Branch: master ▾ Find file Copy path

Flight_delay / Project_ML.ipynb

dongzhang84 Add files via upload

5a13a77 16 minutes ago

1 contributor

Raw Blame History

3089 lines (3088 sloc) | 465 KB

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

In [2]:
```
data = pd.read_csv('airline_delay.csv')
```

In [3]:
```
data.shape
```

Out[3]: (256285, 21)

In [4]:
```
data.head(10)
```

Out[4]:

	year	month	carrier	carrier_name	airport	airport_name	arr_flights	arr_del15	carrier_ct	weather_ct	...
0	2003	6	AA	American Airlines Inc.	ABQ	Albuquerque, NM: Albuquerque International Sun...	307.0	56.0	14.68	10.79	...
1	2003	6	AA	American Airlines Inc.	ANC	Anchorage, AK: Ted Stevens Anchorage Internati...	90.0	27.0	7.09	2.00	...
2	2003	6	AA	American Airlines Inc.	ATL	Atlanta, GA: Hartsfield-Jackson Atlanta Intern...	752.0	186.0	33.99	27.82	...
3	2003	6	AA	American Airlines Inc.	AUS	Austin, TX: Austin - Bergstrom International	842.0	174.0	60.24	20.54	...
4	2003	6	AA	American Airlines Inc.	BDL	Hartford, CT: Bradley International	383.0	55.0	14.90	8.91	...
5	2003	6	AA	American Airlines Inc.	BHM	Birmingham, AL: Birmingham-Shuttlesworth Inter...	89.0	12.0	2.79	2.19	...
6	2003	6	AA	American Airlines Inc.	BNA	Nashville, TN: Nashville International	445.0	82.0	25.44	11.98	...
7	2003	6	AA	American Airlines Inc.	BOS	Boston, MA: Logan International	1266.0	225.0	69.43	23.66	...
8	2003	6	AA	American Airlines Inc.	BUR	Burbank, CA: Bob Hope	119.0	27.0	7.49	4.65	...
9	2003	6	AA	American Airlines Inc.	BWI	Baltimore, MD: Baltimore/Washington Internatio...	593.0	101.0	17.56	20.49	...

10 rows × 21 columns

In [5]:
```
data['delayrate'] = data['arr_del15']/data['arr_flights']
data['delaymin']  = data['arr_delay']/data['arr_del15']
```

In [6]:
```
data.head(10)
```

Out[6]:

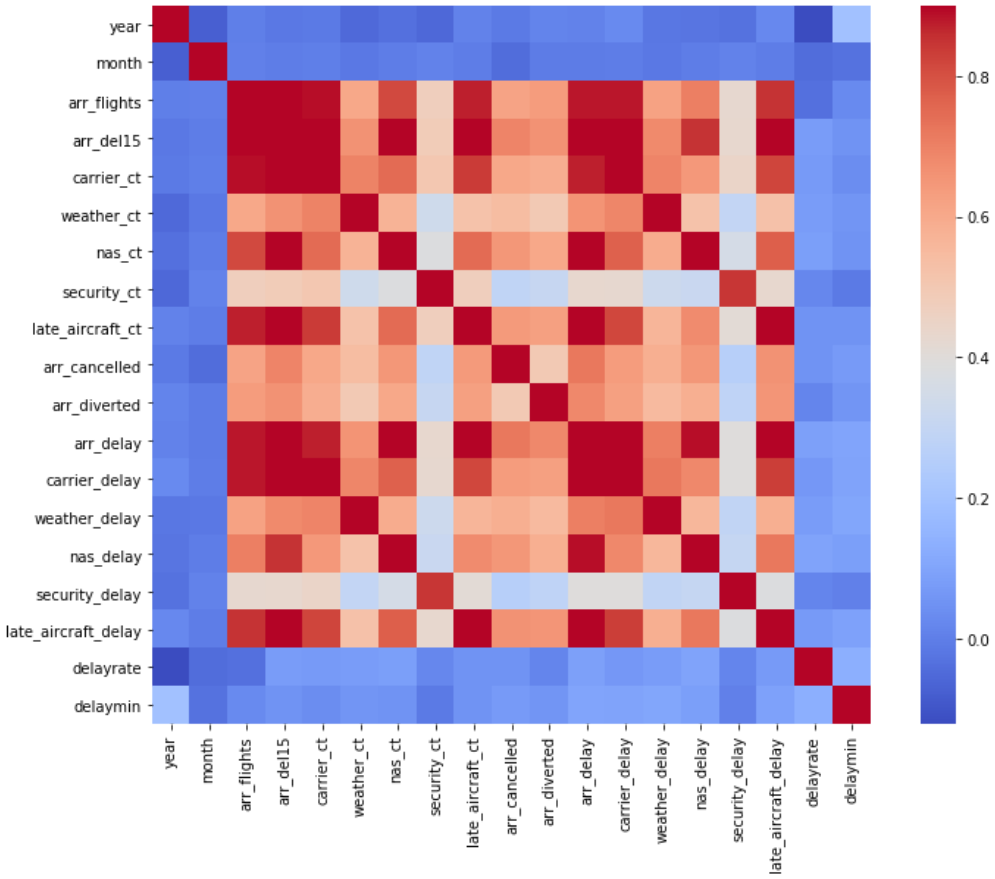
	year	month	carrier	carrier_name	airport	airport_name	arr_flights	arr_del15	carrier_ct	weather_ct	...
0	2003	6	AA	American Airlines Inc.	ABQ	Albuquerque, NM: Albuquerque International Sun...	307.0	56.0	14.68	10.79	...
1	2003	6	AA	American Airlines Inc.	ANC	Anchorage, AK: Ted Stevens Anchorage Internati...	90.0	27.0	7.09	2.00	...
2	2003	6	AA	American Airlines Inc.	ATL	Atlanta, GA: Hartsfield-Jackson Atlanta Intern...	752.0	186.0	33.99	27.82	...
3	2003	6	AA	American Airlines Inc.	AUS	Austin, TX: Austin - Bergstrom	842.0	174.0	60.24	20.54	...

				Airlines Inc.		International					
4	2003	6	AA	American Airlines Inc.	BDL	Hartford, CT: Bradley International	383.0	55.0	14.90	8.91	...
5	2003	6	AA	American Airlines Inc.	BHM	Birmingham, AL: Birmingham-Shuttlesworth Inter...	89.0	12.0	2.79	2.19	...
6	2003	6	AA	American Airlines Inc.	BNA	Nashville, TN: Nashville International	445.0	82.0	25.44	11.98	...
7	2003	6	AA	American Airlines Inc.	BOS	Boston, MA: Logan International	1266.0	225.0	69.43	23.66	...
8	2003	6	AA	American Airlines Inc.	BUR	Burbank, CA: Bob Hope	119.0	27.0	7.49	4.65	...
9	2003	6	AA	American Airlines Inc.	BWI	Baltimore, MD: Baltimore/Washington Internatio...	593.0	101.0	17.56	20.49	...

