IE8990: Adv. Data Analytics for Complex Systems

- Lab 3
  - Shrinkage methods for regression

Function glmnet

- glmnet(x, y, family=c("gaussian","binomial","poisson","multinomial","cox","mgaussian"), weights, offset=NULL, alpha = 1, nlambda = 100,…)
- alpha is
  - Ridge: α=0
  - Lasso: α=1 is the default
  - elastic-net: with range $\alpha \in [0,1]$

- https://www.rdocumentation.org/packages/glmnet/versions/2.0-18/topics/glmnet

Load library and data

```
# Load libraries, get data & set seed for reproducibility
---------------------
set.seed(123)    # seed for reproducibility
library(glmnet)  # for ridge regression
library(dplyr)   # for data cleaning
library(psych)   # for function tr() to compute trace of
a matrix

data("mtcars")
```

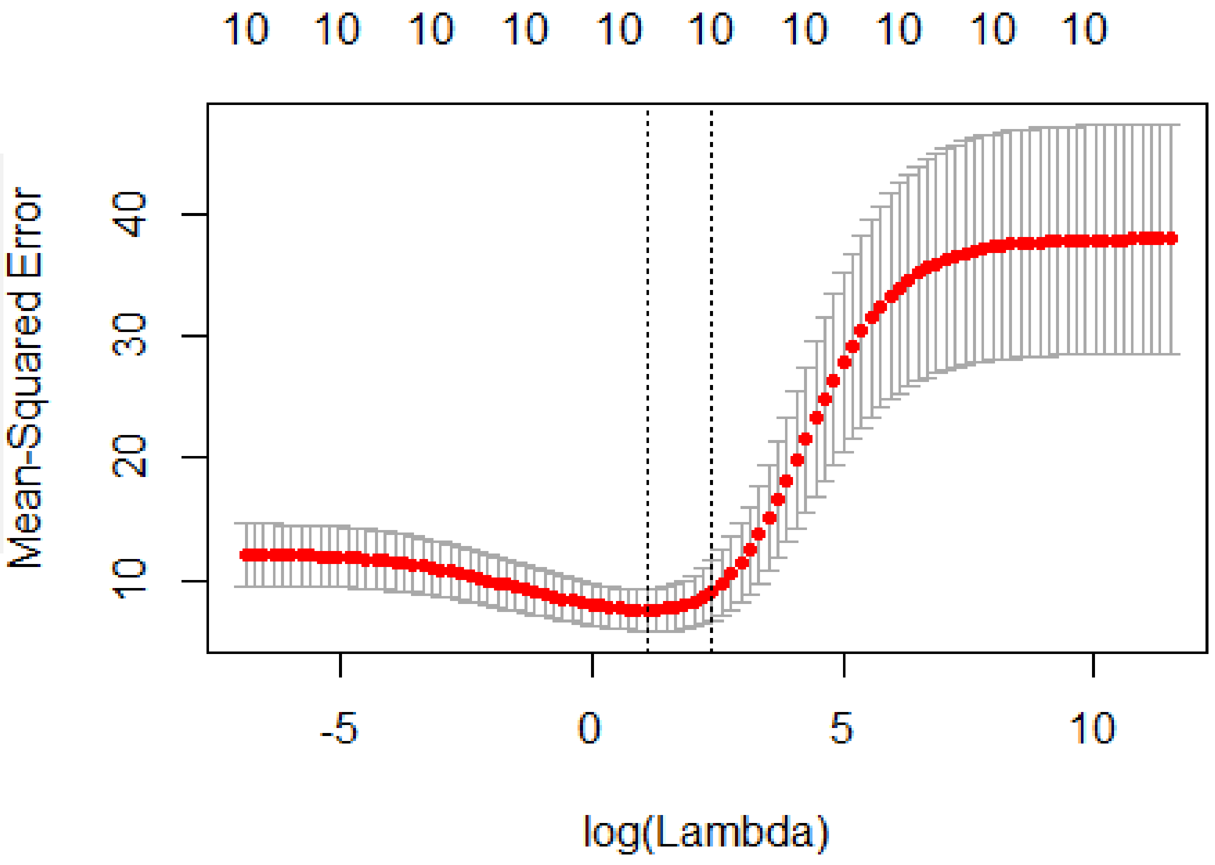
```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0
                  gear carb
Mazda RX4            4    4
Mazda RX4 Wag        4    4
Datsun 710           4    1
Hornet 4 Drive       3    1
Hornet Sportabout    3    2
Valiant              3    1
```

Ridge regression: CV

- Use cross validation to find the best lambda

```r
# Perform 10-fold cross-validation to select lambda
-----------------------------
lambdas_to_try <- 10^seq(-3, 5, length.out = 100)

# Setting alpha = 0 implements ridge regression
ridge_cv <- cv.glmnet(X, y, alpha = 0, lambda = lambdas_to_try,
                      standardize = TRUE, nfolds = 10)
# Plot cross-validation results
plot(ridge_cv)
```



Ridge regression: CV

```
# Best cross-validated lambda
lambda_cv <- ridge_cv$lambda.min
```

- `lambda.min` is the value of λ that gives minimum mean cross-validated error.
- The other λ saved is `lambda.1se`, which gives the most regularized model such that error is within one standard error of the minimum.

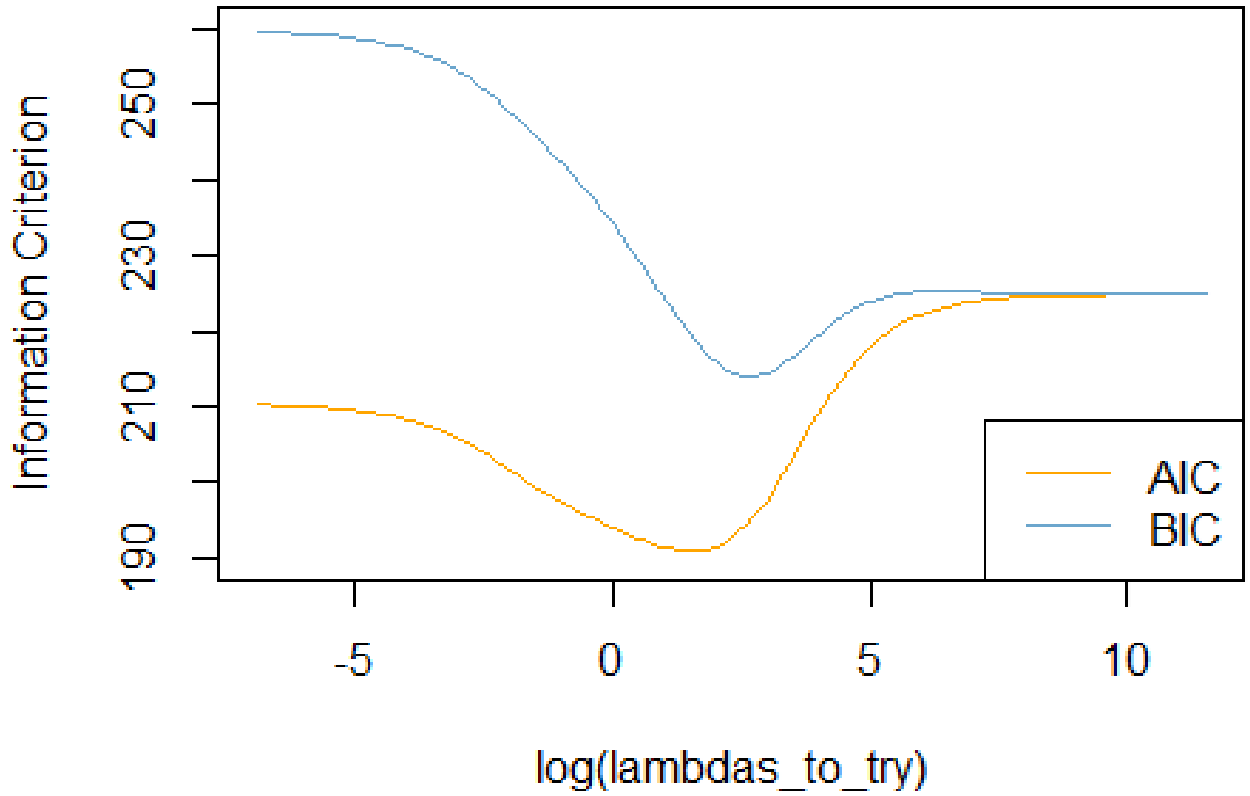
AIC and BIC

```
# Use information criteria to select lambda -----------------------------------
X_scaled <- scale(X)
aic <- c()
bic <- c()
for (i in seq(lambdas_to_try)) {
  # Run model
  model <- glmnet(X, y, alpha = 0, lambda = lambdas_to_try[i], standardize = TRUE)
  # Extract coefficients and residuals (remove first row for the intercept)
  betas <- as.vector((as.matrix(coef(model))[-1, ]))
  resid <- y - (X_scaled %*% betas)
  # Compute hat-matrix and degrees of freedom
  ld <- lambdas_to_try[i] * diag(ncol(X_scaled))
  H <- X_scaled %*% solve(t(X_scaled) %*% X_scaled + ld) %*% t(X_scaled)
  df[i] <- tr(H)
  # Compute information criteria
  aic[i] <- nrow(X_scaled) * log(t(resid) %*% resid) + 2 * df[i]
  bic[i] <- nrow(X_scaled) * log(t(resid) %*% resid) + 2 * df[i] * log(nrow(X_scaled))
}
```

- Here get familiar with how to use a for-loop in R
- You will need it to do Q2 in HW1

AIC and BIC

- What can we observe from the plot?
  - Lambda vs AIC/BIC
  - AIC vs BIC



HW 1 Q2: Elastic Net

- Use the same data set we used in class
- Explore how the performance of an Elastic net model varies w.r.t
  - Different lambda
  - Different alpha
- Can you find a better model than ridge and the LASSO?
- Hints:
  - Nest the cv.glmnet function in a new for-loop, and iterate across different choices of alpha
  - Use cross validation to find the lowest mean squared error