# Assignment 2

Chang An Le Harry Jr

2/9/2022

```
library(jsonlite)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.7     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.1

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

```
#install.packages("nycflights13")
library(nycflights13)
```

## Question 1 - Retrenched Employees by Industry

1. Use the Data API link on the webpage to find the resource id of this data. Download the full dataset, and name it as r_emp.

```
root_url = "https://data.gov.sg"
url1 = paste(root_url,
             "/api/action/datastore_search?",
             "resource_id=3d180571-81d3-4834-a759-8374806b731e",
             sep = "")
r_emp_json = fromJSON(url1)
str(r_emp_json)
```

```
## List of 3
##  $ help   : chr "https://data.gov.sg/api/3/action/help_show?name=datastore_search"
##  $ success: logi TRUE
##  $ result :List of 5
##   ..$ resource_id: chr "3d180571-81d3-4834-a759-8374806b731e"
##   ..$ fields     :'data.frame':  6 obs. of  2 variables:
##   .. ..$ type: chr [1:6] "int4" "text" "text" "text" ...
```

```
##   .. ..$ id  : chr [1:6] "_id" "quarter" "industry1" "retrench" ...
##   ..$ records    :'data.frame':  100 obs. of  6 variables:
##   .. ..$ _id                   : int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ retrench              : chr [1:100] "6170" "560" "2100" "-" ...
##   .. ..$ retrench_term_contract: chr [1:100] "1060" "480" "160" "-" ...
##   .. ..$ quarter               : chr [1:100] "1998-Q1" "1998-Q1" "1998-Q1" "1998-Q1" ...
##   .. ..$ retrench_permanent    : chr [1:100] "5110" "90" "1940" "-" ...
##   .. ..$ industry1             : chr [1:100] "manufacturing" "construction" "services" "others" ...
##   ..$ _links     :List of 2
##   .. ..$ start: chr "/api/action/datastore_search?resource_id=3d180571-81d3-4834-a759-8374806b731e"
##   .. ..$ next : chr "/api/action/datastore_search?offset=100&resource_id=3d180571-81d3-4834-a759-837
##   ..$ total      : int 400
```

```
r_emp_json$result$`_links`
```

```
## $start
## [1] "/api/action/datastore_search?resource_id=3d180571-81d3-4834-a759-8374806b731e"
##
## $`next`
## [1] "/api/action/datastore_search?offset=100&resource_id=3d180571-81d3-4834-a759-8374806b731e"
```

```
r_emp = r_emp_json$result$records
total_records = r_emp_json$result$total

times = floor(total_records/100)
for (i in 1:times) {
  url = paste(root_url,
              "/api/action/datastore_search?",
              "offset=", i,
              "00&",
              "resource_id=3d180571-81d3-4834-a759-8374806b731e",
              sep = "")
  r_emp_json = fromJSON(url)
  r_emp = rbind(r_emp, r_emp_json$result$records)
}
dim(r_emp)
```

```
## [1] 400   6
```

With 380 rows and 6 columns, all we need to do is to remove the id column (which is redundant)

```
r_emp = r_emp[,-(1)]
head(r_emp)
```

```
##   retrench retrench_term_contract quarter retrench_permanent     industry1
## 1     6170                  1060 1998-Q1               5110 manufacturing
## 2      560                   480 1998-Q1                 90  construction
## 3     2100                   160 1998-Q1               1940      services
## 4        -                     - 1998-Q1                  -        others
## 5     5010                   120 1998-Q2               4890 manufacturing
## 6      600                   170 1998-Q2                430  construction
```

```
dim(r_emp) #to check
```

```
## [1] 400   5
```

2. Data manipulation tasks:
   a. Convert retrench, retrench_term_contract, and retrench_permanent to numeric;

Before conversion:

```
sapply(r_emp, class)
```

```
##               retrench retrench_term_contract               quarter
##            "character"            "character"           "character"
##     retrench_permanent              industry1
##            "character"            "character"
```

After conversion:

```
r_emp$retrench = as.numeric(r_emp$retrench)
```

```
## Warning: NAs introduced by coercion
```

```
r_emp$retrench_term_contract = as.numeric(r_emp$retrench_term_contract)
```

```
## Warning: NAs introduced by coercion
```

```
r_emp$retrench_permanent = as.numeric(r_emp$retrench_permanent)
```

```
## Warning: NAs introduced by coercion
```

```
sapply(r_emp,class)
```

```
##               retrench retrench_term_contract               quarter
##              "numeric"              "numeric"           "character"
##     retrench_permanent              industry1
##              "numeric"            "character"
```

b) Convert industry1 to factor;

```
r_emp$industry1 = as.factor(r_emp$industry1)

sapply(r_emp,class) #to check
```

```
##               retrench retrench_term_contract               quarter
##              "numeric"              "numeric"           "character"
##     retrench_permanent              industry1
##              "numeric"               "factor"
```

c) Compute the summary statistics of all variables in the r_emp data.

```
summary(r_emp)
```

```
##     retrench    retrench_term_contract   quarter          retrench_permanent
##  Min.   :  10   Min.   :  10           Length:400         Min.   :  10.0
##  1st Qu.: 160   1st Qu.:  40           Class :character   1st Qu.: 110.0
##  Median : 770   Median :  85           Mode  :character   Median : 700.0
##  Mean   :1086   Mean   : 133                              Mean   : 993.8
##  3rd Qu.:1580   3rd Qu.: 160                              3rd Qu.:1530.0
##  Max.   :9100   Max.   :1430                              Max.   :7870.0
##  NA's   :54     NA's   :106                               NA's   :61
##          industry1
##  construction :100
##  manufacturing:100
##  others       :100
##  services     :100
##
##
##
```

3. Explore the dataset by yourself. Answer one question you find interesting about the data. Include the code you used, and summarize (in words) what you found.

Before exploring the dataset, I have chosen to replace all the "NA" values within the dataset to 0. Based on the website, such "NA" values indicate that "Data is negligible or not significant". Furthermore, since "Data are rounded to the nearest 10", it would be justifiable to convert the "NA" values to 0 in my opinion.

```
r_emp[is.na(r_emp)] = 0 #replace all "NA" with 0
r_emp #check if above code was successful
```

```
##     retrench retrench_term_contract quarter retrench_permanent     industry1
## 1       6170                   1060 1998-Q1               5110 manufacturing
## 2        560                    480 1998-Q1                 90  construction
## 3       2100                    160 1998-Q1               1940      services
## 4          0                      0 1998-Q1                  0        others
## 5       5010                    120 1998-Q2               4890 manufacturing
## 6        600                    170 1998-Q2                430  construction
## 7       2130                    150 1998-Q2               1990      services
## 8          0                      0 1998-Q2                  0        others
## 9       4330                    290 1998-Q3               4040 manufacturing
## 10       600                    380 1998-Q3                220  construction
## 11      2680                    310 1998-Q3               2370      services
## 12        10                     10 1998-Q3                  0        others
## 13      5190                    300 1998-Q4               4890 manufacturing
## 14       680                    210 1998-Q4                470  construction
## 15      2720                     70 1998-Q4               2650      services
## 16        20                     20 1998-Q4                  0        others
## 17      1220                     70 1999-Q1               1160 manufacturing
## 18       420                    180 1999-Q1                240  construction
## 19      2060                     50 1999-Q1               2010      services
## 20         0                      0 1999-Q1                  0        others
## 21      1980                    220 1999-Q2               1760 manufacturing
## 22       110                     30 1999-Q2                 70  construction
```

```
## 23      1530      20 1999-Q2      1510      services
## 24         0       0 1999-Q2         0        others
## 25      1870      80 1999-Q3      1790 manufacturing
## 26       110      40 1999-Q3        70  construction
## 27      1580      50 1999-Q3      1530      services
## 28         0       0 1999-Q3         0        others
## 29      3300      10 1999-Q4      3280 manufacturing
## 30       190      60 1999-Q4       130  construction
## 31      1150      90 1999-Q4      1050      services
## 32        10       0 1999-Q4         0        others
## 33      5120      80 2000-Q1      5040 manufacturing
## 34       140      60 2000-Q1        90  construction
## 35       950      30 2000-Q1       920      services
## 36         0       0 2000-Q1         0        others
## 37       590       0 2000-Q2       590 manufacturing
## 38        60      10 2000-Q2        60  construction
## 39      1230      10 2000-Q2      1210      services
## 40         0       0 2000-Q2         0        others
## 41       810      10 2000-Q3       790 manufacturing
## 42        60      20 2000-Q3        40  construction
## 43       950      30 2000-Q3       920      services
## 44         0       0 2000-Q3         0        others
## 45       980       0 2000-Q4       980 manufacturing
## 46       110      40 2000-Q4        70  construction
## 47       950      40 2000-Q4       910      services
## 48         0       0 2000-Q4         0        others
## 49      2020     250 2001-Q1      1770 manufacturing
## 50       160       0 2001-Q1       150  construction
## 51      1340      20 2001-Q1      1320      services
## 52         0       0 2001-Q1         0        others
## 53      4120     520 2001-Q2      3610 manufacturing
## 54       170      70 2001-Q2       100  construction
## 55      1970      60 2001-Q2      1900      services
## 56        20       0 2001-Q2        20        others
## 57      5480     340 2001-Q3      5140 manufacturing
## 58       260      90 2001-Q3       170  construction
## 59      3050      90 2001-Q3      2960      services
## 60       100       0 2001-Q3       100        others
## 61      4060     140 2001-Q4      3920 manufacturing
## 62       320      50 2001-Q4       270  construction
## 63      4430     110 2001-Q4      4320      services
## 64        80       0 2001-Q4        80        others
## 65      2000      40 2002-Q1      1960 manufacturing
## 66       310      70 2002-Q1       240  construction
## 67      2800     140 2002-Q1      2660      services
## 68         0       0 2002-Q1         0        others
## 69      1750     390 2002-Q2      1360 manufacturing
## 70       250      40 2002-Q2       210  construction
## 71      2530      60 2002-Q2      2470      services
## 72        50       0 2002-Q2        50        others
## 73      2260      20 2002-Q3      2240 manufacturing
## 74       140      20 2002-Q3       120  construction
## 75      1850      40 2002-Q3      1810      services
## 76       110     100 2002-Q3        20        others
```

```
## 77      3650   60 2002-Q4  3590 manufacturing
## 78       300   30 2002-Q4   270  construction
## 79      2110   30 2002-Q4  2080      services
## 80        10    0 2002-Q4    10        others
## 81      1950   40 2003-Q1  1910 manufacturing
## 82       210   60 2003-Q1   160  construction
## 83      2440  100 2003-Q1  2340      services
## 84         0    0 2003-Q1     0        others
## 85      2690  160 2003-Q2  2530 manufacturing
## 86       210   40 2003-Q2   170  construction
## 87      2610  170 2003-Q2  2430      services
## 88        10    0 2003-Q2    10        others
## 89      1620   40 2003-Q3  1590 manufacturing
## 90       200   20 2003-Q3   180  construction
## 91      2270  190 2003-Q3  2090      services
## 92       220    0 2003-Q3   210        others
## 93      1220   20 2003-Q4  1210 manufacturing
## 94       120    0 2003-Q4   120  construction
## 95      1450   30 2003-Q4  1420      services
## 96        40    0 2003-Q4    40        others
## 97      1560   10 2004-Q1  1550 manufacturing
## 98        70   10 2004-Q1    70  construction
## 99      1240   30 2004-Q1  1210      services
## 100      140    0 2004-Q1   140        others
## 101      800   10 2004-Q2   790 manufacturing
## 102      200   10 2004-Q2   190  construction
## 103     1110   40 2004-Q2  1070      services
## 104       10    0 2004-Q2    10        others
## 105      810   40 2004-Q3   770 manufacturing
## 106       60    0 2004-Q3    60  construction
## 107     1140   80 2004-Q3  1060      services
## 108       80    0 2004-Q3    80        others
## 109     1540  160 2004-Q4  1380 manufacturing
## 110       80   20 2004-Q4    60  construction
## 111     1730   40 2004-Q4  1690      services
## 112       80    0 2004-Q4    70        others
## 113     1270   10 2005-Q1  1260 manufacturing
## 114      140   90 2005-Q1    50  construction
## 115      870   10 2005-Q1   850      services
## 116        0    0 2005-Q1     0        others
## 117     1270   20 2005-Q2  1250 manufacturing
## 118      160   90 2005-Q2    70  construction
## 119      820   30 2005-Q2   790      services
## 120        0    0 2005-Q2     0        others
## 121     2050  220 2005-Q3  1830 manufacturing
## 122      150  120 2005-Q3    30  construction
## 123     1030   90 2005-Q3   950      services
## 124        0    0 2005-Q3     0        others
## 125     2500   30 2005-Q4  2470 manufacturing
## 126       90   40 2005-Q4    60  construction
## 127      790  110 2005-Q4   670      services
## 128       10    0 2005-Q4    10        others
## 129     2600   10 2006-Q1  2590 manufacturing
## 130       60    0 2006-Q1    60  construction
```

```
##  131      980   20 2006-Q1      960      services
##  132       40    0 2006-Q1       40        others
##  133     1960   40 2006-Q2     1920 manufacturing
##  134      390   70 2006-Q2      320  construction
##  135     1080   60 2006-Q2     1020      services
##  136       10    0 2006-Q2       10        others
##  137     1910   50 2006-Q3     1860 manufacturing
##  138       30   20 2006-Q3       10  construction
##  139      630   30 2006-Q3      600      services
##  140        0    0 2006-Q3        0        others
##  141     2390   30 2006-Q4     2360 manufacturing
##  142       10   10 2006-Q4        0  construction
##  143      980  140 2006-Q4      840      services
##  144       20    0 2006-Q4       20        others
##  145     1440   60 2007-Q1     1380 manufacturing
##  146        0    0 2007-Q1        0  construction
##  147      650   80 2007-Q1      570      services
##  148       10    0 2007-Q1       10        others
##  149     1430   80 2007-Q2     1350 manufacturing
##  150       10    0 2007-Q2       10  construction
##  151      590   40 2007-Q2      560      services
##  152        0    0 2007-Q2        0        others
##  153     1310   50 2007-Q3     1250 manufacturing
##  154       50   40 2007-Q3       10  construction
##  155      980  420 2007-Q3      570      services
##  156       10   10 2007-Q3        0        others
##  157     1320   60 2007-Q4     1270 manufacturing
##  158       20   10 2007-Q4       10  construction
##  159      770   70 2007-Q4      700      services
##  160        0    0 2007-Q4        0        others
##  161     1810   90 2008-Q1     1720 manufacturing
##  162       10    0 2008-Q1       10  construction
##  163      590   50 2008-Q1      540      services
##  164       10    0 2008-Q1       10        others
##  165     1230   40 2008-Q2     1190 manufacturing
##  166       20    0 2008-Q2       20  construction
##  167      640   50 2008-Q2      590      services
##  168        0    0 2008-Q2        0        others
##  169     2230  550 2008-Q3     1680 manufacturing
##  170      130   50 2008-Q3       70  construction
##  171      820  230 2008-Q3      590      services
##  172        0    0 2008-Q3        0        others
##  173     5160 1350 2008-Q4     3800 manufacturing
##  174      390  240 2008-Q4      150  construction
##  175     3820  300 2008-Q4     3520      services
##  176       40   10 2008-Q4       30        others
##  177     9100 1220 2009-Q1     7870 manufacturing
##  178      350  250 2009-Q1      100  construction
##  179     3300  390 2009-Q1     2910      services
##  180       20    0 2009-Q1       10        others
##  181     2820  340 2009-Q2     2480 manufacturing
##  182      240   90 2009-Q2      150  construction
##  183     2910  370 2009-Q2     2540      services
##  184        0    0 2009-Q2        0        others
```

```
## 185      870      110 2009-Q3      760 manufacturing
## 186      140      100 2009-Q3       40  construction
## 187     1430      150 2009-Q3     1280      services
## 188       30        0 2009-Q3       30        others
## 189      860       50 2009-Q4      810 manufacturing
## 190      250      100 2009-Q4      160  construction
## 191     1080      100 2009-Q4      980      services
## 192       40        0 2009-Q4       40        others
## 193     1000      270 2010-Q1      740 manufacturing
## 194      340      240 2010-Q1      100  construction
## 195     1060       90 2010-Q1      970      services
## 196        0        0 2010-Q1        0        others
## 197     1140      100 2010-Q2     1040 manufacturing
## 198      150       60 2010-Q2       80  construction
## 199      990      110 2010-Q2      890      services
## 200        0        0 2010-Q2        0        others
## 201      970      280 2010-Q3      690 manufacturing
## 202      170       50 2010-Q3      120  construction
## 203      790      160 2010-Q3      630      services
## 204        0        0 2010-Q3        0        others
## 205     1370       70 2010-Q4     1310 manufacturing
## 206      690      450 2010-Q4      250  construction
## 207     1120      200 2010-Q4      930      services
## 208        0        0 2010-Q4        0        others
## 209     1440       70 2011-Q1     1370 manufacturing
## 210      310      220 2011-Q1       90  construction
## 211     1010       90 2011-Q1      910      services
## 212        0        0 2011-Q1        0        others
## 213      600       80 2011-Q2      520 manufacturing
## 214      410      220 2011-Q2      190  construction
## 215     1020      100 2011-Q2      920      services
## 216       10        0 2011-Q2       10        others
## 217      770      300 2011-Q3      470 manufacturing
## 218      100       70 2011-Q3       30  construction
## 219     1050      190 2011-Q3      870      services
## 220       40        0 2011-Q3       40        others
## 221     1660      110 2011-Q4     1550 manufacturing
## 222      240       80 2011-Q4      160  construction
## 223     1360      140 2011-Q4     1220      services
## 224        0        0 2011-Q4        0        others
## 225      750      150 2012-Q1      610 manufacturing
## 226      260       40 2012-Q1      220  construction
## 227     1580      130 2012-Q1     1450      services
## 228        0        0 2012-Q1        0        others
## 229      520       50 2012-Q2      480 manufacturing
## 230      180       50 2012-Q2      130  construction
## 231     1510      150 2012-Q2     1360      services
## 232        0        0 2012-Q2        0        others
## 233     1200       90 2012-Q3     1100 manufacturing
## 234      140       90 2012-Q3       50  construction
## 235     1510      240 2012-Q3     1270      services
## 236       10        0 2012-Q3       10        others
## 237     1580       50 2012-Q4     1530 manufacturing
## 238       70       30 2012-Q4       40  construction
```

```
## 239     1690  280 2012-Q4  1410      services
## 240        0    0 2012-Q4     0        others
## 241      680   10 2013-Q1   670 manufacturing
## 242      130   40 2013-Q1    90  construction
## 243     1300   60 2013-Q1  1240      services
## 244       10    0 2013-Q1    10        others
## 245     1630   30 2013-Q2  1610 manufacturing
## 246      250  100 2013-Q2   160  construction
## 247     1190  140 2013-Q2  1050      services
## 248        0    0 2013-Q2     0        others
## 249     1250   40 2013-Q3  1210 manufacturing
## 250      260   90 2013-Q3   170  construction
## 251     1200  190 2013-Q3  1020      services
## 252        0    0 2013-Q3     0        others
## 253     1430   90 2013-Q4  1350 manufacturing
## 254      480  180 2013-Q4   300  construction
## 255     1740   70 2013-Q4  1670      services
## 256        0    0 2013-Q4     0        others
## 257      820   10 2014-Q1   810 manufacturing
## 258      400  240 2014-Q1   160  construction
## 259     1890  280 2014-Q1  1610      services
## 260        0    0 2014-Q1     0        others
## 261      710  220 2014-Q2   490 manufacturing
## 262      280   80 2014-Q2   200  construction
## 263     1420  100 2014-Q2  1320      services
## 264        0    0 2014-Q2     0        others
## 265     1270  230 2014-Q3  1040 manufacturing
## 266      210   40 2014-Q3   170  construction
## 267     2030  130 2014-Q3  1900      services
## 268        0    0 2014-Q3     0        others
## 269     1170  170 2014-Q4  1000 manufacturing
## 270      800  340 2014-Q4   460  construction
## 271     1930  200 2014-Q4  1730      services
## 272        0    0 2014-Q4     0        others
## 273      950   30 2015-Q1   920 manufacturing
## 274      610  350 2015-Q1   260  construction
## 275     1930  180 2015-Q1  1750      services
## 276       10    0 2015-Q1    10        others
## 277      870  130 2015-Q2   750 manufacturing
## 278      230  110 2015-Q2   120  construction
## 279     2100  120 2015-Q2  1980      services
## 280       50    0 2015-Q2    50        others
## 281      920  130 2015-Q3   780 manufacturing
## 282      430  300 2015-Q3   130  construction
## 283     2120  270 2015-Q3  1850      services
## 284        0    0 2015-Q3     0        others
## 285     2480  140 2015-Q4  2340 manufacturing
## 286      520  250 2015-Q4   260  construction
## 287     2360  130 2015-Q4  2230      services
## 288       20    0 2015-Q4    20        others
## 289     1790  240 2016-Q1  1550 manufacturing
## 290      390  230 2016-Q1   150  construction
## 291     2530  150 2016-Q1  2380      services
## 292        0    0 2016-Q1     0        others
```

```
## 293     1380                130 2016-Q2              1250 manufacturing
## 294      350                140 2016-Q2               210  construction
## 295     3000                130 2016-Q2              2870      services
## 296       70                  0 2016-Q2                70        others
## 297     1120                160 2016-Q3               950 manufacturing
## 298      600                350 2016-Q3               250  construction
## 299     2510                200 2016-Q3              2310      services
## 300        0                  0 2016-Q3                 0        others
## 301     1990                130 2016-Q4              1860 manufacturing
## 302      580                320 2016-Q4               260  construction
## 303     2840                170 2016-Q4              2670      services
## 304       20                  0 2016-Q4                20        others
## 305      890                140 2017-Q1               740 manufacturing
## 306      660                290 2017-Q1               370  construction
## 307     2440                 80 2017-Q1              2370      services
## 308       10                  0 2017-Q1                10        others
## 309      840                130 2017-Q2               710 manufacturing
## 310      470                250 2017-Q2               220  construction
## 311     2330                140 2017-Q2              2190      services
## 312        0                  0 2017-Q2                 0        others
## 313      730                140 2017-Q3               590 manufacturing
## 314      490                130 2017-Q3               370  construction
## 315     2180                220 2017-Q3              1960      services
## 316        0                  0 2017-Q3                 0        others
## 317     1330                 50 2017-Q4              1290 manufacturing
## 318      400                160 2017-Q4               230  construction
## 319     1950                120 2017-Q4              1830      services
## 320       10                  0 2017-Q4                 0        others
## 321      510                 70 2018-Q1               440 manufacturing
## 322      350                120 2018-Q1               220  construction
## 323     1470                 80 2018-Q1              1390      services
## 324        0                  0 2018-Q1                 0        others
## 325      820                 50 2018-Q2               770 manufacturing
## 326      470                120 2018-Q2               350  construction
## 327     1740                100 2018-Q2              1640      services
## 328        0                  0 2018-Q2                 0        others
## 329      870                100 2018-Q3               770 manufacturing
## 330      200                 80 2018-Q3               120  construction
## 331     1800                120 2018-Q3              1680      services
## 332        0                  0 2018-Q3                 0        others
## 333      380                 70 2018-Q4               310 manufacturing
## 334      180                 70 2018-Q4               110  construction
## 335     1950                130 2018-Q4              1820      services
## 336        0                  0 2018-Q4                 0        others
## 337     1040                 30 2019-Q1              1020 manufacturing
## 338      280                 70 2019-Q1               210  construction
## 339     1900                120 2019-Q1              1780      services
## 340        0                  0 2019-Q1                 0        others
## 341      490                 70 2019-Q2               420 manufacturing
## 342      150                 70 2019-Q2                80  construction
## 343     1680                 60 2019-Q2              1620      services
## 344        0                  0 2019-Q2                 0        others
## 345      600                100 2019-Q3               490 manufacturing
## 346      160                 50 2019-Q3               110  construction
```

```
## 347     1690      70 2019-Q3    1620      services
## 348       20       0 2019-Q3      20        others
## 349      670      20 2019-Q4     650 manufacturing
## 350      270     180 2019-Q4     100  construction
## 351     1730      40 2019-Q4    1690      services
## 352       10       0 2019-Q4      10        others
## 353      720      20 2020-Q1     700 manufacturing
## 354      140      50 2020-Q1      90  construction
## 355     2360     120 2020-Q1    2240      services
## 356       10       0 2020-Q1      10        others
## 357     1550     150 2020-Q2    1400 manufacturing
## 358      440     100 2020-Q2     340  construction
## 359     6120     640 2020-Q2    5480      services
## 360       20       0 2020-Q2      20        others
## 361     2070      70 2020-Q3    2000 manufacturing
## 362      340      80 2020-Q3     250  construction
## 363     6710     830 2020-Q3    5880      services
## 364       10      10 2020-Q3       0        others
## 365      990     100 2020-Q4     900 manufacturing
## 366       70       0 2020-Q4      70  construction
## 367     4580    1430 2020-Q4    3150      services
## 368       10       0 2020-Q4      10        others
## 369      320      20 2021-Q1     300 manufacturing
## 370       20      10 2021-Q1      10  construction
## 371     1930      50 2021-Q1    1880      services
## 372        0       0 2021-Q1       0        others
## 373      760      10 2021-Q2     750 manufacturing
## 374       90      20 2021-Q2      70  construction
## 375     1480     150 2021-Q2    1330      services
## 376       20       0 2021-Q2      20        others
## 377      360      10 2021-Q3     360 manufacturing
## 378       90      10 2021-Q3      90  construction
## 379     1450     200 2021-Q3    1250      services
## 380        0       0 2021-Q3       0        others
## 381      280      40 2021-Q4     240 manufacturing
## 382       40      10 2021-Q4      40  construction
## 383     1160      40 2021-Q4    1120      services
## 384       30       0 2021-Q4      30        others
## 385      510      10 2022-Q1     500 manufacturing
## 386       60      20 2022-Q1      40  construction
## 387      730      30 2022-Q1     700      services
## 388       20       0 2022-Q1      20        others
## 389      170      10 2022-Q2     160 manufacturing
## 390       50      20 2022-Q2      30  construction
## 391      610      20 2022-Q2     590      services
## 392        0       0 2022-Q2       0        others
## 393      250     170 2022-Q3      80 manufacturing
## 394       10       0 2022-Q3      10  construction
## 395     1050      40 2022-Q3    1010      services
## 396        0       0 2022-Q3       0        others
## 397     1180     410 2022-Q4     770 manufacturing
## 398      150     110 2022-Q4      40  construction
## 399     1670      40 2022-Q4    1630      services
## 400        0       0 2022-Q4       0        others
```

Now, I'd like to find out - which particular years had the highest number of retrenchments in total (regardless of the industry)?

```
r_emp2 = mutate(r_emp, year = substr(quarter, 1, 4)) #create year column
head(r_emp2) #to check
```

```
##   retrench retrench_term_contract quarter retrench_permanent     industry1 year
## 1     6170                   1060 1998-Q1               5110 manufacturing 1998
## 2      560                    480 1998-Q1                 90  construction 1998
## 3     2100                    160 1998-Q1               1940      services 1998
## 4        0                      0 1998-Q1                  0        others 1998
## 5     5010                    120 1998-Q2               4890 manufacturing 1998
## 6      600                    170 1998-Q2                430  construction 1998
```

```
r_emp3 = select(r_emp2, retrench, year)
head(r_emp3) #relevant columns only, to check
```

```
##   retrench year
## 1     6170 1998
## 2      560 1998
## 3     2100 1998
## 4        0 1998
## 5     5010 1998
## 6      600 1998
```

```
r_emp4 = aggregate(r_emp3$retrench, by=list(year = r_emp3$year), FUN=sum)
names(r_emp4)[names(r_emp4) == "x"] = "retrench_total"

arrange(r_emp4, desc(retrench_total))#aggregated dataset based on year
```

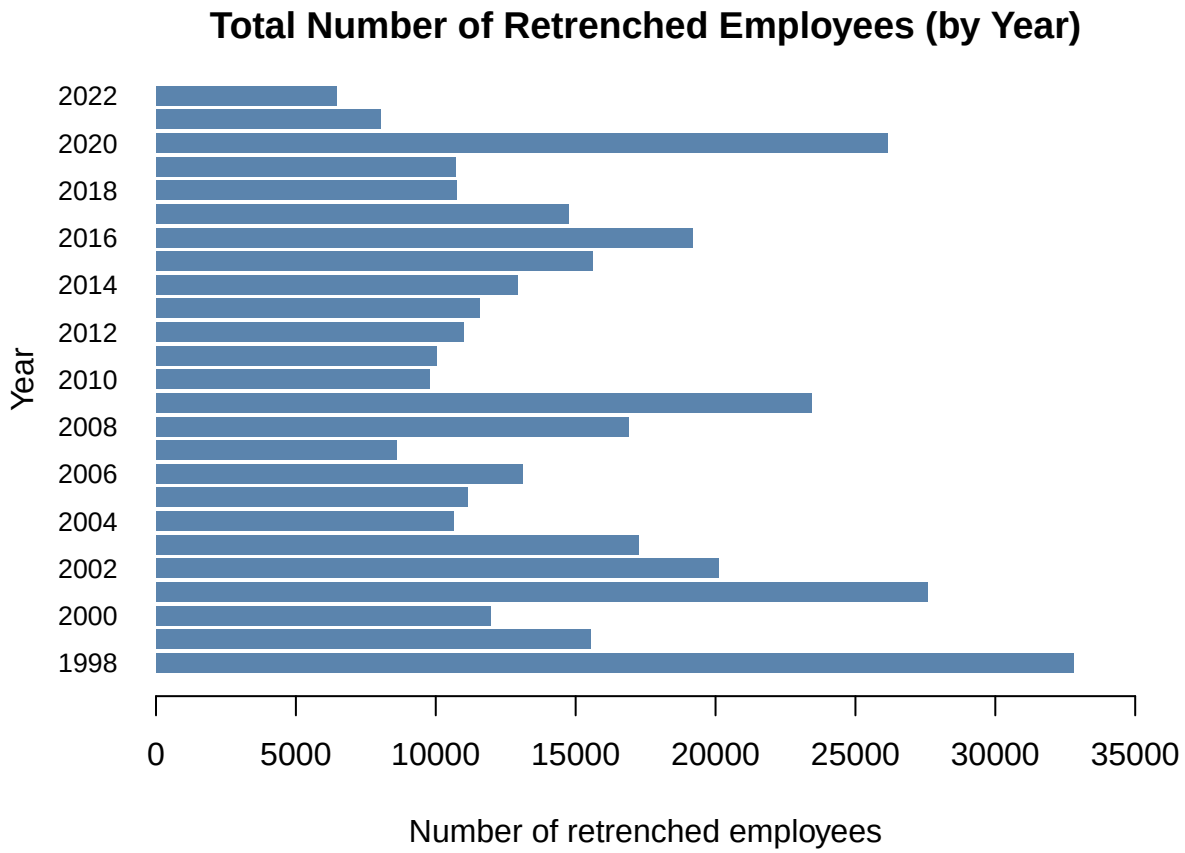
```
##    year retrench_total
## 1  1998          32800
## 2  2001          27580
## 3  2020          26140
## 4  2009          23440
## 5  2002          20120
## 6  2016          19170
## 7  2003          17260
## 8  2008          16900
## 9  2015          15600
## 10 1999          15530
## 11 2017          14730
## 12 2006          13090
## 13 2014          12930
## 14 2000          11950
## 15 2013          11550
## 16 2005          11150
## 17 2012          11000
## 18 2018          10740
## 19 2019          10690
## 20 2004          10650
## 21 2011          10020
```

```
## 22 2010           9790
## 23 2007           8590
## 24 2021           8030
## 25 2022           6460
```

```
head(r_emp4) #to check output
```

```
##   year retrench_total
## 1 1998          32800
## 2 1999          15530
## 3 2000          11950
## 4 2001          27580
## 5 2002          20120
## 6 2003          17260
```

```
par(las=1) #axis labels always horizontal
par(mar = c(4,5,2,2))
barplot(retrench_total ~ year,
        horiz = TRUE, data = r_emp4,
        ylab = "Year", xlab = "Number of retrenched employees",
        main = "Total Number of Retrenched Employees (by Year)",
        cex.names = 0.8, cex.axis = 1, xlim = c(0,35000), border = NA,
        col=rgb(0.2,0.4,0.6,0.8))
```




From the barplot above, we can tell that the top 5 years with the highest number of retrenched employees are:

1. 1998 (Asian Financial Crisis)
2. 2001 (Recession due to dot.com bust)
3. 2020 (COV1D-19 Outbreak)
4. 2009 (Aftermath of 2008 Global Financial Crisis)
5. 2002 (SARS Outbreak)

It is noted that in those particular years, Singapore was experiencing signs of economic recession/downturn/slowdown due to the events mentioned above, which explains the higher numbers of retrenched employees in the country.

## Question 2 - New York Flights data

1. How many flights departed NYC on December 25th 2013?

There are certain rows in the dataset that consist of NA values for dep_time and arr_time. Before we can investigate the dataset, such rows should be removed as they do not represent flights that actually departed NYC, which is the main emphasis of this entire question

```
full_flights = na.omit(flights) #remove all rows with NA values before investigating dataset
full_flights
```

```
## # A tibble: 327,346 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # ... with 327,336 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
#filter out flights that departed NYC on Dec 25, 2013
dec_25_nyc_flights = subset(full_flights, year == 2013 & month == 12 & day == 25)
dec_25_nyc_flights
```

```
## # A tibble: 715 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013    12    25      456            500        -4      649            651
##  2  2013    12    25      524            515         9      805            814
##  3  2013    12    25      542            540         2      832            850
##  4  2013    12    25      546            550        -4     1022           1027
##  5  2013    12    25      556            600        -4      730            745
##  6  2013    12    25      557            600        -3      743            752
##  7  2013    12    25      557            600        -3      818            831
```

```
##  8  2013    12    25      559            600        -1      855            856
##  9  2013    12    25      559            600        -1      849            855
## 10  2013    12    25      600            600         0      850            846
## # ... with 705 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
duplicated(dec_25_nyc_flights) #to check for duplicate records
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [469] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [481] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [505] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [517] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [529] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [553] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [565] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [577] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [589] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [613] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [637] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [673] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [685] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [697] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [709] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
dim(dec_25_nyc_flights)
```

```
## [1] 715  19
```

There were 715 flights that departed NYC on December 25th, 2013.

2. From the full dataset, flights, extract all flights originated from the JFK airport. Name the new object as data1.

```
data1 = subset(full_flights, origin == "JFK")
data1
```

```
## # A tibble: 109,079 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      542            540         2      923            850
##  2  2013     1     1      544            545        -1     1004           1022
##  3  2013     1     1      557            600        -3      838            846
##  4  2013     1     1      558            600        -2      849            851
##  5  2013     1     1      558            600        -2      853            856
##  6  2013     1     1      558            600        -2      924            917
##  7  2013     1     1      559            559         0      702            706
##  8  2013     1     1      606            610        -4      837            845
##  9  2013     1     1      611            600        11      945            931
## 10  2013     1     1      613            610         3      925            921
## # ... with 109,069 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

3. Select the following six columns: year, month, day, dep_delay, dest, air_time. Replace data1 with the new object with these selected columns.

```
myvars = c("year", "month", "day", "dep_delay", "dest", "air_time")
data1 = data1[,myvars]
data1
```

```
## # A tibble: 109,079 x 6
##     year month   day dep_delay dest  air_time
##    <int> <int> <int>     <dbl> <chr>    <dbl>
##  1  2013     1     1         2 MIA        160
##  2  2013     1     1        -1 BQN        183
##  3  2013     1     1        -3 MCO        140
##  4  2013     1     1        -2 PBI        149
##  5  2013     1     1        -2 TPA        158
##  6  2013     1     1        -2 LAX        345
##  7  2013     1     1         0 BOS         44
##  8  2013     1     1        -4 ATL        128
##  9  2013     1     1        11 SFO        366
## 10  2013     1     1         3 RSW        175
## # ... with 109,069 more rows
```

4. Create a new variable air_time_hrs in data1. The new variable is constructed as: air_time_hrs = air_time / 60

```
data1 = mutate(data1, air_time_hrs = air_time/60)
data1
```

```
## # A tibble: 109,079 x 7
##     year month   day dep_delay dest  air_time air_time_hrs
##    <int> <int> <int>     <dbl> <chr>    <dbl>        <dbl>
##  1  2013     1     1         2 MIA        160        2.67
##  2  2013     1     1        -1 BQN        183        3.05
##  3  2013     1     1        -3 MCO        140        2.33
##  4  2013     1     1        -2 PBI        149        2.48
##  5  2013     1     1        -2 TPA        158        2.63
##  6  2013     1     1        -2 LAX        345        5.75
##  7  2013     1     1         0 BOS         44        0.733
##  8  2013     1     1        -4 ATL        128        2.13
##  9  2013     1     1        11 SFO        366        6.1
## 10  2013     1     1         3 RSW        175        2.92
## # ... with 109,069 more rows
```

5. What is the average departure delay time for all flights in the dataset?

```
ans = mean(data1$dep_delay, na.rm = TRUE)
ans
```

```
## [1] 12.02361
```

Ans: 12.02361 mins

6. What is the mean departure delay on each day in 2013? (Hint: Use groupby() summarize(), and the pipe operator %>%.)

```
data12 = data1 %>%
  group_by(year, month, day) %>%
  summarize(mean_dep_delay = mean(dep_delay, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the
## `.groups` argument.
```

```
data12
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [12]
##     year month   day mean_dep_delay
##    <int> <int> <int>          <dbl>
##  1  2013     1     1          12.1
##  2  2013     1     2           8.22
##  3  2013     1     3          13.5
##  4  2013     1     4          10.5
##  5  2013     1     5           7.73
##  6  2013     1     6           6.02
##  7  2013     1     7           3.91
##  8  2013     1     8           3.76
##  9  2013     1     9           5.71
## 10  2013     1    10           2.20
## # ... with 355 more rows
```

7. How does departure delay vary with destination airport dest? Which destination airport has the highest delay time of any flight departing from NYC?

```
data13 = data1 %>%
  group_by(dest) %>%
  summarize(mean_dep_delay_dest = mean(dep_delay))

data13 #to check mean_dep_delay_time across all dest airports, departing from JFK
```

```
## # A tibble: 70 x 2
##    dest  mean_dep_delay_dest
##    <chr>               <dbl>
##  1 ABQ                 13.7
##  2 ACK                  6.45
##  3 ATL                 10.5
##  4 AUS                 14.0
##  5 BHM                  7
##  6 BNA                 18.7
##  7 BOS                 11.6
##  8 BQN                  6.68
##  9 BTV                 10.4
## 10 BUF                 13.2
## # ... with 60 more rows
```

```
arrange(data13, desc(mean_dep_delay_dest)) #CVG is dest with highest mean_dep_delay_time, when consider
```

```
## # A tibble: 70 x 2
##    dest  mean_dep_delay_dest
##    <chr>               <dbl>
##  1 CVG                 27.3
##  2 EGE                 23.4
```

```
##  3 SAT                  22.9
##  4 MCI                  22.6
##  5 CMH                  22.0
##  6 ORD                  21.2
##  7 SDF                  21.2
##  8 MSP                  20.7
##  9 DEN                  20.1
## 10 STL                  20
## # ... with 60 more rows
```

Based on the above, the destination airport with the highest delay time of any flight leaving JFK on average is Cincinnati/Northern Kentucky International Airport (CVG), with a reported time of 27.332983 mins.

If we were to investigate a similar statistic for all flights departing from any airport in NYC, however:

```
full_flights_q7 = full_flights %>%
  group_by(dest) %>%
  summarize(mean_dep_delay_dest = mean(dep_delay))

full_flights_q7 #to check mean_dep_delay_time across all dest airports, departing from any airport in N
```

```
## # A tibble: 104 x 2
##    dest  mean_dep_delay_dest
##    <chr>               <dbl>
##  1 ABQ                 13.7
##  2 ACK                  6.45
##  3 ALB                 23.4
##  4 ANC                 12.9
##  5 ATL                 12.4
##  6 AUS                 13.0
##  7 AVL                  8.15
##  8 BDL                 17.7
##  9 BGR                 19.2
## 10 BHM                 29.0
## # ... with 94 more rows
```

```
arrange(full_flights_q7, desc(mean_dep_delay_dest)) #TUL is dest with highest mean_dep_delay_time, when
```

```
## # A tibble: 104 x 2
##    dest  mean_dep_delay_dest
##    <chr>               <dbl>
##  1 TUL                 34.9
##  2 CAE                 33.8
##  3 OKC                 29.2
##  4 BHM                 29.0
##  5 TYS                 28.4
##  6 JAC                 27.5
##  7 DSM                 26.1
##  8 RIC                 23.6
##  9 MSN                 23.5
## 10 ALB                 23.4
## # ... with 94 more rows
```

Based on the above, the destination airport with the highest delay time of any flight leaving NYC on average is Tulsa International Airport (TUL), with a reported time of 34.887755 mins.

8. Explore the flights dataset by yourself. Answer one question you find interesting about the data. Include the code you used, and summarize (in words) what you found.

Considering all flights departing from NYC in 2013, which month has the highest departure and arrival delay time on average?

```
full_flights_agg_dd = aggregate(full_flights$dep_delay, list(full_flights$month), FUN = mean, na.rm = TR
full_flights_agg_dd1 = full_flights_agg_dd %>%
  rename(month = Group.1, mean_dep_delay = x)

arrange(full_flights_agg_dd1, desc(mean_dep_delay)) #aggregated dataset for mean departure delay time by
```

```
##    month mean_dep_delay
## 1      7      21.522179
## 2      6      20.725614
## 3     12      16.482161
## 4      4      13.849187
## 5      3      13.164289
## 6      5      12.891709
## 7      8      12.570524
## 8      2      10.760239
## 9      1       9.985491
## 10     9       6.630285
## 11    10       6.233175
## 12    11       5.420340
```

```
full_flights_agg_ad = aggregate(full_flights$arr_delay, list(full_flights$month), FUN = mean, na.rm = TR
full_flights_agg_ad1 = full_flights_agg_ad %>%
  rename(month = Group.1, mean_arr_delay = x)

arrange(full_flights_agg_ad1, desc(mean_arr_delay)) #aggregated dataset for mean arrival delay time by m
```

```
##    month mean_arr_delay
## 1      7     16.7113067
## 2      6     16.4813296
## 3     12     14.8703553
## 4      4     11.1760630
## 5      1      6.1299720
## 6      8      6.0406524
## 7      3      5.8075765
## 8      2      5.6130194
## 9      5      3.5215088
## 10    11      0.4613474
## 11    10     -0.1670627
## 12     9     -4.0183636
```

Based on the 2 aggregated datasets above, it seems that July is the month with the both highest departure and arrival delay times on average when considering all flights departing from NYC in 2013 (excluding NA values from cancelled flights etc). This is an interesting finding considering that July is considered the peak

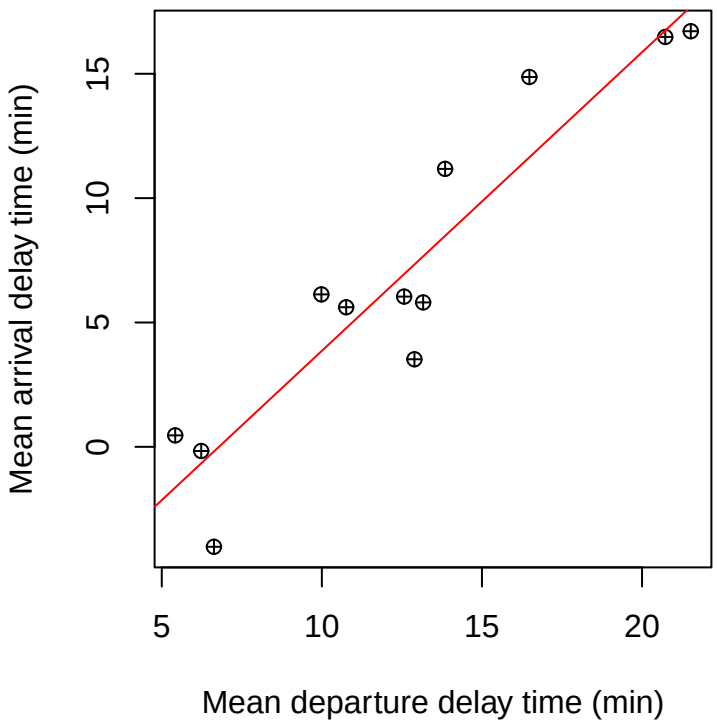
period of summer in New York, so by right the absence of heavy rainfall/snowfall should have reduced the average flight delay times, especially when departing from NYC.

If we were to further investigate the relationship between the mean departure delay and arrival times by month:

```
combined = merge(full_flights_agg_ad1, full_flights_agg_dd1, by = "month")
combined #merge both datasets by month
```

```
##    month mean_arr_delay mean_dep_delay
## 1      1      6.1299720       9.985491
## 2      2      5.6130194      10.760239
## 3      3      5.8075765      13.164289
## 4      4     11.1760630      13.849187
## 5      5      3.5215088      12.891709
## 6      6     16.4813296      20.725614
## 7      7     16.7113067      21.522179
## 8      8      6.0406524      12.570524
## 9      9     -4.0183636       6.630285
## 10    10     -0.1670627       6.233175
## 11    11      0.4613474       5.420340
## 12    12     14.8703553      16.482161
```

```
par(mar = c(4,9,4,9))
plot(mean_arr_delay ~ mean_dep_delay, data = combined, pch = 10,
     ylab = "Mean arrival delay time (min)", xlab = "Mean departure delay time (min)",
      main = "R/s between Monthly Mean Departure and Arrlival Flight Delay Times")
abline(lm(mean_arr_delay ~ mean_dep_delay, data = combined), col = "red")
```




From the above plot, it is pretty obvious that the mean departure delay time (by month) shares a positive relationship with the mean departure arrival time of flights from NYC. Logically speaking, this makes sense especially when an increase in departure delay time consequently increases the arrival delay time of a flight on average.

## Question 3 - Demographic score in Peru (Optional)

1. Read the CSV into R and name the object as dem. Describe whether it is considered "tidy data" and explain why.

```
dem = read_csv("Data/democracy_score.csv")
```

```
## Rows: 96 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): country
## dbl (9): YEAR1952, YEAR1957, YEAR1962, YEAR1967, YEAR1972, YEAR1977, YEAR198...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dem
```

```
## # A tibble: 96 x 10
```

```
##    country    YEAR1952 YEAR1957 YEAR1962 YEAR1967 YEAR1972 YEAR1977 YEAR1982
##    <chr>         <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1 Albania          -9       -9       -9       -9       -9       -9       -9
##  2 Argentina        -9       -1       -1       -9       -9       -9       -8
##  3 Armenia          -9       -7       -7       -7       -7       -7       -7
##  4 Australia        10       10       10       10       10       10       10
##  5 Austria          10       10       10       10       10       10       10
##  6 Azerbaijan       -9       -7       -7       -7       -7       -7       -7
##  7 Belarus          -9       -7       -7       -7       -7       -7       -7
##  8 Belgium          10       10       10       10       10       10       10
##  9 Bhutan          -10      -10      -10      -10      -10      -10      -10
## 10 Bolivia          -4       -3       -3       -4       -7       -7        8
## # ... with 86 more rows, and 2 more variables: YEAR1987 <dbl>, YEAR1992 <dbl>
```

The above is not considered "tidy data" as year is a numerical variable that should also have its own column.

2. Create a new object dem1 that contains the democracy scores of Peru only.

```
dem1 = subset(dem, country == "Peru")
dem1
```

```
## # A tibble: 1 x 10
##   country YEAR1952 YEAR1957 YEAR1962 YEAR1967 YEAR1972 YEAR1977 YEAR1982
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Peru          -2        5       -6        5       -7       -7        7
## # ... with 2 more variables: YEAR1987 <dbl>, YEAR1992 <dbl>
```

3. Convert dem1 into a tidy format, using what we learned in Week 6.

```
names(dem1) = gsub(pattern = "YEAR", replacement = "", x = names(dem1))
dem1
```

```
## # A tibble: 1 x 10
##   country `1952` `1957` `1962` `1967` `1972` `1977` `1982` `1987` `1992`
##   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Peru        -2      5     -6      5     -7     -7      7      7     -3
```

```
dem2 = dem1 %>%
  gather(`1952`:`1992`, key = "year", value = "score")

dem2
```

```
## # A tibble: 9 x 3
##   country year  score
##   <chr>   <chr> <dbl>
## 1 Peru    1952     -2
## 2 Peru    1957      5
## 3 Peru    1962     -6
## 4 Peru    1967      5
## 5 Peru    1972     -7
## 6 Peru    1977     -7
## 7 Peru    1982      7
## 8 Peru    1987      7
## 9 Peru    1992     -3
```