**Applying linear models**

In the lesson, you reviewed linear models and how they can "break" when certain assumptions are not met. For example, one assumption is that the response variable is continuous and that a Gaussian distribution for the response can be applied for the model formulation. In case of a binary response variable, the response distribution is Binomial providing for the estimated yy to be bounded by 0 and 1.

Using the information provided in the lesson, which of the following statements is **TRUE**?

**Answer the question**

**50 XP**

**Possible Answers**

The linear model can only be used if all the explanatory variables are continuous.

Suppose that the response variable for a study is treated continuously. Then a GLM with the binomial probability distribution should be used for the analysis.

The estimated probabilities from the Binomial GLM is between 0 and 1.

The estimated probabilities from the linear model are between 0 and 1 if the response is binary.

Take Hint (-15xp)

**Incorrect Submission**

Recall what is the range of model predictions when a linear model is fitted to the data with a binary response.

Did you find this hint helpful?

YesNo

**Linear model, a special case of GLM**

In this exercise you will fit a linear model two ways, one using the ols() function and one using the glm() function. This will show how a linear model is a special case of a generalized linear model (GLM).

You will use the preloaded salary dataset introduced in the video.

Recall that the linear model in Python is defined as:

ols(formula = 'y ~ X', data = my\_data).fit()

and the generalized linear model can be trained using

glm(formula = 'y ~ X', data = my\_data, family = sm.families.\_\_\_).fit()

**Instructions 1/2**

**50 XP**

* [2](javascript:void(0))
* Import the statsmodels.api with the common alias sm, and the ols and glm modules from the statsmodels.formula.api.
* Fit a linear model by predicting Salary with Experience using the salary dataset.

import statsmodels.api as sm

from statsmodels.formula.api import ols, glm

# Fit a linear model

model\_lm = ols(formula = 'Salary ~ Experience',

data = salary).fit()

# View model coefficients

print(model\_lm.params)

Intercept 25792.200199

Experience 9449.962321

dtype: float64

<script.py> output:

Intercept 25792.200199

Experience 9449.962321

dtype: float64

Fit a GLM using the same formula and data, as for the linear model, but this time include the Gaussian() family as additional input.

from statsmodels.formula.api import ols, glm

import statsmodels.api as sm

# Fit a GLM

model\_glm = glm(formula = '\_\_\_\_ ~ \_\_\_\_',

data = \_\_\_\_,

family = sm.families.\_\_\_\_).fit()

# View model coefficients

print(model\_glm.params)