

Homework-3 Problem-3

Arinjay Jain

```
library(MASS)
boston_df <- Boston
```

Part a:

```
## Changing chas variable into factor.
boston_df$chas <- as.factor(boston_df$chas)
attach(boston_df)
```

```
#creating vectoe for all p values
p_values <- c()
```

```
# model with "zn"
model_zn <- lm(crim ~ zn)
smry_zn <- summary(model_zn)
print(smry_zn)
```

```
##
## Call:
## lm(formula = crim ~ zn)
##
## 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
```

```
pvalue_zn <- smry_zn$coefficients[2,4]
p_values <- append(p_values, pvalue_zn)
```

```

# model with "indus"
model_indus <- lm(crim ~ indus)
smry_indus <- summary(model_indus)
print(smry_indus)

##
## Call:
## lm(formula = crim ~ indus)
##
## 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

pvalue_indus <- smry_indus$coefficients[2,4]
p_values <- append(p_values, pvalue_indus)

# model with "chas"
model_chas <- lm(crim ~ chas)
smry_chas <- summary(model_chas)
print(smry_chas)

```

```

##
## Call:
## lm(formula = 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 ***
## 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

```

```
pvalue_chas <- smry_chas$coefficients[2,4]
p_values <- append(p_values, pvalue_chas)
```

```
# model with "nox"
model_nox <- lm(crim ~ nox)
smry_nox <- summary(model_nox)
print(smry_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox)
##
## 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
```

```
pvalue_nox <- smry_nox$coefficients[2,4]
p_values <- append(p_values, pvalue_nox)
```

```
# model with "rm"
model_rm <- lm(crim ~ rm)
smry_rm <- summary(model_rm)
print(smry_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm)
##
## 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
```

```
pvalue_rm <- smry_rm$coefficients[2,4]
p_values <- append(p_values, pvalue_rm)
```

```
# model with "age"
model_age <- lm(crim ~ age)
smry_age <- summary(model_age)
print(smry_age)
```

```
##
## Call:
## lm(formula = crim ~ age)
##
## 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
```

```
pvalue_age <- smry_age$coefficients[2,4]
p_values <- append(p_values, pvalue_age)
```

```
# model with "dis"
model_dis <- lm(crim ~ dis)
smry_dis <- summary(model_dis)
print(smry_dis)
```

```
##
## Call:
## lm(formula = 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
```

```
pvalue_dis <- smry_dis$coefficients[2,4]
p_values <- append(p_values, pvalue_dis)
```

```
# model with "rad"
model_rad <- lm(crim ~ rad)
smry_rad <- summary(model_rad)
print(smry_rad)
```

```
##
## Call:
## lm(formula = 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
```

```
pvalue_rad <- smry_rad$coefficients[2,4]
p_values <- append(p_values, pvalue_rad)
```

```
# model with "tax"
model_tax <- lm(crim ~ tax)
smry_tax <- summary(model_tax)
