



Hochschule  
Kaiserslautern  
University of  
Applied Sciences

Informatik und  
Mikrosystemtechnik  
Zweibrücken

## Bericht Projekt MTI/SE SS 20

Master-Studiengang (PO Version 2018)

**Informatik**

**Schwerpunkt: Mensch-Technik Interaktion**

Titel:

**CODA19: Entwicklung eines Shiny Dashboards in R  
mit Hilfe von Covid-19 Daten**

CODA19: Development of a Shiny Dashboard in R with the help of Covid-19 data

vorgelegt von: Marco Miles Noll  
Julian Bernhart  
Jens Cedric Schug

vorgelegt am: 26. August 2020

vorgelegt bei: Prof. Dr. Manfred Brill

durchgeführt bei: Hochschule Kaiserslautern  
Amerikastraße 1  
66482 Zweibrücken  
Deutschland

## Inhalt

Motivation und Einführung .....	1
Ordnerstruktur und Design vom Dashboard .....	3
Automatisierung .....	5
Problematik während der Entwicklung .....	6
Temporary Working Directory: .....	6
Verwendung von Makefiles in RStudio und Problembehandlung .....	6
Pandoc .....	7
Erläuterung der Daten .....	8
Praktische Umsetzung des Dashboards .....	9
Über Shiny .....	9
Grober Aufbau der Gesamtapp .....	9
DashboardSidebar .....	9
DashboardBody .....	9
Weltkarte .....	9
Statistik .....	10
Forecasts .....	11
Glossar .....	13
Quellen und Impressum .....	14
Ausblick und Aufgabenverteilung .....	15
Fazit .....	16
Quellenverzeichnis .....	17

## Motivation und Einführung

Seit Anfang des Jahres 2020 beeinträchtigt das SARS-CoV-2 und die damit einhergehende Pandemie die Weltbevölkerung. Daraus resultierten verheerende Einschränkungen in verschiedensten Bereichen des Alltags, sowie systemrelevanten und kritischen Infrastrukturen. Um den Überblick über das Infektionsgeschehen zu behalten, ist es unabdingbar, kontinuierlich epidemiologische Kenngrößen zu erfassen, anhand derer politische Entscheidungen und Maßnahmen getroffen werden, um den Auswirkungen der Pandemie entgegen zu wirken. Es existieren weltweit verschiedene Institute zur Erfassung von epidemiologischen Kennzahlen. Beispiele hierfür sind die WHO oder das Robert-Koch-Institut in Deutschland. Diese sind zur Beobachtung des Auftretens von Krankheiten und relevanter Gesundheitsgefahren in der Bevölkerung, sowie dem Ableiten und wissenschaftliche Begründen der erforderlichen Maßnahmen zum wirkungsvollen Schutz der Gesundheit der Bevölkerung zuständig.

Diese Ausarbeitung ist an Studenten gerichtet, die das Projekt weiterführen wollen. Daher werden im Folgenden Aufbau und Umsetzung des Projekts dargestellt, um einen einfacheren Einstieg in das Projekt zu gewährleisten. Zudem werden aufgetretene Probleme, mögliche Lösungen, eine Retrospektive und Vorschläge zur Weiterentwicklung beschrieben. Empfehlenswert sind Vorkenntnisse in der Programmiersprache R, sowie Grundkenntnisse in Statistik und Data Science.

Nach dem Vergleich mehrerer Dashboards während der Recherchephase wurde festgestellt, dass die bisher existierenden Dashboards größtenteils unübersichtlich, nicht interaktiv und keine Beschreibungen bezüglich des Inhalts enthalten. Aufgrund dieser Erkenntnis wurde als Ziel für das Projekt ein übersichtliches, interaktives Dashboard inklusive Glossar zu epidemiologischen Fachbegriffen gewählt.

Als Datengrundlage für das Projekt dient ein Datensatz von DataHub.io, die eine Zusammenfassung der Daten zur Covid-19-Pandemie aus verschiedenen Quellen als CSV-Dateien bereitstellen. Zu finden ist dieser Datensatz unter [\[1\]](#).

Die folgende Tabelle zeigt eine Übersicht über die im Projekt verwendeten Technologien.

Verwendete Technologie	Version/Anmerkung
R Programmiersprache	Debug-Versionen: 3.6.1, 3.6.3, 4.0.0, 4.0.2 Release: 3.6.1  Anmerkung: Eine Empfehlung für die Vertiefung der R-Kenntnisse, Link zum Buch
RStudio	Debug-Versionen: 1.2.5001, 1.2.5033, 1.3.1056 Release: 1.2.5001
R Shiny-Package	Debug-/Release-Version: 1.4.0.2  Anmerkung: Zum Einstieg in das Erstellen von Shiny-Dashboards wurde folgende Ressourcen gewählt: <ul style="list-style-type: none"><li>• <a href="https://shiny.rstudio.com/tutorial/written-tutorial/lesson1/">https://shiny.rstudio.com/tutorial/written-tutorial/lesson1/</a></li><li>• <a href="https://rstudio.com/wp-content/uploads/2015/02/shiny-cheatsheet.pdf">https://rstudio.com/wp-content/uploads/2015/02/shiny-cheatsheet.pdf</a></li></ul>
R Shiny Dashboard	Debug-/Release-Version: 0.7.1
R Leaflet-Package	Debug-/Release-Version: 2.0.3
Batchfile	
Betriebssysteme	Debug-/Release-Version: Windows 10 Home Windows 10 Pro

Das Projekt findet man unter [\[2\]](#) oder :



## Ordnerstruktur und Design vom Dashboard

Für die Projektstruktur wurde festgelegt, Ressourcen und Logik zu trennen. Alle Daten, die für das Dashboard verwendet werden, befinden sich zentral in dem data-Ordner. Dieser beinhaltet die Daten zum Verlauf der Corona-Pandemie und die Einträge für das Glossar.

Das Dashboard des Projekts wurde im Hinblick auf die Benutzerfreundlichkeit und die Übersichtlichkeit der Daten umgesetzt. Dem Nutzer sollen möglichst viele Informationen sinngebend dargestellt werden, dabei wurde sich auf die wichtigsten, in diesem Kontext befindlichen Kenngrößen beschränkt. Die Inhalte wurden vollständig in der englischen Sprache verfasst, um potenziell eine größere Zielgruppe zu erreichen.

Das Dashboard ist in folgende zwei Bereiche aufgeteilt. Im linken Bereich befindet sich die Navigationsleiste, welche das Menü repräsentiert. Im rechten Teil des Dashboards wird der Inhalt angezeigt, welcher sich je nach Kontext automatisch anpasst. Die Größenverhältnisse von Navigationsleiste und Inhaltsanzeige beträgt circa eins zu zehn, welche sich beim Anklicken eines Icons in der Kopfleiste der Seite auch ändern lässt. Dabei klappt die Navigationsleiste zur Seite und der Inhalt wird auf der gesamten Fläche vollständig dargestellt. Bei der Navigationsleiste wurde sich auf verschiedene Oberkapitel geeinigt. Diese Kapitel wurden anhand ihrer Priorität in absteigender Reihenfolge gelistet.

Als ersten Menüpunkt wird eine Weltkarte dargestellt. Diese **Worldmap** zeigt die bestätigten, kumulierten Covid-19-Erkrankten in Abhängigkeit des Landes, in Form von Kreisen an. Die Größe der Kreise richtet sich nach der Anzahl der bestätigten, erkrankten Covid-19-Fälle (Confirmed Cases). Aus Gründen der Darstellung und Übersichtlichkeit musste durch einen logarithmischen Faktor die Darstellung korrigiert werden. Möchte man nähere Details zu einem spezifischen Land erfahren, kann man durch das Anklicken des Kreises für das jeweilige Land einen Tooltip aufrufen. Unter der Weltkarte befinden sich zusätzlich vier Anzeigen, die Zusatzinformationen zur globalen Datenlage liefern. Dazu zählen die **Global Active Cases**, die Anzahl an momentan aktiv infizierten Personen, die **Global Infected**, die Anzahl der Personen die insgesamt infiziert waren (geheilte und aktiv Infizierte), die **Global Recovered**, die Personen die keine Anzeichen einer Infektion mehr zeigen und die **Global Deceased**, die Personen die an Covid-19 gestorben sind. Die Daten passen sich automatisch anhand eines Zeitstrahls in der Navigationsleiste an. Der Zeitstrahl beginnt ab dem **22. Januar 2020** und endet meistens ein bis zwei Tage vor dem aktuellen Datum, bedingt durch die Datenquellen und Datenerhebung. Der Zeitstrahl wird solange aktualisiert, wie die Erhebung der Covid-19 Daten weitergeführt wird.

Im Folgenden wurde sich statistisch mit den Covid-19 Daten auseinandergesetzt. Unter **Statistic** im Navigationsmenü kann auf diese Daten zugegriffen werden. Der Statistikteil benutzt denselben Zeitraum des Zeitstrahls in der Navigationsleiste. Zusätzlich ist die Statistik länderspezifisch. In der Navigationsleiste kann mit Hilfe einer Dropdown-Liste oder Eingabe eines Namens ein Land ausgewählt werden. Darüber hinaus ist es auch möglich bei einigen Ländern die Provinzen oder Staaten aufzurufen (z.b. China) um genauere Details zu erhalten. Ein Liniendiagramm zeigt die quantitativen Entwicklungsverläufe der Infizierten (rot), der Geheilten (grün), der an Covid-19 Verstorbenen (schwarz) und der aktiven Fälle (blau). Die Farben wurden symbolisch für den jeweils geltenden Zustand gewählt. Das andere Liniendiagramm beschreibt **epidemiologische Maßzahlen** und gibt sie visuell als Liniengraph aus. Dabei handelt es sich um die **Prävalenz** (Anzahl der Neuerkrankten), die Sterblichkeitsrate (**all case mortality**) und die fallbezogene Letalitätsrate (**case fatality rate**). Analog zur Weltkarte werden hier auf die aktiven Fälle, die Gesamtanzahl der

Infizierten, die Personen die keine Anzeichen einer Infektion mehr zeigen und die Personen die an Covid-19 gestorben sind angezeigt.

Das Dashboard ist in der Lage potenzielle **Vorhersagen** über den Verlauf der Covid-19 Maßzahlen für ein Land treffen zu können. Dabei wird in verschiedenen Graphen der bisherige Verlauf analysiert und anhand dessen eine Vorhersage (prediction) getroffen. Da Aussagen über die Zukunft unpräzise sind, werden darüber hinaus zwei Konfidenzintervalle von **80** und **95 %** zusätzlich angegeben. Ein Bericht über die Vorhersagen lässt sich durch den **Save Forecasts**-Button herunterladen. Dies dient vor allem dem Vergleich der tatsächlichen Werte mit den vorhergesagten Werten und der Dokumentation.

In einer tabellarischen Übersicht werden die wichtigsten Begriffe, Kennzahlen und Informationen über das Thema Covid-19 vermittelt.

Unter **Sources** findet man die verwendeten Datenquelle für das Dashboard, die Visualisierungen und das Glossar.

Im **Imprint**-Bereich sind Kontaktdaten zu den Erstellern und des Betreuers des Covid-19-Dashboards hinterlegt.

## Automatisierung

Ziel war es, für das Projekt eine **Datenpipeline** zu erstellen, mit deren Hilfe ein Dashboard automatisiert auf Grundlage der zur Verfügung stehenden Daten generiert werden kann. Hierbei sollen über diese Pipeline folgende Aufgaben in gegebener Reihenfolge erfüllt werden:

- Zunächst soll das **Working Directory** für das R Environment festgelegt werden.
- Die benötigten **Daten werden heruntergeladen** und falls notwendig wird zuvor ein entsprechender Ordner angelegt. Der Nutzer wird gefragt, ob der Ordner angelegt werden soll, bei einer Verneinung wird der Prozess abgebrochen.
- Die Daten werden für die Darstellung auf dem Dashboard **aufbereitet**.
- Das **Dashboard** wird in Form einer **Website** erstellt.
- Der **Server** für die Website wird gestartet.

Die einzelnen Schritte, die zum Erstellen des Dashboards notwendig sind, bauen aufeinander auf. Das bedeutet, sobald sich eine Datei an einer Stelle ändert, muss der gesamte Prozess erneut ausgeführt werden, um das Dashboard auf den aktuellen Stand zu bringen.

Der erste Lösungsansatz war die Umsetzung mit einem **Makefile**. RStudio bietet die Möglichkeit ein Makefile für das **Kompilieren** der einzelnen Quellcodesegmente zu nutzen. Während des Projekts sind jedoch Schwierigkeiten aufgetreten, die in dessen Rahmen nicht behoben werden konnten, weshalb aufgrund der fortgeschrittenen Zeit eine andere Lösung genutzt wurde. Die Nutzung eines Makefiles, die entsprechende Anpassung von RStudio, die aufgetretenen Probleme und entsprechende Lösungsansätze werden im nächsten Kapitel beschrieben.

Auf Grund der fortgeschrittenen Zeit im Projekt und der Komplexität des Problems wurde sich auf eine andere, ersatzweise Lösungsmöglichkeit geeinigt, die in vorangegangenen Projekten erfolgreich verwendet wurde.

Aktuell ist eine **Batch-Datei** die Lösung, mit der die **Pipeline** umgesetzt wurde. Dieses koordiniert die Nutzung der verschiedenen **R-Skripte**, sodass es mit dem Ausführen der Datei möglich ist das Dashboard zu erstellen und den entsprechenden Server zu starten.

Damit die **Run.bat** wie vorgesehen funktioniert, müssen sich die notwendigen Skripte auf derselben Ebene im Projekt befinden. Zudem muss in der Datei der die Pfadangabe zur **Rscript.exe** in Zeile 4 angepasst werden.

## Problematik während der Entwicklung

### Temporary Working Directory:

Dieses Problem betraf sowohl die Nutzung des **Makefiles** als auch die des **Batchfiles**. Dieses **Temporary Working Directory**, im Folgenden als **TMPDIR** bezeichnet, wird von R immer dann benötigt, wenn Dateien heruntergeladen oder extrahiert und nach der Verwendung entfernt werden sollen. Es stellt, wie der Name schon sagt, einen temporären Zwischenspeicher für Daten dar. Für das Projekt wurden zwei Rechner mit Windows 10 Home und einer mit Windows 10 Pro als Betriebssystem verwendet. Die hier beschriebene Problematik betraf eine der Maschinen mit Windows 10 Home.

Die Problematik bestand darin, dass es zu Beginn des Projekts, nach einem **Windows Update**, der Maschine nicht mehr möglich war das TMPDIR zu finden. Somit war beispielsweise das Inkludieren von R-Packages nicht möglich. Dies lässt sich normaler Weise mit einer Anpassung der Umgebungsvariablen für R beheben. Das beschriebene Vorgehen wird im Anschluss an die Problembeschreibung erläutert (In diesem Fall konnte das Problem jedoch nicht behoben werden). Das TMPDIR konnte nicht verwendet werden und im Verlauf des Projekts sind weitere Probleme aufgetreten, für die keine Lösung gefunden wurde. Beispielsweise wurden durch den Versuch R-Skripte auszuführen Packages und DLL-Dateien aus der bestehenden R-Installation gelöscht, auch sind teilweise die IDE oder der Rechner abgestürzt. Nach mehrfacher Neuinstallation, der für das Projekt notwendigen Software und dem Anpassen der Umgebungsvariablen, wurde durch RStudio keinerlei Fehlermeldungen bezüglich des TMPDIR oder fehlender Dateien mehr ausgegeben, obwohl weder die IDE noch R wie intendiert funktionierten. Auch durch den Befehl **tempdir** auf Konsole wurde das richtige Verzeichnis ausgegeben. Trotzdem konnte keine Funktion festgestellt werden. Das Testen oder Debuggen der R-Skripte war somit auf dem betroffenen Rechner nicht möglich. Nach weiteren **Windows Updates** und einer **erneuten Installation von RStudio** und der neusten **R-Version (4.0.2)** konnte, nach Anpassung der Umgebungsvariablen, die Software wieder ohne Einschränkungen genutzt werden. Allerdings konnten während des Projekts weder die genauen Ursachen für die Probleme noch die Lösungen identifiziert werden. Zudem liegen nicht ausreichend Informationen vor, um die Problematik zu reproduzieren. Die verschiedenen Versionen des Batchfiles und der R-Skripte mussten somit auf den beiden verbleibenden Maschinen getestet und debugged werden.

### Verwendung von Makefiles in RStudio und Problembehandlung

Im Folgenden wird das Einrichten von RStudio für die Nutzung von Makefiles und das Anpassen der zuvor erwähnten **Umgebungsvariablen** beschrieben.

Ein Makefile erlaubt es, über RStudio plattformunabhängig die Skripte zu kompilieren und auf verschiedenen Betriebssystemen dasselbe Dashboard automatisiert zu erstellen.

Um Makefiles in RStudio verwenden zu können, muss die IDE entsprechend angepasst werden. Eine Anleitung hierzu findet sich unter [\[3\]](#). Zuvor muss jedoch das Rtools-Package für Windows installiert werden. RStudio benötigt das **RTools-Package** um das Build Command make nutzen zu können. Eine Downloadmöglichkeit und die Anleitung zur Installation befinden sich unter [\[4\]](#).



Es gibt mehrere Möglichkeiten, um die besprochenen Umgebungsvariablen anzupassen. Die Abbildung aus [5] zeigt eine Übersicht zu den entsprechenden Dateien:

File	Who Controls	Level	Limitations
.Rprofile	User or Admin	User or Project	None, sourced as R code.
.Renviron	User or Admin	User or Project	Set environment variables only.
Rprofile.site	Admin	Version of R	None, sourced as R code.
Renviron.site	Admin	Version of R	Set environment variables only.
rsession.conf	Admin	Server	Only RStudio settings, only single repository.
repos.conf	Admin	Server	Only for setting repositories.

Die Lösung die im Projekt abschließend funktioniert hat, betrifft die Datei .Renviron, die sich meist im Dokumente-Ordner befindet. Sollte sich die Datei nicht in dem Ordner befinden, kann sie mit Hilfe der **Powershell** (Shift + Rechtsklick im Ordner) und der Eingabe **Add-Content c:\Users\\$env:USERNAME\Documents\.Renviron {KEY}="{VALUE}"** erstellt werden. Der Inhalt der Datei ist wie folgt aufgebaut:

```
PATH="{RTOOLS40_HOME}\usr\bin;${PATH}"
TMPDIR="{R-4.0.2_HOME}\library"
TMP="{R-4.0.2_HOME}\library"
TEMP="{R-4.0.2_HOME}\library"
```

Zudem war es notwendig das RTools-Package zu den Pfadvariablen von Windows hinzuzufügen. Die Angaben zu TMPDIR, TMP und TEMP legen das Temporary Working Directory fest und sollten dementsprechend denselben Pfad beinhalten. Nach dem Neustart von RStudio kann das TMPDIR mit der Methode tempdir() abgefragt werden. Die folgende Abbildung zeigt ein Beispiel des Aufrufs der Methode auf der Konsole von RStudio. Die Ordner für das Hinterlegen der Daten werden automatisch angelegt.

```
> tempdir()
[1] "D:\\R\\R-4.0.2\\library\\RtmpsHwLbH"
> |
```

## Pandoc

Für das Erstellen des Forecast-Reports benötigt RStudio Pandoc. Aus diesem Grund ist es wichtig Pandoc in den **Umgebungsvariablen** als **Pfad-Variable** des jeweiligen Betriebssystems zu setzen. Pandoc befindet sich meistens unter:

```
C:\Program Files\RStudio\bin\pandoc
```

## Erläuterung der Daten

In der App werden mehrere Dataframes benutzt, die sich gegenseitig erweitern und in dem R-Skript **prepareData** erzeugt werden. Das Dataframe **data\_from\_github** enthält alle benötigten Datensätze und wird vor allem für die **Weltkartendarstellung** durch das R-Package **leaflet** genutzt. Die dafür benötigten Daten befinden sich in der „**time-series-19-covid-combined.csv**“. Dem Dataframe wurde auch eine Spalte für die logarithmische Skalierung hinzugefügt, um die Größe der Kreise auf der Weltkarte zu vereinheitlichen. Des Weiteren wird im Dataframe **population\_data** die **Anzahl der Bevölkerung nach Ländern und Provinzen** angegeben. Im späteren Verlauf des R-Skripts findet ein **left-join** zwischen den Dataframes **data\_from\_github** und **population\_data** statt. Nach diesem Vorgang wird die Anzahl der aktiven Fälle (Aktive Fälle: = Anzahl v. Infizierten – Anzahl v. Todesfällen – Anzahl v. Genesenen) dem Dataframe **data\_from\_github** beigefügt. Der Datenstruktur werden in **drei Spalten** die **epidemiologischen Kennzahlen** (Prävalenz, Gesamtmortalität und Letalität) berechnet und hinzugefügt. Die entsprechenden Funktionen zur Berechnung sind in der Präsentation von [\[6\]](#) zu finden. Das Dataframe ist ausreichend für die Darstellung der Weltkarte. Für die Auswahl der Länder und der eventuellen Provinzen, sowie das Vorberechnen der Vorhersagen ist die Datenstruktur allerdings zu schwach. Aus diesem Grund wurde sich für eine weitaus komplexere Datenstruktur entschieden. Dies gelingt mit den Listenstrukturen von **Splitted\_Global\_DF** und dessen Nachfolger **Splitted\_Global\_DF\_States**. Wie bereits erwähnt ist die erste Listenstruktur veraltet und kann in näherer Zukunft entfernt werden. **Splitted\_Global\_DF** ist eine Liste von Dataframes, welche für jedes Land ein Dataframe beinhaltet. Für die Vorberechnungen der Vorhersagen jedoch immer ein Dataframe übergeben werden muss und es öfter vorkommt, dass die Länder mehrere Staaten besitzen, mussten die Staaten ebenfalls in neue Dataframes aufgeteilt werden. Dies wurde durch die **map()**-Methode des **modelr** Packages begünstigt. Dadurch entsteht dann die Listenstruktur **Splitted\_Global\_DF\_States**. Diese Liste enthält die Dataframes der einzelnen Länder und wiederum Dataframes der einzelnen Staaten. Dadurch kann eine fehlerfreie Vorberechnung der Vorhersagen bewerkstelligt werden. Es wurde sich für diesen Ansatz entschieden, da so die Daten für die Vorhersage nicht jedes Mal erneut in Echtzeit berechnet werden müssen und das Übermitteln der Daten an den generierten Vorhersagebericht erleichtert wird. Das Vorberechnen der Vorhersagen wurde mithilfe von sechs Funktionen im R-Skript realisiert. Wobei für jede Vorhersage (Anzahl der Infizierten, Genesenen und Todesfälle) zuerst eine Hilfsfunktion als Bindeglied zwischen dem übermittelten Dataframes (**Forecast\_Confirmed**, **Forecast\_Deaths**, **Forecast\_Recovered**) und der tatsächlichen Vorberechnung implementiert wurde. Bei der tatsächlichen Vorberechnung der einzelnen Vorhersagen ist das Verfahren gleichbleibend, lediglich die genutzte Datenspalte unterscheidet sich von Fall zu Fall. Zuerst wird eine Liste mit Date-Objekten generiert, damit die später ausgegebenen Zeitstempel für den Benutzer verständlich ausgegeben werden. Im Anschluss wird die ARIMA-Methode für das sogenannte Fitting genutzt (dazu später mehr im Kapitel **Forecasts**). Abschließend wird für das übermittelte Dataframe die Vorhersage erzeugt und dann als Dataframe zurückgegeben. Deshalb sind die drei Listen für die Vorhersagen strukturell zu der Liste der **Splitted\_Global\_DF\_States** identisch.

# Praktische Umsetzung des Dashboards

## Über Shiny

Das Dashboard wurde größtenteils mit der Hilfe des R-Package **Shiny** realisiert. Shiny bietet sehr viele Möglichkeiten unkompliziert **interaktive Webanwendungen** und Dashboards zu erstellen. Eine Shiny-App benötigt lediglich zwei Parameter zum Starten: das **UI**, welches die Darstellung der Inhalte enthält und den **Server**, der für die Datenverarbeitung zuständig ist.

## Grober Aufbau der Gesamtapp

Für das UI wurde die Bibliothek **shinydashboard** verwendet, welches ein fertiges Grundgerüst eines Dashboards zur Verfügung stellt. In der Designphase wurde sich für ein zweigeteiltes Dashboard entschieden. Die Hauptinstanz **DashboardPage** enthält durch diese Designentscheidung die Module **DashboardSidebar** und **DashboardBody**.

## DashboardSidebar

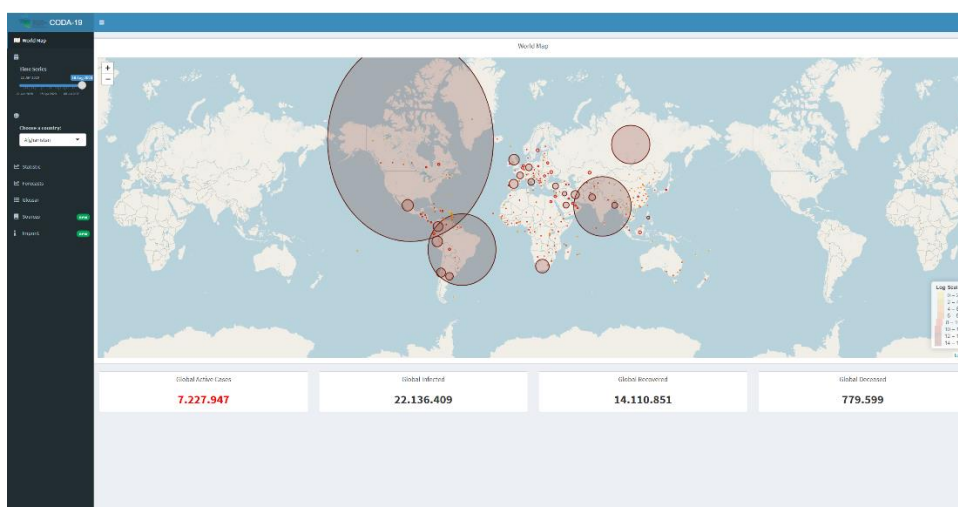
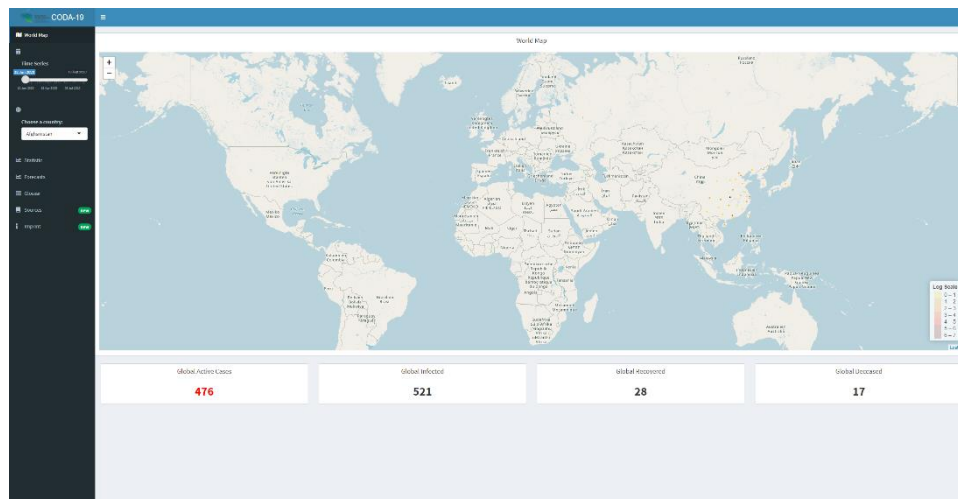
**DashboardSidebar** ist das seitliche Menü der Applikation. Es enthält die Oberbegriffe des Inhalts, die durch Menüitems implementiert wurden. **Menuitems** können einen Namen, ein Icon, einen TabName, der die Referenz für den Inhalt bildet oder einen Darstellungstypen (Slider, Kalender usw.) besitzen. Das seitliche Menü hat den Vorteil, dass es modular aufgebaut ist, d.h. es können weitere Inhalte ohne großen Aufwand hinzugefügt werden. Dazu muss lediglich ein weiteres Menüitem hinzugefügt werden und bei **TabName** der Referenzname des neuen Inhalts angegeben werden. Insgesamt enthält das Dashboard acht Menüitems: Die Weltkarte, den Zeitstrahl, die Auswahl des Landes, die Statistik, die Vorhersagen, das Glossar, die Quellen und das Impressum. Der Zeitstrahl und die Auswahl der Länder wurde in der Navigationsleiste umgesetzt, alle anderen Inhalte befinden sich im DashboardBody. Der Zeitstrahl wurde durch ein SliderInput umgesetzt, dabei wird das Minimum und das Maximum auf das Minimum bzw. Maximum aus dem Datum **Date** der Datenquelle gesetzt. Die Auswahl der Länder konnte durch einen SelectInput realisiert werden, der als Input eine Liste der Länder aus der Datenquelle enthält.

## DashboardBody

Im DashboardBody wird der Hauptinhalt angezeigt. Durch sogenannte TabItems, die den TabNamen beinhalten können die Inhalte passend zu ihrem Oberbegriff dargestellt werden.

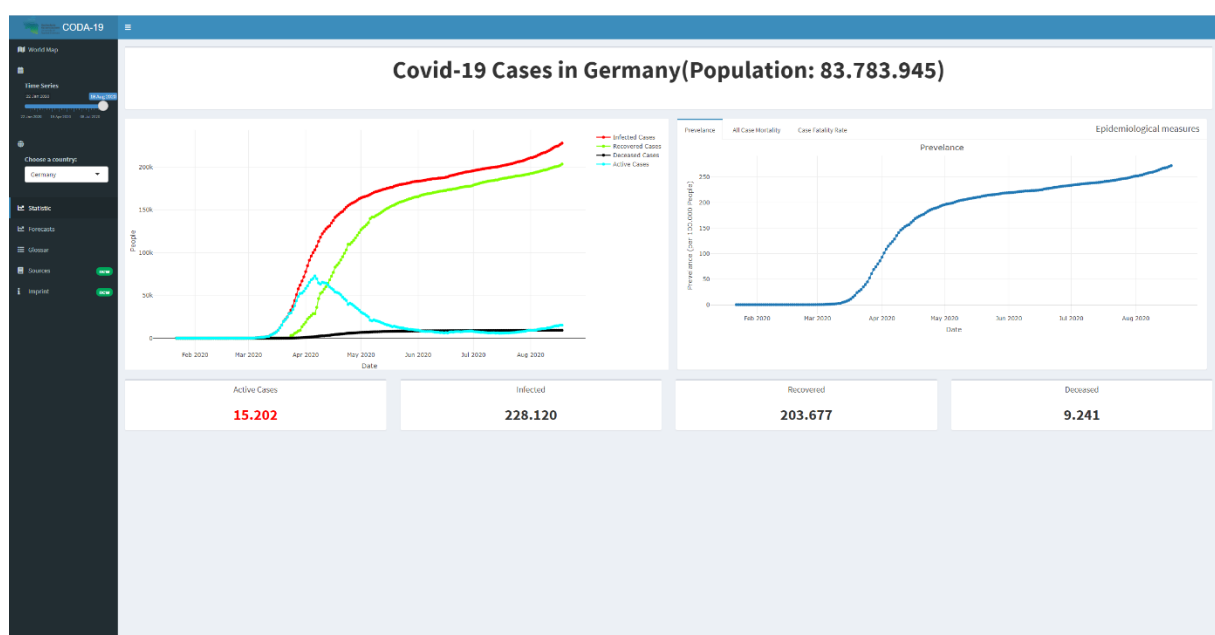
## Weltkarte

Die Weltkarte wurde mit Hilfe des Pakets **Leaflet** realisiert, welches die Covid-19-Daten benutzt und grafisch darstellt.



## Statistik

Bei der Statistik wurde **Plotly** als Hilfsbibliothek benutzt, die **interaktive** Graphen ermöglicht.



## Forecasts

Das grobe Vorgehen für das Vorkalkulieren der Vorhersagedaten wurde bereits bei der Datenstruktur erwähnt. In diesem Abschnitt wird das R-Package „forecast“ genauer vorgestellt und aufgezeigt welche Methoden für die Berechnung für die Vorhersagen ausgewählt wurden. Das forecast-Package wird vor allem für Time Series und Lineare Modelle genutzt [\[7\]](#). In diesem Projekt wurde das Package vorrangig für eine Time Series Analyse benutzt. Bei einer Time Series Analyse handelt es sich zumeist um einen zeitlichen Verlauf, welcher anhand gegebener numerischer Daten auf einen Trend hinweist und wird daher oft in Anwendungsfällen für den Aktienmarkt genutzt [\[8\]](#). Im Fall des Packages stehen hierfür drei Vorhersagemethoden zur Verfügung: ETS, Automatic ARIMA/ARIMA und STL. Die Methoden werden in den folgenden Absätzen kurz betrachtet, Automatic ARIMA und ARIMA werden zu einem Abschnitt zusammengefasst.

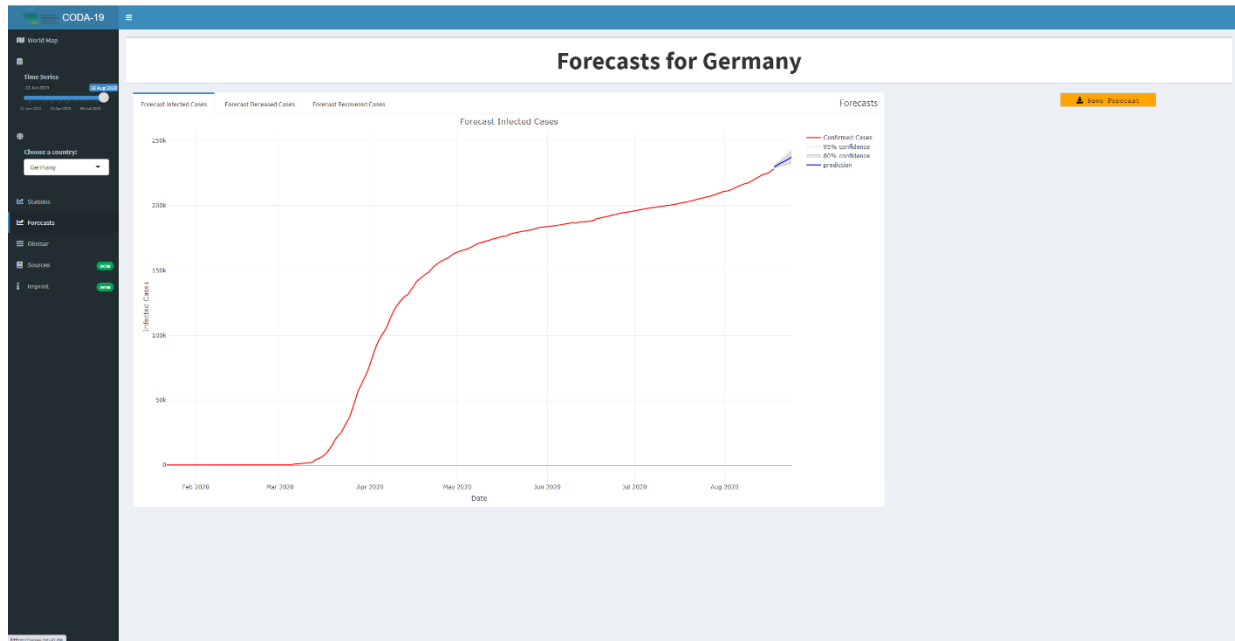
Bei **ARIMA (Autoregressive integrated moving average)** handelt es sich um ein Regressionsverfahren. Hierbei werden die gewichteten Summen aus zurückliegenden Messwerten und Zufallseinflüssen sowie der Stationarität zu einem Modell zusammengefügt und geben so einen nicht-saisonalen Trend vor. Bei einer Time Series bedeutet die Stationarität, dass die zugrunde gelegte Verteilungsfunktion der Messwerte zeitlich konstant ist. Zeitreihen, die eine veränderliche Varianz und veränderliche höhere Momente aufweisen können von ARIMA nicht beschrieben werden [\[9\]](#).

Bei **ETS (Exponential Smoothing)** handelt es sich um eine Erweiterung der ARIMA Methode. Hierbei steht vor allem durch die „Holt-Winters' seasonal method“ saisonale Datensätze im Fokus. Dabei wird bei der Übergabe der Datensätze angegeben, ob es sich zum Beispiel um Quartalsdaten ( $m=4$ ) oder monatliche Daten ( $m=12$ ) handelt. Dabei wird zwischen zwei Methoden unterschieden, einmal die additive und die multiplikative Methode. Die additive Methode wird bevorzugt, wenn die Varianz der saisonalen Daten im Zeitverlauf konstant bleiben, während die multiplikative Methode benutzt wird, wenn hohe Abweichungen von Zeit zu Zeit auftauchen. Deswegen wird diese Vorhersagemethode gerne genutzt, wenn es zu hohen Abweichungen kommt [\[10\]](#).

**STL (Seasonal and Trend decomposition using Loess)** zerlegt alle Arten von saisonalen Daten. Das Zerlegen von Time Series in die Daten-, Trend-, Saison- und Restebene gibt Einblick in die Entwicklung der Daten über die Zeit. Damit bei dieser Methode eine Vorhersage stattfinden kann werden die Vorhersagen für die saisonalen und die angepassten saisonalen Komponenten separat errechnet. Bei Ersterer wird lediglich eine naive Methode für die Vorhersage angewendet, somit verändert sich die Vorhersage nur geringfügig. Die Vorhersage für die angepasste saisonale Komponente benötigt jedoch eine nicht-saisonale Vorhersagemethode, hier eignet sich zum Beispiel das ARIMA Modell [\[11\]](#).

In diesem Projekt wurde sich für ein **ARIMA** Modell entschieden, da die Erfahrung gezeigt hat, dass diese Vorhersagen die geringsten Abweichungen von den tatsächlichen Daten aufwiesen. Um Vorhersagen mit dem genannten R-Package zu gewährleisten, benötigten alle der drei genannten Methoden ein Fitting-Objekt. Für jede der Methoden gibt es eine entsprechende Fit-Methode. Da das Vorhersagen von neuen Infektions-, Genesenen- und Todesfällen zum Ende des Projektes hinzugefügt wurde und die Zeit für selbstdefinierte Fittings gefehlt hat, wurde die Methode **auto.arima()** gewählt, welche das Fitting automatisch durchführt. Danach wurde das Fitting-Objekt der **forecast()** Funktion übergeben und ein **Vorhersagezeitraum von sieben Tagen** angegeben. Um eine korrekte Darstellung des Datums, der durch die Vorhersage erzeugten Zeitstempel im späteren Diagramm zu gewährleisten, musste eine zusätzliche Liste mit entsprechenden Daten angegeben werden. Das Speichern des Forecasts wurde über ein Markdown Template geregelt. Wählt der Benutzer die Schaltfläche „Save Forecast“ aus, werden die für das Template benötigten Daten als

**YAML** Parameter als Array übergeben, im Header der Vorlage eingelesen und an den entsprechenden Stellen die Daten der Vorhersage adaptiert.



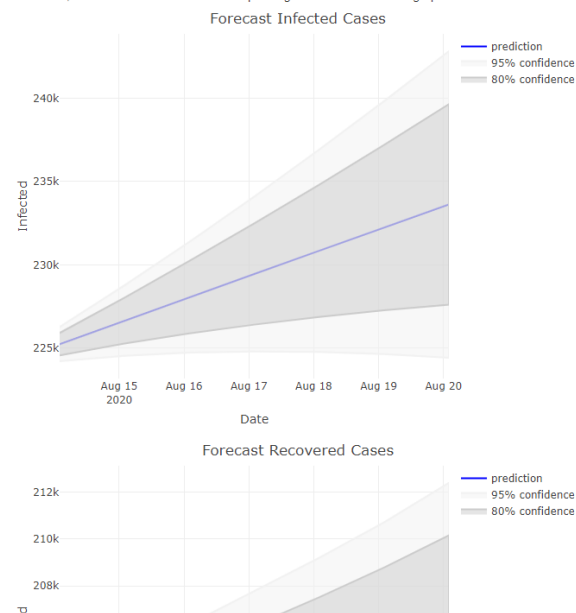
## Forecast Report

Forecast Data from Germany, download on: 2020-08-16

This report contains forecast data presented in plots and tables. You can use this data to compare manually, how good the comparison was. To do this, you have to wait until one of the dates has been reached. Then go to the statistics page and compare the predicted values with the actual values.

### Plot Output

Hereinafter, the forecast data from the corresponding data are shown in the graphs below:

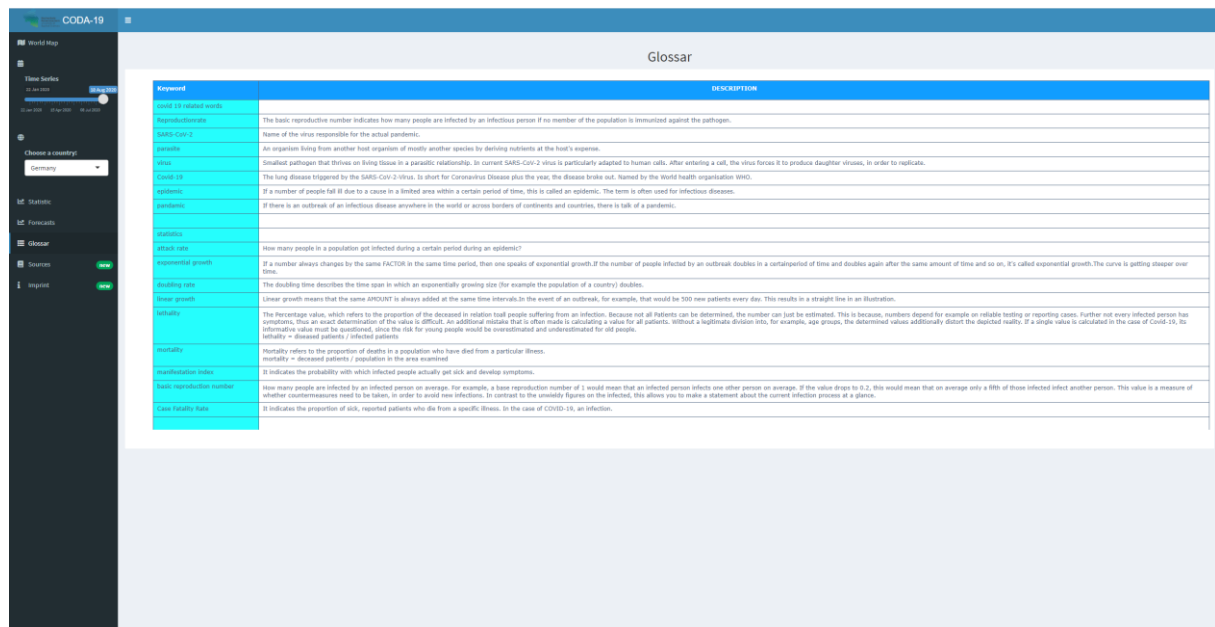


## Glossar

In Plotly ist es auch möglich interaktive Tabellen zu generieren, hierbei wurden die Begriffe und Erklärungen vorher in eine **CSV**-Datei geschrieben. Shiny liest diese Daten aus der CSV-Datei und fügt sie in die Plotlytabelle ein und gibt diese dann aus.

Das Glossar ist in folgende Bereiche aufgeteilt:

- **covid 19 related words:** Grundlegende Begriffe bezüglich des Virus und der damit einhergehenden Erkrankung.
- **statistics:** Erläuterung der verschiedenen epidemiologischen Kennzahlen, die der Erfassung des Infektionsgeschehen dienen.
- **activities:** Eine Beschreibung der verschiedenen Maßnahmen, um die Pandemie einzudämmen.
- **immunology:** Erklärung verschiedener Begriffe bezüglich einer Infektion.

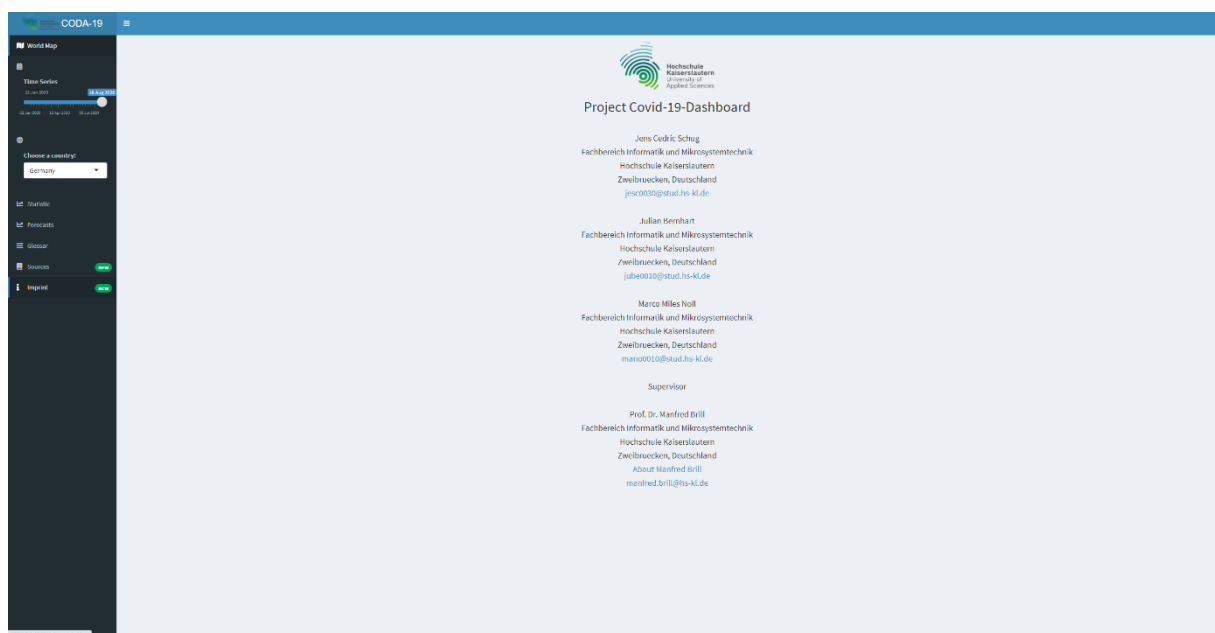
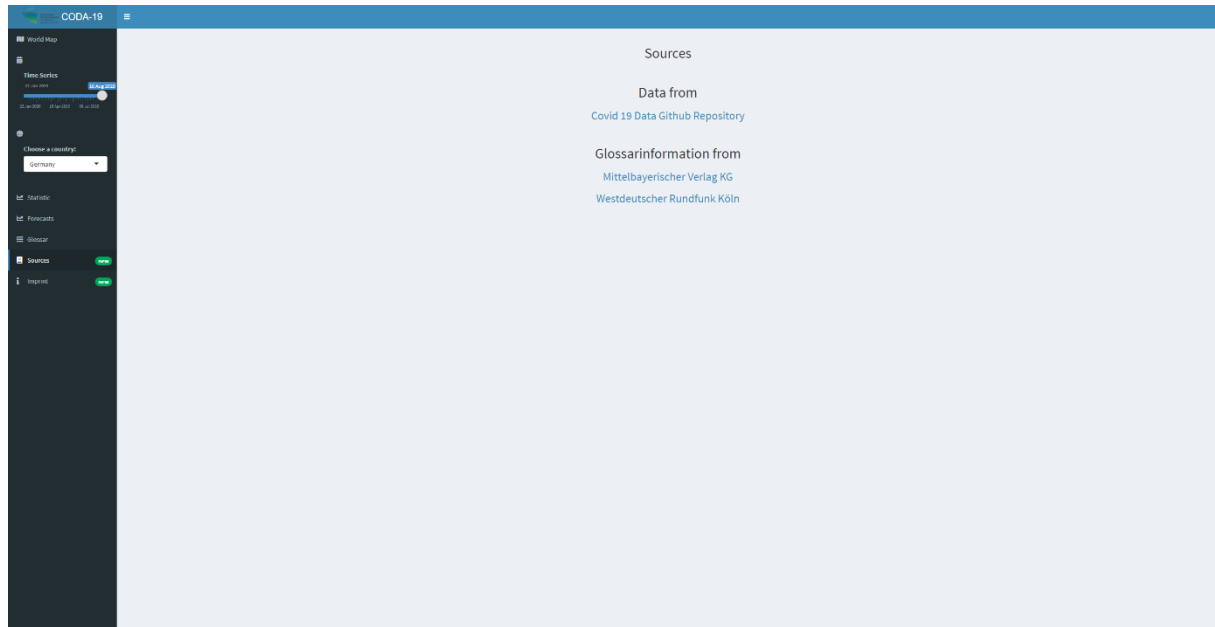


Keyword	DESCRIPTION
covid 19 related words	
reproduction number	The basic reproduction number indicates how many people are infected by an infectious person if no member of the population is immunized against the pathogen.
reproduction number	Name of the virus responsible for the actual pandemic.
sars-cov-2	An organism living from another host organism of mostly another species by deriving nutrients at the host's expense.
pathogen	Smallest pathogen that thrives on living tissue in a parasitic relationship. In current SARS-CoV-2 virus is particularly adapted to human cells. After entering a cell, the virus forces it to produce daughter viruses, in order to replicate.
virus	The lung disease triggered by the SARS-CoV-2 virus. In short for Coronavirus Disease plus the year, the disease broke out. Named by the World Health Organization WHO.
covid-19	If a number of people fall ill due to a disease in a limited area within a certain period of time, this is called an epidemic. The term is often used for infectious diseases.
epidemic	If there is an outbreak of an infectious disease anywhere in the world or across borders of continents and countries, there is talk of a pandemic.
pandemic	
incubation	
attack rate	How many people in a population got infected during a certain period during an epidemic?
exponential growth	If a number always changes by the same FACTOR in the same time period, then one speaks of exponential growth. If the number of people infected by an outbreak doubles in a certain period of time and doubles again after the same amount of time and so on, it's called exponential growth. The curve is getting steeper over time.
doubling rate	The doubling time describes the time span in which an exponentially growing size (for example the population of a country) doubles.
linear growth	Linear growth means that the same AMOUNT is always added at the same time intervals. In the event of an outbreak, for example, that would be 100 new patients every day. This results in a straight line in an illustration.
fatality	The Percentage value, which refers to the proportion of the deceased in relation to all people suffering from an infection. Because not all patients can be determined, the number can just be estimated. This is because, numbers depend for example on reliable testing or reporting cases. Further not every infected person has symptoms, thus an exact determination of the value is difficult. An additional mistake that is often made is calculating a value for all patients. Without a legitimate division into, for example, age groups, the determined values additionally distort the depicted reality. If a single value is calculated in the case of Covid-19, its fatality = deceased patients / infected patients
mortality	Mortality refers to the proportion of deaths in a population who have died from a particular illness.
mortality	mortality = deceased patients / population in the area examined
mortality index	It indicates the probability with which infected people actually get sick and develop symptoms.
basic reproduction number	How many people are infected by an infected person on average. For example, a basic reproduction number of 1 would mean that an infected person infects one other person on average. If the value drops to 0.2, this would mean that on average only a fifth of those infected infect another person. This value is a measure of whether countermeasures need to be taken, in order to avoid new infections. In contrast to the mortality figures on the infected, this allows you to make a statement about the current infection process at a glance.
case fatality rate	It indicates the proportion of sick, reported patients who die from a specific illness. In the case of COVID-19, an infection.

Das Glossar dient dazu, Begriffe, die oft im Zusammenhang mit Infektionskrankheiten auftreten, zu erläutern. Somit soll das Dashboard neben der Darstellung der Daten zur Pandemie auch Informationen beitragen, die das Verständnis über die Graphen und die Thematik vertiefen. Zu finden unter [\[2\]](#)

## Quellen und Impressum

Die Daten im Quellen- und Impressumsverzeichnis wurden, durch die Möglichkeit, dass Shiny HTML-Tags unterstützt, vorher formatiert und danach entsprechend eingefügt.





## Ausblick und Aufgabenverteilung

Was würden wir anders machen oder verbessern?

- Nicht so lange mit aktuell nicht lösbaren Fehlern aufhalten.
- Klarere Absprachen treffen, Aufgaben präziser formulieren.
- Anzahl der Meetings erhöhen, für Statusupdates.
- Ausführlicheres Dokumentieren von Design-Planung, Ideen, Abläufen und Inhalten.
- Code cleanup.

Wie kann das Projekt weitergeführt werden?

- Funktionierendes Buildtools.
- Design anpassen, visuelle Aufbereitung.
- Datenverwaltung und -nutzung optimieren.
- Skalierung von Kreisen auf der Weltkarte anpassen.

Welche zukünftigen Erweiterungen sind denkbar?

- Automatisierter Vergleich von Forecast und tatsächlichen Werten.
- Vergleich von Daten aus verschiedenen Quellen.
- Einbeziehen von Daten bezüglich getroffener Maßnahmen. Auf diese Weise können die Folgen verschiedener Maßnahmen auf den Verlauf der Pandemie verdeutlicht werden.

Aufgabenverteilung:

Aufgabe	Beteiligte
Dokumentation	Cedric, Julian, Marco
Dashboard (Einarbeitung in Shiny)	Cedric, Julian, Marco
Dashboard-WorldMap	Cedric, Marco
Dashboard-Statistics	Cedric
Dashboard-Forecast	Cedric, Julian
Dashboard-Glossar	Julian, Marco
Dashboard-Sources/-Imprint	Marco
Pipeline	Julian, Marco
Daten aufbereiten	Cedric, Julian, Marco

## Fazit

Die Einarbeitung in die Daten und die Entwicklung des Shiny Dashboards war sehr zeitintensiv. Shiny dient als visuelle Ergänzung für R und erlaubt eine deutlich fokussierte Darstellung der Daten im Vergleich von Plots oder R-Notebooks. Eine bessere Planung und Ideenfindung im Voraus hätte die Arbeitsaufteilung verbessert und das Konzept des Dashboards konkretisiert. Ebenfalls wären somit doppelte Arbeiten vermieden worden. Da das Thema der Corona-Pandemie jedoch sehr aktuell war, wurde das Interesse in der Gruppe geweckt und die Motivation bezüglich der Arbeitsweise konnte dadurch bedingt im hohen Maße erhalten bleiben.

Da das Projekt sowohl zeitlich als auch kausal an den Verlauf der Pandemie gekoppelt ist, gibt es einen zeitlichen Rahmen für die Relevanz, wodurch diese ab dem Einführen erfolgreicher Gegenmaßnahmen, wie einen Impfstoff, zunehmend abnimmt.

## Quellenverzeichnis

- [1] „<https://github.com/datasets/covid-19>,“ [Online].
- [2] Noll, Schug, Bernhart, „<https://github.com/Anker13/COVID-19-Shiny-Dashboard>,“ [Online].
- [3] J. Bryan, „<https://stat545.com/make-test-drive.html>,“ [Online].
- [4] „<https://cran.r-project.org/bin/windows/Rtools/>,“ [Online].
- [5] A. Gold, „<https://support.rstudio.com/hc/en-us/articles/360047157094-Managing-R-with-Rprofile-Renviron-Rprofile-site-Renviron-site-rsession-conf-and-repos-conf>,“ [Online].
- [6] „<https://www.krebsregister-bayern.de/Papers/EpidemiologischeGrundbegriffe.pdf>,“ [Online].
- [7] „<https://cran.r-project.org/web/packages/forecast/forecast.pdf>,“ [Online].
- [8] „<https://www.investopedia.com/terms/t/timeseries.asp>,“ [Online].
- [9] „<http://www.reiter1.com/Glossar/ARIMA.htm>,“ [Online].
- [10] R. J. Hyndman und G. Athanasopoulos, „<https://otexts.com/fpp2/holt-winters.html>,“ [Online].
- [11] R. J. Hyndman und G. Athanasopoulos, „<https://otexts.com/fpp2/forecasting-decomposition.html>,“ [Online].