In [1]:

```
# The normal imports
import numpy as np
from numpy.random import randn
import pandas as pd

# Import the stats library from numpy
from scipy import stats

# These are the plotting modules adn libraries we'll use:
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

# Command so that plots appear in the iPython Notebook
%matplotlib inline
```

In [13]:

```
# Again seaborn comes with a great dataset to play and learn with
flight_dframe = sns.load_dataset('flights')
```

In [14]:

```
#Preview
flight_dframe.head()
```

Out[14]:

	year	month	passengers
0	1949	January	112
1	1949	February	118
2	1949	March	132
3	1949	April	129
4	1949	May	121

In [15]:

```
# Let's pivot this dataframe do its easier to manage
flight_dframe = flight_dframe.pivot("month","year","passengers")

#Show
flight_dframe
```

Out[15]:

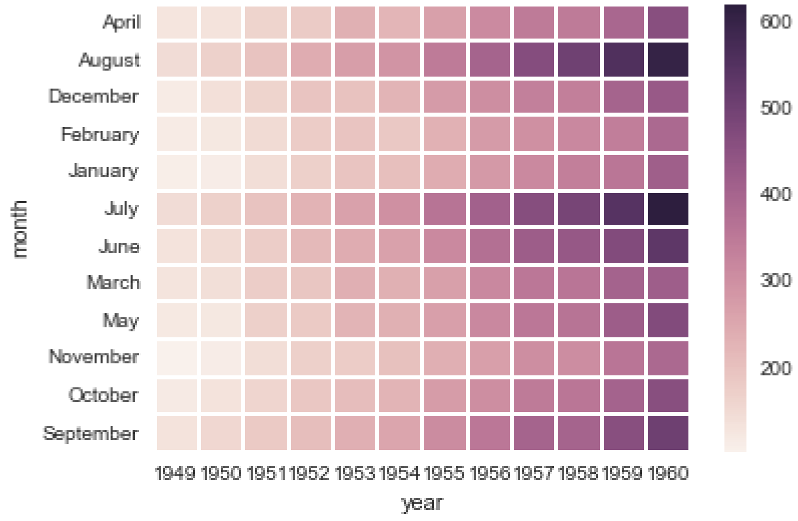
year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
April	129	135	163	181	235	227	269	313	348	348	396	461
August	148	170	199	242	272	293	347	405	467	505	559	606
December	118	140	166	194	201	229	278	306	336	337	405	432
February	118	126	150	180	196	188	233	277	301	318	342	391
January	112	115	145	171	196	204	242	284	315	340	360	417
July	148	170	199	230	264	302	364	413	465	491	548	622
June	135	149	178	218	243	264	315	374	422	435	472	535
March	132	141	178	193	236	235	267	317	356	362	406	419
May	121	125	172	183	229	234	270	318	355	363	420	472
November	104	114	146	172	180	203	237	271	305	310	362	390
October	119	133	162	191	211	229	274	306	347	359	407	461
September	136	158	184	209	237	259	312	355	404	404	463	508

In [11]:

```
# This dataset is now in a clear format to be dispalyed as a heatmap
sns.heatmap(flight_dframe)
```

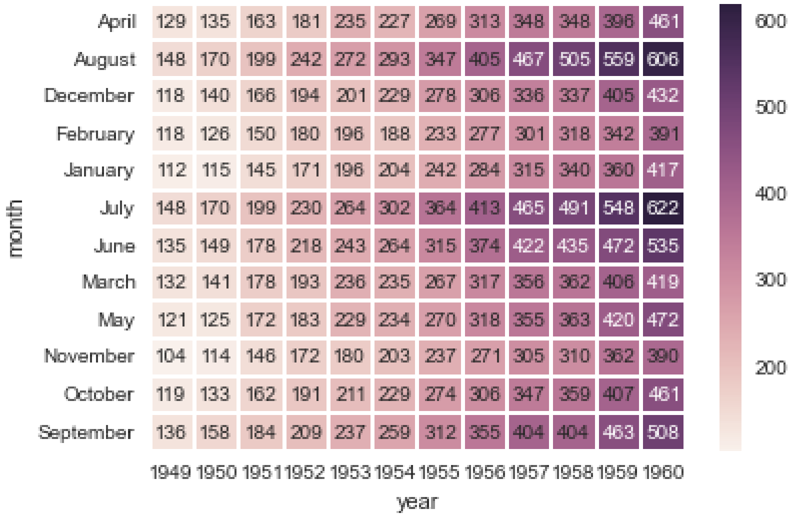
Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x18ada518>
```



```
In [17]: # We also have the option to annotate each cell
         sns.heatmap(flight_dframe,annot=True,fmt='d')
```

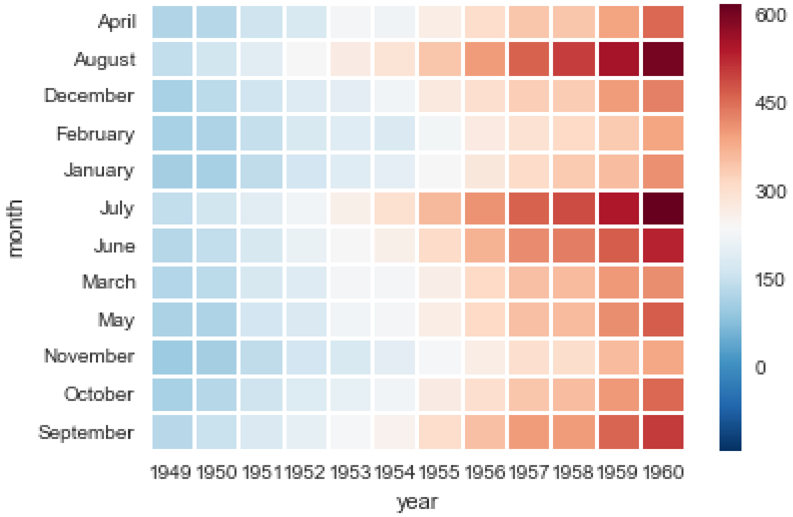
```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x1b51ae80>
```



```
In [18]: # seaborn will automatically try to pick the best color scheme for your dataset,
```

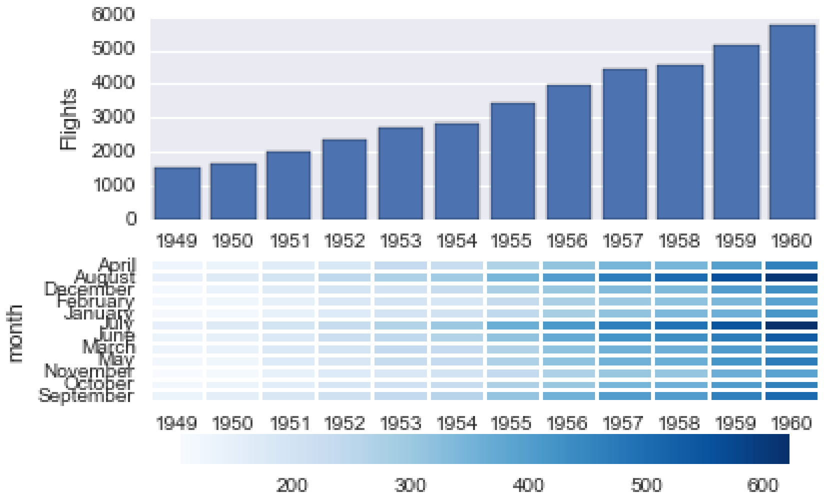
```
In [19]: # We can choose our own 'center' for our colormap
         sns.heatmap(flight_dframe,center=flight_dframe.loc['January',1955])
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1bca9a90>
```

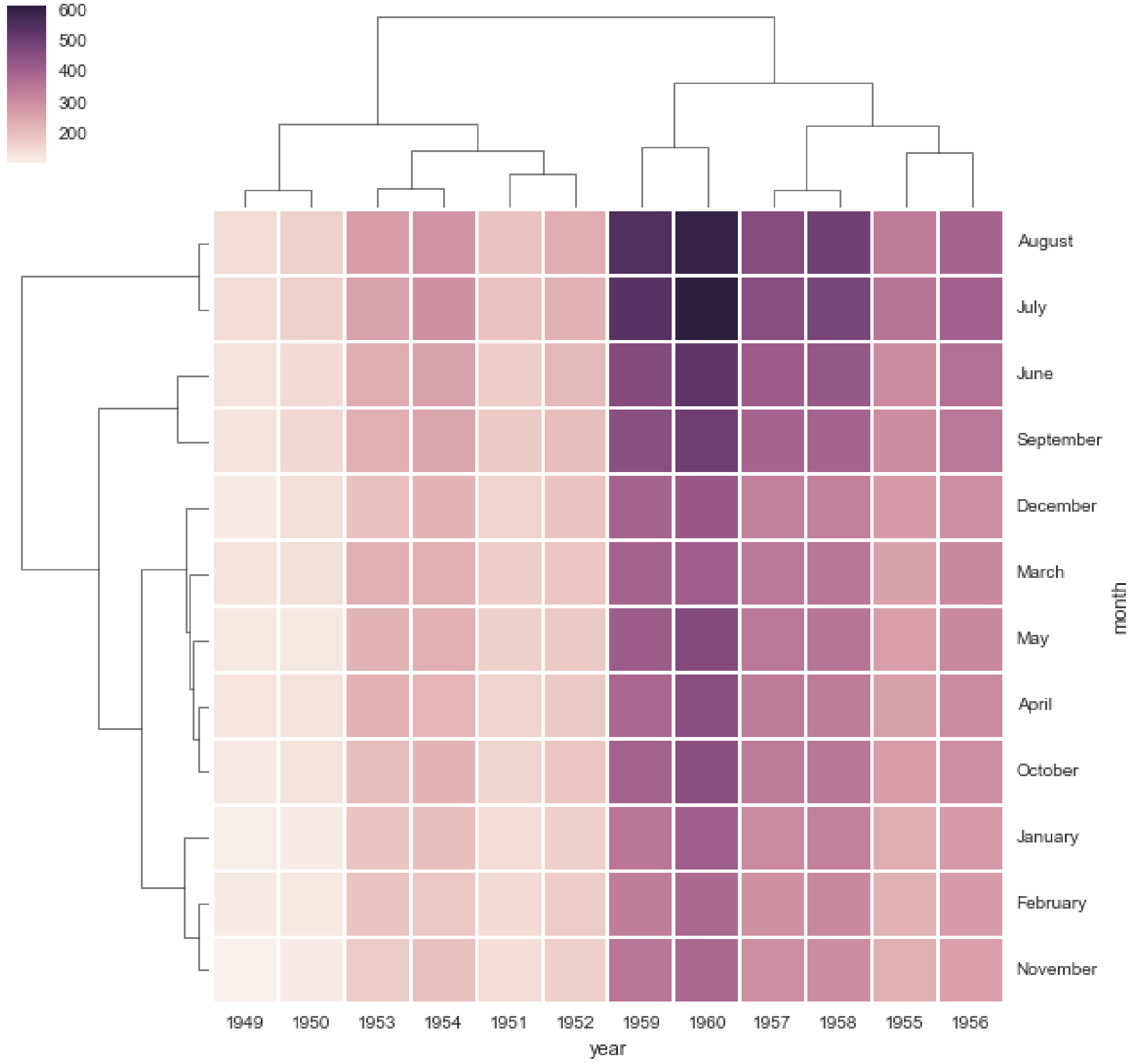


```
In [177]: # heatmap() can be used on an axes for a subplot to create more informative figur
          f, (axis1,axis2) = plt.subplots(2,1)

          yearly_flights = flight_dframe.sum()

          # Since yearly_flights is a weird format, we'll have to grab the values we want w

          years = pd.Series(yearly_flights.index.values)
          years = pd.DataFrame(years)

          flights = pd.Series(yearly_flights.values)
          flights = pd.DataFrame(flights)

          # Make the dframe and name columns
          year_dframe = pd.concat((years,flights),axis=1)
          year_dframe.columns = ['Year','Flights']



          # Create the bar plot on top
          sns.barplot('Year',y='Flights',data=year_dframe, ax = axis1)

          # Create the heatmap on bottom
          sns.heatmap(flight_dframe,cmap='Blues',ax=axis2,cbar_kws={"orientation": "horizon
```

```
Out[177]: <matplotlib.axes._subplots.AxesSubplot at 0x28bfcb38>
```

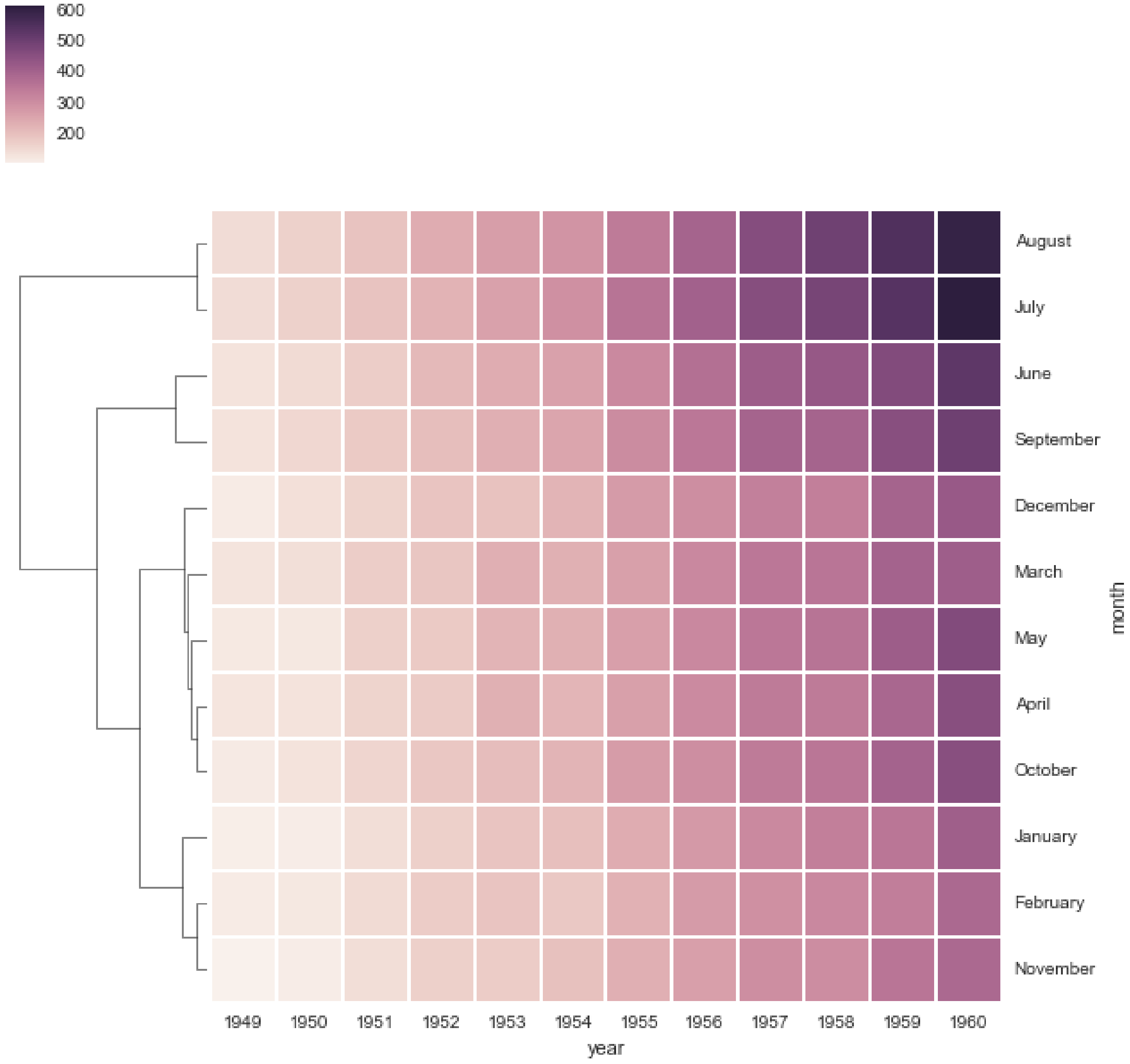


