Too cheap to be true – Detecting invalid values in product prices and index values

Manuel Koller[1], Alexandra Schindlar[1]

[1]*Statistics Austria*

Abstract

Consumer price indices (CPIs) are key indicators for the monetary policy and hence reported to the central banks. The National Statistical Institutes provide each month the inflation rate, which is the outcome of the consumer price survey and hence the input for the monetary policy. Starting from the task to develop quantitative indicators for the CPI price survey the authors developed an application for validation of the price survey and the computation of the inflation rate with the aim to minimize errors in the price survey and embedded the work in the framework of quality assurance as theoretical basis. The quantitative indicators were developed within and classified to four different aims: The overview over the price observation per month and over time, the increase of the quality of the price survey, the detection and minimisation of errors and the improvement of the sample itself. The resulting application allows to monitor the validation process within the production of monthly results on the one hand. Moreover, the defined quantitative indicators are developed for and can be applied on different aggregation levels of the price and index data: The price observation itself, some transformations of the observed prices and on the resulting index numbers, as well as on monthly and annual rates of change and on contributions to the inflation rate. The application itself mimics the validation process that was done with less automatic support prior to its introduction. In the end, conclusions for the future and open issues are addressed as the application of time series models or the introduction of new variables into the price survey, which are needed to improve the development of quantitative indicators.

Keywords: CPI/HICP, Outlier detection, Medcouple, Time series models, Nearest neighbours

1. Introduction

Consumer price indices (CPIs) are one of the most well know metrics usually produced by National Statistical Institutes (NSI). The CPI and the derived inflation rate are key economic indicators used by central banks to set interest rates. The use of the CPI for monetary policy is the end result of a laborious monthly production process inside the NSIs that start with a comprehensive consumer price survey (Statistics Austria, 2022).

In the course of the monthly CPI calculation, similar but not identical tasks with typical risks and sources of errors arise each period and on each stage of the process. An example for such a task on raw data level could be the collection of the newest price data for wood pellet fuel and a corresponding typical risk for error could be erroneously noting down the price for 15 kilograms instead of 50. Data collection software may be able to mitigate this through as much information and feedback as possible (e.g. warnings triggered by threshold violations or other notifications), yet raw data cannot be granted as valid before detailed plausibility checks have been carried out. Despite of the rather repetitive character of this task, it has to be done carefully and in a timely manner to assure that only high quality price information is used in the price index calculation.

Contributors on various levels of the calculation process have therefore developed individual strategies for evaluating the data they are working with as well as the results of their processing steps. In order to reduce the monthly workload and minimize potential for errors as well as to standardise the plausibility check process as a whole, a set of functionalities to support people involved with quality control was developed and crafted into an interactive user application, easily accessible via a standard web browser. To facilitate a quick transition from using previous plausibility check strategies to working with the application, it was designed to mimic well established concepts and enrich them with additional meaningful indicators. These quantitative indicators were developed within and classified to four different aims: The overview of the price observation per month and over time, the increase of the quality of the price survey, the detection and minimisation of errors and the improvement of the sample itself.

One additional requirement for the application was to enable people, even with only limited or no IT (programming) skills at all, to carry out typical data analyses and visualisations quickly by only selecting the month in questions and a few additional setup parameters (like custom threshold values, base periods for the index series etc.).

2. Data sources for the calculation of the CPI and their particularities

For the calculation of the Austrian CPI there are currently three main data sources: Data collected locally in stores (traditional data), data collected by barcode scanners at checkout which are aggregated on a weekly basis and transferred to Statistics Austria (scanner data) and data collected online via web scraping scripts (web scraping data). Each of the sources has its specific requirements when it comes to plausibility checking (Statistics Austria, 2022).

For traditional data, the focus lies mostly on the product prices themselves as well as their development/consistency over time. Large changes or discontinuity often indicates incorrect data and therefore should be inspected in more detail. For scanner- and web scraping data, besides prices also assigned COICOP and lowest level aggregate (LLA)[1] classifications can be incorrect. Furthermore, the completeness of data collection/transmission is more complicated to evaluate since, compared to traditional price collection where there is a fixed market basket with a predefined amount of sample cases to be collected, all available/sold products shall be considered. Therefore, the number of products can vary (especially for seasonal products like winter/summer sports equipment) and can’t be checked against a constant number of expected products.

---

[1] The lowest level aggregate (LLA) is Statistic Austrias finest granularity CPI coding schema and is nested within COICOP level 5. For example, the COICOP group „chocolate“ consists of the LLA „milk chocolate“, „chocolate boxes“ and „chocolate bars“.

Due to the different nature of scanner and web scraping data compared to traditional data, for the former two plausibility checks on raw data level were outsourced to separate, specific data pipelines and will not be part of this paper. Raw data checks implemented in the app described in this paper are tailored towards locally collected data. Checks in later calculation stages (LLA and above, when all 3 sources are merged already) include scanner data as well as web scraping data.

3. Defining indicators to detect incorrect values

Detecting outliers is an important topic in many statistical fields. There are various approaches to efficiently search for incorrect values with each of them coming with its own advantages and disadvantages (Smiti, 2020).

In price statistics identification and cleaning or removing of outliers is an important topic as well. There are a few additional specificities which can have an effect on product prices and therefore also on the assessment of what is or is not considered an incorrect value. For example, seasonality, changes in product quality or product replacements might explain large price changes which would otherwise, in the absence of such specificities, be flagged as suspicious. Another challenge of price data lies in different price development patterns for different product classes. Some products (e.g. fees for public transport) usually show only one price change per year where as others change on a daily basis (e.g. gas). So, for some products a price change might indicate problems (if happened in an unexpected period) whereas for others no price changes might be suspicious.

To account for the particularities of price data a set of different outlier detection methods was implemented.

Outlier detection …

- … based on rates of change, used both on raw price data and LLA.
- … based on comparison of current index values for similar product groups, defined by similar index developments in the past (experimental).
- … based on time series model forecasts for separate index series (experimental).

The later 2 approaches are considered experimental at this point, since final decisions about best hyperparameter selection are not made yet. Currently it is up to the user to experiment with individual parameter settings. We hope that experience from product experts using the functions in combination with theoretical considerations will help to make a final decision regarding the best setup parameters.

Beside products flagged as suspicious by one or more of the three outlier detection methods, also products belonging to high impact LLAs, are re-evaluated again in any case before the CPI is released. High impact LLAs are defined as product groups having a high influence on the inflation rate of the current period or explaining a high portion of the changes in inflation rates between the current and the previous period.

3.1 Outlier detection based on monthly and annual rates of change

Even within the same product category, price variations can be large and even if single product prices are considered as statistical outliers, they might indeed be correct prices of very exclusive brands. Excluding products on the basis of high prices alone seems unjustified and in the case of traditional price collection deciding if these products are indeed relevant for the market is difficult, since there is a lack of information regarding revenue or number of sales. For this reason, we focus on monthly and annual rates of change in product prices instead of product prices themselves. Current rates of change of product prices can be evaluated either against previous rates of change within the same product or against current rates of change of other (in best case similar) products. In a first step the distribution of all rates of change used for comparison is estimated. Typically, these are not normally distributed so a Medcouple adjustment of the interquartile range (IQR) is applied to account for the skewness (Brys, Hubert, & Struyf, 2004). If the current rate of change lays more than 1.5 times the adjusted IQR below Q1 or 1.5 times the adjusted IQR above Q3 the current value gets flagged as potential error (Hubert & Vandervieren, 2008).

From our experience comparing the current monthly rate of change against previous ones within the same product/LLA yields the best results. For this approach a data series that extends long enough into the past is needed (currently we are using 36 months as default and 24 months to cover product which were not available in the market basket 36 months ago). Newly introduced products and LLA cannot be evaluated this way and must be checked otherwise.

Comparing the current rate of change against a distribution based on current rates of change of similar products/LLA (defined by COICOP-hierarchy) did not achieve satisfactory results. The reason therefore might be, that even within similar product categories price and index developments can vary[2]. An alternative approach for using information from fewer but closer similar LLA will be described in the following section 3.2.

---

[2] In order to estimate a distribution, a corresponding amount of measured values is required. Therefore, for this approach, the group of similar cases must be defined rather broadly, which can be an explanation for the high variability within the group of “similar” products. For some LLA index series, this means that all values of the LLA nested in COICOP level 2 must be considered for comparison, so true similarity of all considered index series cannot be seen as granted.

3.2 Outlier detection based on comparison of similar LLA (experimental)

Instead of comparing the current monthly rate of change against a distribution based on many similar LLAs as described under 1, this approach tries to find a limited amount of (very) close neighbours using the KNN algorithm (Beygelzimer et al., 2023) to compare the current rate of change to (Dang, Ngan, & Liu, 2015). The similarity between LLAs is estimated using data from the past 12 to 36 months, excluding the current one. Univariate similarity evaluation can be based on monthly or annually rates of change or on the index value itself. So far there was no clear advantage for preferring one particular variable. Therefore, all three possibilities are available in the application and can be picked by the user at runtime. Normalising the values within each index series before calculating similarity values helps finding similar neighbours with comparable index movement but different amplitude in change. As a similarity/distance measure simple Euclidian distance is used, but there might be room for improvement by using more specific distance measurements.

In case the value of an LLA lies above or below all its neighbours' value it gets flagged as critical. In addition to the flag a parameter, indicating the amount of deviation from the neighbours is presented. It is calculated by dividing the difference between the max/min value within the neighbours and the current LLA's value by the range within the neighbours, so that high values represent large deviations. Just as with the distance function itself we think that redefining a more specific calculation method can increase the methods precision.

Since there are no final decisions about the best suited distance function as well as which variables and normalisation methods should be recommended to the application user, this approach is still considered as experimental.

3.3 Outlier detection based on time series model forecasts (experimental)

Previously described approaches for outlier detection did not emphasise seasonality in particular. For some product groups with strong seasonal pattern (e.g. clothing) it could be a missed opportunity to not use this information. Therefore, the last approach emphasises on recognising seasonal patterns, if present, and using them for outlier detection by forecasting theoretically expected values for the current period (Isbister, 2015).

ARIMA forecasting was implemented using the auto.arima function from the forecast package (Hyndman & Khandakar, 2008). It automatically detects the best fitting model specification, including seasonality, for the supplied time series data which is very helpful for this task, since otherwise tailoring specific models for each product group by hand would be a time-consuming task. True current index values that lie outside the 95% confidence interval for the predicted value are flagged and need to be evaluated, values outside the 80% confidence interval will lead to a warning and should be checked if there is enough capacity left. This approach is marked experimental since there is no final decision about judging if an automatically fitted ARIMA model yields a good enough fit to the data to consider it as reliable enough for outlier detection. Some index series just do not follow patterns that could be captured using time series models. For these cases it is very likely to get false positive critical flags which results in unjustified manual evaluation work for product experts.

4. Discussion

Providing an easy to access application to apply previously described methods to the data in an easy and quick manner has proven valuable so far. Especially outlier detection based on monthly rates of change (3.1) is an easy to understand and therefore well accepted approach for finding questionable price and index values. Feedback from product experts provided the information, that comparisons based on monthly rates of change yield better results than those on annual rates of change. Product prices and index values having a larger monthly rate of change in the current period than in the defined evaluation periods are checked in any case. In most cases values turn out to be correct, but since the evaluation using the application is quick we think not further restricting outlier flagging and therefore accepting a rather large amount of false positive hits to reliably find the few true incorrect outliers is justified.

One major drawback of using the monthly rate of change as described in 3.1 is, that seasonality is not considered at all. Seasonally expected large changes which are missing in a period will not be detected by this approach and large changes occurring in a typical seasonal period will still get flagged as outliers.

Focusing on comparisons based on similar LLA indices, accounts for seasonal patterns implicitly and breaks of these patterns will be flagged as long as they still occur in some neighbouring LLA. One big advantage of this method, compared to both other ones, is that regular but not strict periodical cyclic patterns can be evaluated. Governmental fees for example might be increased once a year but at different time periods for each year. Travel expenditures like flight tickets or hotel prices typically increase around Easter holidays. The nearest neighbours approach will flag those codes not showing this increase as long as it was recorded in some of the neighbouring LLAs. The downside is, that false values in a whole group of similar LLA (for example, if said governmental fee increase was overlooked in all LLAs) will not be detected since all the neighbours show the same incorrect pattern as the target itself. In addition, it is difficult to define cut-off values to determine if the nearest neighbours found are indeed similar enough to the target to be considered reliable. Currently, we are using only the closest 10% of neighbouring LLAs found (for details see appendix 1.6).

This results in LLA neighbour groups which mostly make sense in terms of content, but still results in a lot of false positive outlier flags.

The ARIMA approach is very well suited for typical seasonal LLA like clothing or fruits and vegetables. Breaks in seasonal patterns or trends are caught reliable by the used auto ARIMA model. Just as with the nearest neighbour approach, again a difficult task is to decide if a model fits well enough to the data and thus can be considered reliable for outlier detection. Currently, we are using the average monthly residuals to display the accuracy of the model since we think it is an easy to understand measurement and therefore should be well accepted by the users. Also using the confidence intervals to decide weather a value gets flagged or not helps to adjust the strictness depending on the fit of the model. The downside of using time series models for outlier detection is, that some LLAs don't follow a timeseries pattern at all. They might be dependent on other indices and follow them with a certain lag or on external factors fluctuating more or less randomly. In best case models applied to such data will yield a bad fit, in worst case some false patterns are recognised. More complex methods like multivariate recurrent neural networks (Su et al., 2019) might be a major improvement regarding this problem but where not tested by us yet.

In general, for all approaches used, we think that high sensitivity is more important than high precision. Avoiding errors is an important task, especially since revising the CPI is a time and resource intensive process. Therefore, we are willing to take a higher amount of false positive hits and try to compensate by speeding up the evaluation process of flagged cases, rather than risking overlooking some true errors. In addition, it has to be stated, that true errors are rather infrequent in daily business, since our colleagues at data collection are doing a very good job already. For the quality of the CPI this is very positive, for developing outlier detection methods it complicates the task since there is just a small sample of true error cases to work with.

One important additional consideration concerns the time period when COVID had a large impact on the market. Usually using longer timelines for evaluation should result in more stable outcomes, but since COVID times were marked by frequent price imputations and in general atypical price movements we think it is better to not use these periods for evaluating current price changes (especially for seasonal data, since the seasonal pattern were less present or missing at all during COVID crisis).

5. Outlook

The application is not considered final but as an evolving system which grows and adapts to new requirements. Critical milestones for improving currently available features will be selecting reliable setup parameters both for the experimental nearest neighbour and the ARIMA approaches. Therefore, feedback from our product experts is collected and analysed.

In general, user suggestions on reworking or adding features are highly appreciated and get adopted. One possible additional feature could be an assistance system for users to quickly send mail notifications including short summaries for questionable products/LLA to product experts.

Acknowledgment

We want to thank all our colleagues, especially product experts, for sharing their knowledge and providing helpful feedback while developing, testing and using the app. We also want to thank EUROSTAT for supporting the development of the app with a grant.

References

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., & Li, S. (2023). *FNN: Fast nearest neighbor search algorithms and applications. R package version 1.1.3.2.* https://CRAN.R-project.org/package=FNN.

Brys, G., Hubert, M., & Struyf, A. (2004). A Robust Measure of Skewness. *Journal of Computational and Graphical Statistics, 13*(4), 996–1017. https://doi.org/10.1198/106186004X12632

Chang W, Borges Ribeiro B (2021). *shinydashboard: Create Dashboards with 'Shiny'.* R package version 0.7.2. https://CRAN.R-project.org/package=shinydashboard

Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2024). *shiny: Web Application Framework for R*. R package version 1.8.1.1. https://github.com/rstudio/shiny, https://shiny.posit.co/

Dang, T. T., Ngan, H. Y., & Liu, W. (2015, July). Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. In *2015 IEEE International Conference on Digital Signal Processing (DSP)* (pp. 507-510). IEEE. https://doi.org/10.1109/ICDSP.2015.7251924

Hubert, M., & Vandervieren, E. (2008). An adjusted boxplot for skewed distributions. *Computational statistics & data analysis*, *52*(12), 5186-5201. https://doi.org/10.1016/j.csda.2007.11.008

Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, *27*(3), 1–22. https://doi.org/10.18637/jss.v027.i03

Isbister, T. (2015). Anomaly detection on social media using ARIMA models.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Smiti, A. (2020). A critical overview of outlier detection methods. *Computer Science Review*, *38*, 100306. https://doi.org/10.1016/j.cosrev.2020.100306

Statistics Austria (2022). *Standard-Dokumentation zum Verbraucherpreisindex und Harmonisierten Verbraucherpreisindex.* https://www.statistik.at/fileadmin/shared/QM/Standarddokumentationen/VW/std_v_vpi_hvpi.pdf

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019, July). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2828-2837). https://doi.org/10.1145/3292500.3330672

Appendix

1. Application Overview

The application was developed in R 4.3.3 (R Core Team, 2024) and relies on the packages shiny (Chang et al., 2024) and shinydashboard (Chang & Borges Ribeiro, 2021). Amongst other features, the app allows user to apply outlier detection methods and browse through index and product data flagged as critical. Data visualisations can be generated to quickly evaluate questionable data. In addition, outputs generated by the application can also be saved as .xlsx files.

For data protection reasons, the weblink to the application is only accessible within Statistics Austria IT infrastructure and is not publicly available. Unfortunately, as this is an internal tool, there is no English version of the application, so the example screenshots are all in German.

1.1 Index viewer tab

To quickly visualize index series the application provides the possibility to select various indices available in our database, load them and plot them as line charts or display them in tables. For easier navigation through the many index positions a hierarchical filter system for preselecting subgroups is provided. Different CPI concepts (e.g. the harmonised CPI or the CPI with constant tax rates) can be processed simultaneously and compared to each other.

Figure 1: Data selection panel for the index viewer tab. Users can select which index tables (national consumer price index, European harmonised consumer price index and/or European harmonised consumer price index with constant taxes) and timeframe to be loaded and displayed. The base year determines which time period is used as index base and is set to 100.

