BitTiger DS501 Week 1 HW

Meina Wang

Question 1
You're about to get on a plane to Seattle. You want to know if you should bring an umbrella. You call 3 random friends of yours who live there and ask each independently if it's raining. Each of your friends has a 2/3 chance of telling you the truth and a 1/3 chance of messing with you by lying. All 3 friends tell you that "Yes" it is raining. What is the probability that it's actually raining in Seattle.

Answer 1:

Base on the Bayes' theorem.

$$\boxed{\mathrm{P(raining|all\ say\ raining)} = \frac{P(all\ say\ raining|raining)P(raining)}{P(all\ say\ raining)}}$$

$\boxed{\mathrm{P}(all\ say\ raining|\mathrm{raining})}$ is the probability that all 3 friends are telling the truth. Therefore,
$\boxed{\mathrm{P(all\ say\ raining|raining)} = (\frac{2}{3})^3}$.
$\boxed{\mathrm{P(all\ say\ raining)} = \mathrm{P(all\ say\ raining|raining)P(raining)} + P(all\ say\ raining|not\ raining)P(not\ raining)}$.
We assume the $\boxed{\mathrm{P(raining)} = p}$, which we can get calculate from past observation or experience. Then
$\boxed{\mathrm{P(all\ say\ raining)} = (\frac{2}{3})^3 p + \left(\frac{1}{3}\right)^3 (1-\mathrm{p})}$.

Substitute all the above values into the Bayes' theorem, we get

$$\boxed{\mathrm{P(raining|all\ say\ raining)} = \frac{(\frac{2}{3})^3\mathrm{p}}{(\frac{2}{3})^3 p + \left(\frac{1}{3}\right)^3 (1-\mathrm{p})} = \frac{8p}{7p+1}}.$$

Once we can get the $\boxed{\mathrm{P(raining)}}$, we can plug in the above equation and calculate the probability it's actually raining given all friends saying it's raining.

Question 2
You have two coins. One of is fair and the other is biased and comes up heads with probability 3/4. You randomly pick coin and flip it twice¨ and get heads both times. What is the probability that you picked the fair coin?

Answer 2:

Base on the Bayes' theorem.

$$\boxed{\mathrm{P(fair\ coin|two\ heads)} = \frac{P(two\ heads|fair\ coin)P(fair\ coin)}{P(two\ heads)}}$$

Given the description, $\boxed{\mathrm{P(two\ heads|fair\ coin)} = (\frac{1}{2})^2}$, and $\boxed{\mathrm{P(two\ heads|biased\ coin)} = (\frac{3}{4})^2}$. Assume that the chance of picking up either fair coin or biased coin is the same. So $\boxed{\mathrm{P(fair\ coin)} = \frac{1}{2}}$.

$$\boxed{\mathrm{P(two\ heads)} = \mathrm{P(two\ heads|faircoin)P(fair\ coin)} + P(two\ heads|biased\ coin)P(biased\ coin) = (\frac{1}{2})^2\frac{1}{2} + (\frac{3}{4})^2\frac{1}{2} = \frac{13}{32}}.$$

With Bayes' theorem, substitute all the above values, $\boxed{P(\text{fair coin}|\text{two head}) = \frac{\left(\frac{1}{2}\right)^2 \frac{1}{2}}{\frac{13}{32}} = \frac{4}{13}}$.

Therefore, the probability that one picked the fair coin is $\boxed{\frac{4}{13}}$.

Question 3
Provide a simple example of how an experimental design can help answer a question about behavior. How does experimental data contrast with observational data?

Answer 3:

Taking the example of the relationship between chocolate consumption with the number of Nobel prize winner. Observational data might suggest a country with higher chocolate consumption has more Nobel prize winners. We can design an experiment to test this statement. First, we would randomly split counties into two groups, and only provide one group with chocolate. Then we can use hypothesis testing and come to the conclusion that the number of Nobel prize winners do not show significant difference between the two groups, which proves the previous statement wrong.

