Homework 4

Context

This assignment reinforces ideas in Module 4: Constrained Optimization. We focus specifically on implementing quantile regression and LASSO.

Due date and submission

Please submit (via Canvas) a PDF containing a link to the web address of the GitHub repo containing your work for this assignment; git commits after the due date will cause the assignment to be considered late. Due date is Wednesday, 4/2 at 10:00AM.

Repository link: https://github.com/dliao1/bios731_hw4_liao

Points

Problem	Points
Problem 0	20
Problem 1	20
Problem 2	30
Problem 3	30

Dataset

The dataset for this homework assignment is in the file `cannabis.rds`. It comes from a study conducted by researchers at the University of Colorado who are working to develop roadside tests for detecting driving impairment due to cannabis use. In this study, researchers measured levels of THC—the main psychoactive ingredient in cannabis—in participants' blood and then collected other biomarkers and had them complete a series of neurocognitive tests. The goal of the study is to understand the relationship between performance on these neurocognitive tests and the concentration of THC metabolites in the blood.

The dataset contains the following variables:

- `id`: subject id
- `t_mmr1`: Metabolite molar ratio—a measure of THC metabolites in the blood. This is the outcome variable.
- `p_*`: variables with the `p_` prefix contain measurements related to pupil response to light.
- `i_*`: variables with the `i_` prefix were collected using an iPad and are derived from neurocognitive tests assessing reaction time, judgment, and short-term memory.
- `h_*`: Variables related to heart rate and blood pressure.

Problem 0

This "problem" focuses on structure of your submission, especially the use git and GitHub for reproducibility, R Projects to organize your work, R Markdown to write reproducible reports, relative paths to load data from local files, and reasonable naming structures for your files.

To that end:

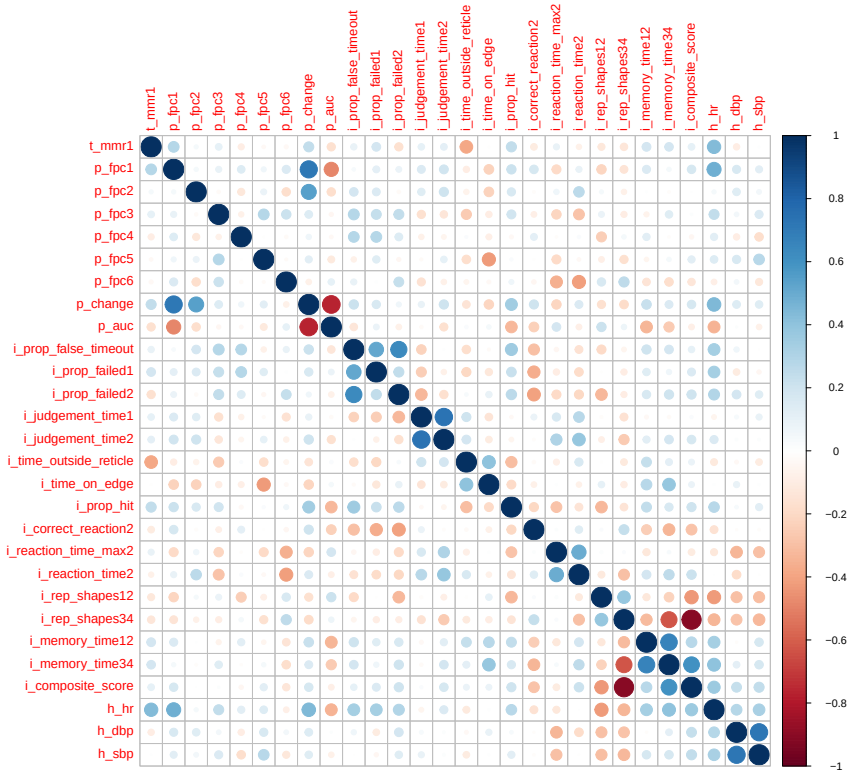
- Create a public GitHub repo + local R Project; I suggest naming this repo / directory bios731_hw4_YourLastName (e.g. bios731_hw4_wrobel for Julia)
- Submit your whole project folder to GitHub
- Submit a PDF knitted from Rmd to Canvas. Your solutions to the problems here should be implemented in your .Rmd file, and your git commit history should reflect the process you used to solve these Problems.

Problem 1: Exploratory data analysis

Perform some EDA for this data. Your EDA should explore the following questions:

- What are n and p for this data?
- What is the distribution of the outcome?
- How correlated are variables in the dataset?

