NHS Wales Data

Mark Baber

2021

# Web Scraping NHS Wales

This report will go over the steps taken to web scrpe data from an NHS Wales website. This involves:

* Downloading the files into rstudio
* Carrying out some EDA
* Plotting the data
* Feature Extraction/Creation

## 0 - Set a Working Directory

Usually when working within R, I start by creating a new folder and a project file, this can be done via the gui (top right of rstudio) or via the console.

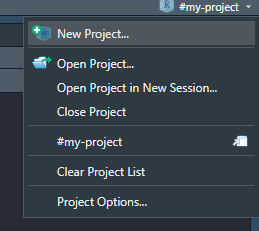


Figure : Create Project Via RStudio

getwd()

## [1] "C:/Users/markb/OneDrive/university/msc-work/msc-project-data-science/#my-project"

Whilst I often switch between operating systems (Linux & Windows) this isn’t always an ideal way of setting the project, especially when using online cloud storage. This works much better via Git && Github.

## 1 - EDA

Firstly, when working with scripts and data science coding, it is usually very helpful to set the seed, for reproducibility.

For the seed I set it as my student number from USW - but can be anything.

# 17076749  
set.seed(17076749)

Next would be to set the libraries which could be useful throughout this report, most jobs within r can be done with the base set libraries, but sometimes it’s fun to explore other methods which make some tasks a lot easier/faster. These packages aren’t set in stone, and are dynamic for this report.

Next would be to import the data, this part proved to be quite difficult as I kept getting different errors when trying to download the files locally. I will add the code just in case I find something which sparks an idea.

|  |
| --- |
| **Meta Data** |

[GP Practice Analysis and Patient Registrations by Practice](https://nwssp.nhs.wales/ourservices/primary-care-services/general-information/data-and-publications/gp-practice-analysis/)

The data shows the number of items prescribed by each practice by month and the number of patients registered with each practice.

Data relating to patient registrations by practice is extracted from NHAIS (National Health Application and Infrastructure Services) system each quarter.

The Health Board, practice code, postcode and count by age band and gender are included. Please note that patients 95 and over have been grouped together due to potential risk of disclosure.

With a better idea of the data after looking at the meta data above, this will give us a better idea of what to look for within the dataset - and a good reminder for us throughout the report.

# set the url to test  
url <- "https://nwssp.nhs.wales/ourservices/primary-care-services/primary-care-services-documents/gp-practice-analysis-docs/patient-registrations-july-2021"   
# notice it doesn't end with .xlsx or .xls  
  
# set a destination of where to save the file (with extension?)  
destfile <- "data/testfile"  
  
# try to download the file with built in download.file  
download.file(url, # what to save  
 destfile, # where to save  
 mode="wb")# wb convert it to 2 cols  
  
# After some more research, I found some documentation saying about how   
# difficult using XLSX files can be within R.  
  
# browseURL('http://j.mp/2aFZUrJ')  
  
# found a package rio 'R, Input, Output'  
# will try to use this for the XLSX files.  
# library(rio)  
# import(url, format = "xlsx")  
# rio still didn't fix the issue.

It looks like the download works - lets try to import the data as a dataframe and have a look - this didn’t work with the built in r functions, and the package *rio* also didn’t help with this. My thought is that the file is failing to download correctly, but i’m unsure why (as I can download the package via the command line (Bash) or by clicking on it.)

rm(df)

## Warning in rm(df): object 'df' not found

df <- read\_excel(path = "data/testfile")  
head(df)

## # A tibble: 6 x 2  
## `Row Labels` `Sum of count`  
## <chr> <dbl>  
## 1 W00005 8942  
## 2 CF83 4AZ 8942  
## 3 Aneurin Bevan 8942  
## 4 W00007 7862  
## 5 LL11 3NS 7862  
## 6 Betsi Cadwaladr Uni 7862

This is the best I could do whilst trying to download a file from the NHS Wales website, this was done with the extension ‘.xlsx’ and without. So I will download them **manually** by just clicking the links on the website and saving them to a data folder within my project.

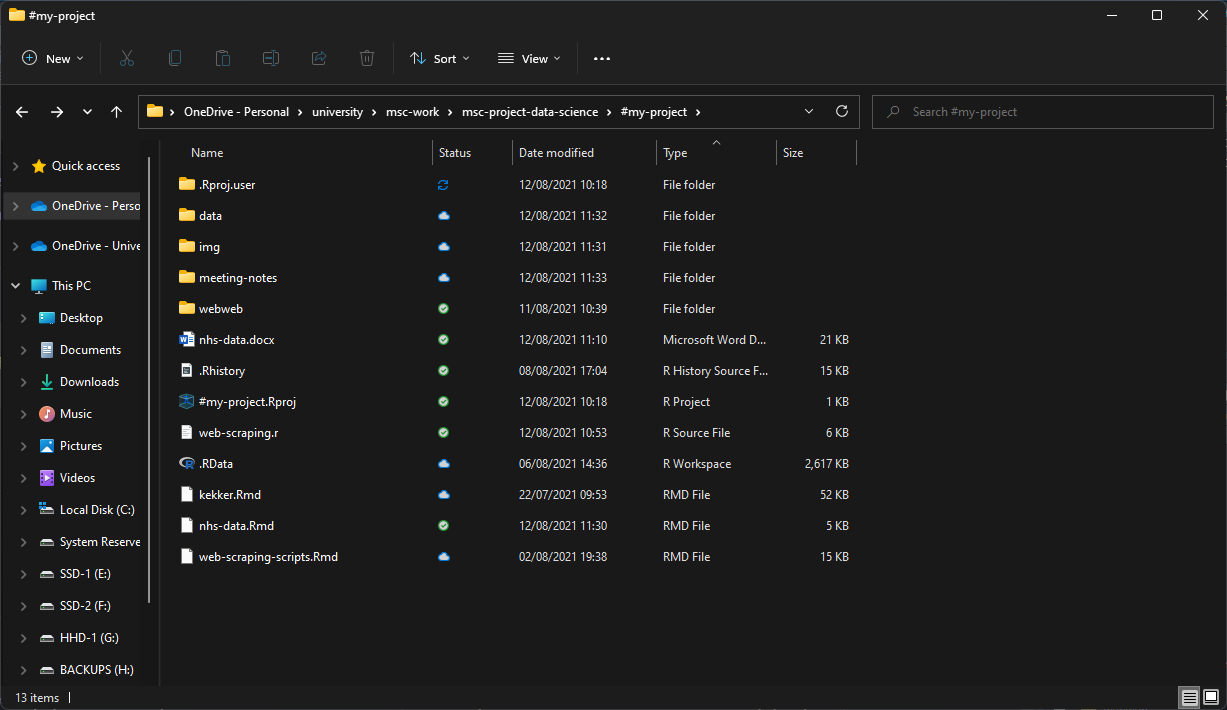


Figure : Project Folder Layout

After creating a data folder and an img folder (for screenshots), now we can read the saved datasets from that data folder with read\_excel. Doing them all at once is quite demanding, but it’s not too bad.

Now that we have created 6 data frames each containing a month of GP visits, these are taken quarterly by the look at: - Jan - April - July - October

|  |
| --- |
| **Combining Data Frames** |

Whilst these 6 datasets could be explored one at a time, lets try to combine these datasets and explore the data as a whole. First we would need to inspect the datasets to make sure they’re following the same structure throughout. An easy way to check the columns is by using the function *names*.

print(names(df.april20))

## [1] "Period" "PracticeCode" "PostCode"   
## [4] "OrgCode" "AgeBand" "MaleCount"   
## [7] "FemaleCount" "IndeterminateCount" "Count"

print(names(df.july20))

## [1] "Period" "PracticeCode" "PostCode"   
## [4] "OrgCode" "AgeBand" "MaleCount"   
## [7] "FemaleCount" "IndeterminateCount" "Count"

print(names(df.october20))

## [1] "Period" "PracticeCode" "PostCode"   
## [4] "OrgCode" "AgeBand" "MaleCount"   
## [7] "FemaleCount" "IndeterminateCount" "Count"

print(names(df.jan21))

## [1] "Period" "PracticeCode" "PostCode"   
## [4] "OrgCode" "AgeBand" "MaleCount"   
## [7] "FemaleCount" "IndeterminateCount" "Count"

print(names(df.april21))

## [1] "Period" "PracticeCode" "postcode"   
## [4] "OrgCode" "HAName" "AgeBand"   
## [7] "MaleCount" "Femalecount" "Indeterminatecount"  
## [10] "count"

print(names(df.july21))

## [1] "Period" "PracticeCode" "postcode"   
## [4] "OrgCode" "HAName" "AgeBand"   
## [7] "MaleCount" "Femalecount" "Indeterminatecount"  
## [10] "count"

Here we can see that the last 2 data frames have an additional column, *HAName*. From what I can tell, this is the Local Health Board which is also denoted by the OrgCode - To make things easier, lets grab the Local Health Boards, figure out which one refers to which code, and replace the code with the LHB.

To begin, lets create a data frame and get the unique Health Boards.

# create a data frame.  
df <- data.frame()  
# get unique HealthBoards  
uHBs <- unique(df.april21$HAName)  
# print LHBs  
print(uHBs)

## [1] "Aneurin Bevan" "Betsi Cadwaladr Uni" "Hywel Dda"   
## [4] "Cardiff And Vale Uni" "Cwm Taf Morgannwg UHB" "Swansea Bay UHB"   
## [7] "Powys Teaching"

Now we can see the LHBs, we can use these to filter one of the newer data frames which has the HAName/LHB.

# Filter the last data frame with the unique health board  
# as a new dataframe  
my.AB <- filter(df.july21, HAName == uHBs[1]) # Aneurin Bevan  
my.BC <- filter(df.july21, HAName == uHBs[2]) # Betsi Cadwaladr Uni  
my.HD <- filter(df.july21, HAName == uHBs[3]) # Hywel Dda  
my.CV <- filter(df.july21, HAName == uHBs[4]) # Cardiff And Vale Uni  
my.CTM <- filter(df.july21, HAName == uHBs[5]) # Cwm Taf Morgannwg UHB  
my.SB <- filter(df.july21, HAName == uHBs[6]) # Swansea Bay UHB  
my.PT <- filter(df.july21, HAName == uHBs[7]) # Powys Teaching

Above we have filtered out the LHBs from the latest data frame, next we can check each filtered data frame for the OrgCode.

