# Demographic Information

## Load Data

```
library(patchwork)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# library(ggplot2)

rm(list=ls())
OUPUTDATA_LOCATION = file.path(getwd(),"Data/MCS Data/Processed Data")

# Load Original Data
load(file.path(OUPUTDATA_LOCATION,"/DataWithExclusionsNoImputations/df0_BothGroups_FINAL.Rdata"))
```

### Ethnicity

Cleaning required. They report ethnicity differently in Englamdf

```
df0 = df0_BothGroups
Ethnicity_England = df0$S2_Ethnicity_England_BDCEEAA0 %>%
                    dplyr::recode(
                                  `1` = "White",
                                  `2` = "White",
                                  `3` = "White",

                                  `4` = "Mixed",
                                  `5` = "Mixed",
                                  `6` = "Mixed",
                                  `7` = "Mixed",

                                  `8` = "Indian",
                                  `9` = "Pakistani",
```

```
                                    `10` = "Bangladeshi",
                                    `11` = "Other",

                                    `12` = "Black",
                                    `13` = "Black",
                                    `14` = "Black",

                                    `15` = "Chinese",

                                    `95` = "Other",
                                    )
```

```
## Warning: Unreplaced values treated as NA as .x is not compatible. Please specify
## replacements exhaustively or supply .default
```

```
Ethnicity_Wales = df0$S2_Ethnicity_Wales_BDCEWAA0 %>%
                    dplyr::recode(
                                    `1` = "White",
                                    `2` = "White",
                                    `3` = "White",
                                    `4` = "White",

                                    `5` = "Mixed",
                                    `6` = "Mixed",
                                    `7` = "Mixed",
                                    `8` = "Mixed",

                                    `9` = "Indian",
                                    `10` = "Pakistani",
                                    `11` = "Bangladeshi",
                                    `12` = "Other",

                                    `13` = "Black",
                                    `14` = "Black",
                                    `15` = "Black",

                                    `16` = "Chinese",

                                    `95` = "Other",
                                    )
```

```
## Warning: Unreplaced values treated as NA as .x is not compatible. Please specify
## replacements exhaustively or supply .default
```

```
Ethnicity_Scotland = df0$S2_Ethnicity_Scotland_BDCESAA0 %>%
                    dplyr::recode(
                                    `1` = "White",
                                    `2` = "White",
                                    `3` = "White",
                                    `4` = "White",

                                    `5` = "Mixed",
```

```
                                    `6` = "Indian",
                                    `7` = "Pakistani",
                                    `8` = "Bangladeshi",

                                    `9` = "Chinese",

                                    `10` = "Other",

                                    `11` = "Black",
                                    `12` = "Black",
                                    `13` = "Black",

                                    `95` = "Other",
                                    )
```

```
## Warning: Unreplaced values treated as NA as .x is not compatible. Please specify
## replacements exhaustively or supply .default
```

```
Ethnicity_NI = df0$S2_Ethnicity_NI_BDCENAA0 %>%
                     dplyr::recode(
                                    `1` = "White",

                                    `2` = "Chinese",

                                    `3` = "White",
                                    `4` = "White",

                                    `5` = "Mixed",

                                    `6` = "Indian",
                                    `7` = "Pakistani",
                                    `8` = "Bangladeshi",

                                    `9` = "Chinese",

                                    `10` = "Other",

                                    `11` = "Black",
                                    `12` = "Black",
                                    `13` = "Black",

                                    `95` = "Other",
                                    )
```

```
## Warning: Unreplaced values treated as NA as .x is not compatible. Please specify
## replacements exhaustively or supply .default
```

```
Ethnicity_Matrix_original =
                   cbind(
                     df0$S2_Ethnicity_England_BDCEEAA0,
                     df0$S2_Ethnicity_Wales_BDCEWAA0,
                     df0$S2_Ethnicity_Scotland_BDCESAA0,
```

```
                  df0$S2_Ethnicity_NI_BDCENAA0
                )


Ethnicity_Matrix = cbind(Ethnicity_England,
                         Ethnicity_Wales,
                         Ethnicity_Scotland,
                         Ethnicity_NI)

# Ethnicity_Matrix %>%
#   apply(., 1, function(x) length(which(!is.na(x)))) %>%
#   table()

Country = Ethnicity_Matrix %>%
            apply(.,1, function(x) which(!is.na(x))) %>%
            unlist()

Ethnicity = Ethnicity_Matrix %>%
            apply(., 1, function(x) as.vector(na.omit(x))[1])

# Ethicity = Ethnicity_Matrix[]



# Check i haven't missed any codes
# Ethnicity_Matrix_original %>%
#   as.data.frame() %>%
#   slice(which(apply(Ethnicity_Matrix,1,function(x) length(which(!is.na(x))))==0)) %>%
#   unlist(c(t(.))) %>%
#   table()
```

