Documentation for divratio in Stata

Christopher V. Lau

January 6, 2023

Contents

1 Introduction

The divratio package was created to simplify and generalize the process of computing diversion ratios with individual-level data using the semiparametric method, described in Raval, Rosenbaum, and Tenn (2017).[1] This document describes how to use the package by detailing the accompanying template and the various options built into the command. While this package can be used to estimate diversion ratios in many different settings where individual-level data is accessible, it is frequently most useful for hospital merger investigations where discharge or claims data is regularly available. As a result, many references will be related back to a healthcare setting. For example, "store locations" will be referred to as hospitals, individuals will be referred to patients or discharges, etc.

While the package was written with the hopes of being flexible, some basic structure of the data is required to run the command. Specifically, each observation should represent an individual with a specified choice. For example, in the healthcare context, each observation should be a patient discharge from a specified hospital. The specified hospital represents the revealed preferences of the patient, and thus is considered as the choice variable. The

[1] Raval, Devesh, Ted Rosenbaum, and Steve Tenn. "Semiparametric Discrete Choice Model: An Application to Hospital Mergers." Economic Inquiry, 2017, 55 (4), 1919-1944

individual-level data should also specify the ownership of the store chosen, since that is what is being affected by a merger. In the hospital merger context, the healthcare system (owner) of the chosen hospital (store) should be specified as its own variable. The ownership variable should nest the store location variable. In other words, if one observation says Hospital A belongs to System 1, all other observations that have Hospital A as their choice should also have System 1 as the owner. This consistency is necessary for the command to run properly.

For users of the statistical programming language R, a companion package can be found on the CRAN repository, named "`healthcare.antitrust`".

2 The Template

The most straightforward way to implement diversion ratios is by using the template code `divratio_template.do`, which requires the analyst to fill in some case-specific information. The template was written with the intention of analyzing hospital mergers with discharge data, thus some additional modification may be required if trying to analyze a different setting. I will review the template to ensure the reader understands what each step of the code is doing.

2.1 Preliminaries

The first section is labeled as "Preliminaries." This section identifies the party systems, inputs the cleaned data, and creates variables that are useful for properly running the `divratio` command.

2.1.1 Block 1: Identifying the parties

The first block of code identifies the relevant hospital systems, and should read:

```
global a = "System 5" // Input a-side system
global b = "System 7" // Input b-side system
```

This block is where the analyst should identify the systems involved in the merger. Specifically, the analyst should replace "System 5" and "System 7" with the exact strings of the party system names in the data. For example, if the `current_system` variable identified System 5 as "System 5 Healthcare", the first line of the block should read: `global a = "System 5 Healthcare"`.

2.1.2 Block 2: Reading the data

The second block of code in the template inputs the data, and should read:

```
use "...\sim_data.dta", clear // Input discharge data
```

The data `sim_data.dta` is a simulated fictitious dataset generated in R that can be created from the script `create_sim_data.R` and mimics the basic structure of real hospital discharge data. The analyst should replace "`sim_data.dta`" with source of the individual-level data specific to the matter.

2.1.3 Block 3: Creating choice identifiers

The third and final block of code in the preliminaries section creates flags for all hospitals owned by the party systems and creates labels for them. The code should read:

```
foreach x in "a" "b" {
     * Identify system observations *
     gen ‘x’_side0 = (strpos(system,"$‘x’") > 0)
     label var ‘x’_side0 "$‘x’"

     * Create hospital flags *
     display "{ul: $‘x’}"
     tab hospital if ‘x’_side0 == 1
     global num_‘x’ = r(r)
     quietly levelsof hospital if ‘x’_side0 == 1, local(hosp)
     local j = 1
     foreach i of local hosp {
          gen ‘x’_side‘j’ = (hospital == "‘i’")
          label var ‘x’_side‘j’ "‘i’’"
          local j = ‘j’ + 1
          }
     }
```

If the A-side system owns N hospitals and the B-side system owns M hospitals, the above code should create $N+M+2$ variables. It should create two variables, a_side0 and b_side0, that represents flags for observations that visited any A-side or B-side hospital, respectively. In addition, it should create flags for whether the observation visited any of the individual hospitals owned by either side, a_side1,...,a_sideN and b_side1,...,b_sideM. In addition, this block should ensure that the system or hospital's name as identified in current_system and hospital is the variable label for its corresponding flag variable.

