FinTech 545 - Week 2

Renjie Wang

January 27, 2024

1. (a) Using normalized formula:

   The first moment (mean) is calculated with:

   $$\widehat{\mu_1} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

   The second moment (variance) is calculated with:

   $$\widehat{\mu_2} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \widehat{\mu_1})^2$$

   The third moment (skewness) is calculated with:

   $$\widehat{\mu_3} = \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - \widehat{\mu_1})^3}{(s^2)^{\frac{3}{2}}}$$

   where biased variance is $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu_1})^2$.

   The fourth moment (excess kurtosis) is calculated with:

   $$\widehat{\mu_4} = \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - \widehat{\mu_1})^4}{(s^2)^2} - 3$$

   where biased variance is $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu_1})^2$.

   The results are:

mean	variance	skewness	excess kurtosis
1.0490	5.4272	0.8806	23.1222

   (b) Using statistical packages: Here I am using the following packages: For the first and second moment calculation - `numpy` is used

   `numpy.mean(), numpy.var()`

   for $\widehat{\mu_1}, \widehat{\mu_2}$ respectively.

   For the third and fourth moment calculation - `scipy.stats` is used

   `scipy.stats.skew(), scipy.stats.kurtosis()`

for $\widehat{\mu_3}, \widehat{\mu_4}$ respectively.

The results are:

mean	variance	skewness	excess kurtosis
1.0490	5.4218	0.8806	23.1222

(c) To figure out whether the statistical package functions are biased, we can apply the hypothesis testing method.

We need then formulate a null hypothesis for each estimator for different moments. Using `numpy.random.normal()`, we can easily generate 100 realizations of standard normal random variables.

For the standard normal distribution, the first four moments are as follows:

$$\mu_1 = 0$$

$$\mu_2 = 1$$

$$\mu_3 = 0$$

$$\mu_4 = 0$$

where $\mu_1, \mu_2, \mu_3, \mu_4$ are mean, variance, skewness and excess kurtosis respectively.

After deriving the sample, we can use the package to calculate the sample mean, variance, skewness and excess kurtosis for this simulation. We can then repeat this process for another 999 times (1000 simulations total) and store the data for the first four moments $\widehat{\mu_1}, \widehat{\mu_2}, \widehat{\mu_3}, \widehat{\mu_4}$.

The null hypothesis is based on the true value for the first four moments of standard normal distribution. A t-test is essential for investigating whether the package functions are biased. The t-statistic of μ is given by

$$t = \frac{\widehat{\mu} - \mu}{s/\sqrt{n}}$$

where $\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n}\widehat{\mu_i}$, μ is the theoretical value and s is the standard deviation of $\widehat{\mu}$.

We can also derive the p-value for the hypothesis testing.

The results are:

	mean	variance	skewness	excess kurtosis
t-statistic	-0.3134	-2.4871	0.3354	-3.1613
p-value	0.7540	0.0130	0.7374	0.0016

Given the t-test result, under the significance level of $\alpha = 0.05$, we fail to reject the null hypothesis for the package function estimator of mean and skewness.

We reject the null hypothesis for the variance and kurtosis, which means the package functions for variance and kurtosis are biased.

2. (a) $\hat{\beta}^{OLS}$ is solved with the formula: $\hat{\beta}^{OLS} = (X'X)^{-1}(X'Y)$, and the variance s^2 is solved with the formula: $s^2 = \frac{\epsilon'\epsilon}{n-2}$.

The MLE estimators are derived by maximizing the log-likelihood function.

Using `scipy.optimize.minimize()`, the MLE estimators are acquired. (The method of "nelder-mead" is used for optimization.)

The fitted $\hat{\beta}$ and standard deviation of ϵ under OLS and MLE optimization are:

	$\hat{\beta}_0$	$\hat{\beta}_1$	s
OLS	-0.0874	0.7753	1.0088
MLE	-0.0874	0.7753	1.0038

The beta estimations are basically the same between OLS and MLE. However, the standard deviation of the OLS error to the fitted MLE σ, since the s^2 of OLS is the unbiased estimate of the variance, while the fitted MLE σ^2 is biased.

(b) Under the assumption of error following a t distribution. The fitted $\hat{\beta}$ and standard deviation of ϵ under MLE are:

	$\hat{\beta}_0$	$\hat{\beta}_1$	s
MLE	-0.0973	0.6750	0.8551

To distinguish which model is the better fit, we need to use the metric as a standard. Here, I used adjusted R^2 as an indicator. The adjusted R^2 for the normality assumption is 0.3423, while for t distribution is 0.3363. This tells us that normality assumption is a better fit.

(c) The distribution of X_2 given each observed value of $X_1 = a$ is:

$$X_2 \sim N(\bar{\mu}, \bar{\Sigma})$$

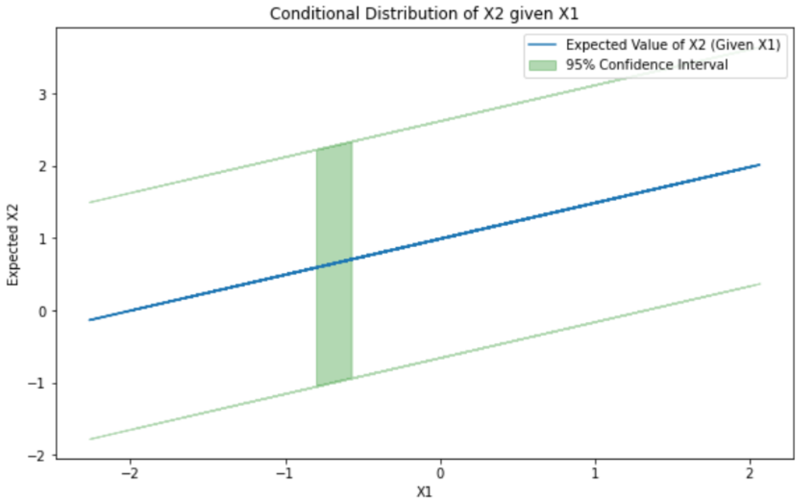
where

$\bar{\mu} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(a - \mu_1)$

$\bar{\Sigma} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{21}$

$\bar{\mu}$ is a vector of 100 elements - The first five elements are 0.592912, 1.211941, 1.582989, 1.133166, 1.056346. $\bar{\Sigma}$ is 0.698216.

The plot of the expected value along with the 95% confidence interval:



(d) From the assumption, we can derive that

$$Y - X\beta \sim N(0, \sigma^2 I_n)$$

The likelihood function is given by

$$\begin{aligned}\ell(\beta, \sigma^2) &= \prod_{i=1}^{n} f(Y|X;\beta,\sigma^2)\\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_i - X_i\beta)^2}{2\sigma^2}}\end{aligned}$$

Take the log on both sides, we can derive the log-likelihood function:

$$\ell\ell(\beta, \sigma^2) = -\frac{n}{2}\ln(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(Y_i - X_i\beta)^2}{2\sigma^2}$$

We need to maximize the log-likelihood function. Therefore, we may take the derivative of the function to derive the estimator for β and σ^2

$$\begin{aligned}\frac{\partial}{\partial\beta}\ell\ell(\beta,\sigma^2)) &= -2\sum_{i=1}^{n} \frac{(Y_i - X_i\beta)(-X_i)}{2\sigma^2}\\ \frac{\partial}{\partial\sigma^2}\ell\ell(\beta,\sigma^2)) &= -\frac{n}{2\sigma^2} + \sum_{i=1}^{n} \frac{(Y_i - X_i\beta)^2}{2(\sigma^2)^2}\end{aligned}$$

Set the derivative to be 0 respectively -

For $\hat{\beta}$,

$$\begin{aligned}-2\sum_{i=1}^{n} \frac{(Y_i - X_i\hat{\beta})(-X_i)}{2\sigma^2} &= 0\\ \frac{1}{\sigma^2}(X'Y - X'X\hat{\beta}) &= 0\\ X'X\hat{\beta} &= X'Y\\ \hat{\beta} &= (X'X)^{-1}(X'Y)\end{aligned}$$

The equation is valid when $X'X$ is non-singular.

For $\widehat{\sigma^2}$,

$$
\begin{aligned}
-\frac{n}{2\widehat{\sigma^2}} + \sum_{i=1}^{n} \frac{(Y_i - X_i\beta)^2}{2(\widehat{\sigma^2})^2} &= 0 \\
\frac{n}{2\widehat{\sigma^2}} &= \sum_{i=1}^{n} \frac{(Y_i - X_i\beta)^2}{2(\widehat{\sigma^2})^2} \\
\widehat{\sigma^2} &= \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i\hat{\beta})^2
\end{aligned}
$$

3. We apply Python's `SARIMAX()` function to fit AR(1) through AR(3), and MA(1) through MA(3) models to the data. The result is:

	Log Likelihood	AIC	BIC
AR(1)	-919.096	1842.192	1850.621
AR(2)	-917.017	1840.033	1852.677
AR(3)	-745.228	1498.456	1515.314
MA(1)	-968.996	1941.992	1950.421
MA(2)	-928.777	1863.553	1876.197
MA(3)	-928.700	1865.401	1882.259

The best of fit is the **AR(3)** model. A higher likelihood and a lower AIC/BIC are all indicators of a better fit.