Project 1

STAT 28

Data Introduction

You will be looking at a dataset from Centers for Medicare and Medicaid Services (https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/medicare-provider-charge-data/inpatient.html). Medicare is the US program that assists in covering the costs of health expenses for people who are 65 or older, as well as some younger people with disabilities. The dataset we will be using gives the *cumulative* charges for procedures billed to Medicare for more than 3,000 U.S. hospitals for Fiscal Year 2011 (Fiscal Year: the 12-month period ending on 30 September of that year, having begun on 1 October of the previous calendar year). The dataset is intended to help Medicare recipients to have a sense of the costs at different institutes or for different procedures (Medicare beneficiaries still will have remaining out-of-pocket costs after the federal government pays its portion).

You will be using this dataset to practice topics including probability distributions, visualization of distributions, and group comparison methods.

The dataset is setup so that each combination of diagnosis and hospital is a separate entry. So the first entry gives information regarding the total costs at Fairbanks Memorial Hospital in Arkansas (the hospital) for the diagnosis of simple pneumonia and pleurisy with complications. A diagnosis/hospital combination is only included if the hospital has charges related to the particular diagnosis. Keep in mind that 'hospital' covers a range of institutes of different sizes. Some hospitals might be big organizations, while some might be more local clinics that do not handle specialized conditions or procedures.

Variable definitions

The data is found in the dataset `combinedData.csv`. This file is a comma-deliminated text file with headers for each column variable.

The descriptions below of each the variables are largely taken from the website.

- `DRG.Definition`: Name of diagnosis [Diagnosis-related group=DRG]. "CC" added to the end stands for complication or comorbidity due to the diagnosis and "MCC" stands for a major complication or comorbidity.
- `Provider.Id`: Hospital ID
- `Provider.Name`: Name of Hospital
- `Provider.City`: City of the Hospital
- `Provider.State`: State of the hospital
- `Total.Discharges`: the number of beneficiaries who were released from the inpatient hospital after receiving care for this diagnosis
- `Average.Covered.Charges`: The provider's average charge for services covered by Medicare for all discharges in the DRG. These will vary from hospital to hospital because of differences in hospital charge structures.
- `Average.Total.Payments`: "The average total payments to all providers for the MS-DRG including the MS-DRG amount, teaching, disproportionate share, capital, and outlier payments for all cases. Also included in average total payments are co-payment and deductible amounts that the patient is

responsible for and any additional payments by third parties for coordination of benefits." You can interpret this variable as the total amount that paid by the Medicare *and* patients to the hospitals

- `Average.Medicare.Payments`: "The average amount that Medicare pays to the provider for Medicare's share of the MS-DRG. Average Medicare payment amounts include the MS-DRG amount, teaching, disproportionate share, capital, and outlier payments for all cases. Medicare payments DO NOT include beneficiary co-payments and deductible amounts nor any additional payments from third parties for coordination of benefits." You can interpret this variable as the amount paid by the Medicare. The difference between `Average.Medicare.Payments` and `Average.Total.Payments` will will assume to be the average amount paid by the patients at that hospital.
- `Provider.Zip.Code`: Zip code of Hospital. The zip code is a system used in the U.S. to facilitate the delivery of mail and is five numbers printed directly after the address. They generally indicate a coherent region of the US that is coherent for delivering mail. They can vary widely in size and shape, and can cross city or town boundaries (but not state boundaries).
- `regions`: A variable created for this project that classifies the hospital into one of four regions of the US ("midwest","northeast","south", and "west"). These classifications were made based on the state in which the hospital is located. The geographic regions are divided as follows

