
Interaktive Grafiken – Ein einfaches und effektives Mittel zur Vermittlung komplexer Zusammenhänge

1

Stefan Englert

Zusammenfassung

Wenn ein Bild mehr als tausend Worte sagen kann, so bieten dynamische interaktive Grafiken ein noch größeres Potenzial. In der Lehre eignen sich diese insbesondere als einfaches und effektives Mittel zur Visualisierung und damit zur Vermittlung komplexer Sachverhalte.

Wir beschreiben hier zwei Möglichkeiten, wie sich solche interaktive Grafiken sehr einfach realisieren lassen. Beide Möglichkeiten sind technisch leicht umsetzbar und benötigen insbesondere keine weiteren Installationen in den Unterrichtsräumen. Sie sind somit geeignet, um in der Lehre verwendet werden zu können.

Anhand zweier Beispiele, die aus einer Lehrveranstaltung für Studierende der Humanmedizin übernommen wurden, demonstrieren wir, wie diese direkt in eine Lehrveranstaltung integriert bzw. aus dem bestehenden Unterrichtsmaterial heraus entwickelt wurden.

Zusätzliches Lehrmaterial zur einfachen Anwendung der eingereichten Unterrichtsideen steht auf der Springer-Homepage <http://www.springer.com/978-3-662-54824-0> zur Verfügung.

S. Englert (✉)
Institut für Medizinische Biometrie und Informatik (IMBI),
Ruprecht-Karls-Universität Heidelberg, Marsilius-Arkaden,
Im Neuenheimer Feld 130.3, D-69120 Heidelberg, Deutschland
E-Mail: englert@imbi.uni-heidelberg.de

1.1 Einleitung

Dieser Beitrag zu interaktiven Grafiken ist durch die Lehrveranstaltung „Querschnittsbereich Epidemiologie, Medizinische Biometrie und Medizinische Informatik“ motiviert, welche für Studierende der Humanmedizin an der Universitätsklinik Heidelberg angeboten wird. Die Themen der Veranstaltung umfassen neben einer Einführung in Zufallsvariablen und Wahrscheinlichkeitsverteilungen die Bereiche Prognose, Diagnose und Therapie. Dabei werden auch anspruchsvollere statistische Verfahren, wie diagnostische Gütekriterien oder komplexere statistische Testverfahren, behandelt. Ziel ist es jeweils, dass die Studierenden die unterschiedlichen Einflussfaktoren begreifen und ein Verständnis für die Zusammenhänge entwickeln.

Um beispielsweise ein Grundverständnis verschiedener statistischer Wahrscheinlichkeitsverteilungen zu vermitteln, wurden lange Zeit mehrere verschiedenfarbige Linien innerhalb einer Grafik dargestellt. Diese Linien repräsentierten dabei jeweils eine andere Parameterkonstellation. In komplexeren Situationen, etwa zur Illustration der Abhängigkeit einer Teststatistik oder Fallzahlplanung von einem oder mehreren Einflussparametern, war eine derartige Darstellung nicht direkt möglich. Hier wurde der zugrundeliegende Zusammenhang anhand mehrerer hintereinander folgender Grafiken illustriert und so den Studierenden nähergebracht.

Leichter ist dies anhand einer einzigen, jedoch dynamischen Grafik möglich. Die technische Machbarkeit stellte lange ein Hindernis dar. Mittlerweile steht den Dozenten eine ganze Sammlung dieser interaktiven Grafiken zur Verfügung, die sie in die Lehrveranstaltung einbauen können.

Im Folgenden werden wir sehen, dass Softwarelösungen existieren, die diese Umsetzung sehr einfach möglich machen. Dazu werden zwei unterschiedliche Herangehensweisen beschrieben. Erstere basiert auf einer Lösung von Wolfram Research, den Entwicklern des kommerziellen mathematisch-naturwissenschaftlichen Softwarepakets „Mathematica“. Die zweite Herangehensweise basiert auf der freien Programmiersprache „R“.

Beide Herangehensweisen haben dabei gemeinsam, dass keine speziellen Installationen im Unterrichtsraum notwendig sind; ein Beamer, an den man seinen Laptop anschließen kann, genügt. Alle weiteren Schritte lassen sich direkt auf dem Laptop durchführen. Eine Internetverbindung erleichtert jeweils den Zugriff auf die interaktiven Grafiken, ist jedoch nicht zwingend notwendig. Beide Ansätze eignen sich demnach hervorragend um schnell und effektiv in bestehende Unterrichtsmaterialien eingebaut werden zu können.

1.2 Methodik

Im Folgenden werden zwei einfache und effektive Möglichkeiten präsentiert, wie sich interaktive Grafiken für eine Lehrveranstaltung erstellen und verwenden lassen.

1.2.1 Wolfram Research Demonstrations

Wolfram Research hat für sein kommerzielles mathematisch-naturwissenschaftliches Softwarepaket „Mathematica“ ein spezielles Dateiformat für interaktive Dokumente entwickelt, das Computable Document Format oder kurz CDF. Im Gegensatz zum bekannteren Portable Document Format oder kurz PDF von Adobe Acrobat ist der Inhalt des Dokumentes nicht statisch, sondern reagiert dynamisch bzw. interaktiv auf Veränderungen durch den Nutzer. Wolfram Research spricht hier von sogenannten *Demonstrations*.

Zur Erstellung und Bearbeitung dieser interaktiven Demonstrations ist eine Vollversion von Mathematica notwendig. Innerhalb der Software erlaubt das Kommando `Manipulate` Interaktivität und so die Visualisierung davon, wie Graphen (und andere Terme) sich in Abhängigkeit von einem oder mehreren Parametern verhalten. Die Verwendung ist dabei denkbar einfach. In `Manipulate` wird zuerst der Term angegeben, der von einem Parameter abhängt, beispielsweise eine Plot-Anweisung. Als Zweites folgt der Parameter, der veränderbar sein soll, zusammen mit einer Bereichsangabe in der Form `{variable, von, bis}`. Als Ergebnis erhält man nicht nur eine Grafikausgabe, sondern es wird zusätzlich ein Schieberegler eingeblendet, mit dem der angegebene Parameter variiert werden kann. Je nach Einstellung des Reglers wird die zugehörige Grafik neu berechnet.

`Manipulate` kann beispielsweise dazu verwendet werden, die Abhängigkeit einer Verteilung von einem oder mehreren Parametern darzustellen. Der folgende Befehl erzeugt innerhalb von Mathematica eine (statische) Grafik der Dichte einer Standardnormalverteilung (S. Abb. 1.1):

```
Plot[PDF[NormalDistribution[0,1],x],{x,-6,6}].
```

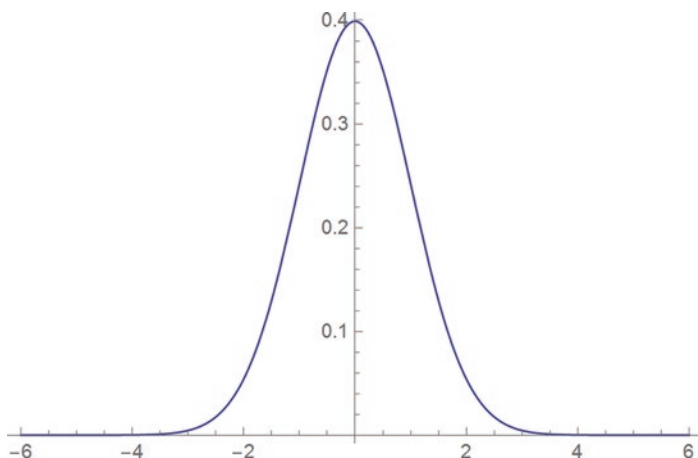


Abb. 1.1 Statische Grafik der Dichte einer Standardnormalverteilung aus Mathematica

Um diese Grafik interaktiv bezüglich des Streuungsparameters zu gestalten (s. [Abb. 1.2](#)), genügt der folgende Befehl in Mathematica:

```
Manipulate[Plot[PDF[NormalDistribution
[0,σ],x],{x,-6,6}],{σ,1,2}].
```

Die fertige interaktive Grafik kann nun aus Mathematica heraus im Computable Document Format abgespeichert und als Datei komfortabel weiterverteilt werden. Um diese anzusehen, ist dabei nur ein frei verfügbares Anzeigeprogramm, der Computable Document Format Player, notwendig.

Hier lässt sich eine Parallele zu Adobe Acrobat und dem PDF Format ziehen. Während das Programm zur direkten Erstellung und Bearbeitung der Dokumente, der Adobe Acrobat Pro, kostenpflichtig ist, steht das Anzeigeprogramm, der Adobe Acrobat Reader, kostenlos zum Download zur Verfügung.

Im Gegensatz zum PDF lassen sich neue Demonstrationen nicht durch andere Programme erzeugen. Da nicht jede Schule bzw. Universität über eine Volumenlizenz für Mathematica verfügt, ist dies eine bedeutende Einschränkung. Die direkte Erstellung neuer Demonstrationen ist somit mit zusätzlichen Kosten verbunden. Glücklicherweise ist eine Neuerstellung häufig überhaupt nicht notwendig. Im Rahmen eines ebenfalls von Wolfram Research gegründeten Wolfram Demonstration Projekts™ werden Demonstrationen zu unterschiedlichen Themengebieten gesammelt und frei zur Verfügung gestellt. Mittlerweile stehen insgesamt mehr als 10.000 fertige derartige Demonstrationen zur Verfügung (s. [Abb. 1.3](#)).

Im Bereich Statistik mit aktuell annähernd 1000 Demonstrationen finden sich fertige Lösungen zu unterschiedlichen allgemeinen Prognose-, Diagnose- und Testverfahren,

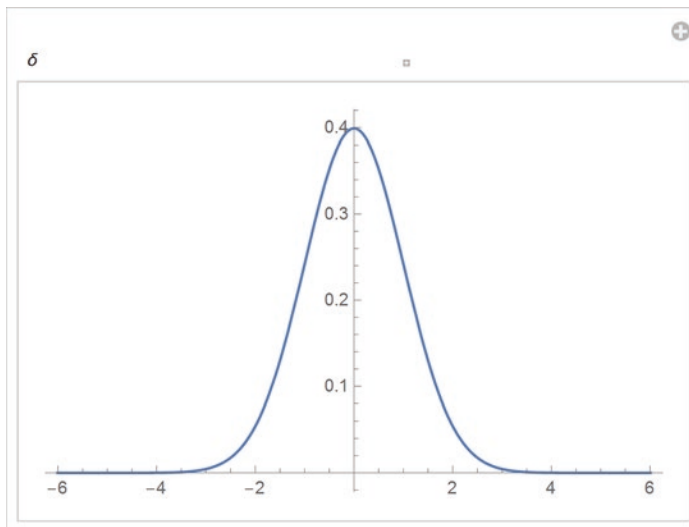


Abb. 1.2 Interaktive Grafik der Dichte einer Normalverteilung aus Mathematica

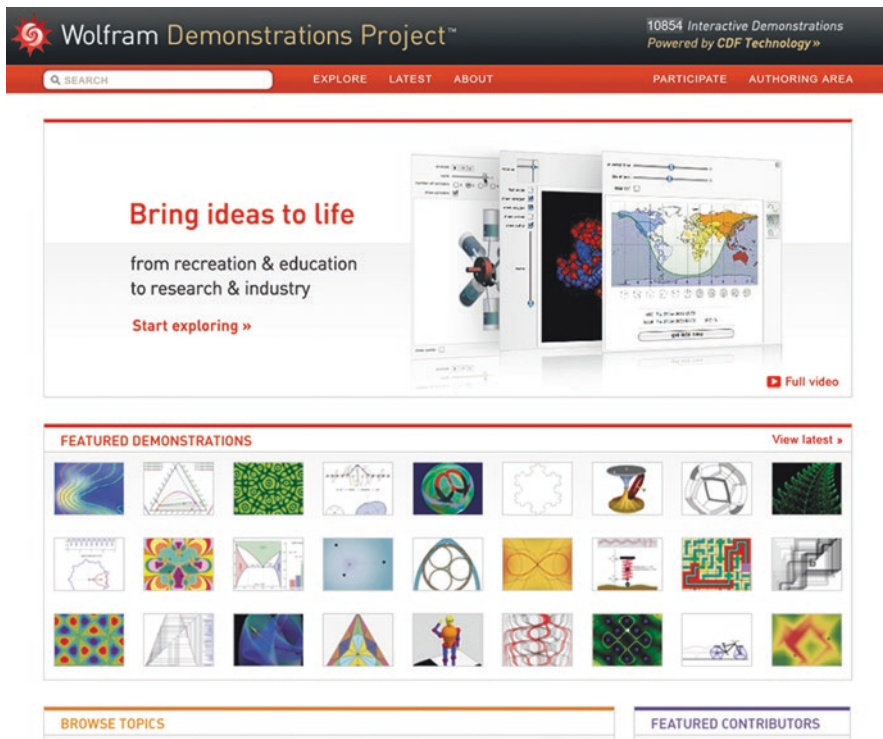


Abb. 1.3 Screenshot des Wolfram Demonstration Project mit über 10.000 fertigen interaktiven Demonstrationen (<http://demonstrations.wolfram.com/>; reproduziert mit Erlaubnis von Wolfram Research, Inc.)

sowie zu vielen Spezialthemen wie dem Satz von Bayes, Simpson's Paradox, dem Benfordschen Gesetz und vielem mehr.

Jede einzelne Demonstration ist dabei ausführlich dokumentiert. Sofern der Computational Document Format Player auf dem PC installiert ist, können die Demonstrationen auch direkt aus dem Browser heraus interaktiv verwendet werden. Sie können immer auch als Datei heruntergeladen werden und sind so ohne Internetverbindung nutzbar.

1.2.2 R Shiny-Applikationen

Die zweite Herangehensweise zur Erstellung interaktiver Grafiken, die wir hier vorstellen möchten, sind sogenannte R Shiny-Applikationen. Interaktive Grafiken sind dabei fast nur ein Nebenprodukt. Bei R Shiny-Applikationen handelt es sich vielmehr um eine Möglichkeit, professionell aussehende Cloud-basierte interaktive Ausgaben mit interaktiver Benutzeroberfläche zu erstellen. Diese wurden von R Studio entwickelt und basieren auf der freien Programmiersprache R. Die zusätzliche Funktionalität lässt sich dabei komfortabel direkt in R als Erweiterungspaket einbinden.

Zur Programmierung derartiger Applikationen wird nur Wissen in der Programmiersprache R benötigt. Ein R-Programmierer kann R Shiny sehr schnell lernen und Applikationen erstellen. Selbst für komplexe und professionelle Benutzeroberflächen sind keine weiteren Programmierkenntnisse, insbesondere nicht in HTML oder JAVA, notwendig.

Die Entwickler von R Shiny stellen auf ihrer Homepage ein zweieinhalb-stündiges Tutorial bereit (<http://shiny.rstudio.com/tutorial/>). Dieses beschreibt umfassend alle notwendigen Schritte von der Erstellung einer R Shiny Applikation bis zur Verbreitung über eine öffentlich zugängliche Homepage.

Wenn die für die Lehrveranstaltung benötigten Grafiken schon in R programmiert wurden, so lassen sich diese besonders einfach in eine interaktive R Shiny-Applikation überführen. Beispielsweise erzeugt der linke Codeabschnitt in Tab. 1.1 eine Grafik der diskreten Dichtefunktion der Binomialverteilung. Dazu werden zuerst die beiden Parameter n , für die Fallzahl, und p , für die Erfolgswahrscheinlichkeit definiert und anschließend wird die Grafik über einen Plot-Befehl ausgegeben. Der prinzipielle Aufbau ist bei einer R Shiny Applikation der gleiche (s. Tab. 1.1 rechts).

Für eine R Shiny-Applikation muss das „shiny“-Paket installiert und im Code geladen werden. Darauf folgen Codeabschnitte, die einerseits die Benutzeroberfläche (User

Tab. 1.1 Code in Rs zur grafischen Darstellung der diskreten Dichtefunktion der Binomialverteilung

	Code zu Erzeugung einer statischen R-Grafik	Code zur Erzeugung einer interaktiven R-Grafik (R Shiny-Applikation)
1		library(shiny)
2		
3		ui <- fluidPage(
4	<code>n <- 10</code>	sliderInput(„n“,
5		label = „n“, value = 10,
6		min = 1, max = 25),
7	<code>p <- 0.2</code>	sliderInput(„p“,
8		label = „p“, value = 0.2,
9		min = 0, max = 1),
10		plotOutput(„pdfPlot“)
11)
12		server <- function(input, output) {
13		output\$pdfPlot <- renderPlot({
14	<code>plot(0:n,</code>	plot(0:input\$n,
15	<code>dbinom(0:n, n, p),</code>	dbinom(0:input\$n, input\$n,
16	<code>type = „h“,</code>	input\$p),
17	<code>xlab = „Anzahl Treffer“,</code>	type = „h“,
18	<code>ylab =</code>	xlab = „Anzahl Treffer“,
19	<code>„Wahrscheinlichkeit“</code>	ylab = „Wahrscheinlichkeit“
20	<code>)</code>)})
21		}
22		shinyApp (ui = ui, server = server)

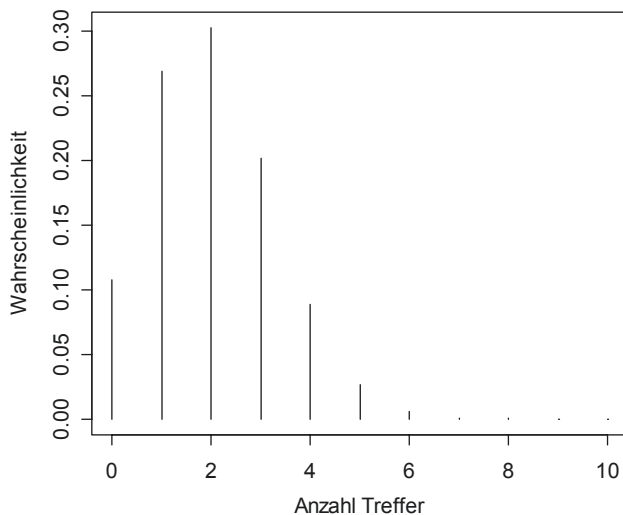
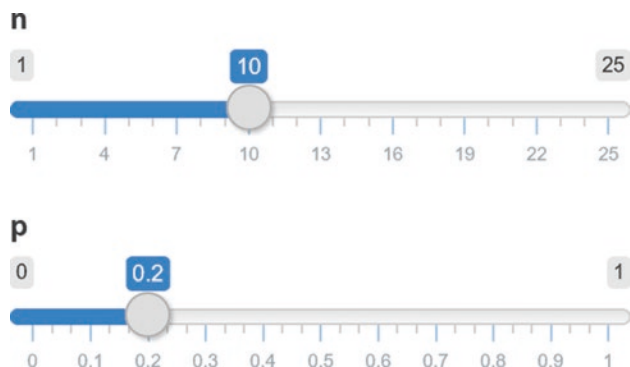


Abb. 1.4 Diskrete Dichtefunktion der Binomialverteilung

Interface, UI) und andererseits die zur Durchführung notwendigen Berechnungen (Server) definieren. Der Code bleibt dabei sehr gut lesbar und ist mit R-Kenntnissen nahezu direkt verständlich. In unserem Beispiel besteht die Benutzeroberfläche aus zwei Schiebereglern, je einer für die Fallzahl n und die Erfolgswahrscheinlichkeit p , sowie der Grafikausgabe. Interaktiv berechnet wird, basierend auf diesen Eingaben, lediglich die Grafikausgabe. Für weitere Details zu den verwendeten Befehlen verweisen wir auf die oben erwähnte Einführung zu R Shiny-Applikationen.

Ausgeführt in R erzeugen beide Codeteile von [Tab. 1.1](#) erst einmal eine Grafik wie in [Abb. 1.4](#). Die R Shiny-Applikation wird dabei in einem Browserfenster geöffnet und es befinden sich neben der Grafik zusätzlich zwei Steuerelemente wie in [Abb. 1.5](#). Werden die zugehörigen Schieberegler verändert, so aktualisiert sich direkt und interaktiv die zugehörige Grafik

Abb. 1.5 Zusätzliche Steuerelemente der R Shiny-Applikation



R Shiny bietet zusätzlich die Möglichkeit, die Applikation kostenlos im Internet zu veröffentlichen. Für die oben genannte Applikation haben wir dies unter dem Link:

<https://imbi.shinyapps.io/binom-app-min/>

getan. Für eine detaillierte Beschreibung der Vorgehensweise verweisen wir wieder auf das Tutorial zu R Shiny.

Die fertige R Shiny-Applikation kann über den Link direkt in einem Webbrowser angezeigt werden, ohne dass weitere Software oder Plug-Ins nötig sind. Dadurch funktioniert sie, unabhängig vom jeweiligen Betriebssystem, auf PCs, Tablets und Smartphones.

Endnutzer der veröffentlichten Applikation – dies können sowohl Lehrende als auch Studenten sein – benötigen insbesondere auch keine Kenntnisse mehr in der Programmiersprache R.

1.3 Beispielanwendung

Die folgenden zwei Beispiele sind ein Auszug aus den im Kurs „Querschnittsbereich Epidemiologie, Medizinische Biometrie und Medizinische Informatik“ an der Universitätsklinik in Heidelberg verwendeten Anwendungen. Die Beispiele stammen aus den Bereichen Diagnose (Beispiel 1) und Therapie (Beispiel 2). Wir gehen dabei jeweils kurz auf den thematischen Hintergrund ein und heben den Sachverhalt hervor, der durch eine interaktive Grafik illustriert und erklärt werden soll.

1.3.1 Beispiel 1 Wolfram Research Demonstrations: ROC Kurve

Im Bereich Diagnose werden in der Lehrveranstaltung einfache Diagnosetests mit binären Messgrößen behandelt. Dabei werden die diagnostischen Gütekriterien der Sensitivität und Spezifität erklärt und anhand von Beispielen von den Studierenden berechnet. Die Beispiele entstammen dabei einem quantitativen Diagnosetest, aus dem durch unterschiedliche Schwellenwerte binäre Diagnosetests konstruiert wurden. Letztendlich wird so zur Receiver Operating Characteristic (ROC)-Kurve übergeleitet, welche wie folgt definiert ist:

Receiver Operating Characteristic (ROC)-Kurven entstehen, wenn man bei quantitativen Diagnosetests die kritische Schwelle für die Test-Positivität variiert. Dabei erhält man eine Familie binärer Diagnosetests, die alle unterschiedliche Sensitivität und Spezifität haben. Die Paare (Sensitivität, $1 - \text{Spezifität}$) bilden die ROC-Kurve.

Die Fläche unter der ROC-Kurve drückt die Fähigkeit des Diagnosetests aus, zwischen „krank“ und „gesund“ zu unterscheiden.

Wir wollen nun diesen Sachverhalt anhand einer Demonstration darstellen. Sowohl im Wolfram Demonstration Project™ als auch in den elektronischen Materialien zu diesem

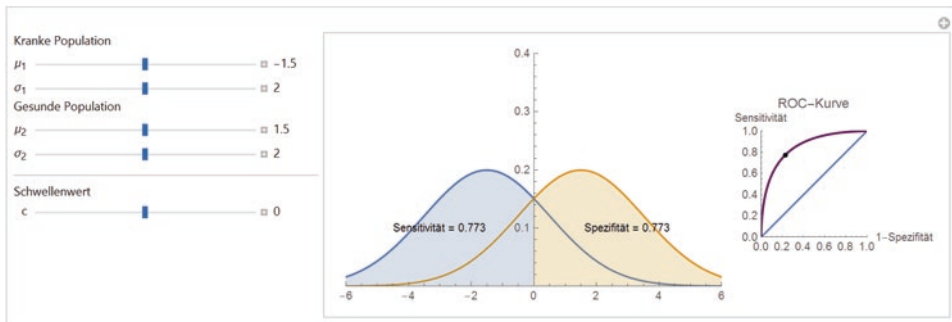


Abb. 1.6 Screenshot aus der Demonstration zur ROC-Kurve

Beitrag befindet sich eine geeignete Demonstration, anhand derer die Funktionsweise und Konstruktion der ROC-Kurve erklärt werden kann.

Die Demonstration zeigt sowohl für eine kranke Population als auch für eine gesunde Population die Verteilung der Messgröße (s. [Abb. 1.6](#)). Für einen festgelegten Schwellenwert wird die Sensitivität und Spezifität des zugehörigen binären Diagnosetests berechnet und in die Grafik eingetragen. Zusätzlich wird das Paar (Sensitivität, 1 – Spezifität) in eine separaten Grafik gezeigt, in welcher auch die gesamte ROC-Kurve eingetragen ist.

Variiert man nun den Schwellenwert, so ändert sich auch die Sensitivität und Spezifität des zugehörigen binären Diagnosetests (s. [Abb. 1.7](#)). In der Demonstration kann dies direkt mitverfolgt werden. So kann beispielsweise live in der Lehrveranstaltung gezeigt werden, wie der Punkt die ROC-Kurve entlangwandert und diese so beschreibt.

Ebenso kann aufgezeigt werden, wie sich die ROC-Kurve verändert, wenn sich die kranke und gesunde Population stärker unterscheiden bzw. ähnlicher sind (s. [Abb. 1.8](#)). Erwartungsgemäß wird bei größerer Differenzierung die Fläche unter der ROC-Kurve größer. Sie hat den Wert 0.5, falls es keine Trennung zwischen der kranken und gesunden Population gibt.

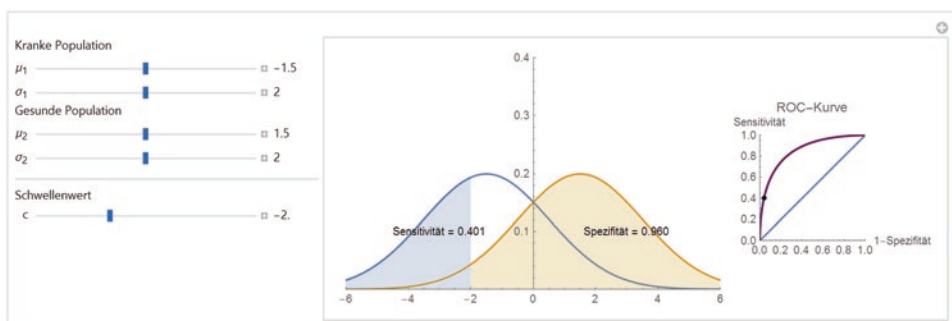


Abb. 1.7 Screenshot aus der Demonstration zur ROC-Kurve (mit verändertem Schwellenwert)

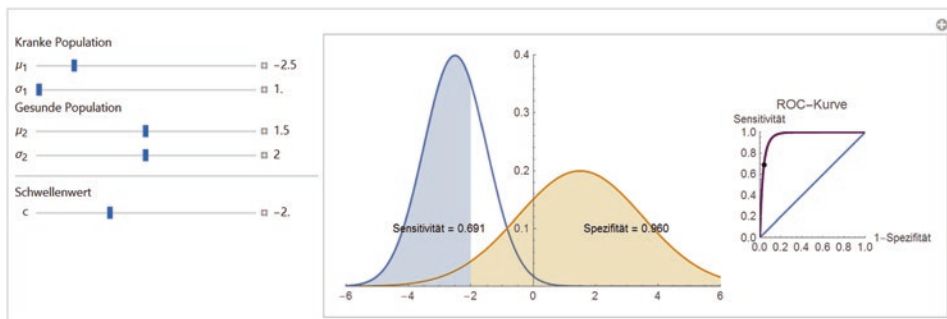


Abb. 1.8 Screenshot aus der Demonstration zur ROC-Kurve (mit veränderten Parametern für die Verteilung der kranken Population)

1.3.2 Beispiel 2 R Shiny-Applikation: p-Wert für Binomialtest

Im Bereich Therapie wird in der Lehrveranstaltung das Prinzip des statistischen Testens anhand des Binomialtests behandelt. Dabei wird die Formulierung der korrekten Null- und Alternativhypothese, die Wahl einer passenden Prüfgröße und insbesondere auch der p-Wert behandelt. Der p-Wert des Binomialtests ist wie folgt definiert:

Für einen Binomialtest beschreibt der p-Wert die Wahrscheinlichkeit, den in der Stichprobe beobachteten Wert oder einen in Richtung der Alternativhypothese noch extremeren Wert zu beobachten, wenn in Wahrheit die Nullhypothese zutrifft.

Um dies geeignet zu illustrieren, wurde das in [Abschn. 1.2.2](#) beschriebene Minimalbeispiel einer R Shiny-Applikation noch erweitert und auf der Internetseite

<https://imbi.shinyapps.io/binom-app/>

veröffentlicht. Die fertige Applikation erlaubt verschiedene Einstellungsmöglichkeiten für die Anzahl der Versuche, die Erfolgswahrscheinlichkeit, sowie für die Wahrscheinlichkeit, die berechnet werden soll. Der zugehörige Bereich wird dabei in der Dichtefunktion der Binomialverteilung hervorgehoben und die zugehörige Wahrscheinlichkeit ausgegeben (s. [Abb. 1.9](#)).

Zur Berechnung des p-Werts stellt man die Anzahl der Versuche und die Erfolgswahrscheinlichkeit gemäß der Nullhypothese ein und definiert, dass alle Wahrscheinlichkeiten, die größer als (extremer) oder gleich der angegebenen Grenze sind, summiert werden sollen. Der zugehörige Wert wird dann direkt durch die Applikation berechnet und

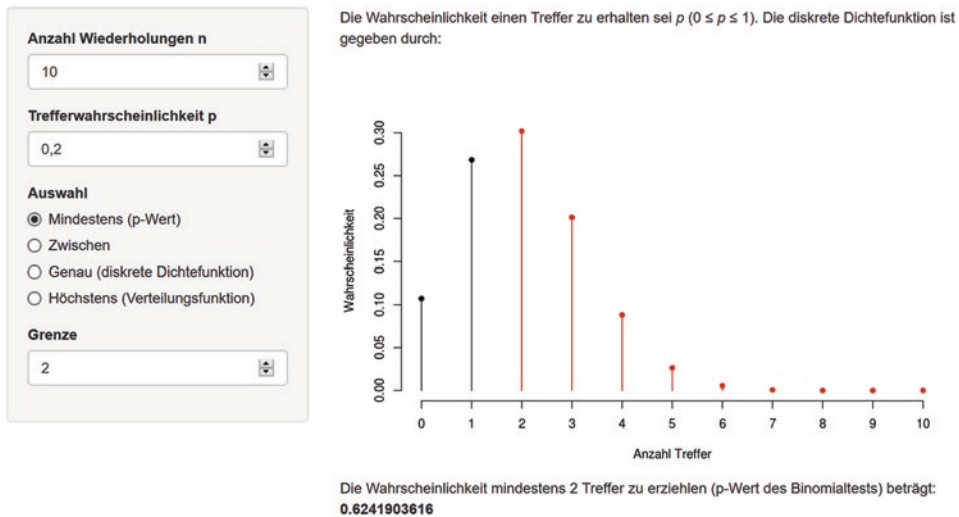


Abb. 1.9 Screenshot der R Shiny-Applikation zur Illustration des p-Werts des Binomialtests

ausgegeben. Durch eine Veränderung der Parameter kann mitverfolgt werden, wie sich der p-Wert des Binomialtests verändert.

Gleichmaßen kann diese Applikation dazu genutzt werden, die Power des statistischen Tests zu illustrieren. Dazu passt man zuerst die Grenze so lange an, bis der p-Wert gerade unter das festgelegte Signifikanzniveau (beispielsweise 0.05) fällt. Dieser Wert entspricht der Grenze, ab der die Nullhypothese abgelehnt werden kann und der Binomialtest ein signifikantes Ergebnis liefert. Passt man nun die Erfolgswahrscheinlichkeit auf den unter der Alternative angenommenen Wert an, so erhält man die statistische Power. Diese wird direkt aus der Applikation heraus berechnet.

1.4 Diskussion und Ausblick

Abschließend möchten wir nochmals die Vor- und Nachteile der beiden hier vorgestellten Herangehensweisen zur Realisierung interaktiver Grafiken gegenüberstellen (s. [Tab. 1.2](#))

Zusammenfassend lässt sich sagen, dass Softwarelösungen existieren, die interaktive Grafiken sehr einfach ermöglichen. Wolfram Demonstrations überzeugen dabei in erster Linie dadurch, dass durch das Wolfram Demonstration Project™ eine große Sammlung mit vorgefertigten Inhalten zur Verfügung steht. R Shiny-Applikationen eignen sich vor allem für Anwender, die bereits mit der Programmiersprache R vertraut sind.

Tab. 1.2 Vor- und Nachteile von Wolfram Demonstrations und R Shiny-Applikationen

Wolfram Demonstrations		R Shiny-Applikationen	
Direkte Nutzung in der Lehre			
⊕	Bibliothek mit über 10.000 fertigen interaktiven Grafiken vorhanden	⊖	Erstellung einer R Shiny-Applikation muss erst erlernt werden
⊖	Installation von Zusatzsoftware nötig	⊕	Direkt aus dem Browser aufrufbar, keine Zusatzsoftware/ Plug-Ins nötig
⊖	Nur auf PC anwendbar	⊕	Auf PC, Tablet und Smartphone anwendbar
Erstellung eigener interaktiver Grafiken			
⊖	Erstellung nur mittels Software „Mathematica“ möglich (kostenpflichtig)	⊕	Sowohl R Studio als auch R Shiny sind frei verfügbar
⊕	Erstellung interaktiver Grafiken sehr einfach möglich; lediglich ein einziges zusätzliches Kommando notwendig	⊕	Erstellung interaktiver Grafiken sehr einfach möglich; ausführliche Einführung ist vorhanden
Weiternutzungsmöglichkeiten			
⊕	Dateien können direkt ausgetauscht bzw. auf Lehreportalen zur Verfügung gestellt werden	⊖	Direkter Austausch als R Programm erfordert Erfahrung des Anwenders mit der Programmiersprache R
⊕	Dateien können auch ohne Internetverbindung ausgeführt werden	⊖	Öffentliche fertige Applikationen können nur bei bestehender Internetverbindung ausgeführt werden

Anhang

Folgende elektronische Materialien zu diesem Beitrag finden Sie online:

- Interaktive Grafiken im Computable Document Format zur
 - Receiver Operating Characterisitics (ROC) Kurve als „ROC.cdf“
 - Standardisierung einer Normalverteilung als „Standardisierung.cdf“
 - Dichte- und Verteilungsfunktion einer Standardnormalverteilung als „Standardnormalverteilung.cdf“
 - Teststatistik des T-Tests als „TTeststatistik.cdf“
 - Verteilungsfunktion der Binomialverteilung als „Binomialverteilung.cdf“

- Verteilungsfunktion der Chi-Quadrat-Verteilung als „ChiQuadratVerteilung.cdf“
 - Verteilungsfunktion der (Standard-)Normalverteilung als „Normalverteilung.cdf“
 - Verteilungsfunktion der T-Verteilung als „TVerteilung.cdf“
- 2 R Shiny-Applikation im R Dateiformat zur
- Verteilungsfunktion der Binomialverteilung als „BinomAppMinimal.R“ (Minimalbeispiel)
 - Verteilungsfunktion der Binomialverteilung als „BinomApp.R“
- 3 R Shiny-Applikation als Webseite zur
- Verteilungsfunktion der Binomialverteilung: <https://imbi.shinyapps.io/binom-app-min/> (Minimalbeispiel)
 - Verteilungsfunktion der Binomialverteilung: <https://imbi.shinyapps.io/binom-app/>

Weblinks

R Shiny Applikationen Tutorial: <http://shiny.rstudio.com/tutorial/>

Wolfram CDF Player: <http://www.wolfram.com/cdf-player/>

Wolfram Demonstrations Project™: <http://demonstrations.wolfram.com/>

Zeig mir mehr Biostatistik!

Mehr Ideen und neues Material für einen guten
Biometrie-Unterricht

Vonthein, R.; Burkholder, I.; Muche, R.; Rauch, G.

2017, X, 150 S. 100 Abb., 54 Abb. in Farbe., Softcover

ISBN: 978-3-662-54824-0