HW6 Math158

Joshua Jansen-Montoya

2022-10-13

Problem 7.3

Using the divusa data: 1. Fit a regression model with divorce as the response and unemployed, femlab, marriage, birth and military as predictors. Compute the condition numbers and interpret their meanings. 2. For the same model, compute the VIFs. Is there evidence that collinearity causes some predictors not to be significant? Explain. 3. Does the removal of insignificant predictors from the model reduce the collinearity? Investigate.

Answer 7.3

1. We can fit the regression model as follows,

```
library(faraway)
lmod <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data = divusa)
x <- model.matrix(lmod)[,-1]
e <- eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
## [1]  1.000000  7.432684  8.532498 13.757290 25.150782
```

Looking at these results, we can see that we have non-zero condition numbers, as well as some large condition numbers which would indicate a high level of collinearity in our model. 2. Using the same values, we can compute the VIFs as follows,

```
library(faraway)
lmod <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data = divusa)
require(faraway)
x <- model.matrix(lmod)[,-1]
vif(x)
```

```
## unemployed     femlab   marriage      birth   military
##   2.252888   3.613276   2.864864   2.585485   1.249596
```

We can note that we do not have any VIF's greater than 5 or 6 whihc indicates that we may not have the collinearity that we thought that we had. 3. We can look at the summary of our model as follows,

```
library(faraway)
lmod <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data = divusa)
summary(lmod)
```

```
##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusa)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -3.8611 -0.8916 -0.0496  0.8650  3.8300
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.48784    3.39378   0.733   0.4659
## unemployed  -0.11125    0.05592  -1.989   0.0505 .
## femlab       0.38365    0.03059  12.543  < 2e-16 ***
## marriage     0.11867    0.02441   4.861 6.77e-06 ***
## birth       -0.12996    0.01560  -8.333 4.03e-12 ***
## military    -0.02673    0.01425  -1.876   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 71 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9152
## F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16
```

Using these results, we can remove unemployed and military from our model, from which we will recalculate our condition numbers and our VIFs as desired,

```
library(faraway)
lmod <- lm(divorce ~ femlab + marriage + birth, data = divusa)
require(faraway)
x <- model.matrix(lmod)[,-1]
vif(x)
```

```
##   femlab marriage    birth
## 1.893390 2.201891 2.008469
```

```
e <- eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
## [1]  1.000000  7.432012 13.659660
```

We can note that we have reduced the VIF values of the model, and that we have done away with our high condition numbers which indicates that the removal of the insignificant predictors from the model does remove the collinearity to some degree.

Problem 7.4

For the longley data, fit a model with Employed as the response and the other variables as predictors. 1. Compute and comment on the condition numbers. 2. Compute and comment on the correlations between the predictors. 3. Compute the variance inflation factors.

Answer 7.4

1. We can compute the condition numbers as follows,

```
library(faraway)
lmod <- lm(Employed ~ GNP.deflator + GNP+ Unemployed + Armed.Forces + Population + Year, data = longl
x <- model.matrix(lmod)[,-1]
e <- eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
## [1]    1.00000   17.85504   25.15256   60.78472 1647.47771 5751.21560
```

After computing the condition numbers we can see that we seem to have a very high level of collinearity within our data set as indicated by such high condition numbers. 2. We can compute the correlations between the predictors as follows,

```
library(faraway)
lmod <- lm(Employed ~ GNP.deflator + GNP+ Unemployed + Armed.Forces + Population + Year, data = longl
x <- model.matrix(lmod)[,-1]
round(cor(x[,1:6]),2)
```

```
##              GNP.deflator  GNP Unemployed Armed.Forces Population Year
## GNP.deflator         1.00 0.99       0.62         0.46       0.98 0.99
## GNP                  0.99 1.00       0.60         0.45       0.99 1.00
## Unemployed           0.62 0.60       1.00        -0.18       0.69 0.67
## Armed.Forces         0.46 0.45      -0.18         1.00       0.36 0.42
## Population           0.98 0.99       0.69         0.36       1.00 0.99
## Year                 0.99 1.00       0.67         0.42       0.99 1.00
```

Looking at our correlation matrix, we can see that there seems to be a high level of correlation between GNP.deflator with GNP, Population, and Year, as well as GNP with Population and Year, and Population with Year. These correlations could be the source of our collinearity or at least a high contributer. 3. We can compute the variance inflation factors as follows,

```
library(faraway)
lmod <- lm(Employed ~ GNP.deflator + GNP+ Unemployed + Armed.Forces + Population + Year, data = longl
require(faraway)
x <- model.matrix(lmod)[,-1]
vif(x)
```

```
## GNP.deflator          GNP   Unemployed Armed.Forces   Population         Year
##    135.53244   1788.51348     33.61889      3.58893    399.15102    758.98060
```

We can see that we have very high VIF values for all of our variables except Armed.Forces which argees with our earlier calculations. ## Problem 7.6 Using the cheddar data, fit a linear model with taste as the response and the other three variables as predictors. 1. Is the predictor Lactic statistically significant in this model? 2. Give the R command to extract the p-value for the test of $\beta_{lactic} = 0$. Hint: look at summary()$coef. 3. Add normally distributed errors to Lactic with mean zero and standard deviation 0.01 and refit the model. Now what is the p-value for the previous test? 4. Repeat this same calculation of adding errors to Lactic 1000 times within for loop. Save the p-values into a vector. Report on the average p-value. Does this much measurement error make a qualitative difference to the conclusions? 5. Repeat the previous question but with a standard deviation of 0.1. Does this much measurement error make an important difference?

Answer 7.6

1. Looking at the following linear mode, we can see that,

```
library(faraway)
lmod <- lm(taste ~ Acetic + H2S + Lactic, data = cheddar)
summary(lmod)
```

```
##
## Call:
## lm(formula = taste ~ Acetic + H2S + Lactic, data = cheddar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.390  -6.612  -1.009   4.908  25.449
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -28.8768    19.7354  -1.463  0.15540   
## Acetic        0.3277     4.4598   0.073  0.94198   
## H2S           3.9118     1.2484   3.133  0.00425 **
## Lactic       19.6705     8.6291   2.280  0.03108 * 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 10.13 on 26 degrees of freedom
## Multiple R-squared:  0.6518, Adjusted R-squared:  0.6116 
## F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

We can see that Lactic is significant to a 0.05% level as desired. 2. We can find the p value relating to the $\beta_{lactic} = 0$ as follows,

```
library(faraway)
lmod <- lm(taste ~ Acetic + H2S + Lactic, data = cheddar)
summary(lmod)$coef
```

```
##                Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) -28.8767696  19.735418 -1.4631952 0.155399149
## Acetic        0.3277413   4.459757  0.0734886 0.941979774
## H2S           3.9118411   1.248430  3.1334077 0.004247081
## Lactic       19.6705434   8.629055  2.2795710 0.031079481
```

is 0.0311. 3. Now, we can add normally distributed error to Lactic with mean 0 and STDev 0.01 and refit the model as follows,

```
library(faraway)
lmod <- lm(taste ~ Acetic + H2S + I(Lactic + rnorm(length(Lactic), 0, 0.01)), data = cheddar)
summary(lmod)$coef
```

```
##                                               Estimate Std. Error     t value
## (Intercept)                                 -28.982418  19.651193 -1.47484262
## Acetic                                        0.250820   4.440839  0.05648031
## H2S                                           3.875196   1.244936  3.11276848
## I(Lactic + rnorm(length(Lactic), 0, 0.01))  20.200956   8.638662  2.33843571
##                                                Pr(>|t|)
## (Intercept)                                 0.152262182
## Acetic                                      0.955390997
## H2S                                         0.004468293
## I(Lactic + rnorm(length(Lactic), 0, 0.01)) 0.027327787
```

```
summary(lmod)$coef[4, 4]
```

```
## [1] 0.02732779
```

We can see that in this instance, we obtian an estimated p value of 0.0326, which is very close to our original value. 4. Repeating this 1000 times and taking the average, we find that the average is,

```
library(faraway)
vec = vector(,1000)
for (i in 1:1000) {
  lmod <- lm(taste ~ Acetic + H2S + I(Lactic + rnorm(length(Lactic), 0, 0.01)), data = cheddar)
  vec[i] =  summary(lmod)$coef[4, 4]
}
x <- mean(vec[i])
```

```
x
```

```
## [1] 0.0277058
```

The mean value we obtain is 0.0332 which is consistent with what we found earlier. 5. Repeating the before mentioned experiment with a STDev of

```
library(faraway)
vec = vector(,1000)
for (i in 1:1000) {
  lmod <- lm(taste ~ Acetic + H2S + I(Lactic + rnorm(length(Lactic), 0, 0.1)), data = cheddar)
  vec[i] =  summary(lmod)$coef[4, 4]
}
x <- mean(vec[i])
x
```

```
## [1] 0.02975039
```

We can note that using the rnorm with 0.1, then it follows that our mean p value changes greatly, with more than doubling. So yes, the measurement error does make a difference. ## Problem 7.8 Use the fat data, fitting the model described in Section 4.2. 1. Compute the condition numbers and variance inflation factors. Comment on the degree of collinearity observed in the data. 2. Cases 39 and 42 are unusual. Refit the model without these two cases and recompute the collinearity diagnostics. Comment on the differences observed from the full data fit. 3. Fit a model with brozekas the response and just age, weight and height as predictors. Compute the collinearity diagnostics and compare to the full data fit. 4. Compute a 95% prediction interval for brozek for the median values of age, weight and height. 5. Compute a 95% prediction interval for brozek for age=40, weight=200 and height=73. How does the interval compare to the previous prediction? 6. Compute a 95% prediction interval for brozek for age=40, weight=130 and height=73. Are the values of predictors unusual? Comment on how the inter- val compares to the previous two answers. ## Answer 7.8 1. We can calculate the VIF and condition numbers for the mentioned model as follows,

```
library(faraway)
lmod <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip + thigh + knee + ankle + bicep
x <- model.matrix(lmod)[,-1]
e <- eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
##  [1]   1.00000  17.47144  25.30482  58.60610  83.59121 100.63222 137.89717
##  [8] 175.28623 192.61449 213.00748 228.15747 268.20620 555.67072
```

```
require(faraway)
vif(x)
```

```
##       age    weight    height      neck     chest     abdom       hip     thigh
##  2.250450 33.509320  1.674591  4.324463  9.460877 11.767073 14.796520  7.777865
##      knee     ankle    biceps   forearm     wrist
##  4.612147  1.907961  3.619744  2.192492  3.377515
```

Looking at our values, we can see that there seems to appear to be a high level of collinearity within our dataset. 2. Refitting the model without cases 39 and 42, we can see that we obtain,

```
library(faraway)
fatRemoved <- fat[-c(39, 42), ]
lmod <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip + thigh + knee + ankle + bicep
x <- model.matrix(lmod)[,-1]
e <- eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
##  [1]   1.00000  18.39787  26.21547  61.53224  91.07633 114.44792 148.72518
##  [8] 178.80871 202.08708 211.78359 240.69468 276.35018 554.79777
```

```
require(faraway)
vif(x)
```

```
##       age    weight    height      neck     chest     abdom       hip     thigh
##  2.278191 45.298843  3.439587  3.978898 10.712505 11.967580 12.146249  7.153711
##      knee     ankle    biceps   forearm     wrist
##  4.441752  1.810253  3.409524  2.422878  3.263677
```

Removing these values, we can see that there is a very small change in our calculated values for VIF and for our condition numbers, indicating that removing these values does not have a strong effect on the model. 3. Fitting the reduced version of the model, we can see that we obtain,

```
library(faraway)
lmod <- lm(brozek ~ age + weight + height, data=fat)
x <- model.matrix(lmod)[,-1]
e <- eigen(t(x) %*% x)
sqrt(e$val[1]/e$val)
```

```
## [1]  1.00000 13.51194 22.67250
```

```
require(faraway)
vif(x)
```

```
##      age   weight   height
## 1.032253 1.107050 1.140470
```

We can note that looking at these values, we have much lower values of collinearity, as indicated by our VIFs as well as significantly lower values for our condition numbers. This indicates that this model has a lower level of collinearity as desired. 4. We can calculate the 95% confidence interval for our reduced model as follows,

```
library(faraway)
medAge = median(fat$age)
medWeight = median(fat$weight)
medHeight = median(fat$height)
lmod <- lm(brozek ~ age + weight + height, data=fat)
predict(lmod, data.frame(age = medAge, weight = medWeight, height = medHeight), interval="predict", le
```

```
##        fit      lwr      upr
## 1 18.28132 7.659609 28.90304
```

5. Now caculating the 95% confidence interval for our reduced model with the new values, we obtain,

```
predict(lmod, data.frame(age = 40, weight = 200, height = 73), interval="predict", level = 0.95)
```

```
##        fit      lwr      upr
## 1 20.47854 9.837784 31.11929
```

We can note that the CI is more wide in this instance but not by a significant margin (within about 0.1 of each other). 6. Now caculating the 95% confidence interval for our reduced model with the new values, we obtain,

```
predict(lmod, data.frame(age = 40, weight = 130, height = 73), interval="predict", level = 0.95)
```

```
##        fit       lwr     upr
## 1 7.617419 -3.101062 18.3359
```

These results do not really make sense, since it is impossible to have a negative amount of body fat. Thus, while the other two intervals seem like they could be reasonable, this interval does not make physical sense.