ECON 386 **Midterm Exam Practice Tasks**

Dr Levkoff

#1) (Cleaning Data & Outlier Removal) The next set of exercises involve applying two methods of outlier removal: i) deletion, and ii) deletion when necessary. Import the *NHIS_RAW.csv* dataset that we've been working with in class.

a) Use the *hist()* and *summary()* functions to explore the distribution of the following variables: *BMI, SLEEP (measured in hours), educ(measured in years), height (measured in inches), weight (measured in lbs)*. What are some indications that there are outliers in the data? Is the mean close to the median? If not, this could be a sign that the data is skewed towards the presence of an outlier – we should first remove these if we are going to do any predictive analysis (and will do so in the following steps). A summary of the larger data set from which this one was generated can be found here: https://www.cdc.gov/nchs/nhis/nhis_2007_data_release.htm. You may note that data coded as 996+ refer to missing responses or invalid responses. Also note that it is impossible to sleep more than 24 hours in a day.

b) Take note of the dimensionality of the data by using the function *dim()*. It tells you the number of rows (observations) and columns (variables) in the data set. This will be important to keep track of as we clean the data by removing observations associated with outlier responses.

c) There are several ways of cleaning a data via outlier removal in R. One of them takes advantage of the *subset()* function (pull up the documentation page if necessary). The cleaning process will essentially generate a sequence of new data sets at each step by "cutting" bad observations out: For example, if we wanted to create a "new" dataset (let's call it NHIS_1 since it is the first step in the cleaning process) by removing any observation (row) in the data that responded sleeping more than 24 hours per day, the code to do this is:

NHIS_1<-subset(NHIS, SLEEP<=24) ##deletes observations with invalid sleep response

To see how many observations that this step "trimmed," again, use *dim(NHIS_1)* and compare this to dim(NHIS). What happened to the number of rows? What percentage of our observations did we lose in this cleaning step?

d) Now we will work with the clean(er) data set *NHIS_1* which we created in the previous step. Repeat the process in part a) by creating new data sets (in sequence as you clean them) by keeping only observations that satisfy and removing outliers that don't satisfy the following constraints: *height<90; weight<450; BMI<60; educ<35*.

e) Alternative to creating a sequence of datasets, *{NHIS_1, NHIS_2,…}*, which may help us track all of the changes at each step of the iterative cleaning process, we could have done everything in a single step (much faster) by running the following code:
NHIS_A<-subset(NHIS, SLEEP<=24 & BMI<60 & educ<35 & height<90 & weight<450)

f) What percentage of observations were "trimmed away" once you've cleaned up each of the five variables above? Use the *dim()* function to compare the final cleaned data set to the original.

What is the down side to cleaning the data using this method of outlier removal (deletion of observations)?

g) You'll notice we weren't concerned with the *HHX*, *FMX*, or *FPX* variables. We can purge (remove) a column (variable) from the dataset by doing the following (here for just the *HHX* variable, but you can do it for the others as well):

NHIS_A$HHX<-NULL

Again, use the *dim()* function to verify that the resulting dataset has one fewer columns.

h) The last method of outlier removal cut out an entire observation, even when only a single variable response was invalid - we lost other pieces of useful information by removing *all* of the variable data associated with that observation. This next option of cleaning up a variable, rather than "killing" the whole observation, simply replaces an invalid response by recoding it with the "*NA*" value. When R sees "*NA*" it knows to omit that particular observation without deleting the entire row of variable information. For example, let's create a new dataset called NHIS_B to apply this alternate methodology by starting fresh with the original dataset:

NHIS_B<-NHIS

Next, we will use an alternate method of subsetting to assign invalid responses a value of "*NA*" rather than removing the entire observation. For example, we can clean the sleep variable using this method by running:

NHIS_B$SLEEP[NHIS_B$SLEEP>24]<-NA ##assigns NA to the invalid responses

Note that even after this step, the dimensionality of the data hasn't changed (verify using *dim()*) because we didn't delete the observations – we simply recoded the invalid responses.

i) Repeat this process for the remaining variables of interest. Use the View() and dim() functions to verify that you're not losing any information throughout this process.

j) If you were to compute a summary statistic of a variable after recoding some of the values as "*NA,*" you'll notice the following code results in a value of "NA":

mean(NHIS_B)

This is because the mean is undefined for NA values present in the data. In order to have R "skip" these newly coded observations/variables, use the following command:

mean(NHIS_B, na.rm=TRUE) ##tells R to ignore the NA values

Many functions in R have an argument to control what R does with NA observations.

