# Vignette ifwtrends

```
library(ifwtrends)
library(knitr)
library(dplyr)
#>
#> Attache Paket: 'dplyr'
#> Die folgenden Objekte sind maskiert von 'package:stats':
#>
#>     filter, lag
#> Die folgenden Objekte sind maskiert von 'package:base':
#>
#>     intersect, setdiff, setequal, union
library(tibble)
library(ggplot2)
```

## Einordnung

Es gibt eine große Literatur, welche Google Trends Daten in unterschiedlichen Modellen benutzt, um verschiedene ökonomische Variablen zu prognostizieren. Einen ausführlichen Überblick gibt hierzu der Bericht *Big Data in der makroökonomischen Analyse* (Kieler Beiträge zur Wirtschaftspolitik Nr. 32), Abschnitt 2.3.4. Zusammenfassend lässt sich sagen, dass die zusätzliche Verwendung von Google Daten dort, wo schon andere Indikatoren vorliegen, zu keiner relevanten Verbesserung der Prognosen führt. Prognosen welche nur auf Google Daten beruhen schneiden aber meist ähnlich gut ab wie solche nur mit klassischen Indikatoren. Die Stärke der Google Daten liegt vor allem in ihrer hohen regionalen und zeitlichen Verfügbarkeit. Dies führt unserer Ansicht nach zu drei zentralen Anwendungsfeldern für Google Daten in der Prognose.

1. Als Prognosevariablen, wenn klassische, quartalsweise erscheinende, Variablen noch nicht vorliegen.
2. Als Prognosevariablen für Länder für die sonst wenige Indikatoren vorliegen.
3. Als Echtzeitindikatoren für die wirtschaftliche Aktivität in Krisenzeiten, wenn schnell politische Entscheidungen gefällt werden müssen.

## Google-Trends Daten

### Suchanfragen

Google stellt Zeitreihen der relativen Häufigkeit eines Suchbegriffes zur Verfügung. Hier folgt zunächst ein grundlegende Beschreibung der Daten um dann deren statistische Besonderheiten näher zu beleuchten.

## Google-Trends Daten

Google stellt Zeitreihen der relativen Häufigkeit eines Suchbegriffes in Form von Google Trends zur Verfügung. Hier folgt zunächst eine grundlegende Beschreibung der Daten, um dann deren statistische Besonderheiten näher zu beleuchten.

Die Zeitreihen reichen bis 2004-01-01 zurück und können geografisch eingeschränkt werden, z.B. nach Ländern oder subnationalen Entitäten. Hier sollte beachtet werden, dass Google in einigen Ländern, insbesondere autoritär regierten wie China, nicht verfügbar ist und für diese Länder deshalb keine Daten vorhanden sind. Bei den möglichen Suchanfragen unterschiedet man zwischen *terms*, *topics* und *categories*. Google definiert einen *term* wie folgt (Google1 2021):

"Search terms show matches for all terms in your query, in the language given.

- If you search the term 'banana,' results include terms like 'banana' or 'banana sandwich'
- If you specify 'banana sandwich,' results include searches for 'banana sandwich,' as well as 'banana for lunch' and 'peanut butter sandwich'".

Die Definition von eines *topic* ist (ebd.):

"Topics are a group of terms that share the same concept in any language. Topics display below search terms.

If you search the topic 'London,' your search includes results for topics such as: * 'Capital of the UK' * 'Londres', which is 'London' in Spanish".

Die Eingabe eines Suchbegriffs als Topic führt also vor allem zu einer Invarianz des Suchbegriffs gegenüber der Landessprache. Die Differenzierung zwischen Topic und Term ist aber lediglich online in der Google Suchmaske möglich. Bei der Nutzung der R-Funktionen zum Herunterladen von Google-Daten sollte der Suchbegriff also immer in der jeweiligen Landessprache eingegeben werden. Wir unterscheiden deshalb im Folgenden nicht weiter zwischen *term* und *topic* sondern bezeichnen beides als *Suchbegriff*.

Um weiter zu spezifizieren, welche Daten in den Index eingehen sollen, kann bei einer Abfrage zusätzlich eine *category* angegeben werden. Bei Google heißt es hierzu (Google2 2021:

"If you're using Trends to search for a word that has multiple meanings, you can filter your results to a certain category to get data for the right version of the word. For example, if you search for "jaguar," you can add a category to indicate if you mean the animal or the car manufacturer."

Des Weiteren kann der Index auch nur für eine Kategorie ohne Angabe eines speziellen Suchbegriffs berechnet werden, dann gehen alle Suchanfragen welche Google dieser zuordnet in den Index mit ein. Dieses Vorgehen wird im Folgenden häufiger verwendet.

## Statistische Besonderheiten

### Berechnung des Index

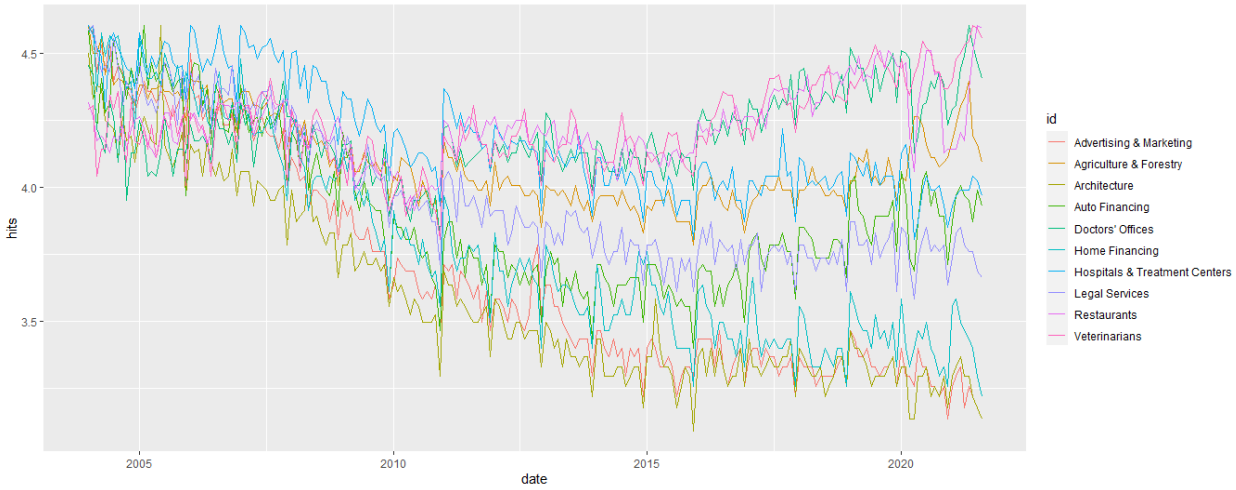
Der Google Index ist ein relativer Index. Er gibt den Anteil der Suchanfragen eines Suchbegriffs/Kategorie zum Zeitpunkt $t$ an der Gesamtzahl der Suchanfragen zu diesem Zeitpunkt $t$ an, normiert mit einer multiplikativen Konstanten, sodass das Maximum des Index im betrachteten Zeitraum bei 100 liegt.

$$SVI_{ct} = \frac{SV_{ct}}{SVT_t} \cdot C_c$$

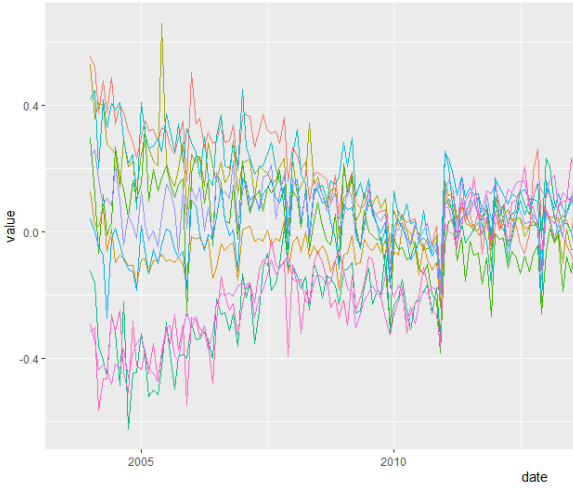
(Woloszko 2020). Hierbei hängt $C_c$ vom Zeitfenster ab für welches die Daten heruntergeladen werden. Der Wert des Indexes zum Zeitpunk $t$ kann also unterschiedlich sein, je nachdem welches Zeitfenster man herunterlädt. Dies sollte bei der Arbeit mit den Daten immer beachtet werden. Insbesondere kann für frühere Zeiträume nicht der Index bis zu diesem Zeitpunkt abgeschnitten werden, sondern muss immer neu heruntergeladen werden.

### Allgemeiner Trend

Da die Nutzung von Google und damit auch die Anzahl verschiedener Suchbegriffe stark zugenommen hat, sinkt für jeden einzelnen Suchbegriff der relative Anteil im Zeitverlauf @ref(comtrend).




Im Abschnitt ?? beschreiben wir, wie um diesen Trend bereinigt werden kann und wie dies in unserem Paket implementiert ist.




@ref(trendadj) zeigt die um diesen gemeinsamen Trend bereinigten Reihen.

### Frequenz

Die Google Daten können für Zeitfenster von 12 Monaten auf Tagesbasis abgefragt werden. Für Zeiträume bis 5 Jahre können wöchentliche Daten heruntergeladen werden. Für alle längeren Zeiträume liegen die Daten nur monatlich vor. Da wir die 12-Monatsfenster der täglichen Reihen und die 5-Jahres-Fenster der wöchentlichen Reihen beliebig wählen können, kann mit der Chow-Lin Methode für lange Zeiträume eine Reihe auf Tagesbasis erstellt werden, welche konsistent mit den wöchentlichen und monatlichen Reihen ist. Wir folgen dabei Eichenauer et. al 2020.

### Strukturbruch

Im Januar 2011 wurde die regionale Erfassung der Suchanfragen geändert. Dadurch wird in regional eingeschränkten Reihen in 2011 ein Bruch sichtbar (@ref(comtrend). Auch in 2016 wurde die Methode

zur Datenerhebung nochmals verändert, was auch einen Strukturbruch in den Reihen zur Folge hat. Wir sind hier der Literatur gefolgt, welche diese Strukturbrüche meistens nicht weiter beachtet. Eine einfache Methode das Problem zu umgehen ist bei der Betrachtung von Änderungsraten die betreffenden Zeiträume auszulassen. Woloszko (2020) ist nach unserem Kenntnisstand das einzige Paper welches diese Strukturbrüche genauer adressiert.

## Funktionen

### pca

## Die Funktion pca nimmt als Argumente mehrere Suchwörter oder Kategorien entgegen. Des weiteren eine Region, das Start- und Enddatum (Default: 2006-01-01 und heute) sowie die Anzahl der zu berechnenden Hauptkomponenten (Default: Anzahl der Zeireihen). Für die Zeitreihen wird hier momentan monatliche Frequenz angenommen.

Die Funktion `pca` nimmt als Argumente mehrere Suchwörter oder Kategorien entgegen. Des weiteren eine Region, das Start- und Enddatum (Default: 2006-01-01 und heute) sowie die Anzahl der zu berechnenden Hauptkomponenten (Default: Anzahl der Zeitreihen). Für die Zeitreihen wird hier momentan eine monatliche Frequenz angenommen.

```
pca(keywords = c("ikea", "saturn"),
    categories = 0,
    geo = "DE",
    start = "2018-01-01",
    end = Sys.Date())
#> # A tibble: 46 x 5
#>    date          PC1     PC2  ikea saturn
#>    <date>      <dbl>   <dbl> <int>  <int>
#>  1 2018-01-01   2.81   3.51     82     56
#>  2 2018-02-01  -4.78  -0.544    75     51
#>  3 2018-03-01  -6.56  -1.89     73     50
#>  4 2018-04-01 -16.8   -6.85     64     43
#>  5 2018-05-01 -17.3  -12.2      59     45
#>  6 2018-06-01 -15.5  -13.1      59     47
#>  7 2018-07-01 -12.8   -9.97     63     48
#>  8 2018-08-01  -4.35   2.59     78     50
#>  9 2018-09-01   4.16  -0.509    79     59
#> 10 2018-10-01   1.01   4.40     82     54
#> # ... with 36 more rows

pca(keywords = NA,
    categories = c(651),
    geo = "DE",
    start = "2018-01-01",
    end = Sys.Date())
#> # A tibble: 46 x 3
#>    date          PC1 `651`
```

```
#>    <date>      <dbl> <int>
#>  1 2018-01-01 -3.98     37
#>  2 2018-02-01 -3.98     37
#>  3 2018-03-01 -3.98     37
#>  4 2018-04-01 -5.98     35
#>  5 2018-05-01  1.02     42
#>  6 2018-06-01 -1.98     39
#>  7 2018-07-01 -3.98     37
#>  8 2018-08-01 34.0      75
#>  9 2018-09-01 -0.978    40
#> 10 2018-10-01 -1.98     39
#> # ... with 36 more rows
```

## roll

Die Funktion `roll` gibt `pca` für jeden Zeitpunkt in einem gegebenen Zeitfenster neu an. Dies gibt eine Tabelle mit den Zeitpunkten im Zeitfenster `start_period` bis `end` zurück. In jeder Spalte ist dann die Ausgabe von pca, wobei `end` der jeweilige Zeitpunkt im Zeitfenster ist. Die restlichen Zeilen zum Tabellenende sind `NA`s. Dies kann insbesondere zur Prognoseevaluation genutzt werden. Sowohl `pca` als auch `roll` könnten bei Bedarf umgeschrieben werden, sodass statt den Hauptkomponenten andere Faktoren berechnet werden

```
roll(keywords = "ikea",
     geo = "DE",
     start_series = "2018-01-01",
     start_period = "2018-05-01",
     end = "2018-12-01")
#> [[1]]
#> # A tibble: 121 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-01    81
#>  2 2018-01-02    90
#>  3 2018-01-03    89
#>  4 2018-01-04    88
#>  5 2018-01-05    85
#>  6 2018-01-06   100
#>  7 2018-01-07    94
#>  8 2018-01-08    67
#>  9 2018-01-09    62
#> 10 2018-01-10    60
#> # ... with 111 more rows
#>
#> [[2]]
#> # A tibble: 152 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-01    76
#>  2 2018-01-02    88
#>  3 2018-01-03    84
#>  4 2018-01-04    86
#>  5 2018-01-05    85
#>  6 2018-01-06   100
#>  7 2018-01-07    90
```

```
#>  8 2018-01-08    63
#>  9 2018-01-09    60
#> 10 2018-01-10    57
#> # ... with 142 more rows
#>
#> [[3]]
#> # A tibble: 182 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-01    78
#>  2 2018-01-02    87
#>  3 2018-01-03    88
#>  4 2018-01-04    88
#>  5 2018-01-05    82
#>  6 2018-01-06    97
#>  7 2018-01-07    91
#>  8 2018-01-08    62
#>  9 2018-01-09    57
#> 10 2018-01-10    56
#> # ... with 172 more rows
#>
#> [[4]]
#> # A tibble: 213 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-01    74
#>  2 2018-01-02    92
#>  3 2018-01-03    86
#>  4 2018-01-04    87
#>  5 2018-01-05    87
#>  6 2018-01-06   100
#>  7 2018-01-07    90
#>  8 2018-01-08    66
#>  9 2018-01-09    59
#> 10 2018-01-10    60
#> # ... with 203 more rows
#>
#> [[5]]
#> # A tibble: 244 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-01    77
#>  2 2018-01-02    90
#>  3 2018-01-03    88
#>  4 2018-01-04    86
#>  5 2018-01-05    86
#>  6 2018-01-06   100
#>  7 2018-01-07    86
#>  8 2018-01-08    65
#>  9 2018-01-09    59
#> 10 2018-01-10    59
#> # ... with 234 more rows
#>
```

```
#> [[6]]
#> # A tibble: 39 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-07    94
#>  2 2018-01-14    88
#>  3 2018-01-21    87
#>  4 2018-01-28    92
#>  5 2018-02-04    87
#>  6 2018-02-11    88
#>  7 2018-02-18    84
#>  8 2018-02-25    83
#>  9 2018-03-04    81
#> 10 2018-03-11    80
#> # ... with 29 more rows
#>
#> [[7]]
#> # A tibble: 43 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-07    92
#>  2 2018-01-14    85
#>  3 2018-01-21    84
#>  4 2018-01-28    91
#>  5 2018-02-04    83
#>  6 2018-02-11    82
#>  7 2018-02-18    79
#>  8 2018-02-25    81
#>  9 2018-03-04    79
#> 10 2018-03-11    80
#> # ... with 33 more rows
#>
#> [[8]]
#> # A tibble: 47 x 2
#>    time       value
#>    <date>     <int>
#>  1 2018-01-07    91
#>  2 2018-01-14    84
#>  3 2018-01-21    82
#>  4 2018-01-28    90
#>  5 2018-02-04    86
#>  6 2018-02-11    84
#>  7 2018-02-18    83
#>  8 2018-02-25    79
#>  9 2018-03-04    77
#> 10 2018-03-11    81
#> # ... with 37 more rows
```

### daily_series

Für lange Zeitfenster liegen keine täglichen Daten vor, sondern nur monatliche. Die Funktion `daily_series` zieht zunächst für rollierende Zeiträume mehrere Stichproben und schätzt daraus dann mit der Chow-Lin-Methode für den ganzen Zeitraum tägliche Daten. Diese sind konsistent mit den Monatsdaten. Da momentan

sehr viele Samples gezogen werden, verursacht die Funktion viele Suchanfragen bei Google, was nach einigen Malen zur vorübergehenden Sperrung der IP führt. Die Anzahl der gezogenen Fenster ist momentan unter der aus dem Originalcode um Anfragen zu sparen. Die dadurch hervorgerufene Abweichung scheint im Moment sehr gering bis 0 zu sein. Eine genaue Evaluation konnte jedoch wegen dem noch nicht gelösten IP-Problem noch nicht durchgeführt werden, steht also noch aus.

```
daily_series(keyword = c("arbeitslos"),
             geo = "DE",
             from = "2021-06-01")
#>  num 46.8
#>  num 4
#>  num 4
#> # A tibble: 130 x 2
#>    time       value
#>    <date>     <dbl>
#>  1 2021-06-01  49.0
#>  2 2021-06-02  55.0
#>  3 2021-06-03  28.0
#>  4 2021-06-04  24.0
#>  5 2021-06-05  47.0
#>  6 2021-06-06  32.0
#>  7 2021-06-07  46.0
#>  8 2021-06-08  49.0
#>  9 2021-06-09  65.0
#> 10 2021-06-10  50.0
#> # ... with 120 more rows
```

## factorR2

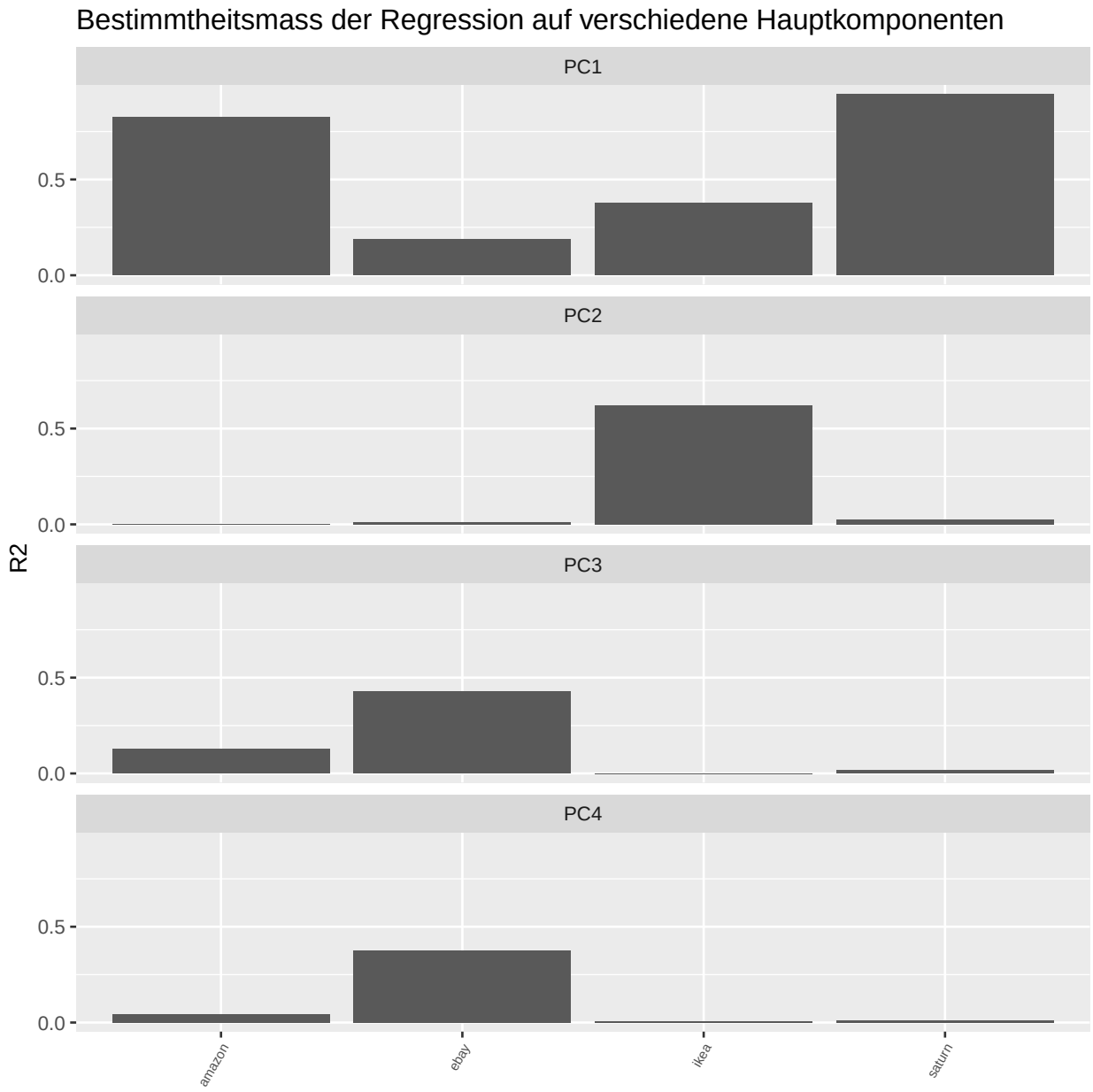
Für schon berechnete Faktoren `factors` aus Zeitreihen `series` ist dies eine Methode, die Erklärungskraft der Faktoren zu bestimmen. Dabei wird für jeden Faktor eine Regression auf jede Zeitreihe vorgenommen und das jeweilige $R^2$ in einer Tabelle abgetragen. Mit Wahl des Parameters `plot=TRUE` wird zusätzlich ein Barplot ausgegeben.

```
dat <- pca(keywords = c("ikea", "saturn", "amazon", "ebay"),
    categories = 0,
    geo = "DE",
    start = "2018-08-01",
    end = Sys.Date())
#> [time]: 'date'

series <- dat %>% select(date, 6:9)
factors <- dat %>% select(date, 2:5)

factorR2(series, factors, plot = T)
#> $res
#> # A tibble: 4 x 5
#>   factors      ikea saturn  amazon   ebay
#>   <chr>       <dbl>  <dbl>   <dbl>  <dbl>
#> 1 PC1     0.375     0.946  0.826   0.187
#> 2 PC2     0.620     0.0269 0.00336 0.0108
#> 3 PC3     0.0000331 0.0148 0.127   0.428
```

```
#> 4 PC4     0.00482   0.0124 0.0441  0.374
#>
#> $plot
```




## Prognose von Einzelhandelsumsätzen

Hier wollen wir die monatliche Veränderung der Einzelhandelsumsätze (Saisonbereinigt) für Deutschland mithilfe von Goole Daten Vorhersagen. Als Ausgangspunkt benutzen wir die Zeitreihen welches das RWI für eine ähnliche Prognose anhand der VGR ausgewählt hat. (https://www.rwi-essen.de/konsumindikator). Diese Reihen werden saisonbereinigt und log-transformiert. Davon betrachten wir dann die erste Differenz. Es gehen jeweils der kontemporäre sowie der um eins und der um zwei gelaggte Datenpunkt der Google-Reihen in das Modell ein.

## Fehlermeldungen

Sollte beim Benutzen der Fehler

```
Error widget$status_code == 200 is not TRUE
```

auftreten, so bedeutet dies, dass das Paket nicht mehr auf Google Trends Daten zugreifen kann, da die IP des benutzten Rechners gesperrt wurde. Beheben kann man dies natürlich nicht. Die einzigen beiden Alternativen, um weiterarbeiten zu können, sind:

- Mit einem anderen Rechner weiterarbeiten.
- Warten, bis die IP-Sperre aufgehoben wurden ist. I.d.R. hält der Bann etwa einen Tag bzw. bis zum Ende des Tages an.

## Quellen

Google1 2021: https://support.google.com/trends/answer/4359550?hl=en/ Google2 2021: https://support.google.com/trends/answer/4359597?hl=en/ Woloszko 2020: Wolozsko, N. (2020): *Tracking activity in real time with Google Trends.* OECD Department working Paper No. 1634.
Eichenauer et. al 2020 *Constructing daily economic sentiment indices based on Google trends.* KOF Working Papers No. 484, 2020.