Name:

Student# : Your signature:

The University of Western Ontario
Department of Statistical & Actuarial Sciences
SS2864b 2020

Final Take-Home Exam

Instructor: H. Yu April 19, 2020 Start 10:00 am with 24 hours due.

Instructions

a. All questions are to be answered by all students. Please print out this front page and **sign it**. You can use R-markdown or other means to generate a proper formatted document. When you are done, try to combine your signed page with your exam into one pdf or word file (otherwise submit two separate pdf or word files to owl). Submit your pdf or word file(s) before the due date: 10:00 am, April 20.

b. The mark allotment per (sub)question is given after each question number. Total mark is 90.

c. This exam has two purposes: to teach you some more things and to evaluate you. The former is more important than the latter. You are required to solve all questions in this exam. Try to follow steps given in a question.

Honour Pledge

Western University is a community of students, faculty, and staff involved in learning, teaching, research, and other activities. The University seeks to provide an environment of free and creative inquiry within which critical thinking, human values, and practice skills are cultivated and sustained. It is committed to a mission and to principles that will foster excellence and create an environment where its students, faculty, and staff can grow and flourish [1]. Members of the University accept a commitment to maintain and uphold the purposes of the University and, in particular, its standards of scholarship [2].

By submitting this take home exam, I unequivocally state that all work is entirely my own, and is submitted with an understanding of Western University's policy on "Scholastic Discipline for Undergraduate Students" [2].

---

[1] Text quoted from the Western University code of Student Conduct (effective April 25, 2019) available at: `https://www.uwo.ca/univsec/pdf/board/code.pdf`

[2] Text quoted from Western University's Academic Calendar under Academic Rights and Responsibilities, from the section on Scholastic Discipline for Undergraduate Students, available at:
`https://www.westerncalendar.uwo.ca/PolicyPages.cfm?Command=showCategory&PolicyCategoryID=1&SelectedCalendar=Live&ArchiveID=#Page_20`

1. [30] Jarque-Bera statistic is defined as

$$JB_n = \frac{n\gamma_n^2}{6} + \frac{n(\kappa_n - 3)^2}{24},$$

where $\gamma_n = \frac{1}{ns_n^3}\sum_{i=1}^n (X_i - \bar{X})^3$ and $\kappa_n = \frac{1}{ns_n^4}\sum_{i=1}^n (X_i - \bar{X})^4$ are standardized sample skewness and kurtosis respectively, and $\bar{X}$ and s_n are the sample mean and standard deviation respectively. If $X_1, \ldots, X_n$ are iid r.v.'s from Normal population, then JB_n follows $\chi^2(2)$ distribution (asymptotically). However, when n is small (less than 200), the p-value calculated from $\chi^2(2)$ is not accurate. In the following you will use both $\chi^2(2)$ and Monte Carlo simulation to find its p-value.

   (a) [6] Write a function **JB** with argument x and the return value should be the JB_n. Test your function with x=rnorm(100).

   (b) [6] Simulate the first data set $x1$ from

   ```
   n=100; eps=0; x1=rnorm(n, 2, 1+5*rbinom(n,1,eps))
   ```

   and two additional data sets $x2$ with eps=0.01 and $x3$ with eps=0.05, respectively. Use **qqnorm** and **qqline** on those three data sets and comment your findings.

   (c) [6] Use Monte Carlo simulation to simulate JB_n values under normal assumption (H_0). For each simulation, generate a normal data as x=rnorm(n) and then use the function **JB** from (a) to compute JB_n. Repeat it K times. You should write a function **JB.MC** with arguments n and $K = 50000$ (n is the sample size) to implement such a procedure and the return values should be a vector of those JB_n (K of them).

   (d) [6] Generate one output data set from **JB.MC** with $n = 100$ and plot its histogram (as density with more break points or you can plot the density curve generated by the function **density**). Then overlay $\chi^2(2)$ density with col=2. Comment your findings, in particularly, addressing their right tail behaves/matches.

   (e) [6] Use the function **JB** from (a) with $x1, x2, x3$ as inputs to compute their JB_n values. Then, for each calculated JB_n value, compute their p-values (right tails) through $\chi^2(2)$ distribution (the CDF of $\chi^2(2)$ in R is pchisq(q,df=2)) and the output data set from (d). Comment your findings, in particularly, addressing their differences. Do those p-value match what you have found in (b)? You may run (b) a few times to get some consistency and set a proper seed.

2. [30] Implement bootstrap procedure for AR(1) time series models. AR(1) is of the form

$$X_t = \theta X_{t-1} + \varepsilon_t,$$

where $\{\varepsilon_t\}$ is a sequence of iid errors with mean 0 and variance σ^2. A least estimator of θ is

$$\hat{\theta} = \frac{\sum_{t=1}^{n} X_{t-1}X_t}{\sum_{t=1}^{n} X_t^2}.$$

Do the following steps to implement the bootstrap inference for θ. No looping is allowed except (c)(ii).

