# Project #2

Winter Nguyen

2026-01-27

---

## Problem #1 (55 points)

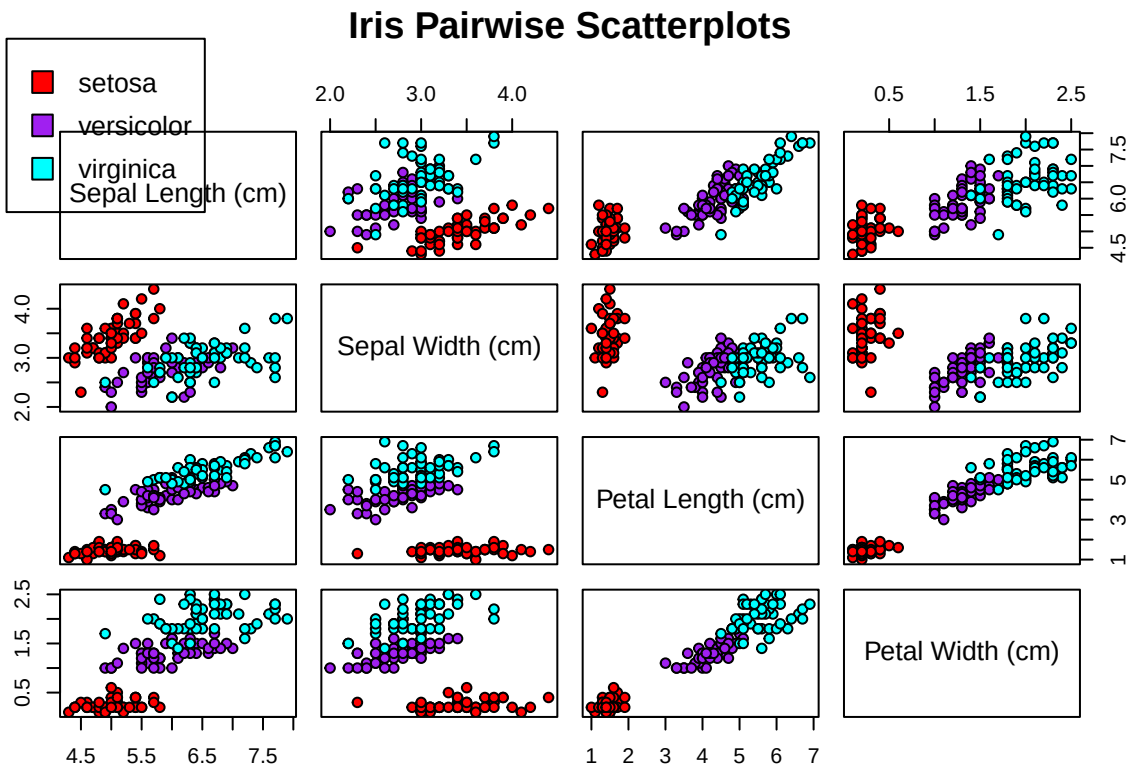
The `iris` data set is built-in in `R`. Start by studying the documentation of the data set, i.e., by entering `?iris` in the console. To familiarize yourselves with the architecture of an iris flower, go to:

US Forest Service

Your next step is exploratory data analysis.

**(10 points)** Which plot would you use to display pairwise associations between different measurements? How do you make sure that the different species are color-coded? Display the plot and write a few sentences about your conclusions.

```
species=iris$Species
species_colors = c("red", "purple", "cyan")[unclass(iris$Species)]
pairs(
  iris[, 1:4],
  main = "Iris Pairwise Scatterplots",
  labels =c("Sepal Length (cm)","Sepal Width (cm)", "Petal Length (cm)","Petal Width (cm)"),
  pch = 21,
  bg  = c("red", "purple", "cyan")[unclass(iris$Species)]
)

legend(
  "topleft",
  legend = levels(iris$Species),
  fill = c("red", "purple", "cyan"),
  cex = 0.8,
  xpd=TRUE
)
```




I used the pairs() function to visualize the pairwise relationships among the measurements. From the plots, Petal Length and Petal Width show a strong positive linear relationship across all three species

Sepal Length and Sepal Width exhibit a moderate positive correlation for Virginica and Versicolor, and this relationship is slightly stronger in Setosa.

Additionally, Sepal Length and Petal Length display a strong positive linear relationship for Versicolor and Virginica, but this relationship is not evident in Setosa. Similar patterns are observed for Sepal Length vs. Petal Width and Sepal Width vs. Petal Length, where Setosa differs from the other two species.

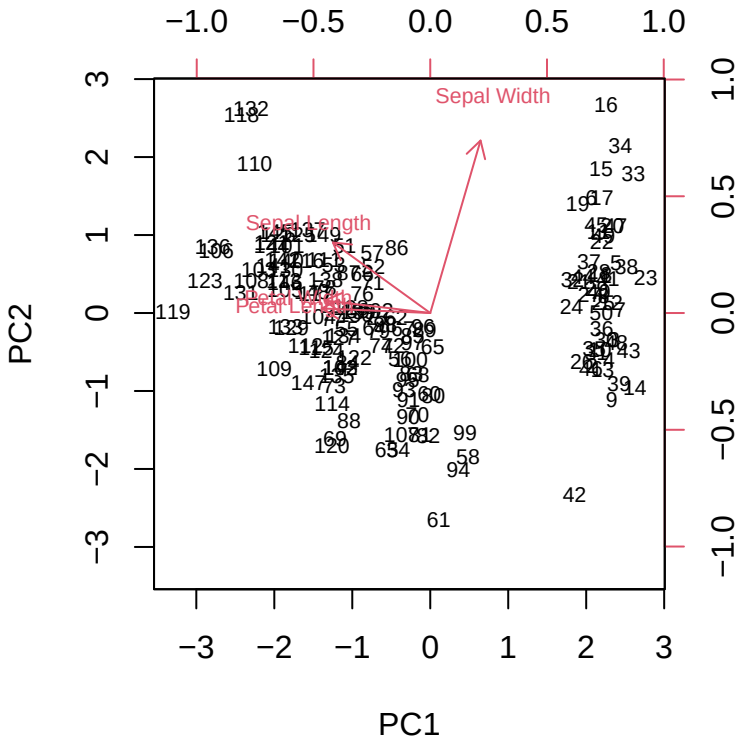
Overall, there is a positive linear relationship among all the measurements, with petal dimensions showing the strongest correlations for Versicolor and Virginica. The variables behave differently in Setosa, where the relationships are generally weaker or follow a distinct pattern, but still strong between Pedal Length and Petal Width.

### Principal Component Analysis (PCA)

**(20 points)** Perform the PCA on the explanatory components of the above data, provide the report, and the relevant plots.

```
pr.out=prcomp(iris[, 1:4], scale=TRUE) # exclude Species because it is category

#I just change the rotation based on my preference
pr.out$rotation = -pr.out$rotation
pr.out$x = -pr.out$x
```

```
#I changed the columns name, my preference
rownames(pr.out$rotation) =c("Sepal Length","Sepal Width", "Petal Length","Petal Width")
biplot(pr.out, scale = 0,
       cex=0.7,
      bg  = c("red", "purple", "cyan")[unclass(iris$Species)])
```

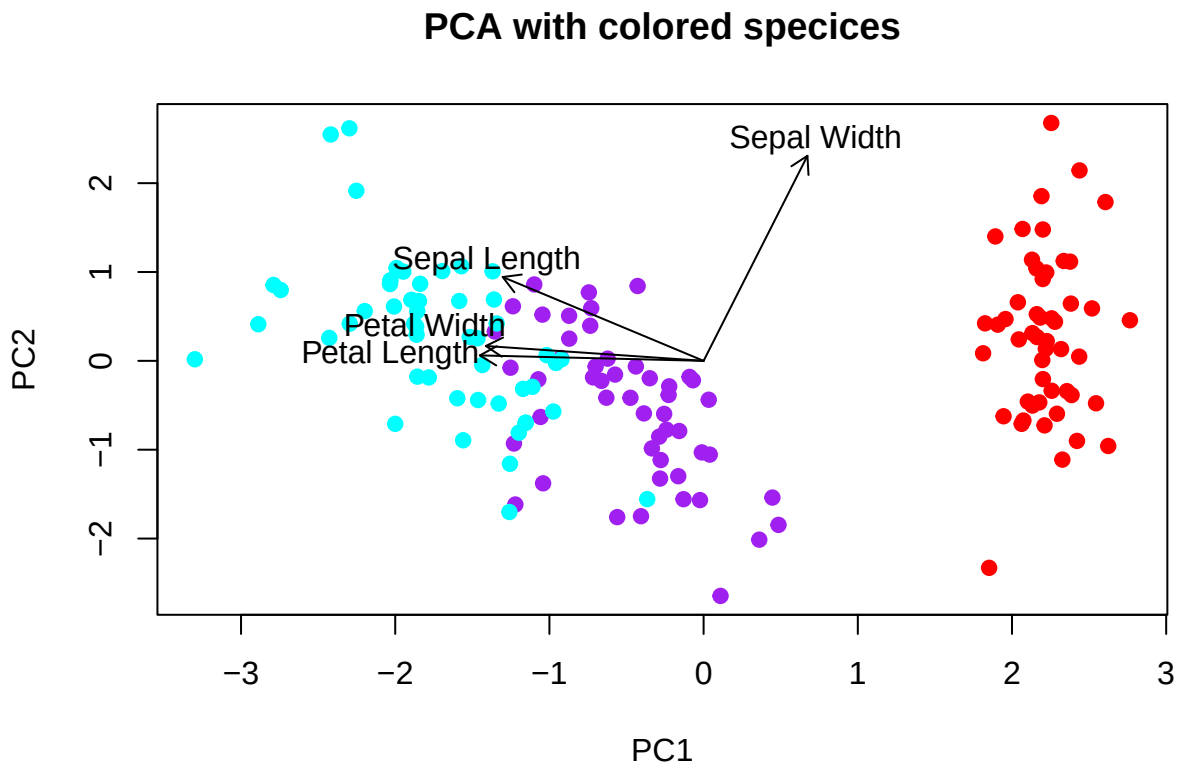



```
summary(pr.out)
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Below I provide another PCA version to see more information.

```
loadings=pr.out$rotation
scores=pr.out$x
plot(scores[,1], scores[,2], #PC1 and PC2
     col=c("red", "purple", "cyan")[unclass(iris$Species)],
     pch=19,
     xlab="PC1",
     ylab="PC2",
     main="PCA with colored specices")
```

```r
# adding vector for each variables
arrows(0, 0, loadings[,1]*2.5, loadings[,2]*2.5, length = 0.1, col = "black")
#labeling vectors
text(loadings[1,1]*2.7, loadings[1,2]*3.0, labels = rownames(loadings)[1], col = "black")
text(loadings[2,1]*2.7,loadings[2,2]*2.7,labels=rownames(loadings)[2],col="black")
text(loadings[3,1]*3.5,loadings[3,2]*2.7,labels=rownames(loadings)[3],col="black")
text(loadings[4,1]*3.2,loadings[4,2]*6,labels=rownames(loadings)[4],col="black")
```

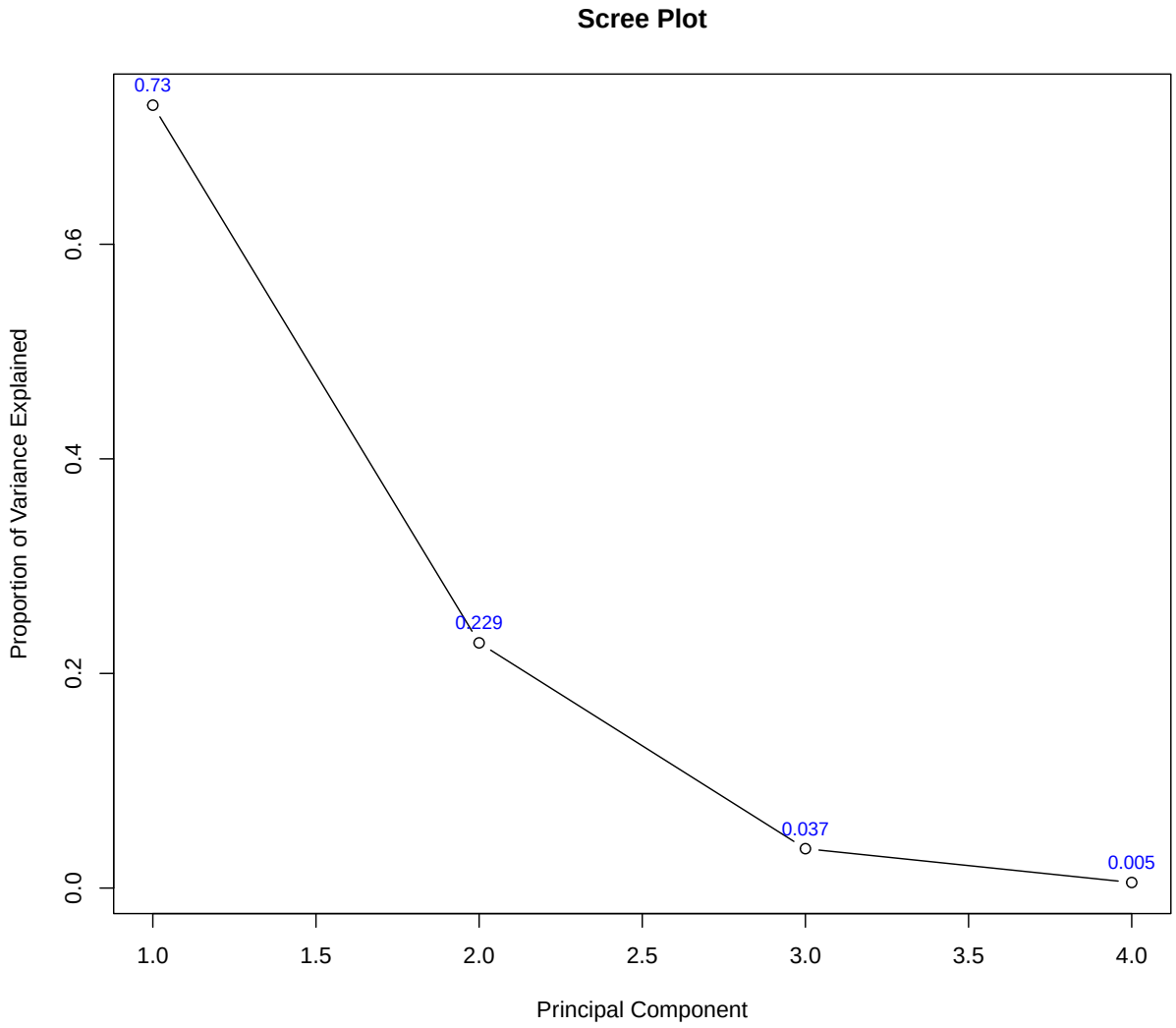



From the biplot, PC1 (horizontal) primarily captures overall flower size.Petal Length, and Petal Width all point strongly in the same (leftward) direction, Sepal Length also point to left but not as strong as Pedal variables, indicating that these three measurements are positively correlated and together represent the main source of variation among samples. PC2 (vertical) is dominated by Sepal Width, which points in the opposite (rightward) direction, suggesting that flowers with larger petals and longer sepals tend to have narrower sepals. The opposing directions of Sepal Width and the other three features indicate a negative correlation between sepal width and the overall flower size traits.

Notice that, Setosa's cluster is distinct from the other two species, suggesting the size gap, Setosa is significantly smaller than the other two species.

```r
# I calculate the Proportion of Variance Explained by Principal Component.
pr.var = pr.out$sdev^2
pve=pr.var/sum(pr.var)

#plotting the proportion of variance explained vs principal component
plot(pve, type = "b",
```

```
     xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     main = "Scree Plot")

text(x = 1:length(pve),
     y = pve,
     labels = round(pve, 3),
     pos = 3,    # position: 3 = above
     cex = 0.8, # text size
     col = "blue")
```



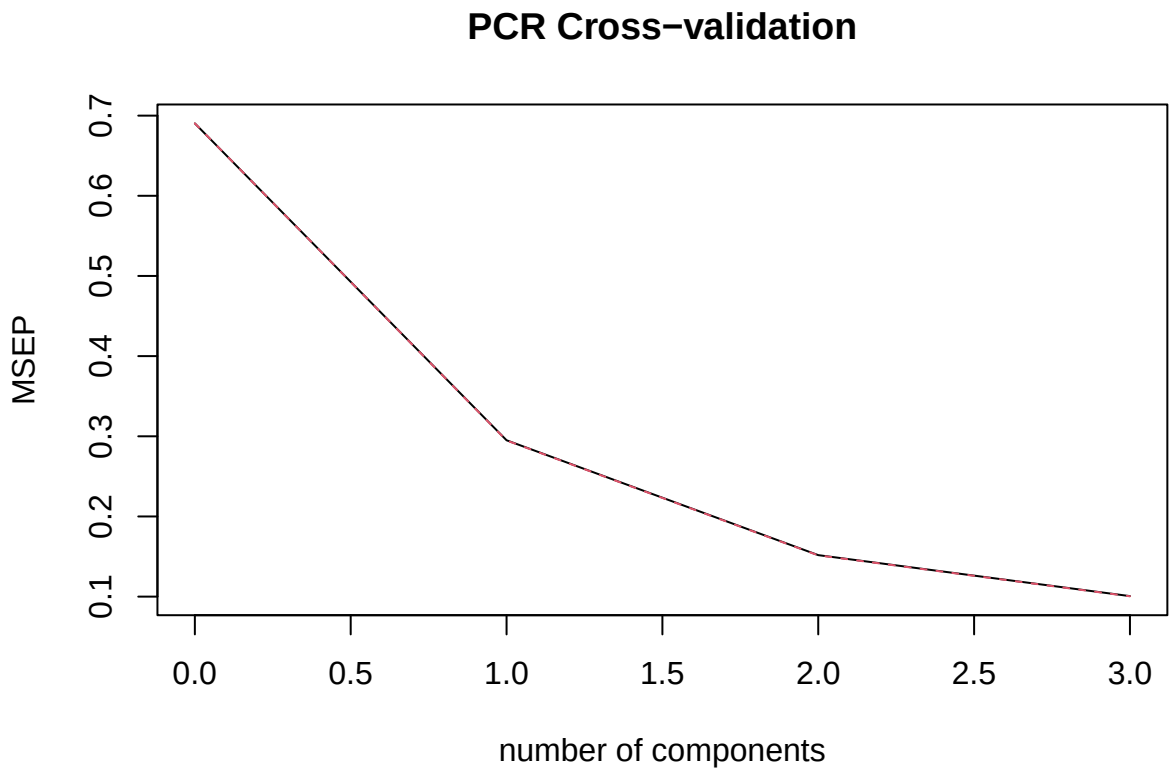

The plot confirms the idea that first component explain the most variance. The proportion of variance explained (PVE) by each principal component is approximately 72.96%, 22.85%, 3.67%, and 0.52% for PC1 through PC4, respectively, are consistent with the summary table. Together, the first two principal components explain about 95.8% of the total variance, indicating that most of the information in the data set can be effectively represented in two dimensions.

### Principal Components Regression (PCR)

Your next task is to predict `Sepal.Length` from the other variables in the `iris` dataset.

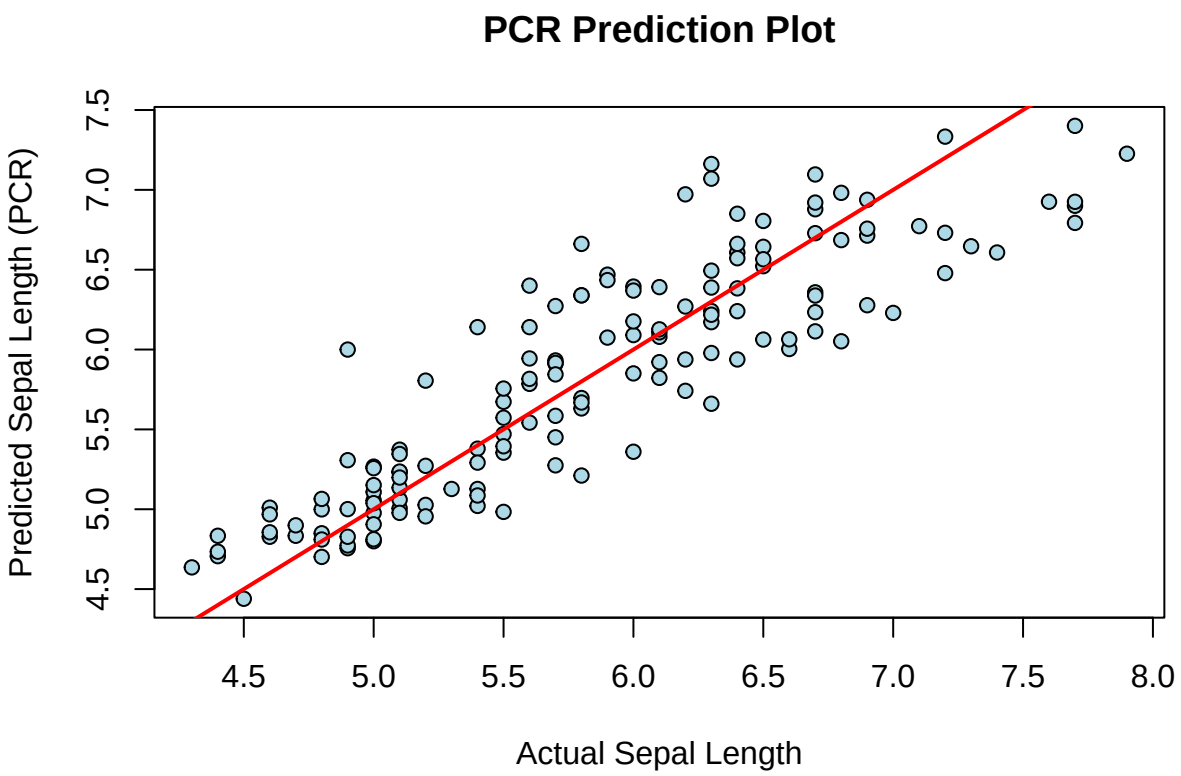
**(15 points)** Run the PCR, provide an explanation for the output, and display the relevant plots (both validation and prediction).

