Team-11 Members: Gege Yao (UIN: 668250011), Kumari Neha Priya (UIN: 653098178), Celeste Huang (UIN: 670733154)

Assess

1. Data Preprocessing

Import and read CSV dataset

```
In[∘]:= rawDataCSV = Import[
           "https://raw.githubusercontent.com/Kumari-Neha-Priya/BDI-513/refs/heads/main/flights_sample_10k.csv", "CSV"];
```

Print column names and rows data

```
In[∘]:= columnNames = rawDataCSV⟦1⟧; (*First row as column names*)
        dataRows = rawDataCSV⟦2 ;;⟧; (*Remaining rows as data*)
```

```
In[∘]:= columnNames (*Preview all columns*)
Out[∘]=
        {FL_DATE, AIRLINE, AIRLINE_DOT, AIRLINE_CODE, DOT_CODE, FL_NUMBER, ORIGIN, ORIGIN_CITY, DEST, DEST_CITY,
         CRS_DEP_TIME, DEP_TIME, DEP_DELAY, TAXI_OUT, WHEELS_OFF, WHEELS_ON, TAXI_IN, CRS_ARR_TIME, ARR_TIME,
         ARR_DELAY, CANCELLED, CANCELLATION_CODE, DIVERTED, CRS_ELAPSED_TIME, ELAPSED_TIME, AIR_TIME, DISTANCE,
         DELAY_DUE_CARRIER, DELAY_DUE_WEATHER, DELAY_DUE_NAS, DELAY_DUE_SECURITY, DELAY_DUE_LATE_AIRCRAFT}
```

```
In[•]:= dataRows⟦;; 5⟧ (*Preview first 5 rows*)
Out[•]=
  {{2022-07-19, SkyWest Airlines Inc., SkyWest Airlines Inc.: OO, OO, 20 304, 3371, SAN, San Diego, CA, SFO,
    San Francisco, CA, 1705, 1700., -5., 13., 1713., 1816., 11., 1834, 1827., -7., 0., , 0., 89., 87., 63., 447., , , , , },
   {2022-09-13, Republic Airline, Republic Airline: YX, YX, 20 452, 3552, CMH, Columbus, OH, ORD, Chicago, IL,
    1119, 1118., -1., 16., 1134., 1125., 11., 1147, 1136., -11., 0., , 0., 88., 78., 51., 296., , , , , },
   {2022-07-09, SkyWest Airlines Inc., SkyWest Airlines Inc.: OO, OO, 20 304, 4660, CVG, Cincinnati, OH, ORD,
    Chicago, IL, 1118, 1113., -5., 18., 1131., 1124., 8., 1155, 1132., -23., 0., , 0., 97., 79., 53., 264., , , , , },
   {2021-03-19, United Air Lines Inc., United Air Lines Inc.: UA, UA, 19 977, 325, DEN, Denver, CO, MCI,
    Kansas City, MO, 1815, 1815., 0., 15., 1830., 2042., 5., 2051, 2047., -4., 0., , 0., 96., 92., 72., 533., , , , , },
   {2020-01-03, United Air Lines Inc., United Air Lines Inc.: UA, UA, 19 977, 1561, IAH, Houston, TX, SFO,
    San Francisco, CA, 1000, 1001., 1., 21., 1022., 1205., 9., 1227, 1214., -13., 0., , 0., 267., 253., 223., 1635., , , , , }}
```

View the structure/dimension of the dataset

```
In[•]:= (*Total Count of Rows and Columns*)
        rowCount = Length[dataRows];
        columnCount = Length[columnNames];
        Print["Total Rows: ", rowCount];
        Print["Total Columns: ", columnCount];

        Total Rows: 10 000

        Total Columns: 32
```

Read the metadata of the dataset

```
In[ ]:= (*Import the HTML file*)
htmlData =
  Import["https://raw.githubusercontent.com/Kumari-Neha-Priya/BDI-513/refs/heads/main/dictionary.html", "Data"];

(*Extract all tables from the HTML data*)
tables = Cases[htmlData, {__List}, Infinity];

(*Extract the first table and its column headers*)
table = tables⟦1⟧; columnHeaders = table⟦1⟧;

(*Extract relevant data from the first and last columns*)
firstColumnIndex = 1;
lastColumnIndex = Length[columnHeaders];
relevantRows = Map[Function[row, {row⟦firstColumnIndex⟧, row⟦lastColumnIndex⟧}], Rest[table]];

(*Convert the relevant rows to a Dataset for better formatting and add headers*)
relevantData = Dataset[Prepend[relevantRows, {columnHeaders⟦firstColumnIndex⟧, columnHeaders⟦lastColumnIndex⟧}]];

(*Display the dataset*)relevantData
```

Out[∘]=

FL_DATE	Flight Date (yyyymmdd)
AIRLINE_CODE	Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffi
DOT_CODE	An identification number assigned by US DOT to identify a unique airline (carrier). A unique ai
FL_NUMBER	Flight Number
ORIGIN	Origin Airport
ORIGIN_CITY	Origin Airport, City Name
DEST	Destination Airport
DEST_CITY	Destination Airport, City Name
CRS_DEP_TIME	CRS Departure Time (local time: hhmm)
DEP_TIME	Actual Departure Time (local time: hhmm)
DEP_DELAY	Difference in minutes between scheduled and actual departure time. Early departures show n
TAXI_OUT	Taxi Out Time, in Minutes
WHEELS_OFF	Wheels Off Time (local time: hhmm)
WHEELS_ON	Wheels On Time (local time: hhmm)
TAXI_IN	Taxi In Time, in Minutes
CRS_ARR_TIME	CRS Arrival Time (local time: hhmm)
ARR_TIME	Actual Arrival Time (local time: hhmm)
ARR_DELAY	Difference in minutes between scheduled and actual arrival time. Early arrivals show negative
CANCELLED	Cancelled Flight Indicator (1=Yes)
CANCELLATION_CODE	Specifies The Reason For Cancellation

rows 1–20 of **30**

Count missing data in columns in the dataset

```
In[ ]:= (*Count Missing Values Column-wise*)
missingValues = Table[With[{column = columnNames⟦colIndex⟧, (*Column Name*) values = dataRows⟦All, colIndex⟧
      (*Column Values*)}, <|"Column" → column, "MissingCount" → Count[values, _Missing | Null | ""],
      "PercentageMissing" → N[Count[values, _Missing | Null | ""]] / rowCount * 100,
      "DataType" → If[Length[DeleteCases[values, _Missing | Null | ""]] > 0,
        Head[First[DeleteCases[values, _Missing | Null | ""]]], "Unknown"]|>], {colIndex, columnCount}];

(*Output as Dataset for Easy Viewing*)
Dataset[missingValues]
```

Out[∘]=

Column	MissingCount	PercentageMissing	DataType
FL_DATE	0	0.0	String
AIRLINE	0	0.0	String
AIRLINE_DOT	0	0.0	String
AIRLINE_CODE	0	0.0	String
DOT_CODE	0	0.0	Integer
FL_NUMBER	0	0.0	Integer
ORIGIN	0	0.0	String
ORIGIN_CITY	0	0.0	String
DEST	0	0.0	String
DEST_CITY	0	0.0	String
CRS_DEP_TIME	0	0.0	Integer
DEP_TIME	238	2.38	Real
DEP_DELAY	238	2.38	Real
TAXI_OUT	241	2.41	Real
WHEELS_OFF	241	2.41	Real
WHEELS_ON	246	2.46	Real
TAXI_IN	246	2.46	Real
CRS_ARR_TIME	0	0.0	Integer
ARR_TIME	246	2.46	Real
ARR_DELAY	261	2.61	Real

rows 1–20 of **32**

Drop the column CANCELLATION_CODE and fill the missing values with 0

```
In[∘]:= (*Define the original column names in their original order*)
        originalColumnNames = {"FL_DATE", "AIRLINE", "AIRLINE_DOT", "AIRLINE_CODE", "DOT_CODE", "FL_NUMBER", "ORIGIN",
           "ORIGIN_CITY", "DEST", "DEST_CITY", "CRS_DEP_TIME", "DEP_TIME", "DEP_DELAY", "TAXI_OUT", "WHEELS_OFF",
           "WHEELS_ON", "TAXI_IN", "CRS_ARR_TIME", "ARR_TIME", "ARR_DELAY", "CANCELLED", "CANCELLATION_CODE",
           "DIVERTED", "CRS_ELAPSED_TIME", "ELAPSED_TIME", "AIR_TIME", "DISTANCE", "DELAY_DUE_CARRIER",
           "DELAY_DUE_WEATHER", "DELAY_DUE_NAS", "DELAY_DUE_SECURITY", "DELAY_DUE_LATE_AIRCRAFT"};

        (*Define the columns to drop*)
        columnsToDrop = {"CANCELLATION_CODE"};

        (*Update column names to remove the specified columns*)
        columnNames = DeleteCases[originalColumnNames, Alternatives @@ columnsToDrop];

        (*Convert raw data to associations*)
        dataRows = Map[AssociationThread[originalColumnNames, #] &, rawDataCSV[[2 ;;]]];

        (*Drop the specified columns*)
        dataRows = Map[KeyDrop[#, columnsToDrop] &, dataRows];

        (*Ensure the new column order is maintained*)
        dataRows = Map[AssociationThread[columnNames, Lookup[#, columnNames]] &, dataRows];

        (*Replace Empty or Missing Values with 0*)
        filledDataRows = Map[Map[If[MissingQ[#] || # === Null || # === "", 0, #] &, #] &, dataRows];
```

View the structure/dimension of the dataset after dropping the column CANCELLATION_CODE

```
In[∘]:= (*Total Count of Rows and Columns*)
        rowCount = Length[dataRows];
        columnCount = Length[columnNames];
        Print["Total Rows: ", rowCount];
        Print["Total Columns: ", columnCount];
```

```
Total Rows: 10 000

Total Columns: 31
```

Count missing data in columns in the dataset after filling all missing values

In[∘]:=

```
(*Count Missing Values Column-wise after dropping all missing values*)
missingValues = Table[With[{column = columnNames⟦colIndex⟧, (*Column Name*) values = filledDataRows⟦All, colIndex⟧
      (*Column Values*)}, <|"Column" → column, "MissingCount" → Count[values, _Missing | Null | ""],
      "PercentageMissing" → N[Count[values, _Missing | Null | ""]] / rowCount * 100,
      "DataType" → If[Length[DeleteCases[values, _Missing | Null | ""]] > 0,
        Head[First[DeleteCases[values, _Missing | Null | ""]]], "Unknown"] |>], {colIndex, columnCount}];

(*Output as Dataset for Easy Viewing*)
Dataset[missingValues]
```

Out[∘]=

Column	MissingCount	PercentageMissing	DataType
FL_DATE	0	0.0	String
AIRLINE	0	0.0	String
AIRLINE_DOT	0	0.0	String
AIRLINE_CODE	0	0.0	String
DOT_CODE	0	0.0	Integer
FL_NUMBER	0	0.0	Integer
ORIGIN	0	0.0	String
ORIGIN_CITY	0	0.0	String
DEST	0	0.0	String
DEST_CITY	0	0.0	String
CRS_DEP_TIME	0	0.0	Integer
DEP_TIME	0	0.0	Real
DEP_DELAY	0	0.0	Real
TAXI_OUT	0	0.0	Real
WHEELS_OFF	0	0.0	Real
WHEELS_ON	0	0.0	Real
TAXI_IN	0	0.0	Real
CRS_ARR_TIME	0	0.0	Integer
ARR_TIME	0	0.0	Real
ARR_DELAY	0	0.0	Real

rows 1–20 of **31**

Verify the structure of the cleaned dataset after previous modifications

```
In[ ]:= (*Check if dataRows is loaded correctly*)
        Print[Head[dataRows]]; (*Check the structure of dataRows*)
        Print[Length[dataRows]]; (*Length of dataRows*)
        Print[dataRows⟦1⟧ ] (*Print the first row to check its structure*)
```

```
List

10 000

<|FL_DATE → 2022-07-19, AIRLINE → SkyWest Airlines Inc., AIRLINE_DOT → SkyWest Airlines Inc.: OO,
 AIRLINE_CODE → OO, DOT_CODE → 20 304, FL_NUMBER → 3371, ORIGIN → SAN, ORIGIN_CITY → San Diego, CA, DEST → SFO,
 DEST_CITY → San Francisco, CA, CRS_DEP_TIME → 1705, DEP_TIME → 1700., DEP_DELAY → -5., TAXI_OUT → 13.,
 WHEELS_OFF → 1713., WHEELS_ON → 1816., TAXI_IN → 11., CRS_ARR_TIME → 1834, ARR_TIME → 1827., ARR_DELAY → -7.,
 CANCELLED → 0., DIVERTED → 0., CRS_ELAPSED_TIME → 89., ELAPSED_TIME → 87., AIR_TIME → 63., DISTANCE → 447.,
 DELAY_DUE_CARRIER → , DELAY_DUE_WEATHER → , DELAY_DUE_NAS → , DELAY_DUE_SECURITY → , DELAY_DUE_LATE_AIRCRAFT → |>
```

```
In[ ]:= (*Check the structure of filledDataRows*)
        Print[Head[filledDataRows]]; (*Should be a list*)
        Print[Head[filledDataRows⟦1⟧]]; (*Should be an Association*)
        Print[Length[filledDataRows]]; (*Should be 10000 rows*)
        Print[filledDataRows⟦1⟧]; (*Should display the first row with all columns*)
```

```
List

Association

10 000

<|FL_DATE → 2022-07-19, AIRLINE → SkyWest Airlines Inc., AIRLINE_DOT → SkyWest Airlines Inc.: OO,
 AIRLINE_CODE → OO, DOT_CODE → 20 304, FL_NUMBER → 3371, ORIGIN → SAN, ORIGIN_CITY → San Diego, CA, DEST → SFO,
 DEST_CITY → San Francisco, CA, CRS_DEP_TIME → 1705, DEP_TIME → 1700., DEP_DELAY → -5., TAXI_OUT → 13.,
 WHEELS_OFF → 1713., WHEELS_ON → 1816., TAXI_IN → 11., CRS_ARR_TIME → 1834, ARR_TIME → 1827., ARR_DELAY → -7.,
 CANCELLED → 0., DIVERTED → 0., CRS_ELAPSED_TIME → 89., ELAPSED_TIME → 87., AIR_TIME → 63., DISTANCE → 447.,
 DELAY_DUE_CARRIER → 0, DELAY_DUE_WEATHER → 0, DELAY_DUE_NAS → 0, DELAY_DUE_SECURITY → 0, DELAY_DUE_LATE_AIRCRAFT → 0|>
```

Convert real types values to integer type and view the dataset rows data

```
In[ ]:= (*Automatically detect numeric columns from the first row of filledDataRows*)
numericColumns = Select[Keys[filledDataRows⟦1⟧], StringMatchQ[ToString[filledDataRows⟦1, #⟧], NumberString] &];

(*Convert numeric strings to integers while retaining non-numeric fields*)
correctedData =
  Map[Function[row, AssociationMap[If[MemberQ[numericColumns, #], (*Convert to integer if numeric column*)
       If[StringMatchQ[ToString[row[#]], NumberString], IntegerPart[ToExpression[row[#]]], 0],
       (*Keep original value for non-numeric columns*)row[#]] &, Keys[row]]], filledDataRows];

(*Convert corrected data into a proper Dataset*)
dataset = Dataset[correctedData];

(*Display the first 50 rows of the dataset*)
dataset⟦;; 50⟧
```

Out[∘]=

FL_DATE	AIRLINE	AIRLINE_DOT
2022–07–19	SkyWest Airlines Inc.	SkyWest Airlines Inc.: OO
2022–09–13	Republic Airline	Republic Airline: YX
2022–07–09	SkyWest Airlines Inc.	SkyWest Airlines Inc.: OO
2021–03–19	United Air Lines Inc.	United Air Lines Inc.: UA
2020–01–03	United Air Lines Inc.	United Air Lines Inc.: UA
2023–06–18	SkyWest Airlines Inc.	SkyWest Airlines Inc.: OO
2023–05–20	SkyWest Airlines Inc.	SkyWest Airlines Inc.: OO
2019–09–23	PSA Airlines Inc.	PSA Airlines Inc.: OH
2022–04–23	Endeavor Air Inc.	Endeavor Air Inc.: 9E
2019–11–22	Spirit Air Lines	Spirit Air Lines: NK
2022–01–28	Delta Air Lines Inc.	Delta Air Lines Inc.: DL
2021–11–27	American Airlines Inc.	American Airlines Inc.: AA
2019–11–26	Envoy Air	Envoy Air: MQ
2020–10–04	SkyWest Airlines Inc.	SkyWest Airlines Inc.: OO
2021–08–10	Mesa Airlines Inc.	Mesa Airlines Inc.: YV
2022–07–26	United Air Lines Inc.	United Air Lines Inc.: UA
2019–08–01	Frontier Airlines Inc.	Frontier Airlines Inc.: F9
2019–12–04	Southwest Airlines Co.	Southwest Airlines Co.: WN
2023–06–06	Republic Airline	Republic Airline: YX
2023–03–14	Delta Air Lines Inc.	Delta Air Lines Inc.: DL

rows 1–20 of **50** | columns 1–10 of **31**

Count missing data in columns and verify the data type of all columns in the dataset after converting real type numbers to integer type

```
In[∘]:= (*Count Missing Values Column-wise after converting all real type numbers to integer type*)
missingValues = Table[With[{column = Keys[correctedData⟦1⟧]⟦colIndex⟧, values = correctedData⟦All,
         Keys[correctedData⟦1⟧]⟦colIndex⟧⟧}, <|"Column" → column, "MissingCount" → Count[values, _Missing | Null | ""],
      "PercentageMissing" → N[Count[values, _Missing | Null | ""]] / Length[correctedData] * 100, "DataType" →
       If[Length[DeleteCases[values, _Missing | Null | ""]] > 0, Head[First[DeleteCases[values, _Missing | Null | ""]]],
         "Unknown"]|>], {colIndex, Length[Keys[correctedData⟦1⟧]]}];

(*Output as Dataset for Easy Viewing*)
Dataset[missingValues]
```

Out[•]=

Column	MissingCount	PercentageMissing	DataType
FL_DATE	0	0.0	String
AIRLINE	0	0.0	String
AIRLINE_DOT	0	0.0	String
AIRLINE_CODE	0	0.0	String
DOT_CODE	0	0.0	Integer
FL_NUMBER	0	0.0	Integer
ORIGIN	0	0.0	String
ORIGIN_CITY	0	0.0	String
DEST	0	0.0	String
DEST_CITY	0	0.0	String
CRS_DEP_TIME	0	0.0	Integer
DEP_TIME	0	0.0	Integer
DEP_DELAY	0	0.0	Integer
TAXI_OUT	0	0.0	Integer
WHEELS_OFF	0	0.0	Integer
WHEELS_ON	0	0.0	Integer
TAXI_IN	0	0.0	Integer
CRS_ARR_TIME	0	0.0	Integer
ARR_TIME	0	0.0	Integer
ARR_DELAY	0	0.0	Integer

rows 1–20 of **31**

Benchmarking and Classification

2. Exploratory Data Analyses (EDA): Statistical Visualizations

These analyses help explore trends, distributions, and relationships in the data in the dataset

For Question 1 : Which airport/airline is more likely to experience delays and cancellations?

a. Airports and their frequencies for flights:

Purpose and Insights : The airport frequency plot helps visualize the distribution of flights across different airports in the dataset . The top 10 airports and their frequencies plot helps identify the most represented airports in the dataset, prioritize them for further analysis of on-time performance and delays, and assess which ones are more likely to have higher delay or cancellation rates.

Outbound (Departing) flights

In[]:=

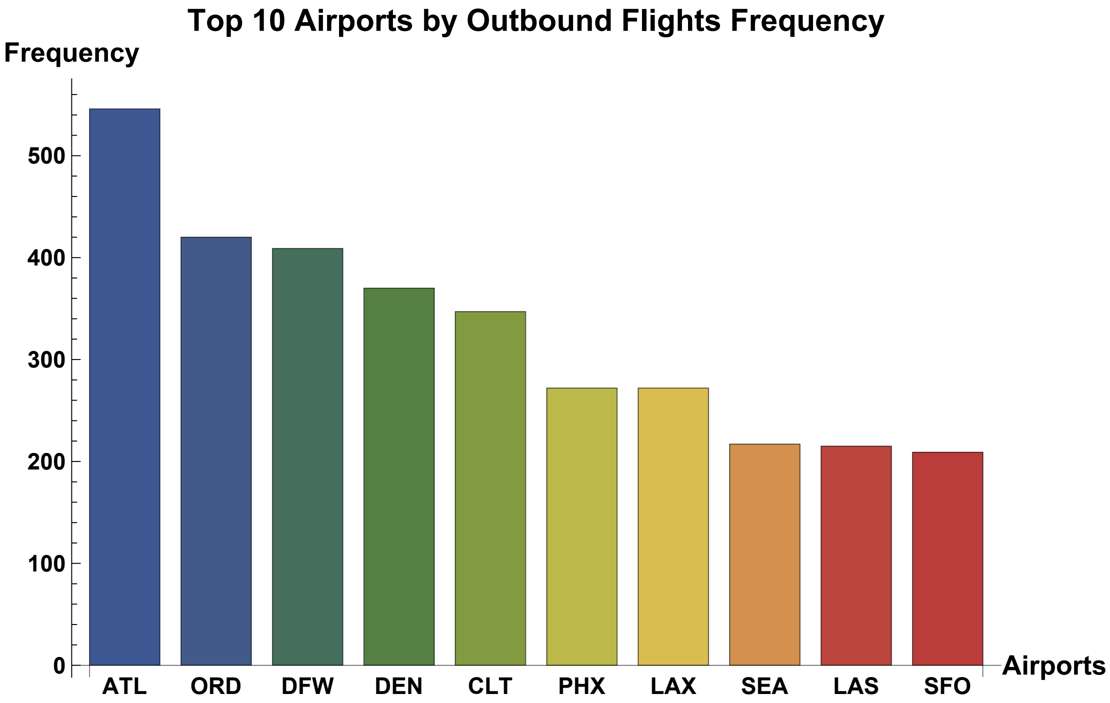
```
(*Outbound flights*)
(*Extract the'ORIGIN' column to get airports*)
airportData = Normal[dataset[All, "ORIGIN"]];

(*Count the frequency of each airport*)
airportCounts = Tally[airportData];

(*Sort by frequency in descending order*)
sortedAirportCounts = ReverseSortBy[airportCounts, Last];

(*Separate airports and their frequencies*)
airports = sortedAirportCounts⟦All, 1⟧;
frequencies = sortedAirportCounts⟦All, 2⟧;

(*Select top 10 airports based on frequency*)
top10Airports = Take[sortedAirportCounts, 10];
top10AirportsNames = top10Airports⟦All, 1⟧;
top10AirportsFrequencies = top10Airports⟦All, 2⟧;

(*Create a table for display*)
airportTable = Dataset[AssociationThread[{"Airport", "Frequency"} → #] & /@ top10Airports];

(*Show the table*)
Print["Top 10 Airports by Frequency (Outbound):"];
airportTable

(*Create a bar chart with vertically oriented airport names*)
BarChart[top10AirportsFrequencies, ChartLabels → Placed[Rotate[#, 0 Degree] & /@ top10AirportsNames, Below],
 ChartStyle → "DarkRainbow", BarSpacing → 0.3, LabelStyle → {FontSize → 12, Bold}, ImageSize → Large,
 AxesLabel → {Style["Airports", FontSize → 14, Bold], Style["Frequency", FontSize → 14, Bold]},
 TicksStyle → Directive[FontSize → 10],
 PlotLabel → Style["Top 10 Airports by Outbound Flights Frequency", FontSize → 16, Bold]]
```

```
Top 10 Airports by Frequency (Outbound):
```

Out[•]=

Airport	Frequency
ATL	546
ORD	420
DFW	409
DEN	370
CLT	347
PHX	272
LAX	272
SEA	217
LAS	215
SFO	209

Out[•]=



Purpose: The bar graph displays the top 10 airports with the highest number of outbound flights. It ranks the airports by frequency to identify the busiest airports in the dataset.

Y-Axis (Frequency):

- Represents the number of outbound flights for each airport.
- The scale starts at 0 and increases in consistent intervals, making it easy to compare the frequencies visually.

X-Axis (Airports):

- Lists the airport codes (e.g., ATL, ORD, DFW) for the top 10 busiest airports.
- Each code corresponds to a unique airport.

Insights:

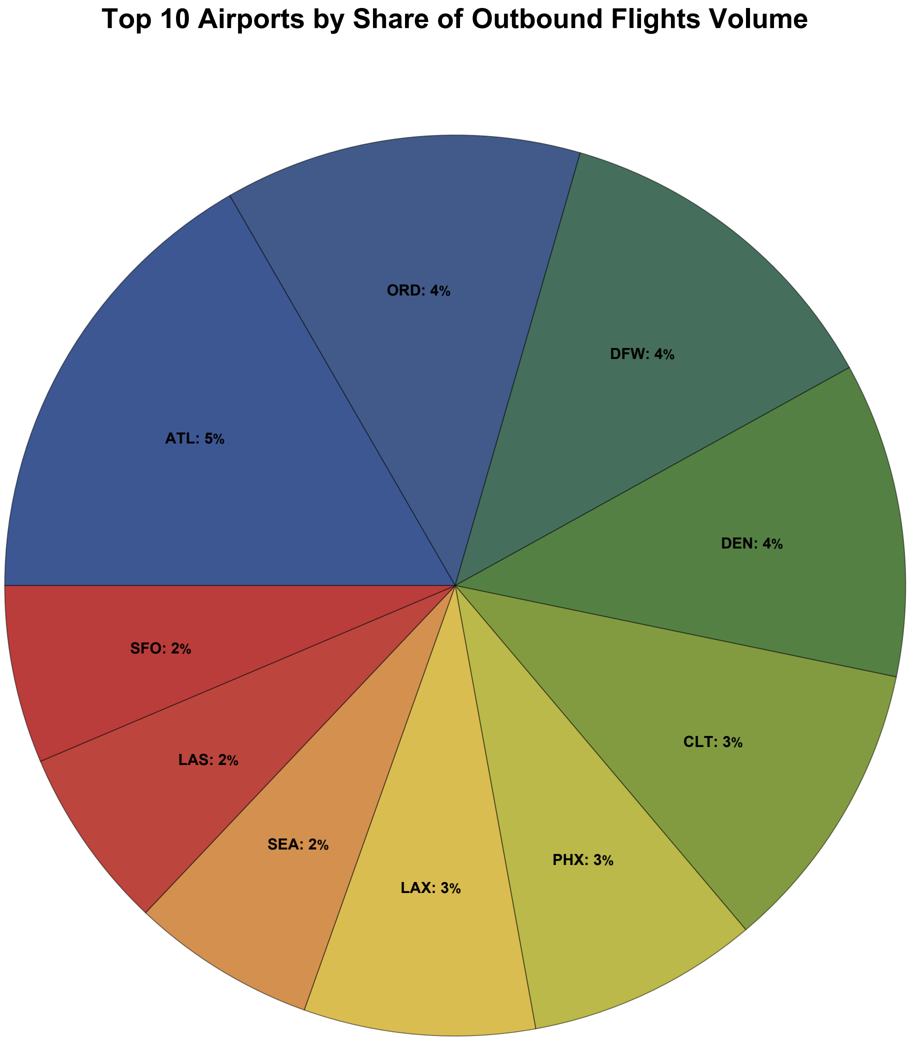
1. Most Frequent Airport:
   - ATL (Atlanta) has the highest frequency of outbound flights, exceeding 500.
2. Comparative Analysis:
   - ORD (Chicago O’Hare) and DFW (Dallas-Fort Worth) follow closely behind ATL.
   - SFO (San Francisco) has the lowest frequency among the top 10, but it still ranks significantly high compared to others in the

dataset.

3. Geographic Diversity:
   - The top airports are spread across different regions of the United States, indicating widespread air travel activity.

```
In[∘]:= (*Outbound flights*)
(*Calculate flight volume per airport*)
airportVolume = Normal[GroupBy[dataset, #ORIGIN &, Length]];

(*Calculate total number of flights*)
totalFlights = Total[Values[airportVolume]];

(*Select the top 10 airports based on flight volume*)
top10AirportVolume = TakeLargest[airportVolume, 10];

(*Calculate share of flight volume for each of the top 10 airports*)
top10AirportShares = AssociationThread[Keys[top10AirportVolume], Values[top10AirportVolume] / totalFlights];

(*Create Pie Chart for the Top 10 Airports by Share of Flight Volume*)
PieChart[Values[top10AirportShares],
 ChartLabels → Placed[MapThread[(Style[ToString[#1] <> ": " <> ToString[Round[#2 * 100, 1]] <> "%", Bold, 8]) &,
     {Keys[top10AirportShares], Values[top10AirportShares]}], "RadialCenter"],
 ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airports by Share of Outbound Flights Volume",
 ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]

(*Sum the frequencies of the top 10 airports*)
top10Flights = Total[Values[top10AirportVolume]];

(*Calculate the percentage of total flights for the top 10 airports*)
perc = Round[top10Flights / totalFlights, 0.01];

(*Print the result*)
Print["The top 10 airports provide ", perc * 100, "% of the total Outbound flights in the US."];
```

Out[∘]=



```
The top 10 airports provide 33.% of the total Outbound flights in the US.
```

Purpose: This pie chart represents the percentage share of outbound flights contributed by each of the top 10 busiest airports in the dataset. It highlights the relative importance of each airport in the context of outbound air traffic.

Insights:

1. Highest Share:
   - ATL (Atlanta) accounts for the largest share of outbound flights (5%), emphasizing its role as a major hub.
2. Other Major Contributors:
   - ORD (Chicago O’Hare), DEN (Denver), and DFW (Dallas-Fort Worth) each contribute 4%.
   - These airports are significant players in outbound air traffic.
3. Smaller Contributions:
   - Airports like SFO (San Francisco), LAS (Las Vegas), and SEA (Seattle) contribute 2% each.
4. Overall Impact:
   - Collectively, the top 10 airports handle 33.3% of all outbound flights in the dataset, underscoring their dominance in air travel.

Inbound (incoming) flights

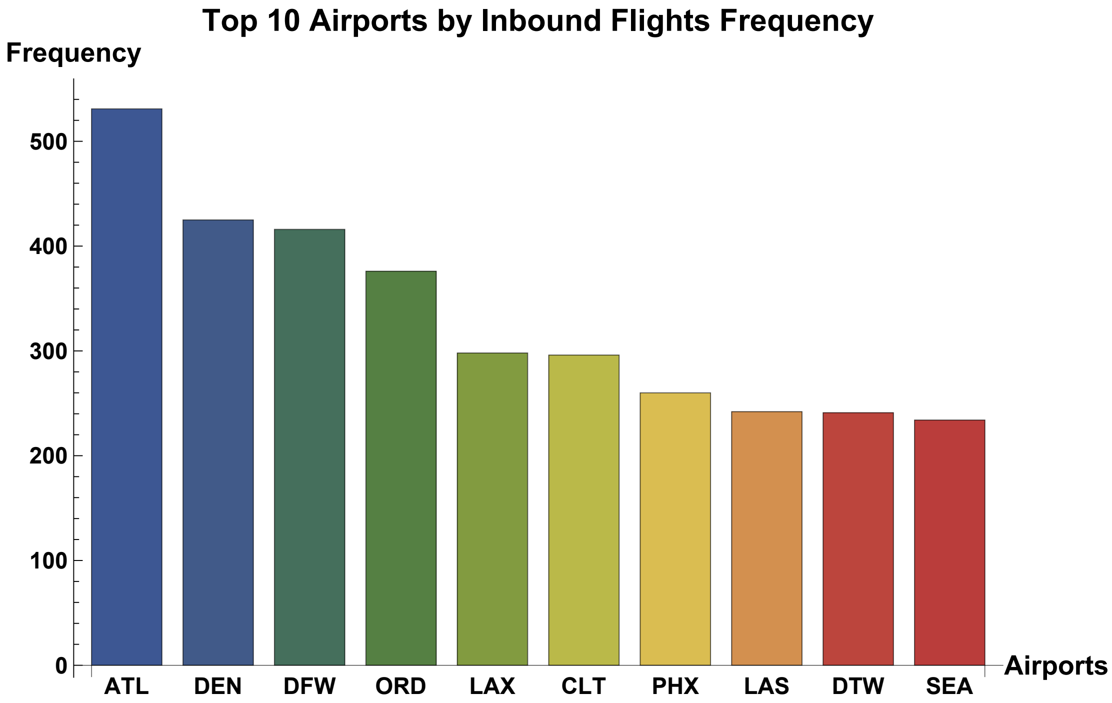
```
In[•]:= (*Inbound flights*)
        (*Extract the'DEST' column to get airports*)
        airportData1 = Normal[dataset[All, "DEST"]];

        (*Count the frequency of each airport*)
        airportCounts1 = Tally[airportData1];

        (*Sort by frequency in descending order*)
        sortedAirportCounts1 = ReverseSortBy[airportCounts1, Last];

        (*Separate airports and their frequencies*)
        airports1 = sortedAirportCounts1[[All, 1]];
        frequencies1 = sortedAirportCounts1[[All, 2]];

        (*Select top 10 airports based on frequency*)
        top10Airports1 = Take[sortedAirportCounts1, 10];
        top10AirportsNames1 = top10Airports1[[All, 1]];
        top10AirportsFrequencies1 = top10Airports1[[All, 2]];

        (*Create a table for display*)
        airportTable1 = Dataset[AssociationThread[{"Airport", "Frequency"} → #] & /@ top10Airports1];

        (*Show the table*)
        Print["Top 10 Airports by Frequency(Inbound):"];
        airportTable1

        (*Create a bar chart with vertically oriented airport names*)
        BarChart[top10AirportsFrequencies1, ChartLabels → Placed[Rotate[#, 0 Degree] & /@ top10AirportsNames1, Below],
         ChartStyle → "DarkRainbow", BarSpacing → 0.3, LabelStyle → {FontSize → 12, Bold}, ImageSize → Large,
         AxesLabel → {Style["Airports", FontSize → 14, Bold], Style["Frequency", FontSize → 14, Bold]},
         TicksStyle → Directive[FontSize → 10],
         PlotLabel → Style["Top 10 Airports by Inbound Flights Frequency", FontSize → 16, Bold]]
```

```
Top 10 Airports by Frequency(Inbound):
```

Out[•]=

Airport	Frequency
ATL	531
DEN	425
DFW	416
ORD	376
LAX	298
CLT	296
PHX	260
LAS	242
DTW	241
SEA	234

Out[∘]=



Purpose: The bar graph displays the top 10 airports with the highest number of inbound flights. It provides a ranking of the busiest airports based on inbound flight activity.

Y-Axis (Frequency):

- Represents the number of inbound flights for each airport.
- The scale starts at 0 and increases incrementally, allowing clear comparisons between airport frequencies.

X-Axis (Airports):

- Lists the airport codes (e.g., ATL, DEN, DFW) for the top 10 busiest airports in terms of inbound flights.
- Each code corresponds to a specific airport.

(Insights):

1. Most Frequent Airport:
   - ATL (Atlanta) has the highest inbound flight frequency, with a count exceeding 500.
2. Close Competitors:
   - DEN (Denver) and DFW (Dallas-Fort Worth) rank second and third, respectively, with frequencies above 400.
3. Comparative Analysis:

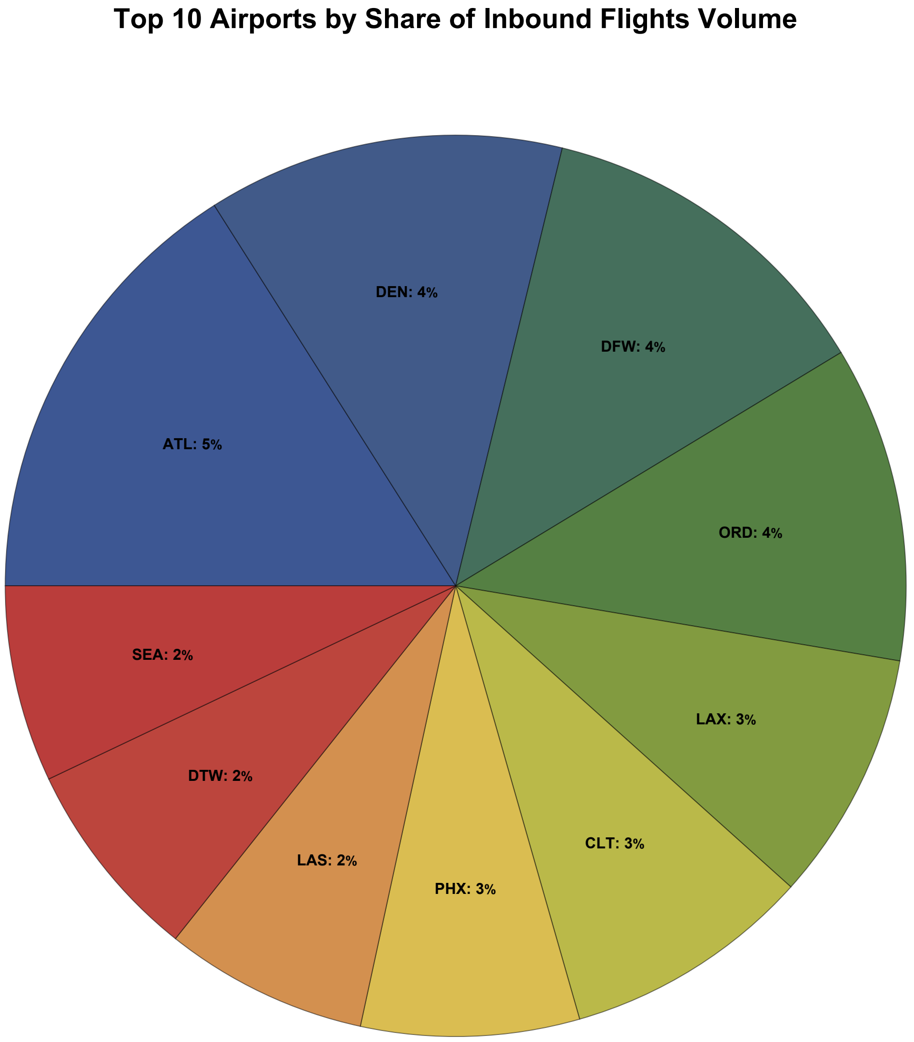
- The airports with lower inbound flight frequencies among the top 10 include DTW (Detroit) and SEA (Seattle).

4. Geographic Spread:

- The top airports are spread across different regions of the United States, suggesting a well-distributed network of inbound flights.

```
In[ ]:= (*Inbound flights*)
(*Calculate flight volume per airport*)
airportVolume1 = Normal[GroupBy[dataset, #DEST &, Length]];

(*Calculate total number of flights*)
totalFlights1 = Total[Values[airportVolume1]];

(*Select the top 10 airports based on flight volume*)
top10AirportVolume1 = TakeLargest[airportVolume1, 10];

(*Calculate share of flight volume for each of the top 10 airports*)
top10AirportShares1 = AssociationThread[Keys[top10AirportVolume1], Values[top10AirportVolume1] / totalFlights1];

(*Create Pie Chart for the Top 10 Airports by Share of Flight Volume*)
PieChart[Values[top10AirportShares1],
 ChartLabels → Placed[MapThread[(Style[ToString[#1] <> ": " <> ToString[Round[#2 * 100, 1]] <> "%", Bold, 8]) &,
    {Keys[top10AirportShares1], Values[top10AirportShares1]}], "RadialCenter"],
 ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airports by Share of Inbound Flights Volume",
 ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]

(*Sum the frequencies of the top 10 airports*)
top10Flights1 = Total[Values[top10AirportVolume1]];

(*Calculate the percentage of total flights for the top 10 airports*)
perc1 = Round[top10Flights1 / totalFlights1, 0.01];

(*Print the result*)
Print["The top 10 airports provide ", perc1 * 100, "% of the total Inbound flights in the US."];
```

Out[•]=



```
The top 10 airports provide 33.% of the total Inbound flights in the US.
```

Purpose: This pie chart visualizes the percentage share of inbound flights handled by the top 10 busiest airports. It shows the contribution of each airport to the total inbound flight volume.

Insights:

1. Dominant Airports:
   - ATL (Atlanta) has the highest share of inbound flights at 5%, reaffirming its position as a major hub.
   - DEN (Denver), DFW (Dallas-Fort Worth), and ORD (Chicago O’Hare) each handle 4% of inbound flights.
2. Smaller Shares:
   - Airports like SEA (Seattle) and DTW (Detroit) contribute 2% each, representing smaller yet significant inbound traffic.
3. Overall Contribution:
   - The top 10 airports collectively handle 33.3% of all inbound flights, reflecting their strategic importance in the US air traffic network.

b . Cancellations and Delays by Airport:

Purpose and Insights: The plots will display the frequency of cancellations and delays for top 10 airports. This helps to identify which airports experience more delays and cancellations.

Cancellations:

```
In[∘]:= (*Group data by ORIGIN and sum CANCELLED*)
airportCancels = Normal[GroupBy[dataset, #ORIGIN &, (*Group by ORIGIN airport*)
    Total[Lookup[#, "CANCELLED", 0]] & (*Sum "CANCELLED" for each group*)]];

(*Convert to a Table Format*)
airportTable = Dataset[KeyValueMap[<|"Airport" → #1, "Cancellations" → #2|> &, airportCancels]];

(*Display the Table*)
airportTable
```

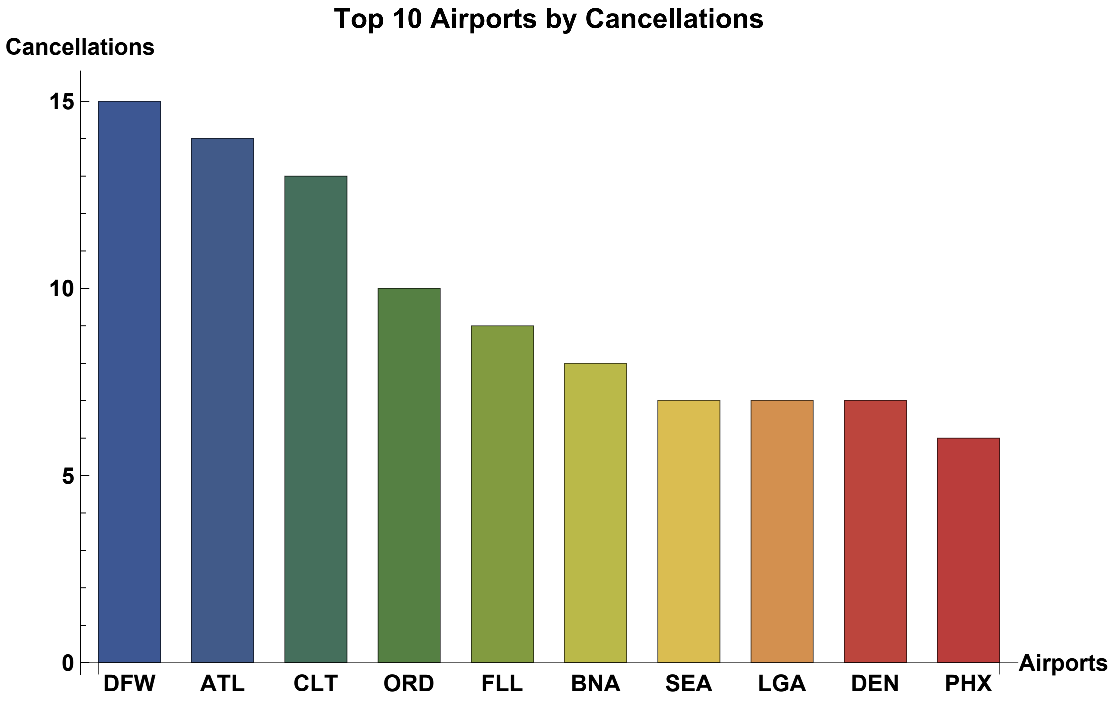
Out[∘]=

Airport	Cancellations
SAN	3
CMH	1
CVG	2
DEN	7
IAH	3
ATW	0
ABQ	1
CRW	0
TRI	0
LAS	0
SLC	1
ORD	10
DFW	15
OAK	1
FLL	9
EWR	6
TPA	2
ACY	0
TUS	0
SAT	1

rows 1–20 of **321**

```
In[•]:= (*Filter and group data by ORIGIN and sum CANCELLED*)
airportCanc = Normal[GroupBy[dataset, #ORIGIN &, Function[flights, Total[Lookup[flights, "CANCELLED", 0]]]]];

(*Ensure airportCanc is not empty*)
If[Length[airportCanc] > 0, (*Flatten the grouped data into a list of pairs {airport,cancellation count}*)
 flattenedAirportCanc = KeyValueMap[Function[{origin, count}, {origin, count}], airportCanc];
 (*Sort by cancellation count in descending order*)sortedAirportCanc = SortBy[flattenedAirportCanc, Last, Greater];
 (*Get top 10 airports with highest cancellations*)top10Airports = Take[sortedAirportCanc, 10];

 (*Create Bar Chart for top 10 airports with x-axis labels*)
 BarChart[Last /@ top10Airports, ChartLabels → Placed[Rotate[#, 0 Degree] & /@ First /@ top10Airports, Below],
  ChartStyle → "DarkRainbow", AxesLabel → {"Airports", "Cancellations"}, BarSpacing → 0.5,
  PlotLabel → "Top 10 Airports by Cancellations", LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large],
 (*If no data available*)Print["No data available for cancellations by airport."]]
```

Out[•]=



Purpose:

- The “**Top 10 Airports by Cancellations**” graph visualizes the total number of flight cancellations for the top 10 airports with the highest cancellation counts.

- It allows for identifying the airports most affected by cancellations, aiding in operational analysis and decision-making.

Y-Axis:

- Label: “Cancellations”
- Represents the total number of flight cancellations for each airport.
- The scale increases in consistent intervals, allowing for easy comparison of cancellation counts across airports.

X-Axis:

- Label: “Airports”
- Represents the top 10 airport codes (e.g., DFW, ATL, CLT) ranked by the number of cancellations.
- Each bar corresponds to a specific airport.

Insights:

1. Most Cancellations:
   - DFW (Dallas-Fort Worth) leads with the highest number of cancellations.
   - ATL (Atlanta) and CLT (Charlotte) follow closely behind.
2. Other Airports:
   - Airports like FLL (Fort Lauderdale) and BNA (Nashville) also appear in the top 10, showing moderate cancellation counts.

Delays:

Arrival Delays

```
In[∘]:= (*Group data by ORIGIN and calculate the average ARR_DELAY for each airport*)
avgArrDelays = Normal[GroupBy[dataset, #ORIGIN &, (*Group by origin airport*)
    Function[flights, N[Mean[Lookup[flights, "ARR_DELAY", 0]]]] (*Use N to get decimal values*)]];

(*Sort the average delays in descending order*)
sortedArrDelays = SortBy[avgArrDelays, -# &]; (*Sort by negative of the average ARR_DELAY for descending order*)

(*Convert to a more readable format and show in a table*)
sortedArrDelaysTable = Dataset[AssociationThread[Keys[sortedArrDelays], Values[sortedArrDelays]]]
```

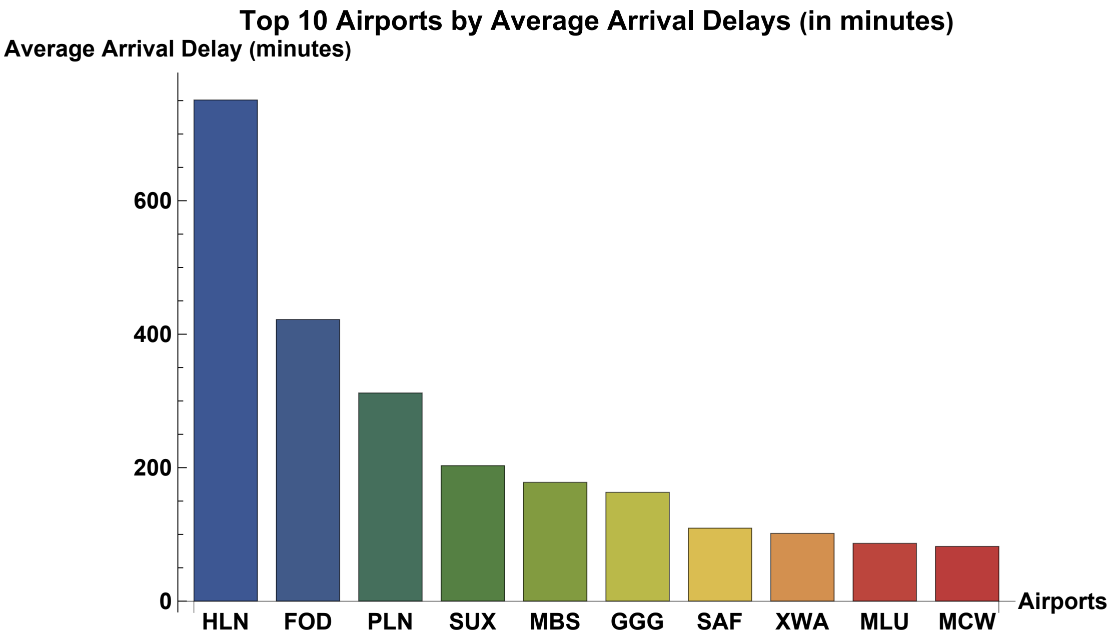
Out[•]=

HLN	751.0
FOD	422.0
PLN	312.0
SUX	203.0
MBS	178.0
GGG	163.0
SAF	109.333
XWA	101.5
MLU	86.6
MCW	82.0
MOT	71.0
ATY	64.0
CMX	64.0
PSM	62.0
SRQ	60.3684
SBN	58.1176
LIH	50.1304
GPT	47.75
PBG	42.0
CDV	41.5

rows 1–20 of **321**

```
In[∘]:= (*Plot average arrival delays as a bar chart for top 10 airports*)
        BarChart[Values[Take[sortedArrDelays, 10]], ChartLabels → Placed[Keys[Take[sortedArrDelays, 10]], "Inside"],
         BarSpacing → 0.3, ChartStyle → "DarkRainbow", AxesLabel → {"Airports", "Average Arrival Delay (minutes)"},
         PlotLabel → "Top 10 Airports by Average Arrival Delays (in minutes)",
         LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]
```

Out[∘]=



Purpose:

The “**Top 10 Airports by Average Arrival Delays (in minutes)**” graph visualizes the average arrival delay (in minutes) for the top 10 airports with the highest delays. It helps identify airports that experience significant delays, providing insights for operational improvements and traveler expectations.

Y-Axis:

- Label: “Average Arrival Delay (in minutes)”
- Represents the average delay time for arrivals at each airport.
- The scale increases in minutes, showing the magnitude of delays for each airport.

X-Axis:

- Label: “Airports”

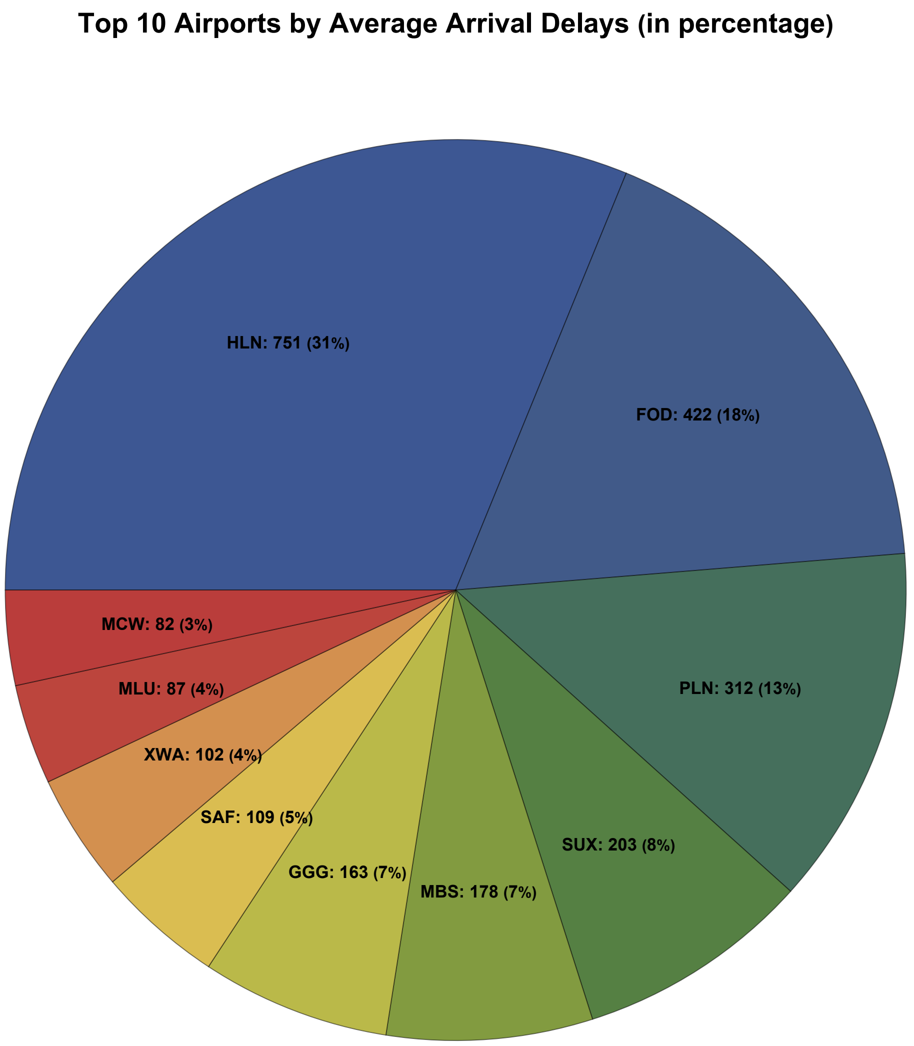
- Represents the airport codes (e.g., HLN, FOD, PLN) for the top 10 airports with the highest average arrival delays.
- Each bar corresponds to one airport.

Insights:

1. Longest Delays:
   - HLN (Helena) has the highest average arrival delay, exceeding 600 minutes, which is a significant outlier.
2. Moderately High Delays:
   - FOD (Fort Dodge) and PLN (Pellston) have delays averaging around 400 and 300 minutes, respectively.
3. Lower Delays in Top 10:
   - Airports like MLU (Monroe) and MCW (Mason City) have delays averaging around 100 minutes, much lower than HLN but still notable.
4. Operational Focus:
   - HLN requires immediate attention to address excessive delays, while airports with moderate delays could benefit from efficiency improvements.

In[◦]:=
```
Module[{data, total, percentages, labels}, data = Take[Values[sortedArrDelays], 10];
 total = Total[data];
 percentages = (100 * # / total) & /@ data;
 labels = MapThread[Style[ToString[#1] <> ": " <> ToString[Round[#2, 1]] <> " (" <> ToString[Round[#3, 1]] <> "%)",
     Bold, 9] &, {Keys[Take[sortedArrDelays, 10]], data, percentages}];
 PieChart[percentages, ChartLabels → Placed[labels, "RadialCenter"],
  ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airports by Average Arrival Delays (in percentage)",
  ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]]
```

Out[•]=



Purpose:

The “**Top 10 Airports by Average Arrival Delays (in percentage)**” pie chart visualizes the average proportion of arrival delay (in minutes) for the top 10 airports with the highest delays. It helps identify airports that experience significant delays, providing insights for operational improvements and traveler expectations.

Segments:

- Each segment represents an airport code (e.g., HLN, FOD, PLN) for the top 10 airports with the highest average proportion for arrival delays.
- The size of each segment corresponds to the magnitude and percentage of the average delay time for arrivals at each airport.

Insights:

1. Longest Delays:
   - HLN (Helena) has the highest average delay, accounting for 31% of the total, making it a significant outlier with 751 minutes.
2. Moderately High Delays:
   - FOD (Fort Dodge) and PLN (Pellston) contribute 18% and 13%, with delays averaging 422 and 312 minutes, respectively.
3. Lower Delays in Top 10:
   - Airports like MLU (Monroe) and MCW (Mason City) have much smaller shares, contributing 3-4%, with average delays around 87 and 82 minutes.
4. Operational Focus:
   - HLN requires immediate attention to address excessive delays, while airports like FOD and PLN would benefit from targeted improvements in operational efficiency.

◆ Departure Delays

```
In[ ]:= (*Group data by ORIGIN and calculate the average DEP_DELAY for each airport*)
        (*Group data by ORIGIN and calculate the average ARR_DELAY for each airport*)
        avgDepDelays = Normal[GroupBy[dataset, #ORIGIN &, (*Group by origin airport*)
            Function[flights, N[Mean[Lookup[flights, "DEP_DELAY", 0]]]] (*Use N to get decimal values*)]];

        (*Sort the average delays in descending order*)
        sortedDepDelays = SortBy[avgDepDelays, -# &]; (*Sort by negative of the average ARR_DELAY for descending order*)

        (*Convert to a more readable format and show in a table*)
        sortedDepDelaysTable = Dataset[AssociationThread[Keys[sortedDepDelays], Values[sortedDepDelays]]]
```

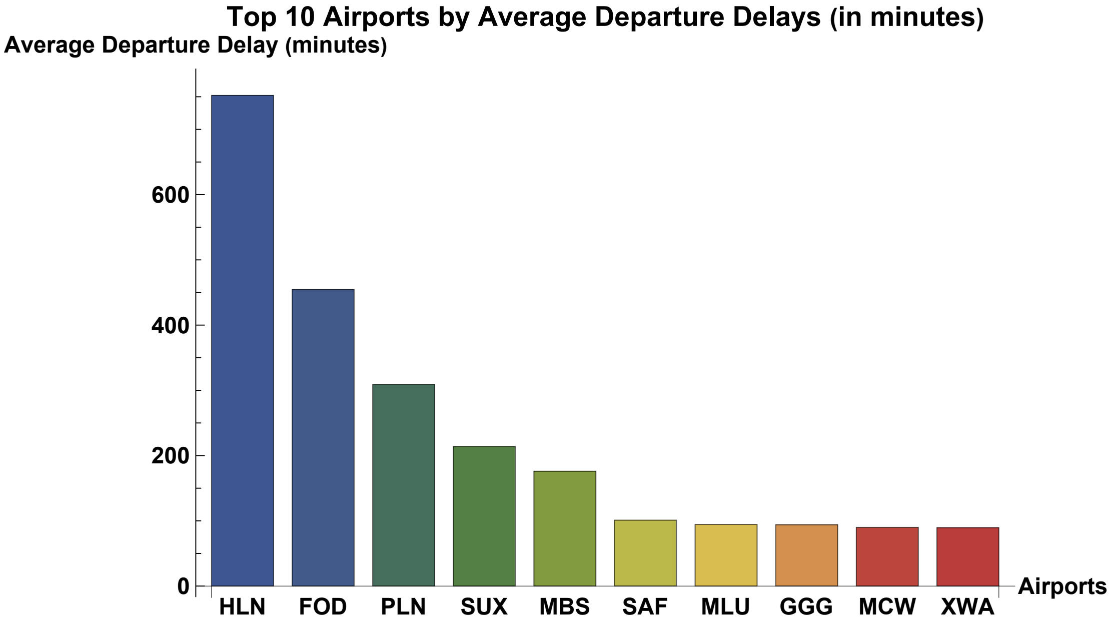
Out[•]=

HLN	752.0
FOD	454.5
PLN	309.0
SUX	214.0
MBS	176.0
SAF	101.0
MLU	94.4
GGG	94.0
MCW	90.0
XWA	89.5
MOT	67.0
SBN	65.1176
SRQ	63.2632
CMI	62.0
OTH	62.0
CMX	59.0
ATY	55.0
GPT	54.5
PSM	53.0
LIH	49.9565

rows 1–20 of **321**

In[∘]:=
```
(*Plot average departure delays as a bar chart for top 10 airports*)
BarChart[Values[Take[sortedDepDelays, 10]], ChartLabels → Placed[Keys[Take[sortedDepDelays, 10]], "Inside"],
 BarSpacing → 0.3, ChartStyle → "DarkRainbow", AxesLabel → {"Airports", "Average Departure Delay (minutes)"},
 PlotLabel → "Top 10 Airports by Average Departure Delays (in minutes)",
 LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]
```

Out[∘]=



Purpose:

The bar chart “**Top 10 Airports by Average Departure Delays (in minutes)**” illustrates the average departure delay (in minutes) for the top 10 airports with the most significant delays. This helps identify airports where flights are commonly delayed before departure, aiding stakeholders in targeting improvement strategies.

Y-Axis:

- Label: “Average Departure Delay (in minutes)”
- Represents the average delay in departure times for flights departing from each airport.
- The scale increases in consistent intervals of minutes, providing a clear comparison.

X-Axis:

- Label: “Airports”

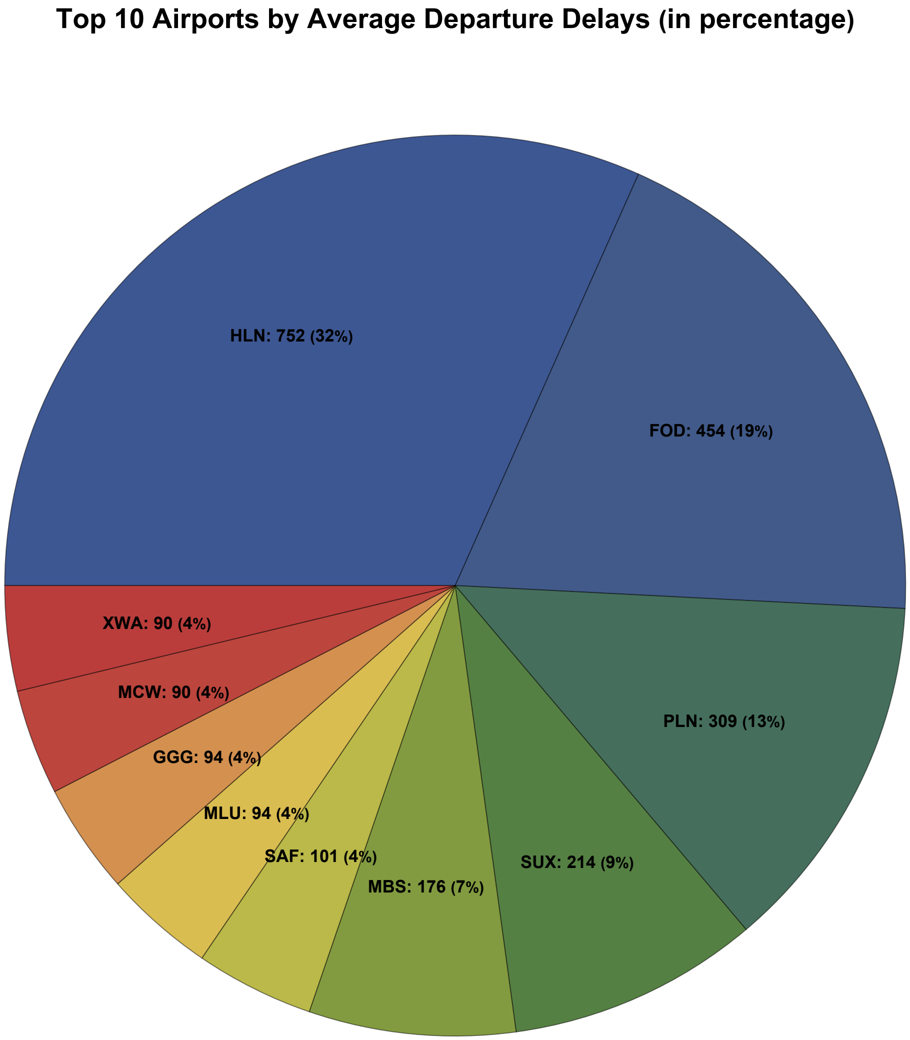
- Represents the airport codes (e.g., HLN, FOD, PLN) for the top 10 airports with the highest average departure delays.
- Each bar corresponds to one airport.

Insights:

1. Longest Departure Delays:
   - HLN (Helena) has the highest average departure delay, exceeding 600 minutes, making it an extreme outlier.
2. Significant Delays:
   - FOD (Fort Dodge) and PLN (Pellston) experience average delays of 400 minutes and 300 minutes, respectively.
3. Lower Delays in the Top 10:
   - Airports like XWA (Williston) and MCW (Mason City) have delays averaging around 100 minutes, significantly lower than HLN but still notable.
4. Operational Observations:
   - HLN may require urgent attention to reduce its extreme delays, while other airports in the top 10 could benefit from moderate operational improvements.

```
In[∘]:= (*Create a Pie Chart for top 10 airports based on average departure delays*)

Module[{data, total, percentages, labels}, data = Take[Values[sortedDepDelays], 10];
 total = Total[data];
 percentages = (100 * # / total) & /@ data;
 labels = MapThread[Style[ToString[#1] <> ": " <> ToString[Round[#2, 1]] <> " (" <> ToString[Round[#3, 1]] <> "%)",
     Bold, 9] &, {Keys[Take[sortedDepDelays, 10]], data, percentages}];
 PieChart[percentages, ChartLabels → Placed[labels, "RadialCenter"],
  ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airports by Average Departure Delays (in percentage)",
  ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]]
```

Out[•]=



Purpose:

The “**Top 10 Airports by Average Departure Delays (in percentage)**” pie chart visualizes the average proportion of departure delay (in minutes) for the top 10 airports with the highest delays. It helps identify airports that experience significant delays, providing insights for operational improvements and traveler expectations.

Segments:

- Each segment represents an airport code (e.g., HLN, FOD, PLN) for the top 10 airports with the highest average departure delays.
- The size of each segment corresponds to the magnitude and percentage of the average delay time for departures at each airport.

Insights:

1. Longest Delays:
   - HLN (Helena) has the highest average departure delay, with 752 minutes, which is a significant outlier.
2. Moderately High Delays:
   - FOD (Fort Dodge) and PLN (Pellston) have delays averaging around 454 and 309 minutes, respectively.
3. Lower Delays in Top 10:
   - Airports like SAF (Santa Fe) and MCW (Mason City) have delays averaging around 101 and 90 minutes, much lower than HLN but still notable.
4. Operational Focus:
   - HLN requires immediate attention to address excessive delays, while airports with moderate delays could benefit from efficiency improvements.

◆ Total Delays

```
In[•]:= (*Average Delays = Mean of Arrival and Departure Delays*)
        (*Group by airport and calculate the average delay for each airport*)
        airportDelays = GroupBy[dataset, #ORIGIN &, Function[flights, Mean[Flatten[Lookup[flights, {"DELAY_DUE_CARRIER",
                "DELAY_DUE_WEATHER", "DELAY_DUE_NAS", "DELAY_DUE_SECURITY", "DELAY_DUE_LATE_AIRCRAFT"}, 0]]]]]];

        (*Convert grouped data into a single flat list with "Airport" and "Average Delays" keys*)
        flattenedAirportDelays =
          KeyValueMap[Function[{origin, delays}, <|"Airport" → origin, "Average Delays" → N[delays]|>], airportDelays];
        (*Apply N to get decimal format*)

        (*Sort the average delays in descending order*)
        sortedDelays = SortBy[flattenedAirportDelays, -#["Average Delays"] &];

        (*Convert to a more readable format and show in a table*)
        sortedDepTable = Dataset[sortedDelays];

        (*Display the table*)
        sortedDepTable
```

Out[∘]=

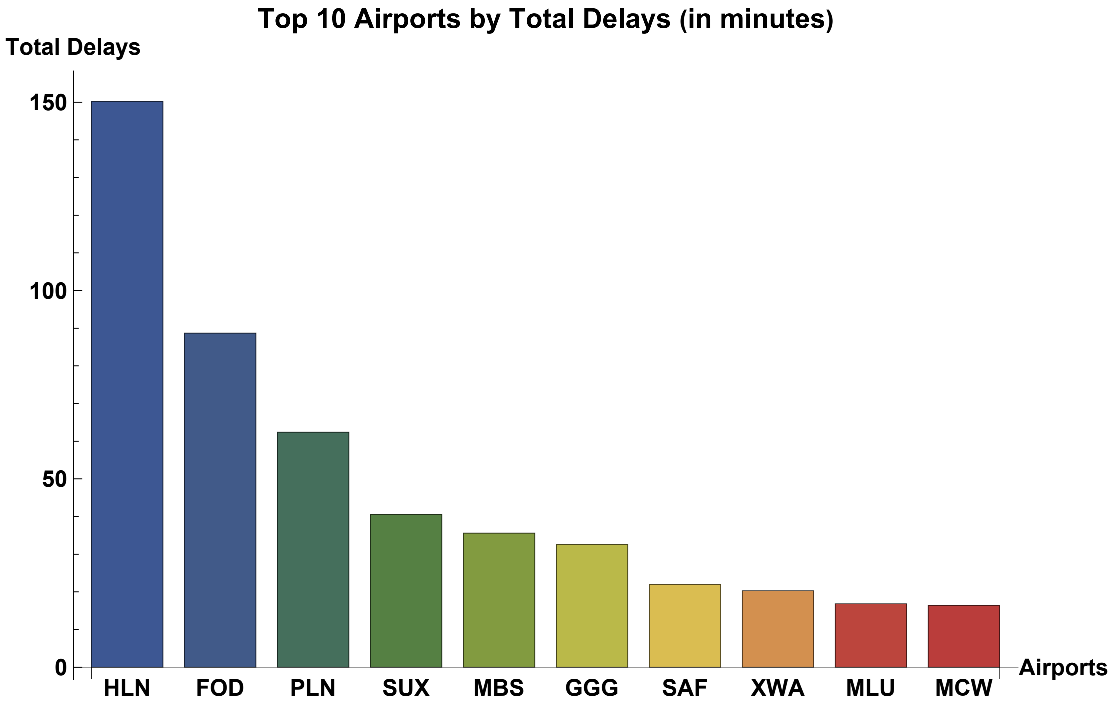
Airport	Average Delays
HLN	150.2
FOD	88.7
PLN	62.4
SUX	40.6
MBS	35.6
GGG	32.6
SAF	21.9333
XWA	20.3
MLU	16.84
MCW	16.4
SBN	13.6824
MOT	13.6
SRQ	13.3263
ATY	12.9
CMX	12.8
PSM	12.4
CMI	12.0
LIH	11.313
GPT	10.6
PBG	10.4

rows 1–20 of **321**

In[∘]:=

```
(*Extract the airport codes and delays from sortedDelays dataset*)
airportCodes3 = sortedDelays[All, "Airport"];
delays = sortedDelays[All, "Average Delays"];

(*Convert to lists for BarChart*)
airportCodesList = Normal[airportCodes3];
delaysList = Normal[delays];

(*Plot the average total delays as a bar chart for top 10 airports*)
BarChart[delaysList⟦;; 10⟧, ChartLabels → Placed[airportCodesList⟦;; 10⟧, Below],
 BarSpacing → 0.3, ChartStyle → "DarkRainbow", (*Use a predefined color scheme*)
 AxesLabel → {"Airports", "Total Delays"}, PlotLabel → "Top 10 Airports by Total Delays (in minutes)",
 LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]
```

Out[∘]=



Purpose:

The “**Top 10 Airports by Total Delays (in minutes)**” bar chart visualizes the total delay times for the top 10 airports with the highest delays. This chart helps identify airports experiencing significant delays, providing insights for operational improvements and traveler expectations.

Y-Axis:

- Label: “Total Delays (in minutes)”
- Representation: The total delay time for each airport.
- Scale: Increases in minutes, showing the magnitude of delays for each airport.

X-Axis:

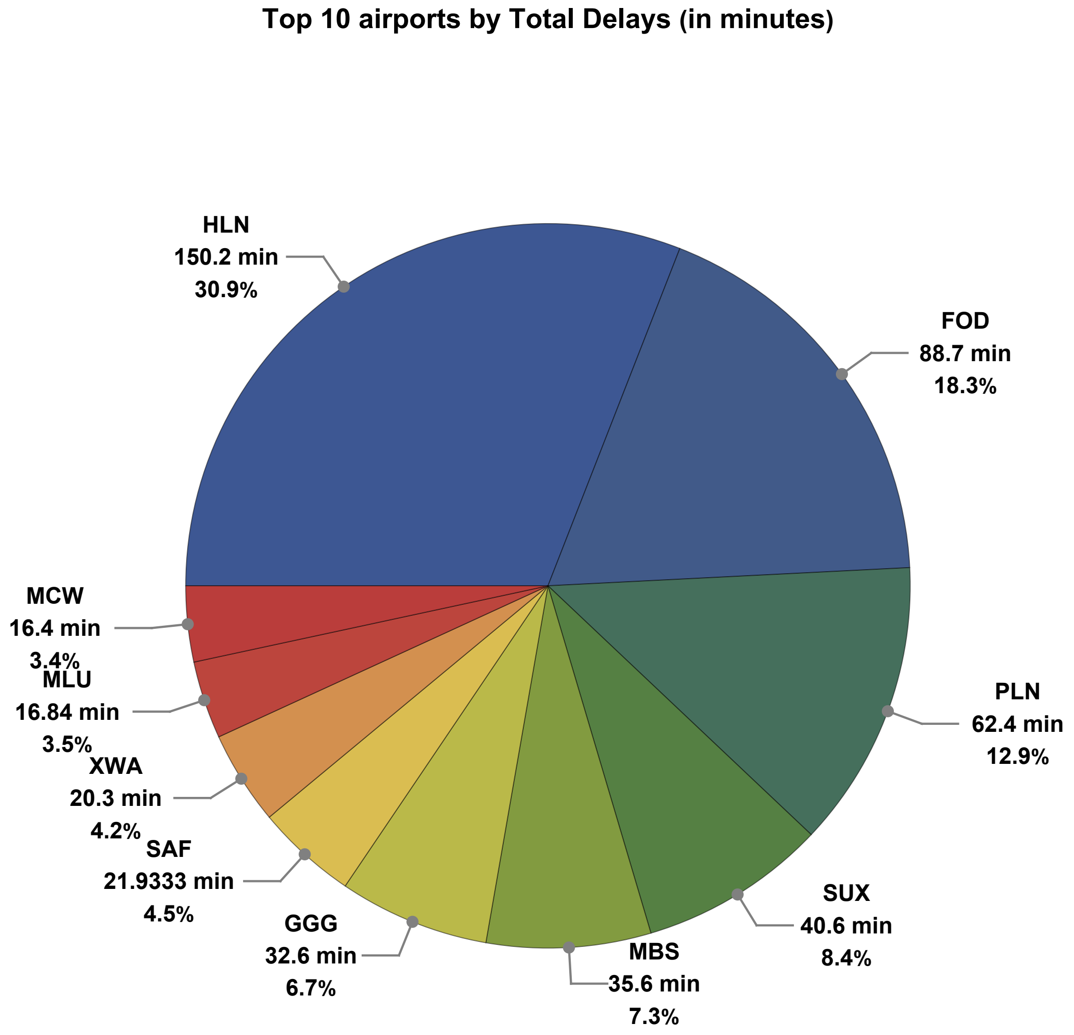
- Label: “Airports”
- Representation: The airport codes (e.g., HLN, FOD, PLN) for the top 10 airports with the highest total delays.
- Bars: Each bar corresponds to one airport.

Insights:

1. Longest Delays:
   - HLN (Helena) has the highest total delays, exceeding 150 minutes, which is a significant outlier.
2. Moderately High Delays:
   - FOD (Fort Dodge) and PLN (Pellston) have delays averaging around 100 and 60 minutes, respectively.
3. Lower Delays in Top 10:
   - Airports like MLU (Monroe) and MCW (Mason City) have delays averaging around 30 minutes, much lower than HLN but still notable.
4. Operational Focus:
   - HLN requires immediate attention to address excessive delays, while airports with moderate delays could benefit from efficiency improvements.

```
In[ ]:= PieChart[delaysList⟦ ;; 10⟧,
         ChartLabels → Placed[MapThread[Function[{code, delay}, StringJoin[code, "\n", ToString[delay],
               " min\n", ToString[NumberForm[100 delay / Total[delaysList⟦ ;; 10⟧], {3, 1}]], "%"]],
             {airportCodesList⟦ ;; 10⟧, delaysList⟦ ;; 10⟧}], "RadialCallout"], ChartStyle → "DarkRainbow",
         PlotLabel → "Top 10 airports by Total Delays (in minutes)", LabelStyle → Directive[FontSize → 12, Bold],
         ImageSize → Large]
```

Out[•]=



Purpose:

The “**Top 10 Airports by Total Delays (in minutes)**” pie chart visualizes the percentage share of total delay times for the top 10 airports with the highest delays. It helps identify airports that experience significant delays, providing insights for operational improvements and traveler expectations.

Segments:

- Each segment represents an airport code (e.g., HLN, FOD, PLN) for the top 10 airports with the highest total delays.
- The size of each segment corresponds to the magnitude and percentage of the total delay time for arrivals and departures at each airport.

Insights:

1. Longest Delays:
    - HLN (Helena) has the highest total delays, with 150.2 minutes.
2. Moderately High Delays:
    - FOD (Fort Dodge) and PLN (Pellston) have delays totaling around 88.7 and 62.4 minutes, respectively.
3. Lower Delays in Top 10:
    - Airports like MLU (Monroe) and MCW (Mason City) have delays totaling around 16.84 and 16.4 minutes, much lower than HLN but still notable.
4. Operational Focus:
    - HLN requires immediate attention to address excessive delays, while airports with moderate delays could benefit from efficiency improvements.

c . Trends in Delays and Cancellations Over Time:

Purpose and Insights: Line plot shows the average delay or cancellation rates across months or years. This helps to spot any time-based patterns in delays or cancellations and analyze how delays or cancellations evolve over time, aiding in identifying potential seasonal trends or areas for operational improvements.

1. By Month and Year :

```
In[•]:= (*Extract the year,month,ARR_DELAY,DEP_DELAY,and CANCELLED for each flight*)monthlyData =
          Map[Function[row, <|"Year" → StringTake[row["FL_DATE"], {1, 4}], "Month" → StringTake[row["FL_DATE"], {6, 7}],
              "ARR_DELAY" → row["ARR_DELAY"], "DEP_DELAY" → row["DEP_DELAY"], "CANCELLED" → row["CANCELLED"]|>], dataset];

        (*Convert monthly data into a proper table format*)
        monthlyDataTable = Dataset[monthlyData];

        (*Group by Year and Month and calculate averages*)
```

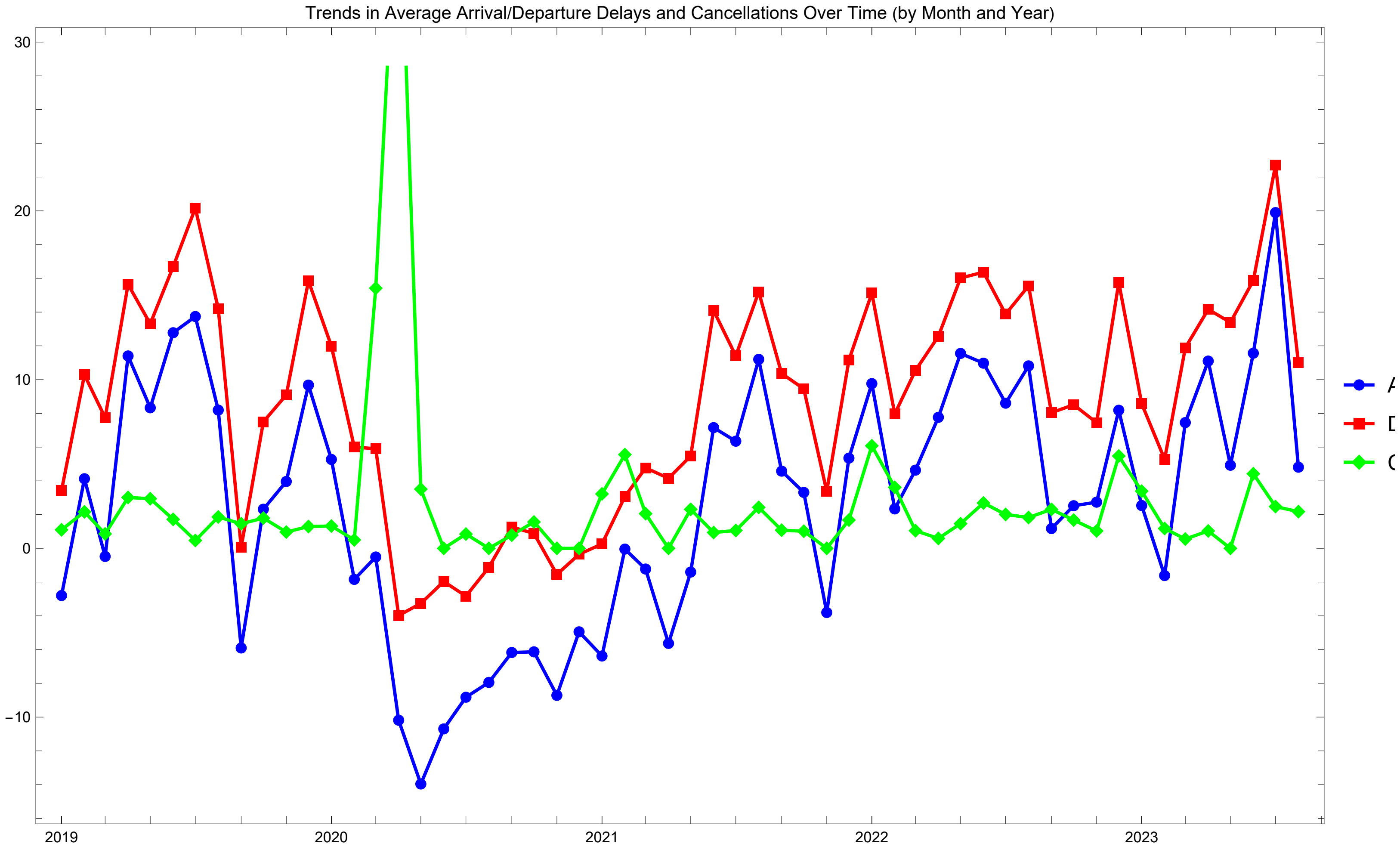
```
groupedData = GroupBy[monthlyData, {#["Year"], #["Month"]} &,
   Function[flights, <|"AvgARRDelay" → N[Mean[Lookup[flights, "ARR_DELAY", 0]]], "AvgDEPDelay" →
      N[Mean[Lookup[flights, "DEP_DELAY", 0]]], "AvgCancelled" → N[Mean[Lookup[flights, "CANCELLED", 0]]]|>]];

(*Convert the grouped data into a more readable format*)
groupedDataTable = Dataset[KeyValueMap[<|"Year" → #1⟦1⟧, "Month" → #1⟦2⟧, "AvgARRDelay" → #2["AvgARRDelay"],
      "AvgDEPDelay" → #2["AvgDEPDelay"], "AvgCancelled" → #2["AvgCancelled"]|> &, groupedData]];
groupedDataTable
(*Prepare data for plotting*)
arrivalDelaysData = {#["Year"], #["Month"], #["AvgARRDelay"]} & /@ groupedDataTable;
departureDelaysData = {#["Year"], #["Month"], #["AvgDEPDelay"]} & /@ groupedDataTable;
cancelledData = {#["Year"], #["Month"], #["AvgCancelled"]} & /@ groupedDataTable;

(*Sort the data by Year and Month*)
sortedArrivalDelays = SortBy[arrivalDelaysData, {First, #⟦2⟧ &}]; (*Sorting by Year and Month*)
sortedDepartureDelays = SortBy[departureDelaysData, {First, #⟦2⟧ &}];
sortedCancelledData = SortBy[cancelledData, {First, #⟦2⟧ &}];

(*Convert Year-Month to DateObject*)
convertToDateObject[year_, month_] := DateObject[{ToExpression[year], ToExpression[month], 1}];

(*Prepare the data for plotting*)
arrivalDelaysDataWithDate = {convertToDateObject[#⟦1⟧, #⟦2⟧], #⟦3⟧} & /@ sortedArrivalDelays;
departureDelaysDataWithDate = {convertToDateObject[#⟦1⟧, #⟦2⟧], #⟦3⟧} & /@ sortedDepartureDelays;
cancelledDataWithDate = {convertToDateObject[#⟦1⟧, #⟦2⟧], #⟦3⟧ * 100} & /@ sortedCancelledData;
(*Scale cancellation values by 100*)

(*Create the DateListPlot with proper scaling*)
DateListPlot[{arrivalDelaysDataWithDate, departureDelaysDataWithDate, cancelledDataWithDate},
 PlotStyle → {Blue, Red, Green}, (*Blue for arrival delays,Red for departure delays,Green for cancellations*)
 PlotMarkers → Automatic, (*Show markers on the plot*)AxesLabel → {"Year-Month", "Value (scaled)"},
 PlotLabel → "Trends in Average Arrival/Departure Delays and Cancellations Over Time (by Month and Year)",
 PlotLegends → {"Arrival Delay", "Departure Delay", "Cancellations (x100)"}, ImageSize → 800]
```

Out[∘]=

Year	Month	AvgARRDelay	AvgDEPDelay	AvgCancelled
2022	07	8.60302	13.8794	0.0201005
2022	09	1.17341	8.04046	0.0231214
2021	03	−1.22603	4.76712	0.0205479
2020	01	5.26872	11.9868	0.0132159
2023	06	11.5635	15.8729	0.0441989
2023	05	4.92308	13.3897	0.0
2019	09	−5.90291	0.0631068	0.0145631
2022	04	7.76471	12.5647	0.00588235
2019	11	3.96135	9.09179	0.00966184
2022	01	9.76243	15.1271	0.0607735
2021	11	−3.80328	3.37705	0.0
2020	10	−6.13281	0.882812	0.015625
2021	08	11.199	15.199	0.0242718
2019	08	8.1907	14.2	0.0186047
2019	12	9.67241	15.8448	0.012931
2023	03	7.45304	11.884	0.00552486
2022	06	10.9731	16.3656	0.0268817
2023	04	11.0979	14.1598	0.0103093
2019	02	4.12432	10.2811	0.0216216
2023	08	4.80978	11.0	0.0217391

rows 1–20 of **56**

Out[∘]=



Purpose:

The goal is to analyze seasonal variations, spikes, and overall trends over the time period from 2019 to 2023.

Axes:

- X-Axis: Represents time, labeled by year and month, spanning 2019 to 2023.
- Y-Axis: Represents average delays (in minutes) for arrival and departure, and total cancellations (scaled down by 100 for visibility).

Insights:

1. Arrival and Departure Delay Trends:

• Both delays exhibit seasonal spikes with noticeable peaks, especially in late 2022 and early 2023.

• Departure delays (red) are generally higher than arrival delays (blue), indicating more disruptions before takeoff than during flight.

2. Cancellations:

• The green line (scaled cancellations) shows sharp spikes in 2020, likely corresponding to the onset of COVID-19 when air travel saw unprecedented cancellations.

• Post-2021, cancellations stabilize at relatively low levels but still align with delay trends.

3. Interdependence:

• Peaks in cancellations often correlate with spikes in both arrival and departure delays, suggesting that systemic disruptions (e.g., weather, operational issues) contribute to all three metrics.

4. Operational Challenges in Late 2022 and 2023:

• The significant increase in delays and cancellations during this period may indicate heightened travel demand, weather disruptions, or operational inefficiencies.

Key Takeaway:

The chart highlights systemic issues in flight operations, with correlations between delays and cancellations. The data provides valuable insights for airlines and airports to address peak-time disruptions and improve reliability.

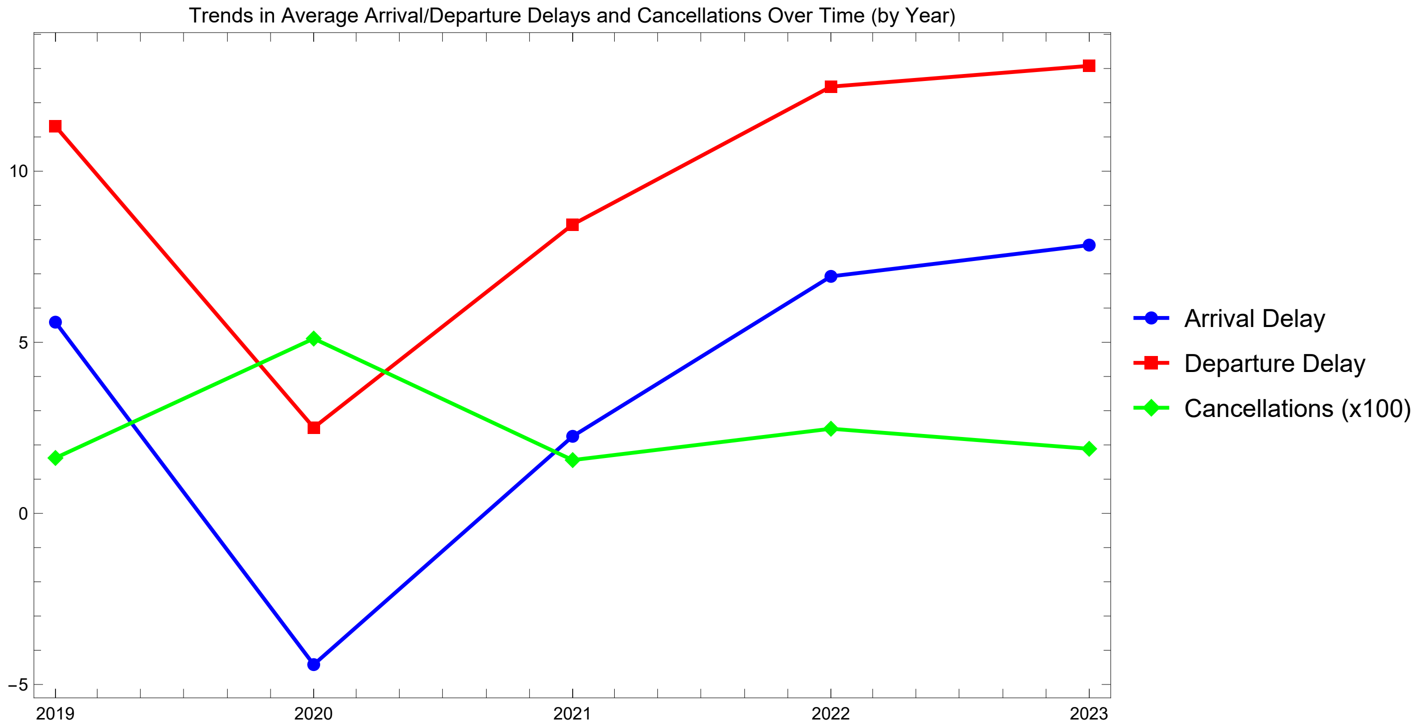
2. By Year:

```
In[•]:= (*Extract the Year,ARR_DELAY,DEP_DELAY,and CANCELLED for each flight*)
        yearlyData = Map[Function[row, <|"Year" → StringTake[row["FL_DATE"], {1, 4}],
             "ARR_DELAY" → row["ARR_DELAY"], "DEP_DELAY" → row["DEP_DELAY"], "CANCELLED" → row["CANCELLED"] |>], dataset];

        (*Convert yearly data into a proper table format*)
        yearlyDataTable = Dataset[yearlyData];

        (*Group by Year and calculate the averages,ensuring decimal values*)groupedData1 = GroupBy[yearlyData,
           #["Year"] &, Function[flights, <|"AvgARRDelay" → N[Mean[Lookup[flights, "ARR_DELAY", 0]]], "AvgDEPDelay" →
              N[Mean[Lookup[flights, "DEP_DELAY", 0]]], "AvgCancelled" → N[Mean[Lookup[flights, "CANCELLED", 0]]] |>]];

        (*Convert the grouped data into a more readable format with decimals*)
        groupedDataTable1 =
          Dataset[KeyValueMap[<|"Year" → #1, "AvgARRDelay" → #2["AvgARRDelay"], "AvgDEPDelay" → #2["AvgDEPDelay"],
              "AvgCancelled" → #2["AvgCancelled"] |> &, groupedData1]];
        groupedDataTable1

        (*Prepare data for plotting*)
        arrivalDelaysData1 = {#["Year"], #["AvgARRDelay"]} & /@ groupedDataTable1;
        departureDelaysData1 = {#["Year"], #["AvgDEPDelay"]} & /@ groupedDataTable1;
        cancelledData1 = {#["Year"], #["AvgCancelled"] * 100} & /@ groupedDataTable1; (*Scale cancellation values by 100*)

        (*Sort the data by Year*)
        sortedArrivalDelays1 = SortBy[arrivalDelaysData1, First];
        sortedDepartureDelays1 = SortBy[departureDelaysData1, First];
        sortedCancelledData1 = SortBy[cancelledData1, First];

        (*Create the DateListPlot*)
        DateListPlot[{sortedArrivalDelays1, sortedDepartureDelays1, sortedCancelledData1}, PlotStyle → {Blue, Red, Green},
         PlotMarkers → Automatic, (*Show markers on the plot*)AxesLabel → {"Year", "Value (Scaled)"},
         PlotLabel → "Trends in Average Arrival/Departure Delays and Cancellations Over Time (by Year)",
         PlotLegends → {"Arrival Delay", "Departure Delay", "Cancellations (x100)"}, ImageSize → Large]
```

Out[•]=

Year	AvgARRDelay	AvgDEPDelay	AvgCancelled
2022	6.9258	12.4704	0.024735
2021	2.25158	8.44088	0.0155718
2020	−4.42007	2.50481	0.0510817
2023	7.83962	13.0775	0.0188679
2019	5.58745	11.3131	0.0161863

Out[•]=



Purpose:

The purpose is to observe long-term trends, understand annual variability, and identify years with significant disruptions.

Axes:

- X-Axis: Represents years from 2019 to 2023.
- Y-Axis:

♣ Blue and Red Lines: Average delays in minutes for arrivals and departures.

♣ Green Line: Average cancellations (scaled by 100 for visualization).

Insights:

1. Arrival and Departure Delays:

• 2023 and 2022 had the highest average delays:

♣ Departure delays in 2023 peaked at over 13 minutes on average, while arrival delays also increased to over 7.8 minutes.

♣ 2020 saw the lowest average arrival delay, dropping to -4.42 minutes, possibly due to reduced air traffic during the COVID-19 pandemic.

• Departure delays are consistently higher than arrival delays across all years, suggesting pre-flight challenges contribute heavily to overall delays.

2. Cancellations:

• 2020 had the highest average cancellations (scaled to 5.1%) due to COVID-19 travel disruptions.

• Cancellation rates dropped significantly in subsequent years, with 2022 and 2023 averaging around 1.8% and 2.4%, respectively.

3. Impact of COVID-19:

• The drastic decrease in delays and increased cancellations in 2020 aligns with the onset of the pandemic, where travel restrictions likely caused fewer but more systematic disruptions.

4. Post-COVID Recovery:

• The upward trend in delays from 2021 to 2023 reflects a return to pre-pandemic air traffic levels, coupled with operational inefficiencies during the recovery phase.

Key Takeaways:

• The data highlights critical years (e.g., 2020 for cancellations, 2023 for delays) for analyzing the causes of disruptions.

• Airlines and airports could focus on reducing departure delays, which are consistently higher than arrival delays.

• Further investigation into the systemic issues post-pandemic (2022-2023) can help address growing delays.

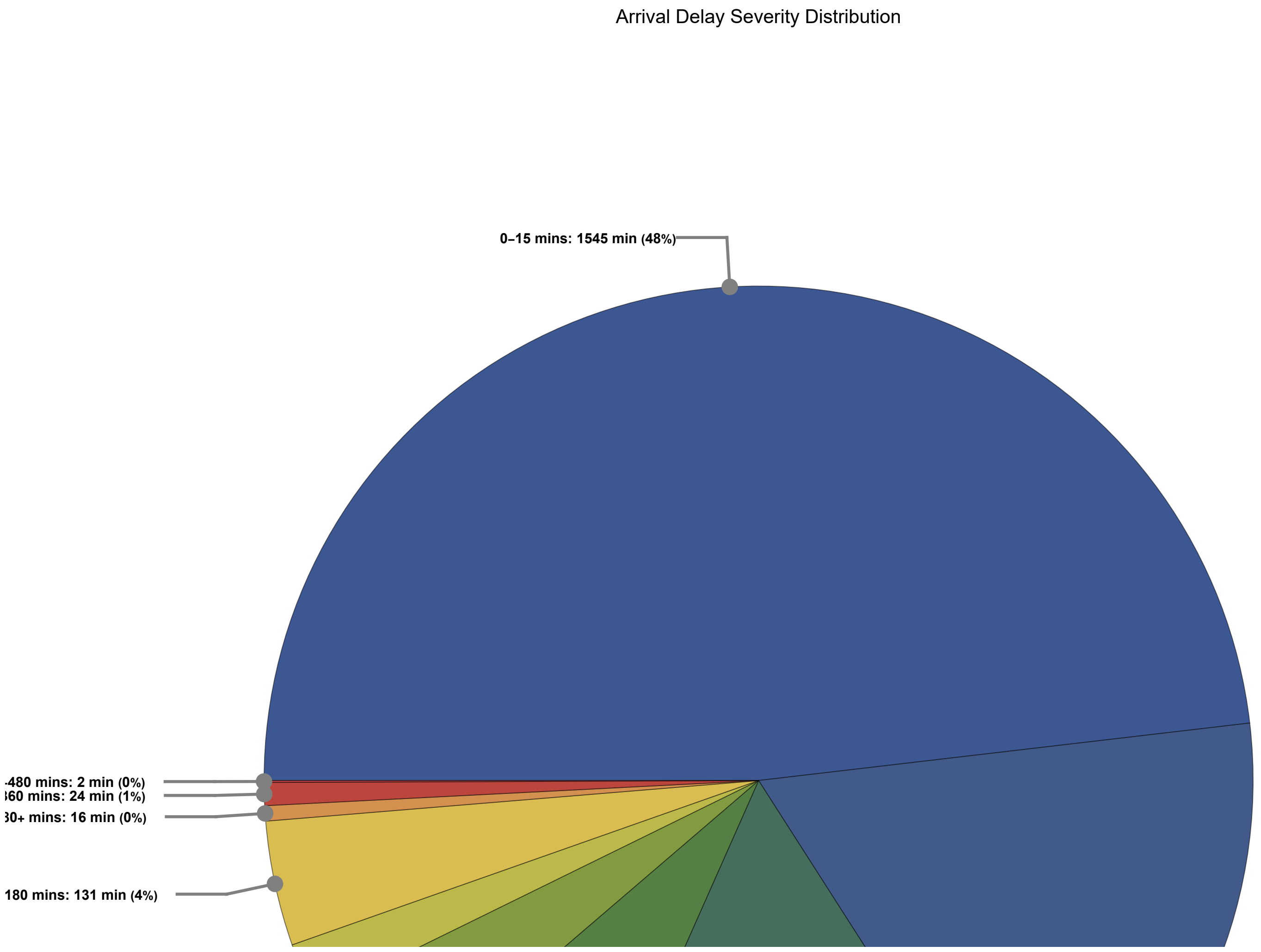
d. Delay Severity Distribution:

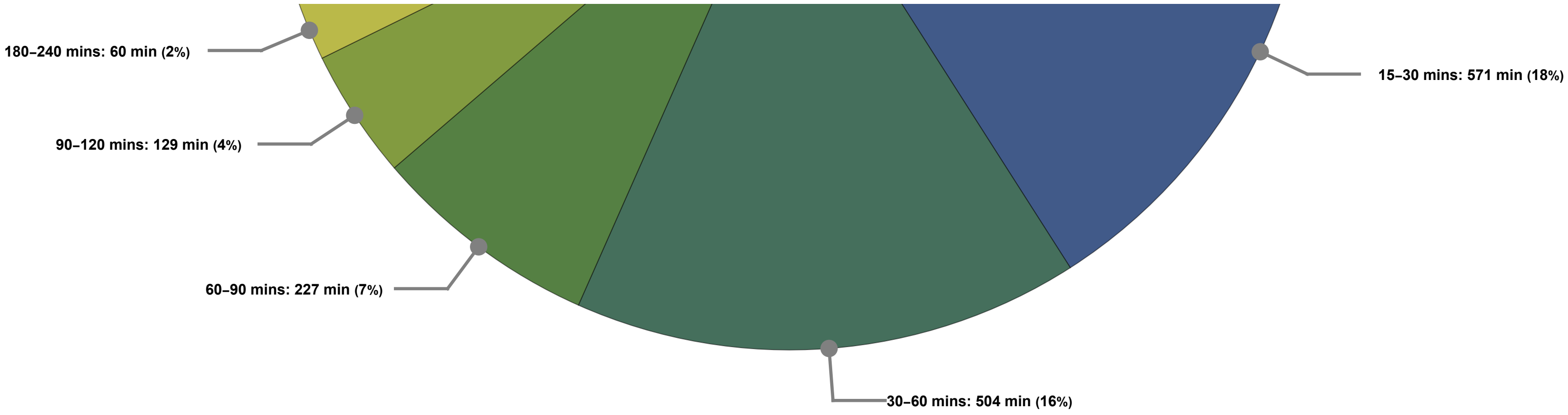
Purpose and Insights: Classify delays into severity buckets and give insights into how severe delays typically are. It helps assess the overall distribution of delay times and understand how severe delays are, which could influence the likelihood of cancellations. We have used the following types of charts for visualization:

i. Pie chart : Delay severity categories (e.g., 0–15 minutes, 15–60 minutes, 60+ minutes).
ii. Histogram: Categorizes delays into severity buckets (e.g., 0–15 minutes, 15–60 minutes, 60+ minutes).

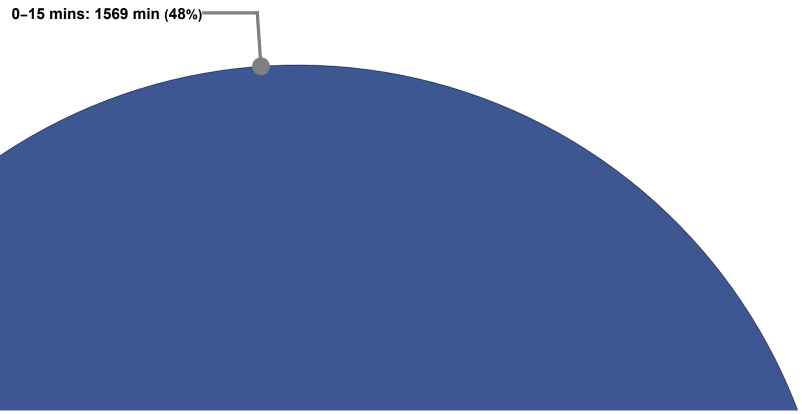
```
In[◦]:= (*Extract delay data for ARR_DELAY and DEP_DELAY*)
arrDelayData = Map[Function[row, row["ARR_DELAY"]], Select[dataset, Not[MissingQ[#["ARR_DELAY"]]] &]];
depDelayData = Map[Function[row, row["DEP_DELAY"]], Select[dataset, Not[MissingQ[#["DEP_DELAY"]]] &]];

(*Filter out non-positive delay values*)
arrDelayData = Select[arrDelayData, # > 0 &]; (*Only consider positive ARR_DELAY*)
depDelayData = Select[depDelayData, # > 0 &]; (*Only consider positive DEP_DELAY*)

(*Classify delays into severity categories using the custom function*)
arrDelayCategories =
  Map[Function[delay, If[delay ≤ 15, "0-15 mins", If[delay ≤ 30, "15-30 mins", If[delay ≤ 60, "30-60 mins", If[delay ≤ 90,
         "60-90 mins", If[delay ≤ 120, "90-120 mins", If[delay ≤ 180, "120-180 mins", If[delay ≤ 240, "180-240 mins",
            If[delay ≤ 360, "240-360 mins", If[delay ≤ 480, "360-480 mins", "480+ mins"]]]]]]]]]], arrDelayData];

depDelayCategories =
  Map[Function[delay, If[delay ≤ 15, "0-15 mins", If[delay ≤ 30, "15-30 mins", If[delay ≤ 60, "30-60 mins", If[delay ≤ 90,
         "60-90 mins", If[delay ≤ 120, "90-120 mins", If[delay ≤ 180, "120-180 mins", If[delay ≤ 240, "180-240 mins",
            If[delay ≤ 360, "240-360 mins", If[delay ≤ 480, "360-480 mins", "480+ mins"]]]]]]]]]], depDelayData];

(*Count occurrences of each category for both ARR_DELAY and DEP_DELAY*)
arrDelayCategoryCounts = Counts[arrDelayCategories];
depDelayCategoryCounts = Counts[depDelayCategories];

(*Create pie charts for both ARR_DELAY and DEP_DELAY severity categories*)
(*Convert the counts to key-value pairs*)
arrDelayCategoryList = Normal[arrDelayCategoryCounts];
depDelayCategoryList = Normal[depDelayCategoryCounts];

(*Create a side-by-side pie chart for ARR_DELAY and DEP_DELAY*)
Row[{PieChart[Values[arrDelayCategoryList], ChartLabels →
     Placed[MapThread[Function[{category, value}, Style[ToString[category] <> ": " <> ToString[Round[value, 1]] <>
          " min (" <> ToString[Round[100 value / Total[Values[arrDelayCategoryList]], 1]] <> "%)", Bold, 8]],
       {Keys[arrDelayCategoryList], Values[arrDelayCategoryList]}], "RadialCallout"],
   PlotLabel → "Arrival Delay Severity Distribution", ImageSize → 850, ChartStyle → "DarkRainbow"],
  PieChart[Values[depDelayCategoryList], ChartLabels →
    Placed[MapThread[Function[{category, value}, Style[ToString[category] <> ": " <> ToString[Round[value, 1]] <>
          " min (" <> ToString[Round[100 value / Total[Values[depDelayCategoryList]], 1]] <> "%)", Bold, 8]],
      {Keys[depDelayCategoryList], Values[depDelayCategoryList]}], "RadialCallout"],
   PlotLabel → "Departure Delay Severity Distribution", ImageSize → 850, ChartStyle → "DarkRainbow"]}]
```

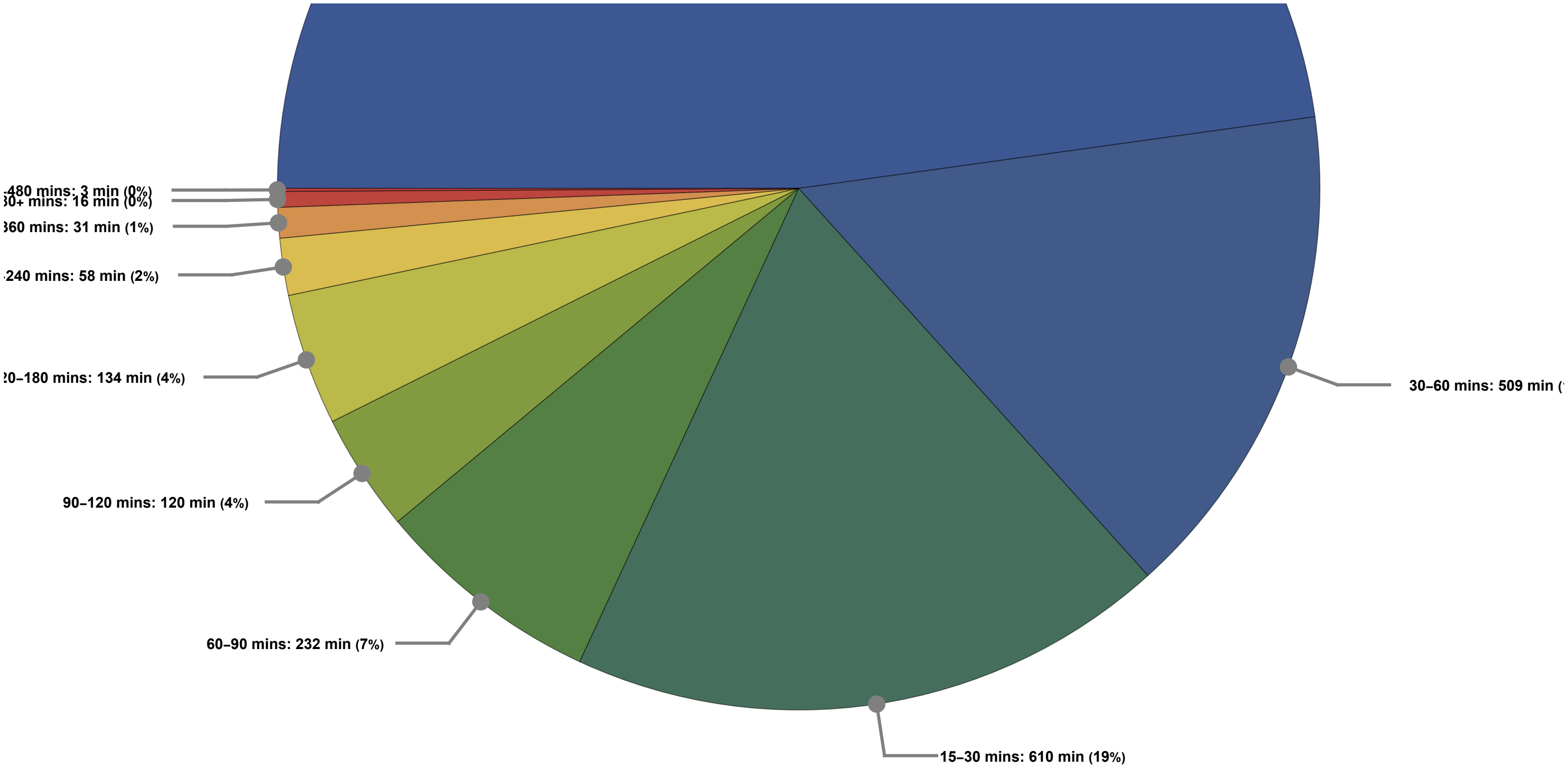
Out[∘]=





Departure Delay Severity Distribution





Purpose:

The “**Arrival Delay Severity Distribution**” pie chart visualizes the distribution of arrival delays across various time intervals. It helps identify the frequency and severity of delays, providing insights into patterns and potential areas for operational improvements.

Segments:

- Each segment represents a different range of delay times and the corresponding number of occurrences.
- The size of each segment corresponds to the number of occurrences within that delay time range.

Insights:

1. Most Frequent Delays:

- The largest segment represents delays of 0-15 minutes with 1545 (48%) occurrences, indicating that most delays are relatively short.

2. Moderate Delays:

- 15-30 minutes and 30-60 minutes segments have 571 (18%) and 504 (16%) occurrences respectively, showing a moderate level of delay severity.

3. Significant Delays:

- The segments for delays of 60-90 minutes and 90-120 minutes have 227 (7%) and 129 (4%) occurrences, indicating less frequent but still significant delays.

4. Severe Delays:

- Delays of 120-180 minutes, 180-240 minutes, and 240-360 minutes have 131 (4%), 60 (2%), and 24 (1%) occurrences respectively, showing a notable impact on operations and passengers.

5. Rare but Extreme Delays:

- 360-480 minutes and 480+ minutes segments are the smallest with 2 and 16 occurrences, representing rare but very severe delays.

Operational Focus:

- Most Frequent Delays: Efforts should focus on reducing the most common short delays to improve overall performance.
- Severe Delays: Airports experiencing significant and severe delays require targeted interventions to mitigate these issues.
- Extremely Rare Delays: While rare, strategies to manage extreme delays should be in place to handle these situations effectively.

Purpose:

The “**Departure Delay Severity Distribution**” pie chart visualizes the distribution of departure delays across various time intervals. It helps identify the frequency and severity of delays, providing insights into patterns and potential areas for operational improvements.

Segments:

- Each segment represents a different range of delay times and the corresponding number of occurrences.
- The size of each segment corresponds to the number of occurrences within that delay time range.

Insights:

1. Most Frequent Delays:

- The largest segment represents delays of 0-15 minutes with 1569 (48%) occurrences, indicating that most delays are relatively short.

2. Moderate Delays:

- 15-30 minutes and 30-60 minutes segments have 610 (19%) and 509 (16%) occurrences respectively, showing a moderate level of delay severity.

3. Significant Delays:

- The segments for delays of 60-90 minutes and 90-120 minutes have 232 (7%) and 120 (4%) occurrences, indicating less frequent but still significant delays.

4. Severe Delays:

- Delays of 120-180 minutes, 180-240 minutes, and 240-360 minutes have 134 (4%), 58 (2%), and 31 (1%) occurrences respectively, showing a notable impact on operations and passengers.

5. Rare but Extreme Delays:

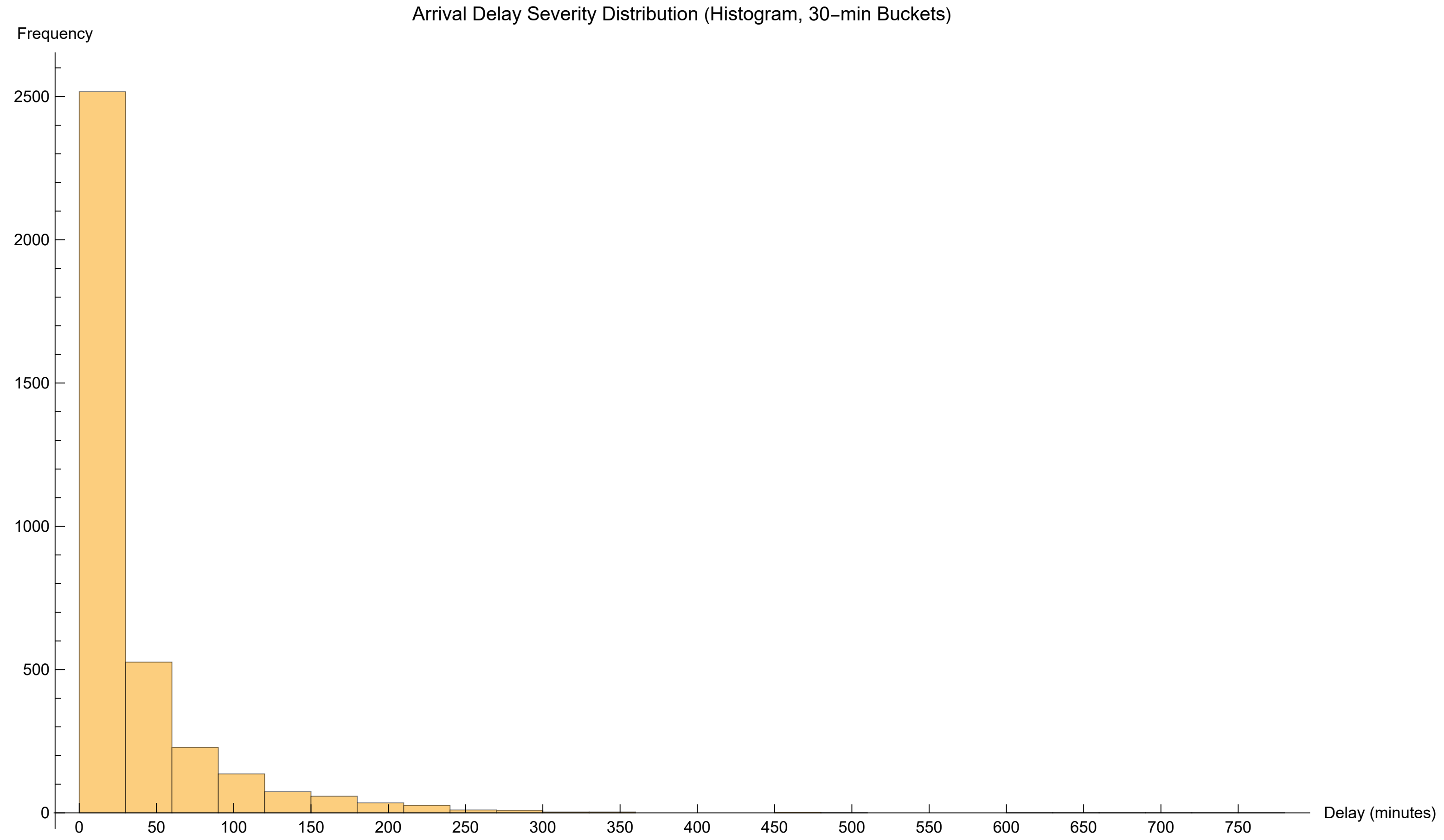
- The 360-480 minutes segment is the smallest with 1 occurrence, representing rare but very severe delays.

Operational Focus:

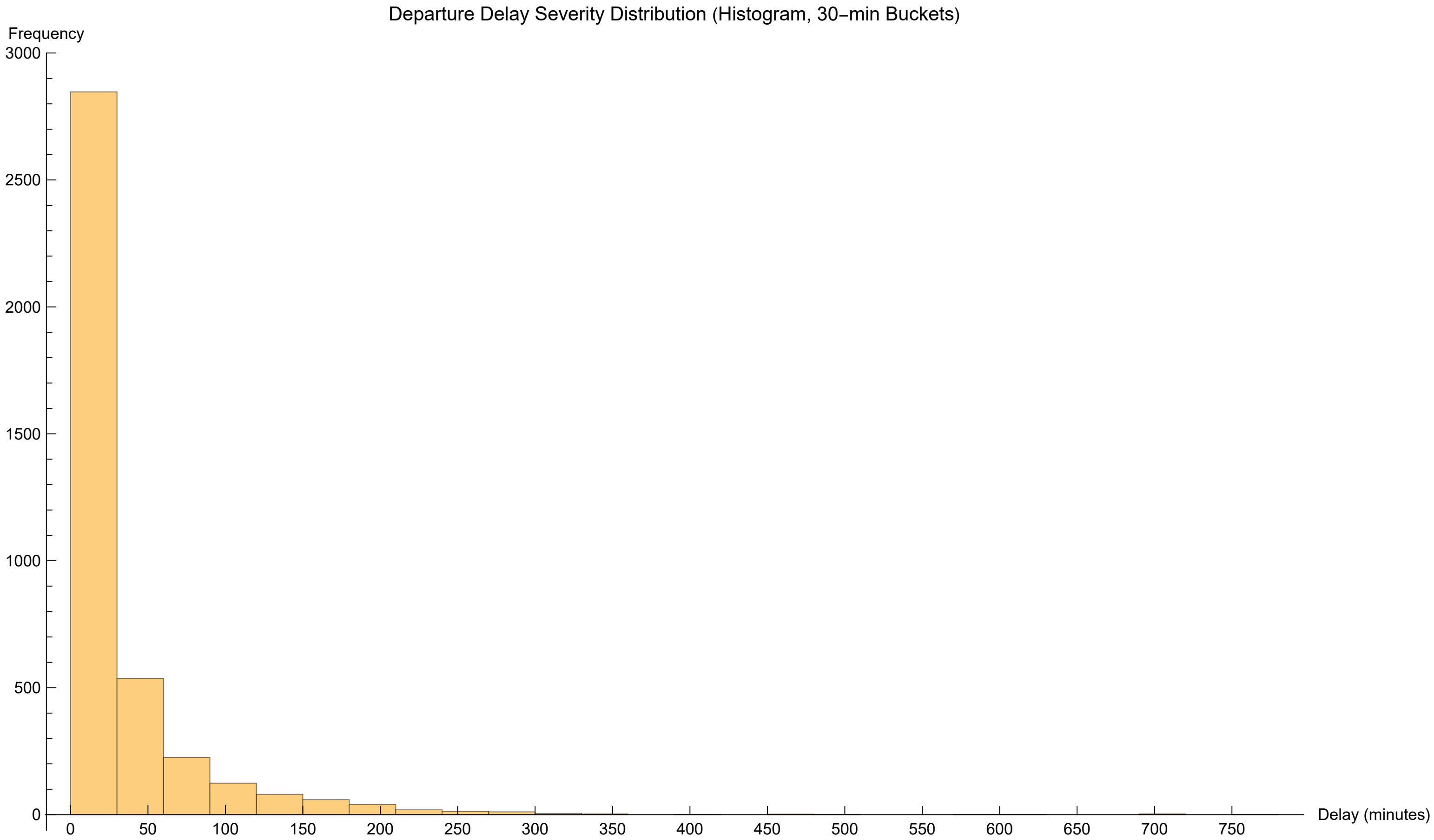
- Most Frequent Delays: Efforts should focus on reducing the most common short delays to improve overall performance.
- Severe Delays: Airports experiencing significant and severe delays require targeted interventions to mitigate these issues.
- Extremely Rare Delays: While rare, strategies to manage extreme delays should be in place to handle these situations effectively.

```
In[•]:= (*Extract delay data and filter valid numeric values*)
        delayData = Map[Function[row, row["ARR_DELAY"]], Select[dataset, NumericQ[#["ARR_DELAY"]] &]];

        (*Define binning specification as a range with equal bin width*)
        binSpec = {0, 800, 30}; (*Start at 0,end at 800,with 30-minute intervals*)

        (*Plot the histogram with custom x-axis ticks*)
        Histogram[delayData, binSpec, ChartLabels → Placed[Automatic, Below], AxesLabel → {"Delay (minutes)", "Frequency"},
         PlotLabel → "Arrival Delay Severity Distribution (Histogram, 30-min Buckets)",
         Ticks → {Range[0, 800, 50], Automatic}, (*Custom ticks every 50 minutes*) ImageSize → 800]

        (*Extract delay data and filter valid numeric values*)
        delayData1 = Map[Function[row, row["DEP_DELAY"]], Select[dataset, NumericQ[#["DEP_DELAY"]] &]];

        (*Define binning specification as a range with equal bin width*)
        binSpec1 = {0, 800, 30}; (*Start at 0,end at 800,with 30-minute intervals*)

        (*Plot the histogram with custom x-axis ticks*)
        Histogram[delayData1, binSpec1, ChartLabels → Placed[Automatic, Below], AxesLabel → {"Delay (minutes)", "Frequency"},
         PlotLabel → "Departure Delay Severity Distribution (Histogram, 30-min Buckets)",
         Ticks → {Range[0, 800, 50], Automatic}, (*Custom ticks every 50 minutes*) ImageSize → 800]
```

Out[ ]=


Arrival Delay Severity Distribution (Histogram, 30–min Buckets)
Frequency
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
Delay (minutes)

Out[∘]=



Purpose:

The histogram, titled “**Arrival Delay Severity Distribution (Histogram, 30-min Buckets)**”, visualizes the frequency distribution of arrival delays in 30-minute intervals. Its purpose is to identify the severity and frequency of delays, helping analyze patterns in flight punctuality.

Y-Axis:

- Label: “Frequency”
- Represents the number of flights within each delay interval.
- Higher bars indicate a higher frequency of flights experiencing delays in that range.

X-Axis:

- Label: “Delay (minutes)”
- Represents delay durations divided into 30-minute intervals (e.g., 0-30, 30-60, etc.).
- Captures the severity of delays, extending up to over 750 minutes.

Insights:

1. Majority of Delays:
    - The majority of arrival delays fall within the 0-30 minutes bucket, showing that most delays are relatively minor.
    - The frequency sharply declines for delays beyond 30 minutes.
2. Long Delays Are Rare:
    - Delays exceeding 120 minutes occur infrequently, as reflected by the significantly smaller bars beyond this range.
    - Extreme delays (e.g., over 600 minutes) are outliers.
3. Operational Implications:
    - Airlines can focus on mitigating short delays, which are more common and likely easier to address.
    - Understanding the causes of extreme outliers can help reduce the rare but severe disruptions.

Purpose:

The histogram, titled “**Departure Delay Severity Distribution (Histogram, 30-min Buckets)**”, visualizes the frequency distribution of departure delays in 30-minute intervals. It helps understand the patterns of departure delays and their severity, aiding in identifying operational inefficiencies and their frequency.

Y-Axis:

- Label: “Frequency”
- Represents the number of flights experiencing departure delays within each time interval.
- Taller bars indicate a higher occurrence of delays within the corresponding range.

X-Axis:

- Label: “Delay (minutes)”
- Represents the duration of departure delays, categorized into 30-minute intervals (e.g., 0-30, 30-60, etc.).
- Shows the severity of delays, extending to over 750 minutes.

Insights:

1. Majority of Delays:

- Most departure delays occur within the 0-30 minutes bucket, as indicated by the tallest bar.
- This suggests that short delays are the most frequent and likely represent minor disruptions.

2. Longer Delays Are Rare:

• The frequency of delays significantly drops beyond 30 minutes, with delays exceeding 200 minutes being extremely rare.

3. Outliers:

• A few flights experience very long delays (e.g., beyond 600 minutes), which are outliers and require further investigation.

4. Operational Opportunities:

• Since short delays are the most common, efforts to streamline processes and reduce minor disruptions could have the most significant impact on improving departure punctuality.

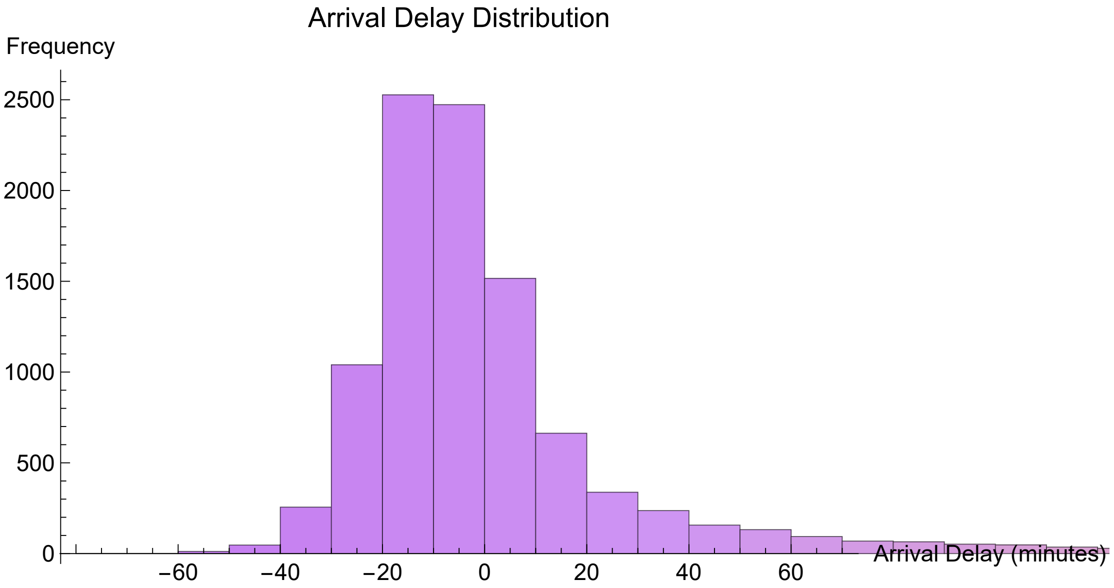
e. Statistical Distributions:

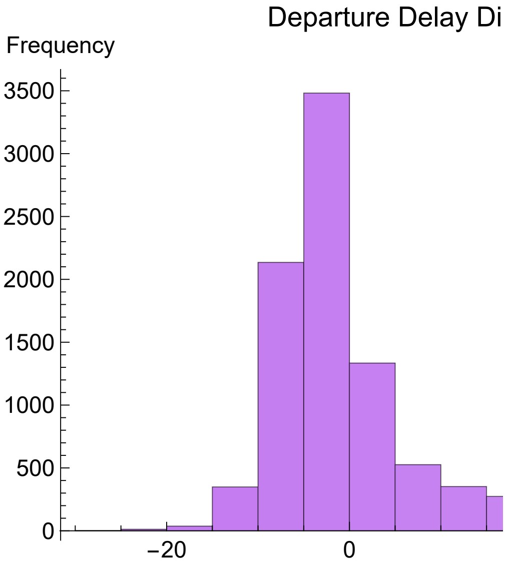
1. Departure Delays vs . Arrival Delays :

Purpose and Insights : It helps to identify which type of delay—departure or arrival—is more common across airports . It helps understand whether delays are mainly happening before takeoff or after landing . A higher frequency of departure delays suggests operational issues at the airport or with takeoff processes, while arrival delays indicate congestion or challenges at the destination airport or during landing .

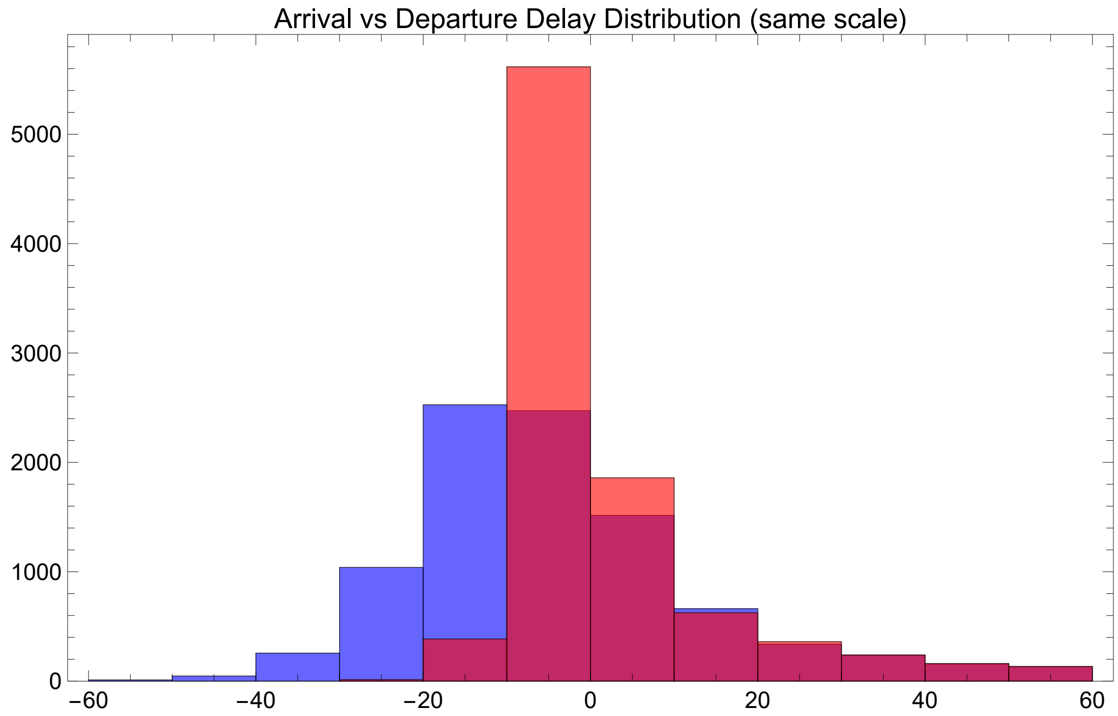
```
In[]:= (*Histogram for Arrival and Departure Delays*)
       arrivalDelayDistribution = Histogram[dataset[All, "ARR_DELAY"], Automatic,
          ChartStyle → "Pastel", AxesLabel → {"Arrival Delay (minutes)", "Frequency"},
          PlotLabel → "Arrival Delay Distribution", LabelStyle → {FontSize → 12}, ImageSize → Large];

       departureDelayDistribution = Histogram[dataset[All, "DEP_DELAY"],
          Automatic, ChartStyle → "Pastel", AxesLabel → {"Departure Delay (minutes)", "Frequency"},
          PlotLabel → "Departure Delay Distribution", LabelStyle → {FontSize → 12}, ImageSize → Large];

       (*Display both distributions*)
       Grid[{{arrivalDelayDistribution, departureDelayDistribution}}]

       (*Define the data for Arrival and Departure Delays*)
       arrivalDelays = dataset[All, "ARR_DELAY"];
       departureDelays = dataset[All, "DEP_DELAY"];

       (*Create histogram for Arrival and Departure Delays in the same graph*)
       delayHistogram =
         Show[Histogram[arrivalDelays, {Range[-60, 60, 10]}, PlotRange → All, AxesLabel → {"Delay (minutes)", "Frequency"},
           PlotLabel → "Arrival vs Departure Delay Distribution (same scale)", LabelStyle → {FontSize → 12},
           ImageSize → Large, ChartStyle → Directive[Blue, Opacity[0.6]], (*This controls the bar color and opacity*)
           Frame → True, Ticks → {Range[-60, 60, 10], Range[0, 3000, 500]}], (*Arrival Delay*)
          Histogram[departureDelays, {Range[-60, 60, 10]}, PlotRange → All, AxesLabel → {"Delay (minutes)", "Frequency"},
           PlotLabel → "Arrival vs Departure Delay Distribution (same scale)", LabelStyle → {FontSize → 12},
           ImageSize → Large, ChartStyle → Directive[Red, Opacity[0.6]], (*Departure delay color and opacity*)
           Frame → True, Ticks → {Range[-60, 60, 10], Range[0, 3000, 500]}] (*Departure Delay*)];

       (*Display the histogram*)
       delayHistogram
```

Out[•]=


Arrival Delay Distribution
Frequency
2500
2000
1500
1000
500
0
−60
−40
−20
0
20
40
60
Arrival Delay (minutes)


Departure Delay Di
Frequency
3500
3000
2500
2000
1500
1000
500
0
−20
0

Out[•]=



Purpose:

This visualization compares the distributions of arrival and departure delays. It includes:

1. Arrival Delay Distribution (Top Left): Frequency of arrival delays.
2. Departure Delay Distribution (Top Right): Frequency of departure delays.
3. Combined Distribution (Bottom): A direct comparison of arrival and departure delay distributions on the same scale.

The purpose is to analyze delay patterns and identify similarities or differences between arrival and departure delays.

Axes:

- X-Axis: Represents delay times (in minutes), ranging from -60 (early) to 60 (late).
- Y-Axis: Represents the frequency of flights within each delay bucket.

Insights:

1. **Arrival Delay Distribution (Top Left)**

- Most arrival delays cluster around -20 to 0 minutes, indicating that many flights arrive early or on time.
- There is a smaller tail of positive delays (flights arriving late), with fewer flights delayed beyond 60 minutes.

2. **Departure Delay Distribution (Top Right)**

- Departure delays are heavily concentrated around 0 to 20 minutes, showing that minor departure delays are common.
- Unlike arrivals, the tail for departure delays is slightly longer, suggesting more frequent occurrences of significant delays.

3. **Combined Distribution (Bottom)**

- Overlap:

  o Both arrival and departure delays center near 0 minutes, reflecting similar patterns of minimal delays.

- Differences:

  o Departure delays show a higher frequency of positive delays (late departures), while arrival delays have a more balanced spread with many negative delays (early arrivals).

- Severe Delays:

  o Long delays (beyond 40 minutes) are more pronounced in departure delays, possibly amplifying their impact on subsequent arrival times.

Key Takeaways:

- Operational Focus: Addressing frequent minor departure delays (e.g., 0-20 minutes) could significantly improve overall timeliness.
- Arrival Efficiency: The negative delays (early arrivals) suggest potential efficiency in handling in-flight operations.
- Late Flights: Reducing severe departure delays can mitigate their cascading effects on arrival delays.

2. Flight Delays Due to Various Reasons :

Purpose and Insights : This analysis identifies delay patterns by type and total, providing a detailed view of operational inefficiencies, weather impacts, security bottlenecks, and late aircraft issues at the top 10 airports with the highest delays . The distribution of delay types—carrier, weather, NAS, security, and late aircraft—is visualized through bar and pie charts, illustrating the proportion and total contribution of each delay type at these airports .

How It Helps :

i . Carrier and NAS Delays : Pinpoints operational inefficiencies for targeted airline or system improvements .

ii . Weather Delays : Highlights region - specific vulnerabilities, aiding in weather preparedness strategies .

iii . Security Delays : Suggests areas for procedural enhancements to minimize disruptions .

iv . Late Aircraft Delays : Indicates cascading effects of previous flight delays, guiding scheduling and turnaround process improvements .

v . Total Delays : Identifies overall delay hotspots, enabling focused operational and infrastructural enhancements .

```
In[∘]:= (*Group delay data by airport and sum delays for each type*)
airportDelays = Normal[GroupBy[dataset, #["ORIGIN"] &, Function[
      flights, <|"CarrierDelay" → Total[Lookup[flights, "DELAY_DUE_CARRIER", 0]], "WeatherDelay" →
        Total[Lookup[flights, "DELAY_DUE_WEATHER", 0]], "NASDelay" → Total[Lookup[flights, "DELAY_DUE_NAS", 0]],
       "SecurityDelay" → Total[Lookup[flights, "DELAY_DUE_SECURITY", 0]],
       "LateAircraftDelay" → Total[Lookup[flights, "DELAY_DUE_LATE_AIRCRAFT", 0]]|>]]];
(*Convert to a dataset for easy handling*)
airportDelayTable = Dataset[KeyValueMap[<|"Airport" → #1, "CarrierDelay" → #2["CarrierDelay"],
       "WeatherDelay" → #2["WeatherDelay"], "NASDelay" → #2["NASDelay"], "SecurityDelay" → #2["SecurityDelay"],
       "LateAircraftDelay" → #2["LateAircraftDelay"]|> &, airportDelays]];

In[∘]:=

In[∘]:= (*Calculate total delays for each airport by summing up all delay types*)
totalDelaysPerAirport = AssociationThread[Keys[airportDelays], (*Airport codes*)Map[
     Function[airportData, Total[{airportData["CarrierDelay"], airportData["WeatherDelay"], airportData["NASDelay"],
        airportData["SecurityDelay"], airportData["LateAircraftDelay"]}]], Values[airportDelays]]];

(*Sort airports by total delays in descending order and take the top 10*)
sortedAirports = SortBy[totalDelaysPerAirport, -# &];
top10Airports = Take[sortedAirports, UpTo[10]]; (*Use UpTo for safety*)

(*Filter the airportDelayTable for the top 10 airports*)
top10AirportDelays = Select[airportDelayTable, MemberQ[Keys[top10Airports], #["Airport"]] &];

(*Display the top 10 airport delays*)
top10AirportDelays
```

```
(*Extract delays for each type across the top 10 airports*)
carrierDelaysTop10 = Flatten[top10AirportDelays[All, "CarrierDelay"]];
weatherDelaysTop10 = Flatten[top10AirportDelays[All, "WeatherDelay"]];
nasDelaysTop10 = Flatten[top10AirportDelays[All, "NASDelay"]];
securityDelaysTop10 = Flatten[top10AirportDelays[All, "SecurityDelay"]];
lateAircraftDelaysTop10 = Flatten[top10AirportDelays[All, "LateAircraftDelay"]];

(*Create bar charts for each delay type for the top 10 airports*)

carrierDelayBarChart = BarChart[carrierDelaysTop10, ChartLabels → Placed[Keys[top10Airports], Below],
   (*Add airport names below each bar*)ChartStyle → "DarkRainbow",
   AxesLabel → {"Airports", "Carrier Delay (minutes)"}, PlotLabel → "Carrier Delay Distribution (Top 10 Airports)",
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large];

weatherDelayBarChart = BarChart[weatherDelaysTop10, ChartLabels → Placed[Keys[top10Airports], Below],
   (*Add airport names below each bar*)ChartStyle → "DarkRainbow",
   AxesLabel → {"Airports", "Weather Delay (minutes)"}, PlotLabel → "Weather Delay Distribution (Top 10 Airports)",
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large];

nasDelayBarChart = BarChart[nasDelaysTop10, ChartLabels → Placed[Keys[top10Airports], Below],
   (*Add airport names below each bar*)ChartStyle → "DarkRainbow",
   AxesLabel → {"Airports", "NAS Delay (minutes)"}, PlotLabel → "NAS Delay Distribution (Top 10 Airports)",
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large];

securityDelayBarChart = BarChart[securityDelaysTop10,
   ChartLabels → Placed[Keys[top10Airports], Below], (*Add airport names below each bar*)ChartStyle → "DarkRainbow",
   AxesLabel → {"Airports", "Security Delay (minutes)"}, PlotLabel → "Security Delay Distribution (Top 10 Airports)",
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large];

lateAircraftDelayBarChart = BarChart[lateAircraftDelaysTop10,
   ChartLabels → Placed[Keys[top10Airports], Below], (*Add airport names below each bar*)
   ChartStyle → "DarkRainbow", AxesLabel → {"Airports", "Late Aircraft Delay (minutes)"},
   PlotLabel → "Late Aircraft Delay Distribution (Top 10 Airports)",
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large];
```

```
(* Total delays for all 5 types of delays*)
(*Create a new table with airport code and total delays*)
airportTotalDelays = Map[Function[airportData, <|"Airport" → airportData["Airport"],
     "TotalDelay" → airportData["CarrierDelay"] + airportData["WeatherDelay"] + airportData["NASDelay"] +
       airportData["SecurityDelay"] + airportData["LateAircraftDelay"] |>], top10AirportDelays];

(*Convert to Dataset format for better readability*)
airportTotalDelaysTable = Dataset[airportTotalDelays];

(*Display the new table*)
airportTotalDelaysTable

(*Create a Bar Chart for Total Delays for Top 10 Airports*)
totalDelaysBarChart = BarChart[airportTotalDelays〚All, "TotalDelay"〛,
   ChartLabels → Placed[Keys[top10Airports], Below], ChartStyle → "DarkRainbow",
   AxesLabel → {"Airports", "Total Delay (minutes)"}, PlotLabel → "Total Delays for Top 10 Airports",
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large, BarSpacing → 0.5];

(*Display histograms in a grid for the top 10 airports*)
Grid[{{carrierDelayBarChart, weatherDelayBarChart},
  {nasDelayBarChart, securityDelayBarChart}, {lateAircraftDelayBarChart, totalDelaysBarChart}}]

(*Summing up all delays for each type for the top 10 airports*)
totalDelaysByTypeTop10 = <|"Carrier Delay" → Total[carrierDelaysTop10],
   "Weather Delay" → Total[weatherDelaysTop10], "NAS Delay" → Total[nasDelaysTop10],
   "Security Delay" → Total[securityDelaysTop10], "Late Aircraft Delay" → Total[lateAircraftDelaysTop10] |>;

(*Create a pie chart showing the proportion of each type of delay in the top 10 airports*)
PieChart[Values[totalDelaysByTypeTop10],
 ChartLabels → Placed[MapThread[Function[{type, value}, StringJoin[type, ": ", ToString[Round[value, 1]],
      " min (", ToString[Round[100 value / Total[Values[totalDelaysByTypeTop10]], 1]], "%)"]],
    {Keys[totalDelaysByTypeTop10], Values[totalDelaysByTypeTop10]}], "RadialCenter"],
 PlotLabel → "Proportion of Delay Types in Top 10 Airports", ImageSize → Large,
 ChartStyle → "DarkRainbow", LabelStyle → Directive[FontSize → 10, Bold]]
```

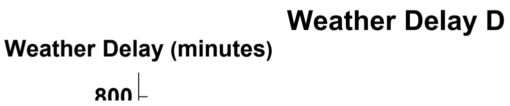
Out[∘]=

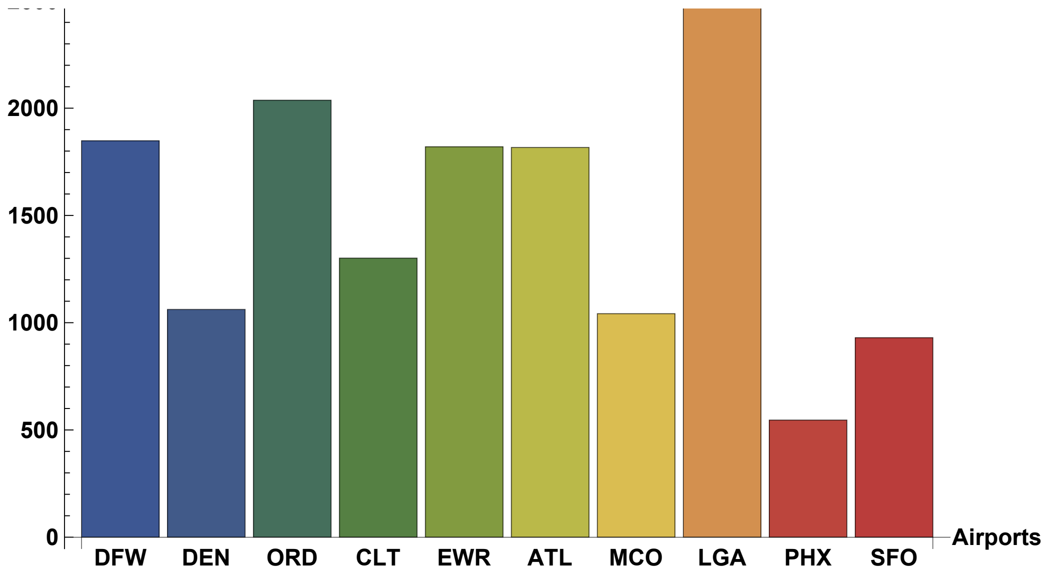
Airport	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay
DEN	1848	271	694	0	2189
ORD	1062	786	842	0	2243
DFW	2037	793	633	0	2908
EWR	1301	216	1375	16	1434
CLT	1820	57	791	0	1797
ATL	1817	121	1172	0	1085
PHX	1042	58	140	0	1140
MCO	2471	46	774	0	898
LGA	546	40	642	0	1157
SFO	930	0	482	0	927

Out[∘]=

Airport	TotalDelay
DEN	5002
ORD	4933
DFW	6371
EWR	4342
CLT	4465
ATL	4195
PHX	2380
MCO	4189
LGA	2385
SFO	2339

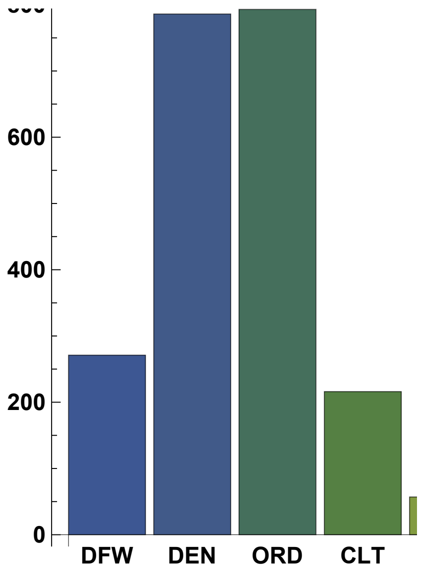
Out[∘]=





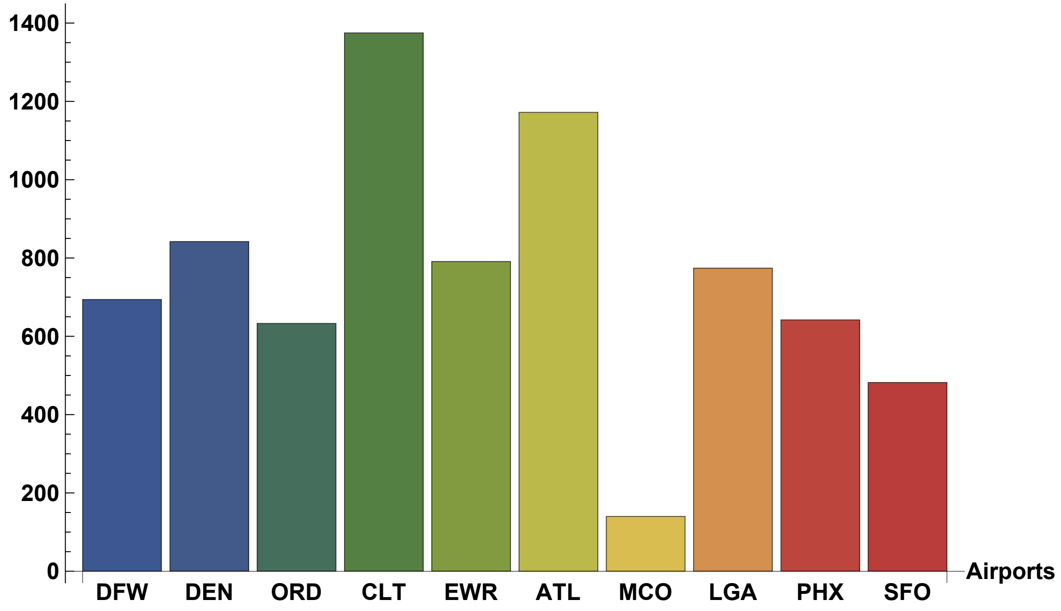
NAS Delay Distribution (Top 10 Airports)

NAS Delay (minutes)



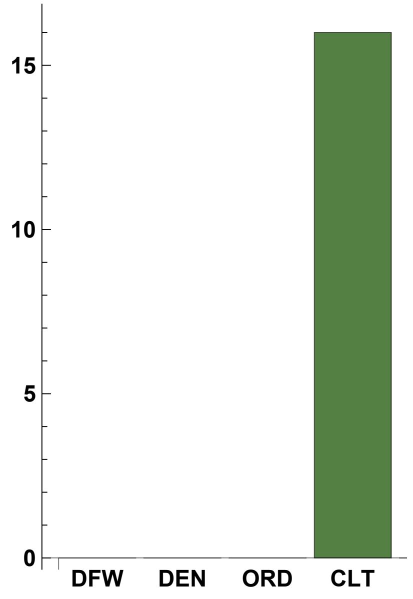
Security Delay D

Security Delay (minutes)



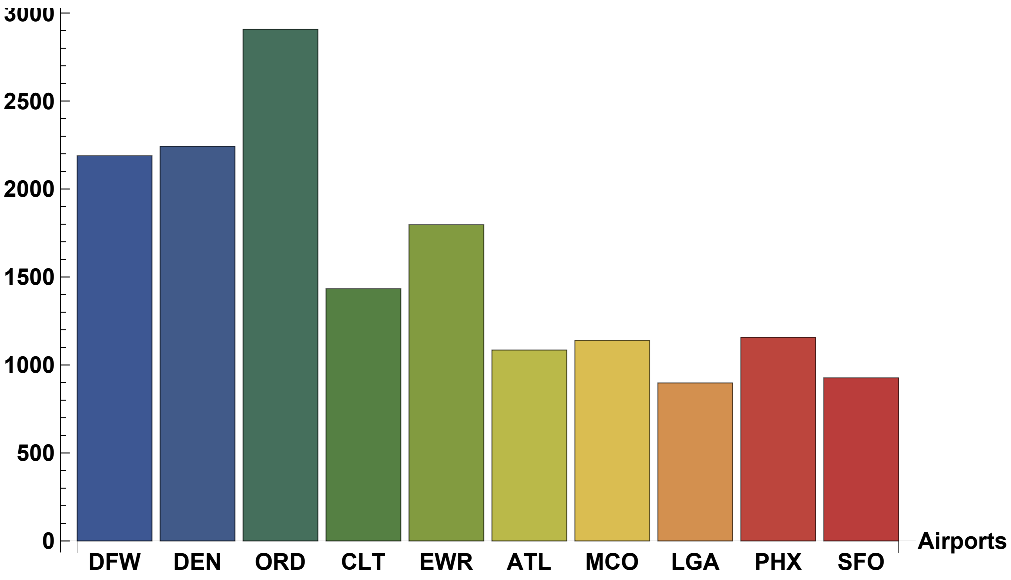
Late Aircraft Delay Distribution (Top 10 Airports)

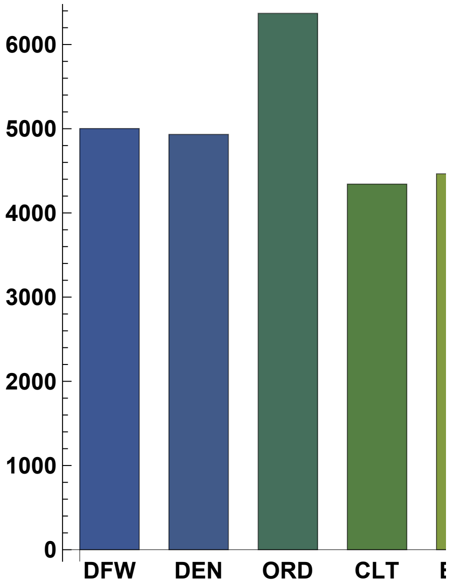
Late Aircraft Delay (minutes)



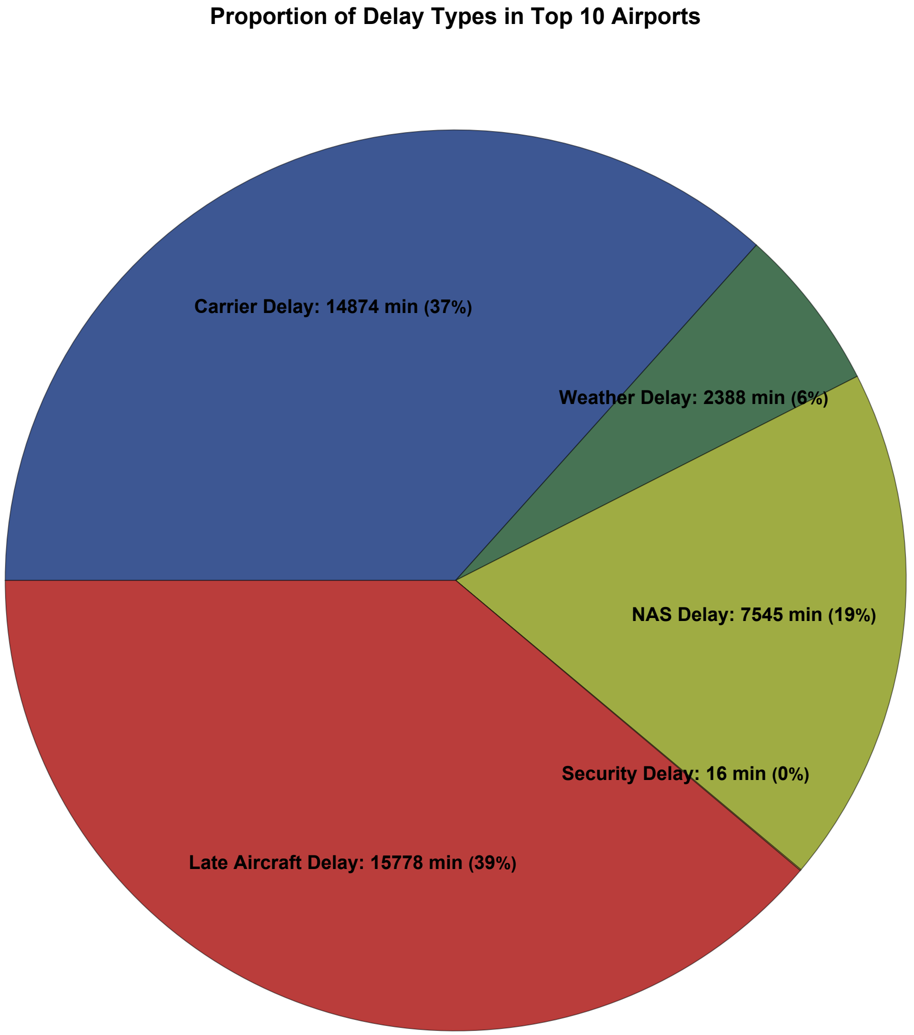
Total Delay

Total Delay (minutes)


3000
2500
2000
1500
1000
500
0
Airports
DFW
DEN
ORD
CLT
EWR
ATL
MCO
LGA
PHX
SFO


6000
5000
4000
3000
2000
1000
0
DFW
DEN
ORD
CLT
E

Out[•]=



Purpose:

The graphs visualize the distribution of different types of delays (Carrier, Weather, NAS, Security, Late Aircraft) and the total delays for the top 10 airports in the United States. These graphs help identify airports that experience significant delays, providing insights for operational improvements and traveler expectations.

Graphs:

1. Carrier Delay Distribution (Top 10 Airports)

2. Weather Delay Distribution (Top 10 Airports)

3. NAS Delay Distribution (Top 10 Airports)

4. Security Delay Distribution (Top 10 Airports)

5. Late Aircraft Delay Distribution (Top 10 Airports)

6. Total Delays for Top 10 Airports

7. Proportion of Delay Types in Top 10 Airports

Y-Axis (For Bar Graphs):

- Label: "Total Delay (minutes)"
- Representation: The total delay time for each type of delay.
- Scale: Increases in minutes, showing the magnitude of delays for each airport.

X-Axis (For Bar Graphs):

- Label: "Airports"
- Representation: The airport codes (e.g., DFW, DEN, ORD) for the top 10 airports with the highest delays.
- Bars: Each bar corresponds to one airport.

Segments (For Pie Chart):

- Each segment represents a different type of delay.
- The size of each segment corresponds to the proportion of the total delays for each type.

Insights:

1. **Carrier Delay Distribution (Top 10 Airports)**

- Longest Delays:
  - LGA (LaGuardia) has the highest carrier delay, exceeding 2500 minutes, which is a significant outlier.
- Moderately High Delays:
  - ORD (O'Hare) and DFW (Dallas/Fort Worth) have delays averaging around 2200 and 2000 minutes, respectively.
- Lower Delays in Top 10:
  - Airports like PHX (Phoenix) and SFO (San Francisco) have delays averaging around 500 and 1000 minutes, respectively.
- Operational Focus:
  - LGA requires immediate attention to address excessive delays, while airports with moderate delays could benefit from efficiency improvements.

2. **Weather Delay Distribution (Top 10 Airports)**

- Longest Delays:
  - DEN (Denver) and ORD (O'Hare) have the highest weather delays, each around 800 minutes.
- Moderately High Delays:
  - CLT (Charlotte) and DFW (Dallas/Fort Worth) have delays averaging around 400 and 300 minutes, respectively.
- Lower Delays in Top 10:
  - Airports like SFO (San Francisco) have negligible delays, indicating better weather management.
- Operational Focus:
  - Airports with moderate delays should enhance weather mitigation strategies to reduce delays.

3. **NAS Delay Distribution (Top 10 Airports)**

- Longest Delays:
  - CLT (Charlotte) has the highest NAS delay, exceeding 1400 minutes.
- Moderately High Delays:
  - ATL (Atlanta) and EWR (Newark) have delays averaging around 1200 and 1000 minutes, respectively.
- Lower Delays in Top 10:
  - Airports like MCO (Orlando) and PHX (Phoenix) have delays averaging around 200 and 600 minutes, respectively.
- Operational Focus:
  - CLT and other airports with significant NAS delays need to focus on enhancing air traffic management and reducing delays.

4. **Security Delay Distribution (Top 10 Airports)**

- Insights:
  - Most airports have negligible security delays, with only CLT (Charlotte) having a measurable delay of around 15 minutes.
- Operational Focus:
  - Airports are generally performing well in managing security delays. Maintaining or enhancing current security protocols can ensure minimal disruptions.

5. **Late Aircraft Delay Distribution (Top 10 Airports)**

- Longest Delays:
  - ORD (O'Hare) has the highest late aircraft delay, followed by DEN (Denver) and DFW (Dallas/Fort Worth).
- Moderately High Delays:
  - CLT (Charlotte) and EWR (Newark) have moderate delays, indicating a need for improved scheduling and turnaround times.
- Lower Delays in Top 10:
  - Airports like PHX (Phoenix) and SFO (San Francisco) have relatively lower late aircraft delays.
- Operational Focus:
  - Airports with high late aircraft delays should improve coordination and efficiency to minimize these delays.

6. **Total Delays for Top 10 Airports**
   - Longest Delays:
     - ORD (O’Hare) has the highest total delays, followed by DEN (Denver) and DFW (Dallas/Fort Worth).
   - Moderately High Delays:
     - CLT (Charlotte) and EWR (Newark) have notable total delays, requiring targeted operational improvements.
   - Lower Delays in Top 10:
     - Airports like PHX (Phoenix) and SFO (San Francisco) have relatively lower total delays.
   - Operational Focus:
     - Airports with high total delays should prioritize strategies to reduce delays across all categories.
7. **Proportion of Delay Types in Top 10 Airports**
   - Largest Proportion:
     - Late Aircraft Delay (39%) and Carrier Delay (37%) has the largest proportion, indicating it is a significant contributor to overall delays.
   - Moderate Proportions:
     - NAS Delay (19%) also contributes notably to total delays.
   - Smaller Proportions:
     - Weather Delay (6%) and Security Delay (~0%) have smaller proportions, indicating better management in these areas.
   - Operational Focus:
     - Focusing on reducing NAS and Late Aircraft delays can significantly improve overall airport performance.

f. Summary Statistics Table:

Purpose and Insights: This shows all key statistics in a table, such as number of flights, airports, airlines, cancellations, diversions, flight times, and delays and help visualize the dataset with descriptive statistics. It also provides a concise view of central tendency (mean, median) and variability (standard deviation) for numerical columns, useful for detecting outliers or understanding typical values.

```
In[•]:= (*Compute Total Values*)
totalFlights = Length[dataset];
totalCancellations = Total[dataset[All, "CANCELLED"]];
totalAirports1 = Length[DeleteDuplicates[dataset[All, "ORIGIN"]]];
totalAirports2 = Length[DeleteDuplicates[dataset[All, "DEST"]]];
totalAirlines = Length[DeleteDuplicates[dataset[All, "AIRLINE"]]];
totalDiverted = Total[dataset[All, "DIVERTED"]];
totalFlightTime = Total[Select[dataset[All, "AIR_TIME"], NumericQ]]
totalArrivalDelays = Total[Select[dataset[All, "ARR_DELAY"], NumericQ]];
totalDepartureDelays = Total[Select[dataset[All, "DEP_DELAY"], NumericQ]];

(*Compute Descriptive Statistics*)
computeStats[column_] := Module[{data = Select[dataset[All, column], NumericQ]}, <|"Mean" → Mean[data],
    "Median" → Median[data], "Standard Deviation" → StandardDeviation[data], "Min" → Min[data], "Max" → Max[data] |>];

(*Numerical Columns*)
delayStats = <|"Arrival Delay" → computeStats["ARR_DELAY"], "Departure Delay" → computeStats["DEP_DELAY"],
   "Cancellations" → computeStats["CANCELLED"], "Diversions" → computeStats["DIVERTED"] |>;

(*Summary Table:Key Metrics in Side-by-Side Format*)
summaryStatsTable = Dataset[KeyValueMap[Function[{metric, value}, <|"Metric" → metric, "Value" → value|>], <|
     "Total Flights" → totalFlights, "Total Airlines" → totalAirlines,
     "Total Departure Airports" → totalAirports1, "Total Arrival Airports" → totalAirports2,
     "Total Cancellations" → totalCancellations, "Total Diverted" → totalDiverted,
     "Total Flight Time (minutes)" → totalFlightTime, "Total Arrival Delays (minutes)" → totalArrivalDelays,
     "Total Departure Delays (minutes)" → totalDepartureDelays|>]];

(*Summary Table:Detailed Statistics*)
detailedStatsTable = Dataset[
   KeyValueMap[Function[{metric, stats}, <|"Metric" → metric, "Mean" → stats["Mean"], "Median" → stats["Median"],
      "Standard Deviation" → stats["Standard Deviation"], "Min" → stats["Min"], "Max" → stats["Max"] |>], delayStats]];

(*Display Tables*)
{summaryStatsTable, detailedStatsTable}
```

Out[∘]=

```
1 098 199
```

Out[∘]=

{

Metric	Value
Total Flights	10 000
Total Airlines	18
Total Departure Airports	321
Total Arrival Airports	310
Total Cancellations	242
Total Diverted	19
Total Flight Time (minutes)	1 098 199
Total Arrival Delays (minutes)	38 739
Total Departure Delays (minutes)	97 810

,

Metric	Mean	Median	Standard Deviation	Min	Max
Arrival Delay	3.8739	−6	50.1566	−75	1163
Departure Delay	9.781	−2	48.3203	−28	1130
Cancellations	0.0242	0	0.153677	0	1
Diversions	0.0019	0	0.0435497	0	1

}

For Question 2 : Which airline company has the highest on - time rate?

a. Airlines and their frequencies for flights:

Purpose and Insights : The airline frequency plot helps visualize the distribution of flights across different airlines in the dataset . The airlines and their frequencies plot helps identify the most represented airlines in the dataset, prioritize them for further analysis of on-time performance and delays, and assess which ones are more likely to have higher delay or cancellation rates.

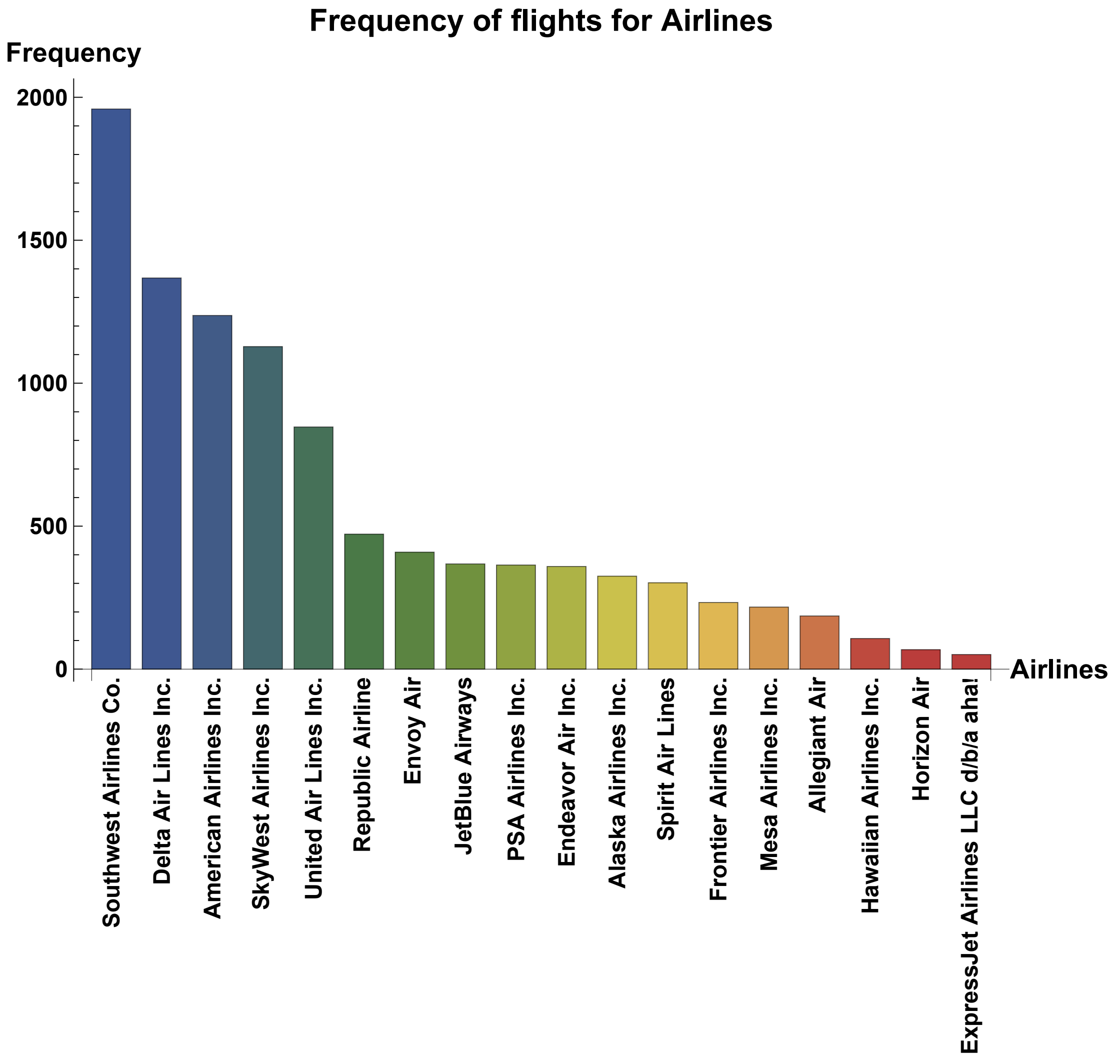
```
In[∘]:= (*Extract the'AIRLINE' column*)airlineData = Normal[dataset[All, "AIRLINE"]];

(*Count the frequency of each airline*)
airlineCounts = Tally[airlineData];

(*Sort by frequency in descending order*)
sortedAirlineCounts = ReverseSortBy[airlineCounts, Last];

(*Separate airlines and their frequencies*)
airlines = sortedAirlineCounts⟦All, 1⟧;
frequencies = sortedAirlineCounts⟦All, 2⟧;

(*Create a table for display*)
airlineTable = Dataset[AssociationThread[{"Airline", "Frequency"} → #] & /@ sortedAirlineCounts];

(*Show the table*)
Print["Airline Frequencies:"];
airlineTable

(*Create a bar chart with vertically oriented airline names*)
BarChart[frequencies, ChartLabels → Placed[Rotate[#, 90 Degree] & /@ airlines, Below],
 ChartStyle → "DarkRainbow", BarSpacing → 0.3, LabelStyle → {FontSize → 12, Bold}, ImageSize → Large,
 AxesLabel → {Style["Airlines", FontSize → 14, Bold], Style["Frequency", FontSize → 14, Bold]},
 TicksStyle → Directive[FontSize → 10], PlotLabel → Style["Frequency of flights for Airlines", FontSize → 16, Bold]]
```

```
Airline Frequencies:
```

Out[∘]=

Airline	Frequency
Southwest Airlines Co.	1959
Delta Air Lines Inc.	1368
American Airlines Inc.	1237
SkyWest Airlines Inc.	1128
United Air Lines Inc.	847
Republic Airline	472
Envoy Air	409
JetBlue Airways	368
PSA Airlines Inc.	364
Endeavor Air Inc.	359
Alaska Airlines Inc.	325
Spirit Air Lines	302
Frontier Airlines Inc.	233
Mesa Airlines Inc.	217
Allegiant Air	186
Hawaiian Airlines Inc.	107
Horizon Air	68
ExpressJet Airlines LLC d/b/a aha!	51

Out[ ]=



Purpose:

The “**Frequency of Flights for Airlines**” bar chart visualizes the frequency of flights for various airlines. This chart provides insights into the operational scale and market share of different airlines, helping to understand travel trends and competitive dynamics.

Y-Axis:

- Label: “Frequency”

- Representation: The number of flights conducted by each airline.
- Scale: Increases with the number of flights, showing the relative frequency for each airline.

X-Axis:

- Label: “Airlines”
- Representation: The names of the airlines.
- Bars: Each bar represents one airline, with its height indicating the number of flights.

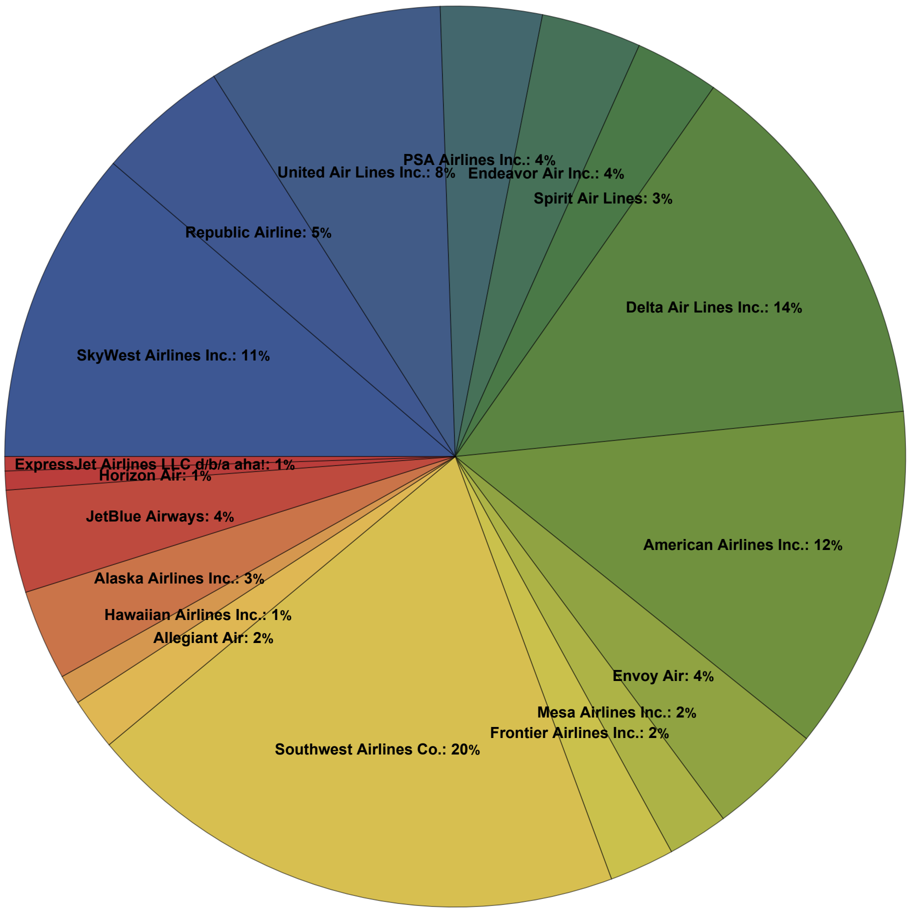
Insights:

1. Highest Frequency:
   - Southwest Airlines Co. has the highest frequency of flights, indicating its significant market share and operational scale.
2. Major Competitors:
   - Delta Air Lines Inc., American Airlines Inc., and United Air Lines Inc. follow, showing robust flight frequencies that highlight their competitive positions in the market.
3. Moderate Frequency:
   - Other airlines, such as JetBlue Airways Corp. and Alaska Airlines Inc., have moderate frequencies, indicating their niche markets or specific operational focus.
4. Lower Frequency:
   - Airlines like Spirit Airlines Inc. and Frontier Airlines Inc. show lower frequencies, possibly reflecting more focused routes or smaller market segments.
5. Operational Insights:
   - Airlines with higher frequencies might prioritize maintaining operational efficiency and expanding their route networks, while those with lower frequencies may focus on niche markets or specific customer segments.

```
In[•]:= (*Calculate flight volume per airline*)
        flightVolume1 = Normal[GroupBy[dataset, #AIRLINE &, Length]];

        (*Calculate total number of flights*)
        totalFlights1 = Total[Values[flightVolume1]];

        (*Calculate share of flight volume for each airline*)
        airlineShares = AssociationThread[Keys[flightVolume1], Values[flightVolume1] / totalFlights1];

        (*Create Pie Chart for Airlines by Share of Flight Volume*)
        PieChart[Values[airlineShares],
         ChartLabels → Placed[MapThread[(Style[ToString[#1] <> ": " <> ToString[Round[#2 * 100, 1]] <> "%", Bold, 8]) &,
            {Keys[airlineShares], Values[airlineShares]}], "RadialCenter"], ChartStyle → "DarkRainbow",
         PlotLabel → "Airlines by Share of Flight Volume", ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]


        (*Sum the frequencies of the top 5 airlines*)
        top5Flights1 = Total[TakeLargest[frequencies, 5]];

        (*Calculate the percentage*)
        perc2 = Round[top5Flights1 / totalFlights1, 0.01];

        (*Print the result*)
        Print["The top 5 airlines provide ", perc2 * 100, "% of the total flights in the US."];
```

Out[∘]=

Airlines by Share of Flight Volume



```
The top 5 airlines provide 65.% of the total flights in the US.
```

Purpose:

The “**Airlines by Share of Flight Volume**” pie chart visualizes the distribution of flight volumes among various airlines in the United

States. This chart provides insights into the market share and operational scale of different airlines, helping to understand travel trends and competitive dynamics.

Segments:

• Each segment represents an airline and its corresponding share of the total flight volume.

• The size of each segment corresponds to the proportion of the total flights provided by that airline.

Insights:

1. Largest Market Share:

• Southwest Airlines Co. has the largest market share, with 20% of the total flights, indicating its significant presence in the U.S. aviation market.

2. Major Competitors:

• Delta Air Lines Inc. (14%) and American Airlines Inc. (12%) are key competitors, showcasing their strong market positions.

• SkyWest Airlines Inc. (11%) and United Air Lines Inc. (8%) also have considerable shares.

3. Moderate Market Share:

• Airlines like Republic Airline (5%), PSA Airlines Inc. (4%), and Endeavor Air Inc. (4%) hold moderate shares, reflecting their operational focus.

4. Lower Market Share:

• Spirit Airlines Inc. (3%) and JetBlue Airways (4%) have lower shares, possibly focusing on specific routes or market segments.

5. Operational Insights:

• The top 5 airlines provide 65.8% of the total flights in the U.S., highlighting their dominant market positions. Strategies for maintaining this dominance include expanding route networks and enhancing operational efficiency.

• Airlines with smaller shares can explore niche markets or improve service quality to increase their market share.

b. On - Time Performance by Airline:

Purpose and Insights: By comparing on-time rates across airlines, we can identify which airlines perform better and are more punctual, which is essential for comparing performance across airlines and also sets the foundation for further analysis. We have used the following types of charts for visualization:

i. Bar Chart: This chart will show the on-time performance of airlines, where the on-time rate is defined as flights with ARR_DELAY <= 0 or DEP_DELAY<=0, and CANCELLED == 0, and DIVERTED == 0.

ii. Box Plot: This illustrates the spread and distribution of on-time rates, revealing any potential outliers or variations.

iii. Pie Chart: This offers a visual breakdown of the share of on-time rates among airlines, ideal for understanding proportions.

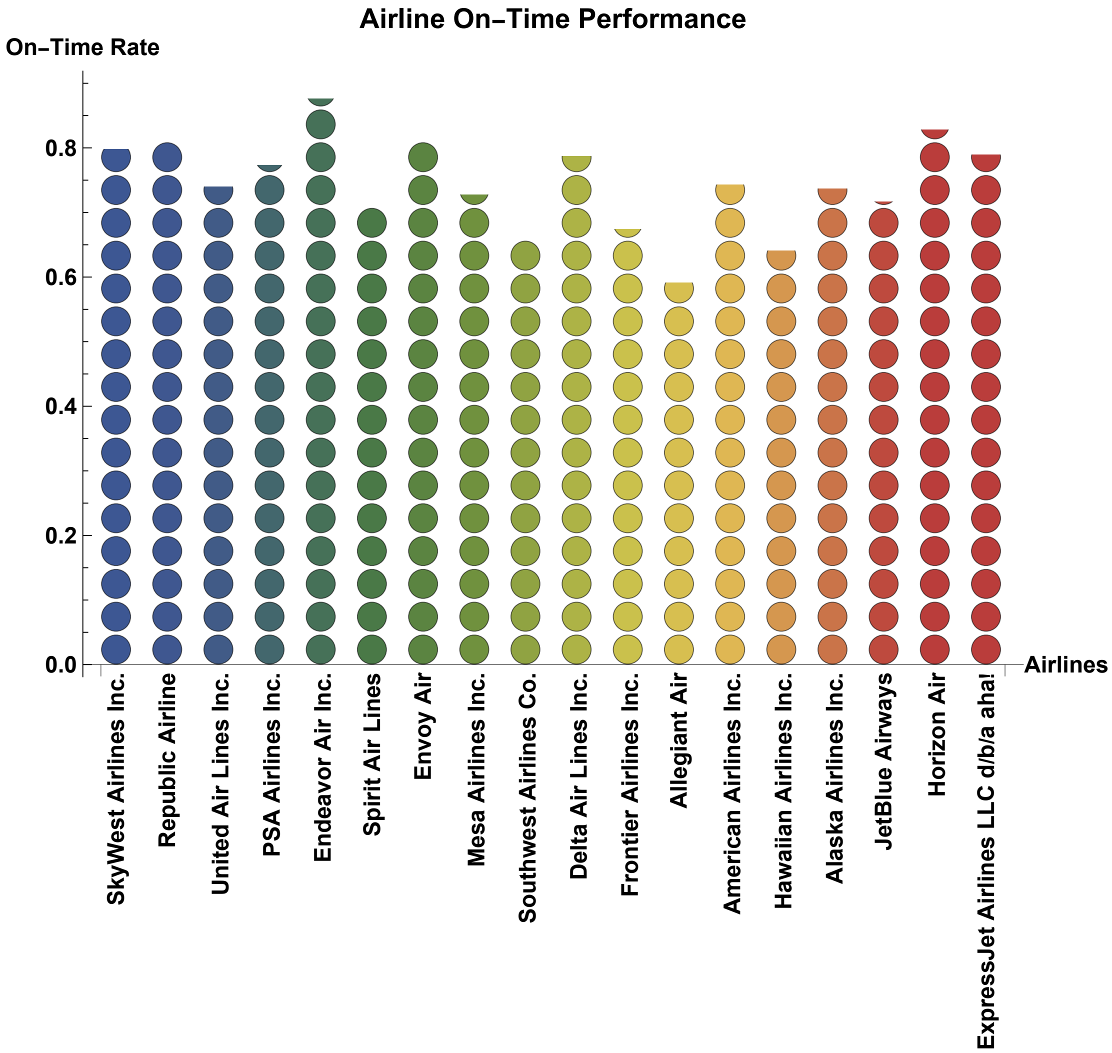
```
In[•]:= (*Filter the dataset to keep only the rows that satisfy the conditions:ARR_DELAY≤0,
DEP_DELAY≤0,CANCELLED==0,DIVERTED==0*)
filteredData = Select[dataset, ((#⟦"ARR_DELAY"⟧ ≤ 0 || #⟦"DEP_DELAY"⟧ ≤ 0) && #⟦"CANCELLED"⟧ == 0 && #⟦"DIVERTED"⟧ == 0) &];

(*Group the filtered data by AIRLINE and count the number of valid flights for each airline*)
onTimeCounts = Normal[GroupBy[filteredData, #["AIRLINE"] &, Length]];

(*Calculate the total number of flights for each airline (including on-time and non-on-time flights)*)
totalFlights = Normal[GroupBy[dataset, #["AIRLINE"] &, Length]];

(*Calculate the on-time rate for each airline as a fraction of its own total flights*)
onTimeRates =
  AssociationThread[Keys[onTimeCounts], N[Values[onTimeCounts] / Lookup[totalFlights, Keys[onTimeCounts], 1], 2]];

(*Convert the on-time rates to a table format for better visualization*)
onTimeRatesTable = Dataset[KeyValueMap[<|"Airline" → #1, "On-Time Rate" → #2|> &, onTimeRates]];

(*Display the on-time rates table*)
onTimeRatesTable
```

Out[◦]=

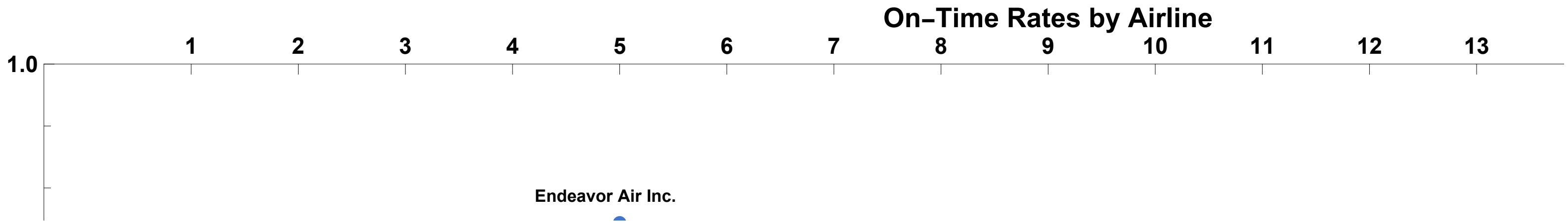
Airline	On–Time Rate
SkyWest Airlines Inc.	0.79
Republic Airline	0.81
United Air Lines Inc.	0.74
PSA Airlines Inc.	0.77
Endeavor Air Inc.	0.87
Spirit Air Lines	0.71
Envoy Air	0.8
Mesa Airlines Inc.	0.72
Southwest Airlines Co.	0.65
Delta Air Lines Inc.	0.78
Frontier Airlines Inc.	0.67
Allegiant Air	0.59
American Airlines Inc.	0.74
Hawaiian Airlines Inc.	0.64
Alaska Airlines Inc.	0.73
JetBlue Airways	0.71
Horizon Air	0.82
ExpressJet Airlines LLC d/b/a aha!	0.78

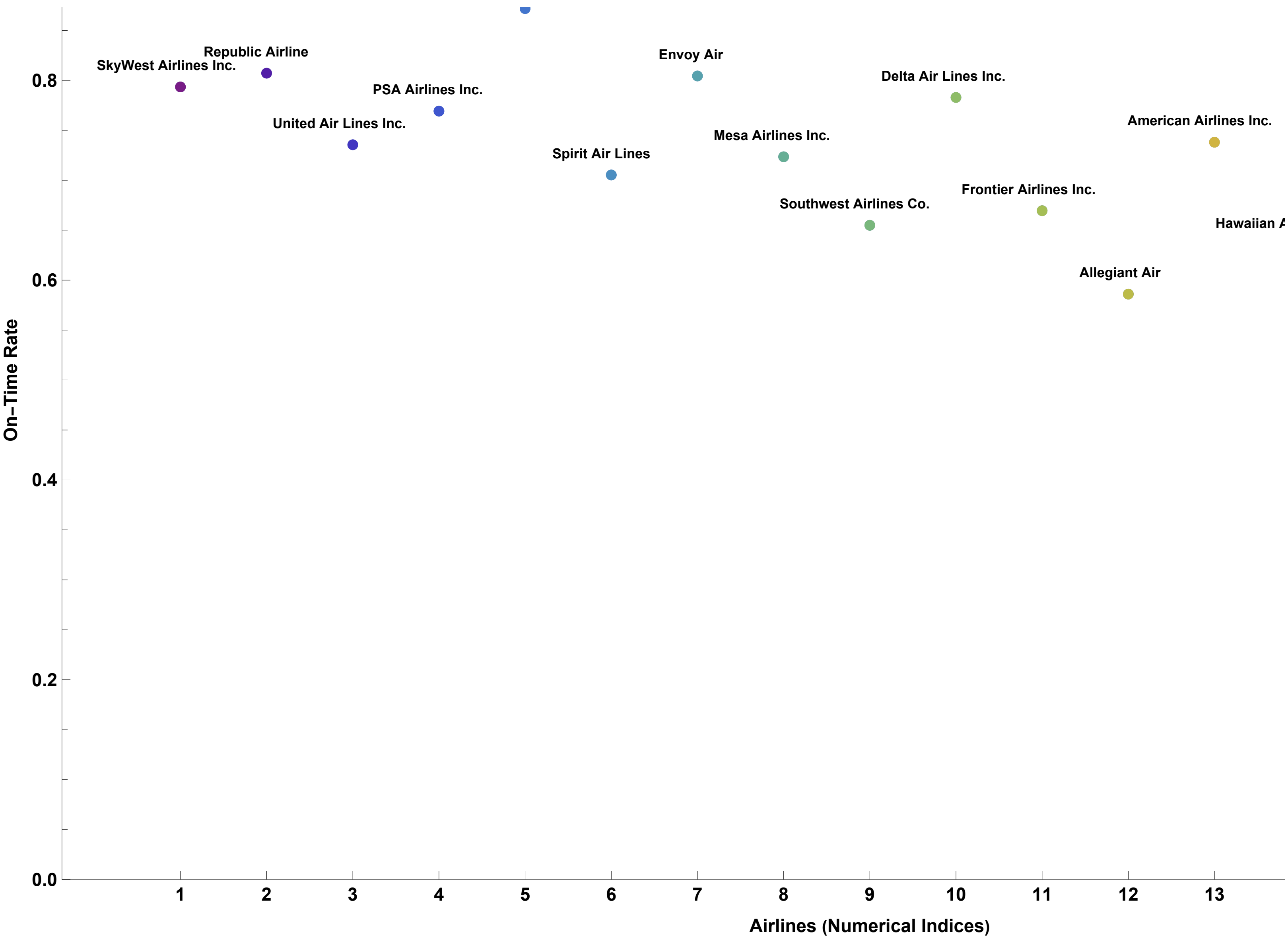
```
In[ ]:= (*Create a Bar Chart for On-Time Rates of Airlines*)
        BarChart[Values[onTimeRates], ChartLabels → Placed[Rotate[#, 90 Degree] & /@ Keys[onTimeRates], Below],
         ChartStyle → "DarkRainbow", ChartElements → ●, AxesLabel → {"Airlines", "On-Time Rate"}, BarSpacing → 0.5,
         PlotLabel → "Airline On-Time Performance", LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]

        (*Define colors for airlines*)
        airlineColors =
          AssociationThread[Keys[onTimeRates], ColorData["Rainbow"] /@ Subdivide[0, 1, Length[onTimeRates] - 1]];
        (*Dot plot with airline names displayed near dots and distinct colors*)
        ListPlot[MapIndexed[Style[{#2[[1]], #1}, airlineColors[Keys[onTimeRates][[#2[[1]]]]]] &, Values[onTimeRates]],
         PlotRange → {0, 1}, PlotStyle → PointSize[Large], Axes → False, Frame → True,
         FrameTicks → {Range[Length[onTimeRates]], (*Numerical indices on x-axis*) Automatic},
         FrameLabel → {"Airlines (Numerical Indices)", "On-Time Rate"}, PlotLabel → "On-Time Rates by Airline",
         Epilog → {MapIndexed[Text[Style[Keys[onTimeRates][[#2[[1]]]], Bold, FontSize → 9, Black], {#2[[1]], #1 + 0.02},
              (*Slight offset for better readability*) {0.2, -0.2} (*Offset to position near dots*)] &,
            Values[onTimeRates]]}, ImageSize → 1100, LabelStyle → Directive[Bold, FontSize → 12]]

        (*Pie chart for on-time rates with airline names*) PieChart[Values[onTimeRates],
         ChartLabels → Placed[MapThread[Style[ToString[#1] <> ": " <> ToString[NumberForm[#2, {4, 2}]], Bold, 8] &,
            {Keys[onTimeRates], Values[onTimeRates]}], "RadialCenter"], ChartStyle → "DarkRainbow",
         PlotLabel → "On-Time Rate Distribution by Airline", LabelStyle → Directive[Bold, FontSize → 12], ImageSize → 700]
```

Out[•]=


Airline On–Time Performance
On–Time Rate
0.8
0.6
0.4
0.2
0.0
Airlines
SkyWest Airlines Inc.
Republic Airline
United Air Lines Inc.
PSA Airlines Inc.
Endeavor Air Inc.
Spirit Air Lines
Envoy Air
Mesa Airlines Inc.
Southwest Airlines Co.
Delta Air Lines Inc.
Frontier Airlines Inc.
Allegiant Air
American Airlines Inc.
Hawaiian Airlines Inc.
Alaska Airlines Inc.
JetBlue Airways
Horizon Air
ExpressJet Airlines LLC d/b/a aha!

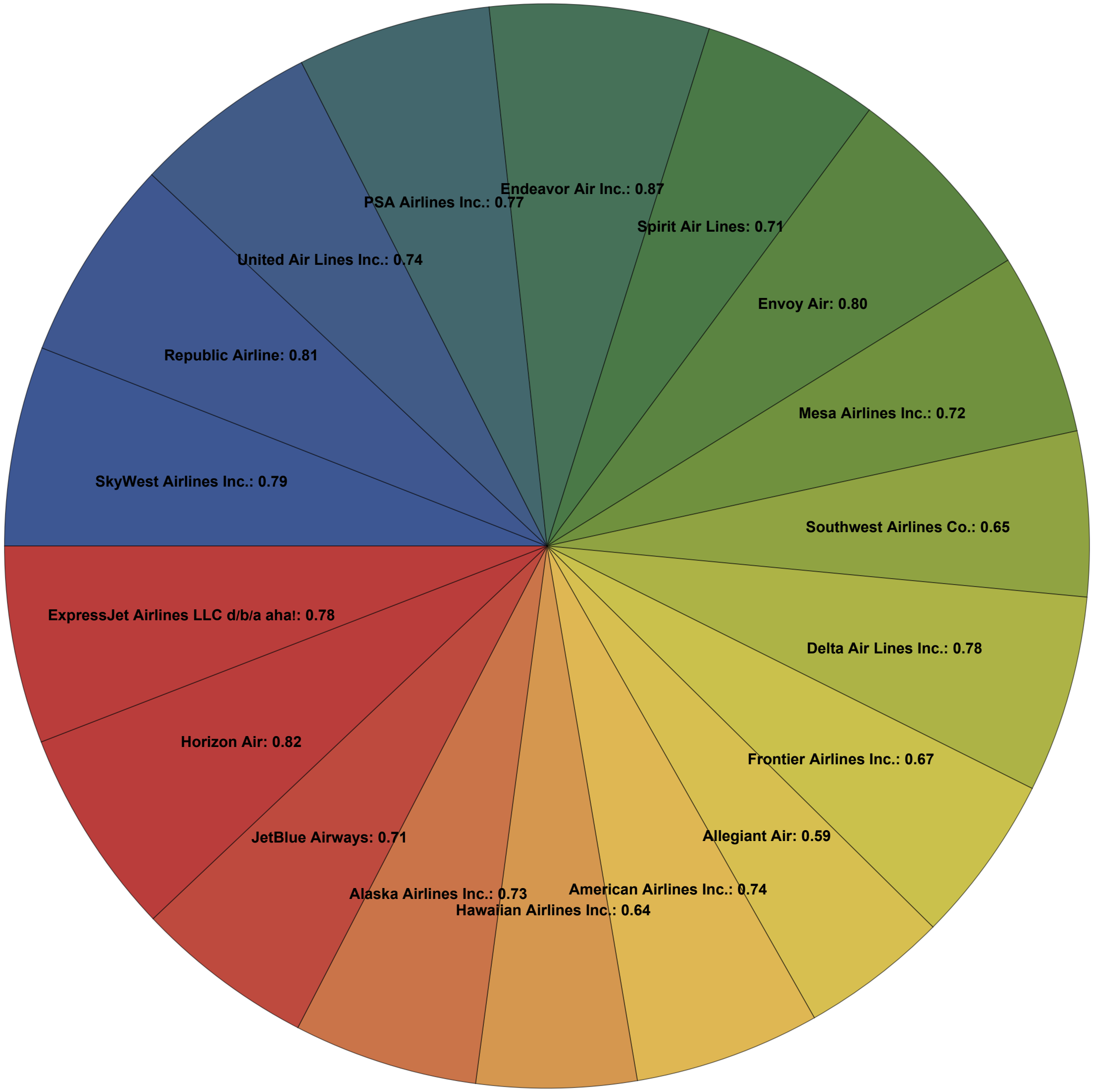
Out[•]=


On–Time Rates by Airline
1
2
3
4
5
6
7
8
9
10
11
12
13
1.0
Endeavor Air Inc.



Out[∘]=

On–Time Rate Distribution by Airline



Purpose: The bar chart, titled “**Airline On-Time Performance**,” visualizes the on-time performance rates for various airlines. It helps in identifying which airlines have the best and worst on-time performance, assisting travelers in making informed decisions.

Y-Axis:

- Label: “On-Time Rate”
- Represents: The on-time performance rate for each airline, ranging from 0.0 to 1.0.
- Higher bars: Indicate better on-time performance.

X-Axis:

- Label: “Airlines”
- Represents: Different airlines, including SkyWest Airlines Inc., Republic Airline, United Air Lines Inc., and others.

Insights:

1. Best On-Time Performance:
   - Endeavor Air, Horizon Air, and Republic Airline: Show the highest on-time performance rates.
2. Moderate On-Time Performance:
   - Envoy Air, SkyWest Airlines Inc., ExpressJet Airlines LLC d/b/a aha!:, Delta Air Lines Inc., PSA Airlines Inc., United Air Lines Inc., American Airlines Inc., Alaska Airlines Inc.: Have moderate on-time performance rates.
3. Lower On-Time Performance:
   - Allegiant Air, Hawaiian Airlines Inc., Southwest Airlines Co, Frontier Airlines Inc. : Show comparatively lower on-time performance rates.
4. Visual Representation:
   - Colored Dots: Represent different performance levels, making it easier to compare on-time rates across airlines.

Key Takeaways:

- Top Performers: Airlines like Endeavor Air, Horizon Air, and Republic Airline have excellent on-time performance, indicating efficient operations.
- Improvement Needed: Airlines with lower on-time performance, such as SAllegiant Air, Hawaiian Airlines Inc., Southwest Airlines Co, Frontier Airlines Inc., may need to focus on improving their schedules.
- Passenger Impact: Understanding on-time performance rates helps passengers choose more reliable airlines, ensuring better travel experiences.

Purpose: The scatter plot, titled “**On-Time Rates by Airline**”, visualizes the on-time performance rates for various airlines. It helps in identifying which airlines have the best and worst on-time performance, assisting travelers in making informed decisions.

Y-Axis:

- Label: “On-Time Rate”
- Represents: The on-time performance rate for each airline, ranging from 0.0 to 1.0.

- Higher points: Indicate better on-time performance.

X-Axis:

- Label: “Airlines (Numerical Indices)”
- Represents: Different airlines, each assigned a numerical index.

Insights:

1. Best On-Time Performance:

- Endeavor Air, Horizon Air, and Republic Airline: Show the highest on-time performance rates.

2. Moderate On-Time Performance:

- Envoy Air, SkyWest Airlines Inc., ExpressJet Airlines LLC d/b/a aha!:, Delta Air Lines Inc., PSA Airlines Inc., United Air Lines Inc., American Airlines Inc., Alaska Airlines Inc.: Have moderate on-time performance rates.

3. Lower On-Time Performance:

- Allegiant Air, Hawaiian Airlines Inc., Southwest Airlines Co, Frontier Airlines Inc.: Show comparatively lower on-time performance rates.

4. Visual Representation:

- Colored Dots: Represent different performance levels, making it easier to compare on-time rates across airlines.

Purpose: The pie chart, titled “**On-Time Rate Distribution by Airline**”, visualizes the distribution of on-time rates for various airlines. It provides a comparison of the on-time performance rates, helping identify which airlines have the best and worst performance.

Segments:

- PSA Airlines Inc.: 0.77
- Endeavor Air Inc.: 0.87
- Spirit Air Lines: 0.71
- Envoy Air: 0.80
- Mesa Airlines Inc.: 0.72
- Southwest Airlines Co.: 0.65
- Delta Air Lines Inc.: 0.78
- Frontier Airlines Inc.: 0.67
- Allegiant Air: 0.59
- Hawaiian Airlines Inc.: 0.64
- American Airlines Inc.: 0.74
- Alaska Airlines Inc.: 0.73

- JetBlue Airways: 0.71
- Horizon Air: 0.82
- ExpressJet Airlines LLC dba aha!: 0.78
- SkyWest Airlines Inc.: 0.79
- Republic Airline: 0.81
- United Air Lines Inc.: 0.74

Insights:

1. Best On-Time Rates:
   - Endeavor Air, Horizon Air, and Republic Airline: Have the highest on-time rates.
2. Moderate On-Time Rates:
   - Envoy Air, SkyWest Airlines Inc., ExpressJet Airlines LLC d/b/a aha!:, Delta Air Lines Inc., PSA Airlines Inc., United Air Lines Inc., American Airlines Inc., Alaska Airlines Inc.: Have moderate on-time rates.
3. Lower On-Time Rates:
   - Allegiant Air, Hawaiian Airlines Inc., Southwest Airlines Co, Frontier Airlines Inc.: Show lower on-time rates.

Key Takeaways:

- Top Performers: Airlines with higher on-time rates, such as Endeavor Air Inc. and Horizon Air, are likely operating efficiently.
- Improvement Needed: Airlines with lower on-time rates, such as Allegiant Air and Frontier Airlines Inc., may need to focus on improving their punctuality.
- Passenger Impact: Understanding on-time rates can help passengers select more reliable airlines.
- Visual Clarity: The pie chart effectively shows the distribution and relative performance of airlines, making it easier to understand at a glance.

c . Cancellations and Delays by Airline:

Purpose and Insights: The plots will display the frequency of cancellations and delays for top 10 airlines. This helps to identify which airlines experience more delays and cancellations.

Cancellations:

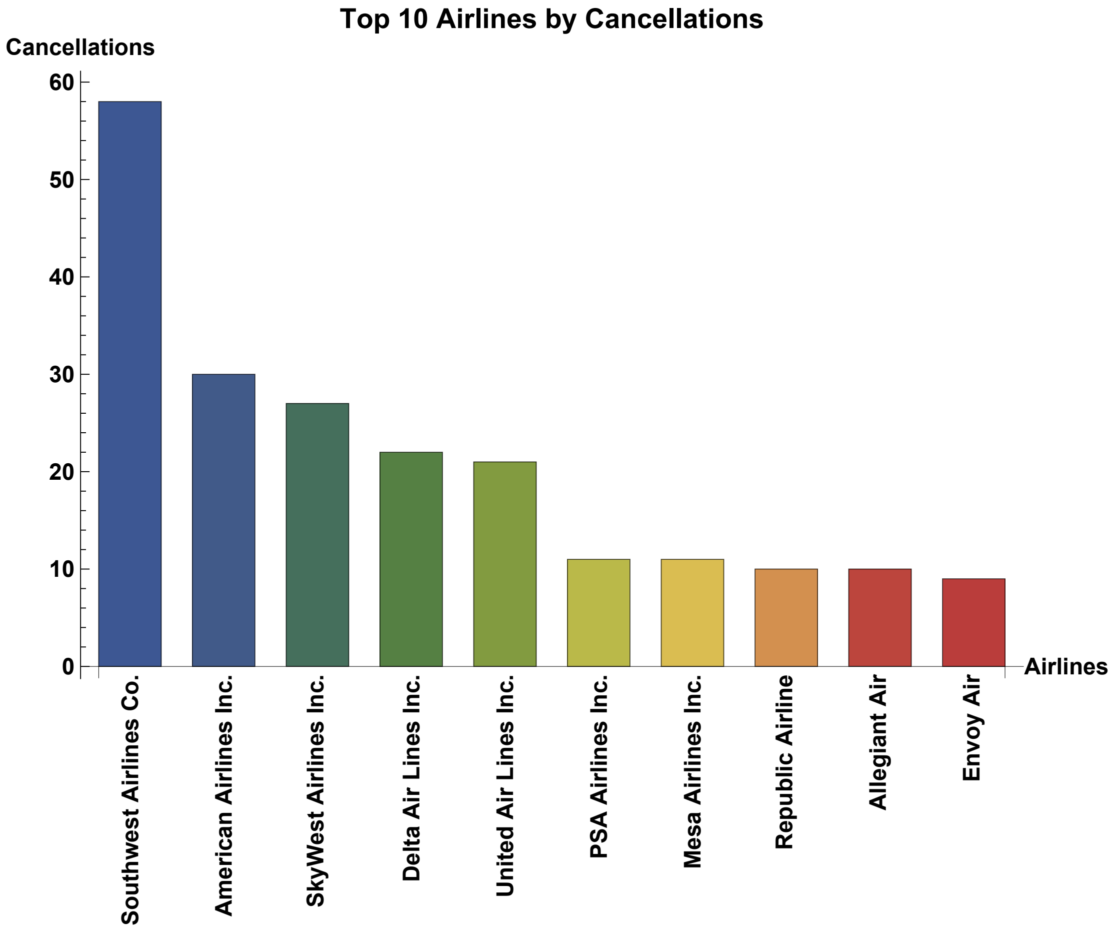
```
In[•]:= (*Group data by AIRLINE and sum CANCELLED*)
        airlineCancels = Normal[GroupBy[dataset, #AIRLINE &, (*Group by AIRLINE*)
            Total[Lookup[#, "CANCELLED", 0]] & (*Sum "CANCELLED" for each group*)]];

        (*Convert to a Table Format*)
        airlineTable = Dataset[KeyValueMap[<|"Airline" → #1, "Cancellations" → #2|> &, airlineCancels]];

        (*Display the Table*)
        airlineTable
```

Out[∘]=

Airline	Cancellations
SkyWest Airlines Inc.	27
Republic Airline	10
United Air Lines Inc.	21
PSA Airlines Inc.	11
Endeavor Air Inc.	6
Spirit Air Lines	7
Delta Air Lines Inc.	22
American Airlines Inc.	30
Envoy Air	9
Mesa Airlines Inc.	11
Frontier Airlines Inc.	3
Southwest Airlines Co.	58
Allegiant Air	10
Hawaiian Airlines Inc.	2
Alaska Airlines Inc.	5
JetBlue Airways	8
Horizon Air	1
ExpressJet Airlines LLC d/b/a aha!	1

```
In[◦]:= (*Filter and group data by AIRLINE and sum CANCELLED*)
airlineCanc = Normal[GroupBy[dataset, #AIRLINE &, Function[flights, Total[Lookup[flights, "CANCELLED", 0]]]]];

(*Ensure airlineCanc is not empty*)
If[Length[airlineCanc] > 0, (*Flatten the grouped data into a list of pairs {airline,cancellation count}*)
 flattenedAirlineCanc = KeyValueMap[Function[{airline, count}, {airline, count}], airlineCanc];
 (*Sort by cancellation count in descending order*)sortedAirlineCanc = SortBy[flattenedAirlineCanc, Last, Greater];
 (*Get top 10 airlines with highest cancellations*)top10Airlines = Take[sortedAirlineCanc, 10];

 (*Create Bar Chart for top 10 airlines with rotated x-axis labels*)
 BarChart[Last /@ top10Airlines, ChartLabels → Placed[Rotate[#, 90 Degree] & /@ First /@ top10Airlines, Below],
  ChartStyle → "DarkRainbow", AxesLabel → {"Airlines", "Cancellations"}, BarSpacing → 0.5,
  PlotLabel → "Top 10 Airlines by Cancellations", LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large],
 (*If no data available*)Print["No data available for cancellations by airline."]]
```

Out[ ]=



Purpose:

The bar chart, titled "**Top 10 Airlines by Cancellations**", visualizes the total number of flight cancellations for the top 10 airlines. This analysis highlights which airlines face the highest number of cancellations, offering insights into their operational challenges.

Y-Axis:

- Label: "Cancellations"
- Represents the total number of cancellations for each airline.
- Higher bars indicate more frequent cancellations.

X-Axis:

- Label: “Airlines”
- Lists the top 10 airlines with the highest cancellation counts, such as Southwest Airlines Co., American Airlines Inc., and others.

Insights:

1. Most Cancellations:
   - Southwest Airlines Co. leads with the highest number of cancellations, significantly outpacing other airlines.
2. Moderate Cancellations:
   - American Airlines Inc. and SkyWest Airlines Inc. also have high cancellation counts, although they trail Southwest by a wide margin.
3. Lower but Notable Cancellations:
   - Airlines like Republic Airline, Allegiant Air, and Envoy Air have lower cancellation counts but still rank within the top 10.
4. Operational Considerations:
   - High cancellations could result from operational inefficiencies, resource shortages, or weather-related disruptions. Airlines with higher cancellation rates may need to evaluate these factors.

Key Takeaways:

- High Impact: Airlines like Southwest may need to prioritize improving scheduling, fleet management, or contingency planning.
- Customer Experience: Frequent cancellations can negatively impact customer trust, making mitigation strategies crucial for maintaining loyalty.
- Industry Insights: Comparing cancellation rates across airlines helps identify industry-wide trends or airline-specific challenges.

Delays:

◆ Arrival Delays

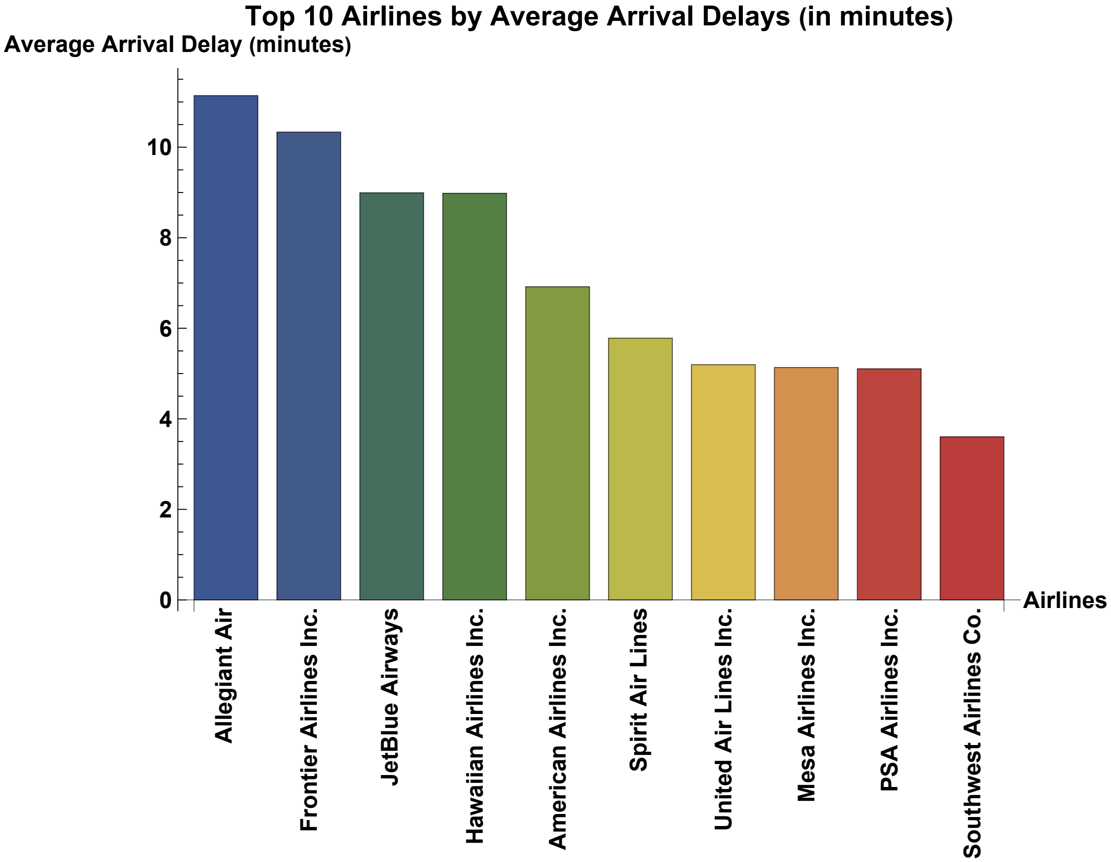
```
In[◦]:= (*Group data by AIRLINE and calculate the average ARR_DELAY for each airline*)
avgArrDelays1 = Normal[GroupBy[dataset, #AIRLINE &, (*Group by AIRLINE*)
    Function[flights, N[Mean[Lookup[flights, "ARR_DELAY", 0]]]] (*Use N to get decimal values*)]];

(*Sort the average delays in descending order*)
sortedArrDelays1 = SortBy[avgArrDelays1, -# &]; (*Sort by negative of the average ARR_DELAY for descending order*)

(*Convert to a more readable format and show in a table*)
sortedArrDelaysTable1 = Dataset[AssociationThread[Keys[sortedArrDelays1], Values[sortedArrDelays1]]]
```

Out[◦]=

Allegiant Air	11.1398
Frontier Airlines Inc.	10.3348
JetBlue Airways	8.99185
Hawaiian Airlines Inc.	8.98131
American Airlines Inc.	6.91754
Spirit Air Lines	5.78146
United Air Lines Inc.	5.19599
Mesa Airlines Inc.	5.13364
PSA Airlines Inc.	5.1044
Southwest Airlines Co.	3.60286
ExpressJet Airlines LLC d/b/a aha!	3.43137
Envoy Air	2.49633
SkyWest Airlines Inc.	2.28369
Republic Airline	2.06356
Alaska Airlines Inc.	2.0
Horizon Air	1.80882
Delta Air Lines Inc.	1.05629
Endeavor Air Inc.	−4.76045

```
In[∘]:= (*Plot average arrival delays as a bar chart for top 10 airlines*)BarChart[Values[Take[sortedArrDelays1, 10]],
         ChartLabels → Placed[Rotate[#, 90 Degree] & /@ Keys[sortedArrDelays1], Below], BarSpacing → 0.3,
         ChartStyle → "DarkRainbow", AxesLabel → {"Airlines", "Average Arrival Delay (minutes)"},
         PlotLabel → "Top 10 Airlines by Average Arrival Delays (in minutes)",
         LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]
```

Out[∘]=



Purpose:

The bar chart, titled "**Top 10 Airlines by Average Arrival Delays (in minutes)**", illustrates the average arrival delay (in minutes) for the top 10 airlines. This analysis highlights which airlines experience the most delays in arrivals, providing insights into operational performance.

Y-Axis:

- Label: “Average Arrival Delay (in minutes)”
- Represents the average delay in minutes for flights arriving for each airline.
- Higher bars indicate longer average delays.

X-Axis:

- Label: “Airlines”
- Represents the top 10 airlines with the highest average arrival delays, including Allegiant Air, Frontier Airlines Inc., and others.

Insights:

1. Highest Average Arrival Delays:

- Allegiant Air experiences the highest average arrival delay, exceeding 10 minutes.
- Frontier Airlines Inc. and JetBlue Airways follow closely with similar delays.

2. Moderate Arrival Delays:

- Airlines like Hawaiian Airlines Inc., American Airlines Inc., and Spirit Airlines have average arrival delays between 6 to 8 minutes.

3. Lowest Average Arrival Delays in Top 10:

- Southwest Airlines Co. has the lowest average delay among the top 10, around 5 minutes, suggesting relatively better arrival performance.

4. Operational Observations:

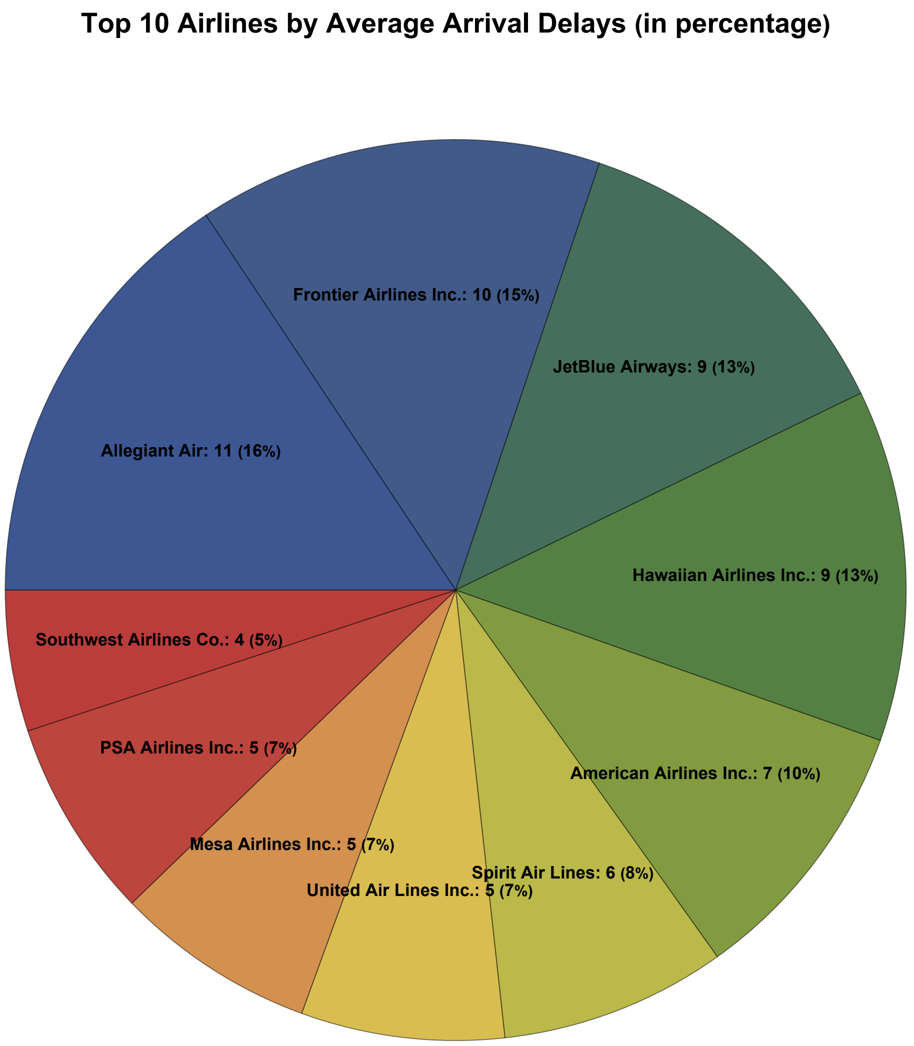
- Airlines with higher delays might face systemic issues, such as scheduling inefficiencies, resource constraints, or external factors like weather.

Key Takeaways:

- High-Delay Airlines: Airlines like Allegiant Air and Frontier Airlines may need to focus on operational improvements to reduce delays.
- Comparative Advantage: Southwest Airlines demonstrates better arrival punctuality, which could enhance its reputation and customer satisfaction.
- Customer Perspective: Travelers prioritizing on-time arrivals can use this data to select airlines with lower average delays.

```
In[•]:= Module[{data, total, percentages, labels}, data = Take[Values[sortedArrDelays1], 10];
          total = Total[data];
          percentages = (100 * # / total) & /@ data;
          labels =
           MapThread[Style[ToString[#1] <> ": " <> ToString[Round[#2, 1]] <> " (" <> ToString[Round[#3, 1]] <> "%)", Bold, 9] &,
             {Keys[Take[sortedArrDelays1, 10]], data, percentages}];
          PieChart[percentages, ChartLabels → Placed[labels, "RadialCenter"],
           ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airlines by Average Arrival Delays (in percentage)",
           ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]]
```

Out[ ]=



Purpose:

The “**Top 10 Airlines by Average Arrival Delays (in percentages)**” pie chart visualizes the average arrival delays for the top 10 airlines. It helps in identifying which airlines have the highest and lowest average arrival delays, aiding travelers in making informed decisions.

Segments:

- Allegiant Air: 11 minutes (16%)
- Frontier Airlines Inc.: 10 minutes (15%)
- JetBlue Airways: 9 minutes (13%)
- Hawaiian Airlines Inc.: 9 minutes (13%)
- American Airlines Inc.: 7 minutes (10%)
- Spirit Air Lines: 6 minutes (8%)
- PSA Airlines Inc.: 5 minutes (7%)
- Mesa Airlines Inc.: 5 minutes (7%)
- United Air Lines Inc.: 5 minutes (7%)
- Southwest Airlines Co.: 4 minutes (5%)

Insights:

1. Highest Average Arrival Delays:
   - Allegiant Air: Leads with the highest average arrival delay of 11 minutes, accounting for 16%.
   - Frontier Airlines Inc. and JetBlue Airways: Follow closely with 15% and 13% minutes respectively.
2. Lowest Average Arrival Delays:
   - Southwest Airlines Co.: Has the lowest average arrival delay of 4 minutes, accounting for 5%.
   - Other Airlines: Such as PSA Airlines Inc., Mesa Airlines Inc., and United Air Lines Inc. have relatively lower delays compared to the leaders.

◆ Departure Delays

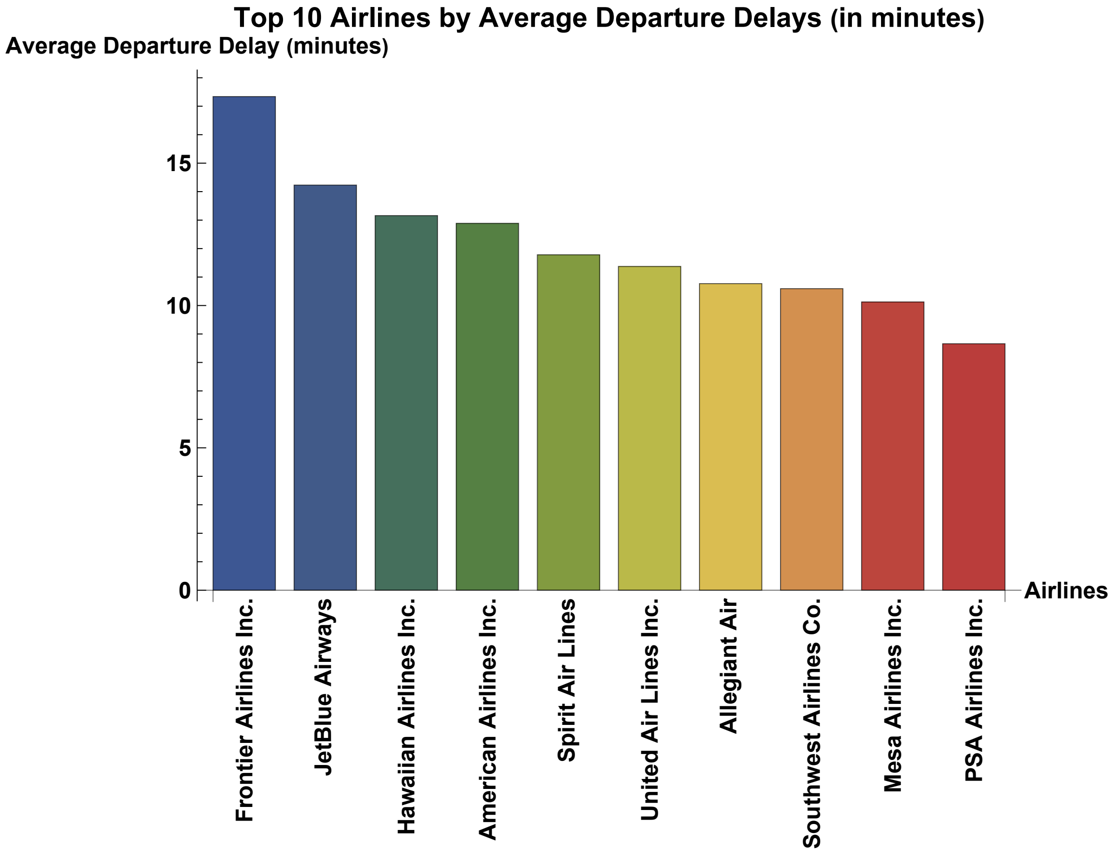
```
In[∘]:= (*Group data by AIRLINE and calculate the average DEP_DELAY for each airline*)avgDepDelays1 = Normal[
          GroupBy[dataset, #AIRLINE &, (*Group by airline*)Function[flights, N[Mean[Lookup[flights, "DEP_DELAY", 0]]]]]];

        (*Sort the average delays in descending order*)
        sortedDepDelays1 = SortBy[avgDepDelays1, -# &]; (*Sort by negative of the average ARR_DELAY for descending order*)

        (*Convert to a more readable format and show in a table*)
        sortedDepDelaysTable1 = Dataset[AssociationThread[Keys[sortedDepDelays1], Values[sortedDepDelays1]]]
Out[∘]=
```

Frontier Airlines Inc.	17.3391
JetBlue Airways	14.231
Hawaiian Airlines Inc.	13.1589
American Airlines Inc.	12.886
Spirit Air Lines	11.7815
United Air Lines Inc.	11.3719
Allegiant Air	10.7688
Southwest Airlines Co.	10.5916
Mesa Airlines Inc.	10.1244
PSA Airlines Inc.	8.65659
SkyWest Airlines Inc.	8.56028
Delta Air Lines Inc.	8.42032
ExpressJet Airlines LLC d/b/a aha!	7.47059
Envoy Air	6.33007
Republic Airline	5.99788
Alaska Airlines Inc.	5.08
Endeavor Air Inc.	3.3649
Horizon Air	0.882353

In[∘]:=

```
(*Plot average departure delays as a bar chart for top 10 airlines*)BarChart[Values[Take[sortedDepDelays1, 10]],
 ChartLabels → Placed[Rotate[#, 90 Degree] & /@ Keys[sortedDepDelays1], Below], BarSpacing → 0.3,
 ChartStyle → "DarkRainbow", AxesLabel → {"Airlines", "Average Departure Delay (minutes)"},
 PlotLabel → "Top 10 Airlines by Average Departure Delays (in minutes)",
 LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]
```

Out[∘]=



Purpose:

The bar chart, titled “**Top 10 Airlines by Average Departure Delays (in minutes)**”, highlights the average departure delay (in minutes) for the top 10 airlines with the most significant delays. It provides insights into which airlines experience longer departure delays and where improvements might be needed.

Y-Axis:

- Label: “Average Departure Delay (in minutes)”
- Represents the average delay in minutes for flights departing from each airline.
- Higher bars indicate longer average departure delays.

X-Axis:

- Label: “Airlines”
- Represents the top 10 airlines with the highest average departure delays, including Frontier Airlines Inc., JetBlue Airways, and others.

Insights:

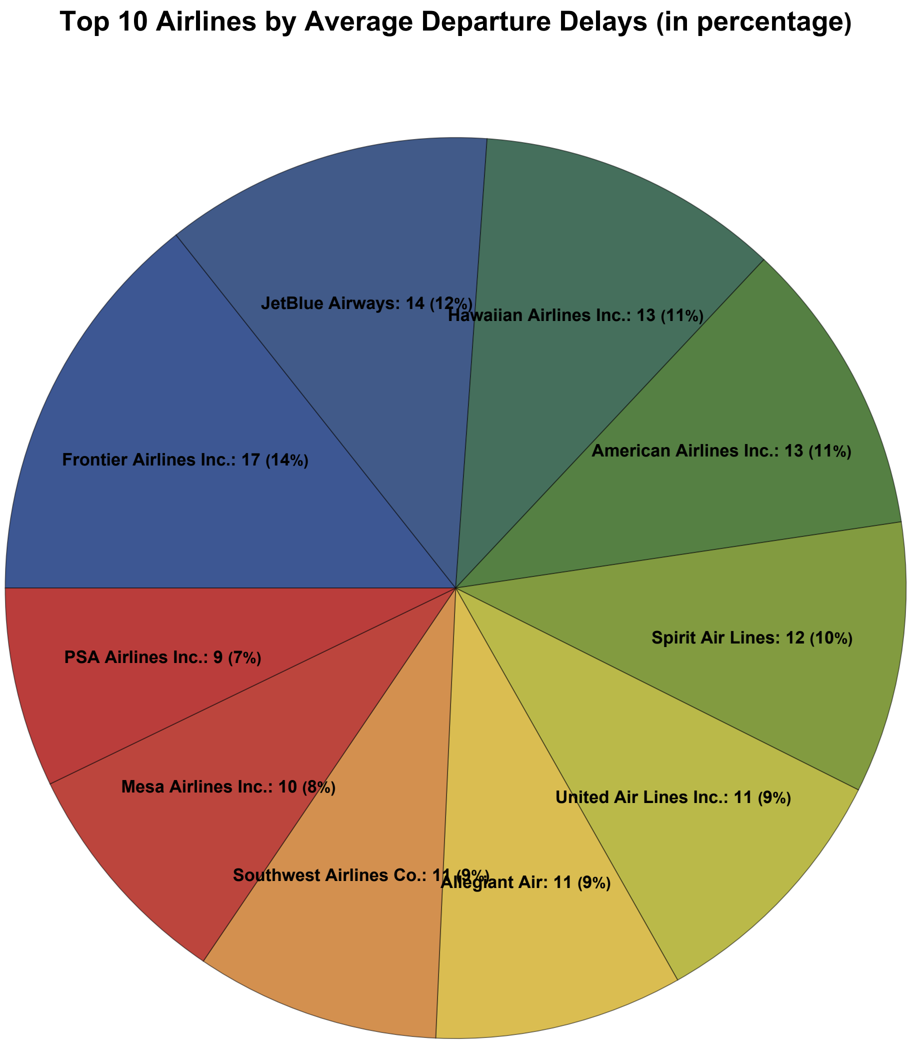
1. Highest Average Departure Delays:
   - Frontier Airlines Inc. has the highest average departure delay, exceeding 15 minutes.
   - JetBlue Airways and Hawaiian Airlines Inc. follow closely, with delays around 13-14 minutes.
2. Moderate Departure Delays:
   - Airlines like United Airlines Inc., Spirit Airlines, and Allegiant Air have average delays between 10-12 minutes.
3. Lowest Average Departure Delays in Top 10:
   - PSA Airlines Inc. has the lowest departure delays among the top 10, averaging around 10 minutes.
4. Operational Challenges:
   - Airlines with high departure delays may face issues such as gate turnaround inefficiencies, resource constraints, or external factors like weather conditions.

Key Takeaways:

- High-Delay Airlines: Airlines like Frontier Airlines and JetBlue Airways may need to focus on reducing operational delays to improve performance.
- Better Performers: PSA Airlines and Mesa Airlines have comparatively lower delays and might already employ more effective operational practices.
- Customer Implications: Frequent delays could impact customer satisfaction, making this analysis valuable for passengers prioritizing punctuality.

```
In[◦]:= Module[{data, total, percentages, labels}, data = Take[Values[sortedDepDelays1], 10];
          total = Total[data];
          percentages = (100 * # / total) & /@ data;
          labels =
           MapThread[Style[ToString[#1] <> ": " <> ToString[Round[#2, 1]] <> " (" <> ToString[Round[#3, 1]] <> "%)", Bold, 9] &,
            {Keys[Take[sortedDepDelays1, 10]], data, percentages}];
          PieChart[percentages, ChartLabels → Placed[labels, "RadialCenter"],
           ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airlines by Average Departure Delays (in percentage)",
           ImageSize → Large, LabelStyle → Directive[FontSize → 12, Bold]]]
```

Out[•]=



Top 10 Airlines by Average Departure Delays (in percentage)

Purpose:

The "**Top 10 Airlines by Average Departure Delays (in percentage)**"pie chart visualizes the average departure delays for the top 10 airlines. It aids in identifying which airlines have the highest and lowest average departure delays, assisting travelers in making informed decisions.

Segments:

- Frontier Airlines Inc.: 17 minutes (14%)
- JetBlue Airways: 14 minutes (12%)
- Hawaiian Airlines Inc.: 13 minutes (11%)
- American Airlines Inc.: 13 minutes (11%)
- Spirit Air Lines: 12 minutes (10%)
- United Air Lines Inc.: 11 minutes (9%)
- Allegiant Air: 11 minutes (9%)
- Southwest Airlines Co.: 11 minutes (9%)
- Mesa Airlines Inc.: 10 minutes (8%)
- PSA Airlines Inc.: 9 minutes (7%)

Insights:

1. Highest Average Departure Delays:
   - Frontier Airlines Inc. contributes the largest share, accounting for 14%, making it a standout in terms of delays.
   - JetBlue Airways and Hawaiian Airlines Inc.: Follow closely with 14 (12%) and 13 minutes (11%), respectively.
2. Moderate and Low Delays:
   - PSA Airlines Inc.: Has the lowest average departure delay of 9 minutes, with a share of 7%.
   - Other Airlines: Such as Mesa Airlines Inc. , Southwest Airlines Co., and Allegiant Air show moderate delays, each contributing around 8%, compared to the leaders.

Total Delays

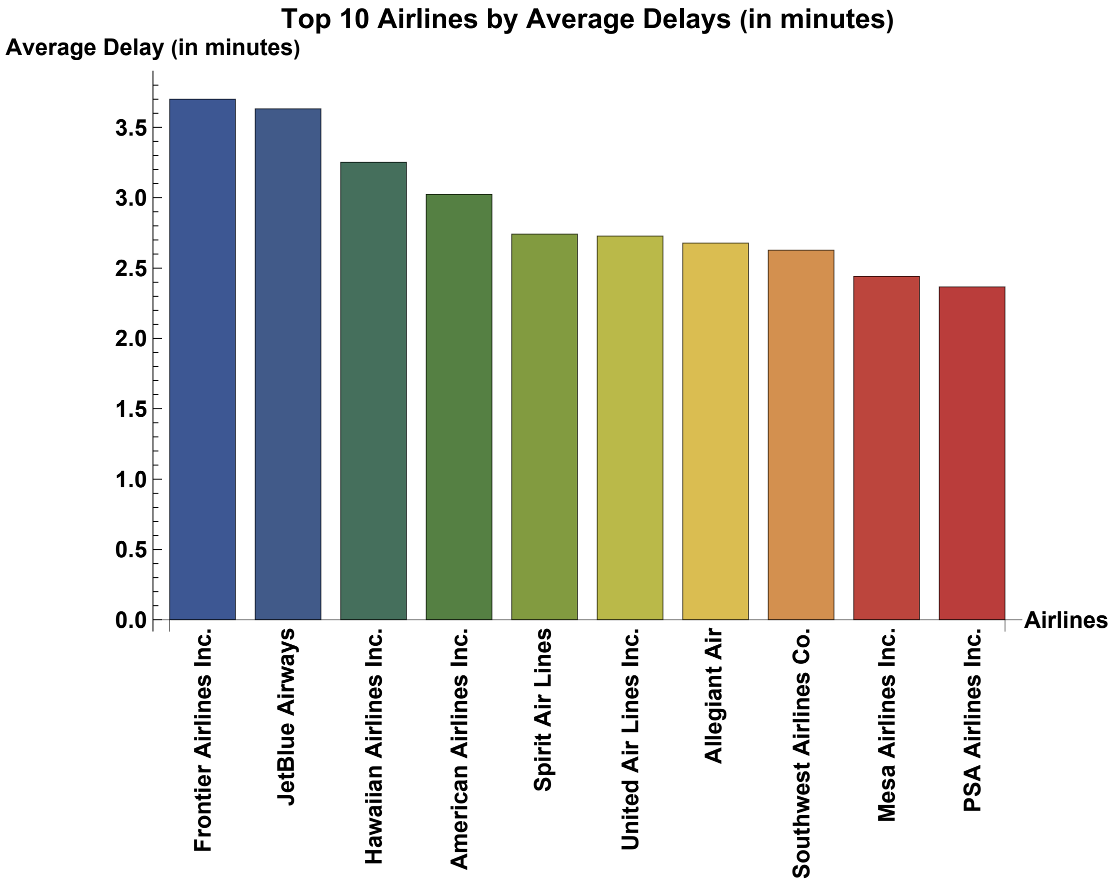
```
In[∘]:= (*Average Delays=Mean of Arrival and Departure Delays*)
        (*Group by airline and calculate the average delay for each airline*)
        airlineDelays1 = GroupBy[dataset, #AIRLINE &, Function[flights, Mean[Flatten[Lookup[flights, {"DELAY_DUE_CARRIER",
                 "DELAY_DUE_WEATHER", "DELAY_DUE_NAS", "DELAY_DUE_SECURITY", "DELAY_DUE_LATE_AIRCRAFT"}, 0]]]]];

        (*Convert grouped data into a single flat list with "Airline" and "Average Delays" keys*)
        flattenedAirlineDelays =
          KeyValueMap[Function[{airline, delays}, <|"Airline" → airline, "Average Delays" → N[delays]|>], airlineDelays1];
        (*Apply N to get decimal format*)

        (*Sort the average delays in descending order*)
        sortedDelays1 = SortBy[flattenedAirlineDelays, -#["Average Delays"] &];

        (*Convert to a more readable format and show in a table*)
        sortedDepTable1 = Dataset[sortedDelays1];

        (*Display the table*)
        sortedDepTable1
```

Out[∘]=

Airline	Average Delays
JetBlue Airways	3.7
Frontier Airlines Inc.	3.63176
Allegiant Air	3.25161
American Airlines Inc.	3.0228
PSA Airlines Inc.	2.74176
Spirit Air Lines	2.72781
United Air Lines Inc.	2.67792
Mesa Airlines Inc.	2.62765
ExpressJet Airlines LLC d/b/a aha!	2.43922
Hawaiian Airlines Inc.	2.36636
Republic Airline	2.28602
SkyWest Airlines Inc.	2.26507
Envoy Air	2.11002
Southwest Airlines Co.	1.95926
Delta Air Lines Inc.	1.95673
Alaska Airlines Inc.	1.80185
Endeavor Air Inc.	1.45404
Horizon Air	1.08824

```
In[•]:= (*Extract the airline names and average delays from sortedDelays1 dataset*)
        top10Airlines3 = Take[sortedDelays1, 10]; (*Take top 10 entries*)
        airlineNames3 = top10Airlines3[All, "Airline"]; (*Extract airline names*)
        delays1 = top10Airlines3[All, "Average Delays"]; (*Extract average delays*)

        (*Convert to lists for BarChart*)
        airlineNames3List = Normal[airlineNames3];
        delaysList1 = Normal[delays1];

        (*Plot the average total delays as a bar chart for top 10 airlines*)
        BarChart[delaysList1, ChartLabels → Placed[Rotate[#, 90 Degree] & /@ Keys[sortedDepDelays1], Below],
         BarSpacing → 0.3, ChartStyle → "DarkRainbow", (*Use a predefined color scheme*)
         AxesLabel → {"Airlines", "Average Delay (in minutes)"}, PlotLabel → "Top 10 Airlines by Average Delays (in minutes)",
         LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]
```

Out[∘]=



Purpose: The bar chart, titled “**Top 10 Airlines by Average Delays (in minutes)”**, highlights the average delays in minutes for the top 10 airlines with the most significant delays. It provides insights into which airlines experience longer average delays and where improvements might be needed.

Y-Axis:

- Label: “Average Delay (in minutes)”
- Represents: The average delay in minutes for flights operated by each airline.
- Higher bars: Indicate longer average delays.

X-Axis:

- Label: “Airlines”

- Represents: The top 10 airlines with the highest average delays, including Frontier Airlines Inc., JetBlue Airways, and others.

Insights:

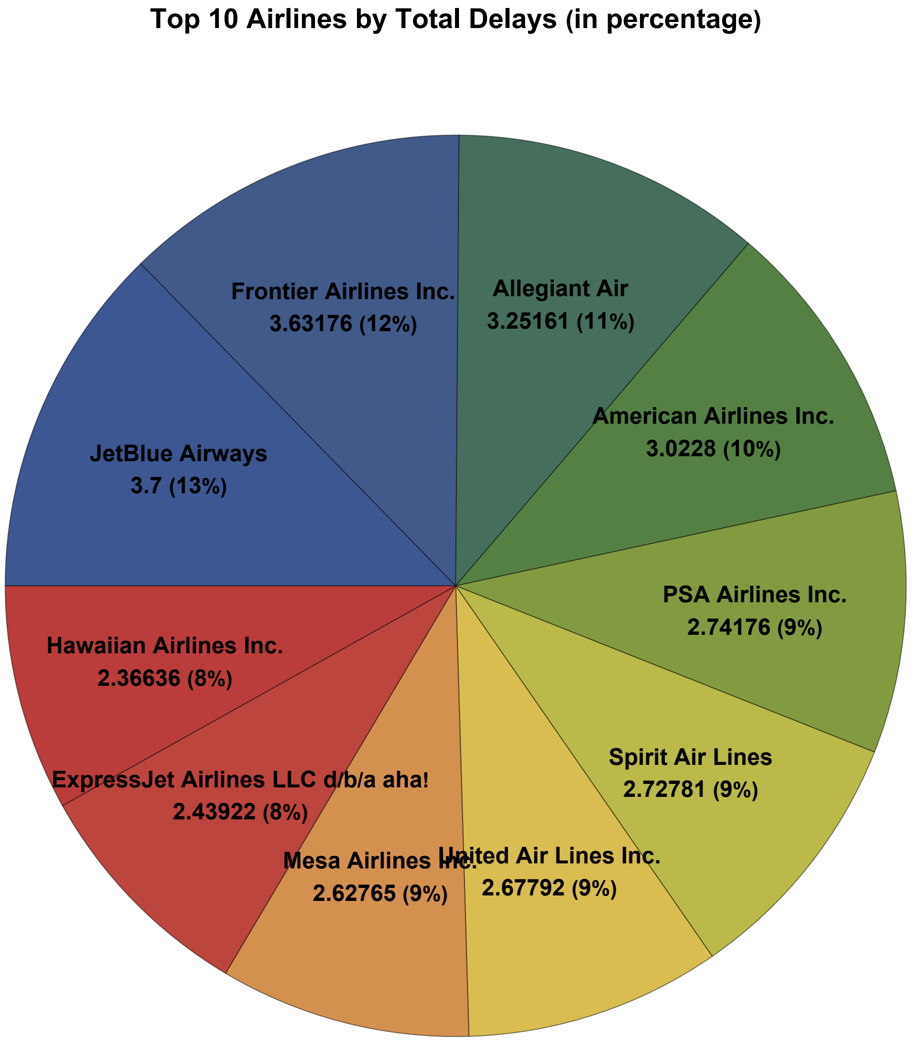
1. Highest Average Delays:
   - Frontier Airlines Inc.: Has the highest average delay, approximately 3.5 minutes.
   - JetBlue Airways and Hawaiian Airlines Inc.: Follow closely, with delays around 3.4 and 3.2 minutes, respectively.
2. Moderate Delays:
   - American Airlines Inc., Spirit Air Lines, and United Air Lines Inc.: Have average delays ranging from approximately 2.8 to 3.1 minutes.
3. Lowest Average Delays in Top 10:
   - PSA Airlines Inc.: Has the lowest average delay among the top 10, around 2.5 minutes.
4. Operational Challenges:
   - Airlines with high average delays may face issues such as gate turnaround inefficiencies, resource constraints, or external factors like weather conditions.

Key Takeaways:

- High-Delay Airlines: Airlines like Frontier Airlines Inc. and JetBlue Airways may need to focus on reducing operational delays to improve performance.
- Better Performers: PSA Airlines Inc. and Mesa Airlines Inc. have comparatively lower delays and might already employ more effective operational practices.
- Customer Implications: Frequent delays could impact customer satisfaction, making this analysis valuable for passengers prioritizing punctuality.

```
In[ ]:= Module[{data, total, percentages, labels}, data = delaysList1⟦;; 10⟧;
         total = Total[data];
         percentages = (100 * # / total) & /@ data;
         labels = MapThread[Function[{code, delay, percentage}, StringJoin[code, "\n", ToString[delay],
             " (", ToString[Round[percentage, 1]], "%)"]], {airlineNames3List⟦;; 10⟧, data, percentages}];
         PieChart[percentages, ChartLabels → Placed[labels, "RadialCenter"],
          ChartStyle → "DarkRainbow", PlotLabel → "Top 10 Airlines by Total Delays (in percentage)",
          LabelStyle → Directive[FontSize → 12, Bold], ImageSize → Large]]
```

Out[•]=



Purpose: The pie chart, titled "**Top 10 Airlines by Total Delays (in percentage)**", highlights the total delay times (in minutes) proportion for the top 10 airlines with the most significant delays. It provides insights into which airlines experience longer delays and where improvements might be needed.

Segments:

- JetBlue Airways: 3.7 minutes (13%)
- Frontier Airlines Inc.: 3.63176 minutes (12%)
- Allegiant Air: 3.25161 minutes (11%)
- American Airlines Inc.: 3.0228 minutes (10%)
- PSA Airlines Inc.: 2.74176 minutes (9%)
- Spirit Air Lines: 2.72781 minutes (9%)
- United Air Lines Inc.: 2.67792 minutes (9%)
- Mesa Airlines Inc.: 2.62765 minutes (9%)
- ExpressJet Airlines LLC d/b/a aha!: 2.43922 minutes (8%)
- Hawaiian Airlines Inc.: 2.36636 minutes (8%)

Insights:

1. Highest Total Delays:
   - JetBlue Airways: Leads with the highest total delay time of 3.7 minutes, accounting for 13%.
   - Frontier Airlines Inc. and Allegiant Air follow closely with 12% and 11%, respectively.
2. Moderate Total Delays:
   - Airlines such as American Airlines Inc., PSA Airlines Inc., and Spirit Air Lines contribute 9–10%, indicating moderate levels of delays.
3. Lowest Total Delays in Top 10:
   - Hawaiian Airlines Inc. has the smallest share of delays, contributing 8% (around 2.4 minutes).
4. Operational Challenges:
   - Airlines with high total delays: May face issues such as gate turnaround inefficiencies, resource constraints, or external factors like weather conditions.

Develop Business Logic and Predictive Models

1. Random Forest Classification Model : For Predicting Likelihood of Delay and Cancellation by Airport and Airlines

a. Modelling

```
In[▫]:= numericDataset = Map[Association, Normal[dataset]];

        categorizeColumn[columnData_] := AssociationThread[Union[columnData] → Range[Length[Union[columnData]]]];

        (*Map AIRLINE and ORIGIN to numeric codes*)
        airlineCodes = categorizeColumn[numericDataset⟦All, "AIRLINE"⟧];
        originCodes = categorizeColumn[numericDataset⟦All, "ORIGIN"⟧];

        (*Replace categorical values with numeric codes for AIRLINE and ORIGIN*)
        numericDataset =
          Map[Function[assoc, assoc~Join~<|"AIRLINE" → Lookup[airlineCodes, assoc["AIRLINE"], Missing["NotMapped"]],
              "ORIGIN" → Lookup[originCodes, assoc["ORIGIN"], Missing["NotMapped"]]|>], numericDataset];

        (*Add new columns for DELAYED and CANCELLED_LABEL*)
        numericDataset =
          Map[Function[assoc, assoc~Join~<|"DELAYED" → If[assoc["ARR_DELAY"] > 0 || assoc["DEP_DELAY"] > 0, 1, 0],
              "CANCELLED_LABEL" → If[assoc["CANCELLED"] > 0, 1, 0]|>], numericDataset];

        (*Check the first few rows of numericDataset with new columns
         Print["First few rows of numericDataset with DELAYED and CANCELLED_LABEL:"];
        Take[numericDataset,2];*)

        (*Prepare the target vectors for prediction*)
        delayTargetList = numericDataset⟦All, "DELAYED"⟧;
        cancelTargetList = numericDataset⟦All, "CANCELLED_LABEL"⟧;

        (*Check the first few entries of the target variables
         Print["First few entries of delayTargetList: ",Take[delayTargetList,2]]
        Print["First few entries of cancelTargetList: ",Take[cancelTargetList,2]]*)

        (*Prepare the features for training,i.e.,the columns used for prediction*)
        features = {"AIRLINE", "ORIGIN", "DEST", "CRS_DEP_TIME", "DEP_TIME", "CRS_ARR_TIME", "ARR_TIME",
           "CRS_ELAPSED_TIME", "DISTANCE", "DELAY_DUE_WEATHER", "DELAY_DUE_CARRIER", "DELAY_DUE_NAS",
   "DELAY_DUE_SECURITY", "ELAPSED_TIME", "AIR_TIME", "WHEELS_ON", "WHEELS_OFF", "TAXI_IN", "TAXI_OUT"};

featureDataList = Map[Function[assoc, KeyTake[assoc, features]], numericDataset];

(*Check the first few rows of featureDataList
 Print["First few rows of featureDataList:"];
Take[featureDataList,2];*)

(*Clean the data by removing any missing values or empty rows*)
validPositions = Select[Range[Length[featureDataList]], FreeQ[featureDataList⟦#⟧, _?MissingQ] &];

(*Ensure cleaned data and targets are aligned and not empty*)
cleanedFeatureDataList = featureDataList⟦validPositions⟧;
cleanedDelayTargetList = delayTargetList⟦validPositions⟧;
cleanedCancelTargetList = cancelTargetList⟦validPositions⟧;

(*Check the cleaned data
 Print["First few rows of cleaned featureDataList:"];
Take[cleanedFeatureDataList,2];
Print["First few entries of cleanedDelayTargetList: ",Take[cleanedDelayTargetList,5]];
Print["First few entries of cleanedCancelTargetList: ",Take[cleanedCancelTargetList,5]];*)

(*Outputs for Target Lists*)
(* Print["First few entries of cleanedDelayTargetList: ",Take[cleanedDelayTargetList,5]];
Print["First few entries of cleanedCancelTargetList: ",Take[cleanedCancelTargetList,5]];*)

(*Train Random Forest for Delay Prediction*)
delayRandomForest = Predict[cleanedFeatureDataList → cleanedDelayTargetList, Method → "RandomForest"];
Print["Trained Delay Random Forest Model: ", delayRandomForest];

(*Train Random Forest for Cancellation Prediction*)
cancelRandomForest = Predict[cleanedFeatureDataList → cleanedCancelTargetList, Method → "RandomForest"];
Print["Trained Cancellation Random Forest Model: ", cancelRandomForest];

(*Evaluate Random Forest Models and Convert Probabilities to Binary*)
delayPredictions = delayRandomForest[cleanedFeatureDataList];
```

```
cancelPredictions = cancelRandomForest[cleanedFeatureDataList];

(*Convert continuous probability outputs to binary predictions based on a threshold of 0.5*)
delayBinaryPredictions = Map[If[# ≥ 0.5, 1, 0] &, delayPredictions];
cancelBinaryPredictions = Map[If[# ≥ 0.5, 1, 0] &, cancelPredictions];

(*Check the first few binary predictions
Print["First few binary delay predictions: ",Take[delayBinaryPredictions,5]];
Print["First few binary cancel predictions: ",Take[cancelBinaryPredictions,5]];*)

(*Calculate Accuracy for Delay Prediction*)
delayTruePositives = Count[Transpose[{delayBinaryPredictions, cleanedDelayTargetList}], {1, 1}];
delayTrueNegatives = Count[Transpose[{delayBinaryPredictions, cleanedDelayTargetList}], {0, 0}];
delayCorrectPredictions = delayTruePositives + delayTrueNegatives;

Print["Number of Correct Delay Predictions: ", delayCorrectPredictions];

(*Calculate Accuracy for Cancellation Prediction*)
cancelTruePositives = Count[Transpose[{cancelBinaryPredictions, cleanedCancelTargetList}], {1, 1}];
cancelTrueNegatives = Count[Transpose[{cancelBinaryPredictions, cleanedCancelTargetList}], {0, 0}];
cancelCorrectPredictions = cancelTruePositives + cancelTrueNegatives;

Print["Number of Correct Cancellation Predictions: ", cancelCorrectPredictions];

(*Calculate total number of predictions for accuracy*)
totalDelayPredictions = Length[cleanedDelayTargetList];
totalCancelPredictions = Length[cleanedCancelTargetList];

(*Calculate and print accuracy as a percentage*)
delayAccuracyPercentage = N[(delayCorrectPredictions / totalDelayPredictions) * 100, 2];
cancelAccuracyPercentage = N[(cancelCorrectPredictions / totalCancelPredictions) * 100, 2];

Print["Delay Prediction Accuracy (Percentage): ", NumberForm[delayAccuracyPercentage, {5, 2}], "%"];
Print["Cancellation Prediction Accuracy (Percentage): ", NumberForm[cancelAccuracyPercentage, {5, 2}], "%"];

(*Precision,Recall,and F1 for Delay Prediction*)
```

```
delayTruePositives = Count[Transpose[{delayBinaryPredictions, cleanedDelayTargetList}], {1, 1}];
delayFalsePositives = Count[Transpose[{delayBinaryPredictions, cleanedDelayTargetList}], {1, 0}];
delayFalseNegatives = Count[Transpose[{delayBinaryPredictions, cleanedDelayTargetList}], {0, 1}];
delayPrecision = N[delayTruePositives / (delayTruePositives + delayFalsePositives), 2];
delayRecall = N[delayTruePositives / (delayTruePositives + delayFalseNegatives), 2];
delayF1 = N[2 * (delayPrecision * delayRecall) / (delayPrecision + delayRecall), 2];

Print["Delay Precision: ", NumberForm[delayPrecision, {4, 2}]];
Print["Delay Recall: ", NumberForm[delayRecall, {4, 2}]];
Print["Delay F1-Score: ", NumberForm[delayF1, {4, 2}]];

(*Precision,Recall,and F1 for Cancellation Prediction*)
cancelTruePositives = Count[Transpose[{cancelBinaryPredictions, cleanedCancelTargetList}], {1, 1}];
cancelFalsePositives = Count[Transpose[{cancelBinaryPredictions, cleanedCancelTargetList}], {1, 0}];
cancelFalseNegatives = Count[Transpose[{cancelBinaryPredictions, cleanedCancelTargetList}], {0, 1}];
cancelPrecision = N[cancelTruePositives / (cancelTruePositives + cancelFalsePositives), 2];
cancelRecall = N[cancelTruePositives / (cancelTruePositives + cancelFalseNegatives), 2];
cancelF1 = N[2 * (cancelPrecision * cancelRecall) / (cancelPrecision + cancelRecall), 2];

Print["Cancellation Precision: ", NumberForm[cancelPrecision, {4, 2}]];
Print["Cancellation Recall: ", NumberForm[cancelRecall, {4, 2}]];
Print["Cancellation F1-Score: ", NumberForm[cancelF1, {4, 2}]];
```

```
Trained Delay Random Forest Model: PredictorFunction[ ⊞ Input type: Mixed (number: 19)
                                                        Method: RandomForest ]

Trained Cancellation Random Forest Model: PredictorFunction[ ⊞ Input type: Mixed (number: 19)
                                                               Method: RandomForest ]

Number of Correct Delay Predictions: 8306

Number of Correct Cancellation Predictions: 9999

Delay Prediction Accuracy (Percentage): 83.00%

Cancellation Prediction Accuracy (Percentage): 100.00%

Delay Precision: 0.96

Delay Recall: 0.62
Delay F1-Score: 0.80

Cancellation Precision: 1.00

Cancellation Recall: 1.00

Cancellation F1-Score: 1.00
```

Code Summary:

1. Data Preprocessing:

- Categorization: The code categorizes the categorical columns, “AIRLINE” and “ORIGIN”, by assigning them numeric codes using the categorizeColumn function.
- Feature Engineering: It adds new columns to the dataset, such as DELAYED (indicating whether the flight was delayed) and CANCELLED_LABEL (indicating whether the flight was canceled).
- Data Cleaning: It removes rows with missing values in the selected features to ensure clean and valid data for training.

2. Target Variables:

The target variables for prediction are:

- DELAYED: Whether a flight was delayed (1 for delay, 0 for no delay).
- CANCELLED_LABEL: Whether a flight was canceled (1 for canceled, 0 for not canceled).

3. Model Training:

Random Forest Models: It trains two separate Random Forest models:

- One to predict flight delays (delayRandomForest).
- One to predict flight cancellations (cancelRandomForest).

4. Prediction and Evaluation:

- Predictions: After training, the models are used to predict the likelihood of delays and cancellations for each flight.
- Thresholding: The model outputs continuous probabilities, which are converted to binary values (1 or 0) based on a threshold of 0.5.
- Accuracy: The accuracy of the models is calculated as the percentage of correct predictions out of the total predictions.
- Precision, Recall, and F1-Score: These evaluation metrics are calculated for both delay and cancellation predictions, providing a more detailed measure of model performance.

5. Output Explanation:

1. Model Training:

- Delay Random Forest Model: Successfully trained a Random Forest model for predicting delays.
- Cancellation Random Forest Model: Successfully trained a Random Forest model for predicting cancellations.

2. Accuracy:

- Delay Prediction Accuracy: The model correctly predicted delays with 83.00% accuracy.
- Cancellation Prediction Accuracy: The model correctly predicted cancellations with 100.00% accuracy.

3. Precision, Recall, and F1-Score:

a. Delay Prediction:

- Precision: 0.96 (The model correctly predicted delays 96% of the time when it predicted a delay).
- Recall: 0.62 (The model detected 62% of all actual delays).
- F1-Score: 0.80 (A balance between precision and recall, indicating good overall performance for delay prediction).

b. Cancellation Prediction:

- Precision: 1.00 (The model correctly predicted cancellations 100% of the time when it predicted a cancellation).
- Recall: 1.00 (The model detected 100% of actual cancellations).
- F1-Score: 1.00 (Perfect performance for cancellation prediction).

b. Analysis and Business Logic

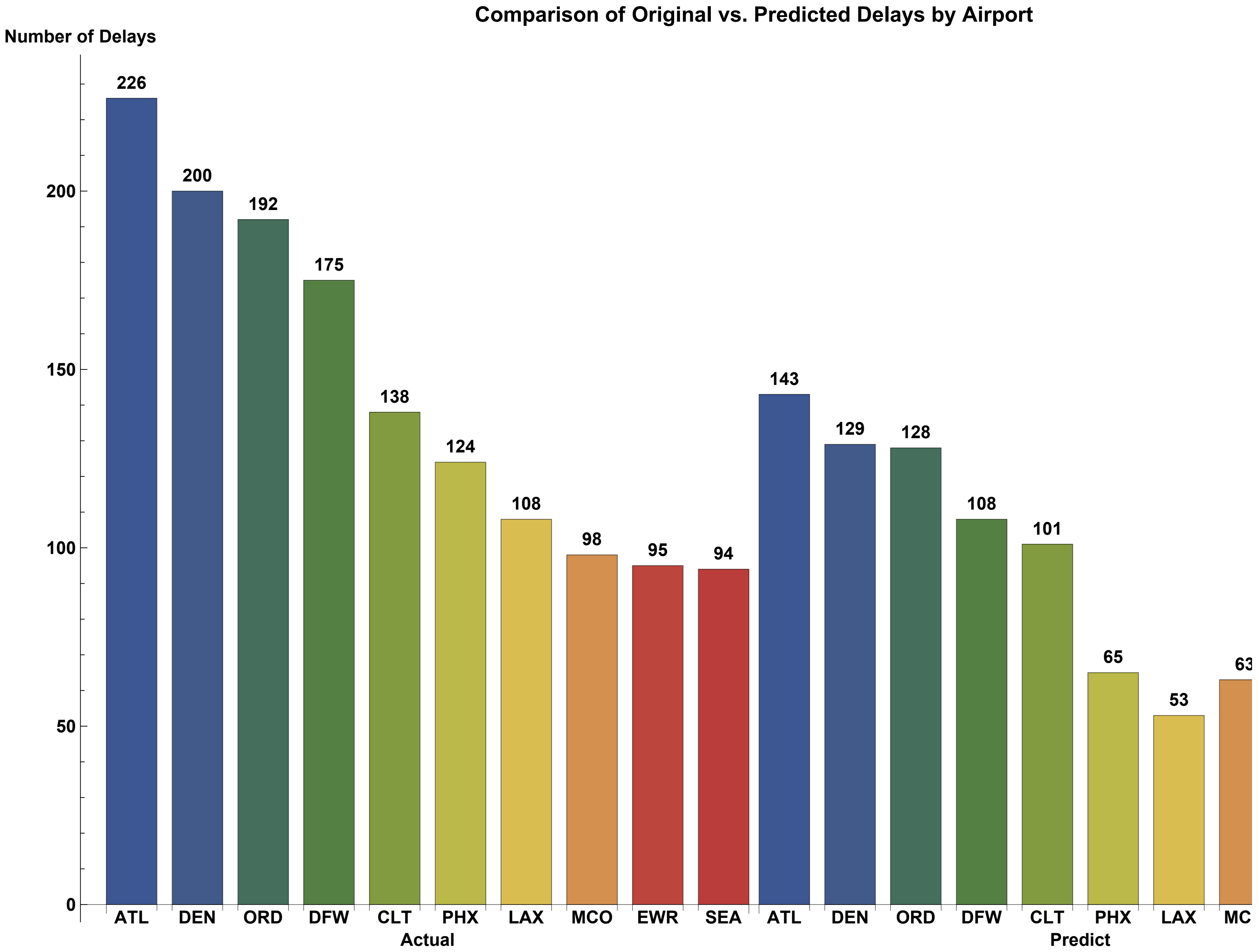
i . Delay and Cancellation Likelihood

- **Airports by Delay Likelihood**

```
In[•]:= (*Calculate the number of delays per airport for original and predicted*)
        (*For original delays:Group data by airport and sum up the delay counts*)
        originalDelayCounts = Normal[GroupBy[numericDataset, #ORIGIN &, Total[Lookup[#, "DELAYED", 0]] &]];

        (*For predicted delays:Group by airport and sum the binary predictions for delays*)
        predictedDelayCounts =
          Normal[GroupBy[Transpose[{numericDataset⟦All, "ORIGIN"⟧, delayBinaryPredictions}], First, Total[Last /@ #] &]];

        (*Take the top 10 airports by original delay counts*)

        (*Sort original delays in descending order and take the top 10*)
        sortedOriginalDelays = ReverseSortBy[originalDelayCounts, Last];
        top10OriginalDelays = Take[sortedOriginalDelays, 10];

        (*Extract airport names and delay counts for the top 10*)
        top10Airports = Keys[top10OriginalDelays];
        originalDelays = Values[top10OriginalDelays];
        reverseAirportCodes = Association[Reverse /@ Normal[originCodes]];
        top10AirportsWithNames = Map[Function[origin, reverseAirportCodes[origin]], top10Airports];
        top10AirportsWithNames

        (*Match the same airports for predicted delays*)
        (*Retrieve the predicted delays for the top 10 airports in the same order*)
        predictedDelays = Lookup[predictedDelayCounts, top10Airports, 0];
        predictedDelays

        (*Generate a grouped bar chart to compare original delays with predicted delays*)
        comparisonChartDelay =
          BarChart[{originalDelays, predictedDelays}, ChartLabels → {{"Actual", "Predict"}, top10AirportsWithNames},
           PlotLabel → "Comparison of Original vs. Predicted Delays by Airport",
           AxesLabel → {"Airports", "Number of Delays"}, BarSpacing → 0.3, LabelStyle → Directive[FontSize → 12, Bold],
           ChartStyle → "DarkRainbow", ImageSize → 1000, ChartLayout → "Grouped", LabelingFunction → Above];

        comparisonChartDelay
Out[•]=
        {ATL, DEN, ORD, DFW, CLT, PHX, LAX, MCO, EWR, SEA}
```

```
Out[•]=
        {143, 129, 128, 108, 101, 65, 53, 63, 81, 65}
```

Out[∘]=



Purpose: The bar chart, titled "**Comparison of Original vs. Predicted Delays by Airport**", highlights the actual number of delays versus the predicted number of delays for various airports after running a random forest model for predicting delays and cancellations of flights. It provides insights into the accuracy of the model and helps identify areas for improvement in delay predictions.

Y-Axis:

- Label: “Number of Delays”
- Represents: The total number of flight delays for each airport.
- Higher bars: Indicate a greater number of delays.

X-Axis:

- Label: “Airports”
- Represents: Different airports with their respective actual and predicted delay counts.

Insights:

1. Most Delays (Actual vs. Predicted):
   - ATL (Atlanta): Leads with the highest number of actual delays (226) but has a significantly lower number of predicted delays (143).
   - DEN (Denver) and ORD (Chicago O’Hare): Follow with actual delays of 200 and 192, and predicted delays of 129 and 128, respectively.
2. Moderate Discrepancies:
   - DFW (Dallas-Fort Worth): Shows actual delays of 175 and predicted delays of 108.
   - CLT (Charlotte) and PHX (Phoenix): Have actual delays of 138 and 124, with predicted delays of 101 and 65.
3. Lower Delay Counts:
   - LAX (Los Angeles), MCO (Orlando), EWR (Newark), and SEA (Seattle): Have actual delays ranging from 94 to 108, while predicted delays range from 65 to 53.

Key Takeaways:

- Operational Focus: Improving prediction accuracy can help in better resource allocation and operational planning for airlines and airports.
- Passenger Impact: Understanding delay patterns can assist passengers in planning their travel better, potentially opting for airports with lower predicted delays.

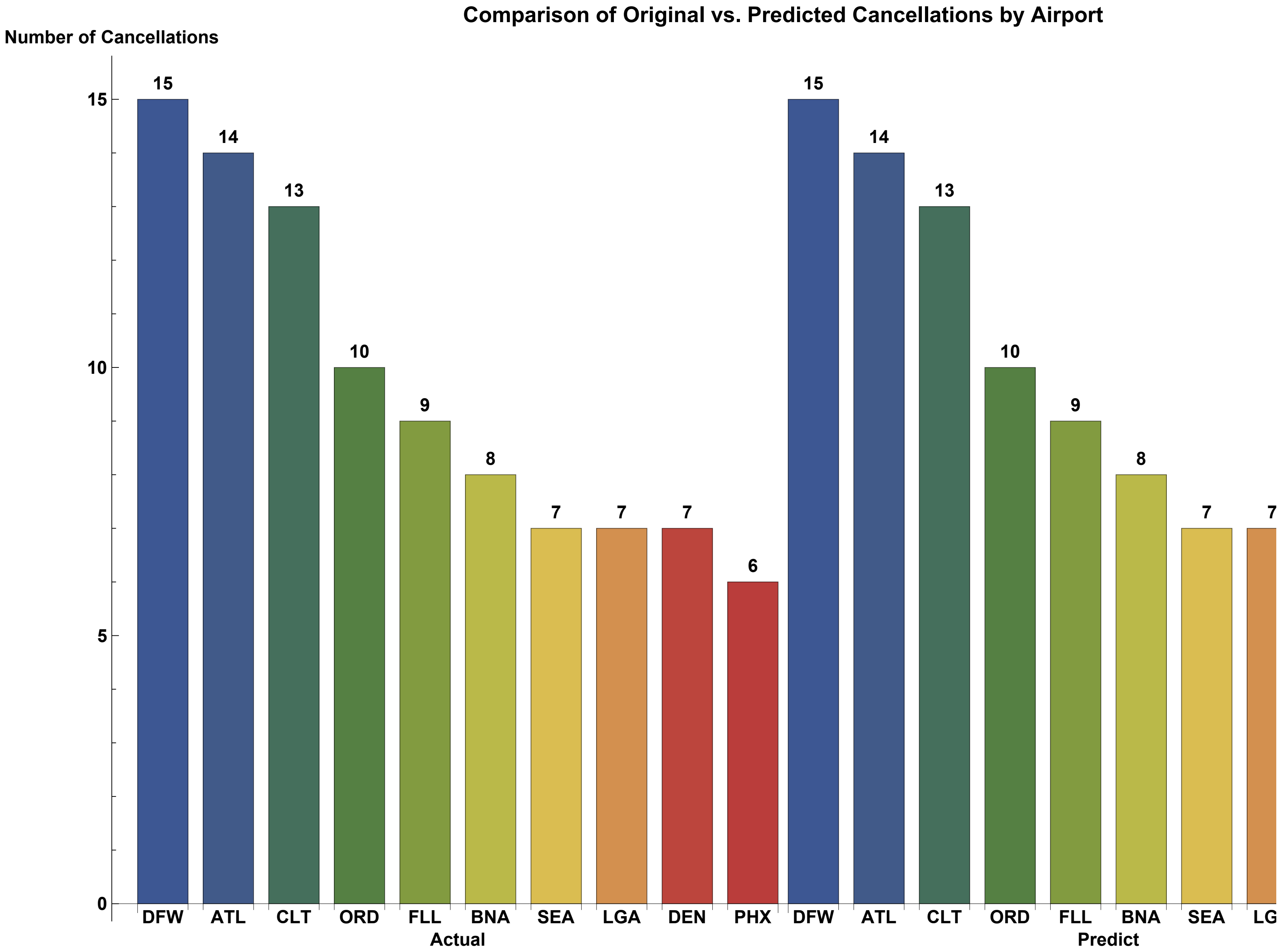
Airports by Cancellation Likelihood

```
In[∘]:= (*Calculate the number of cancellations per airport for original and predicted*)
        (*For original cancellations:Group data by airport and sum up the cancellation counts*)
        originalCancelCounts = Normal[GroupBy[numericDataset, #ORIGIN &, Total[Lookup[#, "CANCELLED_LABEL", 0]] &]];

        (*For predicted cancellations:Group by airport and sum the binary predictions for cancellations*)
        predictedCancelCounts =
          Normal[GroupBy[Transpose[{numericDataset〚All, "ORIGIN"〛, cancelBinaryPredictions}], First, Total[Last /@ #] &]];

        (*Sort original cancellations in descending order and take the top 10*)
        sortedOriginalCancellations = ReverseSortBy[originalCancelCounts, Last];
        top10OriginalCancellations = Take[sortedOriginalCancellations, 10];

        (*Extract airport names and cancellation counts for the top 10*)
        top10CancelAirports = Keys[top10OriginalCancellations];
        originalCancellations = Values[top10OriginalCancellations];
        reverseAirportCodes1 = Association[Reverse /@ Normal[originCodes]];
        top10AirportsWithNames1 = Map[Function[origin, reverseAirportCodes1[origin]], top10CancelAirports];
        top10AirportsWithNames1

        (*Retrieve the predicted cancellations for the top 10 airports in the same order*)
        predictedCancellations = Lookup[predictedCancelCounts, top10CancelAirports, 0];
        predictedCancellations

        (*Generate a grouped bar chart to compare original cancellations with predicted cancellations*)
        comparisonChartCancellation = BarChart[{originalCancellations, predictedCancellations},
           ChartLabels → {{"Actual", "Predict"}, top10AirportsWithNames1},
           PlotLabel → "Comparison of Original vs. Predicted Cancellations by Airport",
           AxesLabel → {"Airports", "Number of Cancellations"}, BarSpacing → 0.3, LabelStyle → Directive[FontSize → 12, Bold],
           ChartStyle → "DarkRainbow", ImageSize → 1000, ChartLayout → "Grouped", LabelingFunction → Above];

        comparisonChartCancellation
```

```
Out[∘]=
        {DFW, ATL, CLT, ORD, FLL, BNA, SEA, LGA, DEN, PHX}
```

```
Out[∘]=
        {15, 14, 13, 10, 9, 8, 7, 7, 7, 6}
```

Out[∘]=



Purpose: The bar chart, titled "**Comparison of Original vs. Predicted Cancellations by Airport**", highlights the actual number of cancellations versus the predicted number of cancellations for various airports after running a random forest model for predicting cancellations and delays of flights. It provides insights into the accuracy of the model and helps identify areas for improvement in cancellation predictions.

Y-Axis:

- Label: “Number of Cancellations”
- Represents: The total number of flight cancellations for each airport.
- Higher bars: Indicate a greater number of cancellations.

X-Axis:

- Label: “Airports”
- Represents: Different airports with their respective actual and predicted cancellation counts.

Insights:

1. Most Cancellations (Actual vs. Predicted):

- DFW (Dallas-Fort Worth): Leads with 15 cancellations in both actual and predicted counts.
- ATL (Atlanta): Shows 14 actual cancellations and 14 predicted cancellations.
- CLT (Charlotte): Has 13 actual cancellations and 13 predicted cancellations.

2. Lower Cancellation Counts:

- BNA (Nashville), SEA (Seattle), LGA (New York LaGuardia), and DEN (Denver): All show matching actual and predicted counts with 7, 7, 7, and 6 cancellations, respectively.

3. Model Performance:

- Significant Differences: The model predictions are accurate in all the cases,showcasing the robustness of the predictive algorithm employed.
- Operational Use: These insights can help airlines and airports focus on improving prediction accuracy and better managing resources to reduce cancellations.

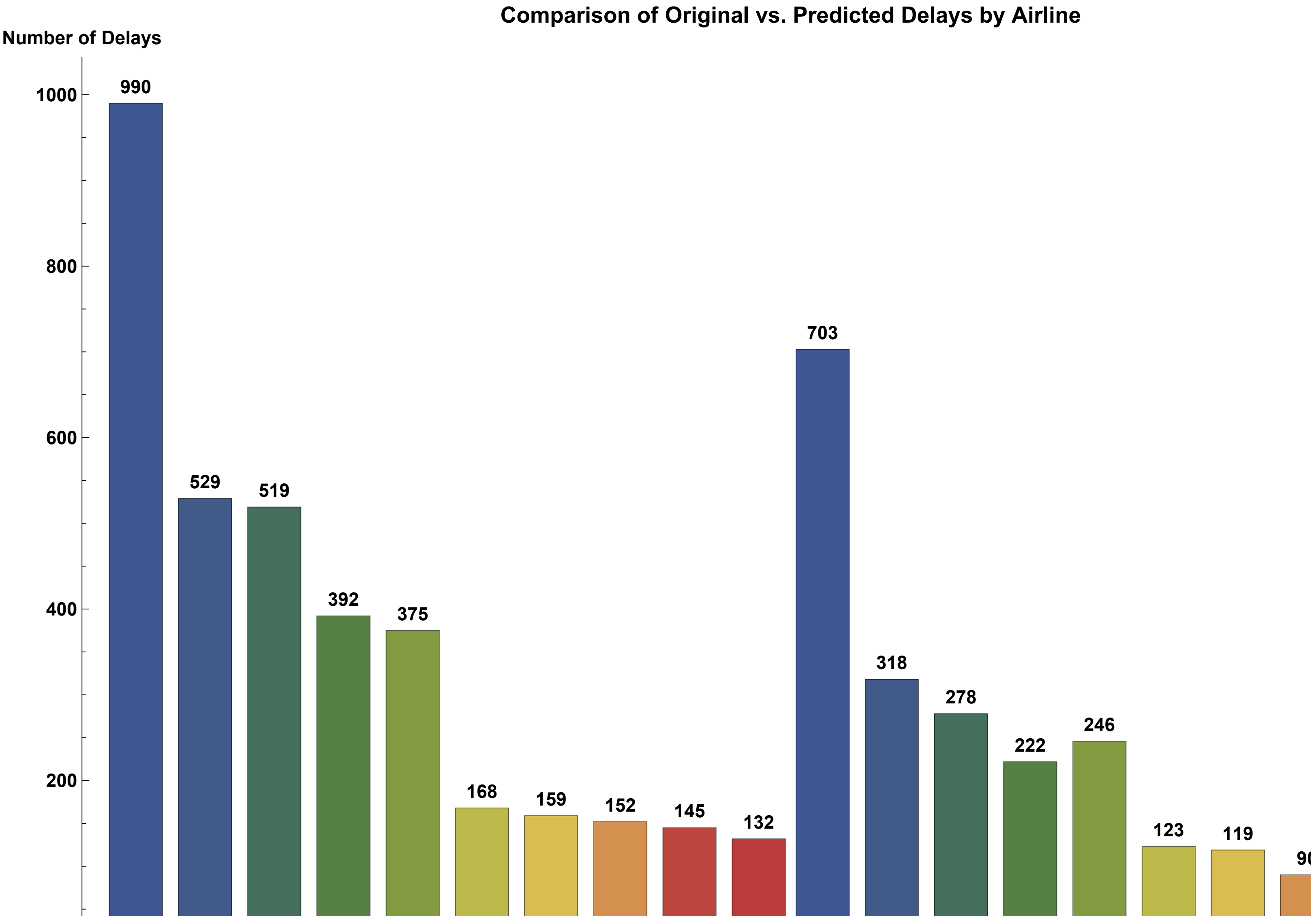
Key Takeaways:

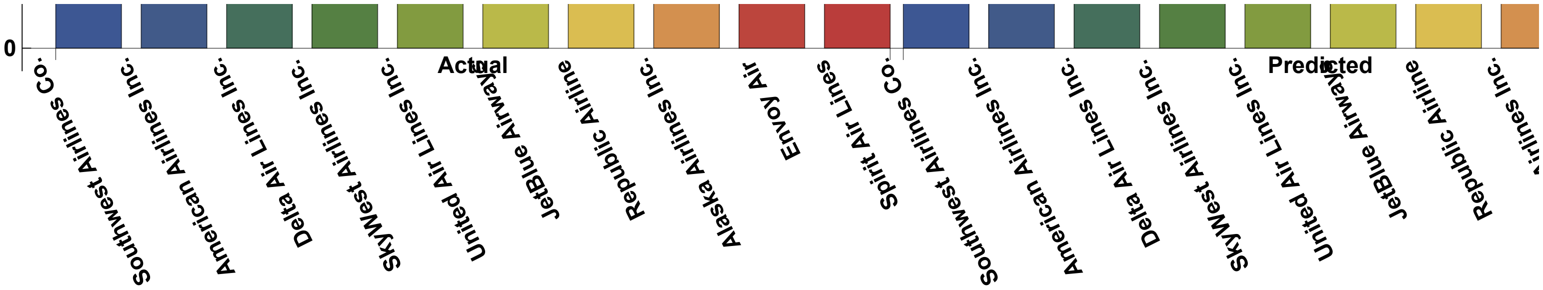
- Operational Focus: Improving prediction accuracy can help in better resource allocation and operational planning for airlines and airports.
- Passenger Impact: Understanding cancellation patterns can assist passengers in planning their travel better, potentially opting for airports with lower predicted cancellations.

Airlines by Delay Likelihood

```
In[•]:= (*Calculate the number of delays per airline for original and predicted*)
        (*For original delays:Group data by airline and sum up the delay counts*)
        originalDelayCounts1 = Normal[GroupBy[numericDataset, #AIRLINE &, Total[Lookup[#, "DELAYED", 0]] &]];

        (*For predicted delays:Group by airline and sum the binary predictions for delays*)
        predictedDelayCounts1 =
          Normal[GroupBy[Transpose[{numericDataset⟦All, "AIRLINE"⟧, delayBinaryPredictions}], First, Total[Last /@ #] &]];

        (*Sort original delays in descending order and take the top 10*)
        sortedOriginalDelays1 = ReverseSortBy[originalDelayCounts1, Last];
        top10OriginalDelays1 = Take[sortedOriginalDelays1, 10];

        (*Extract airline names and delay counts for the top 10*)
        top10Airlines = Keys[top10OriginalDelays1];
        originalDelays1 = Values[top10OriginalDelays1];

        (*Create a reverse mapping from numeric airline names to airline names*)
        reverseAirlineCodes = Association[Reverse /@ Normal[airlineCodes]];
        top10AirlinesWithNames = Map[Function[airline, reverseAirlineCodes[airline]], top10Airlines];
        top10AirlinesWithNames

        (*Retrieve the predicted delays for the top 10 airlines in the same order*)
        predictedDelays1 = Lookup[predictedDelayCounts1, top10Airlines, 0];
        predictedDelays1

        (*Generate a grouped bar chart to compare original delays with predicted delays*)
        comparisonChartDelay1 = BarChart[{originalDelays1, predictedDelays1},
           ChartLabels → {{"Actual", "Predicted"}, Placed[Rotate[#, 90] & /@ top10AirlinesWithNames, Below]},
           PlotLabel → "Comparison of Original vs. Predicted Delays by Airline",
           AxesLabel → {"Airlines", "Number of Delays"}, BarSpacing → 0.3, LabelStyle → Directive[FontSize → 12, Bold],
           ChartStyle → "DarkRainbow", ImageSize → 1000, ChartLayout → "Grouped", LabelingFunction → Above];

        comparisonChartDelay1
```

```
Out[•]=
  {Southwest Airlines Co., American Airlines Inc., Delta Air Lines Inc., SkyWest Airlines Inc.,
   United Air Lines Inc., JetBlue Airways, Republic Airline, Alaska Airlines Inc., Envoy Air, Spirit Air Lines}
Out[•]=
  {703, 318, 278, 222, 246, 123, 119, 90, 89, 87}
Out[•]=
```





Purpose: The bar chart, titled "**Comparison of Original vs. Predicted Delays by Airline**", highlights the actual number of delays versus the predicted number of delays for various airlines after running a random forest model for predicting delays and cancellations of flights. It provides insights into the accuracy of the model and helps identify areas for improvement in delay predictions.

Y-Axis:

- Label: "Number of Delays"
- Represents: The total number of flight delays for each airline.
- Higher bars: Indicate a greater number of delays.

X-Axis:

- Label: "Airlines"
- Represents: Different airlines with their respective actual and predicted delay counts.

Insights:

1. Most Delays (Actual vs. Predicted):

- Southwest Airlines Co.: Leads with the highest number of actual delays (990) but has a significantly lower number of predicted delays (703).
- American Airlines Inc.: Shows 529 actual delays and 318 predicted delays.
- Delta Air Lines Inc.: Has 519 actual delays and 278 predicted delays.

2. Moderate Discrepancies:

- SkyWest Airlines Inc.: Shows actual delays of 392 and predicted delays of 222.
- United Air Lines Inc.: Has actual delays of 375 and predicted delays of 246.

3. Lower Delay Counts:

- JetBlue Airways, Republic Airline, Alaska Airlines Inc., Envoy Air, and Spirit Air Lines: Have actual delays ranging from 132 to 168, while predicted delays range from 87 to 123.

Key Takeaways:

- Operational Focus: Improving prediction accuracy can help in better resource allocation and operational planning for airlines.

- Passenger Impact: Understanding delay patterns can assist passengers in planning their travel better, potentially opting for airlines with lower predicted delays.

Airlines by Cancellation Likelihood

```
In[*]:= (*Calculate the number of cancellations per airline for original and predicted*)
(*For original cancellations:Group data by airline and sum up the cancellation counts*)
originalCancelCounts1 = Normal[GroupBy[numericDataset, #AIRLINE &, Total[Lookup[#, "CANCELLED_LABEL", 0]] &]];

(*For predicted cancellations:Group by airline and sum the binary predictions for cancellations*)
predictedCancelCounts1 =
  Normal[GroupBy[Transpose[{numericDataset[[All, "AIRLINE"]], cancelBinaryPredictions}], First, Total[Last /@ #] &]];

(*Sort original cancellations in descending order and take the top 10*)
sortedOriginalCancellations1 = ReverseSortBy[originalCancelCounts1, Last];
top10OriginalCancellations1 = Take[sortedOriginalCancellations1, 10];

(*Extract airline names and cancellation counts for the top 10*)
top10CancelAirlines1 = Keys[top10OriginalCancellations1];
originalCancellations1 = Values[top10OriginalCancellations1];
reverseAirlineCodes1 = Association[Reverse /@ Normal[airlineCodes]];
top10AirlinesWithNames1 = Map[Function[airline, reverseAirlineCodes1[airline]], top10CancelAirlines1];
top10AirlinesWithNames1

(*Retrieve the predicted cancellations for the top 10 airlines in the same order*)
predictedCancellations1 = Lookup[predictedCancelCounts1, top10CancelAirlines1, 0];
predictedCancellations1

(*Generate a grouped bar chart to compare original cancellations with predicted cancellations*)
comparisonChartCancellation1 = BarChart[{originalCancellations1, predictedCancellations1},
   ChartLabels → {{"Actual", "Predict"}, Rotate[#, 90] & /@ top10AirlinesWithNames1},
   PlotLabel → "Comparison of Original vs. Predicted Cancellations by Airline",
   AxesLabel → {"Airlines", "Number of Cancellations"}, BarSpacing → 0.3, LabelStyle → Directive[FontSize → 12, Bold],
   ChartStyle → "DarkRainbow", ImageSize → 1000, ChartLayout → "Grouped", LabelingFunction → Above];

comparisonChartCancellation1
```

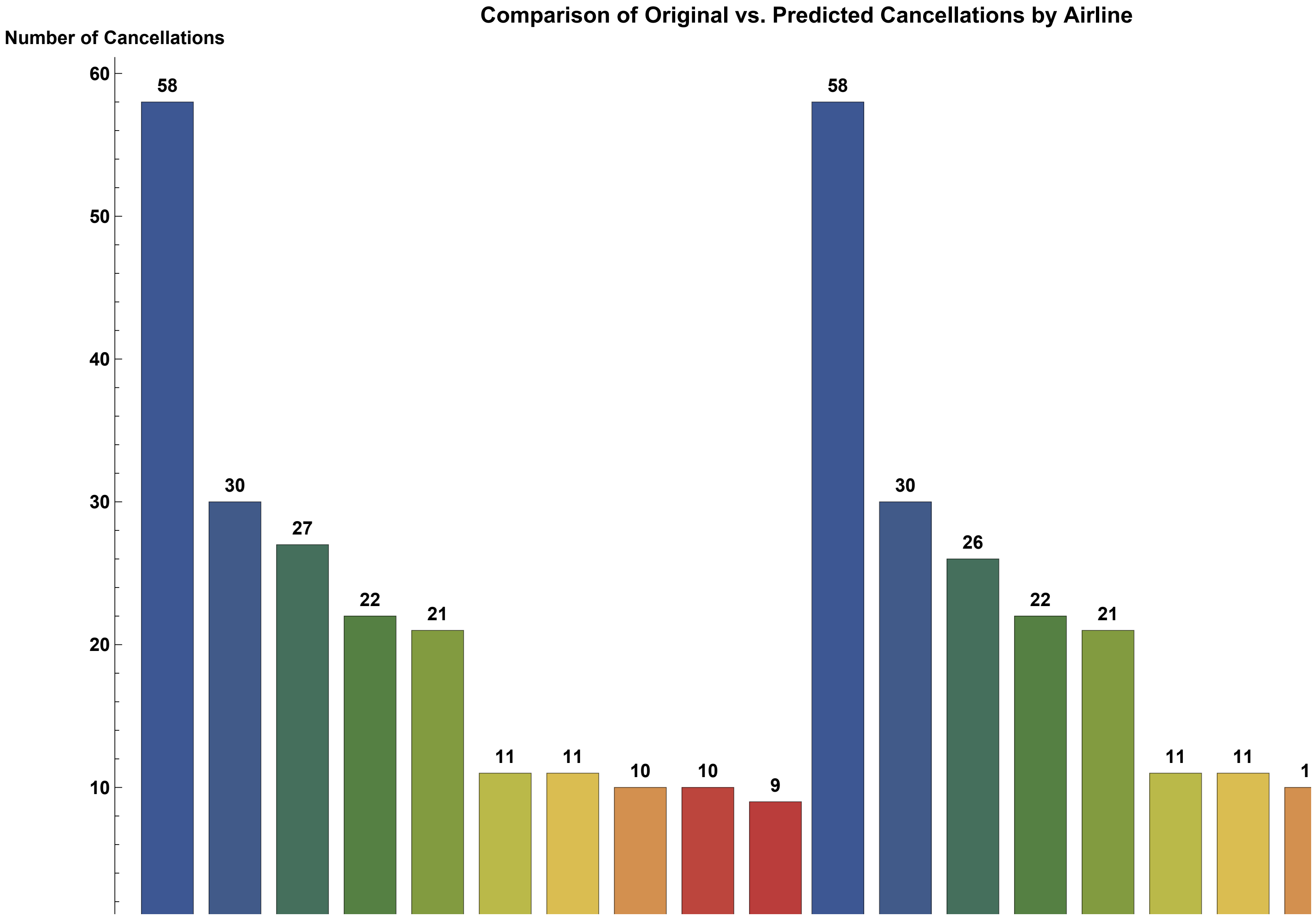
Out[∘]=

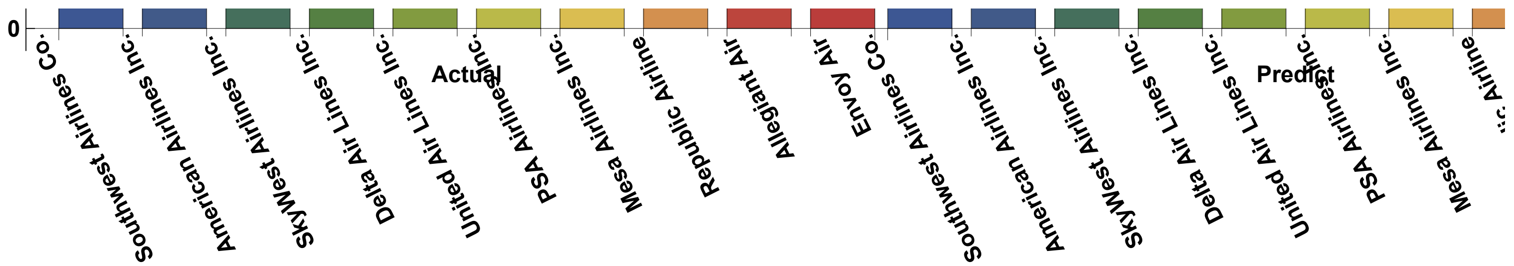
```
{Southwest Airlines Co., American Airlines Inc., SkyWest Airlines Inc., Delta Air Lines Inc.,
 United Air Lines Inc., PSA Airlines Inc., Mesa Airlines Inc., Republic Airline, Allegiant Air, Envoy Air}
```

Out[∘]=

```
{58, 30, 26, 22, 21, 11, 11, 10, 10, 9}
```

Out[∘]=





Purpose: The bar chart, titled "**Comparison of Original vs. Predicted Cancellations by Airline**", highlights the actual number of cancellations versus the predicted number of cancellations for various airlines after running a random forest model for predicting cancellations and delays of flights. It provides insights into the accuracy of the model and helps identify areas for improvement in cancellation predictions.

Y-Axis:

- Label: "Number of Cancellations"
- Represents: The total number of flight cancellations for each airline.
- Higher bars: Indicate a greater number of cancellations.

X-Axis:

- Label: "Airlines"
- Represents: Different airlines with their respective actual and predicted cancellation counts.

Insights:

1. Most Cancellations (Actual vs. Predicted):
- Southwest Airlines Co.: Shows 58 actual cancellations and 58 predicted cancellations.
- American Airlines Inc.: Has 30 actual cancellations and 30 predicted cancellations.
- SkyWest Airlines Inc.: Shows 27 actual cancellations and 26 predicted cancellations.

2. Lower Cancellation Counts:
- PSA Airlines Inc. and Mesa Airlines Inc.: Both show 11 actual and predicted cancellations.
- Republic Airlines and Allegiant Air: Each shows 10 actual and predicted cancellations.
- Envoy Air: Shows 9 actual and predicted cancellations.

3. Model Performance:
- Overall Accuracy: The model predictions closely match the actual cancellations, indicating a robust predictive algorithm.
- Operational Use: These insights can help airlines focus on maintaining accuracy in predictions and managing resources effectively to reduce cancellations.

Key Takeaways:

- Operational Focus: Consistent accuracy in prediction can aid in better resource allocation and operational planning for airlines.
- Passenger Impact: Understanding cancellation patterns can assist passengers in planning their travel better, potentially opting for airlines with lower predicted cancellations.

ii . Airlines by On - Time Performance

```
In[•]:= (*Filter the dataset to keep only the rows that satisfy the conditions:ARR_DELAY≤0,
DEP_DELAY≤0,CANCELLED==0,DIVERTED==0*) filteredOntimeData =
  Select[dataset, ((#⟦"ARR_DELAY"⟧ ≤ 0 || #⟦"DEP_DELAY"⟧ ≤ 0) && #⟦"CANCELLED"⟧ == 0 && #⟦"DIVERTED"⟧ == 0) &];

(*Group the filtered data by AIRLINE and count the number of valid flights for each airline*)
onTimeCounts = Normal[GroupBy[filteredOntimeData, #["AIRLINE"] &, Length]];

(*Calculate the total number of flights per airline*)
totalFlights = Normal[GroupBy[dataset, #["AIRLINE"] &, Length]];

(*Calculate the on-time rate for each airline as a fraction of its own total flights*)
onTimeRates =
  AssociationThread[Keys[onTimeCounts], N[Values[onTimeCounts] / Lookup[totalFlights, Keys[onTimeCounts], 1], 2]];

(*Convert the on-time rates to a dataset format for better visualization*)
OnTimeRatesTable = Dataset[KeyValueMap[<|"Airlines" → #1, "Actual On-time Rates" → #2|> &, onTimeRates]];

(*Display the Real on-time rates table*)
OnTimeRatesTable
```

Out[•]=

Airlines	Actual On–time Rates
SkyWest Airlines Inc.	0.79
Republic Airline	0.81
United Air Lines Inc.	0.74
PSA Airlines Inc.	0.77
Endeavor Air Inc.	0.87
Spirit Air Lines	0.71
Envoy Air	0.8
Mesa Airlines Inc.	0.72
Southwest Airlines Co.	0.65
Delta Air Lines Inc.	0.78
Frontier Airlines Inc.	0.67
Allegiant Air	0.59
American Airlines Inc.	0.74
Hawaiian Airlines Inc.	0.64
Alaska Airlines Inc.	0.73
JetBlue Airways	0.71
Horizon Air	0.82
ExpressJet Airlines LLC d/b/a aha!	0.78

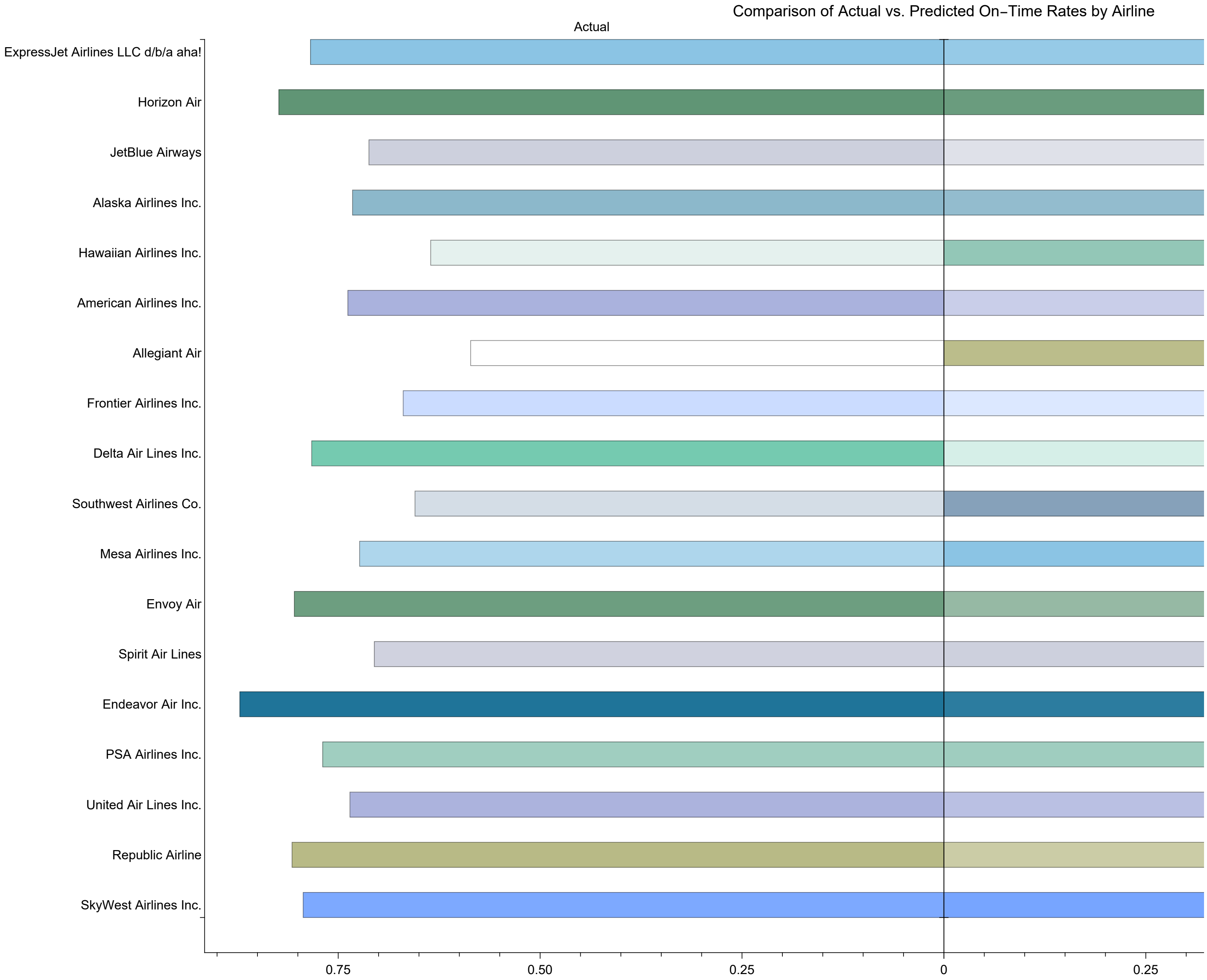
```
In[•]:= (*Use delayBinaryPredictions to identify on-time flights*)
        onTimePredictions = Select[Transpose[{cleanedFeatureDataList, delayBinaryPredictions}], Last[#] == 0 &];

        (*Group predicted on-time flights by AIRLINE and count the number of flights per airline*)
        onTimePredictedCounts = Normal[GroupBy[onTimePredictions⟦All, 1⟧, #["AIRLINE"] &, Length]];

        (*Calculate the total number of flights per airline in the cleaned dataset*)
        totalFlights = Normal[GroupBy[cleanedFeatureDataList, #["AIRLINE"] &, Length]];

        (*Calculate the predicted on-time rate for each airline as a fraction of the total flights*)
        onTimePredictedRates = AssociationThread[Keys[onTimePredictedCounts],
           N[Values[onTimePredictedCounts] / Lookup[totalFlights, Keys[onTimePredictedCounts], 1], 2]];

        (*Use airlineCodes to map numeric codes back to airline names*)
        airlineNames = KeyValueMap[#2 → #1 &, airlineCodes];
        onTimePredictedRatesWithNames =
          AssociationThread[Lookup[airlineNames, Keys[onTimePredictedRates]], Values[onTimePredictedRates]];

        (*Convert the on-time rates to a dataset format with appropriate column names for better visualization*)
        predictedOnTimeRatesTable =
          Dataset[KeyValueMap[<|"Airlines" → #1, "Predicted On-time Rates" → #2|> &, onTimePredictedRatesWithNames]];

        (*Display the predicted on-time rates table*)
        predictedOnTimeRatesTable
```

Out[∘]=

Airlines	Predicted On–time Rates
SkyWest Airlines Inc.	0.8
Republic Airline	0.75
United Air Lines Inc.	0.71
PSA Airlines Inc.	0.77
Endeavor Air Inc.	0.86
Spirit Air Lines	0.71
American Airlines Inc.	0.74
Envoy Air	0.78
Mesa Airlines Inc.	0.78
Frontier Airlines Inc.	0.64
Southwest Airlines Co.	0.64
Delta Air Lines Inc.	0.8
Allegiant Air	0.68
Hawaiian Airlines Inc.	0.79
Alaska Airlines Inc.	0.72
JetBlue Airways	0.67
Horizon Air	0.81
ExpressJet Airlines LLC d/b/a aha!	0.76

```
In[•]:= (*Extract data from the tables*)
        airlines = Normal[OnTimeRatesTable[All, "Airlines"]];
        actualRates = Normal[OnTimeRatesTable[All, "Actual On-time Rates"]];
        predictedRates = Normal[predictedOnTimeRatesTable[All, "Predicted On-time Rates"]];

        (*Create a paired bar chart to compare actual and predicted on-time rates*)
        comparisonChartOnTime = PairedBarChart[actualRates,
           predictedRates, BarSpacing → {0, 0, 1}, (*Make bars close to each other for comparison*)
           ChartLabels → {Placed[{"Actual", "Predicted"}, Above], None, Placed[airlines, "LeftAxis"]},
           (*Labels for datasets and airlines*)
           PlotLabel → "Comparison of Actual vs. Predicted On-Time Rates by Airline",
           AxesLabel → {Style["On-Time Rate (%)", 12, Bold]}, Method → {"LabelStyle" → "Short"},
           ChartStyle → {{Opacity[1], Opacity[0.8]}, None, 66},
           ColorFunction → Function[{height}, Opacity[height]],
           ImageSize → 1200];

        (*Display the chart*)
        comparisonChartOnTime
```

Out[ ]=



Purpose:

The “**Comparison of Actual vs. Predicted On-Time Rates by Airline**” bar chart visualizes how accurately on-time performance for various airlines aligns with predictions. This chart provides insights into predictive reliability and operational performance for different airlines.

Y-Axis:

- Representation: The names of the airlines being compared.
- Scale: Lists each airline, from ExpressJet Airlines LLC d/b/a aha! to SkyWest Airlines Inc.

X-Axis:

- Representation: The rate of on-time flights for each airline, depicted as percentages or proportions between 0 and 1.
- Bars: Each airline has two bars—one representing actual on-time rates and the other showing predicted on-time rates.

Insights:

1. Alignment Between Actual and Predicted:

- Airlines such as Horizon Air and Alaska Airlines Inc. show close alignment between actual and predicted rates, indicating strong predictive reliability.

2. Significant Gaps:

- Some airlines, such as Envoy Air, show notable differences between actual and predicted rates, suggesting potential discrepancies in predictive modeling or operational variability.

3. High On-Time Performance:

- Airlines like Endeavor Air and Delta Air Lines Inc. demonstrate high actual and predicted on-time rates, reflecting efficient operations.

4. Moderate and Lower Rates:

- Airlines such as Spirit Airlines Inc. and Frontier Airlines Inc. exhibit moderate or lower actual and predicted on-time rates, possibly due to operational or scheduling constraints.

5. Operational and Predictive Insights:

- Airlines with strong alignment between predicted and actual rates likely rely on robust data analytics. Variability in alignment could highlight opportunities for improving predictive accuracy or addressing operational challenges.

Key Takeaways:

1. Predictive Accuracy Varies Across Airlines:

- While some airlines, such as Horizon Air and Alaska Airlines Inc., exhibit close alignment between actual and predicted on-time rates, others, like Envoy Air, display significant gaps, highlighting variability in predictive modeling success.

2. High-Performing Airlines Stand Out:

- Airlines like Endeavor Air and Delta Air Lines Inc. show consistently high on-time performance, both in actual and predicted rates, emphasizing operational efficiency and reliability.

3. Opportunities for Improvement:

- Airlines with lower alignment or on-time rates, such as Spirit Airlines Inc. and Frontier Airlines Inc., could focus on enhancing operational processes and predictive models to boost performance and customer satisfaction.

4. Operational Complexity Reflected in Variability:

- The differences in actual versus predicted rates suggest that some airlines face operational challenges or external factors that impact punctuality beyond what predictions account for.

5. Insights for Strategic Decision-Making:

- The comparison underscores the importance of accurate predictive analytics for planning and optimizing operations, as well as identifying areas where airlines can improve service reliability.

In[ ]:=

c. Actionable Insights

Based on the model outcomes, we propose the following operational changes as recommendations to the top 10 airports and airlines experiencing severe delays and cancellations:

a. Implementing better traffic management or predictive maintenance.
b. Adjusting flight schedules to reduce congestion during peak hours.
c. Improving handling of weather-related delays by having more flexible staffing or better communication with ground services.

Based on the model outcomes, we propose the following recommendations to the top 10 airports and airlines experiencing poor on-time performance rates:

a. Schedule adjustments, potentially moving flights away from peak congestion times.
b. Maintenance improvements, ensuring that technical issues do not contribute to delays.
c. Review reasons for delays (e.g., operational issues, crew management, etc.) and implement efficiency improvements.

2. Random Forest Regression and Classification Model: For Predicting Delay Duration and Delay Duration Binned into Short, Medium, and Long Delays

a. Modelling

```
In[∘]:= (*Load and Preprocess the Data*)
numericDataset = Map[Association, Normal[dataset]];

(*Categorize categorical variables *)
categorizeColumn[columnData_] := AssociationThread[Union[columnData] → Range[Length[Union[columnData]]]];

(*Map AIRLINE and ORIGIN to numeric codes*)
airlineCodes = categorizeColumn[numericDataset⟦All, "AIRLINE"⟧];
originCodes = categorizeColumn[numericDataset⟦All, "ORIGIN"⟧];

(*Replace categorical values with numeric codes for AIRLINE and ORIGIN*)
numericDataset =
  Map[Function[assoc, assoc~Join~<|"AIRLINE" → Lookup[airlineCodes, assoc["AIRLINE"], Missing["NotMapped"]],
       "ORIGIN" → Lookup[originCodes, assoc["ORIGIN"], Missing["NotMapped"]]|>], numericDataset];

(*Add new columns for DELAYED and CANCELLED_LABEL*)
numericDataset =
  Map[Function[assoc, assoc~Join~<|"DELAYED" → If[assoc["ARR_DELAY"] > 0 || assoc["DEP_DELAY"] > 0, 1, 0],
       "CANCELLED_LABEL" → If[assoc["CANCELLED"] > 0, 1, 0]|>], numericDataset];

(*Prepare the target vector for delay duration prediction*)
delayTargetList = numericDataset⟦All, "ARR_DELAY"⟧; (*Using ARR_DELAY as delay duration*)

(*Prepare features for training*)
features = {"AIRLINE", "ORIGIN", "DEST", "CRS_DEP_TIME", "DEP_TIME", "CRS_ARR_TIME", "ARR_TIME", "CRS_ELAPSED_TIME",
    "DISTANCE", "DELAY_DUE_WEATHER", "DELAY_DUE_CARRIER", "DELAY_DUE_NAS", "DELAY_DUE_SECURITY"};

featureDataList = Map[Function[assoc, KeyTake[assoc, features]], numericDataset];
```

```
(*Clean the data by removing missing values or empty rows*)
validPositions = Select[Range[Length[featureDataList]], FreeQ[featureDataList⟦#⟧, _?MissingQ] &];
cleanedFeatureDataList = featureDataList⟦validPositions⟧;
cleanedDelayTargetList = delayTargetList⟦validPositions⟧;

(*Train the Random Forest model for Delay Duration Prediction (Regression)*)
delayDurationRandomForest = Predict[cleanedFeatureDataList → cleanedDelayTargetList, Method → "RandomForest"];
Print["Trained Delay Duration Random Forest Model: ", delayDurationRandomForest];

(*Make Predictions for Delay Duration*)
delayDurationPredictions = delayDurationRandomForest[cleanedFeatureDataList];

(*Calculate the threshold for classification*)
(*Here we classify the delay duration into three categories:0-15 minutes (short),
15-60 minutes (medium), >60 minutes (long)*)
categorizeDelay[delay_] := Which[delay ≤ 15, "short", delay ≤ 60, "medium", delay > 60, "long"];

(*Convert actual and predicted delay durations into categories*)
actualCategories = Map[categorizeDelay, cleanedDelayTargetList];
predictedCategories = Map[categorizeDelay, delayDurationPredictions];

(*Calculate accuracy,precision,recall,and F1 score for the delay duration classification*)
(*Helper function to calculate precision,recall,and F1 score*)
confusionMatrix[actual_, predicted_] := Module[{tp, fp, tn, fn},
   tp = Count[Transpose[{actual, predicted}], {"short", "short"}] + Count[Transpose[{actual, predicted}],
      {"medium", "medium"}] + Count[Transpose[{actual, predicted}], {"long", "long"}];
   fp = Count[Transpose[{actual, predicted}], {"short", "medium"}] + Count[Transpose[{actual, predicted}],
      {"short", "long"}] + Count[Transpose[{actual, predicted}], {"medium", "short"}] +
     Count[Transpose[{actual, predicted}], {"medium", "long"}] + Count[Transpose[{actual, predicted}],
      {"long", "short"}] + Count[Transpose[{actual, predicted}], {"long", "medium"}];
   fn = Count[Transpose[{actual, predicted}], {"medium", "short"}] +
     Count[Transpose[{actual, predicted}], {"long", "short"}] + Count[Transpose[{actual, predicted}],
      {"short", "long"}] + Count[Transpose[{actual, predicted}], {"medium", "long"}];
   tn = Length[actual] - (tp + fp + fn);
   {tp, fp, tn, fn}];
```

```
(*Calculate confusion matrix components*)
{tp, fp, tn, fn} = confusionMatrix[actualCategories, predictedCategories];

(*Calculate accuracy,precision,recall,and F1 score for binning*)
accuracy = (tp + tn) / (tp + fp + tn + fn);
precision = tp / (tp + fp);
recall = tp / (tp + fn);
f1Score = 2 * (precision * recall) / (precision + recall);

(*Print the results in decimal format with two decimal places*)
Print["Accuracy for Delay Duration Classification: ", StringForm["`1`", N[accuracy, 2]]];
Print["Precision for Delay Duration Classification: ", StringForm["`1`", N[precision, 2]]];
Print["Recall for Delay Duration Classification: ", StringForm["`1`", N[recall, 2]]];
Print["F1 Score for Delay Duration Classification: ", StringForm["`1`", N[f1Score, 2]]];

(*RMSE evaluation for regression*)
delayDurationRMSE = Sqrt[Mean[(delayDurationPredictions - cleanedDelayTargetList)^2]];
Print["Root Mean Squared Error (RMSE) for Delay Duration (Regression): ",
  StringForm["`1`", N[delayDurationRMSE, 2]]];
```

```
Trained Delay Duration Random Forest Model: PredictorFunction[ ⊞  Input type: Mixed (number: 13)
                                                                  Method: RandomForest           ]
                                                               Data not saved. Save now

Accuracy for Delay Duration Classification: 0.9251`2.

Precision for Delay Duration Classification: 0.9509`2.

Recall for Delay Duration Classification: 0.9735845192996826047`2.

F1 Score for Delay Duration Classification: 0.9621085647796833106`2.

Root Mean Squared Error (RMSE) for Delay Duration (Regression): 26.851275705507266`
```

Code Summary:

1. Data Preprocessing:

- Categorization: The code categorizes the “AIRLINE” and “ORIGIN” columns by assigning numeric codes using the categorizeColumn function.

- Feature Engineering: New columns are added for DELAYED (indicating whether the flight was delayed) and CANCELLED_LABEL (indicating whether the flight was canceled).

- Feature Selection: A subset of features is selected for training the model, focusing on columns such as “AIRLINE”, “ORIGIN”, “DEST”, “CRS_DEP_TIME”, “DEP_TIME”, etc.

- Data Cleaning: The dataset is cleaned by removing rows containing missing values in the selected features.

2. Target Variables:

- The target variable for prediction is ARR_DELAY, which represents the arrival delay duration (in minutes).

3. Model Training:

- Random Forest Model (Regression): A Random Forest regression model is trained to predict the actual delay duration based on the selected features.

4. Prediction and Classification:

- Delay Duration Predictions: After training, the model predicts the delay duration for each flight.
- Categorizing Delays: The delay durations are categorized into three ranges:
- Short: Delay of 0-15 minutes
- Medium: Delay of 15-60 minutes
- Long: Delay of more than 60 minutes.
- The actual and predicted delays are then classified into these categories.

5. Performance Evaluation:

- Confusion Matrix: A confusion matrix is calculated to evaluate the model’s performance in classifying delays into the three categories.

- Accuracy, Precision, Recall, and F1 Score: These metrics are calculated based on the confusion matrix to evaluate the classification performance.

- RMSE (Root Mean Squared Error): As the model is a regression model, RMSE is calculated to measure the error in predicting delay durations.

6. Output Explanation:

1. Model Training:

- Trained Delay Duration Random Forest Model: Successfully trained a Random Forest model for predicting delay durations. The model uses a mix of 13 features.

2. Performance Metrics for Delay Duration Classification:

- Accuracy: The model correctly classified delay categories (short, medium, long) with an accuracy of 92.51%.
- Precision: The model’s precision (the percentage of correct predictions among all positive predictions) for delay classification is 95.09%.
- Recall: The model detected 97.36% of all actual delays, meaning it is very effective at identifying delays.
- F1-Score: The F1-Score, a balanced measure of precision and recall, is 96.21%, indicating good performance in both identifying delays and minimizing false positives.

3. Root Mean Squared Error (RMSE):

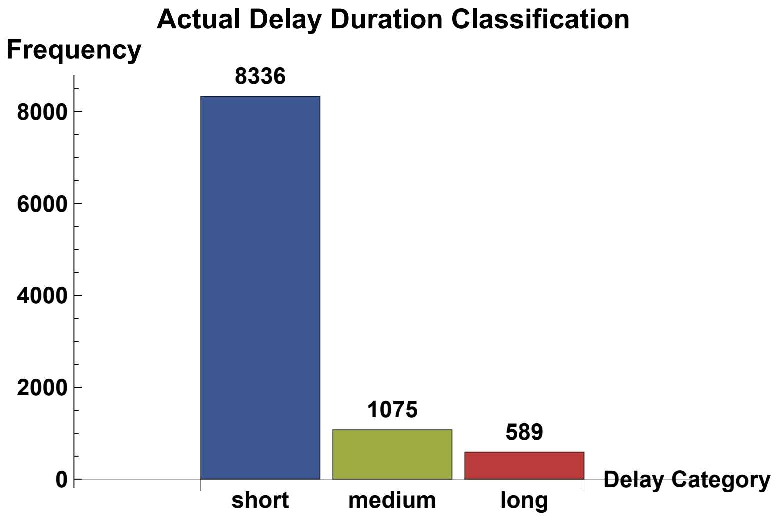
- The RMSE for delay duration prediction is 26.85 minutes, which indicates the average deviation of the predicted delay from the actual delay. Lower RMSE means more accurate predictions.

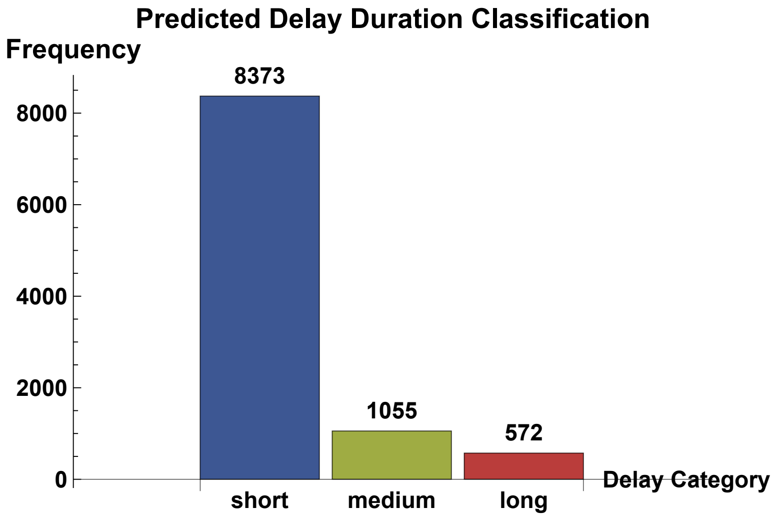
```
In[ ]:=
```

b. Analysis and Business Logic

```
In[ ]:= (*Count the frequencies of each category*)
actualCounts = Counts[actualCategories];
predictedCounts = Counts[predictedCategories];

(*Visualization of actual delay categories with numerical labels on bars*)
actualChart =
  BarChart[Values[actualCounts], ChartLabels → Placed[Keys[actualCounts], Below], ChartStyle → "DarkRainbow",
   PlotLabel → "Actual Delay Duration Classification", AxesLabel → {"Delay Category", "Frequency"},
   TicksStyle → Directive[FontSize → 14, Bold], (*Make X-axis ticks larger and bold*)
   AxesStyle → Directive[FontSize → 14], (*Adjust font size of the axis labels*)
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → 400, LabelingFunction → Above];

(*Visualization of predicted delay categories with numerical labels on bars*)
predictedChart =
  BarChart[Values[predictedCounts], ChartLabels → Placed[Keys[predictedCounts], Below], ChartStyle → "DarkRainbow",
   PlotLabel → "Predicted Delay Duration Classification", AxesLabel → {"Delay Category", "Frequency"},
   TicksStyle → Directive[FontSize → 14, Bold], AxesStyle → Directive[FontSize → 14],
   LabelStyle → Directive[FontSize → 12, Bold], ImageSize → 400, LabelingFunction → Above];

(*Display the charts side by side*)
Row[{actualChart, predictedChart}, Spacer[30]]
```

Out[•]=





Purpose: The bar chart, titled "**Comparison of Original vs. Predicted Delay Durations**", highlights the actual versus predicted delay durations for various flights after running a random forest model for predicting delays and cancellations. It provides insights into the accuracy of the model and helps identify areas for improvement in delay duration predictions.

Y-Axis:

- Label: “Frequency”
- Represents: The number of flights within each delay duration category.
- Higher bars: Indicate a higher frequency of delays in that category.

X-Axis:

- Label: “Delay Category”
- Represents: Different delay duration categories: short, medium, and long.

Insights:

1. Short Delay Durations:
   - Actual: 8336 flights.
   - Predicted: 8373 flights.
   - The model slightly overestimates the short delay durations.
2. Medium Delay Durations:
   - Actual: 1075 flights.
   - Predicted: 1055 flights.
   - The model slightly underestimates the medium delay durations.
3. Long Delay Durations:
   - Actual: 589 flights.
   - Predicted: 572 flights.
   - The model slightly underestimates the long delay durations.
4. Model Performance:
   - Accuracy: The model predictions closely match the actual delay durations, indicating a robust predictive algorithm.
   - Discrepancies: Minor discrepancies in the short and long delay categories suggest areas for potential fine-tuning.

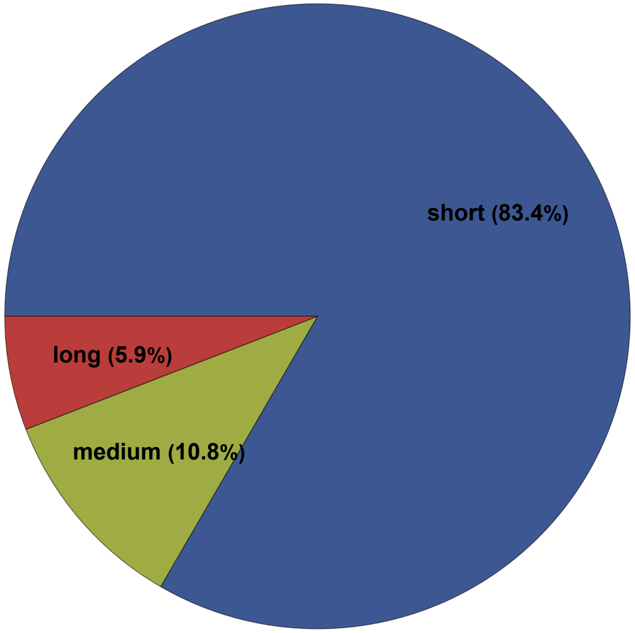
Key Takeaways:

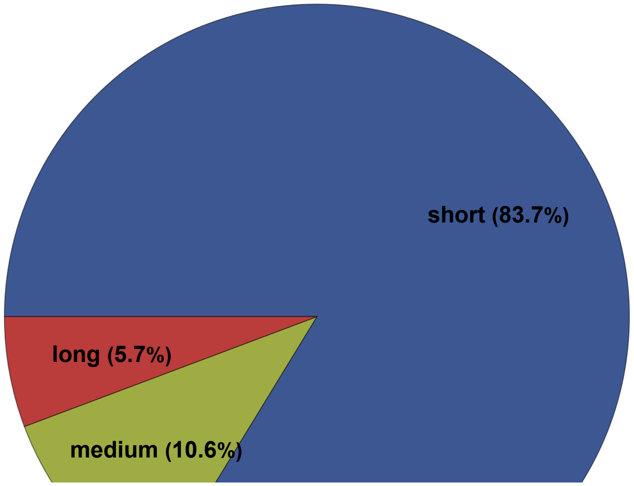
- Model Accuracy: The random forest model provides accurate predictions for delay durations, reflecting a well-tuned algorithm.
- Operational Focus: Consistent accuracy in prediction can aid in better resource allocation and operational planning for airlines.
- Passenger Impact: Understanding delay patterns can assist passengers in planning their travel better, potentially opting for flights with lower predicted delays.

```
In[*]:= (*Count the frequencies of each category*)actualCounts = Counts[actualCategories];
        predictedCounts = Counts[predictedCategories];

        (*Calculate the total counts to get percentages*)
        totalActual = Total[Values[actualCounts]];
        totalPredicted = Total[Values[predictedCounts]];

        (*Calculate percentages for actual and predicted categories*)
        actualPercentages = (Values[actualCounts] / totalActual) * 100;
        predictedPercentages = (Values[predictedCounts] / totalPredicted) * 100;

        (*Create labels with category names and percentage values for actual categories*)
        actualLabels = MapThread[Function[{category, percentage},
             category <> "  (" <> ToString[Round[percentage, 0.1]] <> "%)"], {Keys[actualCounts], actualPercentages}];

        (*Create labels with category names and percentage values for predicted categories*)
        predictedLabels =
          MapThread[Function[{category, percentage}, category <> "  (" <> ToString[Round[percentage, 0.1]] <> "%)"],
           {Keys[predictedCounts], predictedPercentages}];

        (*Create pie chart for actual delay categories*)
        actualPieChart =
          PieChart[actualPercentages, ChartLabels → actualLabels, PlotLabel → "Actual Delay Duration Classification",
           LabelStyle → Directive[FontSize → 12, Bold], ImageSize → 400, ChartStyle → "DarkRainbow"];

        (*Create pie chart for predicted delay categories*)
        predictedPieChart =
          PieChart[predictedPercentages, ChartLabels → predictedLabels, PlotLabel → "Predicted Delay Duration Classification",
           LabelStyle → Directive[FontSize → 12, Bold], ImageSize → 400, ChartStyle → "DarkRainbow"];

        (*Display the pie charts side by side*)
        Row[{actualPieChart, predictedPieChart}, Spacer[30]]
```

Out[*]=

Actual Delay Duration Classification




Predicted Delay Duration Classification

Purpose: The pie charts, titled "**Actual Delay Duration Classification**" and "**Predicted Delay Duration Classification**", highlight the actual versus predicted delay durations for various flights after running a random forest model for predicting delays and cancellations. They provide insights into the accuracy of the model and help identify areas for improvement in delay duration predictions.

Segments:

Actual Delay Duration Classification:

- Short: 83.4%
- Medium: 10.8%
- Long: 5.9%

Predicted Delay Duration Classification:

- Short: 83.7%
- Medium: 10.6%
- Long: 5.7%

Insights:

1. Short Delay Durations:
    - Actual: 83.4%
    - Predicted: 83.7%
    - The model slightly overestimates the short delay durations.
2. Medium Delay Durations:
    - Actual: 10.8%
    - Predicted: 10.6%
    - The model slightly underestimates the medium delay durations.
3. Long Delay Durations:
    - Actual: 5.9%
    - Predicted: 5.7%
    - The model slightly underestimates the long delay durations.
4. Model Performance:

- Accuracy: The model predictions closely match the actual delay durations, indicating a robust predictive algorithm.
- Discrepancies: Minor discrepancies in the short and long delay categories suggest areas for potential fine-tuning.

Key Takeaways:

- Model Accuracy: The random forest model provides accurate predictions for delay durations, reflecting a well-tuned algorithm.
- Operational Focus: Consistent accuracy in prediction can aid in better resource allocation and operational planning for airlines.
- Passenger Impact: Understanding delay patterns can assist passengers in planning their travel better, potentially opting for flights with lower predicted delays.

```
In[∘]:=
```

c. Actionable Insights

Based on the model outcomes, we propose the following recommendations to reduce flight delays:

1. Short Delay Durations:

*** Optimize Turnaround Times: Improve gate operations (boarding, deplaning) to reduce short delays.**

*** Schedule Buffer Times: Add small buffer windows between flights to accommodate minor delays without impacting subsequent departures.**

2. Medium Delay Durations:

*** Proactive Resource Allocation: Ensure enough ground crew and equipment are available to handle moderate delays.**

*** Manage Air Traffic & Weather: Monitor weather conditions and air traffic to anticipate and mitigate delays.**

3. Long Delay Durations:

*** Improve Communication: Provide timely updates to passengers during long delays to manage expectations and reduce frustration.**

*** Contingency Plans: Develop strategies like re-routing flights or offering alternate flights to minimize the impact of extended delays.**

4. General Operational Focus:

*** Efficient Gate Operations: Streamline boarding, security checks, and baggage handling to reduce the chances of delays, particularly for short and medium durations.**

*** Better Flight Scheduling: Avoid tight schedules and plan for realistic turnaround times to ensure smoother operations.**

*** Airline-Weather Coordination: Integrate real-time weather data to adjust flight plans and prevent weather-related disruptions.**

In[ ]:=