EBA35001 Fall 2022

Take home exam

Jonas Moss

1. You need to be selective about the output you show. Only show output that supports your argument! If you use Jupyter Notebooks, you may hide the output of a cell using a semi-colon `;`. We will deduct points from shoddily written reports plagued by noisy outputs.
2. Make your plots look nice. Add appropriate axis labels, legends and so on.
3. *"Brevity is the soul of wit."* Strive not to write too much. We prefer pithy to lengthy expositions.
4. The exercises are equally weighted. Each exercise gives $0-2$ points and there are 30 of them. That's a maximum of 60 points.

1 Binary regression

We will use the Rain in Australia data set, called `weatherAUS.csv`, in this exercise. The data set is available on the Github page for this course. Our goal is to predict if there will be rain tomorrow, stored as `RainTomorrow` in the data set.

(a) Exploration

(i)

1. Import the data set as `weather`. The response `RainTomorrow` must be modified before we can fit the logistic model. Why? Transform `RainTomorrow` and `RainToday` to something more suitable for binary regression analysis.
2. There are many `NA` values in the data set, but we cannot use these. Remove every row containing `NA`s from the data set.

(ii)

1. How many unique values are there in the `Date` column?
2. How many unique values are there in the `Location` column?
3. It's not possible to fit `RainTomorrow ~ Date * Location`. Why?
4. We won't be using `Date` anymore, so remove `Date` from the weather data.

(iii)

What are the unique locations in the data set? Display each unique location along with how many times it appears in the data set.

Some of the covariates encode wind directions. Justify your answers to the questions below!

1. List all the possible wind directions.
2. What are the wind direction covariaties? Do all of them have the same set of possible wind directions?
3. How many parameters are fitted in the model `"RainTomorrow ~ WindGustDir"`?

(b) Locations

(i)

Fit a logistic model with `Location` as its only covariate.

1. What is the probability of rain tomorrow in `AliceSprings`?
2. Which location has the smallest probability of rain? And what is the probability?
3. Which location has the highest probability of rain? And what is the probability?

(ii)

1. Make a table that displays the probability of rain tomorrow for each location along with a confidence interval. Be sure to include `AliceSprings`!

2. Fit a model you can use to predict the probability of rain tomorrow from the probability of rain today, ignoring location. What's the McFadden R^2 of this model? (Display the R^2 without using the summary method.)

(iii)

Fit a model using both `location` and `rain`. Then fit a model with interactions between location and `RainToday`. Interpret and compare the parameters of the models. Which model do you prefer?

(c) Fitting

(i)

Make a function `all_column` that takes a data frame `data` and a response name `name` and outputs a formula including all column names in `data` on the right-hand side. For instance, if the data frame `data` contains the columns "Donald", "Huey", "Dewey", and "Louie", `all_column(data, "Donald")` should output `"Donald ~ Huey + Dewey + Louie"`. Run `all_columns(weather, "RainTomorrow")` do demonstrate that it works.

(ii)

Fit a logistic model for `RainTomorrow` with all covariates. How many parameters was fitted in this model? (Don't use `summary` for this! Use an argument or a method from `statsmodels`.)

(iii)

Fit at least five logistic regression models on this data set and choose your favorite. Make sure to justify your choice.

(iv)

Investigate the effect of link functions on your choice of models. Using the same models as in your previous exercise, change the link function from the logistic link to the Probit link, Cauchit link, and cloglog link. Report the BICs of the models in a table like this:

	Logistic	Probit	Cauchit	Cloglog
Model 1				
Model K				

(Hint: You need to take a good look at the documentation of the `glm` *function of statsmodels. Also see the lecture notes.)*

(v)

1 Make a ROC curve for the logistic model `RainTomorrow ~ Location * RainToday` and interpret it.

2. What is the AUC of this model?

2 Linear regression

We'll use the `CPSSW8` data set in this exercise.

(a) Exploration

(i)

Fit the model `np.log(earnings) ~ age + education + gender`.

1. Show the output of the model and interpret each coefficient.
2. Why is there no coefficient for `female`?

(ii)

Fit a model with an interaction term between gender and education. Interpret the coefficient `gender[T.male]:education`. What is the p-value for this coefficient?

(iii)

Using the model in the previous exercise, predict the wage (not on the log scale!) for:

1. A female with `education = 20` and age = `40`
2. A male with `education = 20` and age = `28`.

(b) Regions

(i)

What are the unique "regions" in this data set? Make a table containing 95% confidence intervals for the average earnings for each combination of region and gender.

	Region 1	...	Region K
Male	(l, u)		(l, u)
Female	(l, u)		(l, u)

Here (l, u) are the lower and upper limits of a confidence interval. Be sure to give "Region 1" et cetera appropriate names!

(ii)

Add the "region" term to the model `np.log(wage) ~ age + education + gender`.

1. Which region is the reference class in this model?
2. Is the `region` covariate significant?
3. Which region has the worst effect on wages?

(iii)

Add the interaction term `region * education` to the previous model.