2.2 Semiparametric Analysis

The next section provides examples of the type of analyses that can be done with the divratio command; the specifics of the command will be discussed next.

3 The divratio Command

The divratio command calculates diversion ratios using a semiparametric discrete choice model, outlined in Raval, Rosenbaum, and Tenn (2017). In short, the procedure partitions the individual-level data based on patient characteristics. These partitions are commonly referred to as bins. The model assumes that the discharge share of hospitals within the bin represents the choice probability of patients who have those characteristics. If it is assumed that these choice probabilities take a logit functional form, it follows that diversions will be share-based. In other words, the probability that a patient diverts to Hospital B when Hospital A is no longer available will be equal to the share of patients who visited Hospital

B conditional on not visiting Hospital A. Aggregating across all patients will provide an estimate of the number of patients who divert to each hospital when another hospital is no longer available.

One fundamental issue with the above description is that the data may not be rich enough to provide reliable estimates for each bin. In particular, there may not be enough observations within each bin for the share of hospitals to be an accurate representation of choice probabilities. In this case, the semiparametric model re-partitions the data based on the same patient characteristics less the least important characteristic and repeats the binning analysis.

The re-partitioning can be done in two ways: without replacement and with replacement. Without replacement will re-partition based on the observations in bins that were too small to previously assign choice probabilities. With replacement will re-partition based on the entire sample, whether an observation was previously assigned a choice probability or not. The decision to re-partition without or with replacement involves a variance-bias tradeoff the analyst will have to make: Without replacement will have less bias, but will have greater variance. Resampling without replacement was the method used in Raval, Rosenbaum, and Tenn (2017), while resampling with replacement was used in the companion paper Raval, Rosenbaum, and Wilson (2021).[2]

3.1 The General Procedure

In this section, I describe at a high level the general procedure underlying the code to estimate the semiparametric diversions. This is not meant to be a detailed description of the semiaparametric method, but rather a reference for users to know what exactly is going on under the hood of the code.

3.1.1 Step 1: Establish Relevant Geographies

The first step is to establish the relevant geographies in which to form the estimation sample. Establishing a relevant geography is basically identifying the geographic service areas of the merging parties. In hospital merger analysis, this will typically reflect the patient zip codes that the parties are most likely to draw from. Note that this is *not the relevant geographic market* as defined by the Horizontal Merger Guidelines, but typically a broader region that is independent of the relevant geographic market. Generally, this step is used to identify and omit outlier observations based on geographic location.

The general procedure is as follows:

1. Calculate the percent of patients of interest (as described below) within each zip code.

2. Order the zip codes in descending order by size of the percentage previously calculated.

3. Calculate corresponding cumulative shares.

4. Keep all zip codes that have a cumulative share below a specified service area percentage.

[2] Raval, Devesh, Ted Rosenbaum, and Nathan Wilson. “Using Disaster Induced Closures to Evaluate Discrete Choice Models of Hospital Demand.” RAND Journal of Economics, 2022, 53 (3), 561-589.

There are four versions of this procedure based on the patients of interest: (1) combined, (2) A-side, (3) B-side, or (4) the union of the A-side and B-side. **Combined** considers the patients who either went to an A-side hospital or B-side hospital. **A-side** considers the patients who just went to an A-side hospital. **B-side** considers the patients who just went to an B-side hospital. **Union** separately calculates the A-side and B-side service areas and keeps the zip codes that either side has served.

The output of this step is a list of zip codes that represent the geography in which patients are to be considered in the estimation sample.

3.1.2 Step 2: Identify Relevant Products

The second step is to establish the relevant products for the estimation sample. Unlike the relevant geographies, this will generally reflect the relevant product market in hospital cases. For example, the relevant product market may be a cluster market of services that are not directly substitutable, but have similar competitive conditions. As a result, those services will need to be identified a priori. Diagnosis related group (DRG) codes can be used to represent different products.

There are four ways to consider identifying the relevant products: (1) overlap, (2) A-side, (3) B-side, and (4) the union of the A-side and B-side. **Overlap** represents all products (DRG) that are in common with both the A-side and B-side. **A-side** represents all products that are offered by the A-side. **B-side** represents all products that are offered by the B-side. **Union** separately finds the products offered by the A-side and B-side and keeps all products that either hospital offers.

