Development and evaluation of methods to visualise Airlines Delays using various visualization techniques.

Sapna Gupta

X14115824
13th Dec 2015.

Msc Data Analytics Year (2014-2016)

Abstract

Data is free but the information hidden inside this data is priceless. Various visualization techniques are applied on a massively huge Airlines dataset to answer various business queries concerning flight delays like cause of delay and forecasting future delays. Dataset used for this analysis consist of flight transactions, flight routes information with around 7 million records and 2 GB in size. A picture is worth thousand words and reading a picture can trigger a number of thoughts such as extracting, comparing or aggregating numerical values. Most of the charts that are presented in this report will trigger some of these thoughts. Results are represented in the form of dashboard and story. There is also a demonstration of comparative analysis using different types of technologies.

Contents

Introduction

The Federal Aviation Administration (FAA) considers a flight to be delayed when it is 15 minutes later than its scheduled time. A cancellation occurs when the airline does not operate the flight at all for a certain reason. When flights are cancelled or delayed, passengers may be entitled to compensation due to rules obeyed by every flight company, usually Rule 240, or Rule 218 in certain locations. This rule usually specifies that passengers may be entitled to certain reimbursements, including a free room if the next flight is the day after the cancelled one, a choice of reimbursement, rerouting, phone calls, and refreshments.

Flight delays are an inconvenience to passengers. A delayed flight can be costly to passengers by making them late to their personal scheduled events. A passenger who is delayed on a multi-plane trip could miss a connecting flight. Anger and frustration can occur in delayed passengers. In the United States, passengers are not entitled to compensation when a delay occurs, not even a cut of fees airlines must pay federal authorities for long delays. Airlines are required to pay for lodging costs of passengers if the delay or a cancellation is through their own fault, but not if the cause is beyond their control, such as weather. So every passenger is very careful while choosing the right Airlines, right time and right destination to fly. Aviation industry is always interested in finding the cause of delays and work very closely with airline industry in improving the services which causes flight delays.

This project is to create visual representation of historical flight data to help passenger to choose best airlines, good destinations and better time to fly. Second type of visualization is created for business users in the aviation industry who wants to know more than just top airlines and good destinations. Third type of visual representation is created for industry experts who have some or more statistical knowledge.

Project Outline

Keeping three types of audience in mind, 3 different types of dashboards are created.

1. ✈ Passengers while booking their flights need to know best airlines to choose from, good time to fly and which are the busy airports can be avoided. Passengers usually don’t have much time and interest in any kind of interaction with the dashboards. They need all the information in less than 30 sec. So a static dash is required with all the information of the interest of a passenger.

2. ✈ Business users have more time than passenger and more interested to find insights from the data present in the company. But not every business users have statistical knowledge, so interactive dashboard is required where users can drill down the problem and find trends etc.

3. ✈ Third type of dashboard cater to industry experts who have statistical knowledge and are interested in finding the relationships between two or more parameters which are causing flight delays, for these type of people time is not an issue.

Dataset Used

In this project, we use publicly available data originally from the Bureau of Transportation Statistics to analyse and predict flight departure delays for a subset of commercial flights in the United States. The original dataset contains information for all commercial flights in the US from 1987 to 2008. Since the data set is extremely large (several million records) we selected only one years: 2008 of data. This dataset has 29 variables. Variable highlighted with colour are used for building dashboards.

Variable descriptions

	Name	**Description**
1	Year	2008
2	Month	01-12
3	DayofMonth	01-31
4	DayOfWeek	1 (Monday) - 7 (Sunday)
5	DepTime	actual departure time (local, hhmm)
6	CRSDepTime	scheduled departure time (local, hhmm)
7	ArrTime	actual arrival time (local, hhmm)
8	CRSArrTime	scheduled arrival time (local, hhmm)
9	UniqueCarrier	unique carrier code
10	FlightNum	flight number
11	TailNum	plane tail number
12	ActualElapsedTime	in minutes
13	CRSElapsedTime	in minutes
14	AirTime	in minutes
15	ArrDelay	arrival delay, in minutes
16	DepDelay	departure delay, in minutes
17	Origin	origin IATA airport code
18	Dest	destination IATA airport code
19	Distance	in miles
20	TaxiIn	taxi in time, in minutes
21	TaxiOut	taxi out time in minutes
22	Cancelled	was the flight cancelled?

23	CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24	Diverted	1 = yes, 0 = no
25	CarrierDelay	in minutes
26	WeatherDelay	in minutes
27	NASDelay	in minutes
28	SecurityDelay	in minutes
29	LateAircraftDelay	in minutes

This dataset is a good mix of categorical, discrete, date, continuous and range variables.

Catagorical Variables: UniqueCarrier , Origin , Dest , CancellationCode

Discreate Variables: DayofMonth , DayOfWeek , Year , Month , Cancelled , Diverted, FlightNum

DayofMonth are **aliased** with the month names.

DayOfWeek are **aliased** with the week names.

Continouse Variables: Distance . ArrDelay , DepDelay , LateAircraftDelay , SecurityDelay , NASDelay , WeatherDelay , AirTime

Date variable is **calculated field** is formed by concatenating DayofMonth, Month and Year field in the dataset.

CONCAT(CONCAT(CONCAT(CONCAT(DayofMonth,'-'), Month),'-'),Year) AS FLIGHT_DATE

Range Variables is also **calculated field** is created using Tableau bin functionality. Size of the bin can be entered dynamically using parameter.

Choose Distance Bin Size
91

Tools used

1. HDFS is used to host the csv file from the web.
2. HIVE is used to load the csv file into tables and segment this data into different tables. Hive was used to leverage Hadoop MapReduce parallel processing capability in order process fast queries on this massively large dataset.
3. Tableau for visual representation.
4. R was used for statistical analysis. RHIVE package used to lift data from HIVE data warehouse into R server. – Although R server crashed many times due to the size of the dataset loaded.
5. Tableau with R is used to do predictive analysis.

Data Processing Steps

Date is downloaded as csv file from Bureau of Transportation Statistics(http://stat-computing.org/dataexpo/2009/the-data.html). This csv file if loaded into Hadoop distributed file system HDFS initially. Then the data was loaded into HIVE, where explorative analysis was done on the acquired dataset. This dataset has 7009728 records, analysing all these 7 million records in any dashboard can lead to saviour performance issue on the dashboard. It was decided to segment this data into small tables based on the insights found in this dataset. HIVE was chosen for explorative analysis. Results of explorative analysis are discussed below and code and detailed results can be found in the Appendix section.

17265 flights were diverted which were only 0.25 % of the whole dataset, so these were removed from the selected records.

137434 flights were cancelled, these were only 1.96 % of the whole dataset. A separate table called **FLIGHTS_CANCELLED** was created to analyse these records separately. Only relevant attributed were selected for this table creation.

```
CREATE TABLE FLIGHTS_CANCELLED AS SELECT
CONCAT(CONCAT(CONCAT(CONCAT(DAY,'-'),MONTH),'-'),YEAR) AS
FLIGHT_DATE , day, MONTH , DAY_OF_WEEK , unique_carrier , ORIGIN , DEST ,
cancellation_code FROM flight_data WHERE CANCELLED = 1;
```

5388838 Flights were **on time**, which were the largest portion (76.88 %) of the whole dataset, these were separated into a table called **FLIGHTS_ON_TIME**.

```
CREATE TABLE FLIGHTS_ON_TIME AS SELECT
CONCAT(CONCAT(CONCAT(CONCAT(DAY,'-'),MONTH),'-'),YEAR) AS
FLIGHT_DATE , day, MONTH , DAY_OF_WEEK , unique_carrier , ORIGIN , DEST  ,
ARR_DELAY , DEP_DELAY FROM flight_data WHERE arr_delay <= 15  AND arr_delay
IS NOT NULL;
```

1466191 Flights were **actually delayed** in the whole year these were 20.92 % of the whole dataset. These were separated into a table named **FLIGHTS_DELAYED**.

```
CREATE TABLE FLIGHTS_DELAYED AS SELECT
CONCAT(CONCAT(CONCAT(CONCAT(DAY,'-'),MONTH),'-'),YEAR) AS
FLIGHT_DATE ,day, MONTH , DAY_OF_WEEK , unique_carrier , ORIGIN , DEST  ,
ARR_DELAY , DEP_DELAY, dep_time , arr_time ,air_time, carrier_delay ,weather_delay,
nas_delay, security_delay, late_aircraft_delay  FROM flight_data WHERE arr_delay > 15;
```

These tables were loaded into R using RHIVE package for further statistical analysis.

These tables were also loaded into Tableau using MapR ODBC connection for creating interactive visual representations. Fig. 1 shows a data flow pipelines created for this project.

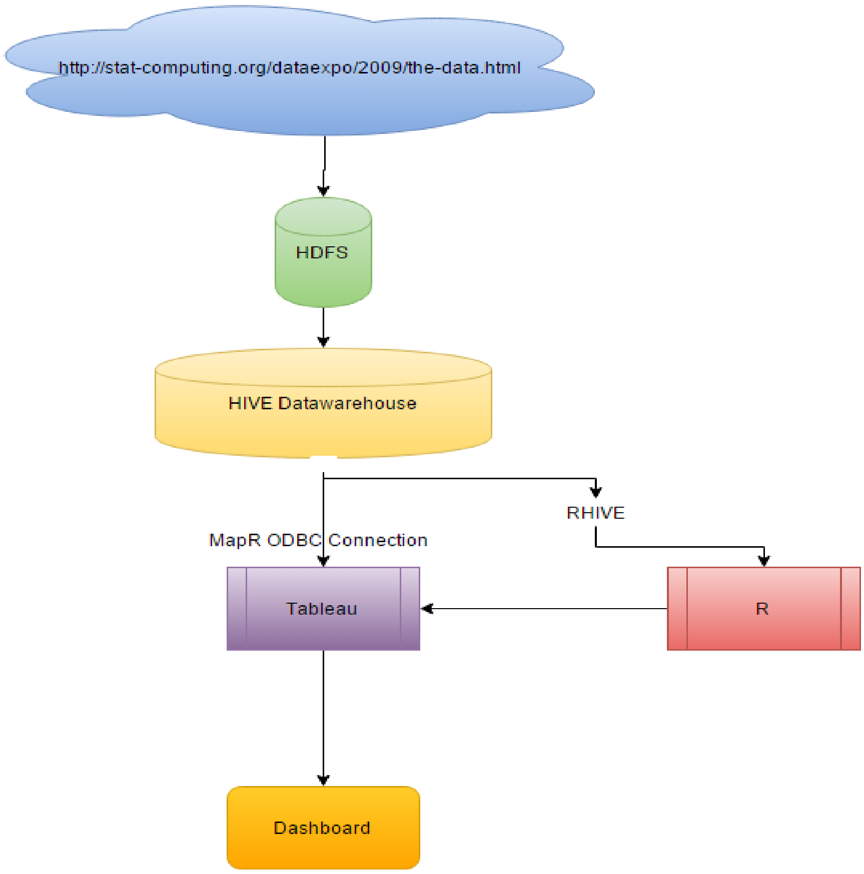


Figure: 1, Data flow diagram of ✈ dataset

Related Work

Before starting designing dashboards, an online survey was carried out to find best practices to build effective dashboards. During this survey we found some interesting golden rules to follow while designing dashboard [4], these are:

1. **Prioritize through positioning:** A dashboard displays data through charts and gauges, but not all data is equally important. Certain data might be extremely important and it might be that the rest of the information that is displayed on the dashboard can only be comprehended if this data is analysed first. Under such circumstances a dashboard designer must position the important chart to the **top-left corner** of the dashboard, since this region receives the preliminary **attention** of a viewer (**F-Shaped Pattern**).
2. **Facilitate comparative analysis:** In case you have two charts that are meant to be contrasted, then it is best to arrange them side by side. This arrangement signifies the need for comparison.
3. **Customize chart scale for optimal data presentation:** Sometimes the data displayed on a chart has a very narrow range. This makes the task of analysing the data a lot difficult. Such situations call for manipulation of the chart scale. The chart scale should be adjusted so that, its lower limit and upper limit are close to the lower and upper

limit of the data range. This adjustment will help in accentuating the ups and downs of the plotted data, thereby making analysis easier.

