GATHERING DATA FROM MESSY SOURCES

ROHAN ALEXANDER

INTRODUCTION

OVERVIEW

- Gathering, preparing, and cleaning data are essential steps before any type of quantitative analysis can be conducted.
- In this seminar, I will use R to ethically gather data from real world sources including an API, Wikipedia, and PDFs.
- I will then prepare and clean these data based on several principals that I will introduce.

RECOMMENDED READING

- Wickham, Hadley, 2014, 'rvest: easy web scraping with R', 24 November, https://blog.rstudio.com/2014/11/24/rvest-easy-web-scraping-with-r/.
- Alexander, Rohan, 2019, 'Gathering and analysing text data', 3 January, https://rohanalexander.com/posts/2019-01-03-gathering-and-analysing-text-data/.
- Henze, Martin, 2020, 'Web Scraping with rvest + Astro Throwback', 23 January, https://heads0rtai1s.github.io/2020/01/23/rvest-intro-astro/.
- Silge, Julia, 2017, 'Scraping CRAN with rvest', 5 March, https://juliasilge.com/blog/scraping-cran/.

KEY SKILLS AND CONCEPTS

1. BEGIN WITH AN END IN MIND.
2. START SIMPLE, THEN ITERATE.
3. CONTROL YOURSELF
4. SHOW DON'T TELL.

KEY SOFTWARE

- R
  - R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- R Studio
  - https://rstudio.com/

KEY LIBRARIES

- library(janitor)
  - Sam Firke (2020). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package version 2.0.1. https://CRAN.R-project.org/package=janitor
- library(pdftools)
  - Jeroen Ooms (2019). pdftools: Text Extraction, Rendering and Converting of PDF Documents. R package version 2.3.
- library(tidyverse)
  - Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686,
- library(stringi)
  - Gagolewski M. and others (2020). R package stringi: Character string processing facilities.

KEY FUNCTIONS

- case_when()
- distinct()
- geom_col()
- ggplot()
- group_by()
- head()
- if_else()
- map_dfr()
- mutate()
- mutate_all()
- mutate_at()
- pdf_text()
- pivot_longer()
- rbind()
- remove_empty()
- rename()
- select()
- separate()
- str_detect()
- str_remove_all()
- str_replace()
- str_squish()
- str_to_title()
- stri_split_lines()
- ungroup()
- unique()
- write_csv()

