Doing Democratic Data Analysis

Corban Nemeth

2020-04-27

Contents

Preface

I believe that data, *in the hands of public administrators and policy analysts*[1], has the power to transform the way government works.

Big questions will, and should, be asked of big data— the role of government in regulating algorithmic bias, facial recognition, and consumer data privacy is a vital conversation. However, these topics should not detract or deter public administrators and policy analysts from leaning into **small data** for decision-making purposes.

Public administrators and analysts who are data literate will be able to make and inform better decisions while avoiding the pitfalls posed by the latest technological trends. This book represents an opportunity for public administrators and policy analysts to join their subject matter expertise with foundation principles and practices of democratic data analysis— data analysis that is **transparent, relevant, and grounded in the context of ethical and effective governance**.

Who is this guide for?

This guide is for:

- the budget analyst at the Department of Fish and Wildlife who has to compile a monthly report analyzing revenues,
- the manager at the Department of Social and Health Services who is tracking inventory, and
- the research analyst working for the state Legislature who wants to incorporate data into her work session on the latest policy debate.

[1] *Not IT departments*

What will I learn?

You will learn an opinionated framework for data analysis in public sector organizations. By opinionated, I mean that I will teach you what I think is the right way to do things given my own experience as a public sector policy and data analyst. Your experience might differ– and that's great. I hope that where you can use your experience in place of mine, you do to the fullest extent. With that in mind, it is often said that you have to know the rules to break them, so I will teach you the "rules" as I understand them.

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Chapter 1

Introduction

1.1 Principles and Practices

This handbook is composed of five principles of democratic data analysis. Democratic data analysis is: * Tidy * Reproducible * Honest * Visual, and; * Decision-oriented.

Each of those principles has a seperate chapter in this book. As this isn't a how-to manual, per say, each chapter will begin with a description of the *principles* outlined above and arguments for why they are important. Each chapter will then have a *practices* section where I walk through examples of how to implement these principles in common situations. Chapters will conclude with a link to other resources for additonal information.

Why seperate principles and practices? The nature of data analysis is heavily influenced by the technologies that we have access to. Whether it be the venerable pivot table, or a new-school dashboard platform, or a data-oriented programming language, the principles that I lay out here supersede specific technologies. Think of it like grammer. You may write by hand, on a computer, using text-to-speech. You may be writing a poem, a novel, an argument, or an instruction manual. But the basic rules of grammer are relevant in whatever medium you choose. Similarily, this guide will teach you the basic "grammar" of democratic data analysis. This will alllow you to apply this knowledge in whatever platform or technology you are interested in or have access to. But similar to learning language, it helps to practice. That's where the *practices* section comes in to play.

This guide will include examples in both Excel and `R`. Government runs on Excel, so all of the examples and exercises will be Excel compatible. If you are comfortable with Excel[1] and want to challenge yourself, boost your resume, and

[1] aka you use `vlookup`, `index(match)`, pivot tables, or *Get & Transform* on a somewhat

become a data wizard[2], I would highly recommend learning `R`. This guide will wshow examples on how to analyze data democratically using both tools, but focusing on the principles and practices that are vital regardless of technology. I will link to specific resources that provide more detailed walk through's as necessary.

1.2 How to Think about Democratic Data Analysis

What is data analysis? It may be easier to start with what data analysis isn't.

1. Data analysis isn't math.

- Calculations are great, but `a7 + b8` in Excel is deterministic. It gives you one answer. This book is not interested in data analysis that gives you the right answer, because there is no such thing. There are many answers to many questions, depending on how those questions are asked and how the data is analyzed.

2. Data analysis isn't statistics.

- This book is about reading and telling the story of your data in a way that can complement expertise and experience to make better decisions. Statistics are often used as a cheap stand-in for domain expertise and are often abused in favor of trusting the analyst or administrator to back up their assumptions with both quantitative and qualitiative data.

3. Data analysis isn't research methods.

- No set of tools and practices can stand in for asking the right questions, and transforming data into information to answer that question. This book will give you the tools to work with your quantitative data to answer relevant questions, but all good analysis begins with a good question.

1.3 Principles of Democratic Data Analysis

- Think in terms of fields, not values
- Leave breadcrumbs for others (and your future self)

regular basis

[2]For example, I used R to create this entire website

- Create a data pipeline and DO NOT DESTROY UNDERLYING DATA– build
- Make assumptions, and document them!
- Show AND tell your results

1.4 The Language of Data Analysis[3]

As I mentioned before, democratic data analysis has an underlying structure, like grammar. There are rules so these sentances (hopefully) make sense to you, the reader. Similarily, by following common conventions of tidy data analysis, others will be able to "read" your analysis like you are reading this sentance. And also, like grammar, you can break the rules– but it helps to know them first.

Here are a couple definitions that will help as you move through this text. Don't worry about memorizing them, as I will refer back to these definitions frequently.

- Fields
  - A field is a fancy name for a column. From here on out, every calculation, manipulation, formula, you name it, will be on a column. I want you to forget that you could ever modify a lone cell in Excel. No more formulas in cells. No more typing in values to a cell. Certainly no more writing over data in a cell. Democratic data analysis depends on formulas that work on entire fields. Everything you would need to do to a single cell in Excel can– and should!– be done to an entire column. This will be immensly valuable, as you will hopefully see while working through this material.
- Variables
  - A variable is something in your data that can change. That's it! Variables become very important when looking at how to structure your data.
- Observations
  - Observations make up the rows of your dataset. Each observation should correspond to a specific "thing." This will make more sense later, I promise.
- Values
  - Values are the actual data in your table. Each value belongs to 1 (one) observation and 1 (one) variable.
- Table

[3]Adapted from Hadley Wickham's paper on Tidy Data

- A table is the grouping of all observations of a similar type.

You may already be able to see how these definitions foreshadow some of what is coming in later sections. For example, there are no references to cells. This is intentional. The most importatant distinction between democratic data analysis and simply working in excel is that in democratic data analysis, (virtually) *everything* is done on the `field` level. Changes are made to entire columns, calculations are made on entire columns. Thinking in `fields` is the first step on the path to democratic data enlightement. Having data formatted in the strucutre outlined above forces good data hygiene that will pay massive dividends later on.

Chapter 2

Tidy Data

2.1 Cleaning vs Tidying

My wife gives me a hard time because I hate cleaning, but love tidying. Similar things could be said about my mentality when it comes to data cleaning versus data tidying. Unfortunately, as in with life, one must clean before one tidies. But let's start with some conceptual definitions.

Cleaning refers to the process of scrubbing the data into a way that makes sense to you, the analyst. Oftentimes, and especially in public sector organizations, the data is not clean. Whether you are looking at the output of a SurveyMonkey survey or a canned report that is run from the IT department, your data will come in all shapes and sizes.

Here is the first major departure from what you may have been taught about data analysis in Excel. When you get messy data *do not* change individual cell values (if you can at all help it). Recall from the introductory chapter the difference between cells and fields. Fields, as a reminder, are columns that represent one variable. Whenever possible, use data analysis tools to make changes to the entire field, rather than specific cells. Most data analysis software, outside of Excel, make it difficult or impossible to change individual cell values. This is important for several reasons, most of which we will get to in the next chapter on reproducibility. But for now, thinking in terms of fields, and making changes to entire fields, will save you *a lot* of work and headache in the long run. Let's look at a sample dataset that may be similar to one you would encounter in real life. Here is a survey collected by a field manager of a local parks and recreation department on employment.

```
library(tidyverse)

sites <- tribble(
```

```
  ~"Employee", ~"Location", ~"Telecommute?", ~"Hire Date",
  "ron swanson", "Pawnee City Hall", "never", "Unknown",
  "Knope, Leslie", "Field Duty", "1 day/week", "2011-6-1",
  "Andy Dwyer", "sullivan street pit", "40 hours", "March 1, 2013",
  "Jerry Gergich", "City Hall", "never", "6/1/1985",
  "Garry Gergich", "City Hall", "never", "6/1/1985",
  "ben wyatt", "Partridge, Minnesota","" , "Jan. 1, 2010"
)

sites %>% datatable(
    extensions = 'Buttons',
    options = list(dom = 'Bfrtip',
                   buttons = 'excel',
                   searching = FALSE))
```

In this example, it would be trivial to go in to the Excel file and clean up the dates, names, and locations by hand. However, you could imagine this survey replicated for a department of forty employees. It quickly becomes unfeasable to make those edits by hand. When this is the case, there are functions in Excel and R that will make your life much easier.

Here is annotated code for how I would go about cleaning this table in R. The friendly syntax of the `tidyverse` packages makes it easy to follow along, even if you aren't comfortable writing it yourself.

```
sites_cleaned <- sites %>% #creating a new table called "sites_cleaned", starting with
  mutate(Employee = if_else(Employee == "Knope, Leslie", "Leslie Knope", Employee)) %>%
  separate(Employee, into = c("first_name", "last_name")) %>%
  rename(location = Location,
         telecommute_hours =`Telecommute?`,
         hire_date = `Hire Date`) %>%
  mutate(first_name = str_to_title(first_name),
         last_name = str_to_title(last_name),
         location = str_to_title(location)) %>%
  mutate(location = case_when(
           str_detect(location, "City Hall") ~ "In Office",
           str_detect(location, "Field") ~ "In Field",
           str_detect(location, "Street") ~ "In Field",
           TRUE ~ "Other"),
         telecommute_hours = case_when(
           telecommute_hours == "never" ~ 0,
           telecommute_hours == "1 day/week" ~ 8,
           telecommute_hours == "40 hours" ~ 40
         )
         )
```

Table 2.1: Visits to the Dept. of Retirement Services in a given month

Employee	Phone Visits	Office Visits	Online Visits
Danielle	6	11	23
Ramona	11	5	18
Ross	10	10	10

2.2 Thinking in Pivot Tables– From Wide to Long.

Pivot tables are amazing. THey are the world's most common, most helpful, and most underrated data analysis tool. PowerBI interactive charts and graphs are just pivot tables in disguise. Understanding what is needed to make a pivot table work is the key to the wide world of data analysis.

A pivot table groups data by field and allows the user to drag fields to the rows or columns of the pivot table. This is effective when each field is a variable (something that can change), and each row is a seperate observation of some phenomena of interest.

In short, pivot tables depend on **tidy data**.

Tidy data is the way your data should be organized before you begin your analysis. In tidy data, each column is a *variable*, each row is an *observation*, and each table is an *associated set of observations*. What does that mean in practice? Consider the following example.

Below is a table[1] that shows types of retirement visits for a month at a state's Department of Retirement Services by the employee who took the visit and the visit type.

```
visits <- tribble(
  ~"Employee", ~"Phone Visits", ~"Office Visits", ~"Online Visits",
  "Danielle", 6, 11, 23,
  "Ramona", 11, 5, 18,
  "Ross", 10, 10, 10
)

knitr::kable(visits, caption = "Visits to the Dept. of Retirement Services in a given month")
```

Data are frequently displayed in this "wide" format. It works great for presentation, but not great for data analysis.

The shortcomings of data in this format may become apparent when you attempt to work with the data in a pivot table. This is becuase our columns aren't

[1] Data was created for demonstration purposes

Table 2.2: Visits to the Dept. of Retirement Services in a given month

Employee	Visit Type	Number of Visits
Danielle	Phone Visits	6
Danielle	Office Visits	11
Danielle	Online Visits	23
Ramona	Phone Visits	11
Ramona	Office Visits	5
Ramona	Online Visits	18
Ross	Phone Visits	10
Ross	Office Visits	10
Ross	Online Visits	10

truly variables. You can drag the fields from the top row to the grey box below, for columns, and the left, for rows. This becomes unmanegable quickly.

```
rpivotTable::rpivotTable(visits, width = "60%", height = "60%")
```

Let's apply our criteria of tidy data to this set:

- Variables
  - At first glance, it doensn't look like this is a problem. But think again. Is `phone visits` really a variable? Or is the real variable of interest number of visits? And are our column names actually variables too (type of visit)?

Let's take another swing at setting up our table for data analysis purposes. This can be accomplished easily in R using the code below, or in Excel by loading the data with `Get and Transform` -> selecting the three "visits" columns -> right clicking -> and selecting "unpivot columns."

```
#We have already loaded the "tidyverse" library so we do not have to do it again

pivot_visits <- visits %>% #we are editing the "visits" table already created by stori
  pivot_longer(-Employee, names_to = "Visit Type", values_to = "Number of Visits") #us


knitr::kable(pivot_visits, caption = "Visits to the Dept. of Retirement Services in a
```

Now this is a table that is much easier to analyze in an Excel pivot table or with a variety of R functions. Using data in this format, it is easy to recreate the original table for presentation, while also giving a variety of options for

Table 2.3: Visits to the Dept. of Retirement Services in a given month by employee and associated client retirements

Employee	Phone Visits	Phone Retirements	Office Visits	Office Retirements	Online Visits	Online Retire
Danielle	6	4	11	8	23	
Ramona	11	7	5	3	18	
Ross	10	8	10	7	10	

formatting and plotting. Use the pivot table below to recreate the original table using the tidy data. *Hint- Instead of Count, select Sum -> Number of Visits as the value field. It is far easier to work with fields when they are in a tidy format.

```
rpivotTable::rpivotTable(pivot_visits, width = "60%", height = "400px")
```

When we get to the next chapter, you will learn several alternatives to pivot tables that use the same principles, but are more reproducible.

2.3 Using lower level data

Let's introduce a slightly more complicated tidy data problem, using the same base data as before.

```
visits_retirements <- tribble(
  ~"Employee", ~"Phone Visits", ~"Phone Retirements", ~"Office Visits", ~"Office Retirements", ~"
  "Danielle", 6, 4, 11, 8, 23, 15,
  "Ramona", 11, 7, 5, 3, 18, 15,
  "Ross", 10, 8, 10, 7, 10, 9
)

knitr::kable(visits_retirements, caption = "Visits to the Dept. of Retirement Services in a given
```

Hopefully you will see a similar pattern here. Now, there are three variables: Visit type, number of visits, and number of retirements. Again, this data works fine for presentation but could use tidying to ease in analysis.

```
visits_retirements %>%
  DT::datatable(
    extensions = 'Buttons',
    options = list(dom = 'Bfrtip',
                   buttons = 'excel',
                   searching = FALSE))
```

Try to tidy this in R or Excel Get and Transform. See this footnote[2] or look at the code if you need a hint.

```
visits_retirements_tidy <- visits_retirements %>%
  pivot_longer(cols = -Employee,
               names_to = c("Visit Location", "Type"),
               names_sep = " ")

print(visits_retirements_tidy)
```

```
## # A tibble: 18 x 4
##    Employee `Visit Location` Type        value
##    <chr>    <chr>            <chr>       <dbl>
##  1 Danielle Phone            Visits          6
##  2 Danielle Phone            Retirements     4
##  3 Danielle Office           Visits         11
##  4 Danielle Office           Retirements     8
##  5 Danielle Online           Visits         23
##  6 Danielle Online           Retirements    15
##  7 Ramona   Phone            Visits         11
##  8 Ramona   Phone            Retirements     7
##  9 Ramona   Office           Visits          5
## 10 Ramona   Office           Retirements     3
## 11 Ramona   Online           Visits         18
## 12 Ramona   Online           Retirements    15
## 13 Ross     Phone            Visits         10
## 14 Ross     Phone            Retirements     8
## 15 Ross     Office           Visits         10
## 16 Ross     Office           Retirements     7
## 17 Ross     Online           Visits         10
## 18 Ross     Online           Retirements     9
```

In this case, we actually pivoted too far. It will probably be more useful to have the counts of visits and retirements in their own category. Keep in mind the scope of the observation– It is perfectly valid for each to have their own column, as it is visits and retirements per month.

```
visits_retirements_tidy2 <- visits_retirements_tidy %>%
  pivot_wider(id_cols = c(Employee, `Visit Location`, Type), names_from = Type, values_

print(visits_retirements_tidy2)
```

```
## # A tibble: 9 x 4
```

[2] powerquery hints

```
##   Employee `Visit Location` Visits Retirements
##   <chr>    <chr>              <dbl>       <dbl>
## 1 Danielle Phone                  6           4
## 2 Danielle Office                11           8
## 3 Danielle Online                23          15
## 4 Ramona   Phone                 11           7
## 5 Ramona   Office                 5           3
## 6 Ramona   Online                18          15
## 7 Ross     Phone                 10           8
## 8 Ross     Office                10           7
## 9 Ross     Online                10           9
```

