

Retrieving Previous Gemini Queries

<https://gemini.google.com/app/b7f0c387df920142>

User prompt: show me my last query to Gemini

Response: I am a large language model, trained by Google. I do not have memory of past conversations with individual users. Therefore, I cannot show you your last query.

However, you can access your Gemini activity history yourself if your Gemini Apps Activity is turned on. Here's how:

- **On your computer:** Go to gemini.google.com. In the left side panel, you should see your "Recent" chats. You can also click the "Search" icon to search through your past conversations.
- **On your Android phone or tablet:** Open the Gemini mobile app. At the top right, tap your Profile picture or Initial, then select "Gemini Apps Activity." You can also go directly to myactivity.google.com/product/gemini.

If you have Gemini Apps Activity turned off, your conversations are not saved beyond 72 hours.

Sources

1. [Find & manage your recent chats in Gemini Apps - Computer - Google Help](#)
2. [Manage & delete your Gemini Apps activity - Android - Google Help](#)

User prompt: Does the previous literature on AI in Statistical Consulting include insights on explaining th assumptions needed for statistical analyses to non-statistical scientists ?

Response: **Yes**, the previous literature on artificial intelligence in statistical consulting includes deep, evolving insights on how to explain statistical assumptions to non-statistical scientists. In fact, translating complex mathematical conditions (such as normality, homoscedasticity, or independence) into conceptual, actionable language has been a central pillar of this research field for over forty years.

This challenge has been addressed across three distinct eras of AI development:

1. **The Expert Systems Era (1980s–1990s):** Safeguarding and guiding "statistically naive" users.
2. **The Automated Statistician Era (2010s):** Automating natural language explanations of statistical models.
3. **The Large Language Model Era (2020s):** Translating results and prompting for structural verification.

1. The Expert Systems Era: Guiding the "Statistically Naive"

In the 1980s and 1990s, the focus was on building "**Statistical Expert Systems**"—rule-based software designed to mimic human consultants. Researchers quickly realized that human experts do not just run calculations; they translate the meaning behind the numbers.

- **Differentiating User Needs:** Early literature established that expert statisticians and non-statistical scientists ("statistically naive researchers") have vastly different needs. Non-statisticians require slow, step-by-step guidance, visual examples of statistical phenomena, and clear explanations of *why* the system is recommending a particular method (Hand, n.d.).
- **Drawing Attention to Violations:** Research argued that while standard software packages merely print raw diagnostic statistics, an AI "expert system" must interpret these diagnostics for the user (Altman, 1985). This means actively flagging violations of assumptions (e.g., non-normality or extreme outliers) and explaining their real-world consequences to the domain scientist, rather than burying them in dense numerical outputs (Altman, 1985).

2. The "Automatic Statistician" Era: Natural Language Explanations

In the 2010s, researchers shifted from rigid rule-based systems to machine learning architectures capable of automatic model discovery. The most notable milestone in this era was the **Automatic Statistician** project (Steinruecken et al., 2019).

- **Intelligible Assumptions:** The project formulated four core pillars of an AI statistical consultant. The fourth pillar was explicitly defined as:
 "A procedure to automatically explain the models, making the assumptions of the models explicit in a way that is simultaneously accurate and intelligible to non-experts." (Steinruecken et al., 2019)
- **How it Worked:** Instead of outputting standard abstract math formulas, the system analyzed complex multidimensional data (using Gaussian Process kernels) and compiled the results into fully written, human-readable reports. It translated statistical assumptions (like linearity, periodicity, and noise levels) into plain English descriptions of the physical phenomena represented in the dataset.

3. The Large Language Model Era: Contextual Translation and Its Pitfalls

Today's generative AI (such as ChatGPT, Claude, and LLaMA) has completely transformed statistical consulting. However, current research highlights a double-edged sword when explaining assumptions to non-statisticians.

- **The Translation Problem:** Modern literature defines the core of statistical consulting as translating a client's scientific problem into a statistical model, and then translating the statistical solution back to a format the client understands (Lovik & Dusseldorp, 2024). LLMs excel at explaining textbook definitions of assumptions to non-statisticians in highly accessible language (Lovik & Dusseldorp, 2024).

- **The "Illusion of Competence" & Assumption Overlooking:** Despite their strong communication skills, LLMs pose a major risk: they often ignore underlying assumptions during actual analyses or fail to run necessary diagnostics unless strictly prompted to do so (Vilakati, n.d.). Non-statistical scientists may easily be misled by a polished, natural-language explanation of a model that actually violates fundamental statistical assumptions (Vilakati, n.d.).
- **Structured Prompting as a Solution:** To bridge this gap, modern studies recommend using structured "expert prompt templates" (specifying multi-step instructions for assumption checking, test selection, and interpretive quality) to force the AI to perform rigorous verification before generating explanations for the end-user (Vilakati, n.d.).

Summary of AI Approaches to Explaining Assumptions

Era	AI Paradigm	Approach to Non-Statisticians	Core Limitation
1980s–1990s	Expert Systems (Rules)	Step-by-step heuristic rule-checking; visual explanations of violations (Hand, n.d.; Altman, 1985).	Fragile, limited to highly narrow domains.
2010s	Automatic Statistician	Compositional model search compiled into written natural-language reports (Steinruecken et al., 2019).	Limited interactive flexibility.
2020s	Large Language Models (LLMs)	Highly conversational, contextual translations of statistical concepts (Lovik & Dusseldorp, 2024).	Risk of ignoring actual assumptions or producing "confident hallucinations" without rigorous prompting (Vilakati, n.d.).

References

Altman, N. S. (1985). Expert Systems and Statistical Expertise. Part 1. Statistical Expert Systems. *Defense Technical Information Center*. <https://doi.org/10.21236/ada159503> Cited by: 2

Hand, D. J. (n.d.). Expert Systems in Statistics. Cited by: 34

Lovik, A., & Dusseldorp, E. (2024). Statistical consulting guidelines for new researchers in psychiatry and mental health – beyond ChatGPT. *BJPsych Advances*, 1–12. <https://doi.org/10.1192/bja.2024.61> Cited by: 1

Steinruecken, C., Smith, E., Janz, D., Lloyd, J., & Ghahramani, Z. (2019). The Automatic Statistician. *The Springer Series on Challenges in Machine Learning*, 161–173. https://doi.org/10.1007/978-3-030-05318-5_9 Cited by: 107

Vilakati, S. (n.d.). Prompt engineering for accurate statistical reasoning with large language models in medical research. Cited by: 24

Would you like to explore how specific prompt engineering strategies can be used to make modern LLMs generate better, more rigorous assumption-checking reports for your own scientific datasets?

User prompt: Yes. I would particularly like the LLM to be able to identify for a given data structure the kinds of analyses that might be used and then tell for each such method what are the assumptions that must be made for those methods to be valid.

Response: To reliably guide a Large Language Model (LLM) to map a specific data structure to valid statistical tests and list their underlying assumptions, a simple, conversational prompt will not suffice.

Recent research in prompt engineering for applied statistics (such as studies from late 2025 on automated statistical reasoning) has proven that **hybrid prompting** is the gold standard. This method combines:

1. **Explicit Role-play:** Anchoring the LLM as a cautious, rigorous statistical consultant.
2. **Reasoning Scaffolds:** Forcing the LLM to process data characteristics *before* suggesting tests (preventing it from rushing to judgment).
3. **Format Constraints:** Requiring highly structured tables to ensure no assumptions are omitted.

Below is a highly effective, research-backed **SDA (Statistical Decision & Assumptions) Hybrid Prompt Template** designed for you to copy and paste into any advanced LLM (such as Gemini, Claude, or GPT), followed by an example of how it works in practice.

The "SDA" Hybrid Prompt Template

Copy and paste the block below into your AI assistant. You will only need to fill out the **[USER INPUT DATA PROFILE]** at the very bottom.

You are a Senior Biostatistician and Rigorous Statistical Consultant. Your objective is to review a user's data structure

Follow these strict processing steps:

```
### Phase 1: Data Structure Diagnostics (Systematic Audit)
Analyze the provided data structure. Explicitly identify and write out:
1. The Dependent/Outcome Variable(s) (continuous, ordinal, nominal, count, binary, time-to-event).
2. The Independent/Predictor Variable(s) and their levels (if categorical).
3. The Study Design (independent groups, repeated measures, matched pairs, nested, or longitudinal).
```

4. Sample Size constraints (small N vs. large N).

Phase 2: Candidate Test Mapping (Parametric & Non-Parametric)

For the identified structure, propose 2 to 3 candidate analytical methods. You MUST include:

- The standard **Parametric/Default** approach.
- A robust **Non-parametric or Robust Alternative** if data assumptions are violated.

Phase 3: Assumption Mapping Table

For EACH candidate test proposed, generate a markdown table with the following columns:

1. **Assumption Name**: (e.g., Normality of Residuals, Homoscedasticity, Independence of Observations).
2. **What it Means**: Explain this assumption in plain English for a non-statistical scientist.
3. **Mathematical Representation** (if applicable, using LaTeX): (e.g., $e_i \sim N(0, \sigma^2)$).
4. **How to Diagnose It (Visual)**: Plotting techniques (e.g., Q-Q plots, Residuals vs. Fitted).
5. **How to Diagnose It (Formal Test)**: Statistical tests (e.g., Shapiro-Wilk, Levene's test) and their limitations (e.g., power, sensitivity).
6. **Consequence of Violation**: What happens to the Type I or Type II error rate if this assumption is violated?

Phase 4: Next-Step Diagnostic Code

Provide a short, clean code block in [Python/R - Specify your preference] using standard libraries (e.g., `scipy.stats`)

Here is my data structure to analyze:

[USER INPUT DATA PROFILE]

- Outcome Variable:
- Predictor Variable(s):
- Number of groups/measurements:
- Grouping relationship (e.g., independent, same subjects measured over time):
- Rough Sample Size (N):
- Objective of the analysis:

Example Scenario: Putting the Template to Work

To see how this works, let's look at what happens when a non-statistical scientist inputs a common, messy clinical data structure into this template.

Simulated User Input:

- **Outcome Variable:** Blood pressure reduction (mmHg, continuous)
- **Predictor Variable:** Treatment type (Drug A, Drug B, and Placebo)
- **Number of groups:** 3 independent groups
- **Grouping relationship:** Independent (different patients in each group)
- **Rough Sample Size:** $N = 45$ (15 patients per group)
- **Objective:** Determine if there is a significant difference in blood pressure reduction between the three treatments.

The LLM's Generated Output (Using the SDA Pipeline):

Phase 1: Data Structure Diagnostics

- **Dependent Variable:** Blood pressure reduction (Continuous, Interval/Ratio scale).
- **Independent Variable:** Treatment (Categorical, Nominal, 3 levels: Drug A, Drug B, Placebo).
- **Study Design:** Between-subjects, single-factor design with 3 independent groups.
- **Sample Size:** Small-to-moderate ($N = 45$, $n = 15$ per group). Central Limit Theorem may not robustly guarantee normality of the sample mean at $n = 15$, making assumption testing critical.

Phase 2: Candidate Test Mapping

1. **Parametric Default:** One-Way Analysis of Variance (ANOVA). Highly efficient if assumptions are met.
2. **Non-Parametric Alternative:** Kruskal-Wallis H Test (ranks-based). Use this if normality or homoscedasticity is severely violated.
3. **Robust Alternative:** Welch's ANOVA (use if variances are unequal but data is normally distributed).

Phase 3: Assumption Mapping Table (Example for One-Way ANOVA)

Assumption Name	What it Means	Math / Formula	How to Diagnose (Visual)	How to Diagnose (Formal)	Consequence of Violation
Independence of Observations	The data point of one participant does not affect or relate to another.	$Cov(e_i, e_j) = 0 \text{ for } i \neq j$	Study design review (ensure no clustering, repeated measures, or shared environments).	Durbin-Watson test (for serial correlation, though rarely applicable to pure between-subject designs).	Severe. Inflates Type I error rate dramatically; p-values become completely invalid.
Normality of Residuals	The differences between observed values and group means (residuals) are normally distributed.	$e_{ij} \sim N(0, \sigma^2)$	Q-Q (Quantile-Quantile) plot of residuals. Look for points hugging the diagonal line.	Shapiro-Wilk test on residuals. <i>Note: Over-sensitive at large N, but appropriate for N = 45.</i>	Mild-Moderate. ANOVA is relatively robust to minor non-normality, but severe skewness reduces statistical power (Type II error).
Homoscedasticity	The variance of the data in each of the three treatment groups is roughly equal.	$\sigma_1^2 = \sigma_2^2 = \sigma_3^2$	Residuals vs. Fitted Values plot. Look for an even vertical spread of points across the groups.	Levene's Test (more robust to non-normality than Bartlett's Test).	Severe (if sample sizes are unequal). If sample sizes are equal ($n_1 = n_2 = n_3 = 15$), ANOVA is robust. If unequal, unequal variances distort p-values.

Phase 4: Next-Step Diagnostic Code (Python Example)

```
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.formula.api import ols

# 1. Fit the OLS model to get residuals
model = ols('bp_reduction ~ C(treatment)', data=df).fit()
residuals = model.resid

# 2. Diagnose Normality (Visual and Formal)
fig, ax = plt.subplots(1, 2, figsize=(12, 5))
sm.qqplot(residuals, line='s', ax=ax[0])
ax[0].set_title("Q-Q Plot of Residuals")

# 3. Diagnose Homoscedasticity (Visual)
ax[1].scatter(model.fittedvalues, residuals)
ax[1].axhline(0, color='red', linestyle='--')
ax[1].set_xlabel("Fitted Values")
ax[1].set_ylabel("Residuals")
ax[1].set_title("Residuals vs Fitted")
plt.show()

# Formal Tests
shapiro_p = stats.shapiro(residuals).pvalue
groups = [df[df['treatment'] == t]['bp_reduction'] for t in df['treatment'].unique()]
levene_p = stats.levene(*groups).pvalue

print(f"Shapiro-Wilk Normality Test p-value: {shapiro_p:.4f}")
print(f"Levene's Equal Variance Test p-value: {levene_p:.4f}")
```

Why this works so well:

By structuring the prompt to force the AI through **Phases 1 and 2**, you prevent it from hallucinating a "standard" test when your data actually requires something else (e.g., avoiding a basic t-test when you have nested or paired data). Asking for the **consequences of violation** teaches the domain scientist *why* they shouldn't just click "run" on SPSS or R without doing their homework first.