Datenanalysen mit R und der Google Analytics API

Datenanalysen mit R und der Google Analytics API – websiteboosting.com

Das Programm bzw. die Programmiersprache R erfährt in den letzten Jahren eine steigende Nachfrage. Neben dem erfreulichen Aspekt, dass es Open Source und damit komplett kostenfrei ist, kann R durch Pakete flexibel erweitert werden. Neben klassischen Datamining-Paketen und fortgeschrittenen…
Das Programm bzw. die Programmiersprache R erfährt in den letzten Jahren eine steigende Nachfrage. Neben dem erfreulichen Aspekt, dass es Open Source und damit komplett kostenfrei ist, kann R durch Pakete flexibel erweitert werden. Neben klassischen Datamining-Paketen und fortgeschrittenen Analysemethoden wie neuronalen Netzen existieren diverse Möglichkeiten, den Google-Kosmos mit R anzusprechen. Die Anbindung der API von Google Analytics, Search Console bzw. der komplette Zugriff auf Sheets/Docs sind exemplarische Beispiele hierfür. Daten können in beliebiger Kombination abgefragt, aufbereitet, analysiert und visualisiert/exportiert werden.

Analysen mit R – „flexibel wie ein Schweizer Taschenmesser“

R wird seit vielen Jahren durch eine große Community kontinuierlich weiterentwickelt. Vergleichbar mit Add-ins bei Excel können in R weitere Funktionen und Schnittstellen zum Standardsystem hinzugefügt werden. So schön diese Flexibilität und Leistungsstärke klingt (und tatsächlich ist), so unspektakulär ist die Nutzerfreundlichkeit des Systems: Eingabe von Programmcode anstelle klickbarer Icons und Mausklicks à la Excel. Was jetzt als Nachteil klingt, ist gleichzeitig ein Riesenvorteil: Sofern der Programmcode erstellt ist, kann dieser jederzeit „abgespielt“ werden, die entsprechenden Analysen/Visualisierungen dauern wenige Sekunden/Minuten und können unternehmensweit geteilt werden.

Jeder Schritt ist transparent und durch den Einsatz von Variablen voll flexibel – einmalig die ID des Google Analytics-Kontos ausgeschaut und alle Analysen können automatisch von Neuem starten. R ist quasi ein „großes Excel-Makro“, das die Daten auf Befehl Schritt für Schritt verarbeitet.

Die Software R kann unter www.r-project.org/ kostenfrei heruntergeladen werden und ist nach der Installation sofort einsatzbereit. Es empfiehlt sich, zusätzlich RStudio zu installieren (www.rstudio.com/products/rstudio/download/), welches einen deutlichen Mehrwert in der Bedienung des Systems bietet (Abb. 1).

