HW01

Students: Vivek Sharma - Quynh Le

We've worked together in pair to do this homework.

Problem 1.

Write a function confBand(x, y, conf=0.95) taking in a predictor vector (x1; : : : ; xn) and a response vector y = (y1; : : : ; yn) and return a plot with the points (x1; y1); : : : ; (xn; yn), the least squares line, and the confidence band at level conf. Apply your function to hp and mpg from the 04cars dataset.

```
library("ggplot2")
```

```
confBand <- function(x, y, conf = 0.95 ) {
  data_f = data.frame(x = x, y = y)
  model <- lm(y ~ x, data = data_f)
  model_summary = summary(model)
  p = model_summary$df[1]
  n_minus_p  = model_summary$df[2]
  F_val <- qf(conf, p, n_minus_p)
  K   = sqrt(p*F_val)
  OUT  = predict(model, newdata = data_f, se.fit = TRUE, interval = "confidence", level = conf)
  y_h = OUT$fit[,1]
  lb = y_h - K*OUT$se.fit
  ub = y_h + K*OUT$se.fit
  final_df = data.frame(lower_bound = lb, mean = y_h, upper_bound = ub, y = y, x = x)
  beta_0 = model$coefficients[1]
  beta_1 = model$coefficients[2]
  pl = ggplot(data = final_df, aes(x = x, y = y)) +
    geom_point() +
    geom_ribbon(data = final_df, aes(ymin = lower_bound, ymax = upper_bound), alpha = 0.2, fill = 'red') +
    geom_abline(slope = beta_1, intercept = beta_0, col = 'red')
  return (pl)
}
```

```
library(ggplot2)
load("04cars.rda") # loads cars dataset"
tmp = dat[,c(13,15,16,18,19)] # extract selected variables
tmp = tmp[complete.cases(tmp),] # extracts complete cases
tmp = as.data.frame(tmp)
names(tmp) = c("hp","mpg","wt","len","wd") # abbreviate names

dat = tmp
hp = dat[,"hp"]
mpg = dat[, "mpg"]
data_f = data.frame(hp, mpg)
data_f
```

```
##      hp mpg
## 1   103  34
## 2   103  34
## 3   140  37
## 4   140  37
## 5   140  37
## 6   132  36
## 7   132  36
## 8   130  33
## 9   110  36
## 10  130  33
## 11  130  33
## 12  115  38
## 13  117  44
## 14  115  38
## 15  103  33
## 16  103  33
## 17  103  33
## 18  138  34
## 19  138  34
## 20  138  34
## 21  138  30
## 22  104  33
## 23  104  32
## 24  124  32
## 25  124  32
## 26  124  32
## 27  115  37
## 28  126  35
## 29  126  35
## 30  140  33
## 31  140  35
## 32  140  35
## 33  140  35
## 34  140  35
## 35  140  35
## 36  108  38
## 37  155  31
## 38  155  31
## 39  119  31
## 40  119  30
## 41  130  40
## 42  130  40
## 43  130  40
## 44  108  43
## 45  108  39
## 46  108  43
## 47  175  30
## 48  180  32
## 49  145  34
## 50  200  30
## 51  180  32
## 52  150  29
## 53  150  29
## 54  150  30
## 55  200  28
## 56  200  29
## 57  150  28
## 58  150  28
## 59  170  28
## 60  155  27
## 61  201  26
## 62  160  34
## 63  160  34
## 64  127  37
```

```
## 65  160  30
## 66   93  51
## 67   73  66
## 68  170  27
## 69  170  27
## 70  170  27
## 71  160  32
## 72  155  27
## 73  163  34
## 74  175  26
## 75  165  28
## 76  140  32
## 77  175  29
## 78  200  30
## 79  140  33
## 80  182  28
## 81  165  28
## 82  165  28
## 83  155  27
## 84  157  33
## 85  210  29
## 86  157  33
## 87  225  29
## 88  110  51
## 89  115  31
## 90  180  31
## 91  100  46
## 92  150  31
## 93  200  31
## 94  200  29
## 95  170  31
## 96  184  29
## 97  205  29
## 98  200  30
## 99  240  28
## 100 200  30
## 101 240  28
## 102 200  32
## 103 200  28
## 104 250  27
## 105 200  29
## 106 232  27
## 107 220  27
## 108 150  30
## 109 232  27
## 110 224  25
## 111 224  25
## 112 240  30
## 113 240  30
## 114 194  26
## 115 194  26
## 116 260  26
## 117 280  26
## 118 192  26
## 119 189  30
## 120 215  26
## 121 224  25
## 122 224  25
## 123 201  26
## 124 205  25
## 125 230  26
## 126 245  26
## 127 265  28
## 128 265  28
## 129 170  29
## 130 200  30
```

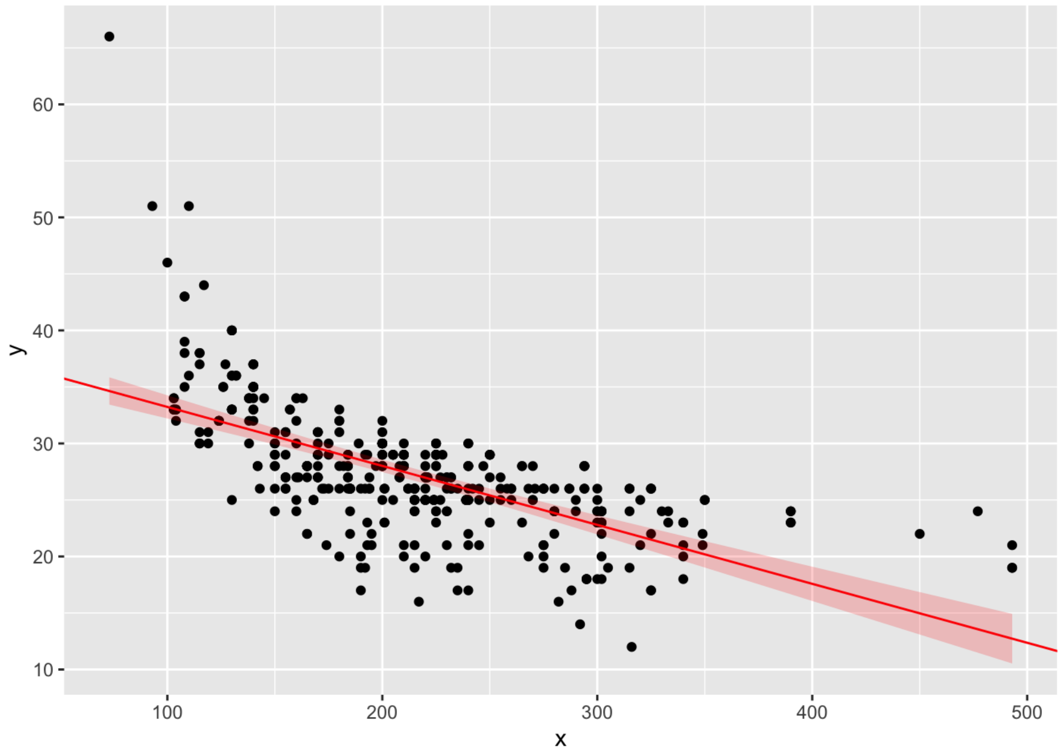
```
## 131 165  28
## 132 165  27
## 133 212  26
## 134 210  29
## 135 210  29
## 136 225  29
## 137 200  30
## 138 115  30
## 139 170  31
## 140 170  29
## 141 270  28
## 142 170  30
## 143 220  28
## 144 220  26
## 145 220  25
## 146 220  27
## 147 220  25
## 148 184  29
## 149 184  27
## 150 184  27
## 151 225  30
## 152 225  30
## 153 225  29
## 154 184  28
## 155 205  29
## 156 205  29
## 157 255  25
## 158 255  27
## 159 200  28
## 160 239  25
## 161 260  26
## 162 255  26
## 163 227  25
## 164 225  29
## 165 215  25
## 166 215  24
## 167 232  26
## 168 232  26
## 169 168  25
## 170 168  25
## 171 215  26
## 172 215  26
## 173 224  25
## 174 302  23
## 175 210  28
## 176 210  28
## 177 220  29
## 178 250  29
## 179 212  26
## 180 210  29
## 181 190  26
## 182 270  25
## 183 208  27
## 184 247  28
## 185 300  25
## 186 208  28
## 187 194  27
## 188 225  24
## 189 225  24
## 190 220  27
## 191 220  25
## 192 250  25
## 193 300  24
## 194 330  24
## 195 340  20
## 196 225  28
```

```
## 197 225  30
## 198 325  26
## 199 325  26
## 200 325  26
## 201 240  28
## 202 275  26
## 203 300  26
## 204 275  26
## 205 340  23
## 206 340  23
## 207 235  26
## 208 294  28
## 209 390  24
## 210 294  28
## 211 294  28
## 212 390  24
## 213 220  25
## 214 300  23
## 215 290  25
## 216 280  24
## 217 280  24
## 218 239  25
## 219 239  25
## 220 239  25
## 221 349  21
## 222 302  24
## 223 493  19
## 224 215  26
## 225 302  22
## 226 221  27
## 227 302  20
## 228 275  26
## 229 302  24
## 230 210  29
## 231 210  30
## 232 197  28
## 233 242  26
## 234 268  26
## 235 290  24
## 236 450  22
## 237 180  28
## 238 225  28
## 239 250  29
## 240 333  24
## 241 333  23
## 242 184  28
## 243 225  29
## 244 320  25
## 245 350  25
## 246 350  25
## 247 215  25
## 248 193  29
## 249 260  25
## 250 280  24
## 251 240  25
## 252 172  26
## 253 294  26
## 254 294  26
## 255 390  23
## 256 390  23
## 257 300  23
## 258 142  28
## 259 142  28
## 260 302  23
## 261 493  21
## 262 493  19
```

```
## 263 192  29
## 264 349  22
## 265 210  28
## 266 210  28
## 267 271  26
## 268 287  26
## 269 287  26
## 270 315  26
## 271 315  24
## 272 315  26
## 273 477  24
## 274 228  29
## 275 258  26
## 276 227  27
## 277 300  24
## 278 180  33
## 279 138  32
## 280 295  18
## 281 320  21
## 282 295  18
## 283 295  18
## 284 230  21
## 285 232  19
## 286 275  19
## 287 285  19
## 288 325  17
## 289 316  12
## 290 275  20
## 291 300  18
## 292 305  19
## 293 240  17
## 294 265  23
## 295 225  23
## 296 325  22
## 297 275  21
## 298 185  26
## 299 275  21
## 300 210  20
## 301 240  22
## 302 193  21
## 303 195  21
## 304 192  19
## 305 282  16
## 306 235  19
## 307 235  17
## 308 230  24
## 309 302  18
## 310 292  14
## 311 288  17
## 312 210  21
## 313 215  21
## 314 215  19
## 315 240  21
## 316 185  26
## 317 340  18
## 318 143  26
## 319 185  22
## 320 245  21
## 321 230  24
## 322 325  17
## 323 220  20
## 324 268  20
## 325 165  22
## 326 201  23
## 327 160  25
## 328 160  24
```

```
## 329 173  26
## 330 150  24
## 331 190  19
## 332 217  16
## 333 174  21
## 334 130  25
## 335 160  27
## 336 180  20
## 337 165  22
## 338 161  27
## 339 220  25
## 340 340  21
## 341 184  26
## 342 200  30
## 343 250  23
## 344 130  33
## 345 155  26
## 346 280  22
## 347 315  19
## 348 104  33
## 349 215  24
## 350 168  25
## 351 221  27
## 352 302  24
## 353 155  26
## 354 245  25
## 355 130  36
## 356 250  29
## 357 140  34
## 358 108  35
## 359 165  28
## 360 165  28
## 361 155  29
## 362 130  36
## 363 115  30
## 364 170  31
## 365 270  25
## 366 170  29
## 367 208  27
## 368 190  17
## 369 185  26
## 370 180  26
## 371 215  25
## 372 150  26
## 373 215  25
## 374 193  23
## 375 190  20
## 376 240  25
## 377 240  25
## 378 195  22
## 379 200  25
## 380 201  23
## 381 240  26
## 382 240  25
## 383 185  26
## 384 185  26
## 385 185  24
## 386 230  27
## 387 230  27
```

```
plt = confBand(x = data_f$hp, y = data_f$mpg, conf = 0.95)
plt
```



Problem 2.

Let n = 100 and draw x1; : : : ; xn iid Unif(0; 1), which stay fixed in what follows. Repeat the following experiment N = 1000 times. - Generate yi = 1 + xi + “i, with”i i.i.d. N(0; 0:2). - Compute the 99% confidence band and record whether it contains the true line, or not. Summarize the result of this numerical experiment by returning the proportion of times (out of N) that the confidence band contained the true line.

```r
# constants and initial params
n <-100
x <-  runif(n, min = 0, max = 1)
N <- 1000
mu <-  0; var <- 0.2; sd <- var**0.5
cnt <- 0
cnt2 <- 0
set.seed(42)


for(i in 1:N){
  e <-  rnorm(n, mu, sd) #generate epsilon ~ Normal(0, 0.2)
  y_true = 1 + x  #True Line
  y <-  1 + x + e
  lr <- lm(y ~ x)

  conf <-  confint(lr, level = 0.99)
  upper  <-  ((conf[2,2]*x) + conf[1,2])
  lower <-  ((conf[2,1]*x) + conf[1,1])

  # check whether the band contains true line
  temp <- data.frame(cbind(y_true, lower, upper))
  names(temp) <- c("y_true", "lower", "upper")
  outliers <- subset(temp, y_true < lower | y_true > upper)


  if (nrow(outliers) == 0){
    cnt <- cnt + 1
  }
  else {
    cnt2 <- cnt2 + 1
  }

}
# returning proportion of times (out of N) that the confidence band contained the true line
print(cnt/N)
```

```
## [1] 0.987
```