The output of this step is a list of DRGs that represent the products to be considered in the cluster market for the estimation sample.

3.1.3 Step 3: Identifying the Choice Set

The third step identifies which alternatives are considered as possible choices in the model. The general procedure is as follows:

1. Create the full estimation sample. This involves restricting the data to individuals who live within the relevant geographies and who purchased the relevant product. In hospital cases, this means that every observation is a patient who lives within the zip codes produced in Step 1 and who was admitted to a hospital under a DRG produced in Step 2.

2. With the full sample, calculate the share of each hospital.

3. Rename any hospital that does not belong to either party and has a share below a specified cutoff "Outside." That is, hospitals that patients within the relevant geography visit that do not have significant share are relegated to the outside option.

The output of this step is a list of alternatives (hospitals) that represent the patient's choice set.

3.1.4 Step 4: Estimate Choice Probabilities Though Binning Procedure

The binning procedure begins by first converting the data into an individual-choice panel. This is done by first flagging the individual's observed choice (e.g., by appending "`_chosen`" onto the the relevant variable names), then crossing the individual-level data with the choice set produced in Step 3. With this panel dataset, the following algorithm is used to estimate patient choice probabilities:

Binning algorithm: Suppose there are N individuals in the data, indexed by i, and each individual has K choices, indexed by k, such that the total number of observations in the individual-choice panel is $N \times K$. Let G be the total number of characteristics required to bin over and set $g = G$. Let $\mathbf{V}_g = (v_1, ..., v_g)$ represent the grouping variables in a pre-specified order. Start with the individual-choice panel data and initialize individual choice probabilities p_i as an empty variable.

1. If sampling without replacement, drop observations in which choice probabilities p_{ik} are not missing. Create a group ID based on grouping variables $\mathbf{V}_g$. Collapse the data by choice and group ID.

2. Count the number of individuals within each grouping. If the number of individuals is less than the pre-specified threshold, drop the group.

3. For the remaining groups, calculate the observed share of each choice by group ID.

4. Merge back in individual-choice panel. Set choice probabilities p_i equal to the group share calculated in the previous step if merged. Otherwise, keep as missing.

5. If $g > 0$, save the individual-choice data with updated choice probabilities, set $g = g-1$, and return to Step 1. Otherwise, drop all ungrouped observations and stop.

The output of this step should be an individual-choice panel that includes choice probabilities. In practice, the program actually does something slightly different than the algorithm outlined above. In particular, it continually appends the group shares for each iteration onto the individual-choice panel and then does the assignment to choice probabilities at the end. This becomes practical in case the analyst wants a closer inspection of what the group shares look like at each iteration.

3.1.5 Step 5: Calculating Diversion Ratios

With the estimated choice probabilities, the last step estimates the diversion ratios from each of the specified party hospital to all hospitals in the choice set. The program:

1. Calculates the choice probability for each hospital j if hospital k were excluded as:

$$\frac{p_{ij}}{1 - p_{ik}}.$$

2. Sum these choice probabilities across hospitals to obtain the expected number of discharges of hospital j after hospital k is excluded. Subtract the observed number of discharges at hospital j to obtain the number of patients who diverted to hospital j from hospital k.

3. Divide the number of patients who diverted to hospital j from hospital k by the number of original discharges from hospital k to obtain the diversion ratio form k to j.

The output of this step should result in the final product: diversion ratios from the reference hospital to all other hospitals in the determined choice set.

3.2 Syntax

The basic syntax of the `divratio` command is given by:

```
divratio a_side b_side loc_id firm_id geo prod [if], [options]
```

Each input is discussed, in turn:

- `a_side` represents the flag for the A-side hospital the analyst is interested in considering diversions from. These flags were created in *Block 4* of the template. For example, if we are interested in estimating where patients diverted to when all A-side hospitals were excluded, the input would be `a_side0`. On the other hand, if we were interested in estimating where patients would divert to when only the A-side's second hospital were excluded, the input would be `a_side2`.

- `b_side` represents an analogous input as `a_side`, but for the B-side hospitals. In particular, the code will compute two sets of diversions, one for when the A-side is the excluded hospital and another for when the B-side is the excluded hospital.

- `loc_id` represents a categorical variable that represents hospital ("store location") identifiers, such as the name of the hospital. This represents the patient's choice.

