In [51]:
```
pip install squarify
```

```
Requirement already satisfied: squarify in c:\users\mat\anaconda3\lib\site-packages
(0.4.3)
Note: you may need to restart the kernel to use updated packages.
```

In [52]:
```
import pandas as pd
import numpy as np
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
import json

pd.options.display.float_format = '{:,.2f}'.format
```

In [53]:
```
pip install geopandas
```

```
Requirement already satisfied: geopandas in c:\users\mat\anaconda3\lib\site-packages
(0.13.2)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: shapely>=1.7.1 in c:\users\mat\anaconda3\lib\site-pack
ages (from geopandas) (2.0.1)
Requirement already satisfied: pandas>=1.1.0 in c:\users\mat\anaconda3\lib\site-packa
ges (from geopandas) (1.5.3)
Requirement already satisfied: fiona>=1.8.19 in c:\users\mat\anaconda3\lib\site-packa
ges (from geopandas) (1.9.4.post1)
Requirement already satisfied: packaging in c:\users\mat\anaconda3\lib\site-packages
(from geopandas) (23.0)
Requirement already satisfied: pyproj>=3.0.1 in c:\users\mat\anaconda3\lib\site-packa
ges (from geopandas) (3.6.0)
Requirement already satisfied: cligj>=0.5 in c:\users\mat\anaconda3\lib\site-packages
(from fiona>=1.8.19->geopandas) (0.7.2)
Requirement already satisfied: click-plugins>=1.0 in c:\users\mat\anaconda3\lib\site-
packages (from fiona>=1.8.19->geopandas) (1.1.1)
Requirement already satisfied: certifi in c:\users\mat\anaconda3\lib\site-packages (f
rom fiona>=1.8.19->geopandas) (2023.7.22)
Requirement already satisfied: attrs>=19.2.0 in c:\users\mat\anaconda3\lib\site-packa
ges (from fiona>=1.8.19->geopandas) (22.1.0)
Requirement already satisfied: click~=8.0 in c:\users\mat\anaconda3\lib\site-packages
(from fiona>=1.8.19->geopandas) (8.0.4)
Requirement already satisfied: six in c:\users\mat\anaconda3\lib\site-packages (from
fiona>=1.8.19->geopandas) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\mat\anaconda3\lib\s
ite-packages (from pandas>=1.1.0->geopandas) (2.8.2)
Requirement already satisfied: numpy>=1.21.0 in c:\users\mat\anaconda3\lib\site-packa
ges (from pandas>=1.1.0->geopandas) (1.23.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\mat\anaconda3\lib\site-packag
es (from pandas>=1.1.0->geopandas) (2022.7)
Requirement already satisfied: colorama in c:\users\mat\anaconda3\lib\site-packages
(from click~=8.0->fiona>=1.8.19->geopandas) (0.4.6)
```

In [54]:
```
import geopandas as gpd
from bokeh.io import output_notebook, show, output_file
from bokeh.plotting import figure
from bokeh.models import GeoJSONDataSource, LinearColorMapper, ColorBar
from bokeh.palettes import brewer
from bokeh.models import HoverTool
```

In [55]:
```
df = pd.read_csv(r'D:\bp_sales_dataset.csv')
```

In [56]:
```
df.columns = [col.lower().replace(' ', '_') for col in df.columns]
```

In [57]:
```
df.head()
```

Out[57]:

	id	date	description	qty	retail	subtotal	discount	tax	total	customer	sou
0	500945	3/14/2022	AIR TOOL SWITCH SPORT BLK	1	20.00	20.00	0.00	0%	20.00	CHRISTOPHER VASQUEZ	N
1	500992	3/14/2022	AETHON, CRYSTAL SMOKE/WHITE FOTOTEC SUNGLASSE...	-1	79.99	-79.99	0.00	9.25%	-87.39	SUSAN JABLONSKI	N
2	501065	3/14/2022	EVO, HIGHTAIL, PLATFORM PEDALS, BODY: ALUMINUM...	1	34.99	34.99	0.00	9.25%	38.23	MARCIA MCDONALD	N
3	501065	3/14/2022	BH - PEDAL - INSTALL - PEDALS - PAIR	1	5.00	5.00	0.00	0%	5.00	MARCIA MCDONALD	N
4	501065	3/14/2022	ALIGN II HLMT MIPS CPSC BLK/BLKREFL S/M	1	55.00	55.00	0.00	9.25%	60.09	MARCIA MCDONALD	N

In [58]:
```
df = df.drop(['subtotal','tax', 'source', 'work_order_internal_note'], axis=1)
```

