In [2]:
```python
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
%matplotlib inline
```

In [3]:
```python
mo_exp = pd.read_csv('Monthly_exp.csv', sep=',',
                     encoding='ISO-8859-1')
```

In [4]:
```python
mo_exp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   ï»¿ID         310 non-null    object 
 1   Description   310 non-null    object 
 2   Category      310 non-null    object 
 3   Amount (CAD)  310 non-null    float64
 4   Month         310 non-null    object 
 5   Year          310 non-null    int64  
dtypes: float64(1), int64(1), object(4)
memory usage: 14.7+ KB
```

In [5]:
```python
mo_exp.describe()
```

Out[5]:

	Amount (CAD)	Year
count	310.000000	310.000000
mean	63.069000	2021.796774
std	89.421176	0.540277
min	1.560000	2021.000000
25%	14.532500	2021.000000
50%	31.355000	2022.000000
75%	73.857500	2022.000000
max	721.930000	2023.000000

In [6]:
```python
print(mo_exp.columns.tolist())
```

```
['ï»¿ID', 'Description', 'Category', 'Amount (CAD)', 'Month', 'Year']
```

In [7]:
```python
mo_exp.head()
```

Out[7]:

	ï»¿ID	Description	Category	Amount (CAD)	Month	Year
0	EXP001	88 Supermarket	Groceries	82.76	21-Jul	2021
1	EXP002	Ikea	Others	14.22	21-Jul	2021
2	EXP003	Poke Bar	Dine out	19.91	21-Jul	2021
3	EXP004	Donair Dude	Dine out	10.91	21-Jul	2021
4	EXP005	Safeway	Groceries	30.46	21-Jul	2021

In [8]:
```python
# total expenses

total_exp = mo_exp['Amount (CAD)'].sum()
print(total_exp)
```

```
19551.39
```

In [9]:
```python
# number of month

number_of_months = np.size((mo_exp['Month'].unique()))
print(number_of_months)
```

```
20
```

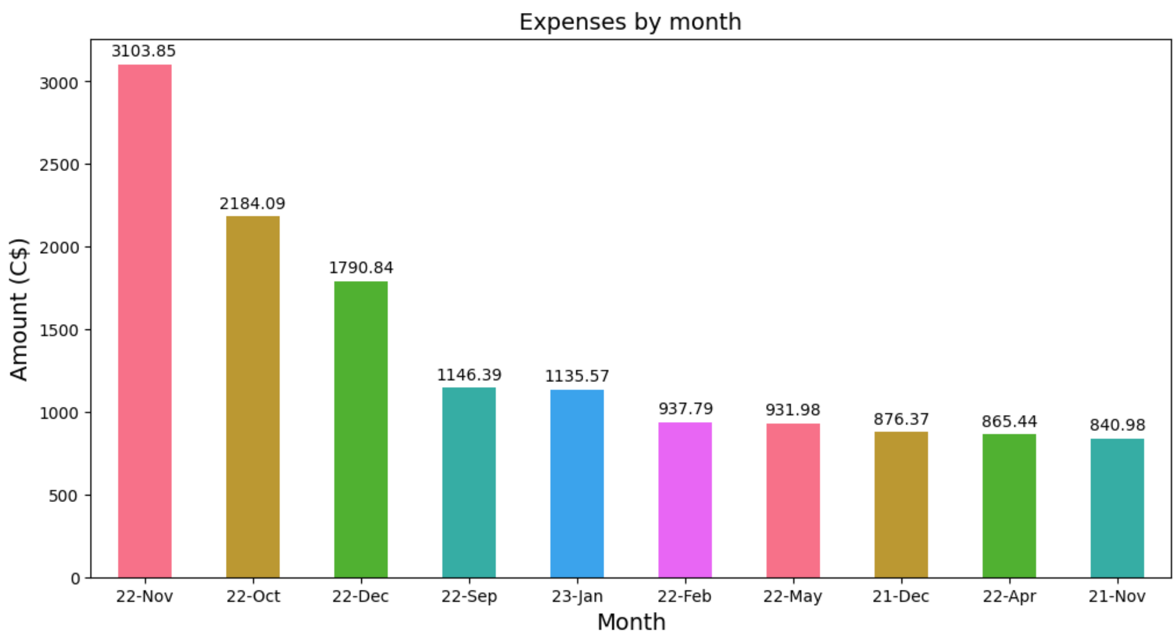
In [10]:
```python
# average expenses

avg_exp_per_month = total_exp / number_of_months
print(avg_exp_per_month)
```

```
977.5695
```