print(smry_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax)
##
## 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

pvalue_tax <- smry_tax$coefficients[2,4]
p_values <- append(p_values, pvalue_tax)

# model with "ptratio"
model_ptratio <- lm(crim ~ ptratio)
smry_ptratio <- summary(model_ptratio)
print(smry_ptratio)

##
## Call:
## lm(formula = crim ~ ptratio)
##
## 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

pvalue_ptratio <- smry_ptratio$coefficients[2,4]
p_values <- append(p_values, pvalue_ptratio)

# model with "black"
model_black <- lm(crim ~ black)
smry_black <- summary(model_black)
print(smry_black)

##
## Call:
## lm(formula = crim ~ black)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609  <2e-16 ***
## black       -0.036280   0.003873  -9.367  <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
pvalue_black <- smry_black$coefficients[2,4]
p_values <- append(p_values, pvalue_black)
```

```
# model with "lstat"
model_lstat <- lm(crim ~ lstat)
smry_lstat <- summary(model_lstat)
print(smry_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat)
##
## 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
```

```
pvalue_lstat <- smry_lstat$coefficients[2,4]
p_values <- append(p_values, pvalue_lstat)
```

```
# model with "medv"
model_medv <- lm(crim ~ medv)
smry_medv <- summary(model_medv)
print(smry_medv)
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-9.071	-4.022	-2.343	1.298	80.957

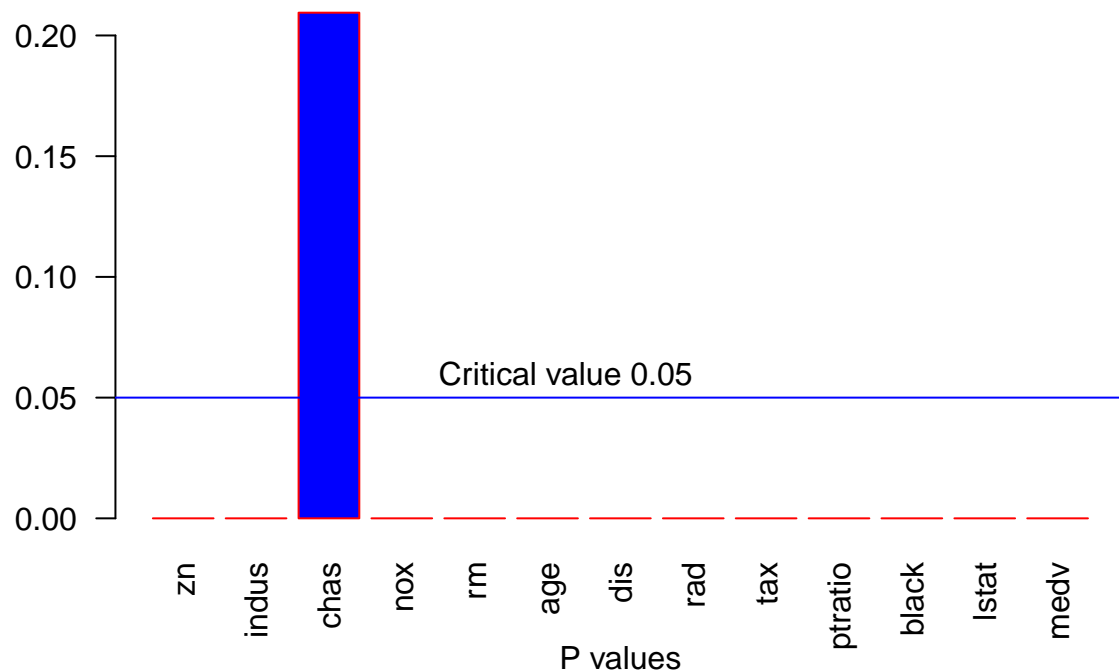
```
##
## Coefficients:
```

	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
```

```
pvalue_medv <- smry_medv$coefficients[2,4]
p_values <- append(p_values, pvalue_medv)
```

```
# Plot the bar chart on P values
barplot(p_values, las=2, names.arg=colnames(boston_df)[-1], xlab="P values", col="blue", border="red")
abline(h=0.05, lwd=1, lty="solid", col="blue")
text(7, 0.06, "Critical value 0.05")
```



```
print("Here we can clearly see the P-value of chas is greater than '0.05' that means chas is not a sign")
```

```
## [1] "Here we can clearly see the P-value of chas is greater than '0.05' that means chas is not a
significant predictor for our model and we can include all other variables in our future model."
```

```
## Correlation matrix
```

```
res <- cor(Boston)
print(res)
```

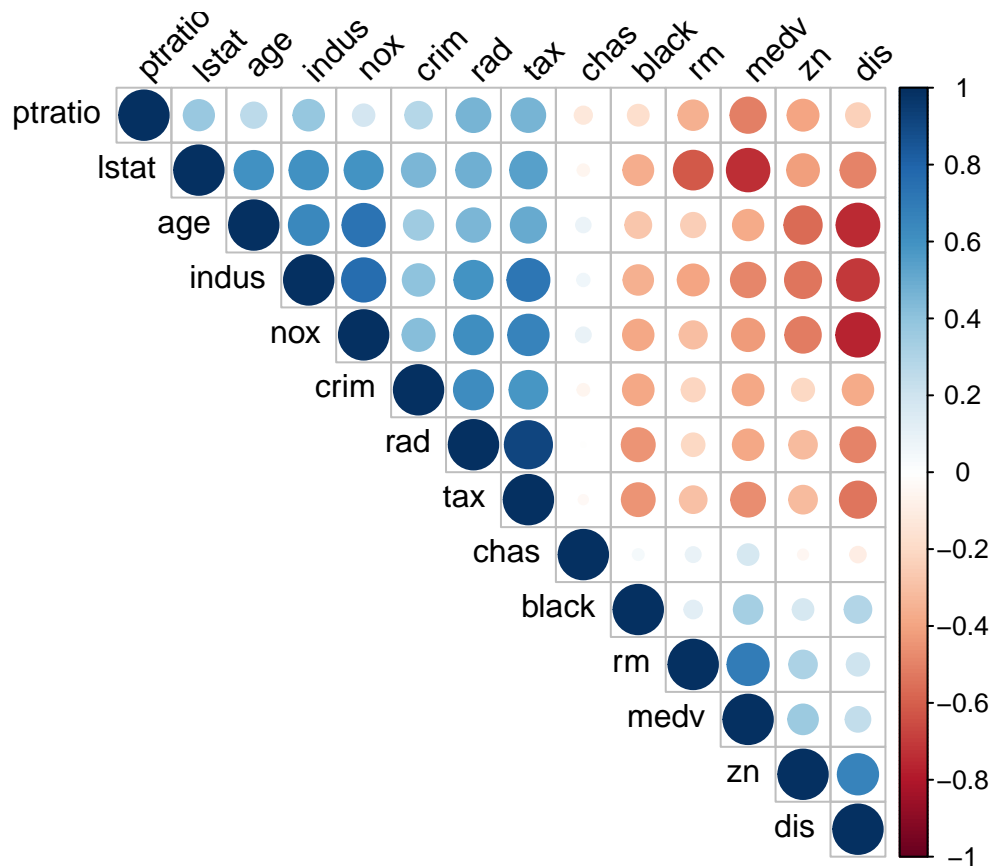


```
##          crim          zn          indus          chas          nox
## crim      1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
## zn        -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
## indus      0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
## chas      -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
## nox       0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
## rm        -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
## age       0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
## dis       -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
## rad       0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
## tax       0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
## ptratio   0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
## black     -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064
## lstat     0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
## medv     -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
##          rm          age          dis          rad          tax          ptratio
## crim     -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431  0.28994556
## zn        0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332 -0.3916785
## indus     -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018  0.3832476
## chas      0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652 -0.1215152
## nox      -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320  0.1889327
## rm        1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783 -0.3555015
## age      -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559  0.2615150
## dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158 -0.2324705
## rad      -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819  0.4647412
## tax      -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000  0.4608530
## ptratio  -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304  1.0000000
## black     0.12806864 -0.27353398  0.29151167 -0.444412816 -0.44180801 -0.1773833
## lstat    -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341  0.3740443
## medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593 -0.5077867
##          black          lstat          medv
## crim     -0.38506394  0.4556215 -0.3883046
## zn        0.17552032 -0.4129946  0.3604453
## indus     -0.35697654  0.6037997 -0.4837252
## chas      0.04878848 -0.0539293  0.1752602
## nox      -0.38005064  0.5908789 -0.4273208
## rm        0.12806864 -0.6138083  0.6953599
## age      -0.27353398  0.6023385 -0.3769546
## dis      0.29151167 -0.4969958  0.2499287
## rad      -0.44441282  0.4886763 -0.3816262
## tax      -0.44180801  0.5439934 -0.4685359
## ptratio  -0.17738330  0.3740443 -0.5077867
## black     1.00000000 -0.3660869  0.3334608
## lstat    -0.36608690  1.0000000 -0.7376627
## medv     0.33346082 -0.7376627  1.0000000
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(res, type = "upper", order = "hclust",
          tl.col = "black", tl.srt = 45, sig.level = 0.05)
```



Part b:

```
full_model <- lm(crim ~ ., data = boston_df)
smry_full<- summary(full_model)
print(smry_full)
```

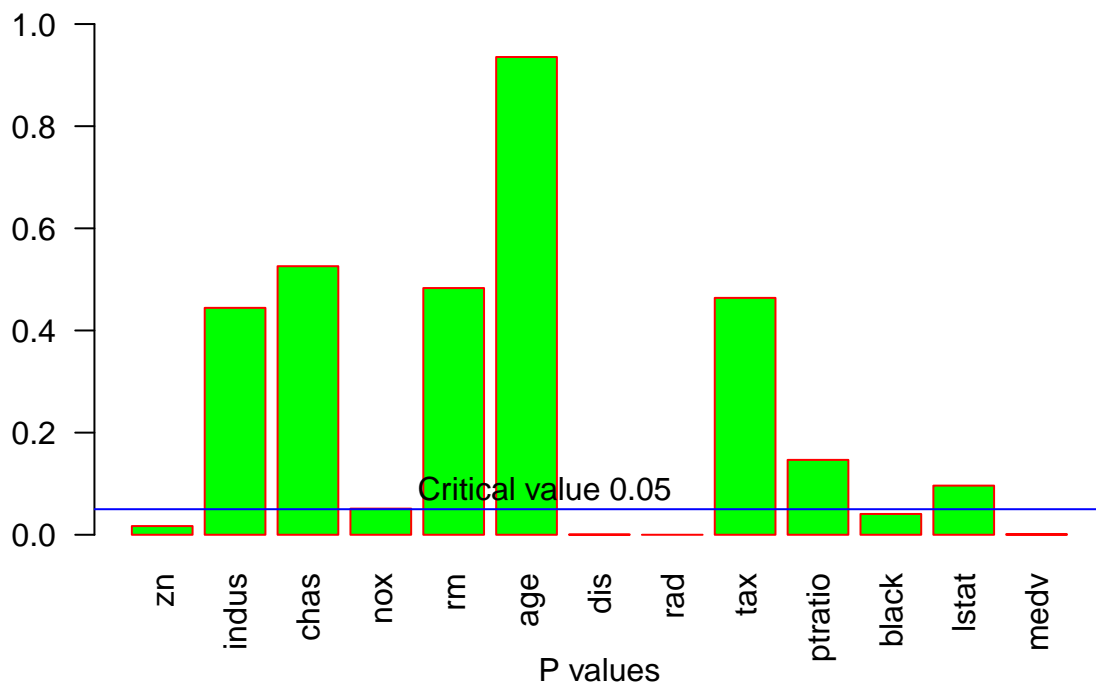
```
##
## Call:
## lm(formula = crim ~ ., data = boston_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.924  -2.120  -0.353   1.019  75.051
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228    7.234903   2.354 0.018949 *
## zn           0.044855    0.018734   2.394 0.017025 *
## indus       -0.063855    0.083407  -0.766 0.444294
## chas1       -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
```

```
p_values_full_model <- smry_full$coefficients[2:14,4]
```

```
# Plot the bar chart on P values
```

```
barplot(p_values_full_model,las=2,names.arg=colnames(boston_df)[-1],xlab="P values",ylim = c(0,1),col="red",
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.09, "Critical value 0.05")
```



```
print("From summary of full model we can reject the null hypothesis for 'zn', 'dis', 'rad', 'black' and
```

```
## [1] "From summary of full model we can reject the null hypothesis for 'zn', 'dis', 'rad',
'black' and 'medv' predictors."
```

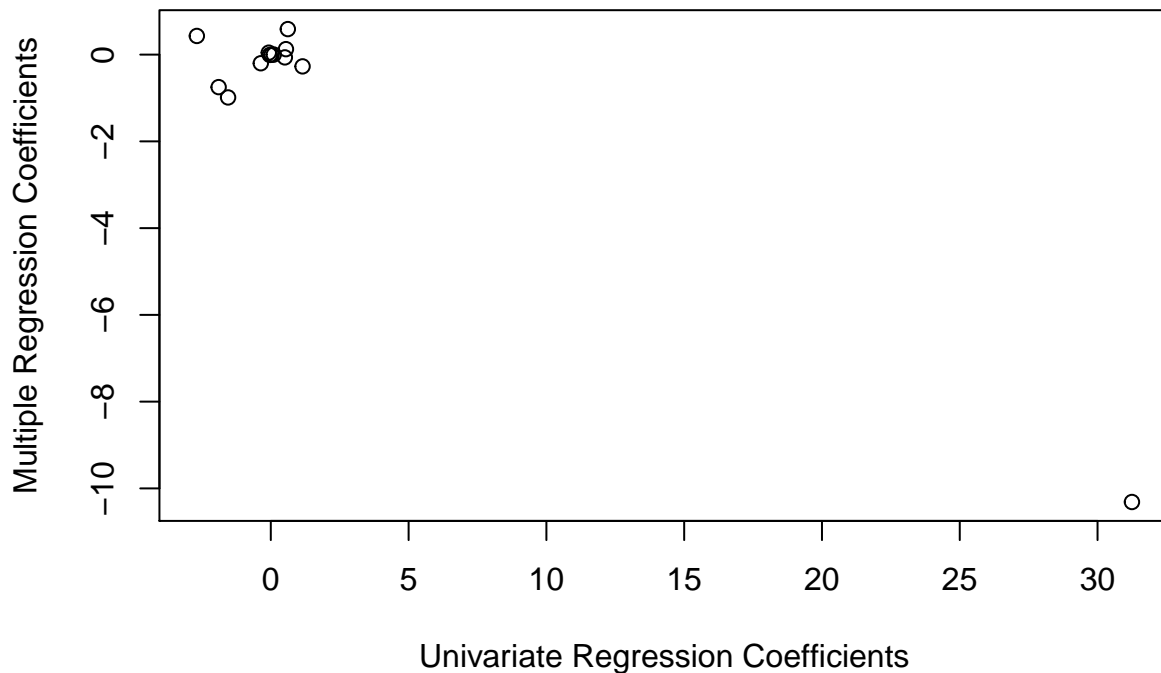
Part c:

```
# getting all coeff into single vector
simple_reg_coff <- c() #empty vector

simple_reg_coff <- c(model_zn$coefficients[2],model_indus$coefficients[2],model_chas$coefficients[2],model_dis$coefficients[2],
                    model_rad$coefficients[2],
                    model_tax$coefficients[2],
                    model_ptratio$coefficients[2],
                    model_black$coefficients[2],
                    model_lstat$coefficients[2],
                    model_medv$coefficients[2])

multiple_reg_coff <- full_model$coefficients[-1]

plot(simple_reg_coff,multiple_reg_coff,xlab = "Univariate Regression Coefficients", ylab = "Multiple Regression Coefficients")
```



```
print("Here we can see the difference in simple regression coefficients and multiple regression coefficients")
```

```
## [1] "Here we can see the difference in simple regression coefficients and multiple regression coefficients and the reason behind is in the simple regression models we are considering one predictor at a time due to this the estimate coefficient beta represent the average effect of an increase in the predictor, not taking other predictors into account. On other side we have multiple
```

regression model the estimate coefficients beta represents the average effect of an increase in the predictor, while holding other predictors fixed. It does make sense for the multiple regression to suggest no relationship between the response and some of the predictors while the simple linear regression implies the opposite because the correlation between the predictors show some strong relationships between some of the predictors."

```
## Correlation matrix
```

```
cor_Mat <- cor(Boston[-c(1,4)])
print(cor_Mat)
```

```
##          zn      indus      nox      rm      age      dis
## zn      1.0000000 -0.5338282 -0.5166037  0.3119906 -0.5695373  0.6644082
## indus   -0.5338282  1.0000000  0.7636514 -0.3916759  0.6447785 -0.7080270
## nox     -0.5166037  0.7636514  1.0000000 -0.3021882  0.7314701 -0.7692301
## rm       0.3119906 -0.3916759 -0.3021882  1.0000000 -0.2402649  0.2052462
## age     -0.5695373  0.6447785  0.7314701 -0.2402649  1.0000000 -0.7478805
## dis      0.6644082 -0.7080270 -0.7692301  0.2052462 -0.7478805  1.0000000
## rad     -0.3119478  0.5951293  0.6114406 -0.2098467  0.4560225 -0.4945879
## tax     -0.3145633  0.7207602  0.6680232 -0.2920478  0.5064556 -0.5344316
## ptratio -0.3916785  0.3832476  0.1889327 -0.3555015  0.2615150 -0.2324705
## black    0.1755203 -0.3569765 -0.3800506  0.1280686 -0.2735340  0.2915117
## lstat   -0.4129946  0.6037997  0.5908789 -0.6138083  0.6023385 -0.4969958
## medv     0.3604453 -0.4837252 -0.4273208  0.6953599 -0.3769546  0.2499287
##          rad      tax      ptratio      black      lstat      medv
## zn     -0.3119478 -0.3145633 -0.3916785  0.1755203 -0.4129946  0.3604453
## indus   0.5951293  0.7207602  0.3832476 -0.3569765  0.6037997 -0.4837252
## nox     0.6114406  0.6680232  0.1889327 -0.3800506  0.5908789 -0.4273208
## rm     -0.2098467 -0.2920478 -0.3555015  0.1280686 -0.6138083  0.6953599
## age     0.4560225  0.5064556  0.2615150 -0.2735340  0.6023385 -0.3769546
## dis    -0.4945879 -0.5344316 -0.2324705  0.2915117 -0.4969958  0.2499287
## rad     1.0000000  0.9102282  0.4647412 -0.4444128  0.4886763 -0.3816262
## tax     0.9102282  1.0000000  0.4608530 -0.4418080  0.5439934 -0.4685359
## ptratio 0.4647412  0.4608530  1.0000000 -0.1773833  0.3740443 -0.5077867
## black  -0.4444128 -0.4418080 -0.1773833  1.0000000 -0.3660869  0.3334608
## lstat   0.4886763  0.5439934  0.3740443 -0.3660869  1.0000000 -0.7376627
## medv   -0.3816262 -0.4685359 -0.5077867  0.3334608 -0.7376627  1.0000000
```

```
## find most correlated variable
```

```
cor_Mat[lower.tri(cor_Mat,diag=TRUE)]<-NA
cor_Cof<- as.data.frame(as.table(cor_Mat))
#removing NA
cor_Cof<-cor_Cof[complete.cases(cor_Cof),]
cor_Cof<-cor_Cof[order(abs(cor_Cof$Freq),decreasing = TRUE),]
# TOP 5 STRONGEST ABSOLUTE CORRELATION
cor_Cof[1:5,]
```

```
##      Var1 Var2      Freq
## 91    rad  tax  0.9102282
## 63    nox  dis -0.7692301
## 26   indus nox  0.7636514
## 65    age  dis -0.7478805
## 143 lstat medv -0.7376627
```

```
print("Here we can see, 'age' and 'dis' having negative correlation , In SLR 'crim' versus 'age', we saw
```

```
## [1] "Here we can see, 'age' and 'dis' having negative correlation , In SLR 'crim' versus 'age',
we saw higher values of 'age' are associated with higher values of 'crim', even though 'age' does
not actually affect 'crim'. So 'age' is a surrogate for 'dis'; 'age' gets credit for the effect of
'dis' on 'crim'."
```

Part d:Non-linear

```
## Y=??0+??1X+??2X2+??3X3+??.

## I am skipping chas as perditor because it is a factor variable
poly_p_values <- c()
poly_p_values_2 <- c()
poly_p_values_3 <- c()

# poly model with "zn"
poly_model_zn <- lm(crim ~ poly(zn, 3))
smry_poly_model_zn <- summary(poly_model_zn)
print(smry_poly_model_zn)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## 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
```

```
poly_pvalue_zn <- smry_poly_model_zn$coefficients[2,4]
poly_pvalue_zn_2 <- smry_poly_model_zn$coefficients[3,4]
poly_pvalue_zn_3 <- smry_poly_model_zn$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_zn)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_zn_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_zn_3)
```

```

# poly model with "indus"
poly_model_indus <- lm(crim ~ poly(indus, 3))
smry_poly_model_indus <- summary(poly_model_indus)
print(smry_poly_model_indus)

##
## Call:
## lm(formula = crim ~ poly(indus, 3))
##
## 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

poly_pvalue_indus <- smry_poly_model_indus$coefficients[2,4]
poly_pvalue_indus_2 <- smry_poly_model_indus$coefficients[3,4]
poly_pvalue_indus_3 <- smry_poly_model_indus$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_indus)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_indus_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_indus_3)

# poly model with "nox"
poly_model_nox <- lm(crim ~ poly(nox, 3))
smry_poly_model_nox <- summary(poly_model_nox)
print(smry_poly_model_nox)

```

```

##
## Call:
## lm(formula = crim ~ poly(nox, 3))
##
## 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
```

```
poly_pvalue_nox <- smry_poly_model_nox$coefficients[2,4]
poly_pvalue_nox_2 <- smry_poly_model_nox$coefficients[3,4]
poly_pvalue_nox_3 <- smry_poly_model_nox$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_nox)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_nox_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_nox_3)
```

```
# poly model with "rm"
poly_model_rm <- lm(crim ~ poly(rm, 3))
smry_poly_model_rm <- summary(poly_model_rm)
print(smry_poly_model_rm)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3))
##
## 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
```

```
poly_pvalue_rm <- smry_poly_model_rm$coefficients[2,4]
poly_pvalue_rm_2 <- smry_poly_model_rm$coefficients[3,4]
poly_pvalue_rm_3 <- smry_poly_model_rm$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_rm)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_rm_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_rm_3)
```

```
# poly model with "age"
poly_model_age <- lm(crim ~ poly(age, 3))
smry_poly_model_age <- summary(poly_model_age)
print(smry_poly_model_age)
```



```
##
## Call:
## lm(formula = crim ~ poly(age, 3))
##
## 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
```

```
poly_pvalue_age <- smry_poly_model_age$coefficients[2,4]
poly_pvalue_age_2 <- smry_poly_model_age$coefficients[3,4]
poly_pvalue_age_3 <- smry_poly_model_age$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_age)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_age_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_age_3)
```

```
# poly model with "dis"
poly_model_dis <- lm(crim ~ poly(dis, 3))
smry_poly_model_dis <- summary(poly_model_dis)
print(smry_poly_model_dis)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3))
##
## 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

poly_pvalue_dis <- smry_poly_model_dis$coefficients[2,4]
poly_pvalue_dis_2 <- smry_poly_model_dis$coefficients[3,4]
poly_pvalue_dis_3 <- smry_poly_model_dis$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_dis)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_dis_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_dis_3)

# poly model with "rad"
poly_model_rad <- lm(crim ~ poly(rad, 3))
smry_poly_model_rad <- summary(poly_model_rad)
print(smry_poly_model_rad)

##
## Call:
## lm(formula = crim ~ poly(rad, 3))
##
## 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

poly_pvalue_rad <- smry_poly_model_rad$coefficients[2,4]
poly_pvalue_rad_2 <- smry_poly_model_rad$coefficients[3,4]
poly_pvalue_rad_3 <- smry_poly_model_rad$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_rad)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_rad_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_rad_3)

# poly model with "tax"
poly_model_tax <- lm(crim ~ poly(tax, 3))
smry_poly_model_tax <- summary(poly_model_tax)
print(smry_poly_model_tax)

##
## Call:
## lm(formula = crim ~ poly(tax, 3))
```

```
##
## 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
```

```
poly_pvalue_tax <- smry_poly_model_tax$coefficients[2,4]
poly_pvalue_tax_2 <- smry_poly_model_tax$coefficients[3,4]
poly_pvalue_tax_3 <- smry_poly_model_tax$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_tax)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_tax_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_tax_3)
```

```
# poly model with "ptratio"
poly_model_ptratio <- lm(crim ~ poly(ptratio, 3))
smry_poly_model_ptratio <- summary(poly_model_ptratio)
print(smry_poly_model_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3))
##
## 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
```

```
poly_pvalue_ptratio <- smry_poly_model_ptratio$coefficients[2,4]
poly_pvalue_ptratio_2 <- smry_poly_model_ptratio$coefficients[3,4]
```

```
poly_pvalue_ptratio_3 <- smry_poly_model_ptratio$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_ptratio)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_ptratio_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_ptratio_3)
```

```
# poly model with "black"
```

```
poly_model_black <- lm(crim ~ poly(black, 3))
smry_poly_model_black <- summary(poly_model_black)
print(smry_poly_model_black)
```

```
##
## Call:
## lm(formula = crim ~ poly(black, 3))
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-13.096	-2.343	-2.128	-1.439	86.790

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	3.6135	0.3536	10.218	<2e-16 ***
## poly(black, 3)1	-74.4312	7.9546	-9.357	<2e-16 ***
## poly(black, 3)2	5.9264	7.9546	0.745	0.457
## poly(black, 3)3	-4.8346	7.9546	-0.608	0.544

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
poly_pvalue_black <- smry_poly_model_black$coefficients[2,4]
poly_pvalue_black_2 <- smry_poly_model_black$coefficients[3,4]
poly_pvalue_black_3 <- smry_poly_model_black$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_black)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_black_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_black_3)
```

```
# poly model with "lstat"
```

```
poly_model_lstat <- lm(crim ~ poly(lstat, 3))
smry_poly_model_lstat <- summary(poly_model_lstat)
print(smry_poly_model_lstat)
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3))
##
## 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
```

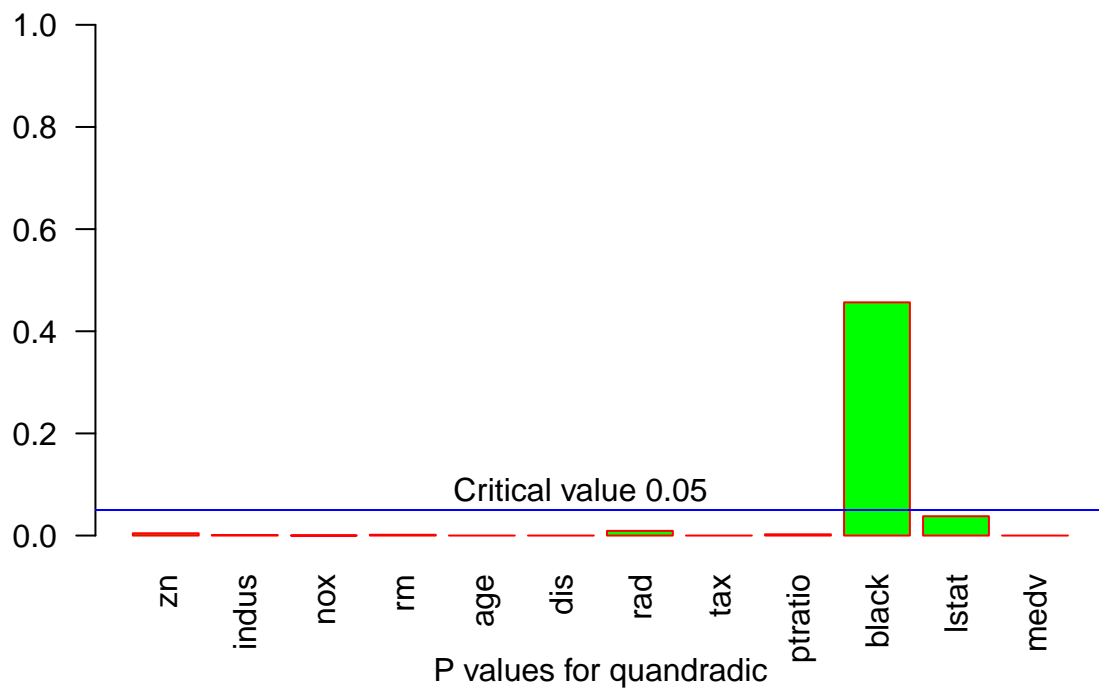
```
poly_pvalue_lstat <- smry_poly_model_lstat$coefficients[2,4]
poly_pvalue_lstat_2 <- smry_poly_model_lstat$coefficients[3,4]
poly_pvalue_lstat_3 <- smry_poly_model_lstat$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_lstat)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_lstat_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_lstat_3)
```

```
# poly model with "medv"
poly_model_medv <- lm(crim ~ poly(medv, 3))
smry_poly_model_medv <- summary(poly_model_medv)
print(smry_poly_model_medv)
```

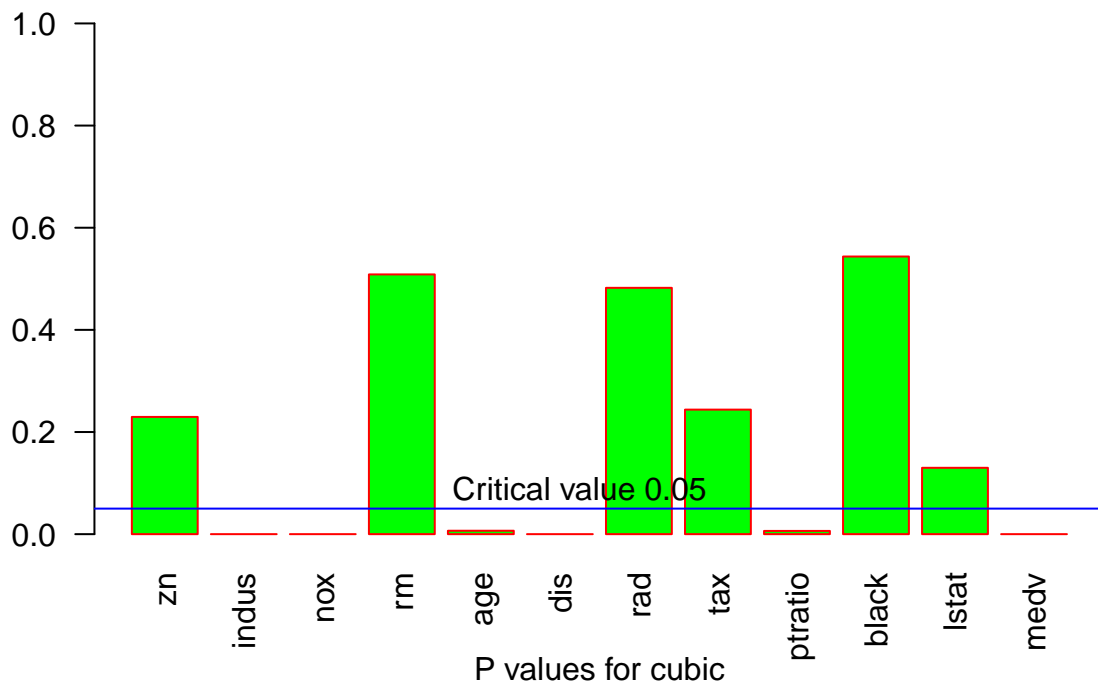
```
##
## Call:
## lm(formula = crim ~ poly(medv, 3))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.614     0.292  12.374  < 2e-16 ***
## poly(medv, 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
```

```
poly_pvalue_medv <- smry_poly_model_medv$coefficients[2,4]
poly_pvalue_medv_2 <- smry_poly_model_medv$coefficients[3,4]
poly_pvalue_medv_3 <- smry_poly_model_medv$coefficients[4,4]
poly_p_values <- append(poly_p_values, poly_pvalue_medv)
poly_p_values_2 <- append(poly_p_values_2, poly_pvalue_medv_2)
poly_p_values_3 <- append(poly_p_values_3, poly_pvalue_medv_3)
```

```
# Plot the bar chart on P values for quadratic
barplot(poly_p_values_2,las=2,names.arg=colnames(boston_df)[-c(1,4)],xlab="P values for quadratic",ylim = c(0,1))
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.09, "Critical value 0.05")
```



```
# Plot the bar chart on P values for cubic
barplot(poly_p_values_3,las=2,names.arg=colnames(boston_df)[-c(1,4)],xlab="P values for cubic",ylim = c(0,1))
abline(h=0.05,lwd=1, lty="solid", col="blue")
text(7, 0.09, "Critical value 0.05")
```



```
print("From both graph P value for quandratic and p value for cubic we can conclude that 'zn', 'rm', 'r
```

```
## [1] "From both graph P value for quandratic and p value for cubic we can conclude that 'zn',
'rm', 'rad', 'tax', 'black' and 'lstat' as perdictor are not statistically significant but for
other predictors like 'indus', 'nox', 'age', 'dis', 'ptratio' and 'medv' with acceptable p value.
In case of full model incuding all predictors in non-linear is not visible"
```