Table of ethnicity percentages. Note that there are missing values for 541 participants.

```
EthnicityTbl =
  Ethnicity %>%
    table() %>%
    data.frame() %>%
    dplyr::arrange(Freq) %>%
    dplyr::mutate(Percent = format(Freq/sum(Freq), nsmall=4,digits=0))

EthnicityTbl
```

```
##             . Freq Percent
## 1     Chinese   11  0.0011
## 2       Other  120  0.0122
## 3 Bangladeshi  188  0.0191
## 4      Indian  250  0.0254
## 5       Mixed  250  0.0254
## 6       Black  264  0.0268
## 7   Pakistani  460  0.0468
## 8       White 8292  0.8431
```

### Age

I couldn't find child age at interview data for MCS1 in the MCS1 data release, but it provided in harmonised BMI data release, with some missing data.

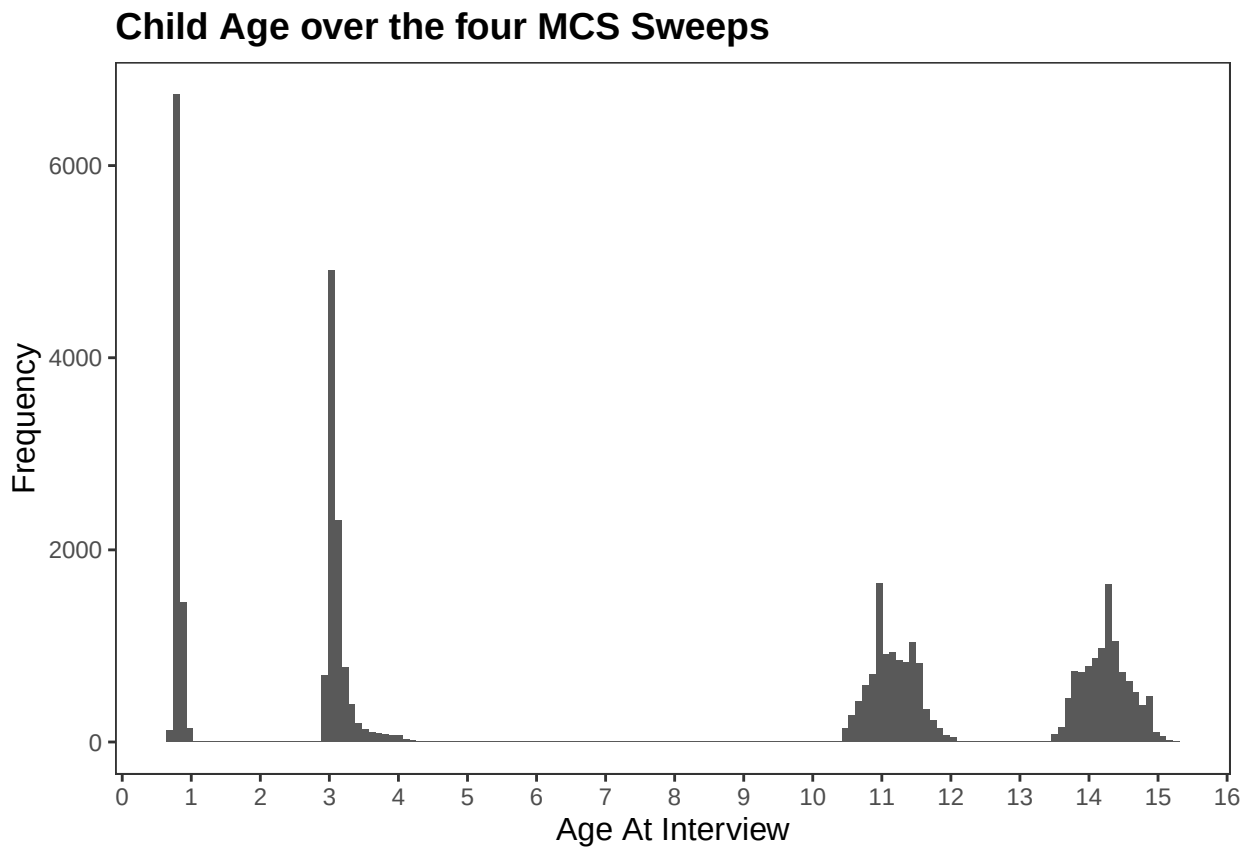
For sweeps 5 & 6 I'm only given month/year of interview and birth, which may explain some odd looking peaks in the data.

```
Age_Matrix = df0_BothGroups %>%
             dplyr::select(S1 = S1_Health_Age, S2=S2_Age, S5=S5_AgeAtInterview, S6=S6_Age_Est) %>%
             tidyr::pivot_longer(cols=S1:S6, values_to = "Age")

AgePlot =
ggplot(Age_Matrix, aes(x=Age)) +
  geom_histogram(bins=150) +
  jtools::theme_apa() +
