Final Project SAheart

Group 28

May 16, 2024

1 Fisher Scoring Algorithm for Probit Regression

Probit regression is an alternative Generalized Linear Model for binary data. Given p covariates and n observations, the model reads:

- **Random Component:** $Y_i \sim \text{Bernoulli}(\mu_i), \quad i = 1, \ldots, n.$
- **Systematic Component:** $\eta_i = X_i^\top \beta.$
- **Link Function:** $\Phi^{-1}(\mu_i) = \eta_i$, where Φ is the cumulative distribution of a standard Gaussian distribution.

As the cumulative distribution function (CDF) Φ of the standard normal Gaussian distribution has no explicit form, we first decided to estimate it with Monte Carlo methods.
We will then derive the equations and update the equations for the parameters of the Fisher Scoring.

1.1 Monte Carlo estimation of the Standard Normal' CDF

We developed an algorithm to compute the empirical cumulative distribution of the standard normal distribution, namely Φ. To do so, we generated a sample of size 100 million realizations from a standard normal distribution $\mathcal{N}(0, 1)$. We then sorted the sample and computed the cumulative probabilities of such. We saved the results in two arrays: y_{values} and x_{values} to avoid running the simulations again in the future.
Then, we wrote a function that would map a value (scalar or vector) x to its value of the empirical CDF.

1.2 Parameters of Fisher Scoring GLM and convergence

The implementation of a Fisher Scoring algorithm for GLM requires the derivation of the following quantities for the probit regression:

- **Probability of success:** μ
  From the link function: $\mu = \Phi(\eta)$
- **Variance of Y:** $Var(Y_i)$
  As the dependent variable is binary: $Var(Y) = \mu \cdot (1 - \mu) + \epsilon$
  where ϵ is a small constant to avoid division by 0 errors
- **Diagonal matrix of weights:** W
  $W = \text{Diag}\left[\frac{1}{\text{Var}(y_i)}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2\right]$, from the general formula
  As μ is the CDF of the standard normal, its derivative with respect to η is the PDF of

the standard normal: $\left(\frac{\partial \mu_i}{\partial \eta_i}\right) = \phi(\eta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}}$

Therefore: $W = \text{Diag}\left[\frac{1}{\text{Var}(y)}(\phi(\eta)^2\right] = \text{Diag}\left[\frac{1}{\mu \cdot (1-\mu)+\epsilon}\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}}\right)^2\right]$

- **Working variate:** Z
  $Z = \eta + \frac{\partial \eta}{\partial \mu}(y - \mu)$, from the general formula

  Therefore: $Z = \eta + \phi(\eta)^{-1}(y - \mu) = \eta + (\frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}})^{-1}(y - \mu)$

- **Update of the regressors:**
  $\beta_{t+1} = (X^\top W X)^{-1} X^\top W Z$, from the general formula

1.3 Description of Probit Fisher Scoring Algorithm

Initialization: We defined a function `probit_glm` which takes input features X and binary response Y. Optionally, an intercept column is added to the feature matrix X if `fit_intercept` is set to `True` (default) and `epsilon` has a default value $\epsilon = 1e^{-6}$.
We set the initial values of the algorithm including the coefficients b_0 (a vector of 0), linear predictor (η), and probabilities (μ) based on the cumulative normal distribution function (`normalCDF`). The initial variance (`var`), weights (`W`) and working variate (`Z`) are calculated according to the formulas derived in the section above.

Iterative Optimization: The algorithm iteratively updates the parameters until convergence is reached. In each iteration, the linear system

$$X^\top W X \beta - X^\top W Z = 0$$

is solved using the Fisher scoring method, where W is the diagonal weight matrix, Z is the working variate, and β is the coefficient vector. This yields the new coefficient vector (b).
Updated values of the linear predictor (η), probabilities (μ), variance (var), weights (W) and working variate (Z) are computed.
The marginal updates of the coefficients are stored in a list for latter diagnostics.

Convergence Monitoring and Reporting: The algorithm monitors convergence by calculating the norm of the difference between consecutive coefficient estimates:

$$\frac{\|b - b_0\|}{\|b_0\| + \epsilon} < \epsilon$$

While convergence is not achieved, the algorithm continues to compute the coefficient vector, storing the values of the previous step in (b_0).
If convergence is achieved, the algorithm reports the final results, including the estimated coefficients, marginal updates, standard errors which are calculated based on the inverse of the Fisher information matrix

$$\text{cov} = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1}, \quad \text{se} = \sqrt{\text{diag}(\text{cov})},$$

number of iterations, and computation time.

2 Bayesian Probit Regression and MCMC

The Bayesian binary probit regression model can be specified as follows:

$$Y_i \sim \text{Bernoulli}(\Phi(\eta_i)), \quad \eta_i = x_i\beta, \quad \beta \sim \pi(\beta)$$

where $\Phi(\cdot)$ denotes the cumulative distribution function (CDF) of the standard normal distribution. β represents a $(D \times 1)$ column vector of regression coefficients, $\pi(\beta)$ is a prior distribution over the model parameters. Finally, η_i denotes the linear predictor given by the inner product of the covariates x_i and the parameter vector β. The posterior distribution for the Bayesian binary probit model is the following,

$$p(\beta|y, X) \propto p(\beta)p(y|\beta, X) = \pi(\beta)\prod_{i=1}^{N} p(y_i|\beta, x_i)$$

$$= \pi(\beta)\prod_{i=1}^{N} \Phi(x_i\beta)^{y_i}(1 - \Phi(x_i\beta))^{(1-y_i)}$$

Performing inference for this model in the Bayesian framework is complicated by the fact that no conjugate prior $\pi(\beta)$ exists for the parameters of the probit regression model. To overcome this problem, Albert and Chib (1993) augmented the original model with an additional auxiliary variable that renders the conditional distributions of the model parameters equivalent to those under a Bayesian normal linear regression model with Gaussian noise; and derived an efficient Gibbs sampling scheme for computing the posterior statistics.
Let us introduce N independent latent variables z_i, where each z_i follows a normal distribution. Then the augmented probit model has the following hierarchical structure, where,

$$y_i = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{if } z_i \leq 0 \end{cases}$$

$$z_i = x_i\beta + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 1), \quad \beta \sim \pi(\beta)$$

where y_i is now deterministic conditional on the sign of the stochastic auxiliary variable z_i. Therefore, we formulate the original problem as a missing data problem where we have a normal regression model on the latent data z_i, and the observed responses y_i are incomplete in that we only observe whether $z_i > 0$ or $z_i \leq 0$.
We are interested in computing the joint posterior distribution of the latent variables z and the model parameters β given the data y and X,

$$p(z, \beta|y, X) \propto p(\beta)p(z|\beta, X)p(y|z) = \pi(\beta)\prod_{i=1}^{N} p(z_i|\beta, x_i)p(y_i|z_i)$$

where we have that,

$$p(z_i|\beta, x_i) = \mathcal{N}(z_i|x_i\beta, 1)$$

$$p(y_i|z_i) = 1(y_i = 1) \cdot 1(z_i > 0) + 1(y_i = 0) \cdot 1(z_i \leq 0)$$

where $1(\cdot)$ is the indicator function, equal to 1 if the quantities inside the function are satisfied, and 0 otherwise.

2.1 Gibbs Sampling Framework

This joint posterior is difficult to normalize and sample from directly. However, computation of the marginal posterior of β and z using Gibbs sampling requires only computing $p(\beta|z, y, X)$ and $p(z|\beta, y, X)$, and these full conditional distributions are of standard forms. It should be noted that $p(\beta|z, y, X) = p(\beta|z, X)$, since β is conditionally independent of y given z.
First, the full conditional of β is given by,

$$p(\beta|z, X) \propto \pi(\beta) \prod_{i=1}^{N} \mathcal{N}(z_i|x_i\beta, 1)$$

This quantity is the posterior density for the normal linear regression model. Using standard linear model results, if we use a constant prior for β, i.e. $\pi(\beta) \propto 1$, then,

$$\beta|z, X \sim \mathcal{N}\left((X^TX)^{-1}X^Tz, (X^TX)^{-1}\right)$$

However, if we assign the proper conjugate prior $\pi(\beta) = \mathcal{N}(\beta_0, Q_0)$, then,

$$\begin{aligned} \beta|z, X &\sim \mathcal{N}(M, V) \\ M &= V(Q_0^{-1}\beta_0 + X^Tz) \\ V &= (Q_0^{-1} + X^TX)^{-1} \end{aligned}$$

Now suppose we knew β. Then we could easily draw the latent variables z_i from their distribution conditional on β and x_i, which is $\mathcal{N}(x_i\beta, 1)$. However, if we also condition on the y_i, we need to take into consideration this additional source of information. Thus, the full conditional of $z_i|\beta, y_i, x_i$ is a truncated normal distribution,

$$z_i|\boldsymbol{\beta}, y_i, \boldsymbol{x}_i \sim \begin{cases} \mathcal{TN}(\boldsymbol{x}_i\boldsymbol{\beta}, 1, 0, \infty) & \text{if } y_i = 1 \\ \mathcal{TN}(\boldsymbol{x}_i\boldsymbol{\beta}, 1, -\infty, 0) & \text{if } y_i = 0 \end{cases}$$

More in detail, In each simulation step of the Gibbs algorithm, we have to sample $\beta|z, X$ and $z_i|\beta, y_i, x_i$ in turn.

2.2 Description of Our Gibbs Sampling Algorithm

Initialization of Parameters: Set the vector β to be a vector of zeros with the same length as the number of covariates, and initialize the latent variables z as a vector of zeros with a length equal to the number of observations in the dataset.

Gibbs Sampling Steps:

1. **Update z given β and y**: For each i, sample z_i from a truncated normal distribution:

$$z_i|\beta, y_i, x_i \sim \begin{cases} \mathcal{TN}(x_i^T\beta, 1, 0, \infty) & \text{if } y_i = 1 \\ \mathcal{TN}(x_i^T\beta, 1, -\infty, 0) & \text{if } y_i = 0 \end{cases}$$

2. **Update β given z and X**: The posterior distribution of β is a normal distribution:

$$\beta|z, X \sim \mathcal{N}(M, V)$$

where

$$\begin{aligned} V &= (Q_0^{-1} + X^TX)^{-1} \\ M &= V(Q_0^{-1}\beta_0 + X^Tz) \end{aligned}$$

Estimation and Convergence: After a predefined number of iterations and a burn-in period, use the samples of β to estimate its posterior mean. The final estimate of β is given by the average of the sampled values after the burn-in period.

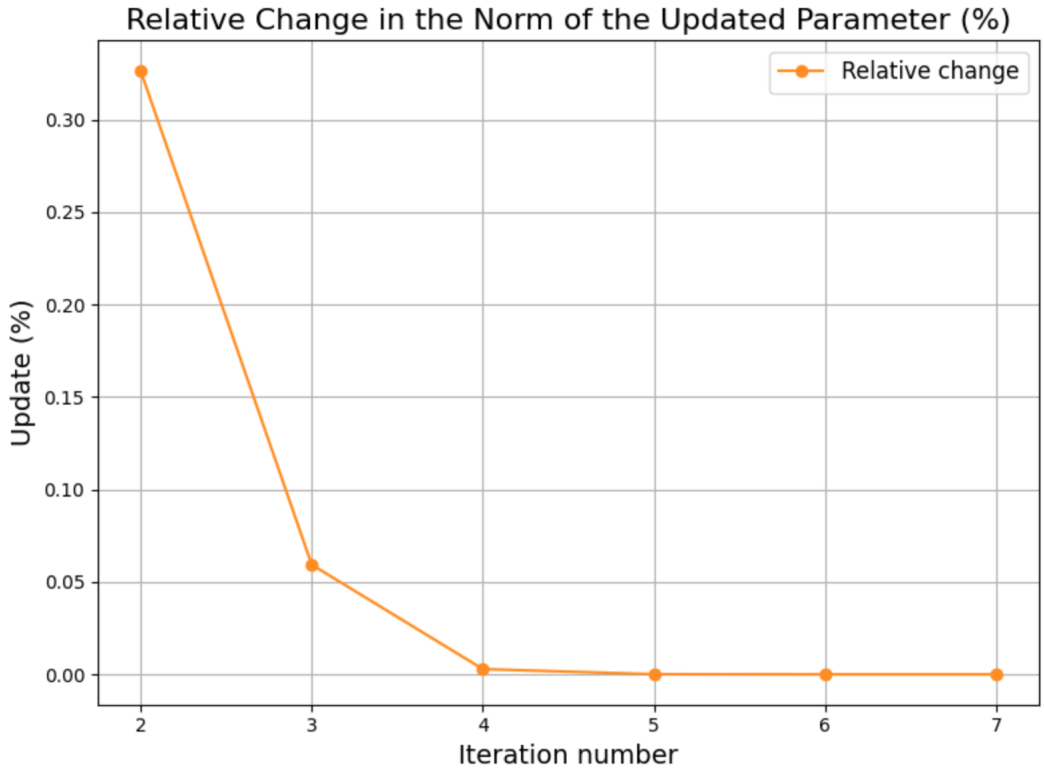
3 Convergence Diagnostics

3.1 Fisher Scoring

In order to assess the convergence of our Fisher scoring algorithm, we stored the percentage update of the parameter β for each iteration in a separate vector. For this algorithm, convergence is reached once the update is negligible up to a small ϵ, implying that the likelihood function is maximized (or very close to the maximum). Each element in the vector is constituted as follows:

$$deltaNorm = [\frac{\|b_1 - b_0\|}{\|b_0\| + \epsilon}, ..., \frac{\|b_{i+1} - b_i\|}{\|b_i\| + \epsilon}, ..., \frac{\|b_{maxiter} - b_{maxiter-1}\|}{\|b_{maxiter-1}\| + \epsilon}], \forall i = 1, ..., maxiter \quad (1)$$

We report the result in the following graph. Note that the first iteration is excluded from the plot due to the initialization of b_0, which is in fact a zero vector, meaning that the update is much larger than subsequent iterations since we have $0 + \epsilon$ at the denominator.

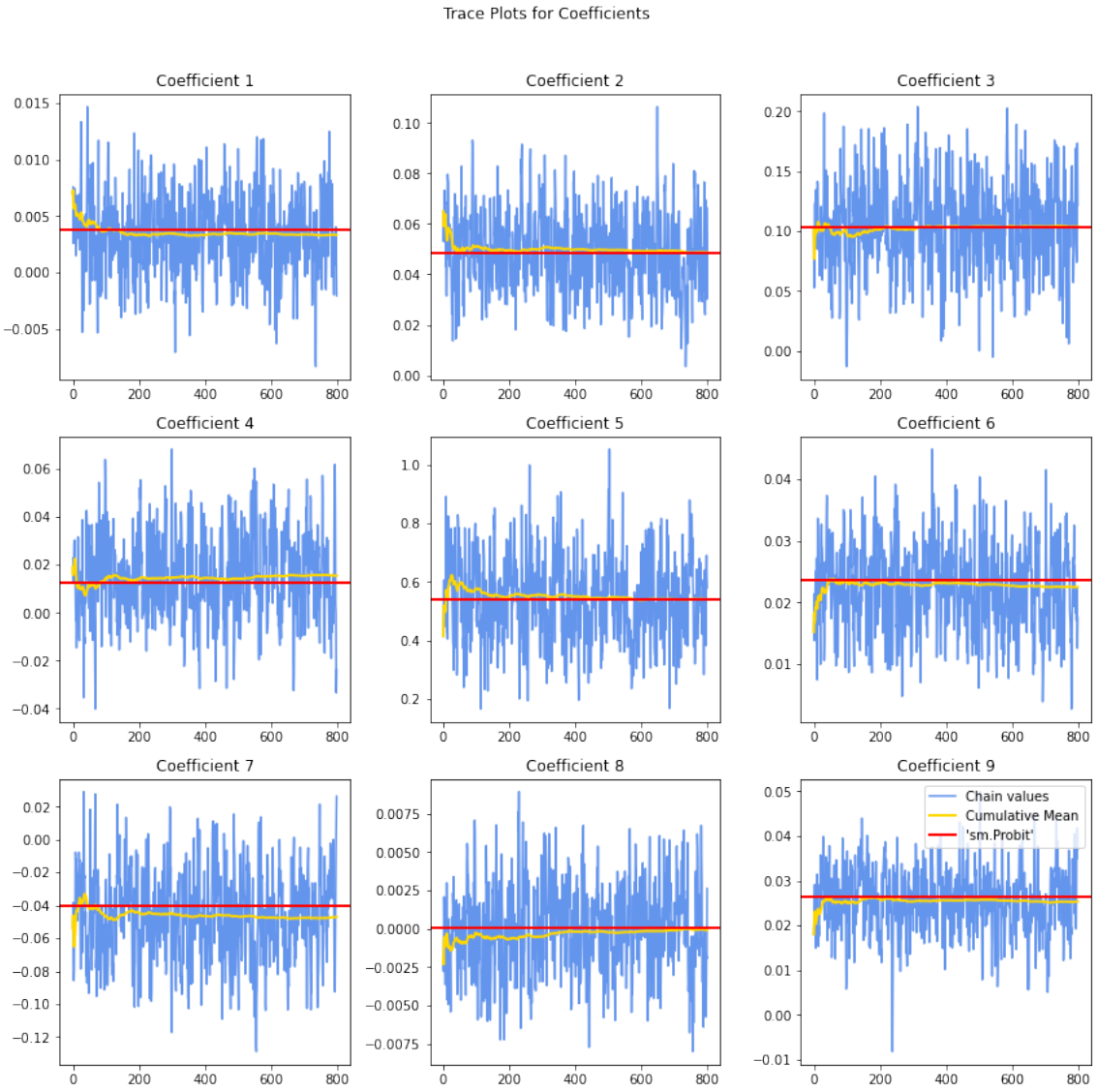


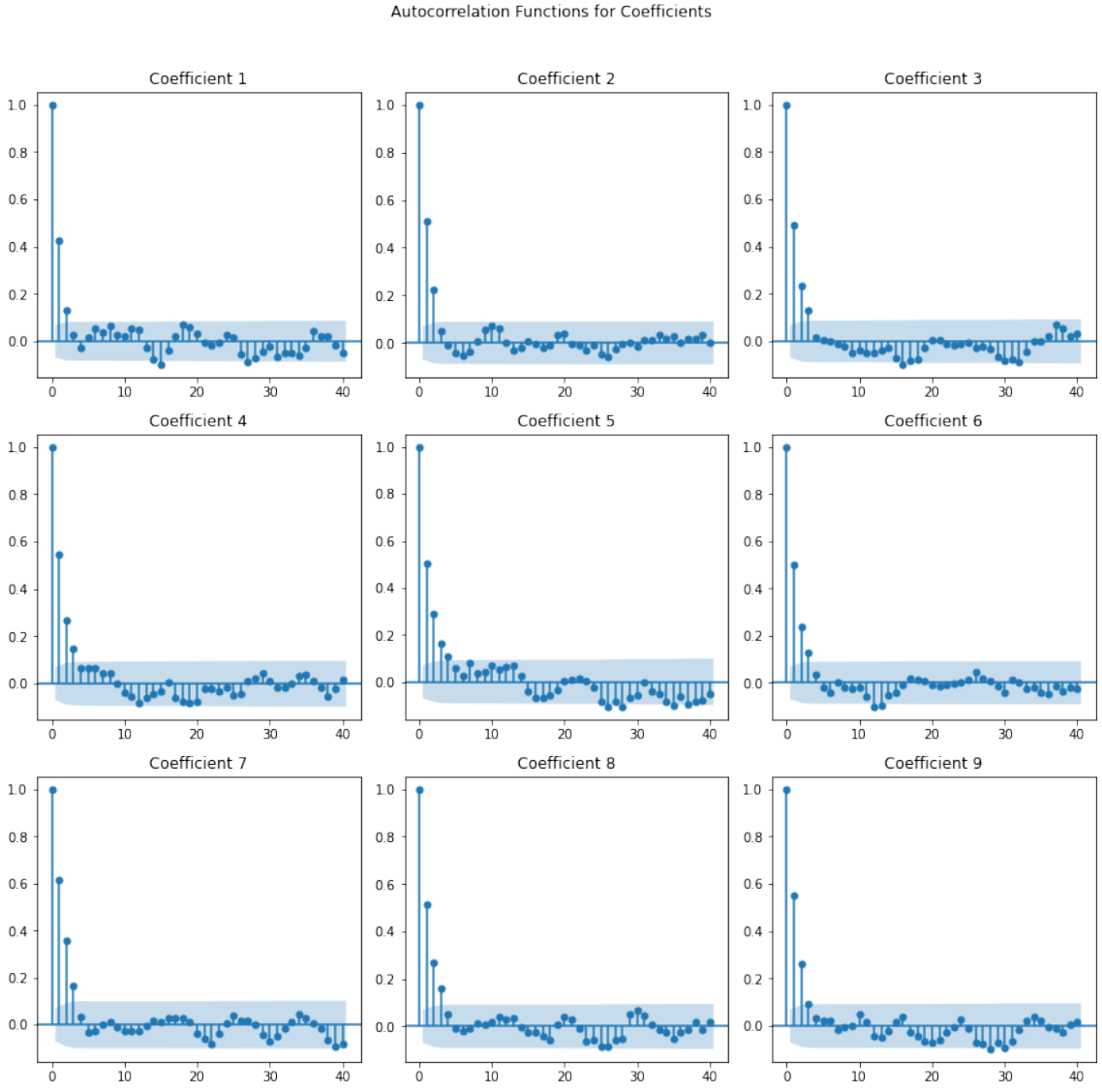
As we can see the algorithm converges quite fast as we expected. From the values stored by the algorithm, we see that after 4 iterations the algorithm reached an update of 0.3%, indicating a very fast convergence rate.

3.2 MCMC

As for the Gibbs sampling algorithm, we have stored the information of each MCMC iteration in the 'beta_chain' matrix, hence, we can now generate some useful plots to assess how the chain evolved over each iteration and what the marginal posterior density of each parameter looks like. The panel shows the trace plot of each parameter, i.e., the values the parameters took at each iteration of the simulation algorithm. As we observe, after a few hundred iterations, the

trace plots show a stable pattern, and the cumulative mean of the chain values converged to the benchmark values of the parameters ('smProbit' outputs).
In addition to this, all the autocorrelation functions converge to 0, indicating that subsequent samples become less and less correlated and that the chain does not get stuck in any region of the parameter space.
These diagnostics suggest that the algorithm reaches a stationary state, and the chain values are representative of the target distribution, providing confidence in the strength of the results.




Autocorrelation Functions for Coefficients
Coefficient 1
Coefficient 2
Coefficient 3
Coefficient 4
Coefficient 5
Coefficient 6
Coefficient 7
Coefficient 8
Coefficient 9

4 Model Interpretation and Final Discussion

Interpret the fitted models, particularly focusing on the role of cholesterol. Discuss the implications and insights obtained from the analysis.
Through the previously described algorithms, we were able to obtain probit estimates for the role of cholesterol level in explaining the occurrence of coronary heart disease. In particular, we define the variable *chd* as our vector Y, and the rest of the dataset as our matrix X. The main covariate of interest (cholesterol level) is *ldl*, while the other variables included in the dataset are treated as controls. We report the results in the following table:

Probit estimation:

		Coefficients	Z	Standard Errors	P-Values	[0.025	0.975]
const		-3.571	-4.749	0.752	0.000	-5.044	-2.097
Beta_1	sbp	0.004	1.106	0.003	0.269	-0.003	0.011
Beta_2	tobacco	0.048	3.044	0.016	0.002	0.017	0.079
Beta_3	ldl	0.103	2.914	0.035	0.004	0.034	0.172
Beta_4	adiposity	0.012	0.713	0.017	0.476	-0.022	0.046
Beta_5	famhist	0.539	3.998	0.135	0.000	0.275	0.803
Beta_6	typea	0.024	3.277	0.007	0.001	0.009	0.038
Beta_7	obesity	-0.040	-1.528	0.026	0.127	-0.092	0.011
Beta_8	alcohol	0.000	0.007	0.003	0.994	-0.005	0.005
Beta_9	age	0.026	3.733	0.007	0.000	0.012	0.040

As we can see the coefficient for "ldl" is positive and statistically significant at the 5% level, with a p-value of 0.035. Thus, we can infer that there is indeed a sizeable positive correlation between higher cholesterol levels and the probability of suffering from coronary heart disease; however, we cannot interpret these coefficients as marginal effects on $P[Y = 1|X]$; in fact, the marginal effect we seek is not constant and depends on the covariates' levels. In order to gain some insight as to the true magnitude of this effect, we refer to the library **statsmodels**, through which we obtain the following table:

Probit Marginal Effects

Dep. Variable:	y					
Method:	dydx					
At:	overall					
	dy/dx	std err	z	P>|z|	[0.025	0.975]
x1	0.0011	0.001	1.108	0.268	-0.001	0.003
x2	0.0139	0.004	3.120	0.002	0.005	0.023
x3	0.0297	0.010	3.004	0.003	0.010	0.049
x4	0.0036	0.005	0.715	0.475	-0.006	0.013
x5	0.1554	0.037	4.188	0.000	0.083	0.228
x6	0.0068	0.002	3.375	0.001	0.003	0.011
x7	-0.0116	0.007	-1.562	0.118	-0.026	0.003
x8	5.64e-06	0.001	0.007	0.994	-0.002	0.002
x9	0.0076	0.002	3.870	0.000	0.004	0.011

By looking at the coefficient for cholesterol (x_3), we can see that the average marginal effect is 0.0297, meaning that on average, for a unit increase in cholesterol level, the probability of suffering from coronary heart disease will increase by around 3 percentage points. Moreover, having a family history involving cases of coronary heart disease, will lead to a 15.5 percentage point higher probability of said disease. Note that these effects embody a simple correlation rather than a causal relationship, as we do not possess information regarding the sampling procedure and other demographic controls could affect the outcome we are studying.