Keep data tidy

A plea to data publishers and developers

Julian Flowers, Seb Fox, James Perry

2018-06-23

knitr::opts\_chunk$set(echo = TRUE, message = FALSE, cache = TRUE, warning = FALSE)  
  
if(!require("pacman")) install.packages("pacman")

## Loading required package: pacman

library(pacman)  
p\_load(readxl, tidyverse, corrplot, tmap, geojsonio, downloader)

My colleague recently blogged about a [“wrangle free world”](https://nhsrcommunity.com/blog/aiming-for-a-wrangle-free-or-reduced-world/). This is the notion that a significant proportion of analytical time is devoted to cleaning, reshaping and reorganising data so it is ready for analysis, rather than analysing it, and that this effort would be significantly reduced if data were made available in easier to use formats.

One approach to this which has gained considerable attention recently is the *tidy data* framework. (Wickham 2014). This borrows heavily from database concepts like [Codd’s rules of normalisation](https://en.wikipedia.org/wiki/Codd%27s_12_rules) and introduces the idea of tidy datasets.

Data is tidy if:

* there is one observation per row
* there is one variable per column
* there is on observational unit per table.

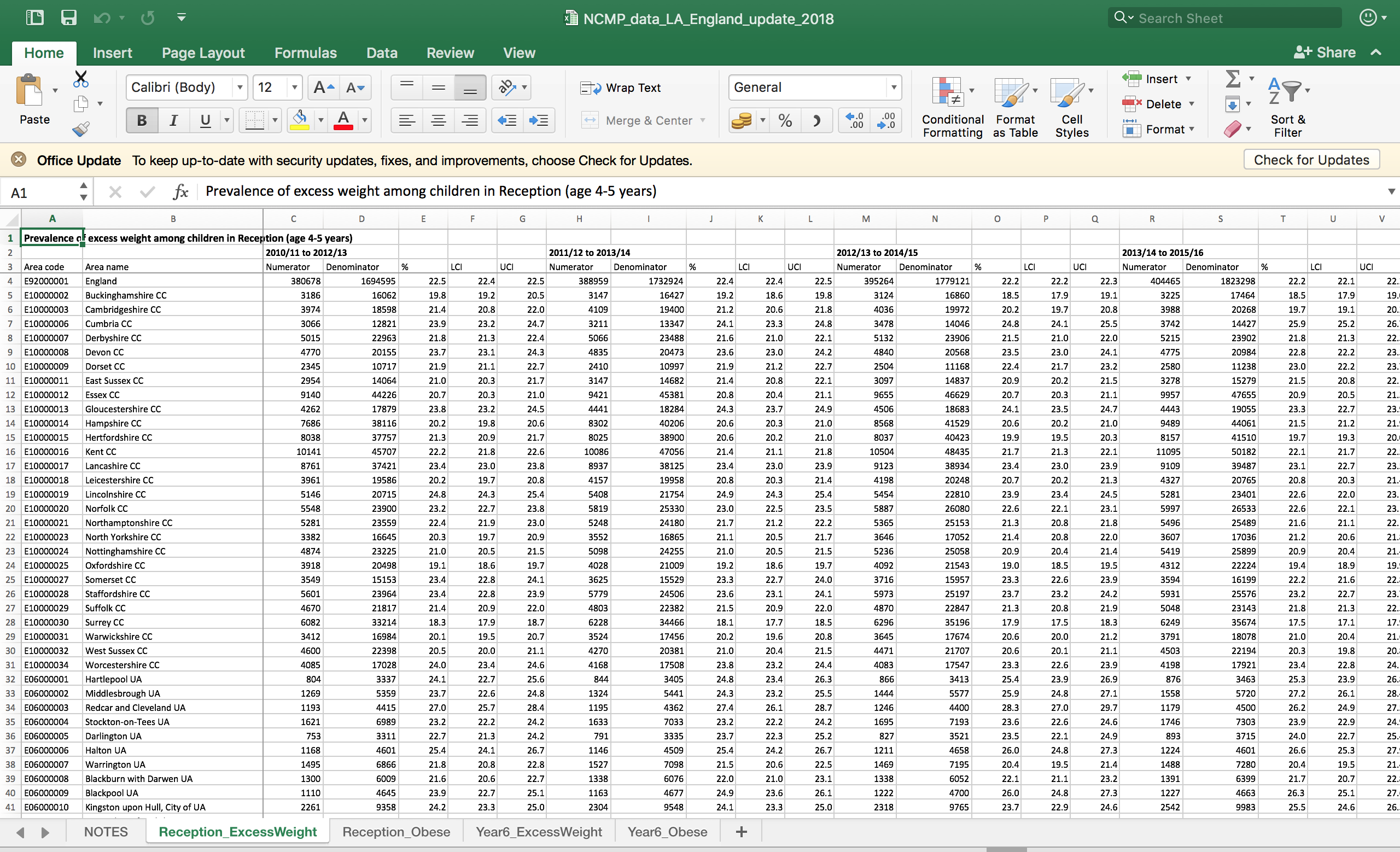
As Wickham says:

A huge amount of effort is spent cleaning data to get it ready for analysis, but there has been little research on how to make data cleaning as easy and effective as possible. This paper tackles a small, but important, component of data cleaning: data tidying. Tidy datasets are easy to manipulate, model and visualize, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table. This framework makes it easy to tidy messy datasets because only a small set of tools are needed to deal with a wide range of un-tidy datasets. This structure also makes it easier to develop tidy tools for data analysis, tools that both input and output tidy datasets. The advantages of a consistent data structure and matching tools are demonstrated with a case study free from mundane data manipulation chores.

Following this a slew of *R* packages has emerged which use this framework and the “go to” data tidying package from Wickham himself is called the tidyverse. This contains the functions of the workhouse data wrangling packages dplyr and tidyr.

Formatted spreadsheets with multiple headers and multiple sheets may have some benefits for tabular data presentation, but they can be unhelpful from an analytical point of view. They can make it difficult to conduct secondary analysis and may waste time and effort in reformatting and reshaping data, and may introduce risk and error from cutting and pasting. They are *untidy*, and if data publishers must continue to put out formatted excel files, we contend they should publish a tidy text or csv formatted file alongside.

Why bother? Let’s look at an example. PHE have just published small area data on childhood obesity on [.GOV.UK](https://www.gov.uk/government/statistics/child-obesity-and-excess-weight-small-area-level-data). This consists of 4 (large) workbooks containing data for MSOAs, electoral wards, CCGs and local authorities. Each workbook has an introductory sheet, and 4 sheets containing data. An example is shown below.



It looks nicely laid out - there are 4 sheets, one for each of obesity and excess weight in reception year and year 6 - but what if we want to look at the relationships between obesity in reception year and year 6 over time?

We would have to cut and paste several sets of data from 2 of the sheets into a new workbook or sheet. We would have to repeat this if we also wanted to look at the same relationship for excess weight.

It is difficult to do this analysis in Excel and to conduct this analysis in R is made unnecessarily difficult by the fact that sheets have multiple headers. Any analysis of these data would be greatly assisted by publishing a tidy version of the data. This means having a single row for each observation and a single column for each variable.

These data can be imported into R and reshaped using readxl and tools from the tidyverse package.

## Download data  
obesity <- download("https://www.gov.uk/government/uploads/system/uploads/attachment\_data/file/701873/NCMP\_data\_MSOA\_update\_2018.xlsx", "ncmp\_msoa.xlsx", mode = "wb")  
  
## Identify sheets  
  
obesity\_sheets <- excel\_sheets("ncmp\_msoa.xlsx")  
  
obesity\_sheets <- obesity\_sheets[2:5]  
  
## Import sheets  
## try one sheet  
  
obesity\_data <- read\_excel("ncmp\_msoa.xlsx", sheet = obesity\_sheets[1], skip = 2, na = "s")  
glimpse(obesity\_data)

