5.5.2   Pie Chart for the Percentage of Fares by City Type

While breaking down the percentage fares by each city type, you have a moment of brilliance: you're going to keep this on-brand by using the company's color scheme of gold, sky blue, and coral.

We will create a pie chart by using the MATLAB approach, using the `plt.pie()` function.

REWIND

To create a pie chart with the `plt.pie()` function, we need an array that contains the values and labels we are plotting.

For the pie chart, each pie wedge will represent a city and its percentage of the total fares. The labels will be the city type.

We can use `type_percents` for the values for each pie wedge and create an array for the labels.

Let's add the following code to a new cell:

```
# Build the percentage of fares by city type pie chart.
plt.pie(type_percents, labels=["Rural", "Suburban", "Urban"])
plt.show()
```

Our pie chart will look like this after we run the cell:
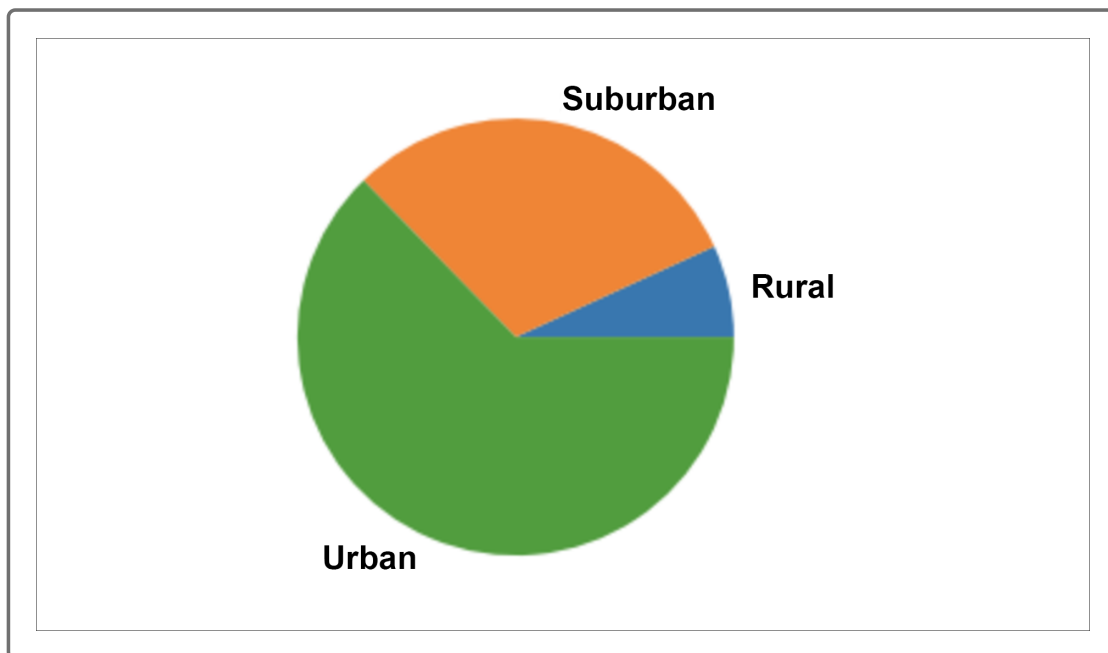
Now that we have a basic pie chart, we can add a title, add the percentages, change the color to adhere to the company color scheme, add a shadow to the pie chart, and adjust the start angle.

Let's edit the `plt.pie()` function with the following added features:

```
# Build the percentage of fares by city type pie chart.
plt.pie(type_percents,
    labels=["Rural", "Suburban", "Urban"],
```

```
      colors=["gold", "lightskyblue", "lightcoral"],
      explode=[0, 0, 0.1],
      autopct='%1.1f%%',
      shadow=True, startangle=150)
plt.title("% of Total Fares by City Type")
# Show Figure
plt.show()
```

When you run the cell, the pie chart should look like this.
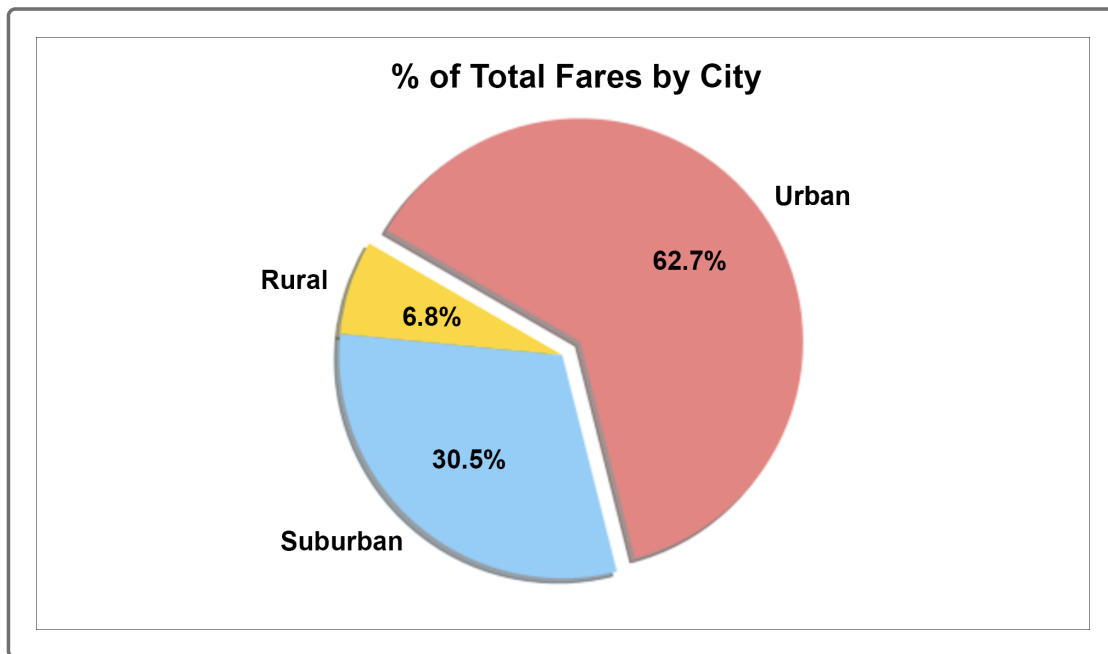
Before we save the pie chart, let's change the font size to 14. To do this, we will need to edit the code block.

First of all, there is no parameter for `fontsize` in pie charts like there is for scatter plots. However, Matpltolib has a way to change the default parameters for charts by using the `rcParams`, which accesses the run and configure settings for the Matplotlib parameters.

To change the default parameters we need to import the matplotlib library, which is not the same as the matplotlib.pyplot we imported for graphing with the `plt()` function. Let's edit our code block above and add `import`

`matplotlib as mpl` on the first line. After the `plt.title()`, we will change the font size by adding `mpl.rcParams['font.size'] = 14`.

The code block should look like this after setting the font size:

```python
# Import mpl to change the plot configurations using rcParams.
import matplotlib as mpl
# Build Pie Chart
plt.subplots(figsize=(10, 6))
plt.pie(type_percents,
    labels=["Rural", "Suburban", "Urban"],
    colors=["gold", "lightskyblue", "lightcoral"],
    explode=[0, 0, 0.1],
    autopct='%1.1f%%',
    shadow=True, startangle=150)
plt.title("% of Total Fares by City Type")
# Change the default font size from 10 to 14.
mpl.rcParams['font.size'] = 14
# Save Figure
plt.savefig("analysis/Fig5.png")
# Show Figure
plt.show()
```

When you run the cell, the final pie chart should look the same as before, but larger.

% of Total Fares by City
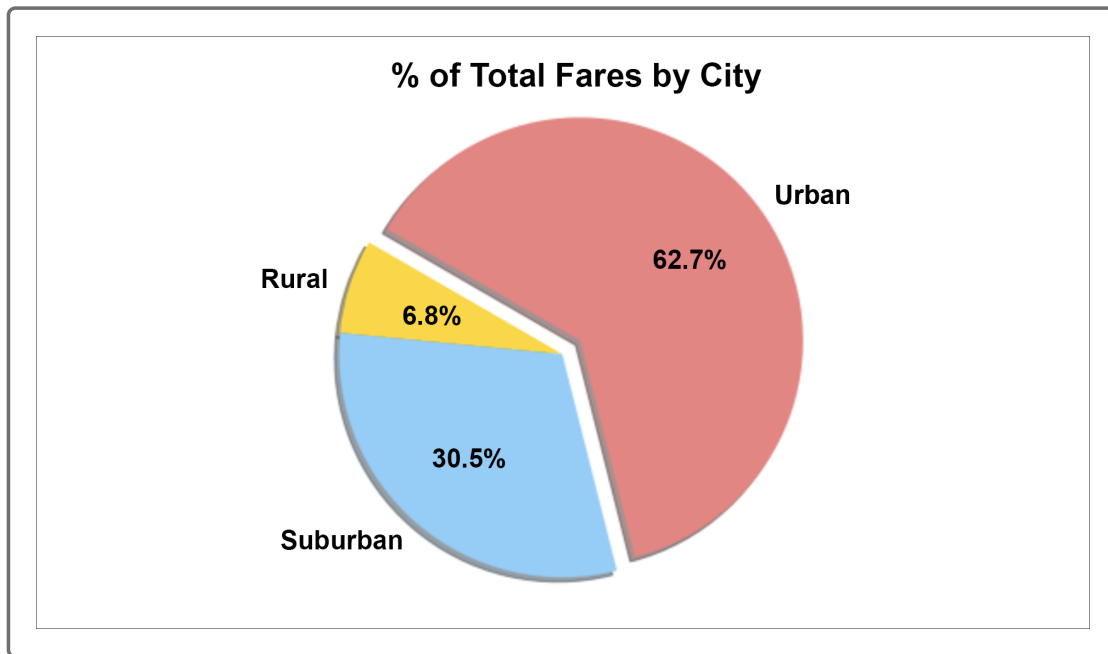
NOTE

> For more information, see the **Matplotlib documentation on customizing Matplotlib with style sheets and rcParams (https://matplotlib.org/stable/tutorials/introductory/customizing.html)** .

Now that you have created the first pie chart showing the percentage of total fares by city type, we need to create two more pie charts: the percentage of total rides by city type and the percentage of total drivers by city type.