Project framing

- **Novel network perspective:** We go beyond the classic airport-node view of air traffic by modeling each flight as a node connected to others via shared routes or feasible connections . This flight-centric network captures fine-grained structure that an airport network might hide (e.g. multiple airlines on the same route, temporal ordering of flights) .
- **Research questions/hypotheses:** We target 3–5 testable questions:
- *Community structure:* Does the flight-centric network naturally split into communities (clusters of flights) corresponding to airlines or geographic regions? (Hypothesis: flights cluster by airline operations or regional hubs.)
- *Connector importance:* Which individual flights act as critical bridges in the network? We expect certain hub-to-hub flights to have high centrality and connect otherwise separate parts of the network.
- *Network robustness:* How robust is the US flight network to disruptions? We hypothesize the network is resilient to random flight removals but highly sensitive to targeted failures at major hubs .
- *Delay propagation:* Do delays cascade through the flight network? Given that failures (e.g. weather delays) can ripple through airline schedules, we will test if a delay on a central flight causes widespread downstream delays.
- *Airline strategy vs performance:* How do different airline network strategies (hub-and-spoke vs point-to-point) affect operational reliability? For example, we expect hub-focused carriers to have more centralized networks (higher connectivity) but potentially larger cascades when disruptions occur, compared to more distributed networks.

Data and network representations

(A) Baseline airport network:
- **Nodes:** Airports (unique IATA/ICAO codes) that appear as an origin or destination in 2024 . Each node can have attributes like latitude/longitude, hub size category, state/region for context.
- **Edges:** Directed routes between airports. We draw an edge from airport *i* to *j* if at least one flight in the dataset flies *i* → *j* during the period . Edge weight w_{ij} equals the total number of flights from *i* to *j* in 2024 . (Additional edge attributes can include number of airlines on the route or average delay on that route.) This yields a weighted directed airport network – a familiar model in air-transport analysis . It serves as a baseline for comparison.

(B) Flight-as-node network:
- **Nodes:** Individual flight instances, each identified by a unique combination of airline, flight number, origin, destination, and departure time . Essentially each row in the on-time performance dataset (after cleaning) becomes a node. Node attributes include airline code, scheduled departure/arrival times, aircraft tail number, etc. (The dataset provides these details, e.g. arrival/departure times and delay indicators .)
- **Edges:** We link flights based on shared routes or connection logic:
- *Same-route links:* If two flights serve the same origin–destination pair (A→B), we connect them. This clusters all flights on a given route, regardless of airline or time, capturing parallel connections on that route .
- *Connection links:* If the destination of flight X is the origin of flight Y, and timing makes a feasible

connection (e.g. flight X arrives at airport M and flight Y departs M within X hours after), we add an edge X→Y . This represents a potential passenger or aircraft connection at airport M. We will choose a reasonable time window (e.g. 2–4 hours) to define "feasible" connections, to reflect realistic transfer opportunities or turnaround of the same plane.
- *Edge characteristics:* Flight–flight edges are directed (order matters for connections) and could be unweighted or weighted by connection strength (e.g. number of passengers likely connecting, if such data were available – here we may assume equal weight). These connections yield a much larger graph where flights are nodes linked by route similarity and schedule connectivity.

(C) Advanced representation – Multilayer network (airline layers):
- We model the system as a multilayer network with each layer corresponding to an airline's sub-network of flights. Within a layer, nodes are flights operated by that airline, and edges are defined as above (route-sharing or connections, but only among that airline's flights). Between layers, we can introduce inter-layer edges at shared airports to capture inter-airline connectivity (e.g. a connection from an American Airlines flight to a Delta flight at the same airport, if within the time window). In practice, inter-layer links may be sparse (most connections happen within the same carrier's network), but this representation lets us examine airline networks in parallel and their points of interaction.
- **Alternate representations considered:** *Temporal network* (treating flights as time-ordered edges in a dynamic network) and *bipartite network* (e.g. airline–airport affiliation) were considered. However, the airline-layered network is most relevant for comparing carriers' structures and simulating cross-airline disruptions. It allows analysis of each airline's network topology in isolation and in aggregate (e.g. identifying a given airport node that appears in multiple layers). We will leverage this multilayer model in advanced analyses (like resilience by airline).

