December 27, 2020

[4]:
```
import pandas as pd
import scipy.stats as ss
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pingouin as pg
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from statsmodels.formula.api import ols
import warnings


warnings.filterwarnings("ignore")


demo = pd.read_spss('demo2017 (1).SAV')
#
# ,
demo = demo.rename(columns={'N_UHC' : 'n_uhc', 'N_UPC' : 'n_upc', 'AGE' : 'age',␣
↪'SEX' : 'sex'})
demo.to_csv('demo.csv', sep='\t', index=False)

health = pd.read_spss('health2017 (1).sav')
health.to_csv('health.csv', sep='\t', index=False)


income = pd.read_spss('income2017 (1).SAV')
income.to_csv('income.csv', sep='\t', index=False)

#    ,
buf = demo.merge(health, how='outer', left_on=['n_uhc', 'n_upc'],␣
↪right_on=['n_uhc', 'n_upc'], suffixes=('', '_y'))
buf = buf.drop(buf.filter(regex='_y$').columns.tolist(), axis=1)
#
buf = buf.rename(columns={'YEAR' : 'year'})
buf.to_csv('buf.csv', sep='\t', index=False)
income = income.rename(columns={'N_UHC' : 'n_uhc', 'RESID' :'resid'})
```

```python
#   ,
df = buf.merge(income, how='outer', left_on=['n_uhc'], right_on=['n_uhc'], suffixes=('', '_y'))
df = df.drop(df.filter(regex='_y$').columns.tolist(), axis=1)
df.to_csv('df.csv', sep='\t', index=False)

#
dataset = df.sample(n=2000, random_state=20001227)
dataset.reset_index(drop=True, inplace=True)
#          ,
dataset = dataset.astype({'n_uhc' : 'Int64', 'n_upc' : 'Int64', 'year' : 'Int64', 'age' : 'Int64', 'nummonth' : 'Int64',\
    'weight' : 'Int64', 'height' : 'Int64', 'HSIZE' : 'Int64', 'ch0_5' : 'Int64', 'ch6_12' : 'Int64', 'ch13_17' : 'Int64',\
    'elder' : 'Int64' , 'HH_BLINT' : 'Int64', 'HH_INT1' : 'Int64', 'HH_INT2' : 'Int64', 'HH_INT3' : 'Int64', 'HH_INT4' : 'Int64'})
dataset.dtypes.to_csv('dtypes.txt', sep='\t', index=False)
dataset.to_csv('dataset.csv', sep='\t', index=False)

#
dataset['Yweight'] = np.floor(dataset['Yweight'])
dataset = dataset.astype({'Yweight' : 'int64'})


print(" ")
print()
print(" ")
print("mean", dataset['totalinc'].mean())
print("median", dataset['totalinc'].median())
print("mode", dataset['totalinc'].mode().values)
print("std", dataset['totalinc'].std())
print("skewness", dataset['totalinc'].skew())
print("kurtosis", dataset['totalinc'].kurtosis())
dataset['totalinc'].plot(kind="hist", title='Total income')
print()
plt.show()

print(" ")
print("mean", dataset['totalexp'].mean())
print("median", dataset['totalexp'].median())
print("mode", dataset['totalexp'].mode().values)
print("std", dataset['totalexp'].std())
print("skewness", dataset['totalexp'].skew())
print("kurtosis", dataset['totalexp'].kurtosis())
dataset['totalexp'].plot(kind="hist", title='Total expense')
print()
plt.show()
```

```
print("")
print("mean", dataset['inc_1'].mean())
print("median", dataset['inc_1'].median())
print("mode", dataset['inc_1'].mode().values)
print("std", dataset['inc_1'].std())
print("skewness", dataset['inc_1'].skew())
print("kurtosis", dataset['inc_1'].kurtosis())
dataset['inc_1'].plot(kind="hist", title='Wages')
print()
plt.show()


print("      ")
print("mean", dataset['inc_6'].mean())
print("median", dataset['inc_6'].median())
print("mode", dataset['inc_6'].mode().values)
print("std", dataset['inc_6'].std())
print("skewness", dataset['inc_6'].skew())
print("kurtosis", dataset['inc_6'].kurtosis())
dataset['inc_6'].plot(kind="hist", title='inc_6')
print()
plt.show()


print("  ")
print("mean", dataset['exp_6'].mean())
print("median", dataset['exp_6'].median())
print("mode", dataset['exp_6'].mode().values)
print("std", dataset['exp_6'].std())
print("skewness", dataset['exp_6'].skew())
print("kurtosis", dataset['exp_6'].kurtosis())
dataset['exp_6'].plot(kind="hist", title='exp_6')
print()
plt.show()


print("")
print("mean", dataset['weight'].mean())
print("median", dataset['weight'].median())
print("mode", dataset['weight'].mode().values)
print("std", dataset['weight'].std())
print("skewness", dataset['weight'].skew())
print("kurtosis", dataset['weight'].kurtosis())
dataset['weight'].plot(kind="hist", title='Weight')
print()
```

```
plt.show()


print("")
print("mean", dataset['height'].mean())
print("median", dataset['height'].median())
print("mode", dataset['height'].mode().values)
print("std", dataset['height'].std())
print("skewness", dataset['height'].skew())
print("kurtosis", dataset['height'].kurtosis())
dataset['height'].plot(kind="hist", title='Height')
print()
plt.show()


print("")
print("mean", dataset['age'].mean())
print("median", dataset['age'].median())
print("mode", dataset['age'].mode().values)
print("std", dataset['age'].std())
print("skewness", dataset['age'].skew())
print("kurtosis", dataset['age'].kurtosis())
dataset['height'].plot(kind="hist", title='Height')
print()
plt.show()

mass_index = dataset['weight'] / (dataset['height']**2)
saved_money = dataset['totalinc'] - dataset['totalexp']

print("  ")
print("mean", mass_index.mean())
print("median", mass_index.median())
print("mode", mass_index.mode().values)
print("std", mass_index.std())
print("skewness", mass_index.skew())
print("kurtosis", mass_index.kurtosis())
mass_index.plot(kind="hist", title='Mass index')
print()
plt.show()

print("")
print("mean", saved_money.mean())
print("median", saved_money.median())
print("mode", saved_money.mode().values)
print("std", saved_money.std())
print("skewness", saved_money.skew())
print("kurtosis", saved_money.kurtosis())
```

```
saved_money.plot(kind="hist", title='Saved money')
print()
plt.show()

#
sns.histplot(x=dataset['sex'])
plt.show()
fig, ax = plt.subplots(figsize=(30, 4))
sns.histplot(x=dataset['region'], ax=ax)
plt.show()
fig, ax = plt.subplots(figsize=(30, 4))
sns.histplot(x=dataset['HTYPE'].dropna(), ax = ax, hue=dataset['HTYPE'].dropna())
plt.show()
sns.histplot(x=dataset['HSIZE'].dropna())
plt.show()
fig, ax = plt.subplots(figsize=(30, 4))
sns.histplot(x=dataset['Educat'], ax = ax, hue=dataset['Educat'])
plt.show()
sns.histplot(x=dataset['healthev'].dropna())
plt.show()

#
#
#    ,      -
weighted_inc_1 = pd.Series([])
for i in range(len(dataset['Yweight'])):
    weighted_inc_1 = weighted_inc_1.append(pd.Series(np.
 ↪repeat(dataset['inc_1'][i], dataset['Yweight'][i])))


print(" ")
print("mean", weighted_inc_1.mean())
print("median", weighted_inc_1.median())
print("mode", weighted_inc_1.mode().values)
print("std", weighted_inc_1.std())
print("skewness", weighted_inc_1.skew())
print("kurtosis", weighted_inc_1.kurtosis())
weighted_inc_1.plot(kind="hist", title='Weighted wages')
print()
plt.show()

#
weighted_saved_money = pd.Series([])
for i in range(len(dataset['Yweight'])):
    weighted_saved_money = weighted_saved_money.append(pd.Series(np.
 ↪repeat(saved_money[i], dataset['Yweight'][i])))
```

```
print(" ")
print("mean", weighted_saved_money.mean())
print("median", weighted_saved_money.median())
print("mode", weighted_saved_money.mode().values)
print("std", weighted_saved_money.std())
print("skewness", weighted_saved_money.skew())
print("kurtosis", weighted_saved_money.kurtosis())
weighted_saved_money.plot(kind="hist")
print()
print()
plt.show()


#
#
log_totalinc = pd.Series(np.log(dataset['totalinc']))
log_totalexp = pd.Series(np.log(dataset['totalexp']))
log_weight = pd.Series(np.log(dataset['weight']))
log_height = pd.Series(np.log(dataset['height']))

print("  ")
print()
print(" ")
print("mean", log_totalinc.mean())
print("median", log_totalinc.median())
print("mode", log_totalinc.mode().values)
print("std", log_totalinc.std())
print("skewness", log_totalinc.skew())
print("kurtosis",log_totalinc.kurtosis())
log_totalinc.plot(kind="hist", title='Total income')
print()
plt.show()

print(" ")
print("mean", log_totalexp.mean())
print("median", log_totalexp.median())
print("mode", log_totalexp.mode().values)
print("std", log_totalexp.std())
print("skewness", log_totalexp.skew())
print("kurtosis", log_totalexp.kurtosis())
log_totalexp.plot(kind="hist", title='Total expense')
print()
plt.show()
```

```
print("")
print("mean", log_weight.mean())
print("median", log_weight.median())
print("mode", log_weight.mode().values)
print("std", log_weight.std())
print("skewness", log_weight.skew())
print("kurtosis", log_weight.kurtosis())
log_weight.plot(kind="hist", title='Weight')
print()
plt.show()


print("")
print("mean", log_height.mean())
print("median", log_height.median())
print("mode", log_height.mode().values)
print("std", log_height.std())
print("skewness", log_height.skew())
print("kurtosis", log_height.kurtosis())
log_height.plot(kind="hist", title='Height')
print()
plt.show()



#
sd_totalinc = (dataset['totalinc'] - dataset['totalinc'].mean())/
 ↪dataset['totalinc'].std()
sd_totalexp = (dataset['totalexp'] - dataset['totalexp'].mean())/
 ↪dataset['totalexp'].std()
sd_inc_1 = (dataset['inc_1'] - dataset['inc_1'].mean())/dataset['inc_1'].std()
sd_inc_6 = (dataset['inc_6'] - dataset['inc_6'].mean())/dataset['inc_6'].std()
sd_exp_6 = (dataset['exp_6'] - dataset['exp_6'].mean())/dataset['exp_6'].std()
sd_weight = (dataset['weight'] - dataset['weight'].mean())/dataset['weight'].
 ↪std()
sd_height = (dataset['height'] - dataset['height'].mean())/dataset['height'].
 ↪std()
sd_age = (dataset['age'] - dataset['age'].mean())/dataset['age'].std()

sd_totalinc.plot(kind="hist", title='Std Total income')
plt.show()
sd_totalexp.plot(kind="hist", title='Std Total expense')
plt.show()
sd_inc_1.plot(kind="hist", title='Std Wages')
plt.show()
sd_inc_6.plot(kind="hist", title='Std inc_6')
plt.show()
```

```
sd_exp_6.plot(kind="hist", title='Std exp_6')
plt.show()
sd_weight.plot(kind="hist", title='Std Weight')
plt.show()
sd_height.plot(kind="hist", title='Std Height')
plt.show()
sd_age.plot(kind="hist", title='Std Age')
plt.show()

#  ,         2017
#
print("   ")
print(ss.ttest_1samp(dataset['inc_1'], 815.25, nan_policy='omit'))
print(ss.ttest_1samp(weighted_inc_1, 815.25, nan_policy='omit'))
print()


males = dataset.loc[dataset['sex'] == 'Male']
females = dataset.loc[dataset['sex'] == 'Female']

#  ,
#
print("      ")
print(ss.ttest_ind(males['inc_1'], females['inc_1'], equal_var=True, nan_policy='omit'))
print(ss.ttest_ind(males['inc_1'], females['inc_1'], equal_var=False, nan_policy='omit'))
print()
# ,
print("     ")
print(ss.levene(males['inc_1'].dropna(), females['inc_1'].dropna()))
print()

grodno = dataset.loc[dataset['region'] == 'Grodno oblast']
mogilev = dataset.loc[dataset['region'] == 'Mogilev oblast']

# inc_6  ,   exp_6
print("       ")
print()
print("  inc_6")
print(ss.ttest_ind(grodno['inc_6'], mogilev['inc_6'], equal_var=True, nan_policy='omit'))
print(ss.ttest_ind(grodno['inc_6'], mogilev['inc_6'], equal_var=False, nan_policy='omit'))
print(ss.levene(grodno['inc_6'].dropna(), mogilev['inc_6'].dropna()))
print()
print("  exp_6")
```

```
print(ss.ttest_ind(grodno['exp_6'], mogilev['exp_6'], equal_var=True,␣
 ↪nan_policy='omit'))
print(ss.ttest_ind(grodno['exp_6'], mogilev['exp_6'], equal_var=False,␣
 ↪nan_policy='omit'))
print(ss.levene(grodno['exp_6'].dropna(), mogilev['exp_6'].dropna()))
print()

#
#
age = pd.Series([])
for i in range(len(dataset['age'])):
    if dataset['age'][i] >= 18 and dataset['age'][i] < 25:
        age = age.append(pd.Series(['18-24']), ignore_index=True)
    elif dataset['age'][i] >= 25 and dataset['age'][i] < 35:
        age = age.append(pd.Series(['25-34']), ignore_index=True)
    elif dataset['age'][i] >= 35 and dataset['age'][i] < 45:
        age = age.append(pd.Series(['35-44']), ignore_index=True)
    elif dataset['age'][i] >= 45 and dataset['age'][i] < 55:
        age = age.append(pd.Series(['45-54']), ignore_index=True)
    elif dataset['age'][i] >= 55 and dataset['age'][i] < 65:
        age = age.append(pd.Series(['55-64']), ignore_index=True)
    else:
        age = age.append(pd.Series(np.nan))

#
wages = pd.Series([])
for i in range(len(dataset['inc_1'])):
    if dataset['inc_1'][i] > 0 and dataset['inc_1'][i] < 400:
        wages = wages.append(pd.Series(['0-400']), ignore_index=True)
    elif dataset['inc_1'][i] >= 400 and dataset['inc_1'][i] < 500:
        wages = wages.append(pd.Series(['400-500']), ignore_index=True)
    elif dataset['inc_1'][i] >= 500 and dataset['inc_1'][i] < 700:
        wages = wages.append(pd.Series(['500-700']), ignore_index=True)
    elif dataset['inc_1'][i] >= 700 and dataset['inc_1'][i] < 1000:
        wages = wages.append(pd.Series(['700-1000']), ignore_index=True)
    elif dataset['inc_1'][i] >= 1000:
        wages = wages.append(pd.Series(['>1000']), ignore_index=True)
    elif abs(dataset['inc_1'][i]) < 0.00000001 or np.isnan(dataset['inc_1'][i]):
        wages = wages.append(pd.Series(np.nan))

#
dataset['cat_wages'] = wages.copy()
dataset['cat_age'] = age.copy()

#
dataset['id'] = range(1, 2001)
```

```
#
data = dataset.groupby(['sex', 'cat_wages'], as_index=False)['id'].count()
data = data.rename(columns={'id' : 'count'})

#
wages_frequencies = data['count'].copy()

pivot = data.pivot_table(values='count',index='sex',columns='cat_wages',␣
 ↪aggfunc=lambda x : x)
sns.heatmap(data=pivot, annot=True, fmt='d')
plt.show()

data = dataset.groupby(['cat_age', 'cat_wages'], as_index=False)['id'].count()
data = data.rename(columns={'id' : 'count'})
pivot = data.pivot_table(values='count',index='cat_age',columns='cat_wages',␣
 ↪aggfunc=lambda x : x)
sns.heatmap(data=pivot, annot=True)
plt.show()

data = dataset.groupby(['Educat', 'cat_wages'], as_index=False)['id'].count()
data = data.rename(columns={'id' : 'count'})
pivot = data.pivot_table(values='count',index='Educat',columns='cat_wages',␣
 ↪aggfunc=lambda x : x)
fig, ax = plt.subplots(figsize=(20, 4))
sns.heatmap(data=pivot, annot=True, fmt='d', ax=ax)
plt.show()

data = dataset.groupby(['Educat', 'sport'], as_index=False)['id'].count()
data = data.rename(columns={'id' : 'count'})
pivot = data.pivot_table(values='count',index='Educat',columns='sport',␣
 ↪aggfunc=lambda x : x)
fig, ax = plt.subplots(figsize=(20, 7))
sns.heatmap(data=pivot, annot=True, fmt='f', ax=ax)
plt.show()

data = dataset.groupby(['sex', 'smoker'], as_index=False)['id'].count()
data = data.rename(columns={'id' : 'count'})
pivot = data.pivot_table(values='count',index='sex',columns='smoker',␣
 ↪aggfunc=lambda x : x)
sns.heatmap(data=pivot, annot=True, fmt='d')
plt.show()

# -    (    )
male_wages_frequencies = pd.Series([])
female_wages_frequencies = pd.Series([])
for i in range(len(wages_frequencies)):
    if i < len(wages_frequencies)/2:
```