#2) (Preprocessing & Exploratory Analysis) This problem will require you to download and import the "power_plants.txt" file (already in the ECON386SP19 repo) and perform some preprocessing to summarize and explore the data with visualizations and summary stats. It is assumed already that you've

properly set your current active working directory appropriately per question #1) and that you've cloned the ECON386SP19 repository from Github.

a) Since it is a slightly different format of file (.txt as opposed to .csv) than "NHIS_Uncleaned.csv" file, the command to import the data is slightly different. We create a new variable to store the data set called *power_plants*:

   power_plants<-read.table('power_plants.txt', header = TRUE)

   Note the *header = TRUE* argument – this makes sure that the column names are treated as the variable label and not as an observation (if you don't do this and use the *View()* function, you'll see what I mean).

b) Use the *View()* function to look at the general structure of the data. You'll notice that each power plant has observations for several years – the structure of this dataset is known as a *panel* dataset (it includes both *time series* elements and *cross-sectional* elements). How many rows and columns are in the dataset? How many power plants? How many years? You can find more details about the measurement of each variable in the codebook found in the GSBA502REPO course repository. The document is called *codebook_power_plants.txt*.

c) Use the *aggregate()* function to summarize data in a particular dimension. For example, if we wanted to sum (add) the total amount of emissions (say, NOX) generated by the power plants in the sample for a given year for all years, we can create this summary table in a new dataset by creating a variable called *NOXByYear* and use the code below to apply the *aggregate()* function:

   NOXTotByYear<-aggregate(NOX ~ Year, power_plants, sum)

   Use the *barplot()* function to generate a time series plot of the aggregate NOX emissions for the firms across the sample. In which year were total emissions of NOX the smallest? The largest? Use the following code to help:

   barplot(NOXByYear$NOX, names.arg = as.character(NOXByYear$Year))

d) Repeat part c), but instead, create a variable called *SO2AvgByYear* that summarizes the SO2 emissions by taking the average (expected value) in each year across all plants. You'll have to think about modifying the code provided in part c) to compute the mean of the emissions rather than the sum of the emissions. Between which two years in the sample did average emissions decrease the most? Use the *barplot()* function to analyze the newly summarized data.

e) The variable *PhaseI* is a binary (dummy) variable that indicates whether or not a firm was regulated by the Clean Air Act of 1990. If the variable is equal to 1, then the firm was regulated and forced to cut back emissions and 0 otherwise. Create a plot that measures *Energy* on the vertical axis and *NOX* on the horizontal. Color the plot by the variable *PhaseI* by running the code below:

   plot(power_plants$SO2, power_plants$Energy, col = power_plants$PhaseI+1)

   You can add a (main) title to your plot, as well as x (xlab) and y (ylab) axis labels:

plot(power_plants$SO2, power_plants$Energy, main = "Energy Production vs. SO2 Emissions \n Daily Averages", xlab = "Short Tons, SO2", ylab = "Mwh (thousands) Generated", col = power_plants$PhaseI+1)

You can also add a legend to see how things are associated:

legend("bottomright", legend=c('Regulated Plants', 'Unregulated Plants'), pch = c(1,1), col = c(2,1))

Try and experimenting with the *pch* and *col* arguments (change the numbers in the c() vector) to see how you can add some variety to the shapes and coloring in the legend (I chose the parameters to match how we colored the data as well as the default shapes).

f) By observing the plot you generated in part f), what can you conclude about comparing the regulated plants to the unregulated plants by looking at the amount of pollutant generated per unit of electricity produced?

g) Repeat the analysis in e) and f) by plotting *Energy* against *SO2*. Do you draw the same conclusion regarding the choice of regulated plants vs. unregulated plants?

#3) (Linear Model Building & Diagnostics) We will now use the above *power_plants.txt* dataset which you have some experience cleaning, summarizing, manipulating, and visualizing, to build some statistical models utilizing our linear regression toolkit.

a) Use the *plot()* function to plot the variable *Energy* on the vertical axis and *GrossCons* on the horizontal axis. What does the slope seem to approximately be? What fraction of the gross energy consumption seems to be wasted (in that it doesn't account for energy production)? Does this seem to differ when looking at the regulated vs. unregulated plants? (hint: color the plot by the *PhaseI* variable using some of the previous code provided).

b) Build a linear model of the form $Energy = \beta_0 + \beta_1 GrossCons + u$ using all of the data available and store it in a variable called *M1*. Are the parameter estimates for the intercept and slope significant? How much does energy production (in Gwh) change with a one unit increase of gross heat consumption (GrossCons)? What does this imply about the efficiency of using energy to generate energy? What percentage of the variation in energy production (*Energy*) across plant-years is explained by variation in gross heat content consumed (*GrossCons*)?

c) If instead, you estimated the model, $GrossCons = \beta_0 + \beta_1 Energy + u$, explain how the slope coefficients are related to those from the previous part (hint: a graph might help – use the *abline()* function). What is true about the R-squared here compared to in part b)? Why?

d) Note that in the previous parts, we got a negative intercept term suggesting that we would be producing negative energy when using zero heat content as an input in the production process. Since this is physically impossible, it may make sense to constrain our regression to force the regression line through the origin (so that zero input generates zero output, consistent with physical laws in this case). To do this, we can simply modify the function call

to generate the linear model above by including a zero on the right-hand side set of variables and store it in a variable called *M2*:

M2<-lm(Energy ~ 0+ GrossCons, power_plants)

e) If we also controlled for emissions and re-estimated the same model above, but with *NOX* and *SO2* also showing up on the right hand side, how do your results change from j) (call this *M3*)? Are all three control variables *statistically significant* in their marginal effects on the output variable (*Energy*)? If not, propose a more appropriate model and re-estimate it (call this *M4*). How does the R-squared change in comparison to when we didn't control for any pollutants?

f) If we also build a model (call this one *M5)* to predict energy production by controlling for *SO2, Capital, Labor,* and *GrossCons*, and forcing the regression through the origin consistent with our observation earlier, are all of the effects of the regressors significant? Are there any unexpected effects captured in your model? Why or why not? What percent of the variation in Energy is explained by the regressors?

g) If you compare the means of the residuals (maybe use the *hist()* function to actually plot the residuals) across models *M2-M5*, you'll notice they behave differently than in model *M1*. Why?

#4) (Non-linear Transformations with Linear Models) We will utilize the *cleaned* NHIS data resulting from #1 part c), the *NHIS_A* cleaned data with outliers removed, to build some models now that you have conducted some / preprocessing and some exploratory analysis.

a) If you were to create two linear regression models that tried to predict the height of a person in the data set given information on their weight (a single variable only), which model "fits" the data better: one that forces the data through the origin? Or one that provides the extra degree of freedom to choose the intercept parameter freely? (hint: compare the measures of "goodness of fit"). What is the tradeoff?

b) If you built a nonlinear model (with an intercept) that also captured the quadratic effect of the weight variable relative to the linear models you build above, does the quadratic effect seem important? Why or why not?

c) Consider building the following linear model where we have transformed one of the variables using the natural logarithm transformation (log() in R): $Height = \beta_1 \ln(Weight) + u$. Estimate the slope coefficient. How do we interpret the beta parameter in this case? Explain.

d) Consider building the following linear model where we have transformed one of the variables using the natural logarithm transformation (log() in R): $\ln(Height) = \beta_1 Weight + u$. Estimate the slope coefficient. How do we interpret the beta parameter in this case? Explain.

e) Consider building the following linear model where we have transformed one of the variables using the natural logarithm transformation (log() in R): $\ln(Height) = \beta_1 \ln(Weight) + u$. Estimate the slope coefficient. How do we interpret the beta parameter in this case? Explain.

f) The "knight owl phenomenon" is the idea that people of higher educational attainment tend to get fewer hours of sleep and spend more time up late at night (this is just a theory). The causal

mechanism could also work in either direction – people that spent more time in college stay up later on average studying because they are in college… Either way, the causal mechanisms suggest an inverse relationship – or negative correlation - between hours of sleep on average and the level of educational attainment. Does this effect show up significantly in the data? Use a linear model to assess the extent to which an extra hour of sleep affects the level of educational attainment. Consider all variants of logs and level combinations and interpret the coefficient estimates in each case. You don't need to force the regression through the origin here.

g) Does the effect describe in e) seem to apply to both men and women in the sample? Explain. (hint: subset appropriately OR add a dummy variable – only use a linear model here with no log transformations)

#5) (The Gaussian / Normal Distribution) The normal (also referred to as the Gaussian) distribution is one of the most important probability distribution functions in statistics. The analytical form of the probability density function (ie: the formula for the "bell shaped curve") is fairly complicated and is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2] \quad for \quad -\infty < x < \infty,$$

where μ and σ are the mean and standard deviation of the underlying distribution, respectively. The corresponding CDF is defined (by calculus) to be

$$P(X < x) = F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2]dy,$$

for which there is no analytical solution. Fortunately for us, R has these functions preprogrammed. We can also use R to independently sample (with replacement) from the same distribution repeatedly:

a) Use the code chunk,

x<-rnorm(n=100, mean=0, sd=1)

to generate a vector of 100 draws from the standard normal distribution, which is a special case of a normal distribution with mean zero and standard deviation one with the draws in a variable called *x*.

b) Use the code chunk,

hist(x, breaks = 30, prob = TRUE, ylim = c(0,.5), xlim = c(-4,4))

to plot the empirical distribution for the 100 draws that you've sampled in a) using 30 break points (bins) using R's base plotting system. Use the code chunk,

curve(dnorm(x, mean = 0, sd = 1), col = "darkblue", lwd = 2, add = TRUE)

to plot the actual density function for the normal distribution on the same diagram to see how well your empirical distribution reflects the actual (population) density function.

c) If you repeat a)-b) with 10,000 draws, how does your answer change? Does the empirical distribution provide a better reflection of the underlying density function? Explain. (hint: this has to do with the *Law of Large Numbers*)

#6) (Verifying the Empirical Rule) Use the function, *pnorm(x, mean=0, sd=1)* to evaluate the cumulative distribution function, $F(x)$, corresponding to the standard (mean=zero, standard deviation=one) normal distribution (you will need to think carefully about how to use the CDF and which values of *x* to plug in).

a) What is the probability that an observation falls within one standard deviation of the mean (on either side of the mean)?
b) What is the probability that an observation falls within two standard deviations of the mean (on either side of the mean)?
c) What is the probability that an observation falls within three standard deviations of the mean (on either side of the mean)?
d) Summarize your results from a)-c) in a table. Do your results change if the normal distribution is not standardized? Explain. (hint: summarizing this result is known formally as the Chebychev's rule)
e) What is the probability of observing a value more extreme than positive 2.5 when sampling from this distribution?
f) What it the probability we observe a value more extreme in absolute value than 1.5 when sampling from this distribution?
g) Given that a standard normal randomly distributed variable *x* is larger than 1, compute the conditional probability that *x* is between 2 and 3 conditional that it is larger than 1 (hint: use Baye's rule and a carefully drawn diagram).

#7) (Verifying the Central Limit Theorem) The CLT is one of the most important results from theoretical statistics. It suggests that, when taking the average of random variables sampled from the same distribution, the distribution of the average of these random variables tends to be normally distributed despite the form of the underlying distribution from which each random variable is sampled. Essentially, the result implies that the assumption of normally distributed data applies to a wide arrange of problems.

a) Use the function *runif(n, a, b)* to sample *n* draws from a uniform distribution defined on the interval *[a,b]*, store the draws in a variable called *x*, and use the *hist( )* function (see above for more arguments or type *?hist* into the console window to pull up the R documentation page) to verify the empirical distribution of the data visually.

