In [126]:
```
import matplotlib.pyplot as plt  # import the module

import matplotlib as mpl
mpl.rcParams['figure.dpi']= 130  # set the resolution to x dpi
```

Let's read another dataset, reporting **data about mall customers,** and let's use this time `csv` methods (it will be easier than before!)

In [127]:
```
import csv

f = open('Mall_Customers.csv')

f_csv = csv.reader(f)
```

In [128]:
```
fieldnames = next(f_csv)
print(fieldnames)

print(len(fieldnames))
```

```
['CustomerID', 'Gender', 'Age', 'Annual Income (k$)', 'Spending Scor
e (1-100)']
5
```

In [129]:

```
gender = []
age = []
income  = []
score = []
customer_id = []


for record in f_csv:

        #customer_id.append(record.split(',')[0])

        print(record)

        customer_id.append(record[0])

        gender.append(record[1].strip()) # .strip() removes all extra
white spaces!

        age.append( int(record[2]) )
        income.append( int(record[3]) )
        score.append( int(record[4]) )
```

```
['1', ' Male', '  19', ' 15', ' 39']
['2', '  Male', ' 21', ' 15', ' 81']
['3', '   Female   ', '20', '16', '6']
['4', 'Female', '23  ', '16  ', '77']
['5', 'Female', '31', '17', '40']
['6', 'Female', '22', '17', '76']
['7', 'Female', '35', '18', '6']
['8', 'Female', '23', '18', '94']
['9', 'Male', '64', '19', '3']
['10', 'Female', '30', '19', '72']
['11', 'Male', '67', '19', '14']
['12', 'Female', '35', '19', '99']
['13', 'Female', '58', '20', '15']
['14', 'Female', '24', '20', '77']
['15', 'Male', '37', '20', '13']
['16', 'Male', '22', '20', '79']
['17', 'Female', '35', '21', '35']
['18', 'Male', '20', '21', '66']
['19', 'Male', '52', '23', '29']
['20', 'Female', '35', '23', '98']
['21', 'Male', '35', '24', '35']
['22', 'Male', '25', '24', '73']
['23', 'Female', '46', '25', '5']
['24', 'Male', '31', '25', '73']
['25', 'Female', '54', '28', '14']
['26', 'Male', '29', '28', '82']
['27', 'Female', '45', '28', '32']
['28', 'Male', '35', '28', '61']
['29', 'Female', '40', '29', '31']
['30', 'Female', '23', '29', '87']
['31', 'Male', '60', '30', '4']
['32', 'Female', '21', '30', '73']
['33', 'Male', '53', '33', '4']
['34', 'Male', '18', '33', '92']
['35', 'Female', '49', '33', '14']
['36', 'Female', '21', '33', '81']
['37', 'Female', '42', '34', '17']
['38', 'Female', '30', '34', '73']
['39', 'Female', '36', '37', '26']
['40', 'Female', '20', '37', '75']
['41', 'Female', '65', '38', '35']
['42', 'Male', '24', '38', '92']
['43', 'Male', '48', '39', '36']
['44', 'Female', '31', '39', '61']
['45', 'Female', '49', '39', '28']
['46', 'Female', '24', '39', '65']
['47', 'Female', '50', '40', '55']
['48', 'Female', '27', '40', '47']
['49', 'Female', '29', '40', '42']
['50', 'Female', '31', '40', '42']
['51', 'Female', '49', '42', '52']
['52', 'Male', '33', '42', '60']
['53', 'Female', '31', '43', '54']
['54', 'Male', '59', '43', '60']
['55', 'Female', '50', '43', '45']
['56', 'Male', '47', '43', '41']
['57', 'Female', '51', '44', '50']
```

```
['58', 'Male', '69', '44', '46']
['59', 'Female', '27', '46', '51']
['60', 'Male', '53', '46', '46']
['61', 'Male', '70', '46', '56']
['62', 'Male', '19', '46', '55']
['63', 'Female', '67', '47', '52']
['64', 'Female', '54', '47', '59']
['65', 'Male', '63', '48', '51']
['66', 'Male', '18', '48', '59']
['67', 'Female', '43', '48', '50']
['68', 'Female', '68', '48', '48']
['69', 'Male', '19', '48', '59']
['70', 'Female', '32', '48', '47']
['71', 'Male', '70', '49', '55']
['72', 'Female', '47', '49', '42']
['73', 'Female', '60', '50', '49']
['74', 'Female', '60', '50', '56']
['75', 'Male', '59', '54', '47']
['76', 'Male', '26', '54', '54']
['77', 'Female', '45', '54', '53']
['78', 'Male', '40', '54', '48']
['79', 'Female', '23', '54', '52']
['80', 'Female', '49', '54', '42']
['81', 'Male', '57', '54', '51']
['82', 'Male', '38', '54', '55']
['83', 'Male', '67', '54', '41']
['84', 'Female', '46', '54', '44']
['85', 'Female', '21', '54', '57']
['86', 'Male', '48', '54', '46']
['87', 'Female', '55', '57', '58']
['88', 'Female', '22', '57', '55']
['89', 'Female', '34', '58', '60']
['90', 'Female', '50', '58', '46']
['91', 'Female', '68', '59', '55']
['92', 'Male', '18', '59', '41']
['93', 'Male', '48', '60', '49']
['94', 'Female', '40', '60', '40']
['95', 'Female', '32', '60', '42']
['96', 'Male', '24', '60', '52']
['97', 'Female', '47', '60', '47']
['98', 'Female', '27', '60', '50']
['99', 'Male', '48', '61', '42']
['100', 'Male', '20', '61', '49']
['101', 'Female', '23', '62', '41']
['102', 'Female', '49', '62', '48']
['103', 'Male', '67', '62', '59']
['104', 'Male', '26', '62', '55']
['105', 'Male', '49', '62', '56']
['106', 'Female', '21', '62', '42']
['107', 'Female', '66', '63', '50']
['108', 'Male', '54', '63', '46']
['109', 'Male', '68', '63', '43']
['110', 'Male', '66', '63', '48']
['111', 'Male', '65', '63', '52']
['112', 'Female', '19', '63', '54']
['113', 'Female', '38', '64', '42']
['114', 'Male', '19', '64', '46']
['115', 'Female', '18', '65', '48']
```

