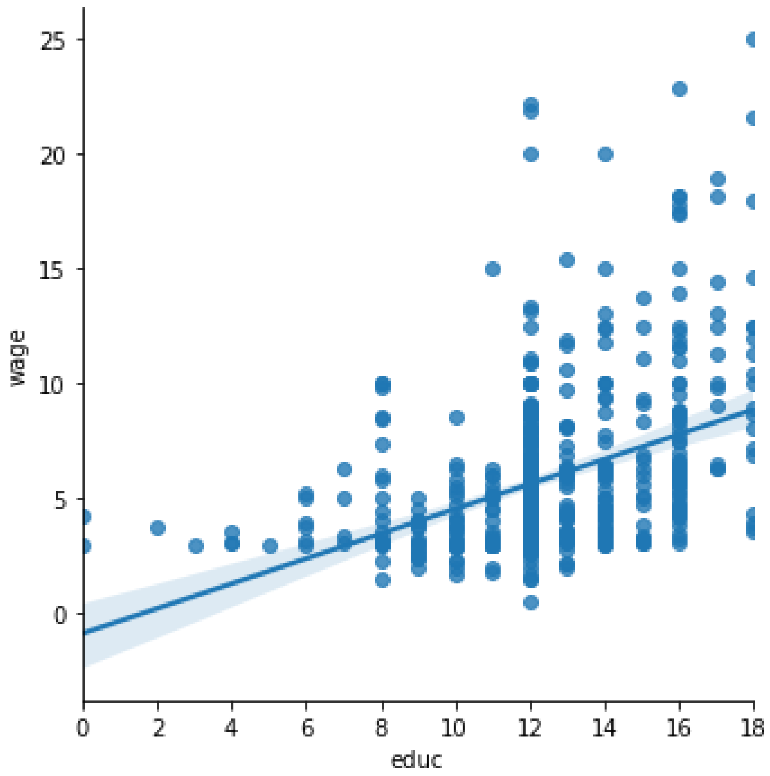
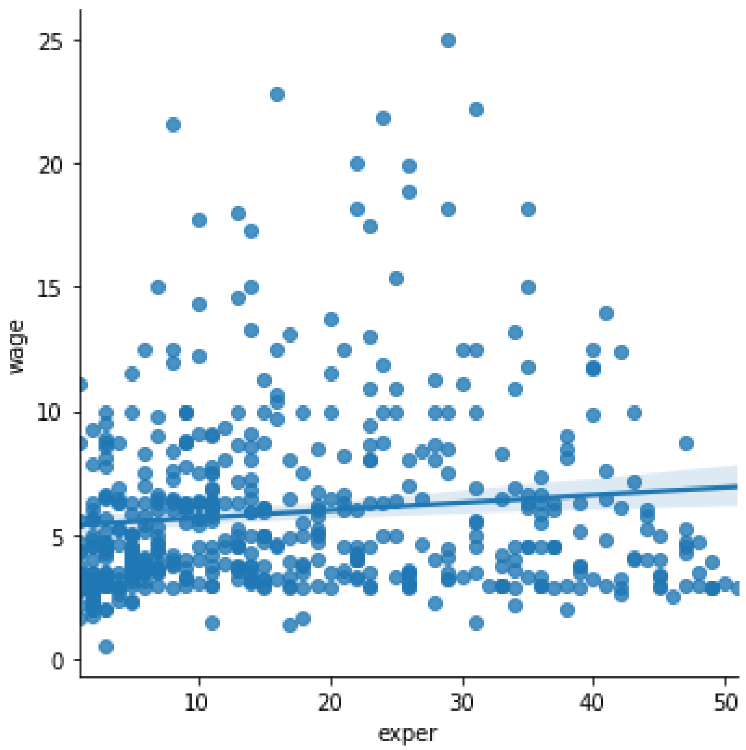
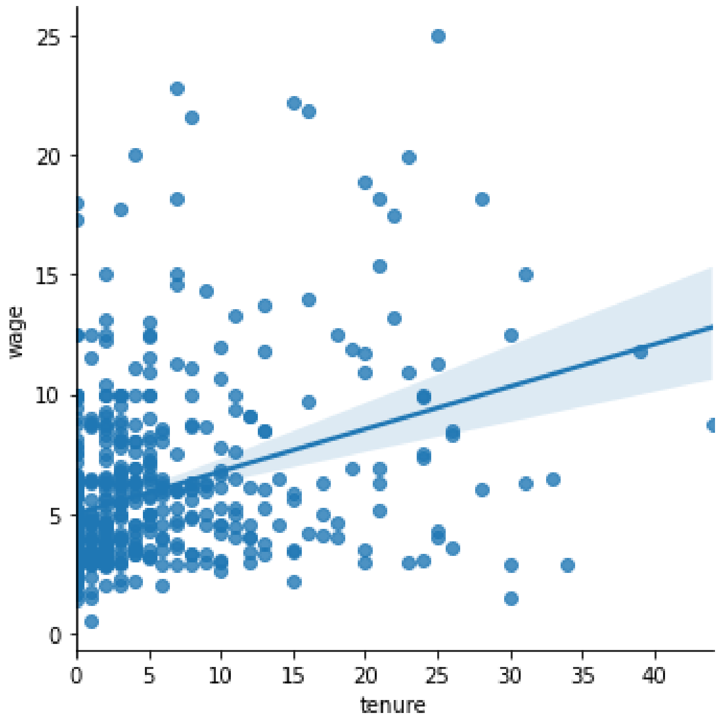
Exercise 1: wage

1. Prepare the data by using dropna. No transformed variables needed.
2. Data visualization: use scatter plot to see the correlation of educ, exper, tenure with wages. We can tell from the plots that 'educ' and 'tenure' have more correlations with wages so we can use them as the two independent variables in our model.







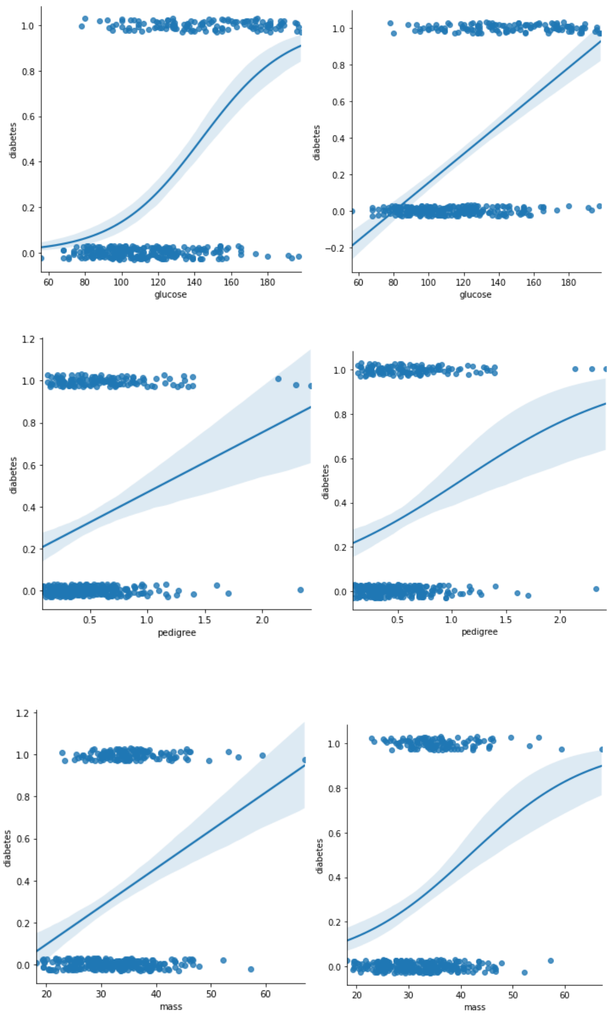
3. OLS is more suitable. As we see that the regression line never goes below 0, so there is no need to use logistic regression for this. Besides, the logit model is better to use when have multiple possible outcomes. Moreover, OLS shows a linear relationship which is more direct for us to infer the correlation.
4. Use smf here to construct model as we don't need to modify matrices but use the data frame. We use the educ and tenure as the independent variables on the right-hand side and wages on the left side. Wage (dependent variable) ~ educ + tenure (independent variable)
5. as we see from the regression line that there is no clear correlations between the wages and experience. Thus, the proposed model only employs years of educations and tenures.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   wage   R-squared:                       0.302
Model:                            OLS   Adj. R-squared:                  0.299
Method:                 Least Squares   F-statistic:                     113.1
Date:                Wed, 11 Nov 2020   Prob (F-statistic):           1.55e-41
Time:                        17:01:32   Log-Likelihood:                -1338.6
No. Observations:                 526   AIC:                             2683.
Df Residuals:                     523   BIC:                             2696.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -2.2216      0.640     -3.470      0.001      -3.479      -0.964
educ           0.5691      0.049     11.661      0.000       0.473       0.665
tenure         0.1896      0.019     10.135      0.000       0.153       0.226
==============================================================================
Omnibus:                      180.898   Durbin-Watson:                   1.791
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              654.041
Skew:                           1.568   Prob(JB):                    9.48e-143
Kurtosis:                       7.473   Cond. No.                         67.2
==============================================================================
```

6. Both the coefficients are greater than 0, so they are positively correlated to the wages. There is no p value greater than 0.05. which means it's less than 0.05 percent chance we are wrong to reject that there is no correlations of education level and tenure regarding to wages. Thus, we know that both the education level and tenure are highly correlated to the wages.
7. The R-squared is 0.302 means there is 30.2% of the data fit the regression model, and it helps to explain how well the model of prediction.
8. We plug in values of educ and tenure to see what makes wages 150. When educ equals 170 and tenure equals 293, the hourly wages is expected to be 150.

Exercise2: Diabetes

1. Prepare the data by drop NA values and replace ‘neg’ with 0 and ‘pos’ with 1.
2. Data visualization. Plot the variables to see the correlations. I made all the plots, from which I think mass, pedigree, and glucose has the highest correlation since the increase in x values leads to higher probability of diabetes (clustered dots on the upper right-hand side)



3. Logistic regression is better here. since we are analyzing the probability, and we can't have a negative probability or greater than one. So we want to make sure the regression

line is between zero and one. Otherwise it's meaningless with values exceeds the boundary, thus we prefer Logistic regression here.

4. $pr(y = 1|x,\, \beta) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2}}$

   This is the model I will use, left hand side would be the probability of getting diabetes, beta0 is constant, x1 is pedigree, x2 is glucose, and x3(extra independent variable) is mass.

5. estimate the model

   the p values for pedigree. glucose, mass are all below 0.05 which means these three terms are all shows strong correlation with diabetes. the pseudo R-square is about 27% which means 27% can be explained by this model.

```
                           Logit Regression Results
==============================================================================
Dep. Variable:               diabetes   No. Observations:                  392
Model:                          Logit   Df Residuals:                      388
Method:                           MLE   Df Model:                            3
Date:                Wed, 11 Nov 2020   Pseudo R-squ.:                  0.2698
Time:                        17:01:43   Log-Likelihood:                -181.85
converged:                       True   LL-Null:                       -249.05
Covariance Type:            nonrobust   LLR p-value:                 6.103e-29
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -8.7920      0.973     -9.037      0.000     -10.699      -6.885
pedigree       1.1715      0.414      2.829      0.005       0.360       1.983
glucose        0.0407      0.005      8.324      0.000       0.031       0.050
mass           0.0681      0.020      3.435      0.001       0.029       0.107
==============================================================================
```

6. The coefficient of pedigree is 1.17, of glucose is 0.04, of mass is 0.06. all of those coefficients are positive which means they are all positively and directly correlated to diabetes. Of these, the coefficient of pedigree is the biggest which means increase in one unit of pedigree leads to more than one-unit change in the possibility of getting diabetes, thus it's the most influential independent variable.
7. Calculate the 25th, 50th, 75th percentile for each independent variable respectively then use them to predict the corresponding diabetes probability. Find the difference between 75th and 50th, which is 0.35047404. And between 50th and 25th which is 0.16348296.

Code:

```
"""
Econ 406
Homework 5
"""

import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
```

```
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
#exercise1 Wage
def first_exercise():
    """
    generate all the output to understand the impact of different variables
    on expected wage rate.
    Returns
    -------
    None.

    """
#1.1 load the data to make sure ready for analysis
    dataset = pd.read_csv("wage.csv")
    dataset = dataset.dropna()

#1.2 data visualization
    plt.scatter(dataset['educ'], dataset['wage'])
    plt.xlabel("education")
    plt.ylabel("wages")
    plt.scatter(dataset['exper'], dataset['wage'])
    plt.xlabel("experience")
    plt.ylabel("wages")
    plt.scatter(dataset['tenure'], dataset['wage'])
    plt.xlabel("tenure")
    plt.ylabel("wages")

#1.3 OLS or Logistic regression
    sns.lmplot(x='educ', y='wage', data=dataset)
    sns.lmplot(x='exper', y='wage', data=dataset)
    sns.lmplot(x='tenure', y='wage', data=dataset)
# As we see that the regression line never goes below 0, so there is no
# need to use logistic regression for this. Besides, the logit model is better
# to use when have multiple possible outcomes.Moreover OLS should be better
# here as it shows a linear relationship which is more direct.

#1.4 data generating process for wages
#1.5 generate the regression table
    df_predict = dataset[['wage', 'educ', 'tenure']]
    mod = smf.ols(formula='wage~educ+tenure', data=df_predict)
    res = mod.fit()
    print(res.summary())
# as we see from the regression line that there is no clear correlations
# between the wages and experience. Thus the proposed modle only employs
```

```
# years of educations and tenures.

#1.6 there is no p value greater than 0.05.
# which means it's less than 0.05 percent chance we are wrong to reject that
# there is no correlations of education level and tenure regarding to wages.
# which means that both the education level and tenure are highly correlated
# to the wages.

#1.7 The R-squared is 0.302 means there is 30.2% of the data fit the
# regression modle, and it helps to explain how well the modle of prediction

#1.8
    hypo = pd.DataFrame({'wage': [150], 'educ': [170], 'tenure': [293]})
    res.predict(hypo)
# when educ equals 170 and tenure equals 293, the hourly wages is expect to
# be 150.

#exercise2: Diabetes
def second_exercise():
    """
    predict whether or not a patient has diabetes, based on certain diagnosis
    measurements included in the dataset.

    Returns
    -------
    None.

    """
#2.1 prep the data
    dataset = pd.read_csv("diabetes.csv")
    dataset = dataset.dropna()
    dataset['diabetes'] = dataset['diabetes'].replace('neg', 0)
    dataset['diabetes'] = dataset['diabetes'].replace('pos', 1)

#2.2 data visualization
    sns.lmplot(x="pedigree", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="pregnant", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="pressure", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="triceps", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="insulin", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="mass", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="mass", y="diabetes", data=dataset, logistic=True,
               y_jitter=0.03)
    sns.lmplot(x="pedigree", y="diabetes", data=dataset, y_jitter=0.03)
```

```
    sns.lmplot(x="pedigree", y="diabetes", data=dataset, logistic=True,
               y_jitter=0.03)
    sns.lmplot(x="glucose", y="diabetes", data=dataset, y_jitter=0.03)
    sns.lmplot(x="glucose", y="diabetes", data=dataset, logistic=True,
               y_jitter=0.03)

#2.3 Logistic regression is more suitable here since we are analyzing the
# probability, and we can't have a negative probability or greater than one.
# so we want to make sure the regression line is between zero and one
# otherwise it's meaningless with values exceeds the boundary, thus we prefer
# Logistic regression here.

#2.4 data genrating model for diabetes
    df_rhs = dataset[['pedigree', 'glucose', 'mass']]
    df_rhs = sm.add_constant(df_rhs)
    df_lhs = dataset['diabetes']
    logit_mod = sm.Logit(df_lhs, df_rhs)
    logit_res = logit_mod.fit()
    print(logit_res.summary())

#2.5 estimate the model
#the p values for pedigree. glucose, mass are all below 0.05 which means these
#three terms are all shows strong correlation with diabetes. the pseudo R-squ
#is about 27% wich means 27% can be explained by this model.

#2.6 the coefficient of pedigree is 1.17, of glucose is 0.04, of mass is 0.06.
#all of those coefficients are positive which means they are all positively
# and directly correlated to diabetes. Of these, the coefficient of pedigree
# is the biggest which means increase in one unit of pedigree leads to more
# than one unit change in the possibility of getting diabetes, thus it's the
# most influential independent variable.

#2.7 print out difference of the possibility for patient with 50th percentile
# and 75th and 25th percentile regarding the independent variables.
    percent_25 = logit_res.predict([1, np.percentile(dataset['pedigree'], 25),
                                    np.percentile(dataset['glucose'], 25),
                                    np.percentile(dataset['mass'], 25)])
    percent_50 = logit_res.predict([1, np.percentile(dataset['pedigree'], 50),
                                    np.percentile(dataset['glucose'], 50),
                                    np.percentile(dataset['mass'], 50)])
    percent_75 = logit_res.predict([1, np.percentile(dataset['pedigree'], 75),
                                    np.percentile(dataset['glucose'], 75),
                                    np.percentile(dataset['mass'], 75)])
    diff_75_and_50 = percent_75 - percent_50
```

```
print(diff_75_and_50)
diff_25_and_50 = percent_50 - percent_25
print(diff_25_and_50)
```