# check if they all have a unique OrgCode  
print(paste('AB = ', unique(my.AB$OrgCode)))

## [1] "AB = 7A6"

print(paste('BC = ', unique(my.BC$OrgCode)))

## [1] "BC = 7A1"

print(paste('HD = ', unique(my.HD$OrgCode)))

## [1] "HD = 7A2"

print(paste('CV = ', unique(my.CV$OrgCode)))

## [1] "CV = 7A4"

print(paste('CTM = ', unique(my.CTM$OrgCode)))

## [1] "CTM = 7A5"

print(paste('SB = ', unique(my.SB$OrgCode)))

## [1] "SB = 7A3"

print(paste('PT = ', unique(my.PT$OrgCode)))

## [1] "PT = 7A7"

Now we know: - 7A1 - Betsi Cadwaladr Uni - 7A2 - Hywel Dda - 7A3 - Swansea Bay UHB - 7A4 - Cardiff And Vale Uni - 7A5 - Cwm Taf Morgannwg UHB - 7A6 - Aneurin Bevan - 7A7 - Powys Teaching

Now that we have the HBs saved and filtered, we can drop those columns from the newer datasets, combine the datasets and then add the HBs back in (in place of the OrgCode). This is a bit over the top, and there is bound to be a better way of doing things.

# DROP HAName  
df.april21 <- df.april21[-5]  
df.july21 <- df.july21[-5]

Here we have dropped the 5th column which is HAName. Next lets look at getting the column names from the older datasets (double check they’re the same for the others) and make sure all column names are the same. *(I noticed the modern datasets didn’t have the same capitalisation on the columns, hence the next steps.)*

# get col names from a dataset from 2020  
colNames <- names(df.april20)  
  
# rename colnames with the same as older datasets  
colnames(df.april21) <- colNames  
colnames(df.july21) <- colNames

Now that the column names have been saved and replaced, we can bind all datasets (using rbind).

# Bind the datasets with rbind.  
# ?rbind  
  
df <- rbind(df.april20, df.july20, df.october20,  
 df.jan21, df.april21, df.july21)

Here we can see that the data frame was created without any issues. Lets replace the column OrgCode for the Health Board - and rename the column name afterwards.

# Replace OrgCode for LHBs.  
df["OrgCode"][df["OrgCode"] == "7A1"] <- "Betsi Cadwaladr Uni"  
df["OrgCode"][df["OrgCode"] == "7A2"] <- "Hywel Dda"  
df["OrgCode"][df["OrgCode"] == "7A3"] <- "Swansea Bay UHB"  
df["OrgCode"][df["OrgCode"] == "7A4"] <- "Cardiff And Vale Uni"  
df["OrgCode"][df["OrgCode"] == "7A5"] <- "Cwm Taf Morgannwg UHB"  
df["OrgCode"][df["OrgCode"] == "7A6"] <- "Aneurin Bevan"  
df["OrgCode"][df["OrgCode"] == "7A7"] <- "Powys Teaching"  
# Sanity check OrgCodes.  
df$OrgCode %>% unique()

## [1] "Aneurin Bevan" "Betsi Cadwaladr Uni" "Hywel Dda"   
## [4] "Cardiff And Vale Uni" "Cwm Taf Morgannwg UHB" "Swansea Bay UHB"   
## [7] "Powys Teaching"

Lets quickly rename OrgCodes to HealthBoard so it is a bit easier to understand.

# rename column 4  
names(df)[4] = "HealthBoard"  
# double check the column names  
print(names(df))

## [1] "Period" "PracticeCode" "PostCode"   
## [4] "HealthBoard" "AgeBand" "MaleCount"   
## [7] "FemaleCount" "IndeterminateCount" "Count"

Great - now we have a combined dataset (More datasets could be added to make things better, for example LSOA which will hopefully be explored later.) - Lets first look at the structure of the data and do some EDA.

|  |
| --- |
| **Expoloratory Data Analysis** |

str(df.april20)

## tibble [39,368 x 9] (S3: tbl\_df/tbl/data.frame)  
## $ Period : num [1:39368] 202004 202004 202004 202004 202004 ...  
## $ PracticeCode : chr [1:39368] "W00005" "W00005" "W00005" "W00005" ...  
## $ PostCode : chr [1:39368] "CF83 4AZ" "CF83 4AZ" "CF83 4AZ" "CF83 4AZ" ...  
## $ OrgCode : chr [1:39368] "7A6" "7A6" "7A6" "7A6" ...  
## $ AgeBand : num [1:39368] 0 1 2 3 4 5 6 7 8 9 ...  
## $ MaleCount : num [1:39368] 33 42 41 43 36 51 42 50 50 54 ...  
## $ FemaleCount : num [1:39368] 35 47 42 56 56 48 43 42 38 38 ...  
## $ IndeterminateCount: num [1:39368] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Count : num [1:39368] 68 89 83 99 92 99 85 92 88 92 ...

From the structure (str) we can see that there are 9 variables with 39,368 inputs. We can also look at the data via the head and tail.

head(df.april20)

## # A tibble: 6 x 9  
## Period PracticeCode PostCode OrgCode AgeBand MaleCount FemaleCount  
## <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 202004 W00005 CF83 4AZ 7A6 0 33 35  
## 2 202004 W00005 CF83 4AZ 7A6 1 42 47  
## 3 202004 W00005 CF83 4AZ 7A6 2 41 42  
## 4 202004 W00005 CF83 4AZ 7A6 3 43 56  
## 5 202004 W00005 CF83 4AZ 7A6 4 36 56  
## 6 202004 W00005 CF83 4AZ 7A6 5 51 48  
## # ... with 2 more variables: IndeterminateCount <dbl>, Count <dbl>

tail(df.april20)

## # A tibble: 6 x 9  
## Period PracticeCode PostCode OrgCode AgeBand MaleCount FemaleCount  
## <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 202004 W98056 SA1 8QY 7A3 NA 4 29  
## 2 202004 W98057 SA1 8QY 7A3 NA 8 17  
## 3 202004 W98608 SA1 4DF 7A3 NA 6 8  
## 4 202004 W98612 SA5 5LB 7A3 NA 1 1  
## 5 202004 W98627 SA12 9PY 7A3 NA 0 4  
## 6 202004 W98785 SA11 1EF 7A3 NA 0 6  
## # ... with 2 more variables: IndeterminateCount <dbl>, Count <dbl>

Along with the summary for some basic descriptive analysis.

summary(df.april20)

## Period PracticeCode PostCode OrgCode   
## Min. :202004 Length:39368 Length:39368 Length:39368   
## 1st Qu.:202004 Class :character Class :character Class :character   
## Median :202004 Mode :character Mode :character Mode :character   
## Mean :202004   
## 3rd Qu.:202004   
## Max. :202004   
##   
## AgeBand MaleCount FemaleCount IndeterminateCount   
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. :0.0000000   
## 1st Qu.:23.00 1st Qu.: 19.00 1st Qu.: 20.00 1st Qu.:0.0000000   
## Median :47.00 Median : 37.00 Median : 36.00 Median :0.0000000   
## Mean :46.75 Mean : 41.09 Mean : 41.27 Mean :0.0003048   
## 3rd Qu.:70.00 3rd Qu.: 56.00 3rd Qu.: 56.00 3rd Qu.:0.0000000   
## Max. :94.00 Max. :753.00 Max. :720.00 Max. :1.0000000   
## NA's :402   
## Count   
## Min. : 0.00   
## 1st Qu.: 40.00   
## Median : 74.00   
## Mean : 82.36   
## 3rd Qu.: 111.00   
## Max. :1473.00   
##

Summary is a very useful function as we can easily see the *min, median, mean, max* along with some quartiles - and most importantly we can see which columns have NA’s.. I’m looking at you **AgeBand**. 402 NA’s is quite a lot, so this would need to be dealt with as best as possible (this could be dropping all the values, even if it is quite a lot - or simply putting the ageband as an outlier so we can easily spot them.)

df.april20$AgeBand %>% unique()

## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24  
## [26] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49  
## [51] 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74  
## [76] 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 NA

# can confirm there are NAs within AgeBand  
  
df.april20$AgeBand %>% is.na() %>% sum()

## [1] 402

# with a count of 402 - summary was correct.

Even when looking at the unique values of the age, we can see that NA is in there. This could be because of a number of reasons, but there are 2 assumptions which stand out to me:

* Someone refused to give their age – which is fine
* They were over 94 which COULD have been set as a cut off point?

Lets now look at the unique values within each column.

* Period

There is 1 period of time which is 2020-04 (April)

df.april20$Period %>%   
 unique() %>% length()

## [1] 1

* Practice Code

There are 414 unique Practice Codes

df.april20$PracticeCode %>%   
 unique() %>% length()

## [1] 414

* Post Code

There are 389 unique PostCodes

df.april20$PostCode %>%   
 unique() %>% length()

## [1] 389

* Org Code

There are 7 unique Org Codes (Health-Boards)

df.april20$OrgCode %>%   
 unique() %>% length()

## [1] 7

* Age Band

There are 96 unique AgeBands - These are just Ages though, we can group them later.

df.april20$AgeBand %>%   
 unique() %>% length()

## [1] 96

* MaleCount

There are 235 unique MaleCount entries (Not sum or count)

df.april20$MaleCount %>%   
 unique() %>% length()

## [1] 235

* FemaleCount

There are 234 unique FemaleCount entries (Not sum or count)

df.april20$FemaleCount %>%   
 unique() %>% length()

## [1] 234

* IndeterminateCount

There are 2 unique IndeterminateCount entries (These could be intersex??)

df.april20$IndeterminateCount %>%   
 unique() %>% length()

## [1] 2

* Count

There are 406 unique entries for count (Not sum or count)

df.april20$Count %>%   
 unique() %>% length()

## [1] 406

Before moving it, it’s important to do something with the missing values within the AgeBand - this can be done by replacing the values with the mean value or just changing them to 0’s. Here, lets change them to the mean value.

df.april20$AgeBand[is.na(df.april20$AgeBand)] <- mean(df.april20$AgeBand, na.rm=TRUE)

Whilst some of the code above didn’t really show anything which stood out (mainly the unique counts) - there are many packages which I found when researching EDA in R with this one below being *skimr* which is great for a Data Summary.

# https://www.rdocumentation.org/packages/skimr/versions/2.1.3  
library(skimr)  
  
df.april20 %>%   
 skim() %>%   
 filter(numeric.sd > 1)

Table : Data summary

|  |  |
| --- | --- |
| Name | Piped data |
| Number of rows | 39368 |
| Number of columns | 9 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: numeric**

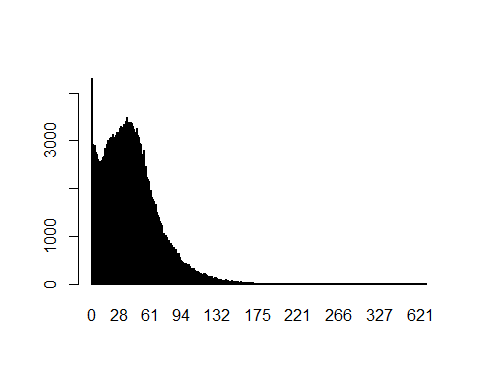
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AgeBand | 0 | 1 | 46.75 | 27.23 | 0 | 23 | 46.75 | 70 | 94 | ▇▇▇▇▇ |
| MaleCount | 0 | 1 | 41.09 | 30.63 | 0 | 19 | 37.00 | 56 | 753 | ▇▁▁▁▁ |
| FemaleCount | 0 | 1 | 41.27 | 30.55 | 0 | 20 | 36.00 | 56 | 720 | ▇▁▁▁▁ |
| Count | 0 | 1 | 82.36 | 60.15 | 0 | 40 | 74.00 | 111 | 1473 | ▇▁▁▁▁ |

Now that some basic analysis has been done on the oldest dataset, lets look at the rest of the datasets and try to combine them as one. First would be to make sure they all have the same columns.

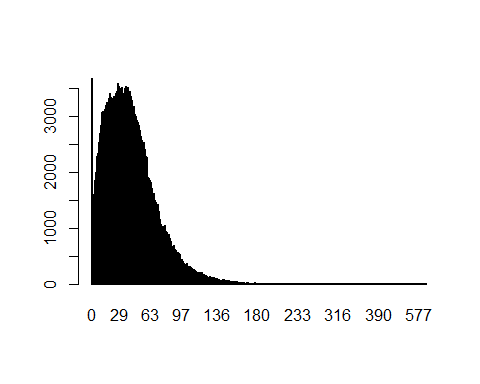
## 2 - Plots

Next lets try to get some plots of the data from 1 month from 2020.

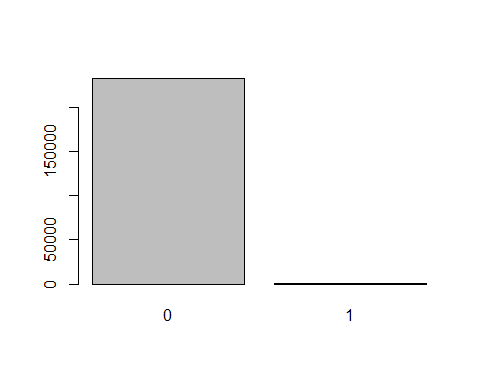
# change frequencies to categories with table  
m.count <- table(df$MaleCount)  
f.count <- table(df$FemaleCount)  
i.count <- table(df$IndeterminateCount)  
  
barplot(m.count)



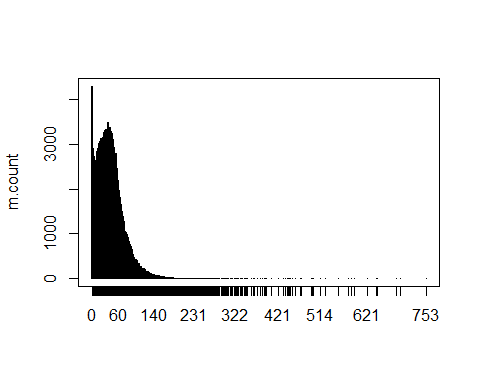
barplot(f.count)



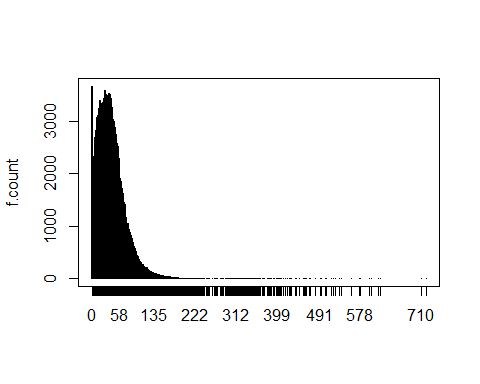
barplot(i.count)



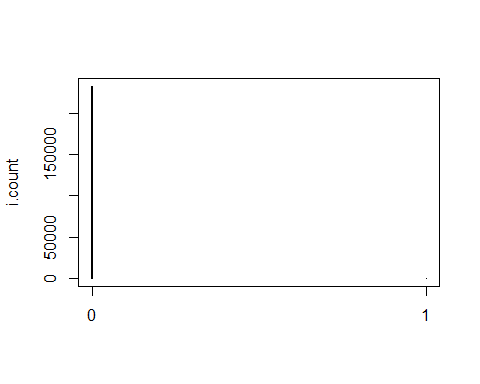
plot(m.count)



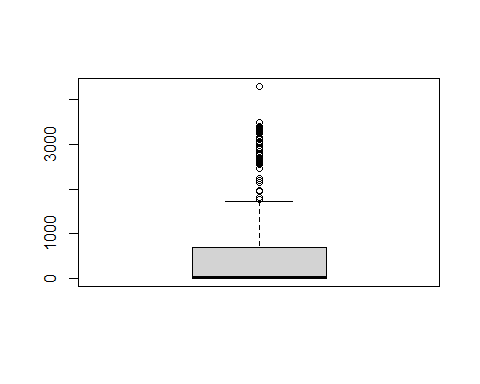
plot(f.count)



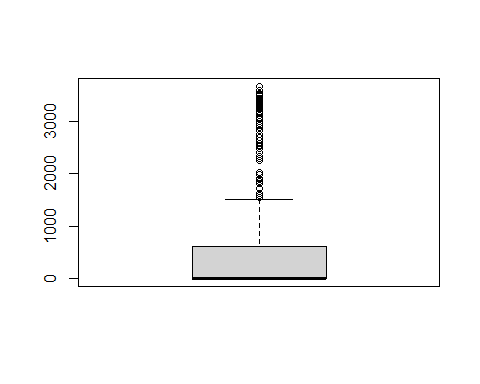
plot(i.count)



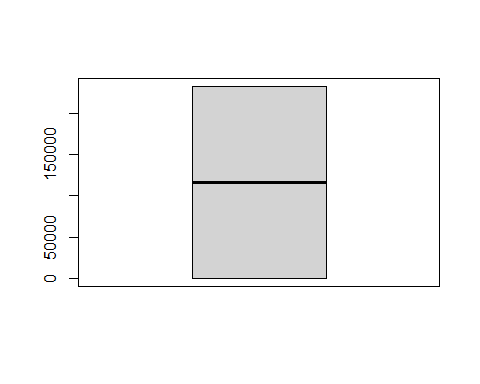
boxplot(m.count)



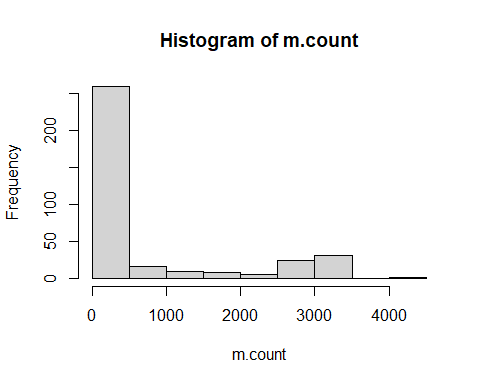
boxplot(f.count)



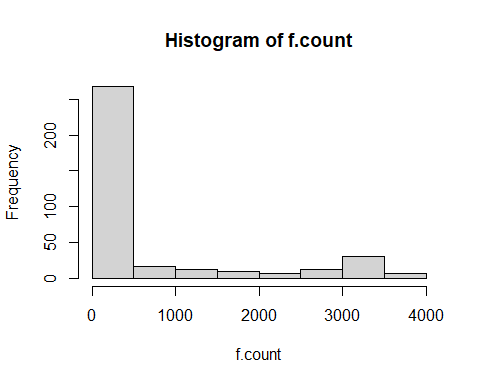
boxplot(i.count)



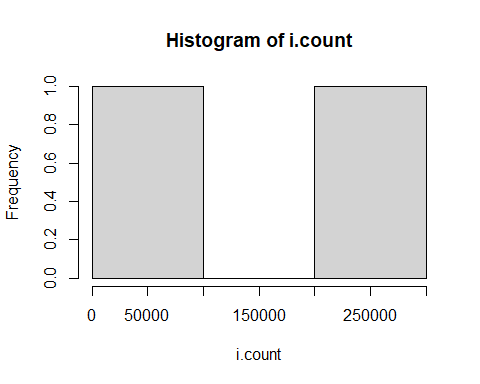
hist(m.count)



hist(f.count)



hist(i.count)



## 3 - Clustering

Whilst this data is quite interesting, it would be better to see if there

Data Science Live Book - Pablo Casas - January 2019 This could also be done with a script and a few packages source: <https://blog.datascienceheroes.com/exploratory-data-analysis-in-r-intro/> accessed 05/08/2021

library(funModeling)

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following object is masked from 'package:Lahman':  
##   
## Label

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## funModeling v.1.9.4 :)  
## Examples and tutorials at livebook.datascienceheroes.com  
## / Now in Spanish: librovivodecienciadedatos.ai

library(tidyverse)   
library(Hmisc)  
# glimpse looks at data (similar to structure)  
glimpse(df.april20)

