Slide 1 — Preliminary Analysis & Findings — St. Martin Rd vs Chong San Rd (sugg

Verbatim:

"Hello. In this section I will present our preliminary analysis comparing St. Martin Road and Chong San Road, and then describe how we will simulate two intervention scenarios — stop merging, and route splitting. For data, we used real-time ETA snapshots from public APIs, consolidated into CSV files. Our 1-hour experiment produced 120 snapshots and a consolidated CSV under `vibeCoding101/PartX_simulation/monitor_outputs_1hr/`. First, I’ll summarize key metrics and observed patterns."

Slide 2 — Key metrics computed (suggested 0:25)

Verbatim:

"In our processing pipeline we compute several core metrics from each snapshot and ETA record. These include: wait time (wait_s = ETA - snapshot_ts), per-route ETA counts, the null-ETA rate, per-minute average wait, and a headway proxy derived from observed ETAs. We also compute a 'cleaned' average (mean_wait_cleaned) after trimming the top 1% of extreme waits to better represent typical conditions. These KPIs form the baseline we will compare against simulated scenarios."

Slide 3 — Wait time distribution (suggested 1:00)

Verbatim:

"Here is the wait-time distribution for the monitored period (see `monitor_outputs_60min/analysis/wait_distribution.png` or `monitor_outputs_until_0830/analysis/wait_distribution.png`). The distribution is right-skewed: most ETAs cluster at short waits—roughly within one to three minutes—but there is a long tail of larger waits. These long-tail values suggest episodes of congestion, vehicle bunching, or delayed reporting by the API. To reduce the influence of these extremes on summary statistics, we also show a trimmed mean (mean_wait_cleaned) for a more representative central tendency."

Slide 4 — Mean wait per minute (cleaned) (suggested 1:10)

Verbatim:

"This time-series plot (files: `mean_wait_cleaned.png` and `combined_mean_wait.png`) shows the minute-by-minute cleaned mean wait. You can also view 5- and 10-minute aggregated versions in the `counts_per_5min` and `counts_per_10min` folders to reduce noise. The main patterns are: average waits increase during peak windows; some snapshots show pronounced spikes indicating temporary instability; and aggregated views make these trends easier to interpret and compare across the two stops."

Slide 5 — Route counts (top 20) (suggested 0:50)

Verbatim:

"This bar chart (`route_counts_top20.png`) lists the most frequently observed routes in our monitoring window. A small set of routes appears far more frequently, indicating higher service frequency or route concentration near these stops. Importantly, if the same route appears in both stops within the same snapshot repeatedly, that can indicate short headways or bus bunching—an operational concern we consider when proposing stop consolidation or route adjustments."

Slide 6 — Null ETA & data issues (suggested 0:35)

Verbatim:

"We observed a non-trivial number of records where `eta == null` in API responses; raw examples can be found in `vibeCoding101/PartX_simulation/monitor_outputs_1hr/` snapshot JSONs. Null ETAs prevent direct wait computation and reduce effective sample size. To handle this, we compute and report a null-rate KPI, exclude null entries from numerical wait calculations, and flag snapshots for further inspection. Addressing nulls could require longer sampling, operator collaboration, or alternative data sources."

Slide 7 — Simulation objectives & contract (suggested 0:40)

Verbatim:

"The high-level objective is to quantify how two interventions change our KPIs relative to baseline: (A) merging the two bus stops into a single stop, and (B) splitting a frequent route so that half the vehicles serve St. Martin and half serve Chong San. Contract / inputs: observed headways and ETA-derived arrival distributions from `monitor_outputs_*`, route frequency by route, and estimated dwell times (set from literature or sensitivity ranges). Outputs: simulated passenger wait distributions, per-route counts at each stop, 90th percentile wait, and null-rate proxy. We will run stochastic replications to estimate means and 95% confidence intervals for these KPIs."

Slide 8 — Scenario A — Merge stops (suggested 0:50)

Verbatim:

"In Scenario A we model collapsing St. Martin and Chong San into one consolidated stop located at the midpoint. Operational assumptions: combined arrival streams for overlapping routes; slightly increased dwell time per stop due to higher boarding volume, modeled as dwell = base_dwell + alpha * arrivals. We will simulate multiple demand levels: current observed demand, +10% and +25% demand. Expected outputs to compare vs baseline: mean wait, 90th percentile wait, and passenger throughput. Hypothesis: merging reduces duplicated service but may increase dwell-driven delay; net effect depends on arrival regularity and boarding volume."

Slide 9 — Scenario B — Split route (half/half) (suggested 0:50)

Verbatim:

"In Scenario B we reassign an identified high-frequency route so roughly half of its vehicles serve St. Martin and half serve Chong San. Implementation assumptions: split operates by vehicle assignment rather than passenger rerouting; headways for each sub-route increase roughly twofold for the split segment. We will vary splitting rules (strict alternating assignment vs probabilistic split) and test sensitivity to compliance. Key comparison metrics: mean wait at each stop, variance of wait, and frequency of same-route co-occurrence. Hypothesis: splitting reduces per-stop crowding and wait variance but may increase average wait if headways grow too large."

Slide 10 — Model mechanics & validation (suggested 0:35)

Verbatim:

"Our simulator is a discrete-event model implemented in Python (see `sim_kmb_stops.py`). Core mechanics: event list of vehicle arrivals drawn from observed ETA-derived interarrival distributions; passenger arrivals modeled as Poisson with rate per-minute estimated from route counts; boarding consumes dwell time; service times and route assignments follow scenario rules. For validation, we compare simulator baseline output with empirical KPIs (mean wait, per-minute counts, route co-occurrence) to ensure the model reproduces observed behavior before running interventions."

Slide 11 — KPIs & visuals from simulation (suggested 0:30)

Verbatim:

"For each scenario we will present: baseline vs scenario KPI table with mean and 95% CI for mean wait and 90th percentile wait; overlaid wait-time histograms; time-series of mean wait per minute; and a small table reporting expected passenger throughput and dwell impact. Figures will be saved into `vibeCoding101/PartX_simulation/sim_outputs/<scenario>/` for reproducibility."

Slide 12 — Preliminary conclusions & next steps (suggested 0:25)

Verbatim:

"To summarize: empirical data shows long-tailed waits and signs of vehicle bunching—these motivate two interventions we will test with our simulator: merging stops and splitting routes. We will validate the simulator against observed KPIs, run sensitivity analyses for demand and dwell assumptions, and then compare KPI changes across scenarios. The next section will detail simulation results and policy recommendations. That concludes Part 3."

--- Appendix slides (optional) ---