Exercise 19-21 Assignment

Alex

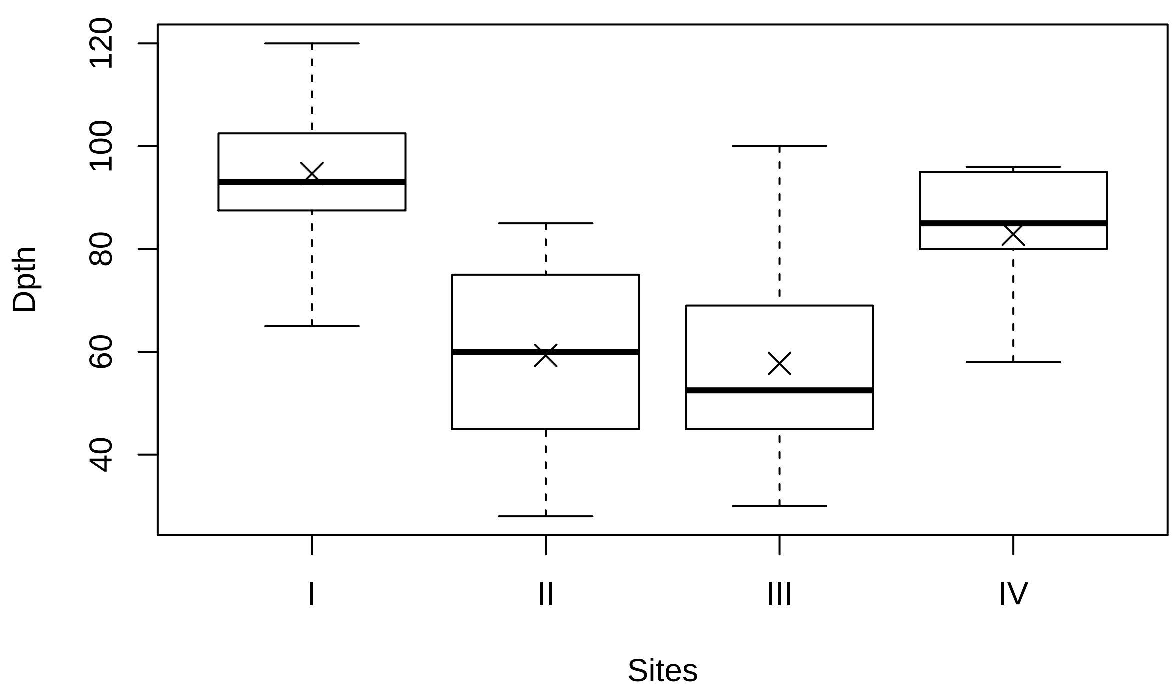
16/10/2019

#Chapter 19 Exercise 19.1

```
## 19.1
#Storing in two vectors records of depth and site of observation
Dpth <- c(93,120,65,105,115,82,99,87,100,90,78,95,93,88,110,85,45,80,28,75,70,65,55,50,40,100,75,65,40,
Sites <- c(rep("I",15),rep("II",10),rep("III",12),rep("IV",9))
# Question a
Dpth_means <- tapply(Dpth,INDEX=Sites,FUN=mean)
Dpth_means
```

```
##        I       II      III       IV
## 94.66667 59.30000 57.75000 82.88889
```

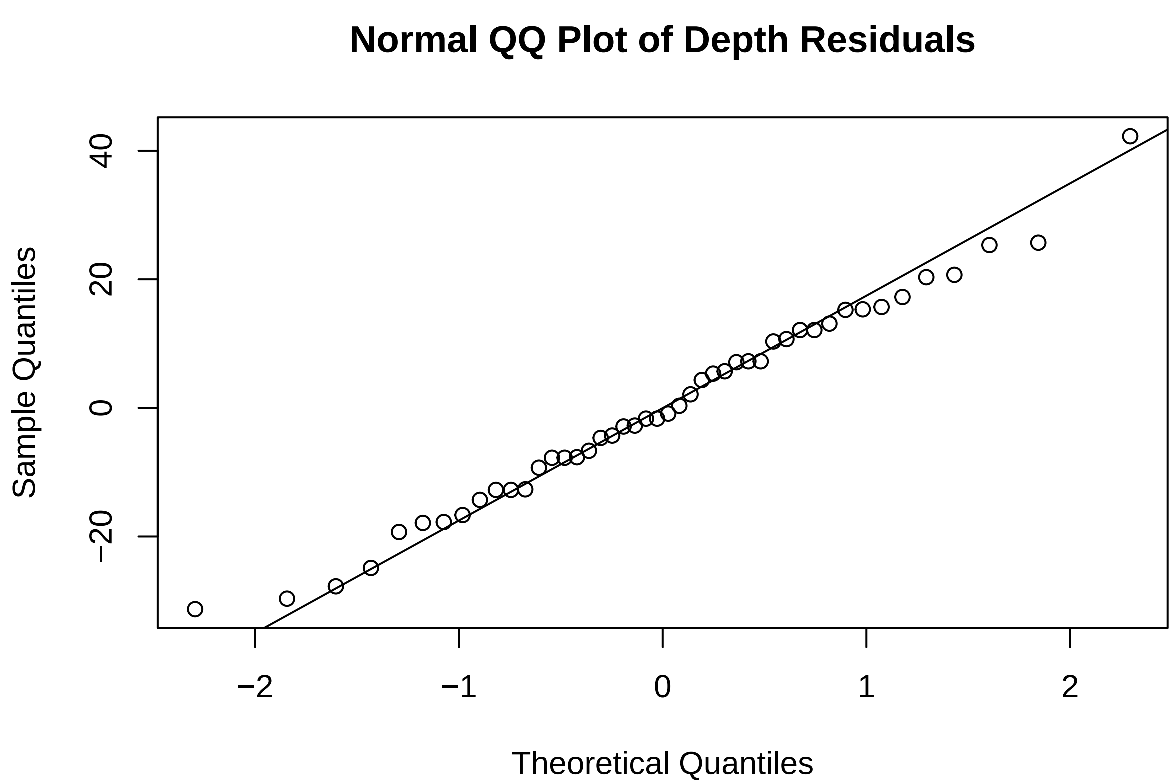
```
boxplot(Dpth~Sites)
points(1:4,Dpth_means,pch=4, cex=1.5)
```



```r
# Question b
#diagnostic checks for normality & equality of Variances
Dpth_meancen <- Dpth-rep(Dpth_means,table(Sites))
Dpth_meancen
```

```
##           I           I           I           I           I           I
##  -1.6666667  25.3333333 -29.6666667  10.3333333  20.3333333 -12.6666667
##           I           I           I           I           I           I
##   4.3333333  -7.6666667   5.3333333  -4.6666667 -16.6666667   0.3333333
##           I           I           I          II          II          II
##  -1.6666667  -6.6666667  15.3333333  25.7000000 -14.3000000  20.7000000
##          II          II          II          II          II          II
## -31.3000000  15.7000000  10.7000000   5.7000000  -4.3000000  -9.3000000
##          II         III         III         III         III         III
## -19.3000000  42.2500000  17.2500000   7.2500000 -17.7500000  15.2500000
##         III         III         III         III         III         III
##   7.2500000  -7.7500000 -27.7500000 -12.7500000  -7.7500000 -12.7500000
##         III          IV          IV          IV          IV          IV
##  -2.7500000  13.1111111 -24.8888889  12.1111111   7.1111111 -17.8888889
##          IV          IV          IV          IV
##  -2.8888889   2.1111111  12.1111111  -0.8888889
```

```r
qqnorm(Dpth_meancen, main = "Normal QQ Plot of Depth Residuals")
qqline(Dpth_meancen)
```



```r
#ii Variance
sds <- tapply(Dpth,INDEX=Sites,FUN=sd)
sds
```

```
##        I       II      III       IV
## 14.37591 18.69076 18.96947 13.55032
```

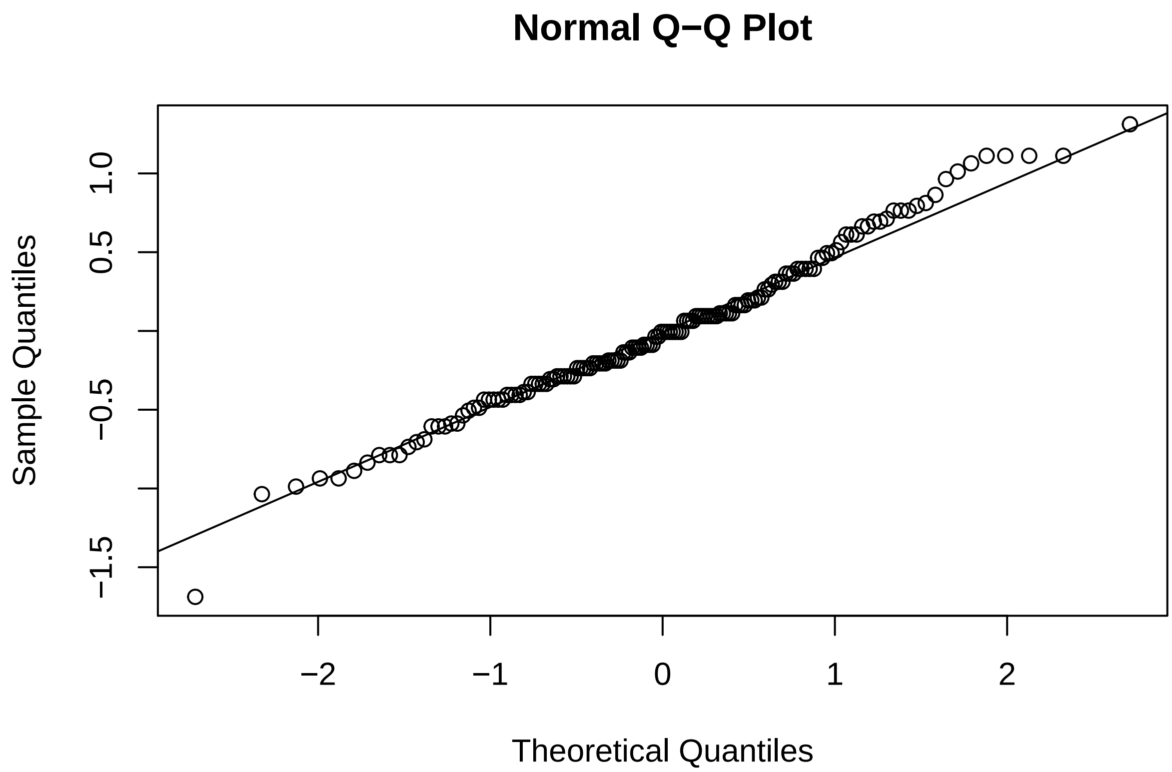
```r
eqlvar<- max(sds)/min(sds)
eqlvar# Less than 2 so variances can be assumed equal according to the rule-of-thumb.
```

```
## [1] 1.399928
```

```r
# Question c
summary(aov(Dpth~Sites)) # Small p-value. Very strong evidence against H0. There is evidence to reject
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Sites        3  12397    4132   15.14 7.99e-07 ***
## Residuals   42  11465     273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

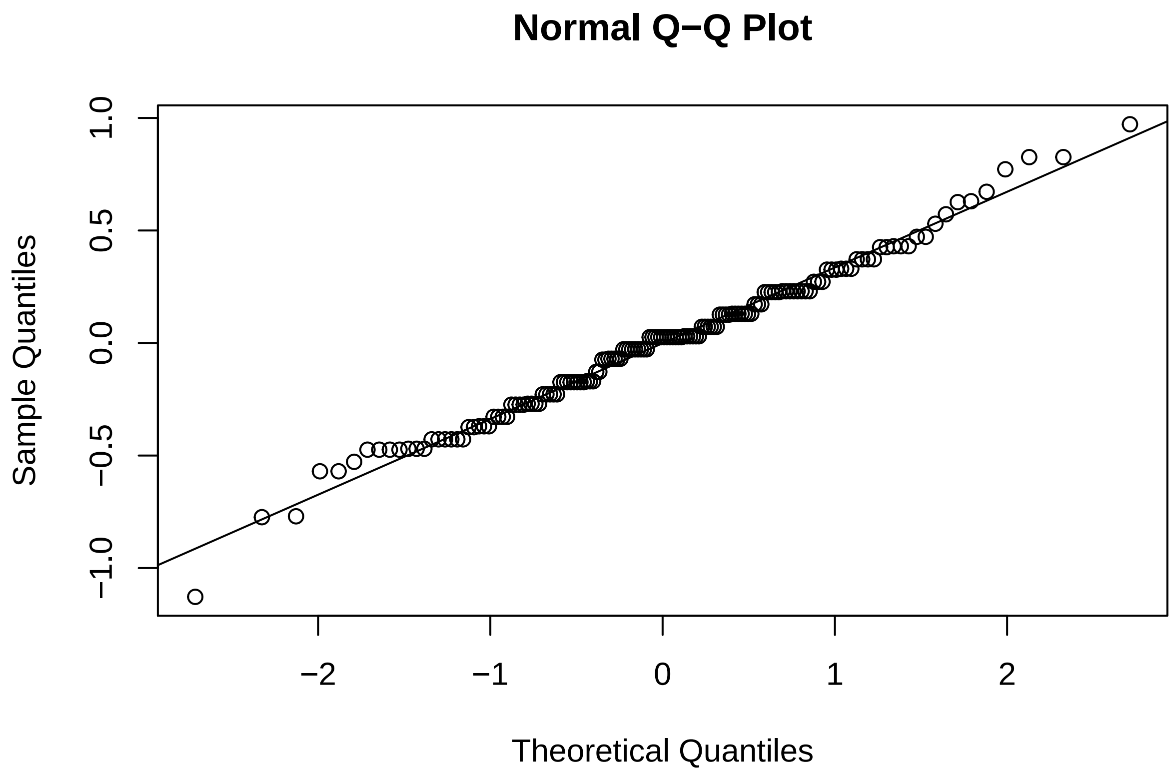
```r
# Question d
data("iris")
meanval <- tapply(iris$Sepal.Length,iris$Species,FUN=mean)
mc <- iris$Sepal.Length-meanval[as.numeric(iris$Species)]
qqnorm(mc)
qqline(mc)
```




```
tapply(iris$Sepal.Length,iris$Species,sd) # max(sd)/min(sd) < 2 so variances may be assumed equal.
```