10 rows × 23 columns

In [7]:
```
corrmat = data.corr()
plt.subplots(figsize=(12,9))
sns.heatmap(corrmat, vmax=0.9, square=True, cmap='coolwarm')
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1244a28d0>



In [8]:
```
data_train = data[['year', 'month','carrier','airport','delayrate','delaymin']]
data_train.dropna(how='all', inplace=True)
#data_train.loc[(data_train!=0).any(axis=1)]
data_train = data_train[data_train['carrier'].str.contains('AA|UA|DL|WN|AS')]
data_train = data_train[data_train['airport'].str.contains('ORD|JFK|ATL|MIA')]
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
```

In [9]:
```
data_train
```

Out[9]:

	year	month	carrier	airport	delayrate	delaymin
2	2003	6	AA	ATL	0.247340	44.698925
35	2003	6	AA	JFK	0.184830	50.872642
48	2003	6	AA	MIA	0.231731	49.762102
56	2003	6	AA	ORD	0.171605	55.135218
111	2003	6	AS	MIA	0.300000	40.000000
115	2003	6	AS	ORD	0.100000	65.333333
324	2003	6	DL	ATL	0.169686	42.808271
366	2003	6	DL	JFK	0.158084	37.833333
377	2003	6	DL	MIA	0.240000	33.069444
389	2003	6	DL	ORD	0.271429	45.075188
1053	2003	6	UA	ATL	0.202740	53.175676
1085	2003	6	UA	JFK	0.117978	43.857143
1095	2003	6	UA	MIA	0.170588	50.482759
1103	2003	6	UA	ORD	0.157272	52.081633
1250	2003	7	AA	ATL	0.271318	54.409524
1283	2003	7	AA	JFK	0.240876	60.838384
1296	2003	7	AA	MIA	0.230839	59.996063
1304	2003	7	AA	ORD	0.259292	79.535349
1359	2003	7	AS	MIA	0.354839	18.727273
1363	2003	7	AS	ORD	0.064516	25.000000
1572	2003	7	DL	ATL	0.207528	54.073215
1615	2003	7	DL	JFK	0.165158	46.294521
1626	2003	7	DL	MIA	0.285246	40.770115
1639	2003	7	DL	ORD	0.365657	72.049724
2303	2003	7	UA	ATL	0.257426	69.134615
2335	2003	7	UA	JFK	0.138743	47.000000
2345	2003	7	UA	MIA	0.232353	55.303797
2353	2003	7	UA	ORD	0.220259	76.719710
2499	2003	8	AA	ATL	0.315175	49.432099
2532	2003	8	AA	JFK	0.281530	65.095376
...	...	...	...	...	...	...
252987	2019	1	AA	ATL	0.183367	48.300546
253028	2019	1	AA	JFK	0.174355	68.922179
253039	2019	1	AA	MIA	0.176902	62.098660
