Hacking Religion: TRS & Data Science in Action

Jeremy H. Kidwell

2023-09-29

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Introduction: Hacking Religion

1.1 Who this book is for

1.2 Why this book?

1.3 The hacker way

1. Tell the truth
2. Do not deceive using beauty
3. Work transparently: research as open code using open data
4. Draw others in: produce reproducible research
5. Learn by doing

1.4 Why programmatic data science?

This isn't just a book about data analysis, I'm proposing an approach which might be thought of as research-as-code, where you write out instructions to execute the various steps of work. The upside of this is that other researchers can learn from your work, correct and build on it as part of the commons. It takes a bit more time to learn and set things up, but the upside is that you'll gain access to a set of tools and a research philosophy which is much more powerful.

1.5 Learning to code: my way

This guide is a little different from other textbooks targetting learning to code. I remember when I was first starting out, I went through a fair few guides, and they all tended to spend about 200 pages on various theoretical bits, how you form an integer, or data structures, subroutines, or whatever, such that it was weeks before I got to actually *do* anything. I know some people, may prefer this approach, but I dramatically prefer a problem-focussed approach to learning. Give me something that is broken, or a problem to solve, which engages the things I want to figure out and the motivation for learning just comes much more naturally. And we know from research in cognitive science that these kinds of problem-focussed approaches can tend to faciliate faster learning and better retention, so it's not just my personal preference, but also justified! It will be helpful for you to be aware of this approach when you get into the book as it explains some of the editorial choices I've made and the way I've structured things. Each chapter focusses on a *problem* which is particularly salient for the use of data science to conduct research into religion. That problem will be my focal point, guiding choices of specific aspects of programming to introduce to you as we work our way around that data set and some of the crucial questions that arise in terms of how we handle it. If you find this approach unsatisfying, luckily there are a number of really terrific guides which lay things out slowly and methodically and I will explicitly signpost some of these along the way so that you can do a "deep dive" when you feel like it. Otherwise, I'll take an accelerated approach to this introduction to data science in R. I expect that you will identify adjacent resources and perhaps even come up with your own creative approaches along the way, which incidentally is how real data science tends to work in practice.

There are a range of terrific textbooks out there which cover all these elements in greater depth and more slowly. In particular, I'd recommend that many readers will want to check out Hadley Wickham's "R For Data Science" book. I'll include marginal notes in this guide pointing to sections of that book, and a few others which unpack the basic mechanics of R in more detail.

1.6 Getting set up

Every single tool, programming language and data set we refer to in this book is free and open source. These tools have been produced by professionals and volunteers who are passionate about data science and research and want to share it with the world, and in order to do this (and following the "hacker way") they've made these tools freely available. This also means that you aren't restricted to a specific proprietary, expensive, or unavailable piece of software to do this work. I'll make a few opinionated recommendations here based on my own preferences and experience, but it's really up to your own style and approach. In fact, given that this is an open source textbook, you can even propose additions to this chapter explaining other tools you've found that you want to share with others.

There are, right now, primarily two languages that statisticians and data scientists use for this kind of programmatic data science: python and R. Each language has its merits and I won't rehash the debates between various factions. For this book, we'll be using the R language. This is, in part, because the R user community and libraries tend to scale a bit better for the work that I'm commending in this book. However, it's entirely possible that one could use python for all these exercises, and perhaps in the future we'll have volume two of this book outlining python approaches to the same operations.

Bearing this in mind, the first step you'll need to take is to download and install R. You can find instructions and install packages for a wide range of hardware on the The Comprehensive R Archive Network (or "CRAN"): https://cran.rstudio.com. Once you've installed R, you've got some choices to make about the kind of programming environment you'd like to use. You can just use a plain text editor like `textedit` to write your code and then execute your programs using the R software you've just installed. However, most users, myself included, tend to use an integrated development environment (or "IDE"). This is usually another software package with a guided user interface and some visual elements that make it faster to write and test your code. Some IDE packages, will have built-in reference tools so you can look up options for libraries you use in your code, they will allow you to visualise the results of your code execution, and perhaps most important of all, will enable you to execute your programs line by line so you can spot errors more quickly (we call this “debugging”). The two most popular IDE platforms for R coding at the time of writing this textbook are RStudio and Visual Studio. You should download and try out both and stick with your favourite, as the differences are largely aesthetic. I use a combination of RStudio and an enhanced plain text editor Sublime Text for my coding.

Once you have R and your pick of an IDE, you are ready to go! Proceed to the next chapter and we’ll dive right in and get started!

2 The 2021 UK Census

2.1 Your first project: building a pie chart

Let's start by importing some data into R. Because R is what is called an object-oriented programming language, we'll always take our information and give it a home inside a named object. There are many different kinds of objects, which you can specify, but usually R will assign a type that seems to fit best.

