Vitamin D: Datenanbindung über R an Google Schnittstellen

Vitamin D:Datenanbindung über R an Google

Schnittstellen – websiteboosting.com

R ist nicht nur Open Source (damit kostenlos) und anpassungsfähig durch vielfältige Pakete, sondern auch offen für andere (technische) Verbindungen. Diese Verbindungen, vulgo API, ermöglichen, flexibel Daten an der Ursprungsquelle abzufragen, entsprechend weiterzuverarbeiten und ggf. modifiziert…
R ist nicht nur Open Source (damit kostenlos) und anpassungsfähig durch vielfältige Pakete, sondern auch offen für andere (technische) Verbindungen. Diese Verbindungen, vulgo API, ermöglichen, flexibel Daten an der Ursprungsquelle abzufragen, entsprechend weiterzuverarbeiten und ggf. modifiziert wieder zurückzuliefern. Google bietet eine Vielzahl von API, die in R eingebunden eine enorme Individualisierung und Automatisierung von analytischen Prozessen und Visualisierungen ermöglichen. Die Excel-Alternative Google Sheets in Kombination mit Google Data Studio ist bereits eine unkomplizierte Lösung für vielfältige Visualisierungen. Gepaart mit der statistischen Leistung von R eröffnet sich durch die weitere Kombination mit API nochmals ein deutlicher Zugewinn an Flexibilität! Tobias Aubele zeigt, wie Sie diese Schnittstellen nutzen können.

R wird kontinuierlich von einer großen Community weltweit weiterentwickelt, Gleiches gilt für die Dienste von Google wie Maps & Co. Unglaubliche Mengen an Daten liegen in diversen Datensilos und „sehnen“ sich danach, verarbeitet bzw. interpretiert zu werden und einen Nutzen zu stiften. Durch die Vielzahl an Silos und steigenden Datenmengen bedarf es einer effektiven und effizienten Verarbeitung, damit letztlich genügend Zeit für die Interpretation der Daten und das Ableiten von Maßnahmen übrig bleibt. Obwohl ggf. sehr zeitaufwendig, wird in Fachbereichen gerne Microsoft Excel eingesetzt, da dort Daten mittels Formeln und Funktion aus diversen Quellen flexibel zusammengeführt und verarbeitet werden können. Leider sind Daten teilweise veraltet, ehe sie letztlich über manuelle Zusammenfassungen und Verdichtungen

für den eigentlichen Nutzungszweck zur Verfügung stehen. Sofern direkt API (Programmierschnittstellen) genutzt werden, können aktuelle Daten unmittelbar am Ursprung abgeholt und zurückgeliefert werden und bieten langfristig ggf. ein hohes Potenzial für automatisierte Abläufe.

Programmcode online verfügbar!

Damit Sie Codezeilen nicht einzeln abtippen müssen, sind zwei der Skripte aus diesem Beitrag direkt online für Sie zum Download hinterlegt:
Das Skript zum Knowlede Graph unter www.websiteboosting.com/code/r-knowledge
Das Skript für Google Sheets unter www.websiteboosting.com/code/r-sheets

Die API – aktuelle Daten direkt vom Ursprung

Über die Google API können diverse Google-Dienste direkt über R angesprochen werden. Nach einer Registrierung unter console.cloud.google.com kann die gewünschten API, wie bspw. die des Google Knowledge Graph, einzeln aktiviert und unmittelbar getestet bzw. genutzt werden (siehe Abb. 1).

