

Das Antragsportal des Forschungsdatenzentrums Gesundheit

Nutzerhandbuch für die Arbeit mit Posit und RStudio

Version 01-00-000

Inhalt

1	Voraussetzungen.....	3
2	Posit Workbench.....	3
2.1	Einloggen in die Workbench.....	3
2.2	Starten neuer Sessions.....	4
3	Erste Schritte mit R-Studio	6
3.1	RStudio-Fenster.....	6
3.2	Erstellen/Verwalten von Skripten.....	8
3.3	Arbeiten mit Paketen (packages) im FDZ	11
4	Arbeiten mit GitLab	11
4.1	Einführung in Git.....	11
4.2	Automatischer Git Zugriff in RStudio.....	11
4.3	Arbeiten mit dem Git Repository	12
5	Datenbankverbindung	16
5.1	Allgemeine Hinweise zum Arbeiten mit der Datenbankverbindung im Analyseraum	16
5.2	Anzeigen des Datensatzzuschnittes	17
5.3	Zusammenstellen neuer Datentabellen.....	18
5.4	Schreiben dieser Datentabelle ins Environment	19

5.5	Schreiben von Tabellen in die Datenbank.....	19
6	Prozess Lauffähigkeitsprüfung.....	20
6.1	Korrekte Benennung Ihres Skriptes.....	20
6.2	Korrekte Benennung Ihrer Ergebnistabellen	20
6.3	Speichern und Pushen Ihres Skriptes ins Git Repository	22
6.4	Beantragen der Lauffähigkeitsprüfung im Antragsportal.....	22
6.5	Einsehen der Ergebnismeldung Ihrer Lauffähigkeitsprüfung	23
7	Beenden und Management von Sessions.....	23
7.1	Wechsel der R-Studio Pro-Sitzung.....	23
7.2	Beenden einer Session.....	23
7.3	Wiederaufnahme einer Session	24
8	Optionale Funktionen	25
8.1	Verwalten Ihrer Sessions	25
8.2	Git Projekt Zugriff in RStudio: Terminal	26
9	FAQ.....	27
9.1	Meine Gitanbindung ist nicht im User interface zu sehen.....	27
9.2	Ich kann ein wichtiges Paket nicht installieren bzw. finde es nicht in der packages Registerkarte	29
9.3	Datenbankverbindung über Neustart der Session.....	31
9.4	Starten einer Session ohne Projekt.....	31
9.5	Ich kann nichts in mein Git Repository pushen	34
9.6	Ich kann meine erzeugten Datentabellen in der Datenbankverbindung nicht anzeigen lassen.....	34
9.7	Ich kann mein R Skript nicht im Antragsportal hochladen	34
9.8	Meine Datenbankverbindung in einer Session ist plötzlich unterbrochen	35

1 Voraussetzungen

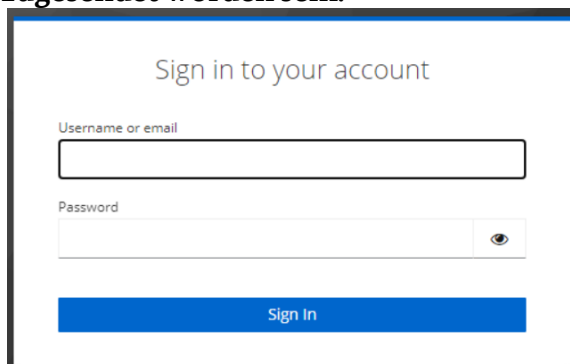
Bevor Sie mit der Arbeit in Ihrem Analyseraum beginnen, lesen Sie bitte die [Programmcoderichtlinien](#), die sich in der Anlage zu den Nebenbestimmungen befinden. **Wichtig! Wenn Sie diese Regeln nicht einhalten, kann nicht sichergestellt werden, dass die Ausgabe Ihrer Ergebnisse mit Ihren R-Skripten ermöglicht wird.**

2 Posit Workbench

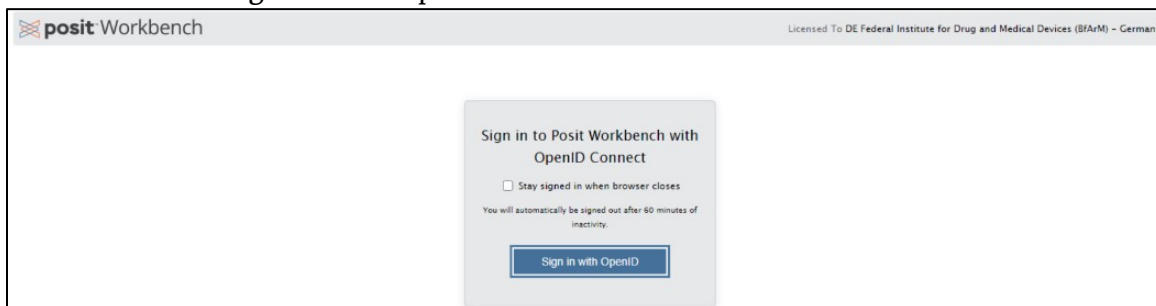
Die Posit Workbench ist die zentrale Plattform, auf der Sie Ihre Analyse Sessions verwalten können. Sie können – je nachdem, welche Sprache Sie im Antrag ausgewählt haben – RStudio oder JupyterLab Sessions starten, pausieren oder schließen. Wenn Sie in Ihrem Antrag mit einem Kollegen bzw. einer Kollegin zusammenarbeiten, können Sie über die Workbench Sessions gemeinsam verwalten oder in individuellen Sessions arbeiten. Zugang zur Workbench erhalten Sie über das Antragsportal über den Button "Analyseraum betreten".

2.1 Einloggen in die Workbench

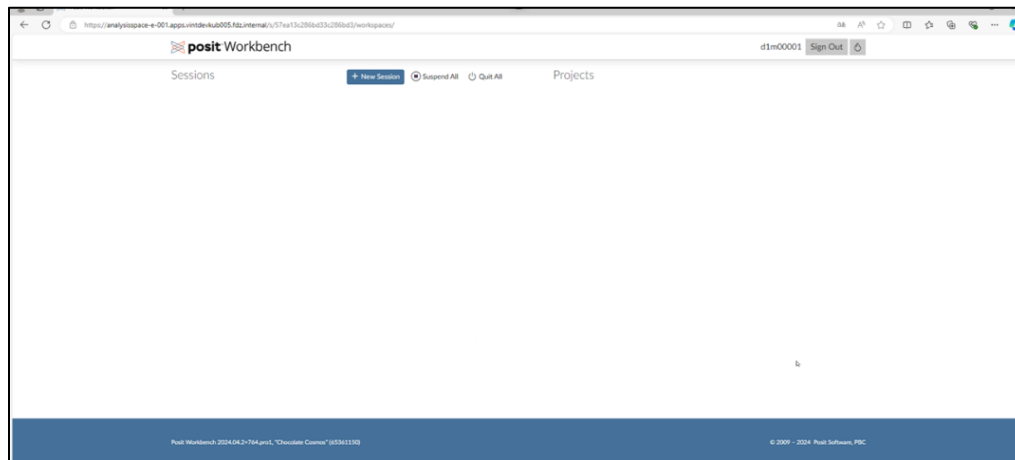
1. Tragen Sie Ihren Anmeldenamen sowie Passwort ein. **Diese sollten Ihnen per E-Mail zugesendet worden sein.**

A screenshot of a login form titled "Sign in to your account". It features two input fields: "Username or email" and "Password". The "Password" field has a small eye icon to its right for toggling visibility. Below the fields is a blue button labeled "Sign in".

2. Klicken Sie auf "Sign in with OpenID"

A screenshot of the Posit Workbench login interface. The header shows the "posit Workbench" logo and a license notice: "Licensed To DE Federal Institute for Drug and Medical Devices (BfArM) - Germany". The main content area contains a box titled "Sign in to Posit Workbench with OpenID Connect". Inside this box, there is a checkbox labeled "Stay signed in when browser closes" which is currently unchecked. Below the checkbox, it says "You will automatically be signed out after 60 minutes of inactivity." At the bottom of the box is a blue button labeled "Sign in with OpenID".

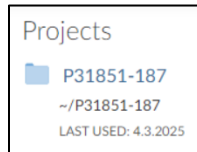
3. Startseite der Workbench



Sie können in Ihrer Workbench R Sessions starten, pausieren und schließen.

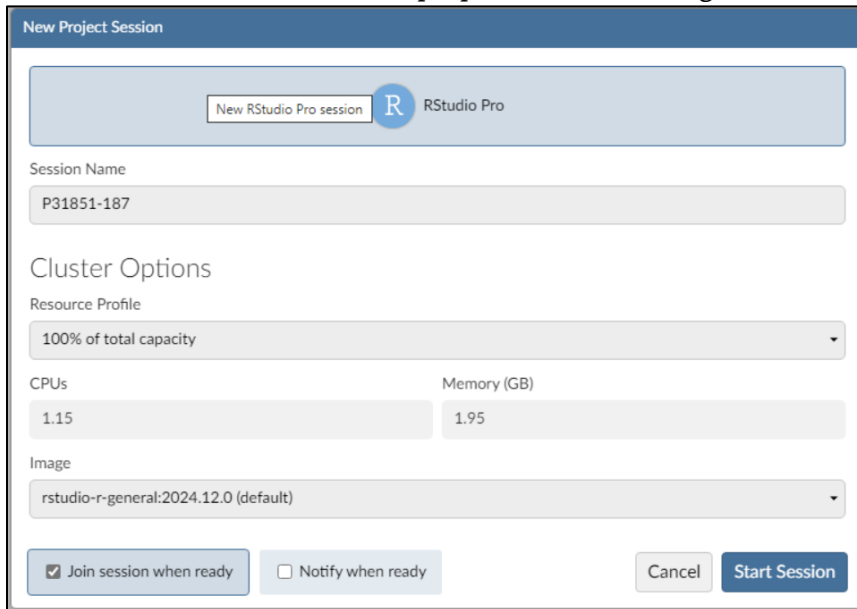
2.2 Starten neuer Sessions

Wenn Sie unter Projects auf der rechten Seite auf den Ordner mit Ihrer Antragsnummer (in diesem Beispiel P31851-187) klicken, können Sie eine neue Session starten:

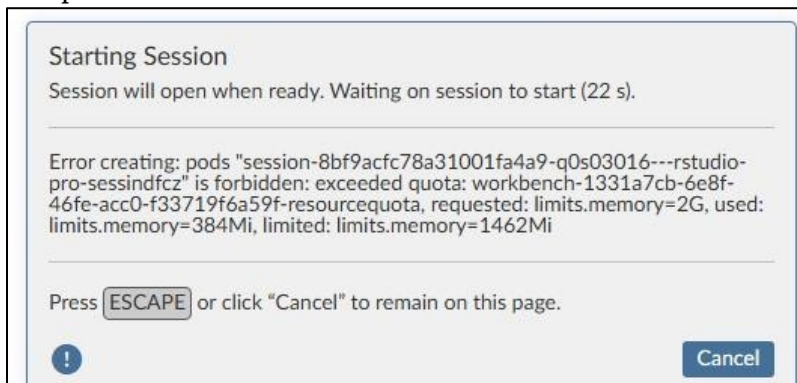


Sollte unter "Projects" kein entsprechender Link vorhanden sein, gehen Sie bitte wie hier vor: [Starten einer Session ohne Projekt](#)

Anschließend erscheint ein Pop up Fenster mit einigen Einstellungen für Ihre Session:



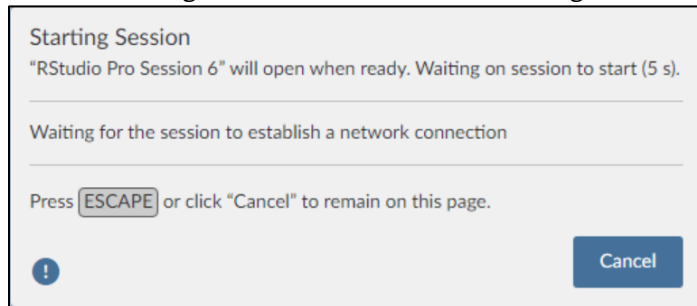
Im Feld "Session Name" können Sie jeder Session einen individuellen Namen geben, um sie leichter zu identifizieren, beispielsweise bei der parallelen Ausführung mehrerer Skripte. Unter "Resource Profile" legen Sie den maximalen CPU-Leistungsanteil für diese Session fest. Die Auswahl von 100% ist sinnvoll, wenn Sie ausschließlich in dieser Session arbeiten. Bei parallelen Analysen empfiehlt sich eine Verteilung der Ressourcen, um die Performance zu optimieren.



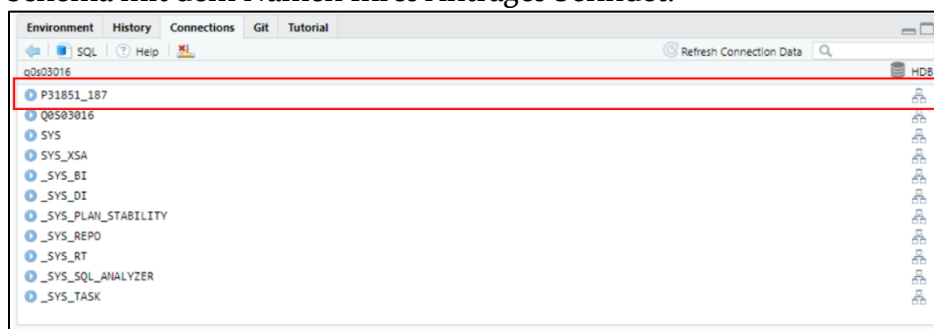
Hinweis: Bitte beachten Sie, dass sich die Auswahl der Ressourcen auf die gesamte Workbench bezieht. Wenn Sie also zu zweit in diesem Analyseraum tätig sind und ein Kollege bzw. eine Kollegin von Ihnen bereits eine aktive Session mit 50% hat, können Sie in Ihrer Session maximal 50% auswählen. Sollten Sie versuchen, eine Session zu starten, obwohl die Ressourcen bereits maximal ausgelastet sind, werden Sie eine Fehlermeldung mit entsprechendem Verweis erhalten:

In diesem Fall überprüfen Sie, ob Sie eine bereits aktive Session schließen, weniger Ressourcen auswählen können oder kontaktieren Sie Ihren Kollegen bzw. Ihre Kollegin über den Gebrauch der Ressourcen.

Zum Starten einer Session klicken Sie auf "Start Session". Anschließend wird Ihre Session automatisch gestartet. Das kann einen Augenblick dauern.



Wenn der Prozess abgeschlossen ist, sollten Sie oben Ihre Datenverbindung einsehen können. Dafür klicken Sie auf den Reiter Connections und überprüfen, ob sich dort ein Schema mit dem Namen Ihres Antrages befindet:



Sollten Sie diesen nicht sehen, können Sie die Session neu starten (mehr dazu: [Datenbankverbindung über Neustart der Session](#)).

3 Erste Schritte mit R-Studio

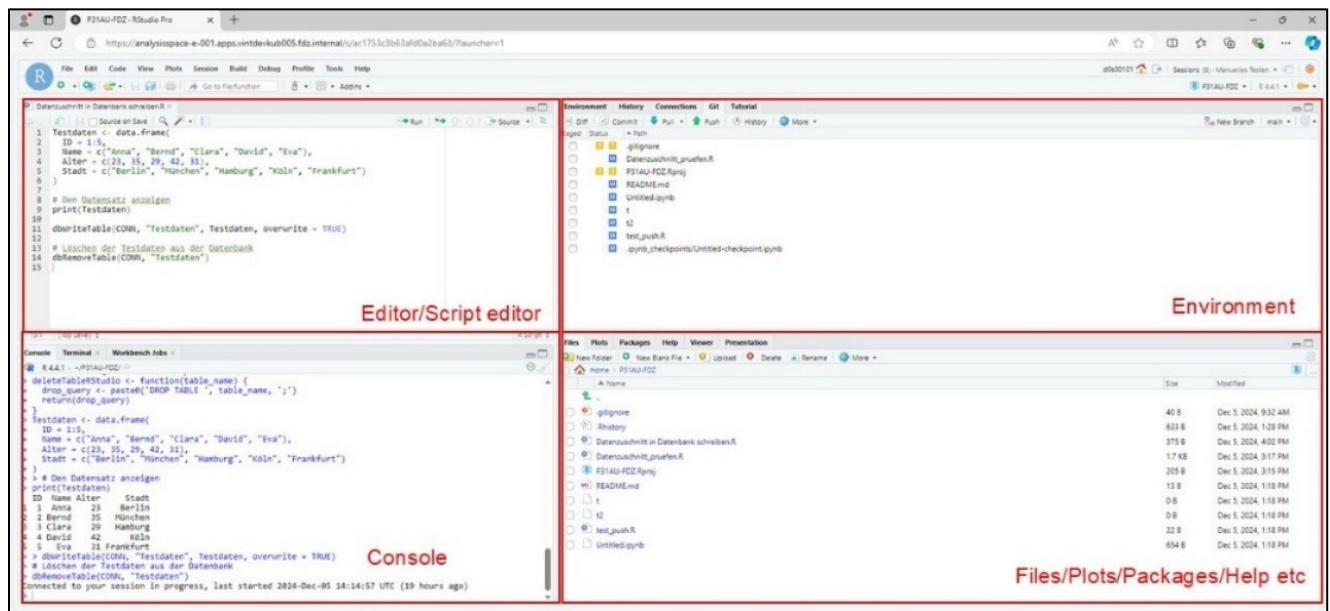
3.1 RStudio-Fenster

In diesem Kapitel wird zunächst auf die Analyseumgebung in Ihrer R Session eingegangen. In einer aktiven R Session sind in der Regel vier verschiedene Fenster zu sehen. Wenn Sie zum ersten Mal eine Session öffnen, kann es sein, dass Sie das Fenster oben links, den "Editor" noch nicht sehen. Wie Sie diesen öffnen, erfahren Sie im Laufe dieses Kapitels.

1. Im **Editor** oben links können Sie Ihre R/RMarkdown Skripte bearbeiten und speichern. Dieses Fenster erscheint automatisch, wenn Sie ein bereits abgespeichertes Skript in Files öffnen oder ein neues Dokument erstellen ([3.2 Erstellen von Skripten](#))

2. Die **Console** (unten links) dient sowohl zum Schreiben und Ausführen kurzer interaktiver R-Befehle als auch zur Anzeige der Ergebnisse Ihrer Analyseskripte.
 - a) Das **Terminal** in RStudio bietet direkten Zugriff auf die System-Shell. Dies ermöglicht die Ausführung von Betriebssystembefehlen, beispielsweise Git-Befehle zum Wechseln von Branches oder zur Versionskontrolle.
 - b) Über die Registerkarte **Workbench Jobs** können Sie parallel die Ausführung verschiedener Analysen steuern.
3. Im Bereich **Environments** werden temporäre R-Objekte angezeigt, die während dieser R-Sitzung erstellt wurden.
 - a) Die **History** Registerkarte zeigt alle Befehle an, die Sie in der aktuellen Sitzung in der Console ausgeführt haben.
 - b) In der Registerkarte **Connections** sehen Sie die Datenbankverbindung, in der Sie auch Ihren Datenzuschnitt einsehen können.
 - c) In der **Git** Registerkarte sehen Sie Dateien aus Ihrem Homeverzeichnis, die sich noch nicht in der aktuellen Version in Ihrem Git Repository befinden. In dieser Registerkarte können Sie die Push und Pull Befehle ausführen. Mehr Informationen dazu finden Sie im Kapitel zu [Arbeiten mit GitLab](#).
 - d) Die **Tutorial** Registerkarte wird verwendet, um interaktive Tutorials auszuführen, die vom learnr-Paket bereitgestellt werden. Diese Tutorials helfen Ihnen, die Grundlagen von R zu erlernen und die Werkzeuge des Tidyverse zu nutzen. Die Registerkarte bietet eine benutzerfreundliche Oberfläche, um Schritt-für-Schritt-Anleitungen direkt in RStudio zu folgen.
4. Rechts im Fenster unter **Files/Plots/Packages etc.** finden Sie eine Auswahl verschiedener Registerkarten, auf die im Folgenden eingegangen werden:
 - a) Unter **Files** werden die Diagramme, Tabellen oder HTML-Ausgaben des ausgeführten Codes zusammen mit den im Homeverzeichnis und Git Repo gespeicherten Dateien angezeigt.
 - b) In der Registerkarte **Plots** sind Diagramme zu sehen, die Sie mit ihren Analyseskripten erzeugt haben. Bitte beachten Sie, dass Diagramme nicht als Ergebnisse am Ende ausgegeben werden können.
 - c) In der Registerkarte **Packages** sehen Sie die vorinstallierten und verfügbaren R-Pakete. Sie können Pakete entweder durch Auswahl in der Liste laden oder mit dem Befehl "library(packagename)".
 - d) Die Registerkarte **Help** bietet Zugang zur R-Dokumentation und anderen Hilferessourcen. Hier können Sie nach Informationen zu R-Funktionen und -Paketen suchen. Wenn Sie beispielsweise in der Console ?install.packages eingeben, wird die entsprechende Hilfeseite in der Help-Registerkarte angezeigt.

- e) Die **Viewer** Registerkarte ermöglicht die Anzeige von Markdown-Dokumenten (gerendert als HTML), HTML-Inhalten und interaktiven Visualisierungen direkt in der IDE. Mit Markdown können Sie Berichte, Dokumentationen oder Dashboards erstellen und diese lokal im Viewer betrachten und analysieren, ohne dass eine Internetverbindung erforderlich ist.
- f) Die **Presentation** Registerkarte dient zur Anzeige interaktiver Präsentationen, die mit dem R Presentation Format erstellt wurden. Diese Präsentationen kombinieren Markdown und R-Code, um HTML5-Präsentationen zu erstellen, die direkt in RStudio angezeigt werden können.



3.2 Erstellen/Verwalten von Skripten

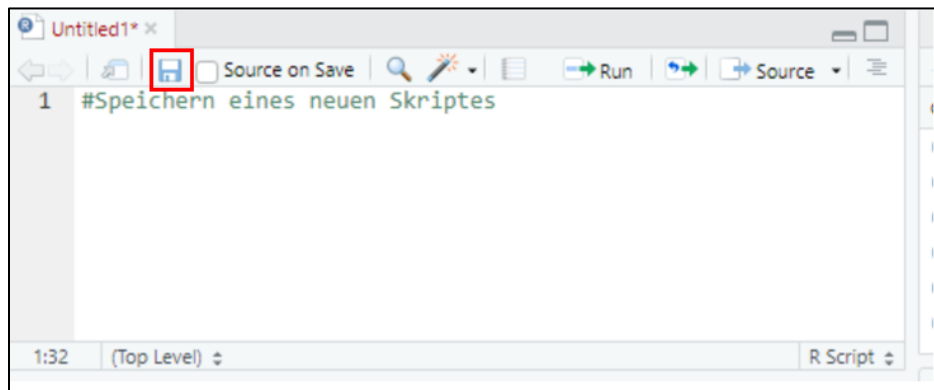
Sie können zum Entwickeln Ihrer Skripte R Skripte oder R Markdown Dokumente erstellen. Sie können ebenfalls Quarto Dokumente erstellen, dieses Format wird jedoch noch nicht in der Lauffähigkeitsprüfung unterstützt und kann daher auch nicht für die Ausgabe von Zwischen- und Endergebnissen verwendet werden.

Bitte beachten Sie, dass kein Einfügen von Inhalten (Strg + C) möglich ist. Wenn Sie Inhalte von beispielsweise alten Skripten nutzen möchten, müssen Sie dieses Skript über das Antragsportal hochladen. Durch den Git Pull Befehl wird dieses Skript in Ihr Homeverzeichnis geladen und steht Ihnen zum Arbeiten zur Verfügung.

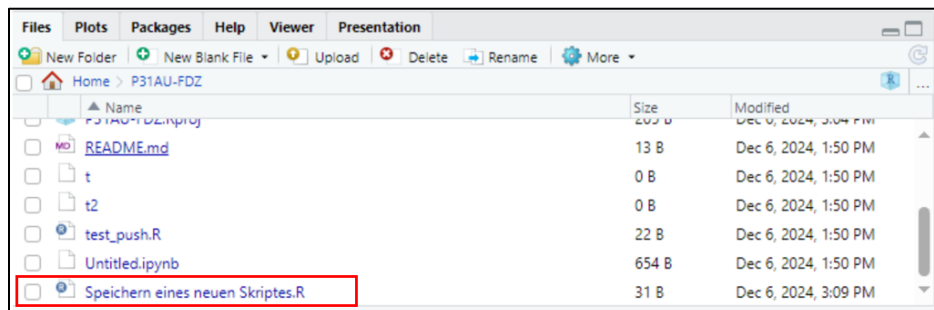
3.2.1 Verwalten von R Skripten

Zum Erstellen eines neuen R Skriptes können Sie entweder über **File > New File > R Skript** auswählen oder Sie klicken auf das grüne Plus Symbol direkt unter File  und wählen R Skript aus.

Zum Speichern eines neuen **R Skriptes** klicken Sie entweder auf das Speichern Symbol:



und vergeben Ihrem Skript einen Namen oder Sie klicken auf File > Save und vergeben dann einen Namen. Ihr Skript wird automatisch im Homeverzeichnis Ihres Analyseraums gespeichert. **Bitte wählen Sie keinen anderen Ordner aus als den, der zum Speichern vorgeschlagen wird.**



Um eine vorhandene Datei zu öffnen, verwenden Sie entweder das Menü **File > Open File** und wählen die entsprechende Datei aus oder klicken Sie in Ihrem Fenster unten rechts unter Files die Datei in Ihrem Homeverzeichnis an.