(Difficult) Create two variables: one called *reps* and one call *draws*. Set *reps* equal to 1000 and *draws* equal to 30. We are going to sample 30 points from the uniform distribution, take the average of these points, and standardize the data (by subtracting the sample mean and dividing through by the sample standard deviation/error). You (well, your computer) will repeat this exercise 1000 times (repetitions) by using a carefully constructed *for* loop (see the R documentation and SWIRL exercise relevant to this function) and storing the average of the 30 draws at each iteration into a vector (you'll need to index this vector for each iteration of the loop). To standardize the data at each repetition/iteration within a loop,

you'll need to divide by the sample standard error for a uniformly distributed variable after subtracting the sample mean from each observation. The mean for the uniform distribution is given by

$$mean = \frac{1}{2}(b + a)$$

The small sample standard error (this is slightly different than the standard deviation) for the uniform distribution is given by

$$se = \frac{\sqrt{\frac{1}{12}(b-a)^2}}{\sqrt{n}}.$$

b) Plot the empirical distribution of the means sampled in the 1000 repetitions using the *hist()* function and compare this to the actual standard normal distribution's probability density function by plotting the theoretical density using the *curve()* function (see #1 part b). Does the empirical distribution of the averages drawn from the uniform distribution look normal?

#8 (Simulation I) In most applications of predictive analytics, practitioners typically never are able to observe the data generating process (but that doesn't stop us from trying to guess it!). However, we will consider exploiting R's environment of drawing random variables to *simulate* a data generating process and apply our regression modeling to see how well the model can be built from the constraints of a particular sample from the DGP.

a) Let's consider synthesizing first a linear DGP of the form $y_{synth} = \beta_0 + \beta_1 x + \varepsilon$, where (β_0, β_1) are population parameters and ε is a random error(shock). We will simulate some draws from the DGP by locking in values of the slope and intercept parameters. Let's start using the values $(\beta_0, \beta_1) = (5,2)$. Let's also randomly sample the errors from a standard normal distribution. Start with a sample of N=2 data points and use the set.seed() command to lock in the random numbers to be generated.

b) To synthesize the x variable data into x_{synth}, let's draw data from a uniform distribution using the *runif()* function from the interval [0,10]. Now synthesize the y_{synth} data using the DGP equation from a) and your draws of the synthesized x and ε. Create a dataframe with two columns: the first containing the synthesized y data and the second containing the synthesized x data.

c) Plot your data in an y_{synth} vs. x_{synth} scatter plot. Plot the DPG line (absent the error term) in red. Do the data fall close to the line or not? Why?

d) Build a linear regression model of the form $\hat{y} = \widehat{\beta_0} + \widehat{\beta_1} x$ from your sample of data. How close are the sample estimates of the intercept and slope to the true population parameters implied by the DGP?

e) Plot the regression line in d) on top of the scatter plot of data and the DGP. How well does the regression line capture the DGP? How well does it fit the data? Would the model work well if we resampled two different points?

f) To compute the *in-sample error* we will take the sum of the squared residuals, divide by the number of observations (take the average squared residual), and then take the square root of the result.

g) What is a bit trickier, is computing the *out-of-sample error*, which normally we don't know since that is based on observations we didn't use / see to build our model. However, since we have simulated a DGP, we can actually compute the simulated out-of-sample error by measuring how far a regression line predicted value $\hat{y}$ is from the actual value implied by the DGP (absent the noise). To do this, we will compute $y_{DGP} = \beta_0 + \beta_1 x_{synth}$ *without* the noisy error term.

h) To compute the *out-of-sample error,* we will then measure the average value of $(\hat{y} - y_{DGP})^2$ and then take the square root to compute what is known as the root mean square error, our measure of the *out-of-sample* error relative to the entire population function (the DGP), and not just relative to the sample of data that we drew (the *in-sample* error measures the latter case).

i) Repeat this process for N=10, 100, and 1000 points sampled from the DGP. What do you notice as the sample size increases for the same model complexity (the linear model with an intercept term and a single slope coefficient)? What happens to the *in-sample error*? The *out-of-sample error*?

#9) (Simulation II & GGPLOT2) Create two vectors of observation of two different variables by sampling 100 draws from the following distributions: the first variable will be sampled from the normal distribution with mean 2 and standard deviation 5. The second variable will be sampled from the uniform distribution over the interval [5,15].

a) Using the ggplot2 package, create a histogram colored by variable that overlays both sets of observations on the same frequency plot of the empirical distributions.

b) Add a nonparametric smoothed kernel density estimator to each of the empirical distributions colored the same way.

c) Add a vertical line that captures the mean of each group in the sample.