DSB205, Winter 2025
Problem Set 2: Ridge Regression, Logistic Regression, Gradient Descent and Newton's Method
Due Feb 14, 2025 at 11:59 pm PST

Submission instructions

- Submit your solutions electronically on the course Gradescope site as PDF files.
- If you plan to typeset your solutions, please use the LaTeX solution template. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app.

In this homework, we will explore questions related to regression and optimization.

1 Ridge regression [10 pts]

Consider ridge regression where we have n pairs of inputs and outputs, $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^m$. The outputs are centered so we don't need a bias term in our regression. We have shown in class that the ridge regression estimator is given by $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I}_m)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$ where $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_n]^{\mathrm{T}}$ is a $n \times m$ matrix.

(a) Given genotype matrix $\boldsymbol{X}$ and vector of phenotypes $\boldsymbol{y}$, we would like to obtain a linear predictor from $\boldsymbol{X}$ to $\boldsymbol{y}$. In a typical GWAS study, why would ridge regression be preferable over linear regression to accomplish this?

In a typical GWAS study, Ridge Regression is preferred over Linear Regression because our design matrix X will not be invertible. This is because we have so many more SNPs (columns) than data from individuals (rows).
If X is a $n \times m$ matrix, X^TX results in a $m \times m$ matrix. However, it is rank limited to n in GWAS-like scenarios. Since it isn't full rank, it isn't invertible. In ridge regression, we add λI to the matrix that needs to be inverted.

(b) Show that $(\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I}_m)$ is always invertible for $\lambda > 0$.

A matrix X is invertible if it is square and $det(X) \neq 0$. The determinant of a matrix is the product of all it's eigenvalues. If any eigenvalue $\lambda = 0$, the determinant would be 0, and the matrix wouldn't be invertible. When adding the λI term, where $\lambda > 0$, all eigenvalues are shifted by the λ term. Thus, there would be no eigenvalues = 0, and the matrix would be invertible. Mathematically, this would be:

$$(X^TX)v = \lambda_i v$$

Above is the property of eigenvalues, which we will sub in below.

$$(X^TX + \lambda I)v = X^TXv + \lambda Iv$$

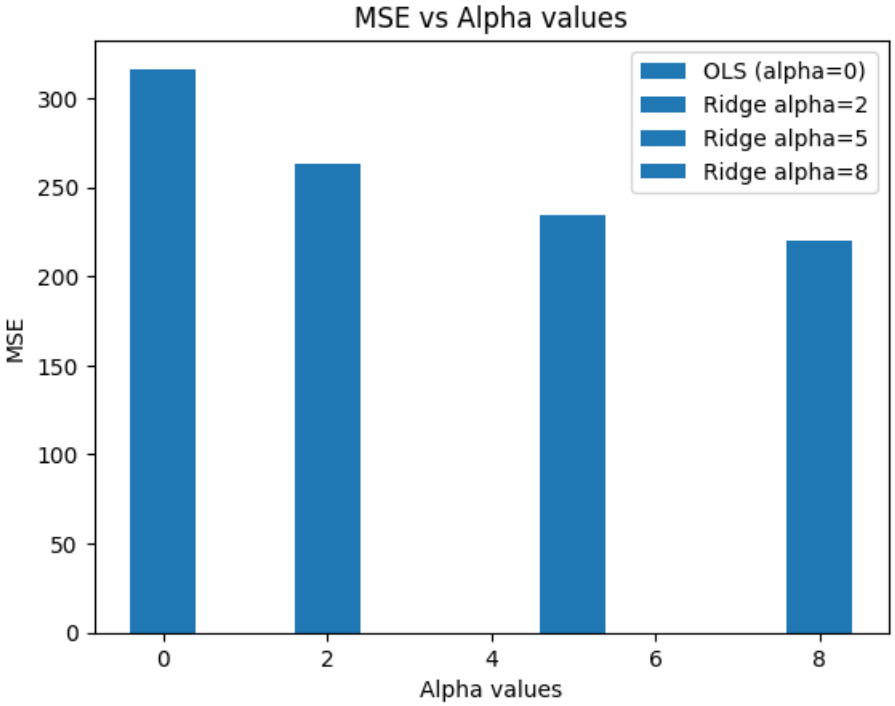
$$(X^TX + \lambda I)v = \lambda_i v + \lambda Iv$$

Using the property that multiplying any vector by the identity matrix $Iv = v$:

$$(X^TX + \lambda I)v = \lambda_i v + \lambda v = (\lambda_i + \lambda)v$$

Our eigenvalues have shifted positively by the λ from regularization. Since the Gram matrix X^TX is positive semidefinite, it has no negative eigenvalues. Therefore, all eigenvalues will now be positive, and the matrix would have a non-zero determinant.