In [11]:
```python
# expenses by month, descending

monthly_exp = mo_exp.groupby('Month')['Amount (CAD)'].sum()
monthly_exp.sort_values(ascending=False)
```

Out[11]:
```
Month
22-Nov    3103.85
22-Oct    2184.09
22-Dec    1790.84
22-Sep    1146.39
23-Jan    1135.57
22-Feb     937.79
22-May     931.98
21-Dec     876.37
22-Apr     865.44
21-Nov     840.98
22-Mar     776.16
21-Sep     686.76
21-Aug     646.99
22-Jan     627.05
22-Jul     543.61
22-Aug     520.94
22-Jun     516.24
23-Feb     513.46
21-Oct     491.55
21-Jul     415.33
Name: Amount (CAD), dtype: float64
```

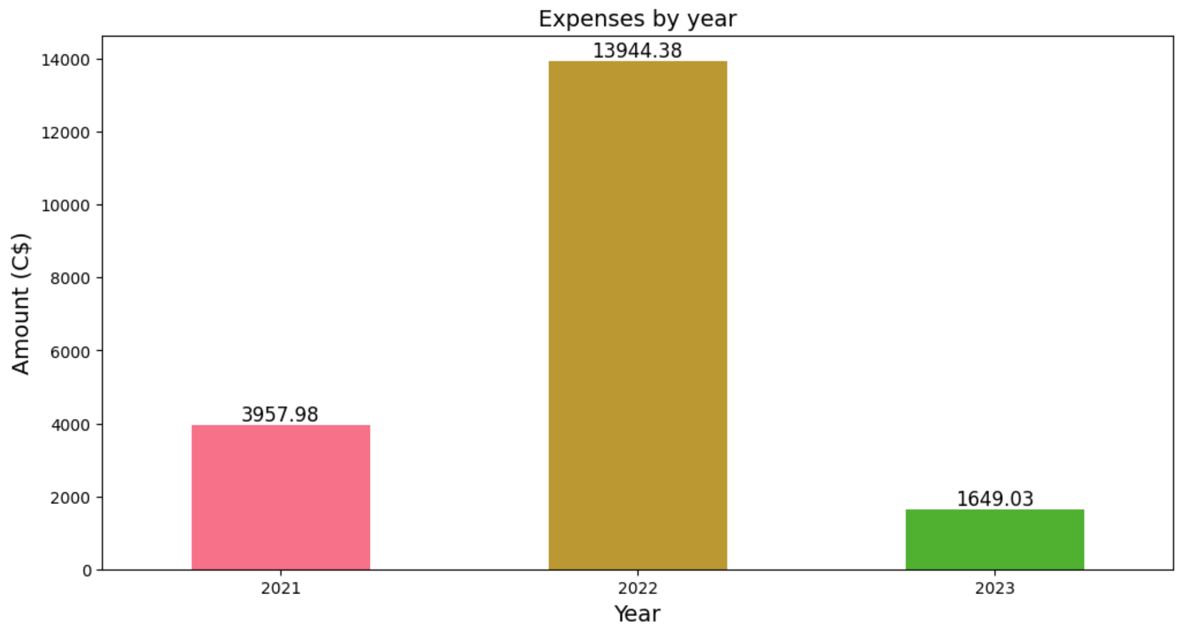
In [12]:
```
# expenses by month, descending, bar chart

monthly_exp = mo_exp.groupby('Month')['Amount (CAD)'].sum().sort_values(ascenc
plots = monthly_exp.head(10).plot(kind='bar',
                                  figsize=(12,6),
                                  color=sns.color_palette('husl'))
plt.xticks(rotation=0)
plt.title('Expenses by month', size=14)
plt.xlabel('Month', size=14)
plt.ylabel('Amount (C$)', size=14)
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.2f'),
                   (bar.get_x()+bar.get_width()/2,
                   bar.get_height()), ha='center', va='center',
                   size=10, xytext=(0,8),
                   textcoords='offset points')
```



In [13]:
```
# expense by year

year_exp = mo_exp.groupby('Year')['Amount (CAD)'].sum()
year_exp
```

Out[13]:
```
Year
2021     3957.98
2022    13944.38
2023     1649.03
Name: Amount (CAD), dtype: float64
```

