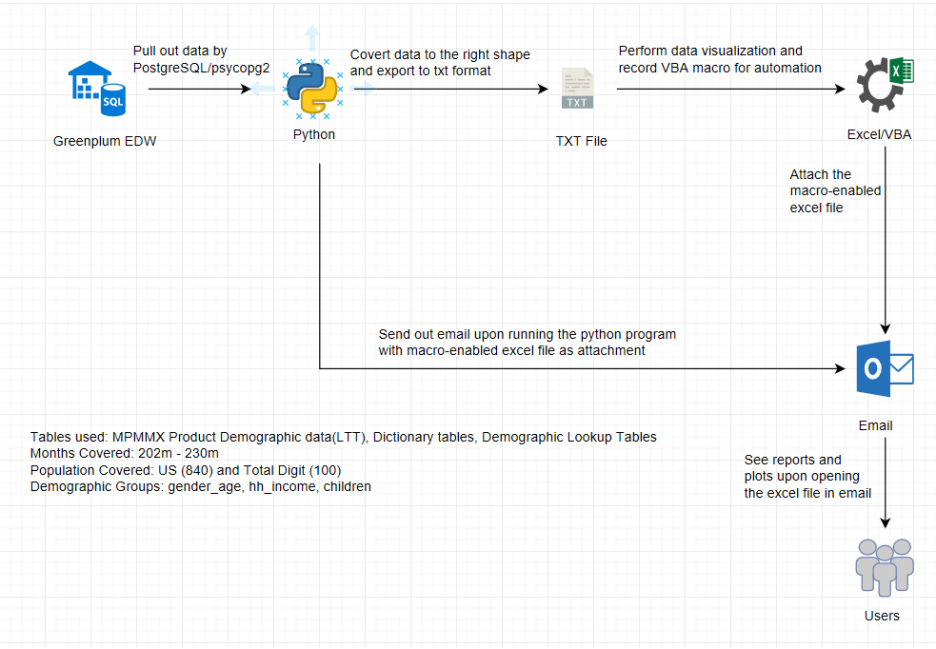
Trend Insight and Anomaly Detection in Demographic Data

Objectives:

- For MMXMP, we hope to generate long-period time-series data/graphs to present the shift in different demographic buckets for entities. Since My Metrix UI can only compare the demographic data for two months, we hope our long-period data/graph can provide some longer-trend or seasonal insights.
- Based on that, we use bollinger bands to identify outliers/anomalies for demographic information for the most recent month on both entity and category level. Also we provide the ranking based on the percentage of reach and deviation to help analysts to prioritize their QA process.
- The entire process is automated in Python and VBA for the future, to run on a monthly basis. Users will receive an email containing the Excel file by running Python program and the Macro VBA embedded in the Excel file will generate all the graphs and tables for users.

Workflow:



Methodology:

All files/code can found in \\CSIADDFS01\SyndicatedOps\MoMX\yunlu&alex

1. Overview:
    - Create Python program to be run monthly that queries/munges proper data sets and writes them to a shared directory.
    - Pull files into excel (from shared directory) and use VBA to create an interactive workbook for visualizations that runs upon opening.
    - Flag potential outliers using bollinger bands and a set threshold.
2. SQL:
    - Basically the SQL code outlines the underlying logic for pulling out time-series entity_level data for different demographic buckets from Greenplum.
    - Create user-defined function to loop through each category to get top 10 entities ranking by unique visitors in WebAgg in the most recent month. Then use while-loop to pull out entity-level and category-level data of three major metrics (UV, Page Views, Duration) and aggregate by web_id, month_id, and according demographic buckets(gender_age_id, hh_income_id, or children_id). Finally merge the table above with dictionary tables to get web_name, category_name, as well as descriptive text for demographic buckets.
    - Since Python is more versatile than SQL and some of our request are restricted by SQL, so we decide to translate the basic logic outlined above to Python to continue our work. Please find more detailed explanation below.
3. Python Program:
    1. Query web_ids of entities ranked top 10 for 'visitors_proj' under each category, obtained from the web_agg table for the current month. Only interested in total internet and US population
        - The heirarchy_id, category_id, category_name and depth are obtained for each web_id by joining the hierarchy_lookup, cat_subcat_map with the full web_agg table where appropriate. Web_ids are then filtered by the following logic: If two web_ids belong to the same hierarchy and category, choose the web_id with the bigger depth (i.e. lower number). This avoids placing multiple copies of the same entities data in the final product, while still allowing for multiple entities from the same hierarchy to appear under their respective categories.
        - Web_ids for full categories are saved in addition to the top 10 for each category
    2. For each demo bucket Query, aggregate, and clean the appropriate data from the mp_mmx_ltt table, only for those web_ids of interest. Apply functions to obtain pct breakdown and bollinger bands.
        - The table is grouped by the web_id, month_id, and the bucket of interest (i.e. hh_income_id, gender_age_id, children_id), and the metrics of interest are summed. This is done month over month, from the specified start month, to the current month. Only the entities that appear every month in the given time span are kept., because we do not want any entity

