Project One for Data Mining

Raymond Anthony Ford
raford2@miners.utep.edu

Due: 06 February 2018

Contents

1 Data Input **1**

2 Data Cleaning and Preparation **3**
2.1 Removing a Variable (b) 3
2.2 Renaming Variables (c) 4
2.3 Inspecting the Variables (a) 4
2.4 Recoding Variables (d, e) 6
2.5 Saving the Cleaned Data (f) 6

3 Exploratory Data Analysis **6**

4 Appendix **8**
4.1 Code for Data Input 8
4.2 Code for Data Cleaning and Preparation 8
4.3 Code for Exploratory Data Analysis 9

1 Data Input

We use the provided code to acquire the data from the Australian daily weather database.

```
dat <- NULL
current.month <- 12
for (i in 1:current.month){
    i0 <- ifelse(i<10, paste("0", i, sep=""), i)
    mth <- paste("2017", i0, sep="")
    bom <- paste("http://www.bom.gov.au/climate/dwo/", mth,
                 "/text/IDCJDW2801.", mth, ".csv", sep="")
    dat.i <- read.csv(bom, skip=6, check.names=FALSE,
    na.strings = c("NA", "", " "))
    dat.i[, 1] <-toupper(month.abb[i])
    dat <- rbind(dat, dat.i)
}
dim(dat)
```

```
## [1] 365  22
```

From the output, we see that we have obtained 22 measurements for 365 days. The first few lines of the data are displayed below via the `head()` command. It should be noted that we could have made this into an actual LaTeX table by changing the variable names, and cleaning it up a bit so that it is more presentable, but that particular task will be accomplished in one of the sub-problems in the next section. This is why the first few lines of the data look the way that they do.

```
head(dat)
```

```
##            Date Minimum temperature (\xb0C) Maximum temperature (\xb0C)
## 1 JAN 2017-01-1                         12.6                         27.8
## 2 JAN 2017-01-2                         15.7                         29.9
## 3 JAN 2017-01-3                         13.1                         27.1
## 4 JAN 2017-01-4                         11.0                         25.1
## 5 JAN 2017-01-5                         11.2                         29.9
## 6 JAN 2017-01-6                         14.5                         30.6
##   Rainfall (mm) Evaporation (mm) Sunshine (hours)
## 1           0.0               NA               NA
## 2           0.0               NA               NA
## 3           0.0               NA               NA
## 4           0.0               NA               NA
## 5           0.0               NA               NA
## 6           1.4               NA               NA
##   Direction of maximum wind gust  Speed of maximum wind gust (km/h)
## 1                             NW                                37
## 2                             SE                                43
## 3                            ENE                                44
## 4                              E                                37
## 5                            SSW                                54
## 6                            ENE                                43
##   Time of maximum wind gust 9am Temperature (\xb0C)
## 1                     14:42                    21.0
## 2                     16:02                    19.6
## 3                     17:17                    19.3
## 4                     16:38                    17.4
## 5                     16:57                    18.5
## 6                     17:54                    20.3
##   9am relative humidity (%) 9am cloud amount (oktas) 9am wind direction
## 1                        74                        1                  N
## 2                        73                        3                 NE
## 3                        53                        4                 SE
## 4                        68                        8                ENE
## 5                        68                        4                 SE
## 6                        69                       NA                 SE
```

```
##   9am wind speed (km/h) 9am MSL pressure (hPa) 3pm Temperature (\xb0C)
## 1                     2                1006.2                    26.7
## 2                     9                1009.2                    27.7
## 3                    17                1018.3                    24.5
## 4                    13                1021.0                    22.8
## 5                     9                1018.4                    28.6
## 6                     7                1017.5                    28.9
##   3pm relative humidity (%) 3pm cloud amount (oktas) 3pm wind direction
## 1                        34                       8                WNW
## 2                        33                       8                  S
## 3                        39                       1                ENE
## 4                        46                       8                  E
## 5                        33                       1                 NE
## 6                        29                       1                 NE
##   3pm wind speed (km/h) 3pm MSL pressure (hPa)
## 1                    22                1004.6
## 2                    13                1008.3
## 3                    28                1016.2
## 4                    15                1018.5
## 5                    15                1014.0
## 6                    24                1014.3
```

2 Data Cleaning and Preparation

We elect to perform the tasks assigned in a different order than assigned. The variable `"Time of maximum wind gust"` will be removed, and then we will rename the variables, as their names are both long, contain spaces, and contain special characters that return an error on my machine.[1] Finally we will go back to the problems as assigned beginning with inspecting the variables for suspicious or problematic records.

2.1 Removing a Variable (b)

We use the code provided to remove the variable `"Time of maximum wind gust"`, to make the analysis more meaningful.

```
dat <- dat[, -c(10)]
```

[1] I believe it is a utf problem, as the following error is displayed: `Error in tolower(completions) : invalid multibyte string 4` .

2.2 Renaming Variables (c)

We modify the code provided to rename the measurements in `dat`. This will decrease the length of the strings for the variables, remove the spaces in their names, and overall make it easier to work with the data. The code and its output is below. The function `colnames()` is used as a sanity check in place of `names()`, to ensure that no semantic error occurred during the deletion of `"Time of maximum wind gust"`.

```
colnames(dat) <- c("Month", "Date", "MinTemp", "MaxTemp", "Rainfall",
                   "Evaporation", "Sunshine", "WindGustDir", "WindGustSpeed",
                   "Temp9am", "Humidity9am", "Cloud9am", "WindDir9am",
                   "WindSpeed9am", "Pressure9am", "Temp3pm", "Humidity3pm",
                   "Cloud3pm", "WindDir3pm", "WindSpeed3pm", "Pressure3pm")
dim(dat)
```

```
## [1] 365  21
```

Looking at the output, we see that the number of variables has decreased by one, and now we double-check that the new variable names were created as intended.

```
names(dat)
```

```
##  [1] "Month"         "Date"          "MinTemp"       "MaxTemp"
##  [5] "Rainfall"      "Evaporation"   "Sunshine"      "WindGustDir"
##  [9] "WindGustSpeed" "Temp9am"       "Humidity9am"   "Cloud9am"
## [13] "WindDir9am"    "WindSpeed9am"  "Pressure9am"   "Temp3pm"
## [17] "Humidity3pm"   "Cloud3pm"      "WindDir3pm"    "WindSpeed3pm"
## [21] "Pressure3pm"
```

2.3 Inspecting the Variables (a)

We begin our inspection of the variables by looking at the number of missing values for each of the variables, the data type for each variable, and the number of unique instances for each variable. The table below presents this summary.

```
vnames <- colnames(dat)
n <- nrow(dat)
out <- NULL
for (j in 1:ncol(dat)){
    vname <- colnames(dat)[j]
    x <- as.vector(dat[,j])
    nmiss <- sum(is.na(x))
    ncomplete <- n-nmiss
    out <- rbind(out, c(col.num=j, v.name=vname,
                mode=mode(x), n.level=length(unique(x)),
                ncom=ncomplete, miss.prop=nmiss/n))
```

```
}
out <- as.data.frame(out)
row.names(out) <- NULL
out
```

```
##    col.num        v.name      mode n.level ncom           miss.prop
## 1        1         Month character      12  365                   0
## 2        2          Date character     365  365                   0
## 3        3       MinTemp   numeric     200  364 0.00273972602739726
## 4        4       MaxTemp   numeric     203  365                   0
## 5        5      Rainfall   numeric      45  363 0.00547945205479452
## 6        6   Evaporation   logical       1    0                   1
## 7        7      Sunshine   logical       1    0                   1
## 8        8   WindGustDir character      17  361   0.010958904109589
## 9        9 WindGustSpeed   numeric      38  361   0.010958904109589
## 10      10       Temp9am   numeric     192  365                   0
## 11      11   Humidity9am   numeric      64  365                   0
## 12      12      Cloud9am   numeric       9  187   0.487671232876712
## 13      13    WindDir9am character      17  334  0.0849315068493151
## 14      14  WindSpeed9am character      21  365                   0
## 15      15   Pressure9am   numeric     198  365                   0
## 16      16       Temp3pm   numeric     200  365                   0
## 17      17   Humidity3pm   numeric      76  365                   0
## 18      18      Cloud3pm   numeric       9  184   0.495890410958904
## 19      19    WindDir3pm character      16  365                   0
## 20      20  WindSpeed3pm   numeric      23  365                   0
## 21      21   Pressure3pm   numeric     208  365                   0
```

We notice that some of the variables have problematic recordings. For instance, `Evaporation` and `Sunshine` are missing a recording for every day in 2017. Moreover, `WindSpeed9am` is classified as a character variable when it should be numeric.

We then print each variable out with the following code to take a closer look at its recorded values.

```
apply(dat, 2, FUN=function(x){table(x, useNA='ifany')})
```

Looking at the values for all of the variables, we confirm that both `Evaporation` and `Sunshine` do not have any recordings for 2017, and that there were 31 instances in which `WindSpeed9am` was recorded as `Calm`. Since there is no reason to include variables for which no recordings were made, we remove the variables `Evaporation` and `Sunshine` from the data.

```
dat <- dat[, -c(6,7)]
```

2.4 Recoding Variables (d, e)

As noted in the previous subsection, there were some problematic recordings for `WindSpeed9am`, as 31 of the recordings were recorded as `Calm` while all other recordings were numeric. We fix this issue by changing all values of `Calm` to `0`, and then changing the vector type of `WindSpeed9am` from character to numeric.

```
dat[dat$WindSpeed9am == "Calm",]$WindSpeed9am <- 0
dat$WindSpeed9am <- as.numeric(dat$WindSpeed9am)
```

Now we define a variable called `RainToday` based on `Rainfall` so that `RainToday` is `1` if `Rainfall` is greater than 1mm and `0` otherwise.

```
RainToday <- NULL
for (i in 1:length(dat$Rainfall)){
    if (dat$Rainfall[i] < 1 | is.na(dat$Rainfall[i]))
        RainToday[i] <- 0
    else
        RainToday[i] <- 1
}
dat$RainToday <- RainToday
```

Next, we define a variable called `RainTomorrow` by shifting `RainToday` one day forward.

```
dat$RainTomorrow <- c(dat$RainToday[2:nrow(dat)], NA)
```

2.5 Saving the Cleaned Data (f)

We now save this data set as a csv file, so that we can submit the data set with along with this project report for grading.

```
write.csv(dat, file='FordRA_Project01_Data.csv')
```

3 Exploratory Data Analysis

Since we're interested in the dichotomous response variable `RainTomorrow`, it seems natural to look at it's association with all of the other variables. Thus, we perform a χ^2 test with `RainTomorrow` and each of the other categorical variables individually. The code that was used is below.

```
for (k in 1:ncol(dat)) {
    vname <- colnames(dat)[k]
    print(k)
    print(vname)
    x <- as.vector(dat[,k])
```

```
    tab <- table(dat$RainTomorrow, x, useNA='no')
    print(tab)
    print(chisq.test(tab))
}
```

We find that there were three variables that were significant at the $\alpha = 0.05$ significance level: `Month`, `Cloud3pm`, and `RainToday`. The p-values of these tests and other pertinent information is given in Table 1.

Variable	χ^2	df	p-value
Month	22.151	11	0.0232
Cloud3pm	15.699	7	0.02801
RainToday	10.098	1	0.0015

Table 1: Test statistics and p-values for χ^2 tests of the variables against RainTomorrow.

These variables seem pretty reasonable choices to include in a model seeking to predict whether or not it will rain on day $k + 1$ given that it rained on day k. Surprisingly, most of the rainy days occurred during March and not during the Winter months of June, July, or August. In fact, the Winter months did not seem to have many days for which it rained.

4 Appendix

This appendix contains all of the R code used in this project and is organized by the problem for which it was used answer the questions for this assignment.

4.1 Code for Data Input

```
dat <- NULL
current.month <- 12
for (i in 1:current.month){
    i0 <- ifelse(i<10, paste("0", i, sep=""), i)
    mth <- paste("2017", i0, sep="")
    bom <- paste("http://www.bom.gov.au/climate/dwo/", mth,
                 "/text/IDCJDW2801.", mth, ".csv", sep="")
    dat.i <- read.csv(bom, skip=6, check.names=FALSE,
    na.strings = c("NA", "", " "))
    dat.i[, 1] <-toupper(month.abb[i])
    dat <- rbind(dat, dat.i)
}
dim(dat)
```

4.2 Code for Data Cleaning and Preparation

```
dat <- dat[, -c(10)]
colnames(dat) <- c("Month", "Date", "MinTemp", "MaxTemp", "Rainfall",
                   "Evaporation", "Sunshine", "WindGustDir", "WindGustSpeed",
                   "Temp9am", "Humidity9am", "Cloud9am", "WindDir9am",
                   "WindSpeed9am", "Pressure9am", "Temp3pm", "Humidity3pm",
                   "Cloud3pm", "WindDir3pm", "WindSpeed3pm", "Pressure3pm")
dim(dat)
names(dat)
vnames <- colnames(dat)
n <- nrow(dat)
out <- NULL # Start building dataframe for variable summaries
for (j in 1:ncol(dat)){
    vname <- colnames(dat)[j]
    x <- as.vector(dat[,j])
    nmiss <- sum(is.na(x))
    ncomplete <- n-nmiss
    out <- rbind(out, c(col.num=j, v.name=vname,
                        mode=mode(x), n.level=length(unique(x)),
```

```
                        ncom=ncomplete, miss.prop=nmiss/n))
}
out <- as.data.frame(out)
row.names(out) <- NULL # End building
out # Print variable types, proportion missing, etc
# Nextline prints all variables and their recordings
apply(dat, 2, FUN=function(x){table(x, useNA='ifany')})
dat <- dat[, -c(6,7)] # Remove Sunshine and Evaporation
dat[dat$WindSpeed9am == "Calm",]$WindSpeed9am <- 0
dat$WindSpeed9am <- as.numeric(dat$WindSpeed9am)
RainToday <- NULL
for (i in 1:length(dat$Rainfall)){
    if (dat$Rainfall[i] < 1 | is.na(dat$Rainfall[i]))
      RainToday[i] <- 0
    else
      RainToday[i] <- 1
}
dat$RainToday <- RainToday
dat$RainTomorrow <- c(dat$RainToday[2:nrow(dat)], NA)
write.csv(dat, file='FordRA_Project01_Data.csv')
```

4.3 Code for Exploratory Data Analysis

```
# Perform and print results for a chi-squared test on each variable
for (k in 1:ncol(dat)) {
    vname <- colnames(dat)[k]
    print(k)
    print(vname)
    x <- as.vector(dat[,k])
    tab <- table(dat$RainTomorrow, x, useNA='no')
    print(tab)
    print(chisq.test(tab))
}
```