

1 Exercici 1

Resumeix gràficament el data set DelayedFlights.csv

In [1]:

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import seaborn as sns
```

In [2]:

```
1 df = pd.read_csv("DelayedFlightsNet.csv", index_col = 0)
```

C:\Users\Nuria\anaconda3\lib\site-packages\numpy\lib\arraysetops.py:583: FutureWarning: elementwise comparison
stead, but in the future will perform elementwise comparison
mask |= (ar1 == a)

In [3]:

```
1 df.info()
```

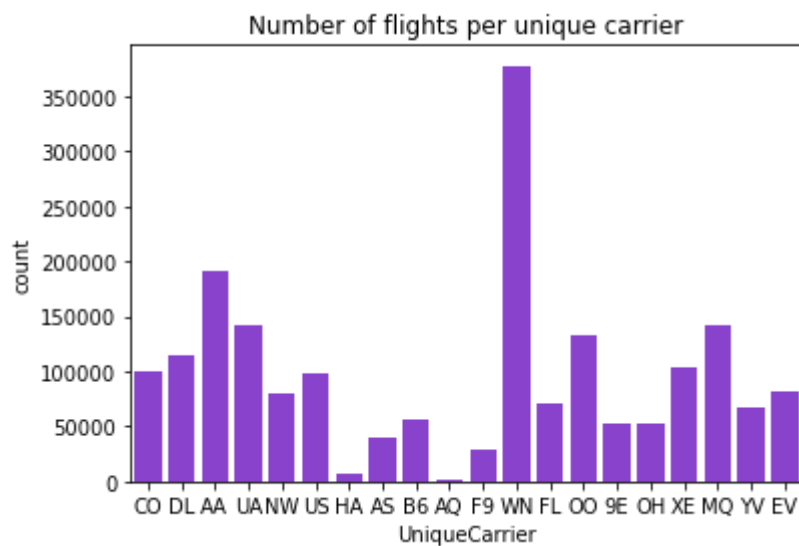
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1936758 entries, 3561206 to 4392215
Data columns (total 31 columns):
 #   Column                Dtype
---  -
 0   Month                 int64
 1   DayOfMonth            int64
 2   DayOfWeek             int64
 3   DepTime               float64
 4   CRSDepTime            int64
 5   ArrTime               float64
 6   CRSArrTime            int64
 7   UniqueCarrier         object
 8   FlightNum             int64
 9   TailNum               object
10   ActualElapsedTime     float64
11   CRSElapsedTime        float64
12   AirTime               float64
13   ArrDelay              float64
14   DepDelay              float64
15   Distance              int64
16   TaxiIn                float64
17   TaxiOut               float64
18   Diverted              int64
19   CarrierDelay          float64
20   WeatherDelay          float64
21   NASDelay              float64
22   SecurityDelay         float64
23   LateAircraftDelay     float64
24   DistanceKm            float64
25   AirTimeH              float64
26   FlightSpeed           float64
27   LateLanding           bool
28   LateTakeOff           bool
29   ElapsedTimeDifference float64
30   ArrivalDifference     float64
dtypes: bool(2), float64(19), int64(8), object(2)
memory usage: 447.0+ MB
```

Crea almenys una visualització per:

- Una variable categòrica (UniqueCarrier)

In [16]:

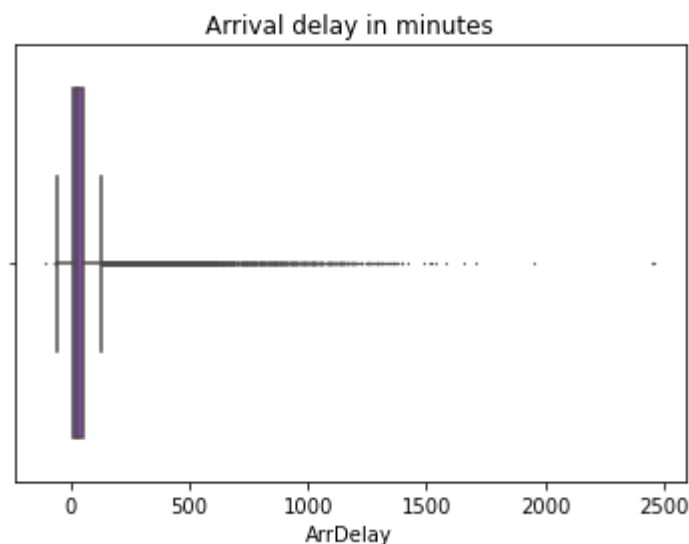
```
1 sns.countplot( x = "UniqueCarrier", data = df, color = "BlueViolet").set(  
2     title = "Number of flights per unique carrier")  
3  
4 plt.savefig("flights_carrier.png")
```



- Una variable numèrica (ArrDelay)

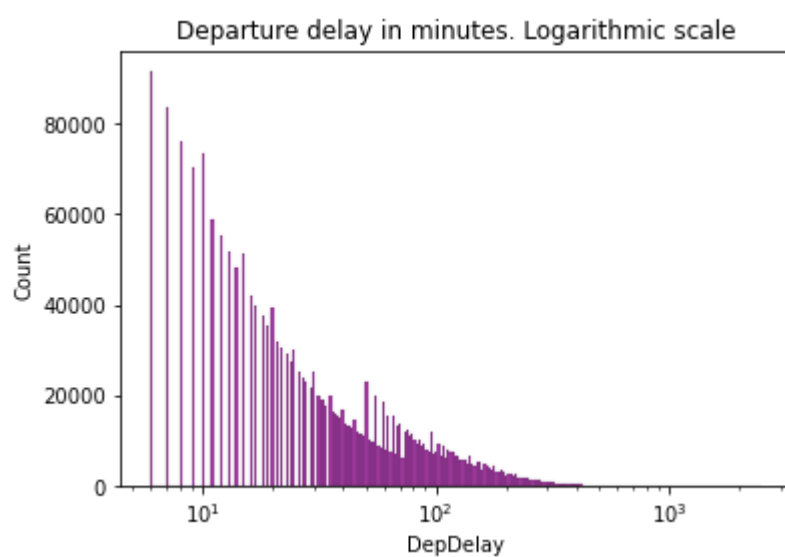
In [17]:

```
1 sns.boxplot(x = "ArrDelay", data = df, color = "DarkOrchid", fliersize = 0.5).set(  
2     title = "Arrival delay in minutes")  
3  
4 plt.savefig("arrdelay.png")
```



In [18]:

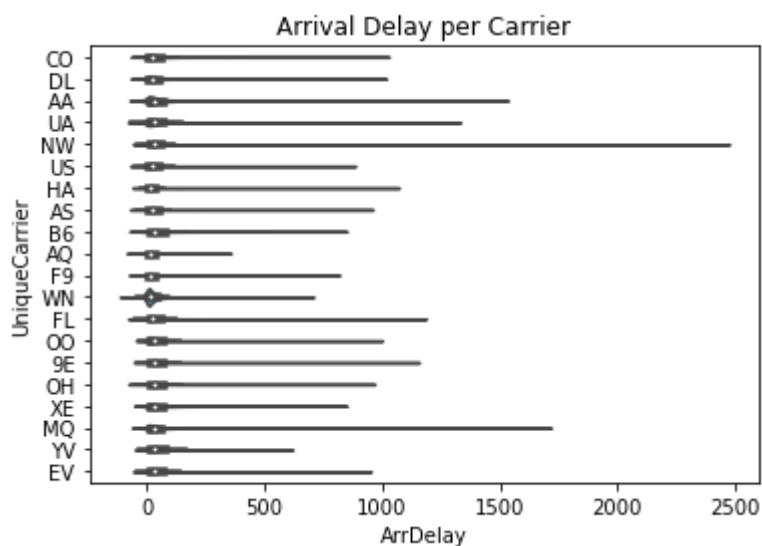
```
1 sns.histplot(df, x = "DepDelay", color = "DarkMagenta", log_scale = True).set(  
2     title = "Departure delay in minutes. Logarithmic scale")  
3  
4 plt.savefig("depdelay.png")
```



- Una variable numèrica i una categòrica (ArrDelay i UniqueCarrier)

In [19]:

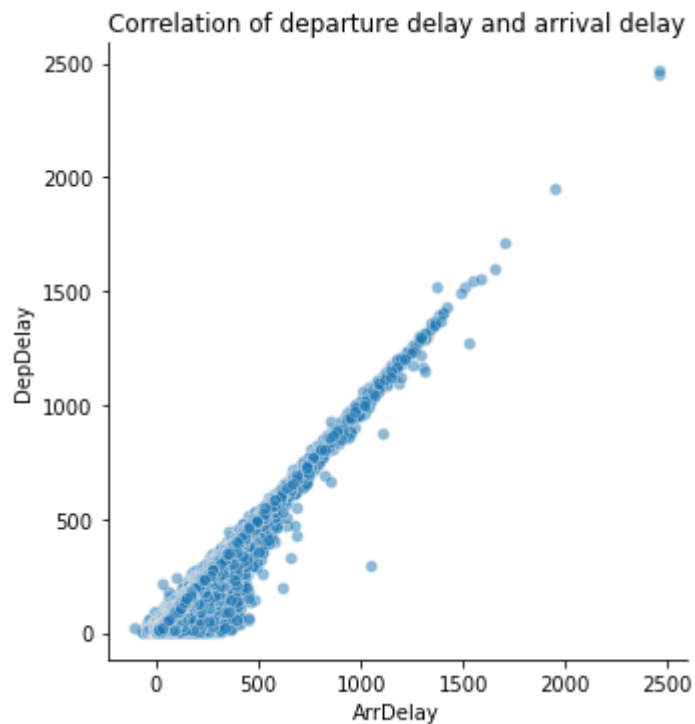
```
1 sns.violinplot(x = "ArrDelay", y = "UniqueCarrier", data = df, scale = "count").se  
2     title = "Arrival Delay per Carrier")  
3  
4 plt.savefig("arrdelay_carrier.png")
```



- Dues variables numèriques (ArrDelay i DepDelay)

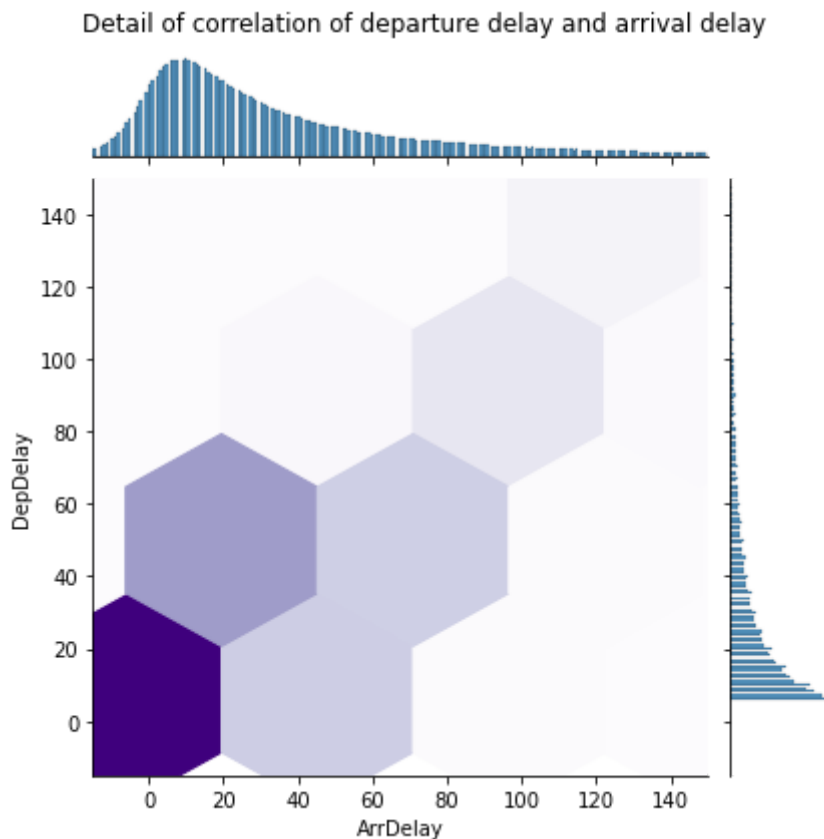
In [3]:

```
1 sns.relplot(x = "ArrDelay", y = "DepDelay", data = df, alpha = 0.5).set(  
2     title = "Correlation of departure delay and arrival delay")  
3  
4 plt.savefig("depdelay_arrdelay.png")
```



In [4]:

```
1 plot = sns.jointplot(x = "ArrDelay", y = "DepDelay", data = df, cmap = "Purples",
2                       xlim = (-15, 150), ylim = (-15, 150))
3 plot.fig.suptitle("Detail of correlation of departure delay and arrival delay")
4 plot.fig.subplots_adjust(top = 0.93)
5 plt.show()
6
7 plt.savefig("detail_depdelay_arrdelay.png")
```

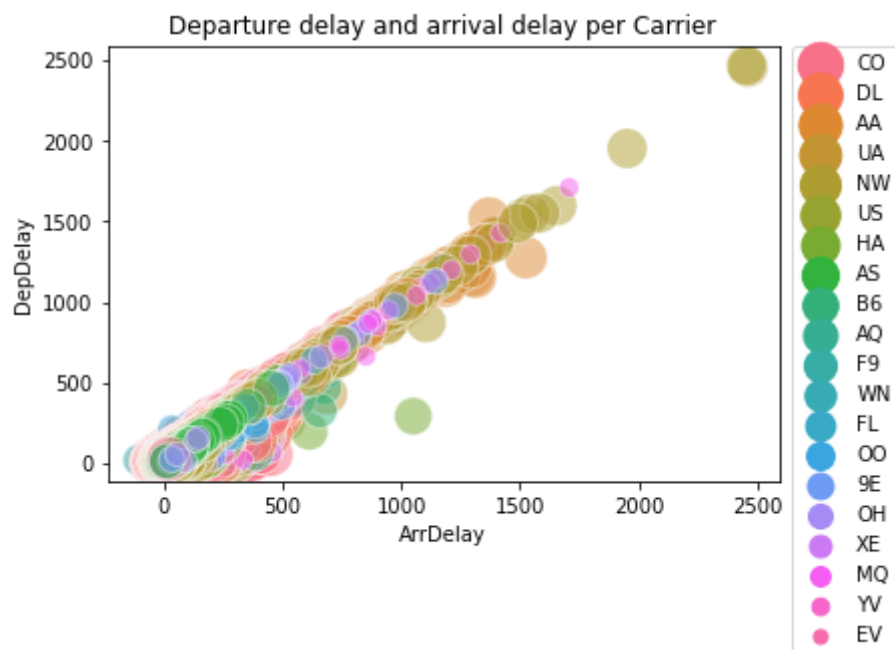


<Figure size 432x288 with 0 Axes>

- Tres variables (ArrDelay, DepDelay i UniqueCarrier)

In [5]:

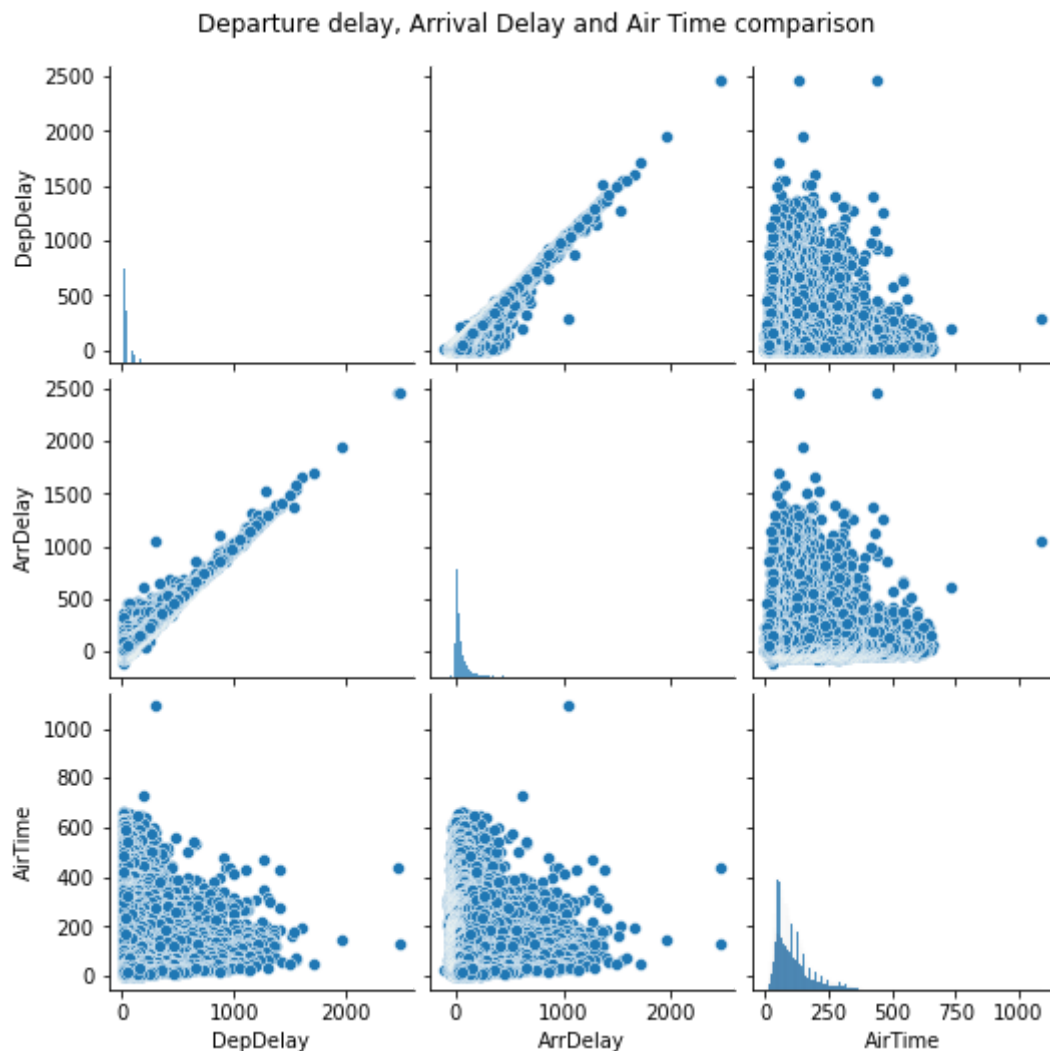
```
1 sns.scatterplot(data = df, x = "ArrDelay", y = "DepDelay", size = "UniqueCarrier",  
2                 s = 20, sizes = (50, 500), hue = "UniqueCarrier"  
3                 ).set(title = "Departure delay and arrival delay per Carrier")  
4 plt.legend(bbox_to_anchor = (1.02, 1), loc = "upper left", borderaxespad = 0)  
5 plt.show()  
6  
7 plt.savefig("depdelay_arrdelay_carrier.png")
```



<Figure size 432x288 with 0 Axes>

In [22]:

```
1 plot = sns.pairplot(df[["DepDelay", "ArrDelay", "AirTime"]])
2 plot.fig.suptitle("Departure delay, Arrival Delay and Air Time comparison")
3 plot.fig.subplots_adjust(top = 0.93)
4 plt.savefig("depdelay_arrdelay_airtime.png")
```

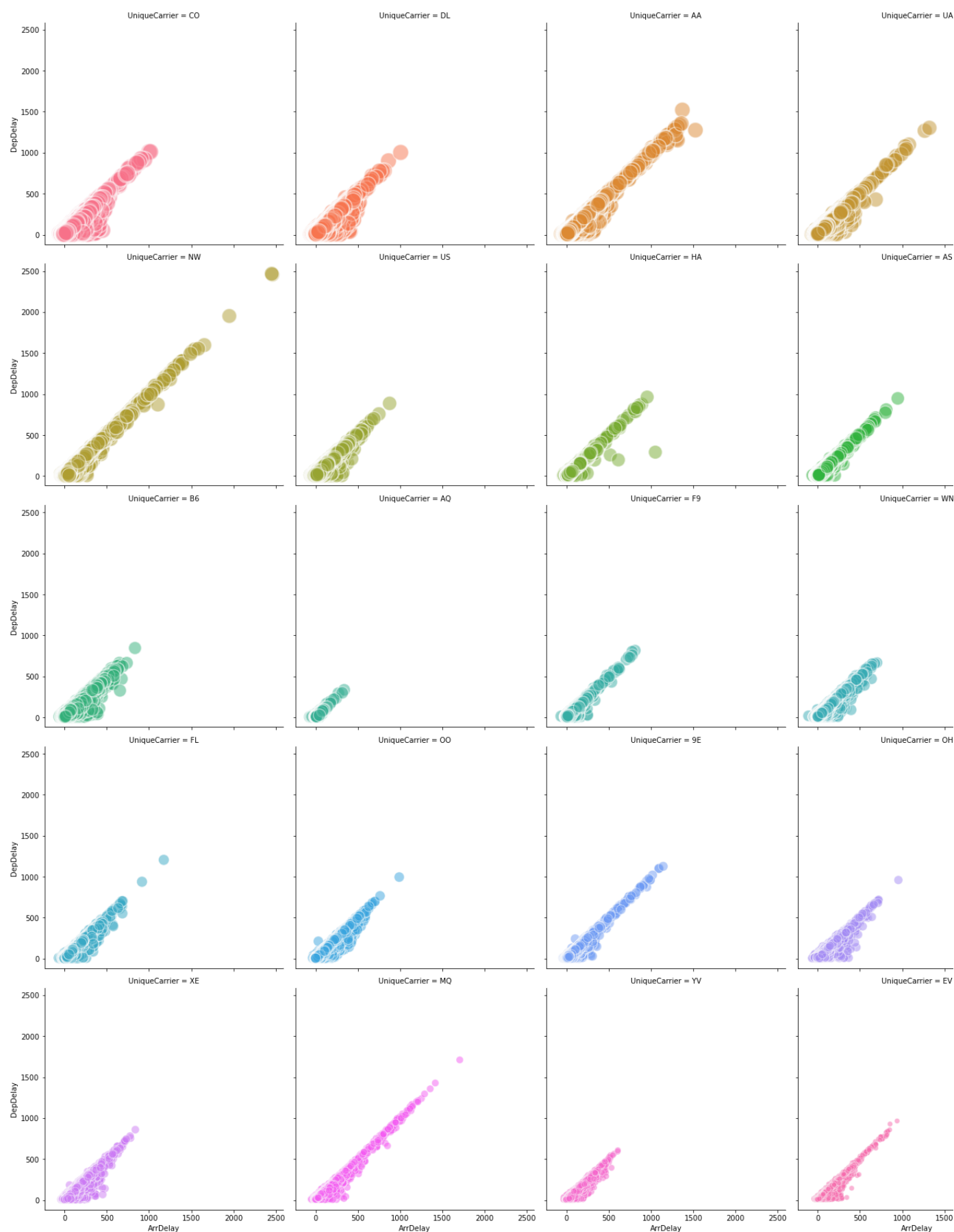


In [19]:

```
1 plot = sns.relplot(data = df, x = "ArrDelay", y = "DepDelay", size = "UniqueCarrier",
2                     s = 20, sizes = (50, 500), hue = "UniqueCarrier", col = "UniqueCarrier")
3
4 plt.legend(bbox_to_anchor = (1.02, 1), loc = "upper left", borderaxespad = 0)
5 plot.fig.suptitle("Departure delay and arrival delay per Carrier", fontsize = 50)
6 plot.fig.subplots_adjust(top = 0.93)
7 plt.show()
8
9 plt.savefig("depdelay_arrdelay_carrier_sep.png")
```

No handles with labels found to put in legend.

Departure delay and arrival delay per Carrier

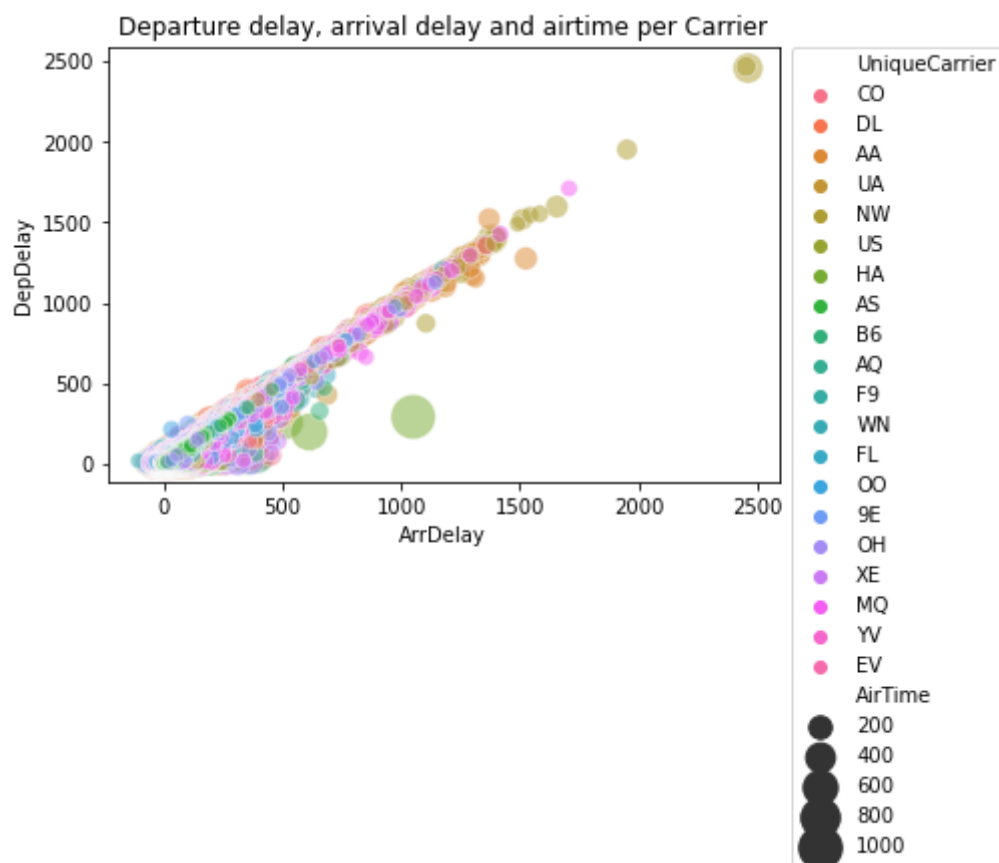


<Figure size 432x288 with 0 Axes>

- Més de tres variables (ArrDelay, DepDelay, AirTime i UniqueCarrier).

In [5]:

```
1 sns.scatterplot(data = df, x = "ArrDelay", y = "DepDelay", hue = "UniqueCarrier",  
2                 alpha = 0.5, s = 20, sizes = (50, 500)  
3                 ).set(title = "Departure delay, arrival delay and airtime per Carri  
4 plt.legend(bbox_to_anchor = (1.02, 1), loc = "upper left", borderaxespad = 0)  
5 plt.show()  
6  
7 plt.savefig("depdelay_arrdelay_carrier_airtime.png")
```



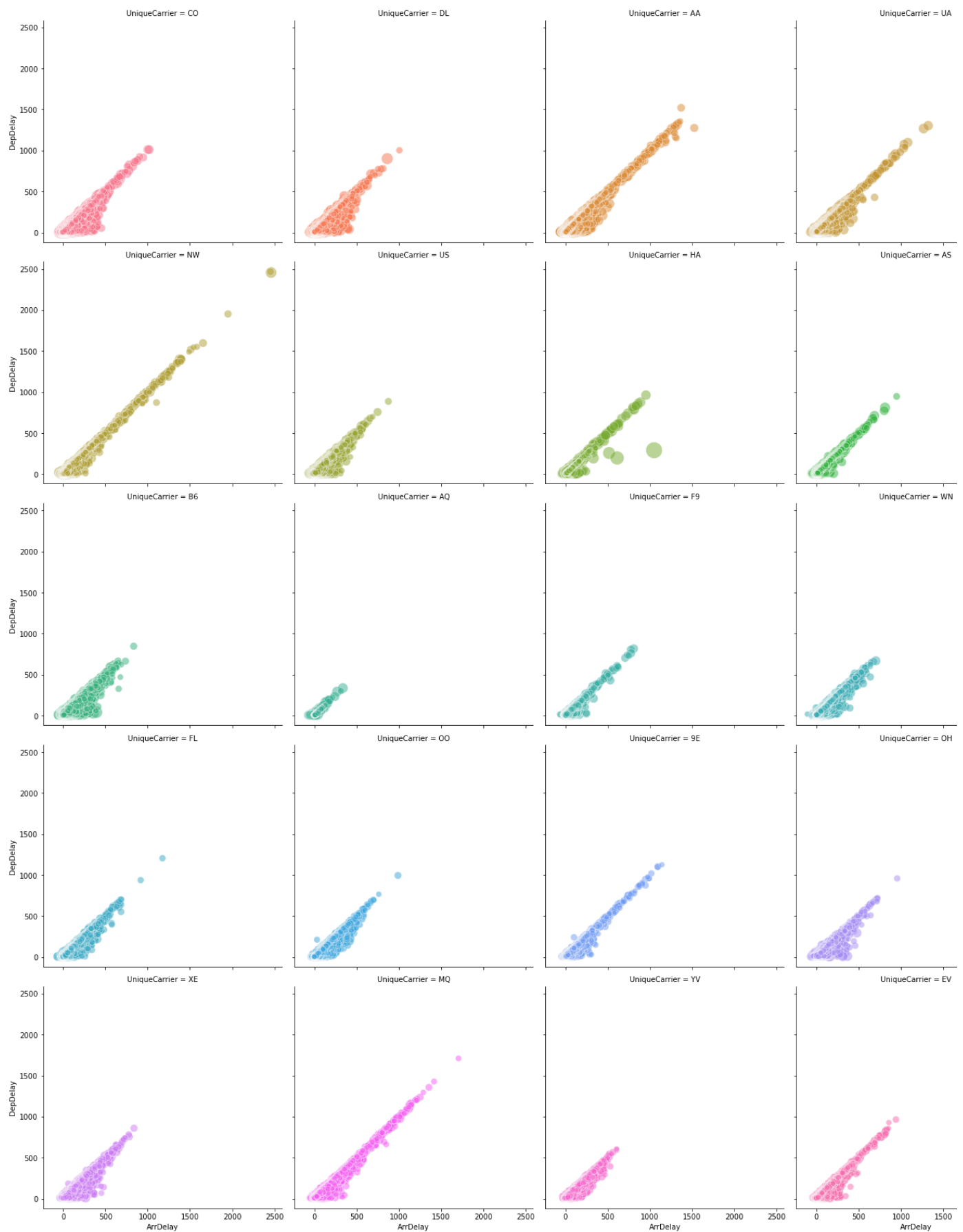
<Figure size 432x288 with 0 Axes>

In [15]:

```
1 plot = sns.relplot(data = df, x = "ArrDelay", y = "DepDelay", hue = "UniqueCarrier",
2                   col = "UniqueCarrier", col_wrap = 4, alpha = 0.5, s = 20, sizes = (50,
3                   )
4 plot.fig.suptitle("Departure delay, arrival delay and airtime per Carrier", fontsize=10)
5 plt.legend(bbox_to_anchor = (1.02, 1), loc = "upper left", borderaxespad = 0, font
6 plot.fig.subplots_adjust(top = 0.93)
7 plt.show()
8
9 plt.savefig("depdelay_arrdelay_carrier_airtime_sep.png")
```

No handles with labels found to put in legend.

Departure delay, arrival delay and airtime per Ca



<Figure size 432x288 with 0 Axes>

