Seaborn tutorial

importing necessary libraries

In [69]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

you can import the datasets from my github datasets repository

In [2]:

```
df=pd.read_csv("tips.csv")
sns.set(style="darkgrid",font="segoe print",font_scale=1.2)
df.head()
```

Out[2]:

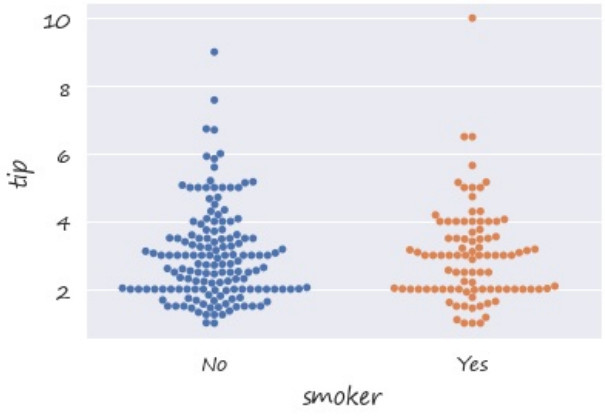
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [3]:

```
sns.swarmplot(x="smoker",y="tip",data=df)
```

Out[3]:

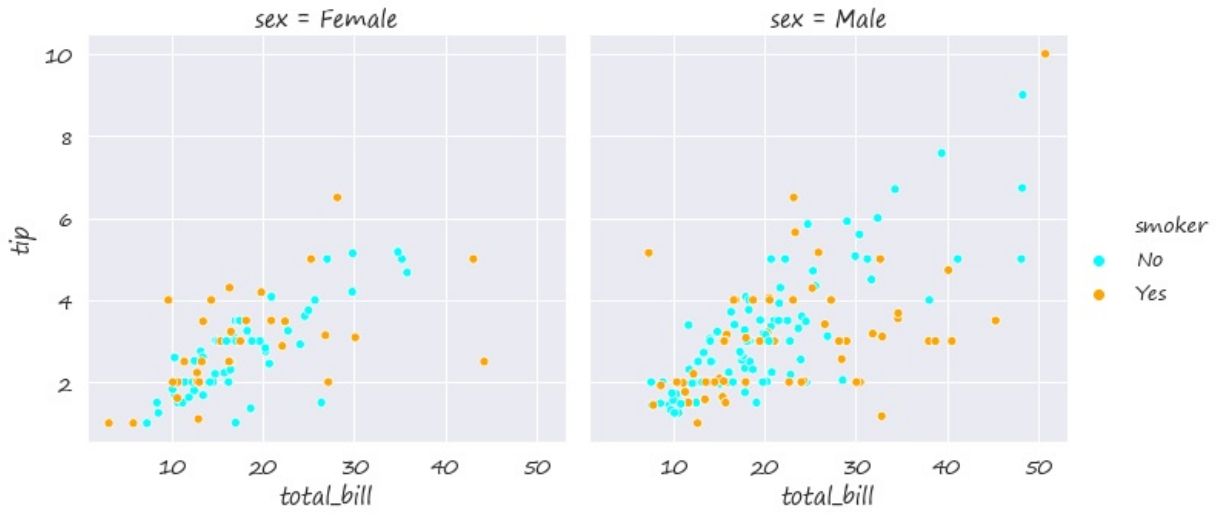
```
<AxesSubplot:xlabel='smoker', ylabel='tip'>
```



relplot numerical analysis for multivariate with scatter plot

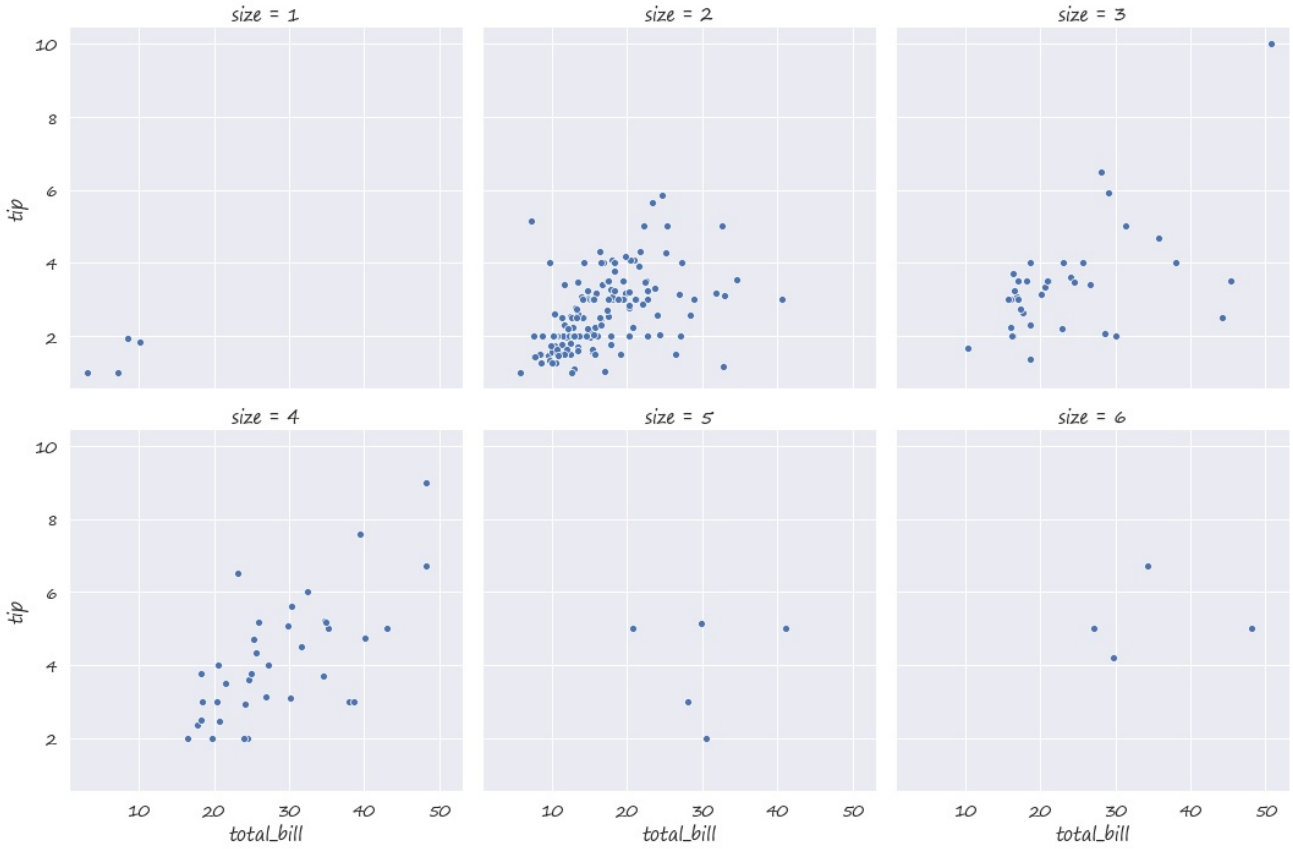
In [3]:

```
sns.relplot(x="total_bill",y="tip",data=df,hue="smoker",palette=["aqua","orange"],col="sex",\
            sizes=[20,200])
plt.show()
```



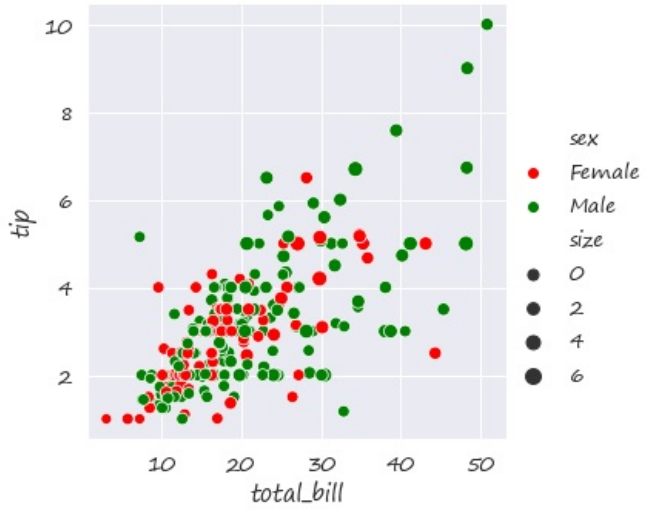
In [4]:

```
sns.relplot(x="total_bill",y="tip",data=df,palette=["aqua"],col="size",col_wrap=3)
plt.show()#max people of size 2
```



In [6]:

```
sns.relplot(x="total_bill",y="tip",data=df,hue="sex",palette=["red","green"],size="size",\
            sizes=(50,100))
plt.show()
```



In [7]:

```
sns.relplot(x="total_bill",y="tip",data=df,hue="time",palette=["b","g"],col="day",col_wrap=2,\
            size="size")
plt.show() #these scatter plots shows that we have potential customers at thursday lunch time and
           #sun and sat night
```

In [9]:

```
print(dir(sns.FacetGrid))
```

```
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__
getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__modul
e__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '
__str__', '__subclasshook__', '__weakref__', '_bottom_axes', '_clean_axis', '_facet_color', '_facet_
plot', '_finalize_grid', '_get_palette', '_inner_axes', '_left_axes', '_legend_out', '_margin_titles
', '_not_bottom_axes', '_not_left_axes', '_update_legend_data', 'add_legend', 'ax', 'despine', 'face
t_axis', 'facet_data', 'map', 'map_dataframe', 'savefig', 'set', 'set_axis_labels', 'set_titles', 's
et_xlabels', 'set_xticklabels', 'set_ylabels', 'set_yticklabels']
```

In [10]:

```
df.groupby("size").count()
```

Out[10]:

size	total_bill	tip	sex	smoker	day	time
1	4	4	4	4	4	4
2	156	156	156	156	156	156
3	38	38	38	38	38	38
4	37	37	37	37	37	37
5	5	5	5	5	5	5
6	4	4	4	4	4	4

In [20]:

```
#df["smoker"].value_counts()
df.groupby("smoker").count()["sex"]
```

Out[20]:

```
smoker
No     151
Yes     93
Name: sex, dtype: int64
```

In [22]:

```
import random as rn
df1=pd.DataFrame(dict(value=[rn.uniform(-1.5,1.5) for i in range(300)],time=np.arange(300)))

df1.head()
```

Out[22]:

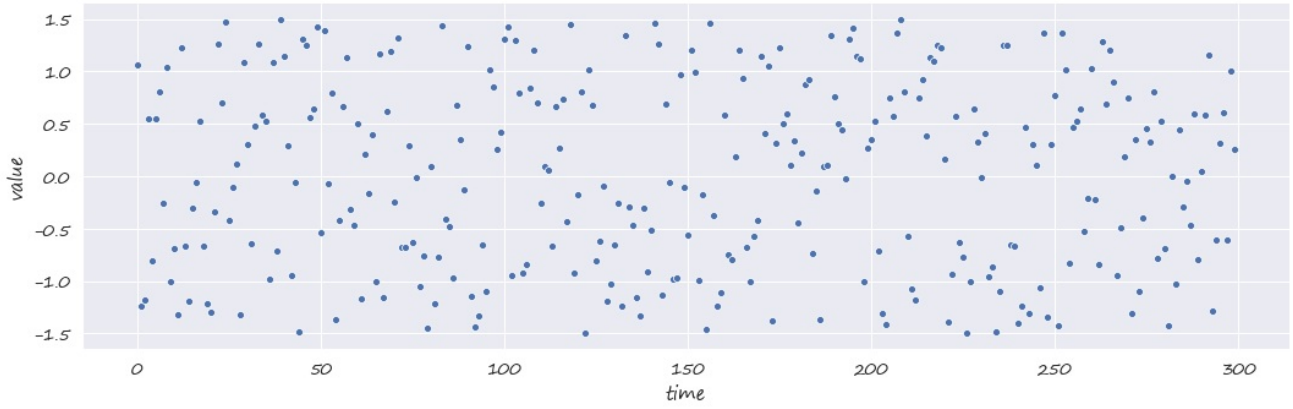
	value	time
0	0.811174	0
1	0.109309	1
2	1.099652	2
3	-1.149550	3
4	-1.059980	4

In [95]:

```
sns.relplot(x="time",y="value",data=df1,aspect=3,kind="scatter")
```

Out[95]:

```
<seaborn.axisgrid.FacetGrid at 0x1af0d542040>
```



In [23]:

```
from sklearn.cluster import KMeans
kms=KMeans(n_clusters=9,algorithm="auto")
kms.fit(df1)
```

Out[23]:

```
KMeans(n_clusters=9)
```

using ML kmeans algorithm to group the sample data in clusters of 9

In [27]:

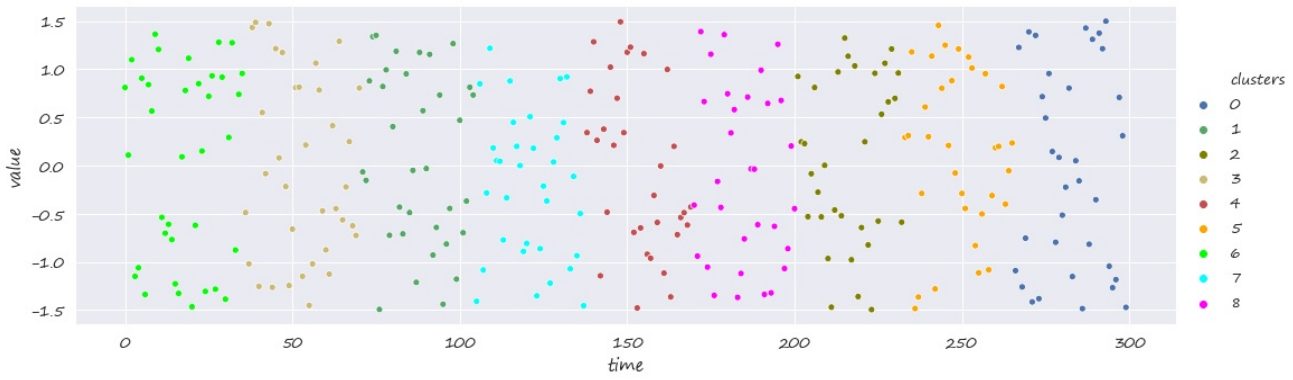
```
ypred=kms.predict(df1)
df1["clusters"]=ypred

df1.head()
```

Out[27]:

	value	time	clusters
0	0.811174	0	6
1	0.109309	1	6
2	1.099652	2	6
3	-1.149550	3	6
4	-1.059980	4	6

