Data Science, Computational Social Science, Big Data:

Programmieren für Sozialwissenschaftler_innen

Laura Kiemes 12250912 & Felix Grams 12293142

2022-07-12

Inhalt

R version 4.2.0 (2022-04-22)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Monterey 12.4

Pakete und Ordner

Zunächst wird das Pacman Paket installiert, falls dies noch nicht installiert wurde. Als nächstes wird dieses zusätzlich geladen. Über Pacman werden anschließend alle verwendeten Pakete installiert, falls dies noch nicht geschehen ist und ansonsten geladen.
Die Ordner Plots und Tables werden erstellt.

```
here::i_am("Googletrends.Rmd")
```

```
## here() starts at /Users/laurakiemes/Library/Mobile Documents/com~apple~CloudDocs/LMU/Soziologie/P10 S
```

```
# Falls pacman nicht installiert ist wird es installiert und geladen
if (!require("pacman")) install.packages("pacman"); library(pacman)
```

```
## Lade nötiges Paket: pacman
```

```
# Falls pakete nicht installiert sind werden sie installiert und Pakete werden geladen
pacman::p_load(tidyverse, gtrendsR, ggsci, maps, rtweet, patchwork, lubridate)

dir.create("plots")
```

```
## Warning in dir.create("plots"): 'plots' existiert bereits
```

```
dir.create("tables")
```

```
## Warning in dir.create("tables"): 'tables' existiert bereits
```

```
date_analysis <- ymd("2022-07-01")
```

Gtrends

Zunächst werden Variationen für die Gtrends-Abfrage angelegt.

```
data("countries") # Datensatz zu Laenderbezeichnungen
countries$country_code[which(countries$name == "GERMANY")] # Abkuerzung fuer Deutschland
```

```
## [1] "DE"
```

```
keywords_5 <- data.frame(
  languages = c("python", "r", "java", "php", "javascript"), # Schluesselworte Programmieren

  text = c("sql", "excel", "word", "powerpoint", "tableau")) # Schlusselworte Programme
geos <- c("DE", "") # interessierende Laender
times <- c("all", "2017-07-01 2022-07-01", "today 12-m") # interessierende Zeitfenster
list_data <- list() # leere Liste zum Fuellen mit Datensaetzen
list_plots <- list() # leeere Liste zum Fuellen mit Plots
```

Dann iterieren wir über die verschiedenen Iterationen und speichern die erzeugten Objekte als csv Dateien und in einem Listenobjekt.

```
i <- 0 # Index zum Mitlaufen

for (k in keywords_5) {
  for (t in times) {
    for (g in geos) {

      i <- i + 1 # Indexzaehler
      print(paste(i, t, g, k)) # Iteration anzeigen

      list_data[[i]] <- gtrends(
        keyword = k,
        geo = g,
        time = t,
        gprop = "web",
        category = 0,
        hl = "en-US",
        compared_breakdown = TRUE,
        low_search_volume = FALSE,
        cookie_url = "http://trends.google.com/Cookies/NID",
        tz = 0,
        onlyInterest = FALSE
      )
```

```
      write_csv(as.data.frame( # Tabellen erstellen
        list_data[[i]][["interest_over_time"]]),
        file = here::here("tables", paste0("table_", i, ".csv")))

    }
  }
}
```

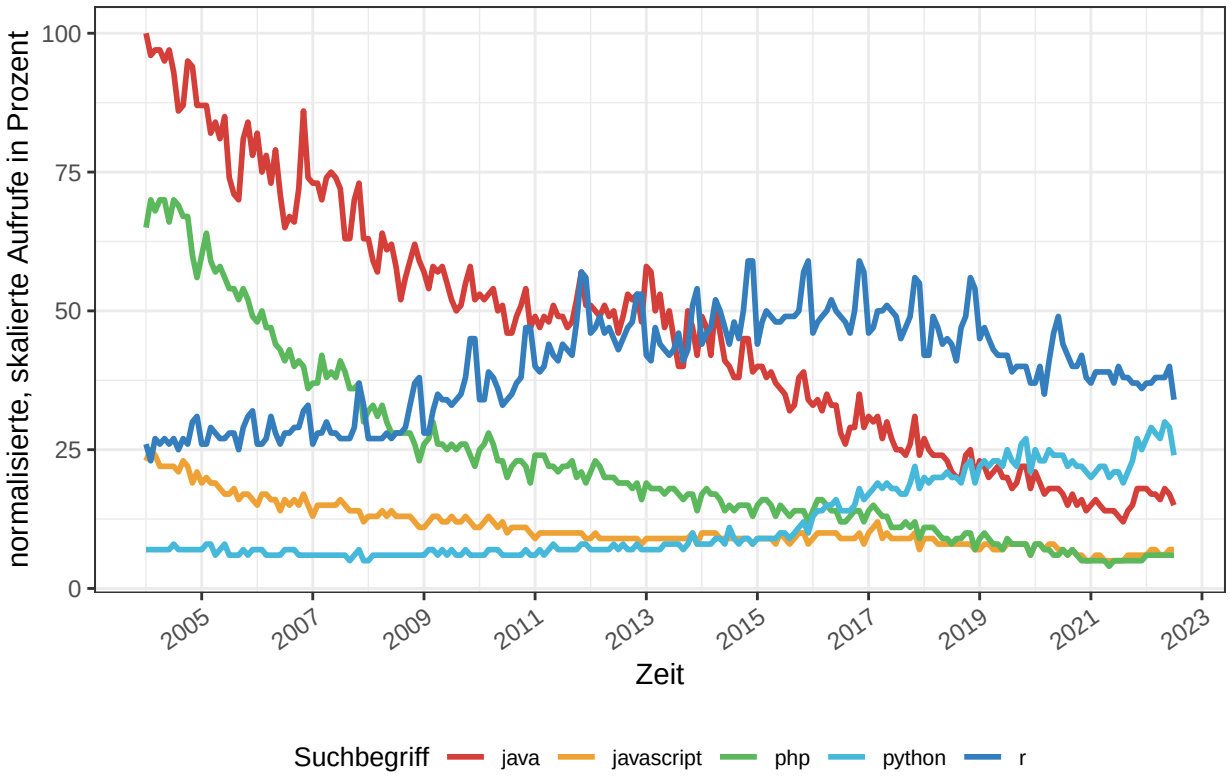
```
## [1] "1 all DE python"     "1 all DE r"          "1 all DE java"
## [4] "1 all DE php"        "1 all DE javascript"
## [1] "2 all  python"     "2 all  r"          "2 all  java"
## [4] "2 all  php"        "2 all  javascript"
## [1] "3 2017-07-01 2022-07-01 DE python"
## [2] "3 2017-07-01 2022-07-01 DE r"
## [3] "3 2017-07-01 2022-07-01 DE java"
## [4] "3 2017-07-01 2022-07-01 DE php"
## [5] "3 2017-07-01 2022-07-01 DE javascript"
## [1] "4 2017-07-01 2022-07-01  python"     "4 2017-07-01 2022-07-01  r"
## [3] "4 2017-07-01 2022-07-01  java"       "4 2017-07-01 2022-07-01  php"
## [5] "4 2017-07-01 2022-07-01  javascript"
## [1] "5 today 12-m DE python"     "5 today 12-m DE r"
## [3] "5 today 12-m DE java"       "5 today 12-m DE php"
## [5] "5 today 12-m DE javascript"
## [1] "6 today 12-m  python"     "6 today 12-m  r"
## [3] "6 today 12-m  java"       "6 today 12-m  php"
## [5] "6 today 12-m  javascript"
## [1] "7 all DE sql"        "7 all DE excel"      "7 all DE word"
## [4] "7 all DE powerpoint" "7 all DE tableau"
## [1] "8 all  sql"        "8 all  excel"      "8 all  word"
## [4] "8 all  powerpoint" "8 all  tableau"
## [1] "9 2017-07-01 2022-07-01 DE sql"
## [2] "9 2017-07-01 2022-07-01 DE excel"
## [3] "9 2017-07-01 2022-07-01 DE word"
## [4] "9 2017-07-01 2022-07-01 DE powerpoint"
## [5] "9 2017-07-01 2022-07-01 DE tableau"
## [1] "10 2017-07-01 2022-07-01  sql"
## [2] "10 2017-07-01 2022-07-01  excel"
## [3] "10 2017-07-01 2022-07-01  word"
## [4] "10 2017-07-01 2022-07-01  powerpoint"
## [5] "10 2017-07-01 2022-07-01  tableau"
## [1] "11 today 12-m DE sql"        "11 today 12-m DE excel"
## [3] "11 today 12-m DE word"       "11 today 12-m DE powerpoint"
## [5] "11 today 12-m DE tableau"
## [1] "12 today 12-m  sql"        "12 today 12-m  excel"
## [3] "12 today 12-m  word"       "12 today 12-m  powerpoint"
## [5] "12 today 12-m  tableau"
```

```
names(list_data) <- c("prog_all_de", "prog_all_int",
                      "prog_5y_de", "prog_5y_int",
                      "prog_12_de", "prog_12m_int",
                      "txt_all_de", "txt_all_int",
                      "txt_5y_de", "txt_5y_int",
                      "txt_12_de", "txt_12m_int")
```

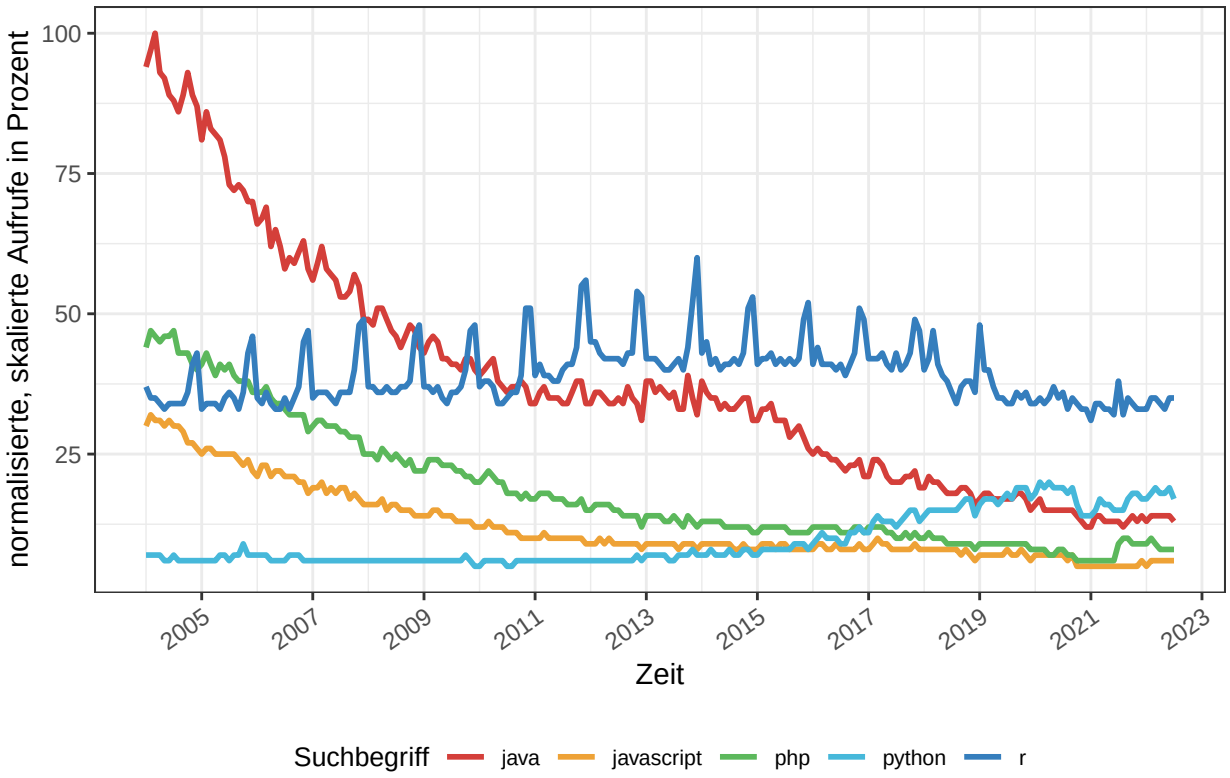
Plots

Anschließend iterieren wir über die erzeugten Listen mit den Datensätzen und erstellen jeweils einen Plot pro Datensatz.

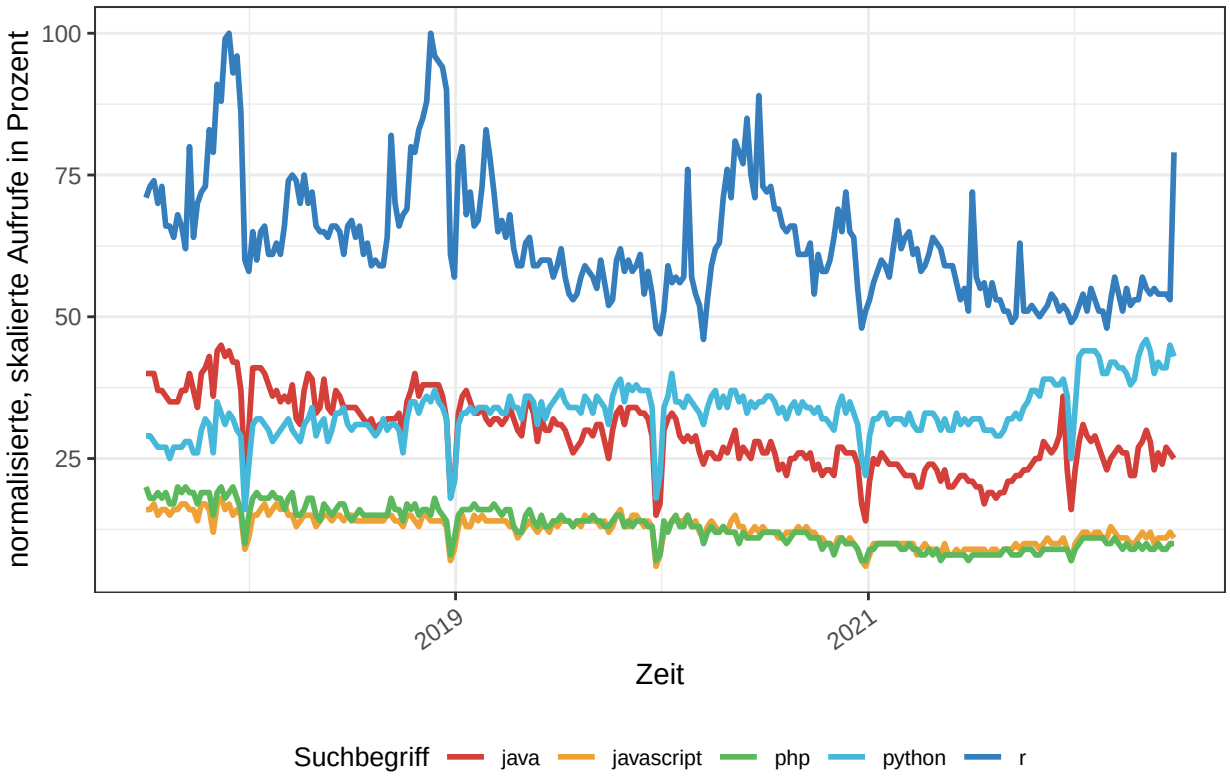
```
for (i in 1:12) {

      list_plots[[i]] <- ggplot(
        data = list_data[[i]][["interest_over_time"]],
        aes(x = as.Date(date),
            y = hits,
            group = factor(keyword),
            color = keyword)) +
        geom_line(size = 1) +
        labs(
          x = "Zeit",
          y = "normalisierte, skalierte Aufrufe in Prozent",
          color = "Suchbegriff",
          caption = paste0("Datenquelle: Google Trends
          (https://www.google.com/trends, abgefragt am ", date_analysis, ").")
          ) +
        theme_bw() +
        theme(
          axis.text.x = element_text(angle = 35, hjust = 1),
          plot.caption = element_text(size = 6, color = "gray60"),
          legend.title = element_text(size = 10),
          legend.text = element_text(size = 8),
          legend.position = "bottom") +
        scale_x_date(date_labels = "%Y",
                     date_breaks = "2 years",
                     date_minor_breaks = "1 year") +
      #  scale_y_continuous(labels = scales::percent_format(scale = 1)) +
        ggsci::scale_colour_locuszoom()

      print(last_plot()) # anzeigen lassen

}
```



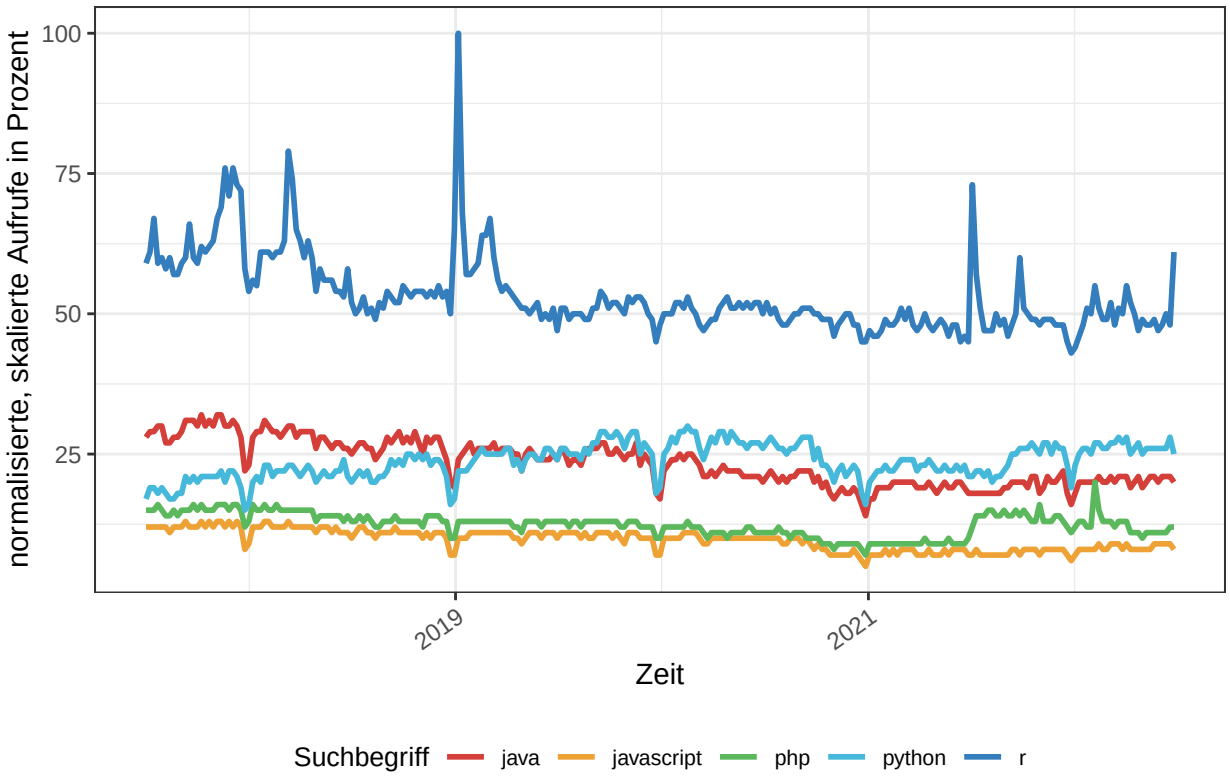
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



