**CASESTUDY**

#set the working directory

setwd("C:/Users/manju/OneDrive/Desktop")

#read the data from csv file

df <- read.csv("corre.csv")

#display the contents of the data frame

df

print("summary() function provides a quick overview of the data,including statistics (min,1st quartile,median,mean,3rd quartile,max)")

summary(df)

#maximum value

max(df$Height.Inches.)

max(df$Weight.Pounds.)

#minimum value

min(df$Height.Inches.)

min(df$Weight.Pounds.)

#range

range(df$Height.Inches.)

range(df$Weight.Pounds.)

print("It calculates the mean (average) for the Height.Inches and Weight.Pounds columns, providing a measure of central tendency")

#mean for weight and Height

mean(df$Height.Inches)

mean(df$Weight.Pounds)

print(" The median() function calculates the median for the same columns, which represents the middle value when the data is sorted. It's another measure of central tendency.")

#median

median(df$Height.Inches)

median(df$Weight.Pounds)

print("he sd() function computes the standard deviation for the Height.Inches and Weight.Pounds columns. It measures the spread or dispersion of data points.")

#standard deviation

sd(df$Height.Inches)

sd(df$Weight.Pounds)

print("var(): This calculates the variance for the same columns, which is a measure of how much individual data points differ from the mean.")

#variance

var(df$Height.Inches.)

var(df$Weight.Pounds.)

print(" Quartiles help you understand data distribution and potential outliers.")

#quartiles

quantile(df$Height.Inches.)

quantile(df$Weight.Pounds.)

print("barplot() and hist(): These functions are used for data visualization. barplot() creates bar charts to visualize the frequency of values in the columns, and hist() creates histograms to visualize data distribution.")

#bar chat

barplot(table(df$Height.Inches.))

barplot(table(df$Weight.Pounds.))

#histogram

hist(df$Height.Inches.,main="Heights of persons",col = c("pink","yellow","blue"),xlab= "Heights",ylab = "No.of persons")

hist(df$Weight.Pounds.,main="Weights of persons",col = c("pink","yellow","blue"),xlab= "Heights",ylab = "No.of persons")

print("plot(): It's used to create scatter plots for the Height.Inches and Weight.Pounds columns to visualize relationships and distributions.")

#scatter plot

plot(df$Height.Inches.)

plot(df$Weight.Pounds.)

#plot with type

plot(df$Height.Inches.,df$Weight.Pounds.,'b')

print("scatter.smooth(): This function creates a scatter plot with a smooth curve to visualize trends in the data.")

#scatter plot with smooth curve

scatter.smooth(df$Height.Inches.,df$Weight.Pounds.)

print("library(scatterplot3d) and scatterplot3d(): These lines load the scatterplot3d library and create a 3D scatter plot to visualize data in three dimensions.")

library(scatterplot3d)

scatterplot3d(df$Height.Inches.,df$Weight.Pounds.,angle=120,color="pink")

print("boxplot(): It creates box plots for the Height.Inches and Weight.Pounds columns to visualize data distribution, variability, and potential outliers.")

#box plot

boxplot(df$Height.Inches.,df$Weight.Pounds.)

print("cor(): This computes the correlation coefficient between Height.Inches and Weight.Pounds to measure the strength and direction of their relationship.")

#correlation

corr <- cor(df$Height.Inches,df$Weight.Pounds)

corr

plot(df$Height.Inches,df$Weight.Pounds,main = "Weight vs Height",xlab = "Weight",ylab = "Height")

print("It helps us understand how changes in the independent variables are associated with changes in the dependent variable.")

print("In this code Height.Inches as the dependent variable and Weight.Pounds as the independent variable:")

#regression

r=lm(df$Height.Inches~df$Weight.Pounds)

r

print("summary(r): It provides a summary of the linear regression model's coefficients, R-squared value, and other relevant statistics.")

summary(r)

print("confint(r): Calculates confidence intervals for the coefficients of the regression model, which helps assess the precision of the parameter estimates.")

confint(r)

print("predict(): It's used to make predictions based on the linear regression model. In this case, it predicts Height.Inches based on Weight.Pounds for new data points.")

d = data.frame(v = c(112,171))

predict(r,newdata=d)

**OUTPUT:**

> setwd("C:/Users/manju/OneDrive/Desktop")

> df <- read.csv("corre.csv")

> df

index Height.Inches. Weight.Pounds.

