FinTech 545 Homework 1

Haonan Xu

January 26, 2024

1. (a) The distribution moment for a random variable X is:

$$\mu_n = E[(x-c)^n]$$

where E is the expectation operator. For the first moment, we set $c = 0$ to get the mean, denoted $\mu = E[x]$. For higher moments, we set $c = \mu$.

The 2nd moment is variance, $\sigma^2 = E[(x-\mu)^2]$

The 3rd moment is skewness, $\widehat{\mu_3} = E[(\frac{x-\mu}{\sigma})^3]$

The 4th moment is kurtosis, $\widehat{\mu_4} = E[(\frac{x-\mu}{\sigma})^4]$. The excess kurtosis is kurtosis subtracted by 3. We use the population variance, not the sample variance. For the 3rd and 4th moment, we use standardized values, hence divided by σ. The results are:

mean	variance	skewness	excess kurtosis
1.0490	5.4218	0.8806	23.1222

(b) Using statistical packages

`numpy.mean(), numpy.var(), scipy.stats.skew(), scipy.stats.kurtosis()`

to generate the estimates

mean	variance	skewness	excess kurtosis
1.0490	5.4218	0.8806	23.1222

Compared to answers in part (a), the estimates are nearly identical.

(c) To test whether the statistical package functions are biased, we can simulate random samples with known estimators. For each simulation, we want small sample size (n = 100), because smaller n generates greater difference between the bias corrected estimator and the biased estimator. When n gets larger, unbiased estimator will converge to biased estimator.

We choose standard normal distribution because we know the true value of distribution moment estimators: $\mu = 0$, $\sigma^2 = 1$, skewness $= 0$, excess kurtosis $= 0$. For each estimator, our null hypothesis is that the difference between estimators given by statistical package ($\widehat{\mu_n}$) function and the true estimators (μ_n) is 0, while the alternative hypothesis suggests that the estimate by package function is not equal to the true estimate.

$$H_0 : \widehat{\mu_n} - \mu_n = 0,\ H_1 : \widehat{\mu_n} - \mu_n \neq 0$$

We simulated the random sample 10000 times, each time calculating the package estimators, and use t-test to test against the true estimator. For example, The test statistics of $\widehat{\mu}$ is given by

$$t = \frac{\bar{\widehat{\mu}} - \mu_0}{s/\sqrt{n}}$$

where $\bar{\widehat{\mu}}_i = \frac{1}{n}\sum_{i=1}^{n}\widehat{\mu}_i$ is the sample mean of $\widehat{\mu}$. The theoretical value is $\mu_0 = 0$, and s is the variance $\widehat{\mu}$.

		mean	variance	skewness	kurtosis
0	unbiased estimate	0.000000	1.000000	0.000000	0.000000
1	t stat	0.349512	-2.948428	-0.166497	-4.682037
2	p-value	0.726779	0.003268	0.867799	0.000003

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 μ and skewness. `numpy.mean()` and `stats.skew()` yields unbiased expected value.

We reject the null hypothesis for the other estimators, σ^2, and kurtosis. The estimators given by package functions are not equal to the true estimators. The package function for these two distribution moments are biased.

2. (a) The fitted $\hat{\beta}$ and standard deviation of ϵ under OLS and MLE optimization are:

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

The estimates under OLS and MLE are very close to each other.

(b) We assume the errors follow T distribution. The fitted $\hat{\beta}$ and standard deviation of ϵ under MLE are:

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

We calculate the adjusted R^2 for MLE optimization under both normality and T assumption. The adj. R^2 under normal assumption is 0.3423, which is slightly higher than T assumption's 0.3363. Although both R^2 are not large, we can still say that normal distribution assumption for ϵ fits better.

