0. Data Preprocessing
    a. First, I read the csv file “techSalaries2017.csv”in and do a sanity check to check that if there is NULL or 0 in the outcome variable “totalyearlycompensation.” This is to see whether I need to clean the NULL value of outcome variable when I do the regression. I found out there is not NULL or 0 value in the outcome variable.

```
# check whether we need to consider the case whether our predictor is null for single regressic
print(df["totalyearlycompensation"].isnull().any())
print(0 in df["totalyearlycompensation"].values)
```

    It means that when doing regression, I don’t need to clean the outcome variable. However, I may still need to clean the predictors.
    b. Second, I computed the # of lines of data in the csv and the # of lines of gender that is not categorized as “Male” or “Female”. This is to see whether I could only encode Male and Female and leave other value as NA. I found out the # of total records is 62642 and the # of other genders reported is only 401, far less than the total records.

```
[6]: # compare number of other genders with total records
     print(df.shape)
     print(df.groupby('gender').totalyearlycompensation.count())
```

```
(62642, 27)
gender
Female                                6999
Male                                 35702
Other                                  400
Title: Senior Software Engineer          1
Name: totalyearlycompensation, dtype: int64
```

    Therefore, and also according to the question 4, I restricted the gender variable to “Male” and “Female” and encoded them to 0 and 1, encoding other values to NAN.

```
[25]: # encode gender
      def encodeGender(x):
          if(x == 'Male'):  return 0
          elif(x == 'Female'): return 1
      df['gender'] = df['gender'].apply(lambda x: encodeGender(x))
      print(df['gender'])
      print(df['gender'].unique())
```

```
0        NaN
1        NaN
2        NaN
3        NaN
4        NaN
         ...
62637    NaN
62638    NaN
62639    NaN
62640    NaN
62641    0.0
Name: gender, Length: 62642, dtype: float64
[nan  0.  1.]
```

    c. Third, I computed the median of “totalyearlycompensation”. This is to prepare for question 5 — to predict if the compensation is high based on some predictors. I found out the median is 188,000. Then I created a new ‘high’ outcome variable for all the data records and if the compensation is greater or

equal than 188,000 I made it 1 and 0 otherwise.

```
[27]: # create the new high outcome variable
      def ifHigh(x):
          if(x >= 188000): return 1
          elif(x < 188000): return 0
      df['high'] = df['totalyearlycompensation'].apply(lambda x: ifHigh(x))
      print(df['high'])

      0        0
      1        0
      2        1
      3        1
      4        0
              ..
      62637    1
      62638    1
      62639    1
      62640    1
      62641    1
      Name: high, Length: 62642, dtype: int64
```

1. Using multiple linear regression: What is the best predictor of total annual compensation, how much variance is explained by this predictor vs. the full multiple regression model?
    a. To answer this question, I ran 18 single linear regression models on each 18 variable with not null records and 1 multiple linear regression on all the 16 variables with not null records. The 18 variables included 'yearsofexperience', 'yearsatcompany', 'gender', 'Masters_Degree', 'Bachelors_Degree', 'Doctorate_Degree', 'Highschool', 'Some_College', 'Race_Asian', 'Race_White', 'Race_Two_Or_More', 'Race_Black', 'Race_Hispanic', 'Age', 'Height', 'Zodiac', 'SAT', and 'GPA'. I used r^2 to evaluate each of the 18 models. The 16 variables is the same as above but excludes “Highschool” and “Race_Two_Or_More.” Before I ran the multiple linear regression, I split the whole cleaned data to 80% train and 20% test set. I validated the multiple regression model with r-squared and I also computed its mean squared error (MSE). Particularly, I extracted the variable with highest beta of the multiple regression model to take a further look.
    b. The reason I first ran the single regression is to get a general idea how the single predictor contributes to the outcome. The reason I drop “Highschool” and “Race_Two_Or_More” in the multiple regression is to avoid model overdetermination because there are 5 education and race category and we only have 4 degree of freedom. The reason I evaluated the models with r^2 is that r^2 indicates how many variance of outcome variable is explained by the predictor, which is exactly being asked by the question. The reason I first prepared the test and train set for the multiple regression is that this split will be reused to fit ridge and lasso to compare the performance of the model. The reason I normalized the data in the multiple regression is that it could make beta of predictors more reasonable and give a standard output of MSE which would be easy for me to compare it with the MSE from 2nd and 3rd question. The reason I extracted the beta of 'yearsofexperience' is to see if the result of multiple regression aligns with the result of single regression.

c. I found that the best single predictor is 'yearsofexperience' with the highest r^2 of 0.179, indicating that 17.9% of the variance of the outcome variable is explained by this predictor.

```
best predictor:  yearsofexperience ; r^2:  0.1788267653287875
```

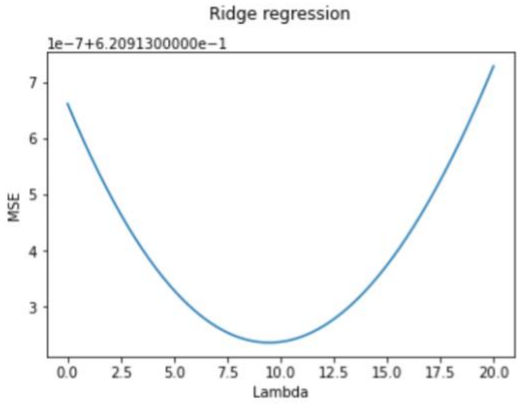
For the multiple regression (OLS), the r^2 is 0.284 and MSE is 0.621. Hence, 28.4% of the variance of outcome variable is explained by the full predictors. The beta of 'yearsofexperience' is 0.389 which has the greatest absolute value among the betas of other predictors. This result aligns with the single regression model because a greater beta of predictor means a greater contribution to the outcome variable.

```
total R^2 =  0.28443484759564897
total MSE =  0.6209136619634104
beta =  [ 0.38872095 -0.05539633 -0.00782092  0.12226189  0.03850138  0.15737497
  0.00694138 -0.04628727 -0.03233076 -0.01570833 -0.0180951  -0.00375198
 -0.00647966  0.01203714  0.26074659  0.01767211]
```

d. According to single and OLS, yearsofexpereience is the best predictor. However, running OLS will give a better prediction than the single predictor case because it has a higher r^2.

2. Using ridge regression to do the same as in 1): How does the model change or improve compared to OLS? What is the optimal lambda?
    a. To answer this question, I first did the hyperparameter tuning with the prepared train and test normalized dataset to find the optimal lambda. After I found the optimal lambda, I ran a ridge regression with that lambda on the 16 variables as the OLS used with the prepared test normalized dataset, and I evaluated the model by using MSE.
    b. The reason I ran ridge on full predictors is that running ridge on single predictor does not make sense because it is often used to regularize multiple parameters and not a single one. The reason I did hyperparameter tuning is to find the optimal lambda that could give the model the lowest MSE. The reason I used MSE to evaluate the model because MSE is one of the model's metrics and it will be easy to compare the result with other questions.
    c. I found that the optimal lambda is 9.46. and the MSE is 0.620, which is lower than the MSE from OLS by 0.001. Even though there is some improvement but it

is very small.

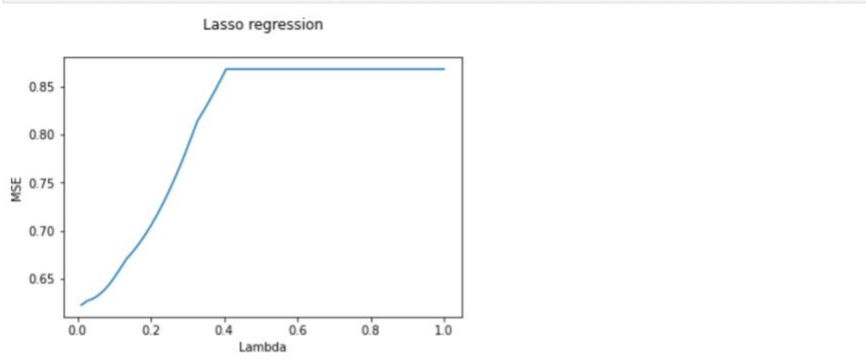



```
optimal_lambda:  9.46
total MSE =  0.6198337164372303
beta:  [ 0.38660262 -0.05752978 -0.01408504  0.12166705  0.03942418  0.15868238
  0.00443147 -0.05257047 -0.03734839 -0.02090044 -0.0240739  -0.00279029
 -0.00550065  0.01263361  0.25681038  0.01652383]
```

d. I think for this problem, it is not necessary to use ridge regression because it only gives us a 0.001 improvement of MSE, which is not very significant.

3. Using Lasso regression to do the same as in 1): How does the model change now? How many of the predictor betas are shrunk to exactly 0? What is the optimal lambda now?
    a. To answer this question, I first did the hyperparameter tuning with the prepared train and test normalized dataset to find the optimal lambda. After I found the optimal lambda, I ran a lasso regression with that lambda on the 16 variables as the OLS used with the prepared train and test normalized dataset, and I evaluated the model by using MSE.
    b. The reason I ran Lasso on full predictors is that running lasso on single predictor does not make sense because Lasso is often used to do features extraction. The reason I did hyperparameter tuning is to find the optimal lambda that could give the model the lowest MSE. The reason I used MSE to evaluate the model because MSE is one of the model's metrics and it will be easy to compare the result with other questions.
    c. I found that the optimal lambda in the range [0.01,1,1001] is 0.01. (In fact, by trying range with high granularity I found the optimal lambda converges to 0). And the total MSE is 0.736. The betas of 'Some_College', 'Race_Asian', 'Race_White', 'Race_Black', 'Race_Hispanic', 'Age', 'Height' has shrinked to 0 with

this lambda, as shown below.




```
optimal_lambda:  0.01
total MSE =  0.7364831656724227
beta:  [ 0.36651721 -0.03535304 -0.00384126  0.06265668 -0.00770337  0.12844365
 -0.         -0.          0.         -0.         -0.          0.
 -0.          0.00253833  0.25414972  0.01171298]
```

d. In fact, the total MSE of the model is greater than OLS, and since the optimal lambda converges to 0 and we shouldn't consider negative lambdas(if so we are overfitting the data), we should use OLS instead of Lasso for this question. Second, I think that some of the betas has shrinked to zero means we did the feature extraction: the nonzero features left is the features that could have a relative major impact on the outcome variable, which are 'yearsofexperience', 'yearsatcompany', 'gender', 'Masters_Degree', 'Bachelors_Degree', 'Doctorate_Degree', 'Zodiac', 'SAT', and 'GPA'.

4. There is controversy as to the existence of a male/female gender pay gap in tech job compensation. Build a logistic regression model (with gender as the outcome variable) to see if there is an appreciable beta associated with total annual compensation with and without controlling for other factors.
    a. To answer this question, I first encoded the gender to 0 and 1 (specified in 0.a). Then I ran 2 logistic regressions. One has only the total annual compensation (not controlling for other factors) as the predictor, the other one has all the variables including 'totalyearlycompensation', 'yearsofexperience', 'yearsatcompany', 'Masters_Degree', 'Bachelors_Degree', 'Doctorate_Degree', 'Some_College', 'Race_Asian', 'Race_White', 'Race_Two_Or_More', 'Race_Black', 'Race_Hispanic', 'Age', 'Height', 'Zodiac', 'SAT', and 'GPA' (controlling for other factors) as the predictors. And I evaluated the model using precision and recall score. In the end, I printed the betas of the "totalyearlycompensation" variable in both models.
    b. The reason why I encoded the gender to 0 and 1 is because of hint 4. The reason why I normalized the total year compensation is to make input data and beta appears reasonable. The reason why I extracted the beta from the model is to see if there is an appreciable beta in 2 different models and compared them.
    c. Even though the precision and recall score of both models are similar, the beta of total year compensation is very different. I found that beta of total year compensation is -0.1627 without control and is -0.0362 with control, as shown below.

```
Precision = 18.6%, Recall = 65.2%
beta without control:  -0.16273131609588215

Precision = 21.1%, Recall = 65.4%
beta with control: -0.03619834300603163
```

d. It means that, if we add other factors in the logistic regression model, the beta of total annual compensation drops. It indicates that the ability of using "total annual compensation" to predict the gender becomes weaker if we take other factors into account. I would say there is an appreciable beta when we only predict the gender based on total annual compensation, which is -0.1627. However, there is not an appreciable beta when we also take other predictors into account, which is -0.0362. Therefore, I wouldn't say there exists a male/female gender pay gap in tech job compensation because if we take all the factors to classify someone as male/female, yearly compensation is not a strong predictor that helps us do that.

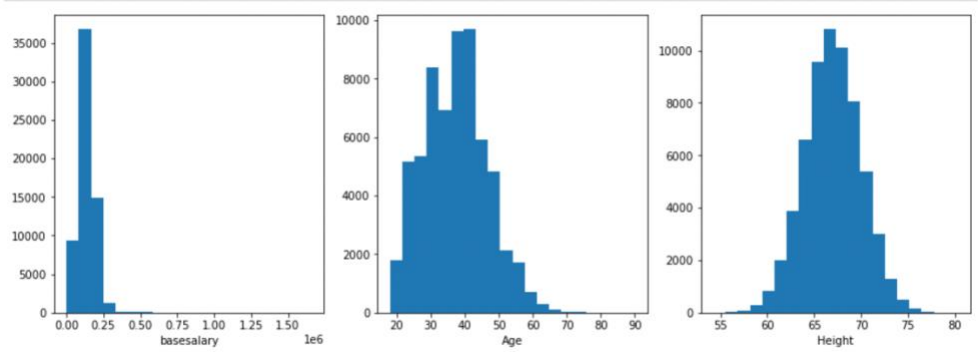
5. Build a logistic regression model to see if you can predict high and low pay from years of relevant experience, age, height, SAT score and GPA, respectively.
   a. To answer this question, I first created a new column called 'high'. (specified in 0.b). Then I normalized the predictor data and split the data to be training and test set. I ran 5 logistic regression models on the train set with years of experience, age, height, sat, and GPA, respectively, as asked in the question. Finally, I used precision and recall score metrics to evaluate the models.
   b. The reason why I split the data is I need to both train the model and validate the model, and so I need train and test dataset to do that. The reason why I evaluated the model with precision and recall score is that this is a classification problem(the outcome variable only has binary value), not a regression problem. MSE will not work here. And precision and recall score are both good ways to validate the classification model.
   c. I found the result as shown below.

```
yearsofexperience : Precision = 68.1%, Recall = 61.4%
Age : Precision = 58.5%, Recall = 56.8%
Height : Precision = 50.6%, Recall = 50.8%
SAT : Precision = 59.0%, Recall = 63.6%
GPA : Precision = 58.8%, Recall = 62.4%
```

   d. Precision score means the percentage of predicted high compensation we classify correctly, and the recall score means the percentage of real high compensation we classify correctly. And based on the result, yearsofexperience has the highest precision score, and SAT has the highest recall score. All of the predictors excepts for height has the precision and recall score obviously higher than 50%. Hence, if we allow a precision and recall score of 55%, we can use those predictors to predict high or low pay. Personally, I would choose yearsofexperience as the best predictor to classify because it has an outstanding

higher precision score than others and SAT's recall score is not outstandingly high.

6. EC1: Is salary, height or age normally distributed? Does this surprise you? Why or why not?
    a. To answer this question, I plot the histogram of the three variables "base salary", "age", and "height".
    b. The reason why I plot histogram is that histogram can gives us the shape of the distribution of the random variable to determine if it is normally distributed or not.
    c. The result is shown below.



    d. I think the base salary and age is not normally distributed but height is normally distributed because only it has a bell shape. It is easy to understand that the base salary is right skewed because only few people earn very high in the real world(80-20 rule). However, the age distribution surprises me because I don't know there are some elder people working on the industry even when they are above 70 years old until I see this data. The height is normal does not surprise me because it is objective and depends on nature, and it should be a normal distribution.

7. EC2: Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious.
    a. I want to find out what's the first 10 states in the US that have the highest average annual compensation and which state has the most influence on the 'high' annual compensation. To do this, I first found out which job title has the max number of records, which is software engineer with a total of 41231 lines. Then, I encoded the 'location' to 'state' in the US and removed the data records that is not in the US. Then I computed the average annual compensation of the different states and sort them in descending order to pick the highest 10. In the end, I ran a logistic regression with all the state dummy variables computed from 'state' to classify high and low compensation and found the state with the highest beta.
    b. The reason why I selected the job title that has the max number of rows is because it allows me to get the largest number of data to fit the model. And a uniform job title can better show the compensation discrepancy because of the difference in locations. The reason I computed the average is that I tried to find

out the first 10 states that has the highest compensation. The reason I used a logistic regression to classify high or low based on states and to find the highest beta is to see which state has the most appreciable beta and then has the highest influence when classifying high or low compensation based on states.

c. The result is shown below.

	state	totalyearlycompensation
3	CA	263195.505543
45	WA	229083.084242
33	NY	224675.973608
4	CO	186606.635071
18	MA	183038.994801
10	HI	180333.333333
36	OR	171389.408100
25	MT	166200.000000
37	PA	161000.000000
42	UT	160189.349112

```
State with the highest absolute beta:  CA , beta:  2.3768332248247304
```

d. The result gives a list of top 10 states with highest compensation and the first three are CA, WA and NY. From the logistic regression, we can see there is an appreciable beta in front of CA, which is positive 2.377. This also aligns with the result of the states with average highest compensation we just found. Since the high is 1 and low is 0, the result means that software engineer in CA is most likely to be predicted as having high compensation in the technology industry.