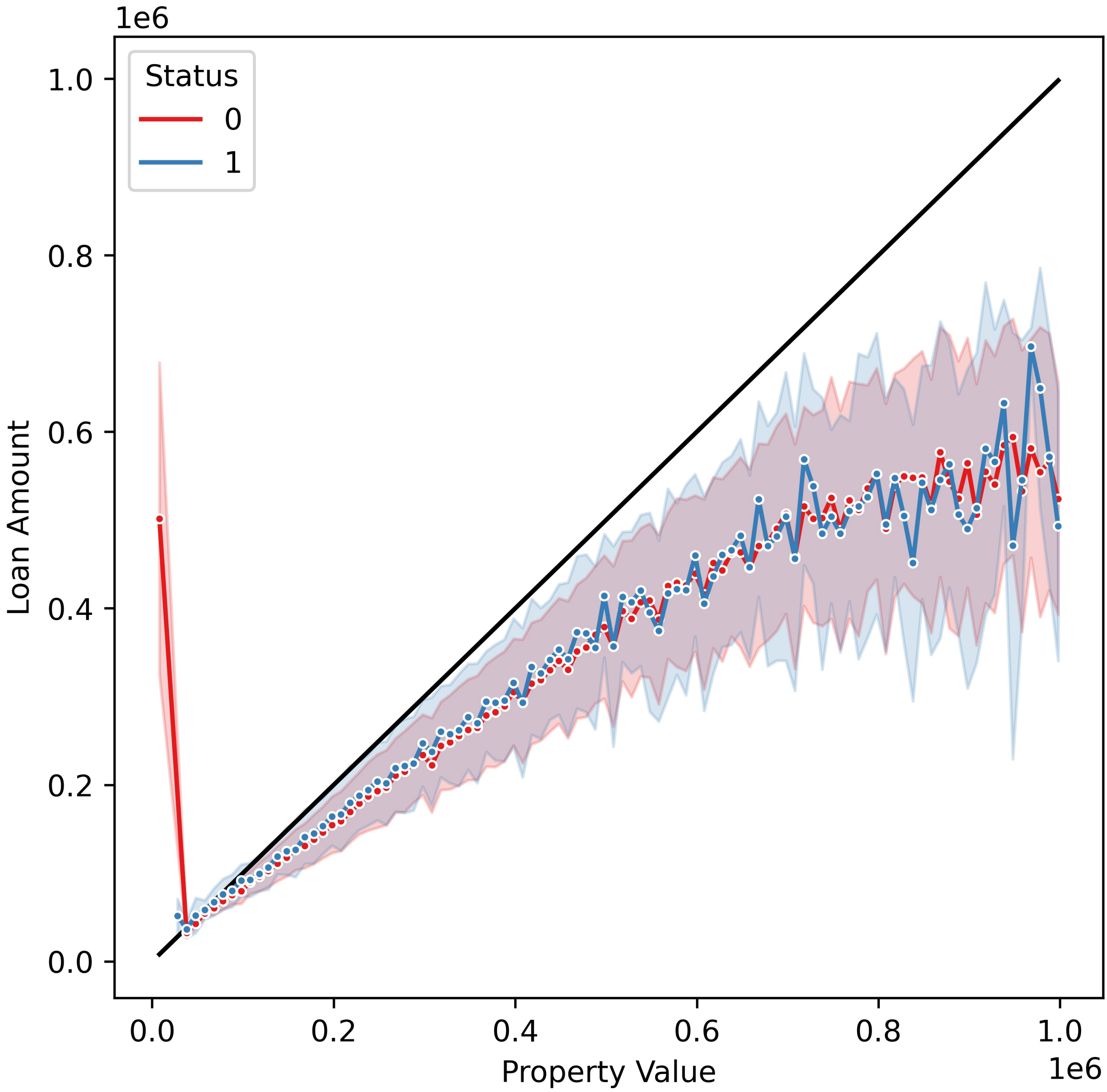
B-3:

To draw a plot between numeric and categorical column, use box-plot, violin-plot, boxen-plot or swarm-plot plots.
The boxen plot between loan amount and Gender is as follows:



The "Joint" gender category seems to have higher median loan amount.
The 'Female' and 'Not available' gender category seems to have lower median for loan amount.
Across all categories, there are outliers

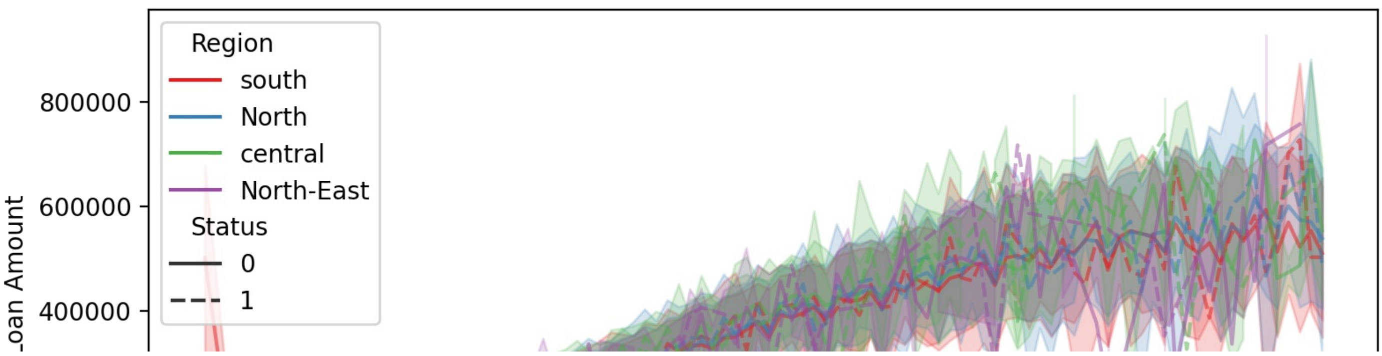
B-4:
The required plot between property value and loan amount, differentiated by Status is:

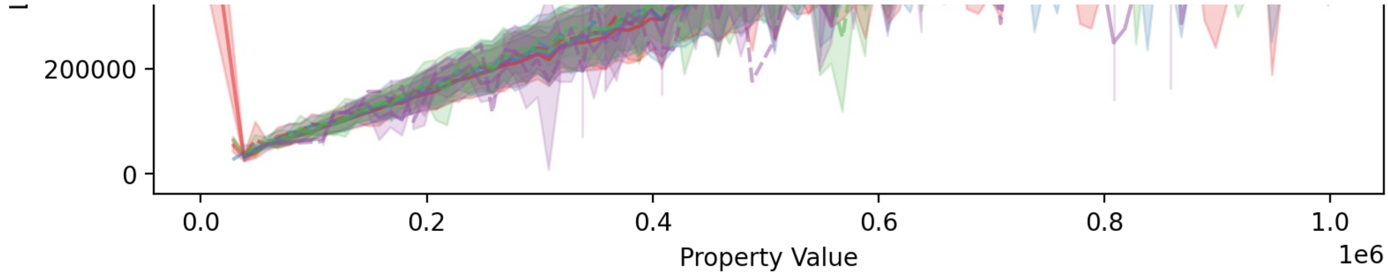


The black line in the plot depicts property_value=loan_amount.
If the loan amount was very close to (or exceed) the property value, then it is defaulted most of the time.
For loan amoutns more than 400,000, the defaulted status and have high variance then non-default staus (blue area is typically more than the red area).

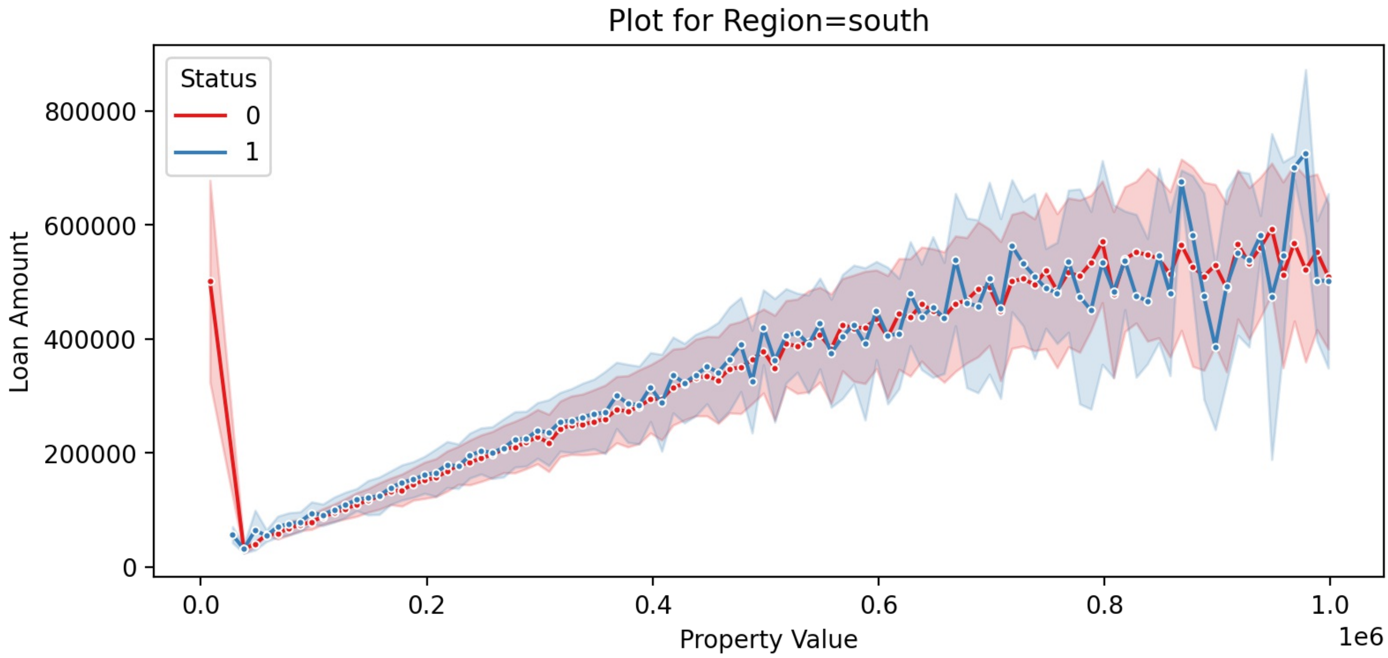
B-5:

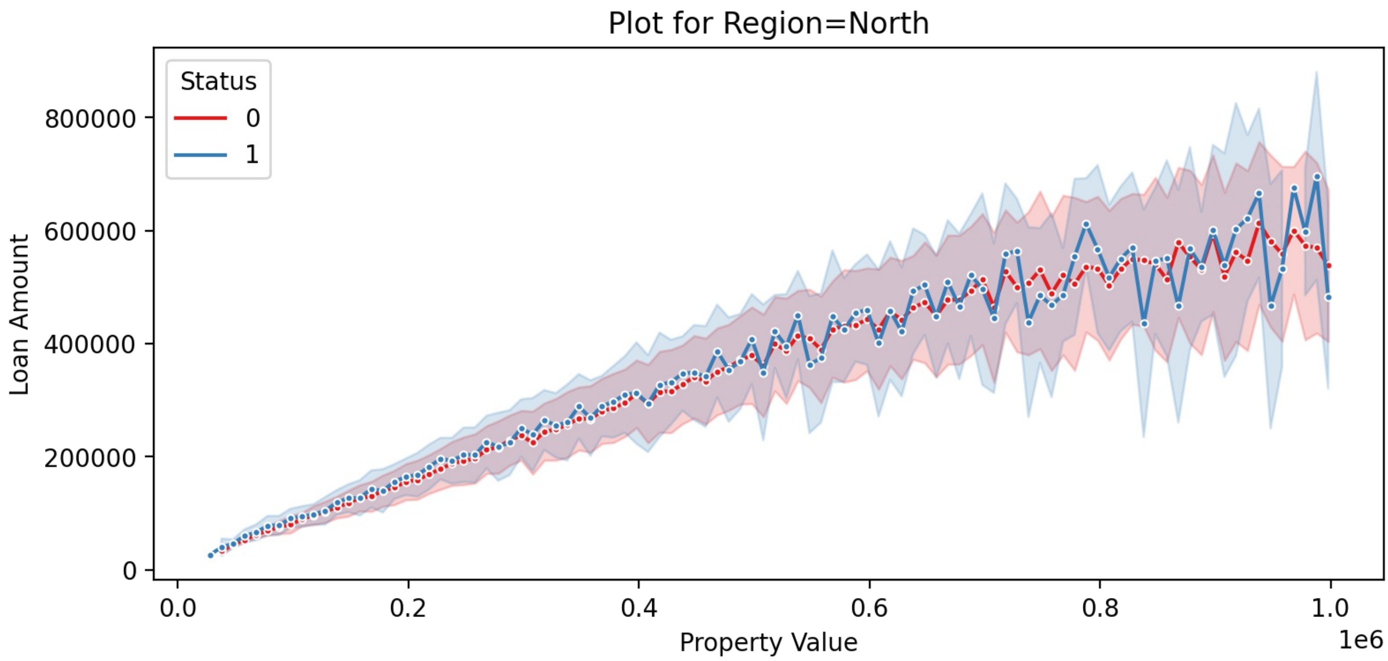
To diferentiate line or scatter plot twice, we can use hue and style kwargs.
The required plot between property value and loan amount, differentiated by Status and Region:

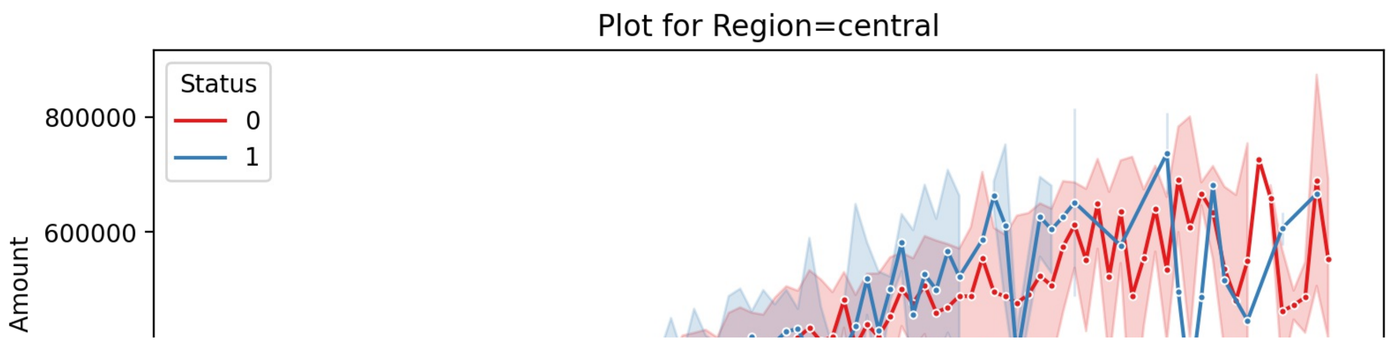


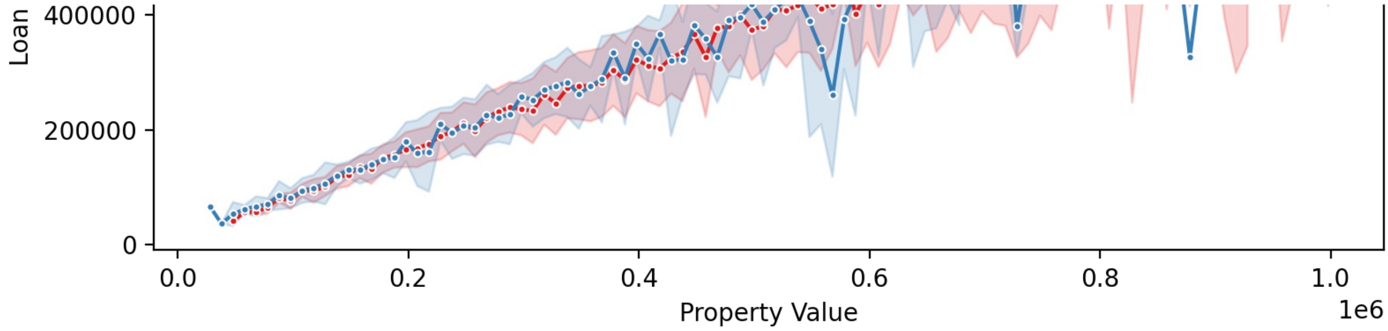


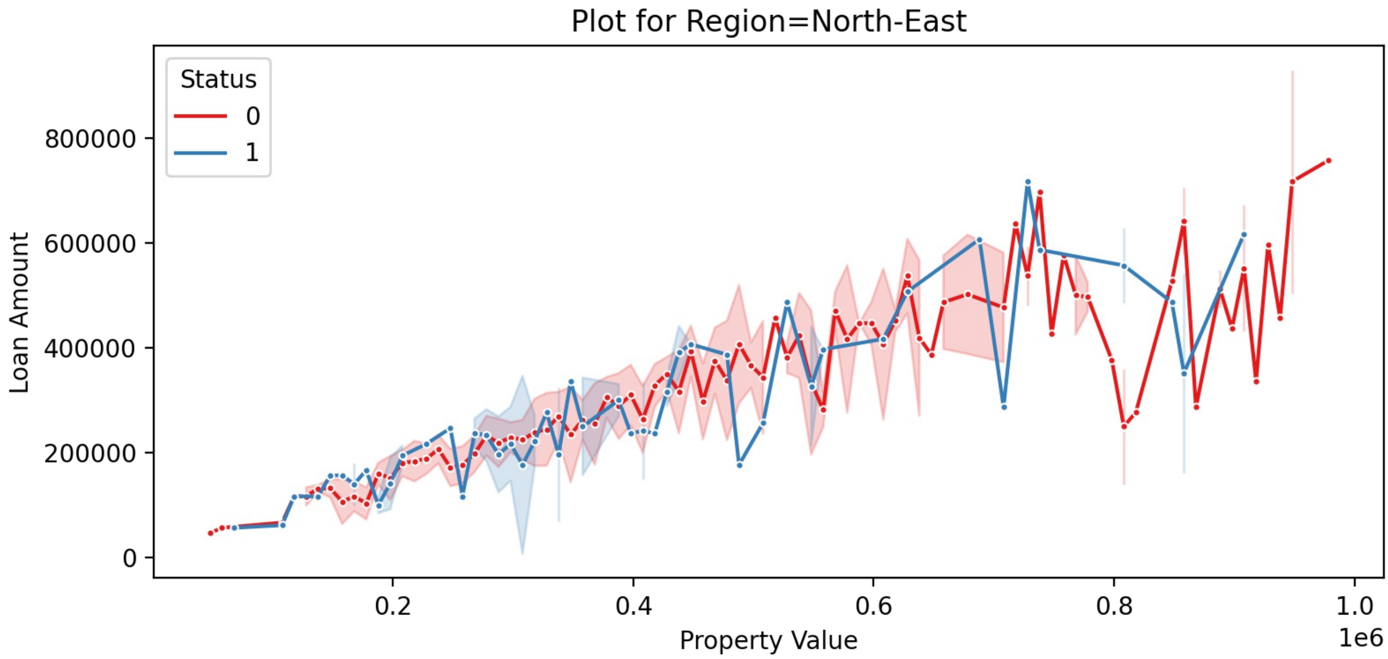

```
Usually, the above plot may be cluttered, and hard to interpret.
Thus, we can draw multiple plots, where each plot is for one Region.
For example:
```





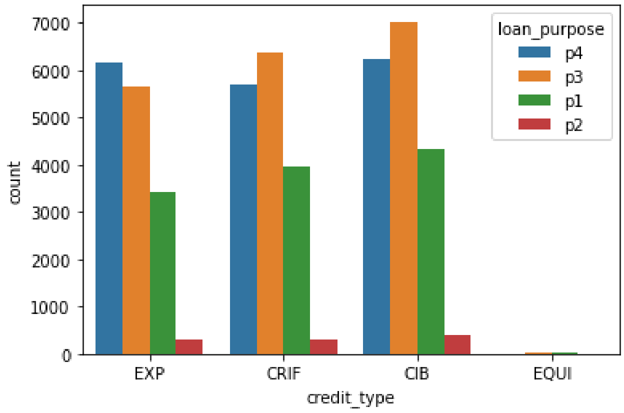






B-6:

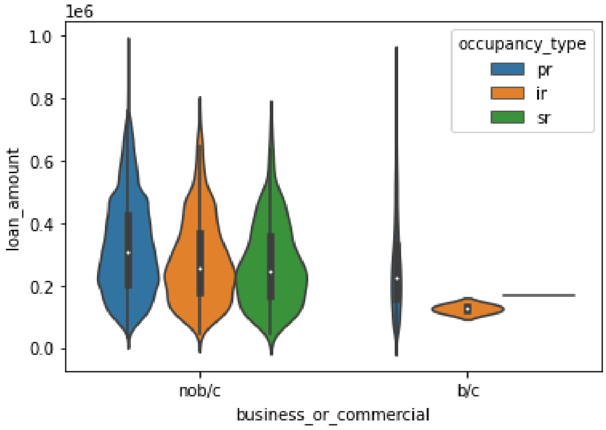
A single categorical columns can be depicted by histogram (histplot or countplot).
We use hue to differentiate by categorical column.
The count plot of credit type, differentiated by loan purpose is:



B-7:

To draw a plot between numeric and categorical column, use box-plot, violin-plot, boxen-plot or swarm-plot plots. We use hue to differentiate by categorical column.

The plot on the loan amount differentiated by business or commercial and occupancy type is:



B-8:

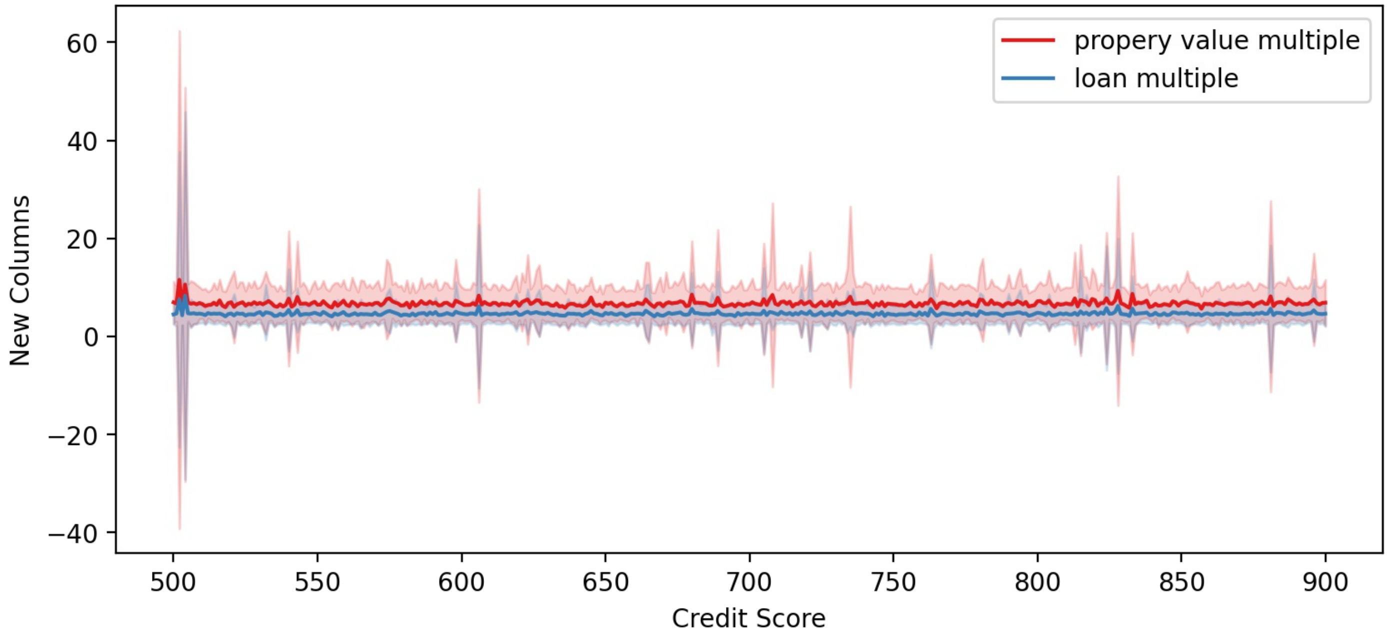
A new column in a dataframe can be created by assigning df["new column name"]= the new column values.

The new columns in the dataframe are:

	propery value multiple	loan multiple
0	7.079946	4.852642
1	10.472973	6.878754
2	6.001048	3.229822
3	9.863946	3.018707
4	4.688419	3.428131
5	5.150463	4.528356
6	9.976105	6.668160
7	2.343750	2.094184

B-9:

The required plot is:



~-