In the example below, we're going to read in data from a comma separated value file ("csv") which has rows of information on separate lines in a text file with each column separated by a comma. This is one of the standard plain text file formats. R has a function you can use to import this efficiently called "read.csv". Each line of code in R usually starts with the object, and then follows with instructions on what we're going to put inside it, where that comes from, and how to format it:

If you'd like to explore this all in a bit more depth, you can find a very helpful summary in R for Data Science, chapter 8, "data import".

```
# R Setup -----------------------------------------------------------------
setwd("/Users/kidwellj/gits/hacking_religion_textbook/hacking_religion")
library(here) # much better way to manage working paths in R across multiple instances
```

```
here() starts at /Users/kidwellj/gits/hacking_religion_textbook
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
here::i_am("chapter_1.qmd")
```

```
here() starts at /Users/kidwellj/gits/hacking_religion_textbook/hacking_religion
```

```
uk_census_2021_religion <- read.csv(here("example_data", "census2021-ts030-rgn.csv"))
```

2.1.1 Examining data:

What's in the table? You can take a quick look at either the top of the data frame, or the bottom using one of the following commands:

```
head(uk_census_2021_religion)
```

```
                 geography   total no_religion christian buddhist  hindu jewish
1               North East 2647012     1058122   1343948     7026  10924   4389
2               North West 7417397     2419624   3895779    23028  49749  33285
3 Yorkshire and The Humber 5480774     2161185   2461519    15803  29243   9355
4            East Midlands 4880054     1950354   2214151    14521 120345   4313
5            West Midlands 5950756     1955003   2770559    18804  88116   4394
6                     East 6335072     2544509   2955071    26814  86631  42012
  muslim   sikh other no_response
1  72102   7206  9950      133345
2 563105  11862 28103      392862
3 442533  24034 23618      313484
4 210766  53950 24813      286841
5 569963 172398 31805      339714
6 234744  24284 36380      384627
```

This is actually a fairly ugly table, so I'll use an R tool called kable to give you prettier tables in the future, like this:

```r
knitr::kable(head(uk_census_2021_religion))
```

geography	total	no_religion	christian	buddhist	hindu	jewish	muslim	sikh	other	no_response
North East	2647012	1058122	1343948	7026	10924	4389	72102	7206	9950	133345
North West	7417397	2419624	3895779	23028	49749	33285	563105	11862	28103	392862
Yorkshire and The Humber	5480774	2161185	2461519	15803	29243	9355	442533	24034	23618	313484
East Midlands	4880054	1950354	2214151	14521	120345	4313	210766	53950	24813	386841
West Midlands	5950756	1955003	2770559	18804	88116	4394	569963	172398	31805	339714
East	6335072	2544509	2955071	26814	86631	42012	234744	24284	36380	384627

You can see how I've nested the previous command inside the `kable` command. For reference, in some cases when you're working with really complex scripts with many different libraries and functions, they may end up with functions that have the same name. You can specify the library where the function is meant to come from by preceding it with :: as we've done `knitr::` above. The same kind of output can be gotten using `tail`:

```r
knitr::kable(tail(uk_census_2021_religion))
```

	geography	total	no_religion	christian	buddhist	hindu	jewish	muslim	sikh	other	no_response
5	West Midlands	5950756	1955003	2770559	18804	88116	4394	569963	172398	31805	339714
6	East	6335072	2544509	2955071	26814	86631	42012	234744	24284	36380	384627
7	London	8799728	2380404	3577681	77425	453034	145466	1318754	144543	86759	815662
8	South East	9278068	3733094	4313319	54433	154748	18682	309067	74348	54098	566279

	geography	total	no_religion	christian	buddhist	hindu	jewish	muslim	sikh	other	no_response
9	South West	5701186	2513369	2635872	24579	27746	7387	80152	7465	36884	367732
10	Wales	3107494	1446398	1354773	10075	12242	2044	66947	4048	15926	195041

2.1.2 Parsing and Exploring your data

The first thing you're going to want to do is to take a smaller subset of a large data set, either by filtering out certain columns or rows. Now let's say we want to just work with the data from the West Midlands, and we'd like to omit some of the columns. We can choose a specific range of columns using `select`, like this:

You can use the `filter` command to do this. To give an example, `filter` can pick a single row in the following way:

```
uk_census_2021_religion_wmids <- uk_census_2021_religion %>%
  filter(geography=="West Midlands")
```

Now we'll use select in a different way to narrow our data to specific columns that are needed (no totals!).

Some readers will want to pause here and check out Hadley Wickham's "R For Data Science" book, in the section, "Data visualisation" to get a fuller explanation of how to explore your data.

In keeping with my goal to demonstrate data science through examples, we're going to move on to producing some snappy looking charts for this data.

2.2 Making your first chart

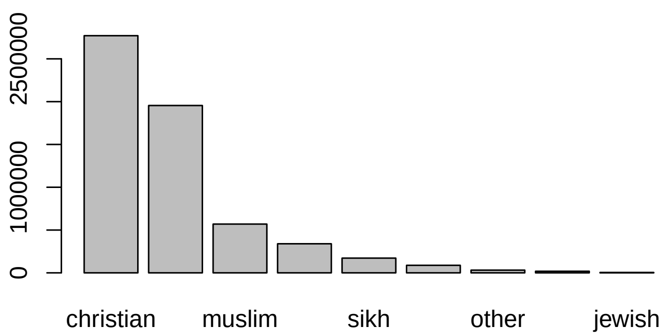
We've got a nice lean set of data, so now it's time to visualise this. We'll start by making a pie chart:

```
uk_census_2021_religion_wmids <- uk_census_2021_religion_wmids %>% select(no_religion:no_res
uk_census_2021_religion_wmids <- gather(uk_census_2021_religion_wmids)
```

There are two basic ways to do visualisations in R. You can work with basic functions in R, often called "base R" or you can work with an alternative library called ggplot:

2.2.0.1 Base R

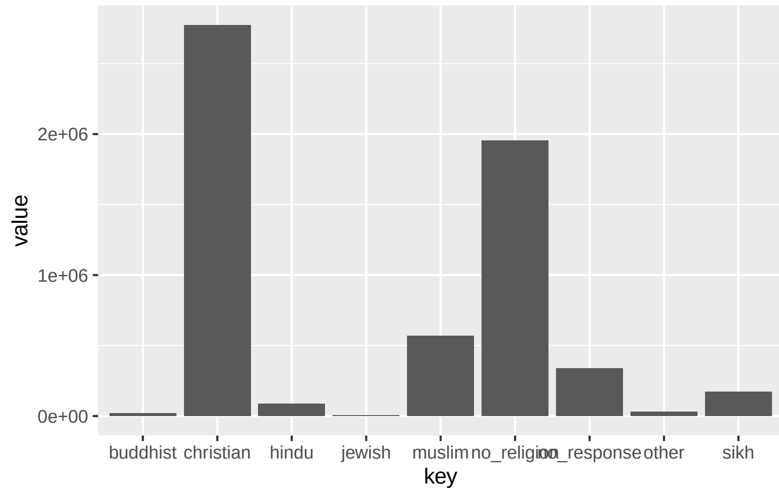
```
df <- uk_census_2021_religion_wmids[order(uk_census_2021_religion_wmids$value,decreasing = T
barplot(height=df$value, names=df$key)
```



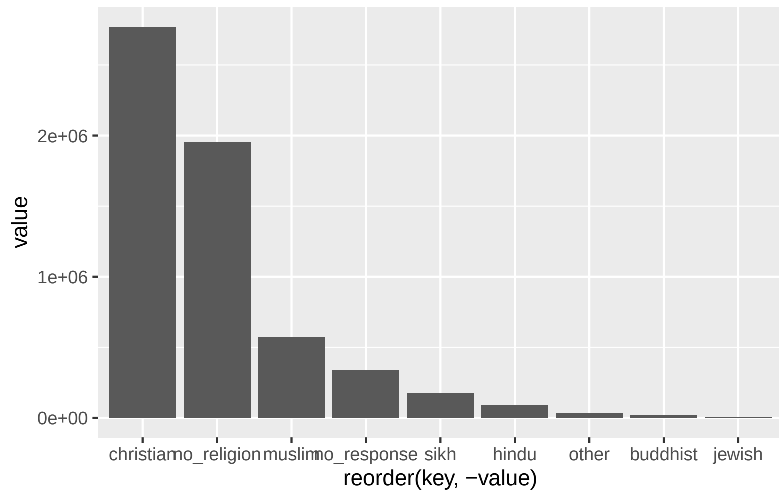
2.2.0.2 GGPlot

```
ggplot(uk_census_2021_religion_wmids, aes(x = key, y = value)) +   (1)
  geom_bar(stat = "identity")
```

(2) We'll re-order the column by size.




```
ggplot(uk_census_2021_religion_wmids, aes(x= reorder(key,-value),value)) + geom_bar(stat ="i
```



Let's assume we're working with a data set that doesn't include a "totals" column and that we might want to get sums for each column. This is pretty easy to do in R:

```
uk_census_2021_religion_totals <- uk_census_2021_religion %>% select(no_religion:no_response
uk_census_2021_religion_totals <- uk_census_2021_religion_totals %>%
```

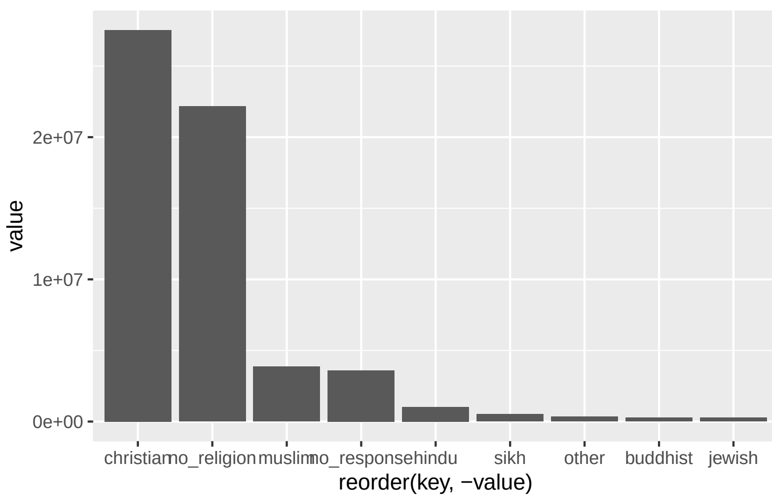
```
  summarise(across(everything(), ~ sum(., na.rm = TRUE)))  ②
uk_census_2021_religion_totals <- gather(uk_census_2021_religion_totals) ③
ggplot(uk_census_2021_religion_totals, aes(x= reorder(key,-value),value)) + geom_bar(stat ="
```

① First, remove the column with region names and the totals for the regions as we want just integer data.

② Second calculate the totals. In this example we use the tidyverse library `dplyr()`, but you can also do this using base R with `colsums()` like this: `uk_census_2021_religion_totals <- colSums(uk_census_2021_religion_totals, na.rm = TRUE)`. The downside with base R is that you'll also need to convert the result into a dataframe for `ggplot` like this: `uk_census_2021_religion_totals <- as.data.frame(uk_census_2021_religion_totals)`

③ In order to visualise this data using ggplot, we need to shift this data from wide to long format. This is a quick job using gather()

④ Now plot it out and have a look!



You might have noticed that these two dataframes give us somewhat different results. But with data science, it's much more interesting to compare these two side-by-side in a visualisation. We can join these two dataframes and plot the bars side by side using `bind()` - which can be done by columns with cbind() and rows using rbind():

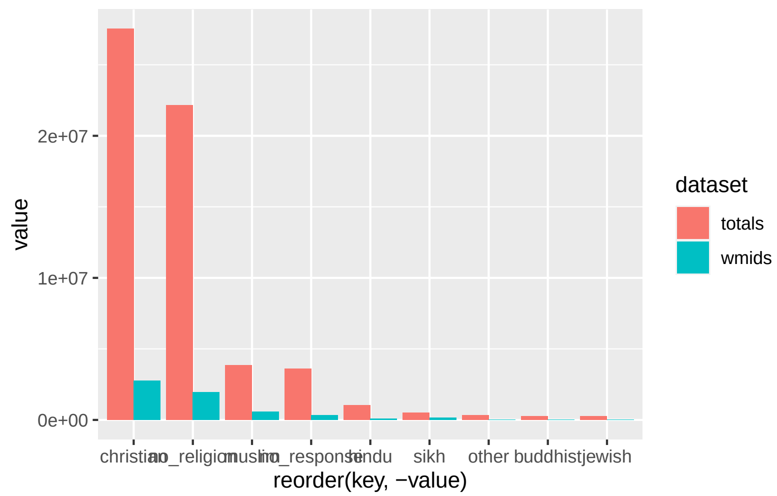
```
uk_census_2021_religion_merged <- rbind(uk_census_2021_religion_totals, uk_census_2021_relig
```

Do you notice there's going to be a problem here? How can we tell one set from the other? We need to add in something idenfiable first! This isn't too hard to do as we can simply create a new column for each with identifiable information before we bind them:

```
uk_census_2021_religion_totals$dataset <- c("totals")
uk_census_2021_religion_wmids$dataset <- c("wmids")
uk_census_2021_religion_merged <- rbind(uk_census_2021_religion_totals, uk_census_2021_relig
```

Now we're ready to plot out our data as a grouped barplot:

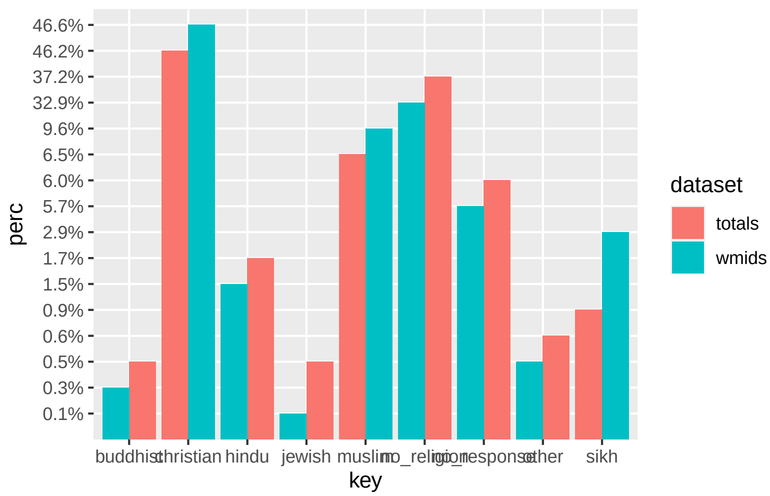
```
ggplot(uk_census_2021_religion_merged, aes(fill=dataset, x= reorder(key,-value), value)) + g
```



If you're looking closely, you will notice that I've added two elements to our previous ggplot. I've asked ggplot to fill in the columns with reference to the `dataset` column we've just created. Then I've also asked ggplot to alter the `position="dodge"` which places bars side by side rather than stacked on top of one another. You can give it a try without this instruction to see how this works. We will use stacked bars in a later chapter, so remember this feature.

If you inspect our chart, you can see that we're getting closer, but it's not really that helpful to compare the totals. What we need to do is get percentages that can be compared side by side. This is easy to do using another `dplyr` feature `mutate`:

```
uk_census_2021_religion_totals <- uk_census_2021_religion_totals %>%
  dplyr::mutate(perc = scales::percent(value / sum(value), accuracy = 0.1, trim = FALSE)) ③
uk_census_2021_religion_wmids <- uk_census_2021_religion_wmids %>%
  dplyr::mutate(perc = scales::percent(value / sum(value), accuracy = 0.1, trim = FALSE))
uk_census_2021_religion_merged <- rbind(uk_census_2021_religion_totals, uk_census_2021_relig
ggplot(uk_census_2021_religion_merged, aes(fill=dataset, x=key, y=perc)) + geom_bar(position
```



Now you can see a very rough comparison

Add time series data for 2001 and 2011 census, change to grouped bar plot:

https://r-graphics.org/recipe-bar-graph-grouped-bar#discussion-8

References

3 Survey Data: Spotlight Project

In the last chapter we explored some high level data about religion in the UK. This was a census sample, which usually refers to an attempt to get as comprehensive a sample as possible. But this is actually fairly unusual in practice. Depending on how complex a subject is, and how representative we want our data to be, it's much more common to use selective sampling, that is survey responses at n=100 or n=1000 at a maximum. The advantage of a census sample is that you can explore how a wide range of other factors - particularly demographics - intersect with your question. And this can be really valuable in the study of religion, particularly as you will see as we go along that responses to some questions are more strongly correlated to things like economic status or educational attainment than they are to religious affiliation. It can be hard to tell if this is the case unless you have enough of a sample to break down into a number of different kinds of subsets. But census samples are complex and expensive to gather, so they're quite rare in practice.

For this chapter, I'm going to walk you through a data set that a colleague (Charles Ogunbode) and I collected in 2021. Another problem with smaller, more selective samples is that researchers can often undersample minoritised ethnic groups. This is particularly the case with climate change research. Until the time we conducted this research, there had not been a single study investigating the specific experiences of people of colour in relation to climate change in the UK. Past researchers had been content to work with large samples, and assumed that if they had done 1000 surveys and 50 of these were completed by people of colour, they could "tick" the box. But 5% is actually well below levels of representation in the UK generally, and even more sharply the case for specific communities. And if we bear in mind that non-white respondents are (of course!) a highly heterogenous group, we're even more behind in terms of collecting data that can improve our knowledge. Up until recently researchers just haven't been paying close enough attention to catch the significant neglect of the empirical field that this represents.

While I've framed my comments above in terms of climate change research, it is also the case that, especially in diverse societies like the USA, Canada, the UK etc., paying attention to non-majority groups and people and communities of colour automatically draws in a strongly religious sample. This is highlighted in one recent study done in the UK, the "Black British Voices Report" in which the researchers observed that "84% of respondents described themselves as religious and/or spiritual". My comments above in terms of controlling for other factors remains important here - these same researchers also note that "despire their significant important to the lives of Black Britons, only 7% of survey respondents reported that their religion was more defining of their identity than their race".

We've decided to open up access to our data and I'm highlighting it in this book because it's a unique opportunitiy to explore a dataset that emphasises diversity from the start, and by extension, provides some really interesting ways to use data science techniques to explore religion in the UK.

4 Loading in some data

```
# R Setup -----------------------------------------------------------------
setwd("/Users/kidwellj/gits/hacking_religion_textbook/hacking_religion")
library(here)
```

```
here() starts at /Users/kidwellj/gits/hacking_religion_textbook
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(haven) # used for importing SPSS .sav files
here::i_am("chapter_2.qmd")
```

```
here() starts at /Users/kidwellj/gits/hacking_religion_textbook/hacking_religion
```

```
climate_experience_data <- read_sav(here("example_data", "climate_experience_data.sav"))
```

The first thing to note here is that we've drawn in a different type of data file, this time from an `.sav` file, usully produced by the statistics software package SPSS. This uses a different R Library (I use `haven` for this). The upside is that in some cases where you have survey data with both a code and a value like "1" is eqivalent to "very much agree" this will preserve both in the R dataframe that is created. Now that you've loaded in data, you have a new R dataframe called "climate_experience_data" with a lot of columns with just under 1000 survey responses.

5 How can you ask about religion?

One of the challenges we faced when running this study is how to gather responsible data from surveys regarding religious identity. We'll dive into this in depth as we do analysis and look at some of the agreements and conflicts in terms of respondent attribution. Just to set the stage, we used the following kinds of question to ask about religion and spirituality:

1. Question 56 asks respondents simply, "What is your religion?" and then provides a range of possible answers. We included follow-up questions regarding denomination for respondents who indicated they were "Christian" or "Muslim". For respondents who ticked "Christian" we asked, "What is your denomination?" nad for respondents who ticked "Muslim" we asked "Which of the following would you identify with?" and then left a range of possible options which could be ticked such as "Sunni," "Shia," "Sufi" etc.

This is one way of measuring religion, that is, to ask a person if they consider themselves formally affiliated with a particular group. This kind of question has some (serious) limitations, but we'll get to that in a moment.

We also asked respondents (Q57): "Regardless of whether you belong to a particular religion, how religious would you say you are?" and then provided a slider from 0 (not religious at all) to 10 (very religious).

We included some classic indicators about how often respondents go to worship (Q58): Apart from weddings, funerals and other special occasions, how often do you attend religious services? and (Q59): "Q59 Apart from when you are at religious services, how often do you pray?"

- More than once a week (1)
- Once a week (2)
- At least once a month (3)
- Only on special holy days (4)
- Never (5)

Each of these measures a particular kind of dimension, and it is interesting to note that sometimes there are stronger correlations between how often a person attends worship services (weekly versus once a year) and a particular view, than there is between their affiliation (if they are Christian or Pagan). We'll do some exploratory work shortly to see how this is the case in our sample. We also included a series of questions about spirituality in Q52 and used a nature relatedness scale Q51.

You'll find that many surveys will only use one of these forms of question and ignore the rest. I think this is a really bad idea as religious belonging, identity, and spirituality are far too complex to work off a single form of response. We can also test out how these different attributions relate to other demographic features, like interest in politics, economic attainment, etc.

> So *who's* religious?
>
> As I've already hinted in the previous chapter, measuring religiosity is complicated. I suspect some readers may be wondering something like, "what's the right question to ask?" here. Do we get the most accurate representation by asking people to self-report their religious affiliation? Or is it more accurate to ask individuals to report on how religious they are? Is it, perhaps, better to assume that the indirect query about practice, e.g. how frequently one attends services at a place of worship may be the most reliable proxy?
> Highlight challenges of various approaches pointing to literature.

Let's dive into the data and see how this all works out. We'll start with the question 56 data, around religious affiliation:

```
religious_affiliation <- as_tibble(as_factor(climate_experience_data$Q56))          ①
names(religious_affiliation) <- c("response")  ②
religious_affiliation <- filter(religious_affiliation, !is.na(response)) ③
```

There are few things we need to do here to get the data into initial proper shape. This might be called "cleaning" the data:

1. Because we imported this data from an SPSS `.sav` file format using the R `haven()` library, we need to start by adapting the data into a format that our visualation engine ggplot can handle (a dataframe).
2. Next we'll rename the columns so these names are a bit more useful.
3. We need to omit non-responses so these don't mess with the counting (these are `NA` in R)

If we pause at this point to view the data, you'll see it's basically just a long list of survey responses. What we need is a count of each unique response (or `factor`). This will take a few more steps:

```
religious_affiliation_sums <- religious_affiliation %>%
  dplyr::count(response, sort = TRUE) %>%     ①
  dplyr::mutate(response = forcats::fct_rev(forcats::fct_inorder(response)))    ②
religious_affiliation_sums <- religious_affiliation_sums %>%
  dplyr::mutate(perc = scales::percent(n / sum(n), accuracy = .1, trim = FALSE)) ③
```

① First we generate new a dataframe with sums per category and
② ...sort in descending order
③ Then we add new column with percentages based on the sums you've just generated

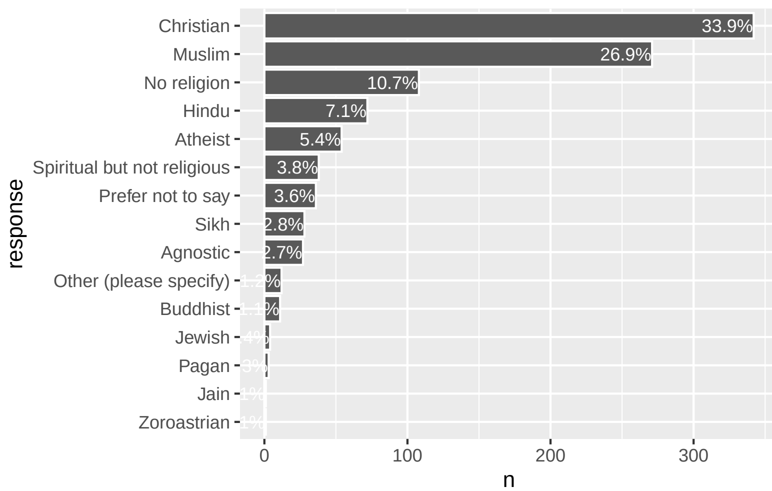
That should give us a tidy table of results, which you can see if you view the contents of our new `religious_affiliation_sums` dataframe:

```
head(religious_affiliation_sums)
```

```
# A tibble: 6 x 3
```

```
  response                                 n perc
  <fct>                                <int> <chr>
1 Christian                              342 "33.9%"
2 Muslim                                 271 "26.9%"
3 No religion                            108 "10.7%"
4 Hindu                                   72 " 7.1%"
5 Atheist                                 54 " 5.4%"
6 Spiritual but not religious             38 " 3.8%"
```

```r
# make plot
ggplot(religious_affiliation_sums, aes(x = n, y = response)) +
  geom_col(colour = "white") +
  ## add percentage labels
  geom_text(aes(label = perc),
            ## make labels left-aligned and white
            hjust = 1, nudge_x = -.5, colour = "white", size=3)
```



Add colours Use mutate to put "prefer not to say" at the bottom
Info here: https://r4ds.had.co.nz/factors.html#modifying-factor-levels

6 Q56 follow-ups

caption <- “Christian Denomination” # TODO: copy plot above for Q56 to add two additional plots using climate_experience_data_named$Q56bandclimate_{e}xperience_{d}ata_{n}amed$Q56c
# Religious Affiliation b - Christian Denomination Subquestion
christian_denomination <- qualtrics_process_single_multiple_choice(climate_experience_data_named$Q56b)ch$
$-chart_{s}ingle_{r}esult_{f}lextable(climate_{e}xperience_{d}ata_{n}amed$Q56b,
desc(Count)) christian_denomination_table save_as_docx(christian_denomination_table, path = “./figures/q56_religious_affiliation_xn_denomination.docx”)

christian_denomination_hi <- filter(climate_experience_data_named, Q56 == “Christian”, Q57_bin == “high”) christian_denomination_hi <- qualtrics_process_single_multiple_choice(christian_denomination_hi$Q56b) christian_denomination_hi

7 Religious Affiliation c - Muslim Denomination Subquestion

caption <- “Islamic Identity” # Should the label be different than income since the data examined is the Affiliation? # TODO: adjust plot to factor using numbered responses on this question (perhaps also above) religious_affiliationc <- qualtrics_process_single_multiple_choice(climate_experience_data_named$Q56c)religious_affiliationc_plot < -plot_horizontal_bar(religious_affiliationc)religious_affiliationc_plot < -religious_affiliationc_plot + labs(caption = caption, x = "", y = "")religious_affiliationc_plotggsave("figures/q56c_religious_affiliation.png", width = 20, height = 10, units = "cm")religious_affiliationc_table < -chart_single_result_flextable(climate_experience_data_named$Q56c, Count) religious_affiliationc_table save_as_docx(religious_affiliationc_table, path = “./figures/q56_religious_affiliation_islam.docx”)

8 Q57

9 Religiosity

caption <- “Respondent Religiosity” religiosity <- qualtrics_process_single_multiple_choice(as.character(climate_
$-plot_horizontal_bar(religiosity)religiosity_plot < -religiosity_plot+$
$labs(caption = caption, x = "", y = "")religiosity_plotggsave("figures/q57_religiosity_plot.png", width =$
20, $height = 10, units = "cm")religiosity_table < -chart_single_result_flextable(climate_experience_data_named$Q57_
desc(Variable)) religiosity_table save_as_docx(religious_affiliationc_table,
path = “./figures/q57_religiousity.docx”)

10 Q58

caption <- “Respondent Attendance of Religious Services” religious_service_attend <- qualtrics_process_single_multiple_choice(climate_experience_data_named$Q58)religi$
$-plot_horizontal_bar(religious_service_attend)religious_service_attend_plot <$
$-religious_service_attend_plot + labs(title = caption, x = "", y =$
$"")religious_service_attend_plotggsave("figures/q58_religious_service_attend.png", width =$
$20, height = 10, units = "cm")religious_service_attend_table <$
$-chart_single_result_flextable(climate_experience_data_named$Q58,
Count) religious_service_attend_table save_as_docx(religious_service_attend_table,
path = “./figures/q58_religious_service_attend.docx”)

11 Faceted plot working with 3x3 grid

df <- select(climate_experience_data, Q52_bin, Q53_bin, Q57_bin, Q58) names(df) <- c("Q52_bin", "Q53_bin", "Q57_bin", "response") facet_names <- c(`Q52_bin` = "Spirituality", `Q53_bin` = "Politics L/R", `Q57_bin` = "Religiosity", `low`="low", `medium`="medium", `high`="high") facet_labeller <- function(variable,value){return(facet_names[value])} df$response < -factor(df$response, ordered = TRUE, levels = c("1", "2", "3", "4", "5")) df$response < -fct_recode(df$response, "More than once a week" = "1", "Once a week" = "2", "At least once a month" = "3", "Only on special holy days" = "4", "Never" = "5") df %>% # we need to get the data including facet info in long format, so we use pivot_longer() pivot_longer(!response, names_to = "bin_name", values_to = "b") %>% # add counts for plot below count(response, bin_name, b) %>% group_by(bin_name,b) %>% mutate(perc=paste0(round(n*100/sum(n),1),"%")) %>% # run ggplot ggplot(aes(x = n, y = "", fill = response)) + geom_col(position=position_fill(), aes(fill=response)) + geom_text(aes(label = perc), position = position_fill(vjust=.5), size=2) + scale_fill_brewer(palette ="Dark2", type = "qual") + scale_x_continuous(labels = scales::percent_format()) + facet_grid(vars(b), vars(bin_name), labeller=as_labeller(facet_names)) + labs(caption = caption, x = "", y ="") + guides(fill = guide_legend(title = NULL)) ggsave("figures/q58_faceted.png", width = 30, height = 10, units = "cm")

12 Q59

caption <- “Respondent Prayer Outside of Religious Services”
prayer <- qualtrics_process_single_multiple_choice(climate_experience_data_named$Q59)prayer_plot < -plot_horizontal_bar(prayer)prayer_plot < -prayer_plot + labs(caption = caption, x = "", y = "")prayer_plot ggsave("figures/q59_prayer.png", width = 20, height = 10, units = "cm")prayer_table < -chart_single_result_flextable(climate_experience_data_named$Q59, Count) prayer_table save_as_docx(prayer_table, path = “./figures/q59_prayer.docx”)