The experimental data applies a treatment to a group, and it attempts to isolate the effects of the treatment. Therefore, we can have conclusion of causation from experimental data. Whereas the observational data does not attempt to influence the variable of interest, so the results can only be associations, not causation.

Question 4
In a study of emergency room waiting times, investigators consider a new and the standard triage systems. To test the systems, administrators selected 20 nights and randomly assigned the new triage system to be used on 10 nights and the standard system on the remaining 10 nights. They calculated the nightly median waiting time (MWT) to see a physician. The average MWT for the new system was 3 hours with a variance of 0.60 while the average MWT for the old system was 5 hours with a variance of 0.68. Consider the 95% confidence interval estimate for the differences of the mean MWT associated with the new system. Assume a constant variance. What is the interval? Subtract in this order (New System - Old System).

Answer 4:

The emergency room waiting time should follow a normal distribution.
The Z value for 95% CI is 1.960,

Using the two sample t-test, the new system X1 follows the distribution of N(3, 0.60), and the old system X2 follows the distribution of N(5, 0.68). The pooled sample standard deviation is

$$\boxed{s_p = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}} = \sqrt{\frac{9*0.60 + 9*0.68}{10+10-2}} = 0.8}$$

The standard error is

$$\boxed{SE = s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}} = 0.8*\sqrt{\frac{1}{10}+\frac{1}{10}} \approx 0.3578}$$

From looking up (here: https://www.medcalc.org/manual/t-distribution.php) the Z-value associated with 95% confidence level with 2.5% on each tail, and 18 degree of freedom, Z = 2.101

Using the Z-value, the confidence interval is

$$\boxed{\mathrm{CI} = (\mathrm{X1} - \mathrm{X2}) \pm \mathrm{Z} * \mathrm{SE} = (3 - 5) \pm 2.101 * 0.3578 = -2 \pm 0.7516 \approx -2.75, -1.25}$$

Therefore, the CI is [-2.75, -1.25]

Question 5

Using lending club dataset examine relationship between each feature and response (interest rate). Pick 5 categorical and 5 numeric features which you think are the most predictive with reasoning.

Answer 5 (font color in blue is R code):

Numerical features:

I first selected column with only numerical values into another df called loan_nums.

```
nums <- sapply(loan, is.numeric)
loan_nums <- loan[, nums]
```

Similarly, columns with only characters were selected into another df called loan_categories

```
categories <- sapply(loan, is.character)
loan_categories <- loan[, categories]
```

First, check the numeric features and see if any of the features have constant values.

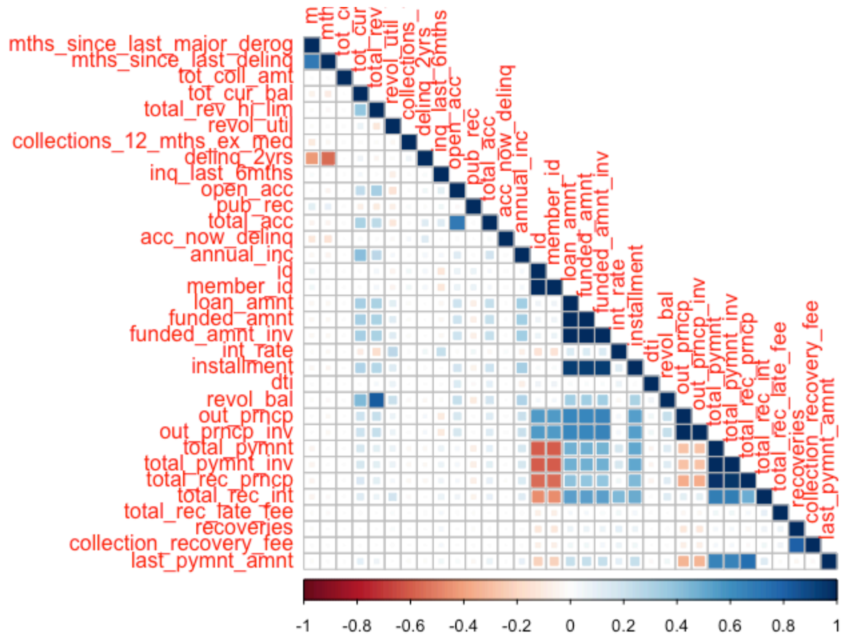
```
sapply(loan_nums, function(col) length(unique(col)))
```

From there it showed the variable 'policy_code' has only 1 unique level. So, delete this variable since it does not provide any predictive power.

```
loan_nums_no_constant <- loan_nums[, sapply(loan_nums, function(x) {length(unique(x)) > 1})]
```

Then compute the Pearson correlation between each variable and 'int_rate', and plot the values.

```
nums_corr <- cor(loan_nums_no_constant,use = 'pairwise.complete.obs')
nums_corr
corrplot(nums_corr, method = 'square', tl.cex = 1, type = 'lower')
```



From this plot we can see the correlations associated with 'int_rate'. However, it is difficult to tell the exact value from this plot.

Next, build a simple linear model with all the features and predict the 'int_rate'.

```
ols = lm(int_rate~.,loan_nums_no_constant)
summary(ols)
```

The summary of the model is:

```
Call:
lm(formula = int_rate ~ ., data = loan_nums_no_constant)

Residuals:
    Min      1Q  Median      3Q     Max
-59.483  -2.203  -0.329   1.802  54.873

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.309e+01  4.356e-02 300.426  < 2e-16 ***
mths_since_last_major_derog 3.932e-03  4.501e-04   8.734  < 2e-16 ***
mths_since_last_delinq     -5.487e-03  4.969e-04 -11.043  < 2e-16 ***
tot_coll_amt                4.423e-07  3.500e-07   1.264  0.20626
tot_cur_bal                -1.438e-06  5.347e-08 -26.899  < 2e-16 ***
total_rev_hi_lim           -4.696e-05  6.842e-07 -68.630  < 2e-16 ***
revol_util                  7.702e-03  3.888e-04  19.809  < 2e-16 ***
collections_12_mths_ex_med  2.376e-01  3.600e-02   6.599 4.15e-11 ***
delinq_2yrs                 4.766e-02  6.462e-03   7.375 1.65e-13 ***
inq_last_6mths              8.549e-01  7.210e-03 118.570  < 2e-16 ***
open_acc                    6.111e-02  1.899e-03  32.178  < 2e-16 ***
pub_rec                     1.833e-01  1.069e-02  17.146  < 2e-16 ***
total_acc                  -2.517e-02  8.335e-04 -30.202  < 2e-16 ***
acc_now_delinq              1.061e+00  6.409e-02  16.558  < 2e-16 ***
annual_inc                 -5.029e-06  1.267e-07 -39.682  < 2e-16 ***
```

```
id                      2.520e-08  9.625e-09   2.618  0.00884 **
member_id              -4.308e-08  9.146e-09  -4.710 2.48e-06 ***
loan_amnt              -1.641e-04  3.380e-04  -0.486  0.62724
funded_amnt             1.262e-02  1.966e-03   6.418 1.38e-10 ***
funded_amnt_inv        -1.247e-02  1.927e-03  -6.472 9.72e-11 ***
installment             4.023e-03  1.049e-04  38.370  < 2e-16 ***
dti                     5.919e-03  3.069e-04  19.287  < 2e-16 ***
revol_bal               4.301e-05  9.435e-07  45.585  < 2e-16 ***
out_prncp               1.431e-03  1.865e-03   0.767  0.44289
out_prncp_inv          -1.498e-03  1.866e-03  -0.803  0.42199
total_pymnt             2.002e+00  9.348e+00   0.214  0.83045
total_pymnt_inv         1.992e-02  1.727e-03  11.535  < 2e-16 ***
total_rec_prncp        -2.022e+00  9.348e+00  -0.216  0.82874
total_rec_int          -2.020e+00  9.348e+00  -0.216  0.82890
total_rec_late_fee     -2.012e+00  9.348e+00  -0.215  0.82959
recoveries             -2.021e+00  9.348e+00  -0.216  0.82881
collection_recovery_fee -1.348e-03  2.613e-04  -5.158 2.49e-07 ***
last_pymnt_amnt         5.166e-04  3.150e-06 164.006  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.245 on 212343 degrees of freedom
  (675003 observations deleted due to missingness)
Multiple R-squared:  0.4247,	Adjusted R-squared:  0.4246
F-statistic:  4898 on 32 and 212343 DF,  p-value: < 2.2e-16
```

From comparing summary output, the top 5 numeric features with the highest t values and lowest p values are:
last_pymnt_amnt
inq_last_6mths
total_rev_hi_lim
revol_bal
installment

Categorical features:

To select the top 5 categorical variables most predictive towards 'int_rate', first, select all the categorical variables and store into a new dataframe.

```
categories <- sapply(loan, is.character)
loan_categories <- loan[, categories]
summary(loan_categories)
length(loan_categories)
```

Since the int_rate is numerical values, we need to append it to the above dataframe 'categories'.

```
loan_categories['int_rate'] <-loan_nums['int_rate']
```

Then, check how many levels are in each variable. Delete any variables that only has one level (constant values), since it does not have any predictive power toward 'int_rate'.

```
sapply(loan_categories, function(col) length(unique(col)))
```

```
loan_categories_no_constant <- loan_categories[, sapply(loan_categories,
                                             function(x) {length(unique(x)) > 1})]
```

The output from the above sapply function showing number of levels in each variables is:

```
          term               grade             sub_grade
             2                   7                    35
     emp_title          emp_length        home_ownership
        299273                  12                     6
verification_status         issue_d          loan_status
             3                 103                    10
     pymnt_plan                 url                 desc
             2              887379                124471
       purpose               title              zip_code
            14               63146                   935
    addr_state     earliest_cr_line   initial_list_status
            51                 698                     2
  last_pymnt_d        next_pymnt_d     last_credit_pull_d
            99                 101                   104
application_type verification_status_joint    issue_year
             2                   4                     9
      int_rate
           542
```

As can see, some of the variables have too many levels (e.g. 'url', 'emp_title', 'title', etc.), and I realized that I need to preprocess the data.

Next, I tried to one-hot encoding to preprocess the data. However, RStudio showed error every time and session then got aborted. It was probably due to the laptop ran out of RAM.

To try another way, a another linear model was built with categorical variables that don't have too many levels (levels < 99), and the t values and p values associated with each variables towards predicting 'int_rate' were outputted below.

```
ols_cate = lm(int_rate~term + grade + sub_grade + emp_length + home_ownership +
           verification_status + loan_status + pymnt_plan + purpose +
           addr_state + initial_list_status + application_type + issue_year +
           verification_status_joint,loan_categories_no_constant)
summary(ols_cate)
```

The summary of the model is:

```
Call:
lm(formula = int_rate ~ term + grade + sub_grade + emp_length +
    home_ownership + verification_status + loan_status + pymnt_plan +
    purpose + addr_state + initial_list_status + application_type +
    issue_year + verification_status_joint, data = loan_categories_no_constant)

Residuals:
     Min       1Q   Median       3Q      Max
-20.1891  -0.2329  -0.0426   0.2618   5.9391

Coefficients: (7 not defined because of singularities)
                                        Estimate Std. Error  t value Pr(>|t|)
```

(Intercept)	2.799535	0.302543	9.253	< 2e-16	***
term 60 months	-0.010003	0.001441	-6.940	3.92e-12	***
gradeB	6.516217	0.004219	1544.686	< 2e-16	***
gradeC	9.504763	0.004394	2163.110	< 2e-16	***
gradeD	12.626303	0.005096	2477.511	< 2e-16	***
gradeE	15.911980	0.006515	2442.233	< 2e-16	***
gradeF	19.123404	0.010950	1746.436	< 2e-16	***
gradeG	20.160835	0.022130	911.017	< 2e-16	***
sub_gradeA2	0.708019	0.004902	144.438	< 2e-16	***
sub_gradeA3	1.372714	0.004854	282.778	< 2e-16	***
sub_gradeA4	1.757090	0.004457	394.215	< 2e-16	***
sub_gradeA5	2.554843	0.004253	600.675	< 2e-16	***
sub_gradeB1	-3.368163	0.003422	-984.285	< 2e-16	***
sub_gradeB2	-2.384828	0.003349	-712.085	< 2e-16	***
sub_gradeB3	-1.494138	0.003233	-462.206	< 2e-16	***
sub_gradeB4	-0.627663	0.003241	-193.664	< 2e-16	***
sub_gradeB5	NA	NA	NA	NA	
sub_gradeC1	-2.388301	0.003436	-695.179	< 2e-16	***
sub_gradeC2	-1.856051	0.003446	-538.538	< 2e-16	***
sub_gradeC3	-1.259382	0.003472	-362.676	< 2e-16	***
sub_gradeC4	-0.671803	0.003492	-192.365	< 2e-16	***
sub_gradeC5	NA	NA	NA	NA	
sub_gradeD1	-2.350548	0.004509	-521.261	< 2e-16	***
sub_gradeD2	-1.595050	0.004685	-340.457	< 2e-16	***
sub_gradeD3	-1.082602	0.004801	-225.509	< 2e-16	***
sub_gradeD4	-0.518531	0.004839	-107.164	< 2e-16	***
sub_gradeD5	NA	NA	NA	NA	
sub_gradeE1	-2.776285	0.006592	-421.180	< 2e-16	***
sub_gradeE2	-2.276885	0.006672	-341.242	< 2e-16	***
sub_gradeE3	-1.661149	0.006911	-240.362	< 2e-16	***
sub_gradeE4	-0.885423	0.007191	-123.122	< 2e-16	***
sub_gradeE5	NA	NA	NA	NA	
sub_gradeF1	-2.326271	0.011954	-194.604	< 2e-16	***
sub_gradeF2	-1.641633	0.012476	-131.583	< 2e-16	***
sub_gradeF3	-1.043825	0.012907	-80.874	< 2e-16	***
sub_gradeF4	-0.513765	0.013603	-37.769	< 2e-16	***
sub_gradeF5	NA	NA	NA	NA	
sub_gradeG1	-0.523912	0.024892	-21.047	< 2e-16	***
sub_gradeG2	-0.271966	0.025863	-10.516	< 2e-16	***
sub_gradeG3	-0.072319	0.027415	-2.638	0.00834	**
sub_gradeG4	-0.120578	0.029740	-4.054	5.03e-05	***
sub_gradeG5	NA	NA	NA	NA	
emp_length1 year	0.001970	0.002939	0.670	0.50274	
emp_length10+ years	0.013443	0.002223	6.046	1.48e-09	***
emp_length2 years	0.006701	0.002707	2.476	0.01330	*
emp_length3 years	0.008284	0.002788	2.972	0.00296	**
emp_length4 years	0.001797	0.003013	0.596	0.55087	
emp_length5 years	0.015925	0.002966	5.369	7.93e-08	***
emp_length6 years	0.021649	0.003205	6.755	1.43e-11	***
emp_length7 years	0.019138	0.003170	6.038	1.56e-09	***
emp_length8 years	0.014842	0.003184	4.662	3.14e-06	***
emp_length9 years	0.013832	0.003436	4.026	5.67e-05	***
emp_lengthn/a	0.008435	0.003223	2.617	0.00888	**

home_ownershipMORTGAGE	0.084142	0.301424	0.279	0.78013	
home_ownershipNONE	0.676544	0.310356	2.180	0.02927	*
home_ownershipOTHER	0.104414	0.303932	0.344	0.73119	
home_ownershipOWN	0.093615	0.301428	0.311	0.75613	
home_ownershipRENT	0.091318	0.301425	0.303	0.76192	
verification_statusSource Verified	-0.003593	0.001402	-2.563	0.01038	*
verification_statusVerified	0.022754	0.001488	15.296	< 2e-16	***
loan_statusCurrent	-0.097342	0.002743	-35.483	< 2e-16	***
loan_statusDefault	-0.006983	0.015171	-0.460	0.64532	
loan_statusDoes not meet the credit policy. Status:Charged Off	-2.078593	0.019974	-104.067	< 2e-16	***
loan_statusDoes not meet the credit policy. Status:Fully Paid	-1.559654	0.013090	-119.153	< 2e-16	***
loan_statusFully Paid	-0.002893	0.002738	-1.057	0.29055	
loan_statusIn Grace Period	-0.073520	0.007093	-10.365	< 2e-16	***
loan_statusIssued	-0.057002	0.006350	-8.977	< 2e-16	***
loan_statusLate (16-30 days)	-0.026279	0.011062	-2.376	0.01752	*
loan_statusLate (31-120 days)	-0.014581	0.005491	-2.655	0.00792	**
pymnt_plany	-0.043880	0.165116	-0.266	0.79043	
purposecredit_card	0.007051	0.005685	1.240	0.21489	
purposedebt_consolidation	-0.001811	0.005615	-0.322	0.74709	
purposeeducational	0.285799	0.026350	10.846	< 2e-16	***
purposehome_improvement	-0.003192	0.006032	-0.529	0.59669	
purposehouse	0.044887	0.010232	4.387	1.15e-05	***
purposemajor_purchase	0.001147	0.006825	0.168	0.86649	
purposemedical	0.015108	0.007933	1.904	0.05686	.
purposemoving	0.026314	0.009036	2.912	0.00359	**
purposeother	0.024916	0.006116	4.074	4.62e-05	***
purposerenewable_energy	0.019797	0.022475	0.881	0.37839	
purposesmall_business	-0.046565	0.007587	-6.137	8.40e-10	***
purposevacation	0.009893	0.009416	1.051	0.29341	
purposewedding	0.013226	0.012161	1.088	0.27679	
addr_stateAL	-0.010300	0.012166	-0.847	0.39718	
addr_stateAR	-0.002079	0.012834	-0.162	0.87132	
addr_stateAZ	-0.023952	0.011704	-2.047	0.04071	*
addr_stateCA	-0.025194	0.011215	-2.246	0.02468	*
addr_stateCO	-0.012264	0.011753	-1.044	0.29672	
addr_stateCT	-0.018925	0.011990	-1.578	0.11448	
addr_stateDC	-0.025927	0.015358	-1.688	0.09137	.
addr_stateDE	-0.011010	0.015237	-0.723	0.46994	
addr_stateFL	-0.025284	0.011318	-2.234	0.02549	*
addr_stateGA	-0.021358	0.011533	-1.852	0.06404	.
addr_stateHI	-0.004227	0.013541	-0.312	0.75490	
addr_stateIA	-0.251862	0.140262	-1.796	0.07255	.
addr_stateID	-0.289807	0.151336	-1.915	0.05549	.
addr_stateIL	-0.020141	0.011459	-1.758	0.07880	.
addr_stateIN	-0.012551	0.011978	-1.048	0.29470	
addr_stateKS	-0.012030	0.012571	-0.957	0.33857	
addr_stateKY	-0.024007	0.012471	-1.925	0.05424	.
addr_stateLA	-0.012825	0.012222	-1.049	0.29403	
addr_stateMA	-0.029405	0.011700	-2.513	0.01196	*
addr_stateMD	-0.029132	0.011687	-2.493	0.01268	*
addr_stateME	-0.099793	0.025367	-3.934	8.35e-05	***
addr_stateMI	-0.016075	0.011641	-1.381	0.16730	

```
addr_stateMN                           -0.015831   0.011862   -1.335  0.18202
addr_stateMO                           -0.014620   0.011951   -1.223  0.22120
addr_stateMS                           -0.101547   0.013971   -7.269 3.64e-13 ***
addr_stateMT                           -0.000344   0.015172   -0.023  0.98191
addr_stateNC                           -0.016891   0.011605   -1.456  0.14552
addr_stateND                           -0.079825   0.026328   -3.032  0.00243 **
addr_stateNE                           -0.076091   0.018863   -4.034 5.49e-05 ***
addr_stateNH                           -0.013104   0.013678   -0.958  0.33805
addr_stateNJ                           -0.027244   0.011482   -2.373  0.01765 *
addr_stateNM                           -0.012247   0.013372   -0.916  0.35976
addr_stateNV                           -0.025603   0.012064   -2.122  0.03381 *
addr_stateNY                           -0.025847   0.011285   -2.290  0.02200 *
addr_stateOH                           -0.015644   0.011525   -1.357  0.17467
addr_stateOK                           -0.012890   0.012544   -1.028  0.30416
addr_stateOR                           -0.019903   0.012192   -1.632  0.10259
addr_statePA                           -0.023182   0.011503   -2.015  0.04386 *
addr_stateRI                           -0.018554   0.013915   -1.333  0.18242
addr_stateSC                           -0.005989   0.012218   -0.490  0.62402
addr_stateSD                            0.016869   0.016547    1.019  0.30798
addr_stateTN                           -0.007013   0.012035   -0.583  0.56005
addr_stateTX                           -0.014627   0.011290   -1.296  0.19510
addr_stateUT                           -0.023837   0.012929   -1.844  0.06522 .
addr_stateVA                           -0.019460   0.011576   -1.681  0.09274 .
addr_stateVT                           -0.007491   0.016592   -0.451  0.65164
addr_stateWA                           -0.017842   0.011732   -1.521  0.12832
addr_stateWI                           -0.008462   0.012132   -0.697  0.48550
addr_stateWV                           -0.002747   0.013632   -0.202  0.84027
addr_stateWY                            0.001736   0.016063    0.108  0.91393
initial_list_statusw                   -0.061073   0.001236  -49.415  < 2e-16 ***
application_typeJOINT                  -0.022247   0.040433   -0.550  0.58217
issue_year2008                          1.073269   0.024007   44.707  < 2e-16 ***
issue_year2009                          2.677300   0.023150  115.650  < 2e-16 ***
issue_year2010                          2.106747   0.022593   93.246  < 2e-16 ***
issue_year2011                          2.593007   0.022659  114.438  < 2e-16 ***
issue_year2012                          3.983333   0.022490  177.115  < 2e-16 ***
issue_year2013                          4.166249   0.022444  185.626  < 2e-16 ***
issue_year2014                          3.209039   0.022443  142.984  < 2e-16 ***
issue_year2015                          2.470872   0.022456  110.033  < 2e-16 ***
verification_status_jointNot Verified  -0.047563   0.050957   -0.933  0.35062
verification_status_jointSource Verified  0.078993   0.078112    1.011  0.31188
verification_status_jointVerified             NA         NA       NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.522 on 887240 degrees of freedom
Multiple R-squared:  0.9858,	Adjusted R-squared:  0.9858
F-statistic: 4.466e+05 on 138 and 887240 DF,  p-value: < 2.2e-16
```

From comparing summary output, the top 5 numeric features with the highest t values and lowest p values are:

grade
sub_grade

issue_year
loan_status
initial_list_status

So, ‘last_pymnt_amnt’, ‘inq_last_6mths’, ‘total_rev_hi_lim’, ‘revol_bal’, ‘installment’ are the 5 numerical, and ‘grade’, ’sub_grade’, ‘issue_year’, ‘loan_status’, ‘initial_list_status’ are the 5 categorical variables that are most predictive to interest rate give the higher t values and lower p values from the regression models, which indicates these variables are meaning additions to the model in predicting the response variable ‘int_rate’.