## Observations: 6,791  
## Variables: 29  
## $ `MSOA code` <chr> "E02006534", "E02006535", "E02006536", "E020065...  
## $ `MSOA name` <chr> "Adur 001", "Adur 002", "Adur 003", "Adur 004",...  
## $ `LA code` <chr> "E07000223", "E07000223", "E07000223", "E070002...  
## $ `LA name` <chr> "Adur", "Adur", "Adur", "Adur", "Adur", "Adur",...  
## $ Numerator <dbl> 60, 36, 47, 64, 53, 52, 25, 46, 74, 31, 58, 42,...  
## $ Denominator <dbl> 238, 205, 198, 314, 282, 202, 143, 205, 256, 14...  
## $ `%` <dbl> 25.21008, 17.56098, 23.73737, 20.38217, 18.7943...  
## $ LCI <dbl> 20.11713, 12.96292, 18.34627, 16.29702, 14.6654...  
## $ UCI <dbl> 31.09057, 23.35241, 30.12814, 25.18324, 23.7619...  
## $ Numerator\_\_1 <dbl> 53, 43, 34, 71, 45, 37, 23, 47, 70, 38, 60, 52,...  
## $ Denominator\_\_1 <dbl> 206, 186, 167, 314, 241, 175, 134, 182, 261, 15...  
## $ `%\_\_1` <dbl> 25.72816, 23.11828, 20.35928, 22.61146, 18.6722...  
## $ LCI\_\_1 <dbl> 20.24131, 17.64051, 14.95100, 18.33178, 14.2578...  
## $ UCI\_\_1 <dbl> 32.10367, 29.68396, 27.10054, 27.55319, 24.0695...  
## $ Numerator\_\_2 <dbl> 47, 47, 35, 71, 42, 44, 28, 43, 79, 44, 57, 63,...  
## $ Denominator\_\_2 <dbl> 235, 215, 194, 329, 270, 193, 138, 198, 290, 17...  
## $ `%\_\_2` <dbl> 20.00000, 21.86047, 18.04124, 21.58055, 15.5555...  
## $ LCI\_\_2 <dbl> 15.38675, 16.85637, 13.26775, 17.47690, 11.7189...  
## $ UCI\_\_2 <dbl> 25.57827, 27.85246, 24.05581, 26.34020, 20.3585...  
## $ Numerator\_\_3 <dbl> 45, 45, 37, 82, 45, 55, 31, 44, 87, 42, 69, 76,...  
## $ Denominator\_\_3 <dbl> 244, 222, 211, 346, 277, 187, 139, 198, 305, 17...  
## $ `%\_\_3` <dbl> 18.44262, 20.27027, 17.53555, 23.69942, 16.2454...  
## $ LCI\_\_3 <dbl> 14.07863, 15.50855, 12.99810, 19.52287, 12.3684...  
## $ UCI\_\_3 <dbl> 23.78487, 26.04337, 23.23395, 28.45357, 21.0459...  
## $ Numerator\_\_4 <dbl> 41, 48, 50, 84, 53, 60, 45, 49, 85, 40, 67, 82,...  
## $ Denominator\_\_4 <dbl> 273, 263, 249, 360, 323, 228, 158, 233, 287, 16...  
## $ `%\_\_4` <dbl> 15.01832, 18.25095, 20.08032, 23.33333, 16.4086...  
## $ LCI\_\_4 <dbl> 11.267528, 14.050999, 15.576186, 19.259832, 12....  
## $ UCI\_\_4 <dbl> 19.73992, 23.36502, 25.49361, 27.96993, 20.8379...

Note that the sheets use an “s” to denote suppressed data due to small numbers. These data are missing from dataset so I have replaced them with NA on import.

The next steps are:

1. Import all sheets and turn into a single data frame
2. Add metadata
3. Convert to tidy data format

We can achieve the import and metadata steps with a *for* loop.

df <- data.frame()  
  
for (sheet in obesity\_sheets){  
   
 obesity\_data <- read\_excel("ncmp\_msoa.xlsx", sheet = sheet, skip = 2, na = "s") %>% mutate(ind = sheet)  
   
 df1 <- obesity\_data %>% select(ind, everything()) %>% gather(metric, value, Numerator:ncol(.))  
   
 df <- bind\_rows(df, df1)   
}

Now we have a dataset of 814920 rows and 7 columns. Just 2 more steps:

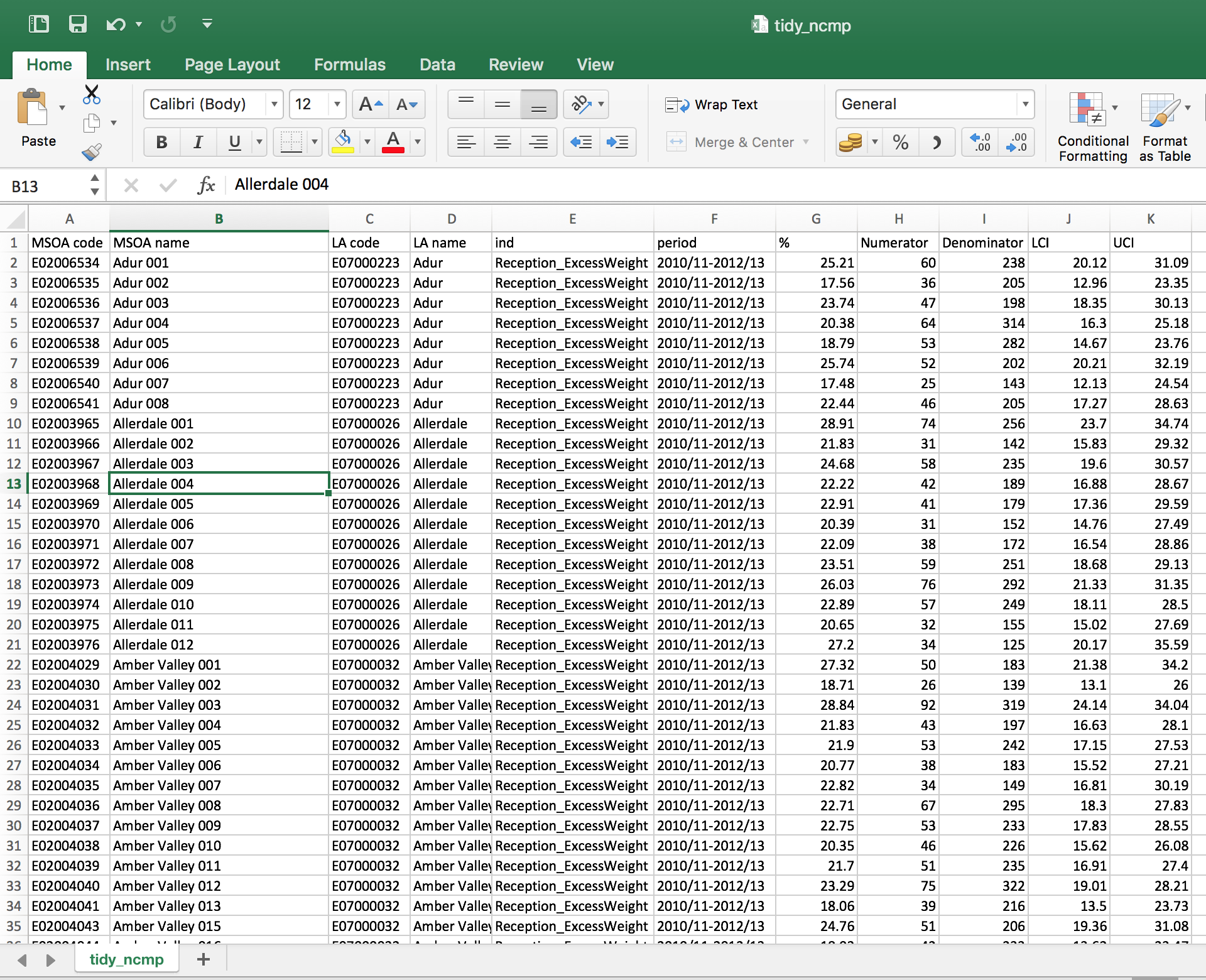
1. Add time periods
2. Tidy variable labels for the metric field and round data

df %>%  
 group\_by(ind, metric) %>%  
 count()

## # A tibble: 120 x 3  
## # Groups: ind, metric [120]  
## ind metric n  
## <chr> <chr> <int>  
## 1 Reception\_ExcessWeight % 6791  
## 2 Reception\_ExcessWeight %\_\_1 6791  
## 3 Reception\_ExcessWeight %\_\_2 6791  
## 4 Reception\_ExcessWeight %\_\_3 6791  
## 5 Reception\_ExcessWeight %\_\_4 6791  
## 6 Reception\_ExcessWeight Denominator 6791  
## 7 Reception\_ExcessWeight Denominator\_\_1 6791  
## 8 Reception\_ExcessWeight Denominator\_\_2 6791  
## 9 Reception\_ExcessWeight Denominator\_\_3 6791  
## 10 Reception\_ExcessWeight Denominator\_\_4 6791  
## # ... with 110 more rows

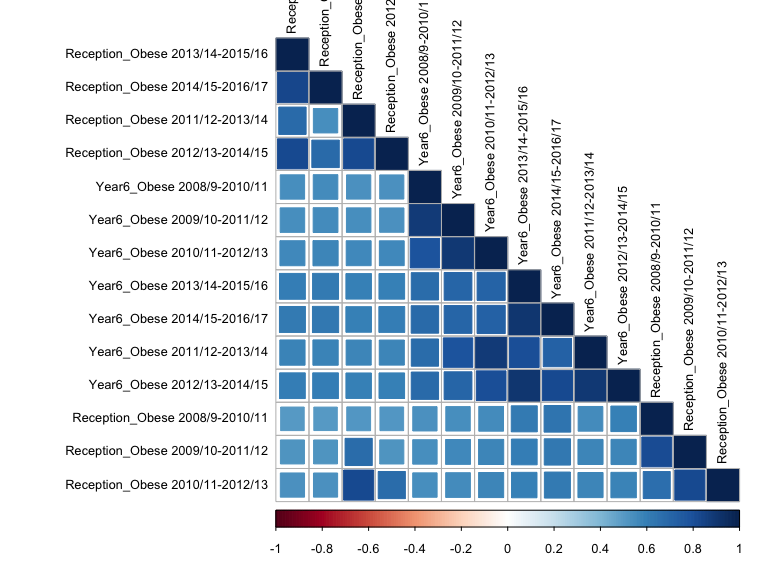
df1 <- df %>%  
 separate(metric, remove = FALSE, c("metric1", "period"), sep = "\_\_") %>%  
 mutate(value = round(value, 2),   
 period = case\_when(str\_detect(ind, "Excess") & is.na(period) ~ "2010/11-2012/13", str\_detect(ind, "Excess") & period == "1" ~ "2011/12-2013/14",  
 str\_detect(ind, "Excess") & period == "2" ~ "2012/13-2014/15",  
 str\_detect(ind, "Excess") & period == "3" ~ "2013/14-2015/16",  
 str\_detect(ind, "Excess") & period == "4" ~ "2014/15-2016/17",  
 str\_detect(ind, "Obese") & is.na(period) ~ "2008/9-2010/11", str\_detect(ind, "Obese") & period == "1" ~ "2009/10-2011/12",  
 str\_detect(ind, "Obese") & period == "2" ~ "2010/11-2012/13",  
 str\_detect(ind, "Obese") & period == "3" ~ "2011/12-2013/14",  
 str\_detect(ind, "Obese") & period == "4" ~ "2012/13-2014/15",  
 str\_detect(ind, "Obese") & period == "5" ~ "2013/14-2015/16",  
 str\_detect(ind, "Obese") & period == "6" ~ "2014/15-2016/17"  
 )  
 )  
  
  
df\_wide <- df1 %>%  
 select(`MSOA code`, `MSOA name`, `LA code`, `LA name` , ind, metric1, period, value) %>%   
 spread(metric1, value) %>%  
 select(c(`MSOA code`:period, Numerator, Denominator, `%`, LCI, UCI ))  
  
  
  
## write\_csv(df\_wide, "tidy\_ncmp.csv") export tidied table.

Now we have a tidy data frame where each row is a single observation. It looks like this.



We can now look at the trend in the association of overweight and obesity between reception year and year 6 for example.

df1 %>%  
 filter(metric1 == "%") %>%  
 mutate(index = paste(ind, period)) %>%  
 select(-c(metric, period, ind)) %>%   
 #slice(c(1, 27165, 53791, 80698))  
 spread(index, value) %>%   
 mutate\_if(is.numeric, function(x) ifelse(is.na(x), median(x, na.rm = TRUE), x)) %>%  
 #select(`Reception\_ExcessWeight 2010/11-2012/13`:`Year6\_Obese 2008/9-2010/11`) %>%   
 select(contains("Obese")) %>%  
 #pairs(panel = panel.smooth)  
 cor(.) %>%  
 corrplot(tl.cex = .8, tl.col = "black", method = "square", order = "hclust", number.cex = .5, type = "lower")

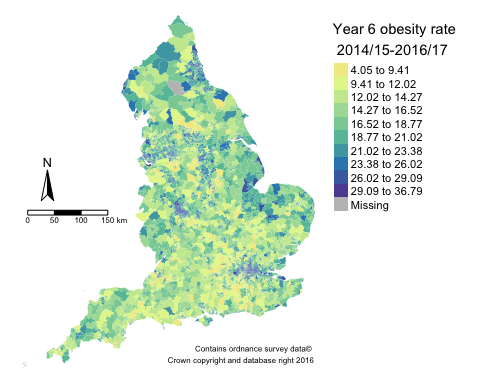


Obesity correlations

This shows that correlations for obesity are stronger in year 6 than in reception year and persist over time.

We can now also map the data easily.

palette <- RColorBrewer::brewer.pal(10, "Spectral")  
credits <- "Contains ordnance survey data© \nCrown copyright and database right 2016"  
  
## get boundary file for MSOAs from http://geoportal.statistics.gov.uk/datasets?q=MSOA\_Boundaries\_2011&sort=name  
  
shape <- "https://opendata.arcgis.com/datasets/826dc85fb600440889480f4d9dbb1a24\_2.geojson"  
  
shape <- geojson\_read(shape, what = "sp")  
  
shape <- subset(shape, substr(msoa11cd, 1, 1) == "E") ## just English MSOAs  
  
map\_data <- df1 %>%  
 filter(metric1 == "%", ind == "Year6\_Obese", period == "2014/15-2016/17")  
  
shape@data <- shape@data %>%  
 left\_join(map\_data, by = c("msoa11cd" = "MSOA code"))  
  
t <- tm\_shape(shape) +  
 tm\_fill("value", style = "kmeans" , n = 10,  
 palette = palette, title = "Year 6 obesity rate\n 2014/15-2016/17") +  
 tm\_credits( credits, size = 0.5, align = "right") +  
 tm\_layout(legend.outside = TRUE,   
 frame = FALSE) +  
 tm\_compass(position = c("left", "center")) +   
 tm\_scale\_bar(position = c("left", "center"))  
  
t



Map of MSOA obesity rates in year 6

And finally, we could repeat this for the other geographies and combine them all into a single dataset.

The markdown file for this note is available [here](https://github.com/julianflowers/random-chicken/edit/master/tidy_data_plea1.Rmd). Please feel free to suggest improvements.

# A plea to data publishers

Hopefully this vignette shows that although it is possible to reshape spreadsheets with multiple sheets and untidy data into a tidy format for analysis in R, it is a lot of effort. This could be ameliorated by data publishers in 3 simple steps.

1. Creating a tidy dataset in the first place and publish this as a .txt or .csv file alongside the spreadsheets
2. For missing data use NA or a numeric code (NOT 0) - or at least agreeing some common practice.
3. Publishing a code book.

This would make data more open, easier to reuse, and encourage reproducibility.

## Addendum

I have subsequently become aware of a new package tidyxl which may make this job easier. Also the author of the package has written a book about tidy data and wrangling spreadsheets (<https://nacnudus.github.io/spreadsheet-munging-strategies/tidy-clean.html>). And finally, the GSS and ONS are thinking along the same lines and are developing a set of Python tools. This is called Data Baker - <https://scraperwiki.com/2015/03/databaker-making-spreadsheets-usable/>.

## References

Wickham, Hadley. 2014. “Tidy data.” *Journal of Statistical Software* 46 (10):1–23. <https://doi.org/10.18637/jss.v059.i10>.