1. Interpret the coefficients of this model.
2. Explain what `region[T.South]:education` means using one sentence.
3. Test if `region * education` is significant.

(iv)

Still using the previous model.

1. Predict the wages of a female of age 40 living the in the south with 0 education. Will your predicted wages increase or decrease if you change the region to Northeast?
2. Do you think it is best to use the model with the interaction term `region * education` or the model without the interaction term? Justify your choice.
3. Suppose you learn that Robin is `35` and resides in the south. Can you use the model in (iii) to predict his / her wage? Why or why not?

(c) Model fit and model choice

(i)

We're using `"np.log(wages) ~ age + education * gender"` in this subexercise.

1. Plot the distribution of the residuals using a QQ plot and a histogram. Are the residuals normal? If not, do they deviate from normality in a way we expect to have serious implications for inference?
2. Make a residual plot, residuals versus fitted. Make some comment on its looks!

(ii)

You need to make a good predictive model for the log wages. Try out at least five models and make an informed choice between them. Be sure to justify your choice, and be sure to try out at least one transformation of the covariates!

(iii)

Report the R^2s of the models from (ii) in a table.

3 Simulations

We will take a look at the Jarque-Bera test. This tests if a univariate data set matches the normal distribution. More specifically, it tests if the sample skewness and kurtosis match the normal distribution. The population values of the skewness and kurtosis are defined by

$$\text{Skewness} = \frac{E(X-\mu)^3}{\text{Var}(X)^{2/3}}$$

and

$$\text{Kurtosis} = \frac{E(X-\mu)^4}{\text{Var}(X)^2},$$

where $\mu = EX$. Roughly speaking, the skewness measures how skewed a distribution is, while the kurtosis measures its “tailedness”.

(a) Implementing the function.

The sample skewness and kurtosis are defined as

$$S = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^{3/2}},$$

$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^{2}}.$$

Here $\overline{x}$ denotes the sample mean of a vector.

(i)

Implement the sample skewness as the function `skewness` taking `x` as an argument. Run the function on the vector `np.array([7,0.3,1,33,0,6])`.

(ii)

Now implement the sample kurtosis as the function `kurtosis` taking a Numpy array `x` as an argument. Run the function on the vector `np.array([2,1,0.5,1,2,5])`. (Please implement your own function, even if Numpy or Scipy might have it implemented already.)

(iii)

Make the `jarque_beta` function, taking `x` as an argument, that implements the Jarque-Bera test. To do this, use that the definition of the Jarque-Bera test is $\frac{n}{6}(S^2 + \frac{1}{4}(K-3)^2)$, where n is the sample size S is the sample skewness, and K is the sample kurtosis.

(b) Simulations

(i)

Make a function that simulates `n` observations from a normal distribution with mean `mu` and standard deviation `sigma` then calculates the Jarque-Beta test on these values. Do this `n_reps` times, and return the resulting test values as a Numpy vector. (Use the signature `jarque_bera_normal(n, mu, sigma, n_reps)`).

(ii)

Using `n = 100` and `n_reps = 10**5`, call `jarque_bera_normal` with your choice of `mu` and `sigma`. Make a histogram of the values. Moreover, according to Jarque-Bera test, the distribution of the Jarque-Bera test should be approximately χ^2-distributed with 2 degrees of freedom. To verify this, add a line plot of the χ^2-distributed with 2 degrees of freedom to the histogram. Comment how well the lines match. (***Hint:*** To plot the χ^2-distribution you must consult the Numpy documentation.)

(iii)

Consider the null hypothesis

$$H_0 : \text{The true distribution is normal.}$$

Since the Jarque-Bera test is χ^2-distributed with 2 degrees of freedom, we can calculate its p-value using `scipy.stats.chi2`. Explain *how* you would do this and *why* the result is a p-value.

(c) Power of the functions

The definition of a p-value only mentions the null-hypothesis, but in order for it to be useful it must have **power** against some reasonable alternative. The power of a test is, for a fixed significance level α and alternative hypothesis, the probability that it is able to detect that H_0 isn't true.

(i)

Make a function `simulate_jarque_bera` that takes three arguments `n`, `n_reps` and `random` as arguments. The `random` argument should be a random generator taking one `size` argument. (E.g. `lambda size: rng.normal(mu, sigma, size)`, `lambda size: rng.exponential(lambda, size)`). It should simulate the Jarque-Bera test as we did in (b), but with the supplied distribution `random` instead of the normal distribution.

(ii)

Make a function `power_jarque_bera(n, n_reps, random, alpha = 0.05)`. The first three arguments are the same as the previous exercise, and `alpha` is a significance level. It should return the approximate probability that the `Jarque-Bera` test will be significant at the `alpha` level when the true distribution is `random`.

(iii)

Use the `power_jarque_bera(n, n_reps, random, alpha = 0.05)` function to calculate the power of the Jarque-Bera test for 5 different choices of distributions (the `random` argument) and put them into a table with $n = 50, 100, 1000$. One of these five distributions should be the Laplace distribution.