KAPITEL 1

Explorative Datenanalyse

Dieses Kapitel erläutert Ihnen den ersten Schritt in jedem datenwissenschaftlichen Projekt: die Datenexploration.

Die klassische Statistik konzentrierte sich fast ausschließlich auf die *Inferenz*, einen manchmal komplexen Satz von Verfahren, um aus kleinen Stichproben Rückschlüsse auf eine größere Grundgesamtheit zu ziehen. Im Jahr 1962 forderte John W. Tukey (*https://oreil.ly/LQw6q*) (siehe Abbildung 1-1) in seinem bahnbrechenden Aufsatz »The Future of Data Analysis« [Tukey-1962] eine Reform der Statistik. Er schlug eine neue wissenschaftliche Disziplin namens *Datenanalyse* vor, die die statistische Inferenz lediglich als eine Komponente enthielt. Tukey knüpfte Kontakte zu den Ingenieurs- und Informatikgemeinschaften (er prägte die Begriffe *Bit*, kurz für Binärziffer, und *Software*). Seine damaligen Ansätze haben bis heute überraschend Bestand und bilden einen Teil der Grundlagen der Data Science. Der Fachbereich der explorativen Datenanalyse wurde mit Tukeys im Jahr 1977 erschienenem und inzwischen als Klassiker geltendem Buch *Exploratory Data Analysis* [Tukey-1977] begründet. Tukey stellte darin einfache Diagramme (z.B. Box-Plots und Streudiagramme) vor, die in Kombination mit zusammenfassenden Statistiken (Mittelwert, Median, Quantile usw.) dabei helfen, ein Bild eines Datensatzes zu zeichnen.

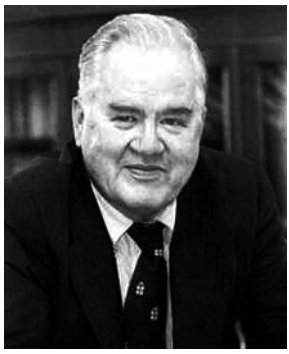

Abbildung 1-1: John Tukey, der bedeutende Statistiker, dessen vor über 50 Jahren entwickelte Ideen die Grundlage der Data Science bilden

Mit der zunehmenden Verfügbarkeit von Rechenleistung und leistungsfähigen Datenanalyseprogrammen hat sich die explorative Datenanalyse weit über ihren ursprünglichen Rahmen hinaus weiterentwickelt. Die wichtigsten Triebkräfte dieser Disziplin waren die rasche Entwicklung neuer Technologien, der Zugang zu mehr und umfangreicheren Daten und der verstärkte Einsatz der quantitativen Analyse in einer Vielzahl von Disziplinen. David Donoho, Professor für Statistik an der Stanford University und ehemaliger Student Tukeys, verfasste einen ausgezeichneten Artikel auf der Grundlage seiner Präsentation auf dem Workshop zur Hundertjahrfeier von Tukey in Princeton, New Jersey [Donoho-2015]. Donoho führt die Entwicklung der Data Science auf Tukeys Pionierarbeit in der Datenanalyse zurück.

Strukturierte Datentypen

Es gibt zahlreiche unterschiedliche Datenquellen: Sensormessungen, Ereignisse, Text, Bilder und Videos. Das *Internet der Dinge* (engl. *Internet of Things* (IoT)) produziert ständig neue Informationsfluten. Ein Großteil dieser Daten liegt unstrukturiert vor: Bilder sind nichts anderes als eine Zusammenstellung von Pixeln, wobei jedes Pixel RGB-Farbinformationen (Rot, Grün, Blau) enthält. Texte sind Folgen von Wörtern und Nicht-Wortzeichen, die oft in Abschnitte, Unterabschnitte usw. gegliedert sind. Clickstreams sind Handlungsverläufe eines Nutzers, der mit einer Anwendung oder einer Webseite interagiert. Tatsächlich besteht eine große Herausforderung der Datenwissenschaft darin, diese Flut von Rohdaten in verwertbare Informationen zu überführen. Um die in diesem Buch behandelten statistischen Konzepte in Anwendung zu bringen, müssen unstrukturierte Rohdaten zunächst aufbereitet und in eine strukturierte Form überführt werden. Eine der am häufigsten vorkommenden Formen strukturierter Daten ist eine Tabelle mit Zeilen und Spalten – so wie Daten aus einer relationalen Datenbank oder Daten, die für eine Studie erhoben wurden.

Es gibt zwei grundlegende Arten strukturierter Daten: numerische und kategoriale Daten. Numerische Daten treten in zwei Formen auf: *kontinuierlich*, wie z.B. die Windgeschwindigkeit oder die zeitliche Dauer, und *diskret*, wie z.B. die Häufigkeit des Auftretens eines Ereignisses. *Kategoriale* Daten nehmen nur einen bestimmten Satz von Werten an, wie z.B. einen TV-Bildschirmtyp (Plasma, LCD, LED usw.) oder den Namen eines Bundesstaats (Alabama, Alaska usw.). *Binäre* Daten sind ein wichtiger Spezialfall kategorialer Daten, die nur einen von zwei möglichen Werten annehmen, wie z.B. 0 oder 1, ja oder nein oder auch wahr oder falsch. Ein weiterer nützlicher kategorialer Datentyp sind *ordinalskalierte* Daten, bei denen die Kategorien in einer Reihenfolge geordnet sind; ein Beispiel hierfür ist eine numerische Bewertung (1, 2, 3, 4 oder 5).

Warum plagen wir uns mit der Taxonomie der Datentypen herum? Es stellt sich heraus, dass für die Zwecke der Datenanalyse und der prädiktiven Modellierung der Datentyp wichtig ist, um die Art der visuellen Darstellung, der Datenanalyse

oder des statistischen Modells zu bestimmen. Tatsächlich verwenden datenwissenschaftliche Softwareprogramme wie *R* und *Python* diese Datentypen, um die Rechenleistung zu optimieren. Noch wichtiger ist es, dass der Datentyp einer Variablen ausschlaggebend dafür ist, wie das Programm die Berechnungen für diese Variable handhabt.

Schlüsselbegriffe zu Datentypen

Numerisch
: Daten, die auf einer numerischen Skala abgebildet sind.

Kontinuierlich
: Daten, die innerhalb eines Intervalls einen beliebigen Wert annehmen können.

Synonyme
: intervallskaliert, Gleitkommazahl, numerisch

Diskret
: Daten, die nur ganzzahlige Werte annehmen können, wie z. B. Häufigkeiten bzw. Zählungen.

Synonyme
: Ganzzahl, Zählwert

Kategorial
: Daten, die nur einen bestimmten Satz von Werten annehmen können, die wiederum einen Satz von möglichen Kategorien repräsentieren.

Synonyme
: Aufzählungstyp, Faktor, faktoriell, nominal

Binär
: Ein Spezialfall des kategorialen Datentyps mit nur zwei möglichen Ausprägungen, z. B. 0/1, wahr/falsch.

Synonyme
: dichotom, logisch, Indikatorvariable, boolesche Variable

Ordinalskaliert
: Kategoriale Daten, die eine eindeutige Reihenfolge bzw. Rangordnung haben.

Synonym
: geordneter Faktor

Softwareingenieure und Datenbankprogrammierer fragen sich vielleicht, warum wir überhaupt den Begriff der *kategorialen* und *ordinalskalierten* Daten für unsere Analyse benötigen. Schließlich sind Kategorien lediglich eine Sammlung von Text- (oder numerischen) Werten, und die zugrunde liegende Datenbank übernimmt au-

tomatisch die interne Darstellung. Die explizite Bestimmung von Daten als kategoriale Daten im Vergleich zu Textdaten bietet jedoch einige Vorteile:

- Die Kenntnis, dass Daten kategorial sind, kann als Signal dienen, durch das ein Softwareprogramm erkennen kann, wie sich statistische Verfahren wie die Erstellung eines Diagramms oder die Anpassung eines Modells verhalten sollen. Insbesondere ordinalskalierte Daten können als `ordered.factor` in *R* angegeben werden, wodurch eine benutzerdefinierte Ordnung in Diagrammen, Tabellen und Modellen erhalten bleibt. In *Python* unterstützt `scikit-learn` ordinalskalierte Daten mit der Methode `sklearn.preprocessing.OrdinalEncoder`.
- Das Speichern und Indizieren kann optimiert werden (wie in einer relationalen Datenbank).
- Die möglichen Werte, die eine gegebene kategoriale Variable annehmen kann, werden in dem Softwareprogramm erzwungen (wie bei einer Aufzählung).

Der dritte »Vorteil« kann zu unbeabsichtigtem bzw. unerwartetem Verhalten führen: Das Standardverhalten von Datenimportfunktionen in *R* (z.B. `read.csv`) besteht darin, eine Textspalte automatisch in einen `factor` umzuwandeln. Bei nachfolgenden Operationen auf dieser Spalte wird davon ausgegangen, dass die einzigen zulässigen Werte für diese Spalte die ursprünglich importierten sind und die Zuweisung eines neuen Textwerts eine Warnung verursacht sowie einen Eintrag mit dem Wert `NA` (ein fehlender Wert) erzeugt. Das `pandas`-Paket in *Python* nimmt diese Umwandlung nicht automatisch vor. Sie können jedoch in der Funktion `read_csv` eine Spalte explizit als kategorial spezifizieren.

Kernideen

- Daten werden in Softwareprogrammen typischerweise in verschiedene Typen eingeteilt.
- Zu den Datentypen gehören numerische (kontinuierlich, diskret) und kategoriale (binär, ordinalskaliert).
- Die Datentypisierung dient als Signal für das Softwareprogramm, wie die Daten zu verarbeiten sind.

Weiterführende Literatur

- Datentypen können verwirrend sein, da sich Typen überschneiden und die Taxonomie in einem Softwareprogramm von der in einem anderen abweichen kann. Auf der R-Tutorial-Webseite (*https://oreil.ly/2YUoA*) können Sie die Taxonomie in *R* nachvollziehen. Die `pandas`-Dokumentation (*https://oreil.ly/UGX-4*) beschreibt die verschiedenen Datentypen in *Python* und wie sie verändert werden können.

- Datenbanken sind in ihrer Einteilung der Datentypen detaillierter und berücksichtigen Präzisionsniveaus, Datenfelder fester oder variabler Länge und mehr (siehe den W3Schools-SQL-Leitfaden (*https://oreil.ly/cThTM*).)

Tabellarische Daten

Der typische Bezugsrahmen für eine Analyse in der Data Science ist ein *tabellarisches Datenobjekt* (engl. *Rectangular Data Object*), wie eine Tabellenkalkulation oder eine Datenbanktabelle.

»Tabellarische Daten« ist der allgemeine Begriff für eine zweidimensionale Matrix mit Zeilen für die Beobachtungen (Fälle) und Spalten für die Merkmale (Variablen); in *R* und *Python* wird dies als *Data Frame* bezeichnet. Die Daten sind zu Beginn nicht immer in dieser Form vorhanden: Unstrukturierte Daten (z. B. Text) müssen zunächst so verarbeitet und aufbereitet werden, dass sie als eine Reihe von Merkmalen in tabellarischer Struktur dargestellt werden können (siehe »Strukturierte Datentypen« auf Seite 2). Daten in relationalen Datenbanken müssen für die meisten Datenanalyse- und Modellierungsaufgaben extrahiert und in eine einzelne Tabelle überführt werden.

Schlüsselbegriffe zu tabellarischen Daten

Data Frame
: Tabellarische Daten (wie ein Tabellenkalkulationsblatt) sind die grundlegende Datenstruktur für statistische und maschinelle Lernmodelle.

Merkmal
: Eine Spalte innerhalb einer Tabelle wird allgemein als *Merkmal* (engl. *Feature*) bezeichnet.

Synonyme
: Attribut, Eingabe, Prädiktorvariable, Prädiktor, unabhängige Variable

Ergebnis
: Viele datenwissenschaftliche Projekte zielen auf die Vorhersage eines *Ergebnisses* (engl. *Outcome*) ab – oft in Form eines Ja-oder-Nein-Ergebnisses (ob beispielsweise in Tabelle 1-1 eine »Auktion umkämpft war oder nicht«). Die *Merkmale* werden manchmal verwendet, um das *Ergebnis* eines statistischen Versuchs oder einer Studie vorherzusagen..

Synonyme
: Ergebnisvariable, abhängige Variable, Antwortvariable, Zielgröße, Ausgabe, Responsevariable

Eintrag
: Eine Zeile innerhalb einer Tabelle wird allgemein als *Eintrag* (engl. *Record*) bezeichnet.

Synonyme
: Fall, Beispiel, Instanz, Beobachtung

Tabelle 1-1: Ein typisches Data-Frame-Format

Kategorie	Währung	Verkäufer-Rating	Dauer	Schluss-tag	Schluss-preis	Eröffnungs-preis	umkämpft?
Musik/Film/Spiel	USD	3249	5	Mon	0.01	0.01	0
Musik/Film/Spiel	USD	3249	5	Mon	0.01	0.01	0
Automobil	USD	3115	7	Die	0.01	0.01	0
Automobil	USD	3115	7	Die	0.01	0.01	0
Automobil	USD	3115	7	Die	0.01	0.01	0
Automobil	USD	3115	7	Die	0.01	0.01	0
Automobil	USD	3115	7	Die	0.01	0.01	1
Automobil	USD	3115	7	Die	0.01	0.01	1

In Tabelle 1-1 gibt es eine Kombination aus Mess- oder Zähldaten (z. B. Dauer und Preis) und kategorialen Daten (z. B. Kategorie und Währung). Wie bereits erwähnt, ist eine besondere Form der kategorialen Variablen eine binäre Variable (ja/nein oder 0/1), wie in der Spalte ganz rechts in Tabelle 1-1 – eine Indikatorvariable, die angibt, ob eine Auktion umkämpft war (mehrere Bieter hatte) oder nicht. Diese Indikatorvariable ist zufällig auch eine *Ergebnisvariable*, wenn das Modell vorhersagen soll, ob eine Auktion umkämpft sein wird oder nicht.

Data Frames und Tabellen

Klassische Datenbanktabellen haben eine oder mehrere Spalten, die als Index bezeichnet werden und im Wesentlichen eine Zeilennummer darstellen. Dies kann die Effizienz bestimmter Datenbankabfragen erheblich verbessern. In *Pythons* `pandas`-Bibliothek wird die grundlegende tabellarische Datenstruktur durch ein Data-Frame-Objekt umgesetzt. Standardmäßig wird automatisch ein ganzzahliger Index für ein Data-Frame-Objekt basierend auf der Reihenfolge der Zeilen erstellt. In `pandas` ist es auch möglich, mehrstufige bzw. hierarchische Indizes festzulegen, um die Effizienz bestimmter Operationen zu verbessern.

In *R* ist die grundlegende tabellarische Datenstruktur mittels eines `data.frame`-Objekts implementiert. Ein `data.frame` hat auch einen impliziten ganzzahligen Index, der auf der Zeilenreihenfolge basiert. Der standardmäßige `data.frame` in *R* unterstützt keine benutzerdefinierten oder mehrstufigen Indizes. Jedoch kann über das Argument `row.names` ein benutzerdefinierter Schlüssel erstellt werden. Um diesem Problem zu begegnen, werden immer häufiger zwei neuere Pakete eingesetzt: `data.table` und `dplyr`. Beide unterstützen mehrstufige Indizes und bieten erhebliche Beschleunigungen bei der Arbeit mit einem `data.frame`.

Unterschiede in der Terminologie

Die Terminologie bei tabellarischen Daten kann verwirrend sein. Statistiker und Data Scientists verwenden oftmals unterschiedliche Begriffe für ein und denselben Sachverhalt. Statistiker nutzen in einem Modell *Prädiktorvariablen*, um eine *Antwortvariable* (engl. *Response*) oder eine *abhängige Variable* vorherzusagen. Ein Datenwissenschaftler spricht von *Merkmalen* (engl. *Features*), um eine *Zielgröße* (engl. *Target*) vorherzusagen. Ein Synonym ist besonders verwirrend: Informatiker verwenden den Begriff *Stichprobe* (engl. *Sample*) für eine einzelne Datenzeile, für einen Statistiker ist eine *Stichprobe* hingegen eine Sammlung von Datenzeilen.

Nicht tabellarische Datenstrukturen

Neben tabellarischen Daten gibt es noch andere Datenstrukturen.

Zeitreihendaten umfassen aufeinanderfolgende Messungen derselben Variablen. Sie sind das Rohmaterial für statistische Prognosemethoden und auch eine zentrale Komponente der von Geräten – dem Internet der Dinge – erzeugten Daten.

Räumliche Daten- bzw. Geodatenstrukturen, die bei der Kartierung und Standortanalyse verwendet werden, sind komplexer und vielfältiger als tabellarische Datenstrukturen. In der *Objektdarstellung* (engl. *Object Representation*) stehen ein Objekt (z.B. ein Haus) und seine räumlichen Koordinaten im Mittelpunkt der Daten. Die *Feldansicht* (engl. *Field View*) hingegen konzentriert sich auf kleine räumliche Einheiten und den Wert einer relevanten Metrik (z.B. Pixelhelligkeit).

Graphen- (oder Netzwerk-)Datenstrukturen werden verwendet, um physikalische, soziale oder abstrakte Beziehungen darzustellen. Beispielsweise kann ein Diagramm eines sozialen Netzwerks wie Facebook oder LinkedIn Verbindungen zwischen Menschen im Netzwerk darstellen. Ein Beispiel für ein physisches Netzwerk sind Vertriebszentren, die durch Straßen miteinander verbunden sind. Diagrammstrukturen sind für bestimmte Arten von Fragestellungen nützlich, wie z.B. bei der Netzwerkoptimierung und bei Empfehlungssystemen.

Jeder dieser Datentypen hat seine eigene spezifische Methodologie in der Data Science. Der Schwerpunkt dieses Buchs liegt auf tabellarische Daten, dem grundlegenden Baustein der prädiktiven Modellierung.

Graphen in der Statistik

In der Informatik und der Informationstechnologie bezieht sich der Begriff *Graph* typischerweise auf die Darstellung von Verbindungen zwischen Entitäten und auf die zugrunde liegende Datenstruktur. In der Statistik wird der Begriff *Graph* verwendet, um sich auf eine Vielzahl von Darstellungen und Visualisierungen zu beziehen, nicht nur von Verbindungen zwischen Entitäten. Zudem bezieht er sich ausschließlich auf die Visualisierung und nicht auf die Datenstruktur.

Kernideen

- Die grundlegende Datenstruktur in der Data Science ist eine rechteckige Matrix, in der die Zeilen den Beobachtungen entsprechen und die Spalten den Variablen (Merkmalen).
- Die Terminologie kann verwirrend sein; es gibt eine Vielzahl von Synonymen, die sich aus den verschiedenen Disziplinen ergeben, die zur Data Science beitragen (Statistik, Informatik und Informationstechnologie).

Weiterführende Literatur

- Dokumentation zu Data Frames in *R* (*https://oreil.ly/NsONR*)
- Dokumentation zu Data Frames in *Python* (*https://oreil.ly/oxDKQ*)

Lagemaße

Variablen für Mess- oder Zähldaten können Tausende von unterschiedlichen Werten haben. Ein grundlegender Schritt bei der Erkundung Ihrer Daten ist die Ermittlung eines »typischen Werts« für jedes Merkmal (Variable) – ein sogenanntes Lagemaß (engl. *Estimates of Location*): eine Schätzung darüber, wo sich die Mehrheit der Daten konzentriert (d.h. ihre zentrale Tendenz).

Schlüsselbegriffe zu Lagemaßen

Mittelwert
: Die Summe aller Werte dividiert durch die Anzahl der Werte.

Synonyme
: arithmetisches Mittel, Durchschnitt

Gewichteter Mittelwert
: Die Summe aller Werte, die jeweils mit einem Gewicht bzw. einem Gewichtungsfaktor multipliziert werden, geteilt durch die Summe aller Gewichte.

Synonym
: gewichteter Durchschnitt

Median
: Der Wert, bei dem die Hälfte der Daten oberhalb und die andere Hälfte unterhalb dieses Werts liegt.

Synonym
: 50%-Perzentil

Perzentil
: Der Wert, bei dem *P* % der Daten unterhalb dieses Werts liegen.

Synonym
: Quantil

Gewichteter Median
: Der Wert, bei dem die Summe der Gewichte der sortierten Daten exakt die Hälfte beträgt und der die Daten so einteilt, dass sie entweder oberhalb oder unterhalb diesen Werts liegen.

Getrimmter Mittelwert
: Der Mittelwert aller Werte, nachdem eine vorgegebene Anzahl von Ausreißern entfernt wurde.

Synonym
: gestutzter Mittelwert

Robust
: Nicht sensibel gegenüber Ausreißern.

Ausreißer
: Ein Datenwert, der sich stark von den übrigen Daten unterscheidet.

Synonym
: Extremwert

Auf den ersten Blick mag für Sie die Ermittlung einer zusammenfassenden Größe, die Aufschluss über einen vorliegenden Datensatz gibt, ziemlich trivial erscheinen: Sie nehmen einfach den *Mittelwert*, der sich für den Datensatz ergibt. Tatsächlich ist der Mittelwert zwar leicht zu berechnen und relativ zweckmäßig, aber er ist nicht immer das beste Maß zur Bestimmung eines Zentralwerts. Aus diesem Grund haben Statistiker mehrere alternative Schätzer zum Mittelwert entwickelt und befürwortet.

Metriken und Schätzwerte

Statistiker verwenden oft den Begriff *Schätzwert* für einen aus den vorliegenden Daten berechneten Wert, um zwischen dem, was wir aus den Daten ziehen, und der theoretisch wahren oder tatsächlichen Sachlage zu unterscheiden. Data Scientists und Geschäftsanalysten sprechen bei einem solchen Wert von einer *Metrik*. Der Unterschied spiegelt den Ansatz der Statistik im Vergleich zur Datenwissenschaft wider: Die Berücksichtigung von Unsicherheit steht im Mittelpunkt der statistischen Disziplin, währnd in der Datenwissenschaft konkrete geschäftliche oder organisatorische Ziele im Fokus stehen. Daher kann man sagen, dass Statistiker Schätzungen durchführen und Data Scientists Messungen vornehmen.

Mittelwert

Das grundlegendste Lagemaß ist der *Mittelwert* (genauer, das arithmetische Mittel) oder auch der *Durchschnitt*. Der Mittelwert entspricht der Summe aller Werte dividiert durch die Anzahl von Werten. Betrachten Sie die folgende Zahlenfolge: {3 5 1 2}. Der Mittelwert beträgt (3 + 5 + 1 + 2) / 4 = 11 / 4 = 2,75. Sie werden auf das Symbol $\bar{x}$ (ausgesprochen als »x quer«) stoßen, das verwendet wird, um den Mittel-

wert einer Stichprobe, die aus einer Grundgesamtheit gezogen wurde, darzustellen. Die Formel zur Berechnung des Mittelwerts für eine Menge von Werten x_1, x_2, ..., x_n lautet:

$$\text{Mittelwert} = \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

N (oder *n*) bezieht sich auf die Gesamtzahl aller Einträge bzw. Beobachtungen. In der Statistik wird es großgeschrieben, wenn es sich auf eine Grundgesamtheit bezieht, und kleingeschrieben, wenn es auf eine Stichprobe aus einer Grundgesamtheit abzielt. In der Data Science ist diese Unterscheidung nicht von Relevanz, weshalb Sie beide Möglichkeiten in Betracht ziehen können.

Eine Variante des Mittelwerts ist der *getrimmte Mittelwert*, den Sie berechnen, indem Sie eine feste Anzahl sortierter Werte an jedem Ende weglassen und dann den Mittelwert der verbleibenden Werte bilden. Für die sortierten Werte $x_{(1)}$, $x_{(2)}$, ..., $x_{(n)}$, wobei $x_{(1)}$ der kleinste Wert und $x_{(n)}$ der größte ist, wird der getrimmte Mittelwert mit p kleinsten und größten weggelassenen Werten durch folgende Formel berechnet:

$$\text{getrimmter Mittelwert} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n - 2p}$$

Durch die Verwendung des getrimmten Mittelwerts wird der Einfluss von Extremwerten beseitigt. Zum Beispiel werden bei internationalen Tauchmeisterschaften die höchste und die niedrigste Punktzahl der fünf Kampfrichter gestrichen, und als Endpunktzahl wird der Durchschnitt der Punktzahlen der drei verbleibenden Kampfrichter gewertet (*https://oreil.ly/uV4P0*). Dies macht es für einen einzelnen Kampfrichter schwierig, das Ergebnis zu manipulieren, etwa um den Kandidaten seines Landes zu begünstigen. Getrimmte Mittelwerte sind sehr verbreitet und in vielen Fällen der Verwendung des gewöhnlichen Mittelwerts vorzuziehen (siehe »Median und andere robuste Lagemaße« auf Seite 11 für weitere Erläuterungen).

Eine weitere Möglichkeit der Mittelwertbildung ist der *gewichtete Mittelwert*. Zur Berechnung multiplizieren Sie jeden Datenwert x_i mit einem benutzerdefinierten Gewicht w_i und dividieren die daraus resultierende Summe durch die Summe der Gewichte. Die Formel für den gewichteten Mittelwert lautet dementsprechend:

$$\text{gewichteter Mittelwert} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Den gewichteten Mittelwert verwendet man hauptsächlich aus zwei Gründen:

- Einige Werte weisen von sich aus eine größere Streuung auf als andere – um den Einfluss stark streuender Beobachtungen zu verringern, erhalten sie ein

geringeres Gewicht. Wenn wir z. B. den Mittelwert von mehreren Sensoren bilden und einer der Sensoren weniger genau misst, können wir die Daten dieses Sensors niedriger gewichten.

- Unsere erhobenen Daten repräsentieren die verschiedenen Gruppen, an deren Messung wir interessiert sind, nicht gleichmäßig. Beispielsweise ist es möglich, aufgrund der Art und Weise, wie ein Onlineversuch durchgeführt wurde, einen Datensatz zu gewinnen, der nicht alle Gruppen in der Nutzerbasis wahrheitsgemäß abbildet. Zur Korrektur können wir den Werten der Gruppen, die unterrepräsentiert sind, ein höheres Gewicht beimessen.

Median und andere robuste Lagemaße

Der *Median* entspricht dem mittleren Wert der sortierten Liste eines Datensatzes. Wenn es eine gerade Anzahl von Datenpunkten gibt, ist der mittlere Wert eigentlich nicht im Datensatz enthalten, weshalb der Durchschnitt der beiden Werte, die die sortierten Daten in eine obere und eine untere Hälfte teilen, verwendet wird. Verglichen mit dem Mittelwert, bei dem alle Beobachtungen berücksichtigt werden, beruht der Median nur auf den Werten, die sich in der Mitte des sortierten Datensatzes befinden. Dies mag zwar nachteilig erscheinen, da der Mittelwert wesentlich empfindlicher in Bezug auf die Datenwerte ist, aber es gibt viele Fälle, in denen der Median ein besseres Lagemaß darstellt. Angenommen, wir möchten die durchschnittlichen Haushaltseinkommen in den Nachbarschaften um den Lake Washington in Seattle unter die Lupe nehmen. Beim Vergleich der Ortschaft Medina mit der Ortschaft Windermere würde die Verwendung des Mittelwerts zu sehr unterschiedlichen Ergebnissen führen, da Bill Gates in Medina lebt. Wenn wir stattdessen den Median verwenden, spielt es keine Rolle, wie reich Bill Gates ist – die Position der mittleren Beobachtung bleibt unverändert.

Aus den gleichen Gründen wie bei der Verwendung eines gewichteten Mittelwerts ist es auch möglich, einen *gewichteten Median* zu ermitteln. Wie beim Median sortieren wir zunächst die Daten, obwohl jeder Datenwert ein zugehöriges Gewicht hat. Statt der mittleren Zahl ist der gewichtete Median ein Wert, bei dem die Summe der Gewichte für die untere und die obere »Hälfte« der sortierten Liste gleich ist. Wie der Median ist auch der gewichtete Median robust gegenüber Ausreißern.

Ausreißer

Der Median wird als *robustes* Lagemaß angesehen, da er nicht von *Ausreißern* (Extremfällen) beeinflusst wird, die die Ergebnisse verzerren könnten. Ausreißer sind Werte, die sehr stark von allen anderen Werten in einem Datensatz abweichen. Die genaue Definition eines Ausreißers ist etwas subjektiv, obwohl bestimmte Konventionen in verschiedenen zusammenfassenden Statistiken und Diagrammen verwendet werden (siehe »Perzentile und Box-Plots« auf Seite 21). Nur weil ein Datenwert einen Ausreißer darstellt, macht es ihn nicht ungültig oder fehlerhaft (wie im

vorherigen Beispiel mit Bill Gates). Dennoch sind Ausreißer oft das Ergebnis von Datenfehlern, wie z. B. von Daten, bei denen verschiedene Einheiten vermischt wurden (Kilometer gegenüber Metern), oder fehlerhafte Messwerte eines Sensors. Wenn Ausreißer das Ergebnis fehlerhafter bzw. ungültiger Daten sind, wird der Mittelwert zu einer falschen Einschätzung der Lage führen, wohingegen der Median immer noch seine Gültigkeit behält. Ausreißer sollten in jedem Fall identifiziert werden und sind in der Regel eine eingehendere Untersuchung wert.

Anomalieerkennung

Im Gegensatz zur gewöhnlichen Datenanalyse, bei der Ausreißer manchmal informativ sind und manchmal stören, sind bei der *Anomalieerkennung* die Ausreißer von Interesse, und der größere Teil der Daten dient in erster Linie dazu, den »Normalzustand« zu definieren, an dem die Anomalien gemessen werden.

Der Median ist nicht das einzige robuste Lagemaß. Tatsächlich wird häufig der getrimmte Mittelwert verwendet, um den Einfluss von Ausreißern zu vermeiden. So bietet z. B. die Entfernung der unteren und oberen 10% der Daten (eine übliche Wahl) Schutz vor Ausreißern, es sei denn, der Datensatz ist zu klein. Der getrimmte Mittelwert kann als Kompromiss zwischen dem Median und dem Mittelwert gesehen werden: Er ist robust gegenüber Extremwerten in den Daten, verwendet jedoch mehr Daten zur Berechnung des Lagemaßes.

Weitere robuste Lagemaße

Statistiker haben eine Vielzahl anderer Lagemaße entwickelt, und zwar in erster Linie mit dem Ziel, einen Schätzer zu entwickeln, der robuster und auch effizienter als der Mittelwert ist (d. h. besser in der Lage, kleine Unterschiede hinsichtlich der Lage zwischen Datensätzen zu erkennen). Während diese Methoden für kleine Datensätze durchaus nützlich sein können, dürften sie bei großen oder selbst bei mittelgroßen Datensätzen keinen zusätzlichen Nutzen bringen.

Beispiel: Lagemaße für Einwohnerzahlen und Mordraten

Tabelle 1-2 zeigt einen Auszug der ersten paar Zeilen eines Datensatzes, der Informationen zu den Einwohnerzahlen und Mordraten für jeden US-Bundesstaat enthält (Zensus 2010). Die Einheit für die Mordrate wurde mit »Morde pro 100.000 Personen pro Jahr« gewählt.

Tabelle 1-2: Die ersten Zeilen des data.frame, der Auskunft über die Einwohnerzahlen und Mordraten der einzelnen Bundesstaaten gibt

	Bundesstaat	Einwohnerzahl	Mordrate	Abkürzung
1	Alabama	4.779.736	5,7	AL
2	Alaska	710.231	5,6	AK
3	Arizona	6.392.017	4,7	AZ

Tabelle 1-2: Die ersten Zeilen des data.frame, der Auskunft über die Einwohnerzahlen und Mordraten der einzelnen Bundesstaaten gibt (Fortsetzung)

	Bundesstaat	Einwohnerzahl	Mordrate	Abkürzung
4	Arkansas	2.915.918	5,6	AR
5	California	37.253.956	4,4	CA
6	Colorado	5.029.196	2,8	CO
7	Connecticut	3.574.097	2,4	CT
8	Delaware	897.934	5,8	DE

Berechnen Sie den Mittelwert, den getrimmten Mittelwert und den Median für die Einwohnerzahlen in *R*:[1]

```
> state <- read.csv('state.csv')
> mean(state[['Population']])
[1] 6162876
> mean(state[['Population']], trim=0.1)
[1] 4783697
> median(state[['Population']])
[1] 4436370
```

In *Python* können wir zur Berechnung des Mittelwerts und des Medians die pandas-Methoden des Data Frame verwenden. Den getrimmten Mittelwert erhalten wir durch die Funktion `trim_mean` aus dem Modul `scipy.stats`:

```
state = pd.read_csv('state.csv')
state['Population'].mean()
trim_mean(state['Population'], 0.1)
state['Population'].median()
```

Der Mittelwert ist größer als der getrimmte Mittelwert, der wiederum größer als der Median ist.

Dies liegt daran, dass der getrimmte Mittelwert die fünf größten und fünf kleinsten Bundesstaaten ausschließt (`trim=0.1` entfernt 10% an beiden Enden der Verteilung). Wenn wir die durchschnittliche Mordrate für das Land berechnen wollen, müssen wir dazu den gewichteten Mittelwert oder den Median heranziehen, um die unterschiedlich hohe Anzahl an Einwohnern in den Bundesstaaten zu berücksichtigen. Da *R* in seiner Standardbibliothek keine Funktion für den gewichteten Median umfasst, müssen wir zu diesem Zweck zunächst das Paket `matrixStats` installieren:

```
> weighted.mean(state[['Murder.Rate']], w=state[['Population']])
[1] 4.445834
> library('matrixStats')
> weightedMedian(state[['Murder.Rate']], w=state[['Population']])
[1] 4.4
```

1 Der *R*- und der *Python*-Code sind auf das Wesentliche reduziert. Den vollständigen Code sowie die Datensätze zum Herunterladen finden Sie unter *https://github.com/gedeck/practical-statistics-for-data-scientists*.

Bei *Python* ist die Funktion zur Berechnung des gewichteten Mittelwerts im `NumPy`-Paket enthalten. Für den gewichteten Median können wir speziell das Paket wquantiles (*https://oreil.ly/4SIPQ*) verwenden:

```
np.average(state['Murder.Rate'], weights=state['Population'])
wquantiles.median(state['Murder.Rate'], weights=state['Population'])
```

Im vorliegenden Fall sind der gewichtete Mittelwert und der gewichtete Median in etwa gleich groß.

Kernideen

- Das wesentliche Lagemaß ist der Mittelwert, der jedoch empfindlich auf Extremwerte (Ausreißer) reagiert.
- Andere Maße (Median, getrimmter Mittelwert) sind weniger empfindlich gegenüber Ausreißern und ungewöhnlich verteilten Daten und daher robuster.

Weiterführende Literatur

- In dem Wikipedia-Artikel zur zentralen Tendenz (*https://oreil.ly/qUW2i*) werden verschiedene Lagemaße ausführlich erläutert.
- John Tukeys Standardwerk aus dem Jahr 1977, *Exploratory Data Analysis* (Pearson), erweist sich nach wie vor als eine beliebte Lektüre.

Streuungsmaße

Die Lage ist nur eine Dimension bei der Zusammenfassung eines Merkmals. Eine zweite Dimension, die *Streuung* (engl. *Variability*) – auch *Variabilität* oder *Dispersion* genannt –, misst, ob die Datenwerte eng zusammenliegen oder weit gestreut sind. Die Streuung ist das Herzstück der Statistik: Sie wird gemessen, reduziert, es kann unterschieden werden zwischen zufälliger und tatsächlicher Streuung, die verschiedenen Quellen der wahren Streuung können identifiziert und Entscheidungen in Gegenwart der Streuung können getroffen werden.

Schlüsselbegriffe zu Streuungsmaßen

Abweichung
: Die Differenz zwischen den beobachteten Werten und dem Lagemaß (engl. *Deviation*).

Synonyme
: Fehler, Residuen

Varianz
: Die Summe der quadrierten Abweichungen vom Mittelwert dividiert durch $n - 1$, wobei n die Anzahl der Beobachtungen ist.

  Synonym
  : mittlerer quadratischer Fehler

Standardabweichung
: Die Quadratwurzel der Varianz.

Mittlere absolute Abweichung
: Der Mittelwert der Absolutwerte der Abweichungen vom Mittelwert.

  Synonyme
  : l1-Norm, Manhattan-Norm

Mittlere absolute Abweichung vom Median
: Der Median der Absolutwerte der Abweichungen vom Median.

Spannweite
: Die Differenz zwischen dem größten und dem kleinsten Wert in einem Datensatz (engl. *Range*).

Ordnungsstatistik
: Eine auf den Datenwerten basierende Metrik, sortiert vom kleinsten zum größten.

  Synonym
  : Rang

Perzentil
: Der Wert, bei dem P % der Werte diesen Wert oder weniger und (100-P) % diesen Wert oder mehr annehmen.

  Synonym
  : Quantil

Interquartilsabstand
: Die Differenz zwischen dem 75%-Perzentil und dem 25%-Perzentil.

  Synonym
  : IQR

So wie es verschiedene Möglichkeiten gibt, die Lage zu messen (Mittelwert, Median usw.), so gibt es auch verschiedene Möglichkeiten, das Ausmaß der Streuung zu bestimmen.

Standardabweichung und ähnliche Maße

Die meistgenutzten Streuungsmaße basieren auf den Differenzen bzw. *Abweichungen* zwischen den Lagemaßen und den beobachteten Daten. Für eine gegebene Zahlenfolge {1, 4, 4} ist der Mittelwert 3 und der Median 4. Die Abweichungen vom Mittelwert entsprechen den jeweiligen Differenzen: $1 - 3 = -2$, $4 - 3 = 1$, $4 - 3 = 1$.

Diese Abweichungen geben uns Aufschluss darüber, wie weit die Daten um den Zentralwert herum gestreut sind.

Eine Möglichkeit, die Streuung zu messen, besteht darin, einen typischen Wert für diese Abweichungen zu schätzen. Eine Mittelwertbildung über die Abweichungen selbst würde uns nicht viel sagen – die negativen Abweichungen würden die positiven ausgleichen. Tatsächlich ist auch im vorliegenden Beispiel die Summe der Abweichungen vom Mittelwert genau null. Stattdessen können wir den Mittelwert der Absolutwerte der Abweichungen vom Mittelwert bilden. Im vorhergehenden Beispiel sind die Absolutwerte der Abweichungen {2 1 1}, und ihr Mittelwert ergibt (2 + 1 + 1) / 3 = 1,33. Dieses Maß wird als *mittlere absolute Abweichung* bezeichnet und mit der folgenden Formel ermittelt:

$$\text{Mittlere absolute Abweichung} = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

wobei $\bar{x}$ für den Stichprobenmittelwert steht.

Die bekanntesten Streuungsmaße sind die *Varianz* und die *Standardabweichung*, die auf den quadratischen Abweichungen beruhen. Die Varianz ist der Durchschnitt der quadrierten Abweichungen, und die Standardabweichung ist wiederum die Quadratwurzel der Varianz:

$$\text{Varianz} = s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$

$$\text{Standardabweichung} = s = \sqrt{\text{Varianz}}$$

Die Standardabweichung ist viel leichter zu interpretieren als die Varianz, da sie auf dieselbe Skala wie die Originaldaten bezogen ist. Dennoch mag es mit ihrer komplizierteren und weniger intuitiven Formel merkwürdig erscheinen, dass die Standardabweichung in der Statistik gegenüber der mittleren absoluten Abweichung bevorzugt wird. Sie verdankt ihre Vorrangstellung der statistischen Theorie: Mathematisch gesehen, ist es sehr viel vorteilhafter, quadrierte Werte zu verwenden – und nicht Absolutwerte –, insbesondere in statistischen Modellen.

Die Anzahl der Freiheitsgrade und die Frage, ob n oder n – 1?

In Statistikbüchern finden Sie für gewöhnlich einen Abschnitt, der erklärt, warum wir $n - 1$ im Nenner der Formel für die Varianz anstelle von n haben, was uns zum Konzept der *Freiheitsgrade* führt. Diese Unterscheidung ist an sich nicht von großer Bedeutung, da n im Allgemeinen so groß ist, dass es keinen besonderen Unterschied macht, ob man durch n oder $n - 1$ dividiert. Aber falls es Sie interessiert, hier folgt die Erklärung. Sie basiert auf der Prämisse, dass Sie auf Basis einer Stichprobe Schätzungen über eine Grundgesamtheit (Population) vornehmen möchten.

Wenn Sie intuitiverweise n im Nenner der Varianzformel verwenden, unterschätzen Sie den wahren Wert der Varianz und der Standardabweichung in der Grundgesamtheit. Dies wird als ein *verzerrter* Schätzer (engl. *biased*) bezeichnet. Wenn Sie jedoch $n - 1$ anstelle von n einsetzen, ermitteln Sie einen *unverzerrten* (engl. *unbiased*) bzw. erwartungstreuen Schätzer der Varianz.

Um vollständig zu erklären, warum die Verwendung von n zu einem verzerrten Schätzer führt, müssen wir den Begriff der Freiheitsgrade heranziehen, der die Anzahl der Einschränkungen bei der Berechnung eines Schätzers berücksichtigt. In diesem Fall gibt es $n - 1$ Freiheitsgrade, da es eine Randbedingung gibt: Die Standardabweichung hängt von der Berechnung des Stichprobenmittelwerts ab. In den meisten Anwendungsfällen müssen sich Data Scientists keine Gedanken über die Anzahl der Freiheitsgrade machen.

Weder die Varianz noch die Standardabweichung oder die mittlere absolute Abweichung ist gegenüber Ausreißern und Extremwerten robust (siehe »Median und andere robuste Lagemaße« auf Seite 11 für eine Erläuterung zu den robusten Lagemaßen). Die Varianz und die Standardabweichung sind besonders empfindlich gegenüber Ausreißern, da sie auf den quadrierten Abweichungen beruhen.

Ein robustes Streuungsmaß ist die *mittlere absolute Abweichung vom Median* (engl. *Median Absolute Deviation from the Median*, MAD):

$$\text{Mittlere absolute Abweichung vom} = \text{Median}\left(\left|x_1 - m\right|, \left|x_2 - m\right|, ..., \left|x_N - m\right|\right)$$

wobei m dem Median entspricht. Wie der Median wird auch die mittlere absolute Abweichung vom Median nicht durch Extremwerte beeinflusst. Es ist auch möglich, eine getrimmte Standardabweichung analog zum getrimmten Mittelwert zu berechnen (siehe »Mittelwert« auf Seite 9).

Die Varianz, die Standardabweichung, die mittlere absolute Abweichung und die mittlere absolute Abweichung vom Median sind keine äquivalenten Streuungsmaße – selbst dann nicht, wenn die Daten normalverteilt sind. So ist die Standardabweichung immer größer als die mittlere absolute Abweichung, die ihrerseits größer als die mittlere absolute Abweichung vom Median ist. Manchmal wird die mittlere absolute Abweichung vom Median mit einem konstanten Skalierungsfaktor multipliziert, um den Wert für den Fall, dass die Daten normalverteilt sind, genau so zu skalieren wie die Standardabweichung. Der üblicherweise verwendete Faktor von 1,4826 bedeutet, dass 50% der Normalverteilung in den Bereich ±MAD fallen (siehe z.B. *https://oreil.ly/SfDk2*).

Streuungsmaße auf Basis von Perzentilen

Ein anderer Ansatz zur Schätzung der Streuung basiert auf der Betrachtung der Streuung der sortierten Daten. Statistiken, die auf sortierten (d.h. in einer Rang-

folge geordneten) Daten basieren, werden als *Ordnungsstatistiken* bezeichnet. Das grundlegende Maß ist die *Spannweite*: die Differenz zwischen dem größten und dem kleinsten Wert. Die Minimal- und Maximalwerte selbst sind zwar durchaus interessant und bei der Identifizierung von Ausreißern nützlich, aber die Spannweite erweist sich als äußerst empfindlich gegenüber Ausreißern und ist als allgemeines Streuungsmaß nicht sehr hilfreich.

Um der Anfälligkeit gegenüber Ausreißern vorzubeugen, können wir vor der Ermittlung der Spannweite Werte an beiden Enden der Daten weglassen. Formal basieren diese Arten von Schätzern auf Unterschieden zwischen *Perzentilen*. In einem Datensatz ist das *P*%-Perzentil so definiert, dass mindestens *P* % der Werte diesen Wert oder weniger und mindestens (100 – *P*) % der Werte diesen Wert oder mehr annehmen. Um zum Beispiel das 80%-Perzentil zu ermitteln, müssen Sie die Daten zunächst sortieren. Dann gehen Sie, beginnend beim kleinsten Wert, 80% der Strecke zum größten Wert weiter. Der Median ist übrigens ein und dasselbe wie das 50%-Perzentil. Ein Perzentil ist im Wesentlichen dasselbe wie ein *Quantil*, wobei Quantile durch Bruchzahlen angegeben werden (das 0,8-Quantil ist also dasselbe wie das 80%-Perzentil).

Ein gebräuchliches Streuungsmaß ist die Differenz zwischen dem 25%-Perzentil und dem 75%-Perzentil, der sogenannte *Interquartilsabstand* (engl. *Interquartile Range*, IQR). Hier ist ein einfaches Zahlenbeispiel: {3,1,5,3,6,7,2,9}. Wir sortieren diese Zahlenfolge, um {1,2,3,3,5,6,7,9} zu erhalten. Das 25%-Perzentil liegt bei 2,5 und das 75%-Perzentil bei 6,5, sodass der Interquartilsabstand 6,5 – 2,5 = 4 beträgt. Die Softwareprogramme können leicht unterschiedliche Ansätze haben, die dann unterschiedliche Ergebnisse hervorbringen (siehe folgenden Hinweis); in der Regel fallen diese Unterschiede jedoch gering aus.

Bei sehr großen Datensätzen kann die Berechnung der genauen Perzentile rechnerisch sehr aufwendig sein, da dazu alle Datenwerte sortiert werden müssen. Maschinelle Lern- und Statistikprogramme verwenden spezielle Algorithmen, wie [Zhang-Wang-2007], um einen Näherungswert für ein Perzentil zu erhalten, der sehr schnell berechnet werden kann und eine gewisse Genauigkeit gewährleistet.

Perzentile: Genaue Definition

Wenn wir eine gerade Anzahl an Werten haben (*n* ist gerade), dann ist das Perzentil im Sinne der vorhergehenden Definition mehrdeutig. Tatsächlich könnten wir jeden Wert zwischen der Ordnungsstatistik $x_{(j)}$ und $x_{(j+1)}$ nehmen, wobei *j* Folgendes erfüllt:

$$100 * \frac{j}{n} \leq P < 100 * \frac{j+1}{n}$$

In formaler Hinsicht ist das Perzentil ein gewichteter Durchschnitt:

$$\text{Perzentil}(P) = (1 - w)x_{(j)} + wx_{(j+1)}$$

für ein gegebenes Gewicht *w* zwischen 0 und 1. In den verschiedenen verfügbaren Statistikprogrammen gibt es leicht unterschiedliche

Ansätze für die Auswahl von *w*. Tatsächlich bietet die *R*-Funktion `quantile` neun verschiedene Alternativen zur Berechnung des Quantils. Von kleinen Datensätzen abgesehen, brauchen Sie sich in der Regel keine Gedanken darüber zu machen, wie ein Perzentil genau berechnet wird. In *Python* unterstützt das `numpy.quantile` fünf Ansätze, wobei die lineare Interpolation voreingestellt ist.

Beispiel: Streuungsmaße für die Einwohnerzahlen der Bundesstaaten in den USA

Tabelle 1-3 (Tabelle 1-2 wird der Einfachheit halber erneut dargestellt) zeigt die ersten paar Zeilen im Datensatz, in dem die Einwohnerzahlen und Mordraten für jeden US-Bundesstaat enthalten sind.

Tabelle 1-3: Die ersten Zeilen des data.frame mit den Einwohnerzahlen und Mordraten nach Bundesstaaten

	Bundesstaat	Einwohnerzahl	Mordrate	Abkürzung
1	Alabama	4.779.736	5,7	AL
2	Alaska	710.231	5,6	AK
3	Arizona	6.392.017	4,7	AZ
4	Arkansas	2.915.918	5,6	AR
5	California	37.253.956	4,4	CA
6	Colorado	5.029.196	2,8	CO
7	Connecticut	3.574.097	2,4	CT
8	Delaware	897.934	5,8	DE

Unter Verwendung der in *R* integrierten Funktionen für die Standardabweichung, den Interquartilsabstand (IQR) und die mittlere absolute Abweichung vom Median können wir Streuungsmaße für die Einwohnerstatistiken der Bundesstaaten berechnen:

```
> sd(state[['Population']])
[1] 6848235
> IQR(state[['Population']])
[1] 4847308
> mad(state[['Population']])
[1] 3849870
```

Für ein Data-Frame-Objekt stehen uns in der `pandas`-Bibliothek verschiedene Methoden zur Berechnung der Standardabweichung und der Quantile zur Verfügung. Nach Ermittlung der Quantilswerte können wir den IQR berechnen. Für die robuste mittlere absolute Abweichung vom Median verwenden wir die Funktion `robust.scale.mad` aus dem `statsmodels`-Paket:

```
state['Population'].std()
state['Population'].quantile(0.75) - state['Population'].quantile(0.25)
robust.scale.mad(state['Population'])
```

Die Standardabweichung ist fast doppelt so groß wie die MAD (in *R* wird die Skalierung der mittleren absoluten Abweichung vom Median standardmäßig so angepasst, dass der Mittelwert die gleiche Skalierung besitzt). Dies ist nicht weiter verwunderlich, da die Standardabweichung gegenüber Ausreißern sensibel ist.

Kernideen

- Die Varianz und die Standardabweichung sind die am weitesten verbreiteten und routinemäßig berichteten Streuungsmaße.
- Beide sind empfindlich gegenüber Ausreißern.
- Zu den robusteren Maßen gehören die mittlere absolute Abweichung, die mittlere absolute Abweichung vom Median und Perzentile (Quantile).

Weiterführende Literatur

- David Lanes Online-Statistik-Ratgeber hat einen Abschnitt über Perzentile (*https://oreil.ly/o2fBI*).
- Kevin Davenport hat einen nützlichen Beitrag auf *R*-Bloggers (*https://oreil.ly/E7zcG*) über Abweichungen vom Median und ihre robusten Eigenschaften verfasst.

Exploration der Datenverteilung

Alle von uns behandelten Maße fassen die Daten in einer einzigen Zahl zusammen, um die Lage oder die Streuung der Daten zu beschreiben. Es ist auch wertvoll, zu untersuchen, wie die komplette Verteilung der Daten aussieht.

Schlüsselbegriffe zur Exploration von Verteilungen

Box-Plot
: Ein von Tukey eingeführtes Diagramm zur schnellen Visualisierung der Datenverteilung.

Synonyme
: Box-Whisker-Plot, Kastengrafik

Häufigkeitstabelle
: Eine Übersicht über die Anzahl der numerischen Werte, die in eine Menge von Intervallen (Klassen, engl. *Bins*) fallen.

Histogramm
: Ein Diagramm der Häufigkeitstabelle mit den Intervallen auf der x-Achse und der Anzahl (oder dem relativen Anteil) auf der y-Achse. Balkendiagramme sind zwar ähnlich, sollten aber nicht mit Histogrammen verwechselt werden (siehe »Binäre und kategoriale Daten untersuchen« auf Seite 28 für eine Erläuterung des Unterschieds).

Dichtediagramm
: Eine geglättete Version des Histogramms, oft basierend auf einer *Kerndichteschätzung* (engl. *Kernel Density Estimate*).

Perzentile und Box-Plots

In »Streuungsmaße auf Basis von Perzentilen« auf Seite 17 sind wir der Frage nachgegangen, wie Perzentile zur Messung der Streuung der Daten verwendet werden können. Perzentile sind auch nützlich, um die gesamte Verteilung zusammenfassend darzustellen. Es ist üblich, die Quartile (25%-, 50%- und 75%-Perzentile) und die Dezile (10%-, 20%-, ..., 90%-Perzentile) anzugeben. Perzentile sind besonders aussagekräftig, wenn man die *Enden* bzw. *Ränder* (die äußeren Bereiche) der Verteilung zusammenzufassend darstellen möchte. In der breiten Öffentlichkeit ist in diesem Zusammenhang oft von der Redewendung »das eine Prozent« die Rede, die genutzt wird, um Reiche im obersten 99%-Perzentil der Vermögens- bzw. Einkommensverteilung zu charakterisieren.

Tabelle 1-4 stellt einige Perzentile der Mordraten in den Bundesstaaten dar. In *R* können wir uns die Werte mithilfe der Funktion `quantile` ausgeben lassen:

```
quantile(state[['Murder.Rate']], p=c(.05, .25, .5, .75, .95))
   5%   25%   50%   75%   95%
1.600 2.425 4.000 5.550 6.510
```

In *Python* können Sie für einen Data Frame die `pandas`-Methode `quantile` nutzen, um sich die Perzentile ausgeben zu lassen:

```
state['Murder.Rate'].quantile([0.05, 0.25, 0.5, 0.75, 0.95])
```

Tabelle 1-4: Perzentile der Mordraten in den Bundesstaaten

5%	25%	50%	75%	95%
1,60	2,42	4,00	5,55	6,51

Der Median liegt bei vier Morden pro 100.000 Einwohner. In den Daten gibt jedoch eine beträchtliche Streuung: Das 5%-Perzentil beträgt nur 1,6 und das 95%-Perzentil 6,51.

Die von Tukey [Tukey-1977] eingeführten *Box-Plots* stützen sich auf Perzentile und bieten eine rasche Möglichkeit, die Verteilung Ihrer Daten zu visualisieren. Abbil-

dung 1-2 zeigt einen in *R* erstellten Box-Plot für die Einwohnerzahlen der Bundesstaaten:

```
boxplot(state[['Population']]/1000000, ylab='Einwohnerzahl (in Millionen)')
```

Die pandas-Bibliothek bietet eine Reihe von grundlegenden informativen Diagrammen, die für Data Frames genutzt werden können; darunter auch Box-Plots:

```
ax = (state['Population']/1_000_000).plot.box()
ax.set_ylabel('Einwohnerzahl (in Millionen)')
```

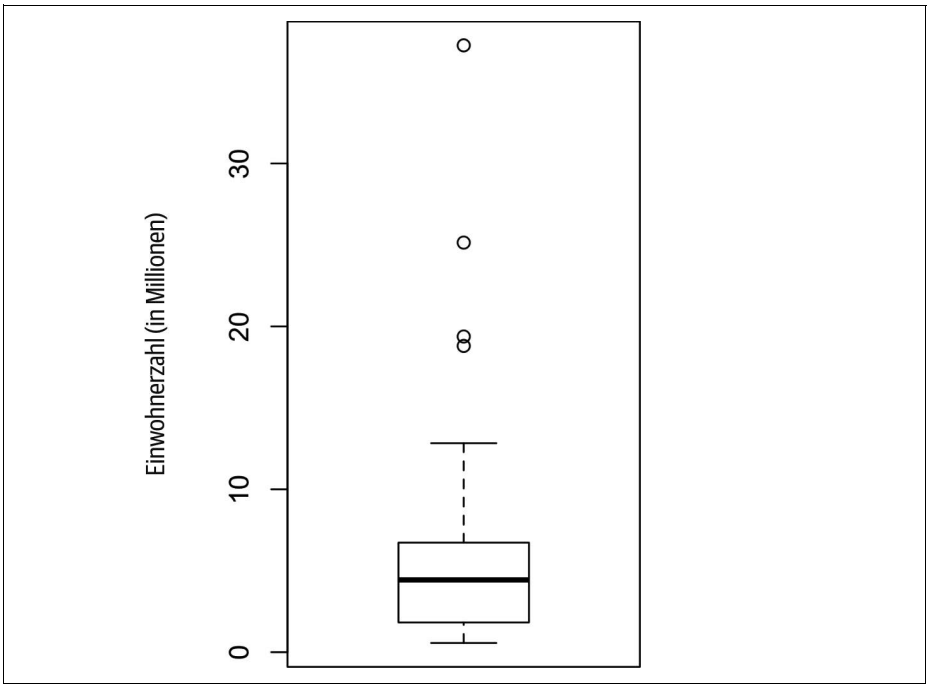

Abbildung 1-2: Box-Plot für die Einwohnerzahlen der Bundesstaaten

Bei diesem Box-Plot können wir auf einen Blick erkennen, dass die mittlere Einwohnerzahl der Bundesstaaten etwa fünf Millionen beträgt (Median), die Einwohnerzahl für die Hälfte der Staaten zwischen etwa zwei und sieben Millionen liegt und dass es einige bevölkerungsreiche Ausreißer gibt. Der obere und der untere Rand des Rechtecks (Box) kennzeichnen jeweils das 75%- bzw. 25%-Perzentil. Der Median wird durch die fett gehaltene horizontale Linie in der Box angezeigt. Die beiden vertikalen gestrichelten Linien, die als *Whisker* oder auch Antennen bezeichnet werden, erstrecken sich über den oberen und unteren Rand der Box, um den Bereich, in dem der überwiegende Teil der Daten liegt, zu kennzeichnen. Es gibt zahlreiche Varianten von Box-Plots (siehe z.B. die Dokumentation der *R*-Funktion `boxplot` [R-base-2015]). Standardmäßig verlängert die *R*-Funktion die Whisker bis zu dem Datenpunkt, der am weitesten über die Box hinausgeht – jedenfalls immer dann, wenn dieser Wert nicht mehr als das 1,5-Fache des IQR be-

trägt. Die `matplotlib`-Bibliothek verwendet die gleiche Implementierung. In anderen Softwareprogrammen kann eine abweichende Regel angewandt werden.

Alle Datenpunkte außerhalb der Whisker werden als einzelne Punkte oder Kreise dargestellt (die in der Regel als Ausreißer angesehen werden).

Häufigkeitstabellen und Histogramme

Eine Häufigkeitstabelle teilt den Wertebereich einer Variablen bzw. eines Merkmals in gleich große Intervalle auf und gibt uns Auskunft darüber, wie viele Werte jeweils in jedes Intervall fallen. Tabelle 1-5 zeigt Ihnen eine in *R* erstellte Häufigkeitstabelle für die Einwohnerzahlen der Bundesstaaten:

```
breaks <- seq(from=min(state[['Population']]),
              to=max(state[['Population']]), length=11)
pop_freq <- cut(state[['Population']], breaks=breaks,
              right=TRUE, include.lowest=TRUE)
table(pop_freq)
```

Die Funktion `pandas.cut` erzeugt eine Zahlenfolge (`Series`-Objekt), die die Werte auf die einzelnen Intervalle abbildet. Mit der Methode `value_counts` erhalten wir die Häufigkeitstabelle:

```
binnedPopulation = pd.cut(state['Population'], 10)
binnedPopulation.value_counts()
```

Tabelle 1-5: Eine Häufigkeitstabelle für die Einwohnerzahlen der Bundesstaaten

Intervallnummer	Intervall	Häufigkeit	Bundesstaaten
1	563.626–4.232.658	24	WY,VT,ND,AK,SD,DE,MT,RI,NH,ME,HI,ID,NE,WV,NM,NV,UT,KS,AR,MS,IA,CT,OK,OR
2	4.232.659–7.901.691	14	KY,LA,SC,AL,CO,MN,WI,MD,MO,TN,AZ,IN,MA,WA
3	7.901.692–11.570.724	6	VA,NJ,NC,GA,MI,OH
4	11.570.725–15.239.757	2	PA,IL
5	15.239.758–18.908.790	1	FL
6	18.908.791–22.577.823	1	NY
7	22.577.824–26.246.856	1	TX
8	26.246.857–29.915.889	0	
9	29.915.890–33.584.922	0	
10	33.584.923–37.253.956	1	CA

Der Bundesstaat mit der geringsten Einwohnerzahl ist Wyoming mit 563.626 Menschen, und der bevölkerungsreichste ist Kalifornien mit 37.253.956 Einwohnern. Daraus ergibt sich ein Wertebereich von 37.253.956 – 563.626 = 36.690.330, den wir in gleich große Klassen – sagen wir 10 – aufteilen müssen. Bei 10 gleich großen Klassen hat jede Klasse eine Breite von 3.669.033, sodass das erste Intervall von

563.626 bis 4.232.658 reicht. Im Gegensatz dazu liegt in der obersten Klasse für das Intervall 33.584.923 bis 37.253.956 Einwohnern nur ein einziger Bundesstaat: Kalifornien. Die beiden nächstkleineren Klassen sind unbesetzt, bis als Nächstes der Bundesstaat Texas erreicht wird. Es ist wichtig, die leeren Klassen mit einzubeziehen; die Tatsache, dass sich in diesen Intervallen keine Werte befinden, ist eine nützliche Information. Es kann auch hilfreich sein, mit verschiedenen Klassenbreiten bzw. Intervallgrößen zu experimentieren. Wenn sie zu groß sind, treten wichtige Merkmale der Verteilung gegebenenfalls nicht mehr sichtbar hervor. Werden sie zu klein gewählt, ist das Ergebnis zu feingliedrig, und die Fähigkeit, ein adäquates Gesamtbild zu liefern, geht verloren.

Sowohl Häufigkeitstabellen als auch Perzentile fassen die Daten durch die Einteilung in Klassen bzw. Intervalle übersichtlich zusammen. Im Allgemeinen haben Quartile und Dezile in jeder Klasse die gleiche Anzahl an Beobachtungen (Klassen mit gleicher Anzahl – *equal-count bins*), aber die Klassenbreite ist für gewöhnlich unterschiedlich. Bei der Häufigkeitstabelle umfassen die Klassen hingegen eine unterschiedliche Anzahl an Beobachtungen, wohingegen die Klassenbreite identisch ist (gleich breite Klassen – *equal-size bins*).

Ein Histogramm bietet die Möglichkeit, eine Häufigkeitstabelle zu visualisieren. Dabei werden die Klassen auf der x-Achse abgetragen und die Anzahl der Beobachtungen bzw. Häufigkeiten auf der y-Achse. In Abbildung 1-3 erstreckt sich die bei zehn Millionen (1e+07) Einwohnern in der Mitte befindliche Klasse von ungefähr acht bis zwölf Millionen Einwohnern und umfasst insgesamt sechs Beobachtungen. Um ein Histogramm in *R* zu erstellen, das dem in Tabelle 1-5 entspricht, müssen Sie lediglich die Funktion `hist` mit dem Argument `breaks` verwenden:

```
hist(state[['Population']], breaks=breaks)
```

Die `pandas`-Bibliothek unterstützt die Erstellung von Histogrammen für DataFrame-Objekte mit der Methode `DataFrame.plot.hist`. Mit dem Schlüsselwortargument `bins` können Sie die Anzahl der Klassen bestimmen. Die verfügbaren `plot`-Methoden geben jeweils ein Objekt zurück, das die Koordinaten in Bezug auf die Achsen bereithält und eine weitere Feinabstimmung des Diagramms mithilfe der `matplotlib`-Bibliothek ermöglicht:

```
ax = (state['Population'] / 1_000_000).plot.hist(figsize=(4, 4))
ax.set_xlabel('Einwohnerzahl (in Millionen)')
```

Das Histogramm wird in Abbildung 1-3 gezeigt. Im Allgemeinen werden Histogramme so erstellt:

- Auch unbesetzte Klassen werden in die Darstellung mit einbezogen.
- Die Klassen sind gleich breit.
- Die Wahl der Anzahl der Klassen (oder, äquivalent, der Klassenbreite) ist dem Anwender überlassen.

- Die Balken sind direkt aneinander angrenzend – es entsteht kein Abstand zwischen den Balken, es sei denn, es liegt eine unbesetzte Klasse vor.

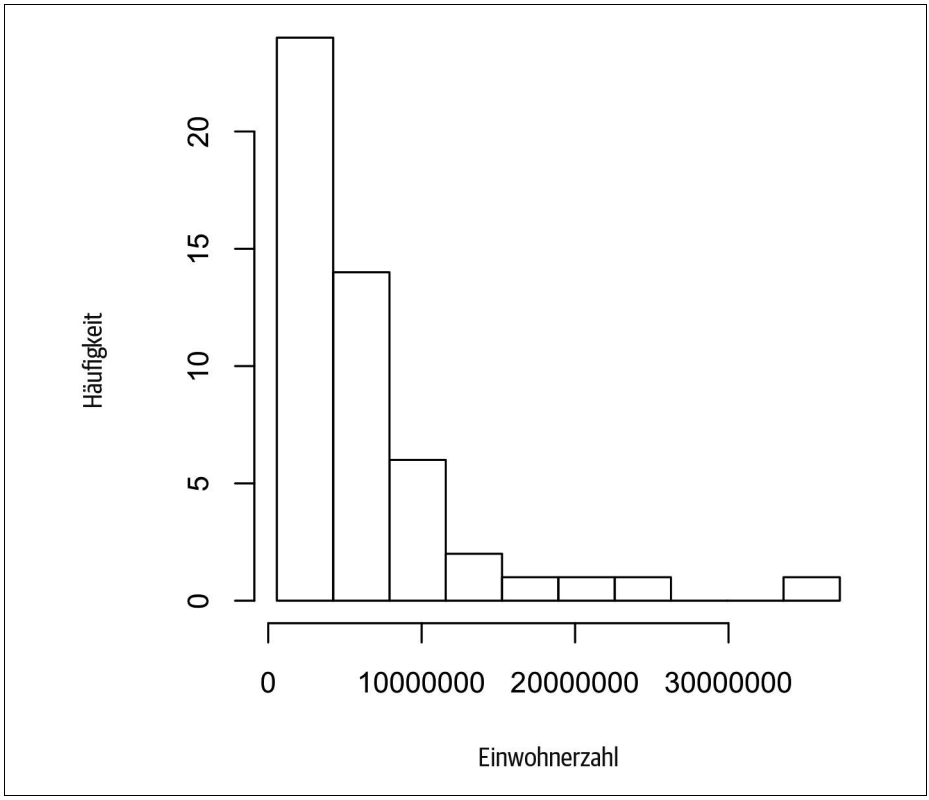

Abbildung 1-3: Ein Histogramm der Einwohnerzahlen der Bundesstaaten

Statistische Momente

In der statistischen Theorie werden die Lage und die Streuung als die ersten und zweiten *Momente* einer Verteilung bezeichnet. Das dritte Moment ist als *Schiefe* (engl. *Skewness*) und vierte als *Wölbung* (engl. *Kurtosis*) bekannt. Die Schiefe bezieht sich darauf, ob die Daten zu größeren oder kleineren Werten verzerrt sind, und die Wölbung gibt die Tendenz der Daten zu Extremwerten an. In der Regel werden Metriken zur Messung von Schiefe und Wölbung nicht herangezogen; stattdessen werden diese durch Visualisierungen wie die in den Abbildungen 1-2 und 1-3 erkundet.

Dichtediagramme und -schätzer

Eng verwandt mit dem Histogramm ist das Dichtediagramm, das die Verteilung der Datenwerte in Form einer durchgängigen Linie zeigt. Ein Dichtediagramm kann man sich als geglättetes Histogramm vorstellen, wobei es jedoch normalerweise direkt aus den Daten durch eine *Kerndichteschätzung* berechnet wird (siehe

[Duong-2001] für ein kurzes Einführungsbeispiel).Abbildung 1-4 stellt ein Histogramm dar, das von einer Dichteschätzung überlagert ist. In *R* können Sie eine Kerndichteschätzung mithilfe der Funktion `+density()+` vornehmen:

```
hist(state[['Murder.Rate']], freq=FALSE)
lines(density(state[['Murder.Rate']]), lwd=3, col='blue')
```

`pandas` bietet ebenfalls eine `density`-Methode zur Erstellung eines Dichtediagramms. Mit dem Argument `bw_method` steuern Sie die Glättung der Dichtekurve:

```
ax = state['Murder.Rate'].plot.hist(density=True, xlim=[0,12], bins=range(1,12))
state['Murder.Rate'].plot.density(ax=ax) ❶
ax.set_xlabel('Mordrate (pro 100.000)')
```

❶ In den `plot`-Funktionen können Sie meist ein optionales Argument (`ax`) bereitstellen, das bewirkt, dass die Abbildung in dasselbe Diagramm eingezeichnet wird.

Ein wesentlicher Unterschied zu dem Histogramm, das in Abbildung 1-3 gezeigt wird, besteht in der Skalierung der y-Achse: Ein Dichtediagramm entspricht der Darstellung des Histogramms, das einen relativen Anteil wiedergibt, und keine Absolutwerte (Anzahl bzw. Häufigkeit; Sie geben dies in *R* mit dem Argument `freq=FALSE` an). Beachten Sie, dass die Gesamtfläche unter der Dichtekurve 1 beträgt, und anstelle der Klassenanzahl berechnen Sie hierbei Flächen, die unterhalb der Kurve zwischen zwei beliebigen Punkten auf der x-Achse liegen, die dem relativen Anteil der zwischen diesen beiden Punkten liegenden Verteilung zur Gesamtverteilung entspricht.

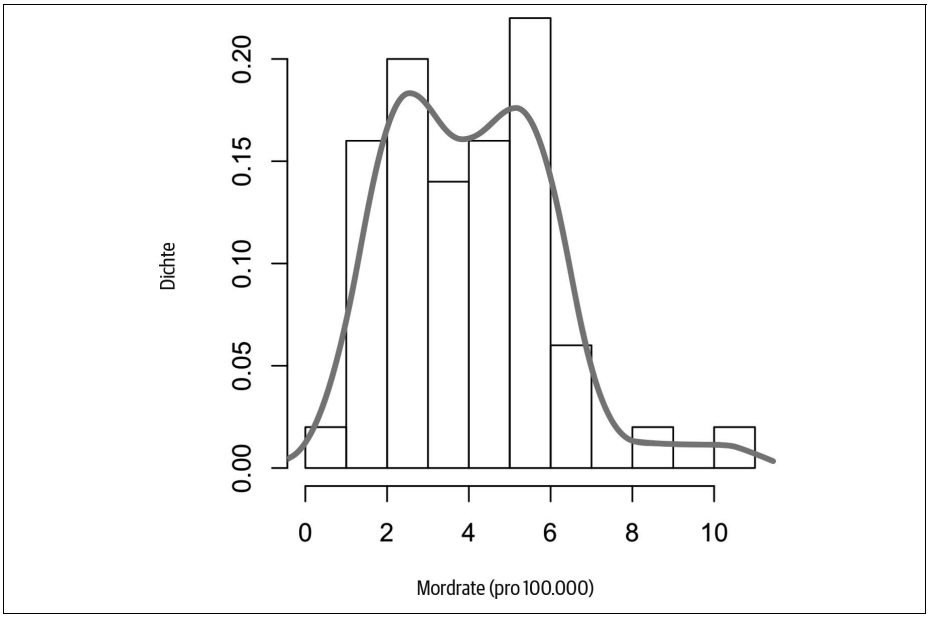

Abbildung 1-4: Die geschätzte Dichtefunktion für die Mordraten aller Bundesstaaten

Dichteschätzung

Die Dichteschätzung ist ein weitreichendes Thema mit einer langen Geschichte in der statistischen Literatur. Tatsächlich wurden über 20 *R* Pakete veröffentlicht, die Funktionen zur Dichteschätzung bereitstellen. [Deng-Wickham-2011] geben einen umfassenden Überblick über die verschiedenen Implementierungen in *R*, wobei sie insbesondere die Pakete `ASH` und `KernSmooth` empfehlen. Auch `pandas` und `scikit-learn` bieten hervorragende Methoden zur Dichteschätzung. Für viele datenwissenschaftliche Fragestellungen braucht man sich über die verschiedenen Arten von Dichteschätzern keine Gedanken zu machen; es genügt meist, die Basisfunktionen zu verwenden.

Kernideen

- In einem Histogramm werden die Häufigkeit der Beobachtungen auf der y-Achse und die jeweiligen Variablenwerte auf der x-Achse abgetragen. Es vermittelt auf einen Blick einen Eindruck von der Datenverteilung.
- Eine Häufigkeitstabelle entspricht der tabellarischen Darstellung der in einem Histogramm dargestellten Häufigkeiten.
- Ein Box-Plot – bei dem der obere Rand der Box beim 75 %- und der untere beim 25 %-Perzentil liegen – vermittelt ebenfalls einen schnellen Überblick über die Datenverteilung; er wird oft nebeneinander dargestellt, um Verteilungen zu vergleichen.
- Ein Dichtediagramm ist eine geglättete Version eines Histogramms; es erfordert eine Funktion zur Schätzung der Kurve auf Grundlage der Daten (wobei natürlich mehrere Schätzverfahren möglich sind).

Weiterführende Literatur

- Ein Professor der SUNY Oswego hat eine Schritt-für-Schritt-Anleitung zum Erstellen eines Box-Plots (*https://oreil.ly/wTpnE*) zur Verfügung gestellt.
- Die verschiedenen in *R* zur Verfügung stehenden Ansätze zur Dichteschätzung werden in Henry Dengs und Hadley Wickhams Artikel (*https://oreil.ly/TbWYS*) »Density estimation in R« behandelt.
- In dem Blog »*R*-Bloggers« finden Sie einen hilfreichen Beitrag zur Erstellung von Histogrammen in *R* (*https://oreil.ly/Ynp-n*), der auch Anpassungselemente wie die Einteilung in Klassen (*Binning*) bzw. die Auswahl der Klassengrenzen (engl. *Breaks*) erläutert.
- Das Blog hält auch einen ähnlichen Blogbeitrag zu Box-Plots in *R* (*https://oreil.ly/0DSb2*) bereit.

- Matthew Conlen hat eine interaktive Webpräsentation (*https://oreil.ly/bC9nu*) veröffentlicht, die die Auswirkungen der Wahl verschiedener Kerne und Bandbreiten auf die Kerndichteschätzer demonstriert.

Binäre und kategoriale Daten untersuchen

Bei kategorialen Daten genügen einfache Angaben in Form von relativen Anteilen bzw. Prozentsätzen, um die Datenlage nachvollziehbar wiederzugeben.

Schlüsselbegriffe zur Exploration kategorialer Daten

Modus
: Die am häufigsten vorkommende Kategorie oder der am häufigsten vorkommende Wert in einem Datensatz (engl. *Mode*).

Erwartungswert
: Wenn die Kategorien mit einem numerischen Wert verknüpft werden können, ergibt sich ein Durchschnittswert auf Basis der Eintrittswahrscheinlichkeiten der Kategorien (engl. *Expected Value*).

Balkendiagramm
: Die Häufigkeiten oder relativen Anteile aller Kategorien, in Form von Balken dargestellt (engl. *Bar Chart*).

Kreisdiagramm
: Die Häufigkeiten oder relativen Anteile aller Kategorien in Form von Kreissektoren als Teile eines Kreises, auch als Kuchen- oder Tortendiagramm bekannt (engl. *Pie Chart*).

Sich eine Übersicht über eine binäre Variable oder eine kategoriale Variable mit einigen wenigen Kategorien zu verschaffen, ist eine ziemlich einfache Angelegenheit: Wir müssen nur den Anteil von »Einsen« bzw. Erfolgen im binären Fall oder in Bezug auf kategoriale Daten die Anteile der im Fokus stehenden Kategorien ermitteln. Zum Beispiel zeigt Tabelle 1-6 die prozentuale Verteilung der verspäteten Flüge am Flughafen Dallas/Fort Worth im Jahr 2010, aufgeschlüsselt nach ihrem Verspätungsgrund. Die Verspätungen werden dabei in Kategorien unterteilt, die Aufschluss über den Grund der Verspätung geben. Hierzu zählen solche, die dem Verantwortungsbereich der Fluggesellschaft unterliegen, Verspätungen, die auf die Flugverkehrskontrolle (FVK) zurückzuführen sind, auf das Wetter, auf die Sicherheitsvorkehrungen oder auch auf ein verspätet eintreffendes Flugzeug für einen Anschlussflug.

Tabelle 1-6: Prozentuale Verteilung der Verspätungen am Flughafen Dallas/Fort Worth, aufgeschlüsselt nach ihrem Verspätungsgrund

Fluggesellschaft	FVK	Wetter	Sicherheit	Anschluss
23.02	30.40	4.03	0.12	42.43

Die auch häufig in der Tagespresse anzutreffenden Balkendiagramme sind ein gängiges visuelles Hilfsmittel zur Darstellung einer einzelnen kategorialen Variablen. Die Kategorien werden auf der x-Achse und die jeweiligen Häufigkeiten oder die relativen Anteile auf der y-Achse dargestellt. Abbildung 1-5 zeigt die Flugverspätungen, sortiert nach Verspätungsgrund, für den Flughafen Dallas/Fort Worth (DFW) im Jahr 2010. Sie kann relativ simpel mit der *R*-Funktion `barplot` erstellt werden:

```
barplot(as.matrix(dfw) / 6, cex.axis=0.8, cex.names=0.7,
        xlab='Verspätungsgrund', ylab='Anzahl')
```

Die pandas-Bibliothek unterstützt ebenfalls Balkendiagramme:

```
ax = dfw.transpose().plot.bar(figsize=(4, 4), legend=False)
ax.set_xlabel('Verspätungsgrund')
ax.set_ylabel('Anzahl')
```

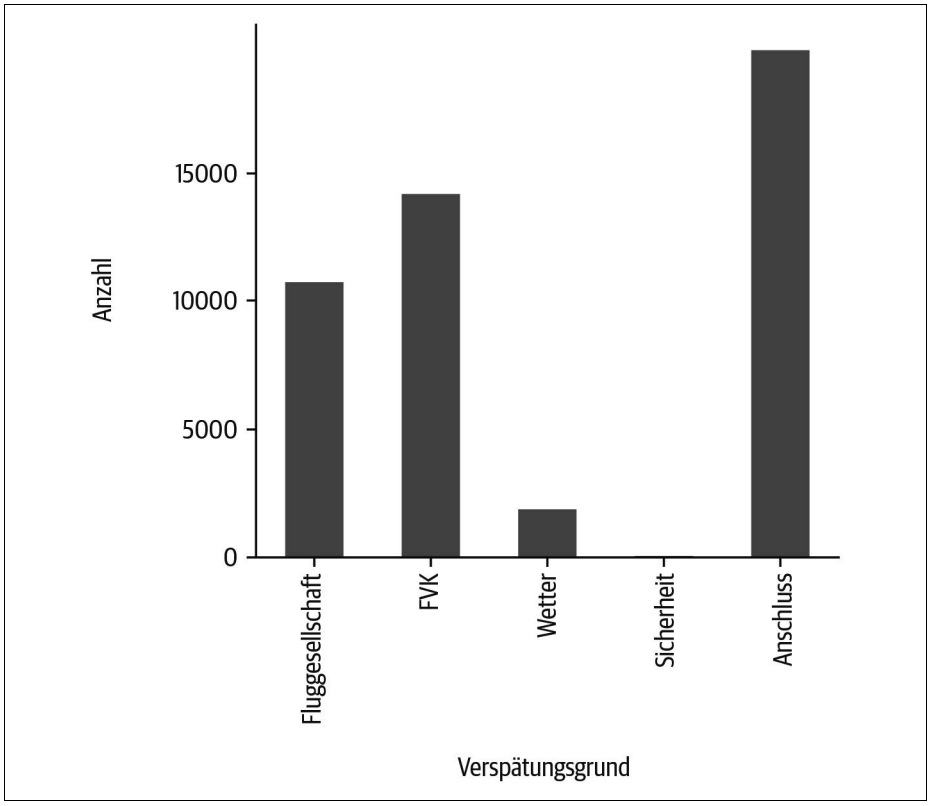

Abbildung 1-5: Balkendiagramm der Flugverspätungen am DFW nach Verspätungsgrund

Beachten Sie, dass ein Balkendiagramm einem Histogramm ähnelt; in einem Balkendiagramm repräsentiert die x-Achse verschiedene Kategorien einer Faktorvariablen, während in einem Histogramm die x-Achse die Werte einer einzelnen Variablen numerisch skaliert darstellt. In einem Histogramm werden die Balken typischerweise aneinander angrenzend dargestellt, wobei Lücken auf Werte hinweisen, die in den Daten nicht vorkommen. In einem Balkendiagramm werden die Balken getrennt voneinander dargestellt.

Kreisdiagramme stellen eine Alternative zu Balkendiagrammen dar, obwohl Statistiker und Datenvisualisierungsexperten im Allgemeinen Kreisdiagramme als weniger informativ ansehen (siehe [Few-2007]).

Numerische Daten als kategoriale Daten

In »Häufigkeitstabellen und Histogramme« auf Seite 23 betrachteten wir Häufigkeitstabellen, die auf einer Klasseneinteilung der Daten basieren. Dadurch werden die numerischen Daten implizit in einen geordneten Faktor umgewandelt. In diesem Sinne sind Histogramme und Balkendiagramme einander ähnlich, außer dass die Kategorien auf der x-Achse im Balkendiagramm nicht geordnet sind. Die Konvertierung numerischer Daten in kategoriale Daten ist ein wichtiger und weitverbreiteter Schritt in der Datenanalyse, da er die Komplexität (und die Größe bzw. den Umfang) der Daten verringert. Dies hilft beim Aufdecken von Beziehungen zwischen Merkmalen, insbesondere in den ersten Schritten einer Analyse.

Modus

Der Modus ist der Wert – bzw. die Werte für den Fall, dass verschiedene Werte gleich häufig auftreten –, der am häufigsten in den Daten auftritt. Zum Beispiel ist der Modus des Verspätungsgrunds am Flughafen Dallas/Fort Worth der »verzögerte Anschlussflug«. Ein weiteres Beispiel: In den meisten Teilen der Vereinigten Staaten wäre der Modus der religiösen Gesinnung »christlich«. Der Modus ist eine einfache, zusammenfassende Statistik für kategoriale Daten. Bei numerischen Daten wird er im Allgemeinen nicht verwendet.

Erwartungswert

Ein besonderer Typ kategorialer Daten sind Daten, bei denen die Kategorien diskrete Werte auf derselben Skala repräsentieren oder diesen zugeordnet werden können. Ein Vermarkter für eine neue Cloud-Technologie bietet zum Beispiel zwei Servicevarianten an, eine zum Preis von 300 $ pro Monat und eine weitere zum Preis von 50 $ pro Monat. Der Vermarkter bietet kostenlose Webinare an, um Leads zu generieren, und das Unternehmen geht davon aus, dass sich 5 % der Teilnehmer für den Service für 300 $, 15 % für den Service für 50 $ und 80 % für keinen Service anmelden werden. Diese Daten lassen sich zu Zwecken der Wirtschaftlichkeitsberechnung in einem einzigen »Erwartungswert« zusammenfassen, der eine Art gewichteter Mittelwert ist, bei dem die Gewichte Wahrscheinlichkeiten entsprechen.

Der Erwartungswert wird wie folgt berechnet:

1. Multiplizieren Sie jedes der einzelnen Ergebnisse mit seiner Eintrittswahrscheinlichkeit.
2. Summieren Sie die ermittelten Werte.

Im Beispiel des Cloud-Service beträgt der Erwartungswert eines Webinar-Teilnehmers somit 22,50 $ pro Monat, der sich wie folgt ergibt:

$$EW = (0,05)(300) + (0,15)(50) + (0,80)(0) = 22,5$$

Der Erwartungswert ist in Wirklichkeit eine Art gewichteter Mittelwert: Er spiegelt künftige Erwartungen wider, wobei die Berechnung auf Wahrscheinlichkeitsgewichten basiert, die oft auf einem subjektiven Urteil beruhen. Der Erwartungswert stellt ein grundlegendes Konzept in der Unternehmensbewertung und Kapitalbedarfsrechnung dar, wo es z.B. darum geht, den Erwartungswert der Gewinne aus einer Neuanschaffung für einen Zeithorizont von fünf Jahren oder die erwarteten Kosteneinsparungen durch eine neue Patientenverwaltungssoftware in einer Klinik zu ermitteln.

Wahrscheinlichkeiten

Wir sind zuvor auf die *Wahrscheinlichkeit* des Auftretens eines Ereignisses eingegangen (die sogenannte Eintrittswahrscheinlichkeit). Die meisten Menschen haben ein intuitives Verständnis von Wahrscheinlichkeit und begegnen dem Konzept häufig bei Wettervorhersagen (Regenwahrscheinlichkeit) oder Sportanalysen (Gewinnwahrscheinlichkeit). Sport- und Spielergebnisse werden häufiger als Quoten ausgedrückt, die sich leicht in Wahrscheinlichkeiten umformulieren lassen. (Wenn die Wahrscheinlichkeit, dass eine Mannschaft gewinnt, 2 zu 1 ist, ist ihre Gewinnwahrscheinlichkeit 2/(2+1) = 2/3.) Überraschenderweise kann das Konzept der Wahrscheinlichkeit jedoch bei seiner Begriffsbestimmung Anlass zu tiefgreifenden philosophischen Diskussionen geben.

Glücklicherweise brauchen wir hier keine formale mathematische oder philosophische Definition. Aus unserer Perspektive ist die Wahrscheinlichkeit, dass ein Ereignis eintritt, einfach die relative Häufigkeit, mit der es auftreten würde – jedenfalls dann, wenn die Situation immer und immer wieder, und zwar unzählige Male, wiederholt werden könnte. Meistens handelt es sich dabei um ein imaginäres Konstrukt, aber es bietet uns die Möglichkeit, ein angemessenes praktisches Verständnis für Wahrscheinlichkeiten zu entwickeln.

Kernideen

- Kategoriale Daten werden typischerweise in Form relativer Anteilswerte zusammengefasst und können in einem Balkendiagramm visualisiert werden.
- Die Kategorien können verschiedene Dinge darstellen (Äpfel und Orangen, männlich und weiblich), die Stufen bzw. Niveaus einer Faktorvariablen (niedrig, mittel und hoch) oder numerische Daten, die in Intervalle bzw. Klassen eingeteilt wurden.
- Der Erwartungswert entspricht der Summe der Produkte der einzelnen Werte mit ihrer Eintrittswahrscheinlichkeit (eine mit den Eintrittswahrscheinlichkeiten der Werte gewichtete Summe) und wird oft verwendet, um die Niveaus von Faktorvariablen zusammenzufassen.

Weiterführende Literatur

Ein Statistikkurs ist nicht wirklich vollständig, wenn er nicht auch eine Lektion über irreführende Diagramme (*https://oreil.ly/rDMuT*) enthält, die sich oftmals auf Balken- und Kreisdiagramme bezieht.

Korrelation

Bei der explorativen Datenanalyse wird in vielen Projekten (sei es in der Data Science oder in der Forschung) zunächst die Korrelation zwischen den Prädiktoren selbst sowie zwischen den Prädiktoren und einer Zielvariablen untersucht. Man spricht davon, dass die Variablen X und Y (jeweils als Messdaten erfasst) positiv korreliert sind, wenn hohe Werte von X mit hohen Werten von Y und niedrige Werte von X mit niedrigen Werten von Y einhergehen. Wenn hohe Werte von X mit niedrigen Werten von Y zusammenfallen und umgekehrt, sind die Variablen negativ korreliert.

Schlüsselbegriffe zur Korrelation

Korrelationskoeffizient
: Eine Metrik, die angibt, wie eng numerische Variablen miteinander in Beziehung stehen (reicht von –1 bis +1).

Korrelationsmatrix
: Eine Tabelle, in der die Variablen sowohl in den Zeilen als auch in den Spalten abgebildet sind und die Zellwerte die Korrelationen zwischen den Variablen darstellen.

Streudiagramm
: Eine Visualisierung, bei der die x-Achse den Wert einer Variablen und die y-Achse den Wert einer anderen angibt (engl. *Scatterplot*).

Betrachten Sie diese beiden Variablen, die insofern perfekt miteinander korreliert sind, dass beide – von niedrigen Werten ausgehend – kontinuierlich ansteigen:

- v1: {1, 2, 3}
- v2: {4, 5, 6}

Das Skalarprodukt beider Vektoren, auch inneres Produkt oder Punktprodukt genannt (engl. *Dot Product* bzw. *Vector Sum of Products*), ergibt $1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$. Versuchen Sie nun, einen von beiden Vektoren neu anzuordnen und das Skalarprodukt erneut zu berechnen – das Skalarprodukt wird niemals höher als 32 sein. Das Skalarprodukt könnte also als ein Maß verwendet werden, d.h., die ermittelte Summe von 32 könnte mit vielen zufälligen anderen Kombinationen verglichen werden (tatsächlich bezieht sich diese Idee auf einen Schätzer, der auf einer Resampling-Verteilung basiert; siehe »Permutationstest« auf Seite 101). Die mit diesem Maß erzeugten Werte sind jedoch nicht so aussagekräftig, außer in Bezug auf die Resampling-Verteilung (d.h., aus den gegebenen Daten werden wiederholt Stichproben gezogen).

Von größerem Nutzen ist die standardisierte Variante: der *Korrelationskoeffizient*, der einen Schätzer der Korrelation zwischen zwei Variablen darstellt, der immer auf derselben Skala liegt. Um den *pearsonschen Korrelationskoeffizienten* zu berechnen, multiplizieren wir jeweils die Abweichungen vom Mittelwert der Elemente von Variable 1 mit denen von Variable 2, bilden die Summe dieser Produkte und dividieren das Ergebnis durch das Produkt der Standardabweichungen:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

Beachten Sie, dass wir durch $n - 1$ statt n dividieren (siehe »Die Anzahl der Freiheitsgrade und die Frage, ob n oder n – 1?« auf Seite 16 für weitere Erläuterungen). Der Korrelationskoeffizient liegt immer zwischen +1 (perfekte positive Korrelation) und –1 (perfekte negative Korrelation); 0 bedeutet, dass die Variablen unkorreliert sind.

Variablen können in einem nicht linearen Zusammenhang zueinander stehen. In diesem Fall ist der Korrelationskoeffizient möglicherweise keine brauchbare Metrik. Ein Beispiel hierfür ist der Zusammenhang zwischen den Steuersätzen und den erhobenen Steuereinnahmen: Wenn die Steuersätze – ausgehend von null – steigen, steigen auch die erhobenen Einnahmen. Sobald die Steuersätze jedoch ein gewisses hohes Niveau erreichen und sich einem Satz von 100 % nähern, nimmt die Steuerumgehung zu, und die Steuereinnahmen sinken sogar.

Tabelle 1-7, die als *Korrelationsmatrix* bezeichnet wird, zeigt die Korrelation zwischen den Tagesrenditen von Aktien der Telekommunikationsbranche von Juli 2012 bis Juni 2015. Aus der Tabelle können Sie ersehen, dass Verizon (VZ) und ATT (T) am stärksten korreliert sind. Level 3 (LVLT), bei dem es sich um ein Infrastrukturunternehmen handelt, weist die geringste Korrelation zu den anderen

auf. Beachten Sie, dass sich auf der Diagonalen nur Einsen befinden (die Korrelation einer Aktie mit sich selbst ist 1) und dass die Informationen oberhalb und unterhalb der Diagonalen redundant sind.

Tabelle 1-7: Korrelation zwischen den Tagesrenditen von Aktien der Telekommunikationsbranche

	T	CTL	FTR	VZ	LVLT
T	1.000	0.475	0.328	0.678	0.279
CTL	0.475	1.000	0.420	0.417	0.287
FTR	0.328	0.420	1.000	0.287	0.260
VZ	0.678	0.417	0.287	1.000	0.242
LVLT	0.279	0.287	0.260	0.242	1.000

Korrelationstabellen wie Tabelle 1-7 werden üblicherweise visualisiert, um die Beziehung zwischen mehreren Variablen anschaulicher darzustellen. Abbildung 1-6 zeigt die Korrelation der Tagesrenditen der wichtigsten börsengehandelten Indexfonds (ETFs). In *R* können wir dies leicht mit dem Paket `corrplot` umsetzen:

```
etfs <- sp500_px[row.names(sp500_px) > '2012-07-01',
                 sp500_sym[sp500_sym$sector == 'etf', 'symbol']]
library(corrplot)
corrplot(cor(etfs), method='ellipse')
```

Es ist möglich, das gleiche Diagramm in *Python* zu erstellen. Es gibt jedoch leider keine Implementierung in einem der gängigen Pakete. Die meisten unterstützen allerdings die Visualisierung von Korrelationsmatrizen mithilfe von Heatmaps. Der folgende Code zeigt Ihnen die Umsetzung mithilfe des Moduls `seaborn.heatmap`. Im GitHub-Repository des Buchs stellen wir Ihnen zusätzlich eine *Python*-Implementierung der umfassenderen Visualisierung zur Verfügung:

```
etfs = sp500_px.loc[sp500_px.index > '2012-07-01',
                    sp500_sym[sp500_sym['sector'] == 'etf']['symbol']]
sns.heatmap(etfs.corr(), vmin=-1, vmax=1,
            cmap=sns.diverging_palette(20, 220, as_cmap=True))
```

Die ETFs für den S&P 500 (SPY) und den Dow-Jones-Index (DIA) weisen eine hohe Korrelation auf. In ähnlichem Maße sind der QQQ und der XLK, die hauptsächlich aus Technologieunternehmen bestehen, positiv korreliert. Defensive ETFs, wie z.B. diejenigen, die den Goldpreis (GLD), den Ölpreis (USO) oder die Marktvolatilität (VXX) abbilden, neigen dazu, nur schwach oder negativ mit den anderen ETFs korreliert zu sein. Die Ausrichtung der Ellipsen zeigt an, ob zwei Variablen positiv (Ellipse zeigt nach rechts oben) oder negativ korreliert sind (Ellipse zeigt nach links oben). Die Schattierung und die Breite der Ellipsen zeigen die Stärke der Korrelation an: Dünnere und dunklere Ellipsen bilden einen stärkeren Zusammenhang ab.

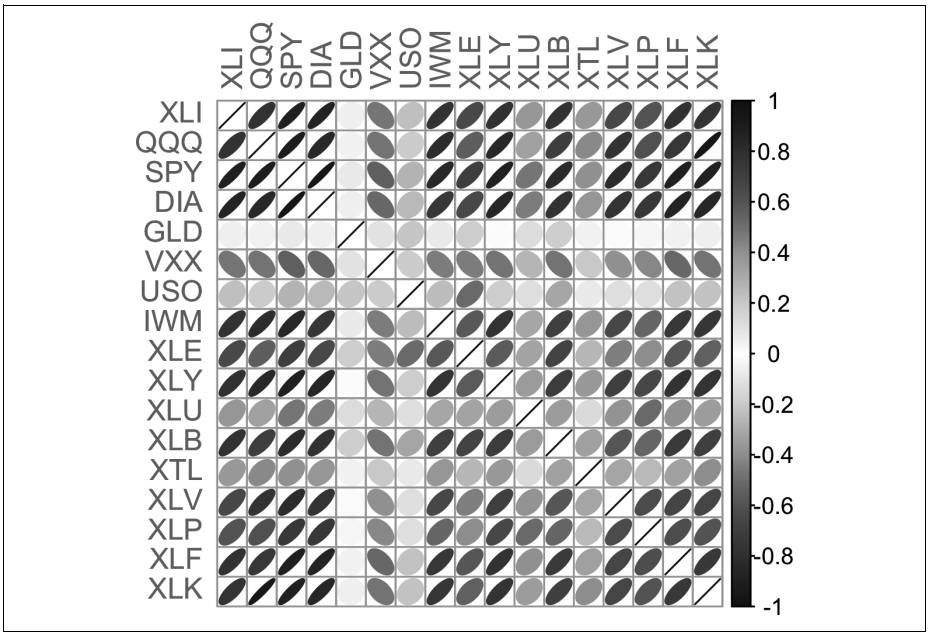

Abbildung 1-6: Die Korrelation zwischen ETF-Renditen

Wie der Mittelwert und die Standardabweichung ist auch der Korrelationskoeffizient empfindlich gegenüber Ausreißern in den Daten. Die Softwarepakete bieten robuste Alternativen zum herkömmlichen Korrelationskoeffizienten. Beispielsweise verwendet das *R*-Paket `robust` (*https://oreil.ly/isORz*) die Funktion `covRob` zur Berechnung eines robusten Korrelationsmaßes. Die Methoden im `scikit-learn`-Modul *sklearn.covariance* (*https://oreil.ly/su7wi*) implementieren eine Vielzahl von Ansätzen.

Weitere Korrelationsmaße

Bereits vor langer Zeit schlugen Statistiker andere Korrelationskoeffizienten vor, wie z.B. das *Spearman'sche Rho* oder das *Kendall'sche Tau*. Diese Korrelationskoeffizienten basieren auf dem Rang der Daten. Da sie mit Rängen und nicht mit den Werten arbeiten, sind diese Maße robust gegenüber Ausreißern und können mit bestimmten Arten von Nichtlinearitäten umgehen. Data Scientists können sich bei explorativen Analysen jedoch im Allgemeinen an dem Korrelationskoeffizienten nach Pearson und seinen robusten Alternativen orientieren. Rangbasierte Maße eignen sich vor allem bei kleineren Datensätzen und bestimmten Hypothesentests.

Streudiagramme

Die standardmäßige Vorgehensweise bei der Visualisierung des Zusammenhangs von zwei beobachteten Variablen ist die Verwendung eines Streudiagramms. Die x-Achse stellt eine Variable und die y-Achse eine andere dar, und jeder Punkt auf

dem Diagramm entspricht einer Beobachtung. In Abbildung 1-7 sehen Sie ein Diagramm, das die Korrelation der Tagesrenditen der Unternehmen ATT und Verizon abbildet. In *R* können Sie das Diagramm mit dem folgenden Befehl erstellen:

```
plot(telecom$T, telecom$VZ, xlab='ATT (T)', ylab='Verizon (VZ)')
```

Das gleiche Diagramm kann in *Python* mit der `pandas`-Methode `scatter` erzeugt werden:

```
ax = telecom.plot.scatter(x='T', y='VZ', figsize=(4, 4), marker='$\u25EF$')
ax.set_xlabel('ATT (T)')
ax.set_ylabel('Verizon (VZ)')
ax.axhline(0, color='grey', lw=1)
ax.axvline(0, color='grey', lw=1)
```

Die Renditen stehen in einer positiven Beziehung: Obwohl sie sich um den Wert null gruppieren, steigen oder sinken die Aktien an den meisten Tagen gleichzeitig (oberer rechter und unterer linker Quadrant). Es gibt weniger Tage, an denen eine Aktie deutlich sinkt, während die andere steigt oder umgekehrt (unterer rechter und oberer linker Quadrant).

Obwohl in dem Diagramm in Abbildung 1-7 nur 754 Datenpunkte angezeigt werden, wird es offenbar schwierig, Details in der Mitte des Diagramms zu erkennen. Wir werden später noch sehen, wie wir die Transparenz der Punkte verändern oder Hexagonal-Binning- sowie Dichtediagramme verwenden können, um weitere Strukturen in den Daten aufzudecken.

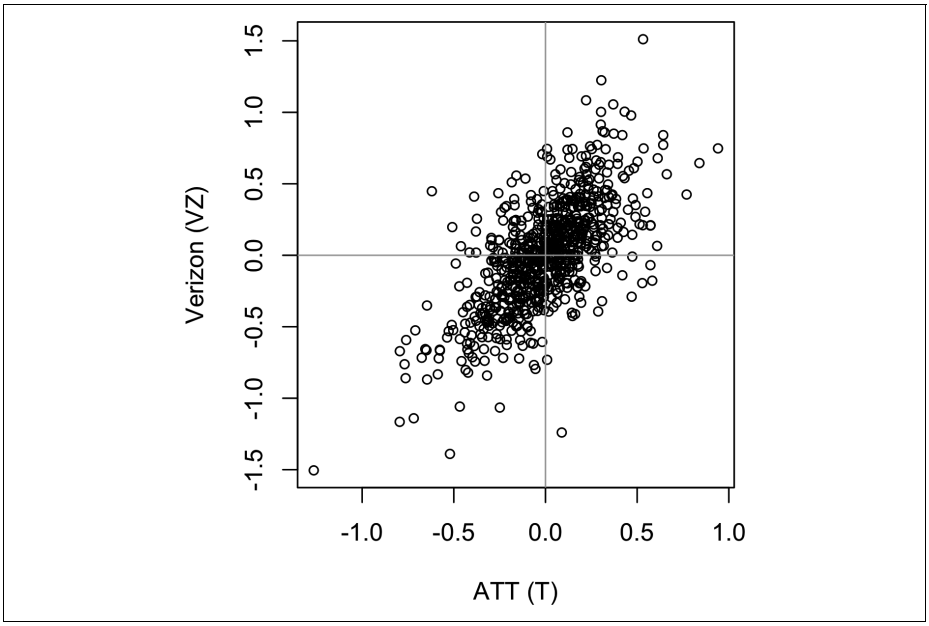

Abbildung 1-7: Streudiagramm zur Darstellung der Korrelation der Tagesrenditen von ATT und Verizon

Kernideen

- Der Korrelationskoeffizient misst, wie stark zwei gepaarte Variablen (z. B. Größe und Gewicht von Individuen) miteinander in Zusammenhang stehen.
- Wenn hohe Werte der einen Variablen mit hohen Werten der anderen einhergehen, stehen sie in einem positiven Zusammenhang.
- Wenn hohe Werte der einen Variablen mit niedrigen Werten der anderen einhergehen, stehen sie in einem negativen Zusammenhang.
- Der Korrelationskoeffizient ist ein standardisiertes Maß, das immer zwischen –1 (perfekte negative Korrelation) und +1 (perfekte positive Korrelation) liegt.
- Ein Korrelationskoeffizient von null bedeutet, dass die Variablen unkorreliert sind. Seien Sie sich dennoch bewusst, dass auch zufällig generierte Daten positive oder negative Werte für den Korrelationskoeffizienten aufweisen können.

Weiterführende Literatur

Eine ausgezeichnete Behandlung des Themas bietet das Buch *Statistics* von David Freedman, Robert Pisani und Roger Purves (4. Auflage, W. W. Norton, 2007).

Zwei oder mehr Variablen untersuchen

Vertraute Maße wie der Mittelwert und die Varianz beziehen sich immer nur eine einzelne Variable (*univariate Analyse*). Die Korrelationsanalyse (siehe »Korrelation« auf Seite 32) ist eine wichtige Methode, um zwei Variablen miteinander zu vergleichen (*bivariate Analyse*). In diesem Abschnitt befassen wir uns weiterhin mit Maßen und Diagrammen, und zwar insbesondere mit solchen, die auf die Exploration von mehr als zwei Variablen abzielen (*multivariate Analyse*).

Schlüsselbegriffe zur Exploration von zwei oder mehr Variablen

Kontingenztafeln
: Eine Kreuztabelle mit den Häufigkeiten von zwei oder mehr kategorialen Variablen (engl. *Contingency Table*).

Hexagonal-Binning-Diagramm
: Ein Diagramm für zwei numerische Variablen, wobei die Beobachtungen in Sechsecken zusammengefasst sind.

Konturdiagramm
: Ein Diagramm, bei dem die Wahrscheinlichkeitsdichtefunktion zweier numerischer Variablen wie eine topografische Karte abgebildet wird (engl. *Contour Plot*).

Violin-Plot
: Ähnlich wie ein Box-Plot, zeigt allerdings darüber hinaus noch die geschätzte Dichtefunktion.

Wie bei der univariaten Analyse möchten wir auch bei der bivariaten Analyse einerseits zusammenfassende statistische Kenngrößen ermitteln und andererseits anschauliche Visualisierungen erstellen. Die geeignete Art der bi- bzw. multivariaten Analyse hängt vom Datentyp ab – je nachdem, ob die Daten als numerische oder kategoriale Variablen vorliegen.

Hexagonal-Binning- und Konturdiagramme (Diagramme für mehrere numerische Variablen)

Streudiagramme sind durchaus geeignet, solange die Anzahl an Datenpunkten relativ gering ist. Das Diagramm mit den Aktienrenditen in Abbildung 1-7 bildet nur etwa 750 Datenpunkte ab. Bei Datensätzen, die Hunderttausende oder Millionen von Datenpunkten haben, erweist sich ein Streudiagramm als zu stark verdichtend, weshalb wir ein anderes Verfahren benötigen, um den Zusammenhang adäquat visualisieren zu können. Betrachten wir zur Veranschaulichung den Datensatz `kc_tax`, der die Steuerbemessungswerte von Wohnimmobilien in King County, Washington, wiedergibt. Um uns auf den wesentlichen Teil der Daten zu konzentrieren, entfernen wir mit der Funktion `subset` zunächst Beobachtungen für sehr teure und sehr kleine sowie auch große Wohnungen:

```
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 &
                  SqFtTotLiving > 100 &
                  SqFtTotLiving < 3500)
nrow(kc_tax0)
432693
```

In `pandas` filtern wir den Datensatz wie folgt:

```
kc_tax0 = kc_tax.loc[(kc_tax.TaxAssessedValue < 750000) &
                     (kc_tax.SqFtTotLiving > 100) &
                     (kc_tax.SqFtTotLiving < 3500), :]
kc_tax0.shape
(432693, 3)
```

Abbildung 1-8 zeigt ein *Hexagonal-Binning-Diagramm*, das die Beziehung zwischen der fertiggestellten Wohnfläche und dem steuerlich geschätzten Wert von Immobilien in King County abbildet. Anstatt einfach die einzelnen Datenpunkte abzubilden, die sich als dunkle, zusammenhängende Wolke darstellen würden, sind sie in sechseckige Felder gruppiert, die so eingefärbt werden, dass die Anzahl der Beobachtungen in diesem Feld widergespiegelt wird. In diesem Diagramm ist der positive Zusammenhang zwischen der Anzahl der fertiggestellten Wohnfläche und dem steuerlich geschätzten Wert deutlich zu erkennen. Ein interessantes De-

tail ist, dass wir zusätzliche Verdunklungen über dem (dunkelsten) unteren Hauptbereich erkennen können, die auf Häuser schließen lassen, die zwar jenen im Hauptbereich flächenmäßig gleichen, aber einen höheren steuerlich ermittelten Wert aufweisen.

Abbildung 1-8 wurde mit dem beeindruckenden *R*-Paket `ggplot2` erzeugt, das von Hadley Wickham entwickelt wurde [ggplot2]. `ggplot2` ist eines von mehreren modernen Softwarepaketen, die zur anspruchsvollen explorativen visuellen Datenanalyse entwickelt wurden (siehe »Mehrere Variablen visualisieren« auf Seite 44):

```
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(color='white') +
  theme_bw() +
  scale_fill_gradient(low='white', high='black') +
  labs(x='Fertiggestellte Wohnfläche (in Quadratfuß)',
       y='Steuerlich geschätzter Wert')
```

In *Python* können Hexagonal-Binning-Diagramme mit der `pandas`-Data-Frame-Methode `hexbin` leicht erstellt werden:

```
ax = kc_tax0.plot.hexbin(x='SqFtTotLiving', y='TaxAssessedValue',
                         gridsize=30, sharex=False, figsize=(5, 4))
ax.set_xlabel('Fertiggestellte Wohnfläche (in Quadratfuß)')
ax.set_ylabel('Steuerlich geschätzter Wert')
```

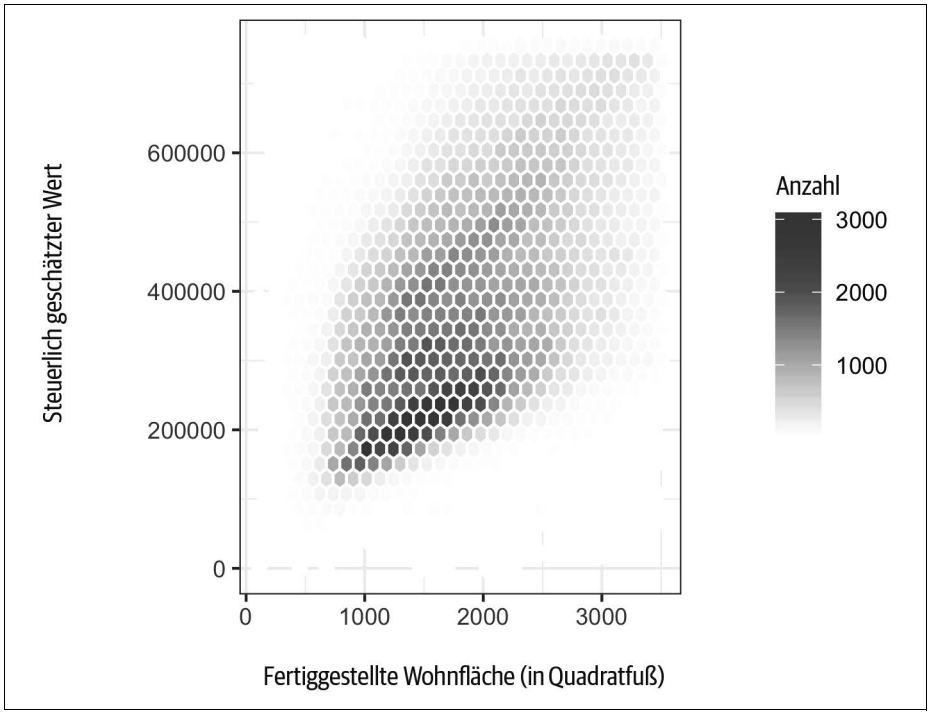

Abbildung 1-8: Hexagonal-Binning-Diagramm zur Darstellung des Zusammenhangs zwischen dem steuerlich geschätzten Wert und der fertiggestellten Wohnfläche von Immobilien

In Abbildung 1-9 werden Konturen (Niveau- bzw. Höhenlinien) verwendet, die einem Streudiagramm überlagert sind, um die Beziehung zwischen zwei numerischen Variablen zu visualisieren. Die Höhenlinien bilden im Wesentlichen eine topografische Karte für zwei Variablen ab; jedes Höhenlinienband stellt eine bestimmte Dichte von Punkten dar, die zunimmt, wenn man sich dem »Peak« nähert. Dieses Diagramm offenbart einen ähnlichen Zusammenhang wie das in Abbildung 1-8: Es gibt einen zweiten Peak »nördlich« des Hauptpeaks. Dieses Diagramm wurde ebenfalls mit dem `ggplot2`-Paket mit der integrierten Funktion `geom_density2d` erstellt:

```
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point(alpha=0.1) +
  geom_density2d(color='white') +
  labs(x='Fertiggestellte Wohnfläche (in Quadratfuß)',
       y='Steuerlich geschätzter Wert')
```

In *Python* können Sie ein Konturdiagramm mit der `seaborn`-Funktion `kdeplot` erzeugen:

```
ax = sns.kdeplot(kc_tax0.SqFtTotLiving, kc_tax0.TaxAssessedValue, ax=ax)
ax.set_xlabel('Fertiggestellte Wohnfläche (in Quadratfuß)')
ax.set_ylabel('Steuerlich geschätzter Wert')
```

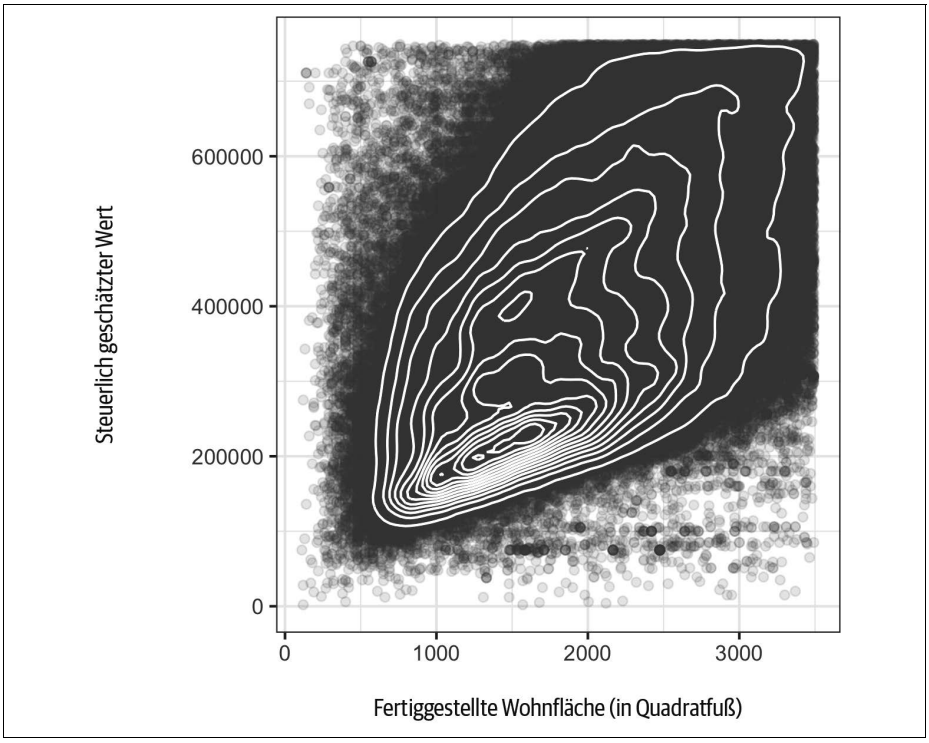

Abbildung 1-9: Konturdiagramm zur Darstellung des Zusammenhangs zwischen dem steuerlich geschätzten Wert und der fertiggestellten Wohnfläche von Immobilien

Es gibt noch weitere Diagrammtypen, die die Beziehung zwischen zwei numerischen Variablen aufzeigen können, unter anderem *Heatmaps*. Heatmaps, Hexagonal-Binning- und Konturdiagramme bilden alle eine zweidimensionale Dichtefunktion ab. Daher sind sie die natürlichen Analoga zu Histogrammen und Dichtediagrammen.

Zwei kategoriale Variablen

Einen gebräuchlichen Ansatz, den Zusammenhang zweier kategorialer Variablen zusammenzufassen, bietet die Kontingenztafel – eine Tabelle, die die absoluten Häufigkeiten der verschiedenen Kombinationen der Kategorien (bzw. der Ausprägungen der beiden kategorialen Variablen) angibt. Die Kontingenztabelle in Tabelle 1-8 zeigt den Zusammenhang von zwei kategorialen Daten, zum einen solche, die das Rating eines Privatkredits angeben, und zum anderen solche, die das Ergebnis bzw. den Erfolg hinsichtlich der Rückzahlung dieses Kredits wiedergeben. Sie basiert auf Daten, die vom Lending Club, einem führenden Unternehmen im Peer-to-Peer-Kreditgeschäft, zur Verfügung gestellt wurden. Das Rating bzw. die Einstufung hat die Ausprägungen A (hoch) bis G (niedrig). Die Ausprägung des Ergebnisses der Kreditrückzahlung ist entweder »vollständig bezahlt«, »zurzeit in Rückzahlung«, »verspätet zurückgezahlt« oder »abgeschrieben« (also ein Kreditausfall – der Restbetrag des Darlehens wird voraussichtlich nicht eingetrieben). Die Tabelle zeigt sowohl die absolute als auch – in Bezug auf die jeweiligen Zeilen – die relative Häufigkeit. Besser eingestufte Kredite weisen im Vergleich zu schlechter eingestuften Krediten einen sehr niedrigen Prozentsatz an verspäteten bzw. abgeschriebenen Rückzahlungen auf.

Tabelle 1-8: Kontingenztabelle der Krediteinstufung in Bezug auf den tatsächlichen Status

Einstufung	abgeschrieben	zurzeit in Rückzahlung	vollständig zurückgezahlt	verspätet	Total
A	1562	50051	20408	469	72490
	0.022	0.690	0.282	0.006	0.161
B	5302	93852	31160	2056	132370
	0.040	0.709	0.235	0.016	0.294
C	6023	88928	23147	2777	120875
	0.050	0.736	0.191	0.023	0.268
D	5007	53281	13681	2308	74277
	0.067	0.717	0.184	0.031	0.165
E	2842	24639	5949	1374	34804
	0.082	0.708	0.171	0.039	0.077
F	1526	8444	2328	606	12904
	0.118	0.654	0.180	0.047	0.029
G	409	1990	643	199	3241
	0.126	0.614	0.198	0.061	0.007
Total	22671	321185	97316	9789	450961

Zusätzlich zu den absoluten Häufigkeiten können bei Kontingenztafeln auch relative Häufigkeiten für die Spalten- und Gesamtwerte dargestellt werden. Die wahrscheinlich gebräuchlichste Nutzung von Kontingenztafeln sind Pivot-Tabellen in Excel. In *R* können Kontingenztafeln mithilfe der Funktion `CrossTable` des Pakets `descr` erzeugt werden; der folgende Code wurde zur Erstellung von Tabelle 1-8 verwendet:

```
library(descr)
x_tab <- CrossTable(lc_loans$grade, lc_loans$status,
                    prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
```

In *Python* können wir die Pivot-Tabelle mit der Methode `pivot_table` erzeugen. Das Argument `aggfunc` erlaubt uns, die absoluten Häufigkeiten zu ermitteln. Die Berechnung der relativen Häufigkeiten gestaltet sich etwas aufwendiger:

```
crosstab = lc_loans.pivot_table(index='grade', columns='status',
                                aggfunc=lambda x: len(x), margins=True) ❶

df = crosstab.loc['A':'G',:].copy() ❷
df.loc[:,'Charged Off':'Late'] = df.loc[:,'Charged Off':'Late'].div(df['All'],
                                                                    axis=0) ❸
df['All'] = df['All'] / sum(df['All']) ❹
perc_crosstab = df
```

❶ Mit dem Schlüsselwortargument `margins` werden die Werte spalten- und zeilenweise summiert.

❷ Wir erstellen eine Kopie der Pivot-Tabelle und lassen die ermittelten Gesamtsummen der Spalten außen vor.

❸ Wir teilen die Werte der einzelnen Zeilen durch die Gesamtsumme der jeweiligen Zeile.

❹ Wir dividieren die Spalte `'All'` durch die Gesamtsumme.

Kategoriale und numerische Variablen

Mit Box-Plots (siehe »Perzentile und Box-Plots« auf Seite 21) lassen sich Verteilungen numerischer Variablen, die anhand einer kategorialen Variablen gruppiert wurden, auf einfache Weise visuell vergleichen. Beispielsweise könnten wir vergleichen, wie der prozentuale Anteil der Flugverspätungen für die einzelnen Fluggesellschaften variiert. Abbildung 1-10 zeigt den prozentualen Anteil der Flüge der verschiedenen Fluggesellschaften, die innerhalb eines Monats verspätet waren und bei denen die Verspätung auf die Fluggesellschaft selbst zurückzuführen ist:

```
boxplot(pct_carrier_delay ~ airline, data=airline_stats, ylim=c(0, 50))
```

Die `pandas-boxplot`-Methode nimmt das Argument `by` entgegen, wodurch der Datensatz in verschiedene Gruppen aufgeteilt wird und sich die einzelnen Box-Plots erstellen lassen:

```
ax = airline_stats.boxplot(by='airline', column='pct_carrier_delay')
ax.set_xlabel('')
ax.set_ylabel('Tägliche Flugverspätungen (in %)')
plt.suptitle('')
```

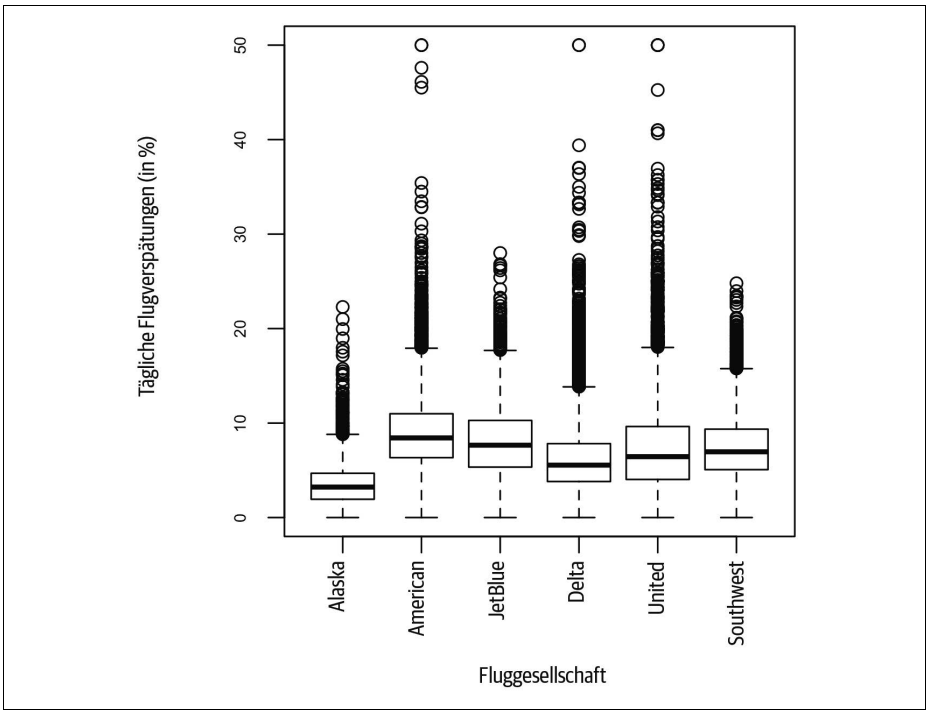

Abbildung 1-10: Ein Box-Plot des prozentualen Anteils der Flüge, die verspätet waren und bei denen die Verspätung auf die Fluggesellschaft zurückzuführen ist

Die Fluggesellschaft Alaska Airlines hat die geringsten Verspätungen, bei American Airlines gab es die größten: Das untere Quartil für American Airlines ist höher als das obere Quartil für Alaska Airlines.

Der *Violin-Plot*, eingeführt von [Hintze-Nelson-1998], ist eine Erweiterung des Box-Plots und bildet die geschätzte Dichtefunktion ab, wobei die Wahrscheinlichkeitsdichte auf der y-Achse abgetragen wird. Die Dichtefunktion wird zusätzlich gespiegelt und die resultierende Silhouette ausgefüllt. Dadurch entsteht eine Grafik, die einer Violine ähnelt. Der Vorteil des Violin-Plots ist, dass er besser als der Box-Plot in der Lage ist, Nuancen in der Verteilung aufzuzeigen. Andererseits werden im Box-Plot die Ausreißer in den Daten deutlicher hervorgehoben. Im Paket `ggplot2` steht die Funktion `geom_violin` bereit, um einen Violin-Plot zu erstellen:

```
ggplot(data=airline_stats, aes(airline, pct_carrier_delay)) +
  ylim(0, 50) +
  geom_violin() +
  labs(x='', y='Tägliche Flugverspätungen (in %)')
```

Im seaborn-Paket können Violin-Plots durch Nutzung der violinplot-Methode erstellt werden:

```
ax = sns.violinplot(airline_stats.airline, airline_stats.pct_carrier_delay,
                    inner='quartile', color='white')
ax.set_xlabel('')
ax.set_ylabel('Tägliche Flugverspätungen (in %)')
```

Das entsprechende Diagramm wird in Abbildung 1-11 dargestellt. Der Violin-Plot zeigt eine Ballung in der Verteilung nahe null für Alaska Airlines und, in geringerem Ausmaß, für Delta Airlines. Dieses Merkmal ist im Box-Plot nicht so offensichtlich. Sie können auch einen Violin-Plot mit einem Box-Plot kombinieren, indem Sie die Funktion geom_boxplot (in *R*) als zusätzliche Komponente dem Diagramm hinzufügen (obwohl dies am besten funktioniert, wenn Farben verwendet werden).

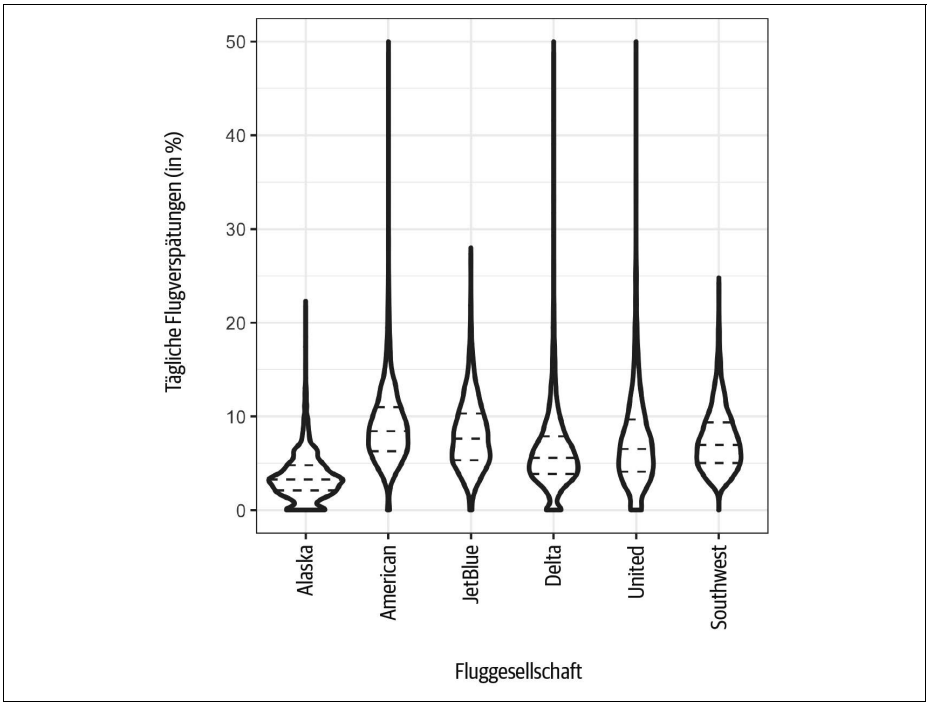

Abbildung 1-11: Ein Violin-Plot des prozentualen Anteils der Flüge, die verspätet waren und bei denen die Verspätung auf die Fluggesellschaft selbst zurückzuführen ist

Mehrere Variablen visualisieren

Die Diagrammtypen, die für den Vergleich zweier Variablen verwendet werden – Streu-, Hexagonal-Binning-Diagramme und Box-Plots –, lassen sich allesamt mithilfe des Ansatzes der *Konditionierung* leicht auf weitere Variablen erweitern. Ein Beispiel hierfür war in Abbildung 1-8 zu sehen, hier wurde die Beziehung zwischen

der fertiggestellten Wohnfläche von Immobilien (in Quadratfuß) und ihrem steuerlich geschätzten Wert aufgezeigt. Wir hatten festgestellt, dass es eine Häufung von Immobilien zu geben scheint, die einen höheren steuerlich geschätzten Wert pro Quadratfuß haben. Wenn wir die Daten weiter unterteilen und wie in Abbildung 1-12 zusätzlich die Postleitzahl der jeweiligen Immobilie als Drittvariable berücksichtigen, können wir den Effekt der Lage der Immobilie aufdecken. Jetzt ist das Bild viel klarer: Der steuerlich geschätzte Wert ist bei einigen Postleitzahlen (98105, 98126) viel höher als bei anderen (98108, 98188). Diese Diskrepanz erklärt auch die in Abbildung 1-8 beobachtete Clusterbildung.

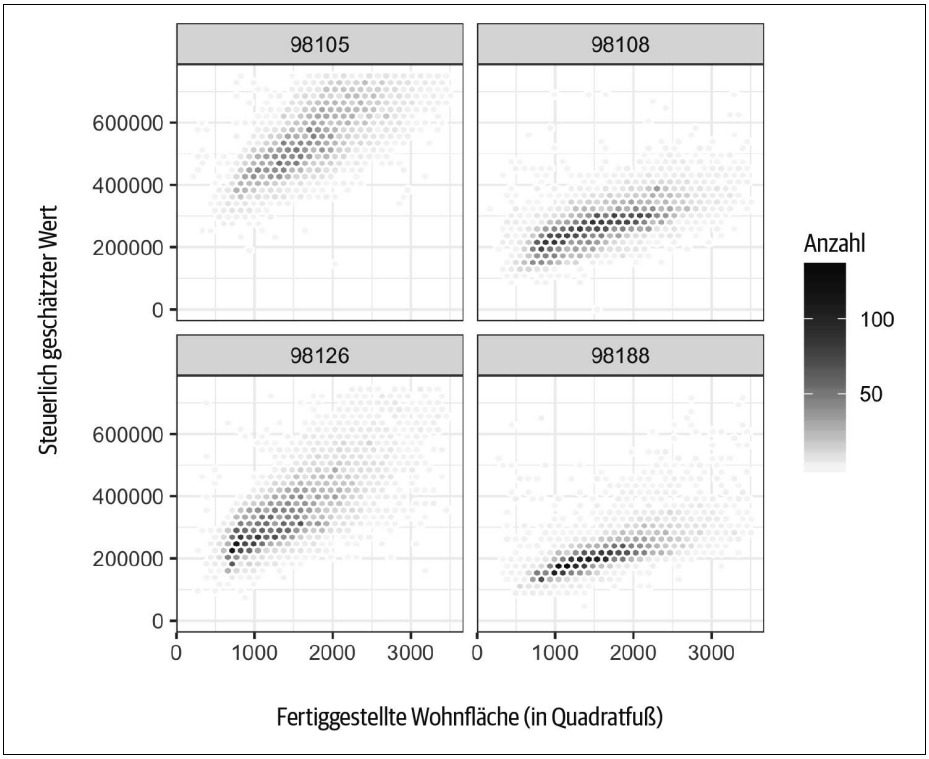

Abbildung 1-12: Steuerlich geschätzter Immobilienwert in Abhängigkeit von der fertiggestellten Wohnfläche (in Quadratfuß), nach Postleitzahlen unterteilt

Abbildung 1-12 haben wir mithilfe des `ggplot2`-Pakets unter Verwendung von *facets*, d. h. mit einer Drittvariablen (in diesem Fall der Postleitzahl), erstellt:

```
ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
         aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(color='white') +
  theme_bw() +
  scale_fill_gradient(low='white', high='blue') +
  labs(x='Fertiggestellte Wohnfläche (in Quadratfuß)',
       y='Steuerlich geschätzter Wert') +
  facet_wrap('ZipCode') ❶
```

❶ Verwenden Sie die ggplot-Funktionen facet_wrap bzw. facet_grid, um die Drittvariable zu spezifizieren.

Die meisten *Python*-Pakete stützen sich für ihre Visualisierungen auf die Bibliothek matplotlib. Obwohl es prinzipiell auch mit der matplotlib-Bibliothek möglich ist, differenziertere Darstellungen zu erstellen, kann der Code schnell komplex werden. Glücklicherweise bietet seaborn eine relativ einfache Möglichkeit, diese Diagramme zu erzeugen:

```
zip_codes = [98188, 98105, 98108, 98126]
kc_tax_zip = kc_tax0.loc[kc_tax0.ZipCode.isin(zip_codes),:]
kc_tax_zip

def hexbin(x, y, color, **kwargs):
    cmap = sns.light_palette(color, as_cmap=True)
    plt.hexbin(x, y, gridsize=25, cmap=cmap, **kwargs)

g = sns.FacetGrid(kc_tax_zip, col='ZipCode', col_wrap=2) ❶
g.map(hexbin, 'SqFtTotLiving', 'TaxAssessedValue',
      extent=[0, 3500, 0, 700000]) ❷
g.set_axis_labels('Fertiggestellte Wohnfläche (in Quadratfuß)',
                  'Steuerlich geschätzter Wert')
g.set_titles('Postleitzahl {col_name:.0f}')
```

❶ Verwenden Sie die Argumente col und row, um die Drittvariable anzugeben. Für eine einzelne Drittvariable können Sie das Argument col zusammen mit col_wrap nutzen, um das Facettendiagramm in mehrere Quadranten aufzuteilen.

❷ Mit der Methode map wird die Funktion hexbin auf die hinsichtlich der verschiedenen Postleitzahlen untergliederten Teilmengen des ursprünglichen Datensatzes angewandt. Durch die Angabe von extent definieren Sie, wie weit sich die x- und y-Achsen erstrecken sollen.

Das Konzept der Konditionierung von Variablen in grafischen Darstellungen wurde mit *Trellis-Grafiken*, die von Rick Becker, Bill Cleveland und anderen bei Bell Labs entwickelt wurden, eingeführt [Trellis-Graphics]. Diese Idee hat sich auf verschiedene moderne Visualisierungsprogramme übertragen, wie z. B. dem lattice- [lattice] und dem ggplot2-Paket in *R* und den seaborn- [seaborn] und Bokeh-Modulen [bokeh] in *Python*. Drittvariablen stellen ebenfalls einen integralen Bestandteil von Business-Intelligence-Plattformen wie Tableau und Spotfire dar. Mit dem Aufkommen enormer Rechenleistung haben moderne Visualisierungsplattformen die bescheidenen Anfänge der explorativen Datenanalyse weit hinter sich gelassen. Die Schlüsselkonzepte und Werkzeuge, die vor einem halben Jahrhundert entwickelt wurden (z. B. einfache Box-Plots), bilden jedoch immer noch eine Grundlage solcher Systeme.

Kernideen

- Hexagonal-Binning- und Konturdiagramme sind nützliche Werkzeuge, die eine gleichzeitige visuelle Exploration zweier numerischer Variablen ermöglichen, ohne von riesigen Datenmengen überwältigt zu werden.
- Kontingenztabellen sind das gängigste Werkzeug, um die Häufigkeiten von zwei kategorialen Variablen zu betrachten.
- Box-Plots und Violin-Plots ermöglichen Ihnen, den Zusammenhang zwischen einer numerischen Variablen und einer kategorialen Variablen darzustellen.

Weiterführende Literatur

- Das Buch *Modern Data Science with R* von Benjamin Baumer, Daniel Kaplan und Nicholas Horton (Chapman & Hall/CRC Press, 2017) bietet eine ausgezeichnete Präsentation von »einer Grammatik für Grafiken« (das »gg« in `ggplot`).
- Ein weiteres, vom Entwickler des `ggplot2`-Pakets geschriebenes Buch mit dem Titel *ggplot2: Elegant Graphics for Data Analysis* von Hadley Wickham (Springer, 2009) ist ebenfalls eine ausgezeichnete Ressource.
- Josef Fruehwald hat eine webbasierte Anleitung für das `ggplot2`-Paket (*https://oreil.ly/zB2Dz*) bereitgestellt.

Zusammenfassung

Die von John Tukey begründete explorative Datenanalyse (EDA) schuf den Grundstein für unser heutiges Verständnis der Data Science. Der Kerngedanke der EDA ist, dass der erste und wichtigste Schritt in jedem Projekt, bei dem mit Daten gearbeitet wird, darin liegt, sich die Daten anzusehen. Durch die Zusammenfassung und Visualisierung der Daten können Sie wertvolle Erkenntnisse für das Projekt gewinnen.

In diesem Kapitel wurden mehrere Konzepte vorgestellt, die von einfachen statistischen Maßzahlen, z. B. Lage- und Streuungsmaßen, bis hin zu aussagekräftigen visuellen Darstellungen reichen, die die Beziehungen zwischen mehreren Variablen, wie in Abbildung 1-12, untersuchen. Die vielfältigen Werkzeuge und Verfahren, die von der Open-Source-Gemeinschaft entwickelt wurden (und werden), haben in Verbindung mit der Ausdruckskraft der Programmiersprachen *R* und *Python* eine Fülle von Möglichkeiten zur Exploration und zur Analyse von Daten geschaffen. Die explorative Datenanalyse sollte ein Grundpfeiler jedes datenwissenschaftlichen Projekts sein.