Doing Democratic Data Analysis

Corban Nemeth

2020-05-07

Contents

Preface

I believe that data, *in the hands of public administrators and policy analysts*, 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**.

In this guide, I present 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.

This is guide is not an excel how-to manual. My hope is that the principles and practices outlined here will allow you to explore whatever analysis tools you are interested in in a democratic manner. With that said, practical examples are given in Excel and in R.

Chapter 1

Introduction

> Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions and supporting decision-making.

- *Wikipedia (shame on me)*

1.1 Principles and Practices

In this handbook, I propose four principles of democratic data analysis. Democratic data analysis is:

- Tidy
- Pivot-able
- Reproducible,
- Uncertainty-oriented, and
- Audience-focused

As this isn't a how-to manual, each chapter will begin with a description of the *principles* outlined above and arguments for why they are important. This will be followed by *practices* section where I walk through examples of how to implement these principles in common situations and provide additional materials for you to learn how to apply these principles using common data analysis tools.

Why maintain the distinction between principles and practices? Data analysis is driven 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 in this handbook supersede specific technologies. Think of it like grammar. 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 grammar are relevant in whatever medium you choose. Similarly, this guide will teach you the basic "grammar" of democratic data analysis. This will allow 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. It isn't much to use to study grammar without ever writing a sentence.

The principles section of 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 become a data superstar, I would highly recommend learning `R`.

1.2 What is data analysis?

Data analysis is the process of transforming numbers on a page to insight into the real world. It's looking at a table and gaining insight from it. Data analysis can be as simple as adding totals into a column to see cumulative effects, or as complicated as time-series forecasting. But fundamentally, all data analysis is taking inputs and applying those inputs to the real world to gain insight into the real world. It also may be helpful to think about 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 qualitative data.

3. Data analysis isn't research methods.

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

- 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 The Grammar of Data Analysis[2]

As I mentioned before, democratic data analysis has an underlying structure, like a sentence There are rules so these sentences (hopefully) make sense to you, the reader. Similarly, by following common conventions of democratic data analysis, others will be able to "read" your analysis like you are reading this sentence. 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.
- 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. Each variable should have its own field.
- Observations
  - Observations make up the rows of your data set. 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
  - A table is the grouping of all observations of a similar type.

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

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 important 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 enlightenment. Having data formatted in the structure outlined above forces good data hygiene that will pay massive dividends later on.

Chapter 2

Tidy Data

Tidy data refers to having your data organized in a specific manner suitable for analysis. An obscene amount of time in data analysis is spent getting data into a tidy format. This chapter will walk through common problems and approaches in cleaning and tidying your data, that will make it easier for others to follow and easier for you to work across technologies and subject areas. Keeping your data in a tidy format for analysis will help because it is a fundamentally flexible way of working with data. Keeping scattered, lose data in spreadsheets is a sure way to cause confusion for yourself and others.

2.1 The Importance of Tables

If I could convince you of one thing, it would be the value in keeping your data in a table-based excel format. Again, this is true for data analysis purposes. If you are trying to do math or statistics, then the table based framework can sometimes fall short. But if you are concerned with working with data to draw conclusions, then tables are the way to go.

2.2 Cleaning vs Tidying

I hate cleaning, but love 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 Survey Monkey survey or a canned report that is run from the IT department, your data will

come in all shapes and sizes. Cleaning data is the process of transforming data values into ones that make sense for the purposes of your analysis.

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)
library(DT)

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 unfeasible 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. You can accomplish all these similar transformations using Get & Transform in Excel.

```
#creating a new table called "sites_cleaned", starting with the old table "sites"
sites_cleaned <- sites %>%
  #switching the order of names that are backwards
  mutate(Employee = if_else(Employee == "Knope, Leslie", "Leslie Knope", Employee)) %>%
  #seperate employee names into two columns
  separate(Employee, into = c("first_name", "last_name")) %>%
  #renaming column names to standard format
  rename(location = Location,
         telecommute_hours =`Telecommute?`,
         hire_date = `Hire Date`) %>%
  #changing first and last names and locations to Title Case
  mutate(first_name = str_to_title(first_name),
         last_name = str_to_title(last_name),
         location = str_to_title(location)) %>%
  #coding location data to three categories, In Office, In Field, or Other
  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"),
  #coding hours to numeric
         telecommute_hours = case_when(
           telecommute_hours == "never" ~ 0,
           telecommute_hours == "1 day/week" ~ 8,
           telecommute_hours == "40 hours" ~ 40
         )
         )

#print to datatable
sites_cleaned%>% datatable(
    extensions = 'Buttons',
    options = list(dom = 'Bfrtip',
                   buttons = 'excel',
                   searching = FALSE))
```

This may seem like a lot of work for a small table. But as your data grows, it is much easier to operate on entire fields at a time. This is especially true in Get & Transform, which makes it very easy to transform and clean data using all the same steps as the R code above. As you can see, our table is now "cleaned" and organized consistently.