  labs(title="Child Age over the four MCS Sweeps",y="Frequency", x="Age At Interview") +
  scale_x_continuous(minor_breaks =seq(0,16,by=.5),breaks=seq(0,16,by=1), labels=seq(0,16,by=1))

AgePlot
```

```
## Warning: Removed 2841 rows containing non-finite values (stat_bin).
```




```
ggsave(file=file.path("Plots","AgeMCS.png"), device="png", width=6, height=3)
```

```
## Warning: Removed 2841 rows containing non-finite values (stat_bin).
```

```
AgePlotTrimmed =
  AgePlot + coord_cartesian(y=c(0,2000))

ggsave(plot=AgePlotTrimmed,file=file.path("Plots","AgeMCS_trimmed.png"), device="png", width=6, height=3

## Warning: Removed 2841 rows containing non-finite values (stat_bin).
```

### Child Age Table

```
Age_Matrix %>%
  dplyr::group_by(name) %>%
  summarise(mean=mean(Age, na.rm=TRUE),
            sd=sd(Age, na.rm=TRUE),
            N_NotMissing = length(which(!is.na(Age))))
```

```
## # A tibble: 4 x 4
##   name    mean     sd N_NotMissing
##   <chr>  <dbl>  <dbl>        <int>
## 1 S1     0.807 0.0398         8452
## 2 S2     3.13  0.195          9844
## 3 S5    11.2   0.326          9991
## 4 S6    14.3   0.338         10376
```

### Gender

```
colnames(df0_BothGroups)[grepl("imd",tolower(colnames(df0_BothGroups)))]
```

```
## [1] "S1_IMD_Income"       "S1_IMD_Employment"   "S1_IMD_Health"
## [4] "S1_IMD_Education"    "S1_IMD_Housing"      "S1_IMD_Crime"
## [7] "S1_IMD_Living"       "S6_SES_IMD_Combined" "S1_IMDDeprivation"
```

```
df0_BothGroups %>%
  dplyr::select(S1_Main_Sex,S2_Sex_bhcsex00,S2_Sex,S5_Sex_ECCSEX00) %>%
  tidyr::pivot_longer(cols=S1_Main_Sex:S5_Sex_ECCSEX00) %>%
  dplyr::group_by(name) %>%
  summarise(N_NotMissing = length(which(!is.na(value))))
```

```
## # A tibble: 4 x 2
##   name            N_NotMissing
##   <chr>                  <int>
## 1 S1_Main_Sex            10376
## 2 S2_Sex                  9853
## 3 S2_Sex_bhcsex00         9853
## 4 S5_Sex_ECCSEX00         9991
```

```
df0_BothGroups %>%
  dplyr::summarise(N_Male = length(which(S2_Sex_bhcsex00==1)),
                   N_Female = length(which(S2_Sex_bhcsex00==2)),
                   Percent_Male = N_Male /(N_Male + N_Female),
                   Percent_Female = N_Female /(N_Male + N_Female)
                   )
```

```
##   N_Male N_Female Percent_Male Percent_Female
## 1   4873     4980    0.4945702      0.5054298
```

```
  # table(.) %>%
  # `names<-` (c("male", "female"))
```

### Geographic Data

Interview Country

```
S1Geo = data.table::fread("Data/MCS Data/Raw Data/UKDA-4683-tab MCS1/tab/mcs1_geographically_linked_data
S1Geo = S1Geo[match(df0_BothGroups$MCSID, S1Geo$mcsid),]

S1Geo %>%
  dplyr::mutate(country_new = dplyr::recode(aactry00,
                `1` = "England",
                `2` = "Wales",
                `3` = "Scotland",
                `4` = "Northern Ireland")) %>%
  group_by(country_new) %>%
  summarise(N=n()) %>%
  mutate(Perc = round(N/sum(N), digits=4))
```

```
## # A tibble: 4 x 3
##   country_new          N   Perc
##   <chr>            <int>  <dbl>
## 1 England           6625 0.638
## 2 Northern Ireland  1029 0.0992
## 3 Scotland          1172 0.113
## 4 Wales             1550 0.149
```

### Who were the responders in the first and second sweep of the study?

```
ResponderTable =
df0_BothGroups %>%
  dplyr::select(S1_Main_Identity_AMDRES00,S2_Main_Identity_BMDRES00) %>%
  tidyr::pivot_longer(cols=S1_Main_Identity_AMDRES00:S2_Main_Identity_BMDRES00) %>%
  mutate(value = dplyr::recode(value,
                               `1`= "Natural Mother",
                               `2`= "Natural Father",
                               `3`= "Adoptive Mother",
                               `5`= "Foster Mother",
                               `6`= "Foster Father",
                               `9`= "Grandmother")) %>%
  dplyr::group_by(name) %>%
  summarise(N_Mother = length(which(value== "Natural Mother")),
            N_Father = length(which(value== "Natural Father")),
            N_Adoptive_Mother = length(which(value== "Adoptive Mother")),
            N_Foster_Mother = length(which(value== "Foster Mother")),
            N_Fother_Father = length(which(value== "Foster Father")),
```

```
            N_Grandmother = length(which(value== "Grandmother")),
            ) %>%
  dplyr::select(-name) %>%
  t() %>%
  `colnames<-`((c("S1","S2"))) %>%
  dplyr::as_tibble(rownames = NA) %>%
  rownames_to_column() %>%
  mutate(S1_Percent = round(S1/sum(S1),4),
         S2_Percent = round(S2/sum(S2),3))
```

```
## Warning: Unreplaced values treated as NA as .x is not compatible. Please specify
## replacements exhaustively or supply .default
```

```
ResponderTable
```

```
## # A tibble: 6 x 5
##   rowname              S1    S2 S1_Percent S2_Percent
##   <chr>             <int> <int>      <dbl>      <dbl>
## 1 N_Mother          10367  9682     0.999       0.987
## 2 N_Father              5   113     0.0005      0.012
## 3 N_Adoptive_Mother     1     2     0.0001      0
## 4 N_Foster_Mother       1     0     0.0001      0
## 5 N_Fother_Father       0     1     0           0
## 6 N_Grandmother         2     8     0.0002      0.001
```

```
# table(df0_BothGroups$S1_Main_Identity_AMDRES00)
# table(df0_BothGroups$S2_Main_Identity_BMDRES00)
```

### Responder Education Level

Missing responder data for S1 is 7 and for S2 is 539

```
S1_Qualifications =
table(df0_BothGroups$S1_SES_EducationMain_AMDNVQ00) %>%
  .[-1] %>%
  `names<-`((c(paste0("NVQ Level ",1:5),"Other Qualification","No Qualifications"))) %>%
  rbind(., (./sum(.)) ) %>%
  rbind.data.frame(., cumsum(.[2,])) %>%
  `rownames<-`(c("freq", "prob", "cumprob"))


S2_Qualifications =
table(df0_BothGroups$S2_Main_Education_BMDNVQ00) %>%
  .[-1] %>%
  `names<-`((c(paste0("NVQ Level ",1:5),"Other Qualification","No Qualifications"))) %>%
  rbind(., (./sum(.)) ) %>%
  rbind.data.frame(., cumsum(.[2,])) %>%
  `rownames<-`(c("freq", "prob", "cumprob"))

S1_Qualifications
```

```
##          NVQ Level 1  NVQ Level 2  NVQ Level 3  NVQ Level 4  NVQ Level 5
## freq    791.00000000 2913.0000000 1500.0000000 3235.0000000 434.00000000
## prob      0.07628508    0.2809336    0.1446620    0.3119877   0.04185553
## cumprob   0.07628508    0.3572186    0.5018806    0.8138683   0.85572379
##         Other Qualification No Qualifications
## freq           266.00000000      1230.0000000
## prob             0.02565339         0.1186228
## cumprob          0.88137718         1.0000000
```

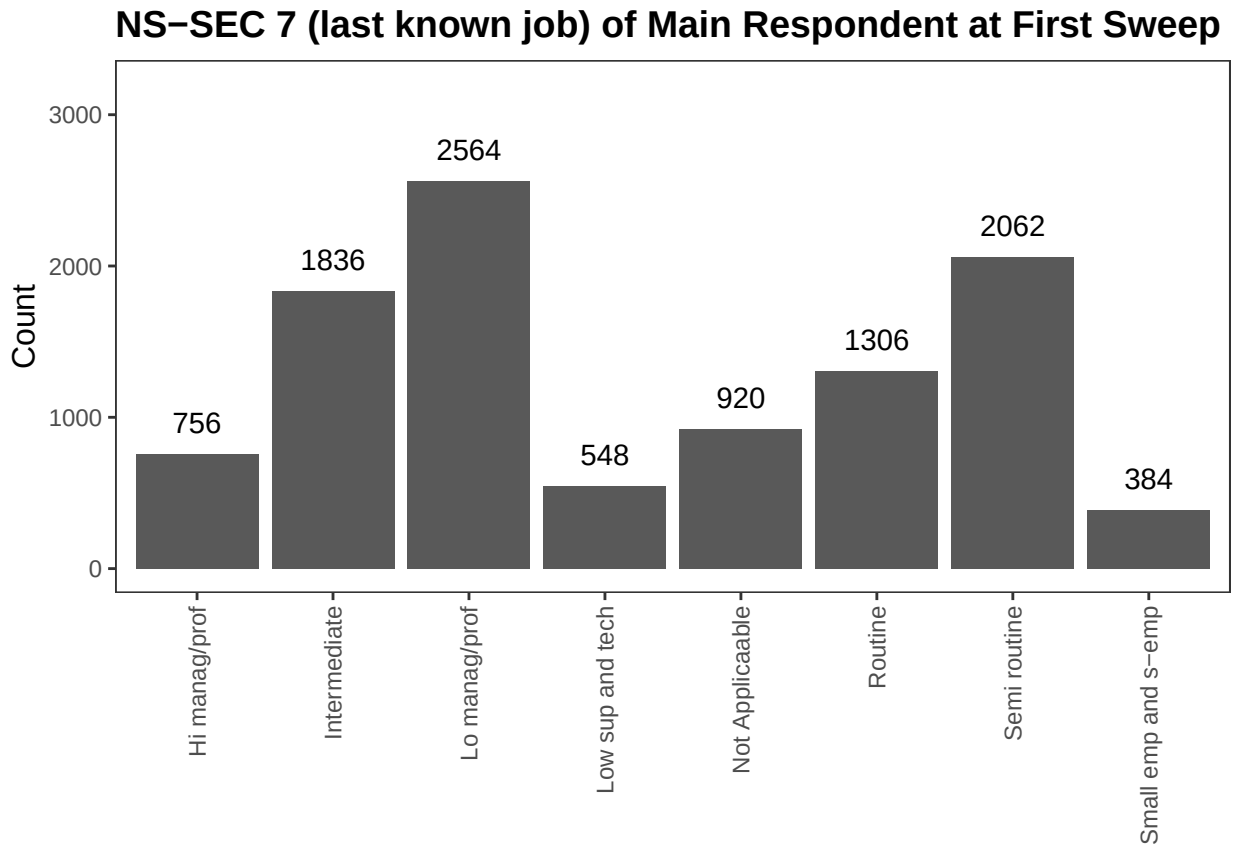
```
S2_Qualifications
```

```
##          NVQ Level 1  NVQ Level 2  NVQ Level 3  NVQ Level 4  NVQ Level 5
## freq    712.00000000 2666.0000000 1468.0000000 3203.0000000 458.00000000
## prob      0.07237979    0.2710176    0.1492325    0.3256074   0.04655891
## cumprob   0.07237979    0.3433974    0.4926299    0.8182373   0.86479618
##         Other Qualification No Qualifications
## freq           279.00000000      1051.0000000
## prob             0.02836231         0.1068415
## cumprob          0.89315848         1.0000000
```

**Employment Level**

```
S1Der = data.table::fread("Data/MCS Data/Raw Data/UKDA-4683-tab MCS1/tab/mcs1_derived_variables.tab", s
S1Der = S1Der[match(df0_BothGroups$MCSID, S1Der$MCSID),]

S1Der %>%
  dplyr::mutate(S1_Main_Employ = dplyr::recode(AMD07C00,
                                               `1` = "Hi manag/prof",
                                               `2` = "Lo manag/prof",
                                               `3` = "Intermediate",
                                               `4` = "Small emp and s-emp",
                                               `5` = "Low sup and tech",
                                               `6` = "Semi routine",
                                               `7` = "Routine",
                                               `-1` = "Not Applicaable",
                                               )) %>%
  ggplot(.,aes(x=S1_Main_Employ)) +
  geom_bar() +
  geom_text(stat='count', aes(label=..count..), vjust=-1) +
  jtools::theme_apa() +
  labs(title="NS-SEC 7 (last known job) of Main Respondent at First Sweep", x=NULL, y="Count") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ylim(c(0000,3200))
```




```
S1Der$AMD07C00 %>%
  table() %>%
  rbind(., (./sum(.)) ) %>%
  rbind.data.frame(., cumsum(.[2,])) %>%
  `rownames<-`(c("freq", "prob", "cumprob")) %>%
  `colnames<-`(c("Not Applicaable","Hi manag/prof","Lo manag/prof","Intermediate","Small emp and s-emp"
  format(.,nsmall=5,digits=0)
```

```
##         Not Applicaable Hi manag/prof Lo manag/prof Intermediate
## freq          920.00000     756.00000    2564.00000   1836.00000
## prob            0.08867       0.07286       0.24711      0.17695
## cumprob         0.08867       0.16153       0.40864      0.58558
##         Small emp and s-emp Low sup and tech Semi routine    Routine
## freq              384.00000        548.00000   2062.00000 1306.00000
## prob                0.03701          0.05281      0.19873    0.12587
## cumprob             0.62259          0.67540      0.87413    1.00000
```

## Proportion of productive families in S5 & S6 excluded from analyses