['116', 'Female', '19', '65', '50']
['117', 'Female', '63', '65', '43']
['118', 'Female', '49', '65', '59']
['119', 'Female', '51', '67', '43']
['120', 'Female', '50', '67', '57']
['121', 'Male', '27', '67', '56']
['122', 'Female', '38', '67', '40']
['123', 'Female', '40', '69', '58']
['124', 'Male', '39', '69', '91']
['125', 'Female', '23', '70', '29']
['126', 'Female', '31', '70', '77']
['127', 'Male', '43', '71', '35']
['128', 'Male', '40', '71', '95']
['129', 'Male', '59', '71', '11']
['130', 'Male', '38', '71', '75']
['131', 'Male', '47', '71', '9']
['132', 'Male', '39', '71', '75']
['133', 'Female', '25', '72', '34']
['134', 'Female', '31', '72', '71']
['135', 'Male', '20', '73', '5']
['136', 'Female', '29', '73', '88']
['137', 'Female', '44', '73', '7']
['138', 'Male', '32', '73', '73']
['139', 'Male', '19', '74', '10']
['140', 'Female', '35', '74', '72']
['141', 'Female', '57', '75', '5']
['142', 'Male', '32', '75', '93']
['143', 'Female', '28', '76', '40']
['144', 'Female', '32', '76', '87']
['145', 'Male', '25', '77', '12']
['146', 'Male', '28', '77', '97']
['147', 'Male', '48', '77', '36']
['148', 'Female', '32', '77', '74']
['149', 'Female', '34', '78', '22']
['150', 'Male', '34', '78', '90']
['151', 'Male', '43', '78', '17']
['152', 'Male', '39', '78', '88']
['153', 'Female', '44', '78', '20']
['154', 'Female', '38', '78', '76']
['155', 'Female', '47', '78', '16']
['156', 'Female', '27', '78', '89']
['157', 'Male', '37', '78', '1']
['158', 'Female', '30', '78', '78']
['159', 'Male', '34', '78', '1']
['160', 'Female', '30', '78', '73']
['161', 'Female', '56', '79', '35']
['162', 'Female', '29', '79', '83']
['163', 'Male', '19', '81', '5']
['164', 'Female', '31', '81', '93']
['165', 'Male', '50', '85', '26']
['166', 'Female', '36', '85', '75']
['167', 'Male', '42', '86', '20']
['168', 'Female', '33', '86', '95']
['169', 'Female', '36', '87', '27']
['170', 'Male', '32', '87', '63']
['171', 'Male', '40', '87', '13']
['172', 'Male', '28', '87', '75']
['173', 'Male', '36', '87', '10']

['174', 'Male', '36', '87', '92']
['175', 'Female', '52', '88', '13']
['176', 'Female', '30', '88', '86']
['177', 'Male', '58', '88', '15']
['178', 'Male', '27', '88', '69']
['179', 'Male', '59', '93', '14']
['180', 'Male', '35', '93', '90']
['181', 'Female', '37', '97', '32']
['182', 'Female', '32', '97', '86']
['183', 'Male', '46', '98', '15']
['184', 'Female', '29', '98', '88']
['185', 'Female', '41', '99', '39']
['186', 'Male', '30', '99', '97']
['187', 'Female', '54', '101', '24']
['188', 'Male', '28', '101', '68']
['189', 'Female', '41', '103', '17']
['190', 'Female', '36', '103', '85']
['191', 'Female', '34', '103', '23']
['192', 'Female', '32', '103', '69']
['193', 'Male', '33', '113', '8']
['194', 'Female', '38', '113', '91']
['195', 'Female', '47', '120', '16']
['196', 'Female', '35', '120', '79']
['197', 'Female', '45', '126', '28']
['198', 'Male', '32', '126', '74']
['199', 'Male', '32', '137', '18']
['200', 'Male', '30', '137', '83']

In [130]:
```
age
```

Out[130]: [19,
21,
20,
23,
31,
22,
35,
23,
64,
30,
67,
35,
58,
24,
37,
22,
35,
20,
52,
35,
35,
25,
46,
31,
54,
29,
45,
35,
40,
23,
60,
21,
53,
18,
49,
21,
42,
30,
36,
20,
65,
24,
48,
31,
49,
24,
50,
27,
29,
31,
49,
33,
31,
59,
50,
47,
51,

69,
27,
53,
70,
19,
67,
54,
63,
18,
43,
68,
19,
32,
70,
47,
60,
60,
59,
26,
45,
40,
23,
49,
57,
38,
67,
46,
21,
48,
55,
22,
34,
50,
68,
18,
48,
40,
32,
24,
47,
27,
48,
20,
23,
49,
67,
26,
49,
21,
66,
54,
68,
66,
65,
19,
38,
19,
18,

19,
63,
49,
51,
50,
27,
38,
40,
39,
23,
31,
43,
40,
59,
38,
47,
39,
25,
31,
20,
29,
44,
32,
19,
35,
57,
32,
28,
32,
25,
28,
48,
32,
34,
34,
43,
39,
44,
38,
47,
27,
37,
30,
34,
30,
56,
29,
19,
31,
50,
36,
42,
33,
36,
32,
40,
28,
36,

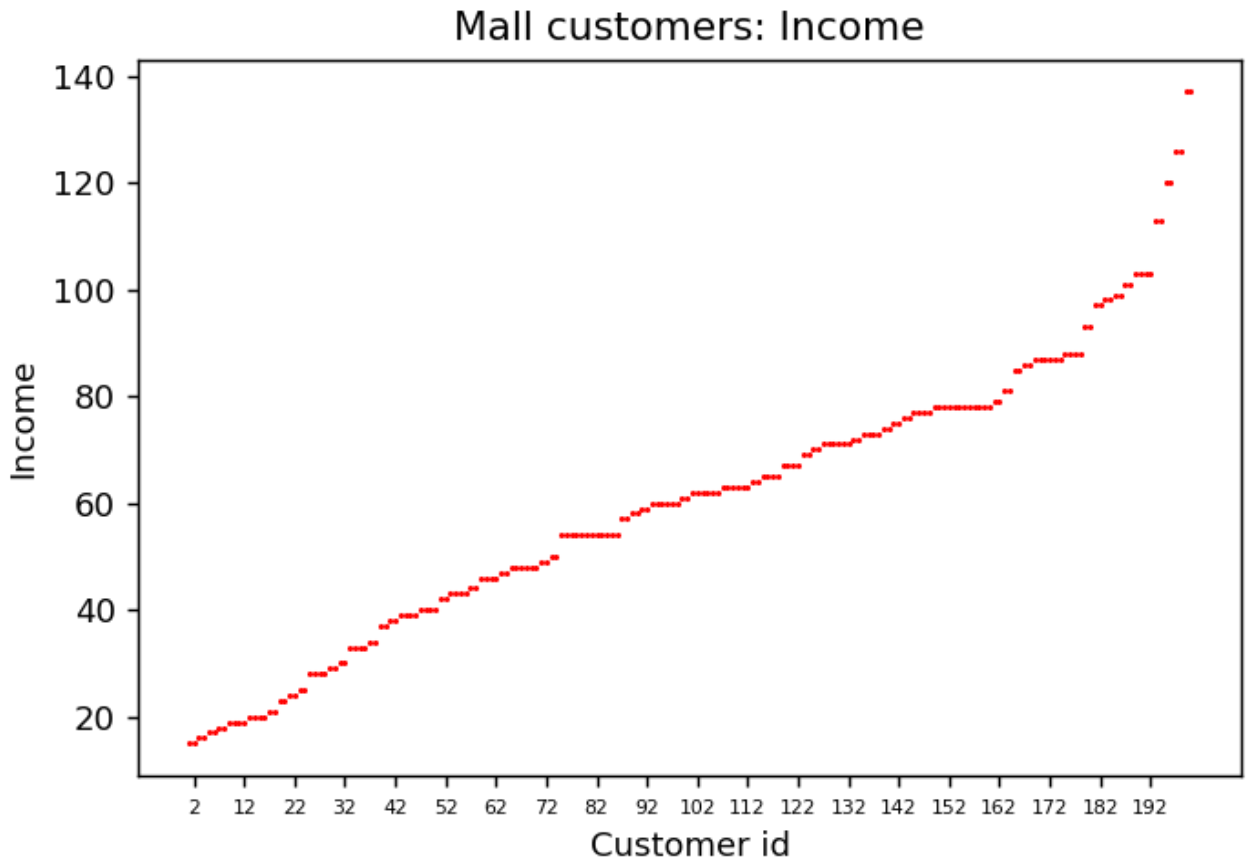
```
36,
52,
30,
58,
27,
59,
35,
37,
32,
46,
29,
41,
30,
54,
28,
41,
36,
34,
32,
33,
38,
47,
35,
45,
32,
32,
30]
```

