

Econ 573: Problem Set 2 - Part 2

```
Auto = read.csv("C:/Users/mateo/OneDrive - University of North Carolina at Chapel Hill/Courses/Spring 2025/Econ 573/Auto.csv")
```

Question 8

(a) Use the `lm()` function to perform a simple linear regression with `mpg` as the response and `horsepower` as the predictor. Use the `summary()` function to print the results. Comment on the output.

```
Auto$horsepower <- as.numeric(as.character(Auto$horsepower)) #Convert the horsepower variable from categorical to numeric
```

```
## Warning: NAs introduced por coerción
```

```
lmpg = lm(mpg~horsepower, data = Auto) #Perform the simple linear regression
summary(lmpg)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.935861    0.717499   55.66  <2e-16 ***
## horsepower   -0.157845    0.006446  -24.49  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

There seems to be a small negative relationship between the amount of miles per gallon for a car and its horsepower. From its p-value, which is very close to 0, we can infer strong evidence for this relationship such that it is statistically significant. The predicted coefficient for horsepower, $\beta_1 = -0.157845$, tells us that every one-unit increase in horsepower decreases mpg by -0.157845.

```
horsepower98 = data.frame(horsepower = 98)
predict(lmpg, horsepower98, interval = "confidence") #Average mpg for cars with 98 horsepower
```

```
##          fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```
predict(lmpg, horsepower98, interval = "prediction") #Predicted mpg for a randomly selected ca
r with 98 horsepower
```

```
##          fit      lwr      upr
## 1 24.46708 14.8094 34.12476
```

From these results we can interpret that we are 95% confident that the mean mpg for cars with 98 horsepower is between 23.80 and 25.40. For an individual car with 98 horsepower, we predict with 95% confidence that its mpg will be between 20.50 and 28.34.

Question 9

(a) Produce a scatterplot matrix which includes all of the variables in the data set.

```
pairs(Auto[, sapply(Auto, is.numeric)]) #Creates a scatterplot matrix for all the variables, sel
ecting only numeric columns
```



(b) Compute the matrix of correlations between the variables using the function `cor()`. You will need to exclude the name variable, `cor()` which is qualitative.

```
cor(Auto[, sapply(Auto, is.numeric)]) #Creates the matrix of correlations between variables, exc
luding the name variable.
```

```
##          mpg  cylinders displacement horsepower      weight
## mpg      1.0000000 -0.7762599  -0.8044430          NA -0.8317389
## cylinders -0.7762599  1.0000000   0.9509199          NA  0.8970169
## displacement -0.8044430  0.9509199   1.0000000          NA  0.9331044
## horsepower      NA          NA          NA          1          NA
## weight      -0.8317389  0.8970169   0.9331044          NA  1.0000000
## acceleration  0.4222974 -0.5040606  -0.5441618          NA -0.4195023
## year         0.5814695 -0.3467172  -0.3698041          NA -0.3079004
## origin       0.5636979 -0.5649716  -0.6106643          NA -0.5812652
##          acceleration      year      origin
## mpg      0.4222974  0.5814695  0.5636979
## cylinders -0.5040606 -0.3467172 -0.5649716
## displacement -0.5441618 -0.3698041 -0.6106643
## horsepower      NA          NA          NA
## weight      -0.4195023 -0.3079004 -0.5812652
## acceleration  1.0000000  0.2829009  0.2100836
## year         0.2829009  1.0000000  0.1843141
## origin       0.2100836  0.1843141  1.0000000
```

(c) Use the `lm()` function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the `summary()` function to print the results. Comment on the output.

```
mlmpg = lm(mpg~cylinders + displacement + horsepower + weight + acceleration + year + origin, data = Auto)
summary(mlmpg)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.218435   4.644294  -3.707  0.00024 ***
## cylinders    -0.493376   0.323282  -1.526  0.12780
## displacement  0.019896   0.007515   2.647  0.00844 **
## horsepower   -0.016951   0.013787  -1.230  0.21963
## weight       -0.006474   0.000652  -9.929 < 2e-16 ***
## acceleration  0.080576   0.098845   0.815  0.41548
## year          0.750773   0.050973  14.729 < 2e-16 ***
## origin        1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

The weight, year, and origin of the car appear to be the most statistically significant predictors, with p-values close to 0, indicating a strong relationship with mpg. The displacement predictor also seems statistically significant with a p-value close to 0.001. The remaining predictors have p-values closer to 1, suggesting they lack statistical significance in explaining variations in mpg.

The origin, year, and displacement predictors show a positive relationship with mpg. Specifically, origin has the strongest effect on mpg, with a coefficient of 1.4261 for one category relative to the baseline, followed by year with an effect of 0.7508 for each unit increase. Displacement has a smaller positive effect, with a coefficient of 0.0199 per unit increase. Conversely, weight exhibits a negative effect, reducing mpg by 0.0065 for every unit increase in weight.

(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
par(mfrow = c(2,2)) #Set plot layout to 2x2 grid
plot(mlmpg) #Generate plot
```



The diagnostic plots reveal several insights about the model fit. The Q-Q plot shows that the residuals are approximately normally distributed, with minor deviations at the tails. The Residuals vs. Fitted plot suggests a slight non-linearity in the relationship between the predictors and mpg. The Scale-Location plot indicates mild heteroscedasticity, with residual variability increasing slightly as fitted values increase. Finally, the Residuals vs. Leverage plot does not highlight any high-leverage or overly influential observations, suggesting that no single data point disproportionately affects the model.

*(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?*

We can run a linear regression model incorporating interaction terms between the most statistically significant predictors to assess whether these interactions provide additional explanatory power for understanding the variability in miles per gallon (mpg).

```
mpg_interac = lm(mpg~ cylinders + displacement + horsepower + weight + acceleration + year + origin + weight*year + weight*origin + weight*displacement + year*origin + year*displacement + origin*displacement + horsepower*year + horsepower*origin + horsepower*weight+ horsepower*displacement, data = Auto)
summary(mpg_interac)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##      acceleration + year + origin + weight * year + weight * origin +
##      weight * displacement + year * origin + year * displacement +
##      origin * displacement + horsepower * year + horsepower *
##      origin + horsepower * weight + horsepower * displacement,
##      data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.1616 -1.4868 -0.0497  1.3327 11.3031
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.093e+01  2.680e+01  -1.900  0.05818 .
## cylinders         4.073e-01  3.108e-01   1.311  0.19080
## displacement    -2.015e-01  1.203e-01  -1.676  0.09461 .
## horsepower        9.445e-01  2.328e-01   4.058 6.03e-05 ***
## weight         -1.644e-02  1.568e-02  -1.048  0.29523
## acceleration    -7.872e-02  9.972e-02  -0.789  0.43036
## year             1.400e+00  3.320e-01   4.217 3.10e-05 ***
## origin          -3.146e+00  5.495e+00  -0.573  0.56731
## weight:year       7.611e-05  1.905e-04   0.400  0.68968
## weight:origin     1.263e-03  1.131e-03   1.117  0.26474
## displacement:weight 2.669e-05  6.345e-06   4.207 3.24e-05 ***
## year:origin       3.171e-02  6.966e-02   0.455  0.64919
## displacement:year  1.123e-03  1.441e-03   0.779  0.43639
## displacement:origin 2.687e-02  1.299e-02   2.068  0.03928 *
## horsepower:year    -1.164e-02  2.893e-03  -4.024 6.93e-05 ***
## horsepower:origin  -5.776e-02  2.056e-02  -2.810  0.00522 **
## horsepower:weight  -1.419e-05  1.827e-05  -0.777  0.43778
## displacement:horsepower -2.864e-05  1.405e-04  -0.204  0.83855
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.755 on 374 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.8808, Adjusted R-squared:  0.8754
## F-statistic: 162.6 on 17 and 374 DF, p-value: < 2.2e-16
```

From the regression results, we observe that most interaction coefficients are not only small but also statistically insignificant. However, there are a few exceptions. With a p-value close to 0, the interaction terms weight:displacement and horsepower:year provide strong evidence of statistical significance, with an effect of 1.814×10^{-5} and 6.93×10^{-5} , respectively, on mpg. The interaction terms horsepower:origin and displacement:origin also appear to be somewhat significant, with the former having a p-value close to 0.001 and the latter close to 0.01.

Additionally, we observe that the inclusion of interaction terms causes for the displacement, weight and origin predictors that were previously determined to be statistically significant to become insignificant. However, in the case of horsepower, strong evidence for its statistical significance was now found. This effect can be attributed to

increased multicollinearity between the interaction terms and their associated individual predictors.

(f) Try a few different transformations of the variables, such as $\log(X)$, \sqrt{X} , X^2 . Comment on your findings.

Logarithmic Transformation of Weight:

```
log_lmmpg = lm(mpg ~ cylinders + displacement + horsepower + log(weight) + acceleration + year +
origin, data = Auto)
summary(log_lmmpg)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + log(weight) +
##     acceleration + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.382 -1.973 -0.016  1.681 12.803
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  133.138912   11.531157   11.546 < 2e-16 ***
## cylinders     -0.432749    0.299977   -1.443 0.149946
## displacement  0.020977    0.006825    3.074 0.002265 **
## horsepower    -0.010072    0.012546   -0.803 0.422593
## log(weight)   -21.858785    1.651400  -13.237 < 2e-16 ***
## acceleration  0.135179    0.089798    1.505 0.133051
## year          0.788784    0.047596   16.573 < 2e-16 ***
## origin        1.011407    0.262262    3.856 0.000135 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.091 on 384 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.8459, Adjusted R-squared:  0.8431
## F-statistic: 301.2 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(log_lmmpg)
```



With the logarithmic transformation of the weight predictor, we don't see any changes in the significance of the predictors. However, we do see an increase in the Adjusted R-squared compared to the original model ($0.8431 > 0.8182$) indicating a better fit. Residual plots exhibit virtually no changes compared to the original model.

Square Root Transformation of Acceleration and Horsepower:

```
sqrt_lmmpg = lm(mpg ~ cylinders + displacement + sqrt(horsepower) + weight + sqrt(acceleration)
+ year + origin, data = Auto)
summary(sqrt_lmmpg)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + sqrt(horsepower) +
##     weight + sqrt(acceleration) + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.615 -2.038 -0.138  1.761 12.945
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.9378664   6.6729385  -0.290  0.77166
## cylinders      -0.5175082   0.3161160  -1.637  0.10243
## displacement    0.0210864   0.0072379   2.913  0.00379 **
## sqrt(horsepower) -1.2539322   0.3141308  -3.992 7.86e-05 ***
## weight         -0.0052428   0.0006955  -7.539 3.46e-13 ***
## sqrt(acceleration) -1.2478493   0.8448937  -1.477  0.14051
## year           0.7220943   0.0500773  14.420 < 2e-16 ***
## origin         1.5109369   0.2699801   5.596 4.17e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.272 on 384 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.8274, Adjusted R-squared:  0.8243
## F-statistic: 263 on 7 and 384 DF, p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(sqrt_lmmpg)
```



With the square root transformation applied to acceleration and horsepower, we observe that the transformed horsepower predictor gained statistical significance. Additionally, the Adjusted R-squared value increased compared to the original model ($0.8243 > 0.8182$). However, residual plots remain virtually unchanged from those of the original model.

Quadratic Transformation of Horsepower and Weight:

```
quad_lmmpg = lm(mpg ~ cylinders + displacement + I(horsepower^2) + I(weight^2) + acceleration +
year + origin, data = Auto)
summary(quad_lmmpg)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + I(horsepower^2) +
##      I(weight^2) + acceleration + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1323  -2.3350  -0.0783   2.0358  13.4260
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.817e+01  4.802e+00  -5.867 9.61e-09 ***
## cylinders      -4.925e-01  3.565e-01  -1.381   0.168
## displacement  -3.060e-03  8.447e-03  -0.362   0.717
## I(horsepower^2) 7.734e-05  4.744e-05   1.630   0.104
## I(weight^2)    -7.649e-07  9.869e-08  -7.750 8.29e-14 ***
## acceleration   1.350e-01  9.512e-02   1.420   0.157
## year           7.491e-01  5.483e-02  13.661 < 2e-16 ***
## origin         1.438e+00  3.053e-01   4.709 3.48e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.613 on 384 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.7896, Adjusted R-squared:  0.7858
## F-statistic: 205.9 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(quad_lmmpg)
```



When applying a quadratic transformation to horsepower and weight, the acceleration predictor lost its statistical significance. Furthermore, the transformed model exhibited a lower Adjusted R-squared value compared to the original model ($0.7858 < 0.8182$) suggesting a worse overall fit. Residual plots also show no noticeable differences from the original model.

Question 13

```
set.seed(1)
```

(a) Using the `rnorm()` function, create a vector, `x`, containing 100 observations drawn from a $N(0, 1)$ distribution. This represents a feature, X .

```
x = rnorm(100, mean = 0, sd = 1)
```

(b) Using the `rnorm()` function, create a vector, `eps`, containing 100 observations drawn from a $N(0, 0.25)$ distribution—a normal distribution with mean zero and variance 0.25.

```
eps = rnorm(100, mean = 0, sd = sqrt(0.25))
```

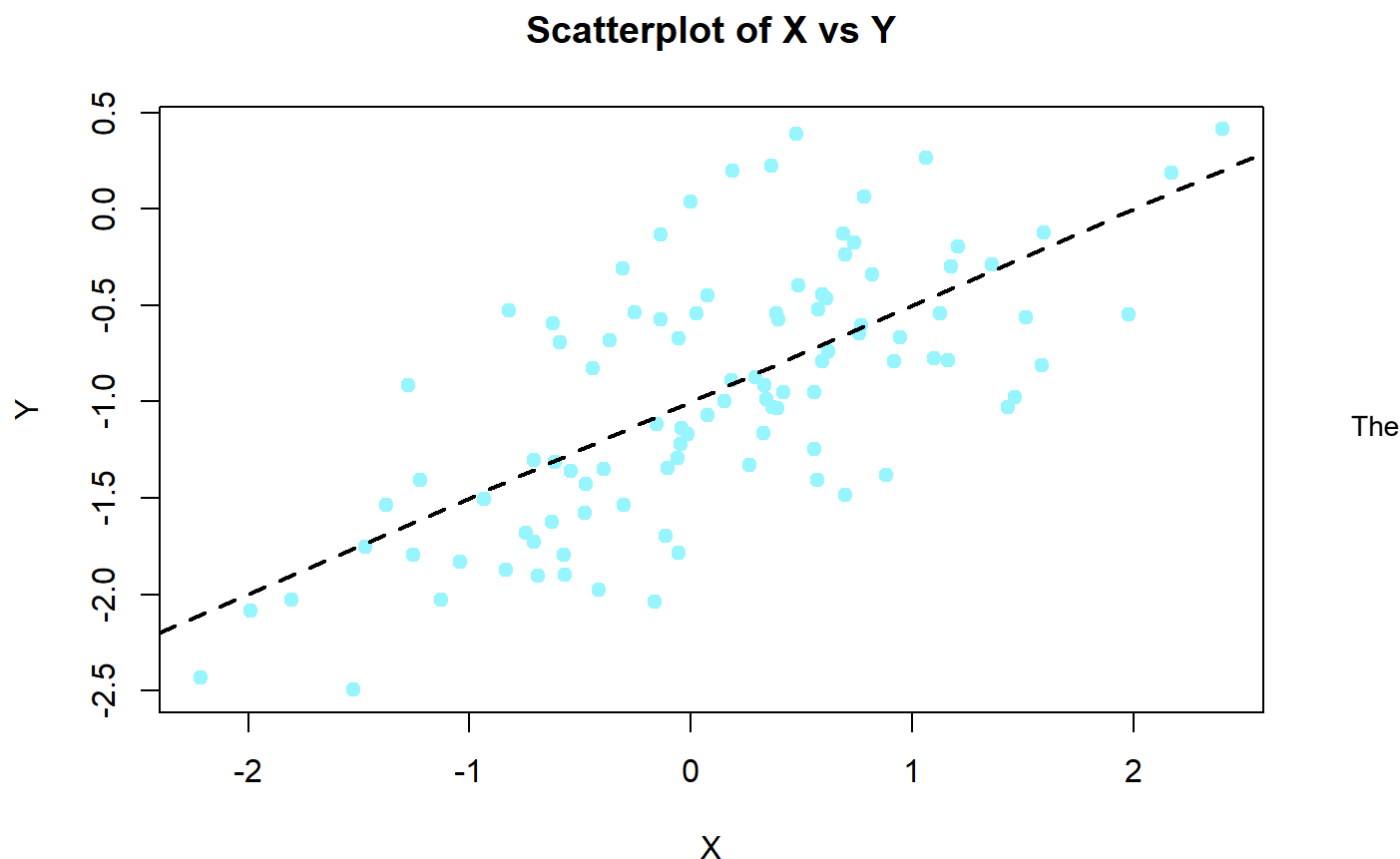

(c) Using x and eps , generate a vector y according to the model $Y = -1 + 0.5X + \epsilon$. (3.39) What is the length of the vector y ? What are the values of β_0 and β_1 in this linear model?

```
y = -1 + 0.5*x + eps
```

The length of the vector will be 100, the same as both x and eps . The value of the intercept parameter is $\beta_0 = -1$, while the value of the slope parameter is $\beta_1 = 0.5$.

(d) Create a scatterplot displaying the relationship between x and y . Comment on what you observe.

```
plot(x, y, main="Scatterplot of X vs Y", xlab="X", ylab="Y", pch=19, col="cadetblue1")
abline(-1, 0.5, col="black", lwd=2, lty=2)
```



scatterplot shows a positive, approximately linear relationship between X and Y . As X increases, Y tends to increase as well. However, the random noise introduced by the eps term creates some variability around the linear trend, resulting in a moderate spread of points across the regression line.

(e) Fit a least squares linear model to predict y using x . Comment on the model obtained. How do $\hat{\beta}_0$ and $\hat{\beta}_1$ compare to β_0 and β_1 ?

```
lm = lm(y~x)
summary(lm)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885     0.04849  -21.010  < 2e-16 ***
## x            0.49947     0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

The estimated intercept $\hat{\beta}_0 = -0.98632$ is close to the true value $\beta_0 = -1$. Similarly, the estimated coefficient for x $\hat{\beta}_1 = 0.51058$ is close to the true value $\beta_1 = 0.5$. The small deviation of these estimated values from the true values is due to the random noise introduced by the error term, ϵ . However, we can conclude that the model accurately captures the true relationship between x and y .

Additionally, there is strong evidence that x is a statistically significant predictor, as indicated by a p-value of $2 * 10^{-16}$. However, the R-squared value for the model is very low at 0.4346, or the model is relatively low at 0.4346, meaning that only 43.46% of the variability in Y is explained by the predictor variable, x , in this linear regression model.

Question 15

```
Boston <- read.csv("C:/Users/mateo/OneDrive - University of North Carolina at Chapel Hill/Courses/
s/Spring 2025/Econ 573/ALL+CSV+FILES+-+2nd+Edition+-+corrected/ALL CSV FILES - 2nd Edition/Boston.csv", header = TRUE)
Boston <- Boston[, sapply(Boston, is.numeric)]
Boston <- Boston[, sapply(Boston, function(col) length(unique(col)) > 1)]
Boston <- Boston[, !names(Boston) %in% "X"]
Boston$chas <- as.factor(Boston$chas)
```

(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.

```

model_zn = lm(crim ~ zn, data = Boston)
model_chas = lm(crim ~ chas, data = Boston)
model_indus = lm(crim ~ indus, data = Boston)
model_nox = lm(crim ~ nox, data = Boston)
model_rm = lm(crim ~ rm, data = Boston)
model_age = lm(crim ~ age, data = Boston)
model_dis = lm(crim ~ dis, data = Boston)
model_rad = lm(crim ~ rad, data = Boston)
model_tax = lm(crim ~ tax, data = Boston)
model_ptratio = lm(crim ~ ptratio, data = Boston)
model_lstat = lm(crim ~ lstat, data = Boston)
model_medv = lm(crim ~ medv, data = Boston)

```

```
summary(model_zn)
```

```

##
## Call:
## lm(formula = crim ~ zn, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.429 -4.222 -2.620  1.250  84.523
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675 < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic: 21.1 on 1 and 504 DF,  p-value: 5.506e-06

```

```
summary(model_chas)
```

```
##
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## 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 ***
## chas1        -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
```

```
summary(model_indus)
```

```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374     0.66723  -3.093  0.00209 **
## indus        0.50978     0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF, p-value: < 2.2e-16
```

```
summary(model_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.371  -2.738  -0.974   0.559   81.728
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699   -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(model_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.604  -3.952  -2.654   0.989  87.197
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm           -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807, Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
summary(model_age)
```

```
##
## Call:
## lm(formula = crim ~ age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
summary(model_dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## 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
```

```
summary(model_rad)
```

```
##
## Call:
## lm(formula = crim ~ rad, data = Boston)
##
## 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
```

```
summary(model_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369    0.815809  -10.45 <2e-16 ***
## tax          0.029742    0.001847   16.10 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(model_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.654  -3.985  -1.912   1.825  83.353
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469      3.1473  -5.607 3.40e-08 ***
## ptratio      1.1520      0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
summary(model_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054      0.69376  -4.801 2.09e-06 ***
## lstat        0.54880      0.04776  11.491 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic: 132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(model_medv)
```



```
##
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.071  -4.022  -2.343   1.298  80.957
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.79654    0.93419   12.63  <2e-16 ***
## medv        -0.36316    0.03839   -9.46  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF, p-value: < 2.2e-16
```

```
par(mfrow = c(3, 4))

plot(Boston$zn, Boston$crim, main = "zn vs crim", xlab = "zn", ylab = "Crime Rate")
abline(model_zn, col = "red", lwd = 2)

plot(Boston$indus, Boston$crim, main = "indus vs crim", xlab = "indus", ylab = "Crime Rate")
abline(model_indus, col = "red", lwd = 2)

plot(Boston$nox, Boston$crim, main = "nox vs crim", xlab = "nox", ylab = "Crime Rate")
abline(model_nox, col = "red", lwd = 2)

plot(Boston$rm, Boston$crim, main = "rm vs crim", xlab = "rm", ylab = "Crime Rate")
abline(model_rm, col = "red", lwd = 2)

plot(Boston$age, Boston$crim, main = "age vs crim", xlab = "age", ylab = "Crime Rate")
abline(model_age, col = "red", lwd = 2)

plot(Boston$dis, Boston$crim, main = "dis vs crim", xlab = "dis", ylab = "Crime Rate")
abline(model_dis, col = "red", lwd = 2)

plot(Boston$rad, Boston$crim, main = "rad vs crim", xlab = "rad", ylab = "Crime Rate")
abline(model_rad, col = "red", lwd = 2)

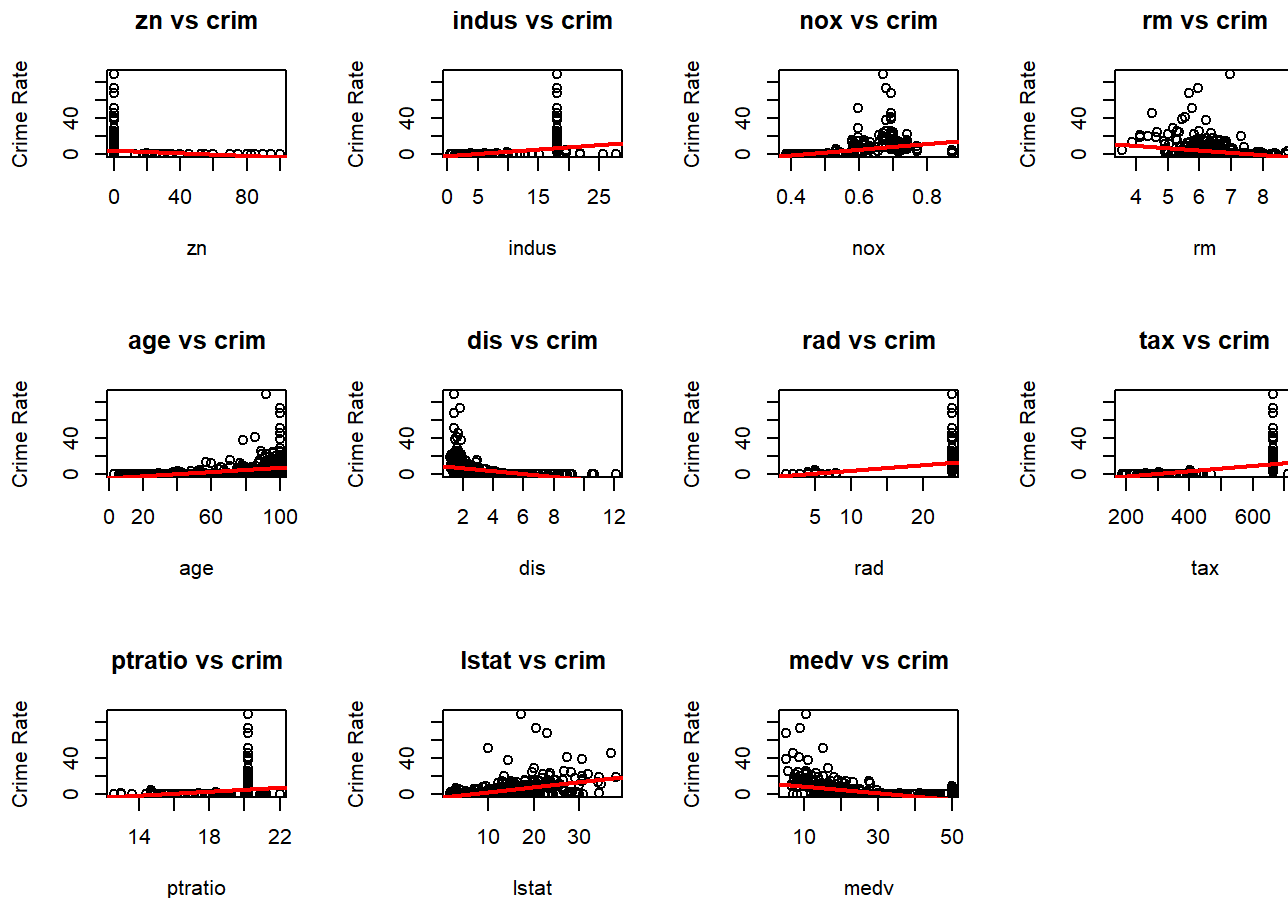
plot(Boston$tax, Boston$crim, main = "tax vs crim", xlab = "tax", ylab = "Crime Rate")
abline(model_tax, col = "red", lwd = 2)

plot(Boston$ptratio, Boston$crim, main = "ptratio vs crim", xlab = "ptratio", ylab = "Crime Rate")
abline(model_ptratio, col = "red", lwd = 2)

plot(Boston$lstat, Boston$crim, main = "lstat vs crim", xlab = "lstat", ylab = "Crime Rate")
abline(model_lstat, col = "red", lwd = 2)

plot(Boston$medv, Boston$crim, main = "medv vs crim", xlab = "medv", ylab = "Crime Rate")
abline(model_medv, col = "red", lwd = 2)

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



of the simple linear regression models, we observe strong evidence of statistical significance for each predictor, with p-values close to 0. This suggests that, when considered individually, each predictor has a strong association with per capita crime rate (crim).

However, this is expected because each model only includes one predictor at a time, meaning the regression is not accounting for interactions or correlations between multiple predictors. Some of these relationships may weaken when we include multiple predictors in a multiple regression model, due to multicollinearity or shared explanatory power.

(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 $H_0 : \beta_j = 0$?

```
multi_model <- lm(crim ~ ., data = Boston)
summary(multi_model)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.534 -2.248 -0.348  1.087 73.923
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.7783938   7.0818258   1.946 0.052271 .
## zn          0.0457100   0.0187903   2.433 0.015344 *
## indus       -0.0583501   0.0836351  -0.698 0.485709
## chas1       -0.8253776   1.1833963  -0.697 0.485841
## nox         -9.9575865   5.2898242  -1.882 0.060370 .
## rm          0.6289107   0.6070924   1.036 0.300738
## age        -0.0008483   0.0179482  -0.047 0.962323
## dis        -1.0122467   0.2824676  -3.584 0.000373 ***
## rad         0.6124653   0.0875358   6.997 8.59e-12 ***
## tax        -0.0037756   0.0051723  -0.730 0.465757
## ptratio    -0.3040728   0.1863598  -1.632 0.103393
## lstat       0.1388006   0.0757213   1.833 0.067398 .
## medv       -0.2200564   0.0598240  -3.678 0.000261 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.46 on 493 degrees of freedom
## Multiple R-squared:  0.4493, Adjusted R-squared:  0.4359
## F-statistic: 33.52 on 12 and 493 DF, p-value: < 2.2e-16
```

In the multiple regression model, the number of statistically significant predictors is drastically reduced compared to the simple regression models. In this case, only dis (distance to employment centers), rad (accessibility to highways), and medv (median home value) remain statistically significant, with p-values close to 0.

This suggests that we can confidently reject the null hypothesis, $H_0 : \beta_j = 0$ for these predictors at all conventional confidence levels.

(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.

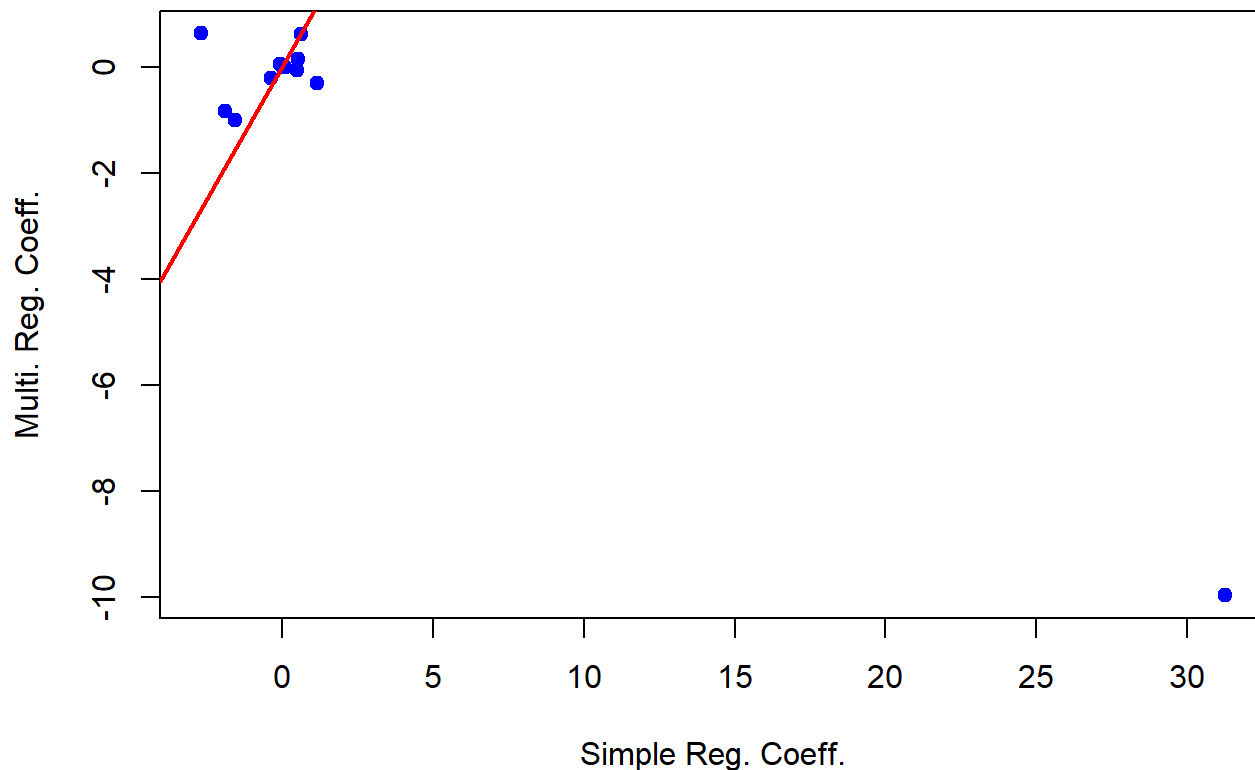
```

simple_coefficients <- sapply(names(Boston)[-1], function(var) {
  model <- lm(as.formula(paste("crim ~", var)), data = Boston) # Fit model
  coef(model)[2] # Extract the coefficient (slope)
})
multi_coefficients <- coef(multi_model)[-1]
plot(simple_coefficients, multi_coefficients,
     xlab = "Simple Reg. Coeff.",
     ylab = "Multi. Reg. Coeff.",
     main = "Simple vs Multiple Reg. Coefficients",
     pch = 19, col = "blue")

abline(a = 0, b = 1, col = "red", lwd = 2)

```

Simple vs Multiple Reg. Coefficients



```

coefficient_differences <- simple_coefficients - multi_coefficients
sorted_differences <- sort(coefficient_differences, decreasing = TRUE)
print(sorted_differences)

```

##	nox.nox	ptratio.ptratio	indus.indus	lstat.lstat	age.age
##	41.206117748	1.456055544	0.568126442	0.410004185	0.108634506
##	tax.tax	rad.rad	zn.zn	medv.medv	dis.dis
##	0.033517899	0.005445616	-0.119645016	-0.143103563	-0.538654944
##	chas.chas1	rm.rm			
##	-1.067398999	-3.312961886			

```
largest_changes <- sort(abs(coefficient_differences), decreasing = TRUE)
print(largest_changes)
```

##	nox.nox	rm.rm	ptratio.ptratio	chas.chas1	indus.indus
##	41.206117748	3.312961886	1.456055544	1.067398999	0.568126442
##	dis.dis	lstat.lstat	medv.medv	zn.zn	age.age
##	0.538654944	0.410004185	0.143103563	0.119645016	0.108634506
##	tax.tax	rad.rad			
##	0.033517899	0.005445616			

The comparison between simple and multiple regression coefficients reveals that predictors like nox and rm experience substantial changes, with their coefficients drastically decreasing in the multiple regression model. This suggests these predictors are highly correlated with others and their effects are diminished when accounting for additional variables. In contrast, predictors like lstat, ptratio, and age remain consistent across both models, indicating they are strong independent predictors of crim.

(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 $Y = \beta_0 + \beta_1X + \beta_2X^2 + \beta_3X^3 + \epsilon$.

```
# Fit polynomial regression models for each predictor
model_zn <- lm(crim ~ poly(zn, 3), data = Boston)
model_indus <- lm(crim ~ poly(indus, 3), data = Boston)
model_nox <- lm(crim ~ poly(nox, 3), data = Boston)
model_rm <- lm(crim ~ poly(rm, 3), data = Boston)
model_age <- lm(crim ~ poly(age, 3), data = Boston)
model_dis <- lm(crim ~ poly(dis, 3), data = Boston)
model_rad <- lm(crim ~ poly(rad, 3), data = Boston)
model_tax <- lm(crim ~ poly(tax, 3), data = Boston)
model_ptratio <- lm(crim ~ poly(ptratio, 3), data = Boston)
model_lstat <- lm(crim ~ poly(lstat, 3), data = Boston)
model_medv <- lm(crim ~ poly(medv, 3), data = Boston)

summary(model_zn)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3722   9.709 < 2e-16 ***
## poly(zn, 3)1  -38.7498     8.3722  -4.628 4.7e-06 ***
## poly(zn, 3)2   23.9398     8.3722   2.859 0.00442 **
## poly(zn, 3)3  -10.0719     8.3722  -1.203 0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824, Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF, p-value: 1.281e-06
```

```
summary(model_indus)
```

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.614     0.330  10.950 < 2e-16 ***
## poly(indus, 3)1  78.591     7.423  10.587 < 2e-16 ***
## poly(indus, 3)2  -24.395     7.423  -3.286 0.00109 **
## poly(indus, 3)3  -54.130     7.423  -7.292 1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF, p-value: < 2.2e-16
```

```
summary(model_nox)
```

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3216  11.237 < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249 < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297, Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF, p-value: < 2.2e-16
```

```
summary(model_rm)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.485  -3.468  -2.221  -0.015   87.219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758 < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191 0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662 0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779, Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF, p-value: 1.067e-07
```

```
summary(model_age)
```



```
##
## Call:
## lm(formula = crim ~ poly(age, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3485  10.368 < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697 < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724 0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
summary(model_dis)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.757 -2.588  0.031  1.267 76.378
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3259  11.087 < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010 < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
summary(model_rad)
```

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.2971  12.164 < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093 < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:  0.4, Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF, p-value: < 2.2e-16
```

```
summary(model_tax)
```

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3047  11.860 < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436 < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167   0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF, p-value: < 2.2e-16
```

```
summary(model_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.833 -4.146 -1.655  1.408  82.697
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.614      0.361  10.008 < 2e-16 ***
## poly(ptratio, 3)1  56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2  24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3 -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

```
summary(model_lstat)
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135      0.3392  10.654 <2e-16 ***
## poly(lstat, 3)1  88.0697      7.6294  11.543 <2e-16 ***
## poly(lstat, 3)2  15.8882      7.6294   2.082  0.0378 *
## poly(lstat, 3)3 -11.5740      7.6294  -1.517  0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
summary(model_medv)
```

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.614      0.292  12.374 < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426 < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409 < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

The results reveal strong evidence of non-linear relationships between several predictors and crime rate (crim). Notably, nox (nitric oxide concentration), dis (distance to employment centers), and medv (median home value) exhibit highly significant quadratic and cubic terms, with p-values close to 0. This indicates complex non-linear associations that impact crime rate. Predictors like indus (non-retail business acres), age (proportion of older housing), and ptratio (pupil-teacher ratio) show moderate non-linear effects with significant quadratic and cubic terms with their p-values being less than 0.01

In contrast, predictors such as zn (residential land zoning), rad (accessibility to highways), and tax (property tax rate) demonstrate weak or no significant non-linear effects, suggesting simpler models may suffice for them. In conclusion, we could incorporate non-linear terms for key predictors like nox, dis, and medv to enhance our model.