From here, it is easy to do calculations based on fields, rather than cells. For example, in R or Get and Transform, you could add the following:

```
visits_pct <- visits_retirements_tidy2 %>%
  mutate(pct_retirements = Retirements / Visits)

print(visits_pct)
```

```
## # A tibble: 9 x 5
##   Employee `Visit Location` Visits Retirements pct_retirements
##   <chr>    <chr>              <dbl>       <dbl>           <dbl>
## 1 Danielle Phone                  6           4           0.667
## 2 Danielle Office                11           8           0.727
## 3 Danielle Online                23          15           0.652
## 4 Ramona   Phone                 11           7           0.636
## 5 Ramona   Office                 5           3           0.6
## 6 Ramona   Online                18          15           0.833
## 7 Ross     Phone                 10           8           0.8
## 8 Ross     Office                10           7           0.7
## 9 Ross     Online                10           9           0.9
```

And then, one of the most useful things you can do is develop formulas by grouping of rows. For example, you may want to know the total number of visits and retirements by retiree, regardless of visit location. That can be accomplished in a pivot table.

2.4 So how is this democratic?

2.5 Practice problems

Chapter 3

Reproducible Analysis

3.1 Principles to Make Your Life Easier

Many things take more time to do up front, but save you from massive headaches down the road. Brushing your teeth. Oil changes. Preventative maintenence is the name of the game. The same thing applies in democratic data analysis. Learning how to brush the teeth of your analysis will pay massive dividends down the road, as someone else (or you, more likely), need to go back through and understand, replicate, or validate your findings.

The second principle of democratic data analysis is reproducability. By this, I mean anything that makes it easy to look at your analysis and understand what is going on. This is where classic data analysis in Excel falls short. I believe it is almost a universal experience in the public sector to recieve a workbook full of broken links, formulas pointing in every direction, and no sense of where the original data is or what has happened to it since.

In thinking about creating reproducible data analysis, it is important to keep in mind that data analasys should be structured from beginning to end, like a story. In the beginning, there is raw data that you pulled from a report, compiled yourself, or otherwise recieved. In Act 1, you use the practices we learned in the previous section to make the raw data tidy– without distroying the original data. You should use tools that allow to to non-destructively manipulate and iterate on your data. Both Get & Transform and R allow you to do this by default. In Act 2, which will be the next chapter, you use your data to create a picture of the world before you share it with others in the final Act 3.

The practices of reproducability that you will use here apply throughout the other chapters. It may seem like a waste of time, but if you have ever come back to a complicated excel workbook after spending even days away, this will make your life much easier.

3.1.1 Do Not Destroy

As I mentioned before, the existential dread that occurs when opening someone else's workbook and immediatly recieving broken links, color-coding[1], and a spiderweb of formulas may be a universal experience in the public sector. But there is a better way to do things. Reproducible analysis is linear. It progresses in a certain direction– from data load to final analysis. Things happen discretely. The blessing and curse of spreadsheets is that they are unbooud by time. There is no natural direction, just a sea of little boxes spreading out as far as the eye can see[2]. However, there are ways to impose a linear structure to your analysis.

The first thing I want to emphasize is PLEASE DO NOT DESTROY, ALTER, OR MANIPULATE YOUR UNDERLYING DATA. Your underlying data is like the foundation of your house.

3.1.2 Comment Everything

Comments are wonderful. They are notes to yourself that you should leave at almost every step of your analysis. I frequently do not leave comments. Never have I come back to an uncommented data transformation and been happy with my past self. At worst, leaving comments takes a couple seconds of your time you will never get back. At best, it saves you or your organization from a massive headache when you are able to catch your own errors or update your analysis easier in the future.

3.2 Practices

3.3 Comment Comment Comment

[1] for the love of democracy, PLEASE do not color code your data

[2] This is where programming languages such as R have an inherent advantage. Code runs in order, from first to last

Chapter 4

Data Modeling

4.1 Why Model?

Models transform data into decision making. ## Example two

4.2 Assumptions

Chapter 5

Visualization

5.1 Show and Tell

##Vizualizaiton is anything that presents your evidence– think critically about it!

Chapter 6

Applications

6.1 Tying it all together

Chapter 7

Resources