# Merging Cancer Incidence and Mutation Status

*Anne Pajon and Mark Dunning*

*13 October 2015*

## Contents



## 1 Introduction

From James:-

> *I need to retrieve data from public repositories and do some simple manipulation..... Retrieve data on cancer incidence statistics from UK/EU, USA, Aisa (I would start with CRUK but am not sure of other data sources yet) Retrieve data on mutation incidence across multiple cancer types (I have used CBioPortal, but would like to query ICGC) Retrieve data on companion diagnostics and drug therapies Combine these data to identify the number of patients who would be eligible for a specific treatment (e.g. Breast Cancer patients with HER2 amplification would be given Herceptin) Combine these data to identify the number of patients who could be eligible for a specific treatment (e.g. any cancer patients with HER2 amplification could be given Herceptin) Compare would/could population sizes*

## 2 Obtaining Cancer Incidence Rates

A url to official statistics on the Cancer Research Uk website was given. The file is in Excel (`.xls`) format, which is not one of the usual file types that R can handle (i.e. `.txt`, `.csv`, `.tsv`). However, the `gdata` package can read `.xls` files that are locally-stored, and even files that are available online. To read the file, we can give the url to the `read.xls` function. As with all R packages, we first have to install `gdata` if we do not have it.

```
install.packages("gdata")
```

```
library(gdata)
url <- "http://www.cancerresearchuk.org/sites/default/files/cstream-node/inc_20common_mf.xls"
crukStats <- read.xls(url)
head(crukStats)
```

```
##   The.20.Most.Common.Cancers.in.2011      X    X.1     X.2 X.3
## 1            Number of New Cases, UK                         NA
## 2                       Cancer Site   Male Female Persons  NA
## 3                     Breast (C50)    349 49,936          NA
## 4                   Lung (C33-C34) 23,770 19,693  43,463  NA
## 5                   Prostate (C61) 41,736                   NA
## 6                  Bowel (C18-C20) 23,171 18,410  41,581  NA
```

We observe that the data frame created is not ideal, as the data we want to analyse actually start in row 3 of the table. The functions used to read data into R (`read.csv`, `read.delim`, etc) are special cases of `read.table`, which has a

plethora of options that can be specified. See `?read.table` for full details. Particularly useful in this case is the ability to *skip* lines from the file. Another change we make is to make sure that missing data in the table are represented appropriately. In the original table, missing values are represented by a blank cell. However, for numerical analysis it is often to use the special R value `NA`. The `na.strings` argument to `read.xls` will ensure that blank cells get converted to `NA`.

```
crukStats <- read.xls(url,skip=2,na.strings = "",stringsAsFactors=FALSE)
head(crukStats)
```

```
##                       Cancer.Site   Male Female Persons  X
## 1                   Breast (C50)    349 49,936    <NA> NA
## 2                 Lung (C33-C34) 23,770 19,693  43,463 NA
## 3                 Prostate (C61) 41,736   <NA>    <NA> NA
## 4                Bowel (C18-C20) 23,171 18,410  41,581 NA
## 5       Malignant Melanoma (C43)  6,495  6,853  13,348 NA
## 6 Non-Hodgkin Lymphoma (C82-C85)  6,926  5,857  12,783 NA
```

At this point, you can enter `View(crukStats)` in RStudio and be able to view the data.

## 2.1 Data Manipulation in base R

If we are only interested in the first 20 rows (the 20 most-common cancers) we can *subset* the data frame to contain rows 1 to 20 and all columns. Subsetting in R is done using the square brackets `[]` with a row and column index separted by a comma `,`. i.e. ***`[row,column]`***. Where `row` and `column` are both vectors. If we omit the column index all rows will be returned; and vice-versa. So the command to subset the first 20 rows and all columns is as follows:-

```
crukStats <- crukStats[1:20,]
```

Both the male and female counts are not ameanable for analysis, as they have comma's within. Thus, R will treat these as characters (text) and not be able to perform numeric operations. We can replace the comma using the `gsub` function. This function will replace all occurences of a specified character with a different string. To access a particular column in a data frame, we *can* use it's numeric index as we saw above. However, it is a better practice to refer to the column by name. This is done by using the `$` operator. Typing the following and pressing ***TAB*** should bring up a list of all the columns (*variables*) that are currently in the `crukStats` object. Thus, you can easily select the one you are interested in rather than typing the full name by-hand and running the risk of making a mistake.

```
crukStats$
```

We want to replace a comma with a blank string in the vector `crukStats$Male` and `crukStats$Female`. However, the result of doing the `gsub` will be a character vector. Therefore we need to convert to a numeric value using `as.numeric`.

```
crukStats$Male <- as.numeric(gsub(",", "", crukStats$Male))
crukStats$Female <- as.numeric(gsub(",", "", crukStats$Female))
head(crukStats)
```

```
##                       Cancer.Site  Male Female Persons  X
## 1                   Breast (C50)   349  49936    <NA> NA
## 2                 Lung (C33-C34) 23770  19693  43,463 NA
## 3                 Prostate (C61) 41736     NA    <NA> NA
## 4                Bowel (C18-C20) 23171  18410  41,581 NA
## 5       Malignant Melanoma (C43)  6495   6853  13,348 NA
## 6 Non-Hodgkin Lymphoma (C82-C85)  6926   5857  12,783 NA
```

If we now try and add the Male and Female counts together we can use the + operator. Addition in R (and indeed other numeric operations, -, *, / etc) will be applied to each item separately.

```
crukStats$Male + crukStats$Female
```

```
##  [1] 50285 43463    NA 41581 13348 12783 10399 10144  9365  8773  8616
## [12]    NA  8332    NA  7089  6767  4792  4348    NA  2727
```

As you see, there is a problem with the results. For Sites such as *Prostate* where we have Male, but not Female, cases, the Total that is calculated is `NA`. Obviously this is not ideal as we would like the Total to be just the Male cases. The solution would be to use `0` to represent missing values rather than `NA`.

But how do we identify the NAs? In R, there are a series of functions that can test whether a specified value, or vector of values, is of a particular *data type*. For example, `is.numeric(10)` returns the value TRUE. On the other hand, `is.numeric("ten")` returns FALSE. The function `is.na` can be used to see where `NA` values occur in an object; which could be a data frame.

```
is.na(crukStats)
```

```
##   Cancer.Site  Male Female Persons    X
## 1       FALSE FALSE  FALSE    TRUE TRUE
## 2       FALSE FALSE  FALSE   FALSE TRUE
## 3       FALSE FALSE   TRUE    TRUE TRUE
## 4       FALSE FALSE  FALSE   FALSE TRUE
## 5       FALSE FALSE  FALSE   FALSE TRUE
## 6       FALSE FALSE  FALSE   FALSE TRUE
```

We can *re-assign* particular values in a data frame using the assignment operator <- and specifying a replacement value. To re-assign all the `NA` values to `0` we can use:

```
crukStats[is.na(crukStats)] <- 0
```

Which has the desired effect :-

```
##                      Cancer.Site  Male Female Persons X
## 1                  Breast (C50)   349  49936       0 0
## 2                Lung (C33-C34) 23770  19693  43,463 0
## 3                Prostate (C61) 41736      0       0 0
## 4               Bowel (C18-C20) 23171  18410  41,581 0
## 5       Malignant Melanoma (C43)  6495   6853  13,348 0
## 6 Non-Hodgkin Lymphoma (C82-C85)  6926   5857  12,783 0
```

The total can now be calculated and added as a new column in the data frame.

```
crukStats$Total <- crukStats$Male + crukStats$Female
head(crukStats)
```

```
##                      Cancer.Site  Male Female Persons X Total
## 1                  Breast (C50)   349  49936       0 0 50285
## 2                Lung (C33-C34) 23770  19693  43,463 0 43463
## 3                Prostate (C61) 41736      0       0 0 41736
## 4               Bowel (C18-C20) 23171  18410  41,581 0 41581
## 5       Malignant Melanoma (C43)  6495   6853  13,348 0 13348
## 6 Non-Hodgkin Lymphoma (C82-C85)  6926   5857  12,783 0 12783
```

Another problem we could tackle is the naming of the Cance Sites in the first column. Eventually we want to merge these data with another table. To have the best chance of being able to do this effectively, we need to ensure consistent naming between the two tables. A good start is to trim the names with have in the first column of our `crukStats` data frame to include just the organ name without the labels in brackets.

The stringr package has lots of useful functions for manipulating, trimming, finding (etc...) strings in R. For this example, we will only use the `str_split_fixed` function, but please do check out the documentation for stringr. You are sure to need some of the other functions at a later point!

The purpose of `str_split_fixed` is to *split* a particular piece of text into a pre-defined number of pieces. A particular *string* is used to define where each string is split. In our example, we can split each entry in the `Cancer.Site` column

using the blank space " " character. The result is a data frame with two columns; the first column being the text that occurs to the left of the first space, and the second column being everything to the right of the space character. i.e. `Breast (C50)` gets split into two elements; `Breast` and `(C50)`. We can then re-assign the `Cancer.Site` to be the first column in the output of `str_split_fixed`.

```
library(stringr)
tab <- str_split_fixed(crukStats$Cancer.Site, " ",2)
head(tab)
```

```
##      [,1]          [,2]
## [1,] "Breast"      "(C50)"
## [2,] "Lung"        "(C33-C34)"
## [3,] "Prostate"    "(C61)"
## [4,] "Bowel"       "(C18-C20)"
## [5,] "Malignant"   "Melanoma (C43)"
## [6,] "Non-Hodgkin" "Lymphoma (C82-C85)"
```

```
crukStats$Cancer.Site <- tab[,1]
head(crukStats)
```

```
##   Cancer.Site  Male Female Persons X Total
## 1      Breast   349  49936       0 0 50285
## 2        Lung 23770  19693  43,463 0 43463
## 3    Prostate 41736      0       0 0 41736
## 4       Bowel 23171  18410  41,581 0 41581
## 5   Malignant  6495   6853  13,348 0 13348
## 6 Non-Hodgkin  6926   5857  12,783 0 12783
```

For the analysis, we also want to compute the percentage that each cancer contributes to overall cancer incidences. We can do this in two stages, the first of which is to sum-up the totals for all individual cancer using the `sum` function. We can then divide all the individual cases by the overall total. We can do this in one step.

```
totalCases <- sum(crukStats$Total)
crukStats$Percentage <- 100*(crukStats$Total / totalCases)
head(crukStats)
```

```
##   Cancer.Site  Male Female Persons X Total Percentage
## 1      Breast   349  49936       0 0 50285  16.584598
## 2        Lung 23770  19693  43,463 0 43463  14.334621
## 3    Prostate 41736      0       0 0 41736  13.765035
## 4       Bowel 23171  18410  41,581 0 41581  13.713914
## 5   Malignant  6495   6853  13,348 0 13348   4.402331
## 6 Non-Hodgkin  6926   5857  12,783 0 12783   4.215987
```

Finally, we can notice that the `Percentage` and `X` columns are not that useful in the data frame anymore. Removing a column (or row) is done by using a - sign in front of the column index.

whereas:-

```
crukStats[,c(4,5)]
```

```
##   Persons X
## 1       0 0
## 2  43,463 0
## 3       0 0
## 4  41,581 0
## 5  13,348 0
## 6  12,783 0
```

selects the 4th and 5th columns....

```
crukStats[,-c(4,5)]
```

```
##   Cancer.Site  Male Female Total Percentage
## 1      Breast   349  49936 50285  16.584598
## 2        Lung 23770  19693 43463  14.334621
## 3    Prostate 41736      0 41736  13.765035
## 4       Bowel 23171  18410 41581  13.713914
## 5   Malignant  6495   6853 13348   4.402331
## 6 Non-Hodgkin  6926   5857 12783   4.215987
```

will remove them. Or rather it prints what that the data frame looks like without the 4th and 5th columns. To remove permanently we need to create a new object or re-assign an existing one.

```
crukStats <- crukStats[,-c(4,5)]
head(crukStats)
```

```
##   Cancer.Site  Male Female Total Percentage
## 1      Breast   349  49936 50285  16.584598
## 2        Lung 23770  19693 43463  14.334621
## 3    Prostate 41736      0 41736  13.765035
## 4       Bowel 23171  18410 41581  13.713914
## 5   Malignant  6495   6853 13348   4.402331
## 6 Non-Hodgkin  6926   5857 12783   4.215987
```

If we wish we could write this data frame to a file. Writing an `xls` file is not supported. However, we can write tab-delimited and comma-separated files. The generic function for writing a data frame is `write.table`. We have control over what column separator is used, the default being a space. To use tab we can specify `\t`.

```
write.table(crukStats, file="cancerStatsCleaned.txt",sep="\t")
```

## 2.2 Data Manipulation in dplyr (advanced....)

```
crukStats <- read.xls(url,skip=2,na.strings = "")
head(crukStats)
```

```
##                      Cancer.Site   Male Female Persons  X
## 1                  Breast (C50)    349 49,936    <NA> NA
## 2                Lung (C33-C34) 23,770 19,693  43,463 NA
## 3                Prostate (C61) 41,736   <NA>    <NA> NA
## 4               Bowel (C18-C20) 23,171 18,410  41,581 NA
## 5       Malignant Melanoma (C43)  6,495  6,853  13,348 NA
## 6 Non-Hodgkin Lymphoma (C82-C85)  6,926  5,857  12,783 NA
```

```
library(tidyr)
library(dplyr)
crukStats <- tbl_df(crukStats)
crukStats <- mutate(crukStats, Cancer.Site= str_split_fixed(Cancer.Site, " ",2)[,1])
```

```
crukStats <- mutate(crukStats, Male = as.numeric(gsub(",", "",Male))) %>%
              mutate(Female = as.numeric(gsub(",", "",Female)))
crukStats <- mutate(crukStats, Male = ifelse(is.na(Male),0,Male)) %>%
              mutate(Female = ifelse(is.na(Female),0,Female))  %>%
              mutate(Total = Male + Female)
crukStats <- crukStats[1:20,]
```

```
crukStats <- select(crukStats, -c(Persons,X))
crukStats
```

```
## Source: local data frame [20 x 4]
##
##    Cancer.Site  Male Female Total
##          (chr) (dbl)  (dbl) (dbl)
## 1       Breast   349  49936 50285
## 2         Lung 23770  19693 43463
## 3     Prostate 41736      0 41736
## 4        Bowel 23171  18410 41581
## 5    Malignant  6495   6853 13348
## 6  Non-Hodgkin  6926   5857 12783
## 7      Bladder  7452   2947 10399
## 8       Kidney  6257   3887 10144
## 9       Brain,  4650   4715  9365
## 10    Pancreas  4328   4445  8773
## 11   Leukaemia  5014   3602  8616
## 12      Uterus     0   8475  8475
## 13  Oesophagus  5582   2750  8332
## 14       Ovary     0   7116  7116
## 15     Stomach  4615   2474  7089
## 16        Oral  4510   2257  6767
## 17     Myeloma  2660   2132  4792
## 18       Liver  2776   1572  4348
## 19      Cervix     0   3064  3064
## 20     Thyroid   769   1958  2727
```

```
crukStats <- mutate(crukStats,Percentage = 100*(Total / sum(Total)))
crukStats
```

```
## Source: local data frame [20 x 5]
##
##    Cancer.Site  Male Female Total Percentage
##          (chr) (dbl)  (dbl) (dbl)      (dbl)
## 1       Breast   349  49936 50285 16.5845984
## 2         Lung 23770  19693 43463 14.3346207
## 3     Prostate 41736      0 41736 13.7650353
## 4        Bowel 23171  18410 41581 13.7139144
## 5    Malignant  6495   6853 13348  4.4023311
## 6  Non-Hodgkin  6926   5857 12783  4.2159873
## 7      Bladder  7452   2947 10399  3.4297154
## 8       Kidney  6257   3887 10144  3.3456133
## 9       Brain,  4650   4715  9365  3.0886898
## 10    Pancreas  4328   4445  8773  2.8934410
## 11   Leukaemia  5014   3602  8616  2.8416605
## 12      Uterus     0   8475  8475  2.7951570
## 13  Oesophagus  5582   2750  8332  2.7479939
## 14       Ovary     0   7116  7116  2.3469425
## 15     Stomach  4615   2474  7089  2.3380376
## 16        Oral  4510   2257  6767  2.2318381
## 17     Myeloma  2660   2132  4792  1.5804593
## 18       Liver  2776   1572  4348  1.4340228
```

