ANALYTICAL REPORT

Resort Operations Dashboard & Data Pipeline

Self-directed portfolio piece: Decision Infrastructure, on a fully synthetic dataset

Data Pipeline | ETL | Dashboard Design | Reconciliation

Prepared for	Arbutus Visual Analytics
Prepared by	Marc Charbonneau, Principal Analyst, Arbutus Visual Analytics
Date	July 18, 2026
Confidentiality	Self-directed portfolio piece: fully synthetic dataset, no real resort, advisor, or firm

EXECUTIVE SUMMARY

Written for a non-technical audience.

Most independent ski resorts run visitation, weather/snow, lift status, and staffing as four disconnected systems, checked separately every morning rather than viewed together. This piece builds a Python ETL pipeline that cleans and joins those four sources into one daily-operations fact table, and a live, interactive dashboard the operations team can check every morning instead of four separate systems. All data in this report and its accompanying dashboard is fully synthetic. See Conclusions, right after the infographic overview, for the full summary.

OPERATING DAYS MODELLED
152
One fabricated ski season, mid-November through mid-April
↓ jump to finding

RAW SOURCE SYSTEMS UNIFIED
4
Lift gate scans, unique daily visitation, weather/snow conditions, lift-status/downtime log, and staffing shifts
↓ jump to finding

DOWNTIME LOG ROWS CLEANED AWAY
75
45 exact duplicate log entries removed, 30 adjacent events merged into a single outage: exactly what a naive daily total would double-count
↓ jump to finding

AVERAGE LIFT UPTIME
98.3%
Across all 6 lifts and the full season, after cleaning the downtime log
↓ jump to finding

SCANS-PER-VISITOR RATIO
3.39×
Confirms summed lift-gate scans cannot be used as a total-visitation figure: one visitor rides several lifts several times a day
↓ jump to finding

ENGAGEMENT OVERVIEW

Resort Operations Dashboard & Data Pipeline

A self-directed Decision Infrastructure portfolio piece for Arbutus Visual Analytics

1 PROBLEM

The Challenge

- An operations team checks 3-4 disconnected systems every morning: ticket/gate scans, a weather station feed, a lift-status log, and a staffing spreadsheet
- None of those systems agree on basic questions: total visitation, whether staffing matches expected demand, or which lift is actually driving downtime
- A one-time report cannot solve this: the answer needs to keep working every operating day of the season

2 WORKFLOW

How We Built It

- Generated four fabricated raw source files reproducing the real reconciliation problems a unified pipeline has to solve
- Built a Python ETL pipeline: de-duplicate and merge the downtime log, reconcile lift scans against true unique visitation, aggregate shift-level staffing to daily headcount
- Joined all four sources into one daily-operations fact table plus dashboard-ready aggregates
- Built the dashboard from reusable chart and KPI components, so a new metric can be added without rebuilding the interface from scratch

3 SOLUTION

What We Delivered

- A live, interactive dashboard (not static images), reading the pipeline's output
- A documented, re-runnable pipeline with logged data-quality checks at every cleaning step, not a one-off spreadsheet fix
- A single daily view that replaces four separate morning checks with one

Fully synthetic dataset, generated to match the structure of a real independent ski resort's operations data. No real resort, advisor, or firm data appears anywhere in this piece.

Conclusions

Written for a non-technical audience.

WHAT WE SET OUT TO ANSWER

Can a small, well-built ETL pipeline and dashboard turn four disconnected resort operations systems into one recurring, trustworthy daily view for the operations team?

WHAT WE FOUND

- Three concrete data-quality problems (duplicate/adjacent downtime log entries, lift-scan-vs-visitor reconciliation, shift-level staffing aggregation) had to be solved before the four sources could be safely joined: exactly the kind of hidden discrepancy a manual daily check across four separate systems would miss.
- Average lift uptime across the season (98.3%) and the season-total downtime Pareto by lift both required the cleaned, de-duplicated downtime log; the raw log alone would have overstated total downtime.
- Weather has a real but modest relationship with daily visitation (r = 0.108) once calendar effects are in play: an honestly-reported, operationally useful finding rather than an inflated headline correlation.
- Normalizing gate scans by each lift's rated capacity flips the naive read: the beginner carpet has the lowest raw scan count of all six lifts but the highest capacity utilization, while the six-pack chair has the second-highest raw scan count but the lowest utilization. Raw volume alone would have pointed at the wrong lift for capacity planning.
- The dashboard was built from reusable chart and KPI components rather than one-off page code, so new lifts, departments, or metrics can be added to the daily view without rebuilding it from scratch.

