Data_Insights

March 11, 2021

[571]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
%matplotlib inline
from datetime import date, timedelta
import statsmodels.api as sm
sns.set_theme(style="whitegrid")
```

[572]:

```
c = sns.color_palette("RdBu", 10)[-3]
```

[573]:

```
def calculate_age(born):
    today = date.today()
    try:
        birthday = born.replace(year=today.year)
    except ValueError: # raised when birth date is February 29 and the current year is not a leap year
        birthday = born.replace(year=today.year, month=born.month+1, day=1)
    if birthday > today:
        return today.year - born.year - 1
    else:
        return today.year - born.year
```

0.1 Data Insight

by Murong (Sophie) Cui

Srocket Central Pty Ltd has given us a new list of 1000 potential custoomers with their demographics and attributes. However, these customers do not have prior transaction history with the organisation.

The marketing team at Sprocket Central Pty Ltd is sure that, if correctly analysed, the data would reveal useful customer insights which could help optimise resource allocation for targeted marketing. Hence, improve performance by focusing on high value customers.

For context, Sprocket Central Pty Ltd is a long-standing KPMG client whom specializes in high-quality bikes and accessible cycling accessories to riders. Their marketing team is looking to boost business by analysing their existing customer dataset to determine customer trends and behaviour.

In building this recommendation, we need to start with a PPT which outlines the approach which we will be taking. 3 phases as follows - Data Exploration, Model Development and Interpretation.

Prepare a detailed approach for completing the analysis including activities - i.e. understanding the data distributions, feature engineering, data transformations, modeling, results interpretation and reporting. Think detailed plan needs to be presented to the client to get a sign-off. Please advise what steps would take.

The analysis includes a detailed approach for ouor strategy behind each of the 3 phases including activities involved in each - i.e. understanding the data distributions, feature engineering, data transformations, modelling, results interpretation and reporting. This detailed plan needs to be presented to the client to get a sign-off.

- Data Exploration
- Model Development
- Interpretation

0.1.1 Data Exploration

`newCustoemr` 1000 new customers without prior purchases history

0.1.2 Data Wrangling

1. Uniqueness: The dataset have 1000 rows and 23 columns. Each row contains one potential customer's demographic and address infomation.
2. Missing value: 29 on `last_name`, 17 on `DOB`, 106 on `job_title`, 165 on `job_industry_category`.
3. Validity: convert 'U' to missing value
4. Consistency

[574]:
```
newCustomer_raw = pd.read_excel('data/KPMG_VI_New_raw_data_update_final.xlsx',␣
 ↪sheet_name = 'NewCustomerList', header = 1)
newCustomer = newCustomer_raw.copy()
```

[575]:
```
print('missing value \n', newCustomer.isnull().sum())
print('----------------')
print('\n shape', newCustomer.shape)
print('----------------')
print('Data Type', newCustomer.info())
print('----------------')
newCustomer.head(2)
```

```
missing value
 first_name                             0
last_name                             29
gender                                 0
past_3_years_bike_related_purchases    0
DOB                                   17
job_title                            106
job_industry_category                165
```

```
wealth_segment                           0
deceased_indicator                       0
owns_car                                 0
tenure                                   0
address                                  0
postcode                                 0
state                                    0
country                                  0
property_valuation                       0
Unnamed: 16                              0
Unnamed: 17                              0
Unnamed: 18                              0
Unnamed: 19                              0
Unnamed: 20                              0
Rank                                     0
Value                                    0
dtype: int64
---------------

 shape (1000, 23)
---------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 23 columns):
first_name                             1000 non-null object
last_name                              971 non-null object
gender                                 1000 non-null object
past_3_years_bike_related_purchases    1000 non-null int64
DOB                                    983 non-null datetime64[ns]
job_title                              894 non-null object
job_industry_category                  835 non-null object
wealth_segment                         1000 non-null object
deceased_indicator                     1000 non-null object
owns_car                               1000 non-null object
tenure                                 1000 non-null int64
address                                1000 non-null object
postcode                               1000 non-null int64
state                                  1000 non-null object
country                                1000 non-null object
property_valuation                     1000 non-null int64
Unnamed: 16                            1000 non-null float64
Unnamed: 17                            1000 non-null float64
Unnamed: 18                            1000 non-null float64
Unnamed: 19                            1000 non-null float64
Unnamed: 20                            1000 non-null int64
Rank                                   1000 non-null int64
Value                                  1000 non-null float64
dtypes: datetime64[ns](1), float64(5), int64(6), object(11)
```

```
memory usage: 179.8+ KB
Data Type None
---------------
```

[575]:

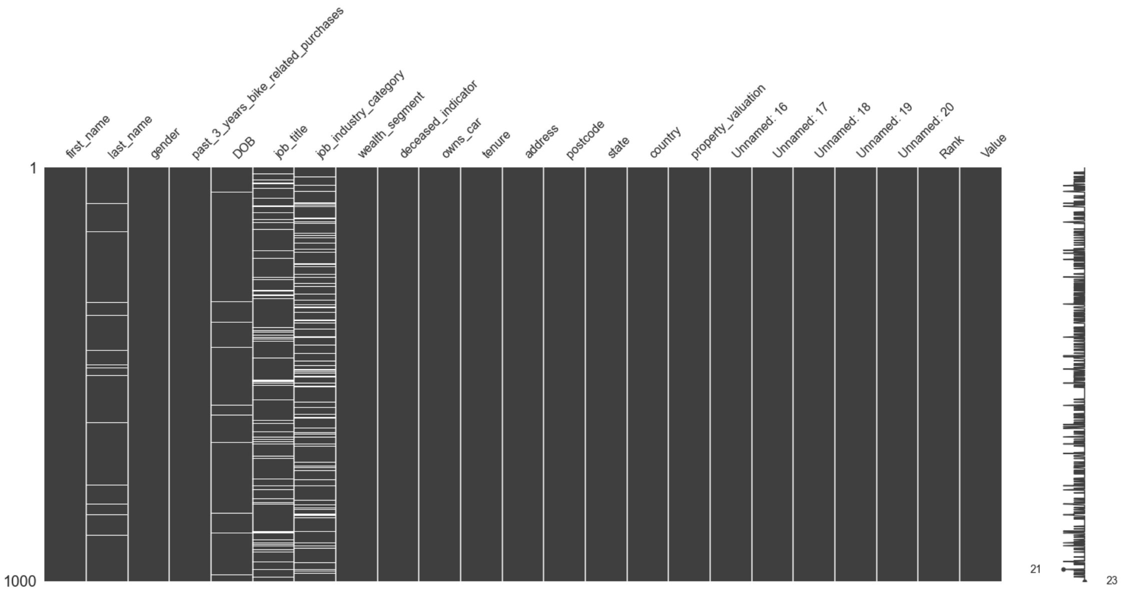
```
  first_name last_name gender  past_3_years_bike_related_purchases        DOB  \
0    Chickie   Brister   Male                                   86 1957-07-12
1      Morly    Genery   Male                                   69 1970-03-22

             job_title job_industry_category wealth_segment  \
0      General Manager         Manufacturing  Mass Customer
1  Structural Engineer              Property  Mass Customer

  deceased_indicator owns_car  …  state    country  property_valuation  \
0                  N      Yes  …    QLD  Australia                   6
1                  N       No  …    NSW  Australia                  11

  Unnamed: 16 Unnamed: 17  Unnamed: 18  Unnamed: 19  Unnamed: 20  Rank  \
0        0.56        0.70       0.8750     0.743750            1     1
1        0.89        0.89       1.1125     0.945625            1     1

     Value
0  1.71875
1  1.71875

[2 rows x 23 columns]
```

[576]:

```python
# 1
assert newCustomer.duplicated().sum() == 0
```

[577]:

```python
# 2
msno.matrix(newCustomer);
```



[578]:

```
# 3
newCustomer['gender'] = newCustomer['gender'].replace('U', np.nan)
newCustomer['DOB'] = pd.to_datetime(newCustomer['DOB'], errors='coerce')
now = pd.to_datetime('now')
now
# Timestamp('2019-04-14 00:00:43.105892')

newCustomer['age'] = ((now - newCustomer['DOB']).astype('<m8[Y]'))
newCustomer['age_group'] = pd.cut(newCustomer.age, bins = [10, 19, 29, 39, 49,␣
 ↪59, 69, 79, 90])

newCustomer['bigger_age_group'] = pd.cut(newCustomer.age, bins = [10, 30, 50, ␣
 ↪70, 90]).astype(str)
```

[579]:

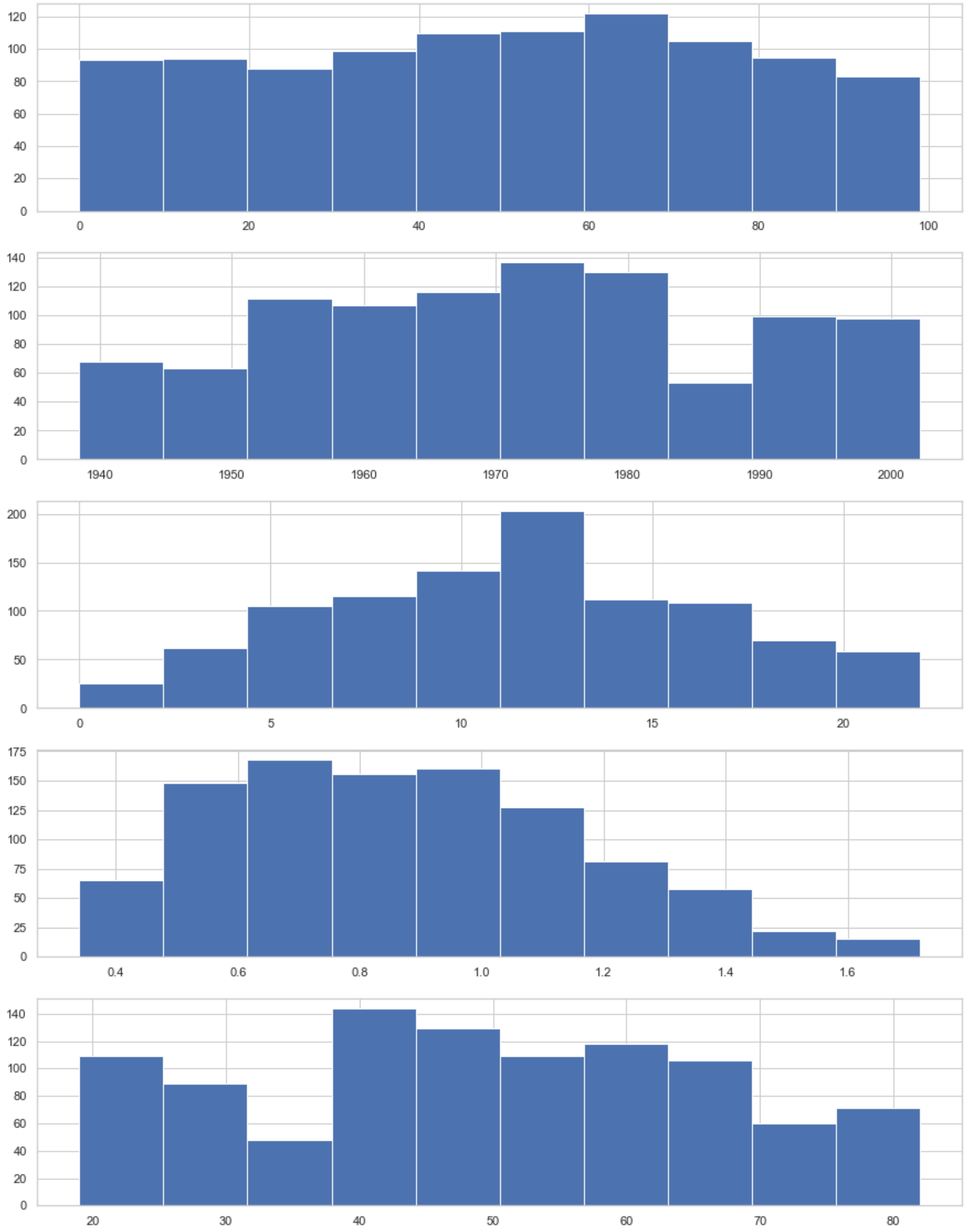
```
# 4
print(newCustomer.DOB.min(), newCustomer.DOB.max())
print('--------------------')
print(newCustomer.gender.value_counts())
#print('--------------------')
#print(newCustomer.job_title.value_counts())
print('--------------------')
print(newCustomer.job_industry_category.value_counts())
print('--------------------')
print(newCustomer.wealth_segment.value_counts())
print('--------------------')
print(newCustomer.deceased_indicator.value_counts())
print('--------------------')
print(newCustomer.owns_car.value_counts())
print('-------------------')
print(newCustomer.state.value_counts())
print('-------------------')
print(newCustomer.country.value_counts())
```

```
1938-06-08 00:00:00 2002-02-27 00:00:00
-------------------
Female    513
Male      470
Name: gender, dtype: int64
-------------------
Financial Services    203
Manufacturing         199
Health                152
Retail                 78
Property               64
IT                     51
Entertainment          37
Argiculture            26
Telecommunications     25
Name: job_industry_category, dtype: int64
-------------------
Mass Customer        508
High Net Worth       251
Affluent Customer    241
Name: wealth_segment, dtype: int64
-------------------
N    1000
Name: deceased_indicator, dtype: int64
-------------------
No     507
Yes    493
Name: owns_car, dtype: int64
-------------------
NSW    506
VIC    266
QLD    228
Name: state, dtype: int64
-------------------
Australia    1000
Name: country, dtype: int64
```

[580]:

```
plt.figure(figsize = (15, 20))
plt.subplot(5,1,1)
newCustomer.past_3_years_bike_related_purchases.hist();
plt.subplot(5,1,2)
```

```
newCustomer.DOB.hist();
plt.subplot(5,1,3)
newCustomer.tenure.hist();
plt.subplot(5,1,4)
newCustomer.Value.hist();
plt.subplot(5,1,5)
newCustomer.age.hist();
```



0.1.3 Data Insight

Comparing old customers vs potential customers

[581]:
```
# import the data
customerDemo = pd.read_csv('data/customerDemographic.csv')
customerAddress = pd.read_csv('data/customerAddress.csv')
transaction = pd.read_csv('data/transactions.csv')
```

[582]:
```
# joining dataframes
customer = customerDemo.merge(customerAddress, on = 'customer_id', how = 'left')
```

[583]:
```
customer.head(2)
```

[583]:
```
   customer_id first_name  last_name  gender  \
0            1    Laraine  Medendorp  Female
1            2        Eli    Bockman    Male

   past_3_years_bike_related_purchases        DOB               job_title  \
0                                   93 1953-10-12     Executive Secretary
1                                   81 1980-12-16  Administrative Officer

  job_industry_category wealth_segment deceased_indicator  \
0                Health  Mass Customer                  N
1    Financial Services  Mass Customer                  N

                        default owns_car  tenure              address  \
0                            "'      Yes    11.0   060 Morning Avenue
1  <script>alert('hi')</script>      Yes    16.0  6 Meadow Vale Court

   postcode            state    country  property_valuation
0    2016.0  New South Wales  Australia                10.0
1    2153.0  New South Wales  Australia                10.0
```

[584]:
```
# add age
customer['DOB'] = pd.to_datetime(customer['DOB'], errors='coerce')
now = pd.to_datetime('now')
now
# Timestamp('2019-04-14 00:00:43.105892')

customer['age'] = ((now - customer['DOB']).astype('<m8[Y]'))
customer['age_group'] = pd.cut(customer.age, bins = [10, 19, 29, 39, 49, 59,␣
 ↪69, 79, 90])

customer['age_group'] = customer.age_group.astype(str)
```

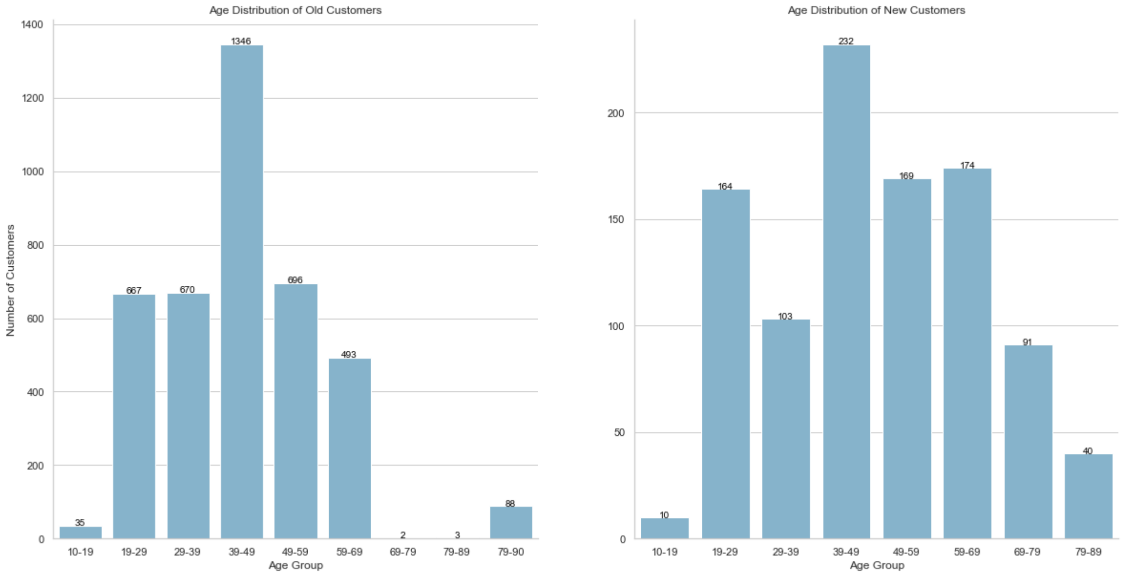
```
customer['bigger_age_group'] = pd.cut(customer.age, bins = [10, 30, 50,  70,␣
 ↪90]).astype(str)
```

0.1.4 Plots Age Distribution

[585]:

```
# age distribution

plt.figure(figsize = (20, 10))

# first plot
plt.subplot(121)

temp = customer.groupby('age_group')['customer_id'].count().reset_index()
#temp
g = sns.barplot(x = 'age_group', y = 'customer_id', data = temp, color = c);

for index, row in temp.iterrows():
    g.text(index, row.customer_id, round(row.customer_id,2), color='black',␣
 ↪ha="center", fontsize=10)

g.set(xlabel='Age Group', ylabel='Number of Customers', title = 'Age␣
 ↪Distribution of Old Customers')

g.set_xticks(range(len(temp))) # <--- set the ticks first
g.
 ↪set_xticklabels(['10-19','19-29','29-39','39-49','49-59','59-69','69-79','79-89',␣
 ↪'79-90'])


# second plot
plt.subplot(122)

temp = newCustomer.groupby('age_group').size().to_frame('size').reset_index()
#temp
g = sns.barplot(x = 'age_group', y = 'size', data = temp, color = c);
#g.text(row.age_group,row.customer_id, round(row.customer_id,2), color='black',␣
 ↪ha="center")
for index, row in temp.iterrows():
    g.text(index, row['size'], round(row['size'],2), color='black',␣
 ↪ha="center", fontsize=10)
g.set(xlabel='Age Group', ylabel='', title = 'Age Distribution of New␣
 ↪Customers')

g.set_xticks(range(len(temp))) # <--- set the ticks first
```

```
g.
 ↪set_xticklabels(['10-19','19-29','29-39','39-49','49-59','59-69','69-79','79-89',␣
 ↪'79-90'])


sns.despine();
```



Age Distribution of Old Customers

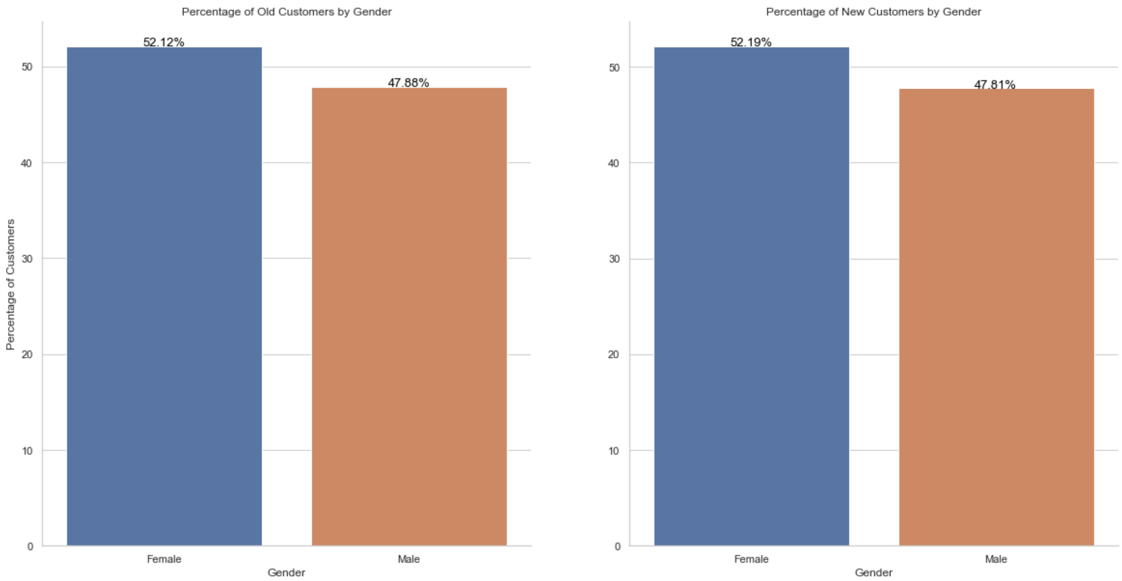
Age Distribution of New Customers

```
[586]: c = sns.color_palette("RdBu", 10)[-3]
```

0.1.5 Plots Gender

```
[587]: # Gender
#newCustomer.groupby('gender')['past_3_years_bike_related_purchases'].sum()
plt.figure(figsize = (20, 10))
# first plot
plt.subplot(121)
temp = customer\
.groupby('gender')\
.size()\
.to_frame('size')\
.apply(lambda x: 100*x/x.sum())\
.reset_index()

g = sns.barplot(x = 'gender', y = 'size', data = temp);
for index, row in temp.iterrows():
    g.text(index, row['size'], str(round(row['size'],2))+'%', color='black',␣
 ↪ha="center", fontsize=13)
```

```
g.set(xlabel='Gender', ylabel='Percentage of Customers', title = 'Percentage of Old Customers by Gender') ;

# second plot
plt.subplot(122)

#plt.figure(figsize = (10, 10))

temp = newCustomer\
.groupby('gender')\
.size()\
.to_frame('size')\
.apply(lambda x: 100*x/x.sum())\
.reset_index()

g = sns.barplot(x = 'gender', y = 'size', data = temp);
for index, row in temp.iterrows():
    g.text(index, row['size'], str(round(row['size'],2))+'%', color='black', ha="center", fontsize=13);
g.set(xlabel='Gender', ylabel='', title = 'Percentage of New Customers by Gender') ;

sns.despine();
```

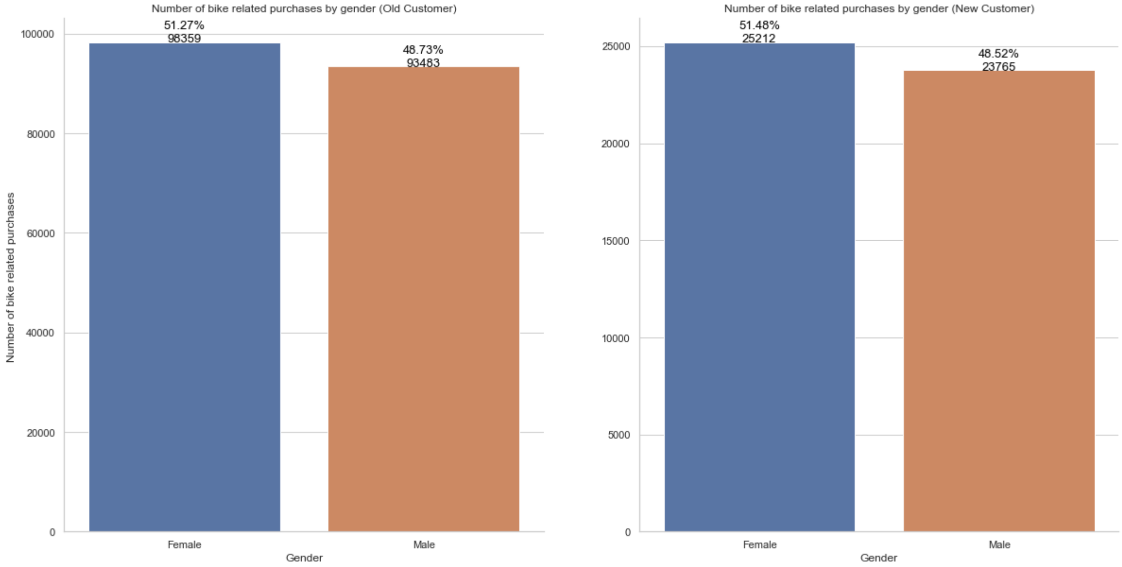


0.1.6 Plot Number of Purchases by Gender

[588]:

```
plt.figure(figsize = (20, 10))

# first plot
plt.subplot(121)
temp = customer\
.groupby('gender')['past_3_years_bike_related_purchases']\
.sum()\
.to_frame('sum')\
.apply(lambda x: x)\
.reset_index()

temp['percentage'] =  100 * temp['sum']  / temp['sum'].sum()

g = sns.barplot(x = 'gender', y = 'sum', data = temp);

for index, row in temp.iterrows():
    g.text(index, row['sum'],␣
 ↪str(round(row['percentage'],2))+'%'+'\n'+str(row['sum']), color='black',␣
 ↪ha="center", fontsize=13)

g.set(xlabel='Gender', ylabel='Number of bike related purchases ', title =␣
 ↪'Number of bike related purchases by gender (Old Customer)') ;

# second plot
plt.subplot(122)

temp = newCustomer\
.groupby('gender')['past_3_years_bike_related_purchases']\
.sum()\
.to_frame('sum')\
.apply(lambda x: x)\
.reset_index()

temp['percentage'] =  100 * temp['sum']  / temp['sum'].sum()

g = sns.barplot(x = 'gender', y = 'sum', data = temp);

for index, row in temp.iterrows():
    g.text(index, row['sum'],␣
 ↪str(round(row['percentage'],2))+'%'+'\n'+str(row['sum']), color='black',␣
 ↪ha="center", fontsize=13)

g.set(xlabel='Gender',ylabel = '', title = 'Number of bike related purchases by␣
 ↪gender (New Customer)') ;
```

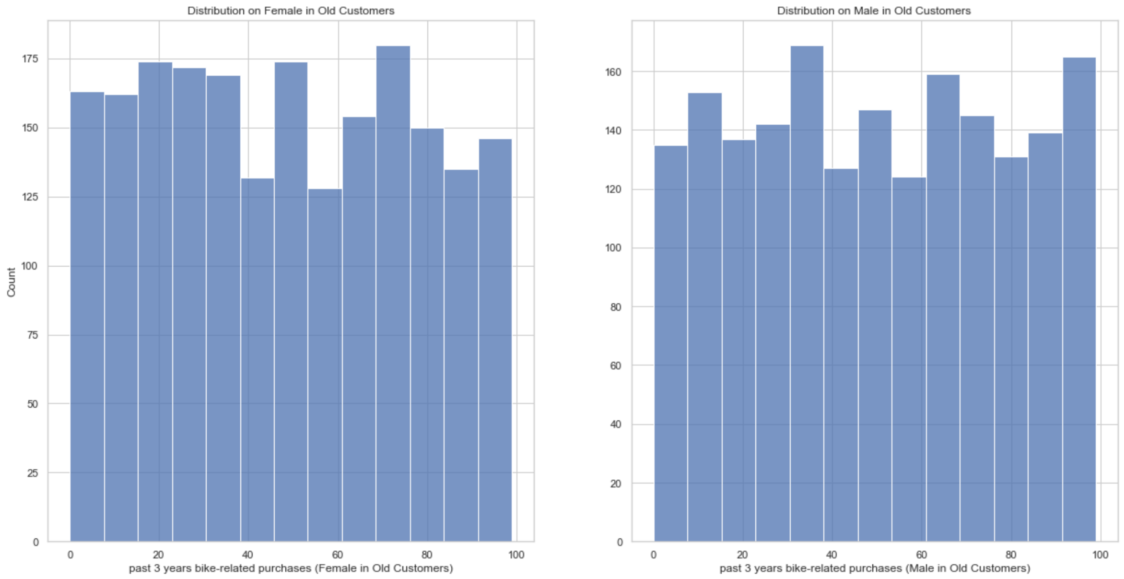
```
sns.despine();
```



By looking at plots, two questions raised. 1. Is male making more bike related purchases than female? 2. Is the new cohort making more bike related purchases than female?

[589]:

```
oldFemale_purchases = customer[customer.gender == 'Female']['past_3_years_bike_related_purchases']
oldMale_purchases = customer[customer.gender == 'Male']['past_3_years_bike_related_purchases']

plt.figure(figsize = (20, 10))
plt.subplot(121)
g = sns.histplot(oldFemale_purchases);
g.set(xlabel = 'past 3 years bike-related purchases (Female in Old Customers)', ylabel = 'Count', title = 'Distribution on Female in Old Customers');
plt.subplot(122)
g = sns.histplot(oldMale_purchases);
g.set(xlabel = 'past 3 years bike-related purchases (Male in Old Customers)', ylabel = '', title = 'Distribution on Male in Old Customers');
```

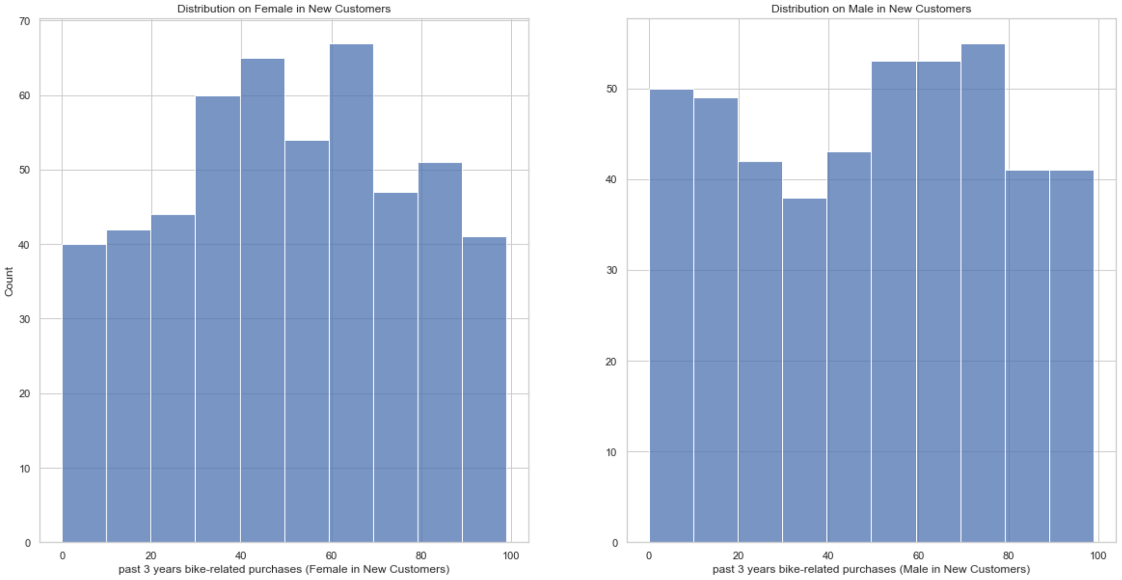


```
[590]: sm.stats.ttest_ind(oldFemale_purchases, oldMale_purchases)
```

```
[590]: (-1.8164681244288128, 0.0693751314702939, 3910.0)
```

```
[591]: newFemale_purchases = newCustomer[customer.gender ==␣
        ↪'Female']['past_3_years_bike_related_purchases']
       newMale_purchases = newCustomer[customer.gender ==␣
        ↪'Male']['past_3_years_bike_related_purchases']

       plt.figure(figsize = (20, 10))
       plt.subplot(121)
       g = sns.histplot(newFemale_purchases);
       g.set(xlabel = 'past 3 years bike-related purchases (Female in New Customers)',␣
        ↪ylabel = 'Count', title = 'Distribution on Female in New Customers');
       plt.subplot(122)
       g = sns.histplot(newMale_purchases);
       g.set(xlabel = 'past 3 years bike-related purchases (Male in New Customers)',␣
        ↪ylabel = '', title = 'Distribution on Male in New Customers');
```

```
/Users/murongcui/opt/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
/Users/murongcui/opt/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:2: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
```



[592]:

```
sm.stats.ttest_ind(newFemale_purchases, newMale_purchases)
```

[592]:

```
(0.6894517491175025, 0.4907032432436518, 974.0)
```

Hypothesis:

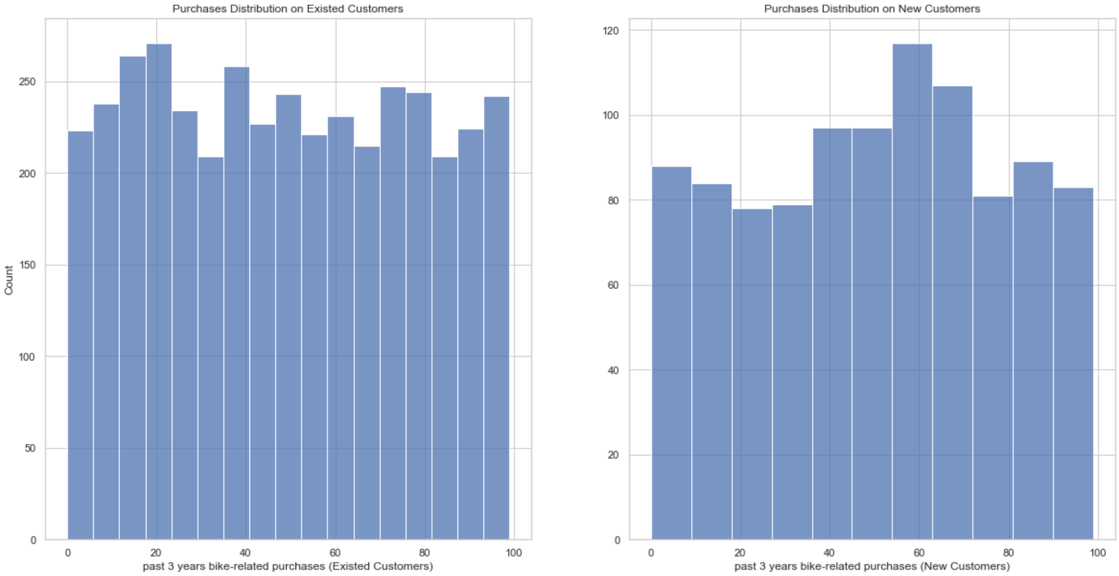
$$H_0 : \mu_{female} = \mu_{male}$$

$$H_a : \mu_{female} \neq \mu_{male}$$

Since the p-values is quite high, we cannot reject the Null hypothesis that the difference in the female group and male group is zero.

[593]:

```
oldCustomer_purchases = customer['past_3_years_bike_related_purchases']
newCustomer_purchases = newCustomer['past_3_years_bike_related_purchases']

plt.figure(figsize = (20, 10))
plt.subplot(121)
g = sns.histplot(oldCustomer_purchases);
g.set(xlabel = 'past 3 years bike-related purchases (Existed Customers)',␣
 ↪ylabel = 'Count', title = 'Purchases Distribution on Existed Customers');

plt.subplot(122)
g = sns.histplot(newCustomer_purchases);
g.set(xlabel = 'past 3 years bike-related purchases (New Customers)', ylabel =␣
 ↪'', title = 'Purchases Distribution on New Customers');
```



[594]:

```
sm.stats.ttest_ind(oldCustomer_purchases, newCustomer_purchases)
```

[594]:

```
(-0.9377267068386276, 0.3484301857683023, 4998.0)
```

Hypothesis:

$$H_0 : \mu_{old} = \mu_{new}$$

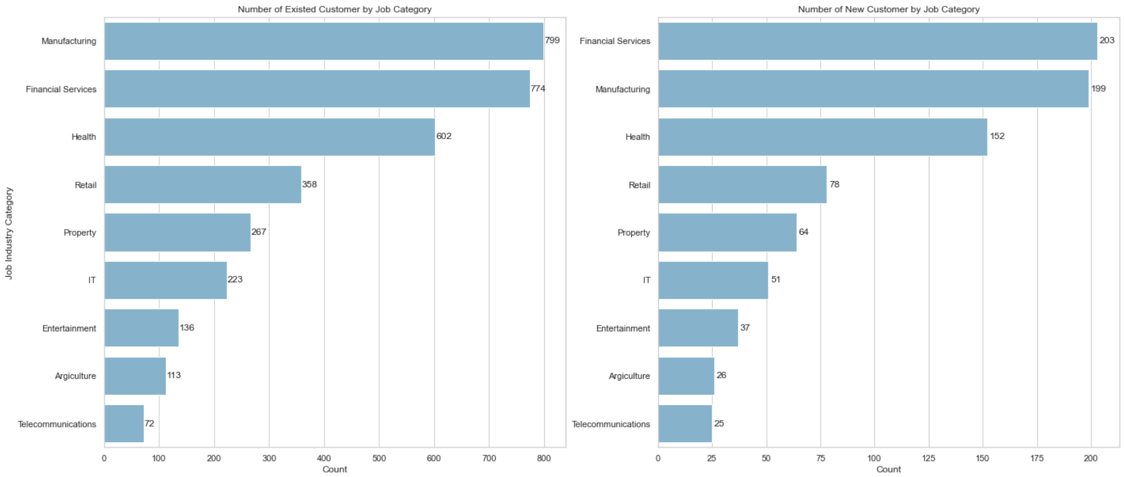
$$H_a : \mu_{old} \neq \mu_{new}$$

Since the p-value (~0.348) is quite high, we cannot reject the Null hypothesis that the difference in the existed customers and new customers is zero.

0.1.7 Job Industry Category

[595]:

```
plt.figure(figsize = (23, 10))
# first plot
plt.subplot(121)

temp = customer\
.groupby('job_industry_category')\
.size()\
.to_frame('size')\
.sort_values(['size'], ascending=False)\
.reset_index()

g = sns.barplot(y = 'job_industry_category', x = 'size', data = temp, color =␣
 ↪c);

for i in range(9):
    count = temp['size'][i]
    pct_string = '{:d}'.format(count)
    plt.text(count + 1, i, pct_string, va='center')
g.set(ylabel = 'Job Industry Category', xlabel = 'Count', title = 'Number of␣
 ↪Existed Customer by Job Category') ;