```
##     setosa versicolor  virginica
##  0.3524897  0.5161711  0.6358796
```

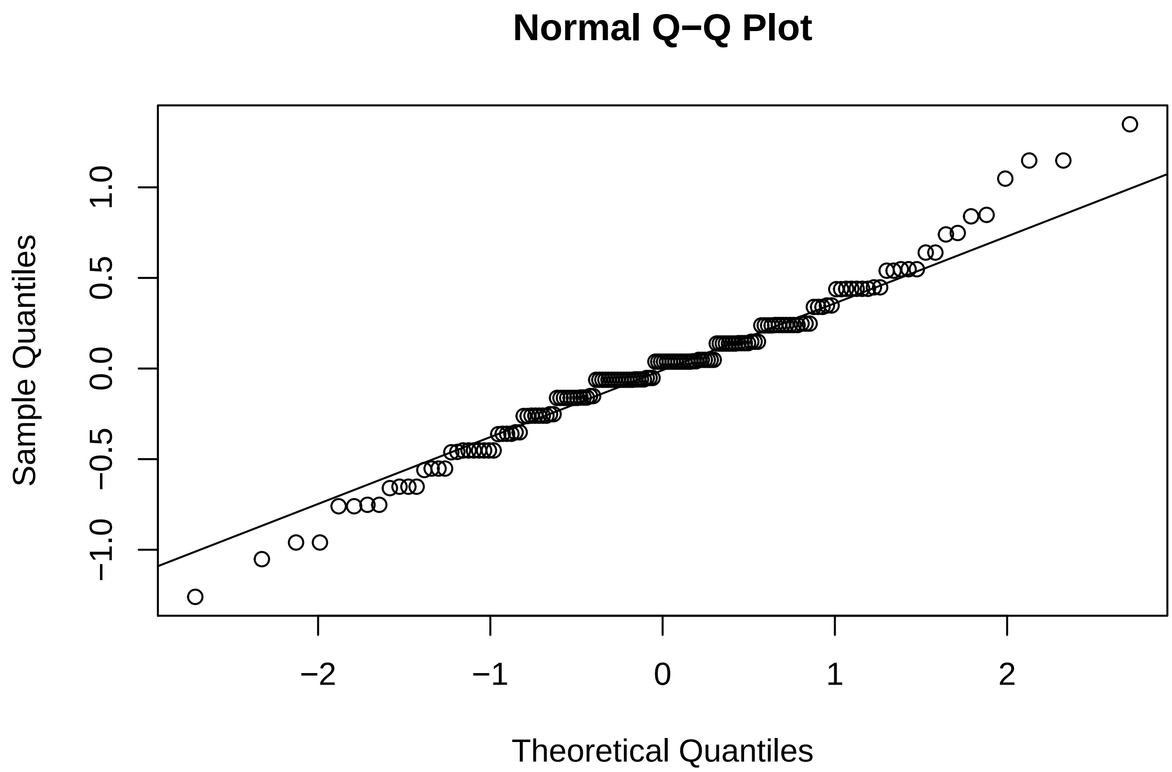
```
meanval2 <- tapply(iris$Sepal.Width,iris$Species,FUN=mean)
mc <- iris$Sepal.Width-meanval2[as.numeric(iris$Species)]
qqnorm(mc)
qqline(mc) # Looks approximately normal.
```




```r
tapply(iris$Sepal.Width,iris$Species,FUN=sd) # max(sd)/min(sd) < 2 so variances may be assumed equal.
```

```
##     setosa versicolor  virginica
##  0.3790644  0.3137983  0.3224966
```

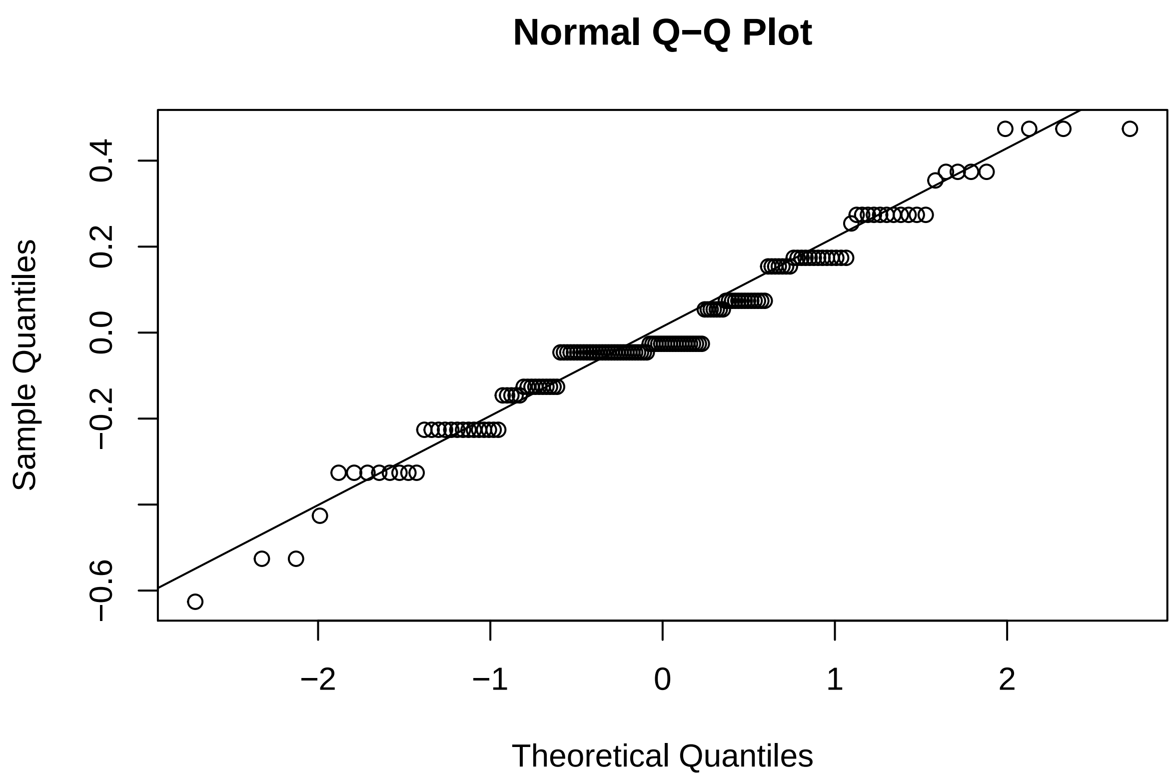
```r
meanval3 <- tapply(iris$Petal.Length,iris$Species,FUN=mean)
mc <- iris$Petal.Length-meanval3[as.numeric(iris$Species)]
qqnorm(mc)
qqline(mc) # Looks approximately normal; some deviation though.
```




```
tapply(iris$Petal.Length,iris$Species,sd) # max(sd)/min(sd) > 2 so variances may not be assumed equal.
```

```
##     setosa versicolor  virginica
##  0.1736640  0.4699110  0.5518947
```

```
meanval4 <- tapply(iris$Petal.Width,iris$Species,FUN=mean)
mc <- iris$Petal.Width-meanval4[as.numeric(iris$Species)]
qqnorm(mc)
qqline(mc) # Looks approximately normal.
```




```r
tapply(iris$Petal.Width,iris$Species,FUN=sd) # max(sd)/min(sd) > 2 so variances may not be assumed equa
```

```
##     setosa versicolor  virginica
##  0.1053856  0.1977527  0.2746501
```

```r
# Question e
summary(aov(Sepal.Length~Species,data=iris))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## Species       2  63.21  31.606   119.3 <2e-16 ***
## Residuals   147  38.96   0.265                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(aov(Sepal.Width~Species,data=iris))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## Species       2  11.35   5.672   49.16 <2e-16 ***
## Residuals   147  16.96   0.115                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Both p-values are very small. Strong evidence to reject H0 and conclude that the mean sepal lengths a
```

#Exercise 19.2

```r
## 19.2
# Question a
data("quakes")
depth_fac_events <- cut(quakes$depth,breaks=c(0,200,400,680))
depth_fac_events
```

```
##    [1] (400,680] (400,680] (0,200]   (400,680] (400,680] (0,200]
##    [7] (0,200]   (0,200]   (200,400] (400,680] (400,680] (200,400]
##   [13] (400,680] (400,680] (0,200]   (200,400] (0,200]   (400,680]
##   [19] (400,680] (400,680] (400,680] (200,400] (400,680] (0,200]
##   [25] (400,680] (400,680] (0,200]   (400,680] (400,680] (200,400]
##   [31] (400,680] (0,200]   (200,400] (200,400] (0,200]   (400,680]
##   [37] (200,400] (400,680] (400,680] (200,400] (200,400] (0,200]
##   [43] (400,680] (0,200]   (200,400] (0,200]   (200,400] (0,200]
##   [49] (400,680] (0,200]   (400,680] (200,400] (0,200]   (400,680]
##   [55] (400,680] (400,680] (400,680] (400,680] (400,680] (400,680]
##   [61] (400,680] (200,400] (400,680] (0,200]   (400,680] (200,400]
##   [67] (400,680] (400,680] (400,680] (0,200]   (0,200]   (200,400]
##   [73] (0,200]   (400,680] (400,680] (400,680] (0,200]   (200,400]
##   [79] (200,400] (200,400] (0,200]   (200,400] (400,680] (200,400]
##   [85] (400,680] (0,200]   (0,200]   (400,680] (0,200]   (0,200]
##   [91] (0,200]   (200,400] (400,680] (200,400] (200,400] (200,400]
##   [97] (200,400] (200,400] (0,200]   (400,680] (0,200]   (400,680]
##  [103] (400,680] (0,200]   (400,680] (200,400] (0,200]   (0,200]
##  [109] (0,200]   (0,200]   (200,400] (200,400] (400,680] (400,680]
##  [115] (400,680] (400,680] (0,200]   (0,200]   (0,200]   (200,400]
##  [121] (0,200]   (0,200]   (400,680] (400,680] (200,400] (0,200]
##  [127] (400,680] (400,680] (0,200]   (200,400] (0,200]   (400,680]
##  [133] (0,200]   (400,680] (400,680] (0,200]   (0,200]   (0,200]
##  [139] (0,200]   (200,400] (400,680] (400,680] (200,400] (200,400]
##  [145] (400,680] (0,200]   (0,200]   (0,200]   (400,680] (400,680]
##  [151] (0,200]   (0,200]   (200,400] (0,200]   (0,200]   (400,680]
##  [157] (0,200]   (400,680] (0,200]   (0,200]   (400,680] (400,680]
##  [163] (200,400] (0,200]   (0,200]   (0,200]   (200,400] (200,400]
##  [169] (400,680] (200,400] (400,680] (400,680] (400,680] (0,200]
##  [175] (200,400] (200,400] (400,680] (200,400] (400,680] (200,400]
##  [181] (400,680] (0,200]   (0,200]   (200,400] (400,680] (0,200]
##  [187] (200,400] (400,680] (400,680] (200,400] (200,400] (200,400]
##  [193] (400,680] (400,680] (400,680] (400,680] (200,400] (0,200]
##  [199] (400,680] (400,680] (200,400] (400,680] (200,400] (0,200]
##  [205] (400,680] (200,400] (400,680] (400,680] (200,400] (400,680]
##  [211] (400,680] (200,400] (200,400] (0,200]   (400,680] (400,680]
##  [217] (0,200]   (400,680] (200,400] (200,400] (0,200]   (0,200]
##  [223] (0,200]   (200,400] (400,680] (0,200]   (200,400] (0,200]
##  [229] (200,400] (0,200]   (400,680] (200,400] (400,680] (0,200]
##  [235] (400,680] (400,680] (0,200]   (400,680] (0,200]   (400,680]
##  [241] (200,400] (0,200]   (0,200]   (0,200]   (200,400] (400,680]
##  [247] (400,680] (0,200]   (400,680] (0,200]   (0,200]   (0,200]
##  [253] (400,680] (0,200]   (0,200]   (400,680] (200,400] (200,400]
```

