MCP ANALYTICS

Correlation Analysis

Statistical analysis report with 8 sections — interactive charts, validated methodology, and AI-generated insights.

Generated **July 18, 2026**

Sections **8**

Platform **mcpanalytics.ai**

Contents

EXECUTIVE SUMMARY

Executive Summary

Key Metrics

Observations	**120**
Columns Analysed	**5**
Pairs Tested	**10**
Significant Pairs	**6**
Strongest |r|	**0.959**
Strongest Pair	**Ad Spend ~ Site Visits**

Interpretation

The metric landscape is densely interconnected: 6 of 10 pairs show statistically significant movement together ($p < 0.05$). The strongest relationship is Ad Spend ~ Site Visits ($r = 0.959$, $p = 2.2e-66$), a near-perfect positive co-movement. This tight web of associations indicates that most metrics in the dataset are not independent; they rise and fall as part of a few underlying drivers rather than in isolation.

OVERVIEW

Analysis Overview

Pairwise Pearson correlations across 5 columns and 120 observations.

Configuration

N Observations	120
N Columns	5
N Pairs	10
N Significant	6

Interpretation

This analysis examines which metrics move together by computing Pearson correlation coefficients—a measure of linear association ranging from -1 (perfect opposition) to +1 (perfect co-movement)—across 5 columns and 120 observations. The 10 possible pairs were tested for statistical significance; 6 of 10 pairs are significant at $p < 0.05$. Correlation identifies association only; it does not establish causation. The metric r quantifies the strength and direction of paired movements, enabling identification of metrics that rise and fall in tandem or in opposition.

DATA PREPARATION

Data Quality

Column typing, imputation, and exclusions.

Data Quality

Initial Rows	120
Final Rows	120
Rows Removed	0

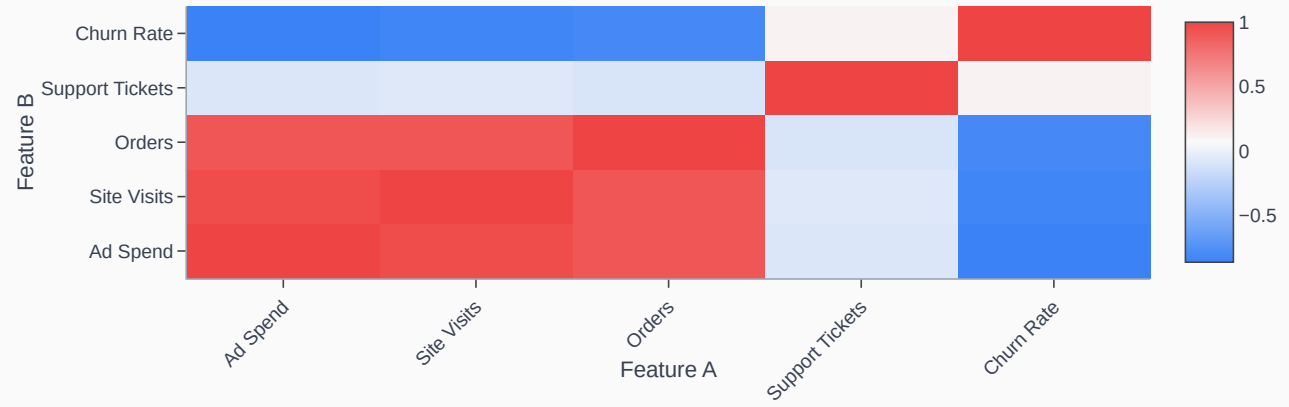
Interpretation

All 120 rows were retained; no rows were removed during preprocessing. Missing values were imputed using each column's median, ensuring no data loss. All 5 mapped columns (Ad Spend, Site Visits, Orders, Support Tickets, Churn Rate) were usable and included in the correlation analysis. Correlations were computed using pairwise complete observations, preserving the full dataset for relationship detection.

FIGURE 4

Correlation Matrix

Every pairwise correlation across the mapped columns.



Interpretation

Three pairs exhibit strong positive correlation (r ≥ 0.7): Ad Spend ~ Site Visits (0.959), Ad Spend ~ Orders (0.91), and Site Visits ~ Orders (0.91). These form a cohesive block, suggesting they measure related phenomena—traffic and transaction volume scale together with advertising investment. Three pairs show strong negative correlation (r ≤ -0.7): Ad Spend ~ Churn Rate (-0.862), Site Visits ~ Churn Rate (-0.833), and Orders ~ Churn Rate (-0.812). Support Tickets stands apart, showing weak or negligible correlations with all other metrics (range -0.095 to 0.112), indicating it operates independently of the main traffic-and-sales cluster.

TABLE 5

Strongest Relationships

Top column pairs ranked by absolute correlation, with significance.

Pair	Correlation	P Value	N	Strength	Significance
Ad Spend ~ Site Visits	0.959	2.20e-66	120	strong	***
Ad Spend ~ Orders	0.91	4.77e-47	120	strong	***
Site Visits ~ Orders	0.91	4.99e-47	120	strong	***
Ad Spend ~ Churn Rate	-0.862	1.24e-36	120	strong	***
Site Visits ~ Churn Rate	-0.833	4.05e-32	120	strong	***
Orders ~ Churn Rate	-0.812	2.21e-29	120	strong	***
Support Tickets ~ Churn Rate	0.112	0.222	120	weak	
Orders ~ Support Tickets	-0.095	0.304	120	weak	
Ad Spend ~ Support Tickets	-0.082	0.374	120	weak	
Site Visits ~ Support Tickets	-0.069	0.457	120	weak	

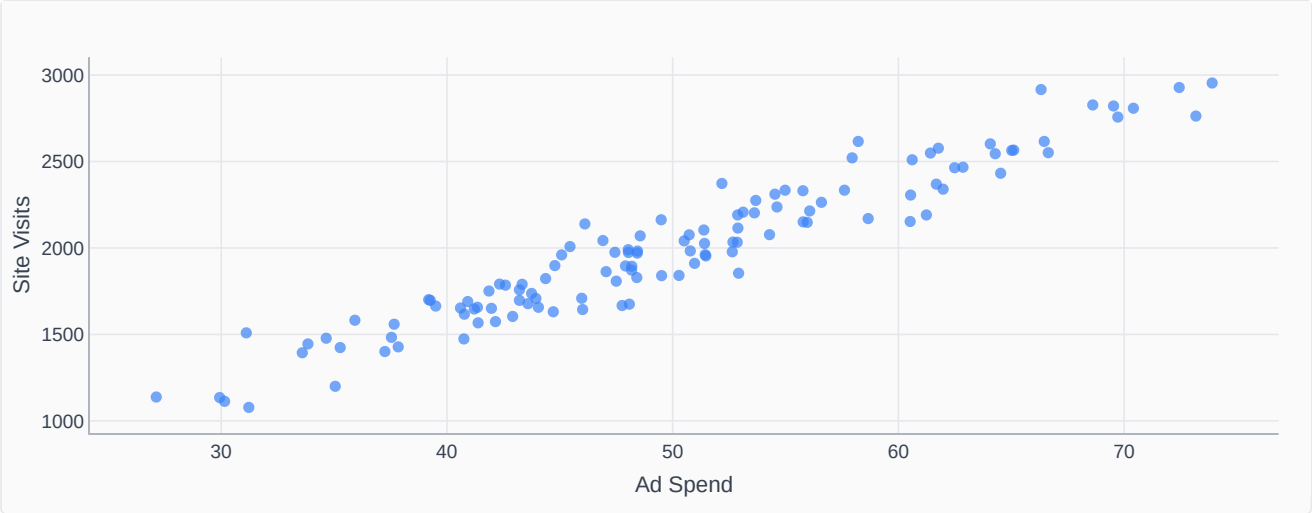
Interpretation

The top six relationships all achieve statistical significance ($p < 0.001$) and strong magnitudes ($|r| \geq 0.81$). Ad Spend ~ Site Visits dominates at $r = 0.959$. Ad Spend ~ Orders and Site Visits ~ Orders tie at $r = 0.91$. The three Churn Rate pairs—with Ad Spend (-0.862), Site Visits (-0.833), and Orders (-0.812)—reveal a consistent inverse pattern: higher engagement and spending associate with lower churn. The remaining four pairs (Support Tickets with others, $p > 0.22$) are weak and not statistically reliable, confirming Support Tickets' isolation from the main network.

FIGURE 6

Strongest Pair

The strongest relationship plotted point by point.



Interpretation

The Ad Spend vs Site Visits scatter (r = 0.959) shows a tight linear band across the 120 observations, with Ad Spend ranging from 27.12 to approximately 50+ and Site Visits from 1078 to 2139+. The points cluster closely around an upward trend with minimal scatter, confirming that the high correlation reflects genuine linear co-movement rather than a few extreme outliers. The consistent, tight relationship validates the strength of the r = 0.959 coefficient as a reliable summary of their joint behavior.

METHODOLOGY

Methodology

Statistical method, assumptions, and diagnostics

Correlation Analysis

Standard-library analysis: pairwise correlations across the numeric columns you choose. A full correlation heatmap, the strongest relationships ranked with significance tests, and a scatter of the single strongest pair — so you can see which metrics move together, how tightly, and whether it's statistically real. Works on any dataset: map 2 or more numeric columns.

