

In [1]:

```
# 导入所需要的函数模块
from marvel.AntPrint import ant_read_data, ant_write_data # 读取数据模块
from marvel.AntPrint import ant_print_all
from marvel.AntPrint import result_printer
from marvel.AntPrint.model import PSM
# 导入建模所需要的包

from scipy import stats
import seaborn as sns
import numpy as np
import pandas as pd
from scipy.stats import norm
import statsmodels.api as sm
from scipy.stats.mstats import winsorize
from linearmodels.iv.absorbing import AbsorbingLS, Interaction
from linearmodels import PanelOLS
from marvel.AntPrint import *
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
from statsmodels.formula.api import poisson
import patsy
from itertools import product
from sklearn.metrics import pairwise_distances
from statsmodels.stats.sandwich_covariance import cov_cluster
import itertools
```

Last executed in 46 s, finished in 20:39:36 2025-06-30

In [2]:

```
df = ant_read_data('workdata', mid_table = True)
```

Last executed in 15 s, finished in 20:39:51 2025-06-30

In [3]:

```
df['报告期1'] = df['报告期'].astype('str')
df['报告期1'] = pd.to_datetime(df['报告期1'])
df['year_month'] = df['报告期1'].apply(lambda x: x.strftime("%Y-%m"))

# 固定效应
df['type'] = df['period_length'].astype('category')
df['year_month_cat'] = df['year_month'].astype('category')
df['citycode_cat'] = df['citycode'].astype('category')
df['time_city'] = df.apply(lambda row: '{}_{}'.format(row['year_month_cat'],
row['citycode_cat']), axis=1).astype('category')
df['time_type'] = df.apply(lambda row: '{}_{}'.format(row['year_month_cat'],
row['type']), axis=1).astype('category')
df['用户id_rank'] = df['匿名化用户id_rank'].astype('category')

# 聚类层级
df['cluster_id'] = df['匿名化用户id_rank'].astype('category')
```

Last executed in 7 s, finished in 20:39:58 2025-06-30

In [4]:

```
regdata = df
```

Last executed in 0 s, finished in 20:39:58 2025-06-30

正文

表1 描述性统计

In [5]:

```
ant_print_all(regdata[['new_use', 'period_length', '性别', '是否有医保', '是否有社保',
'年龄', 'n_module_consumpt', 'all_consumpt', 'online_consumpt',
'offline_consumpt', 'num_consumpt', '是否基金交易', '余额宝余额', '基金余额']],
model_method="describe")
```

describe

	new_use	period_length	性别	是否有医保	是否有社保 \
count	60060.000000	60060.000000	60060.000000	60060.000000	60060.000000
mean	0.739011	6.640293	0.400733	0.433333	0.332967
std	0.439178	5.084558	0.490051	0.495540	0.471279
min	0.000000	3.000000	0.000000	0.000000	0.000000
25%	0.000000	3.000000	0.000000	0.000000	0.000000
50%	1.000000	4.000000	0.000000	0.000000	0.000000
75%	1.000000	8.000000	1.000000	1.000000	1.000000
max	1.000000	21.000000	1.000000	1.000000	1.000000

	年龄	n_module_consumpt	all_consumpt	online_consumpt \
count	60060.000000	60060.000000	60060.000000	60060.000000
mean	37.105128	5.145022	3929.583921	868.356031
std	12.558085	3.652004	8664.305087	1852.267457
min	18.000000	0.000000	0.000000	0.000000
25%	28.000000	2.000000	176.922500	30.000000
50%	34.000000	5.000000	1010.210000	250.950000
75%	45.000000	8.000000	3338.412500	809.402500
max	87.000000	21.000000	58067.040000	12774.180000

	offline_consumpt	num_consumpt	是否基金交易	余额宝余额 \
count	60060.000000	60060.000000	60038.000000	60060.000000
mean	2911.814177	30.225108	0.234518	5013.346014
std	7609.723784	43.815044	0.423701	16153.969313
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	4.000000	0.000000	0.000000
50%	327.000000	16.000000	0.000000	16.010000
75%	1985.160000	41.000000	0.000000	1073.037500
max	52202.970000	2121.000000	1.000000	108925.310000

基金余额

count	60060.000000
mean	960.688443
std	5751.150588
min	0.000000
25%	0.000000
50%	0.000000
75%	0.100000
max	48361.160000

表2 基准回归

In [6]:

```
lnty_list = ['lnn_module_consumpt', 'lnall_consumpt', 'lnnum_consumpt']

cats = regdata[['用户id_rank', 'time_city', 'time_type']] # 固定效应
cluster_id = regdata['cluster_id'] # 聚类层级
x = regdata["new_use"] # 解释变量

models = []
results = []
for i in lnty_list:
    y = regdata[i] # 被解释变量
    model = AbsorbingLS(y, x, absorb=cats)
    result = model.fit(method="lsmr", cov_type="clustered", clusters=
cluster_id)
    models.append(model)
    results.append(result)

# 结果输出
ant_print_all(result_printer, model_method="summary_col",
               results=results, model_names =lnty_list,
               regressor_order = ["new_use"], drop_omitted=True,
               info_list=["rsquared"], info_dict={"count": lambda x: x.nobs},
info_order=["rsquared", "count"], float_format='%.3f')
```

	lnn_module_consumpt	lnall_consumpt	lnnum_consumpt
new_use	0.048*** (0.011)	0.144*** (0.044)	0.073*** (0.021)
rsquared	0.753	0.711	0.769
count	60060.000	60060.000	60060.000

Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01

表3 消费便利性的提升

In [7]:

```
categories = ['商品购买', '生活服务', '交通出行', '充值缴费', '文化休闲', '餐饮美食', '余额宝余额', '基金余额']
dummy_list = [f'{board}_dum' for board in categories]

cats = regdata[['用户id_rank', 'time_city', 'time_type']]
cluster_id = regdata['cluster_id']
x = regdata["new_use"]

if enviro()=='仿真环境':
    pass
else:
    models = []
    results = []
    for i in dummy_list:
        y = regdata[i]
        model = AbsorbingLS(y, x, absorb=cats)
        result = model.fit(method="lsmr", cov_type="clustered", clusters=
cluster_id)
        models.append(model)
        results.append(result)

# 结果输出
ant_print_all(result_printer, model_method="summary_col",
               results=results, model_names=dummy_list,
               regressor_order = ["new_use"], drop_omitted=True,
               info_list=["rsquared"], info_dict={"count": lambda x: x.
nobs}, info_order=["rsquared", "count"], float_format='%.3f')
```

```
=====
                商品购买_dum  生活服务_dum  交通出行_dum  充值缴费_dum  文化休闲_dum  餐饮
美食_dum  余额宝余额_dum  基金余额_dum
-----
```

```
new_use  0.015**    0.016**    0.013*     0.017**    0.021**    0.018**    0.001
0.011*
          (0.007)    (0.007)    (0.007)    (0.008)    (0.008)    (0.007)    (0.007)
(0.006)
rsquared 0.570      0.409      0.522      0.560      0.444      0.570      0.696
0.863
count    60060.000  60060.000  60060.000  60060.000  60060.000  60060.000  60060.000
60060.000 60060.000
=====
```

Standard errors in parentheses.

* p<.1, ** p<.05, ***p<.01

Last executed in 9 s, finished in 20:40:12 2025-06-30

表4 控制疫情封控的影响 ¶

In [8]:

```
# p_case_人口平减 p_case_人口平减为lock_down指标
regdata['newuse_p_case'] = regdata['new_use'] * regdata['p_case_人口平减']

covid_var = ["p_case"]

regdata['median_coverage_breadth'] = regdata['coverage_breadth'].median() #
coverage_breadth 支付宝覆盖指数

reg = regdata.dropna(subset=['p_case_人口平减'])
```

Last executed in 0 s, finished in 20:40:12 2025-06-30

表4 列1

In [9]:

```
cats = reg[['用户id_rank','time_city','time_type']]
cluster_id = reg['cluster_id']

if envir()=='仿真环境':
    pass
else:
    for j in covid_var:
        interact = f'newuse_{j}'
        x = reg[["new_use", interact]]

        y_list = ['lnoffline_consumpt']
        models = []
        results = []
        for i in y_list:
            y = reg[i]
            model = AbsorbingLS(y, x, absorb=cats)
            result = model.fit(method="lsmr", cov_type="clustered", clusters
=cluster_id)
            models.append(model)
            results.append(result)

        # 结果输出
        ant_print_all(result_printer, model_method="summary_col",
                      results=results, model_names = y_list,
                      regressor_order = ["new_use",interact,j],
drop_omitted=True,
                      info_list=["rsquared"], info_dict={"count": lambda x:
x.nobs}, info_order=["rsquared", "count"],float_format='% .3f')
```

```
=====
lnoffline_consumpt
-----
```

```
new_use      0.181***
              (0.061)
newuse_p_case -0.798*
              (0.434)
rsquared      0.687
count         58762.000
=====
```

Standard errors in parentheses.

* p<.1, ** p<.05, ***p<.01

Last executed in 2 s, finished in 20:40:14 2025-06-30

表4 列2

In [10]:

```
reg['coverage_breadth_group'] = (reg['coverage_breadth'] > reg[
'median_coverage_breadth']).astype(int)

reg_high = reg[reg['coverage_breadth_group'] == 1]
cats = reg_high[['用户id_rank', 'time_city', 'time_type']]
cluster_id = reg_high['cluster_id']

if envir()=='仿真环境':
    pass
else:
    for j in covid_var:
        interact = f'newuse_{j}'
        x = reg_high[["new_use", interact]]

        y_list = ['lnonline_consumpt']
        models = []
        results = []
        for i in y_list:
            y = reg_high[i]
            model = AbsorbingLS(y, x, absorb=cats)
            result = model.fit(method="lsmr", cov_type="clustered", clusters
=cluster_id)
            models.append(model)
            results.append(result)

        # 结果输出
        ant_print_all(result_printer, model_method="summary_col",
                      results=results, model_names = y_list,
                      regressor_order = ["new_use", interact, j],
drop_omitted=True,
                      info_list=["rsquared"], info_dict={"count": lambda x:
x.nobs}, info_order=["rsquared", "count"], float_format='% .3f')
```



```

*****

=====
                    lnonline_consumpt
-----
new_use           0.125**
                  (0.064)
newuse_p_case    -0.625*
                  (0.345)
rsquared          0.685
count            29304.000
=====
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
*****

```

Last executed in 2 s, finished in 20:40:16 2025-06-30

表4 列3

In [11]:

```
reg_low = reg[reg['coverage_breadth_group'] == 0]
cats = reg_low[['用户id_rank', 'time_city', 'time_type']]
cluster_id = reg_low['cluster_id']

if envir()=='仿真环境':
    pass
else:
    for j in covid_var:
        interact = f'newuse_{j}'
        x = reg_low[["new_use", interact]]

        y_list = ['lnonline_consumpt']
        models = []
        results = []
        for i in y_list:
            y = reg_low[i]
            model = AbsorbingLS(y, x, absorb=cats)
            result = model.fit(method="lsmr", cov_type="clustered", clusters
=cluster_id)
            models.append(model)
            results.append(result)

        # 结果输出
        ant_print_all(result_printer, model_method="summary_col",
                      results=results, model_names = y_list,
                      regressor_order = ["new_use", interact, j],
drop_omitted=True,
                      info_list=["rsquared"], info_dict={"count": lambda x:
x.nobs}, info_order=["rsquared", "count"], float_format='%0.3f')
```

```

*****

=====
                    lnonline_consumpt
-----
new_use            0.158***
                   (0.061)
newuse_p_case     2.255
                   (1.515)
rsquared           0.722
count              29458.000
=====
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
*****

```

Last executed in 1 s, finished in 20:40:16 2025-06-30

表5 调节效应

In [12]:

```
adjust = regdata.dropna(subset=['残疾干部残联比例'])
adjustment_var = ['残疾干部残联比例', '村级残协', '残疾证', '建设面积', '县域政务网站无障碍比例']

cats = adjust[['用户id_rank', 'time_city', 'time_type']]
cluster_id = adjust['cluster_id']

if enviro()=='仿真环境':
    pass
else:
    for j in adjustment_var:
        x = adjust[["new_use", f"newuse_{j}"]]

        models = []
        results = []
        y = adjust['lnn_module_consumpt']
        model = AbsorbingLS(y, x, absorb=cats)
        result = model.fit(method="lsmr", cov_type="clustered", clusters=
cluster_id)
        models.append(model)
        results.append(result)

        # 结果输出
        ant_print_all(result_printer, model_method="summary_col",
                      results=results, model_names = ['lnn_module_consumpt'
],
                      regressor_order = ["new_use", f"newuse_{j}"],
drop_omitted=True,
                      info_list=["rsquared"], info_dict={"count": lambda x:
x.nobs}, info_order=["rsquared", "count"],float_format='%0.5f')
```

```

*****

=====
                        lnn_module_consumpt
-----
new_use                -0.00345
                        (0.01667)
newuse_残疾干部残联比例 0.40193***
                        (0.10779)
rsquared               0.75346
count                 60038.00000
=====
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
*****

```

```

*****

=====
                        lnn_module_consumpt
-----
new_use                -0.01541
                        (0.01811)
newuse_村级残协 0.02537***
                        (0.00616)
rsquared               0.75344
count                 60038.00000
=====
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
*****

```

```

*****

=====
                        lnn_module_consumpt
-----
new_use                -0.00050
                        (0.01763)
newuse_残疾证 0.01341***
                        (0.00406)
rsquared               0.75339
count                 60038.00000
=====
Standard errors in
parentheses.
* p<.1, ** p<.05, ***p<.01
*****

```

```

*****

```

```

=====
                        lnn_module_consumpt
-----
new_use      0.00058
              (0.01818)
newuse_建设面积 0.01521***
              (0.00493)
rsquared     0.75338
count        60038.00000
=====
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
*****

*****

=====
                        lnn_module_consumpt
-----
new_use      -0.01930
              (0.02065)
newuse_县域政务网站无障碍比例 0.07861***
              (0.02120)
rsquared     0.75339
count        60038.00000
=====
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
*****

```

Last executed in 6 s, finished in 20:40:23 2025-06-30