Allgemeines

Die vorgesehene Bearbeitungszeit für das Übungsblatt endet am Freitag, 20.11.2020, um 12:00 Uhr. Kurze Zeit später werden die Lösung und das nächste Übungsblatt hochgeladen.

Falls Sie anonymes Peer-Feedback zu Ihrem Code erhalten wollen, laden Sie Ihre Lösung zu den Programmieraufgaben innerhalb der Frist in das ILIAS-Übungsobjekt "Abgaben" hoch. Tun Sie dies bitte nur, wenn auch Sie bereit sind, zu den Lösungen zweier Kommiliton:innen Feedback zu geben. Anderseits ist für Sie kein Feedback sichtbar und Ihre zufällig zugelosten Partner:innen warten ebenfalls vergebens. Feedback kann erst nach Ablauf der Abgabefrist gegeben werden. Geben Sie das Feedback bis zum Mittwoch, 25.11.2020, 23:55 Uhr. Erst nach Ablauf dieser Feedbackfrist wird dann das erhaltene Feedback sichtbar.

In der Form des Feedbacks sind Sie frei. Sie können gerne Stichpunkte verwenden. Grundidee ist, dass Sie vom Code anderer lernen können und wiederum Ihrerseits Komiliton:innen Verbesserungsvorschläge machen. Beispielsweise können Sie anmerken, wenn Teile des Codes einfacher, eleganter oder effizienter geschrieben werden könnten. Achten Sie darauf, konstruktiv zu bleiben und auch positive Punkte zu erwähnen.

1 Online-Test 1

Der erste Online-Test steht ab dem 02.11.2020, 00:00 Uhr im ILIAS bereit. Er beschäftigt sich mit den Vorlesungskapiteln 1 (Einleitung) und 2 (statistische Grundlagen, bis inklusive der statistischen Tests).

2 Deskriptive Statistik und Grundlagen von R

Ziel dieser Aufgabe ist es, grundlegende Datenstrukturen und Sprachkonstrukte von R kennenzulernen und sie für deskriptive Statistik anzuwenden.

a) [**Tutorial**] Im ILIAS steht ein PDF bereit, welche die Grundlagen von R erläutert. Dies soll Ihnen lediglich einen groben Überblick vermitteln und als Nachschlagewerk dienen. Der Fokus der Übungen liegt auf der Anwendung von R auf spezifische Data-Mining-Probleme, nicht auf dem umfänglichen Erlernen der Sprache.

b) [**Standardabweichung (1)**] Schreiben Sie eine Funktion `stddev()`, welche als Parameter einen Vektor `x` und einen Bool'schen Wert `population` erhält. Die Funktion soll mittels einer Schleife die Standardabweichung berechnen. Falls `population` den Wert `TRUE` annimmt, soll mit der Vektorlänge n normalisiert werden, ansonsten mit $n - 1$. Die Funktion soll eine Liste mit den Komponenten `result` und `n` zurückgeben. Falls `x` einen fehlenden Wert (`NA`) enthält, soll die Funktion mittels `stopifnot()` eine Exception werfen. *(Basis)*

c) [**Standardabweichung (2)**] Berechnen Sie für einen zufälligen, normalverteilten Vektor die Differenz zwischen dem Ergebnis Ihrer Funktion und der eingebauten Funktion `sd()`. Wiederholen Sie dies, sodass Sie einen Vektor von Differenzen erhalten. Geben Sie eine statistische Zusammenfassung des Differenzvektors aus und plotten Sie ein Histogramm. Wie interpretieren Sie die Ergebnisse? *(Basis)*

d) [**Standardabweichung (3)**] Nutzen Sie das Paket `microbenchmark`, um die Performance Ihrer Schleifen-basierten Funktion mit der eingebauten Funktion `sd()` zu vergleichen. Ersetzen Sie die Schleife durch `sum()` und wiederholen Sie den Benchmark.

Welche Schlussfolgerungen ziehen Sie aus dem Vergleich? *(Vertiefung)*

e) **[Plots (1)]** Nutzen Sie den weltbekannten[1] Iris-Datensatz mittels Zugriff auf die Variable `iris`. Wählen Sie eines der Attribute aus und erstellen Sie ein Histogramm sowie einen Boxplot. Versuchen Sie, beim Histogramm die Anzahl der Buckets zu verändern. Erstellen Sie außerdem einen Scatterplot, der ein Attribut auf der einen Achse hat und ein anderes Attribut auf der anderen Achse. Färben Sie die Datenpunkte dabei entsprechend der Spalte `Species` ein. *(Basis)*

f) **[Plots (2)]** Nutzen sie das Paket `ggplot2`, um die Plots aus der vorigen Aufgabe in etwas hübscherer Form zu erstellen. Versuchen Sie, den Titel des Plots, die Achsentitel und das Farbschema zu ändern. (Dies können Sie im Übrigen auch für die vorherigen Plots machen.) *(Vertiefung)*

3 Statistische Tests

Ziel dieser Aufgabe ist es, anhand des `iris`-Datensatzes verschiedene statistische Tests anzuwenden und zu interpretieren. Da R als Sprache für statistische Berechnungen konzipiert ist, können wir von einer Vielzahl eingebauter Test profitieren.

a) **[χ^2-Test (1)]** Nutzen Sie den χ^2-Test (Funktion: `chisq.test()`), um festzustellen, wie die numerischen Attribute mit dem Attribut `Species` zusammenhängen. Mittels der Funktion `cut()` können Sie dabei numerische Attribute diskretisieren. Was bedeuten die Ergebnisse? *(Basis)*

b) **[χ^2-Test (2)]** Erstellen Sie Boxplots der numerischen Attribute, gruppiert nach `Species`. Wie hängen die Plots mit den Ergebnissen der statistischen Tests zusammen? *(Vertiefung)*

c) **[Kolmogorov-Smirnov-Test]** Nutzen Sie den Kolmogorov-Smirnov-Test (Funktion: `ks.test()`), um festzustellen, ob das Attribut `Sepal.Width` für die beiden `Species` namens `versicolor` und `virginica` unterschiedlich verteilt ist. Wiederholen Sie die Analyse für das Attribut `Petal.Width` und interpretieren Sie die Ergebnisse. *(Vertiefung)*

d) **[Wilcoxon–Mann–Whitney-Test]** Nutzen Sie den Wilcoxon–Mann–Whitney-Test (Funktion: `wilcox.test()`), um festzustellen, ob das Attribut `Sepal.Width` für die beiden `Species` namens `versicolor` und `virginica` unterschiedlich verteilt ist. Wiederholen Sie die Analyse für das Attribut `Petal.Width` und interpretieren Sie die Ergebnisse. *(Vertiefung)*

[1] Zumindest so bekannt, dass er einen Wikipedia-Eintrag hat: https://en.wikipedia.org/wiki/Iris_flower_data_set