4. **Appropriate selection of charts:** For maximum impact, it is essential that you choose the right chart for you data. The pie chart is often used inappropriately. A pie chart is actually meant for plotting percentages but, it is sometimes used for plotting non-percentage data such as sales, revenue, quantity etc.
5. **Proper formatting of numbers:** It doesn't make sense to have a chart that displays numbers with unnecessary accuracy. If the chart is cluttered with very large numbers. So, it is best to restrict the number of decimal places to 1 or 2. And, scale large numbers by defining a proper scaling parameter. The K,M scale can be applied to financial charts to scale down numbers which are greater than thousand and million.
6. **Data-Ink ratio**: The Data-Ink ratio is a concept introduced by Edward Tufte in 1983. The data-ink ratio is the proportion of Ink that is used to present actual data compared to the total amount of ink (or pixels) used in the entire display. Good graphics should include only data-Ink. Non-Data-Ink is to be deleted everywhere where possible. The reason for this is to avoid drawing the attention of viewers of the data presentation to irrelevant elements. The goal is to design a display with the highest possible data-ink ratio without eliminating something that is necessary for effective communication [2].
7. **Chartjunk:** The term chartjunk was also coined by Edward Tufte in 1983. Chartjunk is a term for unnecessary or confusing visual elements in charts and graphs. Markings and visual elements can be called chartjunk if they are not part of the minimum set of visuals necessary to communicate the information understandably. Examples of unnecessary elements which might be called chartjunk include heavy or dark grid lines, ornamented chart axes and display frames, pictures or icons within data graphs, and ornamental shading [3].
8. **Overlay Types:** Kong & Agrawal in 2012 introduce graphical overlays—visual elements that are layered onto charts to facilitate a larger set of chart reading tasks. These overlays directly support the lower-level perceptual and cognitive processes that viewers must perform to read a chart. Five main types of overlays are (1) reference structures such as **gridlines**, (2) **highlights** such as outlines around important marks, (3) redundant encodings such as numerical data **labels**, (4) summary statistics such as the **mean or max** and (5) **annotations** such as descriptive text for context [1].

Implementation of visual representations

Flight cancelation analysis

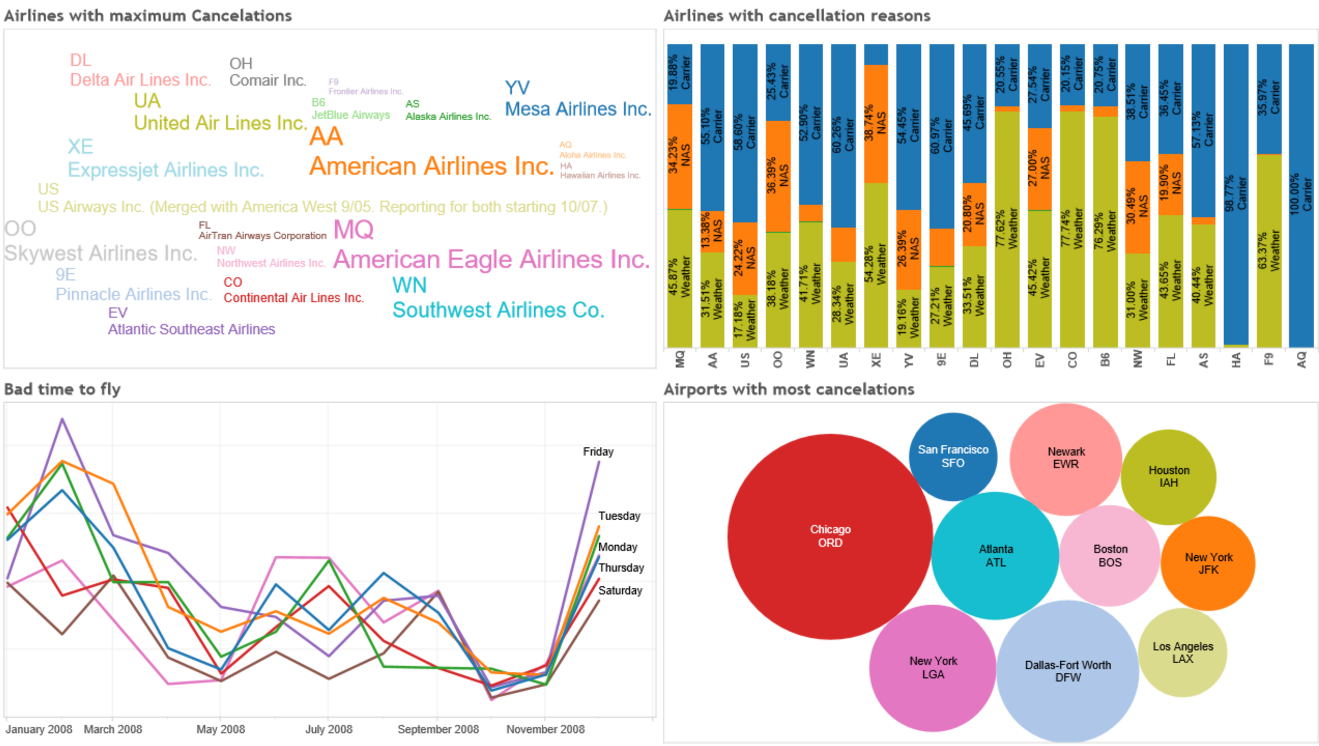


Fig: 1 Dashboard with flights cancelation analysis

Purpose of the Dashboard: Flight cancelation is serious concern, which can lead to a big lose in terms of money and time. If there is any cancelation then reason of cancellation is also very important. Goal of these dashboards is to answer some of the common questions in less than 30 sec. These questions are:

Which Airport had maximum cancelations in one year?
Which Airlines had maximum cancelations in one year?
When were the most flight cancelations in one year?
What were the common reasons of the cancellations?

Target Audience: Dashboards are designed for common public, who intend to book flight for their next travel. These people don't have much time to interact with the dashboards, so no interactivity is provided.

	Variables Used	Description	Type	Used As
1	FlightDate	Date	Continuous	Axis in timeline Chart
2	UniqueCarrier	unique carrier code	Categorical	Colour hue , Mark
3	Origin	origin IATA airport code	Categorical	Colour hue
4	Cancelled	was the flight cancelled?	Discrete	Filter
5	CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)	Categorical	Colour in the Bars

6	Number of Flights	Number	Continuous	Size of Bouble , Size of Words
7	DayOfWeek	1 (Monday) - 7 (Sunday)	Discrete	Colour of Lines (hue)

Results: American eagle airlines had maximum cancelations and these cancelation were due to bad weather and carrier delay. Avoid AQ completely which had all the delays due to bad carrier performance. Friday is not a good day to fly. There was a peek in flights cancelation in the month of Feb 2008 and in Dec 2008. Chicago (ORD) and NY (LGA) had maximum cancelations.

Prioritize through positioning: Word cloud chart is placed on top left corner, representing very important information like which airlines had maximum. Word cloud chart is used as it is the modest way to represent any information, these charts are easy to read and understand.

Facilitate comparative analysis: These two dashboards are places next to each other in the

final story points, as they are built for same type of audience and represent contrast in the dataset. Which airlines had maximum cancelation and which airlines had good performance in terms of on time arrivals.

Customize chart scale for optimal data presentation: Chart scales are omitted as they are not relevant to the information presented in the charts.

Appropriate selection of charts: Charts used in the two dashboards in fig 1 and fig2 were selected to answer basic question asked by any traveller with no interactivity.

Proper formatting of numbers: Number formatting was used in Top Right chart in dashboard 1 to show percentage of cancellation, this was used to represent clear understanding of amount of cancelation and to avoid clutter caused by using the big real numbers.

Data-Ink ratio: Data to ink ratio is very well kept in mind while designing all the charts except for chart number 2 in dashboard 1. This was done to represent verity on the dashboards.

Chartjunk: Chartjunks can be seen in 100% bar chart. This is to give some attractive looks to the dashboards.

Overlay types used: Highlighting key information using bright colours is the only overlay techniques used. Size of the bubble in bubble chart and size of the word in the word cloud is used to represent number of flights under each category.

There is no interactivity on this dashboards so these can be placed on the company web page or newspaper articles.

On time Flights analysis

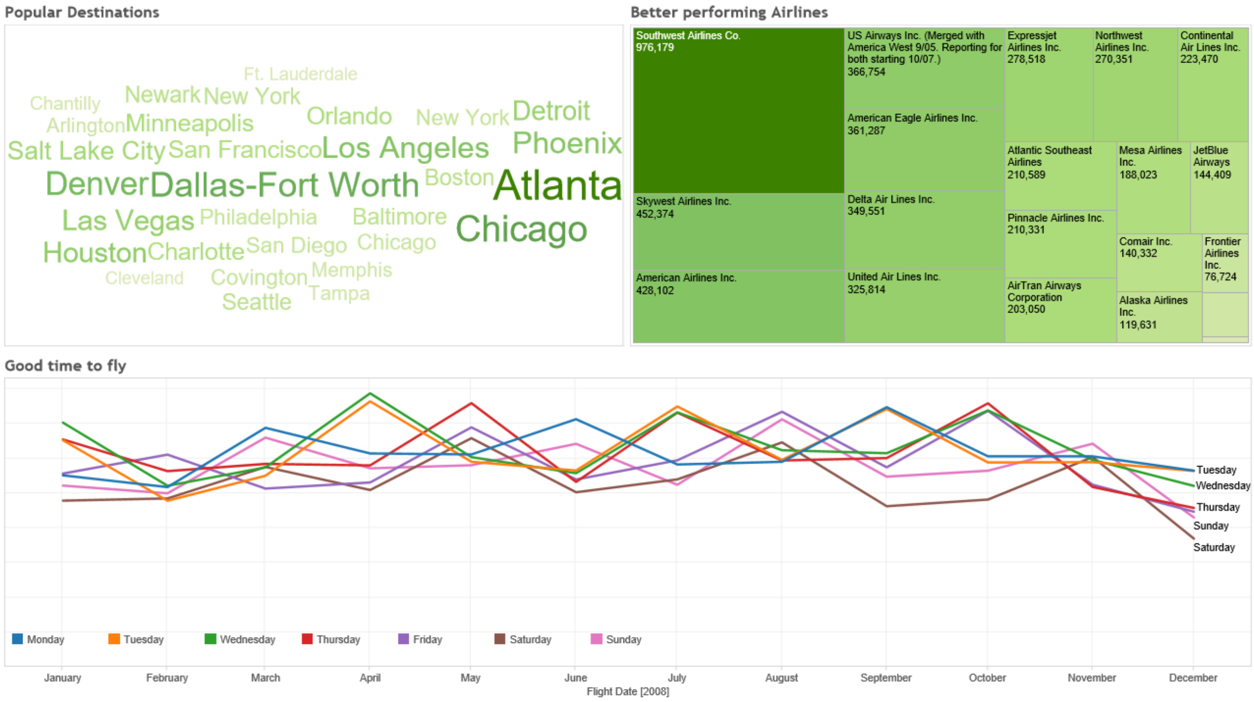


Fig: 2 Dashboard with flights on-time analysis

Purpose: While booking your flight, we are always concerned of few factors in case to make our travel peaceful. These are choice of good airlines, better airport, and good time to fly. Goal of these dashboards was to answer some of the common questions in less than 30 sec. These questions are,

Which airline is the biggest and had flight arrivals on time?
Which airport performed better in terms on-time arrivals?
What is the good time to fly in the whole year?

Target Audience: Dashboards are designed for common public, who intend to book flight for their next travel. These people don't have much time to interact with the dashboards, so interactivity is provided.

	Variables Used	Description	Type	Used As
1	FlightDate	Date	Continuous	Axis in time Series Chart
2	UniqueCarrier	[unique carrier code]	Categorical	Colour , Mark
3	Origin	[origin IATA airport code]	Categorical	Colour
4	Number of Flights	Number	Continuous	Size of Words , Size of Box , Colour of the Words from light to Dark (hue , satura-tion)

5	DayOfWeek	1 (Monday) - 7 (Sun-day)	Discrete	Colour of Lines

Results: Prefer Southwest airlines with 18% of market share of on line flights. Atlanta airport has most number of on-time flights. Fly in April, May, and September for uninterrupted travel.

Prioritize through positioning: Word cloud chart is placed on top left corner, representing very important information like which airport had maximum on time flight arrivals. These charts are easy to read and understand. Line chart need long space so it is placed at the bottom.

Facilitate comparative analysis: These two dashboards are places next to each other in the final story points, as they are built for same type of audience and represent contrast in the dataset. Which airlines had maximum cancelation and which airlines had good performance in terms of on time arrivals.

Customize chart scale for optimal data presentation: Chart scales are omitted as they are not relevant to the information presented in the charts.

Appropriate selection of charts: Charts used in Fig. 2 were selected to answer basic question asked by any traveller while booking next flight.

Proper formatting of numbers: Number formatting was used in Top Right corner in dashboard 2 to show percentage of market share of each airlines.

Data-Ink ratio: Data to ink ratio is very well kept in mind while designing all the charts except for the tree map chart on top right corner of the dashboard 2. This was done to represent verity on the dashboards.

Chartjunk: Chartjunks can be seen Tree Map in dashboard 2. This is due to give some attractive looks to the dashboards.

Overlay types used: Highlighting key information using bright colours is the only overlay techniques used. Size of the box in tree map chart and size of the word in the word cloud is used to represent number of flights under each category.

There is no interactivity on this dashboards so these can be placed on the company web page or newspaper articles.

Flight delay analysis

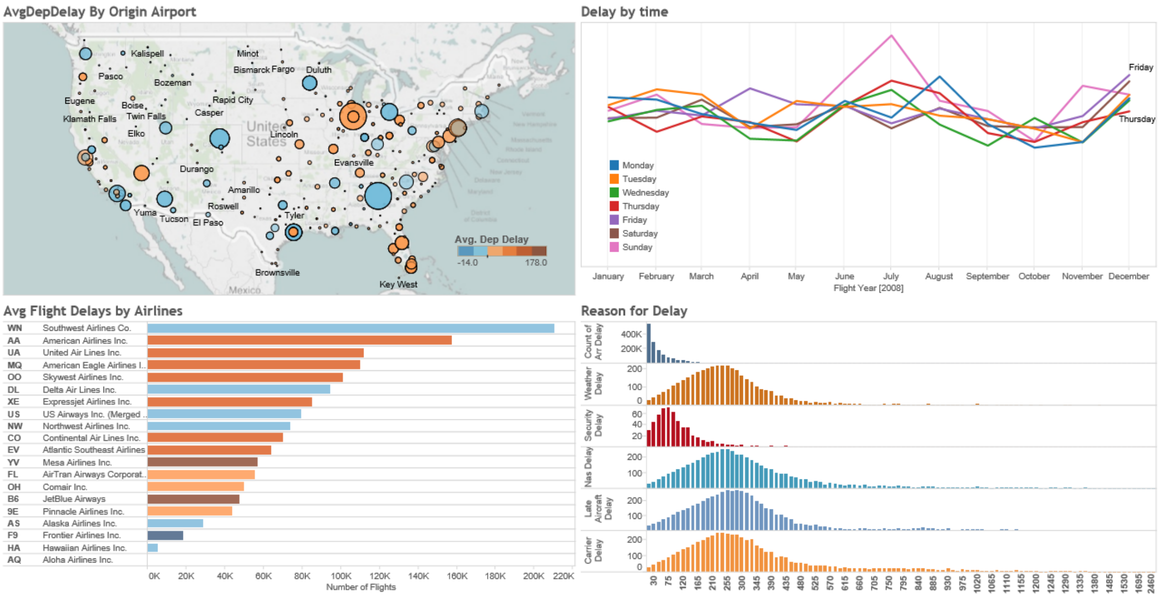


Fig: 3 Dashboard with flights delay analysis

Purpose of the Dashboard: These dashboards were created to answer questions which airport had maximum delays, what time of the year there were maximum delays, what were the various reasons of the delay, which airlines had maximum delays.

Target Audience: Dash boards in Fig. 3 is build keeping business users in mind who wants to know more than just 4 basic questions. These people wants to interact with the dashboard using filters and actions.

	Variables Used	Description	Type	Used As
1	FlightDate	Date	Continuous	Axis in time Series Chart , Position
2	UniqueCarrier	unique carrier code	Categorical	Colour in the Bars , Mark
3	Origin	origin IATA airport code	Categorical	Position on Map , Glut Due to number of Circles
4	Number of Flights	Number	Continuous	Size of Circle on Map , Size of Bars , (hue , saturation)
5	DayOfWeek	1 (Monday) - 7 (Sun-day)	Discrete	Colour of Lines
6	ArrDelay	arrival delay, in minutes	Continuous	Colour of Bar and Colour of Circles on Map
7	CarrierDelay	in minutes	Continuous	Distribution in Histogram , Colour
8	WeatherDelay	in minutes	Continuous	Distribution in Histogram , Colour
9	NASDelay	in minutes	Continuous	Distribution in Histogram , Colour
10	SecurityDelay	in minutes	Continuous	Distribution in Histogram , Colour
11	LateAircraftDelay	in minutes	Continuous	Distribution in Histogram , Colour

Results Southwest airline having most number of flights. Mesa airlines and JetBlue airlines with most delays. Chicago airport and New York (JFK), Las Vegas, Sen Francisco, Orlando with most delays, it is obvious as these are most favourite tourist places. There is a peak on 1st july 2008, that could be due to first day of summer holidays in schools. Friday has largest probability of delayed flights. December, June and July have largest percentage of delayed flights. Security delay is the main cause of most of the delays.

Prioritize through positioning: US map is used in the first chart in the dashboard 1, as it is easy to relate with human perception and is placed on the top left corner. This makes the whole dashboard very interesting. After looking at this chart first user can easily understand the problem discussed on high level. It creates interest in user to read other charts to know more about the main problem.

Appropriate selection of charts: Careful selection of chart was made keeping business users as audience and also show verity at the same time. Map is selected to show relative position of the airports. Histogram is used to show distribution of various delays.

Proper formatting of numbers: Number are not used to save space on the charts.

Data-Ink ratio: Data to ink ratio is maintained.

Chartjunk: Chartjunks are kept under control.

Overlay types used: Dim grid lines are used where ever require, colour is used to differentiate between different airlines and airports. Size is used to sow different number of flights.

Interactivity: Filter other charts based on the airline selected.

These dashboards are designed for company internal websites, where employees would want to interact with the dashboard to find more information.

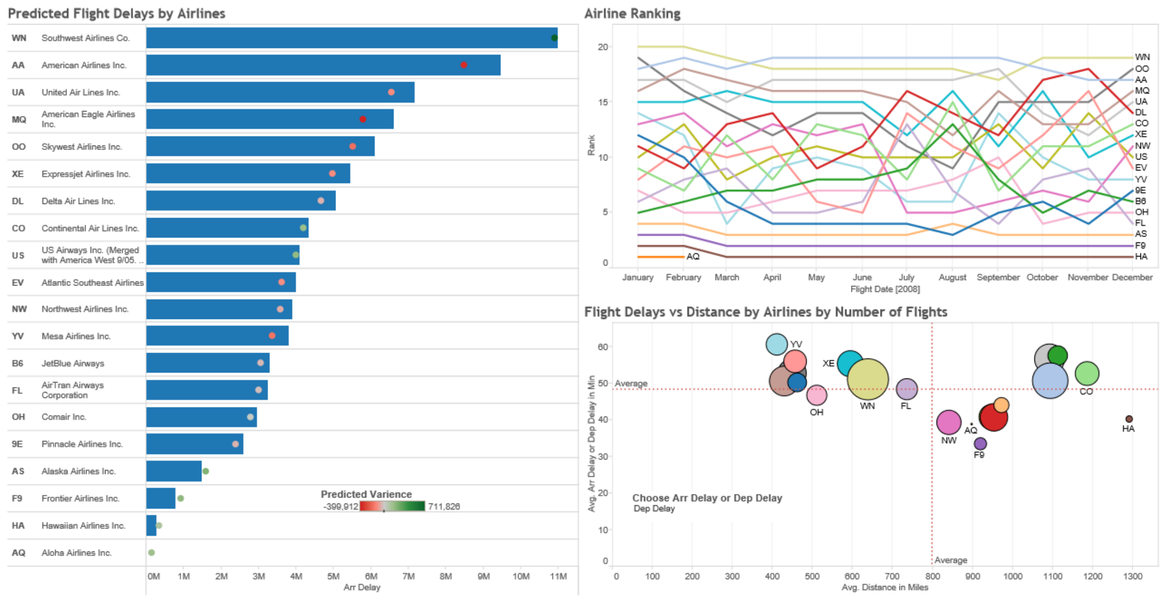


Fig: 4 Dashboard with Airlines performance and predicted flight delays.

Purpose of the Dashboard: These dashboards were created to answer questions like what is the ranking of each airlines in terms of average departure delays. To see predicted delay for each airlines and find dependency of delay based on the distance of the flight.

Target Audience: Dashboard is build keeping business users in mind who wants to know more than just 4 basic questions. These people wants to interact with the dashboard using filters and actions.