253050	2019	1	AA	ORD	0.217593	79.277778
253091	2019	1	AS	ATL	0.083333	20.333333
253115	2019	1	AS	JFK	0.190909	57.035714
253132	2019	1	AS	ORD	0.260684	68.639344
253230	2019	1	DL	ATL	0.104614	65.900759
253298	2019	1	DL	JFK	0.145455	71.141304
253313	2019	1	DL	MIA	0.190278	50.094891
253328	2019	1	DL	ORD	0.245614	129.190476
254215	2019	1	UA	ATL	0.213235	47.620690

254276	2019	1	UA	MIA	0.194570	58.465116
254287	2019	1	UA	ORD	0.234800	111.539658
254323	2019	1	WN	ATL	0.145107	43.635659
254692	2019	2	AA	ATL	0.243154	49.819820
254733	2019	2	AA	JFK	0.164647	64.783550
254745	2019	2	AA	MIA	0.186288	60.515228
254756	2019	2	AA	ORD	0.248266	74.373048
254797	2019	2	AS	ATL	0.406250	36.307692
254821	2019	2	AS	JFK	0.221939	70.747126
254838	2019	2	AS	ORD	0.375000	79.236111
254932	2019	2	DL	ATL	0.148027	69.575865
254997	2019	2	DL	JFK	0.130396	87.368243
255011	2019	2	DL	MIA	0.165625	58.132075
255026	2019	2	DL	ORD	0.272358	117.174129
255907	2019	2	UA	ATL	0.213115	61.292308
255965	2019	2	UA	MIA	0.200000	59.602564
255975	2019	2	UA	ORD	0.233719	90.446002
256010	2019	2	WN	ATL	0.201450	53.604069

2755 rows × 6 columns

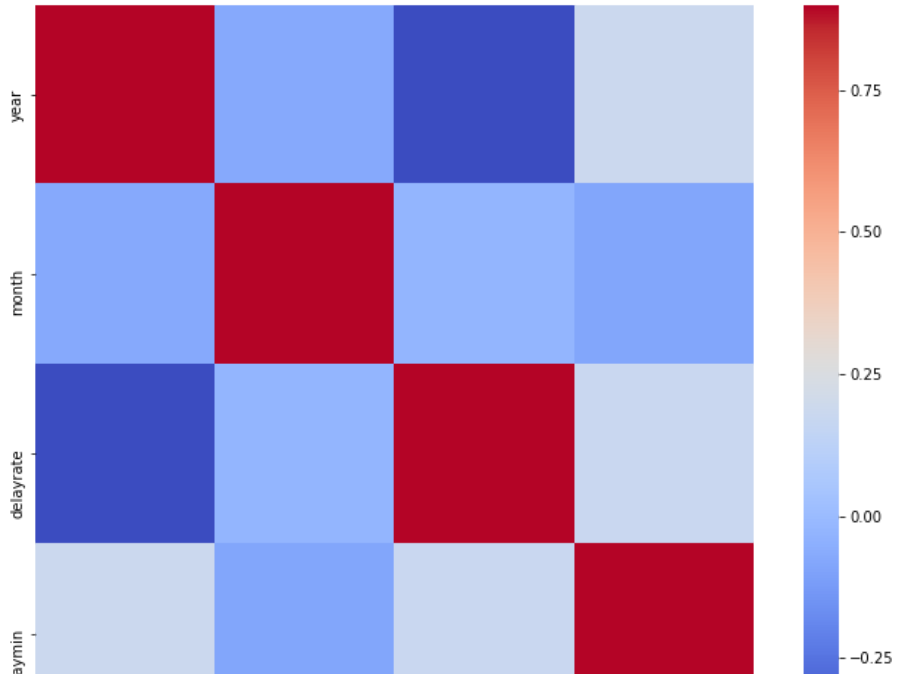
In [10]:
```
#data_train['delayrate'] = data_train['arr_del15']/data_train['arr_flights']
#data_train['delaymin']  = data_train['arr_delay']/data_train['arr_del15']
data_train = data_train[data_train['delaymin'] < 300]
data_train = data_train[data_train['delayrate'] < 0.7]
#data_train['delayrate'] = data_train['arr_del15'].div(data_train.arr_flights, axis=0)
#data_train['delaymin'] = data_train['arr_delay'].div(data_train.arr_del15, axis=0)
#data_train.drop(['arr_del15','arr_flights','arr_delay'],axis=1,inplace=True)
```

In [11]:
```
data_train.shape
```

Out[11]: (2741, 6)

In [12]:
```
corrmat = data_train.corr()
plt.subplots(figsize=(12,9))
sns.heatmap(corrmat, vmax=0.9, square=True, cmap='coolwarm')
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x10f3759e8>



