Practicum Problems

June 5, 2022

1 q1

[2]:
```
library("tidyverse")
library('ggplot2')
library(datasets)
data(iris)
```

[3]:
```
summary(iris)
```

```
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
setosa    :50
versicolor:50
virginica :50
```

[4]:
```
head(iris)
```

A data.frame: 6 × 5

	Sepal.Length <dbl>	Sepal.Width <dbl>	Petal.Length <dbl>	Petal.Width <dbl>	Species <fct>
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

[5]:
```
unique(iris['Species'])
```

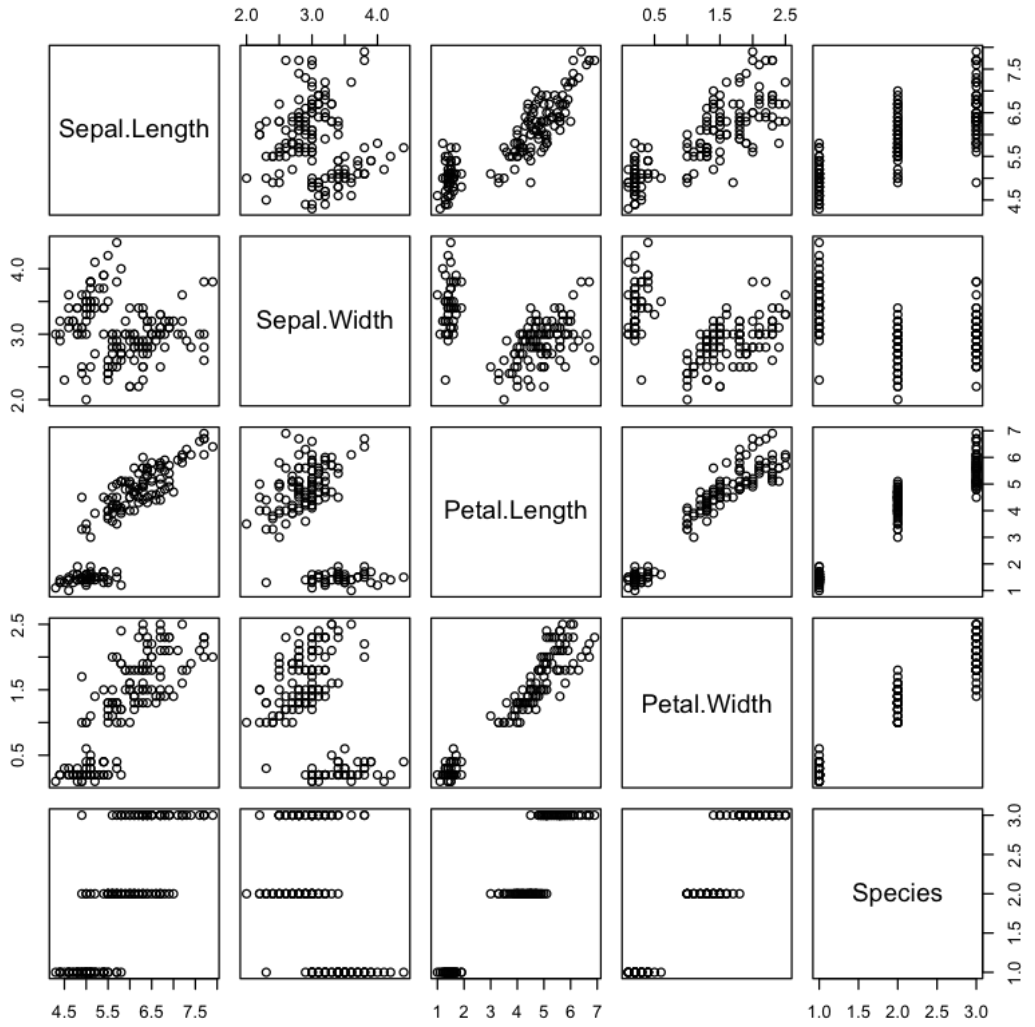
A data.frame: 3 × 1

	Species <fct>
1	setosa
51	versicolor
101	virginica

[6]:
```
str(iris)
```

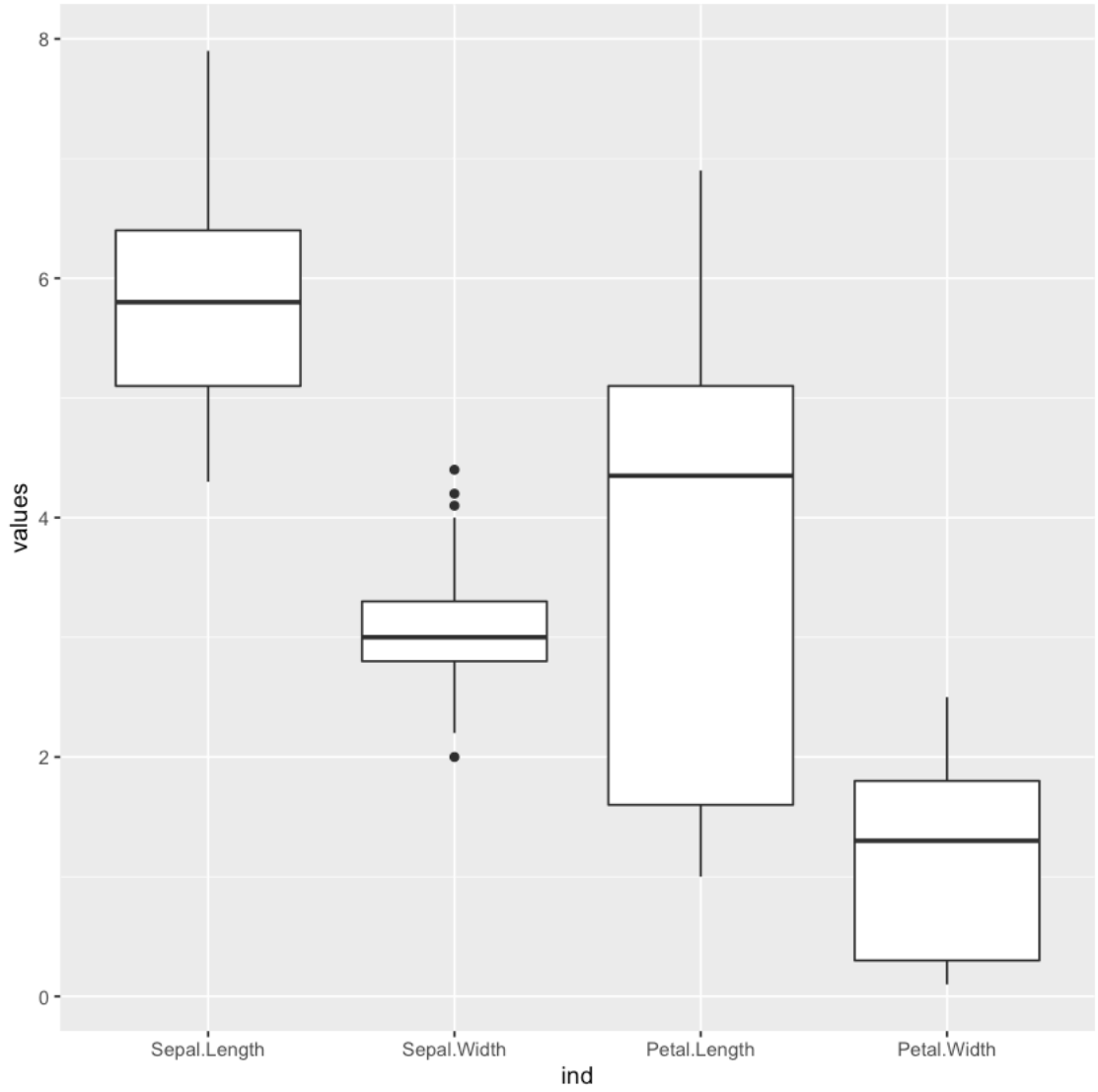
```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 …
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 …
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 …
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 …
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1
1 …
```

[7]:
```
pairs(iris)
```



[8]:
```
ggplot(data=stack(iris), mapping = aes(x=ind, y = values)) +
         geom_boxplot()
```

```
Warning message in stack.data.frame(iris):
"non-vector columns will be ignored"
```



[9]:
```
SepalLength <- iris$Sepal.Length
SepalWidth <- iris$Sepal.Width
PetalLength <- iris$Petal.Length
PetalWidth <- iris$Petal.Width
IQR(SepalLength)
```

1.3

[10]:
```
IQR(SepalWidth)
```

0.5

[11]:
```
IQR(PetalLength)
```

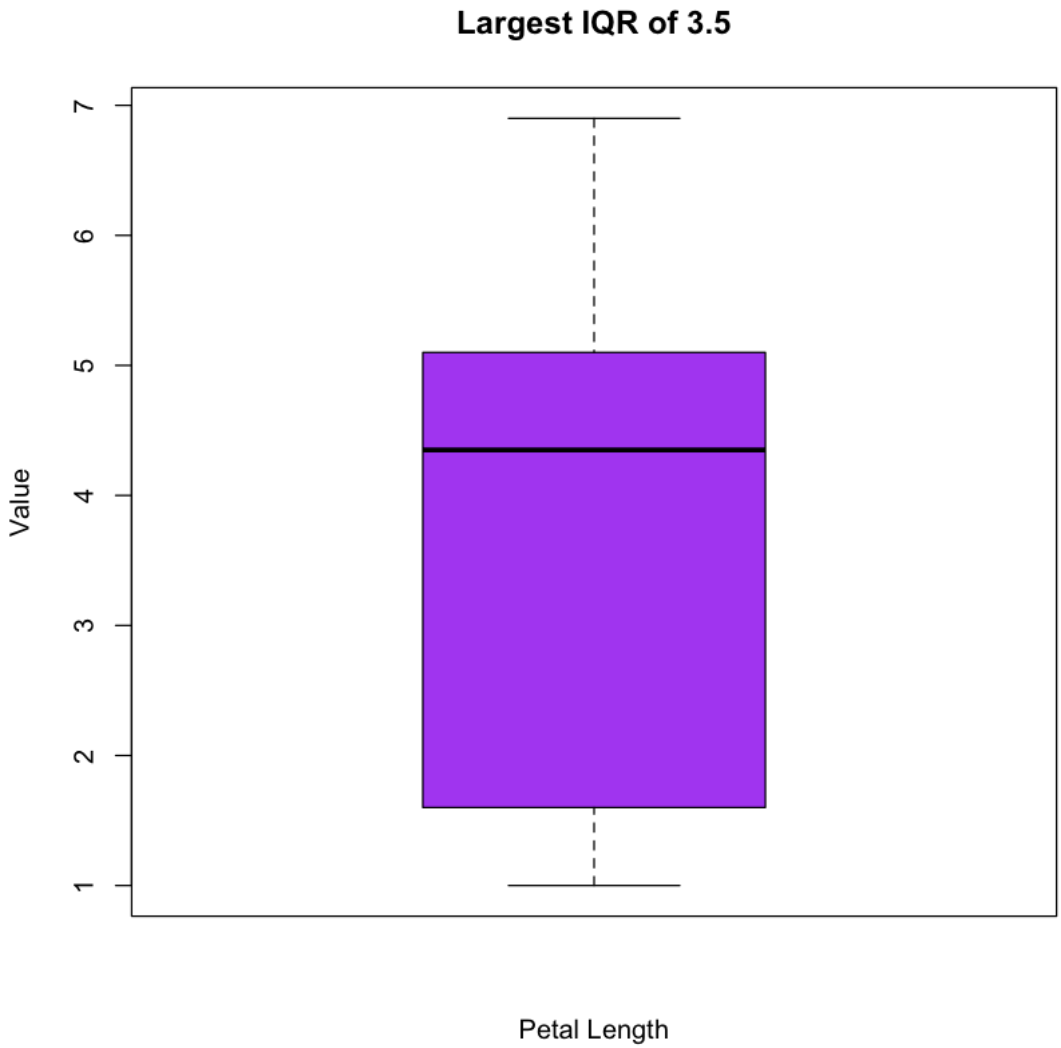
3.5

[12]:
```
IQR(PetalWidth)
```

1.5

Highlight the Petal.Length feature. It has the highest empiral IQR at 3.5.