WHAT WE RECOMMEND

1. Adopt the unified daily-operations view as the morning routine in place of checking ticketing, weather, lift-status, and staffing systems separately.
2. Extend the same fact table with a next-day staffing forecast once a full season of real data is available, using the weather feed already flowing into the pipeline.
3. Feed the cleaned downtime log back into lift maintenance scheduling, since the Pareto view makes it straightforward to identify which lift is disproportionately driving outages and prioritize it for service.

Abbreviations

ETL	Extract, Transform, Load: the standard pattern for a data pipeline that moves raw data into a clean, analysis-ready form
KPI	Key Performance Indicator
RFID	Radio-Frequency Identification, the gate-scan technology used for lift-ticket/pass validation

1. The Operations Problem and the Synthetic Dataset

> This section covers the real operations problem this piece addresses, why a synthetic dataset was used, and the structure of the four fabricated source systems.

1.1. The Operations Problem

Independent ski resorts typically run several disconnected systems: a point-of-sale/ticketing system counting unique daily visitors, an RFID gate-scan system logging lift usage, a weather station feed, a lift-status log maintained by lift operations, and a staffing schedule maintained separately by each department. A survey of ski-industry professionals conducted by SAM (Ski Area Management) magazine and Eternity found that 63% of resorts report either no unified analytics (35%) or only a partial analytics capability (28%). The majority of the industry is working from exactly this kind of siloed picture. The operations team's morning routine, checking each system separately, cannot easily answer questions like "are we staffed correctly for today's expected visitation" or "which lift is actually driving most of our downtime this season," because no single system has all the inputs.

1.2. A Fully Synthetic Season

This piece uses a fully fabricated one-season dataset (mid-November through mid-April, 152 operating days), not real data from any resort. No public dataset combines lift-level gate scans, a lift-status/downtime log, and department-level staffing at daily granularity for any real resort, so synthetic generation was the only realistic path regardless of preference. This has the added benefit that the data could be engineered specifically to reproduce the real reconciliation problems a unified pipeline has to solve (Section 2), rather than being constrained by whatever a found dataset happened to contain.

1.3. Four Source Systems

The synthetic dataset models six lifts (a gondola, a six-pack chair, a quad chair, a double chair, a T-bar, and a beginner carpet, each with a different rated capacity) and four departments (Lift Operations, Ski Patrol, Guest Services, Food & Beverage) across four fabricated raw files: daily unique visitation (from a ticketing/pass system), per-lift daily RFID gate-scan counts, daily weather and snow conditions, a lift-status/downtime event log, and AM/PM shift-level staffing counts by department. Demand was modelled with a realistic seasonal shape (building toward a December-holiday and February peak, tapering by April), weekend and holiday multipliers, and a same-day snowfall ("powder day") multiplier, so the resulting dataset has the same kind of calendar- and weather-driven structure a real season would.

2. The Pipeline: Cleaning, Reconciliation, and Aggregation

> The three concrete data-quality problems the pipeline has to solve before the four sources can be joined into one fact table.

2.1. De-duplicating and Merging the Downtime Log

The raw lift-status log, as generated, includes two deliberate real-world quirks: about 15% of downtime events are logged twice by different systems (an exact duplicate row), and about 10% of events are immediately followed by a short, separate log entry under a 10-minute gap that represents the same physical outage (e.g. a restart attempt) rather than a second independent failure. The pipeline removes exact duplicates first (45 of 324 raw rows in this dataset), then merges any remaining events on the same lift and day whose gap falls within a 10-minute threshold (a further 30 events merged), before summing downtime minutes per lift per day. Skipping either step would overstate total downtime and understate lift uptime.

2.2. Reconciling Lift Scans Against True Visitation

Summed RFID gate-scan counts across all six lifts run about 3.39× higher than the ticketing system's unique daily visitor count, since one visitor rides several different lifts several times a day. The pipeline treats unique_visitors (from the ticketing/pass system) as the only correct total-visitation figure, and uses per-lift scan counts exclusively for per-lift utilization, never summed as a resort-wide visitation number. This distinction is easy to get wrong if the two data sources are merged without checking what each one actually measures. Raw scan counts alone still favor larger lifts by construction, so the pipeline goes one step further and divides each lift's season-total scans by its own rated hourly capacity, producing a size-independent utilization rate: which lift runs closest to its own limit, not just which lift is busiest in absolute terms.

2.3. Aggregating Shift-Level Staffing to Daily Headcount

Staffing data arrives as AM/PM shift-level rows per department, and the correct daily figure is peak concurrent headcount (the maximum of the two shifts), not the sum of both shifts, which would double-count staff whose coverage spans the day. The pipeline aggregates via that maximum, department by department, before summing across departments to a daily total.

2.4. The Unified Fact Table

After cleaning, reconciling, and aggregating each source independently, the pipeline joins all four on date into one 152-row, 15-column daily-operations fact table (visitation, weather/snow, total staffing, staffing ratio, total downtime minutes, and lift uptime percentage), plus two dashboard-specific aggregate views (season-total downtime by lift, and a weather/visitation correlation dataset). This fact table, not any individual raw source, is what the dashboard and any future automated reporting should read from.

3. Dashboard Design and What It Shows

> The interactive dashboard built on this fact table, what it shows, and the design choices behind it.

3.1. Built to Extend, Not Just to View

The dashboard's KPI tiles, time-series chart, Pareto chart, ranked-bar chart, and correlation scatter were built from reusable chart and KPI components on top of Apache ECharts, rather than page-specific chart code. This means a new metric the operations team wants next season, a lift added to the fleet, or a fifth department added to the staffing view can be built by adding a component to the dashboard, not by rebuilding the interface. The dashboard is a custom-built page rather than an embedded BI-platform report, which avoids per-seat licensing and keeps it hosted alongside the resort's other public-facing pages instead of behind a separate login.

3.2. What the Dashboard Shows

The published dashboard presents five KPI tiles (total and average daily visitors, average lift uptime, average staffing ratio, and total downtime hours for the season), a dual-axis time series of daily visitation against total staffing across the full season, a Pareto chart of season-total downtime minutes by lift (identifying which lift disproportionately drives outages), a ranked-bar chart of each lift's gate-scan volume as a share of its own rated capacity (identifying which lift runs closest to its own limit, regardless of size), and a scatter plot of same-day snowfall against unique visitors with a fitted trend line.

3.3. A Modest, Honestly-Reported Weather Correlation

The correlation between same-day snowfall and unique visitors is positive but modest ($r = 0.108$) across the full season. This is reported as-is rather than adjusted to look more dramatic: it indicates that calendar effects (weekday versus weekend, and holidays) are the dominant driver of daily visitation in this dataset, with snowfall acting as a real but secondary factor. This is a plausible, useful operational finding (staffing should be planned primarily around the calendar, with snow as a secondary same-day adjustment) rather than a claim that snow alone predicts turnout.

3.4. A Static JSON Data Source, by Design

The published dashboard reads a static JSON file produced by the pipeline at build time, not a live database connection. This is a deliberate simplification appropriate for a demonstration built on fabricated data with no real backend to protect: a live database would add real hosting and security surface for no benefit here. A production deployment for a real resort would point the identical dashboard at a live database connection instead: the frontend does not change, only the data source.

Recommendations

1. Treat lift-gate scan counts and ticketing-system visitor counts as answering two different questions (per-lift utilization vs. total visitation) and never sum scans across lifts as a resort-wide visitation figure.
2. Build any recurring lift-downtime reporting on top of a de-duplication and adjacent-event-merge step, not a raw log total, to avoid overstating downtime and understating uptime.
3. Plan staffing primarily around calendar effects (weekday/weekend, holidays), with same-day snowfall as a secondary, still-measurable adjustment factor, rather than the primary demand driver.

Limitations and Caveats

The following limitations apply to this analysis:

Fully synthetic dataset: every figure in this report and its accompanying dashboard derives from a fabricated season built to match the structure of a real independent ski resort's operations data, not from any real resort's actual figures. Absolute numbers (visitation, downtime, staffing ratios) are illustrative only.

Descriptive, not predictive: this is a current-state operations dashboard, not a demand-forecasting model. A next iteration could add next-day staffing forecasts driven by the weather feed already in the pipeline, but that was deliberately left out here to keep the dashboard focused on what happened and what is happening now, rather than what might happen next.

Static data source: the published dashboard reads a static JSON snapshot generated at build time, not a live database connection, appropriate for this demonstration but not the architecture a production deployment would use.

Single fabricated season: the dataset covers one modelled season; a real deployment would accumulate multiple seasons before any year-over-year comparison could be made reliably.

Peak-concurrent staffing simplification: daily staffing is aggregated as the maximum of two shift blocks (AM/PM) per department, a reasonable but simplified proxy for true concurrent headcount; a system with individual staff scheduling records could compute this more precisely.

AI Usage Disclosure

This report's production and formatting were completed with AI-assisted tools. All methodology, architecture, and analytical review were completed by the author.