Week 8 Solutions

Okay, we had a set of quite difficult tasks to do last week. I hope you attempted at least some of them. These tasks were best suited as a continuation of your week 8 handout, so I'm assuming here that you've run all of the code from that handout and are doing the tasks after having run that. As a reminder, I'll repeat the task before giving a solution to each one:

> Repeat the left_join we did earlier, but this time try joining the spatial data to the data where we use generative AI to create short variable names (hint: you might come across some difficulties with the key variable, check the lecture slides about how you join data with two different names for the key!)

In this case I'll use the example of the AI-generated short names that I got on the handout. The trick to this task is that, unless you're very very lucky, the generative AI model probably created a name for your "key" variable that was different to the one in the westminster data. As a reminder, in the westminster dataset the "key" variable was called CODE, but in the AI generated names (for me anyway) it was called geo_code.

That means, in order to join the westminster spatial data to the age_structure data with the AI generated labels you needed to tell it that the codes had two different labels. The clue to doing this was in the lecture slides. Something like the following would work:

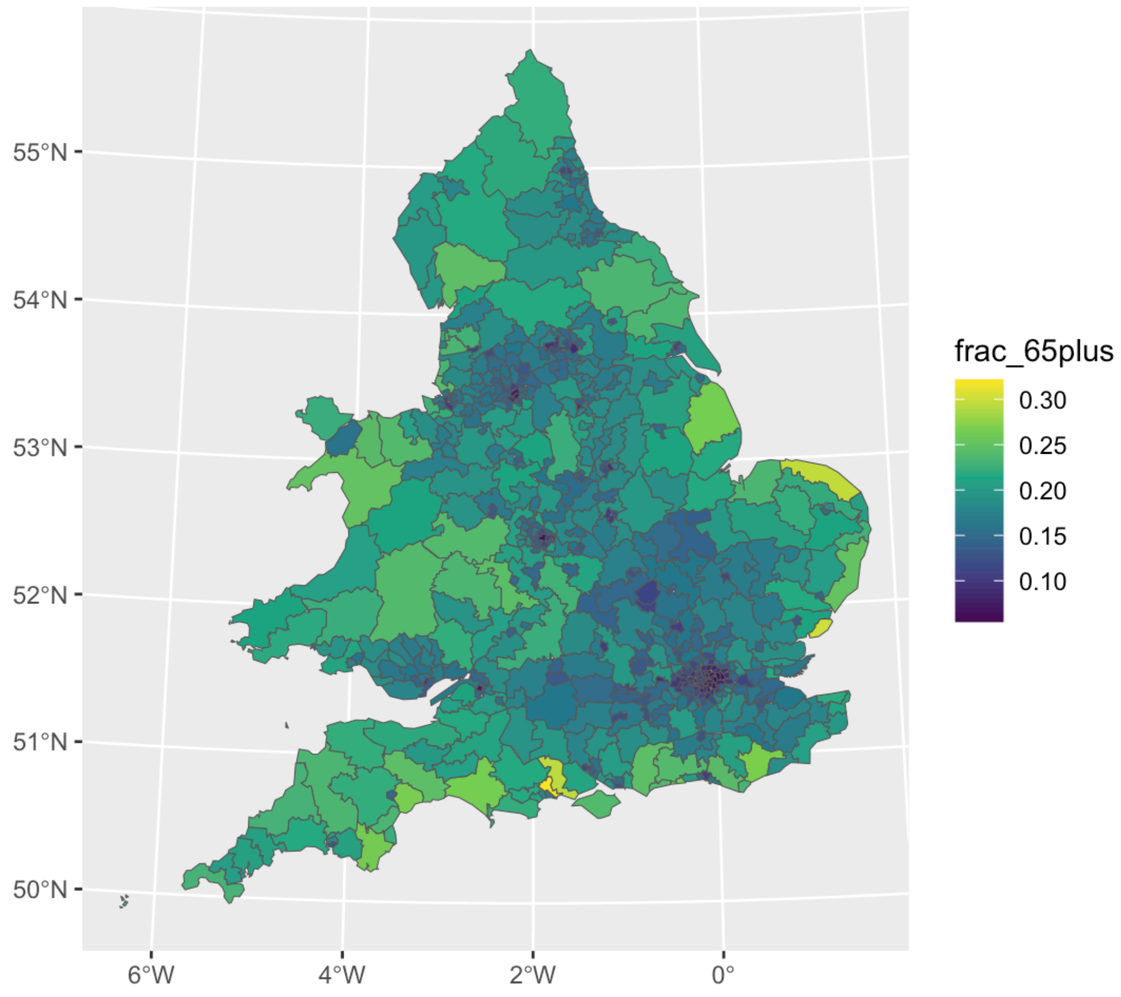
```
age_with_map_ai <-
  left_join(westminster, age_structure_ai, by = c("CODE"="geo_code"))
```

In other words, CODE on the left (westminster) is equal to geo_code on the right (age_structure_ai).

> please draw a map of the fraction of people in each parliamentary constituency who are 65 or older. Which constituencies in England and Wales have the smallest fractions of people aged 65+; which have the largest?

