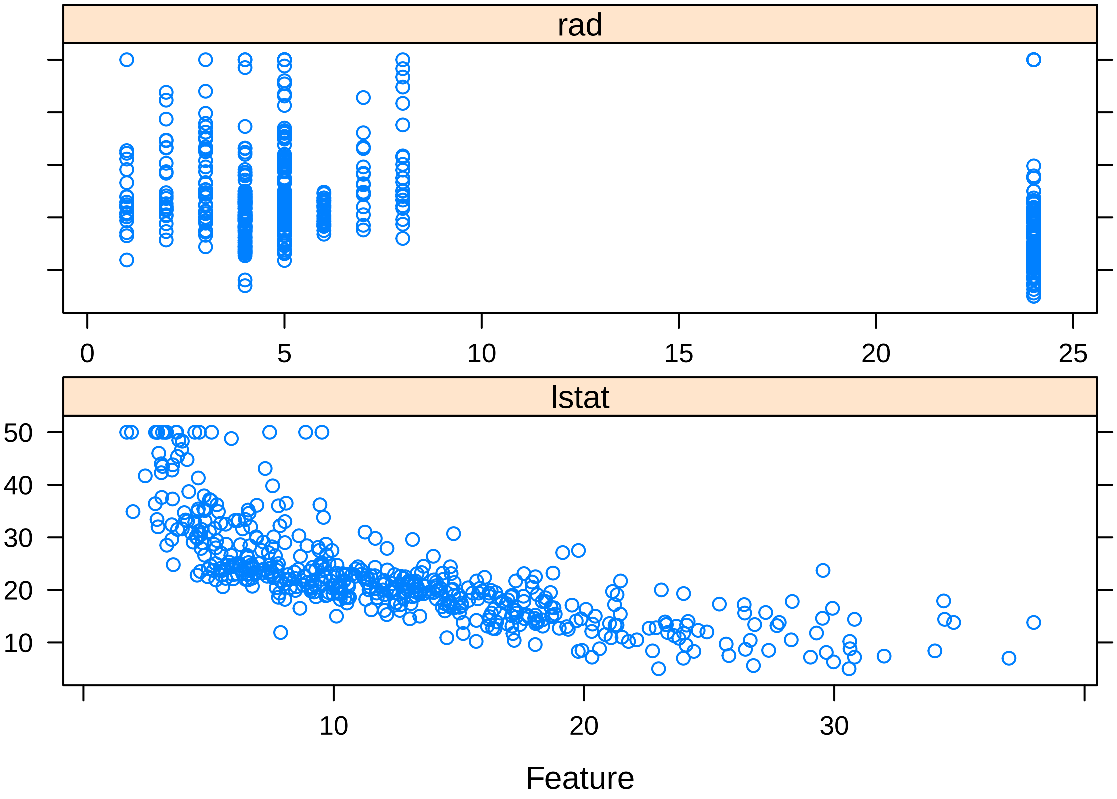
# Stat 542 Spring 2018 Homework One

*Yizhou Zhang*

*Feb 9th, 2018*

### Question 1a: Perform a descriptive analysis on all variables. Comment on any potential issues and address them if needed.

1. First drop all observations with missing values
2. Factorize the variable `chas`
3. Use feature plot to identify the initial relationships. The result shows that the variable `lstat` needs to be log-transformed, and the variable `rad` better be binarized at a cutoff of 20.



4. The correlation matrix shows that `tax` and `rad` has high correlation of `0.91`. I will remove `rad` given its relevant issues in both 3 and 4 discussed above.

```
##           rad       tax
## rad 1.0000000 0.9102282
## tax 0.9102282 1.0000000
```

**Question 1b: Perform the best subset selection using BIC criterion. Report the best model (the selected variables and their parameters).**

The best subset selects a model with nine parameters (eight variables plus one intercept), and the parameter estimates are shown below.

```
##   (Intercept)          crim         chas1           nox            rm
##  47.805597556  -0.071377176   2.528689471 -12.219664540   2.985686728
##           dis       ptratio             b         lstat
##  -1.188796324  -0.768813364   0.006785714  -8.813049628
```

**Question 1c: Perform i) forward stepwise selection using AIC criterion; and ii) backward stepwise selection using Marrow's Cp criterion. Compare these two models with the model in part b).**

The forward AIC approach selects a model with 13 parameters (12 variables plus one intercept), as shown below. This is larger than the model selected in b),probably due to BIC's greater penalty. The coefficients are smaller in the forward AIC approach, since there are more variables and each variables explains a smaller share of total variance.

```
##   (Intercept)          crim            zn         indus         chas1
##  49.111651098  -0.087407004   0.017991356  -0.061297924   2.557350516
##           nox            rm           age           dis           tax
## -13.708507657   2.703867939   0.022591293  -1.250866995   0.002088060
##       ptratio             b         lstat
##  -0.749004093   0.006757854  -9.106865870
```

$C_p$ is equivalent to $AIC$ for linear Gaussian models. The results above below confirm this by showing that backward $C_p$ and $AIC$ select the same model with 11 parameters (10 variables and one intercept). This is less parsimonious than the model selected in part b) with 9 parameters.

According to the documentation of `extractAIC` function, Mallow's $C_p$ is implemented in `step` function if we specify the `scale` argument as the estimated error from full model. In other words, the `step` function should be specified as `step(fit, direction = "backward" , trace = FALSE, scale = sqr_sigma )`, where `sqr_sigma = (summary(fit)$sigma)^2`.

```
# Question ii)
sqr_sigma = (summary(fit)$sigma)^2
fit_cp_bwd = step(fit, direction = "backward" , trace = FALSE, scale = sqr_sigma )
coef(fit_cp_bwd)
```

```
##   (Intercept)          crim            zn         chas1           nox
##  48.411794066  -0.078139774   0.020712125   2.474193741 -13.753825905
##            rm           age           dis       ptratio             b
##   2.778622384   0.021837593  -1.214367937  -0.729220680   0.006581712
##         lstat
##  -9.116490337
```

```
# then show that AIC is equivalent to Cp under linear Gaussian model
fit_aic_bwd = step(fit, direction = "backward" , trace = FALSE)
coef(fit_aic_bwd)
```

```
##   (Intercept)          crim            zn         chas1           nox
##  48.411794066  -0.078139774   0.020712125   2.474193741 -13.753825905
##            rm           age           dis       ptratio             b
##   2.778622384   0.021837593  -1.214367937  -0.729220680   0.006581712
##         lstat
##  -9.116490337
```

**Question 1d:Comment on the advantages and disadvantages of the selection algorithms (best subset, forward, backward and stepwise). If you get different results using these three algorithms (assume that you use the same selection criterion), would you prefer some over others? Why?**

Best subset selection guarantees global optimal solution, but it is very computationally heavy when $p$ is large.Forward, Backward and stepwise selection are faster in computation but return only local optimum, in contrat to best subset selection.Backward selection can only used when $n > p$, while forward selection can always be used. Therefore, best subset selection is preferred when $p$ is not large, and forward selection might better fit a situation with large $p$.

**Question 1e:Comment on the advantages and disadvantages of the three selection criteria ($AIC, BIC, C_p$). If you get different results using these three criteria (assuming the same algorithm), would you prefer some over others? Why?**

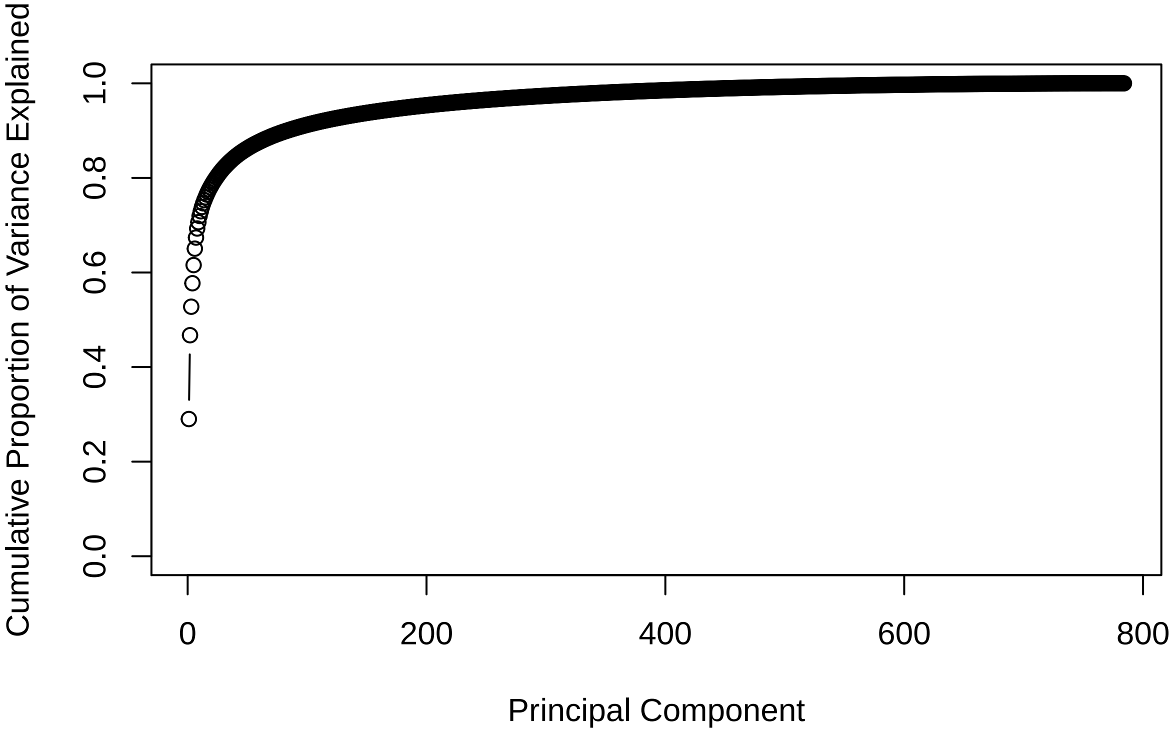
BIC puts much heavier penalty than AIC. For small samples, BIC is more likely to choose an overly simple model than AIC. However, as the sample sizes increases BIC is consistently converging towards the "correct" model, and this is not true for AIC. Therefore, for samll samples AIC may be better than BIC but for large samples BIC seems better.

$C_p$ is equivalent to AIC for linear Gaussian models. Its penalty term also does not depend on sample size, and the loss function $RSS$ is monotonic with $-LogLikelihood$.

## Question 2a:Provide a short summary of the dataset and the research goal.

Fashion-MNIST is a dataset of Zalando's article images, with 60,000 examples in the training set and a test set of 10,000 observations.Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. In other words, there are 784 features. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. The first column is the label representing the article of clothing. The rest of the columns contain the pixel-values of the associated image.

### Principal Component Analysis to reduce dimension



The plot above shows the cumulatative variation explained as we increase the number of Principla components. It shows that 23 factors explain approximately 80% of total variation. So I transform both the test set and the training set to have 23 features as the first 23 principal components.

### Define the function to estimate euclidean function

To speed up the computation, I get rid of the step of taking square root.This should not change the relative distance between points.

```
euclideanDist <- function(a, b){
  d = 0
  for(i in c(1:(length(a)-1) ))
  {
    d = d + (a[[i]]-b[[i]])^2
  }
  return(d)
}
```

### Define the KNN algorithm

In the algorithm below, if there is a tie then whichever class appears first will be chosen.

```
knn_predict = function(test_data, train_data, k_value){
  pred = c()  #empty pred vector
  #LOOP-1
  for(i in c(1:nrow(test_data))){   #looping over each record of test data
    eu_dist =c()
    eu_cls = c()
    #LOOP-2-looping over train data
    for(j in c(1:nrow(train_data))){

      #adding euclidean distance b/w test data point and train data to eu_dist vector
      eu_dist = c(eu_dist, euclideanDist(test_data[i,], train_data[j,]))

      #adding class variable of training data in eu_cls
      eu_cls = c(eu_cls, train_data[j,][[1]])
    }

    eu = data.frame(eu_cls, eu_dist)

    eu = eu[order(eu$eu_dist),]
    eu = eu[1:k_value,]

    #Loop 3: loops over eu and counts classes of neibhors.
    pred = c(pred,as.numeric(names(sort(table(eu$eu_cls), decreasing = TRUE))[1]))
  }
  return(pred) #return pred vector
}
```

Define a function to calculate the accuracy of predicted results compared to the actual labels of the test set. Use Cross Validation to choose the best K through subsets of training data (480 samples)and test data (100 samples). The CV process chooses 10 from one to fifteen. This is a local optimum rather than global given a relatively small search range. The degrees of freedom for the test set is 10000/10 = 1000.

```
calc_acc = function(actual, predicted) {
  mean(actual == predicted)

}
sml_trn = stratified(trn, "V1",0.008) # the train size for CV is 480
sml_tst  =stratified(tst, "V1",0.01)  # the test set for CV has size 100

for (i in 1:15){
fitted =knn_predict(sml_tst, sml_trn, i)
accuracy[i] = calc_acc(actual = sml_tst$V1, predicted = fitted)
}
```

Use the developed algorithm to predict the full test set using a sub-sample of the training set. The subsample consists of 780 observations. The accuracy is 74.41%.

```
med_trn = stratified(trn, "V1",0.013)
# the size of the training set for full test set prediction is 780
final_med = knn_predict(tst, med_trn, 10)
medsample = calc_acc(tst$V1, final_med)
```

**Question 2d: Can you suggest some approaches that can speed up the computation (even at some minor cost of prediction accuracy)?**

1. Perform Principal Component Analysis to reduce the dimensionality of the data.
2. To choose the best parameter $k$, use a subset of training/test data instead of the full samples. The subsets are chosen through stratified sampling.
3. Neglect the step of taking square root when calculating Euclidean distance.
4. Use a subset of training sample in the prediction of full test set.