# Renfrewshire Council Social Care Data Analysis

David Henderson

2017-12-11

## 1 Preliminaries

### 1.1 Packages

```
library(dplyr)
library(descriptr)
library(janitor)
library(forcats)
library(lubridate)
library(purrr)
library(tidyr)
library(tibbletime)
library(ggplot2)
library(cowplot)
library(magrittr)
library(knitr)
library(kableExtra)
library(extrafont)
theme_set(theme_cowplot(font_size = 12, font_family = "Arial Narrow"))
ubdc_palette <- c("#13AFD6", "#E6E600", "#F07329", "#35B14E", "#D7509A", "#2165AF",
                  "#BCD032", "#866BAC", "#7A8082", "#545A5D", "#E2D988", "#628DB7",
                  "#929B9A", "#93B8DA", "#31649B", "#FBF8D0", "#ACB2B4", "#D1DAE2")
```

### 1.2 Options

Can’t set .RProfile in safe haven so set some global options here

```
options(digits = 2,                #Set number of decimal places to 2
        scipen = 10,               #Do not use scientific convention except for very large numbers
        continue = "... ",         #nicer console continuation icon (instead of "+")
        width = 90,
        papersize = "letter",
        formatR.indent = 2,
        knitr.table.format = "latex")
```

## 2 Introduction

The research question this analysis aims to answer is

> “How does social care use vary within and across each financial year for which data is provided?”

The analysis aims to provide some validation for the way hours of home care are reported by the social care survey (SCS) collected by the Scottish Government.[1] This information is collected annually at the end of March each year. What I want to know is if the figure reported in the SCS is indicative of the hours of care an individual receives throughout a financial year.

Data has been provided by Renfrewshire Council on all social care delivered in the years 2006-2016. I will be analysing this data to answer the research question.

# 3 Load and clean `id_gender_age`

Load and clean the raw data files `id_gender_age_raw`

## 3.1 Load raw data

```
load("Research/raw_data/id_gender_age_raw.rds")
```

## 3.2 Clean and check variables

I'll start with `id_gender_age`

```
glimpse(id_gender_age_raw)
```

The output here shows there are 16,286 observations of 3 variables; "Client ID", "Year of Birth", and "Gender".

Jobs to do:-

- Create "clean" object and then tidy it by:-
- Tidying the variable names
- Coerceing variables to correct format
- Checking variable quality i.e. look for missing values and outliers

## 3.3 Tidy variable names

```
id_gender_age <- id_gender_age_raw #creates a duplicate that I will edit

rm(id_gender_age_raw) #takes raw data out of memory

names(id_gender_age) <- c("id", "yob", "gender")
```

## 3.4 Coerce variables to correct format

```
# id is an integer - I'll coerce to character
id_gender_age$id <- as.character(id_gender_age$id)
```

[1] A further question regarding the distribution of social care according to socioeconomic position will be completed at a later date

```
# yob is an integer - needs to be a Date object
id_gender_age$yob <- ymd(sprintf("%d-01-01",id_gender_age$yob))
#Set year of birth to 1st january of that year

#gender is a character - factorise
id_gender_age$gender <- factor(id_gender_age$gender,
                               levels = c("Female", "Male"),
                               labels = c("Female", "Male"))
```

## 3.5 Check variable quality

Any duplicates?

```
nrow(id_gender_age) - nrow(distinct(id_gender_age))
```

```
## [1] 0
```

No, good

## 3.6 overall stats

Quick summary using `descriptr` package

```
screener(id_gender_age)
```

```
## -----------------------------------------------------------------------
## |  Column Name  |  Data Type  |  Levels   |  Missing  |  Missing (%)  |
## -----------------------------------------------------------------------
## |      id       |  character  |    NA     |     0     |       0       |
## |      yob      |    Date     |    NA     |     0     |       0       |
## |    gender     |   factor    |Female Male|    13     |     0.08      |
## -----------------------------------------------------------------------
##
##  Overall Missing Values           13
##  Percentage of Missing Values     0.03 %
##  Rows with Missing Values         13
##  Columns With Missing Values      1
```

Ok, so hardly any missing data - good.
I'll look at each of the variables in more detail

## 3.7 id

1st thing to check here is that there are no duplicates in this variable

```
anyDuplicated(id_gender_age$id)
```

```
## [1] 0
```

Good. So we have unique id numbers for 16286 individuals

### 3.8 yob

Quick look at the structure here

```
id_gender_age %>%
  ggplot(aes(year(yob))) +
  geom_histogram(bins = 15) +
  scale_x_continuous(breaks = scales::pretty_breaks())
```

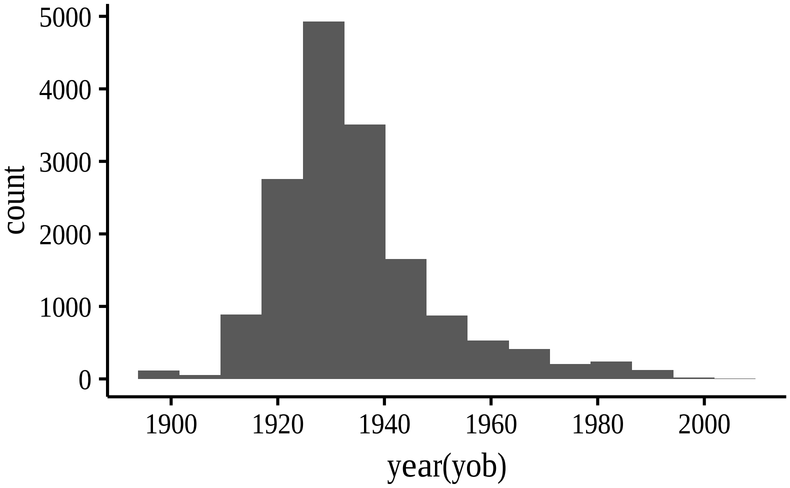

Figure 1: Distribution of year of birth

An ugly plot! I've used as few bins as possible to keep the plot non-disclosive. The important point is the implausible spike in births in 1900.

```
id_gender_age %>% filter(year(yob) == 1900) %>% count
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.2
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   112
```

```
id_gender_age %>% filter(year(yob) >= 1901 & year(yob) <= 1910) %>% count
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    68
```

Above I show the data shows 112 people born in 1900 whilst only 68 born between 1901 and 1910. I think it is safe to assume these are missing data (As a piece of comfort the linkage report for the main project has also found an improbable spike in the numbers born 1899 and 1900).

These will be dropped in the next section.

### 3.9 gender

We saw counts of each gender above but good to visualise as well. Interestingly, a basic geom_bar would show no NA values!! I'm going to pre-summarise then add pecentage labels so we don't miss this important info in the plot.

```
gender_summary <-
  id_gender_age %>%
  group_by(gender) %>%
  summarise(N = n()) %>%
  mutate(freq = N / sum(N),
         pct = round((freq * 100),2))

ggplot(gender_summary, aes(gender, N)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(pct,"%"), x = gender, y = N),
            size = 4, color = "black", vjust = -0.2) +
  labs(title = "Count of Gender",
       subtitle = "with percentage of total",
       x = "Gender",
       y = "Count")
```

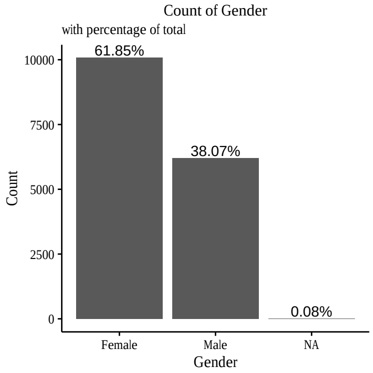

Figure 2: Count of gender

### 3.10 Conclusions on id_gender_age

- yob variable has been set as a Date object. I have set everyone's date to the 1st of January on the year of their birth. Will have to consider how this is used if calculating age etc.
- Very little missing data here - strong argument to drop these records but need to see the rest of the files first
- Age distribution makes sense but some very old and very young people in there, and a large "bump" in the year 1900 which may be a code for NA.

## 4 Load and clean home_care

### 4.1 Load data

```
load("Research/raw_data/home_care_raw.rds")
```

### 4.2 Glimpse home_care

```
glimpse(home_care_raw)
```

This data file has 106,111 observations of 8 variables; "Client ID" (integer), "Provider Type" (character), "Workers" (integer), "Days per week" (integer), "Hours per day" (double), "SCH_START_" (character), "SCH_END_DA" (character), and "Service Type" (character).

