Regression Analysis of Gene and Environmental Variables

Group 7
Tingting Xuan Han Zhou

AMS 578
Regression Analysis, Spring 2018
Multiple Regression Computing Project

Abstract

The project is to analyze the given data about patient records and find a best multi-linear model to predict the dependent variable Y. This report shows the regression analysis methods we use and the final result we get. First chapter is introduction which gives some basic information about our data and also the motivation of our study; second chapter is methods introduction which shows exactly the steps we take to get our final formula; third chapter shows the results we get and the validation of our model; and the final chapter gives the conclusion and discussion.

Introduction

There are three files given for this project. The first file contains the patient identifier and the dependent variable Y, which is a continuous variable. The second file contains the patient identifier and values of six environment variables called E1 to E6, which are all continuous variables. The third file contains the patient identifier and the twenty-five independent indicator variables called G1 to G25, which are binary variables with value 0 or 1. There are 1399 records of patients with random order in total which consists of 32 variables. And there are 26.96% uncomplete records in our data. These 32 variables include one Y variable which was served as the dependent variable and 6 environmental variables (listed from E1 to E6) and 25 gene variables (listed from G1 to G25) which were treated as our independent variables. What we did next was trying to combine all the data together and apply R language to find a model best fitted our dataset. The R programming language was used for data processing and analysis.

The motivation for this analysis comes from the field of mental illness where researchers have long sought to model depression (among other ailments) with genetics. A paper named "Influence of Life Stress on depression: Moderation by a Polymorphism in the 5-HTT Gene" (2003, Avshalom Caspi et al) was provided and aimed to predict depression in adults with environmental variables as well as genetic factors. The results showed that gene-environment interaction in which the presence of a certain genotype would multiply an environmental variable and came up with a highly significant p-value when added to the model. However, another report written by Neil Risch et al. found that the number of stressful life events was significantly

associated with depression. No association was found between 5-HTTLPR genotype and depression in any of the individual studies nor in the weighted average and no interaction effect between genotype and stressful life events on depression was observed.

Methods

1. Data cleaning and processing

The first task in our analysis was to load the three data sheets in R, merge all of them and delete the uncomplete records. We found that three data sheets all contain the variable ID, and in each sheet, every ID shows once and no absence. So, we first sorted each data sheets by patient ID, and used the function *cbind()* to combine all variables together. Then the function *complete.cases()* was used to find the index of observations, which have no missing values across the entire row. Picking up the records with the index which we got just now, a new data frame was created and we named it as "mydata". And after deleting the rows of missing values, there are 1022 observations left with 32 variables, which is 73.05% of the total number of data.

2. Checking the correlation among the variables

Correlation coefficients between the independent variables and the dependent variables are calculated in order to have a general view of which variable may have strong correlation with Y. Here we listed all correlations.

	E1	E2	E3	E4	E5	E6	G1	G2
correlation	0.079754	0.103268	0.584638	0.141554	-0.00549	0.090456	-0.02901	0.02733
	G3	G4	G5	G6	G7	G8	G9	G10
correlation	0.022716	-0.03376	0.026859	0.010586	-0.03033	-0.01699	0.030074	-0.0549
	G11	G12	G13	G14	G15	G16	G17	G18
correlation	0.045257	0.025728	0.01209	-0.00421	0.017228	-0.01498	0.007634	0.03764
	G19	G20	G21	G2	G23	G24	G25	
correlation	-0.00516	-0.0212	-0.0072	0.012897	-0.02156	0.015396	-0.04313	

We found that the correlation between E1, E2, E3, E4, E6 and Y are over 0.05 respectively, and the correlation between E3 and Y are 0.585 which implied that E3 and Y are highly correlated.

So we should put more focus on the variables listed above. We calculated the correlation coefficients among the independent variables as well and found that the absolute value of all correlation coefficients were around or smaller than 0.1 which indicated that there are very little or no multicollinearity among the independent variables.

3. Constructing linear regression models

In this part, I will show you the details about what methods we used and how to get our final model step by step. We use some variable selection methods in this project. First, I want to give some background knowledge about these methods. And then I will show the formula we get for each step.

(a) Background Knowledge about Variable Selection Methods

I. Akaike Information Criterion (AIC)

The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection. AIC is founded on information theory: it offers an estimate of the relative information lost when a given model is used to represent the process that generated the data.

Suppose that we have a statistical model of some data. Let k be the number of estimated parameters in the model. Let $\hat{L}$ be the maximum value of the likelihood function for the model. Then the AIC value of the model is the following.

$$\mathrm{AIC} = 2k - 2\ln\hat{L}$$

Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. Thus, AIC rewards goodness of fit (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of the number of estimated parameters. The penalty discourages overfitting, because

increasing the number of parameters in the model almost always improves the goodness of the fit.