[13]:
```
boxplot(iris$Petal.Length, main="Largest IQR of 3.5", xlab="Petal Length",ylab␣
↪="Value", col="purple" )
```



Calculate the parametric standard deviation for each feature

[14]:
```
sd(SepalLength)
```

0.828066127977863

[15]:
```
sd(SepalWidth)
```

0.435866284936698

[16]:
```
sd(PetalLength)
```

1.76529823325947

[17]:
```
sd(PetalWidth)
```

0.762237668960346

[18]:
```
mean(SepalLength)
```

5.84333333333333

[19]:
```
median((SepalLength))
```

5.8

[20]:
```
mean(SepalWidth)
```

3.05733333333333

[21]:
```
median((SepalWidth))
```

3

[22]:
```
mean(PetalLength)
```

3.758

[23]:
```
median((PetalLength))
```

4.35

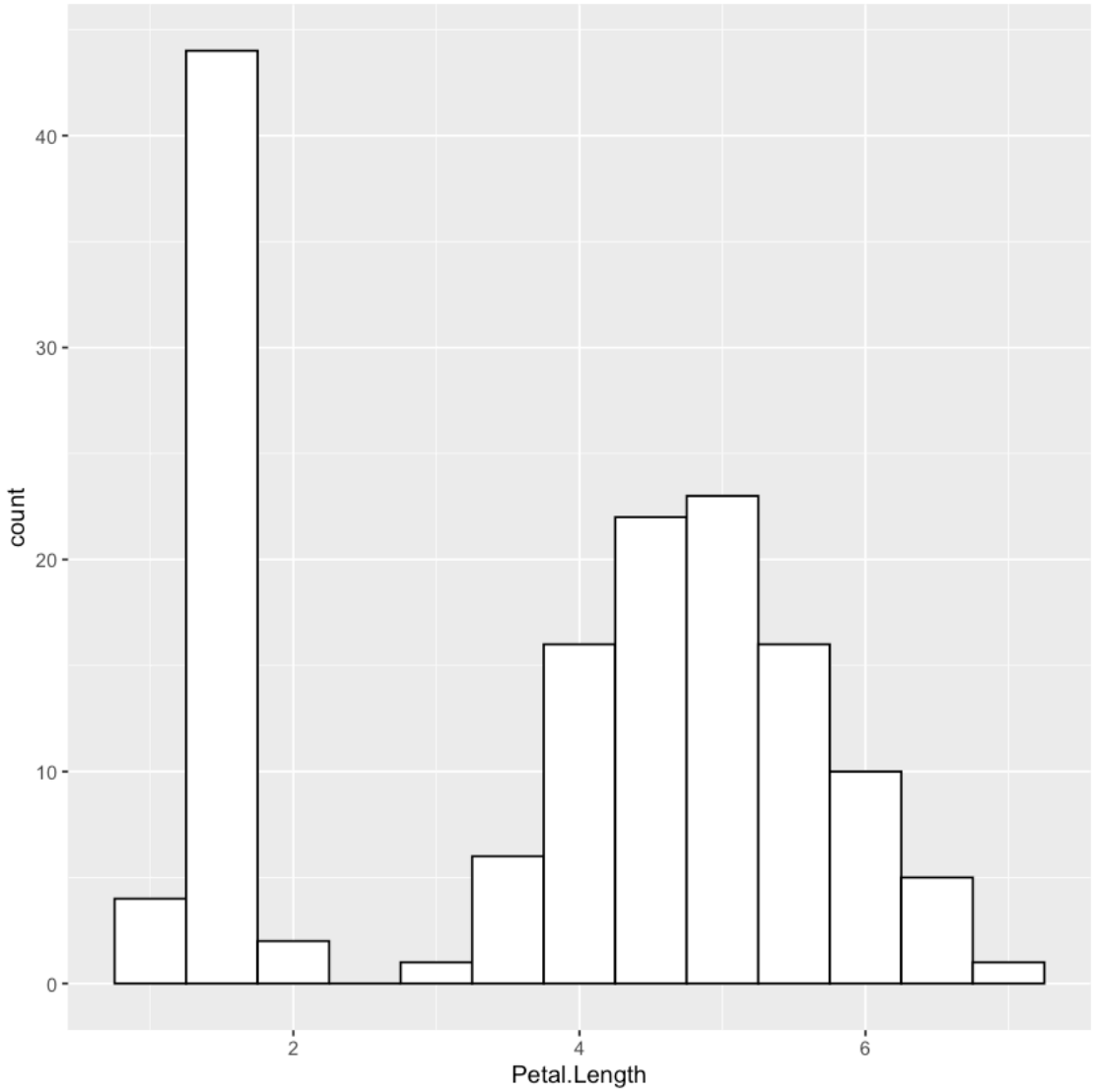
[24]:
```
mean(PetalLength)
```

3.758

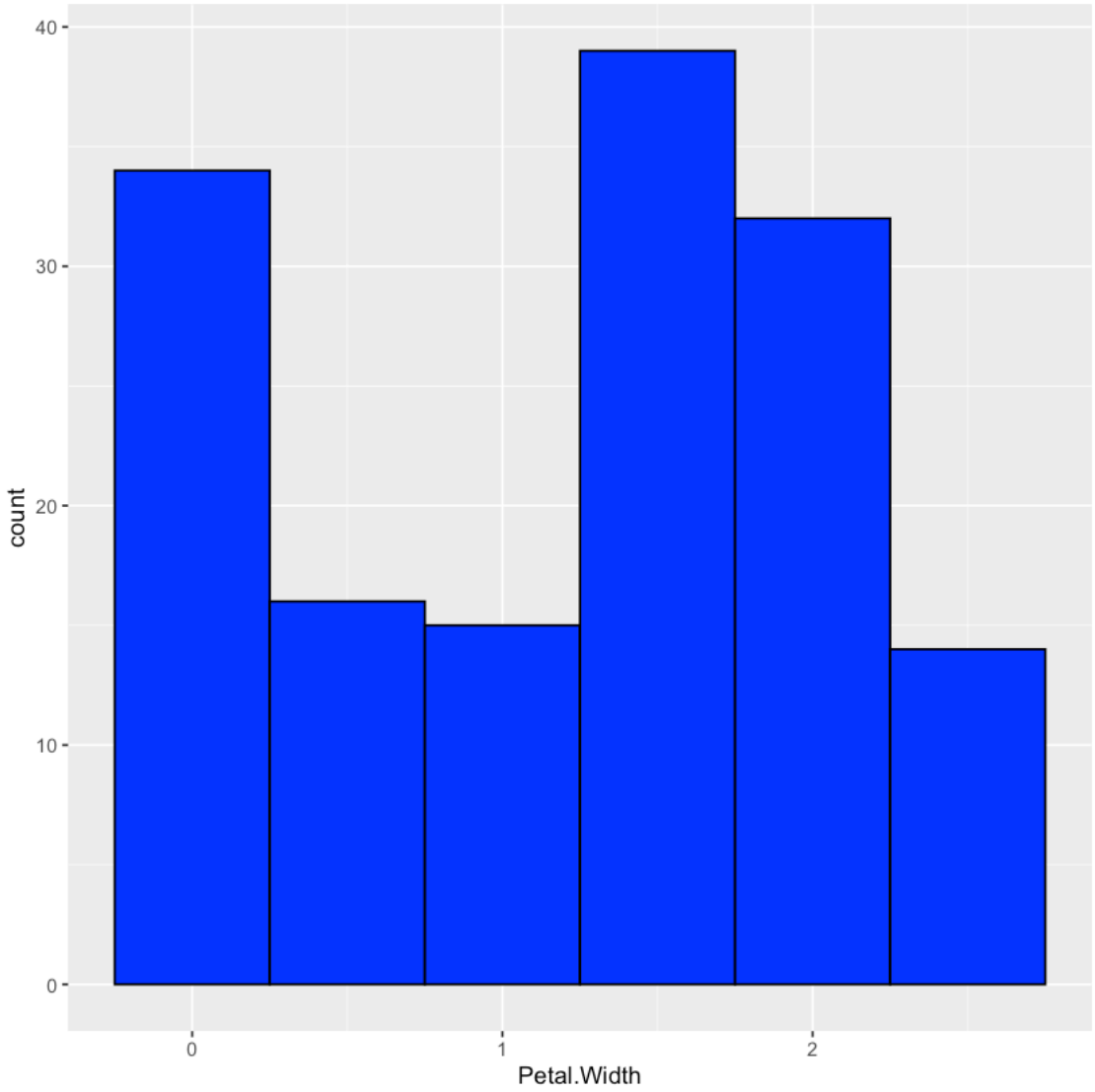
[25]:
```
median((PetalLength))
```

4.35

[26]:
```
ggplot(iris, aes(x=Petal.Length)) +
  geom_histogram(color="black", fill="white", binwidth = .5)
```