13 Faceted plot working with 3x3 grid

df <- select(climate_experience_data, Q52_bin, Q53_bin, Q57_bin, Q59) names(df) <- c("Q52_bin", "Q53_bin", "Q57_bin", "response") facet_names <- c(`Q52_bin` = "Spirituality", `Q53_bin` = "Politics L/R", `Q57_bin` = "Religiosity", `low`="low", `medium`="medium", `high`="high") facet_labeller <- function(variable,value){return(facet_names[value])} df$response < -factor(df$response, ordered = TRUE, levels = c("1", "2", "3", "4", "5")) df$response < -fct_recode(df$response, "More than once a week" = "1", "Once a week" = "2", "At least once a month" = "3", "Only on special holy days" = "4", "Never" = "5") df %>% # we need to get the data including facet info in long format, so we use pivot_longer() pivot_longer(!response, names_to = "bin_name", values_to = "b") %>% # add counts for plot below count(response, bin_name, b) %>% group_by(bin_name,b) %>% mutate(perc=paste0(round(n*100/sum(n),1),"%")) %>% # run ggplot ggplot(aes(x = n, y = "", fill = response)) + geom_col(position=position_fill(), aes(fill=response)) + geom_text(aes(label = perc), position = position_fill(vjust=.5), size=2) + scale_fill_brewer(palette ="Dark2", type = "qual") + scale_x_continuous(labels = scales::percent_format()) + facet_grid(vars(b), vars(bin_name), labeller=as_labeller(facet_names)) + labs(caption = caption, x = "", y ="") + guides(fill = guide_legend(title = NULL)) ggsave("figures/q59_faceted.png", width = 30, height = 10, units = "cm")

14 Comparing with attitudes surrounding climate change

15 Q6

q6_data <- qualtrics_process_single_multiple_choice_unsorted_streamlined(climate_experience_data$Q6)

title <- “Do you think the climate is changing?”

level_order <- c(“Don<80><99>t know”, “Definitely not changing”, “Probably not changing”, “Probably changing”, “Definitely changing”) ## code if a specific palette is needed for matching fill = wheel(ochre, num = as.integer(count(q6_data[1]))) # make plot q6_data_plot <- ggplot(q6_data, aes(x = n, y = response, fill = fill)) + geom_col(colour = “white”) + ## add percentage labels geom_text(aes(label = perc), ## make labels left-aligned and white hjust = 1, colour = “black”, size=4) + # use nudge_x = 30, to shift position ## reduce spacing between labels and bars scale_fill_identity(guide = “none”) + ## get rid of all elements except y axis labels + adjust plot margin theme_ipsum_rc() + theme(plot.margin = margin(rep(15, 4))) + easy_center_title() + # with thanks for helpful info on doing wrap here: https://stackoverflow.com/questions/21878974/wrap-long-axis-labels-via-labeller-label-wrap-in-ggplot2 scale_y_discrete(labels = wrap_format(30), limits = level_order) + theme(plot.title = element_text(size =18, hjust = 0.5), axis.text.y = element_text(size =16)) + labs(title = title, x = ““, y =”“)

q6_data_plot

ggsave(“figures/q6.png”, width = 18, height = 12, units = “cm”)

16 Subsetting

16.1 Q57 subsetting based on Religiosity ——

climate_experience_data <- climate_experience_data %>% mutate(Q57_bin = case_when(Q57_1 > mean(Q57_1) + sd(Q57_1) ~ “high”, Q57_1 < mean(Q57_1) - sd(Q57_1) ~ “low”, TRUE ~ “medium”) %>% factor(levels = c(“low”, “medium”, “high”)))

16.2 Subsetting based on Spirituality ——

16.2.1 Nature relatedness ——

17 Calculate overall mean nature-relatedness score based on six questions:

climate_experience_data$Q51_score <- rowMeans(select(climate_experience_data, Q51_remote_vacation:Q51_heritage))

18 Create low/med/high bins based on Mean and +1/-1 Standard Deviation

climate_experience_data <- climate_experience_data %>% mutate(Q51_bin = case_when(Q51_score > mean(Q51_score) + sd(Q51_score) ~ "high", Q51_score < mean(Q51_score) - sd(Q51_score) ~ "low", TRUE ~ "medium") %>% factor(levels = c("low", "medium", "high")))

18.0.1 Spirituality scale ————————————————————————————

19 Calculate overall mean spirituality score based on six questions:

```
climate_experience_data$Q52_score <- rowMeans(select(climate_experience_data,
Q52a_1:Q52f_1))
```

20 Create low/med/high bins based on Mean and +1/-1 Standard Deviation

climate_experience_data <- climate_experience_data %>% mutate(Q52_bin = case_when(Q52_score > mean(Q52_score) + sd(Q52_score) ~ "high", Q52_score < mean(Q52_score) - sd(Q52_score) ~ "low", TRUE ~ "medium") %>% factor(levels = c("low", "medium", "high")))

> **What is Religion?**
>
> Content tbd

> **Hybrid Religious Identity**
>
> Content tbd

> **What is Secularisation?**
>
> Content tbd

References

21 Mapping churches: geospatial data science

Guides to geographies: https://rconsortium.github.io/censusguide/ https://ocsi.uk/2019/03/18/lsoas-leps-and-lookups-a-beginners-guide-to-statistical-geographies/

Extact places of worship from Ordnance survey open data set Calculate proximity to pubs

References

22 Data scraping, corpus analysis and wordclouds

References

23 Summary

An open textbook introducing data science to religious studies

References