

MCP ANALYTICS

Time Series Forecast

Statistical analysis report with 9 sections — interactive charts, validated methodology, and AI-generated insights.

Generated **July 18, 2026**

Sections **9**

Platform **mcpanalytics.ai**

Contents

Executive Summary	Executive Summary
Overview	Analysis Overview
Data Preparation	Data Preparation
Figure 4	Forecast
Table 5	Trend & Seasonality
Figure 6	Seasonal Pattern
Table 7	Forecast Accuracy
Methodology	Methodology
Appendix	Analysis Code

EXECUTIVE SUMMARY

Executive Summary

Key Metrics

Observations	104
Frequency	weekly
Forecast Horizon	26
Projected Change (%)	7.65
Holdout MAPE (%)	4.52
Method	Holt-Winters exponential smoothing (level + trend + additive seasonality)

Suggested Interpretation

Total Sales is projected to rise 7.65% over the next 26 weeks, from 697.95 to 751.34. The underlying trend is upward at +2.02 per week, compounding to approximately +52.5 over the forecast horizon. A repeating seasonal pattern swings Total Sales by 52.1 peak-to-trough within each year. Backtested on a 34-week holdout, the model achieved 4.52% MAPE—well below the 5% threshold for excellent business forecasting—confirming high reliability for planning and target-setting.

OVERVIEW

Analysis Overview

Weekly forecast of Total Sales over 26 weeks.

Configuration

N Observations	104
Horizon	26
Frequency	weekly
Seasonal Model	True

Suggested Interpretation

Holt-Winters exponential smoothing was selected to forecast 26 weeks of Total Sales from a foundation of 104 weekly observations. The model choice reflects sufficient historical depth—two full years—to reliably estimate a repeating seasonal pattern alongside trend. The series required no cleaning: all 104 rows were already regularly spaced, with no duplicate dates or gaps. The forecast assumes current patterns persist and cannot anticipate one-off events; 80% and 95% prediction intervals quantify uncertainty around the central projection.

DATA PREPARATION

Data Preparation

How the raw rows became a regular time series.

Data Quality

Initial Rows	104
Final Rows	104
N Duplicate Rows	0
N Interpolated	0
N Dropped	0
Frequency	weekly

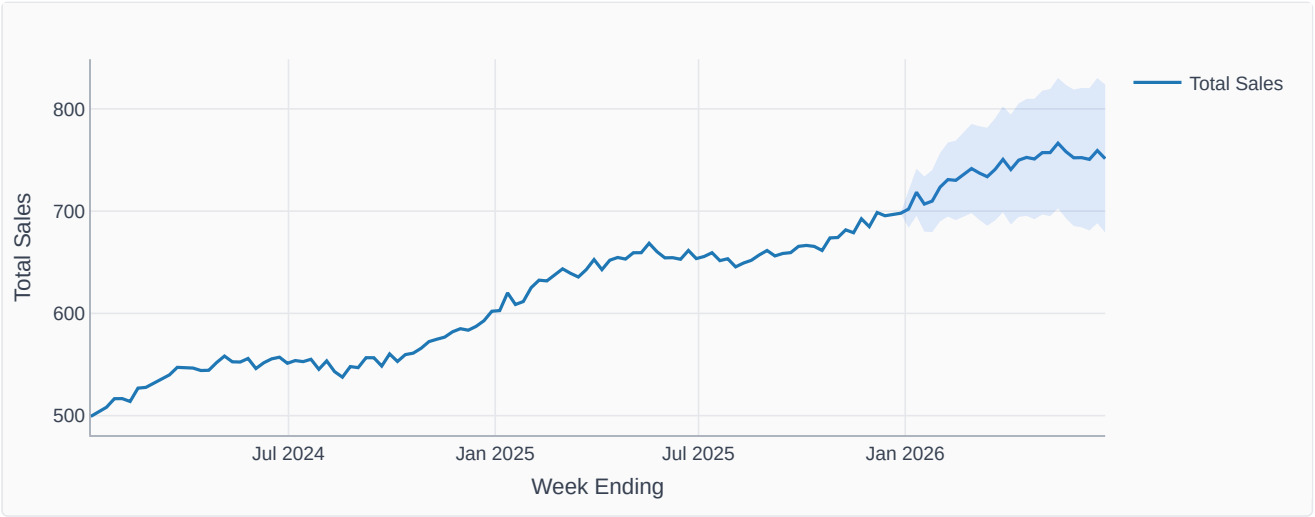
Suggested Interpretation

The raw 104 rows required minimal preparation. Each date appeared exactly once, so no aggregation by sum was necessary. The weekly cadence was inferred from the median gap between consecutive dates, and no gaps existed in the series—it was already regular. All 104 observations were retained as the final time series. This clean structure allowed direct fitting without interpolation, reducing the risk of artificially imposed patterns.

FIGURE 4

Forecast

Total Sales history and 26-week projection with 95% prediction band.



Suggested Interpretation

The forecast projects Total Sales to reach 751.34 by week 26, with a 95% prediction interval spanning 679 to 823.69 (band width 144.69). Over the observed history, the prediction band hugs actuals; over the forecast, it widens as uncertainty compounds. The widening band reflects genuine future uncertainty and should anchor planning ranges rather than point estimates alone. The central line shows steady upward drift consistent with the +2.02 per-week trend.

TABLE 5

Trend & Seasonality

The series decomposed into level, trend, and seasonal swing.

Component	Description	Magnitude
Level	Estimated current level of Total Sales	700.1
Trend	Average change in Total Sales per week (upward)	2.018
Seasonality	Peak-to-trough swing of the repeating weekly pattern within a year	52.09

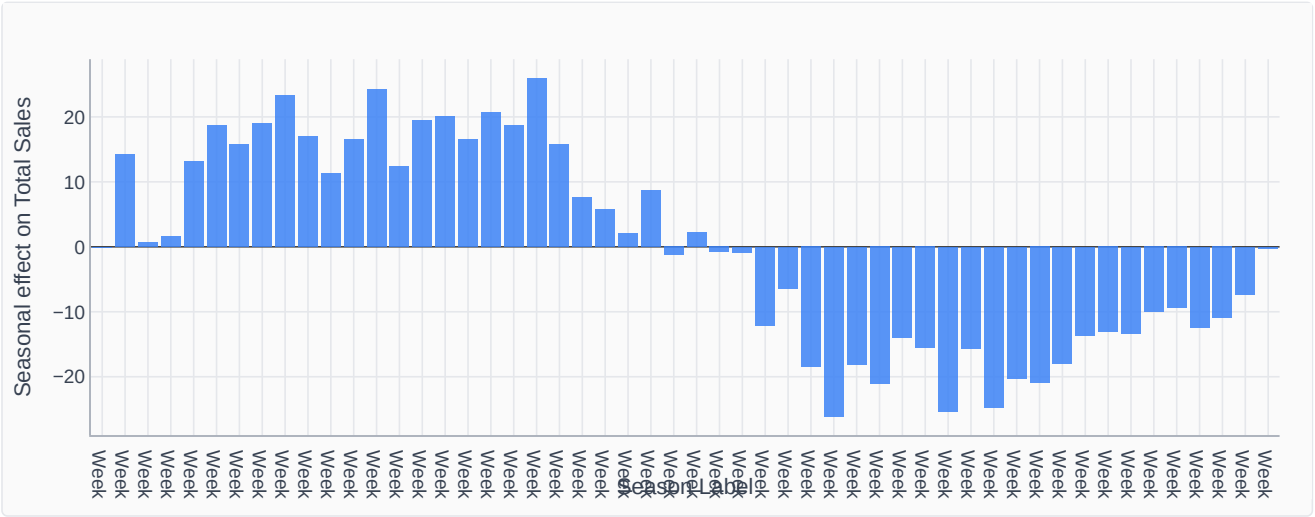
Suggested Interpretation

The decomposition reveals three drivers of Total Sales movement: a current level of 700.119, a per-week upward trend of +2.0183, and a seasonal swing of 52.092 peak-to-trough. The trend dominates directional movement, adding roughly +52.5 over the 26-week horizon. Seasonality is substantial but secondary—it modulates the trend rather than reversing it. Week-to-week changes reflect both the steady upward drift and the repeating seasonal pattern layered atop it.

FIGURE 6

Seasonal Pattern

Within-year seasonal effects on Total Sales.



Suggested Interpretation

Within each year, Total Sales exhibits a strong seasonal cycle. Week 20 is the seasonal peak, running +25.987 above trend, while Week 33 is the trough at -26.105 below trend—a full swing of 52.092. The first half of the year (Weeks 1–20) tends above trend, peaking in spring; the second half (Weeks 21–52) trends below. The forecast already incorporates this pattern, so projected values account for which weeks in the cycle fall within the 26-week horizon.

TABLE 7

Forecast Accuracy

Backtested holdout error — how far off this model tends to be.

