|  |  |
| --- | --- |
| **In this assignment we’ll work some more on regression modeling.**  **Learning objectives:**   1. **Confidence intervals** 2. **Multiple linear regression** 3. **Curvilinear regression, categorical regression, interactions** | |
| The mean/average is an example of a point estimator of a population parameter. As a point estimator, it is an estimate of a single value. Confidence intervals are range estimators. Rather than estimating a single value it estimates **a range that is likely to include the true unknown population parameter with some level of certainty** |  |
| We have to choose a confidence level; our level of confidence that the interval includes the population parameter. The table to the right shows how our chosen confidence level corresponds to the resulting interval. The greater the level of confidence (e.g., 99.90) the greater the confidence interval estimator. The smaller the level of confidence, the smaller the confidence interval |  |
| RCopy and run the code to the right. We create a vector, calculate the mean, and then calculate the lower and upper components to the confidence interval. | **v <- c(1,4,3,2,6,5,4,5)**  **x\_bar <- mean(v)**  **lower <- x\_bar - 1.96\*(sqrt(var(v))/sqrt(length(v)))**  **upper <- x\_bar + 1.96\*(sqrt(var(v))/sqrt(length(v)))** |
| The most precise way to understand this is that if we randomly sampled from the same population an infinite number of times, we would expect that 95% of the intervals calculated from that sample would include the true population parameter. Here is how you word this correctly in practice:   * There is an 95% chance that the confidence interval contains the true (unknown) population mean (good) * There is an 95% chance that the true (unknown) population mean lies within the interval (bad)   The reason why the second is wrong is that the population mean has no ‘chance’--it IS some value with certainty, we just don’t know it. However, we are uncertain about the interval, so the 95% chance concerns the interval, not the population parameter. | |
| Note that we choose the confidence level, but there is a trade off. The larger the confidence level, the more confident we are that the interval contains the population parameter, but, the larger the interval (100% confidence level is infinity). The convention is to use a 95% confidence interval. | |
| Now for more regression. We’re going to first create some synthetic data using a for loop. The value of ‘i’ is an index. It keeps track of the number of iterations in the loop, and can be used as a value in the loop as well. Note here that every iteration of the loop is setting the value of an element in the matrix. The function rnorm(1) generates a random number sampled from a normal distribution. Read through the code and try to figure out what the *expected* values of the variables are |  |
| Now we’ll turn the matrix into a data frame, and label the columns |  |
| Now we’re going to create two matrices (X and Y) from the dataframe |  |
| Above you can see how the regression equation is represented in matrix form. Note that the matrix of independent variables includes a column of ‘1s’. This is the intercept. | |
| Using the above, we can estimate the coefficients (β0, β1, β2) directly using the solve() function, which can solve algebraic equations in R. See: <https://www.statmethods.net/advstats/matrix.html> for more information about how to write these kinds of equations in R. It may be useful to be familiar with this, since sometimes you will see equations written in matrix form in publications that you want to solve in R (rather than by hand). | 详细的公式推导见GPT |
| Take a look again at how we generated the data above. In this case, since the data are synthetic, we know what the ***true*** coefficients are.  **Q1. Looking at this code, figure out the true expected coefficients of β0, β1, β2. You may need to read up on the rnorm() function. Explain your reasoning.**  **分别是0, 2, -1.5** |  |
| Now let’s generate some predictions from the model. The vector b we created earlier contains estimated coefficients. The intercept coefficient estimate (b0) is the first element, the x1 coefficient (b1) is the second element and the x2 coefficient (b2) is the third element. |  |
| Write and run the code to the right to get predictions. The vector **p** co ntains the predictions from the model. |  |
| **Q2. Explain why the predictions in p are not the same as the values of Y (in less than two sentences). Hint: look at the equations we used in the data synthesis step.** | |
| We can also manually calculate model error |  |
| As you already know, there is an easier way to use R to perform regression analysis. | |
| The code to the right widll create a regression data object, summarize the regression output and create predictions and residual error |  |
| We can also plot the relationship between the predicted and actual vlcalues of Y (the dependent variable). |  |
|  | |
| Write and run the code: | |
| The above creates synthetic data but includes a variable that has a **curvilinear** relationship between one of the independent variables and the dependent variable. Run the code to the right. And look at the output below: |  |
|  | |
| **Q3. Provide a brief description of the above results from the regression analysis (no more than two sentences). *Note: Your results will not be exactly the same as above, but similar.*** | |
| Now we are going to plot the predictions against the dependent variable p. |  |
|  | |
| Looking at the above figure, there is a bit of an irregularity in the scatter of points. Note that predictions between 0 and 4 correspond with values of Y that are between 0 and 5. In a properly specified model, we should expect the scatter to be more even above and below the line on the graph. You can also see what appears to be a flaring (ei火焰) at high and low predictions such that low predictions and high predictions both correspond with larger range of values of Y. | |
| **Q4. Plot X1 variable by Y and the X2 variable by Y and comment on the plots produced (in less than two sentences)** | |
| What this illustrates is the value of plotting out the data before modeling to get a sense of the possible form of relationships between variables. This way we can run a different model that should account for the relationship better. |  |
| **Q5. Generate code that creates a new vector of predictions/fitted values and plots them on a graph along with the observed values of Y. Provide an interpretation of this graph.** | |
| Here we create another synthetic data set. The X2 variable has three values (0, 1 or 2). |  |
| We will convert X2 into a factor variable. |  |
| Write and run the code to the right |  |
| **Q6. Provide an interpretation of the results of the model. Pay careful attention to how you interpret the categorical variable.** | |
| In our final example, we are going to look at simple interactions between a categorical variable and a continuous variable. Write and run the code below: | |
| Run the model |  |
| Turn the X3 variable into a factor |  |
| An **interaction** between variables occurs when the association between one independent variable and dependent variable varies based on the values of another independent variable. For example, if you take medication that is supposed to control blood pressure (BP is the dependent variable, medicine the independent) and then eat grapefruit (another independent variable) the association between medicine and blood pressure can change.  Interactions between variables should generally be identified based on our understanding rather than statistical diagnostics. In other words, you should already think, based on your knowledge of the data and the field, that an interaction will exist. However, it is possible to explore interactions graphically before modeling.  Here we will create some three-way boxplots. There is one categorical independent variable and two continuous independent variables. If an interaction exists it will because two (or more) variables combine (through multiplication) to have an association with the dependent variables independent of their additive association. | |
|  | |
|  |  |
| Note that the relationship between X3 and Y varies depending on the value of X2; when X2 is above the mean, “no” is associated with higher values of y than “yes”. When X2 is below the mean, “yes” is associated with higher values of y than “no”. | |
| Write and run the code to the right |  |
| Write and run the code to the right |  |
| **Q7. Find a way to compare these two model results in a way that illustrates the role of the interaction in predicting values of Y. Graphs/plots are preferred over tables and numbers.** | |
| **Go to the link:**  [**https://docs.google.com/spreadsheets/d/1JzllHwMxJGc4p54oafRS7naMDXBa1BzJGYABhcGctms/edit?usp=share\_link**](https://docs.google.com/spreadsheets/d/1JzllHwMxJGc4p54oafRS7naMDXBa1BzJGYABhcGctms/edit?usp=share_link)  And download the file to your computer. The file has gold prices by year.  **Q8. Go online to find another variable (measured at the year-level) and merge it to these data. Next, use regression to model the price of gold as a function of the new variable you have added AND year (in one model). Explore the data for interactions, non-linear associations, etc. Provide an interpretation of the results that is meaningful. *Note: it is not a problem if the variable you find is not associated with the price of gold.***  **Q9. Once you have the model completed, come up with a way to validate the model you created with this dataset:**  <https://docs.google.com/spreadsheets/d/1IEqA9iplxDMGjmvMBMElh3dD2IXLOdx5FnsFoLluwCo/edit?usp=share_link>  **Q10. Is the model you created any good? Why/why not?** | |
| Linear regression makes certain assumptions about data, and is surprisingly effective even when some of those assumptions are violated. For example, it assumes that the dependent variable is approximately normal and measured on a continuous scale. Even when this is not true, linear regression can often be used to make useful predictions.  Nevertheless, there are times when linear regression is not well suited to the data we have. One example is when our dependent variable is a dichotomous categorical variable—like true or false, yes or no, or 1 or 0.  Consider the situation in which a researcher wants to predict whether or not a person will walk to work given information about their income, age and level of education. Copy and paste the code below:  age <- sample(c(20,30,40,50,60,70,80,90,100),1000,replace=TRUE)  income\_1000s <- sample(c(20,30,40,50,60,70,80,90,100),1000,replace=TRUE)  education <- sample(c(1,2,3,4,5),1000,replace=TRUE)  linear\_form <- 22 + age\*-0.2 + income\_1000s\*-0.1 + education\*.01  walk <- rbinom(1000,size=1,prob=(1/(1 + exp(-linear\_form)))) | |
| The code above creates 3 independent variables and one dependent. I have made all the independent variables numeric for simplicity, though some/all could have been categorical  The variable linear\_form describes the linear association between these three independent variables and some outcome, but the outcome is not dichotomous (1 or 0). We use a function to ‘squish’ these values into a sigmoid function so that they are between 1 and 0, and then use these as probabilities to randomly sample 1s and 0s from a binomial distribution.  The result is a dataset in the form we are looking for—one dichotomous dependent variable, and 3 independent variables. | |
| **Q11. Use the lm() function to model walk as a function of age, income\_1000s + education. Take a look at the fitted values (of the dependent variable), and comment on whether or not they make sense and why/why not.** | |
| Logistic regression is a type of generalized linear model (GLM) used for predicting a binary outcome based on one or more predictors. The logistic function ‘maps’ any input into a value between 0 and 1, which can be interpreted as the probability of a particular class.  To fit the model, we use:  fit <- glm(walk ~ age + income\_1000s + education, family = "binomial")  In this example, the glm() function takes two main arguments:   * The formula for the model: walk ~ age + income\_1000s + education * family: This tells R to use logistic regression (family = "binomial"). | |
| Run the code to the right to see a summary of model results. Note that the coefficients in the summary look similar to the values that we included in the linear\_form equation above. |  |
| We can also take a look at the model fitted values.  **Q12. Think about Q11, and describe why these fitted values make more sense.** |  |
| Unfortunately, logistic regression models do not have a convenient and simple measure of overall model fit (like an adjusted R squared). Instead, we can consider comparing the null model (with no predictors) to the fitted model. This can be done qualitatively by comparing the Null deviance and the Residual deviance:    The residual deviance has a smaller ‘deviance’, which is generally better. Note that the difference in degrees of freedom is due to the number of parameters in the model. | |
| There are a number of statistical tests to more formally assess model fit, but best practice is to do some sort of cross-validation. In this case, you would hold back some data, fit the model, and see how good it is at predicting the values in the held-back data.  **Q13. Do the following:**   1. **Randomly separate the data (all four variables) into a training data set (of 700 observations) and testing data set (300 observations)** 2. **Use logistic regression to model the training data** 3. **Find a way to predict the dependent variable of the test data set using the model estimated using the training data** 4. **Compare the observed and predicted values, and make a judgment about the quality of the model. Justify your conclusion about the quality of the model**   **Submit to avenue before next class.** | |