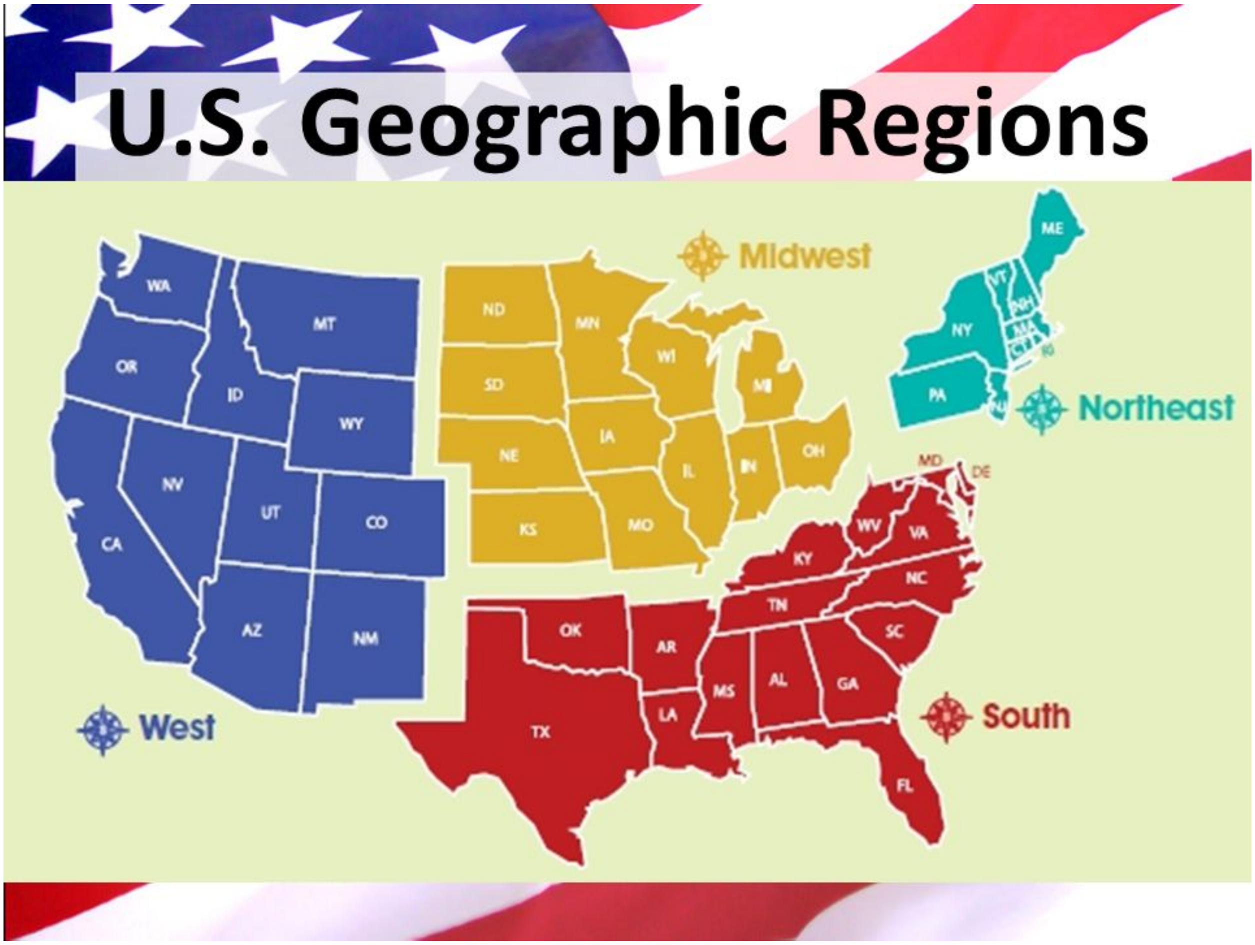

Figure 1: **The geographic region map of the United States**

- `Urban`: A variable created by for this project that classifies the hospital into their level of rural versus urban. We rely on the Census Bureau which divides the country into regions and assigns those regions one of three values:

- Urbanized Areas (UAs) of 50,000 or more people;
- Urban Clusters (UCs) of at least 2,500 and less than 50,000 people.
- Rural encompasses all population, housing, and territory not included within either of the above urban areas.

