



CORRELAID

GOOD CAUSES. BETTER EFFECTS.

Programmieraufgabe oder Lückentext?

Wir visualisieren Daten und automatisieren Reports (für
Anfänger:innen)

Wer wir sind



Wir sind ein deutschlandweites Netzwerk von über 1,700 Data Scientists, die die Welt durch Data Science verbessern wollen.

#MetaWeltretter





CORRELAID
GOOD CAUSES. BETTER EFFECTS.

Und wer seid Ihr?

Frage 1 – Wie IT-affin seid Ihr? (SelectOne)

Naja.../Ich komme klar/Ich bin Profi!


























Frage 2 – Eure Kenntnisse mit R? (SelectOne)

Noch nie gehört/Mal aufgemacht/Ich hab mal einen Kurs besucht/Ich führe eigene Analysen in R durch

Frage 3 - Was erhofft Ihr Euch vom Workshop? (MC)


Verständnis für neue Technologien/Code lesen lernen/Selber programmieren!

Agenda

Themen	Methodik	Zeitaufwand
Gemeinsamer Start	    	10min
Theoretische Einführung	    	20min
Angeleitetes Lernen (Übung 1 bis 5)	    	45min
Peer Learning (Übung 6 bis 10)	    	30min
Ausblick	    	15min



Spielregeln für unsere Zusammenarbeit

- 
- A close-up, shallow depth-of-field photograph of a chessboard. The board is made of light and dark squares. Several chess pieces are visible: a white pawn in the foreground, a dark rook, a white rook, and a dark king in the background. The lighting is warm and focused on the pieces.
- 1: Fragen gerne immer im Chat stellen
 - 2: Arbeitet aktiv in Euren Skripten mit
 - 3: Nutzt den technischen Support-Raum

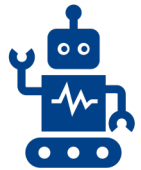
Ziele des Workshops

- Anwendungsmöglichkeiten für die Technologie RShiny erkennen
- Code lesen können
- Grundverständnis für die technische Implementierung gewinnen
- Interesse wecken mehr zu lernen

Prinzipien des Workshops

- Alle Pflichtaufgaben können durch das Kopieren von Codesegmenten mit kleinen Änderungen erledigt werden
- Bitte lest die Aufgabenstellungen gründlich durch (##### Übung X)
- Grüner Text kennzeichnet Kommentare, in denen wichtige Informationen für Euch stehen

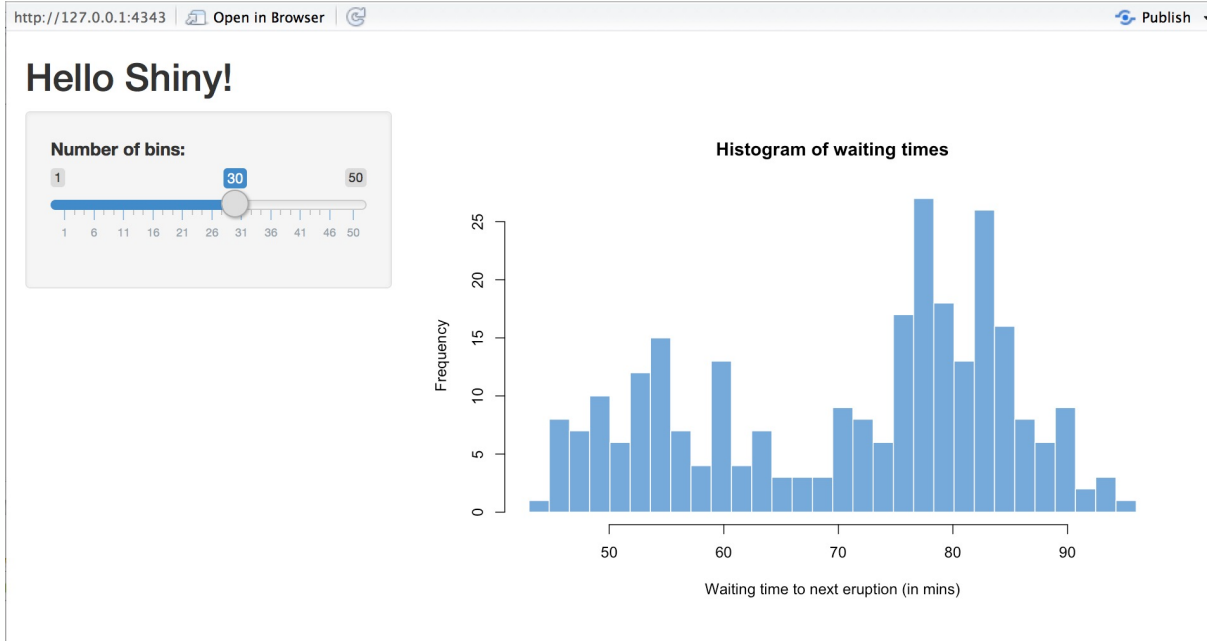
Öffnet am Besten parallel
Zoom und RStudio –
idealerweise arbeitet Ihr
dabei mit zwei Monitoren.



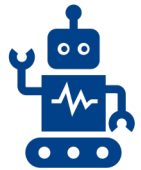
Web-Applikationen begegnen uns überall – doch was ist das eigentlich?



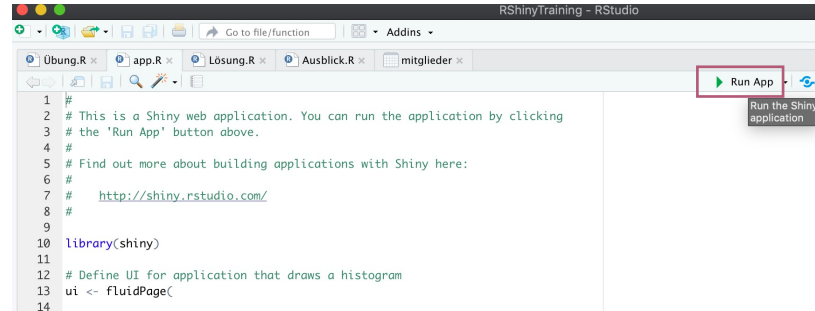
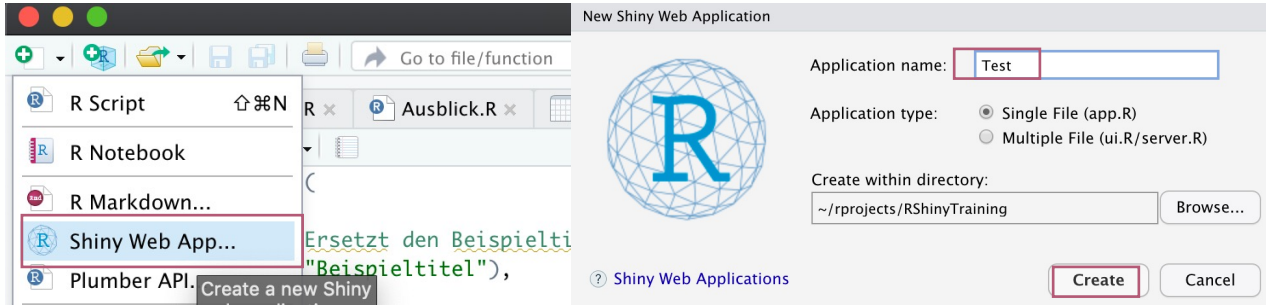
Shiny ist ein R-Package, das es Euch ermöglicht ganz leicht interaktive Web-Applikationen zu bauen



Packages... Was ist das eigentlich? Packages sind Module, die man in seine Codeskripte laden kann. Sie enthalten alle möglichen Funktionalitäten - wie z.B. die Funktion `mean()`. Mit ihm müsst Ihr die mathematische Formel für den Mittelwert nicht mehr selbst bestimmen, sondern Ihr könnt die Funktion anwenden und ihn direkt berechnen.



Öffnet nun Eure erste Shiny-Applikation



RShiny besteht aus zwei Hauptbestandteilen: User Interface (UI) und Server

Aussehen der App



UI

(User interface)

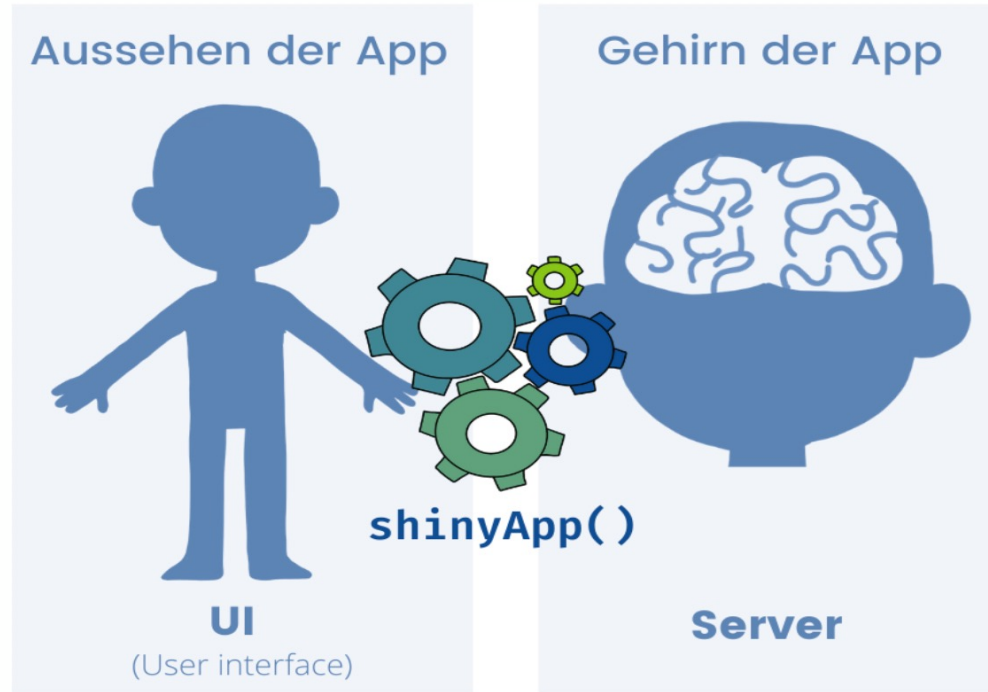
Gehirn der App



Server



User Interface (UI) und Server funktionieren nicht ohne einander



Zusammen bauen wir heute diese App

Mitgliederdaten und Feedbackumfrage - Fantasie e.V.

Wähle den Ort aus:

Berlin ▼

Wähle das Beitrittsjahr aus:

☒ 2012

☐ 2014

☐ 2015

☐ 2016


☐ 2017

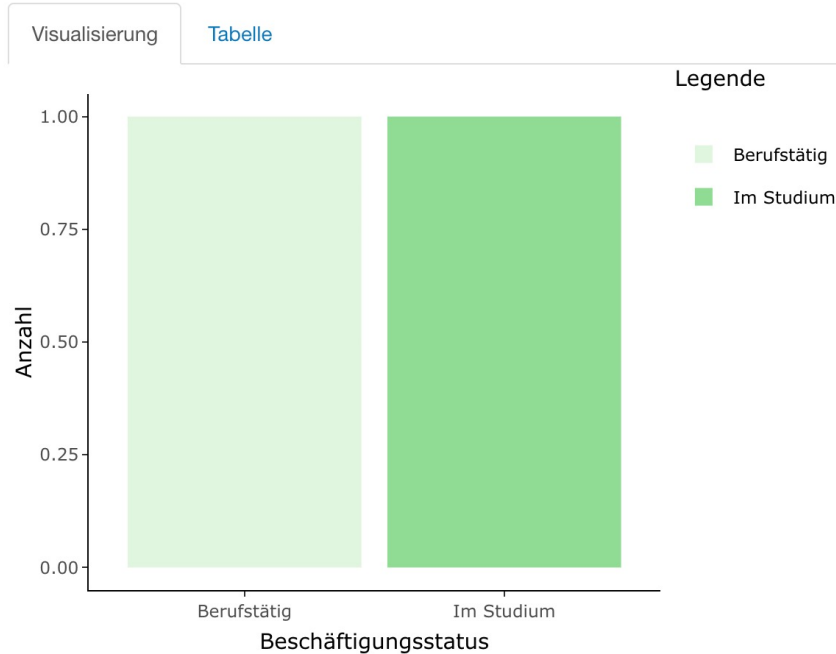
☐ 2018

☐ 2019

☐ 2020

☐ 2021

 Download



Das User Interface (ui.R) bestimmt das Aussehen der App (I)

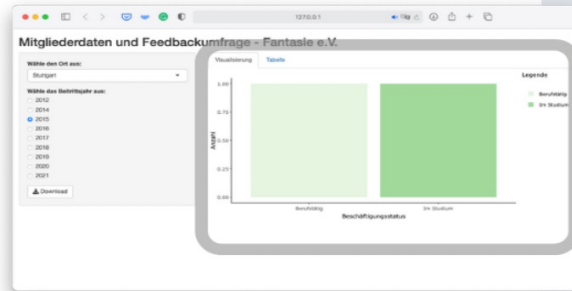


```
ui <- fluidPage(  
  titlePanel("Titel der App"),  
  sidebarLayout(  
    sidebarPanel(  
      selectInput("Auswahl")  
    ),  
    mainPanel(  
      tabsetPanel(  
        tabPanel("Visualisierung",  
          Balken-  
          diagramm  
        ),  
        tabPanel("Tabelle",  
          Tabelle  
        )  
      )  
    )  
  )  
)
```

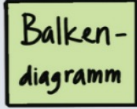

Was
möchte
ich
sehen?



Das User Interface (ui.R) bestimmt das Aussehen der App (II)



```
ui <- fluidPage(  
  titlePanel("Titel der App"),  
  sidebarLayout(  
    sidebarPanel(  
      selectInput("Auswahl")  
    ),
```

```
    mainPanel(  
      tabsetPanel(  
        tabPanel("Visualisierung",  
            
        ),  
        tabPanel("Tabelle",  
            
        )  
      )  
    )  
  )  
)
```

Anzeige
der App



Der Server (server.R) kreiert das Gehirn der App

```
server <- function(input, output, session) {  
  # Erstelle die Visualisierung
```

Balken-
diagramm


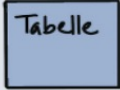
```
# Erstelle die Tabelle
```

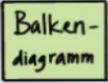
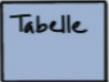
Tabelle

```
}
```



shinyApp() verbindet die beiden Teile miteinander

```
ui <- fluidPage(  
  titlePanel("Titel der App"),  
  sidebarLayout(  
    sidebarPanel(  
      selectInput("Auswahl")  
    ),  
    mainPanel(  
      tabsetPanel(  
        tabPanel("Visualisierung",  
            
        ),  
        tabPanel("Tabelle",  
            
        )  
      )  
    )  
  )  
)
```

```
server <- function(input, output, session) {  
  # Erstelle die Visualisierung  
    
  # Erstelle die Tabelle  
  
```

```
shinyApp(ui = ui, server = server)
```

Öffnet nun das RProjekt, in dem wir heute arbeiten werden

The screenshot shows the RStudio interface. The main editor window displays a script file named 'RShinyTraining - main - RStudio'. The script contains R code for loading packages, setting up the environment, and loading data. The console window at the bottom shows the output of the script, including a message about the 'tidy' package and a table of data. The file explorer on the right shows the project structure, including a 'Global Environment' window and a 'Files' window listing various files and folders.

```
## Die installierten Packages müsst Ihr laden. Erst so werden die Funktionalitäten zugänglich.
## Dazu nutzt Ihr den Befehl "library(Package-Name)" - ohne Anführungszeichen!
## Hinweis: Ihr könnt auch ohne library arbeiten. Mit der Notation package::funktion() greift Ihr direkt auf die Funktionen zu.

library(shiny) # Das RShiny Package.
library(dplyr) # Das dplyr Package, mit dem wir tolle Datenbereinigungen vornehmen können.
library(ggplot2) # Das ggplot2 Package, mit dem wir die Daten visualisieren können.
library(knitr) # Hiermit knitten wir unseren Report.
library(tinytex) # Dieses LaTeX-Package gibt uns die Option, den Report als PDF zu exportieren.

#### Übung 2: Lade nun auch das Package "tidy".

#####

# 3) DATEN LADEN
## Ihr Script sucht - genau wie Ihr es tun würdet - die genutzten Daten.
## Sogenannte "Paths" (zu dt. Wege) geben an, wo lokale Dokumente abliegen.
## Dabei werden die verschiedenen Ordner mit "/" getrennt und der gesamte Path in Anführungszeichen gesetzt.
## Als absolute Paths bezeichnet man die Option, den gesamten Path vorzugeben, also eine Reise von Berlin nach München mit allen Zwischenstationen zu definieren.
## Relative Paths starten bei einem gesetzten Startpunkt, wo z.B. Ihr Script liegt. Damit können Scripts leichter geteilt werden.
## Die Idee ist: Wenn Ihr erst in Dresden losfährt, dann braucht Ihr auch erst ab die Fahrstrecke.
## So müssen Eure Kolleg:innen nicht erstmal den Path anpassen, wenn sie Ihr Script auf ihrem PC ausführen möchten.
## Hinweis: In R gibt's dafür für neue Projekte einen tollen Trick - Nutzt RProjekte und schiebt alle Eure Dokumente in den entsprechenden Ordner (https://rproj.org/).
## Alle Pfade beginnen dann in Eurem RProject-Ordner.
## Übrigens: Mit sogenannten APIs könnt Ihr digital erhobene Daten ebenfalls in R laden. Schaut mal, ob das auch für Eure Umfragetools funktioniert!

## Hier laden wir die Datensätze, die Ihr am besten immer in einem Ordner namens "Daten" speichert.

15:58 [Untitled] R Script

Console Terminal R Markdown Jobs
~/projects/projektzyklus-workshops/10_datanalyse-interactive-dashboards/RShinyTraining/
> runApp("Ausblick.R")

Listening on http://127.0.0.1:3938
'summarise()' has grouped output by 'Mitglieds-ID'. You can override using the '.groups' argument.
Table<grb (4 x 6) "arrange": 5 grubs
  z     cells      name      grub
1 1 (1-2,1-2) arrange  gtable[layout]
2 2 (3-4,1-2) arrange  gtable[layout]
3 3 (3-4,3-4) arrange  gtable[layout]
4 4 (3-4,5-6) arrange  gtable[layout]
5 5 (1-2,3-6) arrange  gtable[colhead-fg]

> runApp("Übung.R")

Listening on http://127.0.0.1:3938
```

Die einzelnen Dateien könnt Ihr unten rechts öffnen und die App mit “Run App” ausführen

The screenshot shows the RStudio interface with a script editor on the left containing R code for a Shiny application. The code includes comments in German and R commands for installing packages and running the application. A red box highlights the 'Run App' button in the top right. A red box labeled '1. Dateien öffnen' points to a file explorer window on the right, which lists files in the 'RTraining' directory. The console at the bottom shows the output of the 'runApp()' command, including warnings about package versions and masked objects.

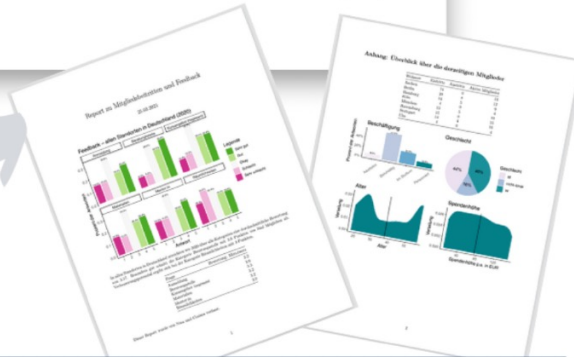
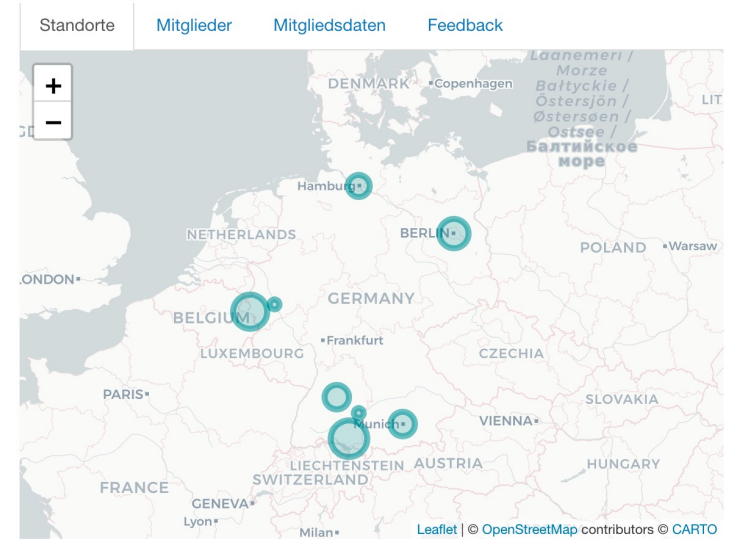
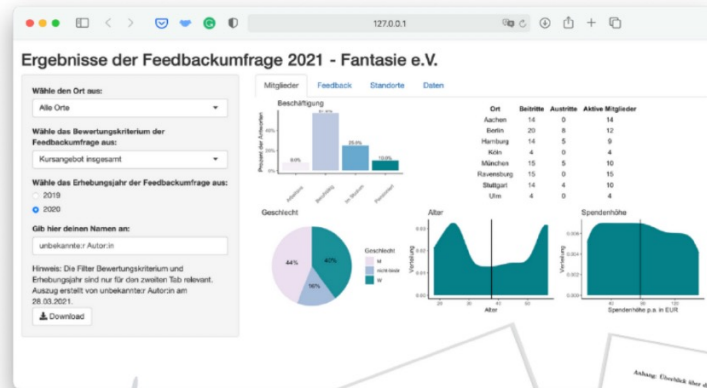
2. App ausführen

1. Dateien öffnen

```
1 # INTERAKTIVE VISUALISIERUNGEN MIT (R) SHINY
2
3 ## Dieses Codescript hilft Euch eure erste Shiny Web-Applikation zu bauen.
4 ## Ihr könnt den Code ausführen, indem Ihr oben rechts "Run App" anklickt.
5 ## Mehr Informationen findet Ihr unter: http://shiny.rstudio.com/
6 ## Hilfe zu Funktionen findet Ihr über help(funktion) oder ?funktion
7
8 #####
9
10 # 1) RELEVANTE PACKAGES INSTALLIEREN
11 ## Sag, Packages (zu dt. Pakete) enthalten alle wichtigen Funktionalitäten, die Ihr braucht.
12 ## So müsst Ihr beispielsweise die Formel des Mittelwerts nicht selbst definieren.
13 ## Ihr könnt einfach die Funktion mean() nutzen.
14 ## Packages werden über den Befehl "install.packages("Package-Name")" installiert.
15 ## Los geht es: Löscht die Anführungszeichen vor und hinter dem Codesegment, das mit "install.packages..." anfängt.
16 ## Markiert das Codesegment und führt es mit CTRL + Enter aus.
17 ## Hinweis: Folgt unten in der Console der Ausführung des Codes und bestätigt die Installation von Abhängigkeiten ggf. mit "Yes".
18
19 "
20 install.packages(c(
21   'shiny',
22   'ggplot2',
23   'tidyr',
24   'dplyr',
25   'plotly',
26   'DT',
27   'rnorm'
28 ))
29
30 Console Terminal Jobs
31 ~/Downloads/RTraining/
32 > library(shiny); runApp("Übung.R")
33
34 Warnmeldung:
35 Paket 'shiny' wurde unter R Version 4.0.2 erstellt
36 Warnung: package 'dplyr' was built under R version 4.0.2
37
38 Attaching package: 'dplyr'
39
40 The following objects are masked from 'package:stats':
41
42   filter, lag
43
44 The following objects are masked from 'package:base':
45
46   intersect, setdiff, setequal, union
47
48 Warnung: package 'ggplot2' was built under R version 4.0.2
49 Warnung: package 'knitr' was built under R version 4.0.2
50 Warnung: package 'tinytex' was built under R version 4.0.2
```

Name	Size	Modified
..		
..Rhistory	28.1 KB	Mar 25, 2022
..Ausblick.R	17.1 KB	Mar 30, 2022
..Daten		
..Lösung.R	9.1 KB	Mar 25, 2022
..report.Rmd	1.3 KB	Mar 25, 2022
..report_ausblick.Rmd	11 KB	Mar 30, 2022
..RShinyTraining.Rproj	205 B	Mar 31, 2022
..Schummelblatt_ggplot2.pdf	456.7 KB	Mar 25, 2022
..Schummelblatt_rmarkdown.pdf	656.6 KB	Mar 25, 2022
..Schummelblatt_shiny.pdf	621.4 KB	Mar 25, 2022
..Übung.R	8.1 KB	Mar 25, 2022

Und jetzt? Ihr könnt den Code ganz leicht auf andere Projekte übertragen



Und Eure ShinyApp veröffentlichen

Mit unsensiblen Daten kann man die ShinyApp ganz einfach und kostenlos über shinyapps.io veröffentlichen

Vor dem Veröffentlichen auf jeden Fall beachten: Alle Packages müssen installiert sein (mit `install.packages(...)`) -- das darf allerdings nicht im Code erscheinen.

Schritt-für-Schritt-Anleitung:

1. Installiert `rsconnect` (`install.packages('rsconnect')`)
2. Erstellt ein kostenloses Benutzerkonto auf shinyapps.io
3. Folgt Methode 1 wie [hier](#) beschrieben: Kopiert das Token und fügt es in Euer RStudio ein
4. (Wenn Ihr die App vorab noch einmal testen wollt: `runApp()`)
5. Zum Schluss: Veröffentlicht Eure ShinyApp mit `deployApp()`

Einen kleinen Moment warten... und das war's! Eure App ist online!

Online findet Ihr tolle Weiterbildungsformate

The screenshot displays the Dataquest website interface. At the top, the 'DATAQUEST' logo is on the left, and a yellow button labeled 'GET FREE LIFETIME ACCESS' is on the right, accompanied by search and help icons. Below this is a dark blue header for the 'CAREER PATH' titled 'Data Analyst in R', with the subtitle 'Learn how to manipulate and analyze data.' and a green 'Start Learning for Free' button. A progress bar shows seven steps, with Step 1 being the active and completed step. The main content area is titled 'Introduction to R' and indicates 'STEP 1 of 7 - 5% Done'. It lists two courses: 'Introduction to Data Analysis in R' (labeled 'COURSE 1 / 4') and 'Data Structures in R' (labeled 'COURSE 2 / 4'). Under the first course, a list of topics is shown, with 'Introduction to Programming in R' marked as completed with a green checkmark. The other topics are 'Arithmetic Expressions and Variables in R', 'Logical Expressions in R', 'Data Manipulation with R: Basics', and 'Project: Install RStudio'. Each course entry includes a 'Go to course page' link.

Hier noch ein paar Tipps für Weiterbildungsformate aus unserem Netzwerk

Zu R:

- [Data Analyst in R](#)
- [The R Bootcamp](#)

Zu Shiny:

- [R Studio Shiny Tutorial](#)
- [Hadley Wickham: "Mastering Shiny"](#)
- [Konstantin Gavras and Nick Baumann: Shiny Apps: Development and Deployment auf Methods Bites](#)
- [Garrett Golemund: "How to understand reactivity in R"](#)

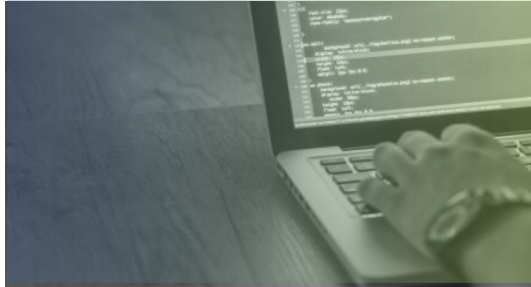
Zu Hosting:

- [Shinyapps.io Tutorial](#)

Click-and-Drop Code-Schummeleien:

- [Graphiken mit esquisse](#)

Und natürlich gibt es von uns noch Hilfestellungen



PROJEKTE

Wir führen pro-bono Datenanalyseprojekte für gemeinnützige Organisationen durch.



BILDUNG

Wir vernetzen engagierte sozial denkende Datenanalytist:innen und bieten ihnen Möglichkeiten ihr Wissen anzuwenden und zu erweitern.



DIALOG

Wir treten in den Dialog über den Wert und Nutzen von Daten und Datenanalysen für das Gemeinwohl





CORRELAID



For instant updates on our work
follow us on Twitter
@CorrelAid



Receive the latest infos in our
projects via Facebook
@WeAreCorrelAid



Sign up for our mailing list on
www.correlaid.org