ST552 Homework 5

Brian Cervantes Alvarez
February 27, 2024

Problem 1

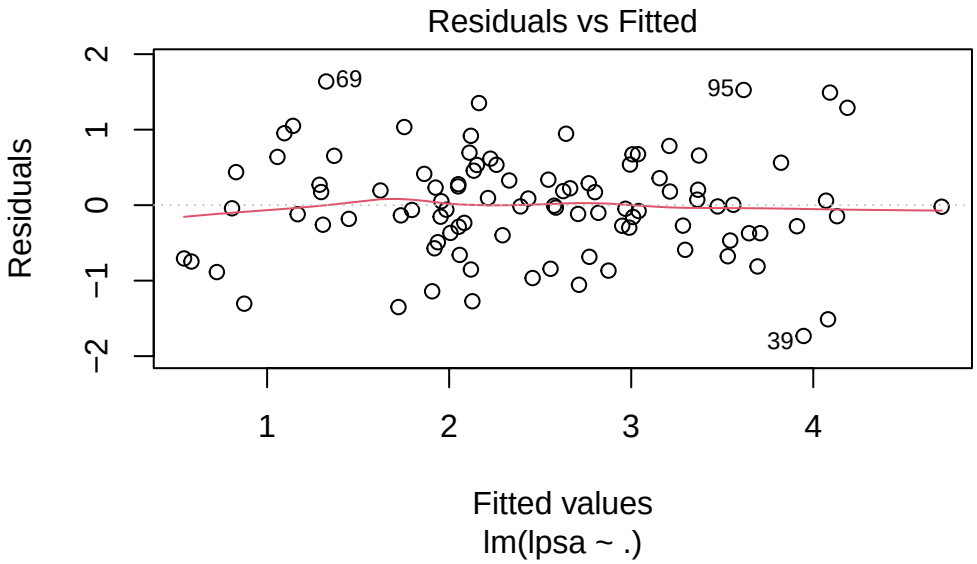
```r
library(faraway)
library(car)
library(ggplot2)

data(prostate)
prostateModel <- lm(lpsa ~ ., data = prostate)
summary(prostateModel)
```

```
Call:
lm(formula = lpsa ~ ., data = prostate)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7331 -0.3713 -0.0170  0.4141  1.6381

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.669337   1.296387   0.516  0.60693
lcavol       0.587022   0.087920   6.677 2.11e-09 ***
lweight      0.454467   0.170012   2.673  0.00896 **
age         -0.019637   0.011173  -1.758  0.08229 .
lbph         0.107054   0.058449   1.832  0.07040 .
svi          0.766157   0.244309   3.136  0.00233 **
lcp         -0.105474   0.091013  -1.159  0.24964
gleason      0.045142   0.157465   0.287  0.77503
pgg45        0.004525   0.004421   1.024  0.30886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7084 on 88 degrees of freedom
Multiple R-squared:  0.6548,    Adjusted R-squared:  0.6234
F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

Part A