II. Stepwise Selection

In statistics, stepwise regression is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure. In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some pre-specified criterion. In our project, we use AIC as the pre-specified criterion.

There are three ways to conduct stepwise selection. (1) Forward selection, which involves starting with no variables in the model, testing the addition of each variable using a chosen model fit criterion, adding the variable (if any) whose inclusion gives the most statistically significant improvement of the fit, and repeating this process until none improves the model to a statistically significant extent. (2) Backward elimination, which involves starting with all candidate variables, testing the deletion of each variable using a chosen model fit criterion, deleting the variable (if any) whose loss gives the most statistically insignificant deterioration of the model fit, and repeating this process until no further variables can be deleted without a statistically significant loss of fit. (3) Bidirectional elimination, a combination of the above, testing at each step for variables to be included or excluded.

(b) Our Results in Each Step

I. Linear Part

First, we set up a full linear model which predicts Y using all predictive variables.

fit.linear.full <- lm(formula = Y ~ ., data = mydata)

The R^2 and Adjust R^2 for this full linear model are 0.39 and 0.3709. And the p-value for F-test is smaller than 2.2e-16, which means there is significant linear relation between Y and these predictive variables.

And then we took a look at each coefficient and its significance. We found that for most variables, they are not significant. There are only 6 significant variables with p-value smaller than 0.05, which are E1, E2, E3, E4, E6 and G20. So, we should pay more attention to them in the further steps. The p-value of these variables showed in the following table.

II. Quadratic Part

Second, we set up a full quadratic model which predicts Y using all predictive variables, their square and the cross-product of each pair of variables.

fit.quad.full <- lm(formula = Y ~ .^2, data = mydata)

The R^2 and Adjust R^2 for this full linear model is 0.6874 and 0.3922. We can notice that the R^2 for quadratic model is much bigger than the linear model, which means the quadratic terms give some important information for prediction. But since it also introduces massive number of parameters and most of them are actually useless, so the Adjust R^2 doesn't change much. So what we have to do now, is to reduce the complexity of this big quadratic formula.

And we used the stepwise selection method with AIC, which has already introduced in the former part, to achieve our goal. Here we use the *step()* function in R.

step(null, scope = list(upper = fit.quan.full), data = mydata, direction = "both")

And it gave the best model based on AIC, which is

formula = Y ~ E3 + E4 + E2 + E6 + E1 + G20 + E2:E6 + E4:E1 + E3:E2

And the coefficient and p-value of each term shows in the following table.

	Estimate	P-value
(Intercept)	4.78E+07	9.55E-01
E3	1.12E+07	3.34E-67
E4	1.68E+06	1.24E-02
E2	-7.65E+05	4.12E-01
E6	6.65E+05	2.57E-01
E1	7.75E+04	9.05E-01
G20	-1.19E+09	4.03E-02
E2:E6	2.66E+03	1.25E-03
E4:E1	1.71E+03	3.96E-02
E3:E2	1.52E+03	6.95E-02

Since choosing variables is a problem related to multi-comparison, by the Bernoulli theorem, the restriction for the significance will be stricter. So we need to get rid of the variables with p-values which are not smaller enough.

What we did here also using the stepwise selection method with backward elimination. We found the variable with the biggest p-value and dropped it to get a new model. We did this one by one to eliminate terms with p-value smaller 0.005. Finally, we ended up with a model in which there were only terms with p-value smaller that 7.31e-09. And the final model is

final_quad_model <- lm(formula = Y ~ E3 + E2:E6 + E4:E1, data = mydata)

And the coefficient and p-value of each term shows in the Result part.

The R^2 and Adjust R^2 of this model are 0.6352 and 0.6342. Compared to the full quadratic model, the R^2 doesn't change much, while the Adjust R^2 increased a lot. So it's a proof that this model we got has a good performance.

III. Cubic part

Finally, we worked on the cubic model. Since there are 31 predictive variables, so the full cubic formula will have 29791 terms, which is much more than the number of records we have and not allowed. We cannot use 1022 observations to estimate more than 1022 parameters. So what we did is to select the some more crucial variables first and build the full cubic model using what we selected. Considering the significant variables in the full linear model and the ones chosen

by the step function in quadratic model, there are only 6 variables left (E1, E3, E4, E2, E6, G20), which have a strong relation with the dependent variable Y.

Using these 6 variables to generate a cubic model with all linear, quadratic and cubic terms.

```
data1 <- as.data.frame(cbind(Y, E1, E2, E3, E4, E6, G20))
fit.cubic.full <- lm(Y ~ (.)^3, data = data1)
```

Based on the result we got, the R^2 and Adjust R^2 are 0.408 and 0.3832, which is smaller than the quadratic model. And there are no significant cubic terms. So the cubic model is not what we want, and there shouldn't be any cubic terms in our formula.

Result

Our final model is the following formula

$$Y = \beta_1 E_3 + \beta_2 E_2 * \beta_6 + \beta_3 E_4 * \mathrm{E}_1$$

And the value of the coefficients in the formula shows in the following table.

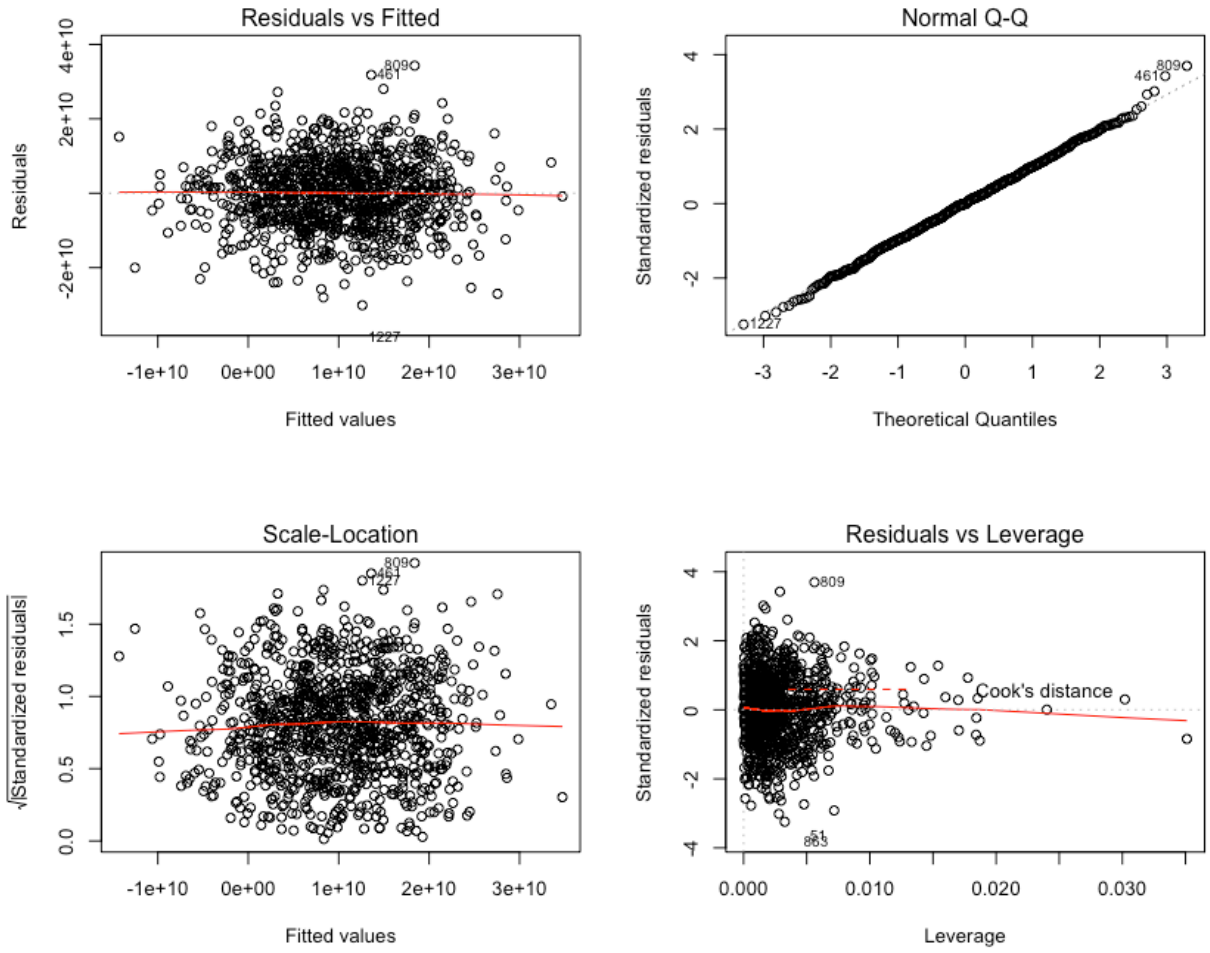
Variable	E3	E2:E6	E4:E1
Estimate	11767958	3196.893	2819.818
P-value	1.82E-157	1.07E-11	7.31E-09

The R^2 and Adjust R^2 of this model are 0.6352 and 0.6342. The F-statistic is 591.5 on 3 and 1019 degree of freedom, and the p-value for F-test is smaller than 2.2e-16.