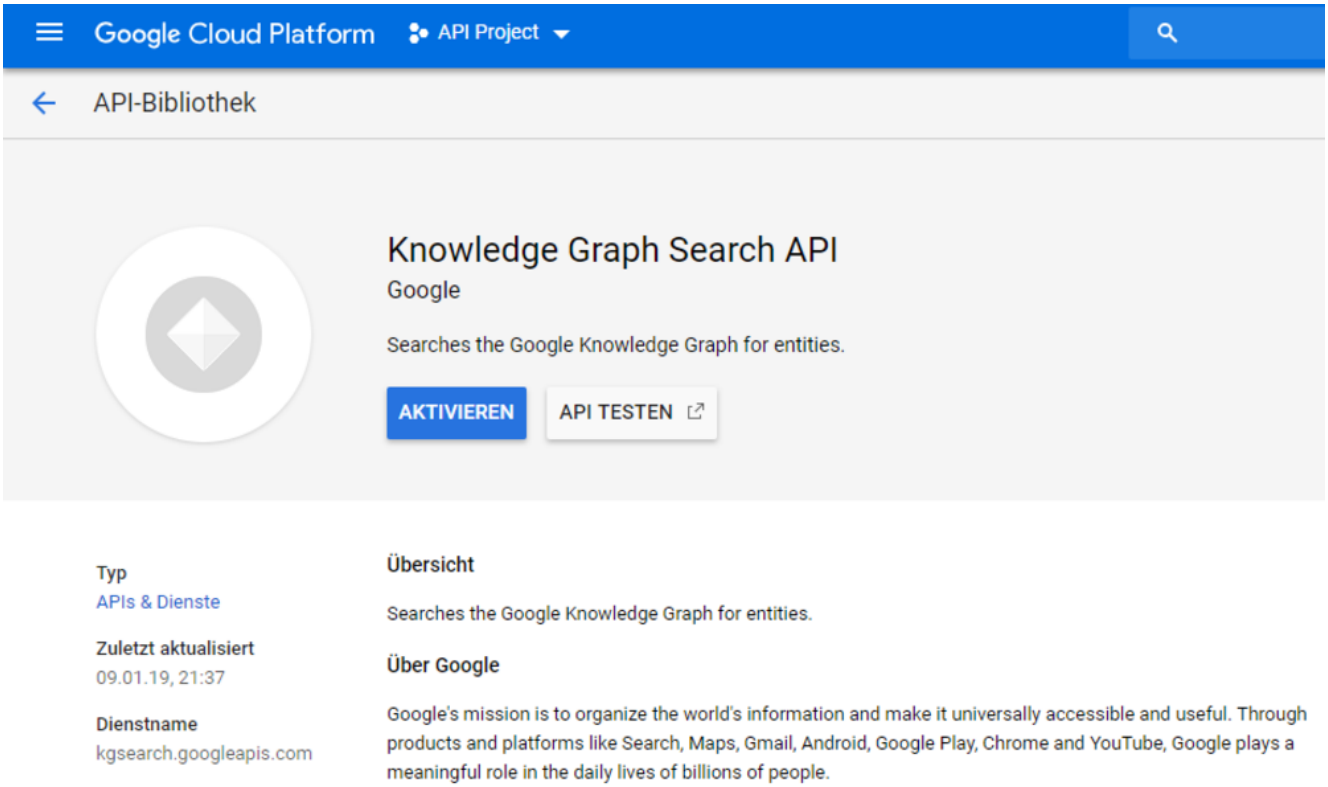


Abb. 1: Aktivierung einer Google API

Weiterhin können anschließend die notwendigen Anmeldedaten wie ein API-Schlüssel oder eine OAuth-Client-ID erstellt werden (siehe Abb. 2). Damit wird sichergestellt, dass der Datenzugriff nur von autorisierten Nutzern ermöglicht wird.