DATA

N = 120 observations

ASSUMPTIONS

- Relationships are approximately linear (Pearson r)
- Columns are numeric or cleanly convertible

LIMITATIONS

- Correlation is not causation
- Non-linear or clustered relationships can hide from Pearson r — check the scatter
- Outliers can inflate or mask correlations

SOFTWARE & CITATION

MCP Analytics · mcpanalytics.ai

APPENDIX

Analysis Code

Complete R source code for this analysis

Correlation Analysis — What Moves Together

Computes pairwise Pearson correlations across the numeric columns the user selects: full matrix heatmap, strongest relationships ranked with significance tests, and a scatter of the dominant pair.

A.1 Why This Method?

Correlation is the fastest map of a dataset's relationships: one number per pair, directly comparable, with a significance test separating real co-movement from noise. It is the standard first step before regression, forecasting, or KPI pruning.

A.2 What This Analysis Covers

- Full correlation matrix (heatmap)
- Strongest relationships ranked with p-values
- The dominant pair plotted point by point

A.3 Standard Library

Platform standard-library module (LAT-1441): runs on ANY dataset via the semantic mapping {feature_1..feature_N}. All narrative is derived from the user's own column names and computed values.

```
suppressPackageStartupMessages(library(DT))
suppressPackageStartupMessages(library(htmlwidgets))
suppressPackageStartupMessages(library(arrow))
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(rmarkdown))
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(tidyr))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(stringr))
suppressPackageStartupMessages(library(lubridate))
suppressPackageStartupMessages(library(broom))
suppressPackageStartupMessages(library(Matrix))
suppressPackageStartupMessages(library(cluster))
suppressPackageStartupMessages(library(data.table))
```

A.4 Core Analysis Pipeline

```
compute_shared <- function(df, params, col_map = list()) {
  # === SHARED EXPORTS ===
  #   initial_rows/final_rows/rows_removed  $ row accounting
  #   feature_names     $ named character — semantic -> humanized names
  #   used_features     $ character — semantic names used
  #   dropped_features  $ character — excluded columns
  #   cor_long_df       $ data.frame(feature_a, feature_b, correlation) — full matrix
  #   pairs_df          $ data.frame(pair, correlation, p_value, n, strength, significance)
  #   top_pair_df       $ data.frame(value_a, value_b) — <=1000 sample of strongest pair
  #   top_pair_names    $ character(2) — humanized names of the strongest pair
  #   top_r / top_p     $ numeric — strongest pair stats
  #   n_sig             $ integer — pairs significant at p<0.05
  #   metrics / json_output
  # === /SHARED EXPORTS ===
```

Step 1: Discover mapped features

```
initial_rows <- nrow(df)
  feat_cols <- grep("^feature_[0-9]+$", names(df), value = TRUE)
  feat_cols <- feat_cols[order(as.integer(sub("^feature_", "", feat_cols)))]
  if (length(feat_cols) < 2) {
    stop("column_mapping must map at least two feature columns(feature_1, feature_2)")
  }
  feature_names <- setNames(humanize_semantic(feat_cols, col_map), feat_cols)
```

Step 2: Coerce numeric (95% rule); impute median; drop unusable

```
dropped_features <- character(0)
  for (fc in feat_cols) {
    v <- df[[fc]]
    if (!is.numeric(v)) {
      conv <- suppressWarnings(as.numeric(as.character(v)))
      n_orig <- sum(!is.na(v) & as.character(v) != "")
      if (n_orig > 0 && sum(!is.na(conv)) >= 0.95 * n_orig) {
        df[[fc]] <- conv
      } else {
        dropped_features <- c(dropped_features, fc); next
      }
    }
    v <- df[[fc]]
    med <- median(v, na.rm = TRUE)
    if (is.na(med)) { dropped_features <- c(dropped_features, fc); next }
    v[is.na(v)] <- med
    df[[fc]] <- v
    if (isTRUE(var(v) == 0) || is.na(var(v))) {
      dropped_features <- c(dropped_features, fc)
    }
  }
  used_features <- setdiff(feat_cols, dropped_features)
  if (length(used_features) < 2) {
    stop(paste0("Correlation needs at least two usable numeric columns; only ",
                length(used_features), " remained after cleaning."))
  }
  X <- df[, used_features, drop = FALSE]
  final_rows <- nrow(X)
  rows_removed <- initial_rows - final_rows
  if (final_rows < 10) stop(sprintf("Only %d usable rows — need at least 10.", final_rows))
```