In [28]:

```
sns.relplot(x="time",y="value",hue="clusters",data=df1,aspect=3,\
            palette=["b","g","olive","y","r","orange","lime","aqua","magenta"])
plt.show()
```



In [29]:

```
df=pd.read_csv("fmri.csv")  #you can get it from my github datasets repository
df.head()
```

Out[29]:

	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970

In [31]:

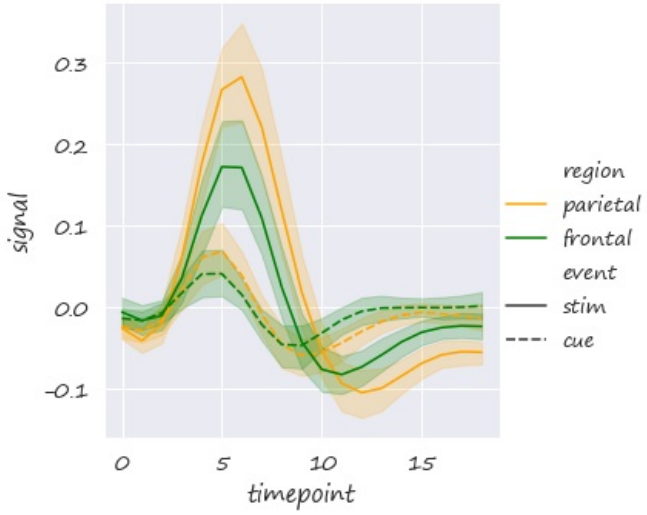
```
df.isnull().sum()
```

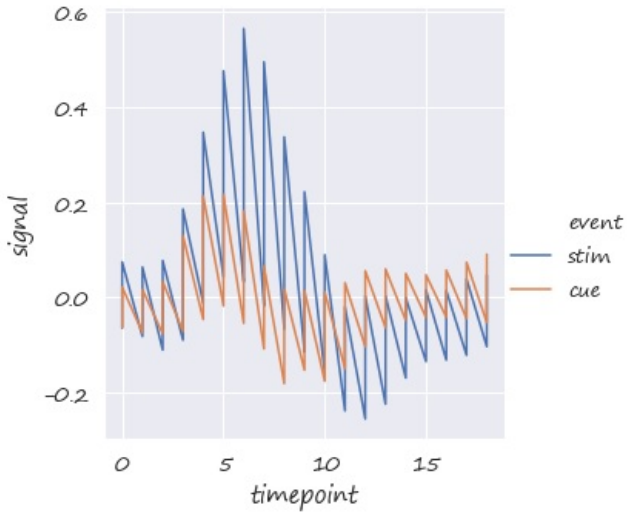
Out[31]:

```
subject      0
timepoint    0
event        0
region       0
signal       0
dtype: int64
```

In [55]:

```
sns.relplot(x="timepoint",y="signal",hue="region",style="event",palette=["orange","green"],\
            data=df,kind="line")
sns.relplot(x="timepoint",y="signal",hue="event",data=df,kind="line",estimator=None,markers=True)
plt.show()
```





In [37]:

```
df.dtypes
```

Out[37]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

In [38]:

```
df.head()
```

Out[38]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Correlations and Heatmap

In [45]:

```
correlation=df.corr()
correlation
```

Out[45]:

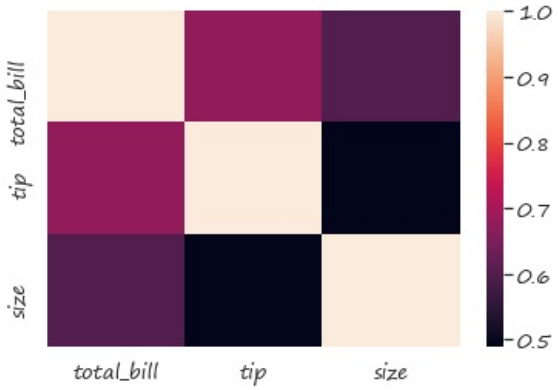
	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

In [46]:

```
sns.heatmap(correlation)
```

Out[46]:

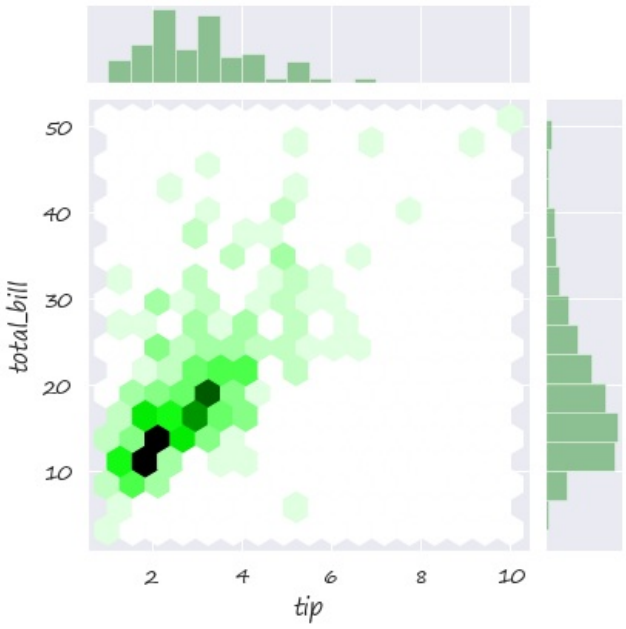
```
<AxesSubplot:>
```



Jointplot- awesome things, tells about the distribution of data

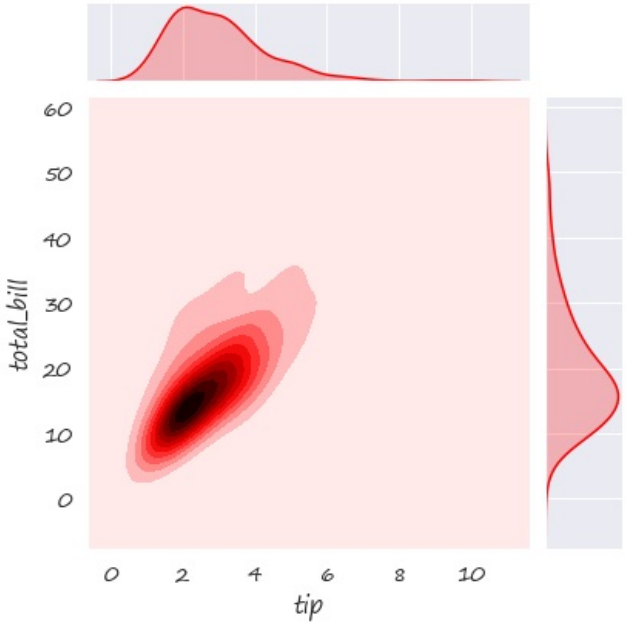
In [63]:

```
sns.jointplot(x="tip",y="total_bill",data=df,kind="hex",color="green")
plt.show()
```



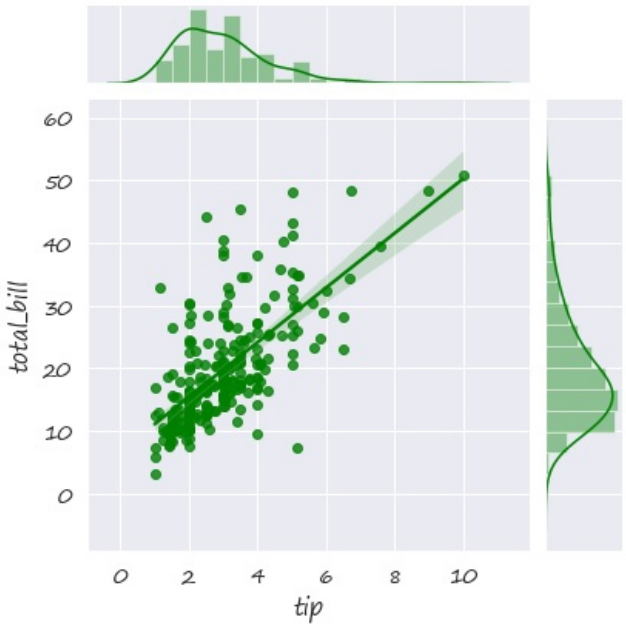
In [76]:

```
sns.jointplot(x="tip",y="total_bill",data=df,kind="kde",color="red")
plt.show()
```



In [80]:

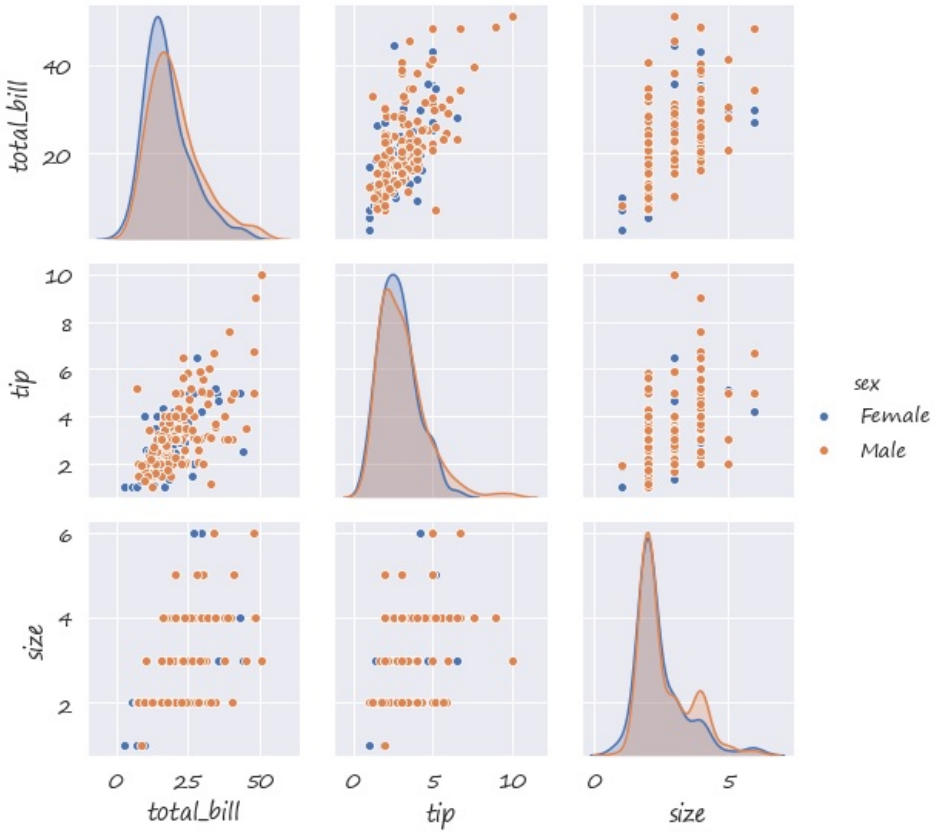
```
sns.jointplot(x="tip",y="total_bill",data=df,kind="reg",color="green")
plt.show()
```



Pairplots cominations for each of the parameters

In [83]:

```
sns.pairplot(df,hue="sex")
plt.show()
```

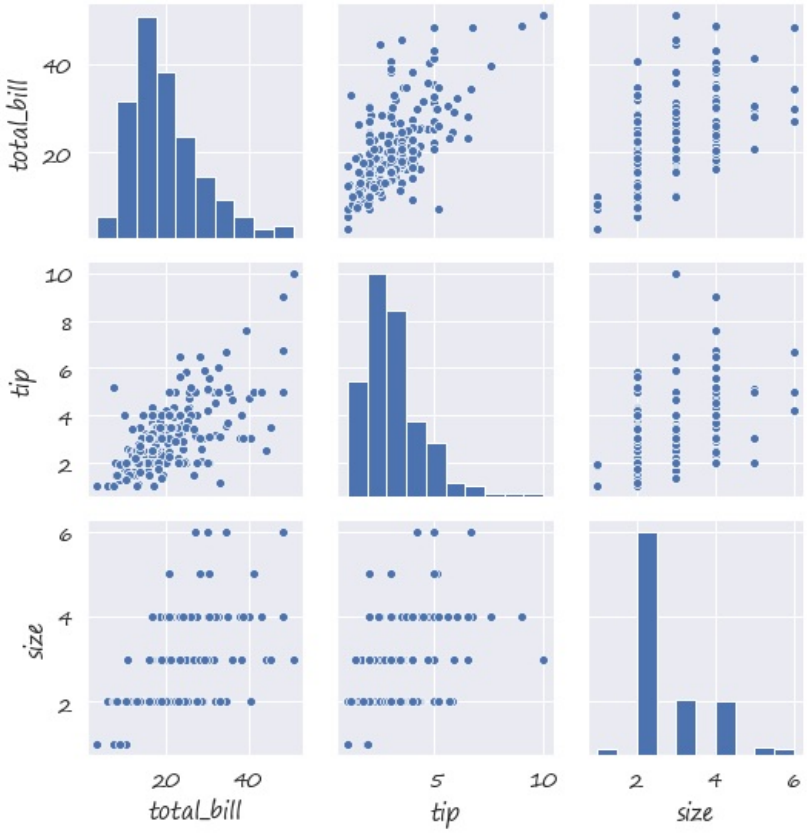


In [88]:

```
sns.pairplot(df,palette=["green","red"])
```

Out[88]:

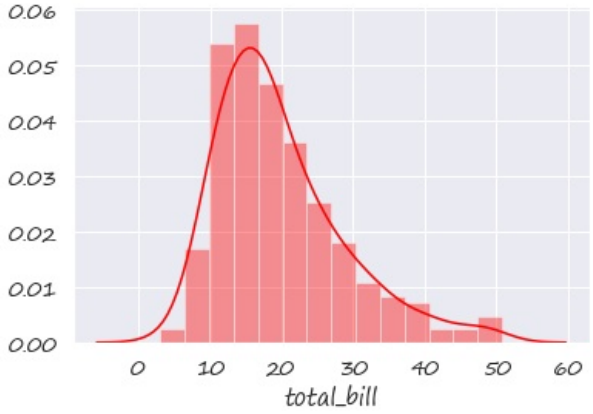
```
<seaborn.axisgrid.PairGrid at 0x19d61322910>
```