Summarize key findings from your EDA in one paragraph and 2-3 figures or tables.

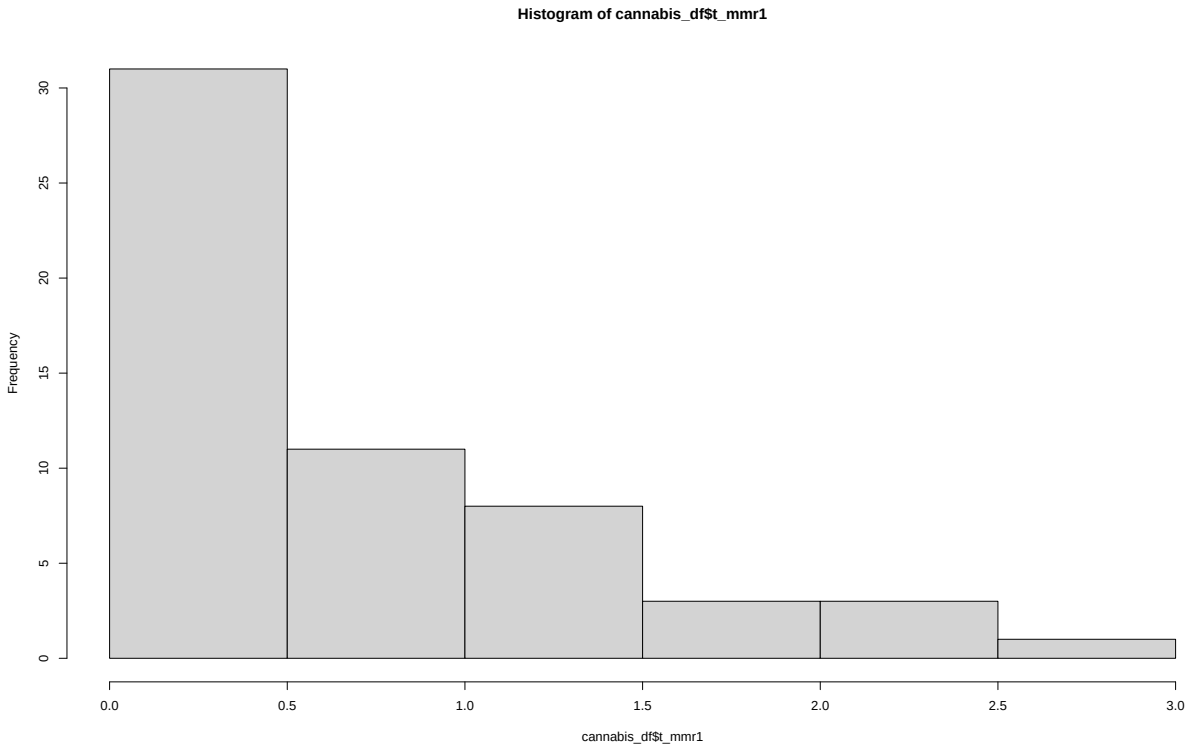


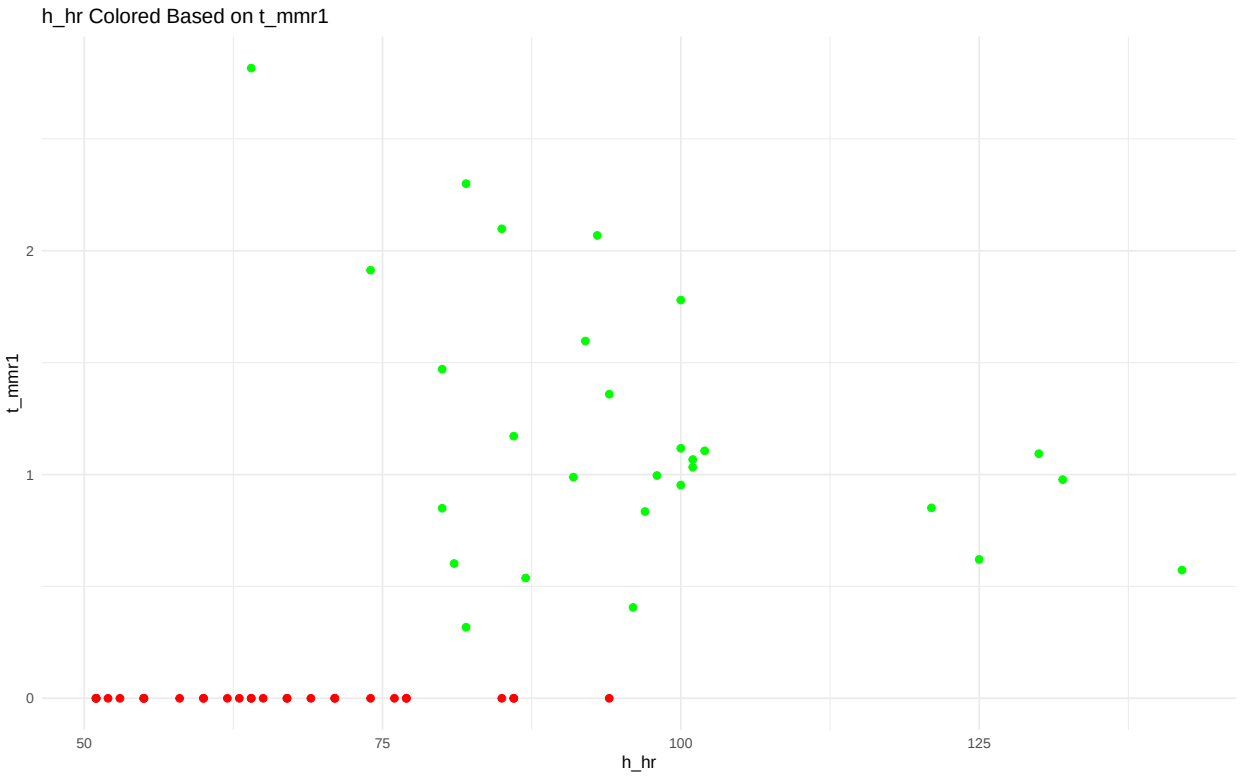
var1	var2	correlation
p_fpc1	p_change	0.7166244

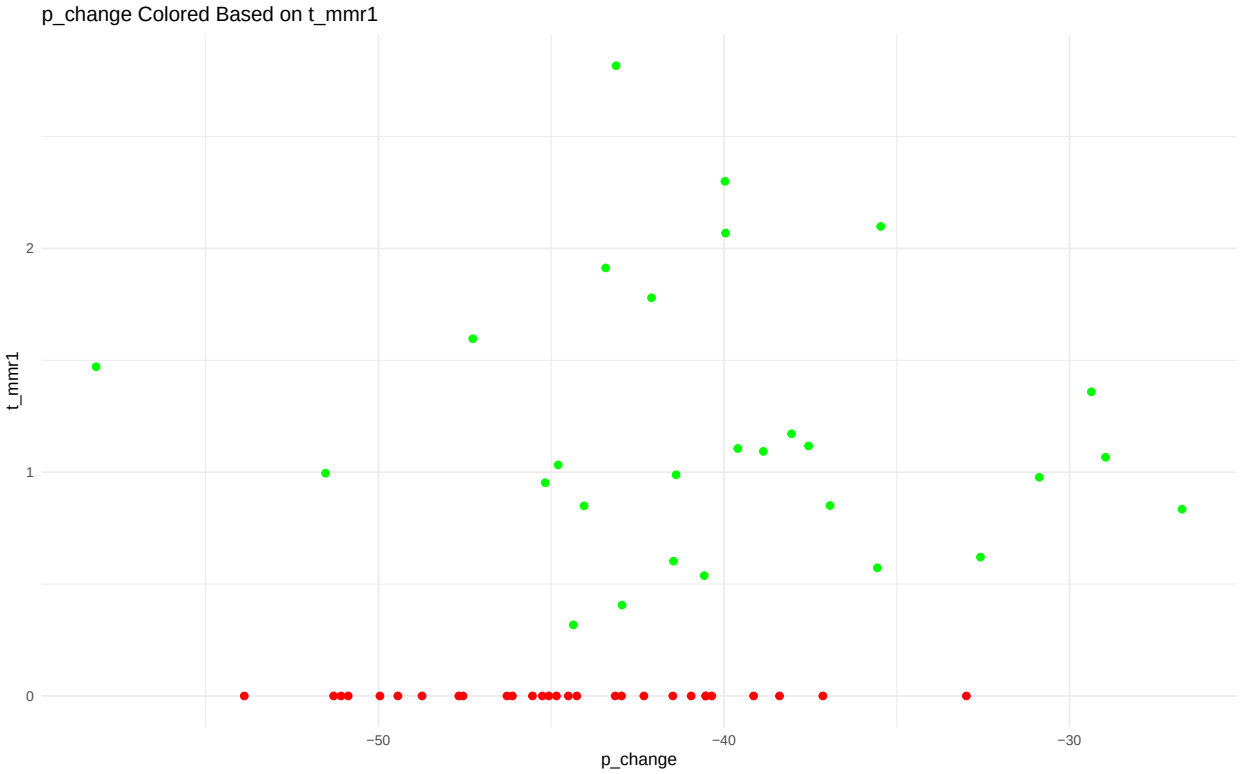
var1	var2	correlation
p_fpc2	p_change	0.5491598
p_change	p_auc	-0.7610100
i_prop_false_timeout	i_prop_failed1	0.5120145
i_prop_false_timeout	i_prop_failed2	0.6399012
i_judgement_time1	i_judgement_time2	0.7398758
i_rep_shapes34	i_memory_time34	-0.6207037
i_memory_time12	i_memory_time34	0.6629413
i_rep_shapes34	i_composite_score	-0.9010134
i_memory_time34	i_composite_score	0.6029671
h_dbp	h_sbp	0.7215615

```
## [1] 57
```

```
## [1] 27
```




h_hr Colored Based on t_mmr1
t_mmr1
2
1
0
50
75
100
125
h_hr


p_change Colored Based on t_mmr1
t_mmr1
2
1
0
−50
−40
−30
p_change

Summary

To summarize, n is 57 for the 57 individuals in the dataset and p is 27 for the number of predictors (covariates in the dataset). After creating a separate table of variables that had correlations higher than an absolute value of 0.5, it looks like most of the variables that had to do with the iPad neurocognitive tests were pretty strongly correlated - for example, i_rep_shapes34 and i_composite_score had a strong negative correlation of -0.9 but i_memory_time34 and i_composite_score had a moderately positive correlation of 0.6.

The distribution of the outcome is highly right skewed - looking at the histogram, it looks like the majority of values are clustered by a metabolite molar ratio of THC of 0. This is partly why I chose to include the two graphs plotting covariates like heart rate and p_change and colored points based on whether the molar ratio of THC was 0 or not - there aren't many obvious linear trends, but it looks like the individuals with higher THC molar ratios tended to have a larger spread of both both heart rates and percentage changes in pupil responses.

Problem 2: Quantile regression

Use linear programming to estimate the coefficients for a quantile regression. You need to write a function named `my_rq`, which takes a response vector y, a covariate matrix X and quantile τ , and returns the estimated coefficients. Existing linear programming functions can be used directly to solve the LP problem (for example, `simplex` function in the `boot` package, or `lp` function in the `lpSolve` package).

- Use your function to model `t_mmr1` from the cannabis data using `p_change` (percent change in pupil diameter in response to light), `h_hr` (heart rate), and `i_composite_score` (a composite score of the ipad variables) as variables.
- Compare your results with though estimated using the `rq` function in R at quantiles $\tau \in \{0.25, 0.5, 0.75\}$.
- Compare with mean obtain using linear regression
- Summarize findings

When explaining your results, be sure to explain what LP method you used for estimating quantile regression.

My Implementation

tau	intercept	x2	x3	x4
0.25	-0.1500508	0.0114183	0.0081762	0.3840630
0.50	-0.2834261	0.0121938	0.0135727	0.1981945
0.75	-1.1140459	0.0041789	0.0272804	-0.5809066

Using rq() in R

tau	(Intercept)	p_change	h_hr	i_composite_score
0.25	-0.1500508	0.0114183	0.0081762	0.3840630
0.50	-0.2834261	0.0121938	0.0135727	0.1981945
0.75	-1.1140459	0.0041789	0.0272804	-0.5809066

OLS

(Intercept)	p_change	h_hr	i_composite_score
-0.1787929	0.0076668	0.014351	-0.2382275

Summary

Looking at the 3 summary tables for my implementation, rq(), and OLS, it looks like my implementation (that used the simplex function to solve the LP problem) and that of rq() generated the same results, but they different from the output obtained using linear regression. It looks like at quantiles 0.25 and 0.5, quantile regression picks up on the positive associations of each of p_change, h_hr, and i_composite_score with t_mmr1, but at a quantile of 0.75, the association of i_composite_score and t_mmr1 becomes negative, switching to -0.58. In contrast, the mean obtained using linear regression has smaller beta values, especially when we compare those with the results of quantile regression at the median (tau of 0.5).

Problem 3: Implementation of LASSO

As illustrated in class, a LASSO problem can be rewritten as a quadratic programming problem.

1. Many widely used QP solvers require that the matrix in the quadratic function for the second order term to be positive definite (such as `solve.QP` in the `quadprog` package). Rewrite the quadratic programming problem for LASSO in matrix form and show that the matrix is not positive definite, thus QP solvers like `solve.QP` cannot be used.

From slide 27 of the lecture slides, we are trying to max:

$$\max_{\beta_j^+,\beta_j^-} -\sum_i \left(y_i - \sum_j \beta_j^+ x_{ij} + \sum_j \beta_j^- x_{ij} \right)^2 \text{ s.t.} \sum_j \left(\beta_j^+ + \beta_j^-\right) \leq \lambda, \beta_j^+, \beta_j^- \geq 0$$

If we factor the x_{ij} , we get

$$-\sum_i \left(y_i - \left[x_{ij} (\sum_j (\beta_j^+ + \beta_j^-) \right] \right)^2$$

Putting it into matrix form and redistributing, we get:

$$(Y - \left[X(\beta_j^+ - \beta_j^-)\right])^T (Y - \left[X(\beta_j^+ - \beta_j^-)\right])$$

$$(Y - \left[X\beta_j^+ - X\beta_j^-\right])^T (Y - \left[X\beta_j^+ - X\beta_j^-\right])$$

Defining new matrices:

$$X' = \begin{bmatrix} X \\ -X \end{bmatrix}$$

$$\beta' = \begin{bmatrix} \beta_j^+ \\ \beta_j^- \end{bmatrix}$$

we can rewrite as:

$$(Y - X'\beta')^T(Y - X'\beta')$$

Expanding, we can see that $\beta'^T X'^T X'\beta'$ is the second order term. Substituting back in our matrix definition of X', we see that:

$$\begin{aligned}(Y - X'\beta')^T(Y - X'\beta') &= Y^TY - 2Y^TX'\beta' + \beta'^TX'^TX'\beta' \\ &= Y^TY - 2Y^TX'\beta' + \beta'^T \begin{bmatrix} X^TX & -X^TX \\ -X^TX & X^TX \end{bmatrix} \beta'\end{aligned}$$

We can see that the determinant of this matrix is going to be 0 - by definition, the determinant of a positive definite matrix is not 0, hence, this matrix is not positive definite.

2. The `LowRankQP` function in the `LowRankQP` package can handle the non positive definite situation. Use the matrix format you derived above and `LowRankQP` to write your own function `my_lasso()` to estimate the coefficients for a LASSO problem. Your function needs to take three parameters: Y (response), X (predictor), and *lambda* (tuning parameter), and return the estimated coefficients.

- Use your function to model `log(t_mmr1)` from the cannabis data using all other variables as potential covariates in the model
- Compare your results with those estimated using the `cv.glmnet` function in R from the `glmnet` package
- Summarize findings

The results will not be exactly the same because the estimation procedures are different, but trends (which variables are selected) should be similar.

```
## [1] 0.5513642
```

predictor	my_lasso	glmnet
p_fpc1	0.0002	0.0012
p_fpc2	0.0011	0.0000
p_fpc3	0.0022	0.0000
p_fpc4	-0.0015	0.0000
p_fpc5	0.0056	0.0000
p_fpc6	0.0020	0.0000
p_change	-0.0031	0.0000
p_auc	0.0227	0.0000
i_prop_false_timeout	0.0000	0.0000
i_prop_failed1	0.0000	0.0000
i_prop_failed2	0.0000	-3.8174
i_judgement_time1	0.3010	0.0000
i_judgement_time2	0.0000	0.0000
i_time_outside_reticle	-0.0019	-0.0322
i_time_on_edge	0.0399	0.0000
i_prop_hit	0.0000	0.0508
i_correct_reaction2	0.0000	0.0000
i_reaction_time_max2	0.0000	0.0000
i_reaction_time2	0.0000	0.0000
i_rep_shapes12	-0.1017	0.0000
i_rep_shapes34	0.0152	0.0000
i_memory_time12	0.0000	0.0000

predictor	my_lasso	glmnet
i_memory_time34	0.0000	0.0000
i_composite_score	0.0000	0.0000
h_hr	0.0001	0.1258
h_dbp	-0.0017	0.0000
h_sbp	-0.0002	0.0000

Summary

After looking at the results of both my implementation of LASSO (with a lambda of 0.5) and those of cv.glmnet, the predictors selected were mostly the same, with both models selecting p_fpc1, i_time_outside_reticle, and h_hr. Both models also only selected a few sparse predictors. However, it is interesting that i_prop_failed2 was selected in glmnet with a beta of -3.8, but not in my model, and it would be interesting to further analyze why.