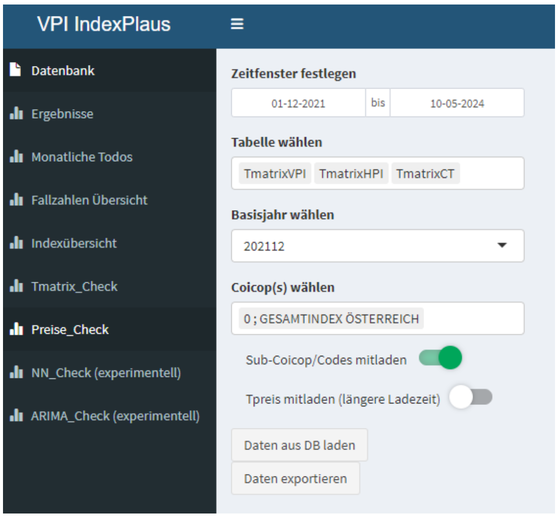
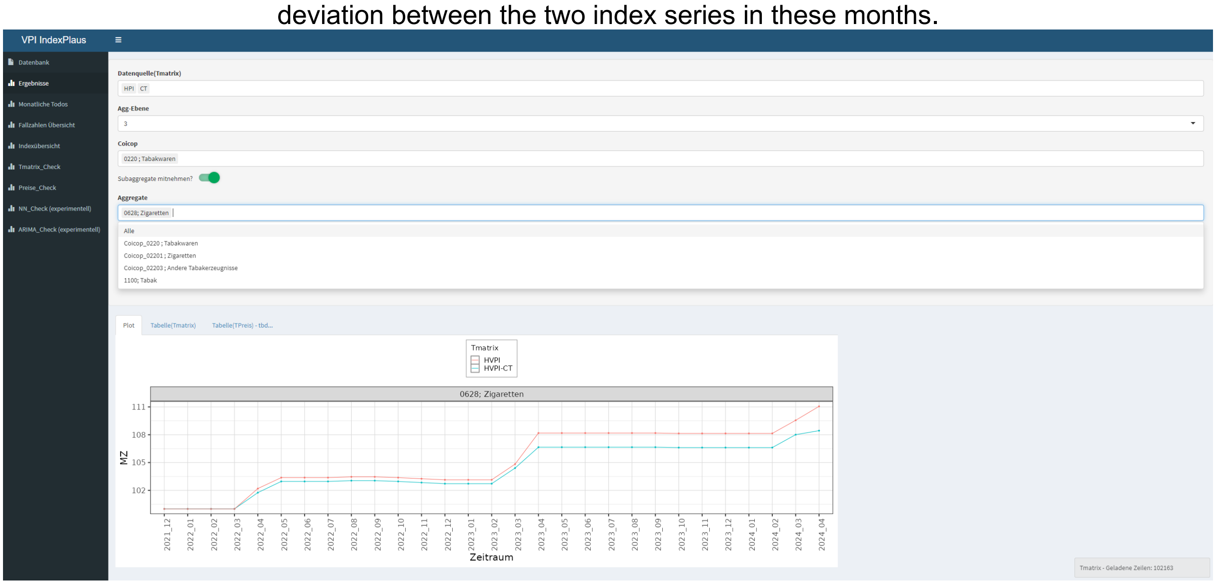


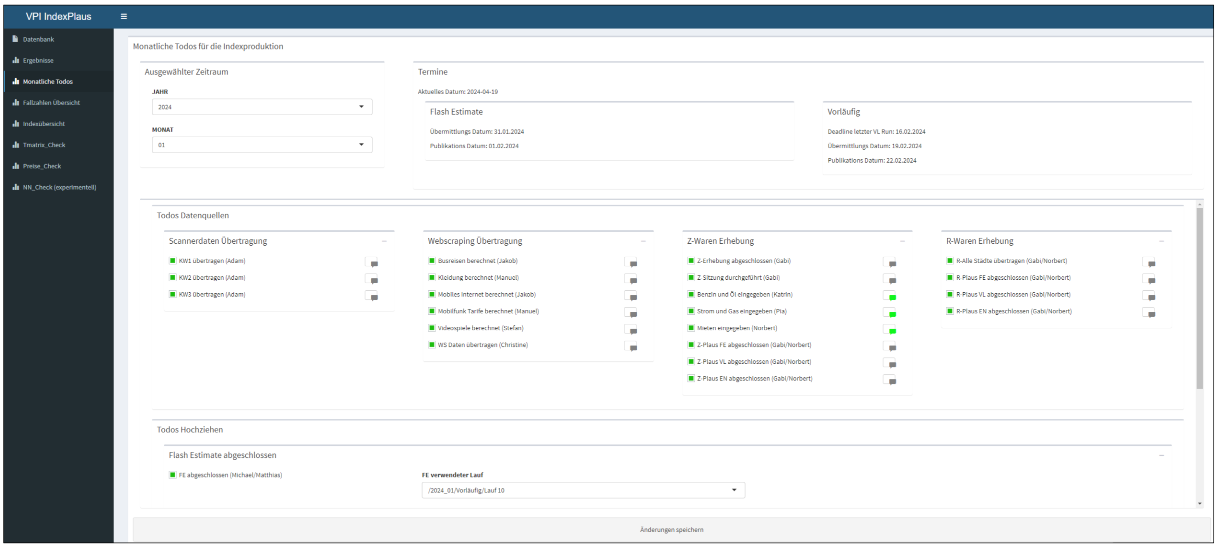
Figure 2: Index viewer example, showing the development of the harmonised index of consumer prices with constant tax rates compared to the normal harmonised index of consumer prices for COICOP group cigarettes. Usually in April taxes for tobacco are increased which can be seen by a deviation between the two index series in these months.



1.2 Progress tab

This module lists transmission deadlines and checkboxes for the single milestone tasks of the monthly CPI production process. As soon as tasks are finished they get marked as finished and follow up processes can start. There is also the possibility to note down some comments if necessary. For the final milestones there is also the option to note which calculation run was used for the flash estimate transmission and for the final results. This helps tracking changes between the two transmissions later on.

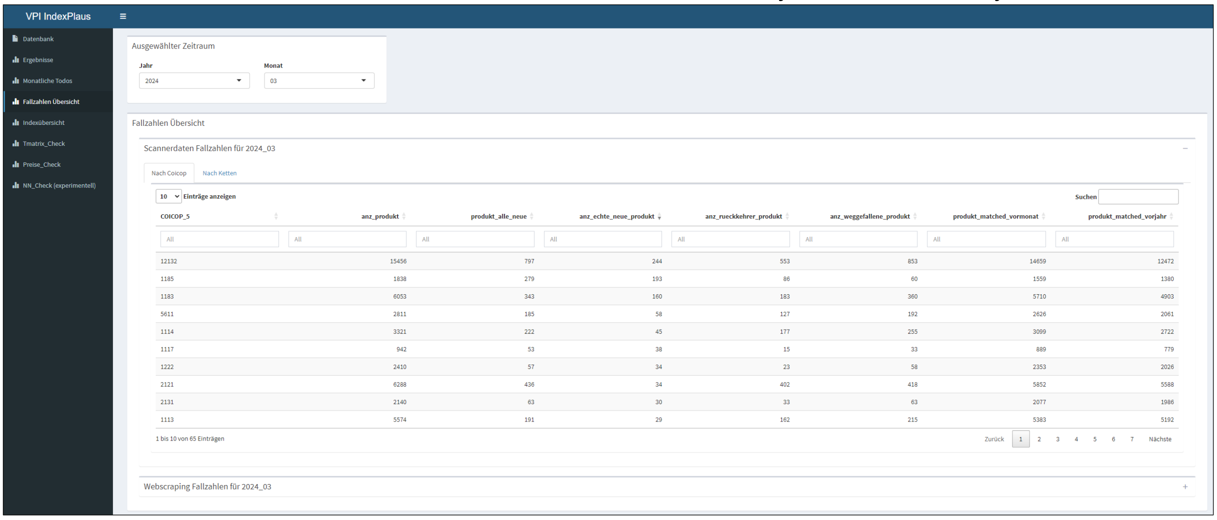
Figure 3: Example screenshot for a fully completed month. All milestones have been marked as done. Green message symbols indicate additional comments for some of the milestones



1.3 Overview for scanner and webscraping data tab

As previously mentioned plausibility checking for raw scanner and webscraping data is done in separate data pipelines. However, to provide information for people not working directly with raw scanner or webscraping data, but are involved in the aggregation and evaluation process, a case summary module was introduced. It gives a quick overview on how many products were used for calculation in total, how many of those products were new and how many previously consider products dropped out of the market in table or chart form.

Figure 4.1: Example screenshot of a case summary for scanner data by COICOP level 5. For example, for code 12132 in total there were 15,456 products available. 244 of them were new on the market, 12,472 have been available in the data 1 year before already.



COICOP_5	anz_produkt	produkt_alle_neue	anz_echte_neue_produkt	anz_rueckkehrer_produkt	anz_weggefallene_produkt	produkt_matched_vormonat	produkt_matched_vorjahr
12132	15456	797	244	553	853	14659	12472
1185	1838	279	193	86	60	1559	1380
1183	6053	343	160	183	360	5710	4903
5611	2811	185	58	127	192	2626	2061
1114	3321	222	45	177	255	3099	2722
1117	942	53	38	15	33	889	779
1222	2410	57	34	23	58	2353	2026
2121	6288	436	34	402	418	5852	5588
2131	2140	63	30	33	63	2077	1986
1113	5574	191	29	162	215	5383	5192

Figure 4.2: Clicking on a row in the table shown in Figure 3.1 opens a dialog window showing further details regarding the selected code. Case counts for the current month and the 12 previous months are displayed in table form as well as line charts. Strong changes in this numbers can indicate problems and should be examined further.

COICOP_5: 12132 von 2023_03 bis 2024_03

Fallzahlen

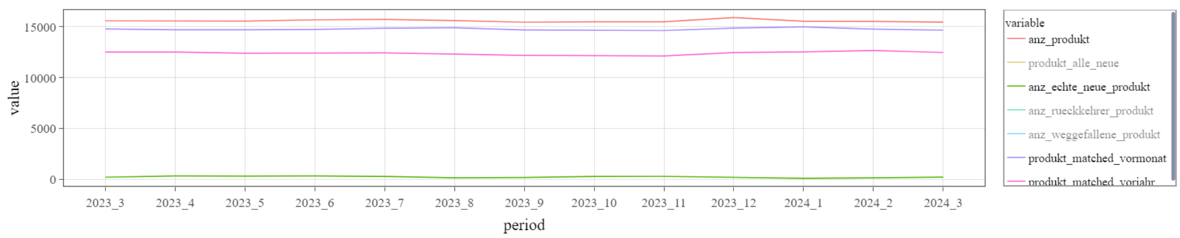
13 Einträge anzeigen

Suchen

jahr	monat	anz_produkt	produkt_alle_neue	anz_echte_neue_produkt	anz_rueckkehrer_produkt	anz_weggefallene_produkt	produkt_matched_vormonat	produkt_matched_vorjahr
All	All	All	All	All	All	All	All	All
2023	3	15576	798	233	565	862	14778	12519
2023	4	15582	887	356	531	881	14695	12513
2023	5	15553	850	343	507	879	14703	12392
2023	6	15676	944	361	583	821	14732	12407
2023	7	15718	863	307	556	821	14855	12438
2023	8	15597	703	167	536	824	14894	12316
2023	9	15441	758	204	554	914	14683	12183
2023	10	15480	824	308	516	785	14656	12162
2023	11	15479	853	314	539	854	14626	12127
2023	12	15897	1031	202	829	613	14866	12462
2024	1	15539	550	123	427	908	14989	12527
2024	2	15512	741	169	572	768	14771	12667
2024	3	15456	797	244	553	853	14659	12472

1 bis 13 von 13 Einträgen

Zurück 1 Nächste



Schließen

Figure 4.3: For COICOP group chocolate a seasonal pattern in the number of sold products can be seen. This is explained by special Christmas and Easter chocolate editions returning before and exiting the marked after the holidays. The red line indicates number of products in total, the green line shows products which have been available on the market in the past, but not in the previous month (the line peeks before Christmas and Easter) and the blue line shows products that have been available in the previous month but are dropped out in the current one (in May 2023 Easter products disappeared from the market, in February 2024 Christmas products where replaced by Easter products).

COICOP_5: 1183 von 2023_03 bis 2024_03

Fallzahlen

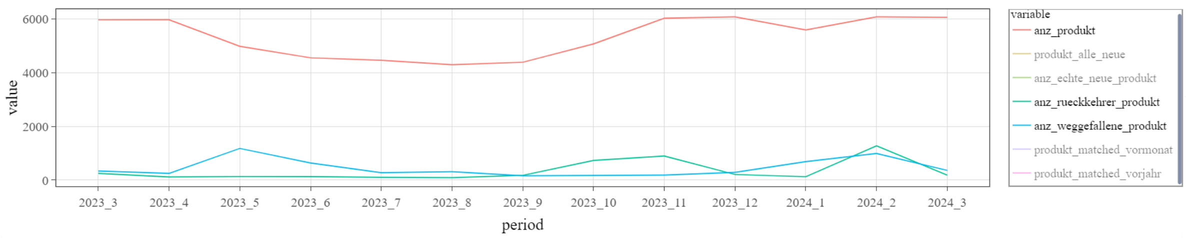
13 Einträge anzeigen

Suchen

jahr	monat	anz_produkt	produkt_alle_neue	anz_echte_neue_produkt	anz_rueckkehrer_produkt	anz_weggefallene_produkt	produkt_matched_vormonat	produkt_matched_vorjahr
All	All	All	All	All	All	All	All	All
2023	3	5961	515	267	248	337	5446	4793
2023	4	5963	249	131	118	247	5714	4808
2023	5	4974	188	56	132	1177	4786	3884
2023	6	4549	209	80	129	634	4340	3596
2023	7	4454	180	76	104	275	4274	3601
2023	8	4293	149	58	91	310	4144	3469
2023	9	4384	252	76	176	161	4132	3570
2023	10	5065	847	118	729	166	4218	4035
2023	11	6022	1141	248	893	184	4881	4797
2023	12	6070	336	127	209	288	5734	4972
2024	1	5584	198	72	126	684	5386	4471
2024	2	6070	1475	203	1272	989	4595	4953
2024	3	6053	343	160	183	360	5710	4903

1 bis 13 von 13 Einträgen

Zurück 1 Nächste



Schließen

1.4 Checks on raw data level tab

This module is supposed to be used mainly by product experts to get a quick overview over the data collected by price collectors. A summery table showing counts of collected prices in total, changed prices and processing flags set gives a short overview over the current data base. Single product data can be browsed either on raw price level, which does not include any quality adjustment effects, or on single product index value level, including all adjustments applied. Cases can be filtered by product group and sorted by “extremeness parameters” based on the outlier detection. The most frequently used indicator here is the monthly rate of change.

An on click feature on single products enables users to draw timeline charts for the current price/single product index value and its rate of change for the past 24 months.

Figure 5.1: Example screenshot for the results of the checks on raw single product level. The top two tables give an overview on recorded changes of prices and single product index values (left table) and over all the calculation flags set by the price collection team. In this example one can see, that the A-flag (which indicates special offer prices) was used less frequently in April than in the previous month but almost as often in the previous year’s April. This can be explained by end of winter season sales, especially for clothing, which typically occurs in March and is not available in April any longer.
The table below shows the rate of change for all single products. They can be either inspected on raw unprocessed price level (RPREIS tab) or on single product index level, including already applied quality adjustments (MZ tab). Users can filter for products flagged as critical and can sort by rate of change value to find the most extreme cases.

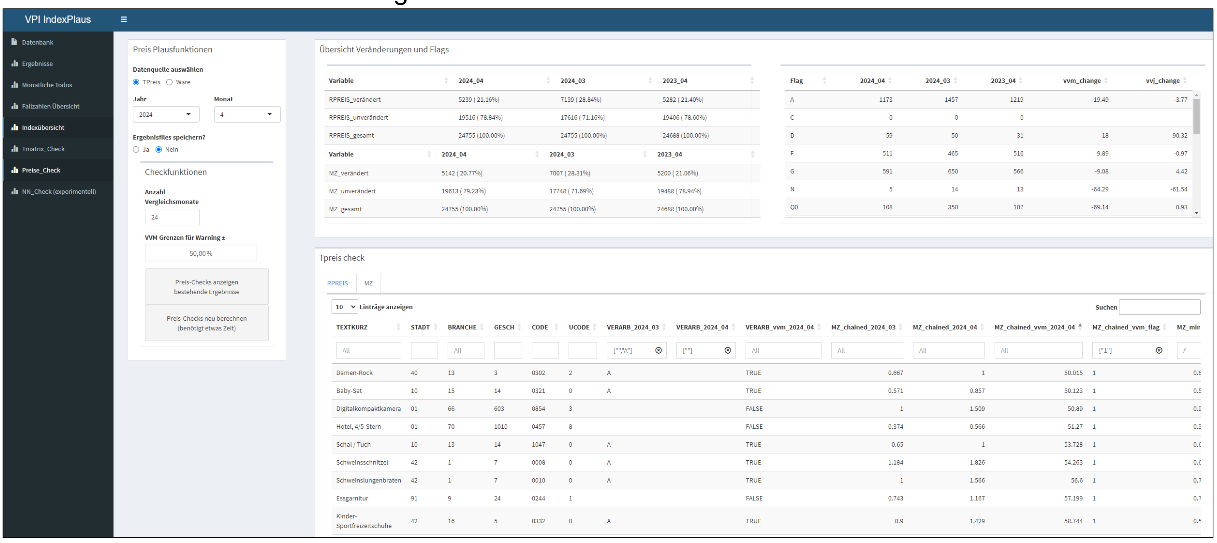
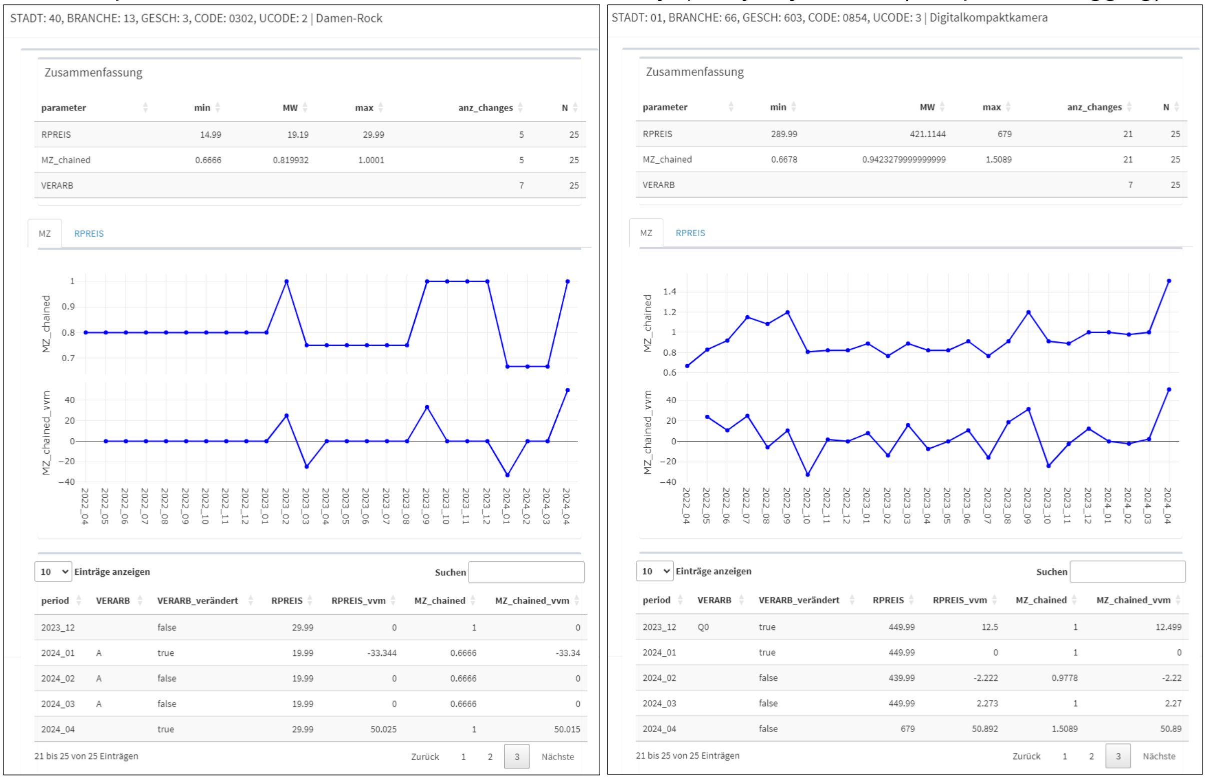


Figure 5.2: By clicking on a desired row in the table in Figure 5.1 additional information regarding the selected product will be displayed in table and chart form.

The left side example shows data for a skirt which got flagged as critical due to its high monthly rate of change of ~50%. A quick inspection of the timeline and the table shows, that the product was sold on reduced price in the last 3 month (indicated by the A flag in column VERARB) and now returned back to its original price. Therefore, this data can be confirmed as correct (false positive flagging).

The right-side example shows data collected for a digital camera. Just as the skirt this product got flagged as critical because of its high monthly rate of change of ~50%. Since no clear explanation for this huge price change could be found in the data, the case got forwarded to our product experts and was re-evaluated. The re-evaluation showed, that a previously unnoticed change in quality of the observed product occurred and was not accounted for by quality adjustment (true positive flagging).



1.5 Checks on aggregated data level (index level) tab

Checks on index level data (especially LLA) are another main feature of the application. All outlier detection methods described in Chapter 3 are or will be available here and can be applied by the user (currently nearest neighbour and ARIMA are listed in separate modules since they are considered experimental and will be moved over as soon as final decisions regarding setup parameters are made). Length of comparison periods as well as base year of the index series can be adjusted in case the recommended parameters do not fit the needs of the user (for example if younger index series, existing less than 36 months, should be included).

In case critical result are found they can be either reported to specialised product experts or further evaluated using the functions provided in the checks on raw data module (0) to identify corresponding critical cases on product level. It is also possible to apply the checks directly on all products, but this is rather not recommended due to the increased runtime.

Beside outlier detection users can also compare the current calculation status with previous runs to monitor changes in the results. A similar feature is available for comparing results for the harmonised index of consumer prices (HICP) and the HICP with constant taxes (HICP_CT). In periods with no tax changes both HICP and HICP_CT are supposed to follow the same trend (same monthly rate of change for all index series), in month with tax changes the index lines will deviate from each other for the aggregates affected by the tax changes.

Figure 6.1: Example screenshots for the checks at COICOP and LLA level, based on monthly rate of change. The top table shows a summary over the sums of weights on all aggregation levels. Any deviation from 100 indicates errors and needs to be evaluated in any case.
The lower table shows the results of the calculation for every aggregate. Beside the current rate of change also descriptive statistics over the past 36 months are presented. In case the current rate of change is larger or smaller than the max/min of the past 36 month, the absolute deviation to the previous extreme value is given in column Abs_diff_max. For example, the aggregate Teilkaskoversicherungen (partially comprehensive insurance) showed a rate of change of +8.6% in April. The largest previously observed rate of change was +4.6%, thus the current one exceeds this value by 4%-points and has to be re-evaluated to confirm the unexpectedly high change.

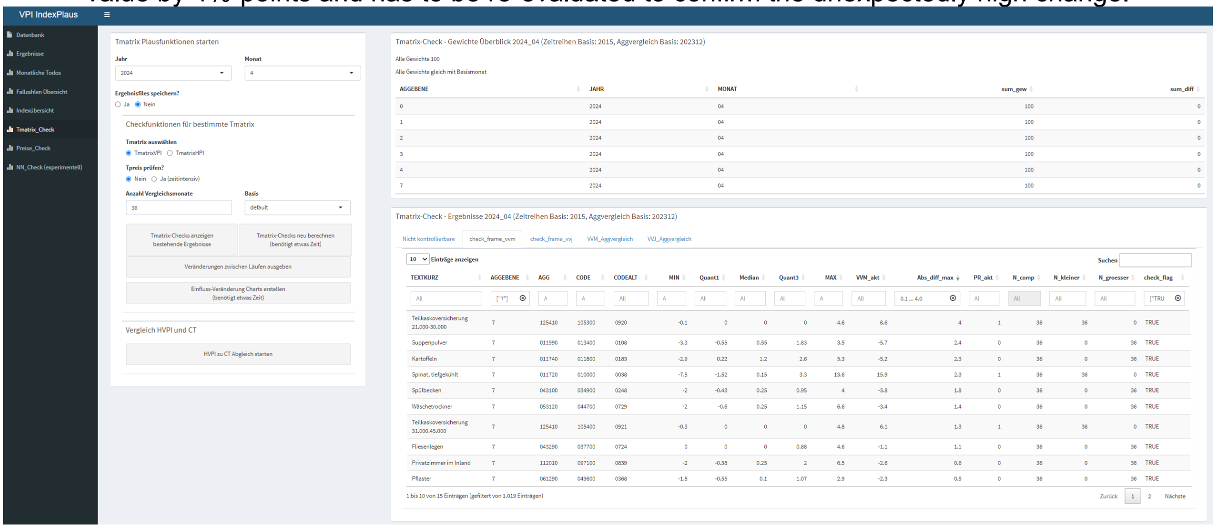


Figure 6.2: Before each new calculation run the CPI a backup with the current results is saved. To keep track of all the changes made over all the calculation runs during a month, the app offers a feature to compare the results from different runs and mark those COICOP and LLA groups that have changed in the last run. The example screenshot shows the comparison between Aprils current data base status (which holds the results from the last calculation run) compared to run 3 of the same month. The first table gives a summary of cases and changes found in each run. The second table shows the summed weights for all COICOP/LLA levels for the current and the previous run.
Table 3 shows all changes found between the current and the previous run. In this example, some changes in LLA related to clothing where found between the two runs.
The comparison feature also helps to reduce evaluation expenses since, after a new run, only changed LLA need to be re-evaluated in case a full check of the results has already been done after on of the previous calculation runs.

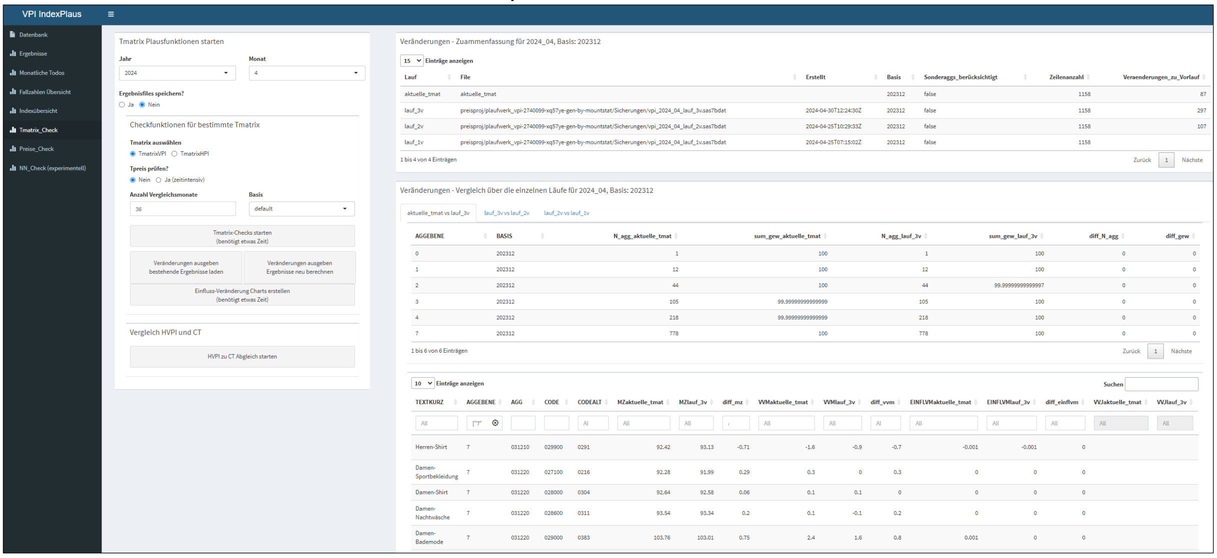
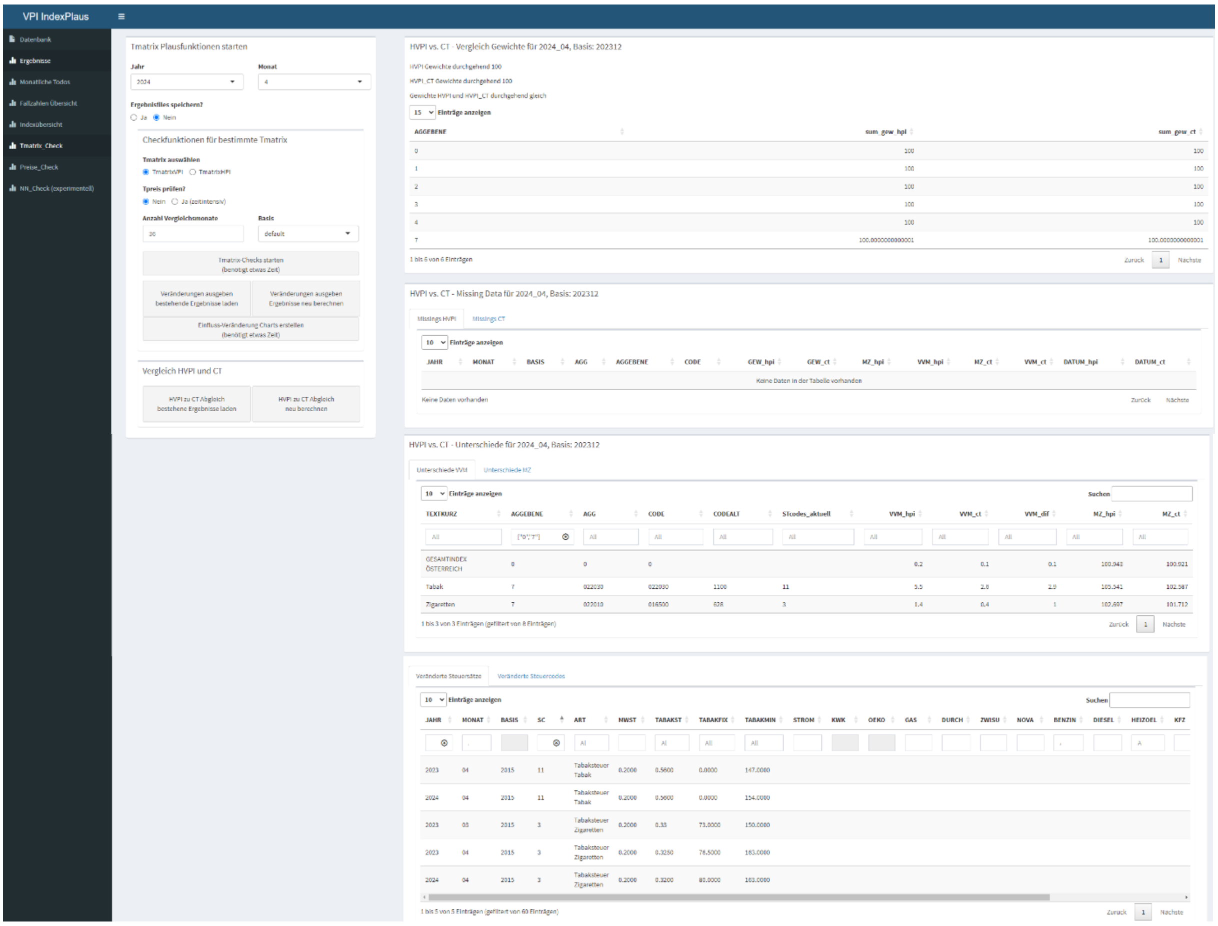


Figure 6.3: The example screenshot shows the results of comparing the HICP to the HICP_CT for April. For cigarettes and tobacco differences in the monthly rate of change where found. This is caused by a tax change in this month (see also Figure 2) which is considered in the HICP but not in the HICP_CT (where the same taxes as in previous December are applied). Whenever there is a tax change we expect to find differences between HICP and HICP_CT in the monthly rate of change. In the follow up month we still expect index values themselves to be different in HICP and HICP_CT but monthly rates of change to be the same.
The first two tables show again the weighted sums on all COICOP and LLA levels as well as a list of missing aggregates in case there are some. The third table contains the differences found between HICP and HICP_CT, either for the monthly rate of change or, in the second tab, for the index values. The last table gives an overview over all the tax changes reported in our data base, so users can confirm if found or missing differences are indeed justified by the current tax situation.



1.6 Nearest neighbour plausibility check tab – Experimental

In the current setup, users are required to define several parameters before initiating the outlier detection process. These parameters include the number of neighbours to search for, the time periods considered for the search, and the target measure for similarity evaluation, which can be the index value itself, or the monthly or yearly rate of change. Additionally, users must select a standardization method (z-transformation, min/max-transformation, or no transformation) for the chosen variable. One crucial parameter is a slider that determines the percentile (denoted as p) of the distribution over all distances found for all neighbours across all codes, signifying the proximity threshold for neighbour selection. Essentially, only the p% of neighbours with the smallest distances to their target code are deemed close enough for outlier detection. However, determining the optimal settings for these parameters poses a challenge, as it involves a trade-off between using only the most reliable and close neighbours for each COICOP code while still being able to find enough close neighbours for a sufficient amount of codes at all. Unfortunately, at the current state no best values for parameter setup can be recommended to the user.

After defining all required parameters calculations are carried out and a results-table, summarising the results for all neighbours found for each code, is displayed. This table can be filtered for entries marked as critical (target variable lies above or below all found close neighbour's target value). An on click feature allows to display summary tables and to graph timeline charts for the index value and the selected target value for the selected Code as well as all neighbours found. This allows a very quick evaluation so false positive outlier flags are considered less costly than false negative results.

Figure 7.1: The example screenshot shows the summary table for a nearest neighbour outlier check for April. The results are shown for using all, in this case four, neighbours (indicated by the columns nbg_...) and for using only neighbours considered as close enough (indicated by the columns near_nbg_...). In this example the unstandardized monthly rate of change from the past 12 months was used as target variable for the nearest neighbour search and the user specified, that only the closest 10% of neighbours found shall be considered as close enough. For evaluation we focus on the near_nbg results only and re-evaluate cases flagged as critical (see Figure 7.2 and 7.3).

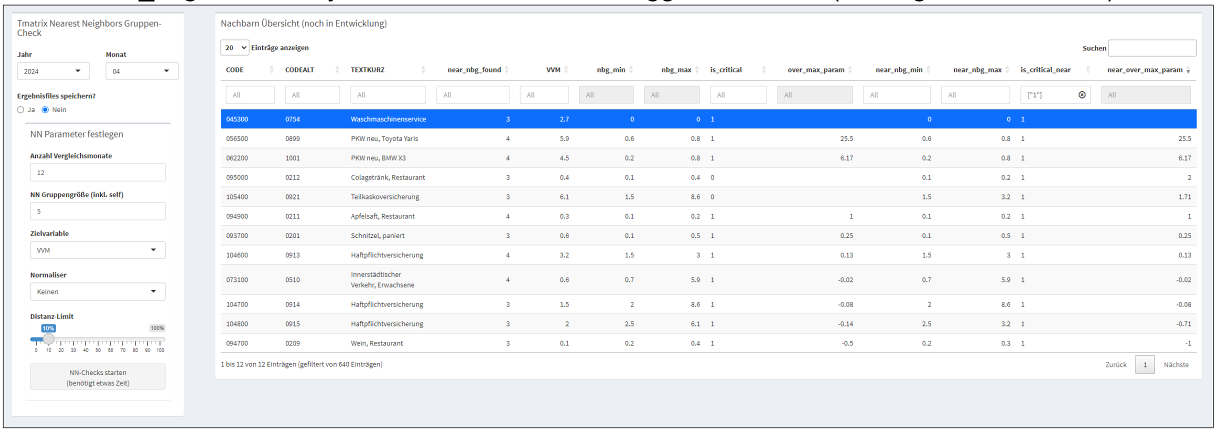


Figure 7.2: Clicking on one of the rows shown in the table in Figure 7.1 opens a tab showing the nearest neighbour outlier results for the selected LLA, in this example Waschmaschinenservice (Repair of washing machines), in detail. From the four neighbours found (Household services, Periodical assessment of road worthiness, Household services and Brokerage account fee) only three are considered close enough for outlier detection. Although brokerage account fee was the fourth closest neighbour, its distance to the target LLA was considered to big for a reliable neighbour. Although our product experts could confirm the increase, flagging this LLA was considered useful by them since the increase was rather unexpected and re-evaluation was justified.

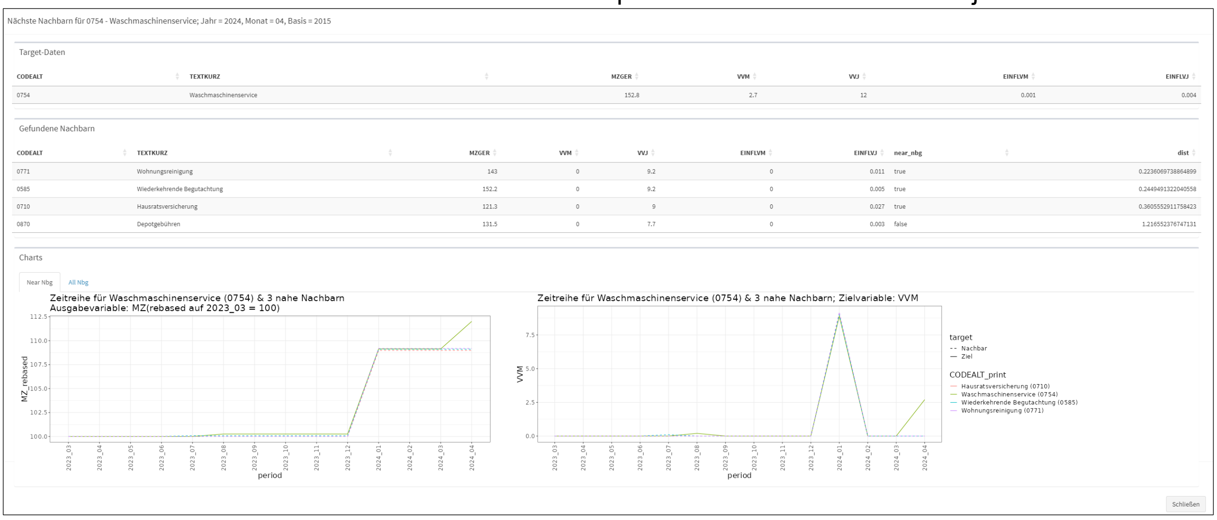
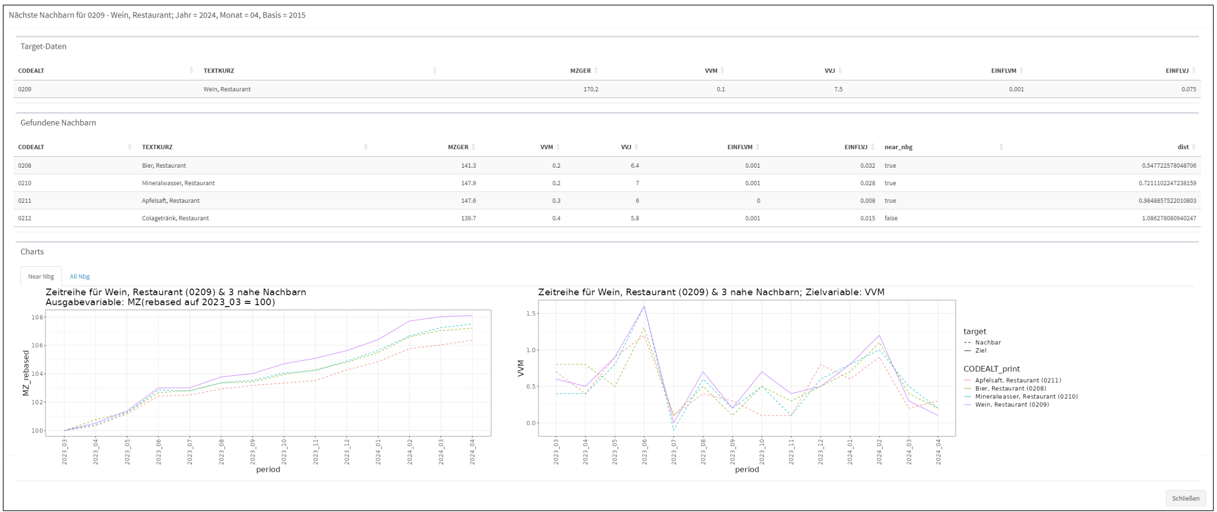


Figure 7.3: This example screenshot shows the nearest neighbour outlier results for LLA Wein, Restaurant (A glas of wine - restaurant) in detail. All four neighbours (Beer – restaurant, Mineral water – restaurant, Apple juice – restaurant, Coke - restaurant) found were considered close enough to be reliable for outlier detection.

In this example product experts considered the flagging as to strict. Even though the monthly rate of change of the target LLA is smaller than all the neighbours' one, the deviation is too little to justify re-evaluation of the whole LLA.



Nächste Nachbarn für 0209 - Wein, Restaurant; Jahr = 2024, Monat = 04, Basis = 2015

Target-Daten

CODEALT	TEXTKURZ	MZGER	VVM	VVJ	EINFLVM	EINFLVJ
0209	Wein, Restaurant	170.2	0.1	7.5	0.001	0.075

Gefundene Nachbarn

CODEALT	TEXTKURZ	MZGER	VVM	VVJ	EINFLVM	EINFLVJ	near_nbg	dist
0208	Bier, Restaurant	141.3	0.2	6.4	0.001	0.032	true	0.547722578048706
0210	Mineralwasser, Restaurant	147.9	0.2	7	0.001	0.028	true	0.7211102247238159
0211	Apfelsaft, Restaurant	147.6	0.3	6	0	0.008	true	0.9848857522010803
0212	Colagetränk, Restaurant	139.7	0.4	5.8	0.001	0.015	false	1.086278080940247

Charts

1.7 ARIMA plausibility check tab – Experimental

To start the auto.arima based time series outlier detection process, users have to specify how many previous years (1 to 5) should be used to fit the model.

Smoothing can be applied and if so is done by a rolling mean over the last 3 to 12 months. When it comes to outlier detection, it is not clear yet if smoothing can increase the quality of the results. One advantage is that small fluctuations in index values are mitigated by smoothing, generally resulting in better-fitting time series models with narrower confidence intervals. On the other hand, smoothing also affects true outliers, in worst case, hiding them completely.

Currently users are encouraged to try both the approach with and without smoothing to gain better understanding for which one is more useful in practice (Using smoothed values for model fitting and the unsmoothed current value for evaluation against the predicted value cannot be recommended: Due to the model's narrower confidence interval and greater volatility in unsmoothed data, their combination results in a large number of false positive flags).

After fitting the auto.arima model using data from the previous 12 to 60 month as defined by the user, it is used to predict the value for the current period. If the current month's measured value falls outside the 95% confidence interval of the model prediction, an error flag is set.

To indicate the severity of the deviation between predicted and measured value, the distance between the measured value and the closer predicted confidence interval limit is taken and divided by the range of the confidence interval, so that high values indicate critical deviations.

All results, including the error flag as well as the deviation parameter, are displayed in a table. Filtering the table for critical values based on the error flag and sorting by the deviation parameter in ascending order helps to identify the most problematic cases quickly. Again, an on click feature is implemented which opens a dialog window with additional information about the selected case.

Figure 8.1: Example screenshot showing the results table after time series outlier search based on a three years time series without applying smoothing. The table has been filtered for critical cases (column flag_arima) only and was sorted ascending by severity (column over_max_param_95). Teilkaskoversicherungen (partially comprehensive insurance) was flagged as critical because its measured value (127.79) was 1.38 times the confidence interval range (115.51 – 120.67) higher than the upper limit of the predicted confidence interval (120.67)

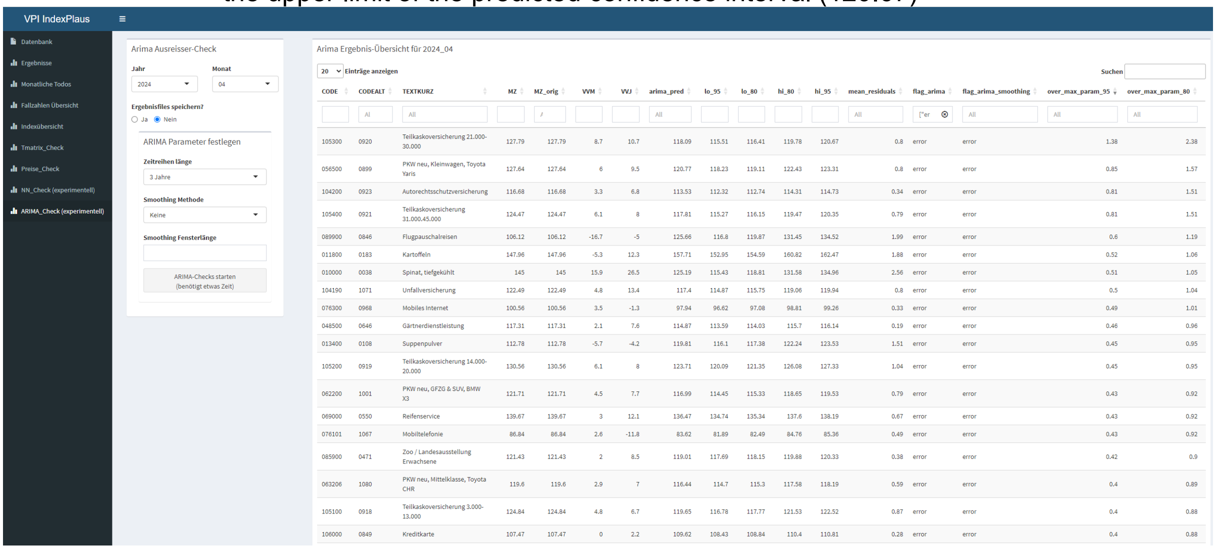


Figure 8.2: Example screenshot showing the detailed results for Teilkaskoversicherungen (partially comprehensive insurance), which was flagged as critical by the time series outlier detection. Although the present index increase is indeed worthy of examination, the quality of the Arima prediction in this example must still be questioned. Since there seems to be no obvious seasonality in the data and larger index changes occur more or less unsystematic, the model is dominated by its autoregression component. Therefore, index changes cannot be predicted accurately by using time series data only.

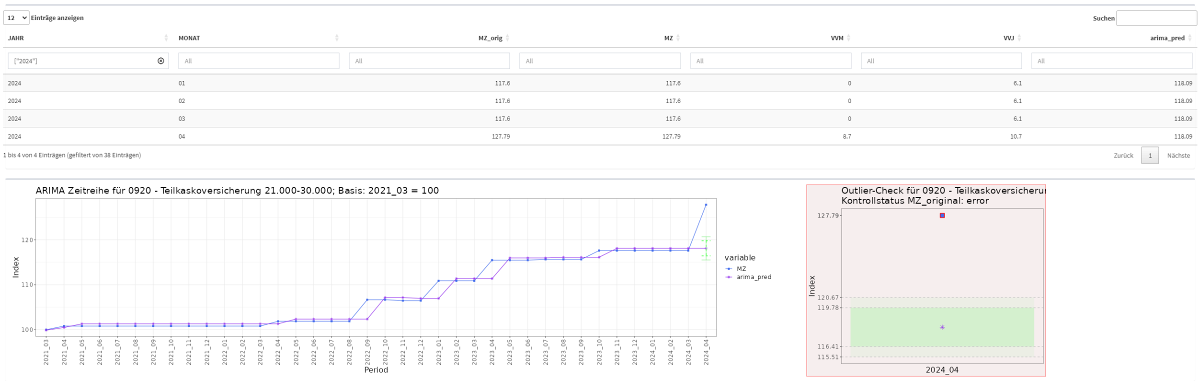


Figure 8.2: Example screenshot showing the detailed results for Zoo/Landesausstellung Erwachsene (zoo/exhibition, adult ticket), which was flagged as critical by the time series outlier detection. In contrast to the previous example in Figure 8.1 here a clear seasonal index pattern with an index increase from December to January can be seen. This pattern is well captured by the Arima model. The unexpected index increase in this years April on the other hand does not seem to fit the pattern and therefore got flagged by the model.

12 Einträge anzeigen

Suchen

JAHR	MONAT	MZ_orig	MZ	VVM	VVJ	arima_pred
["2024"]	All	All	All	All	All	All
2024	01	118.03	118.03	3.3	5	118.28
2024	02	119.04	119.04	0.8	6	118.52
2024	03	119.04	119.04	0	6	119.1
2024	04	121.43	121.43	2	8.5	119.01

1 bis 4 von 4 Einträgen (gefiltert von 38 Einträgen)

Zurück 1 Nächste

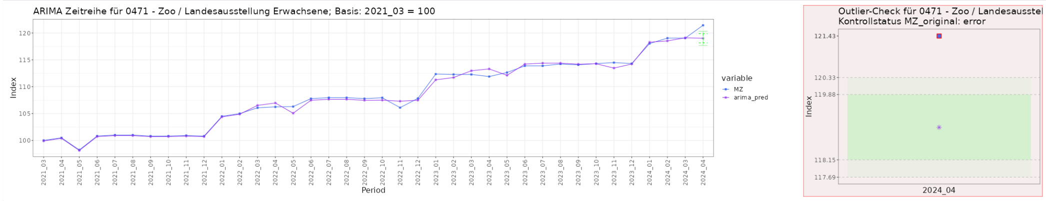


Figure 8.3: Example screenshot showing the detailed results for Damen Blazer (Ladies Blazer), which was not flagged as critical by the time series outlier detection. Clothing shows a very strong seasonal pattern. Since there was no deviation from this pattern in this years April, the measured index value lies well within the predicted confidence interval.

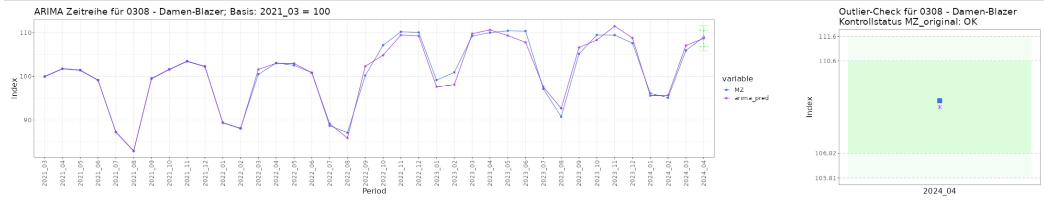
12 Einträge anzeigen

Suchen

JAHR	MONAT	MZ_orig	MZ	VVM	VVJ	arima_pred
["2024"]	All	All	All	All	All	All
2024	01	96.1	96.1	-10.7	-3.2	95.58
2024	02	95.12	95.12	-1	-5.8	95.66
2024	03	105.94	105.94	11.4	-3	107.08
2024	04	108.97	108.97	2.9	-0.9	108.71

1 bis 4 von 4 Einträgen (gefiltert von 38 Einträgen)

Zurück 1 Nächste



1.8 Index Summary tab

The summary module is designed to give a quick overview of the current index results on the highest two COICOP levels, including timelines for the past two or five years. Users can choose between a simple time series mode, in which all data is presented at the oldest available base, or a comparison mode, in which the data is grouped by year and based on the previous year's December, so that each year-group's index line starts with 100.

In addition, high influential COICOP codes for the overall index result and for the index change between current and previous month can be filtered and plotted.

Figure 9.1: Example screenshot showing the CPI results in comparison mode on the highest COICOP level. This year's index development is significantly lower than the one in the previous two years.

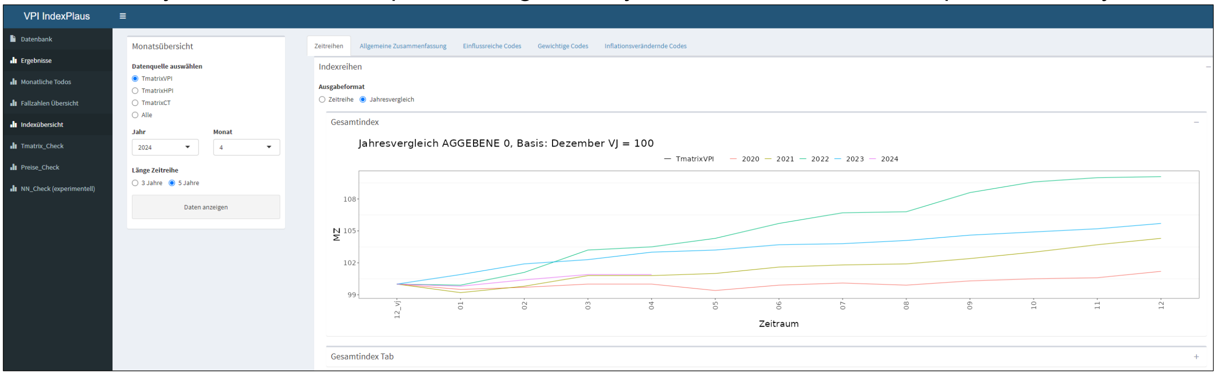


Figure 9.2: Example screenshot showing the CPI results in simple time line mode on the second highest COICOP level since December 2019.

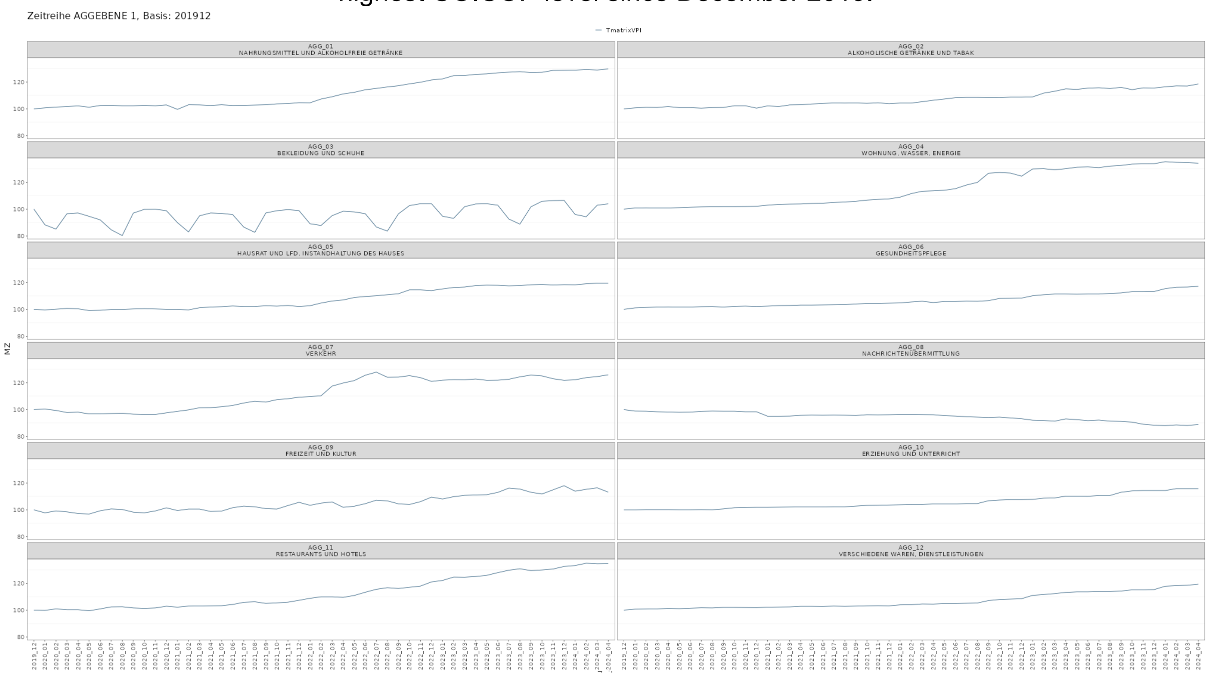
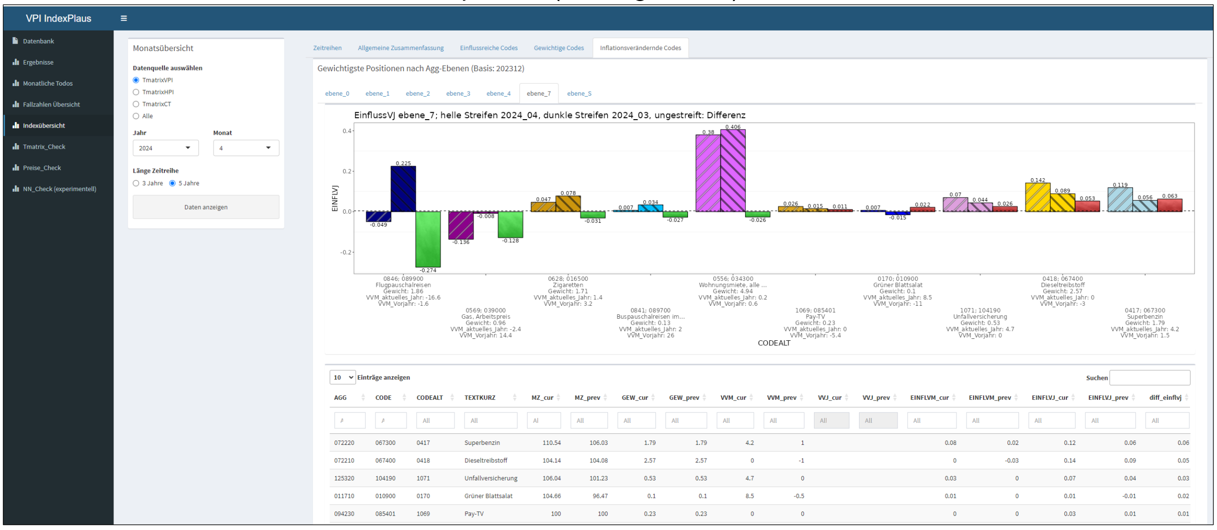


Figure 9.3: Example screenshot shows the 5 LLA with the largest positive and the 5 LLA with the largest negative impact on the change in April's inflation compared to the previous month. Flugpauschalreisen (Holiday package by plane) had is the LLA contributing most to the index drop from Mai to April. This is explained by a base effect due to last year's disruption of the typical seasonal pattern (see Figure 9.4).



AGG	CODE	CODEALT	TEXTKURZ	MZ_cur	MZ_prev	GEW_cur	GEW_prev	VVM_cur	VVM_prev	VVJ_cur	VVJ_prev	EINFLVM_cur	EINFLVM_prev	EINFLVJ_cur	EINFLVJ_prev	diff_einflvj
072220	067300	0417	Superbenzin	110.54	106.03	1.79	1.79	4.2	1			0.08	0.02	0.12	0.06	0.06
072210	067400	0418	Dieseltreibstoff	104.14	104.08	2.57	2.57	0	-1			0	-0.03	0.14	0.09	0.05
125320	104190	1071	Unfallversicherung	106.04	101.23	0.53	0.53	4.7	0			0.03	0	0.07	0.04	0.03
011710	010900	0170	Grüner Blattsalat	104.66	96.47	0.1	0.1	8.5	-0.5			0.01	0	0.01	-0.01	0.02
094230	085401	1069	Pay-TV	100	100	0.23	0.23	0	0			0	0	0.03	0.01	0.01

Figure 9.4: Index viewer output for the LLA Flugpauschalreisen (Holiday package by plane). From Mai 2023 to April 2023 an otherwise typical seasonal drop could not be observed in the data.