R ile Veri Analizine Giriş: 2019 Yılı PTF - SMF Verisi

Baran Dogru

1/7/2020

Contents

1 Giriş **1**

1.1 Amaç 1

1.2 Dökümanda Kullanılan Veri Seti 1

2 dplyr ile Veri Manipülasyonu **2**

2.1 Hazırlıklar 2

2.2 Önemli Fonksiyonlar 3

3 ggplot2 ile Veri Görselleştirme **7**

3.1 Hazırlıklar 7

3.2 Saçılım Grafiği (Scatter Plot) 8

3.3 Sütunlu Grafik (Bar Chart) 12

3.4 Çizgi Grafiği (Line Chart) 16

Appendix **19**

RStudio Cloud'da Çalışma 19

Veri Seti Düzenlemeleri 22

1 Giriş

1.1 Amaç

Bu dökümanın temel amacı, `R` yazılım dilini ve spesifik olarak `dplyr`, `ggplot2`, `lubridate` ve `markdown` gibi paket ve araçları kullanarak Elektrik Piyasasında yapılabilecek analizlere örnekler göstermek ve bu sektörde çalışan insanlara destek olmaktır.

1.2 Dökümanda Kullanılan Veri Seti

Bu dökümanda kullanılan veri seti EPİAŞ'ın Raporlama Sayfasından elde edilen 01.01.2019 - 31.12.2019 tarihli Piyasa Takas Fiyatı (PTF) - Sistem Marjinal Fiyatı (SMF) verisidir.

Veri setinin ham haline EPİAŞ'ın Raporlama Sayfası 'ndan ya da buraya tıklayarak ulaşabilirsiniz.

```
# yukarida buraya yazan yere ham verinin .xls hali linklenecek
```

Bu veri setinin ham halinin örneklerde rahatlıkla kullanılabilmesi adına bazı format ve isim değişlikleri yapılmıştır ve bu yazının kapsamını aştığından bu dönüşümlerin nasıl yapıldığı anlatılmayacaktır. İlgilenenler için ise bu kodlar Veri Seti Düzenlemeleri kısmında bulunabilir.

Veri setinin örneklere uygun şekilde hazırlanmış haline ise aşağıdaki kodu R'da çalıştırarak ulaşabilirsiniz.

```
# gerekli duzenlemelerin yapildigi veri setini rds olarak indirmelerini saglayacak kod
```

Bu aşamada veri seti okunduktan sonra geri kalan bütün hazırlıklar her bölümde ayrı ayrı detaylı bir şekilde anlatılacaktır. Örneklerle ilgilenmeye başlamadan önce çalışma dizininizin doğru yerde belirlendiğinden emin olmak adına `getwd()` komutunu konsola yazarak üzerinde çalıştığınız belgenin bilgisayarınızda kayıtlı olduğu yerle uyumlu olduğunu kontrol edebilirsiniz.

2 dplyr ile Veri Manipülasyonu

Bu bölümde hedeflenen şey, çok önemli ve oldukça güçlü bir data manipülasyonu paketi olan `dplyr` paketinin önemli fonksiyonlarına dair örnekler vererek nasıl kullanılabileceklerini göstermektir. Bu bölümde ele alınacak fonksiyonlar şu şekildedir:

`select/rename`
`filter`
`distinct`
`arrange`
`mutate/transmute`
`group_by/summarise`

Yukarıda bahsedilen bütün fonksiyonlar ayrı ayrı ele alınacak ve her biri için örnek kullanımlar gösterilecektir. Ayrıca "pipe operator" olarak adlandırılan "Bağlantı Operatörü" (`%>%`) de kısaca anlatılacak ve bütün döküman boyunca kullanılacaktır.

2.1 Hazırlıklar

Bu bölüme başlamadan önce yapılması gereken şey `dplyr` paketini indirmek ve paketi yüklemek. Paketi indirmek için `install.packages("dplyr")` , paketi yüklemek için ise `library(dplyr)` komutları kullanılabilir.

Bu bölümde devam etmeden önce veri setini indirdiğinizden emin olun. Henüz indirmediyseniz Veri Seti bölümünü ziyaret ediniz.

```
# dplyr paketinin indirilmesi
install.packages("dplyr")
# dplyr paketinin yüklenmesi
library(dplyr)
```

Tablonun ilk haline `glimpse()` fonksiyonu ile göz atılabilir.

```
ptfsmf %>% glimpse()
```

```
## Observations: 8,760
## Variables: 5
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ PTF   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
```

```
## $ SMF   <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ PDF   <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ NDF   <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.5...
```

Burada "Bağlantı Operatörü"nün (`%>%`) kullanımına dikkat edilmelidir. Bu operatör sayesinde kullanılan tablo tek bir kez yazılarak istenildiği kadar işlem yapılabilir ve bu sayede daha tertipli bir koda ulaşılabilir.

2.2 Önemli Fonksiyonlar

2.2.1 `select/rename`

`select` fonksiyonu belirli sutünları seçmek için kullanılır. Örneğin tablonun sadece Tarih ve Piyasa Takas Fiyatı'nı (PTF) göstermesi isteniyorsa,

```
ptfsmf %>% select(Tarih, PTF) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 2
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ PTF   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
```

Veya (bu örnek için pek faydalı gibi görünmese de) içinde "P" harfini barındıran sütunlar seçilmek istendiğinde `contains` kelimesi kullanılabilir.

```
ptfsmf %>% select(contains("P")) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 2
## $ PTF <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.50,...
## $ PDF <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0...
```

Bu kelime gibi `select` fonksiyonu içinde kullanılabilecek diğer kelimeler: `starts_with`, `ends_with`, `matches` olarak sıralanabilir.

Aynı zamanda, eğer sütunların sırası biliniyorsa iki sütun arasındaki her sütunu seçmek için `:` operatörü kullanılabilir. Örneğin Sistem Marjinal Fiyatı (SMF) ve Negatif Dengesizlik Fiyatı (NDF) arasındaki sütunları seçmek için,

```
ptfsmf %>% select(SMF:NDF) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 3
## $ SMF <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0...
## $ PDF <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0...
## $ NDF <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.53,...
```

`rename` fonksiyonu ise adından da anlaşılabileceği üzere bir sütunun adını değiştirmek için kullanılır. Örneğin "PTF" sütununun ismini "ptf" olarak değiştirmek isteniyorsa `rename` kullanılabilir.

```
ptfsmf %>% rename(ptf = PTF) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 5
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ ptf   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
## $ SMF   <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ PDF   <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ NDF   <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.5...
```

2.2.2 `filter`

`filter` fonksiyonu temel olarak istenilen koşulları sağlayan satırları seçmek için kullanılır. Örneğin Piyasa Takas Fiyatı'nın 250'den düşük olduğu satırlar incelenebilir.

```
ptfsmf %>% filter(PTF < 250) %>% glimpse()
```

```
## Observations: 2,453
## Variables: 5
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ PTF   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
## $ SMF   <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ PDF   <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ NDF   <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.5...
```

Gözlem sayısının 8670'den 2453'e düştüğü görülebiliyor. Aynı şekilde bir değerin büyüklüğü `>`, küçüklük veya eşitliği `<=`, büyüklük veya eşitliği `>=`, eşitliği ise `==` sembolleriyle incelenebilir.

Birden fazla koşulun sağlanması gerekiyorsa ise, "VE" operatörü için `&`, "VEYA" operatörü için `|` sembolleri kullanılmalıdır. Örneğin Pozitif Dengesizlik Fiyatı'nın 200'den küçük, Negatif Dengesizlik Fiyatı'nın ise 200'den büyük olduğu satırlar incelenebilir.

```
ptfsmf %>% filter(PDF<200 & NDF>200) %>% glimpse()
```

```
## Observations: 1,256
## Variables: 5
## $ Tarih <dttm> 2019-01-01 17:00:00, 2019-01-01 18:00:00, 2019-01-01 21:00:0...
## $ PTF   <dbl> 286.93, 291.52, 285.18, 204.89, 287.66, 296.99, 296.86, 293.1...
## $ SMF   <dbl> 172.00, 172.00, 173.00, 172.98, 168.60, 173.00, 190.00, 190.0...
## $ PDF   <dbl> 166.84, 166.84, 167.81, 167.79, 163.54, 167.81, 184.30, 184.3...
## $ NDF   <dbl> 295.54, 300.27, 293.74, 211.04, 296.29, 305.90, 305.77, 301.9...
```

Bu koşulları sağlayan 1256 satır olduğu görülüyor. Eğer bir koşulu sağlamayan satırlar aranıyorsa ise koşulun başına `!` yazılmasıyla bu ters etki taratılabilir.

2.2.3 `distinct`

`distinct` fonksiyonu bir veya birden fazla sütunu baz alarak özgün satırları bulmak için kullanılır. Örneğin var olan tabloda her bir saat için 2 tane satır var ve bunlardan kurtulmak isteniyor. Öncelikle tabloya göz atılırsa,

```
ptfsmf %>% head(8) # head fonksiyonu da print ve glipmse fonksiyonlari ile benzer islevde
```

```
## # A tibble: 8 x 5
##   Tarih                PTF   SMF   PDF   NDF
##   <dttm>             <dbl> <dbl> <dbl> <dbl>
## 1 2019-01-01 00:00:00 100.    5    4.85 103.
## 2 2019-01-01 00:00:00 100.    5    4.85 103.
## 3 2019-01-01 01:00:00  96.7  95.0 92.2   99.6
## 4 2019-01-01 01:00:00  96.7  95.0 92.2   99.6
## 5 2019-01-01 02:00:00  81.6  79.6 77.2   84.0
## 6 2019-01-01 02:00:00  81.6  79.6 77.2   84.0
## 7 2019-01-01 03:00:00  38.6   0    0     39.7
## 8 2019-01-01 03:00:00  38.6   0    0     39.7
```

Bu durumdan kurtulmak için `distinct` fonksiyonu kullanılabilir. Örneğin,

```
ptfsmf %>% distinct(Tarih) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 1
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
```

Dikkat edileceği üzere yalnızca Tarih sütunu korundu. Diğer sütunları da korumak için `.keep_all = TRUE` komutu kullanılmalıdır.

```
ptfsmf %>% distinct(Tarih, .keep_all = TRUE) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 5
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ PTF   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
## $ SMF   <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ PDF   <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ NDF   <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.5...
```

2.2.4 arrange

`arrange` fonksiyonu Excel'deki sıralama özelliğine benzetilebilir. Varsayılan durumunda A'dan Z'ye ya da küçükten büyüğe sıralanır. Tam tersi sıralama için `desc()` fonksiyonu kullanılmalıdır. Örneğin artan Piyasa Takas Fiyatlarına göre sıralamak için,

```
ptfsmf %>% arrange(PTF) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 5
## $ Tarih <dttm> 2019-02-17 09:00:00, 2019-03-24 12:00:00, 2019-03-24 13:00:0...
## $ PTF   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ SMF   <dbl> 30, 5, 0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ PDF   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ NDF   <dbl> 30.90, 5.15, 0.00, 0.00, 0.00, 10.30, 0.00, 0.00, 0.00, 0.00,...
```

Bir başka örnekte Tarih, Pozitif Dengesizlik Fiyatı (PDF) ve Negatif Dengesizlik Farkı (NDF) sütunları seçilip azalan NDF'ye göre sıralanabilir.

```
ptfsmf %>% select(Tarih, PDF, NDF) %>%
            arrange(desc(NDF)) %>%
            glimpse()
```

```
## Observations: 8,760
## Variables: 3
## $ Tarih <dttm> 2019-06-27 15:00:00, 2019-06-27 16:00:00, 2019-06-27 14:00:0...
## $ PDF   <dbl> 436.50, 436.50, 453.74, 385.24, 303.44, 307.22, 369.73, 339.5...
## $ NDF   <dbl> 515.00, 514.99, 502.40, 501.76, 462.47, 449.82, 438.96, 432.6...
```

Dikkat edileceği üzere NDF değerleri azalarak devam ediyor. Ayrıca `arrange` fonksiyonunun içine birden fazla değer girerek ilk değerin eşitliği durumunda ikinci değerin karar vermesi sağlanabilir.

2.2.5 `mutate/transmute`

`mutate` fonksiyonu genellikle var olan değişkenlerle yapılan operasyonlar sonucu yeni değişkenler (sütunlar) yatarmak için kullanılır. Örneğin Pozitif Dengesizlik Fiyatı (PDF) ve Negatif Dengesizlik Fiyatı (NDF) arasındaki Dengesizlik Fiyatı (DF) sütununda gösterilmek istenirse,

```
ptfsmf %>% mutate(DF = PDF - NDF) %>% glimpse()
```

```
## Observations: 8,760
## Variables: 6
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ PTF   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
## $ SMF   <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ PDF   <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ NDF   <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.5...
## $ DF    <dbl> -98.54, -7.43, -6.84, -39.74, -11.87, -11.47, -11.47, -25.10,...
```

Bu fonksiyonun içerisinde base R fonksiyonları da kullanılabilir. Örneğin yeni bir Denge sütununda PDF NDF'den büyükse "Pozitif", tam tersiyse "Negatif, eşitlik varsa ise"Dengeli" yanıtı alabilmek için,

```
ptfsmf %>% mutate(Denge = ifelse(PDF>NDF,"Pozitif","Negatif"),
                  Denge = ifelse(PDF==NDF,"Dengeli",Denge)) %>%
            glimpse()
```

```
## Observations: 8,760
## Variables: 6
## $ Tarih <dttm> 2019-01-01 00:00:00, 2019-01-01 01:00:00, 2019-01-01 02:00:0...
## $ PTF   <dbl> 100.38, 96.72, 81.60, 38.58, 11.52, 11.14, 11.14, 24.37, 34.5...
## $ SMF   <dbl> 5.00, 95.04, 79.60, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ PDF   <dbl> 4.85, 92.19, 77.21, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,...
## $ NDF   <dbl> 103.39, 99.62, 84.05, 39.74, 11.87, 11.47, 11.47, 25.10, 35.5...
## $ Denge <chr> "Negatif", "Negatif", "Negatif", "Negatif", "Negatif", "Negat...
```

`transmute` fonksiyonu da `mutate` fonksiyonu ile benzer bir işleve sahip olmasının yanında `select` fonksiyonunun belirli sütunları seçme işlevine de sahiptir. Örneğin PTF/SMF oranını Tarih'in de gösterileceği şekilde sergileyip bu orana göre artan şekilde sıralamak için,

```
ptfsmf %>% transmute(Tarih, PTF, SMF, PTF_SMF_Oran = PTF/SMF) %>%
            arrange(PTF_SMF_Oran) %>%
            glimpse()
```

```
## Observations: 8,760
## Variables: 4
## $ Tarih        <dttm> 2019-02-17 09:00:00, 2019-03-24 12:00:00, 2019-03-24 ...
## $ PTF          <dbl> 0.00, 0.00, 0.00, 2.00, 1.00, 0.99, 0.99, 1.00, 1.01, ...
## $ SMF          <dbl> 30.00, 5.00, 10.00, 244.00, 100.00, 90.99, 75.99, 76.0...
## $ PTF_SMF_Oran <dbl> 0.000000000, 0.000000000, 0.000000000, 0.008196721, 0....
```

2.2.6 group_by/summarise

Bu iki fonksiyon çoğunlukla özetleme tabloları çıkarmak için kullanılır. `group_by` sütunlara göre veriyi gruplamak, `summarise` ise gruplanan bu verilere göre istenen özeti çıkarmakla görevlidir. Örneğin günlük ortalama PTF fiyat bilgisi için (Burada "POSIX" formatında olan Tarih sütununu "Date" formatına çevirebilmek için as.Date() fonksiyonu kullanılmıştır.),

```
gunluk_ort_ptf <- ptfsmf %>%
                  mutate(Gun = as.Date(Tarih)) %>%
                  group_by(Gun) %>%
                  summarise(Gunluk_Ort_PTF = mean(PTF)) %>%
                  glimpse()
```

```
## Observations: 365
## Variables: 2
## $ Gun            <date> 2019-01-01, 2019-01-02, 2019-01-03, 2019-01-04, 201...
## $ Gunluk_Ort_PTF <dbl> 121.0229, 228.7592, 238.6304, 212.3496, 244.3354, 23...
```

Burada kullanılan `mean` fonksiyonu dışında maksimum değer için `max`, minimum değer için `min`, medyan için `median`, total değer için `sum` ve grupladığımız değişkene ait gözlem sayısını saymak için ise `n()`fonksiyonları kullanılabilir.

3 ggplot2 ile Veri Görselleştirme

Bu bölümde `ggplot2` paketinden yararlanarak veri görselleştirmenin nasıl yapılabileceği farklı fonksiyonlar ve örnekler üzerinden gösterilecektir.

3.1 Hazırlıklar

Bu bölüme başlamadan önce yapılması gereken `ggplot2` paketini indirmek ve yüklemek. Paketi indirmek için `install.packages("ggplot2")` , paketi yüklemek için ise `library(ggplot2)` komutları kullanılabilir. (Veri setini indirmediyseniz lütfen Veri Seti bölümünü ziyaret ediniz.)

```
# ggplot2 paketinin indirilmesi
install.packages("ggplot2")
# ggplot2 paketinin yüklenmesi
library(ggplot2)
```

Bu bölümde daha ayrıntılı renklendirmelerin nasıl yapılabileceğinin gösterilebilmesi adına veri seti üzerinde bazı düzenlemeler gerekiyor. Örneğin 2019 Haziran ayı için saatlere göre Piyasa Takas Fiyatı (PTF) ve Sistem Marjinal Fiyatı (SMF) incelenecek olursa (Dönüşüm aşamasında kullanılan `pivot_longer` fonksiyonu, bu dökümanın kapsamı dışında olduğundan burada açıklanmayacaktır.),

```
plot_df <- ptfsmf %>% mutate(Saat = hour(Tarih), Tarih = as.Date(Tarih)) %>%
                      filter(Tarih>"2019-05-31" & Tarih<"2019-07-01") %>%
                      group_by(Saat) %>%
                      summarise(PTF = mean(PTF), SMF = mean(SMF)) %>%
                      ungroup() %>%
                      pivot_longer(cols=PTF:SMF, names_to="Piyasa_Degiskeni", values_to="Fiyatlar") %>%
                      glimpse()
```

```
## Observations: 48
## Variables: 3
## $ Saat             <int> 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8,...
## $ Piyasa_Degiskeni <chr> "PTF", "SMF", "PTF", "SMF", "PTF", "SMF", "PTF", "...
## $ Fiyatlar         <dbl> 230.58533, 176.08567, 253.01567, 216.25033, 236.95...
```

Bu dönüşümleri uygulamadan bu bölümde kullanılacak olan veri setine erişmek isterseniz aşağıda gördüğünüz kodu çalıştırabilirsiniz.

```
# plot datasini rds olarak indirebilecekleri kod
```

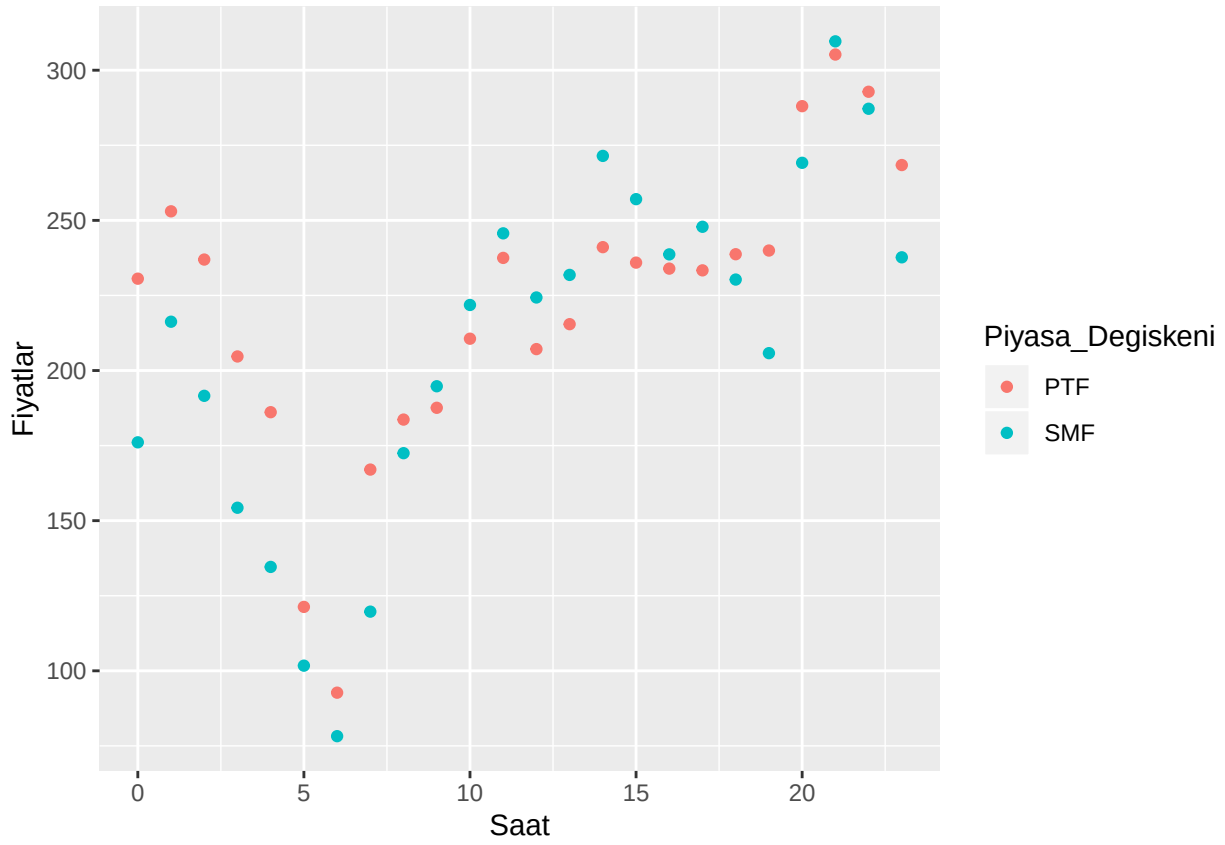
`dplyr` paketinde birden fazla fonksiyonu aynı veri setine uygulamak için "Bağlantı Operatörü" (`%>%`) kullanılırken `ggplot2` paketinde ise birden fazla etkiyi aynı ayna uygulamak için (`+`) operatörü kullanılıyor. Bu durum başlangıç grafiğinin üzerine farklı görseller eklenmesi olarak düşünülebilir.

3.2 Saçılım Grafiği (Scatter Plot)

İlk olarak bahsedilecek grafik türü olan "Scatter Plot" temel olarak var olan verinin noktasal dağılımını göstermek için kullanılır. Burada `ggplot` fonksiyonu ile de ilk kez karşılaşıldığından bazı özelliklerinden bahsedilmelidir. Fonksiyonun içine ilk yazılan parametre kullanılacak verinin ismidir. Daha sonra `aes()` yardımıyla grafiğin x ve y koordinatları belirlenir. `aes()` içerisinde ayrıca `color`, `fill`, `alpha`, `shape`, `size` gibi özellikler de belirlenebilir.

"Scatter Plot" için kullanılan fonksiyon ise `geom_point()`tir. İlk grafik şu şekilde ortaya çıkacaktır.

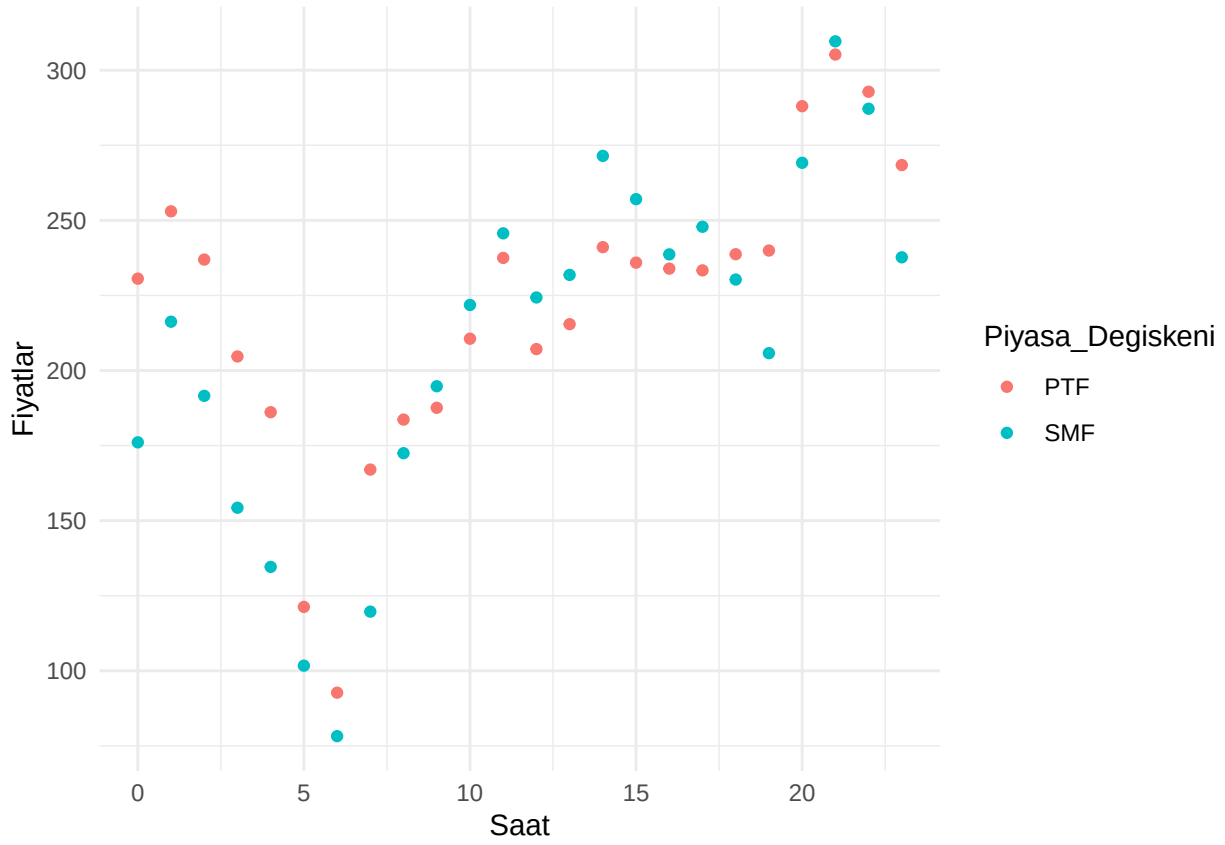
```
sc_plot <- ggplot(plot_df, aes(x=Saat, y=Fiyatlar, color=Piyasa_Degiskeni)) +
           geom_point()
sc_plot
```



Görülebileceği üzere öncelikle veri seti, sonrasında ise `aes()` içerisinde x ve koordinatları ile renk belirlendi.

Grafiği görsel olarak daha güzel hale getirmek için öncelikle arka plandaki gri kısım `theme_minimal()` fonksiyonu yardımıyla kaldırılabilir.

```
sc_plot <- sc_plot + theme_minimal()
sc_plot
```

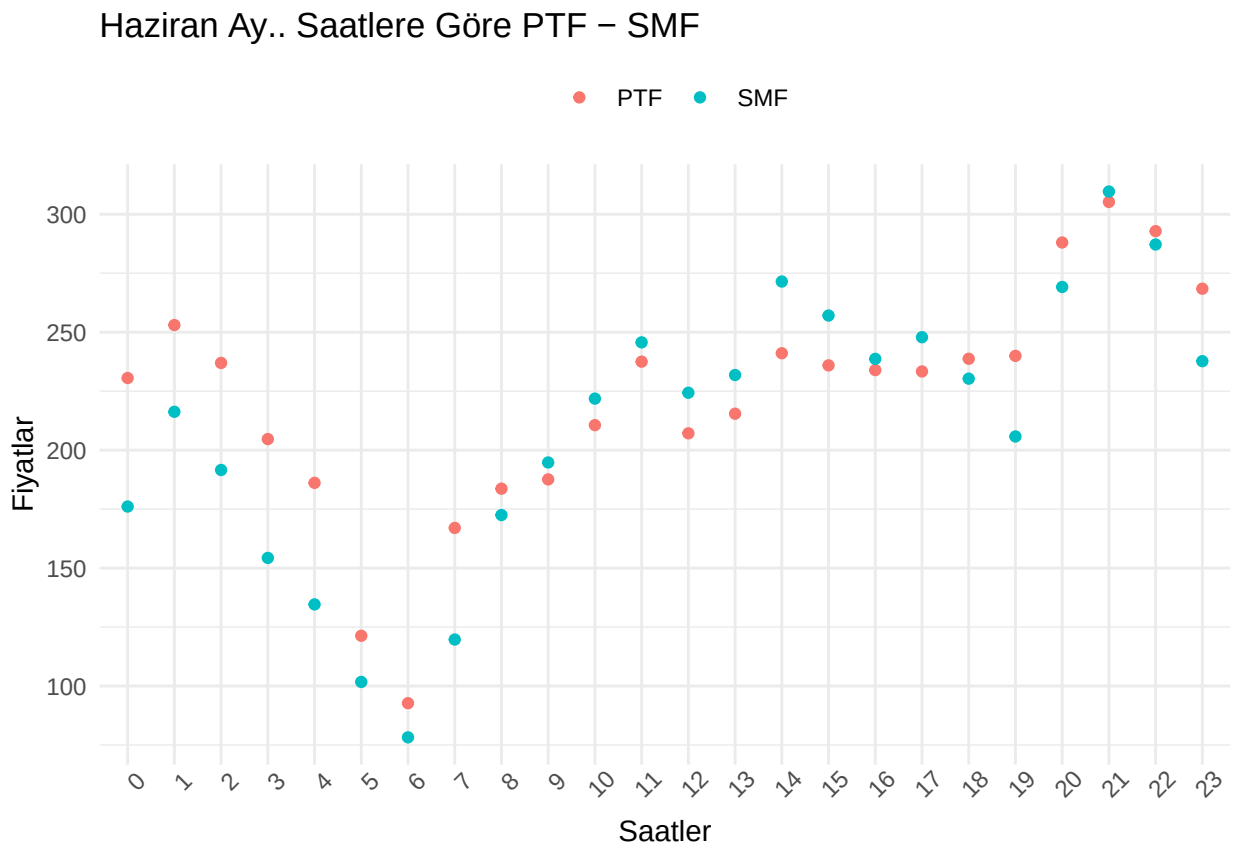


Grafiğin ve eksenlerinin ismini değiştirmek, eksenlerdeki numaralandırma sıklığını ayarlamak, x eksenindeki yazıları döndürmek ve lejantı istenilen şekilde konumlamak için aşağıdakiler uygulanmalıdır.

```
sc_plot <- sc_plot +
            labs(title = "Haziran Ayı Saatlere Göre PTF - SMF",
                 x = "Saatler",
                 y = "Fiyatlar") +
            scale_x_discrete(limits=c(0:23)) +
            theme(axis.text.x = element_text(angle = 45), legend.position = "top",
                  legend.title = element_blank())
sc_plot
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

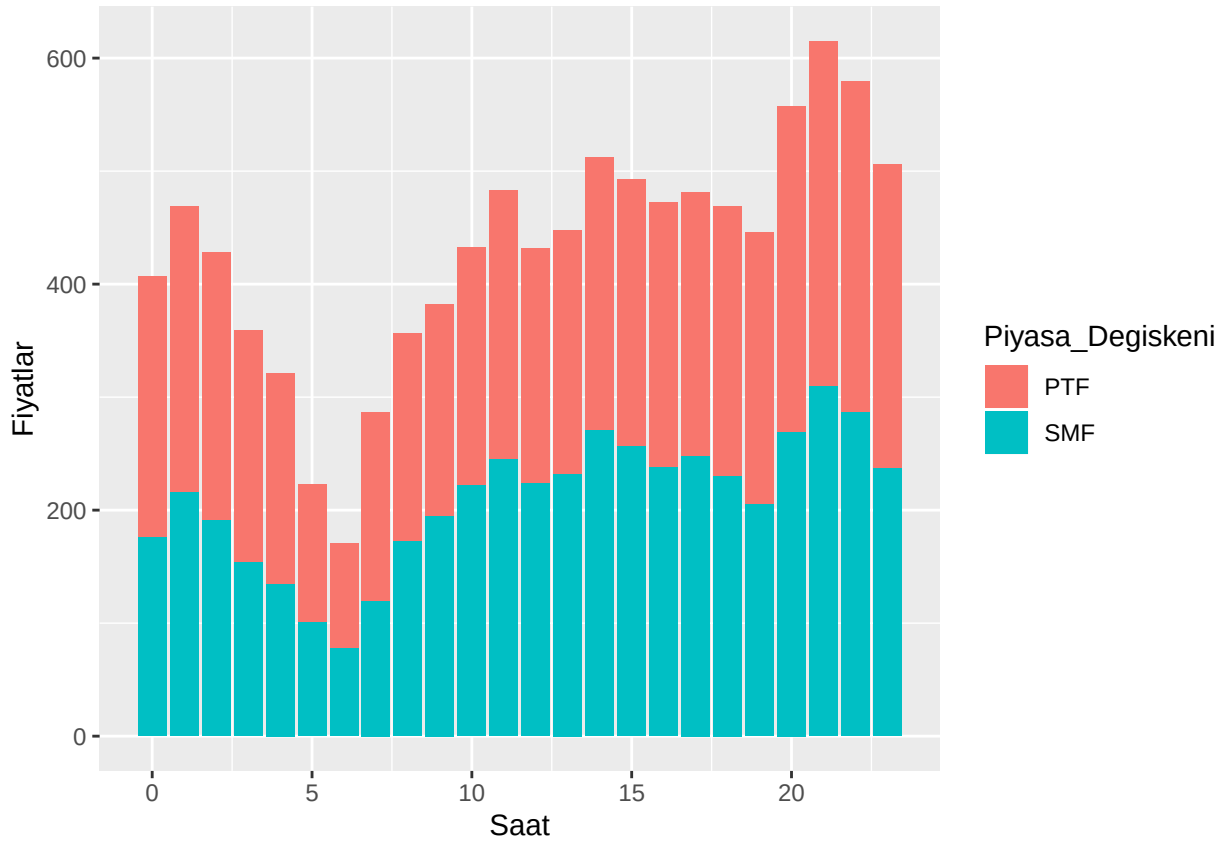
```
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>
```



3.3 Sütunlu Grafik (Bar Chart)

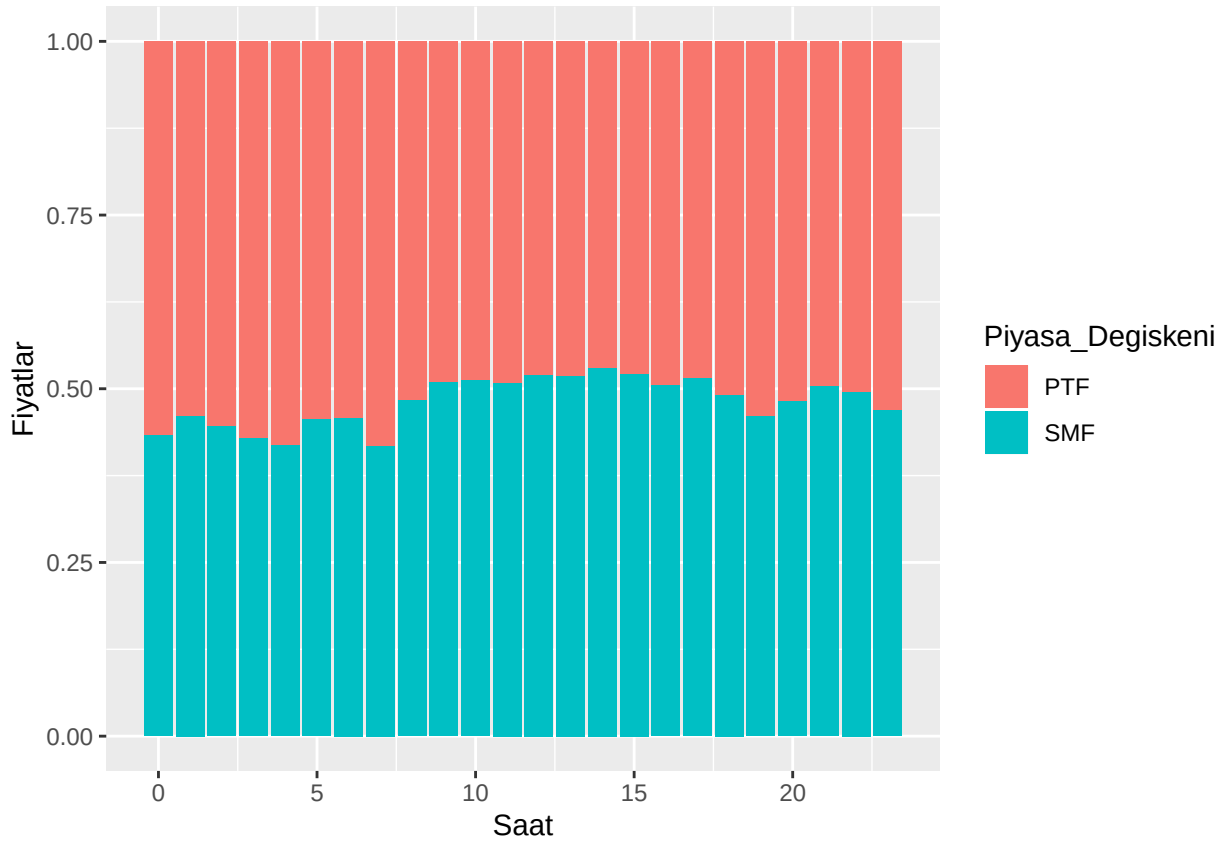
"Bar Chart"ta ise veriler sütunlarda gösterilir. "Scatter Plot"taki `geom_point()` yerine `geom_bar()` fonksiyonu, `aes()` içerisinde `color` yerine ise `fill` kullanılır. (Burada `geom_bar` içerisinde `stat="identity"` yazılması zorunludur.) İlk grafik şu şekilde ortaya çıkacaktır.

```
bar_plot <- ggplot(plot_df, aes(x=Saat, y=Fiyatlar, fill=Piyasa_Degiskeni)) +
            geom_bar(stat="identity")
bar_plot
```



Bu grafiği yüzdesel oranlarına göre tam olarak sığdırmak için ise,

```
bar_plot <- ggplot(plot_df, aes(x=Saat, y=Fiyatlar, fill=Piyasa_Degiskeni)) +
            geom_bar(stat="identity", position="fill")
bar_plot
```



Son olarak da daha aşina olunduğu üzere sütunları yan yana yerleştirmek için (yukarıda yapılan güzelleştirme operasyonları da uygulanırsa),

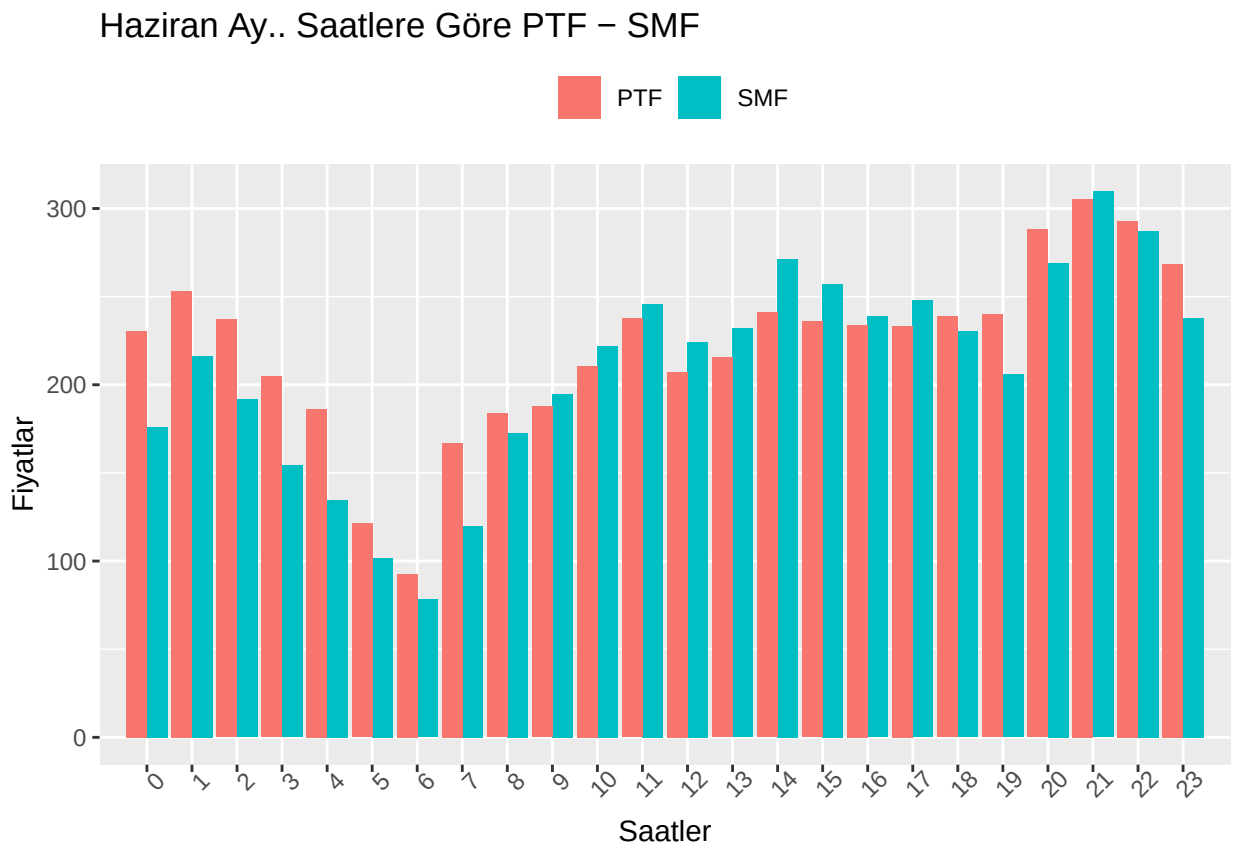
```
bar_plot <- ggplot(plot_df, aes(x=Saat, y=Fiyatlar, fill=Piyasa_Degiskeni)) +
            geom_bar(stat="identity", position="dodge") +
            labs(title = "Haziran Ayı Saatlere Göre PTF - SMF",
                 x = "Saatler",
                 y = "Fiyatlar") +
            scale_x_discrete(limits=c(0:23)) +
            theme(axis.text.x = element_text(angle = 45), legend.position = "top",
                  legend.title = element_blank())
bar_plot
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>
```

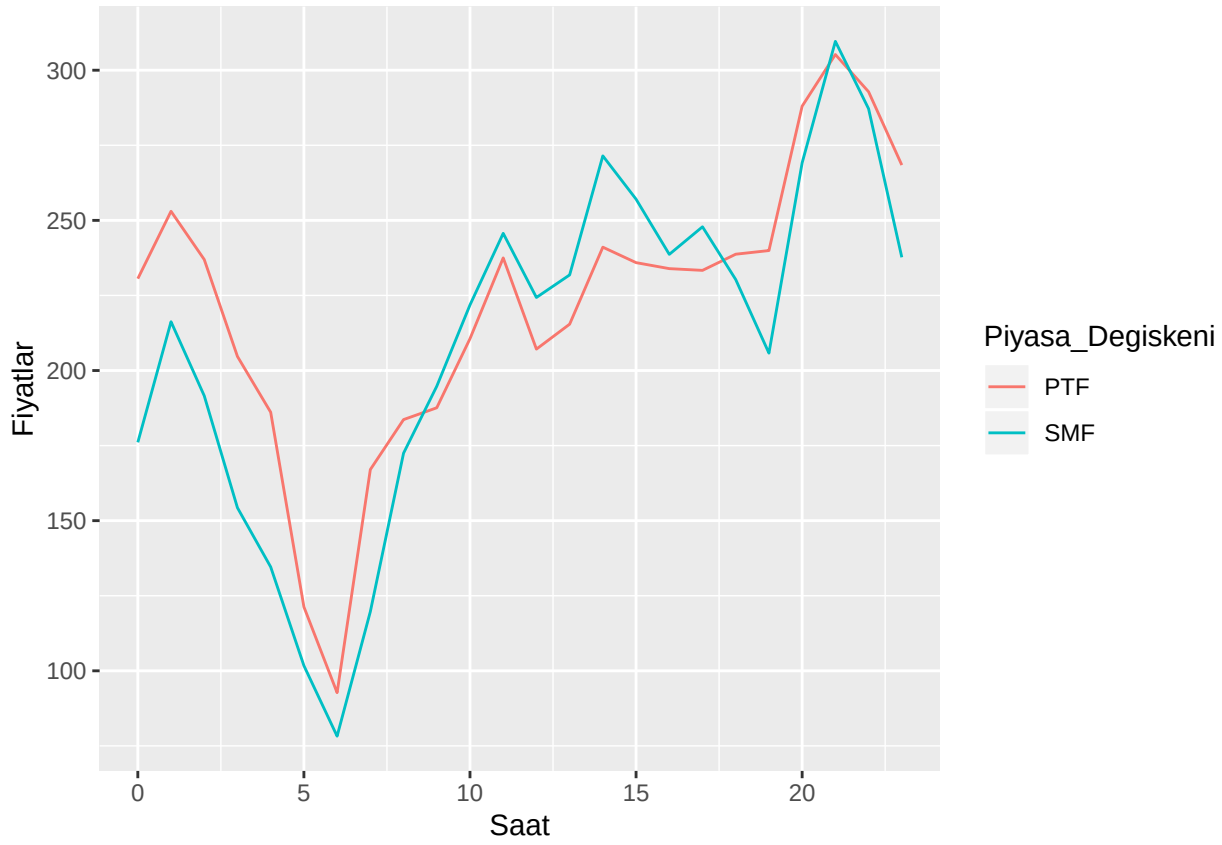


Son iki grafikte dikkat edilmesi gereken nokta sütunları dikeyde oranları göz önüne alınarak sığdırmak için `position="fill"`, yan yana yerleştirmek için ise `position="dodge"` kullanılır.

3.4 Çizgi Grafiği (Line Chart)

"Line Chart"larda ise tahmin edilebileceği üzere `geom_line()` kullanılacaktır.

```
ln_plot <- ggplot(plot_df, aes(x=Saat, y=Fiyatlar, color=Piyasa_Degiskeni)) +
           geom_line()
ln_plot
```

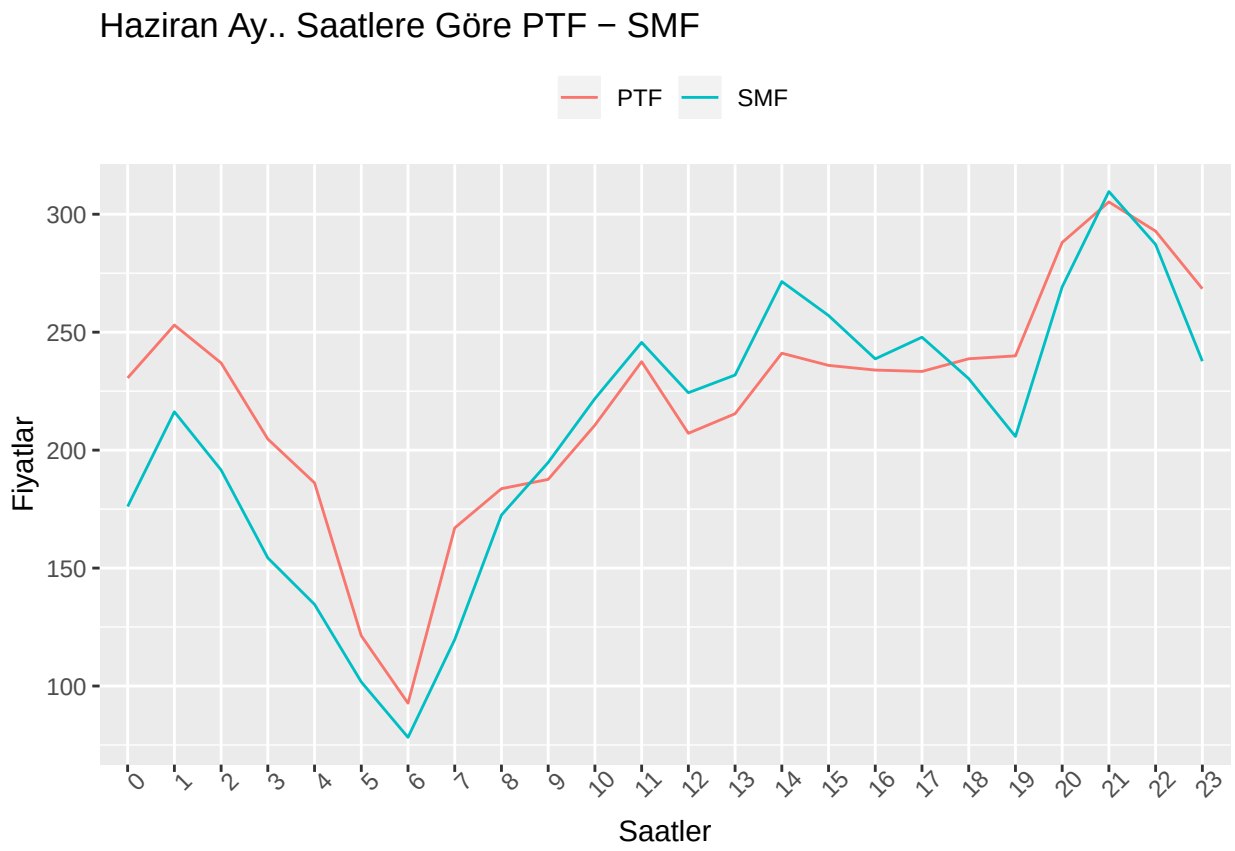


Yine daha güzel görünümlü bir grafik elde etmek için gerekli fonksiyonlar da yazılırsa,

```
ln_plot <- ln_plot +
           labs(title = "Haziran Ayı Saatlere Göre PTF - SMF",
                x = "Saatler",
                y = "Fiyatlar") +
           scale_x_discrete(limits=c(0:23)) +
           theme(axis.text.x = element_text(angle = 45), legend.position = "top",
                 legend.title = element_blank())
ln_plot
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Haziran Ayı Saatlere Göre PTF - SMF' in 'mbcsToSbcs': dot
## substituted for <b1>
```



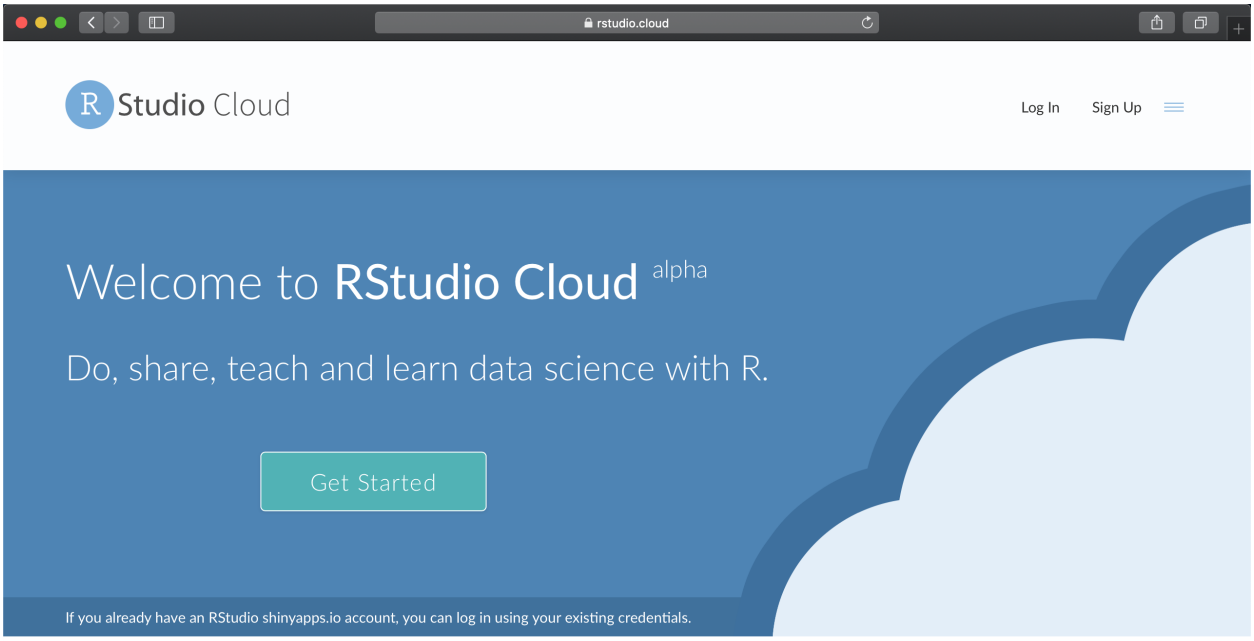
Appendix

RStudio Cloud'da Çalışma

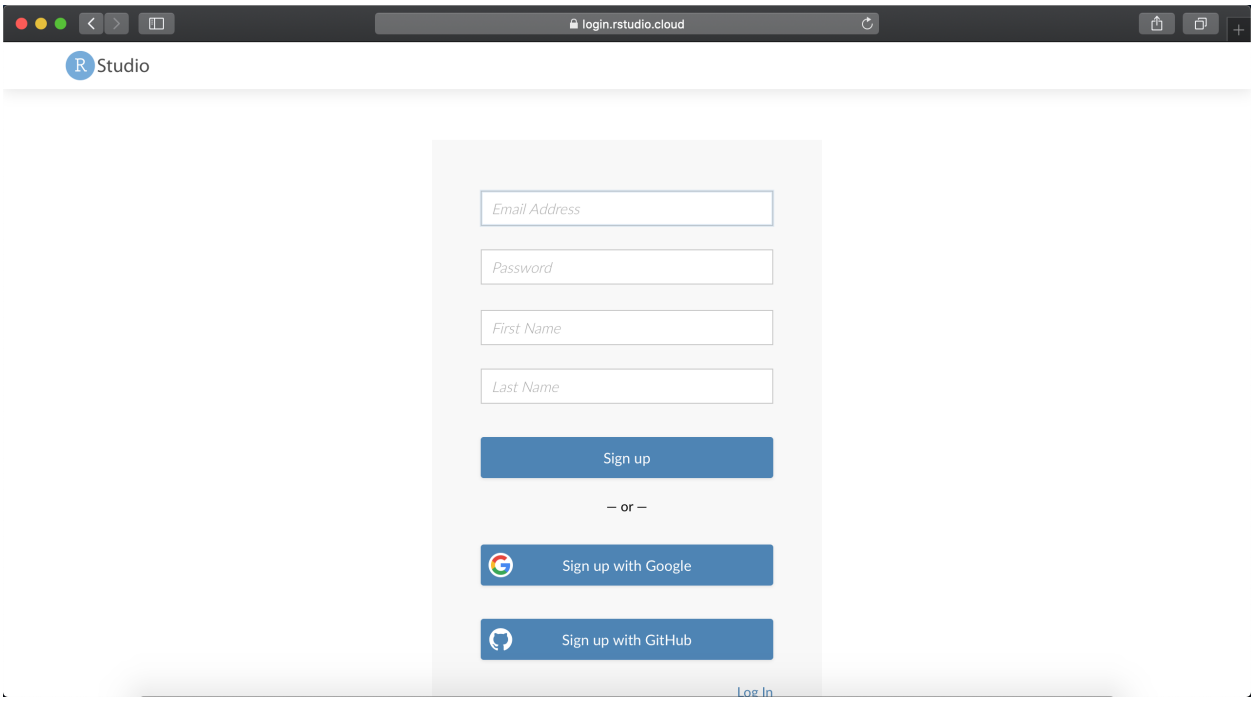
RStudio Cloud, R ve RStudio'yu bilgisayarınıza indirmeden çevrimiçi olarak kodlarınızı yazabileceğiniz, çalışmalarınızı is arkadaşlarınızla rahatça paylaşabileceğiniz, gerekli paketleri bilgisayarınıza yükleyip yüklemediğinizi dert etmeyeceğiniz tamamen ücretsiz bir platformdur.

RStudio Cloud'a erişmek ve platformu kullanmak için aşağıdaki adımları izleyebilirsiniz.

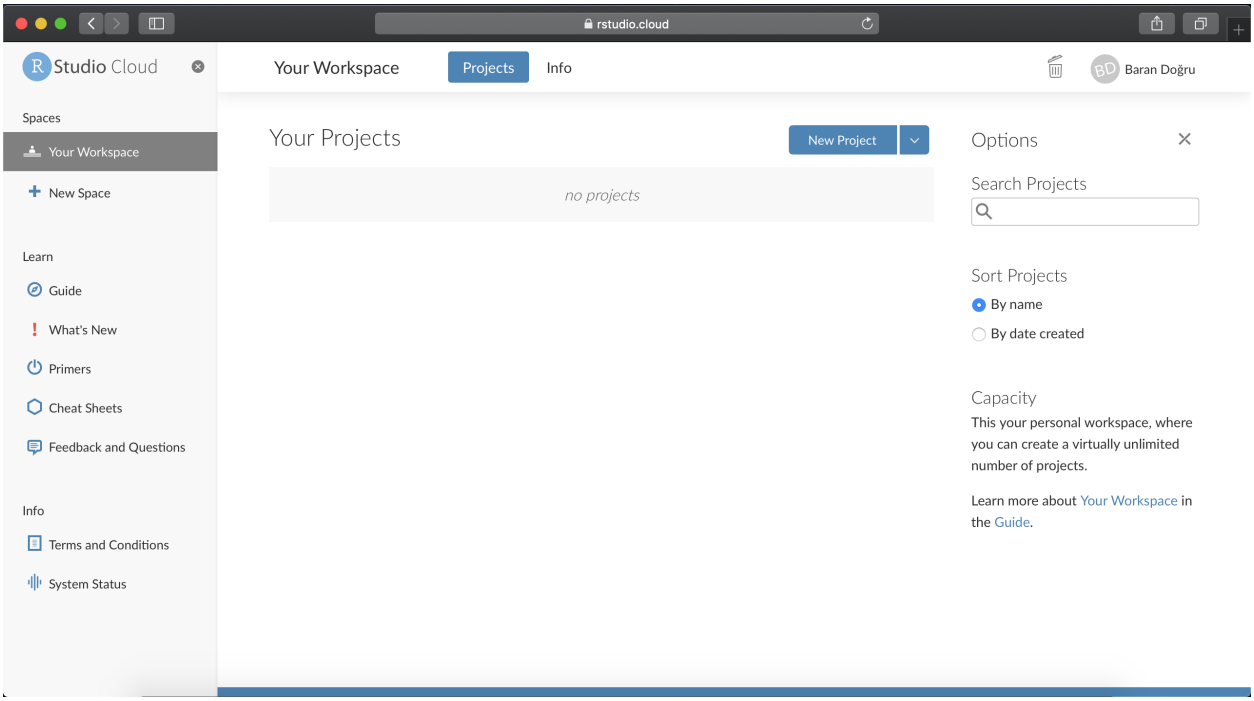
1. Tercih ettiğiniz web tarayıcıda "RStudio Cloud" yazarak aratılınca karşınıza çıkan ilk linke tıkladığınızda bu ekranla karşılaşacaksınız. Ana ekranın sağ üst köşesindeki "Sign Up" butonuna tıklayarak kayıt ekranına ulaşın.



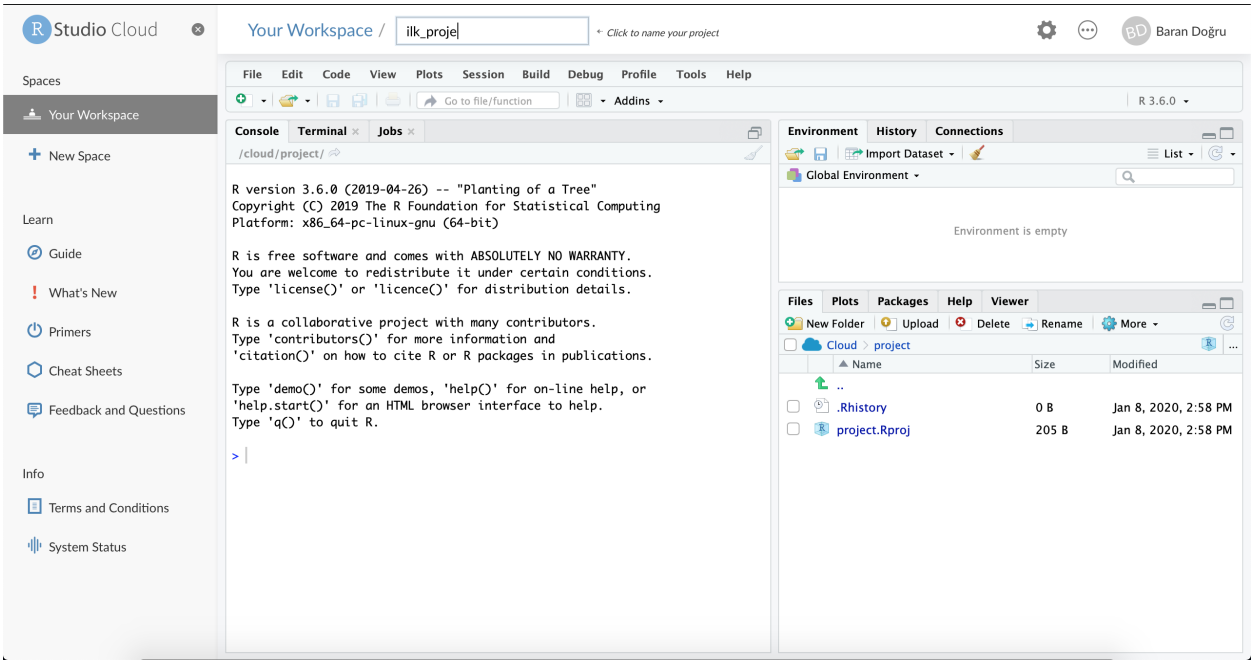
2. Karşınıza çıkan kayıt formunu doldurun.



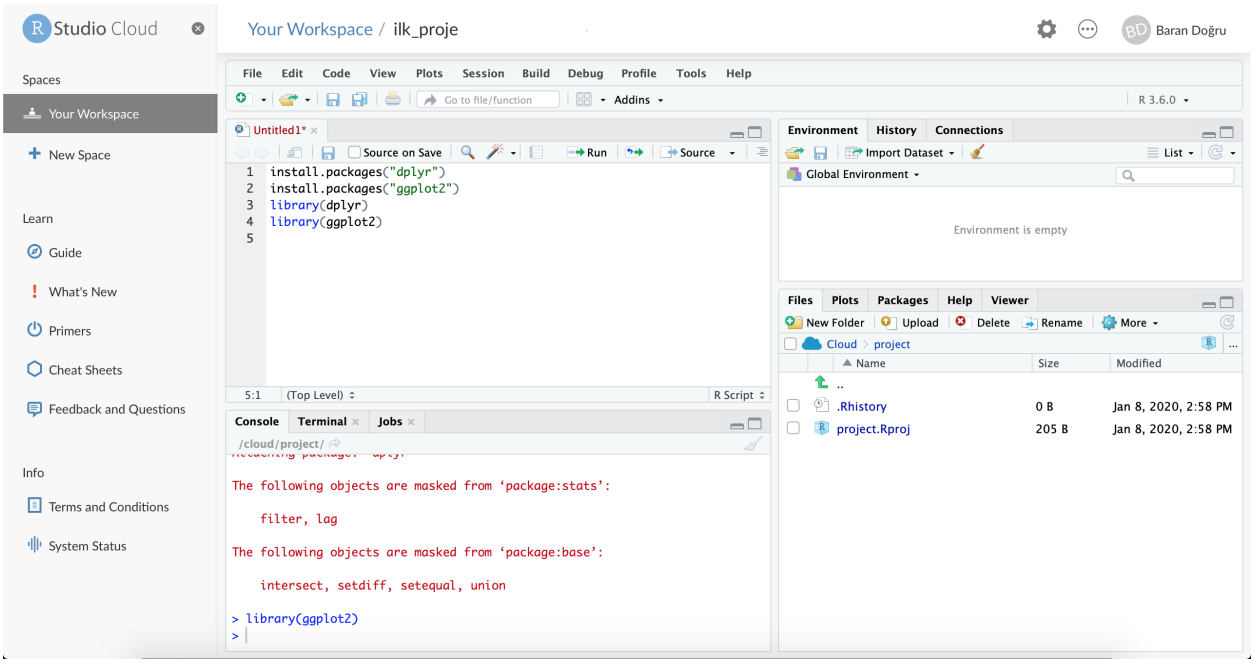
3. Kaydınızı tamamlayın ve hesabınıza giriş yapın. Artık RStudio Cloud'u kullanmaya hazırsınız.



4. İlk projenizi oluşturmak için “New Project” butonuna tıklayın.



5. Açılan ilk projenize R Script dosyası eklemek için ise üstteki bardan “File”, “New File” ve “R Script” sırasıyla seçin. Artık R’da kod yazmaya hazırsınız.



Veri Seti Düzenlemeleri

Bu bölümde ilgilenenler için veri setinin ham hali elde edildikten sonra dplyr ile Veri Manipülasyonu bölümünde kullanılan haline dönüştürmek için gereken kodlar paylaşılacaktır.

```
# pozitif ve negatif dengesizlik sutunlarinin isimlerinin duzenlenmesi
colnames(ptfsmf)[4] = "PDF"
colnames(ptfsmf)[5] = "NDF"

# tarih ve saatin POSIX formatina cevrilmesi
ptfsmf <- ptfsmf %>% mutate(Tarih_Yeni = as.POSIXct(ptfsmf$Tarih,format="%d.%m.%y %H:%M", "GMT")) %>%
                     select(Tarih_Yeni, PTF, SMF, PDF, NDF) %>%
                     rename(Tarih = Tarih_Yeni)

# geri kalan 4 sutunun numeric classina cevrilmesi
ptfsmf[,c(2:5)] <- lapply(ptfsmf[,c(2:5)],
                          function(x) as.numeric(gsub(",", ".", gsub("\\.", "", x))))
```