- `firm_id` represents a categorical variable that represents the hospital's system ("the owning firm") identifiers, such as the hospital system's name. As mentioned above, it should be the case that each hospital in `loc_id` belongs to only one system in `firm_id`.

- `geo` represents a categorical variable that represents geographic regions used to calculate service areas; for example, zip codes.

- `prod` represents a categorical variable that represents products offered by the hospitals to identify overlapping services; for example, DRGs.

3.3 Options

There are several options available for the `divratio` command that represent alternative choices the user can make when estimating the model. Each option is discussed, in turn:

- `groups(varlist)` specifies a list of patient characteristics used to create bins ordered by most important to least important, which should be determined by the analyst and will depend on case-specific facts. The code will partition patient characteristics based on this list, and each iteration will drop the right-most variable until no variables in the list are left. An example of a group characteristic specification is:

  ```
  groups(pat_county pat_zip mdc drg age female)
  ```

  In this case, the first iteration will create bins based on patient county, patient zip code, MDC, DRG, age, and gender. The next iteration will create bins based on patient county, patient zip code, MDC, and age. If this option is omitted, the command will calculate aggregate share-based diversions. It is often good practice to make variables that have finer levels of granularity less important, since they are more likely to create bins that have too few observations.

- `wtp` represents a flag such that when used, willingness-to-pay analysis will also be included in the results section of the output.

- `ungrouped(string)` specifies what the program should do with observations that are not grouped in the binning stages. That is, after the iterative procedure is completed, there may still be some characteristic categories that did not have enough observations to be considered large enough. This option decides what to do with those "ungrouped" observations; there are two potential inputs into this option:

  - `drop` (default) drops the ungrouped observations.
  - `keep` keeps the ungrouped observations and assigns their choice probability as aggregate shares.

- `weight(varlist)` specifies a quantity in which each observation may be weighed. For example, if an observation in the data represents customer counts, rather than individual customers, then the count variable should be included as the input to this option. If omitted, the model will assume each observation represents an individual.

- `allow_within` represents a flag such that when used, diversions are computed to allow patients to divert to other hospitals that are owned by the same system, as specified by `firm_id`. For example, if this option is used, diversions will be calculated from A-side Hospital 1 to A-side Hospital 2. If omitted, the model eliminates from patients' choice sets other hospitals owned by the same system when considering diversion. For example, patients that can no longer access A-side Hospital 1 are not allowed to chose A-side Hospital 2.

- `replacement` represents a flag such that when used, bin probabilities will be calculated by resampling the entire dataset for all customers that satisfy the grouping level. That is, sampling will be done with replacement at each iteration. If this option is omitted, the bin probabilities will be calculated using only customers that were not previously used to calculate bin probabilities at higher grouping levels. That is, sampling will be done without replacement at each iteration.

- `geo_ref(string)` specifies the reference hospital(s) in which to compute geographic service areas. There are four possible inputs:

    - `a-side` calculates diversions over the service area for the A-side hospital specified in input `a_side`.
    - `b-side` calculates diversions over the service area for the B-side hospital specified in input `b_side`.
    - `union` calculates diversions over the union of the service area of the A-side hospital and the service area of the B-side hospital.
    - `combined` (default) calculates diversions of the combined service area of the A-side and B-side hospitals together.

- `prod_ref(string)` specifies the reference hospital(s) in which to consider the relevant products to compute diversions. There are four possible inputs:

    - `a-side` calculates diversions for the products offered by the A-side hospital specified in input `a_side`.
    - `b-side` calculates diversions for the products offered by the B-side hospital specified in input `b_side`.
    - `union` calculates diversions for the products offered by either the A-side or the B-side.
    - `overlap` (default) calculates diversions for the products that are both offered by the A-side hospital and the B-side hospital.

- `svc_pct(real)` specifies the percentage of observations that should be used to calculate service areas. Inputs should be entered as percentages, e.g., 75, and not decimals, e.g., 0.75. If omitted, 75 percent service areas will be calculated.

- `min_group_size(integer)` specifies the minimum number of observations in which a bin can be considered “large” enough to be used to estimate choice probabilities. This is the “tuning parameter” in Raval, Rosenbaum, and Tenn (2017). The authors recommend that practitioners use cross validation in identifying a proper tuning parameter. While this is not built into the package, researchers may want to test the robustness of the minimum group size to ensure results are not sensitive to this parameter. If omitted, the default minimum number of observations per bin is set to 20.

- `outside_cutoff(real)` specifies what share percentage to use when considering which hospitals are considered as in the choice set. Hospitals with aggregate shares within the specified relevant geography below this cutoff will be designated as an "outside option". If omitted, a cutoff of 0.005% is used.

- `topcode(real)` - Provides a maximum value to choice probabilities for WTP calculations, since WTP under the logit formula is undefined if the choice probability is equal to 1. If omitted, default is set to 0.99.

- `save_inter(string)` is used to save the intermediate files that are created in the analysis. The input specifies a *file location*. Saved intermediate files include (under the example that the 90 percent service area was used, the `geo` input was zip codes, and the `prod` input was DRGs):

  - `RAW - A-side 90 Service Area Regions.dta` - Zip codes in the A-side's 90 percent service area.
  - `RAW - B-side 90 Service Area Regions.dta` - Zip codes in the B-side's 90 percent service area.
  - `RAW - Combined 90 Service Area Regions.dta` - Zip codes in the combined 90 percent service area.
  - `RAW - A-side 90 Service Area Sample.dta` - Sample that consists of just observations in the A-side's 90 percent service area.
  - `RAW - B-side 90 Service Area Sample.dta` - Sample that consists of just observations in the B-side's 90 percent service area.
  - `RAW - Combined 90 Service Area Sample.dta` - Sample that consists of just observations in the combined 90 percent service area.
  - `RAW - A-side Products.dta` - DRGs offered by the A-side in the combined 90 percent service area.
  - `RAW - B-side Products.dta` - DRGs offered by the B-side in the combined 90 percent service area.
  - `RAW - Overlapping Products.dta` - Overlapping DRGs offered by both parties in the combined 90 percent service area.
  - `RAW - 90 Service Area Overlapping Products Sample.dta` - Sample that consists of just observations diagnosed with overlapping DRGs in the combined 90 percent service area.
  - `RAW - Choice Set.dta` - List of hospitals in the choice set.
  - `RAW - Panel Estimation Sample.dta` - Panel data that includes all individuals in the relevant geography and choices in the relevant product.
  - `RAW - Probabilities by Group.dta` - Panel data that includes the bin shares for each iteration of the binning procedure.
  - `Results - Choice Probabilities.dta` - Panel data that includes the final choice probabilities from the binning algorithm. Represents the raw input for the diversion ratio calculations.

- ○ `Results - Diversions.dta` - Final results of diversion ratios from A-side and B-side to all hospitals in the choice set.
- ○ `Results - Willingness-to-pay.dta` - Final results for willingness-to-pay analysis.

If omitted, no intermediate files will be saved.

- `inter_input(string)` specifies a file location of the bin probabilities calculated from previous runs of the program. This can be used to save time when executing the command multiple times–the command skips the binning procedure and uses the choice probabilities from previous runs. If the `save_inter` option is used on previous runs, then the input file will by given by "`Results - Choice Probabilities.dta`".
- `output(string)` is used to save the final diversion results to a specified file location, including a *file name*. This can be used to specifically save the results without having to save the other intermediate files. The file name specified here will overwrite the results file from the `save_inter` option.
- `disp_str(integer)` specifies the number of characters for string variables to be displayed in the output results. If omitted, the results will display 30 characters.

3.4 Output

To detail the output, an example with simulated discharge data is used. The following commands were executed to produce the output below:

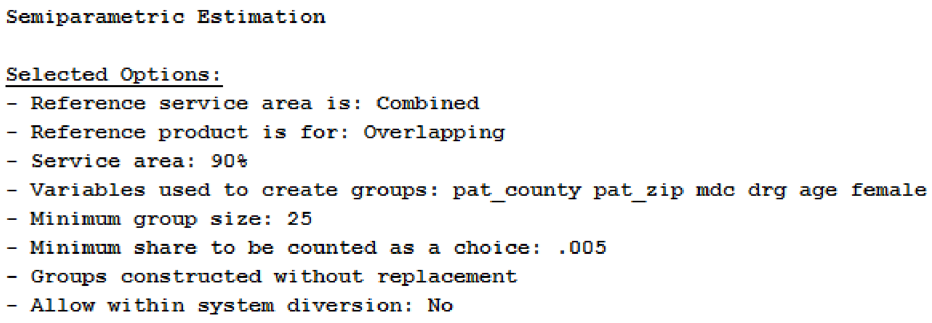
```
global bins "pat_county pat_zip mdc drg age female"
divratio a_side0 b_side0 hospital system pat_zip drg, ///
     groups($bins) geo_ref(combined) prod_ref(overlap) svc_pct(90) ///
     min_group_size(25)
```

The first line creates a global for the specified patient characteristics that will be used for the binning procedure, ordered from most to least important in this example. The remaining lines implement the `divratio` package, specifying that system-level diversions be calculated (by inputting `a_side0` and `b_side0`).

The output displayed comes in three sections: (1) *Selected Options*, which reviews the options used to run the command, (2) *Intermediate Calculations*, which provides information and diagnostics of the intermediate steps, and (3) *Results*, which displays a table that includes diversions from the A-side and the B-side. Each section is discussed, in turn.

3.4.1 Selected Options

The *Selected Options* section exists to provide a review of the assumptions and options underlying the model run by the `divratio` command. Below is a sample printout of the *Selected Options* section.



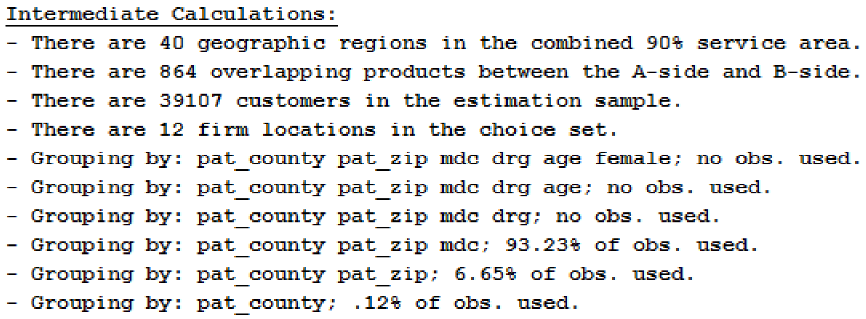
```
Semiparametric Estimation

Selected Options:
- Reference service area is: Combined
- Reference product is for: Overlapping
- Service area: 90%
- Variables used to create groups: pat_county pat_zip mdc drg age female
- Minimum group size: 25
- Minimum share to be counted as a choice: .005
- Groups constructed without replacement
- Allow within system diversion: No
```

The first through third bullets review assumptions about the relevant geographic regions and products; in this case, the relevant geographic region is the combined 90 percent service area and the relevant products to analyze overlapping DRGs. The fourth bullet displays the (ordered) variables used for the binning procedure. The next two bullets report the cutoffs for establishing appropriate bin sizes and outside options. Finally, the seventh bullet reports whether the binning procedure was done with or without replacement and the last reports whether within-system diversions are allowed.

3.4.2 Intermediate Calculations

The *Intermediate Calculations* section exists to provide some additional information that is useful in evaluating how well the semiparametric estimation is performing. Below is a sample printout of the *Intermediate Calculations* section.



```
Intermediate Calculations:
- There are 40 geographic regions in the combined 90% service area.
- There are 864 overlapping products between the A-side and B-side.
- There are 39107 customers in the estimation sample.
- There are 12 firm locations in the choice set.
- Grouping by: pat_county pat_zip mdc drg age female; no obs. used.
- Grouping by: pat_county pat_zip mdc drg age; no obs. used.
- Grouping by: pat_county pat_zip mdc drg; no obs. used.
- Grouping by: pat_county pat_zip mdc; 93.23% of obs. used.
- Grouping by: pat_county pat_zip; 6.65% of obs. used.
- Grouping by: pat_county; .12% of obs. used.
```

I walk through each intermediate calculation:

- The first reports the number of geographic regions in the relevant service area, specified by `geo` and `geo_ref`. In this case, the calculation reports that there are 40 zip codes in the combined 90 percent service area.
- The second reports the number of relevant products that are used in the analysis, specified by `prod` and `prod_ref`. In this case, the calculation reports that are 864 overlapping DRGs.
- The third reports the number of customers that are used in the estimation sample after restricting to the relevant geographic area and relevant products. In this case, there

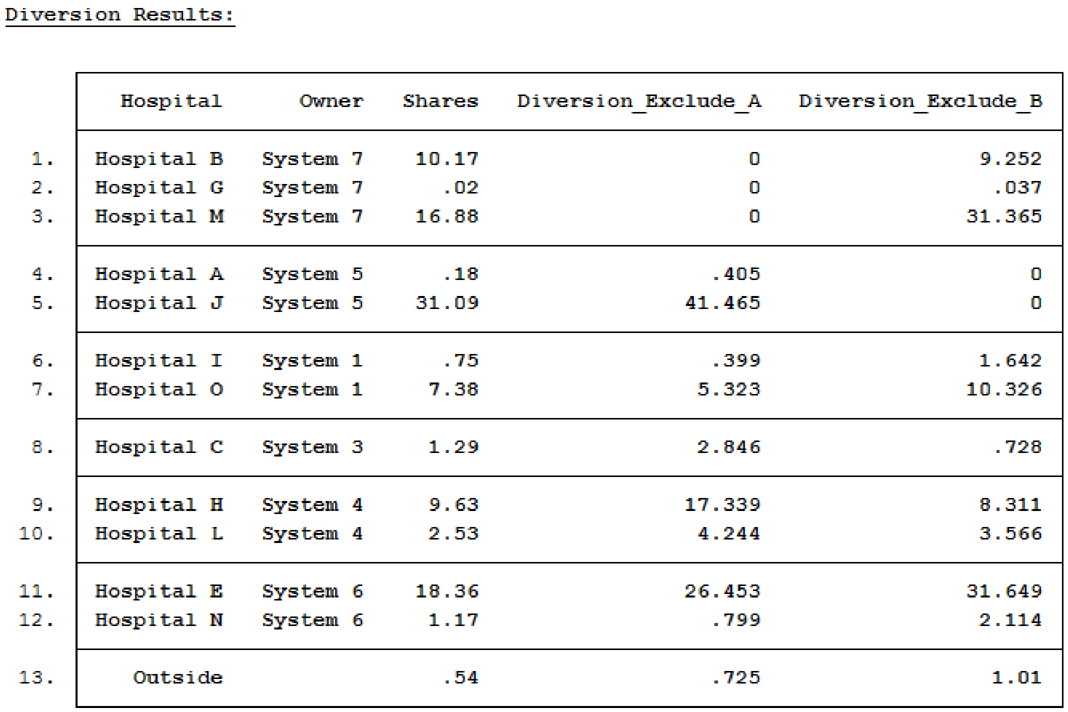
are 39,107 patients who were diagnosed with one of the overlapping DRGs and live in one of the zip codes in the combined 90 percent service area.

- The fourth reports the number of firms that qualify as an alternative in the choice model, which includes all firms that have an aggregate share above the percentage specified in `outside_cutoff`. In this case, there are 12 hospitals in the choice model.

- The remaining calculations provide progress updates on what iterations of the binning method have been completed. In addition, each line presents the percent of observations that were grouped using the specified variables. In this case, the first three iterations were skipped because no observations were used. In other words, partitions based on all of the specified characteristics in `groups` cuts the data too fine, such that there are no bins of at least the size specified in `min_group_size`. The fourth iteration is the first in which observations are used: 93.23% of observations are in bins that are used in the semiparametric model. This implies that the estimates are not able to control for characteristics as fine as DRG, but can control for MDC (as well as county and zip code). These intermediate calculations are useful for providing diagnostics on what characteristics the model is and is not actually using.

3.4.3 Results

The *Results* section provides the diversion ratio estimates from the semiparametric model. Below is a sample printout of the *Results* section.



```
Diversion Results:

     +--------------------------------------------------------------------------+
     |   Hospital      Owner   Shares   Diversion_Exclude_A   Diversion_Exclude_B |
     |--------------------------------------------------------------------------|
  1. | Hospital B   System 7    10.17                     0                 9.252 |
  2. | Hospital G   System 7      .02                     0                  .037 |
  3. | Hospital M   System 7    16.88                     0                31.365 |
     |--------------------------------------------------------------------------|
  4. | Hospital A   System 5      .18                  .405                     0 |
  5. | Hospital J   System 5    31.09                41.465                     0 |
     |--------------------------------------------------------------------------|
  6. | Hospital I   System 1      .75                  .399                 1.642 |
  7. | Hospital O   System 1     7.38                 5.323                10.326 |
     |--------------------------------------------------------------------------|
  8. | Hospital C   System 3     1.29                 2.846                  .728 |
     |--------------------------------------------------------------------------|
  9. | Hospital H   System 4     9.63                17.339                 8.311 |
 10. | Hospital L   System 4     2.53                 4.244                 3.566 |
     |--------------------------------------------------------------------------|
 11. | Hospital E   System 6    18.36                26.453                31.649 |
 12. | Hospital N   System 6     1.17                  .799                 2.114 |
     |--------------------------------------------------------------------------|
 13. |    Outside                 .54                  .725                  1.01 |
     +--------------------------------------------------------------------------+
```

The first column represents the hospital that is being diverted to, while the second specifies the corresponding system. The third column presents the share of observations in the estimation sample that chose each hospital. Note that these are **not** market shares because the estimation sample is not the same as the relevant geographic market. The fourth column represents the diversion ratio estimates from the specified A-side hospital in `a_side`, while the fifth column represents the diversion ratio estimates from the specified B-side hospital in `b_side`.

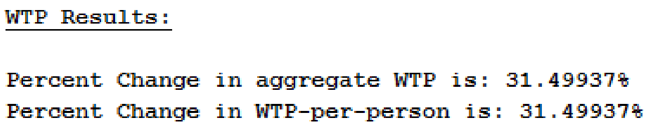
The diversion table produced can be read as follows: Take the fifth row and fourth column that reads "41.466". This means that 41.467% of patients would go to Hospital J if all of the A-side hospitals (which was specified as System 5 in the template) were no longer available. Likewise, the third row and fifth column suggests that 31.365% of patients would go to Hospital M if all of the B-side hospitals (specified as System 7) were no longer available.

The results will always be organized such that the A-side hospitals specified in `a_side` are at the top, the B-side hospitals specified in `b_side` are second, and the remaining systems are organized in alphabetical order. The hospitals within the system will always be sorted in alphabetical order.

3.4.4 The `wtp` Option

When the `wtp` option is used, the `divratio` command also calculates the change in willingness-to-pay and presents the output after the diversion ratio results. The `wtp` option will always group together hospitals that are in the same system as defined by the input `firm_id`. As a result, estimates of WTP will be the same for whichever `a_side` and `b_side` flags are used (e.g., inputs `a_side0` and `b_side0` will produce the same WTP results as inputs `a_side1` and `b_side2`). In addition, the `allow_within` option will not apply to the WTP results, since WTP is fundamentally a system-based analysis, thus allowing within-system diversions does not apply.

I review the results of the `wtp` command. Below is a printout of the output.

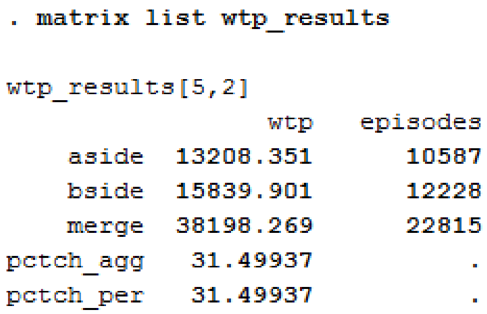


```
WTP Results:

Percent Change in aggregate WTP is: 31.49937%
Percent Change in WTP-per-person is: 31.49937%
```

The output displays the percent change in aggregate WTP and percent change in in WTP-per-person (discharge), which are both 31.5 percent. In general, these two percentages do not need to be the same. In particular, they will be different in mergers where some subset of the parties' hospitals form the post-merger system, such as in the formation of a joint venture.

The option also saves more detailed results as a 5×2 matrix named `wtp_results`. Below is a printout of the matrix.



```
. matrix list wtp_results

wtp_results[5,2]
                  wtp   episodes
    aside   13208.351      10587
    bside   15839.901      12228
    merge   38198.269      22815
pctch_agg    31.49937          .
pctch_per    31.49937          .
```

The first column presents the estimates of WTP, while the second presents the corresponding individuals associated with the entities specified by the rows. The first row displays the aggregate WTP and episodes (number of observations) for the A-side system–dividing the two columns provides an estimate of WTP-per-person. The second row displays the analogous estimates for the B-side system. The third row presents the aggregate WTP and episodes of the post-merger entity. The last two rows reproduce the estimated percent change in WTP and WTP-per-person.