1 1 65.78331 112.99250

2 2 71.51521 136.48730

3 3 69.39874 153.02690

4 4 68.21660 142.33540

5 5 67.78781 144.29710

6 6 68.69784 123.30240

7 7 69.80204 141.49470

8 8 70.01472 136.46230

9 9 67.90265 112.37230

10 10 66.78236 120.66720

11 11 66.48769 127.45160

12 12 67.62333 114.14300

13 13 68.30248 125.61070

14 14 67.11656 122.46180

15 15 68.27967 116.08660

16 16 71.09160 139.99750

17 17 66.46100 129.50230

18 18 68.64927 142.97330

19 19 71.23033 137.90250

20 20 67.13118 124.04490

21 21 67.83379 141.28070

22 22 68.87881 143.53920

23 23 63.48115 97.90191

24 24 68.42187 129.50270

25 25 67.62804 141.85010

26 26 67.20864 129.72440

27 27 70.84235 142.42350

28 28 67.49434 131.55020

29 29 66.53401 108.33240

30 30 65.44098 113.89220

31 31 69.52330 103.30160

32 32 65.81320 120.75360

33 33 67.81630 125.78860

34 34 70.59505 136.22250

35 35 71.80484 140.10150

36 36 69.20613 128.74870

37 37 66.80368 141.79940

38 38 67.65893 121.23190

39 39 67.80701 131.34780

40 40 64.04535 106.71150

41 41 68.57463 124.35980

42 42 65.18357 124.85910

43 43 69.65814 139.67110

44 44 67.96731 137.36960

45 45 65.98088 106.44990

46 46 68.67249 128.76390

47 47 66.88088 145.68370

48 48 67.69868 116.81900

49 49 69.82117 143.62150

50 50 69.08817 134.93250

51 51 69.91479 147.02190

52 52 67.33182 126.32850

53 53 70.26939 125.48390

54 54 69.10344 115.70840

55 55 65.38356 123.48920

56 56 70.18447 147.89260

57 57 70.40617 155.89870

58 58 66.54376 128.07420

59 59 66.36418 119.37010

60 60 67.53700 133.81480

61 61 66.50418 128.73250

62 62 68.99958 137.54530

63 63 68.30355 129.76040

64 64 67.01255 128.82400

65 65 70.80592 135.31650

66 66 68.21951 109.61130

67 67 69.05914 142.46840

68 68 67.73103 132.74900

69 69 67.21568 103.52750

70 70 67.36763 124.72990

71 71 65.27033 129.31370

72 72 70.84278 134.01750

73 73 69.92442 140.39690

74 74 64.28508 102.83510

75 75 68.24520 128.52140

76 76 66.35708 120.29910

77 77 68.36275 138.60360

78 78 65.47690 132.95740

79 79 69.71947 115.62330

80 80 67.72554 122.52400

81 81 68.63941 134.62540

82 82 66.78405 121.89860

83 83 70.05147 155.37670

84 84 66.27848 128.94180

85 85 69.20198 129.10130

86 86 69.13481 139.47330

87 87 67.36436 140.89010

88 88 70.09297 131.59160

89 89 70.17660 121.12320

90 90 68.22556 131.51270

91 91 68.12932 136.54790

92 92 70.24256 141.48960

93 93 71.48752 140.61040

94 94 69.20477 112.14130

95 95 70.06306 133.45700

96 96 70.55703 131.80010

97 97 66.28644 120.02850

98 98 63.42577 123.09720

99 99 66.76711 128.14320

100 100 68.88741 115.47590

101 101 64.87434 102.09270

102 102 67.09272 130.35300

103 103 68.34761 134.18420

104 104 65.61073 98.64133

105 105 67.75551 114.55990

106 106 68.02120 123.49170

107 107 67.66193 123.04800

108 108 66.31460 126.47720

109 109 69.43706 128.41700

110 110 63.83624 127.19410

111 111 67.72277 122.05620

112 112 70.05098 127.60640

113 113 70.18602 131.64230

114 114 65.94588 111.89550

115 115 70.00700 122.03900

116 116 68.61129 128.55470

117 117 68.80817 132.67920

118 118 69.76212 136.06320

119 119 65.45539 115.94030

120 120 68.82534 136.90410

121 121 65.80030 119.88040

