# Programming for Data Science

*Coursework Project*

**NAME – Linal De Silva**

**UOL STUDENT NUMBER - 200640778**

**MODULE – Programming for Data Science (ST2195)**

**Table of content**


- Introduction ……. 3
- Question 1 ……..4
- Question 2……..6
- Question 3……..7
- Question 4……..8
- Question 5…… ……10

**Question 1**

## When is the best time of day, day of the week, and time of year to fly to minimize delays?

- Best time of the day -

First we merge two datasets (2005,2006) and rename it as datacomb. Then by taking its 2 columns CRSDepTime and DepDelay, we can create a function to find the mean DepDelay for every minute in CRSDepTime using the dataset datacomb. Then it gives us,

| CRSDepTime | DepDelay |
|---|---|
| 0 | 3.00000000 |
| 1 | 3.53846154 |
| 4 | 6.25000000 |
| 5 | 3.21774194 |
| 8 | 47.75288809 |
| 10 | 4.44031739 |
| 11 | -1.28571429 |
| 15 | 8.50898831 |
| 17 | 72.50000000 |
| 18 | 56.00000000 |
| 20 | 13.32171211 |
| 21 | 4.71111111 |
| 23 | 9.80704441 |
| 25 | 5.10254839 |
| 28 | 17.25806452 |
| 29 | 2.84375000 |
| 30 | 8.58263091 |
| 34 | -2.54545455 |
| 35 | 4.91635266 |
| 36 | 3.55681818 |
| 38 | 13.00000000 |

1-21 of 1,223 rows  Previous 1 2 3 4 5 6 44 Next

*Table 1 -CRsDepTime with DepDelay*

*It is assumed that time of the days is exact time of the day rather than nearest (e.g. to hour)*

From that 1223 rows we can find the time which has the minimum mean DepDelay by using a function. Therefor it outputs 350. Then we can suggest that by analyzing the data of 2 years, 03:50am is the best time of the day to fly to minimize delays.

- Best day of the week –

Using the same dataset datacomb, we can take out 2 columns named Dayofweek and Depdelay to create a function where we can find the mean DepDelay on every day of week. Which gives us,

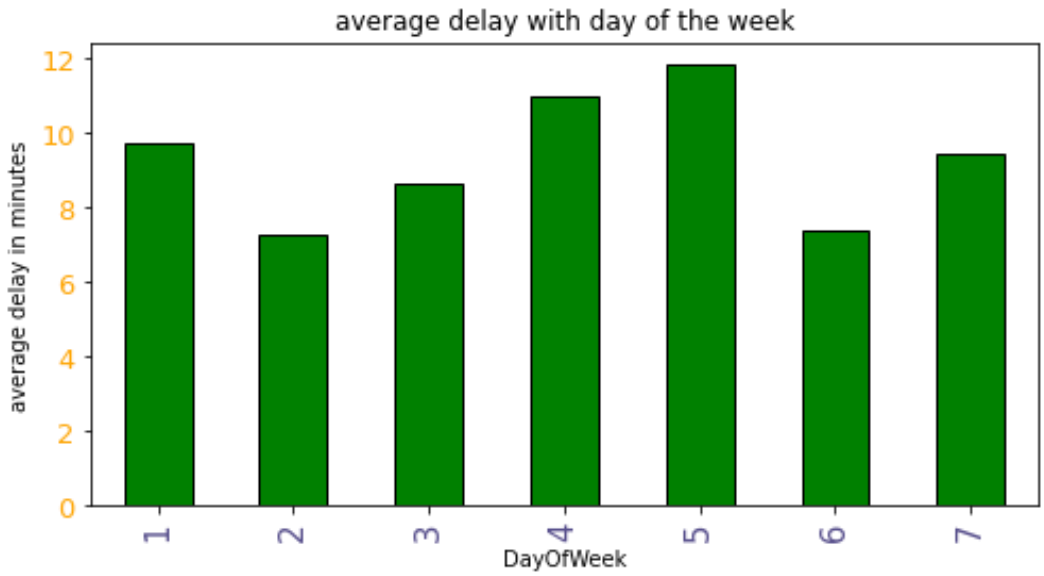



*Figure 1 – average delay time for each day of week*

From reviewing the above data, we can see that Tuesday **(7.287008)** and Saturday **(7.411122)** both has almost similar mean DepDelays. But it is abvious that Tuesday is the best day to fly to minimize delays.