```
library(pls)
##
## Attaching package: 'pls'
## The following object is masked from 'package:stats':
##
##     loadings
set.seed(1)
pcr.out= pcr(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width,
                 data = iris,
                 scale = TRUE,   # standardize variables
                 validation = "CV")

#plot the validation
validationplot(pcr.out, val.type = "MSEP", main = "PCR Cross-validation")
```




```
#plot the predictions
predictions= predict(pcr.out, ncomp = 2, newdata = iris)
plot(iris$Sepal.Length, predictions,
     xlab = "Actual Sepal Length",
```

```
     ylab = "Predicted Sepal Length (PCR)",
     main = "PCR Prediction Plot",
     pch = 21, bg = "lightblue")
abline(0, 1, col = "red", lwd = 2)
```



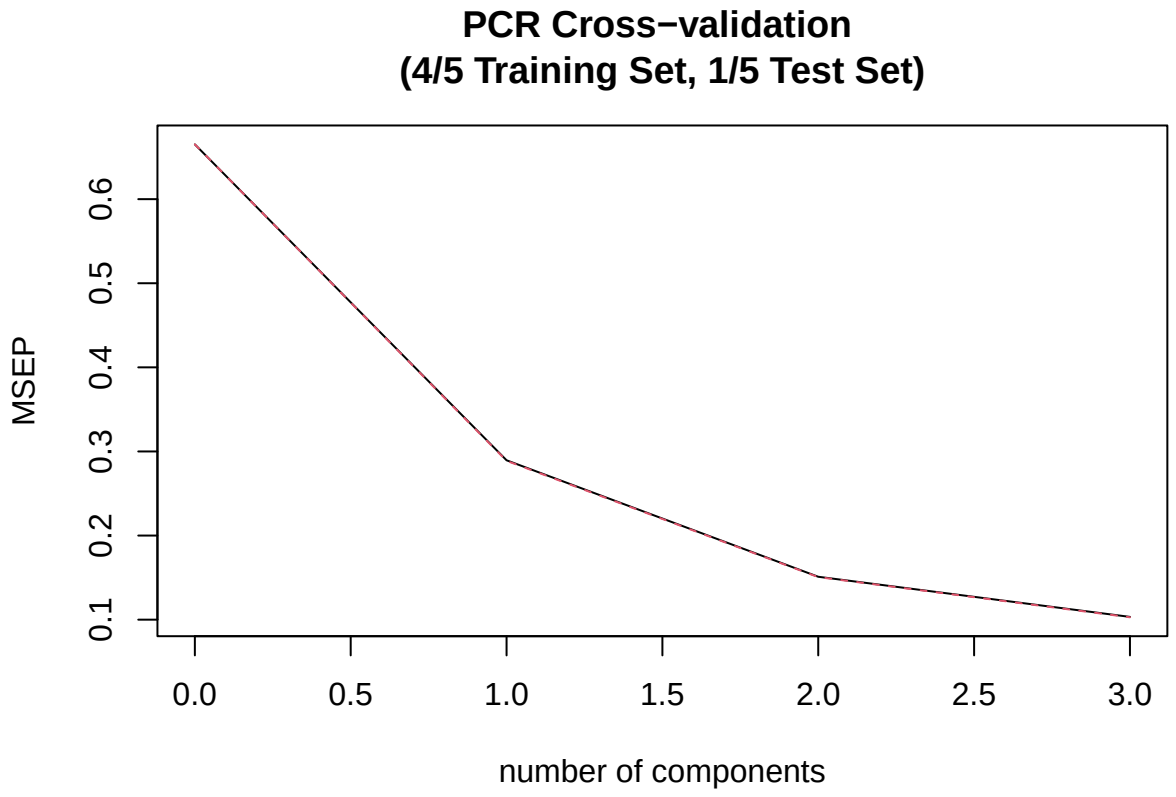

```
summary(pcr.out)
## Data:    X dimension: 150 3
##  Y dimension: 150 1
## Fit method: svdpc
## Number of components considered: 3
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps
## CV          0.8308   0.5432   0.3896   0.3172
## adjCV       0.8308   0.5429   0.3892   0.3169
##
## TRAINING: % variance explained
##                1 comps  2 comps  3 comps
## X                74.05    98.84   100.00
## Sepal.Length     57.97    78.47    85.86
```

The validation plot tells us that there is a major drop in MSE at the first component (from 0.7 to 0.3, so 0.4 in value), this means the PCA1 captures major variation in predictors. Then it continues to drop 0.15

in value at PC2, and then continues to drop at PCA3 but not too much (0.05 in value). So this makes sense since according to the PCR summary, the first two principal components explained 98.84% of the variance. However, This model using the same data to train and to test the model, meaning that the model has already seen those exact data points during training, so the predictions will be unrealistically good. This introduces bias and does not reflect real-world predictive performance.

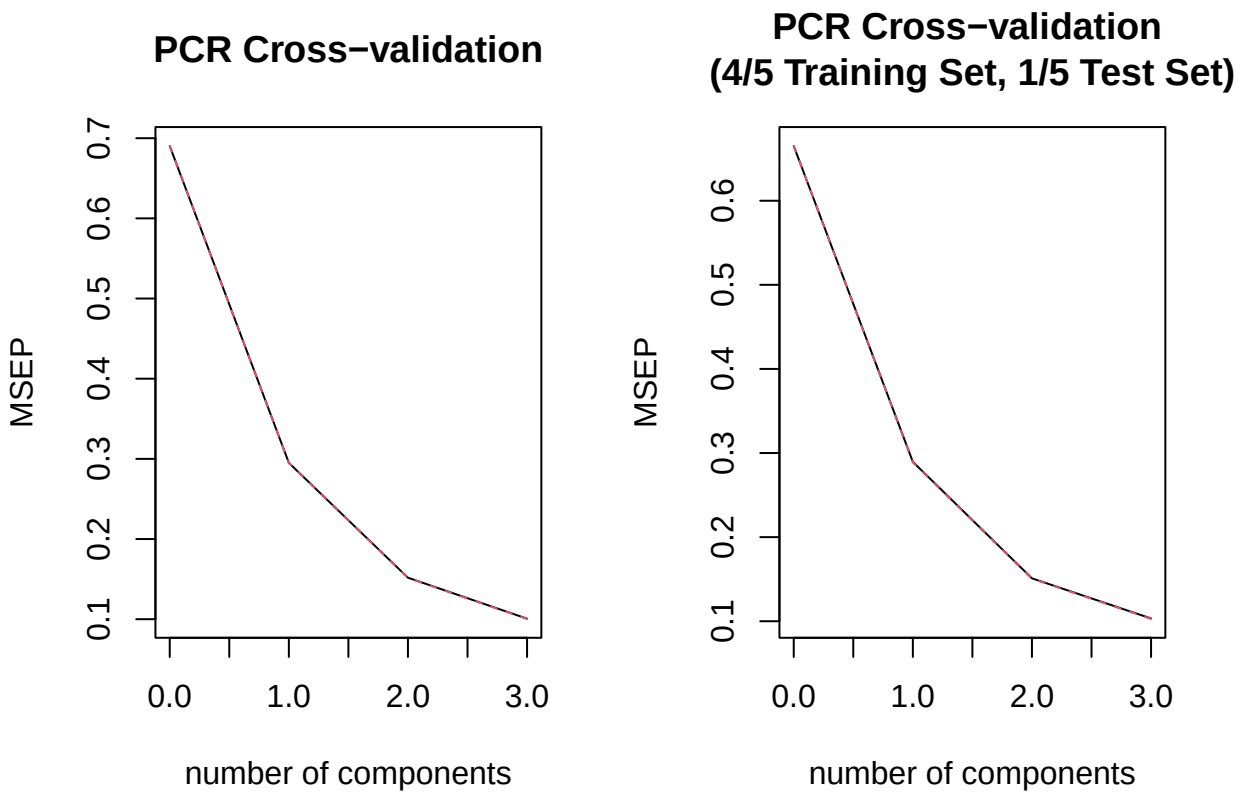
**(10 points)** Split your dataset into training (4/5 of the data) and testing (1/5 of the data). Provide the mean squared error and an appropriate plot.

```
set.seed(200)

#spliting the data
training_index=sample(1:nrow(iris), size=(4/5)*nrow(iris))
training_set=iris[training_index,]
test_set=iris[-training_index,]

#performing pcr on splitting data
pcr_out_training=pcr(Sepal.Length~Sepal.Width + Petal.Length + Petal.Width, data=training_set, scale=TRU
validationplot(pcr_out_training,
               val.type = "MSEP",
               main = "PCR Cross-validation \n(4/5 Training Set, 1/5 Test Set)")
```




```
#ploting side by side the validation plot of full data set and splitted data set
par(mfrow = c(1, 2))
validationplot(pcr.out,
               val.type = "MSEP",
```

```
               main = "PCR Cross-validation")
validationplot(pcr_out_training,
               val.type = "MSEP",
               main = "PCR Cross-validation \n(4/5 Training Set, 1/5 Test Set)")
```



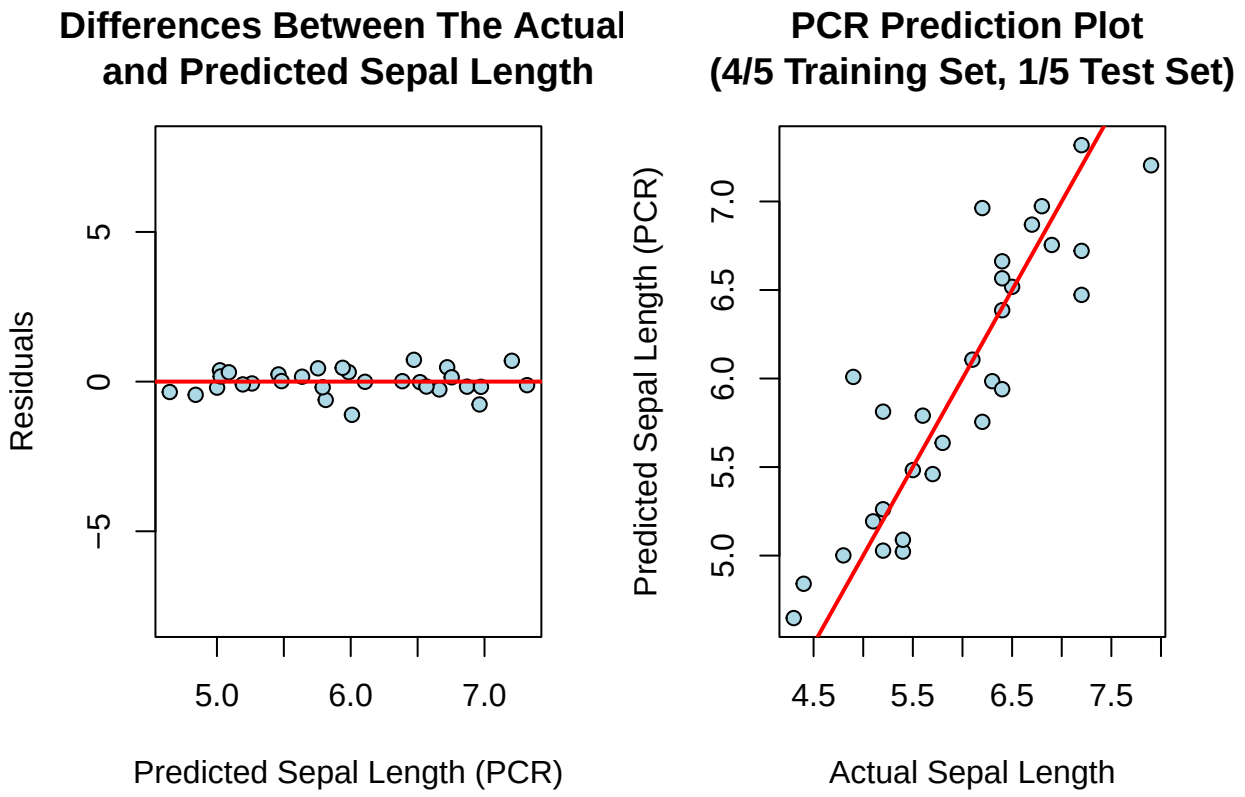

```
par(mfrow = c(1, 1))
```

The full dataset PCR model and the model trained on the 4/5 training subset have nearly identical cross-validation MSEP curves. This indicates that the optimal number of components and the model's predictive performance (as reflected by the MSEP) are stable across different data splits. The fact that the smaller 4/5 training set achieves a similar minimum MSEP to the full dataset suggests that the training set is sufficiently large and that adding more data would not substantially improve model accuracy. Since in both models, the MSEP only decreases (no increase after certain component) showing that there is no evidence of overfitting.

```
#Using model on training set to predict sepal length on test set
predictions_on_test_set= predict(pcr_out_training, ncomp = 2, newdata =test_set )
mse = mean((test_set$Sepal.Length - predictions_on_test_set)^2)
bias= mean(test_set$Sepal.Length - predictions_on_test_set) # very close to 0 so
                                                           # I assume it is unbias
range_sepal_length=range(iris$Sepal.Length)
```

The MSE is 0.163. I calculated the bias which is the expected value of residuals and I got -0.005631, very close to 0, so I assume the model is unbias. Then the standard deviation calculated from MSE is 0.404 that means on average, my predicted sepal lengths are off by 0.404 cm which is small compared to the range of sepal lengths (4.3, 7.9) cm

```r
#plotting side by side the preidiction plot and also the residuals plot,
#making it easier to interpret.
par(mfrow = c(1, 2))
residuals_pred_real=test_set$Sepal.Length-predictions_on_test_set
plot(predictions_on_test_set,residuals_pred_real,
     xlab = "Predicted Sepal Length (PCR)",
     ylab = "Residuals",
     main = "Differences Between The Actual \nand Predicted Sepal Length",
     pch = 21, bg = "lightblue",
     ylim = c(-max(test_set$Sepal.Length),max(test_set$Sepal.Length))) # I change the
                            # range to illustrate how big or small residuals really is
                            # compared to its possible range
abline(h=0, col="red", lwd=2)

plot(test_set$Sepal.Length, predictions_on_test_set,
     xlab = "Actual Sepal Length",
     ylab = "Predicted Sepal Length (PCR)",
     main = "PCR Prediction Plot \n(4/5 Training Set, 1/5 Test Set)",
     pch = 21, bg = "lightblue")
abline(0, 1, col = "red", lwd = 2)
```




```r
par(mfrow = c(1, 1))
```

There is a strong linear relationship between predicted and actual sepal lengths which is what we should expect. In the residual plot, we can see the residuals scatter around the zero line, and relatively small

compared to the range of possible values of residual.

```
#this part is to check on my interpretation more accurately
lm_pred_real=lm(predictions_on_test_set~test_set$Sepal.Length)
summary(lm_pred_real)
##
## Call:
## lm(formula = predictions_on_test_set ~ test_set$Sepal.Length)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51683 -0.25330 -0.06199  0.22718  0.85416
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.38978    0.45475   3.056  0.00489 **
## test_set$Sepal.Length  0.76841    0.07528  10.207 6.11e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3612 on 28 degrees of freedom
## Multiple R-squared:  0.7882, Adjusted R-squared:  0.7806
## F-statistic: 104.2 on 1 and 28 DF,  p-value: 6.114e-11
```

With p-value is less than 0.05, and the coefficient correlation R-squared = 0.78 which is a strong but not perfect correlation, confirming my previous interpretation. Also, the residual standard error is 0.3612 which is consistent with my RSE is 0.404 (calculated RSE from MSE). The small difference maybe due to degree of freedom adjustment.

## Problem #2 (20+5+10+10=45 points)

Solve **Problem 3.7.15** (page 128) from the textbook.

*Hint:* The command `lapply` could be useful.

This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

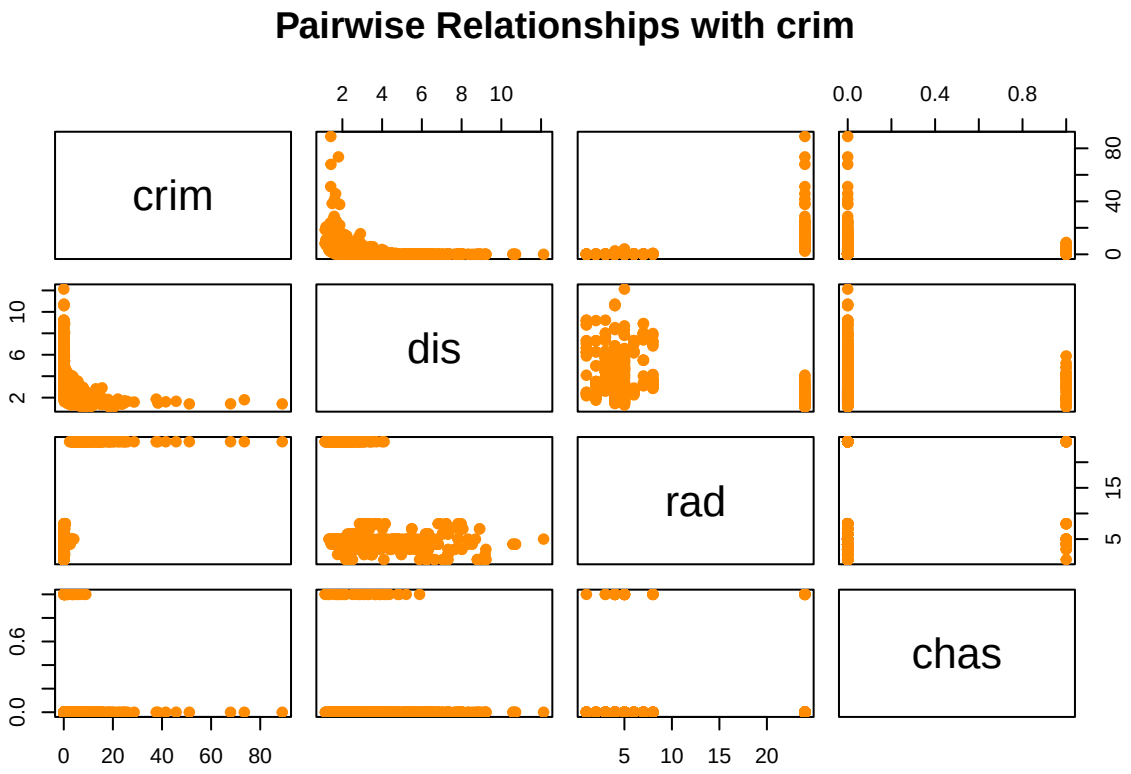
CRIM - per capita crime rate by town ZN - proportion of residential land zoned for lots over 25,000 sq.ft. INDUS - proportion of non-retail business acres per town. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise) NOX - nitric oxides concentration (parts per 10 million) RM - average number of rooms per dwelling AGE - proportion of owner-occupied units built prior to 1940 DIS - weighted distances to five Boston employment centres RAD - index of accessibility to radial highways TAX - full-value property-tax rate per $10,000 PTRATIO - pupil-teacher ratio by town BLACK - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town LSTAT - % lower status of the population MEDV - Median value of owner-occupied homes in $1000's

(a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions

```r
# we predict per capital crime rate
# so I extract all other predictors except "crim"
library(MASS)
library(car)
## Loading required package: carData
predictors=Boston[,2:length(Boston)] # "crim" index is 1, so I exclude it
crim_predictors= lapply(predictors, function(x) lm(Boston$crim ~ x)) # perform simple
results = data.frame(
  Predictor = names(predictors),
  Coefficient = sapply(crim_predictors, function(y) coef(y)[2]),
  P_value = sapply(crim_predictors, function(y) summary(y)$coefficients[2, 4])
)
results
##           Predictor Coefficient      P_value
## zn.x             zn -0.07393498 5.506472e-06
## indus.x       indus  0.50977633 1.450349e-21
## chas.x         chas -1.89277655 2.094345e-01
## nox.x           nox 31.24853120 3.751739e-23
## rm.x             rm -2.68405122 6.346703e-07
## age.x           age  0.10778623 2.854869e-16
## dis.x           dis -1.55090168 8.519949e-19
## rad.x           rad  0.61791093 2.693844e-56
## tax.x           tax  0.02974225 2.357127e-47
## ptratio.x   ptratio  1.15198279 2.942922e-11
## black.x       black -0.03627964 2.487274e-19
## lstat.x       lstat  0.54880478 2.654277e-27
## medv.x         medv -0.36315992 1.173987e-19
```

Based on the results, all the p-values except for "chas" are under 0.05. So except "chas", there are significant associations between those predictors and "crim".

```r
#performing linear regression on all predictors except "chas"
# Make sure predictors are correctly defined by name
chas = predictors$chas
dis  = predictors$dis
rad  = predictors$rad

# Fit models
crim_vs_chas = lm(Boston$crim ~ chas)
crim_vs_dis  = lm(Boston$crim ~ dis)
crim_vs_rad  = lm(Boston$crim ~ rad)


pairs(Boston[, c("crim", "dis", "rad", "chas")],
      main = "Pairwise Relationships with crim",
      col = "darkorange", pch = 19)
```




From the pairwise relationships, "dis" shows a strong negative nonlinear association with crime rate, suggesting that areas farther from employment centers tend to have lower crime. In contrast, rad has a positive association: neighborhoods with higher highway accessibility exhibit higher crime levels. The chas variable, representing proximity to the Charles River, shows no clear pattern, consistent with its lack of statistical significance. However, these plots are influenced by outliers and variable types: crim is heavily right-skewed, "rad" and "chas" are discrete, and several relationships are nonlinear. So, it is better to use polynomial models and using boxplots for categorical predictors, or we can transform crim to high and low crim and apply logistic regression.

```
summary(crim_vs_chas)
##
## Call:
## lm(formula = Boston$crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

The relationship between crime rate and proximity to the Charles River "chas" is weak and not statistically significant (p = 0.209). Neighborhoods near the river tend to have slightly lower crime on average, but the difference is small and not meaningful. Since there is no relationship, I will skip the boxplot.

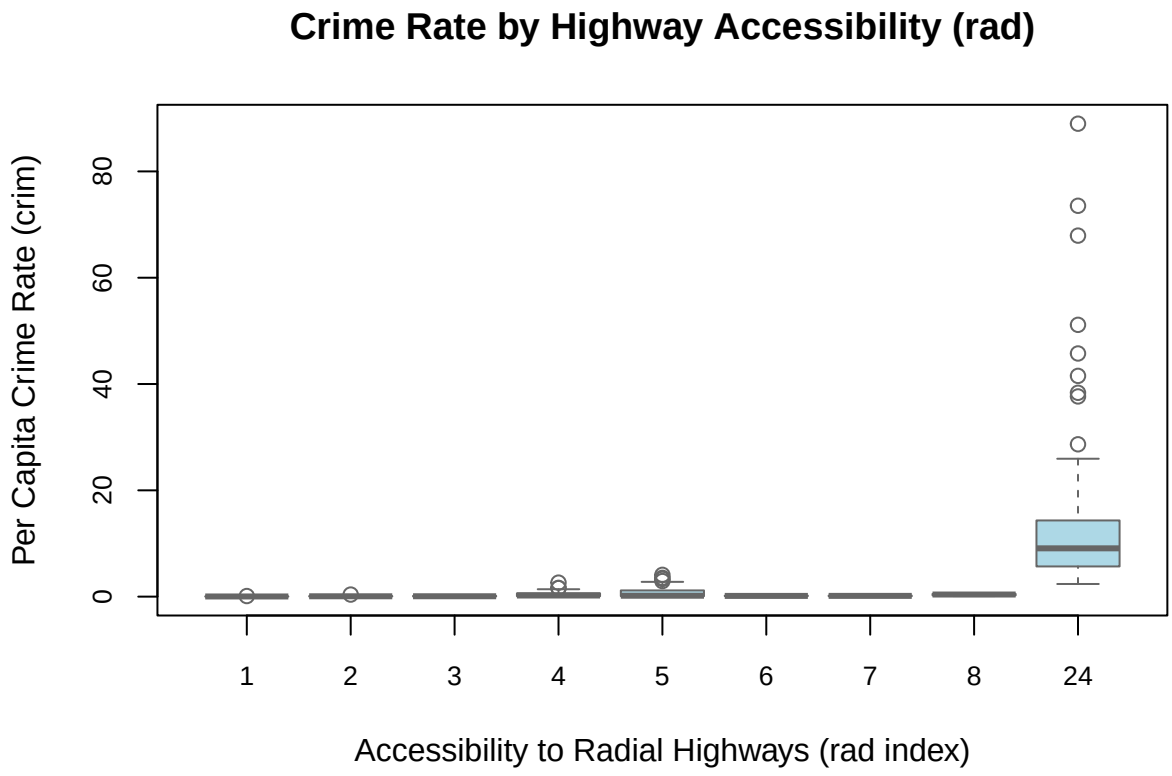
```
summary(crim_vs_dis)
##
## Call:
## lm(formula = Boston$crim ~ dis)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

There is a strong and statistically significant negative relationship between crime rate and distance to employment centers "dis". Areas farther from the city (larger "dis") tend to have substantially lower crime rates.

```
summary(crim_vs_rad)
##
## Call:
## lm(formula = Boston$crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

The accessibility to radial highways "rad" has a strong positive and statistically significant association with crime rate. Neighborhoods with greater access to highways tend to experience much higher crime levels, and rad alone explains nearly 40% of the variability in crime. Below is the box plot for each area.

```r
boxplot(crim ~ as.factor(rad),
        data = Boston,
        main = "Crime Rate by Highway Accessibility (rad)",
        xlab = "Accessibility to Radial Highways (rad index)",
        ylab = "Per Capita Crime Rate (crim)",
        col = "lightblue",
        border = "gray40",
        cex.axis = 0.8)
```




(b) Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$

```r
# perform multiple linear regression with reponse to be criminal rate per capita
# the rest are predictors
mlr=lm(crim~., data=Boston)
summary(mlr)
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

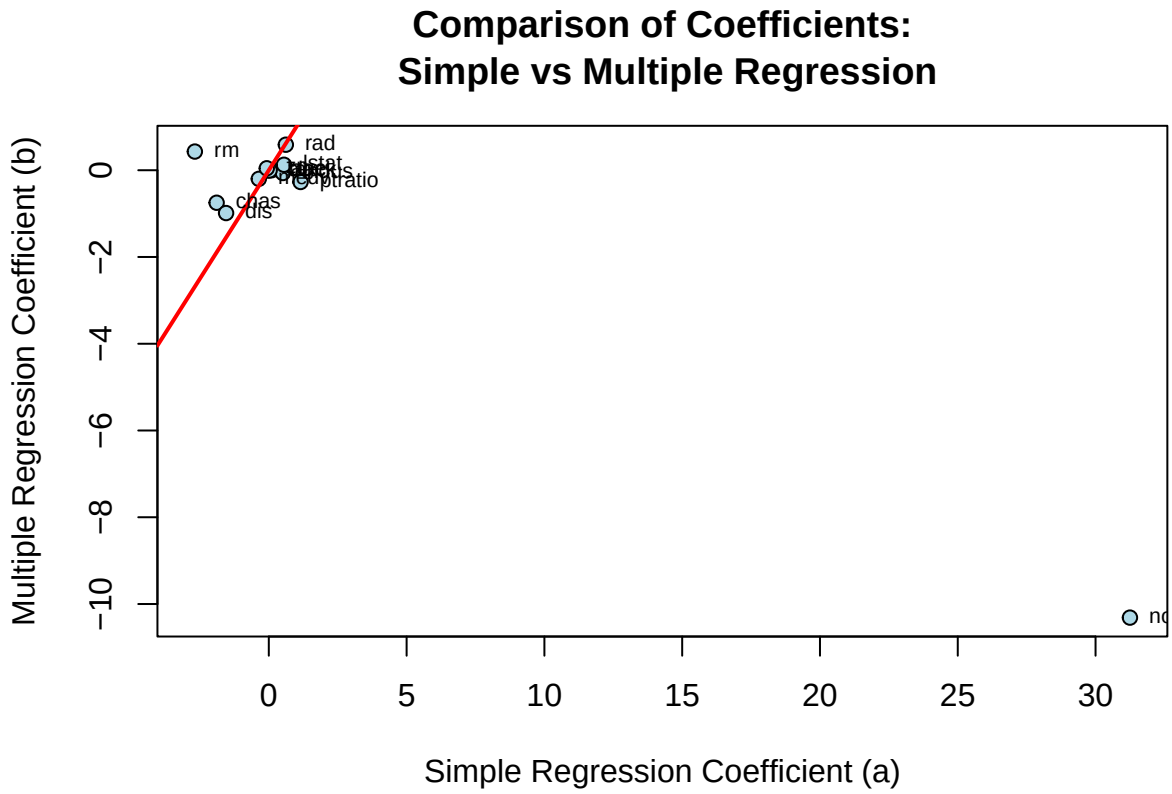
Overall, the model demonstrates a moderate fit, explaining about 45% of the variation in crime rates among Boston neighborhoods, the p value is less than 0.05, so there are significant association between response and predictors ,however, this doesn't apply for each predictor. Based on the summary, there are associations between predictors and "crim" only the following predictors: "zn" , "dis", "rad", "black", and "medv" (we reject the null hypothesis for these predictors).

(c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

In b) we eliminate most of the predictors ("indus", "chas", "nox", "rm", "age", "tax", "ptratio", "lstat") compared to just "chas" in a). This may result from correlation between predictors - multicollinearity. So, only "zn", "dis", "rad", "black" have significant associations with "crim".

```
multi_coef= coef(mlr)[-1]  # remove intercept
multi_results= data.frame(
  Predictor= names(multi_coef),
  Multiple_Coefficient = multi_coef)
comparison= merge(results, multi_results, by = "Predictor")
head(comparison)
##   Predictor Coefficient      P_value Multiple_Coefficient
## 1       age  0.10778623 2.854869e-16          0.001451643
## 2     black -0.03627964 2.487274e-19         -0.007537505
## 3      chas -1.89277655 2.094345e-01         -0.749133611
## 4       dis -1.55090168 8.519949e-19         -0.987175726
## 5     indus  0.50977633 1.450349e-21         -0.063854824
## 6     lstat  0.54880478 2.654277e-27          0.126211376
#plotting
plot(comparison$Coefficient, comparison$Multiple_Coefficient,
```

```
     xlab = "Simple Regression Coefficient (a)",
     ylab = "Multiple Regression Coefficient (b)",
     main = "Comparison of Coefficients:\n Simple vs Multiple Regression",
     pch = 21, bg = "lightblue")
abline(0, 1, col = "red", lwd = 2)  # line of equality
text(comparison$Coefficient, comparison$Multiple_Coefficient,
     labels = comparison$Predictor, pos = 4, cex = 0.7)
```




```
vif(mlr)
##       zn    indus     chas      nox       rm      age      dis      rad
## 2.325094 3.987753 1.094326 4.551563 2.258113 3.100801 4.289041 7.158834
##      tax  ptratio    black    lstat     medv
## 9.195495 1.984489 1.369741 3.561476 3.772856
```

The comparison plot between simple and multiple regression coefficients shows that coefficients change substantially in magnitude and even direction (namely "nox"). This suggests strong multicollinearity in the Boston dataset. Some predictors, like nox and indus, are highly correlated with others, leading to inflated coefficients in the simple models. Predictors close to the diagonal line, "rad" for example, maintain similar effects across both models, implying independent predictors.

To assess multicollinearity among predictors, the Variance Inflation Factor (VIF) was computed for all variables in the multiple linear regression model. Most predictors had VIF values below 5, indicating acceptable levels of correlation. However, two variables "rad" (VIF = 7.16) and "tax" (VIF = 9.20) exhibited high multicollinearity. This suggests that a large portion of their variance can be explained by other predictors, meaning they provide overlapping information rather than unique contributions to the model.

Both rad and tax describe aspects of urban infrastructure:

rad measures accessibility to radial highways, and

tax reflects property tax rates that are typically higher in densely urbanized areas.

Because these two variables are highly correlated with each other (correlation approximately 0.9), including both can inflate standard errors and make coefficient estimates unstable.

To address this issue, "tax" should be from the model since it is less interpretable in the context of crime prediction, while "rad" was retained as it provides a clearer physical interpretation of accessibility and urbanization.

```
new_data=Boston[, -10] #tax is 10, eliminate tax
new_mlr= lm(crim~., data=new_data)
vif(new_mlr)
##       zn    indus     chas      nox       rm      age      dis      rad
## 2.193080 3.226111 1.083922 4.542222 2.257707 3.098035 4.286741 2.361408
##  ptratio    black    lstat     medv
## 1.982986 1.369740 3.539632 3.695636
```

After removing tax, all remaining predictors had VIF values below 5, indicating that multicollinearity was no longer a concern and the model's estimates became more stable and reliable.

(d) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form

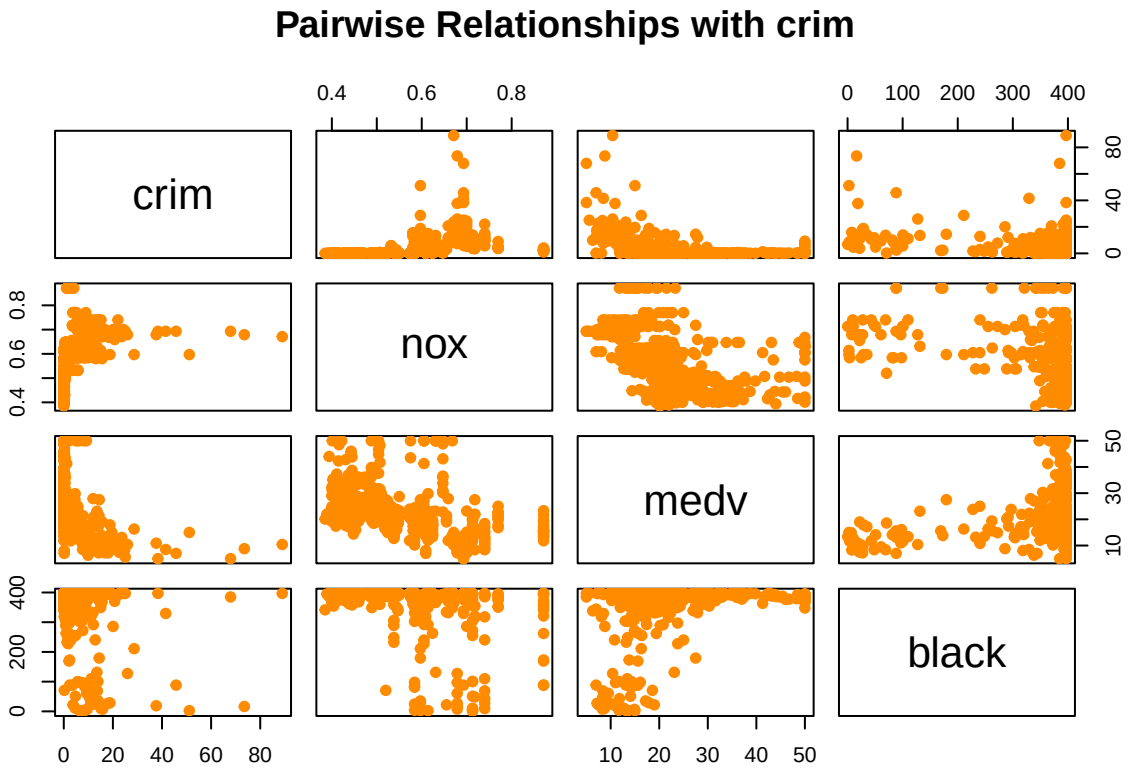
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

```
# create a data frame for nonlinear-model,
# containing predictors name, its quadratic p-value and cubic p-value
nonlinear_results = data.frame(Predictor = names(predictors),
                               X2 = NA,
                               X3 = NA)
for (i in 1:length(predictors)) {
  x = predictors[[i]]
  model = lm(Boston$crim ~ x + I(x^2) + I(x^3))
  coefs = summary(model)$coefficients
  if (nrow(coefs) >= 4) {  # check that cubic term exists
    nonlinear_results$X2[i] = coefs[3, 4] #[3,4] p-value for quadratic term
    nonlinear_results$X3[i] = coefs[4, 4] #[4,4] p-value for cubic term
  }
}
nonlinear_results
##    Predictor           X2           X3
## 1         zn 9.375050e-02 2.295386e-01
## 2      indus 3.420187e-10 1.196405e-12
## 3       chas           NA           NA
## 4        nox 6.811300e-15 6.961110e-16
## 5         rm 3.641094e-01 5.085751e-01
## 6        age 4.737733e-02 6.679915e-03
## 7        dis 4.941214e-12 1.088832e-08
## 8        rad 6.130099e-01 4.823138e-01
## 9        tax 1.374682e-01 2.438507e-01
## 10   ptratio 4.119552e-03 6.300514e-03
```

```
## 11      black 4.741751e-01 5.436172e-01
## 12      lstat 6.458736e-02 1.298906e-01
## 13       medv 3.260523e-18 1.046510e-12
```

Based on the results, many predictors have non-linear associations. For adding the cubic terms, the following predictors : “indus”, “nox”, “age”, “dis”, ptratio, “medv” became significant. When only quadratic terms were added, the predictors: “zn”, “rm”, “rad”, “tax”, “lstat” were significant but lost significance after including the cubic terms. Only “black” has no non-linear relationship, as adding quadratic or cubic terms did not make its association significant.

```
nox = predictors$nox
black  = predictors$black
medv  = predictors$medv

# Fit models
crim_vs_black = lm(Boston$crim ~ black)
crim_vs_nox  = lm(Boston$crim ~ nox)
crim_vs_medv  = lm(Boston$crim ~ rad)


pairs(Boston[, c("crim", "nox", "medv", "black")],
      main = "Pairwise Relationships with crim",
      col = "darkorange", pch = 19)
```




The predictors “nox”, “medv”, and “black” were chosen to represent environmental, economic, and demographic factors related to crime, also they have different relationship with “crim”. The plot shows that

"nox" has a non-linear positive relationship with crime, indicating higher crime in more urbanized areas. "medv" is non-linear strongly negatively related to crime, suggesting wealthier neighborhoods experience less crime. The relationship between "black" and crime appears weaker and less consistent (a U-shape), possibly reflecting more complex social factors.