

Excel AI Assistant (AI Name)

Excel AI Assistant

Introduction

Excel AI Assistant lets you run quick analyses and create charts using plain English commands.

Always wrap column names in braces { }. Use where to filter before an operation. Results are placed back into Excel (or returned as images).

Format:

```
<operation> {columnA} and {columnB} where {columnC} is value
```

This project is designed using sales data and works perfectly on excel file having rows <= 1 lakh.

What it *can* do

- Basic aggregations and statistics (sum, mean, median, min, max, std).
 - Charts: bar, line , scatter, box, histogram, pie, heatmap, scatter-matrix.
 - Regression: linear, polynomial (degree 2), multiple (multi-variable), log.
 - Filtering with where {column} is value.
 - Money-aware axis formatting and top-N grouping for categories.
 - Sampling for large data (keeps plotting responsive).
-

What it *cannot* do (Must Read Before Using)

- Advanced forecasting/time-series modeling (ARIMA, Prophet).
- **Not suitable for working with dates**
- Fuzzy column-name matching — braces must match exactly.
- Full ML training or multi-output regression beyond basic linear/multiple fits.
- Text / NLP analysis of cell contents.

NOTE: If you place statistic result in same file that has columns, the AI will show error for further calculations. You can save all of your result in new sheet in same file.

FILTERING (`where`) — usage examples (short)

```
sum {numeric} where {category} is Value
sum {numeric} where {category} = Value (not use ==)
sum {numeric} where {category} < Value
sum {numeric} where {category} > Value
mean {numeric} where {numeric_column} is 2020
bar chart of {category} and {numeric} where {region} is West
```

Notes: `where` supports text, numeric. Use `where {column} is value`.



STATISTICS & AGGREGATION

Use these commands to **calculate numeric results** from your Excel data.
All operations support optional **where filtering**.

◆ General Rules

- Works **only on numeric columns**
 - Column names **must be inside {}**
 - `where` filters are applied **before calculation**
 - Result is written back to Excel
-

SUM

Adds all numeric values in a column.

Basic

```
sum {numeric_column}
```

With filtering

```
sum {numeric_column} where {category_column} is value (Can be
Categorical)
sum {numeric_column} where {region_column} is East
```

MEAN

Calculates average value.

Basic

```
mean {numeric_column}
```

With filtering

```
mean {numeric_column} where {category_column} is value  
mean {numeric_column} where {region_column} is West
```

AVERAGE

Alias of `mean`.

Basic

```
average {numeric_column}
```

With filtering

```
average {numeric_column} where {category_column} is value  
average {numeric_column} where {region_column} is South
```

MEDIAN

Returns the middle value.

Basic

```
median {numeric_column}
```

With filtering

```
median {numeric_column} where {category_column} is value  
median {numeric_column} where {region_column} is Europe
```

MIN

Finds the smallest value.

Basic

```
min {numeric_column}
```

With filtering

```
min {numeric_column} where {category_column} is value  
min {numeric_column} where {region_column} is Asia
```

MAX

Finds the largest value.

Basic

```
max {numeric_column}
```

With filtering

```
max {numeric_column} where {category_column} is value  
max {numeric_column} where {region_column} is North
```

STANDARD DEVIATION (`std`)

Measures spread of data.

Basic

```
std {numeric_column}
```

With filtering

```
std {numeric_column} where {category_column} is value  
std {numeric_column} where {region_column} is Central
```

VARIANCE

Measures variability (square of std).

Basic

```
variance {numeric_column}
```

With filtering

```
variance {numeric_column} where {category_column} is value  
variance {numeric_column} where {region_column} is East
```

COUNT

Counts number of valid rows (non-null).

Basic

```
count {column}
```

With filtering

```
count {column} where {category_column} is value  
count {column} where {region_column} is West
```

MODE

Value appears most frequently in a dataset.

Basic

```
mode {numeric_column}
```

With filtering

```
mode {numeric_column} where {category_column} is value  
mode {numeric_column} where {region_column} is Europe
```

RANGE

Difference between max and min.

Basic

```
range {numeric_column}
```

With filtering

```
range {numeric_column} where {category_column} is value  
range {numeric_column} where {region_column} is Europe
```

COMPARE (CORRELATION)

Difference between max and min.

Basic

```
compare {column_1} and {column_2}
```

With filtering

```
compare {column_1} and {column_2} where {category_column} is value  
compare {column_1} and {column_2} where {region_column} is Europe
```

SUMMARY STATISTICS

Single, simple numbers that describes the main features of a dataset.

Basic

```
summary {numeric_column}
```

With filtering

```
summary {numeric_column} where {category_column} is value  
summary {numeric_column} where {region_column} is Europe
```

✗ What STATISTICS cannot do

- Cannot operate on text columns
 - Cannot compute rolling / window stats
 - Cannot compare more than two numeric columns in one command
 - Cannot compute conditional logic beyond `where {column} is value`
-

✓ Best Practices

- Clean numeric columns before running stats
 - Use `where` to reduce noise
 - Avoid ID columns for statistics
 - Dates must be real Excel dates or clean text dates
-

CHARTS (operation heading → examples + short note)

BAR CHART

Use: compare categories or time-aggregated numeric totals.

Basic:

```
bar chart of {category_column} and {numeric_column}
```

Where-filters:

```
bar chart of {category_column} and {numeric_column} where  
{region_column} is West  
bar chart of {category_column} and {numeric_column} where  
{category_column} is "Top Sellers"
```

Notes: If X is date → groups by year (multi-year) or month (single-year). High-cardinality categories are truncated to top N.

LINE CHART (Time Series)

Use: trend over time.

Basic:

```
line chart of {numeric_column} and {numeric_column}
```

Where-filters:

```
line chart of {numeric_column} and {numeric_column} where  
{region_column} is East  
line chart of {numeric_column} and {numeric_column} where  
{category_column} is Office
```

Notes: X must be time-like. If not recognized, the assistant will attempt to detect date columns.

SCATTER PLOT

Use: relationship between two numeric variables.

Basic:

```
scatter plot of {numeric_column1} and {numeric_column2}
```

Where-filters:

```
scatter plot of {numeric_column1} and {numeric_column2} where  
{category_column} is Widgets  
scatter plot of {unit_cost} and {unit_price} where {region} is Europe
```

Notes: Both columns must be numeric. Money formatting applied to axes when appropriate.

BOX PLOT

Use: distribution + outliers; single column or grouped by category.

Basic (single):

```
box plot of {numeric_column}
```

Grouped:

```
box plot of {numeric_column} by {category_column}
```

Where-filters:

```
box plot of {numeric_column} by {category_column} where {region} is West  
box plot of {numeric_column} where {numeric_column} < 2020  
box plot of {numeric_column} by {category_column} where {category_column} is "Top 5"
```

HISTOGRAM

Use: distribution shape.

Basic:

```
Histogram plot of {numeric_column}
```

Where-filters:

```
Histogram plot of {numeric_column} where {region} is East  
Histogram plot of {numeric_column} where {numeric_column} < 2019  
Histogram plot of {numeric_column} where {category_column} is Electronics
```

PIE CHART

Use: part-to-whole proportions.

Basic:

```
pie chart of {category_column} and {numeric_column}
```

Where-filters:

```
pie chart of {category_column} and {numeric_column} where {region} is West  
pie chart of {category_column} and {numeric_column} where {numeric_column} > 2020  
pie chart of {category_column} and {numeric_column} where {category_column} is Top
```

HEATMAP (Correlation)

Use: correlation among numeric columns.

Basic:

```
Heatmap plot
```

With selection:

```
Heatmap plot of {numeric_column1}, {numeric_column2},  
{numeric_column3}  
Heatmap plot where {category_column} is Electronics  
Heatmap plot where {numeric_column} > 2020
```

Notes: Automatically picks numeric columns when not specified.

SCATTER MATRIX

Use: pairwise relationships across many numeric columns.

Basic:

```
scatter matrix
```

Filtered:

```
scatter matrix where {region} is East  
scatter matrix where {category} is Furniture  
scatter matrix where {numeric_column} is 2021
```

Notes: Sampling applied for large datasets.

REGRESSION & FITS (each method as heading)

REGRESSION (Simple Linear)

Basic:

```
regression of {numeric_x} and {numeric_y}
```

Where-filters:

```
regression of {numeric_x} and {numeric_y} where {category} is  
Electronics  
regression of {numeric_x} and {numeric_y} where {numeric_column} <  
2020  
regression of {numeric_x} and {numeric_y} where {region} is West
```

Output: slope, intercept, R² and scatter plot with fit line.

POLYNOMIAL REGRESSION

Basic:

```
polynomial regression of {numeric_x} and {numeric_y}
```

Where-filters: same where variations as above.

Notes: attempts polynomial fit and reports metrics.

MULTIPLE REGRESSION (multi-variable)

Basic:

```
multiple regression of {numeric_y} on {numeric_x1}, {numeric_x2}
```

Where-filters:

```
multiple regression of {numeric_y} on {x1}, {x2} where {category} is Electronics  
multiple regression of {numeric_y} on {x1}, {x2} where {numeric_column} > 2020  
multiple regression of {numeric_y} on {x1}, {x2} where {region} is West
```

Notes: Returns coefficients and R². Use only numeric predictors.

LOG-TRANSFORM (Log regression)

Basic:

```
log regression of {numeric_x} and {numeric_y}
```

Where-filters: same pattern.

Notes: requires positive values; returns transformed-fit metrics.

EXAMPLES: combined complex commands

```
bar chart of {order_date} and {total_profit} where {region} is East  
scatter plot of {unit_cost} and {unit_price} where {item_type} is "Clothing"  
regression of {ad_spend} and {sales} where {market} is "Online"
```