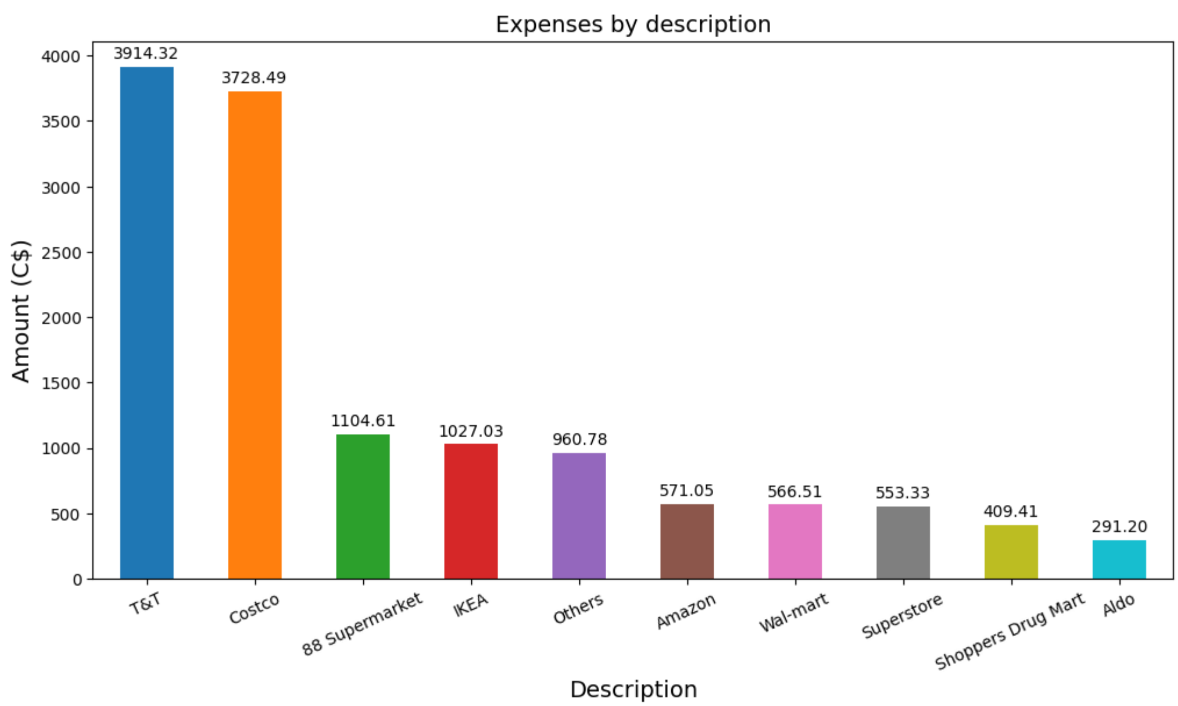
In [14]:
```
# expense by year, bar chart

plots = year_exp.plot(kind='bar', color=sns.color_palette('husl'),
              figsize=(12,6))
plt.xticks(rotation=0)
plt.title('Expenses by year', size=14)
plt.xlabel('Year', size=14)
plt.ylabel('Amount (C$)', size=14)
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.2f'),
                  (bar.get_x()+bar.get_width()/2,
                  bar.get_height()), ha='center', va='center',
                  size=12, xytext=(0,8),
                  textcoords='offset pixels')
```



In [15]:
```
# expenses by description, descending

description_exp = mo_exp.groupby('Description')['Amount (CAD)'].sum().sort_val
description_exp
```

Out[15]:
```
Description
T&T                  3914.32
Costco               3728.49
88 Supermarket       1104.61
IKEA                 1027.03
Others                960.78
                      ...
Muji                    6.75
Germain Bakery          6.38
Passion8                6.36
Real Canadian           4.27
Tim Horton              1.56
Name: Amount (CAD), Length: 162, dtype: float64
```