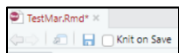
Ihre erzeugten Dateien können Sie in der Registerkarte **Files** einsehen. Das ist Ihr Homeverzeichnis. Dieses Homeverzeichnis ist Ihr Arbeitsverzeichnis, auf das Sie von den verschiedenen R Sessions aus zugreifen können. Bitte beachten Sie, dass ihr Homeverzeichnis noch nicht engmaschig gesichert wird, so dass in Ausnahmefällen Dateien verloren gehen könnten. Insofern sollten Sie Ihre Dateien in Ihrem Git Repository sichern. Die notwendigen Schritte dafür finden Sie im nachfolgenden Kapitel.

3.2.2 Verwalten von RMarkdown Dokumenten

Zum Erstellen eines neuen R Markdown Dokuments können Sie entweder über **File > New File > R Markdown** auswählen oder Sie klicken auf das grüne Plus Symbol direkt unter File

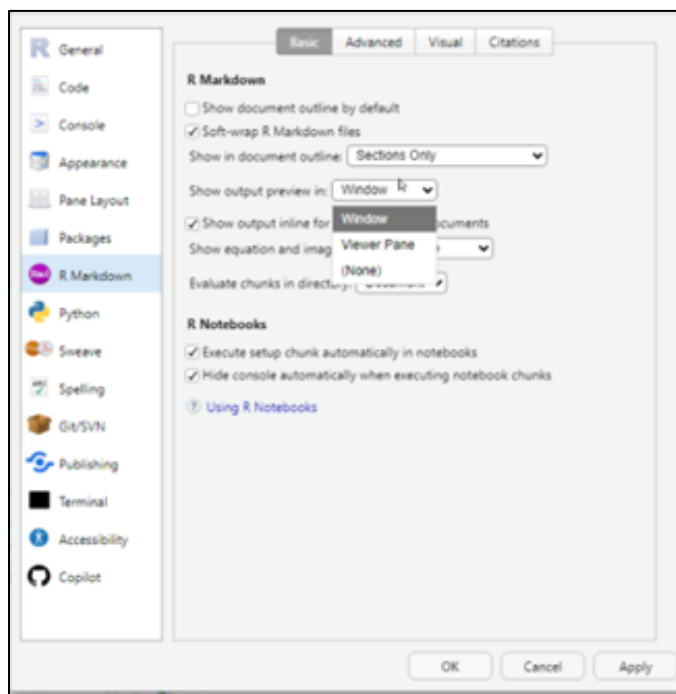


und wählen R Markdown aus. Beim Erstellen eines R Markdown Dokuments werden Sie direkt nach Auswahl von File > New File > R Markdown dazu aufgefordert, Ihrem Output Dokument einen Namen zu geben. Sie können als Output Format HTML, PDF oder Word auswählen. Bitte beachten Sie jedoch, dass Sie am Ende nur Tabellen als Ergebnisse und nicht die HTML/PDF/Word Datei erhalten. Bitte treffen Sie eine Auswahl und klicken anschließend auf "Ok". Nun können Sie mit der Bearbeitung Ihres Dokuments beginnen. Bitte beachten Sie, dass dennoch ein separates Speichern Ihres R Markdown Dokuments erforderlich ist. Dafür klicken Sie entweder auf das Speichern Symbol



oder wählen File > Save und vergeben Ihrem Dokument einen Namen. **Bitte wählen Sie keinen anderen Ordner aus als der, der zum Speichern vorgeschlagen wird.** Wenn Sie rechts neben dem Speichern Symbol das Feld "Knit on Save" auswählen, erfolgt automatisch eine Ergebnisausgabe Ihres Dokuments, wenn Sie dieses Speichern.

Zum Anzeigen Ihrer Ergebnisse können Sie entweder die Ausgabe im Viewer oder im Browser Tab auswählen. Dafür klicken Sie auf die Registerkarte Tools > Global Options > R Markdown in der linken Spalte und treffen anschließend unter "Show output preview in" Ihre Auswahl:



3.3 Arbeiten mit Paketen (packages) im FDZ

Unter der Registerkarte "packages" sehen Sie die Pakete, die Ihnen für die Arbeit im FDZ zur Verfügung gestellt werden. Bitte haben Sie Verständnis dafür, dass wir aufgrund der hohen Sicherheitsanforderungen an unser System zu diesem Zeitpunkt das selbstständige Installieren von Paketen noch nicht zur Verfügung stellen konnten.

Sollten Sie nach einer bestimmten Funktionalität suchen, die ein Paket beinhaltet, welches nicht vorinstalliert ist, finden Sie einen Vorschlag in den [FAQ](#).

Da die zur Verfügung stehenden Pakete vorinstalliert sind, müssen Sie diese **nicht** über den `install.packages()` Befehl installieren. Zum Laden nutzen Sie den `library()` Befehl oder wählen Sie in der Registerkarte "packages" die entsprechenden Pakete aus.

4 Arbeiten mit GitLab

4.1 Einführung in Git

Im FDZ wird GitLab zum Verwalten Ihrer Analyseskripte verwendet. Mithilfe Ihres zugewiesenen Git Repositorys können Sie Ihre Skripte hochladen und sichern. Mit der Push-Funktion können Sie Ihre Skripte nicht nur sichern, sondern auch mit anderen Nutzern und Nutzerinnen in Ihrem Analyseraum teilen. Auf diese Skripte können auch FDZ-Mitarbeiter und -Mitarbeiterinnen zugreifen, um Sie zum Beispiel in Supportanfragen zu unterstützen. Außerdem muss das Skript, das Sie für die Lauffähigkeitsprüfung sowie für die Zwischen- und Endergebnisse verwenden möchten, im Git Repository gespeichert werden.

Das Abspeichern der Skripte im Git Repository ist sehr wichtig, um sicherzustellen, dass Ihre Skripte zu jedem Zeitpunkt gesichert sind. **Daher sollten Sie den aktuellen Stand Ihrer Skripte nach Möglichkeit immer ins Git Repository pushen.** Wie das funktioniert, erfahren Sie in den nachfolgenden Kapiteln.

4.2 Automatischer Git Zugriff in RStudio

Wenn Sie eine RStudio Session öffnen, sollte die Git Anbindung automatisch hergestellt sein. Das können Sie überprüfen, indem Sie die Git Registerkarte aufrufen:



Sollte diese Git Registerkarte nicht automatisch vorhanden sein, dann fahren Sie bitte unter [Manuelle Git Anbindung herstellen](#) fort. Ansonsten können Sie unter "Arbeiten mit dem Git Repository" fortfahren.

4.3 Arbeiten mit dem Git Repository

Sollten Sie noch nie mit GitLab in RStudio gearbeitet haben, können Sie eine Anleitung auf der entsprechenden GitLab Seite finden: [Happy Git and GitHub for the useR - Intro](#)
Wir zeigen Ihnen hier die gängigen Anwendungen, denen Sie im Rahmen Ihrer Arbeit im FDZ begegnen werden.

4.3.1 Git Registerkarte

Die Git-Registerkarte zeigt Dateien an, die Sie während Ihrer Arbeit geändert haben, aber noch nicht ins Repository übertragen (gepusht) wurden. Hier sehen Sie den Status dieser Dateien für die Versionskontrolle.

Hier sind die wichtigsten Kategorien und ihre Bedeutungen:

1. **Unstaged (Nicht bereitgestellt):** Diese Dateien wurden geändert, aber noch nicht für den Commit vorgemerkt. Sie müssen zuerst bereitgestellt werden, bevor sie committet werden können. Diese Dateien sind mit einem  gekennzeichnet.
2. **Modified (Geändert):** Diese Dateien wurden seit dem letzten Commit geändert. Sie können entweder bereitgestellt oder zurückgesetzt werden. Diese Dateien sind ebenfalls mit  gekennzeichnet.
3. **Untracked (Nicht verfolgt):** Diese Dateien sind neu und wurden noch nicht zur Versionskontrolle hinzugefügt. Sie müssen zuerst verfolgt (tracked) werden, um in das Repository aufgenommen zu werden. Diese Dateien sind mit  markiert.
4. **Deleted (Gelöscht):** Diese Dateien wurden aus dem Arbeitsverzeichnis gelöscht und müssen committet werden, um die Löschung im Repository zu bestätigen. Diese Dateien sind mit einem  markiert



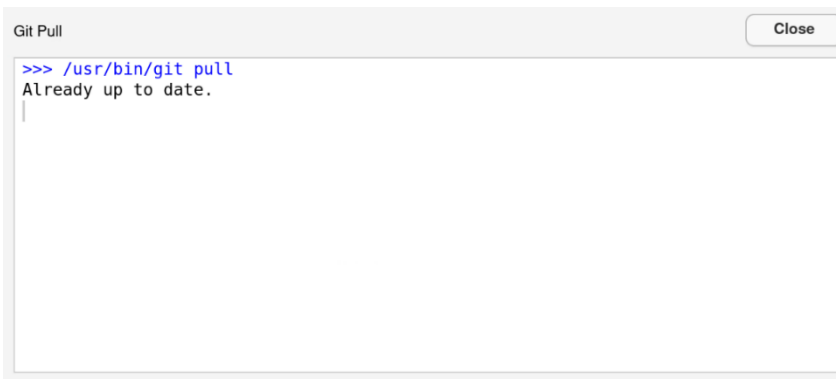
4.3.2 Git Pull

Der Befehl Pull dient in Git zum Lesen aus dem Git Projekt Repository. Dieser Befehl ist insbesondere wichtig, wenn Sie zusammen mit jemandem in einem Git Repository arbeiten. Das stellt sicher, dass Sie immer die aktuellen Dateien in Ihr Homeverzeichnis zum Bearbeiten in Ihrer Session zur Verfügung haben.

- Den Button "Pull" finden Sie in der Git Registerkarte:



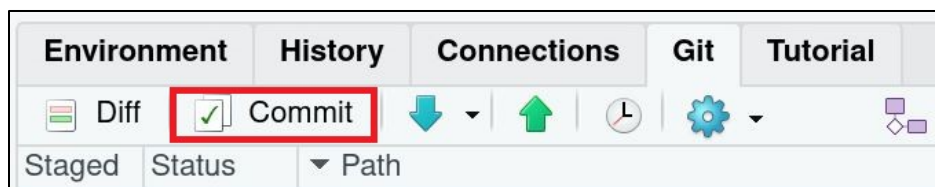
- Klicken Sie auf den Button "Pull" und es öffnet sich automatisch das Fenster "Git Pull":



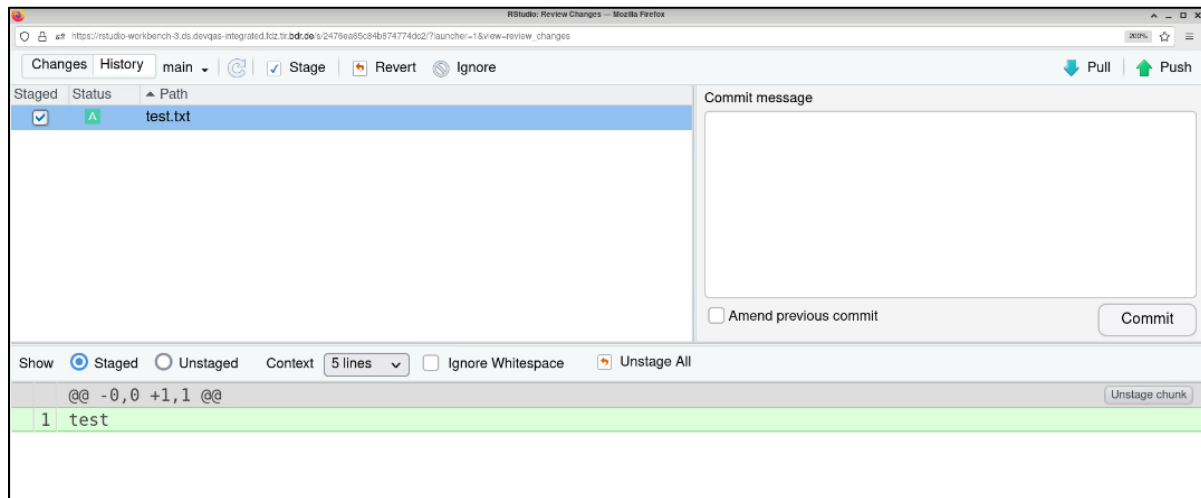
- Klicken Sie auf "Close", um das Fenster "Git Pull" zu schließen.

4.3.3 Git Commit

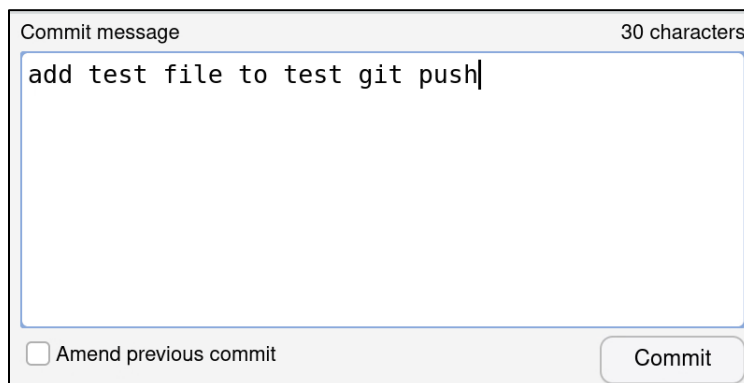
Wenn Sie Dateien (z.B. R Skript) bearbeitet haben, dann speichern Sie dieses zuerst ab, indem Sie auf das Speichersymbol im Source Fenster  klicken oder unter File > Save. Anschließend klicken Sie unter "Git" auf "Commit". Es öffnet sich das Fenster "RStudio: Review Changes" Workspace. Bitte wählen Sie die Datei aus, die Sie im Git Repository speichern wollen:



"RStudio: Review Changes" Workspace:



Fügen Sie eine Beschreibung des veränderten Codes in das Fenster "Commit message" des "RStudio: Review Changes" Workspaces hinzu und drücken Sie auf "Commit":



Es öffnet sich automatisch das Fenster "Git Commit".



Klicken Sie auf "Close", um das Fenster "Git Commit" zu schließen.

4.3.4 Git Push

Den Button "Push" finden Sie in der Git Registerkarte:



Klicken Sie auf den Button "Push" und es öffnet sich automatisch das Fenster "Git Push":

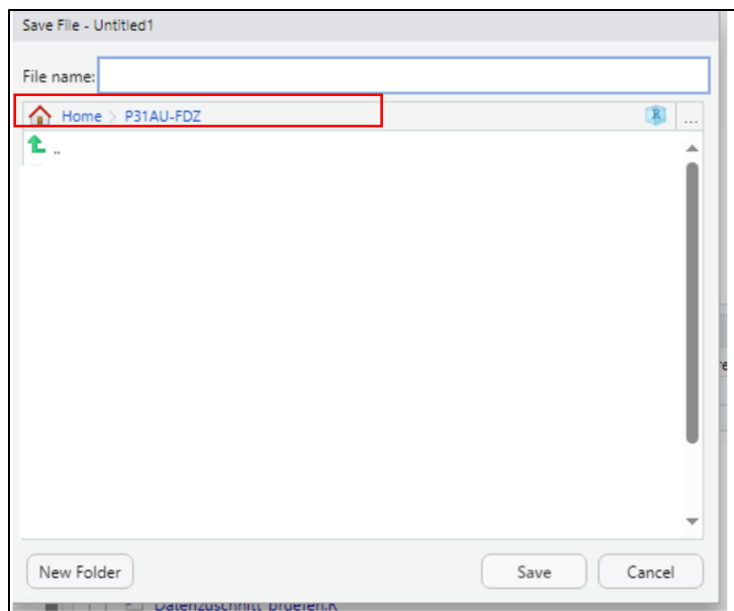


Klicken Sie auf "Close", um das Fenster "Git Push" zu schließen.

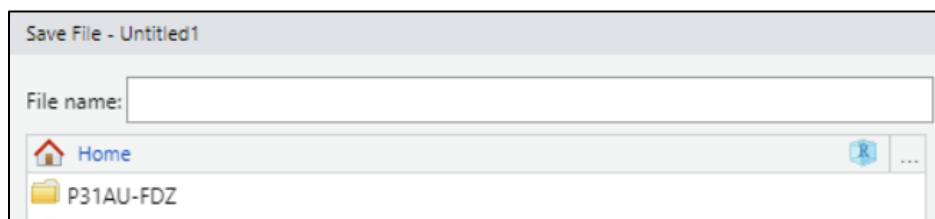
4.3.5 Nutzung des R Projekts (.Rproj)

Die Nutzung von R Projekten bzw. den .Rproj Dateien hilft, alle nötigen Informationen zum Bearbeiten eines Projektes an einem Ort abzuspeichern. Das bedeutet, dass alle nötigen Dateien und Daten eines Projektes in einem Ordner organisiert sind. Durch das Öffnen dieser Datei wird automatisch Ihre Arbeitsumgebung hergestellt, z.B. das Homeverzeichnis, die R-Version sowie die Git Repository Einstellung. Im FDZ nimmt die .Rproj aufgrund der Git Anbindung eine ganz wichtige Rolle ein, da dadurch gewährleistet wird, dass die Anbindung mit Ihrem Git Repository hergestellt wird. Daher sollte **nur** die von uns bereitgestellte .Rproj Datei verwendet werden und **keine zusätzlichen Projekte erzeugt werden**.

Das bedeutet auch, dass Skripte, die Sie abspeichern, immer im Ordner dieser Datei abgespeichert werden sollten. Normalerweise sollten Sie automatisch in diesem Ordner landen, wenn Sie Skripte abspeichern:



Sollten Sie jedoch einmal hinter dem Home Symbol keinen Namen sehen, dann klicken Sie bitte auf die Datei, die nach Ihrem Analyseraum benannt ist:



5 Datenbankverbindung

5.1 Allgemeine Hinweise zum Arbeiten mit der Datenbankverbindung im Analyseraum

Sie arbeiten in Ihrem Analyseraum mit einer Hana Datenbankverbindung. Es ist zu empfehlen, die Schritte zur Datenaufbereitung direkt in der Datenbank mit SQL Befehlen auszuführen. Da das Skript zum Erzeugen von Zwischen- und Endergebnissen auf den Echtdateien ausgeführt wird, wäre das Laden der Datenzuschnitte in das Environment zu ressourcenintensiv.

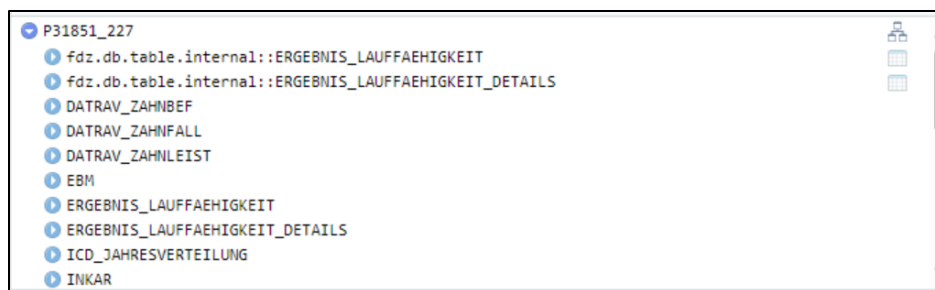
Sie können die SQL Befehle entweder in einem separaten SQL Skript ausführen, welches jedoch aus einem main.R oder main.Rmd aufgerufen wird oder direkt in Ihrem R/RMarkdown Skript ausführen. Die Basis-Befehle zum Anzeigen, Laden und Zusammenführen von Daten werden Sie in den nachfolgenden Kapiteln finden.

5.2 Anzeigen des Datensatzzuschnittes

In Ihrem Analyseraum steht Ihnen ein antragspezifischer Datenzuschnitt zur Verfügung, mit dem Sie Ihre Analyseskripte entwickeln können. Diesen Datenzuschnitt finden Sie in verschiedenen Datentabellen innerhalb Ihres antragspezifischen Schema. Sie finden das Schema in der Registerkarte **"Connections"** oben rechts im Environment Fenster.



Sie können sich die verschiedenen Datentabellen anzeigen lassen, indem Sie das Schema über den blauen Pfeil links vom Namen ausklappen:



Für Ihre Bearbeitung sind zunächst nur die Tabellen mit dem Präfix "DATRAV" relevant. Dort finden Sie die Variablen, die Sie im Antrag ausgewählt haben.

5.2.1 Inhalt der Datentabellen anzeigen lassen

Diese initial in Ihren Analyseraum beladenen Datentabellen sind besonders geschützt, sodass die Inhalte nicht, wie gewohnt, über das Tabellen Symbol , welches gewöhnlich rechts neben der Datentabelle zu sehen ist, einsehen können. Mit folgendem Befehl können Sie sich die ersten Zeilen einer bestimmten Datentabelle anzeigen lassen:

```
dbGetQuery(CONN, "SELECT * FROM P31851_187.DATRAV_SA151 LIMIT 10")
```

Die Ergebnisse werden in der Console angezeigt:

```
> dbGetQuery(CONN, "SELECT * FROM P31851_187.DATRAV_SA151 LIMIT 10")
  SA151_BERICHTSJAHR SA151_GEBURTSJAHR SA151_GESCHLECHT SA151_PSID SA151_SATZART SA151_VERSICHERTENTAGE SA151_VERSTORBEN SA151_VSID
1          2017         1935             1      <NA>          151              365              0 128146110
2          2017         1935             1      <NA>          151              365              0 131635769
3          2017         1935             1      <NA>          151              365              0 128146218
4          2017         1935             1      <NA>          151              365              0 131634572
5          2017         1935             1      <NA>          151              365              0 131634392
6          2017         1935             1      <NA>          151              365              0 131635515
7          2017         1935             1      <NA>          151              365              0 128147194
8          2017         1935             1      <NA>          151              365              0 131636537
9          2017         1935             1      <NA>          151              365              0 131634629
10         2017         1935             1      <NA>          151              365              0 131636780
```

5.3 Zusammenstellen neuer Datentabellen

Sie werden bereits festgestellt haben, dass Ihre Daten nicht in einer einzigen Datei mit allen notwendigen Informationen vorliegen, sondern sich auf viele verschiedene Datentabellen verteilen. Daher empfehlen wir Ihnen, für Ihre Analysen eine umfassende Datentabelle zu erstellen, die alle für Ihre Analysen relevanten Variablen enthält.

Vielleicht sind Sie gewohnt, eine solche Datenaufbereitung mit R Befehlen auszuführen.

Wir empfehlen, solche analysevorbereitenden Arbeiten mit SQL Befehlen auszuführen, da diese am effizientesten und Ressourcen schonendsten sind. Denn wenn Ihr Skript auf den Echtdateien ausgeführt wird, werden sehr viel mehr Ressourcen benötigt. Damit eine effiziente und zeitsparende Bearbeitung Ihres Skriptes gewährleistet werden kann, werden daher SQL Befehle empfohlen.

Nachfolgend finden Sie ein Beispiel, in dem aus dem Stammdatensatz (DATRAV_SA151) die Variablen Geschlecht (SA151_GESCHLECHT) und Geburtsjahr (SA151_GEBURTSJAHR) mit den Variablen ICD Code (SA651_ICD_CODE) und Berichtsjahr (SA651_BERICHTSJAHRAHR) aus dem Datensatz für ambulante Diagnosen (DATRAV_SA651) verbunden wurden.

Anschließend wurde diese neue Datentabelle in die Datenbank geschrieben.

Beispielbefehl Join

```
DATEN_GESAMT <- dbGetQuery(CONN, "
SELECT
  SA151_GESCHLECHT AS SA151_GESCHLECHT,
  SA151_GEBURTSJAHR AS SA151_GEBURTSJAHR,
  SA451_BERICHTSJAHRAHR AS BERICHTSJAHRAHR_451,
  SA551_ICD_CODE AS ICD_CODE_551,
  SA551_BERICHTSJAHRAHR AS BERICHTSJAHRAHR_551
FROM
  DATRAV_SA151
LEFT JOIN
  DATRAV_SA451
ON
  SA151_PSID = SA451_PSID
LEFT JOIN
  DATRAV_SA551 SA551
ON
  SA151_PSID = SA551_PSID
")

# Datentabelle in die Datenbank schreiben
dbWriteTable(CONN, "DATEN_GESAMT ", DATEN_GESAMT , overwrite = TRUE)
```

5.4 Schreiben dieser Datentabelle ins Environment

Wenn Sie eine Datentabelle erzeugt haben, mit der Sie weitere Analysen wie z.B. Regressionsmodelle rechnen wollen, können Sie dies entweder direkt in der Datenbank machen oder diese Tabelle ins Environment schreiben.

```
DATEN_GESAMT <- dbGetQuery(CONN, "SELECT * FROM DATEN_GESAMT")
```

5.5 Schreiben von Tabellen in die Datenbank

Ergebnisse, die Sie als Zwischenergebnisse bzw. Endergebnisse ausgegeben bekommen möchten, müssen Sie als "RT_"-Tabellen in die Datenbank schreiben. Dafür müssen Sie erzeugte Tabellen zuerst als Datentabelle abspeichern und anschließend in die Datenbank schreiben:

```
#Ergebnisse in RT Tabellen schreiben
RT_RESULT <- as.data.frame(summary(logistic_model)$coefficients)

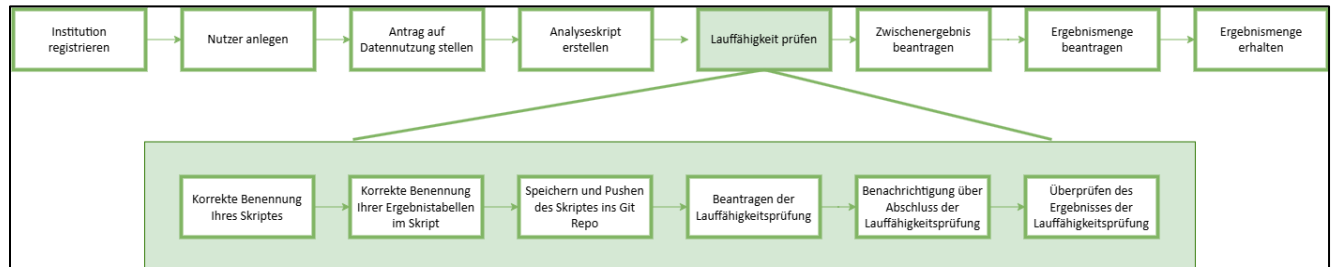
#RT_Tabelle in die Datenbank schreiben
dbWriteTable(CONN, "RT_RESULT", RT_RESULT, overwrite = TRUE)
```

SQL hat bestimmte Voraussetzungen für das Schreiben bzw. Abrufen von Datentabellen. Um Probleme zu vermeiden, beachten Sie bitte folgende Hinweise:

Schreiben Sie Tabellen am besten in Großbuchstaben in die Datenbank, um diese komplikationslos anzeigen zu lassen. Alternativ können Sie diesen Tabellennamen beim Abrufen der Daten (z.B. dbGetQuery) in Anführungszeichen setzen. Dadurch ergeben sich folgende Änderungen im Befehl, z.B.

```
dbGetQuery(CONN, 'SELECT * FROM P31851_43."RT_Result"')
```

6 Prozess Lauffähigkeitsprüfung



Die Prüfung der Lauffähigkeit Ihrer Skripte ist ein sehr wichtiger Arbeitsschritt in Ihrer Antragsbearbeitung. Wie Sie wissen, entwickeln Sie Ihre Skripte - je nach Antrag - entweder auf einem synthetischen Datensatz, der auf Echtdaten basiert, oder auf einem stark pseudonymisierten Datenauszug. Für die Erzeugung Ihrer echten Endergebnisse ist es notwendig, dass Ihre Skripte auch auf dem Echtdatensatz ausgeführt werden können. Sie können die Lauffähigkeit Ihrer Skripte bis zu 10x überprüfen lassen. Bitte beachten Sie hierbei, dass Sie für die Ausgabe Ihrer Zwischenergebnisse sowie Endergebnisse erst eine Lauffähigkeitsprüfung Ihres Skriptes absolvieren müssen. Das bedeutet, dass die Zwischenergebnisse und auch Endergebnisse auf dem letzten Skript, welches für die Lauffähigkeitsprüfung verwendet wurde, basieren.

Im Folgenden erklären wir Ihnen, welche Schritte Sie befolgen müssen, damit die Lauffähigkeit Ihrer Skripte erfolgreich durchlaufen. Bitte lesen Sie zusätzlich die [Programmcoderichtlinien](#), um sicherzustellen, dass Sie die notwendigen Vorgaben zur Benennung Ihres Skriptes sowie Tabellen berücksichtigen.

6.1 Korrekte Benennung Ihres Skriptes

Wie in den [Programmcoderichtlinien](#) erklärt, müssen Sie Ihr Skript, welches Sie für die Erzeugung von Zwischen- und Ergebnismengen verwenden wollen, je nach ausgewählter Programmiersprache **main.r/main.rmd** benennen. Die korrekte Benennung ist für die Ausführung der Lauffähigkeitsprüfung relevant, da es sich um einen automatisierten Prozess handelt und kein Skript ausgewählt wird, welches nicht diesen Namen besitzt. Daher stellen Sie sicher, dass alle Analysen, die Sie für die Lauffähigkeitsprüfung bzw. Zwischen-/Endergebnisse verwenden möchten, in diesem main-Skript enthalten sind.

6.2 Korrekte Benennung Ihrer Ergebnistabellen

Die Ergebnistabellen, die Sie mit Ihrem Skript erzeugen, müssen ebenfalls einer Namenskonvention entsprechen. Nur erzeugte Tabellen, die das Präfix "RT_" besitzen,

können als Zwischen- oder Endergebnisse ausgegeben werden. Daher ist zu empfehlen, diese Namenskonventionen bereits in Ihren Skripten, die Sie für die Lauffähigkeitsprüfung verwenden, umzusetzen. Bitte beachten Sie, dass diese Tabellen in die Datenbank geschrieben werden müssen und nicht ins Environment (s. [Tabellen in Datenbank](#)).

6.2.1 Erstellen und Abspeichern von Tabellen zur Beschreibung der Ergebnistabellen (DD Tabellen)

Die Beschreibung der Ergebnismenge muss innerhalb von sog. Data Description Tabellen (DD Tabellen) erfolgen. Das von Ihnen eingereichte Skript muss zwei Tabellen mit dem Präfix DD_ erzeugen, welche unten dargestellte Form haben:

6.2.2 DD_RESULT_TABLES (Beispiel)

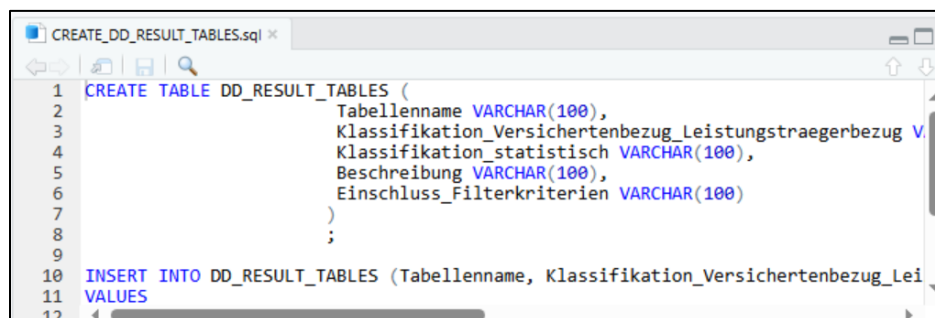
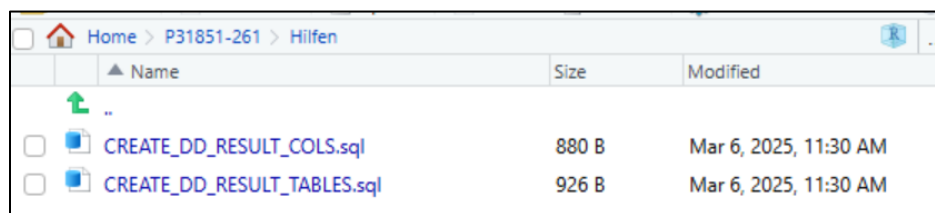
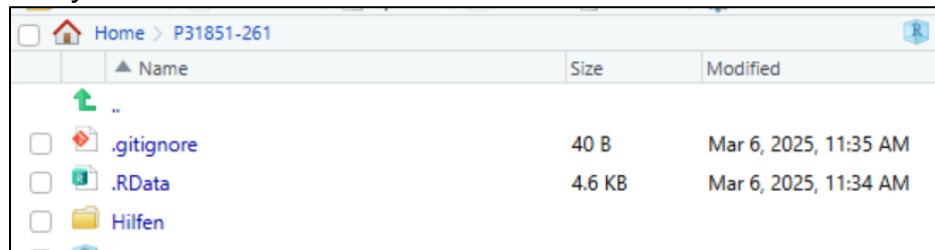
Tabelle n name	Klassifikation_Versichertenbe- zug _Leistungstraegerbezug	Klassifikation _statistisch	Beschreibu ng	Einschluss _Filterkriterien
RT_1	Versichertenbezug	Mittelwert	Kennzahl: Mittelwert des Alters der betrachtete n Versicherte n	Nur Versicherte der Geburtsjahrgän- ge 1980-1989
RT_2	Versichertenbezug	Häufigkeitstabe- lle	Kreuztabelle aus Altersgrupp e und Geschlecht	Alle beantragten Versicherten

6.2.3 DD_RESULT_COLS (Beispiel)

Tabellenname	Klassifikation_statistisch	Spaltenname	Beschreibung
RT_1	Mittelwert	MW_Alter	Mittelwert des Alters der betrachteten Versicherten
RT_1	Fallzahl	CNT_D_PSID	Fallzahl der Versicherten

RT_2	Häufigkeitstabelle	Altersgruppe	Altersgruppe der betrachteten Versicherten
RT_2	Häufigkeitstabelle	Geschlecht	Geschlecht der betrachteten Versicherten
RT_2	Fallzahl	CNT_D_PSID	Fallzahl der Versicherten

Beispielskripte zum Erstellen dieser Tabellen finden Sie im Homeverzeichnis Ihres Analyse-raums im Ordner "Hilfen"



6.3 Speichern und Pushen Ihres Skriptes ins Git Repository

Damit Ihr Skript für die Lauffähigkeitsprüfung verwendet werden kann, müssen Sie dieses in der Version, die Sie verwenden wollen, in Ihr Git Repository pushen. Dafür gehen Sie bitte wie im Kapitel 3.3 Arbeiten mit dem Git Repository vor.

6.4 Beantragen der Lauffähigkeitsprüfung im Antragsportal

Sie haben nun alle relevanten Schritte in Ihrem Analyseraum vorgenommen, um eine Lauffähigkeit Ihres Skriptes beantragen zu können. Die Lauffähigkeitsprüfung können Sie nun im Antragsportal beantragen.

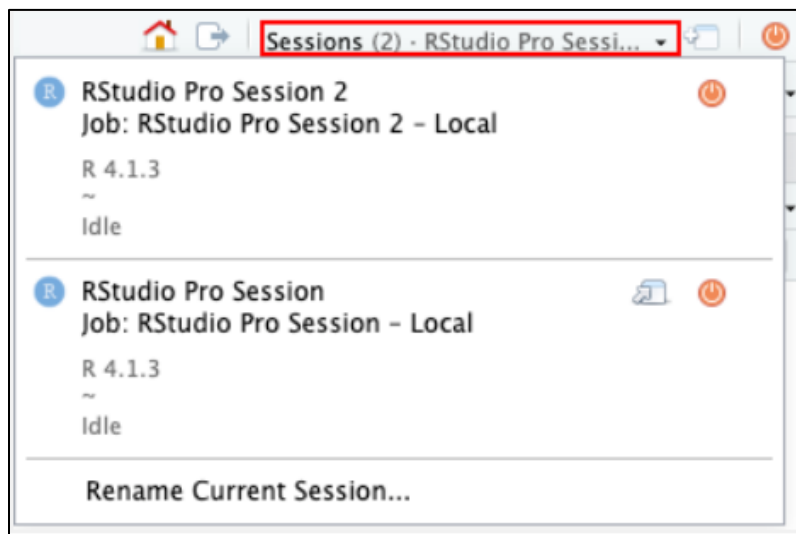
6.5 Einsehen der Ergebnismeldung Ihrer Lauffähigkeitsprüfung

Sie erhalten eine E-Mail, sobald die Lauffähigkeitsprüfung abgeschlossen ist. Das Ergebnis finden Sie in der Registerkarte **Connections** in der Tabelle **"ERGEBNIS_LAUFFAEHIGKEIT"**. Dort können Sie einsehen, ob die Prüfung erfolgreich war oder welche Fehlermeldungen aufgetreten sind, z.B. bei Syntaxfehlern. Bei technischen Fehlern wird der Vermerk **"Technical Error"** angezeigt, und Ihnen wird kein Versuch abgezogen. **Stellen Sie vor der erneuten Beantragung jedoch in jedem Fall sicher, dass sich Ihr main.R oder main.Rmd Skript im Git Repository befindet.**

7 Beenden und Management von Sessions

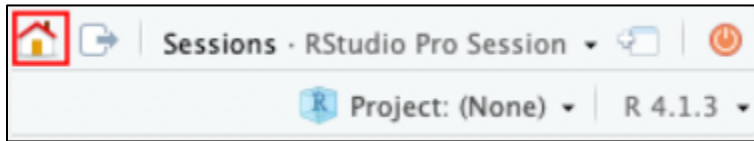
7.1 Wechsel der R-Studio Pro-Sitzung

Sie können jederzeit zwischen Ihren Sessions wechseln. Wählen Sie dazu im Dropdown-Menü "Sessions" oben rechts im Browserfenster die gewünschte Session aus. Diese wird dann automatisch im aktuellen Tab geöffnet.