122 122 67.21474 109.00550

123 123 69.42021 128.27050

124 124 68.94396 135.29130

125 125 67.94150 106.85580

126 126 65.62506 123.29390

127 127 66.49607 109.51430

128 128 67.92809 119.30870

129 129 68.89415 140.24020

130 130 70.24100 133.98410

131 131 68.26623 132.58070

132 132 71.23161 130.69880

133 133 69.09747 115.56370

134 134 64.39693 123.79410

135 135 71.09585 128.14270

136 136 68.21868 135.96460

137 137 65.91721 116.62730

138 138 67.43690 126.82410

139 139 73.90107 151.39130

140 140 69.98149 130.40220

141 141 69.51862 136.20680

142 142 65.18437 113.39890

143 143 68.00869 125.32870

144 144 68.33840 127.58460

145 145 65.18417 107.15640

146 146 68.26209 116.45880

147 147 68.56865 133.84020

148 148 64.49675 112.89010

149 149 68.71053 130.75680

150 150 68.89148 137.75710

151 151 69.54011 125.40360

152 152 67.39964 138.46590

153 153 66.47521 120.81840

154 154 66.01217 140.15390

155 155 72.44434 136.73970

156 156 64.12642 106.11390

157 157 70.98112 158.95620

158 158 67.50124 108.78680

159 159 72.01515 138.77580

160 160 65.31143 115.91360

161 161 67.07509 146.29220

162 162 64.39148 109.87650

163 163 69.37003 139.04990

164 164 68.37921 119.90010

165 165 65.31018 128.30690

166 166 67.13690 127.24280

167 167 68.39468 115.23060

168 168 66.29180 124.79750

169 169 67.18660 126.95110

170 170 65.99156 111.27110

171 171 69.43393 122.60890

172 172 67.97463 124.20840

173 173 67.76133 124.64530

174 174 65.27864 119.51690

175 175 73.83364 139.29830

176 176 66.81312 104.82650

177 177 66.89411 123.04240

178 178 65.73568 118.89230

179 179 65.98283 121.49390

180 180 66.58396 119.24880

181 181 67.11294 135.02390

182 182 65.87481 116.22800

183 183 66.78067 109.17310

184 184 68.73577 124.22370

185 185 66.22666 141.16450

186 186 65.95968 129.15010

187 187 68.58372 127.86930

188 188 66.59347 120.92440

189 189 66.96574 127.64660

190 190 68.08015 101.46930

191 191 70.19025 144.99270

192 192 65.52149 110.95230

193 193 67.45905 132.86250

194 194 67.40985 146.33850

195 195 69.66158 145.58940

196 196 65.79799 120.84310

197 197 66.10558 115.78130

198 198 68.23987 128.30190

199 199 68.02403 127.47180

200 200 71.39044 127.87610

201 201 65.73160 121.49970

202 202 66.43358 112.71480

203 203 70.01309 135.00200

204 204 69.48146 128.67890

205 205 68.62764 124.40620

206 206 68.36275 140.02600

207 207 68.39028 117.51900

208 208 68.77413 143.87370

209 209 69.92360 141.17030

210 210 71.55542 155.94140

211 211 68.44764 134.00930

212 212 66.71398 130.09750

213 213 66.68413 106.22650

214 214 67.93699 112.04890

215 215 68.89855 136.18840

216 216 67.29191 131.23600

217 217 69.57212 131.32310

218 218 67.67479 119.52610

219 219 69.04155 116.99650

220 220 67.96765 138.52550

221 221 65.83982 109.65180

222 222 65.77288 130.15690

223 223 71.14106 137.11140

224 224 67.83055 113.75900

225 225 65.06930 114.97250

226 226 69.70745 127.71490

227 227 69.92983 121.99720

228 228 66.11569 117.96070

229 229 68.61364 127.71410

230 230 68.99760 117.96190

231 231 66.79171 125.15540

232 232 68.02363 141.10260

233 233 69.67258 145.48220

234 234 71.82178 116.06500

235 235 72.74676 135.74580

