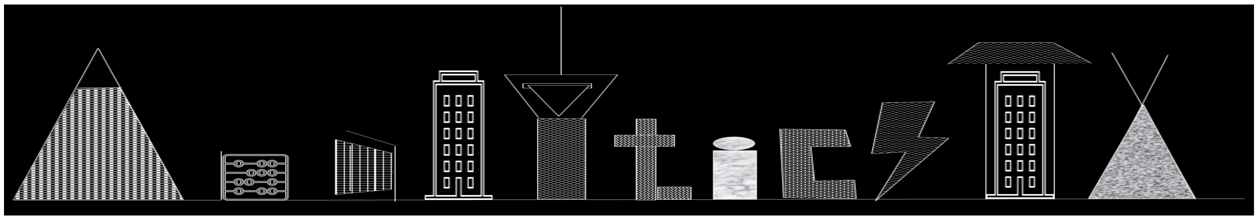
EXAMPLE BY

ANALYTICS TX, LLC

Recreating the Output of the "margins" Function from STATA to Python

Dr. Kruti Lehenbauer

4/2/2023



11703 Huebner Rd.| Ste 106-451| San Antonio, TX 78230 | Ph. 972-292-8338| ANALYTICSTX.COM

The “margins” function from STATA presented in Python

For this illustration of defining and applying the “margins” function of STATA to regression outputs in Python, I am using the dataset “breast_cancer” from sklearn.datasets. A more detailed example of how to create logistic regression using this dataset is available at: https://data.library.virginia.edu/logistic-regression-four-ways-with-python/ and I used Method 1 (statsmodels.formulas.api.Logit()) to create the Logit Model and the corresponding OLS formula for the Linear Regression Model discussed below.

Start by importing pandas, numpy, statsmodels.formula.api and importing the load_breast_cancer dataset from sklearn.datasets. Note that you can import any data using your standard import tools. While this illustration uses the Breast Cancer dataset, the actual Margins Function defined below can be used in any dataset as long as it is called correctly with the four parameters listed in the function. Note that I have skipped some output information in the code below for the sake of brevity.

```
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
from sklearn.datasets import load_breast_cancer
# Loading the data
cancer1 = load_breast_cancer()
print("Predictors: ", cancer1.feature_names)
print("\nResponse: ", cancer1.target_names)
```

```
Predictors:  ['mean radius' 'mean texture' 'mean perimeter' 'mean area' 'mean
smoothness' 'mean compactness' 'mean concavity' 'mean concave points' 'mean
symmetry' 'mean fractal dimension' 'radius error' 'texture error' 'perimeter
error' 'area error' 'smoothness error' 'compactness error' 'concavity error'
'concave points error' 'symmetry error' 'fractal dimension error' 'worst
radius' 'worst texture' 'worst perimeter' 'worst area' 'worst smoothness'
'worst compactness' 'worst concavity' 'worst concave points' 'worst symmetry'
'worst fractal dimension']

Response:  ['malignant' 'benign']
```

```
# Creating the dataframe
cancer = pd.DataFrame(cancer1.data, columns=cancer1.feature_names)
cancer.columns = cancer.columns.str.replace(' ','_')
cancer.shape #Omitted output below
# Add a column for response variable: malignant or benign
cancer['Target']=cancer1.target
```

This dataset does not have any explicit dummy functions, so we will create a dummy based on the “worst_texture” variable. There is no statistical importance of choosing that variable; this is being done to demonstrate the use of the margins function, so please take that into account.

```
# Output of this cell has been omitted but note:
# The dummy variable ‘flagvar’ has a mean of 0.530756
print(cancer.worst_texture.describe())
cancer['flagvar']= np.where(cancer.worst_texture>25,1,0)
print(cancer.flagvar.describe())
# Export csv file for use in STATA to compare results
cancer.to_csv("cancer.csv")
```

Now, we will define the "Margins" function using four main attributes: the name of the dataset (in this example, it would refer to cancer), the dummy variable of interest ("flagvar" in this example), the dependent variable in the regression model ("Target" in this example), and the result of the regression model (this will be obtained after defining the marginsat0_1 function).

```python
# Define the Margins Function:

def marginsat0_1(dataname, dummy, y, result):
    testflag0 = dataname.copy()
    testflag0[dummy]=0
    testflag0['prob0'] = result.predict(testflag0)
    mean0 = round(testflag0.prob0.mean(),5)
    testflag1 = dataname.copy()
    testflag1[dummy]=1
    testflag1['prob1'] = result.predict(testflag1)
    mean1 = round(testflag1.prob1.mean(),5)
    diff = round((mean1-mean0),5)
    print("Pr(" + str(y) + ")|"+ str(dummy) + "=0 : {}".format(mean0),
          "Pr(" + str(y) + ")|"+ str(dummy) + "=1 : {}".format(mean1),
          "Difference: {}".format(diff), sep="\n")
```

