Using R in industry: Whistle stop tour of Public Health reporting using R

Dr Rich Tyler,

Public Health Analyst,

West Sussex County Council


@GeoPsychTy


psychty

What about R for outputs?

- Reasons for using R
- Output examples
  - Infographics
  - Reports
  - Excel outputs (static and interactive)
- Approaches
  - Reading and writing data
  - Strings
  - Loops
  - If statements

- This is not a lesson in R
  - However…
  - Public github repository –
    - https://github.com/psychty

First encounter
MSc Health Psych –
Uni Southampton

PhD research
Uni Portsmouth

Analyst role
West Sussex
County Council

2010 — **2011/12** Forgot about R — 2013/14 — **2015/16** Forgot about R — 2017 — **2018** Forgot about SPSS

Reasons for ~~using R~~ ~~writing code~~ using R ~~for public health~~

- Commenting
- Auditing code
- Sharing analyses and code with others
- Charts, tables, maps, reports all in one place (from farm to fork)
- Multiple reports for subpopulations
- Frequently updating smaller components within a larger analysis
- Minimising ctrl/cmd + c ctrl/cmd + v and other human errors

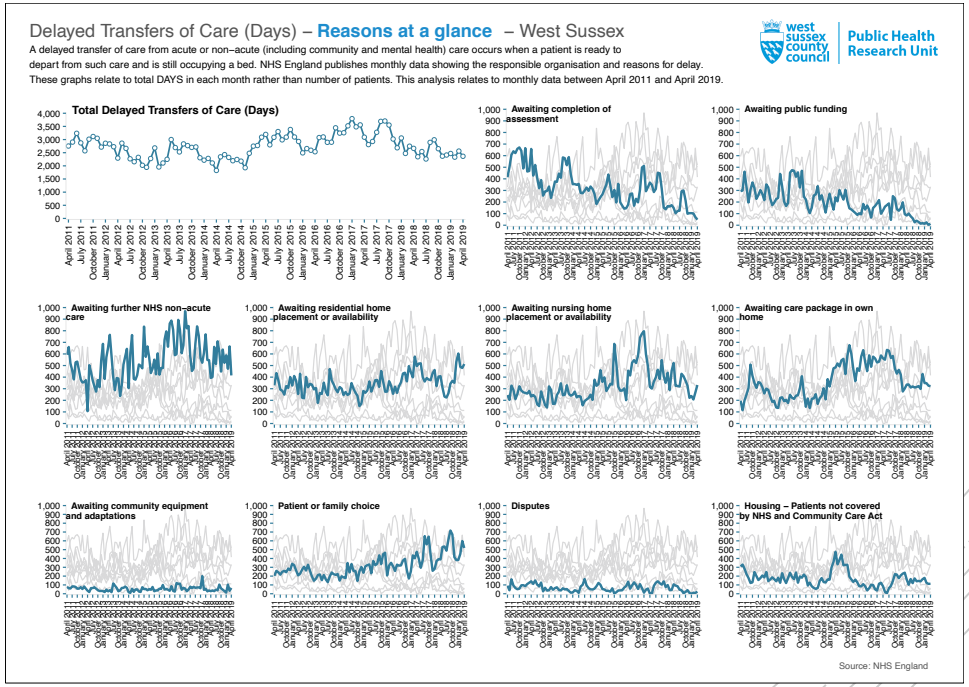
- Getting new software, even open source software, was not straightforward.
- Business case highlighting benefits of flexibility, transparency, auditability, automation, the innovation and development of more sophisticated analysis skills
- Now at least four users in two departments use R for core work
- We have supported colleagues in other local authorities to get started

Delayed Transfers of Care

- NHS England publish a monthly 'situation report' detailing the delayed transfers of care by local authority.
- Each month a new set of data is published as an excel and csv file.
- Output: PDF infographic showing the number of delayed days by reason for delay over time for a single area.

Steps in R:

- Download and compile each month of data into a single file
- Create figures showing the number of days delayed in total and for each reason
- Use grid package to position each figure on an a4 canvas and export as a pdf



- A delayed transfer of care is when a patient (aged 18+) is ready to go home following NHS-funded acute or non-acute care but is still occupying a bed.
- It is important to understand reasons for these delays and how delays change over time.

- Github link - https://github.com/psychty/DTOC

mye and projections.R | Population pyramids.R | 2 - Deprivation.R | Delayed Transfers of Care data coll... | 3 - Proje

Source on Save | Run | Source

```
15  # You can check the directory with getwd()
16  getwd()
17
18  # This is a command to create a sub-directory if it does not exist in the working directory
19  if (!file.exists("~/Delayed Transfers of Care")) {
20    dir.create("~/Delayed Transfers of Care")}
21
22  # This is a command to create a sub-directory if it does not exist in the working directory
23  if (!file.exists("~/Delayed Transfers of Care/Data")) {
24    dir.create("~/Delayed Transfers of Care/Data")}
25
26  # Any new months of data can be downloaded using R, just copy the 'Total Delayed Days Local Authority'
    file for the month and paste it into the download.file("", "./Delayed Transfers of Care/Data/x .xls", mode
    = "wb"). The mode = "wb" saves it as a binary file.
27
28  if (!file.exists("./Delayed Transfers of Care/Data/Apr_19.xls")) {download.file("https://www.england.nhs
    .uk/statistics/wp-content/uploads/sites/2/2019/06/LA-Type-B-April-2019-K35jY.xls","./Delayed Transfers of
    Care/Data/Apr_19.xls", mode = "wb")}
29  if (!file.exists("./Delayed Transfers of Care/Data/Mar_19.xls")) {download.file("https://www.england.nhs
    .uk/statistics/wp-content/uploads/sites/2/2019/05/LA-Type-B-March-2019-9c5JC.xls","./Delayed Transfers of
    Care/Data/Mar_19.xls", mode = "wb")}
30  if (!file.exists("./Delayed Transfers of Care/Data/Feb_19.xls")) {download.file("https://www.england.nhs
    .uk/statistics/wp-content/uploads/sites/2/2019/05/LA-Type-B-February-2019-Fk0FS.xls","./Delayed Transfers
    of Care/Data/Feb_19.xls", mode = "wb")}
31  if (!file.exists("./Delayed Transfers of Care/Data/Jan_19.xls")) {download.file("https://www.england.nhs
    .uk/statistics/wp-content/uploads/sites/2/2019/05/LA-Type-B-January-2019-mxX03.xls","./Delayed Transfers
    of Care/Data/Jan_19.xls", mode = "wb")}
32  if (!file.exists("./Delayed Transfers of Care/Data/Dec_18.xls")) {download.file("https://www.england.nhs
    .uk/statistics/wp-content/uploads/sites/2/2019/05/LA-Type-B-December-2018-JPuaa.xls","./Delayed Transfers
    of Care/Data/Dec_18.xls", mode = "wb")}
33  if (!file.exists("./Delayed Transfers of Care/Data/Nov_18.xls")) {download.file("https://www.england.nhs
    .uk/statistics/wp-content/uploads/sites/2/2019/05/LA-Type-B-November-2018-rDU26.xls","./Delayed Transfers
    of Care/Data/Nov_18.xls", mode = "wb")}
```

Open Link (Command+Click)

227:1 | Clean up Days_organisation | R Script

Console | Terminal | Jobs

~/Dropbox/Work/

>

Environment | History | Connections

Import Dataset | List

Global Environment

Combine if statements with functions to prevent repeating unnecessary code

Really useful when working on different machines or sharing code with others

Files | Plots | Packages | Help | Viewer

Zoom | Export

Remember to use binary mode when downloading files

mye and projections.R | Population pyramids.R | 2 - Deprivation.R | Delayed Transfers of Care data coll... | 3 - Proje

Source on Save | Run | Source

```
129 # Create an empty df ready to put in all the data
130 Days_reason <- data.frame(SHA = factor(), CODE = factor(), NAME = factor(), `A) AWAITING COMPLETION OF
    ASSESSMENT` = numeric(), `B) AWAITING PUBLIC FUNDING` = numeric(), `C) AWAITING FURTHER NON-ACUTE NHS
    CARE` = numeric(), `DI) AWAITING RESIDENTIAL HOME PLACEMENT OR AVAILABILITY` = numeric(), `DII) AWAITING
    NURSING HOME PLACEMENT OR AVAILABILITY` = numeric(), `E) AWAITING CARE PACKAGE IN OWN HOME` = numeric(),
    `F) AWAITING COMMUNITY EQUIPMENT AND ADAPTATIONS` = numeric(), `G) PATIENT OR FAMILY CHOICE` = numeric(),
    `H) DISPUTES` = numeric(), `I) HOUSING - PATIENTS NOT COVERED BY NHS AND COMMUNITY CARE ACT` = numeric(),
    TOTAL = numeric(), PERIOD_YEAR = factor(), check.names = FALSE)
131
132
133 for (i in 1:length(list.files("./Delayed Transfers of Care/Data")))  {
134   Month <- read_excel(paste("./Delayed Transfers of Care/Data", list.files("./Delayed Transfers of Care
    /Data")[i], sep = "/"), sheet = "LA - by reason", skip = 13)
135   Month$PERIOD_YEAR <- as.character(read_excel(paste("./Delayed Transfers of Care/Data", list.files("
    ./Delayed Transfers of Care/Data")[i], sep = "/"), sheet = "LA - by reason", range = "R5C3:R5C3",
    col_names = FALSE))
136   colnames(Month) <- toupper(colnames(Month))
137   colnames(Month) <- gsub("^A.*", "A) AWAITING COMPLETION OF ASSESSMENT", colnames(Month))
138   colnames(Month) <- gsub("^B.*", "B) AWAITING PUBLIC FUNDING", colnames(Month))
139   colnames(Month) <- gsub("^C).*", "C) AWAITING FURTHER NON-ACUTE NHS CARE", colnames(Month))
140   colnames(Month) <- gsub("^F.*", "F) AWAITING COMMUNITY EQUIPMENT AND ADAPTATIONS", colnames(Month))
141   colnames(Month) <- gsub("^I.*", "I) HOUSING - PATIENTS NOT COVERED BY NHS AND COMMUNITY CARE ACT",
    colnames(Month))
142   colnames(Month)[colnames(Month) == "ONS GEOGRAPHY"] <- "CODE"
143   Days_reason <- rbind.fill(Days_reason, Month) # This adds the data to a bigger dataframe called
    Days_reason (which we set up earlier)
144   rm(Month)
145   }
146
147 # Clean up Days_reason ####
148 Days_reason <- subset(Days_reason, !is.na(NAME)) # remove rows with no data for name
149
150 # These commands tidy up some formatting (they are essentially find and replace commands)
151 Days_reason$NAME <- gsub(" UA", "", Days_reason$NAME)
```

145:4 | Days delayed by reason | R Script

Console | Terminal | Jobs

~/Dropbox/Work/

>

Environment | History | Connections

Import Dataset | List

Global Environment

Create a blank dataframe first, then using a loop to append your row x into the larger dataframe.

For every file in the data folder:

1. Create an object called 'Month'
2. Change some of the formatting and column headings
3. Add the Month object to the Days_reason object

DToC reason for delay Days.R

```
jsna = readPNG("./Research Unit.png")

Days_reason <- read_csv("./Delayed Transfers of Care/DToC_Days_Reason_for_Delay_created_June_2019.csv",
col_types = cols(Code = col_character(),  Name = col_character(),  `Awaiting completion of assessment` =
col_double(),  `Awaiting public funding` = col_double(),  `Awaiting further NHS non-acute care` =
col_double(),  `Awaiting residential home placement or availability` = col_double(),  `Awaiting nursing
home placement or availability` = col_double(),  `Awaiting care package in own home` = col_double(),
`Awaiting community equipment and adaptations` = col_double(),  `Patient or family choice` = col_double(),
 Disputes = col_double(),  `Housing - Patients not covered by NHS and Community Care Act` = col_double(),
 Other = col_double(),  `Total Delayed Transfers of Care` = col_double(),  Period = col_character())) %>%
  mutate(Month = as.Date(paste0("01 ", Period), format = "%d %b %Y")) %>%
  arrange(Month)

month_order <-  as.character(unique(Days_reason$Period))

quarter_months <- c(month_order[length(month_order)-96],month_order[length(month_order)-93],month_order[le
ngth(month_order)-90],month_order[length(month_order)-87],month_order[length(month_order)-84],month_order[
length(month_order)-81],month_order[length(month_order)-78],month_order[length(month_order)-75],month_orde
r[length(month_order)-72],month_order[length(month_order)-69],month_order[length(month_order)-66],month_or
der[length(month_order)-63],month_order[length(month_order)-60],month_order[length(month_order)-57]
,month_order[length(month_order)-54],month_order[length(month_order)-51],month_order[length(month_order
)-48],month_order[length(month_order)-45],month_order[length(month_order)-42],month_order[length(month_ord
er)-39],month_order[length(month_order)-36],month_order[length(month_order)-33],month_order[length
(month_order)-30],month_order[length(month_order)-27],month_order[length(month_order)-24],month_order[leng
th(month_order)-21],month_order[length(month_order)-18],month_order[length(month_order)-15],month_order[le
ngth(month_order)-12],month_order[length(month_order)-9],month_order[length(month_order)-6],month_order[le
ngth(month_order)-3], month_order[length(month_order)]])

Days_reason <- Days_reason %>%
  filter(Period %in% month_order)

# Local Authorities ####

# for (i in 1:length(authorities)){
chosen_area = "Portsmouth"

# Time series (interactive) ####
```

11:719 (Top Level) R Script

Console

Environment History Connections

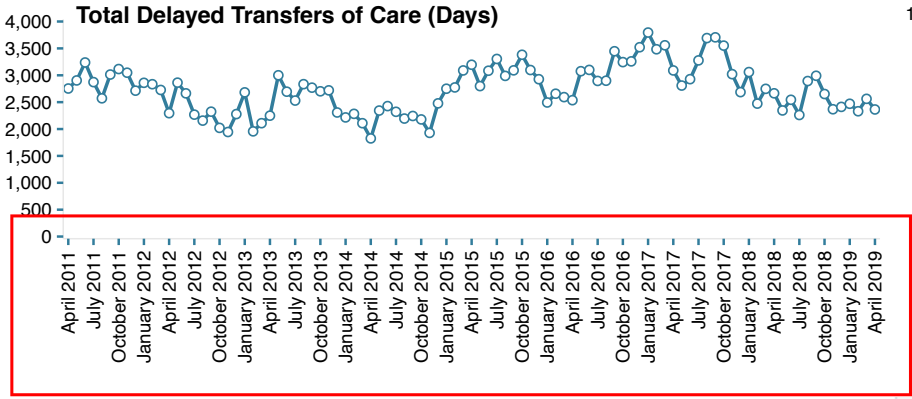
Import Dataset List

Global Environment

Read in the csv file

Delayed Transfers of Care (Days) – **Reasons at a glanc**

A delayed transfer of care from acute or non–acute (including community and mental health) care oc
depart from such care and is still occupying a bed. NHS England publishes monthly data showing th
These graphs relate to total DAYS in each month rather than number of patients. This analysis relate



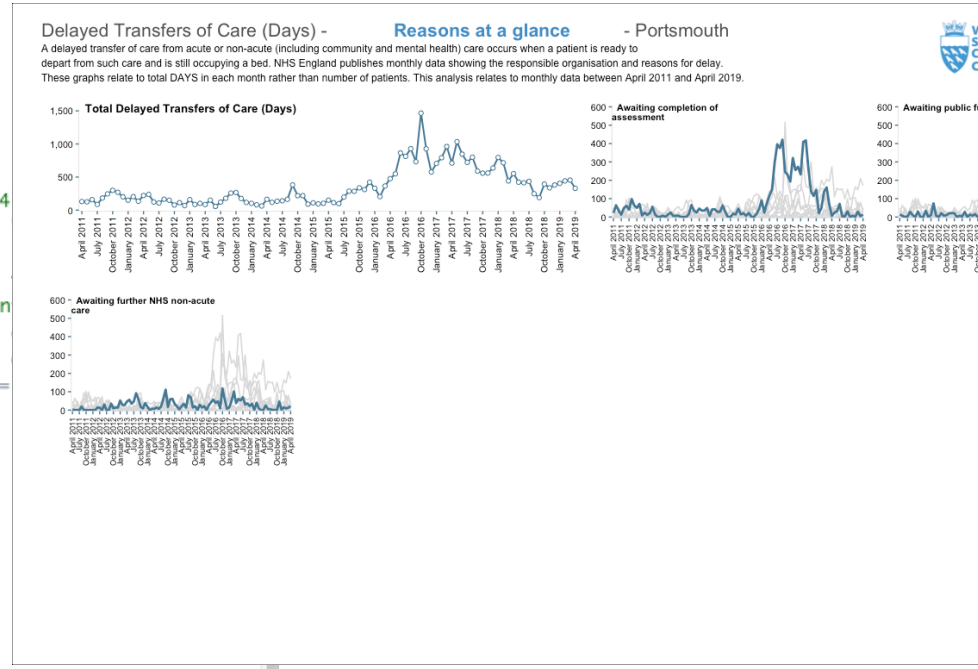
Get iteration_x working, then generalise it

```r
for (i in 1:10){
res <- unique(DToC_LA_long$Reason)[i]
assign(paste("Chart_",res, "_LA", sep = ""),ggplot(data=DToC_LA_long, aes(x=Period, y=Days, group=Reason))
+
  geom_line(size = .5, colour = "#d9d9da") +
  geom_line(data = subset(DToC_LA_long, Reason == res), size = 1, colour = "#327d9c") +
  scale_y_continuous(breaks = seq(0, round_any(max(DToC_LA_long$Days),100, f = ceiling), ifelse(round_any
(max(DToC_LA_long$Days), 100, f = ceiling) > 1000, 200,100)), expand = c(0.01,0), limits = c(0, round_any
(max(DToC_LA_long$Days),ifelse(round_any(max(DToC_LA_long$Days), 100, f = ceiling) > 1000, 200,100), f =
ceiling)),labels = comma) +
  scale_x_discrete(breaks = quarter_months) +
  ggtitle(paste("  ",res, sep = "")) +
  DToC_theme_1())}
```

Assign function allows you to create a dynamic name for an object based on a value, rather than needing to hard code a name
(e.g. name_cannot_be_dynamic <- ggplot())

- Working with grid graphics is a bit of trial and error to work out the best positions of x and y
- It can take a long time
- The payoff is worth the effort in the long term

```r
# Title and logo
pdf(paste("./Delayed Transfers of Care/DToC Days- reasons at a glance to ",month_order[length(month_order)], " ", chosen_area,".pdf"
11.7, height = 8.3) # a4 landscape
grid.newpage()
pushViewport(viewport(layout = grid.layout(24, 37)))
# Define the background colour for each window
grid.rect(gp = gpar(fill = "#ffffff", col = "#000000"))
grid.text("Delayed Transfers of Care (Days) -", just = "left", y = unit(.96, "npc"), x = unit(0.025, "npc"), gp = gpar(col = "#545454
grid.text(" Reasons at a glance ", just = "left", y = unit(.96, "npc"), x = unit(0.32, "npc"), gp = gpar(col = "#1c8ccd", fontsize =
"bold"))
grid.text(paste("- ", chosen_area, sep = ""), just = "left", y = unit(.96, "npc"), x = unit(0.52, "npc"), gp = gpar(col = "#545454",
grid.text(paste("A delayed transfer of care from acute or non-acute (including community and mental health) care occurs when a patien
from such care and is still occupying a bed. NHS England publishes monthly data showing the responsible organisation and reasons for
relate to total DAYS in each month rather than number of patients. This analysis relates to monthly data between ",month_order[1], "
ngth(month_order)],".", sep = ""), just = "left", y = unit(.91, "npc"), x = unit(0.025, "npc"), gp = gpar(col = "#000000", fontsize =
grid.raster(jsna, y = unit(0.98, "npc"), x = unit(0.78, "npc"), vjust = 1, hjust = 0, width = .2)
print(DToC_Total_chart, vp = vplayout(4:10, 2:18))
print(`Chart_Awaiting completion of\nassessment_LA`, vp = vplayout(4:10, 19:27))
print(`Chart_Awaiting public funding\n_LA`, vp = vplayout(4:10, 28:36))
print(`Chart_Awaiting further NHS non-acute\ncare_LA`, vp = vplayout(11:17, 2:9))
print(`Chart_Awaiting residential home\nplacement or availability_LA`, vp = vplayout(11:17, 10:18))
print(`Chart_Awaiting nursing home\nplacement or availability_LA`, vp = vplayout(11:17, 19:27))
print(`Chart_Awaiting care package in own\nhome_LA`, vp = vplayout(11:17, 28:36))
print(`Chart_Awaiting community equipment\nand adaptations_LA`, vp = vplayout(18:23, 2:9))
print(`Chart_Patient or family choice\n_LA`, vp = vplayout(18:23, 10:18))
print(`Chart_Disputes\n_LA`, vp = vplayout(18:23, 19:27))
print(`Chart_Housing - Patients not covered\nby NHS and Community Care Act_LA`, vp = vplayout(18:23, 28:36))
grid.text("Source: NHS England", just = "left", y = unit(.025, "npc"), x = unit(0.87, "npc"), gp = gpar(col = "#545454", fontsize = "8"))
dev.off()
```



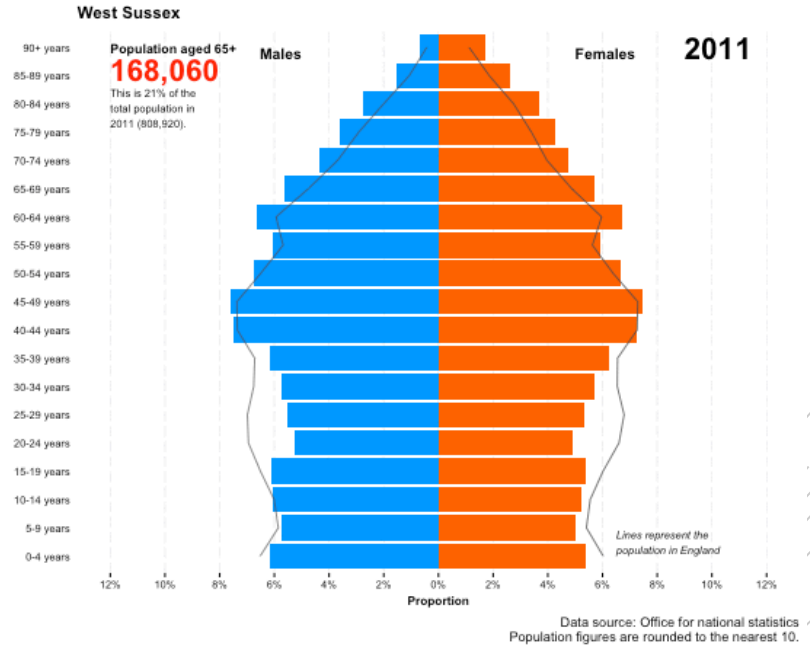
Population pyramid

- Population pyramids show the age and sex distribution of a population.
- Two areas are often compared by overlaying lines with bars.
- These can be animated to show changes over time.

- Office for National Statistics (ONS) publish estimates of the population by sex and single year of age each year in June for the previous year (2018 published a few weeks ago).
- ONS also publish population projections every two years to indicate future population changes based on a set of assumptions about fertility as well as migration and death. The most recent data are based on 2016 mid year estimates.
- Output: GIF image showing the change in age distribution of the population of West Sussex compared to England in 2011 to 2041.

Steps in R:

- Download and compile population estimates and projections for selected areas using NOMIS platform
- Create figures for each area and year and export as png files
- Use magick package to stitch all png files together and create gif files.



- Github link - https://github.com/psychty/Population-pyramids

```r
# If you have run the other scripts you should have a folder in your working direct
it will be created
if(!(file.exists(paste0("./Projecting-Health")))){
  dir.create(paste0("./Projecting-Health"))
}

# If there is no folder for images create one
if(!(file.exists(paste0("./Projecting-Health/Population_pyramid_image_files")))){
  dir.create(paste0("./Projecting-Health/Population_pyramid_image_files"))
}
```

As before, using if statements can help to build safe guards into the workflow

```r
Areas_to_include <- c("Eastbourne", "Hastings", "Lewes","Rother", "Wealden","Adur", "Arun", "Chichester", "Crawley", "Horsham", "Mid Sussex",
"Worthing", "Brighton and Hove", "NHS Brighton and Hove CCG", "NHS Coastal West Sussex CCG", "NHS Crawley CCG","NHS Eastbourne, Hailsham and
Seaford CCG", "NHS Hastings and Rother CCG","NHS High Weald Lewes Havens CCG", "NHS Horsham and Mid Sussex CCG", "West Sussex", "East Sussex",
"England")
```

```r
Comparator_x <- "England"

if(!(Comparator_x %in% Areas_to_include)){
  print(paste0("The comparator you selected (", Comparator_x, ") is not in the list of areas for which we have data. Check spelling and/or if
the data needs to be re-collated"))
}

if(!(file.exists("./Projecting-Health/Area_population_df.csv"))){
  print("Area_population_df is not available, it will be built using the 'Areas_to_include' object")
  source(paste0(github_repo_dir,"/1 - Get data - mye and projections.R"))

  if(file.exists("./Projecting-Health/Area_population_df.csv")){
    print("Area_population_df is now available.")
  }
}
```

Go to file/function | Addins

DToC reason for delay Days.R | Population pyramids.R* | 1 - Get data - mye and projections.R*

Source on Save | Run

```
100
101 NOMIS_codes <- read_csv(paste0(github_repo_dir,"/NOMIS_area_codes.csv"), col_types = cols(
    col_character(),GEOGRAPHY_NAME = col_character(),GEOGRAPHY = col_double(),  Area_Type = co
102
103 Chosen_area_codes <- subset(Areas, Area_Name %in% Areas_to_include) %>%
104   left_join(NOMIS_codes[c("GEOGRAPHY", "GEOGRAPHY_CODE")], by = c("Area_Code" = "GEOGRAPHY
105
106 # Projections ####
107
108 if(!(file.exists("./Projecting-Health/2016 SNPP CCG pop females.csv"))){
109 download.file("https://www.ons.gov.uk/file?uri=/peoplepopulationandcommunity/populationand
    /populationprojections/datasets/clinicalcommissioninggroupsinenglandz2/2016based/snppz2cc
    ./Projecting-Health/CCG_projections_2016_based.zip", mode = "wb")
110 unzip("./Projecting-Health/CCG_projections_2016_based.zip", exdir = "./Projecting-Health")
111 file.remove("./Projecting-Health/CCG_projections_2016_based.zip")
112 }
113
114 ONS_projections_SYOA <- data.frame(GEOGRAPHY = double(),GEOGRAPHY_NAME = character(), GEOG
    character(),PROJECTED_YEAR_NAME = double(),GENDER_NAME = character(),C_AGE_NAME = characte
    AME = character(), OBS_VALUE = double(), OBS_STATUS_NAME = character(), RECORD_COUNT = dou
115
116 for(i in 0:floor(as.numeric(read_csv(paste0("http://www.nomisweb.co.uk/api/v01/dataset/NM_
    ?geography=",paste(as.numeric(Chosen_area_codes$GEOGRAPHY), collapse = ","),"&projected_ye
    ender=1,2&c_age=101...191&measures=20100&select=record_count&recordlimit=1"), col_types =
    T = col_double()))/25000)){
117   df <- read_csv(url(paste0("http://www.nomisweb.co.uk/api/v01/dataset/NM_2006_1.data.csv
    ,paste(as.numeric(Chosen_area_codes$GEOGRAPHY), collapse = ","),"&projected_year=2016...20
    ,2&c_age=101...191&measures=20100&select=geography,geography_name,geography_code,projecte
    ,gender_name,c_age_name,measures_name,obs_value,obs_status_name,record_count&recordoffset=
    col_types = cols(GEOGRAPHY = col_double(),GEOGRAPHY_NAME = col_character(),  GEOGRAPHY_COD
    r(),PROJECTED_YEAR_NAME = col_double(), GENDER_NAME = col_character(),C_AGE_NAME = col_cha
    MEASURES_NAME = col_character(),OBS_VALUE = col_double(), OBS_STATUS_NAME = col_character(
    col_double()))
118
119   ONS_projections_SYOA <- ONS_projections_SYOA %>%
120     bind_rows(df)
121 }
```

117:714 | Projections

Console

https://www.nomisweb.co.uk is a platform for extracting statistics on the population as well as the UK labour market and the census directly into R using the NOMIS api

NOMIS tips

nomisr is brilliant but struggles on my work machine so build the query using read_csv()

NOMIS codes are different to ONS geography codes, so build a list first

NOMIS extracts are limited to 25,000 results at a time, so use a function to calculate how many sets of extracts you'll need (e.g. if there are 100,000 rows you'll need to do four extracts)

NOMIS does not include CCG projections, so you need to download these separately and incorporate.

GEOGRAPHY_CODE	GEOGRAPHY_NAME	GEOGRAPHY	Area_Type
E06000005	Darlington	1879048193	Unitary Authority
E06000047	County Durham	1879048194	Unitary Authority
E06000001	Hartlepool	1879048195	Unitary Authority
E06000002	Middlesbrough	1879048196	Unitary Authority
E06000057	Northumberland	1879048197	Unitary Authority
E06000003	Redcar and Cleveland	1879048198	Unitary Authority
E06000004	Stockton-on-Tees	1879048199	Unitary Authority
E08000037	Gateshead	1879048200	District
E08000021	Newcastle upon Tyne	1879048201	District
E08000022	North Tyneside	1879048202	District

ONS_ccg_projections_df

Filter

	AREA_CODE	AREA_NAME	SEX	AGE_GROUP	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	203
1	E38000021	NHS Brighton and Hove CCG	females	0	1450	1449	1467	1471	1480	1485	1499	1505	1511	1518	1522	1526	1526	1525	15
2	E38000021	NHS Brighton and Hove CCG	females	1	1436	1419	1417	1434	1436	1443	1447	1458	1464	1470	1476	1480	1483	1483	14
3	E38000021	NHS Brighton and Hove CCG	females	2	1355	1421	1403	1400	1416	1417	1423	1425	1436	1441	1447	1452	1455	1458	14
4	E38000021	NHS Brighton and Hove CCG	females	3	1444	1339	1401	1383	1379	1394	1395	1400	1402	1412	1417	1422	1427	1430	14
5	E38000021	NHS Brighton and Hove CCG	females	4	1562	1427													
6	E38000021	NHS Brighton and Hove CCG	females	5	1389	1549													
7	E38000021	NHS Brighton and Hove CCG	females	6	1514	1396													
8	E38000021	NHS Brighton and Hove CCG	females	7	1430	1513													
9	E38000021	NHS Brighton and Hove CCG	females	8	1428	1428													
10	E38000021	NHS Brighton and Hove CCG	females	9	1389	1425													
11	E38000021	NHS Brighton and Hove CCG	females	10	1401	1395													
12	E38000021	NHS Brighton and Hove CCG	females	11	1303	1407													
13	E38000021	NHS Brighton and Hove CCG	females	12	1317	1316													
14	E38000021	NHS Brighton and Hove CCG	females	13	1265	1325													
15	E38000021	NHS Brighton and Hove CCG	females	14	1298	1277													
16	E38000021	NHS Brighton and Hove CCG	females	15	1355	1299													
17	E38000021	NHS Brighton and Hove CCG	females	16	1292	1353													
18	E38000021	NHS Brighton and Hove CCG	females	17	1321	1325													
19	E38000021	NHS Brighton and Hove CCG	females	18	1713	1591													
20	E38000021	NHS Brighton and Hove CCG	females	19	3275	3290													
21	E38000021	NHS Brighton and Hove CCG	females	20	3498	3928													
22	E38000021	NHS Brighton and Hove CCG	females	21	3315	3494													
23	E38000021	NHS Brighton and Hove CCG	females	22	3317	3091													
24	E38000021	NHS Brighton and Hove CCG	females	23	2991	3059													
25	E38000021	NHS Brighton and Hove CCG	females	24	3030	2810													
26	E38000021	NHS Brighton and Hove CCG	females	25	3017	2855	2663	2698	2536	2605	2745	2686	2629	2371	2544	2482	2523	2595	26
27	E38000021	NHS Brighton and Hove CCG	females	26	2756	2859	2716	2546	2572	2445	2483	2602	2545	2494	2439	2411	2353	2392	24

Showing 1 to 30 of 1,288 entries, 30 total columns

NOMIS tips continued

The extract will be a table of data showing one row per area and age and sex, with a row for each year.

You can use dplyr ->
To add a grouping variable, summarise data and then transpose the dataframe (gather) so that you end up with a dataframe of one row per area, sex, age group and year as below

Filter

	Area_name	Area_code	Sex	Age group	Year	Population	Data_type	Age_band_type
1	NHS Brighton and Hove CCG	E38000021	Female	0-4 years	2018	7016	Projected - ONS	5 years
2	NHS Brighton and Hove CCG	E38000021	Female	10-14 years	2018	6911	Projected - ONS	5 years
3	NHS Brighton and Hove CCG	E38000021	Female	15-19 years	2018	8717	Projected - ONS	5 years
4	NHS Brighton and Hove CCG	E38000021	Female	20-24 years	2018	16670	Projected - ONS	5 years
5	NHS Brighton and Hove CCG	E38000021	Female	25-29 years	2018	13015	Projected - ONS	5 years
6	NHS Brighton and Hove CCG	E38000021	Female	30-34 years	2018	10157	Projected - ONS	5 years
7	NHS Brighton and Hove CCG	E38000021	Female	35-39 years	2018	9980	Projected - ONS	5 years
8	NHS Brighton and Hove CCG	E38000021	Female	40-44 years	2018	9356	Projected - ONS	5 years

```
151 ONS_projection_1941_quinary <- ONS_projections_SYOA %>%
152   filter(AGE_GROUP != "All ages") %>%
153   mutate(Age = as.numeric(gsub(" and over", "", AGE_GROUP))) %>%
154   mutate(`Age group` = ifelse(Age <= 4, "0-4 years", ifelse(Age <= 9, "5-9
      years", ifelse(Age <= 24, "20-24 years", ifelse(Age <= 29, "25-29 years",if
      ,ifelse(Age <= 44, "40-44 years", ifelse(Age <= 49, "45-49 years",ifelse(Ag
      <= 64, "60-64 years", ifelse(Age <= 69, "65-69 years",ifelse(Age <= 74, "70
      years", ifelse(Age <= 89, "85-89 years", "90+ years")))))))))))))))))) %>%
155   group_by(AREA_NAME, AREA_CODE, SEX, `Age group`) %>%
156   summarise(`2018` = sum(`2018`, na.rm = TRUE),
157             `2019` = sum(`2019`, na.rm = TRUE),
158             `2020` = sum(`2020`, na.rm = TRUE),
159             `2021` = sum(`2021`, na.rm = TRUE),
160             `2022` = sum(`2022`, na.rm = TRUE),
161             `2023` = sum(`2023`, na.rm = TRUE),
162             `2024` = sum(`2024`, na.rm = TRUE),
163             `2025` = sum(`2025`, na.rm = TRUE),
164             `2026` = sum(`2026`, na.rm = TRUE),
165             `2027` = sum(`2027`, na.rm = TRUE),
166             `2028` = sum(`2028`, na.rm = TRUE),
167             `2029` = sum(`2029`, na.rm = TRUE),
168             `2030` = sum(`2030`, na.rm = TRUE),
169             `2031` = sum(`2031`, na.rm = TRUE),
170             `2032` = sum(`2032`, na.rm = TRUE),
171             `2033` = sum(`2033`, na.rm = TRUE),
172             `2034` = sum(`2034`, na.rm = TRUE),
173             `2035` = sum(`2035`, na.rm = TRUE),
174             `2036` = sum(`2036`, na.rm = TRUE),
175             `2037` = sum(`2037`, na.rm = TRUE),
176             `2038` = sum(`2038`, na.rm = TRUE),
177             `2039` = sum(`2039`, na.rm = TRUE),
178             `2040` = sum(`2040`, na.rm = TRUE),
179             `2041` = sum(`2041`, na.rm = TRUE)) %>%
180   ungroup() %>%
181   mutate(SEX = capwords(SEX)) %>%
182   rename(Area_name = AREA_NAME,
183          Area_code = AREA_CODE,
184          Sex = SEX) %>%
185   mutate(Sex = ifelse(Sex == "Females", "Female", ifelse(Sex == "Males"
186   gather(Year, Population, `2018`:`2041`, factor_key = TRUE) %>%
187   mutate(Data_type = "Projected - ONS",
188          Age_band_type = "5 years")
```

```r
if(exists("Areas_to_include") & file.exists("./Projecting-Health/Area_population_df.csv")){
  print("Both objects are available")
  Area_population_df <- read_csv("./Projecting-Health/Area_population_df.csv", col_types = col
col_character(),Area_Type = col_character(),  Sex = col_character(),Age_group = col_character(
col_double(),Population = col_double(),Data_type = col_character())) %>%
    group_by(Area_Name, Age_band_type, Year, Sex) %>%
    mutate(All_age_population = sum(Population, na.rm = TRUE)) %>%
    mutate(Proportion = Population / All_age_population) %>%
    filter(Age_band_type == "5 years") %>%
    mutate(Age_group  = factor(Age_group, levels = c("0-4 years","5-9 years","10-14 years","15
years","35-39 years","40-44 years","45-49 years","50-54 years","55-59 years","60-64 years","65
years","85-89 years","90+ years")))

  Areas <- read_csv("./Projecting-Health/Area_lookup_table.csv", col_types = cols(LTLA17CD = c
D = col_character(),UTLA17NM = col_character(),FID = col_double()))
  Lookup <- read_csv("./Projecting-Health/Area_types_table.csv", col_types = cols(Area_Code =
(),Area_Type = col_character()))

  if(length(setdiff(Areas_to_include, Area_population_df$Area_Name))>0){
    print("There are some areas chosen that are not in the Area_population_df. The 'Get data -
will overwrite the Area_population_df.")
    source("~/Documents/Repositories/Projecting-Health/Get data - mye and projections.R")
  }

  if(length(setdiff(Areas_to_include, Area_population_df$Area_Name))>0){
    print("There are still some areas chosen that are not in the Area_population_df. Check the
  }

  if(length(setdiff(Areas_to_include, Area_population_df$Area_Name))==0){
    print("The Area_population_df matches the Areas_to_include list.")
  }
}

Years_available <- unique(Area_population_df$Year)

```

Read in the data and add proportions ready for making a population pyramid.

This also has an if statement to check that the list of areas_to_include are actually in the data file 'Area_population_df.csv'

```r
for(i in 1:length(Areas_to_include)){
Area_x <- Areas_to_include[i]

# We need to create a folder for each of our areas
if(!(file.exists(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x)))){
  dir.create(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x))
}

if(!(file.exists(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x,"/Numbers")))){
  dir.create(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x,"/Numbers"))
}

if(!(file.exists(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x,"/Proportion")))){
  dir.create(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x,"/Proportion"))
}

Area_pyramid_df <- Area_population_df %>%
  filter(Area_Name == Area_x)
```

Loop through each area and create a folder for it.

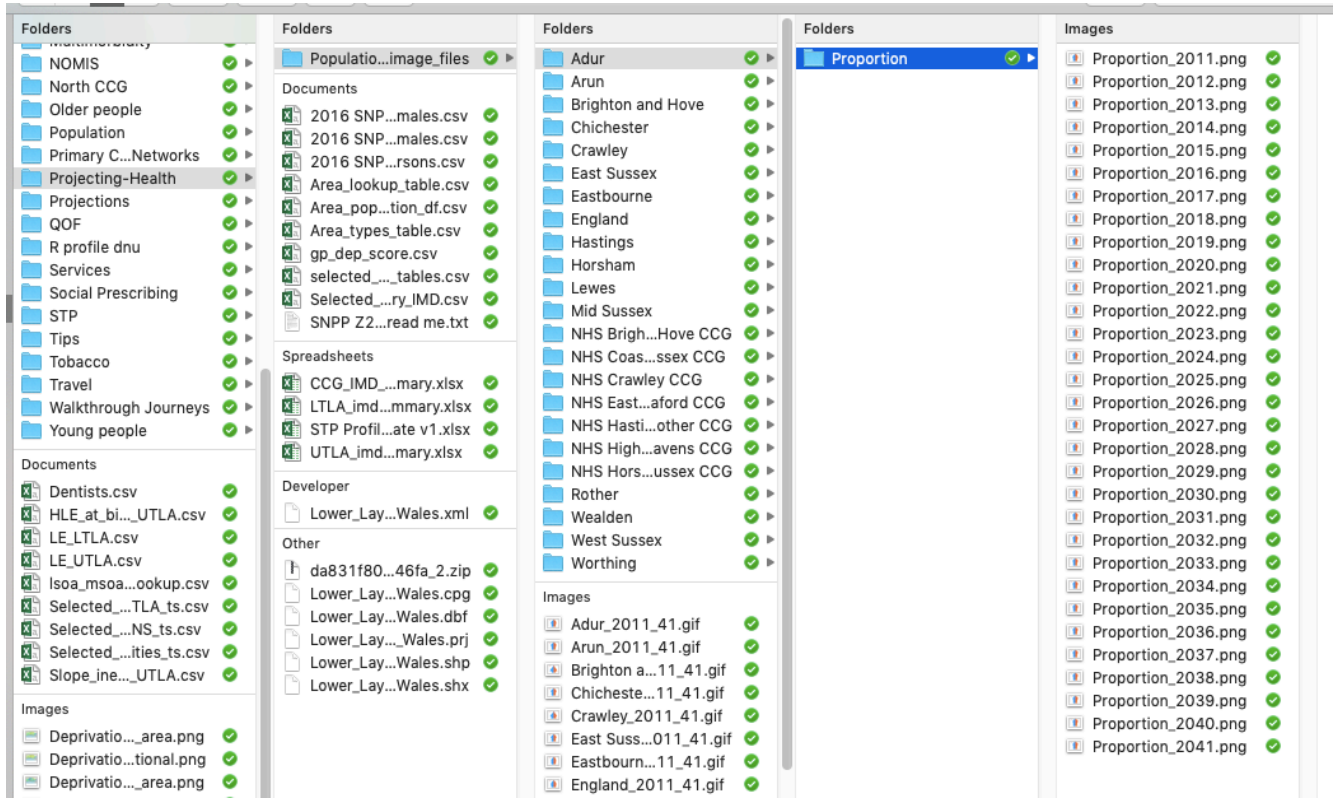
Within the new folder create a folder for pyramids based on numbers and for proportions

Create Area_pyramid_df from the larger file which contains only data for area_x (but all years)

There is a second loop within the first loop to go through each year of the object Area_pyramid_df

For each year create a ggplot object for numbers and for proportions

```r
for(j in 1:length(Years_available)){
  Year_x <- Years_available[j]

  df_bars <- Area_pyramid_df %>%
    filter(Year == Year_x)

df_lines <- Area_population_df %>%
  ungroup() %>%
  filter(Area_Name == "England") %>%
  filter(Year == Year_x) %>%
  rename(Comparator_number = Population,
         Comparator_proportion = Proportion,
         Comparator = Area_Name) %>%
  select(Comparator, Age_group, Sex, Year, Comparator_number, Comparator_proportion)

combined_pyramid <- df_bars %>%
  left_join(df_lines, by = c("Age_group", "Sex", "Year"))

Pyramid_xabsolute_fig <- ggplot(data = combined_pyramid, aes(x = Age_group, y = Population, fill = Sex)) +
  geom_bar(data = subset(combined_pyramid, Sex== "Female"),
           stat = "identity") +
  geom_bar(data = subset(combined_pyramid, Sex== "Male"),
           stat = "identity",
           position = "identity",
           mapping = aes(y = -Population)) +
  scale_fill_manual(values =  c("#ff6600", "#0099ff"), breaks = c("Males","Females")) +
  coord_flip() +
```



The result of the loop

```
280  ggsave(paste0("./Projecting-Health/Population_pyramid_image_f
     iles/",Area_x,"/Numbers/Number_",Year_x,".png"), plot =
     Pyramid_xabsolute_fig, width = 7.5, height = 6, dpi = 75)
281
282  ggsave(paste0("./Projecting-Health/Population_pyramid_image_f
     iles/",Area_x,"/Proportion/Proportion_",Year_x,".png"), plot
     = Pyramid_xperc_fig, width = 7.5, height = 6, dpi = 75)
283
284  }
285  }
286
```

Easiest part >

Use magick and dplyr's map() function

```
287  # Image processing ####
288
289  #install.packages('magick')
290  library(magick)
291
292  for(i in 1:length(Areas_to_include)){
293    Area_x <- Areas_to_include[i]
294  list.files(path = paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x,"/Proportion/"), pattern = "*.png", full.names = T)
295    map(image_read) %>% # reads each path file
296    image_join() %>% # joins image
297    image_animate(fps=2) %>% # animates, can opt for number of loops
298    image_write(paste0("./Projecting-Health/Population_pyramid_image_files/",Area_x,"_2011_41.gif"),
299                quality = 50,
300                density = 100) # write to current dir
301  }
```

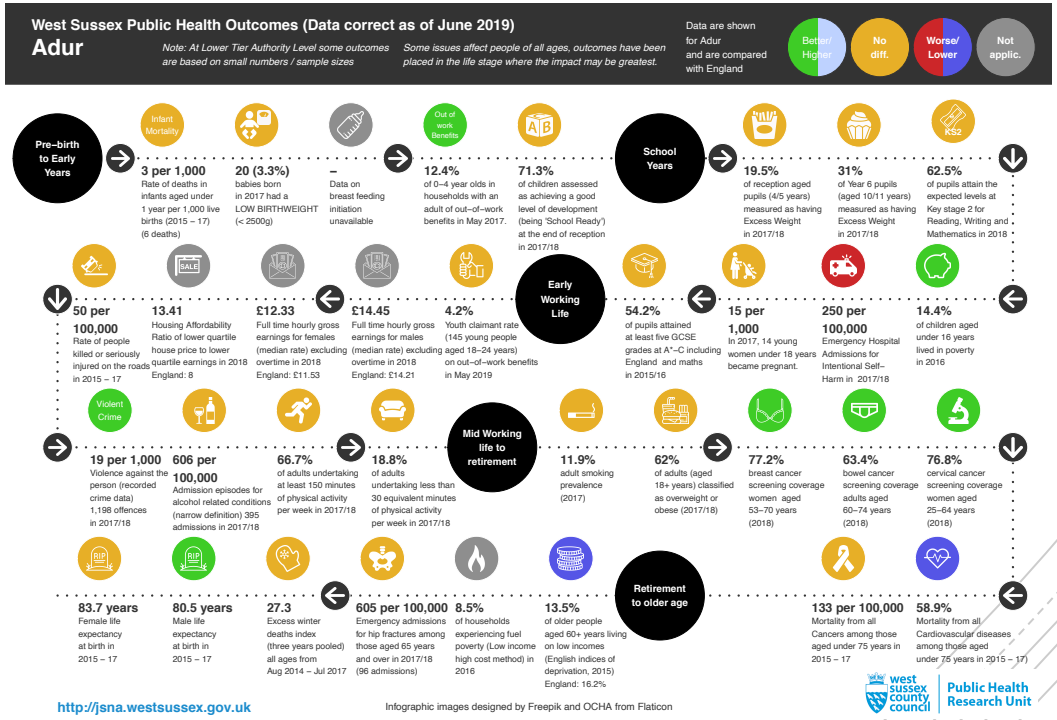
Public Health Journey through indicators infographic

- The journey through indicators infographic is a snapshot of 34 PH outcomes at lower tier LA level compared to England.
- Updates to individual indicators can be automatically incorporated when the script is run.
- Several infographics can be created at once using loops

- Producing an overview snapshot of key indicators is important to understand the population of an area, as well as trends and progress towards targets.
- Indicators are updated frequently but not usually at the same time.
- Output: PDF infographic showing the local area value compared to England for 34 indicators.

Steps in R:

- Read in png icons
- Extract data for 34 indicators
- Calculate confidence intervals and compare to England value
- Use grid package to position, colour and label indicators on a page



- Github link - https://github.com/psychty/la-ph-walkthrough

Contact me for png icons

```r
# This will be the coordinate system for placing our objects in grid later
vplayout <- function(x,y)
  viewport(layout.pos.row = x, layout.pos.col = y)

# This capitalises words in a string
capwords = function(s, strict = FALSE) {
  cap = function(s) paste(toupper(substring(s, 1, 1)),{s = substring(s, 2); if(strict) tolower(s) else s},sep = "",
collapse = " " )
  sapply(strsplit(s, split = " "), cap, USE.NAMES = !is.null(names(s)))}

# We can define some hex colours to use later in our comparisons
better <- "#3ECC26"
no_diff <- "#E7AF27"
worse <- "#CC2629"
not_applic <- "#8E8E8E"
higher = "#BED2FF"
lower = "#5555E6"

# This checks to see if 'Journey through indicators' folder exists in the working directory and if not it will be
created.
if (!file.exists("./Journey through indicators")) {
  dir.create("./Journey through indicators")}

# Read in JSNA logo (only download it if it is not available in your working directory)
if(!file.exists("./Journey through indicators/Research Unit.png")){download.file("https://github.com/psychty/la-ph
-walkthrough/raw/master/Research%20Unit.png", "./Journey through indicators/Research Unit.png", mode = 'wb')} #
This downloads a png image from the West Sussex JSNA website and saves it to your working directory. The if(!file
.exists()) only runs the command if the file does not exist (because we have included an ! at the beginning)
unit_logo = readPNG("./Journey through indicators/Research Unit.png")

# I do not think I can legitimately publically host and make available the png images downloaded from flaticon
(contact me and I can email something).

# Read in infographics icons ####
ind_2a_icon = readPNG("./Journey through indicators/png/ind_2a_icon_people.png")
ind_2b_icon = readPNG("./Journey through indicators/png/ind_2b_icon_scales.png")
```

Specify colours using hex numbers once at the top of the script

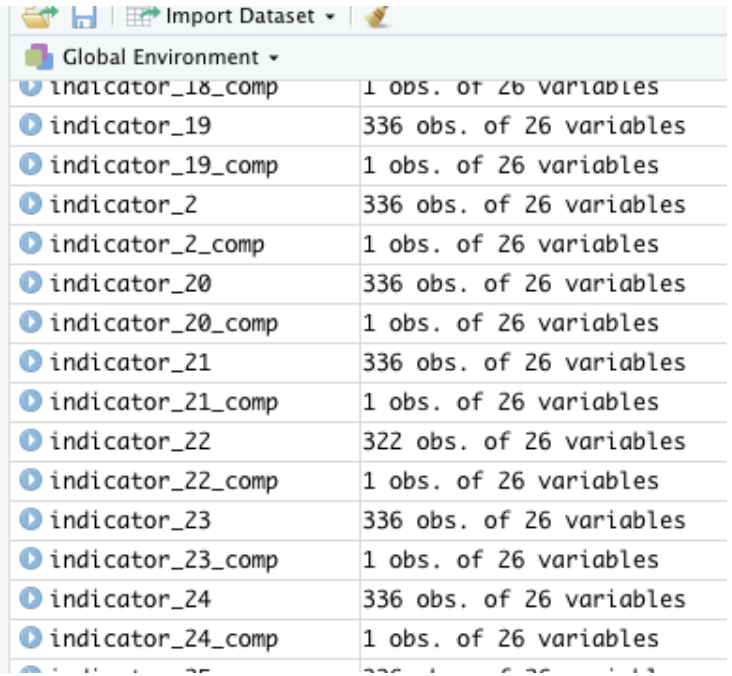
Read in png files

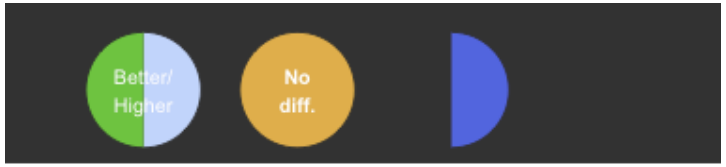
```r
# Indicator 1 - Infant Mortality ####
indicator_1 <- fingertips_data(IndicatorID = 92196, AreaTypeID = 101) %>%
  arrange(desc(Timeperiod)) %>%  # Order by descending year (latest data on top)
  filter(Timeperiod == unique(Timeperiod)[1]) # Now that we have ordered the data, we select the first unique value Timeperiod as this will be the most recent value

indicator_1_comp <- indicator_1 %>%
  filter(AreaName == comp_area)
```

Most of these are available from Public Health England's fingertips platform via an api.

Some require some more work such as school readiness

```r
# Indicator 5 - school Readiness by pupil residency ###
# 2018
download.file("https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/759613/EYFSP_2018_additional_tables_underlying_data.zip",  "./Journey through indicators/EYFSP_2018_Tables.zip")
unzip("./Journey through indicators/EYFSP_2018_Tables.zip", exdir = "./Journey through indicators")

indicator_5 <- read_csv("./Journey through indicators/EYFSP_LAD_ pr_additional_tables_2018.csv", col_types = cols(year = col_double(),level = col_character(),country_code = col_character(), country_name = col_character(),  region_code_pr = col_character(),  region_name_pr = col_character(), lad_code_pr = col_character(),lad_name_pr = col_character(),number_of_pupils = col_double(),  elg_number = col_double(),elg_percent = col_double(),gld_number = col_double(), gld_percent = col_double(), point_score = col_double(), average_point_score = col_double()
)) %>%
  mutate(lad_name_pr = ifelse(is.na(region_name_pr), country_name_pr, ifelse(is.na(lad_name_pr), region_name_pr, lad_name_pr))) %>%
  mutate(lad_code_pr = ifelse(is.na(region_code_pr), country_code_pr, ifelse(is.na(lad_code_pr), region_code_pr, lad_code_pr))) %>%
  mutate(lad_name_pr = gsub(" UA", "", lad_name_pr)) %>%
  rename(Area_name = lad_name_pr,
         Area_code = lad_code_pr) %>%
  select(year, level, Area_code, Area_name, number_of_pupils, gld_number, gld_percent) %>%
  mutate(gld_percent = gld_number / number_of_pupils * 100,
         gld_lci = wilson_lower(gld_number, number_of_pupils, confidence = .95) * 100,
         gld_uci = wilson_upper(gld_number, number_of_pupils, confidence = .95) * 100) %>%
  filter(year == "201718")

indicator_5_comp <- indicator_5 %>%
  filter(level == "Regional") %>%
  summarise(number_of_pupils = sum(number_of_pupils, na.rm = TRUE),
            gld_number = sum(gld_number, na.rm = TRUE)) %>%
  mutate(year = "201718",
         level = "England",
         Area_name = "England",
         Area_code = "E92000001") %>%
  mutate(gld_percent = gld_number / number_of_pupils * 100,
         gld_lci = wilson_lower(gld_number, number_of_pupils, confidence = .95) * 100,
         gld_uci = wilson_upper(gld_number, number_of_pupils, confidence = .95) * 100) %>%
  select(year, level, Area_code, Area_name, number_of_pupils, gld_number, gld_percent, gld_lci, gld_uci)
```

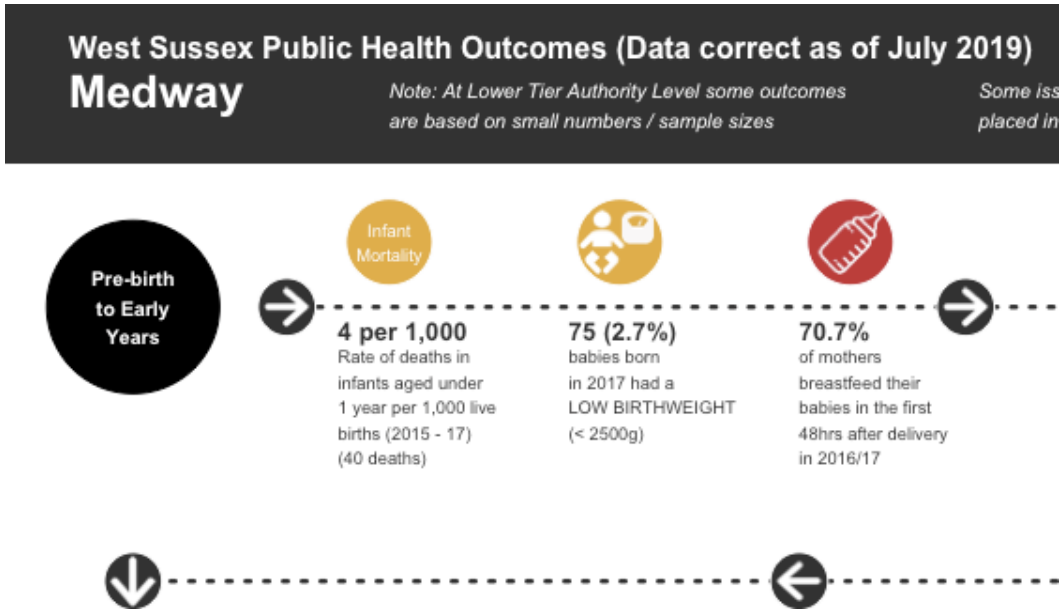





```
grid.circle(x = 0.775, y = 0.94  , r = 0.04, default.units = "npc", name = NULL, gp = gpar(fill
= better, col = "#333333"), draw = TRUE, vp = NULL)
pushViewport(viewport(width = 1, height = 1, clip=TRUE))
grid.text("Better/\nHigher", just = "centre", x = unit(0.775, "npc"), y = unit(.94, "npc"), gp =
gpar(col = "#ffffff", fontsize = "8"))
```

You can add shapes (and clipping to make semi-circles) as well as dotted lines


```
613  # Dotted lines ##
614  grid.lines(x = c(0.1,0.962), y = 0.79, default.units = "npc", name =  NULL, gp = gpar(col =
     "#333333", lty = "dotted", lwd = 2.5))
615  grid.lines(x = 0.962, y = c(0.6,0.79), default.units = "npc", name =  NULL, gp = gpar(col =
     "#333333", lty = "dotted", lwd = 2.5))
```




```
679  # indicator 3 - Breast feeding initiation
680  indicator_3_ch <- subset(indicator_3, AreaName == ch_area)
681  # Higher is better
682  indicator_3_colour <- ifelse(is.na(indicator_3_ch$LowerCI95.0limit), not_applic, ifelse(indicato
     r_3_ch$LowerCI95.0limit > indicator_3_comp$UpperCI95.0limit, better, ifelse(indicator_3_ch$Upper
     CI95.0limit < indicator_3_comp$LowerCI95.0limit, worse, no_diff)))
683
684  grid.circle(x = 0.33, y = 0.835  , r = 0.03, default.units = "npc", name = NULL, gp = gpar(fill
     = indicator_3_colour, col = "#ffffff"), draw = TRUE, vp = NULL)
685  grid.raster(ind_3_icon, x = unit(0.33, "npc"), y = unit(0.835, "npc"),  just = "centre", width =
     .025)
686
687  grid.text(ifelse(is.na(indicator_3_ch$Value),"-",paste(round(indicator_3_ch$Value,1), "%", sep =
     "")), just = "left", x = unit(0.31, "npc"), y = unit(.78, "npc"), vjust = 1, gp = gpar(col =
     "#333333", fontsize = "10", fontface = "bold"))
688  grid.text(ifelse(is.na(indicator_3_ch$Value),"Data on\nbreast feeding\ninitiation\nunavailable"
     ,paste("of mothers\nbreastfeed their\nbabies in the first\n48hrs after delivery\nin ", as
     .character(indicator_3_ch$Timeperiod), sep = "")), just = "left", x = unit(0.31, "npc"), y =
     unit(.76, "npc"), vjust = 1, gp = gpar(col = "#333333", fontsize = "7"))
689
```

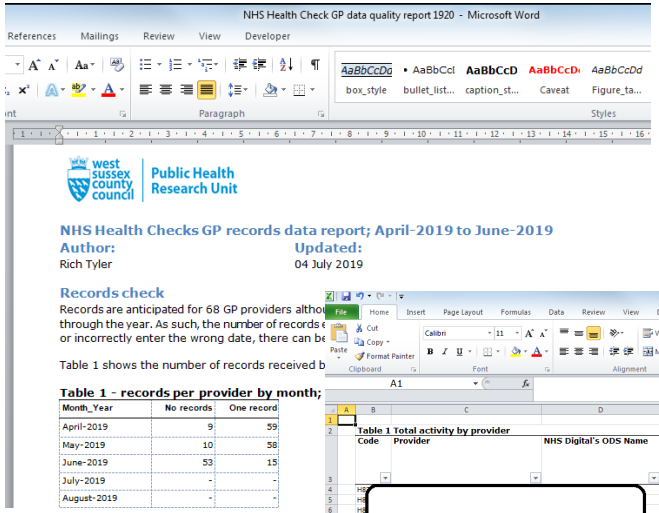
NHS Health Checks provider data

- NHS Health Checks is a national screening programme for adults in England aged 40-74.
- Eligible patients (those not currently diagnosed with certain conditions) are assessed for early signs of stroke, kidney disease heart disease, type 2 diabetes and dementia.
- We do not have access to primary care clinical systems so providers have to tell us what activity they have completed so we can pay them.

- West Sussex Public Health commission primary care and pharmacy providers to conduct NHS Health Checks.
- Providers send in data for which they get paid.
- Pharmacies and GPs use different systems.
- Output: Multiple outputs showing activity and invoicing amounts.

Steps in R:

- Read in data from different sources
- Clean up data
- Identify missing or duplicate entries
- Compare with figures held elsewhere
- Compile word document reports and excel files


NHS Health Check GP data quality report 1920 - Microsoft Word

west sussex county council | Public Health Research Unit

NHS Health Checks GP records data report; April-2019 to June-2019

Author: Rich Tyler

Updated: 04 July 2019

Records check

Records are anticipated for 68 GP providers althou
through the year. As such, the number of records e
or incorrectly enter the wrong date, there can be

Table 1 shows the number of records received b

Table 1 - records per provider by month;

Month_Year	No records	One record
April-2019	9	59
May-2019	10	58
June-2019	53	15
July-2019	-	-
August-2019	-	-

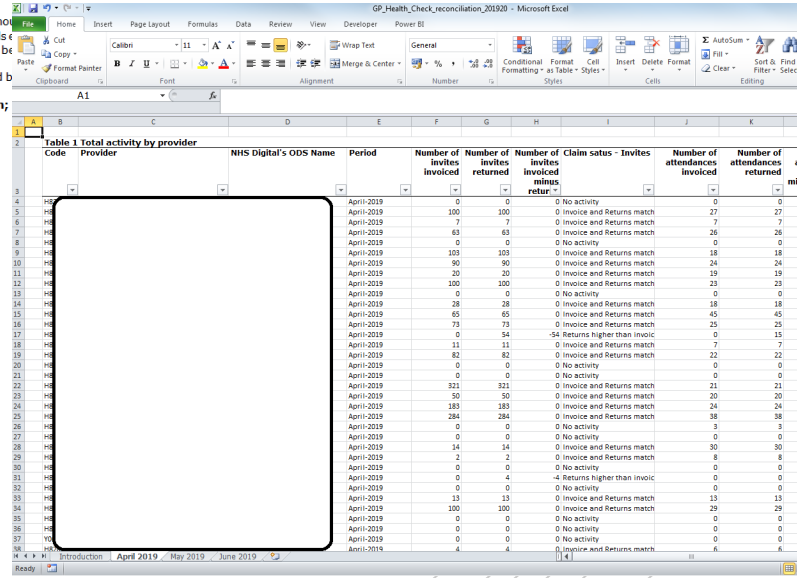
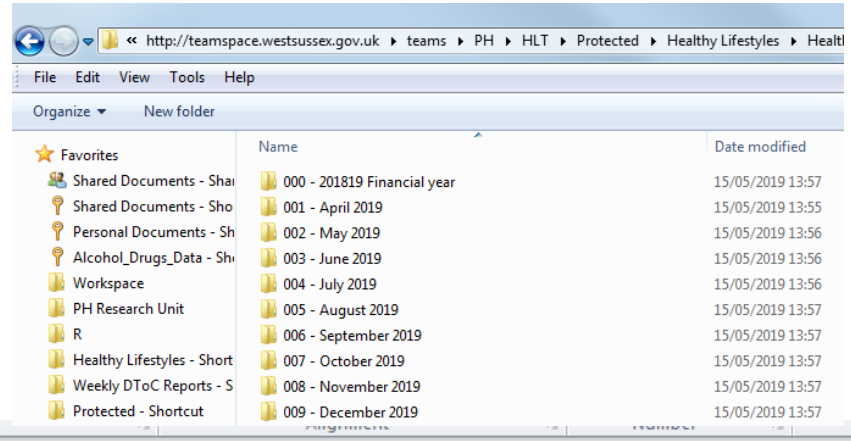

GP_Health_Check_reconciliation_201920 - Microsoft Excel

Table 1 Total activity by provider

Code	Provider	NHS Digital's ODS Name	Period	Number of invites invoiced	Number of invites returned	Number of invites invoiced minus retur	Claim satus - Invites	Number of attendances invoiced	Number of attendances returned
[redacted]			April-2019	0	0	0	No activity	0	0
			April-2019	100	100	0	Invoice and Returns match	27	27
			April-2019	7	7	0	Invoice and Returns match	7	7
			April-2019	63	63	0	Invoice and Returns match	26	26
			April-2019	0	0	0	No activity	0	0
			April-2019	103	103	0	Invoice and Returns match	18	18
			April-2019	90	90	0	Invoice and Returns match	24	24
			April-2019	20	20	0	Invoice and Returns match	19	19
			April-2019	100	100	0	Invoice and Returns match	23	23
			April-2019	0	0	0	No activity	0	0
			April-2019	28	28	0	Invoice and Returns match	18	18
			April-2019	65	65	0	Invoice and Returns match	45	45
			April-2019	73	73	0	Invoice and Returns match	25	25
			April-2019	0	54	-54	Returns higher than invoic	0	15
			April-2019	11	11	0	Invoice and Returns match	7	7
			April-2019	82	82	0	Invoice and Returns match	22	22
			April-2019	0	0	0	No activity	0	0
			April-2019	0	0	0	No activity	0	0
			April-2019	321	321	0	Invoice and Returns match	21	21
			April-2019	50	50	0	Invoice and Returns match	20	20
			April-2019	183	183	0	Invoice and Returns match	24	24
			April-2019	284	284	0	Invoice and Returns match	38	38
			April-2019	0	0	0	No activity	3	3
			April-2019	0	0	0	No activity	0	0
			April-2019	14	14	0	Invoice and Returns match	30	30
			April-2019	2	2	0	Invoice and Returns match	8	8
			April-2019	0	0	0	No activity	0	0
			April-2019	0	4	-4	Returns higher than invoic	0	0
			April-2019	0	0	0	No activity	0	0
			April-2019	13	13	0	Invoice and Returns match	13	13
			April-2019	100	100	0	Invoice and Returns match	29	29
			April-2019	0	0	0	No activity	0	0
			April-2019	0	0	0	No activity	0	0
			April-2019	0	0	0	No activity	0	0
			April-2019	0	4	-4	Returns higher than invoic	0	6

```r
directory_top <- "//teamspace.westsussex.gov.uk/teams/PH/HLT/Protected/Healthy Lifestyles/Health Check and Smoking Performance/GP HC data returns/"

Folders_1910 <- c("001 - April 2019/", "002 - May 2019/", "003 - June 2019/", "004 - July 2019/", "005 - August 2019/", "006 - September 2019/", "007 - October 2019/", "008 - November 2019/", "009 - December 2019/", "010 - January 2020/", "011 - February 2020/", "012 - March 2020/")

for(j in 1:length(Folders_1910)){
  directory_x = paste0(directory_top, Folders_1910[j])
for(i in 1:length(list.files(directory_x))){
  if(length(list.files(directory_x)) == 0){
    next
  }

  file_x = paste0(directory_x, list.files(directory_x)[i])

  df_x = data.frame(Code = as.character(read_excel(file_x, sheet = "Data form", range = "B2", col_names = FALSE)),
                    Name = as.character(read_excel(file_x, sheet = "Data form", range = "B3", col_names = FALSE)),
                    Date = read_excel(file_x, range = "B4", col_names = FALSE, col_types = c("date")),
                    Invited = as.numeric(read_excel(file_x, sheet = "Data form", range = "B6", col_names = FALSE)),
                    Attended = as.numeric(read_excel(file_x, sheet = "Data form", range = "B7", col_names = FALSE)),check.names = FALSE) %>%
    mutate(MonthBeginning = paste0("1-", format(X__1, "%m-%Y"))) %>%
    mutate(Month_Year = format(X__1, "%B-%Y")) %>%
    mutate_if(is.numeric, funs(ifelse(is.na(.), 0, .))) %>%
    select(-X__1)

  compiled_df <- rbind(compiled_df, df_x)
}
}
```

Reading in data when the files are complicated



E41 | 1

	A	B	C	D	E	F	G	H	I
1									
2	GP Practice code:	H8_ _ _							
3	GP Practice name Based on practice code, if the name does not match your practice, check the 'GP Practice Lookup' sheet in this workbook	Surgery							
4	Month:	Jun-19							
5									
6	Total number of people invited	38							
7	Total number of people attended	13							
8									
9	**Attended**	**40 - 49**	**50 - 59**	**60 - 74**					
10	Males	3	0	1					
11	Female	2	4	3					
12									
13									
14	***Smoking***				**Males**	**Females**		**Totals**	
15	Numbers of smokers seen				0	1		1	
16	Of the number of Smokers seen how many were referred into in-house stop smoking i.e.your GP surgery				0	1		1	
17	Of the number of Smokers seen how many were signposted to an NHS Stop Smoking Pharmacy Clinic				0	0		0	
18	Of the number of Smokers seen how many declined the Stop smoking referral/signpost				0	1		1	
19	***Weight Management***								
20	Numbers with BMI > 25 but < 35 referred to Wellbeing Hub for Weight management advice				1	3		4	

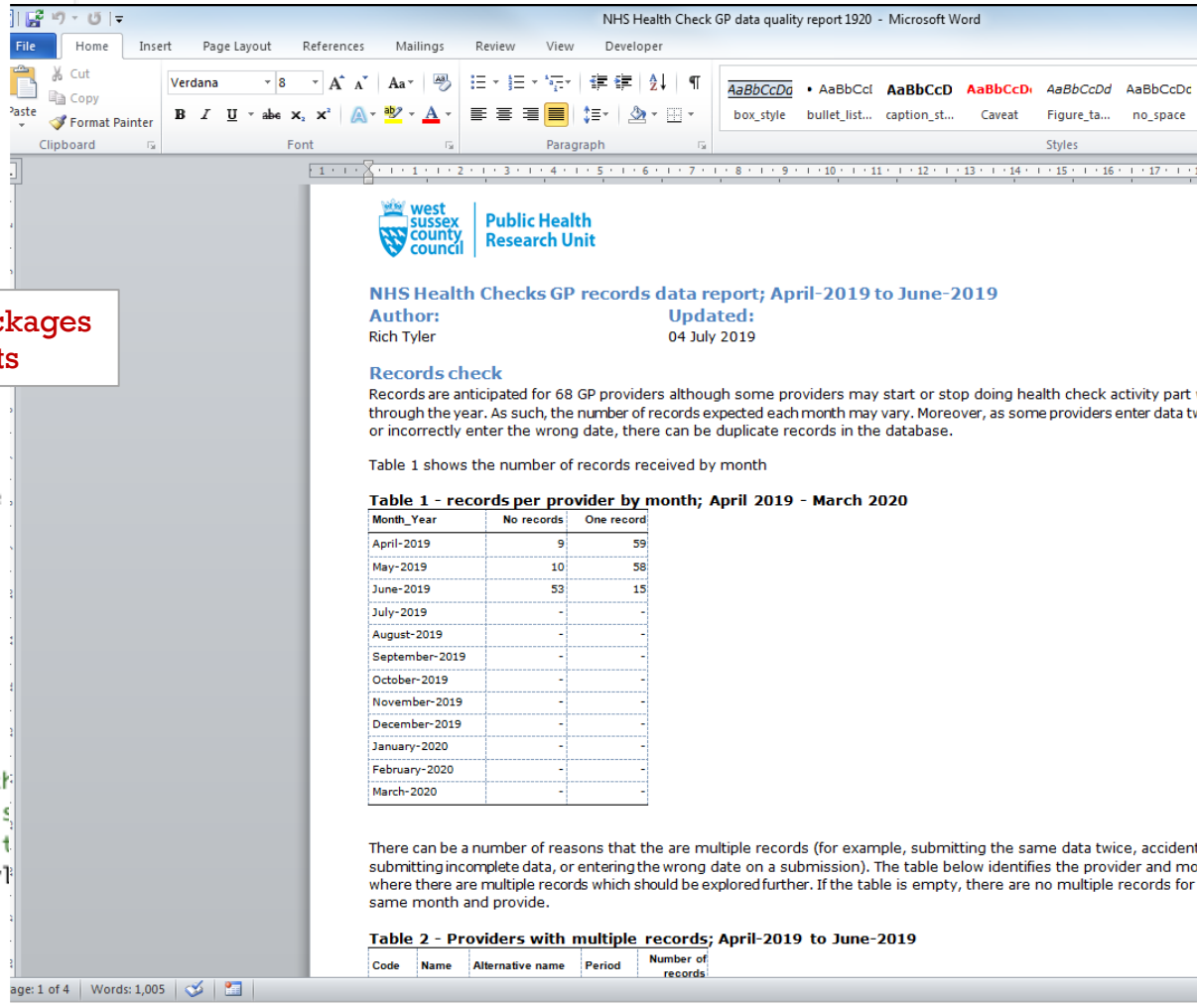
```r
GP_HC_1920_rows <- GP_HC_1920 %>%
  group_by(Code, Month_Year) %>%
  summarise(Number_of_records = n()) %>%
  complete(Code, fill = list(Number_of_records = 0)) %>%  # Fill in any blanks with zeros
  complete(Month_Year, fill = list(Number_of_records = 0)) %>% # Fill in any blanks with zeros
  filter(!(is.na(Month_Year))) %>%  # Not sure why there are some blanks (there are the same number of blanks as there are rows in the data.frame which makes me thnk theres a bug (user error more likely))
  unique() %>% # Again some observations appear the same number of times as the number of rows in the original df
  left_join(GP_codes_HC, by = "Code") %>%
  ungroup()
```

```r
tally_records <- spread(tally_records, `Number of records`, `Number of practices`)

tally_records <- formatted_flext(tally_records)
tally_records <- align(tally_records, align = "right", part = "all")
tally_records <- align(tally_records, j= 1, align = "left", part = "all")
```

Officer and flextable packages to create word documents

```r
my_doc <- read_docx("./Health Checks and Smoking/Word_doc_HC_Smoking.docx") %>%
  body_add_img(src = "./Research_Unit.png", width = 2, height = 0.6, style = "Normal") %>%
  body_add_par(paste0("NHS Health Checks GP records data report; ", min(as.character(unique(GP_HC_1920$Month_Year))), " to ", latest_month), style = "heading 2") %>%
  body_end_section_continuous() %>%
  body_add_par("Author:", style = "heading 2") %>%
  body_add_par("Rich Tyler", style = "Normal") %>%
  body_add_par("Updated:", style = "heading 2") %>%
  body_add_par(paste0(format(Sys.time(), '%d %B %Y')), style = "Normal") %>%
  body_end_section_columns(widths = c(2.5, 2.5), space = .05, sep = FALSE) %>%
  body_add_par("Records check", style = "heading 2") %>%
  body_add_par(paste0("Records are anticipated for ", nrow(GP_codes_HC), " GP providers alth
some providers may start or stop doing health check activity part way through the year. As
the number of records expected each month may vary. Moreover, as some providers enter data
or incorrectly enter the wrong date, there can be duplicate records in the database."), sty
"Normal") %>%
```

```r
print(my_doc, target = paste0("//teamspace.westsussex.gov.uk/DavWWWRoot/teams/PH/HLT/Protected/Healthy Lifestyles/Health Check and Smoking Performance/HC Activity Returns 201920/NHS Health Check GP data quality report 1920.docx"))
```

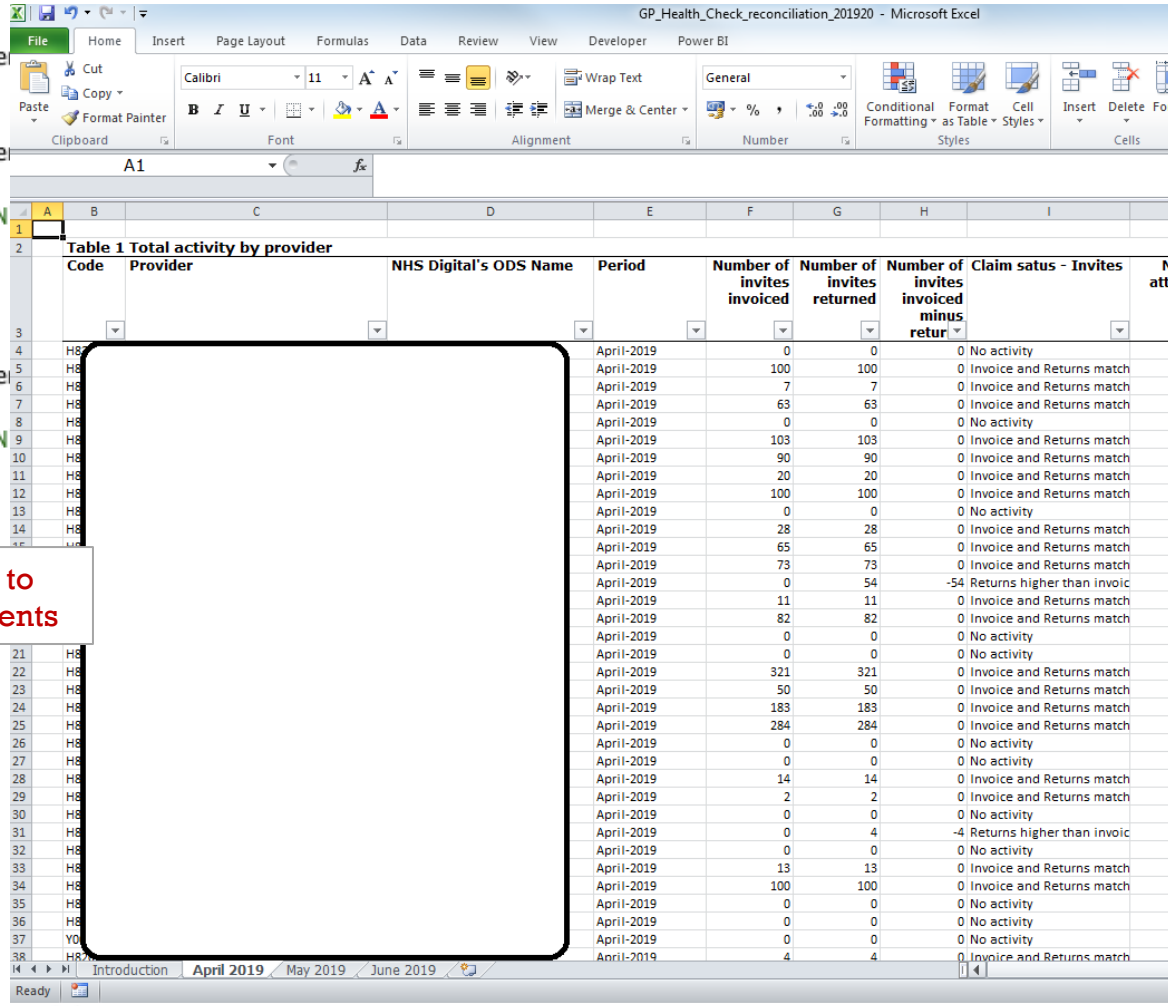


```r
# Styles for the data table row/column names
cs_title <- CellStyle(wb) +
  Font(wb, heightInPoints = 12, isBold = TRUE, isItalic = FALSE, name="Verdana") +   Alignme
(horizontal = "ALIGN_LEFT", wrapText = FALSE)
cs_left <- CellStyle(wb) +
  Font(wb, heightInPoints = 11, isBold = TRUE, isItalic = FALSE, name="Verdana") +   Alignme
(horizontal = "ALIGN_LEFT", vertical = "VERTICAL_TOP", wrapText = TRUE) +
  Border(color = "black", position = c("TOP", "BOTTOM"), pen = c("BORDER_THIN", "BORDER_THIN
cs_left_cell <- CellStyle(wb) +
  Font(wb, heightInPoints = 11, isBold = FALSE, isItalic = FALSE, name="Verdana") +
  Alignment(horizontal = "ALIGN_LEFT", vertical = "VERTICAL_TOP", wrapText = TRUE)
cs_right <- CellStyle(wb) +
  Font(wb, heightInPoints = 11, isBold = TRUE, isItalic = FALSE, name="Verdana") +   Alignme
(horizontal = "ALIGN_RIGHT", vertical = "VERTICAL_TOP", wrapText = TRUE) +
  Border(color = "black", position = c("TOP", "BOTTOM"), pen = c("BORDER_THIN", "BORDER_THIN

cs_thousand_sep <- CellStyle(wb) +
  DataFormat("#,##0")


# Data table ####
sheet <- createSheet(wb, sheetName = "April 2019")
rows  <- createRow(sheet, rowIndex=1:2000)
cells <- createCell(rows, colIndex=1:25)

setCellValue(cells[[2,2]], "Table 1 Total activity by provider")

addMergedRegion(sheet, 2, 2, 2, 14)
setCellStyle(cells[[2,2]], cs_title)

addDataFrame(April_19, sheet, startRow = 4, startColumn = 2, col.names = FALSE, row.names = FALSE)

setColumnWidth(sheet, colIndex = 1, colWidth = 5)
setColumnWidth(sheet, colIndex = 2, colWidth = 10)
setColumnWidth(sheet, colIndex = 3, colWidth = 42)
setColumnWidth(sheet, colIndex = 4, colWidth = 33)
setColumnWidth(sheet, colIndex = 5, colWidth = 18)
```

198:1 # 2019/20 R Scri

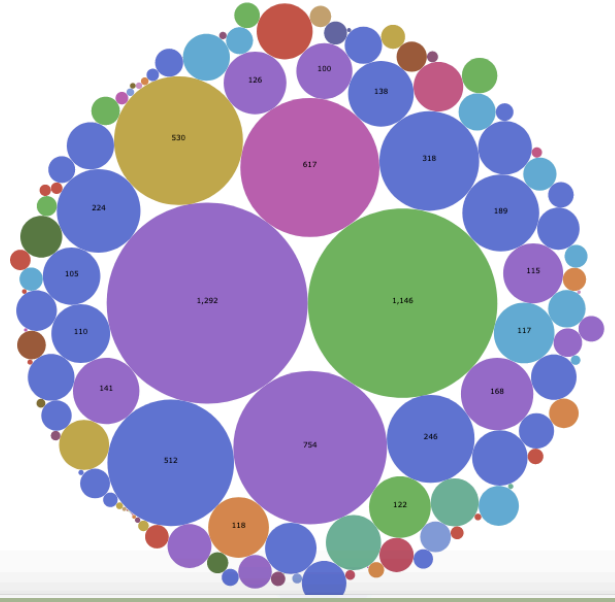
Console

Xlsx and openxlsx packages to create and edit excel documents

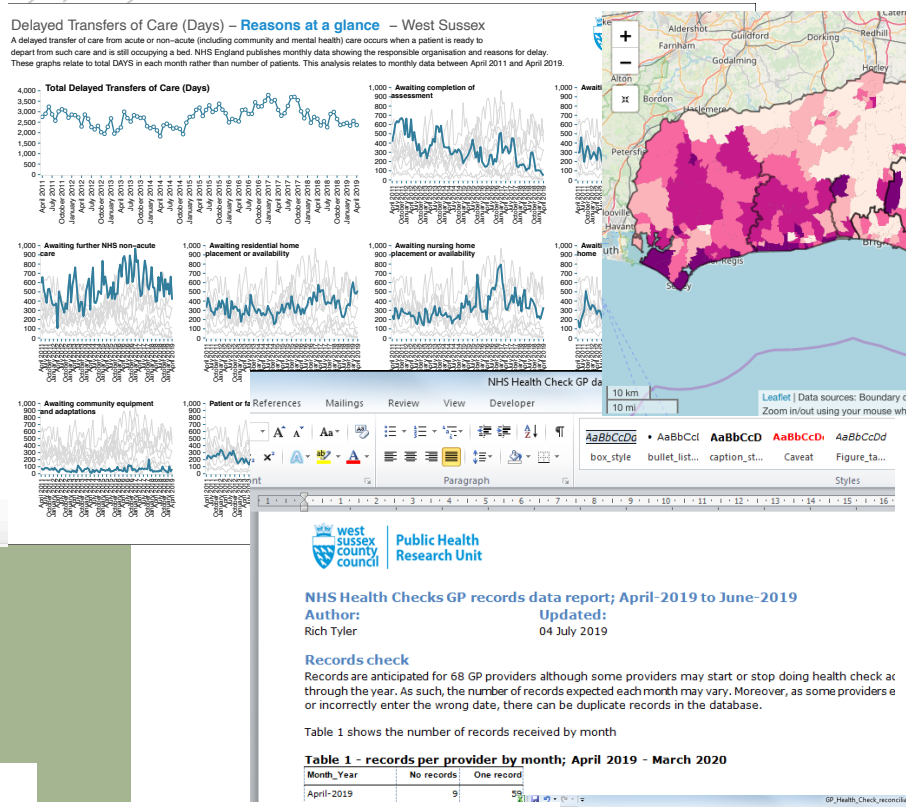


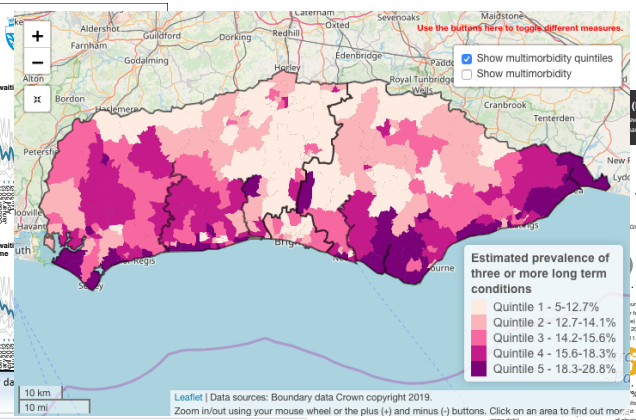

```r
addAutoFilter(sheet,  paste0("B3:", subset(alphabet, num ==
ncol(April_19)+1, select = "let")[[1]], nrow(April_19)+3))
```

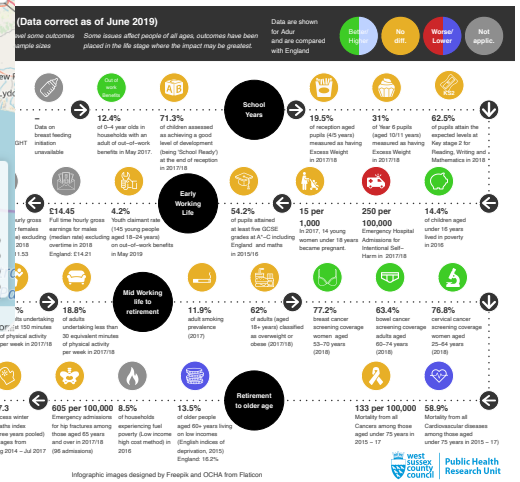
```r
saveWorkbook(wb, file = paste("//teamspace.westsussex.gov.uk
/teams/PH/HLT/Protected/Healthy Lifestyles/Health Check and
Smoking Performance/HC Activity Returns 201920/GP_Health_Check
_reconciliation_201920.xlsx", sep = ""))
```

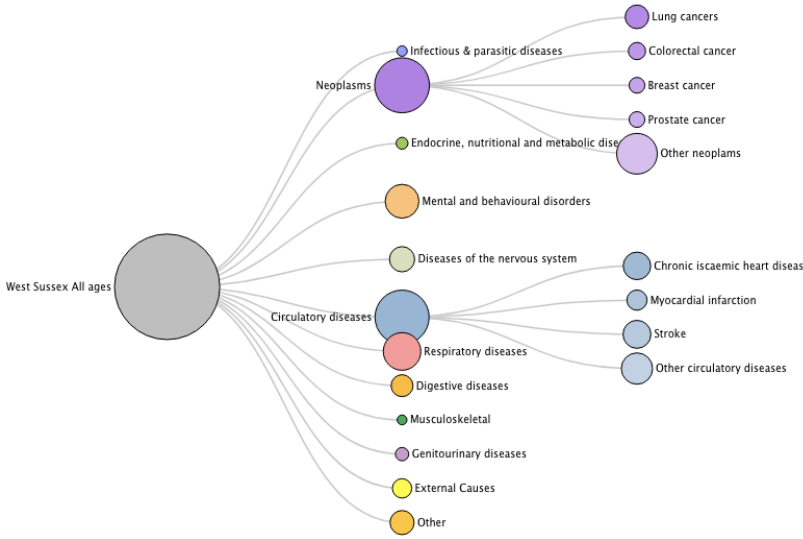


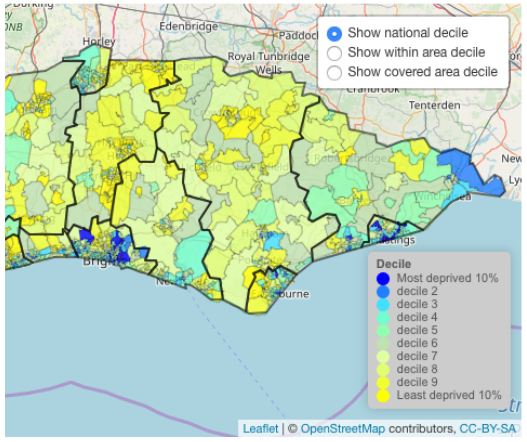
Use R

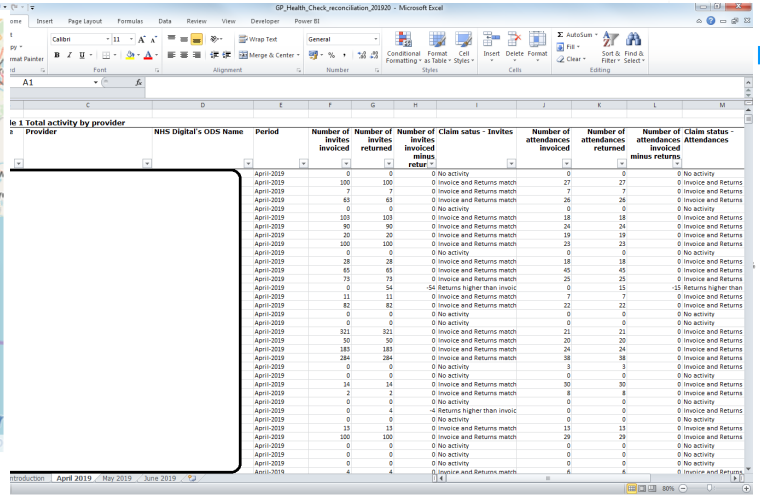


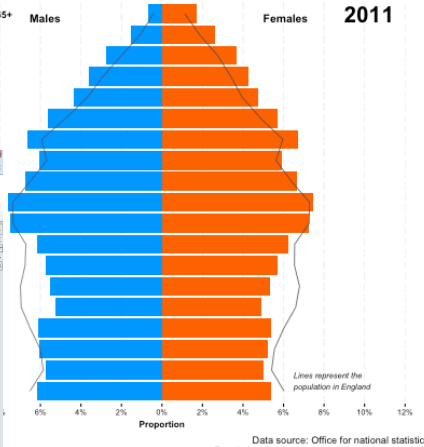












Data source: Office for national statistics
Population figures are rounded to the nearest 10.

What about R for outputs?

- Reasons for using R
- Output examples
  - Infographics
  - Reports
  - Excel outputs (static and interactive)
- Approaches
  - Reading and writing data
  - Strings
  - Loops
  - If statements

- This is not a lesson in R
  - However…
  - Public github repository –
    - https://github.com/psychty

Email – rich.tyler.2018@Hotmail.com