Table 2.1: 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	C
Danielle	6	4	11	8	23	
Ramona	11	7	5	3	18	
Ross	10	8	10	7	10	

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 Retire
  "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
```

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[1] 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)
```

[1] powerquery hints

```
## # 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_from = val

print(visits_retirements_tidy2)
```

```
## # A tibble: 9 x 4
##   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 How is this democratic?

Thinking of your analysis in terms of tidy data is the first step on your way to democratic data analysis.

2.5 Conclusion

2.6 Practices/Resources

Chapter 3

Pivot to Win It

Why did we go through all the trouble of tidying data in the last section? So we can pivot. Thinking in terms of pivots, aka fields and summaries, is the key to effective democratic data analysis

3.1 Pivot Tables and the Meaning of Everything

> *The Ultimate Answer to Life, The Universe and Everything is…* ***the pivot table***

-Douglas Adams/Corban Nemeth

Pivot tables are the world's most common, most helpful, and most underrated data analysis tool. If you understand the mechanics of the pivot table, you have everything you need to be a data analysis expert. PowerBI or Tableau 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.

What is so important about pivot tables? First and foremost, pivot tables force you to think in terms of fields, not in terms of cells. In order for a pivot table to be effective, the data has to be organized in a table. And there is a right and a wrong way to putting data in a table. If your pivot table is not working properly, it is likely because your data isn't tidy.

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 separate 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.

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

#print to datatable
visits%>% datatable(
    extensions = 'Buttons',
    options = list(dom = 'Bfrtip',
                   buttons = 'excel',
                   searching = FALSE),
                   caption = "Visits to the Dept. of Retirement Services in a given mo
                   )
```

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 because our columns aren't truly variables. Remember, variables are elements of an observation that can change. You can drag the fields from the top row to the grey box below, for columns, and the left, for rows. This becomes unmanageable quickly.

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

3.2 Tidy Data- From Wide to Long

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

- Variables

[1] Data was created for demonstration purposes

Table 3.1: 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

- At first glance, it doesn'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 are 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
#we are editing the "visits" table already created by storing it in a new table pivot_visits
pivot_visits <- visits %>%
  #using pivot_longer on every column except "employee" and setting the name of the new columns
  pivot_longer(-Employee, names_to = "Visit Type", values_to = "Number of Visits")


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

Now this is a table that is much easier to analyze in an Excel pivot table or with a variety of R functions. However, it does look worse and is less intuitive for human readers. Thankfully, using data in this format, it is easy to recreate the original table for presentation, while also giving a variety of options for 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")
```

The key with pivot tables, such as the one above, is you are moving *fields* into the rows and columns of a new table. This is often referred to as "pivoting". You are

then selecting a calculation to *summarize* the intersection of the *fields* that you drug into rows and columns. Above, you can click on the "Count" button to see the other ways this lightweight pivot table allows you to *summarize* your data. More powerful data analysis tools, such as PowerPivot in Excel, PowerBI, and several R packages allow you infinitely flexible formulas to define what happens at the intersection of fields in a pivot table. This is the core of all (most) data analysis. Think in terms of fields and calculated summaries and you are well on your way to becoming a democratic data master.

It is also easier to do a variety of calculations on the data now that is in a "tidy" format. For example, creating descriptive statistics across groups is very straightforward. This can also be accomplished in an Excel pivot table with the basic "show values as" functions.

