Example 1

Cesar Acosta - Qile Wang

April 2, 2019

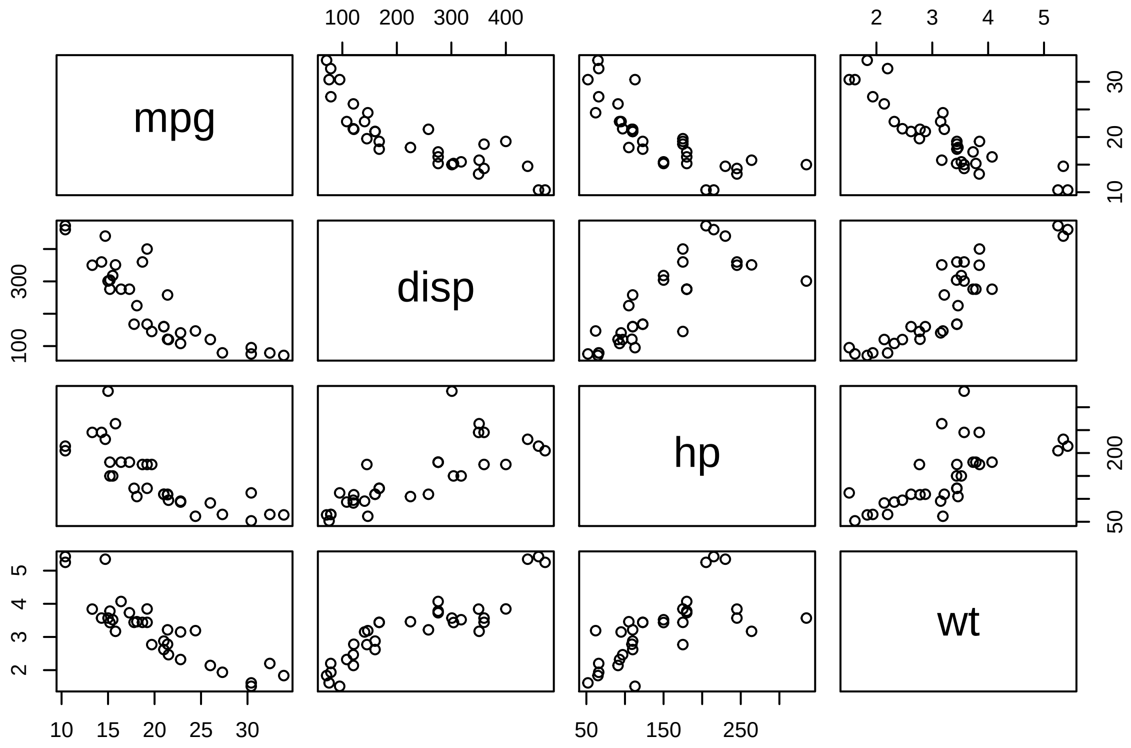
We compare one-response with multiple responses regression models using dataframe `mtcars`. It is of interest to predict mileage, displacement, horse-power, and, weight of cars, using the number of carburators, number of cylinders, and, if the car has automatic transmission as predictors.

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Verify that responses are correlated

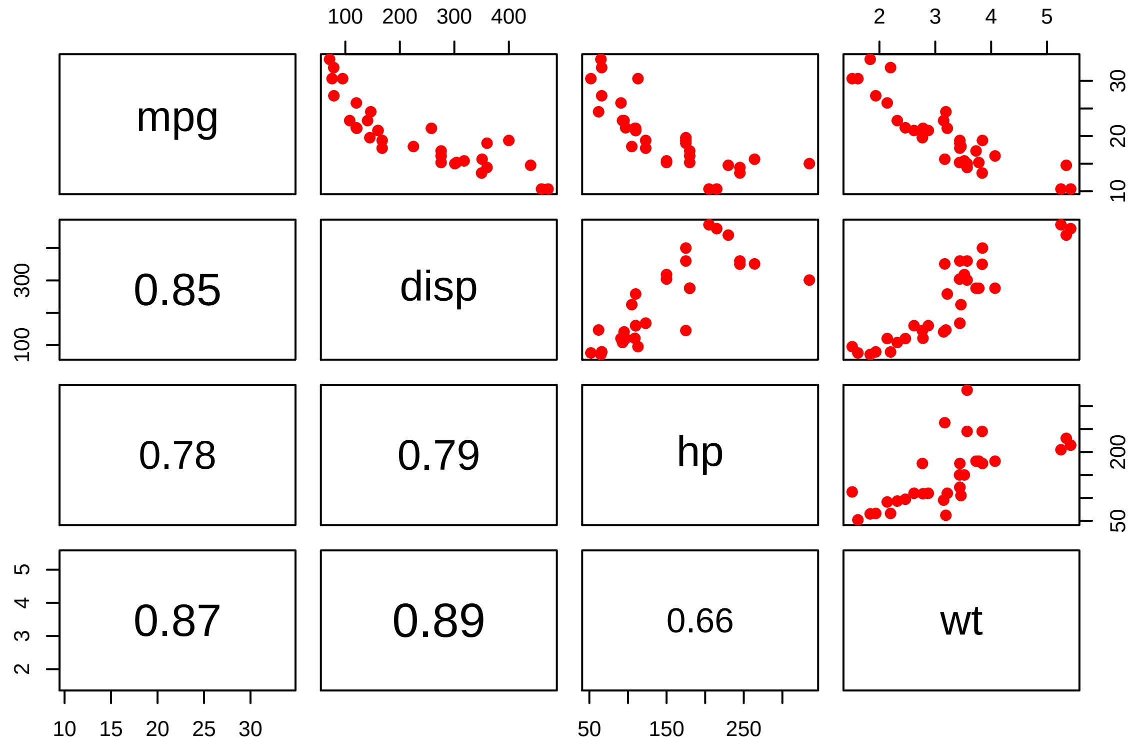
```
d1 = mtcars[,c("mpg","disp","hp","wt")]
pairs(d1)
```



A scatterplot with correlation coefficients can be obtained as follows

```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.44/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r, col=1)
}

pairs(d1, lower.panel = panel.cor,pch=19,col="red")
```



We change cyl to a factor

```
mtcars$cyl <- factor(mtcars$cyl)
```

One-response regression models

Fit regression models for comparison. Find prediction intervals from these models.

```r
m1 <- lm(mpg ~ cyl + am + carb, mtcars)
summary(m1)
```

```

Call:
lm(formula = mpg ~ cyl + am + carb, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.9074 -1.1723  0.2538  1.4851  5.4728 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  25.3203     1.2238  20.690  < 2e-16 ***
cyl6         -3.5494     1.7296  -2.052 0.049959 *  
cyl8         -6.9046     1.8078  -3.819 0.000712 ***
am            4.2268     1.3499   3.131 0.004156 ** 
carb         -1.1199     0.4354  -2.572 0.015923 *  
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 2.805 on 27 degrees of freedom
Multiple R-squared:  0.8113,    Adjusted R-squared:  0.7834 
F-statistic: 29.03 on 4 and 27 DF,  p-value: 1.991e-09
```

```r
m2 <- lm(disp ~ cyl + am + carb, mtcars)
summary(m2)
```

```

Call:
lm(formula = disp ~ cyl + am + carb, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-82.694 -21.442   0.254  26.500 111.779 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  134.325     21.836   6.152 1.42e-06 ***
cyl6          61.843     30.860   2.004   0.0552 .  
cyl8         218.991     32.256   6.789 2.72e-07 ***
am           -43.803     24.086  -1.819   0.0801 .  
carb           1.726      7.768   0.222   0.8258    
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 50.05 on 27 degrees of freedom
Multiple R-squared:  0.858, Adjusted R-squared:  0.8369 
F-statistic: 40.78 on 4 and 27 DF,  p-value: 4.537e-11
```

```
m3 <- lm(hp ~ cyl + am + carb, mtcars)
summary(m3)
```

```

Call:
lm(formula = hp ~ cyl + am + carb, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-41.520 -17.941  -4.378  19.799  41.292 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  46.5201    10.4825   4.438 0.000138 ***
cyl6          0.9116    14.8146   0.062 0.951386    
cyl8         87.5911    15.4851   5.656 5.25e-06 ***
am            4.4473    11.5629   0.385 0.703536    
carb         21.2765     3.7291   5.706 4.61e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.03 on 27 degrees of freedom
Multiple R-squared:  0.893, Adjusted R-squared:  0.8772 
F-statistic: 56.36 on 4 and 27 DF,  p-value: 1.023e-12
```

prediction intervals

```
newdata <- data.frame(cyl=factor(6,levels=c(4,6,8)),am=1,carb=4)
predict(m1, newdata, interval="prediction")
```

```
       fit      lwr      upr
1 21.51824 15.20583 27.83065
```

```
predict(m2, newdata, interval="prediction")
```

```
       fit      lwr     upr
1 159.2707 46.64042 271.901
```

```
predict(m3, newdata, interval="prediction")
```

```
      fit      lwr      upr
1 136.985 82.91553 191.0545
```

Multiple Response Linear Regression Model

Fit a multiple response regression model. Use function `coef()` to display the coefficients.

```
y <- as.matrix(mtcars[,c("mpg","disp","hp","wt")])
mv1 <- lm(y ~ cyl + am + carb, mtcars)
coef(mv1)
```

```
                  mpg      disp         hp         wt
(Intercept) 25.320303 134.32487 46.5201421  2.7612069
cyl6        -3.549419  61.84324  0.9116288  0.1957229
cyl8        -6.904637 218.99063 87.5910956  0.7723077
am           4.226774 -43.80256  4.4472569 -1.0254749
carb        -1.119855   1.72629 21.2764930  0.1749132
```

These coefficients are the same as those found by previous one-response models. The following summary is the same as the summaries found by one-response models.

```
summary(mv1)
```

```
Response mpg :

Call:
lm(formula = mpg ~ cyl + am + carb, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.9074 -1.1723  0.2538  1.4851  5.4728 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  25.3203     1.2238  20.690  < 2e-16 ***
cyl6         -3.5494     1.7296  -2.052 0.049959 *  
cyl8         -6.9046     1.8078  -3.819 0.000712 ***
am            4.2268     1.3499   3.131 0.004156 ** 
carb         -1.1199     0.4354  -2.572 0.015923 *  
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 2.805 on 27 degrees of freedom
Multiple R-squared:  0.8113,    Adjusted R-squared:  0.7834 
F-statistic: 29.03 on 4 and 27 DF,  p-value: 1.991e-09


Response disp :

Call:
lm(formula = disp ~ cyl + am + carb, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-82.694 -21.442   0.254  26.500 111.779 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  134.325     21.836   6.152 1.42e-06 ***
```

```
cyl6          61.843     30.860   2.004   0.0552 .
cyl8         218.991     32.256   6.789 2.72e-07 ***
am           -43.803     24.086  -1.819   0.0801 .
carb           1.726      7.768   0.222   0.8258
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 50.05 on 27 degrees of freedom
Multiple R-squared:  0.858, Adjusted R-squared:  0.8369
F-statistic: 40.78 on 4 and 27 DF,  p-value: 4.537e-11


Response hp :

Call:
lm(formula = hp ~ cyl + am + carb, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-41.520 -17.941  -4.378  19.799  41.292

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  46.5201    10.4825   4.438 0.000138 ***
cyl6          0.9116    14.8146   0.062 0.951386
cyl8         87.5911    15.4851   5.656 5.25e-06 ***
am            4.4473    11.5629   0.385 0.703536
carb         21.2765     3.7291   5.706 4.61e-06 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 24.03 on 27 degrees of freedom
Multiple R-squared:  0.893, Adjusted R-squared:  0.8772
F-statistic: 56.36 on 4 and 27 DF,  p-value: 1.023e-12


Response wt :

Call:
lm(formula = wt ~ cyl + am + carb, data = mtcars)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66317 -0.34384 -0.03802  0.12334  1.19083

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.76121    0.22133  12.476 1.01e-12 ***
cyl6         0.19572    0.31280   0.626  0.53675
cyl8         0.77231    0.32695   2.362  0.02564 *
am          -1.02547    0.24414  -4.200  0.00026 ***
carb         0.17491    0.07874   2.222  0.03489 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
Residual standard error: 0.5073 on 27 degrees of freedom
Multiple R-squared:  0.7659,    Adjusted R-squared:  0.7312
F-statistic: 22.08 on 4 and 27 DF,  p-value: 3.484e-08
```

Predictions with MRLR

```r
newdata <- data.frame(cyl=factor(6, levels=c(4,6,8)), am=1, carb=4)
newdata
```

```
  cyl am carb
1   6  1    4
```

The `predict()` function does not show intervals when used with a MRLR model!

```r
predict(mv1, newdata, interval="confidence")
```

```
       mpg     disp      hp       wt
1 21.51824 159.2707 136.985 2.631108
```

```r
predict(mv1, newdata, interval="prediction")
```

```
       mpg     disp      hp       wt
1 21.51824 159.2707 136.985 2.631108
```

To get intervals with a MRLR model use the function `predictmlm()`

```r
predictmlm <- function(object,newdata,level=0.95,interval = c("confidence", "prediction"))
{
  form <- as.formula(paste("~",as.character(formula(object))[3]))   # ~cyl + am + carb
  znew <- model.matrix(form, newdata)
  head(znew)

  fit <- predict(object, newdata)


  Y <- model.frame(object)[,1]    # responses dataframe
  Z <- model.matrix(object)       # matrix with predictors (categorical as binaries)
  n <- nrow(Y)
  m <- ncol(Y)                    # n of responses
  p <- ncol(Z) - 1                # n predictors (counting binaries instead of categoricals)

  # alpha correction
  alpha = 1 - level
  level = 1 - m*alpha
  sigmas = sigma(object)^2

  fit.var <- diag(znew %*% tcrossprod(solve(crossprod(Z)), znew))
  if(interval[1]=="prediction") fit.var <- fit.var + 1

  constant <- qf(level, df1=m, df2=n-p-m)*m*(n-p-1)/(n-p-m)
  vmat <- (n/(n-p-1)) * outer(fit.var, sigmas)

  # boundaries
  lwr <- fit - sqrt(constant) * sqrt(vmat)
  upr <- fit + sqrt(constant) * sqrt(vmat)

  if(nrow(znew)==1)
  {
    ci <- rbind(fit, lwr, upr)
    rownames(ci) <- c("fit", "lwr", "upr")
  }
  else
  {
    ci <- array(0, dim=c(nrow(znew), m, 3))
    dimnames(ci) <- list(1:nrow(znew), colnames(Y), c("fit", "lwr", "upr") )
    ci[,,1] <- fit
    ci[,,2] <- lwr
    ci[,,3] <- upr
  }
  ci
}
```

The function provides the prediction intervals

```
predictmlm(mv1, newdata, interval="prediction")
```

```
         mpg       disp        hp       wt
fit 21.51824 159.270705 136.98500 2.631108
lwr 12.45592  -2.425511  59.36086 0.992143
upr 30.58056 320.966921 214.60914 4.270072
```

These joint intervals are wider than those found with one-response regression models.

It can also be used for many intervals. In the following we use `predictmlm()` to predict the attributes of three cars

```r
newdata <- data.frame(cyl=factor(c(4,6,8), levels=c(4,6,8)), am=c(0,1,1), carb=c(2,4,6))
newdata
```

```
  cyl am carb
1   4  0    2
2   6  1    4
3   8  1    6
```

```r
predictmlm(mv1, newdata, interval="prediction")
```

```
, , fit

       mpg     disp        hp       wt
1 23.08059 137.7774  89.07313 3.111033
2 21.51824 159.2707 136.98500 2.631108
3 15.92331 319.8707 266.21745 3.557519

, , lwr

        mpg       disp        hp       wt
1 13.871941 -26.529667  10.19560 1.445604
2 12.455915  -2.425511  59.36086 0.992143
3  6.728061 155.802679 187.45471 1.894514

, , upr

       mpg     disp       hp       wt
1 32.28925 302.0846 167.9507 4.776462
2 30.58056 320.9669 214.6091 4.270072
3 25.11856 483.9387 344.9802 5.220524
```