Problem Set #2 - Exploratory Data Analysis

Due Friday September 20 at 5:00 pm

For this problem set you will submit a pdf document to the Canvas site with your answers. Make sure comments tie directly to the appropriate figures and tables. The best way to do that is to use the figure/table caption as your place to comment. Include your code when indicated.

Simulation

1. **Missing Data** We will explore missing data mechanisms and basic imputation strategies. Follow these steps showing your code. Don't forget to set your seeds.

   (a) Set your number of people 'n' to 1000

   (b) Simulate a variable Z $\sim N(0,1)$

   (c) Simulate a variable X where the data generating distribution is:

   $$X = Z + \varepsilon \text{ where } \varepsilon \sim N(0,1)$$

   (d) Calculate the mean of X

   (e) Make 10% of the X values missing. Calculate the mean of this new X. Comment on any differences or similarities

      i. What is the missing data mechanism?

      ii. Do we expect $E(X^c) = E(X^O)$?

      iii. Using the mean of the observed X values, impute in the missing X-values.

      iv. Does this change our estimate for the mean of X? Do we expect it to?

   Hint: If you are unsure if your empirical (estimated) means are similar enough you can consider calculating 95% CIs for the mean

(f) Return to your complete data. Among $Z > 0$, make 10% of the X values missing. Calculate the mean of this new X. Comment on any differences or similarities

   i. What is the missing data mechanism?

   ii. Do we expect $E(X^c) = E(X^O)$?

   iii. Using the mean of the observed X values, impute in the missing X-values. Does this improve our estimate of the mean of X?

   iv. We will perform conditional imputation:

      A. Using the observed X,Z pairs fit a linear regression model, regressing X onto Z. This can be thought of as a simple predictive model for X based on Z.

      B. Using this fit, generate predictive values for X, based on Z. **hint:** use the predict() function for lm.

      C. Calculate new imputed mean of X. How does this compare to the true mean of X.

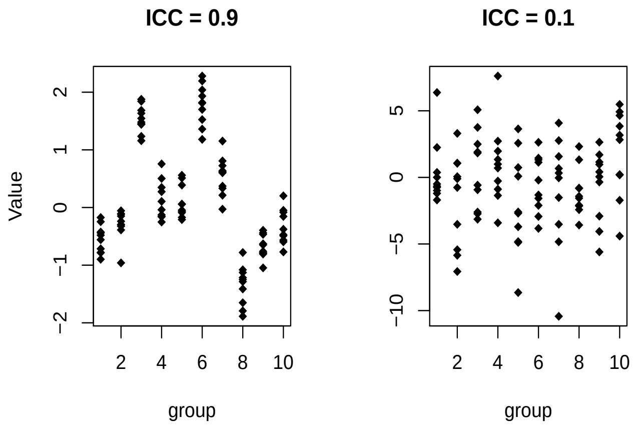
(g) What would happen if in (f) we had set the missingness degree to be 100%? Would we be able to recover the true mean value?
Further Exploration: If you are inclined, go back to (f) and vary that percentage. At what point are we unable to recover the true value. Feel free to make any graphical results to justify your findings.

2. **Intraclass Correlation Coefficient (ICC)** The ICC is a statistic that measures the ratio of within group to between group variance. It assess how tightly correlated a measure is based on some grouping variable. It is particularly useful in longitudinal studies where patients are measured repeatedly overtime. For example we may measure blood pressure on patients multiple times. The ICC will assess whether there is more variation within people than between people. The ICC varies between 0 - 1. A value of 1 indicates that there is more between person variation while a value of 0 indicates more within person variation (see Figure 1)

   There are multiple ways of defining the ICC (see `http://en.wikipedia.org/wiki/Intraclass_correlation`). We will use the random effects definition:

$$Y_{ij} = \mu + \alpha_j + \varepsilon_{ij}$$

   Where Y_{ij} is the outcome for person i in group j; α_j is a random effect shared by all


ICC = 0.9

ICC = 0.1

Figure 1: Hi ICC (left) - values within a group are highly correlated;
Low ICC (right) - there is little correlation among values in a group

people in group j. We assume $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, $\alpha \sim N(0, \sigma_\alpha^2)$ and $\alpha_j \perp \varepsilon_{ij}$. Then:

$$\text{ICC} = \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_\varepsilon^2}$$

(a) Using the random effects definition of an ICC, write a function that simulates data with a user specified ICC. The function should take as arguments a way to specify the desired ICC, a total sample size and the number of groups. You can allow the groups be of the same size. **Show your function code.**
Hint 1: Your function will need to "solve" for the unknown values given a fixed ICC in the above equation
Hint 2: Since you have one equation and two unknowns, you can arbitrarily fix one of the unknowns

(b) Simulate data that has a theoretical ICC of 0.1 and 0.9 across $k = 10$ groups using a total sample size of $n = 100$. Use the R function ICCest() in the ICC package to estimate your empirical ICC and 95% confidence interval.

(c) Repeat using a total sample size of $n = 1,000$.

(d) Keep the sample size at $n = 1,000$ but increase the number of groups to $k = 100$.

(e) Comment on differences in the precision of your estimate based on k and n. Graphically display your results. Feel free to do more testing to justify your conclusions.

Working With Data

Download the Diabetes data set from the UCI Machine Learning Repository:
`https://archive.ics.uci.edu/ml/datasets/Diabetes`
You will need to:

1. click on Data Folder
2. download the file diabetes-data.tar.Z
3. extract the tarball

This dataset contains patient records for 70 individuals, with each person having their own file. Read in the file for one person - you can choose any person. Comment in your write-up which person you chose.

1. **Summarizing Data -**

   (a) Select the 3 pre-meal blood glucose measurements (codes 58, 60, 62). Create a table reporting any relevant summary statistics. Provide any interpretation.

   (b) Calculate the ICC for these 3 measurements. Interpret the results.

2. **Working with Dates -** Time series data involves working with dates. Use the "Time" and "Date" columns to create a new "Date.Time" column. Two useful packages for working with dates are lubridate and chron. **Show your code for the new date creation.**

   (a) Choose an appropriate plot to display the 3 blood glucose measurements over time.

   (b) Using the 3 meal measurements from above, align the data on a daily basis, i.e. transform the data to a wider format where each line represents a day and has a measurement per-person. Comment on any generated missing data. **Show your code.**

   (c) Correlation plots are a way to visualize multivariate relationships. Use the corrplot package to make a correlation plot. What happens if you ignore the missing data?

(d) One version of single imputation appropriate for longitudinal data is called "hot-deck" imputation or "last observed carried forward (locf)". Here the data are ordered (typically by time) and the last observed value is imputed into any time slots without an observed value. Using the data above perform two versions of LOCF and recalculate your correlation plots.

   i. Carry the last observation forward separately for each of the 3 categories

   ii. Carry the last observation forward from the time of day (i.e. impute glucose before lunch using glucose before breakfast).

   Note: The `zoo` package has the function `na.locf()`

   Comment on implications of each approach. Which approach do you think is better?

3. **Smoothing -**

   (a) Fit a lowess smooth over time for each of the three measurements separately. Use the R function `lowess()`. Set the bandwidth f to 0.1. Plot the results

   (b) Set the bandwidth f to 0.9. Plot the results

   (c) What conclusion does each bandwidth provide

   (d) Plot what you think is the "best" bandwidth for each measurement. Comment why you chose this value.

   (e) Choose another smoothing method discussed in class (i.e. running means, kernel smoothing, splines). Do you think this fits the data better, worse or no different? Justify your conclusions visually.

4. **Comparing Results -** Choose someone in the class to compare your results to (who used a different person). Comment on any similarities or differences. Mention the name of the person you worked with.