Project Questions:

- Regional patterns for delays?
- Different standards per state for different weather?
- What weather causes the most delays? Per state?
- Delays by aircraft carrier?
- Delays by airport?
- Delays by day (holiday)?

Completed:

- □ 0. Merge all flight months to create canceled flights (includes canceled and diverted flights) and departed flights datasets **Meghan**
- □ 1. Change FL_DATE to not include the 0:00 at the end **David/Stephen/Talia**
- □ 2. Change the scheduled departure time and actual departure time and wheels off to time format eg: 658 -> 6:58   1543 - > 15:43   56 -> 0:56 **Stephen**
- □ 3. Change those three (schedule dep, actual dep, wheels off) to include FL_DATE and become actual datetimes in the same form as the weather dataset eg: FL_DATE: 12/2/2021, scheduled_departure: 7:58 -> scheduled_departure: 12/2/2021 7:58 **Stephen**
  - □ 3.5. Run code on all dataframe with all months
- □ GitHub repository link **Meghan**
- □

To Do:

- □ 3.8 Redo all Python code into Pyspark **David/Talia** 1/20
- □ 4. Connect that dataframe to a dataframe merged with timezone info/ running the code to turn the times into datetimes on the allmonths and timezones dataset
- □ 5. Use timezone information to change all times to UTC time **Talia** 1/20
- □ 6. Change the datetimes if its overnight so the day goes up if its an overnight flight   eg: departure_time: 2/28/2021 20:45 arrival_time: 2/28/2021 4:25 -> arrival_time: 3/1/2021 4:25 **David/Talia** 1/20
- □ 7. If the above doesn’t work, change scheduled arrival time to be updated scheduled departure time + scheduled elapsed time (Time zone issue? Day Issue?) same for wheels on = wheels off + time in air and arrival time = departure time + actual elapsed time. These columns need to be in date time format as well eg: “6/24/2021 15:02” **David/Talia** 1/20
- □ 8. Then we should be ready to merge with the weather dataset with *“For row in flights: if flights[OriginCity] == weather[City] and weather[starttime] <= flights[ScheduledDepartureTime] and weather[endtime] >=*

flights[ScheduledDepartureTime]: append weather[Row] to row in flights” and *“For row in flights: if flights[DestinationCity] == weather[City] and weather[starttime] <= flights[ScheduledArrivalTime] and weather[endtime] >= flights[ScheduledArrivalTime]: append weather[Row] to row in flights”* 1/20-1/23

- □ 9. Clean full merged dataset (drop columns and etc) **Group** 1/23
- □ 10. Produce ETL Report **Group** 1/23-1/25
- □ 11. Create predictive model with full data (Random Forest) **Group** 1/24
- □ 12. Create predictive model with data assuming you only have the information from before the flight takes off (Random Forest) **Group** 1/25
- □ 13. Find data for Kafka steps **Group** 1/26
- □ 14. Create Kafka consumer and producers **Group** 1/26
- □ 15. Continue as we see fit 1/27

To Do (Independent):

- □ 1. Dashboard in PowerBI with flight data **Meghan** 1/25
- □ 2. EDA with flight data **Meghan** 1/25
- □ 3. EDA with weather data **Meghan** 1/25
- □ 4. Napkin Drawing **Meghan** 1/24
- □ 5.

Communication:

- Teams (send resources, keep each other updated when not on zoom)
- Google Doc (PMP, useful code, relevant resources for specific topics)
- Databricks (individual code that we are working on in our own databricks, important / complete code in Stephen’s Capstone folder)
- Zoom / Teams to meet and go over each other's work