SI2020_dataprep_exercise_wrangling_wsolutn

PFR

August 2020

PFR data prep exercises for data wrangling

This is an R Markdown document for data prep exercises.

This exercise is to 'reshape' the data as an example of data wrangling/munging

Also, later is a comparison with 'dplyr' aggregation

##load data

```
#use setwd("c:/your-path-to-your-project/data/")

W_df = read.table('weather_orig.csv',
                      header=TRUE,sep=",",
                      stringsAsFactors = TRUE)  #try TRUE
head(W_df)
```

```
##         Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 1 2007-11-01 Canberra     8.0    24.3      0.0         3.4      6.3
## 2 2007-11-02 Canberra    14.0    26.9      3.6         4.4      9.7
## 3 2007-11-03 Canberra    13.7    23.4      3.6         5.8      3.3
## 4 2007-11-04 Canberra    13.3    15.5     39.8         7.2      9.1
## 5 2007-11-05 Canberra     7.6    16.1      2.8         5.6     10.6
## 6 2007-11-06 Canberra     6.2    16.9      0.0         5.8      8.2
##   WindGustDir WindGustSpeed WindDir9am WindDir3pm WindSpeed9am
## 1          NW            30         SW         NW            6
## 2         ENE            39          E          W            4
## 3          NW            85          N        NNE            6
## 4          NW            54        WNW          W           30
## 5         SSE            50        SSE        ESE           20
## 6          SE            44         SE          E           20
##   WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am
## 1           20          68          29      1019.7      1015.0        7
## 2           17          80          36      1012.4      1008.4        5
## 3            6          82          69      1009.5      1007.2        8
## 4           24          62          56      1005.5      1007.0        2
## 5           28          68          49      1018.3      1018.5        7
## 6           24          70          57      1023.8      1021.7        7
##   Cloud3pm Temp9am Temp3pm RainToday RISK_MM RainTomorrow
## 1        7    14.4    23.6        No     3.6          Yes
## 2        3    17.5    25.7       Yes     3.6          Yes
## 3        7    15.4    20.2       Yes    39.8          Yes
## 4        7    13.5    14.1       Yes     2.8          Yes
## 5        7    11.1    15.4       Yes     0.0           No
## 6        5    10.9    14.8        No     0.2           No
```

##reshape data First, let's try installing this package. 'reshape' is in base R, but 'reshape2' is a newer version

```
if ("reshape2" %in% rownames(installed.packages())==FALSE)
  { install.packages('reshape2')
} else {print('reshape2 installed already')}
```

```
## [1] "reshape2 installed already"
```

```
library("reshape2")
```

##reshape data Now, imagine that each day we want to list a measurement for each wind direction all in the same row. You might think of it as doing linear regression where each factor level is it's own variable.

1. run this section and notice what the new row looks like, Where are the new columns?

Task: Use WindGustDir and WindGustSpeed as variable names to fill in the command below. The formula indicates that the other variables identify the repeated measurement.

Which variable labels the repeated measurement, which variable has the measurement value?

```
library(reshape2)

# long to wide: ie 'cast' repeated measure into wide table
W_wide   =dcast(W_df,
              formula=Date+Location+ ...~ WindGustDir,
               fill=0,
               value.var="WindGustSpeed")

head(W_wide)
```

```
##         Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 1 2007-11-01 Canberra     8.0    24.3      0.0         3.4      6.3
## 2 2007-11-02 Canberra    14.0    26.9      3.6         4.4      9.7
## 3 2007-11-03 Canberra    13.7    23.4      3.6         5.8      3.3
## 4 2007-11-04 Canberra    13.3    15.5     39.8         7.2      9.1
## 5 2007-11-05 Canberra     7.6    16.1      2.8         5.6     10.6
## 6 2007-11-06 Canberra     6.2    16.9      0.0         5.8      8.2
##   WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am Humidity3pm
## 1         SW         NW            6           20          68          29
## 2          E          W            4           17          80          36
## 3          N        NNE            6            6          82          69
## 4        WNW          W           30           24          62          56
## 5        SSE        ESE           20           28          68          49
## 6         SE          E           20           24          70          57
##   Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday
## 1      1019.7      1015.0        7        7    14.4    23.6        No
## 2      1012.4      1008.4        5        3    17.5    25.7       Yes
## 3      1009.5      1007.2        8        7    15.4    20.2       Yes
## 4      1005.5      1007.0        2        7    13.5    14.1       Yes
## 5      1018.3      1018.5        7        7    11.1    15.4       Yes
## 6      1023.8      1021.7        7        5    10.9    14.8        No
##   RISK_MM RainTomorrow E ENE ESE N NE NNE NNW NW S SE SSE SSW SW W WNW WSW
## 1     3.6          Yes 0   0   0 0  0   0   0 30 0  0   0   0  0 0   0   0
## 2     3.6          Yes 0  39   0 0  0   0   0  0 0  0   0   0  0 0   0   0
## 3    39.8          Yes 0   0   0 0  0   0   0 85 0  0   0   0  0 0   0   0
## 4     2.8          Yes 0   0   0 0  0   0   0 54 0  0   0   0  0 0   0   0
## 5     0.0           No 0   0   0 0  0   0   0  0 0  0  50   0  0 0   0   0
## 6     0.2           No 0   0   0 0  0   0   0  0 0 44   0   0  0 0   0   0
##   NA
## 1  0
## 2  0
## 3  0
## 4  0
## 5  0
## 6  0
```

```
#optional: write.csv(W_cast,file='Weather_castwide.csv')
```

##To reshape data from wide to long use 'melt' command. Similar packages use names like gather/spread, pivot/unpivot

Let's try a selection, group by and aggregation in package 'dpylr'

```
if ("dplyr" %in% rownames(installed.packages())==FALSE)
  { install.packages('dplyr')
} else {print('dplyr installed already')}
```

```
## [1] "dplyr installed already"
```

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

dplyr has some commands to identify groups then summarize data like in SQL. It is good for quick or one-time operations, but if you have a big dataset it may not be efficient like a true database system

```
#First get group ids, 1 id for each Windgustdir
a1 <- group_by(na.omit(W_df), WindGustDir)
a2 <- select(a1,WindSpeed9am,Temp9am)
```

```
## Adding missing grouping variables: `WindGustDir`
```

```
a3 <- summarise(a2,
  avg_speed = mean(WindSpeed9am, na.rm = TRUE),
  avg_temp   = mean(Temp9am, na.rm = TRUE)
)
```

Now compare means from both

```
a3[which(a3[,'WindGustDir']=='ESE'),]
```

```
## # A tibble: 1 x 3
##   WindGustDir avg_speed avg_temp
##   <fct>             <dbl>    <dbl>
## 1 ESE                10.2     13.9
```

```
colMeans(W_wide[which(W_wide$ESE>0),c('WindSpeed9am','Temp9am')])
```

```
## WindSpeed9am      Temp9am
##     10.21739     13.85217
```