# second plot
plt.subplot(122)

temp = newCustomer\
.groupby('job_industry_category')\
.size()\
.to_frame('size')\
.sort_values(['size'], ascending=False)\
.reset_index()

g = sns.barplot(y = 'job_industry_category', x = 'size', data = temp, color =␣
 ↪c);

for i in range(9):
    count = temp['size'][i]
    pct_string = '{:d}'.format(count)
    plt.text(count + 1, i, pct_string, va='center')
g.set(ylabel = '', xlabel = 'Count', title = 'Number of New Customer by Job␣
 ↪Category') ;
```



Among the Existed Customers, the top 1 job industry is Manufacturing and top 2 is Financial Services. Mostly our new customers are still in Fiancial Service and Our Manufacturing is on the second place.

The rest industries are stay on the same order.

[596]:

```
temp = customer.groupby(['bigger_age_group', 'wealth_segment']).size().
 ↪to_frame('size').unstack()
temp = temp.drop('nan')
temp = temp.fillna(0)

temp['size', 'sum'] = temp['size'].sum(axis = 1)

temp = temp['size']


temp1 = newCustomer.groupby(['bigger_age_group', 'wealth_segment']).size().
 ↪to_frame('size').unstack()
temp1 = temp1.drop('nan')
temp1 = temp1.fillna(0)

temp1['size', 'sum'] = temp1['size'].sum(axis = 1)

temp1 = temp1['size']

temp1
```

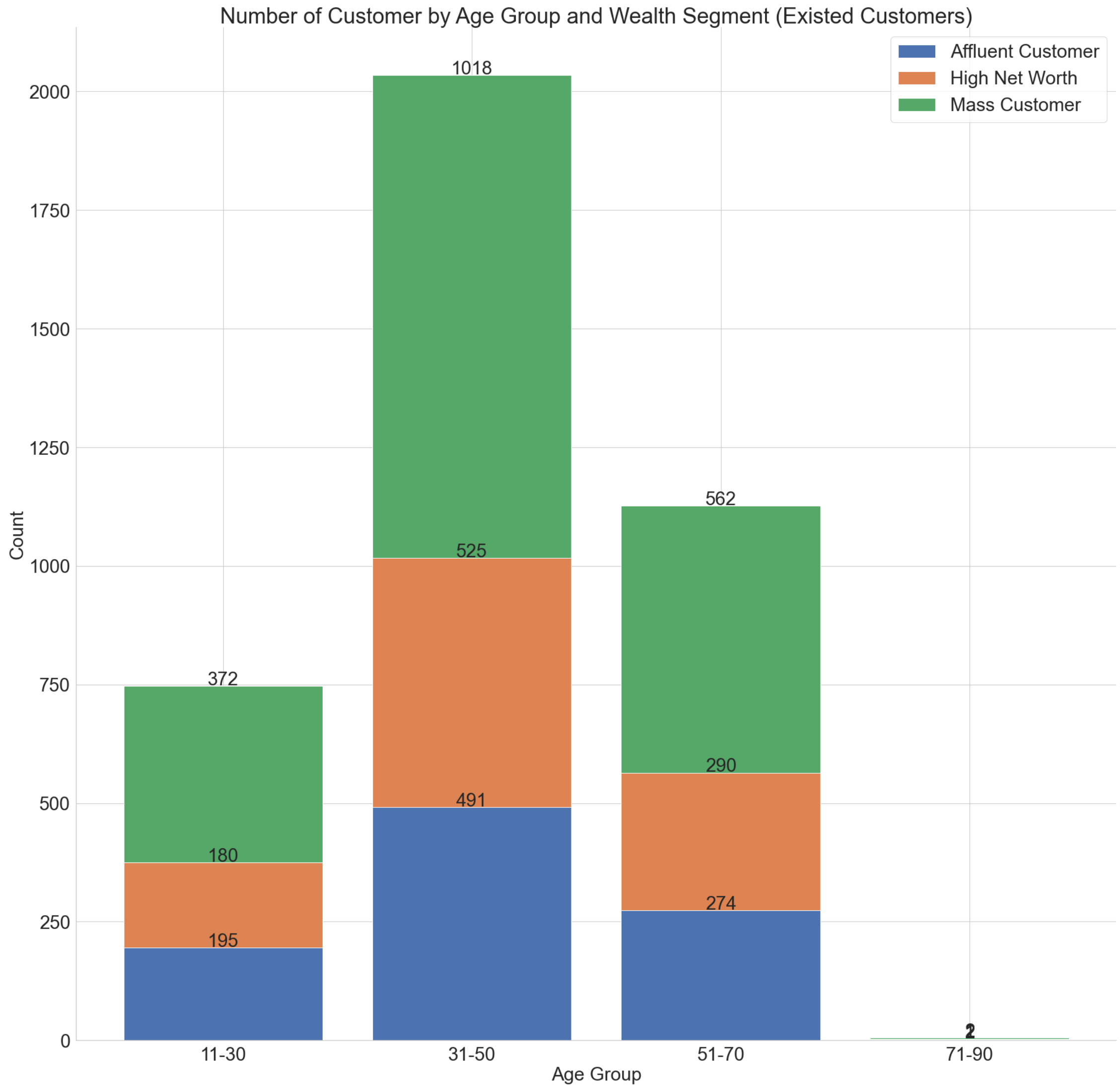
[596]:

```
wealth_segment    Affluent Customer  High Net Worth  Mass Customer  sum
bigger_age_group
(10.0, 30.0]                     55              49             89  193
(30.0, 50.0]                     74              82            170  326
(50.0, 70.0]                     82              92            179  353
(70.0, 90.0]                     24              26             61  111
```

[597]:

```
fig, ax = plt.subplots(figsize = (30, 30))


# First Plot

ax.bar(temp.index, temp['Affluent Customer'], label = 'Affluent Customer');
ax.bar(temp.index, temp['High Net Worth'], bottom=temp['Affluent Customer'],
       label='High Net Worth')

ax.bar(temp.index, temp['Mass Customer'], bottom=temp['High Net Worth'] +␣
 ↪temp['Affluent Customer'],
       label='Mass Customer')

ax.set_xticks(range(len(temp))); # <--- set the ticks first
ax.set_xticklabels(['11-30', '31-50', '51-70', '71-90']);
#plt.xticks(rotation = 0);
```

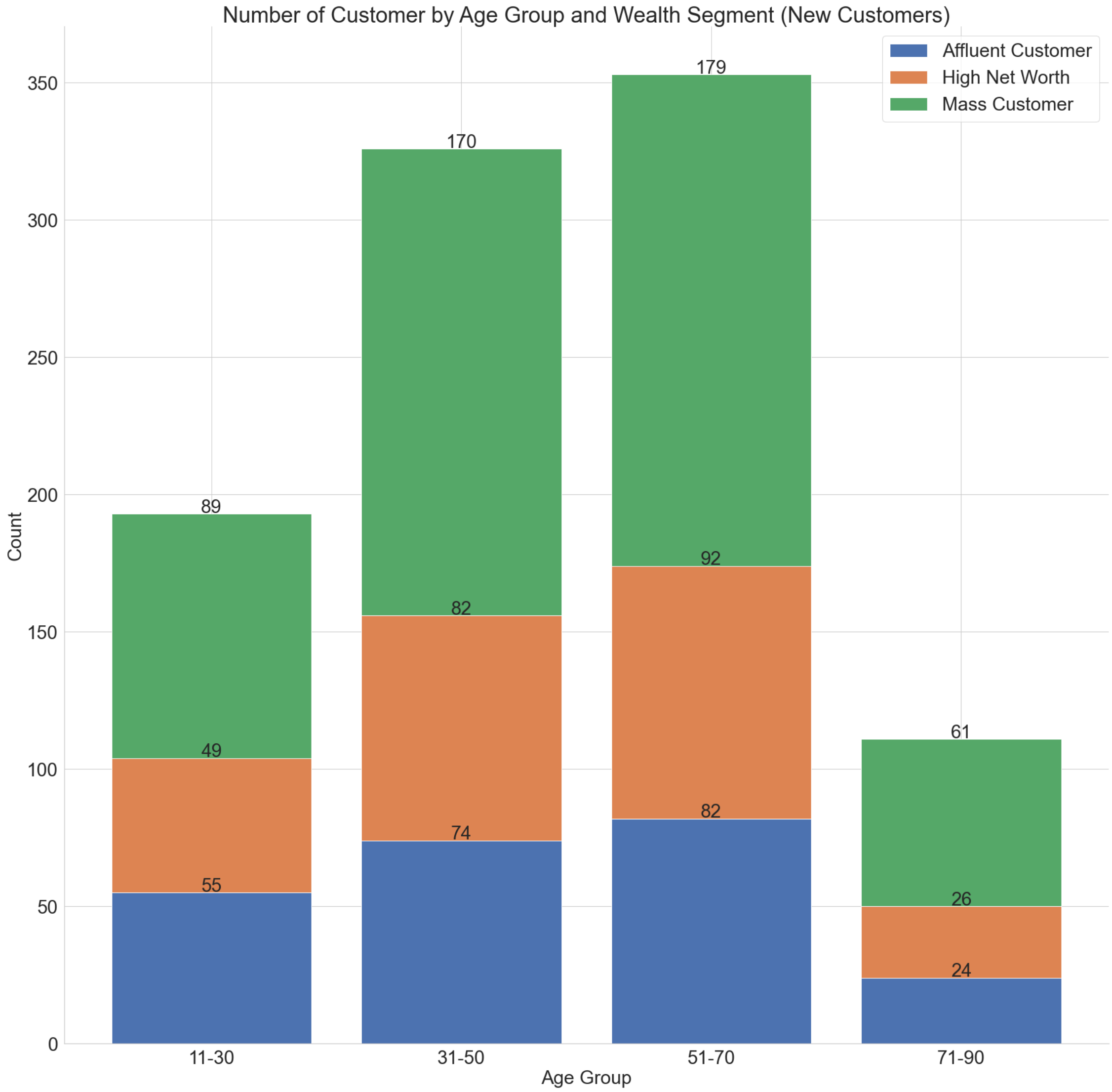
```
# -------------------------stack␣
 ↪annotation---------------------------------------
x = []
y = []
width = []
height = []
value = []
for i in range(len(ax.patches)):
    if i <= 3:
        x.append(ax.patches[i].get_x())
        y.append(ax.patches[i].get_y())
        width.append(ax.patches[i].get_width())
        height.append(ax.patches[i].get_height())
        value.append(ax.patches[i].get_height())
    elif i <= 7:
            x.append(ax.patches[i].get_x())
            y.append(ax.patches[i].get_y())
            width.append(ax.patches[i].get_width())
            height.append(ax.patches[i].get_height() + ax.patches[i-4].
 ↪get_height())
            value.append(ax.patches[i].get_height())
    else:
        x.append(ax.patches[i].get_x())
        y.append(ax.patches[i].get_y())
        width.append(ax.patches[i].get_width())
        height.append(ax.patches[i].get_height() + ax.patches[i-4].get_height()␣
 ↪+ ax.patches[i-8].get_height())
        value.append(ax.patches[i].get_height())

annotation = pd.DataFrame({'x': x, 'y': y, 'width': width, 'height': height,␣
 ↪'value': value})
#␣
 ↪---------------------------------------------------------------------------------

# add annotation
for i, row in annotation.iterrows():
    ax.annotate(s=int(row.value),
                xy=(row.x+row.width/2., row.height),
                ha='center',
                va='center',
                xytext=(0, 10),
                textcoords='offset points',
               fontsize=30)
plt.xticks(fontsize= 30)
plt.yticks(fontsize= 30)
plt.xlabel('Age Group', fontsize = 30)
plt.ylabel('Count', fontsize = 30)
plt.title('Number of Customer by Age Group and Wealth Segment (Existed␣
↪Customers)', fontsize = 35)
ax.legend(fontsize = 30)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
```



[598]:

```
fig, ax = plt.subplots(figsize = (30, 30))

# Second Plot

ax.bar(temp1.index, temp1['Affluent Customer'], label = 'Affluent Customer');
ax.bar(temp1.index, temp1['High Net Worth'], bottom=temp1['Affluent Customer'],
       label='High Net Worth')
```

```
ax.bar(temp1.index, temp1['Mass Customer'], bottom=temp1['High Net Worth'] +␣
 ↪temp1['Affluent Customer'],
       label='Mass Customer')

ax.set_xticks(range(len(temp1))); # <--- set the ticks first
ax.set_xticklabels(['11-30', '31-50', '51-70', '71-90']);
#plt.xticks(rotation = 0);

# -------------------------stack␣
 ↪annotation---------------------------------------
x = []
y = []
width = []
height = []
value = []
for i in range(len(ax.patches)):
    if i <= 3:
        x.append(ax.patches[i].get_x())
        y.append(ax.patches[i].get_y())
        width.append(ax.patches[i].get_width())
        height.append(ax.patches[i].get_height())
        value.append(ax.patches[i].get_height())
    elif i <= 7:
            x.append(ax.patches[i].get_x())
            y.append(ax.patches[i].get_y())
            width.append(ax.patches[i].get_width())
            height.append(ax.patches[i].get_height() + ax.patches[i-4].
 ↪get_height())
            value.append(ax.patches[i].get_height())
    else:
        x.append(ax.patches[i].get_x())
        y.append(ax.patches[i].get_y())
        width.append(ax.patches[i].get_width())
        height.append(ax.patches[i].get_height() + ax.patches[i-4].get_height()␣
 ↪+ ax.patches[i-8].get_height())
        value.append(ax.patches[i].get_height())

annotation = pd.DataFrame({'x': x, 'y': y, 'width': width, 'height': height,␣
 ↪'value': value})
#␣
 ↪---------------------------------------------------------------------------------

# add annotation
for i, row in annotation.iterrows():
    ax.annotate(s=int(row.value),
                xy=(row.x+row.width/2., row.height),
                ha='center',
                va='center',
                xytext=(0, 10),
                textcoords='offset points',
               fontsize=30)
plt.xticks(fontsize= 30)
plt.yticks(fontsize= 30)
plt.xlabel('Age Group', fontsize = 30)
plt.ylabel('Count', fontsize = 30)
plt.title('Number of Customer by Age Group and Wealth Segment (New Customers)',␣
 ↪fontsize = 35)
ax.legend(fontsize = 30)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
```



0.1.8 plot Owns Car

[599]:

```
print(newCustomer.owns_car.value_counts())
print(customer.owns_car.value_counts())
customer.head()
```

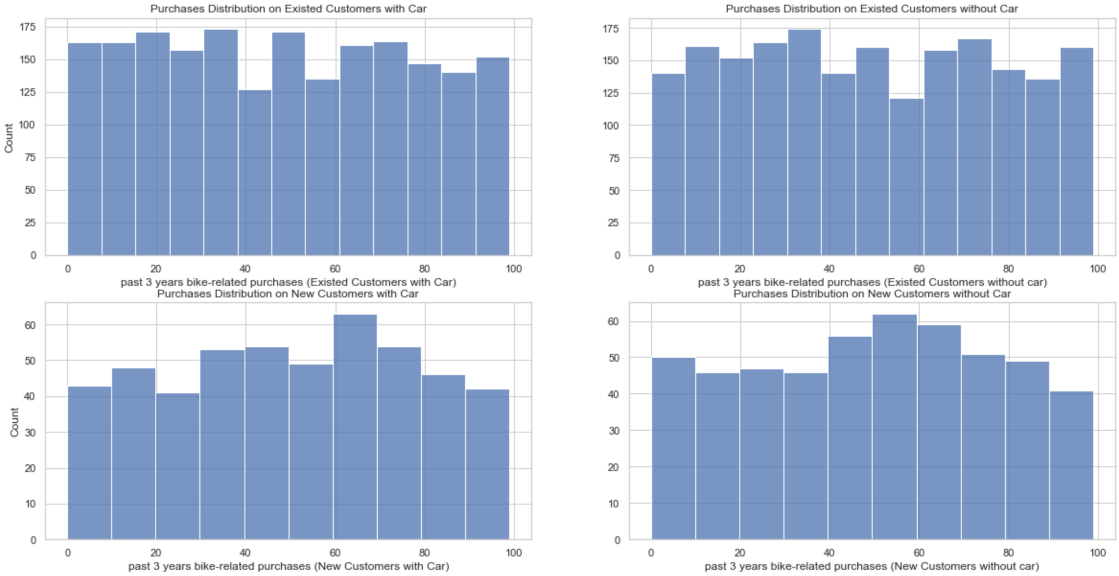
```
No      507
Yes     493
Name: owns_car, dtype: int64
Yes     2024
No      1976
Name: owns_car, dtype: int64
```

[599]:

```
   customer_id      first_name  last_name  gender  \
0            1         Laraine  Medendorp  Female
1            2             Eli    Bockman    Male
2            3           Arlin     Dearle    Male
3            4          Talbot        NaN    Male
4            5  Sheila-kathryn     Calton  Female

   past_3_years_bike_related_purchases        DOB               job_title  \
0                                   93 1953-10-12     Executive Secretary
1                                   81 1980-12-16  Administrative Officer
2                                   61 1954-01-20      Recruiting Manager
3                                   33 1961-10-03                     NaN
4                                   56 1977-05-13           Senior Editor

  job_industry_category     wealth_segment deceased_indicator  … owns_car  \
0                Health      Mass Customer                  N  …      Yes
1    Financial Services      Mass Customer                  N  …      Yes
2              Property      Mass Customer                  N  …      Yes
3                    IT      Mass Customer                  N  …       No
4                   NaN  Affluent Customer                  N  …      Yes

  tenure              address postcode            state    country  \
0   11.0    060 Morning Avenue   2016.0  New South Wales  Australia
1   16.0   6 Meadow Vale Court   2153.0  New South Wales  Australia
2   15.0                   NaN      NaN              NaN        NaN
3    7.0    0 Holy Cross Court   4211.0              QLD  Australia
4    8.0   17979 Del Mar Point   2448.0  New South Wales  Australia

  property_valuation   age     age_group bigger_age_group
0               10.0  67.0  (59.0, 69.0]     (50.0, 70.0]
1               10.0  40.0  (39.0, 49.0]     (30.0, 50.0]
2                NaN  67.0  (59.0, 69.0]     (50.0, 70.0]
3                 9.0  59.0  (49.0, 59.0]     (50.0, 70.0]
4                 4.0  43.0  (39.0, 49.0]     (30.0, 50.0]

[5 rows x 21 columns]
```

[600]:

```
oldCustomer_car = customer[customer.owns_car ==␣
 ↪'Yes']['past_3_years_bike_related_purchases']
oldCustomer_nocar = customer[customer.owns_car ==␣
 ↪'No']['past_3_years_bike_related_purchases']

newCustomer_car = newCustomer[newCustomer.owns_car ==␣
 ↪'Yes']['past_3_years_bike_related_purchases']
newCustomer_nocar = newCustomer[newCustomer.owns_car ==␣
 ↪'No']['past_3_years_bike_related_purchases']

plt.figure(figsize = (20, 10))
plt.subplot(221)
g = sns.histplot(oldCustomer_car);
g.set(xlabel = 'past 3 years bike-related purchases (Existed Customers with␣
 ↪Car)', ylabel = 'Count', title = 'Purchases Distribution on Existed␣
 ↪Customers with Car');

plt.subplot(222)
g = sns.histplot(oldCustomer_nocar);
g.set(xlabel = 'past 3 years bike-related purchases (Existed Customers without␣
 ↪car)', ylabel = '', title = 'Purchases Distribution on Existed Customers␣
 ↪without Car');


plt.subplot(223)
g = sns.histplot(newCustomer_car);
g.set(xlabel = 'past 3 years bike-related purchases (New Customers with Car)',␣
 ↪ylabel = 'Count', title = 'Purchases Distribution on New Customers with␣
 ↪Car');

plt.subplot(224)
g = sns.histplot(newCustomer_nocar);
g.set(xlabel = 'past 3 years bike-related purchases (New Customers without␣
 ↪car)', ylabel = '', title = 'Purchases Distribution on New Customers without␣
 ↪Car');
```



[601]:
```
sm.stats.ttest_ind(oldCustomer_car, oldCustomer_nocar)
```

[601]:
```
(-0.8439944210079887, 0.3987230194978, 3998.0)
```

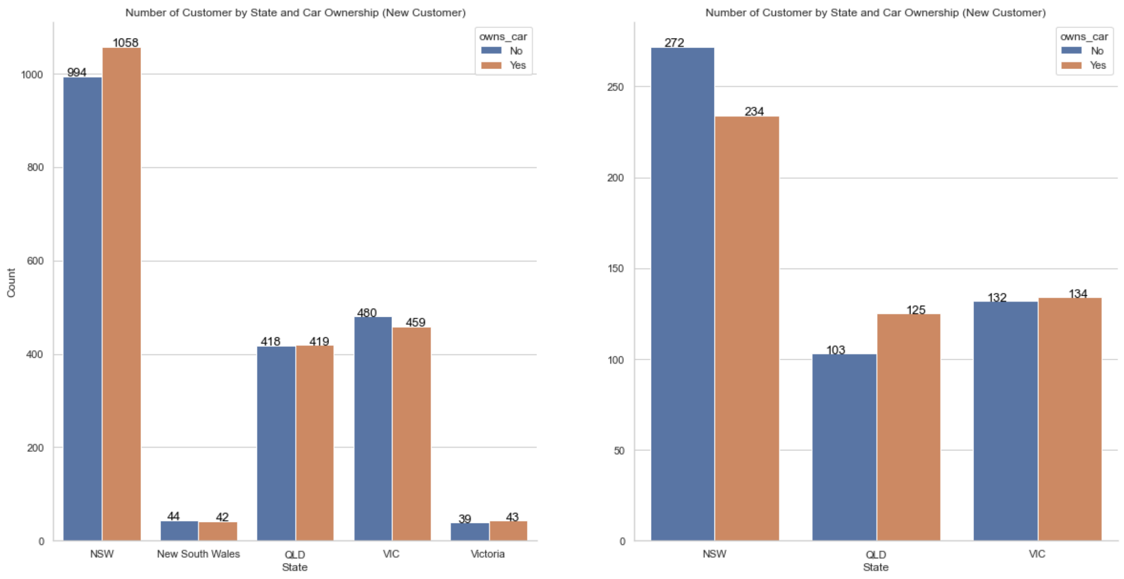
[602]:
```
sm.stats.ttest_ind(newCustomer_car, newCustomer_nocar)
```

[602]:
```
(0.2657746670166247, 0.790467646539127, 998.0)
```

0.2 Car Ownership by Stats

[603]:
```
plt.figure(figsize = (20, 10))

# first plot
plt.subplot(121)

temp = customer.groupby(['state', 'owns_car']).size().to_frame('size').
 ↪reset_index()

g = sns.barplot(x="state", y="size", hue="owns_car", data=temp)

for index, row in temp.iterrows():
    g.text(index/2 - 0.25, row['size'], str(round(row['size'])), color='black',␣
 ↪ha="center", fontsize=13);
g.set(xlabel='State',ylabel = 'Count', title = 'Number of Customer by State and␣
 ↪Car Ownership (New Customer)') ;
```

```python
# second plot
plt.subplot(122)

temp = newCustomer.groupby(['state', 'owns_car']).size().to_frame('size').
 ↪reset_index()

g = sns.barplot(x="state", y="size", hue="owns_car", data=temp)

for index, row in temp.iterrows():
    g.text(index/2 - 0.25, row['size'], str(round(row['size'])), color='black',␣
 ↪ha="center", fontsize=13);
g.set(xlabel='State',ylabel = '', title = 'Number of Customer by State and Car␣
 ↪Ownership (New Customer)') ;


sns.despine();
```



[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: