Data Analysis

1. Reading Both the files

In [40]:

```
with open('HCICarDetails.txt') as f:
    content = f.readlines()
CarList = [x.strip('\n') for x in content]
```

In [31]:

```
with open('HCIQuestions.txt') as f:
    content = f.readlines()
Questions = [x.strip('\n') for x in content]
CarList[0] = CarList[0][1:]
```

In [183]:

```
CarList[:5]
```

Out[183]:

```
['1.  Maruti Suzuki Alto800[1,P,5,a,A]',
 '2.  Hyundai EON[1,P,5,a,A]',
 '3.  Maruti Suzuki Wagon-R[1,P,5,a,A]',
 '4.  Hyundai Grand i10[1,P,5,a,A]',
 '5.  Hyundai Grand i10[1,D,5,a,A]']
```

2. Extracting Data from Text Files

In [89]:

```
CarNames = []
CarFeatures = []
for line in CarList:
    name = ''
    for i,c in enumerate(line[4:]):
        if c == '[':
            break
        name = name+c
    CarNames.append(name)
    features = line[i+5:len(line)-1].split(',')
    CarFeatures.append(features)
CarNames = CarNames[:-1]
CarFeatures = CarFeatures[:-1]
CarFeature = pd.DataFrame(CarFeatures)
```

In [186]:

```
print(len(CarNames),len(CarFeatures))
print(CarNames[:5])
print(CarFeature.head())
```

```
89 89
['Maruti Suzuki Alto800', 'Hyundai EON', 'Maruti Suzuki Wagon-R', 'Hyundai Grand i10', 'Hyundai
Grand i10']
   0  1  2  3  4
0  1  P  5  a  A
1  1  P  5  a  A
2  1  P  5  a  A
3  1  P  5  a  A
4  1  D  5  a  A
```

3. Defining the Features

In [121]:

```
result = dict((i, BrandsList.count(i)) for i in BrandsList)
BrandCount = list(result.values())
Brands = list(result.keys())
print(Brands,BrandCount)
```

```
['Maruti', 'Hyundai', 'Tata', 'Volkswagen', 'Ford', 'Honda', 'Mahindra', 'Mercedes', 'BMW', 'Au
di', 'Jaguar'] [12, 14, 8, 3, 4, 10, 4, 12, 11, 7, 4]
```

In [176]:

```
Segments = ['Hatchback', 'Compact Sedan', 'Sedan', 'Mini-SUV', 'SUV', 'Luxury']
FuelType = ['Petrol', 'Diesel', 'Hybrid', 'Electric']
SeatingCapacity = ['5-seater', '7-seater', '8-seater', '9-seater']
UsageType = ['Family use', 'Office visits', 'Outings', 'Off-roading']
PriceRange = ['1,50,000-4,00,000', '4,00,000-10,50,000', '10,50,000-25,00,000' ,'25,00,000-80,00,000', '80,0
0,000+']
```

4. Plots of Features

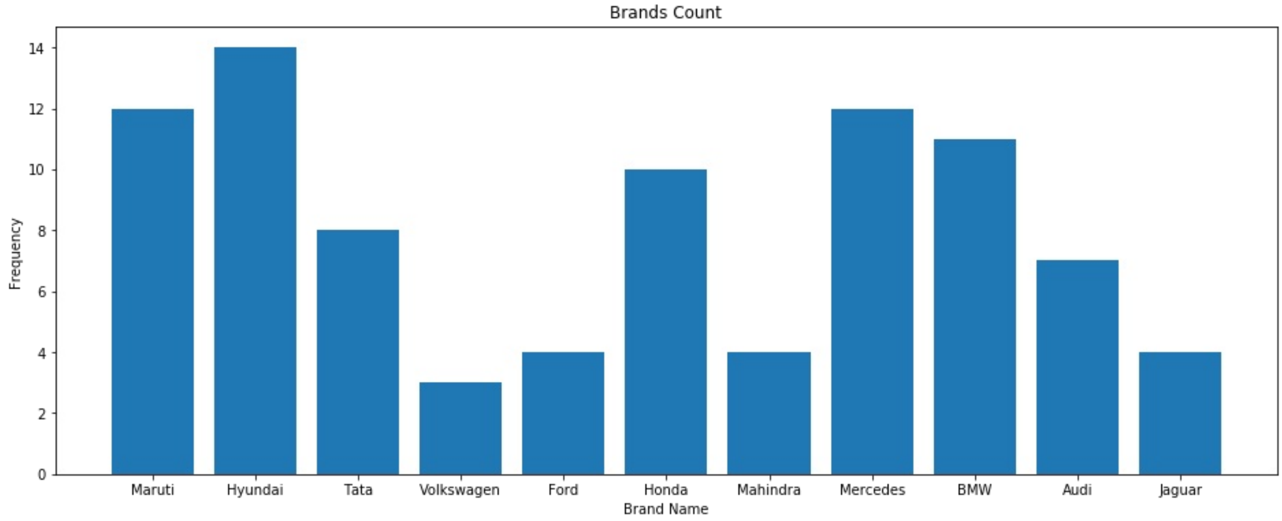
- Brands Analysis

In [187]:

```
plt.figure(figsize=(16,6))
plt.bar(Brands,BrandCount)
plt.title('Brands Count')
plt.xlabel('Brand Name')
plt.ylabel('Frequency')
```

Out[187]:

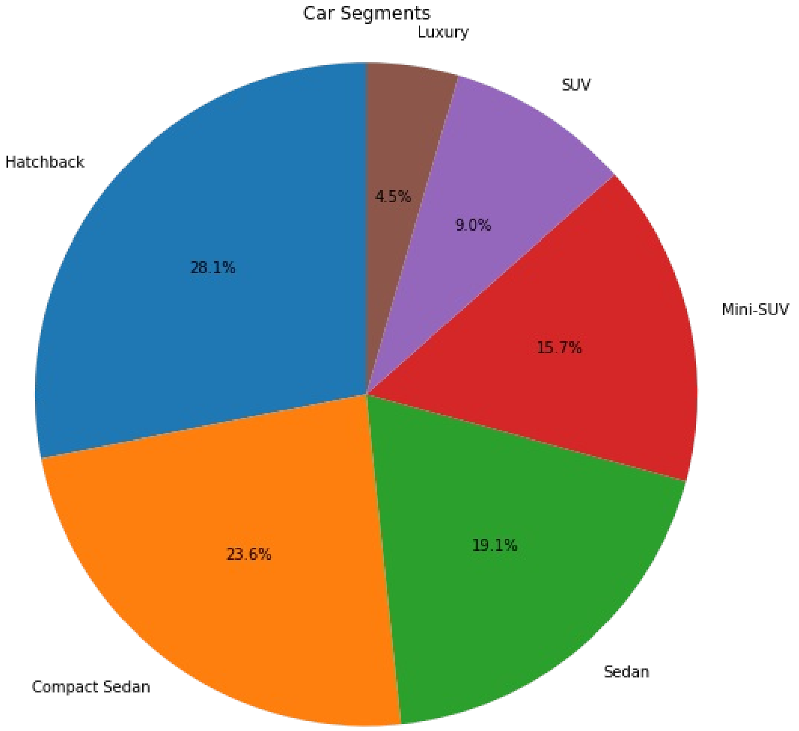
```
Text(0, 0.5, 'Frequency')
```



- Car Segments Analysis

In [188]:

```
SegmentCount = CarFeature[0].value_counts()
plt.figure(figsize=(15,9))
plt.pie(SegmentCount, labels=Segments, autopct='%1.1f%%',startangle=90)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Car Segments')
plt.show()
```



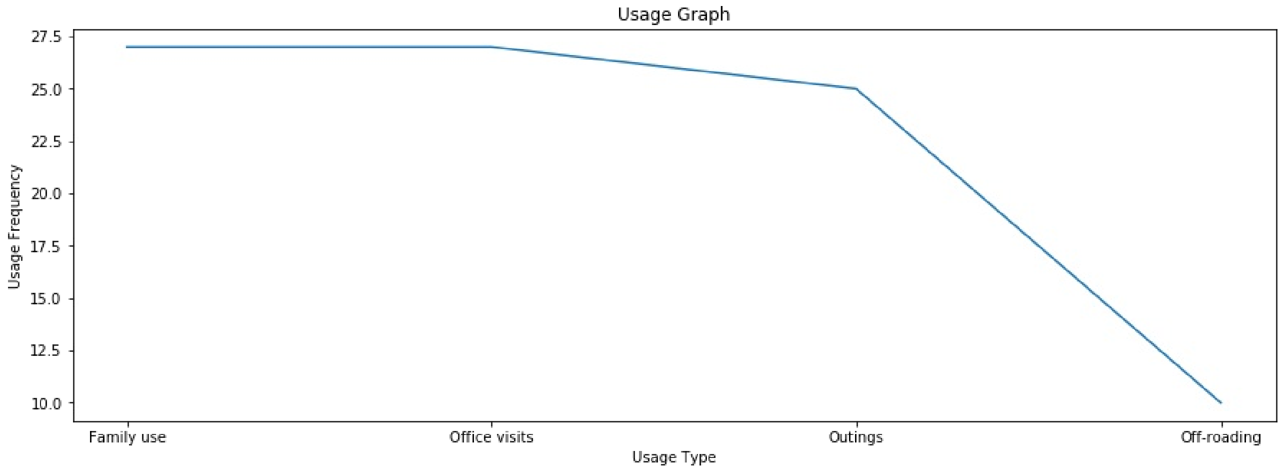
- Car Usage Analysis

In [189]:

```
UsageCount = CarFeature[3].value_counts()
plt.figure(figsize=(15,5))
plt.plot(UsageType,UsageCount)
plt.title('Usage Graph')
plt.xlabel('Usage Type')
plt.ylabel('Usage Frequency')
```

Out[189]:

```
Text(0, 0.5, 'Usage Frequency')
```



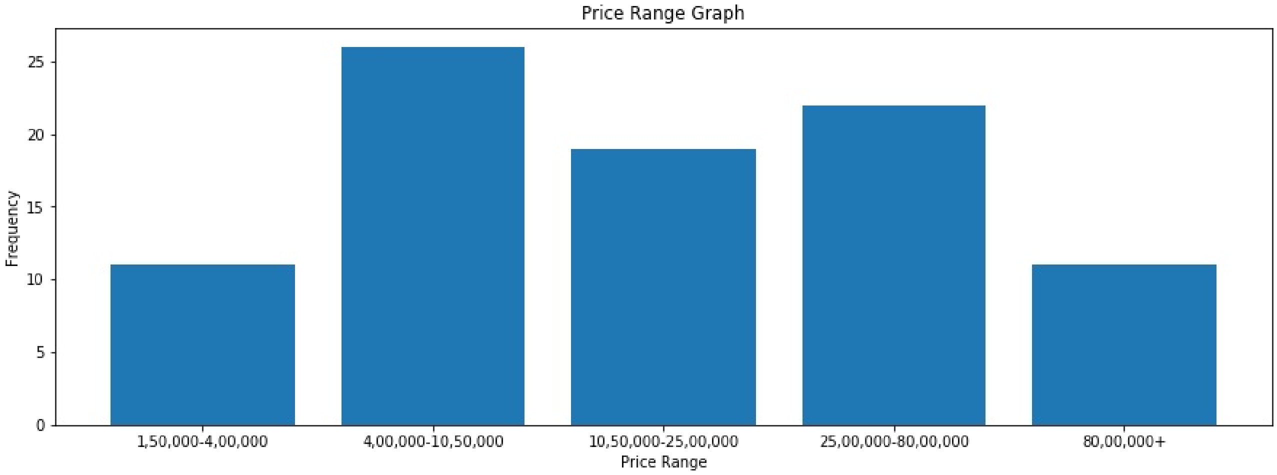
- Price Range Analysis

In [190]:

```
PriceCount = CarFeature[4].value_counts(sort=False)
PriceCount=PriceCount.sort_index()
plt.figure(figsize=(15,5))
plt.bar(PriceRange,PriceCount)
plt.title('Price Range Graph')
plt.xlabel('Price Range')
plt.ylabel('Frequency')
```

Out[190]:

```
Text(0, 0.5, 'Frequency')
```


Price Range Graph
Frequency
Price Range