Step 3: Correlation matrix + per-pair tests

```
C <- suppressWarnings(cor(X, use = "pairwise.complete.obs", method = "pearson"))
  hn <- unname(feature_names[used_features])
```

Long format for the heatmap (diagonal included — reads as the standard matrix)

```
k <- length(used_features)
  cor_long_df <- do.call(rbind, lapply(seq_len(k), function(i) {
    data.frame(feature_a = hn[i], feature_b = hn,
               correlation = round(as.numeric(C[i, ]), 3),
               stringsAsFactors = FALSE)
  }))
  rownames(cor_long_df) <- NULL
```

Pairwise tests (upper triangle only)

```
pair_rows <- list()
  for (i in seq_len(k - 1)) {
    for (j in (i + 1):k) {
      r_ij <- C[i, j]
      if (is.na(r_ij)) next
      ct <- tryCatch(cor.test(X[[i]], X[[j]]), error = function(e) NULL)
      p_ij <- if (!is.null(ct)) ct$p.value else NA_real_
      pair_rows[[length(pair_rows) + 1]] <- data.frame(
        pair = paste0(hn[i], " ~ ", hn[j]),
        a = used_features[i], b = used_features[j],
        correlation = round(r_ij, 3),
        p_value = signif(p_ij, 3),
        n = sum(complete.cases(X[[i]], X[[j]])),
        stringsAsFactors = FALSE
      )
    }
  }
  pairs_all <- do.call(rbind, pair_rows)
  if (is.null(pairs_all) || nrow(pairs_all) == 0) {
    stop("No valid correlation pairs could be computed from the mapped columns.")
  }
  pairs_all <- pairs_all[order(-abs(pairs_all$correlation)), , drop = FALSE]
  rownames(pairs_all) <- NULL
  pairs_all$strength <- ifelse(abs(pairs_all$correlation) >= 0.7, "strong",
                        ifelse(abs(pairs_all$correlation) >= 0.4, "moderate", "weak"))
  pairs_all$significance <- ifelse(is.na(pairs_all$p_value), "",
                            ifelse(pairs_all$p_value < 0.001, "***",
                            ifelse(pairs_all$p_value < 0.01, "**",
                            ifelse(pairs_all$p_value < 0.05, "*", ""))))
  n_sig <- sum(pairs_all$p_value < 0.05, na.rm = TRUE)
  pairs_df <- head(pairs_all[, c("pair", "correlation", "p_value", "n",
                                 "strength", "significance")], 15)
```

Step 4: Strongest pair scatter — <=1000 sample

```r
top <- pairs_all[1, ]
  top_pair_names <- c(feature_names[[top$a]], feature_names[[top$b]])
  set.seed(42)
  sidx <- if (final_rows > 1000) sample(final_rows, 1000) else seq_len(final_rows)
  top_pair_df <- data.frame(
    value_a = X[[top$a]][sidx],
    value_b = X[[top$b]][sidx],
    stringsAsFactors = FALSE
  )
  top_pair_df <- top_pair_df[order(top_pair_df$value_a), ]
  rownames(top_pair_df) <- NULL
  top_r <- top$correlation
  top_p <- top$p_value

  metrics <- list(
    `Observations`        = final_rows,
    `Columns Analysed`    = length(used_features),
    `Pairs Tested`        = nrow(pairs_all),
    `Significant Pairs`   = as.integer(n_sig),
    `Strongest |r|`       = round(abs(top_r), 3),
    `Strongest Pair`      = top$pair
  )

  json_output <- list(
    answer = paste0(
      "Pairwise Pearson correlations on ", length(used_features),
      " columns across ", format(final_rows, big.mark = ","), " rows: ",
      "strongest pair is ", top$pair, " (r = ", top_r,
      if (!is.na(top_p)) paste0(", p = ", top_p) else "", "); ",
      n_sig, " of ", nrow(pairs_all), " pairs significant at p<0.05."
    ),
    cards = lapply(
      c("tldr", "overview", "preprocessing", "correlation_heatmap",
        "strongest_pairs", "top_pair_scatter"),
      function(cid) list(id = cid, metrics = metrics)
    )
  )

  list(
    initial_rows = initial_rows, final_rows = final_rows,
    rows_removed = rows_removed,
    feature_names = feature_names, used_features = used_features,
    dropped_features = dropped_features,
    cor_long_df = cor_long_df, pairs_df = pairs_df, pairs_all = pairs_all,
    top_pair_df = top_pair_df, top_pair_names = top_pair_names,
    top_r = top_r, top_p = top_p, n_sig = n_sig,
    metrics = metrics, json_output = json_output
  )
}
```

This is the exact R code executed to produce the analysis above. Run it with the same dataset to reproduce these results.