

# 0. Objetivo del Estudio

## 1. Estudio Previo JSON

### 1.1 Estructura JSON

El JSON seleccionado trata de una **base de datos aeronáutica, del regulador estadounidense FAA, donde se recogen, por cada uno de los aeropuerto y aerolíneas de estudio los siguientes datos históricos estadísticos:**

#### 1- Vuelos:

- vuelos cancelados
- vuelos puntuales
- vuelos totales
- vuelos retrasados
- vuelos desviados

#### 2- De los vuelos retrasados, las siguientes causas:

- las debidas al aeronave
- las debidas a la climatología
- las debidas a la seguridad
- las debidas a los sistemas de navegación
- las debidas al transportista

#### 3- Los minutos totales de retraso debido a las causas definidas anteriormente

#### 4- Periodo de estudio tanto en años como en meses

En cuanto **al tamaño**, el JSON seleccionado consta de un total de **54.013 documentos** y su estructura es la que se muestra a continuación.

## Estructura documento JSON

```
**{
  "airport" : {
    "code" : "ATL",
    "name" : "Atlanta, GA: Hartsfield-Jackson Atlanta International"
  },
  "statistics" : {
    "flights" : {
      "cancelled" : 5,
      "on time" : 561,
      "total" : 752,
      "delayed" : 186,
      "diverted" : 0
    },
    "# of delays" : {
      "late aircraft" : 18,
      "weather" : 28,
      "security" : 2,
      "national aviation system" : 105,
      "carrier" : 34
    },
    "minutes delayed" : {
      "late aircraft" : 1269,
      "weather" : 1722,
      "carrier" : 1367,
      "security" : 139,
      "total" : 8314,
      "national aviation system" : 3817
    }
  }
}**
```

## 1.2 Importación de librerías necesarias

Inicialmente se cargan las librerías de python necesarias.

```
In [ ]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")
```

## 1.3 Conexión con MONGO ATLAS

Para una conexión a **MONGO ATLAS** se ejecutaría la siguiente cadena de conexión

```
In [19]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.|
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines
```

Para importar el archivo JSON se ejecutaría el siguiente código en un terminal

```
mongoimport --host SatoClusterFAA-shard-0/satoclusterfaa-shard-00-00-gst6h.azure.mongodb.net:27017,satoclusterfaa-
shard-00-01-gst6h.azure.mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb.net:27017 --ssl --username
sato --password <PASSWORD> --authenticationDatabase admin --db FAA_Airports --collection airlines --type JSON --file
/local_path/FAA_AIRPORTS/airlines.json
```

## 1.4 Conexión con MONGO LOCAL

Para una conexión a **MONGO en local** se ejecutaría la siguiente cadena de conexión

```
In [3]: #local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

Para importar el archivo JSON se ejecutaría el siguiente código en un terminal

```
mongoimport -v --host localhost:27017 -d "airports" -c "airlines" --file /local_path/FAA_AIRPORTS/airlines.json
--type json
```

## 1.5 Características básicas de la colección

### *Periodo disponible en estudio*

Con esta consulta se trata de averiguar el período de estudio que incluye la colección

```
In [8]: time_label = db.airlines.distinct("time.label")
```

El código `air.distinct("time.label")` da como resultado una lista de **time.labels** única de la siguiente forma:

```
['2003/6', '2003/7', '2003/8', '2003/9', '2003/10', '2003/11', '2003/12', '2004/1', '2004/2', '2004/3', '2004/4', '2004/5', '2004/6', '2004/7', '2004/8', '2004/9', '2004/10',
'2004/11', '2004/12', '2005/1', '2005/2', '2005/3', '2005/4', '2005/5', '2005/6', '2005/7', '2005/8', '2005/9', '2005/10', '2005/11', '2006/8', '2005/12', '2006/1',
'2006/2', '2006/3', '2006/4', '2006/5', '2006/6', '2006/7', '2006/9', '2006/10', '2006/11', '2006/12', '2007/1', '2007/2', '2007/3', '2007/4', '2007/5', '2007/6', '2007/7',
'2007/8', '2007/9', '2007/10', '2007/11', '2007/12', '2008/1', '2008/2', '2008/3', '2008/4', '2008/5', '2008/6', '2008/7', '2008/8', '2008/9', '2008/10', '2008/11',
'2008/12', '2009/1', '2009/2', '2009/3', '2009/4', '2009/5', '2009/6', '2009/7', '2009/8', '2009/9', '2009/10', '2009/11', '2009/12', '2010/1', '2010/2', '2010/3', '2010/4',
'2010/5', '2010/6', '2010/7', '2010/8', '2010/9', '2010/10', '2010/11', '2010/12', '2011/1', '2011/2', '2011/3', '2011/4', '2011/5', '2011/6', '2011/7', '2011/8', '2011/9',
'2011/10', '2011/11', '2011/12', '2012/1', '2012/2', '2012/3', '2012/4', '2012/5', '2012/6', '2012/7', '2012/8', '2012/9', '2012/10', '2012/11', '2012/12', '2013/1',
'2013/2', '2013/3', '2013/4', '2013/5', '2013/6', '2013/7', '2013/8', '2013/9', '2013/10', '2013/11', '2013/12', '2014/1', '2014/2', '2014/3', '2014/4', '2014/5', '2014/6',
'2014/7', '2014/8', '2014/9', '2014/10', '2014/11', '2014/12', '2015/1', '2015/2', '2015/3', '2015/4', '2015/5', '2015/6', '2015/7', '2015/8', '2015/9', '2015/10',
'2015/11', '2015/12', '2016/1']
```

Con el siguiente código se extraen los años y , por cada uno de ellos, se muestra el número de meses completos existentes en la **Colección**.

```
In [9]: date = []
periodo = {}

for label in time_label:
    date.append(int(label.split("/") [0]))

for n in set(date):
    periodo[n] = date.count(n)

periodo
```

```
Out[9]: {2016: 1,
2003: 7,
2004: 12,
2005: 12,
2006: 12,
2007: 12,
2008: 12,
2009: 12,
2010: 12,
2011: 12,
2012: 12,
2013: 12,
2014: 12,
2015: 12}
```

Este resultado demuestra que el período de tiempo disponible comprende desde 2003 al 2016 . Aún así, el estudio no contempla los 12 meses para los años 2003 y 2016 por lo que, a partir de ahora, no se contemplan dichos años en el estudio. Aún así, se mantienen en la **Colección**.

## Aeropuertos de estudio

Con esta consulta se trata de averiguar los distintos aeropuertos que comprenden el estudio

```
In [12]: len(db.airlines.distinct("airport.name"))
```

```
Out[12]: 29
```

```
In [15]: db.airlines.distinct("airport.name")
```

```
Out[15]: ['Atlanta, GA: Hartsfield-Jackson Atlanta International',
 'Boston, MA: Logan International',
 'Baltimore, MD: Baltimore/Washington International Thurgood Marshall',
 'Charlotte, NC: Charlotte Douglas International',
 'Washington, DC: Ronald Reagan Washington National',
 'Denver, CO: Denver International',
 'Dallas/Fort Worth, TX: Dallas/Fort Worth International',
 'Detroit, MI: Detroit Metro Wayne County',
 'Newark, NJ: Newark Liberty International',
 'Fort Lauderdale, FL: Fort Lauderdale-Hollywood International',
 'Washington, DC: Washington Dulles International',
 'Houston, TX: George Bush Intercontinental/Houston',
 'New York, NY: John F. Kennedy International',
 'Las Vegas, NV: McCarran International',
 'Los Angeles, CA: Los Angeles International',
 'New York, NY: LaGuardia',
 'Orlando, FL: Orlando International',
 'Chicago, IL: Chicago Midway International',
 'Miami, FL: Miami International',
 'Minneapolis, MN: Minneapolis-St Paul International',
 "Chicago, IL: Chicago O'Hare International",
 'Portland, OR: Portland International',
 'Philadelphia, PA: Philadelphia International',
 'Phoenix, AZ: Phoenix Sky Harbor International',
 'San Diego, CA: San Diego International',
 'Seattle, WA: Seattle/Tacoma International',
 'San Francisco, CA: San Francisco International',
 'Salt Lake City, UT: Salt Lake City International',
 'Tampa, FL: Tampa International']
```

## Compañías Aéreas de Estudio

Con esta consulta se trata de averiguar las distintas **aerolíneas** que comprenden el estudio

```
In [17]: len(db.airlines.distinct("carrier.name"))
```

```
Out[17]: 28
```

```
In [18]: db.airlines.distinct("carrier.name")
```

```
Out[18]: ['American Airlines Inc.',  
          'Alaska Airlines Inc.',  
          'JetBlue Airways',  
          'Continental Air Lines Inc.',  
          'Atlantic Coast Airlines',  
          'Delta Air Lines Inc.',  
          'Atlantic Southeast Airlines',  
          'AirTran Airways Corporation',  
          'America West Airlines Inc.',  
          'American Eagle Airlines Inc.',  
          'Northwest Airlines Inc.',  
          'SkyWest Airlines Inc.',  
          'ExpressJet Airlines Inc.',  
          'ATA Airlines d/b/a ATA',  
          'United Air Lines Inc.',  
          'US Airways Inc.',  
          'Southwest Airlines Co.',  
          'Hawaiian Airlines Inc.',  
          'Comair Inc.',  
          'Independence Air',  
          'Frontier Airlines Inc.',  
          'Mesa Airlines Inc.',  
          'Aloha Airlines Inc.',  
          'Pinnacle Airlines Inc.',  
          'Virgin America',  
          'Endeavor Air Inc.',  
          'Envoy Air',  
          'Spirit Air Lines']
```

```
In [ ]:
```

# 1. Distribución anual de Vuelos Totales, Retrasados y Cancelados. Valores medios, máximos y mínimos.

## 1.1 Importación de librerías necesarias

Inicialmente se cargan las librerías de python necesarias.

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")
```

## 1.2 Conexión con MONGO ATLAS / LOCAL

```
In [41]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.||
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb\|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines

#local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

Se crea la variable **air** para facilitar las consultas

```
In [3]: air = db.airlines
```

## 1.3 Distribución anuales.

Con este consulta se pretende obtener la **distibución anual** de:

- Vuelos Totales
- Vuelos Retrasados
- Vuelos Cancelados

Los pasos a seguir son los siguientes.

1. Se seleccionan los años que sean distintos a 2003 y 2016.
2. Se agrupa por año y se calcula la suma total de los Vuelos Totales, Vuelos Retrasados y Vuelos Cancelados.

- **Query 1**

```
In [4]: pipeline1 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                    {"$group": {"_id": "$time.year",
                               "Totales": {"$sum": "$statistics.flights.total"},
                               "Retrasados": {"$sum": "$statistics.flights.delayed"},
                               "Cancelados": {"$sum": "$statistics.flights.cancelled"}}
                    }
                  ]
curs1 = air.aggregate(pipeline1)
```

Con el cursor obtenido de la consulta, se crea un **DataFrame** con los resultados de la misma.

```
In [5]: query1 = list(curs1)
```

```
In [6]: df = pd.DataFrame(query1).set_index(['_id']).sort_index()
df
```

Out[6]:

_id	Cancelados	Retrasados	Totales
2004.0	77940.0	880677.0	4344735.0
2005.0	83188.0	925578.0	4373522.0
2006.0	78475.0	1024612.0	4437952.0
2007.0	102665.0	1129439.0	4538488.0
2008.0	87650.0	965136.0	4307649.0
2009.0	58341.0	787472.0	4038900.0
2010.0	75018.0	732445.0	4050772.0
2011.0	77467.0	715560.0	3923295.0
2012.0	53494.0	658326.0	3955389.0
2013.0	62054.0	805063.0	4108397.0
2014.0	81908.0	796314.0	3824651.0
2015.0	59770.0	707800.0	3870278.0

A partir de lo anterior y con las siguientes consultas se pretende obtener los **valores medios, máximos y mínimos** de los **Vuelos Totales, Vuelos Retrasados y Vuelos Cancelados** durante el período de estudio. Los pasos a seguir son los siguientes.

1. Se seleccionan los años que sean distintos a 2003 y 2016.
2. Se agrupa por año y se calcula la **suma total de los Vuelos Totales, Vuelos Retrasados y Vuelos Cancelados**.

Con la siguiente query se genera la nueva colección **Indicadores\_Anuales** que se guarda en la colección **Indicadores\_Anuales**.

```
In [7]: pipelinela = [{"$match": {"time.year": {"$nin": [2003, 2016]}},  
                    {"$group": {"_id": "$time.year",  
                               "Totales": {"$sum": "$statistics.flights.total"},  
                               "Retrasados": {"$sum": "$statistics.flights.delayed"},  
                               "Cancelados": {"$sum": "$statistics.flights.cancelled"}  
                           }  
                  },  
                    {"$out": "Indicadores_Anuales"}  
                ]  
air.aggregate(pipelinel)
```

```
In [8]: indicadores = db.Indicadores_Anuales
```

#### • Query 2. Valores medios

Con la siguiente consulta sobre la **nueva colección Indicadores\_Anuales** se obtiene las medias.

```
In [9]: curs2 = indicadores.aggregate([{"$group": {"_id": "null",  
                                              "Media_Ind_Retrasados": {"$avg": "$Retrasados"},  
                                              "Media_Ind_Cancelados": {"$avg": "$Cancelados"},  
                                              "Media_Ind_Totales": {"$avg": "$Totales"}  
                                         }  
                                         },  
                                         {"$project": {"_id": 0}}  
                                         ])
```

```
In [10]: Medias_Ind = list(curs2)[0]  
Medias_Ind
```

```
Out[10]: {'Media_Ind_Retrasados': 844035.1666666666,  
          'Media_Ind_Cancelados': 74830.833333333333,  
          'Media_Ind_Totales': 4147835.6666666665}
```

Para calcular los **años donde los vuelos retrasados son mayores a la media**

```
In [11]: year_media_Retrasados = indicadores.find({"Retrasados": {"$gt": Medias_Ind["Media_Ind_Retrasados"]}}, {"_id": 1})  
  
In [12]: year_media_Retrasados = list(year_media_Retrasados)  
year_media_Retrasados  
  
Out[12]: [ {_id': 2007.0},  
          {_id': 2005.0},  
          {_id': 2004.0},  
          {_id': 2006.0},  
          {_id': 2008.0}]
```

Para calcular los **años donde los vuelos cancelados son mayores a la media**

```
In [13]: year_media_Cancelados = indicadores.find({"Cancelados": {"$gt": Medias_Ind["Media_Ind_Cancelados"]}}, {"_id": 1})  
  
In [14]: year_media_Cancelados = list(year_media_Cancelados)  
year_media_Cancelados  
  
Out[14]: [ {_id': 2014.0},  
          {_id': 2007.0},  
          {_id': 2005.0},  
          {_id': 2004.0},  
          {_id': 2010.0},  
          {_id': 2006.0},  
          {_id': 2011.0},  
          {_id': 2008.0}]
```

Para calcular los **años donde los vuelos totales son mayores a la media**

```
In [15]: year_media_Totales = indicadores.find({"Totales": {"$gt": Medias_Ind["Media_Ind_Totales"]}}, {"_id": 1})  
  
In [16]: year_media_Totales = list(year_media_Totales)  
year_media_Totales  
  
Out[16]: [ {_id': 2007.0},  
          {_id': 2005.0},  
          {_id': 2004.0},  
          {_id': 2006.0},  
          {_id': 2008.0}]
```

### • Query 3. Valores máximos

Con la siguiente consulta sobre la **nueva colección Indicadores\_Anuales** se obtiene los máximos.

```
In [17]: curs3 = indicadores.aggregate([{"$group": {"_id": "null",  
                                         "Max_Ind_Retrasados": {"$max": "$Retrasados"},  
                                         "Max_Ind_Cancelados": {"$max": "$Cancelados"},  
                                         "Max_Ind_Totales": {"$max": "$Totales"}  
                                         }  
                                         },  
                                         {"$project": {"_id": 0}}  
                                         ])  
  
In [18]: Max_Ind = list(curs3)[0]  
Max_Ind  
  
Out[18]: {'Max_Ind_Retrasados': 1129439.0,  
          'Max_Ind_Cancelados': 102665.0,  
          'Max_Ind_Totales': 4538488.0}
```

Para calcular el **año donde se produce el máximo de los vuelos retrasados**

```
In [19]: year_max_Retrasados = indicadores.find({"Retrasados": Max_Ind["Max_Ind_Retrasados"]}, {"_id": 1})
```

```
In [20]: year_max_Retrasados = list(year_max_Retrasados)
```

```
year_max_Retrasados
```

```
Out[20]: [ {_id': 2007.0}]
```

Para calcular el **año donde se produce el máximo de los vuelos Cancelados**

```
In [21]: year_max_Cancelados = indicadores.find({"Cancelados": Max_Ind["Max_Ind_Cancelados"]}, {"_id": 1})
```

```
year_max_Cancelados
```

```
Out[22]: [ {_id': 2007.0}]
```

Para calcular el **año donde se produce el máximo de los vuelos Totales**

```
In [23]: year_max_Totales = indicadores.find({"Totales": Max_Ind["Max_Ind_Totales"]}, {"_id": 1})
```

```
year_max_Totales
```

```
Out[24]: [ {_id': 2007.0}]
```

- **Query 4. Valores mínimos**

Con la siguiente consulta sobre la **nueva colección Indicadores\_Anuales** se obtiene los mínimos.

```
In [25]: curs4 = indicadores.aggregate([{"$group": {"_id": "null",  
"Min_Ind_Retrasados": {"$min": "$Retrasados"},  
"Min_Ind_Cancelados": {"$min": "$Cancelados"},  
"Min_Ind_Totales": {"$min": "$Totales"}  
},  
{"$project": {"_id": 0}}  
])  
Min_Ind =list(curs4)[0]  
Min_Ind
```

```
Out[25]: {'Min_Ind_Retrasados': 658326.0,  
'Min_Ind_Cancelados': 53494.0,  
'Min_Ind_Totales': 3824651.0}
```

Para calcular el **año donde se produce el mínimo de los vuelos retrasados**

```
In [26]: year_min_Retrasados = indicadores.find({"Retrasados": Min_Ind["Min_Ind_Retrasados"]}, {"_id": 1})
```

```
year_min_Retrasados
```

```
Out[27]: [ {_id': 2012.0}]
```

Para calcular el **año donde se produce el mínimo de los vuelos Cancelados**

```
In [28]: year_min_Cancelados = indicadores.find({"Cancelados": Min_Ind["Min_Ind_Cancelados"]}, {"_id": 1})
```

```
year_min_Cancelados
```

```
Out[29]: [ {_id': 2012.0}]
```

Para calcular el **año donde se produce el mínimo de los vuelos Totales**

```
In [30]: year_min_Totales = indicadores.find({"Totales": Min_Ind["Min_Ind_Totales"]}, {"_id": 1})
```

```
year_min_Totales
```

```
Out[31]: [ {_id': 2014.0}]
```

Finalmente, y para corroborar los resultados anteriores, se crean los gráficos correspondientes a las evoluciones:

- **Vuelos Cancelados**
- **Vuelos Retrasados**
- **Vuelos Totales**

```
In [32]: fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df.index, y= df.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("CANCELADOS");
ax1.set_xlabel("AÑOS");

sns.barplot(x=df.index, y= df.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("RETRASADOS");
ax2.set_xlabel("AÑOS");

sns.barplot(x=df.index, y= df.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("TOTALES");
ax3.set_xlabel("AÑOS");
```

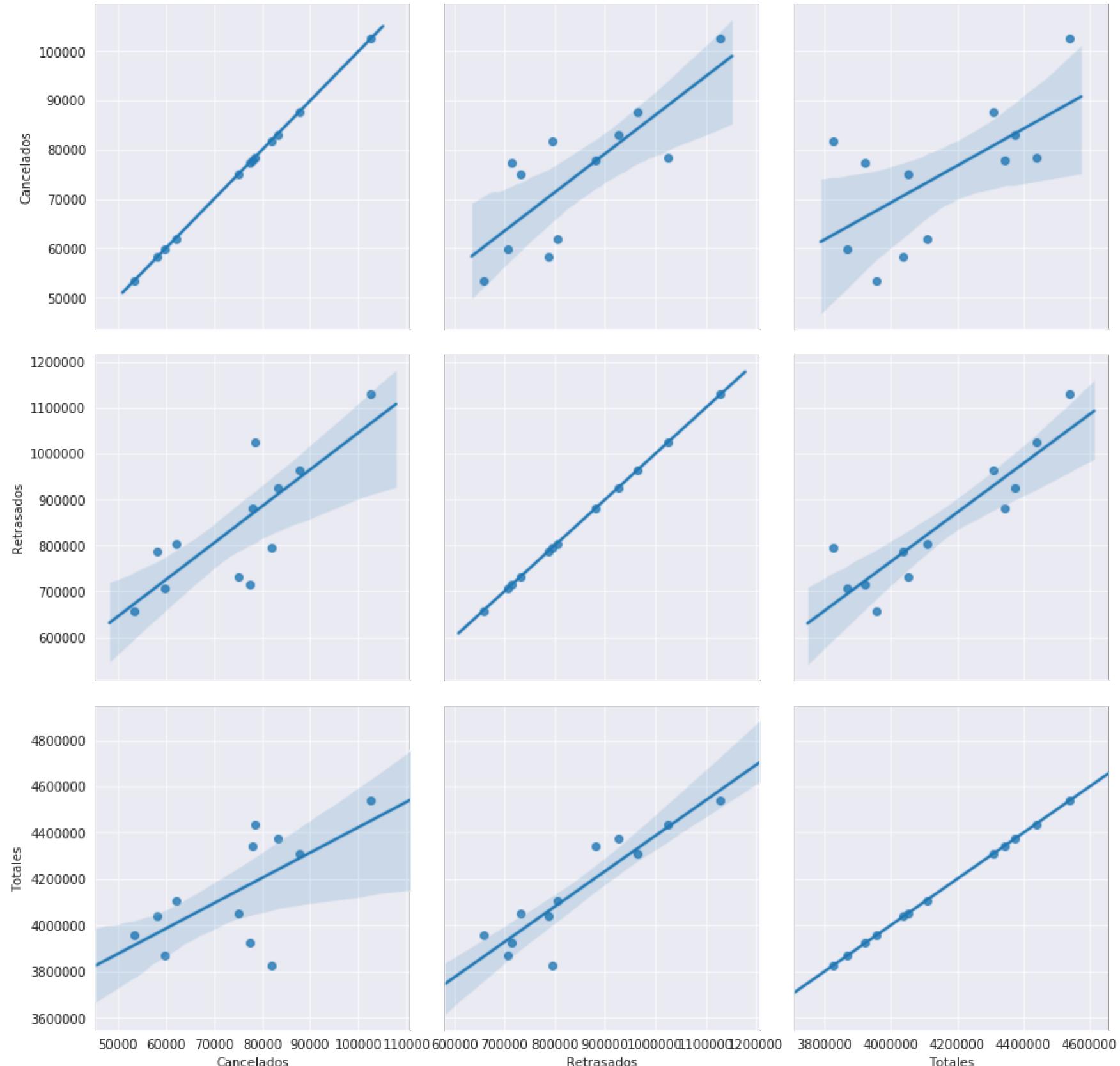


## 2. Relación entre Vuelos Totales y Retrasados, Vuelos Totales y Cancelados y Vuelos Totales y Cancelados.

De la **Query 1**, se toman los valores de los vuelos totales, retrasados y cancelados y se analiza la posible correlación entre ellas. Se busca una posible relación lineal.

```
In [33]: g = sns.PairGrid(df, height= 4);
g.map(sns.regplot);
```

```
/home/sato/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



Con los gráficos anteriores se pone de manifiesto la **correlación (positiva)** existente entre:

- Vuelos Totales y Cancelados
- Vuelos Totales y Retrasados
- Vuelos Cancelados y Retrasados

Es decir, **a mayor volumen de tráfico mayor número en los retrasos y cancelaciones. Además, a mayor número de retrasos mayor número de cancelaciones.**

### 3. Evolución mensual entre 2014-2015 de los Minutos de retraso, Ratio de vuelos cancelados, Ratio de vuelos retrasados y vuelos totales

En este apartado, se analiza la **evolución mensual de dichos indicadores durante los años 2014-2015. Se diseña la siguiente consulta:**

- se filtra los años de estudio
- se calculan los indicadores

```
In [34]: curs5 = air.aggregate([
    {"$match": {"time.year": {"$in": [2014, 2015]}}},
    {"$group": {"_id": {"Year": "$time.year",
                        "Month": "$time.month"},
                "minTotales": {"$sum": "$statistics.minutes delayed.total"},
                "Totales": {"$sum": "$statistics.flights.total"},
                "Cancelados": {"$sum": "$statistics.flights.cancelled"},
                "Retrasados": {"$sum": "$statistics.flights.delayed"}}
    },
    {"$project": {"Year": "$_id.Year",
                  "Month": "$_id.Month",
                  "minTotales": "$minTotales",
                  "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]},
                  "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]},
                  "Vuelos_Totales": "$Totales",
                  "_id": 0
    }
},
 {"$sort": {"Year": 1, "Month": 1}}
])

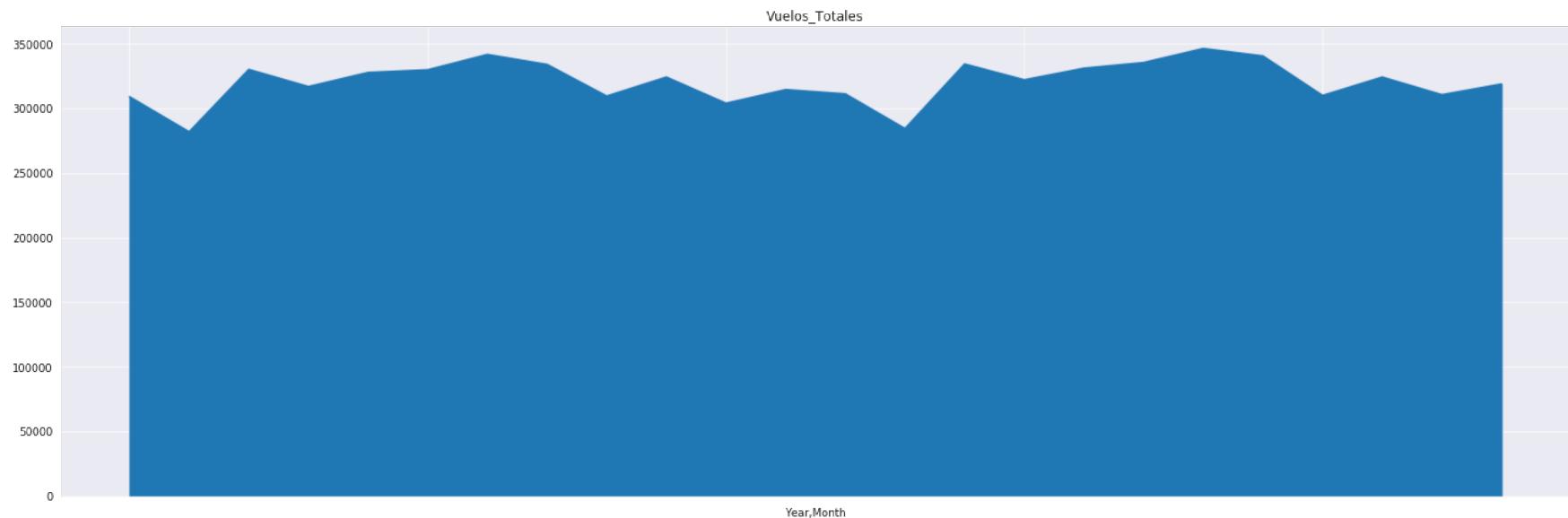
```

```
In [35]: query5 = list(curs5)
```

```
In [36]: df2 = pd.DataFrame(query5).set_index(["Year", "Month"])
```

Se crea el gráfico que nos da la **evolución de Vuelos Totales**

```
In [37]: df2["Vuelos_Totales"].plot(kind= "area", figsize=(25,8), title = "Vuelos_Totales");
```



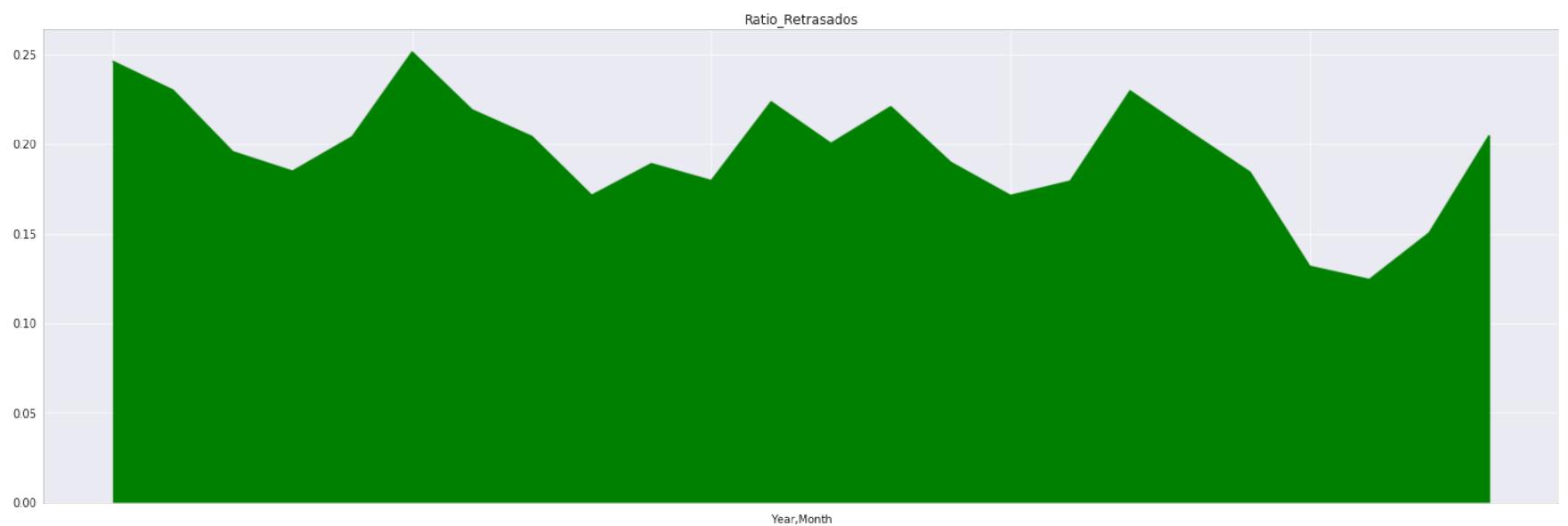
Se crea el gráfico que nos da la **evolución de los Minutos Totales**

```
In [38]: df2["minTotales"].plot(kind= "area", figsize=(25,8), title = "Minutos_Totales", color = "red");
```



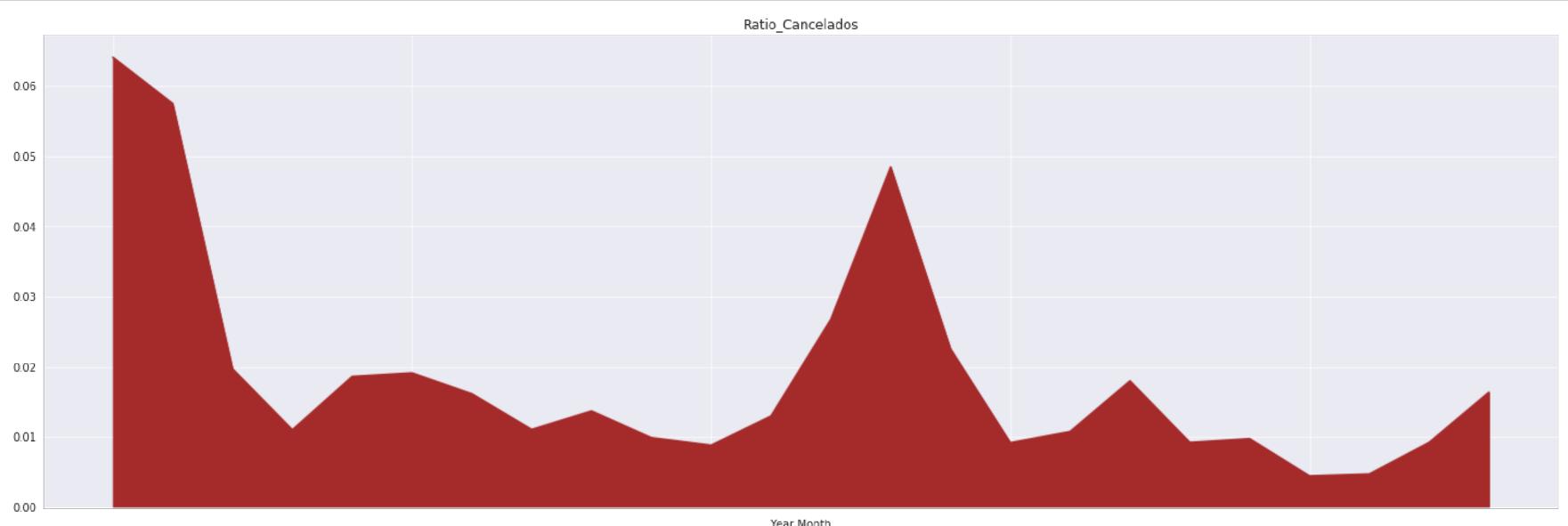
Se crea el gráfico que nos da la **evolución del Ratio Retrasado**

```
In [39]: df2["Ratio_Retrasados"].plot(kind= "area", figsize=(25,8), title = "Ratio_Retrasados", color = "green")  
;
```



Se crea el gráfico que nos da la **evolución del Ratio Cancelados**

```
In [40]: df2["Ratio_Cancelados"].plot(kind= "area", figsize=(25,8), title = "Ratio_Cancelados", color = "brown")  
;
```



```
In [ ]:
```

## 4. Variación anual de Ratios Vuelos Puntuales, Retrasados y Cancelados respecto a los totales.

### 4.1 Importación de librerías necesarias

```
In [2]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")

pd.options.display.float_format = '{:,.2f}'.format
```

### 4.2 Conexión con MONGO ATLAS / LOCAL

```
In [3]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.|
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines

#local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

```
In [3]: air = db.airlines
```

Sería interesante estudiar los **vuelos retrasados, cancelados y puntuales** independientemente del volumen de operaciones existentes ya que, como se ha visto anteriormente, los retrasos y las cancelaciones dependen del volumen de tráfico, esto es, son proporcionales de forma directa.

- **Query 5**

```
In [4]: pipeline5 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                    {"$group": {"_id": "time.year",
                               "Totales": {"$sum": "$statistics.flights.total"},
                               "Retrasados": {"$sum": "$statistics.flights.delayed"},
                               "Cancelados": {"$sum": "$statistics.flights.cancelled"},
                               "Puntuales": {"$sum": "$statistics.flights.on_time"}
                               }
                    },
                    {"$project": {
                               "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]},
                               "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]},
                               "Ratio_Puntuales": {"$divide": ["$Puntuales", "$Totales"]}
                               }
                    }
                  ]
curs5 = air.aggregate(pipeline5)
```

Con el cursor obtenido de la consulta, se crea un **DataFrame** con los resultados de la misma.

```
In [5]: query5 = list(curs5)
```

```
In [9]: df5 = pd.DataFrame(query5).set_index(['_id']).sort_index()
df5.index.name = "Year"
df5
```

Out[9]:

Year	Ratio_Cancelados	Ratio_Puntuales	Ratio_Retrasados
2,004.00	0.02	0.78	0.20
2,005.00	0.02	0.77	0.21
2,006.00	0.02	0.75	0.23
2,007.00	0.02	0.73	0.25
2,008.00	0.02	0.75	0.22
2,009.00	0.01	0.79	0.19
2,010.00	0.02	0.80	0.18
2,011.00	0.02	0.80	0.18
2,012.00	0.01	0.82	0.17
2,013.00	0.02	0.79	0.20
2,014.00	0.02	0.77	0.21
2,015.00	0.02	0.80	0.18

#### 4.3 Gráficos Variación anual de Ratios Vuelos Puntuales, Retrasados y Cancelados respecto a los totales.

Finalmente, se crean los gráficos correspondientes a las evoluciones de:

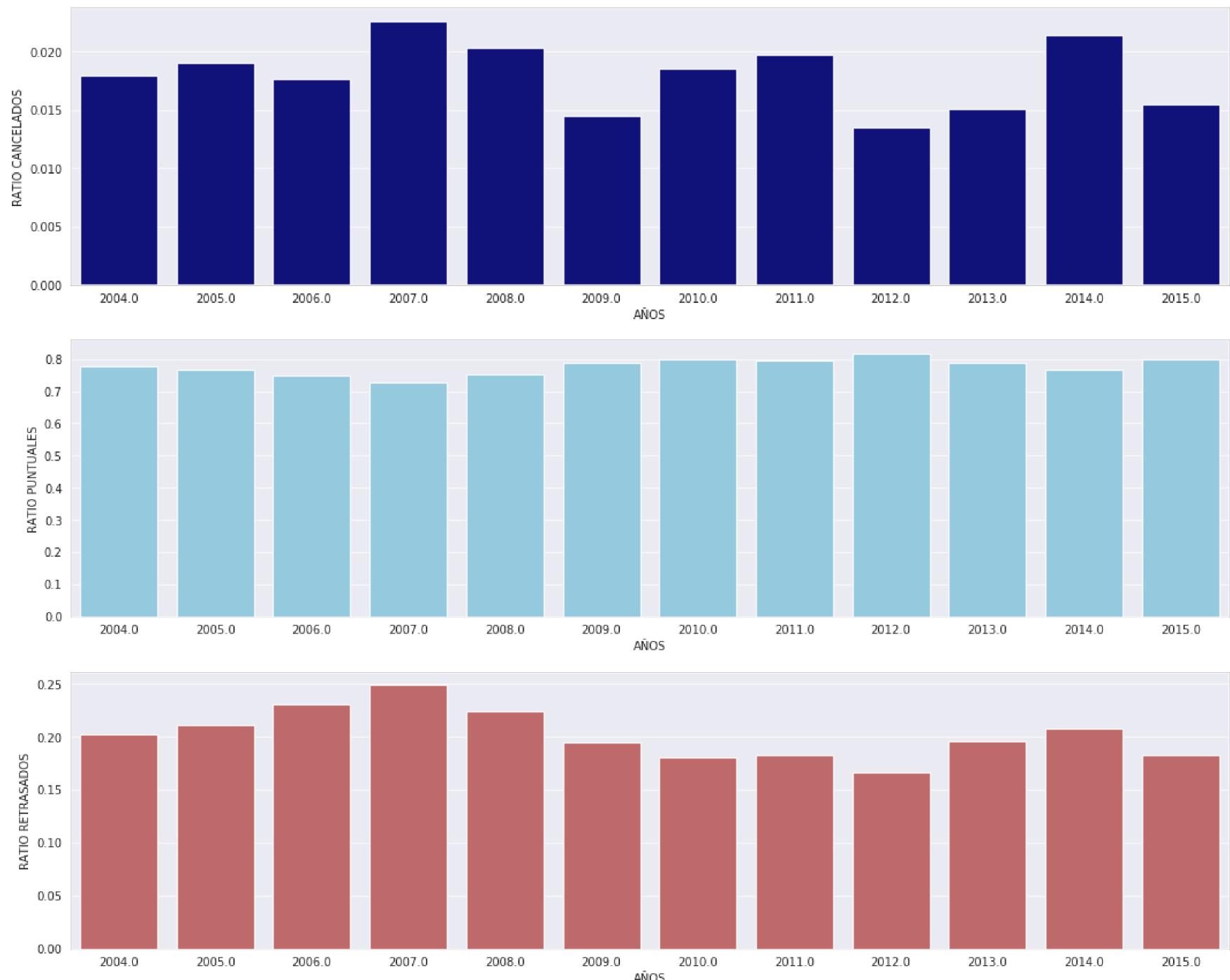
- Ratio Vuelos Cancelados
- Ratio Vuelos Retrasados
- Ratio Vuelos Puntuales

```
In [6]: fig2, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df5.index, y= df5.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("RATIO CANCELADOS");
ax1.set_xlabel("AÑOS");

sns.barplot(x=df5.index, y= df5.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("RATIO PUNTUALES");
ax2.set_xlabel("AÑOS");

sns.barplot(x=df5.index, y= df5.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("RATIO RETRASADOS");
ax3.set_xlabel("AÑOS");
```



#### 4.4 Cálculo de Ratios medios, máximos y mínimos en Vuelos Puntuales, Retrasados y Cancelados respecto a los totales.

Como en el apartado anterior, se pretende obtener los **valores medios, máximos y mínimos** de los Ratios anteriores durante el período de estudio. Los pasos a seguir son los siguientes.

- Se seleccionan los años que sean distintos a 2003 y 2016.
- Se agrupa por año y se calcula los **ratios**.
- Se genera una **nueva colección** con los ratios.
- Se calculan los **valores medios, máximos y mínimos de los ratios anteriores**.

Con la siguiente query se genera la nueva colección **Ratios\_Anuales** que se guarda en **Mongo**.

```
In [7]: pipeline5a = [{"$match": {"time.year": {"$nin": [2003, 2016]}}}, {"$group": {"_id": "$time.year", "Totales": {"$sum": "$statistics.flights.total"}, "Retrasados": {"$sum": "$statistics.flights.delayed"}, "Cancelados": {"$sum": "$statistics.flights.cancelled"}, "Puntuales": {"$sum": "$statistics.flights.on time"}}, {"$project": {"Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]}, "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]}, "Ratio_Puntuales": {"$divide": ["$Puntuales", "$Totales"]}}}, {"$out": "Ratios_Anuales"}]
air.aggregate(pipeline5a);
```

- **Query 6. Ratios Medios**

```
In [8]: ratios = db.Ratios_Anuales
In [9]: pipeline6 = [{"$group": {"_id": "null", "Media_Ratio_Retrasados": {"$avg": "$Ratio_Retrasados"}, "Media_Ratio_Cancelados": {"$avg": "$Ratio_Cancelados"}, "Media_Ratio_Puntuales": {"$avg": "$Ratio_Puntuales"}}, {"$project": {"_id": 0}}]
curs6 = ratios.aggregate(pipeline6)
Medias_Ratio = list(curs6)[0]
Medias_Ratio
Out[9]: {'Media_Ratio_Retrasados': 0.20248100129727428, 'Media_Ratio_Cancelados': 0.017984020651301773, 'Media_Ratio_Puntuales': 0.7771698142878684}
```

Con lo anterior ya se pueden obtener los años en los que:

- **Query 6.1.**

1- Los años en los que los **Cancelaciones fueron mayores en términos relativos** a la media

```
In [10]: cancelMmedia = ratios.find({"Ratio_Cancelados": {"$gt": Medias_Ratio["Media_Ratio_Cancelados"]}}, {"_id": 1, "Ratio_Cancelados": 1}).sort("Ratio_Cancelados", -1)
list(cancelMmedia)
Out[10]: [{'_id': 2007.0, 'Ratio_Cancelados': 0.022620969803159113}, {'_id': 2014.0, 'Ratio_Cancelados': 0.021415810226867758}, {'_id': 2008.0, 'Ratio_Cancelados': 0.020347525993877402}, {'_id': 2011.0, 'Ratio_Cancelados': 0.01974539258454947}, {'_id': 2005.0, 'Ratio_Cancelados': 0.01902082577840011}, {'_id': 2010.0, 'Ratio_Cancelados': 0.018519432838974892}]
```

Se aprecia cómo el **peor año en términos de cancelaciones se produjo en 2007 seguido de 2014**.

- **Query 6.2.**

2- Los años en los que las **Retrasos fueron mayores en términos relativos** a la media

```
In [11]: RetrasoMmedia = ratios.find({"Ratio_Retrasados": {"$gt": Medias_Ratio["Media_Ratio_Retrasados"]}}, {"_id": 1, 'Ratio_Retrasados': 1}).sort("Ratio_Retrasados", -1)
list(RetrasoMmedia)

Out[11]: [{{'_id': 2007.0, 'Ratio_Retrasados': 0.24885798970934814},
{'_id': 2006.0, 'Ratio_Retrasados': 0.230874962144701},
{'_id': 2008.0, 'Ratio_Retrasados': 0.22405168109100812},
{'_id': 2005.0, 'Ratio_Retrasados': 0.21163218111169899},
{'_id': 2014.0, 'Ratio_Retrasados': 0.20820566373245558},
{'_id': 2004.0, 'Ratio_Retrasados': 0.20269981943662846}]
```

Se aprecia cómo el **peor año en términos de retrasos se produjo tambien en 2007 seguido de 2006.**

- **Query 6.3.**

3- Los años en los que la **Puntualidad fue mejor en términos relativos** a la media

```
In [12]: PuntualMmedia = ratios.find({"Ratio_Puntuales": {"$gt": Medias_Ratio["Media_Ratio_Puntuales"]}}, {"_id": 1, 'Ratio_Puntuales': 1}).sort('Ratio_Puntuales', -1)
list(PuntualMmedia)

Out[12]: [{{'_id': 2012.0, 'Ratio_Puntuales': 0.8180302872865349},
{'_id': 2015.0, 'Ratio_Puntuales': 0.7990258064149397},
{'_id': 2010.0, 'Ratio_Puntuales': 0.7982493213639277},
{'_id': 2011.0, 'Ratio_Puntuales': 0.7954803296718702},
{'_id': 2009.0, 'Ratio_Puntuales': 0.788161875758251},
{'_id': 2013.0, 'Ratio_Puntuales': 0.7866391685126827},
{'_id': 2004.0, 'Ratio_Puntuales': 0.7773516681684843}]
```

Se aprecia cómo el **mejor año en términos de puntualidad se produjo en 2012 seguido de 2015.**

```
In [ ]:
```

## 5. Valores medios mensuales en Vuelos Totales, Retrasados y Cancelados.

### 5.1 Importación de librerías necesarias

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")

pd.options.display.float_format = '{:,.2f}'.format
```

### 5.2 Conexión con MONGO ATLAS / LOCAL

```
In [2]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.||
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines

#local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

```
In [3]: air = db.airlines
```

### 5.3 Valores medios mensuales

- **Query 7**

Con este consulta se pretende obtener la distribución media mensual de **Vuelos Totales**, **Vuelos Retrasados** y **Vuelos Cancelados** a lo largo del período de estudio. Los pasos a seguir son los siguientes.

1. Se seleccionan los años que sean distintos a 2003 y 2016.
2. Se agrupa por mes y se calcula la media de los Vuelos Totales, Vuelos Retrasados y Vuelos Cancelados durante el período de estudio.

```
In [4]: pipeline7 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                    {"$group": {"_id": "$time.month",
                               "Totales": {"$avg": "$statistics.flights.total"},
                               "Retrasados": {"$avg": "$statistics.flights.delayed"},
                               "Cancelados": {"$avg": "$statistics.flights.cancelled"}}
                    }
                ]
curs7 = air.aggregate(pipeline7)
```

Con el cursor obtenido de la consulta, se crea un **DataFrame** con los resultados de la misma.

```
In [5]: query7 = list(curs7)
```

```
In [6]: df7 = pd.DataFrame(query7).set_index(['_id']).sort_index()
df7.T
```

Out[6]:

_id	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0
<b>Cancelados</b>	27.89	29.49	18.27	12.94	12.54	17.65	17.36	15.82	12.69	12.50	9.30	23.40
<b>Retrasados</b>	200.64	188.14	205.56	179.87	190.06	238.31	238.58	208.50	146.03	174.56	156.42	239.66
<b>Totales</b>	947.53	883.00	1,004.80	967.90	985.82	984.26	1,030.23	1,018.28	934.88	985.68	930.91	958.14

```
In [7]: fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df7.index, y= df7.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("MEDIA CANCELACIONES");
ax1.set_xlabel("MESES");

sns.barplot(x=df7.index, y= df7.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("MEDIA RETRASOS");
ax2.set_xlabel("MESES");

sns.barplot(x=df7.index, y= df7.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("MEDIA TOTALES");
ax3.set_xlabel("MESES");
```



## 2.3. Gráficos Distribución mes-año de Ratios de Vuelos Puntuales, Retrasados y Cancelados respecto a Totales.

Finalmente, se crean los gráficos correspondientes a las **distribuciones mes-año de los Ratios de Vuelos Puntuales, Retrasados y Cancelados respecto a Totales**.

- Query 8

```
In [8]: pipeline8 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                    {"$group": {"_id": {"year": "$time.year", "month": "$time.month"},
                               "Totales": {"$sum": "$statistics.flights.total"},
                               "Retrasados": {"$sum": "$statistics.flights.delayed"},
                               "Cancelados": {"$sum": "$statistics.flights.cancelled"},
                               "Puntuales": {"$sum": "$statistics.flights.on time"}
                           }
                },
                {"$project": {
                    "year": "$_id.year",
                    "month": "$_id.month",
                    "Ratio_Retrados": {"$divide": ["$Retrasados", "$Totales"]},
                    "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]},
                    "Ratio_Puntuales": {"$divide": ["$Puntuales", "$Totales"]},
                    "_id": 0
                }
            }
        ]
curs8 = air.aggregate(pipeline8)
```

```
In [9]: query8 = list(curs8)
```

```
In [10]: df8 = pd.DataFrame(query8).set_index(["year", "month"]).unstack()
df8.index.name = "year"
df8.head(10)
```

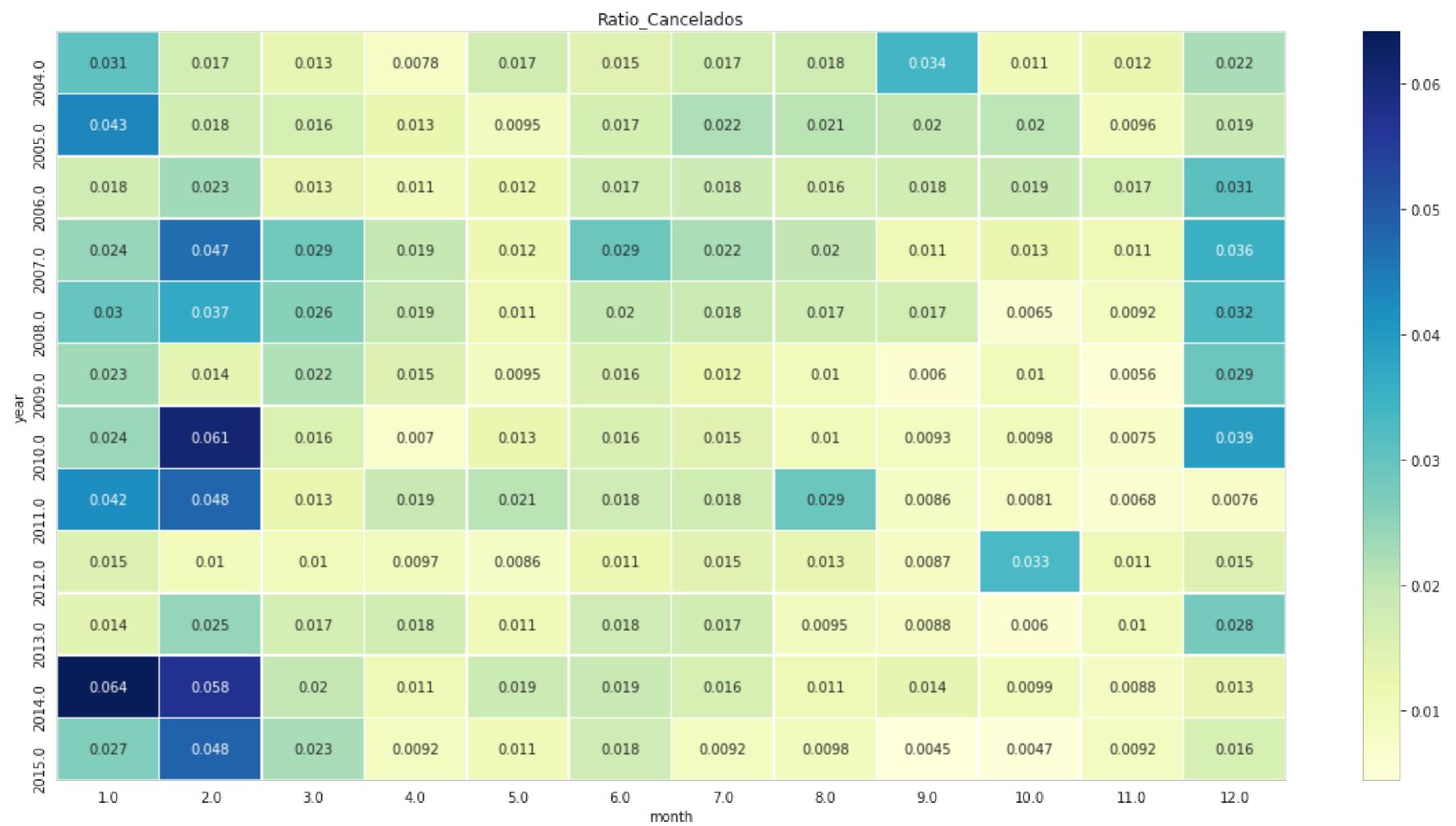
Out[10]:

month	Ratio_Cancelados												Ratio_Retrados											
	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	...	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00	1.00	2.00	3.00
year	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	...	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00	1.00	2.00	3.00
2,004.00	0.03	0.02	0.01	0.01	0.02	0.01	0.02	0.02	0.03	0.01	...	0.18	0.16	0.21	0.25	0.22	0.20	0.13	0.18	0.20	0.26	0.03	0.02	0.01
2,005.00	0.04	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.02	0.02	...	0.22	0.15	0.16	0.24	0.27	0.23	0.16	0.18	0.19	0.27	0.04	0.02	0.01
2,006.00	0.02	0.02	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	...	0.23	0.21	0.21	0.25	0.24	0.22	0.23	0.26	0.22	0.26	0.02	0.02	0.01
2,007.00	0.02	0.05	0.03	0.02	0.01	0.03	0.02	0.02	0.01	0.01	...	0.25	0.23	0.22	0.29	0.28	0.27	0.17	0.21	0.20	0.33	0.02	0.05	0.03
2,008.00	0.03	0.04	0.03	0.02	0.01	0.02	0.02	0.02	0.02	0.01	...	0.26	0.21	0.21	0.28	0.23	0.20	0.14	0.14	0.16	0.31	0.03	0.04	0.03
2,009.00	0.02	0.01	0.02	0.02	0.01	0.02	0.01	0.01	0.01	0.01	...	0.20	0.20	0.19	0.23	0.21	0.20	0.13	0.22	0.11	0.25	0.02	0.01	0.02
2,010.00	0.02	0.06	0.02	0.01	0.01	0.02	0.01	0.01	0.01	0.01	...	0.19	0.14	0.19	0.22	0.21	0.17	0.14	0.15	0.16	0.23	0.02	0.06	0.02
2,011.00	0.04	0.05	0.01	0.02	0.02	0.02	0.03	0.01	0.01	0.01	...	0.19	0.22	0.21	0.21	0.20	0.18	0.15	0.14	0.14	0.15	0.04	0.05	0.01
2,012.00	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.03	...	0.17	0.12	0.16	0.18	0.22	0.19	0.16	0.17	0.13	0.21	0.02	0.01	0.01
2,013.00	0.01	0.02	0.02	0.02	0.01	0.02	0.02	0.01	0.01	0.01	...	0.18	0.21	0.19	0.26	0.25	0.20	0.15	0.15	0.15	0.27	0.01	0.02	0.02

10 rows × 36 columns

```
In [11]: fig1, ax1 = plt.subplots(1, 1, figsize=(20, 10));

ax1 = sns.heatmap(df8["Ratio_Cancelados"], annot=True, linewidths=.5, cmap="YlGnBu");
ax1.set_title("Ratio_Cancelados");
```

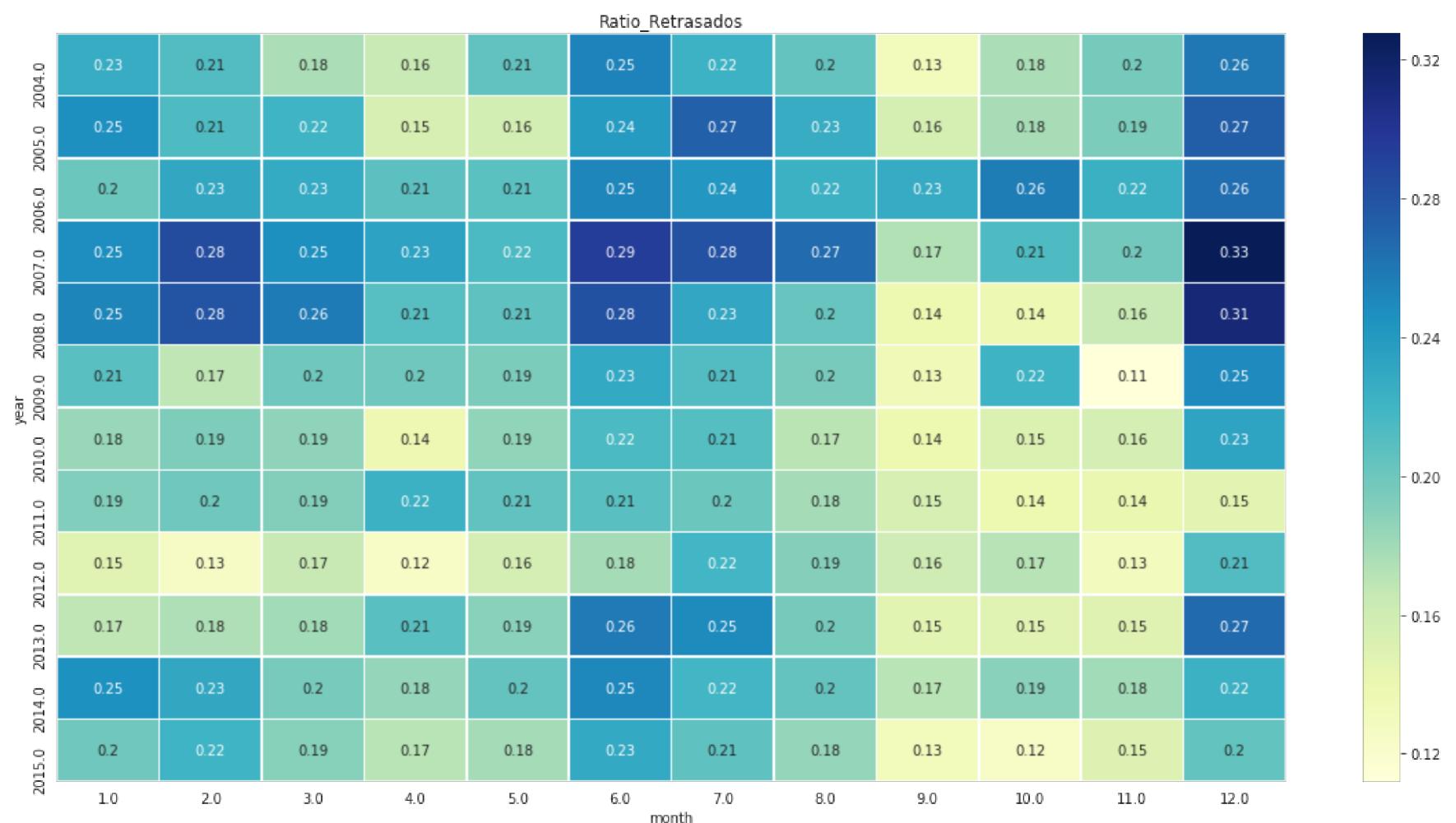


De este gráfico se desprende que:

- los meses donde peores ratios de cancelaciones se han registrado históricamente han sido enero, febrero y diciembre.

```
In [12]: fig2, ax2 = plt.subplots(1, 1, figsize=(20, 10));

ax2 = sns.heatmap(df8["Ratio_Retrasados"], annot=True, linewidths=.5, cmap="YlGnBu");
ax2.set_title("Ratio_Retrasados");
```



De este gráfico se desprende que:

- los meses donde mejores ratios de retraso se han registrado históricamente han sido septiembre, octubre y noviembre.
- los meses donde peores ratios de retraso se han registrado históricamente han sido junio, julio y agosto.

```
In [13]: df_Puntuales = df8.unstack()["Ratio_Puntuales"]

fig3, ax3 = plt.subplots(1, 1, figsize=(20, 10));

ax3 = sns.heatmap(df8["Ratio_Puntuales"], annot=True, linewidths=.5, cmap="YlGnBu");
ax3.set_title("Ratio_Puntuales");
```



De este último gráfico se desprende que:

- los meses donde mejores ratios de puntualidad se registran históricamente han sido septiembre, octubre y noviembre.

```
In [ ]:
```

## 6. Estudio sobre aeropuertos.

En este capítulo se analizan los aeropuertos según los siguientes indicadores:

- volumen de los vuelos totales,
- volumen de vuelos retrasado
- volumen de vuelos cancelados

También se analizan las causas principales y más comunes dentro de los vuelos retrasados. Dichas causas son las que aquí se incluyen:

- causas debidas al avión
- causas debidas al tiempo
- causas debidas a la seguridad
- causas debidas a sistemas de navegación
- causas debidas al transportista

Por último, se analiza un parámetro que indica el **número de minutos de retraso acumulados** que además de los vuelos retrasados, da una idea de la importancia del tiempo de retraso en el aeropuerto.

### 6.1 Importación de librerías necesarias

Inicialmente se cargan las librerías de python necesarias.

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")

pd.options.display.float_format = '{:,.2f}'.format
```

### 6.2 Conexión con MONGO ATLAS / LOCAL

```
In [2]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.|
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb\|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines

#local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

Se crea la variable **air** para facilitar las consultas.

```
In [3]: air = db.airlines
```

### 6.3 Evolución histórica de Ratios Vuelos Puntuales, Retrasados y Cancelados respecto a Vuelos Totales por aeropuerto.

En este apartado se va a extraer la evolución histórica, dentro del período de estudio, de los **ratios de vuelos puntuales, retrasados y cancelados respecto a los vuelos totales por aeropuerto**.

Para ello, se realiza la siguiente consulta en la que:

- se filtran los años de interés
- se agrupa por aeropuerto, año y mes
- en dicha agrupación, se calcula la suma de:
  - vuelos retrasados
  - vuelos cancelados
  - vuelos puntuales
- se crean los campos:
  - ratios vuelos retrasados
  - vuelos vuelos cancelados
  - vuelos vuelos puntuales

Además, de la consulta, se va a **generar una colección "Histórico Aeropuertos"** para facilitar las consultas posteriores..

- **Query 10**

```
In [4]: pipeline9 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},  
                    {"$group": {"_id":  
                                {"Aeropuerto": "$airport.code", "year": "$time.year", "month": "$time.month"},  
                                "Totales": {"$sum": "$statistics.flights.total"},  
                                "Retrasados": {"$sum": "$statistics.flights.delayed"},  
                                "Cancelados": {"$sum": "$statistics.flights.cancelled"},  
                                "Puntuales": {"$sum": "$statistics.flights.on time"}  
                            }  
                    },  
                    {"$project": {  
                                "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]},  
                                "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]},  
                                "Ratio_Puntuales": {"$divide": ["$Puntuales", "$Totales"]}  
                            }  
                    },  
                    {"$out": "Historico_Aeropuertos"}  
                ]  
  
air.aggregate(pipeline9);
```

Se crea la variable **Ads** para facilitar las consultas.

```
In [5]: #Se almacena la colección en una variable  
Ads = db.Historico_Aeropuertos
```

Una vez obtenida la nueva colección, se procede a utilizarla.

## 6.4 Aeropuertos con peores Ratios de Vuelos Puntuales, Retrasados y Cancelados respecto a Vuelos Totales por aeropuerto.

Partiendo de la colección creada, se plantea la pregunta de cuáles son los **aeropuertos que peores Ratios medios han tenido**, durante los años de estudio.

Para esta pregunta se plantea la siguiente consulta.

- Se agupa por aeropuerto
- Se calculan los **ratios medios**

- **Query 11**

```
In [6]: pipeline11 = [{"$group": {"_id": "$_id.Aeropuerto",  
                                "Ratio_Retrasados": {"$avg": "$Ratio_Retrasados"},  
                                "Ratio_Cancelados": {"$avg": "$Ratio_Cancelados"},  
                                "Ratio_Puntuales": {"$avg": "$Ratio_Puntuales"}  
                            }  
                    }  
  
curs11 = db.Historico_Aeropuertos.aggregate(pipeline11)
```

El cursor obtenido se convierte en lista y se almacena en una lista y se convierte el resultado en un datafram. Se ofrece una vista de los 10 primeros aeropuertos.

```
In [7]: query11 = list(curs11)
```

```
In [8]: df11 = pd.DataFrame(query11).set_index("_id").sort_index()
df11.index.name = "Aeropuertos"
df11.head(10)
```

Out[8]:

	Ratio_Cancelados	Ratio_Puntuales	Ratio_Retrasados
--	------------------	-----------------	------------------

Aeropuertos

ATL	0.02	0.78	0.21
BOS	0.03	0.75	0.22
BWI	0.01	0.80	0.18
CLT	0.02	0.80	0.18
DCA	0.03	0.78	0.19
DEN	0.01	0.80	0.18
DFW	0.02	0.79	0.18
DTW	0.02	0.80	0.18
EWR	0.03	0.68	0.29
FLL	0.01	0.77	0.22

Se representa gráficamente los resultados anteriores.

```
In [9]: fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df11.index, y= df11.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("CANCELADOS");
ax1.set_xlabel("AEROPUERTO");

sns.barplot(x=df11.index, y= df11.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("PUNTUALES");
ax2.set_xlabel("AEROPUERTO");

sns.barplot(x=df11.index, y= df11.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("RETRASADOS");
ax3.set_xlabel("AEROPUERTO");
```



En los gráficos anteriores se aprecian los siguientes resultados interesantes:

- Los aeropuertos con más **cancelaciones promedio** son **LGA, ORD y EWR**
- Los aeropuertos con más **retrasos promedio** son **EWG, LGA y SFO**
- Los aeropuertos con **peor puntualidad** son, como cabía esperar, **EWG, LGA y SFO\*\***

Por otro lado,

- El aeropuerto con **mejor puntualidad y menores ratios de retrasos** es **SLC**, que además posee un buen ratio en vuelos cancelados.

Para saber el **nombre de los aeropuerto anteriores**, la siguiente consulta nos da la correspondencia entre los códigos de los anteriores aeropuertos y sus respectivos nombres.

- **Query 12**

```
In [10]: curs12 = air.find(
    {"airport.code": {"$in": ["LGA", "ORD", "EWR", "SFO"]}},
    {"airport.code": 1, "airport.name": 1, "_id": 0}
).limit(4)
```

```
In [11]: query12 = list(curs12)
```

```
In [12]: query12
```

```
Out[12]: [ {'airport': {'code': 'EWR',  
'name': 'Newark, NJ: Newark Liberty International'}},  
{'airport': {'code': 'LGA', 'name': 'New York, NY: LaGuardia'}},  
{'airport': {'code': 'ORD',  
'name': "Chicago, IL: Chicago O'Hare International"}},  
{'airport': {'code': 'SFO',  
'name': 'San Francisco, CA: San Francisco International'}}]
```

## 6.5 Causas más comunes de retraso durante el período de estudio

Nos interesa saber cuáles de las **causas principales de retraso son las más comunes en promedio** durante el período de estudio para lo cual se plantea la siguiente consulta.

- se filtran los años de interés
- se eliminan los documentos que poseen vuelos retrasados nulos
- se crean los campos que tienen en cuenta el cociente entre los vuelos retrasados debidos a una causa y los vuelos totales retrasados:
  - Ratio\_late aircraft: retrasos asociado al aeronave
  - Ratio\_weather: retrasos asociado al tiempo
  - Ratio\_security: retrasos asociado a la seguridad
  - Ratio\_national aviation system: retrasos asociado al sistema de navegación aérea
  - Ratio\_carrier: retrasos asociado al transportista
- Se calculan los valores promedio

- **Query13**

```
In [13]: pipeline13 = [  
    {"$match": {"time.year": {"$nin": [2003, 2016]},  
               "statistics.flights.delayed": {"$ne": 0}}}, #se quitan valores nulos  
    {"$project": {"time.year": 1,  
                "Ratio_late aircraft": {"$divide":  
                                         ["$statistics.# of delays.late aircraft", "$statistics.flights.delayed"]},  
                "Ratio_weather": {"$divide":  
                                         ["$statistics.# of delays.weather", "$statistics.flights.delayed"]},  
                "Ratio_security": {"$divide":  
                                         ["$statistics.# of delays.security", "$statistics.flights.delayed"]},  
                "Ratio_national aviation system": {"$divide":  
                                         ["$statistics.# of delays.national aviation system", "$statistics.flights.delayed"]}  
    },  
    {"$group": {"_id": {"$time.year",  
                      "late": {"$avg": "$Ratio_late aircraft"},  
                      "weather": {"$avg": "$Ratio_weather"},  
                      "security": {"$avg": "$Ratio_security"},  
                      "system": {"$avg": "$Ratio_national aviation system"},  
                      "carrier": {"$avg": "$Ratio_carrier"}  
    }}  
]
```

curs13 = db.airlines.aggregate(pipeline13)

```
In [14]: query13 = list(curs13)
```

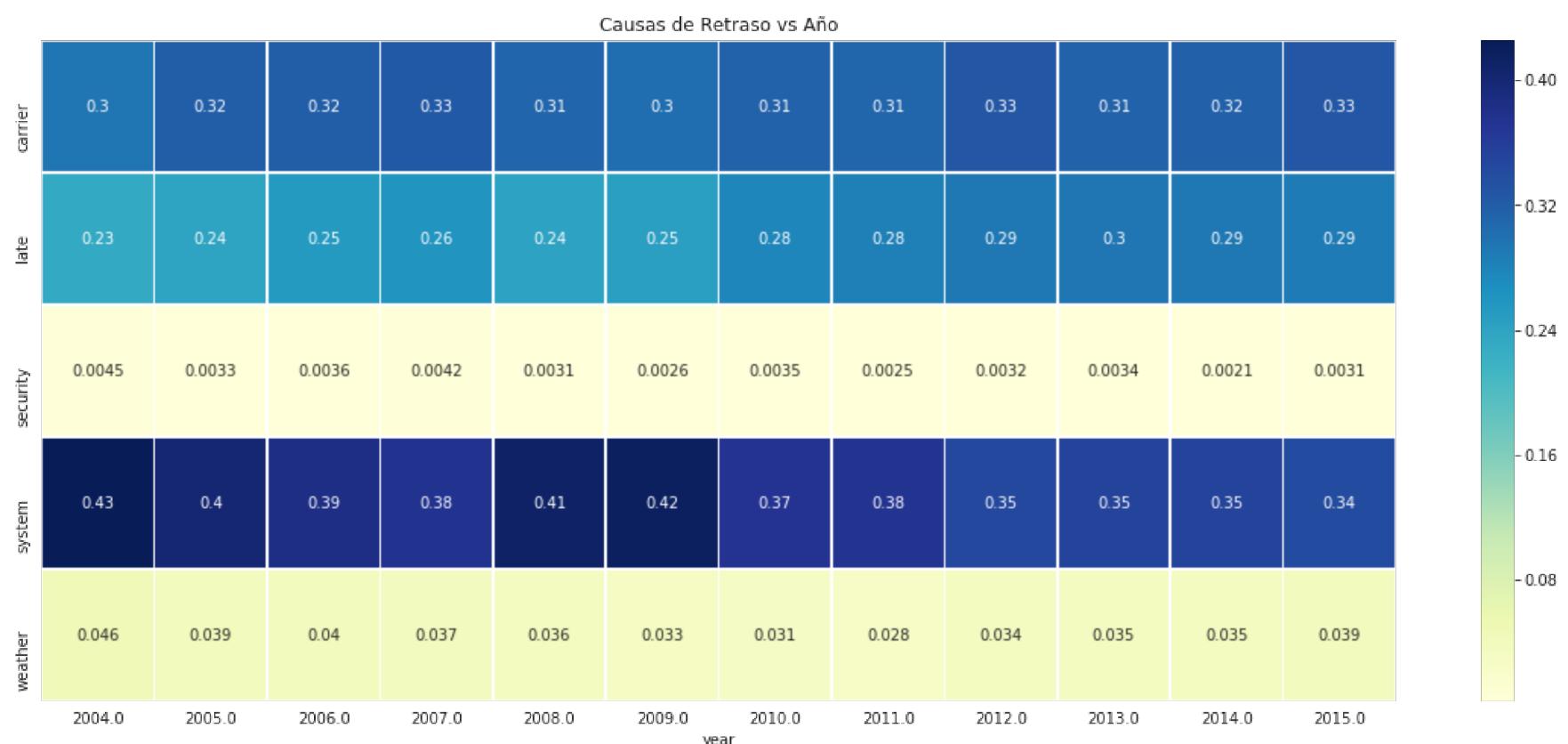
```
In [15]: df13 = pd.DataFrame(query13).set_index("_id").sort_index()
df13.index.name = "year"
df13.index.values
df13
```

Out[15]:

year	carrier	late	security	system	weather
2,004.00	0.30	0.23	0.00	0.43	0.05
2,005.00	0.32	0.24	0.00	0.40	0.04
2,006.00	0.32	0.25	0.00	0.39	0.04
2,007.00	0.33	0.26	0.00	0.38	0.04
2,008.00	0.31	0.24	0.00	0.41	0.04
2,009.00	0.30	0.25	0.00	0.42	0.03
2,010.00	0.31	0.28	0.00	0.37	0.03
2,011.00	0.31	0.28	0.00	0.38	0.03
2,012.00	0.33	0.29	0.00	0.35	0.03
2,013.00	0.31	0.30	0.00	0.35	0.03
2,014.00	0.32	0.29	0.00	0.35	0.04
2,015.00	0.33	0.29	0.00	0.34	0.04

Si se representa gráficamente los resultados anteriores,

```
In [16]: fig2, ax = plt.subplots(1, 1, figsize=(20,8));
ax = sns.heatmap(df13.T, annot=True, linewidths=.5, cmap="YlGnBu");
ax.set_title("Causas de Retraso vs Año");
```



Como se aprecia la **causa principal de retraso** es debida a los sistemas de navegación aerea en todos los años de estudio estando en el intervalo del 30% al 45%. Le siguen las **causas debidas al transportista** y las debidas a la aeronave en el intervalo del 20% al 35%.

Por el contrario, las **causas climatológicas y se seguridad** no llegan a representar ni un 5% ni un 1% respectivamente.

## 6.6 Causas más comunes de retraso durante el período de estudio para los aeropuertos con mayores retrasos

Anteriormente se obtuvo que **los aeropuertos con más retrasos promedio son EWG, LGA y SFO**. Si se quiere obtener **las principales causas** que, en promedio, han sido más frecuentes en el periodo de estudio, se ejecutaría la siguiente consulta:

- se filtran los años de interés
- se filtran los aeropuertos **EWG, LGA y SFO**
- se crean los campos que tienen en cuenta el cociente entre los vuelos retrasados debidos a una causa y los vuelos totales retrasados:
  - Ratio\_late aircraft: retrasos asociado al aeronave
  - Ratio\_weather: retrasos asociado al tiempo
  - Ratio\_security: retrasos asociado a la seguridad
  - Ratio\_national aviation system: retrasos asociado al sistema de navegación aérea
  - Ratio\_carrier: retrasos asociado al transportista
- Se calculan los valores promedio

• **Query 14**

```
In [17]: pipeline14 = [
    {"$match": {"time.year": {"$nin": [2003, 2016]}, 
               "statistics.flights.delayed": {"$ne": 0}, #se quitan valores nulos
               "airport.code": {"$in": ["LGA", "ORD", "EWR", "SFO"]}}},
    {"$project": {
        "airport.code": 1,
        "Ratio_late aircraft": {"$divide": [
            "$statistics.# of delays.late aircraft", "$statistics.flights.delayed"]},
        "Ratio_weather": {"$divide": [
            "$statistics.# of delays.weather", "$statistics.flights.delayed"]},
        "Ratio_security": {"$divide": [
            "$statistics.# of delays.security", "$statistics.flights.delayed"]},
        "Ratio_national aviation system": {"$divide": [
            "$statistics.# of delays.national aviation system",
            "$statistics.flights.delayed"]},
        "Ratio_carrier": {"$divide": [
            "$statistics.# of delays.carrier", "$statistics.flights.delayed"]]}},
    {"$group": {"_id": "airport.code",
               "late": {"$avg": "$Ratio_late aircraft"}, 
               "weather": {"$avg": "$Ratio_weather"}, 
               "security": {"$avg": "$Ratio_security"}, 
               "system": {"$avg": "$Ratio_national aviation system"}, 
               "carrier": {"$avg": "$Ratio_carrier"}}
    }
]
curs14 = db.airlines.aggregate(pipeline14)
```

```
In [18]: query14 = list(curs14)
```

```
In [19]: df14 = pd.DataFrame(query14).set_index("_id").sort_index()
df14.index.name = "Aeropuertos"
df14
```

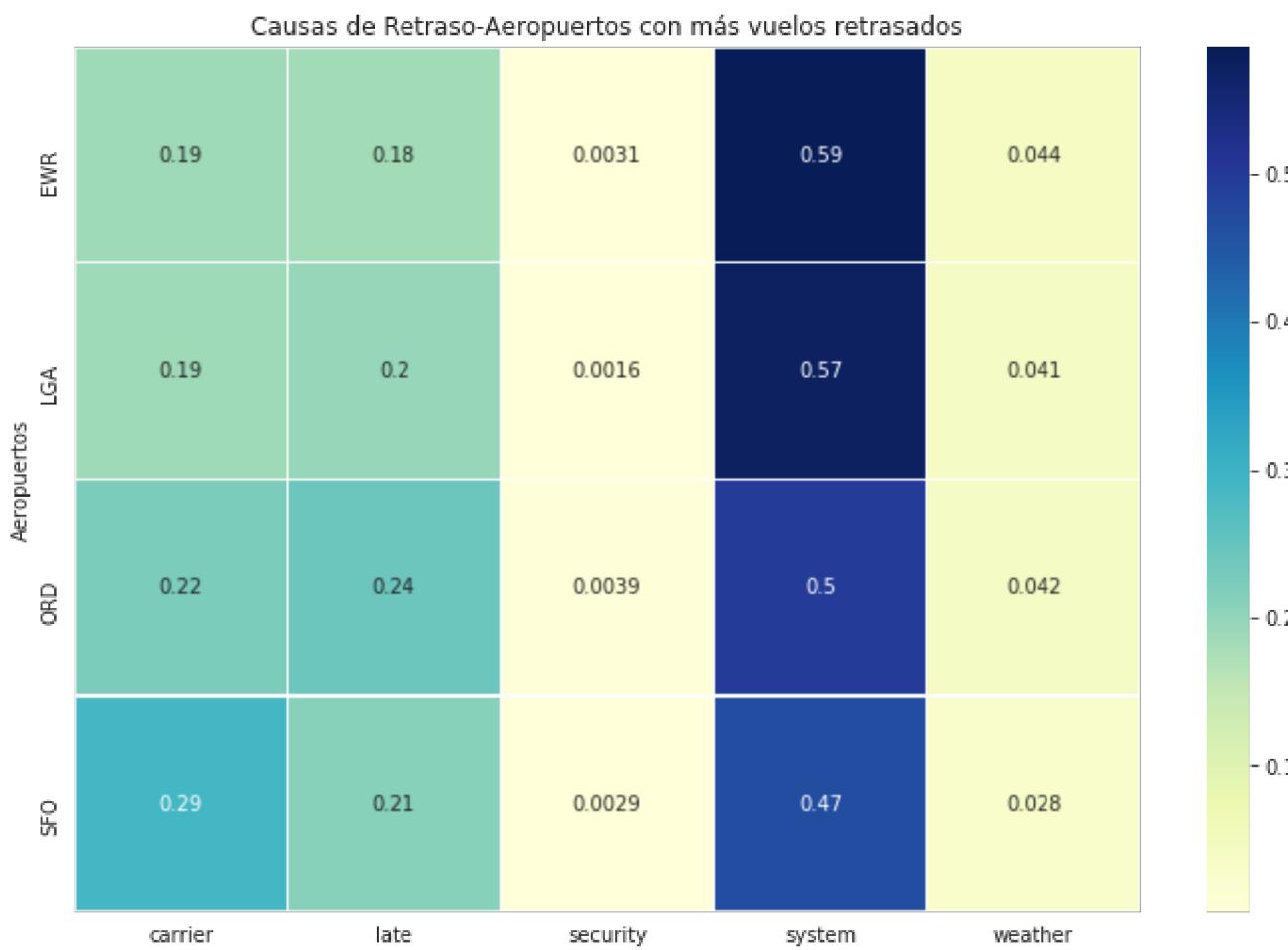
Out[19]:

	carrier	late	security	system	weather
<b>Aeropuertos</b>					
<b>EWR</b>	0.19	0.18	0.00	0.59	0.04
<b>LGA</b>	0.19	0.20	0.00	0.57	0.04
<b>ORD</b>	0.22	0.24	0.00	0.50	0.04
<b>SFO</b>	0.29	0.21	0.00	0.47	0.03

Si se representa gráficamente los resultados anteriores,

```
In [20]: fig3, ax = plt.subplots(1, 1, figsize=(12,8));

ax = sns.heatmap(df14, annot=True, linewidths=.5, cmap="YlGnBu");
ax.set_title("Causas de Retraso-Aeropuertos con más vuelos retrasados");
```



Se observa que:

- en general estos aeropuertos **siguen los resultados anteriores en cuanto a principales causas de retraso**
- más del **47% de los retrasos en estos aeropuertos son debidos a los sistemas de navegación aérea**.
- casi entre el **40% y 50% de los retrasos, dependiendo del aeropuerto, son debidas al transportista y al aeronave**

## 6.7 Minutos de retraso promedio por aeropuerto y causa del retraso

También es interesante estudiar la **variable minutos de retraso asociada a una determinada causa** que da una idea de la importancia de los tipos de retraso, en términos de tiempo, en el que los vuelos no operaron a la hora prevista por dichas causas.

Nos preguntamos **qué aeropuertos poseen, en promedio, mayores minutos de retraso para una causa determinada**. Para ello se diseña la siguiente consulta:

- se filtran los años de estudio
- se agrupa por código de aeropuerto
- se calculan los promedios de los minutos de retraso por causa de retraso

```
In [21]: pipeline15 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                      {"$group": {"_id": "$airport.code",
                                  "totales": {"$avg": "$statistics.minutes delayed.total"},
                                  "late": {"$avg": "$statistics.minutes delayed.late aircraft"},
                                  "weather": {"$avg": "$statistics.minutes delayed.weather"},
                                  "carrier": {"$avg": "$statistics.minutes delayed.carrier"},
                                  "security": {"$avg": "$statistics.minutes delayed.security"},
                                  "system": {"$avg": "$statistics.minutes delayed.national aviation system"}
                               },
                      {"$sort": {"totales": -1}}
                   ]
curs15 = air.aggregate(pipeline15);
```

```
In [22]: query15 = list(curs15)
```

```
In [23]: df15 = pd.DataFrame(query15).set_index("_id")
df15.index.name = "Aeropuertos"
df15.columns.name = "Minutos Retraso Promedio"
df15.head(10)
```

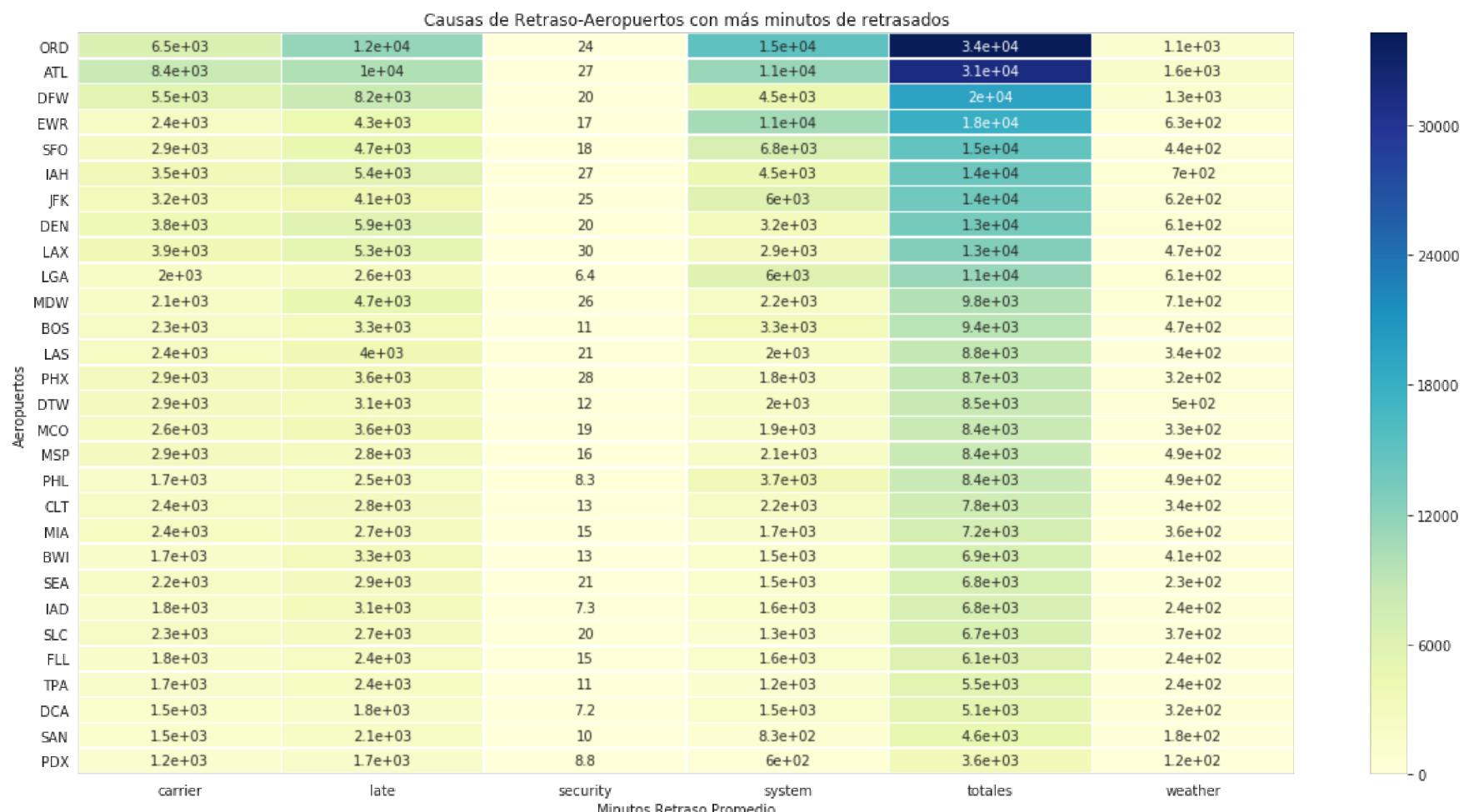
Out[23]:

	Minutos Retraso Promedio	carrier	late	security	system	totales	weather
<b>Aeropuertos</b>							
<b>ORD</b>	6,541.20	11,579.74	24.35	15,006.06	34,290.48	1,139.13	
<b>ATL</b>	8,397.95	10,036.67	26.62	11,382.94	31,473.75	1,629.57	
<b>DFW</b>	5,533.13	8,176.96	20.43	4,541.96	19,536.66	1,264.17	
<b>EWR</b>	2,387.20	4,331.69	16.67	10,674.62	18,037.61	627.43	
<b>SFO</b>	2,892.19	4,669.82	17.76	6,795.59	14,816.98	441.62	
<b>IAH</b>	3,525.54	5,370.08	27.40	4,462.51	14,080.77	695.24	
<b>JFK</b>	3,151.96	4,124.80	24.72	5,981.23	13,898.33	615.61	
<b>DEN</b>	3,801.46	5,852.22	19.59	3,159.31	13,441.95	609.36	
<b>LAX</b>	3,914.40	5,326.81	30.03	2,895.24	12,637.60	471.11	
<b>LGA</b>	2,049.01	2,649.61	6.37	5,963.29	11,276.82	608.54	

Representando lo anterior de forma gráfica

```
In [24]: fig4, ax = plt.subplots(1, 1, figsize=(20, 10))

ax = sns.heatmap(df15, annot=True, linewidths=.5, cmap="YlGnBu");
ax.set_title("Causas de Retraso-Aeropuertos con más minutos de retrasados");
```



Se aprecia que los **tres primeros aeropuertos (ORD, ATL, DFW) con más minutos promedio retrasados acumulados no son los tres primeros que más vuelos promedio retrasados tienen**. Para saber los nombres de estos tres:

#### • Query 16

```
In [25]: curs16 = air.find(
    {"airport.code": {"$in": ["ORD", "ATL", "DFW"]}},
    {"$airport.code": 1, "airport.name": 1, "_id": 0}
).limit(3)
```

```
In [26]: query16 = list(curs16)
```

Con esto, se obtienen los nombres de los aeropuertos

```
In [27]: query16
```

```
Out[27]: [ {'airport': {'code': 'ATL',
   'name': 'Atlanta, GA: Hartsfield-Jackson Atlanta International'}},
  {'airport': {'code': 'DFW',
   'name': 'Dallas/Fort Worth, TX: Dallas/Fort Worth International'}},
  {'airport': {'code': 'ORD',
   'name': "Chicago, IL: Chicago O'Hare International"}]
```

Nos podemos preguntar **si estos tres aeropuertos son los que mayor número de vuelos acumulados** han tenido durante el período de estudio.

- **Query 17**

```
In [28]: pipeline17 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                      {"$group": {"_id": "$airport.code",
                                  "Totales": {"$sum": "$statistics.flights.total"}},
                      },
                      {"$sort": {"Totales": -1}}
                    ]
curs17 = air.aggregate(pipeline17)
```

```
In [29]: query17 = list(curs17)
```

```
In [30]: df17 = pd.DataFrame(query17).set_index("_id")
df17.index.name = "Aeropuertos"
df17.head(10)
```

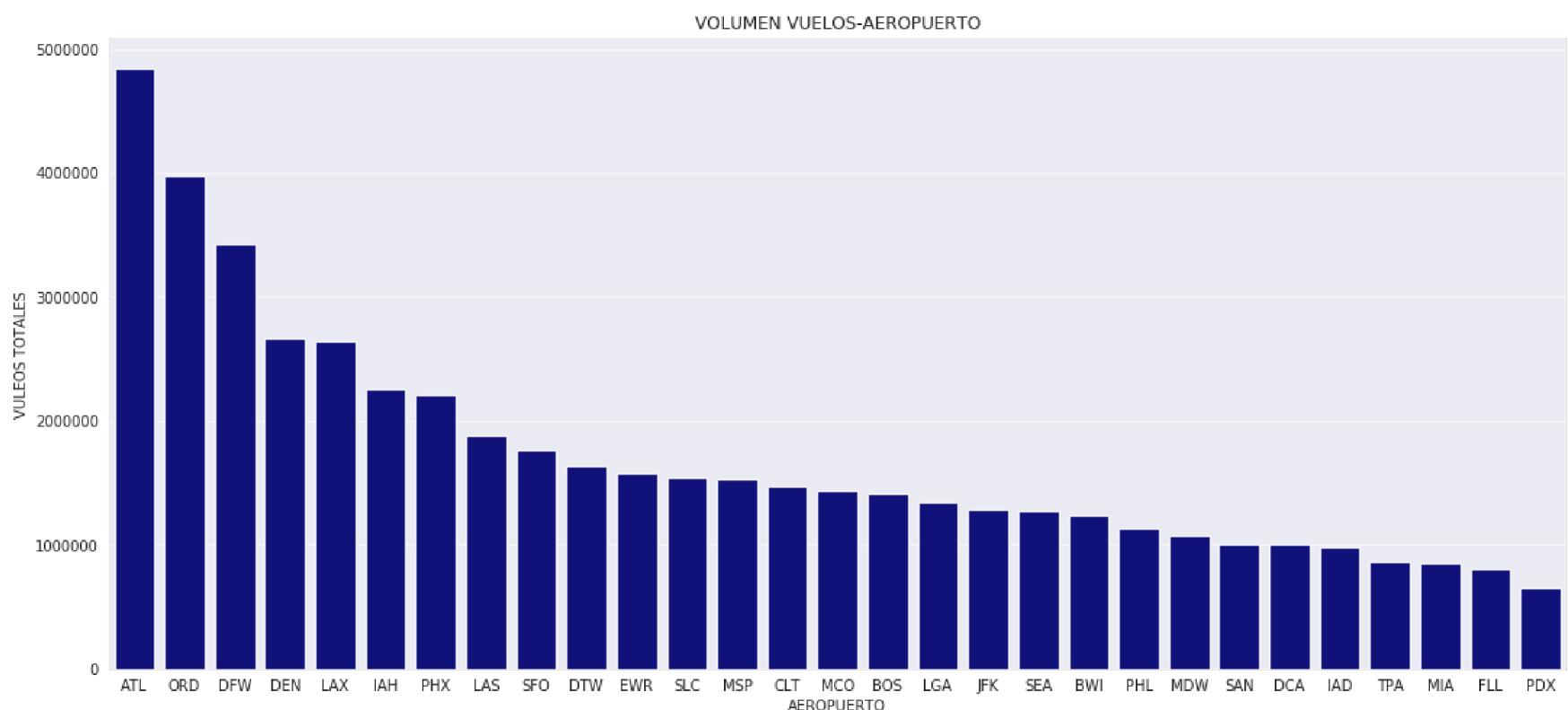
```
Out[30]:
```

Totales	
<u>Aeropuertos</u>	
ATL	4,847,036.00
ORD	3,973,794.00
DFW	3,422,939.00
DEN	2,669,176.00
LAX	2,637,252.00
IAH	2,257,833.00
PHX	2,204,351.00
LAS	1,876,789.00
SFO	1,761,055.00
DTW	1,633,595.00

Representando gráficamente los resultados, se confirma que los **aeropuertos con más minutos de retraso son los que mayor volumen de tráfico han tenido durante el período de estudio**.

```
In [31]: fig5, ax1 = plt.subplots(1, 1, figsize=(18,8));

sns.barplot(x=df17.index, y= df17.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("VUELOS TOTALES");
ax1.set_xlabel("AEROPUERTO");
ax1.set_title("VOLUMEN VUELOS-AEROPUERTO");
```



## 6.8 Evolución anual de Minutos de retraso para los aeropuertos de ORD, ATL y DFW.

En este apartado, se analiza la **evolución anual** en el período de estudio de los minutos de retraso acumulados en los aeropuertos de ORD, ATL y DFW. Se diseña la siguiente consulta:

- se filtra los años de estudio
- se filtran los aeropuertos de **ORD, ATL y DFW**
- se calculan los minutos de retraso

### • Query 18

```
In [32]: pipeline18 = [{"$match": {"time.year": {"$nin": [2003, 2016]}, "airport.code": {"$in": ["ORD", "ATL", "DFW"]}}, {"$group": {"_id": {"Aeropuerto": "$airport.code", "Year": "$time.year"}, "Totales": {"$sum": "$statistics.minutes delayed.total"}}, {"$project": {"Aeropuerto": "$_id.Aeropuerto", "Year": "$_id.Year", "Totales": "$Totales", "_id": 0}}, {"$sort": {"Year": 1}}]
curs18 = air.aggregate(pipeline18);
```

```
In [33]: query18 = list(curs18)
```

```
In [34]: df18 = pd.DataFrame(query18).set_index(["Year", "Aeropuerto"]).unstack()
df18
```

Out[34]:

Aeropuerto	Totales		
	ATL	DFW	ORD
Year			
<b>2,004.00</b>	5,501,464.00	3,238,746.00	6,464,029.00
<b>2,005.00</b>	6,462,164.00	2,800,812.00	4,615,773.00
<b>2,006.00</b>	5,967,595.00	3,175,007.00	7,321,451.00
<b>2,007.00</b>	5,656,546.00	4,355,835.00	7,634,716.00
<b>2,008.00</b>	5,568,010.00	3,268,098.00	6,947,544.00
<b>2,009.00</b>	6,119,887.00	2,924,451.00	3,633,113.00
<b>2,010.00</b>	4,984,857.00	2,421,741.00	3,845,719.00
<b>2,011.00</b>	4,123,957.00	2,272,412.00	4,373,129.00
<b>2,012.00</b>	3,339,484.00	2,626,513.00	3,762,520.00
<b>2,013.00</b>	4,366,598.00	3,398,736.00	4,907,425.00
<b>2,014.00</b>	3,782,402.00	3,887,857.00	5,034,412.00
<b>2,015.00</b>	3,612,426.00	3,257,391.00	4,280,327.00

Representando la tabla anterior

```
In [35]: fig6, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df18.index, y= df18.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("Minutos retraso");
ax1.set_xlabel("Year");
ax1.set_title("ATL");

sns.barplot(x=df18.index, y= df18.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("Minutos retraso");
ax2.set_xlabel("Year");
ax2.set_title("DFW");

sns.barplot(x=df18.index, y= df18.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("Minutos retraso");
ax3.set_xlabel("Year");
ax3.set_title("ORL");
```



Se aprecia que:

- En el **aeropuerto de ATL** parece que los **minutos de retraso han descendido**
- En los **aeropuerto de DFW y ORL** parece que los **minutos de retraso han crecido en los años 2013, 2014 y 2015 después de un período de descenso entre los años 2009, 2010, 2011 y 2012**. Este efecto es más pronunciado en DFW.

## 6.9 Evolución mensual entre 2014-2015 de los Minutos de retraso, Ratio de vuelos cancelados, Ratio de vuelos retrasados y vuelos totales para los aeropuertos de ORD, ATL y DFW.

En este apartado, se analiza la **evolución mensual de dichos indicadores durante los años 2014-2015 en los aeropuertos de ORD, ATL y DFW**. Se diseña la siguiente consulta:

- se filtra los años de estudio
- se filtran los aeropuertos de **ORD, ATL y DFW**
- se calculan los indicadores

• **Query 19**

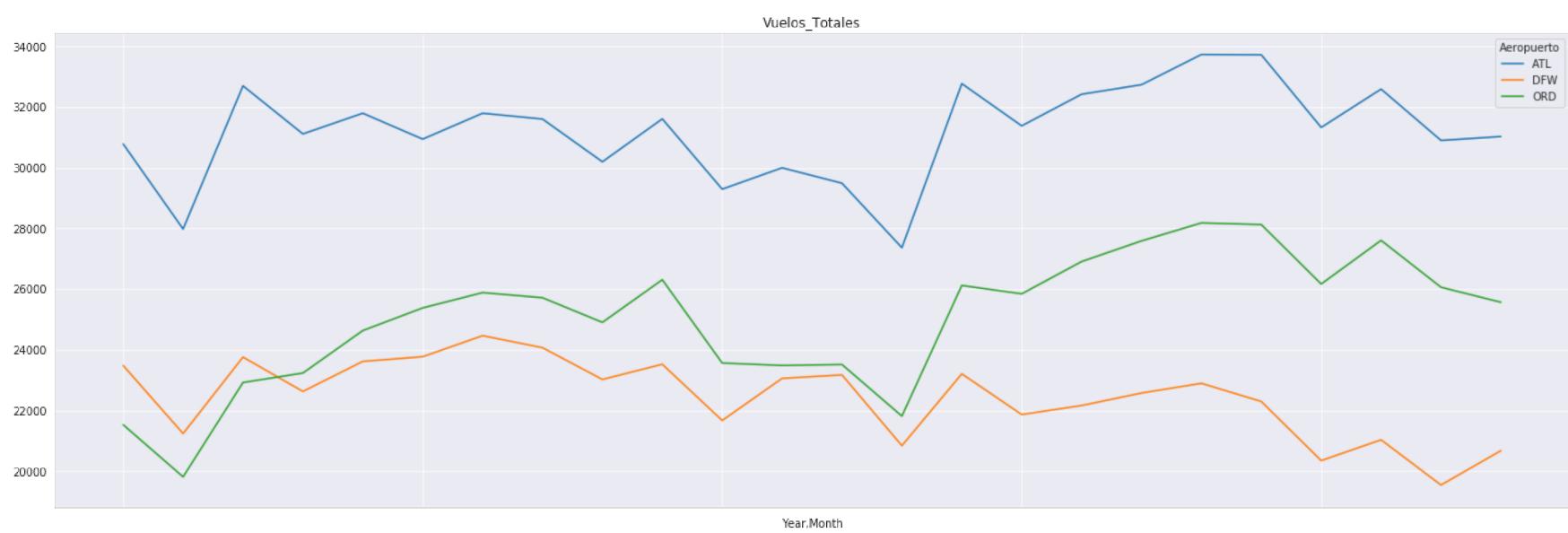
```
In [36]: pipeline19 = [{"$match": {"time.year": {"$in": [2014, 2015]}, "airport.code": {"$in": ["ORD", "ATL", "DFW"]}}, {"$group": {"_id": {"Aeropuerto": "$airport.code", "Year": "$time.year", "Month": "$time.month"}, "minTotales": {"$sum": "$statistics.minutes delayed.total"}, "Totales": {"$sum": "$statistics.flights.total"}, "Cancelados": {"$sum": "$statistics.flights.cancelled"}, "Retrasados": {"$sum": "$statistics.flights.delayed"}}, {"$project": {"Aeropuerto": "$_id.Aeropuerto", "Year": "$_id.Year", "Month": "$_id.Month", "minTotales": "$minTotales", "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]}, "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]}, "Vuelos_Totales": "$Totales", "_id": 0}}, {"$sort": {"Year": 1, "Month": 1}}], curs19 = air.aggregate(pipeline19);
```

```
In [37]: query19 = list(curs19)
```

```
In [38]: df19 = pd.DataFrame(query19).set_index(["Year", "Month", "Aeropuerto"]).unstack()
```

Se crea el gráfico que nos da la **evolución de Vuelos Totales** en los tres aeropuertos seleccionados

```
In [39]: df19["Vuelos_Totales"].plot(kind= "line", figsize=(25,8), title = "Vuelos_Totales");
```

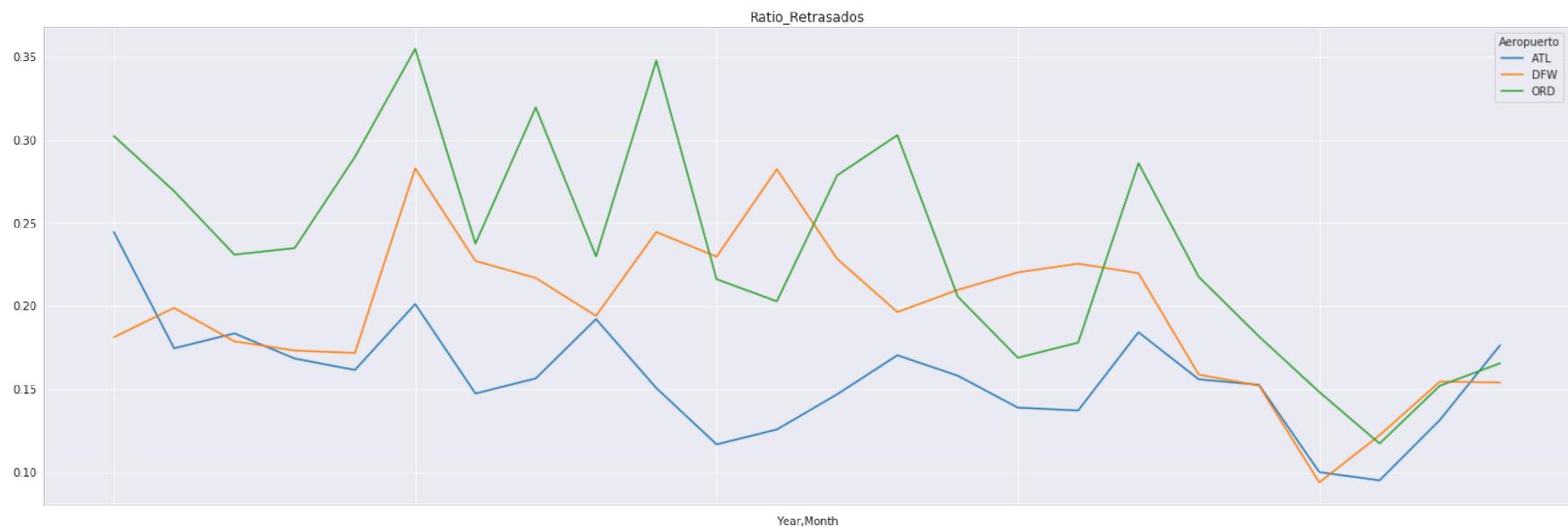


Se aprecia que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015:

- En ATL crece ligeramente el número de vuelos
- En ORD crece el número de vuelos
- En DFW, al contrario que los dos anteriores, decrece el número de vuelos

Para los **Ratios de los Vuelos Retrasados**

```
In [40]: df19["Ratio_Retrasados"].plot(kind= "line", figsize=(25,8), title = "Ratio_Retrasados");
```

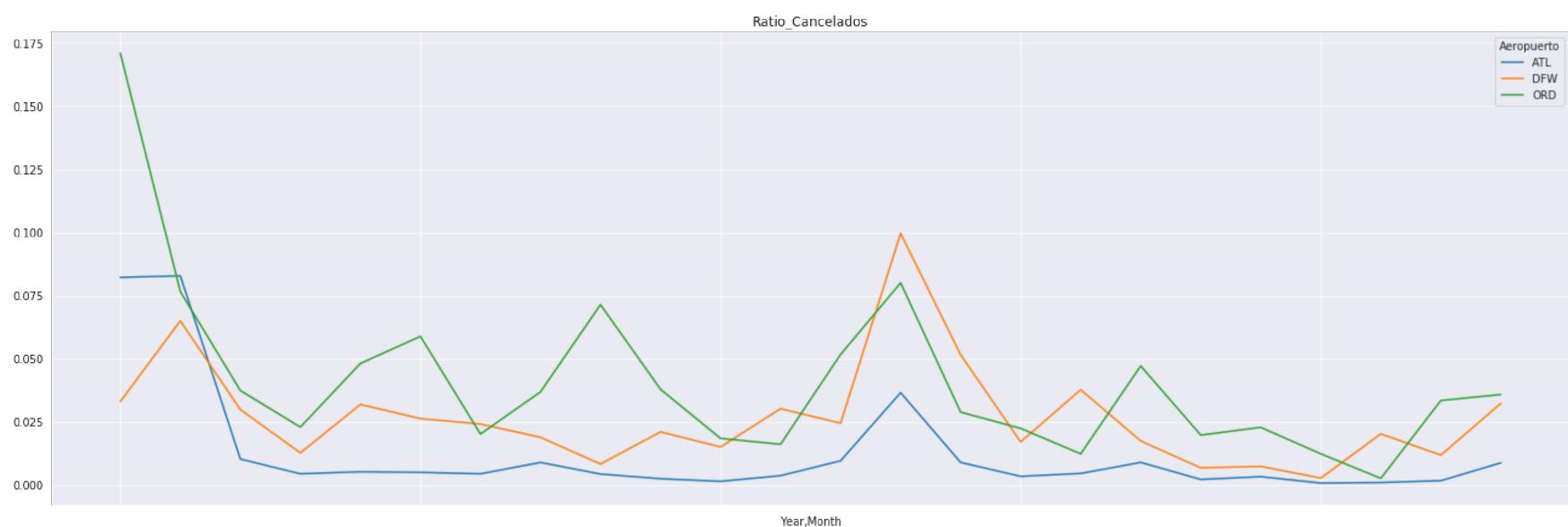


Parece que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015::

- En los **tres aeropuertos el Ratio de retrasados disminuye.**

Para los **Ratios de los Vuelos Retrasados**

```
In [41]: df19["Ratio_Cancelados"].plot(kind= "line", figsize=(25,8), title = "Ratio_Cancelados");
```

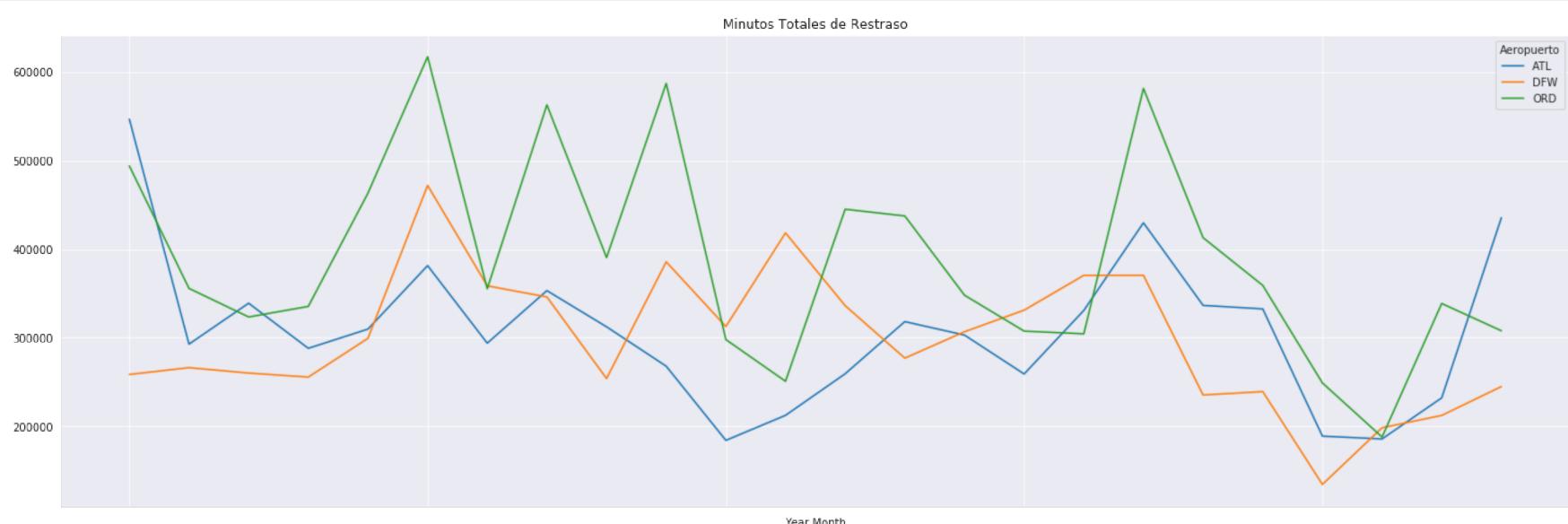


Parece que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015::

- En los **tres aeropuertos el Ratio de Cancelaciones permanece estable.**

Para los **Minutos de retraso totales**

```
In [42]: df19["minTotales"].plot(kind= "line", figsize=(25,8), title = "Minutos Totales de Retraso");
```



Parece que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015::

- En los **tres aeropuertos los minutos de retraso totales permanece ligeramente.**

In [ ]:

## 7. Estudio sobre aerolíneas.

En este capítulo se analizan las aerolíneas según los siguientes indicadores:

- volumen de los vuelos totales,
- volumen de vuelos retrasado
- volumen de vuelos cancelados

También se analizan las causas principales y más comunes dentro de los vuelos retrasados. Dichas causas son las que aquí se incluyen:

- causas debidas al avión
- causas debidas al tiempo
- causas debidas a la seguridad
- causas debidas a sistemas de navegación
- causas debidas al transportista

Por último, se analiza un parámetro que indica el **número de minutos de retraso acumulados** que además de los vuelos retrasados, da una idea de la importancia del tiempo de retraso de la aerolínea.

### 7.1 Importación de librerías necesarias

Inicialmente se cargan las librerías de python necesarias.

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")

pd.options.display.float_format = '{:,.2f}'.format
```

### 7.2 Conexión con MONGO ATLAS / LOCAL

```
In [2]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.|
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb\|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines

#local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

Se crea la variable **air** para facilitar las consultas.

```
In [3]: air = db.airlines
```

### 7.3 Evolución histórica de Ratios Vuelos Puntuales, Retrasados y Cancelados respecto a Vuelos Totales por aerolínea.

En este apartado se va a extraer la evolución histórica, dentro del período de estudio, de los **ratios de vuelos puntuales, retrasados y cancelados respecto a los vuelos totales por aerolínea**.

Para ello, se realiza la siguiente consulta en la que:

- se filtran los años de interés
- se agrupa por aerolínea, año y mes
- en dicha agrupación, se calcula la suma de:
  - vuelos retrasados
  - vuelos cancelados
  - vuelos puntuales
- se crean los campos:
  - ratios vuelos retrasados
  - vuelos vuelos cancelados
  - vuelos vuelos puntuales

Además, de la consulta, se va a **generar una colección "Histórico Aerolíneas"** para facilitar las consultas posteriores..

- **Query 20**

```
In [4]: pipeline20 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},  
                    {"$group": {"_id": {"Aerolinea": "$carrier.code", "year": "$time.year", "month": "$time.month"},  
                               "Totales": {"$sum": "$statistics.flights.total"},  
                               "Retrasados": {"$sum": "$statistics.flights.delayed"},  
                               "Cancelados": {"$sum": "$statistics.flights.cancelled"},  
                               "Puntuales": {"$sum": "$statistics.flights.on time"}  
                           }  
                  },  
                    {"$project": {  
                               "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]},  
                               "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]},  
                               "Ratio_Puntuales": {"$divide": ["$Puntuales", "$Totales"]}  
                           }  
                  },  
                    {"$out": "Historico_Aerolineas"}  
                ]  
  
air.aggregate(pipeline20);
```

Se crea la variable **Aer** para facilitar las consultas.

```
In [5]: #Se almacena la colección en una variable  
Aer = db.Historico_Aerolineas
```

Una vez obtenida la nueva colección, se procede a utilizarla.

## 7.4 Aerolíneas con peores Ratios de Vuelos Puntuales, Retrasados y Cancelados respecto a Vuelos Totales.

Partiendo de la colección creada, se plantea la pregunta de cuáles son las **aerolíneas que peores Ratios medios han tenido**, durante los años de estudio.

Para esta pregunta se plantea la siguiente consulta.

- Se agupa por aerolínea
- Se calculan los **ratios medios**

- **Query 21**

```
In [6]: pipeline21 = [{"$group": {"_id": {"_id.Aerolinea":  
                               "Ratio_Retrasados": {"$avg": "$Ratio_Retrasados"},  
                               "Ratio_Cancelados": {"$avg": "$Ratio_Cancelados"},  
                               "Ratio_Puntuales": {"$avg": "$Ratio_Puntuales"}  
                           }  
                  }  
                ]  
  
curs21 = Aer.aggregate(pipeline21)
```

El cursor obtenido se convierte en lista y se almacena en una lista y se convierte el resultado en un datafram. Se ofrece una vista de los 10 primeros aeropuertos.

```
In [7]: query21 = list(curs21)
```

```
In [8]: df21 = pd.DataFrame(query21).set_index("_id").sort_index()
df21.index.name = "Aerolíneas"
df21.head(10)
```

Out[8]:

	Ratio_Cancelados	Ratio_Puntuales	Ratio_Retrasados
--	------------------	-----------------	------------------

Aerolíneas

Aerolíneas	Ratio_Cancelados	Ratio_Puntuales	Ratio_Retrasados
9E	0.03	0.79	0.18
AA	0.02	0.76	0.22
AQ	0.00	0.81	0.19
AS	0.01	0.82	0.17
B6	0.01	0.75	0.24
CO	0.01	0.76	0.22
DH	0.03	0.76	0.21
DL	0.01	0.80	0.19
EV	0.03	0.73	0.24
F9	0.01	0.78	0.21

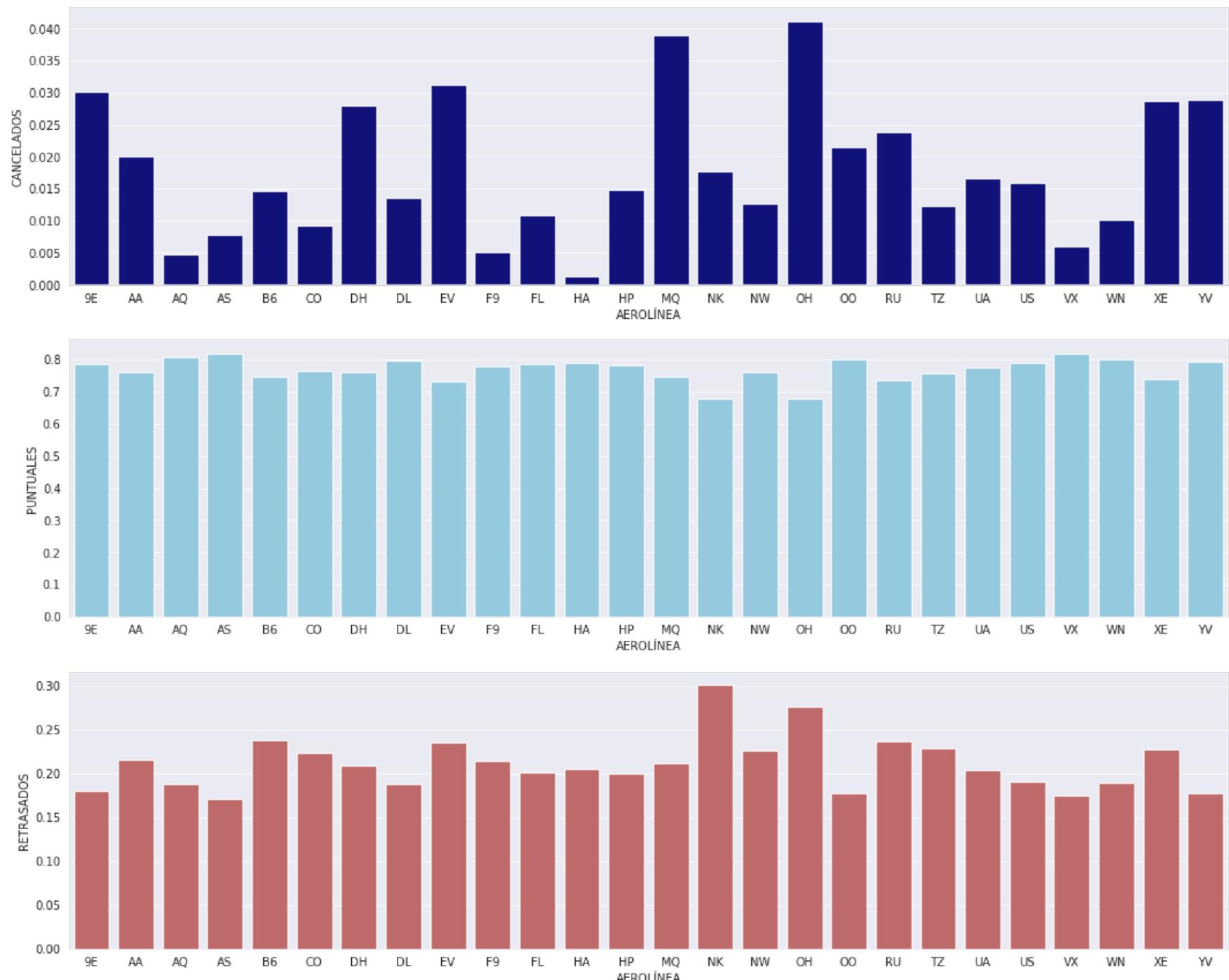
Se representa gráficamente los resultados anteriores.

```
In [9]: fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df21.index, y= df21.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("CANCELADOS");
ax1.set_xlabel("AEROLÍNEA");

sns.barplot(x=df21.index, y= df21.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("PUNTUALES");
ax2.set_xlabel("AEROLÍNEA");

sns.barplot(x=df21.index, y= df21.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("RETRASADOS");
ax3.set_xlabel("AEROLÍNEA");
```



En los gráficos anteriores se aprecian los siguientes resultados interesantes:

- Los aerolíneas con más **cancelaciones promedio** son OH, MQ y EV
- Los aerolíneas con más **retrasos promedio** son NK, OH, B6, EV y RU
- Los aerolíneas con **peor puntualidad** son, como cabía esperar, NK y OH

Por otro lado,

- Las aerolíneas con **mejor puntualidad** son AS, AQ y VX.

Para saber el **nombre de las aerolíneas anteriores**, la siguiente consulta nos da la correspondencia entre los códigos de las anteriores aerolíneas y sus respectivos nombres.

#### • Query 22

```
In [10]: curs22 = air.find(
    {"carrier.code": {"$in": ["OH", "MQ", "EV", "NK", "B6", "RU", "AS", "AQ", "VX"]}},
    {"carrier.code": 1, "carrier.name": 1, "_id": 0}
)
```

```
In [11]: query22 = list(curs22)
```

```
In [12]: query22 = [(doc["carrier"]['code'], doc["carrier"]['name']) for doc in query22]
query22 = set(query22)
```

```
In [13]: query22
```

```
Out[13]: {('AQ', 'Aloha Airlines Inc.'),
 ('AS', 'Alaska Airlines Inc.'),
 ('B6', 'JetBlue Airways'),
 ('EV', 'Atlantic Southeast Airlines'),
 ('EV', 'ExpressJet Airlines Inc.'),
 ('MQ', 'American Eagle Airlines Inc.'),
 ('MQ', 'Envoy Air'),
 ('NK', 'Spirit Air Lines'),
 ('OH', 'Comair Inc.'),
 ('RU', 'ExpressJet Airlines Inc.'),
 ('VX', 'Virgin America')}
```

## 7.5 Causas más comunes de retraso durante el período de estudio para las aerolíneas con mayores retrasos

Anteriormente se obtuvo que **las aerolíneas con más retrasos promedio son NK, OH, B6, EV y RU**. Si se quiere obtener **las principales causas** que, en promedio, han sido más frecuentes en el periodo de estudio, se ejecutaría la siguiente consulta:

- se filtran los años de interés
  - se filtran las aerolíneas **NK, OH, B6, EV y RU**
  - se crean los campos que tienen en cuenta el cociente entre los vuelos retrasados debidos a una causa y los vuelos totales retrasados:
    - Ratio\_late aircraft: retrasos asociado al aeronave
    - Ratio\_weather: retrasos asociado al tiempo
    - Ratio\_security: retrasos asociado a la seguridad
    - Ratio\_national aviation system: retrasos asociado al sistema de navegación aérea
    - Ratio\_carrier: retrasos asociado al transportista
  - Se calculan los valores promedio
- 
- **Query 23**

```
In [14]: pipeline23 = [
    {"$match": {"time.year": {"$nin": [2003, 2016]}, 
                "statistics.flights.delayed": {"$ne": 0}, #se quitan valores nulos
                "carrier.code": {"$in": ["NK", "OH", "B6", "EV", "RU"]}}},
    {"$project": {
        "carrier.code": 1,
        "Ratio_late aircraft": {"$divide": 
            ["$statistics.# of delays.late aircraft", "$statistics.flights.delayed"]},
        "Ratio_weather": {"$divide": 
            ["$statistics.# of delays.weather", "$statistics.flights.delayed"]},
        "Ratio_security": {"$divide": 
            ["$statistics.# of delays.security", "$statistics.flights.delayed"]},
        "Ratio_national aviation system": {"$divide": 
            ["$statistics.# of delays.national aviation system",
             "$statistics.flights.delayed"]},
        "Ratio_carrier": {"$divide": 
            ["$statistics.# of delays.carrier", "$statistics.flights.delayed"]}}},
    {"$group": {"_id": "$carrier.code",
                "late": {"$avg": "$Ratio_late aircraft"}, 
                "weather": {"$avg": "$Ratio_weather"}, 
                "security": {"$avg": "$Ratio_security"}, 
                "system": {"$avg": "$Ratio_national aviation system"}, 
                "carrier": {"$avg": "$Ratio_carrier"}}
    }
]
curs23 = air.aggregate(pipeline23)
```

```
In [15]: query23 = list(curs23)
```

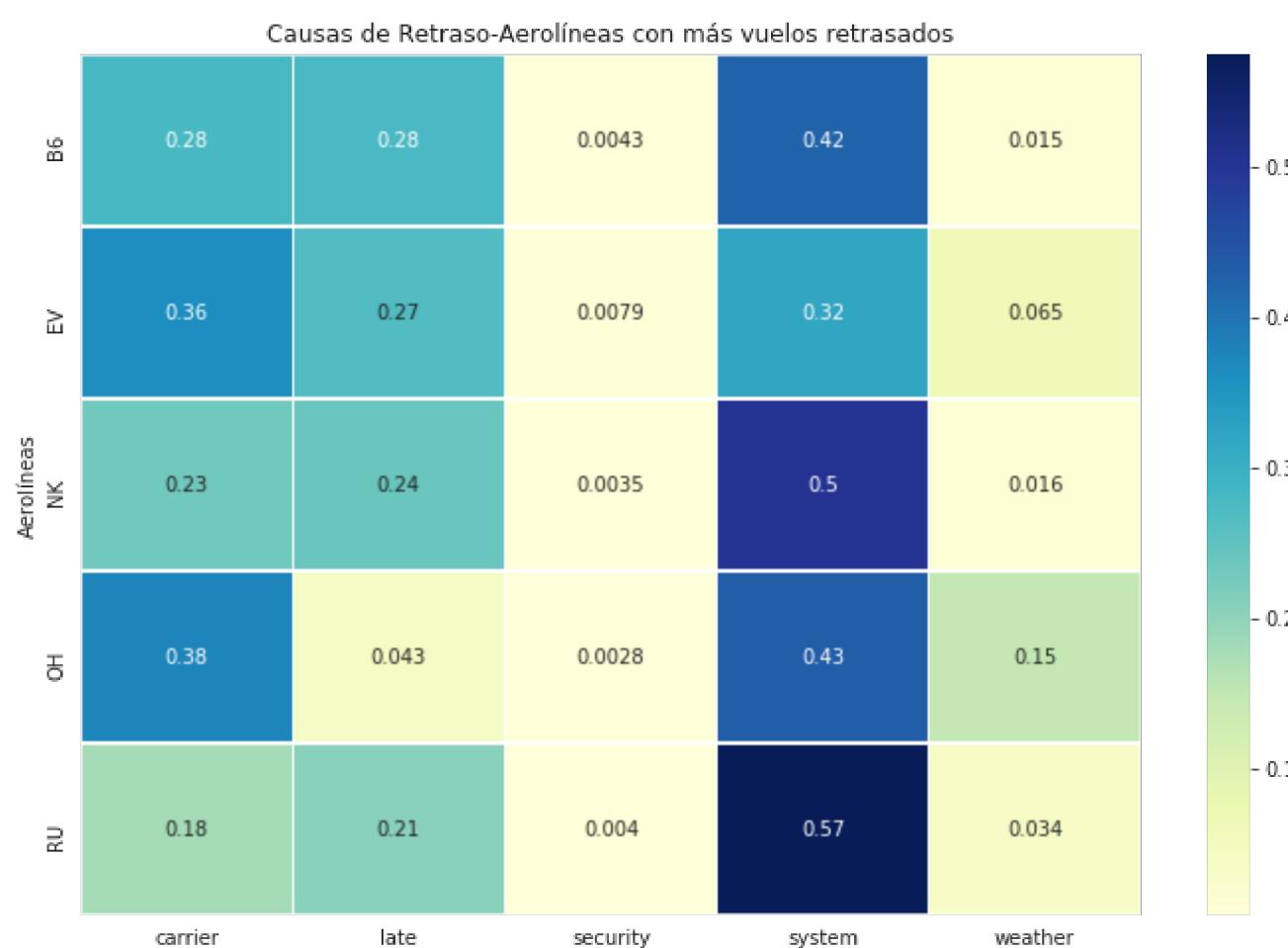
```
In [16]: df23 = pd.DataFrame(query23).set_index("_id").sort_index()
df23.index.name = "Aerolíneas"
df23
```

Out[16]:

	carrier	late	security	system	weather
<b>Aerolíneas</b>					
<b>B6</b>	0.28	0.28	0.00	0.42	0.02
<b>EV</b>	0.36	0.27	0.01	0.32	0.06
<b>NK</b>	0.23	0.24	0.00	0.50	0.02
<b>OH</b>	0.38	0.04	0.00	0.43	0.15
<b>RU</b>	0.18	0.21	0.00	0.57	0.03

Si se representa gráficamente los resultados anteriores,

```
In [17]: fig3, ax = plt.subplots(1, 1, figsize=(12,8));
ax = sns.heatmap(df23, annot=True, linewidths=.5, cmap="YlGnBu");
ax.set_title("Causas de Retraso-Aerolíneas con más vuelos retrasados");
```



Se observa que:

- las causas principales de retraso son en orden decreciente:
  - las debidas a los sistemas de navegación aérea
  - y las debidas al transportista y al aeronave
- RU es la aerolínea con más retrasos debido a los sistemas de navegación aérea
- OH es la aerolínea con más retrasos debido a causas del transportista
- B6 es la aerolínea con más retrasos debido a causas del aeronave

## 7.6 Minutos de retraso promedio por aerolínea y causa del retraso

También es interesante estudiar la variable **minutos de retraso asociada a una determinada causa** que da una idea de la importancia de los tipos de retraso, en términos de tiempo, en el que los vuelos no operaron a la hora prevista por dichas causas.

Nos preguntamos qué aerolíneas poseen, en promedio, mayores minutos de retraso para una causa determinada. Para ello se diseña la siguiente consulta:

- se filtran los años de estudio
- se agrupa por código de aerolínea
- se calculan los promedios de los minutos de retraso por causa de retraso

• **Query 24**

```
In [18]: pipeline24 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                      {"$group": {"_id": "$carrier.code",
                                  "totales": {"$avg": "$statistics.minutes delayed.total"},
                                  "late": {"$avg": "$statistics.minutes delayed.late aircraft"},
                                  "weather": {"$avg": "$statistics.minutes delayed.weather"},
                                  "carrier": {"$avg": "$statistics.minutes delayed.carrier"},
                                  "security": {"$avg": "$statistics.minutes delayed.security"},
                                  "system": {"$avg": "$statistics.minutes delayed.national aviation system"}
                                },
                      {"$sort": {"totales": -1}}
                    ]
curs24 = air.aggregate(pipeline24);

In [19]: query24 = list(curs24)

In [20]: df24 = pd.DataFrame(query24).set_index("_id")
df24.index.name = "Aerolíneas"
df24.columns.name = "Minutos Retraso Promedio"
df24.head(10)
```

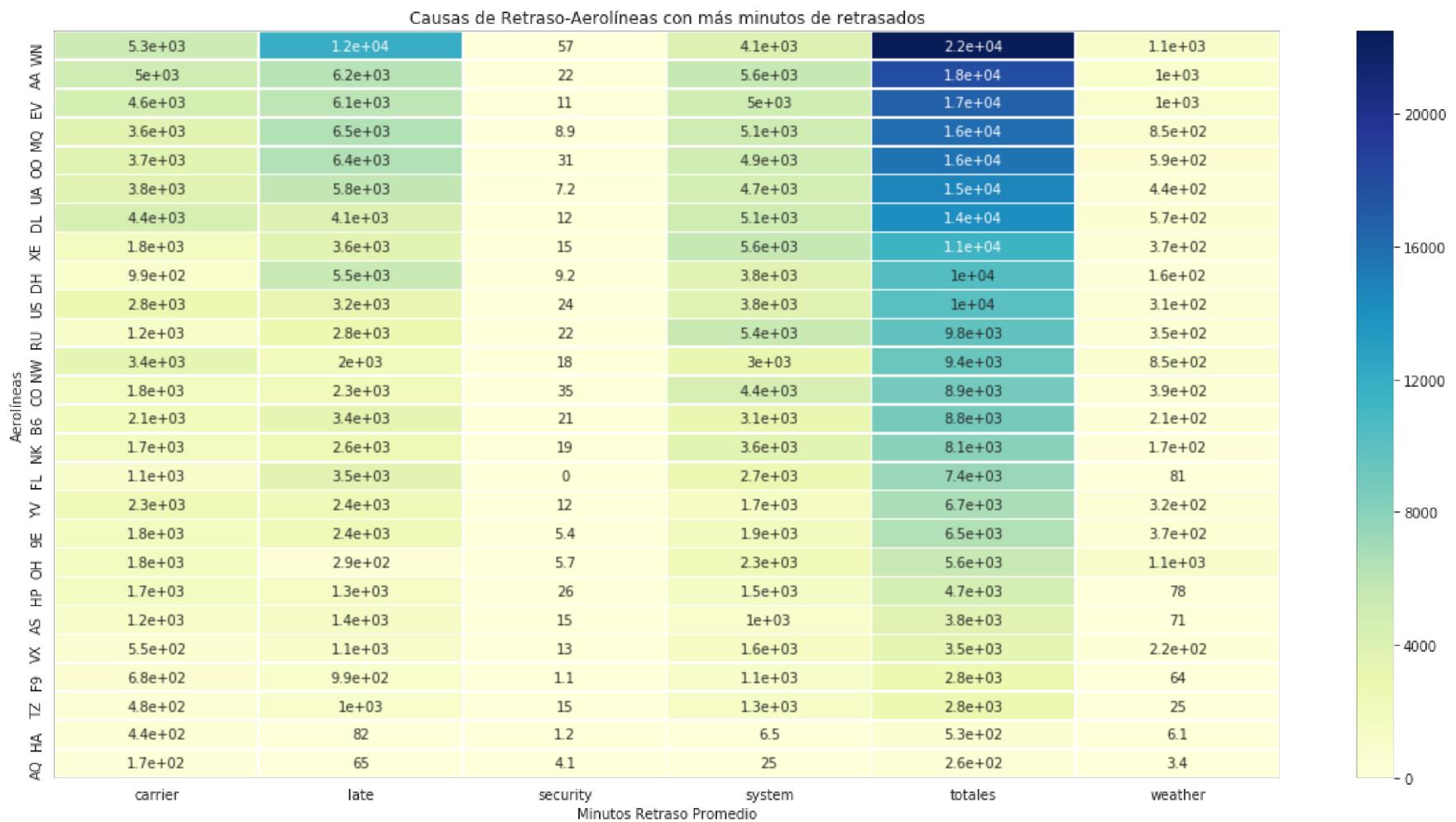
Out[20]:

	Minutos Retraso Promedio	carrier	late	security	system	totales	weather
<b>Aerolíneas</b>							
<b>WN</b>	5,252.72	11,913.96		57.01	4,122.39	22,472.08	1,126.01
<b>AA</b>	5,046.01	6,246.92		21.68	5,619.18	17,962.41	1,028.63
<b>EV</b>	4,629.71	6,054.06		10.77	4,980.95	16,707.46	1,031.95
<b>MQ</b>	3,642.84	6,464.67		8.90	5,091.33	16,060.93	853.18
<b>OO</b>	3,679.65	6,437.29		30.55	4,925.81	15,662.02	588.70
<b>UA</b>	3,815.28	5,817.45		7.16	4,661.15	14,739.24	438.20
<b>DL</b>	4,421.29	4,059.22		11.54	5,076.41	14,136.45	567.98
<b>XE</b>	1,800.75	3,649.61		14.75	5,628.19	11,464.99	371.69
<b>DH</b>	985.95	5,462.13		9.22	3,767.43	10,384.43	159.70
<b>US</b>	2,771.30	3,206.97		24.00	3,816.41	10,130.89	312.22

Representando lo anterior de forma gráfica

```
In [21]: fig4, ax = plt.subplots(1, 1, figsize=(20, 10))

ax = sns.heatmap(df24, annot=True, linewidths=.5, cmap="YlGnBu");
ax.set_title("Causas de Retraso-Aerolíneas con más minutos de retrasados");
```



Se aprecia que dentro de las **cinco primeros aerolíneas (WN, AA, EV, MQ y OO)** con **más minutos promedio retrasados acumulados** tan sólo EV está entre las tres primeras aerolíneas que más vuelos promedio retrasados tienen. Para saber los nombres de estos tres:

#### • Query 25

```
In [22]: curs25 = air.find(
    {"carrier.code": {"$in": ["WN", "AA", "EV", "MQ", "OO"]}},
    {"carrier.code": 1, "carrier.name": 1, "_id": 0}
)
```

```
In [23]: query25 = list(curs25)
```

```
In [24]: query25 = [(doc["carrier"]['code'], doc["carrier"]['name']) for doc in query25]
query25 = set(query25)
```

Con esto, se obtienen los nombres de los aeropuertos

```
In [25]: query25
```

```
Out[25]: {('AA', 'American Airlines Inc.'),
          ('EV', 'Atlantic Southeast Airlines'),
          ('EV', 'ExpressJet Airlines Inc.'),
          ('MQ', 'American Eagle Airlines Inc.'),
          ('MQ', 'Envoy Air'),
          ('OO', 'SkyWest Airlines Inc.'),
          ('WN', 'Southwest Airlines Co.')}{'
```

Nos podemos preguntar **si estas aerolíneas son los que mayor número de vuelos acumulados** han tenido durante el período de estudio.

#### • Query 26

```
In [26]: pipeline26 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},
                      {"$group": {"_id": "$carrier.code",
                                 "Totales": {"$sum": "$statistics.flights.total"}},
                      },
                      ],
                      {"$sort": {"Totales": -1}}
                    ]
curs26 = air.aggregate(pipeline26)
```

```
In [27]: query26= list(curs26)
```

```
In [28]: df26 = pd.DataFrame(query26).set_index("_id")
df26.index.name = "Aerolíneas"
df26.head(10)
```

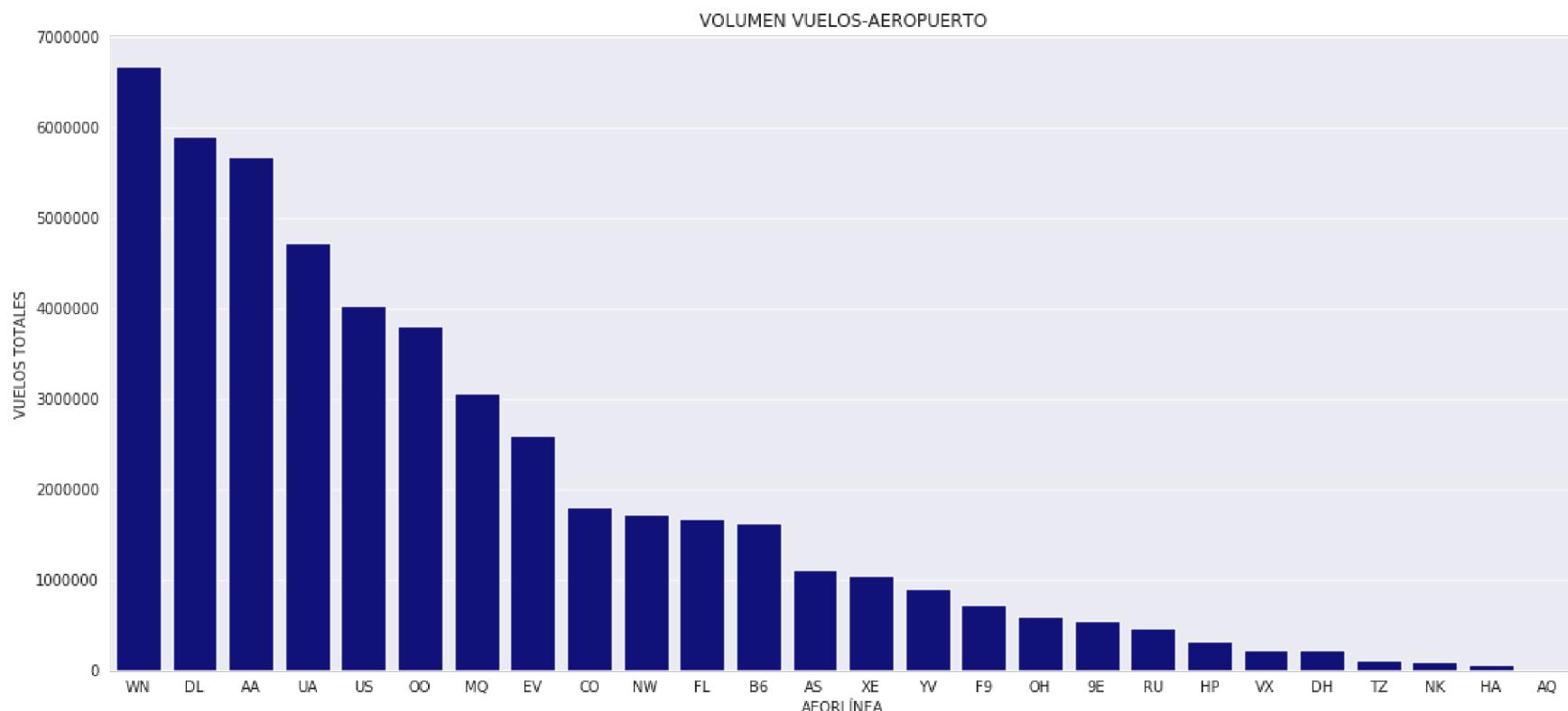
Out[28]:

	Totales
<b>Aerolíneas</b>	
WN	6,686,567.00
DL	5,913,326.00
AA	5,684,755.00
UA	4,733,756.00
US	4,028,264.00
OO	3,810,012.00
MQ	3,071,868.00
EV	2,594,421.00
CO	1,812,000.00
NW	1,720,129.00

Representando gráficamente los resultados, no se confirma que las **aerolíneas con más minutos de retraso son los que mayor volumen de tráfico han tenido durante el período de estudio** excepto en el caso de AA.

```
In [29]: fig5, ax1 = plt.subplots(1, 1, figsize=(18,8));

sns.barplot(x=df26.index, y= df26.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("VUELOS TOTALES");
ax1.set_xlabel("AEORLÍNEA");
ax1.set_title("VOLUMEN VUELOS-AEROPUERTO");
```



## 7.7 Evolución anual de Minutos de retraso para las aerolíneas con código WN, AA, EV.

En este apartado, se analiza la **evolución anual en el período de estudio de los minutos de retraso acumulados en las aerolíneas con código WN, AA y EV**. Se diseña la siguiente consulta:

- se filtra los años de estudio
- se filtran las aerolíneas
- se calculan los minutos de retraso

- **Query 27**

```
In [30]: pipeline27 = [{"$match": {"time.year": {"$nin": [2003, 2016]}, "carrier.code": {"$in": ["WN", "AA", "EV"]}}, {"$group": {"_id": {"Aerolinea": "$carrier.code", "Year": "$time.year"}, "Totales": {"$sum": "$statistics.minutes delayed.total"}}, {"$project": {"Aerolinea": "_id.Aerolinea", "Year": "_id.Year", "Totales": "Totales", "_id": 0}}, {"$sort": {"Year": 1}}], curs27 = air.aggregate(pipeline27);
```

```
In [31]: query27 = list(curs27)
```

```
In [32]: df27 = pd.DataFrame(query27).set_index(["Year", "Aerolinea"]).unstack()
df27
```

Out[32]:

Totales				
Aerolinea	AA	EV	WN	Year
2,004.00	6,913,685.00	1,612,482.00	3,568,638.00	
2,005.00	6,450,471.00	2,012,833.00	3,836,643.00	
2,006.00	6,859,910.00	2,429,812.00	4,506,198.00	
2,007.00	8,571,115.00	2,705,666.00	4,815,576.00	
2,008.00	7,611,852.00	2,087,904.00	5,518,102.00	
2,009.00	5,414,326.00	2,515,116.00	4,413,182.00	
2,010.00	4,416,772.00	2,017,924.00	5,032,529.00	
2,011.00	4,673,714.00	2,333,637.00	5,294,121.00	
2,012.00	4,878,434.00	5,316,063.00	4,736,446.00	
2,013.00	5,036,096.00	6,259,033.00	6,111,259.00	
2,014.00	5,689,227.00	5,924,028.00	8,013,351.00	
2,015.00	6,501,592.00	3,797,429.00	6,581,405.00	

Representando la tabla anterior

```
In [33]: fig6, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18,15));

sns.barplot(x=df27.index, y= df27.iloc[:,0], color= "darkblue", ax=ax1);
ax1.set_ylabel("Minutos retraso");
ax1.set_xlabel("Year");
ax1.set_title("AA");

sns.barplot(x=df27.index, y= df27.iloc[:,1], color= 'SkyBlue', ax=ax2);
ax2.set_ylabel("Minutos retraso");
ax2.set_xlabel("Year");
ax2.set_title("EV");

sns.barplot(x=df27.index, y= df27.iloc[:,2], color= 'IndianRed', ax=ax3);
ax3.set_ylabel("Minutos retraso");
ax3.set_xlabel("Year");
ax3.set_title("WN");
```



Se aprecia que:

- En la aerolínea AA, después de un descenso en los años 2008, 2009 y 2010, se vuelve a un aumento en los minutos de retraso en años posteriores
- En las aerolíneas WN y EV parece que los minutos de retraso han crecido durante los años de estudio.

## 7.8 Evolución mensual entre 2014-2015 de los Minutos de retraso, Ratio de vuelos cancelados, Ratio de vuelos retrasados y vuelos totales para las aerolíneas de WN, AA y EV.

En este apartado, se analiza la evolución mensual de dichos indicadores durante los años 2014-2015 en las aerolíneas de WN, AA y EV. Se diseña la siguiente consulta:

- se filtra los años de estudio
- se filtran las aerolíneas
- se calculan los indicadores

• **Query 28**

```
In [34]: pipeline28 = [{"$match": {"time.year": {"$in": [2014, 2015]}, "carrier.code": {"$in": ["WN", "AA", "EV"]}}}, {"$group": {"_id": {"Aerolinea": "$carrier.code", "Year": "$time.year", "Month": "$time.month"}, "minTotales": {"$sum": "$statistics.minutes delayed.total"}, "Totales": {"$sum": "$statistics.flights.total"}, "Cancelados": {"$sum": "$statistics.flights.cancelled"}, "Retrasados": {"$sum": "$statistics.flights.delayed"}}}, {"$project": {"Aerolinea": "$_id.Aerolinea", "Year": "$_id.Year", "Month": "$_id.Month", "minTotales": "$minTotales", "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]}, "Ratio_Cancelados": {"$divide": ["$Cancelados", "$Totales"]}, "Vuelos_Totales": "$Totales", "_id": 0}}, {"$sort": {"Year": 1, "Month": 1}}]
```

curs28 = air.aggregate(pipeline28);

```
In [35]: query28 = list(curs28)
```

```
In [36]: df28 = pd.DataFrame(query28).set_index(["Year", "Month", "Aerolinea"]).unstack()
```

Se crea el gráfico que nos da la **evolución de Vuelos Totales** en los tres aeropuertos seleccionados

```
In [37]: df28["Vuelos_Totales"].plot(kind= "line", figsize=(25,8), title = "Vuelos_Totales");
```

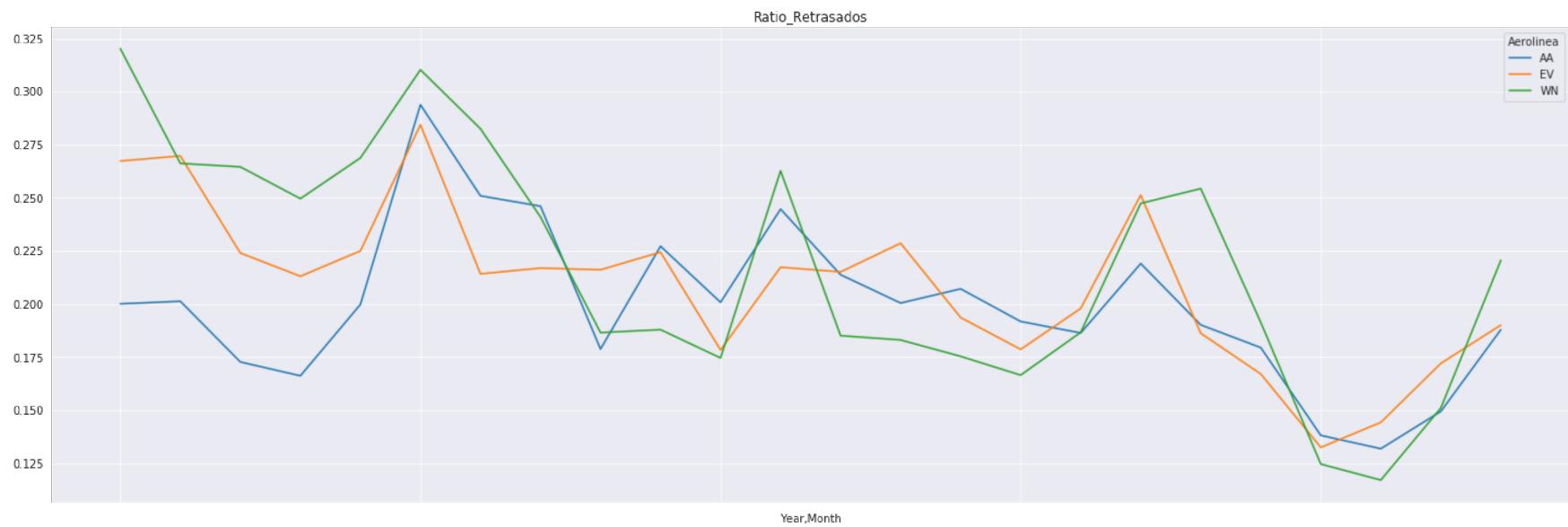


Se aprecia que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015:

- En WN crece el número de vuelos
- En EV desciende el número de vuelos
- En AA se produce un gran aumento en el número de vuelos durante el último semestre de 2015

Para los **Ratios de los Vuelos Retrasados**

```
In [38]: df28["Ratio_Retrasados"].plot(kind= "line", figsize=(25,8), title = "Ratio_Retrasados");
```

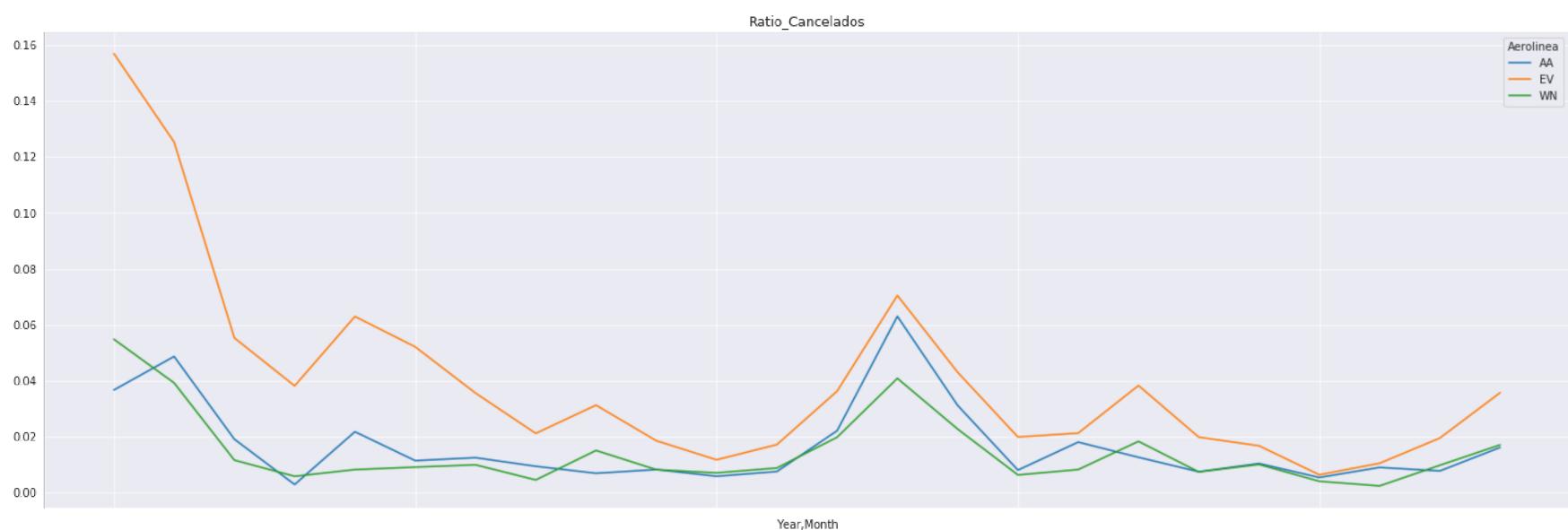


Parece que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015::

- En las **tres aerolíneas el Ratio de retrasados disminuye.**

Para los **Ratios de los Vuelos Retrasados**

```
In [39]: df28["Ratio_Cancelados"].plot(kind= "line", figsize=(25,8), title = "Ratio_Cancelados");
```

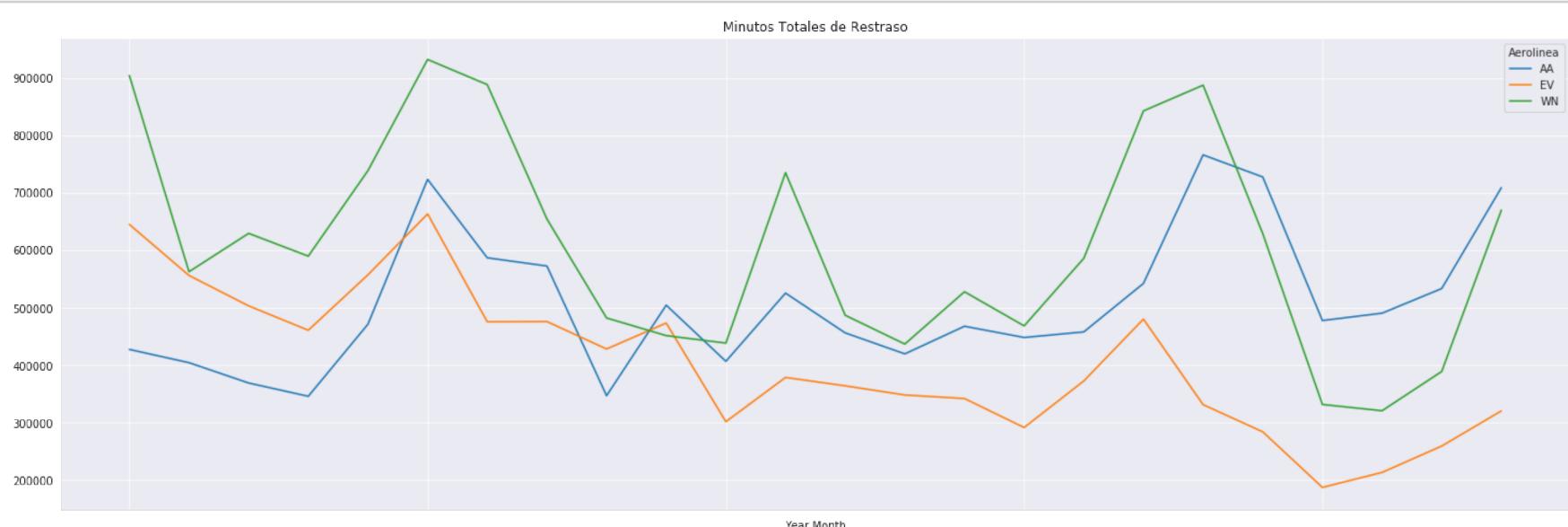


Parece que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015::

- En las **tres aerolíneas el Ratio de Cancelaciones permanece estable.**

Para los **Minutos de retraso totales**

```
In [40]: df28["minTotales"].plot(kind= "line", figsize=(25,8), title = "Minutos Totales de Retraso");
```



Parece que, de acuerdo con la tendencia y no teniendo en cuenta la estacionalidad, durante 2014-2015::

- En las **tres aerolíneas los minutos de retraso totales disminuyen ligeramente.**

In [ ]:

## 8. Relación entre Aeropuertos - Aerolíneas.

En este capítulo anterior se descubrieron las compañías aéreas y los aeropuertos donde los minutos de retraso promedio son mayores:

En este capítulo se analizan, para esos aeropuertos y compañías

1- los volúmenes de tráfico promedio

2- los ratios de:

- volumen de los vuelos totales,
- volumen de vuelos retrasado
- volumen de vuelos cancelados

### 8.1 Importación de librerías necesarias

Inicialmente se cargan las librerías de python necesarias.

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from pymongo import MongoClient

sns.set_style("darkgrid")

pd.options.display.float_format = '{:.2f}'.format
```

### 8.2 Conexión con MONGO ATLAS / LOCAL

```
In [2]: #Mongo Atlas
#URI ="mongodb://sato:<PASSWORD>@satoclusterfaa-shard-00-00-gst6h.|
#azure.mongodb.net:27017,satoclusterfaa-shard-00-01-gst6h.azure.|
#mongodb.net:27017,satoclusterfaa-shard-00-02-gst6h.azure.mongodb|
#.net:27017/test?ssl=true&replicaSet=SatoClusterFAA-shard-0&authSource=admin&retryWrites=true"

#client = MongoClient(URI)
#db = client.FAA_Airlines
#local
client = MongoClient()#"mongodb://localhost:27017")
db = client.airports
```

Se crea la variable **air** para facilitar las consultas.

```
In [3]: air = db.airlines
```

### 8.3 Evolución histórica de Ratios Vuelos Puntuales, Retrasados y Cancelados respecto a Vuelos Totales por aerolínea WN, AA, EV, MQ y OO y aeropuerto de estudio LGA, ORD, EWR y SFO.

En este apartado se va a extraer la evolución histórica, dentro del período de estudio, de los **ratios de vuelos puntuales, retrasados y cancelados respecto a los vuelos totales por aerolínea y aeropuerto de estudio**.

Para ello, se realiza la siguiente consulta en la que:

- se filtran los años de interés
- se agrupa por aerolínea, aeropuerto, año y mes
- en dicha agrupación, se calcula la suma de:
  - vuelos retrasados
  - vuelos cancelados
  - vuelos puntuales
- se crean los campos:
  - ratios vuelos retrasados
  - vuelos vuelos cancelados
  - vuelos vuelos puntuales

Además, de la consulta, se va a **generar una colección "Histórico\_Aeropuertos\_Aerolíneas"** para facilitar las consultas posteriores..

- **Query 29**

```
In [4]: pipeline29 = [{"$match": {"time.year": {"$nin": [2003, 2016]}},  
                      {"$group": {"_id":  
                                {"Aerolinea": "$carrier.code",  
                                 "Aeropuerto": "$airport.code",  
                                 "year": "$time.year",  
                                 "month": "$time.month"},  
                                "Totales": {"$sum": "$statistics.flights.total"},  
                                "Retrasados": {"$sum": "$statistics.flights.delayed"},  
                                "Cancelados": {"$sum": "$statistics.flights.cancelled"},  
                                "Puntuales": {"$sum": "$statistics.flights.on_time"}  
                            }  
                      },  
                      {"$project": {  
                                "Totales": "$Totales",  
                                "Ratio_Retrasados": {"$divide": ["$Retrasados", "$Totales"]},  

```

Se crea la variable **AA** para facilitar las consultas.

```
In [5]: #Se almacena la colección en una variable  
AA = db.Historico_Aeropuertos_Aerolineas
```

Una vez obtenida la nueva colección, se procede a utilizarla.

## 8.4 Promedio de vuelos de las compañías WN, AA, EV, MQ y OO en los aeropuertos LGA, ORD, EWR y SFO durante los años de estudio.

Con la siguiente consulta se pretende calcular el total promedio de vuelos en los años de estudio

- **Query 30**

```
In [6]: pipeline30 = [{"$match": {"_id.Aeropuerto": {"$in": ["LGA", "ORD", "EWR", "SFO"]}, "_id.Aerolinea": {"$in": ["WN", "AA", "EV", "MQ", "OO"]}}}, {"$group": {"_id": {"Aeropuerto": "$_id.Aeropuerto", "Aerolinea": "$_id.Aerolinea"}, "Totales": {"$avg": "$Totales"}}}, {"$project": {"Aeropuerto": "$_id.Aeropuerto", "Aerolinea": "$_id.Aerolinea", "Totales": "$Totales", "_id": 0}}]
```

```
curs30 = AA.aggregate(pipeline30)
```

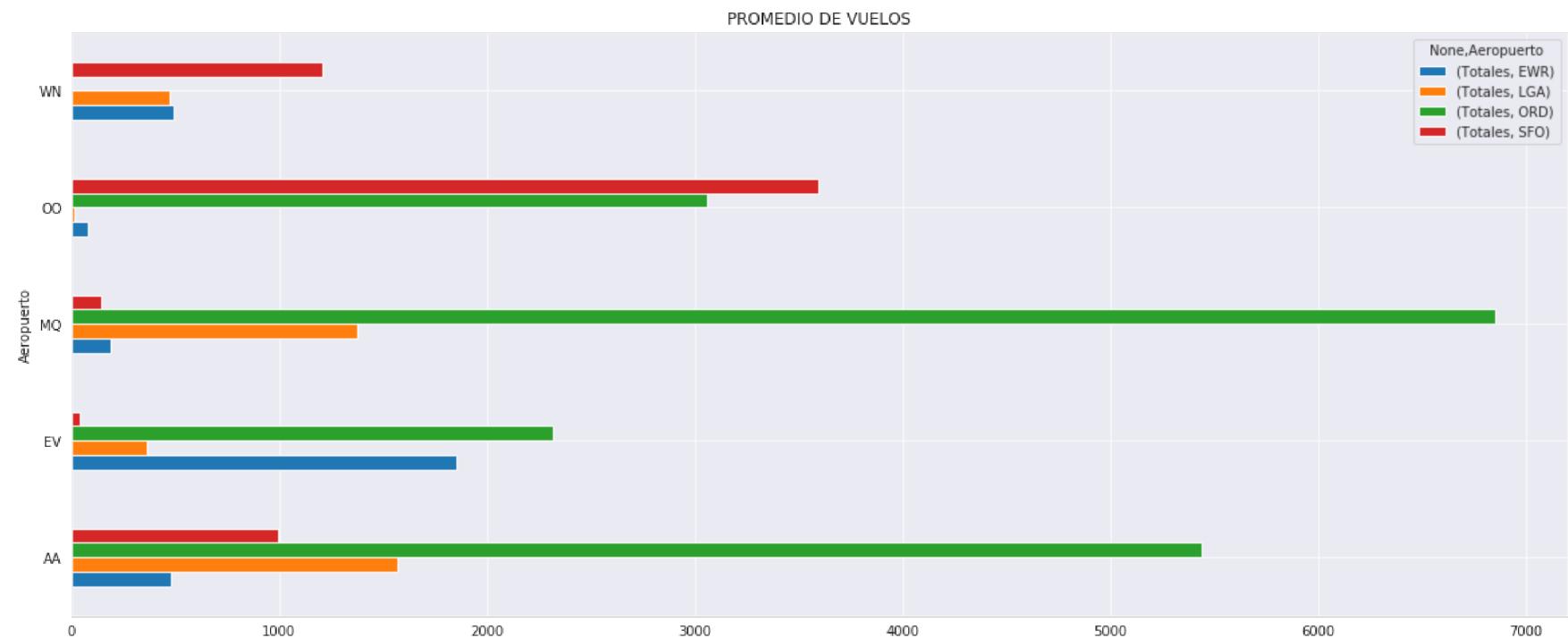
```
In [7]: query30 = list(curs30)
```

```
In [8]: df30 = pd.DataFrame(query30).set_index(["Aerolinea", "Aeropuerto"]).unstack()
df30.index.name = "Aeropuerto"
df30.head(10)
```

Out[8]:

Totales					
Aeropuerto	EWR	LGA	ORD	SFO	
Aeropuerto					
AA	477.36	1,568.11	5,439.27	993.18	
EV	1,854.72	363.71	2,319.36	43.65	
MQ	189.14	1,375.27	6,856.83	147.14	
OO	77.44	16.93	3,061.15	3,598.30	
WN	494.66	475.65	nan	1,211.98	

```
In [9]: df30.plot(kind = "barh", figsize= (20,8), title = "PROMEDIO DE VUELOS");
```



Se aprecia que:

- Para las aerolíneas **AA**, **EV** y **MQ** las rutas con mayor volumen de vuelos promedio tienen origen y destino en ORL.
- Para las aerolíneas **OO** y **WN** las rutas con mayor volumen de vuelos promedio tienen origen y destino en SFQ.

## 8.6 Combinaciones aeropuertos-aerolíneas con peores Ratios de Vuelos Puntuales, Retrasados y Cancelados respecto a Vuelos Totales.

Partiendo de la colección creada, se plantea la pregunta de cuáles son las combinaciones **aeropuertos-aerolíneas que peores Ratios medios han tenido**, durante los años de estudio.

Para esta pregunta se plantea la siguiente consulta.

- Se agrupa por **aeropuerto y aerolínea**
- Se calculan los **ratios medios**

- **Query 31**

```
In [10]: pipeline31 = [{"$match": {"_id.Aeropuerto": {"$in": ["LGA", "ORD", "EWR", "SFO"]},  
                      "_id.Aerolinea": {"$in": ["WN", "AA", "EV", "MQ", "OO"]}}  
},  
                      {"$group": {"_id": {"Aeropuerto": "$_id.Aeropuerto",  
                               "Aerolinea": "$_id.Aerolinea"},  
                               "Ratio_Retrasados": {"$avg": "$Ratio_Retrasados"},  
                               "Ratio_Cancelados": {"$avg": "$Ratio_Cancelados"},  
                               "Ratio_Puntuales": {"$avg": "$Ratio_Puntuales"}  
},  
},  
                      {"$project": {"Aeropuerto": "$_id.Aeropuerto",  
                               "Aerolinea": "$_id.Aerolinea",  
                               "Ratio_Retrasados": "$Ratio_Retrasados",  
                               "Ratio_Cancelados": "$Ratio_Cancelados",  
                               "Ratio_Puntuales": "$Ratio_Puntuales",  
                               "_id": 0  
},  
},  
]  
  
curs31 = AA.aggregate(pipeline31)
```

El cursor obtenido se convierte en lista y se almacena en una lista y se convierte el resultado en un dataframe. Se ofrece una vista de los 10 primeros aeropuertos.

```
In [11]: query31 = list(curs31)
```

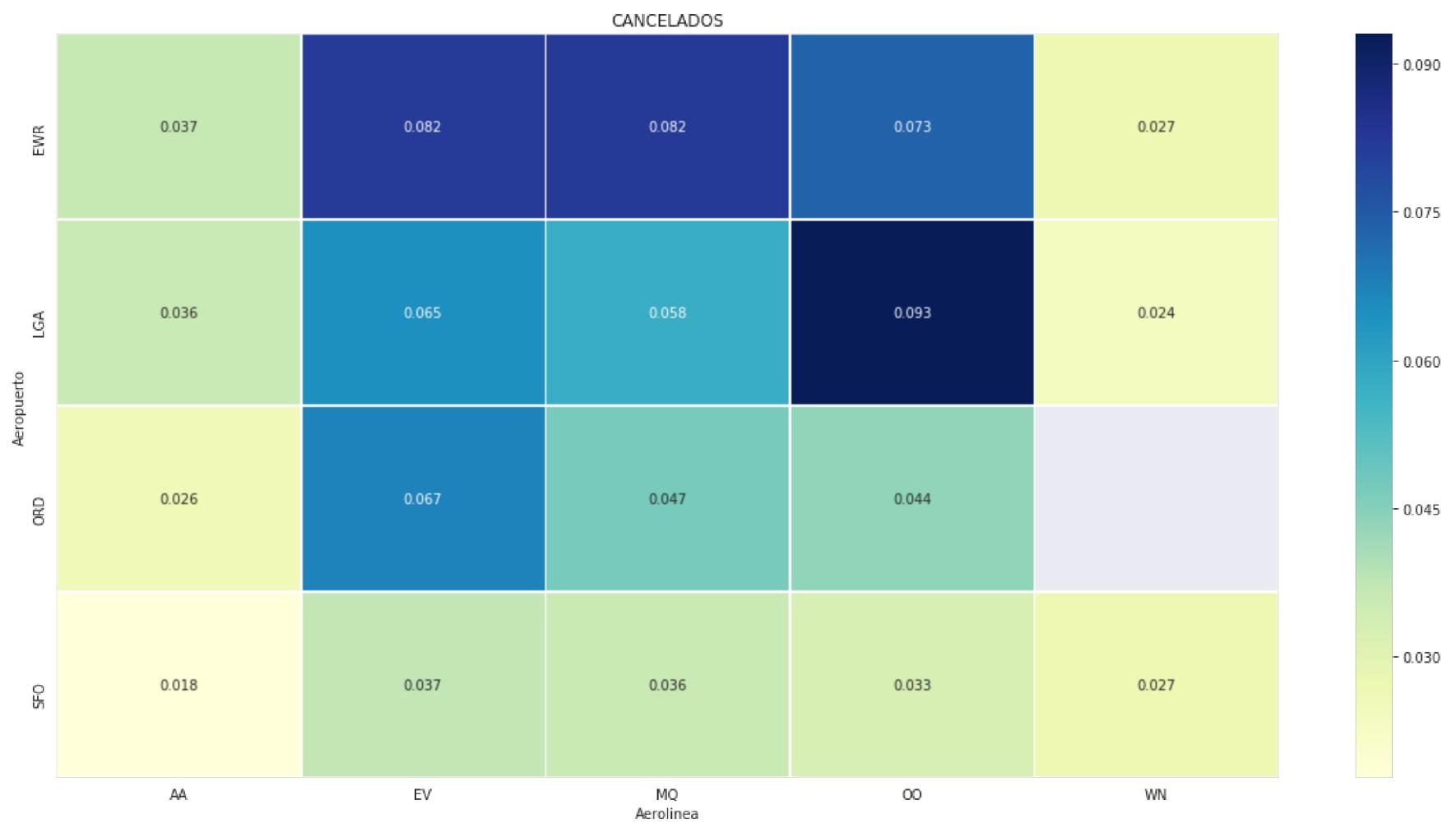
```
In [12]: df31 = pd.DataFrame(query31).set_index(["Aeropuerto", "Aerolinea"]).unstack()  
df31.index.name = "Aeropuerto"  
df31.head(10)
```

Out[12]:

Aerolinea	Ratio_Cancelados				Ratio_Puntuales				Ratio_Retrasados						
	AA	EV	MQ	OO	WN	AA	EV	MQ	OO	WN	AA	EV	MQ	OO	WN
Aeropuerto															
EWR	0.04	0.08	0.08	0.07	0.03	0.66	0.55	0.59	0.49	0.67	0.30	0.35	0.33	0.41	0.30
LGA	0.04	0.07	0.06	0.09	0.02	0.68	0.62	0.68	0.59	0.68	0.27	0.31	0.26	0.28	0.29
ORD	0.03	0.07	0.05	0.04	nan	0.74	0.67	0.71	0.72	nan	0.23	0.25	0.24	0.23	nan
SFO	0.02	0.04	0.04	0.03	0.03	0.69	0.59	0.75	0.70	0.67	0.29	0.38	0.21	0.26	0.30

```
In [13]: fig1, ax1 = plt.subplots(1, 1, figsize=(20,10));

ax1 = sns.heatmap(df31["Ratio_Cancelados"], annot=True, linewidths=.5, cmap="YlGnBu");
ax1.set_title("CANCELADOS");
```

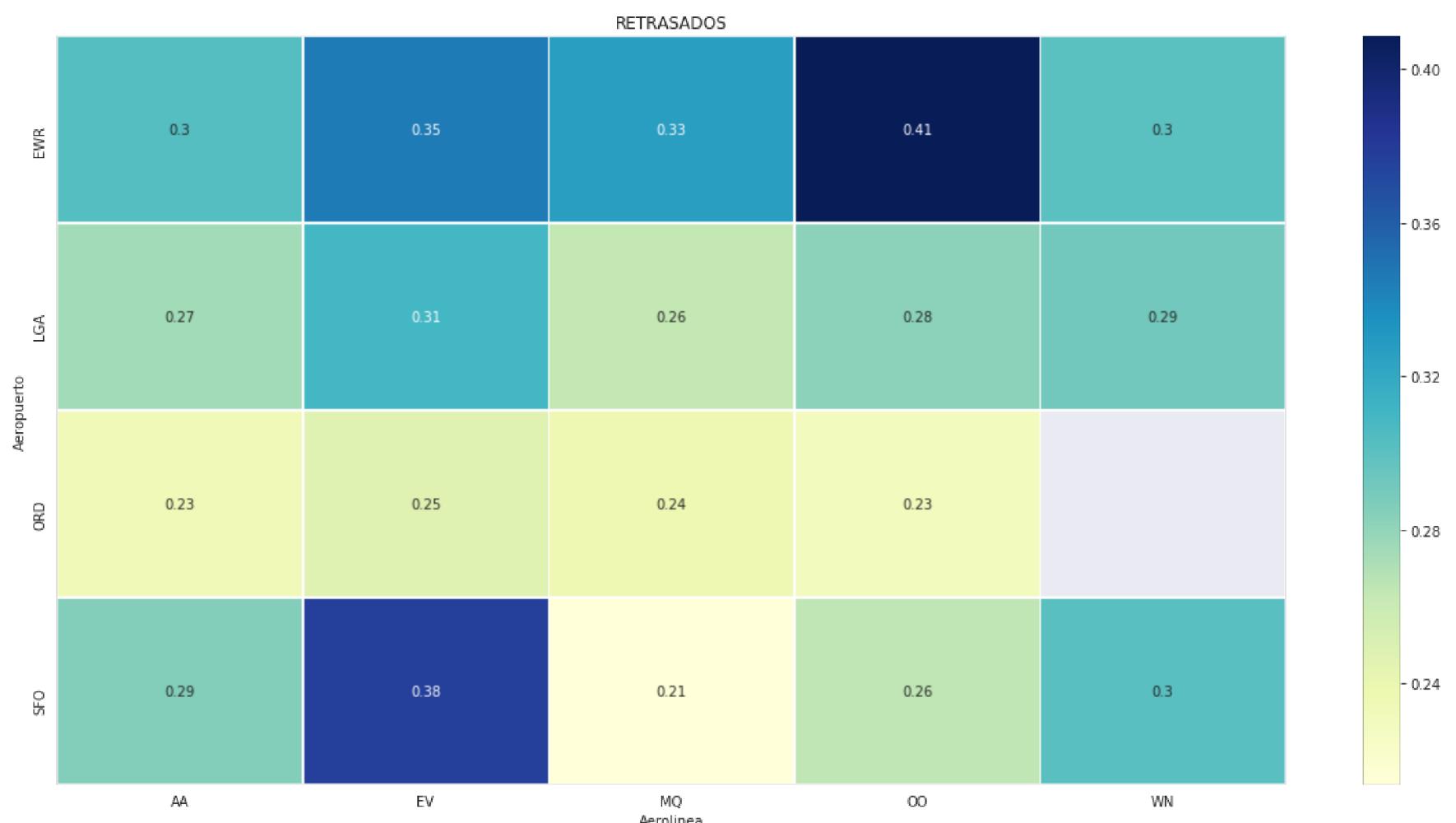


Del gráfico anterior se aprecia:

- De los cuatro aeropuertos, las aerolíneas que más cancelaciones han tenido son **EV, MQ y OO**.
- La que menos cancelaciones ha tenido son **AA y WN**.
- La aerolínea **WN no opera en el aeropuerto de ORD**.

```
In [14]: fig2, ax2 = plt.subplots(1, 1, figsize=(20, 10));

ax2 = sns.heatmap(df31["Ratio_Retrasados"], annot=True, linewidths=.5, cmap="YlGnBu");
ax2.set_title("RETRASADOS");
```

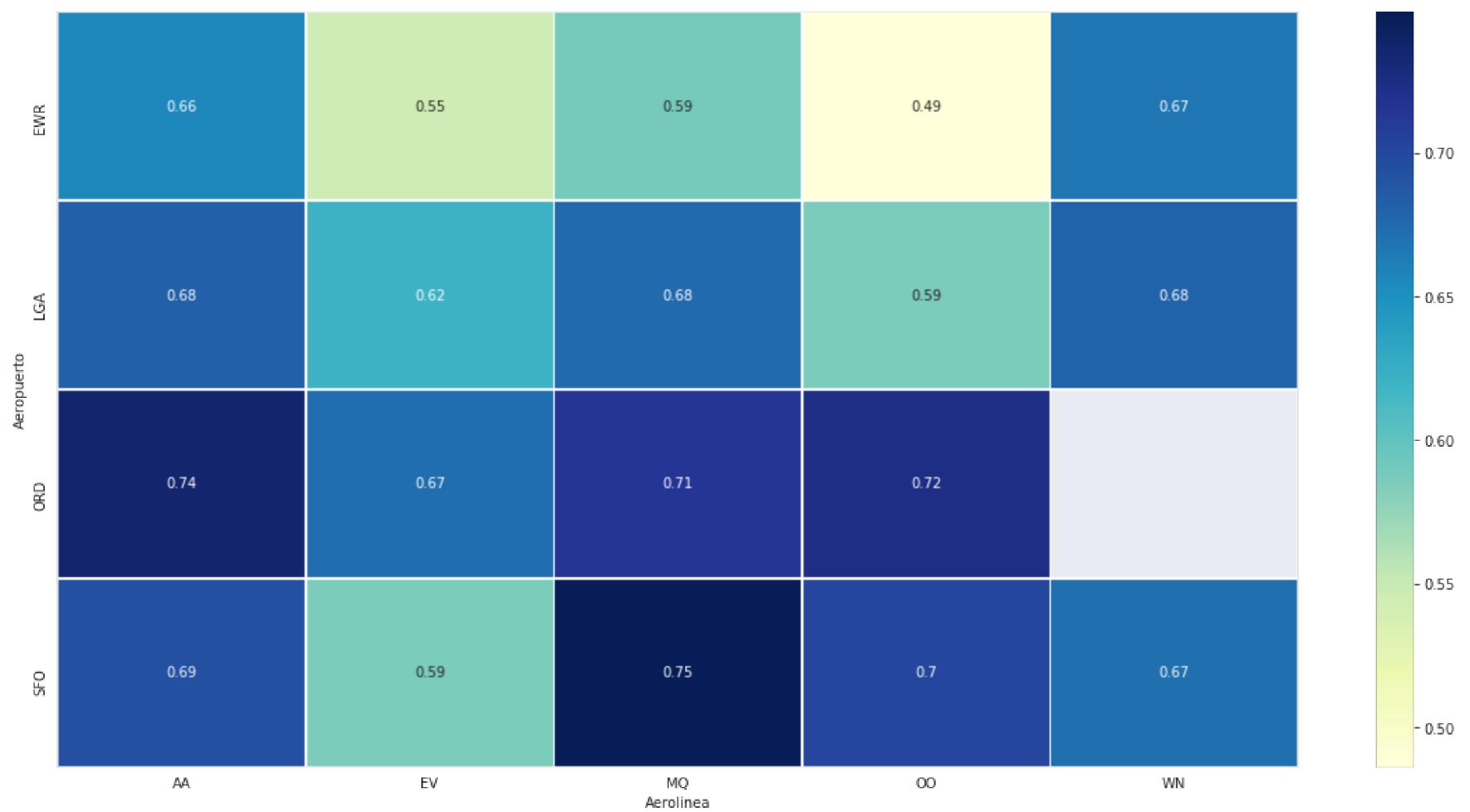


Se aprecia que:

- El aeropuerto con más retrasos son **EWR y LGA** y fundamentalmente debidos a las aerolíneas **OO y EV**
- El aeropuerto con menos retrasos para esas aerolíneas fue **ORL**

```
In [15]: fig3, ax3 = plt.subplots(1, 1, figsize=(20,10));

ax3 = sns.heatmap(df31["Ratio_Puntuales"], annot=True, linewidths=.5, cmap="YlGnBu");
ax2.set_title("PUNTUALES");
```



Se aprecia que las combinaciones aeropuerto-aerolínea más puntuales han sido:

- MQ-SFO,
- AA-ORL,
- OO-ORD

Como recordatorio se muestran a continuación los nombres de las compañías y aeropuertos junto a su código

#### • Query 32. Compañías Aéreas

```
In [16]: curs32 = air.find(
    {"carrier.code": {"$in": ["WN", "AA", "EV", "MQ", "OO"]}},
    {"carrier.code": 1, "carrier.name": 1, "_id": 0}
)
```

```
In [17]: query32 = list(curs32)
```

```
In [18]: query32 = [(doc["carrier"]['code'], doc["carrier"]['name']) for doc in query32]
query32 = set(query32)
```

```
In [19]: query32
```

```
Out[19]: {('AA', 'American Airlines Inc.'),
          ('EV', 'Atlantic Southeast Airlines'),
          ('EV', 'ExpressJet Airlines Inc.'),
          ('MQ', 'American Eagle Airlines Inc.'),
          ('MQ', 'Envoy Air'),
          ('OO', 'SkyWest Airlines Inc.'),
          ('WN', 'Southwest Airlines Co.')}
```

#### • Query 33. Aeropuertos

```
In [20]: curs33 = air.find(
    {"airport.code": {"$in": ["LGA", "ORD", "EWR", "SFO"]}},
    {"airport.code": 1, "airport.name": 1, "_id": 0}
)
```

```
In [21]: query33 = list(curs33)
```

```
In [22]: query33 = [(doc["airport"]['code'], doc["airport"]['name']) for doc in query33]
query33 = set(query33)
```

```
In [23]: query33
```

```
Out[23]: {('EWR', 'Newark, NJ: Newark Liberty International'),
('LGA', 'New York, NY: LaGuardia'),
('ORD', "Chicago, IL: Chicago O'Hare International"),
('SFO', 'San Francisco, CA: San Francisco International')}
```

```
In [ ]:
```