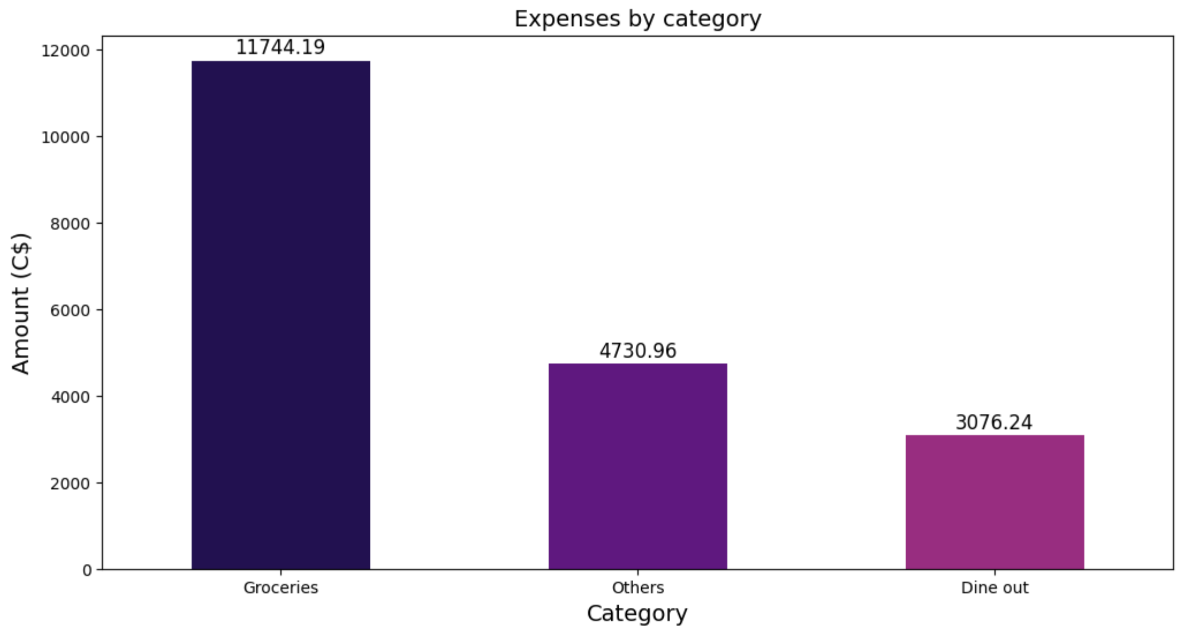
In [16]:
```
# expenses by description, descending, bar chart

plots = description_exp.head(10).plot(kind='bar',
                              color=sns.color_palette('tab10'),
                              figsize=(12,6))
plt.xticks(rotation=25)
plt.title('Expenses by description', size=14)
plt.xlabel('Description', size=14)
plt.ylabel('Amount (C$)', size=14)
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.2f'),
                (bar.get_x()+bar.get_width()/2,
                bar.get_height()), ha='center', va='center',
                size=10, xytext=(0,8),
                textcoords='offset points')
```



In [17]:
```
# expenses by category, descending

category_exp = mo_exp.groupby('Category')['Amount (CAD)'].sum().sort_values(as
category_exp
```

Out[17]:
```
Category
Groceries    11744.19
Others        4730.96
Dine out      3076.24
Name: Amount (CAD), dtype: float64
```

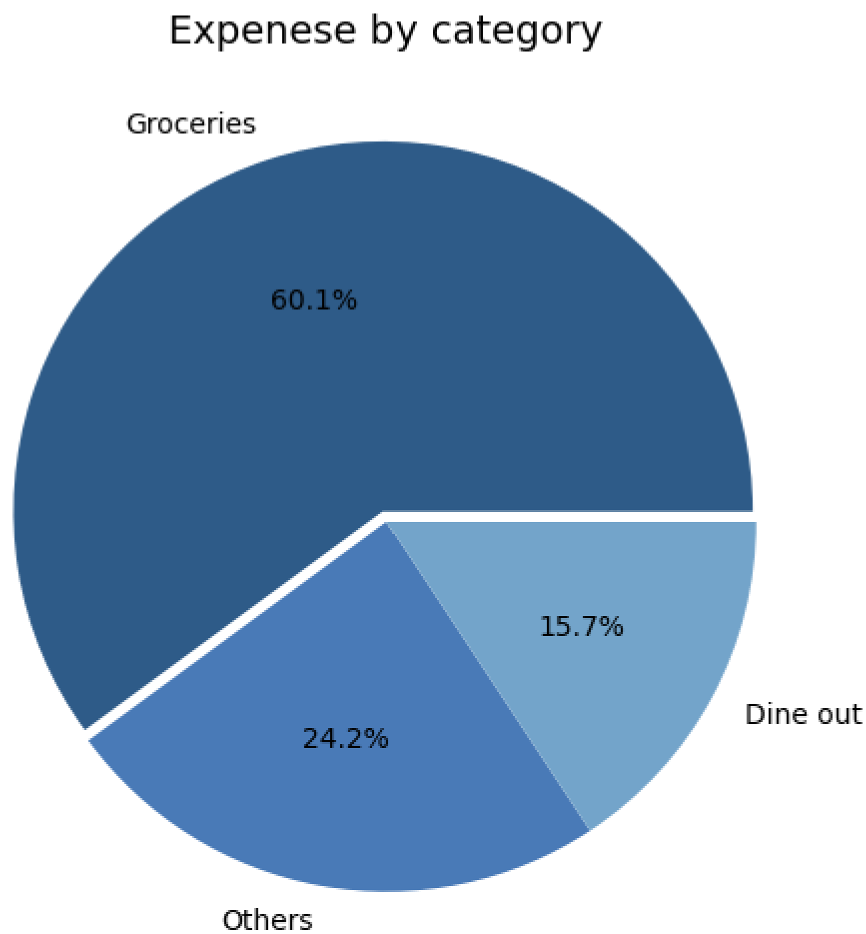
In [18]:
```
# expenses by category, descending, bar chart

plots = category_exp.plot(kind='bar', figsize=(12,6),
                  color=sns.color_palette('magma'))
plt.xticks(rotation=0)
plt.title('Expenses by category', size=14)
plt.xlabel('Category', size=14)
plt.ylabel('Amount (C$)', size=14)
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.2f'),
                  (bar.get_x()+bar.get_width()/2,
                  bar.get_height()), ha='center', va='center',
                  size=12, xytext=(0,8),
                  textcoords='offset points')
```