## Rows: 39,368  
## Columns: 9  
## $ Period <dbl> 202004, 202004, 202004, 202004, 202004, 202004, 202~  
## $ PracticeCode <chr> "W00005", "W00005", "W00005", "W00005", "W00005", "~  
## $ PostCode <chr> "CF83 4AZ", "CF83 4AZ", "CF83 4AZ", "CF83 4AZ", "CF~  
## $ OrgCode <chr> "7A6", "7A6", "7A6", "7A6", "7A6", "7A6", "7A6", "7~  
## $ AgeBand <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1~  
## $ MaleCount <dbl> 33, 42, 41, 43, 36, 51, 42, 50, 50, 54, 56, 58, 48,~  
## $ FemaleCount <dbl> 35, 47, 42, 56, 56, 48, 43, 42, 38, 38, 52, 40, 39,~  
## $ IndeterminateCount <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~  
## $ Count <dbl> 68, 89, 83, 99, 92, 99, 85, 92, 88, 92, 108, 98, 87~

# print to check the status of the dataframe  
print(status(df.april20))

## variable q\_zeros p\_zeros q\_na p\_na q\_inf p\_inf  
## Period Period 0 0.00000000 0 0 0 0  
## PracticeCode PracticeCode 0 0.00000000 0 0 0 0  
## PostCode PostCode 0 0.00000000 0 0 0 0  
## OrgCode OrgCode 0 0.00000000 0 0 0 0  
## AgeBand AgeBand 404 0.01026214 0 0 0 0  
## MaleCount MaleCount 753 0.01912721 0 0 0 0  
## FemaleCount FemaleCount 646 0.01640927 0 0 0 0  
## IndeterminateCount IndeterminateCount 39356 0.99969518 0 0 0 0  
## Count Count 502 0.01275147 0 0 0 0  
## type unique  
## Period numeric 1  
## PracticeCode character 414  
## PostCode character 389  
## OrgCode character 7  
## AgeBand numeric 96  
## MaleCount numeric 235  
## FemaleCount numeric 234  
## IndeterminateCount numeric 2  
## Count numeric 406

# frequency looks at plotting some of the numerical data  
freq(df.april20)

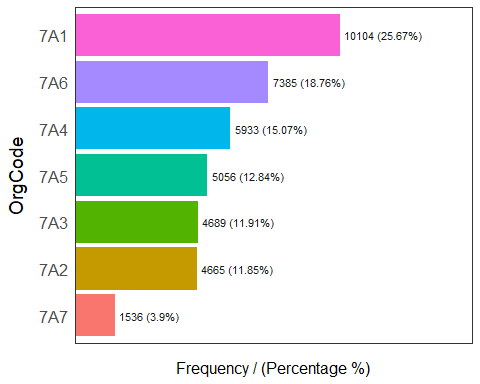
## Warning in freq\_logic(data = data, input = input[i], plot, na.rm, path\_out =  
## path\_out): Skipping plot for variable 'PracticeCode' (more than 100 categories)

## PracticeCode frequency percentage cumulative\_perc  
## 1 W00005 96 0.24 0.24  
## 2 W00007 96 0.24 0.48  
## 3 W00065 96 0.24 0.72  
## 4 W00067 96 0.24 0.96  
## 5 W00076 96 0.24 1.20  
## 6 W00133 96 0.24 1.44  
## 7 W00134 96 0.24 1.68  
## 8 W00141 96 0.24 1.92  
## 9 W00142 96 0.24 2.16  
## 10 W00143 96 0.24 2.40  
## 11 W00144 96 0.24 2.64  
## 12 W91002 96 0.24 2.88  
## 13 W91003 96 0.24 3.12  
## 14 W91004 96 0.24 3.36  
## 15 W91005 96 0.24 3.60  
## 16 W91006 96 0.24 3.84  
## 17 W91007 96 0.24 4.08  
## 18 W91009 96 0.24 4.32  
## 19 W91010 96 0.24 4.56  
## 20 W91013 96 0.24 4.80  
## 21 W91014 96 0.24 5.04  
## 22 W91015 96 0.24 5.28  
## 23 W91016 96 0.24 5.52  
## 24 W91017 96 0.24 5.76  
## 25 W91018 96 0.24 6.00  
## 26 W91019 96 0.24 6.24  
## 27 W91021 96 0.24 6.48  
## 28 W91022 96 0.24 6.72  
## 29 W91023 96 0.24 6.96  
## 30 W91024 96 0.24 7.20  
## 31 W91025 96 0.24 7.44  
## 32 W91026 96 0.24 7.68  
## 33 W91027 96 0.24 7.92  
## 34 W91028 96 0.24 8.16  
## 35 W91029 96 0.24 8.40  
## 36 W91030 96 0.24 8.64  
## 37 W91031 96 0.24 8.88  
## 38 W91032 96 0.24 9.12  
## 39 W91033 96 0.24 9.36  
## 40 W91034 96 0.24 9.60  
## 41 W91036 96 0.24 9.84  
## 42 W91037 96 0.24 10.08  
## 43 W91038 96 0.24 10.32  
## 44 W91039 96 0.24 10.56  
## 45 W91040 96 0.24 10.80  
## 46 W91042 96 0.24 11.04  
## 47 W91043 96 0.24 11.28  
## 48 W91044 96 0.24 11.52  
## 49 W91046 96 0.24 11.76  
## 50 W91047 96 0.24 12.00  
## 51 W91048 96 0.24 12.24  
## 52 W91051 96 0.24 12.48  
## 53 W91053 96 0.24 12.72  
## 54 W91054 96 0.24 12.96  
## 55 W91055 96 0.24 13.20  
## 56 W91056 96 0.24 13.44  
## 57 W91057 96 0.24 13.68  
## 58 W91058 96 0.24 13.92  
## 59 W91059 96 0.24 14.16  
## 60 W91060 96 0.24 14.40  
## 61 W91604 96 0.24 14.64  
## 62 W91610 96 0.24 14.88  
## 63 W91614 96 0.24 15.12  
## 64 W91615 96 0.24 15.36  
## 65 W91622 96 0.24 15.60  
## 66 W91624 96 0.24 15.84  
## 67 W91629 96 0.24 16.08  
## 68 W91636 96 0.24 16.32  
## 69 W92002 96 0.24 16.56  
## 70 W92003 96 0.24 16.80  
## 71 W92005 96 0.24 17.04  
## 72 W92006 96 0.24 17.28  
## 73 W92007 96 0.24 17.52  
## 74 W92008 96 0.24 17.76  
## 75 W92009 96 0.24 18.00  
## 76 W92013 96 0.24 18.24  
## 77 W92014 96 0.24 18.48  
## 78 W92015 96 0.24 18.72  
## 79 W92016 96 0.24 18.96  
## 80 W92018 96 0.24 19.20  
## 81 W92019 96 0.24 19.44  
## 82 W92021 96 0.24 19.68  
## 83 W92022 96 0.24 19.92  
## 84 W92023 96 0.24 20.16  
## 85 W92024 96 0.24 20.40  
## 86 W92025 96 0.24 20.64  
## 87 W92027 96 0.24 20.88  
## 88 W92031 96 0.24 21.12  
## 89 W92033 96 0.24 21.36  
## 90 W92035 96 0.24 21.60  
## 91 W92036 96 0.24 21.84  
## 92 W92037 96 0.24 22.08  
## 93 W92038 96 0.24 22.32  
## 94 W92039 96 0.24 22.56  
## 95 W92040 96 0.24 22.80  
## 96 W92041 96 0.24 23.04  
## 97 W92042 96 0.24 23.28  
## 98 W92044 96 0.24 23.52  
## 99 W92046 96 0.24 23.76  
## 100 W92048 96 0.24 24.00  
## 101 W92050 96 0.24 24.24  
## 102 W92051 96 0.24 24.48  
## 103 W92052 96 0.24 24.72  
## 104 W92053 96 0.24 24.96  
## 105 W92055 96 0.24 25.20  
## 106 W92056 96 0.24 25.44  
## 107 W92058 96 0.24 25.68  
## 108 W92059 96 0.24 25.92  
## 109 W92063 96 0.24 26.16  
## 110 W92064 96 0.24 26.40  
## 111 W92440 96 0.24 26.64  
## 112 W92616 96 0.24 26.88  
## 113 W93001 96 0.24 27.12  
## 114 W93004 96 0.24 27.36  
## 115 W93007 96 0.24 27.60  
## 116 W93009 96 0.24 27.84  
## 117 W93011 96 0.24 28.08  
## 118 W93012 96 0.24 28.32  
## 119 W93014 96 0.24 28.56  
## 120 W93015 96 0.24 28.80  
## 121 W93018 96 0.24 29.04  
## 122 W93019 96 0.24 29.28  
## 123 W93020 96 0.24 29.52  
## 124 W93021 96 0.24 29.76  
## 125 W93022 96 0.24 30.00  
## 126 W93023 96 0.24 30.24  
## 127 W93024 96 0.24 30.48  
## 128 W93025 96 0.24 30.72  
## 129 W93026 96 0.24 30.96  
## 130 W93027 96 0.24 31.20  
## 131 W93028 96 0.24 31.44  
## 132 W93029 96 0.24 31.68  
## 133 W93031 96 0.24 31.92  
## 134 W93032 96 0.24 32.16  
## 135 W93033 96 0.24 32.40  
## 136 W93035 96 0.24 32.64  
## 137 W93036 96 0.24 32.88  
## 138 W93037 96 0.24 33.12  
## 139 W93038 96 0.24 33.36  
## 140 W93039 96 0.24 33.60  
## 141 W93040 96 0.24 33.84  
## 142 W93043 96 0.24 34.08  
## 143 W93044 96 0.24 34.32  
## 144 W93045 96 0.24 34.56  
## 145 W93046 96 0.24 34.80  
## 146 W93047 96 0.24 35.04  
## 147 W93048 96 0.24 35.28  
## 148 W93049 96 0.24 35.52  
## 149 W93051 96 0.24 35.76  
## 150 W93053 96 0.24 36.00  
## 151 W93054 96 0.24 36.24  
## 152 W93055 96 0.24 36.48  
## 153 W93056 96 0.24 36.72  
## 154 W93057 96 0.24 36.96  
## 155 W93058 96 0.24 37.20  
## 156 W93059 96 0.24 37.44  
## 157 W93061 96 0.24 37.68  
## 158 W93063 96 0.24 37.92  
## 159 W93064 96 0.24 38.16  
## 160 W93065 96 0.24 38.40  
## 161 W93067 96 0.24 38.64  
## 162 W93068 96 0.24 38.88  
## 163 W93072 96 0.24 39.12  
## 164 W93075 96 0.24 39.36  
## 165 W93115 96 0.24 39.60  
## 166 W93116 96 0.24 39.84  
## 167 W93125 96 0.24 40.08  
## 168 W93126 96 0.24 40.32  
## 169 W93614 96 0.24 40.56  
## 170 W93619 96 0.24 40.80  
## 171 W93623 96 0.24 41.04  
## 172 W93640 96 0.24 41.28  
## 173 W93643 96 0.24 41.52  
## 174 W94001 96 0.24 41.76  
## 175 W94002 96 0.24 42.00  
## 176 W94003 96 0.24 42.24  
## 177 W94004 96 0.24 42.48  
## 178 W94005 96 0.24 42.72  
## 179 W94006 96 0.24 42.96  
## 180 W94007 96 0.24 43.20  
## 181 W94008 96 0.24 43.44  
## 182 W94009 96 0.24 43.68  
## 183 W94010 96 0.24 43.92  
## 184 W94011 96 0.24 44.16  
## 185 W94014 96 0.24 44.40  
## 186 W94015 96 0.24 44.64  
## 187 W94016 96 0.24 44.88  
## 188 W94017 96 0.24 45.12  
## 189 W94018 96 0.24 45.36  
## 190 W94019 96 0.24 45.60  
## 191 W94020 96 0.24 45.84  
## 192 W94021 96 0.24 46.08  
## 193 W94023 96 0.24 46.32  
## 194 W94024 96 0.24 46.56  
## 195 W94025 96 0.24 46.80  
## 196 W94026 96 0.24 47.04  
## 197 W94027 96 0.24 47.28  
## 198 W94028 96 0.24 47.52  
## 199 W94029 96 0.24 47.76  
## 200 W94030 96 0.24 48.00  
## 201 W94031 96 0.24 48.24  
## 202 W94032 96 0.24 48.48  
## 203 W94033 96 0.24 48.72  
## 204 W94034 96 0.24 48.96  
## 205 W94035 96 0.24 49.20  
## 206 W94036 96 0.24 49.44  
## 207 W94037 96 0.24 49.68  
## 208 W94038 96 0.24 49.92  
## 209 W94039 96 0.24 50.16  
## 210 W94040 96 0.24 50.40  
## 211 W94043 96 0.24 50.64  
## 212 W94046 96 0.24 50.88  
## 213 W94609 96 0.24 51.12  
## 214 W94612 96 0.24 51.36  
## 215 W94622 96 0.24 51.60  
## 216 W94633 96 0.24 51.84  
## 217 W95001 96 0.24 52.08  
## 218 W95004 96 0.24 52.32  
## 219 W95005 96 0.24 52.56  
## 220 W95008 96 0.24 52.80  
## 221 W95009 96 0.24 53.04  
## 222 W95010 96 0.24 53.28  
## 223 W95011 96 0.24 53.52  
## 224 W95012 96 0.24 53.76  
## 225 W95013 96 0.24 54.00  
## 226 W95014 96 0.24 54.24  
## 227 W95016 96 0.24 54.48  
## 228 W95017 96 0.24 54.72  
## 229 W95018 96 0.24 54.96  
## 230 W95019 96 0.24 55.20  
## 231 W95020 96 0.24 55.44  
## 232 W95023 96 0.24 55.68  
## 233 W95024 96 0.24 55.92  
## 234 W95025 96 0.24 56.16  
## 235 W95026 96 0.24 56.40  
## 236 W95027 96 0.24 56.64  
## 237 W95029 96 0.24 56.88  
## 238 W95030 96 0.24 57.12  
## 239 W95031 96 0.24 57.36  
## 240 W95032 96 0.24 57.60  
## 241 W95034 96 0.24 57.84  
## 242 W95035 96 0.24 58.08  
## 243 W95036 96 0.24 58.32  
## 244 W95037 96 0.24 58.56  
## 245 W95038 96 0.24 58.80  
## 246 W95041 96 0.24 59.04  
## 247 W95042 96 0.24 59.28  
## 248 W95043 96 0.24 59.52  
## 249 W95044 96 0.24 59.76  
## 250 W95046 96 0.24 60.00  
## 251 W95049 96 0.24 60.24  
## 252 W95050 96 0.24 60.48  
## 253 W95051 96 0.24 60.72  
## 254 W95054 96 0.24 60.96  
## 255 W95057 96 0.24 61.20  
## 256 W95058 96 0.24 61.44  
## 257 W95059 96 0.24 61.68  
## 258 W95060 96 0.24 61.92  
## 259 W95062 96 0.24 62.16  
## 260 W95063 96 0.24 62.40  
## 261 W95064 96 0.24 62.64  
## 262 W95065 96 0.24 62.88  
## 263 W95067 96 0.24 63.12  
## 264 W95068 96 0.24 63.36  
## 265 W95070 96 0.24 63.60  
## 266 W95071 96 0.24 63.84  
## 267 W95072 96 0.24 64.08  
## 268 W95073 96 0.24 64.32  
## 269 W95076 96 0.24 64.56  
## 270 W95078 96 0.24 64.80  
## 271 W95079 96 0.24 65.04  
## 272 W95081 96 0.24 65.28  
## 273 W95086 96 0.24 65.52  
## 274 W95087 96 0.24 65.76  
## 275 W95290 96 0.24 66.00  
## 276 W95295 96 0.24 66.24  
## 277 W95623 96 0.24 66.48  
## 278 W95633 96 0.24 66.72  
## 279 W95636 96 0.24 66.96  
## 280 W95645 96 0.24 67.20  
## 281 W95647 96 0.24 67.44  
## 282 W96001 96 0.24 67.68  
## 283 W96002 96 0.24 67.92  
## 284 W96003 96 0.24 68.16  
## 285 W96004 96 0.24 68.40  
## 286 W96005 96 0.24 68.64  
## 287 W96006 96 0.24 68.88  
## 288 W96007 96 0.24 69.12  
## 289 W96009 96 0.24 69.36  
## 290 W96010 96 0.24 69.60  
## 291 W96011 96 0.24 69.84  
## 292 W96012 96 0.24 70.08  
## 293 W96013 96 0.24 70.32  
## 294 W96015 96 0.24 70.56  
## 295 W96016 96 0.24 70.80  
## 296 W96017 96 0.24 71.04  
## 297 W96438 96 0.24 71.28  
## 298 W97001 96 0.24 71.52  
## 299 W97002 96 0.24 71.76  
## 300 W97003 96 0.24 72.00  
## 301 W97005 96 0.24 72.24  
## 302 W97006 96 0.24 72.48  
## 303 W97007 96 0.24 72.72  
## 304 W97008 96 0.24 72.96  
## 305 W97009 96 0.24 73.20  
## 306 W97010 96 0.24 73.44  
## 307 W97011 96 0.24 73.68  
## 308 W97013 96 0.24 73.92  
## 309 W97014 96 0.24 74.16  
## 310 W97015 96 0.24 74.40  
## 311 W97016 96 0.24 74.64  
## 312 W97017 96 0.24 74.88  
## 313 W97018 96 0.24 75.12  
## 314 W97020 96 0.24 75.36  
## 315 W97021 96 0.24 75.60  
## 316 W97023 96 0.24 75.84  
## 317 W97024 96 0.24 76.08  
## 318 W97025 96 0.24 76.32  
## 319 W97027 96 0.24 76.56  
## 320 W97028 96 0.24 76.80  
## 321 W97029 96 0.24 77.04  
## 322 W97031 96 0.24 77.28  
## 323 W97032 96 0.24 77.52  
## 324 W97033 96 0.24 77.76  
## 325 W97034 96 0.24 78.00  
## 326 W97036 96 0.24 78.24  
## 327 W97038 96 0.24 78.48  
## 328 W97040 96 0.24 78.72  
## 329 W97041 96 0.24 78.96  
## 330 W97044 96 0.24 79.20  
## 331 W97045 96 0.24 79.44  
## 332 W97046 96 0.24 79.68  
## 333 W97047 96 0.24 79.92  
## 334 W97048 96 0.24 80.16  
## 335 W97052 96 0.24 80.40  
## 336 W97053 96 0.24 80.64  
## 337 W97055 96 0.24 80.88  
## 338 W97056 96 0.24 81.12  
## 339 W97057 96 0.24 81.36  
## 340 W97058 96 0.24 81.60  
## 341 W97060 96 0.24 81.84  
## 342 W97061 96 0.24 82.08  
## 343 W97062 96 0.24 82.32  
## 344 W97063 96 0.24 82.56  
## 345 W97064 96 0.24 82.80  
## 346 W97065 96 0.24 83.04  
## 347 W97067 96 0.24 83.28  
## 348 W97068 96 0.24 83.52  
## 349 W97069 96 0.24 83.76  
## 350 W97286 96 0.24 84.00  
## 351 W97291 96 0.24 84.24  
## 352 W97294 96 0.24 84.48  
## 353 W97296 96 0.24 84.72  
## 354 W97299 96 0.24 84.96  
## 355 W97611 96 0.24 85.20  
## 356 W97616 96 0.24 85.44  
## 357 W97619 96 0.24 85.68  
## 358 W97623 96 0.24 85.92  
## 359 W98001 96 0.24 86.16  
## 360 W98002 96 0.24 86.40  
## 361 W98003 96 0.24 86.64  
## 362 W98004 96 0.24 86.88  
## 363 W98005 96 0.24 87.12  
## 364 W98006 96 0.24 87.36  
## 365 W98007 96 0.24 87.60  
## 366 W98008 96 0.24 87.84  
## 367 W98010 96 0.24 88.08  
## 368 W98012 96 0.24 88.32  
## 369 W98013 96 0.24 88.56  
## 370 W98016 96 0.24 88.80  
## 371 W98017 96 0.24 89.04  
## 372 W98019 96 0.24 89.28  
## 373 W98020 96 0.24 89.52  
## 374 W98021 96 0.24 89.76  
## 375 W98022 96 0.24 90.00  
## 376 W98023 96 0.24 90.24  
## 377 W98024 96 0.24 90.48  
## 378 W98027 96 0.24 90.72  
## 379 W98028 96 0.24 90.96  
## 380 W98030 96 0.24 91.20  
## 381 W98031 96 0.24 91.44  
## 382 W98032 96 0.24 91.68  
## 383 W98033 96 0.24 91.92  
## 384 W98034 96 0.24 92.16  
## 385 W98035 96 0.24 92.40  
## 386 W98036 96 0.24 92.64  
## 387 W98039 96 0.24 92.88  
## 388 W98040 96 0.24 93.12  
## 389 W98041 96 0.24 93.36  
## 390 W98043 96 0.24 93.60  
## 391 W98044 96 0.24 93.84  
## 392 W98045 96 0.24 94.08  
## 393 W98046 96 0.24 94.32  
## 394 W98048 96 0.24 94.56  
## 395 W98049 96 0.24 94.80  
## 396 W98055 96 0.24 95.04  
## 397 W98056 96 0.24 95.28  
## 398 W98057 96 0.24 95.52  
## 399 W98608 96 0.24 95.76  
## 400 W98612 96 0.24 96.00  
## 401 W98627 96 0.24 96.24  
## 402 W98785 96 0.24 96.48  
## 403 W98015 95 0.24 96.72  
## 404 W98053 82 0.21 96.93  
## 405 W00100 77 0.20 97.13  
## 406 W00071 69 0.18 97.31  
## 407 W00165 65 0.17 97.48  
## 408 W00102 64 0.16 97.64  
## 409 W93131 61 0.15 97.79  
## 410 W00166 59 0.15 97.94  
## 411 W00075 58 0.15 98.09  
## 412 W00149 57 0.14 98.23  
## 413 W00073 55 0.14 98.37  
## 414 W00072 34 0.09 100.00

## Warning in freq\_logic(data = data, input = input[i], plot, na.rm, path\_out =  
## path\_out): Skipping plot for variable 'PostCode' (more than 100 categories)

## PostCode frequency percentage cumulative\_perc  
## 1 SA12 7BJ 384 0.98 0.98  
## 2 CF48 1BZ 288 0.73 1.71  
## 3 CF11 7DJ 192 0.49 2.20  
## 4 CF11 9SH 192 0.49 2.69  
## 5 CF5 5LQ 192 0.49 3.18  
## 6 CF71 7DA 192 0.49 3.67  
## 7 CH5 4PJ 192 0.49 4.16  
## 8 CH6 5ER 192 0.49 4.65  
## 9 CH7 3PG 192 0.49 5.14  
## 10 LL11 3NS 192 0.49 5.63  
## 11 LL19 9LN 192 0.49 6.12  
## 12 NP12 2YU 192 0.49 6.61  
## 13 NP13 1BQ 192 0.49 7.10  
## 14 NP15 1AB 192 0.49 7.59  
## 15 NP22 5PW 192 0.49 8.08  
## 16 NP4 6DH 192 0.49 8.57  
## 17 SA1 8QY 192 0.49 9.06  
## 18 SA11 2FP 192 0.49 9.55  
## 19 SA12 9PY 192 0.49 10.04  
## 20 SA5 5AA 192 0.49 10.53  
## 21 SA73 2JW 192 0.49 11.02  
## 22 CF39 0LD 160 0.41 11.43  
## 23 CF10 5HW 96 0.24 11.67  
## 24 CF11 6PG 96 0.24 11.91  
## 25 CF11 6QQ 96 0.24 12.15  
## 26 CF11 6RW 96 0.24 12.39  
## 27 CF11 7AT 96 0.24 12.63  
## 28 CF11 8DG 96 0.24 12.87  
## 29 CF11 9DG 96 0.24 13.11  
## 30 CF11 9DN 96 0.24 13.35  
## 31 CF11 9EE 96 0.24 13.59  
## 32 CF14 1LT 96 0.24 13.83  
## 33 CF14 2FD 96 0.24 14.07  
## 34 CF14 3NB 96 0.24 14.31  
## 35 CF14 3QX 96 0.24 14.55  
## 36 CF14 3XQ 96 0.24 14.79  
## 37 CF14 4QJ 96 0.24 15.03  
## 38 CF14 4UU 96 0.24 15.27  
## 39 CF14 5YU 96 0.24 15.51  
## 40 CF14 7EZ 96 0.24 15.75  
## 41 CF14 9BB 96 0.24 15.99  
## 42 CF15 7YG 96 0.24 16.23  
## 43 CF15 8DZ 96 0.24 16.47  
## 44 CF23 5RH 96 0.24 16.71  
## 45 CF23 5SY 96 0.24 16.95  
## 46 CF23 7SD 96 0.24 17.19  
## 47 CF23 8SQ 96 0.24 17.43  
## 48 CF23 9PN 96 0.24 17.67  
## 49 CF24 1AG 96 0.24 17.91  
## 50 CF24 1YT 96 0.24 18.15  
## 51 CF24 2HB 96 0.24 18.39  
## 52 CF24 2LU 96 0.24 18.63  
## 53 CF24 3JD 96 0.24 18.87  
## 54 CF24 3WD 96 0.24 19.11  
## 55 CF24 4HU 96 0.24 19.35  
## 56 CF3 0EF 96 0.24 19.59  
## 57 CF3 0SH 96 0.24 19.83  
## 58 CF3 3LG 96 0.24 20.07  
## 59 CF31 2PQ 96 0.24 20.31  
## 60 CF31 3NL 96 0.24 20.55  
## 61 CF31 3NW 96 0.24 20.79  
## 62 CF32 7BL 96 0.24 21.03  
## 63 CF32 7NA 96 0.24 21.27  
## 64 CF32 8NN 96 0.24 21.51  
## 65 CF32 9BL 96 0.24 21.75  
## 66 CF32 9SW 96 0.24 21.99  
## 67 CF33 4LD 96 0.24 22.23  
## 68 CF33 6BY 96 0.24 22.47  
## 69 CF34 0SR 96 0.24 22.71  
## 70 CF34 9DT 96 0.24 22.95  
## 71 CF34 9PW 96 0.24 23.19  
## 72 CF35 5PF 96 0.24 23.43  
## 73 CF35 6YP 96 0.24 23.67  
## 74 CF36 5DJ 96 0.24 23.91  
## 75 CF37 2AA 96 0.24 24.15  
## 76 CF37 2DR 96 0.24 24.39  
## 77 CF37 5RW 96 0.24 24.63  
## 78 CF38 1RJ 96 0.24 24.87  
## 79 CF39 8AG 96 0.24 25.11  
## 80 CF39 8SX 96 0.24 25.35  
## 81 CF40 1QN 96 0.24 25.59  
## 82 CF40 2LE 96 0.24 25.83  
## 83 CF40 2QZ 96 0.24 26.07  
## 84 CF40 2SX 96 0.24 26.31  
## 85 CF41 7BD 96 0.24 26.55  
## 86 CF42 5LW 96 0.24 26.79  
## 87 CF42 6DL 96 0.24 27.03  
## 88 CF43 3HB 96 0.24 27.27  
## 89 CF43 4XX 96 0.24 27.51  
## 90 CF44 6HY 96 0.24 27.75  
## 91 CF44 7AY 96 0.24 27.99  
## 92 CF44 7DD 96 0.24 28.23  
## 93 CF44 7PA 96 0.24 28.47  
## 94 CF44 9SL 96 0.24 28.71  
## 95 CF45 3HD 96 0.24 28.95  
## 96 CF45 3SQ 96 0.24 29.19  
## 97 CF45 4EY 96 0.24 29.43  
## 98 CF45 4YB 96 0.24 29.67  
## 99 CF46 5HE 96 0.24 29.91  
## 100 CF46 6HL 96 0.24 30.15  
## 101 CF46 6TE 96 0.24 30.39  
## 102 CF48 1YE 96 0.24 30.63  
## 103 CF48 3AL 96 0.24 30.87  
## 104 CF48 4QU 96 0.24 31.11  
## 105 CF5 2DY 96 0.24 31.35  
## 106 CF5 2SH 96 0.24 31.59  
## 107 CF5 3JT 96 0.24 31.83  
## 108 CF5 4AD 96 0.24 32.07  
## 109 CF5 4LJ 96 0.24 32.31  
## 110 CF5 4RG 96 0.24 32.55  
## 111 CF61 1ST 96 0.24 32.79  
## 112 CF62 5QN 96 0.24 33.03  
## 113 CF62 8AZ 96 0.24 33.27  
## 114 CF62 8GP 96 0.24 33.51  
## 115 CF63 1BA 96 0.24 33.75  
## 116 CF63 4AR 96 0.24 33.99  
## 117 CF63 4YD 96 0.24 34.23  
## 118 CF64 1BX 96 0.24 34.47  
## 119 CF64 3WX 96 0.24 34.71  
## 120 CF64 3XE 96 0.24 34.95  
## 121 CF64 4RE 96 0.24 35.19  
## 122 CF64 5TG 96 0.24 35.43  
## 123 CF72 8AJ 96 0.24 35.67  
## 124 CF72 9AA 96 0.24 35.91  
## 125 CF81 8SA 96 0.24 36.15  
## 126 CF81 8ST 96 0.24 36.39  
## 127 CF82 7WX 96 0.24 36.63  
## 128 CF82 8FQ 96 0.24 36.87  
## 129 CF83 1RB 96 0.24 37.11  
## 130 CF83 1XP 96 0.24 37.35  
## 131 CF83 2AX 96 0.24 37.59  
## 132 CF83 3GH 96 0.24 37.83  
## 133 CF83 3JZ 96 0.24 38.07  
## 134 CF83 4AZ 96 0.24 38.31  
## 135 CF83 8GL 96 0.24 38.55  
## 136 CH4 0NR 96 0.24 38.79  
## 137 CH5 1QA 96 0.24 39.03  
## 138 CH5 1QT 96 0.24 39.27  
## 139 CH5 1SY 96 0.24 39.51  
## 140 CH5 3PA 96 0.24 39.75  
## 141 CH6 5UZ 96 0.24 39.99  
## 142 CH7 1SS 96 0.24 40.23  
## 143 CH7 4RQ 96 0.24 40.47  
## 144 CH8 7GA 96 0.24 40.71  
## 145 CH8 7RS 96 0.24 40.95  
## 146 CH8 7TR 96 0.24 41.19  
## 147 CH8 7TZ 96 0.24 41.43  
## 148 GL15 6TN 96 0.24 41.67  
## 149 LD1 5ES 96 0.24 41.91  
## 150 LD2 3DZ 96 0.24 42.15  
## 151 LD3 8AH 96 0.24 42.39  
## 152 LD3 OAW 96 0.24 42.63  
## 153 LD6 5ED 96 0.24 42.87  
## 154 LD7 1AD 96 0.24 43.11  
## 155 LD8 2RJ 96 0.24 43.35  
## 156 LL11 2SA 96 0.24 43.59  
## 157 LL11 3SA 96 0.24 43.83  
## 158 LL11 4UF 96 0.24 44.07  
## 159 LL12 0TR 96 0.24 44.31  
## 160 LL12 7TH 96 0.24 44.55  
## 161 LL12 9LG 96 0.24 44.79  
## 162 LL12 9NL 96 0.24 45.03  
## 163 LL13 0ED 96 0.24 45.27  
## 164 LL13 7BS 96 0.24 45.51  
## 165 LL13 7DE 96 0.24 45.75  
## 166 LL13 8DB 96 0.24 45.99  
## 167 LL13 8RG 96 0.24 46.23  
## 168 LL13 8TH 96 0.24 46.47  
## 169 LL14 1AA 96 0.24 46.71  
## 170 LL14 2EN 96 0.24 46.95  
## 171 LL14 3AB 96 0.24 47.19  
## 172 LL14 5DH 96 0.24 47.43  
## 173 LL14 6NH 96 0.24 47.67  
## 174 LL15 1BG 96 0.24 47.91  
## 175 LL15 1BP 96 0.24 48.15  
## 176 LL16 3AU 96 0.24 48.39  
## 177 LL16 3BL 96 0.24 48.63  
## 178 LL16 3TH 96 0.24 48.87  
## 179 LL16 3UW 96 0.24 49.11  
## 180 LL17 0LU 96 0.24 49.35  
## 181 LL18 1DA 96 0.24 49.59  
## 182 LL18 1LR 96 0.24 49.83  
## 183 LL18 1LT 96 0.24 50.07  
## 184 LL18 4RS 96 0.24 50.31  
## 185 LL18 5AU 96 0.24 50.55  
## 186 LL20 8RZ 96 0.24 50.79  
## 187 LL21 0DN 96 0.24 51.03  
## 188 LL21 9UB 96 0.24 51.27  
## 189 LL22 8LJ 96 0.24 51.51  
## 190 LL23 7BA 96 0.24 51.75  
## 191 LL24 0BP 96 0.24 51.99  
## 192 LL26 0AR 96 0.24 52.23  
## 193 LL29 7DA 96 0.24 52.47  
## 194 LL29 7LS 96 0.24 52.71  
## 195 LL29 9NP 96 0.24 52.95  
## 196 LL30 1TA 96 0.24 53.19  
## 197 LL30 1YL 96 0.24 53.43  
## 198 LL30 2BL 96 0.24 53.67  
## 199 LL30 3EU 96 0.24 53.91  
## 200 LL31 9NS 96 0.24 54.15  
## 201 LL32 8AT 96 0.24 54.39  
## 202 LL32 8AY 96 0.24 54.63  
## 203 LL32 8LT 96 0.24 54.87  
## 204 LL33 0PE 96 0.24 55.11  
## 205 LL36 9HL 96 0.24 55.35  
## 206 LL40 1LY 96 0.24 55.59  
## 207 LL41 3DW 96 0.24 55.83  
## 208 LL42 1PL 96 0.24 56.07  
## 209 LL48 6AL 96 0.24 56.31  
## 210 LL49 9NU 96 0.24 56.55  
## 211 LL52 0RR 96 0.24 56.79  
## 212 LL53 5NF 96 0.24 57.03  
## 213 LL53 6EG 96 0.24 57.27  
## 214 LL53 8RE 96 0.24 57.51  
## 215 LL54 6HD 96 0.24 57.75  
## 216 LL54 6NN 96 0.24 57.99  
## 217 LL55 1TH 96 0.24 58.23  
## 218 LL55 4SU 96 0.24 58.47  
## 219 LL55 4YY 96 0.24 58.71  
## 220 LL56 4RX 96 0.24 58.95  
## 221 LL57 1AH 96 0.24 59.19  
## 222 LL57 1AY 96 0.24 59.43  
## 223 LL57 2HH 96 0.24 59.67  
## 224 LL57 3NE 96 0.24 59.91  
## 225 LL58 8AL 96 0.24 60.15  
## 226 LL60 6AH 96 0.24 60.39  
## 227 LL61 5YZ 96 0.24 60.63  
## 228 LL62 5NL 96 0.24 60.87  
## 229 LL65 1RA 96 0.24 61.11  
## 230 LL65 1TR 96 0.24 61.35  
## 231 LL65 1UD 96 0.24 61.59  
## 232 LL65 4RS 96 0.24 61.83  
## 233 LL68 9AB 96 0.24 62.07  
## 234 LL74 8TF 96 0.24 62.31  
## 235 LL77 7DU 96 0.24 62.55  
## 236 NP10 8UX 96 0.24 62.79  
## 237 NP10 9DU 96 0.24 63.03  
## 238 NP11 4PQ 96 0.24 63.27  
## 239 NP11 5GX 96 0.24 63.51  
## 240 NP11 6BJ 96 0.24 63.75  
## 241 NP11 6YS 96 0.24 63.99  
## 242 NP12 3NA 96 0.24 64.23  
## 243 NP12 3WA 96 0.24 64.47  
## 244 NP13 2AB 96 0.24 64.71  
## 245 NP13 3AT 96 0.24 64.95  
## 246 NP16 5PZ 96 0.24 65.19  
## 247 NP16 5XP 96 0.24 65.43  
## 248 NP16 5XR 96 0.24 65.67  
## 249 NP16 6SE 96 0.24 65.91  
## 250 NP18 1AZ 96 0.24 66.15  
## 251 NP18 2JB 96 0.24 66.39  
## 252 NP19 0DW 96 0.24 66.63  
## 253 NP19 4TD 96 0.24 66.87  
## 254 NP19 7DQ 96 0.24 67.11  
## 255 NP19 7FY 96 0.24 67.35  
## 256 NP19 8HL 96 0.24 67.59  
## 257 NP19 8XR 96 0.24 67.83  
## 258 NP19 9PS 96 0.24 68.07  
## 259 NP20 2LB 96 0.24 68.31  
## 260 NP20 2WQ 96 0.24 68.55  
## 261 NP20 4EJ 96 0.24 68.79  
## 262 NP20 4JS 96 0.24 69.03  
## 263 NP20 5PJ 96 0.24 69.27  
## 264 NP20 6EY 96 0.24 69.51  
## 265 NP22 3XP 96 0.24 69.75  
## 266 NP22 4LB 96 0.24 69.99  
## 267 NP23 4BR 96 0.24 70.23  
## 268 NP23 5NT 96 0.24 70.47  
## 269 NP23 6EY 96 0.24 70.71  
## 270 NP23 6JG 96 0.24 70.95  
## 271 NP23 7RW 96 0.24 71.19  
## 272 NP25 3EQ 96 0.24 71.43  
## 273 NP25 3PL 96 0.24 71.67  
## 274 NP26 5AB 96 0.24 71.91  
## 275 NP4 5DJ 96 0.24 72.15  
## 276 NP4 7BH 96 0.24 72.39  
## 277 NP4 8AT 96 0.24 72.63  
## 278 NP4 9AW 96 0.24 72.87  
## 279 NP44 1DU 96 0.24 73.11  
## 280 NP44 1RY 96 0.24 73.35  
## 281 NP44 3JS 96 0.24 73.59  
## 282 NP44 3LT 96 0.24 73.83  
## 283 NP44 4TA 96 0.24 74.07  
## 284 NP44 8HW 96 0.24 74.31  
## 285 NP7 5DL 96 0.24 74.55  
## 286 NP7 5PR 96 0.24 74.79  
## 287 NP7 5UH 96 0.24 75.03  
## 288 NP8 1AG 96 0.24 75.27  
## 289 SA1 1HW 96 0.24 75.51  
## 290 SA1 4DE 96 0.24 75.75  
## 291 SA1 4DF 96 0.24 75.99  
## 292 SA1 4HF 96 0.24 76.23  
## 293 SA1 5LF 96 0.24 76.47  
## 294 SA1 8LH 96 0.24 76.71  
## 295 SA10 6UF 96 0.24 76.95  
## 296 SA10 6UH 96 0.24 77.19  
## 297 SA10 9EY 96 0.24 77.43  
## 298 SA11 1EF 96 0.24 77.67  
## 299 SA11 1HW 96 0.24 77.91  
## 300 SA11 3AW 96 0.24 78.15  
## 301 SA11 3EW 96 0.24 78.39  
## 302 SA11 5AL 96 0.24 78.63  
## 303 SA13 2BN 96 0.24 78.87  
## 304 SA13 3DP 96 0.24 79.11  
## 305 SA14 6DP 96 0.24 79.35  
## 306 SA14 7RP 96 0.24 79.59  
## 307 SA14 8TU 96 0.24 79.83  
## 308 SA14 9BN 96 0.24 80.07  
## 309 SA15 2TJ 96 0.24 80.31  
## 310 SA15 3AE 96 0.24 80.55  
## 311 SA15 3BD 96 0.24 80.79  
## 312 SA15 3JH 96 0.24 81.03  
## 313 SA15 5HU 96 0.24 81.27  
## 314 SA15 5TR 96 0.24 81.51  
## 315 SA16 0BN 96 0.24 81.75  
## 316 SA17 4UL 96 0.24 81.99  
## 317 SA18 1EG 96 0.24 82.23  
## 318 SA18 2DA 96 0.24 82.47  
## 319 SA18 2PJ 96 0.24 82.71  
## 320 SA19 6HN 96 0.24 82.95  
## 321 SA2 0GU 96 0.24 83.19  
## 322 SA2 0LJ 96 0.24 83.43  
## 323 SA2 9EA 96 0.24 83.67  
## 324 SA20 0HY 96 0.24 83.91  
## 325 SA3 1AY 96 0.24 84.15  
## 326 SA3 4AJ 96 0.24 84.39  
## 327 SA3 5UA 96 0.24 84.63  
## 328 SA31 1AH 96 0.24 84.87  
## 329 SA31 1EX 96 0.24 85.11  
## 330 SA31 3AX 96 0.24 85.35  
## 331 SA32 7LG 96 0.24 85.59  
## 332 SA33 4AA 96 0.24 85.83  
## 333 SA34 0AJ 96 0.24 86.07  
## 334 SA38 9NS 96 0.24 86.31  
## 335 SA4 3ED 96 0.24 86.55  
## 336 SA4 4BY 96 0.24 86.79  
## 337 SA4 4US 96 0.24 87.03  
## 338 SA4 8TJ 96 0.24 87.27  
## 339 SA42 0TJ 96 0.24 87.51  
## 340 SA43 1JX 96 0.24 87.75  
## 341 SA44 4JX 96 0.24 87.99  
## 342 SA45 9PB 96 0.24 88.23  
## 343 SA46 0DY 96 0.24 88.47  
## 344 SA48 7AA 96 0.24 88.71  
## 345 SA5 5LB 96 0.24 88.95  
## 346 SA5 8QE 96 0.24 89.19  
## 347 SA5 9EA 96 0.24 89.43  
## 348 SA6 5LN 96 0.24 89.67  
## 349 SA6 7AG 96 0.24 89.91  
## 350 SA6 7NZ 96 0.24 90.15  
## 351 SA61 1QX 96 0.24 90.39  
## 352 SA61 1RN 96 0.24 90.63  
## 353 SA62 6SS 96 0.24 90.87  
## 354 SA62 6TW 96 0.24 91.11  
## 355 SA65 9BT 96 0.24 91.35  
## 356 SA67 7AA 96 0.24 91.59  
## 357 SA69 9JW 96 0.24 91.83  
## 358 SA7 9RY 96 0.24 92.07  
## 359 SA70 8AB 96 0.24 92.31  
## 360 SA72 6HL 96 0.24 92.55  
## 361 SA73 1SH 96 0.24 92.79  
## 362 SA8 4JU 96 0.24 93.03  
## 363 SA9 1DS 96 0.24 93.27  
## 364 SA9 2GH 96 0.24 93.51  
## 365 SY13 3DL 96 0.24 93.75  
## 366 SY15 6PF 96 0.24 93.99  
## 367 SY16 1EF 96 0.24 94.23  
## 368 SY18 6EZ 96 0.24 94.47  
## 369 SY20 8EQ 96 0.24 94.71  
## 370 SY21 0RT 96 0.24 94.95  
## 371 SY21 7ER 96 0.24 95.19  
## 372 SY22 5DG 96 0.24 95.43  
## 373 SY23 2DX 96 0.24 95.67  
## 374 SY23 3DU 96 0.24 95.91  
## 375 SY23 3TL 96 0.24 96.15  
## 376 SY23 4PA 96 0.24 96.39  
## 377 SY24 5JE 96 0.24 96.63  
## 378 SY25 6HA 96 0.24 96.87  
## 379 SA1 6TD 95 0.24 97.11  
## 380 SA2 8PG 82 0.21 97.32  
## 381 CF24 0SZ 77 0.20 97.52  
## 382 CH7 1PZ 69 0.18 97.70  
## 383 CF83 1LX 65 0.17 97.87  
## 384 NP20 4SZ 61 0.15 98.02  
## 385 NP4 0AH 59 0.15 98.17  
## 386 LL55 1HU 58 0.15 98.32  
## 387 SA31 3BB 57 0.14 98.46  
## 388 LL13 7YP 55 0.14 98.60  
## 389 LL17 0RS 34 0.09 100.00

