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?

First, I converted gender into dummy variables: 0 as Male, 1 as Female, 2 as Others. I also converted Education to ordinal, 0 being the lowest and 4 the highest. To find the explained variance for the continuous features, I simply computed the correlation matrix. For Education I used spearman's correlation. For all other binary variables I used point biserial correlation. These choices were because linear correlation does not capture the association between a discrete and a continuous variable well. The result is that **years of experience** is the best predictor, having a correlation of 0.403, or r^2 of 0.162. For the full multiple linear regression model, I dropped the original gender. I also dropped race and education but kept the dummy version of them. Then I dropped all rows contating nan values because the amount of nan values (over 50% of data) is impossible to impute. Then, I standardized the continuous predictors so that they will be in the same range. I also scaled the target by a factor of 1e-5 to have a easier read. Using score function, I found the r^2 of the model to be 0.267. It is better than just using years of experience.

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?

First, using sklearn's train test split, I divided the dataset into 80/20 train/test. To find the optimal lambda, I looped through lambdas np.linspace(0, 30, 1000) and used each to build a ridge model and stored the MSEs. Then I found the lambda corresponding to the model with the lowest MSE, which was 2.823.

The comparison Ridge and OLS betas:

	predictor	ridgeBeta	olsBeta
0	yearsofexperience	0.523982	0.532678
1	yearsatcompany	-0.078308	-0.094525
2	Masters_Degree	-0.003061	-0.159531
3	Bachelors_Degree	-0.226917	-0.374875
4	Doctorate_Degree	0.716581	0.575971
5	Highschool	-0.299228	-0.511755
6	Some_College	-0.321484	-0.510624
7	Race_Asian	-0.018917	0.137809
8	Race_White	0.007549	0.182910
9	Race_Two_Or_More	0.152768	0.326564
10	Race_Black	-0.051069	0.122327
11	Race_Hispanic	-0.037231	0.126656
12	Age	-0.005325	-0.006876
13	Height	-0.011140	-0.000498
14	Zodiac	0.021823	0.013973
15	SAT	0.343872	0.344580
16	GPA	0.023066	0.019654
17	0.0	-0.009401	-0.079599
18	1.0	-0.061979	-0.117367
19	2.0	0.071380	0.196966

To compare their performances, I ran 1001 tests each with different 80/20 split. What I found was that OLS could produce extraordinarily high MSEs, while ridge was

very consistent. The mean of OLS MSEs was 3.16e18, but the median was 1.178. The mean of Ridge MSE was 1.309, std was 0.275 and the median was 1.180. To conclude, Ridge performs much better than OLS because Ridge was much more consistent.

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?

For this model, I used the same X and y as question 1 and 2, same train and test split as 2. For lambdas I used np.linspace(1e-5, 30, 1000). 1e-5 because sklearn does not allow 0. However, I have tried many times cross validating with different train and test sets, the optimal lambda was always 1e-5 (near zero). The betas comparison is below.

	predictor	lassoBeta	olsBeta
0	yearsofexperience	0.514652	0.521562
1	yearsatcompany	-0.080931	-0.075527
2	Masters_Degree	0.216790	-0.029444
3	Bachelors_Degree	-0.000000	-0.249831
4	Doctorate_Degree	0.906597	0.656479
5	Highschool	-0.051745	-0.332724
6	Some_College	-0.030740	-0.268425
7	Race_Asian	-0.019224	0.120452
8	Race_White	0.029967	0.158773
9	Race_Two_Or_More	0.150170	0.279269
10	Race_Black	-0.015388	0.121736
11	Race_Hispanic	0.000000	0.117802
12	Age	0.000521	-0.005627
13	Height	-0.006755	-0.008442
14	Zodiac	0.010583	0.019990
15	SAT	0.329014	0.348001
16	GPA	0.038019	0.016451
17	0.0	0.000000	-0.058197
18	1.0	-0.044325	-0.108810
19	2.0	0.089337	0.167006

Bachelors_Degree, Race_Hispanic, and 0.0(Being Male or not) were shrunk to exactly 0. The performance of Lasso was similar to Ridge. The mean MSE over 1001 tests was 1.299 with a standard deviation of 0.262.

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.

First, I build a logistic regression model with just total yearly compensation (without controlling for other factors). The beta of the predictor is -0.115, meaning that for every 100k increase in salary, the log odds that the person is female changes by -0.115, which translates to a change in odds by a factor of 0.891. In plain English, the higher the salary, the less likely, by a very tiny margin, the person is a female. Furthermore, the AUC of the model is exactly 0.5, suggesting that this model is no better than randomly guessing.

Then, I added all other predictors to the model. Below are the betas ranked from highest to lowest absolute value.

	feature	beta
2	Education	-0.071744
5	GPA	-0.067068
1	yearsofexperience	-0.023805
0	totalyearlycompensation	-0.018286
3	Age	-0.006538
4	SAT	-0.000524

Note that totalyearlycompensation is not standardized and has a wider range than the other predictors. Changing the scale of this predictor, for instance, by a factor of 1e-3 instead of 1e-5, makes the beta -0.0012. We also see that a lower value in other features predicts higher probability of the person being female. Although it is true that females are more likely to be paid less (by a small margin), perhaps this is due to less experience, education, etc.?

To conclude briefly, based on the logistic regression model, salary has a very little or perhaps negligible effect on predicting the person's gender, and the inverse stands true. However, it is important to note that this result is only applicable to the population this dataset belongs to— the California tech industry.

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.
   First I build a logistic regression model with all the predictors in this question. Train/test = 0.8/0.2. AUC = 0.685. Betas:

	predictor	beta
0	yearsofexperience	0.142449
1	Age	-0.005169
2	Height	-0.062414
3	SAT	0.003172
4	GPA	-0.000872
5	gender	-0.001687

   The AUC 0.685 suggests that the model is somewhat useful.

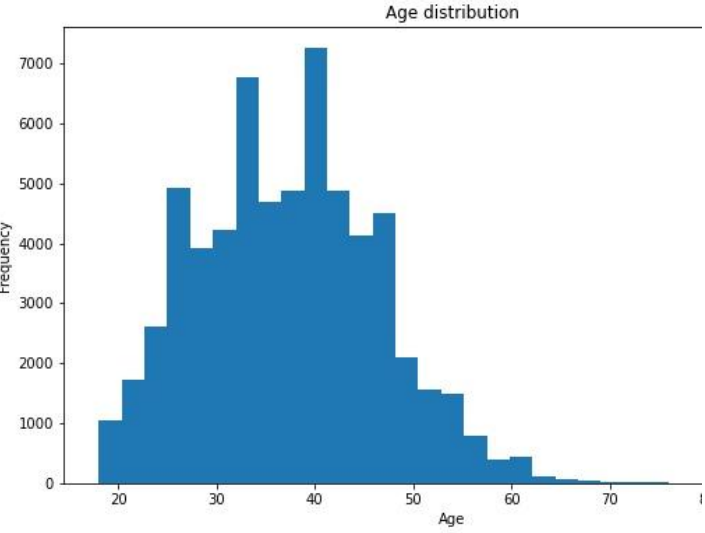
   Then, I build different models with each of the predictors. Below are the Betas of each predictor as the only predictor for a model and AUCs.

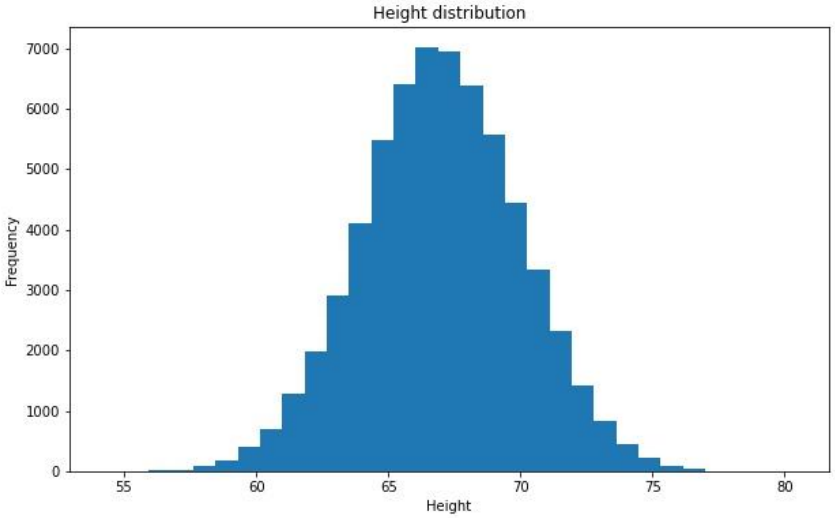
	Beta	AUC
yearsofexperience	0.153003	0.667881
Age	0.051647	0.585103
Height	0.000129	0.500000
SAT	0.000056	0.500000
GPA	1.677976	0.594480