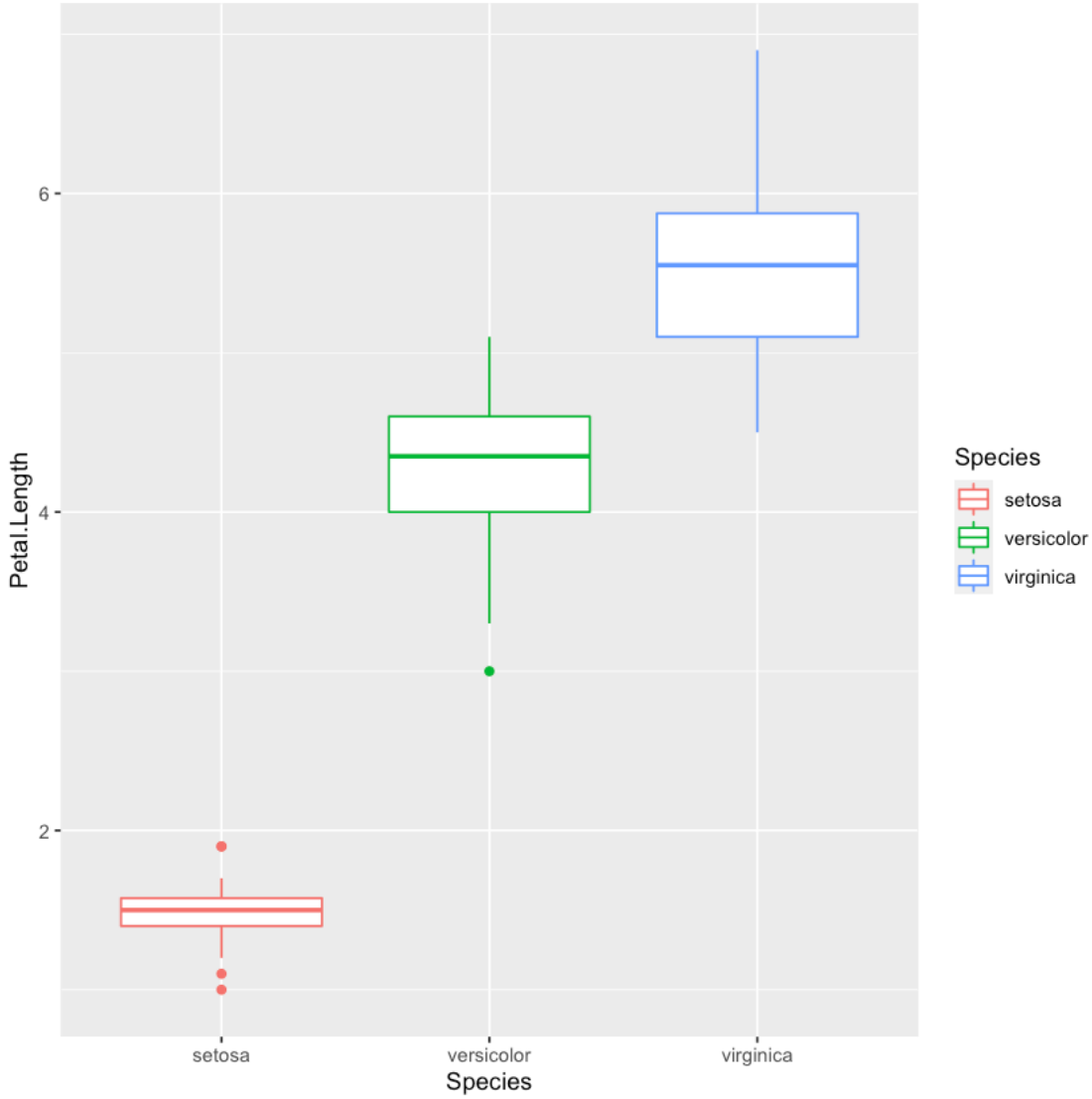
[27]:
```
ggplot(iris, aes(x=Petal.Width)) +
  geom_histogram(color="black", fill="blue", binwidth = .5)
```



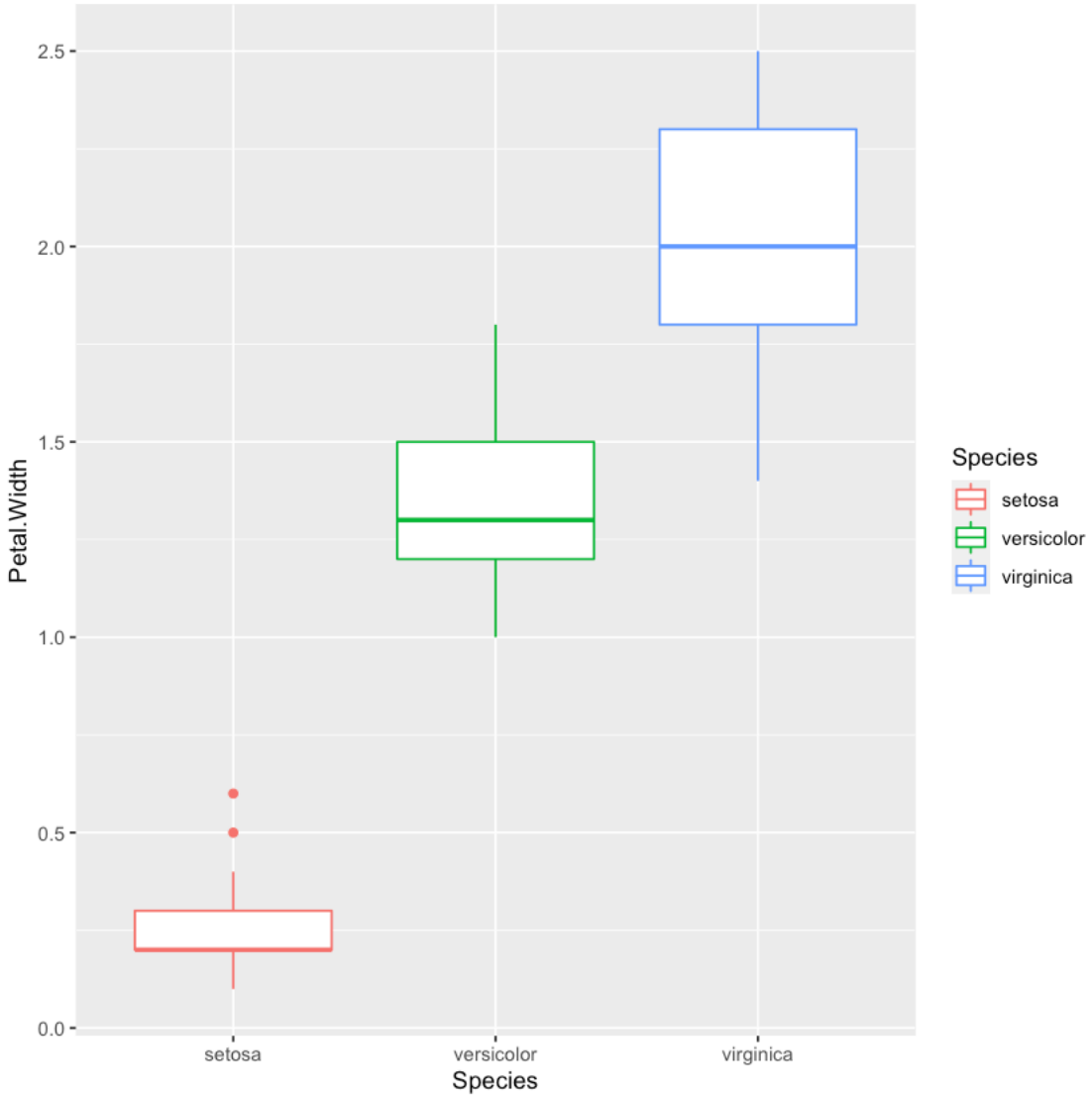
In order to compare the empirical values with the results, parametric and non parametric functions had to be calculated. The parametric functions; mean and standard deviation were calculated; and the non-parametric functions; median and histrogram/boxplot graphs were used. The parametric functions: average and standard deviation, inherently take on the assumption that the data follows a normal distribution. The standard deviation values calculated above for the four features, shows that Petal Length and Petal Width to not follow a normal distribution. Therefore, the results do not agree with the empirical values. Thus, a normal distribution should not be assumed for the two petal features. Many of the observed petals have a significant amount of small data points causing the data to be negatively skewed. Therefore, a histogram and boxplot are better for indicting the true nature of the data.

Use the ggplot2 libraryfrom CRAN to create a colored boxplot for each feature, with a box-whisker per flower species.

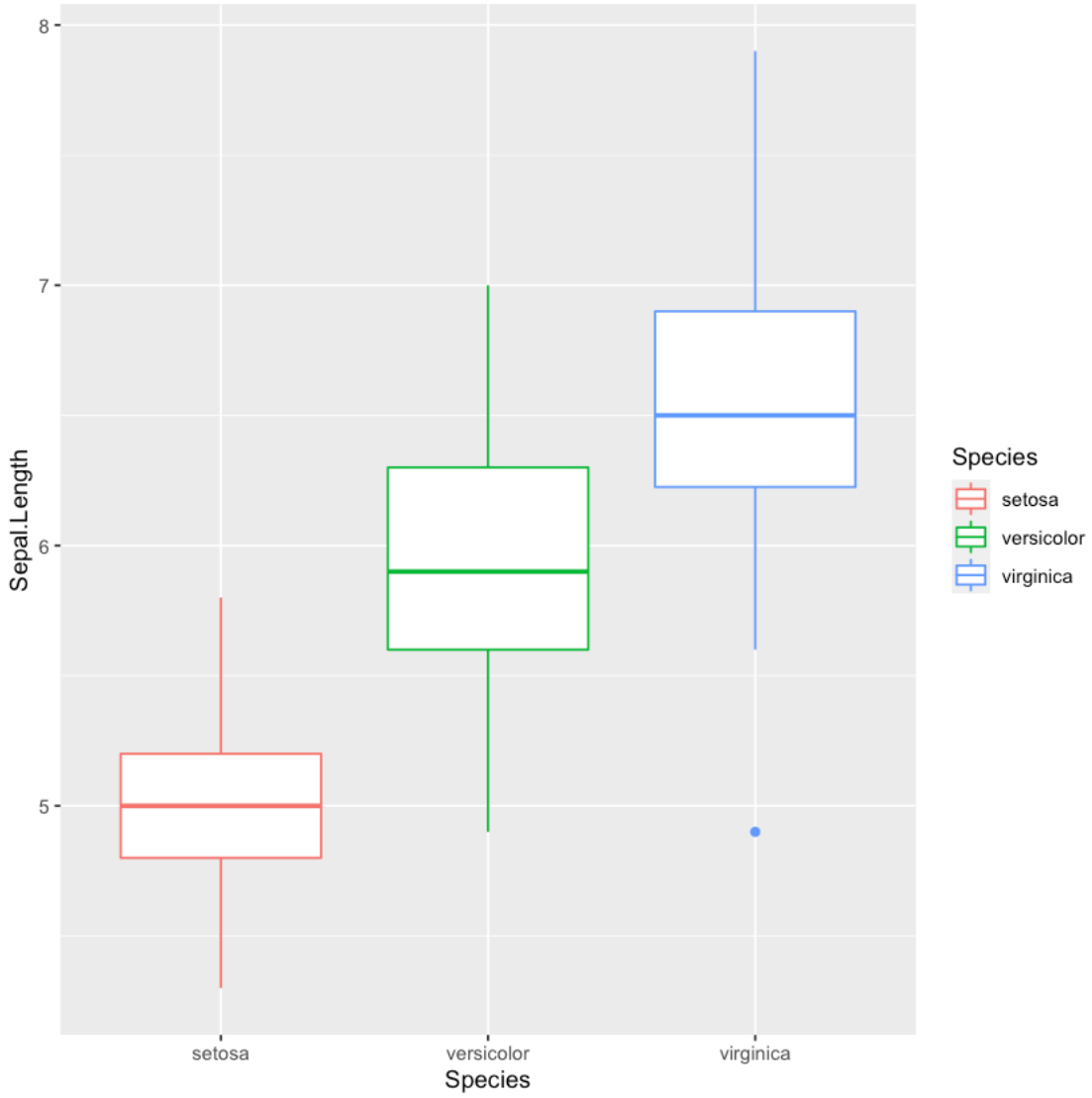
[28]:
```
ggplot(data=iris)+
  geom_boxplot(mapping = aes(x=Species,y=Petal.Length, color = Species))
```