In [131]:
```
#gender
```

This is **not a time series** of data!

What about the distribution of the values of income?

In [132]:
```
plt.title("Mall customers: Income")

plt.xlabel("Customer id")

plt.ylabel("Income")

plt.scatter(customer_id, income, marker='o', s=0.5, color='red')

plt.xticks(customer_id[1::10], fontsize=6)

plt.show()
```



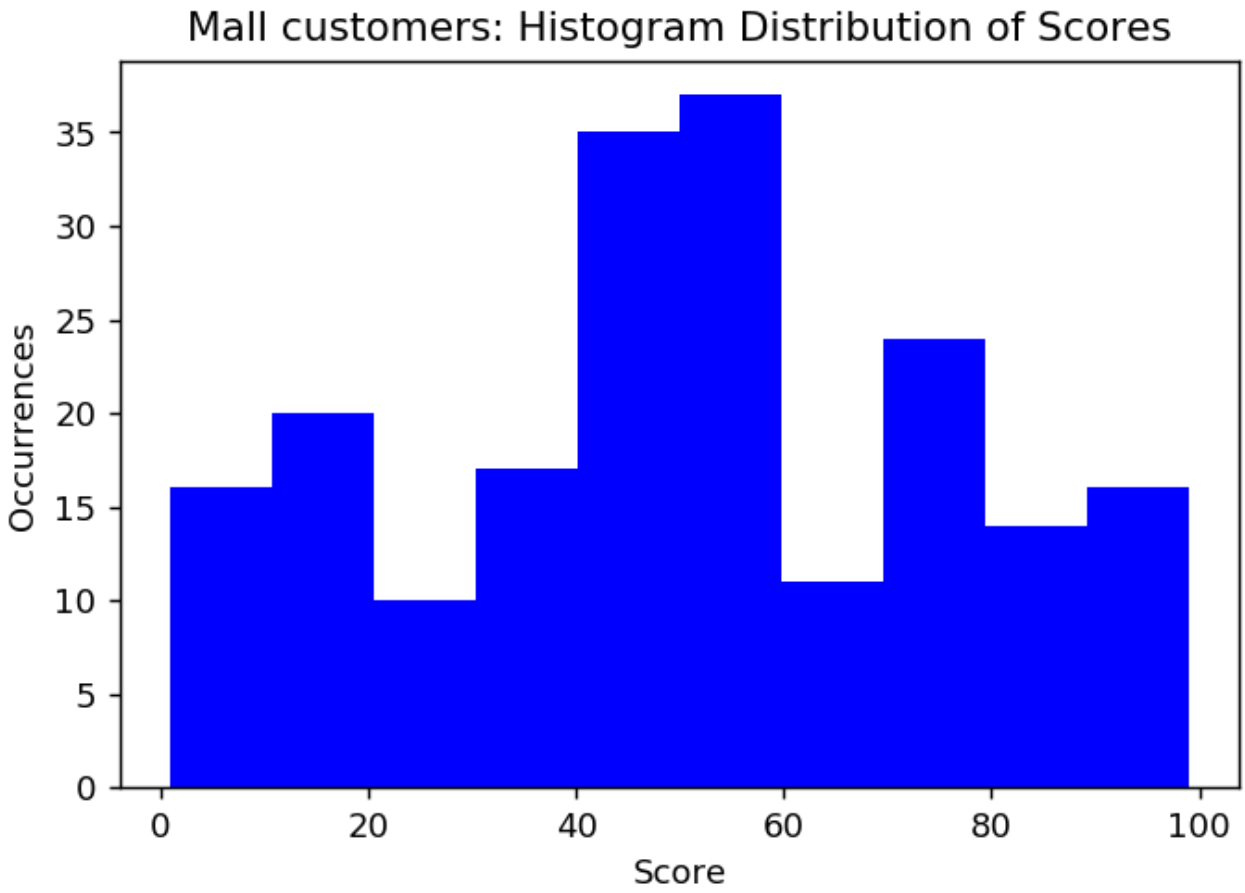
It looks like the income grows with the customer id! (values have been sorted vs. the income)

What about the shopping scores?

In [133]:
```
plt.title("Mall customers: Scores")

plt.xlabel("Customer id")

plt.ylabel("Score")

plt.scatter(customer_id, score, marker='o', s=0.5, color='blue')

plt.xticks(customer_id[1::10], fontsize=6)

plt.show()
```

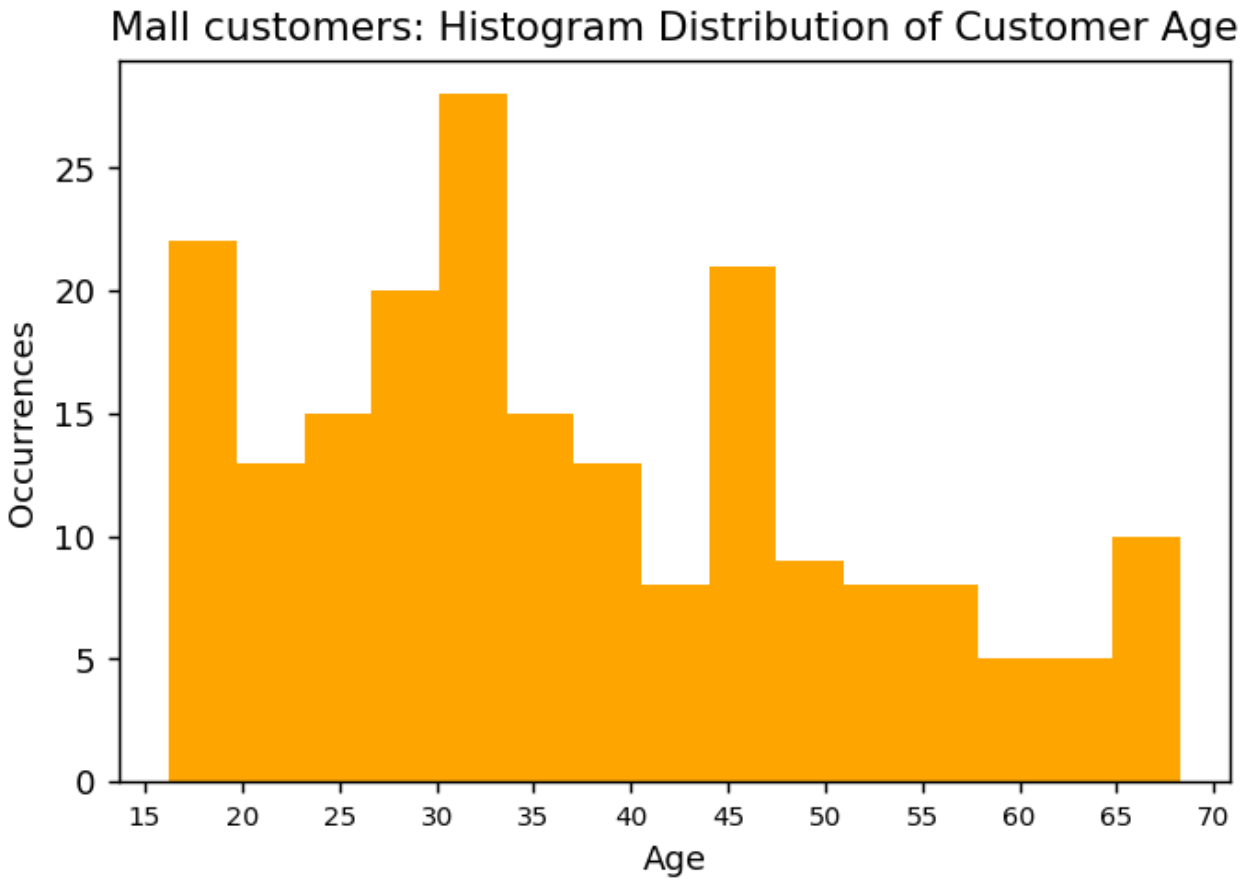
Aggregating the data to check what the **distribution** of the scores look like in the customer population can be useful → **Histogram**

In [134]:
```
plt.title("Mall customers: Histogram Distribution of Scores")

plt.xlabel("Score")

plt.ylabel("Occurrences")

plt.hist(score, bins=10, color='blue')

#plt.xticks(customer_id[1::10], fontsize=6)

plt.show()
```



What about the distribution of the age of the customers?

In [135]:
```
plt.title("Mall customers: Histogram Distribution of Customer Age")

plt.xlabel("Age")

plt.ylabel("Occurrences")

plt.hist(age, bins=15, align='left',
         color='orange')

plt.xticks(range(15, 75, 5), fontsize=8)

plt.show()
```



What about the distribution of male vs. female?

In this case we have two values: number of male and number of female customers.

We need to count them from the `gender` list!

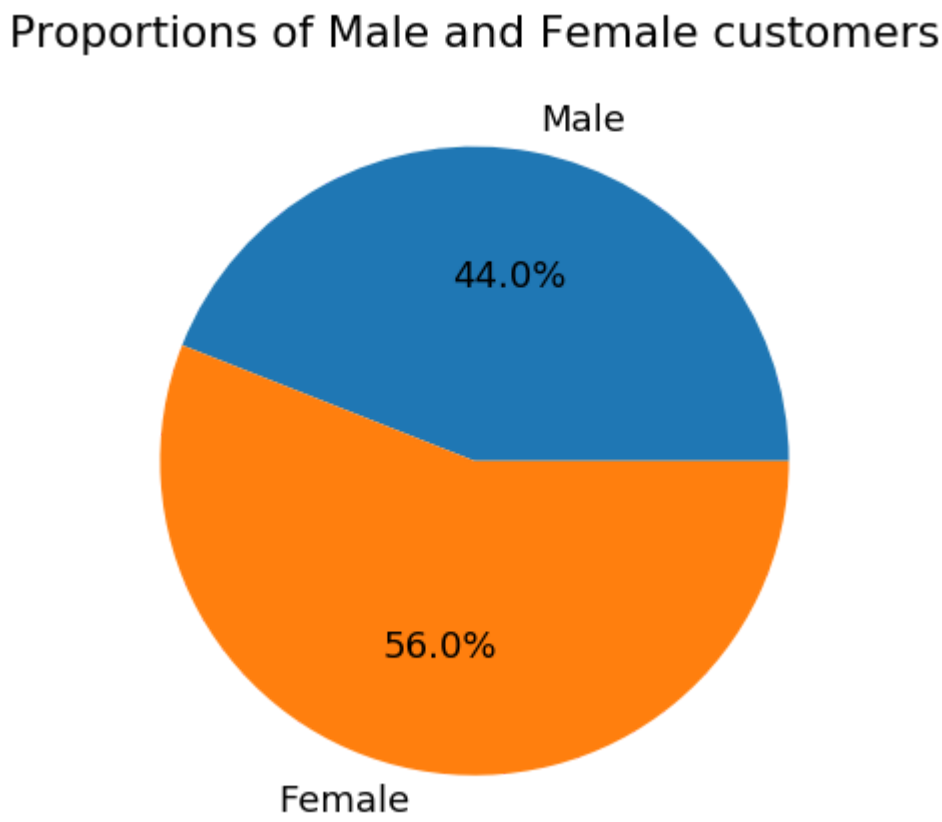
In [136]:
```
n_male = gender.count('Male')

n_female = gender.count('Female')

n_male, n_female
```

Out[136]:
```
(88, 112)
```

How do we effectively show these proportions? (a histogram is not really appropriate for showing proportions)