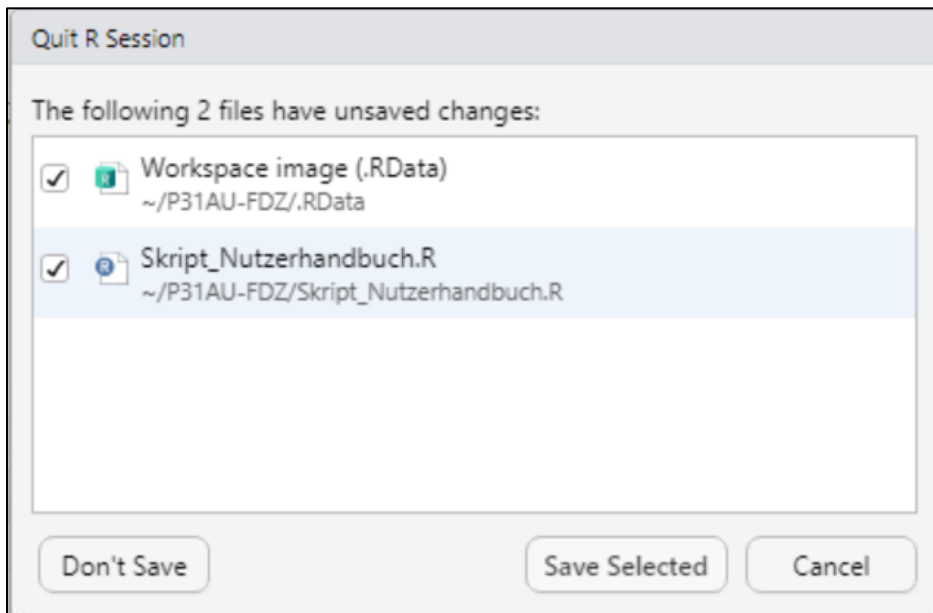
7.2 Beenden einer Session

Wenn Sie zur Posit Workbench-Startseite zurückkehren möchten und zu einem späteren Zeitpunkt die Arbeit wiederaufnehmen möchten, klicken Sie auf das **Home-Symbol**  oben rechts im Browserfenster. Dadurch wird die Posit Workbench-Startseite im aktuellen Browser Tab geöffnet.



Wenn Sie auf die Workbench Seite zurückkehren und die aktuelle Session beenden möchten, können Sie auf das rote Quit Symbol  klicken.

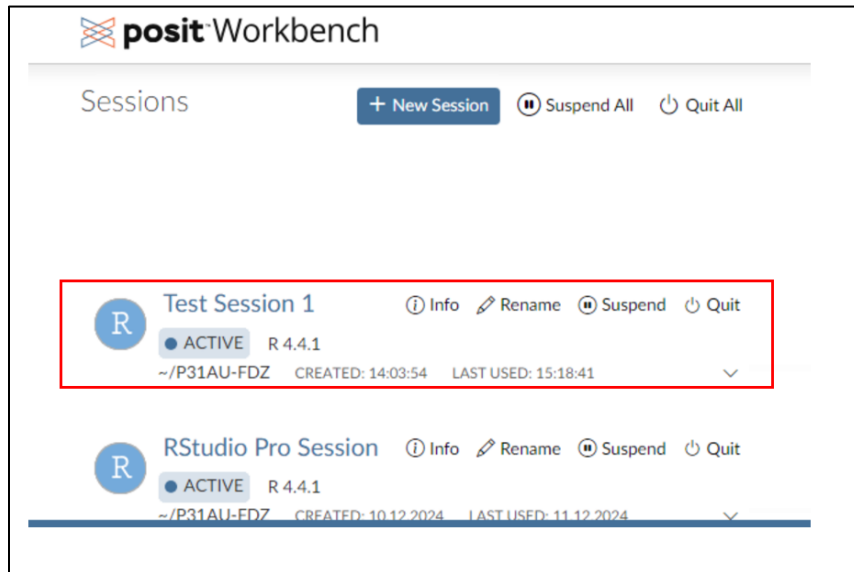
In diesem Fall, werden Sie gefragt, ob Sie Ihren Workspace speichern können. Das Speichern Ihres Workspaces ermöglicht, dass Sie den aktuellen Zustand Ihrer Arbeitsumgebung sichern, einschließlich aller benutzerdefinierten Objekte wie Vektoren, Matrizen, Datenrahmen, Listen und Funktionen.



In diesem Beispiel sehen Sie zum Beispiel, dass ein Skript Änderungen enthält, die noch nicht abgespeichert sind. Daher sollten Sie vor Beenden dieser Session die hier aufgeführten Skripte speichern und in Ihr Git Repository pushen, um sicherzustellen, dass Sie zu einem späteren Zeitpunkt dort weiterarbeiten können.

7.3 Wiederaufnahme einer Session

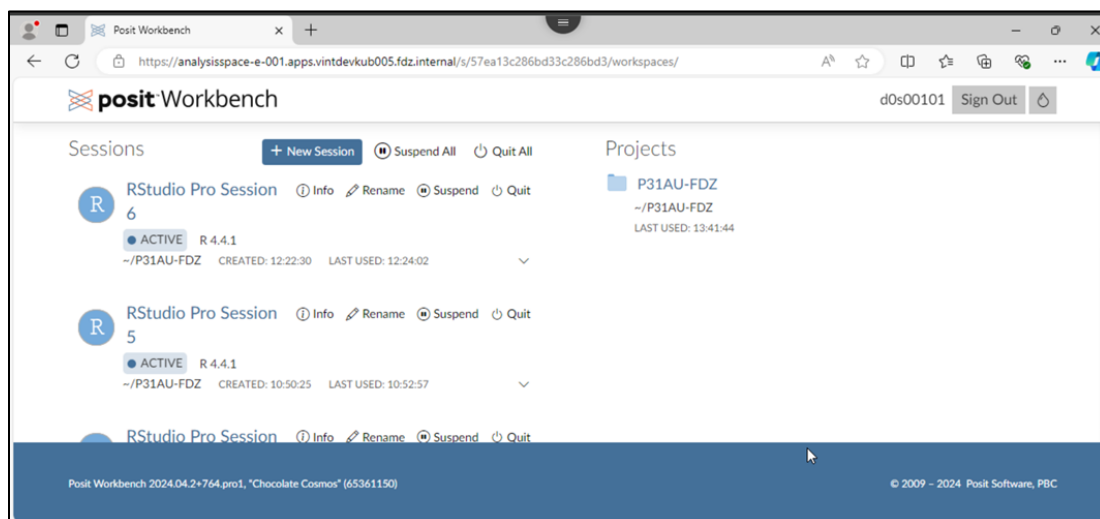
Sie müssen bei Fortsetzung Ihrer Arbeit nicht jedes Mal eine neue Session beginnen, sondern können durch einfaches Anklicken einer vorhandenen Session die Arbeit fortsetzen.



8 Optionale Funktionen

8.1 Verwalten Ihrer Sessions

Auf Ihrer Workbench sehen Sie alle laufenden Sessions.



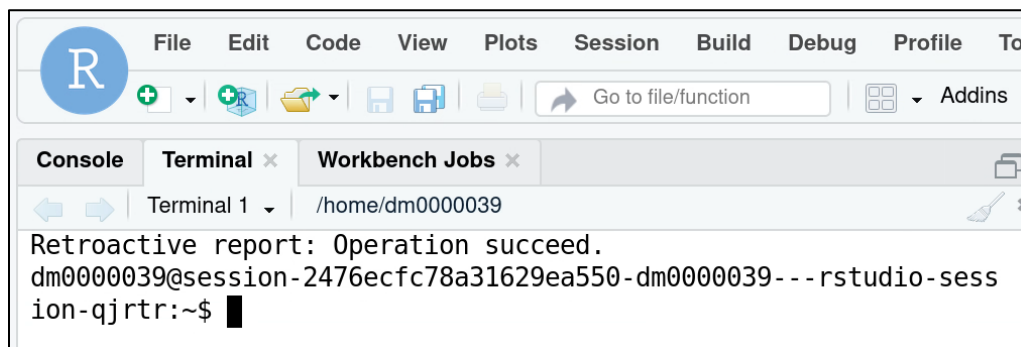
Sie können unter "**Rename**" den Namen Ihrer Session ändern. Durch Klicken von "**Suspend**" können Sie Ihre Session pausieren. Das kann beispielsweise hilfreich sein, wenn Sie in einer anderen Session eine Analyse mit hohem

Ressourcenverbrauch starten wollen. Durch das Pausieren der anderen Sessions können Ressourcen eingespart werden, um die Lauffähigkeit der Analyse sicherzustellen. Über "Suspend All", was Sie oben neben dem Button "+ New Session" finden können, werden alle aktiven Sessions pausiert.

Durch das Klicken von "Quit" wird Ihre Session geschlossen. Ihre Dateien im Homeverzeichnis bleiben jedoch erhalten und sind über möglicherweise noch laufende Sessions oder neu gestartete Sessions weiterhin erreichbar. Durch das Klicken von "Quit All" neben dem Button "Suspend All" können Sie alle laufenden Sessions schließen.

8.2 Git Projekt Zugriff in RStudio: Terminal

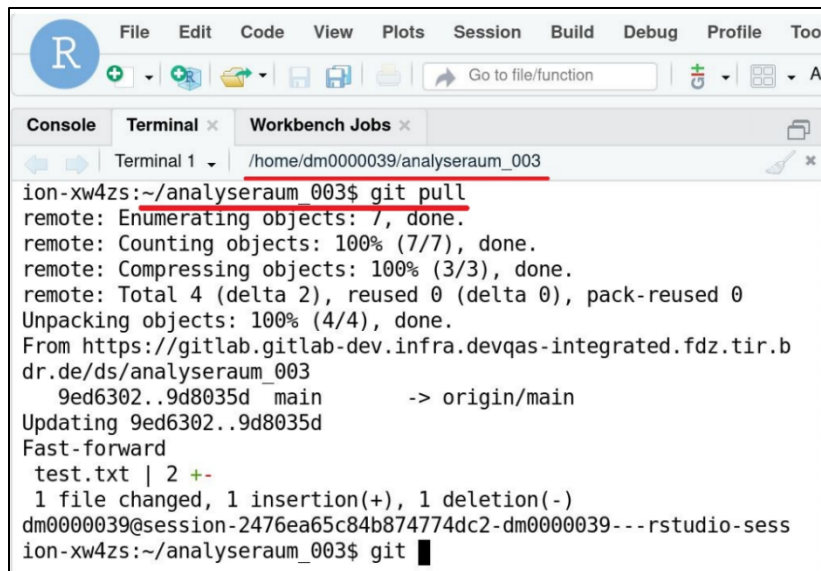
Wenn Sie den Terminal in Ihrer RStudio Workbench Session öffnen, wird Ihr Homeverzeichnis angezeigt.



Wechseln Sie im Terminal in Ihren Git Projektordner "analyseraum_00X" (wobei X für 1, 2 oder 3 steht). Verwenden Sie dazu folgenden Befehl:

```
cd analyseraum_00X
```

Sie können alle Git Kommandos, z.B. "git branch", "git add", "git commit", "git pull", "git push" usw. direkt auf dem Terminal in Ihrem Analyseraum aufrufen.



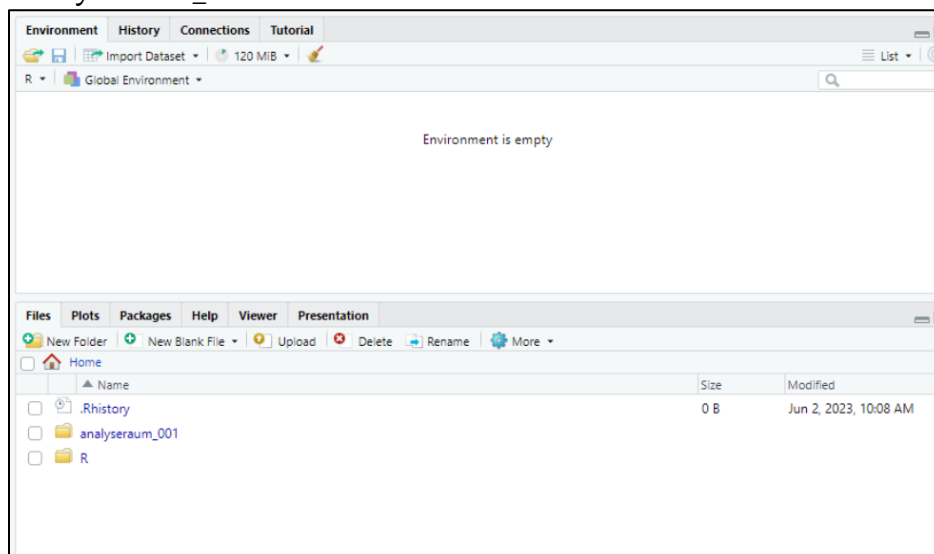
```
ion-xw4zs:~/analyseraum_003$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://gitlab.gitlab-dev.infra.devqas-integrated.fdz.tir.b
dr.de/ds/analyseraum_003
   9ed6302..9d8035d  main      -> origin/main
Updating 9ed6302..9d8035d
Fast-forward
 test.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
dm0000039@session-2476ea65c84b874774dc2-dm0000039---rstudio-sess
ion-xw4zs:~/analyseraum_003$ git
```

9 FAQ

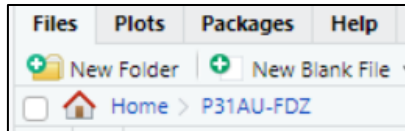
9.1 Meine Gitanbindung ist nicht im User interface zu sehen

Sollte die automatische Git Anbindung einmal nicht funktionieren, kontaktieren Sie bitte den Support, um das Problem zu melden. In der Zwischenzeit können Sie die **Git Anbindung manuell herstellen**, um weiterarbeiten zu können.

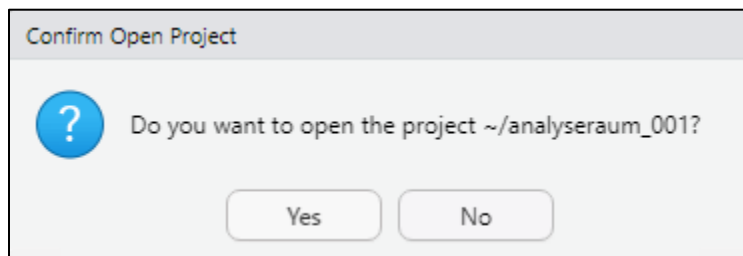
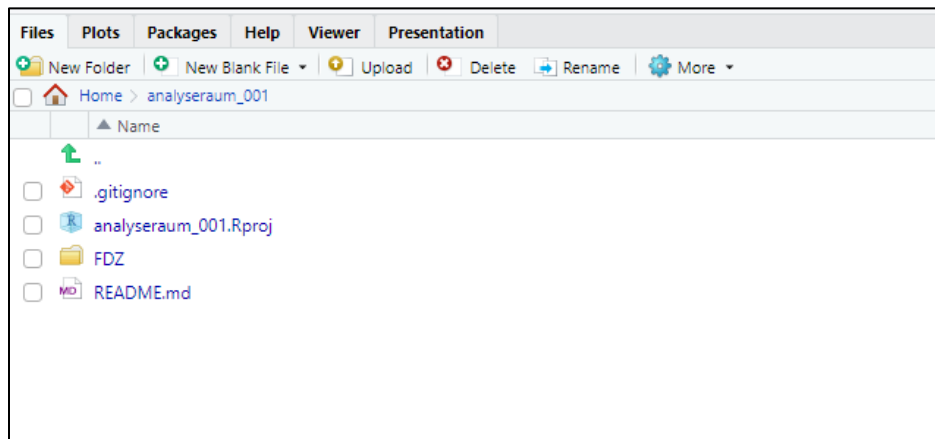
Dafür suchen Sie in Ihren Files im Fenster unten rechts einen Ordner mit "analyseraum_00X":



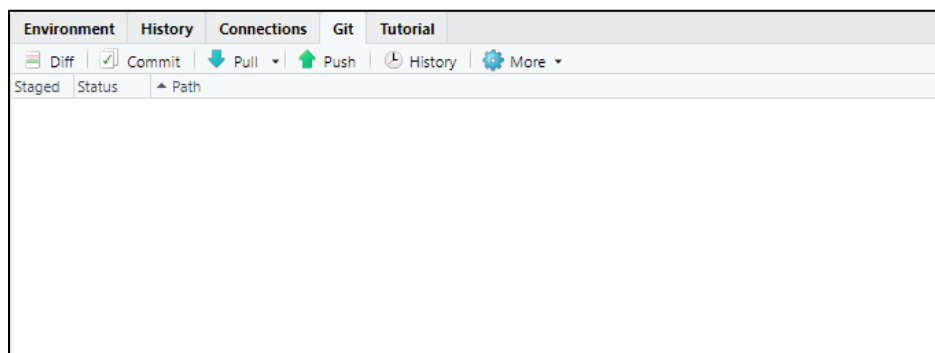
Es kann sein, dass Sie sich bereits in diesem Ordner befinden. Das sehen Sie, indem Sie schauen, ob neben dem Haussymbol nachfolgend hinter "Home" den Namen dieses Ordners sehen



Anschließend klicken Sie auf die Datei mit der Endung "Rproj" (z.B. "analyseraum_001.Rproj"):

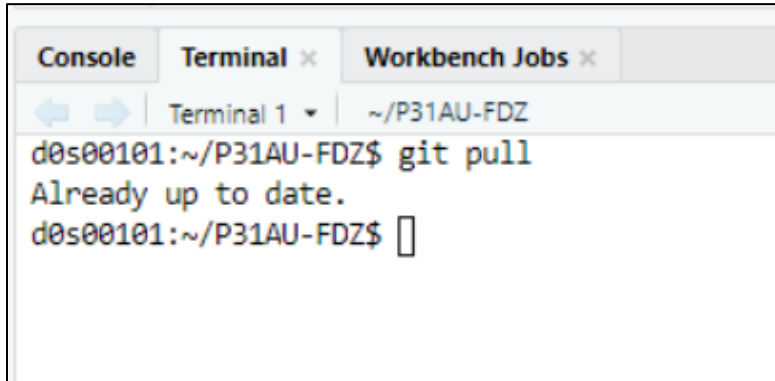


Wenn Sie mit dem "Yes" Button bestätigen, wird der Git Projekt Workspace geöffnet:



Achtung: Falls die "analyseraum_00X.Rproj" Datei unter Ihrem Projektverzeichnis nicht vorhanden ist, können Sie über das Terminal die Git Befehle ausführen. Bitte kontaktieren Sie zusätzlich unbedingt den Support, dass die Projektdatdatei fehlt.

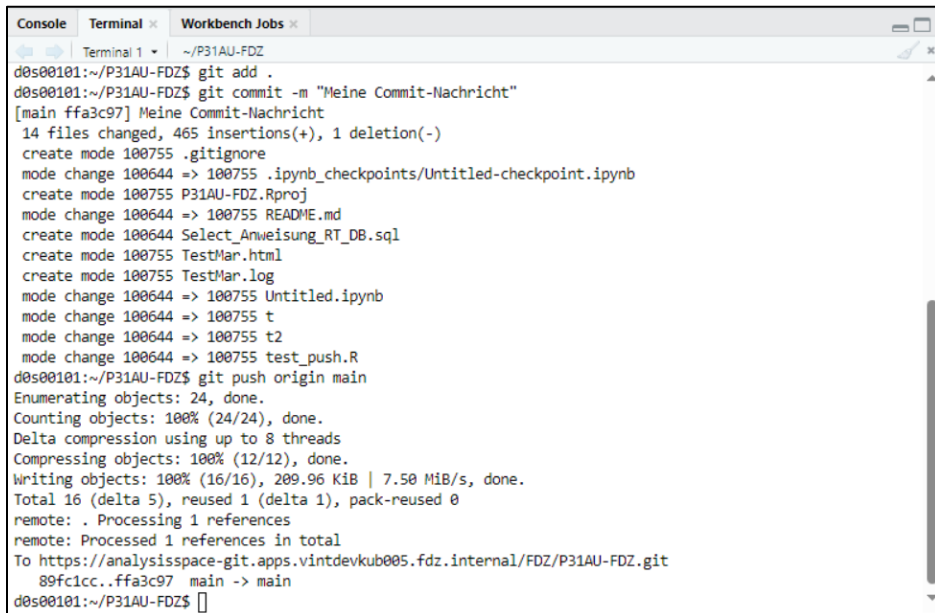
Zum **Pullen** des aktuellen Git Repository Stands, geben Sie bitte "git pull" als Befehl im R-Terminal der Workbench (zweite Registerkarte nach "Console") ein und drücken auf Enter.



```
Console Terminal x Workbench Jobs x
Terminal 1 ~ /P31AU-FDZ
d0s00101:~/P31AU-FDZ$ git pull
Already up to date.
d0s00101:~/P31AU-FDZ$
```

Wenn Sie Ihre Dateien ins Git Repository **pushen wollen**, geben Sie zunächst ins Terminal *git add .* und anschließend *git commit -m "Meine Commit-Nachricht"* ein. In die " " können Sie eine individuelle Nachricht schreiben, die Ihnen für Ihre Versionskontrolle hilft.

Über *git push origin main* werden die Dateien ins Git Repository gepushed.



```
Console Terminal x Workbench Jobs x
Terminal 1 ~ /P31AU-FDZ
d0s00101:~/P31AU-FDZ$ git add .
d0s00101:~/P31AU-FDZ$ git commit -m "Meine Commit-Nachricht"
[main ffa3c97] Meine Commit-Nachricht
14 files changed, 465 insertions(+), 1 deletion(-)
create mode 100755 .gitignore
mode change 100644 => 100755 .ipynb_checkpoints/Untitled-checkpoint.ipynb
create mode 100755 P31AU-FDZ.Rproj
mode change 100644 => 100755 README.md
create mode 100644 Select_Anweisung_RT_DB.sql
create mode 100755 TestMar.html
create mode 100755 TestMar.log
mode change 100644 => 100755 Untitled.ipynb
mode change 100644 => 100755 t
mode change 100644 => 100755 t2
mode change 100644 => 100755 test_push.R
d0s00101:~/P31AU-FDZ$ git push origin main
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 209.96 KiB | 7.50 MiB/s, done.
Total 16 (delta 5), reused 1 (delta 1), pack-reused 0
remote: . Processing 1 references
remote: Processed 1 references in total
To https://analysisspace-git.apps.vintdevkub005.fdz.internal/FDZ/P31AU-FDZ.git
89fc1cc..ffa3c97 main -> main
d0s00101:~/P31AU-FDZ$
```

9.2 Ich kann ein wichtiges Paket nicht installieren bzw. finde es nicht in der packages Registerkarte

Durchsuchen der installierten packages (ls und grep Befehl in Kombination).

Sie können alle Funktionen eines Pakets laden und dann mit grep filtern. Hier wird die Funktion am Beispiel des ARIMA-Modells (AutoRegressive Integrated Moving Average), welches ein leistungsfähiges statistisches Werkzeug zur Zeitreihenanalyse und -vorhersage darstellt. Diese Funktion wird u.a. üblicherweise im forecast package unterstützt, welches jedoch nicht installiert ist. Durch folgende Funktion kann geprüft werden, ob alternative Pakete mit ähnlicher Funktion installiert sind:

Funktionalität in Paketen suchen

```
# Alle installierten Pakete auflisten
installed_packages <- installed.packages()[, "Package"]

# Suche nach Funktionen mit "arima" im Namen
search_arima_functions <- function(keyword) {
  sapply(installed_packages, function(pkg) {
    tryCatch(
      grep(keyword, ls(getNamespace(pkg)), value = TRUE),
      error = function(e) NULL
    )
  })
}

# Nach "arima" suchen
arima_results <- search_arima_functions("arima")

# Nur Pakete mit Treffern anzeigen
arima_results <- arima_results[lengths(arima_results) > 0]

# Ausgabe
print(arima_results)
```

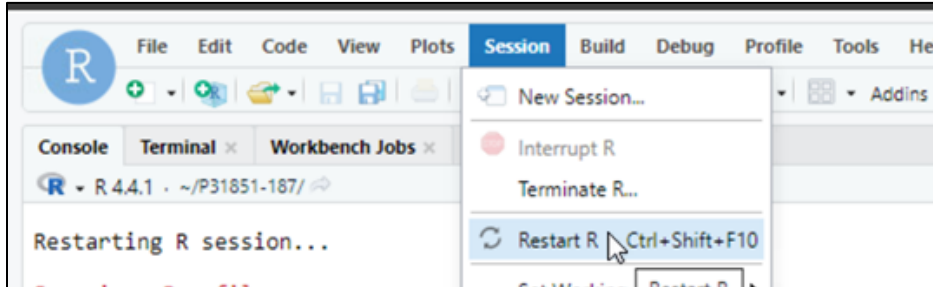
In der Console werden dir die Pakete ausgegeben, welche diese Funktion beinhalten:

```
> # Ausgabe
> print(arima_results)
$stats
[1] "arima"      "arima.sim"  "arima0"     "arima0.diag" "predict.arima0"
[6] "print.arima0" "tsdiag.arima0" "varimax"
```

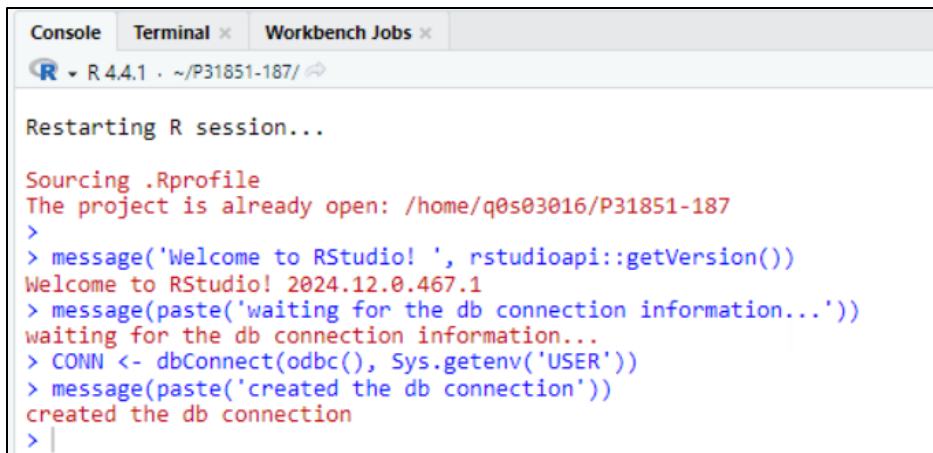
Sollte Ihre gesuchte Funktion in keinem installierten package vorkommen, kontaktieren Sie bitte den Support.

9.3 Datenbankverbindung über Neustart der Session

Manchmal wird die Session nicht automatisch komplett aufgebaut. In diesem Fall klicken Sie auf "Session" in der oberen Menüleiste **Session** und anschließend auf "Restart R"



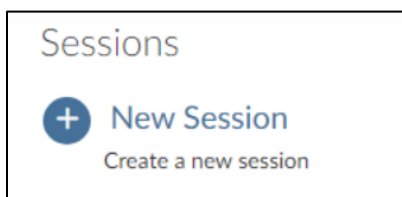
Anschließend sollten Sie in der Console sehen, dass folgender Befehl erfolgreich ausgeführt wurde:



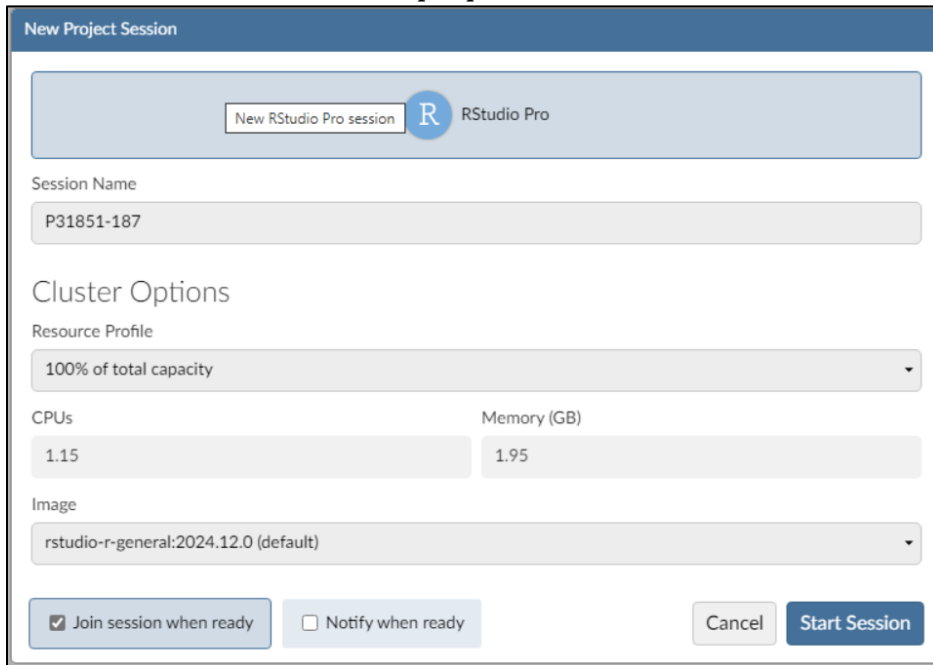
Sollte dieser Befehl nicht ausgeführt worden sein oder Sie finden Ihre Datenschema dennoch nicht in der Connections Registerkarte, kontaktieren Sie bitte den FDZ Support.

9.4 Starten einer Session ohne Projekt

Wenn Sie in Ihrem Analyseraum noch nie eine Session gestartet haben, kann es sein, dass unter "Project" noch kein Link zu Ihrem Analyseraum vorhanden ist. In diesem Fall klicken Sie bitte auf "New Session"



Anschließend erscheint ein Pop-up Fenster:

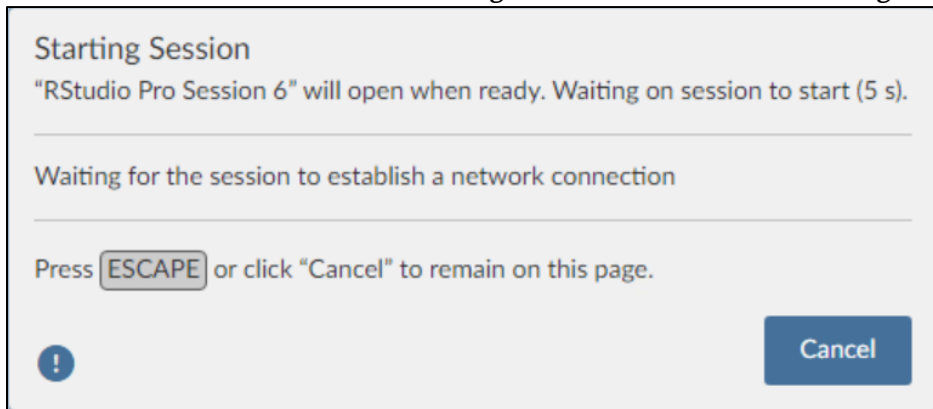


Unter "Session Name" können Sie Ihre Sessions individuell benennen. Das kann nützlich sein, wenn Sie beispielsweise verschiedene langlaufende Skripte parallel in verschiedenen Sessions laufen lassen wollen.

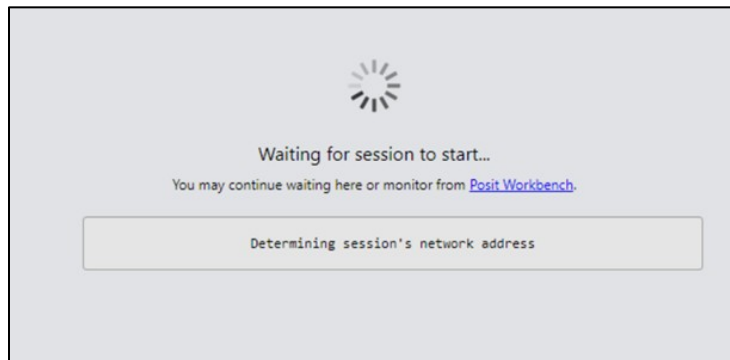
Unter Resource Profile können Sie angeben, wie viel Kapazität der für Sie insgesamt zur Verfügung gestellten CPU Leistung Sie für diese Session maximal verwenden wollen. Wenn Sie nur in einer Session arbeiten wollen, können Sie 100% auswählen. Sollten Sie jedoch parallel in verschiedenen Sessions Analysen laufen lassen wollen, ist eine Aufteilung an dieser Stelle sinnvoll.

Klicken Sie anschließend auf "Start Session".

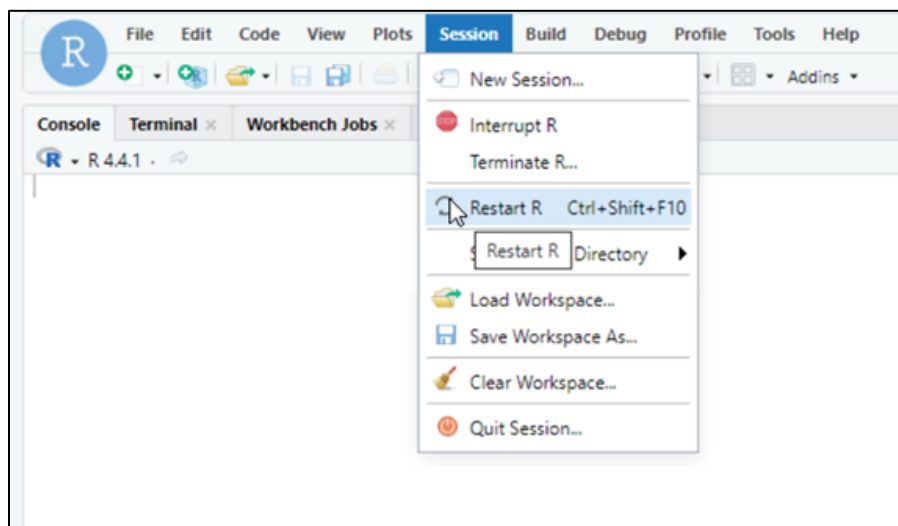
Nun wird Ihre Session automatisch gestartet. Das kann einen Augenblick dauern.



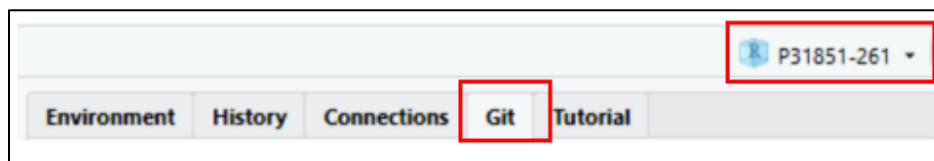
Beim Starten Ihrer Session wird automatisch Ihr R Projekt geladen. Das stellt sicher, dass Ihnen in Ihrem Analyseraum alle Funktionen inkl. der Datenbankverbindung sowie Git Repository Anbindung zur Verfügung stehen. Während dieses Prozesses erscheint kurzzeitig die Session und anschließend beginnt ein Neustart. Kurzzeitig erscheint ein solches Fenster:



Sollte dieses Fenster nicht automatisch erscheinen, klicken Sie bitte unter "Session" "Restart R"



Anschließend sollte das Projekt automatisch geladen sein. Sie sollten die Git Registerkarte sehen sowie den Namen Ihres Antrages:



9.5 Ich kann nichts in mein Git Repository pushen

Sollten Sie eine Fehlermeldung beim Pushen von Dateien in Ihr Git Repository erhalten, führen Sie zunächst den Pull Befehl aus und probieren es erneut. Sollte das Problem immer noch bestehen, notieren Sie sich am besten die Fehlermeldung und wenden sich an den FDZ-Support.

9.6 Ich kann meine erzeugten Datentabellen in der Datenbankverbindung nicht anzeigen lassen

Sollten Sie eine solche oder ähnliche Fehlermeldung beim Anzeigen einer Datentabelle in der Hana Datenbank erhalten:

```
> dbGetQuery(CONN, "SELECT * FROM P31851_39.b_base_populationtest")
Error in `dbGetQuery()`:
! ODBC failed with error 00000 from [SAP AG][LIBODBCCHDB SO][HDBODBC].
X Base table or view not found;259 invalid table name: Could not find table/view B_BASE_POPULATIONTEST in schema
P31851_39: line 1 col 25 (at pos 24)
• <SQL> "SELECT * FROM P31851_39.b_base_populationtest"
i From nanodbc/nanodbc.cpp:1722.
Run `rlang::last_trace()` to see where the error occurred.
>
```

Überprüfen Sie, ob Sie die Datentabelle in Großbuchstaben in die Datenbank geschrieben haben.

```
> dbWriteTable(CONN, "POPULATION_TEST", b_base_populationtest, overwrite=TRUE)
> dbGetQuery(CONN, "SELECT * FROM P31851_39.POPULATION_TEST")
  SA151_BERICHTSJAHR SA151_GEBURTSJAHR SA151_GESCHLECHT
1          2017          1990             1
2          2017          1990             1
3          2017          1990             1
4          2017          1990             1
5          2017          1990             1
6          2017          1990             1
7          2017          1990             1
8          2017          1990             1
9          2017          1990             1
```

9.7 Ich kann mein R Skript nicht im Antragsportal hochladen

Wenn Sie eine Fehlermeldung beim Upload eines R Skriptes erhalten, welches folgende Fehlermeldung erhält "The file content does not look like a r file"

*Pflichtfeld

Dokument hochladen

Kategorie*

Skript

Beschreibung (optional)

Verbleibende Zeichen:

 **Anzeigen_Datentabellen_R.R**
341 Bytes 

 **FEHLER**

Leider ist etwas schief gelaufen. Sollte der Fehler erneut auftreten, wenden Sie sich an Ihren Systemadministrator.

TraceID: 6889e8dd006612b4d8f33ee05b20a9f9

Message: The file content does not look like a r file

Time: 2025-07-30T09:41:51.072277348Z

Gelesen

Dateiname	Beschreibung
-----------	--------------

Dann liegt es daran, dass anteilig zu viele Befehle (z.B. sql-Befehle) enthalten sind, die kein R Code sind. Wenn zu viele SQL Befehle enthalten sind, bietet es sich daher an, den Inhalt des Skriptes im .sql Format hochzuladen und den Inhalt im Analyse Raum in ein R Skript zu übertragen.

9.8 Meine Datenbankverbindung in einer aktiven Session ist plötzlich unterbrochen

Aufgrund von Hintergrundprozessen kann es in seltenen Fällen passieren, dass die Datenbankverbindung neu aufgebaut wird. Daher warten Sie einige Minuten bis maximal wenigen Stunden. Sollte die Verbindung dann noch nicht wieder aufgebaut sein, kontaktieren Sie bitte den Support