In [59]:
```
df['date'] = pd.to_datetime(df['date'])
```

In [60]:
```
df.head()
```

Out[60]:

	id	date	description	qty	retail	discount	total	customer
0	500945	2022-03-14	AIR TOOL SWITCH SPORT BLK	1	20.00	0.00	20.00	CHRISTOPHER VASQUEZ
1	500992	2022-03-14	AETHON, CRYSTAL SMOKE/WHITE FOTOTEC SUNGLASSE...	-1	79.99	0.00	-87.39	SUSAN JABLONSKI
2	501065	2022-03-14	EVO, HIGHTAIL, PLATFORM PEDALS, BODY: ALUMINUM...	1	34.99	0.00	38.23	MARCIA MCDONALD
3	501065	2022-03-14	BH - PEDAL - INSTALL - PEDALS - PAIR	1	5.00	0.00	5.00	MARCIA MCDONALD
4	501065	2022-03-14	ALIGN II HLMT MIPS CPSC BLK/BLKREFL S/M	1	55.00	0.00	60.09	MARCIA MCDONALD

In [61]:
```
from datetime import date
```

In [62]:
```
today = date. today()
```

In [63]:
```
max_date = df['date'].min()
print(max_date)
```

```
2022-03-14 00:00:00
```

In [64]:
```
#change date data to datetime
today=pd.to_datetime(today)
max_date=pd.to_datetime(max_date)
```

In [65]:
```
agg_dict1 = {
    'id': 'count',
    'date': 'max',
    'retail': 'sum'
}

df_rfm = df.groupby('customer').agg(agg_dict1).reset_index()
df_rfm.columns = ['customer', 'frequency', 'max_date', 'monetary']
df_rfm['recency'] = (today - df_rfm['max_date']).dt.days
df_rfm.drop(['max_date'], axis=1, inplace=True)
```

In [ ]:
```

```

In [110…
```

```

In [66]:
```
r_labels, f_labels, m_labels = range(5, 0, -1), range(1,6), range(1,6)
```

In [67]:
```
df_rfm['r_score'] = pd.qcut(df_rfm['recency'], q=5, labels=r_labels).astype(int)
```

In [68]:
```
df_rfm['f_score'] = pd.qcut((df_rfm.rank(method='first'))['frequency'], q=5, labels=f_
```

In [69]:
```
df_rfm['m_score'] = pd.qcut(df_rfm['monetary'], q=5, labels=m_labels).astype(int)
```

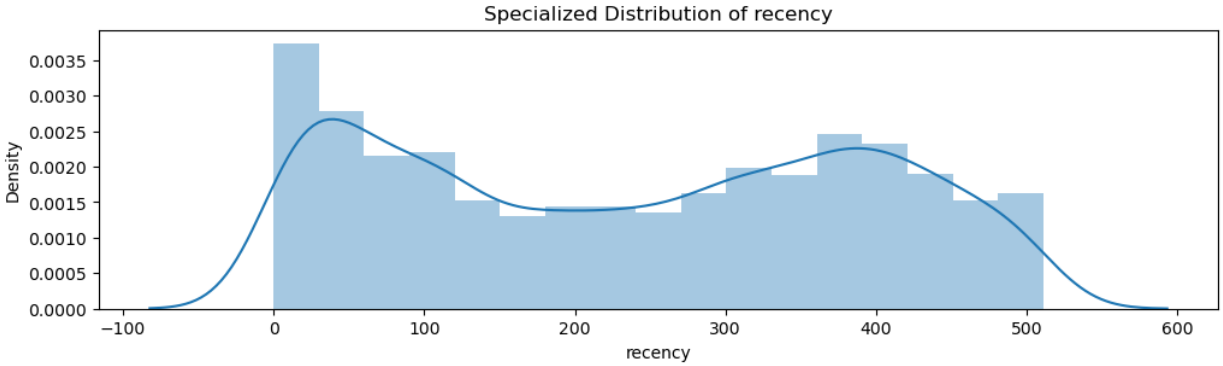
In [70]:
```
df_rfm['rfm_sum'] = df_rfm['r_score'] + df_rfm['f_score'] + df_rfm['m_score']
```

In [71]:
```
def assign_label(df, r_rule, fm_rule, label, colname='rfm_label'):
    df.loc[(df['r_score'].between(r_rule[0], r_rule[1]))
            & (df['f_score'].between(fm_rule[0], fm_rule[1])), colname] = label
    return df
```

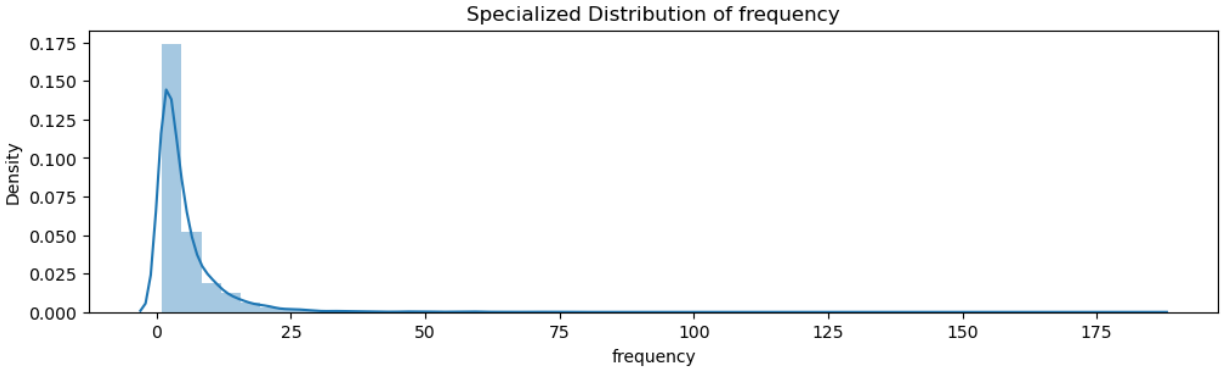
In [72]:
```
df_rfm['rfm_label'] = ''

df_rfm = assign_label(df_rfm, (5,5), (4,5), 'champions')
df_rfm = assign_label(df_rfm, (3,4), (4,5), 'loyal customers')
df_rfm = assign_label(df_rfm, (4,5), (2,3), 'potential loyalist')
df_rfm = assign_label(df_rfm, (5,5), (1,1), 'new customers')
df_rfm = assign_label(df_rfm, (4,4), (1,1), 'promising')
df_rfm = assign_label(df_rfm, (3,3), (3,3), 'needing attention')
df_rfm = assign_label(df_rfm, (3,3), (1,2), 'about to sleep')
df_rfm = assign_label(df_rfm, (1,2), (3,4), 'at risk')
df_rfm = assign_label(df_rfm, (1,2), (5,5), 'cant loose them')
df_rfm = assign_label(df_rfm, (1,2), (1,2), 'hibernating')
```

In [74]:
```
colnames = ['recency', 'frequency', 'monetary']

for col in colnames:
    fig, ax = plt.subplots(figsize=(12,3))
    sns.distplot(df_rfm[col])
    ax.set_title('Specialized Distribution of %s' % col)
    plt.show()
```

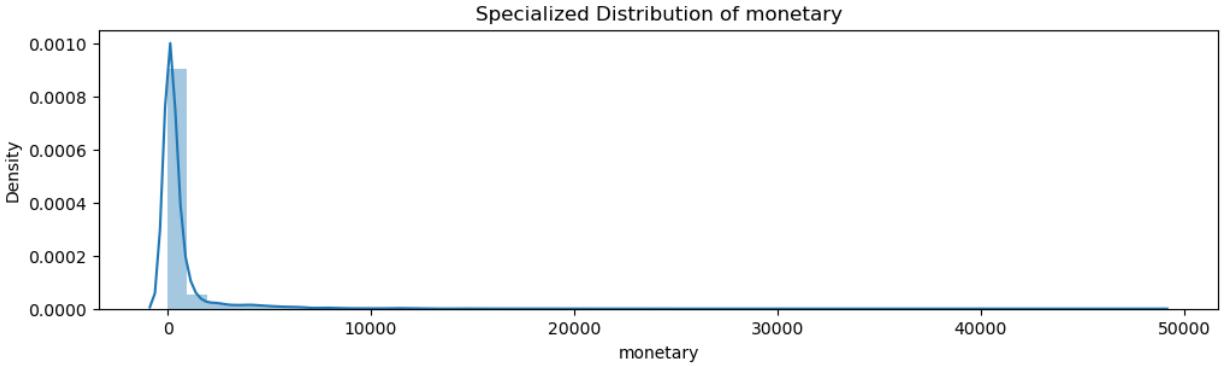
```
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2876042570.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[col])
```




```
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2876042570.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[col])
```



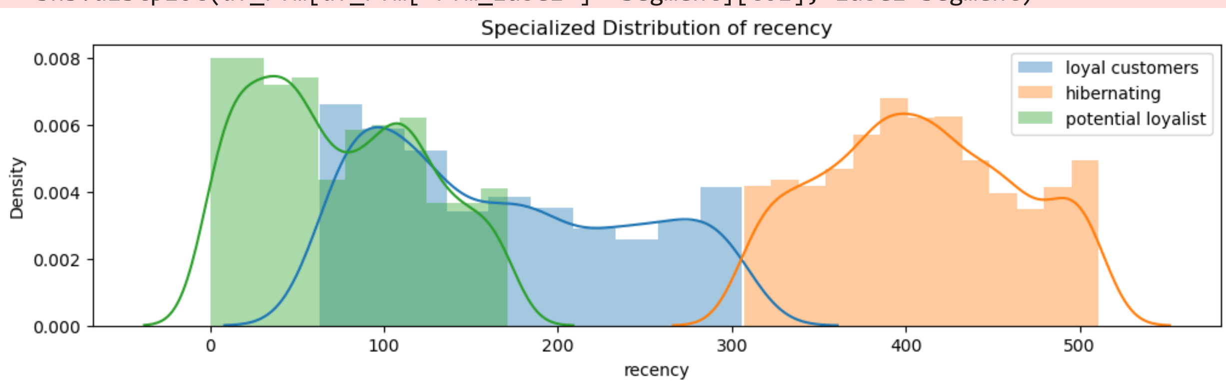
```
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2876042570.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[col])
```



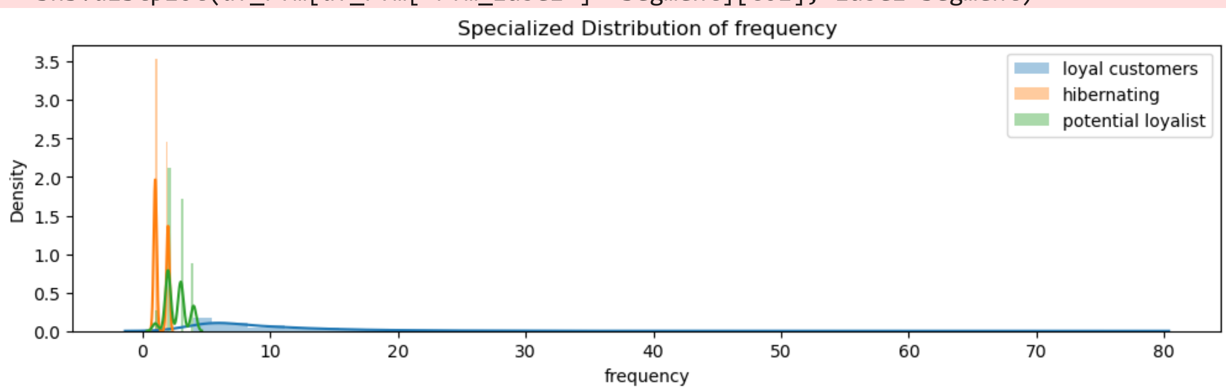
In [ ]:
```
,'loyal customers' 'hibernating', 'potential loyalist'
```

In [76]:
```
segments = ['loyal customers', 'hibernating', 'potential loyalist']

for col in colnames:
    fig, ax = plt.subplots(figsize=(12,3))
    for segment in segments:
        sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
    ax.set_title('Specialized Distribution of %s' % col)
    plt.legend()
    plt.show()
```

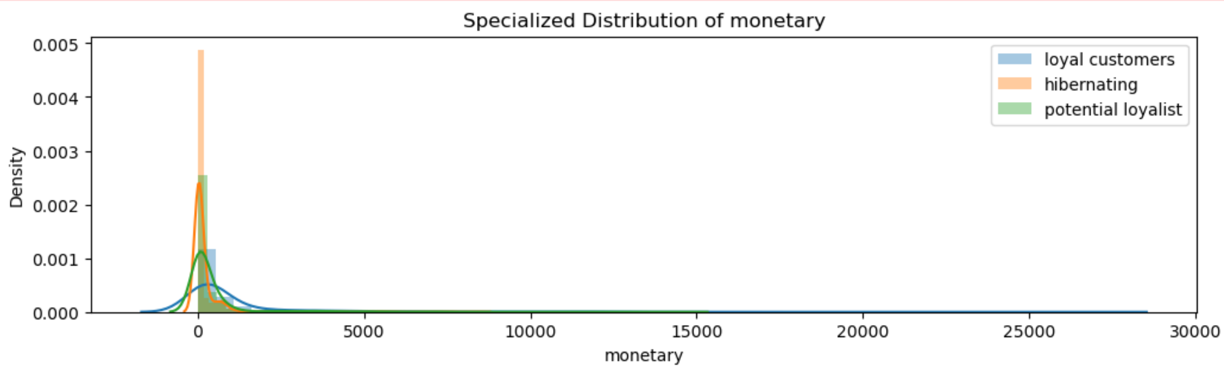
```
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
```



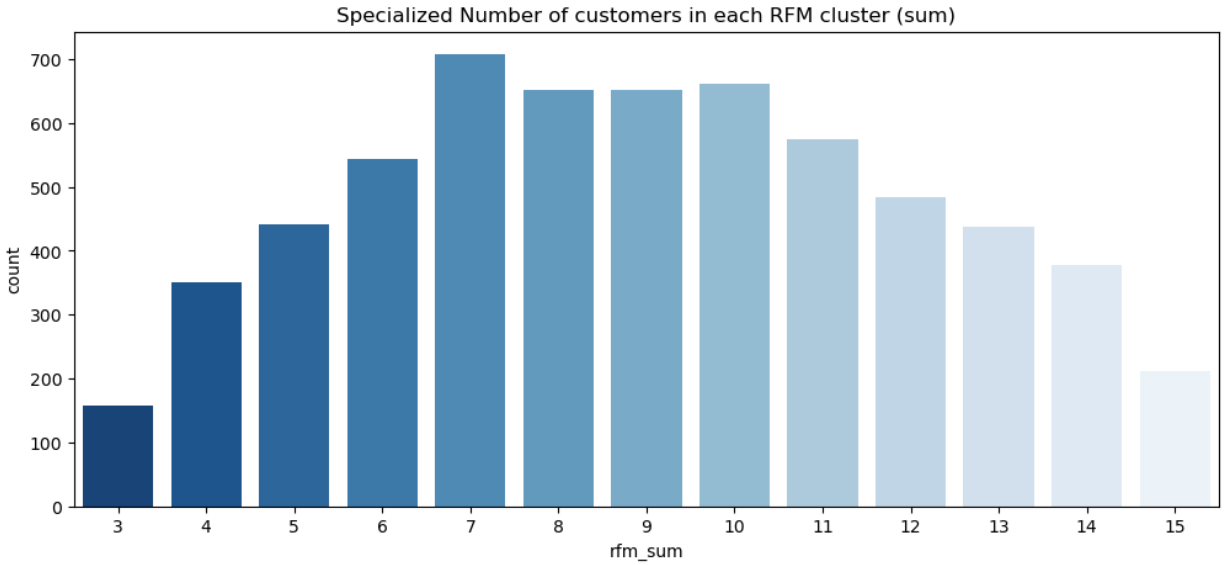
```
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
```

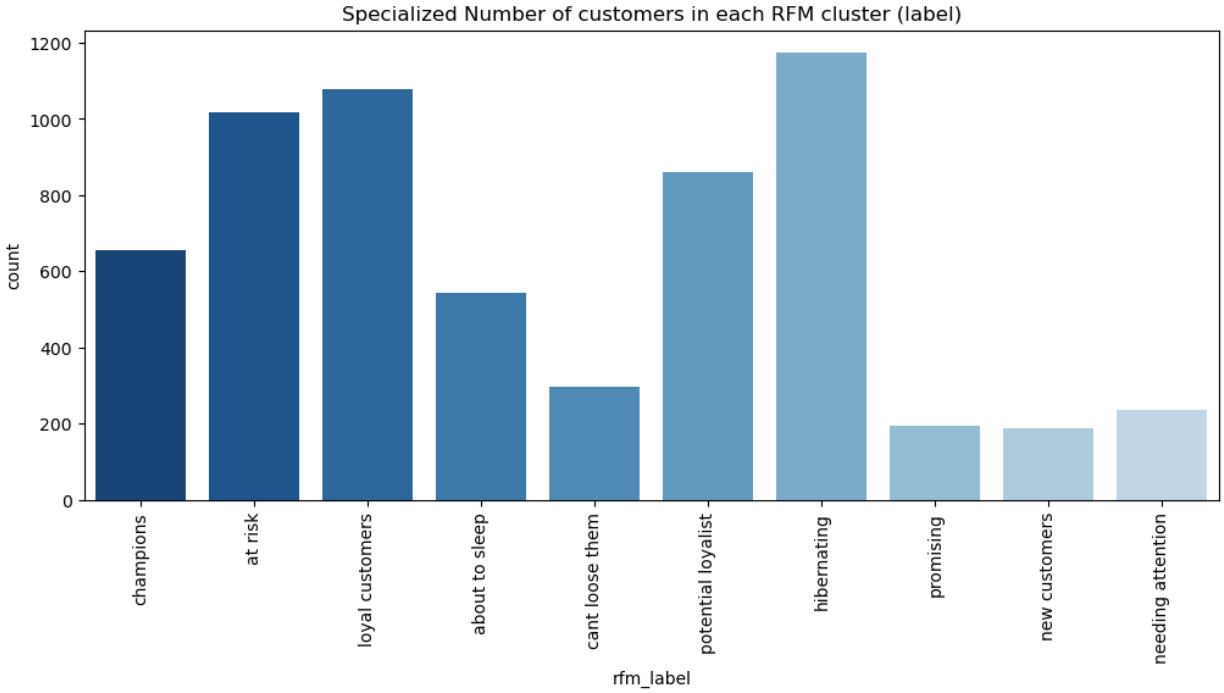


```
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
C:\Users\Mat\AppData\Local\Temp\ipykernel_27712\2290199572.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rfm[df_rfm['rfm_label']==segment][col], label=segment)
```



In [77]:
```
palette = sns.color_palette("Blues_r", n_colors=13)

for rfm_type in ['sum', 'label']:
    fig, ax = plt.subplots(figsize=(12,5))
    sns.countplot(x='rfm_'+rfm_type, data=df_rfm, palette=palette)
    ax.set_title('Specialized Number of customers in each RFM cluster (%s)' % rfm_type
    if rfm_type == 'label':
        plt.xticks(rotation=90)
    plt.show()
```

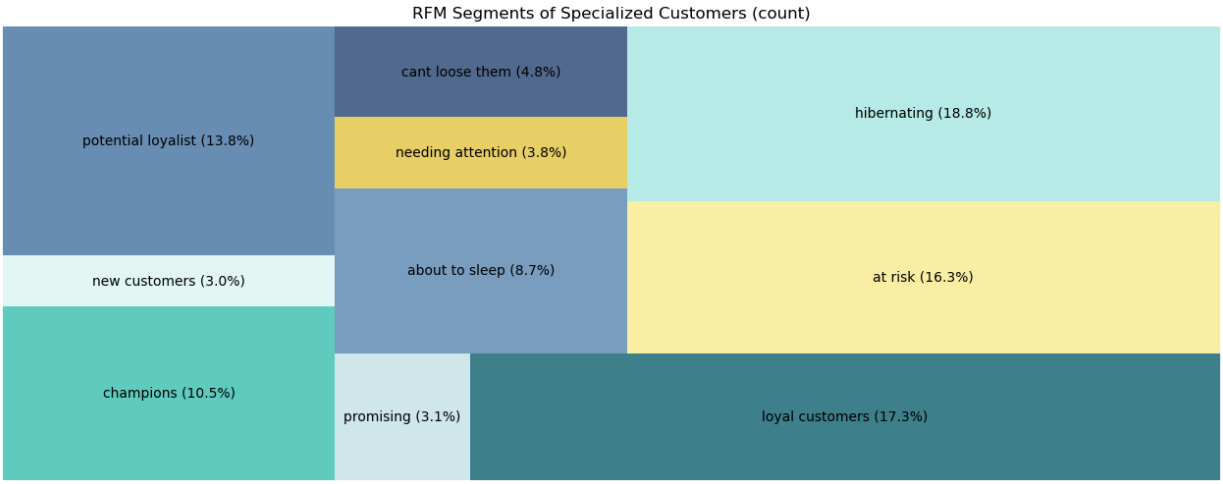


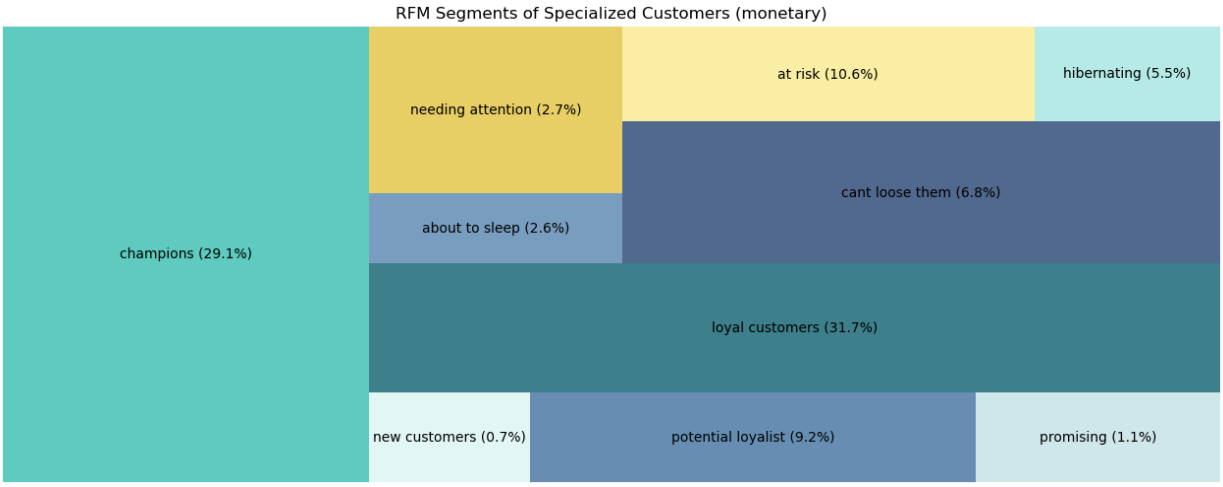


In [79]:
```
agg_dict2 = {
    'customer': 'count',
    'recency': 'mean',
    'frequency': 'mean',
    'monetary': 'sum'
}

df_analysis = df_rfm.groupby('rfm_label').agg(agg_dict2).sort_values(by='recency').res
df_analysis.rename({'rfm_label': 'label', 'customer': 'count'}, axis=1, inplace=True)
df_analysis['count_share'] = df_analysis['count'] / df_analysis['count'].sum()
df_analysis['monetary_share'] = df_analysis['monetary'] / df_analysis['monetary'].sum(
df_analysis['monetary'] = df_analysis['monetary'] / df_analysis['count']
```

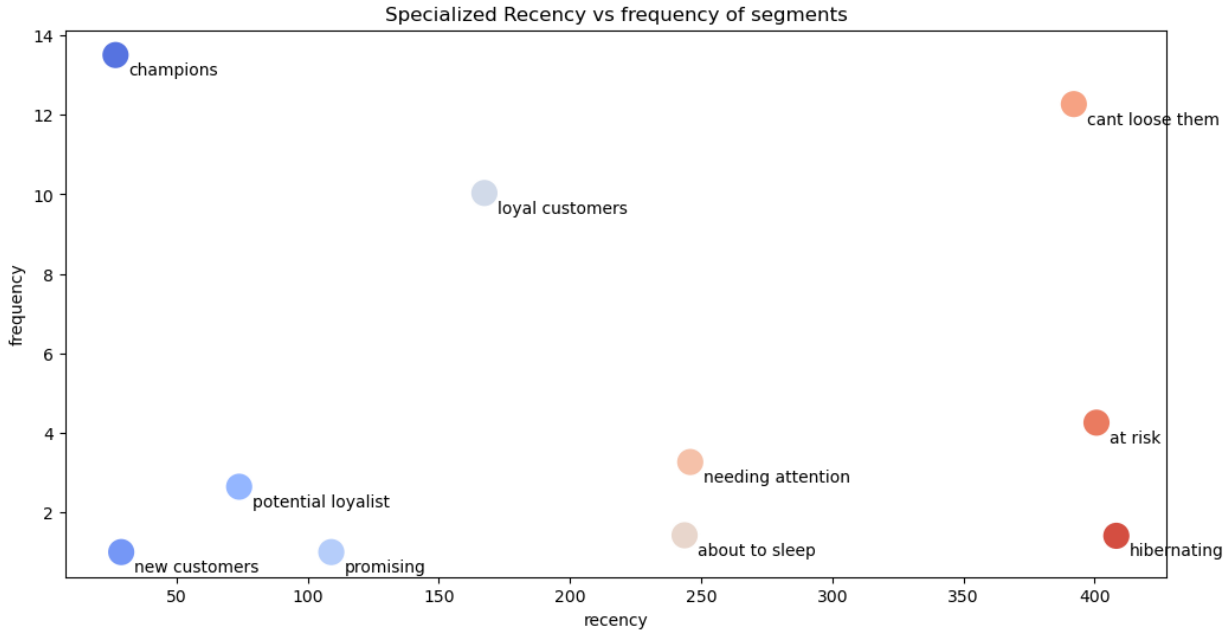
In [80]:
```
colors = ['#37BEB0', '#DBF5F0', '#41729F', '#C3E0E5', '#0C6170', '#5885AF', '#E1C340',

for col in ['count', 'monetary']:
    labels = df_analysis['label'] + df_analysis[col + '_share'].apply(lambda x: ' ({0:

    fig, ax = plt.subplots(figsize=(16,6))
```

```
    squarify.plot(sizes=df_analysis[col], label=labels, alpha=.8, color=colors)
    ax.set_title('RFM Segments of Specialized Customers (%s)' % col)
    plt.axis('off')
    plt.show()
```





In [81]:
```
palette = sns.color_palette("coolwarm", 10)

fig, ax = plt.subplots(figsize=(12,6))
plot = sns.scatterplot(x='recency', y='frequency', data=df_analysis, hue='label', s=30

for i in range(len(df_analysis)):
     plot.text(df_analysis['recency'][i]+5,
               df_analysis['frequency'][i]-0.5,
               df_analysis['label'][i],
               horizontalalignment='left',
               size='medium', color='black')

ax.set_title('Specialized Recency vs frequency of segments')
ax.get_legend().remove()
plt.show()
```



In [44]:
```
display(df_rfm)
```

	customer	frequency	monetary	recency	r_score	f_score	m_score	rfm_sum	rfm_label
0		18	158.94	10	5	5	3	13	champions
1	andrew	3	25.50	411	1	3	1	5	at risk
2	AARON BEHRENS	17	273.91	5	5	5	4	14	champions
3	AARON BOICE	28	498.49	65	4	5	4	13	loyal customers
4	AARON BOWMAN	1	50.00	227	3	1	2	6	about to sleep
...	...	...	...	...	...	...	...	...	...
6243	william parry	5	56.99	31	5	4	2	11	champions
6244	yesenia garcia	9	175.70	502	1	5	3	9	cant loose them
6245	yongbai gong	10	580.22	409	1	5	4	10	cant loose them
6246	zack allen	1	599.99	453	1	2	5	8	hibernating
6247	zack mikalonis	8	118.97	248	3	5	3	11	loyal customers

6248 rows × 9 columns

In [82]:
```
df.head()
```

Out[82]:

	id	date	description	qty	retail	discount	total	customer
0	500945	2022-03-14	AIR TOOL SWITCH SPORT BLK	1	20.00	0.00	20.00	CHRISTOPHER VASQUEZ
1	500992	2022-03-14	AETHON, CRYSTAL SMOKE/WHITE FOTOTEC SUNGLASSE...	-1	79.99	0.00	-87.39	SUSAN JABLONSKI
2	501065	2022-03-14	EVO, HIGHTAIL, PLATFORM PEDALS, BODY: ALUMINUM...	1	34.99	0.00	38.23	MARCIA MCDONALD
3	501065	2022-03-14	BH - PEDAL - INSTALL - PEDALS - PAIR	1	5.00	0.00	5.00	MARCIA MCDONALD
4	501065	2022-03-14	ALIGN II HLMT MIPS CPSC BLK/BLKREFL S/M	1	55.00	0.00	60.09	MARCIA MCDONALD

In [ ]:

In [ ]:

In [46]:

In [ ]:

In [47]:

In [48]:
```
print(today)
```

```
2023-08-07
```

In [61]:

```
2022-03-14 00:00:00
```

In [68]:
```
pd.to_datetime
pd.to_timedelta
```

Out[68]:
```
<function pandas.core.tools.timedeltas.to_timedelta(arg: 'str | int | float | timedel
ta | list | tuple | range | ArrayLike | Index | Series', unit: 'UnitChoices | None' =
None, errors: 'DateTimeErrorChoices' = 'raise') -> 'Timedelta | TimedeltaIndex | Seri
es'>
```

In [85]:
```
agg_dict4 = {
    'product_id': 'count',
    'quantity': 'sum',
    'sales': 'sum',
    'discount': 'sum',
    'profit': 'sum',
    'rfm_sum': 'first',
    'rfm_label': 'first'
}

df_order = df.groupby('order_id').agg(agg_dict4).reset_index()
df_order_segment = df_order.groupby('rfm_label')[['quantity', 'sales', 'discount', 'pr
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[85], line 11
      1 agg_dict4 = {
      2     'product_id': 'count',
      3     'quantity': 'sum',
   (...)
      8     'rfm_label': 'first'
      9 }
---> 11 df_order = df.groupby('order_id').agg(agg_dict4).reset_index()
     12 df_order_segment = df_order.groupby('rfm_label')[['quantity', 'sales', 'disco
unt', 'profit', 'rfm_sum']].mean().reset_index()

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:8402, in DataFrame.groupby(se
lf, by, axis, level, as_index, sort, group_keys, squeeze, observed, dropna)
   8399     raise TypeError("You have to supply one of 'by' and 'level'")
   8400 axis = self._get_axis_number(axis)
-> 8402 return DataFrameGroupBy(
   8403     obj=self,
   8404     keys=by,
   8405     axis=axis,
   8406     level=level,
   8407     as_index=as_index,
   8408     sort=sort,
   8409     group_keys=group_keys,
   8410     squeeze=squeeze,
   8411     observed=observed,
   8412     dropna=dropna,
   8413 )

File ~\anaconda3\lib\site-packages\pandas\core\groupby\groupby.py:965, in GroupBy.__i
nit__(self, obj, keys, axis, level, grouper, exclusions, selection, as_index, sort, g
roup_keys, squeeze, observed, mutated, dropna)
    962 if grouper is None:
    963     from pandas.core.groupby.grouper import get_grouper
--> 965     grouper, exclusions, obj = get_grouper(
    966         obj,
    967         keys,
    968         axis=axis,
    969         level=level,
    970         sort=sort,
    971         observed=observed,
    972         mutated=self.mutated,
    973         dropna=self.dropna,
    974     )
    976 self.obj = obj
    977 self.axis = obj._get_axis_number(axis)

File ~\anaconda3\lib\site-packages\pandas\core\groupby\grouper.py:888, in get_grouper
(obj, key, axis, level, sort, observed, mutated, validate, dropna)
    886         in_axis, level, gpr = False, gpr, None
    887     else:
--> 888         raise KeyError(gpr)
    889 elif isinstance(gpr, Grouper) and gpr.key is not None:
    890     # Add key to exclusions
    891     exclusions.add(gpr.key)

KeyError: 'order_id'
```

In [48]:
```
df_rfm.to_excel(r'D:\bp_sales_rfmdataset.xlsx', index=False)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: