STATISTICAL MODELLING Part IIC, Michaelmas 2019
Practical 4: More on linear regression By courtesy of Dr. S. A. Bacallado and Dr. R. Shah

Solutions

```
file_path <- "https://raw.githubusercontent.com/AJCoca/SM19/master/"
Movies <- read.csv(paste(file_path, "Movies.csv", sep =""))
attach(Movies)

MoviesLM <- lm(log(Total.Gross) ~ log(Opening) + Screens + RT + log(Budget))
lev <- hatvalues(MoviesLM)
high_lev <- which(lev > 3*4/nrow(Movies)) # gives the high leverage obs
lev[high_lev] # actually shows their leverage
```

```
##        10        76        99       111
## 0.1177087 0.1168901 0.4453488 0.1110171
```

Observation 99 has much higher leverage than the rest. This is largely because the budget was so low for the film.

```
Movies[99, ]
```

```
##                      Film.name Total.Gross Opening Screens RT Budget
## 99 The Last House on the Left     32.7522 14.1187    2401 63    0.1
```

We now fit a model without this observation

```
MoviesLM_sub <- lm(log(Total.Gross) ~ log(Opening) +  RT + log(Budget), subset=-99)


## Prediction intervals
Movies2010 <- read.csv(paste(file_path, "Movies2010.csv", sep =""))

pred_intervals <- predict(MoviesLM_sub, Movies2010, interval="prediction")
pred_intervals_trans <- exp(pred_intervals)

target <- Movies2010$Total.Gross
mean((pred_intervals_trans[, 2] < target) & (pred_intervals_trans[, 3] > target))
```

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

```
# (pred_intervals_trans[, 2] < target) gives a logical vector
# The & performs a copmonentwise AND operation
# Applying the mean function first coerces the logical vector into an vector where
# TRUE -> 1 and FALSE -> 0
```

Every film is inside its prediction interval. We can also see how well our model predicts film earnings by displaying the true earnings alongside the predicted earnings.

```
cbind(target, pred_intervals_trans)
```

```
##       target       fit       lwr       upr
## 1   44.87548  48.11580 27.883467  83.02877
## 2   30.10158  44.39826 25.698938  76.70377
## 3   80.01484  79.89386 46.166683 138.26049
## 4   43.31389  54.66047 31.734809  94.14793
## 5   12.48274  16.01772  9.266988  27.68618
## 6   24.07743  23.88221 13.820297  41.26974
```

```
## 7   25.91892  22.71789 13.161467  39.21315
## 8   40.16808  44.60775 25.839034  77.00952
## 9   88.76830 102.23210 59.274186 176.32300
## 10 128.01293 141.70253 82.048902 244.72729
## 11  94.83506 104.89410 60.853561 180.80735
## 12  39.12359  47.68044 27.555473  82.50353
## 13  60.02226  37.69657 21.821296  65.12133
## 14  62.18988 102.73584 59.295654 178.00045
## 15 110.48565 153.98449 88.626207 267.54190
## 16  15.28559  20.01586 11.570445  34.62569
```

Finally, we repeat the procedure with 50% prediction intervals.

```
pred_intervals <- predict(MoviesLM_sub, Movies2010, interval="prediction",level=0.5)
pred_intervals_trans <- exp(pred_intervals)

target <- Movies2010$Total.Gross
mean((pred_intervals_trans[, 2] < target) & (pred_intervals_trans[, 3] > target))
```

```
## [1] 0.5
```

```
# (pred_intervals_trans[, 2] < target) gives a logical vector
# The & performs a copmonentwise AND operation
# Applying the mean function first coerces the logical vector into an vector where
# TRUE -> 1 and FALSE -> 0
```

In this case only half of the observations are in their prediction interval.