Multivariate Linear Regression

Blake Pappas

December 17, 2023

Load the Motor Trend Car Road Tests Data

```
data(mtcars)
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
```

Data Manipulation

```
mtcars$cyl <- factor(mtcars$cyl)
vars <- c("mpg", "disp", "hp", "wt") # Response variables
Y <- as.matrix(mtcars[, vars])
```

Summarizing the Responses

```
colMeans(Y) # Calculates the means of the four response variables
```

```
##       mpg      disp        hp        wt
##  20.09062 230.72188 146.68750   3.21725
```

```
apply(Y, 2, sd) # Calculates the standard deviation of the four response variables
```

```
##         mpg        disp          hp          wt
##   6.0269481 123.9386938  68.5628685   0.9784574
```

```
cov(Y) # Calculates the covariance of the four response variables
```

```
##              mpg       disp         hp         wt
## mpg    36.324103  -633.0972 -320.73206  -5.116685
## disp -633.097208 15360.7998 6721.15867 107.684204
## hp   -320.732056  6721.1587 4700.86694  44.192661
## wt     -5.116685   107.6842   44.19266   0.957379
```

```
cor(Y) # Calculates the correlation matrix for the four response variables
```

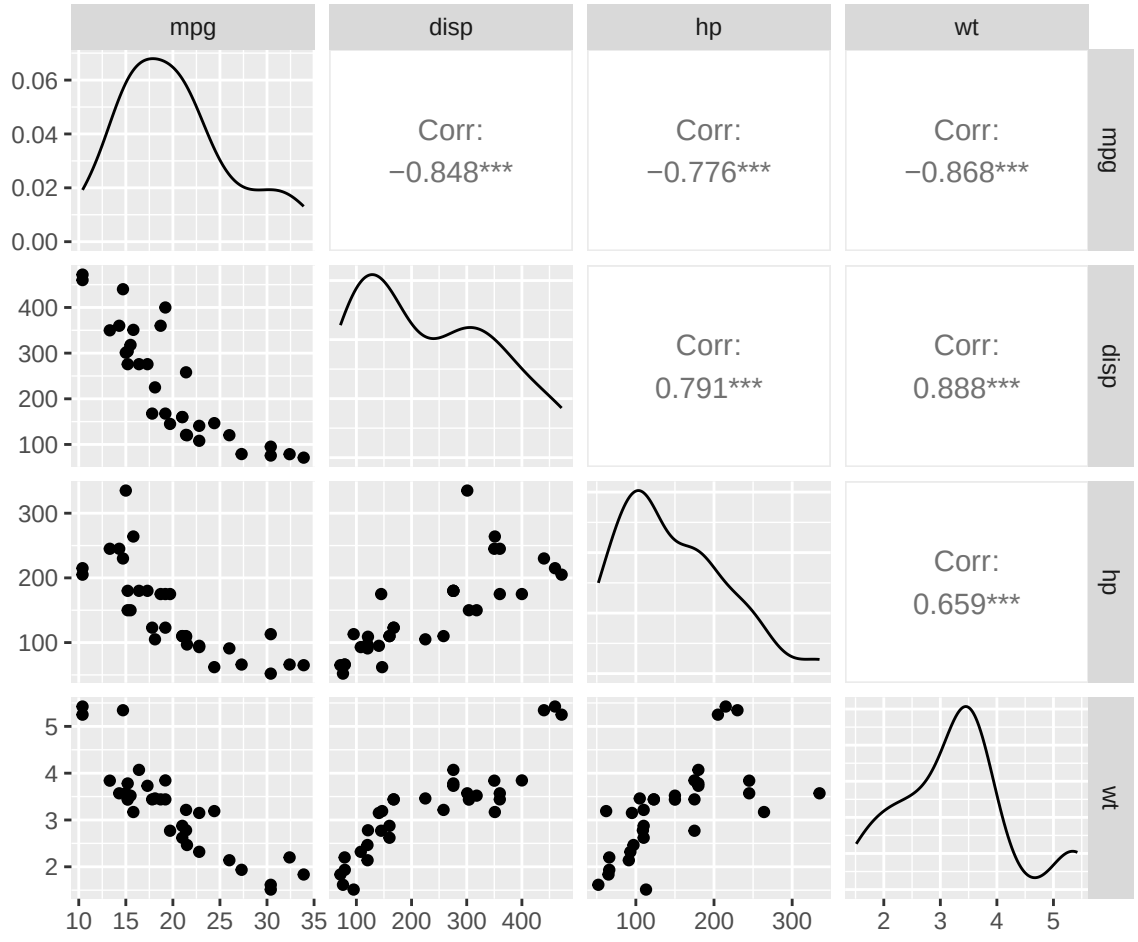
```
##              mpg       disp         hp         wt
## mpg    1.0000000 -0.8475514 -0.7761684 -0.8676594
## disp -0.8475514  1.0000000  0.7909486  0.8879799
## hp   -0.7761684  0.7909486  1.0000000  0.6587479
## wt   -0.8676594  0.8879799  0.6587479  1.0000000
```

```
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(as.data.frame(Y))
```



Fitting Linear Regression

```
mvlm <- lm(Y ~ cyl + am + carb, data = mtcars) # Creates a multiple linear regression model between the
summary(mvlm)
```

```
## 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
```

Sum Squared Cross Product (SSCP)

```r
ybar <- colMeans(Y)
n <- nrow(Y); d <- ncol(Y)
Ybar <- matrix(ybar, n, d, byrow = TRUE)
```

```r
SSCP.Tot <- crossprod(Y - Ybar)
SSCP.Reg <- crossprod(mvlm$fitted.values - Ybar)
SSCP.Err <- crossprod(Y - mvlm$fitted.values)
SSCP.Tot
```

```
##              mpg      disp         hp         wt
## mpg    1126.0472 -19626.01  -9942.694 -158.61723
## disp -19626.0134 476184.79 208355.919 3338.21032
## hp    -9942.6938 208355.92 145726.875 1369.97250
## wt     -158.6172   3338.21   1369.972   29.67875
```

```r
SSCP.Reg + SSCP.Err
```

```
##              mpg      disp         hp         wt
## mpg    1126.0472 -19626.01  -9942.694 -158.61723
## disp -19626.0134 476184.79 208355.919 3338.21033
## hp    -9942.6938 208355.92 145726.875 1369.97250
## wt     -158.6172   3338.21   1369.973   29.67875
```

Estimated Error Covariance Matrix in R

```r
p <- nrow(coef(mvlm)) - 1
SigmaHat <- SSCP.Err / (n - p - 1)
SigmaHat
```

```
##              mpg       disp          hp         wt
## mpg    7.8680094  -53.27166 -19.7015979 -0.6575443
## disp -53.2716607 2504.87095 425.1328988 18.1065416
## hp   -19.7015979  425.13290 577.2703337  0.4662491
## wt    -0.6575443   18.10654   0.4662491  0.2573503
```

Testing If We Need "cyl"

```r
# Tests Whether the Reduced Model Works Better Than the Full Model
mvlm0 <- lm(Y ~ am + carb, data = mtcars)
anova(mvlm, mvlm0, test = "Wilks")
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ cyl + am + carb
## Model 2: Y ~ am + carb
##   Res.Df Df Gen.var.   Wilks approx F num Df den Df    Pr(>F)
## 1     27      29.862
## 2     29  2   43.692 0.16395   8.8181      8     48 2.525e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(mvlm, mvlm0, test = "Pillai")
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ cyl + am + carb
## Model 2: Y ~ am + carb
##   Res.Df Df Gen.var. Pillai approx F num Df den Df    Pr(>F)
## 1     27      29.862
## 2     29  2   43.692 1.0323   6.6672      8     50 6.593e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Confidence and Prediction Intervals

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

# Confidence Interval
predict(mvlm, newdata, interval = "confidence")
```

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

```r
# Prediction Interval
predict(mvlm, newdata, interval = "prediction")
```

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

R does not yet have the capability to produce CIs/PIs with multivariate responses.

Below is the R function used to calculate the multivariate regression CIs and PIs:

```r
pred.mlm <- function(object, newdata, level = 0.95,
                     interval = c("confidence", "prediction")) {
  form <- as.formula(paste("~", as.character(formula(object))[3]))
  xnew <- model.matrix(form, newdata)
  fit <- predict(object, newdata)
  Y <- model.frame(object)[, 1]; X <- model.matrix(object)
  n <- nrow(Y); d <- ncol(Y); p <- ncol(X) - 1
  sigmas <- colSums((Y - object$fitted.values)^2) / (n - p - 1)
  fit.var <- diag(xnew %*% tcrossprod(solve(crossprod(X)), xnew))
  if(interval[1] == "prediction") fit.var <- fit.var + 1
  const <- qf(level, df1 = d, df2 = n - p - d) * d * (n - p - 1) / (n - p - d)
  vmat <- (n / (n - p - 1)) * outer(fit.var, sigmas)
  lwr <- fit - sqrt(const) * sqrt(vmat)
  upr <- fit + sqrt(const) * sqrt(vmat)
  if(nrow(xnew) == 1L) {
    ci <- rbind(fit, lwr, upr)
    rownames(ci) <- c("fit", "lwr", "upr")
```

```
  } else {
    ci <- array(0, dim = c(nrow(xnew), d, 3))
    dimnames(ci) <- list(1:nrow(xnew), colnames(Y), c("fit", "lwr", "upr"))
    ci[, , 1] <- fit; ci[, , 2] <- lwr; ci[, , 3] <- upr
  }
  ci
}

# Confidence Interval
pred.mlm(mvlm, newdata)
```

```
##          mpg     disp        hp       wt
## fit 21.51824 159.2707 136.98500 2.631108
## lwr 16.65593  72.5141  95.33649 1.751736
## upr 26.38055 246.0273 178.63351 3.510479
```

```
# Prediction Interval
pred.mlm(mvlm, newdata, interval = "prediction")
```

```
##           mpg      disp        hp        wt
## fit 21.518240 159.27070 136.98500 2.6311076
## lwr  9.680053 -51.95435  35.58397 0.4901152
## upr 33.356426 370.49576 238.38603 4.7720999
```