And we did a Shapiro test which is used to test the normality of the residuals.

shapiro.test(final_quad_model$residuals)

The Test Statistics W is 0.99895, and p-value is 0.8339. So we can accept H0 which means the residuals follows normal distribution. And here's some plots about our model which prove that our model works well and the residual also looks fine.



Conclusion and Discussion

Though our result model, we can conclude that this dependent variable Y is only related to environment variables. It is independent from the gene variables when given environment variables E1, E2, E3, E4. This can be helpful when we want to predict or explain the dependent variable Y.

When we did multi-linear regression, actually, there is not a specific rule which can help you decide which model is better. During the process, we need to check all kinds of values and determine the next move. Maybe in the future, we can do more research about how to evaluate different models and the compare different methods for model selection.

Reference

1. Caspi, A. , et al., Influence of Life Stress on Depression: Moderation by a Polymorphism in the 5-HTT Gene, Science 301, 386 (2003); DOI: 10.1126/science.1083968
2. Risch, N., Herrell, R. & Lehner, T. et al., Interaction Between the Serotonin Transporter Gene (5-HTTLPR), Stressful Life Events, and Risk ofDepression: A Meta-analysis

Appendix: R codes

```
Y_table <- read.csv("~/Desktop/homework/578/IDYgroup7.csv")
E_table <- read.csv("~/Desktop/homework/578/IDEgroup7.csv")
G_table <- read.csv("~/Desktop/homework/578/IDGgroup7.csv")

Y_table <-  Y_table[order(Y_table$ID), ]
E_table <-  E_table[order(E_table$ID), ]
G_table <-  G_table[order(G_table$ID), ]

whole_data <- cbind(Y = Y_table[,c(-1, -2)], E_table[ ,c(-1,-2)], G_table[ ,c(-1,-2)])
rownames(whole_data) <- as.character(Y_table$ID)
mydata <- whole_data[complete.cases(whole_data), ]
nrow(mydata)/nrow(whole_data)

cor_matrix = cor(mydata)
cor_matrix[1,]
cor_matrix[1,abs(cor_matrix[1,]) > 0.05]
for(i in 2:32){
  cor_vec = cor_matrix[i, -1]
  print(cor_vec[ abs(cor_vec) > 0.1] )
}
attach(mydata)

fit.linear.full <- lm(formula = Y ~ ., data = mydata)
summary(fit.linear.full)
fit.quad.full <- lm(formula = Y ~ .^2, data = mydata)
summary(fit.quad.full)
null <- lm(Y ~ 1, data = mydata)
step(null, scope = list(upper = fit.quan.full), data = mydata, direction = "both")

fit1 <- lm(formula = Y ~ E3 + E4 + E2 + E6 + E1 + G20 + E2:E6 + E4:E1 + E3:E2, data = mydata)
summary(fit1)
fit2 <- lm(formula = Y ~ E3 + E4 + E2 + E6 + E1 + G20 + E2:E6 + E4:E1 + E3:E2 - 1 , data = mydata)
summary(fit2)
fit3 <- lm(formula = Y ~ E3 + E4 + E2 + E6  + G20 + E2:E6 + E4:E1 + E3:E2 - 1, data = mydata)
summary(fit3)
fit4 <- lm(formula = Y ~ E3 + E4 + E6  + G20 + E2:E6 + E4:E1 + E3:E2 - 1, data = mydata)
summary(fit4)
fit5 <- lm(formula = Y ~ E3 + E4 + G20 + E2:E6 + E4:E1 + E3:E2 - 1, data = mydata)
summary(fit5)
fit6 <- lm(formula = Y ~ E3 + E4 + G20 + E2:E6 + E4:E1 - 1, data = mydata)
summary(fit6)
fit7 <- lm(formula = Y ~ E3 + E4 + E2:E6 + E4:E1 - 1, data = mydata)
summary(fit7)
fit8 <- lm(formula = Y ~ E3  + E2:E6 + E4:E1 - 1, data = mydata)
summary(fit8)

final_quad_model <- lm(formula = Y ~ E3  + E2:E6 + E4:E1 - 1, data = mydata)
summary(final_quad_model)
par(mfrow = c(2,2))
plot(final_quad_model)
# choose E1, E3, E4, E2, E6, G20
data1 <- as.data.frame(cbind(Y, E1, E2, E3, E4, E6, G20))
fit.cubic.full <- lm(Y ~ (.)^3, data = data1)
summary(fit.cubic.full)
```