GETTING DATA FROM PDFS INTO R

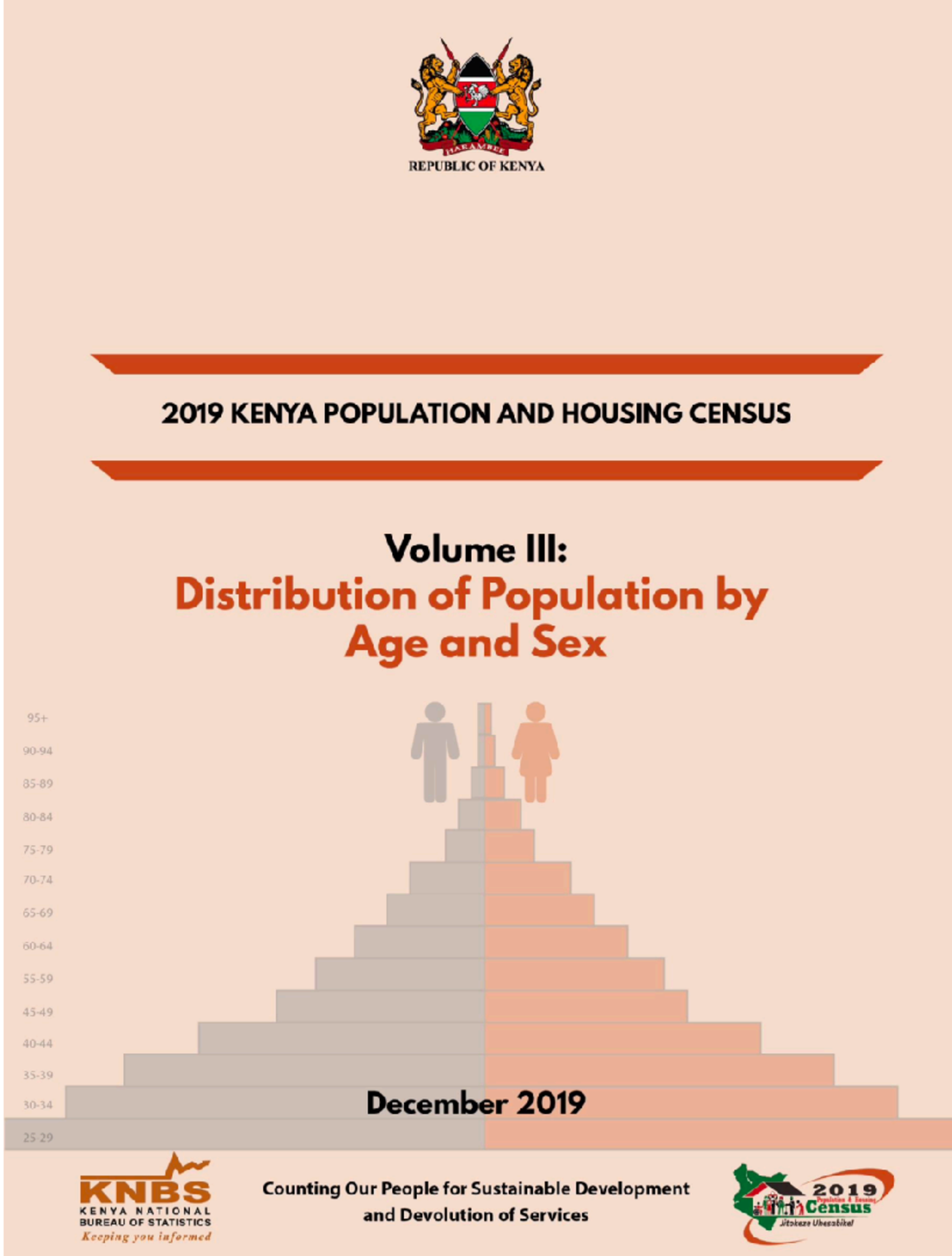
PDFS ARE FOR HUMANS

The distribution of population by age, sex, and administrative unit from the 2019 Kenyan census can be downloaded here:

https://www.knbs.or.ke/?wpdmpro=2019-kenya-population-and-housing-census-volume-iii-distribution-of-population-by-age-sex-and-administrative-units.

It is great that they make it easily available, and it is easy to look-up a particular result.

