STAT 847: Analysis Assignment 2

Chris Binoi Verghese ID: 21092999

```
library(randomForest)
library(rpart)
library(rpart.plot)
library(leaps)
library(factoextra)
library(car)
```

1. (4 points) Using the `randomForest` function in `library(randomForest)`, make five random forests, each one using one of the phenotype variable `Perimeter_Growth` as a response y variable. The forest should use all 10,000 of the gene variables (These are the 9th, ... , 10,008th variables). Give your forest 500 trees, have each tree use 300 gene variables, and set a minimum node size of 1. Sample with replacement. Report the percentage of variance explained by the forest using `print()`.

```
dat = read.csv("F1-Hybrids_Pheno_10000genes.csv")
genes = dat[,9:10008]
pheno = dat[,1:8]
p_var = c(1,2,3,4,5)
for (i in 1:5) {
  forest <- randomForest(x=genes, y = pheno$Perimeter_Growth,
                         ntree = 500, mtry = 300, node_size = 1,
                         replace = TRUE)
  print(forest)
  p_var[i] = round(100 * forest$rsq[length(forest$rsq)], digits = 2)
}
```

```
##
## Call:
##  randomForest(x = genes, y = pheno$Perimeter_Growth, ntree = 500,      mtry = 300, replace = TRUE, n
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 300
##
##           Mean of squared residuals: 1.51286
##                     % Var explained: 40.11
##
## Call:
##  randomForest(x = genes, y = pheno$Perimeter_Growth, ntree = 500,      mtry = 300, replace = TRUE, n
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 300
##
##           Mean of squared residuals: 1.526956
##                     % Var explained: 39.55
##
## Call:
##  randomForest(x = genes, y = pheno$Perimeter_Growth, ntree = 500,      mtry = 300, replace = TRUE, n
```

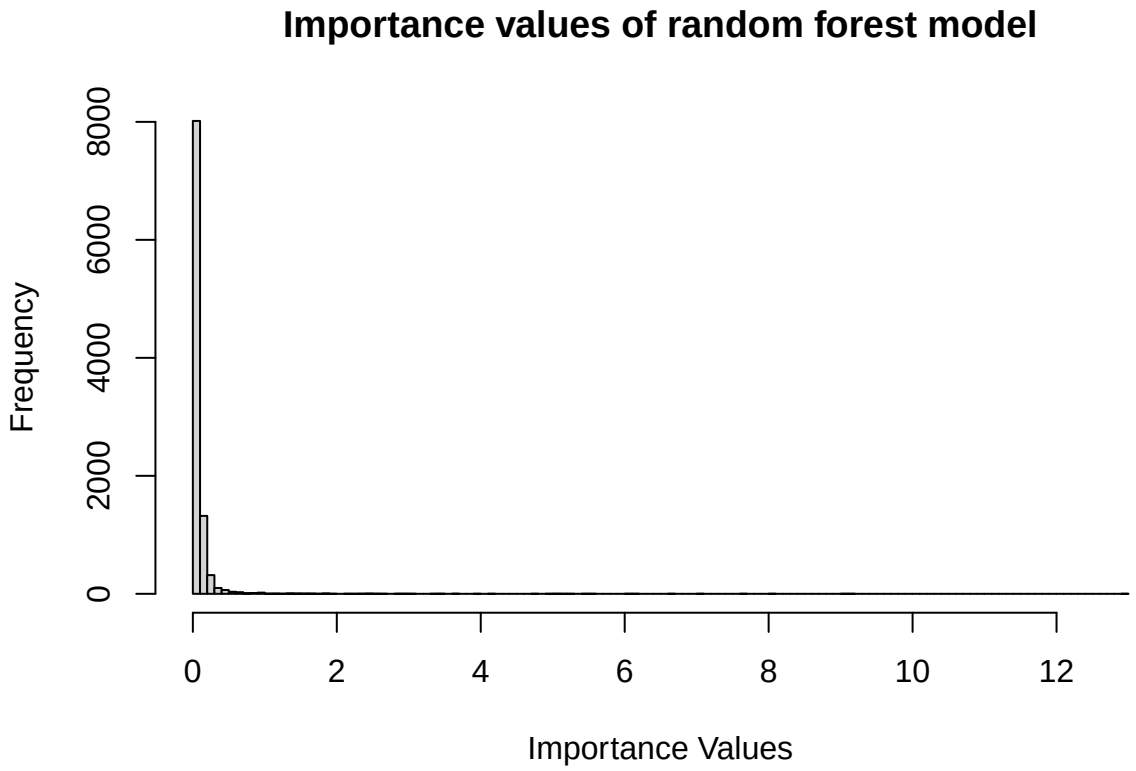
```
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 300
##
##           Mean of squared residuals: 1.541845
##                     % Var explained: 38.96
##
## Call:
##  randomForest(x = genes, y = pheno$Perimeter_Growth, ntree = 500,      mtry = 300, replace = TRUE, n
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 300
##
##           Mean of squared residuals: 1.487995
##                     % Var explained: 41.09
##
## Call:
##  randomForest(x = genes, y = pheno$Perimeter_Growth, ntree = 500,      mtry = 300, replace = TRUE, n
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 300
##
##           Mean of squared residuals: 1.509549
##                     % Var explained: 40.24
```

```
cat("the percentages of variance explained  are:",p_var)
```

```
## the percentages of variance explained  are: 40.11 39.55 38.96 41.09 40.24
```

2. (2 points) Get a `hist()` of the `$importance` values from your random forest model of perimeter growth (not the MPH). Use this to comment on the relative importance of some genes over others in determining perimeter growth. Use 100 bins for the histogram.

```
hist(x = forest$importance,
     main = "Importance values of random forest model",
     xlab = "Importance Values",
     breaks = 100,
     xlim = range(forest$importance),
     plot = TRUE)
```



The largest number of variables (close to 8000 out of 10000) have no importance in helping predict Perimenter Growth in the random Forest and almost all the variable are exhausted before reaching an importance value of 1. However there are a few genes that have an extremely high importance value resulting in the histogram's maximum range reaching up to 12.

Therefore, there are very few genes that have a relatively high importance value in the determination of Perimeter Growth in the random Forest over the 10000 genes provided.

3. (0 marks) Use the following code to make a new dataset that only includes perimeter growth and the most important 50 genetic variables from random forest for perimeter growth. `mod2` is the name of the `randomForest()` output in this case.

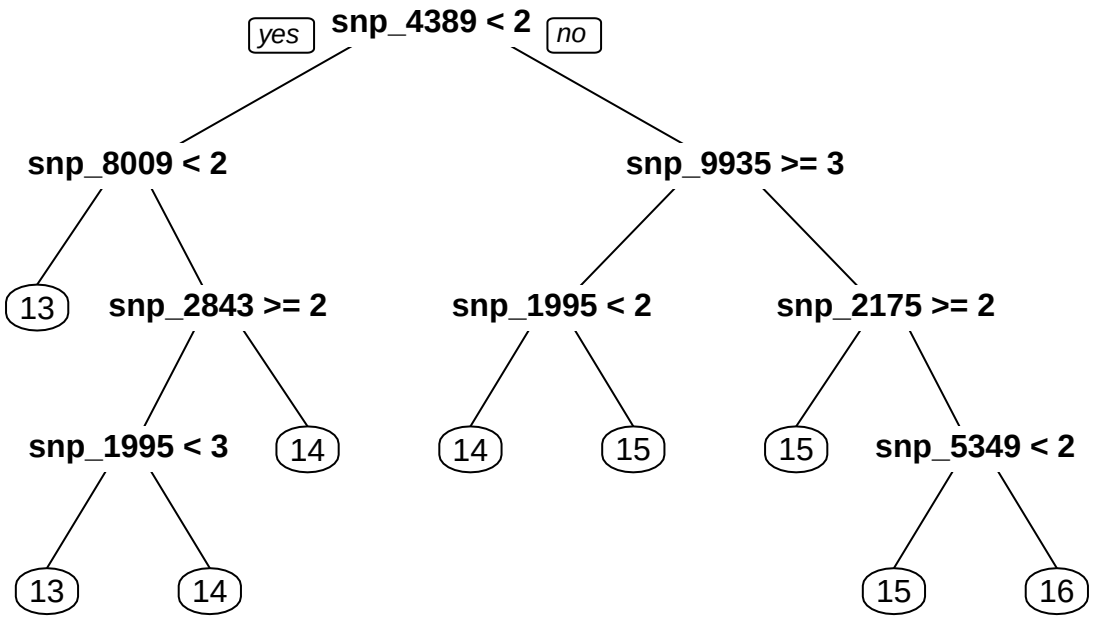
```
# Set a cutoff of the 50th most important variable
cutoff = rev(sort(forest$importance))[50]

# Keep only those 50 variables
idx = which(forest$importance >= cutoff)
genes_imp = genes[,idx]

dat_imp = cbind(pheno$Perimeter_Growth, genes_imp)
names(dat_imp)[1] = "Perimeter_Growth"
```

4. (4 points) Using `rpart`, and this new dataset `dat_imp` (or `genes_imp`) of the 50 most important variables for perimeter growth, create a single regression tree of perimeter growth. Plot the tree with `prp` in the `rpart.plot` package.

```
fit = rpart(dat_imp$Perimeter_Growth ~ ., data = dat_imp)
prp(fit)
```



5. (4 marks) Using `regsubsets` in the `leaps` package, and the new dataset `dat_imp` (or `genes_imp`), use best subsets regression with the Adjusted R-squared criterion. Report the variables of the best model, their coefficients, and the adjusted r-squared of the model.

```
regsub <- regsubsets(dat_imp$Perimeter_Growth ~ ., data = dat_imp, really.big = TRUE, nvmax = 50, nbest
```

```
## Reordering variables and trying again:
```

```
best_model <- which.max(summary(regsub)$adjr2)

coefficients <- coef(regsub,best_model)
coefficients <- coefficients[coefficients != 0]
variables <- names(coefficients)[ -1, drop = FALSE]
best_r2 <- max(summary(regsub)$adjr2)
print("Coefficients of the best subset model: ")
```

```
## [1] "Coefficients of the best subset model: "
```

```
coefficients
```

```
## (Intercept)     snp_918    snp_2180    snp_2863    snp_4587    snp_5053
## 12.05710692  0.29846867 -0.18916880  0.72515276 -0.16154194  0.21366132
##    snp_5349    snp_6144    snp_6103
##  0.17055875 -0.02866498  0.09115097
```

```
cat("\nVariables of best subset:",variables)
```

```
##
## Variables of best subset: snp_918 snp_2180 snp_2863 snp_4587 snp_5053 snp_5349 snp_6144 snp_6103
```

```
cat("\n The adjusted r-squared value of the best subset model: ",best_r2)
```

```
##
##  The adjusted r-squared value of the best subset model:  0.4810209
```

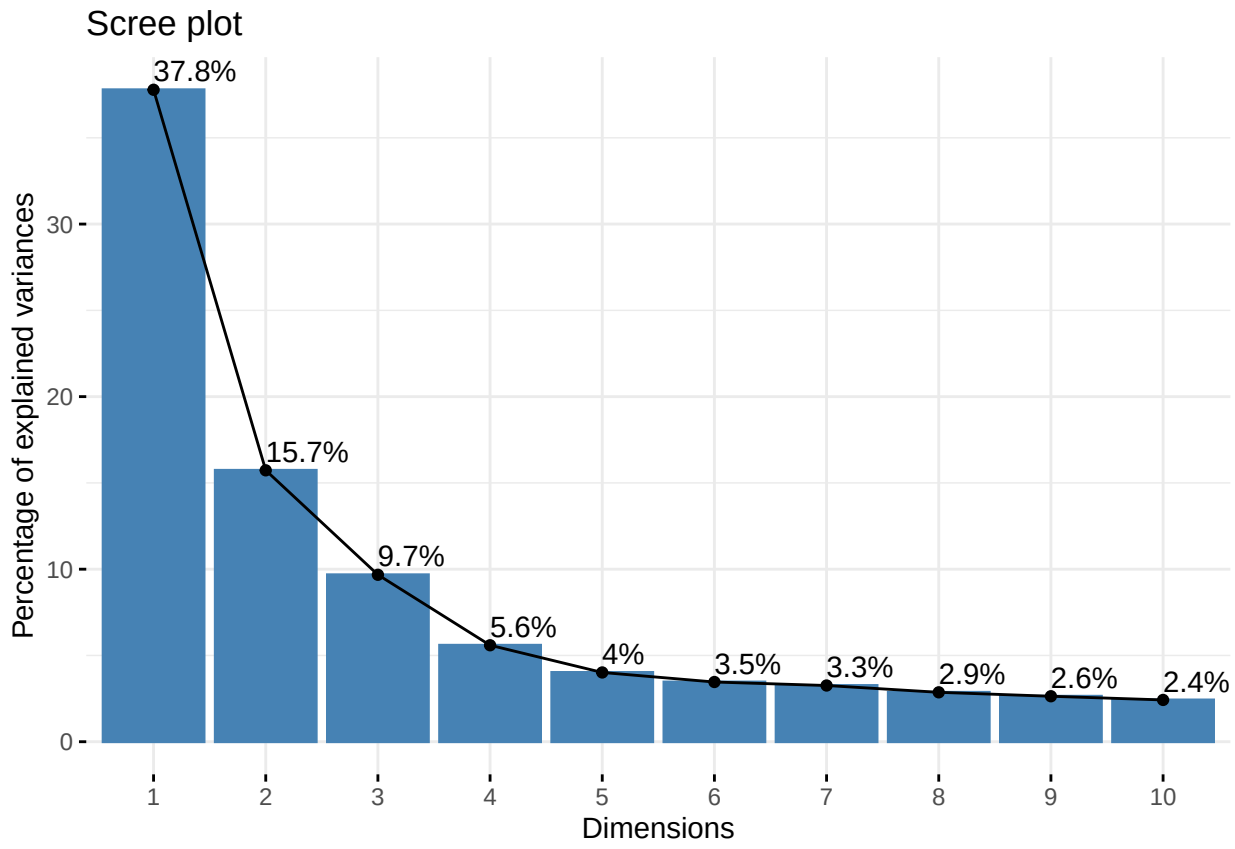
6. (4 marks) Run a PCA on the 50 important variables in `genes_imp`. Report the total (cumulative) variance explained by the first 10 principal components. Plot a scree plot.

```
PCA <- princomp(genes_imp)

print(cumsum(PCA$sdev^2 / sum(PCA$sdev^2))[1:10])
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7     Comp.8
## 0.3778373 0.5351362 0.6319018 0.6878164 0.7279634 0.7625764 0.7951693 0.8238089
##     Comp.9    Comp.10
## 0.8501684 0.8743929
```

```
fviz_screeplot(PCA, addlabels = TRUE)
```



7. (4 marks) Build a linear model of the response variable `Perimeter_Growth` using the first ten PCA dimensions from the previous question, and nothing else. Report the `summary(lm())`. Comment on the difference between this model's adjusted R-squared and how the adjusted r-squared values for the top 10 PCs and the best subsets model are about the same.

```
comp_df = data.frame(Perimeter_Growth = dat_imp$Perimeter_Growth, PCA$scores[,1:10])
pca_lm = lm(dat_imp$Perimeter_Growth ~., data = comp_df)
summary(pca_lm)
```

```
##
## Call:
## lm(formula = dat_imp$Perimeter_Growth ~ ., data = comp_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9381 -0.7364  0.0434  0.6699  4.3059
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.314808   0.060598 236.225  < 2e-16 ***
## Comp.1       0.241417   0.015971  15.116  < 2e-16 ***
## Comp.2       0.179648   0.024752   7.258 2.43e-12 ***
## Comp.3       0.167547   0.031558   5.309 1.93e-07 ***
## Comp.4       0.051339   0.041516   1.237  0.21703
## Comp.5       0.022910   0.048995   0.468  0.64035
## Comp.6      -0.010019   0.052766  -0.190  0.84951
## Comp.7      -0.158170   0.054377  -2.909  0.00385 **
## Comp.8      -0.008527   0.058008  -0.147  0.88321
## Comp.9       0.137445   0.060465   2.273  0.02361 *
## Comp.10     -0.091184   0.063073  -1.446  0.14914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.169 on 361 degrees of freedom
## Multiple R-squared:  0.4752, Adjusted R-squared:  0.4607
## F-statistic: 32.69 on 10 and 361 DF,  p-value: < 2.2e-16
```

```
cat("Adjusted R-squared of linear model using 10 PCA dimension:", summary(pca_lm)$adj.r.squared)
```

```
## Adjusted R-squared of linear model using 10 PCA dimension: 0.4606695
```

```
cat("\nAdjusted R-squared of best subsets regression model:", best_r2)
```

```
##
## Adjusted R-squared of best subsets regression model: 0.4810209
```

Best subset regression aims to maximize the adjusted R-squared value, which represents the proportion of variance in the dependent variable explained by the predictors. Similarly, PCA selects principal components that capture the maximum variance in the data. Thus, both methods strive to optimize the explained variance, resulting in similar adjusted R-squared value.

8. (4 marks) Describe briefly one advantage and one disadvantage of the PCA-based model over the best subsets model. (There are several correct answers, but only the first two will be marked).

One advantage PCA-based models have over best subsets models ivolve its computational efficiency. Best subset regression involves searching through all possible subsets of predictors, which can be computationally expensive and time consuming, especially for large datasets with many predictors. PCA, on the other hand, involves eigenvalue or singular value decomposition, which is computationally efficient.

One disadvantage on the other hand involves its information loss. PCA aims to capture maximum variance in the data, but this does not mean that it is most relevant information for predicting the target variable. Important predictive information may be lost during the dimensionality reduction, resulting in suboptimal model performance.

9. (4 marks) The variance inflation factor of an explanatory variable in a model is a function of how collinear that variable is with the over explanatory variables in the model are. The higher the number, the more collinear and the most the variance estimates of the slopes are being inflated by including that variable. We can find the variable inflation factor with `vif(lm())`, where `vif` is found in the `car` package.

Find the `vif()` of both the PCA-based model and best-subsets model.

Report the VIFs for both models and briefly explain why the PCA-based model has such low inflation factors (1 is the lowest possible).

```
vif(pca_lm)
```

```
##  Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9 Comp.10
##       1       1       1       1       1       1       1       1       1       1
```

```
data_subset <- dat_imp[, c("Perimeter_Growth", variables)]
vif(lm(dat_imp$Perimeter_Growth ~., data = data_subset))
```

```
##  snp_918 snp_2180 snp_2863 snp_4587 snp_5053 snp_5349 snp_6144 snp_6103
## 1.138870 1.333843 1.472024 2.048521 2.523176 1.580688 1.826940 2.610587
```

The PCA-based model has lower inflation factors due to PCs being orthogonal in nature due to it aiming to capture the maximum explained variance. As such, it reduces multicollinearity among the predictors, leading to the lowest Variance Inflation Factors (VIFs). .

On the other hand, a linear model created from the variables from the best subset has higher collinearity due making use of highly correlated predicted giving rise to its higher variance inflation factor scores.