Tidyverse Giriş: Şehirlerin Hava Durumu - Bölüm I
dplyr

AB 2018

Jan 28, 2018

Giriş

Bu döküman tidyverse paket grubuna en temelden girişi sağlamak için yazılmıştır. Tidyverse bir grup özel R paketini kapsayan bir tepe pakettir ve genelde veri manipülasyonu ve görselleştirme amaçlarıyla kullanılır. Bu derste daha çok (en önemli paketler olan) `dplyr` ve `ggplot2` üzerine odaklanacağız ama ilgili diğer paketlerden de fonksiyonları kullanabiliriz.

Diyelim ki sık seyahat eden birisiniz ve hava durumu sizin için önemli, çünkü sıcaklık durumuna göre bavul hazırlayacaksınız. Elimizdeki veri 4 popüler seyahat noktasının (NY, Amsterdam, Londra, Venedik) Kasım 2015 - Ekim 2017 arasındaki geçmiş sıcaklıklarını vermektedir. Ham veri Weather Underground adresinden alınmıştır ve sadece eğitim amaçlı kullanılmaktadır. Sizin yapmanız gereken tidyverse fonksiyonlarıyla bu veri üzerinde işlemler yapmak. Aşağıdaki kısımlarda kodlar içerisindeki boşlukları doldurmanız beklenmektedir.

İpucu: Her zaman yardım dosyalarına göz atabilirsiniz. Bunun için R konsoluna yardım almak istediğiniz fonksiyonun başına `?` koyarak (ör. `?select`) yardım dosyasını açabilirsiniz. Tabi bunun için ilgili paketi yüklemiş olmanız gerekiyor.

Hazırlık

İlk önce `tidyverse` paketini indirin ve kurun. Bir paketi indirmek bir kerelik bir iş ve sadece sunucundan indirmekten oluşmaktadır. Ancak her oturumda `library` veya `require` fonksiyonlarıyla paketleri yüklemeniz gerekmektedir. Bu döküman için aynı zamanda `sehir_sicaklik.RData` dosyasini buradan indirmeniz gerekiyor.

```
# Eger paketi hala indirmediyseniz indirin
install.packages("tidyverse", repos = "https://cran.r-project.org")
# Paketi oturuma yukleyin
library(tidyverse)
# Calisma klasorunuzu belirleyin (sehir_sicaklik.RData
# dosyasinin bulundugu klasor olabilir)
setwd("~/BenimCalismaKlasorum/")
# Veri setini yukleyin
load("sehir_sicaklik.RData")
```

Bu dökümanın ana veri türü `data.frame`, veya daha doğru bir terimle `tibble` formatıdır. Data frame iki boyutlu, yüksek verimli veri tablolarıdır ve her sütun farklı bir veri tipinden oluşabilir (ör. character, factor, numeric, logical). `tibble` özel bir data frame türü olup tidyverse paketiyle birlikte gelmektedir ama işlevi düz data.frame ile çok benzemektedir (bu döküman için bir fark yoktur).

Artık verimizle ilgilenebiliriz.

Şehir Sıcaklık Verisi

```
sehir_sicaklik %>%
    tbl_df()
```

```
## # A tibble: 731 x 7
##      yil    ay   gun Amsterdam Londra    NY Venedik
##  * <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
##  1  2015  11.0  1.00      8.00   8.00  16.0   13.0
##  2  2015  11.0  2.00     10.0   11.0   15.0   10.0
##  3  2015  11.0  3.00      9.00  11.0   16.0    9.00
##  4  2015  11.0  4.00     12.0   11.0   17.0   10.0
##  5  2015  11.0  5.00     13.0   13.0   18.0   12.0
##  6  2015  11.0  6.00     16.0   14.0   21.0   13.0
##  7  2015  11.0  7.00     16.0   14.0   17.0   14.0
##  8  2015  11.0  8.00     12.0   12.0   11.0   13.0
##  9  2015  11.0  9.00     13.0   12.0   11.0   11.0
## 10  2015  11.0 10.0      14.0   14.0   12.0   11.0
## # ... with 721 more rows
```

`%>%` ifadesini fark ettiniz mi? Buna zincir operatörü denmektedir. Veri ile başlar ve işlemleri bir sıra mantığında birbirinin ardına dizer (yukarıdan aşağıya ya da soldan sağa). (*İpucu*: Zincir operatörlerinin arasına satır boşlukları koyabilirsiniz ama hep zincir operatörü satır sonunda kalacak bir şekilde kodlamaya devam edin.)

`tibble`ların bilmeniz gereken bazı özellikleri vardır. İlk kısımda satır ve sütun sayıları verilir (A tibble 731x7). Ayrıca her sütunun altında veri tipi gösterilir. Bu sayede bu veri tablosunun temelleri hakkında daha fazla bilgi sahibi olmuş olursunuz.

