Datensatz: Iris-Datensatz

Der **Iris-Datensatz** ist ein berühmter Datensatz in der Statistik und im maschinellen Lernen, der ursprünglich von dem Biologen und Statistiker **Ronald A. Fisher** veröffentlicht wurde. Er enthält Messdaten von 150 Iris-Blumen und wird oft zur Demonstration von Klassifikationsmethoden und grundlegenden statistischen Analysen verwendet.

Struktur des Iris-Datensatzes

Der Datensatz umfasst **150 Beobachtungen** (Zeilen) und **5 Variablen** (Spalten). Die ersten vier Variablen sind numerische Messwerte von verschiedenen Teilen der Blume, und die fünfte Variable ist die Art (Species) der Iris. Die drei Arten im Datensatz sind:

1. **setosa**
2. **versicolor**
3. **virginica**

Variablen im Detail

1. **Sepal.Length**: Länge des Kelchblatts (Sepal) in Zentimetern.
2. **Sepal.Width**: Breite des Kelchblatts in Zentimetern.
3. **Petal.Length**: Länge des Blütenblatts (Petal) in Zentimetern.
4. **Petal.Width**: Breite des Blütenblatts in Zentimetern.
5. **Species**: Art der Iris-Blume (faktorielle Variable mit den Kategorien *setosa*, *versicolor* und *virginica*).

Beispiel für die ersten Zeilen des Datensatzes

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	versicolor

Bedeutung der Variablen für Analysen

- **Sepal.Length und Sepal.Width**: Die Messungen des Kelchblatts bieten Informationen zur Größe und Form des äußeren Teils der Blume.
- **Petal.Length und Petal.Width**: Die Messungen des Blütenblatts sind oft stärker zwischen den Arten differenziert als die Sepal-Werte und können gut zur Klassifikation der Arten genutzt werden.

- **Species**: Diese Variable ist die Ziel- oder Klassifikationsvariable, die angibt, zu welcher Art die jeweilige Beobachtung gehört.

Der Iris-Datensatz ist vielseitig einsetzbar, um statistische Zusammenfassungen, Visualisierungen und einfache Klassifikationsmethoden zu üben. Da er bereits in R integriert ist, kann er leicht für Analysen verwendet werden.

Aufgaben –(Markdown Dokument inkl. Kommentaare & Output Interpretation)

1. Laden des Datensatzes

- **Aufgabe:** Laden Sie den Iris-Datensatz in Ihre R-Umgebung und zeigen Sie die ersten 6 Zeilen an.
- **Hinweis:** Verwenden Sie head() um die ersten Zeilen des Datensatzes anzuzeigen.

2. Datenstruktur erkunden

- **Aufgabe:** Untersuchen Sie die Struktur des Datensatzes.
- **Hinweis:** Verwenden Sie die Funktion str() um die Struktur des Datensatzes zu sehen, einschließlich der Datentypen und der Anzahl der Zeilen.

3. Statistische Zusammenfassung

- **Aufgabe:** Erstellen Sie eine statistische Zusammenfassung des Datensatzes.
- **Hinweis:** Die Funktion summary() gibt Ihnen einen Überblick über die wichtigsten statistischen Maße (Minimum, 1. Quartil, Median, Mittelwert, 3. Quartil und Maximum) für jede Spalte.

4. Berechnung von Mittelwert und Median

- **Aufgabe:** Berechnen Sie den Mittelwert und den Median der Sepal Length (Sepal-Länge).
- **Hinweis:** Verwenden Sie mean() für den Mittelwert und median() für den Median. Vergessen Sie nicht, die Ergebnisse mit cat() oder print() auszugeben.

5. Berechnung des Modus

- **Aufgabe:** Erstellen Sie eine Funktion, um den Modus der Sepal Width (Sepal-Breite) zu berechnen.
- **Hinweis:** Erstellen Sie eine Fct: modus.

```
modus <- function(x) {
  uniq_x <- unique(x)  # Einzigartige Werte im Vektor
  uniq_x[which.max(tabulate(match(x, uniq_x)))]  # Modus-Berechnung
}
```

Dann Anwendung der Funktion auf der Zielvariable.

6. Histogramm erstellen

- **Aufgabe:** Erstellen Sie ein Histogramm für die Sepal Length und interpretieren Sie die Ergebnisse.
- **Hinweis:** Verwenden Sie ggplot2, um ein Histogramm zu erstellen. Achten Sie darauf, die x-Achse als Sepal Length zu definieren. → Mehr info auf Seite5

7. Boxplot erstellen

- **Aufgabe:** Erstellen Sie ein Boxplot für die Sepal Width nach Art (Species) und interpretieren Sie die Ergebnisse.
- **Hinweis:** In ggplot(), setzen Sie aes(x = Species, y = Sepal.Width).
  → Mehr info auf Seite5

8. Gruppierung und Aggregation

- **Aufgabe:** Berechnen Sie den Mittelwert der Sepal Length für jede Art (Species).
- **Hinweis:** Verwenden Sie die Funktion aggregate(), um den Mittelwert für jede Art zu berechnen. Der allgemeine Aufbau ist aggregate(<Zielvariable> ~ <Gruppierungsvariable>, data = <Datensatz>, FUN = mean).

9. Erstellen einer neuen Variablen

- **Aufgabe:** Fügen Sie eine neue Spalte hinzu, die angibt, ob die Sepal Length größer als 5.0 ist (Ja/Nein).
- **Hinweis:** Verwenden Sie ifelse(), um diese neue Variable zu erstellen. Zum Beispiel: ifelse(daten$Sepal.Length > 5.0, "Ja", "Nein").

10. Speichern des bearbeiteten Datensatzes

- **Aufgabe:** Speichern Sie den bearbeiteten Datensatz in einer CSV-Datei.
- **Hinweis:** Verwenden Sie write.csv(), um den Datensatz zu speichern. Geben Sie den Dateinamen und row.names = FALSE an, um die Zeilennamen nicht zu speichern.

11. Filterung der Daten

- **Aufgabe:** Filtern Sie den Datensatz, sodass nur die Zeilen mit Sepal.Length größer als 6.0 enthalten sind.
- **Hinweis:** Verwenden Sie den subset()-Befehl oder die eckigen Klammern [], um nur die Zeilen auszuwählen, die die Bedingung erfüllen.

12. Berechnung der Standardabweichung

- **Aufgabe:** Berechnen Sie die Standardabweichung der Petal.Length (Blütenblatt-Länge) für alle Einträge im Datensatz.
- **Hinweis:** Die Funktion sd() kann zur Berechnung der Standardabweichung verwendet werden.

13. Berechnung des Interquartilsabstands (IQR)

- **Aufgabe:** Berechnen Sie den Interquartilsabstand (IQR) der Sepal Width für alle Einträge im Datensatz.
- **Hinweis:** Der Interquartilsabstand gibt die Spannweite der mittleren 50 % der Daten an und kann mit IQR()berechnet werden.

14. Streudiagramm erstellen

- **Aufgabe:** Erstellen Sie ein Streudiagramm (Scatterplot) für Sepal.Length und Petal.Length. Stellen Sie sicher, dass die Arten (Species) in verschiedenen Farben angezeigt werden.

- **Hinweis:** Verwenden Sie ggplot2 und legen Sie Species als Farbe (color) fest, um die Arten farblich zu unterscheiden → Mehr info auf Seite5

15. Umbenennen einer Spalte

- **Aufgabe:** Benennen Sie die Spalte Sepal.Length in SepalLength_cm um.
- **Hinweis:** Verwenden Sie die Funktion names(), um die Namen der Spalten anzupassen.

16. Maximal- und Minimalwerte berechnen

- **Aufgabe:** Finden Sie den maximalen und minimalen Wert der Petal.Length und geben Sie diese Werte aus.
- **Hinweis:** Die Funktionen max() und min() geben die maximalen und minimalen Werte eines Vektors zurück.

17. Anzahl der Einträge pro Art zählen

- **Aufgabe:** Zählen Sie, wie viele Einträge es für jede Art (Species) gibt.
- **Hinweis:** Verwenden Sie die Funktion table(), um die Häufigkeiten für die Variable Species zu berechnen.

18. Datensatz sortieren

- **Aufgabe:** Sortieren Sie den Datensatz nach Sepal.Length in absteigender Reihenfolge.
- **Hinweis:** Die Funktion order() kann verwendet werden, um die Reihenfolge eines Data Frames basierend auf einer bestimmten Spalte zu ändern.

19. Zeilen mit fehlenden Werten identifizieren

- **Aufgabe:** Überprüfen Sie, ob im Datensatz fehlende Werte (NA-Werte) vorhanden sind.
- **Hinweis:** Verwenden Sie die Funktionen is.na() und any() oder sum() auf is.na(), um NA-Werte zu finden.

20. Durchschnittliche Petal.Width für jede Art (Species) berechnen

- **Aufgabe:** Berechnen Sie die durchschnittliche Petal Width für jede Art und speichern Sie das Ergebnis in einem neuen Data Frame.
- **Hinweis:** Die Funktion aggregate() kann wie in Aufgabe 8 verwendet werden, um den Mittelwert zu berechnen.

Allgemeine Hinweise zur Erstellung von Plots:

Histogram

```
ggplot(data, aes(x = variable)) +
geom_histogram(binwidth = breite, fill = "Farbe", color ="Randfarbe") +
labs( title = "Titel des Histogramms",
      x = "Beschriftung der X-Achse",
      y = "Beschriftung der Y-Achse" )
```

Boxplot

```
ggplot(data, aes(x = faktorvariable, y = numerische_variable)) +
geom_boxplot(fill = "Farbe", color = "Randfarbe") +
labs( title = "Titel des Boxplots",
      x = "Beschriftung der X-Achse",
      y = "Beschriftung der Y-Achse" )
```

Streudiagramm/ Density plot

```
ggplot(data, aes(x = x_variable, y = y_variable)) +
geom_point(color = "Punktfarbe", size = punktgröße) +
labs( title ="Titel des Streudiagramms",
      x = "Beschriftung der X-Achse",
      y = "Beschriftung der Y-Achse" )
```