```
## [259] (400,680] (400,680] (200,400] (400,680] (0,200]   (400,680]
## [265] (200,400] (0,200]   (0,200]   (0,200]   (400,680] (400,680]
## [271] (0,200]   (400,680] (400,680] (400,680] (400,680] (400,680]
## [277] (400,680] (400,680] (400,680] (400,680] (400,680] (400,680]
## [283] (0,200]   (200,400] (200,400] (400,680] (400,680] (0,200]
## [289] (400,680] (400,680] (400,680] (0,200]   (400,680] (400,680]
## [295] (0,200]   (0,200]   (400,680] (400,680] (0,200]   (200,400]
## [301] (400,680] (200,400] (0,200]   (400,680] (400,680] (400,680]
## [307] (400,680] (400,680] (400,680] (0,200]   (400,680] (400,680]
## [313] (400,680] (400,680] (0,200]   (400,680] (400,680] (200,400]
## [319] (400,680] (200,400] (0,200]   (200,400] (400,680] (0,200]
## [325] (0,200]   (0,200]   (400,680] (0,200]   (0,200]   (0,200]
## [331] (0,200]   (400,680] (0,200]   (200,400] (400,680] (0,200]
## [337] (0,200]   (0,200]   (0,200]   (0,200]   (0,200]   (0,200]
## [343] (0,200]   (0,200]   (0,200]   (0,200]   (0,200]   (0,200]
## [349] (0,200]   (0,200]   (0,200]   (0,200]   (400,680] (0,200]
## [355] (0,200]   (400,680] (0,200]   (400,680] (0,200]   (0,200]
## [361] (400,680] (0,200]   (400,680] (400,680] (200,400] (0,200]
## [367] (400,680] (0,200]   (400,680] (0,200]   (400,680] (200,400]
## [373] (400,680] (400,680] (200,400] (0,200]   (400,680] (0,200]
## [379] (0,200]   (0,200]   (0,200]   (200,400] (0,200]   (200,400]
## [385] (400,680] (200,400] (400,680] (400,680] (0,200]   (0,200]
## [391] (0,200]   (0,200]   (400,680] (400,680] (400,680] (400,680]
## [397] (400,680] (0,200]   (400,680] (400,680] (0,200]   (0,200]
## [403] (0,200]   (400,680] (0,200]   (400,680] (400,680] (0,200]
## [409] (400,680] (0,200]   (0,200]   (0,200]   (0,200]   (400,680]
## [415] (0,200]   (200,400] (0,200]   (200,400] (200,400] (0,200]
## [421] (400,680] (0,200]   (0,200]   (400,680] (0,200]   (0,200]
## [427] (400,680] (400,680] (0,200]   (400,680] (200,400] (400,680]
## [433] (400,680] (400,680] (200,400] (0,200]   (200,400] (400,680]
## [439] (400,680] (200,400] (0,200]   (0,200]   (0,200]   (400,680]
## [445] (0,200]   (0,200]   (0,200]   (400,680] (400,680] (400,680]
## [451] (0,200]   (400,680] (0,200]   (200,400] (0,200]   (0,200]
## [457] (400,680] (0,200]   (400,680] (400,680] (0,200]   (400,680]
## [463] (400,680] (400,680] (200,400] (400,680] (0,200]   (200,400]
## [469] (400,680] (0,200]   (200,400] (0,200]   (0,200]   (0,200]
## [475] (0,200]   (0,200]   (0,200]   (0,200]   (400,680] (0,200]
## [481] (400,680] (200,400] (0,200]   (0,200]   (400,680] (0,200]
## [487] (200,400] (400,680] (400,680] (400,680] (0,200]   (200,400]
## [493] (400,680] (200,400] (200,400] (0,200]   (0,200]   (0,200]
## [499] (400,680] (200,400] (0,200]   (0,200]   (200,400] (0,200]
## [505] (400,680] (400,680] (0,200]   (0,200]   (0,200]   (400,680]
## [511] (0,200]   (0,200]   (0,200]   (400,680] (400,680] (0,200]
## [517] (0,200]   (0,200]   (0,200]   (400,680] (0,200]   (400,680]
## [523] (0,200]   (0,200]   (0,200]   (400,680] (0,200]   (200,400]
## [529] (200,400] (0,200]   (0,200]   (0,200]   (200,400] (0,200]
## [535] (0,200]   (0,200]   (400,680] (0,200]   (0,200]   (400,680]
## [541] (0,200]   (0,200]   (0,200]   (0,200]   (0,200]   (0,200]
## [547] (0,200]   (400,680] (0,200]   (400,680] (400,680] (0,200]
## [553] (0,200]   (0,200]   (0,200]   (0,200]   (0,200]   (0,200]
## [559] (400,680] (0,200]   (400,680] (0,200]   (0,200]   (200,400]
## [565] (200,400] (200,400] (0,200]   (400,680] (200,400] (0,200]
## [571] (0,200]   (0,200]   (400,680] (0,200]   (400,680] (0,200]
## [577] (400,680] (400,680] (400,680] (0,200]   (0,200]   (400,680]
```

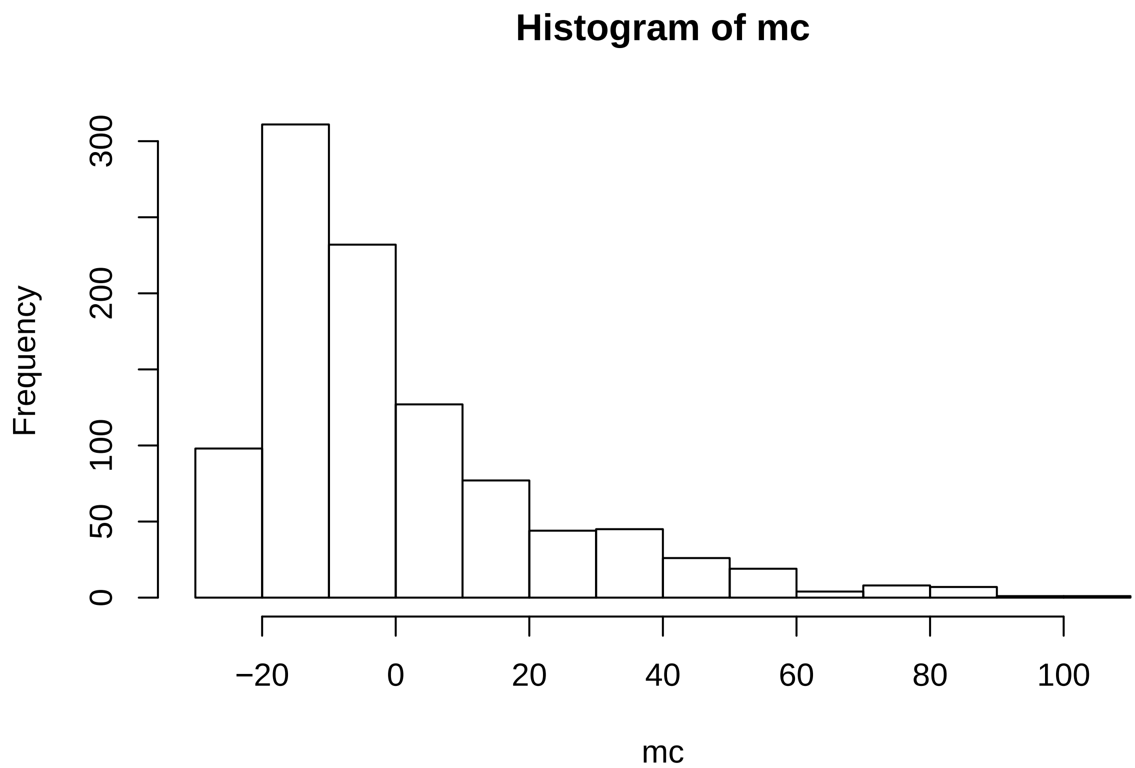
```
##  [583] (0,200]   (400,680] (400,680] (0,200]   (0,200]   (400,680]
##  [589] (400,680] (400,680] (400,680] (400,680] (200,400] (0,200]
##  [595] (0,200]   (0,200]   (0,200]   (400,680] (200,400] (0,200]
##  [601] (0,200]   (0,200]   (400,680] (400,680] (400,680] (0,200]
##  [607] (0,200]   (0,200]   (200,400] (0,200]   (0,200]   (0,200]
##  [613] (200,400] (400,680] (0,200]   (400,680] (0,200]   (0,200]
##  [619] (400,680] (0,200]   (0,200]   (200,400] (400,680] (400,680]
##  [625] (0,200]   (400,680] (0,200]   (400,680] (0,200]   (200,400]
##  [631] (400,680] (0,200]   (0,200]   (400,680] (200,400] (400,680]
##  [637] (200,400] (200,400] (0,200]   (400,680] (400,680] (0,200]
##  [643] (0,200]   (0,200]   (400,680] (0,200]   (0,200]   (0,200]
##  [649] (0,200]   (400,680] (400,680] (400,680] (0,200]   (400,680]
##  [655] (400,680] (0,200]   (400,680] (0,200]   (400,680] (400,680]
##  [661] (200,400] (200,400] (400,680] (400,680] (400,680] (400,680]
##  [667] (400,680] (200,400] (400,680] (200,400] (400,680] (0,200]
##  [673] (400,680] (400,680] (0,200]   (0,200]   (0,200]   (400,680]
##  [679] (400,680] (400,680] (200,400] (200,400] (200,400] (400,680]
##  [685] (0,200]   (200,400] (400,680] (200,400] (400,680] (400,680]
##  [691] (400,680] (200,400] (400,680] (400,680] (400,680] (400,680]
##  [697] (400,680] (200,400] (200,400] (400,680] (200,400] (400,680]
##  [703] (400,680] (400,680] (200,400] (0,200]   (400,680] (0,200]
##  [709] (200,400] (400,680] (0,200]   (0,200]   (0,200]   (0,200]
##  [715] (400,680] (400,680] (400,680] (0,200]   (200,400] (200,400]
##  [721] (0,200]   (400,680] (400,680] (400,680] (200,400] (200,400]
##  [727] (200,400] (200,400] (400,680] (400,680] (400,680] (200,400]
##  [733] (200,400] (400,680] (0,200]   (0,200]   (0,200]   (400,680]
##  [739] (400,680] (200,400] (400,680] (0,200]   (0,200]   (0,200]
##  [745] (200,400] (0,200]   (200,400] (0,200]   (400,680] (400,680]
##  [751] (400,680] (0,200]   (400,680] (0,200]   (400,680] (200,400]
##  [757] (0,200]   (0,200]   (0,200]   (0,200]   (0,200]   (400,680]
##  [763] (200,400] (0,200]   (0,200]   (0,200]   (200,400] (0,200]
##  [769] (0,200]   (0,200]   (0,200]   (200,400] (0,200]   (400,680]
##  [775] (400,680] (400,680] (200,400] (400,680] (0,200]   (400,680]
##  [781] (0,200]   (0,200]   (0,200]   (0,200]   (0,200]   (0,200]
##  [787] (0,200]   (400,680] (0,200]   (400,680] (400,680] (400,680]
##  [793] (400,680] (0,200]   (400,680] (0,200]   (200,400] (400,680]
##  [799] (0,200]   (400,680] (0,200]   (400,680] (0,200]   (400,680]
##  [805] (400,680] (200,400] (400,680] (400,680] (400,680] (400,680]
##  [811] (400,680] (0,200]   (0,200]   (0,200]   (400,680] (400,680]
##  [817] (200,400] (0,200]   (400,680] (400,680] (400,680] (0,200]
##  [823] (400,680] (200,400] (0,200]   (200,400] (200,400] (0,200]
##  [829] (0,200]   (200,400] (200,400] (0,200]   (200,400] (200,400]
##  [835] (200,400] (0,200]   (200,400] (400,680] (0,200]   (200,400]
##  [841] (400,680] (0,200]   (0,200]   (0,200]   (400,680] (400,680]
##  [847] (400,680] (400,680] (400,680] (400,680] (0,200]   (0,200]
##  [853] (0,200]   (0,200]   (400,680] (200,400] (400,680] (200,400]
##  [859] (200,400] (200,400] (400,680] (400,680] (400,680] (400,680]
##  [865] (0,200]   (200,400] (200,400] (0,200]   (0,200]   (200,400]
##  [871] (400,680] (0,200]   (0,200]   (400,680] (400,680] (400,680]
##  [877] (400,680] (400,680] (400,680] (400,680] (200,400] (400,680]
##  [883] (0,200]   (0,200]   (0,200]   (400,680] (200,400] (0,200]
##  [889] (0,200]   (0,200]   (200,400] (400,680] (0,200]   (0,200]
##  [895] (400,680] (400,680] (0,200]   (400,680] (200,400] (400,680]
##  [901] (200,400] (0,200]   (0,200]   (400,680] (400,680] (400,680]
```

```
##  [907] (400,680] (0,200]   (200,400] (0,200]   (400,680] (0,200]
##  [913] (0,200]   (400,680] (0,200]   (0,200]   (200,400] (0,200]
##  [919] (0,200]   (400,680] (0,200]   (0,200]   (400,680] (400,680]
##  [925] (0,200]   (400,680] (0,200]   (0,200]   (200,400] (0,200]
##  [931] (200,400] (400,680] (400,680] (0,200]   (0,200]   (0,200]
##  [937] (400,680] (0,200]   (400,680] (400,680] (400,680] (0,200]
##  [943] (400,680] (0,200]   (0,200]   (0,200]   (400,680] (0,200]
##  [949] (0,200]   (0,200]   (400,680] (0,200]   (0,200]   (400,680]
##  [955] (400,680] (400,680] (0,200]   (0,200]   (400,680] (200,400]
##  [961] (0,200]   (0,200]   (0,200]   (200,400] (0,200]   (400,680]
##  [967] (400,680] (400,680] (0,200]   (0,200]   (0,200]   (0,200]
##  [973] (0,200]   (200,400] (0,200]   (0,200]   (200,400] (0,200]
##  [979] (400,680] (0,200]   (400,680] (0,200]   (0,200]   (0,200]
##  [985] (0,200]   (200,400] (0,200]   (0,200]   (400,680] (0,200]
##  [991] (400,680] (400,680] (200,400] (400,680] (0,200]   (400,680]
##  [997] (200,400] (200,400] (0,200]   (0,200]
## Levels: (0,200] (200,400] (400,680]
```

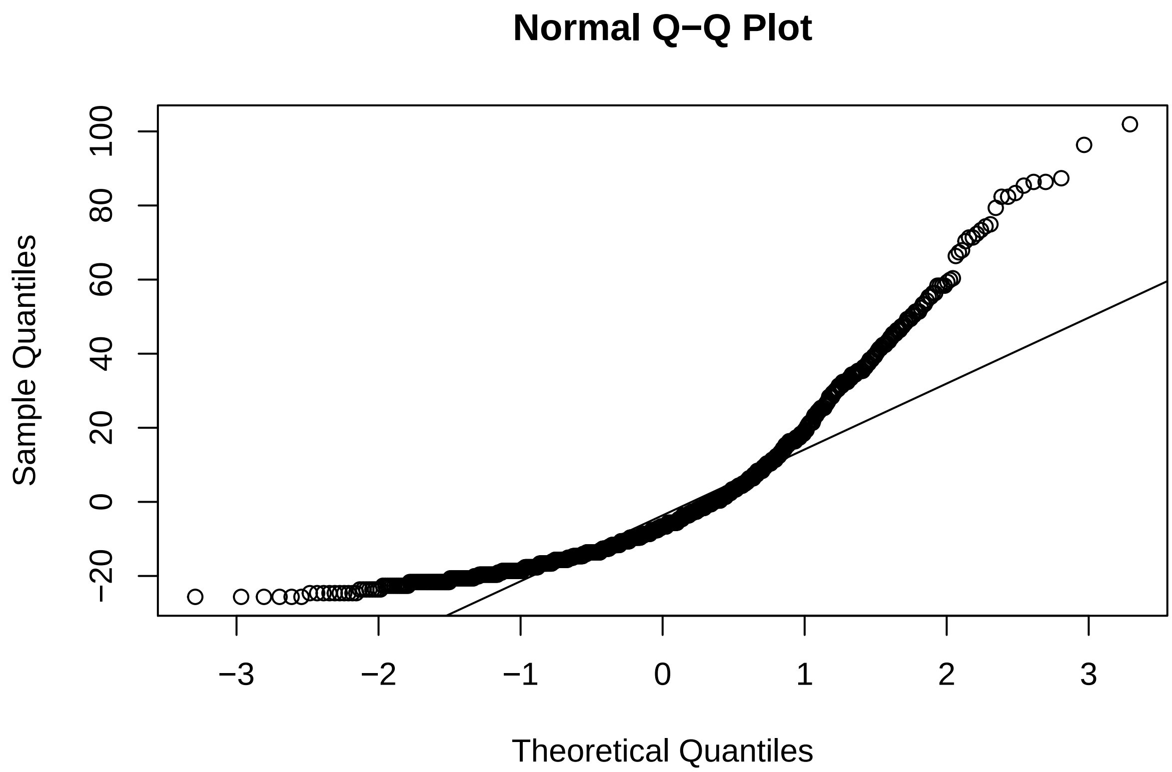
```r
table(depth_fac_events)
```

```
## depth_fac_events
##   (0,200] (200,400] (400,680]
##       418       185       397
```

```r
# Question b
mean_val <- tapply(quakes$stations,depth_fac_events,FUN=mean)
mc <- quakes$stations-mean_val[as.numeric(depth_fac_events)]
hist(mc)
```




```r
qqnorm(mc)
qqline(mc) # Data appear non-normal... Kruskal-Wallis preferred over parametric one-way ANOVA
```



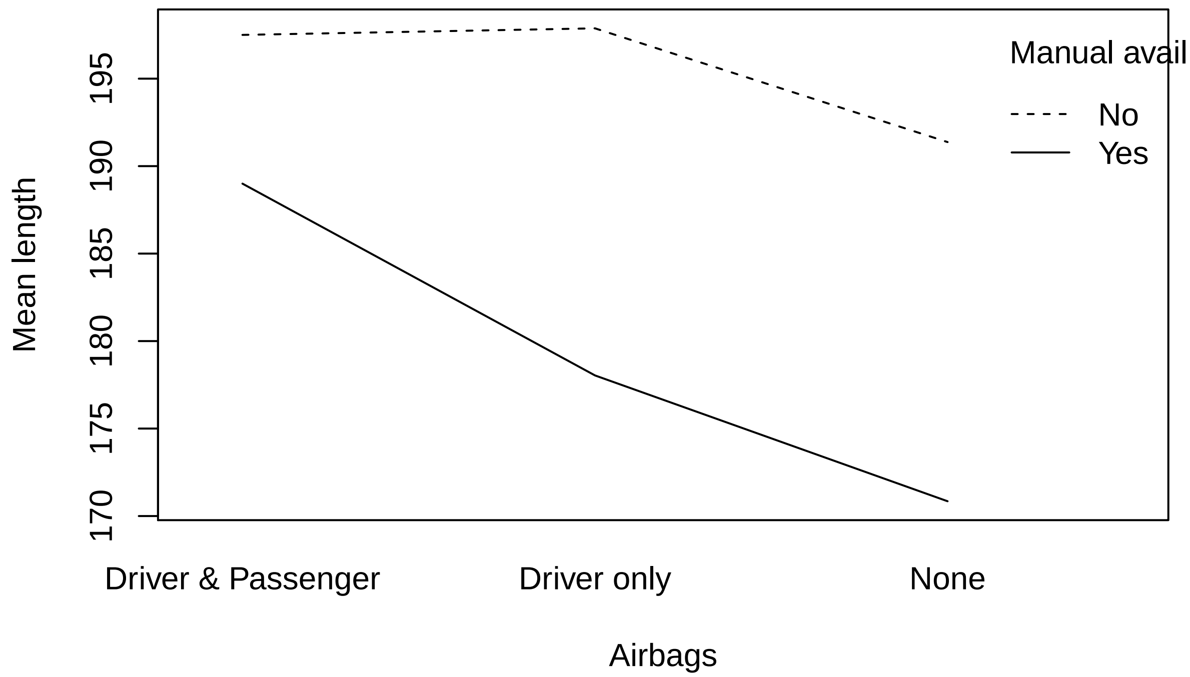

