Problem Sheet 5 - CT5102

Data Transformation with dplyr

1. Based on the **mpg** dataset from **ggplot2**, generate the following tibble which filters all the cars with a **cty** value greater than the median. Ensure that your tibble contains the same columns, and with **set.seed(100)** sample 5 records using **sample_n()**, and store the result in the tibble **ans**.

```
ans
```

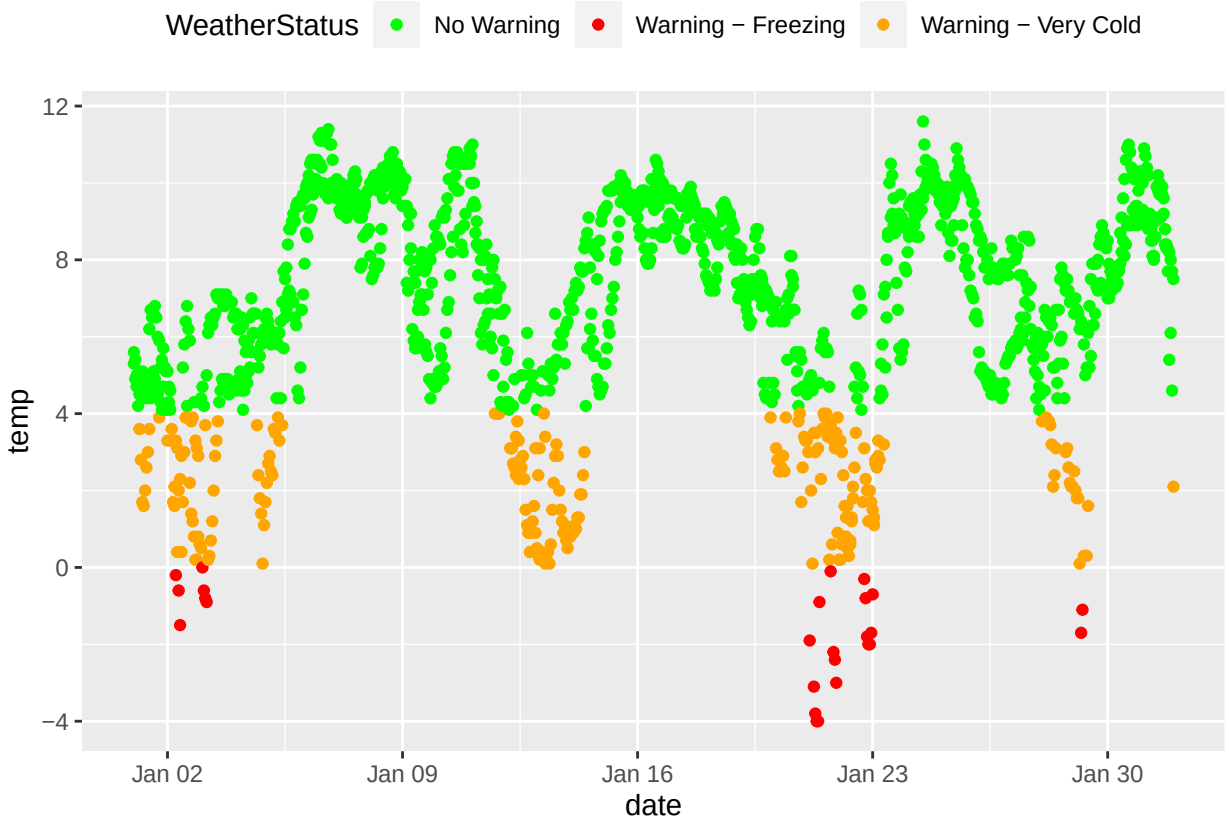
```
## # A tibble: 5 x 7
##   manufacturer model        displ  year   cty   hwy class
##   <chr>        <chr>        <dbl> <int> <int> <int> <chr>
## 1 audi         a4             2    2008    20    31 compact
## 2 volkswagen   jetta          2.5  2008    21    29 compact
## 3 hyundai      sonata         2.4  2008    21    31 midsize
## 4 hyundai      sonata         2.4  1999    18    26 midsize
## 5 toyota       camry solara   3.3  2008    18    27 compact
```

2. Based on the **aimsir17** tibble **observations**, generate the tibble **jan** which contains observations for two weather stations ("DUBLIN AIRPORT" and "MACE HEAD") during the month of January. Add a new column named **WeatherStatus** that contains three possible values: "Warning - Freezing" if the temperature is less than or equal to 0, "Warning - Very Cold" if the temperature is greater than zero and less than or equal to four, and "No Warning" if the temparture is above four. Make use of the **case_when()** function, and replicate the plot.

```
jan
```

```
## # A tibble: 1,488 x 7
```

```
##    station        month   day  hour date                 temp WeatherStatus
##    <chr>          <dbl> <int> <int> <dttm>              <dbl> <chr>
##  1 DUBLIN AIRPORT     1     1     0 2017-01-01 00:00:00   5.3 No Warning
##  2 MACE HEAD          1     1     0 2017-01-01 00:00:00   5.6 No Warning
##  3 DUBLIN AIRPORT     1     1     1 2017-01-01 01:00:00   4.9 No Warning
##  4 MACE HEAD          1     1     1 2017-01-01 01:00:00   5.4 No Warning
##  5 DUBLIN AIRPORT     1     1     2 2017-01-01 02:00:00   5   No Warning
##  6 MACE HEAD          1     1     2 2017-01-01 02:00:00   4.7 No Warning
##  7 DUBLIN AIRPORT     1     1     3 2017-01-01 03:00:00   4.2 No Warning
##  8 MACE HEAD          1     1     3 2017-01-01 03:00:00   4.7 No Warning
##  9 DUBLIN AIRPORT     1     1     4 2017-01-01 04:00:00   3.6 Warning - Very Co~
## 10 MACE HEAD          1     1     4 2017-01-01 04:00:00   4.5 No Warning
## # i 1,478 more rows
```



3. Generate the following tibble (**diam**) based on the **diamonds** tibble from **ggplot2**. Note that the column **PriceMaxColour** is the colour of the diamond with the maximum price

for a given cut.

```
diam
```

```
## # A tibble: 5 x 5
##   cut       NumberDiamonds CaratMean PriceMax PriceMaxColour
##   <ord>              <int>     <dbl>    <int> <ord>
## 1 Ideal              21551     0.703    18806 G
## 2 Premium            13791     0.892    18823 I
## 3 Very Good          12082     0.806    18818 G
## 4 Good                4906     0.849    18788 G
## 5 Fair                1610     1.05     18574 G
```

4. For each class of car, create the tibble **mpg1** that contains a new column that stores the rank of city miles per gallon (**cty**), from lowest to highest. Make use of the **rank()** function in R, and in this function call, set the argument **ties.method** = "first".

```
mpg1
```

```
## # A tibble: 234 x 7
##    manufacturer model      displ  year   cty class   RankCty
##    <chr>        <chr>      <dbl> <int> <int> <chr>     <int>
##  1 chevrolet    corvette     5.7  1999    15 2seater       1
##  2 chevrolet    corvette     6.2  2008    15 2seater       2
##  3 chevrolet    corvette     7    2008    15 2seater       3
##  4 chevrolet    corvette     5.7  1999    16 2seater       4
##  5 chevrolet    corvette     6.2  2008    16 2seater       5
##  6 audi         a4 quattro   2.8  1999    15 compact       1
##  7 audi         a4 quattro   3.1  2008    15 compact       2
##  8 audi         a4           2.8  1999    16 compact       3
##  9 audi         a4 quattro   1.8  1999    16 compact       4
## 10 volkswagen   jetta        2.8  1999    16 compact       5
## # i 224 more rows
```

5. Find the stations with the highest (**temp_high**) and lowest (**temp_low**) annual average temperature values. Use these variables to calculate the average monthly temperature values for the two stations (**m_temps**), and display the data in a plot.

```
temp_low
```

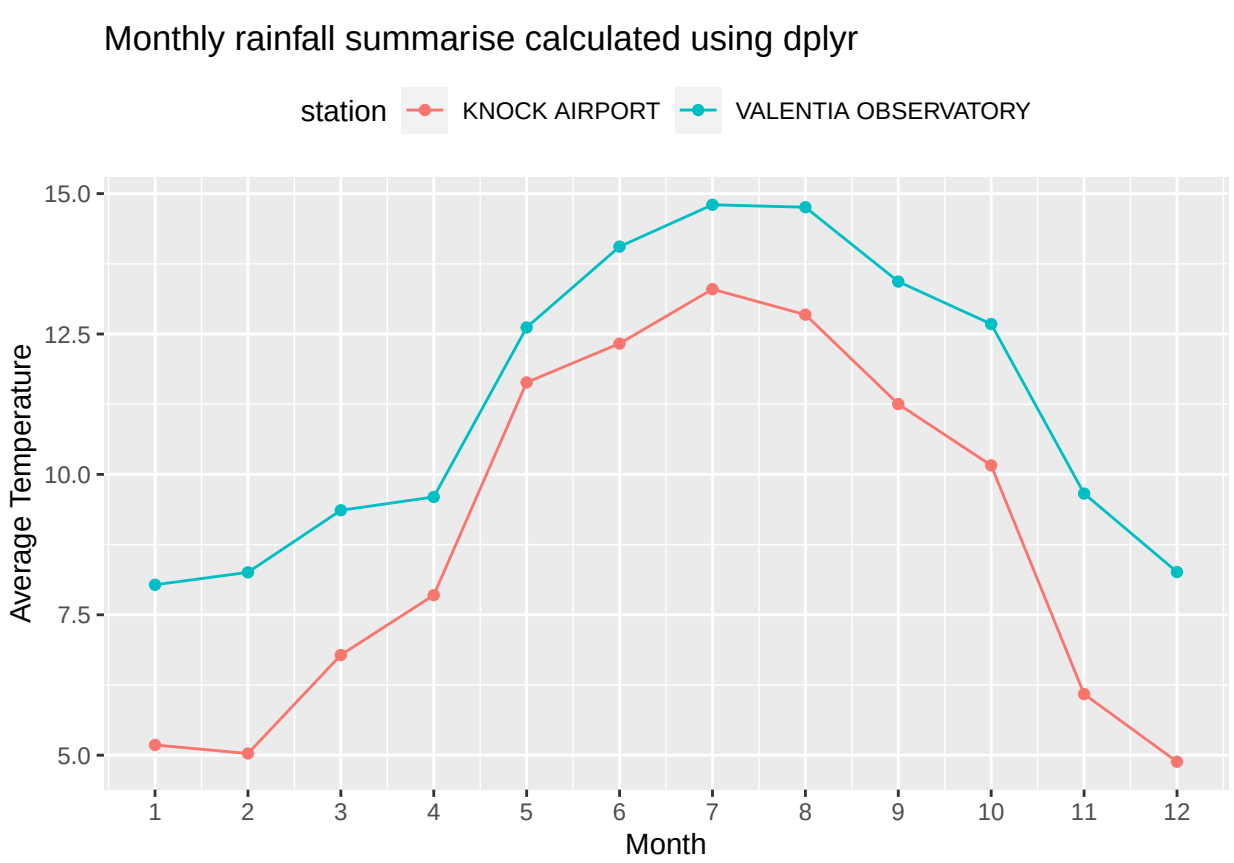
```
## [1] "KNOCK AIRPORT"
```

```
temp_high
```

```
## [1] "VALENTIA OBSERVATORY"
```

```
arrange(m_temps,month,station)
```

```
## # A tibble: 24 x 3
##    station              month AvrTemp
##    <chr>                <dbl>   <dbl>
##  1 KNOCK AIRPORT            1    5.18
##  2 VALENTIA OBSERVATORY     1    8.03
##  3 KNOCK AIRPORT            2    5.03
##  4 VALENTIA OBSERVATORY     2    8.26
##  5 KNOCK AIRPORT            3    6.78
##  6 VALENTIA OBSERVATORY     3    9.36
##  7 KNOCK AIRPORT            4    7.85
##  8 VALENTIA OBSERVATORY     4    9.60
##  9 KNOCK AIRPORT            5   11.6
## 10 VALENTIA OBSERVATORY     5   12.6
## # i 14 more rows
```


Monthly rainfall summarise calculated using dplyr
station
KNOCK AIRPORT
VALENTIA OBSERVATORY
Average Temperature
15.0
12.5
10.0
7.5
5.0
1
2
3
4
5
6
7
8
9
10
11
12
Month