Metric	Value	Interpretation
Model	—	Holt-Winters exponential smoothing (level + trend + additive seasonality)
History points	104	Regularized weekly observations of Total Sales used to fit the model
Forecast horizon	26	Weeks projected beyond the last observed Week Ending
Holdout points	34	Most recent points withheld, then predicted by a model that never saw them
Holdout MAPE (%)	4.52	Average prediction error of 4.5% on unseen data — highly reliable
Holdout MAE	30.12	Average absolute error in units of Total Sales

Suggested Interpretation

The model was backtested by withholding the final 34 weeks, refitting on earlier data, and scoring predictions against the held-out actuals. Results: 4.52% MAPE and 30.122 MAE. At 4.52%, the error lies in the "excellent" range (under 5%) for business forecasting, indicating the model can reliably inform both directional strategy and sales targets. This high accuracy reflects the regular, patterned nature of the weekly sales series and justifies confidence in the forward projection through week 26.

METHODOLOGY

Methodology

Statistical method, assumptions, and diagnostics

Time Series Forecast

Standard-library analysis: forecast any metric from its own history. Maps a date column and a numeric column, infers the data's frequency (daily, weekly, monthly), fits a Holt-Winters exponential smoothing model with automatic seasonal and non-seasonal fallbacks, and projects forward with 95% prediction intervals. Includes a holdout backtest (MAPE/MAE) so you know how much to trust the forecast, plus a breakdown of trend and seasonal pattern.

DATA

N = 104 observations

ASSUMPTIONS

- The future resembles the past: trend and seasonal patterns learned from history persist over the forecast horizon
- Observations are (or can be aggregated to) a regular daily, weekly, or monthly cadence
- One metric, one series — no external drivers or regressors

LIMITATIONS

- Cannot anticipate structural breaks, one-off events, or policy changes not present in the history
- Short histories force a non-seasonal model even when true seasonality exists
- Prediction intervals assume the model's errors are well-behaved; real uncertainty can be larger

SOFTWARE & CITATION

MCP Analytics · mcpanalytics.ai

APPENDIX

Analysis Code

Complete R source code for this analysis

Time Series Forecast — See Where Your Numbers Are Heading

Forecasts a single metric from its own history: parses and regularizes the dates, infers the frequency (daily/weekly/monthly), fits a Holt-Winters exponential smoothing model (additive seasonal when the history supports it, non-seasonal or ARIMA(1,1,1) otherwise), projects forward with 80%/95% prediction intervals, and backtests itself on a holdout window (MAPE/MAE).

A.1 Why This Method?

Holt-Winters is the workhorse of business forecasting: it learns level, trend, and a repeating seasonal pattern directly from the series, needs no external drivers, and degrades gracefully — when the history is too short for seasonality the model simply drops that component. Base R only (stats::HoltWinters + stats::arima); no forecast/prophet packages.

A.2 What This Analysis Covers

- Forecast with 95% prediction band (chart)
- Trend + seasonality decomposition
- Within-cycle seasonal pattern
- Backtested holdout accuracy (MAPE / MAE)

A.3 Standard Library

Platform standard-library module (LAT-1441): runs on ANY dataset via the semantic mapping {date, value}. 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

Step 1: Parse dates robustly (init() may already have Date-coerced)

```
raw_date <- df$date
non_blank <- !(is.na(raw_date) | !nzchar(trimws(as.character(raw_date))))
d <- parse_dates_robust(raw_date)
n_nonblank <- sum(non_blank)
if (n_nonblank == 0) {
  stop(sprintf("The &#x27;%s' column is empty – no dates to forecast from.", date_name))
}
n_unparsed <- sum(non_blank & is.na(d))
if (n_unparsed / n_nonblank > 0.05) {
  stop(sprintf(
    "%d of %d values in &#x27;%s' (%.0f%%) could not be read as dates. Please use a recognizable date format (e.g. 2024-01-31\n",
    n_unparsed, n_nonblank, date_name, 100 * n_unparsed / n_nonblank))
  )
}
```

Step 2: Coerce the metric (95% rule)

```

v <- df$value
if (!is.numeric(v)) {
  conv <- suppressWarnings(as.numeric(as.character(v)))
  n_orig <- sum(!is.na(v) & nzchar(as.character(v)))
  if (n_orig > 0 && sum(!is.na(conv)) >= 0.95 * n_orig) {
    v <- conv
  } else {
    stop(sprintf("The &#x27;%s' column is not numeric – a forecast needs a numeric metric.",
                 value_name))
  }
}
v <- as.numeric(v)

keep <- !is.na(d) & !is.na(v)
n_dropped_rows <- initial_rows - sum(keep)
work <- data.frame(date = d[keep], value = v[keep])

```

Step 4: Infer frequency from the median date gap

```

gaps <- as.numeric(diff(agg$date))
med_gap <- median(gaps)
if (med_gap <= 1.5) {
  freq_label <- "daily"; step_days <- 1; freq <- 7; period_word <- "day"; cycle_word <- "week"
} else if (med_gap >= 5.5 && med_gap <= 8.5) {
  freq_label <- "weekly"; step_days <- 7; freq <- 52; period_word <- "week"; cycle_word <- "year"
} else if (med_gap >= 26 && med_gap <= 35) {
  freq_label <- "monthly"; step_days <- NA; freq <- 12; period_word <- "month"; cycle_word <- "year"
} else {
  freq_label <- "irregular"; step_days <- max(1, round(med_gap)); freq <- 1
  period_word <- sprintf("%d-day period", step_days); cycle_word <- "cycle"
}

```

Step 5: Regular grid + linear interpolation of small gaps

```

if (freq_label == "monthly") {
  snapped <- as.Date(format(agg$date, "%Y-%m-01"))
  agg <- aggregate(value ~ date, data = data.frame(date = snapped, value = agg$value),
    FUN = sum)
  agg <- agg[order(agg$date), , drop = FALSE]
  grid <- seq(min(agg$date), max(agg$date), by = "month")
} else {
  grid <- seq(min(agg$date), max(agg$date), by = step_days)
}
interp <- approx(x = as.numeric(agg$date), y = agg$value,
  xout = as.numeric(grid), rule = 2)
values <- interp$y
n_interp <- sum(!(as.numeric(grid) %in% as.numeric(agg$date)))
if (freq_label != "irregular" && n_interp / length(grid) > 0.5) {
  stop(sprintf(
    "More than half of the %s periods between the first and last %s' are missing – too sparse to forecast reliably.",
    freq_label, date_name))
}

n_points <- length(values)
final_rows <- n_points
rows_removed <- max(0, initial_rows - sum(keep))

```

Step 6: Fit (seasonal HW -> HW trend -> ARIMA) + forecast horizon

```

horizon <- max(2, min(floor(0.25 * n_points), 2 * max(freq, 3), 30))
fitres <- fit_chain(values, freq, horizon)
method <- fitres$method
method_label <- fitres$method_label
seasonal_fit <- isTRUE(fitres$seasonal)

if (freq_label == "monthly") {
  future_dates <- seq(grid[length(grid)], by = "month", length.out = horizon + 1)[-1]
} else {
  future_dates <- grid[length(grid)] + step_days * seq_len(horizon)
}

```

Step 7: Backtest on a holdout window (skipped gracefully if too short)

```

mape <- NA_real_; mae <- NA_real_; holdout_n <- 0L
backtest_note <- NULL
holdout_target <- min(max(freq, 6), floor(n_points / 3))
if (holdout_target >= 4 && (n_points - holdout_target) >= 10) {
  bt <- tryCatch({
    tr <- values[seq_len(n_points - holdout_target)]
    fit_chain(tr, freq, holdout_target)
  }, error = function(e) NULL)
  if (!is.null(bt)) {
    act <- values[(n_points - holdout_target + 1):n_points]
    pred <- bt$mean
    mae <- mean(abs(act - pred))
    nz <- abs(act) > 1e-9
    if (any(nz)) mape <- mean(abs((act[nz] - pred[nz]) / act[nz])) * 100
    holdout_n <- holdout_target
  } else {
    backtest_note <- "Backtest skipped – the model could not be refitted on the shortened training window."
  }
} else {
  backtest_note <- "Backtest skipped – the series is too short to hold out a meaningful test window."
}

```

Step 8: Headline numbers

```

last_actual <- values[n_points]
fc_last <- fitres$mean[horizon]
pct_change <- if (abs(last_actual) > 1e-9) {
  (fc_last - last_actual) / abs(last_actual) * 100
} else NA_real_
direction_word <- if (is.na(pct_change)) "move" else if (pct_change > 2) "rise" else if (pct_change < -2) "fall" else "hold rc

```

Step 9: Components (level / trend / seasonality)

```
idx <- seq_len(n_points)
trend_slope <- as.numeric(coef(lm(values ~ idx))[2])
level_now <- last_actual
seasonal_amplitude <- 0
seasonal_coefs <- numeric(0)
if (seasonal_fit && !is.null(fitres$fit)) {
  co <- coef(fitres$fit)
  if ("a" %in% names(co)) level_now <- as.numeric(co["a"])
  if ("b" %in% names(co)) trend_slope <- as.numeric(co["b"])
  seasonal_coefs <- as.numeric(co[grepl("^s[0-9]+$", names(co))])
  if (length(seasonal_coefs) > 0) {
    seasonal_amplitude <- max(seasonal_coefs) - min(seasonal_coefs)
  }
} else if (method == "hw_trend" && !is.null(fitres$fit)) {
  co <- coef(fitres$fit)
  if ("a" %in% names(co)) level_now <- as.numeric(co["a"])
  if ("b" %in% names(co)) trend_slope <- as.numeric(co["b"])
}
```

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