Analytical Oversights

This document summarizes our understanding of analytical oversights that was developed over the course of our [internship project](https://docs.google.com/presentation/d/1U9B9uPZ5zn2kN3grx9j9rAboJEdZfNv97q6sBx4ZMMs/edit).

# Index

1. [MeanVsMedian](#lxmqrmhiv86t)
2. [Wrong Points](#_x85yun8dt6ty)
3. Top-k Related
   1. [Regression to the mean](#pdtptxg6v9ne)
   2. [Looking at Tails to Find Causes](#dljd3kvkd5qq)
   3. [Top-k versus other](#ekykodjndax0)
   4. [More than just top-k](#kix.ebtgdpz0iloe)
   5. [Duplicates in top-k](#fijhlrcgcycy)
   6. [Top-10 when 9 are present](#kix.mrfyx6m6n1g2)
4. Comparing Apples vs Oranges
   1. [Confounded Comparisons](#19n9cihv8obv)
   2. [Composition Effects in Time Series](#hj1smmw7sas4)
5. Comparing 2 sets Vs Comparing 2 of their subsets (Simpson’s paradox similar)
   1. [Simpson’s paradox](#v69mu8e8y1o)
   2. [Top-down error](#9i5d5pwo6zu8)
6. Comparing Benchmark
   1. [Benchmark Set too Different](#s7qjywawqlgu)
7. Negative Entries causing cancellation
   1. [Attribution with hidden negative](#8rfz69l0clbq)
8. Uneven weights while averaging
   1. [Duration Averaging](#_mbg7u5q7629j)
9. ON HOLD
   1. [Multiple Benchmarks](#_1vbuhk6vrb2v)
   2. [Anscombe Quartet Error](#_f66o5h3k5gqi)
   3. [Prediction cone](#_snz8fhu0zkpx)
   4. [Reclassification](#m4pxg5pi2695)

Oversights

# MeanVsMedian

Author : Anmol

|  |  |
| --- | --- |
| Scope of queries | User uses <summary\_operator = *MEAN>* but using <summary\_operator = *MEDIAN>* in the same query results in a very different result. |
| Detection | Compute the SKEW of the values ,  The exact formula of SKEW is given [here](https://support.microsoft.com/en-us/office/skew-function-bdf49d86-b1ef-4804-a046-28eaea69c9fa).  It represents the skewness of data and is a unitless parameter , hence giving a good estimate of how far the mean and median are. |
| Debiasing suggestion | *MEAN* is different from *MEDIAN* at \_\_\_\_\_\_\_ |
| False positives | Exponential distribution , Clustered data , etc . , as illustrated in this [doc](https://docs.google.com/document/d/1g7fbdOId2V1hIhar6Gy8i8YFEi3_07jM_VE6Y2EoSEs/edit) |
| Detailed Docs Link | Initial Discussion on Hidden Outliers : [Doc](https://docs.google.com/document/d/1EIBRPfhXZZJkid9Ll8-x6rNjpr2Qgi1YB7HlIzxvNIM/edit?ts=5ed8856f#)  Final doc with a bunch of examples and conclusions from meeting : [Doc](https://docs.google.com/document/d/1g7fbdOId2V1hIhar6Gy8i8YFEi3_07jM_VE6Y2EoSEs/edit)  Skew approach explained : [Slides](https://docs.google.com/presentation/d/1WIjn6Ru5X9LUwxzOXV7C7e8q12um65DbLPQP4olFd5Q/edit#slide=id.g80cb0e68aa_2_75) |