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.

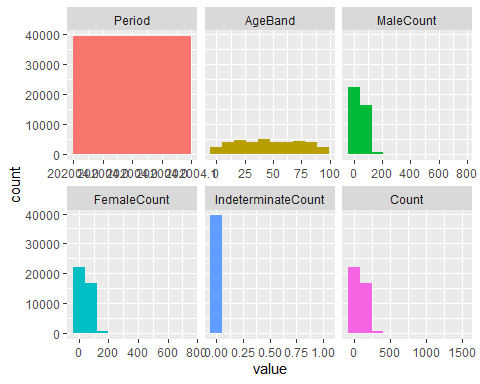


## OrgCode frequency percentage cumulative\_perc  
## 1 7A1 10104 25.67 25.67  
## 2 7A6 7385 18.76 44.43  
## 3 7A4 5933 15.07 59.50  
## 4 7A5 5056 12.84 72.34  
## 5 7A3 4689 11.91 84.25  
## 6 7A2 4665 11.85 96.10  
## 7 7A7 1536 3.90 100.00

## [1] "Variables processed: PracticeCode, PostCode, OrgCode"

# numerical profiling in one function   
# automatically excludes non-numerical variables  
plot\_num(df.april20)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



# describe looks at concise statistical descriptions of the df.  
describe(df.april20)

## df.april20   
##   
## 9 Variables 39368 Observations  
## --------------------------------------------------------------------------------  
## Period   
## n missing distinct Info Mean Gmd   
## 39368 0 1 0 202004 0   
##   
## Value 202004  
## Frequency 39368  
## Proportion 1  
## --------------------------------------------------------------------------------  
## PracticeCode   
## n missing distinct   
## 39368 0 414   
##   
## lowest : W00005 W00007 W00065 W00067 W00071, highest: W98057 W98608 W98612 W98627 W98785  
## --------------------------------------------------------------------------------  
## PostCode   
## n missing distinct   
## 39368 0 389   
##   
## lowest : CF10 5HW CF11 6PG CF11 6QQ CF11 6RW CF11 7AT  
## highest: SY23 3DU SY23 3TL SY23 4PA SY24 5JE SY25 6HA  
## --------------------------------------------------------------------------------  
## OrgCode   
## n missing distinct   
## 39368 0 7   
##   
## lowest : 7A1 7A2 7A3 7A4 7A5, highest: 7A3 7A4 7A5 7A6 7A7  
##   
## Value 7A1 7A2 7A3 7A4 7A5 7A6 7A7  
## Frequency 10104 4665 4689 5933 5056 7385 1536  
## Proportion 0.257 0.118 0.119 0.151 0.128 0.188 0.039  
## --------------------------------------------------------------------------------  
## AgeBand   
## n missing distinct Info Mean Gmd .05 .10   
## 39368 0 96 1 46.75 31.44 4.00 9.00   
## .25 .50 .75 .90 .95   
## 23.00 46.75 70.00 85.00 90.00   
##   
## lowest : 0 1 2 3 4, highest: 90 91 92 93 94  
## --------------------------------------------------------------------------------  
## MaleCount   
## n missing distinct Info Mean Gmd .05 .10   
## 39368 0 235 1 41.09 31.94 3 7   
## .25 .50 .75 .90 .95   
## 19 37 56 79 95   
##   
## lowest : 0 1 2 3 4, highest: 445 514 584 644 753  
## --------------------------------------------------------------------------------  
## FemaleCount   
## n missing distinct Info Mean Gmd .05 .10   
## 39368 0 234 1 41.27 31.17 5 9   
## .25 .50 .75 .90 .95   
## 20 36 56 78 94   
##   
## lowest : 0 1 2 3 4, highest: 502 532 537 559 720  
## --------------------------------------------------------------------------------  
## IndeterminateCount   
## n missing distinct Info Sum Mean Gmd   
## 39368 0 2 0.001 12 0.0003048 0.0006095   
##   
## --------------------------------------------------------------------------------  
## Count   
## n missing distinct Info Mean Gmd .05 .10   
## 39368 0 406 1 82.36 62.14 9 18   
## .25 .50 .75 .90 .95   
## 40 74 111 156 187   
##   
## lowest : 0 1 2 3 4, highest: 947 967 1181 1304 1473  
## --------------------------------------------------------------------------------

# 2 - Descriptive Statistics ####  
mean(df.april20$MaleCount)

## [1] 41.08555

# The mean MaleCount is 41.08  
mean(df.april20$FemaleCount)

## [1] 41.27474

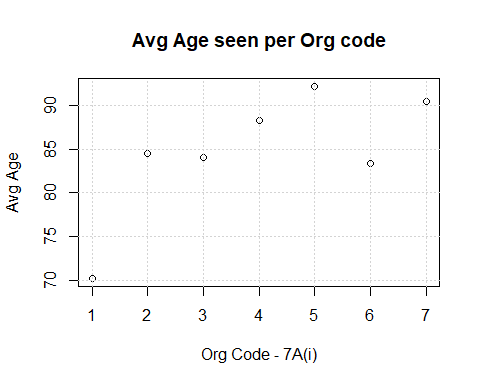
# The mean FemaleCount is 41.27  
# This would suggest more Females get seen than men.  
  
max(df.april20$MaleCount)

## [1] 753

# The mean MaleCount is 41.08  
max(df.april20$FemaleCount)

## [1] 720

# Could we look at the different types with AgeGroups?  
  
# profiling num is great for descriptive statistics  
descStats <- profiling\_num(df.april20)  
write.csv(descStats, file = "descriptive-stats.csv")  
  
# Can we look at who saw the most people on avg,  
# via org code?  
orgPerP <- aggregate(formula = Count ~ OrgCode,  
 data = df.april20,  
 FUN = mean)  
# 3 - Plots ####  
plot(x = orgPerP$Count,  
 main = "Avg Age seen per Org code",  
 xlab = "Org Code - 7A(i)",  
 ylab = "Avg Age")  
grid()



aggregate(formula = Count ~ OrgCode,  
 data = df.april20,  
 FUN = max)

## OrgCode Count  
## 1 7A1 829  
## 2 7A2 967  
## 3 7A3 1473  
## 4 7A4 819  
## 5 7A5 504  
## 6 7A6 340  
## 7 7A7 265

# 4 - Hypothesis Testing ####  
# Not sure what to do here?  
  
  
# Can we look at demand over time  
# Look at plotting the Genders / Age (gender per area)  
# Link LSOA to existing Data sets  
# Can look at Most Popular Visit per Age?  
# what proportion of females went out of the whole pop  
# Which would be count/sum(malecount) + sum(femalecount)  
# Or just for that GP actually  
# Which would be count/malecount + femalecount  
  
# Create an empty df.