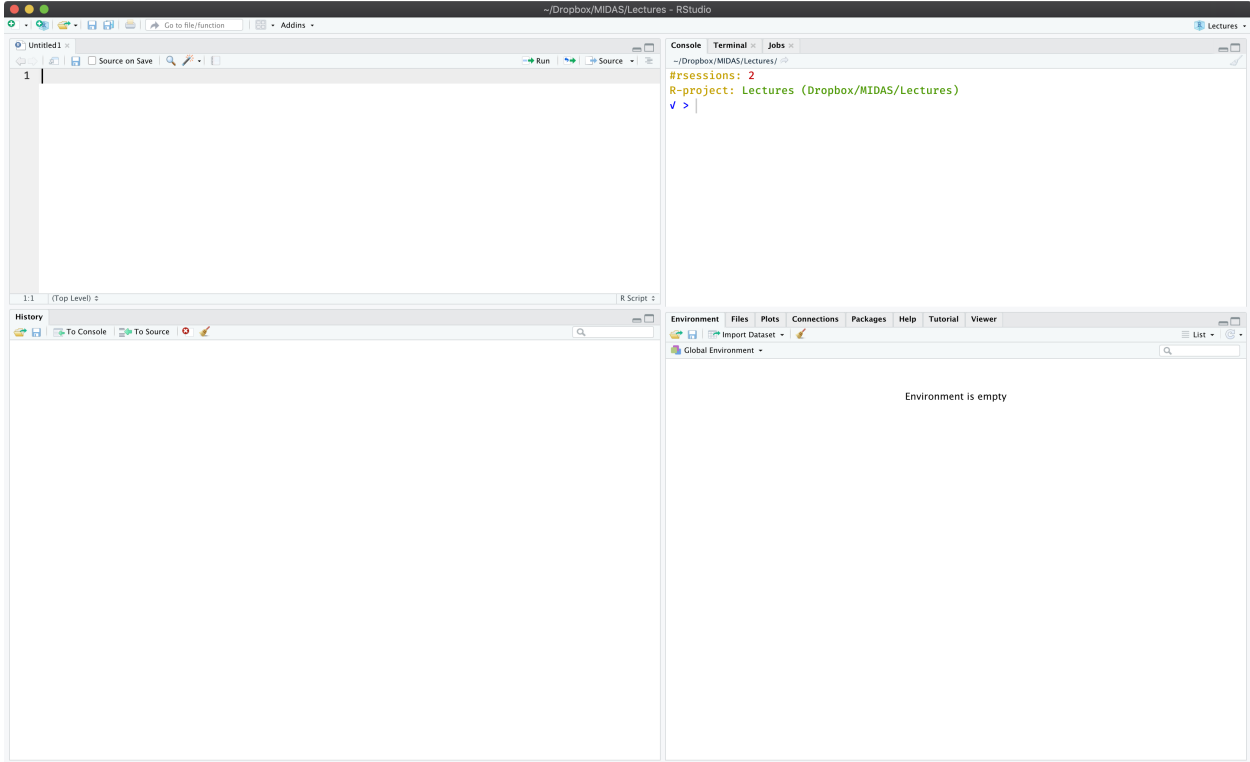
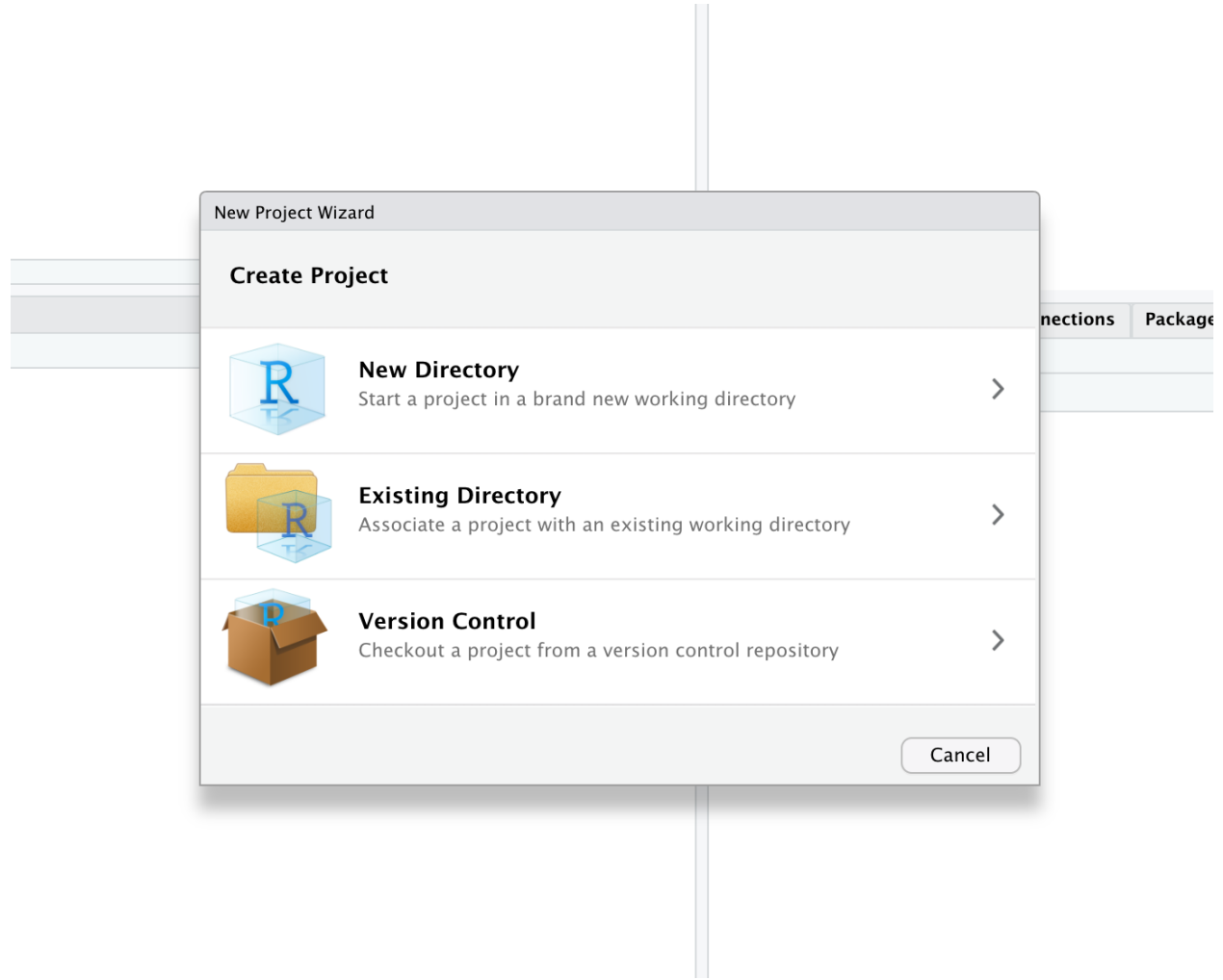


RMarkdown

RStudio Tanıtımı ve Opsiyonları



RStudio Projeleri



Markdown

Markdown, 'markup' dillerinin basitleştirilmiş bir versiyonudur. Bazı elverişsiz kelime işlem uygulamalarının aksine, Markdown ile yazılmış metinler basit ve sezgisel biçimlendirme öğelerini kullanırlar. Hızla akademisyenler, bilim adamları, yazarlar ve daha birçokları için yazma standardı haline gelmiştir. GitHub ve reddit gibi bazı popüler web siteleri yorum kısımlarında Markdown'ı kullanır.

Markdown'da metin biçimlendirmek çok hafif bir öğrenme eğrisine sahiptir. Yazı tipi boyutunu, rengini veya türünü değiştirmek gibi işlemler yapılamaz. Markdown'da, başlıklar ve listeler oluşturmak gibi basit işlemler yapılabilir.

İtalik ve Kalın Yazı Tipi

Markdown'da bir kelimeyi/cümleyi italik yapmak için kelime/cümle altçizgi () veya yıldız (*) işaretleri arasında yazılır. Örneğin: *bu* (bu) italik olacaktır.

Benzer biçimde kalın yazı tipi yapmak için de çift altçizgi () veya çift yıldız (**) kullanılır. Örneğin: **bunun** (**bunun**) gibi.

Genellikle italik için tek altçizgi, kalın yazı tipi için de çift yıldız kullanılır.

Doğal olarak hem italik hem de kalın yazı tipini kullanmak mümkündür. Çift yıldız ve altçizgi (**_) kullanarak kelimeleri/cümleleri kalın italik şekilde yazabiliriz. Örneğin: ***bağrı kelimeler*** (**_bağrı kelimeler_**) gibi.

Yıldızların ve altçizgilerin hangi sırada kullanıldığı önemli değildir ancak önce yıldız sonra altçizgi kullanılması metni okumayı kolaylaştırır.

Başlıklar

Markdown'da 6 farklı seviyede başlık türü vardır:

```
# Markdown
## Başlıklar
### Başlık üç
#### Başlık dört
##### Başlık beş
##### Başlık altı
```

Markdown'da başlık yapmak için başlık olmasını istediğiniz metnin önüne kare/diyez (#) işareti koymalısınız. Başlığımızın hangi büyüklükte/sırada/seviyede olmasını istiyorsanız o kadar kare/diyez (#) işareti kullanılır. Örneğin: birinci sıradaki bir başlık için, (# Markdown), tek kare/diyez işareti kullanılmalıdır.

Bağlantılar

Markdown'da 2 farklı türde bağlantı vardır. Bunlardan ilki *satır içi* bağlantı olarak adlandırılır. Satır içi bağlantı yaratmak için bağlantı metnini köşeli parantezler ([]) içine sonrasında bağlantı adresini de parantezler (()) içine yerleştirmelisiniz. Örneğin: Bağlantı metninde 'Mülkiye!' yazan ve tıklandığında SBF anasayfasına (<http://www.politics.ankara.edu.tr/>) giden bir bağlantı oluşturmak için Markdown'da [Mülkiye!](#) ([Mülkiye!](<http://www.politics.ankara.edu.tr/>)) yazmalısınız.

Diğer bir bağlantı şekli de referans/atıf bağlantısıdır. İsimden de anlaşılacağı üzere bu bağlantı türü metinde başka bir noktaya referans vermek için kullanılır. Örneğin:

Bu [başka bir yere bağlantıdır](#)

Bu ise [diğer bir yere bağlantıdır](#)

Bu tekrar [ilk bağlantıdır](#)

Yukarıdaki referanslar ikinci köşeli parantez içerisinde yer alanlardır: [başka bir yer](#) ve [diğer bir yer](#). Markdown metninin altında bu köşeli parantezler diğer dökümanlara/sitelere bağlantılar olarak tanımlanmıştır. Referans tipi bağlantı kullanmanın avantajı metinde aynı yere tekrar tekrar bağlantı veriliyorsa bu bağlantıların güncellenmesi gerekiyorsa kolaylık sağlamasıdır. Eğer [başka bir yer](#) bağlantısının GitHub değil de GitLab sitesine gitmesini istiyorsak sadece tek bir bağlantı adresini değiştirmemiz yeterlidir.

Referans bağlantıları Markdown metnini pdf veya html dosyalarına çevirdiğimizde görünmezler. Referans bağlantıları köşeli parantez içerisinde isim verilip iki nokta koyduktan sonra adresi vererek yapılabilir.

Bu [\[başka bir yere bağlantıdır\]](#)[\[başka bir yer\]](#)

Bu ise [\[diğer bir yere bağlantıdır\]](#)[\[diğer bir yer\]](#)

Bu tekrar [\[ilk bağlantıdır\]](#)[\[başka bir yer\]](#)

[\[başka bir yer\]](#): <https://github.com/>

[\[diğer bir yer\]](#): <https://www.google.com/>

Resimler

Markdown'da resim eklemek bağlantı eklemeye çok benzer. Resimler de iki farklı şekilde eklenebilir. Bağlantılar ve resimler arasındaki tek fark resimleri eklerken kodun önüne ünlem işareti (!) koyarız.

İlk resim türü satırıçi resim bağlantısıdır. Satırıçi resim bağlantısı eklemek için önce ünlem işareti (!), ardından köşeli parantez ([]) içerisinde resmin alt metni (resim başlığı olarak düşünebilirsiniz) ve son olarak da parantez () içerisinde resmin bağlantısını yazmamız gerekir.

Örneğin, <https://octodex.github.com/images/bannekat.png> adresine satırıçi resim bağlantısı eklemek için aşağıdaki kodu yazabiliriz.

```
![Benjamin Bannekat](../../images/bannekat.png)
```



Şekil 1: Benjamin Bannekat

Referans resimler için de benzer bir yol izleriz.

![İlk Kurucu Baba][İlk Kurucu Baba]

![İlk Kurucu Baba 2][İlk Kurucu Baba 2]

[İlk Kurucu Baba]: ../../images/founding-father.jpg

[İlk Kurucu Baba 2]: ../../images/foundingfather_v2.png



Şekil 2: İlk Kurucu Baba



Şekil 3: İlk Kurucu Baba 2

Listeler

İki farklı türde listeler yaratabiliriz, sırasız ve sıralı. Sırasız liste oluşturmak için listenin her bir elemanının önüne yıldız (*) ekleriz. Her liste elemanı yeni bir satırda olmalıdır. Örneğin, market listesi şu şekilde yazılabilir:

- Süt
- Yumurta
- Ekmek
- Zeytin

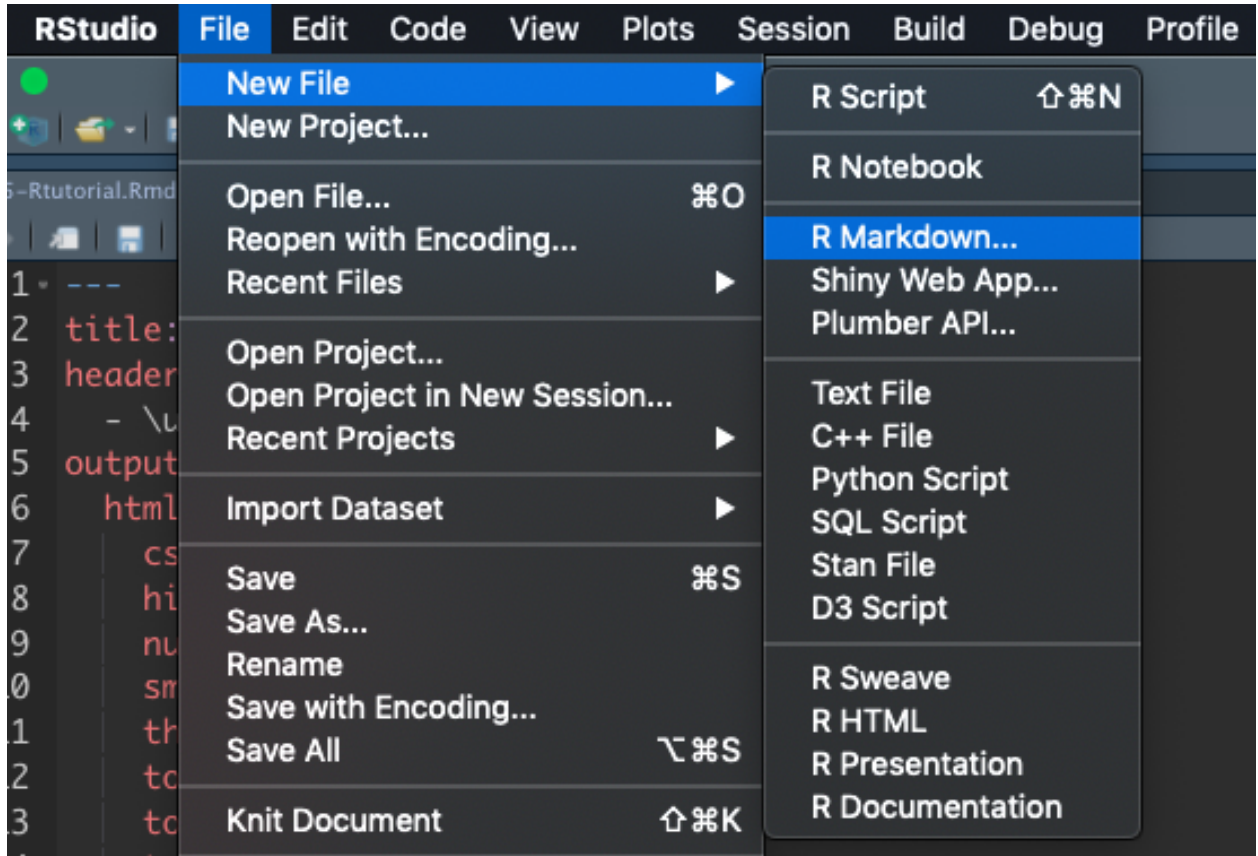
Sıralı listeler ise yıldız yerine sayı kullanılarak oluşturulur. Örneğin:

1. Domates
2. Biber
3. Patlıcan
4. Kabak

RStudio'da Markdown

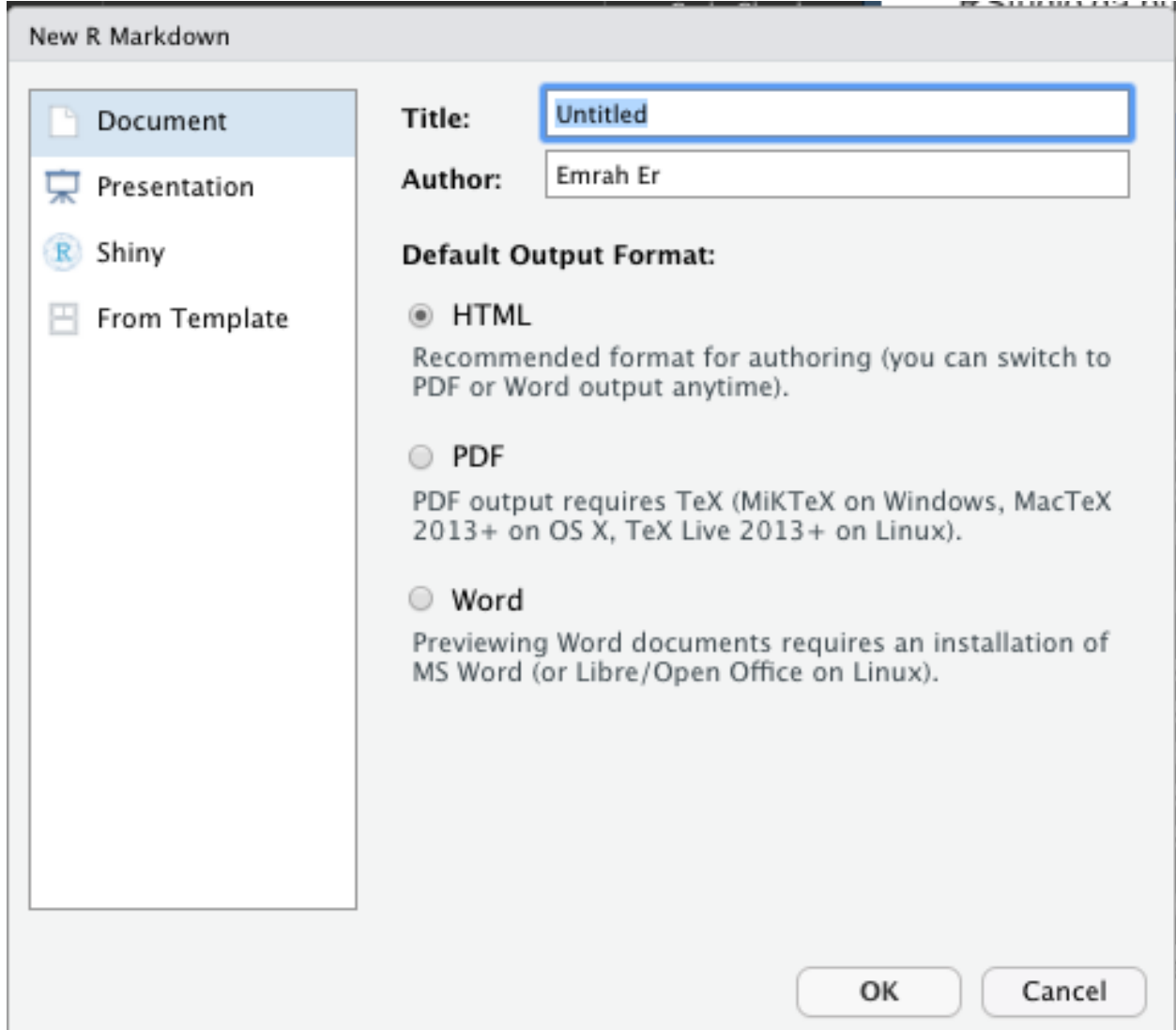
RStudio'da Markdown dosyaları özelleştirilmiş Markdown dosyalarıdır ve R Markdown dosyaları olarak adlandırılır. Markdown ve R Markdown dosyalarının neler olduğuna aşağıda değineceğiz fakat bu bölümde RStudio'da bu dosyaları nasıl yaratacağımız konusuyla başlayalım.

RStudio'da R Markdown dosyası yaratmak için **File > New File > R Markdown** seçeneğini seçin.



Şekil 4: File > New File > R Markdown

Bu seçeneği seçtiğimizde karşımıza bir menü gelir. Bu menüde sol kısımda bulunan seçeneklerden **Document** seçeneğini seçin. Sağ tarafta yer alan kısımlardan **Title** oluşturmak istediğiniz dökümanın başlığını, **Author** ise dökümanın yazarını gösterir. Bunları değiştirebilirsiniz. Sağ tarafta **HTML**, **PDF** ve **Word** olmak üzere 3 farklı seçenek bulunmaktadır. Bunlar R Markdown dökümanınızı işlediğinizde (render) üretilecek çıktı dosyasının türünü gösterir. Bu aşamada **HTML** seçeneğini seçebilir ve **OK**'e tıklayabilirsiniz.



Şekil 5: R Markdown Menü

OK'e tıkladığınızda RStudio'da yeni bir dosya açılacaktır. RStudio otomatik olarak bu dosyayı bazı metinler ile doldurur. Dosyanın ilk 6 satırında YAML Metadata yer alır. Bu bilgiler R Markdown dosyasına ait bilgileri içerir.

Bu dosya konusunda ileride daha fazla açıklama yapacağız. Bu noktada 8. satırdan 31. satıra kadar olan metni silip dosyanızı kaydedebilirsiniz. RStudio'da bu dosyaları nasıl yaratacağımızı öğrendiğimize göre Markdown nedir konusuna geçebiliriz.


```
1 ---
2 title: "İlk R Markdown Dosyam"
3 author: "Emrah Er"
4 date: "3/6/2019"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS
15 Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16
17 When you click the Knit button a document will be generated that includes both content as well as
18 the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code
33 that generated the plot.
```

Şekil 6: R Markdown Dosyası

knitr

knitr, *istatistiki okuryazar programlama* (literate programming) için kullanılan bir R paketidir. **knitr** kullanarak kod ve metni tek bir belge biçiminde birleştirebilirsiniz. **knitr** ile yazılan R Markdown dosyaları pdf ve html dökümanlarına dönüştürülebilir.

knitr ile R Markdown yazmanın güzel bir yanı kodunuzda hata varsa döküman üretmemesidir. Bu sayede kodunuzda yer alan kodları kolaylıkla bulabilirsiniz.

`install.packages("knitr")` komutunu çalıştırarak **knitr** paketini indirebilirsiniz.

Rmarkdown

Bu döküman gibi hem Markdown hem de R kodu içeren dökümanlar R Markdown olarak adlandırılır. **rmarkdown** ise R Markdown belgeleri ile çalışmayı kolaylaştıran bir R paketidir.

RStudio'da yer alan 'Knit' butonu tıklatıldığında `.Rmd` dosyası **knitr** paketine beslenir ve **knitr** belge içindeki tüm R kodlarını çalıştırarak bir Markdown (`.md`) dosyası yaratır. Bu dosya yazdığımız metin ile birlikte yazılan kodları ve kodların çalıştırılması sonucu elde edilen grafik veya tablo gibi özetleri içerir. En son olarak bu markdown dosyası **pandoc** adı verilen bir yazılımla istenilen formata (pdf, html, word) dönüştürülür.



Şekil 7: RMarkdown Akışı

Kod Parçaları (Code Chunks)

RStudio'da kod parçası yaratıp metne yerleştirmenin 3 farklı yolu vardır:

1. Kod parçası başlatmak için ````${code}````, bitirmek için ise ````` yazmak
2. **Ctrl + Alt + I** kısayolunu kullanmak
3. **Add Chunk** butonuna tıklamak

Aşağıda yer alan komutları R Markdown dökümanına yazdığımızda bu bir kod parçası olacaktır.

```
```${code}```  
head(pressure)
summary(pressure)
```
```

Metin **knit** edildiğinde ise bu kod parçası aşağıdaki şekilde gözükecektir.

```
head(pressure)
```

```
##   temperature pressure  
## 1           0    0.0002  
## 2          20    0.0012  
## 3          40    0.0060  
## 4          60    0.0300  
## 5          80    0.0900  
## 6         100    0.2700
```

```
summary(pressure)
```

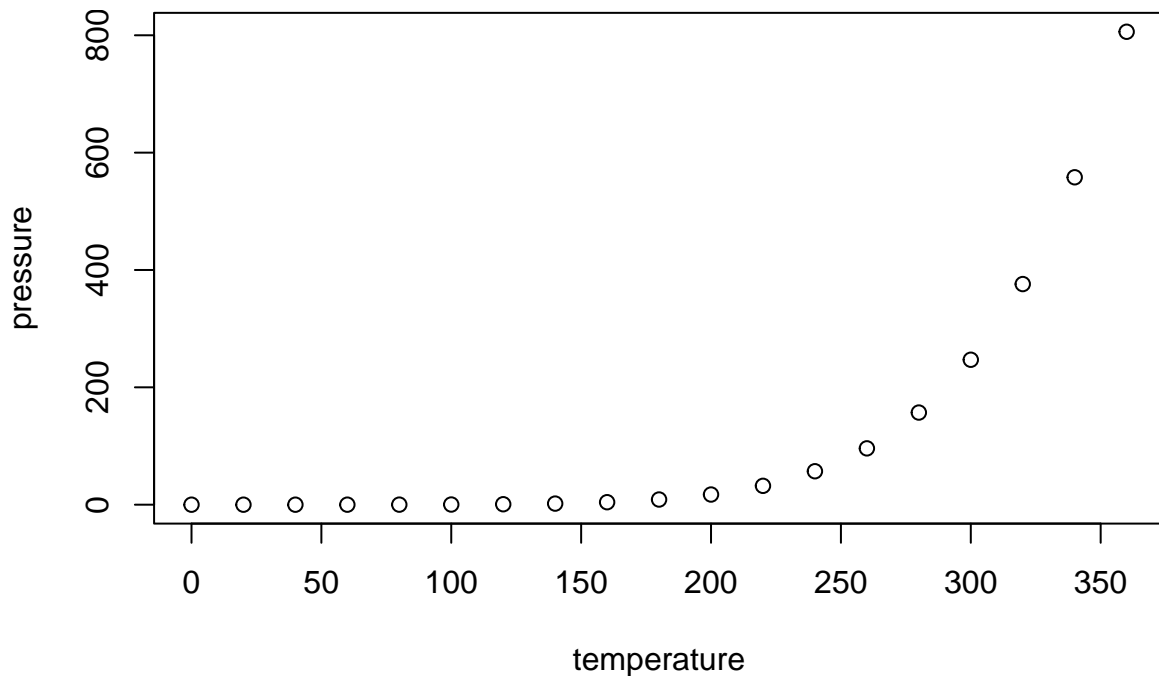
```
##   temperature      pressure
##   Min.   : 0      Min.   : 0.0
##   1st Qu.: 90     1st Qu.: 0.2
##   Median :180     Median : 8.8
##   Mean   :180     Mean   :124.3
##   3rd Qu.:270     3rd Qu.:126.5
##   Max.   :360     Max.   :806.0
```

Hem yazılan kodun hem de kod sonucunun son döküman olan bu belgede görüldüğüne dikkat edin.

Grafikler

Grafikler de benzer biçimde kod parçaları ile üretilebilir.

```
plot(pressure)
```



Satırıcı Hesaplamalar

Yazdığımız kodun ürettiği sonuçları satırıcınde de kullanabilirsiniz. Örneğin, x ve y gibi 2 rassal sayı oluşturup bunları metin içinde kullanmak istersem, ilk olarak sayıları türeten kod parçasını yazarım.

```
''{r}
x <- rnorm(1)
y <- rnorm(1)
'''

(x <- rnorm(1))
```

```
## [1] 0.2211
```

```
(y <- rnorm(1))
```

```
## [1] -0.04568
```

Sonrasında ise ``r x`` ve ``r y`` yazarak bu iki sayıyı metin içinde kullanabilirim. Bu örnekte $x = 0.2211$ ve $y = -0.0457$ 'dir.

Denklemler

LaTeX

Denklemleri yazmak için kullanılması gerek farklı bir 'dil' vardır. *LaTeX* olarak adlandırılan bu dil ile denklemler ve formül yazmak çok kolaydır. Satır içi denklem yazmak için 2 \$ işareti, blok denklem yazmak için ise 2 \$\$ işareti kullanılır. Örneğin, $A = \pi * r^2$ (`$A = \pi*r^{2}$`) bir satır içi denklem iken

$$\mathbb{E}[Y] = \beta_0 + \beta_1 x$$

```
$$  
\mathbb{E}[Y] = \beta_0 + \beta_1 x  
$$
```

bir blok denklemdir. *LaTeX* ile matematiksel ifadeleri yazma konusunda [bu sayfayı](#) inceleyebilirsiniz. Sayfada sıklıkla kullanılan denklemlerin ve Yunan harflerinin nasıl yazıldığını görebilirsiniz.

Sunum

RStudio'da R Markdown dosyası yaratmak için **File > New File > R Markdown** seçeneğini seçin. Bu seçeneği seçtiğimizde karşımıza bir menu gelir. Bu menüde sol kısımda bulunan seçeneklerden **Presentation** seçeneğini seçin. Sağ tarafta yer alan kısımlardan **Title** oluşturmak istediğiniz dökümanın başlığını, **Author** ise dökümanın yazarını gösterir. Bunları değiştirebilirsiniz. Sağ tarafta **HTML (ioslides)**, **HTML (Slidy)**, **PDF (Beamer)** ve **Powerpoint** olmak üzere 4 farklı seçenek bulunmaktadır. Bunlar R Markdown dökümanınızı işlediğinizde (render) üretilcek çıktı dosyasının türünü gösterir. Bu aşamada **PDF (Beamer)** seçeneğini seçebilir ve **OK**'e tıklayabilirsiniz.

OK'e tıkladığınızda RStudio'da yeni bir dosya açılacaktır. RStudio otomatik olarak bu dosyayı bazı metinler ile doldurur. Dosyanın ilk 6 satırında YAML Metadata yer alır. Bu bilgiler R Markdown dosyasına ait bilgileri içerir.

Görsel R Markdown

Daha fazla bilgi için lütfen [Visual R Markdown](#) sayfasına bakınız.

Nasıl Soru Sorulur ve Yardım Alınır?

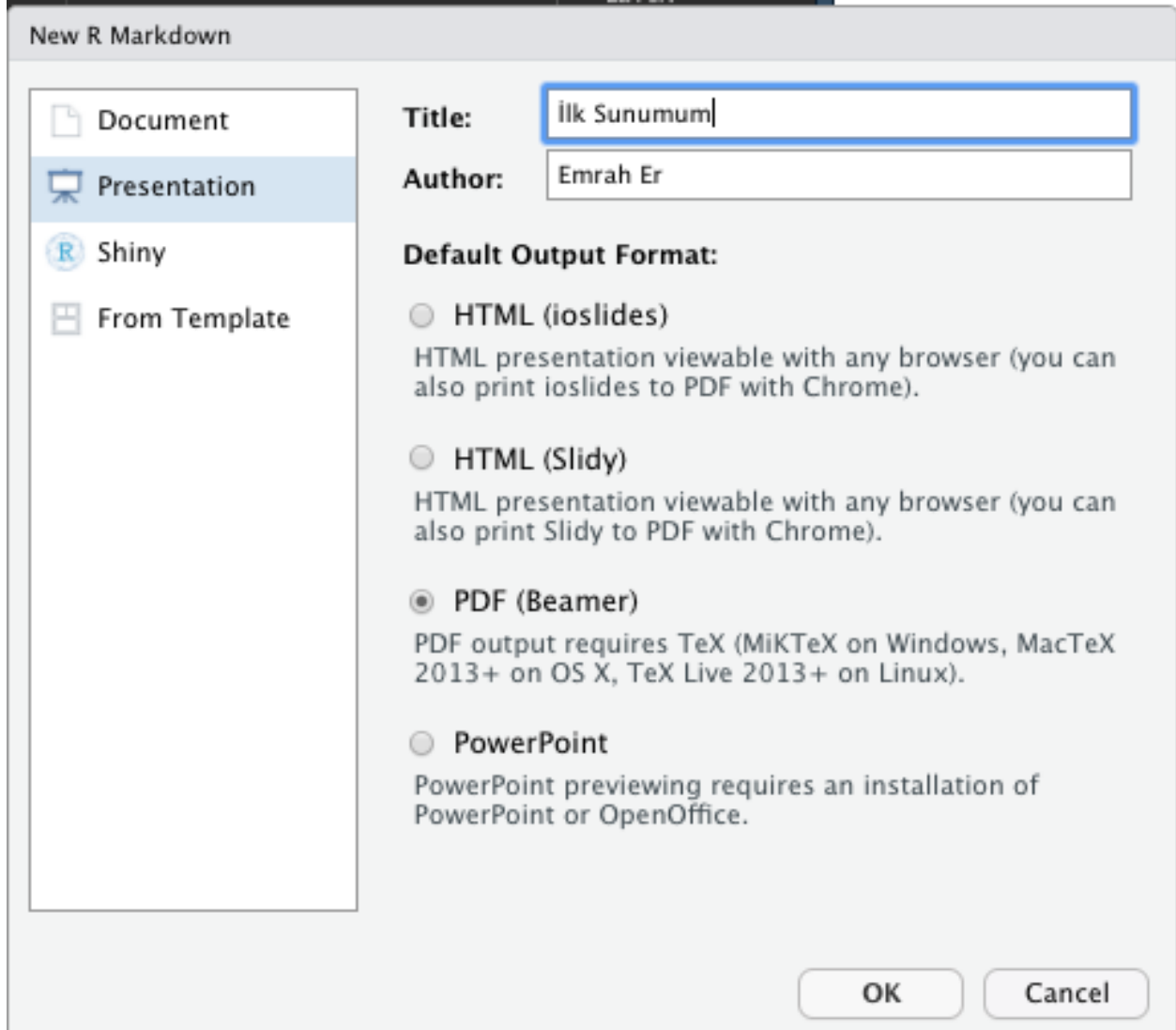
En önemli ve ilk yapılması gereken: [RTFM](#) ve [rtfm!](#)

Cevaplar için aşağıdaki sayfaları kullanabilirsiniz.

- [RSeek](#)
- [Google](#)
- [SO](#)
- [Github](#)

Aşağıdaki bağlantılar soru sorma konusunda faydalı olabilir.

- **(Bu en önemli olanı!)** [Nasıl Akıllıca Soru Sorulur?](#)
- [How to create a Minimal, Reproducible Example](#)
- [How to make a great R reproducible example](#)
- [FAQ: Tips for writing R-related questions](#)
- [Get help!](#)



Şekil 8: R Markdown Menü - Beamer

```

1  ---
2  title: "İlk Sunumum"
3  author: "Emrah Er"
4  date: "3/11/2019"
5  output: beamer_presentation
6  ---
7
8  ```{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = FALSE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown presentation. Markdown is a simple formatting syntax for authoring HTML, PDF, and
15 MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16
17 When you click the Knit button a document will be generated that includes both content as well as
18 the output of any embedded R code chunks within the document.
19
20 ## Slide with Bullets
21
22 - Bullet 1
23 - Bullet 2
24 - Bullet 3
25
26 ## Slide with R Output
27
28 ```{r cars, echo = TRUE}
29 summary(cars)
30 ```
31
32 ## Slide with Plot
33
34 ```{r pressure}
35 plot(pressure)
36 ```

```

Şekil 9: R Markdown Beamer Dosyası

Bazı R paketleri ve RStudio için hazırlanan **kopya kağıtları** (cheatsheet) için de aşağıdaki siteyi inceleyebilirsiniz.

- [RStudio Cheatsheets](#)

reprex

- Ayrıntılı bilgi için [bkz.](#)
- [Şu](#) YouTube videosunu da izleyebilirsiniz.

Kurulum

```
install.packages("reprex")
```

Addins

Render reprex & Reprex selection

Kullanım

```
library(reprex)
(y <- 1:4)
mean(y)
```

Yapacağımız şey kodu seçtikten sonra addin kullanarak bunu reprex haline dönüştürmek.

Kodu seçin ve Addins menüsünden Reprex Selection'ı tıklayın. reprex otomatik olarak panoya (clipboard) kopyalanacaktır.

```
(y <- 1:4)
#> [1] 1 2 3 4
mean(y)
#> [1] 2.5
```

Created on 2021-03-01 by the [reprex package](#) (v1.0.0)

Başka bir örnek yapalım. Bu sefer kodu seçip panoya kopyalayalım ve Addins'ten Render reprex seçelim.

```
library(tidyverse)
count(diamonds, colour)
```

reprex şu şekilde olacaktır.

```
library(tidyverse)
count(diamonds, colour)
#> Error: Must group by variables found in '.data'.
#> * Column 'colour' is not found.
```

Ekran Görüntüsü Alma

Maillerinizde veya Telegram mesajlarında ekran görüntüsü paylaşırken telefon ile ekran fotoğrafı çekmek yerine ekran görüntüsü alma araçlarını kullanabilirsiniz.

MS Windows Ekran Alıntısı Aracı

[Ekran görüntüleri yakalamak için Ekran Alıntısı Aracı'nı kullanma](#) başlıklı yazıyı okuyarak Windows kısayollarını öğrenebilirsiniz.

Mac OS Ekran Resmi veya Ekran Kaydı

Mac kullanıyorsanız kısayol tuşları ile kolaylıkla ekran görüntüsü alabilirsiniz. [Mac'te ekran resmi veya ekran kaydı alma](#) başlıklı yazıyı okuyarak bu kısayolları öğrenebilirsiniz.

RStudio ve Github Test Sürüşü

- [RStudio & Github Youtube Videosu](#)
- [RStudio Cloud & Github Youtube Videosu](#)

Git'i kişiselleştirme

Burada anlatılanlar bir kez yapılmalıdır. Tekrar tekrar yapılmasına gerek yoktur!

```
install.packages("usethis")
library(usethis)
use_git_config(user.name = "Emrah Er", user.email = "eer@eremrah.com")
```

`user.name` kısmına kendi adınızı `user.email` kısmına Github'a üye olurken kullandığınız e-posta adresini yazınız.

Github'da yeni repo oluşturma

- Github'a giriş yapın.
- Yeşil "New Repository" butonuna tıklayın.
 - myrepo
 - Public
 - "Initialize this repository with a README" tıklayın.
 - Yeşil "Create repository" butonuna tıklayın.
- "Code" butonundan HTTPS seçin ve adresi kopyalayın.

RStudio'da repoyu klonlama

- File > New Project > Version Control > Git
- Kopyaladığınız adresi yapıştırın.
- Klasör adı belirleyin.
- Klasörün nereye kaydedileceğini seçin.
- "Open in new session" tıklayın ve "Create Project" seçin.

Değişiklik yapma, kaydetme, commit ve push

- README.md dosyasının içeriğini değiştirin.
- Kaydedin.
- Git sekmesinden önce stage, sonra commit, en son olarak da push.
- Github üzerinde yaptığınız değişikliği kontrol edin.

Değişiklikleri pull etme

- Bu eğer grup (bu kullandığınız 2 farklı bilgisayar da olabilir) ile çalışıyorsanız grubunuzun Github'a push ettiği değişiklikleri bilgisayarınıza indirmeye yarar.
- Github üzerinde README dosyasının içeriğini değiştirelim.
- Git sekmesinden pull seçin.