In this section I will

1. Create *clean* object and rename variables
2. Coerce variables to correct type
3. Assess missingness
4. Visualise variable distribution

## 4.3 Create *clean* object and rename variables

```
home_care <- home_care_raw     #Create duplicate

rm(home_care_raw)              #Remove raw data from memory

names(home_care) <- c("id", "hc_provider", "n_workers", "n_days", "n_hrs_per_day",
                      "hc_start_date", "hc_end_date", "hc_type")
```

## 4.4 Coerce variables to correct type + quality check

- `id` will become a character variable
- `hc_provider` needs factorised
- `n_workers` needs factorised
- `n_days` can stay as an integer for now
- `n_hrs_per_day` stays as double
- `hc_start_date` needs `Date` class
- `hc_end_date` as above
- `hc_type` needs factorised

## 4.5 Count duplicates

```
nrow(home_care) - nrow(distinct(home_care))
```

```
## [1] 26090
```

26,090 duplicated rows - these are exact duplicates, that's 24.6% of the dataframe! I've done some eyeball checking here as it is such a large number, certainly looks like they are there![2]

Now to check the variables...

## 4.6 Coerce id

```
home_care$id <- as.character(home_care$id)
```

How many individuals?

[2] I discussed this with Danny McAllion at Renfrewshire council. The reason there are so many duplicates is because each record relates to one episode of care. Therefore, duplicated records could indicate a 1 hour visit in the morning, a 1 hour visit in the afternoon, and a 1 hour visit in the evening - all with the same start and end dates and of the same type and all from the same provider. For that reason I can't drop them. I will have to aggregate episodes into daily care amounts.

```
home_care$id %>%
  unique(.) %>%
  length(.)
```

```
## [1] 11849
```

## 4.7 Coerce hc_provider

Check how many levels first

```
levels_hc_provider <- unique(home_care$hc_provider)
levels_hc_provider
```

```
## [1] "Care At Home - La"
## [2] "Care At Home - Independent-Exempt Status"
## [3] "Care At Home - Independent"
## [4] "Care At Home - Independent-Night Wakened"
## [5] "Care At Home - Independent-Sleepover"
## [6] "Care At Home - La-Exempt Status"
## [7] "Care At Home - Independent-Exempt Status-Sleepover"
```

So 7 levels - coerce to factor

```
home_care$hc_provider <- factor(home_care$hc_provider,
                             levels = levels_hc_provider,
                             labels = c("Local-Authority", "Ind-exempt", "Independent",
                                        "Ind-Night-awake", "Ind-night-sleep", "LA-exempt",
                                        "Ind-exempt-sleep"))
```

Count each factor level

```
fct_infreq(home_care$hc_provider) %>% fct_count(.)
```

*Not rendered due as potentially disclosive*

Some small numbers here so going to lump; Ind-night-sleep, Ind-exempt-sleep, Ind-Night-awake together.

```
home_care$hc_provider <- fct_lump(home_care$hc_provider, n = 4, other_level = "Ind-Night_combined")
```

And recount

```
fct_infreq(home_care$hc_provider) %>% fct_count(.)
```

```
## # A tibble: 5 x 2
##                    f     n
##               <fctr> <int>
## 1    Local-Authority 76924
## 2        Independent 26705
## 3         Ind-exempt  2165
## 4          LA-exempt   256
## 5 Ind-Night_combined    61
```

No missing data

### 4.8 Coerce n_workers

Check how many levels

```
unique(home_care$n_workers)
```

```
## [1] 1 2 4 3
```

Four levels - this could be treated as ordinal if required - don't need to add labels - pretty obvious.

```
home_care$n_workers <- factor(home_care$n_workers)
```

How many home care visits are multi-staffed?

```
fct_infreq(home_care$n_workers) %>% fct_count(.)
```

*Not rendered as potentially disclosive*
Vast majority with 1 or 2 workers. Very few over 2, I'll need to lump them

```
home_care$n_workers <- fct_lump(home_care$n_workers, n = 1,other_level = "More than 1")
```

And recount

```
fct_infreq(home_care$n_workers) %>% fct_count(.)
```

```
## # A tibble: 2 x 2
##             f     n
##        <fctr> <int>
## 1           1 92256
## 2 More than 1 13855
```

No missing data

### 4.9 Check n_days

Leaving this as an integer - will need to be combined with hours per day to create hours per week. However easier to summarise as a factor.

```
fct_infreq(factor(home_care$n_days)) %>% fct_count()
```

```
## # A tibble: 8 x 2
##        f     n
##   <fctr> <int>
## 1      7 67846
## 2      1 17595
## 3      2  8176
## 4      5  5369
## 5      3  3168
## 6      4  1873
## 7      6  1872
## 8      0   212
```

The majority of care is provided every day. Next level is weekly, then twice a week etc.

212 observations of 0 days per week. This will need to be subset beside hours per week to see if they get some hours of care - if so will need lumped into n_days = 1. If not - missing data and needs dropped.

## 4.10 Check n_hours_per_day

Summary stats

```
summary(home_care$n_hrs_per_day)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.2     0.5     0.5     0.8     1.0    24.0
```

Histogram

```
ggplot(home_care, aes(n_hrs_per_day)) +
  geom_histogram(bins = 10) +
  labs(y = "")
```

No implausible values - most values around about 1 hour - which makes sense. These will need aggregated with number of days per week to give an hours-per-week summary and then epiodes of the same type summed to give an ideas of a package of care.

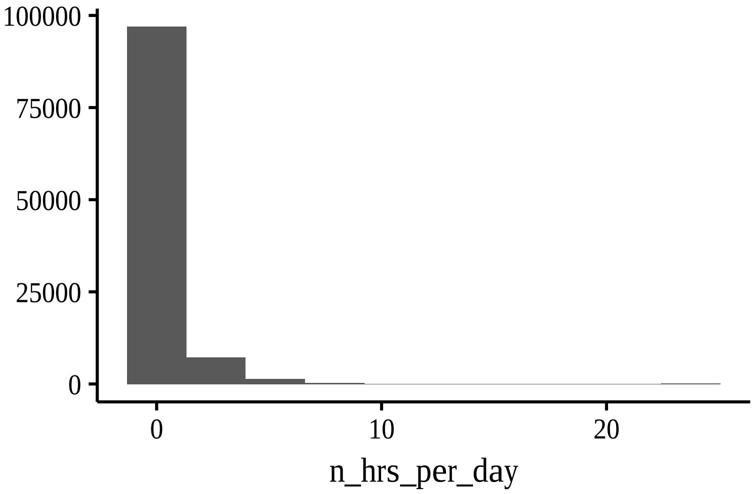

Figure 3: Distribution of hours per day of care received

## 4.11 Check the zero days per week subgroup

Quick look at those getting zero days per week care to see how many hrs of care they are getting

```
home_care %>%
  filter(n_days == 0) %>%
  select(n_hrs_per_day) %>%
  summary(.)
```

```
##  n_hrs_per_day
##  Min.   :0.2
##  1st Qu.:0.5
##  Median :0.5
##  Mean   :0.8
##  3rd Qu.:1.0
##  Max.   :4.0
```

No implausible values but strange to see 0.25hrs per week. This could be part of a care package. i.e. 1x15 min visit per week to e.g. prompt for meds is provided by private company, whilst rest of care is LA provided. This means that the care is provided one time a week - I find that not very likely The more obvious reason is that the zero values are missing data and we don't have a days-per-week value for these observations.

I'll code the zero values as NA

```
home_care$n_days[home_care$n_days == 0] <- NA
```

Check this has worked

```
fct_infreq(factor(home_care$n_days)) %>% fct_count()
```

```
## # A tibble: 8 x 2
##         f     n
##    <fctr> <int>
## 1       7 67846
## 2       1 17595
## 3       2  8176
## 4       5  5369
## 5       3  3168
## 6       4  1873
## 7       6  1872
## 8    <NA>   212
```

## 4.12 Coerce hc_start_date

Date object using `lubridate`

```
home_care$hc_start_date <- dmy(home_care$hc_start_date)
```

Look for implausible values

```
summary(home_care$hc_start_date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2004-05-01" "2008-02-13" "2010-04-16" "2010-11-04" "2013-12-02" "2016-03-30"
```

Looks reasonable - quick plot of start date

```
home_care %>%
  ggplot(aes(hc_start_date)) +
  geom_histogram(bins = 100) +
  scale_x_date(breaks = scales::pretty_breaks(n = 12)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

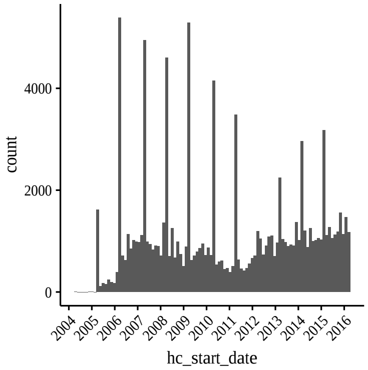

Figure 4: Distribution of start date of care

A couple of points, a)Really obvious seasonal pattern - haven't drilled right down but I'll bet my mortgage it is the start of the financial year. b)Are these new packages or re-assessed packages? c)Looks like run-in data for packages started in 2005 and still running 2006 (start of study period) - would expect the same (run-out) with end-dates. d)Data before April 2005 looks like could be dodgy..."