236 236 67.27951 132.92480

237 237 67.41015 115.66220

238 238 68.53150 114.31840

239 239 68.47126 148.95200

240 240 68.51867 142.18780

241 241 63.72529 134.10400

242 242 67.70483 141.89260

243 243 69.47115 138.74440

244 244 66.70198 134.44900

245 245 65.23126 117.01670

246 246 69.89473 115.67520

247 247 69.83048 134.79050

248 248 65.39790 120.57460

249 249 68.32214 120.08350

250 250 65.93895 84.35980

251 251 70.09805 138.93940

252 252 66.05531 143.52450

253 253 68.23481 123.13470

254 254 65.21758 115.52610

255 255 69.16173 120.98440

256 256 67.60064 120.64730

257 257 67.28621 124.03520

258 258 66.84323 117.22380

259 259 68.08281 127.95980

260 260 66.56991 116.75270

261 261 70.16973 121.99570

262 262 67.85113 123.99520

263 263 66.62309 108.85500

264 264 63.68816 104.65740

265 265 68.03560 132.24800

266 266 66.94100 127.18340

267 267 66.26292 125.12090

268 268 70.73010 133.32200

269 269 70.70844 122.33640

270 270 73.26872 130.26360

271 271 63.79021 116.74310

272 272 67.84109 125.85340

273 273 67.78058 100.62700

274 274 67.23901 118.49450

275 275 66.75701 137.12550

276 276 69.03098 118.69910

277 277 68.52652 124.16410

278 278 64.65419 124.94420

279 279 65.14295 126.38890

280 280 66.74124 121.52550

281 281 66.99923 133.29510

282 282 70.26611 127.40720

283 283 67.57666 125.08000

284 284 66.30582 102.81140

285 285 68.90702 134.22120

286 286 65.60543 100.97580

287 287 68.00189 145.88860

288 288 69.93715 155.30460

289 289 70.14590 138.71750

290 290 66.66364 116.26110

291 291 69.40676 120.60870

292 292 66.80598 114.40830

293 293 67.70048 133.34060

294 294 69.13438 92.74955

295 295 67.53769 131.78250

296 296 65.77912 118.38650

297 297 66.25769 131.14660

298 298 67.39038 130.20330

299 299 64.94327 129.18360

300 300 69.25699 143.14450

301 301 69.07739 138.73070

302 302 69.64403 130.95280

303 303 69.07705 124.30320

304 304 68.01304 135.37020

305 305 68.85664 124.19880

306 306 63.19158 111.23890

307 307 65.57591 129.40370

308 308 69.12101 127.74440

309 309 69.78765 121.33110

310 310 67.48203 118.21830

311 311 69.08809 123.34410

312 312 68.55290 151.91180

313 313 68.21327 115.31080

314 314 67.10120 122.38440

315 315 71.23661 129.23230

316 316 69.11946 138.28240

317 317 65.49848 144.05650

318 318 66.58706 120.84820

319 319 67.99831 113.30340

320 320 72.57747 149.32270

321 321 67.97585 130.63730

322 322 66.62347 122.38700

323 323 67.29574 133.86770

324 324 65.95951 134.83360

325 325 66.54011 137.83630

326 326 68.07723 112.68120

327 327 65.83291 121.19640

328 328 69.23721 114.92620

329 329 64.69413 110.37620

330 330 69.21372 126.03370

331 331 67.16792 118.70080

332 332 67.84670 119.31080

333 333 69.27389 150.73970

[ reached 'max' / getOption("max.print") -- omitted 24667 rows ]

> summary(df)

index Height.Inches. Weight.Pounds.

Min. : 1 Min. :60.28 Min. : 78.01

1st Qu.: 6251 1st Qu.:66.70 1st Qu.:119.31

Median :12500 Median :68.00 Median :127.16

Mean :12500 Mean :67.99 Mean :127.08

3rd Qu.:18750 3rd Qu.:69.27 3rd Qu.:134.89

Max. :25000 Max. :75.15 Max. :170.92

> length(df)

[1] 3

> #maximum value

> max(df$Height.Inches.)