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

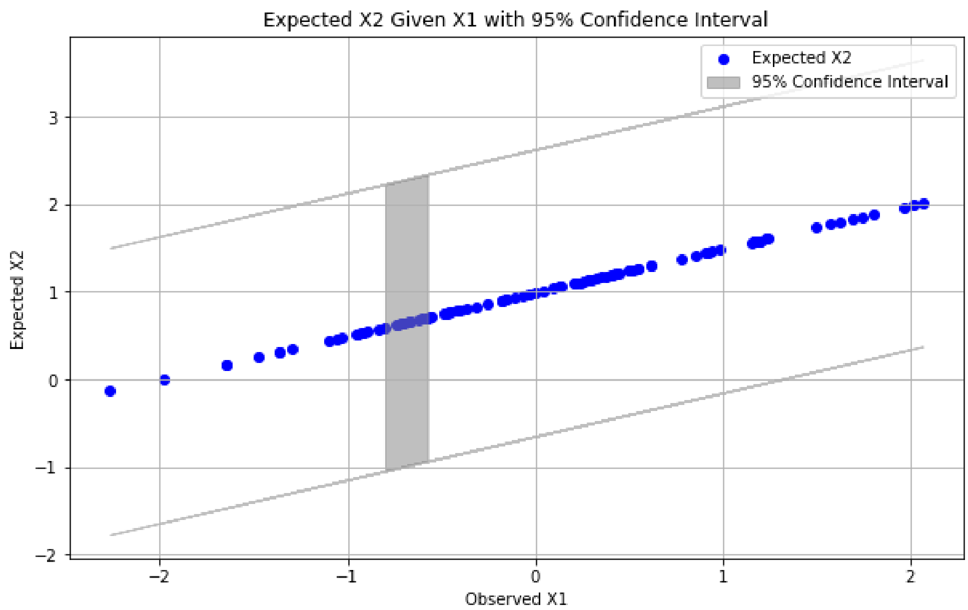
$$X_2 \sim N(\bar{\mu}, \bar{\Sigma})$$

where

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

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

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



(d) Given $\epsilon \sim N(0, \sigma^2 I_n)$, and $Y = X\beta + \epsilon$, we can derive $Y - X\beta \sim N(0, \sigma^2 I_n)$. The likelihood function for the parameters is:

$$\begin{aligned}
\ell(\beta, \sigma^2) &= \prod_{i=1}^{n} f(Y|X; \beta, \sigma^2) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{\sigma^2 2\pi}} e^{-\frac{1}{2}(\frac{\epsilon_i - 0}{\sigma})^2} \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{\sigma^2 2\pi}} e^{-\frac{1}{2}(\frac{Y_i - X_i\beta}{\sigma})^2}
\end{aligned}$$

The log-likelihood function is:

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

To solve the MLE for β, denoted $\hat{\beta}$, we take derivative to the log likelihood function to find out the local maximum, which is also the local maximum of the likelihood function.

$$\begin{aligned}
\hat{\beta} &= \arg\max_{\beta} \ell(\beta) \\
&= \arg\max_{\beta} \ln(\ell(\beta)) \\
\frac{\partial}{\partial\beta}\ln(\ell(\beta)) &= \frac{\partial}{\partial\beta}(-\frac{n}{2}\ln(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(Y_i - X_i\beta)^2}{2\sigma^2}) \\
&= -\frac{\partial}{\partial\beta}\sum_{i=1}^{n} \frac{(Y_i - X_i\beta)^2}{2\sigma^2} \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{n} 2(Y_i - X_i\beta)(-X_i) \\
&= \frac{1}{\sigma^2}\sum_{i=1}^{n} X_i(Y_i - X_i\beta) \\
&= \frac{1}{\sigma^2}X'(Y - X\beta)
\end{aligned}$$

Where X and Y are the entire design matrix and vector. We set the derivative to 0 to solve $\hat{\beta}$

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

We use the same approach to solve the MLE for σ^2, denoted $\widehat{\sigma^2}$.

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

Set the derivative to 0 to solve $\widehat{\sigma^2}$

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

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

Model	Log Likelihood	AIC	BIC
AR(1)	-819.328	1644.656	1657.299
AR(2)	-786.540	1581.079	1597.938
AR(3)	-713.330	1436.660	1457.733
MA(1)	-780.702	1567.404	1580.047
MA(2)	-764.971	1537.941	1554.800
MA(3)	-763.434	1536.868	1557.941

The estimates suggested an **AR(3)** model, as it has the highest likelihood and lowest AIC and BIC.