Distplots for analyzing the dense part of the samples

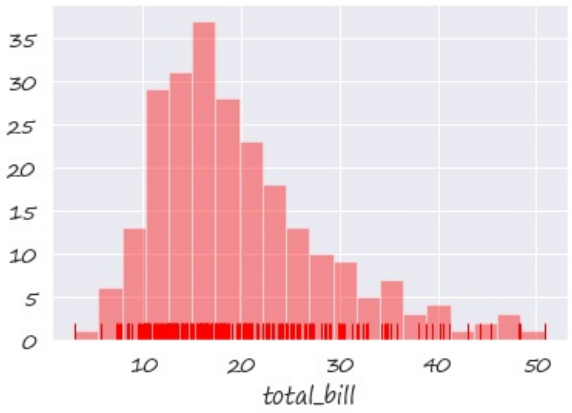
In [93]:

```
sns.distplot(df["total_bill"],color="red")
plt.show() #the y axis is percentage
```



In [98]:

```
sns.distplot(df["total_bill"],kde=False,color="red",bins=20,rug=True)
plt.show()  #Making kde as false will show the yaxis as the frequency
```



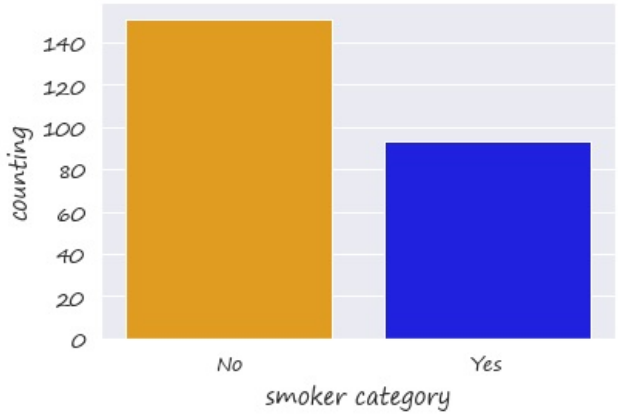
Categorical data

In [136]:

```
ax=sns.countplot("smoker",data=df,palette=["orange","blue"])
ax.set(xlabel="smoker category",ylabel="counting")
```

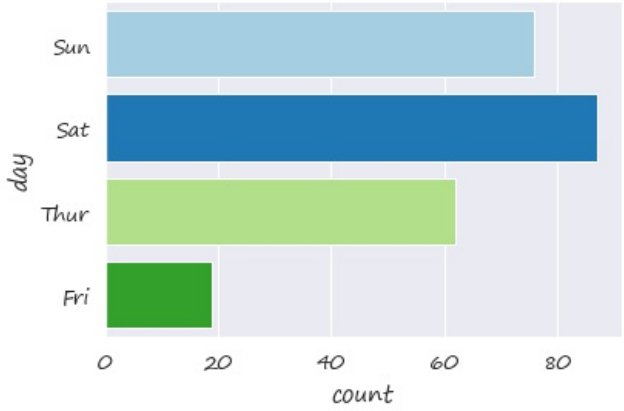
Out[136]:

```
[Text(0.5, 0, 'smoker category'), Text(0, 0.5, 'counting')]
```



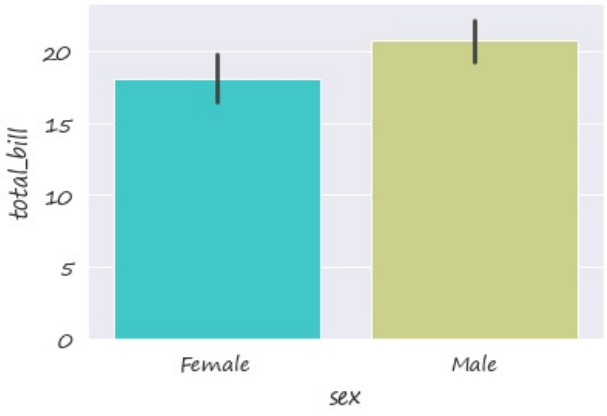
In [119]:

```
sns.color_palette("husl", 8)
sns.countplot(y="day",data=df,saturation=1.2)
plt.show()
```



In [140]:

```
sns.barplot(x="sex",y="total_bill",data=df,palette="rainbow")
#the max values of barplot gives the mean of each type
plt.show()
```



In [128]:

```
df.groupby("sex").mean()["total_bill"] #the bar plot can be verified from here
```

Out[128]:

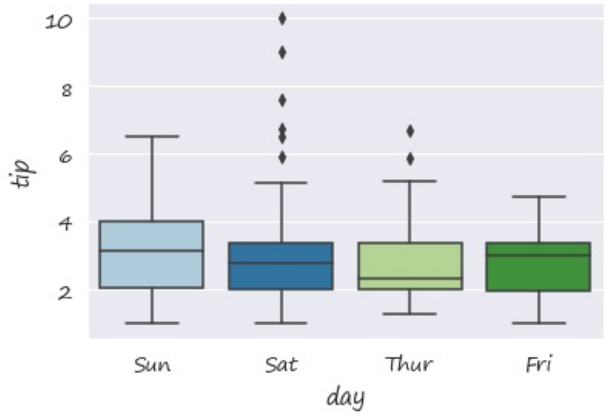
```
sex
Female    18.056897
Male      20.744076
Name: total_bill, dtype: float64
```

In [162]:

```
sns.boxplot(x="day",y="tip",data=df)
```

Out[162]:

```
<AxesSubplot:xlabel='day', ylabel='tip'>
```

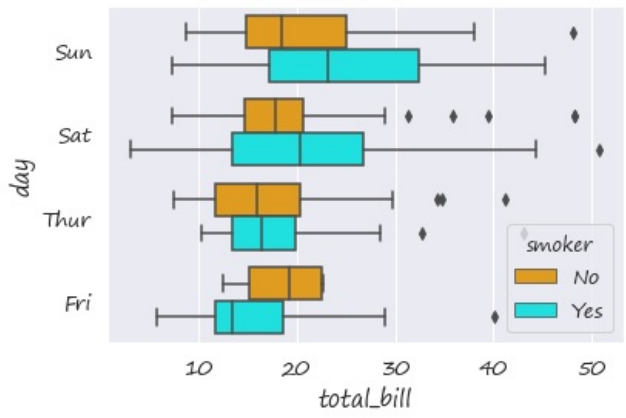


In [179]:

```
sns.boxplot(x="total_bill",y="day",data=df,hue="smoker",palette=["orange","aqua"])
```

Out[179]:

```
<AxesSubplot:xlabel='total_bill', ylabel='day'>
```

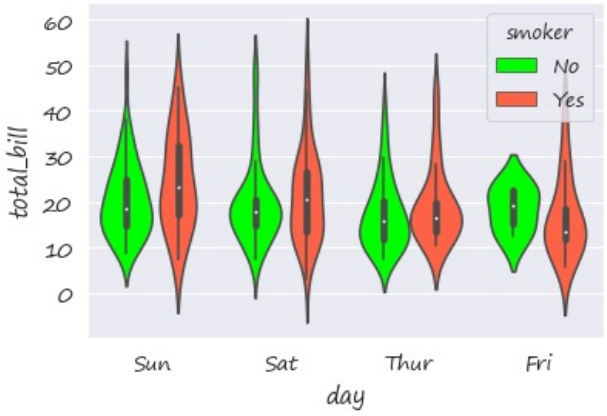


In [187]:

```
sns.violinplot(x="day",y="total_bill",hue="smoker",data=df,palette=["lime","tomato"],saturation=1.3)
```

Out[187]:

```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```

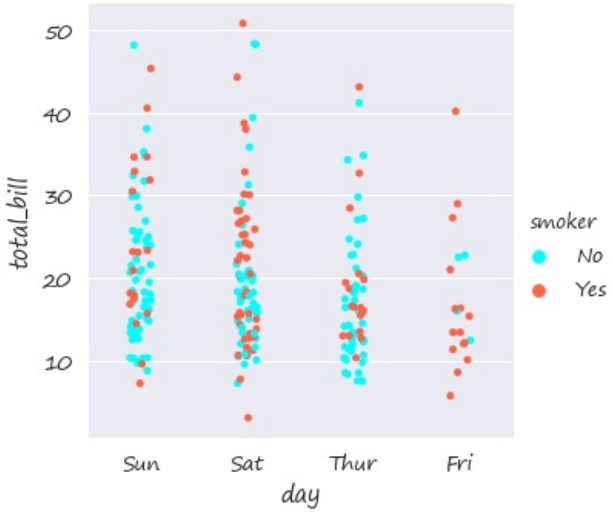


In [190]:

```
sns.catplot(x="day",y="total_bill",hue="smoker",data=df,palette=["aqua","tomato"])
```

Out[190]:

```
<seaborn.axisgrid.FacetGrid at 0x19d726c8e50>
```



In [70]:

```
df2=pd.read_excel("sales.xlsx")  #you can  get the sales dataset from my github datasets repository
df2.head()
```

Out[70]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Customer Name	Province	Region	Custome Segmen
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Muhammed MacIntyre	Nunavut	Nunavut	Sma Busines
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Barry French	Nunavut	Nunavut	Consume
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Barry French	Nunavut	Nunavut	Consume
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Clay Rozendal	Nunavut	Nunavut	Corporat
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Carlos Soltero	Nunavut	Nunavut	Consume

5 rows × 21 columns

Now we will use a real life dataset for the seaborn visualisation practice

In [71]:

```
df2.dtypes
```

Out[71]:

```
Row ID                          int64
Order ID                        int64
Order Date             datetime64[ns]
Order Priority                 object
Order Quantity                  int64
Sales                         float64
Discount                      float64
Ship Mode                      object
Profit                        float64
Unit Price                    float64
Shipping Cost                 float64
Customer Name                  object
Province                       object
Region                         object
Customer Segment               object
Product Category               object
Product Sub-Category           object
Product Name                   object
Product Container              object
Product Base Margin           float64
Ship Date              datetime64[ns]
dtype: object
```

In [72]:

```
df2.isnull().sum()
#we can see there are 63 Nan values in Product Base Margin
```

Out[72]:

```
Row ID                     0
Order ID                   0
Order Date                 0
Order Priority             0
Order Quantity             0
Sales                      0
Discount                   0
Ship Mode                  0
Profit                     0
Unit Price                 0
Shipping Cost              0
Customer Name              0
Province                   0
Region                     0
Customer Segment           0
Product Category           0
Product Sub-Category       0
Product Name               0
Product Container          0
Product Base Margin       63
Ship Date                  0
dtype: int64
```

In [73]:

```
df2[df2.isna().any(axis=1)] #63 rows of Product Base Margin have Nan values
```

Out[73]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Customer Name	Province	Region	Custom Segme
7	98	613	2011-06-17	High	22	905.080	0.09	Regular Air	127.70	42.76	...	Carl Jackson	Nunavut	Nunavut	Corpora
8	103	643	2011-03-24	High	21	2781.820	0.07	Express Air	-695.26	138.14	...	Monica Federle	Nunavut	Nunavut	Corpora
71	988	7110	2011-08-07	Low	22	6396.200	0.02	Regular Air	1902.24	276.20	...	Grant Carroll	Nunavut	Nunavut	Corpora
213	3419	24387	2011-08-10	Critical	17	6048.180	0.04	Delivery Truck	1418.36	349.45	...	Bryan Mills	Northwest Territories	Northwest Territories	Sm Busine
274	4306	30658	2011-03-26	Medium	44	6040.220	0.05	Regular Air	-942.50	138.14	...	Charles McCrossin	Northwest Territories	Northwest Territories	Corpora
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7489	6015	42628	2011-05-05	Not Specified	4	1199.336	0.01	Delivery Truck	224.80	349.45	...	Ken Lonsdale	British Columbia	West	Consum
7508	7124	50822	2012-02-07	Medium	6	1635.290	0.10	Regular Air	7.92	276.20	...	Carol Darley	British Columbia	West	Sm Busine
7833	261	1824	2009-05-04	Critical	39	10656.260	0.06	Regular Air	3116.54	276.20	...	Becky Pak	Alberta	West	Corpora
8160	728	5222	2009-05-26	Not Specified	40	14451.750	0.01	Delivery Truck	4503.63	349.45	...	Tonja Turnell	Alberta	West	Consum
8351	7268	51872	2012-04-23	Medium	10	2193.930	0.09	Regular Air	41.83	238.40	...	Shui Tom	Alberta	West	Consum

63 rows × 21 columns

In [74]:

```
# we will fill it by median values
df2["Product Base Margin"].fillna(df2.median()["Product Base Margin"],inplace=True)
df2.isnull().sum() #Now there are no Nan values in Product Base Margin
```

```
<ipython-input-74-9014ee8de03a>:2: FutureWarning: DataFrame.mean and DataFrame.median with numeric_o
nly=None will include datetime64 and datetime64tz columns in a future version.
  df2["Product Base Margin"].fillna(df2.median()["Product Base Margin"],inplace=True)
```

Out[74]:

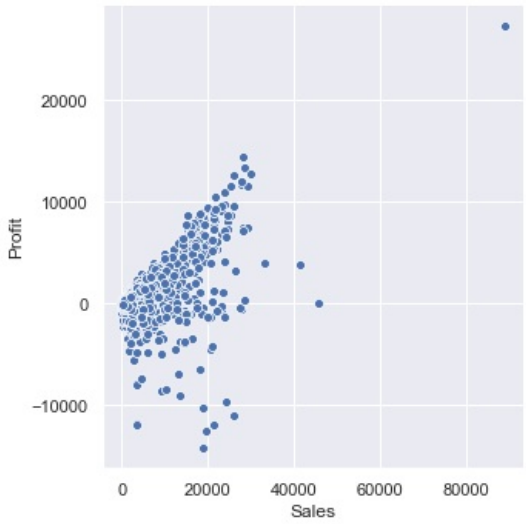
```
Row ID                  0
Order ID                0
Order Date              0
Order Priority          0
Order Quantity          0
Sales                   0
Discount                0
Ship Mode               0
Profit                  0
Unit Price              0
Shipping Cost           0
Customer Name           0
Province                0
Region                  0
Customer Segment        0
Product Category        0
Product Sub-Category    0
Product Name            0
Product Container       0
Product Base Margin     0
Ship Date               0
dtype: int64
```

In [75]:

```
sns.set(style="darkgrid") #darkgrid , whitegrid , dark , white , and ticks (different styles)
sns.relplot(x="Sales",y="Profit",data=df2)
```

Out[75]:

```
<seaborn.axisgrid.FacetGrid at 0x2c17f6b0970>
```

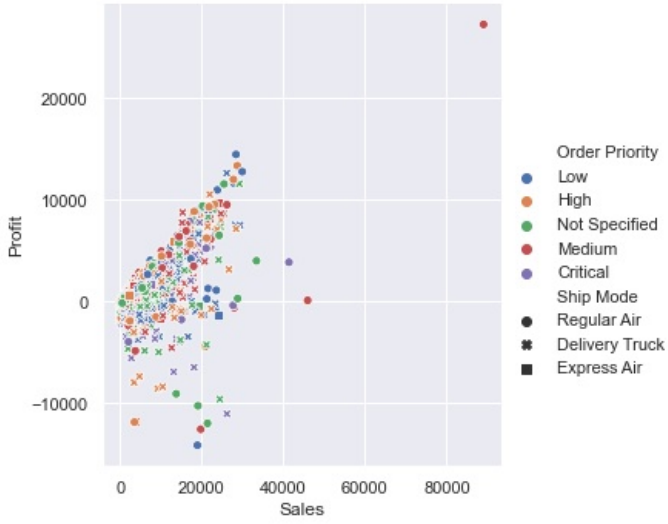


In [11]:

```
sns.relplot(x="Sales",y="Profit",hue="Order Priority",style="Ship Mode",data=df2)
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x2c155887610>
```

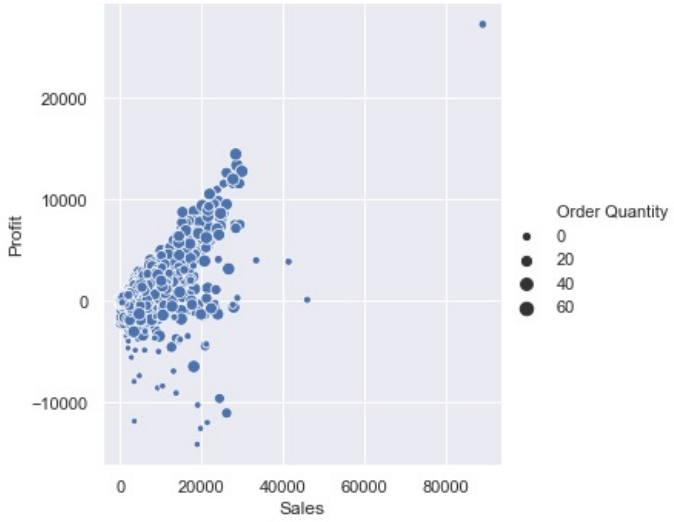


In [76]:

```
sns.relplot(x="Sales",y="Profit",size="Order Quantity",data=df2)# but to tune the sizes manually
```

Out[76]:

```
<seaborn.axisgrid.FacetGrid at 0x2c1786dd8e0>
```

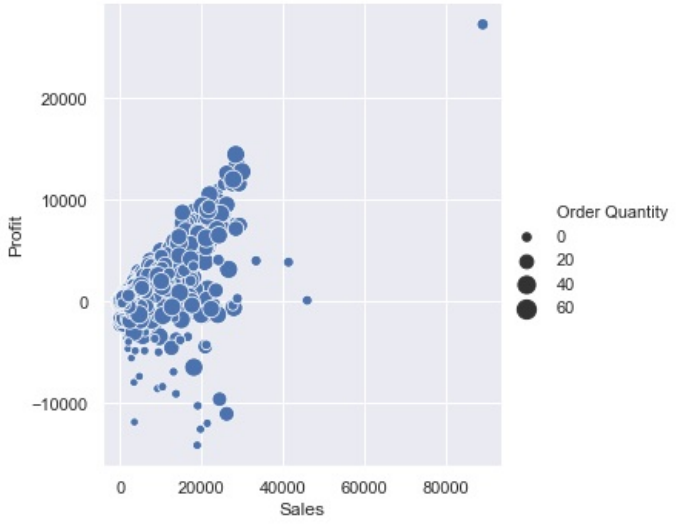


In [13]:

```
sns.relplot(x="Sales",y="Profit",size="Order Quantity",sizes=(30,150),data=df2)
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x2c178d892b0>
```



In [77]:

```
df2.head()
```

Out[77]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Customer Name	Province	Region	Custo Segm
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Muhammed MacIntyre	Nunavut	Nunavut	S Busir
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Barry French	Nunavut	Nunavut	Consu
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Barry French	Nunavut	Nunavut	Consu
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Clay Rozendal	Nunavut	Nunavut	Corpo
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Carlos Soltero	Nunavut	Nunavut	Consu

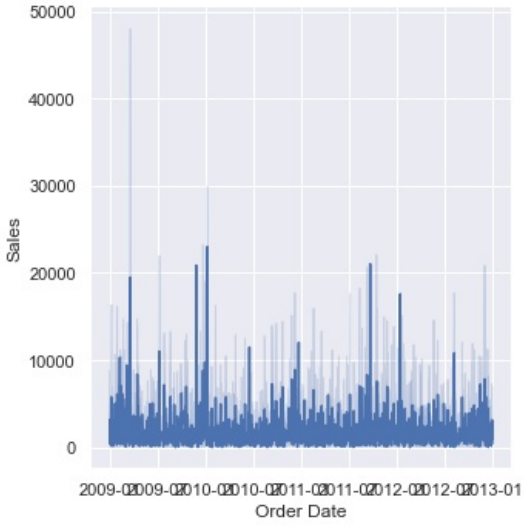
5 rows × 21 columns

In [78]:

```
sns.relplot("Order Date","Sales",data=df2,kind="line")
```

Out[78]:

```
<seaborn.axisgrid.FacetGrid at 0x2c17e5e9b20>
```



In [79]:

```
df2["Order Year"]=df2["Order Date"].dt.year
df2.head()
```

Out[79]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Province	Region	Customer Segment	Prod Categ
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Nunavut	Nunavut	Small Business	Of Supp
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Nunavut	Nunavut	Consumer	Of Supp
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Nunavut	Nunavut	Consumer	Of Supp
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Nunavut	Nunavut	Corporate	Technol
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Nunavut	Nunavut	Consumer	Of Supp

5 rows × 22 columns

In [80]:

```
df2["Order Year"].unique()  #very good method for getting the unique values in a column
```

Out[80]:

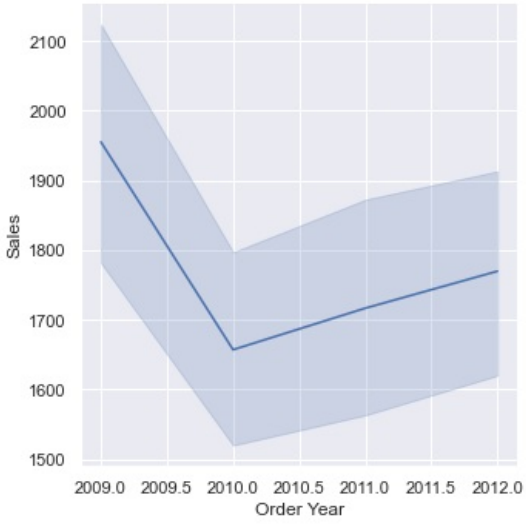
```
array([2010, 2012, 2011, 2009], dtype=int64)
```

In [81]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line")
```

Out[81]:

```
<seaborn.axisgrid.FacetGrid at 0x2c17f891a00>
```

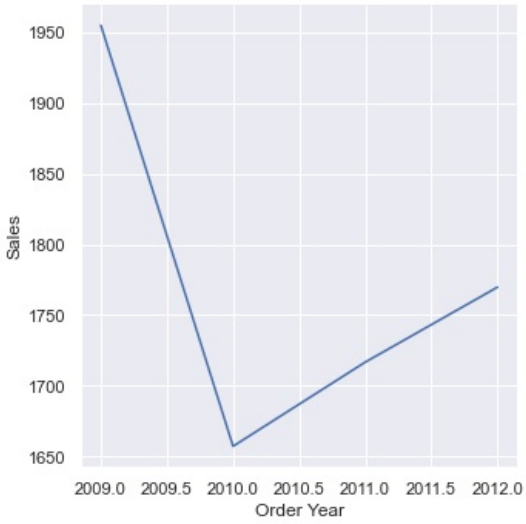


In [82]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",ci=None)
#ci is confidence index of 95% around the main mean line as shown in above fig
```

Out[82]:

```
<seaborn.axisgrid.FacetGrid at 0x2c1004e3670>
```

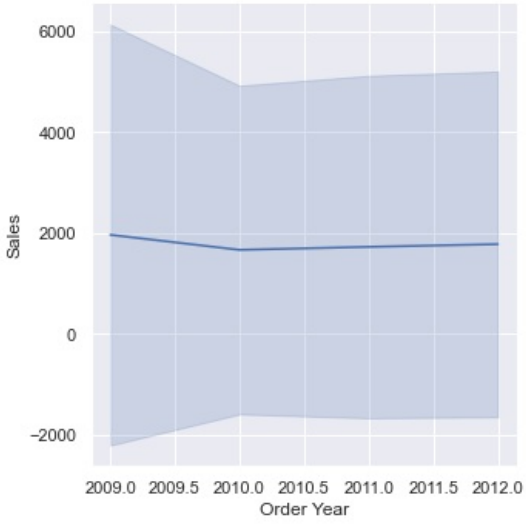


In [83]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",ci="sd")  #standard deviation from the mean line
```

Out[83]:

<seaborn.axisgrid.FacetGrid at 0x2c100942f10>

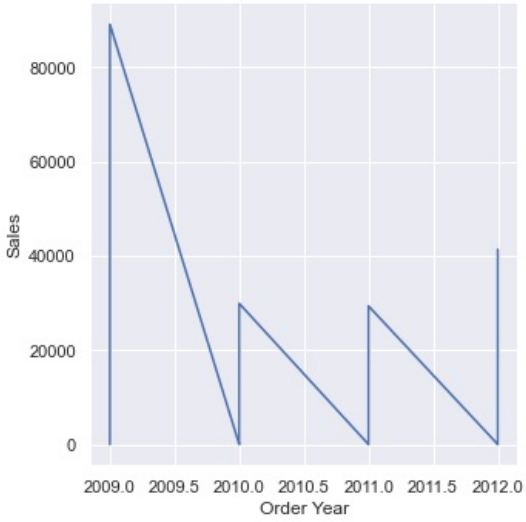


In [21]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",estimator=None) #Real graph
```

Out[21]:

<seaborn.axisgrid.FacetGrid at 0x2c178d00a00>

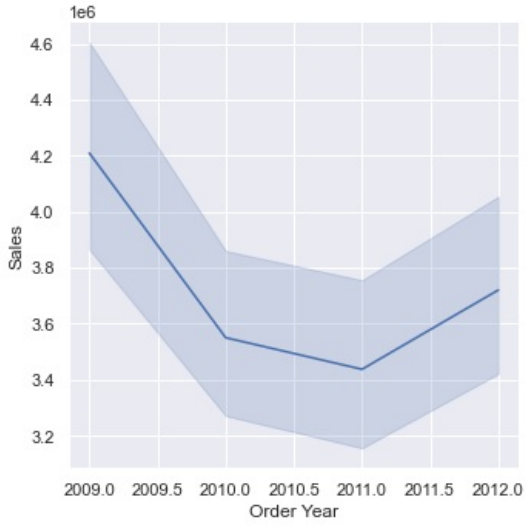


In [22]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",estimator=sum)
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x2c178d05d90>
```



In [84]:

```
df2["Customer Segment"].unique()
```

Out[84]:

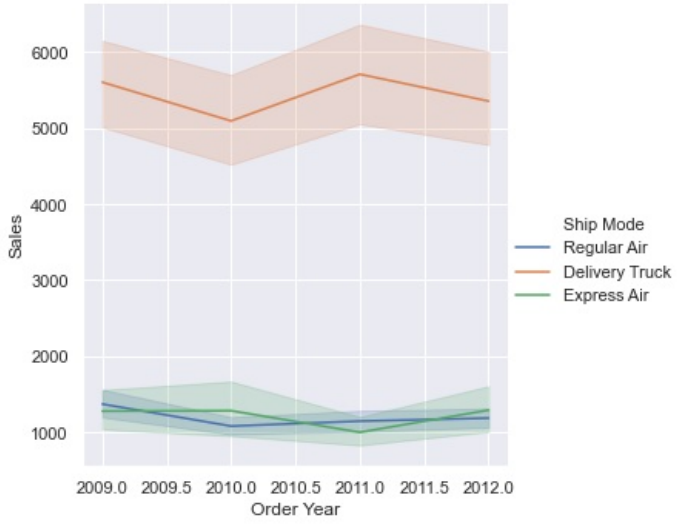
```
array(['Small Business', 'Consumer', 'Corporate', 'Home Office'],
      dtype=object)
```

In [85]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",hue="Ship Mode")
#Most of the sales are happening through Delivery Truck
```

Out[85]:

```
<seaborn.axisgrid.FacetGrid at 0x2c102b95eb0>
```

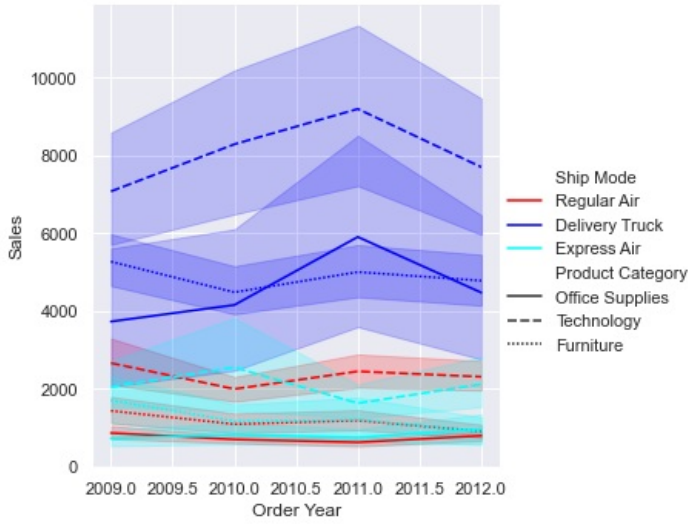


In [86]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",hue="Ship Mode",style="Product Category",\
            palette=["red","blue","aqua"])
#Most of the sales are coming from technology products delivered through truck
```

Out[86]:

```
<seaborn.axisgrid.FacetGrid at 0x2c1038c2a30>
```

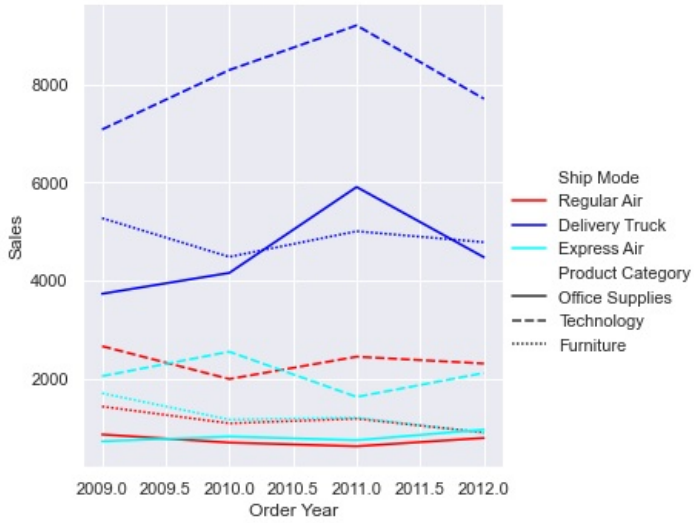


In [27]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",hue="Ship Mode",style="Product Category",\
            palette=["red","blue","aqua"],ci=None)
```

Out[27]:

```
<seaborn.axisgrid.FacetGrid at 0x2c1787a4220>
```

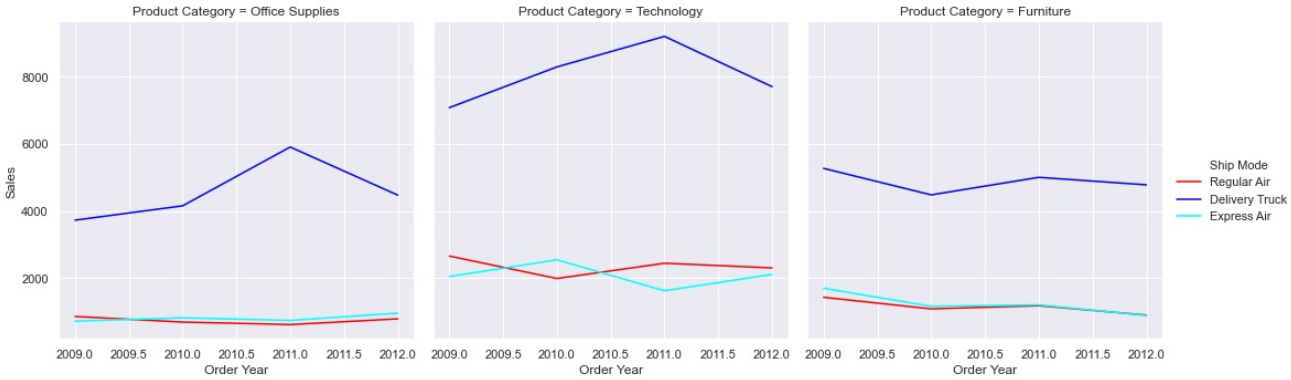


In [28]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",hue="Ship Mode",col="Product Category",\
            palette=["red","blue","aqua"],ci=None)
```

Out[28]:

```
<seaborn.axisgrid.FacetGrid at 0x2c178cc1700>
```

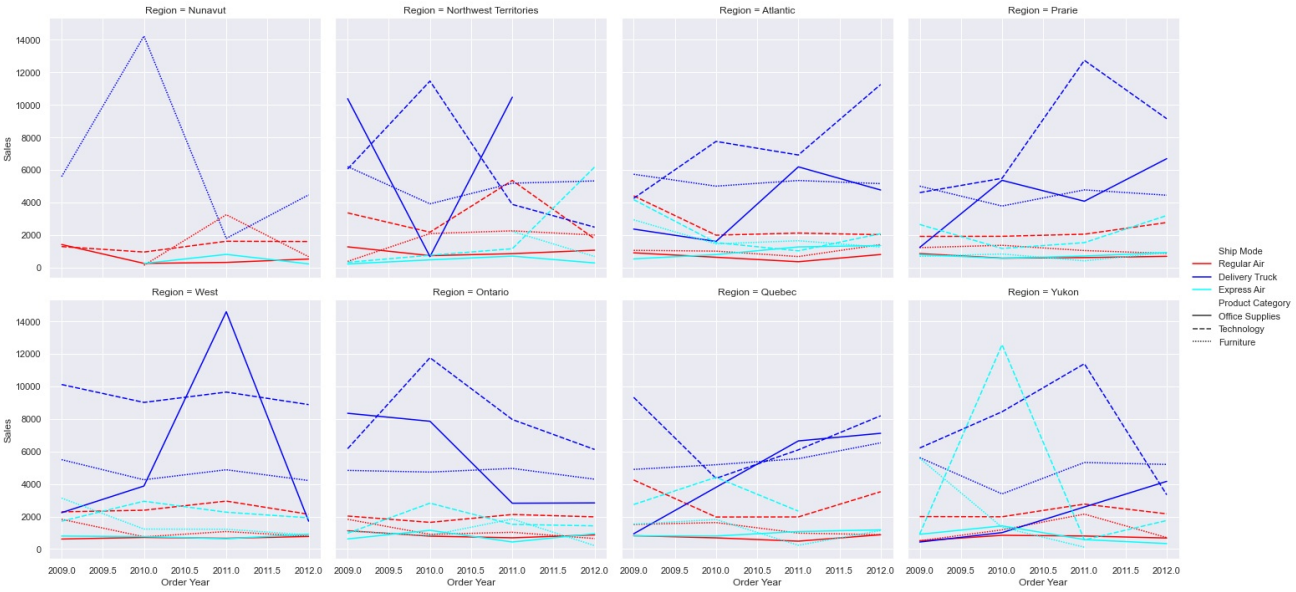


In [29]:

```
sns.relplot("Order Year","Sales",data=df2,kind="line",hue="Ship Mode",style="Product Category",\
            palette=["red","blue","aqua"],ci=None,col="Region",col_wrap=4)
```

Out[29]:

```
<seaborn.axisgrid.FacetGrid at 0x2c178837ca0>
```



In [89]:

```
df2[(df2["Profit"]>10000) & (df2["Customer Segment"]=="Corporate")].count()
```

Out[89]:

```
Row ID                  4
Order ID                4
Order Date              4
Order Priority          4
Order Quantity          4
Sales                   4
Discount                4
Ship Mode               4
Profit                  4
Unit Price              4
Shipping Cost           4
Customer Name           4
Province                4
Region                  4
Customer Segment        4
Product Category        4
Product Sub-Category    4
Product Name            4
Product Container       4
Product Base Margin     4
Ship Date               4
Order Year              4
dtype: int64
```

In [31]:

```
df2.head()
```

Out[31]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Province	Region	Customer Segment	Prod Categ
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Nunavut	Nunavut	Small Business	Of Supp
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Nunavut	Nunavut	Consumer	Of Supp
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Nunavut	Nunavut	Consumer	Of Supp
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Nunavut	Nunavut	Corporate	Technol
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Nunavut	Nunavut	Consumer	Of Supp

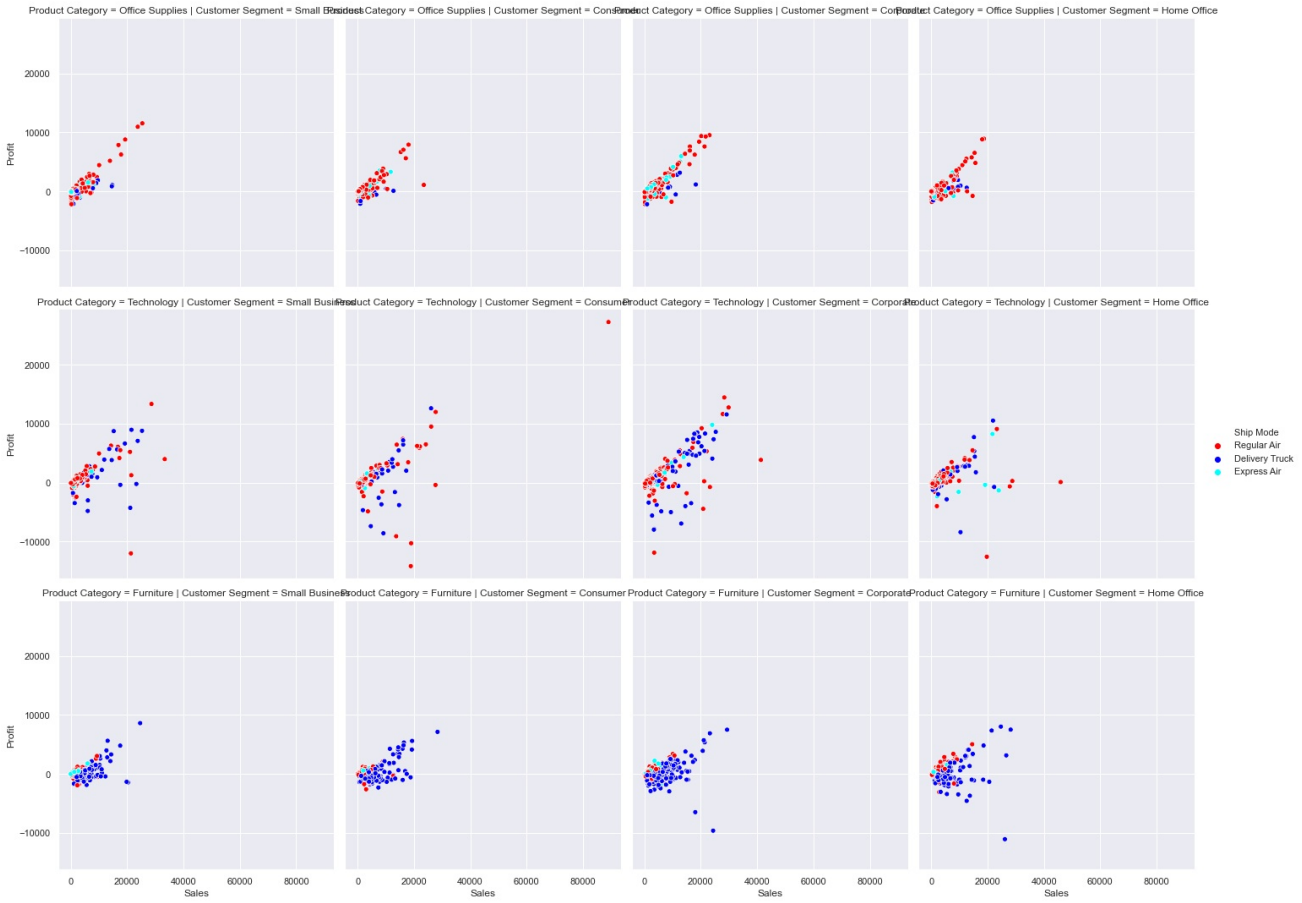
5 rows × 22 columns

In [231]:

```
sns.relplot("Sales","Profit",data=df2,hue="Ship Mode",row="Product Category",palette=\
            ["red","blue","aqua"],ci=None,col="Customer Segment")
```

Out[231]:

<seaborn.axisgrid.FacetGrid at 0x2c11e85ef10>

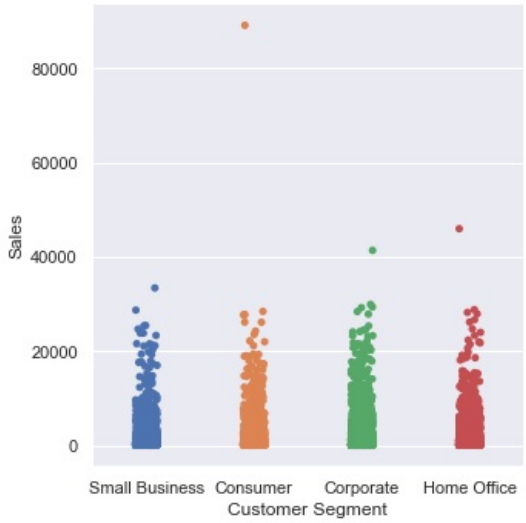


In [91]:

```
sns.catplot(x="Customer Segment",y="Sales",data=df2)
```

Out[91]:

<seaborn.axisgrid.FacetGrid at 0x2c103527a00>

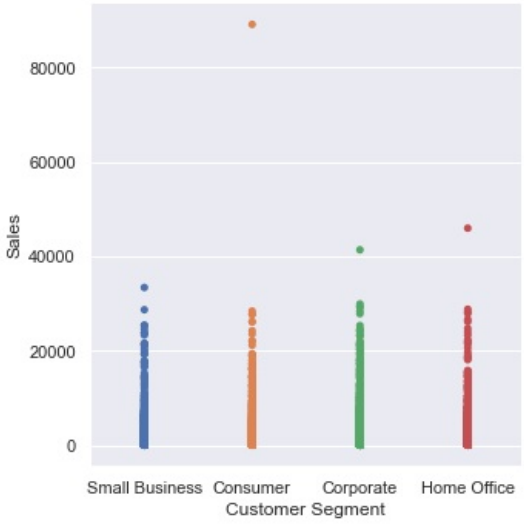


In [92]:

```
sns.catplot(x="Customer Segment",y="Sales",data=df2,jitter=False)
```

Out[92]:

```
<seaborn.axisgrid.FacetGrid at 0x2c103001340>
```

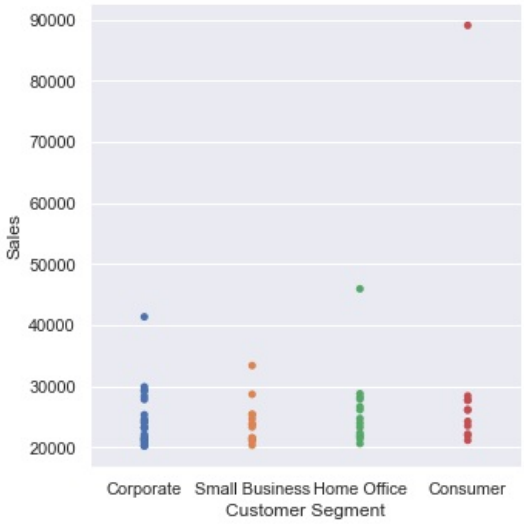


In [35]:

```
sns.catplot(x="Customer Segment",y="Sales",data=df2.query("Sales>20000"),jitter=False)
#In place of sales we can use any column.
```

Out[35]:

```
<seaborn.axisgrid.FacetGrid at 0x2c17b956b20>
```

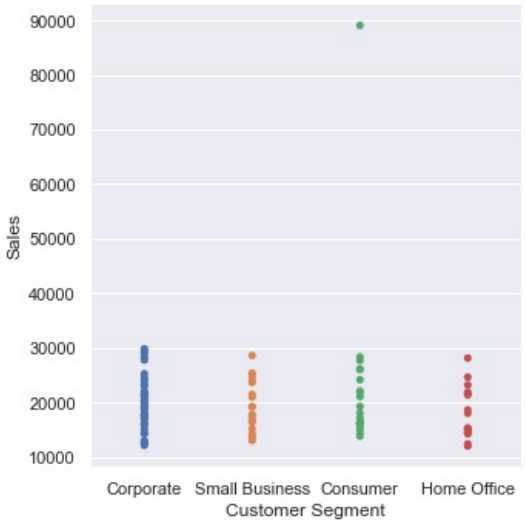


In [36]:

```
sns.catplot(x="Customer Segment",y="Sales",data=df2.query("Profit>5000"),kind="strip",jitter=False)
#Putting conditions on profit , only those with profit>5000 will be drawn
```

Out[36]:

```
<seaborn.axisgrid.FacetGrid at 0x2c17b957b20>
```

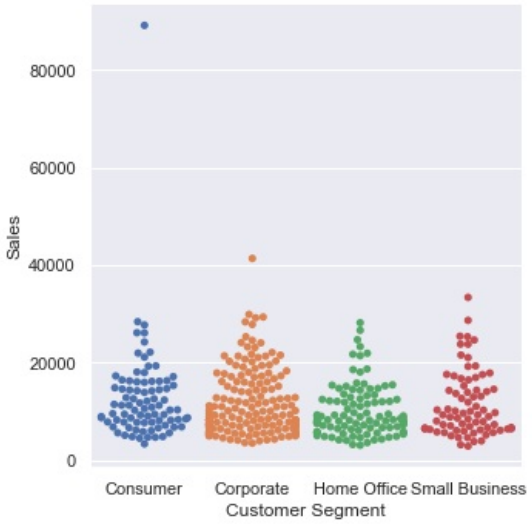


In [93]:

```
sns.catplot(x="Customer Segment",y="Sales",data=df2.query("Profit>1500"),kind="swarm")
```

Out[93]:

```
<seaborn.axisgrid.FacetGrid at 0x2c1038d9460>
```

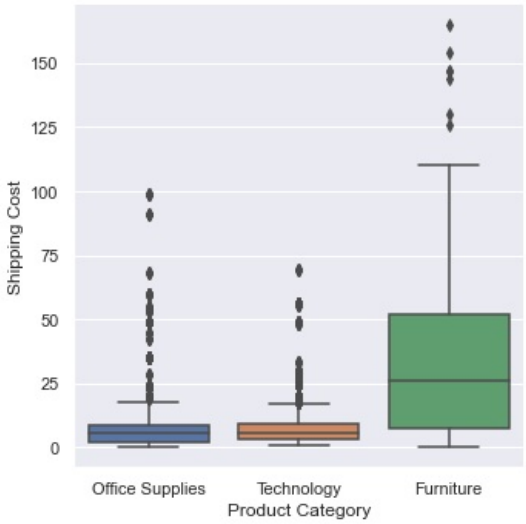


In [52]:

```
sns.catplot(x="Product Category",y="Shipping Cost",data=df2,kind="box")
#the median cost of furniture is very high
```

Out[52]:

```
<seaborn.axisgrid.FacetGrid at 0x2c17c8a4be0>
```



In [54]:

```
df2.head()
```

Out[54]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Province	Region	Customer Segment	Prod Categ
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Nunavut	Nunavut	Small Business	Of Supp
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Nunavut	Nunavut	Consumer	Of Supp
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Nunavut	Nunavut	Consumer	Of Supp
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Nunavut	Nunavut	Corporate	Technol
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Nunavut	Nunavut	Consumer	Of Supp

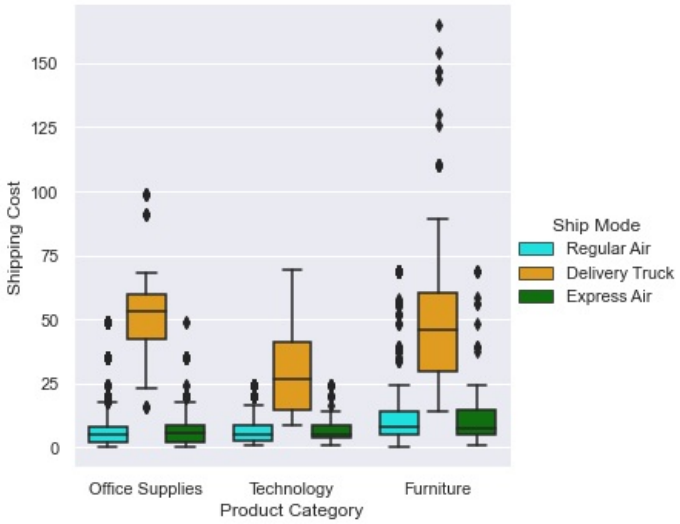
5 rows × 22 columns

In [230]:

```
sns.catplot(x="Product Category",y="Shipping Cost",hue="Ship Mode",data=df2,kind="box",\
            palette=["aqua","orange","green"])
```

Out[230]:

```
<seaborn.axisgrid.FacetGrid at 0x2c122b01ca0>
```

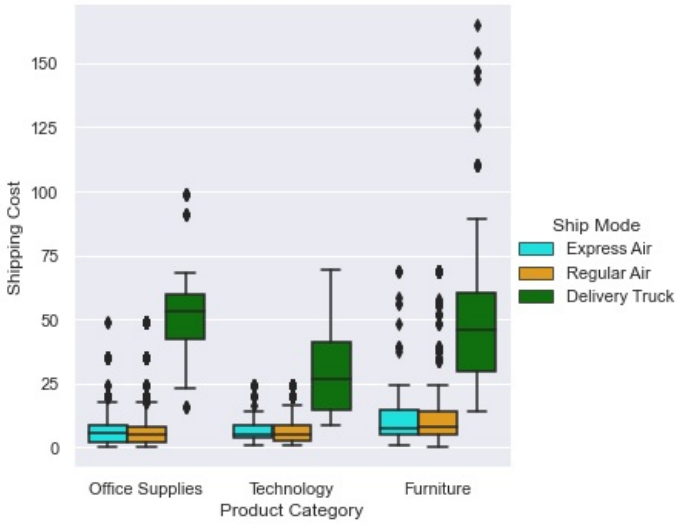


In [229]:

```
sns.catplot(x="Product Category",y="Shipping Cost",hue="Ship Mode",hue_order=\
["Express Air","Regular Air","Delivery Truck"],data=df2,kind="box",palette=["aqua","orange","green"])
#by this way we can change the order of hue
```

Out[229]:

```
<seaborn.axisgrid.FacetGrid at 0x2c122abef70>
```

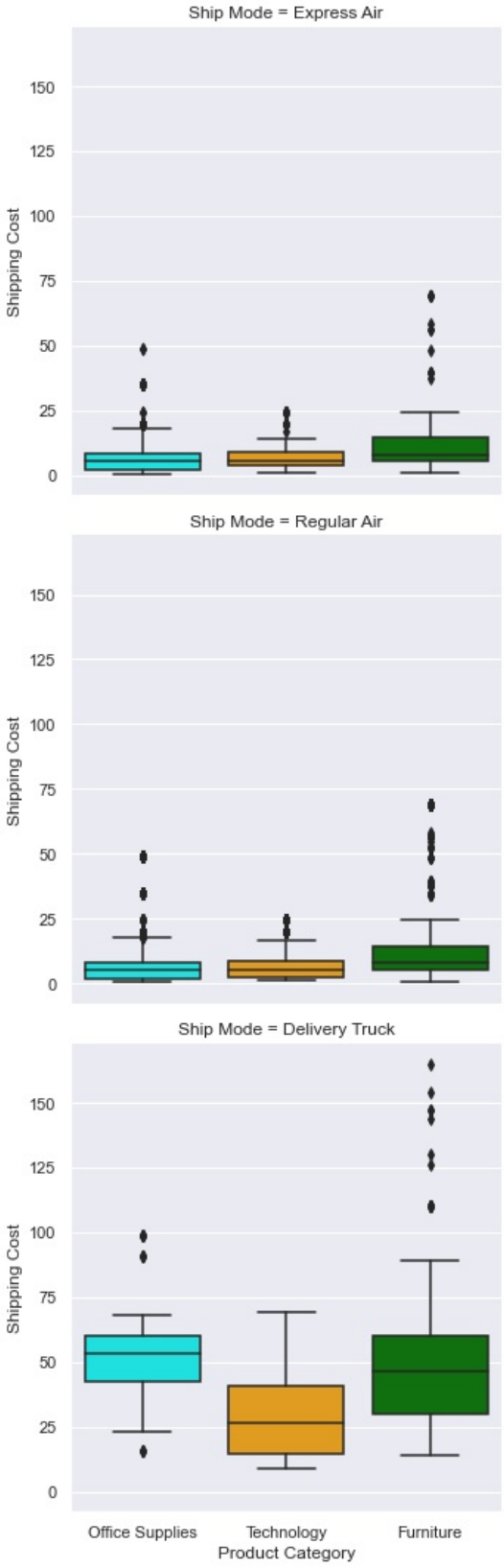


In [228]:

```
sns.catplot(x="Product Category",y="Shipping Cost",row="Ship Mode",data=df2,kind="box",palette=\
            ["aqua","orange","green"],row_order=["Express Air","Regular Air","Delivery Truck"])
#printing them in rows and changing the order of rows
```

Out[228]:

<seaborn.axisgrid.FacetGrid at 0x2c11ffd2490>



In [227]:

```
sns.catplot(x="Product Category",y="Shipping Cost",col="Ship Mode",col_order=\
["Express Air","Regular Air","Delivery Truck"],data=df2,kind="box",palette=["aqua","orange","green"])
#printing them in columns and changing the order of columns
```

Out[227]:

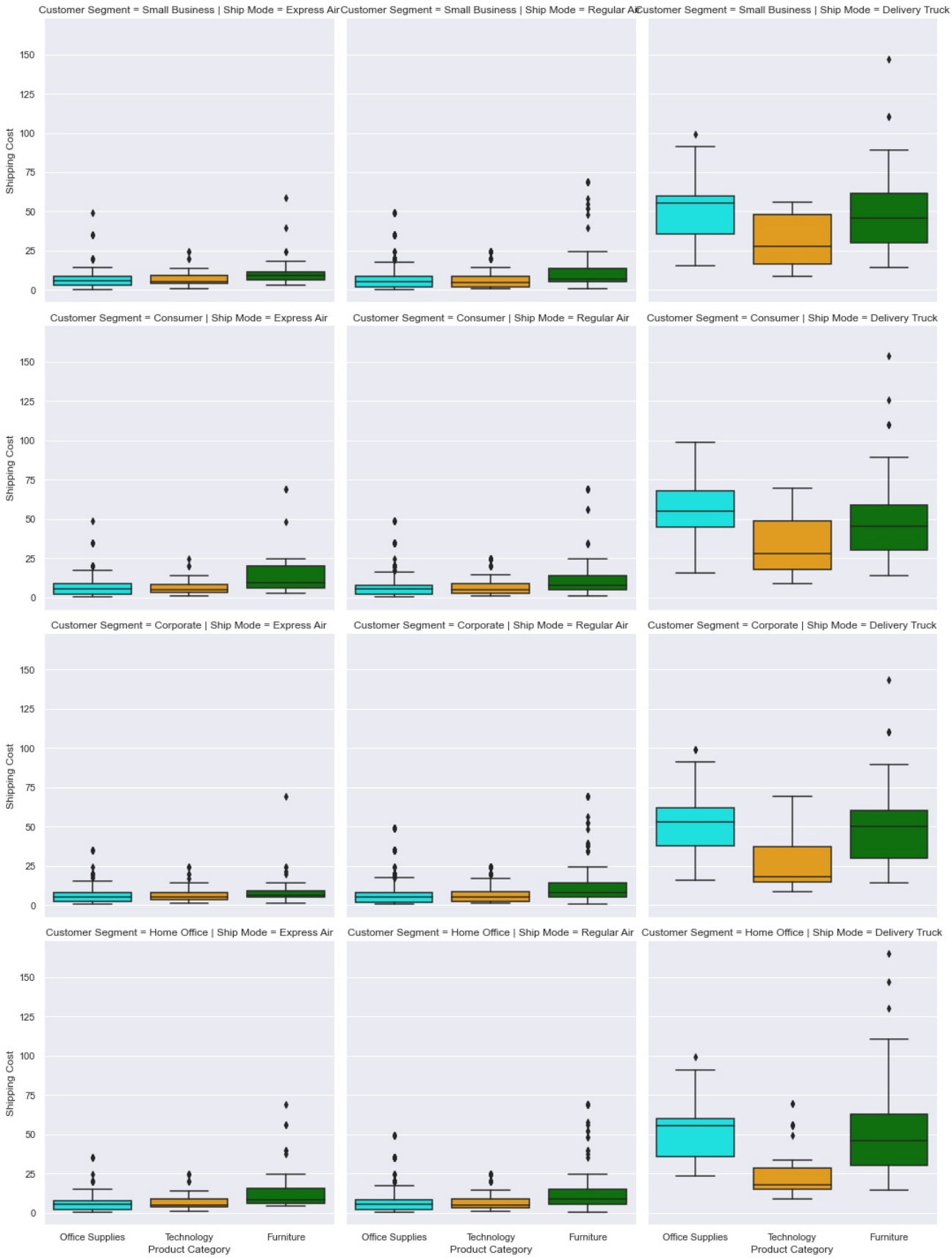
```
<seaborn.axisgrid.FacetGrid at 0x2c11fe41fd0>
```

In [226]:

```
sns.catplot(x="Product Category",y="Shipping Cost",col="Ship Mode",row="Customer Segment",\
col_order=["Express Air","Regular Air","Delivery Truck"],data=df2,kind="box"\
            ,palette=["aqua","orange","green"])
#both rows and columns together
```

Out[226]:

```
<seaborn.axisgrid.FacetGrid at 0x2c12139f2e0>
```


Customer Segment = Small Business | Ship Mode = Express Air
Customer Segment = Small Business | Ship Mode = Regular Air
Customer Segment = Small Business | Ship Mode = Delivery Truck
Customer Segment = Consumer | Ship Mode = Express Air
Customer Segment = Consumer | Ship Mode = Regular Air
Customer Segment = Consumer | Ship Mode = Delivery Truck
Customer Segment = Corporate | Ship Mode = Express Air
Customer Segment = Corporate | Ship Mode = Regular Air
Customer Segment = Corporate | Ship Mode = Delivery Truck
Customer Segment = Home Office | Ship Mode = Express Air
Customer Segment = Home Office | Ship Mode = Regular Air
Customer Segment = Home Office | Ship Mode = Delivery Truck
Shipping Cost
0
25
50
75
100
125
150
Office Supplies
Technology
Furniture
Product Category

In [102]:

```
sns.catplot(x="Ship Mode",y="Shipping Cost",kind="violin",data=df2,palette=["lime","red","yellow"])
```

Out[102]:

```
<seaborn.axisgrid.FacetGrid at 0x2c10ab27670>
```

In [225]:

```
sns.catplot(x="Ship Mode",y="Shipping Cost",hue="Product Category",kind="violin",data=df2,\
            palette=["lime","red","yellow"])
```

Out[225]:

```
<seaborn.axisgrid.FacetGrid at 0x2c11fcfd4c0>
```

In [103]:

```
df2.head()
```

Out[103]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Province	Region	Customer Segment	Produc Categor
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Nunavut	Nunavut	Small Business	Offic Supplie
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Nunavut	Nunavut	Consumer	Offic Supplie
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Nunavut	Nunavut	Consumer	Offic Supplie
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Nunavut	Nunavut	Corporate	Technolog
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Nunavut	Nunavut	Consumer	Offic Supplie

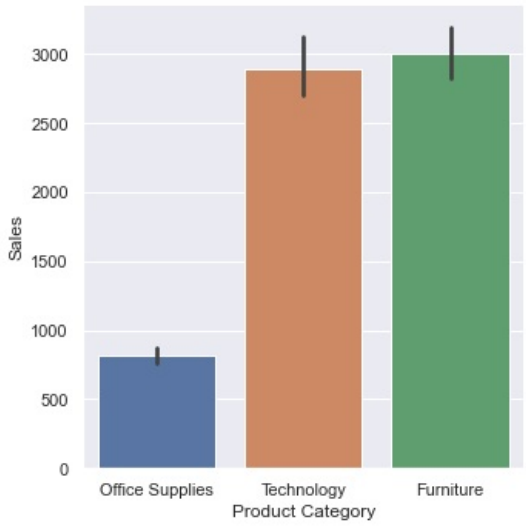
5 rows × 22 columns

In [105]:

```
sns.catplot(x="Product Category",y="Sales",data=df2,kind="bar")
```

Out[105]:

```
<seaborn.axisgrid.FacetGrid at 0x2c103cca520>
```

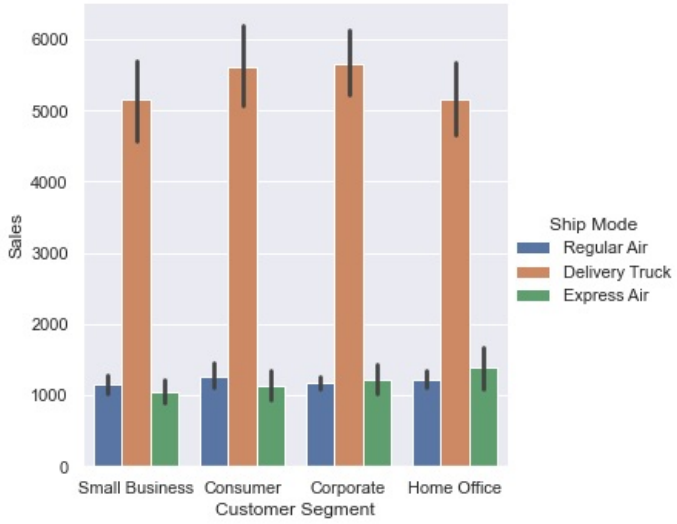


In [116]:

```
sns.catplot(x="Customer Segment",y="Sales",hue="Ship Mode",data=df2,kind="bar",ci=90)
```

Out[116]:

```
<seaborn.axisgrid.FacetGrid at 0x2c1082fdb50>
```

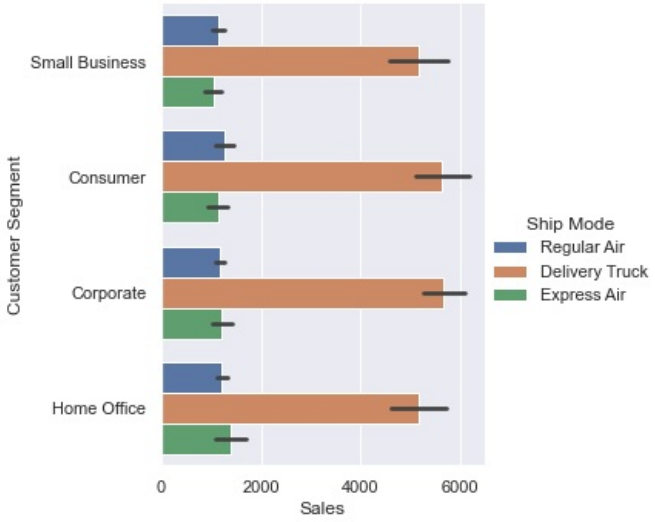


In [117]:

```
sns.catplot(y="Customer Segment",x="Sales",hue="Ship Mode",data=df2,kind="bar",ci=90)
```

Out[117]:

```
<seaborn.axisgrid.FacetGrid at 0x2c10cfc8880>
```

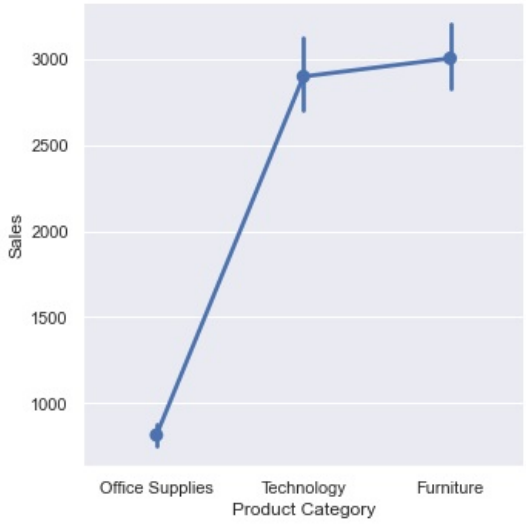


In [121]:

```
sns.catplot(x="Product Category",y="Sales",data=df2,kind="point")
```

Out[121]:

```
<seaborn.axisgrid.FacetGrid at 0x2c10d88c190>
```

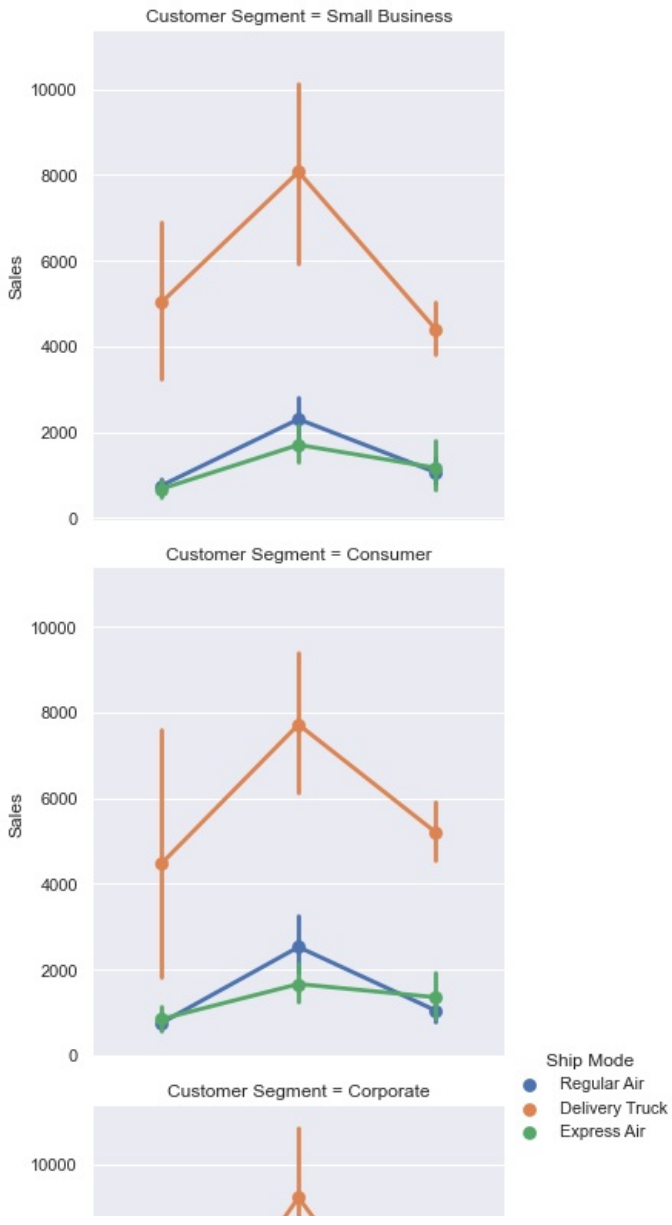


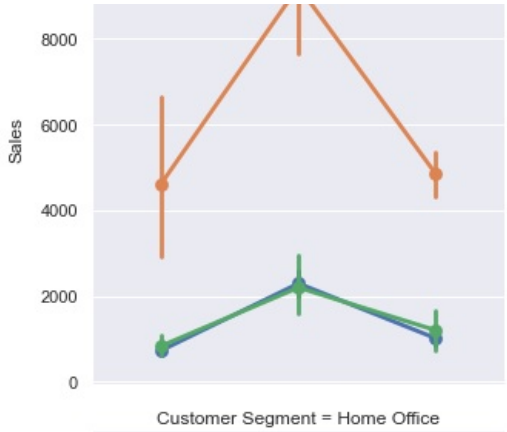
In [224]:

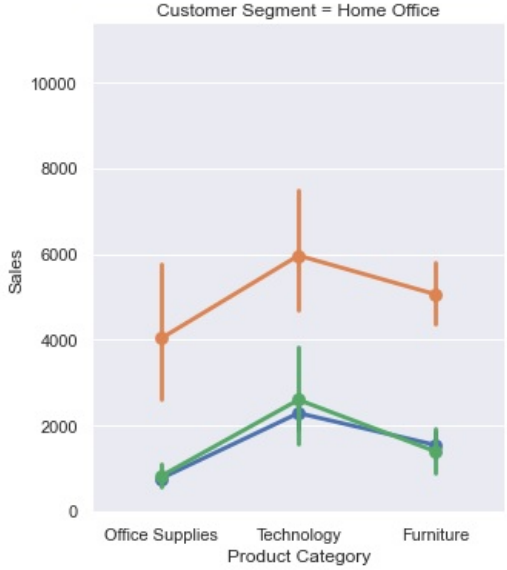
```
sns.catplot(x="Product Category",y="Sales",hue="Ship Mode",row="Customer Segment",data=df2,kind=\
            "point")
```

Out[224]:

```
<seaborn.axisgrid.FacetGrid at 0x2c11ff31a30>
```







Distribution plots with seaborn

In [127]:

```
df2.head()
```

Out[127]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Province	Region	Customer Segment	Prod Categ
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Nunavut	Nunavut	Small Business	Of Supp
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Nunavut	Nunavut	Consumer	Of Supp
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Nunavut	Nunavut	Consumer	Of Supp
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Nunavut	Nunavut	Corporate	Technol
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Nunavut	Nunavut	Consumer	Of Supp

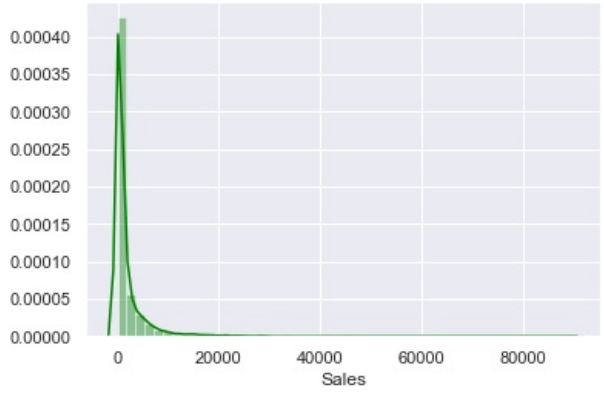
5 rows × 22 columns

In [223]:

```
sns.distplot(df2["Sales"],color="green")# we can see that most of the sales are happening around
# values between 0 and 1000 and then reducing a lot till it is mere negligible at about greater
#than 10000
```

Out[223]:

```
<AxesSubplot:xlabel='Sales'>
```

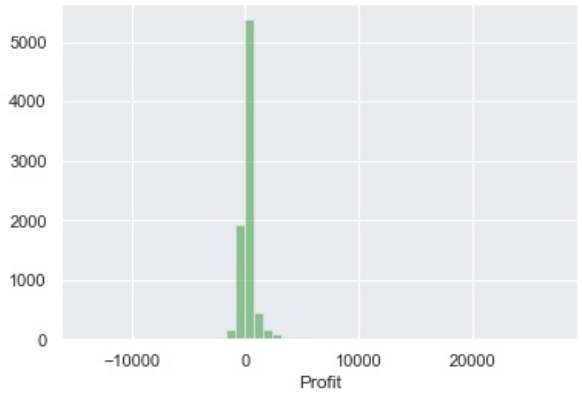


In [143]:

```
sns.distplot(a=df2.Profit,color="green",kde=False)
```

Out[143]:

```
<AxesSubplot:xlabel='Profit'>
```

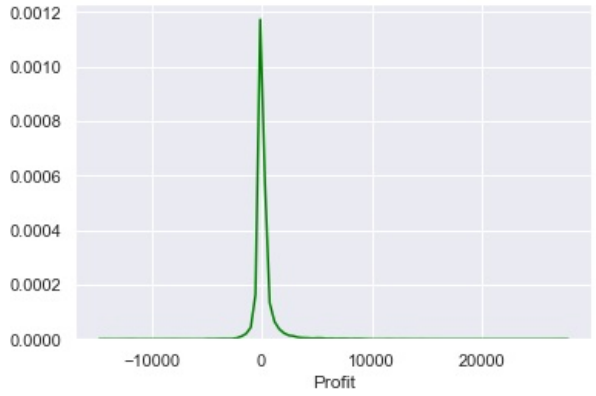


In [144]:

```
sns.distplot(a=df2.Profit,color="green",hist=False)
```

Out[144]:

```
<AxesSubplot:xlabel='Profit'>
```

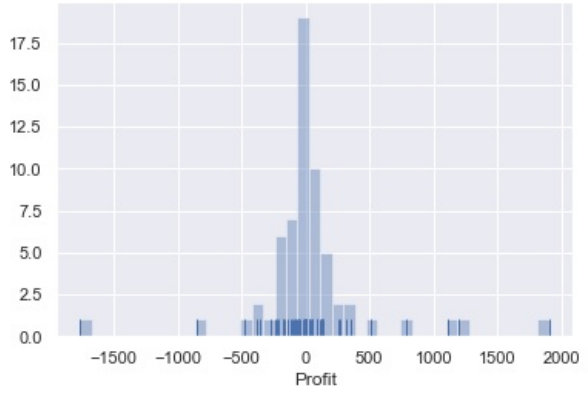


In [222]:

```
sns.distplot(a=df2.Profit[(df2["Ship Mode"]=="Regular Air") & (df2.Region=="Nunavut")],kde=False,\
             rug=True)
#Very Important as the conditions are given so we can visualize specifically each and every subgroup
```

Out[222]:

```
<AxesSubplot:xlabel='Profit'>
```



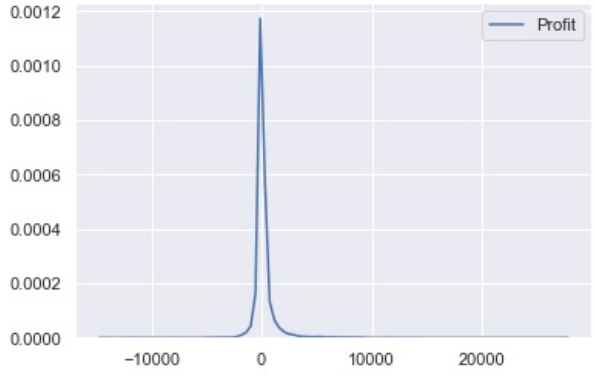
kdeplot

In [165]:

```
sns.kdeplot(df2.Profit)
```

Out[165]:

```
<AxesSubplot:>
```

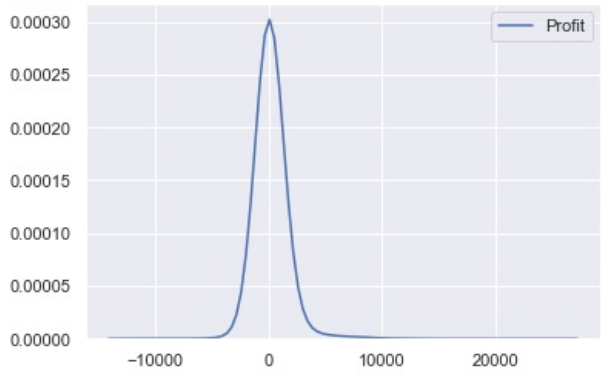


In [166]:

```
sns.kdeplot(df2.Profit,bw=1)
```

Out[166]:

```
<AxesSubplot:>
```



using scipy for plotting probability distributions

In [167]:

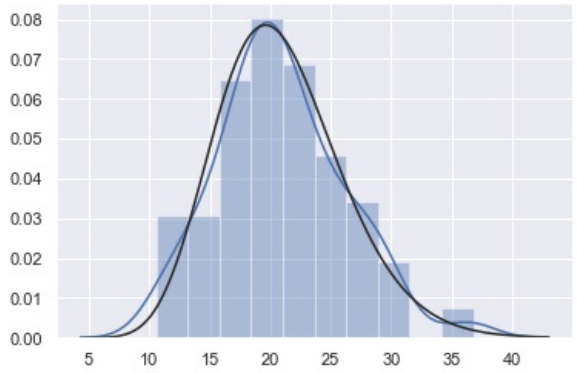
```
from scipy import stats
```

In [187]:

```
a=np.random.gamma(20,size=100)
sns.distplot(a,fit=stats.gamma)
```

Out[187]:

<AxesSubplot:>

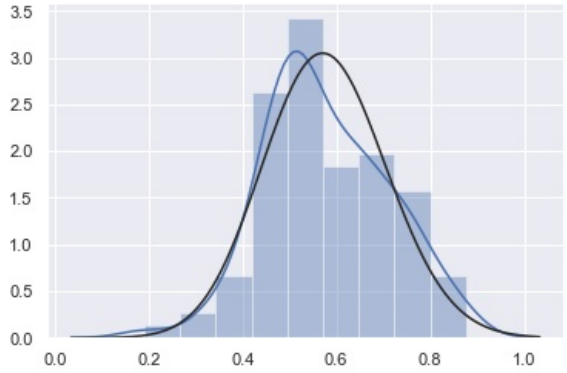


In [196]:

```
b=np.random.beta(7,5,size=100)
sns.distplot(b,fit=stats.beta)
```

Out[196]:

<AxesSubplot:>



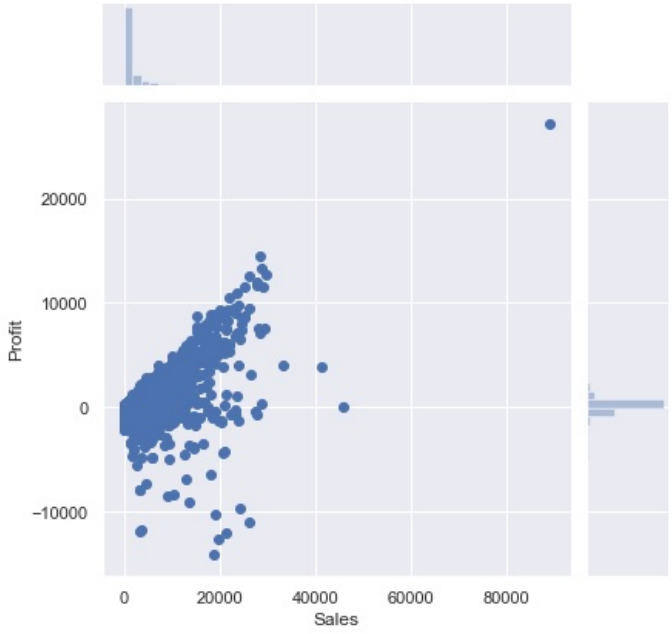
Jointplot

In [198]:

```
sns.jointplot(x="Sales",y="Profit",data=df2)
```

Out[198]:

```
<seaborn.axisgrid.JointGrid at 0x2c11480bbb0>
```

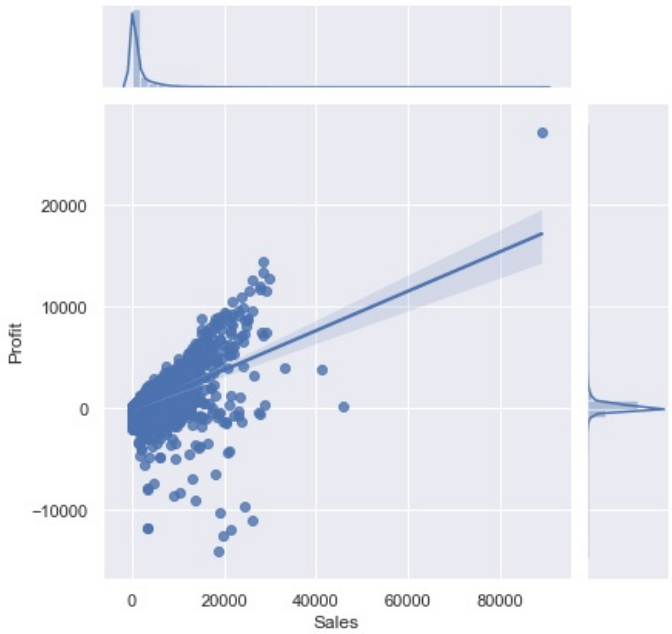


In [203]:

```
sns.jointplot(x="Sales",y="Profit",data=df2,kind="reg")
```

Out[203]:

```
<seaborn.axisgrid.JointGrid at 0x2c10db6fbb0>
```

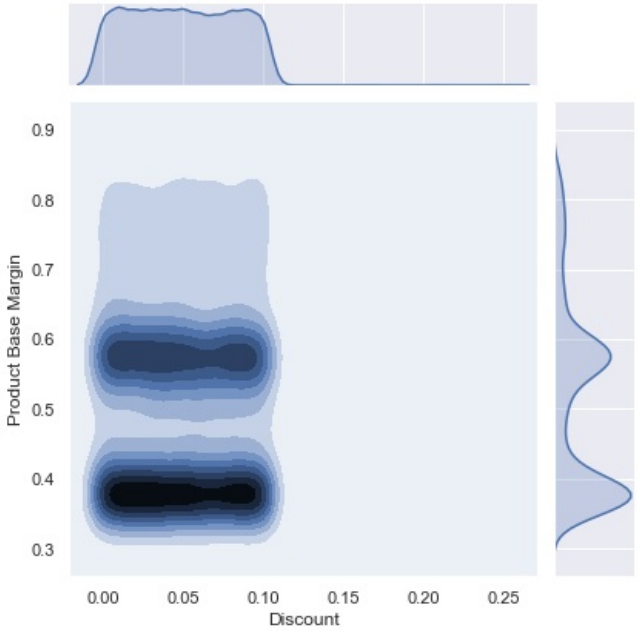


In [221]:

```
sns.jointplot(x="Discount",y="Product Base Margin",data=df2,kind="kde")
```

Out[221]:

```
<seaborn.axisgrid.JointGrid at 0x2c11fc22310>
```



Pairplots

In [213]:

```
sns.pairplot(df2.iloc[:,2:])
```

Out[213]:

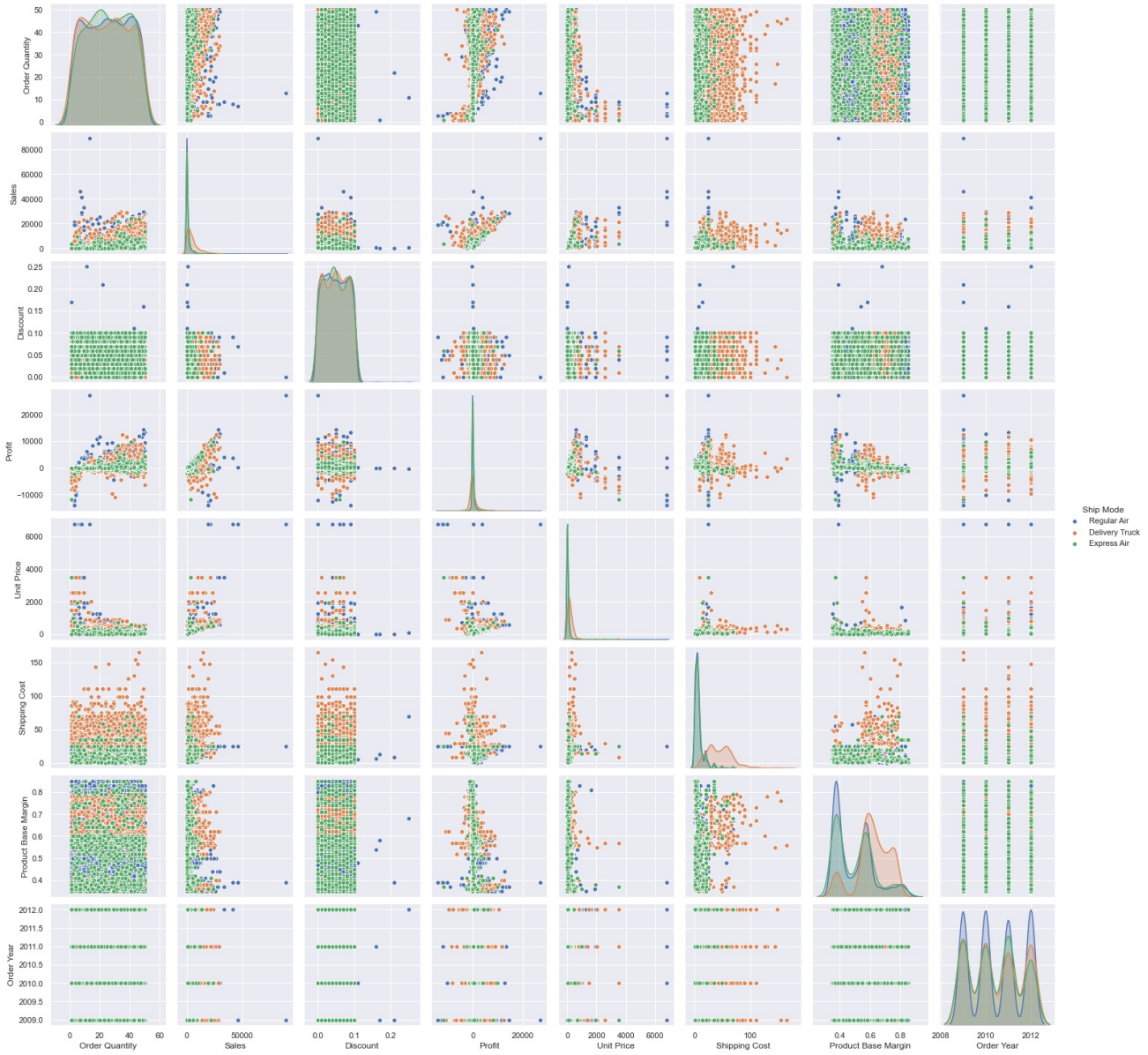
```
<seaborn.axisgrid.PairGrid at 0x2c1179c4f10>
```

In [214]:

```
sns.pairplot(df2.iloc[:,2:],hue="Ship Mode")
```

Out[214]:

<seaborn.axisgrid.PairGrid at 0x2c11a4a2940>



In [215]:

```
df2.head()
```

Out[215]:

	Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	...	Province	Region	Customer Segment	Produc Categor
0	1	3	2010-10-13	Low	6	261.5400	0.04	Regular Air	-213.2500	38.94	...	Nunavut	Nunavut	Small Business	Offic Supplie
1	49	293	2012-10-01	High	49	10123.0200	0.07	Delivery Truck	457.8100	208.16	...	Nunavut	Nunavut	Consumer	Offic Supplie
2	50	293	2012-10-01	High	27	244.5700	0.01	Regular Air	46.7075	8.69	...	Nunavut	Nunavut	Consumer	Offic Supplie
3	80	483	2011-07-10	High	30	4965.7595	0.08	Regular Air	1198.9710	195.99	...	Nunavut	Nunavut	Corporate	Technolog
4	85	515	2010-08-28	Not Specified	19	394.2700	0.08	Regular Air	30.9400	21.78	...	Nunavut	Nunavut	Consumer	Offic Supplie

5 rows × 22 columns

So I think this much is ok for a beginner Happy Learning