task2

January 14, 2022

```
[ ]: import pandas as pd
```

1 Data Cleaning

```
[ ]: cusdemo_df = pd.read_excel('rawdata.xlsx', 'CustomerDemographic')
     cusdemo_df.head()
```

```
/var/folders/25/53b25p9j7k52dz70pl14gl2w0000gn/T/ipykernel_8470/2139341786.py:1:
FutureWarning: Inferring datetime64[ns] from data containing strings is
deprecated and will be removed in a future version. To retain the old behavior
explicitly pass Series(data, dtype={value.dtype})
  cusdemo_df = pd.read_excel('rawdata.xlsx', 'CustomerDemographic')
```

```
[ ]:    customer_id      first_name  last_name  gender  \
     0            1         Laraine  Medendorp       F
     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  \
     0                Health      Mass Customer                  N
     1    Financial Services      Mass Customer                  N
     2              Property      Mass Customer                  N
     3                    IT      Mass Customer                  N
     4                   NaN  Affluent Customer                  N

                                                 default owns_car  tenure
     0                                                "'      Yes    11.0
     1                      <script>alert('hi')</script>      Yes    16.0
```

```
2                              2018-02-01 00:00:00      Yes    15.0
3  () { _; } >_[$($())] { touch /tmp/blns.shellsh…     No     7.0
4                                              NIL      Yes     8.0
```

```
[ ]: cusdemo_df['gender'].unique()
```

```
[ ]: array(['F', 'Male', 'Female', 'U', 'Femal', 'M'], dtype=object)
```

```
[ ]: cusdemo_df.isna().sum()
```

```
[ ]: customer_id                             0
     first_name                              0
     last_name                             125
     gender                                  0
     past_3_years_bike_related_purchases     0
     DOB                                    87
     job_title                             506
     job_industry_category                 656
     wealth_segment                          0
     deceased_indicator                      0
     default                               302
     owns_car                                0
     tenure                                 87
     dtype: int64
```

```
[ ]: cusdemo_df.drop(['first_name', 'last_name', 'default',
                     'job_title'], axis=1, inplace=True)
     cusdemo_df['gender'] = cusdemo_df['gender'].replace({'F': 'Female', 'Femal':␣
      ↪'Female', 'Female': 'Female', 'M': 'Male', 'Male': 'Male', 'U': 'Unknown' })
     cusdemo_df['owns_car'] = cusdemo_df['owns_car'].replace({'Yes': 1, 'No': 0}).
      ↪astype('int')
     cusdemo_df['deceased_indicator'] = cusdemo_df['deceased_indicator'].replace(
         {'Y': 1, 'N': 0}).astype('int')
     cusdemo_df.dropna(inplace=True)
     cusdemo_df.rename(columns={'past_3_years_bike_related_purchases':␣
      ↪'p3bkrel_pur'}, inplace = True)
     cusdemo_df = cusdemo_df.set_index('customer_id')
     cusdemo_df.head()
```

```
[ ]:              gender  p3bkrel_pur        DOB job_industry_category  \
     customer_id
     1            Female           93 1953-10-12                Health
     2              Male           81 1980-12-16    Financial Services
     3              Male           61 1954-01-20              Property
     4              Male           33 1961-10-03                    IT
     6              Male           35 1966-09-16                Retail
```

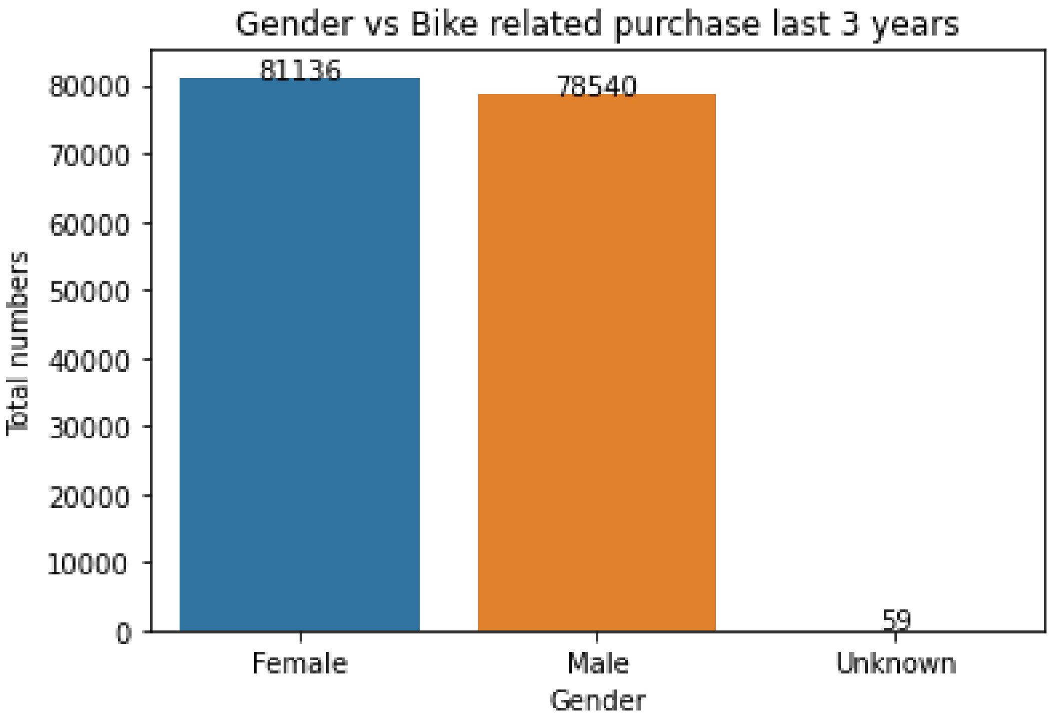
```
            wealth_segment  deceased_indicator  owns_car  tenure
customer_id
1            Mass Customer                   0         1    11.0
2            Mass Customer                   0         1    16.0
3            Mass Customer                   0         1    15.0
4            Mass Customer                   0         0     7.0
6           High Net Worth                   0         1    13.0
```

[ ]:

```
import seaborn as sns
from matplotlib import pyplot as plt


def addlabels(x, y):
    for i in range(len(x)):
        plt.text(i, y[i], y[i], ha = 'center')


y =cusdemo_df.groupby('gender')['p3bkrel_pur'].sum().rename('count').
 ↪reset_index()
sns.barplot(x = 'gender', y = 'count', data= y)
addlabels(y['gender'], y['count'])
plt.title('Gender vs Bike related purchase last 3 years')
plt.xlabel('Gender')
plt.ylabel('Total numbers')
```

[ ]:

```
Text(0, 0.5, 'Total numbers')
```



[ ]:

```
#differentiating age bracket column
import datetime as dt
import numpy as np
cusdemo_df['age'] = ((
    dt.datetime.now() - cusdemo_df['DOB']) / np.timedelta64(1, 'Y')).round(0).
 ↪astype(int)
cusdemo_df['age_bracket'] = (
    (round(cusdemo_df['age'] / 10)) * 10).astype(int)

cusdemo_df.head()
```

[ ]:

```
             gender  p3bkrel_pur        DOB job_industry_category  \
customer_id
1            Female           93 1953-10-12                Health
2              Male           81 1980-12-16    Financial Services
3              Male           61 1954-01-20              Property
4              Male           33 1961-10-03                    IT
6              Male           35 1966-09-16                Retail

             wealth_segment  deceased_indicator  owns_car  tenure  age  \
customer_id
1             Mass Customer                   0         1    11.0   68
2             Mass Customer                   0         1    16.0   41
3             Mass Customer                   0         1    15.0   68
4             Mass Customer                   0         0     7.0   60
6            High Net Worth                   0         1    13.0   55

             age_bracket
customer_id
1                     70
2                     40
3                     70
4                     60
6                     60
```