# Wrong Points

Author : Anmol

|  |  |
| --- | --- |
| Scope of queries | Applying *any intent* on a <metric = A> which contains some entries which are <Wrong Entries = WE> |
| Detection | Before detection we need to ask ourselves what all categories of WE can exist , here is a list of some WE ,   1. **Entries representing NO data :** In some scenarios User can have some of the entries as some extremal values indicating the absence of data.    1. E.g : Some entries being 0 representing there was no entry, this could affect the denominator while taking averages and could also affect the sum because the User might think that he/she took the sum of all entries without realising that some of the entries were missing.    2. E.g : (Here we make the assumption that -99 is an extreme minimum value i.e all values in <A> are greater than -99) Some entries having -99 which represents that no data was present there , so doing any operation on <A> would result in something wrong.    3. **DETECTION :**        1. **Approach :** For an extremal (min or max or 0) value , if it exists a lot of times in <A> then we detect .       2. The above approach does seem incomplete as when the number of -99 present in <A> is less we won’t be able to detect it , but this seems like a case that if we handle it would lead to increasing the number of false positives. 2. **Some Values being greater than the sum of all the rest of the values:** (Only applies for +ve metric , as there would be some sort of cancellation for mixed +ve & -ve metric ) In most scenarios this would be a WE as we don’t expect an entry to be greater than the sum of all the rest of the entries in a +ve metric. This is because most of our summary operators won’t make much sense in these scenarios.    1. E.g : In a metric of *salary of employee* , *profit in this year* , *marks of student , runs scored by team* , *etc.* (In most metrics this holds)    2. C.e (Counter Example) : For any *metric* which grows exponentially this won’t hold , like for some *metric : Population , COVID cases , App downloads (for a viral app) .* As in exponential rates we do expect that later entries will be greater than the sum of all previous entries.    3. **DETECTION :**        1. **Approach :** The approach is the name itself, for any +ve metric check if any value is greater than the sum of the rest of the entries.       2. We could also introduce a threshold here to generalize/soften the idea a little,       3. Compare the difference of the largest value with the sum of the rest of the values with some threshold. |
| Debiasing suggestion | Both categories will have different debiasing suggestions which shouldn’t be too hard to point out the user as these are very explanatory and easy to understand once we detect it. |
| False positives |  |
| Detailed Docs | Initial discussion of Wrong Points Oversight emerging from Hidden outliers : [Doc](https://docs.google.com/document/d/1EIBRPfhXZZJkid9Ll8-x6rNjpr2Qgi1YB7HlIzxvNIM/edit?ts=5ed8856f#) |

# Regression to the Mean

Author : Bhagya

|  |  |
| --- | --- |
| Scope of queries | Top-k query in which date range aspect is passed & the top-k results in the previous date window differ a lot from the present date window.  This oversight basically alerts the user that the top entries in the date window are not consistent & they are in the top-k only for a short period of time & in the longer run these fall back to the mean.  Examples:   * [Regression to the Mean in Flight Tests](http://economics-files.pomona.edu/GarySmith/flightTests.pdf) They wrongly assumed that criticizing the pilots improves their performance. But later it turned out that performance had nothing to do with feedback.  They did not quantify regression to the mean in the paper. * [Average height of parents and their children](https://academic.oup.com/ije/article/34/1/215/638499#10889752) It was found out that the height of parents and children have a negative correlation. Children with parents at the extreme ends of the distribution had heights closer to the population mean height. * [Placebo Effect](https://www.webmd.com/pain-management/what-is-the-placebo-effect#1) |
| Detection | S1 = set of top-k in the given time range  S2 = set of top-k in the just previous time range of same length  Test 1: | S1 ⋂ S2 | ➗ | S1 U S2 | is small  This test checks if very few elements of S1 are there in S2  Test 2:   1. Calculate the k-dimensional rank vector({Mapping of element : It’s rank}) for both the results S1 & S2. 2. Calculate the angle between those 2 vectors using the dot product between the 2. 3. If angle is large it means the 2 vectors differ a lot.   This test helps to check the case when the top-k ranking gets reversed in the other interval. |
| Debiasing suggestion | * If Test 1 is positive : Very Few / None of the top-k in the given date range are present in the previous windows top-k. * Else If Test 2 is positive : Ranks of the top-k in the date range differs much from the previous windows topk |
| False positives | Ex. The user is looking for top-k to give away prizes. |
| Detailed Docs | Detailed analysis of RMT : [Doc](https://docs.google.com/document/d/15BYnPkyZm0c7JX6mbxPBxBQ_erk0AYkYrUMejp5-O30/edit#)  A bunch of examples : [Slides](https://docs.google.com/presentation/d/18r6zRfXO4qeUYOEg7B71hue3vWhDquM3NR8hiMfNSW0/edit#slide=id.g8858a4b49b_13_11) |