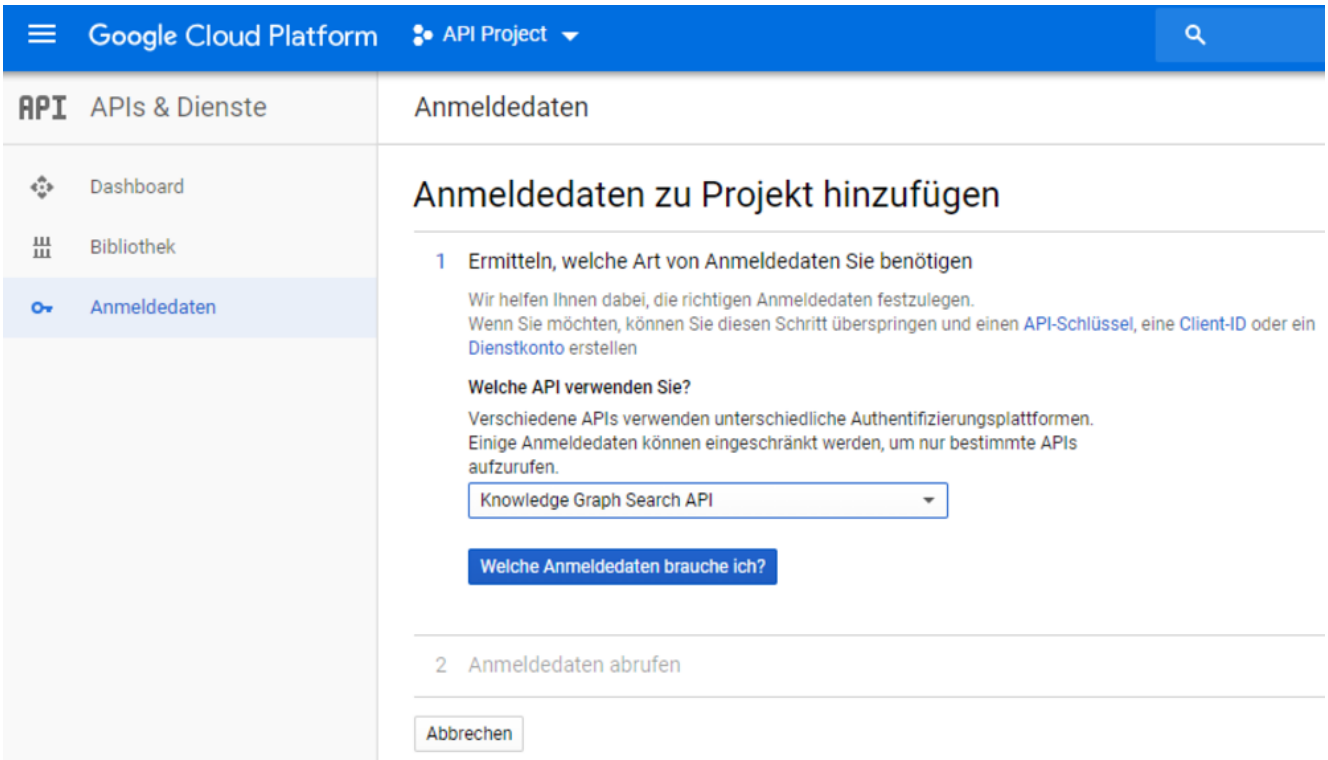


Abb. 2: Erstellung von gesicherten Anmeldemechanismen

Mittels der API werden die Daten direkt aus der Datenbank von

Google abgerufen. Es stehen damit nicht nur aktuelle, sondern auch ggf. erweiterte Daten zur Verfügung. Das heißt, Informationen über Albert Einstein, welche im Graphen am rechten Bildschirmrand angezeigt werden (siehe Abb. 3), stehen direkt zur weiteren Verarbeitung zur Verfügung. Die API ermöglicht die Extraktion weiterer Daten, sollten mit dem Suchbegriff mehrere Entitäten (Personen, Objekte) in Verbindung gebracht werden wie bspw. die Person Hans Albert Einstein.

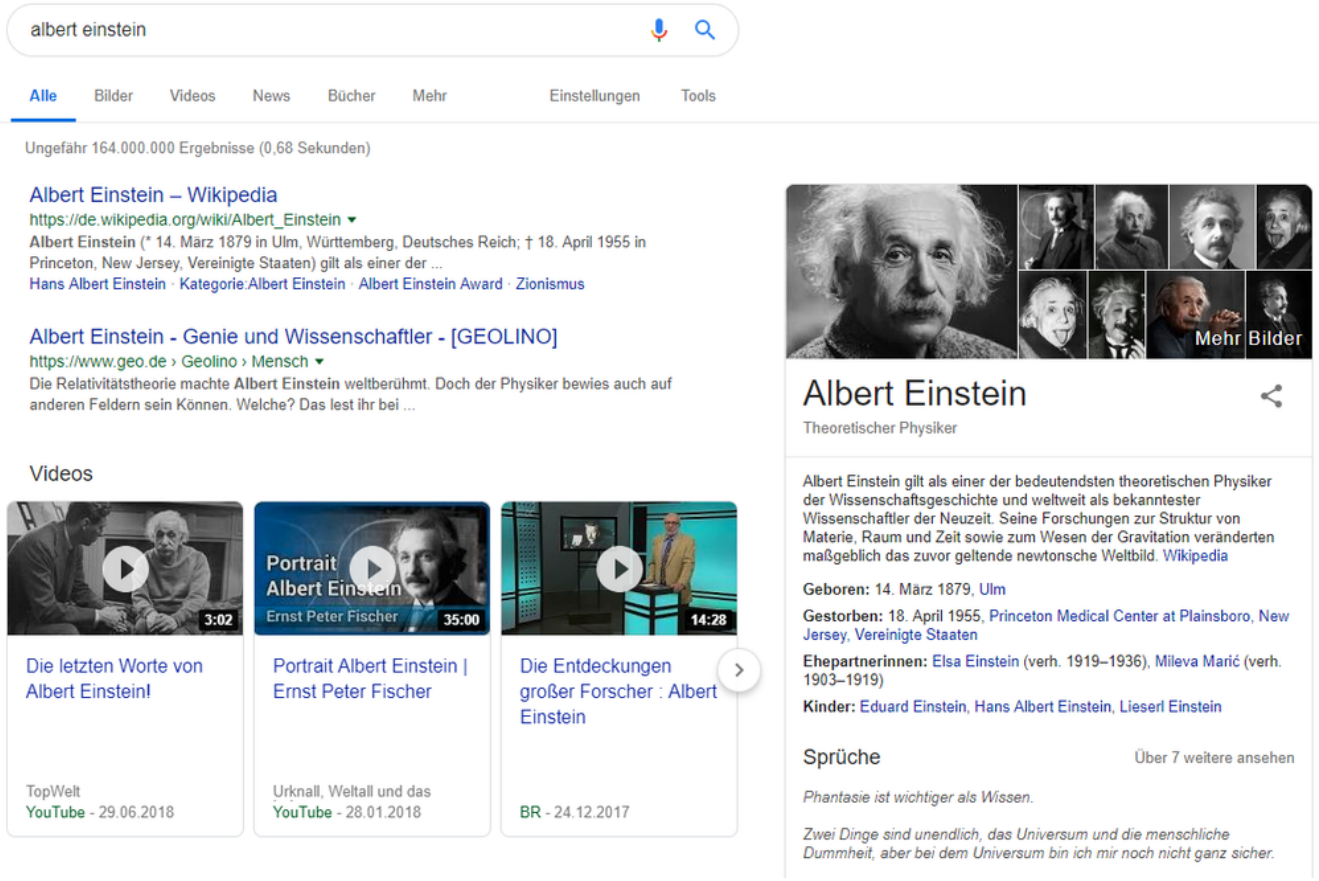


Abb. 3: Google Knowledge Graph von Albert Einstein

Das zur Datenextraktion notwendige R-Paket lautet *GoogleKnowledgeGraphR*, entwickelt von Daniel Schmeh (zur Dokumentation siehe das PDF unter: [einfach.st/r4kgraph)](einfach.st/r4kgraph). Mittels des individuellen API-Keys (Generierung siehe Abb. 2) werden mit dem Befehl

```
gkg_raw
```

die gewünschten Informationen in diversen Sprachen abgerufen. Das heißt, es stehen neben der Beschreibung unmittelbar die

URL des Bildes, die URL des Eintrages sowie Informationen zu den Nutzungsrechten (dennoch prüfen!) zur Verfügung (siehe Abb. 4). Ohne die jeweilige Suchabfrage bei Google starten zu müssen, können die Ergebnisse direkt in R abgerufen und weiterverarbeitet werden. Dies kann zu einer weiteren Vereinfachung und damit zur Zeitersparnis führen.

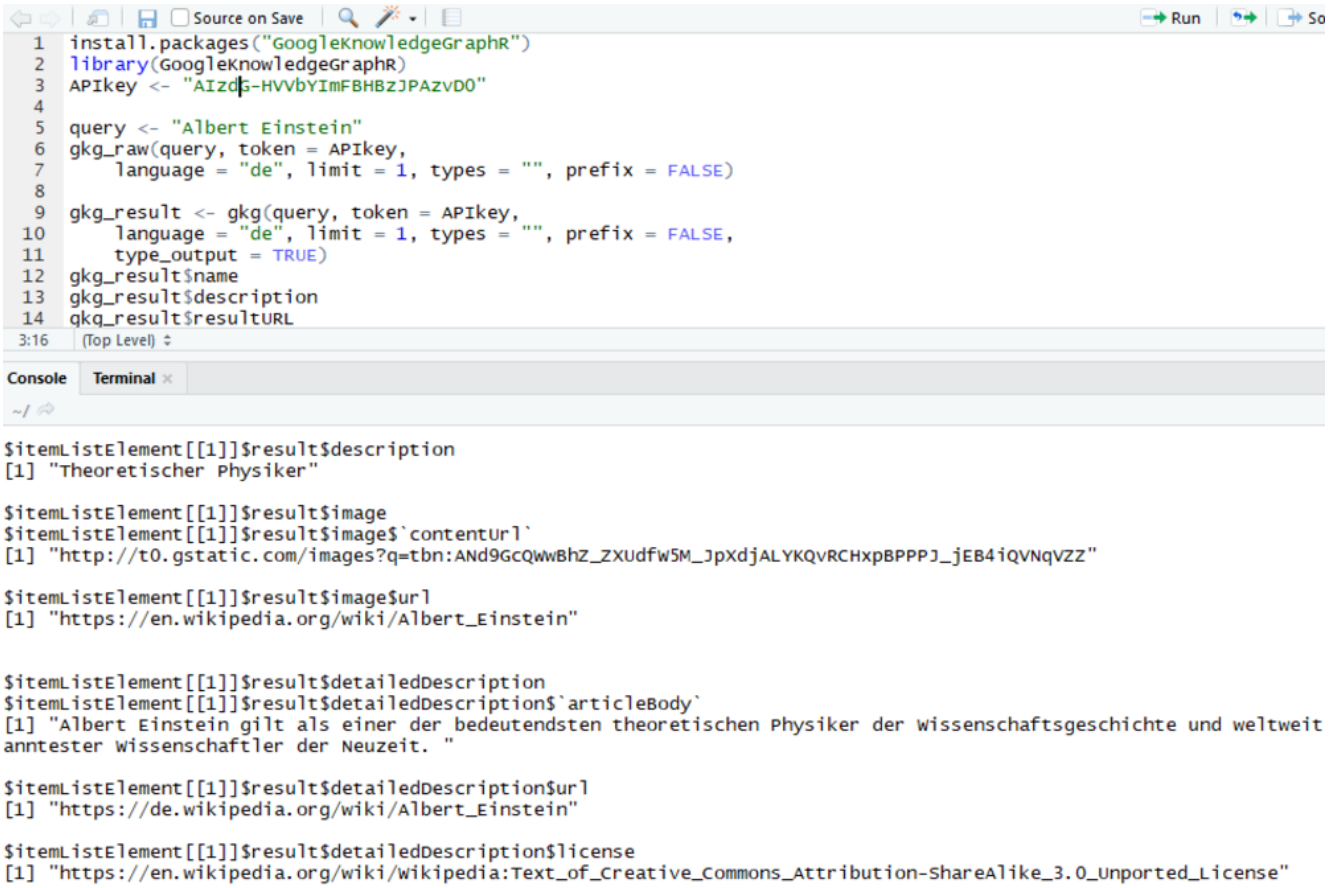


Abb. 4: Aufruf der Knowledge Graph API über R

Ähnlich diesem Prozess erfolgt die generelle Nutzung von Google API (mit R). Es bedarf eines API-Schlüssels bzw. einer Autorisierung sowie eines korrespondierenden Programms (hier R-Paket). In der Praxis können nahezu alle Google API, wie bspw. die Google Translator API oder die Google Maps/Places API, abgefragt werden. Vereinzelt sind die Services kostenpflichtig mit einem flexiblen Abrechnungsmodell wie bspw. auf Basis der Anzahl an API-Abfragen. Damit könnten Textübersetzungen bzw. Abfragen zu Distanzen, Öffnungszeiten, Adressen etc. von Restaurants und anderen Google-Places-Einträgen vollzogen werden. Die entsprechenden R Packages heißen *googleway* und *googleLanguageR*.

Kollaboration mit Google Sheets

Das Besondere an Google Sheets ist der weltweite Zugriff, die große Rechenleistung sowie die gleichzeitige Zusammenarbeit mit mehreren Personen. Weiterhin können Daten aus diversen Quellen zusammengeführt (vergleichbare Excel-Funktionen SVERWEIS bzw. INDEX/VERGLEICH) und letztlich ein Bericht sehr flexibel gestaltet, individualisiert und geteilt werden. Daher erscheint es sinnvoll, die Möglichkeiten von Google Sheets sowie ggf. der nachgelagerten Visualisierung mit Google Data Studio zu nutzen.

Daten in R können an Google Sheets bzw. von Google Sheets übermittelt werden. Hierzu sind die Pakete *googlesheets* (PDF zur Dokumentation siehe: einfach.st/r4sheets) bzw. *googleAuthR* notwendig bzw. hilfreich. Nach der Installation des Paketes über den Befehl

install.packages(“googlesheets“)(siehe Abb. 6)

erfolgt die Freigabe bzw. Authentifizierung des R-Pakets mit Google Drive über

gs_auth() (siehe Abb. 5).

tidyverse-googlesheets benötigt Zugriff auf Ihr Google-Konto

T [redacted]@gmail.com

Dadurch erhält tidyverse-googlesheets diese Berechtigungen:

- Google Drive-Dateien aufrufen, bearbeiten, erstellen und löschen
- Tabellen in Google Drive aufrufen, bearbeiten, erstellen und löschen

Überlegen Sie sich gut, ob Sie tidyverse-googlesheets vertrauen

Eventuell teilen Sie vertrauliche Informationen mit dieser Website oder App.In den Nutzungsbedingungen und der Datenschutzerklärung von tidyverse-googlesheets erfahren Sie alles zum Umgang mit Ihren Daten. In Ihrem Google-Konto können Sie die Zugriffsberechtigungen jederzeit einsehen oder entfernen.

Weitere Informationen zu den Risiken

Abbrechen

Zulassen

Abb. 5. Autorisierung des R-Pakets
Nach erfolgreicher Authentifizierung kann mit dem Befehl

gd_user()

der aktuelle Nutzer sowie der Berechtigungsstatus abgefragt werden. Der Befehl

gs_new

ermöglicht, ein neues Sheet anzulegen sowie den Namen bzw. Arbeitsblattnamen zu vergeben. Ab diesem Zeitpunkt können Daten eingegeben bzw. das Sheet genutzt werden. Der Befehl

gs_delete

löscht das Sheet entsprechend. Das jeweilige Sheet muss vor einer Aktion immer exakt adressiert werden, wozu diverse Möglichkeiten zur Verfügung stehen (siehe Dokumentation). Die exakte Art ist über die URL via

gs_url (siehe Abb. 6),

sofern Namen einzigartig sind, auch hilfsweise über den Titel (Befehl: *gs_title*). Soll das Sheet im Browser aufgerufen werden, muss der Befehl

gs_browse

mit dem Namen des Sheets (im Beispiel in der Variable gs gespeichert) und dem Blattnamen (relative Nummer bzw. absoluter Name) ausgeführt werden.

```
#install.packages("googlesheets") #Einmalig installieren
#install.packages("goolgeAuthR")  #Einmalig installieren
library(googlesheets)
library(googleAuthR)
gs_auth()
#gd_user()                        #Abfrage aktueller Nutzer

gs_new(title = "Analytics", ws_title = "Daten", row_extent = NULL, col_extent = NULL)

gs <- gs_title("Analytics")
gs <- gs_url("https://docs.google.com/spreadsheets/d/15cOt1RsXUxmB9RyrqXWb9yPiFoKxQk5ubA8YbUiKhAE")
gs_browse(gs, ws = 1)            #Google Sheets im Browser öffnen

gs_delete(gs, verbose = TRUE)
```

Abb. 6: Erstellung eines Google Sheets innerhalb von R

Google Sheets mit Daten füllen

Prinzipiell können auch Daten aus Google Analytics in R abgerufen und analysiert werden (siehe den ersten Teil dieses Beitrags in der letzten Ausgabe #53). Dem Google Sheet „Analytics“ kann über den Befehl

gs_ws_new

ein neues Arbeitsblatt mit dem Namen Oktober hinzugefügt werden (siehe Abb. 7) und anschließend mit den Google-Analytics-Daten befüllt werden (Abruf der Daten über den Befehl *google_analytics* aus dem R-Paket googleAnalyticsR). Die neuen Informationen sollen ab der Position (anchor) A1 zeilenweise (byrow = FALSE) eingefügt werden. Sollten dann im späteren Verlauf weitere Zeilen hinzugefügt werden (und dabei die alten Daten nicht überschrieben werden), kann ein Arbeitsblatt ausgelesen (Befehl *gs_read*), die aktuelle Anzahl an benutzten Zeilen festgestellt (nrow) und anschließend über die R-Funktion *paste0* ein neuer Start ermittelt (im Beispiel Zelle A38) und an die Variable gs_*position* übermittelt werden. Ein weiterer Aufruf der Funktion *gs_edit_cells* fügt die Daten demnach nicht wieder an der Position A1 ein, sondern nutzt als initiale Startposition den Inhalt der Variable gs_position.

```
18  library(googleAnalyticsR)
19  ga_auth()
20  ga_id <- "xxx"
21  ga_data <- google_analytics(ga_id,
22                              date_range = c("2018-10-01", "2018-10-31"),
23                              metrics = c("sessions", "bounces"),
24                              dimensions = c("date","landingpagepath"))
25
26  gs_ws_new(gs, ws_title = "Oktober", row_extent = 1000, col_extent = 26, verbose = TRUE)
27  gs <- gs_title("Analytics")
28  gs_edit_cells(gs, ws = "Oktober", input = ga_data, anchor = "A1", byrow = FALSE,
29                col_names = NULL, trim = FALSE, verbose = TRUE)
30
31  gs_lesen <- gs_read(gs, ws = 2, range = NULL, verbose = TRUE)
32  gs_position <- paste0("A",nrow(gs_lesen)+2)
33  gs_position
34
35  gs_edit_cells(gs, ws = 2, input = ga_data, anchor = gs_position, byrow = FALSE,
36                col_names = NULL, trim = FALSE, verbose = TRUE)
```

Abb. 7: Daten an Google Sheets aus R übertragen

Highlight: spezifische Google-Sheets-Funktionen

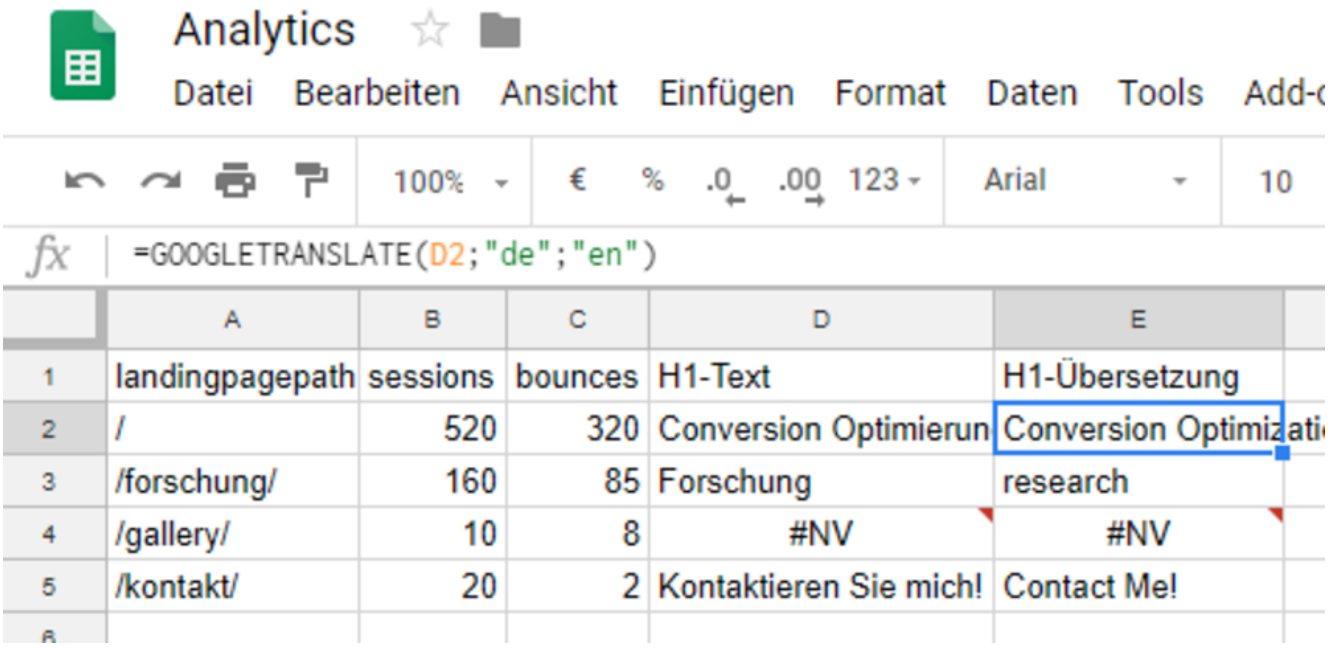
Mittels Austausches von Information an Google Sheets (lesen und schreiben) kann ein flexibler Datenaustausch mit R erfolgen. Ein weiterer Vorteil von Sheets ist die Mannigfaltigkeit spezieller Funktionen. Über die Funktion

```
=IMPORTXML(VERKETTEN("http://www.meineSeite.de";A2);"//h1")
```

kann eine XPath-Abfrage generiert werden, welche Elemente aus beliebigen Webseiten direkt ausliest (hier der H1-Text; siehe Abb. 8). Mittels einer weiteren Funktion

```
=GOOGLETRANSLATE(D2;"de";"en")
```

wird der Inhalt einer Zelle in die englische Sprache übersetzt. Die Funktion Verketten ermöglicht in diesem Zusammenhang, URL-Fragmente aus Google Analytics zu einer kompletten URL zusammenzuführen. Die Daten in Google Sheets können im weiteren Verlauf (manuell) beliebig verändert und anschließend wieder in R eingelesen/ weiterverarbeitet oder mit einer Visualisierung in Google Data Studio verbunden werden.



	A	B	C	D	E
1	landingpagepath	sessions	bounces	H1-Text	H1-Übersetzung
2	/	520	320	Conversion Optimierun	Conversion Optimizati
3	/forschung/	160	85	Forschung	research
4	/gallery/	10	8	#NV	#NV
5	/kontakt/	20	2	Kontaktieren Sie mich!	Contact Me!

Abb. 8: Nutzung von Google-Sheets-spezifischen Funktionen

Google Data Studio hat diverse Verbindungen zu Datenquellen, was für die Analyse und Aufbereitung eine hohe Flexibilität bietet. Weiterhin können beliebige Google-Tabellen als Datenquelle initiiert und anschließend in die Berichte integriert werden – bspw. jeweils als eigene Seiten. Die Anbindung geschieht über zwei Schritte. Erstens muss über den Google Tabellen-Connector das jeweilige Tabellenblatt gewählt und somit als Datenquelle definiert werden (siehe Abb. 9). Zweitens müssten anschließend die damit geschaffenen Datenverbindungen mit einem spezifischen Bericht verbunden und die jeweiligen Datenfelder ausgewählt werden. Zur Visualisierung stehen diverse Möglichkeiten zur Verfügung (siehe Abb. 10).

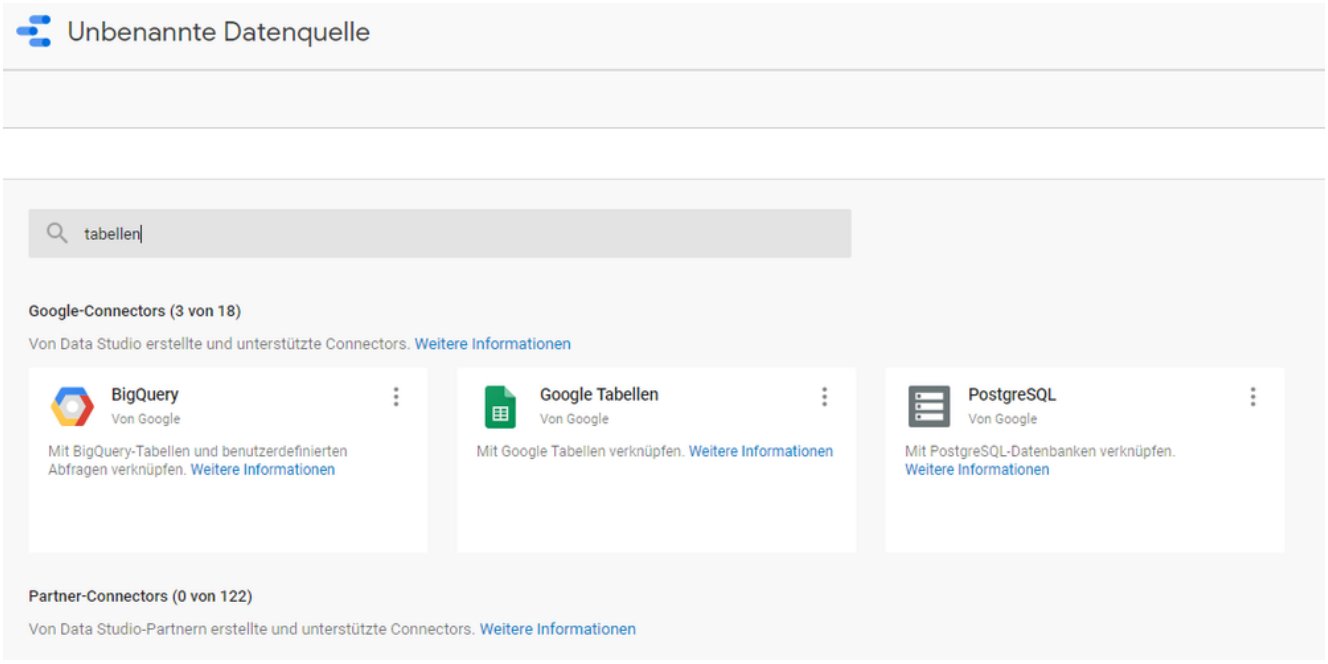


Abb. 9: Auswahl von Google-Tabellen als Data-Studio-Connector

Innerhalb des Berichtes besteht die Möglichkeit, mit mehreren Seiten zu arbeiten und damit bspw. die Sheets-Integration als Detailbericht einer großen Analyse zu nutzen. Weiterhin können einzelne Informationen/Spalten aus Sheets ausgewählt und in diversen Visualisierungsmöglichkeiten auf dem Bericht verortet werden.

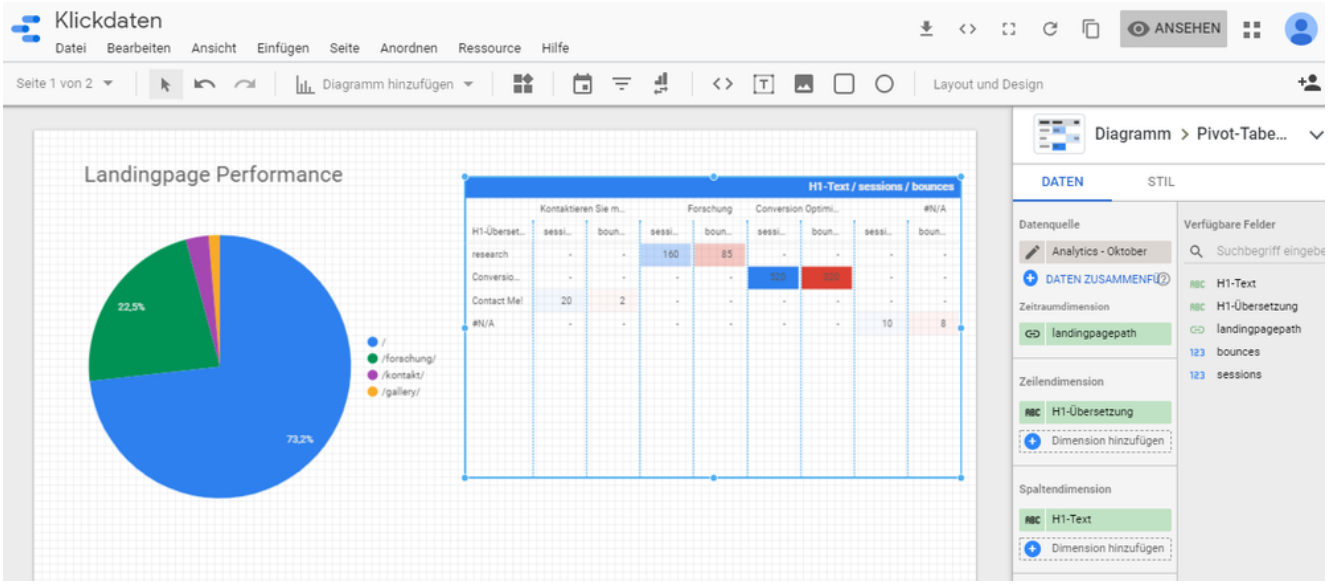


Abb. 10: Berichtsaufbau mit Sheets-Integration

Fazit

Zusammenfassend lässt sich sagen, dass R, Google Sheets, Google Data Studio etc. für sich hervorragende Möglichkeiten bieten. Die wahre Stärke zeigt sich, wie fast immer, in der Kombination verschiedener Faktoren/Quellen. Daten aus diversen Quellen (schnell) zusammenzufassen, aus unterschiedlichen Blickwinkeln zu betrachten und unkompliziert einem definierten Empfängerkreis zur Verfügung zu stellen, sind nur punktuelle Vorteile.

In diesem Sinne: Verbinden Sie Ihre Informationen, kombinieren Sie Daten – ggf. trügt ja der erste Anschein.