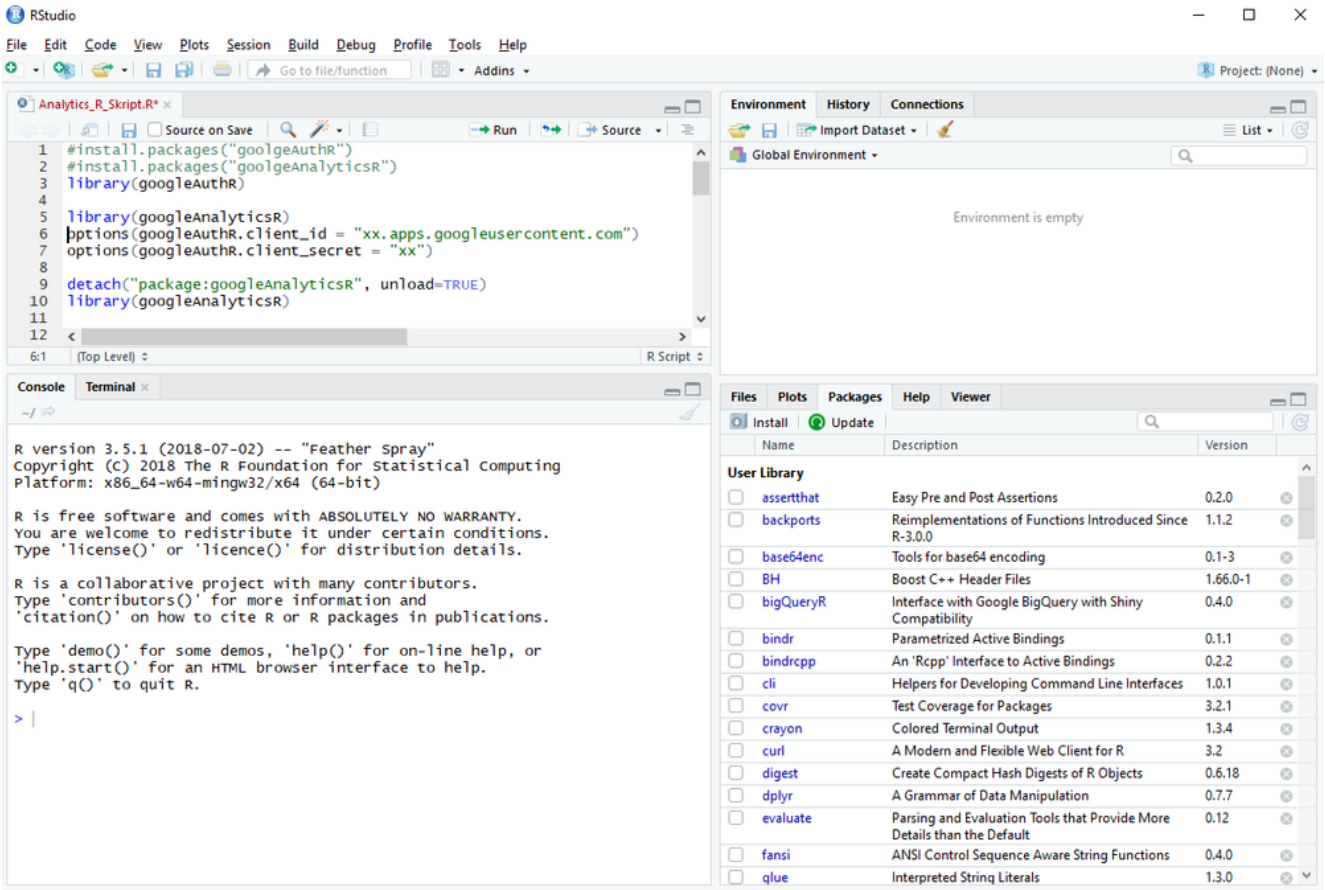

Abb. 1: RStudio „bettet“ die Software R benutzerfreundlich ein und macht die Bedienung komfortabler

Prinzipiell funktioniert R mittels Eingabe und sequenzieller Verarbeitung von Programmcode. Eingaben können sowohl in Form klassischer Rechenoperationen erfolgen (bspw. 3+3) als auch als Definition von Formeln, Funktionen, Variablen, Vektoren und Dataframes. Der Vorteil der Nutzung von Variablen ist,

dass diese nach Definition zur gesamten Laufzeit zur Verfügung stehen und im weiteren Verlauf darauf zurückgegriffen werden kann. Füllt man bspw. die Variable „ga_id“ (Variablennamen sind frei wählbar) mit der Datenansichts-ID von Google Analytics und verweist im weiteren Verlauf der Berechnungen auf die Variable anstelle einer fix definierten Nummer, kann die identische Analyse durch Neudefinition der Variable flexibilisiert werden. Variablen werden in R mittels der Syntax „Variablenname <- Variableninhalt“ definiert.

googleAnalyticsR – das flexible R-Paket

Damit R auf die API von Google Analytics zugreifen kann, muss ein Paket installiert werden (genereller Befehl: install.packages(„Paketname“). Die Installation des Pakets erfolgt nach Befehlseingabe binnen Sekunden, da die Pakete automatisch aus dem Internet geladen werden. Die Erweiterung „dependencies = TRUE“ bewirkt, dass passende Pakete zusätzlich installiert werden, insb. googleAuthR zur OAuth-Authentifizierung des Google-Nutzerkontos mit R (siehe Abb. 2).

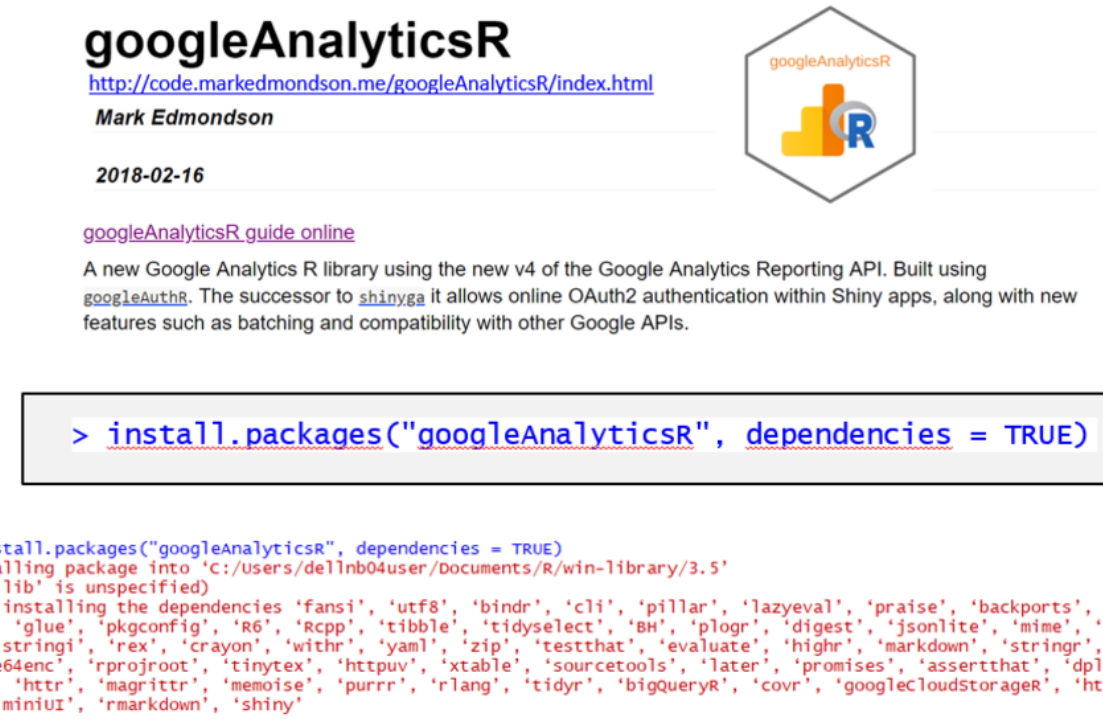

Abb. 2: Einfache Installation von googleAnalyticsR nebst weiteren Paketen direkt in R

Nachdem die gewünschten Pakete installiert wurden, stehen diese prinzipiell zur Verfügung. Damit sie tatsächlich auch genutzt werden können, müssen sie mit dem Befehl library(„Paketname“) geladen werden. Das heißt, nach Absetzen des Befehls „library(“googleAnalyticsR“)“ besteht eine Verbindung von R zur Google Analytics API. Der Befehl „ga_auth()“ bzw. „gar_auth()“ aus dem Paket googleAuthR ermöglicht die Authentifizierung von R mit dem jeweiligen Google-Konto über das OAuth-Verfahren. Ein erstmaliger Aufruf leitet zum Google-Konto-Log-in nebst Angabe der übergebenen Berechtigungen (siehe Abb. 3).

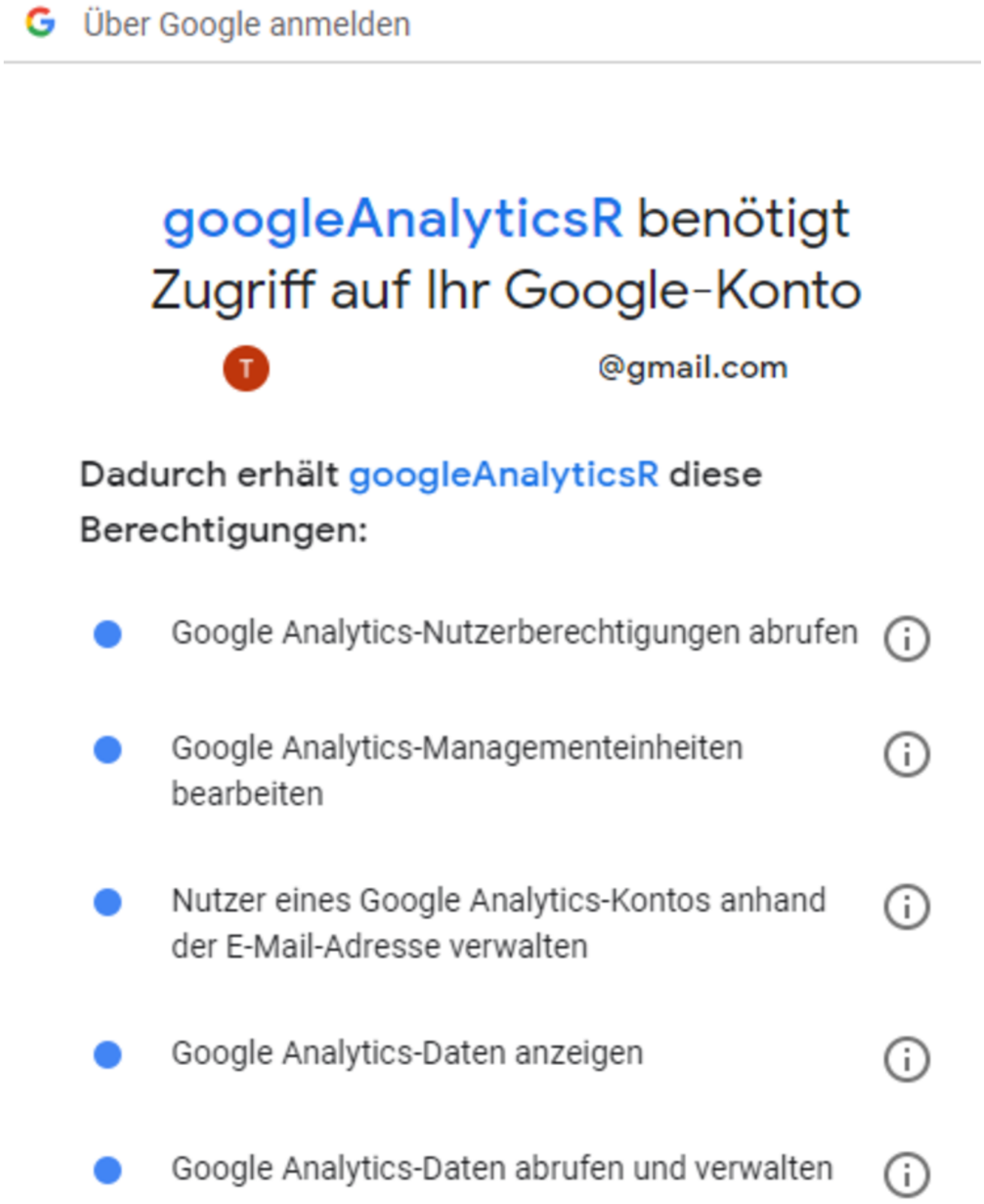

Abb. 3: Zugriff des R-Pakets auf Google-Konto

Sofern das Google-Konto bei Google Analytics mindestens das Recht „Bearbeiten“ auf eine Property hat, kann im weiteren Verlauf direkt die Google API genutzt werden. Wie angesprochen ist es sinnvoll, mit Variablen zu arbeiten und die gewünschte

Auswahl an Metriken und Dimensionen in Vektoren zu speichern. Der Befehl „ga_account_list()“ erzeugt eine Übersicht aller Analytics Propertys nebst Datenansichten (Views) im jeweiligen Google-Konto. Als Beispiel kann eine Variable „account_list“ mit genau dieser Übersicht befüllt (Befehl account_list <- ga_account_list()) und anschließend aufgerufen werden (siehe Abb. 4). Auf Basis dieser Liste kann bspw. eine Variable „ga_id“ dynamisch mit der zweiten Datenansicht befüllt werden. Hierzu wird das Feld viewId mit dem Dollarzeichen ($) aus account_list angesprochen und die zweite Position adressiert. Alternativ könnte die Variable ga_id selbstverständlich direkt mit der jeweiligen View-Nummer befüllt werden.

```
> ga_auth()
Waiting for authentication in browser...
Press Esc/Ctrl + C to abort
Authentication complete.
Token cache file: .httr-oauth
> account_list <- ga_account_list()
> account_list
  accountId   accountName internalWebPropertyId    level                  websiteUrl webPropertyId
1 [redacted]
2 [redacted]
   webPropertyName type   viewId                    viewName
1 [redacted]
2 [redacted]
> ga_id <- account_list$viewId[2]
> ga_id
[1] "65785    "
> google_analytics(ga_id,
+                  date_range = c("2018-10-01", "2018-10-30"),
+                  metrics = c("sessions", "bounces", "bounceRate"),
+                  dimensions = c("date","source", "medium"))
2018-11-15 06:31:51> Downloaded [33] rows from a total of [33].
         date              source   medium sessions bounces bounceRate
1  2018-10-01              google  organic        1       0          0
2  2018-10-02              google  organic        1       1        100
3  2018-10-03                bing  organic        1       0          0
4  2018-10-04              google  organic        1       0          0
```

Abb. 4: R-Befehle zum Abruf von R

Mit dem Befehl „google_analytics“ wird die API angesprochen und die entsprechenden Werte werden in R ausgegeben. Sofern bspw. der Datumbereich sowie die gewünschten Metriken und Dimensionen nebst der View-ID im Befehl google_analytics übergeben werden, werden die Daten unmittelbar abgerufen. Aufgrund des Abrufs der Daten über die API kann ggf. das Sampling von Daten umgangen werden. Hierzu bedarf es beim Aufruf des Befehls google_analytics der Ergänzung um den Parameter anti_sample = TRUE. Die Funktion teilt dabei den API-Aufruf in mehrere Teilaufrufe auf, damit möglichst ungefilterte Daten über die API abgerufen werden. Weiterhin

kann mit dem Parameter max = „Zahl“ die maximale Anzahl von abzurufenden Daten limitiert werden (siehe Abb. 5).

Tipp: Es besteht für Google Analytics ein sog. „Dimensionen- und Metriken-Explorer“, der anzeigt, welche Dimensionen und Metriken generell kombiniert werden dürfen und wie die entsprechenden Felder in der API heißen (bspw. ga:bounceRate für die Absprungrate in %): einfach.st/gareferenzen.

Das Paket googleAnalyticsR nutzt ein allgemeines Kontingent an API-Aufrufen von Google Analytics. Bei intensiver Nutzung und zur Sicherstellung der Verfügbarkeit sollte ein eigenes API-Kontingent über Google Developers beantragt werden (https://console.developers.google.com). Nach Registrierung bei Developers stehen diverse Google APIs zur Verfügung. Als Anpassung in R müssen vor der Authentifizierung die Client-ID sowie der geheime Clientschlüssel als Option angegeben werden (options(googleAuthR.client_id = „xx.apps.googleusercontent.com“) sowie options(googleAuthR.client_secret = „xx“). Beide Informationen sind nach der Registrierung und Aktivierung der Analytics API abrufbar. Bei Google Analytics bietet die API 50.000 Abfragen pro Tag/User bzw. alle 100 Sekunden 100 Abfragen.

Flexibler Export und Datenvisualisierungen mit ggplot2

Sofern das Ergebnis des Abrufs mittels „google_analytics“ in eine Variable/Dataframe gespeichert wird, können neben der Durchführung statistischer Auswertungen (bspw. summary(„Variable“) die Daten in ein spezifisches Dateiformat gespeichert und dann mit anderen Programmen weiterverarbeitet werden. Eine CSV-Datei wird über den Befehl „write.csv“ erzeugt (siehe Abb. 5).

Tipp: Perfekt harmoniert R auch mit Google Sheets, was damit mittelbar eine nachgelagerte Datenvisualisierung mit Google

Data Studio ermöglicht. Das Szenario wäre folgendes: Daten werden aus Google Analytics und anderen Quellen in R analysiert und verarbeitet, automatisch über das R-Paket „googlesheets“ in ein Google Spreadsheet übertragen und anschließend über Data Studio visualisiert und ggf. mit weiteren Datenquellen ergänzt.

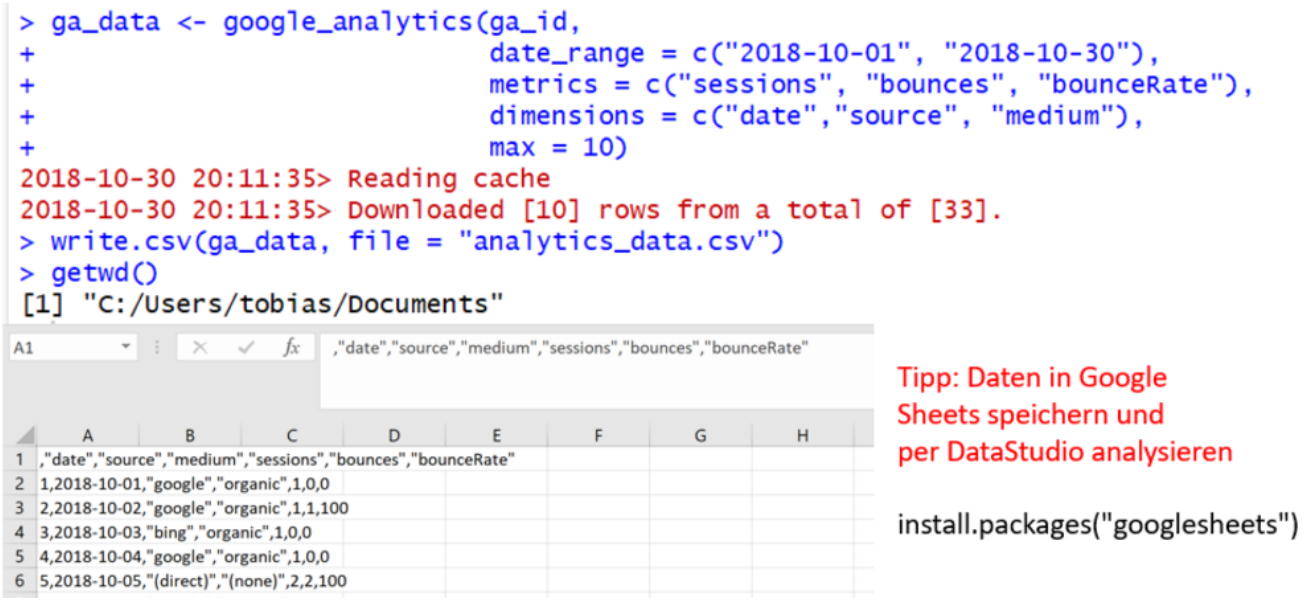

Abb. 5: Daten aus der API in R analysieren und als .csv speichern

R hat herausragende Möglichkeiten der Datenvisualisierung. Es existiert eine Vielzahl an Paketen, welche die Daten mit weiteren Analysen grafisch anreichern. Pakete wie „RColorBrewer“ erzeugen u. a. Heatmaps als Datenvisualisierung. Das Paket „ggplot2“ (install.packages(“ggplot2“) sowie Aktivierung über library(“ggplot2“)) bietet Schaubilder, die in Excel nicht existent sind. Mittels des Befehls geom_boxplot wird bspw. ein Boxplot erstellt, geom_line / geom_smooth erzeugt ein Liniendiagramm nebst nicht linearer Trendlinie (siehe Abb. 6).

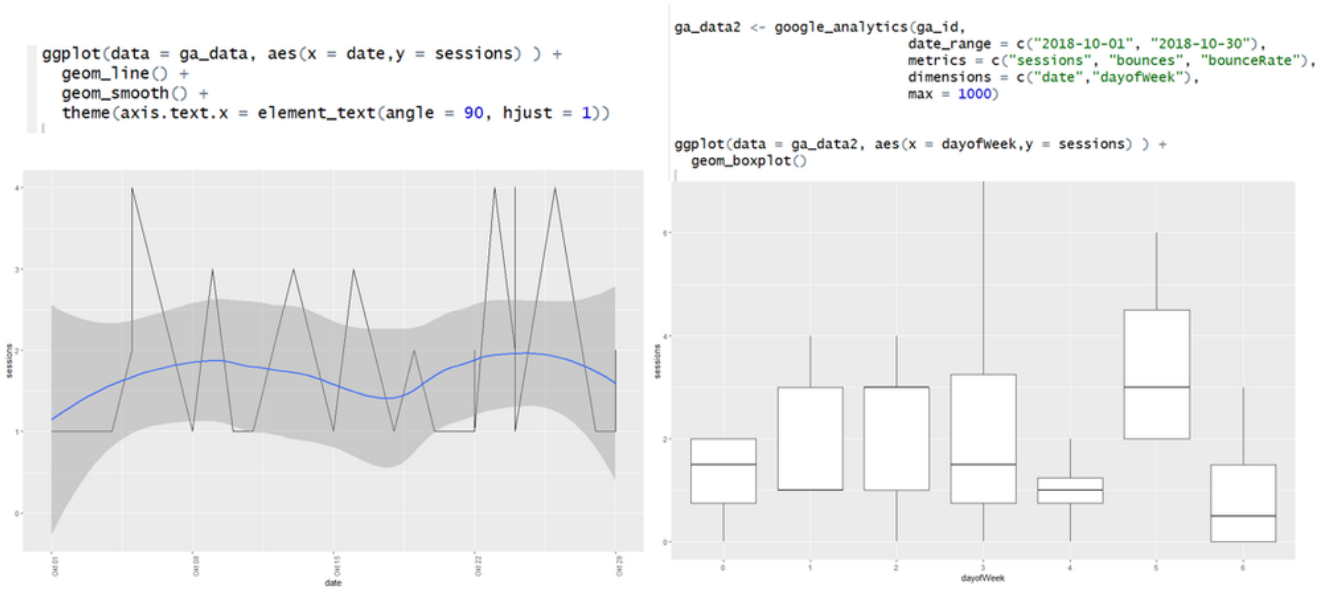

Abb. 6: Datenvisualisierungen mit dem Paket ggplot2

Tipp: Die Visualisierung bzw. Berichterstellung ist in R sehr umfassend durch sogenanntes R Markdown nutzbar. Es wird einmalig der Programmcode erstellt bzw. ein Bericht konzipiert, welcher anschließend als HTML-Seite, PDF oder Word-Dokument ausgegeben wird. Damit sind aufwendiges „Exceln“ und formatieren obsolet, die Corporate Identity ist in hohem Maße in den Berichten standardisierbar.

Richtig interessant werden Analysen und Visualisierungen, wenn diese auf Rohdaten beruhen. Mittels der Analytics API können viele Informationen verarbeitet werden, die es später zu clustern gilt. Eine sinnvolle Ergänzung ist deshalb, die Google-Analytics-Daten durch benutzerdefinierte Dimensionen zu erweitern. Wird jedem Datensatz ein exakter Zeitstempel mitgegeben, jede Session mit einer Session-ID in den Daten ergänzt sowie die Cookie-ID als Klammer für den Besucher verstanden, so könnte eine Customer-Journey-Analyse mit Rohdaten erfolgen. Simo Ahava hat in seinem Blog eine detaillierte Anleitung für diese benutzerdefinierten Dimensionen erstellt (http://einfach.st/simohava). Es ist sehr ratsam, vor der Integration den jeweiligen Datenschutzverantwortlichen zu konsultieren!

In diesem Sinne: Geben Sie R eine Chance! Im ersten Schritt hat es eine „Old-School“-Anmutung, da Befehle über eine Console eingegeben werden müssen. Mit Neugier und Training werden die Abläufe Tag für Tag vereinfacht und Webanalyse bzw. Datenvisualisierung kann wirklich auf „Knopfdruck“ erfolgen – so besteht viel mehr Freiraum und Zeit, über die Ergebnisse nachzudenken bzw. diese zu interpretieren, anstatt Zeit für die Datenaufbereitung ineffizient zu vergeuden. Und zuallerletzt: Das System wird permanent (kostenlos) weiterentwickelt– für jede Anforderung existiert bestimmt ein Paket.

Noch ein Literaturhinweis: Das kostenlose englischsprachige E-Book „Using Google Analytics with R“ von Michal Brys ist unter http://michalbrys.com/book/ in diversen Formaten abrufbar und bietet einen hervorragenden Einstieg in die Nutzung der Analytics API!