- **Pie chart**
- **Bar chart**

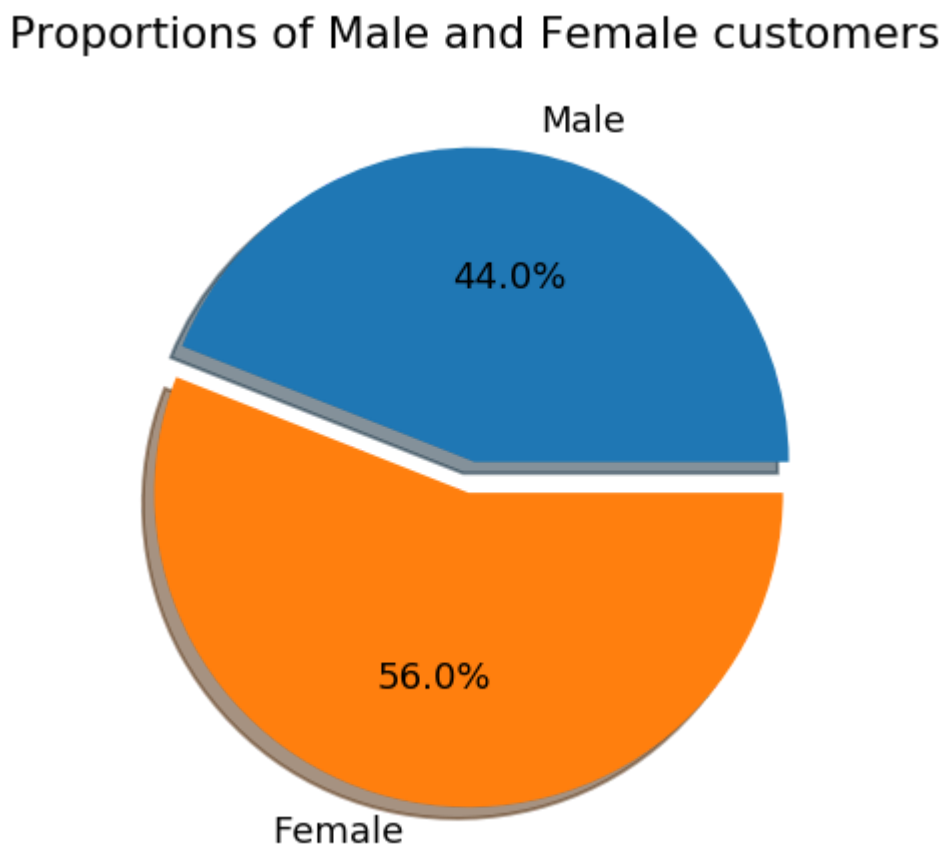
In [137]:
```
plt.title("Proportions of Male and Female customers")

plt.pie([n_male, n_female],
        labels = ['Male', 'Female'],
        autopct="%.1f%%",  # this says to use one decimal digit and t
o use percentages
        textprops={'fontsize': 10})  # this can be used to set proper
ties of text labels

plt.show()
```



Some beautifying of the pie chart.

`shadow` adds a shadowing, while `explode` is tuple, where if the i-th value is greater than 0, it indcates that the i-th slice of the pie will be detached from the rest of the pie (at a distance proportional to the indicated value).

In [138]:
```
plt.title("Proportions of Male and Female customers")

plt.pie([n_male, n_female],
        labels = ['Male', 'Female'], shadow=True, explode = (0, 0.1),
        autopct="%.1f%%")

plt.show()
```



Can we make a pie chart for the customer age?

Yes, but we need to define age ranges, group / count the data accordingly, and store/define explanatory labels.

Let's consider 10-year ranges, starting from the minimal age up to the maximal age in the dataset.

We'll use a dictionary data structure.

In [139]:
```
import math

min_age = min(age)
max_age = max(age)

age_interval = 10

age_ranges = (max_age - min_age) / age_interval

age_ranges = math.ceil(age_ranges)
print(min_age, max_age, age_ranges)
```

```
18 70 6
```

In [140]:
```
age_dict = {}

for r in range(age_ranges):
        range_min = min_age + r * age_interval
        range_max = range_min + age_interval - 1
        range_str = str(range_min) + '-' + str(range_max)
        age_dict[ range_str ] = [range_min, range_max, 0]
```

In [141]:
```
age_dict
```

Out[141]:
```
{'18-27': [18, 27, 0],
 '28-37': [28, 37, 0],
 '38-47': [38, 47, 0],
 '48-57': [48, 57, 0],
 '58-67': [58, 67, 0],
 '68-77': [68, 77, 0]}
```

In [142]:
```
for v in age:
    for r in age_dict:
        if v >= age_dict[r][0] and v <= age_dict[r][1]:
            age_dict[r][2] += 1
            break
```

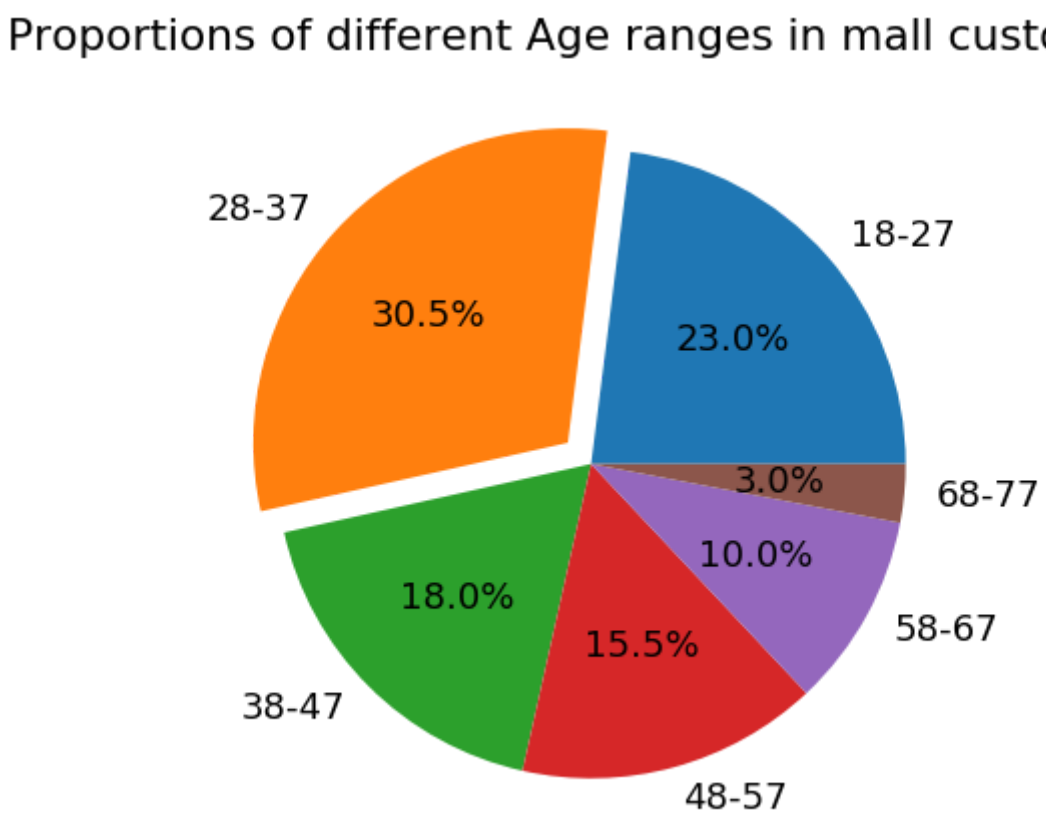
In [143]:
```
age_dict
```

Out[143]:
```
{'18-27': [18, 27, 46],
 '28-37': [28, 37, 61],
 '38-47': [38, 47, 36],
 '48-57': [48, 57, 31],
 '58-67': [58, 67, 20],
 '68-77': [68, 77, 6]}
```

Now we can show the counts in the selected ranges of age as proportions using a pie chart.

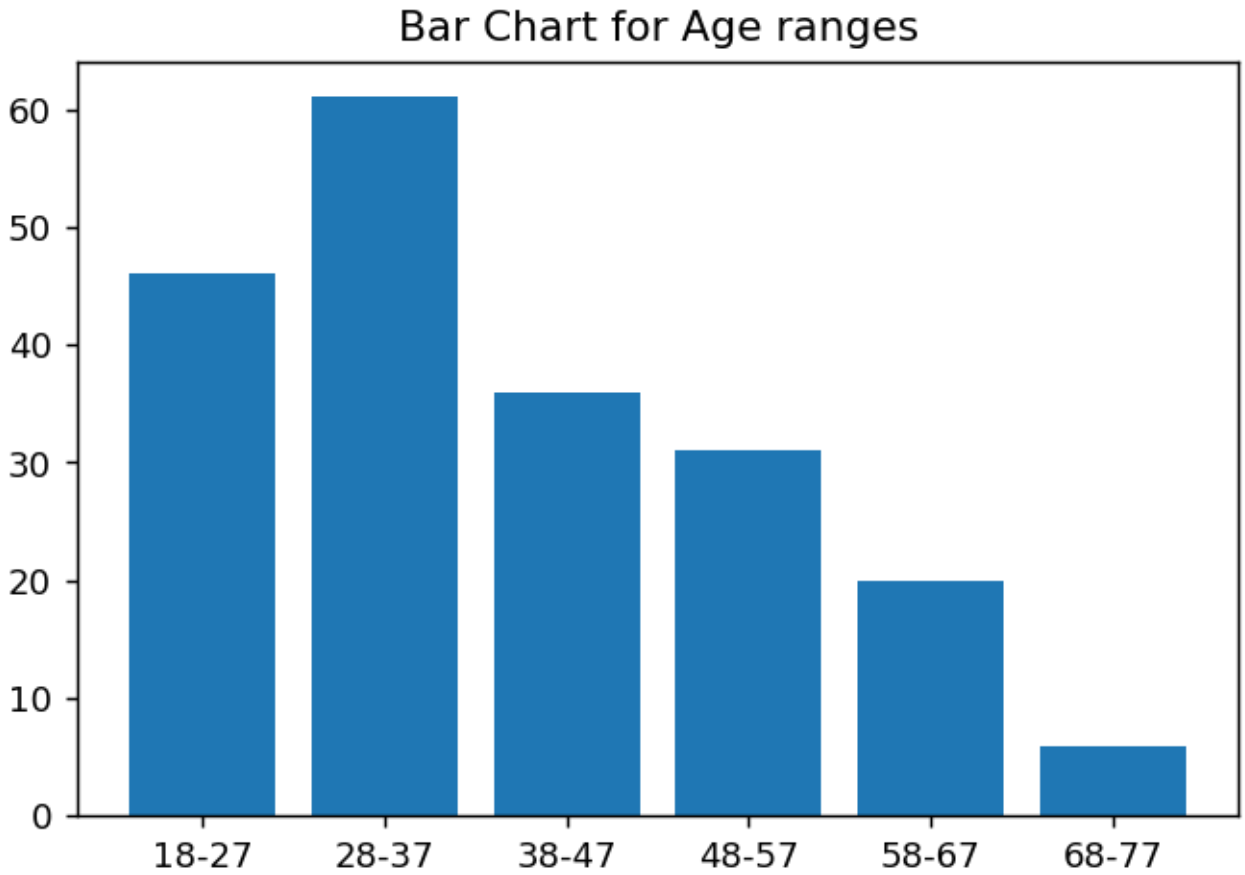
Counts are stored in the list `age_counts`

In [144]:
```
plt.title("Proportions of different Age ranges in mall customers")

age_counts = []
for r in age_dict.values():
    age_counts.append(r[2])

# this is to identify the slice with the largest proportion to explod
e it
max_range = -1
max_range_idx = -1
for i,r in enumerate(age_dict.values()):
    if r[2] > max_range:
        max_range = r[2]
        max_range_idx = i

explode_flag = [0] * len(age_dict)
explode_flag[max_range_idx] = 0.1


plt.pie(age_counts,
        labels = list(age_dict.keys()),
        #shadow=True,
        explode = explode_flag,
        autopct="%.1f%%",
        textprops={'fontsize': 10})

plt.show()
```



We can display the same data using a **bar chart.**

In [145]:
```
plt.title("Bar Chart for Age ranges")

plt.bar(range(1, len(age_counts)+1),          # position of the bars
        age_counts,                           # value/height of the ba
rs
        tick_label = list(age_dict.keys()))  # what to display at the
x ticks

plt.show()
```



Let's read a more interesting file and use `csv` methods!

In [146]:
```
import csv
f = open('zoological_attributes.csv')

f_csv = csv.DictReader(f)
```

In [147]:
```
f_csv.fieldnames
```

Out[147]:
```
['name',
 'hair',
 'feathers',
 'eggs',
 'milk',
 'airborne',
 'aquatic',
 'predator',
 'toothed',
 'backbone',
 'breathes',
 'venomous',
 'fins',
 'legs',
 'tail',
 'domestic',
 'catsize',
 'type']
```

In [148]:
```
len(f_csv.fieldnames)
```

Out[148]:
```
18
```

In [149]:
```
properties = dict.fromkeys(f_csv.fieldnames, 0)
```

In [150]:
```
properties
```

Out[150]:
```
{'name': 0,
 'hair': 0,
 'feathers': 0,
 'eggs': 0,
 'milk': 0,
 'airborne': 0,
 'aquatic': 0,
 'predator': 0,
 'toothed': 0,
 'backbone': 0,
 'breathes': 0,
 'venomous': 0,
 'fins': 0,
 'legs': 0,
 'tail': 0,
 'domestic': 0,
 'catsize': 0,
 'type': 0}
```

In [151]:
```
properties.keys()
```

Out[151]:
```
dict_keys(['name', 'hair', 'feathers', 'eggs', 'milk', 'airborne',
'aquatic', 'predator', 'toothed', 'backbone', 'breathes', 'venomou
s', 'fins', 'legs', 'tail', 'domestic', 'catsize', 'type'])
```

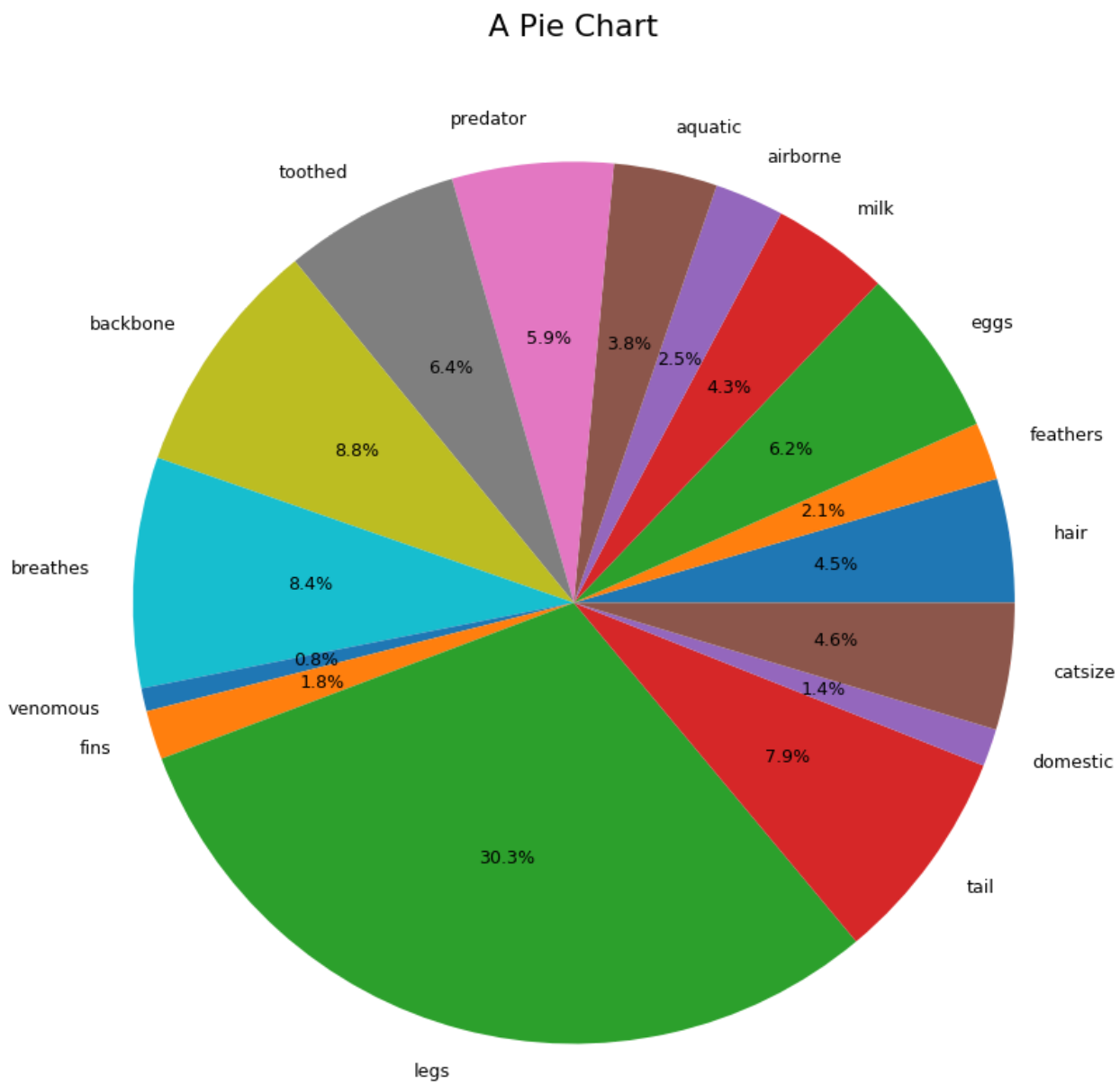
In [152]:
```
f.seek(0)

next(f_csv)

for a in f_csv:
    for p in list(properties.keys())[1:]:
        #print(p, a)
        properties[p] += int(a[p])
```

In [153]:
```
properties
```

Out[153]:
```
{'name': 0,
 'hair': 43,
 'feathers': 20,
 'eggs': 59,
 'milk': 41,
 'airborne': 24,
 'aquatic': 36,
 'predator': 56,
 'toothed': 61,
 'backbone': 83,
 'breathes': 80,
 'venomous': 8,
 'fins': 17,
 'legs': 287,
 'tail': 75,
 'domestic': 13,
 'catsize': 44,
 'type': 286}
```

In [154]:

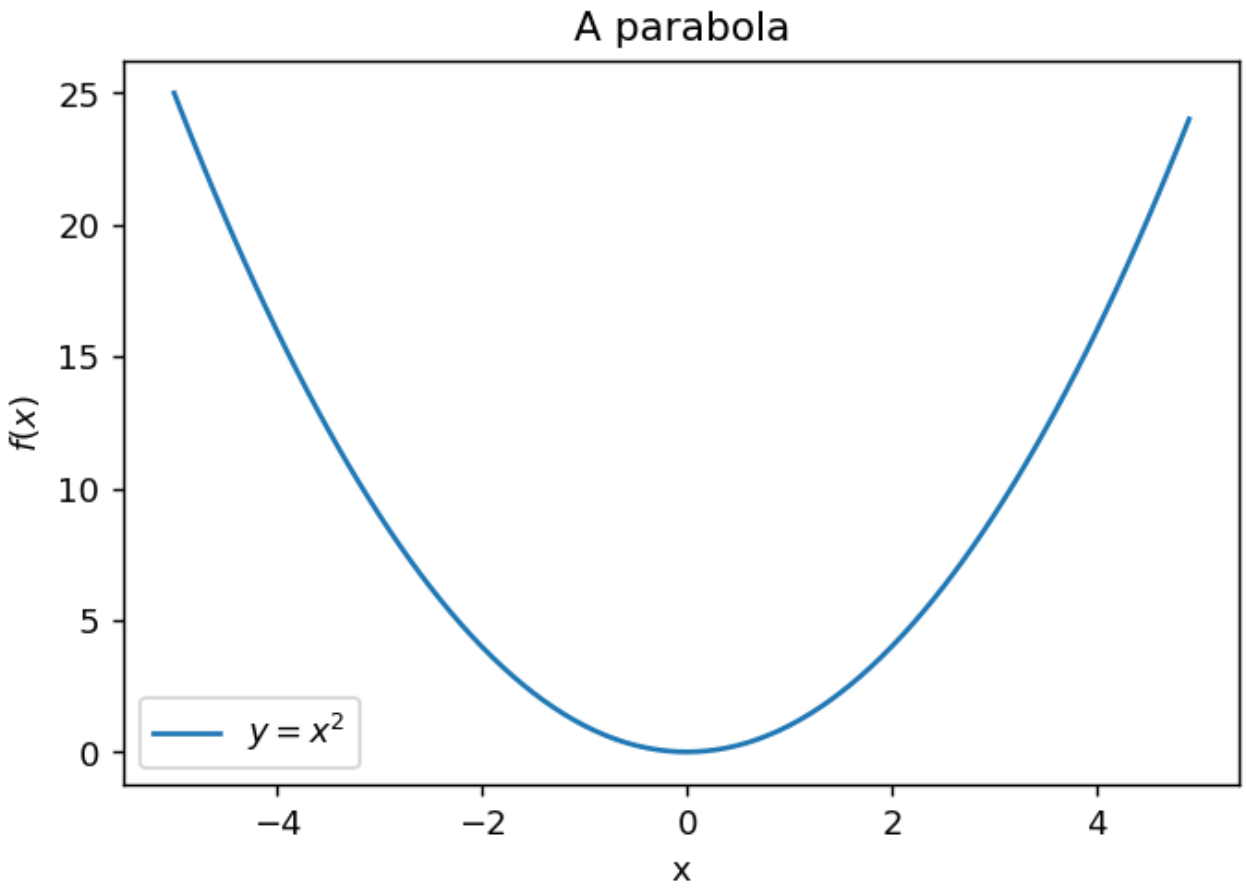
```
xsize = 8
ysize = 8
plt.figure(figsize=(xsize, ysize))
# Make the chart a circle, otherwise it'll be an ellipsis fir xsize d
ifferent from ysize

plt.title("A Pie Chart")

plt.pie(list(properties.values())[1:-1],
        labels=list(properties.keys())[1:-1],
        autopct="%.1f%%",
        textprops={'fontsize': 7})

plt.show()
```



Some more interesting data? What about *nicely* plotting a function?

In [157]:
```
import numpy as np

x = np.arange(-5, 5, 0.1)

y = []
for xi in x:
    y.append(xi * xi)
```

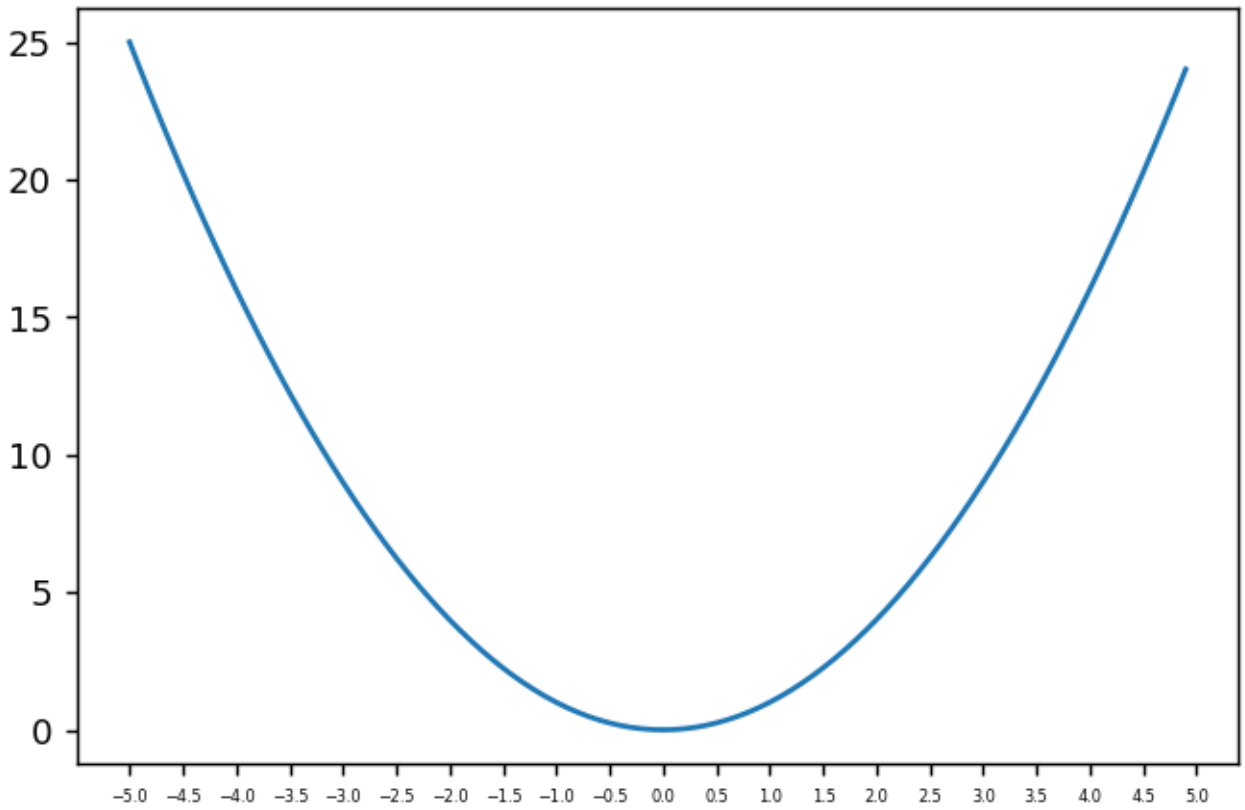
In [158]:
```
plt.title("A parabola")

plt.xlabel("x")
plt.ylabel("$f(x)$")


plt.plot(x, y,
         label='$y = x^2$')

plt.legend()

plt.show()
```



Can I control **xticks** and **yticks**?

In [159]:
```
plt.xticks(np.arange(-5, 5.5, 0.5), fontsize=4.5)

plt.plot(x,y)
```

Out[159]:
```
[<matplotlib.lines.Line2D at 0x7fdf0604b710>]
```



Anothe code example reading one of the datasets suggested for the project

In [160]:

```
import csv

f = open('Car_sales.csv')
csv_data = csv.DictReader(f)
next(csv_data)

unique = {}

for row in csv_data:
    if row['Fuel_efficiency'] not in unique:
        unique[row['Fuel_efficiency']] = 1
    else:
        unique[row['Fuel_efficiency']] += 1

print(unique)
print('Total:', sum(unique.values()))
```

```
{'25': 23, '26': 12, '22': 14, '27': 15, '21': 14, '24': 16, '23': 1
4, '15': 5, '33': 4, '45': 1, '': 3, '29': 2, '16': 3, '17': 3, '1
9': 6, '30': 5, '18': 5, '32': 1, '31': 3, '20': 5, '28': 2}
Total: 156
```