[1] 75.1528

> max(df$Weight.Pounds.)

[1] 170.924

> #minimum value

> min(df$Height.Inches.)

[1] 60.27836

> min(df$Weight.Pounds.)

[1] 78.01476

> #range

> range(df$Height.Inches.)

[1] 60.27836 75.15280

> range(df$Weight.Pounds.)

[1] 78.01476 170.92400

> #mean for weight and Height

> mean(df$Height.Inches)

[1] 67.99311

> mean(df$Weight.Pounds)

[1] 127.0794

> #median

> median(df$Height.Inches)

[1] 67.9957

> median(df$Weight.Pounds)

[1] 127.1577

> #standard deviation

> sd(df$Height.Inches)

[1] 1.901679

> sd(df$Weight.Pounds)

[1] 11.6609

> #variance

> var(df$Height.Inches.)

[1] 3.616382

> var(df$Weight.Pounds.)

[1] 135.9765

> #quartiles

> quantile(df$Height.Inches.)

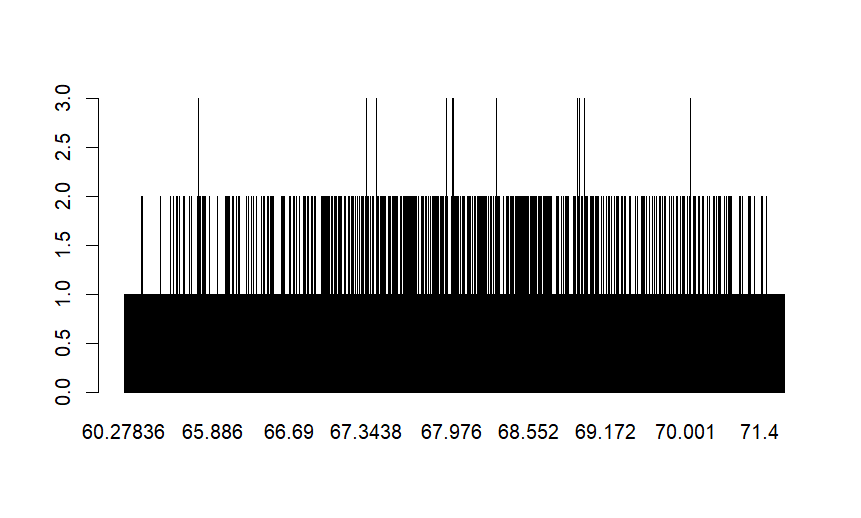
0% 25% 50% 75% 100%

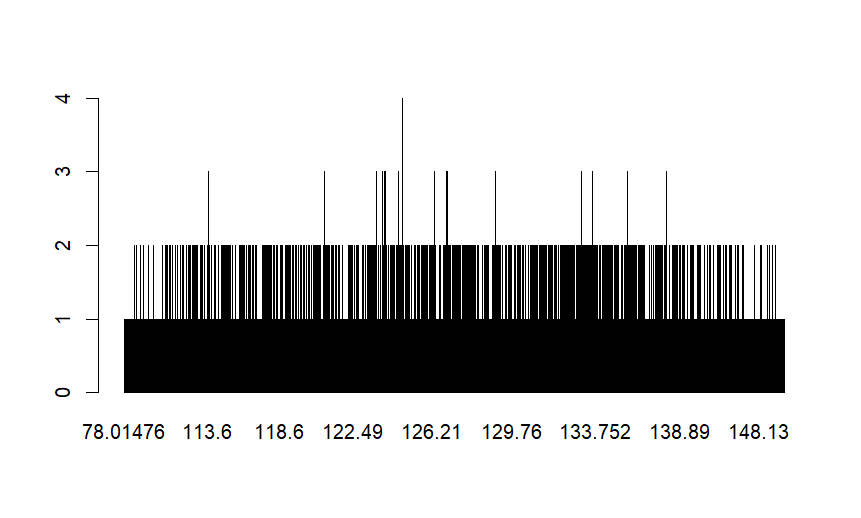
60.27836 66.70440 67.99570 69.27296 75.15280

> quantile(df$Weight.Pounds.)

0% 25% 50% 75% 100%

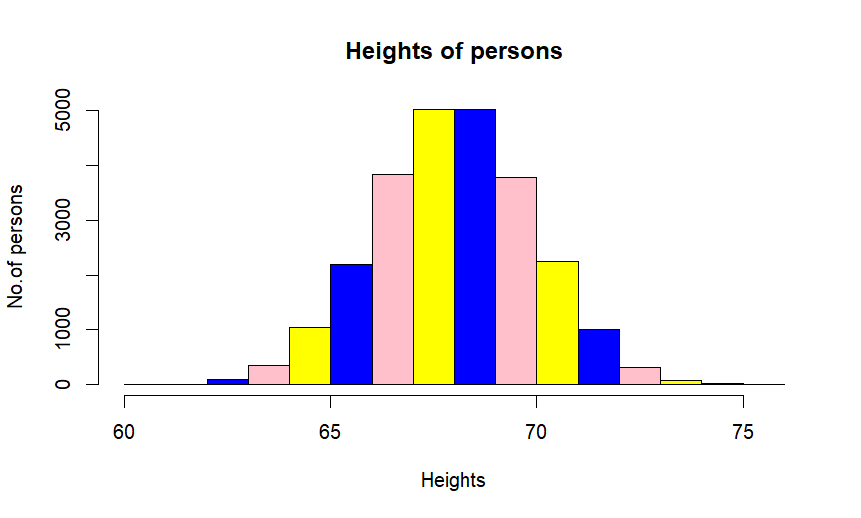
78.01476 119.30868 127.15775 134.89285 170.92400

> #bar chat

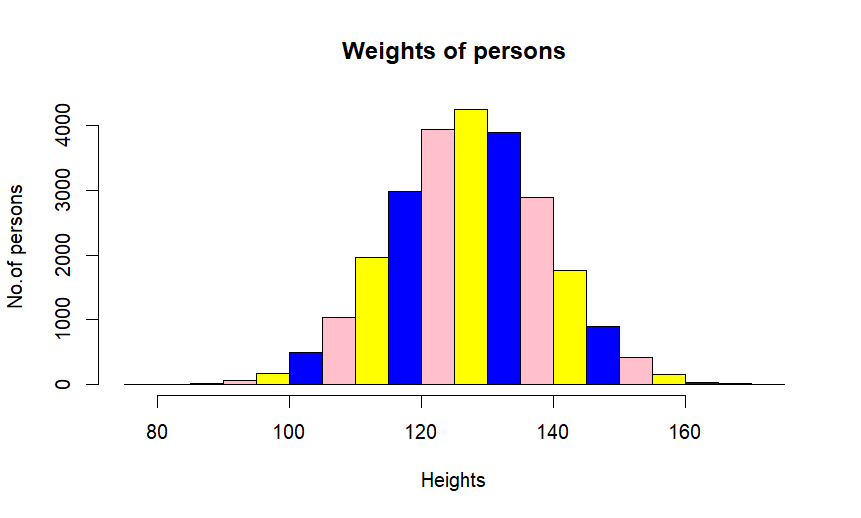
barplot(table(df$Weight.Pounds.))

> #histogram

> hist(df$Height.Inches.,main="Heights of persons",col = c("pink","yellow","blue"),xlab= "Heights",ylab = "No.of persons")

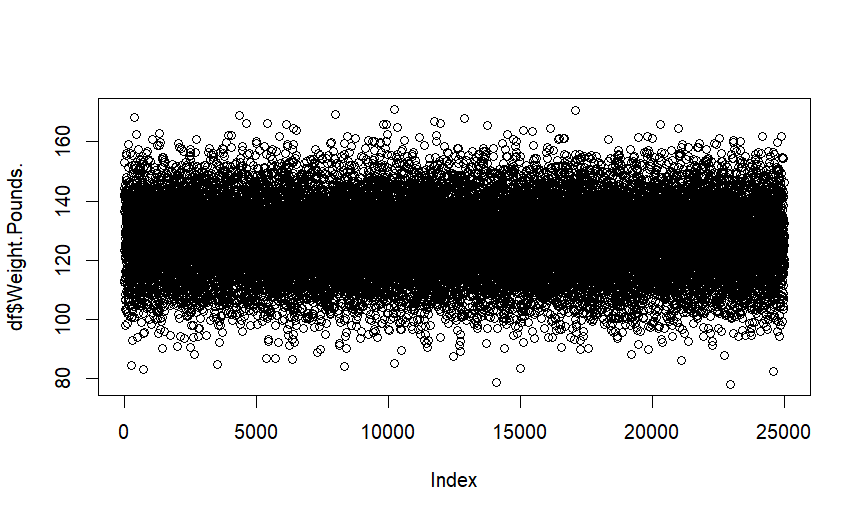


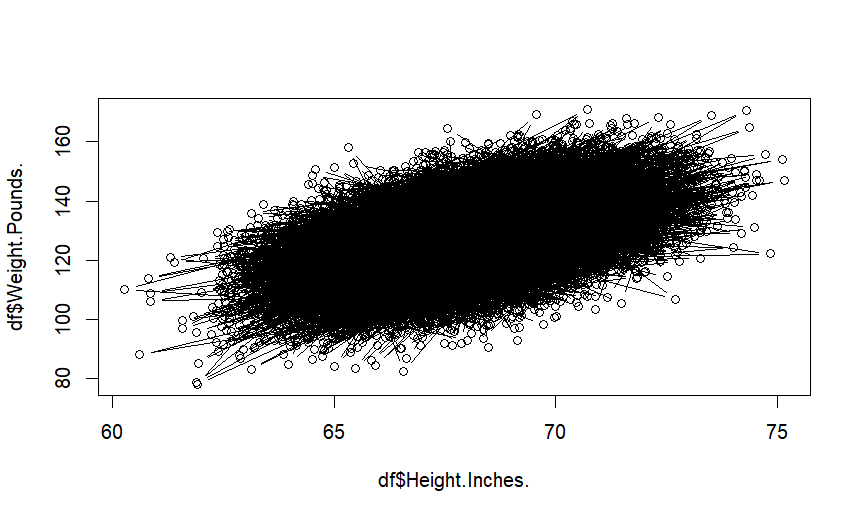
> hist(df$Weight.Pounds.,main="Weights of persons",col = c("pink","yellow","blue"),xlab= "Heights",ylab = "No.of persons")

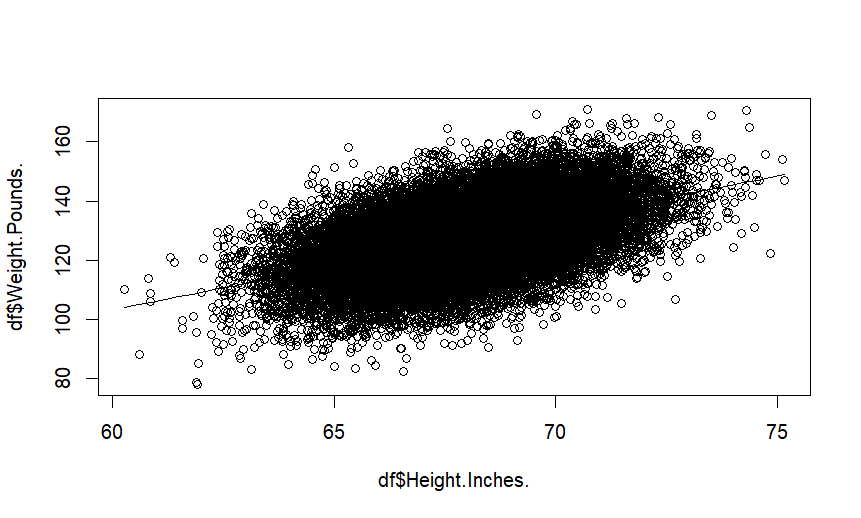


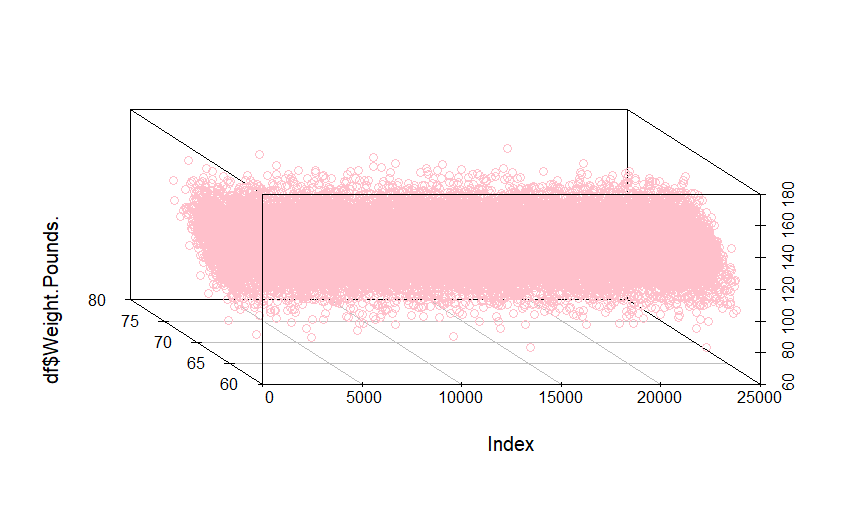
#scatter plot

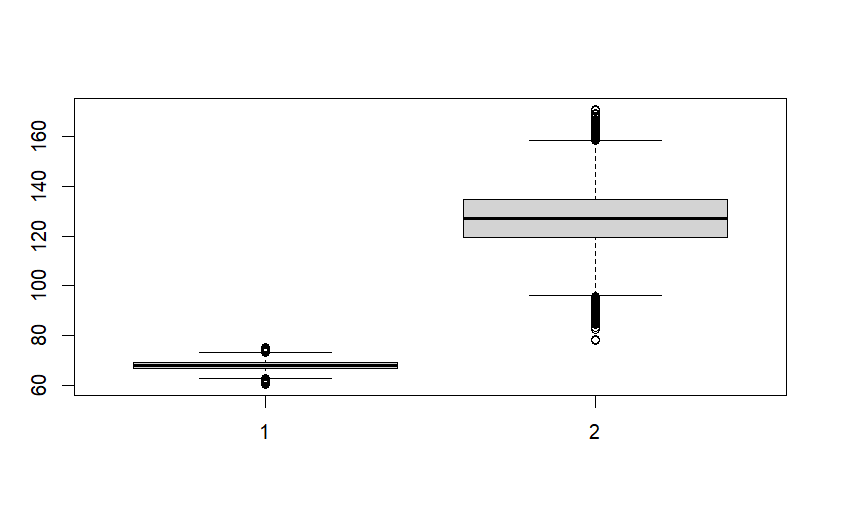
> plot(df$Height.Inches.)











> #correlation

> corr <- cor(df$Height.Inches,df$Weight.Pounds)

> corr

[1] 0.5028585

> #regression

> r=lm(df$Height.Inches~df$Weight.Pounds)

> r

Call:

lm(formula = df$Height.Inches ~ df$Weight.Pounds)

Coefficients:

(Intercept) df$Weight.Pounds

57.57171 0.08201

> summary(r)

Call:

lm(formula = df$Height.Inches ~ df$Weight.Pounds)

Residuals:

Min 1Q Median 3Q Max

-6.3235 -1.0935 0.0132 1.1049 7.2587

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5.757e+01 1.138e-01 506.01 <2e-16 \*\*\*

df$Weight.Pounds 8.201e-02 8.916e-04 91.98 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.644 on 24998 degrees of freedom

Multiple R-squared: 0.2529, Adjusted R-squared: 0.2528

F-statistic: 8461 on 1 and 24998 DF, p-value: < 2.2e-16

> confint(r)

2.5 % 97.5 %

(Intercept) 57.3487039 57.79471593

df$Weight.Pounds 0.0802595 0.08375453

> d = data.frame(v = c(112,171))

> predict(r,newdata=d)

1 2 3 4 5 6 7 8 9 10

66.83789 68.76463 70.12099 69.24421 69.40508 67.68337 69.17527 68.76258 66.78703 67.46727

11 12 13 14 15 16 17 18 19 20

68.02363 66.93224 67.87267 67.61444 67.09163 69.05249 68.19181 69.29652 68.88068 67.74426

21 22 23 24 25 26 27 28 29 30