	Variables Used	Description	Type	Used As
1	FlightDate	Date	Continuous	Axis in Rank Chart , Position
2	UniqueCarrier	unique carrier code	Categorical	Colour of Circles , Colour of Rank Lines , Position on Bar Chart
3	Number of Flights	Number	Continuous	Size of Circle in scatter plot , Size of Bars ,
4	Rank	1 - 20	Order	Position in Rank Chart
5	Predicted De- lay	in Minutes	Continuous	Position and colour of circles on the bar chart (hue , saturation)
6	Distance	in miles	Continuous	Position on the Scatter Plot
7	ArrDelay	arrival delay, in minutes	Continuous	Position on the Scatter Plot
8	DepDelay	Departure delay, in minutes	Continuous	Position on the Scatter Plot

Predicted Delay was calculated using R script -- SCRIPT_REAL("fit <- lm(.arg1 ~ .arg2 + .arg3 + .arg4) fit$fitted",

SUM([Arr Delay]),

SUM([Dep Delay]),

SUM([Distance]),

SUM([Air Time])

)

Results: Overall performance of Southwest Airlines is the best and in the most popular airlines. Predicted delay for these airlines is very close the actual delays. Hawaiian Airlines are the farthest distance airlines with avg delay of 40.05 mins. Mesa Airlines have maximum delays and are short distance flights. Aloha Airlines only operated till Feb 2008. Increasing delays for small carriers, except for Aloha. Delta and US Airways are improving. Prediction of delay is very close to the actual delay.

Prioritize through positioning: Chart with Predicted delay has airlines description which is very important to understand rest of the chart, so it placed on top left corner of the dashboard.

Facilitate comparative analysis: Rank function in tableau is used to show ranking of the airlines in terms of airlines delays.

Appropriate selection of charts: Careful selection of chart was made keeping business users as audience and also show verity at the same time. Bar chart is a very good representation for comparing actual and predicted values.

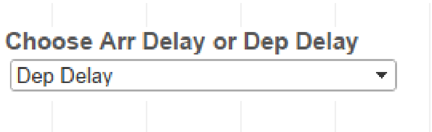
Proper formatting of numbers: Numbers are not used instead size of the circle is used to represent number of flights.

Data-Ink ratio: Data to ink ratio is maintained on each chart.

Chartjunk: Chartjunks are kept under control.

Overlay types used: Dim grid lines are used where ever require, colour is used to differentiate between different airlines. Size is used to sow different number of flights under each category (airlines).

Parameters and Filters are used to give interactivity. Parameter created to change the axis of the chart.



These dashboards are designed for company internal websites, where employees would want to interact with the dashboard to find more information.

Deep dive into flight delay cause analysis

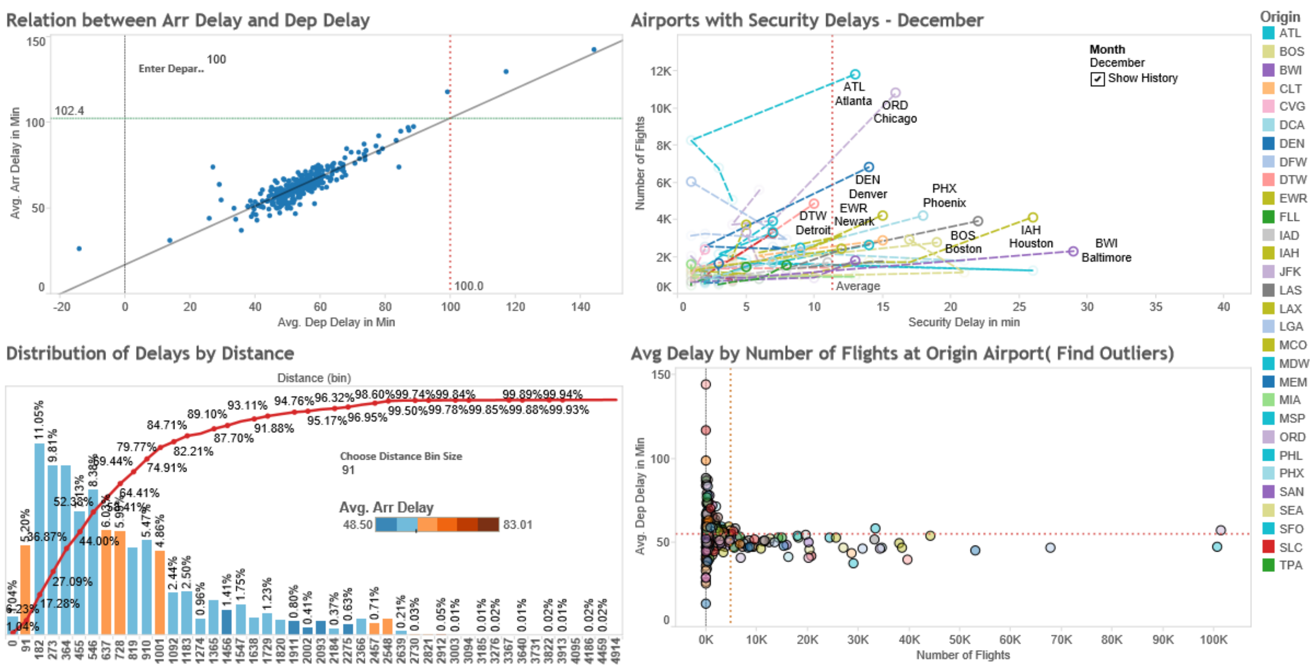


Fig: 5 Dashboard with deep dive into flights delays with a good example of chartjunk(Glut) , animation on 2 chart reduces the clarity of the data on the chart.

Purpose of the Dashboard: Dashboards in Figure 5 is designed to answer some advance questions like:

Predict future flight delays based on the past historical information. Which airports have more security delays? Does number of flights effect security delays? Is the airports facing problem in departure delays due to heavy traffic of flights on that airport?

Target Audience: Dashboards in figure 5 is build keeping industry experts in mind who are interested in finding hidden insights in the dataset. These people have statistical knowledge and understand statistical functions. These people have clear understanding of regression and scatter plots, outliers, and histograms distributions.

	Variables Used	Description	Type	Used As
1	Origin	[origin IATA airport code]	Categorical	Colour of Circles in 3 scatter plots
2	Number of Flights	Number	Continuous	Size of Bar in Histogram , Position in the Scatter plots
4	Predicted Delay	in Minutes	Continuous	Position on the Scatter Plot
5	Distance	in miles	Continuous	Position on the histogram
6	ArrDelay	arrival delay, in minutes	Continuous	Position on the Scatter Plot, Colour of Bars in Histogram (hue , saturation)
7	DepDelay	Departure delay, in minutes	Continuous	Position on the Scatter Plot
8	SecurityDelay	in minutes	Continuous	Position on the Scatter Plot
9	Running Total	in %	Continuous	Position on the histogram

Results: ATL (Newark) is the worst. ORD (Chicago O'Hare) is not good, but also has high volume. DFW (Dallas-Fort Worth) is relatively good - high traffic but relatively small delay. Weather plays a huge role in delays - any kind of precipitation, high winds, or reduced visibility increases delays (scatterplots above). Short distance flights are performing bad.

Regression analysis is performed using scatter plot. Arrival delay is closely related to departure delay with some outliers like TUP where arrival delay is negative. Delays at PUB, BJI, PIR Airport are more than expected delay, there are other factors effecting flight delays.

Trend Line on the chart is used to predict fure delays. P value < 0.0001 shows the model is good predictor. R Square is 0.809.

Used Calculated Field --

Predicted Arival Delay = (0.856761*[Enter Departure Delay]) + 16.75

Trend Lines Model

A linear trend model is computed for average of Arr Delay given average of Dep Delay. The model may be significant at p <= 0.05.

Model formula:	(Avg. Dep Delay + intercept)
Number of modeled observations:	302
Number of filtered observations:	0
Model degrees of freedom:	2
Residual degrees of freedom (DF):	300
SSE (sum squared error):	8280.84
MSE (mean squared error):	27.6028
R-Squared:	0.809527
Standard error:	5.25384
p-value (significance):	< 0.0001

Individual trend lines:

Panes		Line		Coefficients				
Row	**Column**	**p-value**	**DF**	**Term**	**Value**	**StdErr**	**t-value**	**p-value**
Arr Delay	Dep Delay	< 0.0001	300	Avg. Dep Delay	0.856761	0.0239938	35.7076	< 0.0001
				intercept	16.7593	1.348	12.4327	< 0.0001

Most of the market share is for short distance flights, delays is more for very short distance flights.

Prioritize through positioning: Linear Regression analysis in very interesting chart catching attention of all statisticians. Next chart is animation chart showing trend of delays on various airport.

Facilitate comparative analysis: charts on dashboard 6 are placed on top of each other as both compares flight delays over the same time period.

Appropriate selection of charts: Regression analysis and scatter plot goes hand in hand. Distribution and histogram is very good representation.

Proper formatting of numbers: Number formatting is used to save space on the chart.

Data-Ink ratio: Data to ink ratio is maintained.

Chartjunk: Animation and colour in scatter plot is compromise to chart glut.

Overlay Types used: Highlights, gridlines, labels and average reference lines are extensively used. Annotation is used in the second chart in dashboard 5 to draw attention onto the outliers.

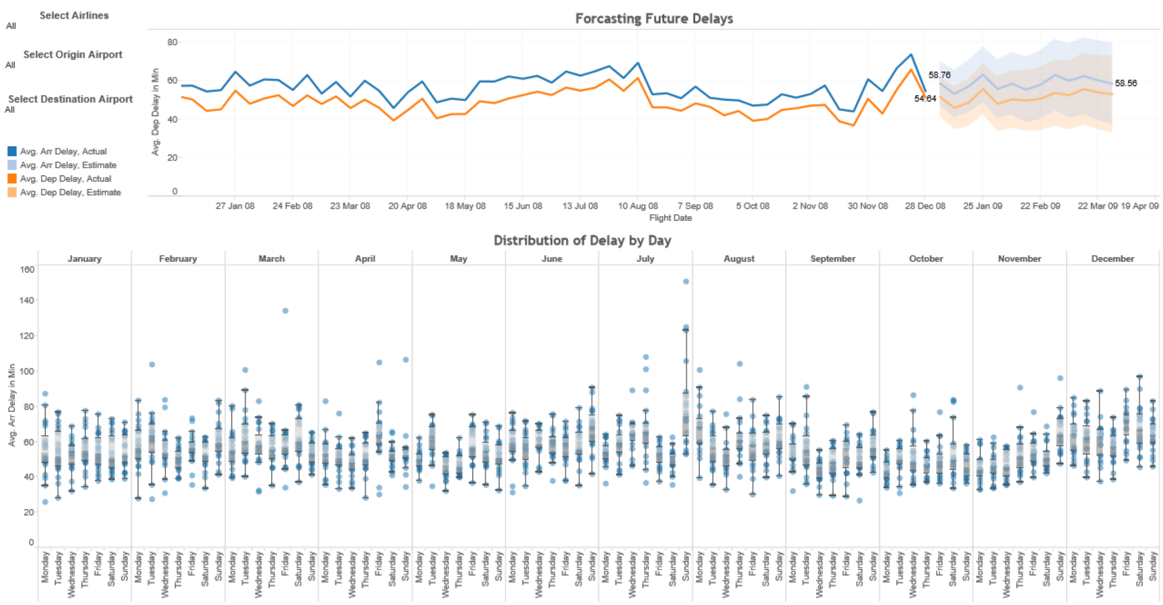


Fig: 6 Dashboard is showing trend, seasonality, forecasting and outliers in the flight dataset.

Purpose of the Dashboard: Dashboards in Figure 6 is designed to answer some advance questions like: Find daily outliers in terms of time. Find average delays during public holidays in US? What is the trend in delay from time to time? Find trends in arrival delay and departure delays and try to forecast future delays.

Target Audience: Dashboards in Figure 6 is build keeping industry experts in mind who are interested in finding hidden insights in the dataset. These people have statistical knowledge and understand statistical functions. These people have clear understanding of box plots and trend lines.

	Variables Used	Description	Type	Used As
1	FlightDate	Date	Continuous	Axis in timeline Chart , Position
2	UniqueCarrier	unique carrier code	Categorical	Filter
3	Origin	origin IATA airport code	Categorical	Filter
4	Dest	Dest IATA airport code	Categorical	Filter
5	DayOfWeek	1 (Monday) - 7 (Sunday)	Discrete	Position in Box Plot
6	Month	1-12	Discrete	Position in Box Plot
7	Dep Delay	In Mins	Continuous	Position on time series, hue

Results:

In July on Sunday we see maximum delays and with outlier AirTran airlines with avg arrival delay of 151 mins. In march HA airlines with maximum delays of around 134 mins. 21st Dec with maximum delays. There is saviour delays expected on 25th Jan 2009.

Options Used to Create Forecasts

Time series:	Week of Flight Date
Measures:	Avg. Arr Delay, Avg. Dep Delay
Forecast forward:	13 weeks (4 Jan 2009 – 29 Mar 2009)
Forecast based on:	30 Dec 2007 – 28 Dec 2008
Ignore last:	No periods ignored
Seasonal pattern:	13 week cycle

Avg. Arr Delay

Initial	Change From Initial	Seasonal Effect				Contribution		Quality
4 Jan 2009	4 Jan 2009 – 29 Mar 2009	High		Low		Trend	Season	
58.76 ± 11.31	-0.19	1 Mar 2009	4.38	11 Jan 2009	-6.01	1.2%	98.8%	Poor

Avg. Dep Delay

Initial	Change From Initial	Seasonal Effect				Contribution		Quality
4 Jan 2009	4 Jan 2009 – 29 Mar 2009	High		Low		Trend	Season	
51.77 ± 9.96	1.32	15 Mar 2009	4.51	11 Jan 2009	-6.04	1.1%	98.9%	Ok

All forecasts were computed using exponential smoothing.

Avg. Arr Delay

Model			Quality Metrics					Smoothing Coefficients		
Level	Trend	Season	RMSE	MAE	MASE	MAPE	AIC	Alpha	Beta	Gamma
Additive	Additive	Additive	5.77	4.51	0.84	8.0%	222	0.461	0.000	0.085

Avg. Dep Delay

Model			Quality Metrics					Smoothing Coefficients		
Level	Trend	Season	RMSE	MAE	MASE	MAPE	AIC	Alpha	Beta	Gamma
Additive	Additive	Additive	5.08	3.75	0.79	7.7%	208	0.500	0.000	0.055

Prioritize through positioning: Forecasting is an interesting topic, therefor is placed on top left corner of the dashboard.

Facilitate comparative analysis: charts on dashboard 6 are placed on top of each other as both compares flight delays over the same time period.

Customize chart scale for optimal data presentation:

Appropriate selection of charts: For forecasting line chart is the best option. Box plot with time gives in depth analysis.

Proper formatting of numbers: Numbers are formatted to 2 decimal places.

Data-Ink ratio: Data to ink ratio is maintained.

Chartjunk: Due to using colours in the boxplot distribution of data points is not very clear.

Overlay Types used: Colours, gridlines, labels and average reference lines are used.

Level of difficulty of these dashboards is higher are only specific type of audience, so cannot be placed in any website, these for the experts.

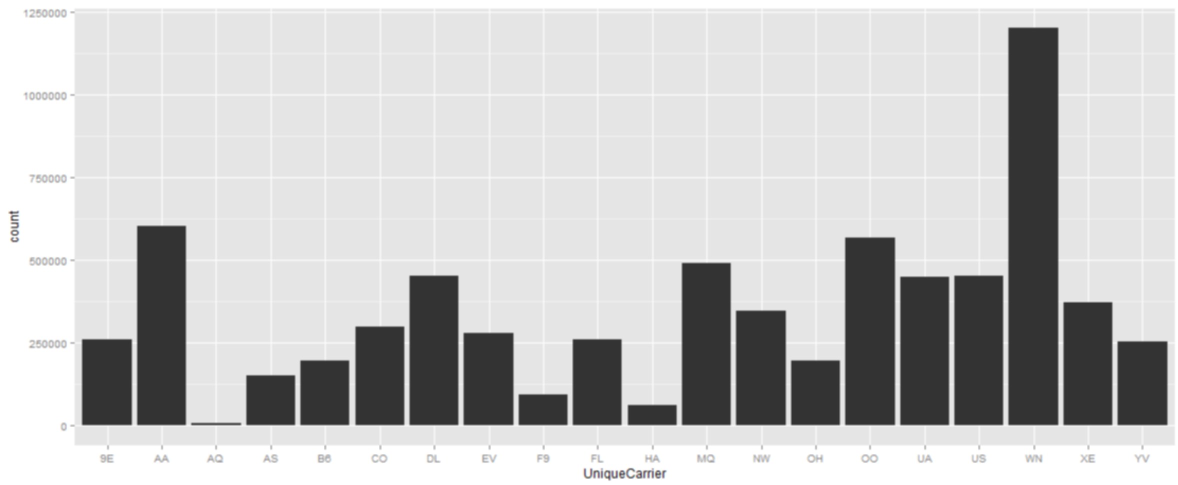
Analysis done using R

hist, plot, boxplot

basic syntax of ggplot

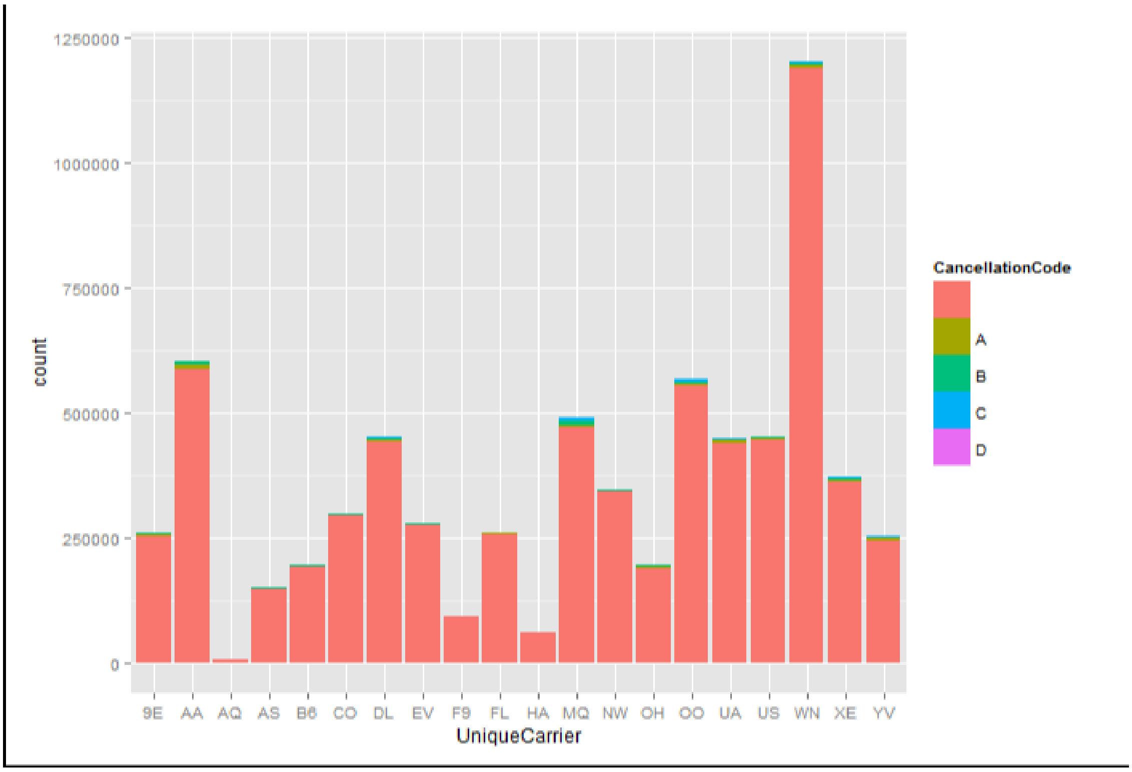
ggplot(dataframe, aesthetics = x coordinate, y coordinate, shape) + layer = geom_XYZ XYZ {bar, histogram}

ggplot(DF, aes(x = UniqueCarrier)) + geom_bar()



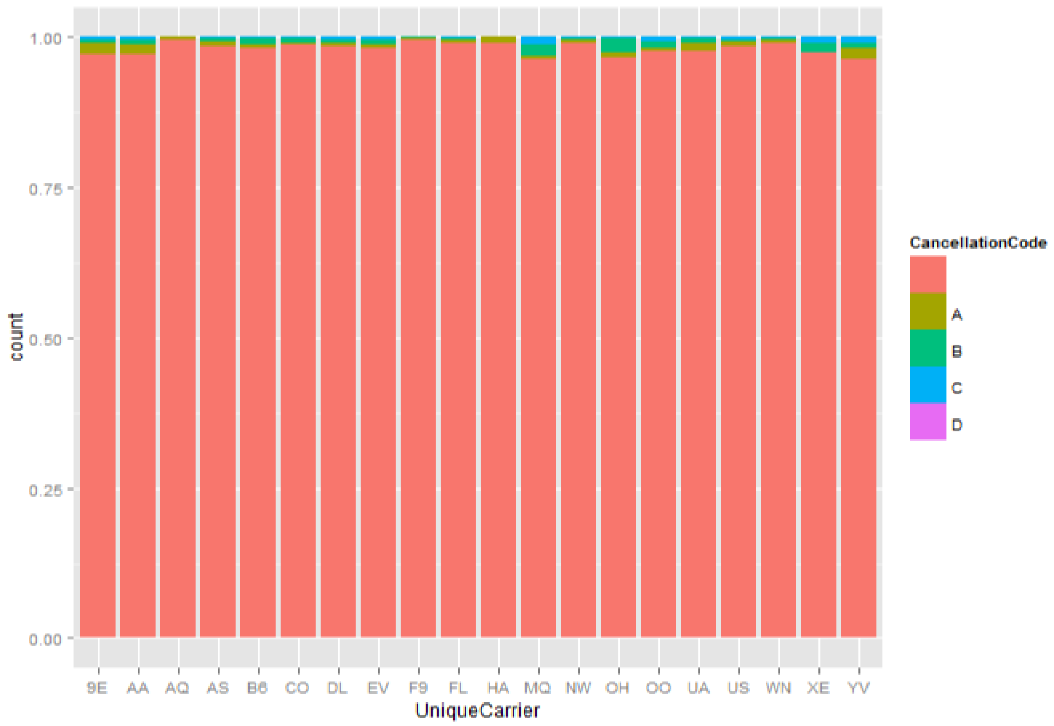
stacked barchart - carrier & cancellationcode

ggplot(DF, aes(x = UniqueCarrier, fill = CancellationCode)) + geom_bar()



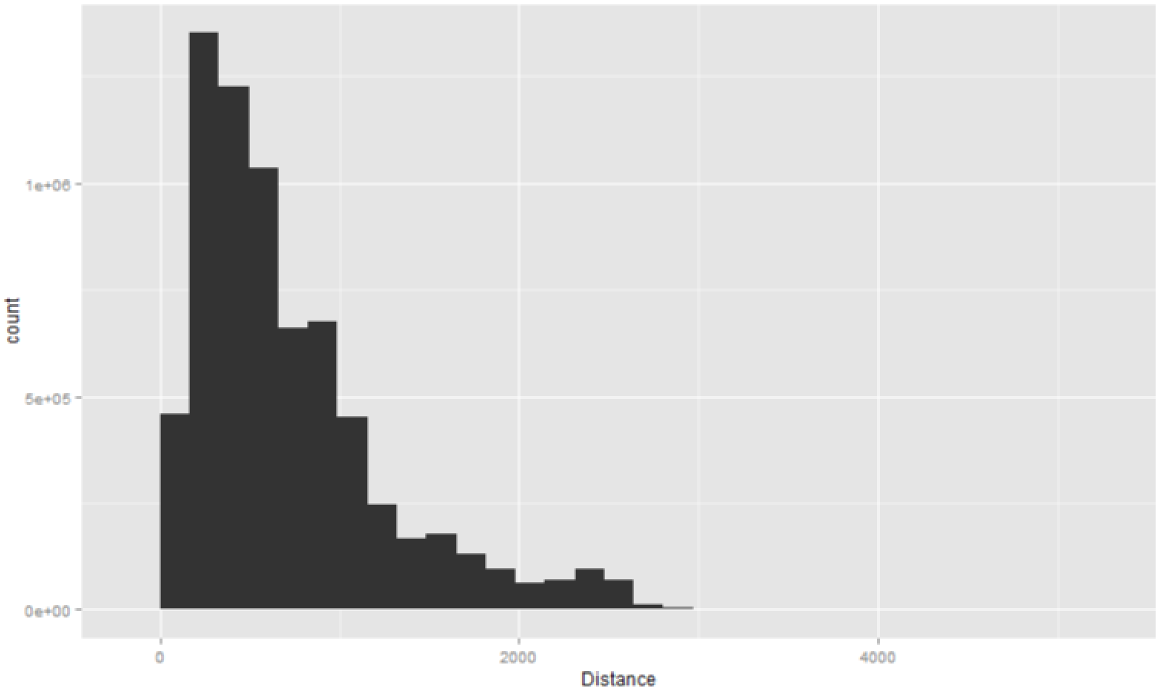
stacked with proportion

ggplot(DF, aes(x = UniqueCarrier, fill = CancellationCode)) + geom_bar(position="fill")



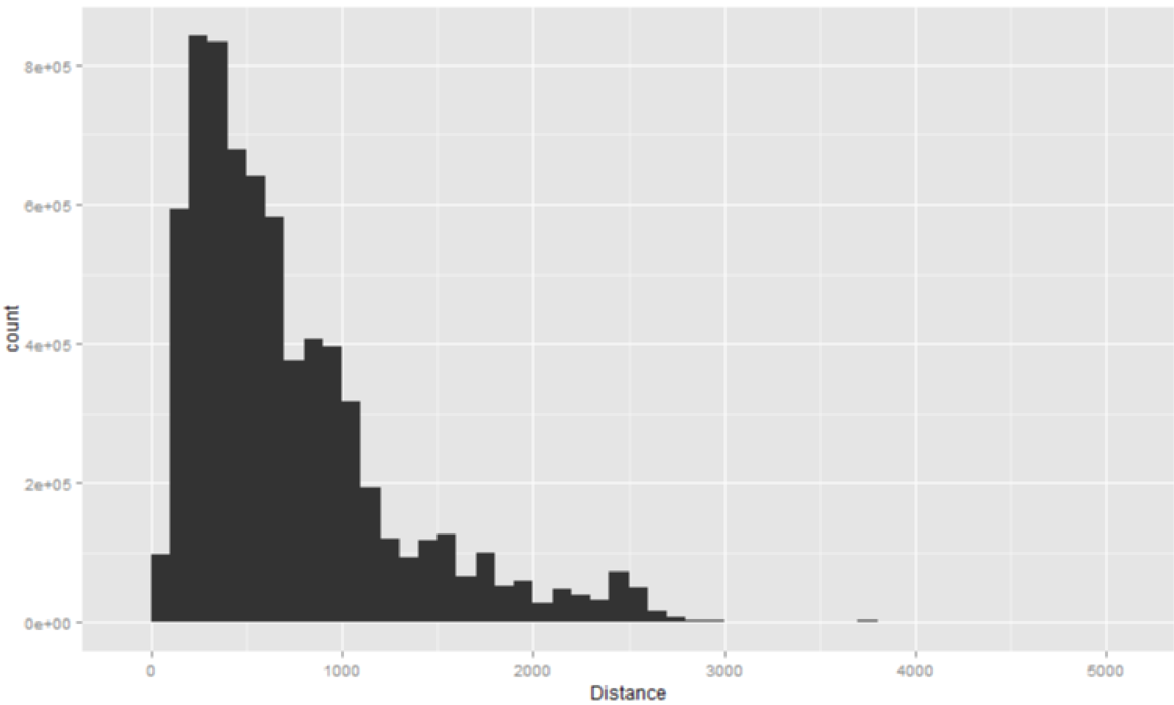
histogram (1 quantative variable)

ggplot(DF, aes(x = Distance)) + geom_histogram()



stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this

ggplot(DF, aes(x = Distance)) + geom_histogram(binwidth = 100)



Presentation of the final work

Interactive version of the design highlighted in the above section is represented in form of a story using story feature in Tableau. A **story** in tableau is a sheet that contains a sequence of worksheets or dashboards that work together to convey information. It is a very good collaborative visualization representation of the work for the management in the company. Story is used to show how the level of difficulty in the dashboard increases from dashboard 1 to dashboard 6. All dashboards are designed based on same dataset, but are designed with different types of the audience. Level of difficulty to read the information increase from dashboard 1 to dashboard 6. There are a number of design elements in the story which can be viewed on this given link. Figure 7 is pictorial representation of these features.

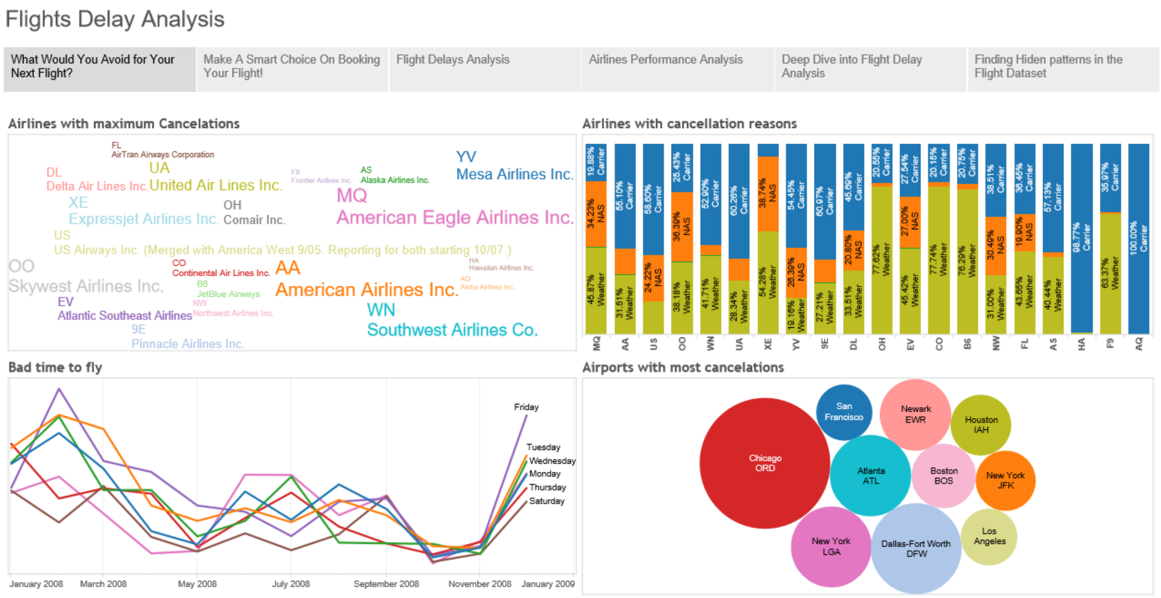


Fig: 7 Story with collaborative representation of all the dashboards created from Flights Delay Analysis.

Conclusion

Data is free but the information hiding in this data is priceless. In this project dataset which was free to download from internet was processed to find interesting insides of the ✈ delay. A picture is worth thousand words, so visual representations was created to share insights with different types of audience. There are some golden rules to keep in mind before designing any dashboard, these are know your audience , know what is the most important information for your target audience keep that information on top left corner of the dashboard. Use overlay techniques to draw attention of the viewer on the right place in the chart. Keep Data to ink ratio in mind while designing charts. Avoid Chartjunks. Dashboards are an important aspect of business management systems and they are referred to during planning and decision making process. Therefore, it is essential to make the dashboard uncluttered and user friendly.

Some more lessons were learned while working with large datasets. Hadoop worked well with this large dataset, queries were very slow in databases like Mysql. R server crashed many times. Charts drawn using R were not very interactive and not very informative. Many tools like Rapidminer failed to draw charts using this large dataset. Tableau worked very well for the dataset chosen and performance of the dashboards was phenomenal.

References

1. Kong, N., & Agrawala, M. (2012). Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12), 2631–2638. http://doi.org/10.1109/TVCG.2012.229
2. Ward, Matthew, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. AK Peters, Ltd., 2010
3. Zoss, A. 2015, Introduction to Data Visualization: Visualization Types, Duke University Libraries
4. DataLabs, 2015, 15 Most Common Types of Data Visualisation
5. http://www.infovis-wiki.net/index.php/Data-Ink_Ratio
6. https://en.wikipedia.org/wiki/Chartjunk
7. http://www.dashboardinsight.com/articles/digital-dashboards/fundamentals/5-things-to-keep-in-mind-while-designing-a-digital-dashboard.aspx
8. http://stat-computing.org/dataexpo/2009/posters/sun.pdf
9. http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/
10. http://stat-computing.org/dataexpo/2009/

Appendix

National federal holidays of USA in 2008

Day	Date	Holiday	Comments
Tuesday	January 01	New Years Day	
Monday	January 21	Martin Luther King Day	Third Monday in January
Monday	February 18	Presidents Day	Third Monday in February
Monday	May 26	Memorial Day	Last Monday in May
Friday	July 04	Independence Day	
Monday	September 01	Labor Day	First Monday in September
Monday	October 13	Columbus Day	2nd Monday in October
Tuesday	November 11	Veterans Day	
Thursday	November 27	Thanksgiving	Fourth Thursday in November
Friday	November 28	Day after Thanksgiving	Fourth Friday in November
Thursday	December 25	Christmas Day	

Test Results

drop table flight_data;

CREATE TABLE flight_data(year INT, month INT, day INT, day_of_week INT, dep_time INT, crs_dep_time INT, arr_time INT, crs_arr_time INT, unique_carrier VARCHAR(10), flight_num INT, tail_num VARCHAR(40), actual_elapsed_time INT, crs_elapsed_time INT, air_time INT, arr_delay INT, dep_delay INT, origin VARCHAR(10), dest VARCHAR(10), distance INT, taxi_in INT, taxi_out INT, cancelled INT, cancellation_code VARCHAR(10), diverted INT, carrier_delay VARCHAR(10), weather_delay VARCHAR(10), nas_delay VARCHAR(10), security_delay VARCHAR(10), late_aircraft_delay VARCHAR(10))

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

STORED AS TEXTFILE;

LOAD DATA INPATH '/data/flight_data/2008.csv' into table flight_data;

SELECT CANCELLED, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/7009728)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM flight_data GROUP BY CANCELLED;

0 6872294 98.04

1 137434 1.96

SELECT DIVERTED, COUNT(*) AS TOTAL_FLIGHTS ,ROUND(((COUNT(*)/7009728)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM flight_data GROUP BY DIVERTED;

0 6992463 99.75

1 17265 0.25

SELECT COUNT(*) AS TOTAL_FLIGHTS ,ROUND(((COUNT(*)/7009728)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM flight_data WHERE arr_delay > 15 ;

1466191 20.92

SELECT COUNT(*) AS TOTAL_FLIGHTS ,ROUND(((COUNT(*)/7009728)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM flight_data WHERE DEP_delay > 15 ;

1276396 18.21

SELECT COUNT(*) AS TOTAL_FLIGHTS ,ROUND(((COUNT(*)/7009728)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM flight_data WHERE arr_delay <= 15 AND arr_delay IS NOT NULL;

5388838 76.88

drop table FLIGHTS_CANCELLED;

CREATE TABLE FLIGHTS_CANCELLED AS SELECT CONCAT(CONCAT(CONCAT(CONCAT(DAY,'-'),MONTH),'-'),YEAR) AS FLIGHT_DATE , day, MONTH , DAY_OF_WEEK , unique_carrier , ORIGIN , DEST , cancellation_code FROM flight_data WHERE CANCELLED = 1;

SELECT UNIQUE_CARRIER, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/137434)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_CANCELLED GROUP BY UNIQUE_CARRIER order by Perc_OF_TOTAL_FLIGHTS desc;

MQ	18331	13.34
AA	17440	12.69
OO	12436	9.05
WN	12389	9.01
UA	10541	7.67
XE	9992	7.27
YV	9219	6.71
9E	7100	5.17
DL	6813	4.96
US	6582	4.79

OH	6462	4.7
EV	5026	3.66
CO	3702	2.69
B6	3205	2.33
NW	2906	2.11
FL	2236	1.63
AS	2139	1.56
HA	570	0.41
F9	303	0.22
AQ	42	0.03

SELECT MONTH, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/137434)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_CANCELLED GROUP BY MONTH order by Perc_OF_TOTAL_FLIGHTS desc;

2	20596	14.99
12	17779	12.94
1	17308	12.59
3	16183	11.78
6	10931	7.95
7	10598	7.71
4	10355	7.53
9	9913	7.21
8	9835	7.16
5	6229	4.53
11	4458	3.24
10	3249	2.36

SELECT DAY_OF_WEEK, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/137434)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_CANCELLED GROUP BY DAY_OF_WEEK order by Perc_OF_TOTAL_FLIGHTS desc;

5	23962	17.44
2	23168	16.86
1	20513	14.93
3	20202	14.7
7	18138	13.2
4	17884	13.01
6	13567	9.87

SELECT ORIGIN, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/137434)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_CANCELLED GROUP BY ORIGIN order by Perc_OF_TOTAL_FLIGHTS desc;

ORD	15050	10.95
DFW	7272	5.29
ATL	5830	4.24
LGA	5753	4.19
EWR	4511	3.28
BOS	3655	2.66
IAH	3261	2.37

JFK	3196	2.33
LAX	2838	2.06
SFO	2790	2.03
DCA	2735	1.99
DEN	2725	1.98
DTW	2583	1.88
IAD	2077	1.51
LAS	2057	1.5
CLT	1986	1.45
PHL	1969	1.43
PHX	1875	1.36
CVG	1853	1.35
MSP	1729	1.26

SELECT DEST, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/137434)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_CANCELLED GROUP BY DEST order by Perc_OF_TOTAL_FLIGHTS desc limit 20;

ORD	16094	11.71
DFW	7716	5.61
ATL	6705	4.88
LGA	5721	4.16
EWR	4608	3.35
BOS	3601	2.62
IAH	3366	2.45
JFK	3200	2.33
LAX	3077	2.24
DTW	2995	2.18
DEN	2984	2.17
SFO	2984	2.17
DCA	2623	1.91
PHX	2554	1.86
CLT	2227	1.62
IAD	2210	1.61
CVG	2144	1.56
MSP	1988	1.45
PHL	1990	1.45
LAS	1805	1.31

SELECT cancellation_code, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/137434)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_CANCELLED GROUP BY cancellation_code order by Perc_OF_TOTAL_FLIGHTS desc;

B	54904	39.95
A	54330	39.53
C	28188	20.51
D	12	0.01

drop table FLIGHTS_ON_TIME;

CREATE TABLE FLIGHTS_ON_TIME AS SELECT CONCAT(CONCAT(CONCAT(CONCAT(DAY,'-'),MONTH),'-'),YEAR) AS FLIGHT_DATE , day, MONTH , DAY_OF_WEEK , unique_carrier , ORIGIN , DEST , ARR_DELAY , DEP_DELAY FROM flight_data WHERE arr_delay <= 15 AND arr_delay IS NOT NULL;

SELECT UNIQUE_CARRIER, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/5388838)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_ON_TIME GROUP BY UNIQUE_CARRIER order by Perc_OF_TOTAL_FLIGHTS desc;

WN	976179	18.11
OO	452374	8.39
AA	428102	7.94
US	366754	6.81
MQ	361287	6.7
DL	349551	6.49
UA	325814	6.05
XE	278518	5.17
NW	270351	5.02
CO	223470	4.15
EV	210589	3.91
9E	210331	3.9
FL	203050	3.77
YV	188023	3.49
B6	144409	2.68
OH	140332	2.6
AS	119631	2.22
F9	76724	1.42
HA	55967	1.04
AQ	7382	0.14

SELECT MONTH, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/5388838)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_ON_TIME GROUP BY MONTH order by Perc_OF_TOTAL_FLIGHTS desc;

8	484800	9.0
5	484325	8.99
10	482378	8.95
7	480267	8.91
4	469710	8.72
9	462443	8.58
3	446837	8.29
1	443955	8.24
11	439743	8.16
6	436783	8.11
2	396241	7.35
12	361356	6.71

```
SELECT DAY_OF_WEEK, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/5388838)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_ON_TIME GROUP BY DAY_OF_WEEK order by Perc_OF_TOTAL_FLIGHTS desc;
```

3	816393	15.15
2	800981	14.86
1	796946	14.79
4	790644	14.67
5	760134	14.11
7	743418	13.8
6	680322	12.62

```
SELECT ORIGIN, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/5388838)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_ON_TIME GROUP BY ORIGIN order by Perc_OF_TOTAL_FLIGHTS desc;
```

ATL	306995	5.7
ORD	233115	4.33
DFW	205529	3.81
DEN	185065	3.43
LAX	174247	3.23
PHX	162509	3.02
IAH	142825	2.65
LAS	137240	2.55
DTW	119382	2.22
SLC	116760	2.17
SFO	104062	1.93
MCO	103711	1.92
MSP	99176	1.84
CLT	95184	1.77
EWR	89402	1.66
SEA	87366	1.62
BOS	86839	1.61
LGA	82183	1.53
BWI	82593	1.53
JFK	80966	1.5

```
SELECT DEST, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/5388838)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_ON_TIME GROUP BY DEST order by Perc_OF_TOTAL_FLIGHTS desc limit 20;
```

ATL	316754	5.88
ORD	239876	4.45
DFW	216677	4.02
DEN	191420	3.55
LAX	167896	3.12
PHX	162889	3.02
IAH	146521	2.72
LAS	135961	2.52
DTW	130836	2.43

SLC	117289	2.18
MSP	104003	1.93
MCO	102918	1.91
CLT	99759	1.85
SFO	98292	1.82
EWR	87301	1.62
BOS	87433	1.62
BWI	84355	1.57
SEA	83609	1.55
JFK	82564	1.53
LGA	75949	1.41

drop table FLIGHTS_DELAYED;

CREATE TABLE FLIGHTS_DELAYED AS SELECT CONCAT(CONCAT(CONCAT(CONCAT(DAY,'-'),MONTH),'-'),YEAR) AS FLIGHT_DATE ,day, MONTH , DAY_OF_WEEK , unique_carrier ,Distance, ORIGIN , DEST , ARR_DELAY , DEP_DELAY, dep_time , arr_time ,air_time, carrier_delay ,weather_delay, nas_delay, security_delay, late_aircraft_delay FROM flight_data WHERE arr_delay > 15;

SELECT UNIQUE_CARRIER, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/1466191)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_DELAYED GROUP BY UNIQUE_CARRIER order by Perc_OF_TOTAL_FLIGHTS desc;

WN	210732	14.37
AA	157383	10.73
UA	112165	7.65
MQ	109874	7.49
OO	101038	6.89
DL	94383	6.44
XE	84896	5.79
US	79332	5.41
NW	73759	5.03
CO	70385	4.8
EV	64278	4.38
YV	57108	3.89
FL	55663	3.8
OH	50363	3.43
B6	47705	3.25
9E	43991	3.0
AS	28861	1.97
F9	18660	1.27
HA	5245	0.36
AQ	370	0.03

SELECT MONTH, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/1466191)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_DELAYED GROUP BY MONTH order by Perc_OF_TOTAL_FLIGHTS desc;

12	163391	11.14
6	158675	10.82

3	151506	10.33
2	150684	10.28
1	143175	9.77
7	135156	9.22
4	117013	7.98
8	115950	7.91
5	114885	7.84
11	78230	5.34
10	69693	4.75
9	67833	4.63

SELECT DAY_OF_WEEK, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/1466191)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_DELAYED GROUP BY DAY_OF_WEEK order by Perc_OF_TOTAL_FLIGHTS desc;5

	248738	16.96
4	221326	15.1
1	216464	14.76
7	212709	14.51
2	205011	13.98
3	200602	13.68
6	161341	11.0

SELECT ORIGIN, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/1466191)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_DELAYED GROUP BY ORIGIN order by Perc_OF_TOTAL_FLIGHTS desc limit 20;

ORD	101414	6.92
ATL	100706	6.87
DFW	67845	4.63
DEN	53126	3.62
EWR	44269	3.02
DTW	39622	2.7
IAH	38578	2.63
LAX	38047	2.59
PHX	34582	2.36
JFK	34195	2.33
SFO	33393	2.28
LAS	33240	2.27
LGA	30929	2.11
MSP	29063	1.98
CLT	28598	1.95
BOS	27067	1.85
MCO	25681	1.75
PHL	24437	1.67
SLC	20824	1.42
SEA	20297	1.38

SELECT DEST, COUNT(*) AS TOTAL_FLIGHTS,ROUND(((COUNT(*)/1466191)*100),2) AS Perc_OF_TOTAL_FLIGHTS FROM FLIGHTS_DELAYED GROUP BY DEST order by Perc_OF_TOTAL_FLIGHTS desc limit 20;

ORD	93455	6.37
ATL	89990	6.14
DFW	55758	3.8
DEN	46682	3.18
EWR	46037	3.14
LAX	44475	3.03
SFO	39060	2.66
LGA	36746	2.51
LAS	34705	2.37
IAH	34504	2.35
PHX	33640	2.29
JFK	32508	2.22
DTW	27963	1.91
BOS	26727	1.82
MCO	26694	1.82
SEA	24128	1.65
MSP	24011	1.64
PHL	24021	1.64
CLT	23790	1.62
SLC	20281	1.38

/* Finding percentage of delays based on different factors */

SELECT MONTH, COUNT(*) AS TOTAL_FLIGHTS, round(avg(arr_delay),2) AS AVG_DELAYS FROM FLIGHTS_DELAYED GROUP BY MONTH order by AVG_DELAYS desc;

7	135156	64.31
12	163391	64.08
6	158675	61.69
8	115950	60.61
2	150684	59.32
1	143175	57.85
3	151506	57.83
11	78230	54.5
4	117013	54.41
5	114885	53.4
9	67833	52.84
10	69693	49.5

SELECT UNIQUE_CARRIER, COUNT(*) AS TOTAL_FLIGHTS, round(avg(arr_delay),2) AS AVG_DELAYS FROM FLIGHTS_DELAYED GROUP BY UNIQUE_CARRIER order by AVG_DELAYS desc;

B6	47705	69.49
YV	57108	66.43
XE	84896	64.45

UA	112165	63.82
EV	64278	62.3
CO	70385	61.66
OO	101038	60.57
AA	157383	60.26
MQ	109874	60.24
9E	43991	59.15
OH	50363	59.02
FL	55663	58.35
DL	94383	53.59
NW	73759	52.86
WN	210732	52.16
US	79332	51.84
AS	28861	51.53
HA	5245	51.09
F9	18660	42.88
AQ	370	41.67

SELECT ORIGIN, COUNT(*) AS TOTAL_FLIGHTS, round(avg(dep_delay),2) AS AVG_dep_DELAYS FROM FLIGHTS_DELAYED GROUP BY ORIGIN order by AVG_dep_DELAYS desc limit 30;

ACY	14	144.14
CMX	33	117.15
PIR	1	99.0
ALO	27	88.67
SPI	333	87.2
PLN	22	86.91
MOT	115	84.66
SLE	22	84.27
ABI	293	83.38
HHH	172	79.09
CIC	331	78.11
EGE	752	78.11
ACK	157	77.09
LNK	572	74.21
GUC	218	74.04
CEC	378	73.92
RDD	414	73.79
LMT	150	73.43
MQT	182	73.42
SBN	1186	72.22
SGF	2057	71.78
ACV	888	70.72

MBS	497	69.61
LCH	141	68.1
YAK	153	67.88
CWA	521	67.48
SBP	855	67.43
ROA	725	67.14
BGM	108	66.99
BPT	16	66.88

SELECT DEST, COUNT(*) AS TOTAL_FLIGHTS, round(avg(arr_delay),2) AS AVG_arr_DELAYS FROM FLIGHTS_DELAYED GROUP BY DEST order by AVG_arr_DELAYS desc;

MQT	352	94.3
SPI	357	80.5
ALO	50	76.24
EWR	46037	75.93
MCN	158	75.92
ORD	93455	73.94
MEI	118	72.95
TEX	28	72.5
CMX	49	72.08
CIC	430	71.79
LMT	136	71.7
ADQ	72	70.25
LNK	624	69.39
CEC	289	69.12
JFK	32508	68.62
ACY	24	68.46
AVP	584	67.14
BPT	45	66.58
CDV	191	66.28
BQN	410	65.87
HHH	234	65.87
CMI	799	65.74
CAE	2730	65.54
CRW	671	65.54
IAD	16782	65.54
GTR	185	65.35
YAK	201	65.27
MBS	678	64.84
FLO	119	64.81
EGE	846	64.79
ELM	212	64.68

LCH	212	64.58
PHL	24021	64.43
CHO	176	64.4
SFO	39060	63.83
INL	10	63.8
BTV	1675	63.72
ACK	174	63.71
CWA	638	63.44
BOS	26727	63.32
ROA	768	63.2
MOD	544	63.14
ABE	1048	63.1
RDD	466	63.03

SELECT DAY_OF_WEEK, COUNT(*) AS TOTAL_FLIGHTS, round(avg(arr_delay),2) AS AVG_DELAYS FROM FLIGHTS_DELAYED GROUP BY DAY_OF_WEEK order by AVG_DELAYS desc;

7	212709	62.42
2	205011	59.41
5	248738	58.98
1	216464	58.28
6	161341	57.41
4	221326	56.48
3	200602	56.09

--SELECT DISTANCE_GROUP, COUNT(*) AS TOTAL_FLIGHTS, round(avg(arr_delay),2) AS AVG_DELAYS FROM FLIGHTS_DELAYED GROUP BY DISTANCE_GROUP order by AVG_DELAYS desc;

```
library(ggplot2)

# loading the graphical library ggplot2

DF <- read.csv("//home//edureka//Downloads//new_2008.csv")

#colnames(DF) <- c("SNo", "Year"....)

dim(DF)


str(DF)


vec <- c("Year", "Month", "DayofMonth", "DayOfWeek", "DepTime", "CRSDepTime", "ArrTime", "CRSArrTime", "UniqueCarrier", "FlightNum", "TailNum", "ActualElapsedTime", "CRSElapsedTime", "AirTime", "ArrDelay", "DepDelay", "Origin", "Dest", "Distance", "TaxiIn", "TaxiOut", "Cancelled", "CancellationCode", "Diverted", "CarrierDelay", "WeatherDelay", "NASDelay", "SecurityDelay", "LateAircraftDelay")

colnames(DF) <- vec


str(DF)


X <- DF$ArrDelay + DF$DepDelay

summary(X)
```

```
Y <- with(DF, ArrDelay + DepDelay)
summary(Y)

#table
table(DF$UniqueCarrier)

with (DF, table(UniqueCarrier))

with (DF, table(UniqueCarrier, CancellationCode))

#proportions
X <- with (DF, table(UniqueCarrier))
X/sum(X)*100

# flights related to Christmas season # take a subset of flights on Dec 25
ChristmasFlights <- subset(DF, DayofMonth == 25)
summary(ChristmasFlights)
dim(ChristmasFlights)
ChristmasWeek <- subset(DF, DayofMonth %in% 21:27)
# between destinations - say origin is from JFK
JFKStuff <- subset(DF, Origin == "JFK")
JFKStuff <- subset(ChristmasFlights, Origin == "JFK" & Dest == "PIT")
Somestations <- subset(ChristmasFlights, Origin %in% c("JFK", "PIT", "LGA"))

# hist, plot, boxplot
# basic syntax of ggplot
# ggplot(dataframe, aesthetics = x coordinate, y coordinate, shape) + layer = geom_XYZ XYZ {bar, histogram}
ggplot(DF, aes(x = UniqueCarrier)) + geom_bar()
# stacked barchart - carrier & cancellationcode
ggplot(DF, aes(x = UniqueCarrier, fill = CancellationCode)) + geom_bar()
# stacked with proportion
ggplot(DF, aes(x = UniqueCarrier, fill = CancellationCode)) + geom_bar(position="fill")
# histogram (1 quantative variable)
ggplot(DF, aes(x = Distance)) + geom_histogram()
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this
ggplot(DF, aes(x = Distance)) + geom_histogram(binwidth = 100)
```

Note : Get data , Clean Data in HIVE , feed data into Tableau and show different techniques using different processing and charts.

```
# Uncomment line below if there's an issue getting mySQL to work
# HIVESERVER2 in unix prompt
```

```
# install.packages("/home/edureka/Downloads/RHive_2.0-0.2.tar.gz", repos = NULL, type = "source")
library(ggplot2)
library(RHive)
rhive.init()
rhive.env()
Sys.setenv(HADOOP_HOME="/usr/lib/hadoop-2.2.0")
Sys.setenv(HADOOP_STREAMING="/usr/lib/hadoop-2.2.0/share/hadoop/tools/lib/hadoop-streaming-2.2.0.jar")
Sys.setenv(HIVE_HOME="/usr/lib/hive-0.13.1-bin")
Sys.setenv(HADOOP_CMD="/usr/lib/hadoop-2.2.0/bin/hadoop")
Sys.setenv(RHIVE_FS_HOME="/home/edureka/Downloads/RHive")
rhive.init()
rhive.env()
rhive.connect(host="192.168.56.102",user="edureka", defaultFS="hdfs://localhost:8020")
rhive.query("show databases")
rhive.query("use airlines")
rhive.query("show tables")
# rhive.query("drop table airport1")
rhive.query("Select * from flights_delayed limit 10 ")
DF <- rhive.query("Select * from flights_delayed")
DF
colnames(DF)
length(rownames(DF))
# rhive.write.table(flight_delays, tableName = 'flight_delays', sep=',')
# rhive.query("Select * from flight_delays limit 10")
# flight_delays2 <- rhive.load.table(tableName = 'flight_delays')
# flight_delays2
dim(DF)
str(DF)
DF$flights_delayed.arr_delay
X <- DF$flights_delayed.arr_delay + DF$flights_delayed.dep_delay
summary(X)
Y <- with(DF, flights_delayed.arr_delay + flights_delayed.dep_delay)
summary(Y)
summary(DF$flights_delayed.day)
# flights related to Christmas season # take a subset of flights on Dec 25
ChristmasFlights <- subset(DF, DF$flights_delayed.day == 25)
summary(ChristmasFlights)
dim(ChristmasFlights)
ChristmasWeek <- subset(DF, flights_delayed.day %in% 21:27)
# between destinations - say origin is from JFK
```

hist, plot, boxplot

basic syntax of ggplot

histogram (1 quantative variable)

ggplot(DF, aes(x = DF$flights_delayed.distance)) + geom_histogram()

stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this

ggplot(DF, aes(x = DF$flights_delayed.distance)) + geom_histogram(binwidth = 100)