- Best time of the year to minimize delays –

By using the columns Month, DayofMonth, CRSDepTime in datacomb we can create a function to find out the month, day of month, exact time which has the minimum DepDelay. Output of that function comes as 23:35pm on 11th of February. But that time can change year to year, therefor we calculate the best month and best day of month separately. Which gives us,

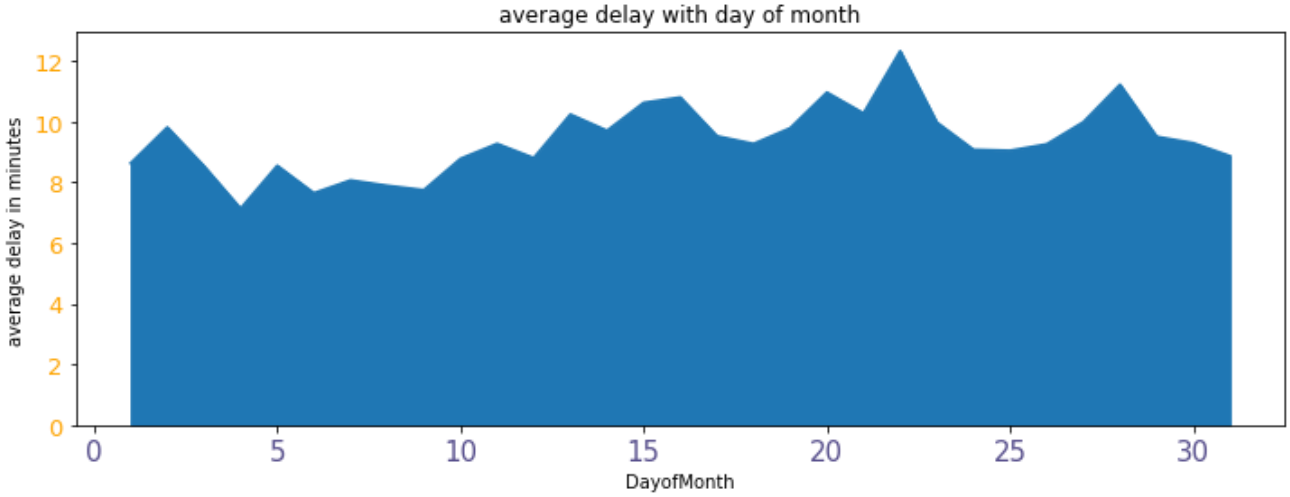



*Figure 2 – average delay time for each day of month*

We can see that 4th day of the month has the least mean DepDelay. Therefor we can conclude that flying on 4th day will minimize delays.

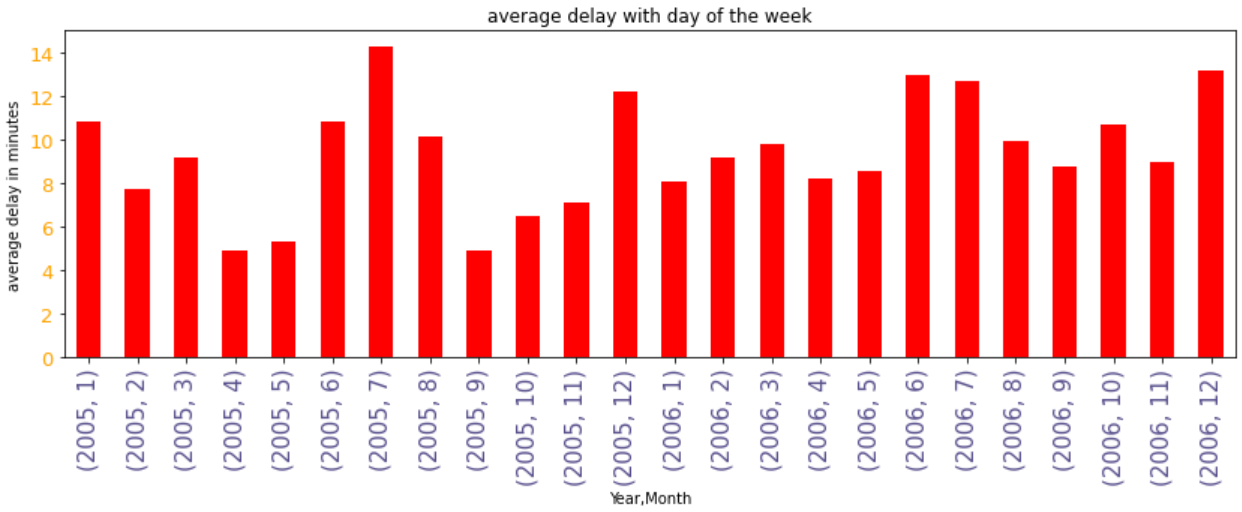



*Figure 3 – average delay for each month of a year*

Analyzing monthly trends in 2005 we can notice that April has the minimum DepDelay, also in 2006 April has minimum DepDelay after January. Therefor we can suggest that flying in April will help people minimize delays

**Question 2**

**Do older planes suffer more delays?**

By using the plane-data dataset given, first we omit NaN and NA values from the dataset. Then we have to remove None and 0000 values in the dataset separately to do any sort of calculations. After that we can change the year column name to year_produced (assuming that year in plane-data is the year which plane was produced). Then we can observe that theirs planes made in 1946 – 2008. To merge the dataset datacomb with plane-data dataset we change the TailNum to tailnum(changing from uppercase to lowercase assuming that both represent same meaning). Then we can merge 2 datasets on tailnum which gives us the mergeddata dataset.

Since year_produced column is in object datatype we change it to integer datatype to do calculations. Then we can create a column called age (how old the plane is) by subtracting year_produced column values from Year (year which the plane departures) column values. After dropping off unwanted columns we can assume a condition as if age of the plane is greater than 10 we call it as "old", if the value of age is less than or equals to 10 we call it as "new". Then we can create a column call "totaldelay" by adding up "ArrDelay" column values to "DepDelay" column values.

Then we can create a function to calculate the mean Depdelay for every age value in the age column, where we can clarify that whether the older the planes got more delayed than new planes. To get a greater picture of the data we can do a visualization on age of the plane with mean DepDelay.

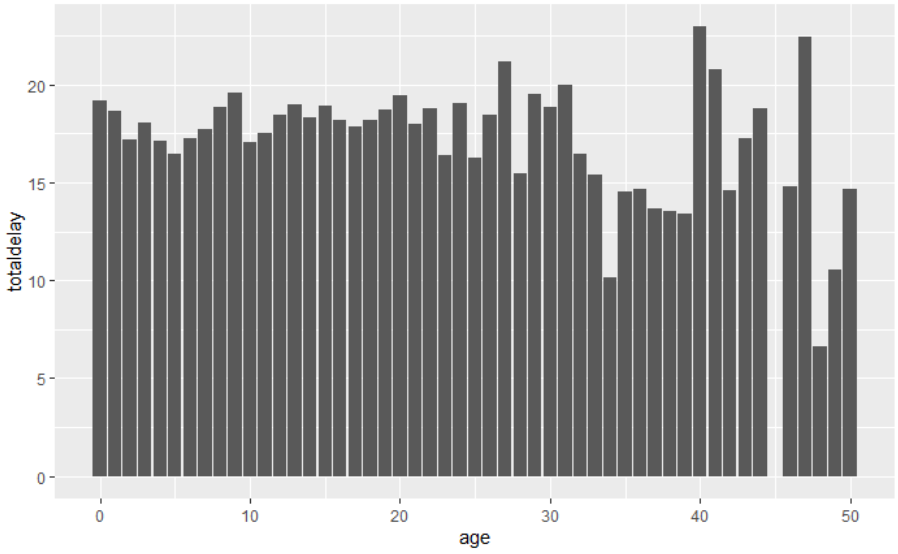



Figure 4 – age of the plane vs the average delay

It is clear from the above visualization that planes which are less than 10 years old are also suffer from delays such as planes which are older than 10 years. There is no huge difference between the means of 2 categories. Therefor we can suggest that suffering from delay of an airplane has no association with how old the plane is.

**Question 3**

**How does the number of people flying between different locations change over time?**

By taking the same datacomb dataset we can create a function to take the count of people fly between different locations. First we can create the function with taking Dest and Year, where we can discover how many number of people had visit different locations in 2005 and 2006.

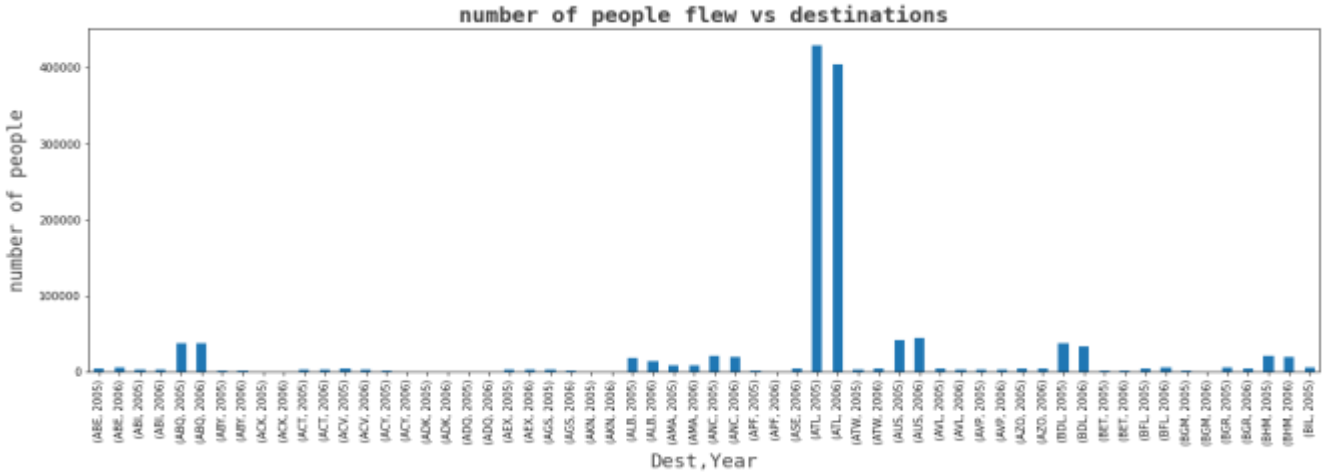



*Figure 5 – number of people come to a specific destination on 2005 and 2006*

From analyzing the sample of 30 destinations we can see that 429800 people has chosen ATL as their destination in 2005 while 404829 people choose it as their destination in 2006, therefor highest number of people had chosen ATL as their destination in both the years. Which is almost 8 times the second highest destination.

Also we can analyze the same sample with how many people have went to a different location from a specific Origin. To Compute that we can create a function with having the counts of the number of people flew from an Origin.

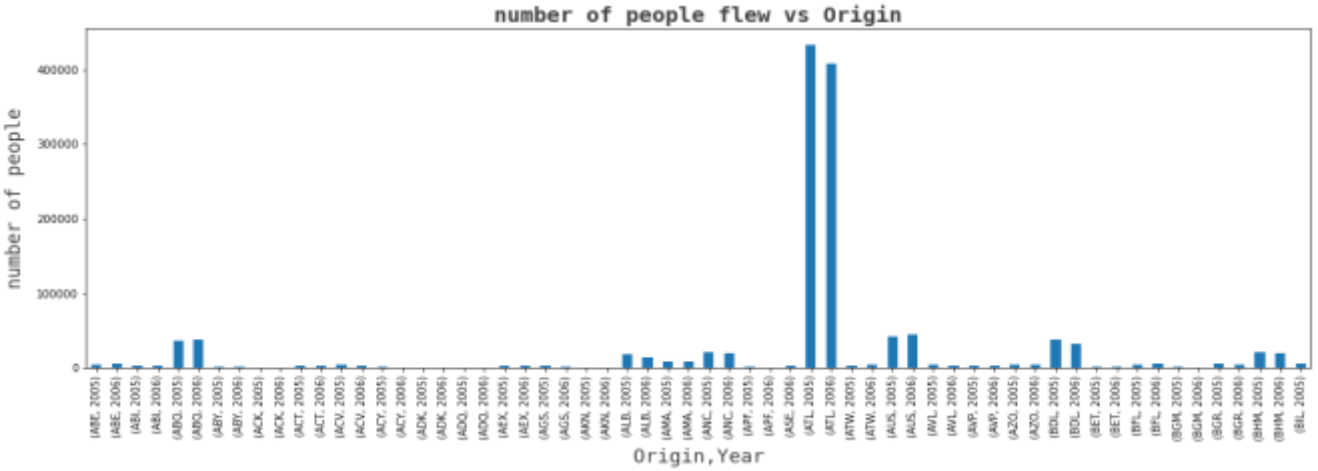



*Figure 6 – number of people fly from a specific destination on 2005 and 2006*

Here we can observe that this graph is almost similar to the previous graph. Which means that similar number of people who has arrived at a specific destination on a year has left the destination as well. Therefor we can suggest from this sample that when a number of people flies to a destination nearly same amount of people flies away from that destination in 2005 and 2006.

**Question 4**

**Can you detect cascading failures as delays in one airport create delays in others?**

To answer this question, we are calling the previously used datacomb dataset again and dropping off every column except ArrDelay and LateAircraftDelay. Therefor we can have a clearer look at the data we need to focus on to check weather if delays in one airport create delays in others.

To check that we can create a condition where x is equal to 0. Then we can take the rows where LateAircraftDelay is greater than x (where x = 0). which means there is a LateAircraftDelay. By taking its first 100 rows we can draw a scatter plot with the line of best fit. Which outputs as,

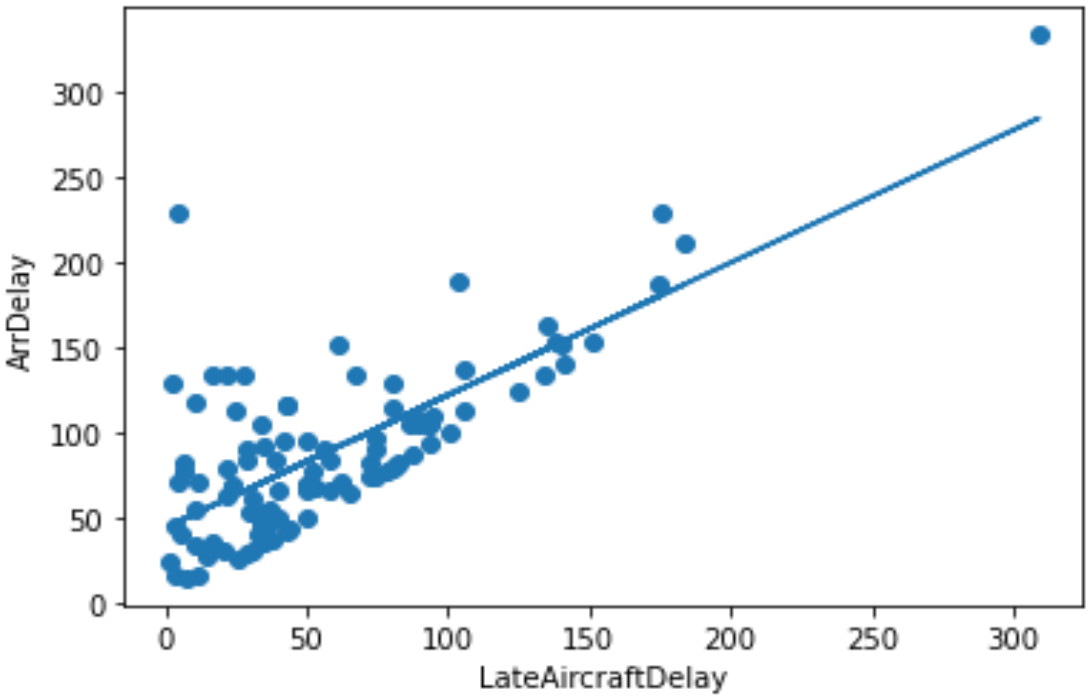



*Figure 7 – scatterplot for ArrDelay when there is aLateAircraftDelay.*

Also we can calculate $R^2$ for the dataset by assuming that the LateAircraftdelay is independent of ArrDelay. Therefor we got 0.547 as $R^2$, where we can say there is some kind of moderate relationship between LateAircraftDelay and ArrDelay.

Another way to prove that theirs cascading errors is by calculating the mean totaldelay when there is LateAircraftDelay. Since Arrdelay includes LateAircraftdelay we can subtract mean LateAircraftdelay **(42.8722)** from mean Arrdelay **(59.8013)** which is outputs as **16.93.**

Then we can create another dataset by taking the rows where LateAircraftdelay is equals to x (0). From that we can do a scatterplot but it is obvious that there can't be any relationship with Arrdelay when there is no LateAircraftDelay. And it is clear that $R^2$ will be 0 as well.

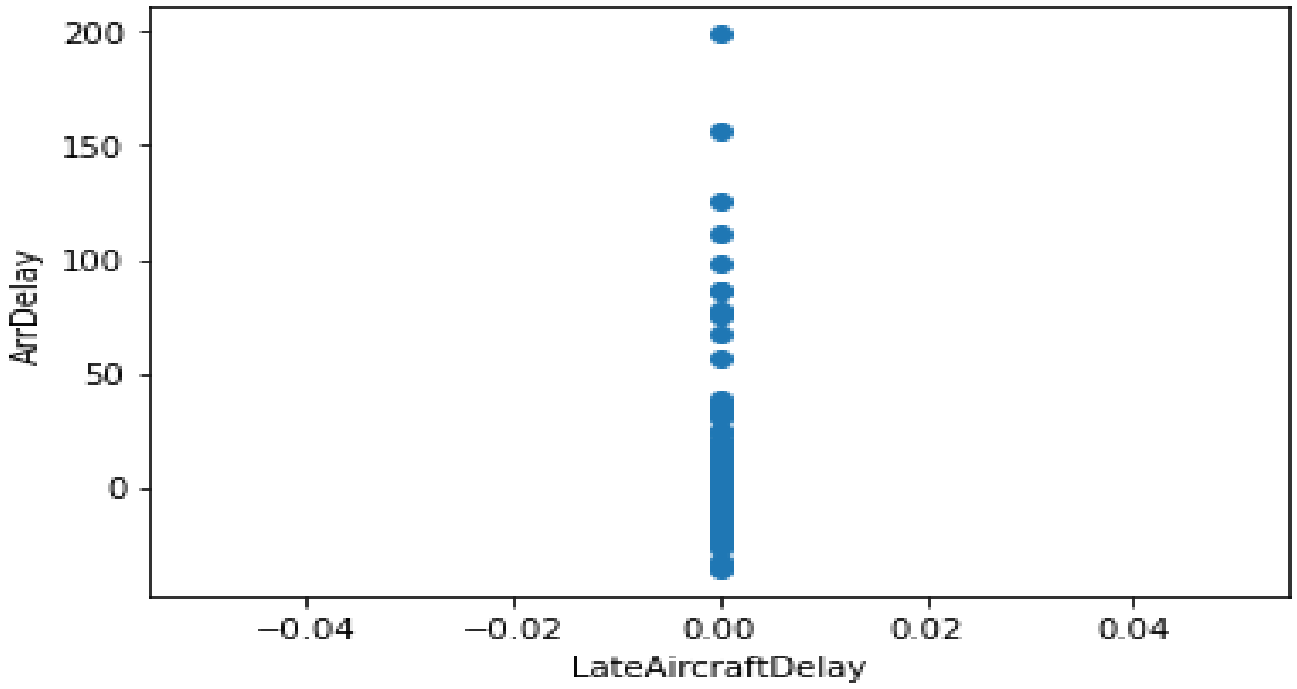


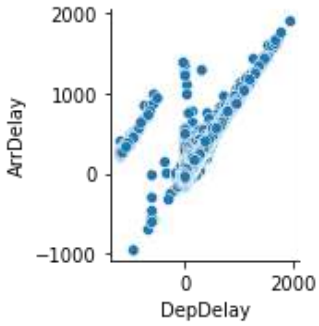

*Figure 8 – scatterplot for ArrDelay when there is no aLateAircraftDelay*.

But when we calculate the mean Arrdelay when LateAircraftdelay is zero, it output as **2.34.** Since mean lateAircraftDelay is o, subtracting mean LateAircraftdelay from mean Arrdelay will give output as **2.34** again.

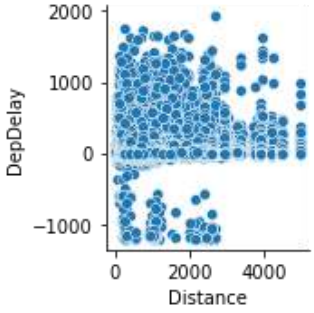
Therefor when comparing the 2 values 16.93 and 2.34 we can see that if there is a LateAircraftdelay, then the Arrival delays is large as 16.93 when compared to no LateAircraftDelay, which is only 2.34. Therefor we can suggest that when there is an arrival delay in one airport (LateAircraftDelay), it causes ArrDelay in other airport.

**Question 5**

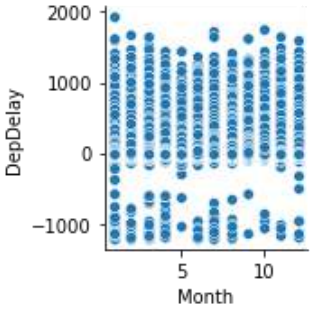
**Use the available variables to construct a model that predicts delays.**




By using the datacomb dataset we can draw a graph to check whether that when theirs minus DepDelays there are some flights having ArrDelays, but there is very small amount of flights where there are depDelays but no ArrDelays, which suggest that when there is no ArrDelays there will be no depDelays most of the time.

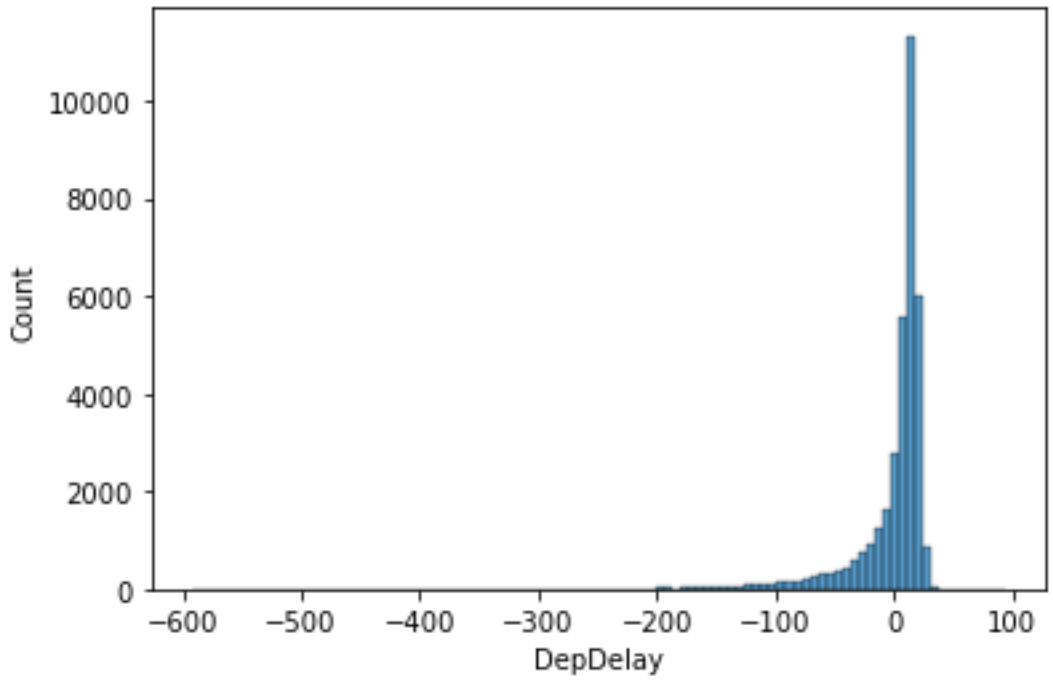



With this graph it is clear that when the distance of the flight is around less than 3000, some flights will departure earlier than CRSDepTime.




Here we can see that in months like January and april some flights do departure early but in months like May and September there is very less number of flights departing early.

Then since the dataset is large we are only taking a sample dataset to create the model. Then after dropping off that rows we are creating dummies for UniqueCarrier and Dest. Then we can create the model by taking y variable as DepDelay while taking x as the columns of datacomb dataset. Then we can split the dataset to train data and test data. Then we can plot a histogram for predicted DepDelays.




With this by using an function we can calculate $R^2$ as 0.01, which suggest vert weak relationship between the data.and we got mean absolute error as 20.88 while getting mean squared error as 1173.17.