**Umgang mit Daten**

**A) Suchen und Sortieren**

**B) Graphische Darstellung von Daten**

**C) Exportieren und Importieren verschiedener Datenformate**

**D) Hausaufgaben**

**A) Suchen und Sortieren**

Daten nach bestimmten Kriterien zu durchsuchen ist eine sehr häufige Aufgabe bei der wissenschaftlichen Datenauswertung. Prinzipiell lässt sich das gut mit Hilfe von logischen Operatoren, Fallunterscheidungen und Schleifen erreichen. Es gibt aber auch den speziellen Befehl find, mit dem man Matrizen nach bestimmten Kriterien durchsuchen kann.

Es kommt häufig vor, dass man Daten nicht in der ursprünglichen Reihenfolge belassen möchte, sondern nach bestimmten Kriterien sortieren. Im Prinzip geht auch das "per Hand" - es kann aber schnell ziemlich viel Arbeit werden. Deshalb gibt es in Matlab den praktischen Befehl sortrows. m2=sortrows(m1,n) sortiert die Zeilen der Matrix m1 gemäß ihren Einträgen in der n-ten Spalte in aufsteigender Reihenfolge. [m2,index]=sortrows(m1,n) gibt zusätzlich einen Vektor der Indizes zurück.

T5A1) Es gibt in Matlab auch einen Befehl, der das Suchen übernimmt: find. Dieser Befehl gibt die Indizes derjenigen Elemente eines Vektors oder einer Matrix zurück, die ungleich 0 sind. Probieren Sie aus:

v=[1 7 5 0 8 0 3 1]; ind\_ungleich\_0=find(v); ind\_gleich\_1=find(v==1);

m=[ 0 1; 5 9; 1 0]; [row\_not0,col\_not0]=find(m); [row\_1,col\_1]=find(m==1);

T5A2) Laden Sie folgende Beobachtungsreihe: [fuetterungen.mat] oder [fuetterungen\_v6.mat]

In dieser Matrix wurden an 31 Tagen eines Monats eingetragen, wie groß die Niederschlagsmenge des Tages in mm war (1. Spalte) und wie oft ein Meisenweibchen sein Nest zum Füttern angeflogen hat (2. Spalte). Finden Sie heraus:

An welchen Tagen dieses Monats ist kein Regen gefallen?

An welchen Tagen hat die Meise häufiger als 50 Mal gefüttert?

An welchen Tagen ohne Niederschlag hat die Meise häufiger als 50 Mal gefüttert?

Wie häufig ist es vorgekommen, dass die Meise an zwei aufeinanderfolgenden Tagen zusammen über 100 Mal gefüttert hat?

\* T5A3) Beantworten Sie die Fragen aus T5A2 mit Hilfe von Schleifen, Fallunterscheidungen und / oder logischen Operatoren, ohne den Befehl find zu benutzen.

T5A4) Sortieren Sie die Matrix fuetterungen einmal nach der Anzahl Fütterungen und einmal nach der Niederschlagsmenge.

**B) Grafische Darstellung von Daten**

Daten graphisch darzustellen ist eine extrem wichtige Aufgabe bei der wissenschaftlichen Datenauswertung. Der wichtigste Befehl hierfür ist der bereits bekannte plot Befehl. In diesem Abschnitt wollen wir einige zusätzliche Möglichkeiten der grafischen Darstellung betrachten.

T5B1) Häufig verarbeitet man mehr Daten als man sinnvoll in einer Abbildung unterbringen kann. Man kann eine Abbildung in mehrere kleine Abbildungen teilen. Mit subplot(2,3,5) erzeugt man eine Matrix aus kleinen Abbildungen mit 2 Zeilen und 3 Spalten, wobei das 5. Fenster aktiv ist. In diesem kann der plot-Befehl normal verwendet werden. Machen Sie eine Abbildung bestehend aus 4 subplots, in denen für x=-5:0.1:5 die Kurven x.^1 bis x.^4 dargestellt sind.

T5B2) Matlab kann auch mehr als ein Grafikfenster gleichzeitig benutzen. Eine neue Abbildung bekommt man mit figure. Mit figure(1) und figure(2) springt man zwischen den aktiven Fenstern. close(1) schließt das zuerst erzeugte Fenster, close all schließt alle Grafikfenster. clf löscht den Inhalt des aktuellen Fensters.

Öffnen Sie zwei Grafikfenster plotten Sie im für den x-Vektor 0.1:0.1:10 im ersten Fenster die Quadratwurzel, im zweiten Fenster das Quadrat von x. Probieren Sie die oben genannten Befehle aus.

\*T5B3) Kuchengrafiken sind eine beliebte Darstellungsmöglichkeit für Prozentanteile. Um die Stimmenverteilung bei der letzten Wahl des Oldenburger Stadtrates darzustellen, würde man z.B. schreiben

prozente=[32.74 25.99 21.24 6.33 7.24 5.44 0.99];

pie(prozente)

Wenn man es noch etwas hübscher haben möchte, kann man z.B. auch benutzen:

pie3(prozente)

pie(prozente,{'SPD','CDU','Gruene','FDP','Linke','BFO','NPD'}), oder

pie3(prozente,[0 0 1 0 0 0 0]).

Was passiert dabei jeweils?

T5B4) Wir haben vor einiger Zeit schon mal den Befehl imagesc benutzt, um uns den Inhalt einer Matrix anzeigen zu lassen. Benutzen Sie ihn, um sich die Verteilung einzelliger Algen in einer Petrischale anzusehen, die als Matrix population in [algen.mat] bzw [algen\_v6.mat] gespeichert ist.

Was bedeuten in dieser Abbildung die beiden Achsen?

In den Vektoren x und y sind Abstände der Messbereiche in x und y Richtung der Petrischale gespeichert. Mit imagesc(x,y,population) bekommt man aussagekräftige Achsenbeschriftungen.

Der Befehl colorbar erzeugt eine Legende für die Farbkodierung der Werte.

T5B5) Dieselbe Algenverteilung lässt sich auch dreidimensional als "Gebirge" darstellen. Der Befehl dafür ist surf(population). Um aussagekräftige Achsen zu bekommen, muss jedem Punkt der Matrix ein x- und ein y-Wert zugeordnet werden (die Werte der Matrix population werden auf der z-Achse aufgetragen). Benutzen Sie dafür die mit abgespeicherten Matrizen xmatrix und ymatrix. Der Befehl colorbar funktioniert auch hier. Probieren Sie den alternativen Befehl mesh aus. Was ist der Unterschied zu surf?

**C) Exportieren und Importieren verschiedener Datenformate**

Wie wir bereits wissen speichert Matlab seine Daten standardmäßig als .mat Dateien ab, die ausschließlich von Matlab selber vernünftig weiterverarbeitet werden können. Häufig möchte man jedoch Matlab mit anderen Programmen kombinieren, die andere Dateiformate verwenden.

Wir werden uns zunächst etwas ausführlicher mit verschiedenen Grafik-Formaten beschäftigen, da diese extrem häufig gebraucht werden, anschließend kommen wir kurz zum Einlesen von Text-Dateien und zur Kompatibilität mit Excel. Wenn Sie Bedarf an ausführlicheren Informationen zum Im- oder Exportieren bestimmter Dateiformate haben schauen Sie bitte in der Hilfe nach.

T5C1) Eine wichtige Voraussetzung zum Verständnis von Grafik-Formaten ist die Farbkodierung mit der man die in einer Grafik genutzten Farben genau festlegen kann. Farben werden in der sogenannten RGB-Kodierung dargestellt, d.h. sie werden durch einen Vektor farbe=[rot gruen blau] definiert. In Matlab werden zur Kodierung für rot, grün und blau Werte zwischen 0 und 1 angegeben, z.B. ist [1 0 0] rot, [0 0 0] schwarz, [1 1 0] gelb, [0.6 0 0.6] dunkel-violett. So lassen sich beliebige Farben mischen.

\*) Wenn Sie Lust auf ausgefeilte Grafik haben: Stellen Sie den Vektor 0:1:10 mit einer durchgezogenen dunkelroten Linie und die einzelnen Punkte mit türkis gefüllten Kreisen dar (hierfür braucht man "line properties", siehe T1H5).

T5C2) Ein wichtiges Grafik-Format ist jpg. jpg-Dateien können in Matlab eingeladen und bearbeitet werden. Darin ist jeder Punkt durch einen Vektor von drei Farbwerten kodiert (allerdings etwas unterschiedlich zum in Matlab üblichen Format). Bei zweidimensionalen Abbildungen ergibt sich also eine 3 dimensionale Matrix der Größe MxNx3 (M und N ist die Anzahl Pixel in vertikaler und horizontaler Richtung). Schauen Sie sich die Arbeitsweise folgenden Skripts an:

[meisenshow.m] hierzu brauchen Sie ausserdem die Datei [bmeise.jpg]

Wie unterscheidet sich die Farbkodierung dieser jpg-Datei von der in Matlab üblichen?

Erweitern Sie die Show um eine Meise mit zwei Köpfen.

Teilen Sie die Matrix in drei Teile, um sich die Rot-, Grün- und Blau-Anteile des Bildes einzeln anzusehen.

T5C3) Man kann Matlab-Abbildungen abspeichern und erneut in Matlab öffnen. Die einfachste Möglichkeit dazu ist, den Menüpunkt ">File >Save as" zu benutzen. Der Standard ist dann, dass die Abbildung mit der Endung .fig abgespeichert wird - dabei handelt es sich um ein Format, was nur von Matlab vernünftig verarbeitet werden kann, dort aber beim erneuten Einladen alle Eigenschaften der Abbildung beibehält.

\*) Probieren Sie aus: Erzeugen Sie sich einen plot mit zwei dargestellten Graphen, z.B. zwei Glockenkurven wie am ersten Kurstag. Speichern Sie die Abbildung im Standardformat ab. Zoomen Sie in die Abbildung hinein und speichern Sie die Abbildung unter einem anderen Namen wieder ab. Schließen Sie dann das Grafikfenster und laden die beiden gerade gespeicherten Abbildungen über ">File >open" im Hauptmenü ein. Probieren Sie, bei der Abbildung des Ausschnitts wieder den gesamten Graphen zu reproduzieren.

T5C4) Wenn man die Abbildungen mit einem anderen Programm benutzen will, z.B. um sie in ein mit Word geschriebenes Protokoll einzubinden, muss man ein anderes Dateiformat benutzen als .fig Dazu kann man wie oben beschrieben abspeichern, muss jedoch das gewünschte Dateiformat auswählen und die Endung der Datei entsprechend anpassen.

\*) Probieren Sie aus, Ihre eben erzeugte Abbildung als jpg-Datei zu speichern, zu schließen und wieder mit Matlab zu öffnen. Was muss man tun, um sie sich wieder anzusehen? Kann man aus einem Ausschnitt der Kurve wieder zur gesamten Kurve zurückkommen?

\*) Probieren Sie aus, sowohl eine Abbildung im .fig Format als auch im .jpg Format in Word zu importieren.

\*T5C5) Da man Grafik meistens sowieso ansehen möchte, bevor man sie abspeichert, bietet sich meistens der Weg über die Menüleiste an. In manchen Fällen (z.B. wenn man für seine Bachelorarbeit viele gleich formatierte Abbildungen braucht) möchte man jedoch aus einem Skript oder einer Funktion heraus Abbildungen speichern. Hierfür gibt es den Befehl saveas. Damit dieser Befehl weiß, welche Grafik gespeichert werden soll, muss diese jedoch beim plotten benannt werden. Probieren Sie aus:

x=1:1:100; y=sqrt(x);

h=plot(x,y);

saveas(h,'testbild.fig')

Schließen Sie die Abbildung und laden Sie testbild.fig neu ein.

Das Praktische an dieser Art des Speicherns ist, dass Matlab direkt verschiedene Formate speichern kann und das gewünschte Dateiformat an der Endung des Dateinamens erkennt. Unter anderem können .jpg, .eps, .pdf und .bmp benutzt werden. (Die gesamte Liste möglicher Formate und ihrer Endungen findet man in der Hilfe unter saveas.)

\*) Probieren Sie, die Abbildung z.B. als bmp abzuspeichern und in Word zu importieren.

T5C6) Eine häufig gebrauchte Anwendung ist das Einlesen von Text-Dateien, die mit einem anderen Programm abgespeichert wurden.

a) Wenn es sich bei der Datei ausschließlich um Zahlen handelt und in jeder Zeile der Datei gleich viele Zahlen enthalten sind, kann man in Matlab den normalen load Befehl verwenden. Erstellen Sie mit einem Textprogramm eine Testdatei testdatei.txt z.B. mit den Einträgen

5 7 9 13.6

0.5 100 1 17

Speichern Sie sie als TEXT-Datei (NICHT word!) z.B. als testdatei.txt ab und laden Sie sie in Matlab mit A=load('testdatei.txt') ein.

\*b) Manchmal hat man Dateien, in denen die numerischen Einträge durch bestimmte Zeichen getrennt sind. z.B. könnten Sie eine Datei testdatei2.txt haben, die ein Semikolon zum Trennen benutzt:

5; 7; 9; 13.6

0.5; 100; 1; 17

Um diese einzulesen, muss Matlab das Trennzeichen mitgeteilt werden. Hierfür gibt es die Funktion dlmread (delimited characters read):

A= dlmread('testdatei2.txt',';')

Probieren Sie auch das aus.

\* c) Komplizierter ist die Sache, wenn eine Datei gemischte numerische und ASCII Daten enthält. Hierfür gibt es die Funktion textread. Sie basiert darauf, dass man durch eine Abfolge von %s (string) und %f (float Zahlen) das Muster angibt, mit dem jede der Zeilen aufgebaut ist. Schauen Sie sich bei Interesse in der Hilfe an, wie diese Funktion funktioniert und probieren Sie sie aus.

d) Umgekehrt kann man auch mit Matlab den Inhalt von Variablen als Textdateien abspeichern, die sich dann mit einem Texteditor ansehen und mit anderen Programmen weiterverarbeiten lassen. Die Syntax dafür lautet save dateiname variablenname -ascii. Erzeugen Sie eine Matrix im Workspace, speichern Sie diese als ascii-File ab und sehen Sie sich diese Datei in einem Texteditor an. Wie sieht eine dreidimensionale Matrix aus?

\* T5C7) Speziell für Nutzerinnen und Nutzer von Excel: Matlab kann excel-Dateien einlesen mit dem Befehl variable=xlsread('dateinme.xls'). Probieren Sie das mit der Datei [entfernungen.xls] aus. Der Befehl xlswrite('dateiname',variable) soll eine Datei im xls-Format speichern (funktioniert aber auf meinem Apple nicht), probieren Sie ihn auf den Windows-Rechnern aus, indem Sie ein grosse Matrix erzeugen, abspeichern und in Excel öffnen. Man kann auch mit copy-paste direkt zwischen dem Array Editor in Matlab und Excel hin und her kopieren, allerdings muss man in Excel zunächst den Bereich markieren, in den eine Matrix kopiert werden soll. Probieren Sie auch das aus.

\* T5C8) Matlab kann im Prinzip sehr viele Formate importieren - dies per Hand zu tun ist aber oft anstrengend. Deshalb gibt es den "Matlab import wizard", um dabei zu helfen. Schauen Sie bei Interesse in der Hilfe nach, wie man ihn verwendet. Für eine Auflistung verschiedener Importfunktionen für verschiedene Fileformate (inklusive Sound und Filmen) suchen Sie bitte in der Hilfe nach dem Schlagwort "File Formats".

**D) Hausaufgabe**

\*T5H1) Üben Sie den find Befehl, indem Sie Ihr Wunschkatzen-Programm (oder die Musterlösung) von Tag 3 (T3H4) so erweitern, dass es die Daten (Geschlecht, Alter, Farben) sämtlicher gefundener Wunschkatzen ausgibt. Sortieren Sie diese Angaben nach dem Alter.

\*T5H2) Benutzen Sie das am zur Aufgabe T4H7 entstandene Programm (wenn Sie die Aufgabe nicht selber gelöst haben, laden Sie sich eine Musterlösung herunter: vogeltabelle\_insa.m), um sich eine Matrix gefangener Vögel zu erstellen. Sortieren Sie diese so, dass, zuerst die Amseln, dann die Rotkehlchen und zuletzt die Meisen in der Matrix stehen, wobei innerhalb dieser Gruppen jeweils zuerst die Männchen und dann die Weibchen aufgelistet werden. Die Tiere einer Art und eines Geschlechts sollen jeweils nach dem Gewicht sortiert sein.