(c) You are given the genotypes $\boldsymbol{X}$ (Q1.training/test.geno) and phenotypes $\boldsymbol{y}$ (Q1.training/test.pheno) of 1000 individuals. Implement ridge regression. Plot the mean squared error (MSE) against different parameter settings of $\lambda = 2, 5, 8$. How does ridge regression compare to the ordinary least squares (OLS) estimator?



As we can see, Ridge Regression is more accurate than the OLS estimator, at least in terms of its Mean Squared Error. It appears that the within the range of {2,8}, the higher the λ or alpha term, the more accurate the estimator is. Note that OLS shouldn't be possible since the matrix isn't invertible. Sklearn's wrapper of the Scipy OLS method most likely uses some form of Pseudoinverse method or Singular Value Decomposition (SVD). When coding out the OLS solution and using the inverse from SKLearn library, the MSE comes out to a very large number (roughly 10 million).

2 Logistic regression, Gradient descent and Newton's method [10 pts]

Given training data of input, output pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \{0, 1\}$, consider a logistic regression model relating y and $\boldsymbol{x}$: $\Pr(y = 1|\boldsymbol{x}) = \sigma(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}_i + \beta_0)$.

In this problem, you will analyze and implement logistic regression for binary classification problems using two different types of optimization approaches for minimizing the negative log likelihood : gradient descent and Newton's method. You are given genotypes and binary phenotype data and would like to fit a logistic regression model relating all SNPs to phenotype. Files hw.2-1.geno consists of 1000 individuals at 10 SNPs and file hw.2-1.pheno contains a binary phenotype.

For this question, include any plots along with your answers (do not include code).

(a) Write the negative log likelihood $\mathcal{NLL}(\boldsymbol{\beta}, \beta_0)$.

$$NLL(\boldsymbol{\beta}, \beta_0) = -\sum_{i=1}^{n} log(\sigma(\boldsymbol{\beta}^T x_i + \beta_0)^{y_i}[1 - \sigma(\boldsymbol{\beta}^T x_i + \beta_0)]^{1-y_i})$$

$$= -\sum_{i=1}^{n} log\, \sigma(\boldsymbol{\beta}^T x_i + \beta_0)^{y_i} + log[1 - \sigma(\boldsymbol{\beta}^T x_i + \beta_0)]^{1-y_i}$$

$$= -\sum_{i=1}^{n} y_i\, log\, \sigma(\boldsymbol{\beta}^T x_i + \beta_0) + (1 - y_i)\, log[1 - \sigma(\boldsymbol{\beta}^T x_i + \beta_0)]$$

(b) Is $\mathcal{NLL}$ convex ? Show why or why not.

To do this, we need to first compute gradient, and then the Hessian.
Let's also augment the intercept into the vector of parameters.

$$\nabla_\beta NLL(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \nabla y_i \, log\, \sigma(\boldsymbol{\beta}^T x_i) + \nabla(1-y_i)\, log[1-\sigma(\boldsymbol{\beta}^T x_i)]$$

Taking gradient of first term:

$$\nabla y_i \, log\, \sigma(\boldsymbol{\beta}^T x_i) = y_i \nabla(log\, \sigma(\boldsymbol{\beta}^T x_i))$$

$$s = \sigma(w) \quad w = \boldsymbol{\beta}^T x_i$$

$$\nabla_\beta(log(s)) = \frac{1}{s}s' = \frac{1}{s}s[1-s]w' = \frac{1}{s}s[1-s]x_i$$

This simplifies to:

$$(1-s)x_i \implies (1-\sigma(\boldsymbol{\beta}^T x_i))x_i$$

Now for the second term:

$$\nabla(1-y_i)\, log(1-\sigma(\boldsymbol{\beta}^T x_i)) = (1-y_i)\nabla\, log(1-\sigma(\boldsymbol{\beta}^T x_i))$$

$$\nabla_\beta(log(1-s)) = \frac{1}{1-s}(1-s)' = \frac{1}{1-s} - s(1-s)w' = -sx_i$$

$$-sx_i \implies -\sigma(\boldsymbol{\beta}^T x_i)x_i$$

Our gradient comes out to:

$$\nabla_\beta NLL(\boldsymbol{\beta}) = -\sum_{i=1}^{n} y_i(1-\sigma(\boldsymbol{\beta}^T x_i))x_i - (1-y_i)\sigma(\boldsymbol{\beta}^T x_i)x_i$$

$$= -\sum_{i=1}^{n}[y_i - y_i\sigma(\boldsymbol{\beta}^T x_i)]x_i - [\sigma(\boldsymbol{\beta}^T x_i) - y_i\sigma(\boldsymbol{\beta}^T x_i)]x_i$$

$$\nabla_\beta NLL(\boldsymbol{\beta}) \implies -\sum_{i=1}^{n}[y_i - \sigma(\boldsymbol{\beta}^T x_i)]x_i$$

Computing the 2nd derivative aka Hessian of matrix:

$$\nabla_\beta^2 = -\sum_{i=1}^{n} \nabla[y_i - \sigma(\boldsymbol{\beta}^T x_i)]x_i = -\sum_{i=1}^{n} \nabla y_i x_i - \nabla\sigma(\boldsymbol{\beta}^T x_i)x_i$$

now computing derivatives of both terms:

$$\nabla_\beta\, y_i x_i = 0$$

$$\nabla_\beta \sigma(\boldsymbol{\beta}^T x_i)x_i = [\sigma(\boldsymbol{\beta}^T x_i)(1-\sigma(\boldsymbol{\beta}^T x_i))(\boldsymbol{\beta}^T x_i)']x_i = [\sigma(\boldsymbol{\beta}^T x_i)(1-\sigma(\boldsymbol{\beta}^T x_i))]x_i x_i^T$$

$$\nabla_\beta^2 = -\sum_{i=1}^{n} -[\sigma(\boldsymbol{\beta}^T x_i)(1-\sigma(\boldsymbol{\beta}^T x_i))]x_i x_i^T$$

$$\nabla_\beta^2 = \sum_{i=1}^{n} [\sigma(\boldsymbol{\beta}^T x_i)(1-\sigma(\boldsymbol{\beta}^T x_i))]x_i x_i^T$$

now let's multiply hessian by any vector v, to check if its positive semidefinite:

$$v^T\nabla^2 v \geq 0 \implies v^T \sum_{i=1}^{n} [\sigma(\boldsymbol{\beta}^T x_i)(1-\sigma(\boldsymbol{\beta}^T x_i))]x_i x_i^T v$$

Both sigmoid functions output a number from 0 to 1, so their summation should be positive. Now, we just need to make sure that multiplying the vectors with x is positive:

$$v^T x_i x_i^T v = (v^T x_i)(x_i^T v) \implies (v^T x_i)^2$$

Multiplying the hessian by vectors v results in a number greater than or equal to 0. Thus, the logistic function is convex.

Optimization: Use the following specifications for the optimization algorithms.

- *Stopping criterion*: Run both algorithms for 100 iterations.
- *Step size*: For gradient descent, we will consider fixed step sizes.
- *Initialization*: Set all parameters to 0.

Here are the parameter updates for gradient descent (for a fixed step size η) and Newton's method.

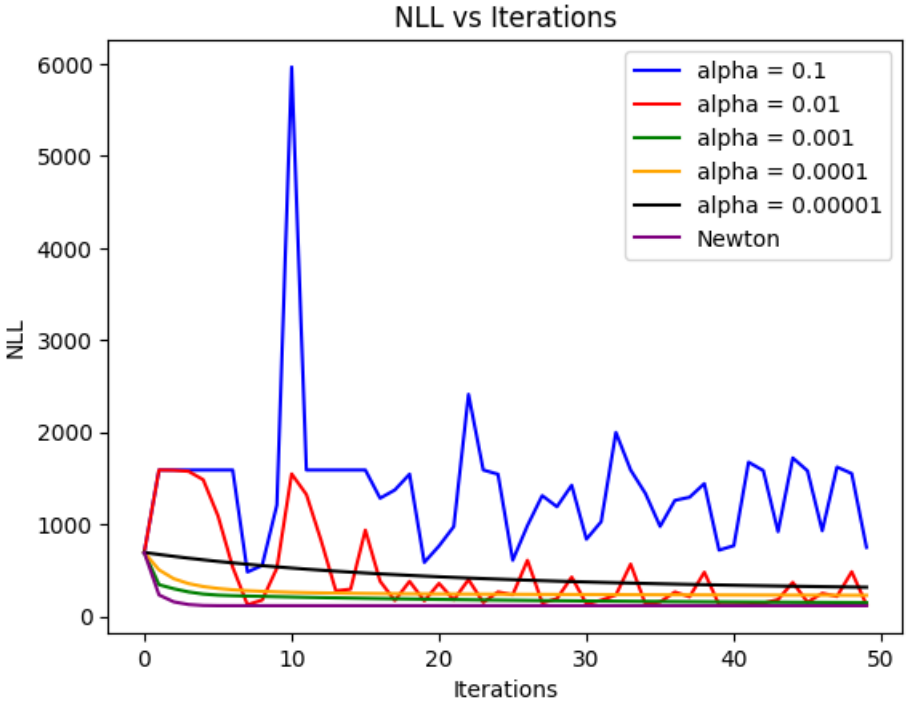
For gradient descent

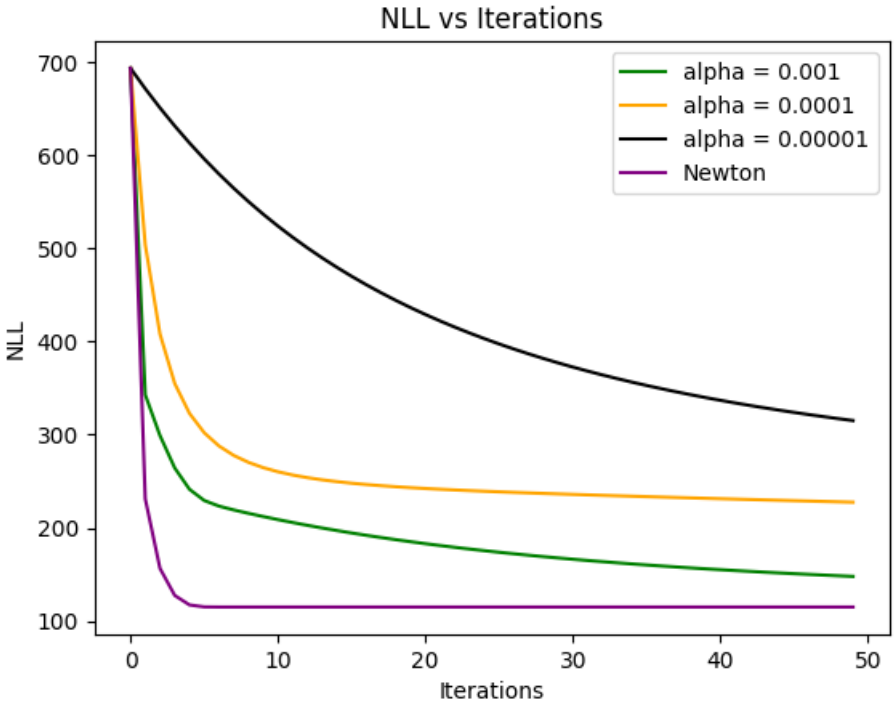
$$\boldsymbol{\beta}^{(t+1)} \leftarrow \boldsymbol{\beta}^{(t)} - \eta \boldsymbol{g}(\boldsymbol{\beta}^{(t)}) \tag{1}$$

For Newton's method

$$\boldsymbol{\beta}^{(t+1)} \leftarrow \boldsymbol{\beta}^{(t)} - \boldsymbol{H}(\boldsymbol{\beta}^{(t)})^{-1} \boldsymbol{g}(\boldsymbol{\beta}^{(t)}) \tag{2}$$

(c) For step sizes $\eta = \{1 \times 10^{-i} : i = 1, 2, 3, 4, 5\}$ for gradient descent, plot the $\mathcal{NLL}$ as a function of iteration $t \in \{1, \ldots, 50\}$. Also plot $\mathcal{NLL}$ as a function of t for Newton's method.





The first graph may be hard to read due to the large step-sizes that overshoot the descent. Here's an upclose of the optimization methods that perform a better job at approaching a low NLL.

(d) What are the β estimates for SNPs 1 and 7 from Newton's method?

When augmenting the matrix (so that we can get an intercept term), the β_1 estimate is about -0.3738 and the β_7 estimate is 0.8848. If we don't augment the matrix (and inherently don't have an intercept term) the β_1 estimate is -0.4988 and the β_7 estimate is 0.8451.

