MTH 4230 R Notes 10

1 Model Diagnostics

- Once a model has been fitted by `lm()`, the functions below will compute various diagnostic measures for identifying ***outliers*** and ***influential points*** and for assessing the severity of ***multicollinearity***.

```
rstandard()            # Studentized residuals
rstudent()             # Delete-one studentized residuals
influence.measures()   # All of hatvalues(), cooks.distance(),
                       # dffits(), dfbetas(), and covratio()
hatvalues()            # Leverage values (diagonal elements of the
                       # hat matrix)
cooks.distance()       # Cooke's distances
dffits()               # DFFITS values
dfbetas()              # DFBETAS values
covratio()             # The covariance ratio statistic
```

- Each of these functions accepts an *lm* object as its main argument. Here's more precisely what they return:

`rstandard()`	Residuals $\boldsymbol{Y_i - \hat{Y}_i}$ standardized by $\sqrt{\mathbf{MSE} \cdot \mathbf{(1 - h_{ii})}}$.
`rstudent()`	Delete-one residuals $\boldsymbol{Y_i - \hat{Y}_{i(i)}}$ standardized by $\sqrt{\mathbf{MSE_{(i)}} \cdot \mathbf{(1 - h_{ii})}}$.
`influence.measures()`	All of `hatvalues()`, `cooks.distance()`, `dffits()`, `dfbetas()`, and `covratio()`.
`hatvalues()`	Diagonal elements $\boldsymbol{h_{ii}}$ of the hat matrix $\boldsymbol{H = X(X^TX)^{-1}X^T}$.
`cooks.distance()`	Cooke's distances.
`dffits()`	DFFITS values.
`dfbetas()`	DFBETAS values.
`covratio()`	The ***covariance ratio statistic***, $\frac{\lvert\mathbf{MSE}_{(i)} \cdot (\boldsymbol{X}_{(i)}^T \boldsymbol{X}_{(i)})^{-1}\rvert}{\lvert\mathbf{MSE} \cdot (\boldsymbol{X}^T\boldsymbol{X})^{-1}\rvert}$, where $\boldsymbol{X_{(i)}}$ is the design matrix with ith observation removed and $\lvert \cdot \rvert$ means the **determinant**. This statistic measures the influence of the ith observation on the variance-covariance matrix of the parameter estimators. Large values indicate influential observations.

- Suppose, for example, we have the following *data frame*:

```
my.data

##    response x1  x2 x3
## 1        21 10 2.4 94
## 2        22 11 1.9 75
## 3        24 13 2.6 66
## 4        20 12 1.8 77
## 5        26 13 1.3 75
## 6        26 16 0.9 81
## 7        25 15 1.2 72
## 8        27 20 1.0 55
## 9        31 18 1.3 44
## 10       34 19 1.0 69
## 11       29 19 0.8 73
## 12       33 22 0.7 90
```

and we fit a regression model to the data:

```
my.reg <- lm(response ~ x1 + x2 + x3, data=my.data)
```

Here, we pass `my.reg` to `influence.measures()`:

```
influence.measures(my.reg)

## Influence measures of
##   lm(formula = response ~ x1 + x2 + x3, data = my.data) :
##
##     dfb.1_  dfb.x1  dfb.x2   dfb.x3   dffit cov.r   cook.d   hat inf
## 1  -0.0780  0.0219  0.0843  0.11714  0.1958 3.293 0.010892 0.493   *
## 2   0.0292 -0.0354 -0.0141 -0.01001  0.0565 2.202 0.000912 0.230
## 3  -0.0414  0.0458  0.0749 -0.00552  0.0879 4.570 0.002208 0.628   *
## 4  -0.2088  0.2832  0.1009  0.00739 -0.5531 0.941 0.071926 0.169
## 5   0.5107 -0.5488 -0.4958 -0.12987  0.6628 1.330 0.108439 0.285
## 6  -0.0760  0.0909  0.1313 -0.04215 -0.1862 2.141 0.009763 0.249
## 7  -0.0956  0.0902  0.0948  0.03313 -0.1422 1.901 0.005685 0.156
## 8  -0.0502 -0.5054 -0.1817  0.77652 -1.5070 0.232 0.361626 0.290
## 9   0.3602  0.0171  0.0324 -0.85912  1.0245 1.820 0.258432 0.483
## 10 -0.1658  0.3567  0.0574  0.01008  0.8767 0.289 0.135315 0.150
## 11 -0.0022 -0.0189  0.0485 -0.02269 -0.1614 1.923 0.007315 0.173
## 12 -0.1527  0.1439  0.0633  0.15508  0.2148 5.498 0.013147 0.693   *
```

Each **row** in the output corresponds to a multivariate **observation** in the **data set**, and asterisks indicate ***influential*** observations. Thus above, the 1st, 3rd, and 12th observations are ***influential***.

- `influence.measures()` returns a *list* of objects. By typing:

```
my.inf <- influence.measures(my.reg)
```

we save the *list* as `my.inf`, and we can look at the names of the objects in the *list* by typing:

```
names(my.inf)

## [1] "infmat" "is.inf" "call"
```

The object `"infmat"` is the *numeric matrix* shown in the output above, whose rows correspond to observations in the data set and whose columns correspond to measures of influence.

The object `"is.inf"` is a *logical matrix* whose whose i, jth element is `TRUE` or `FALSE` depending on whether the ith observation is influential according to the jth influence measure. This is useful for determining by which measure an observation was deemed influential.

To look at `"is.inf"`, we use the dollar sign operator:

```
my.inf$is.inf

##    dfb.1_ dfb.x1 dfb.x2 dfb.x3 dffit cov.r cook.d   hat
## 1   FALSE  FALSE  FALSE  FALSE FALSE  TRUE  FALSE FALSE
## 2   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 3   FALSE  FALSE  FALSE  FALSE FALSE  TRUE  FALSE FALSE
## 4   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 5   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 6   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 7   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 8   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 9   FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 10  FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 11  FALSE  FALSE  FALSE  FALSE FALSE FALSE  FALSE FALSE
## 12  FALSE  FALSE  FALSE  FALSE FALSE  TRUE  FALSE FALSE
```

From the output,

- The 1st observation was deemed influential because of it's ***covariance ratio*** value.
- The 3rd observation was also deemed influential because of it's ***covariance ratio*** value.
- The 12th observation was also deemed influential because of it's ***covariance ratio*** value.

1.1 Variance Inflation Factors

- The function `vif.lm()` below (written by Bill Venables) will compute the ***variance inflation factors*** $\mathbf{(VIF)_k}$ when passed an *lm* object. You must first copy the function definition below into R:

```
vif.lm <- function(object, ...) {
   V <- summary(object)$cov.unscaled
   Vi <- crossprod(model.matrix(object))
         nam <- names(coef(object))
   if(k <- match("(Intercept)", nam, nomatch = FALSE)) {
                  v1 <- diag(V)[-k]
                  v2 <- (diag(Vi)[-k] - Vi[k, -k]^2/Vi[k,k])
                  nam <- nam[-k]
         } else {
                  v1 <- diag(V)
                  v2 <- diag(Vi)
                  warning("No intercept term detected. Results may surprise.")
         }
         structure(v1*v2, names = nam)
}
```

Now we call `vif.lm()`, passing it the *lm* object `my.reg`:

```
vif.lm(my.reg)

##       x1       x2       x3
## 3.521340 3.244021 1.167806
```

From the output, the ***variance inflation factors*** are:

- $\mathbf{(VIF)_1 = 3.52134}$ for the predictor X_1.
- $\mathbf{(VIF)_2 = 3.24402}$ for the predictor X_2.
- $\mathbf{(VIF)_3 = 1.16781}$ for the predictor X_3.