In [25]:
```
%matplotlib inline
# Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# File to Load (Remember to change these)
city_data_to_load = "data/city_data.csv"
ride_data_to_load = "data/ride_data.csv"

# Read the City and Ride Data
city_df=pd.read_csv(city_data_to_load)
ride_df=pd.read_csv(ride_data_to_load)

# Combine the data into a single dataset
combo=pd.merge(city_df,ride_df,)
# Display the data table for preview
combo.head()
```

Out[25]:

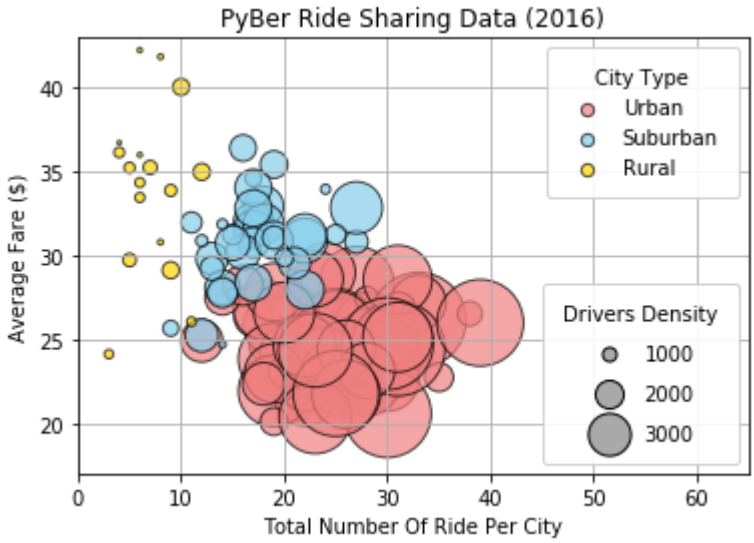
	city	driver_count	type	date	fare	ride_id
0	Richardfort	38	Urban	2018-02-24 08:40:38	13.93	5628545007794
1	Richardfort	38	Urban	2018-02-13 12:46:07	14.00	910050116494
2	Richardfort	38	Urban	2018-02-16 13:52:19	17.92	820639054416
3	Richardfort	38	Urban	2018-02-01 20:18:28	10.26	9554935945413
4	Richardfort	38	Urban	2018-04-17 02:26:37	23.00	720020655850

In [ ]:

Bubble Plot of Ride Sharing Data

In [26]:
```
# Obtain the x and y coordinates for each of the three city types

average_fair_urban = combo[combo["type"] =="Urban"].groupby(['city'])["fare"].
mean()
total_ride_urban = combo[combo["type"] == "Urban"].groupby(['city'])["ride_id"
].count()
total_driver_urban = combo[combo["type"] == "Urban"].groupby(['city'])["driver
_count"].sum()

average_fair_suburban = combo[combo["type"] =="Suburban"].groupby(['city'])["f
are"].mean()
total_ride_suburban = combo[combo["type"] == "Suburban"].groupby(['city'])["ri
de_id"].count()
total_driver_suburban = combo[combo["type"] == "Suburban"].groupby(['city'])[
"driver_count"].sum()

average_fair_rural = combo[combo["type"] =="Rural"].groupby(['city'])["fare"].
mean()
total_ride_rural = combo[combo["type"] == "Rural"].groupby(['city'])["ride_id"
].count()
total_driver_rural = combo[combo["type"] == "Rural"].groupby(['city'])["driver
_count"].sum()
# Build the scatter plots for each city types
plt.xlim(0,65)
plt.ylim(17,43)
plt.grid(True)
plt.title("PyBer Ride Sharing Data (2016)")
plt.xlabel("Total Number Of Ride Per City")
plt.ylabel("Average Fare ($)")
plt.scatter(total_ride_urban,average_fair_urban, marker="o", facecolors="light
coral", edgecolors="black", s=total_driver_urban, label ="Urban", alpha ="0.7"
)
plt.scatter(total_ride_suburban,average_fair_suburban, marker="o", facecolors=
"skyblue", edgecolors="black", s=total_driver_suburban, label = "Suburban", al
pha ="0.7")
plt.scatter(total_ride_rural,average_fair_rural, marker="o", facecolors="gold"
, edgecolors="black", s=total_driver_rural, label = "Rural", alpha ="0.7")
# Incorporate the other graph properties

# Create a legend
lgd = plt.legend(numpoints=1, loc=1, borderpad=1,
           frameon=True, framealpha=0.9, title="City Type")
for handle in lgd.legendHandles:
    handle.set_sizes([40.0])

pws = [4,500,1000,2000,3000]
for pw in pws:
    plt.scatter([], [], s=(pw**2)/2e4, c="darkgray",label=str(pw),edgecolors=
"black")

h, l = plt.gca().get_legend_handles_labels()
plt.legend(h[5:], l[5:], labelspacing=1, title="Drivers Density", borderpad=1,

           frameon=True, framealpha=0.9, loc=4, numpoints=1)

plt.gca().add_artist(lgd)
```

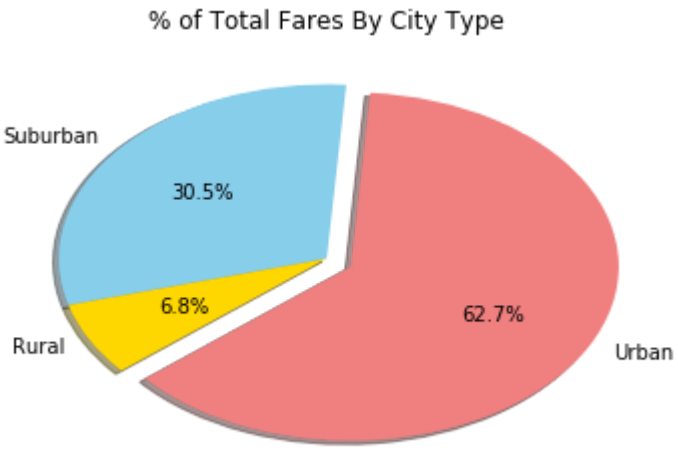
```
plt.show()
```



Total Fares by City Type

In [27]:
```
urban_fare = combo[combo["type"] =="Urban"]["fare"].sum()
suburban_fare = combo[combo["type"] =="Suburban"]["fare"].sum()
rural_fare = combo[combo["type"] =="Rural"]["fare"].sum()

plt.title("% of Total Fares By City Type")
labels = ["Urban", "Suburban", "Rural"]
sizes = [urban_fare, suburban_fare,rural_fare]
colors = ["lightcoral", "skyblue", "gold"]
explode = (0.1, 0, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct="%1.1f%%", shadow=True, startangle=220)
```

Out[27]:
```
([<matplotlib.patches.Wedge at 0xa6d37b8>,
  <matplotlib.patches.Wedge at 0xa6dc2e8>,
  <matplotlib.patches.Wedge at 0xa6dccf8>],
 [Text(1.0683,-0.546573,'Urban'),
  Text(-0.850539,0.697556,'Suburban'),
  Text(-0.973582,-0.511994,'Rural')],
 [Text(0.623173,-0.318834,'62.7%'),
  Text(-0.46393,0.380485,'30.5%'),
  Text(-0.531045,-0.27927,'6.8%')])
```

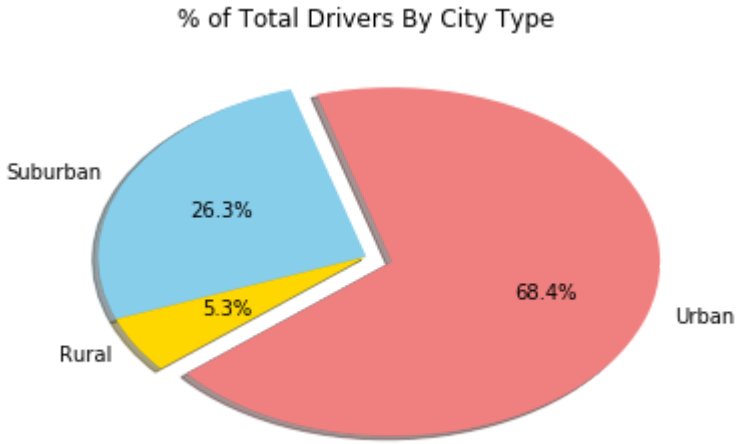


In [28]:
```
# Show Figure
plt.show()
```

Total Rides by City Type

In [29]:
```
urban_drivers = combo[combo["type"] =="Urban"]["driver_count"].count()
suburban_drivers = combo[combo["type"] =="Suburban"]["driver_count"].count()
rural_drivers = combo[combo["type"] =="Rural"]["driver_count"].count()

plt.title("% of Total Drivers By City Type")
labels = ["Urban", "Suburban", "Rural"]
sizes = [urban_drivers, suburban_drivers,rural_drivers]
colors = ["lightcoral", "skyblue", "gold"]
explode = (0.1, 0, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct="%1.1f%%", shadow=True, startangle=220)
```

Out[29]:
```
([<matplotlib.patches.Wedge at 0xa718898>,
  <matplotlib.patches.Wedge at 0xa721320>,
  <matplotlib.patches.Wedge at 0xa721d30>],
 [Text(1.14853,-0.347682,'Urban'),
  Text(-0.986001,0.48765,'Suburban'),
  Text(-0.947535,-0.558728,'Rural')],
 [Text(0.669975,-0.202815,'68.4%'),
  Text(-0.537819,0.265991,'26.3%'),
  Text(-0.516838,-0.304761,'5.3%')])
```

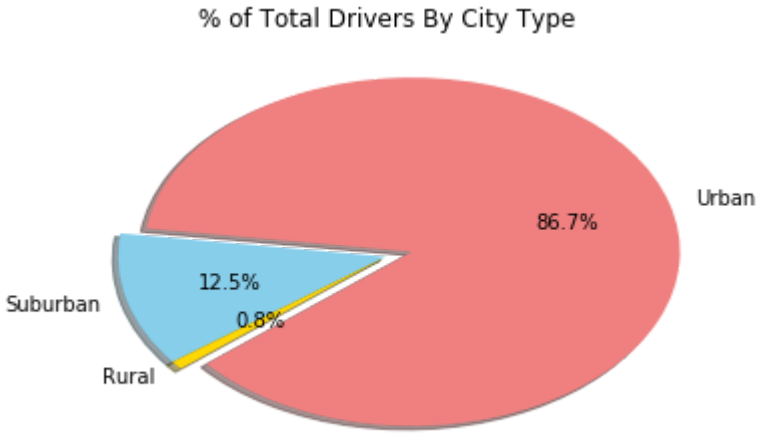


In [30]:
```
# Show Figure
plt.show()
```

Total Drivers by City Type

In [31]:
```
urban_drivers = combo[combo["type"] =="Urban"]["driver_count"].sum()
suburban_drivers = combo[combo["type"] =="Suburban"]["driver_count"].sum()
rural_drivers = combo[combo["type"] =="Rural"]["driver_count"].sum()

plt.title("% of Total Drivers By City Type")
labels = ["Urban", "Suburban", "Rural"]
sizes = [urban_drivers, suburban_drivers,rural_drivers]
colors = ["lightcoral", "skyblue", "gold"]
explode = (0.1, 0, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct="%1.1f%%", shadow=True, startangle=220)
```

Out[31]:
```
([<matplotlib.patches.Wedge at 0x8f26dd8>,
  <matplotlib.patches.Wedge at 0x8f05278>,
  <matplotlib.patches.Wedge at 0x8eef160>],
 [Text(1.15269,0.333622,'Urban'),
  Text(-1.06382,-0.279787,'Suburban'),
  Text(-0.859754,-0.686165,'Rural')],
 [Text(0.672403,0.194613,'86.7%'),
  Text(-0.580267,-0.152611,'12.5%'),
  Text(-0.468957,-0.374272,'0.8%')])
```



In [32]:
```
# Show Figure
plt.show()
```

In [ ]:

In [ ]: