Einführung in R

Bericht

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

Vorgelegt von:	Michael Zwinkel
E-Mail-Adresse:	Michael.Zwinkel@studium.uni-hamburg.de
Matrikelnummer:	6695268
Studiengang:	Informatik
Betreuer:	Julian Kunkel

Hamburg, den 30.09.2016

Abstract

In diesem Bericht geht es darum, dem Leser durch eine Einführung einen Eindruck von R zu vermitteln. Jedes Kapitel hat am Anfang eine kurze Einführung, die beschreibt worum es in dem Kapitel geht.
In der Einführung wird kurz erläutert was R ist und wo es Anwendung findet. Darauf folgt eine kurze Entstehungsgeschichte von R, sowie eine Gegenüberstellung von Vor- und Nachteilen die R mit sich bringt.
Im darauf folgendem Abschnitt werden einige Funktionen von R erklärt, sowie die Weise wie R diese ausgibt. Es wird angefangen mit einfachen mathematischen Formeln, und endet mit einem fiktiven Beispiel, das zeigt wie man in R Datensätze einliest und aus diesen dann statistische Grafiken erstellt.

Contents

1 Einführung

In diesem Kapitel geht es darum, kurz zu erläutern was R ist und wozu es genutzt wird. Die Entstehungsgeschichte von R wird hier ebenfalls dargelegt, gefolgt von Vor- und Nachteilen von R.

R ist eine frei verfügbare Programmiersprache, die der GNU General Public License unterliegt. R gilt als eine General Purpose Language, was bedeutet, dass R nicht nur für bestimmte Anwendungsfälle einsetzbar ist, sondern allgemein für die meisten Aufgaben genutzt werden kann. In erster Linie wird R zur Untersuchung statistischer Probleme genutzt, was in der Wissenschaft, als auch in der Wirtschaft Anwendung findet. Ein besonders wichtiger Punkt von R ist die Erstellung statistischer Grafiken.

1.1 Geschichte von R

1992 wurde R von Ross Ihaka und Robert gentleman entwickelt. R wurde in den Programmiersprachen C (52%), Fortran (26%) und als der Fortschritt groß genug war sogar in R selbst (22%) programmiert. 1993 begann die öffentliche Verbreitung von R. R wurde separat Personen vorgestellt, die bereits versiert in der Programmiersprache S waren, um mögliche Schwächen zu erkennen, hilfreiches Feedback zu erhalten und R bereits in wichtigen Kreisen bekannt zu machen. Seit 1995 steht R unter der GNU General Public License und war zu dem Zeitpunkt die einzige Statistikumgebung die für Linux verfügbar war. 1997 wurde das "R Development Core Team" gegründet, dessen Aufgabe es war R weiter zu entwickeln. Im selben Jahr startete auch CRAN, das Comprehensive R Archive Network, welches als Plattform fungierte um Anwendern das Teilen von selbst geschriebenen Funktionen mit anderen Nutzern zu ermöglichen. Ebenfalls in 1997 wurden die Alpha-Versionen von R für Windows- und Mac-Systeme veröffentlicht. Im Jahr 2000 wurde R als stabil genug betrachtet um es als Version 1.0 zu veröffentlichen. Seit 2001 existiert eine stabile R-version für Mac Betriebssysteme. 2002 wurde die R Foundation for statistical computing gegründet, die für die Außendarstellung von R verantwortlich ist. 2004 wurde R Version 2.0 veröffentlicht, welche nun Lazy Loading unterstützt, das bedeutet dass Werte oder Objekte grundsätzlich bereitgestellt werden, aber erst bei einer konkreten Anfrage auf den Wert, bzw das Objekt, aus der Datenquelle geholt werden. 2005 erschien R Version 2.1. Seit dieser Version unterstützt R verschiedene Sprachversionen, sowie Zeichenkodierungen, von denen besonders UTF-8 sehr wichtig ist. 2010 wurde R Version 2.11 veröffentlicht, welche es nun möglich machte R auf 64-Bit Systemen zu nutzen, sowie bis zu acht terabyte Arbeitsspeicher zu adressieren. 2013 ist das Jahr in dem Version 3.0 veröffentlicht wurde. Hier war

es nun zum ersten Mal möglich Indexwerte von 2^{31} und größer auf 64-Bit Systemen darzustellen. 2015 wurde das "R Consortium" von Unternehmen gegründet die R selbst nutzen, um R im Unternehmensfeld komfortabler einsetzen zu können. Zum jetzigen Zeitpunkt (26.09.2016) ist die neueste R Version 3.3.1 und heißt "Bug in your Hair"

1.2 Vor- Und Nachteile von R

In diesem Kapitel werden die Vor- und Nachteile von R direkt gegenübergestellt um einen Eindruck davon zu vermitteln worin die Stärken und Schwächen dieser Programmiersprache liegen.

Vorteile von R

- Open-Source [1]
- Schnelle Entwicklung [1]
- Keine Lizenzgebühren [1]
- Neue statististische Methoden werden sofort in R umgesetzt [1]
- Probleme werden durch große Nutzergemeinde schnell gelöst
- R läuft auf allen gängigen Betriebssystemen

Nachteile von R

- Keine komplett grafische Oberfläche [1]
- Fehlermeldungen helfen oft nicht weiter [1]
- Funktion von älteren Versionen steht nicht im Vordergrund [1]
- Qualität der Pakete sollte für den Zweck überprüft werden [1]

[1] Christian Heumann, Vorlesung Programmieren in statistischer Software: R

2 Funktionen von R

In diesem Kapitel geht es darum die Funktionen von R zu zeigen. Zuerst wird gezeigt wie R mit mathematische Funktionen umgeht. Die Erstellung von statistischen Grafiken in R wird danach beschrieben.

2.1 Mathematische Funktionen

R ist in der Lage mathematische Rechnungen durchzuführen und mathematische Regeln wie zum Beispiel "Punkt vor Strich" selbstständig durchzuführen. Wenn man eine Rechnung wie zum Beispiel "2+2" in R eingibt wird R eine 4 in der Konsole ausgeben.

```
2+2
2-1
2*2
8/2
2^3
2+3*3*(1+1)
```

Dieser Ausschnitt aus R wird in der Konsole also folgendes ausgeben:
4
1
4
4
8
20
Natürlich ist R auch in der Lage andere mathematische Funktionen zu nutzen, wie zum Beispiel die Quadratwurzel einer bestimmten Zahl zu ziehen, und das Ergebnis dann in der Konsole auszugeben. Im folgenden Beispiel, wieder direkt aus R, werden verschiedene Funktionen gezeigt, die auf die Zahl 17 angewendet werden:

```
sqrt(17)#Quadratwurzel
log(17) #Logarithmus
sin(17) #Sinus
cos(17) #Kosinus
tan(17) #Tangens
```

Anders als bei dem Beispiel oben werden in der Konsole nur die ersten 7 Ziffern des Ergebnisses gezeigt. Dies ist die Standardeinstellung von R.

Am Beispiel der Quadratwurzel wird R in der Konsole also folgendes anzeigen:
4,123105
Wenn man jetzt also ein genaueres Ergebnis zur Quadratwurzel von 17 haben möchte, dann muss man dies R mit einem Befehl mitteilen. Folgende Beispiele um dies zu tun sind wieder direkt aus R:

```
1 options(digits=22)
2 #Aendert die Anzahl der Ziffern
```

Dieser Befehl ändert die Anzahl der Ziffern die R in der Konsole darstellt zu 22. Dies gilt für alle Ziffern die von nun an dargestellt werden.

```
1 print(sqrt(17), digits=22)
2 #Aendert die Anzahl der Ziffern der Funktion sqrt(17)
```

Dieser Befehl ändert die Anzahl der Ziffern die von R in der Konsole dargestellt werden zu 22. Dies gilt allerdings nur für das Ergebnis von Quadratwurzel von 17.

In beiden Beispielen wird R in der Konsole folgendes ausgeben:
4,12310562562
In diesen Beispielen wurde 22 als Anzahl der Ziffern ausgewählt, weil 22 die maximale Anzahl an Ziffern ist, die R darstellen kann. Dies gilt nur für Ziffern die durch eine Rechnung ausgegeben werden, manuell lassen sich natürlich mehr als 22 Ziffern eingeben und werden auch korrekt in der Konsole ausgegeben.

2.2 Statistische Grafiken

Eine wichtige Funktion von R ist die Erstellung von Grafiken im Bereich der Statistik. R wird in der Forschung, sowie im kommerziellen Bereich vorrangig dazu verwendet um Daten für Statistiken in Grafiken darzustellen. R ist in der Lage verschiedene Typen von statistischen Grafiken darzustellen, wie zum Beispiel Säulendiagramme, Kuchendiagramme und Kastendiagramme. Wie diese Diagramme aussehen und erstellt werden wird im folgendem Abschnitt anhand eines Beispiels gezeigt.
In unserem Beispiel geht es um ein fiktives Rennen. Aus diesem Rennen wurden folgende Daten über die Teilnehmer des Rennens festgelegt:

- Startnummer
- Laufzeit
- Geschlecht
- Alter
- Einkommen

Der erste Schritt ist nun unseren Datensatz in R einzulesen. Es ist zwar auch möglich die Daten manuell einzugeben, dies ist bei langen Datensätzen jedoch nicht zu empfehlen. Die Art wie der Datensatz aufgebaut ist bestimmt, wie er korrekt in R eingelesen wird. In unserem Beispiel haben wir eine .txt Datei, die Daten sind mit einem Leerzeichen getrennt, Dezimalzahlen mit einem Punkt.

```
1 dat=read.table("/home/user/Documents/r/rennen.txt",
2 header=T)
```

Wegen des Formats unserer Daten benutzen wir den Befehl "dat=read.table(Quelle)".[1] Andere Formate benutzen unterschiedliche Befehle um die Daten einzulesen. Wenn die Daten zum Beispiel durch ein Komma, und Dezimalzahlen durch einen Punkt getrennt werden, wird der Befehl "read.csv(Quelle)" verwendet. Jetzt sollte man sich überlegen was man mit seiner Grafik zeigen möchte und welches Diagramm sich dazu eignet.
Hier ein Beispiel für ein schlecht gewähltes Diagramm:

```
1 barplot(Einkommen, width=1, space = NULL, names.arg =
    ↪ Geschlecht)
```

Dieser Befehl erstellt ein Säulendiagramm (engl: barplot). Für die Breite der Säulen haben wir den Wert 1 gewählt, für den Abstand der einzelnen Säulen zueinander den Wert 0. header=T bedeutet, dass die Bezeichnungen für die Variablen in der ersten Zeile der .txt-Datei stehen. Die Daten die für dieses Diagramm gewählt wurden sind "Einkommen" und "Geschlecht". Der Befehl "names.arg=geschlecht" sorgt dafür, dass jede Säule mit dem jeweiligen Geschlecht des Teilnehmers beschriftet wird. Die daraus erstellte Grafik sieht so aus:

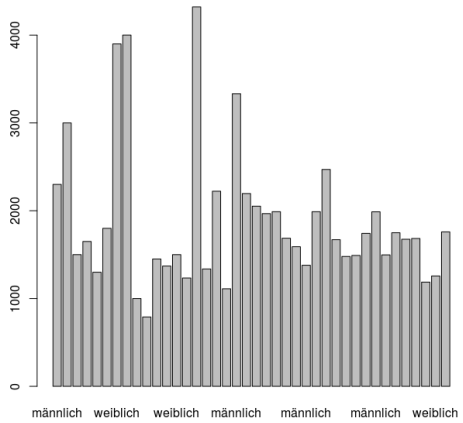

Figure 2.1

[1] 'dat' ist eine Variable. Diese wird nur benutzt, damit nicht bei jedem Ausführen der gesamte Datensatz in der Konsole erscheint.

Man kann also leicht sehen, wieso diese Grafik schlecht gewählt wurde: Es sind so viele Balken, dass die Beschriftung der einzelnen Balken nicht mehr möglich ist. Das Wort "Männlich", oder "Weiblich" ist schon breiter als ein Balken, dies macht es unmöglich zu sagen welcher Balken für welches Geschlecht steht.
Besser für diese Gegenüberstellung eignet sich zum Beispiel ein Kastendiagramm:

```
1 boxplot(Einkommen~Geschlecht)
```

Mit diesem Befehl wird ein Kastendiagramm (engl.: boxplot) erstellt. Die Daten werden einfach hinter den Befehl geschrieben und durch eine Tilde getrennt.
Die daraus erstellte Grafik sieht so aus:

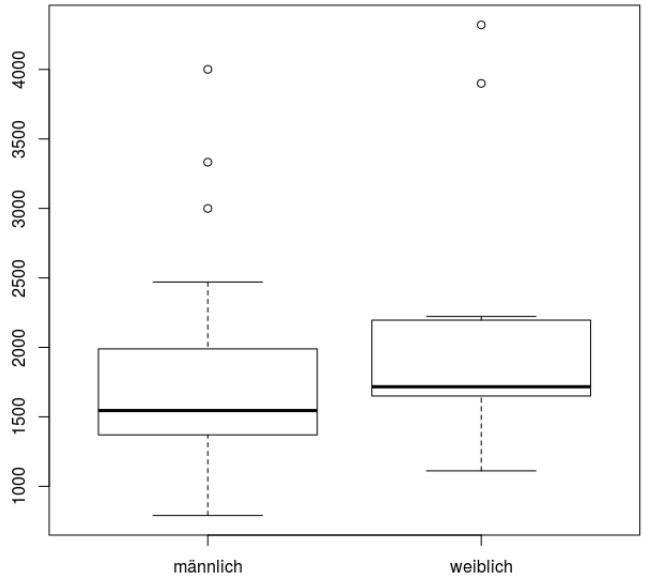

Figure 2.2

Diese Grafik ist schon viel übersichtlicher als unser Säulendiagramm von früher. Man kann sehen wieviel männliche und weibliche Teilnehmer im Schnitt verdienen, und sogar Teilnehmer die sehr von dem Rest abweichen werden durch einzelne Punkte dargestellt. Wir haben nun also ein Diagramm, was auf eine übersichtliche Weise darstellt, wieviel die männlichen und weiblichen Teilnehmer verdienen. Trotzdem ist dieses Diagramm immer noch ein schlechtes Beispiel. Da es in unserem Datensatz um ein Rennen geht, macht es keinen Sinn Einkommen und Geschlecht gegenüberzustellen. Es macht mehr Sinn unsere einzelnen Daten mit der Laufzeit gegenüberzustellen, da Zeit für ein Rennen am wichtigsten ist.

Wir stellen also unsere Daten mit der Laufzeit gegenüber:

2.2.1 Startnummer und Zeit

```
1 plot (Startnummer~Zeit)
```

Zuerst stellen wir die Startnummer der Teilnehmer mit der Laufzeit der Teilnehmer mit einem Plot (grafische Darstellung) gegenüber.

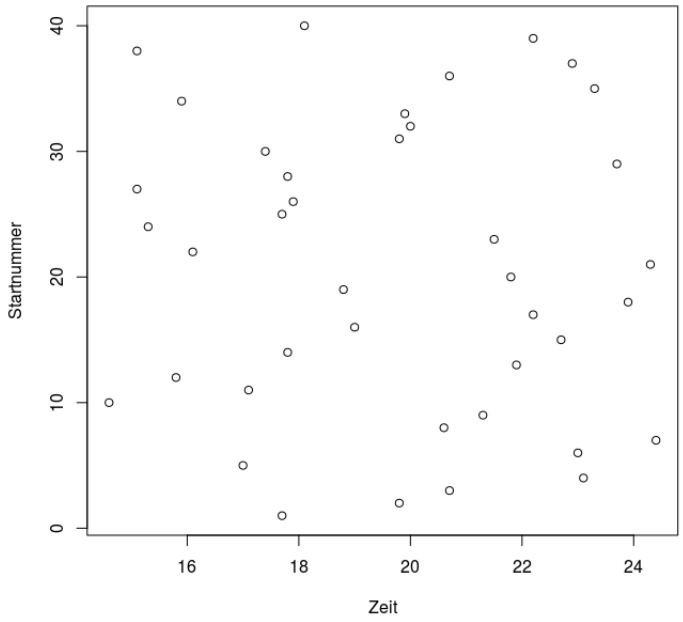

Figure 2.3

Wir sehen, dass unser Plot eine gut gewählte Grafik ist. Die Startnummer hat keinen Einfluss auf die Laufzeit des Rennens, deswegen sind die Punkte in unserem Plot alle wild durcheinander aufgelistet.

2.2.2 Einkommen und Zeit

```
1 plot (Einkommen~Zeit)
```

Hier werden wir das Einkommen und die Laufzeit jedes Teilnehmers gegenüberstellen. Wir werden für diese Gegenüberstellung wieder mit einem Plot arbeiten.

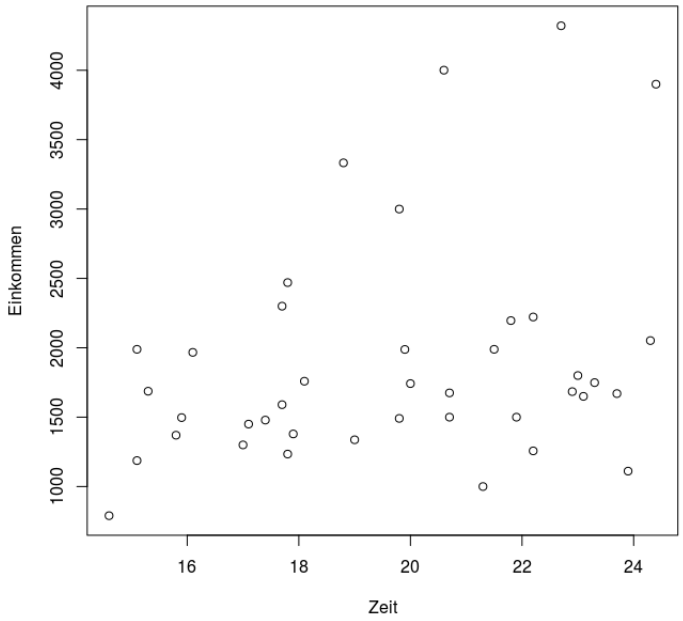

Figure 2.4

Hier sehen wir, dass unser Plot wieder gut gewählt wurde, da das Einkommen offensichtlich keinen Einfluss auf die Laufzeit hat. Wir sehen, dass unsere Punkte wieder wild angeordnet sind, mit ein paar Ausnahmen. Zum Beispiel sind die 3 Teilnehmer mit dem höchsten Einkommen auch die Teilnehmer die länger als der Durchschnitt gebraucht haben um das Rennen zu beenden. Aber auch diese Ausnahmen werden gut sichtbar in der Grafik gezeigt.

2.2.3 Geschlecht und Zeit

```
1 boxplot (Zeit~Geschlecht)
```

Für die Gegenüberstellung von Geschlecht und Zeit, benutzen wir ein Kastendiagramm.

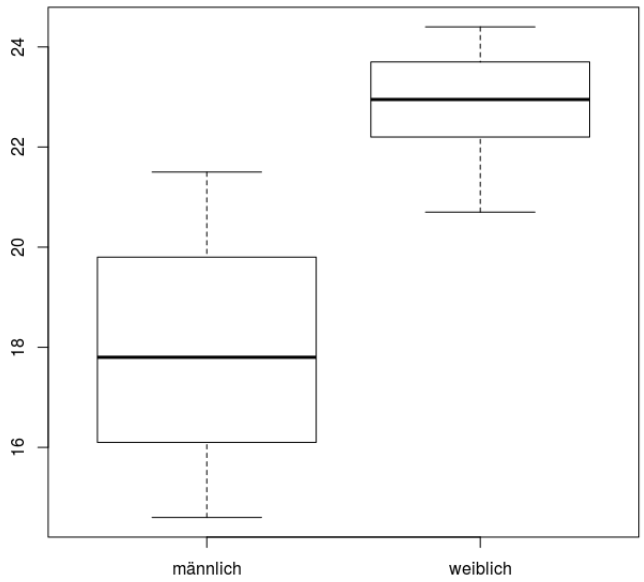

Figure 2.5

Das Kastendiagramm ist gut gewählt, da man nun schnell sehen kann, dass weibliche Teilnehmer des Rennens im Schnitt länger gebraucht haben um das Rennen zu beenden als männliche Teilnehmer. Der Kasten im Kastendiagramm beinhaltet den Bereich in dem die mittleren 50% der Teilnehmer liegen, die schwarze horizontale Linie ist der Durchschnitt des jeweiligen Kastens. Die gestrichelte Linien (Whiskers) die oben und unten vom Kasten zu sehen sind, zeigen die Werte an, die außerhalb der mittleren 50% liegen. Wenn es Teilnehmer gibt, die sehr stark von den Werten der anderen Teilnehmer abweichen, werden diese als einzelne Punkte, sogenannte 'Ausreisser' dargestellt (siehe Figur 2.2).

2.2.4 Alter und Zeit

```
1 plot (Alter~Zeit)
```

Wir stellen Alter und Zeit wieder mit einem Plot gegenüber.

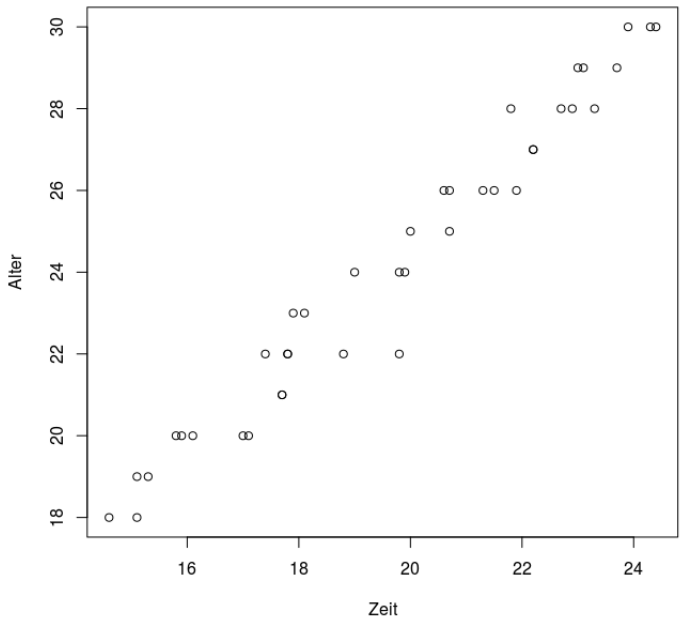

Figure 2.6

Auch hier ist ein Plot gut gewählt, da man hier genau sieht, dass das Alter direkt mit der Laufzeit des einzelnen Teilnehmers in Verbindung steht. Man sieht schnell, dass umso jünger der Teilnehmer ist, desto geringer ist im Schnitt die Laufzeit. So entsteht durch die einzelnen Punkte eine vertikale Linie, die diese Beobachtung anschaulich verdeutlicht.

3 Zusammenfassung

Diese kurze Einführung in die Programmiersprache sollte dem Leser einen Einstieg in R ermöglichen.
Es wurde erklärt, dass R eine frei verfügbare Programmiersprache ist, die 1992 entwickelt wurde und seit 1995 der GNU General Public License unterliegt. Wichtig ist besonders das Jahr 1997, in diesem Jahr wurde CRAN gestartet, um das Teilen von selbstgeschriebenen Funktionen auf einer Plattform zu ermöglichen, sowie das 'R Development Core Team' gegründet um R weiterzuentwickeln.
Die Vor- und Nachteile von R wurden ebenfalls gegenübergestellt. Hier wurden Punkte aufgezählt, die für das Arbeiten mit R wichtig sind, aber auch Punkte die für einige Personengruppen wichtig sind, wenn sie vor der Entscheidung stehen ob sie R überhaupt nutzen wollen. Zum Beispiel wird erwähnt, dass R Open-Source ist, eine schnelle Entwicklung stattfindet (zwischen der Version 3.3.0 und 3.3.1 liegen zum beispiel nur 1,5 Monate) und keine Lizenzgebühren anfallen. Auf der anderen Seite besitzt R keine komplett grafische Oberfläche, die Fehlermeldungen die R ausgibt sind oft nicht hilfreich für den Anwender und die Qualität der Pakete die man für R erhält, sollten genau auf den Zweck überprüft werden, da sonst Fehler auftreten können.
Es wurde beschrieben wie R mit mathematischen Funktionen und Rechnungen umgeht, dass zum Beispiel die Ergebnisse in der Konsole ausgegeben werden und die Anzahl der Ziffern die R ausgeben kann auf bis zu 22 verändert werden kann. Im letzten Abschnitt wurde anhand eines Datensatzes, basierend auf einem fiktiven Rennen, gezeigt wie man Grafiken erstellt und welche Punkte hierfür wichtig sind. Zum Beispiel sollte man sich Gedanken machen, welche Grafik man benutzen und welche Werte man genau in die Grafik mit aufnehmen möchte.

Quellenverzeichnis

- Zitat [1] Christian Heumann, Vorlesung Programmieren in statistischer Software: R
- https://www.r-project.org/
- https://cran.r-project.org
- https://cran.r-project.org/doc/manuals/R-intro.pdf
- Johannes Hain, Statistik mit R Grundlagen der Datenanalyse