The Census Bureau also identifies which of these three types of areas are covered by any zipcode. Thus we assigned each hospital to these areas based on the zip-code in which the hospital is located. However, a zip code frequently covers more than one such region. Therefore the following codes are given in the variable `Urban`:

- 1 = only rural regions in the zipcode
- 2 = combination of rural and Urban Clusters in the zipcode
- 3 = only Urban Clusters in the zipcode
- 4 = combination of Urban Clusters and Urbanized Areas in the zipcode
- 5 = only urbanized Areas in the zipcode
- 0 = a mix of Urbanized Areas and Rural (and perhaps also Urban Clusters) in the zipcode

Except for the last category, you can roughly think of these showing increasingly more urbanization as the numbers increase. However, it's important to keep in mind that while the zip code may have a mix of, say, rural and urban cluster types, we have no idea where the hospital is in the zip code and what portion of the zip code the hospital might serve. In particular, a zip code that has both Urbanized Areas (the highest level of urbanization) and rural areas (`Urban=0`) does not tell us much about the hospital's region.

Project Assignment

Goal The overarching goal of this project is to evaluate how the the amount a patient has to pay (i.e. after Medicare) change for different regions or urban/rural areas of the country.

It is also important to consider that different diagnoses are expected to have different costs – treating heart failure is not going to cost the same as a broken leg. So in making this comparison, we will focus on the following diagnoses that cover a range of different conditions that have different implications as to the hospital stay required and the procedures necessary:

- Chronic Obstructive Pulmonary Disease (COPD) This is an umbrella term used to describe progressive lung diseases, characterized by increasing breathlessness.

  `192 - CHRONIC OBSTRUCTIVE PULMONARY DISEASE W/O CC/MCC`

- Heart failure

  `293 - HEART FAILURE & SHOCK W/O CC/MCC`

- Hip/Pelvis fractures

  `536 - FRACTURES OF HIP & PELVIS W/O MCC`

- Diabetes

  `638 - DIABETES W CC`

Specific Tasks

1. *Data entry* (20 points) Do the following basic setup of the data:
   - Read the data into R
   - Subset your data.frame into only data corresponding to the above diagnoses
   - Make `Urban` and `regions` factors, if they are not already, with relevant labels.

- Create new variables in your data.frame that define the following: the absolute amount the patient pays (`PatientPays`) and the percentage of the payment that is paid by the patient (`PctPatientPays`).
- Create a factor variable `urbanByRegions` in your data.frame that gives the cross of `Urban` and `regions` (e.g. rural, South, rural Northeast, etc.) by using the `:` command.

Here is a simple code to demonstrate how the `:` command works to make a new factor from two existing factors, so that the new factor is the cross of the two. The code first makes two factors of the same length, then creates a new factor with the `:` command that is the cross between these. Notice that before I do `droplevels`, there are empty levels that hang around that are crosses that don't exist (there are no red oranges). `droplevels` creates a factor that gets rid of these:

```
factor1<- factor(c("apples","apples","oranges","apples","oranges"))
factor2<- factor(c("red","green","orange","green","orange"))
crossFactor<-factor1:factor2
levels(crossFactor)
```

```
## [1] "apples:green"   "apples:orange"  "apples:red"     "oranges:green"
## [5] "oranges:orange" "oranges:red"
```

```
table(crossFactor)
```

```
## crossFactor
##   apples:green  apples:orange     apples:red  oranges:green oranges:orange
##              2              0              1              0              2
##    oranges:red
##              0
```

```
crossFactor<-droplevels(crossFactor)
levels(crossFactor)
```

```
## [1] "apples:green"   "apples:red"     "oranges:orange"
```

```
table(crossFactor)
```

```
## crossFactor
##   apples:green     apples:red oranges:orange
##              2              1              2
```

- Apply `droplevels` to your data.frame to remove "extra" levels that exist (make sure you save the result to replace the existing data.frame with the new one after running `droplevels`)

To demonstrate successful completion of this question, print out the `summary` of your changed data.frame at the end of your code. You do not need to provide any commentary.