```
In [74]: # Finally we'll learn about using a clustermap

         # Clustermap will reformat the heatmap so similar rows are next to each other
         sns.clustermap(flight_dframe)

Out[74]: <seaborn.matrix.ClusterGrid at 0x20af8438>
```

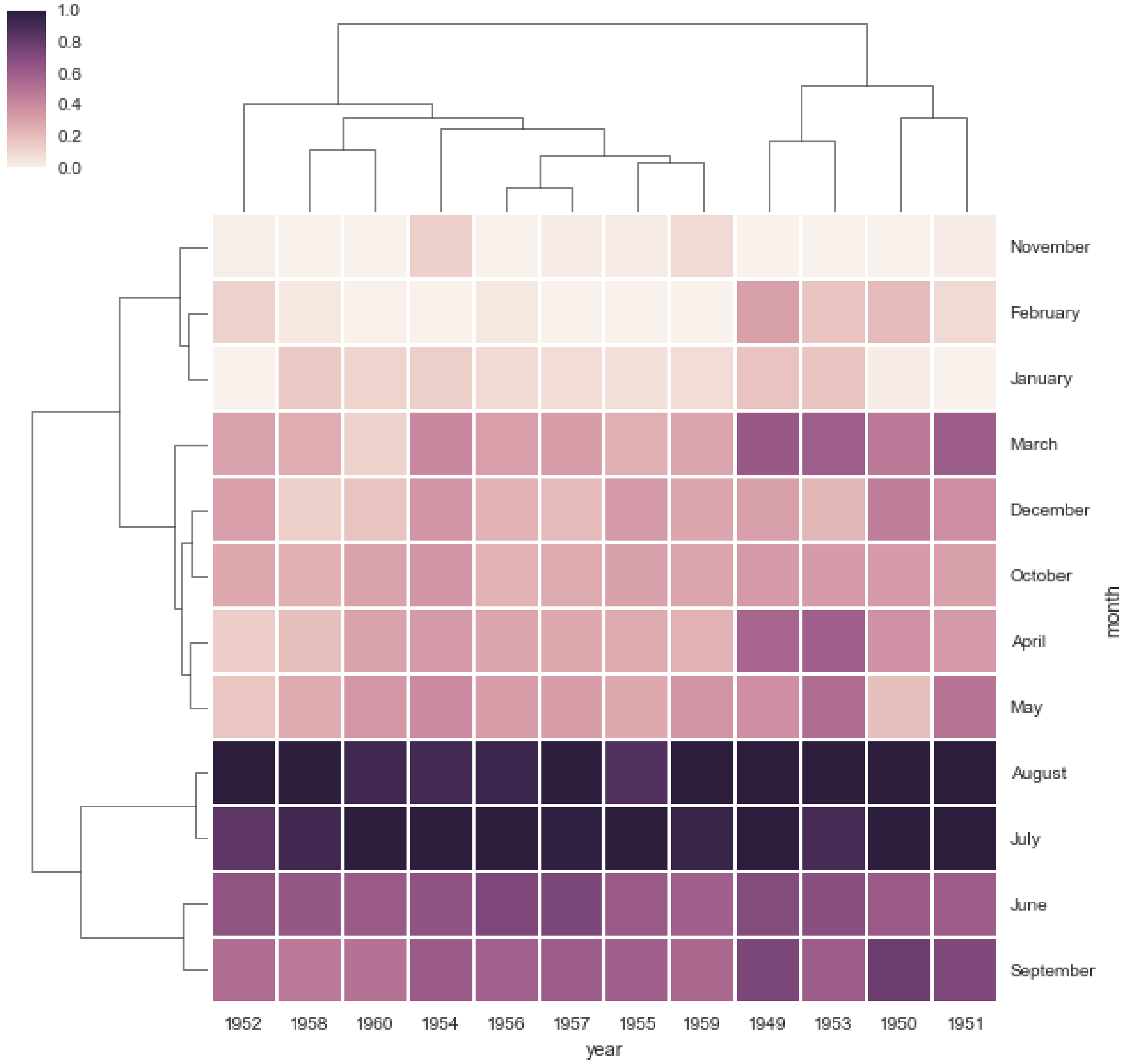


```
In [75]: # Let's uncluster the columns
         sns.clustermap(flight_dframe,col_cluster=False)
```

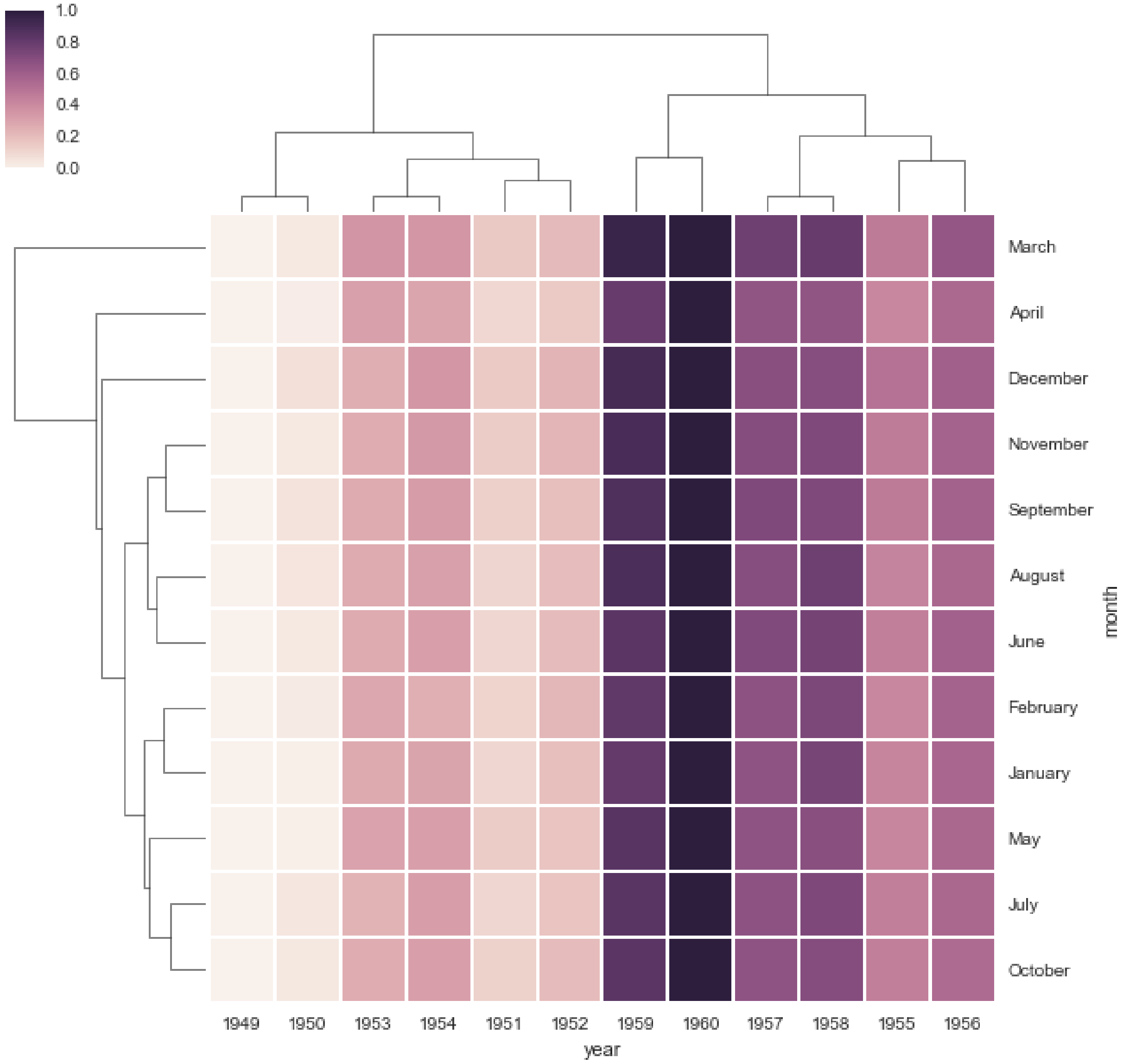
```
Out[75]: <seaborn.matrix.ClusterGrid at 0x220f1cc0>
```



```
In [76]: # Since the number of flights increase every year, we should set a standard scale
         sns.clustermap(flight_dframe,standard_scale=1) # standardize by columns (year)

Out[76]: <seaborn.matrix.ClusterGrid at 0x227967f0>
```

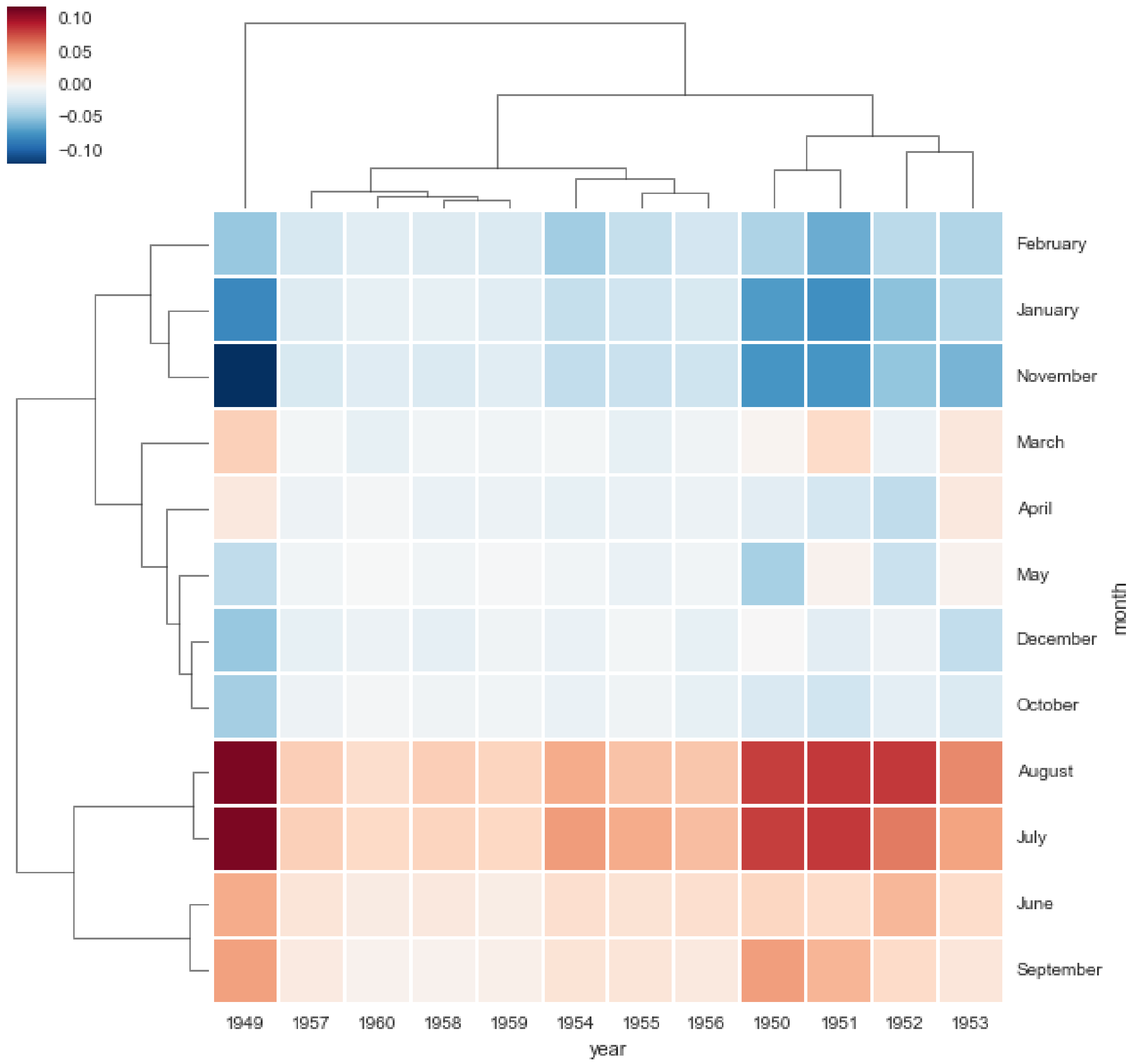


```
In [77]: # Or scale the rows
         sns.clustermap(flight_dframe,standard_scale=0)

Out[77]: <seaborn.matrix.ClusterGrid at 0x2095c2e8>
```



```
In [78]: # Finally we can also normalize the rows by their Z-score.
         # This subtracts the mean and devides by the STD of each column, then teh rows ha
         sns.clustermap(flight_dframe,z_score=1)
```

```
Out[78]: <seaborn.matrix.ClusterGrid at 0x20314160>
```



```
In [79]: # Above we can see which values are greater than the mean and which are below ver
```

```
In [ ]: # CONGRATULATIONS!! We've developed quite a toolbox to hammer out some great data

        # Up next: Projects to apply what we've learned to real datasets!
```