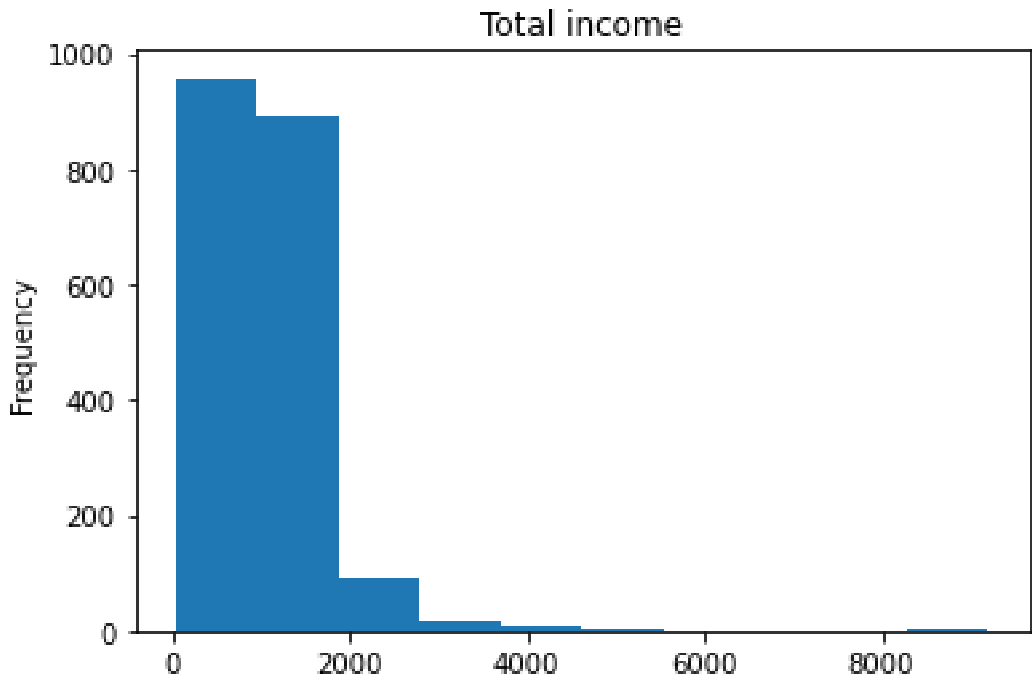
```
        female_wages_frequencies = female_wages_frequencies.append(pd.
 ↪Series([wages_frequencies[i]]), ignore_index=True)
    else:
        male_wages_frequencies = male_wages_frequencies.append(pd.
 ↪Series([wages_frequencies[i]]), ignore_index=True)

#
print("-       ")
print(ss.chisquare(male_wages_frequencies, female_wages_frequencies))
print()

dat = dataset.groupby(['sex', 'Educat'], as_index=False)['id'].count()
dat = dat.rename(columns={'id' : 'count'})
education_frequencies = dat['count'].copy()


male_education_frequencies = pd.Series([])
female_education_frequencies = pd.Series([])
for i in range(len(education_frequencies)):
    if i < len(education_frequencies)/2:
        female_education_frequencies = female_education_frequencies.append(pd.
 ↪Series([education_frequencies[i]]), ignore_index=True)
    else:
        male_education_frequencies = male_education_frequencies.append(pd.
 ↪Series([education_frequencies[i]]), ignore_index=True)

#
print("-       ")
print(ss.chisquare(male_education_frequencies, female_education_frequencies))
print()

data.drop([0, 3], inplace=True)
table = np.array([[data['count'][1], data['count'][2]], [data['count'][4],␣
 ↪data['count'][5]]])

#    ...
print("-          ")
print(ss.chi2_contingency(table))
print()



good = dataset.loc[dataset['healthev'] == 'Good']['inc_1'].dropna()
bad = dataset.loc[dataset['healthev'] == 'Bad']['inc_1'].dropna()
so_so = dataset.loc[dataset['healthev'] == 'Not very good, but not␣
 ↪bad']['inc_1'].dropna()
```

```
#    ,    ,     -

#
print("       ")
print(ss.levene(good, bad, so_so))
print(ss.f_oneway(good, bad, so_so))
print(ss.kruskal(good, bad, so_so))
print()


minsk = dataset.loc[dataset['region'] == 'Minsk city']['inc_1'].dropna()
minsk_obl = dataset.loc[dataset['region'] == 'Minsk oblast']['inc_1'].dropna()
grodno_obl = dataset.loc[dataset['region'] == 'Grodno oblast']['inc_1'].dropna()
brest_obl = dataset.loc[dataset['region'] == 'Brest oblast']['inc_1'].dropna()
gomel_obl = dataset.loc[dataset['region'] == 'Gomel oblast']['inc_1'].dropna()
vitebsk_obl = dataset.loc[dataset['region'] == 'Vitebsk oblast']['inc_1'].
 ↪dropna()
mogilev_obl = dataset.loc[dataset['region'] == 'Mogilev oblast']['inc_1'].
 ↪dropna()

#
print("      ")
print(" ")
print(ss.levene(minsk, minsk_obl, grodno_obl, brest_obl, gomel_obl, vitebsk_obl,␣
 ↪mogilev_obl))
print(ss.f_oneway(minsk, minsk_obl, grodno_obl, brest_obl, gomel_obl,␣
 ↪vitebsk_obl, mogilev_obl))
print(ss.kruskal(minsk, minsk_obl, grodno_obl, brest_obl, gomel_obl,␣
 ↪vitebsk_obl, mogilev_obl))
print()

#
#
print("  ")
print(ss.levene(minsk_obl, grodno_obl, brest_obl, gomel_obl, vitebsk_obl,␣
 ↪mogilev_obl))
print(ss.f_oneway(minsk_obl, grodno_obl, brest_obl, gomel_obl, vitebsk_obl,␣
 ↪mogilev_obl))
print(ss.kruskal(minsk_obl, grodno_obl, brest_obl, gomel_obl, vitebsk_obl,␣
 ↪mogilev_obl))
print()


#     p-value
features = ['cashinc', 'InKind', 'Privlg']
```

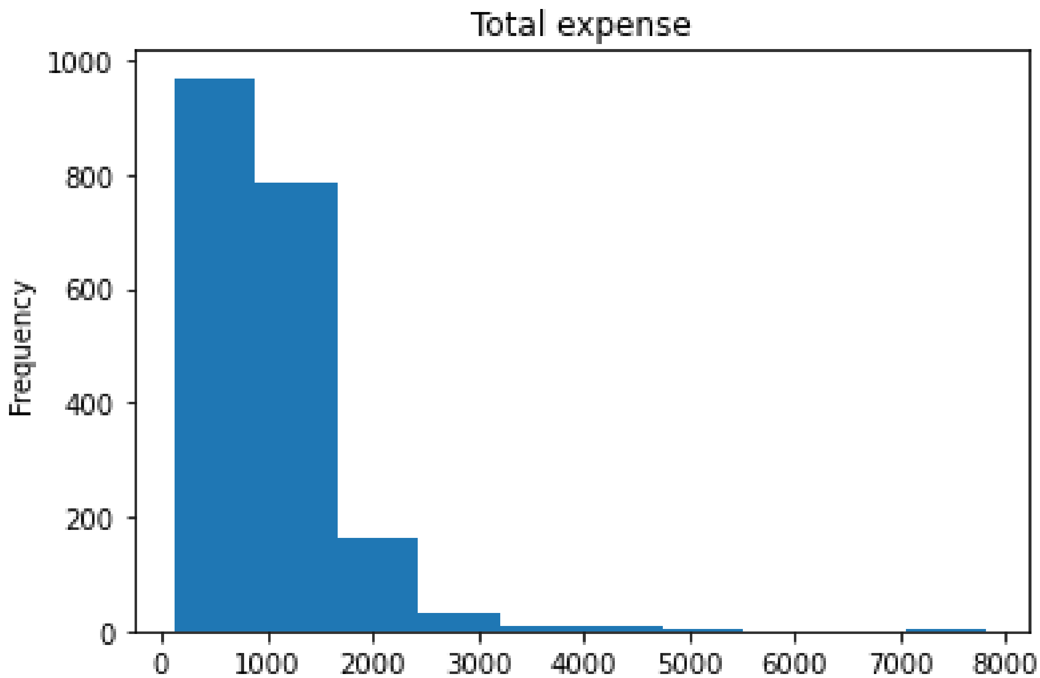
```
X = dataset[dataset['region'] == 'Grodno oblast'][features]
Y = dataset[dataset['region'] == 'Mogilev oblast'][features]

print(" ")
print(pg.multivariate_ttest(X, Y))
print()




features = ['inc_6', 'cashinc', 'InKind', 'Privlg', 'totalinc', 'exp_6', 'totalexp']
corr = dataset[features].corr()
print(" ")
print(corr)
print()

X = dataset[dataset['region'] == 'Grodno oblast'][features].corr()
Y = dataset[dataset['region'] == 'Mogilev oblast'][features].corr()
print("    ")
print()
print(X)
print()
print("    ")
print(Y)
print()

#     0,9
#    ,
#
r_inc_exp = (corr['totalinc']['totalexp'] - corr['totalinc']['cashinc']*corr['totalexp']['cashinc'])/ \
    (((1 - corr['totalinc']['cashinc']**2)*(1 - corr['totalexp']['cashinc']**2))**0.5)

r_inc_cash = (corr['totalinc']['cashinc'] - corr['totalinc']['totalexp']*corr['totalexp']['cashinc'])/ \
    (((1 - corr['totalinc']['totalexp']**2)*(1 - corr['totalexp']['cashinc']**2))**0.5)

r_exp_cash = (corr['totalexp']['cashinc'] - corr['totalinc']['totalexp']*corr['totalinc']['cashinc'])/ \
    (((1 - corr['totalinc']['totalexp']**2)*(1 - corr['totalinc']['cashinc']**2))**0.5)

print("  ")
```

```python
print('totalinc and totalexp without cashinc', r_inc_exp)
print('totalinc and cashinc without totalexp', r_inc_cash)
print('totalexp and cashinc without totalinc', r_exp_cash)
print()



#///
dataset.dropna(inplace=True)
features = ['inc_6', 'InKind', 'Privlg']
X = dataset[features]
y = dataset['exp_6']


X2 = sm.add_constant(X)
est = sm.OLS(y, X2)
est2 = est.fit()
print("  exp_6:  ")
print()
print(est2.summary())
print()
print()

features = ['inc_6', 'InKind', 'Privlg', 'exp_6']
X = dataset[features]
y = dataset['totalexp']

X2 = sm.add_constant(X)
est = sm.OLS(y, X2)
est2 = est.fit()
print("  totalexp:  ")
print()
print(est2.summary())
print()
print()


print("  ")
print("      ")
print(pg.anova(dv='inc_1', between=['cat_age', 'sex', 'Educat'], data=dataset, ss_type=1))
print(pg.anova(dv='inc_1', between=['cat_age', 'sex', 'Educat'], data=dataset, ss_type=2))
print(pg.anova(dv='inc_1', between=['cat_age', 'sex', 'Educat'], data=dataset, ss_type=3))
```

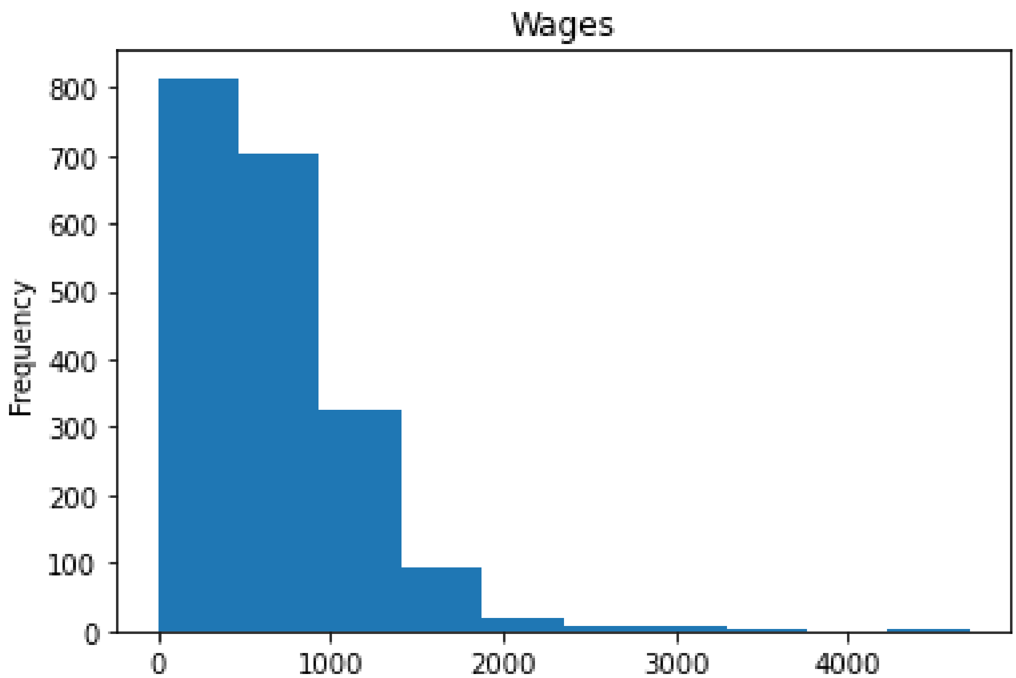
```
mean 1062.4172252122141
median 961.2480087499999
mode [1156.26068958 1278.85764167 2131.73807917]
std 604.1450053990342
skewness 4.1654086557333905
kurtosis 39.69806415891949
```



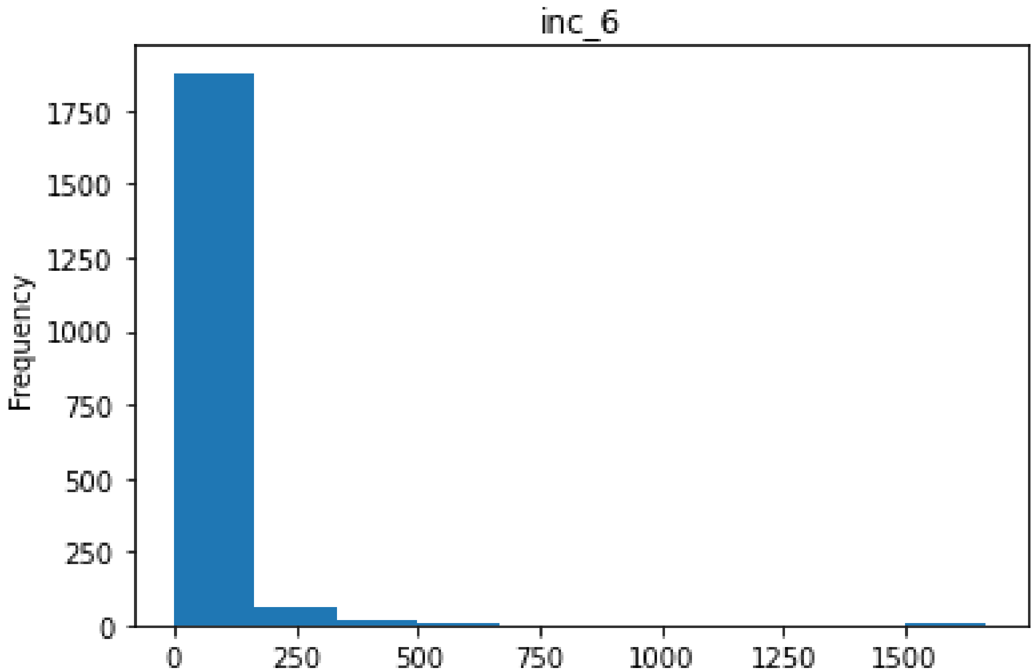

```
mean 1033.7273549132296
median 901.5729166666666
mode [ 963.75416667 1293.09791667 2445.29      ]
std 644.0220783835128
skewness 3.5101625084956645
kurtosis 23.54865183381775
```



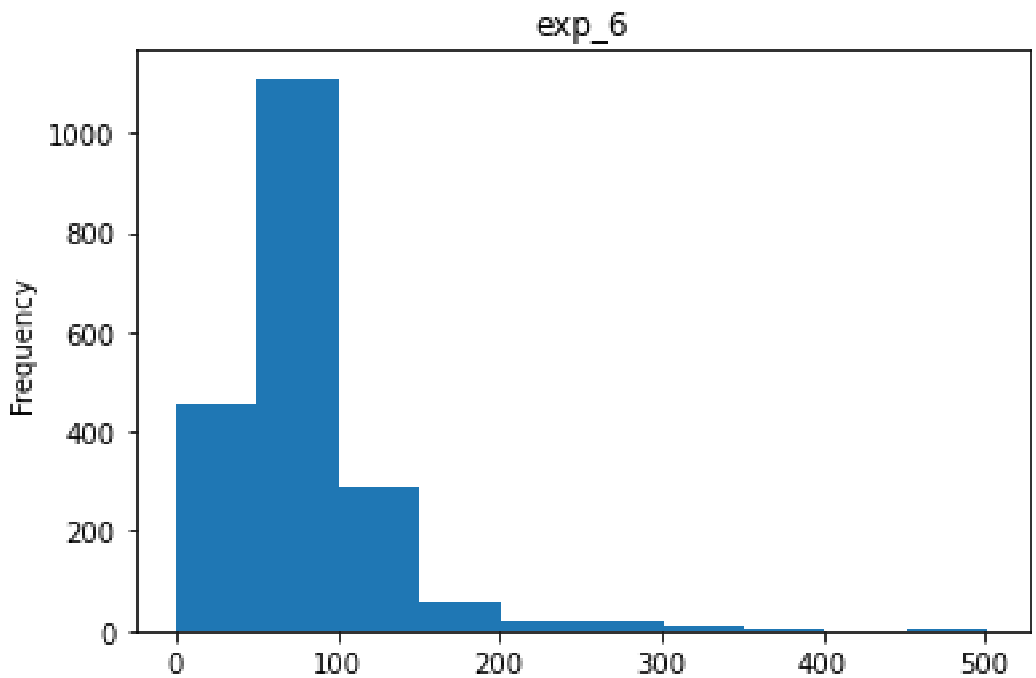

```
mean 627.8884079896326
median 570.0454166666666
mode [0.]
std 545.1554789499738
skewness 1.724278727441116
kurtosis 7.158471888324931
```



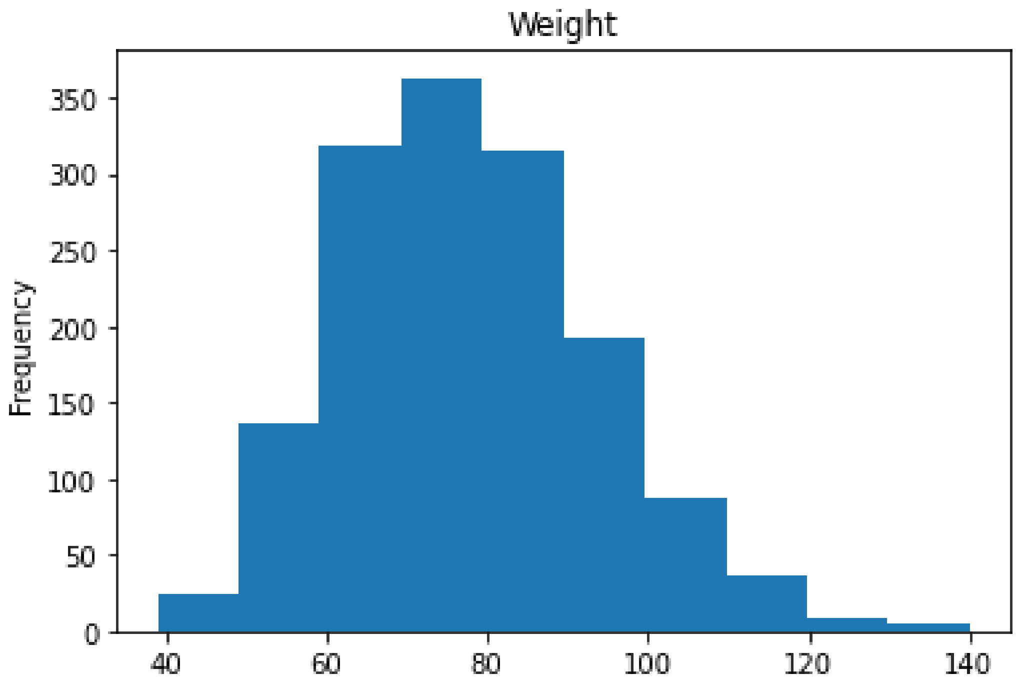

```
mean 19.68769706445797
median 0.0
mode [0.]
std 93.1020026748384
skewness 10.328334977332863
kurtosis 155.26143367139073
```



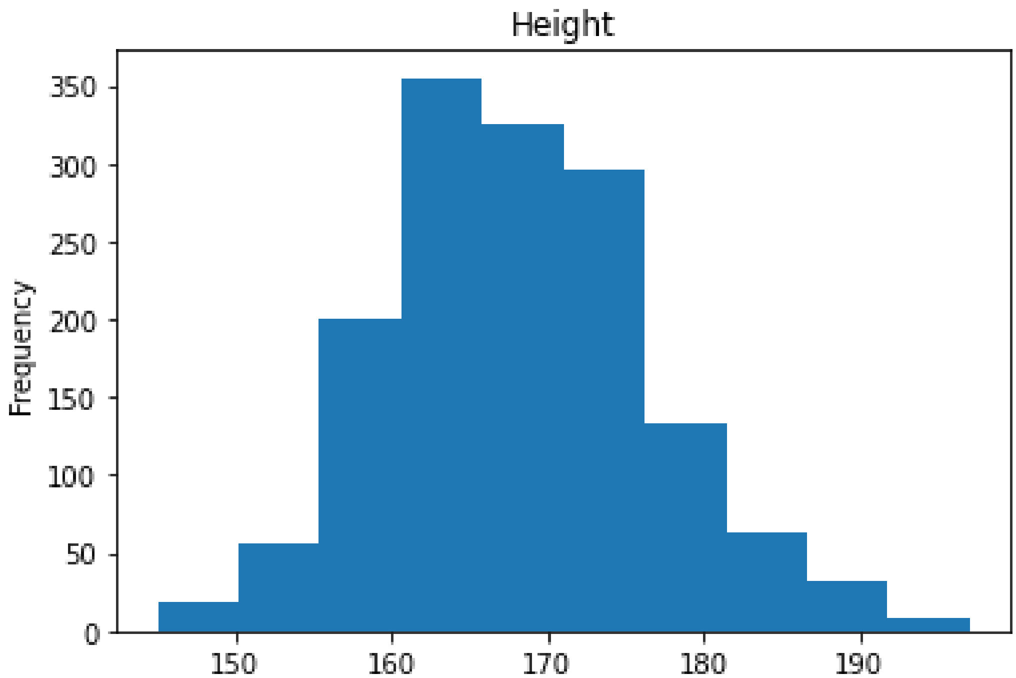

```
mean 80.2368455600631
median 69.07416666666666
mode [0.]
std 49.98299627189684
skewness 2.926749156418027
kurtosis 13.398822013249607
```



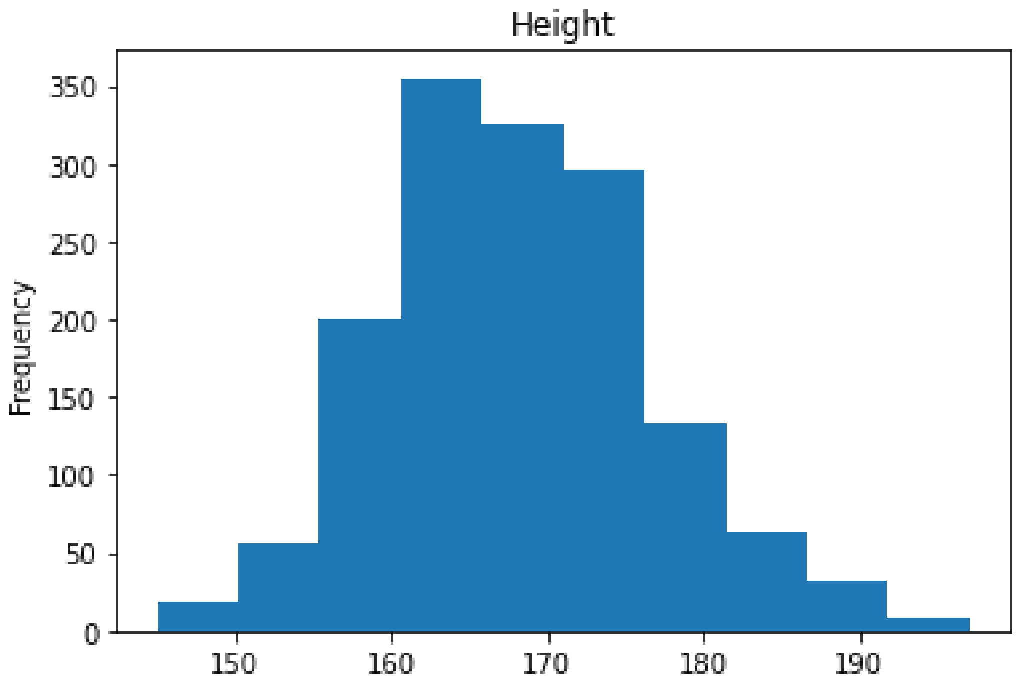

```
mean 77.0724832214765
median 76.0
mode <IntegerArray>
[80]
Length: 1, dtype: Int64
std 15.41200834469597
skewness 0.4783974214918843
kurtosis 0.31446811078435655
```



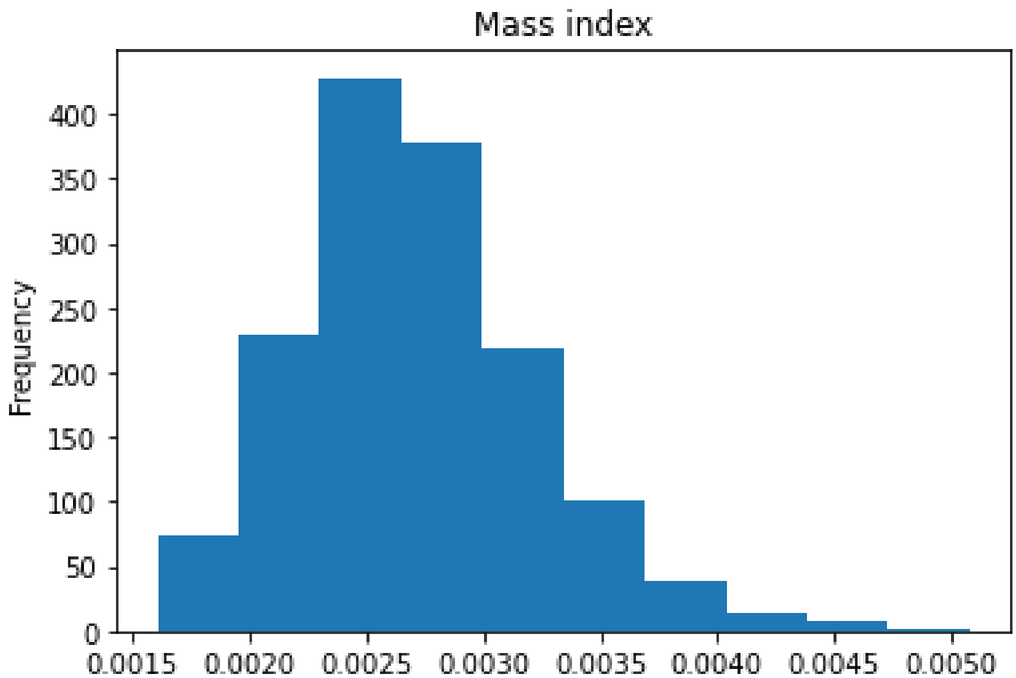

```
mean 168.53557046979867
median 168.0
mode <IntegerArray>
[164]
Length: 1, dtype: Int64
std 8.44054841019285
skewness 0.2986055965042523
kurtosis -0.0920117714571087
```



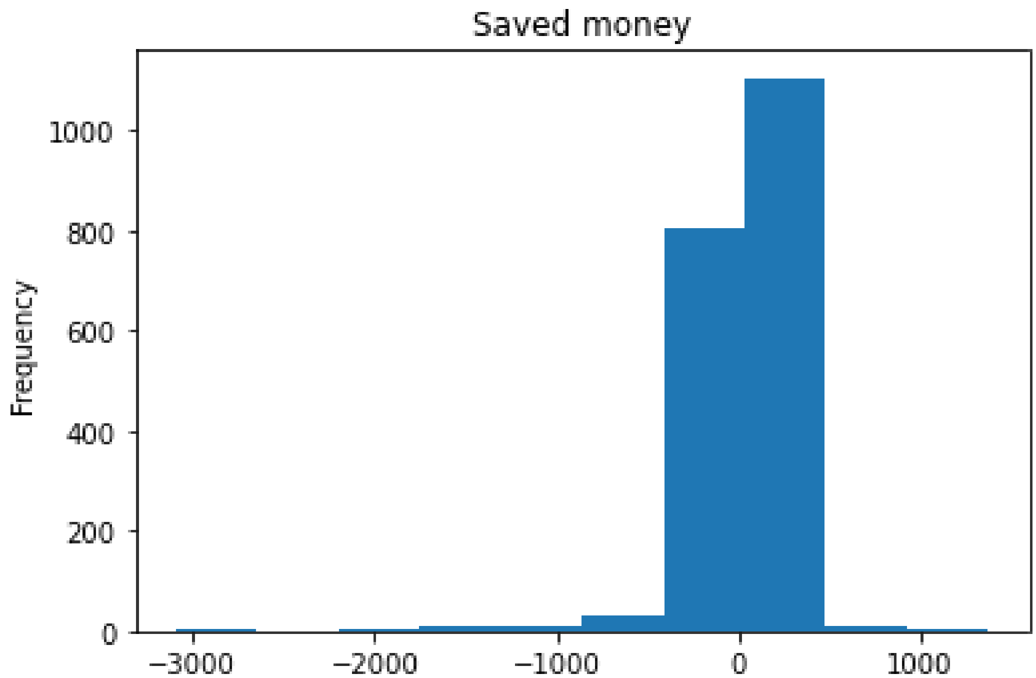

```
mean 40.9395
median 43.0
mode <IntegerArray>
[60, 62]
Length: 2, dtype: Int64
std 22.70761232668
skewness -0.14659718021726592
kurtosis -1.0475493406701586
```

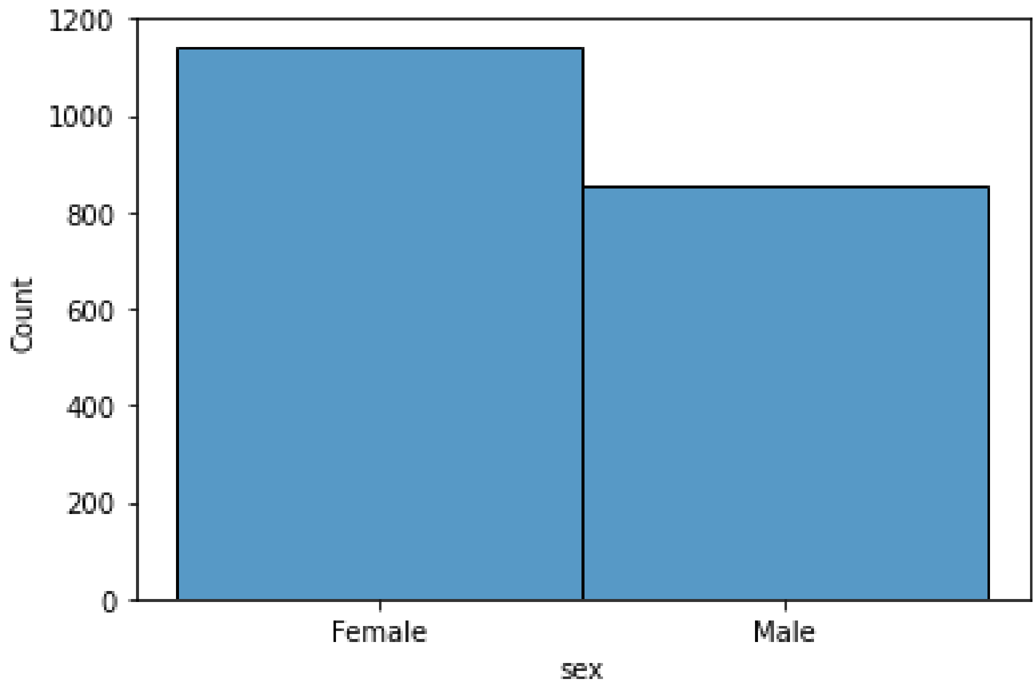


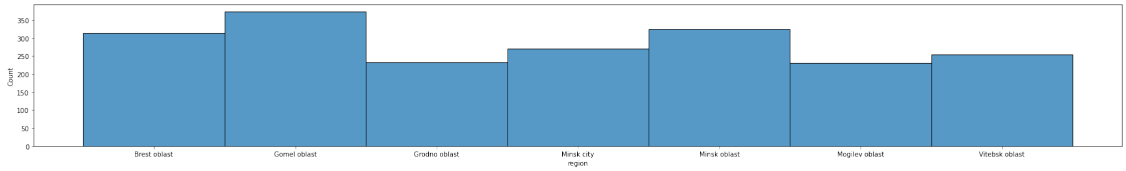

```
mean 0.0027113183574392634
median 0.002657312925170068
mode [0.00297442]
std 0.0005008738683704255
skewness 0.6339346375324817
kurtosis 0.6724687599406525
```

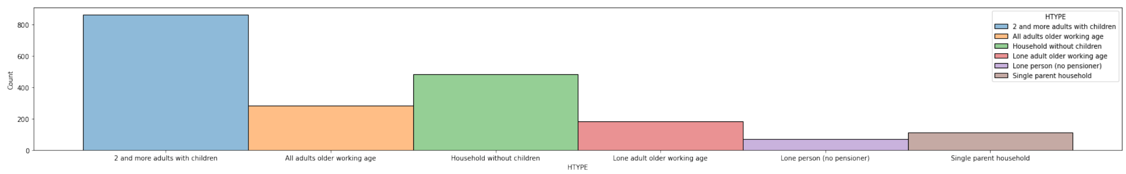


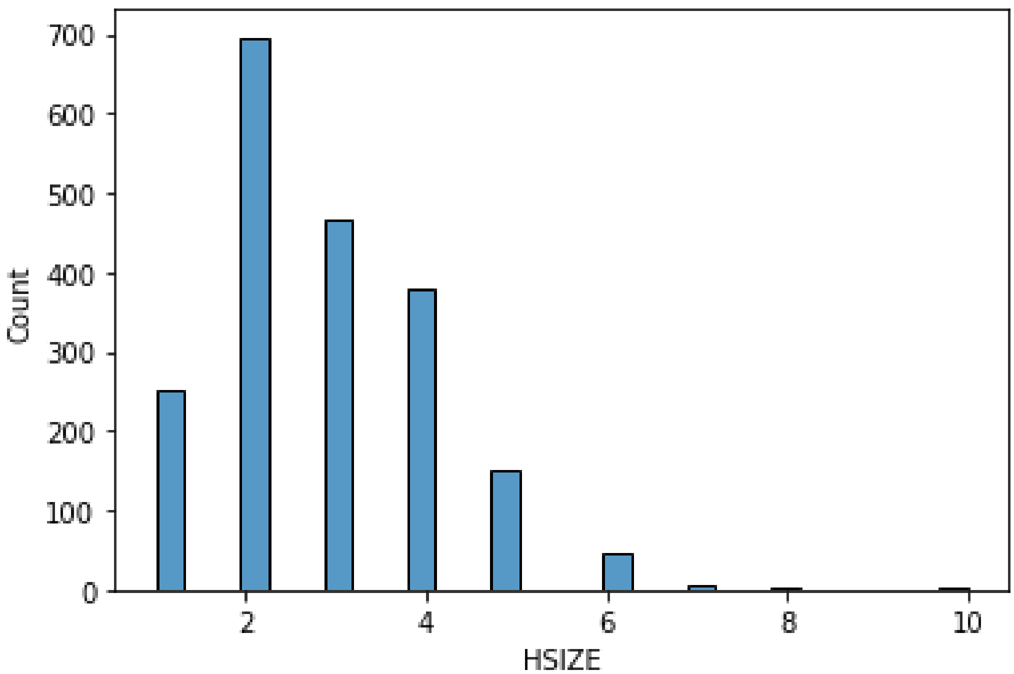

```
mean 28.689870298984406
median 48.04007291666642
mode [-313.55192083  -14.240275    192.50652292]
std 229.98657369682442
skewness -4.4055094421554255
kurtosis 46.123287453033576
```

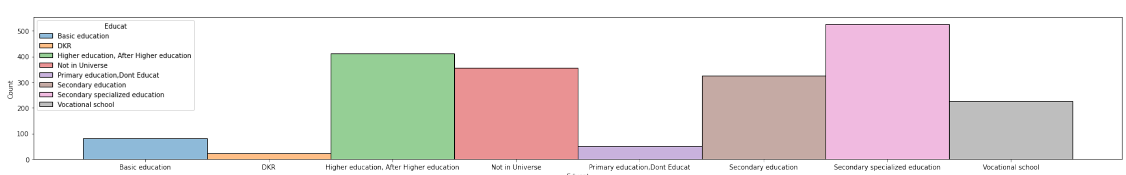

Saved money
Frequency
0
200
400
600
800
1000
−3000
−2000
−1000
0
1000

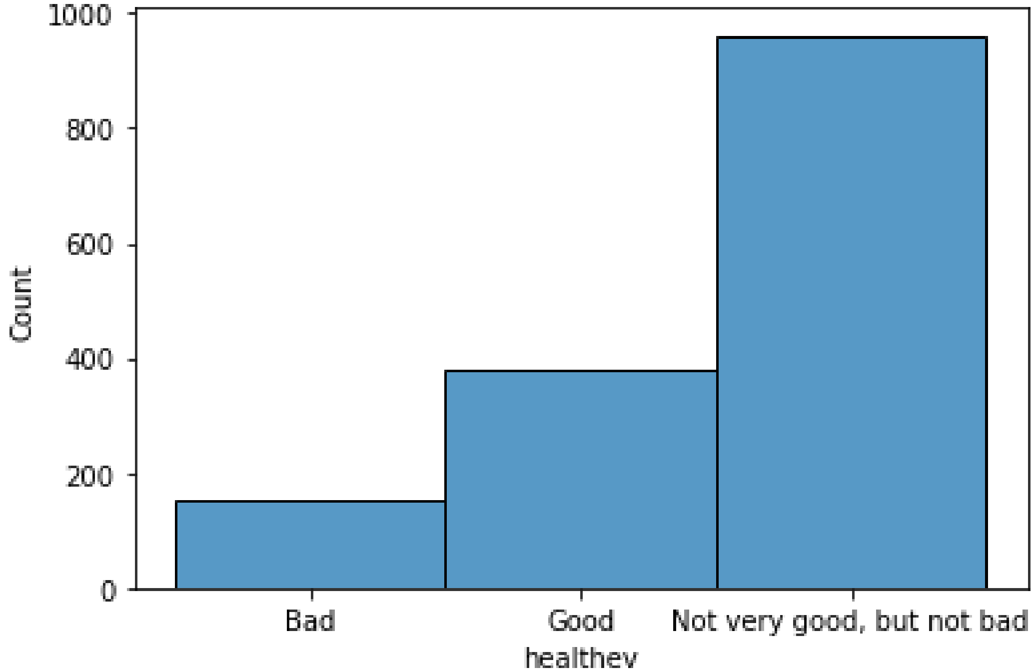

Count
0
200
400
600
800
1000
1200
Female
Male
sex


Count
Brest oblast
Gomel oblast
Grodno oblast
Minsk city
Minsk oblast
Mogilev oblast
Vitebsk oblast
region

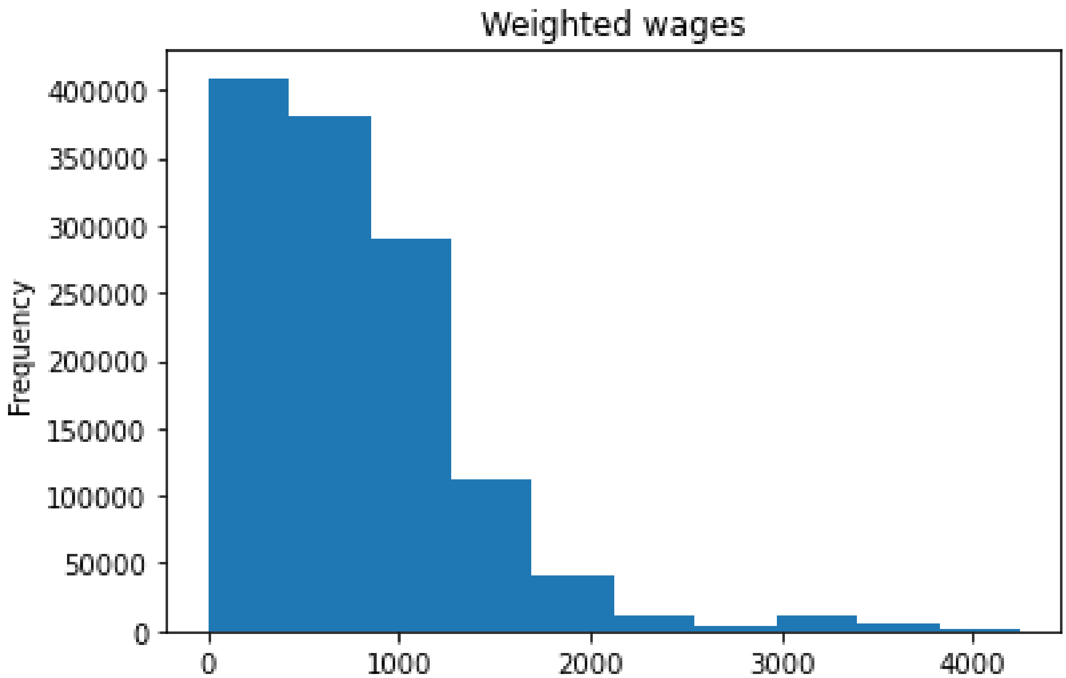

HTYPE
2 and more adults with children
All adults older working age
Household without children
Lone adult older working age
Lone person (no pensioner)
Single parent household
Count
HTYPE


Count
HSIZE

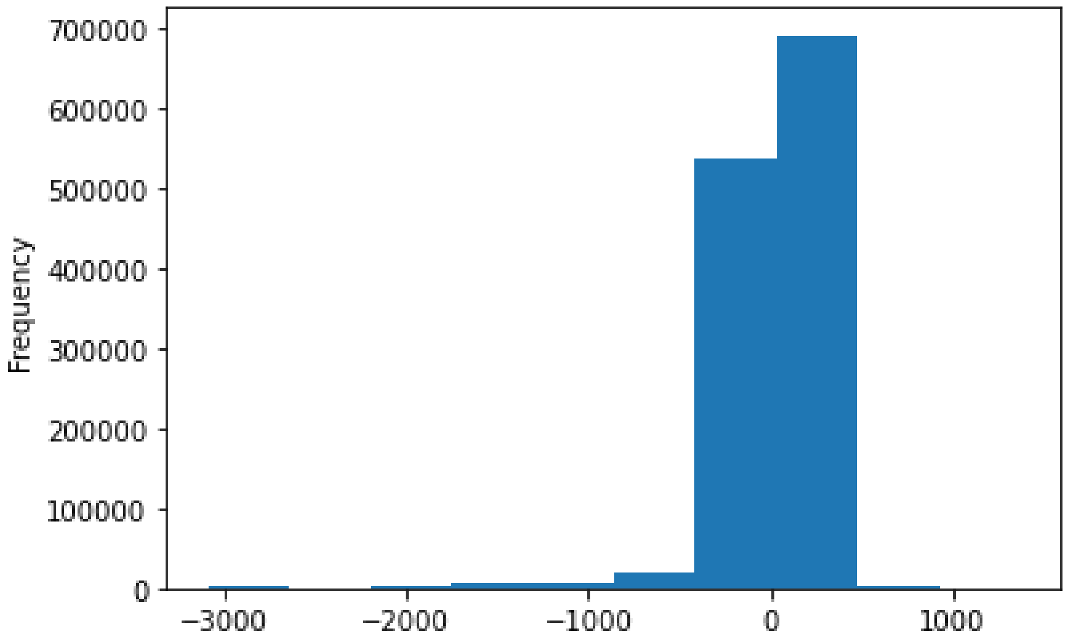

Educat
Basic education
DKR
Higher education, After Higher education
Not in Universe
Primary education,Dont Educat
Secondary education
Secondary specialized education
Vocational school
Count
Educat



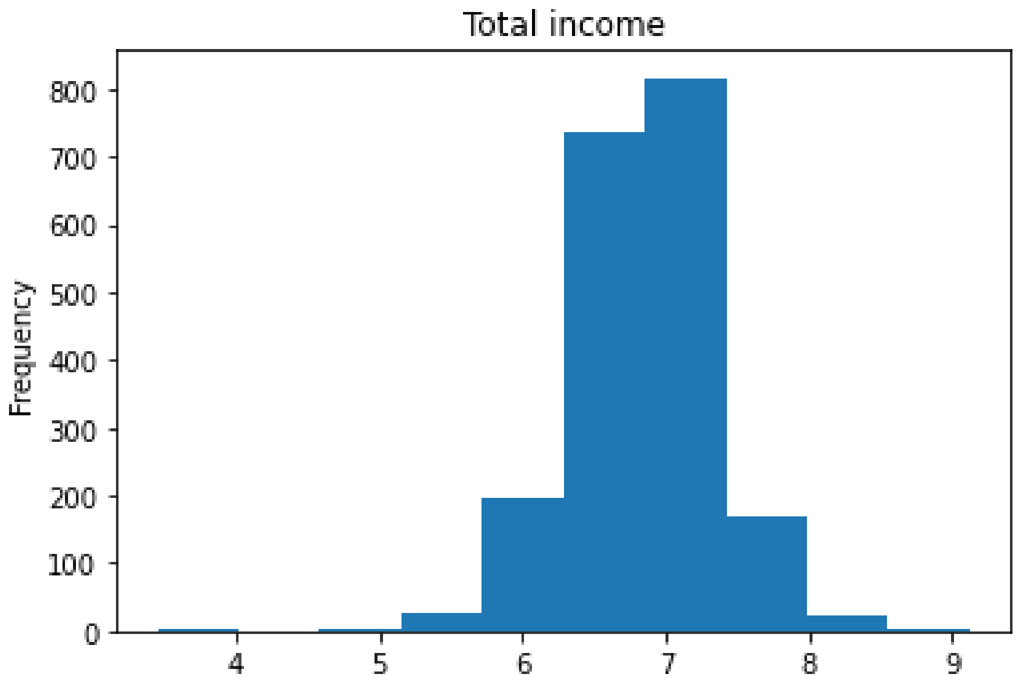

```
mean 735.7465187965693
median 661.1458333333334
mode [0.]
std 613.7362776378013
skewness 1.4103980798170492
kurtosis 3.7565191622449716
```



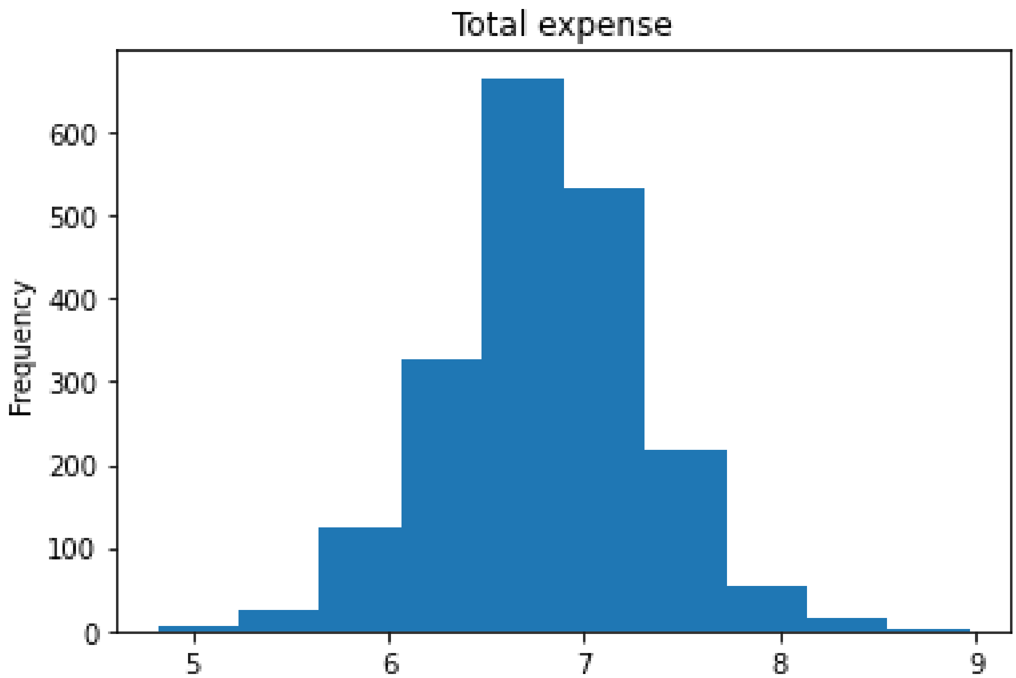

```
mean 11.502298818794205
median 41.09142041666661
mode [-14.240275]
std 278.4133297885361
skewness -5.420171708580864
kurtosis 45.89559296325041
```



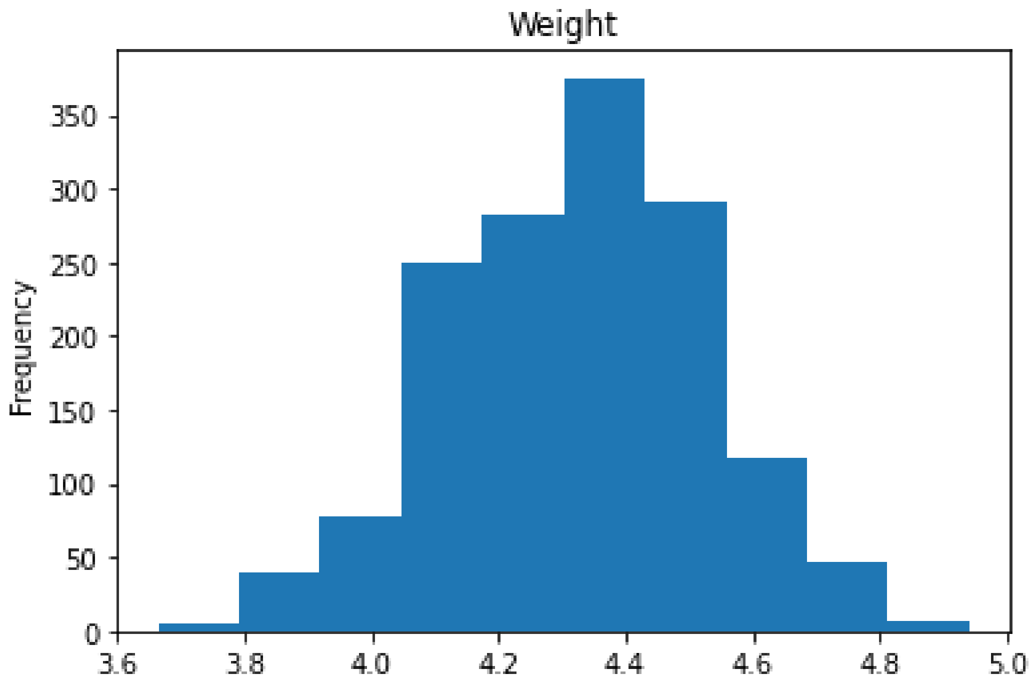

```
mean 6.847886693139559
median 6.868232448193947
mode [7.05294653 7.15372249 7.66469293]
std 0.49268781531114125
skewness -0.3370689886291935
kurtosis 2.9796806124014905
```



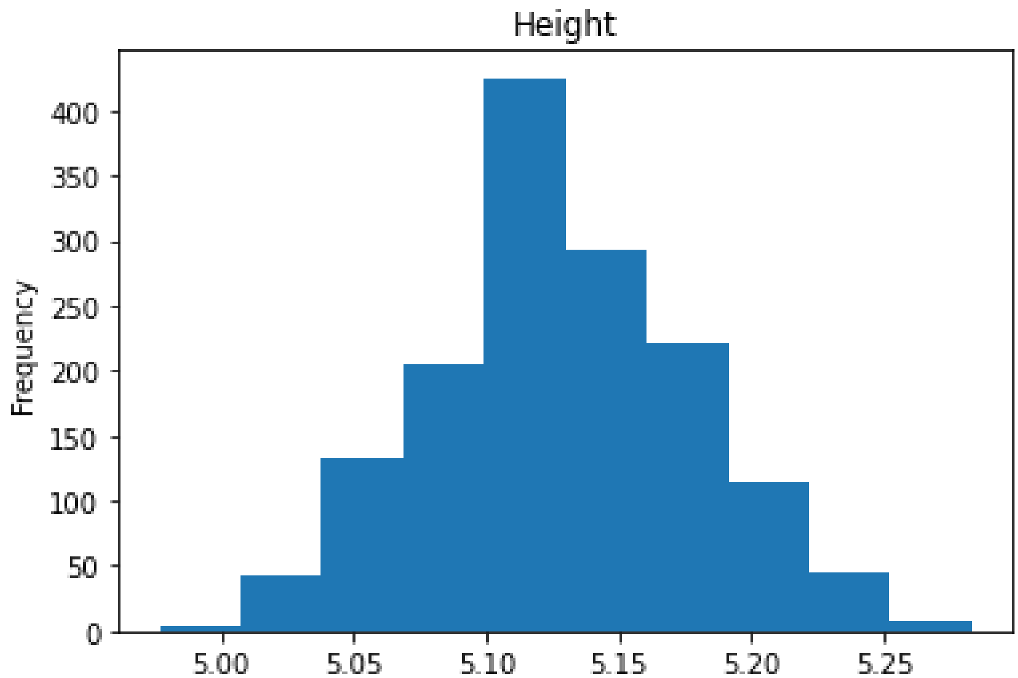

```
mean 6.80128314349141185
median 6.804140923084792
mode [6.87083625 7.1647961  7.801919  ]
std 0.5186757341791384
skewness 0.09558239389779825
kurtosis 0.8888831389351228
```

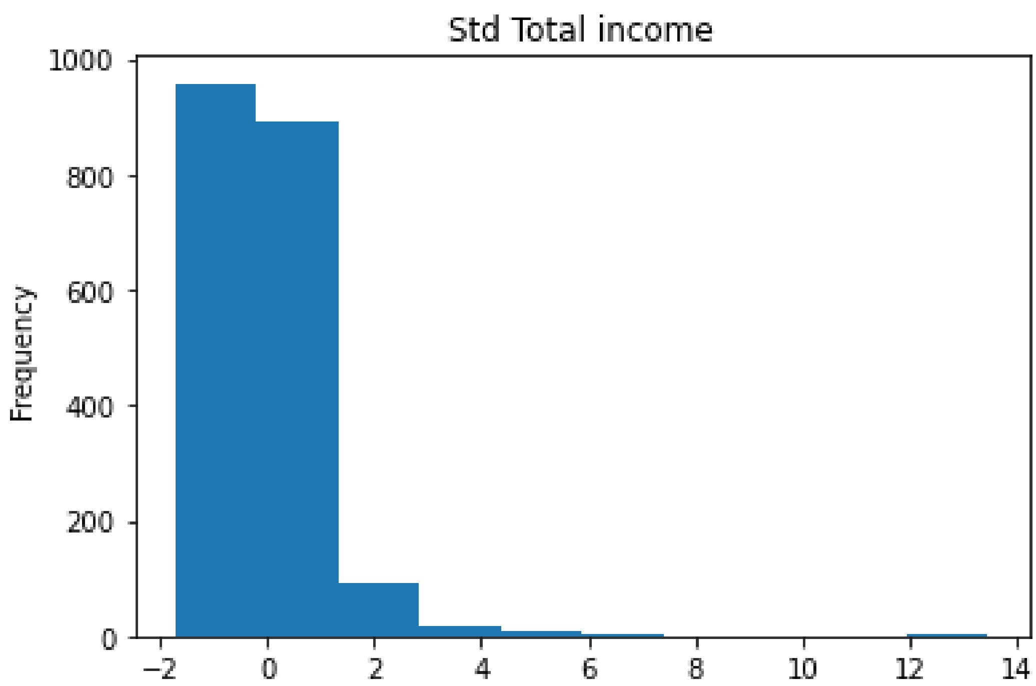


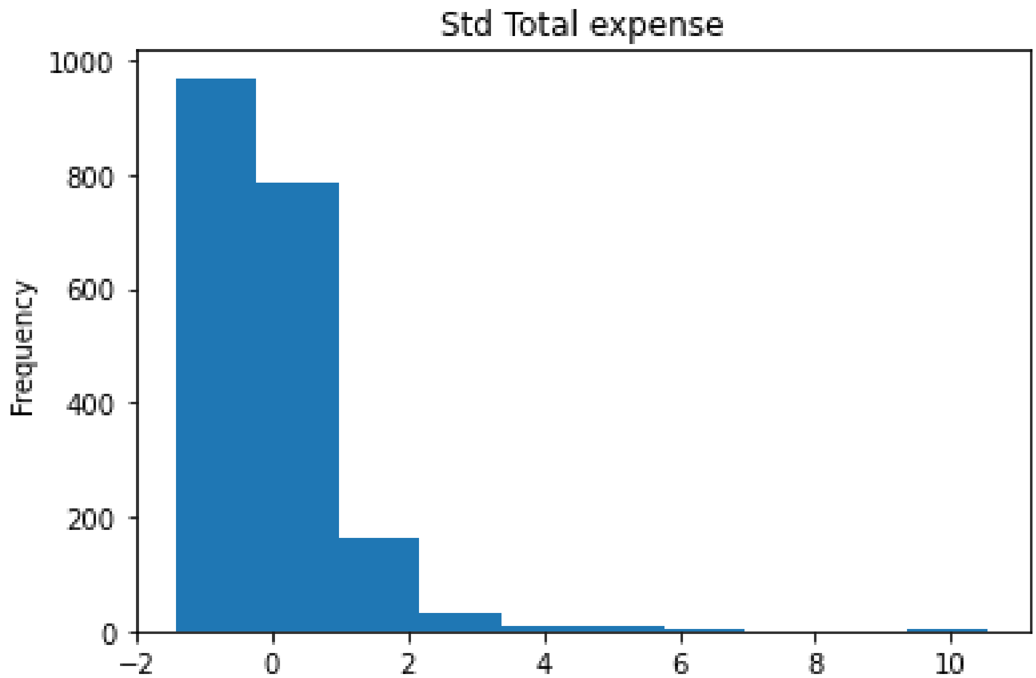

```
mean 4.324863638152l7
median 4.330733340286331
mode [4.38202663]
std 0.20017245106974269
skewness -0.10031748056408972
kurtosis -0.07175754627325759
```



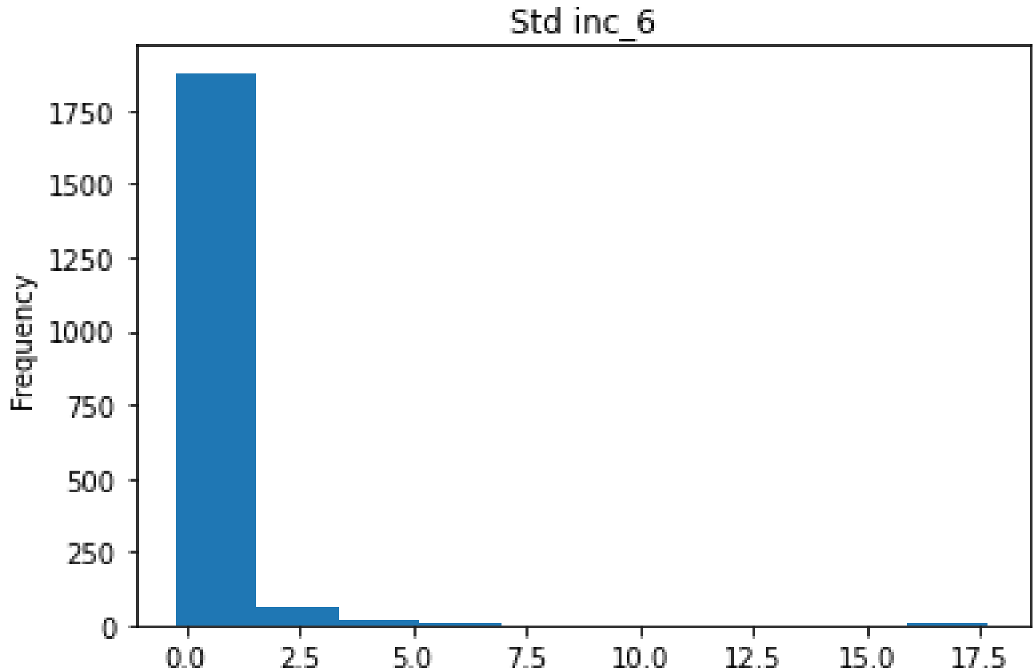

```
mean 5.125901618739775
median 5.123963979403259
mode [5.09986643]
std 0.04985390909357499
skewness 0.1643306665116516
kurtosis -0.1850469475973595
```

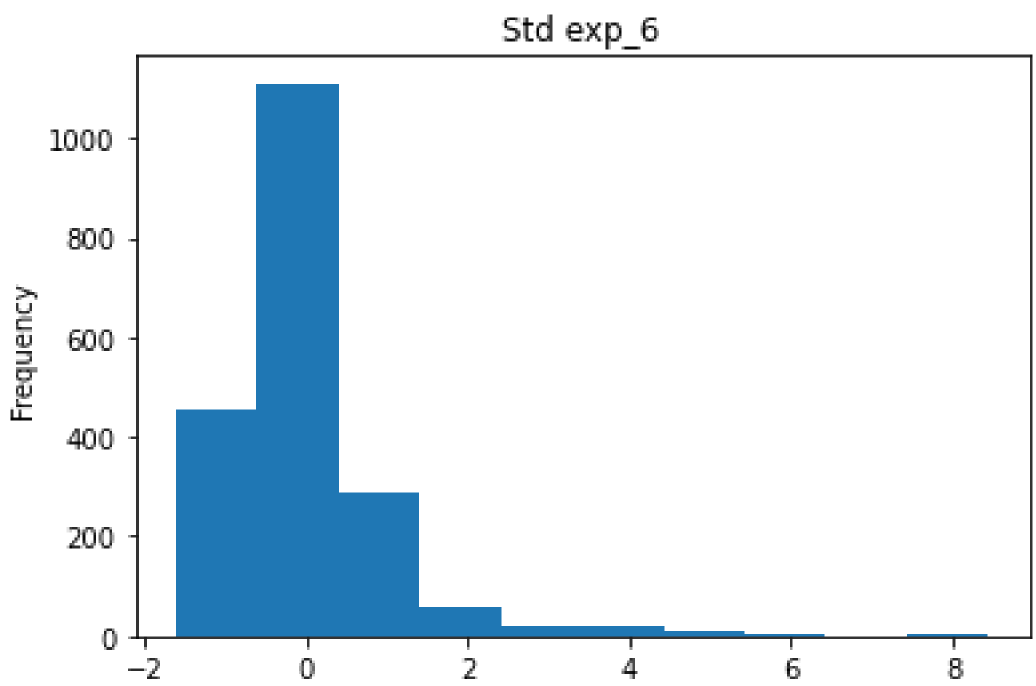

Height
Frequency
400
350
300
250
200
150
100
50
0
5.00
5.05
5.10
5.15
5.20
5.25

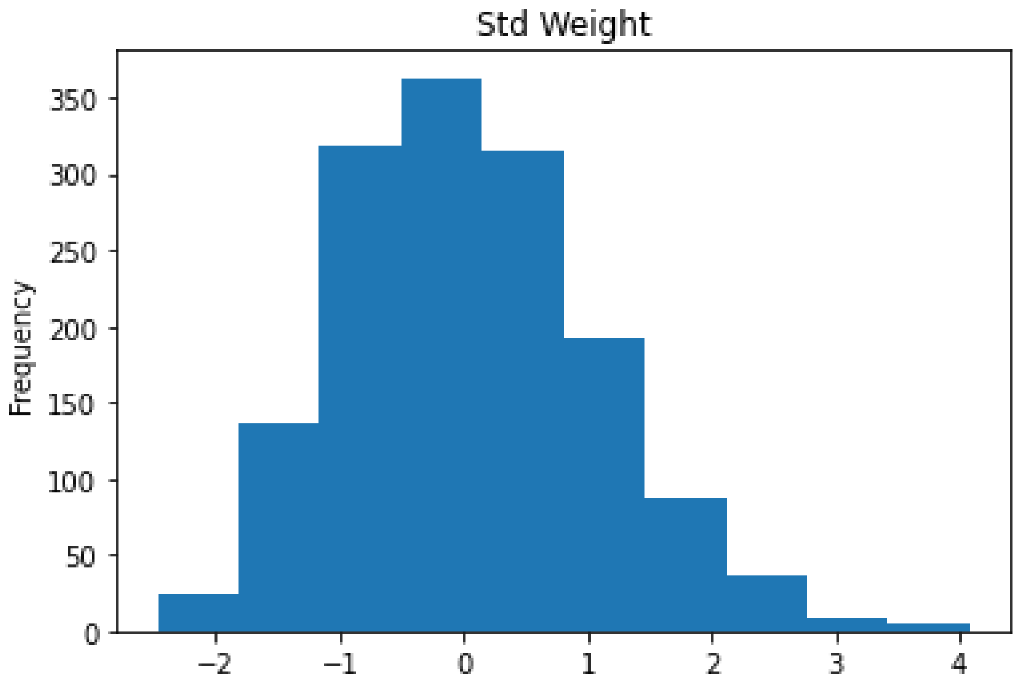

Std Total income
Frequency
1000
800
600
400
200
0
−2
0
2
4
6
8
10
12
14

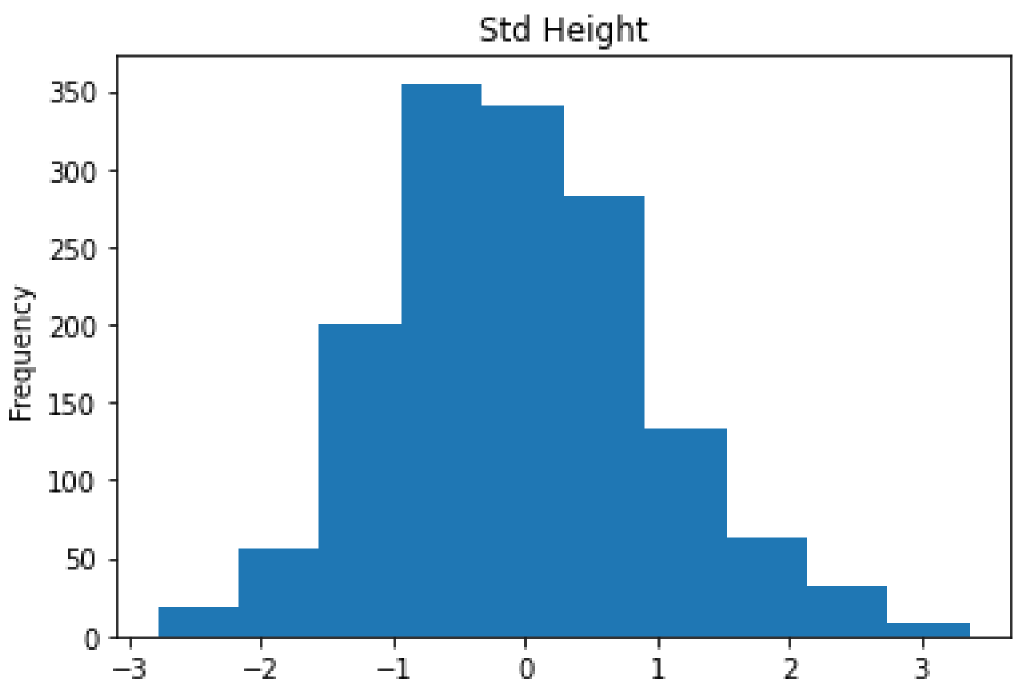

Std Total expense
Frequency
1000
800
600
400
200
0
−2
0
2
4
6
8
10

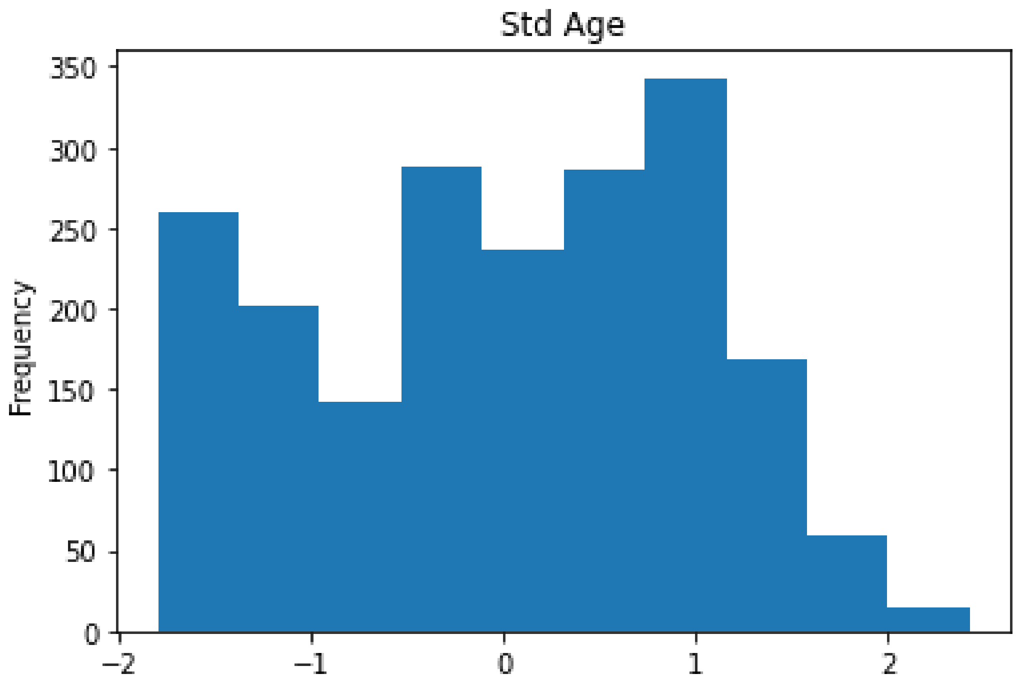

Std Wages
Frequency
800
700
600
500
400
300
200
100
0
0
2
4
6


Std inc_6
Frequency
1750
1500
1250
1000
750
500
250
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

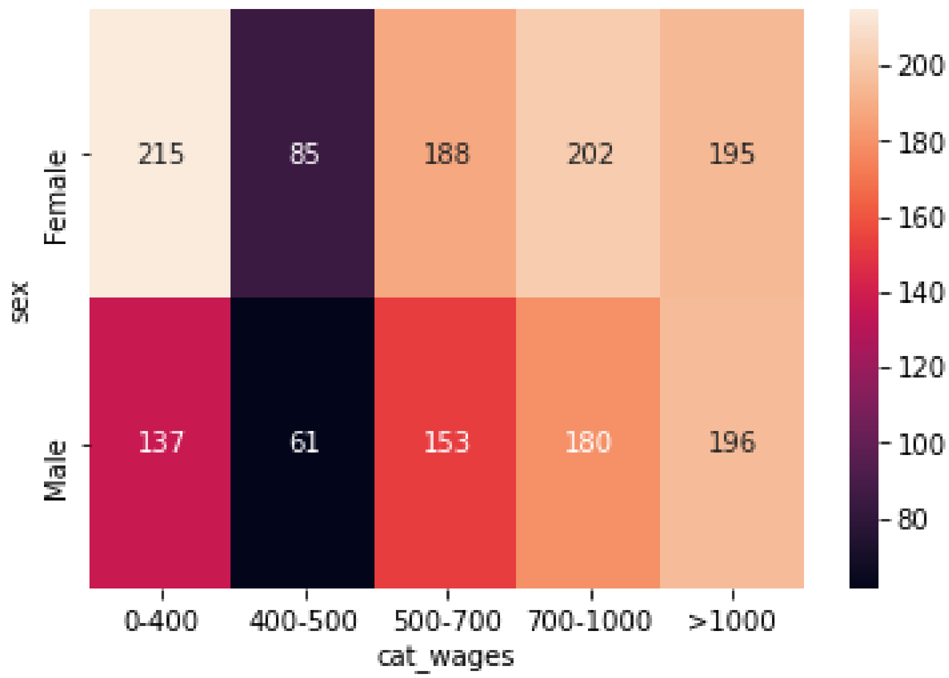

Std exp_6
Frequency
1000
800
600
400
200
0
−2
0
2
4
6
8

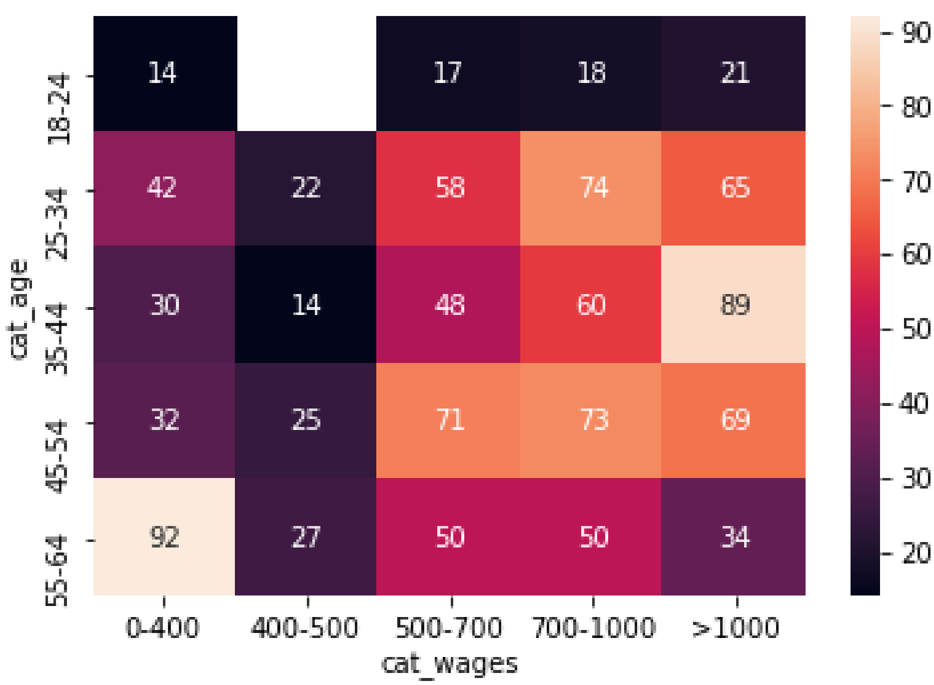

Std Weight
Frequency
350
300
250
200
150
100
50
0
−2
−1
0
1
2
3
4

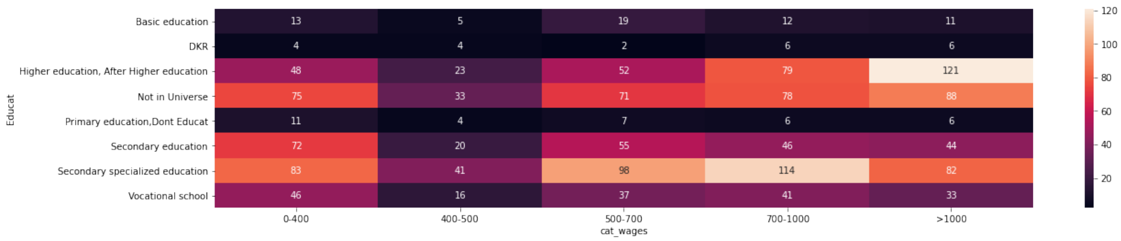

Std Height
Frequency
350
300
250
200
150
100
50
0
−3
−2
−1
0
1
2
3

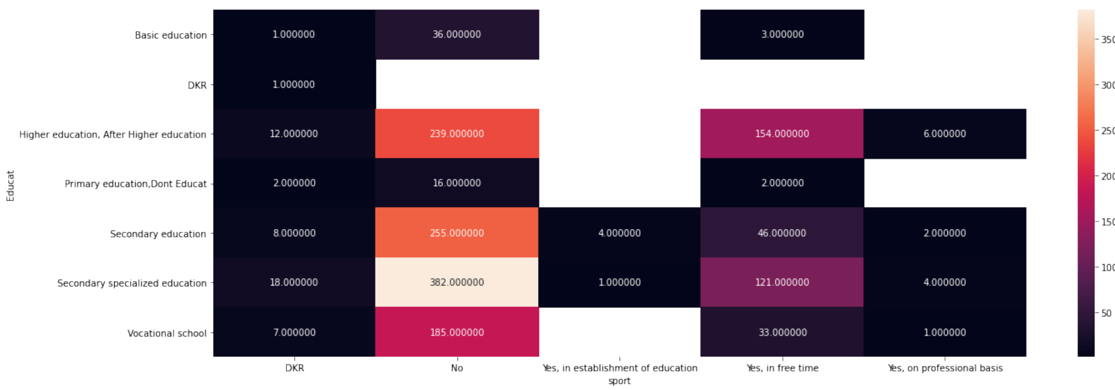


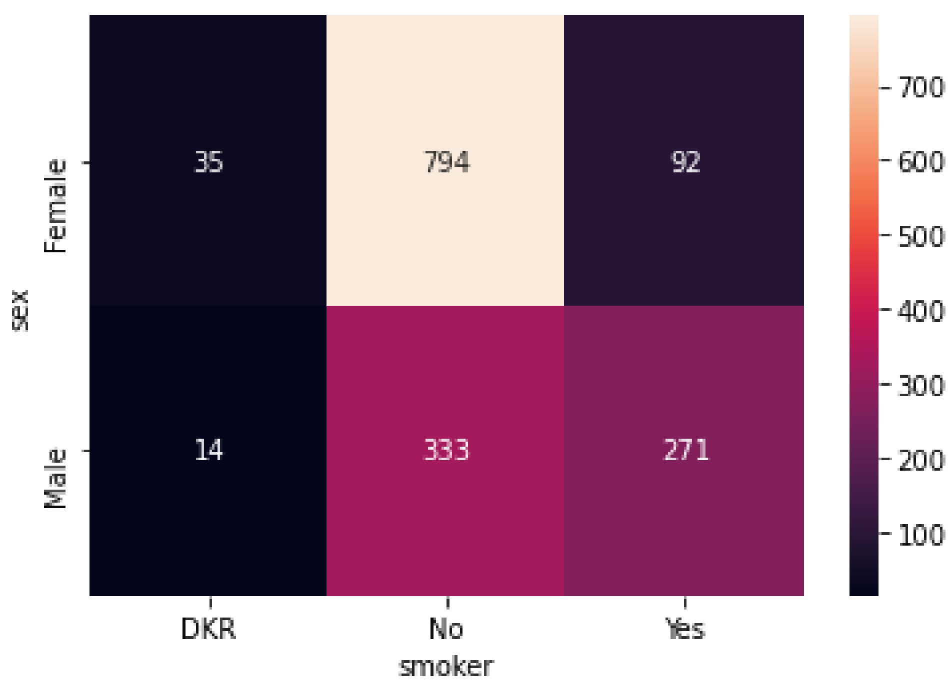

```
Ttest_1sampResult(statistic=-15.262077060148469, pvalue=8.565056071191949e-50)
Ttest_1sampResult(statistic=-145.99305571029743, pvalue=0.0)


Ttest_indResult(statistic=4.866517399106725, pvalue=1.2261113675186157e-06)
Ttest_indResult(statistic=4.842274212068753, pvalue=1.3950954655164872e-06)


LeveneResult(statistic=0.06985812305064411, pvalue=0.7915711118681439)



  inc_6
Ttest_indResult(statistic=0.5300294937207857, pvalue=0.5963494617937921)
Ttest_indResult(statistic=0.5313065424467125, pvalue=0.5954714837844448)
LeveneResult(statistic=0.2809312642139124, pvalue=0.5963494617937921)

  exp_6
Ttest_indResult(statistic=0.6514030508470554, pvalue=0.5151144835274278)
Ttest_indResult(statistic=0.653017963720647, pvalue=0.5140825093913611)
LeveneResult(statistic=1.2402812426644139, pvalue=0.26600353900125434)
```


215
85
188
202
195
137
61
153
180
196
Female
Male
sex
0-400
400-500
500-700
700-1000
>1000
cat_wages
200
180
160
140
120
100
80


14
17
18
21
42
22
58
74
65
30
14
48
60
89
32
25
71
73
69
92
27
50
50
34
18-24
25-34
35-44
45-54
55-64
cat_age
0-400
400-500
500-700
700-1000
>1000
cat_wages
90
80
70
60
50
40
30
20


Basic education
DKR
Higher education, After Higher education
Not in Universe
Primary education,Dont Educat
Secondary education
Secondary specialized education
Vocational school
Educat
0-400
400-500
500-700
700-1000
>1000
cat_wages


Basic education
DKR
Higher education, After Higher education
Primary education,Dont Educat
Secondary education
Secondary specialized education
Vocational school
Educat
DKR
No
Yes, in establishment of education
Yes, in free time
Yes, on professional basis
sport


Female
Male
sex
DKR
No
Yes
smoker

–

```
Power_divergenceResult(statistic=43.99127026273706, pvalue=6.44261894933452e-09)

-
Power_divergenceResult(statistic=159.98572731382302,
pvalue=3.2300260433465685e-31)

-
(229.90213277680573, 6.2616071037989236e-52, 1, array([[670.14899329,
215.85100671],
       [456.85100671, 147.14899329]]))


LeveneResult(statistic=3.099958169054358, pvalue=0.04534693854530934)
F_onewayResult(statistic=52.32411222110441, pvalue=1.1255287459711753e-22)
KruskalResult(statistic=152.40859724222173, pvalue=8.033293077261647e-34)



LeveneResult(statistic=12.82407341807755, pvalue=2.953438885691872e-14)
F_onewayResult(statistic=7.891051890311205, pvalue=2.0208025016642814e-08)
KruskalResult(statistic=23.817905901387288, pvalue=0.0005641022956407008)


LeveneResult(statistic=1.2399729578033607, pvalue=0.28779263975345054)
F_onewayResult(statistic=1.5087764803933286, pvalue=0.18386734203321184)
KruskalResult(statistic=10.109658058867403, pvalue=0.07218709449772381)


                 T2         F  df1  df2      pval
hotelling  0.840659  0.278991    3  454  0.840569


             inc_6   cashinc    InKind    Privlg  totalinc     exp_6  totalexp
inc_6     1.000000  0.191813  0.157950  0.166847  0.214077  0.172870  0.223615
cashinc   0.191813  1.000000  0.038320  0.074791  0.993071  0.337626  0.939652
InKind    0.157950  0.038320  1.000000  0.035987  0.143264  0.062995  0.040092
Privlg    0.166847  0.074791  0.035987  1.000000  0.129783  0.072478  0.079945
totalinc  0.214077  0.993071  0.143264  0.129783  1.000000  0.342947  0.934071
exp_6     0.172870  0.337626  0.062995  0.072478  0.342947  1.000000  0.379241
totalexp  0.223615  0.939652  0.040092  0.079945  0.934071  0.379241  1.000000



             inc_6   cashinc    InKind    Privlg  totalinc     exp_6  totalexp
inc_6     1.000000  0.259942 -0.011367  0.438367  0.280644  0.143984  0.271221
cashinc   0.259942  1.000000 -0.017465  0.172118  0.986343  0.407829  0.900762
InKind   -0.011367 -0.017465  1.000000 -0.057972  0.134825  0.062419 -0.021148
Privlg    0.438367  0.172118 -0.057972  1.000000  0.223619  0.131439  0.169346
```

```
totalinc  0.280644  0.986343  0.134825  0.223619  1.000000  0.417021  0.888538
exp_6     0.143984  0.407829  0.062419  0.131439  0.417021  1.000000  0.389518
totalexp  0.271221  0.900762 -0.021148  0.169346  0.888538  0.389518  1.000000


             inc_6   cashinc    InKind    Privlg  totalinc     exp_6  totalexp
inc_6     1.000000  0.074005  0.004341  0.018588  0.074103 -0.039339  0.041574
cashinc   0.074005  1.000000  0.038876  0.010336  0.991932  0.333700  0.963748
InKind    0.004341  0.038876  1.000000  0.222631  0.161955  0.027083  0.032383
Privlg    0.018588  0.010336  0.222631  1.000000  0.065666  0.151763 -0.010247
totalinc  0.074103  0.991932  0.161955  0.065666  1.000000  0.337021  0.954783
exp_6    -0.039339  0.333700  0.027083  0.151763  0.337021  1.000000  0.351460
totalexp  0.041574  0.963748  0.032383 -0.010247  0.954783  0.351460  1.000000


totalinc and totalexp without cashinc 0.02312050103037548
totalinc and cashinc without totalexp 0.9443277715773899
totalexp and cashinc without totalinc 0.2872354141110042

  exp_6:

                            OLS Regression Results
==============================================================================
Dep. Variable:                  exp_6   R-squared:                       0.033
Model:                            OLS   Adj. R-squared:                  0.030
Method:                 Least Squares   F-statistic:                     11.72
Date:                Sun, 27 Dec 2020   Prob (F-statistic):           1.49e-07
Time:                        20:26:01   Log-Likelihood:                -5542.7
No. Observations:                1040   AIC:                         1.109e+04
Df Residuals:                    1036   BIC:                         1.111e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         81.7179      2.267     36.049      0.000      77.270      86.166
inc_6          0.1060      0.021      5.059      0.000       0.065       0.147
InKind         0.0056      0.024      0.232      0.817      -0.042       0.053
Privlg         0.1206      0.061      1.987      0.047       0.001       0.240
==============================================================================
Omnibus:                      650.334   Durbin-Watson:                   1.913
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7685.147
Skew:                           2.718   Prob(JB):                         0.00
Kurtosis:                      15.158   Cond. No.                         141.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

specified.

```
  totalexp:

                            OLS Regression Results
==============================================================================
Dep. Variable:               totalexp   R-squared:                       0.129
Model:                            OLS   Adj. R-squared:                  0.126
Method:                 Least Squares   F-statistic:                     38.40
Date:                Sun, 27 Dec 2020   Prob (F-statistic):           5.41e-30
Time:                        20:26:01   Log-Likelihood:                -8186.1
No. Observations:                1040   AIC:                         1.638e+04
Df Residuals:                    1035   BIC:                         1.641e+04
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        809.2961     43.253     18.711      0.000     724.423     894.169
inc_6          1.2381      0.270      4.593      0.000       0.709       1.767
InKind        -0.7273      0.309     -2.357      0.019      -1.333      -0.122
Privlg         1.7293      0.773      2.238      0.025       0.213       3.246
exp_6          3.9006      0.395      9.880      0.000       3.126       4.675
==============================================================================
Omnibus:                      912.580   Durbin-Watson:                   2.011
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            36575.538
Skew:                           3.862   Prob(JB):                         0.00
Kurtosis:                      31.007   Cond. No.                         273.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

	Source	SS	DF	MS	F \
0	cat_age	1.117669e+07	4.0	2.794172e+06	12.294292
1	sex	1.612329e+06	1.0	1.612329e+06	7.094209
2	Educat	2.109862e+07	7.0	3.014089e+06	13.261921
3	cat_age * sex	2.276488e+06	4.0	5.691221e+05	2.504124
4	cat_age * Educat	1.219562e+07	28.0	4.355580e+05	1.916445
5	sex * Educat	1.245204e+06	7.0	1.778863e+05	0.782696
6	cat_age * sex * Educat	3.765653e+06	28.0	1.344876e+05	0.591742
7	Residual	2.254557e+08	992.0	2.272739e+05	NaN

```
        p-unc       np2
0  9.235143e-10  0.047232
1  7.858666e-03  0.007101
2  2.018230e-16  0.085574
3  4.082883e-02  0.009996
4  2.984190e-03  0.051317
5  6.018932e-01  0.005493
6  9.552497e-01  0.016428
7           NaN       NaN
                  Source            SS     DF            MS             F  \
0                cat_age -7.881870e-06    4.0 -1.970467e-06 -8.670012e-12
1                    sex  2.256873e-08    1.0  2.256873e-08  9.930188e-14
2                 Educat  1.202096e+09    7.0  1.717281e+08  7.555995e+02
3          cat_age * sex  7.071979e-08    4.0  1.767995e-08  7.779137e-14
4       cat_age * Educat  3.027316e+07   28.0  1.081184e+06  4.757187e+00
5           sex * Educat  5.007504e+06    7.0  7.153577e+05  3.147558e+00
6  cat_age * sex * Educat  5.568907e+06   28.0  1.988895e+05  8.751094e-01
7               Residual  2.254557e+08  992.0  2.272739e+05           NaN

          p-unc           np2
0   1.000000e+00 -3.495973e-14
1   9.999997e-01  1.001027e-16
2  2.801742e-299  8.420683e-01
3   1.000000e+00  3.136749e-16
4   2.772006e-10  1.183799e-01
5   1.385415e-02  2.172800e-02
6   6.246594e-01  2.410525e-02
7            NaN           NaN
                  Source            SS     DF            MS           F  \
0                cat_age  1.460496e+06    4.0  3.651240e+05    1.606537
1                    sex  3.723947e+05    1.0  3.723947e+05    1.638528
2                 Educat  3.811696e+08    7.0  5.445280e+07  239.591090
3          cat_age * sex  1.045254e+06    4.0  2.613135e+05    1.149774
4       cat_age * Educat  2.223961e+07   28.0  7.942718e+05    3.494778
5           sex * Educat  5.837126e+05    7.0  8.338752e+04    0.366903
6  cat_age * sex * Educat  5.568907e+06   28.0  1.988895e+05    0.875109
7               Residual  2.254557e+08  992.0  2.272739e+05         NaN

          p-unc       np2
0   1.704283e-01  0.006436
1   2.008268e-01  0.001649
2  4.866696e-189  0.628344
3   3.316519e-01  0.004615
4   2.034673e-07  0.089786
5   9.000727e-01  0.002582
6   6.246594e-01  0.024105
7            NaN       NaN
```