In [13]:
```
data_train
```

Out[13]:

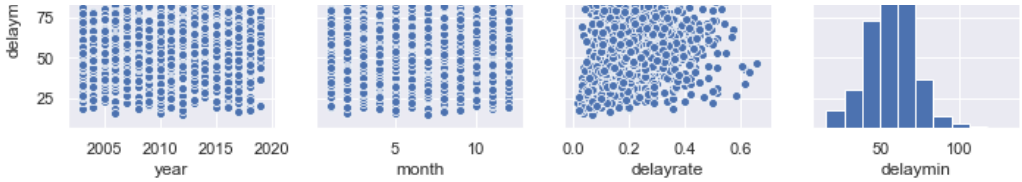
	year	month	carrier	airport	delayrate	delaymin
2	2003	6	AA	ATL	0.247340	44.698925
35	2003	6	AA	JFK	0.184830	50.872642
48	2003	6	AA	MIA	0.231731	49.762102
56	2003	6	AA	ORD	0.171605	55.135218
111	2003	6	AS	MIA	0.300000	40.000000
115	2003	6	AS	ORD	0.100000	65.333333
324	2003	6	DL	ATL	0.169686	42.808271
366	2003	6	DL	JFK	0.158084	37.833333
377	2003	6	DL	MIA	0.240000	33.069444
389	2003	6	DL	ORD	0.271429	45.075188
1053	2003	6	UA	ATL	0.202740	53.175676
1085	2003	6	UA	JFK	0.117978	43.857143
1095	2003	6	UA	MIA	0.170588	50.482759
1103	2003	6	UA	ORD	0.157272	52.081633
1250	2003	7	AA	ATL	0.271318	54.409524
1283	2003	7	AA	JFK	0.240876	60.838384
1296	2003	7	AA	MIA	0.230839	59.996063
1304	2003	7	AA	ORD	0.259292	79.535349
1359	2003	7	AS	MIA	0.354839	18.727273
1363	2003	7	AS	ORD	0.064516	25.000000
1572	2003	7	DL	ATL	0.207528	54.073215
1615	2003	7	DL	JFK	0.165158	46.294521
1626	2003	7	DL	MIA	0.285246	40.770115
1639	2003	7	DL	ORD	0.365657	72.049724
2303	2003	7	UA	ATL	0.257426	69.134615
2335	2003	7	UA	JFK	0.138743	47.000000
2345	2003	7	UA	MIA	0.232353	55.303797
2353	2003	7	UA	ORD	0.220259	76.719710
2499	2003	8	AA	ATL	0.315175	49.432099
2532	2003	8	AA	JFK	0.281530	65.095376
...	...	...	...	...	...	...
252987	2019	1	AA	ATL	0.183367	48.300546
253028	2019	1	AA	JFK	0.174355	68.922179
253039	2019	1	AA	MIA	0.176902	62.098660
253050	2019	1	AA	ORD	0.217593	79.277778
253091	2019	1	AS	ATL	0.083333	20.333333
253115	2019	1	AS	JFK	0.190909	57.035714
253132	2019	1	AS	ORD	0.260684	68.639344
253230	2019	1	DL	ATL	0.104614	65.900759
253298	2019	1	DL	JFK	0.145455	71.141304
253313	2019	1	DL	MIA	0.190278	50.094891
253328	2019	1	DL	ORD	0.245614	129.190476

254215	2019	1	UA	ATL	0.213235	47.620690
254276	2019	1	UA	MIA	0.194570	58.465116
254287	2019	1	UA	ORD	0.234800	111.539658
254323	2019	1	WN	ATL	0.145107	43.635659
254692	2019	2	AA	ATL	0.243154	49.819820
254733	2019	2	AA	JFK	0.164647	64.783550
254745	2019	2	AA	MIA	0.186288	60.515228
254756	2019	2	AA	ORD	0.248266	74.373048
254797	2019	2	AS	ATL	0.406250	36.307692
254821	2019	2	AS	JFK	0.221939	70.747126
254838	2019	2	AS	ORD	0.375000	79.236111
254932	2019	2	DL	ATL	0.148027	69.575865
254997	2019	2	DL	JFK	0.130396	87.368243
255011	2019	2	DL	MIA	0.165625	58.132075
255026	2019	2	DL	ORD	0.272358	117.174129
255907	2019	2	UA	ATL	0.213115	61.292308
255965	2019	2	UA	MIA	0.200000	59.602564
255975	2019	2	UA	ORD	0.233719	90.446002
256010	2019	2	WN	ATL	0.201450	53.604069

2741 rows × 6 columns

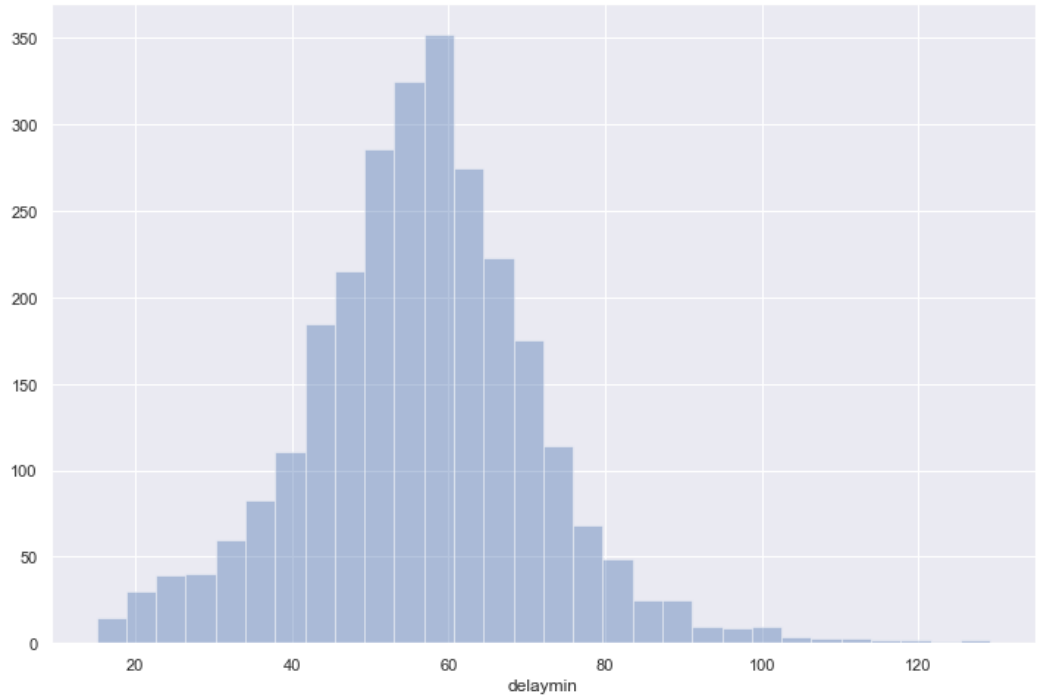
In [14]:
```
sns.set()
sns.pairplot(data_train, size = 2.5)
plt.show()
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/seaborn/axisgrid.py:2065: UserWarning: The `size` parameter has been renamed to `height`; pleaes update your code.
  warnings.warn(msg, UserWarning)
```



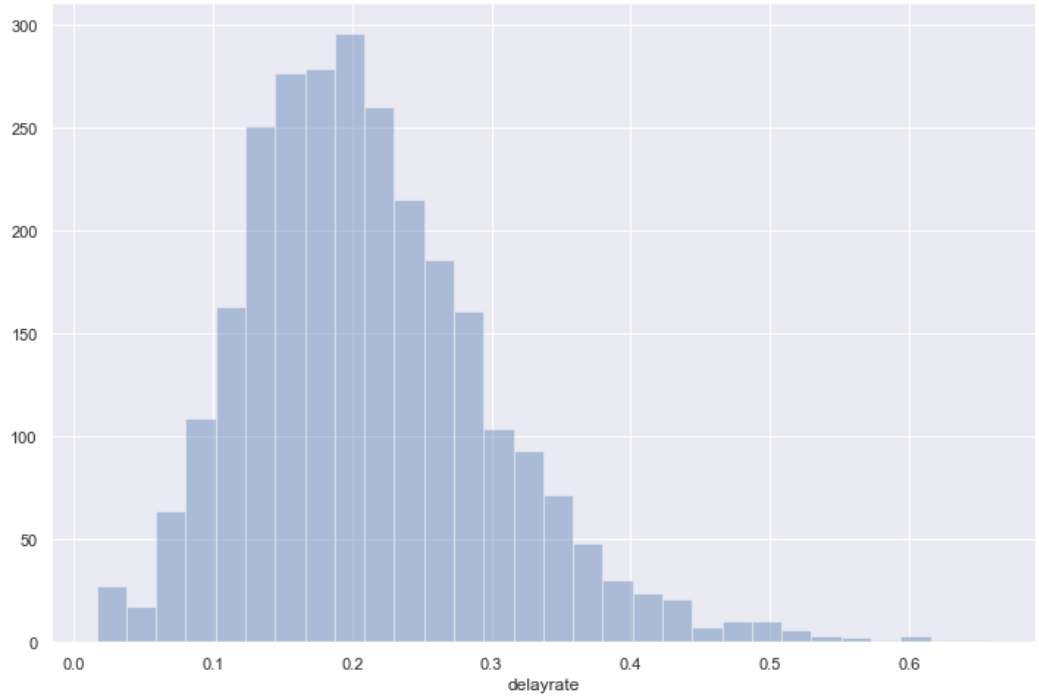
In [15]:
```
fig, ax = plt.subplots(figsize=(12, 8))
sns.distplot(data_train['delaymin'],bins=30,kde=False,ax=ax)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1231c8d68>



In [16]:
```
fig, ax = plt.subplots(figsize=(12, 8))
sns.distplot(data_train['delayrate'],bins=30,kde=False,ax=ax)
```

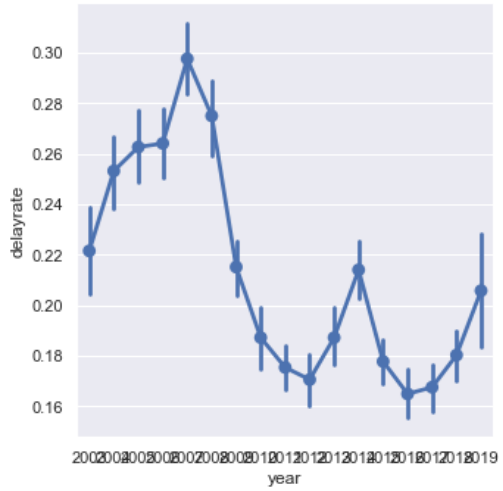
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x12389aef0>



In [17]:
```
sns.factorplot(x='year', y='delayrate', data=data_train)
```

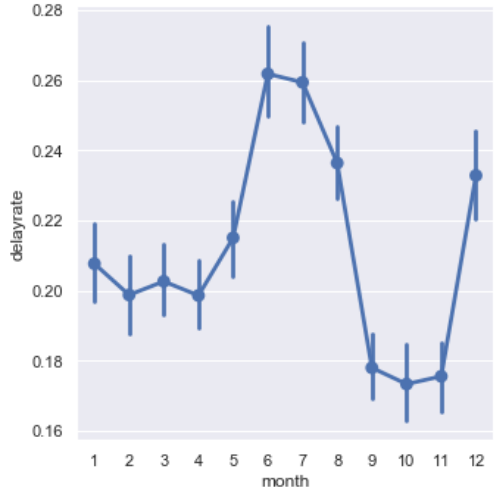
```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/seaborn/categorical.py:3666: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
```

Out[17]: <seaborn.axisgrid.FacetGrid at 0x123afbb00>



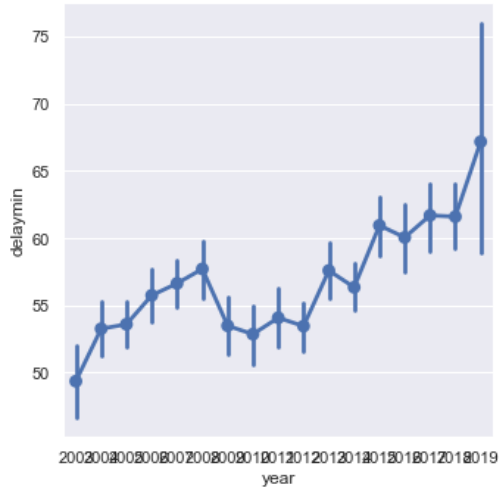
In [18]:
```
sns.factorplot(x='month', y='delayrate', data=data_train)
```

Out[18]: <seaborn.axisgrid.FacetGrid at 0x123b48860>



In [19]:
```
sns.factorplot(x='year', y='delaymin', data=data_train)
```

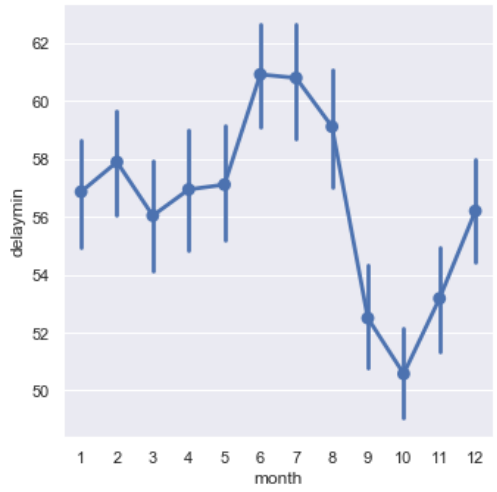
Out[19]: <seaborn.axisgrid.FacetGrid at 0x123c29470>



In [20]:
```
sns.factorplot(x='month', y='delaymin', data=data_train)
```

In [20]: sns.factorplot('month', 'delaymin', data=data_train)

Out[20]: <seaborn.axisgrid.FacetGrid at 0x1232396a0>



In [21]:
```
dummies_airport = pd.get_dummies(data_train['airport'], prefix= 'airport')
dummies_carrier = pd.get_dummies(data_train['carrier'], prefix= 'carrier')