For this example, we will simply select the first 10 columns of the DataFrame that will be predictors in the models. Again, there is no statistical reason for picking these. This model is for illustrative purposes, only. Note that we are estimating both Logit and OLS models below and obtaining the Predicted Values of the Target variable for values of "flagvar" being equal to 1 and 0.

```python
# Select the first 10 columns of the DataFrame as predictors in the models
# Create the formula string

all_columns = ' + '.join(cancer.columns[:10])
formula = "Target ~ "+ all_columns + " + " 'flagvar'
print("Formula: ", formula, "\n")
```

```
Formula:  Target ~ mean_radius + mean_texture + mean_perimeter + mean_area +
mean_smoothness + mean_compactness + mean_concavity + mean_concave_points +
mean_symmetry + mean_fractal_dimension + flagvar
```

```python
# Estimate the Logit model (Output from the LOGIT model omitted below):

log_reg = smf.logit(formula, data=cancer).fit()
log_reg.summary()

# Using the Margins Function
marginsat0_1(cancer, 'flagvar', 'Target', log_reg)
```

```
Pr(Target)|flagvar=0 : 0.68893
Pr(Target)|flagvar=1 : 0.58749
Difference: -0.10144
```

For Logit models, the Pr(Target) refers to the probability of Target being equal to 1 (malignant breast cancer), given that the value of "flagvar" is equal to 0 in the first line of the output. The second line refers to the probability of Target being equal to 1 (malignant breast cancer) given that the value of "flagvar" is equal to 1. The difference being negative indicates that the probability of malignancy is lower for the value of "flagvar" being 1 as compared to being 0. **Important note:** Logit models typically have a binary dependent variable (which is the case in this example). Thus, the Probability values will fall within the range of 0 and 1.

```
# Build the OLS model & apply Margins function (Output from the Regression
model omitted for brevity):
lin_reg = smf.ols(formula, data=cancer).fit()
lin_reg.summary()
marginsat0_1(cancer, 'flagvar', 'Target', lin_reg)
```

```
Pr(Target)|flagvar=0 : 0.71026
Pr(Target)|flagvar=1 : 0.55418
Difference: -0.15608
```

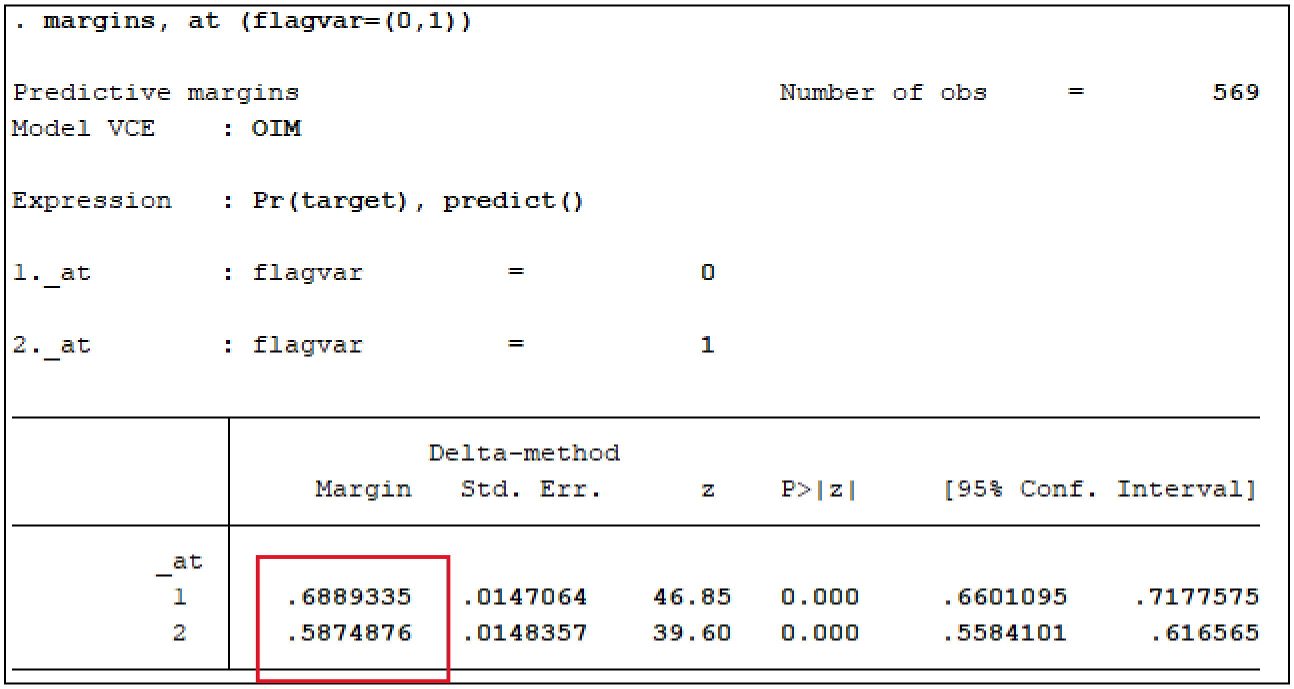
For OLS models, the Pr(Target) refers to the predicted value of the Target, given that the value of "flagvar" is equal to 0 in the first line of the output. The second line refers to the predicted value of the Target given that the value of "flagvar" is equal to 1. The difference being negative indicates that the predicted value of the Target is lower for the value of "flagvar" being 1 as compared to being 0.

Important note: Typically OLS models do not have a binary dependent variable (which is the case in this example). Thus, the Predicted values will fall within the range of the values that the dependent variable takes and are not restricted to being between 0 and 1.

The Outputs from STATA for this model are shared below. Note that we have not indicated in either the Python or the STATA model that our dummy variable is infact a categorical variable. The estimates are based upon the assumption that "flagvar" is a continuous variable that can take any value between 0 and 1. Compare the circled values with our Python results given above!

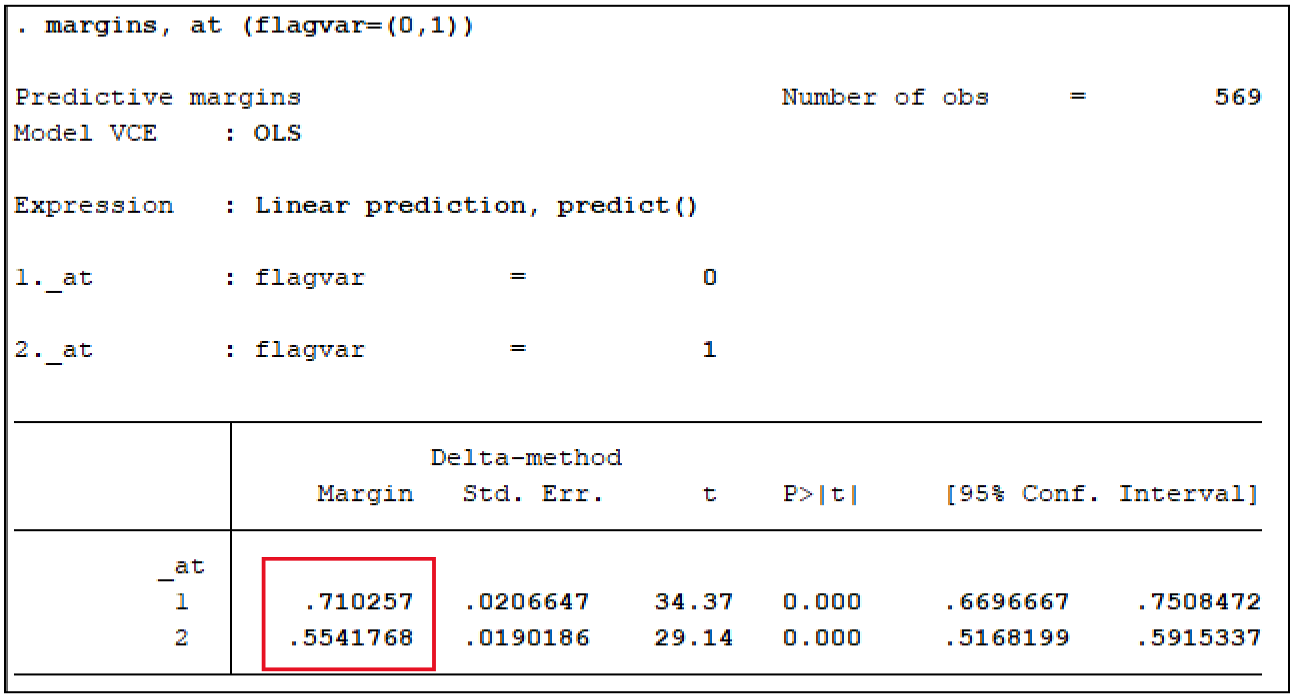
Corresponding Outputs from STATA

LOGIT REGRESSION STATA OUTPUT for margins function:



```
. margins, at (flagvar=(0,1))

Predictive margins                              Number of obs     =        569
Model VCE    : OIM

Expression   : Pr(target), predict()

1._at        : flagvar         =           0

2._at        : flagvar         =           1

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         _at |
          1  |   .6889335   .0147064    46.85   0.000     .6601095    .7177575
          2  |   .5874876   .0148357    39.60   0.000     .5584101     .616565
------------------------------------------------------------------------------
```

ORDINARY LEAST SQUARES REGRESSION STATA OUTPUT for margins function:



```
. margins, at (flagvar=(0,1))

Predictive margins                              Number of obs     =        569
Model VCE    : OLS

Expression   : Linear prediction, predict()

1._at        : flagvar         =           0

2._at        : flagvar         =           1

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         _at |
          1  |    .710257   .0206647    34.37   0.000     .6696667    .7508472
          2  |   .5541768   .0190186    29.14   0.000     .5168199    .5915337
------------------------------------------------------------------------------
```