```
home_care %>%
  filter(hc_start_date < "2005-03-31")
```

*Not rendered as potentially disclosive*

There are a small number of records with start dates before 31st March 2005 and they all look fine with home care continuing for a number of years so we can leave them in.

## 4.13 Coerce 'hc_end_date

```
home_care$hc_end_date <- dmy(home_care$hc_end_date)
```

Quick summary again

```
summary(home_care$hc_end_date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2006-04-02" "2008-04-06" "2011-01-26" "2011-04-06" "2014-02-25" "2017-07-04"
```

Again a quick plot

```
home_care %>%
  ggplot(aes(hc_end_date)) +
  geom_histogram(bins = 100) +
  scale_x_date(breaks = scales::pretty_breaks(n = 12)) +
  theme(axis.text.x = element_text(angle = 45, size = 8,
                                   vjust = 1, hjust = 1))
```

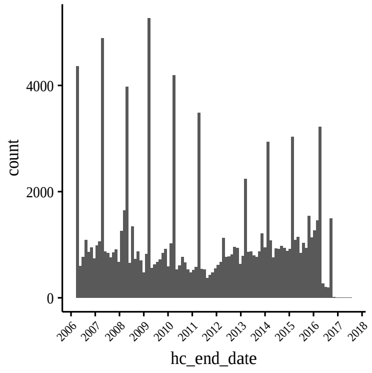

Figure 5: Distribution of end date of care

Not super clear but end-date pattern seems to peak at end of March.

## 4.14 Coerce hc_type

Final variable to clean -going to be a factor again so let's check unique levels 1st

```
unique(home_care$hc_type)
```

```
## [1] "Care at home (Mainstream)" "Housing Support"           "Rapid Response"
## [4] "Extra Care Housing"        "Community Mental Health"   "Overnight Service"
## [7] "Reablement"
```

This may shed some light onto the multiple start and end dates. Looks like it will be different services

Anyhow...factorise

```
home_care$hc_type <- factor(home_care$hc_type,
                        levels = c("Care at home (Mainstream)", "Housing Support",
                                   "Rapid Response", "Extra Care Housing",
                                   "Community Mental Health", "Overnight Service",
                                   "Reablement"))
```

Count

```
fct_infreq(home_care$hc_type) %>% fct_count(.)
```

```
## # A tibble: 7 x 2
##                           f     n
##                      <fctr> <int>
## 1 Care at home (Mainstream) 84224
## 2                Reablement 12794
## 3            Rapid Response  4414
## 4        Extra Care Housing  2773
## 5   Community Mental Health   714
## 6           Housing Support   665
## 7         Overnight Service   527
```

## 4.15 Create duration

I'll create a variable that calculates the duration of each episode of home care. I also want to check there are no home_care episodes that start and end on the same date.

```
home_care %<>%
  mutate(hc_duration = hc_start_date %--% hc_end_date %>% as.duration()) %>%
  filter(hc_duration != as.duration(0))
```

**3604 epidoes of care started and ended on the same day! They have been dropped.**

```
nrow(home_care)
```

```
## [1] 102507
```

How many individuals left in the dataset?

```
home_care$id %>% unique %>% length
```

```
## [1] 11737
```

Plot the duration

```
ggplot(home_care, aes(hc_duration/dyears(1))) +
  geom_histogram(bins = 35) +
  scale_x_continuous(breaks = scales::pretty_breaks()) +
  labs(
    x = "Duration of home care episode (years)"
  )
```

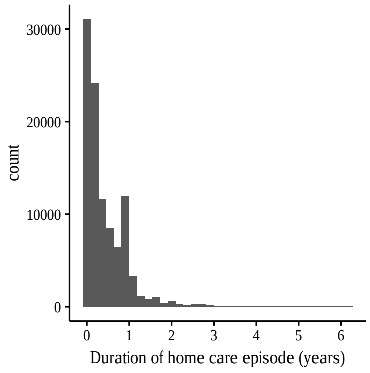

Figure 6: Distribution of duration of care

## 4.16 Hours per week

The data as given has `n_days` and `n_hrs_per_day` variables. I'm going to combine them to give me a `hc_hrs_per_week` variables.

```
home_care %<>%
  mutate(hc_hrs_per_week = n_days * n_hrs_per_day)
```

Plot this new variable

```
ggplot(home_care, aes(hc_hrs_per_week)) +
  geom_histogram()
```

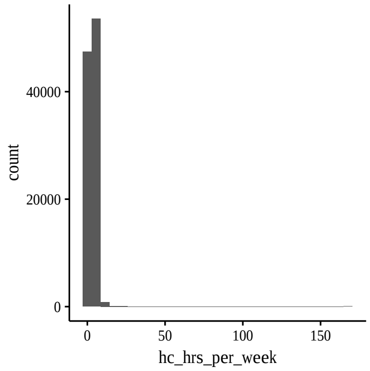

Figure 7: Distribution of total home care hours per week per episode

So the majority of care episodes are very low hours per week. However this value does not take into account the total hours per week when adding all types of care together. That summary is calculated in the next section.

### 4.17 Drop rows

I have already dropped rows where care epsiodes started and ended on the same day. Here I am going to drop all episodes of care that are classified as “Housing Support”, “Extra Care Housing”, and “Overnight Service”. From 2010 the SCS wanted these type of service classified as “Housing Support” and therefore are not included in Home Care hours. 24/7 and “live in” services were also not classified as home care after 2010 so I will keep only packages with a value of less than 24 for n_hrs_per_day. As this is a big change I will assign it to a new object

```
home_care_summary <-
  home_care %>%
  filter(!hc_type %in% c("Housing Support", "Extra Care Housing",
                         "Overnight Service")) %>%
  filter(n_hrs_per_day < 24)
```

## 5 Join, drop, and mutate

To conduct the full analysis I need to join id_gender_age and home_care_summary together. I’ll also drop those individuals with birth years in 1900 and episodes of care where the individual was under 65 years of age. Finally I’ll summarise the total hrs per week of each care package as a measure and create a new interval variable.

### 5.1 Join

```
hc_main <- left_join(home_care_summary, id_gender_age)
```

```
## Joining, by = "id"
```

### 5.2 Drop 1900

In the last section we saw an implausible spike in the number of people born in 1900. I’ll drop these individuals here.

```
hc_main %<>%
  filter(year(yob) != 1900)
```

## 5.3 Drop under 65

1st thing I need to do to drop under 65s is calculate age a the end date of each episode of care. I've chosen end date because some packages last a long period of time - this way I will keep people who turned 65 during their care package.[3]

```
hc_main %<>%
  mutate(age = year(hc_main$hc_end_date) - year(hc_main$yob))
```

Out of interest, how many individuals are there in total?

```
hc_main$id %>% n_distinct
```

```
## [1] 11569
```

And how many under 65?[4]

```
hc_main %>% filter(age < 65) %>% distinct(.$id) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  1619
```

```
hc_main %<>%
  filter(age >= 65)
```

## 5.4 Summarise total hours per week

I want to summarise `hc_main` by adding the variable `total_hrs_per_week`. This variable shows the sum of hours of care (`hc_hrs_per_week`) from the episodes that start and end on the same days (These are the e.g. morning 1hr, afternoon 45 mins, evening 1hr type episodes that makes up a complete **package**).

```
hc_main %<>%
  select(-n_days, -n_hrs_per_day, hc_duration) %>%  #drop unneeded variables
  group_by(id, age, gender, hc_type, hc_provider, n_workers, hc_start_date,
           hc_end_date) %>%
  summarise(total_hrs_per_week = sum(hc_hrs_per_week)) %>%
  arrange(as.numeric(id)) %>%
  ungroup()
```

## 5.5 Create `hc_interval`

Earlier I created a `duration` variable. This was coerced from an `interval` type. I nedd an uncoerced interval to help create a time series and also help with yearly summary stats.

[3] I only received year of birth from Renfrew and I have floored the date-of-birth to 1st January each year so my age calculations are not exactly accurate

[4] I won't be dropping all these individuals - only the observations where they were under the age of 65 when the care ended.

```
hc_main$hc_interval <- hc_main$hc_start_date %--%
  hc_main$hc_end_date
```

# 6 Descriptive statistics

Here I will summarise and describe `hc_main` table

I'll go through the variables one-by-one

## 6.1 id

```
hc_main %>% summarise(n = n_distinct(id))
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 10130
```

There are 38337 observations in this dataframe concerning 10130 individuals.

## 6.2 age

The dataframe shows repeated measures for each episode of care. As time moves on individuals ages increase. To get an accurate summary of the age distribution I'll average each individuals ages for all their years of data and then create summary stats from this. I'll also create an `age_group` variable to lump ages together

```
hc_main %<>%
  group_by(id) %>%
  mutate(mean_age = mean(age)) %>%
  ungroup %>%
  mutate(age_group = cut(hc_main$age, breaks = c(64, 75, 85, 95, Inf))) %>%
  mutate(age_group = fct_recode(age_group,
                               "65-75" = "(64,75]",
                               "76-85" = "(75,85]",
                               "86-95" = "(85,95]",
                               "over95" = "(95,Inf]"))
```

Summary statistics of age (I'm taking the mean ofeach individual's mean here - is that a cardinal sin?)

```
hc_main %>%
  select(id, mean_age) %>%
  distinct(id, .keep_all = TRUE) %>%
  summarise_at(vars(mean_age),
               funs(mean, median, sd, IQR), na.rm = TRUE) %>%
  gather(stat, value) %>% kable(booktabs = TRUE, digits = 2)
```

| stat | value |
|---|---:|
| mean | 81.6 |
| median | 82.0 |
| sd | 7.7 |
| IQR | 11.0 |

Grouped summary stats

```
hc_main %>%
  select(id, age_group) %>%
  distinct(id, .keep_all = TRUE) %>%
  tabyl(age_group) %>% kable(booktabs = TRUE, digits = 2)
```

| age_group | n | percent |
|---|---:|---:|
| 65-75 | 2551 | 0.25 |
| 76-85 | 4601 | 0.45 |
| 86-95 | 2813 | 0.28 |
| over95 | 165 | 0.02 |

```
hc_main %>%
  ggplot(aes(age_group)) +
  geom_bar()
```

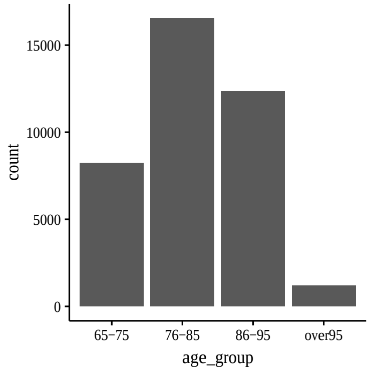

Figure 8: Bar plot of age groups

## 6.3 total_hrs_per_week

Create a group variable

```
hc_main %<>%
  mutate(hrs_group = cut(hc_main$total_hrs_per_week,
                         breaks = c(0, 0.75, 1.75, 2.75, 3.75,
                                    4.75, 5.75, 6.75, 7.75,
                                    8.75, 9.75, 15, 20, Inf))) %>%
  mutate(hrs_group = fct_recode(hrs_group,
                                "<1" = "(0,0.75]",
                                "1 - <2" = "(0.75,1.75]",
                                "2 - <3" = "(1.75,2.75]",
                                "3 - <4" = "(2.75,3.75]",
                                "4 - <5" = "(3.75,4.75]",
                                "5 - <6" = "(4.75,5.75]",
                                "6 - <7" = "(5.75,6.75]",
                                "7 - <8" = "(6.75,7.75]",
                                "8 - <9" = "(7.75,8.75]",
                                "9 - <10" = "(8.75,9.75]",
                                "10-15" = "(9.75,15]",
                                ">15 - 20" = "(15,20]",
                                ">20" = "(20,Inf]"))
```

Summary statistics

```
hc_main %>%
    summarise_at(vars(total_hrs_per_week),
               funs(mean, median, sd, IQR), na.rm = TRUE) %>%
  gather(stat, value) %>% kable(booktabs = TRUE, digits = 2)
```

| stat | value |
|---|---|
| mean | 7.0 |
| median | 6.0 |
| sd | 5.2 |
| IQR | 7.0 |

```
hc_main %>%
  tabyl(hrs_group, sort = TRUE) %>% kable(booktabs = TRUE, digits = 2)
```

| hrs_group | n | percent | valid_percent |
|---|---|---|---|
| 10-15 | 6586 | 0.17 | 0.17 |
| 3 - <4 | 5612 | 0.15 | 0.15 |
| 7 - <8 | 5018 | 0.13 | 0.13 |
| 1 - <2 | 4078 | 0.11 | 0.11 |
| 5 - <6 | 3678 | 0.10 | 0.10 |
| 2 - <3 | 3277 | 0.09 | 0.09 |
| >15 - 20 | 2800 | 0.07 | 0.07 |
| 8 - <9 | 2661 | 0.07 | 0.07 |
| 4 - <5 | 1373 | 0.04 | 0.04 |
| <1 | 1025 | 0.03 | 0.03 |
| 6 - <7 | 900 | 0.02 | 0.02 |
| 9 - <10 | 678 | 0.02 | 0.02 |
| >20 | 514 | 0.01 | 0.01 |
| NA | 137 | 0.00 | NA |

```
hc_main %>%
  ggplot(aes(hrs_group)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45,
                                   hjust = 1, vjust = 1))
```

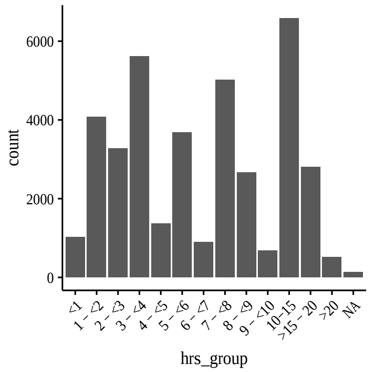

Figure 9: Bar plot of hours groups

## 6.4 hc_interval

I can have a look at the duration again. The calculation in Section 4 was before I dropped some rows.

```
hc_main %>%
  summarise(‘mean_duration(weeks)‘ = mean((as.duration(hc_interval))/dweeks(1)),
            ‘median_duration(weeks)‘ = median((as.duration(hc_interval))/dweeks(1)),
            ‘sd_duration(weeks))‘ = sd((as.duration((hc_interval))/dweeks(1))),
            ‘variance_duration(weeks)‘ = var((as.duration((hc_interval))/dweeks(1)))) %>%
  gather(stat, value) %>%
  kable(booktabs = TRUE)
```

| stat | value |
|---|---|
| mean_duration(weeks) | 22 |
| median_duration(weeks) | 11 |
| sd_duration(weeks)) | 26 |
| variance_duration(weeks) | 682 |

I don’t want to cut the interval variable into groups. Too much hassle! Really all I want to report is the percentage of care epsiodes that lasted 1 year or less

```
hc_main %<>%
  mutate(duration_weeks = as.duration(hc_interval)) %>%
  mutate(duration_weeks = duration_weeks/dweeks(1))
```

```
hc_main %>%
  filter(duration_weeks > 52) %>% count
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  2721
```

So 2721 observations of a total of 38339 are observations of episodes that last longer than a year - 7.1% meaning 92.9% of observations lasted one year or less.

## 6.5 gender

Same as age - only want to count each individual once!!

```
hc_main %>%
  select(id, gender) %>%
  distinct(id, .keep_all = TRUE) %>%
  tabyl(gender, sort = TRUE) %>%
  kable(booktabs = TRUE, digits = 2)
```

| gender | n | percent | valid_percent |
|---|---|---|---|
| Female | 6514 | 0.64 | 0.64 |
| Male | 3612 | 0.36 | 0.36 |
| NA | 4 | 0.00 | NA |

## 6.6 hc_type

```
hc_main %>%
  tabyl(hc_type, sort = TRUE) %>%
  kable(booktabs = TRUE, digits = 2)
```

| hc_type | n | percent |
|---|---|---|
| Care at home (Mainstream) | 32054 | 0.84 |
| Reablement | 4535 | 0.12 |
| Rapid Response | 1720 | 0.04 |
| Community Mental Health | 28 | 0.00 |
| Housing Support | 0 | 0.00 |
| Extra Care Housing | 0 | 0.00 |
| Overnight Service | 0 | 0.00 |

## 6.7 hc_provider

```
hc_main %>%
  tabyl(hc_provider, sort = TRUE) %>%
  kable(booktabs = TRUE, digits = 2)
```

| hc_provider | n | percent |
|---|---|---|
| Local-Authority | 29346 | 0.77 |
| Independent | 8807 | 0.23 |
| Ind-exempt | 184 | 0.00 |
| LA-exempt | 0 | 0.00 |
| Ind-Night_combined | 0 | 0.00 |

## 6.8 n_workers

```
hc_main %>%
  tabyl(n_workers, sort = TRUE) %>%
  kable(booktabs = TRUE, digits = 2)
```

| n_workers | n | percent |
|---|---|---|
| 1 | 34758 | 0.91 |
| More than 1 | 3579 | 0.09 |

# 7 Crosstabs

A look at how variables relate to each other

## 7.1 Age and Gender

```
hc_main %>%
  select(id, gender, age_group) %>%
  distinct(id, .keep_all=TRUE) %>%
  crosstab(gender, age_group) %>%
  adorn_crosstab(denom = "all", show_totals = TRUE) %>%
  kable(booktabs = TRUE)
```

| gender | 65-75 | 76-85 | 86-95 | over95 | Total |
|---|---|---|---|---|---|
| Female | 14.2% (1442) | 29.2% (2956) | 19.6% (1985) | 1.3% (131) | 64.3% (6514) |
| Male | 10.9% (1107) | 16.2% (1644) | 8.2% (827) | 0.3% (34) | 35.7% (3612) |
| NA | 0.0% (2) | 0.0% (1) | 0.0% (1) | 0.0% (0) | 0.0% (4) |
| Total | 25.2% (2551) | 45.4% (4601) | 27.8% (2813) | 1.6% (165) | 100.0% (10130) |

## 7.2 Duration and hc_type

I couldn't render the boxplots and histogram of duration earlier because the outliers would be disclosive. As over 90% of observations have duration of <1 year I'll filter by this and compare by type of care provided.

```
hc_main_plot <- hc_main %>% filter(duration_weeks <= 52) %>%  filter(!is.na(gender))
hc_main_no_type <- hc_main_plot %>% select(-hc_type)
```

```
ggplot(hc_main_plot, aes(duration_weeks, fill = hc_type)) +
  geom_histogram(bins = 25, data = hc_main_no_type, fill = "grey", alpha = 0.5) +
  geom_histogram(bins = 25, colour = "black") +
  facet_wrap(~hc_type) +
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette)
```

Potential plot for the chapter here - might need to drop Mental Health (although I don't think it is disclosive - I can't see anything here) and will definitely need to tidy it up a bit - we'll see.

The important point is that RR and Reablement are short-lived only. Easy to be missed out in SCS if they happen to be in a period not covered by the census. As there are fewer of them then it is unlikely that they stop and start with different hours as much as i would expect Care at home to do.

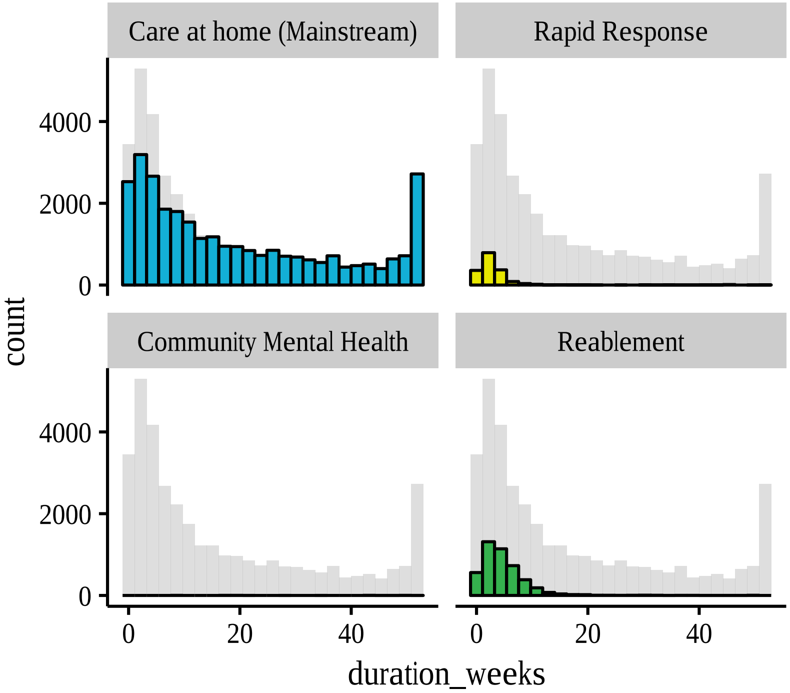

Figure 10: Duration of care by home care type

## 7.3 Gender and hours of care

Same as above but with different variables

```
hc_main_no_gender <- hc_main_plot %>% select(-gender)


ggplot(hc_main_plot, aes(hrs_group, fill = gender)) +
  geom_bar(data = hc_main_no_gender, fill = "grey", alpha = 0.5) +
  geom_bar(colour = "black") +
  facet_wrap(~gender) +
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

Pretty similar distributions across gender - less men as we knew.

## 7.4 Gender and duration

```
ggplot(hc_main_plot, aes(duration_weeks, fill = gender)) +
  geom_histogram(bins = 25, data = hc_main_no_gender, fill = "grey", alpha = 0.5) +
  geom_histogram(bins = 25, colour = "black") +
  facet_wrap(~gender)  +
```

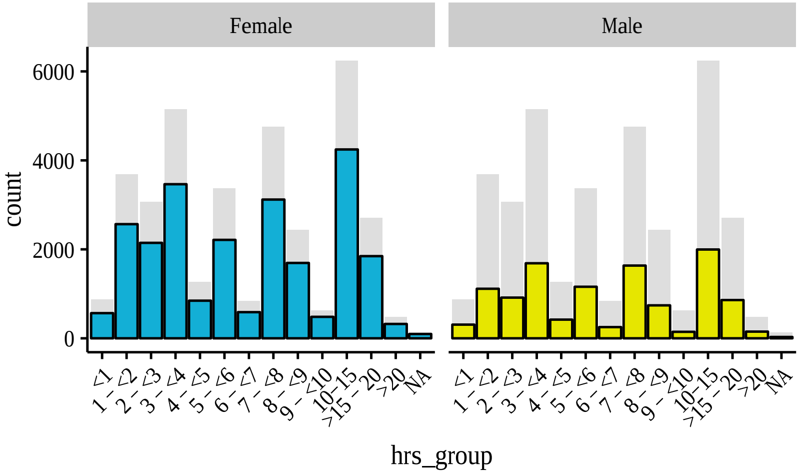

Figure 11: Hours of care by gender

```
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette)
```

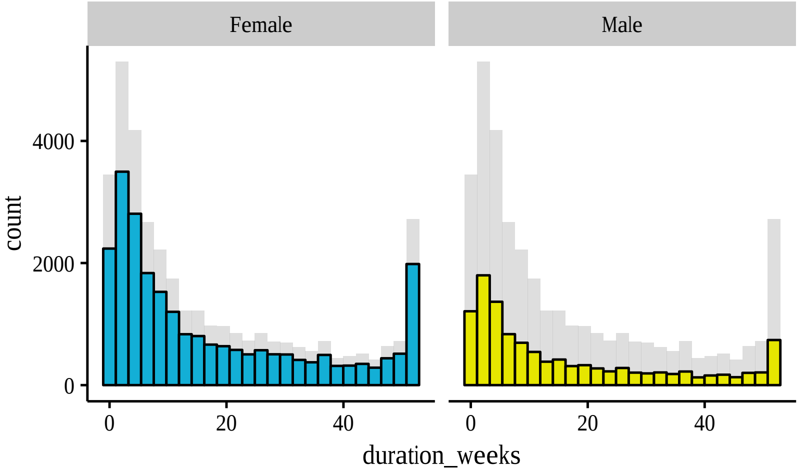

Figure 12: Duration of care by gender

Again no major differences

## 7.5 Type and hours of care

```
ggplot(hc_main_plot, aes(hrs_group, fill = hc_type)) +
  geom_bar(data = hc_main_no_type, fill = "grey", alpha = 0.5) +
  geom_bar(colour = "black") +
  facet_wrap(~hc_type) +
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

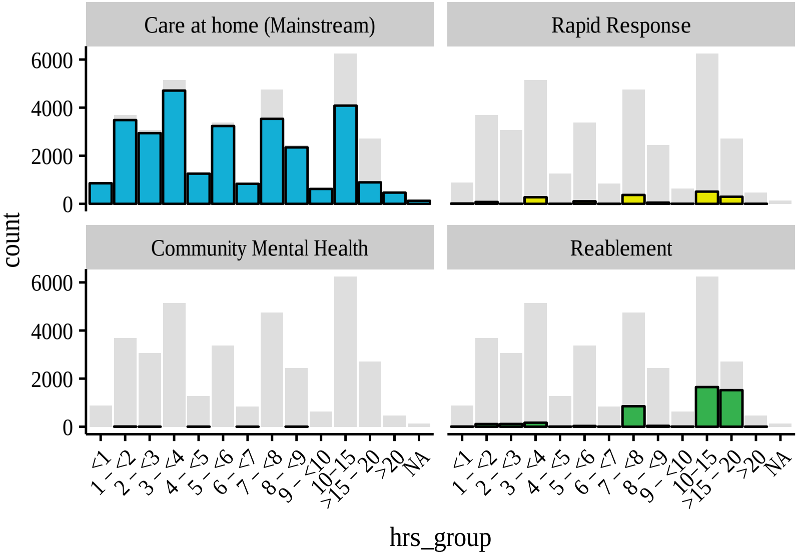

Figure 13: Hours of care by care type

Reablement packages tend to be very intense - 10-20 hrs per week usually.

## 7.6 Age and type

```
ggplot(hc_main_plot, aes(age_group, fill = hc_type)) +
  geom_bar(data = hc_main_no_type, fill = "grey", alpha = 0.5) +
  geom_bar(colour = "black") +
  facet_wrap(~hc_type) +
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

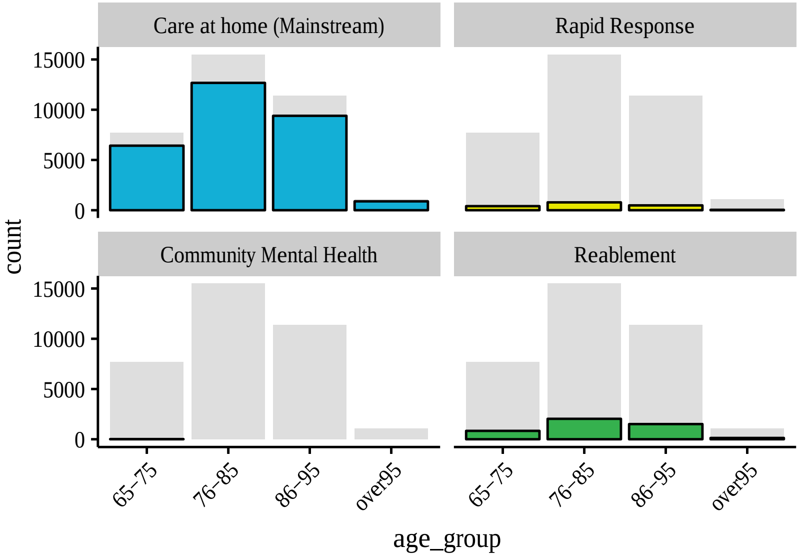

Figure 14: Age group by care type

## 7.7 Age and duration

```
hc_main_no_age <- hc_main_plot %>% select(-age_group)

ggplot(hc_main_plot, aes(duration_weeks, fill = age_group)) +
  geom_histogram(bins = 25, data = hc_main_no_age, fill = "grey", alpha = 0.5) +
  geom_histogram(bins = 25, colour = "black") +
  facet_wrap(~age_group)  +
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette)
```

Age doesn’t seem to affect duration of a care package (Figure 15)

## 7.8 Age and hrs of care

```
ggplot(hc_main_plot, aes(hrs_group, fill = age_group)) +
  geom_bar(data = hc_main_no_age, fill = "grey", alpha = 0.5) +
  geom_bar(colour = "black") +
  facet_wrap(~age_group) +
  guides(fill = FALSE) +
  scale_fill_manual(values = ubdc_palette) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

Surprisingly, age doesn’t affect hrs of care either.(Figure 16)

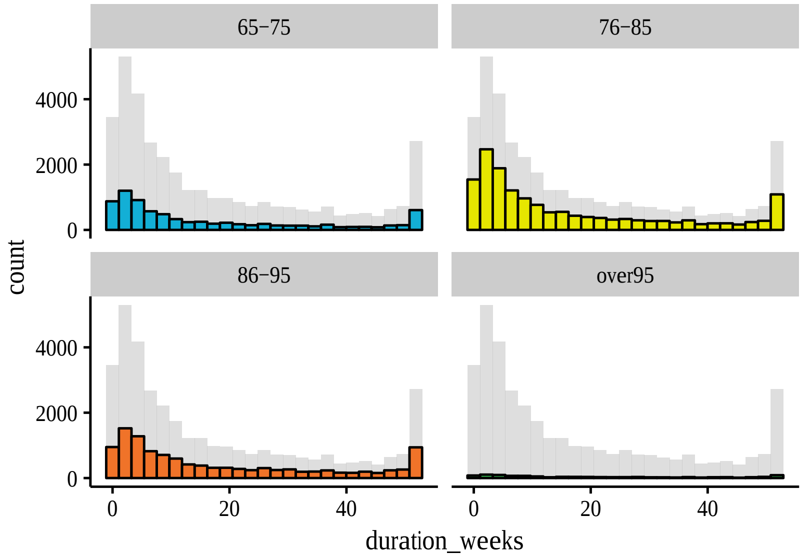

Figure 15: Duration by age group

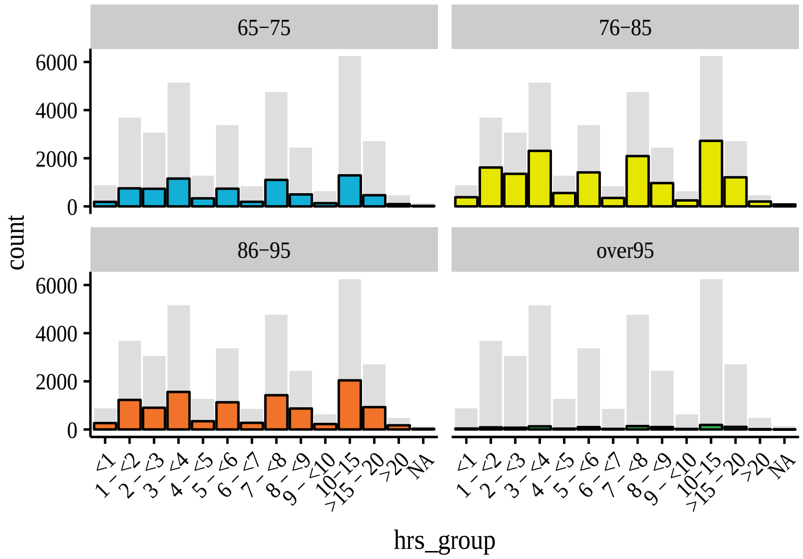

Figure 16: Hours of care by age group

# 8 Create time series

In order to assess the variation in the number of hours of care multiple individuals are receiving over time I need to aggregate the hours of care people get in weekly periods. [5] I have start and end dates of every package of care - but these are highly irregular. People also receive simultaneous packages of care which need to be summed.

In order to do this I am going to identify the hours of home care individuals were receiving on a sepcified day of the week from the end of March 2006 to the beginning of April 2016.

Total hours of care received in a week is then calculated by adding the results for each care type for each week.

## 8.1 Create dates vector

I need a vector of dates with the first Monday of every week for the study period, fortunately a new package `tibbletime` makes this very easy

```
dates <- create_series(2006-03-27 ~ 2016-04-03, 1~w)
dates[c(1:5, 518:523), ] %>% kable(booktabs=TRUE)
```

| date |
|---|
| 2006-03-27 |
| 2006-04-03 |
| 2006-04-10 |
| 2006-04-17 |
| 2006-04-24 |
| 2016-02-22 |
| 2016-02-29 |
| 2016-03-07 |
| 2016-03-14 |
| 2016-03-21 |
| 2016-03-28 |

I've printed the first and last 5 rows to show it has worked.

[5] I previously tried doing this in quarterly periods but ended up underestimating the change in hours between packages. Weekly data, whilst much bigger and more granularis much better for these purposes.

## 8.2 Create function and apply to data

Below is the function I will use to extract the hours of data received during given financial quarters.

```
add_dates <- function(data, date) {
  varname <-  as.character(date)
  mutate(data, !!varname := if_else(date %within% data$hc_interval,
                                    data$total_hrs_per_week,
                                    0))
}
```

In essence this function adds a column to the dataframe with the name of a given date. If that date falls within the home care package time interval the column is given the value of `total_hrs_per_week`. If not it is given the value zero. [6]

Now I will apply the function using all 523 time points from the `dates` object. This creates a list of 523 dataframes with one column added so I bind these together and then remove the duplicated columns. [7]

```
hcts <-
  map(dates$date,
      add_dates,
      data = hc_main) %>%
  bind_cols(.)

hcts <- hcts[!duplicated(as.list(hcts))]
```

I will now add all packages of care together for each individual at each week and then tidy the data into long format assigning this to a new object called `hcts_summ`. I'll also make the `hcts_summ` dataframe "Time aware" using `tibbletime`. This will help with summaries and plotting.

```
hcts_summ <-
  hcts %>%
  group_by(id) %>%
  summarise_at(vars(‘2006-03-27‘:‘2016-03-28‘), sum) %>%
  ungroup %>%
  gather(week, hrs_homecare, ‘2006-03-27‘:‘2016-03-28‘) %>%
  arrange(as.numeric(id))
hcts_summ$week <- ymd(hcts_summ$week)

hcts_summ <- as_tbl_time(hcts_summ, index = week)
```

`hcts_summ` now has 5297990 observations of 3 variables; `id`, `week`, and `hrs_homecare`. This means I have one observation on hrs of homecare for

[6] Ideally I would make the Null value NA but both sides of an `if_else` statement must be of the same type - in this case numeric.

[7] I had a stack overflow fest on this. I am sure there must be a solution using `left_join` but could not get any to work. `purrr::reduce` didn't work for me either!

every week in the 10 year study period for all 10130 individuals. There are 721492 observations with non-zero values.

The `hcts` object will be helpful for assessing contiguous packages of care

# 9 Describe time series

Here I will describe the time_series objects created in section 8

## 9.1 `hrs_homecare`

I want to check the `hrs_homecare` variable 1st of all

```
hcts_summ %>% filter(hrs_homecare != 0) %>%
  summarise(mean = mean(hrs_homecare),
            median = median(hrs_homecare),
            sd = sd(hrs_homecare),
            IQR = IQR(hrs_homecare)) %>%
  kable(booktabs = TRUE)
```

```
## Note: 'index' has been removed. Removing 'tbl_time' class.
```

| mean | median | sd | IQR |
|---:|---:|---:|---:|
| 6.4 | 5.2 | 5.4 | 6.2 |

Difficult to visualise this without being disclosive. The vast majority of packages are under 20 hours per week with some outliers way out with high numbers

## 9.2 Weekly count

Create an object counting the number of people getting care in each week of the time series

```
hcts_week_count <-
  hcts_summ %>%
  filter(hrs_homecare != 0) %>%
  group_by(week) %>%
  summarise(count = n()) %>%
  mutate(year = year(week)) %>%
  ungroup
```

Add in the values of SCS returned by Renfrewshire Council.

```
hcts_week_count %<>%
  mutate(scs = case_when(week == "2006-03-27" ~ 1510,
                         week == "2007-03-26" ~ 1520,
                         week == "2008-03-24" ~ 1490,
                         week == "2009-03-23" ~ 1520,
                         week == "2010-03-22" ~ 1530,
                         week == "2011-03-21" ~ 1290,
                         week == "2012-03-19" ~ 1300,
                         week == "2013-03-18" ~ 1410,
                         week == "2014-03-24" ~ 1520,
                         week == "2015-03-23" ~ 1760,
                         week == "2016-03-21" ~ 1740))
```

And then plot it (Figure 17)

```
hcts_week_count %>%
  ggplot(aes(week, count)) +
  geom_line() +
  geom_point(aes(week, scs)) +
  labs(
    caption = "Points show Renfrew return to SCS"
  ) +
  scale_x_date(breaks = scales::pretty_breaks(n=11))
```

```
## Warning: Removed 512 rows containing missing values (geom_point).
```

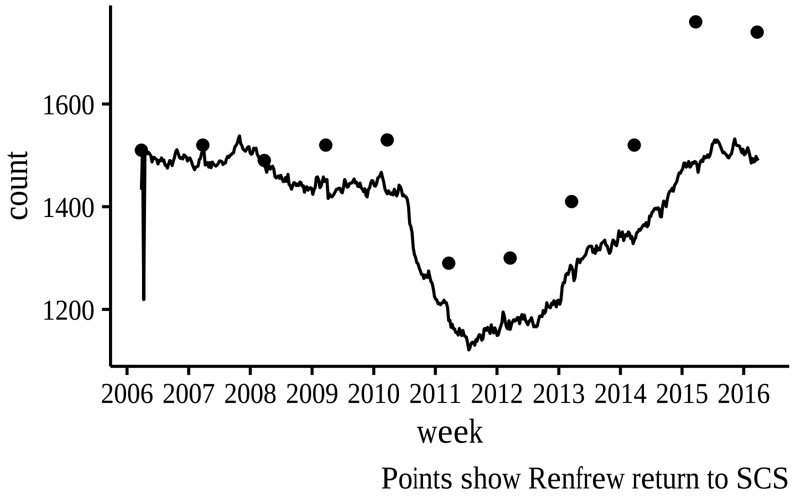

Figure 17: Number of people receiving home care by week

So the first point is my counts are a little lower than the SCS return. Good to see that they follow the same trend though (2009 onwards) - this suggests

I am not counting something structural - may be another type of care not included in the data I have. I don't think the Housing Support and 24/7 care type that I dropped would make any major impact on this - they were very small numbers.

I'll create a table showing the difference between my counts and SCS return in a week as close to the census as possible.

```
hcts_week_count %>%
  filter(!is.na(scs)) %>%
  mutate(diff = count - scs) %>%
  kable(booktabs = TRUE)
```

| week | count | year | scs | diff |
|---|---|---|---|---|
| 2006-03-27 | 1433 | 2006 | 1510 | -77 |
| 2007-03-26 | 1507 | 2007 | 1520 | -13 |
| 2008-03-24 | 1480 | 2008 | 1490 | -10 |
| 2009-03-23 | 1452 | 2009 | 1520 | -68 |
| 2010-03-22 | 1425 | 2010 | 1530 | -105 |
| 2011-03-21 | 1178 | 2011 | 1290 | -112 |
| 2012-03-19 | 1161 | 2012 | 1300 | -139 |
| 2013-03-18 | 1283 | 2013 | 1410 | -127 |
| 2014-03-24 | 1335 | 2014 | 1520 | -185 |
| 2015-03-23 | 1487 | 2015 | 1760 | -273 |
| 2016-03-21 | 1493 | 2016 | 1740 | -247 |

The second thing to note is that pre 2010 this line look the way I want it to (apart from an anomoly early 2006). Some variation but total numbers pretty stable throughout the year.

After 2010 it is all over the place! The SCS points act as a good reference - they are in the end of March every year. 2010-2011 shows a big drop in numbers throughout the year. 2011-2012 is pretty stable and all subsequent years show gradual increases throughout the year with a little stability returning 2015-2016.

This is important beacuse if these are different individuals entering and leaving care services then the SCS is potentially missing a lot of people who receive care. The question is how many individuals are receiving short-term care between two census dates?

## 9.3 Summarise variation in weekly count by year

Need to add a financial year variable to check numbers from April - March

```
hcts_week_count %<>%
  mutate(fin_year = factor(year(hcts_week_count$week) -
                             (month(hcts_week_count$week) <= 03),
                           labels = c("2005/06", "2006/07", "2007/08",
                                      "2008/09", "2009/10", "2010/11",
                                      "2011/12", "2012/13", "2013/14",
                                      "2014/15", "2015/16")
                           )
         )
```

```
levels(hcts_week_count$fin_year)
```

```
##  [1] "2005/06" "2006/07" "2007/08" "2008/09" "2009/10" "2010/11" "2011/12" "2012/13"
##  [9] "2013/14" "2014/15" "2015/16"
```

And now I can visualise with a box plot (Figure 18)

```
hcts_week_count %>%
  filter(fin_year != "2005/06") %>%
  ggplot(aes(fin_year, count, fill = fin_year)) +
  geom_boxplot() +
  scale_fill_manual(values = ubdc_palette) +
  guides(fill = FALSE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  labs(
    x = "Financial Year",
    y = "No. of people"
  )
```

I can also create a summary table of these boxplot values

```
hcts_week_count %>%
  group_by(fin_year) %>%
  filter(fin_year != "2005/06") %>%
  summarise(min = min(count),
            max = max(count),
            mean = mean(count),
            median = median(count),
            range = max(count) - min(count),
            IQR = IQR(count),
```

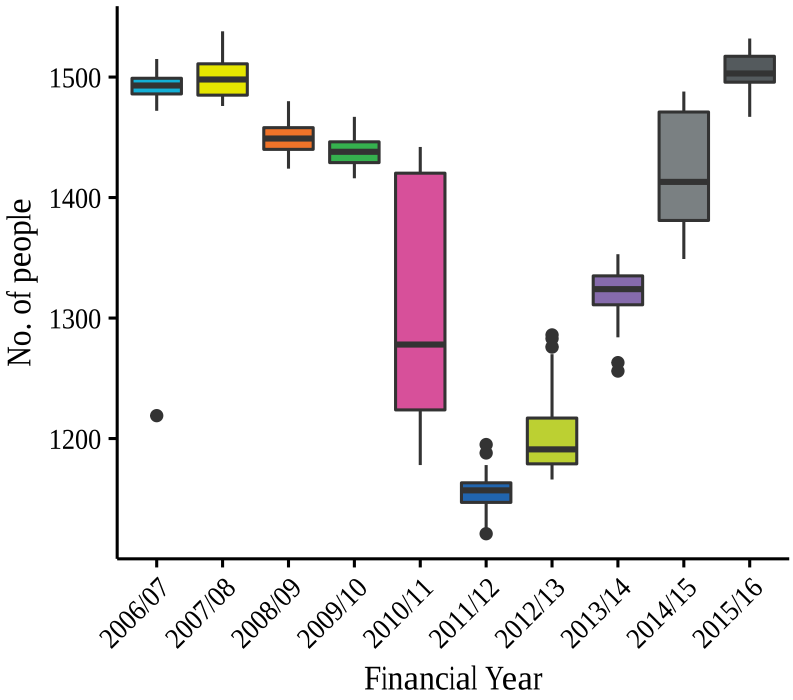

Figure 18: Distribution of weekly count of people receiving home care

```
          sd = sd(count)) %>%
  kable(booktabs = TRUE)
```

```
## Note: 'index' has been removed. Removing 'tbl_time' class.
```

| fin_year | min | max | mean | median | range | IQR | sd |
|---|---|---|---|---|---|---|---|
| 2006/07 | 1219 | 1515 | 1488 | 1493 | 296 | 13 | 39 |
| 2007/08 | 1476 | 1538 | 1498 | 1498 | 62 | 26 | 16 |
| 2008/09 | 1424 | 1480 | 1450 | 1449 | 56 | 18 | 14 |
| 2009/10 | 1416 | 1467 | 1439 | 1438 | 51 | 17 | 12 |
| 2010/11 | 1178 | 1442 | 1311 | 1278 | 264 | 196 | 90 |
| 2011/12 | 1121 | 1195 | 1155 | 1157 | 74 | 16 | 15 |
| 2012/13 | 1166 | 1286 | 1205 | 1191 | 120 | 38 | 35 |
| 2013/14 | 1256 | 1353 | 1321 | 1324 | 97 | 24 | 21 |
| 2014/15 | 1349 | 1488 | 1421 | 1413 | 139 | 90 | 47 |
| 2015/16 | 1467 | 1532 | 1505 | 1503 | 65 | 22 | 14 |

From these figures we can see there was not a lot of variation in the weekly number of people receiving home care before the Financial year 2010/11. That year saw a large, gradual decrease. The follwing years haven't shown as much variation as that year but more than previously. This reflects the fact that the number of people receiving care has recovered to pre 2010 levels.

## 9.4 Calculate weekly difference in number of people getting care

```
hcts_week_count %<>%
  group_by(fin_year) %>%
  mutate(diff = count - lag(count, default = first(count))) %>%
  mutate(pct_change = (count - lag(count)) / lag(count) * 100) %>%
  ungroup
```

Plot it (Figure 19)

```
hcts_week_count %>%
  filter(fin_year != "2005/06") %>%
  filter(diff <200 & diff > -200) %>%
  ggplot(aes(fin_year, diff, fill = fin_year)) +
  geom_boxplot() +
  geom_hline(yintercept = 0, alpha = 0.3) +
  scale_fill_manual(values = ubdc_palette) +
  guides(fill = FALSE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  labs(
    x = "Financial year",
    y = "Difference"
  )
```

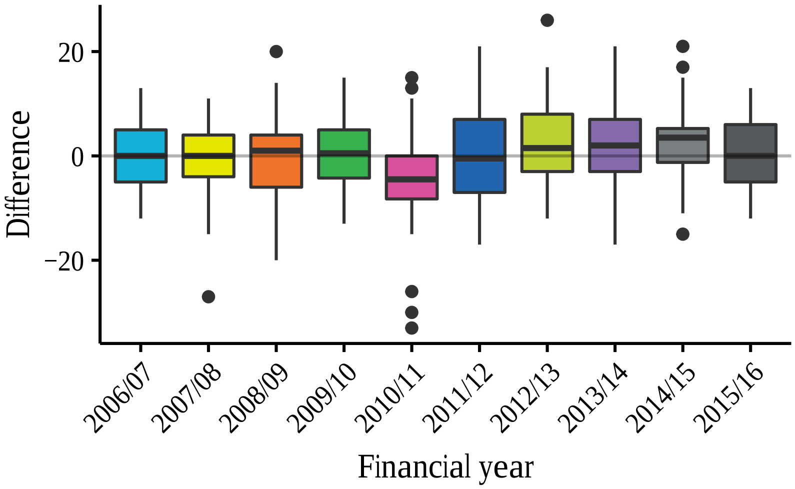

Figure 19: Distribution of weekly difference in people receiving home care

A line plot may be more clear? (Figure 20)

```
hcts_week_count %>%
  filter(diff <200 & diff > -200) %>% #drop the outlier
  ggplot(aes(week, diff, colour = fin_year)) +
  geom_line() +
  geom_hline(yintercept = 0) +
  scale_x_date(breaks = scales::pretty_breaks(n = 11)) +
  scale_colour_manual(values = ubdc_palette) +
  guides(colour = FALSE)
```

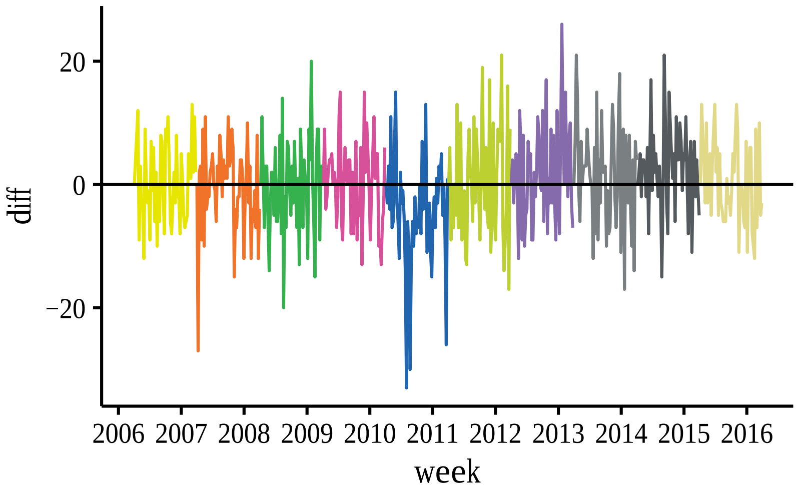

Figure 20: Weekly difference of numebr of people receiving home care

Noisy - Let's make it monthly? (Figure 21)

```
hcts_week_count %>%
  as_period(1~m) %>%
  ggplot(aes(week, diff, colour = fin_year)) +
  geom_line() +
  geom_hline(yintercept = 0) +
  scale_x_date(breaks = scales::pretty_breaks(n = 11)) +
  scale_colour_manual(values = ubdc_palette) +
  guides(colour = FALSE)
```

## 10 Still to do

I still need to assess, at an individual level, the differences in hours of care in the census week and throughout the year.

To do this I need to analyse indivudal's contiguous packages of care.

We have seen that packages are quite short lived - Do they get restarted on the same hours after re-assessment, or are there often large changes in the amount of care received. How often are there breaks in care? (e.g. for hospital stay or holidays etc.)

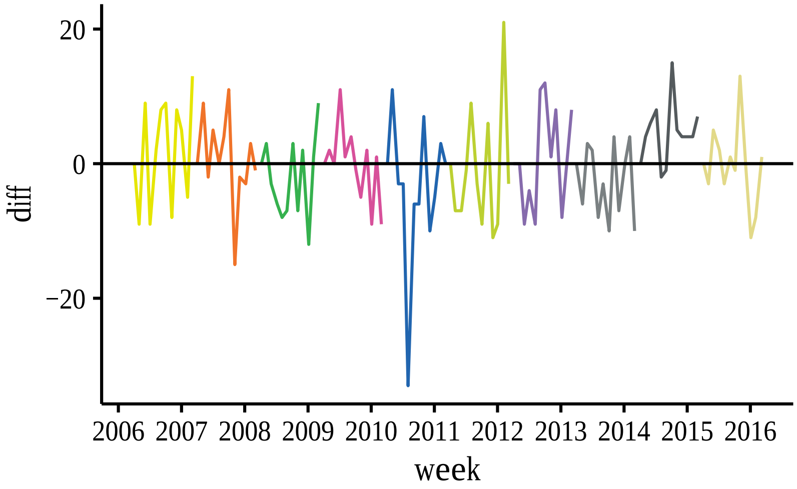

Figure 21: Monthly difference in number of people receiving home care

This last part should complete the anlaysis and I should be able to write up the chapter when it is done. This interim report contains draft plots only and is generated for feedback from supervisors.