General Instructions:

1) You will **reasonably attempt** to **optimize** the model for each of the methods requested.
2) You will provide **an explanation (min 2-3 sentences)** of the output **for each model**.
3) All methods should implement a **train** / **test** split. (70/30 is a good **starting** point*)
4) All results should **evaluate** your model using the testing data.
5) Attach the **output** of the summery for each of the models.
6) You should generate and include a **relevant image** or **screen shot** for each of your models. (This should be a **linear plot, ROC, Confusion Matrix, or diagram** of your results for each of the models.)

* split ratios can be adjusted to evaluate the model when trained using a different amount of the total set. Increasing testing data will give a better evaluation of future performance, while at the same time limiting the model's access to representation data (it's always a trade-off).

Part 1: Regression 20 pts

Complete the following objectives utilizing the data set you selected as ideal for:
Linear Regression

Note: you may use the exercise from Chapter 3 as a reference **link here** --> Chapter 3 (Linear Regression).

1) Pick a set of feature value(s) you prefer. Be consistent such that you keep the previous features when adding more for the higher dimensional models.
Each model should be fit to the same **response**.

a) Linear Regression

b) Polynomial Regression

b) Multi-Linear Regression

c) Natural Cubic Spline

Part 2: Feature Selection / Model Optimization Methods 20 pts

Complete the following objectives utilizing the data set you selected as ideal for:
Feature Selection / Model Optimization Methods

1) Perform a Forward Stepwise Selection
Code:

```
#be sure to load the appropriate library
library(leaps)
```

```
#you'll need to change the input here to use the currently stored values you've imported into R
#if you already have your data loaded from previous steps, the variables x and y below *might* work
```

```
#you might need to play around with the nvmax number to find a good value
regfit.fwd <- regsubsets(Salary ~ ., data = Hitters, nvmax = 11, method = "forward")
summary(regfit.fwd)
```

b) Perform a Backward Stepwise Selection
Code:

```
#be sure to load the appropriate library
library(leaps)
```

```
#use this version of the code to run the backward selection process
#you'll need to change the input here to use the currently stored values you've imported into R
#adjust nvmax based on your interpretation of the results provided by the summary
regfit.bwd<- regsubsets(y ~ ., data = x, nvmax = 14, method = "backward")
summary(regfit.bwd)
```

2) Using the models generated for the feature selection, generate the plots of RSS and Adjusted R^2 (as given in the exercise for chapter 6).

Use the code to generate for parts (a) and (b) and save the plots to upload.

a) Forward Features

```
# you may need to adjust this code, (see source material for reference)

reg.summaryfwd <- summary(regfit.fwd)

par(mfrow = c(1, 2))
plot(reg.summaryfwd $rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
plot(reg.summaryfwd $adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
```

b) Backward Features

```
# you may need to adjust this code, (see source material for reference)

reg.summarybwd <- summary(regfit.bwd)

par(mfrow = c(1, 2))
plot(reg.summarybwd $rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
plot(reg.summarybwd $adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
```

3) PCR
Perform a PCR on your selected data as detailed in Chapter 6 lab. Most of the datasets should have sufficient features to make this task interesting. Generate a plot of the components relative to their fit target.

Classification 20 pts

Complete the following objectives utilizing the data set you selected as ideal for:
Classification

Note: you should use the exercise from Chapter 4, 5, 8, 9 as a reference
link here --> Chapter 4 (Classification)Links to an external site. & Chapter 5 (Sampling)Links to an external site. for Reference [Chapter 8 (Trees)Links to an external site. & Chapter 9 (SVM)Links to an external site.]

1) Generate **two** Classification Models for based on your given data.
These models will include Logistic Regression, Linear Discriminant Analysis, as given in the chapter 4 lab exercise.
Note: You'll need to make sure you **properly encode any categorical data** for your classes or predictions.

Provide the **results for each** of your classification functions as a **confusion matrix**.

2) Generate a Tree Classifier for the classes and predictors used above.

a) Utilize the **tree fit** function using the general template below.
(Please see the first section of **lab 8** for additional info)

```
tree.my_tree <- tree(class_target, input_dataset)
```

b) Using the plot function, generate a **plot** of the tree trained by the process in 2 (a) general code below.

```
plot(tree.my_tree)
text(tree.my_tree, pretty = 0)
```

3) Construct a Support Vector Classifier for some set of classes. (Please see the first section of **lab 9** for additional info)

Keep in mind the simplified SVM's trained using the *e1071* solver in R only work well to generate **binary** boundaries, so you'll want to select **just two classes**** from your data.

[**You can do this in R by selecting the relevant parts of the dataframe, or make adjustments and save them as new .csv file before loading the data for this part of the exercise]

First you'll want to load your selected data into a frame for training.
Note: Be sure to use the argument "as.factor" when passing in non-numerical labels.

If for some reason the code below doesn't work when using copy-paste try typing it in directly or copying it from relevant sections of the Chapter 9 lab [grey boxes].

CODE:

```
#Generic examples for inputs given as x and y here are numericaldata and class label.
dat <- data.frame(x = x, y = as.factor(y))

# Be sure to load the library needed.
library(e1071)
# if you believe your data has isolated pockets of classes, you can use the argument "radial" rather than linear for the training.
svmfit <- svm(y ~ ., data = dat, kernel = "linear", cost = 10, scale = TRUE)

#generate a plot for submission in the next step.
plot(svmfit, dat)

#using the built in tune function, we adjust the value of "cost" a few times (7 values) and see the results for a 10-Fold Cross Validation

tune.out <- tune(svm, y ~ ., data = dat, kernel = "linear",
   ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))

#we output the "best" performance model from the k-fold above
summary(tune.out)

#we copy that model and use summary again to show the parameters of that training
bestmod <- tune.out$best.model
summary(bestmod)
```

Models 20 pts

We've covered a variety of models this semester, and each has it's use and application but some work better than others to solve specific situations.
For the following scenarios please provide a brief response based on the things you have learned about each of the **methods we've discussed** and **implemented**.
(No answer is absolutely right or wrong but you'll need to (briefly*) justify your response)
*Please at least to two sentences per item.

a) (5 points) A friend is starting a company and wants your help to see if they can figure out what factors most closely relate to the relative level of success for key competitors. They have gathered a few factors about each company such as **total inventory**, **number of employees**, **annual operation budget** and **total profits**. What method might you use to help your friend determine if their business model might be a success? Why did you choose this model?

b) (5 points) An advertisement firm has hired you to help them optimize their mailing list. They currently are looking to promote their client's store by sending packages of coupons to select areas. We want to know which postal codes the company should mail to for maximum impact (shoppers come to the store with coupons). They currently have some survey data randomly sampled from homes in the area indicating how likely they were to shop at the client's location. What method might you try first to generate the mailing map? Why?

c) (5 points) A large company has been collecting data about their customers preferences for many years. They've hired you to help them transform the millions of samples and thousands of search and behavior features into a set of simplified features they can use to build a model which provides suggestions to their customers for future services. What method might you suggest first? Why?

d) (5 points) A company that specializes in shipping fruit to grocery stores wants to save money by sorting out bad fruit from good fruit before it goes on the truck. They have presented you with a device that can measure features like weight, color, size, and look for possible bad spots. Each of these measurements is imprecise, and there is significant overlap between the classes for most of the features. What supervised learning methods might you try? Why?