2. *Basic Summaries* (30 points) Provide basic summaries of the two variables `PatientPays` and `PctPatientPays` as well as histograms and comment on them. Consider whether a transformation would be helpful. Also provide a cross-tabulation/contingency table of `Urban` and `regions` and comment on the results and what that means for the groups in your `urbanByRegion` variable. You should remember to separately treat the four diagnoses and consider whether a transformation would result in a better visualization. What problems do we face in looking at Urban with values `1`, `3` or `4`?

3. *Exploration of distributions in groups* (50 points) Create 1-2 useful visualizations of the data to understand how the two variables `PatientPays` and `PctPatientPays` vary within the groups defined by the `urbanByRegions` variable that you created in Q1. Exclude values of `Urban` corresponding to `1`, `3` or `4`. You should remember to separately treat the four diagnoses and consider whether a transformation would result in a better visualization.

   You must provide commentary that explains what plots you are showing, interprets these plots and describes what these plots tell you about the variables.

4. *Perform inference to evaluate the difference between the groups* (80 points) Perform both hypothesis testing and create confidence intervals to assess differences between the groups defined by `urbanByRegions` for both of these variables. You should provide a visualization of the results of your tests and confidence intervals, rather than print outs of p-values and intervals. Again, you should remember to separately treat the four diagnoses and exclude values of `Urban` corresponding to `1`, `3` or `4`.

   You must provide commentary to explain what you have done, interpret the results of your inference, and provide the conclusions you would make based on the results. Your explanation of what you have done should discuss which tests you performed and why, and the impact of any multiple testing corrections you have done.

5. *Comment on the overall analysis* (30 points) Reflect on the analysis you have done above. Consider the following questions to get you brainstorming: What did you run into that made you hesitate as to whether something was a good idea? What are limitations you see of this analysis in understanding how the amount a patient has to pay differs in different parts of the country, both statistical problems and intrinsic problems in this data? What might you do differently or add if you had the freedom to do any analysis? What other questions might you ask using this data? Be creative and critical!

More Instructions

Format You are expected to create a `.Rmd` file for this project from scratch. RStudio will setup a new `.Rmd` file for you if you go to `File > New File` in the menu bar. The text from this pdf should not be part of your `.Rmd` file. Like the homeworks, you will turn in only a compiled .pdf to gradescope. An actual analysis would blend these components together into a single narrative. However, for grading purposes, we have divided the project into the questions described above and each of the questions should be answered in a separate section and appropriately labeled so that you can tell gradescope which pages correspond to which question.

This project is intentionally more open-ended than the homework so as to be more reflective of an actual analysis of the data. The commentary on questions will be half of the points of the grade for each question. The commentary for each should be between a half a page to a page in length, though there is no limit if you need to go longer and some questions might need more. The commentary should be just regular text typed in your `.Rmd` file (it does not need a `>` in front of it like the homework). For the purposes of easy grading, give the code first, and then follow with the commentary (again an actual report would interleave these, but this will make the grading simpler).

Code For the code, make use of `for` loops to cover the four diagnoses; you should not be rewriting your code for each one. Make sure you provide titles to your plots that describe which plots go with which diagnosis. Since cost of diagnoses are on completely different scales, it doesn't make sense to put multiple diagnoses on the same actual plot (though you are welcome to use `par(mfrow=c(...))` to make multiple plots in one figure).

Provide some *very* brief (1-line) comments to orient the grader as to what code corresponds to what. For example, `#code for histogram` or `#code for plotting density curves` etc. However, your commentary should give the explanation of your work, not the comments in the code. The comments are just so someone can quickly find the code that corresponds to the explanation in your commentary text.

You should make sure your code prints out only the information needed for answering the question. For example, some people have been turning in homework that prints out the entire dataset – you should NOT do this! Only print out output that is needed for grading or answering the question, not intermediate steps. You will of course try many things and may print out things for yourself that are not needed, but just don't include that in your submission.

General You should try to think of this project as coming to really think about the data and showing what you've learned. Question 5, in particular, is looking for you to think outside the box about the data beyond the specific steps requested, and to evaluate whether this approach even makes sense or answers the bigger question of interest.