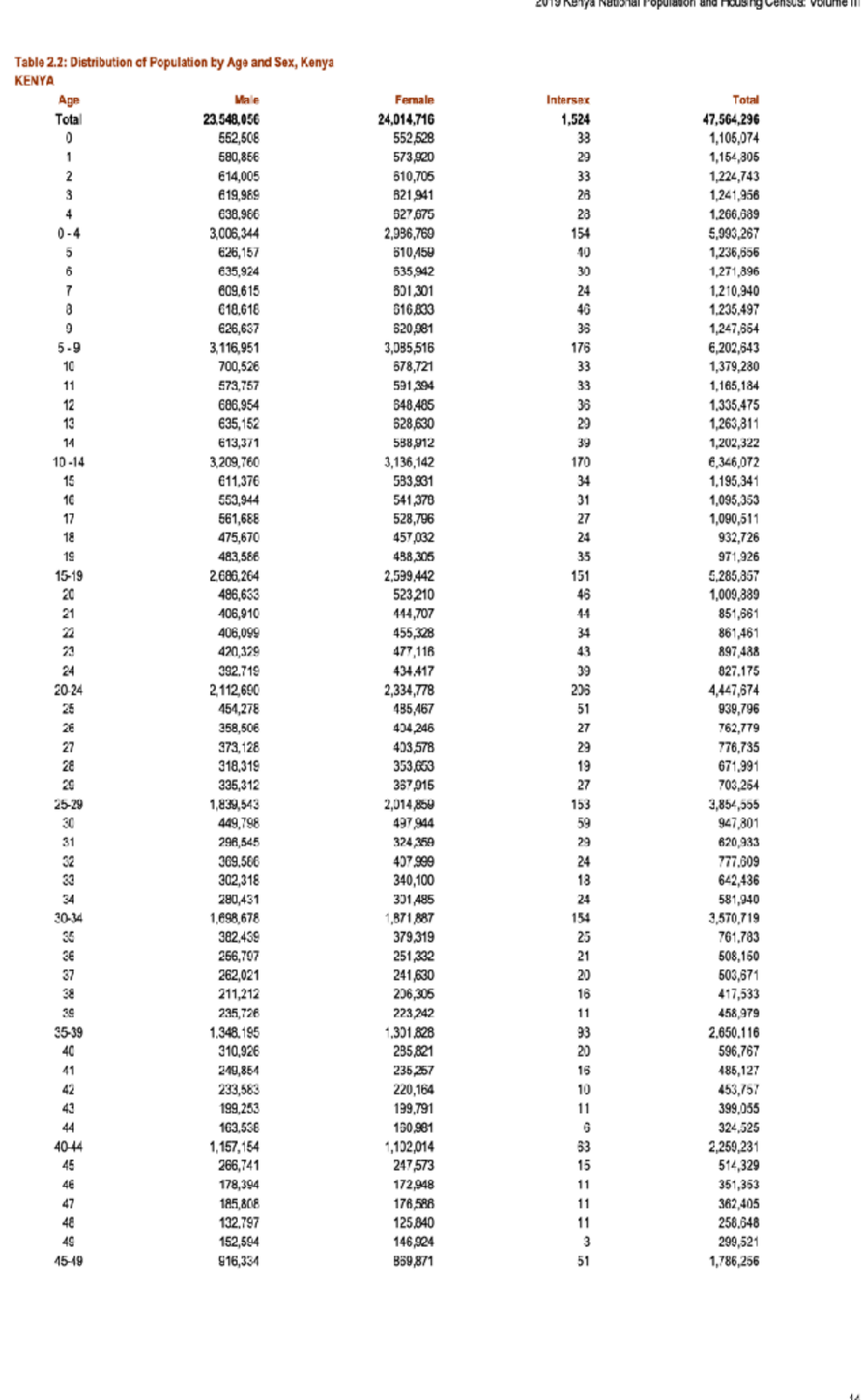
ROHAN ALEXANDER, 22 MAY 2020



PDFS ARE FOR HUMANS

But:

- It is not overly useful to do larger-scale data analysis, such as building a Bayesian hierarchical model.
- We don't know how this PDF was put together so we don't know whether we can trust it.
- We can't manipulate the data to get results that we are interested in.



2019 Kenya National Population and Housing Census: Volume III

Table 2.2: Distribution of Population by Age and Sex, Kenya

KENYA

Age	Male	Female	Intersex	Total
Total	23,548,056	24,014,716	1,524	47,564,296
0	552,508	552,528	38	1,105,074
1	580,856	573,920	29	1,154,805
2	614,005	610,705	33	1,224,743
3	619,969	621,941	28	1,241,956
4	638,986	627,675	23	1,266,689
0 - 4	3,006,344	2,986,769	154	5,993,267
5	626,157	610,459	40	1,236,656
6	635,924	635,942	30	1,271,896
7	609,615	601,301	24	1,210,940
8	618,618	616,833	46	1,235,497
9	626,637	620,981	36	1,247,654
5 - 9	3,116,951	3,085,516	176	6,202,643
10	700,526	678,721	33	1,379,280
11	573,757	591,394	33	1,165,184
12	686,954	648,485	36	1,335,475
13	635,152	628,630	29	1,263,811
14	613,371	588,912	39	1,202,322
10 - 14	3,209,760	3,136,142	170	6,346,072
15	611,376	583,931	34	1,195,341
16	553,944	541,378	31	1,095,353
17	561,688	528,796	27	1,090,511
18	475,670	457,032	24	932,726
19	483,566	488,305	35	971,926
15-19	2,686,264	2,599,442	151	5,285,857
20	486,633	523,210	46	1,009,889
21	406,910	444,707	44	851,661
22	406,099	455,328	34	861,461
23	420,329	477,116	43	897,488
24	392,719	434,417	39	827,175
20-24	2,112,690	2,334,778	206	4,447,674
25	454,278	485,467	51	939,796
26	358,506	404,246	27	762,779
27	373,128	403,578	29	776,735
28	318,319	353,653	19	671,991
29	335,312	367,915	27	703,254
25-29	1,839,543	2,014,859	153	3,854,555
30	449,798	497,944	59	947,801
31	296,545	324,359	29	620,933
32	369,566	407,999	24	777,609
33	302,318	340,100	18	642,436
34	280,431	301,485	24	581,940
30-34	1,698,678	1,871,887	154	3,570,719
35	382,439	379,319	25	761,783
36	256,707	251,332	21	508,150
37	262,021	241,630	20	503,671
38	211,212	206,305	16	417,533
39	235,726	223,242	11	458,979
35-39	1,348,195	1,301,828	93	2,650,116
40	310,926	285,821	20	596,767
41	249,854	235,257	16	485,127
42	233,583	220,164	10	453,757
43	199,253	199,791	11	399,055
44	163,536	150,981	6	324,525
40-44	1,157,154	1,102,014	63	2,259,231
45	266,741	247,573	15	514,329
46	178,394	172,948	11	351,353
47	185,808	176,586	11	362,405
48	132,797	125,840	11	258,648
49	152,594	146,924	3	299,521
45-49	916,334	869,871	51	1,786,256

ROHAN ALEXANDER, 22 MAY 2020

PDFS ARE FOR R!

In this section we convert a PDF of Kenyan census results of counts, by age and sex, by county and sub-county, into a tidy dataset that can be analysed.

I will draw on and introduce a bunch of handy packages including:

- janitor
- pdftools
- tidyverse; and
- stringi.

ROHAN ALEXANDER, 22 MAY 2020

```
> read_csv("/outputs/data/cleaned_dataset.csv")
Parsed with column specification:
cols(
  area = col_character(),
  age = col_character(),
  gender = col_character(),
  number = col_double()
)
# A tibble: 14,382 x 4
   area    age   gender number
   <chr>   <chr> <chr>   <dbl>
 1 Mombasa 0     male    15111
 2 Mombasa 0     female  15009
 3 Mombasa 0     total   30120
 4 Mombasa 1     male    15805
 5 Mombasa 1     female  15308
 6 Mombasa 1     total   31113
 7 Mombasa 2     male    15088
 8 Mombasa 2     female  14837
 9 Mombasa 2     total   29925
10 Mombasa 3     male    14660
# … with 14,372 more rows
```

BEGIN WITH AN END IN MIND.

KEY CONCEPT #1

Planning and then literally sketching out what you want from a final dataset/ graph/paper stops you wasting time and keeps you focused.

SKETCH OUT WHAT YOU WANT

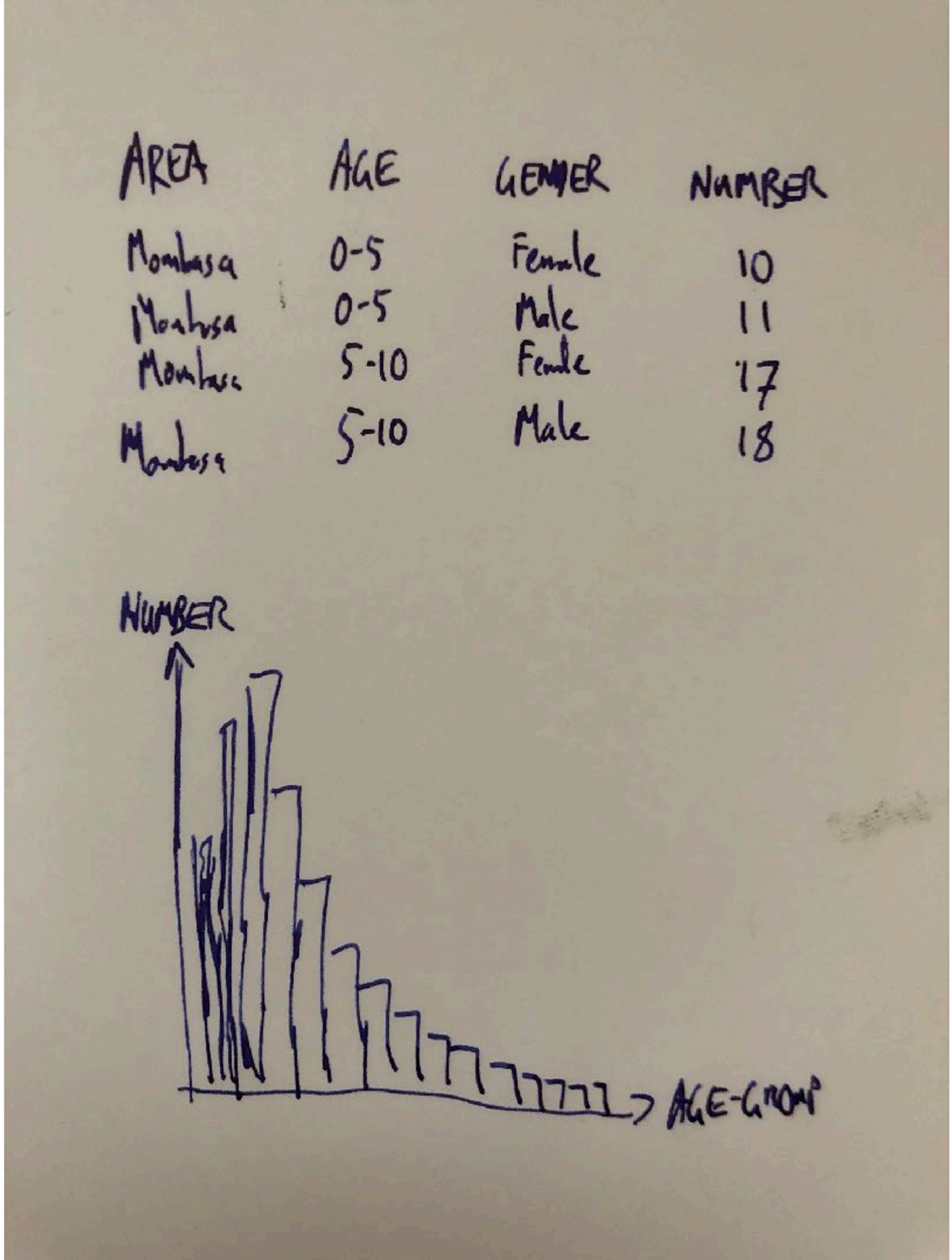
A few quick sketches of an example graph and table will be invaluable.

We need:

- area, age, gender, and then the number of people.

We expect:

- An increase at younger ages, then a decrease



ROHAN ALEXANDER, 22 MAY 2020

WHAT ASPECTS OF PLANNING ARE IMPORTANT?

1. Having a polished output.
2. The planning itself.
3. Fixing a output that will be achieved.
4. All of the above.

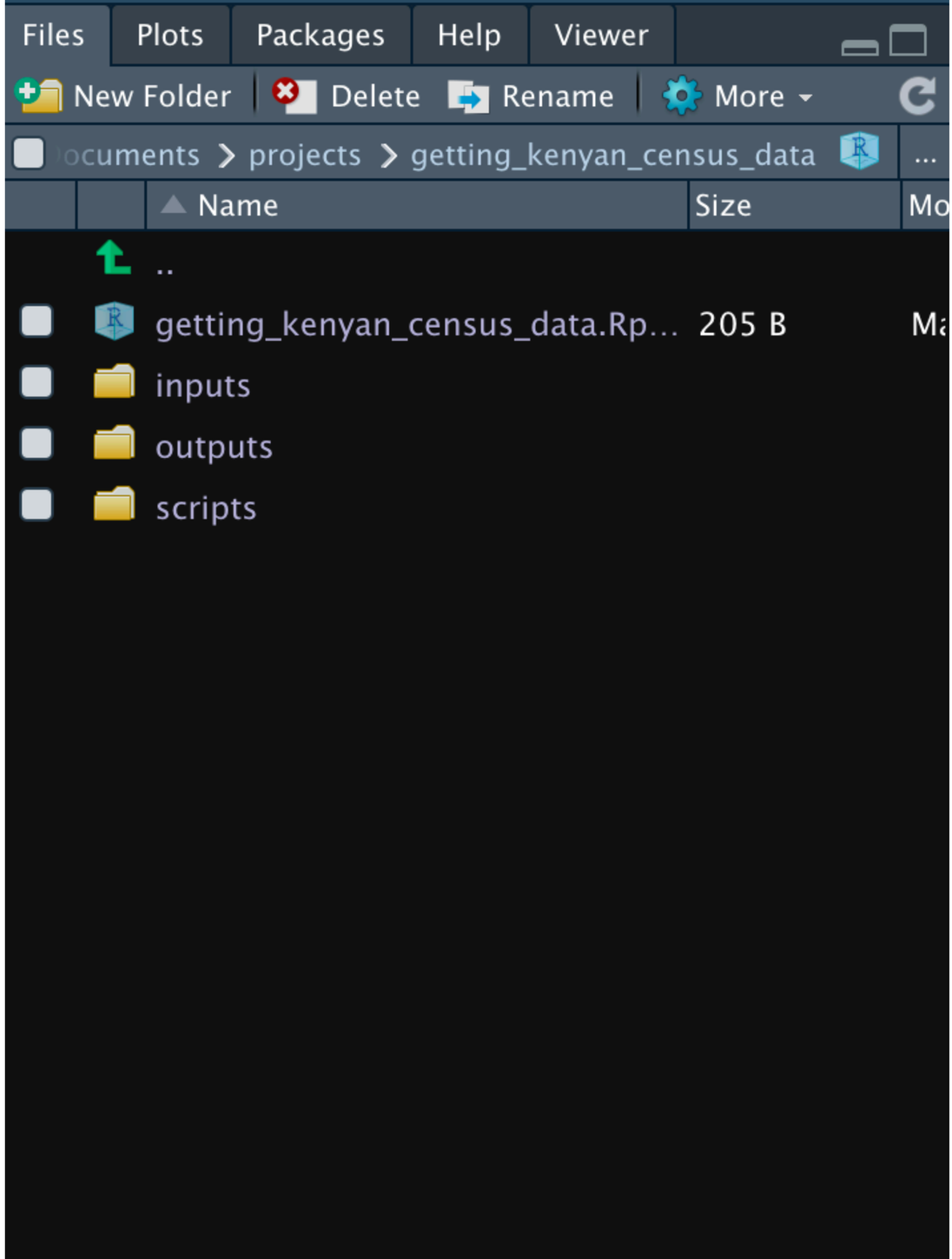
WHAT ASPECTS OF PLANNING ARE IMPORTANT?

1. Having a polished output.
2. **The planning itself.**
3. Fixing a output that will be achieved.
4. All of the above.

SET-UP THE PROJECT

Open R Studio, create a new project called ‘getting_kenyan_census_data’, and in that folder create three sub-folders:

- inputs
- outputs
- scripts



ROHAN ALEXANDER, 22 MAY 2020

ADD THE TOP MATTER

Create a new R Script, and populate some top matter including:

- Purpose
- Author
- Email
- Date

And set-up your workspace by reading in the packages that we need.

READ IN THE PDF

Download the file and save the PDF into the inputs folder. We will never save changes to that PDF.

This enhances reproducibility.

To find the URL that you need navigate through their website: knbs.or.ke.

```
#### Preamble ####
# Purpose: Gather data from the Kenyan census PDF
# Author: Rohan Alexander
# Email: rohan.alexander@utoronto.ca
# Date: 22 May 2020
# Ideas: -
# Issues: -


#### Workspace set-up ####
library(janitor)
library(pdftools)
library(tidyverse)
library(stringi)


#### Get the data ####
download.file(url = "PASTE_THE_URL_HERE",
              destfile = "inputs/kenya_census.pdf")
```

START SIMPLE, THEN ITERATE.

KEY CONCEPT #2

The quickest way to make a complicated model is often to first build a simple model and then complicate it.

READ IN THE PDF AND START SIMPLE

- Read in the PDF using pdftools::pdf_text(),
- Call it ‘all_pages’,
- Then grab page 26 using stringi::stri_split_lines
- Call it page_26.

```
#### Preamble ####
# Purpose: Gather data from the Kenyan census PDF
# Author: Rohan Alexander
# Email: rohan.alexander@utoronto.ca
# Date: 22 May 2020
# Ideas: -
# Issues: -


#### Workspace set-up ####
library(janitor)
library(pdftools)
library(tidyverse)
library(stringi)


#### Get the data ####
download.file(url = "PASTE_THE_URL_HERE",
              destfile = "inputs/kenya_census.pdf")


#### Create a sample ####
all_pages <- pdftools::pdf_text("inputs/kenya_census.pdf")

page_26 <- stringi::stri_split_lines(all_pages[[PASTE_THE_PAGE_HERE]])[[1]]
```

MOVING TO A RECTANGULAR DATASET

- Looking at that page, the pdftools package has done a pretty good job.
- But it is a character vector and most of our experience is with rectangular datasets so we want to make it into a tibble.
- We are going to exploit regularities in the dataset to do this.
    1. Remove the headings and page number
    2. Use the multiple spaces to separate columns

```
> page_26
[1] "                                                                                                2019 Kenya National Population and Housing Census: Volume III"
[2] "Table 2.2: Distribution of Population by Age and Sex, Kenya"
[3] "KENYA"
[4] "        Age                          Male              Female    Intersex                  Total"
[5] "        Total                  23,548,056              24,014,716     1,524             47,564,296"
[6] "          0                       552,580                 552,520        36              1,105,074"
[7] "          1                       580,856                 573,920        29              1,154,805"
[8] "          2                       614,005                 610,705        33              1,224,743"
[9] "          3                       619,939                 621,941        26              1,241,956"
[10] "         4                       638,986                 627,675        28              1,266,689"
[11] "       0-4                     3,006,344               2,986,769       154              5,993,267"
[12] "         5                       626,157                 610,459        40              1,236,656"
[13] "         6                       635,924                 635,942        30              1,271,896"
[14] "         7                       609,615                 601,301        24              1,210,940"
[15] "         8                       618,618                 616,833        46              1,235,497"
[16] "         9                       626,637                 620,981        36              1,247,654"
[17] "       5-9                     3,116,951               3,085,516       176              6,202,643"
[18] "        10                       700,526                 678,721        33              1,379,280"
[19] "        11                       573,757                 591,394        33              1,165,184"
[20] "        12                       686,954                 648,485        36              1,335,475"
[21] "        13                       635,152                 628,630        29              1,263,811"
[22] "        14                       613,371                 588,912        39              1,202,322"
[23] "     10 -14                    3,209,760               3,136,142       170              6,346,072"
[24] "        15                       611,376                 583,931        34              1,195,341"
[25] "        16                       553,944                 541,378        31              1,095,353"
[26] "        17                       561,688                 528,796        27              1,090,511"
[27] "        18                       475,670                 457,032        24                932,726"
[28] "        19                       483,586                 488,305        35                971,926"
[29] "     15-19                    2,686,264               2,599,442       151              5,285,857"
[30] "        20                       486,633                 523,210        46              1,009,889"
[31] "        21                       406,910                 444,707        44                851,661"
[32] "        22                       406,099                 455,328        34                861,461"
[33] "        23                       420,329                 477,116        43                897,488"
[34] "        24                       392,719                 434,417        39                827,175"
[35] "     20-24                    2,112,690               2,334,778       206              4,447,674"
[36] "        25                       454,278                 485,467        51                939,796"
[37] "        26                       358,596                 404,246        27                762,779"
[38] "        27                       373,128                 403,578        29                776,735"
[39] "        28                       318,319                 353,653        19                671,991"
[40] "        29                       335,312                 367,915        27                703,254"
[41] "     25-29                    1,839,543               2,014,859       153              3,854,555"
[42] "        30                       449,798                 497,944        59                947,801"
[43] "        31                       296,545                 324,359        29                620,933"
[44] "        32                       369,586                 407,999        24                777,609"
[45] "        33                       302,318                 340,100        18                642,436"
[46] "        34                       280,431                 301,485        24                581,940"
[47] "     30-34                    1,698,678               1,871,887       154              3,570,719"
[48] "        35                       382,439                 379,319        25                761,783"
[49] "        36                       256,797                 251,332        21                508,150"
[50] "        37                       262,021                 241,630        20                503,671"
[51] "        38                       211,212                 206,305        16                417,533"
[52] "        39                       235,726                 223,242        11                458,979"
[53] "     35-39                    1,348,195               1,301,828        93              2,650,116"
[54] "        40                       310,926                 285,821        20                596,767"
[55] "        41                       249,854                 235,257        16                485,127"
[56] "        42                       233,583                 220,164        10                453,757"
[57] "        43                       199,253                 199,791        11                399,055"
[58] "        44                       163,538                 160,981         6                324,525"
[59] "     40-44                    1,157,154               1,102,014        63              2,259,231"
[60] "        45                       266,741                 247,573        15                514,329"
[61] "        46                       178,394                 172,948        11                351,353"
[62] "        47                       185,808                 176,586        11                362,405"
[63] "        48                       132,797                 125,840        11                258,648"
[64] "        49                       152,594                 146,924         3                299,521"
[65] "     45-49                       916,334                 869,871        51              1,786,256"
[66] "                                                                                                  14"
[67] ""
```

GET RID OF TOP AND BOTTOM AND CONVERT

- Identify the rows that you want to remove and then remove them.

```
> page_26
 [1] "                                                                                2019 Kenya National Population and Housing Census: Volume III"
 [2] "Table 2.2: Distribution of Population by Age and Sex, Kenya"
 [3] "KENYA"
 [4] "        Age                          Male            Female    Intersex                Total"
 [5] "        Total                23,548,056            24,014,716       1,524           47,564,296"
 [6] "          0                     552,508               552,528          38            1,105,074"
 [7] "          1                     580,856               573,920          28            1,154,805"
```

```
[62] "         47                     185,808               176,586          11              362,405"
[63] "         48                     132,797               125,840          11              258,648"
[64] "         49                     152,594               146,924           3              299,521"
[65] "       45-49                    916,334               869,871          51            1,786,256"
[66] "                                                                                              14"
[67] ""
```

- Convert the character vector into a tibble

```
#### Preamble ####
# Purpose: Gather data from the Kenyan census PDF
# Author: Rohan Alexander
# Email: rohan.alexander@utoronto.ca
# Date: 22 May 2020
# Ideas: -
# Issues: -


#### Workspace set-up ####
library(janitor)
library(pdftools)
library(tidyverse)
library(stringi)


#### Get the data ####
download.file(url = "PASTE_THE_URL_HERE",
              destfile = "inputs/kenya_census.pdf")


#### Create a sample ####
all_pages <- pdftools::pdf_text("inputs/kenya_census.pdf")

page_26 <- stringi::stri_split_lines(all_pages[[PASTE_THE_PAGE_HERE]])[[1]]

# Get rid of the top matter
page_26_no_header <- page_26[PASTE_THE_FIRST_ROW_OF_INTEREST_HERE:length(page_26)]

# Get rid of the bottom matter
page_26_no_header_no_footer <- page_26_no_header[1:PASTE_THE_LAST_ROW_OF_INTEREST_HERE]

# Convert into a tibble
demography_data <- tibble(all = page_26_no_header_no_footer)
```

SPLIT INTO COLUMNS

- We exploit the fact that (with a few exceptions that we first correct) if there is a space then we want a new column.

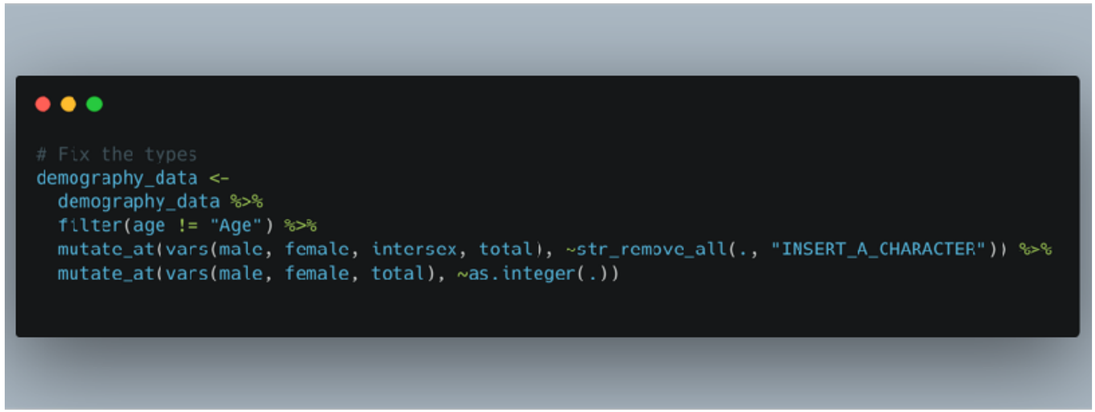
	age	male	female	intersex	total
1	Age	Male	Female	Intersex	Total
2	Total	23,548,056	24,014,716	1,524	47,564,296
3	0	552,508	552,528	38	1,105,074
4	1	580,856	573,920	29	1,154,805
5	2	614,005	610,705	33	1,224,743
6	3	619,989	621,941	26	1,241,956
7	4	638,986	627,675	28	1,266,689

```
# Split columns
demography_data <-
  demography_data %>%
  mutate(all = str_squish(all)) %>% # Any space more than two spaces is
squished down to one
  mutate(all = str_replace(all, "10 -14", "10-14")) %>%
  mutate(all = str_replace(all, "Not Stated", "NotStated")) %>% # Any
space more than two spaces is squished down to one
  separate(col = all,
           into = c("age", "male", "female", "intersex", "total"),
           sep = " ", # Just looking for a space. Seems to work fine
because the tables are pretty nicely laid out
           remove = TRUE,
           fill = "right"
  )
```

ROHAN ALEXANDER, 22 MAY 2020

FIX THE CLASS

- One thing to note is that the ‘numbers’ are still characters. So we want to change them to integers.
- Before we can use as.integer() we need to remove any characters that would cause an NA during this conversion.
- What characters do you think could cause a problem?
- After this, you can save the dataset to the outputs folder.



```
# Fix the types
demography_data <-
  demography_data %>%
  filter(age != "Age") %>%
  mutate_at(vars(male, female, intersex, total), ~str_remove_all(., "INSERT_A_CHARACTER")) %>%
  mutate_at(vars(male, female, total), ~as.integer(.))
```

WHICH OF THE FOLLOWING CHARACTERS DO YOU THINK WILL CAUSE A PROBLEM FOR AS.INTEGER()?

1. 1,000
2. 1-000
3. 1000.00
4. 1OOO

WHICH OF THE FOLLOWING CHARACTERS DO YOU THINK WILL CAUSE A PROBLEM FOR AS.INTEGER()?

1. **1,000**
2. **1-000**
3. 1000.00
4. **1000**

PLEASE PUT THE FOLLOWING INTO ORDER (1MIN)

1. Clean the character vectors
2. Convert character vectors to a tibble
3. Convert PDF to character vectors
4. Fix the classes
5. Get PDF
6. Load libraries
7. Save the cleaned dataset
8. Separate the columns

1. Load libraries
2. Get PDF
3. Convert PDF to character vectors
4. Clean the character vectors
5. Convert character vectors to a tibble
6. Separate the columns
7. Fix the classes
8. Save the cleaned dataset

ROHAN ALEXANDER, 22 MAY 2020

BREAK-OUT GROUPS (10 MIN)

- **We cleaned one page, but there are many pages. One way to deal with this is to change our code into a function.**
- **In R a function has the following form:**

```
function_name <- function(function_inputs) {

function behaviour

 return(output)

}
```

- **We will now form break-out groups and you have 10 minutes to work as a small team to convert your code into something that can loop over all the pages of interest.**
- **Hints:**
  - **map_dfr() takes a function, applies it to a list and then combines all of the outputs into a tibble.**
  - **The pages of interest are pages 30 to 513.**