Explainer #1a: Technical Appendix

Olivier Simard-Casanova*

May 24, 2024

Abstract

The technical appendix documents how I identified structural breaks in my newsletter article Explainer #1a. The newsletter article is available in English[1] and in French[2].

*Independent Economist, Statistician, and Writer. www.simardcasanova.net. Contact: www.simardcasanova.net/contact/.

[1] o.simardcasanova.net/explainer-1a/.

[2] olivier.simardcasanova.net/decryptage-1a/.

Contents

1 Preliminary Notes **3**

2 BEAST **3**

3 Common Processing **3**

4 Figure N2 **4**
4.1 Figure 4
4.2 Data 5
4.3 Statistical Processing 6
4.4 Results 6

5 Figure N4 **9**
5.1 Figure 9
5.2 Data 9
5.3 Statistical Processing 11
5.4 Results 11

6 Figure N5 **13**
6.1 Figure 13
6.2 Data 13
6.3 Statistical Processing 15
6.4 Results 15

7 Figure N7 **17**
7.1 Figure 17
7.2 Data 17
7.3 Statistical Processing 19
7.4 Results 19

Bibliography **21**

1 Preliminary Notes

Figures with numbers beginning with "N" refer to figures in the newsletter article. Figures with a single number refer to figures in the current document.

2 BEAST

BEAST (Bayesian Estimator of Abrupt change, Seasonal change, and Trend) is a Bayesian estimator that decomposes the seasonal component $S(\cdot)$ and the trend $T(\cdot)$ of a time series:

$$y_i = S(t_i, \Theta_S) + T(t_i, \Theta_T) + \epsilon_i. \tag{1}$$

y_i is the value of the time series at period t_i, Θ_S and Θ_T are the seasonal variation signal and the trend signal, and ϵ_i is a noise assumed to be Gaussian.

As the data I use for this article is annual, there is no seasonal component. Equation 1 can be simplified as follow:

$$y_i = T(t_i, \Theta_T) + \epsilon_i. \tag{2}$$

The decomposition enables BEAST to detect structural breaks (i.e. changes in the trend) and to associate a probability of being a "true" structural break to each structural break candidate. BEAST is a meta-model, it detects possible structural breaks and computes the associated probability by averaging the results of a large number of linear submodels.

BEAST uses a Markov chain Monte Carlo method. As a result, BEAST gives slightly different results each time it runs on a given dataset. Structural break dates vary slightly, as do the probabilities. To overcome this randomness, a solution is to run BEAST many times and to aggregate the results of the runs. Given that the aggregation would have required a considerable amount of work, given that this is a newsletter article, and given that the variations in results seem to be very small, aggregating the results of multiple runs would have been an unreasonable effort. Instead, I randomly selected one of the run.

BEAST makes no assumptions on the shape of the data or on the number of actual structural breaks.

The estimator is presented in detail by Zhao et al. 2019.

I use the `Rbeast` package, which implements BEAST for R.

However powerful the estimator is, the quality and accuracy of the results always depend on the quantity and quality of the input data. Most of the data I use for this article is irregular and has many missing observations. Estimates of the "true" date of structural breaks are likely to be less precise than if the data was regular and had no missing observations.

3 Common Processing

The two main functions `Rbeast` provides are `beast()` and `beast.irreg()` (when the data is irregular).

`Rbeast` is able to estimate missing data, according to a time step the user can define with the `deltat` parameter. If `deltat` is not specified, `Rbeast` itself estimates a time step.

As the data I use is annual, there is no need to deseasonalize. This is why I set the `season` option to `"none"`.

I use a variety of conventions to facilitate my work with statistics and data. A guide is available on my website in English[3] and in French[4].

[3] o.simardcasanova.net/statistical-conventions/.

[4] olivier.simardcasanova.net/conventions-statistiques/.

4 Figure N2

4.1 Figure

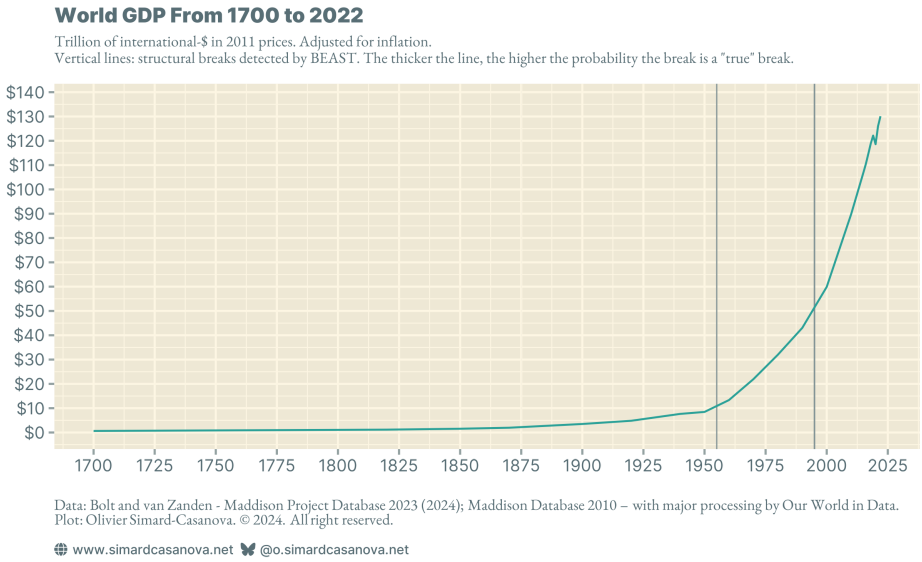

Figure 1: Figure N2.

4.2 Data

Raw data is shown in Table 1.

year	gdp
1	0.179
1000	0.205
1500	0.421
1600	0.561
1700	0.629
1820	1.175
1850	1.547
1870	1.963
1900	3.504
1920	4.825
1940	7.647
1950	8.462
1960	13.338
1970	21.942
1980	31.983
1990	43.034
2000	59.898
2010	89.808
2015	106.872
2016	110.407
2017	114.530
2018	118.766
2019	122.149
2020	118.590
2021	126.005
2022	130.113

Table 1: Raw data (d_2_raw).

4.3 Statistical Processing

The time series is irregular. There is one data point for year 1. Subsequent data points become more frequent as they approach 2022:

- every half a millennium from 1000
- every century from 1500
- every few decades from 1820 (sometimes two, sometimes three decades)
- every decade from 1940
- every half-decade from 2010
- every year from 2015

The time series is not complete, as there are many missing observations. Given the structure of the missing data (as shown in Table 1), I believe it is best to filter data from 1820 onwards, 1820 included, as the data becomes more frequent.

```
d_2 <- d_2_raw %>%
    filter(
        year >= 1820
    )
```

Listing 1: Code for data filtering.

I set a time step of 5 years to estimate the missing observations. A step of one year does not work, probably because the time step between each observation between 1820 and 1940 is too wide (i.e., there is not enough data).

```
beast.irreg(
    y = d_2$gdp,
    time = d_2$year,
    deltat = "5y",
    season = "none",
    start = min(d_2$year)
)
```

Listing 2: Code for the BEAST regression.

4.4 Results

Figure 2 shows the BEAST plot. Figure 3 shows the console output.

Results change with the date selected at the data filtering stage, and with the time step used to estimate the missing observations. The variability is probably not due to BEAST or `Rbeast`, but rather to the fact that the data is irregular and has many missing observations.

The oldest data is also not "real" GDP data, but ex-post reconstructions. By nature, ex-post reconstructions are more imprecise than "real" GDP data.