Daha düzgün bir kontrol `glimpse` fonksiyonu ile yapılablir. Sütun sayısı fazlaysa `glimpse` rahatlık sağlar.

```
glimpse(sehir_sicaklik)
```

```
## Observations: 731
## Variables: 7
## $ yil       <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015...
## $ ay        <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, ...
## $ gun       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1...
## $ Amsterdam <dbl> 8, 10, 9, 12, 13, 16, 16, 12, 13, 14, 13, 13, 11, 11...
## $ Londra    <dbl> 8, 11, 11, 11, 13, 14, 14, 12, 12, 14, 13, 12, 10, 1...
## $ NY        <dbl> 16, 15, 16, 17, 18, 21, 17, 11, 11, 12, 12, 13, 11, ...
## $ Venedik   <dbl> 13, 10, 9, 10, 12, 13, 14, 13, 11, 11, 9, 11, 8, 11,...
```

Veri setimiz 731 satır ve 7 sütundan oluşmaktadır. Her satır bir günü temsil eder. İlk üç sütun (`yil`,`ay` and `gun`) ilgili tarihi belirler. Son dört sütun (`Amsterdam`, `Londra`, `NY` ve `Venedik`) o gün o şehirdeki ortalama sıcaklıkları verir.

Şimdi de veride keşif yapma zamanı.

dplyr

Bu kısımda dplyr paketinin temel fonksiyonlarını ve biraz daha fazlasını göreceğiz. Eğer dplyr cheat sheet ile birlikte bu dökümanı takip ediyorsanız maksimum verimi alacaksınızdır. Cheat sheeti bu adresten edinebilirsiniz. Temel fonksiyonlarımız aşağıdaki gibidir.

- `select/rename`

- filter
- arrange
- mutate/transmute
- group_by/summarise

Basitten başlayıp üzerine devam edeceğiz.

select/rename

`select` fonksiyonu, adı üstünde, sütun seçer. `rename` sadece sütun ismi değiştirir.

1. Tek şehirle başlayalım: Venedik. Tarih bileşenlerini (yil, ay, gun) ve Venedik sütununu seçin. Aşağıdaki kodda `CEVAPBURAYA` yerine doğru ifadeyi yazıp sonucu tekrar etmeye çalışın.

```
sehir_sicaklik %>% select(yil, ay, gun, CEVAPBURAYA)
```

```
## # A tibble: 731 x 4
##      yil    ay   gun Venedik
##  * <dbl> <dbl> <dbl>   <dbl>
##  1  2015  11.0  1.00   13.0
##  2  2015  11.0  2.00   10.0
##  3  2015  11.0  3.00    9.00
##  4  2015  11.0  4.00   10.0
##  5  2015  11.0  5.00   12.0
##  6  2015  11.0  6.00   13.0
##  7  2015  11.0  7.00   14.0
##  8  2015  11.0  8.00   13.0
##  9  2015  11.0  9.00   11.0
## 10  2015  11.0 10.0    11.0
## # ... with 721 more rows
```

2. Diyelim ki sadece şehirlerin sıcaklıklarını istiyorsunuz ve gün önemli değil. İster bütün şehirlerin ismini yazarsınız, isterseniz `:` kullanarak bir seferde seçebilirsiniz.

```
sehir_sicaklik %>% select(CEVAPBURAYA1:CEVAPBURAYA2)
```

```
## # A tibble: 731 x 4
##    Amsterdam Londra    NY Venedik
##  *     <dbl>  <dbl> <dbl>   <dbl>
##  1      8.00   8.00  16.0   13.0
##  2     10.0   11.0   15.0   10.0
##  3      9.00  11.0   16.0    9.00
##  4     12.0   11.0   17.0   10.0
##  5     13.0   13.0   18.0   12.0
##  6     16.0   14.0   21.0   13.0
##  7     16.0   14.0   17.0   14.0
##  8     12.0   12.0   11.0   13.0
##  9     13.0   12.0   11.0   11.0
## 10     14.0   14.0   12.0   11.0
## # ... with 721 more rows
```

3. Bu sefer (`-`) kullanarak istenmeyen sütunlardan kurtulacağız. Diyelim ki NY ve Londra sütunlarını istmiyoruz.

```
sehir_sicaklik %>% select(-CEVAPBURAYA1, -CEVAPBURAYA2)
```

```
## # A tibble: 731 x 5
```

```
##      yil    ay   gun Amsterdam Venedik
##   * <dbl> <dbl> <dbl>     <dbl>   <dbl>
##  1  2015  11.0  1.00      8.00   13.0
##  2  2015  11.0  2.00     10.0    10.0
##  3  2015  11.0  3.00      9.00    9.00
##  4  2015  11.0  4.00     12.0    10.0
##  5  2015  11.0  5.00     13.0    12.0
##  6  2015  11.0  6.00     16.0    13.0
##  7  2015  11.0  7.00     16.0    14.0
##  8  2015  11.0  8.00     12.0    13.0
##  9  2015  11.0  9.00     13.0    11.0
## 10  2015  11.0 10.0      14.0    11.0
## # ... with 721 more rows
```

4. Diyelim ki sadece NY sütununu New York yapmak istiyoruz. Sütun isimlerinde boşluk kullanmak tavsiye edilmemesine karşın bunu çapraz tırnak kullanarak yapabilirsiniz. `rename` sütun seçmez sadece isim değiştirir unutmayın.

```
sehir_sicaklik %>% rename(`CEVAP BURAYA` = NY)
```

```
## # A tibble: 731 x 7
##      yil    ay   gun Amsterdam Londra `New York` Venedik
##   * <dbl> <dbl> <dbl>     <dbl>  <dbl>      <dbl>   <dbl>
##  1  2015  11.0  1.00      8.00   8.00       16.0   13.0
##  2  2015  11.0  2.00     10.0   11.0        15.0   10.0
##  3  2015  11.0  3.00      9.00  11.0        16.0    9.00
##  4  2015  11.0  4.00     12.0   11.0        17.0   10.0
##  5  2015  11.0  5.00     13.0   13.0        18.0   12.0
##  6  2015  11.0  6.00     16.0   14.0        21.0   13.0
##  7  2015  11.0  7.00     16.0   14.0        17.0   14.0
##  8  2015  11.0  8.00     12.0   12.0        11.0   13.0
##  9  2015  11.0  9.00     13.0   12.0        11.0   11.0
## 10  2015  11.0 10.0      14.0   14.0        12.0   11.0
## # ... with 721 more rows
```

Tip: `select` de `rename` işlevine ve daha fazlasına sahiptir.

filter

Filter verilen kriterleri sağlayan satırları döndürür. Herhangi bir kriter seti verebilir ve kombinleyebilirsiniz. Bunun için "ve" (`&`) ve "veya" (`|`) operatörlerini kullanmanız gerekmektedir. Ayrıca büyüktür, küçüktür, büyük eşit, küçük eşit (`<`,`<=`,`>`,`>=`,`==`, `!=`) gibi TRUE/FALSE döndürecek diğer farklı operatörleri de kullanabilirsiniz. Operatörleri birleştirebilir ve parantezler kullanabilirsiniz.

1. Diyelim ki her ayın sadece ilk 3 günüyle ilgileniyoruz.

```
sehir_sicaklik %>%
    filter(gun <= CEVAPBURAYA)
```

```
## # A tibble: 72 x 7
##      yil    ay   gun Amsterdam Londra    NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
##  1  2015 11.0   1.00      8.00   8.00 16.0    13.0
##  2  2015 11.0   2.00     10.0   11.0  15.0    10.0
##  3  2015 11.0   3.00      9.00  11.0  16.0     9.00
##  4  2015 12.0   1.00      9.00  11.0   9.00    6.00
```

```
##  5  2015 12.0   2.00     10.0   12.0  11.0     8.00
##  6  2015 12.0   3.00      9.00  11.0  10.0     8.00
##  7  2016  1.00  1.00      4.00   3.00  3.00    2.00
##  8  2016  1.00  2.00      6.00  10.0   2.00    0
##  9  2016  1.00  3.00      7.00   8.00  4.00    3.00
## 10  2016  2.00  1.00     10.0   12.0  11.0     6.00
## # ... with 62 more rows
```

2. Diyelim ki sadece kasım ayında Venedik'in NY'tan daha sıcak olduğu günleri istiyoruz.

```
sehir_sicaklik %>%
    filter(ay == 11 & CEVAPBURAYA)
```

```
## # A tibble: 20 x 7
##      yil    ay   gun Amsterdam Londra    NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
##  1  2015  11.0  8.00     12.0   12.0  11.0    13.0
##  2  2015  11.0 14.0      11.0   10.0   8.00   11.0
##  3  2015  11.0 15.0      12.0   14.0   9.00   11.0
##  4  2015  11.0 17.0      13.0   13.0   8.00    9.00
##  5  2015  11.0 23.0       3.00   3.00  4.00    6.00
##  6  2015  11.0 24.0       5.00   8.00  4.00    6.00
##  7  2016  11.0  1.00     10.0    9.00  9.00   11.0
##  8  2016  11.0  6.00      7.00   4.00 11.0    12.0
##  9  2016  11.0  7.00      4.00   6.00  8.00   11.0
## 10  2016  11.0 12.0       1.00   8.00  7.00    9.00
## 11  2016  11.0 19.0       6.00   4.00 10.0    11.0
## 12  2016  11.0 20.0       7.00   7.00  3.00   11.0
## 13  2016  11.0 21.0      10.0   10.0   4.00   12.0
## 14  2016  11.0 22.0      10.0    9.00  4.00   14.0
## 15  2016  11.0 23.0       8.00   7.00  4.00   14.0
## 16  2016  11.0 24.0       6.00   9.00  6.00   13.0
## 17  2016  11.0 25.0       3.00   7.00 10.0    13.0
## 18  2016  11.0 26.0       3.00   6.00  7.00   12.0
## 19  2016  11.0 27.0       5.00   7.00  7.00   11.0
## 20  2016  11.0 28.0       1.00   6.00  7.00    8.00
```

3. Diyelim ki Amsterdam'ın Temmuz'da Londra veya Venedik'ten daha sıcak olduğu günleri istiyoruz.

```
sehir_sicaklik %>%
    filter(ay == 7 & (CEVAPBURAYA1 | CEVAPBURAYA2))
```

```
## # A tibble: 21 x 7
##      yil    ay   gun Amsterdam Londra    NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
##  1  2016  7.00  2.00      16.0   14.0  21.0    25.0
##  2  2016  7.00 11.0       19.0   18.0  23.0    27.0
##  3  2016  7.00 12.0       18.0   17.0  24.0    28.0
##  4  2016  7.00 13.0       16.0   14.0  26.0    27.0
##  5  2016  7.00 19.0       21.0   20.0  26.0    27.0
##  6  2016  7.00 20.0       27.0   24.0  25.0    26.0
##  7  2016  7.00 21.0       21.0   19.0  27.0    26.0
##  8  2016  7.00 22.0       21.0   19.0  29.0    26.0
##  9  2016  7.00 23.0       22.0   19.0  31.0    26.0
## 10  2016  7.00 24.0       21.0   19.0  29.0    25.0
## # ... with 11 more rows
```

4. Son olarak biraz da matematiksel işlem ekleyelim. Amsterdam ve Venedik arasındaki mutlak sıcaklık farkının 12 derece veya daha fazla olduğu günleri getirelim.

```
sehir_sicaklik %>%
    filter(abs(CEVAPBURAYA) >= 12)
```

```
## # A tibble: 6 x 7
##     yil    ay   gun Amsterdam Londra    NY Venedik
##   <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
## 1  2016  6.00 25.0       16.0   15.0  24.0    28.0
## 2  2017  7.00 13.0       14.0   14.0  29.0    27.0
## 3  2017  8.00  2.00      18.0   17.0  26.0    30.0
## 4  2017  8.00  4.00      19.0   18.0  25.0    31.0
## 5  2017  8.00  5.00      17.0   16.0  23.0    31.0
## 6  2017  8.00  6.00      16.0   17.0  21.0    29.0
```

arrange

`arrange` basitçe değerleri A'dan Z'ye veya küçükten büyüğe sıralar. Sadece sıralamak istediğiniz sütunları yazmanız yeterli. Ters sırada kullanmak için ilgili sütun isimlerini `desc(column_name)` olarak ifade etmelisiniz.

1. Veriyi NY şehrinin sıcaklığına göre sıralayın.

```
sehir_sicaklik %>%
    arrange(CEVAPBURAYA)
```

```
## # A tibble: 731 x 7
##      yil    ay   gun Amsterdam Londra     NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl>  <dbl>   <dbl>
##  1  2016  2.00 14.0       2.00   3.00 -14.0     6.00
##  2  2016  2.00 13.0       1.00   2.00 -10.0     4.00
##  3  2016  1.00  5.00      6.00   8.00 - 7.00    2.00
##  4  2017  1.00  9.00      6.00   7.00 - 7.00   -2.00
##  5  2016  1.00 19.0      -2.00   0    - 6.00    1.00
##  6  2016  2.00 12.0       2.00   1.00 - 6.00    6.00
##  7  2016 12.0  16.0       6.00   6.00 - 6.00    4.00
##  8  2017  1.00  8.00      4.00   9.00 - 6.00   -2.00
##  9  2017  1.00  7.00      1.00   8.00 - 5.00   -3.00
## 10  2017  3.00 11.0       7.00  10.0  - 5.00    9.00
## # ... with 721 more rows
```

2. Veriyi NY'taki sıcaklığın artışı ve sonra Amsterdam'ın düşüşü şeklinde sıralayın.

```
sehir_sicaklik %>%
    arrange(CEVAPBURAYA1,desc(CEVAPBURAYA2))
```

```
## # A tibble: 731 x 7
##      yil    ay   gun Amsterdam Londra     NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl>  <dbl>   <dbl>
##  1  2016  2.00 14.0       2.00   3.00 -14.0     6.00
##  2  2016  2.00 13.0       1.00   2.00 -10.0     4.00
##  3  2016  1.00  5.00      6.00   8.00 - 7.00    2.00
##  4  2017  1.00  9.00      6.00   7.00 - 7.00  - 2.00
##  5  2016 12.0  16.0       6.00   6.00 - 6.00    4.00
##  6  2017  1.00  8.00      4.00   9.00 - 6.00  - 2.00
##  7  2016  2.00 12.0       2.00   1.00 - 6.00    6.00
```

```
##  8  2016  1.00 19.0      -2.00   0    - 6.00    1.00
##  9  2017  3.00 15.0       9.00  11.0  - 5.00   10.0
## 10  2017  3.00 11.0       7.00  10.0  - 5.00    9.00
## # ... with 721 more rows
```

3. Veriyi azalan tarihe göre sıralayın.

```
sehir_sicaklik %>%
    arrange(CEVAPBURAYA)
```

```
## # A tibble: 731 x 7
##      yil    ay   gun Amsterdam Londra    NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
##  1  2017  10.0  31.0      9.00   9.00  11.0   11.0
##  2  2017  10.0  30.0      8.00   6.00  12.0   13.0
##  3  2017  10.0  29.0     11.0   11.0   18.0    9.00
##  4  2017  10.0  28.0     12.0   10.0   17.0   10.0
##  5  2017  10.0  27.0     12.0    9.00  13.0   13.0
##  6  2017  10.0  26.0     13.0   10.0   13.0   13.0
##  7  2017  10.0  25.0     13.0   14.0   17.0   13.0
##  8  2017  10.0  24.0     13.0   16.0   21.0   13.0
##  9  2017  10.0  23.0     13.0   13.0   20.0   13.0
## 10  2017  10.0  22.0     11.0   11.0   19.0   13.0
## # ... with 721 more rows
```