In [19]:
```
grocercies = mo_exp[mo_exp['Category'] == 'Groceries']
others = mo_exp[mo_exp['Category'] == 'Others']
dine_out = mo_exp[mo_exp['Category'] == 'Dine out']
```

In [20]:

```
# expenses by category, pie chart

data = np.array([grocercies['Amount (CAD)'].sum(), others['Amount (CAD)'].sum(
labels = ['Groceries', 'Others', 'Dine out']
plt.figure(figsize=(12,6))
plt.title('Expenese by category', fontsize=14)

colors = ["#2E5B88", "#497AB7", "#73A4CA"]
explode=[0.03, 0.00, 0.00]

plt.pie(data, labels=labels,
        explode=explode,
        colors=colors,
        autopct='%.1f%%')
plt.show()
```



In [21]:
```
# top 10 expense amount

top_10_exp = mo_exp.sort_values(by='Amount (CAD)',
                                ascending=False)
top_10_exp.head(10)
```

Out[21]:

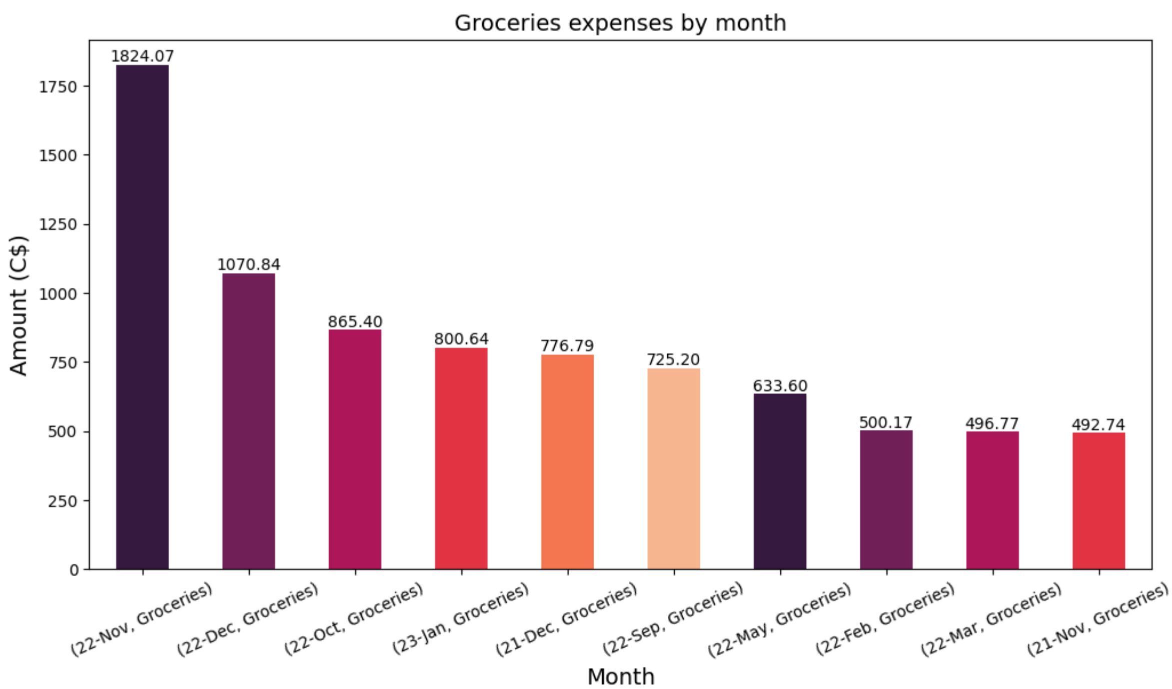
	ï»¿ID	Description	Category	Amount (CAD)	Month	Year
241	EXP254	IKEA	Others	721.93	22-Oct	2022
264	EXP282	Costco	Groceries	584.45	22-Nov	2022
278	EXP300	Costco	Groceries	584.45	22-Dec	2022
247	EXP260	T&T	Groceries	426.71	22-Nov	2022
231	EXP241	T&T	Groceries	364.48	22-Oct	2022
82	EXP083	Costco	Groceries	315.65	21-Dec	2021
290	EXP315	T&T	Groceries	310.66	23-Jan	2023
292	EXP318	Costco	Groceries	288.03	23-Jan	2023
250	EXP263	Costco	Groceries	267.66	22-Nov	2022
301	EXP330	T&T	Groceries	267.48	23-Feb	2023

In [22]:
```
# groceries expenses by month, descending

groc_by_month = mo_exp.groupby(['Month', 'Category'])['Amount (CAD)'].sum().sc
groc_by_month.filter(like='Groceries').head(10)
#groc_by_month.head(10)
```

Out[22]:
```
Month   Category
22-Nov  Groceries    1824.07
22-Dec  Groceries    1070.84
22-Oct  Groceries     865.40
23-Jan  Groceries     800.64
21-Dec  Groceries     776.79
22-Sep  Groceries     725.20
22-May  Groceries     633.60
22-Feb  Groceries     500.17
22-Mar  Groceries     496.77
21-Nov  Groceries     492.74
Name: Amount (CAD), dtype: float64
```

In [23]:

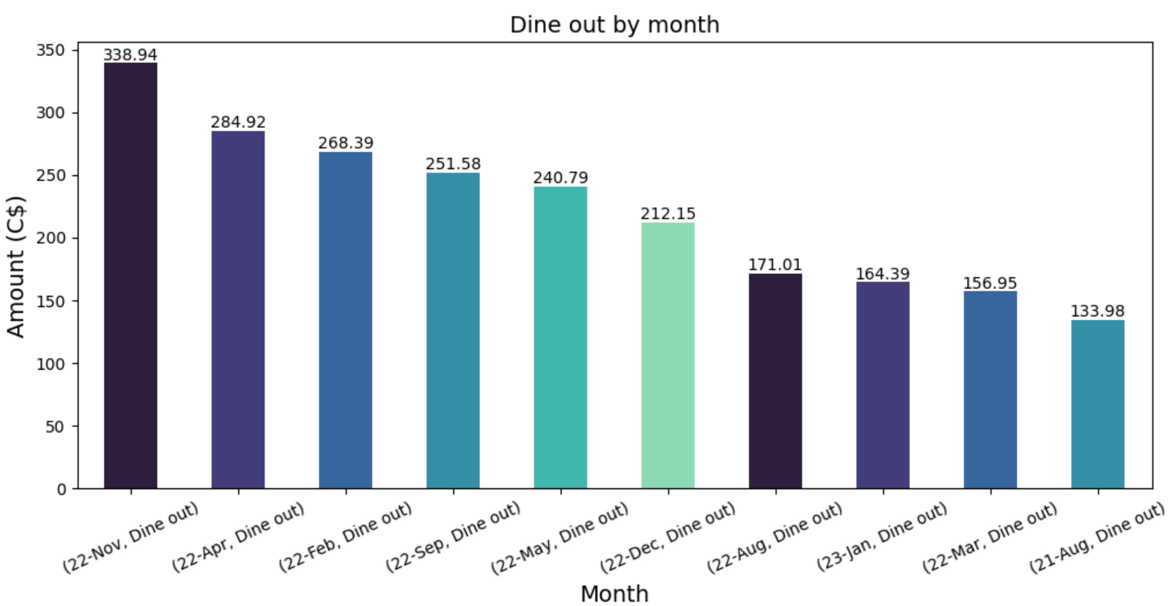
```python
# groceries expenses by month, descending, bar chart

plots = groc_by_month.filter(like='Groceries').head(10).plot(kind='bar',
                              xlabel='Month',
                              title='Groceries expenses by month',
                              figsize=(12,6),
                              color=sns.color_palette('rocket'))
plt.xticks(rotation=25)
plt.title('Groceries expenses by month', size=14)
plt.xlabel('Month', size=14)
plt.ylabel('Amount (C$)', size=14)
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.2f'),
                   (bar.get_x()+bar.get_width()/2,
                   bar.get_height()), ha='center', va='center',
                   size=10, xytext=(0,5),
                   textcoords='offset points')
```



In [24]:
```python
# dine out expenses by month

dineout_by_month = mo_exp.groupby(['Month', 'Category'])['Amount (CAD)'].sum()
dineout_by_month.filter(like='Dine out').head(10)
```

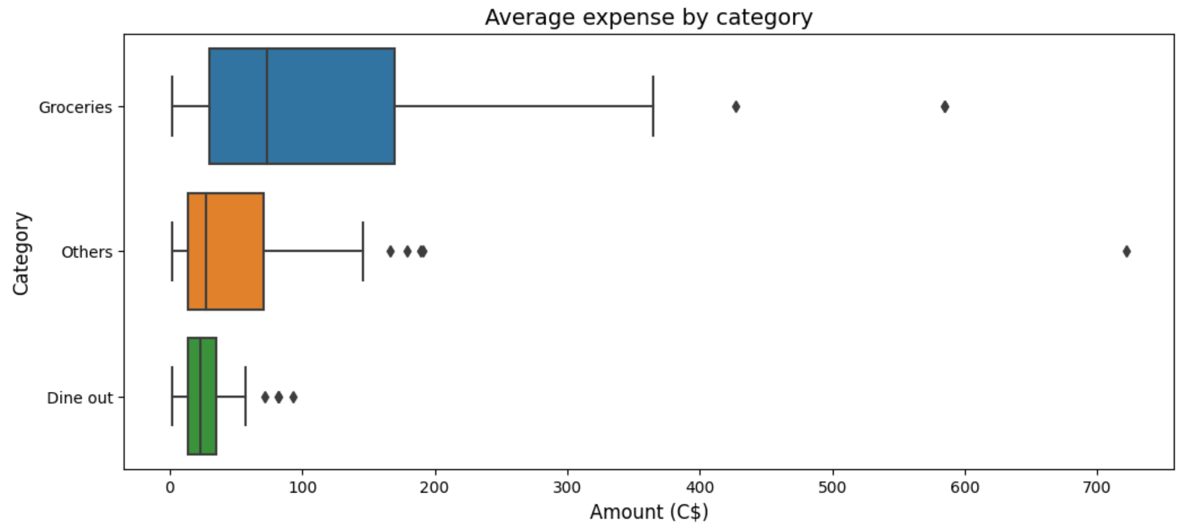
Out[24]:
```
Month   Category
22-Nov  Dine out    338.94
22-Apr  Dine out    284.92
22-Feb  Dine out    268.39
22-Sep  Dine out    251.58
22-May  Dine out    240.79
22-Dec  Dine out    212.15
22-Aug  Dine out    171.01
23-Jan  Dine out    164.39
22-Mar  Dine out    156.95
21-Aug  Dine out    133.98
Name: Amount (CAD), dtype: float64
```

In [29]:
```python
# dine out expenses by month, descending, bar chart

plots = dineout_by_month.filter(like='Dine out').head(10).plot(kind='bar',
                                              figsize=(12, 5),
                                              xlabel='Month',
                                              color=sns.color_palette('mako'))
plt.xticks(rotation=25)
plt.title('Dine out by month', size=14)
plt.xlabel('Month', size=14)
plt.ylabel('Amount (C$)', size=14)
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.2f'),
                  (bar.get_x()+bar.get_width()/2,
                  bar.get_height()), ha='center', va='center',
                  size=10, xytext=(0,5),
                  textcoords='offset points')
```



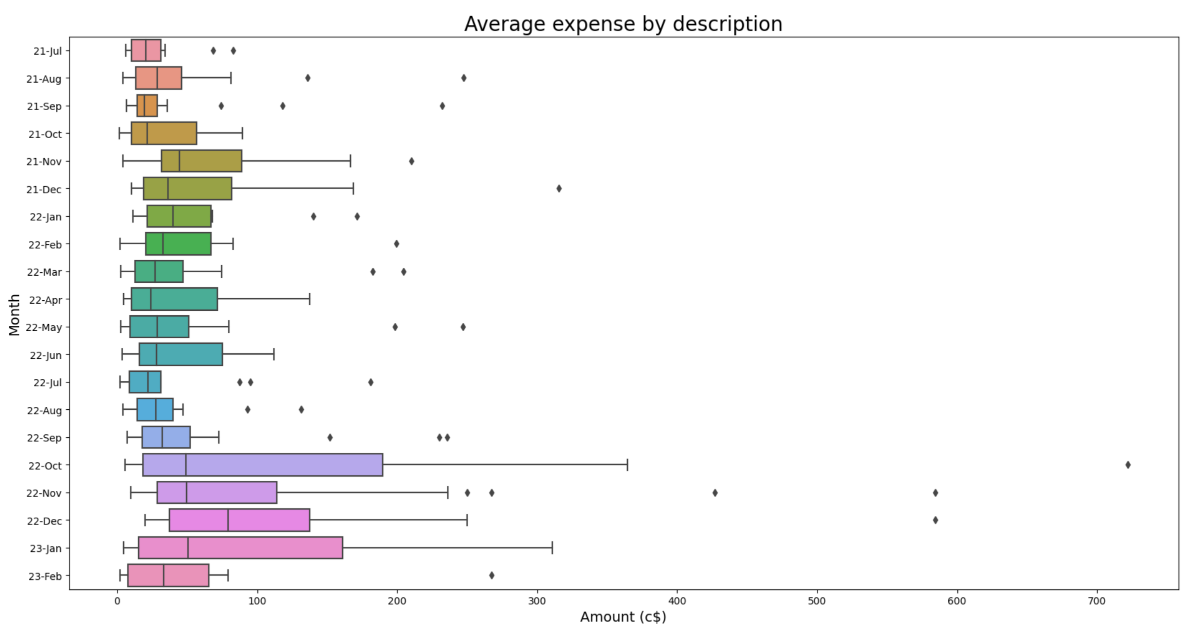
In [26]:
```
plt.figure(figsize=(12,5))
sns.boxplot(orient='h', x='Amount (CAD)', y='Category', data=mo_exp)
plt.xticks(rotation=0)
plt.title('Average expense by category', size=14)
plt.xlabel('Amount (C$)', size=12)
plt.ylabel('Category', size=12)
```

Out[26]:
```
Text(0, 0.5, 'Category')
```



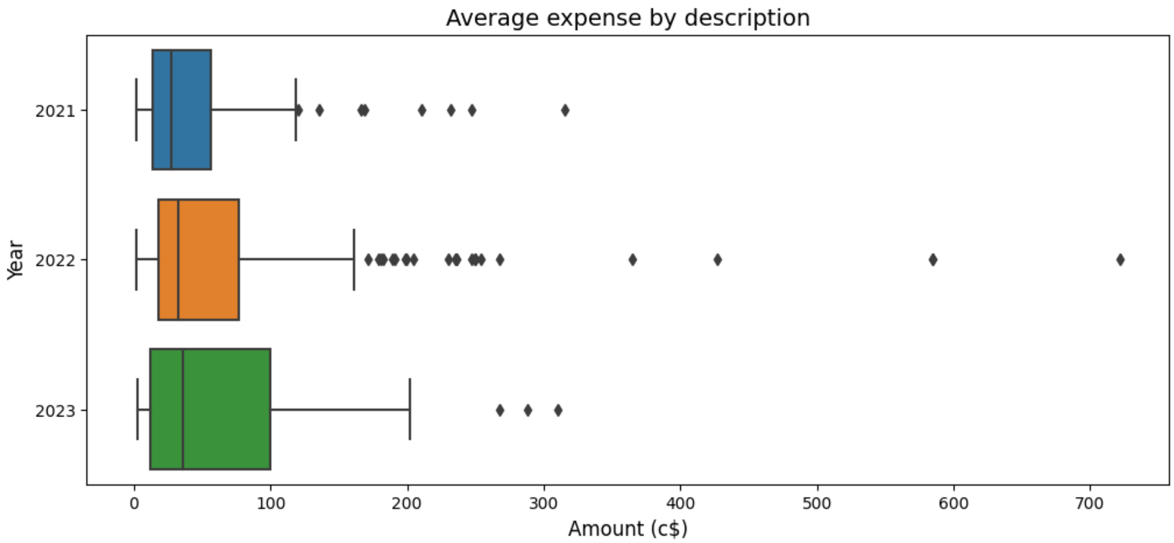
In [27]:
```
plt.figure(figsize=(20,10))
sns.boxplot(orient='h', x='Amount (CAD)', y='Month', data=mo_exp)
plt.xticks(rotation=0)
plt.title('Average expense by description', size=20)
plt.xlabel('Amount (c$)', size=14)
plt.ylabel('Month', size=14)
```

Out[27]:
```
Text(0, 0.5, 'Month')
```



In [28]:
```
plt.figure(figsize=(12,5))
sns.boxplot(orient='h', x='Amount (CAD)', y='Year', data=mo_exp)
plt.xticks(rotation=0)
plt.title('Average expense by description', size=14)
plt.xlabel('Amount (c$)', size=12)
plt.ylabel('Year', size=12)
```

Out[28]:
```
Text(0, 0.5, 'Year')
```



In [ ]:

In [ ]: