Data Science with R
Summarising Data

Graham.Williams@togaware.com

14th February 2014

Visit http://onepager.togaware.com/ for more OnePageR's.

The required packages for this module include:

```
library(rattle)         # The weatherAUS dataset.
library(plyr)           # Group by operations.
```

As we work through this module, new R commands will be introduced. Be sure to review the command's documentation and understand what the command does. You can ask for help using the `?` command as in:

```
?read.csv
```

We can obtain documentation on a particular package using the *help=* option of `library()`:

```
library(help=rattle)
```

This present module is intended to be hands on. To learn effectively, you are encouraged to have R running (e.g., RStudio) and to run all the commands as they appear here. Check that you get the same output, and you understand the output. Try some variations. Explore.

1 Load the Data

We use the full **weatherAUS** dataset from rattle (Williams, 2014) to illustrate data summarisation over a more complex dataset.

```
ds <- weatherAUS
names(ds)  <- normVarNames(names(ds)) # Lower case variable names.
names(ds)
```

```
##  [1] "date"           "location"       "min_temp"
##  [4] "max_temp"       "rainfall"       "evaporation"
##  [7] "sunshine"       "wind_gust_dir"  "wind_gust_speed"
## [10] "wind_dir_9am"   "wind_dir_3pm"   "wind_speed_9am"
....
```

```
head(ds)
```

```
##         date location min_temp max_temp rainfall evaporation sunshine
## 1 2008-12-01   Albury     13.4     22.9      0.6          NA       NA
## 2 2008-12-02   Albury      7.4     25.1      0.0          NA       NA
## 3 2008-12-03   Albury     12.9     25.7      0.0          NA       NA
....
```

```
tail(ds)
```

```
##             date location min_temp max_temp rainfall evaporation sunshine
## 82164 2014-01-25   Darwin     23.5     31.6     34.0         2.0      8.8
## 82165 2014-01-26   Darwin     27.2     31.2      0.0         6.4      4.5
## 82166 2014-01-27   Darwin     24.2     31.4      3.2         5.4      3.6
....
```

```
ds[sample(nrow(ds), 6),]
```

```
##             date     location min_temp max_temp rainfall evaporation
## 55639 2009-06-24     Adelaide      8.7     16.2      1.2         0.6
## 57740 2010-05-24 MountGambier      5.4     13.0      0.6         1.4
## 27977 2011-12-23     Canberra     12.2     27.8      0.6          NA
....
```

```
str(ds)
```

```
## 'data.frame': 82169 obs. of  24 variables:
##  $ date           : Date, format: "2008-12-01" "2008-12-02" ...
##  $ location       : Factor w/ 46 levels "Adelaide","Albany",..: 3 3 3 3 3 ...
##  $ min_temp       : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
....
```

```
summary(ds)
```

```
##       date                location        min_temp       max_temp
##  Min.   :2007-11-01   Canberra: 2163   Min.   :-8.5   Min.   :-3.7
##  1st Qu.:2010-01-31   Sydney  : 2071   1st Qu.: 7.5   1st Qu.:17.8
##  Median :2011-05-23   Adelaide: 1920   Median :11.9   Median :22.3
....
```

2 Dataset Indexing

Often we will be on the lookout for oddities or data typing that need fixing up. Once identified we will use the operations covered in a separate session on *Transform*ing data.

We start by looking at some of the data. This introduces the concept of indexing our data frame.

```
ds[1,]                  # First observation.
```

```
##         date location min_temp max_temp rainfall evaporation sunshine
## 1 2008-12-01   Albury     13.4     22.9      0.6          NA       NA
##   wind_gust_dir wind_gust_speed wind_dir_9am wind_dir_3pm wind_speed_9am
## 1             W              44            W          WNW             20
....
```

```
ds[1,1]                 # First observation's first variable.
```

```
## [1] "2008-12-01"
```

```
ds[1:2,]                # First two observations.
```

```
##         date location min_temp max_temp rainfall evaporation sunshine
## 1 2008-12-01   Albury     13.4     22.9      0.6          NA       NA
## 2 2008-12-02   Albury      7.4     25.1      0.0          NA       NA
##   wind_gust_dir wind_gust_speed wind_dir_9am wind_dir_3pm wind_speed_9am
....
```

```
ds[1:2, 3:4]            # First two observations and variables 3 and 4.
```

```
##   min_temp max_temp
## 1     13.4     22.9
## 2      7.4     25.1
```

```
head(ds[3:4], 2)        # Single dimension treated as variable index.
```

```
##   min_temp max_temp
## 1     13.4     22.9
## 2      7.4     25.1
```

```
head(ds[,3:4], 2)       # Or we can leave the observation index empty.
```

```
##   min_temp max_temp
## 1     13.4     22.9
## 2      7.4     25.1
```

3 Textual Summaries

The `summary()` command provides a quick univariate overview of our dataset.

```
summary(ds, digits=6)
```

```
##       date                location        min_temp       max_temp
##  Min.   :2007-11-01   Canberra: 2163   Min.   :-8.5   Min.   :-3.7
##  1st Qu.:2010-01-31   Sydney  : 2071   1st Qu.: 7.5   1st Qu.:17.8
##  Median :2011-05-23   Adelaide: 1920   Median :11.9   Median :22.3
##  Mean   :2011-05-19   Brisbane: 1920   Mean   :12.0   Mean   :22.9
##  3rd Qu.:2012-08-12   Darwin  : 1920   3rd Qu.:16.7   3rd Qu.:27.8
##  Max.   :2014-01-30   Hobart  : 1920   Max.   :33.9   Max.   :48.1
##                       (Other) :70255   NA's   :615    NA's   :456
##     rainfall      evaporation       sunshine     wind_gust_dir
##  Min.   :  0.0   Min.   : 0      Min.   : 0      W      : 5425
##  1st Qu.:  0.0   1st Qu.: 3      1st Qu.: 5      N      : 5375
##  Median :  0.0   Median : 5      Median : 8      SE     : 5303
##  Mean   :  2.5   Mean   : 5      Mean   : 8      S      : 5243
##  3rd Qu.:  0.8   3rd Qu.: 7      3rd Qu.:11      SW     : 5183
##  Max.   :371.0   Max.   :82      Max.   :14      (Other):49159
##  NA's   :1492    NA's   :29976   NA's   :31915   NA's   : 6481
##  wind_gust_speed  wind_dir_9am    wind_dir_3pm   wind_speed_9am
##  Min.   :  6     N      : 6733   SE     : 6387   Min.   : 0.0
##  1st Qu.: 31     SE     : 5163   W      : 5643   1st Qu.: 7.0
##  Median : 39     E      : 5096   S      : 5600   Median :13.0
##  Mean   : 40     SSE    : 5061   WSW    : 5463   Mean   :14.2
##  3rd Qu.: 48     S      : 4933   SW     : 5360   3rd Qu.:20.0
##  Max.   :135     (Other):48986   (Other):51828   Max.   :87.0
##  NA's   :6441    NA's   : 6197   NA's   : 1888   NA's   :1125
##  wind_speed_3pm  humidity_9am    humidity_3pm    pressure_9am
##  Min.   : 0.0   Min.   :  0.0   Min.   :  0.0   Min.   : 980
##  1st Qu.:13.0   1st Qu.: 57.0   1st Qu.: 37.0   1st Qu.:1013
##  Median :19.0   Median : 70.0   Median : 52.0   Median :1017
##  Mean   :18.8   Mean   : 68.8   Mean   : 51.8   Mean   :1017
##  3rd Qu.:24.0   3rd Qu.: 83.0   3rd Qu.: 66.0   3rd Qu.:1022
##  Max.   :87.0   Max.   :100.0   Max.   :100.0   Max.   :1041
##  NA's   :1139   NA's   :1425    NA's   :1277    NA's   :7786
##   pressure_3pm    cloud_9am       cloud_3pm        temp_9am
##  Min.   : 979   Min.   :0       Min.   :0       Min.   :-5.9
##  1st Qu.:1010   1st Qu.:1       1st Qu.:2       1st Qu.:12.2
##  Median :1015   Median :5       Median :5       Median :16.6
##  Mean   :1015   Mean   :4       Mean   :4       Mean   :16.8
##  3rd Qu.:1020   3rd Qu.:7       3rd Qu.:7       3rd Qu.:21.4
##  Max.   :1040   Max.   :9       Max.   :9       Max.   :40.2
##  NA's   :7763   NA's   :29373   NA's   :30353   NA's   :992
....
```

4 Textual Summaries—Warning

Do be weary of the results provided by `summary()`. The `summary()` command rounds the results to 4 digits by default. This can surprise us sometimes when we find `min()` and the reported minimum value from `summary()` disagree! Let's look at some random data and notice the reported minimum value.

```
eg <- sample(1e6:(1e7-1), 100)
max(eg)
```

```
## [1] 9980468
```

```
min(eg)
```

```
## [1] 1164081
```

```
summary(eg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1160000 2930000 5220000 5290000 7420000 9980000
```

```
summary(eg, digits=4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1164000 2928000 5223000 5294000 7420000 9980000
```

```
summary(eg, digits=5)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1164100 2928100 5223200 5294100 7420300 9980500
```

```
summary(eg, digits=6)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1164080 2928120 5223210 5294080 7420300 9980470
```

```
summary(eg, digits=7)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 1164081 2928123 5223212 5294075 7420295 9980468
```


5 PlyR: Summarise per Group to new Data Frame

The plyr (Wickham, 2012) package provides a clean and consistent approach to transforming data. We can easily, for example, transform a data frame into a new smaller data frame grouped by the location.

```
temps <- ddply(ds, "location", summarise,
               max=max(max_temp, na.rm=TRUE),
               min=min(min_temp, na.rm=TRUE))
temps
```

```
##            location  max  min
## 1          Adelaide 45.7  0.7
## 2            Albany 38.9  1.8
## 3            Albury 44.8 -2.5
....
```

The plyr package also provides the `.()` function as a convenient mechanism for listing variable names without the need to quote them. The function becomes more convenient when we have multiple variables to list.

```
temps <- ddply(ds, .(location), summarise,
               max=max(max_temp, na.rm=TRUE),
               min=min(min_temp, na.rm=TRUE))
temps
```

```
##            location  max  min
## 1          Adelaide 45.7  0.7
## 2            Albany 38.9  1.8
## 3            Albury 44.8 -2.5
....
```

We can review the resulting values, ordered by the maximum temperature.

```
temps[order(temps$max, decreasing=TRUE),]
```

```
##            location  max  min
## 46          Woomera 48.1  0.7
## 21            Moree 47.3 -3.3
## 19 MelbourneAirport 46.8 -0.4
....
```

Similarly, but ordered by the minimum temperature.

```
head(temps[order(temps$min),])
```

```
##        location  max  min
## 23  MountGinini 31.1 -8.5
## 39  Tuggeranong 40.1 -8.2
## 10     Canberra 42.0 -8.0
....
```

6 PlyR: Summarise per Group to Original Data Frame

Transform a data frame by adding the group summaries per original observation, simply by replacing `summarise` with `transform`

```
temps <- ddply(ds, .(location), transform,
               max=max(max_temp, na.rm=TRUE),
               min=min(min_temp, na.rm=TRUE))
```

Now notice that the top few values for `min` and `max` are constant, since they belong to the same group (`Adelaide`).

```
head(temps[c("date", "location", "min_temp", "min", "max_temp", "max")])
```

```
##         date location min_temp min max_temp  max
## 1 2008-07-01 Adelaide      8.8 0.7     15.7 45.7
## 2 2008-07-02 Adelaide     12.7 0.7     15.8 45.7
## 3 2008-07-03 Adelaide      6.2 0.7     15.1 45.7
....
```

If we same a few observations we see the various values of `min` and `max` across different `locations`.

```
temps[sample(nrow(temps), 10),
      c("date", "location", "min_temp", "min", "max_temp", "max")]
```

```
##             date         location min_temp  min max_temp  max
## 53154 2011-08-01            Perth     11.3 -0.6     20.1 43.3
## 43289 2013-11-28        Newcastle     13.8  2.9     34.8 44.1
## 16132 2013-12-15           Cairns     24.3  9.2     32.7 38.6
....
```

7 PlyR: Select One Observation Per Group

We can also select a single observation per group, using some criteria to decide which observation to pick. We replace the `summarise` or `transform` with a function to select the observation of interest.

```
temps <- ddply(ds, .(location),
               function(x) x[x$max_temp == max(x$max_temp, na.rm=TRUE),])
head(temps[1:7])
```

```
##         date location min_temp max_temp rainfall evaporation sunshine
## 1       <NA>     <NA>       NA       NA       NA          NA       NA
## 2 2009-01-28 Adelaide     30.7     45.7        0        13.0     12.5
## 3 2010-01-18   Albany     17.8     38.9        0        11.8     12.8
....
```

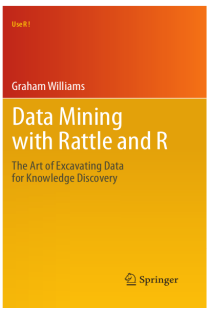
Notice the unexpected rows of missing values. The vector comparison, using `==()`, will return `NA` whenever comparing NA's and an index to `[()` of NA will return an NA row for each observation. We can get around this issue of missing values by testing whether we get `TRUE` from the comparison, rather than `FALSE` or `NA` by using `identical()`.

```
temps <- ddply(ds, .(location),
               function(x) x[sapply(x$max_temp == max(x$max_temp, na.rm=TRUE),
                                    identical, TRUE),])
head(temps[1:7])
```

```
##         date      location min_temp max_temp rainfall evaporation sunshine
## 1 2009-01-28      Adelaide     30.7     45.7        0        13.0     12.5
## 2 2010-01-18        Albany     17.8     38.9        0        11.8     12.8
## 3 2009-02-07        Albury     22.3     44.8        0          NA       NA
....
```

8 Further Reading

The Rattle Book, published by Springer, provides a comprehensive introduction data mining and analytics using Rattle and R. It is available from Amazon. Other documentation on a broader selection of R topics of relevance to the data scientist is freely available from `http://datamining.togaware.com`, including the Datamining Desktop Survival Guide.



This module is one of many OnePageR modules available from `http://onepager.togaware.com`. In particular follow the links on the website with a * which indicates the generally more developed OnePageR modules.

9 References

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Wickham H (2012). *plyr: Tools for splitting, applying and combining data*. R package version 1.8, URL http://CRAN.R-project.org/package=plyr.

Williams GJ (2009). "Rattle: A Data Mining GUI for R." *The R Journal*, **1**(2), 45–55. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

Williams GJ (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery*. Use R! Springer, New York. URL http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896.

Williams GJ (2014). *rattle: Graphical user interface for data mining in R*. R package version 3.0.2, URL http://rattle.togaware.com/.

This document, sourced from SummaryO.Rnw revision 282, was processed by KnitR version 1.5 of 2013-09-28 and took 4.5 seconds to process. It was generated by gjw on nyx running Ubuntu 13.10 with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 4 cores and 12.3GB of RAM. It completed the processing 2014-02-14 06:09:10.