Veri Temizleme

Servet Çetin

2024-07-11

İçindekiler

Önsöz

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

```
1 + 1
```

```
[1] 2
```

1 Giriş

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Özet

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

3 Veri Temizleme

3.1 Giriş

3.1.1 Veri Temizliğinin Önemi

Amaç: Veri temizliğinin neden analiz süreçlerinde kritik bir adım olduğunu detaylandırın.
Özelleşmiş Bilgi: Kirli verilerin, analiz sonuçlarını nasıl olumsuz etkileyebileceği; eksik, hatalı veya tutarsız verilerin çıkarım hatalarına ve yanlış kararlara yol açma riskleri.
R Markdown Önerisi: Kısa bir açıklamanın ardından, bu bölümün önemini vurgulayan birkaç görsel veya örnek ekleyin. Örneğin, temizlenmemiş veri ile temizlenmiş veri arasındaki farkları gösteren küçük bir örnek tablo oluşturabilirsiniz.

3.1.2 Veri Temizleme Tekniklerine Genel Bakış

Amaç: Belirtilen tekniklerin ne işe yaradığı ve neden bu dosyada yer aldıkları hakkında kısa bir tanıtım.
Özelleşmiş Bilgi: Her bir tekniğin veri seti üzerinde nasıl bir etki sağladığını açıklayan kısa tanımlar (filtreleme, çoğaltma giderme, tamamlama vb.).
R Markdown Önerisi: Her tekniği kısa bir açıklamayla tanıtan bir liste oluşturun. Bu listeye tıklanabilir bağlantılar ekleyerek, okuyucunun ilgili bölüme kolayca erişmesini sağlayabilirsiniz.

3.1.3 Çalışmanın Kapsamı ve Hedefi

Amaç: Bu çalışma dosyasının okuyucuya sağlayacağı faydaları açıklayın.
Özelleşmiş Bilgi: Dosyanın veri temizleme sürecindeki adımları öğrenmek isteyenler için bir rehber niteliğinde olduğu, özellikle R'de veri temizleme tekniklerine yönelik örnekler içerdiği belirtilebilir.
R Markdown Önerisi: Çalışmanın hangi seviyedeki kullanıcıya yönelik olduğunu vurgulayın (örneğin, başlangıç seviyesinde veri analistleri). Bu bölüme hedef kitlenizi belirtmek için ayrı bir paragraf ekleyebilirsiniz.

3.1.4 Veri Temizliğinde Karşılaşılan Yaygın Sorunlar

Amaç: Veri temizliği sırasında ortaya çıkan yaygın sorunlara dikkat çekmek.
Özelleşmiş Bilgi: Eksik veriler, hatalı veri girişleri, mükerrer kayıtlar ve aykırı değerler gibi sorunları tanımlayın.
R Markdown Önerisi: Bu bölümde her bir soruna kısa açıklamalarla yer verin. Sorunları görselleştirmek adına, basit bir örnek veri seti oluşturabilir ve sorunları örnek üzerinden göstererek okuyucuya daha net bir anlayış sunabilirsiniz.

3.1.5 Bu Dosyada Kullanılan Araçlar ve R Paketleri

Amaç: Çalışmada hangi R paketlerinin kullanılacağını tanıtmak ve kısa açıklamalarını yapmak.
Özelleşmiş Bilgi: `dplyr`, `mice`, `skimr`, `assertive` gibi paketlerin kullanımı hakkında kısa bilgiler.
R Markdown Önerisi: Her paketin yanında, ne işe yaradığını ve veri temizleme sürecindeki rolünü açıklayan bir liste ekleyin. Bu paketlerin kurulumunu sağlayan kodu (örneğin, `install.packages("dplyr")`) da ekleyebilirsiniz.

3.1.6 Çalışma Dosyasının Yapısı Hakkında Bilgi

Amaç: Dosyanın bölümlerini tanıtmak ve okuyucuya yol gösterici bir içerik tablosu sunmak.
Özelleşmiş Bilgi: Giriş, veri filtreleme, veri çoğaltma giderme vb. ana başlıkların kısa açıklamalarını yaparak, dosyanın genel yapısı hakkında bilgi verin.
R Markdown Önerisi: İçindekiler tablosu (`toc`) ekleyerek her başlığa tıklanabilir bağlantılar sunabilirsiniz.

3.1.7 İlerleyen Bölümlere Kısa Bir Giriş

- **Amaç:** Çalışmada ilerleyen başlıklara dair kısa bir ön bilgi verin.
- **Özelleşmiş Bilgi:** Her bir veri temizleme tekniğinin ne tür problemleri çözmek için kullanıldığını kısaca tanımlayın.
- **R Markdown Önerisi:** Kısa tanıtımlar ile birlikte, okuyucunun ilgisini çekecek bir anlatım stili kullanabilirsiniz. Her başlık hakkında kısa bir özet sunarak, okuyucunun ilgili teknik hakkında daha fazla bilgi edinmesini sağlayabilirsiniz.

3.2 Veri Filtreleme (Data Filtering)

3.2.1 Veri Filtrelemenin Amacı ve Doğru Kriterlerin Belirlenmesi

- **Amaç:** Veri filtrelemenin analiz sürecindeki önemini ve verinin doğruluğu ile güvenilirliğini nasıl artırdığını açıklayın. Filtrelemenin yalnızca gerekli verileri seçerek analiz sürecini nasıl hızlandırdığına değinin.
- **Kriter Belirleme:** Filtreleme işlemi için kullanılacak kriterleri belirlerken dikkat edilmesi gereken noktaları vurgulayın. Örneğin, analiz için belirli bir tarih aralığı, kategori veya değer aralığına göre veriyi daraltmanın analiz sonuçları üzerindeki etkilerini özetleyin.
- **R Markdown Önerisi:** Kısa bir tanıtım yazısı ve örnek veri kriterleri ekleyin. Kriterlerin nasıl belirleneceğini gösteren küçük bir tablo veya özet sunabilirsiniz.

3.2.2 Filtreleme Teknikleri ve Uygulama Örnekleri

- **Temel Teknikler:** R'de filtreleme için kullanılan `dplyr::filter()` gibi ana fonksiyonları tanıtın. Koşullu filtreleme yaparak veri setinde yalnızca belirli kayıtları seçmeyi gösterin.
- **Uygulama Örnekleri:** Tek ve çoklu koşul filtreleme örnekleri ile çeşitli filtreleme senaryolarını gösterin (örneğin, belirli bir kategoriyi veya bir değerin üstündeki kayıtları seçmek gibi). Her bir filtreleme işleminin nasıl yapıldığını ve analiz için nasıl anlam kazandığını açıklayın.

```
library(dplyr)
# Tek kriterle filtreleme
filtered_data_single <- data %>% filter(column_name == "specific_value")

# Çoklu kriterle filtreleme
filtered_data_multi <- data %>% filter(column1 > 10, column2 == "kategori")
```

3.2.3 Filtreleme Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Filtrelenmiş veriyi `summary()`, `head()` veya `skimr::skim()` ile inceleyerek seçilen verilerin filtreleme kriterlerine uygun olup olmadığını doğrulayın. Bu adım, filtreleme işleminin istenen sonuçları verip vermediğini kontrol etmek için gereklidir.
- **Değerlendirme:** Filtreleme sonrası elde edilen veri setinin analiz amacına uygunluğunu ve eksiksiz olup olmadığını değerlendirin. Filtreleme işlemi sonrasında veri setindeki kayıt sayısının azaldığına veya dağılımın nasıl değiştiğine dair kısa bir değerlendirme ekleyin.
- **R Markdown Önerisi:** Filtreleme sonrası veriyi gözden geçirin ve analiz için yeterli veri olup olmadığını değerlendirerek kısa bir yorum ekleyin.

```r
# Filtrelenmiş veriyi inceleme ve değerlendirme
summary(filtered_data_multi)
skimr::skim(filtered_data_multi)
```

3.3 Veri Çoğaltma Giderme (Data Deduplication)

3.3.1 Çoğaltma Giderme İşleminin Amacı ve Önemi

- **Amaç:** Veri çoğaltma giderme işleminin, analiz sürecindeki doğruluğu artırmada oynadığı önemli rolü vurgulayın. Çoğaltılmış verilerin analiz sonuçlarını yanıltabileceği ve gereksiz veri yükü oluşturabileceğini açıklayın.
- **Özelleşmiş Bilgi:** Çoğaltma giderme işleminin veri setinin kalitesini nasıl artırdığını ve doğru sonuçlara ulaşmayı nasıl kolaylaştırdığını özetleyin. Bu bölüm, analizde güvenilir veri kullanmanın önemini ve çoğaltma giderme işlemi yapılmadığında oluşabilecek sorunları ele almalıdır.
- **R Markdown Önerisi:** Çoğaltma giderme işleminin analiz üzerindeki etkisini görselleştiren bir örnek verin. Örneğin, veri setindeki tekrar eden kayıtları çıkarmadan ve çıkardıktan sonraki farklı veri sayısını gösteren bir tablo oluşturabilirsiniz.

3.3.2 Teknikler ve Uygulama Örnekleri

- **Temel Teknikler:** `dplyr` paketinde yer alan `distinct()` gibi temel R fonksiyonlarıyla çoğaltılmış verilerin nasıl temizleneceğini gösterin. `duplicated()` fonksiyonu ile yinelemeleri tespit etmeyi ve `distinct()` ile bunları gidermeyi tanıtın.
- **Uygulama Örnekleri:** Çoğaltma giderme işlemine dair çeşitli senaryolar sunun. Örneğin, yalnızca belirli bir değişken veya değişkenler kombinasyonuna göre yinelemeleri tespit etme ve giderme örneği verin.

```r
library(dplyr)
# Tüm sütunlara göre tekrarları kaldırma
unique_data <- data %>% distinct()

# Belirli sütunlara göre tekrarları kaldırma
unique_data_by_column <- data %>% distinct(column1, column2, .keep_all = TRUE)
```

3.3.3 Çoğaltma Giderme Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Çoğaltma giderme işlemi sonrasında veri setindeki kayıt sayısını kontrol edin ve önceki haliyle karşılaştırarak değişimi değerlendirin. Bu doğrulama adımı, temizleme işleminin başarılı olup olmadığını gösterecektir.

- **Değerlendirme:** Çoğaltma giderme işleminin veri seti üzerinde bıraktığı etkileri değerlendirin. Veri setinin orijinal hali ile çoğaltmaların temizlendiği hali arasındaki farkı özetleyin ve analizde kullanılabilirliğini gözden geçirin.
- **R Markdown Önerisi:** `nrow()` fonksiyonunu kullanarak, veri setinde çoğaltma giderme işlemi öncesi ve sonrası kayıt sayısını karşılaştıran kısa bir tablo veya özet ekleyin.

```
# Çoğaltma giderme öncesi ve sonrası kayıt sayısını karşılaştırma
original_count <- nrow(data)
deduplicated_count <- nrow(unique_data)
list(Orijinal_Kayit_Sayisi = original_count, Tekrarsiz_Kayit_Sayisi = deduplicated_count)
```

3.4 Veri Tamamlama (Data Imputation)

3.4.1 Veri Tamamlamanın Amacı ve Kullanım Alanları

- **Amaç:** Eksik verilerin analiz sonuçlarını nasıl etkileyebileceğini ve veri tamamlama işleminin bu etkiyi nasıl azaltabileceğini açıklayın. Eksik verilerin yanlış sonuçlara yol açabileceği durumlara değinin.
- **Kullanım Alanları:** Eksik veri ile başa çıkmak için veri tamamlama tekniklerinin kullanıldığı durumları tanıtın. Örneğin, tıbbi verilerde veya müşteri bilgilerinde eksik değerlerin analiz için tamamlanması gerektiğini vurgulayın.
- **R Markdown Önerisi:** Veri tamamlama işleminin analiz üzerindeki etkisini göstermek için eksik verilerin yer aldığı ve tamamlandığı örnek bir veri seti oluşturabilirsiniz.

3.4.2 Veri Tamamlama Yöntemleri ve Örnek Uygulama

- **Temel Teknikler:** `mice` ve `tidyr` paketlerindeki `fill()`, `replace_na()`, ve `mice()` fonksiyonları ile eksik verilerin nasıl tamamlanabileceğini gösterin.
- **Uygulama Örnekleri:** Farklı tamamlayıcı yöntemler sunun: basit bir ortalama veya medyanla doldurma, ileri-geri doldurma (`fill()`), ya da çoklu atama ile eksik verilerin tamamlanması.

```
library(tidyr)
library(mice)
# Basit ortalama ile eksik veri tamamlama
data$column_name[is.na(data$column_name)] <- mean(data$column_name, na.rm = TRUE)

# İleri-geri doldurma
data <- data %>% fill(column_name, .direction = "downup")

# Çoklu atama ile veri tamamlama
```

```r
imputed_data <- mice(data, m=5, maxit=50, meth='pmm', seed=500)
completed_data <- complete(imputed_data)
```

3.4.3 Tamamlama Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Eksik verilerin tamamlanmış olup olmadığını kontrol edin. Veri tamamlama işlemi sonrası tamamlanan verilerin istatistiksel olarak mantıklı olup olmadığını doğrulayın.
- **Değerlendirme:** Tamamlama işleminin veri setinin yapısını nasıl değiştirdiğini analiz edin. Örneğin, tamamlanmış veri ile orijinal veri arasındaki farkı karşılaştırarak analiz yapılabilirlik açısından uygun olup olmadığını gözden geçirin.
- **R Markdown Önerisi:** `summary()` veya `skimr::skim()` ile eksik verilerin tamamlanıp tamamlanmadığını analiz eden bir tablo veya özet ekleyin.

```r
# Tamamlama sonrası veri doğrulama
summary(completed_data)
skimr::skim(completed_data)
```

3.5 Veri Standardizasyonu (Data Standardization)

3.5.1 Veri Standardizasyonunun Amacı ve Önemi

- **Amaç:** Veri standardizasyonunun, analiz sürecinde değişkenler arasındaki karşılaştırılabilirliği nasıl artırdığını açıklayın. Standart hale getirilmemiş verilerin analizde yanılgıya yol açabileceği durumlara değinin.
- **Önemi:** Farklı ölçeklere sahip değişkenlerin karşılaştırılabilmesi için standardizasyonun gerekliliğini vurgulayın. Örneğin, yaş ve gelir gibi farklı ölçeklerdeki verilerin ortak bir ölçeğe çekilmesi gerektiğini açıklayın.
- **R Markdown Önerisi:** Standart hale getirilmiş ve getirilmemiş verilerin karşılaştırılabilirliğini göstermek için basit bir örnek veri seti kullanın.

3.5.2 Veri Standardizasyon Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** `scale()` gibi R fonksiyonları ile veriyi normalize etme ve standartlaştırma yöntemlerini tanıtın.
- **Uygulama Örnekleri:** Ortalama ve standart sapma kullanarak veriyi standart hale getirme (z-skoru hesaplama), minimum-maksimum ölçekleme gibi yöntemler sunun. Örnek veri üzerinden her iki yöntemi de uygulayın.

```r
# Ortalama ve standart sapma ile veri standardizasyonu (z-skoru)
standardized_data <- scale(data$numeric_column)

# Min-Max ölçekleme
min_val <- min(data$numeric_column, na.rm = TRUE)
max_val <- max(data$numeric_column, na.rm = TRUE)
data$min_max_scaled <- (data$numeric_column - min_val) / (max_val - min_val)
```

3.5.3 Standardizasyon Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Verinin standart hale getirilip getirilmediğini `summary()` veya `sd()` fonksiyonları ile doğrulayın. Örneğin, standart hale getirilen bir sütunun ortalamasının 0 ve standart sapmasının 1 olup olmadığını kontrol edin.
- **Değerlendirme:** Standardizasyon işleminin veri seti üzerindeki etkilerini analiz edin. Analiz yapılabilirlik açısından verinin uygun olup olmadığını değerlendirin.
- **R Markdown Önerisi:** Standardizasyon sonrası veriyi doğrulayan bir tablo veya özet ekleyin.

```r
# Standardizasyon sonrası doğrulama
summary(standardized_data)
sd(standardized_data)  # Standart sapmanın 1 olup olmadığını kontrol edin
```

3.6 Veri Dönüştürme (Data Transformation)

3.6.1 Veri Dönüştürmenin Amacı ve Önemi

- **Amaç:** Veri dönüştürmenin analiz sürecinde veriyi daha anlamlı hale getirmek ve analiz edilebilirliği artırmak için nasıl kullanıldığını açıklayın.
- **Önemi:** Özellikle kategorik verileri sayısal verilere dönüştürme, değişkenleri logaritmik veya kök dönüşümü gibi işlemlerden geçirmenin analiz sonuçlarını nasıl etkilediğini ele alın.
- **R Markdown Önerisi:** Dönüştürülmüş ve dönüştürülmemiş veri örnekleri üzerinden analiz sonuçlarının nasıl değiştiğini gösterebilirsiniz.

3.6.2 Veri Dönüştürme Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** `mutate()` ve `log()`, `sqrt()` gibi fonksiyonları kullanarak veri dönüştürme işlemlerini tanıtın. Kategorik verileri sayısal verilere dönüştürme (örneğin `factor()`) ve değişkenleri logaritmik dönüşümden geçirme gibi yöntemler sunun.
- **Uygulama Örnekleri:** Sayısal veri dönüşümü, logaritmik dönüşüm, kök dönüşümü ve kategorik veriyi faktör olarak tanımlama örneklerini gösterin.

```r
library(dplyr)
# Logaritmik dönüşüm
data <- data %>% mutate(log_transformed = log(numeric_column))

# Kök dönüşümü
data <- data %>% mutate(sqrt_transformed = sqrt(numeric_column))

# Kategorik veriyi faktör olarak dönüştürme
data$category_column <- as.factor(data$category_column)
```

3.6.3 Dönüştürme Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Dönüştürme işlemi sonrası verinin doğruluğunu ve analiz edilebilirliğini kontrol edin. Örneğin, log dönüşüm sonrası değerlerin dağılımını veya kategorik verinin uygun şekilde faktör olarak ayarlandığını kontrol edin.
- **Değerlendirme:** Dönüşüm işleminin veri seti üzerindeki etkilerini analiz edin. Özellikle sayısal dönüşümden sonra verinin anlamlı ve uygun dağılıma sahip olup olmadığını değerlendirin.
- **R Markdown Önerisi:** Dönüşüm sonrası verinin özetini sunarak dönüşümün veriyi nasıl değiştirdiğini gözlemleyin.

```r
# Dönüşüm sonrası veri doğrulama
summary(data$log_transformed)
summary(data$sqrt_transformed)
summary(data$category_column)
```

3.7 Aykırı Değer Tespiti (Outlier Detection)

3.7.1 Aykırı Değerlerin Önemi ve Belirlenmesi

- **Amaç:** Aykırı değerlerin analiz sürecinde veri dağılımını nasıl bozabileceğini ve analiz sonuçlarını nasıl etkileyebileceğini açıklayın. Aykırı değerlerin yanıltıcı sonuçlar yaratabileceği durumlara değinin.
- **Belirlenmesi:** Aykırı değerlerin hangi koşullarda önemli olduğunu ve hangi durumlarda temizlenmesi gerektiğini özetleyin.
- **R Markdown Önerisi:** Aykırı değerlerin önemini göstermek için aykırı değerler içeren ve temizlenmiş veri örnekleri vererek bu farkı görselleştirin.

3.7.2 Aykırı Değer Belirleme Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** `boxplot()`, `IQR()` ve `z-score` yöntemleriyle aykırı değerleri tespit etme yöntemlerini tanıtın.
- **Uygulama Örnekleri:** IQR yöntemi ile aykırı değer sınırlarını belirleme ve z-skoru kullanarak aşırı uç noktaları tespit etme örneklerini gösterin.

```
# IQR yöntemiyle aykırı değer sınırları
Q1 <- quantile(data$numeric_column, 0.25, na.rm = TRUE)
Q3 <- quantile(data$numeric_column, 0.75, na.rm = TRUE)
IQR_value <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value
outliers_iqr <- data %>% filter(numeric_column < lower_bound | numeric_column > upper_bound)

# Z-skoru ile aykırı değer belirleme
z_scores <- scale(data$numeric_column)
outliers_z <- data %>% filter(abs(z_scores) > 3)
```

3.7.3 Aykırı Değerlerin Değerlendirilmesi ve İşleme

- **Değerlendirme:** Aykırı değerlerin analiz üzerinde etkili olup olmadığını değerlendirin. Analiz için kritik olmayan durumlarda aykırı değerlerin nasıl çıkarılabileceğini veya ele alınabileceğini açıklayın.
- **İşleme:** Aykırı değerleri analiz dışı bırakma, dönüştürme ya da farklı yöntemlerle aykırı değerlerin etkisini azaltma tekniklerini gösterin.
- **R Markdown Önerisi:** Aykırı değerlerin temizlenip temizlenmemesine karar vermek için veri dağılımını analiz eden bir özet veya görselleştirme ekleyin.

```
# Aykırı değerleri temizleme örneği
cleaned_data <- data %>% filter(numeric_column >= lower_bound, numeric_column <= upper_bound)
summary(cleaned_data)
boxplot(cleaned_data$numeric_column)
```

3.8 Veri Doğrulama (Data Validation)

3.8.1 Veri Doğrulamanın Amacı ve Önemi

- **Amaç:** Veri doğrulamanın, veri kalitesini artırmada ve hatalı verilerin analiz sonuçlarını bozmasını önlemedeki önemini açıklayın. Analize başlamadan önce verinin güvenilirliğini sağlamak

için doğrulama adımlarının gerekli olduğuna değinin.
- **Önemi:** Eksik veya hatalı veri girişlerinin analiz üzerindeki etkilerini ele alın. Örneğin, veri doğrulama yapılmadığında analiz sonuçlarında oluşabilecek sapmaları vurgulayın.
- **R Markdown Önerisi:** Veri doğrulamanın etkisini göstermek için doğrulanmamış ve doğrulanmış veri örnekleriyle analiz sonuçlarını karşılaştırın.

3.8.2 Veri Doğrulama Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** `assertive` paketi veya temel R fonksiyonları ile veri doğrulama yöntemlerini tanıtın. Örneğin, değişkenlerin veri türü, değer aralıkları veya eksik değer kontrolü gibi doğrulama adımlarını açıklayın.
- **Uygulama Örnekleri:** Sayısal değişkenlerin pozitif olması, kategorik verilerin belirli sınırlarda olması gibi doğrulama işlemlerini örneklerle gösterin.

```
library(assertive)
# Sayısal sütunun pozitif değerler içermesini doğrulama
assert_all_are_positive(data$numeric_column)
# Belirli bir kategorik sınıf içinde kalmasını doğrulama
assert_all_are_in_set(data$category_column,c("A","B","C"))
# Eksik değerlerin olmadığını kontrol etme
assert_all_are_not_na(data$important_column)
```

3.8.3 Doğrulama Sonrası Değerlendirme ve Düzeltme

- **Değerlendirme:** Veri doğrulama sonrasında eksik veya hatalı veri bulunup bulunmadığını kontrol edin ve doğrulama sürecinin başarılı olup olmadığını değerlendirin.
- **Düzeltme:** Doğrulama sırasında bulunan hataları veya eksiklikleri giderme yöntemlerini açıklayın. Örneğin, eksik verileri tamamlama veya yanlış değerleri düzeltme adımlarını gösterin.
- **R Markdown Önerisi:** Doğrulama sonrası veri yapısını özetleyen bir tablo veya analiz ekleyin.

```
# Eksik veya hatalı verileri inceleme ve düzeltme
summary(data)
data$numeric_column[is.na(data$numeric_column)] <- mean(data$numeric_column,na.rm=TRUE)
```

3.9 Veri Kodlama (Data Encoding)

3.9.1 Veri Kodlamanın Amacı ve Önemi

- **Amaç:** Veri kodlamanın, özellikle kategorik verileri sayısal formata dönüştürerek analiz ve modelleme sürecinde kolaylık sağladığını açıklayın.

- **Önemi:** Kategorik verilerin kodlanmaması durumunda analizde karşılaşılan zorlukları ve kodlamanın bu sorunları nasıl çözdüğünü vurgulayın. Kodlanmış verilerin makine öğrenimi algoritmalarında kullanılabilirliği açısından önemini belirtin.
- **R Markdown Önerisi:** Kodlanmamış ve kodlanmış veri örneklerini karşılaştırarak kodlamanın veriye katkısını görselleştirin.

3.9.2 Veri Kodlama Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** Kategorik verileri faktör olarak kodlama, one-hot encoding gibi yaygın yöntemleri tanıtın. `as.factor()` ile faktör kodlama ve `model.matrix()` ile one-hot encoding işlemlerini gösterin.
- **Uygulama Örnekleri:** Basit faktör kodlama ve one-hot encoding örnekleri sunarak bu işlemleri açıklayın.

```
# Faktör kodlama
data$category_column<-as.factor(data$category_column)

# One-hot encoding
one_hot_encoded_data<-model.matrix(~category_column-1,data=data)
```

3.9.3 Kodlama Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Kodlama işlemi sonrası verinin doğru şekilde kodlanıp kodlanmadığını kontrol edin. Örneğin, kategorik bir değişkenin her sınıfı için ayrı sütun oluşturulduğundan emin olun.
- **Değerlendirme:** Kodlama sonrası veri yapısını analiz ederek veri dönüşümünün veriyi analiz için uygun hale getirip getirmediğini değerlendirin.
- **R Markdown Önerisi:** Kodlama sonrası veri yapısını özetleyen bir tablo veya özet ekleyin.

```
# Kodlama sonrası veri doğrulama
summary(one_hot_encoded_data)
str(one_hot_encoded_data)
```

3.10 Veri Birleştirme (Data Aggregation)

3.10.1 Veri Birleştirmenin Amacı ve Önemi

- **Amaç:** Veri birleştirmenin, veri analizinde özet bilgi elde etmek ve büyük veri setlerini daha yönetilebilir hale getirmek için nasıl kullanıldığını açıklayın.

- **Önemi:** Veri birleştirmenin, özellikle kategori, zaman veya belirli gruplar üzerinden yapılan analizlerde veri anlamlandırma ve analiz sürecini kolaylaştırma açısından önemini vurgulayın.
- **R Markdown Önerisi:** Birleştirilmiş ve birleştirilmemiş veri örnekleri üzerinden, birleştirmenin veri analizi üzerindeki etkisini görselleştirin.

3.10.2 Veri Birleştirme Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** `dplyr` paketindeki `group_by()` ve `summarize()` fonksiyonları ile veri birleştirme işlemlerini tanıtın.
- **Uygulama Örnekleri:** Veriyi belirli kategoriler veya zaman aralıklarına göre gruplama ve her grup için ortalama, toplam gibi özet istatistikleri hesaplama örnekleri gösterin.

```
library(dplyr)
# Kategoriye göre birleştirme ve ortalama hesaplama
aggregated_data<-data%>%group_by(category_column)%>%summarize(mean_value=mean(numeric_column

# Zaman aralığına göre birleştirme ve toplam hesaplama
time_aggregated_data<-data%>%group_by(time_column)%>%summarize(total=sum(numeric_column,na.rm
```

3.10.3 Birleştirme Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Birleştirme işlemi sonrası elde edilen veriyi kontrol edin ve özet istatistiklerin doğruluğunu gözden geçirin. Örneğin, birleştirme sonrası her grubun temsil ettiği değerin mantıklı olup olmadığını kontrol edin.
- **Değerlendirme:** Birleştirme işlemi sonrası verinin analiz için daha anlamlı hale gelip gelmediğini değerlendirin ve grupların anlamlı özet bilgiler içerip içermediğini analiz edin.
- **R Markdown Önerisi:** Birleştirme sonrası veri yapısını özetleyen bir tablo veya özet ekleyin.

```
# Birleştirme sonrası veri doğrulama
summary(aggregated_data)
summary(time_aggregated_data)
```

3.11 Veri Örnekleme (Data Sampling)

3.11.1 Veri Örneklemenin Amacı ve Önemi

- **Amaç:** Veri örneklemenin, büyük veri setlerinden analiz için temsilci bir alt küme elde etmek amacıyla nasıl kullanıldığını açıklayın.

- **Önemi:** Özellikle büyük veri setleri ile çalışırken analiz süresini kısaltmak ve sistem kaynaklarını etkin kullanmak için veri örneklemenin önemini vurgulayın.
- **R Markdown Önerisi:** Örneklenmiş ve tam veri seti üzerinden analiz sonuçlarını karşılaştırarak örneklemenin analiz üzerindeki etkisini gösterin.

3.11.2 Veri Örnekleme Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** R'de basit rastgele örnekleme ve tabakalı örnekleme gibi yöntemleri tanıtın. `sample_n()` ve `sample_frac()` fonksiyonları ile veri örnekleme işlemlerini gösterin.
- **Uygulama Örnekleri:** Belirli bir sayıda rastgele örnek seçme, veri setinin belli bir yüzdesini örnekleme ve belirli gruplardan örnek çekme gibi örnekler sunun.

```
library(dplyr)
# Basit rastgele örnekleme
random_sample<-data%>%sample_n(100)

# Belirli bir yüzdeyle örnekleme
fractional_sample<-data%>%sample_frac(0.1)
```

3.11.3 Örnekleme Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Örnekleme işlemi sonrası elde edilen verinin ana veri setini temsil edip etmediğini kontrol edin. Örneğin, örneklenmiş verinin ana veri seti ile benzer dağılım özelliklerine sahip olup olmadığını inceleyin.
- **Değerlendirme:** Örnekleme işleminin analiz sonuçlarını nasıl etkilediğini değerlendirerek örnek verinin analiz için yeterli olup olmadığını belirleyin.
- **R Markdown Önerisi:** Örneklenmiş veri ve tam veri seti üzerindeki özet istatistikleri karşılaştırarak temsil yeteneğini analiz edin.

```
# Örnekleme sonrası veri doğrulama
summary(random_sample)
summary(fractional_sample)
```

3.12 Veri Temizleme (Data Cleansing)

3.12.1 Veri Temizlemenin Amacı ve Önemi

- **Amaç:** Veri temizlemenin, analiz için doğru ve güvenilir bir veri seti elde etmek amacıyla yapılan işlemler olduğunu açıklayın.

- **Önemi:** Hatalı, eksik veya gereksiz verilerin çıkarılmasının, analiz sonuçlarının doğruluğunu ve güvenilirliğini nasıl artırdığını vurgulayın. Veri temizliğinin analiz sürecindeki temel adımlardan biri olduğunu belirtin.
- **R Markdown Önerisi:** Temizlenmemiş ve temizlenmiş veri setlerini karşılaştırarak veri temizlemenin analiz üzerindeki etkisini gösterin.

3.12.2 Veri Temizleme Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** Eksik değerlerin giderilmesi, aykırı değerlerin tespiti ve kaldırılması, gereksiz sütun veya satırların çıkarılması gibi veri temizleme işlemlerini tanıtın.
- **Uygulama Örnekleri:** `na.omit()`, `filter()`, `select()` gibi fonksiyonlarla eksik verileri kaldırma, belirli koşullara göre satırları filtreleme ve gereksiz sütunları çıkarma gibi örnekler sunun.

```
library(dplyr)
# Eksik değerleri kaldırma
cleaned_data<-na.omit(data)

# Belirli koşullara göre satırları filtreleme (örneğin, aykırı değerleri çıkarma)
cleaned_data<-data%>%filter(numeric_column<upper_bound,numeric_column>lower_bound)

# Gereksiz sütunları çıkarma
cleaned_data<-data%>%select(-unnecessary_column)
```

3.12.3 Temizleme Sonrası Doğrulama ve Değerlendirme

- **Doğrulama:** Temizleme işlemi sonrası veriyi kontrol ederek istenmeyen veya hatalı verilerin tamamen kaldırıldığından emin olun.
- **Değerlendirme:** Veri temizleme işleminin veri yapısını nasıl değiştirdiğini ve analiz için veriyi daha uygun hale getirip getirmediğini değerlendirin.
- **R Markdown Önerisi:** Temizlenmiş veri setini özetleyen bir tablo veya analiz sunarak temizleme işleminin etkisini görselleştirin.

```
# Temizleme sonrası veri doğrulama
summary(cleaned_data)
str(cleaned_data)
```

3.13 Veri Profilleme (Data Profiling)

3.13.1 Veri Profillemenin Amacı ve Önemi

- **Amaç:** Veri profillemenin, veri setinin yapısını anlamak ve veri kalitesini değerlendirmek amacıyla yapıldığını açıklayın.
- **Önemi:** Veri profillemenin, veri temizleme ve analiz adımlarında hangi işlemlerin gerektiğini belirlemede nasıl yardımcı olduğunu vurgulayın. Verinin genel dağılımı, eksik değer oranı, aykırı değerlerin varlığı gibi bilgilerin analiz sürecini nasıl şekillendirdiğine değinin.
- **R Markdown Önerisi:** Profil çıkarılmamış ve çıkarılmış veri setlerini karşılaştırarak veri profillemenin analiz üzerindeki etkisini gösterin.

3.13.2 Veri Profilleme Yöntemleri ve Uygulama Örnekleri

- **Temel Teknikler:** `summary()`, `str()`, `skimr::skim()` gibi fonksiyonlarla veri yapısını inceleyin. Verinin özet istatistiklerini, veri türlerini ve eksik değer oranlarını ortaya koyan veri profilleme işlemlerini gösterin.
- **Uygulama Örnekleri:** Değişkenlerin dağılımı, veri türleri, eksik ve aykırı değerlerin oranlarını analiz ederek veri profilleme örnekleri sunun.

```
library(skimr)
# Veri yapısını özetleme
summary(data)

# Veri türleri ve genel yapı bilgisi
str(data)

# Detaylı veri profilleme
skim(data)
```

3.13.3 Profilleme Sonrası Değerlendirme ve Kullanım

- **Değerlendirme:** Veri profilleme sonrası elde edilen bilgileri kullanarak veri setinin temizleme ve analiz işlemleri için hazır olup olmadığını değerlendirin.
- **Kullanım:** Profilleme sonuçlarına göre veride yapılması gereken işlemleri belirleyin, örneğin eksik değer tamamlama veya aykırı değerlerin ele alınması gibi.
- **R Markdown Önerisi:** Profil çıkarılmış veri setinin özetini sunarak yapılması gereken işlemleri belirten kısa bir analiz ekleyin.

```r
# Profilleme sonrası veri değerlendirme
summary(data)
skim(data)
```

3.14 Sonuç

3.14.1 Çalışma Özeti ve Elde Edilen Bulgular

- **Amaç:** Çalışmanın temel amacını ve veri temizleme süreçlerinin önemini kısaca özetleyin. Yapılan adımları (filtreleme, çoğaltma giderme, tamamlama, vb.) ve her bir adımın analiz sürecine katkısını vurgulayın.
- **Elde Edilen Bulgular:** Her teknikle ilgili olarak elde edilen bulguları kısaca özetleyin. Örneğin, veri temizleme sürecinde eksik verilerin tamamlanması veya aykırı değerlerin temizlenmesiyle verinin güvenilirliğinin nasıl artırıldığını belirtin.
- **R Markdown Önerisi:** Çalışmada yer alan her ana adımı kısaca özetleyen bir liste veya tablo ekleyin.

3.14.2 Çalışmanın Analiz Sürecine Katkısı

- **Analize Etkisi:** Veri temizleme ve profilleme işlemlerinin genel analiz sürecine nasıl katkı sağladığını açıklayın. Analiz sonuçlarının doğruluğuna ve güvenilirliğine olan etkisini belirtin.
- **Gelecek Çalışmalara Katkısı:** Bu veri temizleme süreçlerinin gelecekteki projelerde nasıl bir rehber niteliği taşıyabileceğine değinin. Veri temizleme adımlarının tekrarlanabilirliğini ve diğer veri setlerine uygulanabilirliğini vurgulayın.
- **R Markdown Önerisi:** Çalışmanın analiz sürecine ve gelecekteki projelere katkısını özetleyen kısa bir metin veya görselleştirme ekleyin.

3.14.3 Sonuç ve Öneriler

- **Genel Sonuç:** Veri temizleme işlemlerinin başarıyla tamamlanması ile elde edilen son veri setinin analize hazır hale geldiğini ifade edin. Bu sonuçların çalışmanın genel hedefine nasıl katkı sağladığını özetleyin.
- **Öneriler:** Gelecek çalışmalarda veri temizleme işlemleri sırasında dikkat edilmesi gereken noktaları belirtin. Örneğin, veri kaynağı doğrulaması, daha gelişmiş aykırı değer tespit yöntemleri veya eksik veri stratejileri gibi öneriler sunun.
- **R Markdown Önerisi:** Sonuç kısmında gelecekte yapılacak çalışmalara yönelik öneriler sunarak, okuyucuların veri temizleme süreçlerine daha stratejik yaklaşmalarını sağlayın.

```r
# Sonuç özetini ve önerileri göstermek için bir örnek veri özeti
final_summary <- summary(cleaned_data)
final_summary
```

Referanslar

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.