Does this follow Homoscedasticity?: The plot of residuals versus fitted values does not show significant deviations from constant variance. Although there is a slight increase in variance between fitted values of 1 and 2, it levels off close to 0. This slight fluctuation may not be substantial enough to invalidate the model but warrants a closer inspection. Overall, the data appears to follow the homoscedasticity assumption.

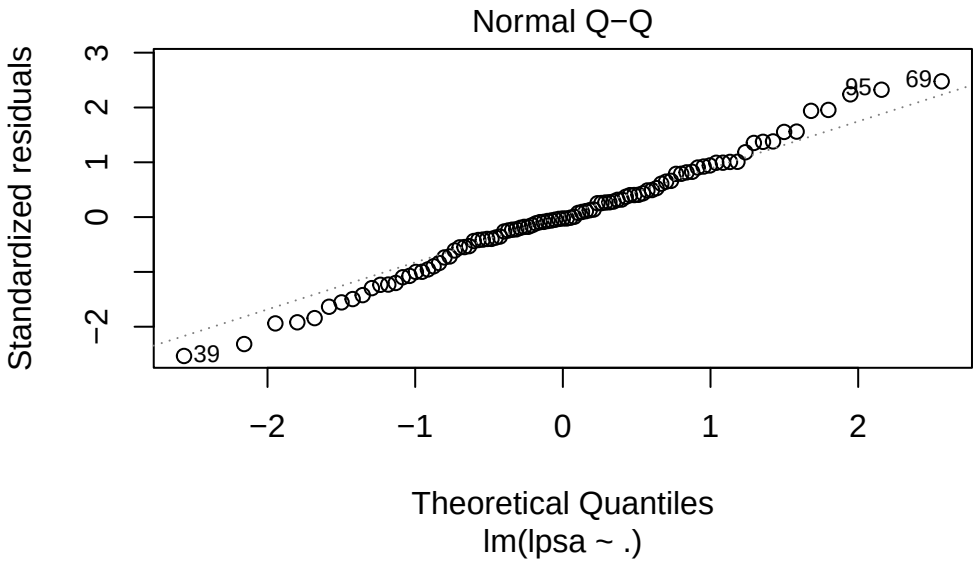
```
plot(prostateModel, which = 1)
```



Part B

Are the residuals following normality?: According to the Q-Q plot, we can safely suggest that the residuals are following normality. While the tails deviate slightly from normality, the majority of the residuals appear to follow a normal distribution.

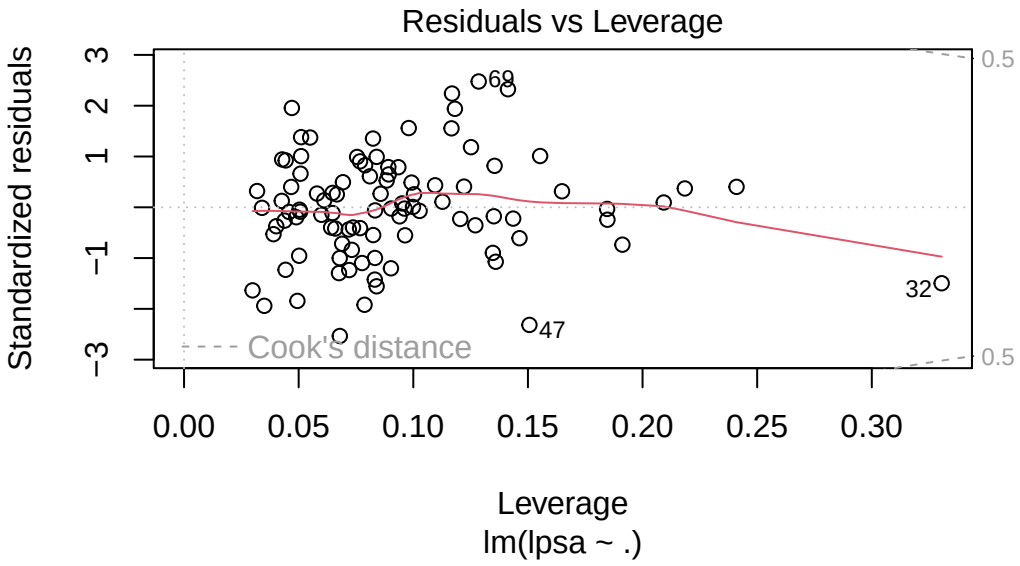
```
plot(prostateModel, which = 2)
```



Part C

Are there any extreme predictor values in our leverage plot?: After standardizing the residuals, we can see that there appears to be a predictor value, specifically 32, that is influencing the regression line. Since this point has high leverage, it may be distorting the outcome and interpretation of the regression model. We need to look into this data value and determine if we should or should not remove it. We would need to look at cook's distance next.

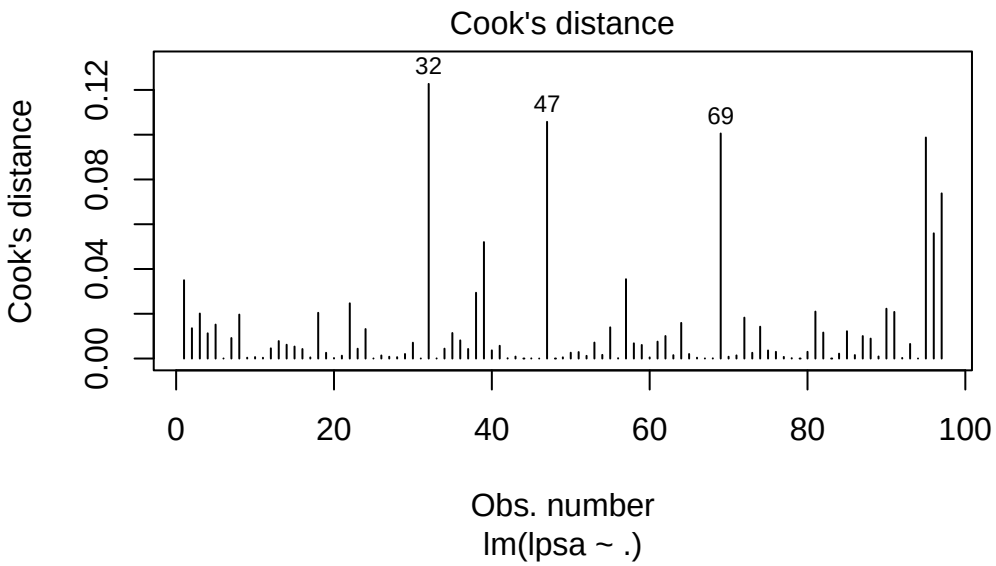
```
plot(prostateModel, which = 5)
```



Part D

Similar to leverage, are there any outliers using Cook's Distance?: If we consider the 3 largest cook's distance observations, we can notice that data value 32 is persistent in being a potential outlier. Given that observation 32 showed up in the residual vs leverage plot, it would be proper practice to fully investigate this value. While observations 47, and 69 have second and third highest cook's distance, they would be included in the model since they did appear to be that influential in the residual vs leverage plot.

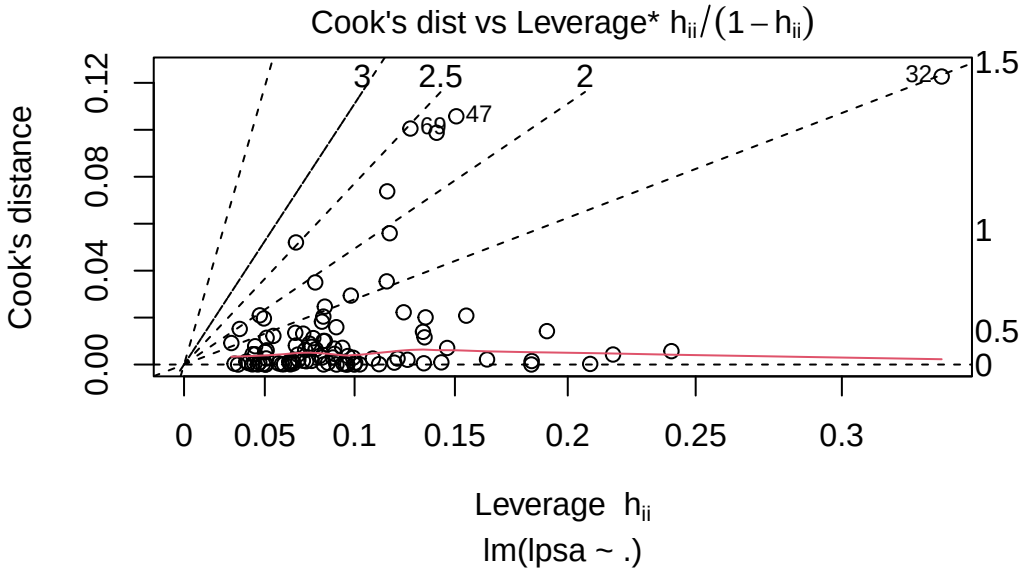
```
plot(prostateModel, which = 4)
```



Part E

Cook's Distance vs Leverage, which data values should we narrow down?: When we look at this plot, we need to consider the high leverage and high influence points. Clearly, there sits on point which has a concerning one and is observation 32. Given that it has both high leverage (far to the right) and high influence (higher up), we can set our exceptions straight and consider removing observation 32 from the model. While observations 69 and 47 have moderately low leverage, but high influence, I would still consider leaving them in the model. The reasoning behind this is those values are not that influential compared to observation 32. Additionally, there could be loss of information and those points could be valid variances. Therefore, we must be careful in deciding the removal of this points.

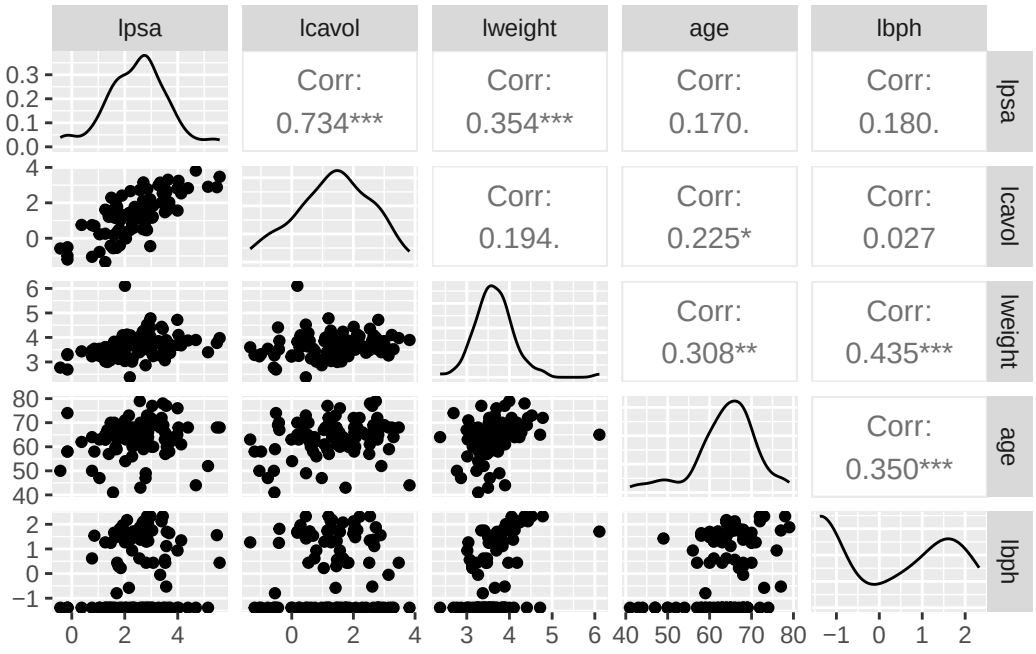
```
plot(prostateModel, which = 6)
```



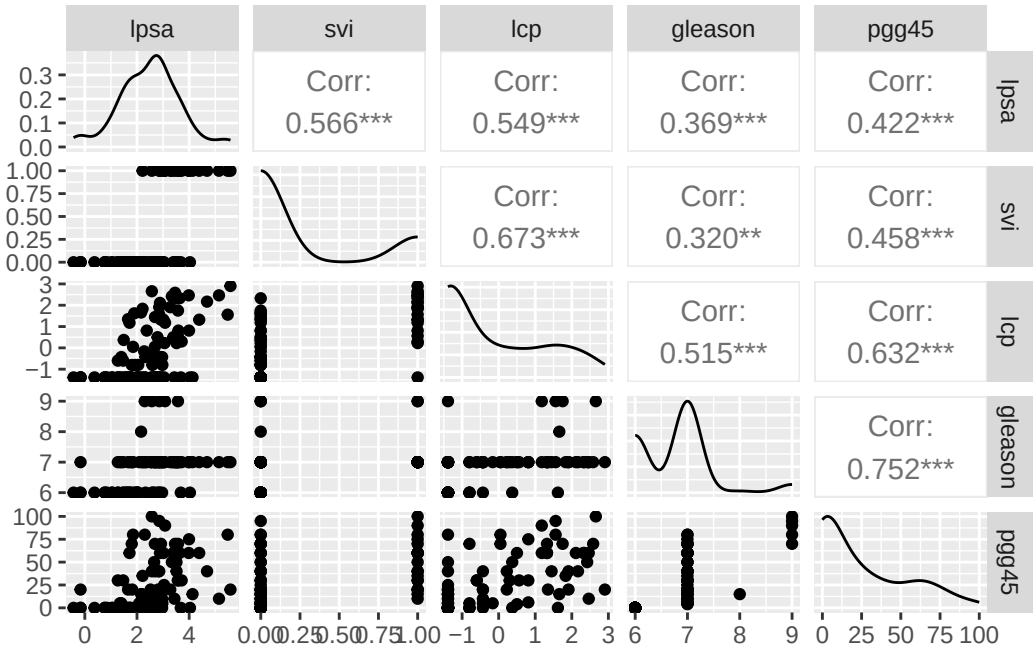
Part F

Looking at the pairs plots for correlation: From the pairs plots, we can see that we have strong linear correlation between the response variable, **lpsa**, and predictors **lacavol**, **lcp and svi**. We have moderate correlation with **lpsa** vs. **gleason, pgg45** and **lweight.** We have weak correlation for **lpsa** vs. **age** and **lbph.** I used ggpairs from the ggsally package so it can give me their correlation values and their distributions. You can clearly see the normally distributed and non-normal distributions for a few of the predictors. One of them is binary and the other categorical which can give insight into their distributions and correlation to the response variable. Generally, if we noticed that most of the these predictors did not behave in a linear sense, then we would have to change our modeling technique to fit the data in a more appropriate manner.

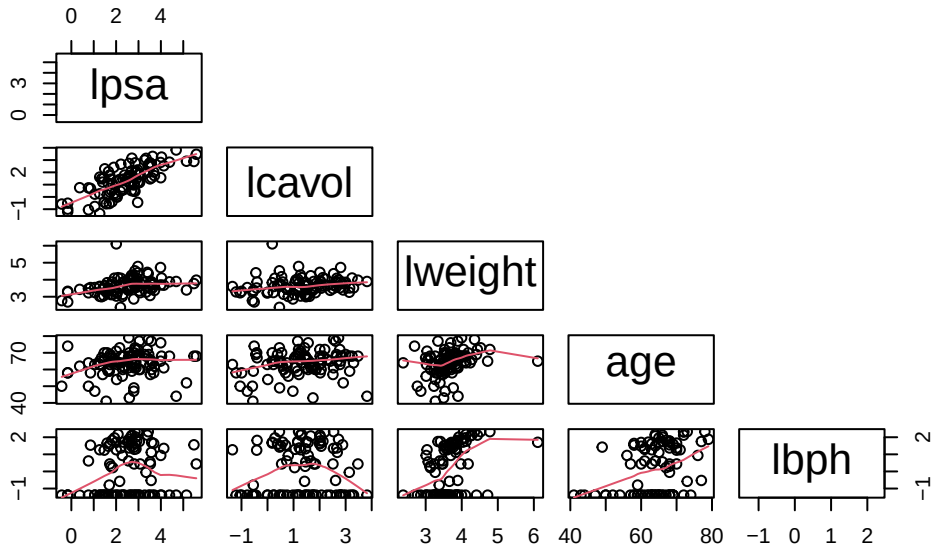
```
library("GGally")
ggpairs(prostate,
        columns = c("lpsa", "lcavol", "lweight", "age", "lbph"),
        progress = FALSE)
```



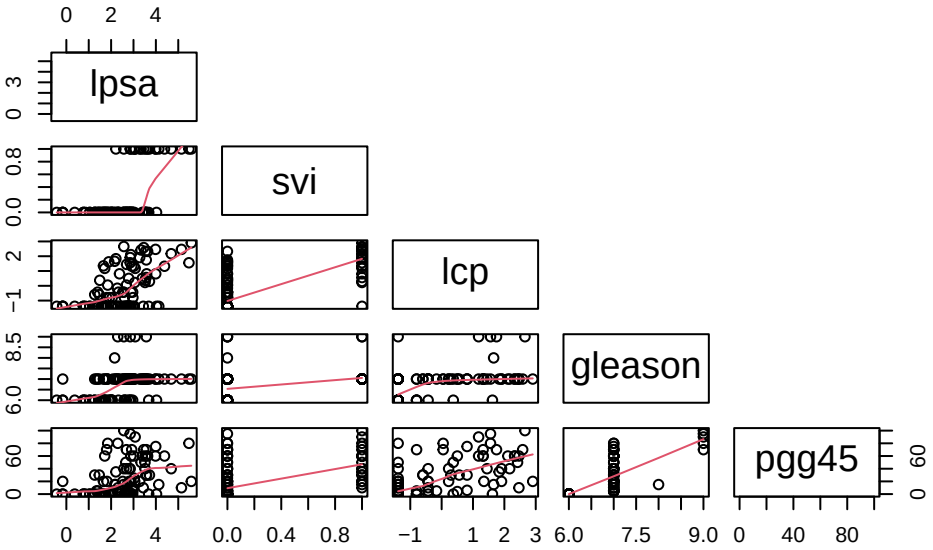

```
ggpairs(prostate,
        columns = c("lpsa", "svi", "lcp", "gleason", "pgg45"),
        progress = FALSE)
```




```r
# Base R
pairs(~ lpsa + lcavol + lweight + age + lbph,
      data = prostate,
      upper.panel = NULL,
      lower.panel = panel.smooth)
```




```r
pairs(~ lpsa + svi + lcp + gleason + pgg45,
      data = prostate,
      upper.panel = NULL,
      lower.panel = panel.smooth)
```


lpsa
svi
lcp
gleason
pgg45

Problem 2

Part A

Here is our model design:

$$\text{lcavol} = \beta_0 + \beta_1 \times \text{lweight} + \epsilon$$

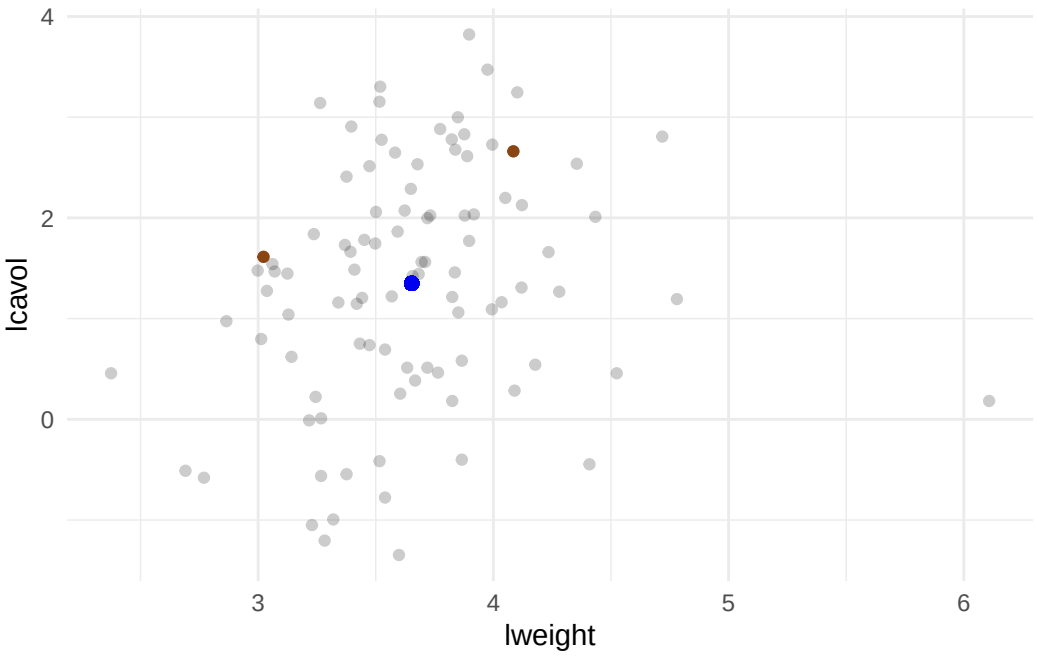
```
ds <- prostate

lcavol <- ds$lcavol
lweight <- ds$lweight

model <- lm(lcavol ~ lweight, data = ds)
summary(model)
```

```
Call:
lm(formula = lcavol ~ lweight, data = ds)

Residuals:
     Min       1Q   Median       3Q      Max
-2.67221 -0.86607  0.07841  0.73684  2.35848

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.3328     0.8804  -0.378   0.7062
lweight       0.4607     0.2389   1.929   0.0567 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.162 on 95 degrees of freedom
Multiple R-squared:  0.03769,   Adjusted R-squared:  0.02756
F-statistic:  3.72 on 1 and 95 DF,  p-value: 0.05674
```

Part B

A scatterplot is drawn with **lweight** on the x-axis and **lcavol** on the y-axis. Observations 13 and 39 are highlighted in the color `chocalate4` (yes, it's a good color), and the center of the data in color `blue2`.

```
library(ggplot2)

# Calculate the means
mean_lcavol <- mean(ds$lcavol)
mean_lweight <- mean(ds$lweight)

pointsOfInterest <- subset(ds, row.names(ds) %in% c('13', '39'))

# Create scatterplot
ggplot(ds, aes(x = lweight, y = lcavol)) +
  geom_point(alpha = 0.2) +
  geom_point(data = pointsOfInterest, color = 'chocolate4') +
  geom_point(aes(x = mean_lweight, y = mean_lcavol), color = 'blue2', size = 2) +
  theme_minimal()
```



Part C

- **Low Leverage:** Both observations have minimal influence on the regression line fit.
- **Squared Euclidean Distance:** Observation 13 is centrally located, suggesting typicality, while observation 39 is much further away, potentially hinting at outlier status.
- **Squared Mahalanobis Distance:** Observation 13's low value indicates close alignment with the overall data distribution. In contrast, observation 39's high value flags it as a potential multivariate outlier.

Now, while both observations have low influence, observation 39 raises concerns as a potential outlier based on its distance from the data's center.

```
# Leverage calculations by hand
X <- as.matrix(cbind(1, ds[, c("lcavol", "lweight")]))
H <- X %*% solve(t(X) %*% X) %*% t(X)
leverages <- diag(H)

# Calculating squared Euclidean distances
meanVals <- colMeans(X)
cX <- X - meanVals
euclidDist <- rowSums(cX^2)

# Calculating squared Mahalanobis distances
covInv <- solve(cov(cX))
mahalanobisDist <- apply(cX, 1, function(row) t(row) %*% covInv %*% row)

# Extracting specific observations
leveragesObs <- leverages[c(13, 39)]
euclideanObs <- euclidDist[c(13, 39)]
mahalanobisObs <- mahalanobisDist[c(13, 39)]

observations <- list(leverage = leveragesObs,
                     euclideanSq = euclideanObs,
                     mahalanobisSq = mahalanobisObs)

print(observations)
```

```
$leverage
        13         39
0.02944870 0.02783697

$euclideanSq
       13        39
 0.466024 17.276260
```

```
$mahalanobisSq
       13        39
0.3159195 7.4835707
```

Part D

This measure adjusts the squared Mahalanobis distance by adding a term that accounts for the size of the dataset. This adjustment does not increase the mahalanobis squared distance of observations 13 and 39 but a siginficant amount. The increase is very minimal and does not drastically change the values. It also shows that this adjustment distance is not zero which could be useful in a non-zero baseline context.

```
n <- nrow(X)
mahalanobisAdj <- 1/n + mahalanobisObs
list(mahalanobisAdj = mahalanobisAdj)
```

```
$mahalanobisAdj
       13        39
0.3262288 7.4938800
```

Problem 3

Part A

We know that the condition number assesses the multicollinearity of the predictors. The condition number is 824.4962, which is significantly above 30. This heavily suggests that there is severe multicollinearity, which can affect the stability and interpretability of the regression coefficients.

```
model <- lm(lpsa ~ ., data = ds)
# Compute condition number
cn <- kappa(model.matrix(model))
cn
```

```
[1] 824.4962
```

Part B

The correlation matrix shows varying degrees of association between different variables. First, **lcavol** and **lcp** have a relatively high correlation (0.675), suggesting potential ***multicollinearity***. Similarly, **lcp** and **pgg45** (0.632), as well as **gleason** and **pgg45** (0.752), show significant correlations, indicating strong relationships. However, other pairs, like **lweight** and **gleason** or **lbph** and **svi**, show very low correlations, indicating that not all variables are strongly associated. This mixed pattern indicates that we should be careful when selecting the variables to include in our models. That would help to minimize and/or avoid multicollinearity issues in the regression analysis.

```
# Compute correlations between predictors
predCor <- cor(ds[, -which(names(ds) == "lpsa")])
predCor
```

```
            lcavol      lweight       age        lbph         svi         lcp
lcavol   1.00000000  0.194128387 0.2249999  0.02734971  0.53884500  0.67531058
lweight  0.19412839  1.000000000 0.3075247  0.43493174  0.10877818  0.10023889
age      0.22499988  0.307524741 1.0000000  0.35018592  0.11765804  0.12766778
lbph     0.02734971  0.434931744 0.3501859  1.00000000 -0.08584327 -0.00699944
svi      0.53884500  0.108778185 0.1176580 -0.08584327  1.00000000  0.67311122
lcp      0.67531058  0.100238891 0.1276678 -0.00699944  0.67311122  1.00000000
gleason  0.43241705 -0.001283003 0.2688916  0.07782044  0.32041222  0.51482991
pgg45    0.43365224  0.050846195 0.2761124  0.07846000  0.45764762  0.63152807
             gleason     pgg45
lcavol   0.432417052 0.4336522
lweight -0.001283003 0.0508462
age      0.268891599 0.2761124
lbph     0.077820444 0.0784600
svi      0.320412221 0.4576476
lcp      0.514829912 0.6315281
gleason  1.000000000 0.7519045
pgg45    0.751904512 1.0000000
```

Part C

```
vifVals <- vif(model)
vifDs <- data.frame(Predictor = names(vifVals), VIF = unname(vifVals))
print(vifDs)
```

```
  Predictor      VIF
1    lcavol 2.054115
2   lweight 1.363704
3       age 1.323599
4      lbph 1.375534
5       svi 1.956881
6       lcp 3.097954
7   gleason 2.473411
8     pgg45 2.974361
```

The Variance Inflation Factors for the predictors show differing levels of multicollinearity. Specifically, **lcp** has the highest VIF value of 3.097954, suggesting moderate multicollinearity with other predictors. In contrast, **age** has the lowest VIF value of 1.323599, showing minimal multicollinearity. However, all VIF values are below the common threshold of 10, which means that while there is some multicollinearity present, it may not severely impact the regression estimates.

Problem 4

Part A

The predictor **Lactic** is statistically significant in the model, with a p-value of 0.03108, indicating a significant association with response variable **taste**.

```r
data(cheddar)
model <- lm(taste ~ ., data = cheddar)
summary(model)
```

```
Call:
lm(formula = taste ~ ., data = cheddar)

Residuals:
    Min      1Q  Median      3Q     Max
-17.390  -6.612  -1.009   4.908  25.449

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -28.8768    19.7354  -1.463  0.15540
Acetic        0.3277     4.4598   0.073  0.94198
H2S           3.9118     1.2484   3.133  0.00425 **
Lactic       19.6705     8.6291   2.280  0.03108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,    Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Part B

```
pval <- summary(model)$coefficients["Lactic", 4]
pval
```

```
[1] 0.03107948
```

Part C

By adding normally distributed errors, $\epsilon = N(0, 0.01)$, to the **Lactic** predictor may enhance the model's fit by introducing slight variability that could align more closely with the inherent noise in the response variable **taste**. This can potentially decrease the residual variance, improving the precision of the estimated relationship between **Lactic** and **taste** and possibly leading to a lower p-value, albeit with variability across different simulations. To ascertain the stability and significance of this effect, a comprehensive simulation would be necessary to observe the distribution and convergence of the p-values.

```
ds <- cheddar
ds$LacticWithError <- ds$Lactic + rnorm(n = nrow(cheddar), mean = 0, sd = 0.01)
modelWithError <- lm(taste ~ LacticWithError + Acetic + H2S, data = ds)
summary(modelWithError)
```

```
Call:
lm(formula = taste ~ LacticWithError + Acetic + H2S, data = ds)

Residuals:
     Min       1Q   Median       3Q      Max
-17.3174  -6.7353  -0.7419   5.2030  25.1761

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -28.6670    19.6631  -1.458  0.15684
LacticWithError  19.9027     8.5546   2.327  0.02805 *
Acetic            0.2753     4.4434   0.062  0.95106
H2S               3.8870     1.2448   3.122  0.00436 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.1 on 26 degrees of freedom
Multiple R-squared:  0.6542,    Adjusted R-squared:  0.6143
F-statistic: 16.39 on 3 and 26 DF,  p-value: 3.489e-06
```

```
summary(modelWithError)$coefficients["LacticWithError", 4]
```

```
[1] 0.02805062
```

Part D

Upon running the simulation, we observed an average p-value of 0.03142896, which closes aligns with the original p-value of 0.03108 without error modification for the **Lactic** predictor. This outcome suggests adding a negligible level of measurement error, characterized by a standard deviation of 0.01, does not substantively alter the statistical significance of the **Lactic** predictor within the context of taste as the response variable. Therefore, the robustness of Lactic as a significant predictor in the model remains effectively unchallenged by such minimal measurement variability.

```
ds <- cheddar

pvals <- numeric(1000)

for(i in 1:1000) {
  ds$LacticWithError <- ds$Lactic + rnorm(n = nrow(ds), mean = 0, sd = 0.01)
  modelWithError <- lm(taste ~ LacticWithError + Acetic + H2S, data = ds)
  pvals[i] <- summary(modelWithError)$coefficients["LacticWithError", 4]
}

meanPval <- mean(pvals)
meanPval
```

```
[1] 0.03152271
```

Part E

In this situation where the standard deviation is 0.1 and introduced to the **Lactic** variable, the resulting p-values range from 0.06 to 0.07. This variability indicates a significant deviation from the initial analysis, suggesting that the **Lactic** variable's statistical significance in predicting taste is compromised. This shows that by having an increased level of measurement error, the **Lactic** predictor's reliability and influence on the taste outcome are brought into question, reflecting the impact substantial measurement errors can have on the conclusions drawn from regression analysis. That is why if we know the measurement error or have a rough estimate of it, that can impact the final conclusions of regression analysis.

```
ds <- cheddar

pvals <- numeric(1000)

for(i in 1:1000) {
  ds$LacticError2 <- ds$Lactic + rnorm(n = nrow(ds), mean = 0, sd = 0.1)
  modelwithHighError <- lm(taste ~ LacticError2 + Acetic + H2S, data = ds)
  pvals[i] <- summary(modelwithHighError)$coefficients["LacticError2", 4]
}

meanPvals <- mean(pvals)
meanPvals
```

```
[1] 0.06888294
```

Problem 5

Part A

The high condition number, 91752.66 suggests that the model's numerical computations are highly sensitive to small changes in the data, which can make the estimates of the coefficients unreliable. The VIF results reveal significant multicollinearity in the fat dataset; Particularly, with **weight**, **siri**, and **density** showing exceptionally high VIF values, indicating these predictors share substantial linear relationships with other variables. In contrast, variables like **age**, **height**, and **ankle** display lower VIFs, suggesting they are less mixed up with the remaining predictors.

```
data(fat)
fatModel <- lm(brozek ~ ., data = fat)
# Compute condition numbers
cn <- kappa(model.matrix(fatModel))
cn
```

```
[1] 91752.66
```

```
# Compute variance inflation factors
vif <- vif(fatModel)
vif
```

```
     siri   density       age    weight    height    adipos      free      neck
74.875747 43.859273  2.291686 97.610708  2.284969 17.869956 56.180584  4.529823
    chest     abdom       hip     thigh      knee     ankle    biceps   forearm
11.339983 19.605509 15.376440  8.416989  4.870155  1.987986  3.769817  2.300792
    wrist
 3.572162
```

Part B

After removing cases 39 and 42, the condition number slightly decreased from $91,752.66$ to $85,317.34$, showing a minor reduction in multicollinearity. Despite this, the VIF values for **weight** and **adipos** significantly increased, creating new multicollinearity issues for these variables compared to the original model, while **height** also showed a notable rise in its VIF value, suggesting increased collinearity concerns post-removal.

```
# Exclude cases 39 and 42
fatReduced <- fat[-c(39, 42), ]
fatModelReduced <- lm(brozek ~ ., data = fatReduced)
# Recompute condition numbers
cn <- kappa(model.matrix(fatModelReduced))
cn
```

```
[1] 85317.34
```

```
# Recompute variance inflation factors
vif <- vif(fatModelReduced)
vif
```

```
     siri    density        age     weight     height     adipos       free
81.015759  42.943053   2.347345 153.951427  32.638667  99.748574  62.467500
     neck      chest      abdom        hip      thigh       knee      ankle
 4.091993  11.429787  17.529752  12.708440   7.582612   4.465379   1.880402
   biceps    forearm      wrist
 3.524484   2.470163   3.455211
```

Part C

Comparing the simplified to the original model, the condition number drastically reduces to $2,965.58$ compared to 91752.66, indicating a significant decrease in multicollinearity. Additionally, VIF values for **age**, **weight**, and **height** are all near 1, again, demonstrating a substantial reduction in multicollinearity compared to the comprehensive model.

```
# Fit the model with limited predictors
fatModelLimited <- lm(brozek ~ age + weight + height, data = fat)

# Compute condition numbers
cn <- kappa(model.matrix(fatModelLimited))
cn
```

```
[1] 2965.58
```

```
# Compute variance inflation factors
vif <- vif(fatModelLimited)
vif
```

```
     age   weight   height
1.032253 1.107050 1.140470
```