(a) [6] Write a function **theta.est** with argument x to compute $\hat{\theta}$, where x is the vector of observations. Notice that $X_0 = 0$ is assumed in the formula. Test your function with x=huron-mean(huron) (the same time series **huron** used in Assignment 5. So please download huron data set from owl) and save the result to **my.est**.

(b) [6] Use the centered huron data x and its theta estimator **my.est** from (a) to compute the residuals as

$$\hat{\varepsilon}_t = X_t - \hat{\theta}X_{t-1},\ t = 1, 2, \ldots, n$$

and save the sample mean centered residuals $\hat{\varepsilon}_1 - \bar{\hat{\varepsilon}}, \ldots, \hat{\varepsilon}_n - \bar{\hat{\varepsilon}}$ to a vector **my.resid**, where $\bar{\hat{\varepsilon}}$ is the mean of those $\hat{\varepsilon}_t$. Check the distribution of **my.resid** by its histogram or density and qqnorm (qqline) and comment your findings.

(c) [12] Write a function **one.boot** with arguments **eps=my.resid** and **theta=my.est** to compute **one** bootstrap estimation of θ. The procedures are

i. Resample **eps** with replacement to get a vector called **eps.star**.

ii. Compute $x_t^* = \hat{\theta}x_{t-1}^* + \varepsilon_t^*,\ t = 1, \ldots, n$ (assume that $x_0^* = 0$).

iii. Compute and return the least estimator $\hat{\theta^*}$ based on $x_1^*, \ldots, x_n^*$ by calling the function **theta.est** from (a).

Test your function with default values to see if it works and test again to see if a different value is produced. Then run 10000 bootstraps (using **replicate** with **one.boot**) to produce a vector **output**.

(d) [6] Use the **output** from (c) to find the histogram or density curve of $\hat{\theta^*} - \hat{\theta}$ and comment your findings. Then construct 95% confidence interval for θ. You need to use the fact that

$$\hat{\theta} - \theta \approx \hat{\theta^*} - \hat{\theta} \text{ in distribution.}$$

Notice that this inference is for the huron time series only.

3. [30] NASA's GISS Surface Temperature Analysis (GISTEMP) is an estimate of global surface temperature change recorded monthly. A file, GLB.Ts_dSST.csv, can be downloaded from owl in Data sets folder. It contains monthly temperatures from 1880 to 2019. Please do the following steps to carry out some basic modelings. Any looping such as for, while, repeat **is not allowed**.

(a) [5] Import the dataset into R as a data frame. Create another data frame to keep only **Jan**, ..., **Dec** 12 columns. Then use **apply** function to generate a vector of yearly average temperatures from 1880 to 2019 and save it as **yearly.temp**. Plot it against **year**=1880:2019 (as a line) and comment your findings (any pattern changes).

(b) [7] Construct an objective function based on (sum of square)

$$\sum_{i=1}^{n}(\mathbf{yearly.temp}[i] - a - b * \mathbf{year}[i])^2$$

and use **nlminb** with start values (-10,0.1) to estimate a, b (saved as **ls.est**). You cannot use **lm** to find a, b though you can use it to check if your answer is right or not. Then compute the residuals as **resid**=**yearly.temp**-**ls.est**[1]-**ls.est**[2]*__year__. For model diagnostic checking based on residuals, do the following steps.

i. Add the fitted line (the fitted values computed as **fitted**=**ls.est**[1]+**ls.est**[2]*__year__) as col=2 to the original yearly temperature plot and comment how good or bad the fitted line is.

ii. Scatter plot of **resid** against **year** and comment out if there are any patterns or it is completely random.

iii. Use **qqnorm** and **qqline** to **resid** to see if it is normally distributed. Comment your findings.

(c) [7] Redo (b) with an objective function based on

$$\sum_{i=1}^{n}(\mathbf{yearly.temp}[i] - a - b * \mathbf{year}[i] - c * (\mathbf{year}[i])^2)^2$$

with start values (300, -1, 0.1). Notice that you have to add this option **scale=c(1, 300, 1000)** to **nlminb** otherwise it doesn't converge (check the usage of scale in **nlminb**'s help). For the fitted line, choose col=3.

(d) [7] Create a dummy vector as **dummy**= c(rep(0,85), rep(1,55)). Redo (b) with an objective function based on

$$\sum_{i=1}^{n}(\mathbf{yearly.temp}[i] - a - b * \mathbf{year}[i] - c * \mathbf{dummy}[i] - d * \mathbf{dummy[i]} * \mathbf{year[i]})^2$$

with start values (-1, 1, -1, 1). For the fitted line, choose col=4.

(e) [4] Comment the similarity and difference among three models. You can overlay three fitted lines together in a new plot to comment. Which is the best fitted model? What are your conclusions of rising temperature rates per decade?

Disclaim: Three models from above do not take the consideration of **yearly.temp** as a time series. Further study is needed for more accurate statistical modelings.