

# TECHNISCHE HOCHSCHULE ULM

## PROJEKTBERICHT - GRUPPE 1

### **R und Shiny Apps**

Studiengang: DSM5

*Alexander Metzler*

[REDACTED]

Mat. Nr.: [REDACTED]

*Yannik Krantz*

[REDACTED]

Mat. Nr.: [REDACTED]

*Florian Max Hauptmann*

[REDACTED]

Mat. Nr.: [REDACTED]

betreut durch  
Lars ANDERSEN  
Julia OBENAUER

February 4, 2024

# CONTENTS

<b>I</b>	<b>Einführung</b>	<b>2</b>
I-1	Kurzinformation . . . . .	2
I-2	Übertragungswege . . . . .	2
<b>II</b>	<b>Planung des bearbeiteten Datensatzes und der App</b>	<b>2</b>
II-1	Vorgehensweise . . . . .	2
II-2	Veränderungen und hinzugefügte Spalten . . . . .	3
II-3	Grafiken und Tabellen . . . . .	3
II-4	Zusätzliche Features . . . . .	3
<b>III</b>	<b>Umsetzung des Datensatzes</b>	<b>4</b>
III-1	Bearbeitung des Datensatzes . . . . .	4
<b>IV</b>	<b>Umsetzung der App</b>	<b>5</b>
IV-1	Wie wurde die App programmiert . . . . .	5
IV-2	Verwendete R-Pakete . . . . .	10
<b>V</b>	<b>Fazit und Beteiligungen</b>	<b>10</b>
V-1	Alexander Metzler . . . . .	10
V-2	Yannik Krantz . . . . .	11
V-3	Florian Max Hauptmann . . . . .	11
<b>VI</b>	<b>Anhang</b>	<b>12</b>
	<b>References</b>	<b>18</b>

## I. EINFÜHRUNG

### 1) Kurzinformation:

Die Krankheit COVID-19, Kurzform für "Coronavirus Disease 2019" [1], ist eine Infektionskrankheit, welche durch den Erreger SARS-CoV-2 verursacht wird. Dieses Virus war Anfang 2020 der Auslöser der COVID-19-Pandemie. Der Hauptübertragungsweg des Virus ist die Tröpfcheninfektion, also über kleine Aerosole, die eingeatmet werden. Zu den häufigsten Symptomen zählen Husten, Fieber, Schnupfen und die Störung des Geschmacks- und Geruchssinns. Besonders risikobehaftet sind vor allem ältere Personen und Personen mit Vorerkrankungen. Die mediane Inkubationszeit beträgt in etwa 4-6 Tage, je nach Virusvariante und die Krankheit dauert etwa 8-10 Tage an, falls keine Therapie durchgeführt wird. [2]

### 2) Übertragungswege:

Der Hauptübertragungsweg des COVID-19 Erregers ist die Tröpfcheninfektion. Dabei werden kleine Partikel in der Luft, welche die Viren enthalten, eingeatmet und gelangen so in den Körper. Die Übertragungswahrscheinlichkeit von einer Infizierten Person zu einer Nichtinfizierten steigt dabei, wenn sich beide Personen in einem geschlossenen Raum aufhalten, besonders wenn dieser schlecht belüftet ist. Es empfiehlt sich demnach bei längerem Aufenthalt in einem Raum regelmäßig zu lüften. Auch das Tragen von Masken wie einem Mund-Nasen-Schutz können das Risiko einer Infektion senken. [2]

## II. PLANUNG DES BEARBEITETEN DATENSATZES UND DER APP

### 1) Vorgehensweise:

#### **Planung**

Die Planung des Projekts begann mit einem "Kickoff-Meeting", bei dem wir uns alle trafen, um die Anforderungen des Projekts zu besprechen und den zugrundeliegenden Datensatz zu analysieren. Dieser erste Schritt ermöglichte es uns, eine klare Vorstellung von den Zielen des Projekts zu erhalten und sicherzustellen, dass alle ein gemeinsames Verständnis der Anforderungen hatten.

Um die Arbeitsabläufe effektiv zu organisieren, nutzten wir das Backlog-Tool von GitHub. Hier wurden alle Aufgaben und Anforderungen in Form von Tasks festgehalten. Jeder Task wurde genauer beschrieben, und es wurde festgelegt, welcher Bereich des Projekts er abdeckte. Ein Beispiel für einen solchen Task war die Erstellung des Tabs "Einführung" in der Shiny-App.

Das Backlog wurde in drei Bereiche unterteilt: "Todo", "In Progress" und "Done". Diese Kategorisierung ermöglichte es uns, den Fortschritt jeder Aufgabe klar zu verfolgen. Jedes Teammitglied konnte sich eine Task aus dem "Todo"-Bereich auswählen und sie sich selbst zuweisen. Dadurch behielten wir stets den Überblick darüber, wer welche Aufgaben übernahm und wie der aktuelle Stand des Projekts war.

#### **Shiny App-Initialisierung und Versionsverwaltung**

Zu Beginn des Projekts wurde eine Shiny-App in R Studio initialisiert, und es wurde ein grober Aufbau programmiert, um die Grundstruktur der App zu definieren. Dieser frühe Schritt ermöglichte es uns, die Basis für die App zu legen und einen Grundbaustein für diese zu haben.

Für die Versionsverwaltung entschieden wir uns, Git zu nutzen. Der Code der Shiny-App wurde auf dem Hauptbranch ("main") online allen Teammitgliedern bereitgestellt.

Sobald die grundlegende Struktur der App vorhanden war, konnte jeder für seine ausgewählte Aufgabe einen Feature-Branch erstellen. Dies ermöglichte es jedem Teammitglied, unabhängig an seiner spezifischen Aufgabe zu arbeiten, ohne den Hauptcode zu beeinträchtigen.

Nachdem ein Task erfolgreich bearbeitet wurde, fand eine kurze Absprache im Team statt. Anschließend wurde nach einem kurzen Review der Änderungen gemeinsam in den Hauptbranch ("main") gemerged. Dabei wurde immer darauf geachtet, dass der Code funktioniert.

## *2) Veränderungen und hinzugefügte Spalten:*

Während der Besprechung der Anforderungen und der Analyse der Daten sammelten wir schon Ideen für die Veränderung des Datensatzes. Es wurde schnell klar, dass der Datensatz bereinigt werden muss. Zudem lag eine Spalte für die Summe aller Impfdosen pro Zeile auf der Hand. Eine weitere Idee war gewisse Einträge anteilig der gesamten Impfdosen in der selben Region und Woche darzustellen. Ebenso musste die Anforderung für eine simulierte Age-Spalte erfüllt werden. Das simulierte Alter, die Datenbereinigung und die restlichen Anpassungen sahen wir jeweils als separate Aufgaben und teilten sie dementsprechend im Team auf.

## *3) Grafiken und Tabellen:*

Um die Anforderungen für Verschiedene Grafiken und Tabellen zu erfüllen, entschieden wir uns für einzelne Tabs für die verschiedenen Themenbereiche. Wir planten im ersten Tab auf die verschiedenen Arten der einzelnen Impfstoffe einzugehen. Mit einer Tabelle und einer Grafik wollten wir die Anzahl der vergebenen Impfdosen zwischen den verschiedenen Impfstoffen vergleichen. Der zweite Tab sollte auf die einzelnen Länder eingehen. Ebenfalls mit einer Grafik und einer Tabelle wollten wir die Unterschiedliche Menge an Impfungen zwischen den einzelnen Ländern vergleichen. Zudem wollten wir ermöglichen eine Zielgruppe auszuwählen und die Länder auf selbe Art für diese zu vergleichen. Im dritten Tab wollten wir mit einer Grafik und einer Tabelle auf die Rate an Verweigerungen der Erstimpfung über gewisse Zeiträume eingehen. Im letzten Tab wollten wir den Zeitlichen verlauf der Impfungen innerhalb einer Zielgruppe darstellen. Auch hier wollten wir wieder, dass Land und Zielgruppe ausgewählt werden können. Die Darstellung sollte in mindestens einer Grafik erfolgen.

## *4) Zusätzliche Features:*

Zur Verbesserung der App wurden folgende zusätzliche Features integriert:

- 1) Dark/Light Mode-Schalter: Ermöglicht Benutzern die Anpassung des Designs an ihre Präferenzen.
- 2) Bootstrap-Framework: Optimiertes Design für mobile Endgeräte und einheitliche Styling-Optionen.
- 3) Navigation-Bar statt TabSetPanel: Intuitive Navigation, die sich dynamisch an die Anzeigegröße anpasst.
- 4) GitHub-Logos in "About Us"-Sektion: Direkte Links zu den GitHub-Profilen der Entwickler und des Projekts.
- 5) Dropdown-Menüs für Datenfilterung: Effiziente Datenerkundung durch gezielte Filterung nach Ländern und Zielgruppen.

### III. UMSETZUNG DES DATENSATZES

#### Anmerkung

In den kommenden Abschnitten werden gelegentlich auftretende Probleme sowie die entsprechenden Lösungsansätze beschrieben. Da diese Probleme in vielen Bereichen aufgetreten sind, haben wir uns dazu entschieden, sowohl die Probleme, als auch die Lösungsansätze direkt zu beschreiben und ihnen keinen separaten Abschnitt zu widmen.

#### 1) Bearbeitung des Datensatzes:

Die erste Anpassung des Datensatzes war das Hinzufügen einer Spalte *simulate\_age* welche, wie in den Anforderungen für das Projekt beschrieben, ein zufälliges Alter aus der Spalte *target\_group* darstellen soll. Hierfür haben wir in einer Funktion eine if/else Abfrage gemacht um die korrekte *target\_group* zu bestimmen und mit der Methode `sample()` eine Zufallszahl in dem erforderlichen Bereich generiert.

Für die nächste Anpassung sind uns ungewöhnlich viele na-Werte in den numerischen Spalten des Datensatzes aufgefallen. Wir haben uns entschieden diese durch 0 auszutauschen. Mit der `mutate_if()` Methode war es möglich die na-Werte aller numerischen Spalten in nur einer Codezeile auszutauschen.

```
df <- df %>% mutate_if(is.numeric, function(x) ifelse(is.na(x),
0, x))
```

Ebenfalls auffällig war die geringe Menge an Einträgen über 0 für die Spalten *DoseAdditional2* – *DoseAdditional5*. So wurden beispielsweise in nur 0.25% der Wochen überhaupt *DoseAdditional4* injiziert. Wir haben uns, daher entschieden die genannten Spalten zur Spalte *MoreAdditionalDoses* zusammen zu fassen. *DoseAdditional1* haben wir zu *AdditionalDose* umbenannt. Da die Spalte *YearWeekISO* sowohl die Information über das Jahr als auch über die Woche enthält, haben wir sie in die Spalten *Year* und *Week* aufgeteilt. Da die Spalte einem festen Muster in der Darstellung folgte, konnten wir das Jahr und die Woche einfach mit der Methode `substring()` ermitteln.

```
df$Year <- as.numeric(substr(df$YearWeekISO, 1, 4))
```

Eine weitere Aufgabe war den Code, um Spalten zu erweitern. Die erste Anpassung, die wir hierfür durchgeführt haben, war eine Summe aller Impfungen in einer Zeile, unter dem Namen *DosesThisWeek*. Ein großes Problem des Datensatzes war, dass einige *target\_group* Gruppen zusätzlich unter *ALL* zusammengefasst waren. Allerdings bezog sich dies nur auf alle *AgeX\_Y* Gruppen ab 18 Jahren. Zählt man alle *AgeX\_Y* Gruppen (>18) in einer Region zusammen gleicht deren Gesamtzahl der in der *target\_group ALL* für die Entsprechende Region. Da in dem Data-Dictionary Dokument zum Datensatz die Gruppe *ALL* mit "overall adults" beschrieben wird, sind wir davon ausgegangen, dass *HCW* und *LTCF* ebenfalls in dieser Gruppe und somit auch in den einzelnen Altersgruppen vertreten sind. Diese Annahme hatte Relevanz für unsere nächste Anpassung. Es sollte auch eine Spalte geben für die wöchentliche Gesamtzahl der Impfungen pro Region. Hierfür haben wir ein separates Dataframe nur mit den Altersgruppen nach Jahr, Woche und Region gruppiert. Die daraus entstandene Spalte *WeeklyDosesPerRegion\_Total* haben wir dem ursprünglichen Dataframe durch `merge()` angefügt. Als letzte Anpassung haben wir uns für den Prozentualen Anteil einer Gruppe, an den gesamten Impfungen einer Region pro Woche, entschieden.

Hierfür haben wir den Prozentualen Anteil von *DosesThisWeek* an *WeeklyDosesPerRegion\_Total* berechnet und in der Spalte *PercentageOfTotalDosesInWeek* vermerkt.

#### IV. UMSETZUNG DER APP

##### 1) Wie wurde die App programmiert:

##### **Aufbau der App**

###### UI Abschnitt

Die "ui.R"-Datei ist durch einen klaren und benutzerfreundlichen Strukturaufbau gekennzeichnet. Eine Navigation-Bar (`navbarPage()`) leitet den Benutzer durch die Anwendung und befindet sich innerhalb einer `fluidPage()`, in der die Benutzeroberflächen-Elemente deklariert werden. Bootstrap [3] wurde für das Design verwendet, um eine optimale Ansicht auf verschiedenen Endgeräten zu gewährleisten. Die `navbarPage()` ist in verschiedene Tabs (`tabPanel()`) unterteilt, um die Navigation und Struktur zu erleichtern.

Am Anfang des Codes wurden zwei Vektoren deklariert, die Dropdown-Auswahlmöglichkeiten in den Tabs ermöglichen. Ein Vektor listet die Länder auf, die im Datensatz enthalten sind, und der andere Vektor enthält die Altersgruppen/Zielgruppen, nach denen gefiltert werden kann. Beide Vektoren haben jeweils einen Eintrag "All". Wenn dieser im Dropdown ausgewählt ist, wird der Datensatz nicht spezifisch gefiltert.

###### Server Abschnitt

Im Server-Teil des Codes wurde anfangs ein Mapping implementiert, um die Lesbarkeit der App zu verbessern. Dies ermöglicht es, bestimmte Codes wie "Age0\_4" in eine verständlichere und ansprechendere Form, wie "Alter 0 bis 4", zu übersetzen. Diese Zuordnungen, wie zum Beispiel von ISO-Ländercodes zu den entsprechenden deutschen Ländernamen oder von Altersgruppen zu den zugehörigen deutschen Bezeichnungen, tragen dazu bei, dass die präsentierten Informationen leichter verständlich sind.

Die Anpassung der ISO-Ländercodes hätte theoretisch bereits während der Datenbereinigung erfolgen können. Die Entscheidung für ein Mapping im Server-Teil wurde getroffen, um nicht deutschsprachigen Benutzern entgegenzukommen, die möglicherweise mit den ISO-Ländercodes vertrauter sind als mit den deutschen Bezeichnungen der Länder. Diese Anpassung verbessert die Benutzerfreundlichkeit insbesondere für internationale Anwender, da die ISO-Codes eine weltweit gebräuchliche und standardisierte Form der Länderkennzeichnung sind. Auf den genaueren Aufbau und die Funktionsweisen der einzelnen Tabs wird in den folgenden Abschnitten ausführlicher eingegangen.

##### Beispiel für das Mapping der Altersgruppen

```
selected_target_groups <-
c(
  "Age0_4", "Age5_9", "Age10_14", "Age15_17", "Age18_24", "Age25_49",
  "Age50_59", "Age60_69", "Age70_79", "Age80+", "AgeUnk"
)
age_names_german <-
c(
  "Alter_0_bis_4", "Alter_5_bis_9", "Alter_10_bis_14", "Alter_15_bis_17",
```

```
"Alter_18_bis_24", "Alter_25_bis_49", "Alter_50_bis_59", "Alter_60_bis_
  69",
"Alter_70_bis_79", "Alter_80+", "Altersgruppe_unbekannt"
)
# mapping of selected target groups and age names
age_mapping <- setNames(age_names_german, selected_target_groups)
```

Wenn zum Beispiel im Dropdown "Age0\_4" ausgewählt wurde, gibt das Mapping "Alter 0 bis 4" zurück. So kann auf das Mapping zugegriffen werden:

```
age_mapping[input$selectedTargetGroup]
```

### Dark- und Lightmode

Durch die Verwendung der bslib-Bibliothek können vorgefertigte Bootstrap-Themes [4] einfach eingebunden werden, z. B. mit dem Befehl `bslib::bs_theme(bootswatch = "theme_name")`. Diese Flexibilität inspirierte uns dazu, jeweils ein attraktives Theme für den Dark- und Light-Modus auszuwählen. Standardmäßig ist das UI im Dark-Modus mit dem Bootstrap-Theme "darkly" gestaltet. Der Wechsel zwischen den Modi wird über einen Schalter realisiert, der als UI-Element das MaterialSwitch [5] von der shinyWidgets-Bibliothek verwendet. Es ist wichtig, das Theme sowohl in der `fluidPage()` als auch in der `navbarPage()` zuzuweisen, um sicherzustellen, dass das Design korrekt übernommen wird [6]. Im Server-Teil wird eine Beobachtungsfunktion [7] verwendet, um das Theme basierend auf der Einstellung des Schalters zu ändern. Der folgende Code zeigt, wie dies umgesetzt wurde:

```
# Dark-Mode-Schalter
observe(session$setCurrentTheme(if (isTRUE(input$theme_switch))
                                dark
                                else
                                light))
```

Durch diese Implementierung bieten wir den Benutzern die Möglichkeit, zwischen einem ansprechenden Dark- und Light-Modus zu wählen, wodurch die Benutzererfahrung weiter verbessert wird. Ein Beispiel ist ein Fig. 1 und Fig. 2 zu sehen.

### Impfstoffe Tab

In diesem Tab sollen für die verschiedenen Impfstoffe (also "Comirnaty", "Moderna", "AstraZeneca", etc.) jeweils zwei Untersuchungen durchgeführt werden. Die erste Untersuchung betrifft die Gesamtvergabe der einzelnen Impfstoffe, also wie viele Impfdosen von einem Impfstoff insgesamt verabreicht wurden. Dargestellt werden einmal die Top 10 in einer Tabelle und einmal das Verhältnis dieser Summen in einem Kreisdiagramm, um sich eine bessere Vorstellung über die prozentualen Verhältnisse dieser Summen zu machen [Siehe Fig. 3 oberer Kasten]. Dafür wurde der Datensatz nach den Impfstoffen gruppiert und dann die zuvor von uns erstellte Spalte "DosesThisWeek" aufsummiert, wie aus dem nachfolgenden Codeblock zu entnehmen ist.

```
df_vaccine_grouped = df_tibble %>%
  group_by(Vaccine) %>%
  summarise(Total = sum(DosesThisWeek)) %>%
  arrange(desc(Total))
```

Die zweite Untersuchung betrifft die "Additional Doses" aus dem Datensatz, also alle Impfdosen, die zusätzlich nach der ersten und zweiten Impfung vergeben wurden. Auch hier soll die Gesamtanzahl visualisiert werden, einmal die Top 10 in einer Tabelle und einmal wieder als Kreisdiagramm [Siehe Fig. 3 unterer Kasten]. Durch diese beiden Visualisierungen soll herausgefunden werden, wer die meisten zusätzlichen Impfungen, also beispielsweise Auffrischungsimpfungen hatte und ob sich Unterschiede zu den insgesamt vergebene Impfdosen aus dem oberen Kasten ablesen lassen. Und tatsächlich ist eine Auffälligkeit zu erkennen: AstraZeneca ist bei den gesamten Impfdosen auf dem 4. Platz, bei den zusätzlichen Impfdosen jedoch nicht mal in den Top 10. Dies könnte darauf schließen, dass AstraZeneca so gut gewirkt hat, dass es keine Auffrischungsimpfungen gebraucht hat, jedoch ist es wahrscheinlicher, dass nachdem die Impfeempfehlung für diesen Impfstoff zurückgezogen wurde, dieser auch nicht mehr vergeben wurde. Es gab also viele erste Dosen, wo die Empfehlung noch bestehend war und danach hat es eben keine zusätzlichen Impfungen mehr gegeben.

### Länder Tab

Der "Länder"-Tab bietet einen detaillierten Einblick in die COVID-19-Impfdaten nach Ländern. Einen Screenshot davon findet man in Fig. 4.

Das Dropdown-Menü für den Zielgruppenfilter ermöglicht es dem Benutzer, die Impfdaten nach verschiedenen Zielgruppen zu filtern. Die Option "All" zeigt die Gesamtzahl der verabreichten Impfdosen unabhängig von einer spezifischen Zielgruppe an.

Die Hauptvisualisierung dieses Tabs ist ein Balkendiagramm, das die Gesamtzahl der verabreichten Impfdosen pro Land darstellt. Dabei wird der Datensatz nach Länder gruppiert und dann die Summe über die in der Datenbereinigung erstellte Spalte "DosesThisWeek" gebildet. Das Diagramm wird mithilfe von ggplot2 erstellt und zeigt die Länder auf der X-Achse und die Gesamtzahl der Impfdosen auf der Y-Achse an.

Folgender Code-Abschnitt zeigt die Filterung des Datensatzes in Abhängigkeit der Auswahl im Dropdown-Menü:

```
output$bar_chart_most_vaccinations <- renderPlot({
  # Filter data based on selected target group
  if (input$selectedTargetGroup_countries == "All") {
    # if no specific target group is selected, consider all
    # target groups
    total_doses_by_country <- df_tibble %>%
      group_by(ReportingCountry) %>%
      summarise(TotalDoses = sum(DosesThisWeek, na.rm = TRUE))
    %>%
    arrange(desc(TotalDoses))
  } else {
    # if a specific target group is selected, filter by that
    # target group
    total_doses_by_country <- df_tibble %>%
      filter(TargetGroup == input$selectedTargetGroup_countries)
    %>%
    group_by(ReportingCountry) %>%
```



```

    summarise(TotalDoses = sum(DosesThisWeek, na.rm = TRUE))
    %>%
    arrange(desc(TotalDoses))
  }

```

Der gefilterte Datensatz wird anschließend zur Visualisierung mit ggplot2 verwendet. Eine zusätzliche Tabelle [8] bietet eine tabellarische Ansicht der Gesamtzahl der verabreichten Impfdosen pro Land. Die Tabelle wird dynamisch basierend auf dem ausgewählten Zielgruppenfilter aktualisiert, was dem Benutzer ermöglicht, spezifische Details zu den Impfdaten in einer übersichtlichen tabellarischen Form zu überprüfen.

### Impfdosen Tab

Der "Impfdosen"-Tab zeigt die Abbildung der verschiedenen Impfdosengruppen über den Gesamtzeitraum, sowie deren Anzahl über den gesamten Zeitraum, aufgeteilt nach Ländern. Einen Screenshot davon findet man in Fig. 5. Zu sehen ist ein Liniendiagramm, dass den Verlauf der einzelnen Impfdosengruppen über den Zeitraum des Datensatzes darstellt. Hierfür wird der Datensatz nach *YearWeekISO* gruppiert und mittels Plotly als interaktiver Graph dargestellt. Wichtig anzumerken ist, dass die *TargetGroup* nach den Altersgruppen (AgeX\_Y) gefiltert werden musste, um Einflüsse redundanter Daten zu verhindern (ALL, HCW, ...). Die numerischen Spalten werden nun aufsummiert und die relevanten werden mittels `add_trace()` dem Plot hinzugefügt. Auch für die Tabelle musste der Datensatz nach relevanten *TargetGroups* gefiltert werden. Zudem werden nach *ReportingCountry* Gruppen gebildet und aufsummiert. Außerdem wurde ein Mapping von den Ländercodes in *ReportingCountry* zu den tatsächlichen Namen vorgenommen. Das Ergebnis wurde als `datatable` in dem Tab genutzt.

### Zielgruppen Tab

Im "Zielgruppen"-Tab liegt der Fokus auf einer detaillierten Untersuchung der Impfungen basierend auf verschiedenen Zielgruppen und Ländern (Fig. 6). Hierbei ermöglichen zwei Dropdown-Menüs dem Benutzer, sowohl das Zielpublikum als auch das spezifische Land für die Analyse auszuwählen. Aufgrund der Vielzahl möglicher Kombinationen dieser Dropdown-Auswahlmöglichkeiten wurden mehrere If-Else-Bedingungen implementiert, um sicherzustellen, dass jede mögliche Auswahl berücksichtigt und der Datensatz anschließend darauf basierend gefiltert wird.

Die visualisierten Daten werden durch einen Liniengraphen und ein Kuchendiagramm repräsentiert. Hierbei wurde das plotly-Paket verwendet. Das Problem hierbei war, dass der Plotly-Output in der App nicht sichtbar wurde. Deshalb mussten wir im UI-Teil die Funktion `plotlyOutput()` aus der Bibliothek plotly verwenden [9]. Der Liniengraph bietet einen zeitlichen Überblick über die gesamten Impfungen für die ausgewählten Zielgruppen und das entsprechende Land. Das Kuchendiagramm ergänzt dies, indem es die Verteilung der Impfstoffdosen auf verschiedene Altersgruppen zeigt. Zusammen ermöglichen diese Visualisierungen eine eingehende Analyse der Impfdaten in Bezug auf spezifizierte Zielgruppen und geografische Regionen.

Als zusätzliches Feature wurden die Überschriften der Plots dynamisch gestaltet, um sich an den getroffenen Auswahlkriterien in den Dropdowns zu orientieren. Dies ermöglicht eine präzisere und aussagekräftigere Benennung der Plots, die den Benutzern eine bessere Orientierung über die dargestellten Daten bietet.

## ”Über Uns” Tab

Der ”Über Uns”-Tab bietet einen Einblick in das Projekt und stellt die Entwickler vor, was man in Fig. 7 sehen kann. Als zusätzliches Element wurden GitHub-Logos neben den Namen der Entwickler eingefügt. Diese dienen als direkte Verweise zu den jeweiligen GitHub-Seiten der Entwickler, um Interessierten die Möglichkeit zu geben, mehr über ihre Arbeit und Beiträge zu erfahren. Zusätzlich wurde ein Link zum GitHub-Projekt eingefügt. Dies ermöglicht den Nutzern, direkt auf das GitHub-Projekt zuzugreifen und sich eingehender mit dem Code und den Entwicklungsbeiträgen vertraut zu machen.

## Bootstrap

Wir haben uns dazu entschieden, für die Front-End Visualisierung das Framework ”Bootstrap” [10] zu verwenden. Dieses lässt sich wie bereits im Abschnitt ”Dark- und Lightmode” sehr einfach in eine Shiny App einbinden und kann danach zum einen dafür benutzt werden, dass die Elemente alle einheitlich aussehen und zum anderen kann man damit ohne viel zusätzlichen Aufwand die App responsiv gestalten. Responsiv bedeutet, dass sich die Anordnung der Plots an die ihnen zu Verfügung gestellten Breite des Bildschirms anpassen. So sind beispielsweise die Plots und Tabellen auf einem Desktop nebeneinander angeordnet, doch auf einem Smartphone, dessen breite erheblich kleiner ist, sind diese untereinander angeordnet. Implementiert man dies nicht, so würden die Diagramme und Tabellen auch auf einem sehr schmalen Bildschirm nebeneinander angezeigt werden, was diese sehr gequetscht und demzufolge unleserlich macht.

Als einheitliches Design haben wir uns dazu entschieden, um Tabellen und Grafiken, die thematisch miteinander zu tun haben, eine Box mit runden Ecken zu zeichnen, um eben diese Zusammengehörigkeit zu unterstreichen. Ausgenommen davon sind der Einführungs- und ”Über uns”-Tab, da diese beiden Tabs keine Grafiken oder Tabellen besitzen.

Eines der größten aufgetretene Probleme war, dass die Tabellen von Shiny, die mit der Funktion `renderTable()` aufgebaut und mit der Funktion `tableOutput()` ausgegeben werden, nicht responsiv schön aussahen. Als mögliche Lösung haben wir das Paket ”DT” [11] verwendet, welches solche Tabellen darstellen kann, die allerdings ein responsives Design haben und zudem noch visuell schön gestaltet sind. Der Funktion `renderUI()` muss dafür nur in einem return-Statement eine DT-Tabelle zurückgegeben werden, damit man die Tabelle dann mit `uiOutput()` ausgeben kann. Wie eine solche Tabelle mithilfe eines Dataframes erstellt wird, ist im folgenden Beispiel zu sehen.

```
table <- DT::datatable(total_doses_by_country,
  style = "bootstrap4", rownames = FALSE)
return(table)
```

Ein weiteres Problem, welches bei der Visualisierung aufgetreten ist, ist dass die Logos bei dem Wechsel vom Lightmode zum Darkmode zunehmend unkenntlich werden. Wir haben überlegt, ob es eine Möglichkeit gibt, diese bei Wechsel des Modus auszutauschen, allerdings haben wir in Shiny keine Lösung dafür gefunden. Es wäre bestimmt möglich mithilfe von beispielsweise JavaScript herauszufinden, wann der Switch betätigt wurde und daraufhin das CSS anzupassen, allerdings haben wir nicht herausgefunden, wie man das in einer Shiny App einbindet und haben uns im Endeffekt dagegen entschieden, das Logo bei wechselndem Modus zu ändern.

## 2) *Verwendete R-Pakete:*

- **shiny** Version 1.7.5.1
- **shinyWidgets** Version 0.8.0
- **readr** Version 2.1.4
- **tibble** Version 3.2.1
- **plotly** Version 4.10.2
- **dplyr** Version 1.1.3
- **bslib** Version 0.5.1
- **DT** Version 0.31

## V. FAZIT UND BETEILIGUNGEN

### 1) *Alexander Metzler:*

Zu meinen Aufgaben im Projekt gehörte die Simulation der Variable "Age" während des Datenbereinigungsprozesses, der erste Entwurf und Aufbau der App mit der Auswahl geeigneter Panels für eine klare Struktur, die Implementierung von Mappings im Server-Teil für verbesserte Lesbarkeit des Codes, die Integration von Bootstrap und die Festlegung von App-Themen, insbesondere die Implementierung des Dark-Light-Mode-Switches. Des Weiteren war ich für die Umsetzung der Tabs "Länder", "Zielgruppen" und "Über Uns" verantwortlich.

Im "Länder"-Tab konzentrierte ich mich auf die Implementierung von Dropdowns und erstellte einen Bar-Chart, um die Dosen pro Land darzustellen. Im "Zielgruppen"-Tab wurden mehrere If-Else-Bedingungen berücksichtigt, um alle möglichen Dropdown-Kombinationen für eine umfassende Analyse abzudecken. Dabei kamen Plotly-Plots zum Einsatz. Der "Über Uns"-Tab erhielt zusätzlich zu den Projektinformationen eine Integration von GitHub-Logos als Referenz zu den Entwicklerprofilen, ergänzt durch einen direkten Link zum GitHub-Projekt.

In der Umsetzung des Projekts konnte ich den Großteil unserer geplanten Funktionen realisieren. Da wir leider nur wenig Zeit hatten, hätte ich noch gerne weitere Shiny-Widgets ausprobiert und mich mit weiteren Visualisierungen auseinandergesetzt.

Der Schwierigkeitsgrad des Projekts variierte. Die anfängliche Implementierung gestaltete sich herausfordernd, da ich mich zunächst in bestimmte Elemente einarbeiten musste. Nachdem das grundlegende Verständnis für die Interaktion zwischen UI und Server-Teil da war, wurde die Umsetzung deutlich leichter. Die Vielzahl verfügbarer Online-Ressourcen und Foren zur Problemlösung erwies sich als äußerst hilfreich. Meiner Meinung war der Datensatz nicht optimal, da jeder Tabelleneintrag nicht einen Patient repräsentierte, wie man es normalerweise gewohnt ist.

Während des Projekts habe ich zahlreiche neue Erkenntnisse gewonnen. Neben den Vorlesungsinhalten habe ich gelernt, das Design der App mithilfe von Bootstrap anzupassen. Zudem konnte ich den Nutzen von zusätzlichen Paketen wie ShinyWidgets entdecken, insbesondere die Integration von UI-Elementen wie dem Material Switch. Außerdem habe ich das erste Mal mit mehreren Personen an einem GitHub-Projekt gearbeitet und ich konnte lernen, wie man mit Merge-Konflikten umgeht.

Die Zusammenarbeit in der Gruppe verlief äußerst positiv, da wir uns bereits einige Zeit kennen. Die Verwendung von Git zur Versionsverwaltung und das Projektmanagement mit dem Backlog von GitHub ermöglichten eine klare Übersicht über die Aktivitäten aller Teammitglieder. Die Absprachen waren effektiv, und die Aufgabenverteilung funktionierte reibungslos.

## 2) Yannik Krantz:

Zu meinen Aufgaben gehörte die Datenbereinigung, das gesamte weitere bearbeiten des Datensatzes und die Umsetzung des Tabs "Impfdosen".

Probleme, auf die ich während des Projekts stieß, waren hauptsächlich in der Bearbeitung des Datensatzes, sowie meiner ursprünglichen Aufgabe zur Umsetzung eines Tabs für die Ablehnungsrate. Es stellte sich leider häufig heraus, dass der Datensatz Eigenschaften aufwies, die unsere ursprünglichen Pläne erschwerten. Zum einen waren gewisse TargetGroups, für die weiteren Zeilen und für Tabellen/Grafiken irrelevant. Zum anderen stellte sich die Spalte *FirstDoseRefused*, auf der die Ablehnungsrate basiert hätte, als unbrauchbar heraus.

Letzten Endes habe ich viel über das Anpassen und Analysieren von Datensätzen in R gelernt. Eine sehr wichtige Lehre, die ich aus diesem Projekt mitgenommen habe, ist dass ich den Datensatz sehr viel genauer prüfe, bevor ich Änderungen vornehme oder Pläne für abhängige Teile der Applikation mache. Vor allem werde ich künftig Spalten auf Auffälligkeiten, wie das Fehlen einer großen Menge an Einträgen, prüfen. Auch über den Nutzen eines gemeinsamen Versionierungssystems in einer Teamarbeit habe ich gelernt. Die Möglichkeit unabhängig voneinander die eigenen Aufgaben an der App zu bearbeiten und unsere Änderungen in kurzen "Meetings" zusammenzutragen hat die Teamarbeit sehr angenehm gemacht.

## 3) Florian Max Hauptmann:

Meine Hauptaufgabe in diesem Projekt war die einheitliche Visualisierung der einzelnen Tabs mithilfe von Bootstrap, da ich schon einmal in einem früheren Projekt damit zu tun hatte. Zudem war ich noch für den Einleitungs- und Impfstoff-Tab verantwortlich.

Der Impfstoffe-Tab wies keine besonderen Probleme auf, da hier keine wirkliche Interaktion mit dem Benutzer möglich war, das einzige war hier einen geeigneten Weg zu finden, wie die Tabellen und Grafiken angeordnet werden, wenn der Bildschirm sich verkleinert. Generell lag die größte Schwierigkeit bei der Visualisierung eher in den Details, da ich viel probieren musste, um die richtige CSS-Konfiguration für die Grafiken zu finden.

Das größte Problem lag wie bereits beschrieben in den Tabellen, welches zum Glück jedoch mithilfe des "DT"-Paketes behoben werden konnte. Und eben das Problem der unleserlichen Logos im Darkmode, welches ich nicht lösen konnte. Als Lösungsansatz hatte ich aber die Idee es mit JavaScript umzusetzen, allerdings konnte ich dies nicht in Shiny einbinde und diese Idee wäre also eine Möglichkeit das Projekt noch zu erweitern, sofern man das möchte.

Während dieses Projektes habe ich viel über sowohl R als auch Shiny gelernt und dazu nochmals mein Wissen über Bootstrap aufgefrischt. Zudem hat mir auch die Verwendung von GitHub extrem geholfen, mich in Versionierungssysteme einzuarbeiten was unsere Projektarbeit und auch die Zusammenarbeit im Team sehr voran gebracht hat. Allgemein verlief die Zusammenarbeit im Team reibungslos, da jeder seine Aufgaben erledigt hat und wir uns bei Probleme immer gegenseitig geholfen haben.

## VI. ANHANG

## COVID-19 Statistik Anwendung

Dark ☐
[Einführung](#) [Impfstoffe](#) [Länder](#) [Impfdosen](#) [Zielgruppen](#) [Über uns](#)

## Einführung zu COVID-19

Kurzinformation

Die Krankheit COVID-19, Kurzform für „Corona Virus Disease 2019“, ist eine Infektionskrankheit, welche durch den Erreger SARS-CoV-2 verursacht wird. Dieses Virus war Anfang 2020 der Auslöser der COVID-19-Pandemie. Der Hauptübertragungsweg des Virus ist die Tröpfcheninfektion, also über kleine Aerosole, die eingeatmet werden. Zu den häufigsten Symptomen zählen Husten, Fieber, Schnupfen und die Störung des Geschmacks- und Geruchssinns. Besonders risikobehaftet sind vor allem ältere Personen und Personen mit Vorerkrankungen. Die mediane Inkubationszeit beträgt in etwa 4-6 Tage, je nach Virusvariante und die Krankheit dauert etwa 8-10 Tage an, falls keine Therapie durchgeführt wird.

Übertragungswege

Der Hauptübertragungsweg des COVID-19 Erregers ist die Tröpfcheninfektion. Dabei werden kleine Partikel in der Luft, welche die Viren enthalten, eingeatmet und gelangen so in den Körper. Die Übertragungswahrscheinlichkeit von einer Infizierten Person zu einer Nichtinfizierten steigt dabei, wenn sich beide Personen in einem geschlossenen Raum aufhalten, besonders wenn dieser schlecht belüftet ist. Es empfiehlt sich demnach bei längerem Aufenthalt in einem Raum regelmäßig zu lüften. Auch das Tragen von Masken wie einem Mund-Nasen-Schutz können das Risiko einer Infektion senken.

Fig. 1. Tab Einführung Lightmode

## COVID-19 Statistik Anwendung

Dark ☒
[Einführung](#) [Impfstoffe](#) [Länder](#) [Impfdosen](#) [Zielgruppen](#) [Über uns](#)

## Einführung zu COVID-19

Kurzinformation

Die Krankheit COVID-19, Kurzform für „Corona Virus Disease 2019“, ist eine Infektionskrankheit, welche durch den Erreger SARS-CoV-2 verursacht wird. Dieses Virus war Anfang 2020 der Auslöser der COVID-19-Pandemie. Der Hauptübertragungsweg des Virus ist die Tröpfcheninfektion, also über kleine Aerosole, die eingeatmet werden. Zu den häufigsten Symptomen zählen Husten, Fieber, Schnupfen und die Störung des Geschmacks- und Geruchssinns. Besonders risikobehaftet sind vor allem ältere Personen und Personen mit Vorerkrankungen. Die mediane Inkubationszeit beträgt in etwa 4-6 Tage, je nach Virusvariante und die Krankheit dauert etwa 8-10 Tage an, falls keine Therapie durchgeführt wird.

Übertragungswege

Der Hauptübertragungsweg des COVID-19 Erregers ist die Tröpfcheninfektion. Dabei werden kleine Partikel in der Luft, welche die Viren enthalten, eingeatmet und gelangen so in den Körper. Die Übertragungswahrscheinlichkeit von einer Infizierten Person zu einer Nichtinfizierten steigt dabei, wenn sich beide Personen in einem geschlossenen Raum aufhalten, besonders wenn dieser schlecht belüftet ist. Es empfiehlt sich demnach bei längerem Aufenthalt in einem Raum regelmäßig zu lüften. Auch das Tragen von Masken wie einem Mund-Nasen-Schutz können das Risiko einer Infektion senken.

Fig. 2. Tab Einführung Darkmode

# COVID-19 Statistik Anwendung

Dark ☐



Einführung **Impfstoffe** Länder Impfdosen Zielgruppen Über uns

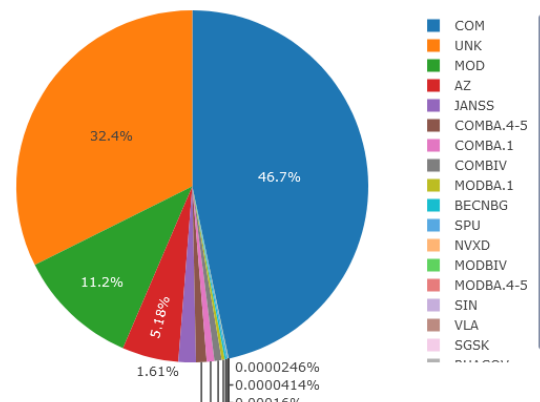
## Vergebene Impfstoffe

In der Tabelle werden die 10 am häufigsten vergebenen Impfstoffe aufgelistet und in dem Kuchendiagramm wird die prozentuale Verteilung der Impfstoffe dargestellt.

### TOP 10 der insgesamt vergebenen Impfstoffe

Impfstoff	Total
COM	1239032665.00
UNK	859032670.00
MOD	296831145.00
AZ	137461666.00
JANSS	42640122.00
COMBA.4-5	25847372.00
COMBA.1	18715298.00
COMBIV	17650211.00
MODBA.1	7355649.00
BECNBG	4674058.00

### Anteil der Impfstoffe an den gesamt vergebenen Impfungen



In der Tabelle werden die 10 Impfstoffe mit den meisten zusätzlichen Dosen aufgelistet und in dem Kuchendiagramm wird diese Verteilung in nochmals grafisch dargestellt.

### TOP 10 der zusätzlichen Dosen der Impfstoffe

Impfstoff	Total
COM	343252186.00
UNK	290392803.00
MOD	158700677.00
COMBA.4-5	25589114.00
COMBA.1	18628333.00
COMBIV	17649756.00
MODBA.1	7348202.00
JANSS	1509337.00
MODBIV	523219.00
BECNBG	224870.00

### Verhältnis der zusätzlichem Impfdosen der Impfstoffe

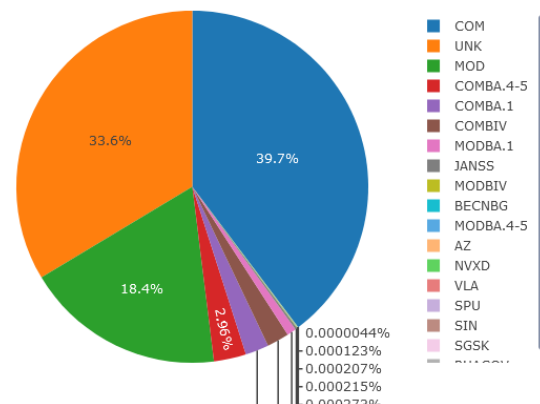


Fig. 3. Tab Impfstoffe

# COVID-19 Statistik Anwendung

Dark ☐



Einführung Impfstoffe **Länder** Impfdosen Zielgruppen Über uns

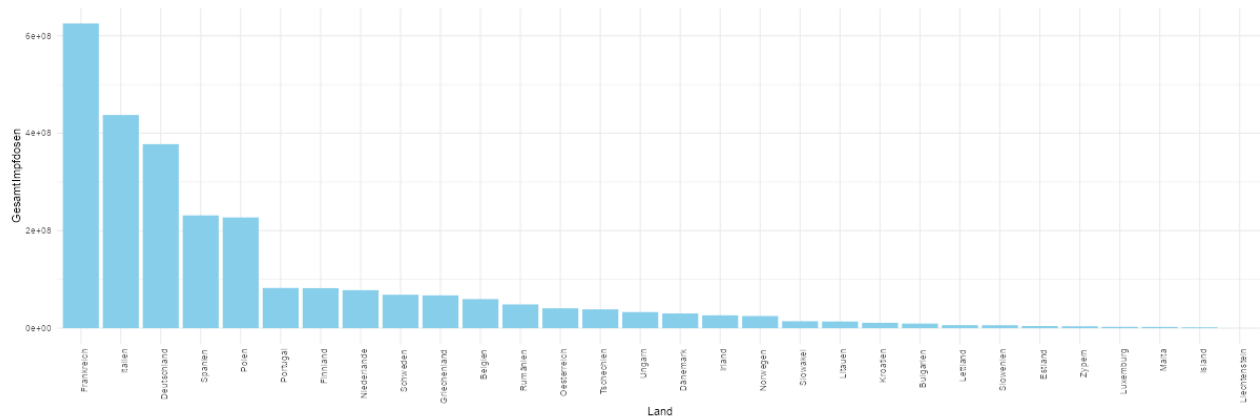
## Länderbezogene Impfstoffvergabe

Hier können Sie eine Altersgruppe auswählen, für welche dann die Gesamtanzahl der verabreichten Impfdosen im jeweiligen Land angezeigt wird.

Wähle eine Zielgruppe aus:

All

### Gesamtdosen pro Land



### Länder mit den meisten Impfdosen

Show 10 entries

Search:

Land	GesamtImpfdosen
Frankreich	625284052
Italien	437424372
Deutschland	377599750
Spanien	231305324
Polen	227128920
Portugal	82437789
Finnland	82078487
Niederlande	77943165
Schweden	68527263
Griechenland	67241814

Showing 1 to 10 of 30 entries

Previous 1 2 3 Next

Fig. 4. Tab Länder

# COVID-19 Statistik Anwendung

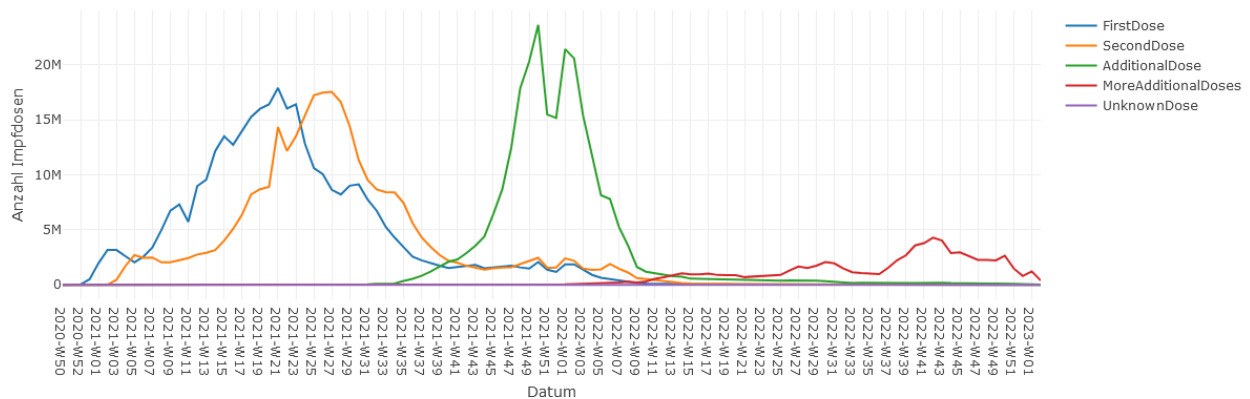
Dark ☐

Einführung Impfstoffe Länder **Impfdosen** Zielgruppen Über uns

## Verschiedene Impfdosenarten

In diesem Liniengraphen wird die zeitliche Vergabe der ersten und zweiten Impfdosis, sowie den zusätzlichen Impfdosen abgebildet.

### Alle Impfdosengruppen abgebildet über die Zeit



In dieser Tabelle werden die verschiedenen Impfdosen pro Land dargestellt. Es wird dabei zwischen der ersten und zweiten, der ersten zusätzlichen Impfdosis und noch allen weiteren zusätzlichen Impfdosen unterschieden.

### Impfdosen pro Land

Show  entries

Search:

Land	Erste Dosis	Zweite Dosis	Zusätzliche Dosis	Weitere zusätzliche Impfdosen
Belgien	8504362	8001291	7131470	4369232
Bulgarien	2074776	1534295	827248	111595
Dänemark	4738040	4657940	3607928	1910186
Estland	796812	715972	479464	106916
Finnland	13590367	13053701	9230985	5164172
Frankreich	100087560	96054710	79992790	22073070
Griechenland	7915031	7144978	5902357	1170887
Irland	4106647	3820142	3045691	1505110
Island	307206	243670	254133	63830
Italien	46244009	43320332	42961730	6755114

Showing 1 to 10 of 29 entries

Previous **1** 2 3 Next

Fig. 5. Tab Impfdosen



# COVID-19 Statistik Anwendung

Dark ☐



Einführung Impfstoffe Länder Impfdosen Zielgruppen Über uns

## Verschiedene Zielgruppen

In diesen beiden Dropdown Menüs können Sie ein Land, sowie eine Altersgruppe auswählen. Im nachfolgenden Liniendiagramm wird dann die Summe der Impfdosen in einer Woche über eine Zeitspanne dargestellt. Im unteren Kuchendiagramm wird dann angezeigt, wie groß der Anteil an verabreichten Impfdosen für eine Altersgruppe in einem zuvor ausgewählten Land ist.

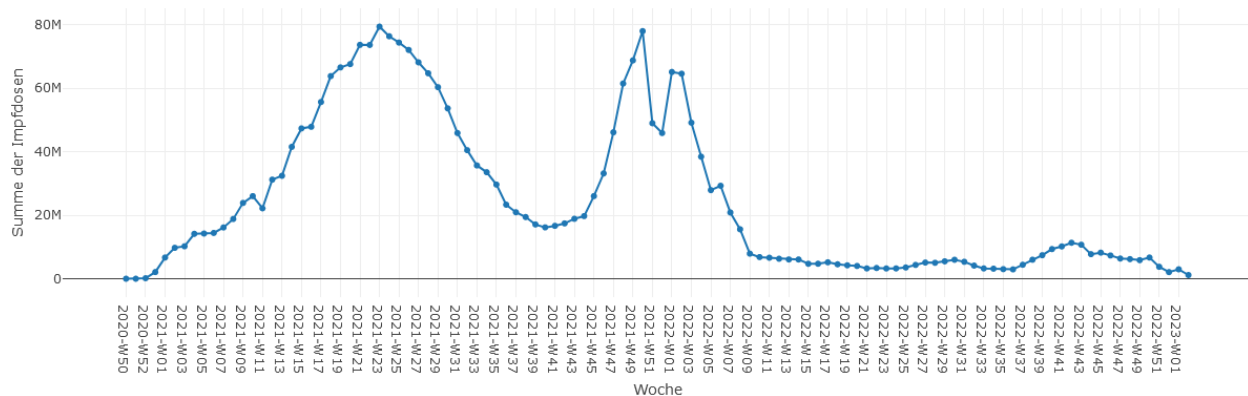
Wähle ein Land aus:

All

Wähle eine Zielgruppe aus:

All

Impfdosen über Zeit in allen Ländern und allen Altersgruppen



Verteilung der Impfdosen nach Altersgruppen

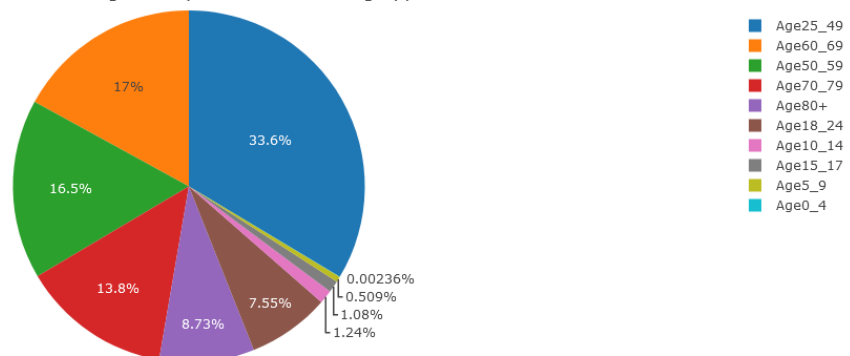


Fig. 6. Tab Zielgruppen

# COVID-19 Statistik Anwendung

Dark ☐



Einführung Impfstoffe Länder Impfdosen Zielgruppen **Über uns**

## Über unsere App

Willkommen zur "Covid App", entwickelt von Studierenden im fünften Fachsemester des Studiengangs [Data Science in der Medizin](#) an der [Technischen Hochschule Ulm](#). Diese App entstand im Rahmen unserer Projektarbeit im Wahlpflichtfach "Einführung in R und Shiny Apps" und dient gleichzeitig als Prüfungsleistung.

## Funktionen der App

Die "Covid App" bietet eine intuitive Benutzeroberfläche, die es ermöglicht, Daten im Zusammenhang mit COVID-19-Impfungen in den Ländern der Europäischen Union und des Europäischen Wirtschaftsraums zu visualisieren. Die App präsentiert umfassende Informationen über die Verteilung von den verschiedenen Impfstoffen, den Impfdosen und Verabreichungsraten.

## Weitere Informationen

Besuchen Sie diese Website für weitere Informationen über den Datensatz: [Hier klicken](#)

## Entwickler

Yannik Krantz 

Alexander Metzler 

Florian Hauptmann 

## GitHub Project

R Shiny Covid App 

Fig. 7. Tab About

## REFERENCES

- [1] TERMDAT - Die Terminologiedatenbank der Bundesverwaltung. Accessed: 2024.01.06. [Online]. Available: <https://www.termdat.bk.admin.ch/search/entry/477401>
- [2] Robert Koch Institut. Accessed: 2024.01.06. [Online]. Available: [https://www.rki.de/DE/Content/InfAZ/N/Neuartiges\\_Coronavirus/Steckbrief.html](https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html)
- [3] Action layout. Accessed: 2024.01.14. [Online]. Available: <https://mastering-shiny.org/action-layout.html>
- [4] Bootswatch. Accessed: 2024.01.13. [Online]. Available: <https://bootswatch.com/>
- [5] Shiny widgets. Accessed: 2024.01.14. [Online]. Available: <https://dreamrs.github.io/shinyWidgets/>
- [6] Stackoverflow dark mode switch. Accessed: 2024.01.14. [Online]. Available: <https://stackoverflow.com/questions/70095322/dark-mode-switch-doesnt-work-using-navbarpage-shiny>
- [7] Rstudio github theming. Accessed: 2024.01.14. [Online]. Available: <https://rstudio.github.io/bslib/articles/theming/#dynamic>
- [8] Render table. Accessed: 2024.01.13. [Online]. Available: <https://shiny.posit.co/r/reference/shiny/latest/rendertable>
- [9] Stackoverflow plotly problem. Accessed: 2024.01.06. [Online]. Available: <https://stackoverflow.com/questions/41255810/r-shinyapp-not-showing-plot-ly-in-browser-but-show-only-graph-in-viewer-pane>
- [10] Bootstrap. Accessed: 2024.01.13. [Online]. Available: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [11] DT Tables. Accessed: 2024.01.13. [Online]. Available: <https://rstudio.github.io/DT/>