

# cbsots Tutorial

Rob van Harreveld

2020-09-24

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Table 70076ned: consumption households</b>	<b>2</b>
<b>3</b>	<b>Starting the the Shiny App</b>	<b>2</b>
<b>4</b>	<b>Creating a new table in the the Shiny App</b>	<b>3</b>
<b>5</b>	<b>Selecting dimension keys</b>	<b>6</b>
<b>6</b>	<b>Specifying the Code</b>	<b>8</b>
<b>7</b>	<b>Saving the timeseries coding</b>	<b>9</b>
<b>8</b>	<b>Retrieving timeseries with <code>get_ts</code></b>	<b>9</b>

## 1 Introduction

Package `cbsodataR` can be used to retrieve data from the [CBS open data interface](#). Most tables are complex multi-dimensional tables, and extracting timeseries with appropriate names will usually require some further R programming. The purpose of `cbsots` is to take this programming off your hands.

The package provides the Shiny App “CBS Timeseries Coding” for specifying which portions of the table you want to retrieve and how the names of the individual timeseries (the so-called “timeseries coding”) should be constructed, and a separate function `get_ts` for actually creating the timeseries based on the timeseries coding created with the Shiny App.

The workflow when using package `cbsots` to retrieve timeseries thus consists of two separate steps:

1. Create or modify the timeseries coding for one or more CBS tables with Shiny App “CBS Timeseries Coding”. The timeseries coding is stored in an rds file.
2. Use function `get_ts` to retrieve the timeseries, using the timeseries coding read from the rds file created in step 1.

Step 1 (using the Shiny App) is only necessary if you add a new table, or want to modify the timeseries coding for an existing table.

In this tutorial, an example of this approach is presented. You can follow this tutorial step by step.

## 2 Table 70076ned: consumption households

For this tutorial we will use CBS table 70076ned, that contains the consumption of households in the Netherlands for different kinds of goods and services. The table contains different types of consumption timeseries, such as volume changes, value changes, volume indices and value indices.

The language of this table is Dutch. Package `cbsots` currently only supports Dutch tables. There are a few English table provided by the CBS open data portal, but the overwhelming majority of the tables are only provided in Dutch. In this tutorial, the timeseries names that we use will also be based on Dutch terminology, because most users of package `cbsots` are familiar with naming conventions based on the Dutch terminology.

The aim of this tutorial is to retrieve the following six timeseries:

Table 1: Table 1: Names of the timeseries that we want to retrieve from table 70076ned

Name	Dutch Description	English Description
c__vi	Totale consumptie, volume-indexcijfers	Total consumption, volume indices
c_divi	Consumptie diensten, volume-indexcijfers	Consumption of services, volume indices
c_govi	Consumptie goederen, volume-indexcijfers	Consumption of goods, volume indices
c__wi	Totale consumptie, waarde-indexcijfers	Total consumption, value indices
c_diwi	Consumptie diensten, waarde-indexcijfers	Consumption of services, value indices
c_gowi	Consumptie goederen, waarde-indexcijfers	Consumption of goods, value indices

All indices have base year 2000, so the timeseries have the value 100 in 2000.

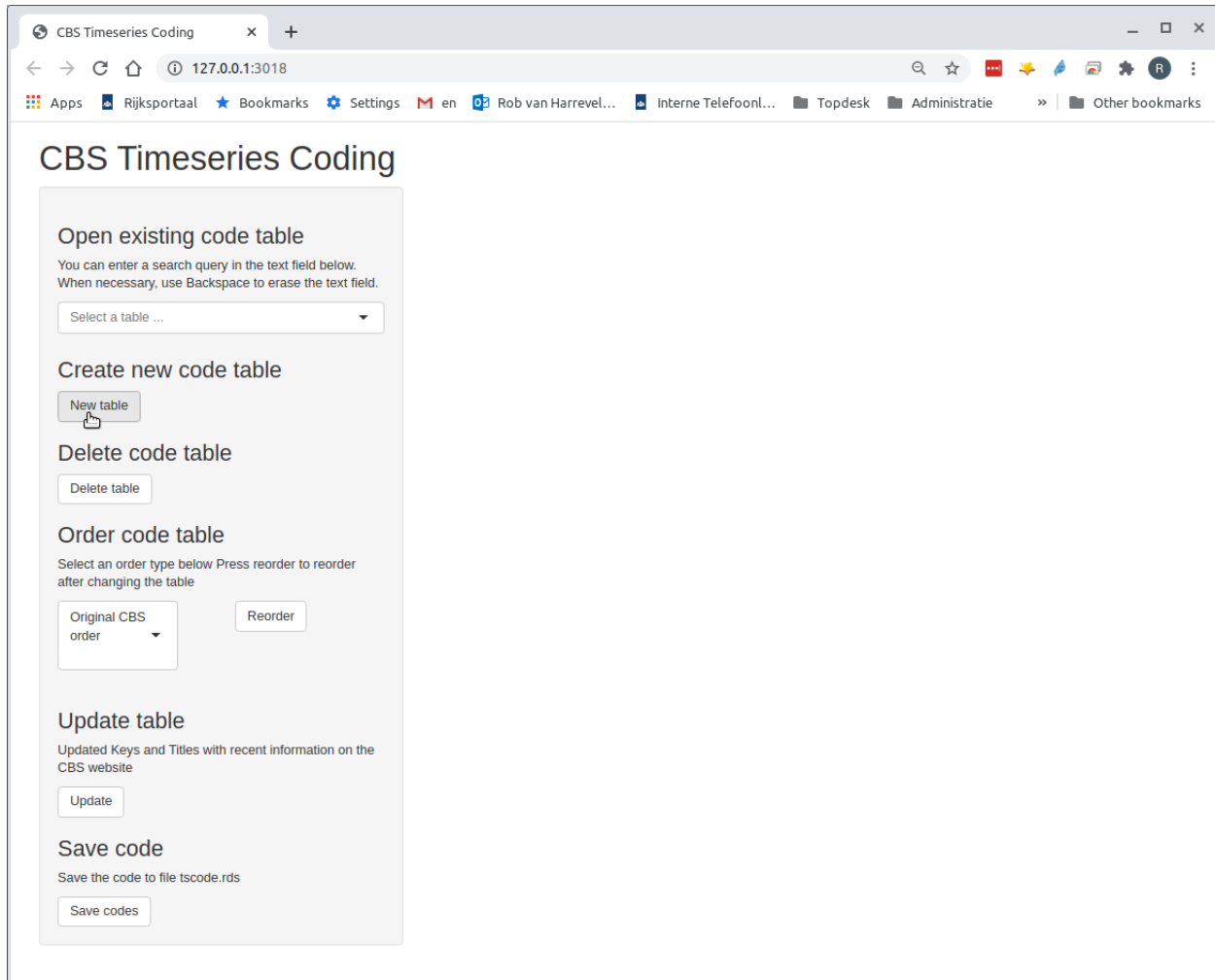
## 3 Starting the the Shiny App

In the remaining part of this Tutorial, I will show how to obtain these timeseries using `cbosts`. You can follow all examples step by step. Because the examples will create an rds file with timeseries coding, I recommend you to run the R code from a clean working directory.

The first step is to start the Shiny app with function `edit_ts_code`:

```
> library(cbsots)
> edit_ts_code("tscode.rds")
```

The first argument ("`tscode.rds`") specifies the name of the rds file were the timeseries coding is stored. The file should not exist yet at this point. The Shiny App should now open in your internet browser:



## 4 Creating a new table in the the Shiny App

Since the file "`tscode.rds`" does not exist yet, there are no existing code tables yet. To create the timeseries coding for table 7006ned, press the **New table** button underneath the text "Create new code table". After a while, a window titled "New table" (see the picture below) appears which allows you to select from the list of available tables. There is some delay between the moment that you press the button and when the Windows appears, because a list of available tables has to be downloaded from the CBS website.

## New Table

You can enter a search query in the text field below. When necessary, use Backspace to erase the text field.

consumptie

- 37741nr - Consumptie, 1988-2010
- 70699ned - Consumptief krediet 1998-2013
- 7205SHFO - Consumptief krediet: 1993 - 1997
- 84094NED - Consumptie naar verbruiksfunctie
- 70076ned - Bestedingen; consumptie huishoudens
- 84093NED - Consumptie naar goederen en diensten
- 81608NED - Consumptie goederen diensten 1969-2012
- 81609NED - Consumptie verbruiksfuncties 1969-2012

Cancel OK

In the drop down list, you can enter a search query, for example "consumptie" (see the picture above). Now a list of table titles containing the word "consumptie" appears. Alternative, you could also have entered the search query "70076" to obtain a list of tables containing "70076" in the title (there is only one such table). From the list, select "70076ned - Bestedingen; consumptie huishoudens". Do not specify the base table<sup>1</sup> in the second drop down list, and press the OK button (see below):

<sup>1</sup>The use of the base table will be described in vignette "The CBS Timeseries Coding Shiny App", which unfortunately still has to be written

## New Table

You can enter a search query in the text field below. When necessary, use Backspace to erase the text field.

70076ned - Bestedingen; consumptie huishoudens

Optionally specify an existing table (the "base table") used to fill in the new table. Leave the text field empty to create an empty new table. When necessary, use Backspace to erase the text field.

Do not use base table

Cancel

OK

Now information about table '70076ned' will be displayed in the Shiny App (some texts are quite small and may be difficult to read, but below I will zoom in on relevant parts of the Shiny App):

## CBS Timeseries Coding

**Open existing code table**

You can enter a search query in the text field below. When necessary, use Backspace to erase the text field.

70076ned - Bestedingen; consumptie huishoudens

**Create new code table**

New table

**Delete code table**

Delete table

**Order code table**

Select an order type below Press reorder to reorder after changing the table

Original CBS \_ Reorder

**Tabel 70076ned - Bestedingen; consumptie huishoudens**

Order used to create names

Topic GoederenEnDiensten

Search ...

Topic GoederenEnDiensten

	Key	Select	Code	Title
1	Volumemutaties_1	<input type="checkbox"/>		Volume - Volumemutaties
2	VolumemutatiesKoopdaggecorrigeerd_2	<input type="checkbox"/>		Volume - Volumemutaties koopdaggecorrigeerd
3	Indexcijfers2000100_3	<input type="checkbox"/>		Volume - Indexcijfers (2000 = 100)
4	Waardemutaties_4	<input type="checkbox"/>		Waarde - Waardemutaties
5	Indexcijfers2000100_5	<input type="checkbox"/>		Waarde - Indexcijfers (2000 = 100)
6	AandelenInDeBinnenlandseConsumptie_6	<input type="checkbox"/>		Waarde - Aandelen in de binnenlandse consumptie
7	AbsoluteWaardenInEuroS_7	<input type="checkbox"/>		Waarde - Absolute waarde in euro's

The right panel shows, among others, two tabs containing a table. Let us have a closer look at the table in the selected tab:

Topic	GoederenEnDiensten			
	Key	Select	Code	Title
1	Volumemutaties_1	<input type="checkbox"/>		Volume - Volumemutaties
2	VolumemutatiesKoopdaggecorrigeerd_2	<input type="checkbox"/>		Volume - Volumemutaties koopdaggecorrigeerd
3	Indexcijfers2000100_3	<input type="checkbox"/>		Volume - Indexcijfers (2000 = 100)
4	Waardemutaties_4	<input type="checkbox"/>		Waarde - Waardemutaties
5	Indexcijfers2000100_5	<input type="checkbox"/>		Waarde - Indexcijfers (2000 = 100)
6	AandelenInDeBinnenlandseConsumptie_6	<input type="checkbox"/>		Waarde - Aandelen in de binnenlandse consumptie
7	AbsoluteWaardeInEuroS_7	<input type="checkbox"/>		Waarde - Absolute waarde in euro's

CBS table 70076ned contains two dimensions: “Topic” and “GoederenEnDiensten” (Goods and Services), besides the time dimension (years, quarters etc.) that is disregarded in the Shiny App. Every CBS table has a dimension called “Topic”, and usually has one or more extra dimensions, such as “GoederenEnDiensten” for this table.

For this particular table, the different “Topics” correspond to different types of consumption timeseries, for example volumemutaties (volume changes), waardemutaties (value changes), and indexcijfers (indices) for both volume and value. Each Topic has a *Key* (a short identifier) and a *Title* (a more detailed description). The *Select* and *Code* columns are the columns that can be modified.

In the next figure upper part of the table for dimension “GoederenEnDiensten” (Goods and Services) is shown. This table is quite long (there are 38 rows), but only the first seven rows are shown. Here find entries for among others, “Consumptie binnenland” (domestic consumption), “Goederen” (Goods), “Voedingsmiddelen” (nourishment), and “Aardappelen, groenten, en fruit” (potatoes, vegetables and fruit).

Topic	GoederenEnDiensten			
	Key	Select	Code	Title
1	8000	<input type="checkbox"/>		Consumptie binnenland
2	9000	<input type="checkbox"/>		Consumptie nationaal
3	4005	<input type="checkbox"/>		1000/4000 Goederen
4	2005	<input type="checkbox"/>		1000/2000 Voedings- en genotmiddelen
5	1000	<input type="checkbox"/>		1000 Voedingsmiddelen
6	1100	<input type="checkbox"/>		1100 Aardappelen, groenten en fruit
7	1200	<input type="checkbox"/>		1200 Vlees en vleeswaren
8	1300	<input type="checkbox"/>		1300 Vlees

## 5 Selecting dimension keys

This table contains 7 Topic and 38 entries for dimensions “GoederenEnDiensten”, so there are 266 timeseries available for this table. As explained in section 2, we are only interested in a subset of 6 timeseries. The Select columns in the tables for “Topic” and “GoederenEnDiensten” determine which timeseries are downloaded.

We are only interested in volume and value indices, so we start by selecting the corresponding Topics (move the mouse to the square and press the left mouse button):


Topic	GoederenEnDiensten			
	Key	Select	Code	Title
1	Volumemutaties_1	<input type="checkbox"/>		Volume - Volumemutaties
2	VolumemutatiesKoopdaggecorrigeerd_2	<input type="checkbox"/>		Volume - Volumemutaties koopdaggecorrigeerd
3	Indexcijfers2000100_3	<input checked="" type="checkbox"/>		Volume - Indexcijfers (2000 = 100)
4	Waardemutaties_4	<input type="checkbox"/>		Waarde - Waardemutaties
5	Indexcijfers2000100_5	<input checked="" type="checkbox"/>		Waarde - Indexcijfers (2000 = 100)
6	AandelenInDeBinnenlandseConsumptie_6	<input type="checkbox"/>		Waarde - Aandelen in de binnenlandse consumptie
7	AbsoluteWaardeInEuroS_7	<input type="checkbox"/>		Waarde - Absolute waarde in euro's

For the dimension “GoederenEnDiensten”, select the rows with titles 8000 - **Binnenlandse consumptie** (domestic consumption). 4005 - **Goederen** (Goods) and 5000 - **Diensten** (Services). To select the last row, scroll down the table.

Because the selected rows are quite far apart in the table, we cannot see at one glance which rows have been selected. It is possible to change the order of the row so that selected rows appear first. On the left panel of the Shiny App, underneath the text “Order code table”, you can select the ordering type: “Original CBS order” (original order of selected table) or “Selected first” (selected rows appear first). See figure below:

## Order code table

Select an order type below Press reorder to reorder after changing the table

Original CBS order 

Reorder

After changing the order from “Original CBS order” to “Selected first”, the result is:

Topic	GoederenEnDiensten			
	Key	Select	Code	Title
1	8000	<input checked="" type="checkbox"/>		Consumptie binnenland
2	4005	<input checked="" type="checkbox"/>		1000/4000 Goederen
3	5000	<input checked="" type="checkbox"/>		5000 Diensten
4	9000	<input type="checkbox"/>		Consumptie nationaal
5	2005	<input type="checkbox"/>		1000/2000 Voedings- en genotmiddelen
6	1000	<input type="checkbox"/>		1000 Voedingsmiddelen

If you select another row, for example the row with Title “2000 - Genotmiddelen”, then the ordering does not change immediately. Press the button **Reorder** to reorder the table again, so that all selected rows appear first in the table.

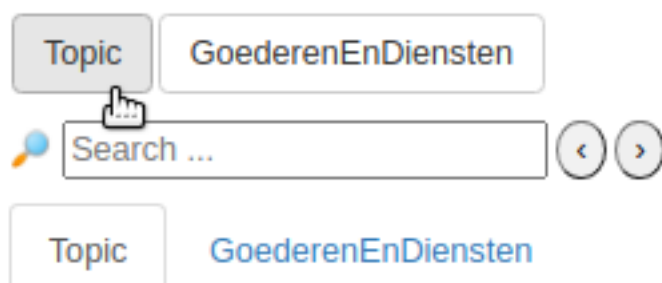
## 6 Specifying the Code

The next step is to fill in data for the Code columns. The text in the Code columns are used to create timeseries names. The names of the timeseries are created by pasting the texts in the Code columns for each combination of “Topic” and “GoederenEnDiensten”.

As explained in Section 2, the names of the timeseries that we want to create should have a prefix based on the type of “GoederenEnDiensten” (Goods and Services): **c\_** (total consumption), **c\_go** (goods), and **c\_di** (services). We also want suffix **vi** for volume indices and **wi** for value indices.

However, by default the text in the code column for “Topic” are the prefixes, and code for the second dimension (here “GoederenEnDiensten”) the suffixes.<sup>2</sup> Which dimension is used for the prefixes and which dimension for the suffixes is determined by the ordering of the buttons above the tabs with dimension table, above the Search text field. See the selection of the Shiny App below:

### Order of dimensions used to create names:



The ordering of these buttons can be changed in the Shiny App: move the mouse cursor to the button **Topic**, then press the left mouse button and drag it to the right of button **GoederenEnDiensten**. Now you can release the button. As a result you will get:

### Order of dimensions used to create names:



Now fill in the Code columns for selected rows:

Topic	GoederenEnDiensten			
	Key	Select	Code	Title
1	Volumemutaties_1	<input type="checkbox"/>		Volume - Volumemutaties
2	VolumemutatiesKoopdaggecorrigeerd_2	<input type="checkbox"/>		Volume - Volumemutaties koopdaggecorrigeerd
3	Indexcijfers2000100_3	<input checked="" type="checkbox"/>	vi	Volume - Indexcijfers (2000 = 100)
4	Waardemutaties_4	<input type="checkbox"/>		Waarde - Waardemutaties
5	Indexcijfers2000100_5	<input checked="" type="checkbox"/>	wi	Waarde - Indexcijfers (2000 = 100)
6	AandelenInDeBinnenlandseConsumptie_6	<input type="checkbox"/>		Waarde - Aandelen in de binnenlandse consumptie

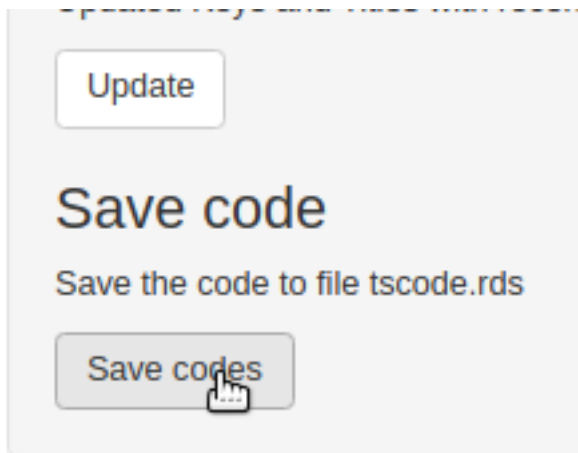
<sup>2</sup>If a table has more than two dimension, the code for the intermediate dimensions are used as infixes



Topic		GoederenEnDiensten		
	Key ▾	Select ▾	Code ▾	Title ▾
1	8000	<input checked="" type="checkbox"/>	c___	Consumptie binnenland
2	4005	<input checked="" type="checkbox"/>	c_go	1000/4000 Goederen
3	5000	<input checked="" type="checkbox"/>	c_di	5000 Diensten
4	0000	<input type="checkbox"/>		Consumptie nationaal

## 7 Saving the timeseries coding

The final step is to save the timeseries coding to a file "tscode.rds": press the **Save codes** buttons on the lower left side of the Shiny App. Changes are not saved automatically. See the figure below):



## 8 Retrieving timeseries with get\_ts

To create the timeseries, first read the timeseries coding created by the Shiny App:

```
> ts_code <- readRDS("tscode.rds")
```

Then function `get_ts` can be used to finally create the timeseries:

```
> data <- get_ts("70076NED", ts_code)
```

Downloading table 70076ned ...

Filters:

\$GoederenEnDiensten

```
[1] "      8000" "      4005" "      5000"
```

Retrieving data from table '70076ned'

```
http://opendata.cbs.nl/ODataFeed/odata/70076ned/TypedDataSet?$format=json&$filter=(GoederenEnDiensten%20
```

Done!

The result is a list with class `table_ts`, with components "Y" (annual timeseries), "Q" (quarterly timeseries) and "M" (monthly timeseries):

```
> data
```

```
table_ts object
```

```
Frequency Y :
```

```
Topleft part of the result (the first 6 rows and 10 columns):
```

	c___vi	c___wi	c_divi	c_diwi	c_govi	c_gowi
1995	80.5	71.5	80.1	69.0	80.9	73.9
1996	84.1	76.1	83.3	73.8	84.9	78.3
1997	87.3	80.8	86.8	79.2	87.7	82.5
1998	91.7	86.7	91.1	85.1	92.2	88.2
1999	96.4	92.9	95.8	91.8	97.0	94.1
2000	100.0	100.0	100.0	100.0	100.0	100.0

```
Frequency Q :
```

```
Topleft part of the result (the first 6 rows and 10 columns):
```

	c___vi	c___wi	c_divi	c_diwi	c_govi	c_gowi
1995Q1	79.9	70.5	79.5	67.9	80.4	73.1
1995Q2	80.5	71.6	80.4	69.1	80.6	74.1
1995Q3	79.5	70.7	80.5	69.9	78.5	71.5
1995Q4	82.1	73.1	80.1	69.3	84.0	76.9
1996Q1	83.6	75.1	82.1	72.2	85.0	77.9
1996Q2	83.8	76.0	83.3	73.6	84.3	78.3

```
Frequency M :
```

```
Topleft part of the result (the first 6 rows and 10 columns):
```

	c___vi	c___wi	c_divi	c_diwi	c_govi	c_gowi
1995M01	80.9	71.1	78.6	67.3	83.3	74.9
1995M02	76.1	66.9	79.1	67.6	73.1	66.3
1995M03	82.8	73.5	80.6	68.7	84.9	78.2
1995M04	78.8	70.2	79.8	68.6	77.9	71.8
1995M05	80.4	71.6	80.7	69.3	80.2	73.9
1995M06	82.3	73.1	80.8	69.4	83.6	76.8

If you are only interested in, for example, annual timeseries, you can specify argument `frequencies = "Y"`. It is also possible to specify the minimum year (argument `min_year`). More details are provided in the description of function `get_ts`.

By default, `get_ts` stores all data downloaded from the CBS in directory `raw_cbs_data`. The name of the directory can be specified with argument `raw_cbs_dir`. If you run `get_ts` a second time, you may notice that the result is much faster, because now the data in directory `raw_cbs_data` is read:

```
> data <- get_ts("70076NED", ts_code)
```

By default, `get_ts` only downloads data from the CBS if there is no data in directory `raw_cbs_data`, or if you have selected a new key in a dimension for the timeseries coding. This behaviour can be changed by specifying argument `refresh = TRUE`. In that case data are always downloaded from the CBS website, so that the data returned by `get_ts` are always the actual values of the CBS data. There is also argument `download`: if `FALSE`, `get_ts` will never download data. For details, check the documentation of function `get_ts`.