```r
# Question c
kruskal.test(quakes$stations~depth_fac_events) # P-value > 0.01 (just barely); retain null. Minimal evi
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  quakes$stations by depth_fac_events
## Kruskal-Wallis chi-squared = 9.0943, df = 2, p-value = 0.0106
```

```r
#Question d
library("MASS")
cars_mean_Lens <- aggregate(Cars93$Length,by=list(Cars93$AirBags,Cars93$Man.trans.avail),FUN=mean)
cars_mean_Lens
```

```
##              Group.1 Group.2        x
## 1 Driver & Passenger      No 197.5000
## 2        Driver only      No 197.8750
## 3               None      No 191.3750
## 4 Driver & Passenger     Yes 189.0000
## 5        Driver only     Yes 178.0370
## 6               None     Yes 170.8462
```

```r
# Question e
Result_of_Inter<- interaction.plot(x.factor=cars_mean_Lens[,1],trace.factor=cars_mean_Lens[,2],respons=
```




```
Result_of_Inter
```

```
## NULL
```

```
# There is some visual indication of interactive behavior owing to the non-parallel nature of the two l
#Question f
summary(aov(Length~AirBags+Man.trans.avail+AirBags:Man.trans.avail,data=Cars93)) # No formal statistica
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)    
## AirBags                  2   3752    1876  18.048 2.78e-07 ***
## Man.trans.avail          1   6388    6388  61.447 1.05e-11 ***
## AirBags:Man.trans.avail  2    433     217   2.084    0.131    
## Residuals               87   9044     104                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#Chapter 20 Exercise 20.1

```
## 20.1
library("MASS")
survfit <- lm(Height~Wr.Hnd,data=survey)
survfit
```

```
##
```

```
## Call:
## lm(formula = Height ~ Wr.Hnd, data = survey)
##
## Coefficients:
## (Intercept)       Wr.Hnd
##     113.954        3.117
```

```r
# Question a
Pred_CI<- predict(survfit,newdata=data.frame(Wr.Hnd=c(12,15.2,17,19.9)),interval="confidence",level=0.9
Pred_CI
```

```
##        fit      lwr      upr
## 1 151.3530 146.0901 156.6159
## 2 161.3262 158.3051 164.3473
## 3 166.9361 164.9985 168.8737
## 4 175.9743 174.3066 177.6420
```

```r
# Question b
incomplete.obs <- which(is.na(survey$Height)|is.na(survey$Wr.Hnd))
rho.xy <- cor(survey$Wr.Hnd,survey$Height,use="complete.obs")
b1 <- sd(survey$Height[-incomplete.obs])/sd(survey$Wr.Hnd[-incomplete.obs])*rho.xy
b1
```

```
## [1] 3.116617
```

```r
b0 <- mean(survey$Height[-incomplete.obs])-b1*mean(survey$Wr.Hnd[-incomplete.obs])
b0
```

```
## [1] 113.9536
```

```r
# Question c
# i
Fitted_Reg_model<- plot(survey$Height~survey$Pulse,xlab="Pulse rate (bpm)",ylab="Height (cm)", main="Mea
Fitted_Reg_model
```

```
## NULL
```

```r
survfit <- lm(Height~Pulse,data=survey)
survfit # Model equation is  y = 177.86 - 0.072x
```

```
##
## Call:
## lm(formula = Height ~ Pulse, data = survey)
##
## Coefficients:
## (Intercept)        Pulse
##   177.85708     -0.07225
```

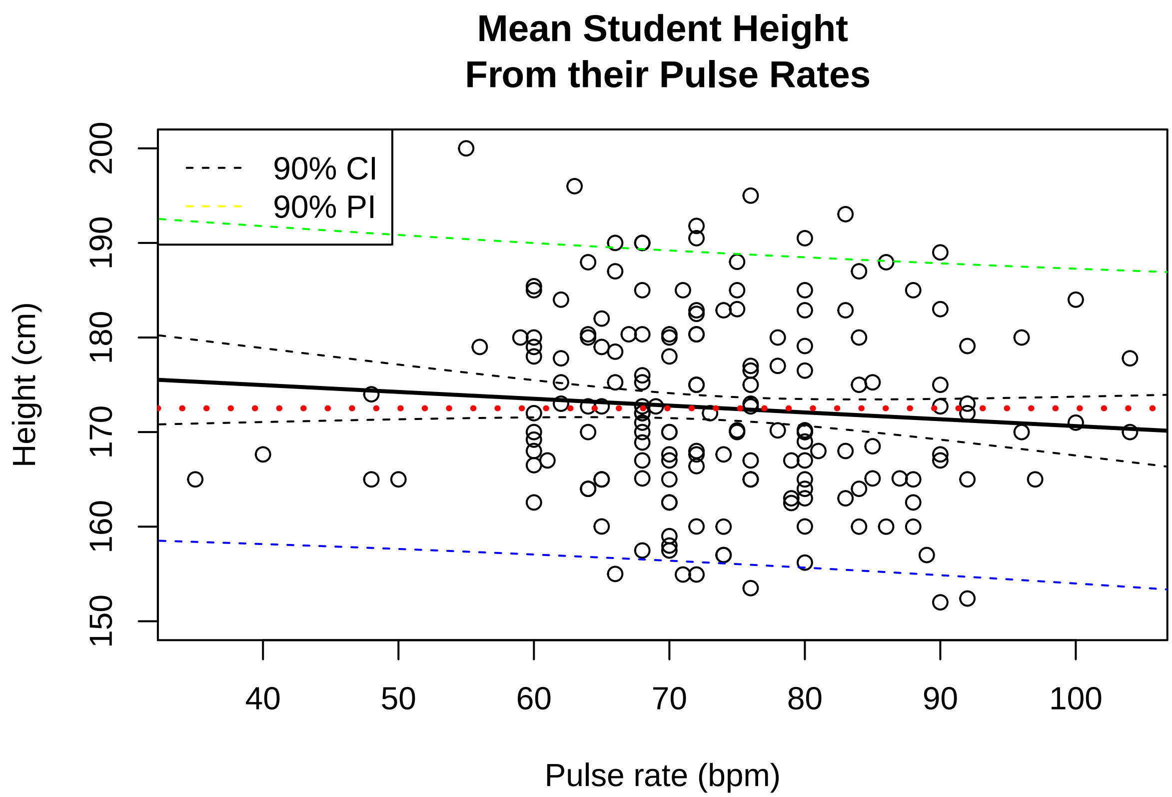
```r
abline(survfit,lwd=2)
# ii
summary(survfit) # For each additional bpm, the mean student height is estimated to decrease by 0.072cm
```

```
##
## Call:
## lm(formula = Height ~ Pulse, data = survey)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.3543  -7.2019  -0.9439   7.2622  26.1168
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 177.85708    4.93485  36.041   <2e-16 ***
## Pulse        -0.07225    0.06598  -1.095    0.275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.884 on 169 degrees of freedom
##   (66 observations deleted due to missingness)
## Multiple R-squared:  0.007046,   Adjusted R-squared:  0.001171
## F-statistic: 1.199 on 1 and 169 DF,  p-value: 0.275
```

```
CI_90<- confint(survfit,level=0.9)
CI_90
```

```
##                    5 %         95 %
## (Intercept) 169.6952304 186.01892281
## Pulse        -0.1813757   0.03687061
```

```
# iii
xseq <- data.frame(Pulse=seq(30,110,length=100))
survfit.ci <- predict(survfit,newdata=xseq,interval="confidence",level=0.9)
survfit.pi <- predict(survfit,newdata=xseq,interval="prediction",level=0.9)
lines(xseq[,1],survfit.ci[,2],lty=2)
lines(xseq[,1],survfit.ci[,3],lty=2)
lines(xseq[,1],survfit.pi[,2],lty=2,col="blue")
lines(xseq[,1],survfit.pi[,3],lty=2,col="green")
legend("topleft",legend=c("90% CI","90% PI"),lty=2,col=c("black","yellow"))
# iv
incomplete.obs <- which(is.na(survey$Height)|is.na(survey$Pulse))
abline(h=mean(survey$Height[-incomplete.obs]),col=2,lty=3,lwd=3) # The line sits in the middle of the C
```




```
# Question d
data("mtcars")
?mtcars
```

```
## starting httpd help server ... done
```

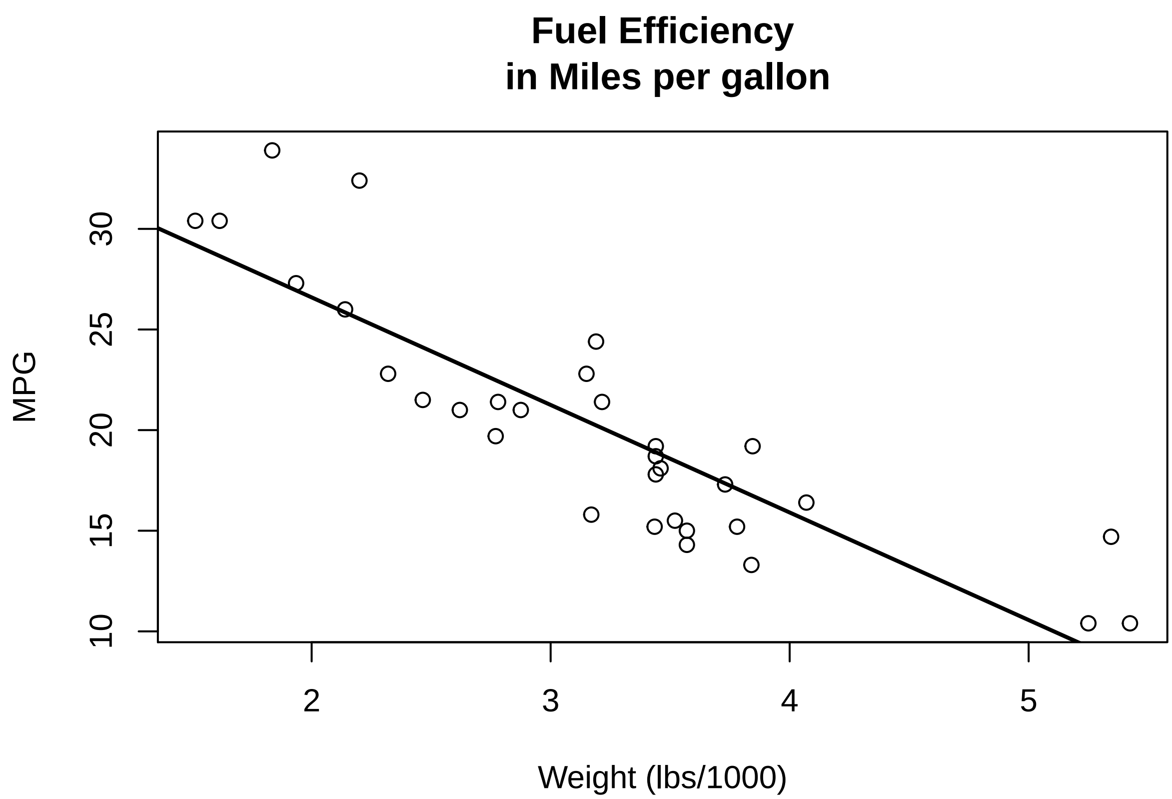
```
fuel_effic<- plot(mtcars$mpg~mtcars$wt,xlab="Weight (lbs/1000)",ylab="MPG", main="Fuel Efficiency\n in M
fuel_effic
```

```
## NULL
```

```
# Question e
carfit <- lm(mpg~wt,data=mtcars)
carfit
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Coefficients:
## (Intercept)           wt
##      37.285       -5.344
```

```r
abline(carfit,lwd=2)
```




```r
# Question f
summary(carfit) # mean MPG = 37.28 - 5.34*weight # For each extra 1000lbs of weight, the mean MPG decre
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

```r
# Question g
PI_car<- predict(carfit,newdata=data.frame(wt=6),interval="prediction",level=0.95)
PI_car# Predicting at 6000lbs seems untrustworthy. Extrapolation is far enough outside the range of the
```

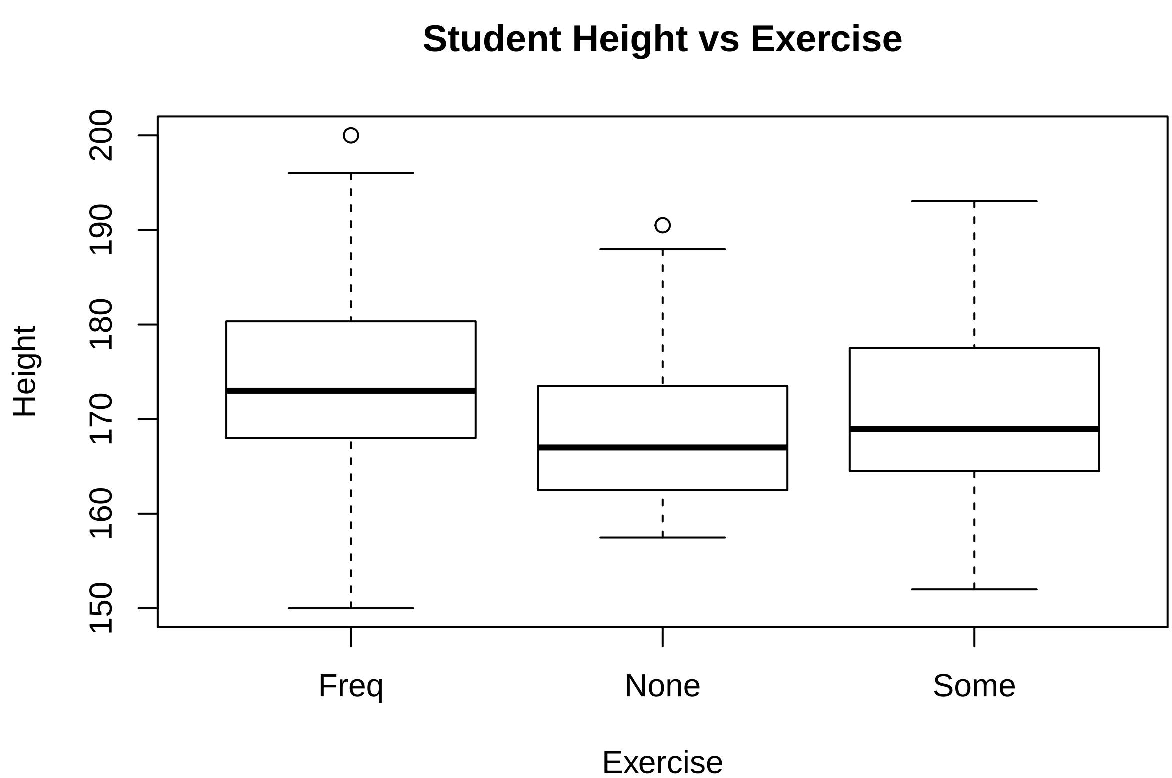
```
##        fit      lwr      upr
## 1 5.218297 -1.85279 12.28938
```

Exercise 20.2

```r
## 20.2
library("MASS")
# Question a
Count_of_stud<- table(survey$Exer)
Count_of_stud
```

```
##
## Freq None Some
##  115   24   98
```

```r
Exer_plt<- boxplot(survey$Height~survey$Exer, xlab = "Exercise", ylab = "Height", main="Student Height v
```



```r
Exer_plt
```

```
## $stats
##        [,1]   [,2]   [,3]
## [1,] 150.00 157.48 152.00
## [2,] 168.00 162.50 164.50
## [3,] 173.00 167.00 168.95
## [4,] 180.34 173.50 177.50
## [5,] 196.00 187.96 193.04
##
## $n
## [1] 105  20  84
##
## $conf
##          [,1]     [,2]     [,3]
## [1,] 171.0973 163.1137 166.7089
## [2,] 174.9027 170.8863 171.1911
##
## $out
## [1] 200.0 190.5
##
## $group
## [1] 1 2
##
## $names
## [1] "Freq" "None" "Some"
```

```r
# Question b
survfit <- lm(Height~Exer,data=survey)
survfit
```

```
##
## Call:
## lm(formula = Height ~ Exer, data = survey)
##
## Coefficients:
## (Intercept)     ExerNone     ExerSome
##     174.607       -5.579       -4.210
```

```r
summary(survfit) # The reference level of the predictor defaults to the first level of the factor, whic
```

```
##
## Call:
## lm(formula = Height ~ Exer, data = survey)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.607  -6.397  -1.607   6.103  25.393
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 174.6067     0.9396 185.836  < 2e-16 ***
```

```
## ExerNone     -5.5787     2.3489  -2.375  0.01847 *
## ExerSome     -4.2098     1.4094  -2.987  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.628 on 206 degrees of freedom
##   (28 observations deleted due to missingness)
## Multiple R-squared:  0.05333,    Adjusted R-squared:  0.04414
## F-statistic: 5.802 on 2 and 206 DF,  p-value: 0.003536
```

```r
# Question c
# It appears that both of the levels for which coefficient estimates were obtained, yielded p-values th
# The coefficient corresponding to 'some' has the smallest p-value of the two additive dummy levels.
# The negative point estimates of both estimates tell us that the model predicts the effect on height o
# The shortest mean height is reserved for those in the 'none' exercise category; the estimated coeffic
# Overall statistical significance of the predictor (in terms of the effect of exercise on height) is s
# Question d
Mheight_of_each <- factor(levels(survey$Exer))
Mheight_of_each
```

```
## [1] Freq None Some
## Levels: Freq None Some
```

```r
Res_of_Pred<- predict(survfit,newdata=data.frame(Exer=Mheight_of_each),interval="prediction")
Res_of_Pred
```

```
##        fit      lwr      upr
## 1 174.6067 155.5349 193.6784
## 2 169.0280 149.5777 188.4783
## 3 170.3969 151.3027 189.4911
```

```r
# Question e
summary(aov(Height~Exer,data=survey)) # Same 'global' P-value as the lm model summary. There is evidenc
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Exer          2   1076   537.8   5.802 0.00354 **
## Residuals   206  19095    92.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 28 observations deleted due to missingness
```

```r
#Question f
ExerReordered <- relevel(survey$Exer,ref="None")
levels(ExerReordered)
```

```
## [1] "None" "Freq" "Some"
```

```r
summary(aov(Height~ExerReordered,data=survey)) # There is no change to the omnibus F-test if we reorder
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
```

```
## ExerReordered   2   1076   537.8   5.802 0.00354 **
## Residuals     206  19095    92.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 28 observations deleted due to missingness
```

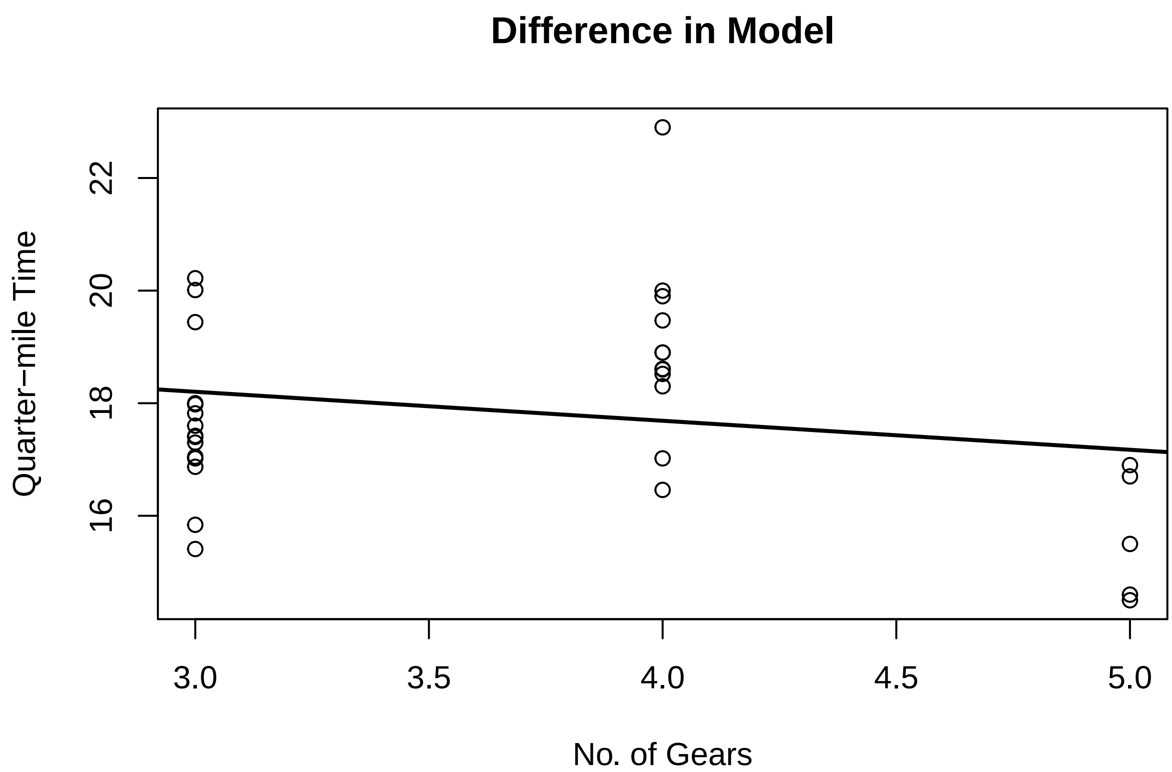
```
#Question g
carfit <- lm(qsec~gear,data=mtcars)
summary(carfit) # The effect of 'gear' when treated as a continuous variable is interpreted as a decrea
```

```
##
## Call:
## lm(formula = qsec ~ gear, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7929 -1.1604 -0.3278  1.2122  5.2122
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.7482     1.6239  12.161    4e-13 ***
## gear         -0.5151     0.4321  -1.192    0.243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.775 on 30 degrees of freedom
## Multiple R-squared:  0.04523,    Adjusted R-squared:  0.01341
## F-statistic: 1.421 on 1 and 30 DF,  p-value: 0.2425
```

```
#Question h
carfit2 <- lm(qsec~factor(gear),data=mtcars)
summary(carfit2) # The effect of 'gear' when treated as a categorical variable now appears to be statis
```

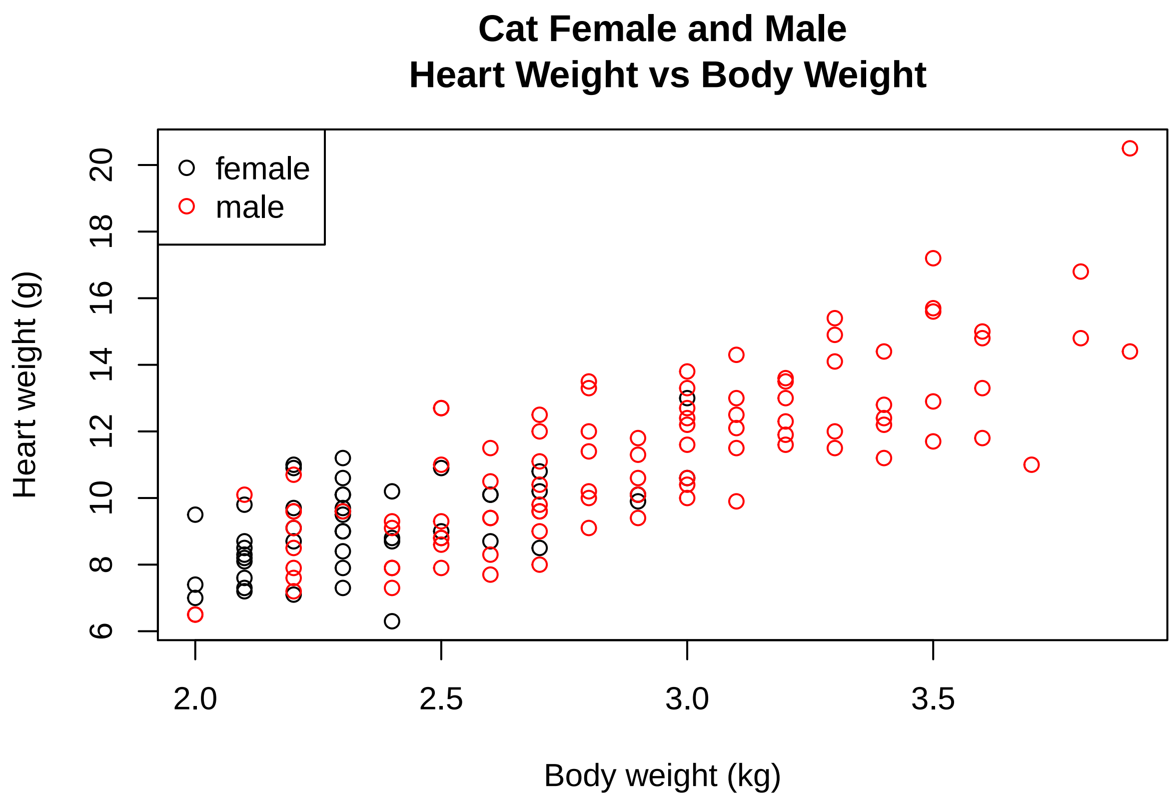
```
##
## Call:
## lm(formula = qsec ~ factor(gear), data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5050 -0.6667 -0.2060  0.6125  3.9350
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.6920     0.3691  47.928  < 2e-16 ***
## factor(gear)4   1.2730     0.5537   2.299  0.02890 *
## factor(gear)5  -2.0520     0.7383  -2.779  0.00946 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.43 on 29 degrees of freedom
## Multiple R-squared:  0.4012, Adjusted R-squared:  0.3599
## F-statistic: 9.715 on 2 and 29 DF,  p-value: 0.0005897
```

```
# Question i
plot(mtcars$qsec~mtcars$gear,xlab="No. of Gears",ylab="Quarter-mile Time",main="Difference in Model")
abline(carfit,lwd=2) # The plot indicates clearly that the difference between the two models is due to
```



#Chapter 21 Exercise 21.1

```
## 21.1
library("MASS")
?cats
# Question a
Catsweight<- plot(cats$Bwt,cats$Hwt,col=cats$Sex,xlab="Body weight (kg)",ylab="Heart weight (g)", main=
legend("topleft",legend=c("female","male"),col=c(1,2),pch=c(1,1))
```




```
Catsweight
```

```
## NULL
```

```
# Females are black, since the levels of the factor vector cats$Sex are in the alphabetical order of "F
# Question b
catsfit <- lm(Hwt~Bwt+Sex,data=cats)
catsfit
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt + Sex, data = cats)
##
## Coefficients:
## (Intercept)          Bwt         SexM
##     -0.4150       4.0758      -0.0821
```

```
summary(catsfit)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt + Sex, data = cats)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -3.5833 -0.9700 -0.0948  1.0432  5.1016
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4149     0.7273  -0.571    0.569
## Bwt           4.0758     0.2948  13.826   <2e-16 ***
## SexM         -0.0821     0.3040  -0.270    0.788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.457 on 141 degrees of freedom
## Multiple R-squared:  0.6468, Adjusted R-squared:  0.6418
## F-statistic: 129.1 on 2 and 141 DF,  p-value: < 2.2e-16
```

```
# i
# "Mean heart weight" = -0.415 + 4.076*"Body weight" - 0.082*"is male"
# For cats of the same sex, the effect of an additional kg of body weight is, on average, an extra 4.07
# The model states the effect of body weight is highly statistically significant -- there is evidence t
# The above notes imply that the inclusion of "sex" as a predictor is statistically unnecessary when it
# ii
names(summary(catsfit))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
summary(catsfit)$r.squared
```

```
## [1] 0.6468035
```

```
# The coefficient of determination, 'R-squared', shows that for your fitted model, 64.5% of the variati
summary(catsfit)$fstatistic
```

```
##    value    numdf    dendf
## 129.1056   2.0000 141.0000
```

```
1-pf(129.1056,df1=2,df2=141)
```

```
## [1] 0
```

```
# Reading from the summary output, or running the line above, the result of the omnibus F-test is a tin
# Question c
ModelPred<- predict(catsfit,TilmanCat=data.frame(Bwt=3.4,Sex="F"),interval="prediction",level=0.95)
```

```
## Warning in predict.lm(catsfit, TilmanCat = data.frame(Bwt = 3.4, Sex = "F"), : predictions on curren
```

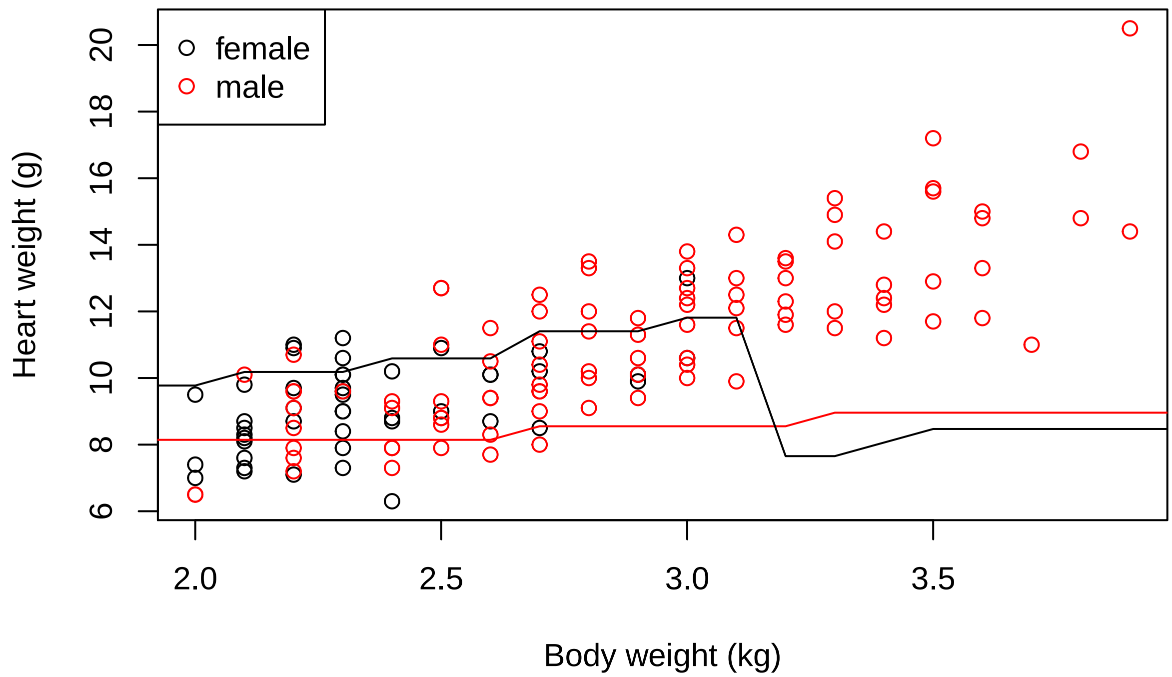
```
ModelPred
```

```
##          fit      lwr      upr
## 1   7.736585 4.817907 10.65526
## 2   7.736585 4.817907 10.65526
## 3   7.736585 4.817907 10.65526
## 4   8.144162 5.229089 11.05924
## 5   8.144162 5.229089 11.05924
## 6   8.144162 5.229089 11.05924
## 7   8.144162 5.229089 11.05924
## 8   8.144162 5.229089 11.05924
## 9   8.144162 5.229089 11.05924
## 10  8.144162 5.229089 11.05924
## 11  8.144162 5.229089 11.05924
## 12  8.144162 5.229089 11.05924
## 13  8.551739 5.639108 11.46437
## 14  8.551739 5.639108 11.46437
## 15  8.551739 5.639108 11.46437
## 16  8.551739 5.639108 11.46437
## 17  8.551739 5.639108 11.46437
## 18  8.551739 5.639108 11.46437
## 19  8.959316 6.047963 11.87067
## 20  8.959316 6.047963 11.87067
## 21  8.959316 6.047963 11.87067
## 22  8.959316 6.047963 11.87067
## 23  8.959316 6.047963 11.87067
## 24  8.959316 6.047963 11.87067
## 25  8.959316 6.047963 11.87067
## 26  8.959316 6.047963 11.87067
## 27  8.959316 6.047963 11.87067
## 28  8.959316 6.047963 11.87067
## 29  8.959316 6.047963 11.87067
## 30  8.959316 6.047963 11.87067
## 31  9.366893 6.455652 12.27813
## 32  9.366893 6.455652 12.27813
## 33  9.366893 6.455652 12.27813
## 34  9.366893 6.455652 12.27813
## 35  9.774470 6.862174 12.68677
## 36  9.774470 6.862174 12.68677
## 37 10.182047 7.267531 13.09656
## 38 10.182047 7.267531 13.09656
## 39 10.182047 7.267531 13.09656
## 40 10.589623 7.671725 13.50752
## 41 10.589623 7.671725 13.50752
## 42 10.589623 7.671725 13.50752
## 43 11.404777 8.476645 14.33291
## 44 11.404777 8.476645 14.33291
## 45 11.404777 8.476645 14.33291
## 46 11.812354 8.877381 14.74733
## 47 11.812354 8.877381 14.74733
## 48  7.654488 4.711894 10.59708
## 49  7.654488 4.711894 10.59708
## 50  8.062065 5.129298 10.99483
## 51  8.469642 5.545573 11.39371
## 52  8.469642 5.545573 11.39371
## 53  8.469642 5.545573 11.39371
```

```
## 54   8.469642  5.545573 11.39371
## 55   8.469642  5.545573 11.39371
## 56   8.469642  5.545573 11.39371
## 57   8.469642  5.545573 11.39371
## 58   8.469642  5.545573 11.39371
## 59   8.877219  5.960710 11.79373
## 60   9.284796  6.374698 12.19489
## 61   9.284796  6.374698 12.19489
## 62   9.284796  6.374698 12.19489
## 63   9.284796  6.374698 12.19489
## 64   9.284796  6.374698 12.19489
## 65   9.692373  6.787532 12.59721
## 66   9.692373  6.787532 12.59721
## 67   9.692373  6.787532 12.59721
## 68   9.692373  6.787532 12.59721
## 69   9.692373  6.787532 12.59721
## 70   9.692373  6.787532 12.59721
## 71   9.692373  6.787532 12.59721
## 72   9.692373  6.787532 12.59721
## 73  10.099950  7.199204 13.00070
## 74  10.099950  7.199204 13.00070
## 75  10.099950  7.199204 13.00070
## 76  10.099950  7.199204 13.00070
## 77  10.099950  7.199204 13.00070
## 78  10.099950  7.199204 13.00070
## 79  10.507527  7.609709 13.40534
## 80  10.507527  7.609709 13.40534
## 81  10.507527  7.609709 13.40534
## 82  10.507527  7.609709 13.40534
## 83  10.507527  7.609709 13.40534
## 84  10.507527  7.609709 13.40534
## 85  10.507527  7.609709 13.40534
## 86  10.507527  7.609709 13.40534
## 87  10.507527  7.609709 13.40534
## 88  10.915104  8.019045 13.81116
## 89  10.915104  8.019045 13.81116
## 90  10.915104  8.019045 13.81116
## 91  10.915104  8.019045 13.81116
## 92  10.915104  8.019045 13.81116
## 93  10.915104  8.019045 13.81116
## 94  10.915104  8.019045 13.81116
## 95  11.322680  8.427208 14.21815
## 96  11.322680  8.427208 14.21815
## 97  11.322680  8.427208 14.21815
## 98  11.322680  8.427208 14.21815
## 99  11.322680  8.427208 14.21815
## 100 11.730257  8.834199 14.62632
## 101 11.730257  8.834199 14.62632
## 102 11.730257  8.834199 14.62632
## 103 11.730257  8.834199 14.62632
## 104 11.730257  8.834199 14.62632
## 105 11.730257  8.834199 14.62632
## 106 11.730257  8.834199 14.62632
## 107 11.730257  8.834199 14.62632
```

```
## 108 11.730257  8.834199 14.62632
## 109 12.137834  9.240017 15.03565
## 110 12.137834  9.240017 15.03565
## 111 12.137834  9.240017 15.03565
## 112 12.137834  9.240017 15.03565
## 113 12.137834  9.240017 15.03565
## 114 12.137834  9.240017 15.03565
## 115 12.545411  9.644665 15.44616
## 116 12.545411  9.644665 15.44616
## 117 12.545411  9.644665 15.44616
## 118 12.545411  9.644665 15.44616
## 119 12.545411  9.644665 15.44616
## 120 12.545411  9.644665 15.44616
## 121 12.952988 10.048147 15.85783
## 122 12.952988 10.048147 15.85783
## 123 12.952988 10.048147 15.85783
## 124 12.952988 10.048147 15.85783
## 125 12.952988 10.048147 15.85783
## 126 13.360565 10.450467 16.27066
## 127 13.360565 10.450467 16.27066
## 128 13.360565 10.450467 16.27066
## 129 13.360565 10.450467 16.27066
## 130 13.360565 10.450467 16.27066
## 131 13.768142 10.851632 16.68465
## 132 13.768142 10.851632 16.68465
## 133 13.768142 10.851632 16.68465
## 134 13.768142 10.851632 16.68465
## 135 13.768142 10.851632 16.68465
## 136 14.175719 11.251650 17.09979
## 137 14.175719 11.251650 17.09979
## 138 14.175719 11.251650 17.09979
## 139 14.175719 11.251650 17.09979
## 140 14.583296 11.650529 17.51606
## 141 14.990872 12.048278 17.93347
## 142 14.990872 12.048278 17.93347
## 143 15.398449 12.444911 18.35199
## 144 15.398449 12.444911 18.35199
```

```r
# Question d
plot(cats$Bwt,cats$Hwt,col=cats$Sex,xlab="Body weight (kg)",ylab="Heart weight (g)")
legend("topleft",legend=c("female","male"),col=c(1,2),pch=c(1,1))
Bwtseq <- seq(min(cats$Bwt)-0.5,max(cats$Bwt)+0.5,length=30)
n <- length(Bwtseq)
catspred <- predict(catsfit,TilmanCat=data.frame(Bwt=rep(Bwtseq,2),Sex=rep(c("M","F"),each=n)))
lines(Bwtseq,catspred[1:n],col=2)
lines(Bwtseq,catspred[(n+1):(2*n)])
```

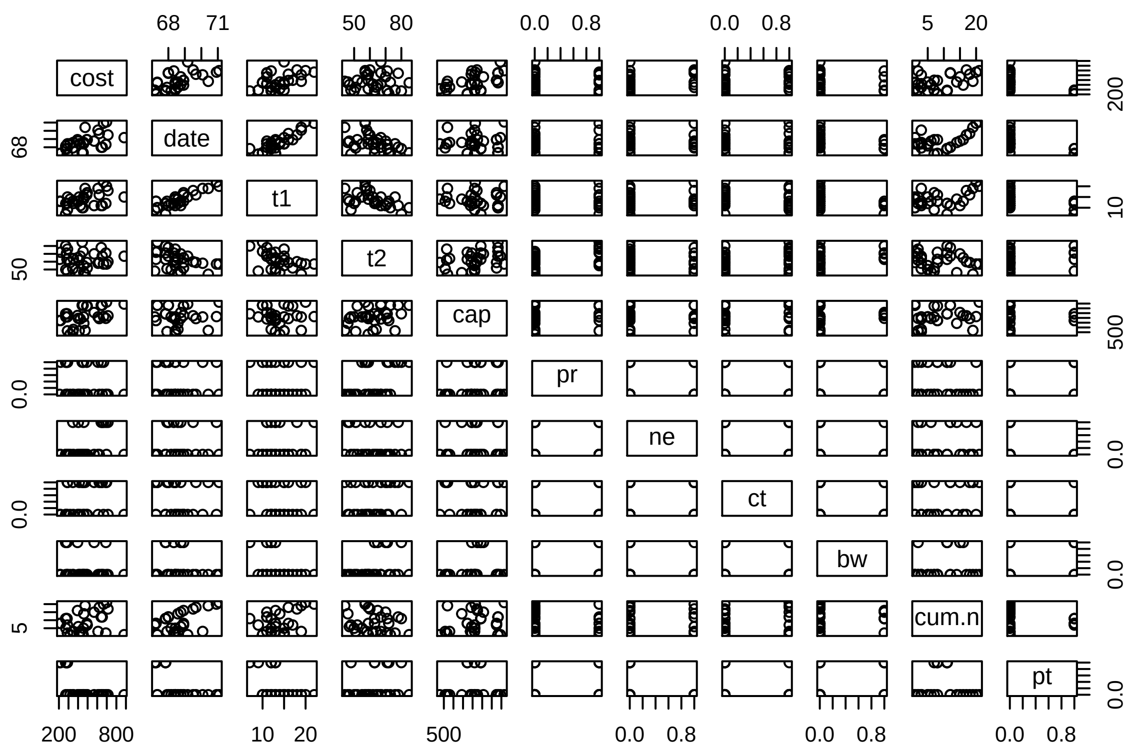



```
catspred
```

```
##         1         2         3         4         5         6         7 
##  7.736585  7.736585  7.736585  8.144162  8.144162  8.144162  8.144162 
##         8         9        10        11        12        13        14 
##  8.144162  8.144162  8.144162  8.144162  8.144162  8.551739  8.551739 
##        15        16        17        18        19        20        21 
##  8.551739  8.551739  8.551739  8.551739  8.959316  8.959316  8.959316 
##        22        23        24        25        26        27        28 
##  8.959316  8.959316  8.959316  8.959316  8.959316  8.959316  8.959316 
##        29        30        31        32        33        34        35 
##  8.959316  8.959316  9.366893  9.366893  9.366893  9.366893  9.774470 
##        36        37        38        39        40        41        42 
##  9.774470 10.182047 10.182047 10.182047 10.589623 10.589623 10.589623 
##        43        44        45        46        47        48        49 
## 11.404777 11.404777 11.404777 11.812354 11.812354  7.654488  7.654488 
##        50        51        52        53        54        55        56 
##  8.062065  8.469642  8.469642  8.469642  8.469642  8.469642  8.469642 
##        57        58        59        60        61        62        63 
##  8.469642  8.469642  8.877219  9.284796  9.284796  9.284796  9.284796 
##        64        65        66        67        68        69        70 
##  9.284796  9.692373  9.692373  9.692373  9.692373  9.692373  9.692373 
##        71        72        73        74        75        76        77 
##  9.692373  9.692373 10.099950 10.099950 10.099950 10.099950 10.099950 
##        78        79        80        81        82        83        84 
## 10.099950 10.507527 10.507527 10.507527 10.507527 10.507527 10.507527 
```

```
##         85        86        87        88        89        90        91
## 10.507527 10.507527 10.507527 10.915104 10.915104 10.915104 10.915104
##         92        93        94        95        96        97        98
## 10.915104 10.915104 10.915104 11.322680 11.322680 11.322680 11.322680
##         99       100       101       102       103       104       105
## 11.322680 11.730257 11.730257 11.730257 11.730257 11.730257 11.730257
##        106       107       108       109       110       111       112
## 11.730257 11.730257 11.730257 12.137834 12.137834 12.137834 12.137834
##        113       114       115       116       117       118       119
## 12.137834 12.137834 12.545411 12.545411 12.545411 12.545411 12.545411
##        120       121       122       123       124       125       126
## 12.545411 12.952988 12.952988 12.952988 12.952988 12.952988 13.360565
##        127       128       129       130       131       132       133
## 13.360565 13.360565 13.360565 13.360565 13.768142 13.768142 13.768142
##        134       135       136       137       138       139       140
## 13.768142 13.768142 14.175719 14.175719 14.175719 14.175719 14.583296
##        141       142       143       144
## 14.990872 14.990872 15.398449 15.398449
```

```r
# The two superimposed lines are positively sloped according to the coefficient for "Bwt", but extremel
# Question e
library("boot")
?nuclear
pairs(nuclear)
```



```r
# Question f
Nuclearfit <- lm(cost~t1+t2,data=nuclear)
summary(Nuclearfit)
```

```
##
## Call:
## lm(formula = cost ~ t1 + t2, data = nuclear)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -273.17  -73.42  -13.40   69.31  360.61
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -242.146    268.020  -0.903  0.37373
## t1            29.908      9.086   3.292  0.00262 **
## t2             4.689      2.945   1.592  0.12224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 150.1 on 29 degrees of freedom
## Multiple R-squared:  0.272,  Adjusted R-squared:  0.2218
## F-statistic: 5.418 on 2 and 29 DF,  p-value: 0.01001
```

```r
# Question g
Nuclearfit2 <- lm(cost~t1+t2+date,data=nuclear)
summary(Nuclearfit2)
```

```
##
## Call:
## lm(formula = cost ~ t1 + t2 + date, data = nuclear)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -208.63  -90.74  -12.07   59.78  324.19
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9232.833   2974.432  -3.104  0.00434 **
## t1             -5.918     14.281  -0.414  0.68176
## t2              4.639      2.601   1.784  0.08535 .
## date          138.324     45.617   3.032  0.00519 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 132.5 on 28 degrees of freedom
## Multiple R-squared:  0.452,  Adjusted R-squared:  0.3933
## F-statistic: 7.698 on 3 and 28 DF,  p-value: 0.000667
```

```r
# By including "date" in the linear model, this completely removes the statistical significance of "t1"
# Question h
Nuclearfit3 <- lm(cost~date+cap+ne,data=nuclear)
summary(Nuclearfit3) # Fitted model is "cost" = -6458 + 95.4*"date of permit issue" + 0.42*"capacity" +
```

```
##
## Call:
## lm(formula = cost ~ date + cap + ne, data = nuclear)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -154.966  -68.202   -3.614   45.919  285.014
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.458e+03  1.216e+03  -5.310 1.19e-05 ***
## date         9.544e+01  1.773e+01   5.382 9.77e-06 ***
## cap          4.157e-01  9.463e-02   4.393 0.000145 ***
## ne           1.261e+02  4.092e+01   3.083 0.004575 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99.74 on 28 degrees of freedom
## Multiple R-squared:  0.6895, Adjusted R-squared:  0.6562
## F-statistic: 20.73 on 3 and 28 DF,  p-value: 2.827e-07
```

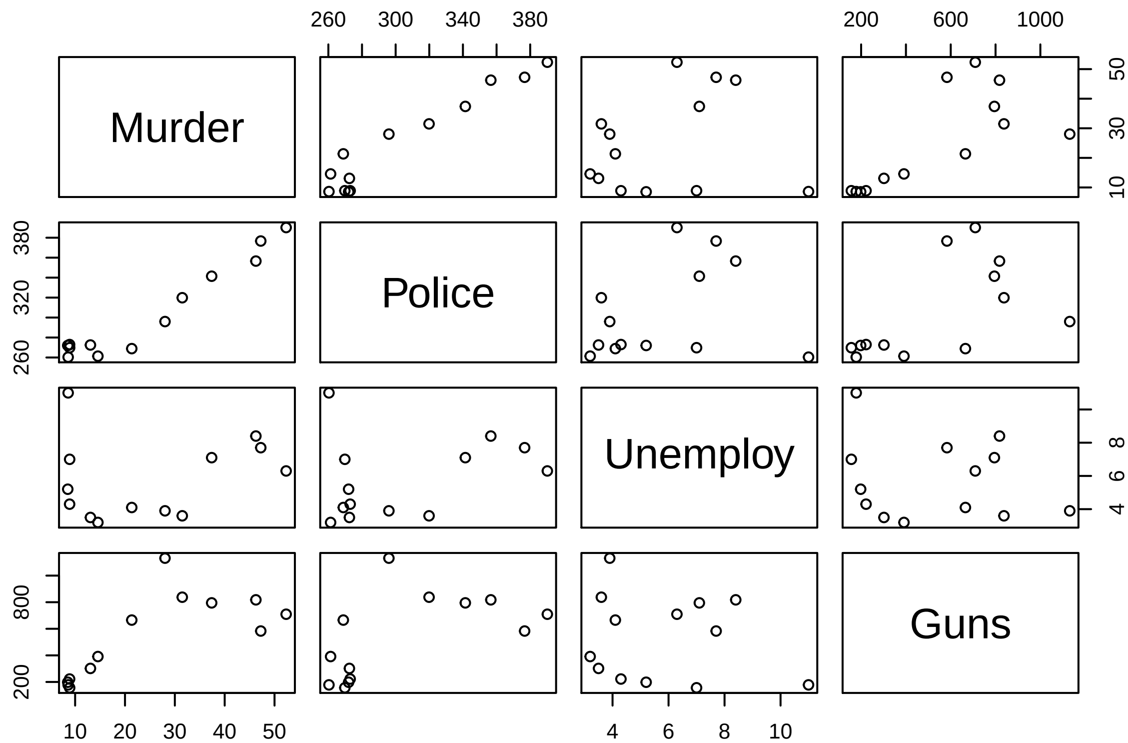
```r
confint(Nuclearfit3) # All intervals exclude null value of zero, reflecting their significance in the m
```

```
##                     2.5 %        97.5 %
## (Intercept) -8949.8032112 -3966.9745900
## date           59.1134640   131.7636535
## cap             0.2218791     0.6095524
## ne             42.3145363   209.9430014
```

```r
# i
Detroit <- data.frame(Murder=c(8.6,8.9,8.52,8.89,13.07,14.57,21.36,28.03,31.49,37.39,46.26,47.24,52.33)
Detroit
```

```
##    Murder Police Unemploy    Guns
## 1    8.60 260.35     11.0  178.15
## 2    8.90 269.80      7.0  156.41
## 3    8.52 272.04      5.2  198.02
## 4    8.89 272.96      4.3  222.10
## 5   13.07 272.51      3.5  301.92
## 6   14.57 261.34      3.2  391.22
## 7   21.36 268.89      4.1  665.56
## 8   28.03 295.99      3.9 1131.21
## 9   31.49 319.87      3.6  837.60
## 10  37.39 341.43      7.1  794.90
## 11  46.26 356.59      8.4  817.74
## 12  47.24 376.69      7.7  583.17
## 13  52.33 390.19      6.3  709.59
```

```r
pairs(Detroit)
```




```r
# The number of police seems to be the single most telling variable for prediction of murder numbers.
# Question j
Murderfit <- lm(Murder~Police+Unemploy+Guns,data=Detroit)
summary(Murderfit)
```

```
##
## Call:
## lm(formula = Murder ~ Police + Unemploy + Guns, data = Detroit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8422 -1.9451  0.2012  0.9172  4.6694
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -68.852509   5.862631 -11.744 9.25e-07 ***
## Police        0.280799   0.024657  11.388 1.20e-06 ***
## Unemploy      0.147248   0.408235   0.361  0.72665
## Guns          0.014177   0.003538   4.007  0.00308 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.89 on 9 degrees of freedom
## Multiple R-squared:  0.9767, Adjusted R-squared:  0.9689
## F-statistic: 125.6 on 3 and 9 DF,  p-value: 1.158e-07
```

```r
summary(Murderfit)$r.squared
```

```
## [1] 0.9766753
```

```r
# "Mean murders" = -68.85 + 0.281*"no. of police" + 0.147*"unemployment" + 0.014*"no. of gun licenses".
# After adjusting for "no. of gun licenses", and "unemployment", each additional police per 100000 popu
# After adjusting for "no. of police", and "unemployment", each additional gun license per 100000 popul
# After adjusting for "no. of gun licenses", and "no. of police", each additional percentage of unemplo
# No, it doesn't make sense to claim that *any* of the relationships are causal, particularly based onl
# Question k
summary(Murderfit)$r.squared
```

```
## [1] 0.9766753
```

```r
# Approx. 97.67% of the variability in mean murder numbers is explained by the three-predictor model (t
Murderfit2 <- lm(Murder~Police+Guns,data=Detroit)
summary(Murderfit2)
```

```
##
## Call:
## lm(formula = Murder ~ Police + Guns, data = Detroit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9424 -2.1068  0.2775  0.9614  4.6408
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -69.002919   5.587647 -12.349 2.23e-07 ***
## Police        0.285048   0.020697  13.772 7.92e-08 ***
## Guns          0.013636   0.003062   4.453  0.00123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.761 on 10 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9716
## F-statistic: 206.3 on 2 and 10 DF,  p-value: 7.417e-09
```

```r
summary(Murderfit2)$r.squared
```

```
## [1] 0.9763381
```

```r
## The coefficient of determination has barely changed from before; approx. 97.63% of the variability i
# Question l
MurderPred<- predict(Murderfit2,newdata=data.frame(Police=c(300,300),Guns=c(500,0)),interval="confidenc
MurderPred
```

```
##        fit      lwr      upr
## 1 23.32948 20.88251 25.77645
## 2 16.51159 10.90530 22.11787
```

#Exercise 21.2

```
## 21.2

# Question a
GalBall <- data.frame(d=c(573,534,495,451,395,337,253),h=c(1,0.8,0.6,0.45,0.3,0.2,0.1))
GalBall
```

```
##     d    h
## 1 573 1.00
## 2 534 0.80
## 3 495 0.60
## 4 451 0.45
## 5 395 0.30
## 6 337 0.20
## 7 253 0.10
```

```
plot(GalBall$d~GalBall$h,pch=19,xlab="Initial height",ylab="Distance traveled", main="Galileo's Ball ex
# Question b
# i
Galfit_order2 <- lm(d~h+I(h^2),data=GalBall)
Galfit_order2
```

```
##
## Call:
## lm(formula = d ~ h + I(h^2), data = GalBall)
##
## Coefficients:
## (Intercept)            h       I(h^2)
##       199.9        708.3       -343.7
```

```
summary(Galfit_order2)
```

```
##
## Call:
## lm(formula = d ~ h + I(h^2), data = GalBall)
##
## Residuals:
##       1       2       3       4       5       6       7
##   8.458 -12.607  -6.177   1.940  13.523   9.170 -14.308
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   199.91      16.76  11.928 0.000283 ***
## h             708.32      74.82   9.467 0.000695 ***
## I(h^2)       -343.69      66.78  -5.147 0.006760 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.64 on 4 degrees of freedom
## Multiple R-squared:  0.9903, Adjusted R-squared:  0.9855
## F-statistic:   205 on 2 and 4 DF,  p-value: 9.333e-05
```

```r
# ii
Galfit_order3 <- lm(d~h+I(h^2)+I(h^3),data=GalBall)
Galfit_order3
```

```
##
## Call:
## lm(formula = d ~ h + I(h^2) + I(h^3), data = GalBall)
##
## Coefficients:
## (Intercept)            h       I(h^2)       I(h^3)
##       155.8       1115.3      -1244.9        547.7
```

```r
summary(Galfit_order3)
```

```
##
## Call:
## lm(formula = d ~ h + I(h^2) + I(h^3), data = GalBall)
##
## Residuals:
##        1        2        3        4        5        6        7
## -0.84138  2.32159 -0.08044 -4.46885  1.89175  3.58091 -2.40359
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   155.776      8.326  18.710 0.000333 ***
## h            1115.298     65.671  16.983 0.000445 ***
## I(h^2)      -1244.943    138.425  -8.994 0.002902 **
## I(h^3)        547.710     83.273   6.577 0.007150 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.011 on 3 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9987
## F-statistic:  1595 on 3 and 3 DF,  p-value: 2.662e-05
```

```r
Galfit_order4 <- lm(d~h+I(h^2)+I(h^3)+I(h^4),data=GalBall)
Galfit_order4
```

```
##
## Call:
## lm(formula = d ~ h + I(h^2) + I(h^3) + I(h^4), data = GalBall)
##
## Coefficients:
## (Intercept)            h       I(h^2)       I(h^3)       I(h^4)
##       138.3       1346.1      -2116.9       1766.4       -561.0
```

```r
summary(Galfit_order4)
```

```
##
## Call:
## lm(formula = d ~ h + I(h^2) + I(h^3) + I(h^4), data = GalBall)
```

```
##
## Residuals:
##       1       2       3       4       5       6       7
##  0.1708 -0.9279  2.3183 -2.3092  0.2576  0.9338 -0.4433
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   138.295      9.066  15.254  0.00427 **
## h            1346.071    106.071  12.690  0.00615 **
## I(h^2)      -2116.913    379.271  -5.582  0.03063 *
## I(h^3)       1766.391    518.566   3.406  0.07644 .
## I(h^4)       -561.014    237.498  -2.362  0.14201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.523 on 2 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9995
## F-statistic:  3024 on 4 and 2 DF,  p-value: 0.0003306
```

```
# These models reveal that the order 3 model is significant in it's highest-order term, and the fit is
# Question c
# out of the three fitted models, the cubic function in height seems preferable -- because, the relatio
Hseq <- seq(0.05,1.05,length=30)
Hseq
```

```
##  [1] 0.05000000 0.08448276 0.11896552 0.15344828 0.18793103 0.22241379
##  [7] 0.25689655 0.29137931 0.32586207 0.36034483 0.39482759 0.42931034
## [13] 0.46379310 0.49827586 0.53275862 0.56724138 0.60172414 0.63620690
## [19] 0.67068966 0.70517241 0.73965517 0.77413793 0.80862069 0.84310345
## [25] 0.87758621 0.91206897 0.94655172 0.98103448 1.01551724 1.05000000
```

```
Galpred <- predict(Galfit_order3,newdata=data.frame(h=Hseq),interval="confidence",level=0.9)
Galpred
```

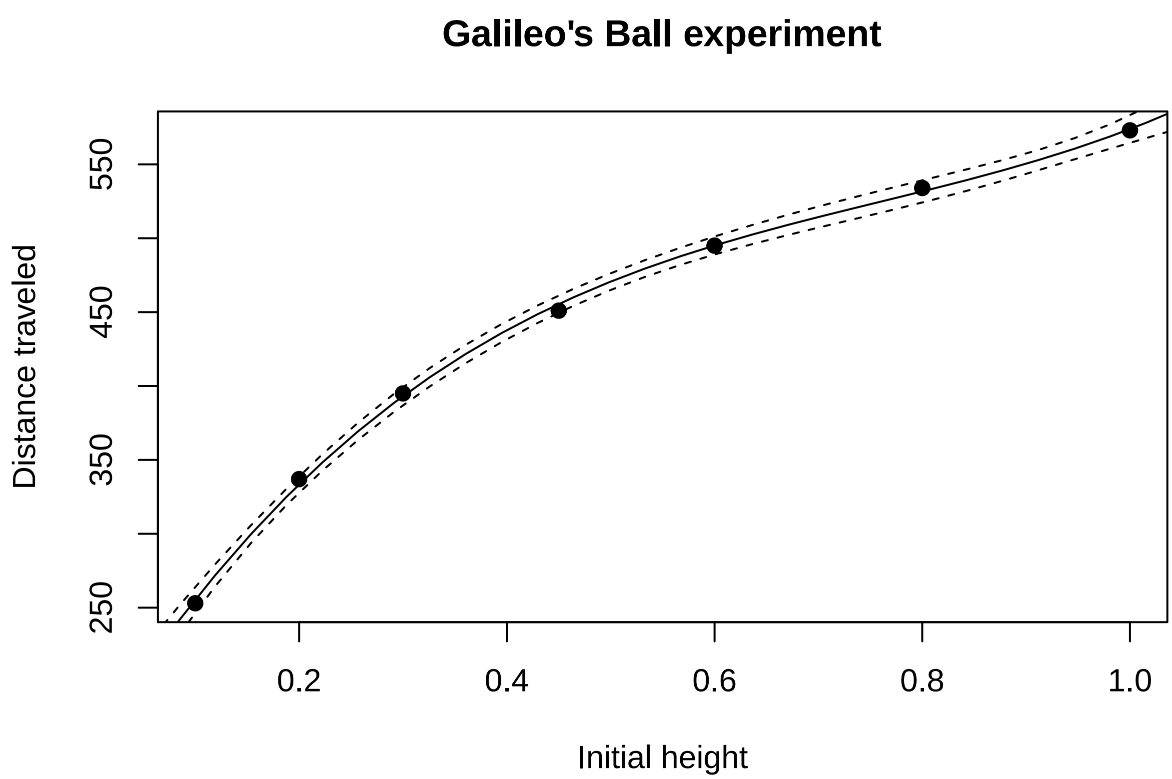
```
##         fit      lwr      upr
## 1  208.4965 195.2671 221.7259
## 2  241.4436 231.5535 251.3338
## 3  271.7603 264.2934 279.2272
## 4  299.5811 293.5557 305.6066
## 5  325.0410 319.5448 330.5372
## 6  348.2746 342.7144 353.8348
## 7  369.4166 363.5779 375.2554
## 8  388.6019 382.5078 394.6960
## 9  405.9651 399.7372 412.1930
## 10 421.6411 415.4181 427.8640
## 11 435.7644 429.6591 441.8697
## 12 448.4700 442.5444 454.3956
## 13 459.8925 454.1442 465.6408
## 14 470.1667 464.5265 475.8068
## 15 479.4273 473.7733 485.0813
## 16 487.8091 481.9972 493.6209
## 17 495.4467 489.3496 501.5439
## 18 502.4751 496.0125 508.9376
```

```
## 19 509.0288 502.1825 515.8750
## 20 515.2426 508.0569 522.4284
## 21 521.2514 513.8249 528.6779
## 22 527.1898 519.6621 534.7174
## 23 533.1925 525.7230 540.6620
## 24 539.3944 532.1270 546.6617
## 25 545.9301 538.9303 552.9298
## 26 552.9344 546.0812 559.7876
## 27 560.5420 553.3912 567.6929
## 28 568.8878 560.6319 577.1436
## 29 578.1064 567.7401 588.4726
## 30 588.3325 574.8501 601.8150
```

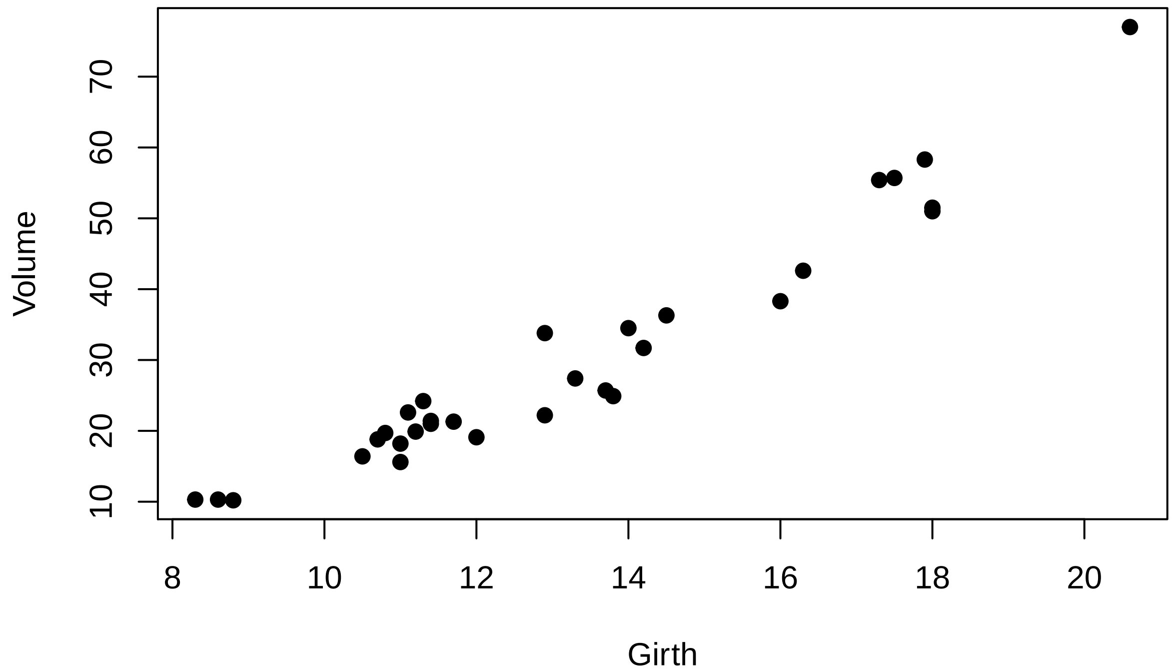
```
lines(Hseq,Galpred[,1])
lines(Hseq,Galpred[,2],lty=2)
lines(Hseq,Galpred[,3],lty=2)
# Question d
library("faraway")
```

```
##
## Attaching package: 'faraway'

## The following objects are masked from 'package:boot':
##
##     logit, melanoma
```



```r
?trees
plot(trees$Volume~trees$Girth,pch=19,xlab="Girth",ylab="Volume")
```




```r
# Question e
tree_fit1 <- lm(Volume~Girth+I(Girth^2),trees)
tree_fit1
```

```
##
## Call:
## lm(formula = Volume ~ Girth + I(Girth^2), data = trees)
##
## Coefficients:
## (Intercept)        Girth   I(Girth^2)
##     10.7863      -2.0921       0.2545
```

```r
summary(tree_fit1) ## "Mean volume" = 10.79 - 2.09*"girth" + 0.254*"girth^2"
```

```
##
## Call:
## lm(formula = Volume ~ Girth + I(Girth^2), data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4889 -2.4293 -0.3718  2.0764  7.6447
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.78627   11.22282   0.961 0.344728
## Girth       -2.09214    1.64734  -1.270 0.214534
## I(Girth^2)   0.25454    0.05817   4.376 0.000152 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.335 on 28 degrees of freedom
## Multiple R-squared:  0.9616, Adjusted R-squared:  0.9588
## F-statistic: 350.5 on 2 and 28 DF,  p-value: < 2.2e-16
```

```r
tree_fit2 <- lm(log(Volume)~log(Girth),trees)
tree_fit2
```

```
##
## Call:
## lm(formula = log(Volume) ~ log(Girth), data = trees)
##
## Coefficients:
## (Intercept)   log(Girth)
##      -2.353        2.200
```

```r
summary(tree_fit2) ## "Mean log(volume)" = -2.35 + 2.20*"log(girth)"
```

```
##
## Call:
## lm(formula = log(Volume) ~ log(Girth), data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.205999 -0.068702  0.001011  0.072585  0.247963
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.35332    0.23066  -10.20 4.18e-11 ***
## log(Girth)   2.19997    0.08983   24.49  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.115 on 29 degrees of freedom
## Multiple R-squared:  0.9539, Adjusted R-squared:  0.9523
## F-statistic: 599.7 on 1 and 29 DF,  p-value: < 2.2e-16
```

```r
# Coefficients of determination are similar; the quadratic model is slightly higher. Both indicate a st
# Question f

# The fitted values of the models themselves are extremely similar. However, the prediction intervals t
# Question g
library("MASS")
car_fit <- lm(mpg~wt+hp+disp,data=mtcars)
summary(car_fit)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp + disp, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.891 -1.640 -0.172  1.061  5.861
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.105505   2.110815  17.579  < 2e-16 ***
## wt          -3.800891   1.066191  -3.565  0.00133 **
## hp          -0.031157   0.011436  -2.724  0.01097 *
## disp        -0.000937   0.010350  -0.091  0.92851
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.639 on 28 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8083
## F-statistic: 44.57 on 3 and 28 DF,  p-value: 8.65e-11
```

```r
# Question h
car_fit <- lm(I(1/mpg)~wt+hp+disp,data=mtcars)
summary(car_fit)
```

```
##
## Call:
## lm(formula = I(1/mpg) ~ wt + hp + disp, data = mtcars)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0163719 -0.0043511  0.0008672  0.0032544  0.0133345
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.496e-03  5.322e-03   1.784  0.08521 .
## wt          9.469e-03  2.688e-03   3.522  0.00149 **
## hp          5.864e-05  2.883e-05   2.034  0.05155 .
## disp        2.456e-05  2.609e-05   0.941  0.35472
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006653 on 28 degrees of freedom
## Multiple R-squared:  0.8518, Adjusted R-squared:  0.8359
## F-statistic: 53.63 on 3 and 28 DF,  p-value: 9.94e-12
```

```r
# Both fits to the mtcars data here provide similar levels of significance for the three predictors; th
```

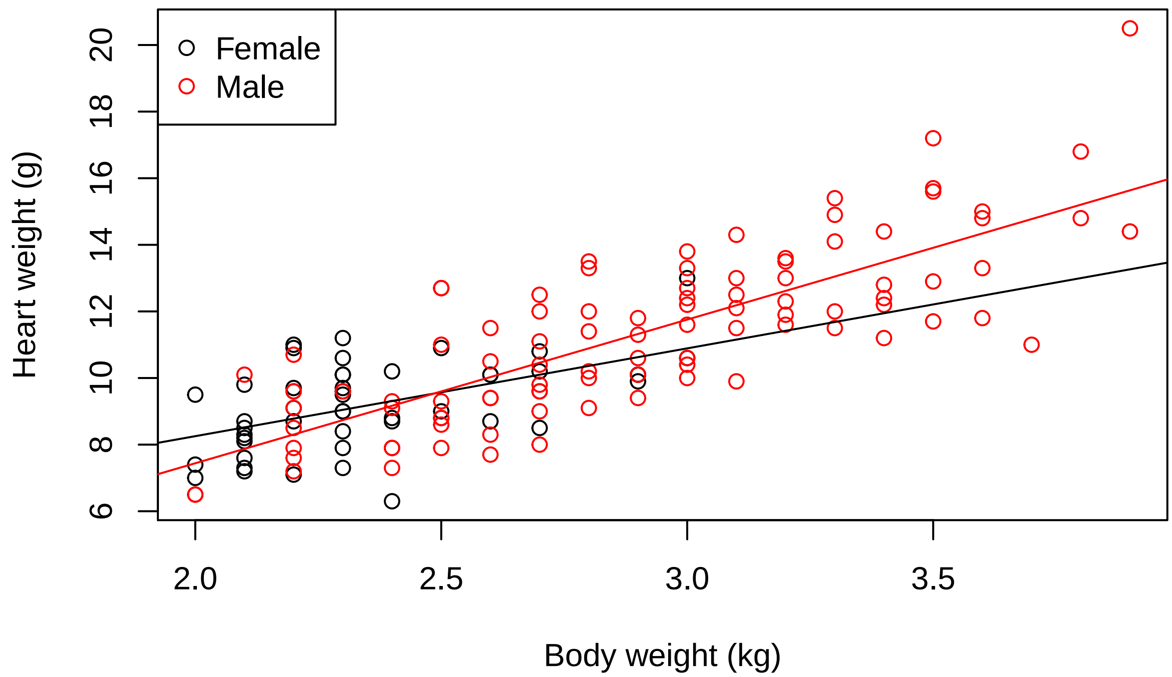
#Exercise 21.3

```r
## 21.3
library("MASS")
# Question a
```

```r
cat_fit <- lm(Hwt~Bwt*Sex,data=cats)
summary(cat_fit)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt * Sex, data = cats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7728 -1.0118 -0.1196  0.9272  4.8646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9813     1.8428   1.618 0.107960
## Bwt           2.6364     0.7759   3.398 0.000885 ***
## SexM         -4.1654     2.0618  -2.020 0.045258 *
## Bwt:SexM      1.6763     0.8373   2.002 0.047225 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.442 on 140 degrees of freedom
## Multiple R-squared:  0.6566, Adjusted R-squared:  0.6493
## F-statistic: 89.24 on 3 and 140 DF,  p-value: < 2.2e-16
```

```r
# The main-effects-only version of the model had a mild negative effect of "sex male", and it was not s
# Question b
plot(cats$Bwt,cats$Hwt,col=cats$Sex,ylab="Heart weight (g)",xlab="Body weight (kg)")
legend("topleft",legend=c("Female","Male"),col=1:2,pch=1)
cat_coefs <- coef(cat_fit)
abline(coef=cat_coefs[1:2])
abline(coef=c(sum(cat_coefs[c(1,3)]),sum(cat_coefs[c(2,4)])),col=2)
```




```r
# Lines of the fitted model are no longer parallel; the effect of the weakly significant interaction is
# Question c
predict(cat_fit,newdata=data.frame(Bwt=3.4,Sex="F"),interval="prediction",level=0.95)
```

```
##        fit      lwr      upr
## 1 11.94512 8.651786 15.23845
```

```r
# Sigma's heart weight predicted from the new model is around 1.5 grams lighter than predicted from the
# Question d
library("faraway")
tree_fit1 <- lm(Volume~Girth+Height,data=trees)
summary(tree_fit1)
```

```
##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4065 -2.6493 -0.2876  2.2003  8.4847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
## Girth         4.7082     0.2643  17.816  < 2e-16 ***
```

```
## Height        0.3393     0.1302   2.607   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.882 on 28 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9442
## F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

```
tree_fit2 <- lm(Volume~Girth*Height,data=trees)
summary(tree_fit2)
```

```
##
## Call:
## lm(formula = Volume ~ Girth * Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5821 -1.0673  0.3026  1.5641  4.6649
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  69.39632   23.83575   2.911  0.00713 **
## Girth        -5.85585    1.92134  -3.048  0.00511 **
## Height       -1.29708    0.30984  -4.186  0.00027 ***
## Girth:Height  0.13465    0.02438   5.524 7.48e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.709 on 27 degrees of freedom
## Multiple R-squared:  0.9756, Adjusted R-squared:  0.9728
## F-statistic: 359.3 on 3 and 27 DF,  p-value: < 2.2e-16
```

```
# Question e
tree_fit3 <- lm(log(Volume)~log(Girth)+log(Height),data=trees)
summary(tree_fit3)
```

```
##
## Call:
## lm(formula = log(Volume) ~ log(Girth) + log(Height), data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.168561 -0.048488  0.002431  0.063637  0.129223
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.63162    0.79979  -8.292 5.06e-09 ***
## log(Girth)   1.98265    0.07501  26.432  < 2e-16 ***
## log(Height)  1.11712    0.20444   5.464 7.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08139 on 28 degrees of freedom
```

```
## Multiple R-squared:  0.9777, Adjusted R-squared:  0.9761
## F-statistic: 613.2 on 2 and 28 DF,  p-value: < 2.2e-16
```

```r
tree_fit4 <- lm(log(Volume)~log(Girth)*log(Height),data=trees)
summary(tree_fit4)
```

```
##
## Call:
## lm(formula = log(Volume) ~ log(Girth) * log(Height), data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.165941 -0.048613  0.006384  0.062204  0.132295
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -3.6869     7.6996  -0.479    0.636
## log(Girth)               0.7942     3.0910   0.257    0.799
## log(Height)              0.4377     1.7788   0.246    0.808
## log(Girth):log(Height)   0.2740     0.7124   0.385    0.704
##
## Residual standard error: 0.08265 on 27 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9753
## F-statistic: 396.4 on 3 and 27 DF,  p-value: < 2.2e-16
```

```r
# The interactive effect is highly significant in the untransformed model from (d), but completely non-
# Question f
car_fit <- lm(mpg~factor(cyl)*hp+wt,data=mtcars)
summary(car_fit)
```

```
##
## Call:
## lm(formula = mpg ~ factor(cyl) * hp + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1864 -1.4098 -0.4022  1.0186  4.3920
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      41.87732    3.23293  12.953 1.37e-12 ***
## factor(cyl)6     -9.98213    5.76950  -1.730 0.095931 .
## factor(cyl)8    -11.72793    4.22507  -2.776 0.010276 *
## hp               -0.09947    0.03487  -2.853 0.008576 **
## wt               -3.05994    0.68275  -4.482 0.000143 ***
## factor(cyl)6:hp   0.07809    0.05236   1.492 0.148335
## factor(cyl)8:hp   0.08602    0.03703   2.323 0.028601 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.3 on 25 degrees of freedom
## Multiple R-squared:  0.8826, Adjusted R-squared:  0.8544
## F-statistic: 31.32 on 6 and 25 DF,  p-value: 1.831e-10
```

```r
# Question g
coef(car_fit)
```

```
##     (Intercept)    factor(cyl)6    factor(cyl)8              hp
##     41.87732085     -9.98213264    -11.72792959     -0.09946598
##              wt factor(cyl)6:hp factor(cyl)8:hp
##     -3.05993524      0.07808919      0.08602496
```

```r
# The interactive effect is between a continuous (hp) and a categorical (factor(cyl)) predictor. As such
coef(car_fit)[4] # When the car has 4 cylinders (reference level), the slope for hp is -0.0995 (to 4 de
```

```
##          hp
## -0.09946598
```

```r
coef(car_fit)[4] + coef(car_fit)[6] # When the car has 6 cylinders, the slope for hp is -0.0995 + 0.078
```

```
##          hp
## -0.02137679
```

```r
coef(car_fit)[4] + coef(car_fit)[7] # When the car has 8 cylinders, the slope for hp is -0.0995 + 0.086
```

```
##          hp
## -0.01344103
```

```r
# This model suggests that as hp increases, mean MPG decreases (for a fixed wt). However, in comparison
# Question h
# i
predict(car_fit,newdata=data.frame(wt=c(2.1,3.9,2.9),hp=c(100,210,200),cyl=c(4,8,6)),interval="confiden
```

```
##        fit      lwr      upr
## 1 25.50486 23.57668 27.43304
## 2 15.39303 14.11928 16.66678
## 3 18.74602 12.29560 25.19644
```

```r
# The first car is the only car that has a point estimate of mean MPG that is higher than your mother's
# ii
# Although the point estimate for Car 3 is much less than 25, looking at the confidence intervals you c
```