In this case, we didn't have a variable that contained all of the people aged 65 and over, but we did have variables for the number of people between 65 and 74, between 75 and 84, and so on. So, in order to complete this task, we know we're going to need to create at least one new variable that represents the number of people aged over 65, and then we're going to want to divide that by the size of the population as a whole to make a fraction:

```
age_with_map %>%
  drop_na(population) %>%
  mutate(
    frac_65plus = (between65and74+between75and84+between85and89+between90andmore)/
population
  ) %>%
  ggplot() +
  aes(fill = frac_65plus) +
  geom_sf() +
  scale_fill_viridis_c()
```



Note that in this case we need to use addition (add across columns), the function `sum` wouldn't work because it refers only to adding up down rows.

That's the first part of the task complete. The second part asked us to find out which constituencies have the smallest and largest fraction of people aged over 65. To do that we can use arrange:

```
age_with_map %>%
  mutate(
    frac_65plus = (between65and74+between75and84+between85and89+between90andmore)/
population
  ) %>%
  select(frac_65plus, NAME, CODE) %>%
  arrange(frac_65plus)
```

```
## Simple feature collection with 632 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 7458.5 ymin: 7638.9 xmax: 655989 ymax: 1218628
## Projected CRS: OSGB36 / British National Grid
## # A tibble: 632 × 4
##    frac_65plus NAME                           CODE                  geometr
y
##          <dbl> <chr>                          <chr>        <MULTIPOLYGON [m]
>
##  1      0.0548 Poplar and Limehouse Boro Const   E140… (((536703.3 180460.2, 5
3…
##  2      0.0654 East Ham Boro Const               E140… (((540948 179412.2, 539
6…
##  3      0.0680 Bethnal Green and Bow Boro Const  E140… (((533743.7 181261.4, 5
3…
##  4      0.0682 West Ham Boro Const               E140… (((539618.9 180235.1, 5
3…
##  5      0.0686 Hackney North and Stoke Newingto… E140… (((533408.8 187982.6, 5
3…
##  6      0.0693 Bermondsey and Old Southwark Bor… E140… (((531335.6 180529.5, 5
3…
##  7      0.0699 Birmingham, Ladywood Boro Const   E140… (((403710 286873.5, 404
2…
##  8      0.0720 Bristol West Boro Const           E140… (((356646.4 172533.3, 3
5…
##  9      0.0727 Vauxhall Boro Const               E140… (((531155.6 180672.7, 5
3…
## 10      0.0729 Hackney South and Shoreditch Bor… E140… (((532965.3 184818.6, 5
3…
## # ℹ 622 more rows
```

So, the constituencies with the lowest proportion of people aged 65 and over were in Poplar and Limehouse, East Ham, and Bethnal Green.

```
age_with_map %>%
  mutate(
    frac_65plus = (between65and74+between75and84+between85and89+between90andmore)/
population
  ) %>%
  select(frac_65plus, NAME, CODE) %>%
  arrange(desc(frac_65plus))
```

```
## Simple feature collection with 632 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 7458.5 ymin: 7638.9 xmax: 655989 ymax: 1218628
## Projected CRS: OSGB36 / British National Grid
## # A tibble: 632 × 4
##    frac_65plus NAME                          CODE                       geometr
y
##          <dbl> <chr>                         <chr>           <MULTIPOLYGON [m]
>
##  1       0.322 Christchurch Co Const         E14000638 (((404634.2 99172.7, 40
3…
##  2       0.302 Clacton Co Const              E14000642 (((607611.4 215627.3, 6
1…
##  3       0.296 North Norfolk Co Const        E14000848 (((591964.8 341209.8, 5
8…
##  4       0.294 New Forest West Co Const      E14000828 (((422037 98369.7, 4256
7…
##  5       0.269 Bexhill and Battle Co Const   E14000557 (((567948.2 134262.4, 5
6…
##  6       0.267 East Devon Co Const           E14000678 (((300161.8 80250.6, 29
8…
##  7       0.266 West Dorset Co Const          E14001031 (((353972.3 85784.8, 34
4…
##  8       0.265 Worthing West Boro Const      E14001055 (((508521 105284.7, 511
1…
##  9       0.264 Louth and Horncastle Co Const E14000798 (((551810.3 367928.1, 5
5…
## 10       0.263 Totnes Co Const               E14001001 (((273572.1 37488.99, 2
7…
## # ℹ 622 more rows
```

The places with the highest proportion were in Christchurch, Clacton (one of the few places where Reform UK won a seat — Nigel Farage's), and North Norfolk.

> please draw a scatterplot of the percentage of people in each constituency aged over 65, and of the percentage of voters who voted for Reform UK in 2024 (hint: you had a copy of that data in your assessment — check whether it has a variable you can use as a key to join it to this data!)

Again, this task is testing your ability to read in data and to join it using two different codes. I downloaded a copy of the ge_2024.csv file that was on the assessment pages and copied it into my R project directory:

```
ge2024 <- read_csv("ge_2024.csv")
```

```
## Rows: 650 Columns: 42
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr (12): ons_id, ons_region_id, constituency_name, region_name, country_nam...
## dbl (30): electorate, valid_votes, invalid_votes, majority, con, lab, ld, ru...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this messag
e.
```

```
names(ge2024)
```

```
##  [1] "ons_id"                  "ons_region_id"
##  [3] "constituency_name"       "region_name"
##  [5] "country_name"            "constituency_type"
##  [7] "member_first_name"       "member_surname"
##  [9] "member_gender"           "result"
## [11] "first_party"             "second_party"
## [13] "electorate"              "valid_votes"
## [15] "invalid_votes"           "majority"
## [17] "con"                     "lab"
## [19] "ld"                      "ruk"
## [21] "green"                   "snp"
## [23] "pc"                      "dup"
## [25] "sf"                      "sdlp"
## [27] "uup"                     "apni"
## [29] "all_other_candidates"    "con_pc"
## [31] "lab_pc"                  "ld_pc"
## [33] "ruk_pc"                  "green_pc"
## [35] "snp_pc"                  "pc_pc"
## [37] "dup_pc"                  "sf_pc"
## [39] "sdlp_pc"                 "uup_pc"
## [41] "apni_pc"                 "all_other_candidates_pc"
```

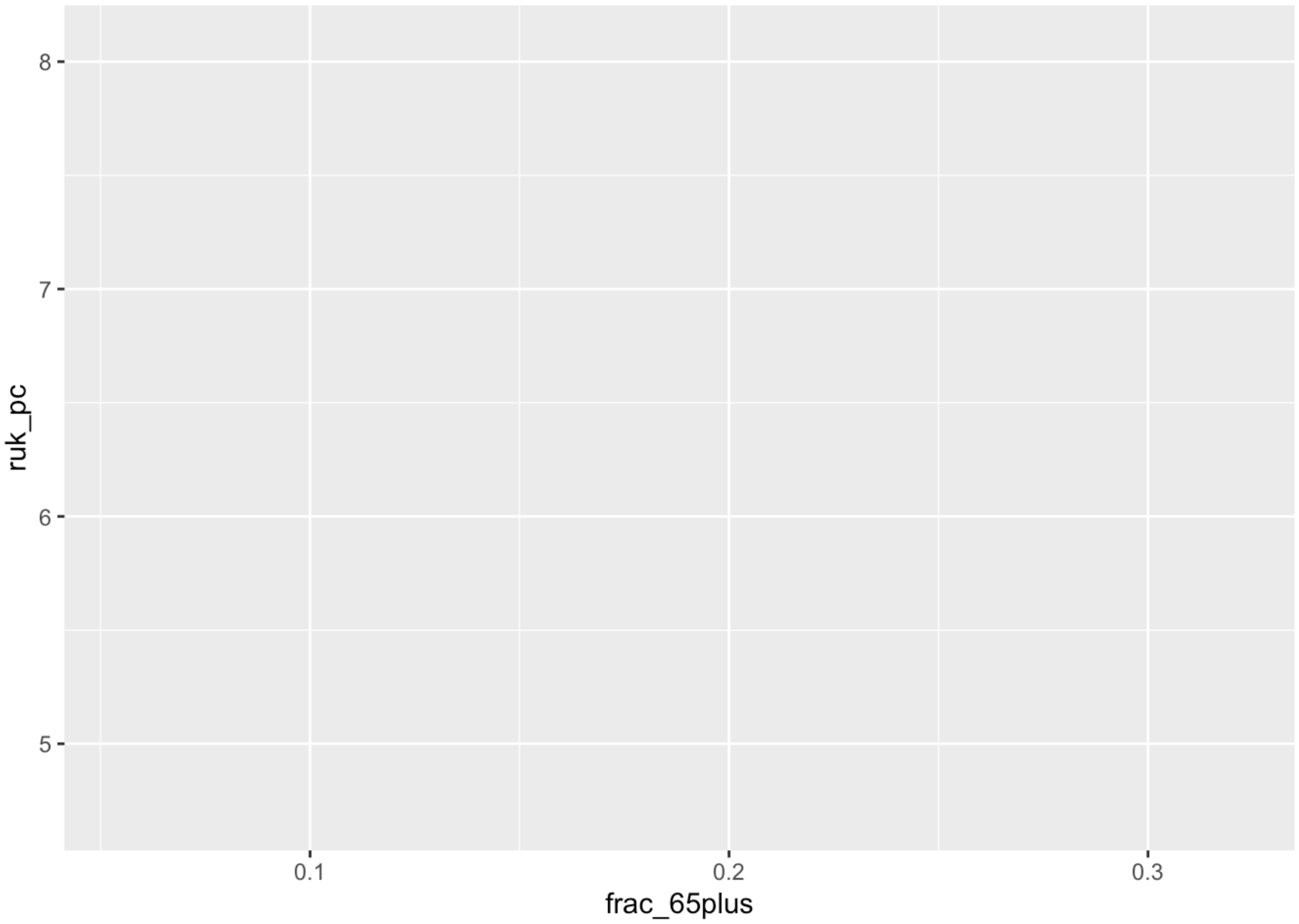
Now I can try and left_join that data. I can see from this file that the key I want to join the data by is called "ons_id", I don't want "one_region_id" because that refers to region rather than parliamentary constituency. I'd expect some of these to be missing due to the mismatch between the 2024 election boundaries and the census boundaries.

```
age_with_map_votes <- left_join(age_with_map, ge2024, by = c("CODE" = "ons_id"))
```

Now that the data has been joined, I can recreate my fraction of people aged 65+ variable and then create a scatterplot of that variable against the proportion of people who voted for Reform UK (ruk_pc):

```
age_with_map_votes %>%
  mutate(
    frac_65plus = (between65and74+between75and84+between85and89+between90andmore)/
population
  ) %>%
  ggplot() +
  aes(x = frac_65plus, y = ruk_pc) +
  geom_point()
```

```
## Warning: Removed 632 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



Oh no, what's happened? Where is our scatterplot? If you read the warning message, we can see that R has removed 632 rows of data because they were missing information about one or the other data.

Okay, so this task was a little more deceptive than it first sounded… How can we map these together? In 2024 the parliamentary constituencies all changed boundaries, so they call got a new set of codes. What we need to do is first search for what is called a "lookup" file — a file that tells us which of the 2024 parliamentary constituencies match up closest to the 2010 ones. There is just such a lookup file here called "Westminster PCON (May 2010) to Westminster PCON (July 2024) Lookup in the UK (V2)": https://geoportal.statistics.gov.uk/datasets/c776b66c0e534b849cae5a5121b7a16a_0/explore (https://geoportal.statistics.gov.uk/datasets/c776b66c0e534b849cae5a5121b7a16a_0/explore)

If we download a copy of that file and move it into our project directory, we can read it into R as follows (I renamed the file to lookup.csv to make the typing a bit easier):

```
lookup <- read_csv("lookup.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for d
etails,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 1407 Columns: 7
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr (4): PCON10CD, PCON10NM, PCON24CD, PCON24NM
## dbl (1): ObjectId
## lgl (2): PCON10NMW, PCON24NMW
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this messag
e.
```

```
names(lookup)
```

```
## [1] "PCON10CD"  "PCON10NM"  "PCON10NMW" "PCON24CD"  "PCON24NM"  "PCON24NMW"
## [7] "ObjectId"
```

```
head(lookup)
```

```
## # A tibble: 6 × 7
##   PCON10CD  PCON10NM              PCON10NMW PCON24CD PCON24NM PCON24NMW ObjectI
d
##   <chr>     <chr>                 <lgl>     <chr>    <chr>    <lgl>        <dbl
>
## 1 E14000530 Aldershot             NA        E140010… Aldersh… NA
1
## 2 E14000844 North East Hampshire  NA        E140010… Aldersh… NA
2
## 3 E14000531 Aldridge-Brownhills   NA        E140010… Aldridg… NA
3
## 4 E14001012 Walsall South         NA        E140010… Aldridg… NA
4
## 5 E14000532 Altrincham and Sale … NA        E140010… Altrinc… NA
5
## 6 E14000533 Amber Valley          NA        E140010… Amber V… NA
6
```

So, this file has a list of the parliamentary constituency codes from 2010 (PCON10CD) and then the closest matching codes from 2024 (PCON24CD). What we want to do is *add the old codes to the new codes* in our election dataset. Let's start by selecting only those two and giving them slightly more informative names:

```
lookup <- lookup %>%
  select(
    old_code = PCON10CD,
    new_code = PCON24CD
  )
```

Now let's join the old_code's onto our data that contains the new codes:

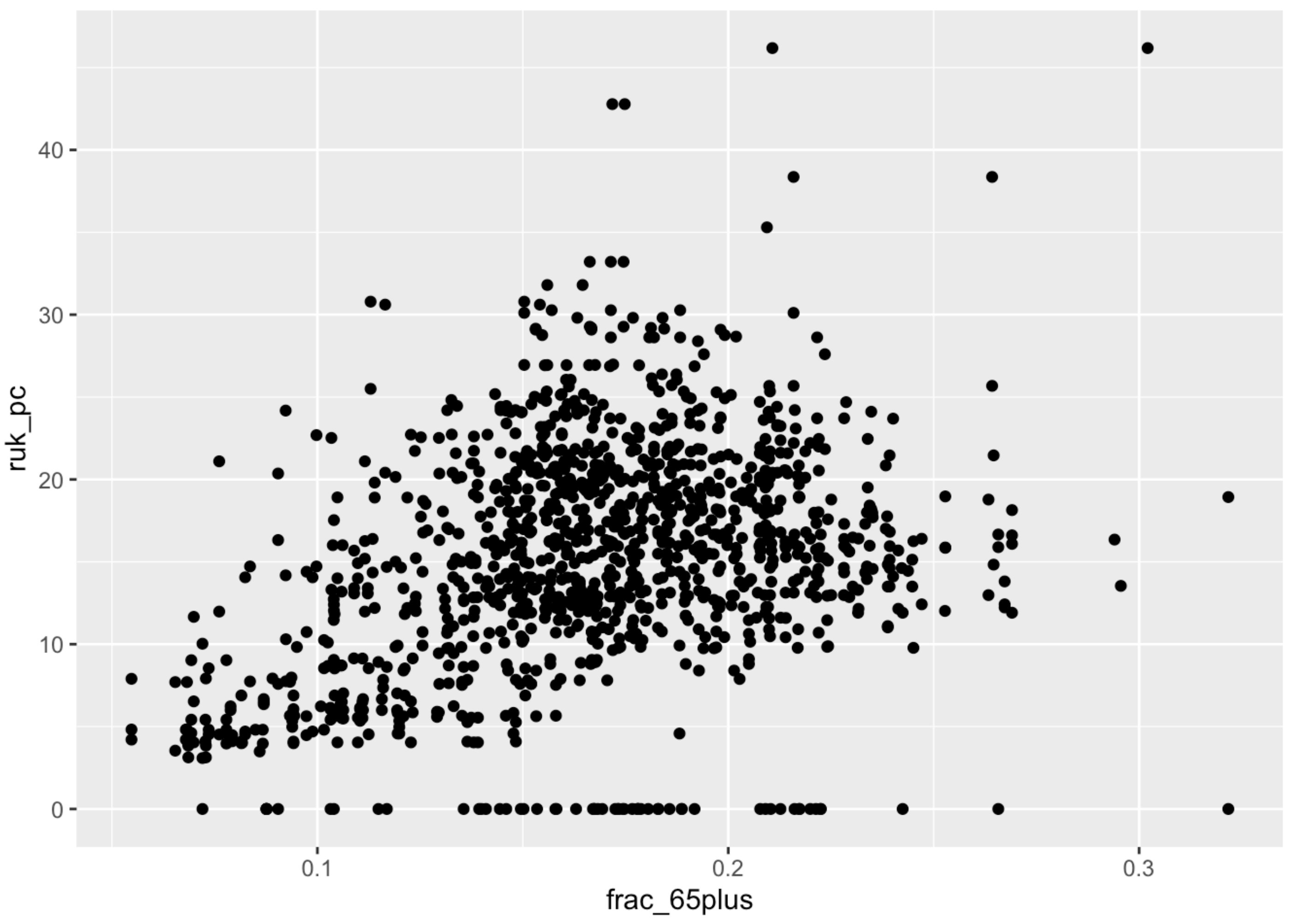
```
ge2024_oldcodes <- left_join(ge2024, lookup, by = c("ons_id" = "new_code"))
```

And finally, let's try re-running our first attempt but this time joining the data using the "old_code" variable in our election date:

```
age_with_map_votes <- left_join(age_with_map, ge2024_oldcodes, by = c("CODE" = "ol
d_code"))

age_with_map_votes %>%
  mutate(
    frac_65plus = (between65and74+between75and84+between85and89+between90andmore)/
population
  ) %>%
  ggplot() +
  aes(x = frac_65plus, y = ruk_pc) +
  geom_point()
```

```
## Warning: Removed 128 rows containing missing values or values outside the scale
range
## (`geom_point()`).
```



Hooray — we finally got there! I don't expect many people to have gotten past the first bit where you saw it deleted all of the rows, but I hope this gives you some insight into the challenges of working with relational data and how useful lookups can be.

> please draw a map of the fraction of people in each constituency who don't have passports (you'll need to find and download this data — excellent practice for your second assessment)

To complete this task, we needed to go back to our Census 2011 Key Statistics page (https://www.nomisweb.co.uk/sources/census_2011_ks (https://www.nomisweb.co.uk/sources/census_2011_ks)) and see if there were any key statistics about passport ownership. You should find table ID KS205EW, where you can download another "bulk.csv" and rename it before adding it to your R project. If we read it in, we get the following (I renamed it to passports.csv):

```
passports <- read.csv("passports.csv")
names(passports)
```

```
##  [1] "date"
##  [2] "geography"
##  [3] "geography.code"
##  [4] "Rural.Urban"
##  [5] "Passports.Held..All.usual.residents..measures..Value"
##  [6] "Passports.Held..No.passport..measures..Value"
##  [7] "Passports.Held..United.Kingdom..measures..Value"
##  [8] "Passports.Held..Republic.of.Ireland..measures..Value"
##  [9] "Passports.Held..Other.Europe..EU.countries..measures..Value"
## [10] "Passports.Held..Other.Europe..Non.EU.countries..measures..Value"
## [11] "Passports.Held..Africa..measures..Value"
## [12] "Passports.Held..Middle.East.and.Asia..measures..Value"
## [13] "Passports.Held..North.America.and.the.Caribbean..measures..Value"
## [14] "Passports.Held..Central.America..measures..Value"
## [15] "Passports.Held..South.America..measures..Value"
## [16] "Passports.Held..Antarctica.and.Oceania..measures..Value"
## [17] "Passports.Held..British.Overseas.Territories..measures..Value"
```

```
head(passports)
```

```
##   date          geography geography.code Rural.Urban
## 1 2011 Berwick-upon-Tweed      E14000554       Total
## 2 2011    Bishop Auckland      E14000569       Total
## 3 2011            Blaydon      E14000574       Total
## 4 2011       Blyth Valley      E14000575       Total
## 5 2011     City of Durham      E14000641       Total
## 6 2011         Darlington      E14000658       Total
##   Passports.Held..All.usual.residents..measures..Value
## 1                                                75718
## 2                                                87143
## 3                                                88281
## 4                                                82174
## 5                                                94375
## 6                                                91417
##   Passports.Held..No.passport..measures..Value
## 1                                        16554
## 2                                        22068
## 3                                        16525
## 4                                        17260
## 5                                        16867
## 6                                        20932
##   Passports.Held..United.Kingdom..measures..Value
## 1                                           57994
## 2                                           64148
## 3                                           70739
## 4                                           64179
## 5                                           73172
## 6                                           67921
##   Passports.Held..Republic.of.Ireland..measures..Value
## 1                                                  167
## 2                                                  132
## 3                                                   117
## 4                                                   100
## 5                                                   364
## 6                                                   210
##   Passports.Held..Other.Europe..EU.countries..measures..Value
## 1                                                          683
## 2                                                          514
## 3                                                          505
## 4                                                          372
## 5                                                         1542
## 6                                                         1458
##   Passports.Held..Other.Europe..Non.EU.countries..measures..Value
## 1                                                               41
## 2                                                               38
## 3                                                               44
## 4                                                               49
## 5                                                              213
## 6                                                               58
##   Passports.Held..Africa..measures..Value
## 1                                       76
## 2                                       90
## 3                                      110
## 4                                       57
## 5                                      389
## 6                                      188
##   Passports.Held..Middle.East.and.Asia..measures..Value
## 1                                                    175
## 2                                                    177
## 3                                                    281
## 4                                                    196
## 5                                                   1643
## 6                                                    674
##   Passports.Held..North.America.and.the.Caribbean..measures..Value
## 1                                                               172
## 2                                                               101
## 3                                                               117
## 4                                                                84
## 5                                                               700
## 6                                                               189
##   Passports.Held..Central.America..measures..Value
## 1                                                 0
## 2                                                 2
## 3                                                 4
## 4                                                 1
## 5                                                13
## 6                                                 2
##   Passports.Held..South.America..measures..Value
## 1                                             18
## 2                                              8
## 3                                              9
## 4                                             10
## 5                                             45
## 6                                              8
##   Passports.Held..Antarctica.and.Oceania..measures..Value
## 1                                                     121
## 2                                                      77
## 3                                                      57
## 4                                                      45
## 5                                                     235
## 6                                                      91
##   Passports.Held..British.Overseas.Territories..measures..Value
## 1                                                             1
## 2                                                             3
## 3                                                             2
## 4                                                             0
## 5                                                            12
## 6                                                             0
```

This has some pretty horrendous names. To save time, I'm just going to select the variables I need and rename them on the go:

```
passports <- passports %>%
  select(
    CODE = geography.code,
    population = Passports.Held..All.usual.residents..measures..Value,
    no_passport = Passports.Held..No.passport..measures..Value
  )
```

I may as well create my new variable, fraction of people who do not have passports, before I join my data to my spatial dataset:

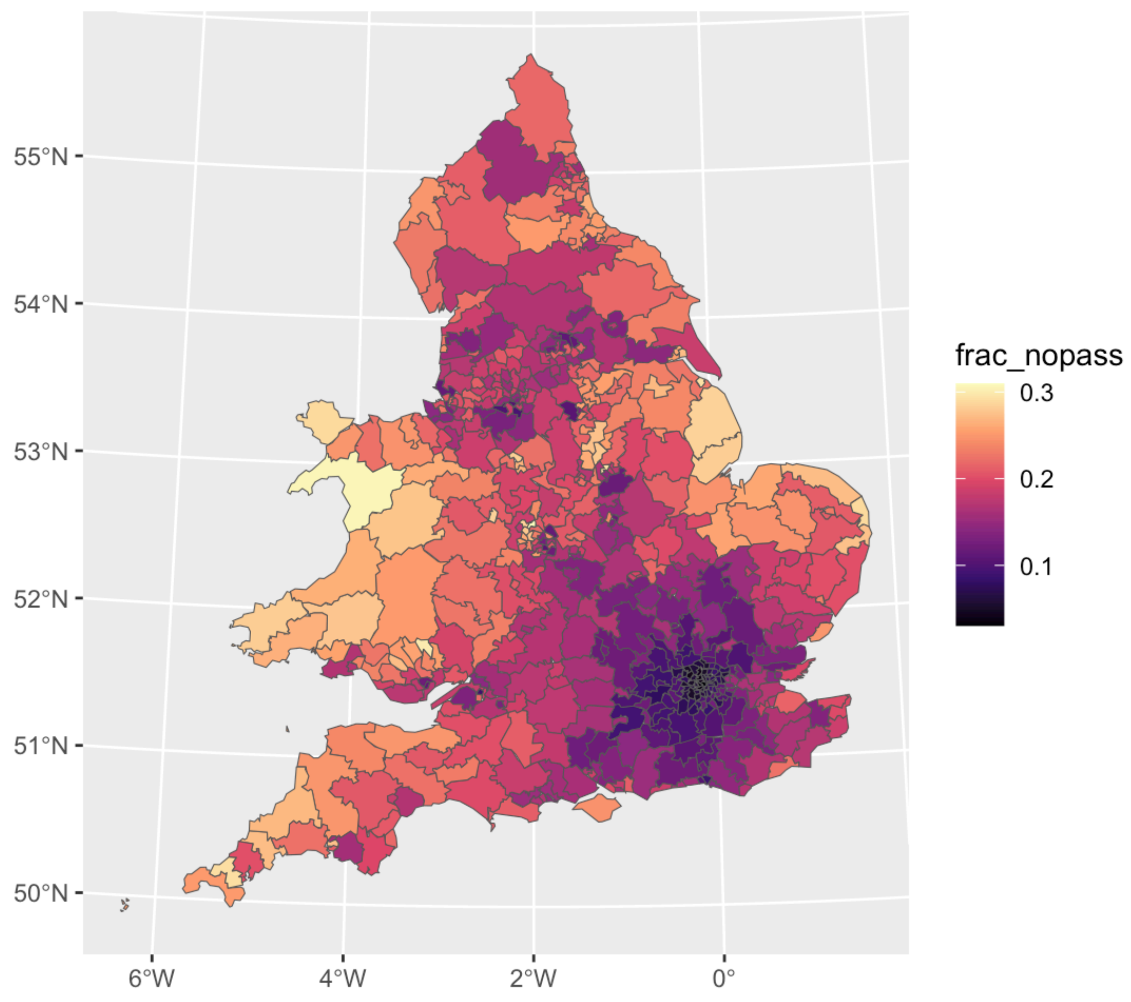
```
passports <- passports %>%
  mutate(
    frac_nopass = no_passport/population
  )
```

Okay, now I can join my data together:

```
age_with_map_passports <- left_join(age_with_map, passports, by="CODE")
```

And finally, create my map:

```
age_with_map_passports %>%
  drop_na(frac_nopass) %>%
  ggplot() +
  aes(fill = frac_nopass) +
  geom_sf() +
  scale_fill_viridis_c(option = "A")
```



What does this tell us about how peoples' ability to travel out of the country differs depending on where they live?

> Recreate one of your maps using the House of Commons Non-Contiguous Cartogram of the UK for constituencies (https://github.com/houseofcommonslibrary/uk-hex-cartograms-noncontiguous (https://github.com/houseofcommonslibrary/uk-hex-cartograms-noncontiguous)? tab=readme-ov-file (https://github.com/houseofcommonslibrary/uk-hex-cartograms-(https://github.com/houseofcommonslibrary/uk-hex-cartograms-) noncontiguous?tab=readme-ov-file)), and post it to Blackboard. This is a "GEOPACKAGE" type mapping file, so you might want to check the lecture slides about this to find out how to read it into R!

Okay, this very last task involved downloading the geopackage file for the House of Commons hex map for the parliamentary constituencies for 2010. This is the "Constituencies.gpkg" file inside the geopackage directory of the House of Commons github pages. If I take that file and move it into my project directory, I can read it in using the following code (adapted from the lecture slides for week 8)…

First, I check which layers are available to use:

```
st_layers("Constituencies.gpkg")
```

```
## Driver: GPKG
## Available layers:
##         layer_name geometry_type features fields                       crs_name
## 1    1 Group names         Point       59      4 OSGB36 / British National Grid
## 2 2 Group outlines Multi Polygon       60      3 OSGB36 / British National Grid
## 3  3 City outlines Multi Polygon       55      4 OSGB36 / British National Grid
## 4 4 Constituencies Multi Polygon      650      6 OSGB36 / British National Grid
## 5     5 Background Multi Polygon        3      2 OSGB36 / British National Grid
## 6     layer_styles            NA        5     12                           <NA>
```

At the very least, what I probably want is the “5 Background” layer and the “4 Constituencies” layer:

```
background <- st_read("Constituencies.gpkg",
                      layer = "5 Background")
```

```
## Reading layer `5 Background' from data source
##   `/Users/calumwebb/Library/Mobile Documents/com~apple~CloudDocs/r/smi105-data_
viz_materials/smi105-teaching-materials/week-08/handout/Constituencies.gpkg'
##   using driver `GPKG'
## Simple feature collection with 3 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 5.987083 ymin: 1.761197 xmax: 56.83708 ymax: 70.35025
## Projected CRS: OSGB36 / British National Grid
```

```
head(background)
```

```
## Simple feature collection with 3 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 5.987083 ymin: 1.761197 xmax: 56.83708 ymax: 70.35025
## Projected CRS: OSGB36 / British National Grid
##   id            Name                           geom
## 1 45 England & Wales MULTIPOLYGON (((47.38708 26...
## 2  2        Scotland MULTIPOLYGON (((32.23708 56...
## 3  3         Ireland MULTIPOLYGON (((10.78708 44...
```

```
constituencies <- st_read("Constituencies.gpkg",
                      layer = "4 Constituencies")
```

```
## Reading layer `4 Constituencies' from data source
##   `/Users/calumwebb/Library/Mobile Documents/com~apple~CloudDocs/r/smi105-data_
viz_materials/smi105-teaching-materials/week-08/handout/Constituencies.gpkg'
##   using driver `GPKG'
## Simple feature collection with 650 features and 6 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 7.337122 ymin: 2.106932 xmax: 55.78812 ymax: 69.48179
## Projected CRS: OSGB36 / British National Grid
```

```
head(constituencies)
```

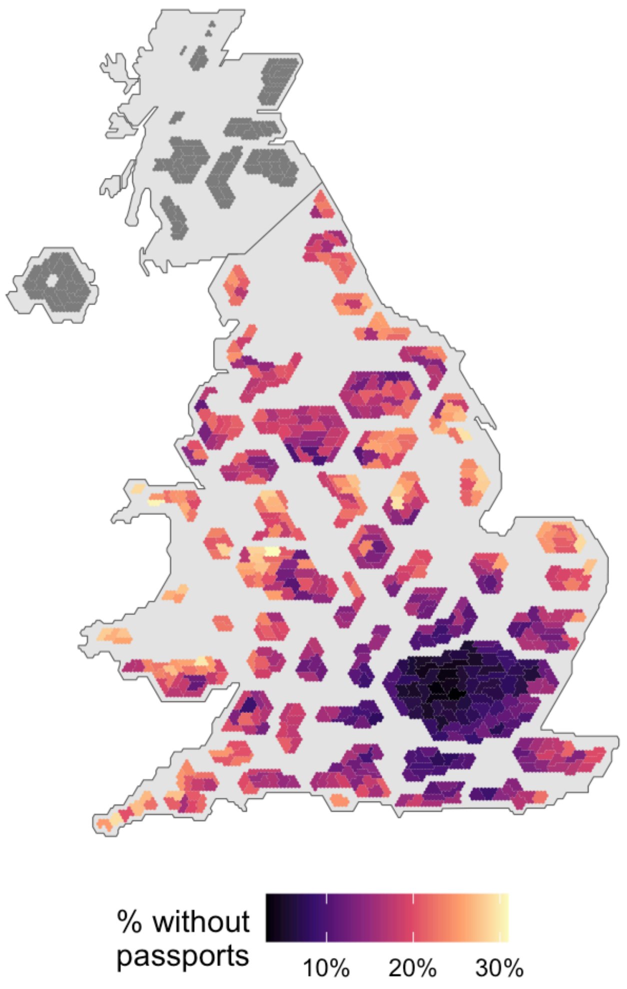
```
## Simple feature collection with 6 features and 6 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 20.53712 ymin: 11.97933 xmax: 46.48812 ymax: 58.31039
## Projected CRS: OSGB36 / British National Grid
##              Group        Group.labe      RegionNati pcon.code
## 1   West of England   West of England      South West E14000602
## 2     West Midlands         West Mids   West Midlands E14000564
## 3     West Midlands         West Mids   West Midlands E14001030
## 4 Lothian & Borders Lothian & Borders        Scotland S14000045
## 5    Cambridgeshire             Cambs East of England E14000855
## 6   Glasgow & Clyde   Glasgow & Clyde        Scotland S14000029
##                   pcon.name  CityLabel                           geom
## 1              Bristol West    Bristol MULTIPOLYGON (((26.53794 12...
## 2      Birmingham, Ladywood Birmingham MULTIPOLYGON (((29.23794 23...
## 3         West Bromwich West       <NA> MULTIPOLYGON (((27.43794 23...
## 4                Midlothian       <NA> MULTIPOLYGON (((28.78794 56...
## 5 North West Cambridgeshire     P'boro MULTIPOLYGON (((46.33782 24...
## 6           Glasgow Central    Glasgow MULTIPOLYGON (((20.98794 57...
```

As we can see, where we want to join our data onto is the constutuencies object. If you check our the first few rows of the data, you can see that we have our key this time in a variable called “pcon.code”. For simplicities sake, let’s just repeat our last map by joining our passports data onto the file:

```
passport_hexmap <- left_join(constituencies, passports, by=c("pcon.code" = "COD
E"))
```

Okay, now let’s try creating our map. Because we have two “layers” here: the background and the “hexes” representing our parliamentary constituencies, we need to use two different datasets within our ggplot code:

```
ggplot() +
  geom_sf(data = background) +
  geom_sf(data = passport_hexmap,
          aes(fill = frac_nopass),
          colour = "transparent") +
  scale_fill_viridis_c(option = "A",
                       labels = scales::label_percent()) +
  theme_void() +
  theme(legend.position = "bottom") +
  labs(fill = "% without\npassports")
```



I've added a couple of other niceities to improve the appearance of the map here. What are the benefits of using this kind of map over the choropleths we've been using so far?

Feedback

If you attempted any of the above tasks — well done! As I said in the workshop, I'd really encourage you to not take an "all or nothing" approach to these tasks. Doing something, even if it's not perfect or doesn't work, will still help you learn and get feedback. The only way we learn to improve is by repeatedly trying, failing, and improving on what we did before. We can't do that if we never start the process. I'd much rather have you try and fail on a weekly task — where it doesn't really matter as it's not graded — than your first time trying and failing being in your graded assessment!

Some specific things people struggled with:

- Remember that your key names either need to be identical across two relational datasets, or you need to use the format for the argument: by = c("keyname1" = "keyname2") when they differ.
- Remember that your spatial data needs to be the first thing that is joined in any kind of join, otherwise the resultant object loses its spatial characteristics.
- To add across columns you need to use + rather than sum, since sum is for adding down rows.
- Remember when you download new data and want to read it you **need to move it into your R project folder**, otherwise R won't be able to find it.