Technical Documentation: SecureNet VPN Plan Order A/B Test

1. Introduction & Business Context

1.1. Purpose of this Document

This document provides a comprehensive technical guide to the A/B/C test conducted to evaluate the impact of different subscription plan orderings on the SecureNet VPN pricing page. It is intended to serve as a complete manual for any analyst, data scientist, or technical stakeholder, explaining the entire analytical pipeline from raw data ingestion to the final strategic recommendations.

1.2. Experimental Hypothesis

The core business question was whether a simple UI change could positively influence user purchasing decisions.

- **Hypothesis:** Changing the display order of subscription plans will cause a statistically significant change in the paid user conversion rate.
- **Control (Variant A):** 1/24/12 month order
- **Treatment (Variant B):** 1/12/24 month order
- **Treatment (Variant C):** 24/12/1 month order

The ultimate result was that the null hypothesis could not be rejected.

2. Analytical Pipeline & Code Architecture

The analysis is executed via the python script, designed as a sequential pipeline to ensure a logical flow from data cleaning to final reporting.

1. **Setup & Configuration:** Global settings for visualizations.
2. **Data Ingestion & Validation (load_and_validate_data):** Loads raw data and handles split sessions.
3. **Data Cleaning & Validation (detect_and_remove_outliers,validate_outlier_removal):** Removes unwanted traffic and validates the process for bias.
4. **Data Transformation (filter_dataset_to_1_12_24_only):** Focuses the dataset on relevant plans.
5. **Statistical Analysis & Business Logic:** Core functions that evaluate the experiment.
6. **Reporting & Visualization:** A suite of functions that generate all visual assets.

3. Data Processing & Validation

3.1. Data Ingestion and Consolidation

- **Function:** load_and_validate_data()
- **Rationale:** The raw dataset contained sessionId values split across multiple rows. This function corrects that issue to ensure each session is a single, unique data point.
- **Process:** The function identifies duplicated sessionIds, removes any with inconsistent dimensional data (e.g., different flags for the same session), and consolidates the valid split sessions by summing metrics.

3.2. Outlier Detection and Bias Check

- **Functions:** detect_and_remove_outliers(), validate_outlier_removal()
- **Rationale:** To prevent non-human bot traffic from skewing the results, a conservative cleaning filter is applied.
- **Process:** The methodology targets only clear bot patterns (impossible browsing speed, zero-duration sessions with activity). A subsequent validation step confirmed the removed population had a **0.0000% conversion rate**, proving the process was sound and did not introduce bias.

4. Statistical Methodology & Rationale

4.1. Key Performance Indicators (KPIs)

- **Primary KPI (Paid Conversion Rate):** The percentage of sessions resulting in a paid subscription. This was chosen over free signups to align directly with revenue goals.
- **Secondary KPIs:** Average Revenue Per User (ARPU) and Subscription Plan Mix.

4.2. Statistical Framework Rationale

- **Z-test for Proportions:** Used to compare conversion rates due to its appropriateness for large, independent samples.
- **Bonferroni Correction:** The significance threshold was adjusted to **$p < 0.025$** to control for the increased risk of false positives (Type I errors) from making multiple comparisons.
- **Statistical Power:** The test achieved **99.9%** power, providing high confidence that the null result is accurate.
- **Confidence Intervals:** Wilson score intervals were used for their accuracy with proportion data.

5. Reporting & Visualization Suite

The script generates a comprehensive suite of visualizations to communicate findings, including primary conversion rates, plan mix distribution, revenue (ARPU), user funnels, and segmentation lift. The rationale for each is detailed in the script's docstrings.

6. Core Functions & Implementation Logic

This section details the logic within key functions that translate raw statistical outputs into business-focused metrics and decisions.

6.1. Data Transformation & Filtering

Function: parse_and_filter_subscriptions()

- **Purpose:** This is a helper function designed to handle the messy, semi-structured data within the subscriptionsTotalByCycle column.
- **Process:** The function can parse data that is either a legitimate Python dictionary or a string representation of a dictionary. It uses ast.literal_eval for safe string evaluation. It then iterates through the keys, using string matching ('12', 'annual', etc.) to identify and count only the 1, 12, and 24-month plans.
- **Rationale:** This robust parsing is necessary because the raw data is not clean. This function ensures that only plans relevant to the test hypothesis are included in the final analysis, preventing other subscription types from contaminating the results.

Function: filter_dataset_to_1_12_24_only()

- **Purpose:** This function applies the parsing logic to the entire dataset to create a new, clean DataFrame for analysis.
- **Process:** It uses pandas' .apply() method to run parse_and_filter_subscriptions on every row. It then overwrites the subscriptionsTotal and subscriptionsTotalByCycle columns with the filtered, cleaned data.
- **Rationale:** Creating a separate, filtered DataFrame (df_filtered) is a best practice. It ensures that the primary analysis is conducted only on data relevant to the hypothesis, while the original cleaned data (df_clean) remains available for other analyses, like the funnel plot which includes free signups.

6.2. Statistical Utility Functions

Function: calculate_effect_size()

- **Purpose:** To calculate Cohen's h, a measure of effect size for proportions.

- **Process:** It takes the success counts and totals for control and treatment groups, converts them to proportions, and applies the arcsine transformation.
- **Rationale:** While the p-value tells us if an effect is statistically significant (i.e., likely not due to random chance), it doesn't tell us the *magnitude* of the effect. Cohen's h provides a standardized measure of this magnitude. It helps answer, "Even if the result was significant, was the difference large enough to be meaningful for the business?" In this case, the very small effect sizes (e.g., -0.002) confirmed the practical insignificance of the results.

Function: run_power_analysis()

- **Purpose:** To determine if the experiment collected enough data to reliably detect a meaningful difference.
- **Process:** The function uses the statsmodels library to calculate the statistical power given the actual sample sizes, a baseline conversion rate, and a pre-defined Minimum Detectable Effect (MDE).
- **Rationale:** This is a critical step in validating an experiment with a null result. A low-power test that finds no difference is inconclusive: the lack of a result could be due to an insufficient sample size. By confirming a power of **99.9%**, this function provides high confidence that our null result is a true reflection of user behaviour, not a methodological flaw.

6.3. Business Logic & Decision Functions

Function: calculate_business_impact()

- **Purpose:** To provide a simple, directional estimate of the impact of the best-performing *treatment* variant compared to the control.
- **Process:** It calculates conversion rates for all groups, identifies the best-performing treatment variant, and then multiplies the difference in conversion rate by that variant's sample size to project the number of "additional conversions."
- **Rationale:** This translates abstract percentages into a tangible number that is easier for stakeholders to understand (e.g., "Variant C would have resulted in approximately 21 fewer conversions").

Function: make_recommendation()

- **Purpose:** To automate the final recommendation based on a predefined set of business rules.
- **Process:** The function recommends implementing a variant only if two conditions are met: 1) the result is statistically significant, and 2) the conversion lift is positive.

- **Rationale:** This logic prevents recommendations based on random noise. A result might show a positive lift but not be statistically significant, meaning it could be due to chance. The function requires both statistical certainty and a positive business outcome before recommending a change.