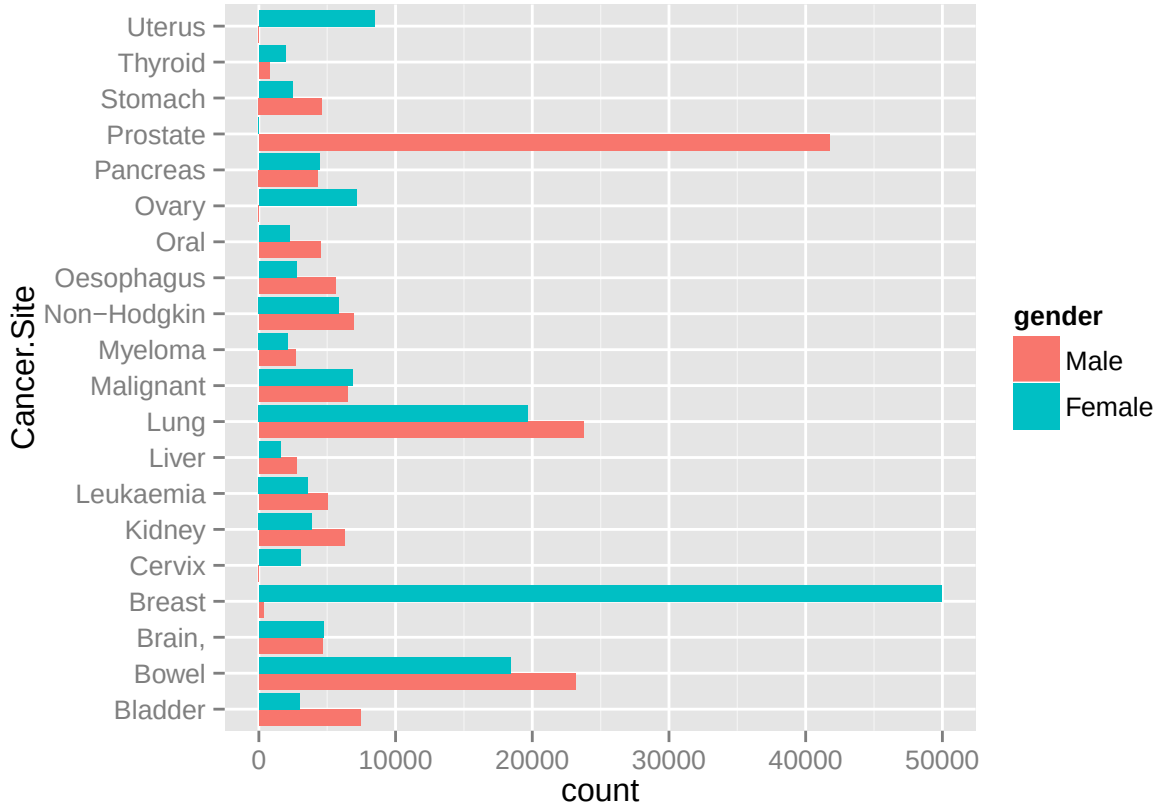
```
## 19      Cervix     0   3064  3064  1.0105441
## 20     Thyroid   769   1958  2727  0.8993974
```

```
library(ggplot2)

analysisDf <- gather(crukStats, key = gender, value=count, Male:Female)
analysisDf
```

```
## Source: local data frame [40 x 5]
##
##    Cancer.Site Total Percentage gender count
##          (chr) (dbl)      (dbl) (fctr) (dbl)
## 1       Breast 50285  16.584598   Male   349
## 2         Lung 43463  14.334621   Male 23770
## 3     Prostate 41736  13.765035   Male 41736
## 4        Bowel 41581  13.713914   Male 23171
## 5    Malignant 13348   4.402331   Male  6495
## 6  Non-Hodgkin 12783   4.215987   Male  6926
## 7      Bladder 10399   3.429715   Male  7452
## 8       Kidney 10144   3.345613   Male  6257
## 9       Brain,  9365   3.088690   Male  4650
## 10    Pancreas  8773   2.893441   Male  4328
## ..         ...   ...        ...    ...   ...
```

```
ggplot(analysisDf, aes(x = Cancer.Site,y=count,fill=gender)) +
  geom_bar(stat="identity",position = "dodge") + coord_flip()
```



# 3 cBioPortal data

```
erbb2 <- tbl_df(read.csv("erbb2_amplifications.csv"))
erbb2
```

```
## Source: local data frame [62 x 4]
##
##             STUDY_ABBREVIATION
##                          (fctr)
## 1                Stomach (TCGA)
## 2              Breast (TCGA pub)
## 3             Stomach (TCGA pub)
## 4          Breast (TCGA pub2015)
## 5                  Breast (TCGA)
## 6                Pancreas (UTSW)
## 7              Uterine CS (TCGA)
## 8     Breast (BCCRC Xenograft)
## 9                Uterine (TCGA)
## 10 CCLE (Novartis/Broad 2012)
## ..                           ...
## Variables not shown: STUDY_NAME (fctr), NUM_OF_CASES_ALTERED (int),
##   PERCENT_CASES_ALTERED (fctr)
```

```
erbb2 <- mutate(erbb2, STUDY_ABBREVIATION = str_split_fixed(STUDY_ABBREVIATION, " ", 2)[,1]) %>%
        mutate(PERCENT_CASES_ALTERED = as.numeric(gsub("%","", PERCENT_CASES_ALTERED))/100) %>%
        rename(Cancer.Site= STUDY_ABBREVIATION) %>%
        select(-STUDY_NAME)
erbb2
```

```
## Source: local data frame [62 x 3]
##
##    Cancer.Site NUM_OF_CASES_ALTERED PERCENT_CASES_ALTERED
##          (chr)                (int)                 (dbl)
## 1      Stomach                   60                 0.136
## 2       Breast                  101                 0.130
## 3      Stomach                   38                 0.130
## 4       Breast                  135                 0.125
## 5       Breast                  135                 0.125
## 6     Pancreas                   12                 0.110
## 7      Uterine                    6                 0.107
## 8       Breast                    8                 0.069
## 9      Uterine                   37                 0.069
## 10        CCLE                   65                 0.065
## ..         ...                  ...                   ...
```

```
combinedDf <- inner_join(crukStats, erbb2)
```

```
## Joining by: "Cancer.Site"
```

```
mutate(combinedDf, Cases.Amplified = Total*PERCENT_CASES_ALTERED) %>%
             select(Cancer.Site, Total,PERCENT_CASES_ALTERED,Cases.Amplified)
```

```
## Source: local data frame [29 x 4]
##
##    Cancer.Site Total PERCENT_CASES_ALTERED Cases.Amplified
##          (chr) (dbl)                 (dbl)           (dbl)
```

```
## 1       Breast 50285               0.130        6537.050
## 2       Breast 50285               0.125        6285.625
## 3       Breast 50285               0.125        6285.625
## 4       Breast 50285               0.069        3469.665
## 5         Lung 43463               0.030        1303.890
## 6         Lung 43463               0.026        1130.038
## 7         Lung 43463               0.022         956.186
## 8         Lung 43463               0.021         912.723
## 9         Lung 43463               0.016         695.408
## 10    Prostate 41736               0.037        1544.232
## ..         ...   ...                 ...             ...
```