Methods (advanced)

We will apply at least **two advanced network science techniques** to these representations, detailing for each: (i) the network used, (ii) inputs/features, (iii) expected outputs (plots/tables), and (iv) interpretation of results.

- **Community detection via Stochastic Block Model (SBM):** We will use a Bayesian or statistical block modeling approach (beyond simple modularity optimization) to detect communities of nodes purely from network structure .
- *(i) Network:* The flight-centric network (or its airline-specific layers) will be input to the SBM. We may run it on the full flight network or separately per airline layer to find communities of flights.
- *(ii) Inputs & features:* The method takes the adjacency matrix of the flight graph (potentially considering edge weights or multilayer structure) without using external labels. We could incorporate node attributes as features in a content-enhanced SBM if needed, but primarily the model will infer groups based on connectivity patterns (who connects to whom).
- *(iii) Outputs:* The SBM will output a partition of flights into communities (and estimates of connection probabilities between communities). We will present a summary of these communities – for example, the number of communities and their size, and interpretative labels. A figure or table will show whether communities correspond to meaningful groupings (e.g. predominantly flights of one airline, or flights centered on the same region/hub). We will also report metrics like the likelihood or entropy of the model and compare with a baseline (like modularity communities) for validation.

- *(iv) Interpretation:* If communities align strongly with airlines, it confirms that flight connectivity is largely structured by airline-specific schedules (homophily by airline). Alternatively, communities might be geographic (flights forming East-coast vs West-coast clusters, for instance). Unexpected

mixes would be insightful (e.g. a community containing flights from multiple airlines all linking a particular airport – suggesting an inter-airline hub effect). This method provides a nuanced view of network substructures beyond what modularity alone can capture, and helps answer our question about inherent clustering of the flight network.

- **Robustness and percolation analysis:** To evaluate network resilience, we will perform failure simulations, examining how the network breaks apart under node removals.

- *(i) Network:* Primarily the airport network (smaller and well-suited for connectivity analysis) will be used to test robustness. We'll also assess each airline layer's flight network separately (to see which airline's network is most fragile).
- *(ii) Inputs:* We will remove nodes or edges following certain strategies and measure the impact. Two main scenarios: (a) **Random failures** – remove a random fraction of airports or flights (simulating random cancellations) and (b) **Targeted attacks** – remove the highest-degree airports (worst-case hub failures) . We will use unweighted connectivity for giant component analysis, and consider edge weights for weighted traffic loss.
- *(iii) Outputs:* A plot of the giant component size (or % of total flights still connected) vs. fraction of nodes removed will illustrate the percolation threshold . We expect a curve that drops sharply when key hubs are removed (lower robustness) compared to a gradual decline for random removals. We'll also produce a table of "most critical nodes": e.g. which single airport removal disconnects the most flights or which airline's hub loss has the biggest impact on its network. These results will be documented with figures (percolation curves) and possibly a map highlighting critical airports.

- *(iv) Interpretation:* This analysis will quantify network resilience. A gentle slope under random removal would mean the network is resilient to random outages (redundancy exists), whereas a precipitous drop under targeted removal indicates vulnerability at hubs. We will interpret which airports are true single points of failure and discuss design implications (e.g. need for backup routes). Comparing layers, we might find that an airline like Southwest (more distributed network) maintains connectivity better under hub removal than a legacy carrier (highly hub-centric). This ties directly to our hypothesis on hub strategies affecting resilience.

- **Flow/contagion modeling of delay propagation:** We will model flight delays as a spreading process on the network, analogous to an epidemic spreading over a contact network.

- *(i) Network:* A **delay propagation network** derived from the flight connections will be used. Here, nodes are flights and directed edges represent a potential propagation path for delays (for example, if flight A's arrival feeds into flight B's departure via the same aircraft or passenger connection). We will build this graph using connection links (from part B) and also link flights that share the same aircraft tail number in sequence (since a late inbound aircraft can directly delay its next leg).
- *(ii) Inputs & features:* We will simulate delays using either a simple probabilistic contagion model or a compartmental model (SIS/SIR). For instance, treat an initial set of flights as "infected" with delay (e.g. a major weather delay affecting some flights), then at each time step propagate to connected flights with a certain transmission probability p (tunable). We can use empirical data to calibrate p (e.g. if historical data shows X% of connecting flights are delayed when inbound is late). Features like connection type might affect p (higher if same plane vs if just passenger connection). We'll likely run Monte Carlo simulations of this process.
- *(iii) Outputs:* We will measure cascade size distributions – e.g. how many flights become delayed in a cascade initiated by a given flight. One figure will show the distribution of delay cascade sizes on a log-log scale, expecting a heavy-tail (mostly small delays, but some large multi-flight

disruptions). Another result is identification of "super-spreader" flights or airports: for example, we'll calculate for each flight/node an influence score (the expected number of other flights it would delay if it were severely delayed first). This might be tabulated to highlight, say, the top 10 flights (often those out of major hubs at peak times) that would cause the largest knock-on delays. We may also produce a time-series visualization of one simulated cascade to illustrate how a delay in one hub propagates through the daily schedule.

- *(iv) Interpretation:* This modeling will reveal whether the network exhibits an epidemic-like threshold for delays. If even small initial delays can lead to unbounded cascades, the network operates near a critical point; if not, most delays remain localized. We'll interpret the heavy-tail distribution of delay events: e.g., "most delay events are contained (affect <5 flights), but a few rare events cause system-wide disruptions – consistent with cascade phenomena." We'll also discuss mitigation: the flights with high propagation centrality are likely candidates for intervention (additional slack or resources) to improve system reliability. This addresses our question on how delays spread and which parts of the network are most vulnerable to widespread disruption.

(In addition to the above, we remain open to incorporating graph embedding or ML techniques if time permits – for example, learning a graph embedding of airports or flights to predict new routes or classify delay risk. However, our priority is the domain-driven analyses above. Any ML/GNN application would be justified by specific needs, such as predicting flight delays using network position as a feature.)

Business analysis module

We will include a business-facing analysis translating our network findings into operational insights for airlines and stakeholders. Our chosen focus is **"Hub strategy and resilience"**: examining how airlines' network structures trade off efficiency and reliability, and what the cost of disruptions might be. Key components:

- **Hub concentration vs. resilience:** For each major airline, we will quantify its dependence on key hub airports. For example, we'll calculate the percentage of the airline's flights (or traffic) that passes through its top 1–2 hubs. This gives a proxy for how a hub outage would impact that airline. We will create a comparative table of airlines: e.g. Airline A operates 40% of flights via one airport (huge hub-dependence) whereas Airline B's largest hub handles only 15% (more distributed network). This metric directly indicates disruption risk – a highly concentrated network may be efficient for connectivity, but a single airport closure would cripple operations.

- **Disruption scenario analysis:** Using our network models, we simulate the impact of an airport shutdown or an airline bankruptcy . For instance, "What if Atlanta (ATL) shuts down for a day?" – we estimate how many flights each airline would cancel and the knock-on effects across the network. The results (perhaps shown in a chart of flights canceled by airline, or reduction in connectivity) serve as a proxy for disruption cost. If the current data doesn't include passenger counts or revenue, we will incorporate external data (e.g. average passengers per flight or estimated revenue per flight) to translate canceled flights into economic impact. For example, merging Bureau of Transportation Statistics data on average load factors or an airline's financial reports could allow us to say "closing ATL for a day might disrupt ~X thousand passengers, approximating $Y million in lost revenue."