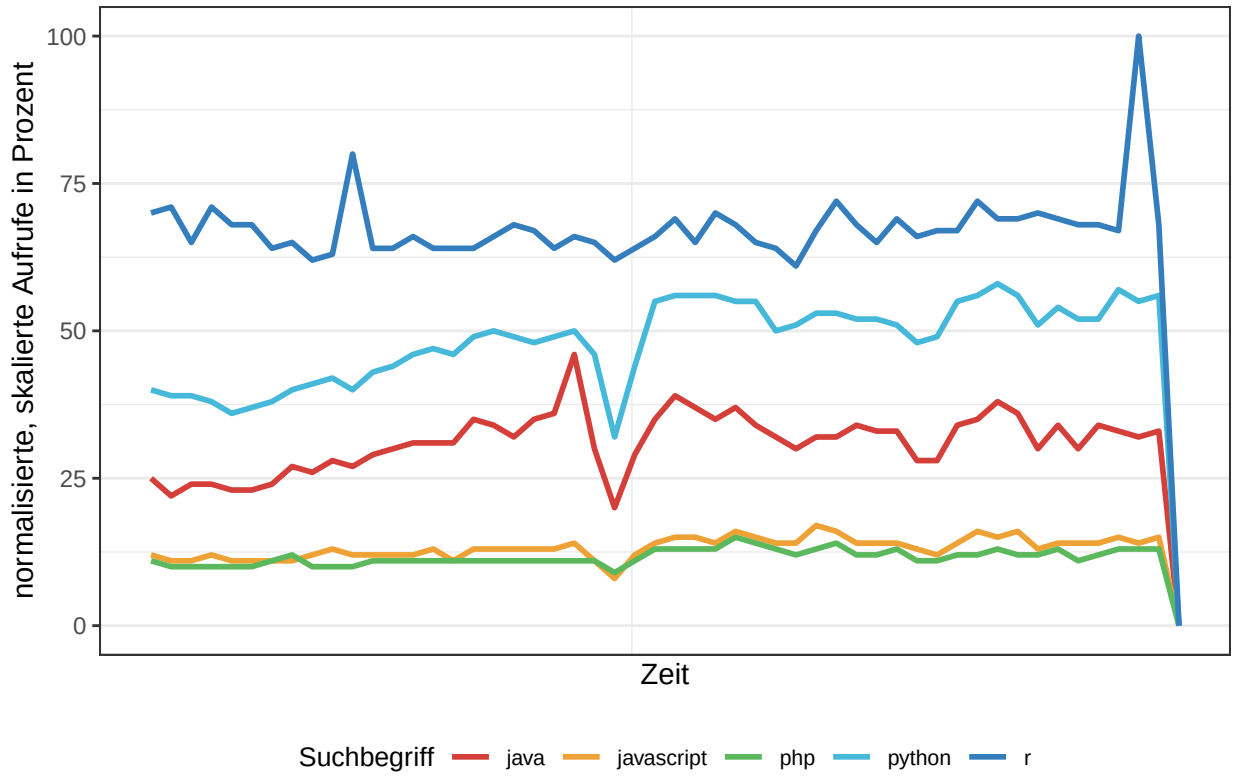
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



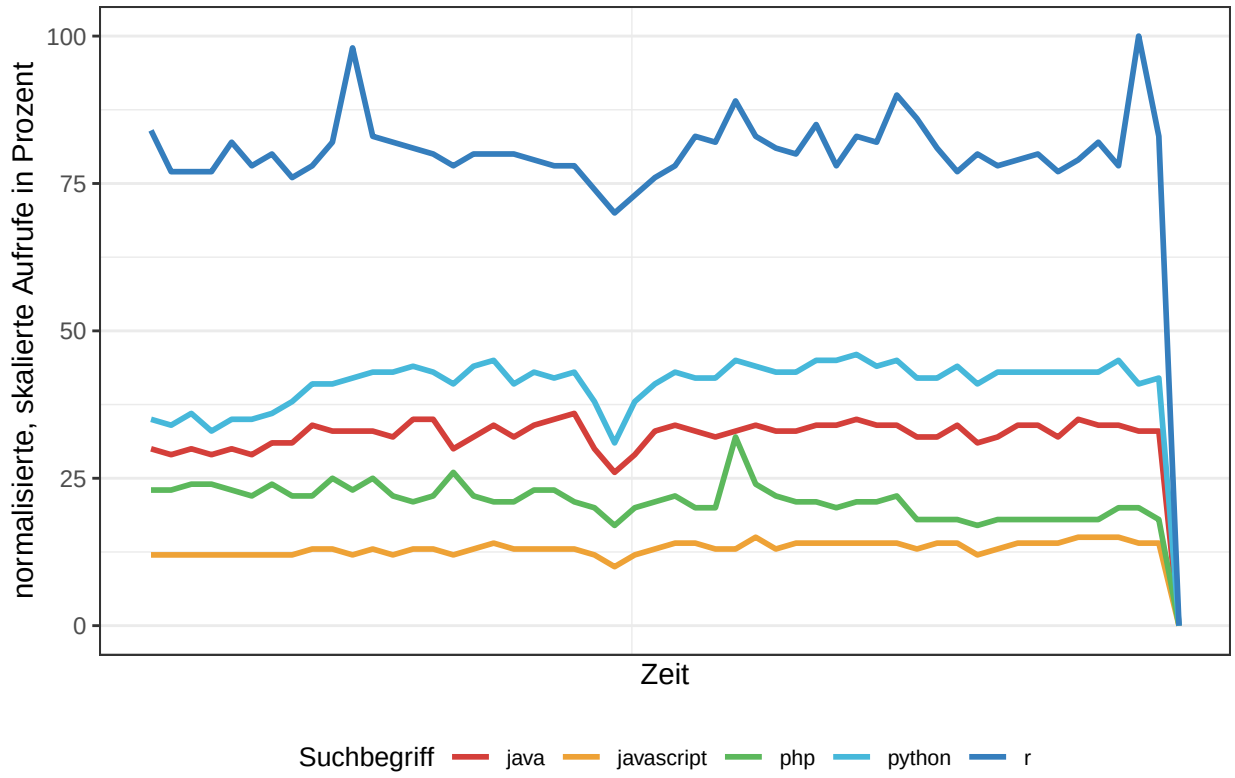
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



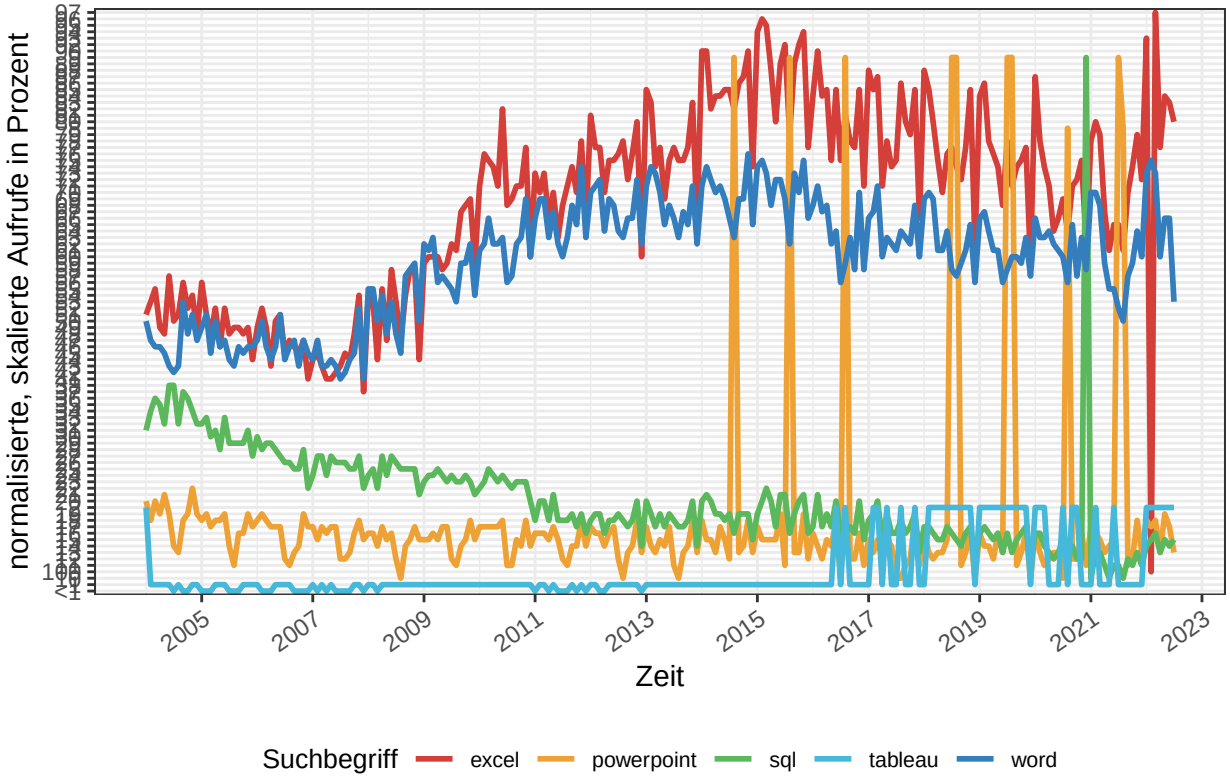
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



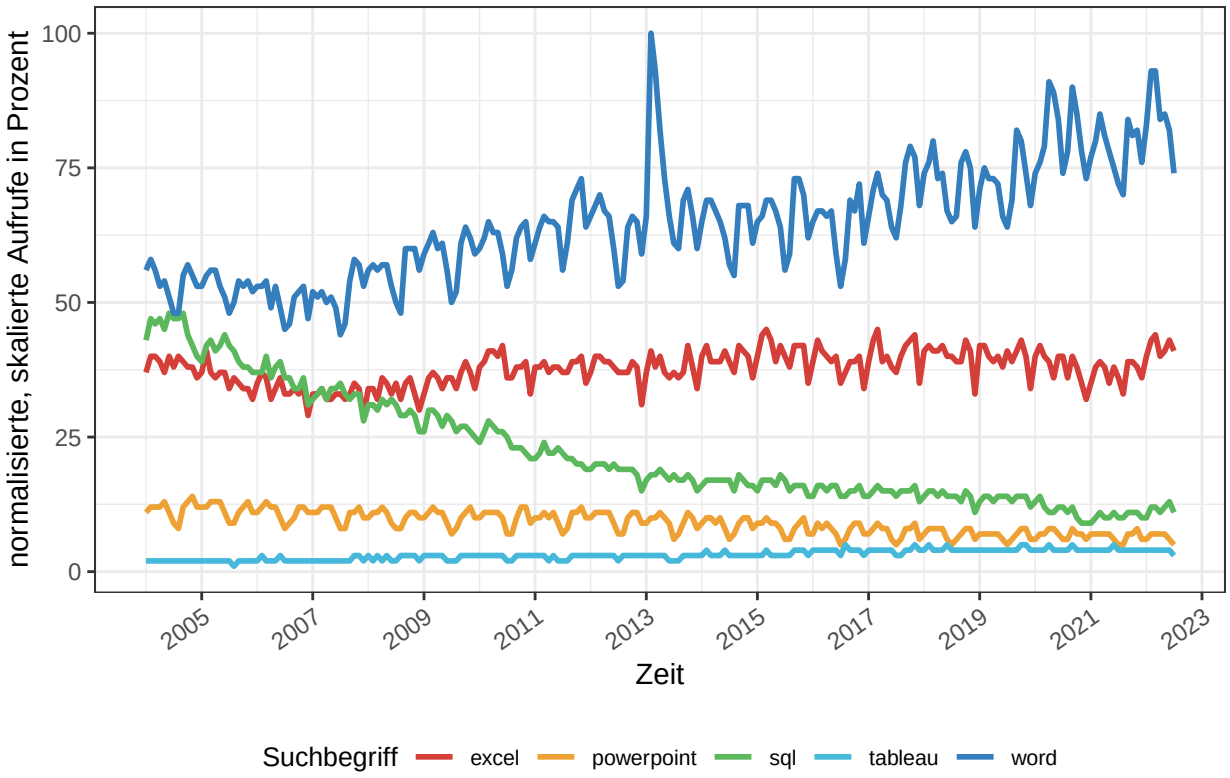
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



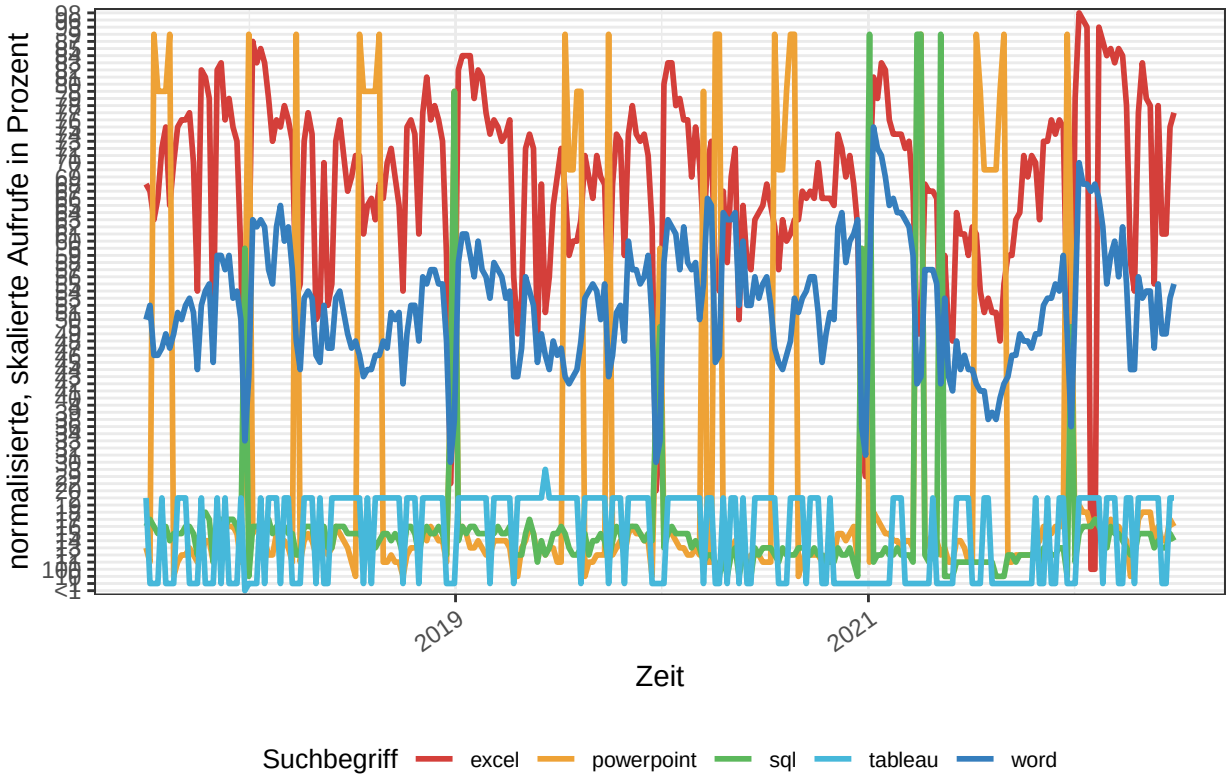
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



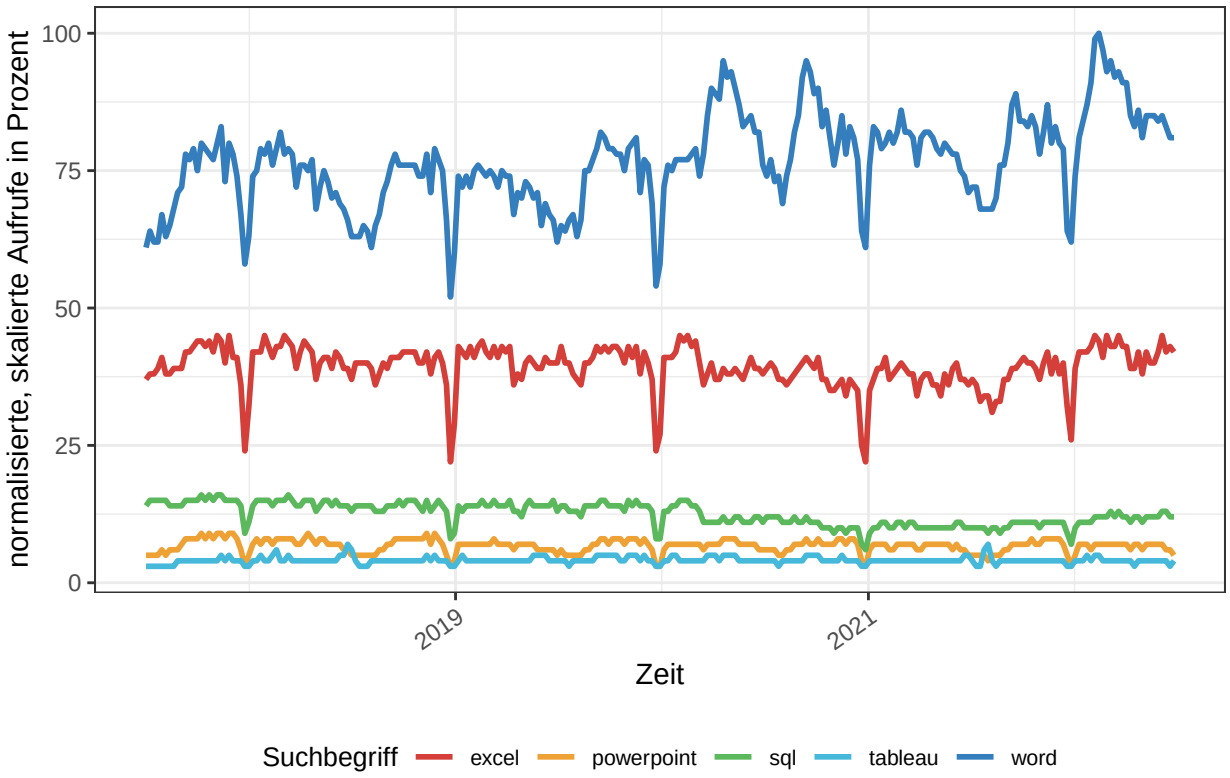
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



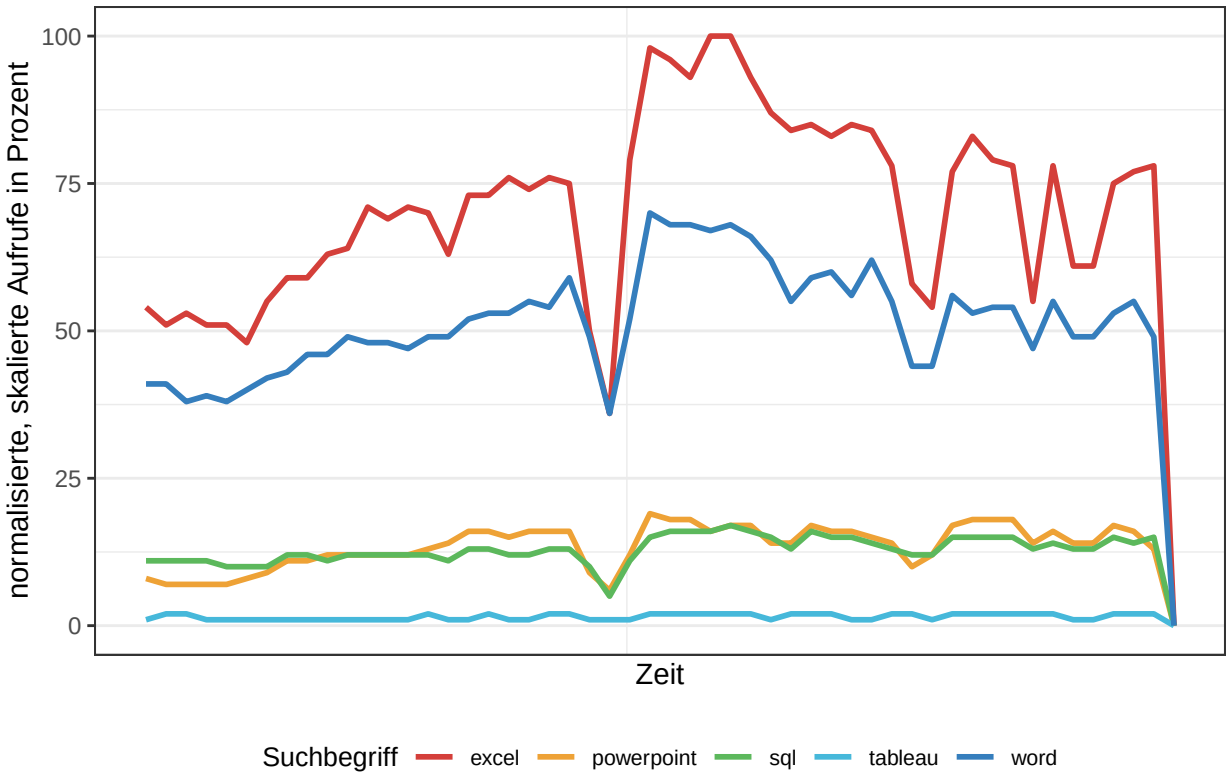
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



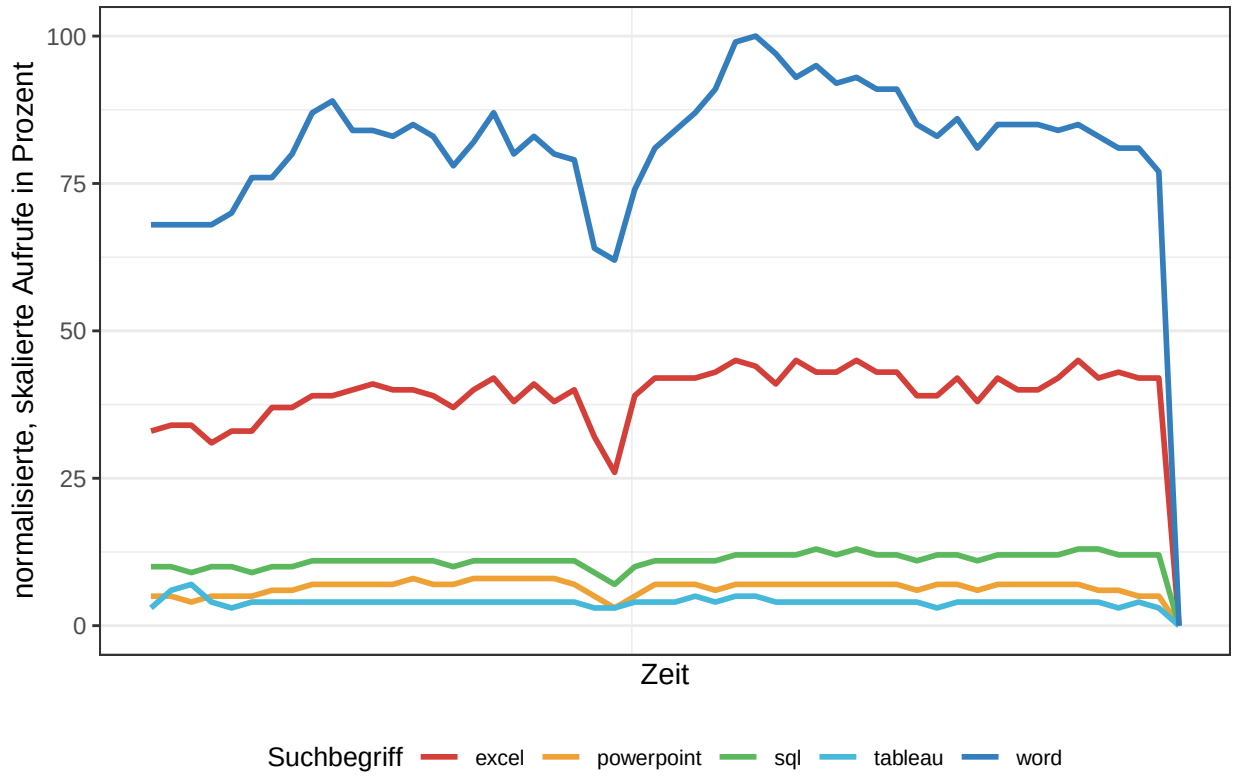
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

```
names(list_plots) <- c("prog_all_de", "prog_all_int",
                       "prog_5y_de", "prog_5y_int",
                       "prog_12_de", "prog_12m_int")
```

Für die erstellten Plots fügen wir als nächstes jeweilis Titel und Untertitel als Vektor bereit und ordnen diese innerhalb einer Schleife basierend auf den Index zu.

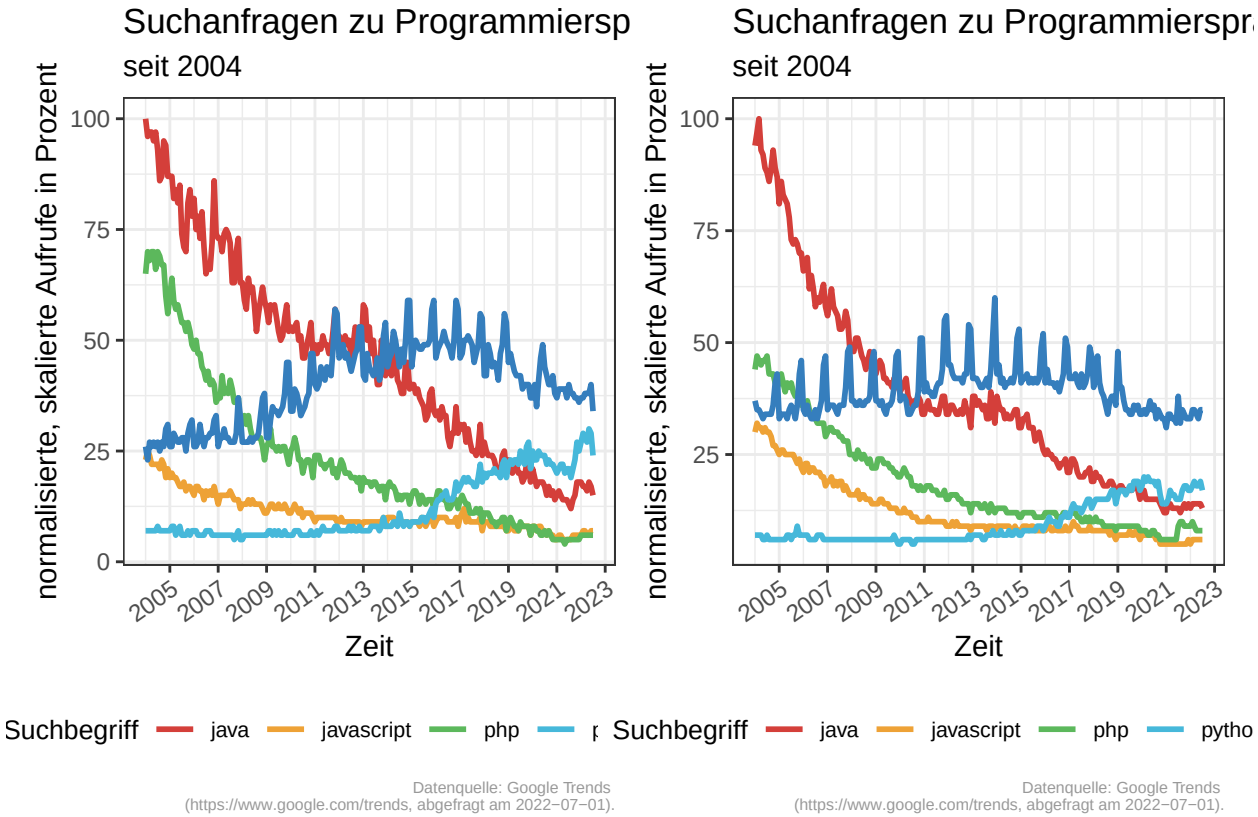
```
indice_5 <- c(3, 4, 9, 10)
indice_12 <- c(5, 6, 11, 12)

titles <- c(
  "Suchanfragen zu Programmiersprachen in Deutschland",
  "Suchanfragen zu Programmiersprachen weltweit",
  "Suchanfragen zu Programmiersprachen in Deutschland",
  "Suchanfragen zu Programmiersprachen weltweit",
  "Suchanfragen zu Programmiersprachen in Deutschland",
  "Suchanfragen zu Programmiersprachen weltweit",
  "Suchanfragen zu Programmen in Deutschland",
  "Suchanfragen zu Programmen weltweit",
  "Suchanfragen zu Programmen in Deutschland",
  "Suchanfragen zu Programmen weltweit",
  "Suchanfragen zu Programmen in Deutschland",
  "Suchanfragen zu Programmen weltweit"
)
```

```r
subtitles <- c(
  "seit 2004",
  "seit 2004",
  "innerhalb der letzten fünf Jahre",
  "innerhalb der letzten fünf Jahre",
  "innerhalb der letzten 12 Monate",
  "innerhalb der letzten 12 Monate",
  "seit 2004",
  "seit 2004",
  "innerhalb der letzten fünf Jahre",
  "innerhalb der letzten fünf Jahre",
  "innerhalb der letzten 12 Monate",
  "innerhalb der letzten 12 Monate"
)

for (n in 1:12) {
  list_plots[[n]] <- list_plots[[n]] + labs(title = titles[[n]], subtitle = subtitles[[n]])

  if (n %in% indice_5) { # falls innerhalb der letzten 5 Jahre
    list_plots[[n]] <- list_plots[[n]] + scale_x_date(date_labels = "%Y",
                                                      date_breaks = "1 years",
                                                      date_minor_breaks = "1 months")
  } else if (n %in% indice_12) { # falls innerhalb der letzten 12 monate
    list_plots[[n]] <- list_plots[[n]] + scale_x_date(date_labels = "%B %Y",
                                                      date_breaks = "1 months")
  }

  ggsave(filename = paste0("plot_", n, ".pdf"),
         plot = list_plots[[n]],
         width = 9, height = 5,
         path = here::here("plots"))

  ggsave(filename = paste0("small_plot_", n, ".pdf"),
         plot = list_plots[[n]],
         width = 6,
         height = 6,
         path = here::here("plots"))

  write_csv(as.data.frame(
    list_data[[1]][["interest_over_time"]]),
    path = here::here("tables", paste0("table_time_", n, ".csv")))
}
```

```
## Warning: The ‘path‘ argument of ‘write_csv()‘ is deprecated as of readr 1.4.0.
## Please use the ‘file‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.

## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
```

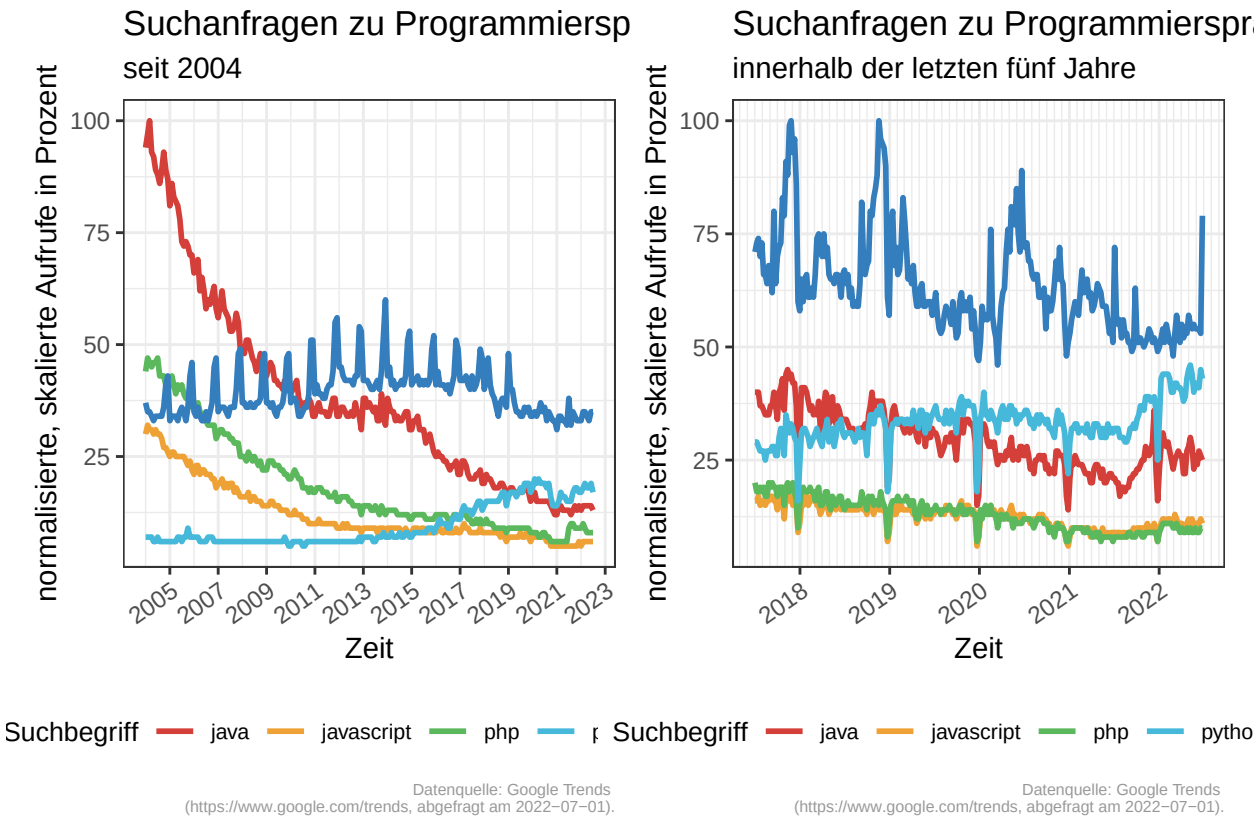
```
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```r
list_plots_2col <- list() # leeere Liste zum Fuellen mit Doppel-Plots

for (i in 1:5) {

  list_plots_2col[[i]] <- list_plots[[i]] + list_plots[[i + 1]]
  print(list_plots_2col[[i]])

  ggsave(filename = paste0("2col_plot_", i, ".pdf"),
         plot = list_plots_2col[[i]],
         width = 12,
         height = 6,
         path = here::here("plots"))

}
```



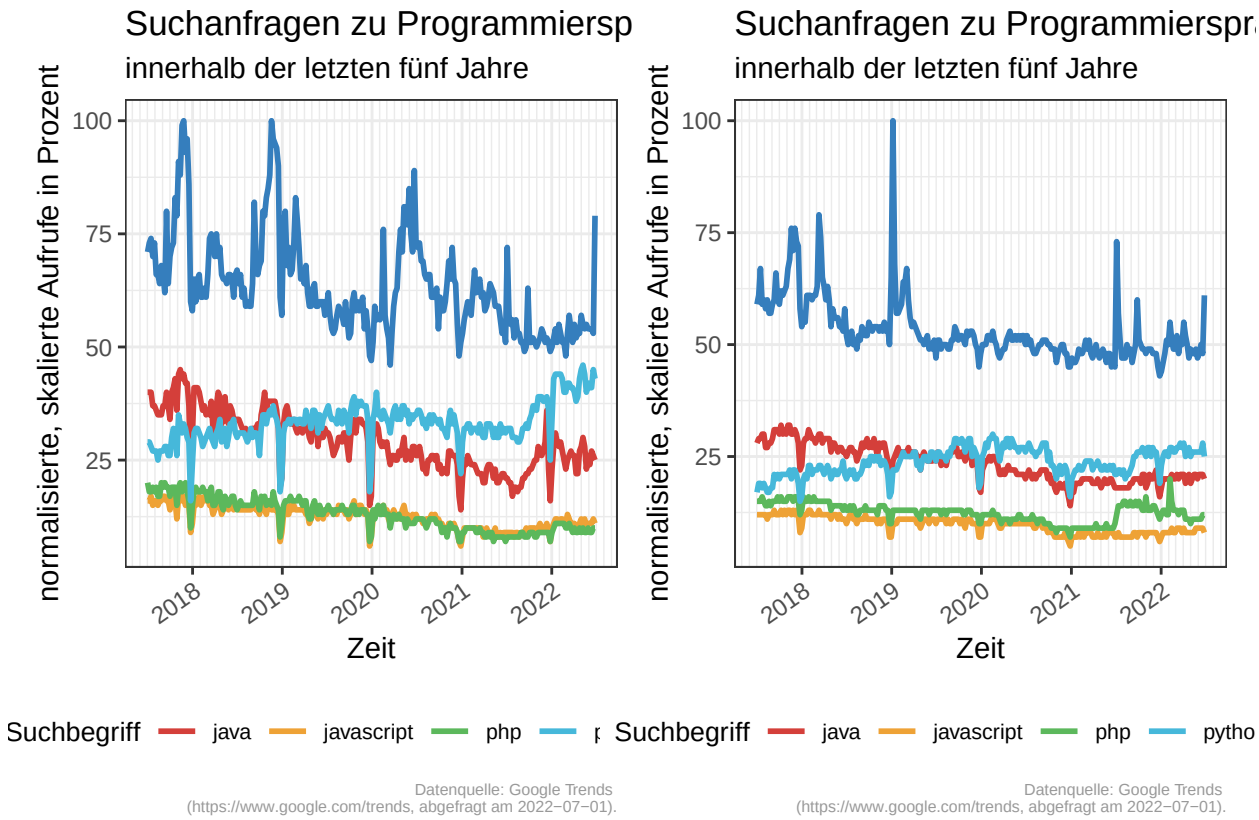
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



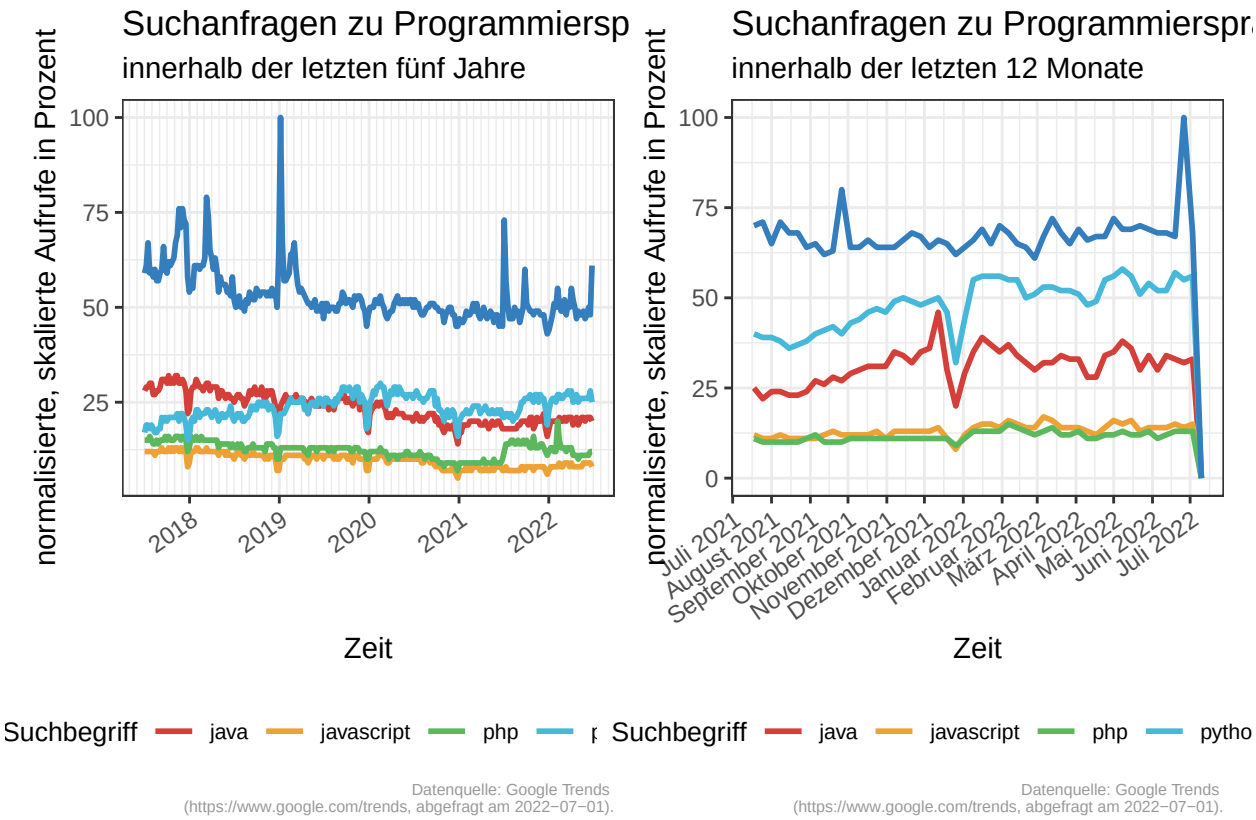
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



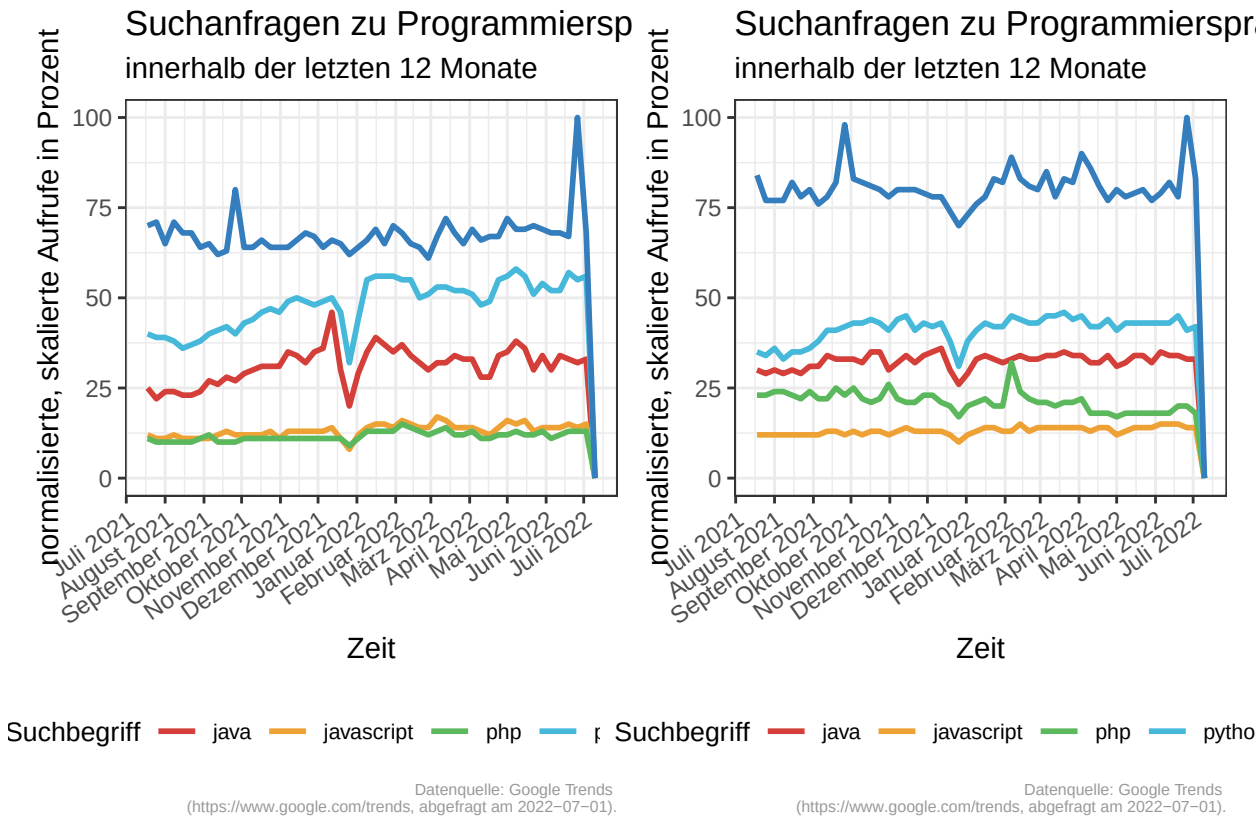
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

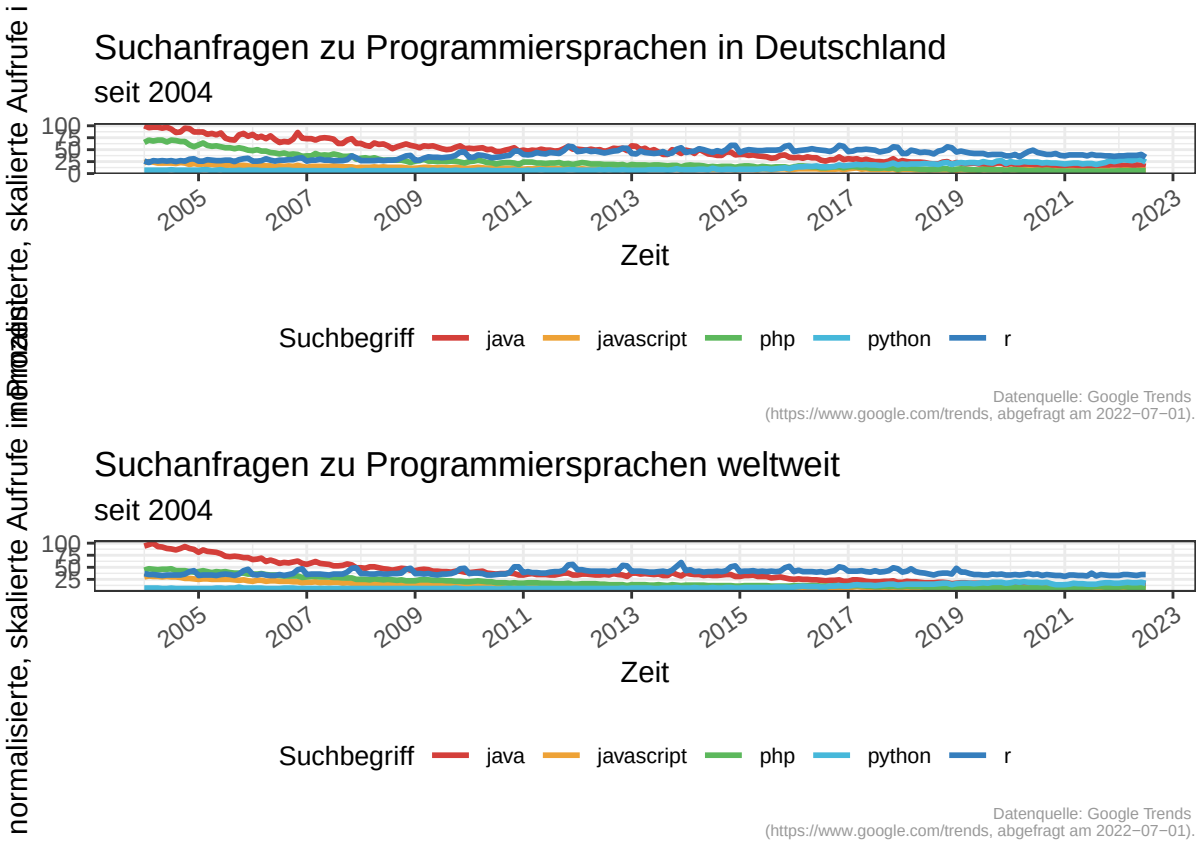


Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

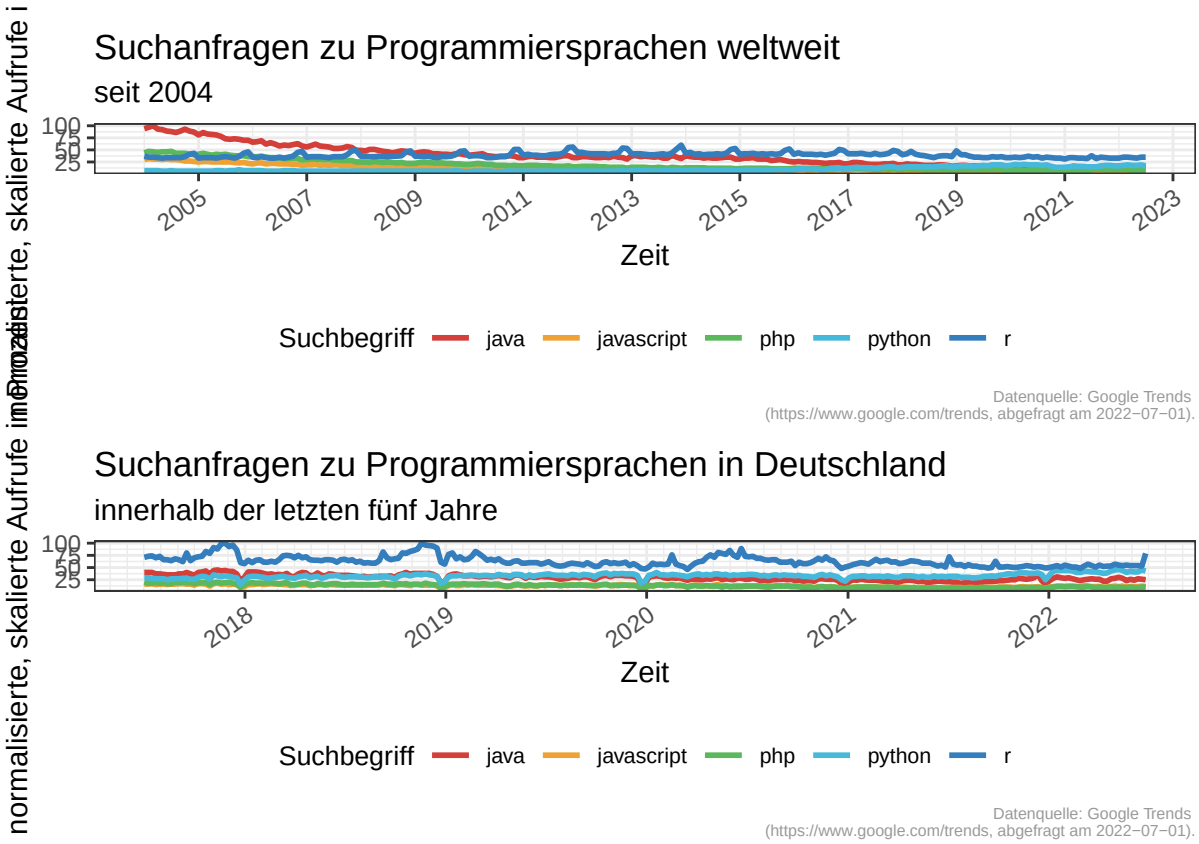
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



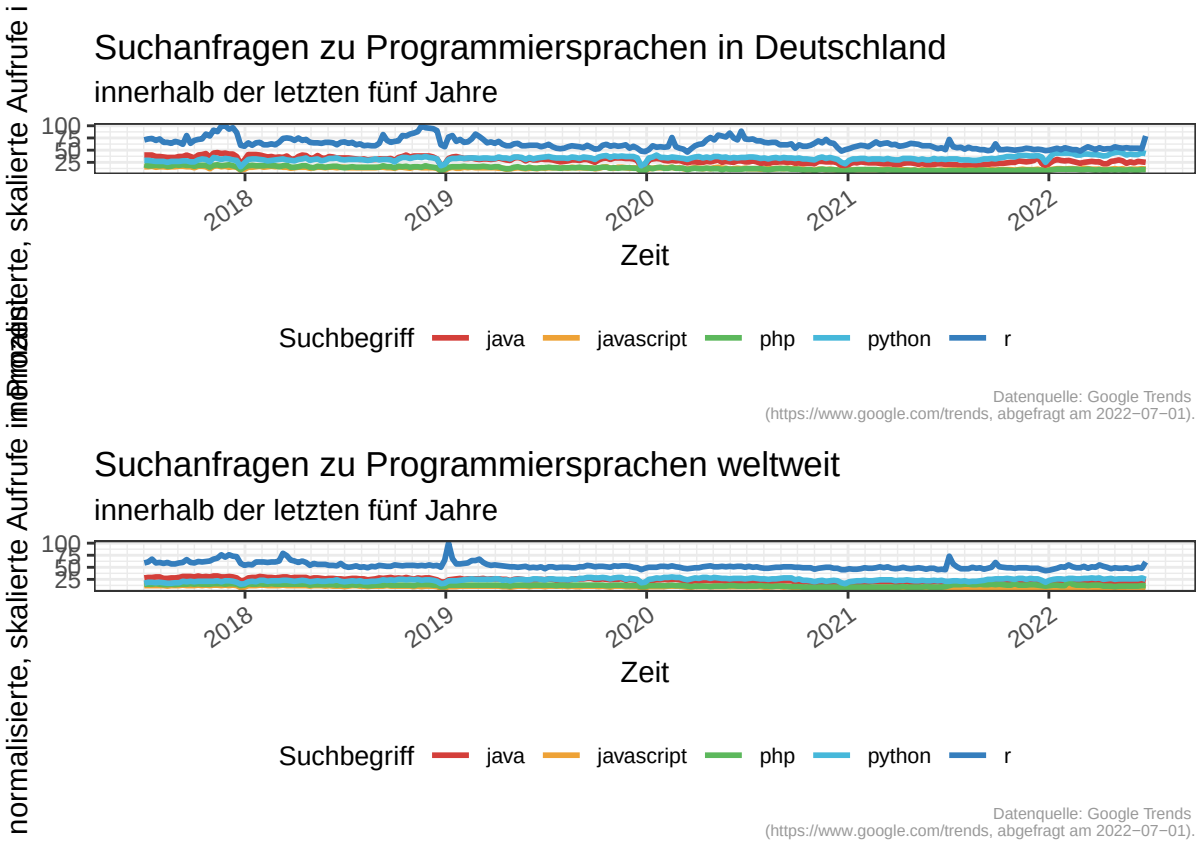

```
list_plots_1col <- list() # leeere Liste zum Fuellen mit Doppel-Plots

for (i in 1:5) {

  list_plots_1col[[i]] <- list_plots[[i]] / list_plots[[i + 1]]
  print(list_plots_1col[[i]])

  ggsave(filename = paste0("1col_plot_", i, ".pdf"),
         plot = list_plots_1col[[i]],
         width = 8,
         height = 8,
         path = here::here("plots"))

}
```


Suchanfragen zu Programmiersprachen in Deutschland
seit 2004
100
75
50
25
0
2005
2007
2009
2011
2013
2015
2017
2019
2021
2023
Zeit
normalisierte, skalierte Aufrufe im Monatsdurchschnitt
Suchbegriff
java
javascript
php
python
r
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

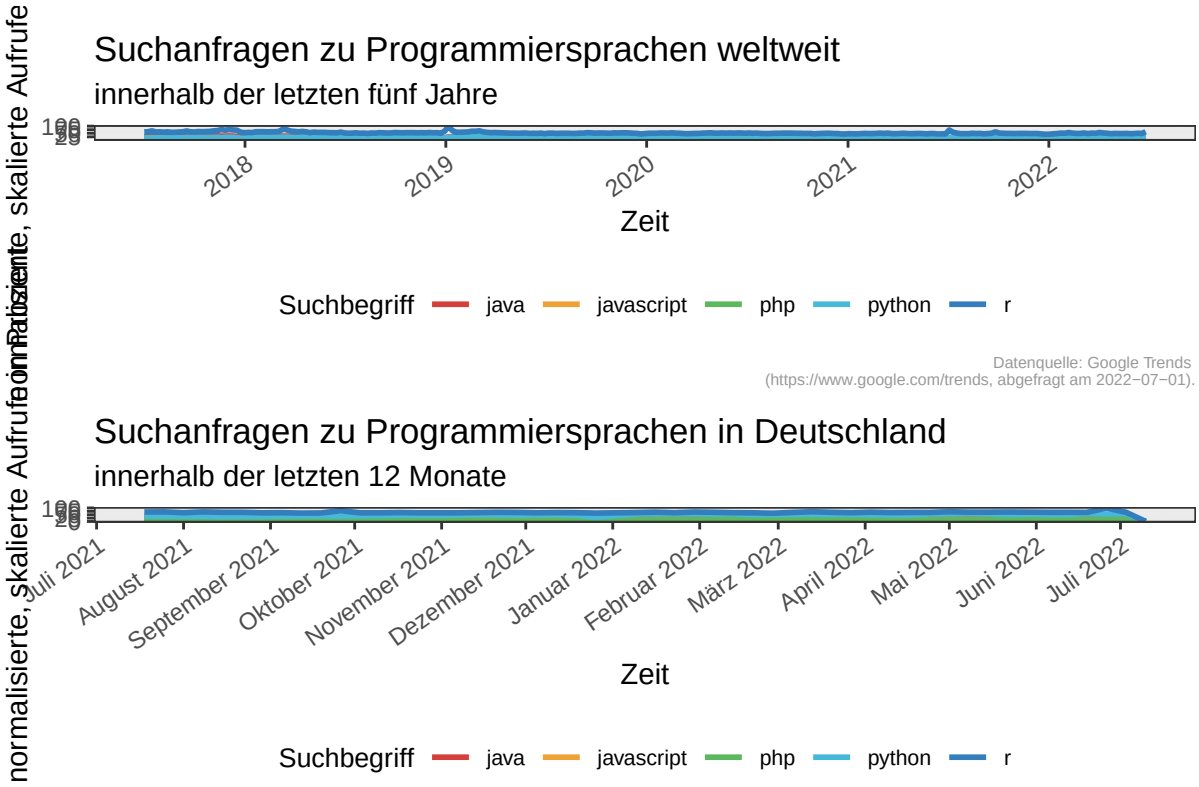
Suchanfragen zu Programmiersprachen weltweit
seit 2004
100
75
50
25
2005
2007
2009
2011
2013
2015
2017
2019
2021
2023
Zeit
Suchbegriff
java
javascript
php
python
r
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).


Suchanfragen zu Programmiersprachen weltweit
seit 2004
100
75
50
25
2005
2007
2009
2011
2013
2015
2017
2019
2021
2023
Zeit
Suchbegriff
java
javascript
php
python
r
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

Suchanfragen zu Programmiersprachen in Deutschland
innerhalb der letzten fünf Jahre
100
75
50
25
2018
2019
2020
2021
2022
Zeit
Suchbegriff
java
javascript
php
python
r
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

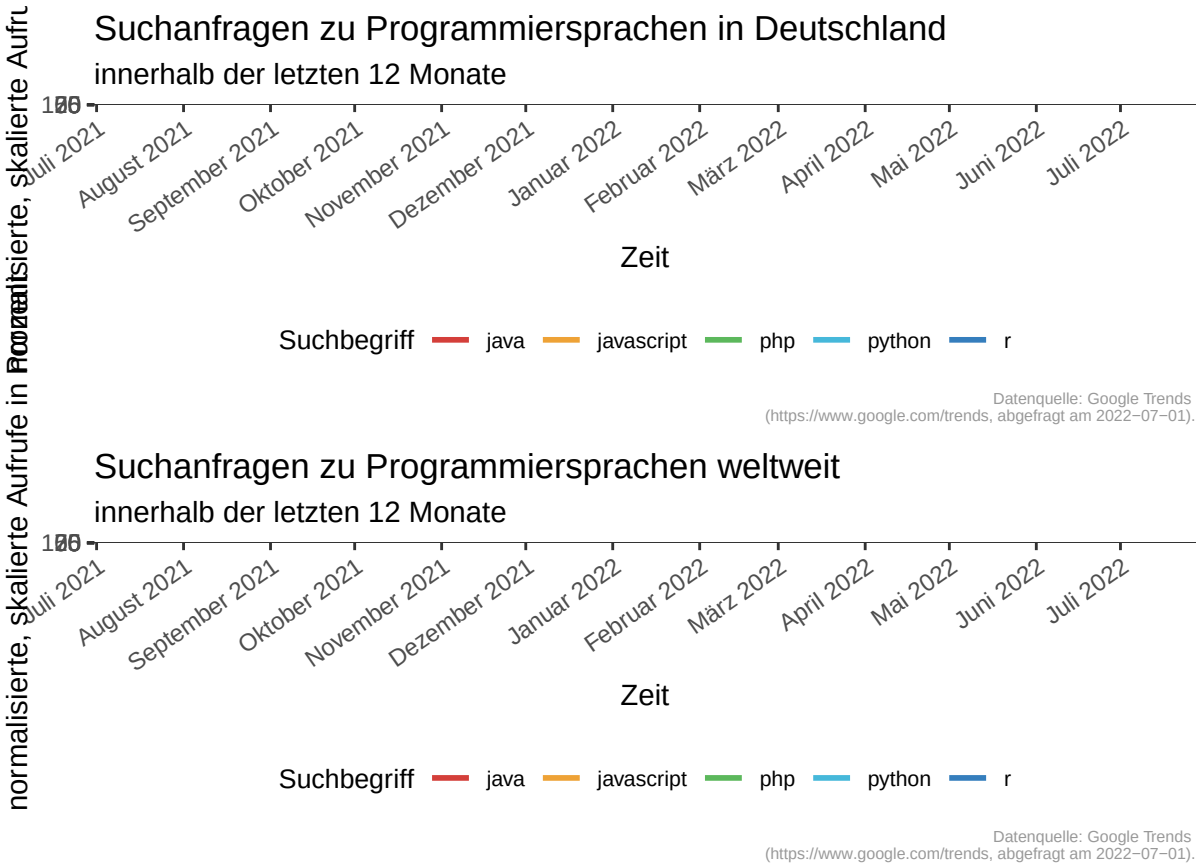

Suchanfragen zu Programmiersprachen in Deutschland
innerhalb der letzten fünf Jahre
normalisierte, skalierte Aufrufe im Monat
100
75
50
25
2018
2019
2020
2021
2022
Zeit
Suchbegriff
java
javascript
php
python
r
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

Suchanfragen zu Programmiersprachen weltweit
innerhalb der letzten fünf Jahre
normalisierte, skalierte Aufrufe im Monat
100
75
50
25
2018
2019
2020
2021
2022
Zeit
Suchbegriff
java
javascript
php
python
r
Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).



Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).

Datenquelle: Google Trends
(https://www.google.com/trends, abgefragt am 2022−07−01).




```
world <- map_data("world") # Weltdaten einlesen

# aendert die Regionnamen zu denen, die Google Trends verwendet
world <- world %>%
  mutate(region = replace(region, region=="USA", "United States")) %>%
  mutate(region = replace(region, region=="UK", "United Kingdom"))

# perform search
# res_world <- gtrends("wantok", time = "all")

# create data frame for plotting
#res_world$interest_by_country %>%
  #filter(location %in% world$region, hits > 0) %>%
  #mutate(region = location, hits = as.numeric(hits)) %>%
  #select(region, hits) -> my_df

my_df <- list_data[[2]][["interest_by_country"]] %>%
  filter(location %in% world$region, hits > 0) %>%
  mutate(region = location, hits = as.numeric(hits))
```

```
## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt
```

```
ggplot() +
  geom_map(data = world,
           map = world,
```

```
          aes(x = long, y = lat, map_id = region),
          fill="#ffffff", color="#ffffff", size=0.15) +
  geom_map(data = my_df,
          map = world,
          aes(fill = hits, map_id = region),
          color="#ffffff", size = 0.15) +
  scale_fill_continuous(low = 'grey', high = 'red') +
  theme(axis.ticks = element_blank(),
        axis.text = element_blank(),
        axis.title = element_blank())
```

```
## Warning: Ignoring unknown aesthetics: x, y
```