```
desc_visits <- pivot_visits %>%
  group_by(`Visit Type`) %>%
  summarise(`Avg Visits` = mean(`Number of Visits`),
            `Total Visits` = sum(`Number of Visits`),
            `Std Dev of Visits` = sd(`Number of Visits`))
```

This is where pivot tables, and the way they force you to think about data, become *enormously* powerful. Enter **PowerPivot**, stage left. PowerPivot allows you to write very detailed and useful functions for the values that appear in the middle of a pivot table. Above, I used R to calculate the average number of visits, total visits, and standard deviation of visits by visit type. But what if I quickly wanted to view this by Employee instead? Or what if I wanted to group by both visit type and employee, and generate the same descriptions? R and PowerPivot make this very easy. I'm not going to go into detail on the mechanics and tutorials of PowerPivot, as it is extensively covered elsewhere– see the Practices section of this guide.

Here's the key– Every fancy data dashboard, analysis technique, interactive visualization– is a pivot table with a "calculated field" (formula for values in a pivot table). That's it. If you can think in terms of fields and calculated values based on those fields, PowerBI, PowerPivot, Tableau, Qlik, ggplot2, etc. will be second nature. The only thing that changes when crating a viz instead of a pivot table is that your fields that you drag into a pivot table become the axis of your chart, and your calculated field becomes a "mark" on your plot. If you are interested more in a universal so-called "grammar of graphics" that is based on the principles outlined here, there is a wealth of resources available. (link to resources.)

3.3 Advanced Data Summaries

Frequently, public administrators are interested in how things have changed over a period of time. Here's an example dataset that shows referrals to the same

Dept. of Retirement Systems we have used in previous examples.

3.4 Conclusion

Thinking in terms of pivot tables is the core of democratic data analysis. This may not feel intuitive at first glance. But thinking this way forces the analyst to avoid the classic pitfalls of *un*democratic data analysis. Those pitfalls are the cell-based, hard-coded, pages and pages of tabs excel workbook nightmare that only the analyst themselves understands (and that's if they are lucky). Thinking in terms of the language of data analysis, emphasizing the use of fields and summaries, forces your analysis to be flexible, understandable, and reproducible. It is easy for an analyst who is also fluent in data analysis to pick up your work and immediately see how it operates because it is constructed using best practices and principles. It is also easy for someone else, or your future self, to write new calculations for data summaries, compare results, and use different fields.

3.5 Practices

Chapter 4

Reproducible Analysis

4.1 Do It For Your Future Self

Many things take more time to do up front, but save you from massive headaches down the road. Brushing your teeth. Oil changes. Preventative maintenance 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 receive 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 analysis 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 received. In Act 1, you use the practices we learned in the previous section to make the raw data tidy– without destroying 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.

4.1.1 Do Not Destroy

As I mentioned before, the existential dread that occurs when opening someone else's workbook and immediately receiving 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 unbound 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. Democratic data analysis starts with a foundation of data, and builds on top of it. Oftentimes, it seems easier to simply click and drag cells around in an excel workbook, changing values here and there as you see fit. This may work in small use cases, but what if you have another idea? Or come up with a different question, where your data needs to be coded differently? Reproducible analysis makes it substantially easier to revise and rewrite after the fact. If you were writing a well-sourced research article, you wouldn't delete your notes and references for the material that didn't make it into your final product. In the same way, keeping record of the changes that you make to your analysis will pay dividends when your approach changes.

4.1.2 Red flags for reproducibility

I'll start with a list of things you want to avoid in the pursuit of reproducible democratic data analysis

4.1.2.1 Copy and Paste

If you find yourself copying and pasting values in an excel workbook, you are not engaging in reproducible analysis– full stop. Copy and paste (or worse, cut and paste) doesn't leave breadcrumbs for you or anyone who may come after you. It is incredibly difficult to follow the trail of an analysis built on top of copy-paste.

4.1.2.2 Repeating Yourself

There is an old adage in programming - *D*on't *R*epeat *Y*ourself. DRY. Keeping your data analysis DRY is a good habit to get in to. If you find yourself

[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

repeating the same task more than three times, chances are there is a better, more programmatic way to go about what you are doing.

What do I mean by repeating yourself? This would be going through every row of an 100-row table to add (or remove)a space between words, capitalizing letters, doing specific calculations. All of these tasks can be **easily** automated using virtually all data analysis tools. This not only saves you time, but makes it easier for your reader to see how the data has changed in the course of your analysis.

To preview the Practices section, Get and Transform allows you to make flexible value transformations on entire fields at a time. This reduces the need for repetitive data cleaning. And if you are already taking my advice and no longer editing individual data cells, you will have a much easier time avoiding repeating yourself.

4.2 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.

4.3 Give Yourself Credit

Don't hard code (e.g. manually type) Excel values in cells. Build calculated summaries as discussed in the pivot-powered tab. Link to data sources. Highlight your expertise. This is easier to do when you use the tools shown in the practices section.

4.4 Version Control

4.4.1 Save as

4.4.2 Git/GitHub

If you are inclined to use R or another programming language for data analysis (and you really should be) then git is an essential tool in your toolkit. Full

disclosure, even I am not an expert on it. But there are several tools available to make version control using git seamless and effortless.

Git is an automated version control system that backs up your changes to the cloud, typically to github.com or another similar provider. It is very cool. For example, you can examine all the source code for this very book on GitHub.

4.5 Practices

The value of Get & Transform Data is that it forces you to build on top of your data foundation. It also conveniently records each step along the way. You even have the ability to save comments right there in the query editor. This is a remarkably easy and intuitive way to build a data transformation pipeline that will be valuable know and into the future.

Chapter 5

Uncertainty Oriented Analysis

All models are wrong, but some are useful

– George Box

5.1 Embracing Uncertainty for Better Outcomes

What is the point of data analysis? Often, it is to use data to summarize the world around you. In a sense, all data analysis is model building, and by definition, a model is a simplified version of the world. Any time the analyst is using data analysis to inform decision making, she is in a sense making a model. Models are commonly thought of to provide answers to the question at hand. Model building, and data analysis more generally, never gives you "the" answer. Data analysis only gives you a answer, if it gives you an answer at all. If you have been following along to this point, you have learned the value of data analysis that is structured and built up, not out. Frequently, data analysis is valuable because it tells you what isn't the answer. Knowing how to build uncertainty into your analysis is incredibly valuable as analysis – models– are used to implement important policies that impact the public good. Incorporating uncertainty into your analysis will make you both more credible and force you to use your subject matter expertise in addition to your data skills.

Any data analysis that simplifies the world (hint– that's all of it) can be considered a model. So I will use those phrases interchangeably in this section.

5.2 Why Model?

Models transform data into decision making. Models are useful exactly because they are wrong! Because models are wrong, we can critically examine the

5.3 Example two

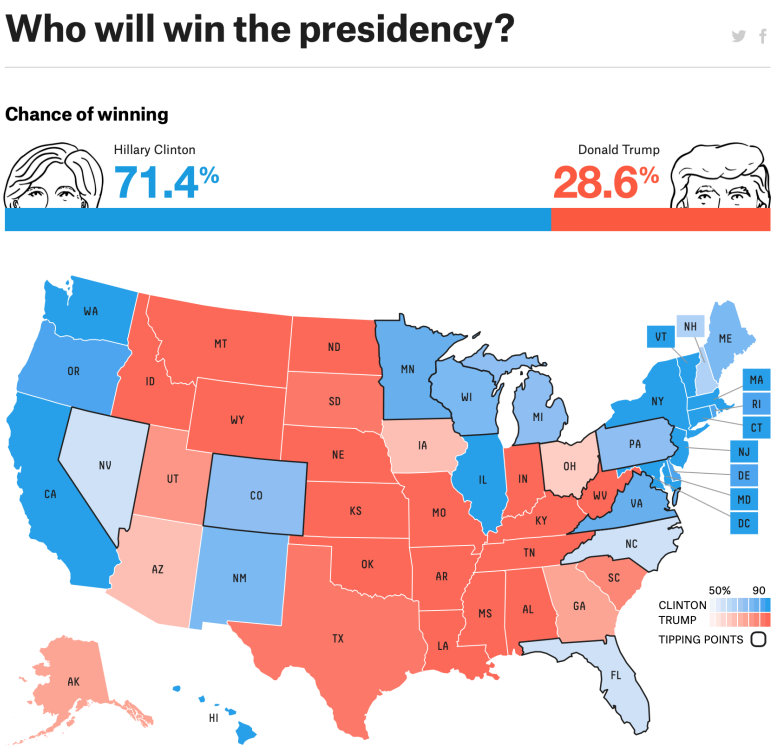

Figure 5.1: electoral 2016 map

5.4 Use Assumptions, and Document Them

(create dummy model regarding time saved through learning democratic data practices)

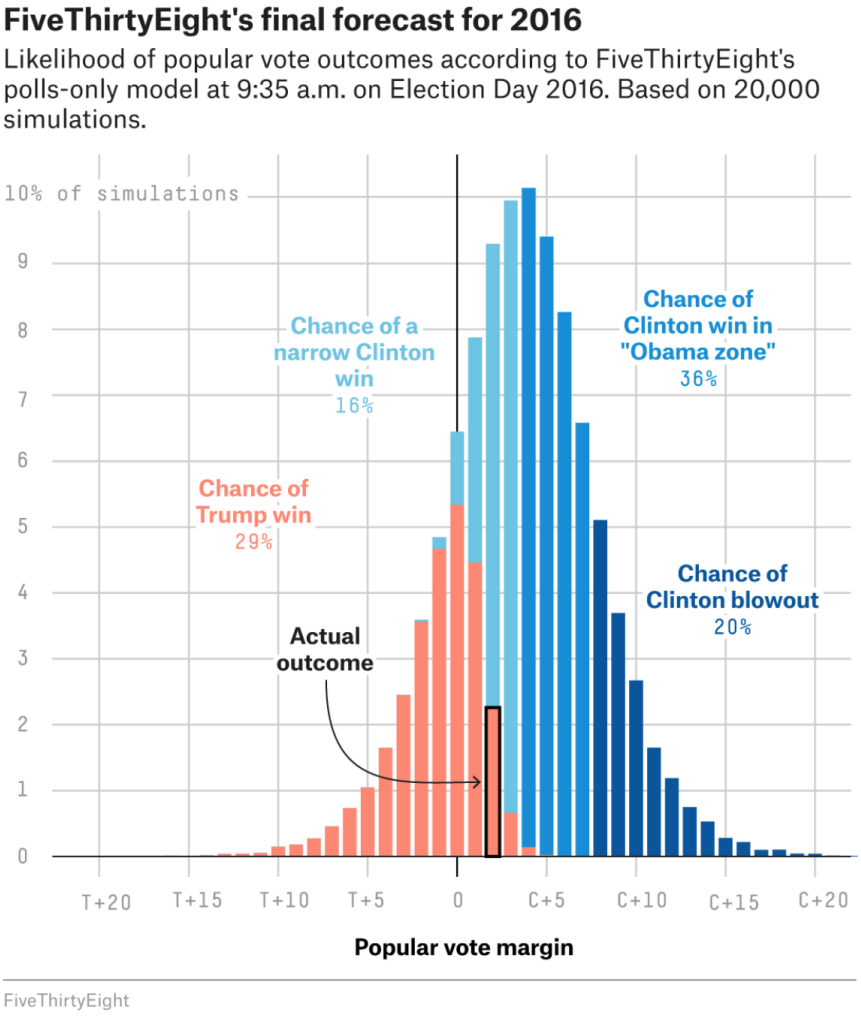

Figure 5.2: distribution of possible results

5.5 Don't get out over your skis

##Practices

Talk about modeling implies math or statistics, when it really doesn't. All a model seeks to do is to simply phenomena in a manner that can be comprehended and used to make decisions in conjunction with subject matter expertise.

Chapter 6

Audience-Focused

Just like writing, your data analysis always has an audience. Whether the audience is you, your coworker, your boss, or a policymaker, knowing your audience helps tailor how you present your findings.

6.1 The audience is you

This is where your reproducible principles are key. Because you are likely to forget that you edited cell B7, did a few copy/pastes, and hard-coded over some values when you come back to your workbook in a few weeks (or days, for that matter).

6.2 The audience is a peer

6.3 The audience is your manager

6.4 Show and Tell- Use highlights and captions. Design your data visualization. Please god don't use the defaults

Sometimes a table is totally appropriate. Sometimes it isn't.

What is the message you are trying to convey? There is no such thing as a neutral visualization.

Text is a data visualization. How can you bundle your data analysis and explanatory text? Designed excel workbooks as backup Rmarkdown/notebook format– the future. Interactive documents Classic papers with accompanying backup.

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

6.5 The audience is a policy/decision maker

When sharing data analysis with a policymaker or decision maker, it is important to keep in mind that the audience will often be much wider than initially assumed. Once your analysis is in the hands of a policymaker, it may go to the press, lobbyist organizations, other interest groups, or others inside or outside your organization. Additionally, policymakers often prefer paper or other "hard copy" type analysis, limiting your flexibility to simply share a file. This poses two unique constraints:

1) The mechanics of your analysis are often hidden in a print or a PDF
2) Others will see your work and either take it for granted or want to dissect it.

This is where having tidy, reproducible, and flexible backup becomes incredibly important. By having this backup ready to go as soon as you present or deliver your analysis to the policymaker, you are putting yourself in a strong position to share backup when questions inevitably arise.

If you are sharing your work electronically, then there are several emerging technologies and techniques that make this process easier. It is possible to embed Excel workbooks within Word documents, which ensures that the recipient has access to both your analysis and your methods. However, this can be tricky for ensuring adequate version control and reproducible, as these workbooks will often link to data that the end user doesn't have access to.

This problem is largely solved by using dashboards and other online solutions. Dashboards are an effective way to share analysis with decision makers because you can present your conclusions and also offer the tailored flexibility for the end user to interact as well. Tools such as PowerBI and Tableau run on the same tidy data and pivot-oriented platform discussed here. It is also straightforward to share the mechanics of the model with anyone who is interested– especially if you follow the best practices for reproducibility outlined earlier. The downside of such dashboard solutions is they can be expensive, proprietary, and it can be difficult to configure them for specific audiences.

6.6 My audience is all of the above- An Introduction to Literate Data Analysis

That brings me to what form democratic data analysis would take in the World-According-To-Corban. Another tool for presenting data analysis to decision-makers are interactive notebooks that merge explanatory text, data, code, and graphics in one file that can be tailored for a variety of audiences. I call this approach ***literate data analysis***[1]. Code-and-text driven notebooks are becoming exceedingly common in academic domains. In fact, this handbook is an example of an interactive HTML document written in `r`[2], but they certainly do not have to be this involved. Interactive notebooks combine text, code, and output in one place. They can be structured so that the file shows either plain-text analysis and charts in a web browser, but all of the code that generates it if you open the file in an editor such as RStudio. For example, here is an R notebook that I authored for a research project on legislative voting behavior, and I'll link to other interactive notebooks in the Practices section.

Here are three platforms of interactive, code-driven data analysis notebooks:

- Rmarkdown notebooks
- Jupyter notebooks
- Observable notebooks

I firmly believe that the notebook approach is the future of policy and data analysis in the public sector. These tools are free, robust, and accessible to those with varying levels of technical expertise. They also inherently solve the challenges of reproducible data analysis that I have been examining here. Version-controlled interactive notebooks are flexible, shareable, editable, and allow both decision-makers and analysts to use the same file to either draw conclusions or validate results.

I dream of a future when budgetary fiscal notes, quantitative bill analysis, and data models in the public sector are written as interactive notebooks, with explanatory text and analysis bundled together, version controlled, and presented with assumptions clearly documented.

This isn't to say that there aren't downsides to these approaches. The largest, as it currently stands, is the learning curve required to use these tools. I have yet to come across a notebook format that is as easily intuitive as an Excel workbook[3]. For now, the biggest downside of these tools is that they require at least a baseline understanding of `r`, `python`, `javascript`, or another similar scripting language. As we have already covered, Excel is a programming language, and probably the world's most common one at that. But Excel is familiar and

[1] derived from Literate Programming, but specific to subject matter expert data analysis
[2] using Rmarkdown files and the bookdown package
[3] Grid looks promising, but I haven't had a chance to full look into it yet

relatively beginner friendly because it hides the fact that the user is actually programming.

Once that baseline level of understanding is acquired, the notebook format becomes an intuitive output for reports. The same document can result in a PDF file to be shared with decision makers, an interactive document to be shared online, and a complete record of your analysis for a peer. Long story short, I'm excited to work towards integrating the notebook philosophy to the public sector as the next frontier of democratic data analysis.

Yihui Xie, the creator of Rmarkdown, summarizes these principles nicely: > I think notebooks are popular for the same reason that explains the popularity of spreadsheets such as Excel. I haven't met a single software engineer who loves Excel. Everyone hates it and makes fun of it, but why do so many users still use it? Again, Excel makes things tangible. You can touch the data (although it is usually a very bad idea), and draw graphics in a sheet that contains the source data (bad idea again). It makes you feel everything is well under your control: oh here is my data, and here is a graph next to it; oh I should use that column to draw the graph instead, so let me change it and I can see the updated graph immediately.2 You can do everything in a single place, and the short distance between the source (data) and the output is ace.

> Excel makes things tangible at the price of making things messy (e.g., it may contain manually edited data that is hard to keep track of, or merged cells or graphs that make it hard for other software to read the data). By comparison, although notebooks can mess up the state, but that is only an intermediate problem. At its core, it is still relatively clean and encourages the reproducibility principle, i.e., you shall use code to generate results automatically instead of manually copying and pasting results in your report. If you are concerned about the state, you can restart the session and recompile the whole notebook from scratch. Spreadsheets are often hopeless here—you cannot easily restart your brain and redo exactly the same things.

-Yihui Xie, Creator of Rmarkdown on notebooks and excel

Chapter 7

Applications

7.1 Tying it all together

How does this play out in the real world?

Saez and Zucman open policy analysis of Warren proposals

Fiscal notes– quantifying uncertainty and employing reproducibility

Questions to ask yourself: How can I pivot off of this data?

How can I think in terms of fields?

How can I make this more democratic?