3 Data analysis [10 pts]

Files hw.2-2.geno contains the genotypes of 500 individuals at 382 SNPs. File hw.2-2.pheno contains 4 continuous phenotypes for each individual. For each phenotype, perform an association analysis of each SNP for the phenotype. To do this, run linear regression for each phenotype against the genotype at each SNP (coded as 0,1 or 2) with an intercept term (in R, you can use the lm function to compute linear regression estimates). We would like to find associations while controlling for the FWER. We will use the Bonferroni procedure to control FWER.

For this question, include any plots along with your answers (do not include code).

(a) What is the significance level at which we reject each single SNP test to control the overall FWER at 0.05 ?

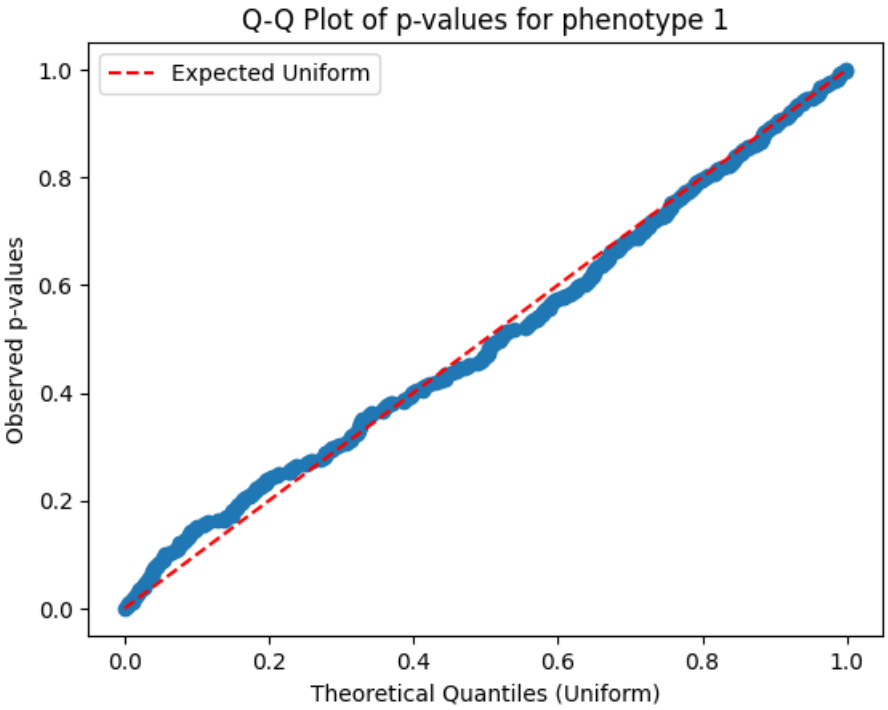
Using the Bonferonni procedure, we would set the significance level to 0.05/m. In this case, m represents the number of SNPs, which is 382. Thus, the level is 0.05/382, about 0.00013.

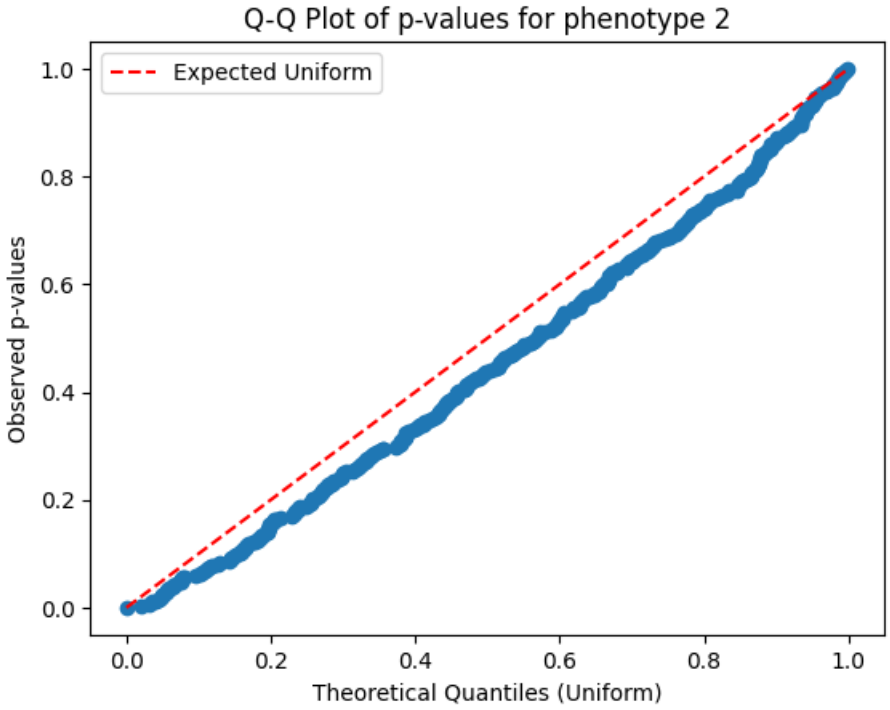
(b) For which SNPs and phenotypes, can we reject the null hypothesis of no association at the chosen signficance level ?

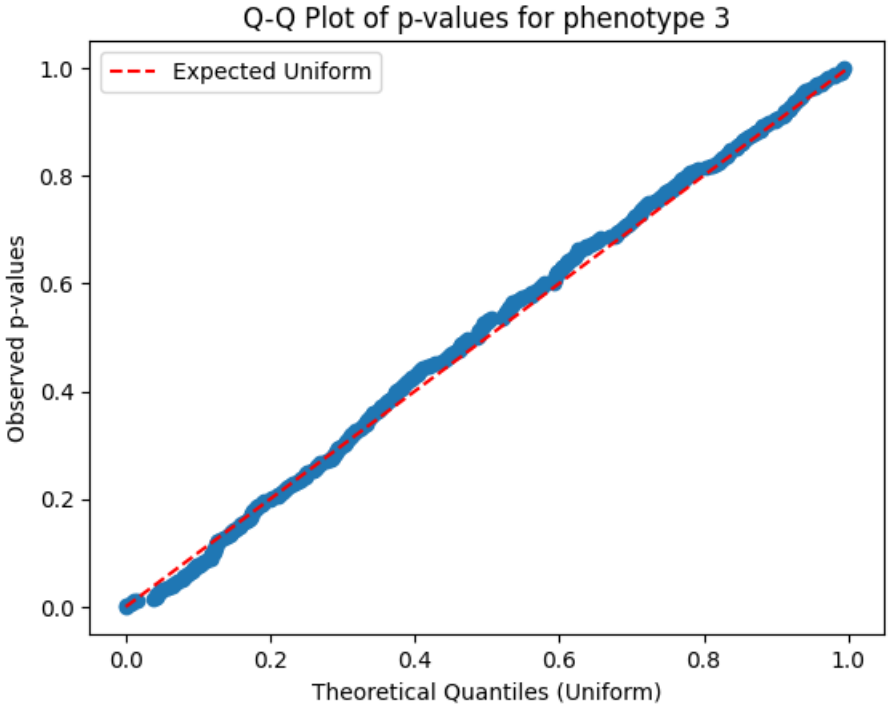
For Phenotype 1, reject no SNPs. For Phenotype 2, reject SNP 258. For Phenotype 3, reject SNP 119. For Phenotype 4, reject SNP 43.

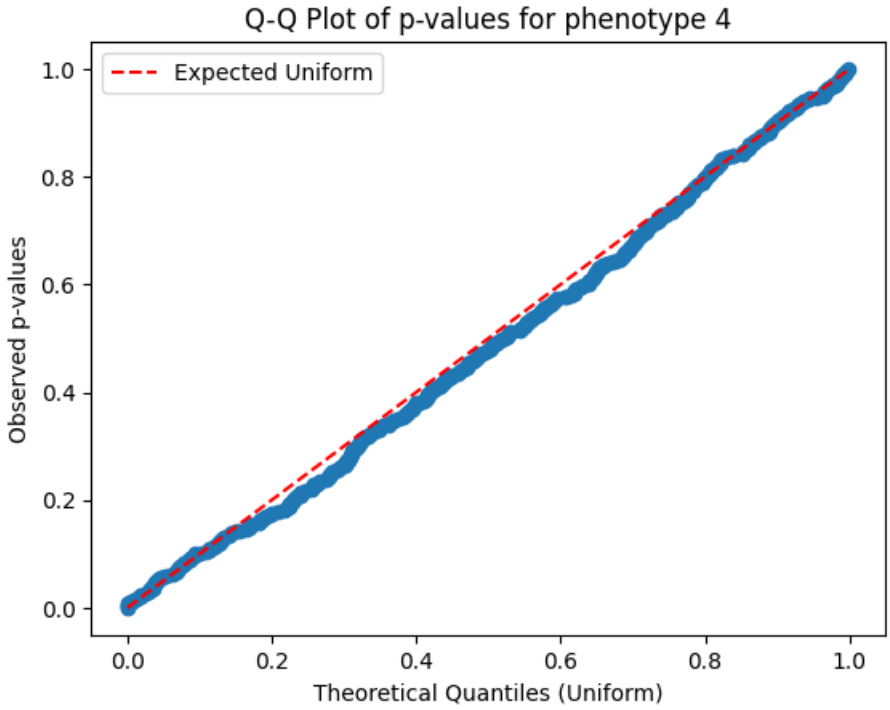
(c) For each association (*i.e.*, rejected null hypothesis), we need to rule out the possibility that the rejection could be a result of the linear regression model being incorrect. If the p-values are not uniformly distributed, this could indicate that the model assumptions are violated. For each phenotype, do the p-values across the SNPs look uniformly distributed ?

For Phenotypes 3 and 4, the histogram plot and q-q plot show a definite uniform distribution. For Phenotype 1 (which we didn't reject a SNP for), the distribution of the Q-Q plot follows a uniform distribution, and the histogram is about uniform (just a bit inconsistent). Thus, the p values for Phenotype 1 are approximately uniform. For Phenotype 2, the histogram is left-skewed and q-q plot distributions aren't aligned. Thus, Phenotype 2 p values aren't uniformly distributed.

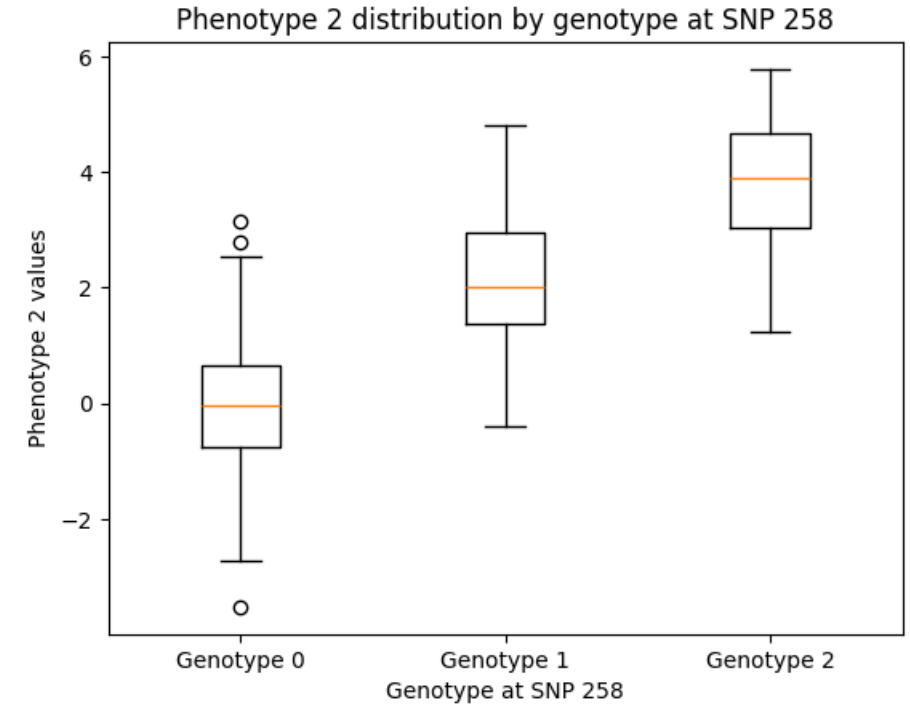

Q-Q Plot of p-values for phenotype 1
Expected Uniform
Observed p-values
Theoretical Quantiles (Uniform)
0.0
0.2
0.4
0.6
0.8
1.0

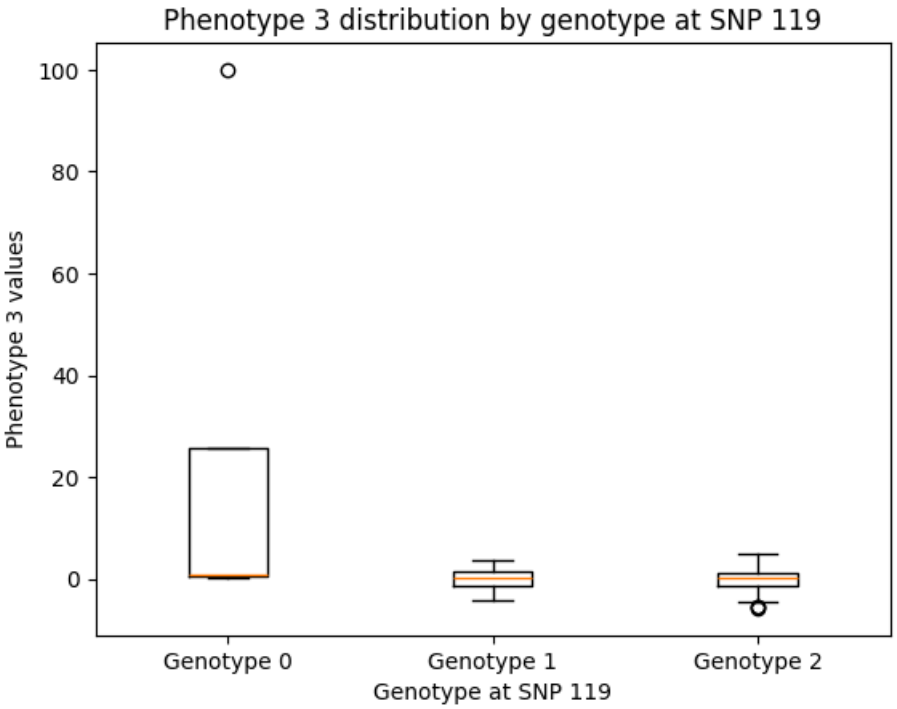

Q-Q Plot of p-values for phenotype 2
Expected Uniform
Observed p-values
Theoretical Quantiles (Uniform)
0.0
0.2
0.4
0.6
0.8
1.0

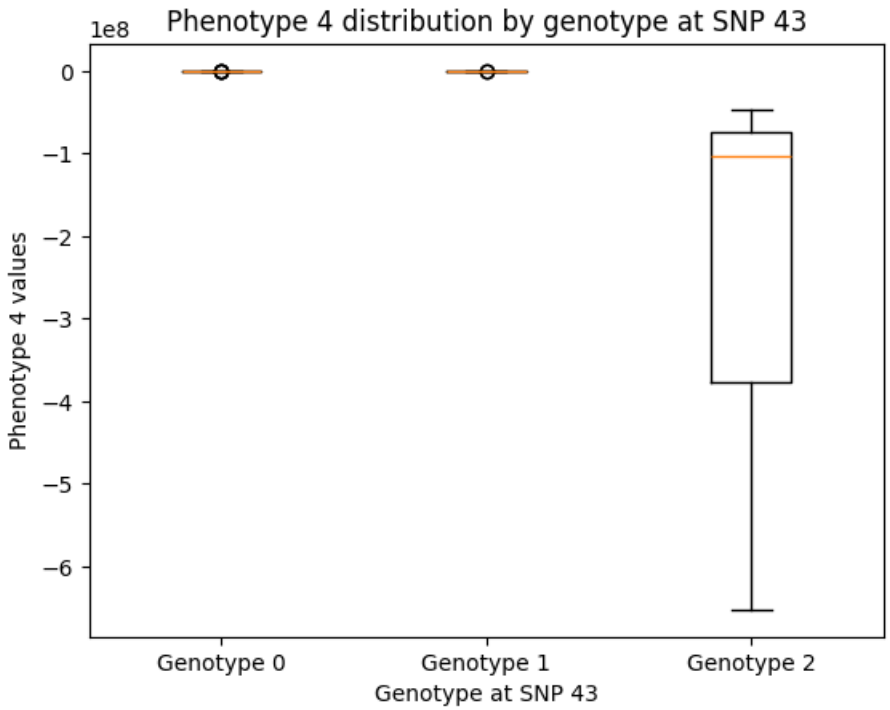

Q-Q Plot of p-values for phenotype 3
Expected Uniform
Observed p-values
Theoretical Quantiles (Uniform)
0.0
0.2
0.4
0.6
0.8
1.0


Q-Q Plot of p-values for phenotype 4
Expected Uniform
Observed p-values
Theoretical Quantiles (Uniform)
0.0
0.2
0.4
0.6
0.8
1.0

(d) An additional check is to plot the relationship between phenotype (on y-axis) vs genotype (on x-axis) for the SNPs that are discovered to be associated. Since the genotype takes values in $\{0, 1, 2\}$, we can use either use a boxplot (boxplot in R) or add random noise to the genotype to aid visualization. Plot this relationship for each SNP-phenotype association. Which of the associations might be a result of model violation? In each of these cases, what assumption do you think is violated ?







For SNP 119 and SNP 43 (and their respective phenotypes), I think the model assumptions of the model being linear is violated. From those 2 plots, we can see that the the phenotype isn't linearly influenced based on the genotype present.