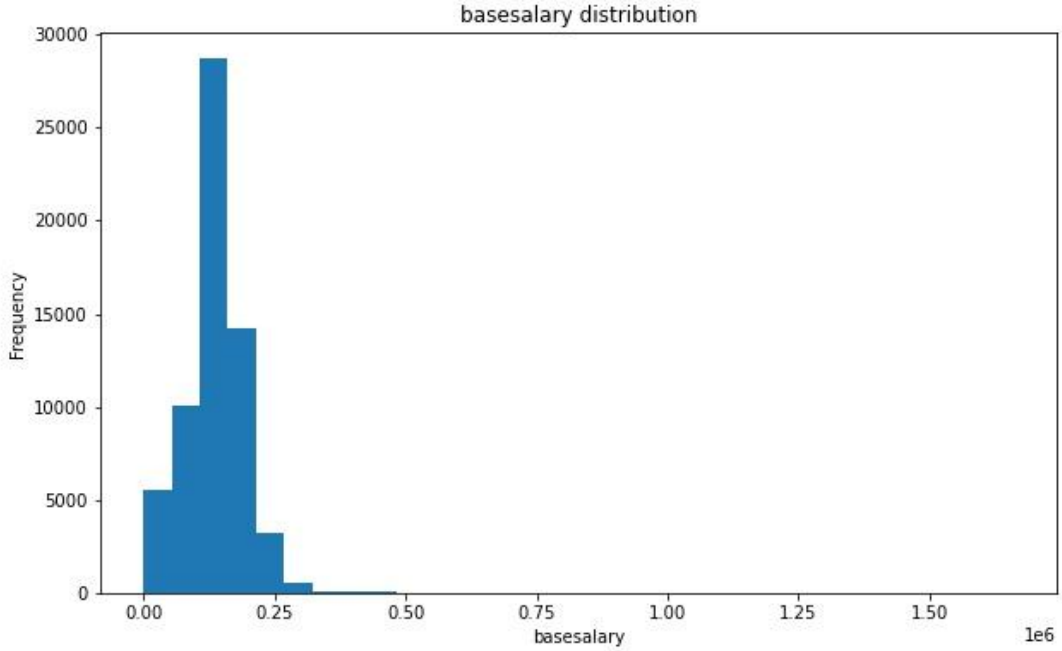
The predictor years of experience alone performs almost as well as the full model. Surprisingly, as the predictor with the lowest beta magnitude in the full model, GPA comes second best in terms of individual performance.

Extra credit:

a) Is salary, height or age normally distributed? Does this surprise you? Why or why or why not?







Above are the histograms for height, age and salary. Height is the only one that is normally distributed (almost perfectly). It does not surprise me because height is normally distributed in the human population, so it is not strange that people in tech industry, not a height demanding industry such as NBA, follows the distribution of the larger population. The same argument applies to the other two. Neither age nor salary is normally distributed among the human population.

b) Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious.

I created separate dataframes based on race and stored them in a dictionary. Then, I build different models, each for an individual race, with predictors in question 5 plus education. The target is high/low pay. Below is the result: column represents the race each model corresponds to. The indices are the features the betas correspond to except for AUC, which is AUC.

	Asian	White	Hispanic	Mixed	Black
yearsofexperience	0.131284	0.133003	0.122491	0.162579	0.139827
Education	0.698454	0.316902	0.182559	0.430228	0.317380
Age	-0.005925	0.004171	-0.010232	-0.004704	0.005820
Height	-0.049839	-0.032249	-0.035511	0.008189	-0.083609
SAT	0.005812	0.005042	0.006139	0.005823	0.001105
GPA	-1.131416	-0.841386	-1.552662	-0.600755	1.029000
AUC	0.672525	0.714011	0.711858	0.721714	0.676969

What I found interesting is that education matters a lot more for Asians than other group. Furthermore, interestingly, the magnitude of beta for GPA when we break separate the entire sample according to race is significantly larger than the GPA beta of the model trained with the entire sample.