# 

# Looking at tails to find causes

Author : Bhagya

|  |  |
| --- | --- |
| Scope of queries | Whenever user asks for top-k [dimensions] according to some metric (say A) then some metric (say B ∈ given dimensions) does not depend on the tail of the list ie. B is similar everywhere.  Examples.  Top - 10 batsman and their age based on average runs per over.  Here it might be the case that the age of the top-10 may be similar to the age of the bottom-10. |
| Detection | Parameter = | Average 1 - Average 2 | / ( | Average 1 | + | Average 2 | ) .  Iterate over all dimensions and   1. If It contains real numbers -    1. Compute statistics for the top-k and <entire distribution>    2. If Sign of both averages are different - NO debiasing    3. Else if parameter < constant\_cut\_off , give a debiasing 2. If It contains dates -    1. Current parameter is useless here as normalizing with the sum of dates is meaningless. Also we can’t sum dates.    2. | Average 1 - Average 2 | / SD as a parameter works. 3. If it contains strings . ( will deal with proportions here )    1. If distinct\_count <= 3 , ie. we have binary / ternary variables    2. Calculate the proportions in top-k and <entire distribution>    3. Check if they are similar - using one of these       1. Cosine similarity       2. Manhattan distance    4. If they seem similar return a debiasing.   If there are multiple dimensions in which we can give a debiasing-  We take the one in which the parameter is minimum (Top-k is most similar to the entire distribution) |
| Debiasing suggestion | The values of <dimension\_name> in top-k are similar to the entire list.    (Also we might give the user a button to query for the other tail)  Note -  **The SUGGESTING NEW QUERIES is easy to implement -**  **With the debiasing suggestion, we also return the request json object for the new function call, with all the parameters preset - And the UI will use that json to query for the other tail.** |
| False positives | If we have mobile numbers in a column, then we might give a debiasing based on that column. |
| Detailed Docs | Detailed analysis of oversight : [Doc](https://docs.google.com/document/d/1JuOufU5Du5IADXJRciSkxt1kPCYX_rOc5T9EJ8mvzdo/edit#heading=h.qotx0elvqt8s) |

# Top-k versus others

Author : Bhagya

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scope of queries | Top-k in which the entries that are not in the results, have a huge impact.  Ex.  A table of taxpayers -- query for top - 3   |  |  | | --- | --- | | Name | Tax paid | | A | 20 | | B | 15 | | C | 13 | | D | 10 | | Continued... | At least 1 |   Here the entries that are not in the top-k have a large summed up value, so here it is useful to look at the items that are not in the top-k |
| Detection | Basically we need to detect the long tail, as in fig.    Why does the tail seem large in the fig? Because it is ~10 times the mean of the top-4.  Parameter - sum of others / mean of top-k (1)  Parameter - sum of others / sum of top-k (2)  (1)/(2) = k  If the value of the parameter is greater than some cut-off we give the suggestion. |
| Debiasing suggestion | The top-k you are looking at represents a very small part of the data, you may want to look at the entries which are not in the top-k.  If parameter is negative :   * The sum of top-k is positive/negative whereas the sum of other rows is of the opposite sign.   ~~OPTION : To add other rows in the top-k results OR add “Others” as a row.~~  ~~The others will be the summation of metric over all entries that could not make up to the top-k.~~ |
| False positives | In some cases, summation of metric does not make sense  Example query - top-10 SAT scores. |

# 

# 

# 

# More than just Top-k

Author : Bhagya

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scope of queries | Top-k queries in which some entries just after the top k entries are similar to the top-k ones.  Motive of giving a suggestion here is that the user does not miss those entries that are metric close to the k-th entry.  Ex.  PM of a country deciding to give funds to top-3 covid affected states.  Result -   |  |  | | --- | --- | | State | Metric (% population infected) | | Delhi | 1% | | Maharashtra | .7% | | Tamil Nadu | .6% | | Gujrat | .59% | | West Bengal | .59% | | UP | .3% |   \* % figures are just made up.  Here Gujrat missed the top-3 (and also funds) as it was not in the top-3. A difference of .01% could also be noise(/ errors in measurement).  So it is nice if we suggest the user to look at the top-5 instead of the top-3.  Suggestion -”Next - 2 States have Metric closeby to the 3rd” [Option : try top - 5] |
| Detection | Here with the suggestion we give we also suggest to look at the top-x entries instead of top-k (x > k).  So, here the parameter works a bit differently(than the rest of the oversights). We decide the range till we consider the next entries.  Let, SD = standard deviation of top-k  Valk = Value of metric at kth position  α = Pre decided parameter  Range = [Valk  - α \* SD , Valk  + α \* SD].  Having extension of the range at both ends helps us generalize both cases of “Maximum top-k” & “Minimum top-k”  Intuition of having SD in deciding range - We need to compare the difference to the k + t th entry with the kth & also we need a unitless parameter, so having variability seems the best option to me.  Options other than SD could be-   1. Mean of top-k 2. Mean of 1th & Kth entry 3. Median of top-k   But SD (IQR can also be considered) is best as it handles cases of -ves & mean close to 0 well. |
| Debiasing suggestion | Next <T - K> entries are close to Kth. [Option - to look at top-T] |
| False positives | For a strict professor this suggestion would be FP.  Also,  In general cases where the user can’t be considerate.  Ex.  Selecting top scores for college admissions.  But there is no harm in giving suggestions. |

# 

# Duplicates in top-k

Author : Bhagya

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scope of queries | Top-k queries in which the result table has duplicates in the dimensions column.  Example -  User’s query :  “Top-2 Students with maximum Marks”  Result table :   |  |  | | --- | --- | | Student | Marks | | A | 90 | | A | 80 |   “Dimensions” = list of column names user enters  Here there are duplicate “dimensions” in the top-2 entries (Both are ‘A’) so this is considered as an oversight.  As Ideally there should be only one metric per “dimensions” (Student here).  This means that possibly the user has missed something to the query, maybe a group by operation or slicing operation.  Original table :   |  |  |  | | --- | --- | --- | | Student | Semester | Marks | | A | 1 | 90 | | A | 2 | 80 | | B | 1 | 79 | | B | 2 | 67 | |
| Detection | Consider only the “dimensions” columns of the rows & see if there are duplicate entries in it. |
| Debiasing suggestion | There are more than 1 row present for a “Student”  Generic -  There are multiple rows present for some of the (“dimension[0]”, “dimension[1]”,..) |
| False positives | There should be **NO** false positives, as having multiple rows for the same <Student> never makes sense. |

# Top-k when less than k present

Author : Bhagya

|  |  |
| --- | --- |
| Scope of queries | Top-k queries in which the results contain less than k entries.    Image - it might be obvious to the user in this case. But if k = 1000 & only 990 entries are present then it is worth pointing out. |
| Detection | Simply,  If the number of entries in the result table are less than k - we give the debiasing.  But if “k = 4” & only 3 entries are present in results - here giving the suggestion might not be useful as this fact would be evident to the user.  Let <M> = Number of rows we see without scrolling.  Also, we would give the oversight only when the result table is large compared to <M>.  Parametrer = ratio of (number of entries in results - k ) / <M> |
| Debiasing suggestion | “Instead of <k> there are only <k’> entries present in the results”  Also we could give a suggested query - see top-<k’> . |
| False positives | Interestingly, NO false positives. |

# Confounded Comparisons

Author : Anmol

|  |  |
| --- | --- |
| Scope of queries | <summary\_operator = S> of some <metric = A> for <set1 = s1> and <set2 = s2> ,  Here , S  But here the <s1> and <s2> are very different in the nature of elements they possess.  In other words <s1> and <s2> are different in other ways so it may not be the best idea to compare them. |
| Detection | * The detection part is very similar to *Composite Effects in Time Series*   + **Try to found another metric** <A’> **as the Confounder :**     - Fix another metric <A’> and check the result of summary operator <S> applied on <A’> for <s1> and <s2> ,       * If <s1> and <s2> are very different on this result then we can conclude that sets <s1> and <s2> are very different on the basis of <A’>.   + **Try to found another dimension** <D> **as the Confounder :**     - Fix a dimension <D> and check whether the elements of <D> in <s1> and <s2> are very different ,       * If <s1> and <s2> differ a lot in dimension <D> then they are different entities on the basis of <D> |
| Debiasing suggestion | It’s possible the differences you see are accounted for by other variables  Show the set of <A’> and <D> that are confounders |
| False positives |  |

# Composition Effects in time series.

Author : Bhagya , Anmol

|  |  |
| --- | --- |
| Scope of queries | Trend queries and in the data, no item is present in some period. |
| Detection | Parameter = Number of periods where < count(metric) is 0 > / Total periods in result.  Parameter 2 = Standard Score over count(metric) over time periods.  We can use the second parameter if we also want to consider the variability in the count over time.  If the parameter exceeds some value, we give the suggestion.  Example of result of trend query -  date avg win\_by\_runs  0 2008-01-01 105  1 2009-01-01 92  2 2010-01-01 98 |
| Debiasing suggestion | 1. Give a reminder that not all units are present in each period. 2. OPTION : to add a count(metric) column besides the average(metric) |
| False positives | Example.  Query : Weekly trend of average number of visitors in tourist places in India.  But due to lockdown all tourist places were closed and the average was zero.  This is not an oversight, as this was an expected result and the user would be aware of the lockdown fact. |
| Detailed Doc : | Detailed Analysis : [Doc](https://docs.google.com/document/d/1HYwmnmpmhjwJsUAxEkmcuea9PTRx83f2tQ4D6lUP44M/edit#heading=h.xffjtuy44pve) |

# Simpson’s Paradox

Author : Chandan

|  |  |
| --- | --- |
| Scope of queries | Whenever a user asks to compare two objects based on any criteria this effect can appear. So this oversight might appear in slice-compare and time-compare.  The query will be like:  Compare the <data> for <value1 and value2> by <column(s)> where <filter(s)> from <date range> |
| Detection | 1. For each dimension not in dimensions we shall add that column in the grouping 2. For each slice we shall compare the dominant percentage of the initial group and the group made just by adding a new column.    1. Dominant percentage can be defined as for how much percentage of the value of the first slice is greater than the second slice. 3. If the difference in between the initial dominant percentage and the dominant percentage after addition of a new column is greater than SIMPSONS\_PARADOX\_DOMINANT\_PERCENTAGE then we shall give an oversight. |
| Debiasing suggestion | The relation between slices might change a lot if you will consider new\_added\_column in grouping. |
| False positives | If we have mobile numbers in a column, then we might not be able to give debiasing suggestions. |
| Detailed Doc | Detailed Analysis : [Doc](https://docs.google.com/document/d/1n3vUxsmmRH0WuyP3gHQeHWVPqsREcdZcLqaUnJbN5dY/edit#heading=h.blgjzjl8f3t2) |

# Top-down error

Author : Anmol , Chandan

|  |  |
| --- | --- |
| My understanding by example | Users compares the average marks (<metric-A>) of a student between 9th grade (<duration\_1>) and 10th grade (<duration\_2>) and the average marks(<A>) turns out to be almost similar , so makes the assumption that the student is still the same even in 10th grade.  But it could be that the students marks in 9th grade in English(<dim D1>) are very low but in grade 10th it is very high and  Students' marks in 9th grade in Math(<dim D2>) are very high but in grade 10th it is very low ,  so overall they cancel out thus leading to similar average total marks of students in 9th grade and 10th grade.  But in this scenario we want to give this oversight : “*Look at Dimension:Math,English it is a lot different among 9th and 10th grade so the student seems to have changed between 9th grade and 10th grade* ” ,  This could be useful for the user in analysing that the student has started focusing more on English but has stopped practising maths , whereas without the oversight the user would have directly assumed that the student is doing the same as before. |
| Scope of queries | Time-compare <metric-A> between <duration\_1> and <duration\_2> and the comparison gives that the results are very similar to each other. |
| Detection | * <A> for <duration\_1> is similar to <A> for <duration\_2> * But , <A> for <duration\_1> over <dim D> is very different to <A> for <duration\_2> over <dim D> * In words <duration\_1> and <duration\_2> look similar when we only look at <A> (maybe due to some cancellation) but when we zoom-in and only restrict ourselves to a portion of <A> which is by fixing a dimension D then they appear to be different. So the assumption of the user about <duration\_1> and <duration\_2> being similar could be wrong. * The above similarities and differences could be computed by our standard methods of normalized/unitless differences between 2 numbers that we have been discussing for other oversights as well. |
| Debiasing suggestion | * <A> between <duration\_1> and <duration\_2> differs if we look at dimension <D> |
| False positives |  |
| Detailed Doc | Detailed Analysis Doc : [Doc](https://docs.google.com/document/d/1ozekvQzy24DNqoUk665ehGdlM34Eyjrsz-1dvn6vh7k/edit?ts=5f0710cd#) |

# Benchmark set too different

Author : Bhagya , Chandan

|  |  |
| --- | --- |
| Scope of queries | Queries in which the user compares a comparatively small slice of the data to the entire data, and the comparison might have large variations.  Ex,  Comparing sales of a single shop with sales of all shops in the market, and since the shops would be having a large variation in number of customers, so it is not meaningful to do such a comparison.  User compares metric Y for slice A to metric on Y full data - Y1 and Y2  Bias is: A and full data also differ on metric X - X1 and X2 |
| Detection | We measure the variability in the result of slice comparison, and if the range of the percentage comparison is large, ex [-100%, 100%] we predict the oversight.  Since the values are percentage, and are already normalized, we may use any of these measures of variability.   1. Standard Deviation 2. IQR 3. Range 4. Variance   (Y1-Y2)/(Y1+Y2) vs (X1-X2)/(X1+X2)  //What if (Y1-Y2)/(Y1+Y2)>0 but (X1-X2)/(X1+X2)<0  // It’s better to handle this case separately |
| Debiasing suggestion | 1. The slice is not a representative of the entire data. 2. ~~OPTION : To plot the distribution of metric over dimension~~   The slice differs on Y but it also differs on … |
| False positives |  |
| Detailed Docs | Detailed Analysis : [Doc](https://docs.google.com/document/d/1vSSNuD9VgyW5M3dWR5VwN4N9pNQCWtBQcuDT34zZJrA/edit) |

# Attribution with hidden negative

Author : Anmol

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Understanding by example | Users have a lot of stores at places A1 , A2 , A3 , and represent the profit of each store in some column. (As in most cases he/she represents the profit by +ve value and -ve value represents the loss).   |  |  | | --- | --- | | Store | Profit | | A1 | +50 | | A2 | -50 | | A3 | +50 |   Now intuitively he wants to calculate the proportion of profit made by store A1 so does the *percentage operation* of A1 over Profit , so it turns out to be  which isn’t entirely true as has the intuitive notion of depicting that *all* the Profit is made by this store which is not entirely true.  Foundationally Percentage / Proportion are designed by statisticians to analyse positive numbers to depict what “**Portion”** of data is represented by <dim D> in the set of all <A> things.  In short the reasoning of why we don’t use Percentage / Proportion with negative numbers is the same reasoning of why we don’t use negative numbers to count things. (For example: “Alice has -5 apples” is meaningless) |
| Scope of queries | Compute *Proportion( or Percentage)* of <dim = D> in <metric = A> but <A> contains negative entries as well. |
| Detection | * If user calculates the proportion of some <dim=D> over <A> , but <A> contains some negative entries then just give this oversight |
| Debiasing suggestion | Some entries of <A> are negative so proportion will be flawed because of denominator containing some cancellation of terms |
| False positives |  |

# Duration Averaging

Author : Bhagya

Co-Author : Anmol

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scope of queries | Show Average of metric query.  Example in slide deck -  “At any moment in time, an unemployed person is likely to be unemployed for a long time; but an average unemployment spell is very short. ”  E - Employed  U - Unemployed  Situation -  Person 1 = UUEEUUEEUUEEUU  Person 2 = EEEEUUUUEEEEUU  User Table -   |  |  |  |  | | --- | --- | --- | --- | | Person | Start week | End Week | Unemployment Period | | 1 | 1 |  | 2 | | 1 | 5 |  | 2 | | 1 | 9 |  | 2 | | 1 | 13 |  | 2 | | 2 | 5 |  | 4 | | 2 | 13 |  | 2 |   “Take every person who is unemployed. Calculate mean length of unemployment for them.” = (2 + 2 + 2 + 2 + 4 + 2 ) / 6 = 2.33  “Take every person who loses their job. Calculate mean length of unemployment for them.”\* = ( (2 + 2 + 2 + 2) / 4 + (4 + 2) / 2 ) / 2 = (2 + 3) / 2 = 2.5  // here (2 + 2 + 2 + 2) / 4 is the length of unemployment for person 1 (averaged over all his unemployment spells)  // similarly (4 + 2) / 2 is for person 2  \* Here we calculate the average for each person, then calculate their average.  Query 1 - Show Average : metric - Unemployment period  Query 2 - Show Average : summary\_operator - ‘Average’ group by : ‘Person’ s metric - Unemployment period    We have a bias in our thinking that both Average metric & Average of Average metric group by dimensions should be close by. But as we saw in the above example if a person changes jobs very frequently the result of both differ a lot.  Also, any analyst can have in mind one of the two queries but the difference between both of them is not very obvious, so the user can compute the wrong one & here the bias comes in his mind. |
| Detection | **Approach (1) :** We need to check if both the results differ a lot.  We can use this as a parameter -  Parameter\* = | Average 1 - Average 2 | / ( | Average 1 | + | Average 2 | ) .  \*We used this parameter to check if Average 1 & Average 2 are similar in looking at tails to find causes oversight.  If the parameter exceeds a cut off we classify the situation as an oversight.  The parameter would have it’s own cons (also discussed in looking at tails)   1. It won’t handle cases where Averages ae close to 0 due to -ves 2. Also it won't apply on dates. (Also this oversight won’t make sense on dates)   Reason to NOT implement this approach :   * Finding the correct weights is infeasible without knowing the context of the data. As discussed in the recent meetings as well , considering any column as the weights could lead to different results thus leading to a huge number of false positives.   **Approach (2) :** Give the oversight when *MEAN* is applied on a column which already contained entries which were mean of something else.  Formally , consider the user creates a column of values <A> (assuming using our add-on) which is created by grouping some other column with summary operator as *MEAN* .  Now if the user applies *MEAN* on <A> then it would lead to composition of *MEAN* operator which could lead to asymmetric weighted mean, thus leading to this oversight.  Therefore , whenever a user applied *MEAN* on a column <A> we detect this oversight. |
| Debiasing suggestion | **Suggestion w.r.t Approach (1) :**  If you consider applying group\_by / not applying then the average would change a lot.  OPTION : to see the result for the other query.  **Suggestion w.r.t Approach (2) :**   * Applying *MEAN* on <A> could lead to a very asymmetric result as each row has a different weight while calculating the *MEAN* as <A> is already a *MEAN* column.   <A> = MEAN of <X>   * Applying *Mean* on <A> is very different from applying *Mean* on **<X>** |
| False positives |  |
| Instance | Visualizing cases of duration averaging : [Slides](https://docs.google.com/presentation/d/1Crfqbt21CK8fGnHqkz3zitclNTsroiMOOi7y4tMCwUk/edit#slide=id.g8bc6c61b39_0_14)  Sendhil’s comment   * Both this and the duration averaging slides are exactly right. They're doing two subtly different errors but arise at the same point. This is "length of average unemployment spell" vs "length of unemployment spell for average person" which is GREAT. It is an issue of whether weighting all people equally or weighting those with frequent spells more. * The other is about sampling unemployed people at a point in time vs looking at all people. But again the bias as above in sampling. When we sample at a point in time we are sampling the unemployed more heavily. |

ON HOLD OVERSIGHTS :

# Multiple Benchmarks :

Reason for being ON HOLD : Too many false positives will exist.

Author : Bhagya

|  |  |
| --- | --- |
| Scope of queries | In slice comparison queries when a group of dimensions is used as a filter ( it’s value as benchmark) the other group is used for comparison.  Compare metric in slice1 = ( D1 = V1, D2=V2, …. Dn=Vn , d = VA ) with  slice2 = ( D1 = V1, D2=V2, …. Dn=Vn , d = VB ).  For the given metric it might make more sense to fix d = VA and compare with some Di .  Point relevant to slice compare.   * While comparing, only the slice value of one dimension changes. Slice value\* of other dimensions remain fixed -- as filters.   d= V\_a vs d= V\_b. another choice d=V\_a vs V\_c  \* Slice Value = pair of ( slicing operator, slicing instance ) |
| Detection | Compute the results with changing the ‘d’ - dimension upon which we compare.  If in some value of d, the results differ much from the original results, we predict this as an oversight. |
| Debiasing suggestion | * If you change the benchmark dimensions, the result looks very different. * Also give a suggestion for a new query. |
| False positives |  |

# Anscombe Quartet Error

Reason for being ON HOLD : Correlation is not a common operation that people often do with Sheets

# Prediction cone

Reason for being ON HOLD : Prediction has a lot of problems in itself , so having oversights for it would be too vast of a topic , as prediction is never perfect.

Reference shared by Sendhil to explain the oversight : [https://www.nhc.noaa.gov/aboutcone.shtml](https://www.google.com/url?q=https://www.nhc.noaa.gov/aboutcone.shtml&sa=D&ust=1595967879880000&usg=AFQjCNGwHzFK5jZwlcU2eWggxG19j1bGSw)

# Template

|  |  |
| --- | --- |
| Scope of queries |  |
| Detection |  |
| Debiasing suggestion |  |
| False positives |  |
| Detailed Docs Links |  |