data_train_class = pd.concat([data_train, dummies_airport, dummies_carrier], axis=1)
data_train_class.drop(['carrier','airport'], axis=1, inplace=True)
```

In [21]:
```
data_train_class.head()
```

Out[21]:

	year	month	delayrate	delaymin	airport_ATL	airport_JFK	airport_MIA	airport_ORD	carrier_AA	carrier_AS
2	2003	6	0.247340	44.698925	1	0	0	0	1	0
35	2003	6	0.184830	50.872642	0	1	0	0	1	0
48	2003	6	0.231731	49.762102	0	0	1	0	1	0
56	2003	6	0.171605	55.135218	0	0	0	1	1	0
111	2003	6	0.300000	40.000000	0	0	1	0	0	1

In [22]:
```
import sklearn.preprocessing as preprocessing

scaler = preprocessing.StandardScaler()

data_train_class['year'] = scaler.fit_transform(data_train_class['year'].values.reshape(-1,
1))
data_train_class['month'] = scaler.fit_transform(data_train_class['month'].values.reshape(-1
, 1))
```

In [23]:
```
data_train_class.head(10)
```

Out[23]:

	year	month	delayrate	delaymin	airport_ATL	airport_JFK	airport_MIA	airport_ORD	carrier_AA	car
2	-1.743289	-0.151895	0.247340	44.698925	1	0	0	0	1	0
35	-1.743289	-0.151895	0.184830	50.872642	0	1	0	0	1	0
48	-1.743289	-0.151895	0.231731	49.762102	0	0	1	0	1	0
56	-1.743289	-0.151895	0.171605	55.135218	0	0	0	1	1	0
111	-1.743289	-0.151895	0.300000	40.000000	0	0	1	0	0	1
115	-1.743289	-0.151895	0.100000	65.333333	0	0	0	1	0	1
324	-1.743289	-0.151895	0.169686	42.808271	1	0	0	0	0	0
366	-1.743289	-0.151895	0.158084	37.833333	0	1	0	0	0	0
377	-1.743289	-0.151895	0.240000	33.069444	0	0	1	0	0	0
389	-1.743289	-0.151895	0.271429	45.075188	0	0	0	1	0	0

In [24]:
```
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn import linear_model, svm, gaussian_process
from sklearn.ensemble import RandomForestRegressor
```

```
#train_X = data_train_class.filter(regex='year|month|airport_.*|carrier_.*')
train_X = data_train_class.filter(regex='year|month')
train_X = train_X.values

train_Y = data_train_class.filter(regex='delaymin')
train_Y = train_Y.values

X = train_X
Y = train_Y

X_train,X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.33, random_state=42)
```

In [25]:
```
Y_train
```

Out[25]:
```
array([[74.98605578],
       [56.62113402],
       [45.35135135],
       ...,
       [67.984     ],
       [59.28571429],
       [65.59813084]])
```

In [26]:
```
clfs = {
        'svm':svm.SVR(),
        'RandomForestRegressor':RandomForestRegressor(n_estimators=400),
        'BayesianRidge':linear_model.BayesianRidge(),
        'LinearRegression':linear_model.LinearRegression()
       }
for clf in clfs:
    try:
        clfs[clf].fit(X_train, Y_train)
        Y_pred = clfs[clf].predict(X_test)
        print(clf + " cost:" + str(np.sum(Y_pred-Y_test)/len(Y_pred)) )
    except Exception as e:
        print(clf + " Error:")
        print(str(e))
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/sklearn/utils/validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/sklearn/svm/base.py:193: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/ipykernel_launcher.py:9: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  if __name__ == '__main__':
svm cost:623.60029790653
RandomForestRegressor cost:522.4253713495868
BayesianRidge cost:309.46381681478266
LinearRegression cost:0.3390497076520312
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/sklearn/utils/validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```