with missing data for some months. The final product (a list of dataframes, month by month) is concatenated to obtain the full time series data.
        - The time series is joined with its appropriate table to gather descriptions of the demographic segments that belong to the current bucket. This is demographics_lookup for hh_income_id and gender_age_lookup for gender_age_id (none for children). The data is then left joined with the cat_subcat_map to obtain category info, leaving the full categories category_name's to be filled with their respective web_names (since they are not present in the cat_subcat_map).
        - Since the original gender_age group is less insightful, we redefine the age boundary for each group. The gender_age_id bucket is regrouped into wider segments and reaggregated to obtain less demo segments.
        - The data are then grouped by the month_id and web_id to find the entity specific monthly pct breakdown of each demo segment for each metric (e.g. 15% of FB page_views are coming from males ages 2-17, for month 230)
        - The time series is then grouped by the web_id and demographic segments, and rolling 12 month avgs and standard deviations are obtained for each metric percentage, for each demo segment, for each web_id. This is used to define upper and lower bounds for each metric, corresponding to 2 std. deviations above and below the moving avg (bollinger bands).
    3. An email is sent to target address automatically after running through the Python file. The email contains the final macro-enabled excel file.
4. Excel:
    - In Excel, we mainly use pivot-tables and line charts for data visualization, and we rely on VBA to automate the process for running on a monthly basis. For this project, we mainly focus on three kinds of demographic information, namely, household income, gender and age combination, and children. Please use according Excel file for your interested demographic information.
    - The 'raw_data' worksheet stores the data imported from the txt file written out by Python. Several cleaning/formatting steps are done.
    - The 'shift' worksheet contains the pivot-table and line chart for time-series data on entity and category level. Use the filter and slicers to look up the category and the entity that you would like to see. The line chart will show the shift trend over 30 months.
    - The 'anomaly' worksheet contains the pivot-table and line chart(bollinger bands) for time_series data on entity/category and demographic bucket level, which helps to identify the anomaly in the most recent month. Use the filter and slicers to look up the category, the entity, and the demographic bucket that you would like to see. The line chart will show the shift trend as well as the rolling upper and lower bounds for anomaly detection.
    - The 'anomaly_stats" worksheet contains the pivot-table to show the deviation (uv_pct -upper_uv_pct or lower_uv_pct - uv_pct) for each flagged category/entity and demographic bucket.

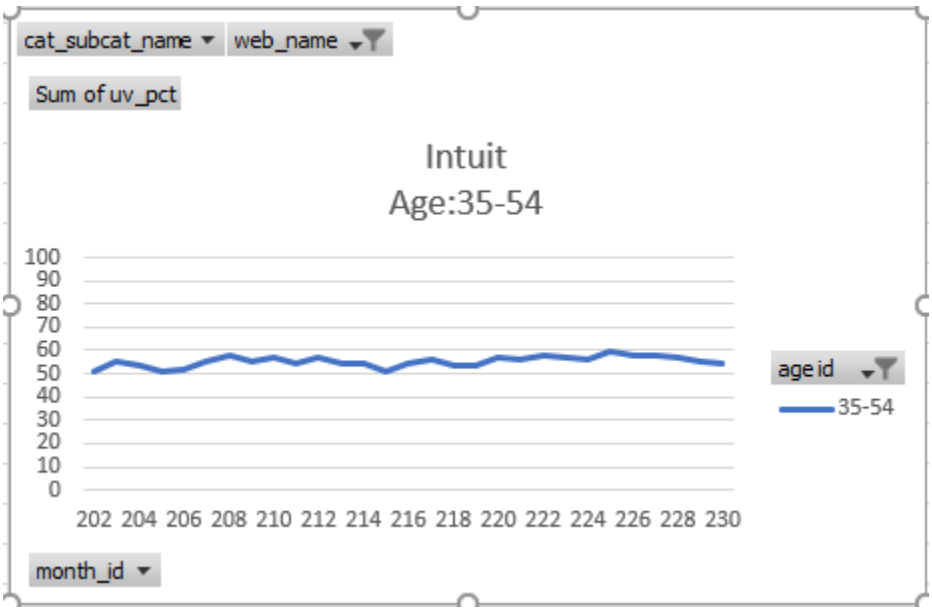
- The 'final_ranking_report' workbook contains the sum of deviation of all three demographic buckets. By multiplying the % reach for each entity/category, we provide the ranking system to prioritize the QA process.
- We record the VBA macro which can generate all tables and graphs mentioned above automatically on a monthly basis. Upon opening "final_ranking_report_macro", users will see the final ranking system as well as graphs for the three demographic buckets.
- There are serval things that user should keep in mind: when you download the file sent by email to your desktop, please do not change the file name; otherwise, the embedded macro will not work. Also, when you would like to save those macro embedded files, please choose "save as" and save them in xlsx format on your own folder, and always make the original xlsm file empty, because otherwise macro code will be ruined. If you accidently save the original xlsm file, please delete it, copy a new/empty one in \\CSIADDFS01\SyndicatedOps\MoMX\yuli\safety_backup and paste into the original file path.

5. Tableau:
   - Graphs are also available in tableau form, though our company currently does not have liscense with Tableau. We may migrate to Looker later.

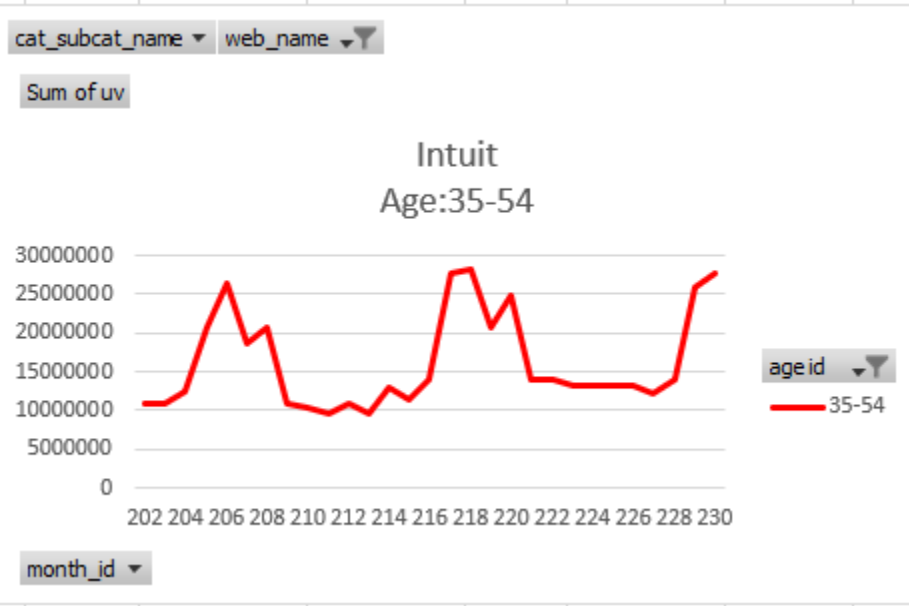
Demo & Insights:

1. Unique Visitor raw number shows a seasonal pattern for most entities under Education, Sports, and Government category, while Unique Visitor percentage shows a less cyclical pattern . The seasonality of UV raw number will cause many false flags for anomalies, so we use UV percentage to counter those false flags in anomaly detection. We can see this from the following example of Intuit/Age: 35-54.

UV percentage:

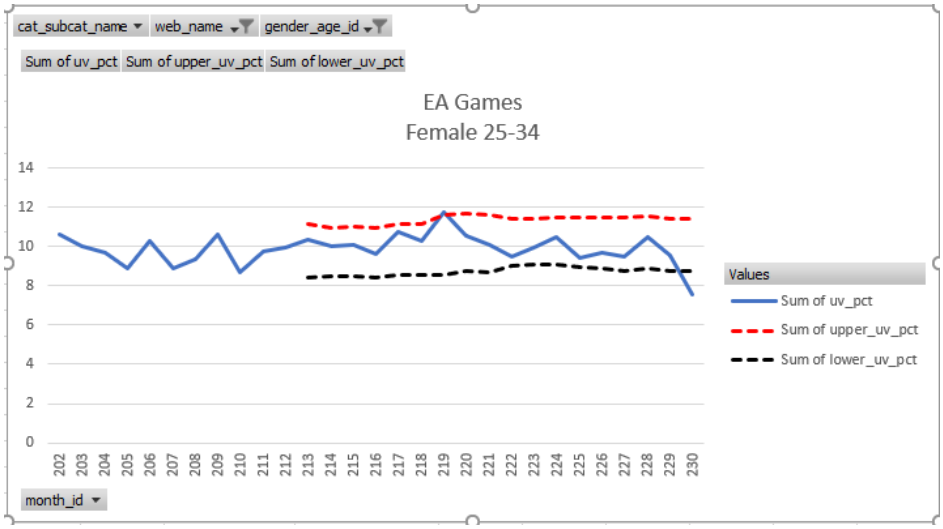


UV raw number:

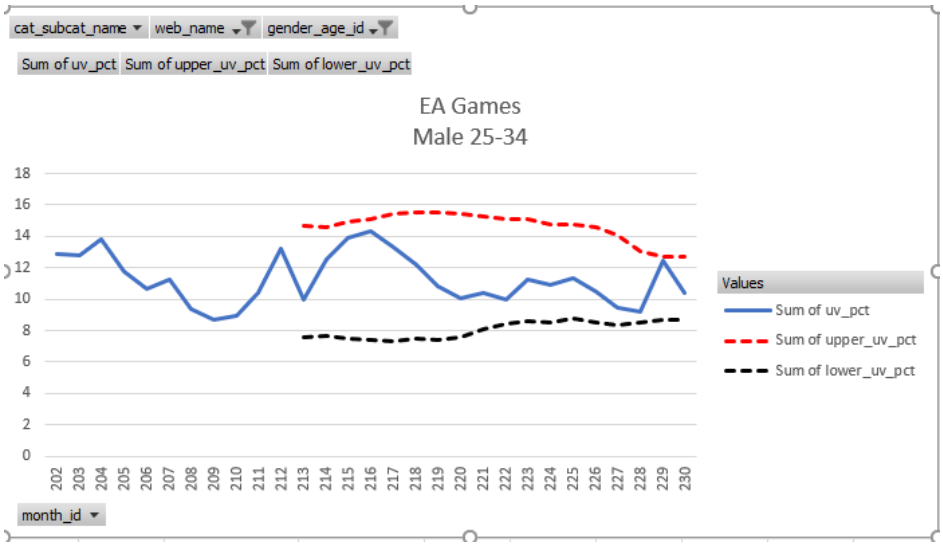


2. Our product could help analysts to visualize the occurrence of anomalies and filter for their relevance. The following graphs show examples of a anomaly and not a anomaly.

Anomaly:



Not a Anomaly:



3. The final ranking system sums up deviation from three demographics for total deviation and multiply it with percentage reach (importance) to get final ranking number. Though Google.com Home Page and EA Games have about the same deviation, their ranking numbers are not close, because of the difference in percentage reach.

	A	B	C	D	E	F	G
1	Column1	gender_age	children	income	sum of deviation	reach_pct	ranking
2	GOOGLE.COM Home Page	6.224033664	3.6926038	3.0639701	12.9806075	0.841422629	10.922177
3	Promotional Servers	9.693267533	0	5.6485439	15.34181139	0.641192941	9.8370612
4	Retail	4.471819274	0	3.4762344	7.948053709	0.850556905	6.760272
5	Telecommunications	8.647669045	0	0.7521083	9.399777318	0.647195648	6.083495
6	Technology	1.927086998	0	6.114282	8.041368955	0.750073409	6.031617
7	Complex Video	1.571220335	0	4.8986088	6.469829148	0.856098376	5.5388102
8	Adobe Advertising Cloud - Potential Reach	0.687614571	0	5.4222528	6.109867383	0.863558122	5.2762256
9	Microsoft Sites	1.821590807	0	4.9455028	6.767093597	0.7786291	5.269056
10	MNI Targeted Media	1.323928731	0	4.1898848	5.513813518	0.897005551	4.9459213
11	Social Media	1.034078717	0	4.2918401	5.325918853	0.874852449	4.6593932
12	Search/Navigation	3.109775829	0	2.1092026	5.218978431	0.888855913	4.6389198
13	GOOGLE.COM	1.374111116	0	3.5564539	4.930565042	0.883662491	4.3569554
14	Corporate Presence	1.441515013	0	3.3758379	4.81735291	0.901540579	4.3430391
15	Google Sites	1.698818161	0	2.2244665	3.923284701	0.962756592	3.7771682
16	Goodway Group	1.78990305	0	2.3893417	4.17924476	0.90157429	3.7678996
17	Services	2.189332551	0	1.7631725	3.952505065	0.950340929	3.7562273
18	Conde Nast Digital	5.069431246	3.7691753	1.2310907	10.06969731	0.360542813	3.630557
19	Directories/Resources	1.173506341	0	2.6726075	3.846113792	0.843594734	3.2445613
20	smartclip Video Advertising Platform - Potential Reach	1.636402945	0	2.0020869	3.638489875	0.878207722	3.1953499
21	Teads - Potential Reach	0	0	3.8025499	3.802549886	0.838828881	3.1896887
22	BING.COM	2.491774348	0	3.6319186	6.123692914	0.519636441	3.182094
23	Entertainment	1.516714256	0	1.8003969	3.317111147	0.93267394	3.0937831
24	News Media Alliance	2.798120864	0	1.4560742	4.254195054	0.692773545	2.9471938
25	Automotive	5.558634567	0	0	5.558634567	0.521221442	2.8972795
26	Financial Services	0	0	3.4032891	3.403289097	0.816630334	2.7792291
27	Family & Youth	0.396525829	0	4.0133064	4.409832197	0.617712933	2.7240104
28	Google Display Network (Video Sites) - Potential Reach	0	0	2.6517054	2.651705374	0.89150867	2.3640183
29	PayPal	4.117237871	0	0	4.117237871	0.547419532	2.2538564
30	Distributed Content	0.624618007	0	2.2601899	2.884807955	0.755915872	2.1806721
31	Games	0	0	2.4806493	2.480649313	0.830512033	2.0602091
32	Wal-Mart	3.005211076	1.2263018	0.3767	4.608212855	0.408495661	1.882435
33	Amazon Sites	2.34991842	0	0.0250116	2.374929996	0.783374887	1.8604605
34	YOUTUBE HomePage	2.877621996	0	0	2.877621996	0.638235411	1.8366003
35	Kargo Publisher Platform	2.342778589	0.5440468	0	2.886825378	0.615117629	1.7757372
36	EA Games - Media Network	11.5619526	3.8893478	0.058051	15.50935145	0.100088675	1.5523104
37	Snapchat, Inc	1.113620337	0.7565149	1.4483032	3.318438457	0.462821182	1.5358436

Business Impact:

- Compared to MyMetrix UI which only shows demographics for two consecutive months, our product will present demographic information covering more than 24 months, providing long-trend/seasonal insights both for customers and analysts.
- Provide a new level for QA process: Analysts now are able to proceed QA on demographic information; The ranking system help to see the summary of anomalies and to prioritize their focus
- Proactive instead of reactive – ComScore appears more confident to clients

Future Improvements:

- Design and implement more reliable algorithm for calculating the upper and lower bounds in Bollinger bands.
- Migrate the data visualization from Excel to tools that our company has license with, i.e. Looker.
- Expand the usage of the project to other matrices, entities, countries, and platforms following the same logic.
- As for the final ranking, we should use weighted sum of deviation. 2-17 age group usually contributes a lot to deviation but is not as important as other groups.