4. En son olarak Londra ve Amsterdam'daki hava sıcaklıkları farkını artan bir şekilde sıralayın.

```
sehir_sicaklik %>%
    arrange(CEVAPBURAYA1 - CEVAPBURAYA2)
```

```
## # A tibble: 731 x 7
##      yil    ay   gun Amsterdam Londra    NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
##  1  2016 12.0  25.0      10.0    0     6.00    6.00
##  2  2015 12.0  25.0       9.00   0    17.0     4.00
##  3  2016  5.00 31.0      18.0   11.0  26.0    19.0
##  4  2016  6.00  1.00     19.0   12.0  24.0    17.0
##  5  2016  4.00 10.0      10.0    4.00  5.00   16.0
##  6  2016  6.00  7.00     20.0   14.0  24.0    22.0
##  7  2016  5.00  6.00     17.0   12.0  11.0    18.0
##  8  2016  5.00  8.00     21.0   16.0  14.0    17.0
##  9  2016  5.00 10.0      19.0   14.0  14.0    18.0
## 10  2016  6.00  3.00     16.0   11.0  19.0    19.0
## # ... with 721 more rows
```

mutate/transmute

`mutate` fonksiyonu sütunlar üzerinde işlem yapmaya yarar. `transmute` benzer bir işleve sahiptir ama `select` etkisi vardır, yani sadece işlem yaptığınız sütunları döndürür.

1. Venedik ve Amsterdam arasındaki sıcaklık farkını gün gün hesaplayın.

```
sehir_sicaklik %>%
    mutate(VAfark = CEVAPBURAYA1 - CEVAPBURAYA2)
```

```
## # A tibble: 731 x 8
##      yil    ay   gun Amsterdam Londra    NY Venedik VAfark
```

```
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>  <dbl>
##  1  2015  11.0  1.00      8.00   8.00  16.0   13.0    5.00
##  2  2015  11.0  2.00     10.0   11.0   15.0   10.0    0
##  3  2015  11.0  3.00      9.00  11.0   16.0    9.00   0
##  4  2015  11.0  4.00     12.0   11.0   17.0   10.0   -2.00
##  5  2015  11.0  5.00     13.0   13.0   18.0   12.0   -1.00
##  6  2015  11.0  6.00     16.0   14.0   21.0   13.0   -3.00
##  7  2015  11.0  7.00     16.0   14.0   17.0   14.0   -2.00
##  8  2015  11.0  8.00     12.0   12.0   11.0   13.0    1.00
##  9  2015  11.0  9.00     13.0   12.0   11.0   11.0   -2.00
## 10  2015  11.0 10.0      14.0   14.0   12.0   11.0   -3.00
## # ... with 721 more rows
```

2. Venedik'in Amsterdam'dan gün gün daha sıcak olup olmadığını hesaplayın.

```
sehir_sicaklik %>%
    mutate(VsicakA = CEVAPBURAYA1 > CEVAPBURAYA2)
```

```
## # A tibble: 731 x 8
##      yil    ay   gun Amsterdam Londra    NY Venedik VsicakA
##    <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl> <lgl>
##  1  2015  11.0  1.00      8.00   8.00  16.0   13.0  T
##  2  2015  11.0  2.00     10.0   11.0   15.0   10.0  F
##  3  2015  11.0  3.00      9.00  11.0   16.0    9.00 F
##  4  2015  11.0  4.00     12.0   11.0   17.0   10.0  F
##  5  2015  11.0  5.00     13.0   13.0   18.0   12.0  F
##  6  2015  11.0  6.00     16.0   14.0   21.0   13.0  F
##  7  2015  11.0  7.00     16.0   14.0   17.0   14.0  F
##  8  2015  11.0  8.00     12.0   12.0   11.0   13.0  T
##  9  2015  11.0  9.00     13.0   12.0   11.0   11.0  F
## 10  2015  11.0 10.0      14.0   14.0   12.0   11.0  F
## # ... with 721 more rows
```

3. Eğer Venedik Amsterdam'dan sıcak ise "daha sicak", değilse "daha soguk" yazın ve sadece tarih sütunları ile sıcak/soğuk bilgisini döndürün.

```
sehir_sicaklik %>%
    transmute(yil,ay,gun,
        VsicakA = ifelse(Venedik > Amsterdam,CEVAPBURAYA1, CEVAPBURAYA2))
```

```
## # A tibble: 731 x 4
##      yil    ay   gun VsicakA
##    <dbl> <dbl> <dbl> <chr>
##  1  2015  11.0  1.00 daha sicak
##  2  2015  11.0  2.00 daha soguk
##  3  2015  11.0  3.00 daha soguk
##  4  2015  11.0  4.00 daha soguk
##  5  2015  11.0  5.00 daha soguk
##  6  2015  11.0  6.00 daha soguk
##  7  2015  11.0  7.00 daha soguk
##  8  2015  11.0  8.00 daha sicak
##  9  2015  11.0  9.00 daha soguk
## 10  2015  11.0 10.0  daha soguk
## # ... with 721 more rows
```

group_by/summarise

`group_by` ve `summarise` özet tablolar (diğer ismiyle pivot tablo, özellikle Excel kullanıcıları bilir) yapmanıza yarar. summarise kendi başına da kullanılabilir veya `group_by` gibi bir gruplama fonksiyonuyla beraber de kullanılabilir. Bu kısım aynı zamanda birden fazla zincir (`%>%`) kullanmaya başlayacağınız ilk zincir ifadeniz olabilir.

İpucu: Eğer gruplamayı kaldırmak istiyorsanız sona `ungroup()` ekleştirin.

1. Venedik ve NY şehirlerinin ortalama sıcaklıklarını hesaplayın.

Calculate the mean Sicakliks of Venedik and NY of data period.

```
sehir_sicaklik %>%
    summarise(Venedik_ort=mean(CEVAPBURAYA1),NY_ort=CEVAPBURAYA2)
```

```
## # A tibble: 1 x 2
##   Venedik_ort NY_ort
##         <dbl>  <dbl>
## 1        14.3   14.4
```

2. Ay ay Amsterdam'ın ortalama sıcaklığını hesaplayın. Değeri 2 basamağa kadar yuvarlayın.

```
sehir_sicaklik %>%
    group_by(CEVAPBURAYA1) %>%
    summarise(Amsterdam_ort=mean(CEVAPBURAYA2))
```

```
## # A tibble: 12 x 2
##       ay Amsterdam_ort
##    <dbl>         <dbl>
##  1  1.00          3.00
##  2  2.00          4.32
##  3  3.00          6.92
##  4  4.00          8.43
##  5  5.00         14.5
##  6  6.00         17.3
##  7  7.00         18.0
##  8  8.00         17.7
##  9  9.00         16.0
## 10 10.0          11.7
## 11 11.0           7.65
## 12 12.0           6.97
```

3. Her yıl ve her ay için Amsterdam'ın NY'tan daha sıcak olduğu gün sayısını hesaplayın.

```
sehir_sicaklik %>%
    group_by(yil,ay) %>%
    summarise(AwarmerN_n=sum(CEVAPBURAYA1 > CEVAPBURAYA2))
```

```
## # A tibble: 24 x 3
## # Groups:   yil [?]
##      yil    ay AwarmerN_n
##    <dbl> <dbl>      <int>
##  1  2015 11.0          11
##  2  2015 12.0          12
##  3  2016  1.00         23
##  4  2016  2.00         16
##  5  2016  3.00          5
##  6  2016  4.00         10
```

```
##  7  2016  5.00          8
##  8  2016  6.00          1
##  9  2016  7.00          1
## 10  2016  8.00          0
## # ... with 14 more rows
```

4. Londra'nın maksimum, minimum ve medyan sıcaklık değerlerini her ay ve her yıl için hesaplayın.

```
sehir_sicaklik %>%
    group_by(yil,ay) %>%
    summarise(Londra_min=CEVAPBURAYA1,Londra_medyan=median(Londra),Londra_max=CEVAPBURAYA2)
```

```
## # A tibble: 24 x 5
## # Groups:   yil [?]
##      yil    ay Londra_min Londra_medyan Londra_max
##    <dbl> <dbl>      <dbl>         <dbl>      <dbl>
##  1  2015 11.0        1.00         11.0        14.0
##  2  2015 12.0        0            10.0        14.0
##  3  2016  1.00       0             6.00       11.0
##  4  2016  2.00       1.00          4.00       12.0
##  5  2016  3.00       2.00          6.00       11.0
##  6  2016  4.00       4.00          8.00       11.0
##  7  2016  5.00       8.00         13.0        16.0
##  8  2016  6.00      11.0          16.0        19.0
##  9  2016  7.00      14.0          18.0        24.0
## 10  2016  8.00      14.0          18.0        24.0
## # ... with 14 more rows
```

İleri Örnekler

Aşağıda tidyverse veri manipülasyonu ile yapabileceğiniz bazı ileri fonksiyonlar da bulunmaktadır.

Lead ve Lag

Diyelim ki arka arkaya satırların arasındaki farkı bilmek istiyorsunuz. O zaman `lag` ve `lead` fonksiyonlarını kullanabilirsiniz. Diyelim ki Amsterdam'ın önceki ve sonraki sıcaklıklara göre farklarını bulmak istiyorum.

```
sehir_sicaklik %>%
    transmute(yil,ay,gun,Amsterdam,A_prev=lag(Amsterdam),A_next=lead(Amsterdam),
              A_prev_diff=Amsterdam-A_prev,A_next_diff=Amsterdam-A_next)
```

```
## # A tibble: 731 x 8
##      yil    ay   gun Amsterdam A_prev A_next A_prev_diff A_next_diff
##    <dbl> <dbl> <dbl>     <dbl>  <dbl>  <dbl>       <dbl>       <dbl>
##  1  2015  11.0  1.00      8.00  NA     10.0        NA         -2.00
##  2  2015  11.0  2.00     10.0    8.00   9.00        2.00        1.00
##  3  2015  11.0  3.00      9.00  10.0   12.0       - 1.00       -3.00
##  4  2015  11.0  4.00     12.0    9.00  13.0         3.00       -1.00
##  5  2015  11.0  5.00     13.0   12.0   16.0         1.00       -3.00
##  6  2015  11.0  6.00     16.0   13.0   16.0         3.00        0
##  7  2015  11.0  7.00     16.0   16.0   12.0         0           4.00
##  8  2015  11.0  8.00     12.0   16.0   13.0       - 4.00       -1.00
##  9  2015  11.0  9.00     13.0   12.0   14.0         1.00       -1.00
```

```
## 10  2015  11.0 10.0      14.0   13.0   13.0         1.00        1.00
## # ... with 721 more rows
```

slice

Slice fonksiyonu verilen indekslerdeki satırları döndürür.

```
sehir_sicaklik %>%
    slice(1:3)
```

```
## # A tibble: 3 x 7
##     yil    ay   gun Amsterdam Londra    NY Venedik
##   <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
## 1  2015  11.0  1.00      8.00   8.00  16.0   13.0
## 2  2015  11.0  2.00     10.0   11.0   15.0   10.0
## 3  2015  11.0  3.00      9.00  11.0   16.0    9.00
```

`group_by` fonksiyonu ile birlikte de kullanılabilir.

```
sehir_sicaklik %>%
    group_by(yil) %>%
    slice(1:3)
```

```
## # A tibble: 9 x 7
## # Groups:   yil [3]
##     yil    ay   gun Amsterdam Londra    NY Venedik
##   <dbl> <dbl> <dbl>     <dbl>  <dbl> <dbl>   <dbl>
## 1  2015 11.0   1.00      8.00   8.00 16.0    13.0
## 2  2015 11.0   2.00     10.0   11.0  15.0    10.0
## 3  2015 11.0   3.00      9.00  11.0  16.0     9.00
## 4  2016  1.00  1.00      4.00   3.00  3.00    2.00
## 5  2016  1.00  2.00      6.00  10.0   2.00    0
## 6  2016  1.00  3.00      7.00   8.00  4.00    3.00
## 7  2017  1.00  1.00      1.00   7.00  7.00    2.00
## 8  2017  1.00  2.00      3.00   2.00  3.00    1.00
## 9  2017  1.00  3.00      4.00   2.00  5.00    3.00
```

Unutmayın ki slice sadece indeks değerini verir ve bu değişebilir.

Gather (toplama) ve Spread (yayma)

Verinizi geniş (çok sütunlu) formattan uzun (az sütun çok satırlı) formata veya tersine dönüştürmek isteyebilirsiniz. Bunlara eritme (melting) ve casting (şekillendirme) de denir. O zaman `gather` ve `spread` fonksiyonlarını kullanabilirsiniz. Başta alışması biraz zaman alır ama kolayca yapabilecek bir hale gelirsiniz.

Diyelim ki her şehrin her ayki ortalama sıcaklıklarını göreceğimiz bir özet tablo istiyoruz. Ama şehirler satırlarda ve aylar sütunlarda olsun.

```
# Veriyi toplayıp uzun formata çevirin
# Ama tarih sütunlarını eklemeyin
sehir_sicaklik_uzun <-
sehir_sicaklik %>%
    gather(key=Sehir,value=Sicaklik,-yil,-ay,-gun)

sehir_sicaklik_uzun
```

```
## # A tibble: 2,924 x 5
##      yil    ay   gun Sehir     Sicaklik
##    <dbl> <dbl> <dbl> <chr>        <dbl>
##  1  2015  11.0  1.00 Amsterdam     8.00
##  2  2015  11.0  2.00 Amsterdam    10.0
##  3  2015  11.0  3.00 Amsterdam     9.00
##  4  2015  11.0  4.00 Amsterdam    12.0
##  5  2015  11.0  5.00 Amsterdam    13.0
##  6  2015  11.0  6.00 Amsterdam    16.0
##  7  2015  11.0  7.00 Amsterdam    16.0
##  8  2015  11.0  8.00 Amsterdam    12.0
##  9  2015  11.0  9.00 Amsterdam    13.0
## 10  2015  11.0 10.0  Amsterdam    14.0
## # ... with 2,914 more rows
```

```
# Simdi group_by ve summarise ile ortalama sicaklik degerlerini her sehir ve ay icin alalim
sehir_sicaklik_uzun %>%
    group_by(ay,Sehir) %>%
    summarise(temp_avg=round(mean(Sicaklik))) %>%
    #Simdi aylari sutunlara dagitalim
    spread(ay,temp_avg)
```

```
## # A tibble: 4 x 13
##   Sehir       `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`  `10`
## * <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Amsterdam  3.00  4.00  7.00  8.00  14.0  17.0  18.0  18.0  16.0  12.0
## 2 Londra     4.00  6.00  8.00  9.00  13.0  17.0  18.0  18.0  16.0  12.0
## 3 NY         2.00  4.00  7.00 13.0   17.0  22.0  26.0  25.0  22.0  16.0
## 4 Venedik    2.00  7.00 11.0  14.0   18.0  22.0  25.0  25.0  20.0  14.0
##    `11`  `12`
## * <dbl> <dbl>
## 1  8.00  7.00
## 2  9.00  8.00
## 3 11.0   7.00
## 4  9.00  5.00
```

_all ve _at ekleri

Ozellikle `mutate` ve `summarise` fonksiyonlarinin "all" ve "at" ekleri olan varyasyonlari bulunmaktadir.

Butun şehirlerin ortalama sıcaklıklarını hesaplayalım. Bunu iki türlü yapabiliriz. Önce şehirleri seçer ve `summarise_all` deriz veya `summarise_at` ile ilgili şehirleri seçeriz.

```
#Metod 1
sehir_sicaklik %>%
    select(Amsterdam:Venedik) %>%
    summarise_all(funs(round(mean(.))))
```

```
## # A tibble: 1 x 4
##   Amsterdam Londra    NY Venedik
##       <dbl>  <dbl> <dbl>   <dbl>
## 1      11.0   12.0  14.0    14.0
```

```
#Metod 2
sehir_sicaklik %>%
    summarise_at(vars(Amsterdam:Venedik),funs(round(mean(.))))
```

```
## # A tibble: 1 x 4
##   Amsterdam Londra    NY Venedik
##       <dbl>  <dbl> <dbl>   <dbl>
## 1      11.0   12.0  14.0    14.0
```

mutate_at fonksiyonunu kullanarak diğer bütün şehirlerin NY şehrinden sıcaklık olarak farkını bulabiliriz.

```
sehir_sicaklik %>%
    mutate_at(vars(Amsterdam,Londra,Venedik),funs(diff_NY=abs(NY-.))) %>%
    select(-Amsterdam,-Londra,-Venedik)
```

```
## # A tibble: 731 x 7
##      yil    ay   gun    NY Amsterdam_diff_NY Londra_diff_NY
##    <dbl> <dbl> <dbl> <dbl>             <dbl>          <dbl>
##  1  2015  11.0  1.00  16.0              8.00           8.00
##  2  2015  11.0  2.00  15.0              5.00           4.00
##  3  2015  11.0  3.00  16.0              7.00           5.00
##  4  2015  11.0  4.00  17.0              5.00           6.00
##  5  2015  11.0  5.00  18.0              5.00           5.00
##  6  2015  11.0  6.00  21.0              5.00           7.00
##  7  2015  11.0  7.00  17.0              1.00           3.00
##  8  2015  11.0  8.00  11.0              1.00           1.00
##  9  2015  11.0  9.00  11.0              2.00           1.00
## 10  2015  11.0 10.0   12.0              2.00           2.00
##    Venedik_diff_NY
##              <dbl>
##  1            3.00
##  2            5.00
##  3            7.00
##  4            7.00
##  5            6.00
##  6            8.00
##  7            3.00
##  8            2.00
##  9            0
## 10            1.00
## # ... with 721 more rows
```

Son Alıştırmalar

Aşağıdaki alıştırmalar öğrencilerin kendilerinin yapması için bırakılmıştır. Sonuçları tekrarlayacak kodlar yazmaya çalışın.

1. Amsterdam'ın Londra'dan daha sıcak ama Venedik'ten daha soğuk olduğu günleri getirin.

```
## # A tibble: 165 x 7
##      yil    ay   gun Amsterdam Londra     NY Venedik
##    <dbl> <dbl> <dbl>     <dbl>  <dbl>  <dbl>   <dbl>
##  1  2015 11.0  21.0       5.00   3.00   9.00    8.00
##  2  2015 11.0  22.0       3.00   1.00   9.00    8.00
##  3  2016  1.00 13.0       4.00   3.00 - 3.00    6.00
##  4  2016  1.00 16.0       2.00   1.00   8.00    4.00
##  5  2016  2.00  3.00      5.00   4.00  11.0     8.00
##  6  2016  2.00 11.0       4.00   3.00 - 4.00    7.00
##  7  2016  2.00 12.0       2.00   1.00 - 6.00    6.00
```

```
##  8  2016  2.00 23.0       4.00   3.00   3.00   11.0
##  9  2016  2.00 24.0       2.00   1.00   9.00   10.0
## 10  2016  2.00 25.0       2.00   1.00   9.00    8.00
## # ... with 155 more rows
```

2. Her yılın her ayı için NY'nin Amsterdam'dan sıcak olduğu günlerde NY ve Amsterdam arasındaki ortalama sıcaklık farkını hesaplayın ve 1 basamağa yuvarlayın. En yüksek farktan en düşüğe sıralayın.

```
## # A tibble: 24 x 3
## # Groups:   yil [3]
##      yil    ay NYwA_diff
##    <dbl> <dbl>     <dbl>
##  1  2016  8.00      8.40
##  2  2016  7.00      8.10
##  3  2017  9.00      7.90
##  4  2016  4.00      7.50
##  5  2017  4.00      7.40
##  6  2017  7.00      7.30
##  7  2017  8.00      6.50
##  8  2016 11.0       6.40
##  9  2016  3.00      6.30
## 10  2016  6.00      6.00
## # ... with 14 more rows
```

3. Her gün için en sıcak şehiri ve sıcaklık değerini döndürün.

```
## # A tibble: 731 x 5
## # Groups:   yil, ay, gun [731]
##      yil    ay   gun Sehir     Sicaklik
##    <dbl> <dbl> <dbl> <chr>        <dbl>
##  1  2015  11.0  1.00 NY            16.0
##  2  2015  11.0  2.00 NY            15.0
##  3  2015  11.0  3.00 NY            16.0
##  4  2015  11.0  4.00 NY            17.0
##  5  2015  11.0  5.00 NY            18.0
##  6  2015  11.0  6.00 NY            21.0
##  7  2015  11.0  7.00 NY            17.0
##  8  2015  11.0  8.00 Venedik       13.0
##  9  2015  11.0  9.00 Amsterdam     13.0
## 10  2015  11.0 10.0  Amsterdam     14.0
## # ... with 721 more rows
```