QMB 6358: Software Tools for Business Analytics
College of Business
University of Central Florida
Fall 2020

Assignment 9

Due Thursday, December 10, 2020 at 9:59 PM
in your GitHub repo.

Instructions:

Complete this assignment within the space on your GitHub repo in a folder called `assignment_09`. You may organize your files any way you like but leave your answers to all questions in this folder.

All of your responses can be completed using the language of your choice, as long as your solutions meet the specifications in each question. Store any printed output by writing or pasting into a document of your choice or pasting comments in your code. This output can also be automated by redirecting ouput from a script in Question 6.

When you are finished, submit your code and any other documents by pushing your changes to your GitHub repo, following the instructions in Question 7. Complete these exercises individualy and `git push` your own work.

Part A: Data Handling and Regression Modelling

Estimate the best regression model you can by solving as many of Questions 1 to 4 as you can. You do not necessarily have to solve them in order.

Question 1:

The folder `assignment_09` contains three `.csv` files: `airplane_sales.csv`, `airplane_specs.csv`, and `airplane_perf.csv`. The first dataset `airplane_sales.csv` contains the following variables.

`SALE_ID`	=	a unique key for each airplane sold
`price`	=	price of an airplane
`age`	=	age of the aircraft, in years

Use this dataset to estimate a regression model to predict the prices of airplanes.

a) Read in the `airplane_sales.csv` dataset and store it in a data frame called `airplane_sales` in your workspace.

b) Calculate and store the printed output from either a `summary` of the data or `describe` the data, according to your choice of software. Use this to get familiar with the contents of the dataset.

c) Estimate a regression model to predict `price` as a function of `age`. Store the printed estimation output with the `print` and/or `summary` command, as appropriate.

Question 2:

Now use two files `airplane_sales.csv` and `airplane_specs.csv` in the folder `assignment_09`. The dataset `airplane_specs.csv` contains the following variables.

`SALE_ID` = a unique key for each airplane sold
`pass` = the number of passengers an airplane can accommodate
`wtop` = an indicator that the wings are above the fuselage
`fixgear` = an indicator for fixed landing gear (i.e. wheels are not retractable)
`tdrag` = an indicator that a wheel is on the tail (a tail-dragger)

Use the variables from both datasets to estimate a better regression model to predict the prices of airplanes.

a) Perform any pre-processing that needs to be done to the files `airplane_sales.csv` and `airplane_specs.csv` before joining them: clean them, `sort` them or `read` them, according to your strategy of choice.

b) Form a dataset `airplane_sales_specs.csv` by `paste`ing, `join`ing, or `merge`ing the datasets, as needed.

c) If not already done in the above, `read` the new dataset and store it in a data frame called `airplane_sales_specs` in your workspace.

d) Calculate and store the printed output from either a `summary` of the data or `describe` the data, according to your choice of software. Use this to get familiar with the contents of the dataset.

e) Estimate a regression model to predict `price` as a function of `age`, `pass`, `wtop`, `fixgear`, and `tdrag`. Store the printed estimation output with the `print` and/or `summary` command, as appropriate.

Question 3:

Now use all three files `airplane_sales.csv`, `airplane_specs.csv`, and `airplane_perf.csv` in the folder `assignment_09`. The dataset `airplane_perf.csv` contains the following variables.

`SALE_ID`	=	a unique key for each airplane sold
`horse`	=	the horsepower of the engine
`fuel`	=	the volume of the fuel tank, in gallons
`ceiling`	=	the maximum flying height of an airplane, in feet
`cruise`	=	the cruising speed, in MPH

Use the variables from these datasets to estimate an even better regression model to predict the prices of airplanes.

a) Perform any pre-processing that needs to be done to the file `airplane_perf.csv` before joining it to the others: clean, `sort` or `read`, according to your strategy of choice.

b) Form a dataset `airplane_full.csv` by `paste`ing, `join`ing, or `merge`ing the datasets, as needed.

c) If not already done in the above, `read` the new dataset and store it in a data frame called `airplane_full` in your workspace.

d) Calculate and store the printed output from either a `summary` of the new variables or `describe` the new variables, according to your choice of software. Use this to get familiar with the contents of the dataset.

e) Estimate a regression model to predict `price` as a function of `age`, `pass`, `wtop`, `fixgear`, and `tdrag`, as well as `horse`, `fuel`, `ceiling`, and `cruise`. Store the printed estimation output with the `print` and/or `summary` command, as appropriate.

Question 4:

Now calculate new variables to estimate a model for airplane prices using a different functional form. Use the variables from your best model from Questions 1 to 3.

a) Create new variables `log_price`, `log_age`, `log_horse`, `log_fuel`, `log_ceiling`, and `log_cruise` from the variables `price`, `age`, `horse`, `fuel`, `ceiling`, and `cruise`, using the logarithm function `log()` in `R` or `math.log()` in `Python`.

b) Calculate and store the printed output from either a `summary` of the new variables or `describe` the new variables, according to your choice of software. Use this to get familiar with the contents of the dataset.

c) Estimate a regression model to predict `log_price` as a function of `log_age`, `pass`, `wtop`, `fixgear`, and `tdrag`, as well as `log_horse`, `log_fuel`, `log_ceiling`, and `log_cruise`. Store the printed estimation output with the `print` and/or `summary` command, as appropriate.

Part B: Function Design and Optimization

Question 5:

Estimate $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_k)'$ by minimizing the sum of squared residuals, defined as

$$SSR(\beta; y, x_1, \ldots, x_k) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_k x_{ki})^2$$

a) Define a function `SSR(beta; ...)` that calculates the sum of squared residuals. Your function should be compatible with the best model from Part A. In particular, it should allow for all k explanatory variables that are used in your model.

b) Test your function by comparing the value to the `SSR` obtained from your best model from Part A. Take the value of `beta` from the estimated coefficients to calculate `SSR(beta; ...)`. Compare this value with `sum(my_lm_model$residuals^2)` in `R` or `sum(reg_model_sm.resid**2)` using the `stats.models` module in `Python`, for example.

c) Use a numerical optimization function to minimize your `SSR(beta; ...)` function.

d) Verify the accuracy of your calculation by printing your optimal parameter values and comparing them with the values in your estimated model from Part A. Validate the optimized value of the `SSR(beta; ...)` function against the values from part (b).

Part C: Software Management and Version Control

Question 6:

Create a UNIX shell script called `assignment_09.sh` that runs all the software to answer Questions 1 to 5 in Parts A and B.

a) Use commands such as `Rscript`, `python3`, or `sqlite3` to run your software.

b) Redirect the output of each script to appropriately-named `.txt` or `.out` files, using the “`>`” operator, to save your output.

c) You can test your script by running `./assignment_09.sh`.

Question 7:

Push your completed files to your GitHub repository following these steps. See the `README.md` and the `GitHub_Quick_Reference.md` in the folder `demo_04_version_control` in the QMB6358F20 course repository for more instructions.

1. Open GitBash and navigate to the folder inside your local copy of your git repo containing your assignments. Any easy way to do this is to right-click and open GitBash within the folder in Explorer. A better way is to navigate with UNIX commands.
2. Enter `git add .` to stage all of your files to commit to your repo. You can enter `git add my_filename.ext` to add files one at a time, such as `my_filename.ext.` in this example.
3. Enter `git commit -m "Describe your changes here"`, with an appropriate description, to commit the changes. This packages all the `add`ed changes into a single unit and stages them to `push` to your online repo.
4. Enter `git push origin master` to push the changes to the online repository. After this step, the changes should be visible on a browser, after refreshing the page.