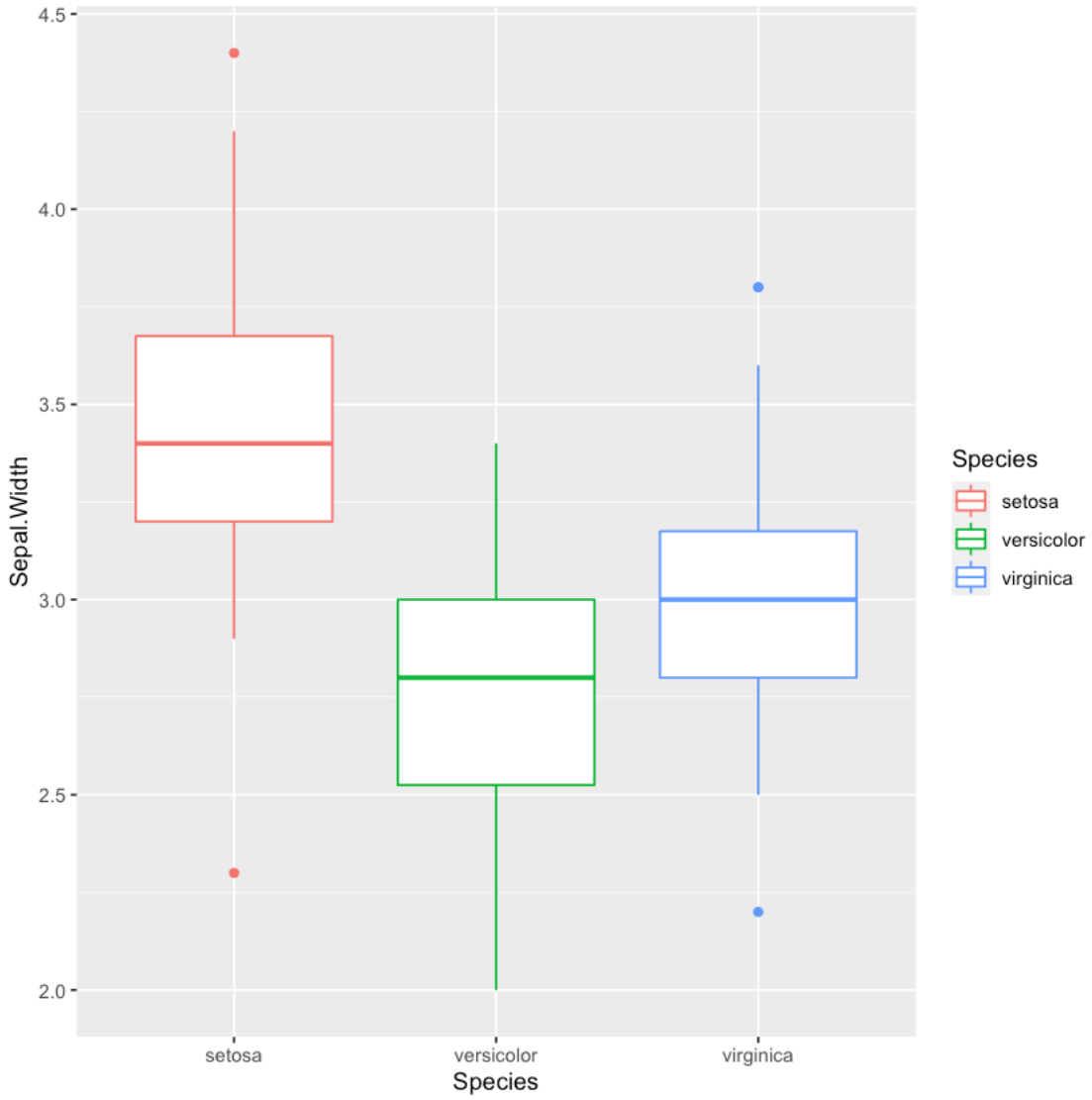
[29]:
```
ggplot(data=iris)+
  geom_boxplot(mapping = aes(x=Species,y=Petal.Width, color = Species))
```



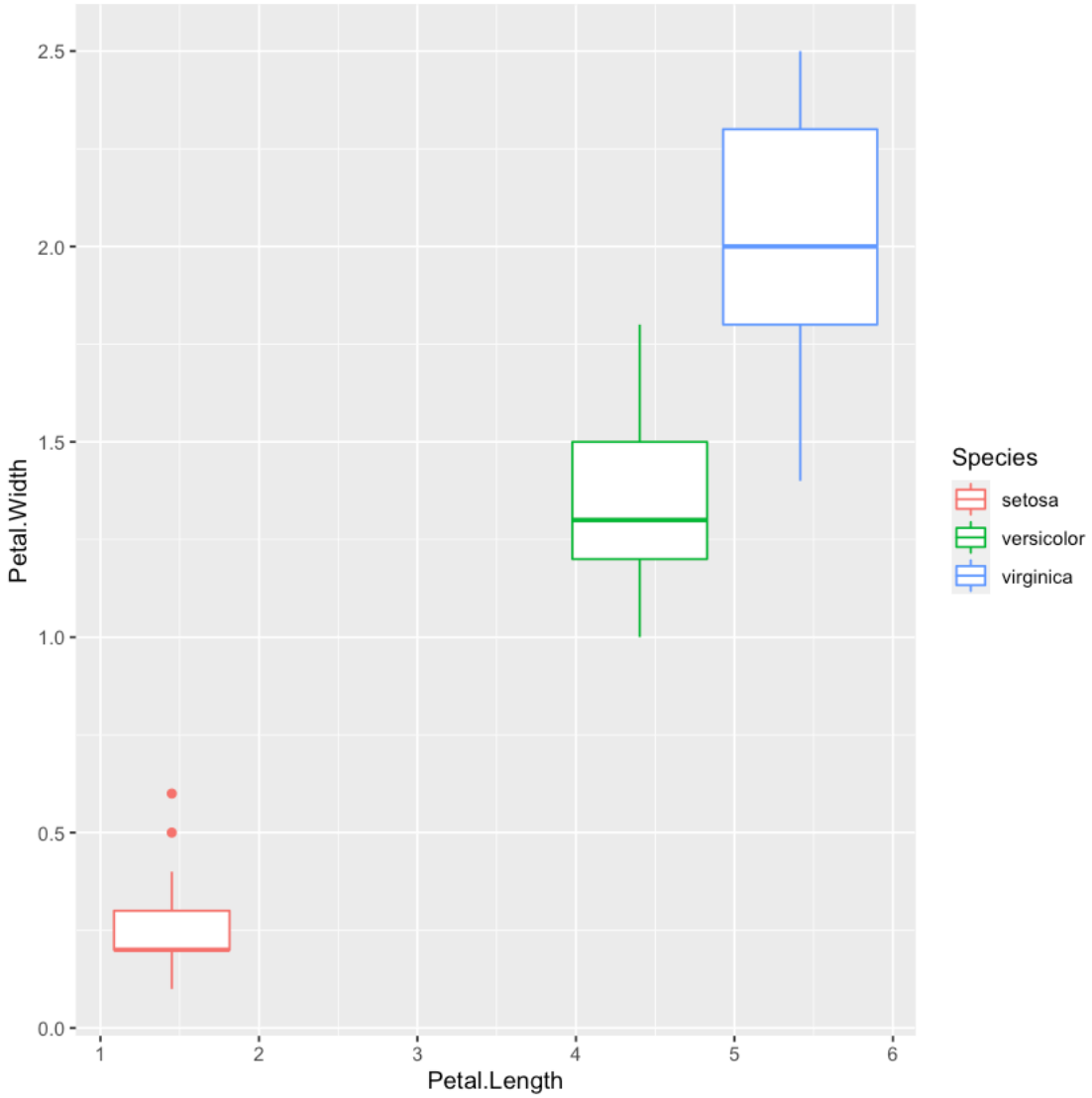
[32]:
```
ggplot(data=iris)+
  geom_boxplot(mapping = aes(x=Species,y=Sepal.Length, color = Species))
```



[33]:
```
ggplot(data=iris)+
  geom_boxplot(mapping = aes(x=Species,y=Sepal.Width, color = Species))
```



[34]:
```
ggplot(data=iris)+
  geom_boxplot(mapping = aes(x=Petal.Length,y=Petal.Width, color = Species))
```



For Virginica the range and IQR between the Petal Length/Width is significantly different. The blue boxplot highlights this.

2 q2

[35]:
```
data(trees)
summary(trees)
```

```
     Girth           Height       Volume     
 Min.   : 8.30   Min.   :63   Min.   :10.20  
 1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40  
 Median :12.90   Median :76   Median :24.20  
 Mean   :13.25   Mean   :76   Mean   :30.17  
```

```
 3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
 Max.   :20.60   Max.   :87   Max.   :77.00
```

[36]:
```
fivenum(trees$Girth)
```

1. 8.3 2. 11.05 3. 12.9 4. 15.25 5. 20.6

[37]:
```
fivenum(trees$Height)
```

1. 63 2. 72 3. 76 4. 80 5. 87

[38]:
```
fivenum(trees$Volume)
```

1. 10.2 2. 19.4 3. 24.2 4. 37.3 5. 77

[39]:
```
head(trees)
```

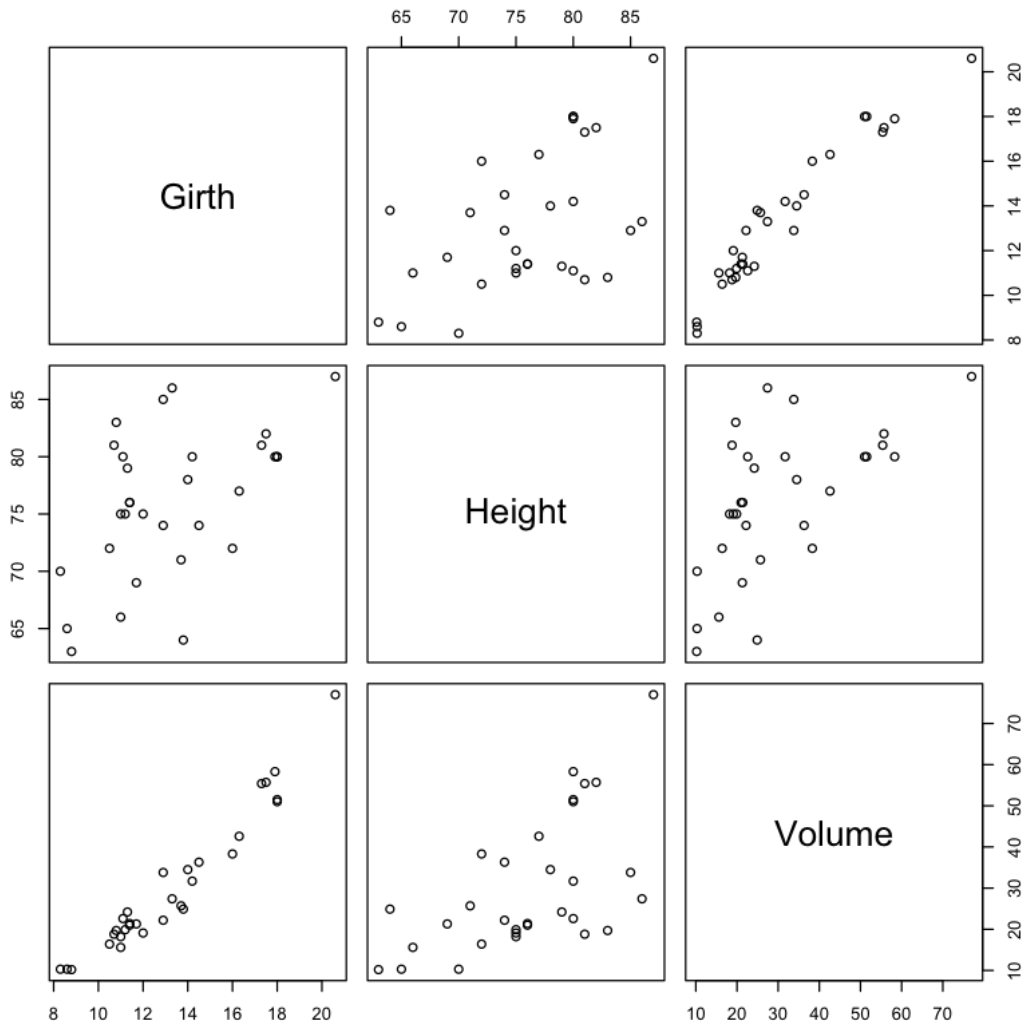
A data.frame: 6 × 3

	Girth <dbl>	Height <dbl>	Volume <dbl>
1	8.3	70	10.3
2	8.6	65	10.3
3	8.8	63	10.2
4	10.5	72	16.4
5	10.7	81	18.8
6	10.8	83	19.7

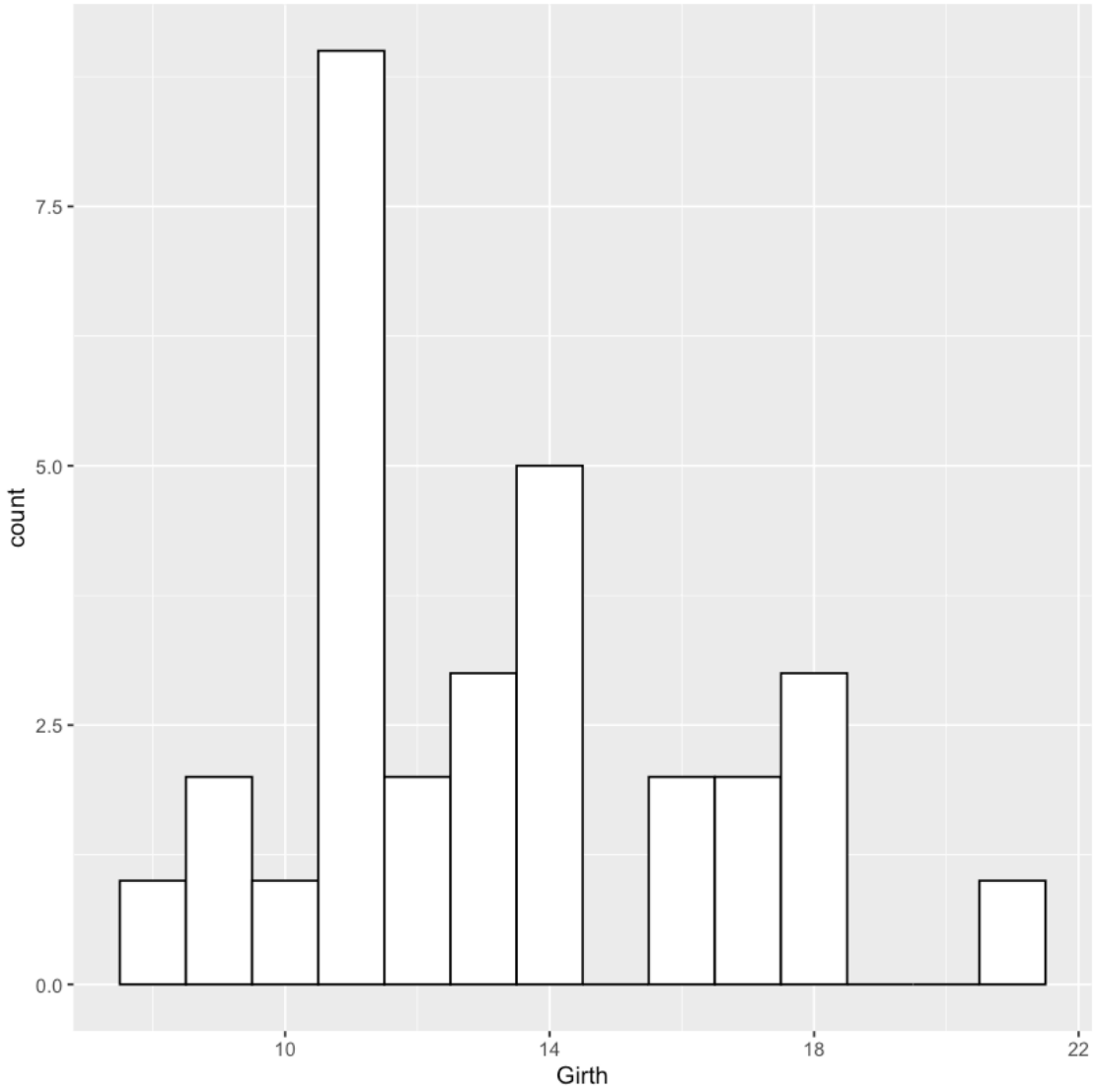
[40]:
```
str(trees)
```

```
'data.frame':   31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 …
 $ Height: num  70 65 63 72 81 83 66 75 80 75 …
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 …
```

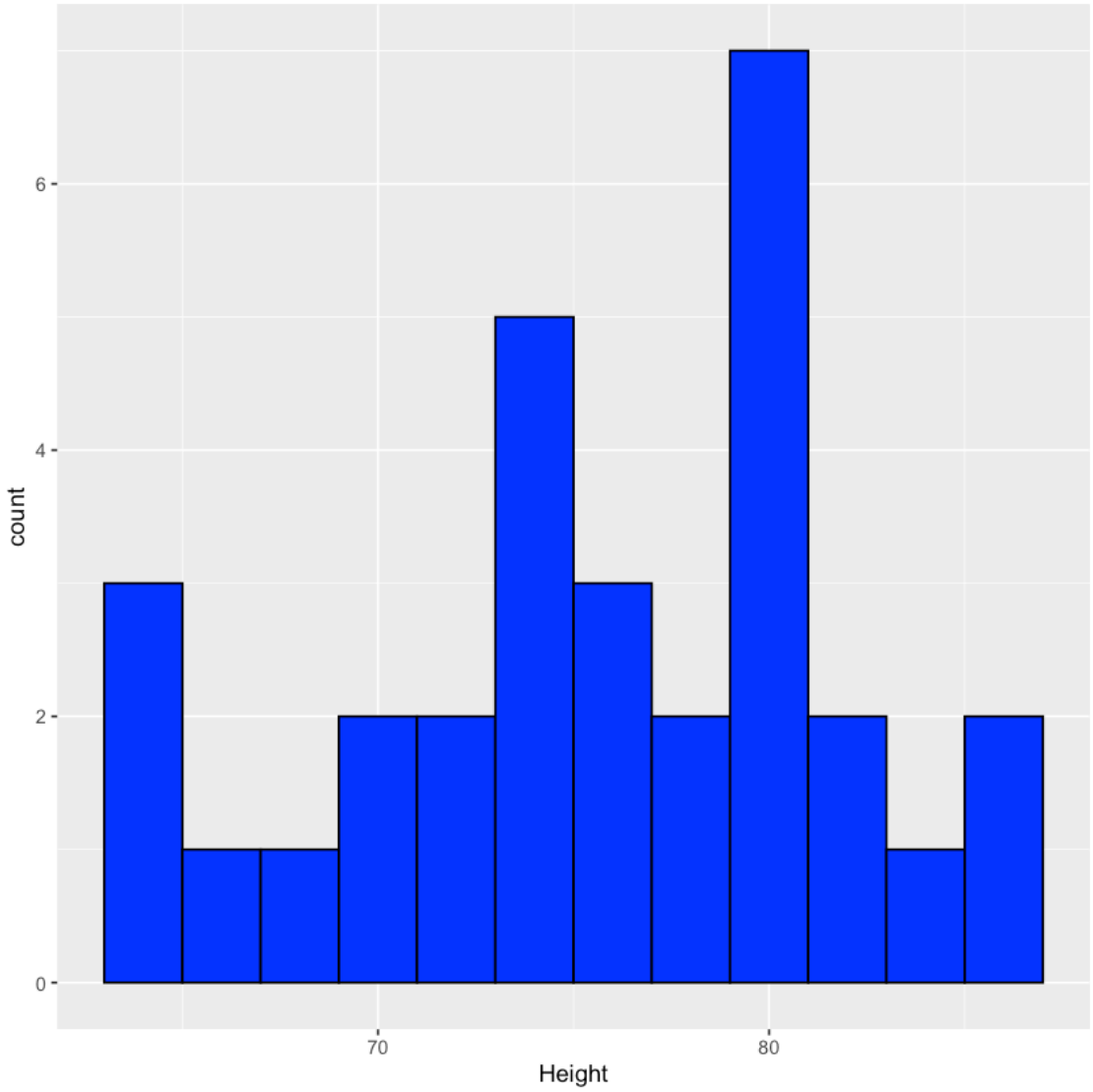
[41]:
```
pairs(trees)
```



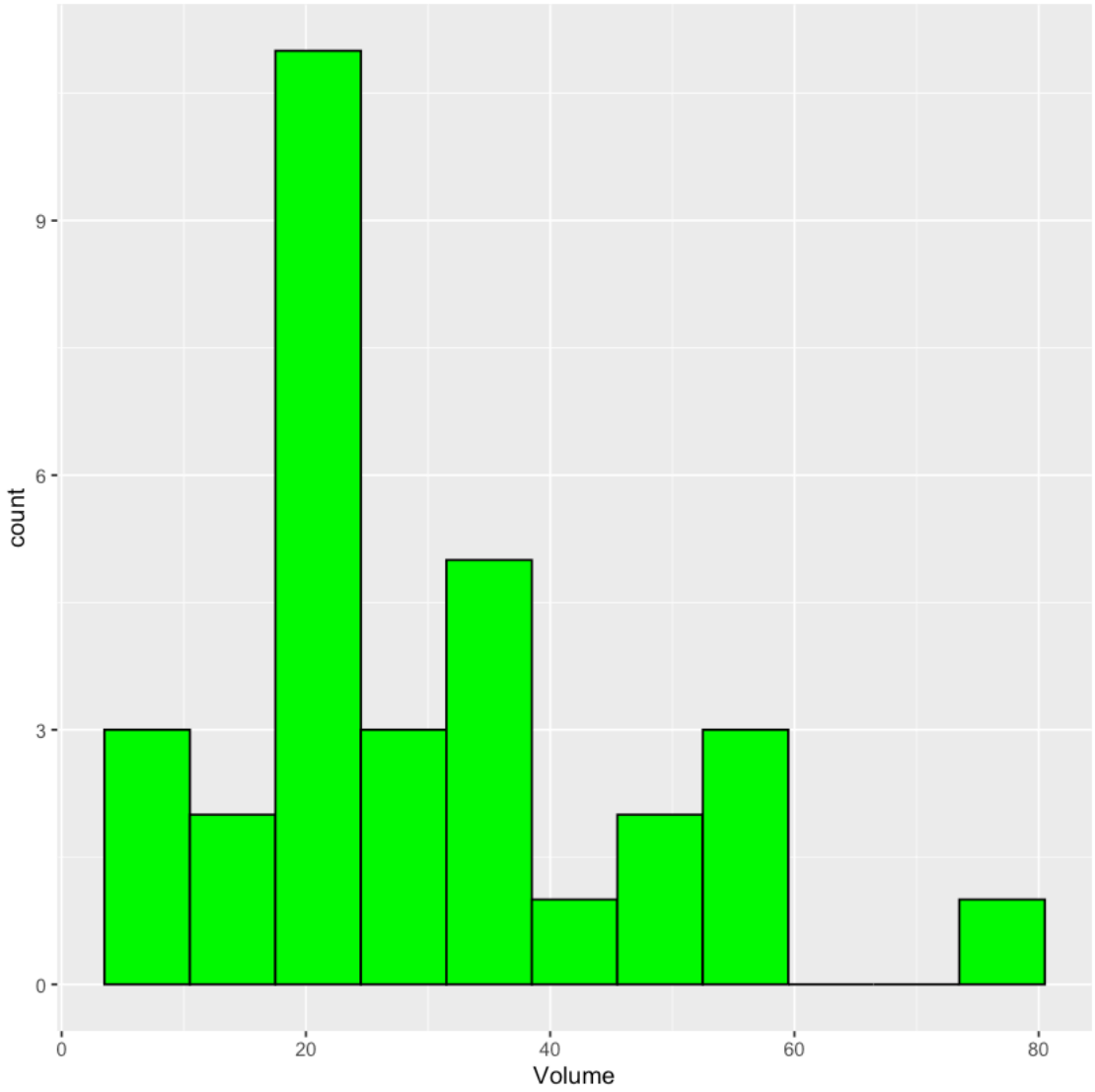
[42]:
```
ggplot(trees, aes(x=Girth)) +
  geom_histogram(color="black", fill="white", binwidth = 1)
```



[43]:
```
ggplot(trees, aes(x=Height)) +
  geom_histogram(color="black", fill="blue", binwidth = 2)
```



[44]:
```
ggplot(trees, aes(x=Volume)) +
  geom_histogram(color="black", fill="green", binwidth = 7)
```



[5]:
```
install.packages("moments")
```

```
The downloaded binary packages are in
/var/folders/fw/l8cr2wvs1hx7xhjf4l8dd_tc0000gn/T//RtmpBdWYJG/downloaded_packages
```

[10]:
```
library("moments")
skewness(trees$Girth)
skewness(trees$Height)
skewness(trees$Volume)
```

0.526316303781635

-0.374869013993998

1.06435747015449

Out of the three variables; Height is the variable that is closest to a perfectly normal distribution. Girth and Volume, also follow a normal distribution, but they are significantly positevly skewed. After plotting the histograms, and adjusting the binwidth length for better visualization; here are the skewness values exhibited by each variable: Girth is positevly skewed, height is slightly negatively skewed, and volume is positively skewed. The values agree with the visual histogram inspection of the Girth and Volume variables. Girth (.526) and volume (1.06) both have positively skewed values. thus, confirming the visual inspection. The mean is greater than the median for both variables. However, for Height the median = the mean (76), but the visual inspection and skewness calculation both show slight negative skewness (-0.374). This is due to the weighted values of 80 and 81; which are causing the slight negative skew.

3 q3

[18]:
```
url="https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/
  ↪auto-mpg.data"
```

[19]:
```
df <- read.csv(file=url, header=FALSE, sep="", as.is =4&9, col.names=␣
  ↪c("mpg","cylinders","displacement","horsepower","weight","acceleration",␣
  ↪"model year", "origin", "car name"))
str(df)
```

```
'data.frame':   398 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 …
 $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 …
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 …
 $ horsepower  : chr  "130.0" "165.0" "150.0" "150.0" …
 $ weight      : num  3504 3693 3436 3433 3449 …
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 …
 $ model.year  : int  70 70 70 70 70 70 70 70 70 70 …
 $ origin      : int  1 1 1 1 1 1 1 1 1 1 …
 $ car.name    : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth
satellite" "amc rebel sst" …
```

[20]:
```
df$horsepower = as.numeric(df$horsepower)
```

```
Warning message in eval(expr, envir, enclos):
"NAs introduced by coercion"
```

[21]:
```
Horse_base <- df$horsepower
Horse_NA_med <- df$horsepower
```

[22]:
```
Horse_NA_med[which(is.na(Horse_base))] <-median(Horse_base, na.rm =TRUE)
Horse_NA_med
```

1. 130 2. 165 3. 150 4. 150 5. 140 6. 198 7. 220 8. 215 9. 225 10. 190 11. 170 12. 160 13. 150 14. 225 15. 95 16. 95 17. 97 18. 85 19. 88 20. 46 21. 87 22. 90 23. 95 24. 113 25. 90 26. 215 27. 200 28. 210 29. 193 30. 88 31. 90 32. 95 33. 93.5 34. 100 35. 105 36. 100 37. 88 38. 100 39. 165 40. 175 41. 153

42. 150 43. 180 44. 170 45. 175 46. 110 47. 72 48. 100 49. 88 50. 86 51. 90 52. 70 53. 76 54. 65 55. 69 56. 60 57. 70 58. 95 59. 80 60. 54 61. 90 62. 86 63. 165 64. 175 65. 150 66. 153 67. 150 68. 208 69. 155 70. 160 71. 190 72. 97 73. 150 74. 130 75. 140 76. 150 77. 112 78. 76 79. 87 80. 69 81. 86 82. 92 83. 97 84. 80 85. 88 86. 175 87. 150 88. 145 89. 137 90. 150 91. 198 92. 150 93. 158 94. 150 95. 215 96. 225 97. 175 98. 105 99. 100 100. 100 101. 88 102. 95 103. 46 104. 150 105. 167 106. 170 107. 180 108. 100 109. 88 110. 72 111. 94 112. 90 113. 85 114. 107 115. 90 116. 145 117. 230 118. 49 119. 75 120. 91 121. 112 122. 150 123. 110 124. 122 125. 180 126. 95 127. 93.5 128. 100 129. 100 130. 67 131. 80 132. 65 133. 75 134. 100 135. 110 136. 105 137. 140 138. 150 139. 150 140. 140 141. 150 142. 83 143. 67 144. 78 145. 52 146. 61 147. 75 148. 75 149. 75 150. 97 151. 93 152. 67 153. 95 154. 105 155. 72 156. 72 157. 170 158. 145 159. 150 160. 148 161. 110 162. 105 163. 110 164. 95 165. 110 166. 110 167. 129 168. 75 169. 83 170. 100 171. 78 172. 96 173. 71 174. 97 175. 97 176. 70 177. 90 178. 95 179. 88 180. 98 181. 115 182. 53 183. 86 184. 81 185. 92 186. 79 187. 83 188. 140 189. 150 190. 120 191. 152 192. 100 193. 105 194. 81 195. 90 196. 52 197. 60 198. 70 199. 53 200. 100 201. 78 202. 110 203. 95 204. 71 205. 70 206. 75 207. 72 208. 102 209. 150 210. 88 211. 108 212. 120 213. 180 214. 145 215. 130 216. 150 217. 68 218. 80 219. 58 220. 96 221. 70 222. 145 223. 110 224. 145 225. 130 226. 110 227. 105 228. 100 229. 98 230. 180 231. 170 232. 190 233. 149 234. 78 235. 88 236. 75 237. 89 238. 63 239. 83 240. 67 241. 78 242. 97 243. 110 244. 110 245. 48 246. 66 247. 52 248. 70 249. 60 250. 110 251. 140 252. 139 253. 105 254. 95 255. 85 256. 88 257. 100 258. 90 259. 105 260. 85 261. 110 262. 120 263. 145 264. 165 265. 139 266. 140 267. 68 268. 95 269. 97 270. 75 271. 95 272. 105 273. 85 274. 97 275. 103 276. 125 277. 115 278. 133 279. 71 280. 68 281. 115 282. 85 283. 88 284. 90 285. 110 286. 130 287. 129 288. 138 289. 135 290. 155 291. 142 292. 125 293. 150 294. 71 295. 65 296. 80 297. 80 298. 77 299. 125 300. 71 301. 90 302. 70 303. 70 304. 65 305. 69 306. 90 307. 115 308. 115 309. 90 310. 76 311. 60 312. 70 313. 65 314. 90 315. 88 316. 90 317. 90 318. 78 319. 90 320. 75 321. 92 322. 75 323. 65 324. 105 325. 65 326. 48 327. 48 328. 67 329. 67 330. 67 331. 93.5 332. 67 333. 62 334. 132 335. 100 336. 88 337. 93.5 338. 72 339. 84 340. 84 341. 92 342. 110 343. 84 344. 58 345. 64 346. 60 347. 67 348. 65 349. 62 350. 68 351. 63 352. 65 353. 65 354. 74 355. 93.5 356. 75 357. 75 358. 100 359. 74 360. 80 361. 76 362. 116 363. 120 364. 110 365. 105 366. 88 367. 85 368. 88 369. 88 370. 88 371. 85 372. 84 373. 90 374. 92 375. 93.5 376. 74 377. 68 378. 68 379. 63 380. 70 381. 88 382. 75 383. 70 384. 67 385. 67 386. 67 387. 110 388. 85 389. 92 390. 112 391. 96 392. 84 393. 90 394. 86 395. 52 396. 84 397. 79 398. 82

The original mean (104.4694) that was calculated for Horsepower, ignored the 7 values in the column with a NA (previously a "?"). Therefore, the number of observations used to calculate the mean was 398 - 7 = 391. Thus, when the median (93.5) was used to fill in the 7 NA values; the mean was slightly lowered to 104.304. This is because the 7 newly inserted 93.5 values very slightly, positely skewed the data to the right. However, this very minor shift is almost negligable because there are 398 total observations.

[23]:
```
median(df$horsepower, na.rm=TRUE)
```

93.5

Original mean

[24]:
```
mean(df$horsepower, na.rm=TRUE)
```

104.469387755102

Mean when NA = median

[25]:
```
mean(Horse_NA_med)
```

104.304020100503

4 q4

[27]:
```
# install.packages("ISLR")
# install.packages("dplyr")
# install.packages("rlang")
# install.packages("tidyr")
# install.packages("tidyverse")
```

```
also installing the dependencies ‘bit’, ‘sass’, ‘rappdirs’, ‘rematch’, ‘bit64’,
‘prettyunits’, ‘bslib’, ‘jquerylib’, ‘tinytex’, ‘assertthat’, ‘blob’, ‘cli’,
‘DBI’, ‘data.table’, ‘gargle’, ‘cellranger’, ‘ids’, ‘clipr’, ‘vroom’, ‘tzdb’,
‘progress’, ‘rmarkdown’, ‘selectr’, ‘broom’, ‘dbplyr’, ‘dtplyr’, ‘forcats’,
‘googledrive’, ‘googlesheets4’, ‘haven’, ‘hms’, ‘lubridate’, ‘modelr’, ‘readr’,
‘readxl’, ‘reprex’, ‘rvest’
```

```
  There is a binary version available but the source version is later:
       binary source needs_compilation
dbplyr  2.1.1  2.2.0             FALSE
```

[3]:
```
library(MASS)
library (ISLR)
```

[4]:
```
library(ggplot2)
library(dplyr)
```

[7]:
```
library(tidyr)
data(Boston)
head(Boston)
```

A data.frame: 6 × 14

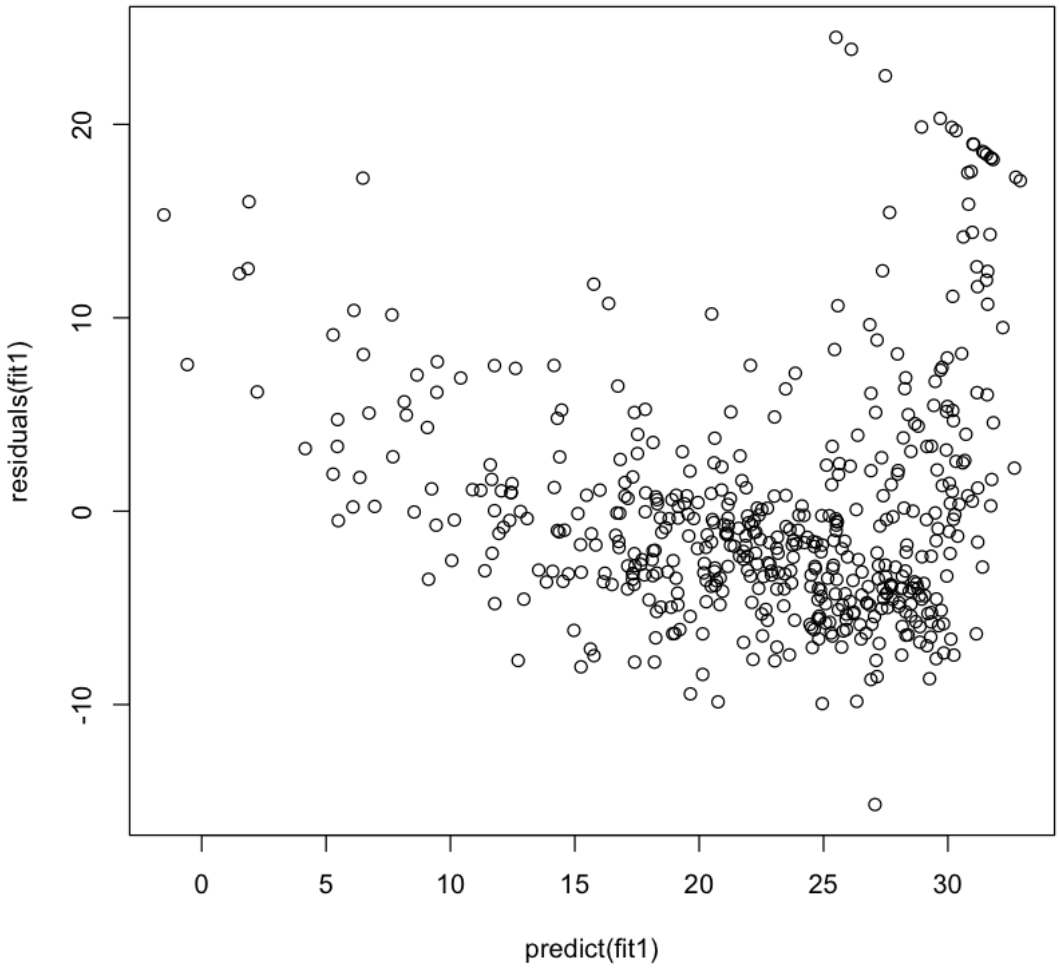
	crim <dbl>	zn <dbl>	indus <dbl>	chas <int>	nox <dbl>	rm <dbl>	age <dbl>	dis <dbl>	rad <int>	tax <db
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222
6	0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222

[8]:
```
fit1 =lm(medv~lstat ,data=Boston )
confint (fit1)
```

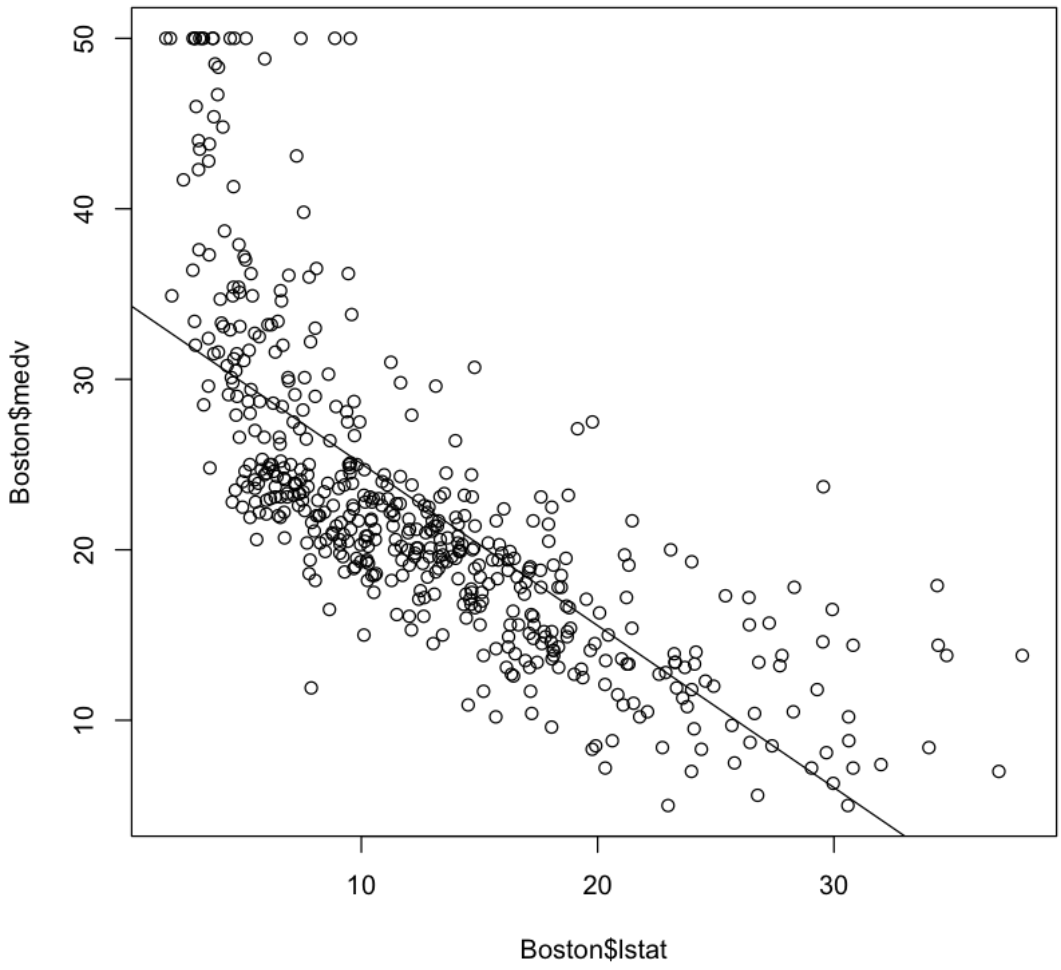
A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	33.448457	35.6592247
lstat	-1.026148	-0.8739505

[9]:
```
plot(predict (fit1), residuals (fit1))
```

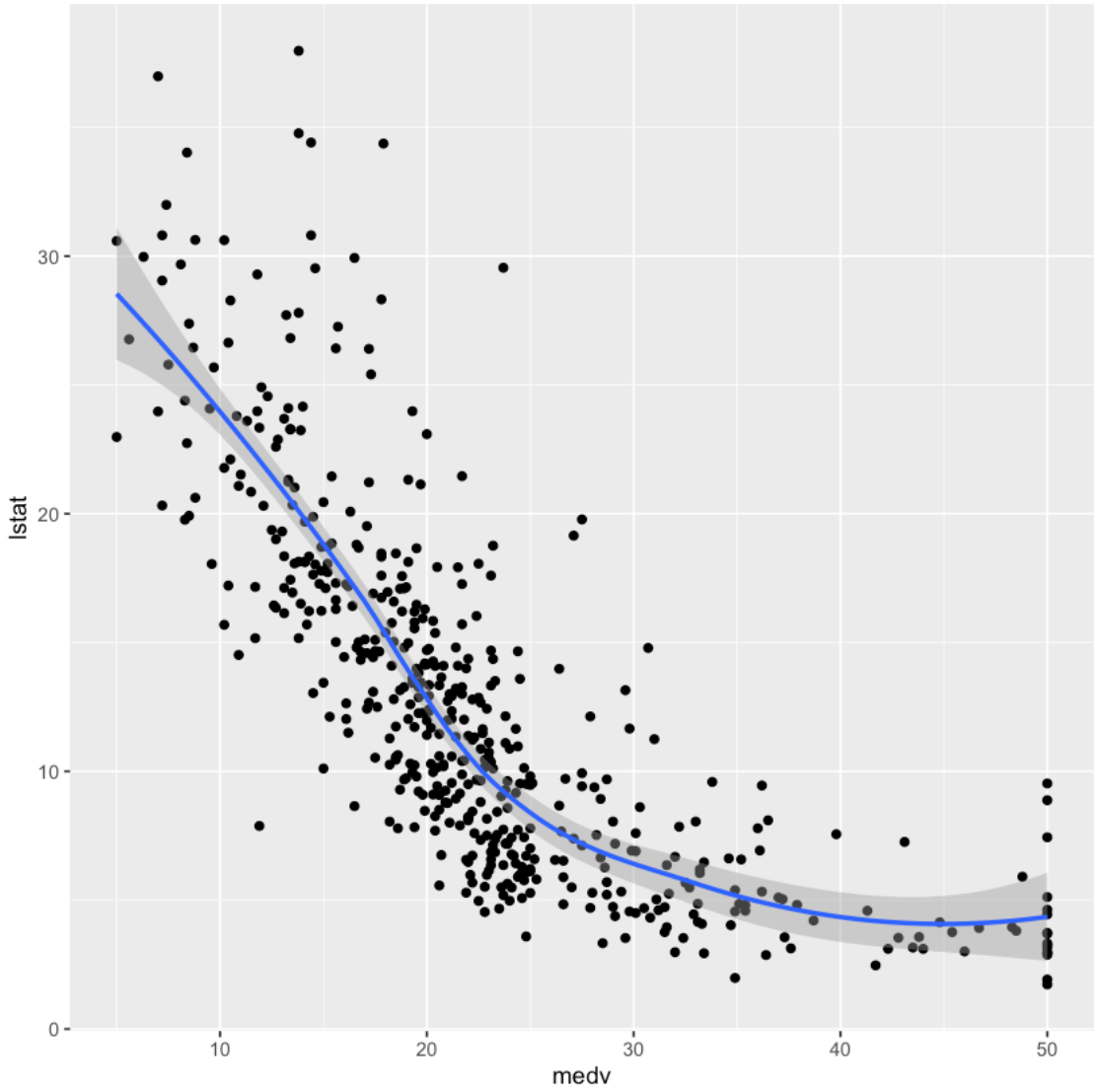


[10]:
```
plot(Boston$lstat ,Boston$medv)
abline (fit1)
```



[11]:
```
ggplot(Boston,aes(medv,lstat))+geom_point()+geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



[12]:
```
predict (fit1 ,data.frame(lstat=c(5 ,10 ,15) ), interval ="confidence")
```

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	29.80359	29.00741	30.59978
2	25.05335	24.47413	25.63256
3	20.30310	19.73159	20.87461

[13]:
```
predict (fit1 ,data.frame(lstat=c(5 ,10 ,15) ), interval ="prediction")
```

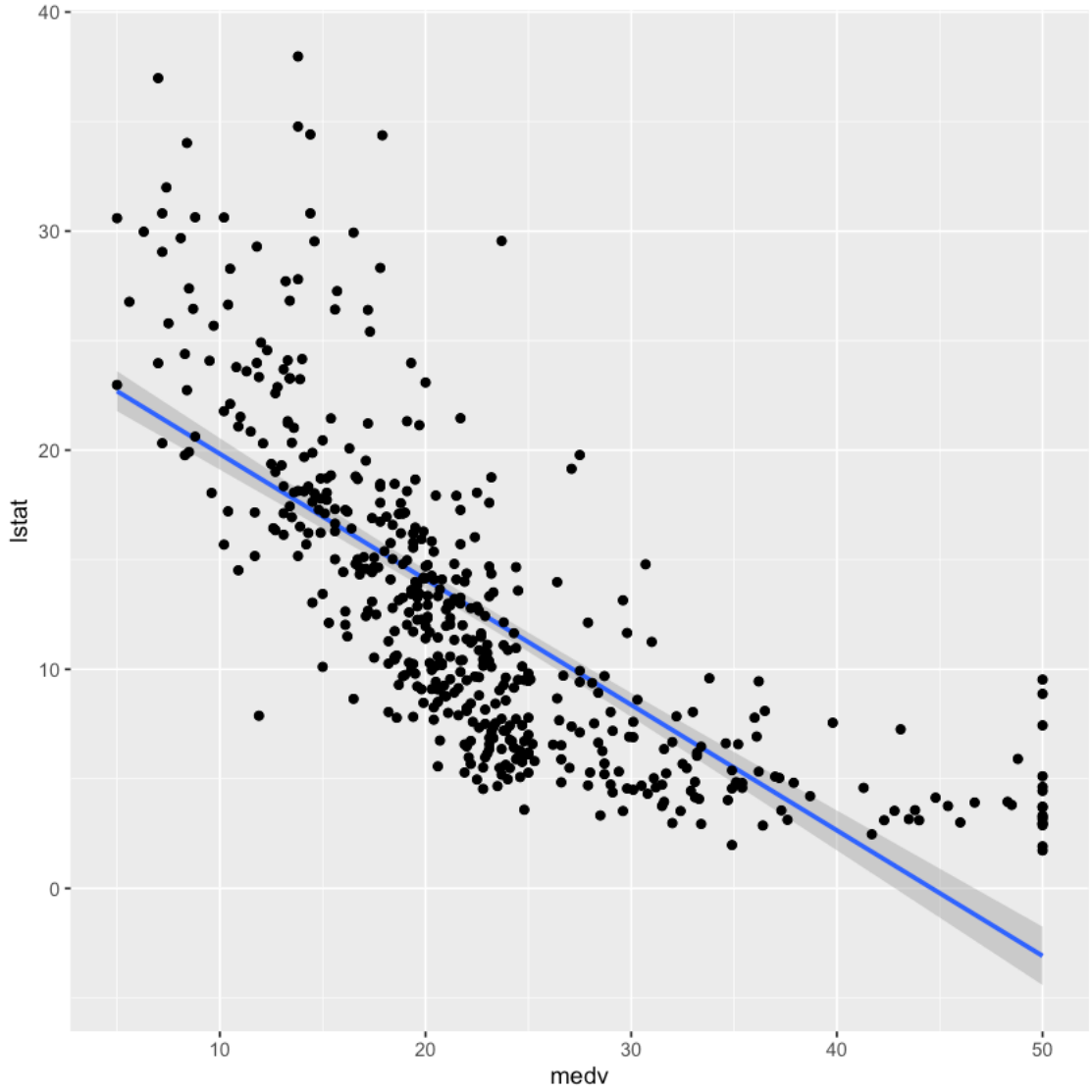
A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	29.80359	17.565675	42.04151
2	25.05335	12.827626	37.27907
3	20.30310	8.077742	32.52846

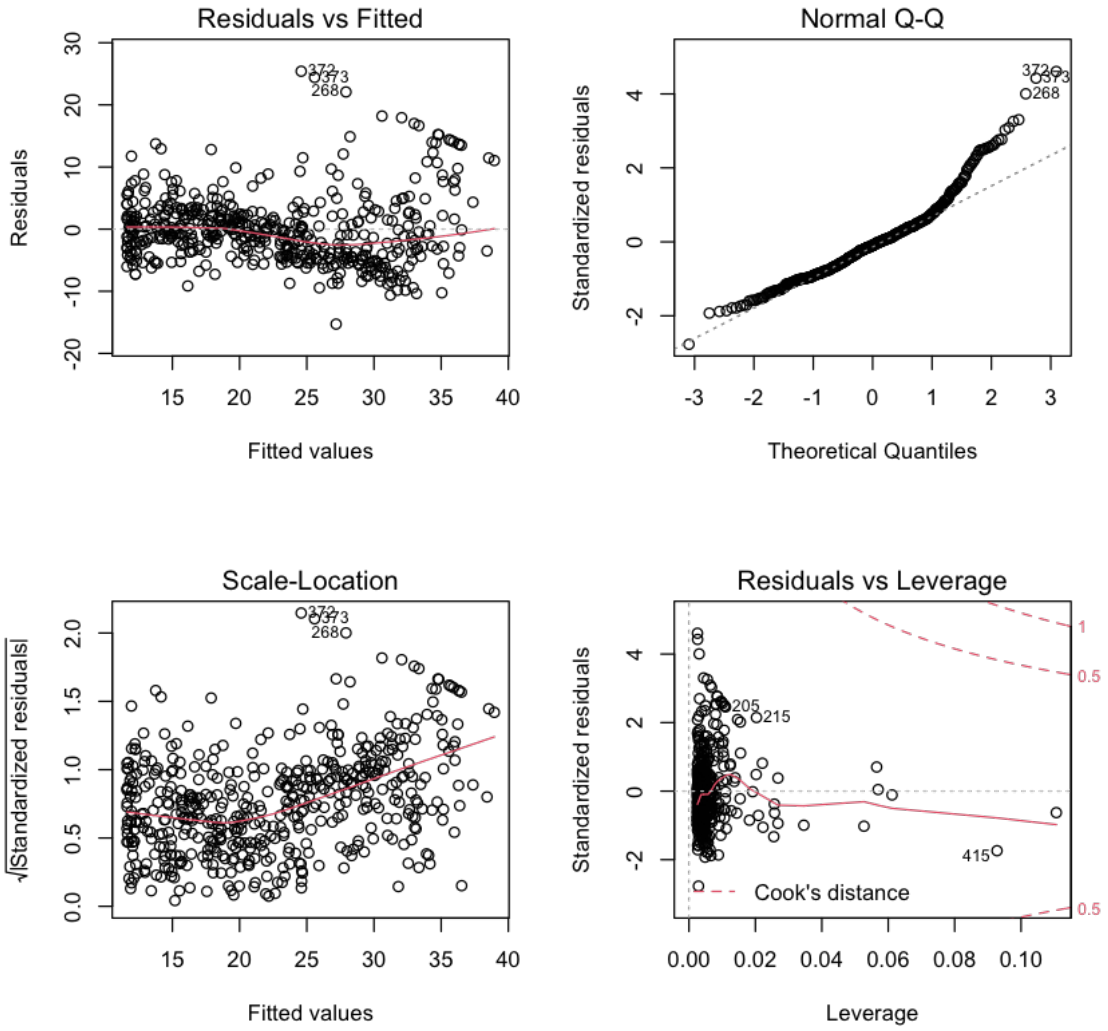
[15]:
```
fit2=lm(medv~lstat +I(lstat ^2),data=Boston)
ggplot(Boston,aes(medv,lstat,I(lstat␣
  ↪^2)))+geom_point()+geom_smooth(method="lm", se=TRUE)+geom_point() +␣
  ↪scale_color_discrete(guide=FALSE) + scale_fill_discrete(guide=FALSE)
```

```
`geom_smooth()` using formula 'y ~ x'

Warning message:
"It is deprecated to specify `guide = FALSE` to remove a guide. Please use
`guide = "none"` instead."
Warning message:
"It is deprecated to specify `guide = FALSE` to remove a guide. Please use
`guide = "none"` instead."
```



[16]:
```
par(mfrow=c(2,2))
plot(fit2)
```



No prediction based on confidence and prediction are not the same Model 1 has only one predictor where as Model 2 has two different predictors. The compare the models we should perform anova which states the null hypothesis as two models fit equally well where as the alternative hypothesis states that the model 2 is a better fit. When we look at Fstatistics and p values it is obvious that model2 fits way better. Since we found a non linearity before, it was a signal that the second model will fit better.