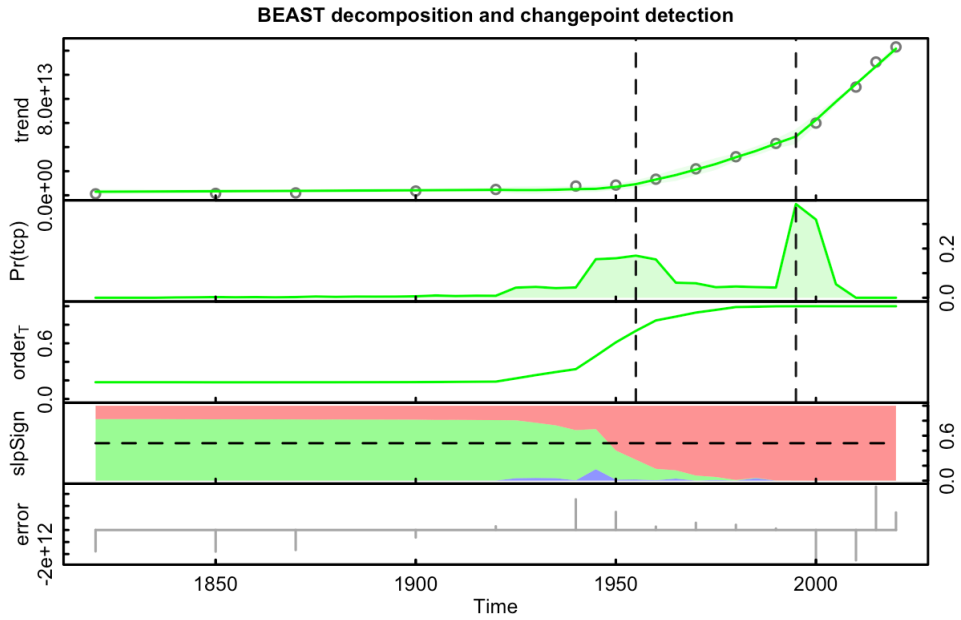

Figure 2: BEAST plot.

```
#####################################################################
#                     Seasonal  Changepoints                        #
#####################################################################
 No seasonal/periodic component present (i.e., season='none')


#####################################################################
#                     Trend  Changepoints                           #
#####################################################################
.-------------------------------------------------------------------.
| Ascii plot of probability distribution for number of chgpts (ncp) |
.-------------------------------------------------------------------.
|Pr(ncp = 0 )=0.000|*                                               |
|Pr(ncp = 1 )=0.158|**********                                      |
|Pr(ncp = 2 )=0.754|************************************************* |
|Pr(ncp = 3 )=0.086|******                                          |
|Pr(ncp = 4 )=0.003|*                                               |
|Pr(ncp = 5 )=0.000|*                                               |
|Pr(ncp = 6 )=0.000|*                                               |
|Pr(ncp = 7 )=0.000|*                                               |
|Pr(ncp = 8 )=0.000|*                                               |
|Pr(ncp = 9 )=0.000|*                                               |
.-------------------------------------------------------------------.
|    Summary for number of Trend ChangePoints (tcp)                 |
.-------------------------------------------------------------------.
|ncp_max    = 9    | MaxTrendKnotNum: A parameter you set           |
|ncp_mode   = 2    | Pr(ncp= 2)=0.75: There is a 75.4% probability  |
|                  | that the trend component has  2 changepoint(s).|
|ncp_mean   = 1.93 | Sum{ncp*Pr(ncp)} for ncp = 0,...,9             |
|ncp_pct10  = 1.00 | 10% percentile for number of changepoints      |
|ncp_median = 2.00 | 50% percentile: Median number of changepoints  |
|ncp_pct90  = 2.00 | 90% percentile for number of changepoints      |
.-------------------------------------------------------------------.
| List of probable trend changepoints ranked by probability of      |
| occurrence: Please combine the ncp reported above to determine    |
| which changepoints below are  practically meaningful              |
'-------------------------------------------------------------------'
|tcp#              |time (cp)                  |prob(cpPr)          |
|------------------|---------------------------|--------------------|
|1                 |1995.000000                |0.79538             |
|2                 |1955.000000                |0.64404             |
|3                 |1875.000000                |0.01800             |
.-------------------------------------------------------------------.


NOTE: the beast output object 'o' is a LIST. Type 'str(o)' to see all
the elements in it. Or use 'plot(o)' or 'plot(o,interactive=TRUE)' to
plot the model output.
```

Figure 3: Console output.

5 Figure N4

5.1 Figure

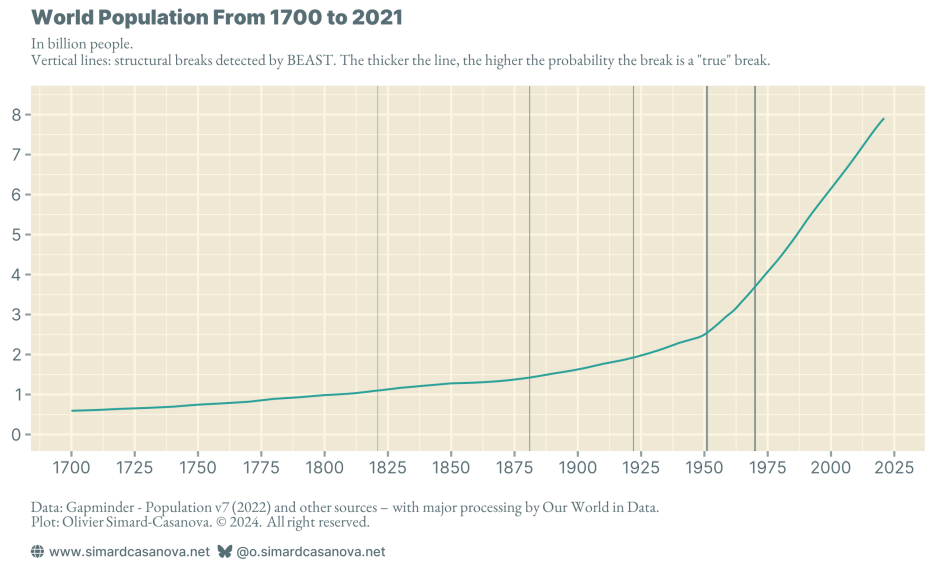

Figure 4: Figure N4.

5.2 Data

A truncated view of the raw data is shown in Table 2. After 1805, data stays annual.

year	population
-10000	4432266
-9000	5616996
-8000	7242892
-7000	9577918
-6000	13201832
-5000	19075771
-4000	28774661
-3000	44487162
-2000	72586076
-1000	110419521
0	232123764
100	236904172
200	240611571
300	227549847
400	241539639
500	253237316
600	271478693
700	278185334
800	285713840
900	310967715
1000	323407571
1100	397923813
1200	444750686
1300	456388961
1400	442480612
1500	503240399
1600	516610918
1700	592706212
1710	613189457
1720	642718111
1730	664743383
1740	695346690
1750	745664133
1760	779891857
1770	818827900
1780	891048681
1790	931577684
1800	985340629
1801	987628322
1802	991098738
1803	994610260
1804	998163240
1805	1001758044

Table 2: Truncated view of the raw data (d_4_raw).

5.3 Statistical Processing

Data frequency increases from 1700 onwards, from one observation per century to one per decade. From 1800 onwards, data become annual. I filter data from 1700 onwards, 1700 included.

```
d_4 <- d_4_raw %>%
    filter(
        year >= 1700
    )
```

Listing 3: Code for data filtering.

I use a time step of one year to estimate the missing observations.

```
beast.irreg(
    y = d_4$population,
    time = d_4$year,
    deltat = "1y",
    season = "none",
    start = min(d_4$year)
)
```

Listing 4: Code for the BEAST regression.

5.4 Results

Figure 5 shows the BEAST plot. Figure 6 shows the console output.

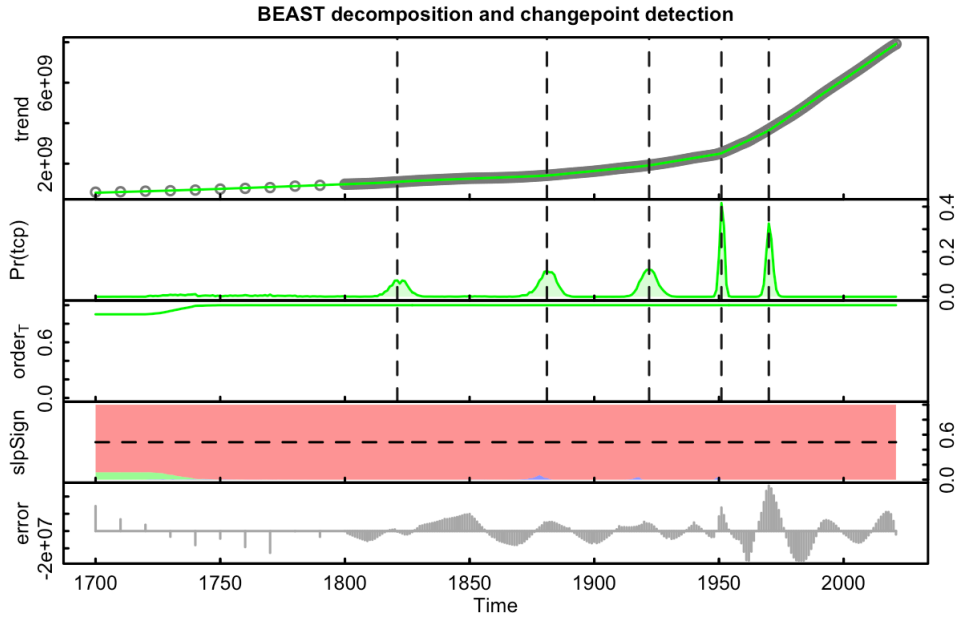

Figure 5: BEAST plot.

```
#####################################################################
#                    Seasonal  Changepoints                         #
#####################################################################
 No seasonal/periodic component present (i.e., season='none')


#####################################################################
#                    Trend  Changepoints                            #
#####################################################################
.--------------------------------------------------------------------.
| Ascii plot of probability distribution for number of chgpts (ncp) |
.--------------------------------------------------------------------.
|Pr(ncp = 0 )=0.000|*                                                |
|Pr(ncp = 1 )=0.000|*                                                |
|Pr(ncp = 2 )=0.000|*                                                |
|Pr(ncp = 3 )=0.000|*                                                |
|Pr(ncp = 4 )=0.011|*                                                |
|Pr(ncp = 5 )=0.946|************************************************ |
|Pr(ncp = 6 )=0.043|***                                              |
|Pr(ncp = 7 )=0.001|*                                                |
|Pr(ncp = 8 )=0.000|*                                                |
|Pr(ncp = 9 )=0.000|*                                                |
|Pr(ncp = 10)=0.000|*                                                |
.--------------------------------------------------------------------.
|    Summary for number of Trend ChangePoints (tcp)                  |
.--------------------------------------------------------------------.
|ncp_max    = 10   | MaxTrendKnotNum: A parameter you set           |
|ncp_mode   = 5    | Pr(ncp= 5)=0.95: There is a 94.6% probability  |
|                  | that the trend component has  5 changepoint(s).|
|ncp_mean   = 5.03 | Sum{ncp*Pr(ncp)} for ncp = 0,...,10            |
|ncp_pct10  = 5.00 | 10% percentile for number of changepoints      |
|ncp_median = 5.00 | 50% percentile: Median number of changepoints  |
|ncp_pct90  = 5.00 | 90% percentile for number of changepoints      |
.--------------------------------------------------------------------.
| List of probable trend changepoints ranked by probability of       |
| occurrence: Please combine the ncp reported above to determine     |
| which changepoints below are  practically meaningful               |
'--------------------------------------------------------------------'
|tcp#              |time (cp)                  |prob(cpPr)           |
|------------------|---------------------------|---------------------|
|1                 |1951.000000                |0.96563              |
|2                 |1970.000000                |0.89333              |
|3                 |1922.000000                |0.46313              |
|4                 |1881.000000                |0.44196              |
|5                 |1821.000000                |0.27487              |
|6                 |1740.000000                |0.04108              |
|7                 |1770.000000                |0.02588              |
|8                 |1760.000000                |0.02329              |
|9                 |1747.000000                |0.02158              |
|10                |1754.000000                |0.01908              |
.--------------------------------------------------------------------.


NOTE: the beast output object 'o' is a LIST. Type 'str(o)' to see all
the elements in it. Or use 'plot(o)' or 'plot(o,interactive=TRUE)' to
plot the model output.
```

Figure 6: Console output.

6 Figure N5

6.1 Figure

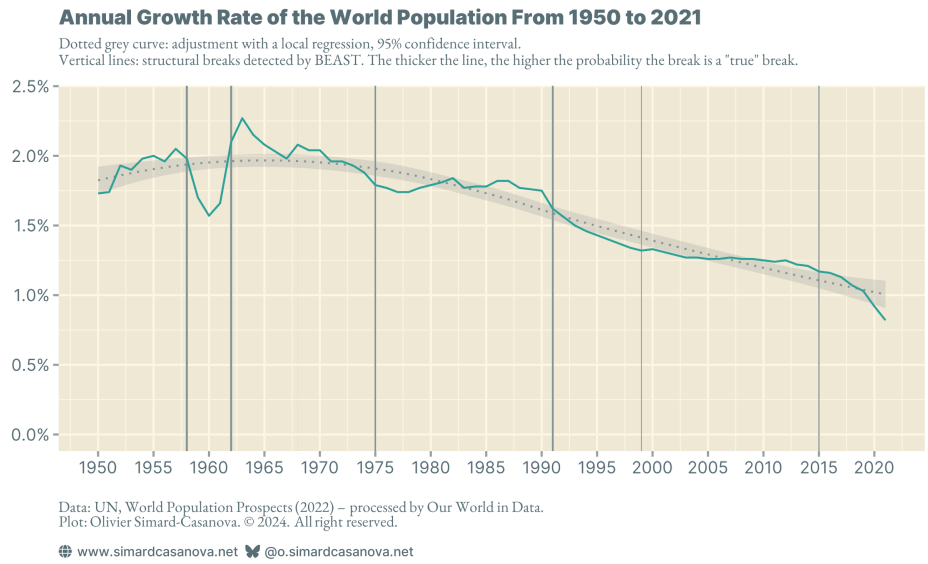

Figure 7: Figure N5.

6.2 Data

A truncated view of the raw data is shown in Table 3. Data is annual over the full time series.

year	growth_rate
1950	1.73
1951	1.74
1952	1.93
1953	1.90
1954	1.98
1955	2.00
1956	1.96
1957	2.05
1958	1.98
1959	1.70
1960	1.57
1961	1.66
1962	2.10
1963	2.27
1964	2.15
1965	2.08
1966	2.03
1967	1.98
1968	2.08
1969	2.04
1970	2.04

Table 3: Truncated view of the raw data (d_5_raw).

6.3 Statistical Processing

As the series is complete and regular, there is no need to filter the data.

```
d_5 <- d_5_raw
```

Listing 5: Code for data filtering.

It is also not necessary to estimate missing observations.

```
beast(
    y = d_5$growth_rate,
    time = d_5$year,
    season = "none",
    start = min(d_5$year)
)
```

Listing 6: Code for the BEAST regression.

6.4 Results

Figure 8 shows the BEAST plot. Figure 9 shows the console output.

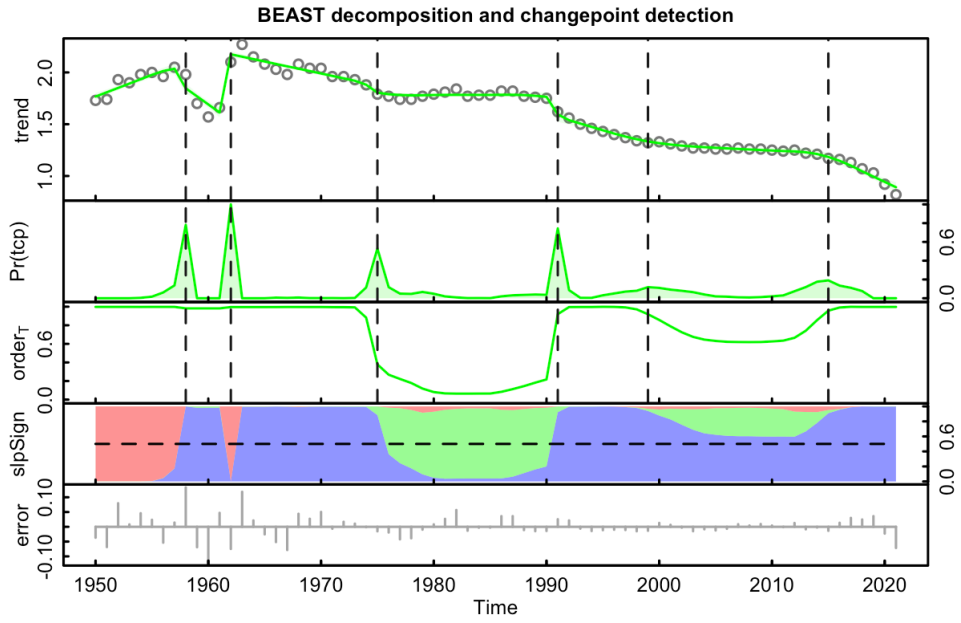

Figure 8: BEAST plot.

```
#####################################################################
#                     Seasonal  Changepoints                         #
#####################################################################
 No seasonal/periodic component present (i.e., season='none')


#####################################################################
#                     Trend  Changepoints                           #
#####################################################################
.-------------------------------------------------------------------.
| Ascii plot of probability distribution for number of chgpts (ncp) |
.-------------------------------------------------------------------.
|Pr(ncp = 0 )=0.000|*                                               |
|Pr(ncp = 1 )=0.000|*                                               |
|Pr(ncp = 2 )=0.003|*                                               |
|Pr(ncp = 3 )=0.001|*                                               |
|Pr(ncp = 4 )=0.040|***                                             |
|Pr(ncp = 5 )=0.173|***********                                     |
|Pr(ncp = 6 )=0.745|************************************************* |
|Pr(ncp = 7 )=0.038|***                                             |
|Pr(ncp = 8 )=0.001|*                                               |
|Pr(ncp = 9 )=0.000|*                                               |
|Pr(ncp = 10)=0.000|*                                               |
.-------------------------------------------------------------------.
|    Summary for number of Trend ChangePoints (tcp)                 |
.-------------------------------------------------------------------.
|ncp_max    = 10   | MaxTrendKnotNum: A parameter you set           |
|ncp_mode   = 6    | Pr(ncp= 6)=0.75: There is a 74.5% probability  |
|                  | that the trend component has  6 changepoint(s).|
|ncp_mean   = 5.78 | Sum{ncp*Pr(ncp)} for ncp = 0,...,10            |
|ncp_pct10  = 5.00 | 10% percentile for number of changepoints      |
|ncp_median = 6.00 | 50% percentile: Median number of changepoints  |
|ncp_pct90  = 6.00 | 90% percentile for number of changepoints      |
.-------------------------------------------------------------------.
| List of probable trend changepoints ranked by probability of      |
| occurrence: Please combine the ncp reported above to determine    |
| which changepoints below are  practically meaningful              |
'-------------------------------------------------------------------'
|tcp#              |time (cp)                 |prob(cpPr)           |
|------------------|--------------------------|---------------------|
|1                 |1962.000000               |0.99987              |
|2                 |1958.000000               |0.99738              |
|3                 |1991.000000               |0.90646              |
|4                 |1975.000000               |0.81112              |
|5                 |2015.000000               |0.61988              |
|6                 |1999.000000               |0.40004              |
|7                 |1968.000000               |0.01958              |
.-------------------------------------------------------------------.


NOTE: the beast output object 'o' is a LIST. Type 'str(o)' to see all
the elements in it. Or use 'plot(o)' or 'plot(o,interactive=TRUE)' to
plot the model output.
```

Figure 9: Console output.

7 Figure N7

7.1 Figure

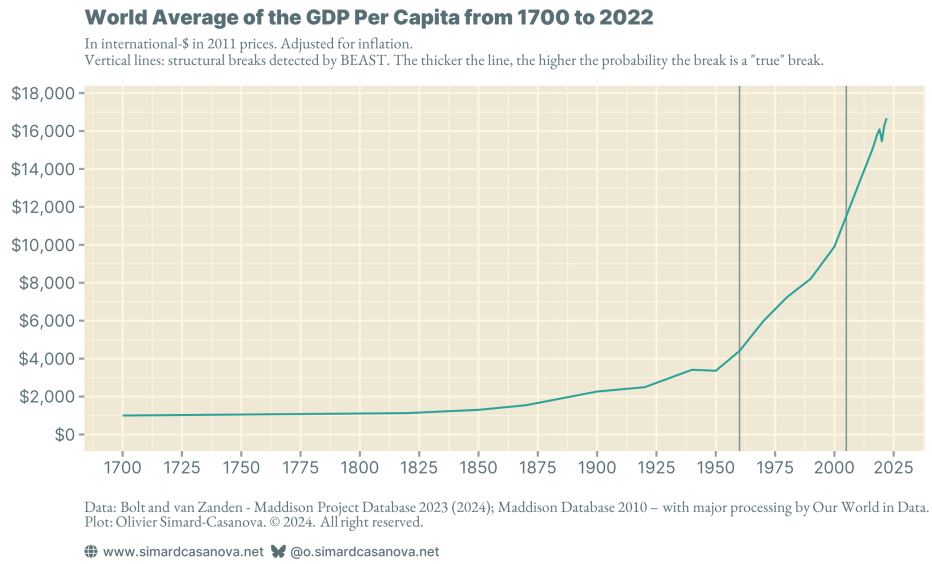

Figure 10: Figure N7.

7.2 Data

Raw data is shown in Table 4.

year	gdp_per_capita
1	800.000
1000	800.000
1500	1000.000
1600	1000.000
1700	1000.000
1820	1127.730
1850	1300.502
1870	1543.227
1900	2265.469
1920	2493.315
1940	3413.199
1950	3360.164
1960	4403.654
1970	5970.658
1980	7239.194
1990	8210.969
2000	9903.860
2010	13121.591
2015	14713.451
2016	15029.054
2017	15416.331
2018	15814.849
2019	16091.449
2020	15461.187
2021	16282.981
2022	16676.750

Table 4: Raw data (d_7_raw).

7.3 Statistical Processing

The data has the same structure as the data of Section 4. I use the same filtering and time step settings.

```
d_7 <- d_7_raw %>%
    filter(
        year >= 1820
    )
```

Listing 7: Code for data filtering.

```
beast.irreg(
    y = d_7$gdp_per_capita,
    time = d_7$year,
    deltat = "5y",
    season = "none",
    start = min(d_7$year)
)
```

Listing 8: Code for the BEAST regression.

7.4 Results

Figure 11 shows the BEAST plot. Figure 12 shows the console output.

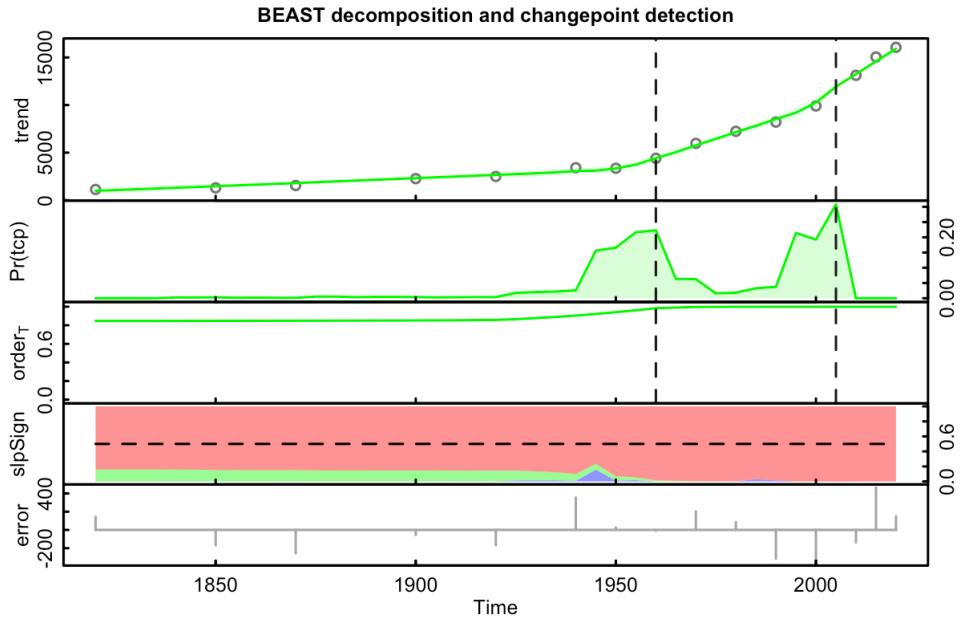

Figure 11: BEAST plot.

```
#####################################################################
#                     Seasonal  Changepoints                        #
#####################################################################
 No seasonal/periodic component present (i.e., season='none')


#####################################################################
#                     Trend  Changepoints                           #
#####################################################################
.-------------------------------------------------------------------.
| Ascii plot of probability distribution for number of chgpts (ncp) |
.-------------------------------------------------------------------.
|Pr(ncp = 0 )=0.000|*                                               |
|Pr(ncp = 1 )=0.197|************                                    |
|Pr(ncp = 2 )=0.760|************************************************ |
|Pr(ncp = 3 )=0.039|***                                             |
|Pr(ncp = 4 )=0.004|*                                               |
|Pr(ncp = 5 )=0.000|*                                               |
|Pr(ncp = 6 )=0.000|*                                               |
|Pr(ncp = 7 )=0.000|*                                               |
|Pr(ncp = 8 )=0.000|*                                               |
|Pr(ncp = 9 )=0.000|*                                               |
.-------------------------------------------------------------------.
|    Summary for number of Trend ChangePoints (tcp)                 |
.-------------------------------------------------------------------.
|ncp_max    = 9    | MaxTrendKnotNum: A parameter you set           |
|ncp_mode   = 2    | Pr(ncp= 2)=0.76: There is a 76.0% probability  |
|                  | that the trend component has  2 changepoint(s).|
|ncp_mean   = 1.85 | Sum{ncp*Pr(ncp)} for ncp = 0,...,9             |
|ncp_pct10  = 1.00 | 10% percentile for number of changepoints      |
|ncp_median = 2.00 | 50% percentile: Median number of changepoints  |
|ncp_pct90  = 2.00 | 90% percentile for number of changepoints      |
.-------------------------------------------------------------------.
| List of probable trend changepoints ranked by probability of      |
| occurrence: Please combine the ncp reported above to determine    |
| which changepoints below are  practically meaningful              |
'-------------------------------------------------------------------'
|tcp#              |time (cp)                  |prob(cpPr)          |
|------------------|---------------------------|--------------------|
|1                 |1960.000000                |0.76138             |
|2                 |2005.000000                |0.75221             |
|3                 |1880.000000                |0.01925             |
.-------------------------------------------------------------------.


NOTE: the beast output object 'o' is a LIST. Type 'str(o)' to see all
the elements in it. Or use 'plot(o)' or 'plot(o,interactive=TRUE)' to
plot the model output.
```

Figure 12: Console output.

Bibliography

Scientific

Zhao, Kaiguang et al. (2019). “Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm”. In: *Remote Sensing of Environment* 232, p. 111181. DOI: 10.1016/j.rse.2019.04.034.