Assignment 2 - Part 1A

Penalized regression

Fernando Cagua

1 How many predictors

```
Credit <- read_csv("Credit.csv")
# a model matrix that includes all interactions
X <- model.matrix(balance ~ . * . , Credit)
# train and test set
set.seed(987654312)
train <-sample(1:nrow(X),nrow(X)/2)
test <- -train
# number of predictors
p <- ncol(X)-1
# linear model
y <- Credit$balance
mod <- lm(y[train] ~ X[train, -1])
```

Including all interactions and the main effects results in $p = 65$ predictors.

2 Selecting tunning parameters

```
grid <- 10^seq(3, -1, length.out = 100)
cv_ridge <- cv.glmnet(X[train,], y[train], alpha = 0, lambda = grid, nfolds = 10)
cv_lasso <- cv.glmnet(X[train,], y[train], alpha = 1, lambda = grid, nfolds = 10,
                      thresh = 1e-10)
# best tunning parameters
best_lambda_ridge <- cv_ridge$lambda.min
best_lambda_lasso <- cv_lasso$lambda.min
# number of features in lasso
n_features <- sum(coef(cv_lasso, s = "lambda.min")[, 1] != 0) - 1
```

The best tunning parameter for the ridge regression is 1.7886495 and the best for the lasso regression is 1.9630407. Inspecting the final model for each regression using `coef.glmnet` reveals that, as expected, the ridge regression produces a final model that contains all the predictors. Contrastingly, the lasso regression produces a final model that contains only a subset of the predictors. Specifically the lasso regression using the lambda that produces the minimum cross-validated error includes 25 features.

3 Comparing test errors

```
prediction <- list(
  linear =  X[test,] %*% coef(mod),
  ridge = predict(cv_ridge, X[test,], s = best_lambda_ridge),
  lasso = predict(cv_lasso, X[test,], s = best_lambda_lasso)
) %>%
```

```
  lapply(`colnames<-`, NULL)
test_error <- function(x) mean((x - y[test])^2)
error <- lapply(prediction, test_error)
```

The test error for the linear model is 7055.9534908, for the ridge regression is 7129.8996966, and for the lasso is 4895.6926418. This indicates that using this particular test error, the lasso regression performed the best.

4 Comparing the predictions

```
as.data.frame(prediction) %>%
  mutate(actual = y[test]) %>%
  reshape2::melt("actual", value.name = "predicted", variable.name = "method") %>%
  ggplot(aes(x = actual, y = predicted)) +
  geom_point(aes(fill = method), shape = 21, show.legend = F, alpha = 0.5) +
  geom_abline(intercept = 0, slope = 1) +
  coord_equal() +
  facet_grid(~method)
```