1. *Monotonicity of loss and regularizer as the regularization parameter changes.* In regularized empirical risk minimization, we choose the parameter $\theta \in \mathbb{R}^d$ to minimize the regularized empirical risk, $\mathcal{L}(\theta) + \lambda r(\theta)$, where $\mathcal{L}(\theta)$ is the empirical risk, $r(\theta)$ is the regularizer, and $\lambda > 0$ is the regularization hyper-parameter. (The exact form of the functions $\mathcal{L}$ and $r$ is irrelevant in this problem.) The hyper-parameter $\lambda$ is used to trade off the two objectives, $\mathcal{L}(\theta)$ and $r(\theta)$. Intuition suggests that as $\lambda$ increases, $r(\theta)$ decreases while $\mathcal{L}(\theta)$ increases. In this exercise we verify that this is the case.

Suppose $0 < \lambda \leq \tilde{\lambda}$. Let $\theta^*$ minimize $\mathcal{L}(\theta) + \lambda r(\theta)$, and $\tilde{\theta}^*$ minimize $\mathcal{L}(\theta) + \tilde{\lambda} r(\theta)$.

(a) Show that $r(\theta^*) \geq r(\tilde{\theta}^*)$. In other words, increasing $\lambda$ will never make our regularization error larger.

(b) Show that $\mathcal{L}(\theta^*) \leq \mathcal{L}(\tilde{\theta}^*)$. That is, increasing $\lambda$ will never decrease our training error.

*Hint.* Use the fact that $\theta^*$ is the minimizer of $\mathcal{L}(\theta) + \lambda r(\theta)$ and similarly for $\tilde{\theta}^*$. This means that for any $\theta$, we have $\mathcal{L}(\theta) + \lambda r(\theta) \geq \mathcal{L}(\theta^*) + \lambda r(\theta^*)$, and similarly for $\tilde{\theta}^*$.

2. *Sequential outlier removal.* We consider the problem of fitting data corrupted with outliers, using a simple sequential outlier removal method. In `outlier_rem.json`, you will find a $250 \times 10$ matrix `U_train` and a 250-vector `v_train` consisting of raw training input and output data, and a $250 \times 10$ matrix `U_test` and a 250-vector `v_test` consisting of raw test input and output data, respectively. We will work with input and output embeddings $x = \phi(u) = u$ and $y = \psi(v) = v$. and you will use a simple linear predictor (without a constant feature) with square loss to fit the model. We will judge model performance using the RMS error on the test set.

A number of the output data entries in the training set have been corrupted (but in a non-obvious way). You do not know which data points have been corrupted, or how many, but you can assume no more than 50. You will explore a simple sequential method to remove the corrupted data points and form a prediction model.

Repeat the following for 50 iterations:

- Create a linear predictor from the training data set.
- Find the data point in the training data set with the largest prediction error.
- Remove the data point from the training data set.

This results in 50 predictors. Plot the test RMS error for them, versus the number of points removed. Give a guess as to how many of the data points were corrupted, with justification.

*Julia hint.* To remove row $i$ from a matrix $X$ and a vector $y$, use $X[\text{setdiff}(1:\text{end}, i), :]$ and $y[\text{setdiff}(1:\text{end}, i)]$, respectively.