- **Efficiency vs reliability:** We will also examine whether airlines with **aggressive operational efficiency** (e.g. very tight connection banks, minimal turnaround times) suffer worse on-time performance. Using the on-time dataset, we can compute each airline's average delay minutes and cancellation rate, and plot that against a network connectivity metric (like average degree of an airport in that airline's network or a centralization index). We expect to see if highly interconnected, tightly scheduled networks (often hub-and-spoke legacy carriers) tend to have more delays compared to more point-to-point ones. This analysis might reveal a trade-off: e.g. Airline X's hub strategy yields great connectivity (many routes, high centrality at hubs) but it has a lower on-time percentage than an airline

with a decentralized network.
- **Outputs for business module:** a concise set of exhibits – for instance, a bar chart of "hub dependence by airline" and a scatter plot of "network connectivity vs avg delay by airline". We will provide an interpretation in business terms: e.g. "Airline A's dominance at its hub is double-edged: operationally efficient for connectivity, but a single thunderstorm there can disrupt nearly half its network – suggesting a need for better contingency planning or secondary hubs." We'll also discuss potential strategies (like interline agreements or strategic slack) in the context of network resilience.
- **Data needs and integration:** The primary data (flight schedules, delays) covers reliability and network structure. If we find it insufficient to quantify *costs*, we will integrate public data on passengers or fares. For example, the T-100 traffic reports (passenger counts per route) or airline load factors can be joined by route to estimate how many travelers a disrupted flight represents. We'll document any such external data sources and how they are merged (by flight number, route, date, etc.). This ensures our business analysis is grounded in quantifiable terms (flights, passengers, dollars).

(Overall, this module translates technical findings into actionable insights on operational reliability, helping airlines balance their network design with service quality.)

Work plan (4 people)

We will organize the project into four parallel workstreams, each led by one team member, with defined responsibilities, deliverables, and integration points. The person initially assigned to data preparation will be reallocated to more analytical tasks now that the data is pre-cleaned.

1. **Workstream 1: Data & Network Construction**
2. **Lead & Responsibilities:** *Person A* (formerly in charge of data cleaning) will spearhead building the datasets into network form. They will load the 2024 on-time performance data, verify its integrity, and construct the baseline airport graph and flight-centric graph as specified.
3. **Tasks:** Filter data to the chosen time window (full year 2024) and scope (e.g. domestic flights), ensure no duplicates or errors. Create node and edge lists for the airport network (aggregate flights per route) and for the flight network (implement rules for edges as in section above). Also construct the multilayer structure (partition flights by airline). *Person A* will write code to generate these networks and calculate basic stats (number of nodes/edges, degree distributions) as a "sanity check."
4. **Interfaces:** They will produce cleaned, ready-to-use network data (adjacency matrices or edge lists) and share with others. This includes a documentation of data schema (e.g. node IDs for flights and airports). They must coordinate with *Person B* to ensure the network format is compatible with analysis tools (e.g. if *Person B* uses a specific community detection library, make sure node IDs are consistent).
5. **Outputs:** Verified network data files, and a brief report or slides on initial findings (e.g. "The airport network has N=300 nodes, E=2000 directed edges; largest hub is ATL with degree X; flight network has M nodes…"). This includes a table of top 10 airports by flights and perhaps a quick visualization of the airport network for reference. These form the basis of the methodology section in the report.

6. **Definition of Done:** All required networks are constructed correctly and efficiently (e.g. no flights missing, edge counts match raw data totals). We define completion as when the team agrees the network representations reflect the dataset (cross-check: does summing all airport-edge weights equal total flights? Does the degree of each airport match known operations?). At that point, *Person A* hands off the network datasets and moves on to assist with advanced

analysis (given data prep took less time than originally budgeted, they will contribute to e.g. running the robustness simulations).

7. **Workstream 2: Exploratory Analysis & Communities**

8. **Lead & Responsibilities:** *Person B* will delve into descriptive network analysis and community detection. This includes calculating classic network metrics and running the advanced community algorithms.
9. **Tasks:** Using the networks from WS1, compute centrality measures (degree, betweenness, etc.) for both airport and flight graphs. Identify key nodes (e.g. rank airports by centrality, find flights with highest degree in flight network). Next, implement community detection: first, a baseline like Louvain modularity on the airport network (for familiar baseline), then the advanced method (SBM or similar) on the flight network. *Person B* will likely use tools (e.g. NetworkX, graph-tool, or a Bayesian SBM package) to fit the block model. They will also perform any needed dimensionality reduction or subsetting if the flight network is very large (ensuring it's still representative).
10. **Interfaces:** Close collaboration with *Person A* on data issues (if any anomalies in network data arise during analysis). Also, *Person B* provides *Person D* with preliminary insights that feed the business angle – e.g. "Airline-based communities are observed" or "These 5 airports have extreme centrality." Such findings can inform what business metrics to compare. *Person B* will also coordinate with *Person C* to share any insights about which nodes are critical (helpful for designing targeted attack simulations in WS3).
11. **Outputs:** A set of plots and tables for the report's results section. For example: distribution plots (degree distribution of airport network), network visualizations (maybe a map of the US with nodes sized by centrality), a community membership summary (e.g. "SBM found 4 clusters, mainly grouping flights by airline – see Figure X"). They will produce a written interpretation for each result (which can later be refined for the report text). These outputs also include any validation of the community detection (e.g. checking modularity score, or comparing detected communities with known airline groupings to quantify alignment).

12. **Definition of Done:** This workstream is complete when all key exploratory questions are answered and at least one robust community detection result is obtained. Concretely: a list of top-20 hubs identified, community assignments that are stable (e.g. rerunning SBM yields similar groupings), and all necessary figures prepared. We will consider it done when *Person B* has documented the findings and no further tweaks to the community analysis are yielding new insights (i.e. diminishing returns reached). The deliverables should be ready to plug into the report.

13. **Workstream 3: Dynamic Simulations (Robustness & Delay Propagation)**

14. **Lead & Responsibilities:** *Person C* will focus on the advanced simulations: the network robustness tests and the delay contagion modeling. This is the technical core for risk analysis in our project.
15. **Tasks:** Implement the **percolation/robustness experiments** on the airport network: write a script to remove nodes in random order and record connectivity, and another for targeted removal (rank by degree or traffic and remove sequentially). This may involve multiple trial runs for random removal to get an average curve with variance. Also perform these simulations on each airline's sub-network (Person C will obtain subgraphs of the airport network or flight network filtered to one airline's operations). Next, set up the **delay propagation simulation**. Person C will use the flight connection network (from WS1) and possibly augment it with tail-number links (they'll need to derive sequences of flights by the same plane from the data). They

will then simulate an epidemic model (could code a custom BFS/DFS for cascades or use an existing library for contagion on networks). They will likely need to tune the model so that baseline matches reality (e.g. ensure that the fraction of flights delayed due to late connections in simulation is similar to that observed via the "LateAircraftDelay" fields in data).

16. **Interfaces:** Person C relies on Person A's data prep for correct network connectivity (especially for building the delay graph). They might need additional data from Person A such as a mapping of flight -> tail number and departure time to link planes. They will also discuss with Person D what specific disruption scenarios or outputs are of interest (for example, Person D might request "simulate if JFK airport flights all start delayed at 6 AM, what happens?" as a scenario to report). Also, Person C and Person B will cross-validate the identified "critical flights" – e.g. if Person B's centrality analysis and Person C's cascade simulation both flag the same flight or airport as critical, that strengthens the result.
17. **Outputs:** Several experiment results: (a) Plot of network connectivity vs nodes removed (for random and targeted) – likely included in the report to demonstrate robustness properties. (b) For each airline, a metric like "% of network disconnected by removal of top hub" to feed into the business analysis (handed to Person D). (c) Results of delay simulations – could be a chart of cascade size distribution, and example narrative of one cascade. Possibly also a heatmap or diagram showing how delays spread across airports (if data allows). All raw results will be distilled into clear visuals for the report. Additionally, Person C will produce a short technical memo on model assumptions (e.g. "p=0.3 was used as transmission probability based on X, simulation run 100 times for averaging") to ensure transparency and reproducibility.

18. **Definition of Done:** The dynamic analyses are done when we have answered: "How does the network break down under failures?" and "How do delays propagate?" to a satisfactory degree. Specifically, we expect to have stable curves (additional simulation runs don't change the outcome significantly) and insightful patterns (e.g. identification of a threshold or confirmation that delay cascades follow a power-law size distribution). Person C's work is considered complete when the outputs are validated (e.g. no coding bugs – confirmed by checking small cases or extreme scenarios) and the results support the statements we want to make (or we've noted why if a hypothesis wasn't supported). At that point, documentation is finalized and results are passed to Person D for inclusion in the storyline.

19. **Workstream 4: Business Analysis & Synthesis**

20. **Lead & Responsibilities:** *Person D* will bridge the technical results with business insights and compile the final deliverables. This includes performing the business-oriented analyses (using results from others and external data) and leading the writing of the report.
21. **Tasks:** Develop the **airline-level comparisons**: using input from WS2 and WS3, Person D will calculate metrics like each airline's hub concentration index, average delay, cancellation rate, etc. They will acquire any external datasets needed (for instance, if we need average passengers per flight, Person D will find that data and merge it by airline or route). They will create visualizations such as the bar chart of hub dependence and the scatter of connectivity vs reliability described in the business module. Person D is also responsible for the **authorship of the narrative**: integrating all findings into a coherent story. They will draft sections of the final report (Introduction, Business Implications, Conclusion) and ensure technical sections (methods, results) written by others are polished and consistent in tone. Additionally, Person D will coordinate the assembly of the **reproducibility package** (see next section), making sure code from all workstreams is organized and documented.
22. **Interfaces:** Person D will receive processed results from Persons B and C – e.g. the list of top hubs per airline, or the effect of removing a certain hub (from Person C's simulations). They will likely hold review meetings with the team after Checkpoint 2 to decide which findings are most

relevant for business discussion. Person D also acts as the editor, so each team member will hand over written summaries of their work, which Person D will integrate. If any result needs clarification or rerun, Person D will coordinate with the respective person to get it done quickly (essentially acting as project manager as well).

23. **Outputs:** The **business analysis deliverables** include a set of comparative tables/figures and a write-up connecting them to industry context. For example, an annotated chart comparing airlines will be produced, with notes on implications (these notes may appear as callouts or in the caption in the report). Person D will also produce the **draft and final report** and ensure all team members' contributions are properly credited. Another output is the slide deck for the final presentation (if required by the course), synthesizing key points – Person D will lead its creation with input slides from others.
24. **Definition of Done:** This workstream is finished when the final report and presentation are completed, reviewed by the team, and ready for submission. Specifically, all analysis points have been translated into plain-language insights, and all figures/tables have captions and explanations. We define done as having no open questions from the audience perspective: the report should read seamlessly, answer the research questions posed, and satisfy the course requirements (checked against the provided rubric). All team members will sign off on the final deliverables, facilitated by Person D.

Reproducibility package + final deliverables

Ensuring our work is reproducible and transparent is a priority. We will provide a well-organized code repository and documentation such that anyone (including graders) can reproduce our results and verify our analysis. Key components of the package and final outputs:

- **Repository structure:** We'll use a clear folder layout. For example:
- `/data/` – contains the input datasets (or links/instructions to obtain them, if data size is too large) and processed data files. The cleaned 2024 flight dataset and any external datasets (in CSV or similar) will be here, along with a data README describing each file.
- `/code/` – contains our analysis scripts, divided by workstream or functionality. Likely subfolders like `code/preprocessing/` (scripts to load and filter raw data, build networks), `code/analysis/` (scripts or notebooks for centrality, community detection, etc.), `code/simulations/` (for robustness and delay propagation code), and `code/business/` (any analysis combining results with external data). Each script will be documented with usage instructions.
- `/results/` – includes generated outputs such as intermediate analysis results (e.g. community assignments, simulation outputs) and the final figures and tables used in the report. We will store raw outputs in machine-readable form (CSV/JSON for tables, PNG/PDF for figures) for traceability.

- `/docs/` – will contain the final report in PDF and any supporting documentation (like the environment file and contribution statement). If a Jupyter notebook is used for analysis, an exported HTML/PDF of it may also be included for reference.

- **Environment and dependencies:** We will provide an `environment.yml` (for Conda) or `requirements.txt` (for pip) listing all packages and versions needed to run our code – for example: Python 3.x, NetworkX (for graph operations), pandas, numpy, plus any specialized libraries (e.g. graph-tool or pySBM for community detection, or PyTorch if a GNN was used). This will allow the TA to create an identical environment. We'll also document any system requirements (if a library needs e.g. C++ runtime, we'll note that). The environment file ensures deterministic setup, and we will test that a fresh environment using it can indeed run our code from scratch.

- **Reproducibility practices:** In our code, we will set random seeds for stochastic processes (community detection algorithms, random removal simulations, etc.) to make results deterministic when needed. For example, we'll use a fixed seed for the final run of the SBM and simulations so that the figures can be regenerated exactly. We will include instructions on how to reproduce each figure: e.g. "Run `code/analysis/03_run_robustness.py` to regenerate Fig. 4." Where possible, we'll include small "unit tests" or assertions in code (for instance, after building the network, assert that sum of out-degrees equals total flights count as a data consistency check). These help catch any discrepancies and give confidence in the results.

- **Final figures and tables:** We will list and describe the key figures and tables in the report, aligning them to the narrative. Tentatively:

- *Figure 1:* Schematic of the flight-centric network (illustration of how flights are nodes connecting via airports) – for introduction.
- *Figure 2:* Degree distribution of the airport network (to show hub-and-spoke nature). Possibly alongside, a similar distribution for flight node degrees (showing most flights have few connections, some have many).
- *Figure 3:* Map or network diagram highlighting top central airports (satisfying curiosity and providing visual context of network geography).
- *Figure 4:* Community detection result – e.g. a visualization of the flight network colored by community, or a bar chart showing composition of each community by airline (to demonstrate communities align with certain airlines).
- *Figure 5:* Robustness curves (giant component vs removed fraction) for random vs targeted removals . Possibly one panel for all-network, another comparing airlines.
- *Figure 6:* Delay cascade size distribution (log-log plot showing many small, few large cascades). And/or a timeline example of a cascade.
- *Figure 7:* Bar chart of hub dependence by airline (business insight chart – e.g. fraction of flights via largest hub for each airline).
- *Figure 8:* Scatter plot of airline connectivity vs average delay (each point an airline, illustrating efficiency vs reliability trade-off).

- *(Plus a few tables)*: e.g. Table of top 10 flights by betweenness (with flight details), Table of community summary (community size and dominant airline), Table of simulation outcomes for specific scenarios (like "closing airport X removes Y flights, affecting Z% of network").
  Each figure/table will be labeled and referenced in the text where the insight is discussed. We ensure that every research question posed is addressed by one or more of these exhibits in the storyline.

- **Authorship and contribution statement:** We will include a clear section in the report listing each team member and their contributions, as per academic guidelines. For instance: *Person A* – data processing and network construction; *Person B* – network analysis and community detection; *Person C* – simulations and advanced modeling; *Person D* – business analysis and report integration. We'll also mention that all members contributed to ideation and results interpretation. Additionally, we'll specify who wrote which report sections (e.g. Methodology by A/B/C for their parts, Business section by D, etc.) to provide transparency. This statement will be agreed upon by all team members.

- **Validation and checks:** In the reproducibility documentation, we will note how we validated our results. For example, we might mention comparing our airport network degree distribution to known statistics from literature as a sanity check, or that we tested our delay simulation on a smaller subset (one day's flights) where we could manually trace delays. Including these notes shows the steps we took to ensure the analysis is sound and can be trusted when reproduced.

By delivering the above package, we ensure anyone can follow our work step-by-step: from raw data to processed networks to final figures. This not only fulfills course requirements but leaves a durable resource for future students or researchers interested in flight networks. All final deliverables – the written report, the code repo (zipped or link), and the presentation slides – will be submitted per the course instructions, on time and in the expected format.