[ ]:

```
cusadd_df = pd.read_excel('rawdata.xlsx', 'CustomerAddress', index_col=0)
cusadd_df.head()
```

[ ]:

```
                         address  postcode            state    country  \
customer_id
1             060 Morning Avenue      2016  New South Wales  Australia
2            6 Meadow Vale Court      2153  New South Wales  Australia
4             0 Holy Cross Court      4211              QLD  Australia
5            17979 Del Mar Point      2448  New South Wales  Australia
6               9 Oakridge Court      3216              VIC  Australia
```

```
             property_valuation
customer_id
1                            10
2                            10
4                             9
5                             4
6                             9
```

```
[ ]: cusadd_df['state'].unique()
```

```
[ ]: array(['New South Wales', 'QLD', 'VIC', 'NSW', 'Victoria'], dtype=object)
```

```
[ ]: cusadd_df['state'] = cusadd_df['state'].replace(
         {'New South Wales': 'NSW', 'QLD': 'QLD', 'VIC': 'VIC', 'NSW': 'NSW',␣
      ↪'Victoria': 'VIC'})
     cusadd_df.head()
```

```
[ ]:                          address  postcode state    country  \
     customer_id
     1             060 Morning Avenue      2016   NSW  Australia
     2            6 Meadow Vale Court      2153   NSW  Australia
     4             0 Holy Cross Court      4211   QLD  Australia
     5            17979 Del Mar Point      2448   NSW  Australia
     6               9 Oakridge Court      3216   VIC  Australia

                  property_valuation
     customer_id
     1                            10
     2                            10
     4                             9
     5                             4
     6                             9
```

```
[ ]: df1 = pd.merge(cusdemo_df, cusadd_df, left_index= True, right_index= True )
     df1.head()
```

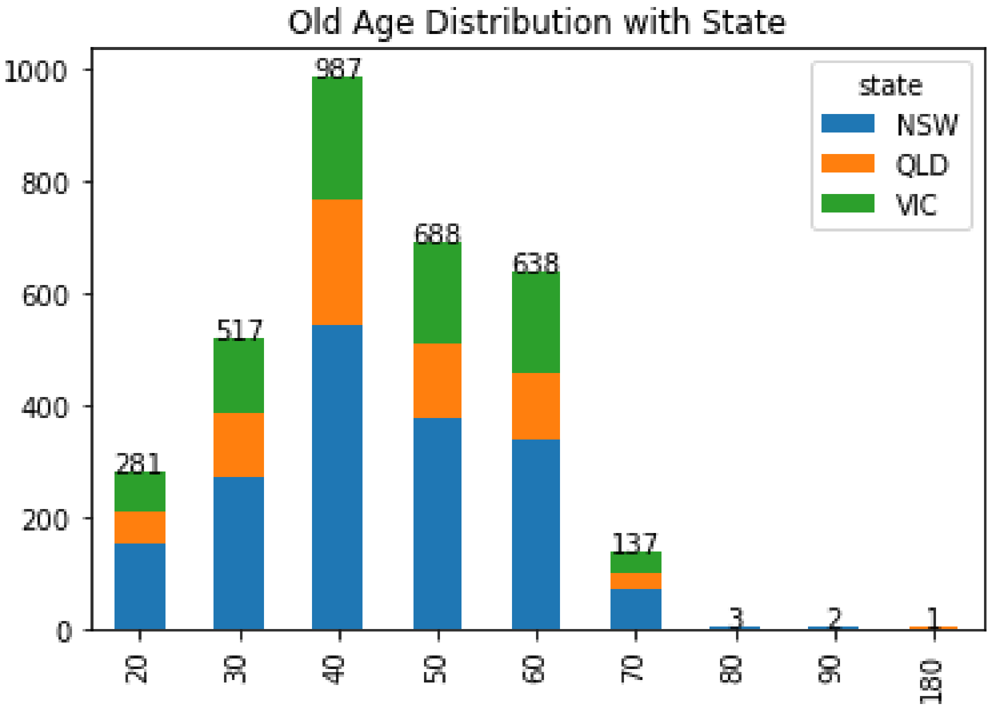
```
[ ]:              gender  p3bkrel_pur        DOB job_industry_category  \
     customer_id
     1            Female           93 1953-10-12                Health
     2              Male           81 1980-12-16    Financial Services
     4              Male           33 1961-10-03                    IT
     6              Male           35 1966-09-16                Retail
     7            Female            6 1976-02-23    Financial Services

                     wealth_segment  deceased_indicator  owns_car  tenure  age  \
     customer_id
```

```
1                 Mass Customer                   0         1     11.0    68
2                 Mass Customer                   0         1     16.0    41
4                 Mass Customer                   0         0      7.0    60
6                High Net Worth                   0         1     13.0    55
7             Affluent Customer                   0         1     11.0    46

             age_bracket              address  postcode state    country  \
customer_id
1                     70    060 Morning Avenue      2016   NSW  Australia
2                     40   6 Meadow Vale Court      2153   NSW  Australia
4                     60    0 Holy Cross Court      4211   QLD  Australia
6                     60      9 Oakridge Court      3216   VIC  Australia
7                     50      4 Delaware Trail      2210   NSW  Australia

             property_valuation
customer_id
1                            10
2                            10
4                             9
6                             9
7                             9
```

```
[ ]: z = df1.groupby('age_bracket')['state'].value_counts().rename('count').
      ↪reset_index()
     tt = z.groupby('age_bracket')['count'].sum().rename('total').reset_index()
```

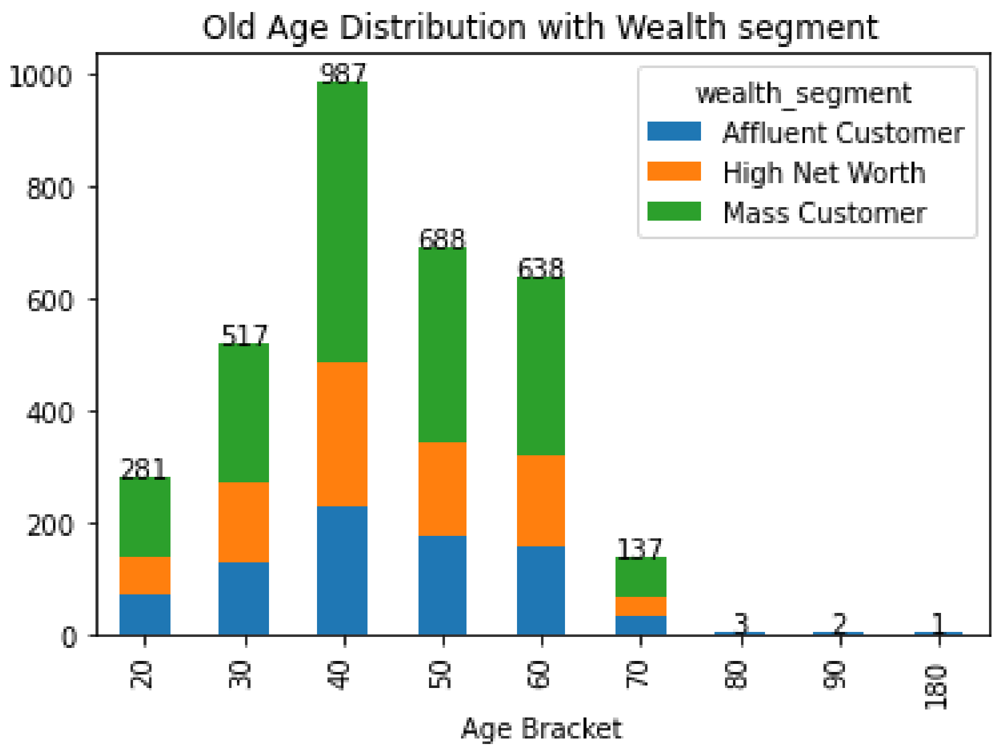
```
[ ]: df1.groupby('age_bracket')['state'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(tt['age_bracket'], tt['total'])
     plt.title('Old Age Distribution with State')
     plt.xlabel('Age Bracket')
```

```
[ ]: Text(0.5, 0, 'Age Bracket')
```



```
[ ]: wealth = df1.groupby('age_bracket')['wealth_segment'].value_counts().rename(
         'count').reset_index()
     ww = wealth.groupby('age_bracket')['count'].sum().rename('total').reset_index()
     df1.groupby('age_bracket')['wealth_segment'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(ww['age_bracket'], ww['total'])
     plt.title('Old Age Distribution with Wealth segment')
     plt.xlabel('Age Bracket')
```

```
[ ]: Text(0.5, 0, 'Age Bracket')
```

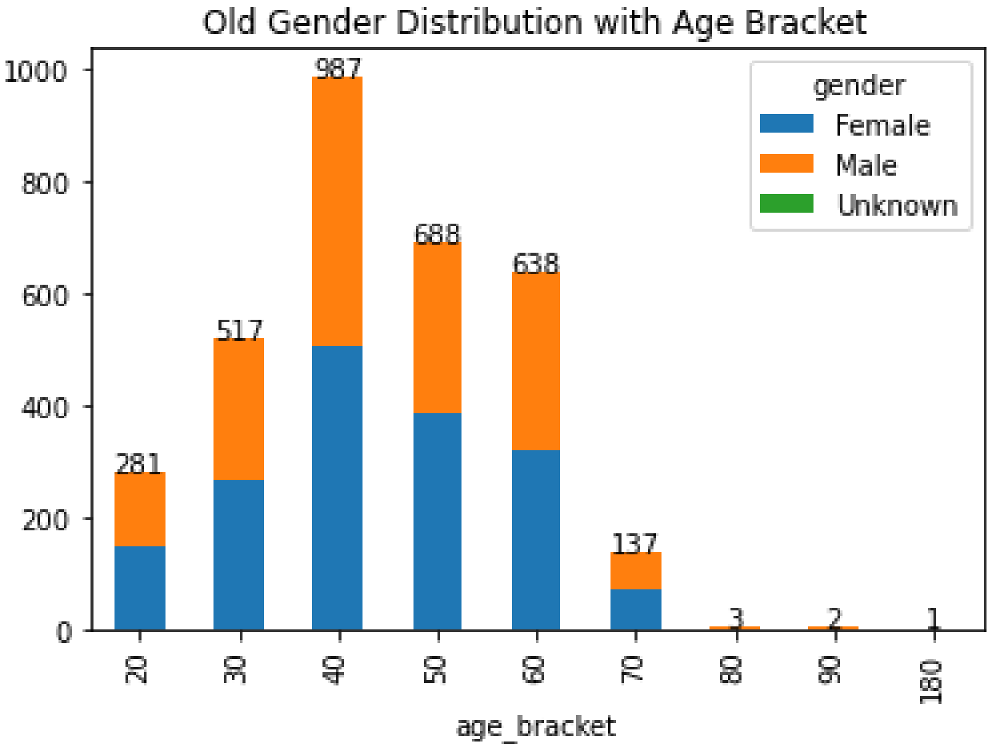


[ ]:

```
gender = df1.groupby('age_bracket')['gender'].value_counts().rename(
    'count').reset_index()
gg = gender.groupby('age_bracket')['count'].sum().rename('total').reset_index()
df1.groupby('age_bracket')['gender'].value_counts().unstack(
    level=1).plot(kind='bar', stacked=True)
addlabels(gg['age_bracket'], gg['total'])
plt.title('Old Gender Distribution with Age Bracket')
```

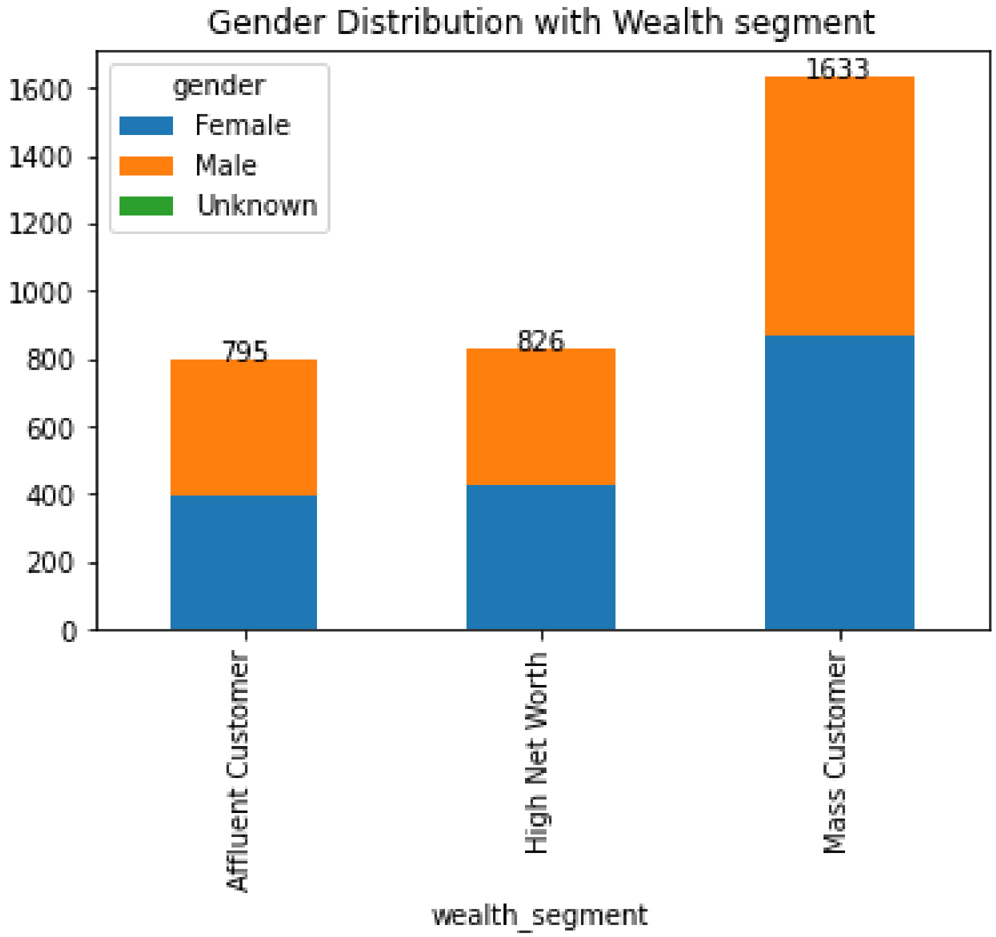
[ ]:

```
Text(0.5, 1.0, 'Old Gender Distribution with Age Bracket')
```



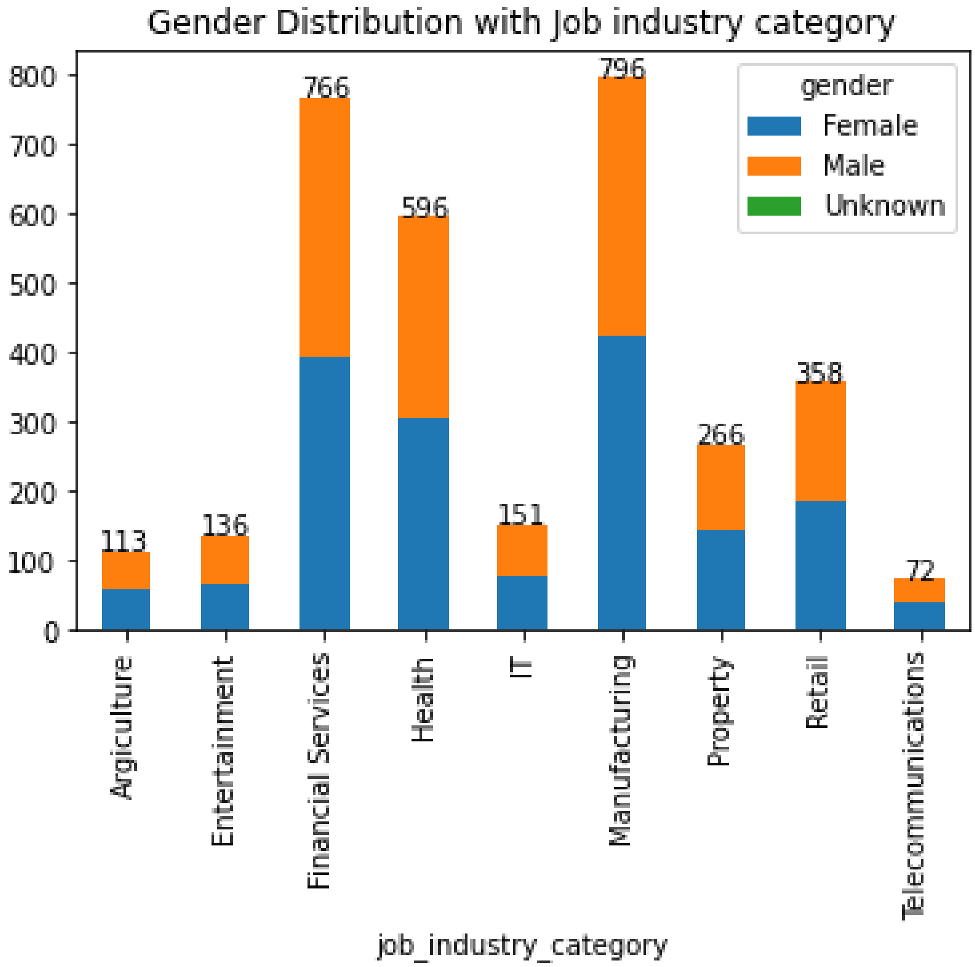
```
[ ]: wealth_gender = df1.groupby('wealth_segment')['gender'].value_counts().rename(
         'count').reset_index()
     wg = wealth_gender.groupby('wealth_segment')[
         'count'].sum().rename('total').reset_index()
     df1.groupby('wealth_segment')['gender'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(wg['wealth_segment'], wg['total'])
     plt.title('Gender Distribution with Wealth segment')
```

```
[ ]: Text(0.5, 1.0, 'Gender Distribution with Wealth segment')
```



```
[ ]: job_cat = df1.groupby('job_industry_category')['gender'].value_counts().rename(
         'count').reset_index()
     jc = job_cat.groupby('job_industry_category')[
         'count'].sum().rename('total').reset_index()
     df1.groupby('job_industry_category')['gender'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(jc['job_industry_category'], jc['total'])
     plt.title('Gender Distribution with Job industry category')
```

```
[ ]: Text(0.5, 1.0, 'Gender Distribution with Job industry category')
```

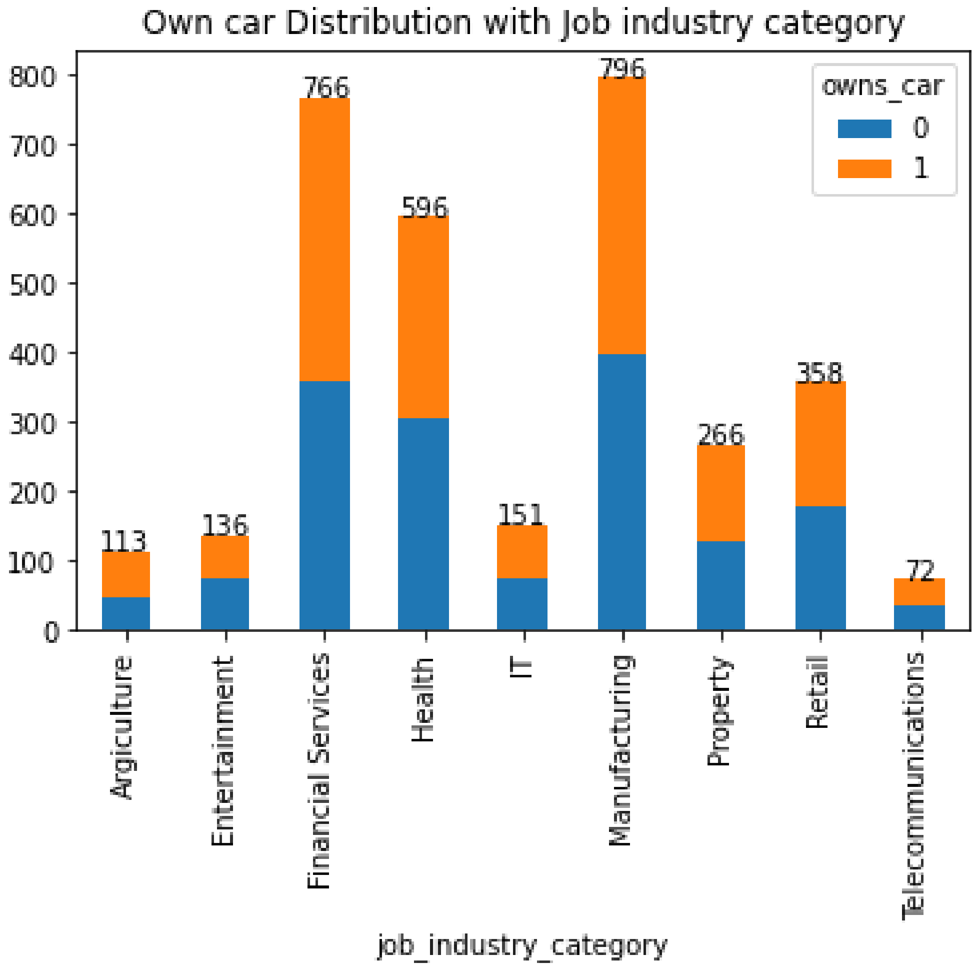


[ ]:

```
job_car = df1.groupby('job_industry_category')['owns_car'].value_counts().
 ↪rename(
    'count').reset_index()
jca = job_car.groupby('job_industry_category')[
    'count'].sum().rename('total').reset_index()
df1.groupby('job_industry_category')['owns_car'].value_counts().unstack(
    level=1).plot(kind='bar', stacked=True)
addlabels(jca['job_industry_category'], jca['total'])
plt.title('Own car Distribution with Job industry category')
```

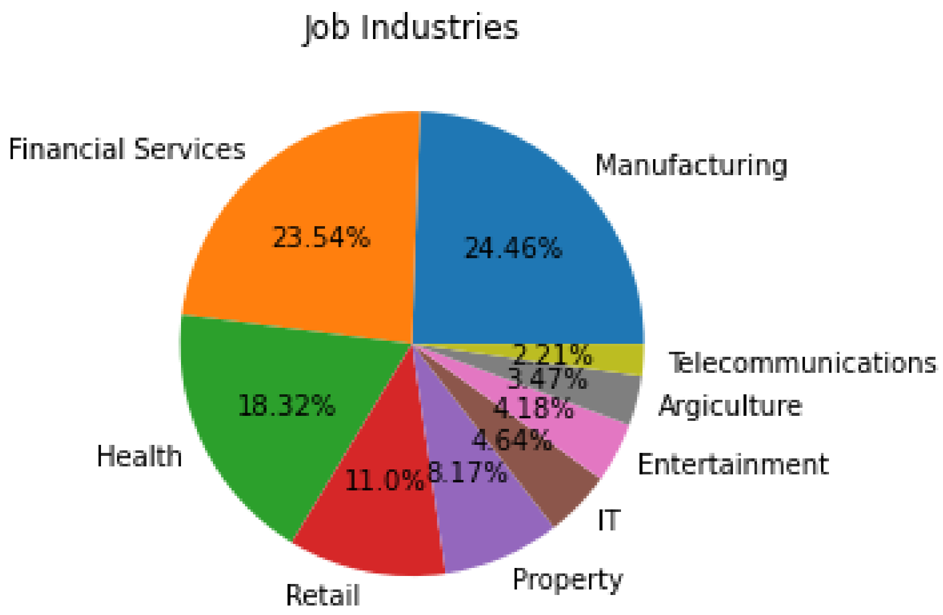
[ ]:

```
Text(0.5, 1.0, 'Own car Distribution with Job industry category')
```



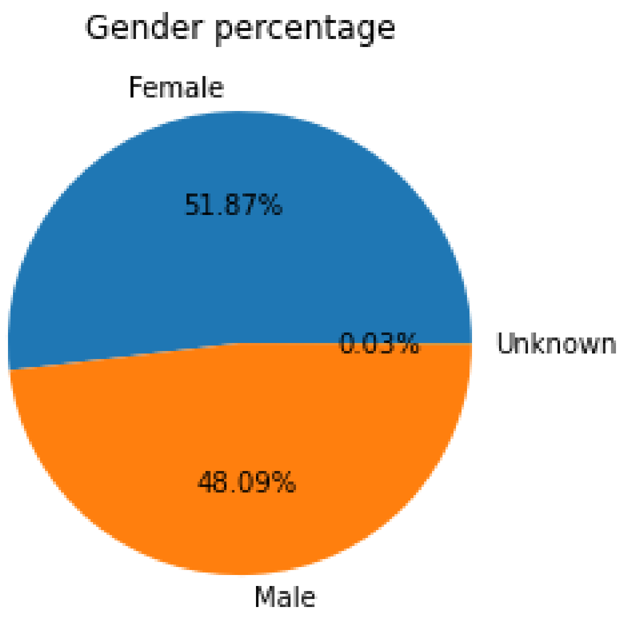
```
[ ]: df1['job_industry_category'].value_counts().plot(
         kind='pie', autopct=lambda pct: str(round(pct, 2)) + '%')
     plt.title('Job Industries')
     plt.ylabel('')
```

```
[ ]: Text(0, 0.5, '')
```



```
[ ]: df1['gender'].value_counts().plot(kind='pie', autopct=lambda pct:␣
      ↪str(round(pct, 2)) + '%')
     plt.title('Gender percentage')
     plt.ylabel('')
```

```
[ ]: Text(0, 0.5, '')
```



[ ]:
```
df2 = pd.read_excel('rawdata.xlsx', 'NewCustomerList')
df2.head()
```

```
/var/folders/25/53b25p9j7k52dz70pl14gl2w0000gn/T/ipykernel_8470/1238392822.py:1:
FutureWarning: Inferring datetime64[ns] from data containing strings is
deprecated and will be removed in a future version. To retain the old behavior
explicitly pass Series(data, dtype={value.dtype})
  df2 = pd.read_excel('rawdata.xlsx', 'NewCustomerList')
```

[ ]:
```
  first_name  last_name  gender  past_3_years_bike_related_purchases  \
0    Chickie    Brister    Male                                   86
1      Morly     Genery    Male                                   69
2    Ardelis  Forrester  Female                                   10
3     Lucine      Stutt  Female                                   64
4    Melinda     Hadlee  Female                                   34

         DOB                   job_title job_industry_category  \
0 1957-07-12             General Manager         Manufacturing
1 1970-03-22         Structural Engineer              Property
2 1974-08-28      Senior Cost Accountant    Financial Services
3 1979-01-28  Account Representative III         Manufacturing
4 1965-09-21           Financial Analyst    Financial Services

      wealth_segment deceased_indicator owns_car  …  state    country  \
0      Mass Customer                  N      Yes  …    QLD  Australia
1      Mass Customer                  N       No  …    NSW  Australia
2  Affluent Customer                  N       No  …    VIC  Australia
3  Affluent Customer                  N      Yes  …    QLD  Australia
4  Affluent Customer                  N       No  …    NSW  Australia

   property_valuation Unnamed: 16 Unnamed: 17  Unnamed: 18  Unnamed: 19  \
0                   6        1.01      1.2625     1.578125     1.341406
1                  11        0.70      0.7000     0.875000     0.743750
2                   5        0.67      0.6700     0.670000     0.670000
3                   1        0.96      1.2000     1.200000     1.200000
4                   9        0.73      0.7300     0.912500     0.912500

   Unnamed: 20  Rank     Value
0            1     1  1.718750
1            1     1  1.718750
2            1     1  1.718750
3            4     4  1.703125
4            4     4  1.703125

[5 rows x 23 columns]
```

[ ]:

```
df2.columns
```

[ ]:

```
Index(['first_name', 'last_name', 'gender',
       'past_3_years_bike_related_purchases', 'DOB', 'job_title',
       'job_industry_category', 'wealth_segment', 'deceased_indicator',
       'owns_car', 'tenure', 'address', 'postcode', 'state', 'country',
       'property_valuation', 'Unnamed: 16', 'Unnamed: 17', 'Unnamed: 18',
       'Unnamed: 19', 'Unnamed: 20', 'Rank', 'Value'],
      dtype='object')
```

[ ]:

```
df2.drop(['first_name', 'last_name', 'job_title', 'property_valuation',␣
 ↪'Unnamed: 16', 'Unnamed: 17', 'Unnamed: 18',
           'Unnamed: 19', 'Unnamed: 20'], axis =1, inplace=True)
df2.rename(columns={'past_3_years_bike_related_purchases': 'p3bkrel_pur'},␣
 ↪inplace=True)
df2['owns_car'] = df2['owns_car'].replace({'Yes': 1, 'No': 0})
df2['deceased_indicator'] = df2['deceased_indicator'].replace({'Y': 1, 'N': 0})
df2.dropna(inplace=True)
df2.head()
```

[ ]:

```
   gender  p3bkrel_pur        DOB job_industry_category     wealth_segment  \
0    Male           86 1957-07-12         Manufacturing      Mass Customer
1    Male           69 1970-03-22              Property      Mass Customer
2  Female           10 1974-08-28    Financial Services  Affluent Customer
3  Female           64 1979-01-28         Manufacturing  Affluent Customer
4  Female           34 1965-09-21    Financial Services  Affluent Customer

   deceased_indicator  owns_car  tenure               address  postcode state  \
0                   0         1      14      45 Shopko Center      4500   QLD
1                   0         0      16     14 Mccormick Park      2113   NSW
2                   0         0      10  5 Colorado Crossing      3505   VIC
3                   0         1       5    207 Annamark Plaza      4814   QLD
4                   0         0      19     115 Montana Place      2093   NSW

     country  Rank     Value
0  Australia     1  1.718750
1  Australia     1  1.718750
2  Australia     1  1.718750
3  Australia     4  1.703125
4  Australia     4  1.703125
```

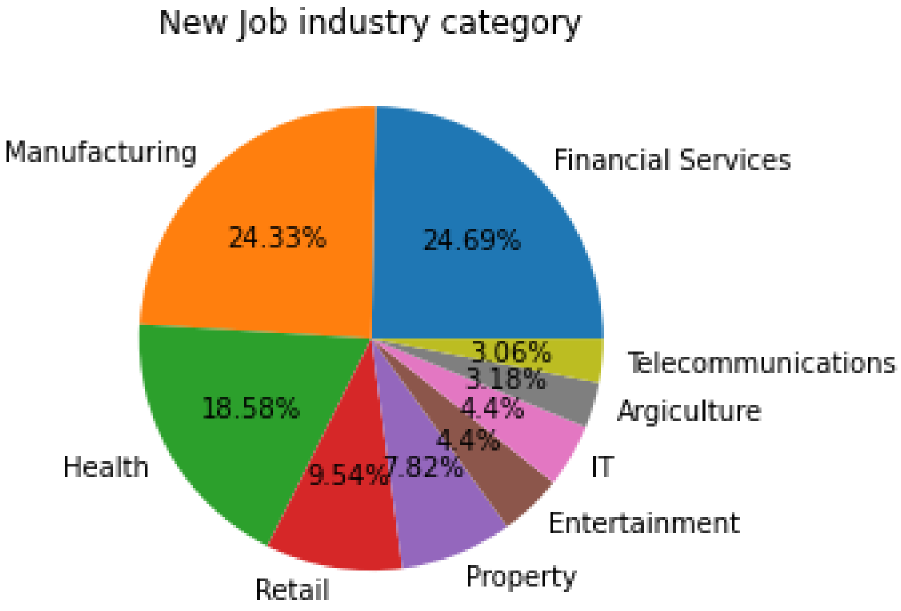
[ ]:

```
df2['age'] = ((
    dt.datetime.now() - df2['DOB']) / np.timedelta64(1, 'Y')).round(0).
 ↪astype(int)
df2['age_bracket'] = (
    (round(df2['age'] / 10)) * 10).astype(int)
df2.head()
```

```
[ ]:   gender  p3bkrel_pur        DOB job_industry_category     wealth_segment  \
     0   Male           86 1957-07-12         Manufacturing      Mass Customer
     1   Male           69 1970-03-22              Property      Mass Customer
     2 Female           10 1974-08-28    Financial Services  Affluent Customer
     3 Female           64 1979-01-28         Manufacturing  Affluent Customer
     4 Female           34 1965-09-21    Financial Services  Affluent Customer

       deceased_indicator  owns_car  tenure               address  postcode state  \
     0                  0         1      14    45 Shopko Center      4500   QLD
     1                  0         0      16    14 Mccormick Park      2113   NSW
     2                  0         0      10  5 Colorado Crossing      3505   VIC
     3                  0         1       5   207 Annamark Plaza      4814   QLD
     4                  0         0      19    115 Montana Place      2093   NSW

         country  Rank     Value  age  age_bracket
     0 Australia     1  1.718750   65           60
     1 Australia     1  1.718750   52           50
     2 Australia     1  1.718750   47           50
     3 Australia     4  1.703125   43           40
     4 Australia     4  1.703125   56           60
```

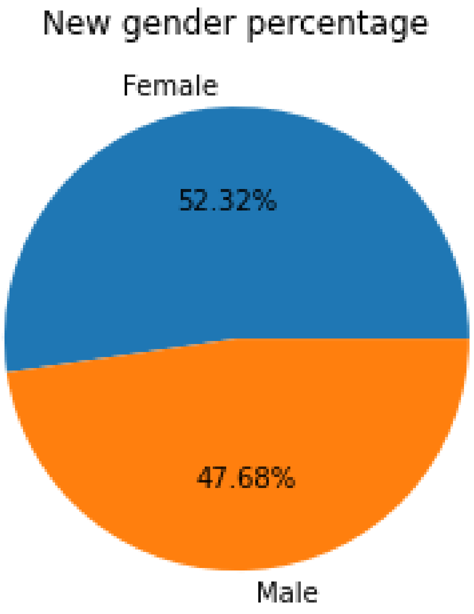
```
[ ]: df2['job_industry_category'].value_counts().plot(kind = 'pie', autopct = lambda␣
      ↪pct: str(round(pct, 2)) + '%')
     plt.ylabel(' ')
     plt.title('New Job industry category')
```

```
[ ]: Text(0.5, 1.0, 'New Job industry category')
```



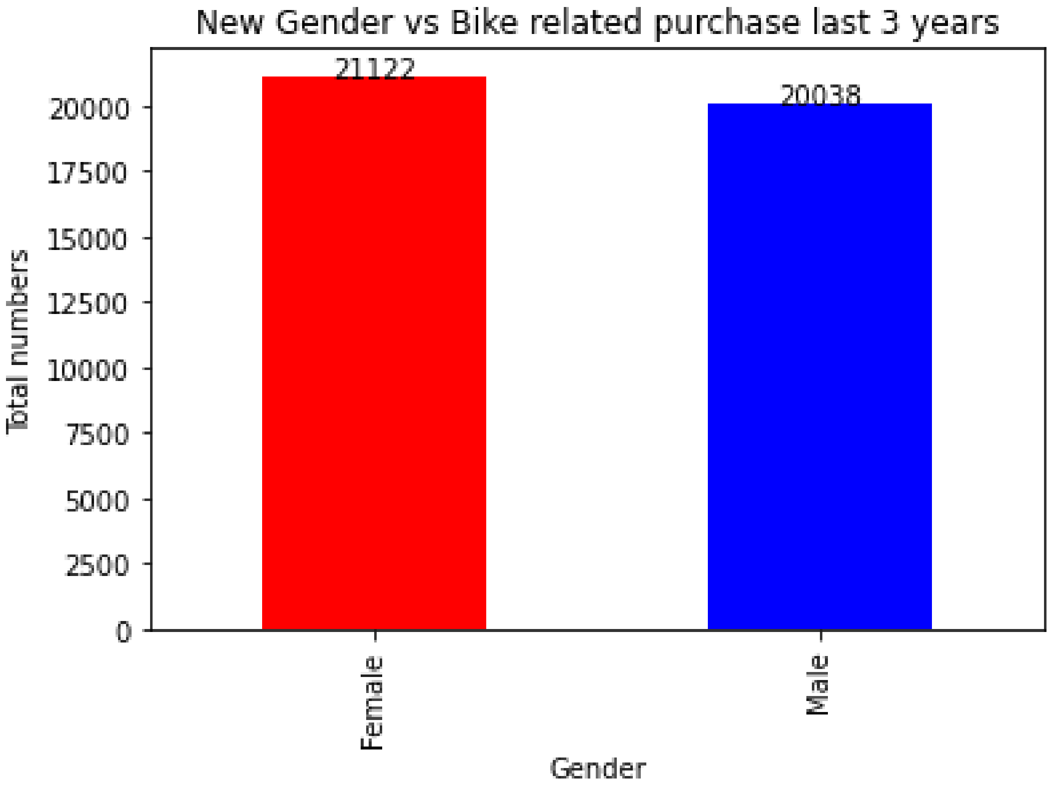
[ ]:
```
df2['gender'].value_counts().plot(kind = 'pie', autopct = lambda pct:␣
 ↪str(round(pct, 2))+ '%')
plt.ylabel(' ')
plt.title('New gender percentage')
```

[ ]:
```
Text(0.5, 1.0, 'New gender percentage')
```



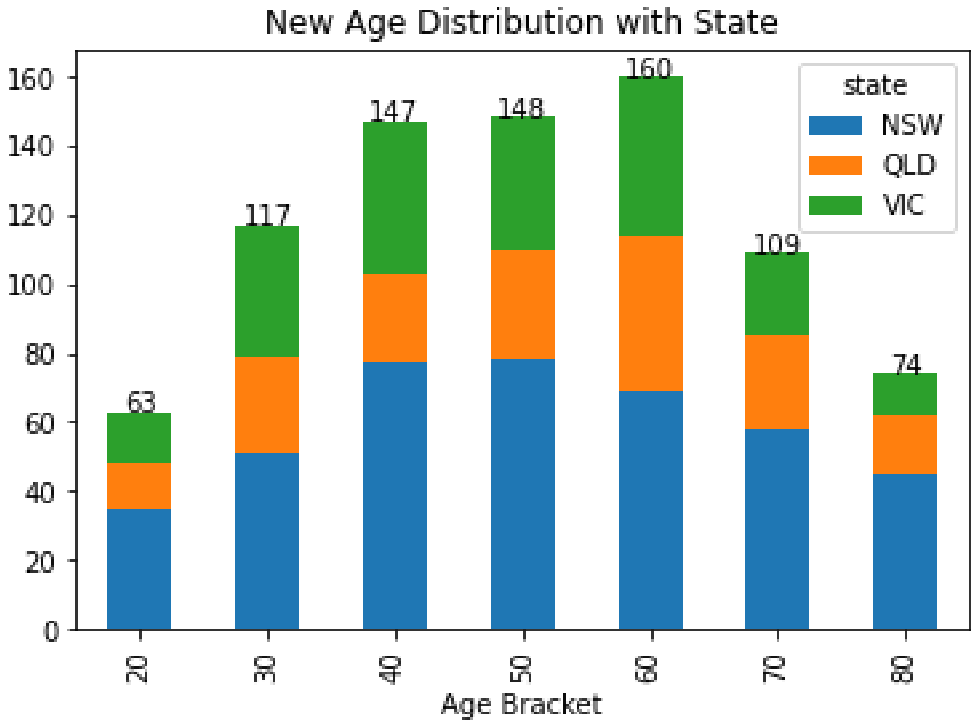
[ ]:
```
nwpur = df2.groupby('gender')['p3bkrel_pur'].sum().rename('count').reset_index()
df2.groupby('gender')['p3bkrel_pur'].sum().plot(
    kind='bar', stacked=True, color=['red', 'blue'])
addlabels(nwpur['gender'], nwpur['count'])
plt.title('New Gender vs Bike related purchase last 3 years')
plt.xlabel('Gender')
plt.ylabel('Total numbers')
```

[ ]:
```
Text(0, 0.5, 'Total numbers')
```



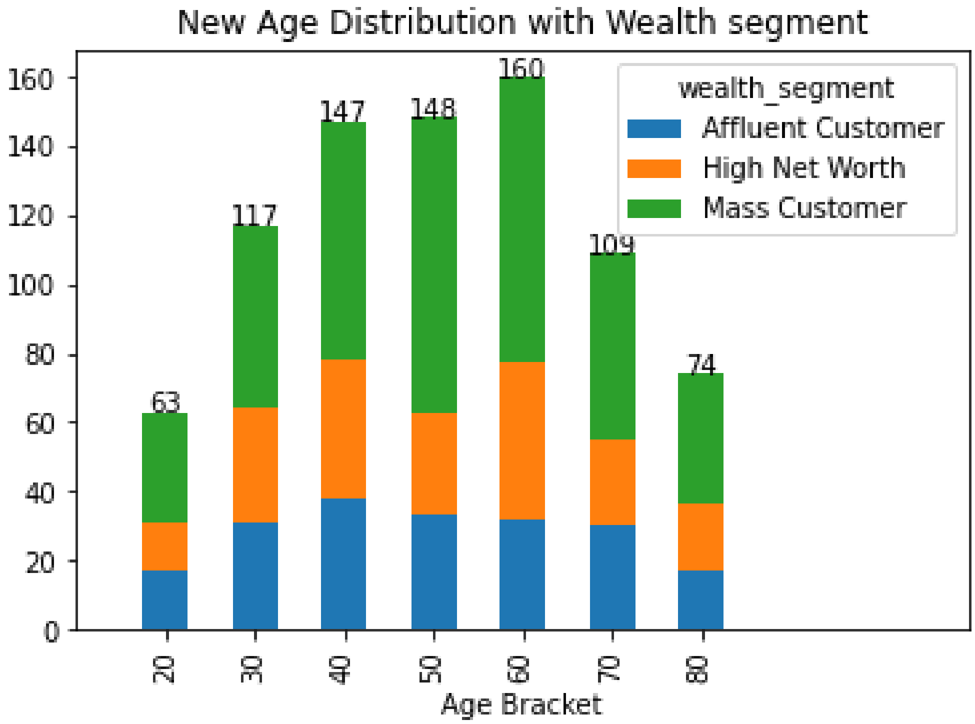
```
[ ]: nz = df2.groupby('age_bracket')['state'].value_counts().rename(
         'count').reset_index()
     ntt = nz.groupby('age_bracket')['count'].sum().rename('total').reset_index()
     df2.groupby('age_bracket')['state'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(ntt['age_bracket'], ntt['total'])
     plt.title('New Age Distribution with State')
     plt.xlabel('Age Bracket')
```

```
[ ]: Text(0.5, 0, 'Age Bracket')
```



```
[ ]: nwealth = df2.groupby('age_bracket')['wealth_segment'].value_counts().rename(
         'count').reset_index()
     nww = nwealth.groupby('age_bracket')['count'].sum().rename('total').
      ↪reset_index()
     df2.groupby('age_bracket')['wealth_segment'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(nww['age_bracket'], nww['total'])
     plt.title('New Age Distribution with Wealth segment')
     plt.xlim(-1, 9)
     plt.xlabel('Age Bracket')
```

```
[ ]: Text(0.5, 0, 'Age Bracket')
```

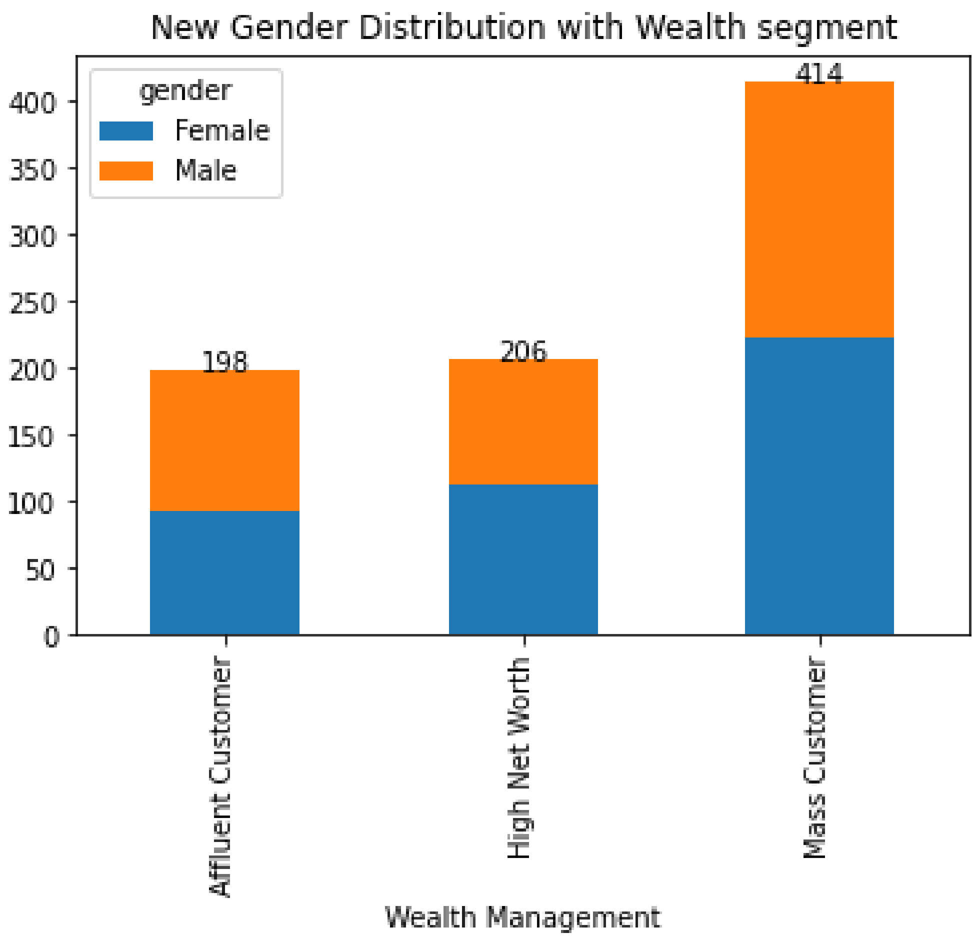


[ ]:

```
nwealth_gender = df2.groupby('wealth_segment')['gender'].value_counts().rename(
    'count').reset_index()
nwg = nwealth_gender.groupby('wealth_segment')[
    'count'].sum().rename('total').reset_index()
df2.groupby('wealth_segment')['gender'].value_counts().unstack(
    level=1).plot(kind='bar', stacked=True)
addlabels(nwg['wealth_segment'], nwg['total'])
plt.title('New Gender Distribution with Wealth segment')
plt.xlabel('Wealth Management')
```

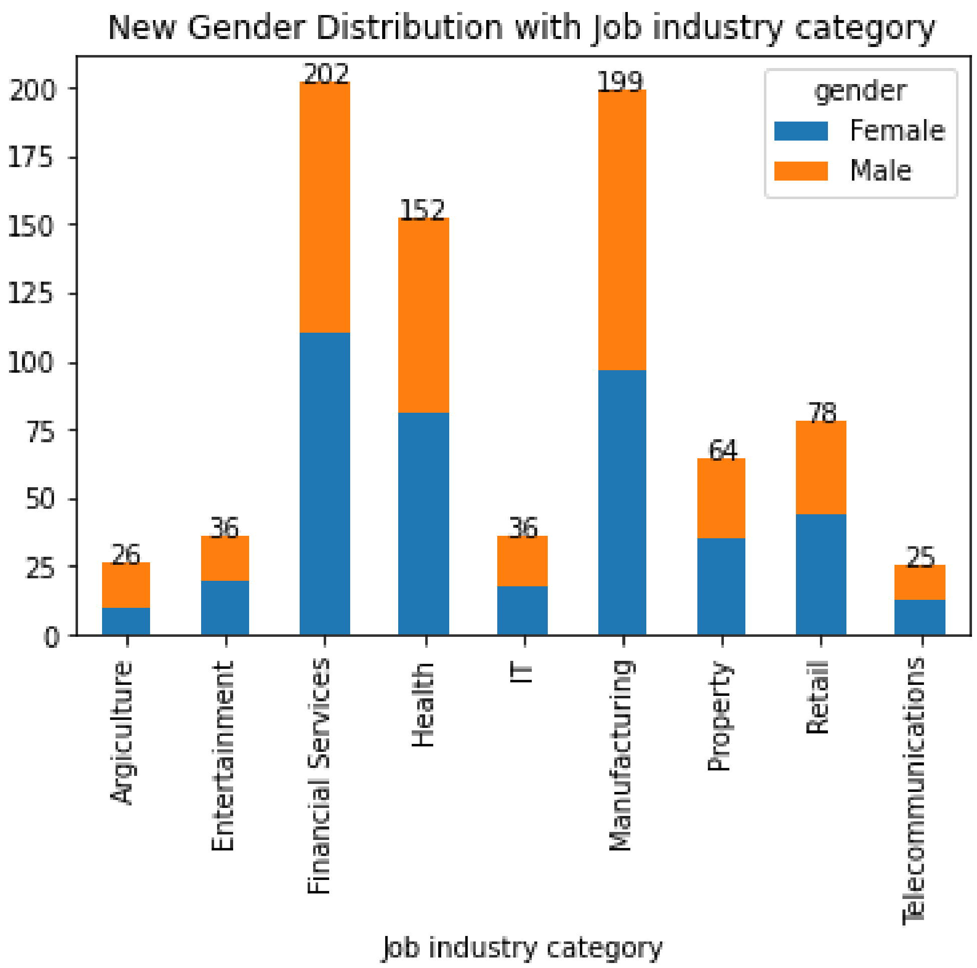
[ ]:

```
Text(0.5, 0, 'Wealth Management')
```



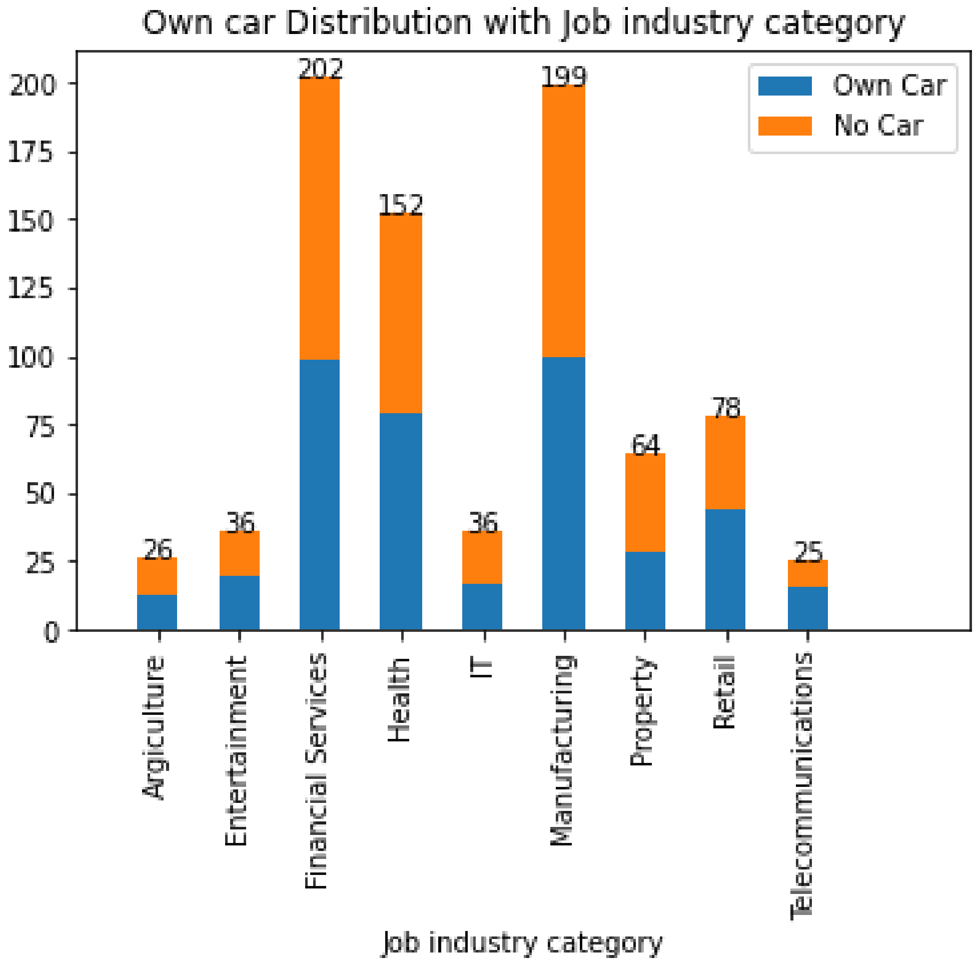
```
[ ]: njob_cat = df2.groupby('job_industry_category')['gender'].value_counts().rename(
         'count').reset_index()
     njc = njob_cat.groupby('job_industry_category')[
         'count'].sum().rename('total').reset_index()
     df2.groupby('job_industry_category')['gender'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(njc['job_industry_category'], njc['total'])
     plt.title('New Gender Distribution with Job industry category')
     plt.xlabel('Job industry category')
```

```
[ ]: Text(0.5, 0, 'Job industry category')
```



```
[ ]: njob_car = df2.groupby('job_industry_category')['owns_car'].value_counts().
      ↪rename(
         'count').reset_index()
     njca = njob_car.groupby('job_industry_category')[
         'count'].sum().rename('total').reset_index()
     df2.groupby('job_industry_category')['owns_car'].value_counts().unstack(
         level=1).plot(kind='bar', stacked=True)
     addlabels(njca['job_industry_category'], njca['total'])
     plt.title('Own car Distribution with Job industry category')
     plt.xlabel('Job industry category')
     plt.legend(['Own Car', 'No Car'])
     plt.xlim([-1, 10])
```

```
[ ]: (-1.0, 10.0)
```


Own car Distribution with Job industry category
Own Car
No Car
200
175
150
125
100
75
50
25
0
202
199
152
78
64
36
36
26
25
Argicultrure
Entertainment
Financial Services
Health
IT
Manufacturing
Property
Retail
Telecommunications
Job industry category