STATS 3001 / STATS 4104 / STATS 7054
Statistical Modelling III
Practical 3 - Assumption checking
Solutions

Week 5

GOAL

The purpose of this practical is to explore the application of influence diagnostics in multiple regression. The learning objectives of the practical are

- To demonstrate the practical application of influence diagnostics;
- To demonstrate the correspondence between leverage and the distribution of the x-values;
- To verify the built-in function for calculating leverage;
- To demonstrate the correspondence between Cook's Distance and changes in the parameter estimates.

DATA

The file `hills.csv` contains the record times in 1984 for 35 Scottish hills races.

The dataset contains the following variables:

- `dist`: The total distance in miles
- `climb`: The total climb in feet
- `time`: The record time in minutes

Interest is focused on modelling `time` using `dist` and `climb` as predictors.

STEPS

- Read in the data

```
pacman::p_load(tidyverse, ggrepel)
```

```
##
## The downloaded binary packages are in
##  /var/folders/pr/yfj4d0gn5gv9gy0pnjtv1nt40000gp/T//RtmpEX0pgT/downloaded_packages
```

```
##
## ggrepel installed
```

```
hills <- read_csv(here::here("data", "hills.csv"))
```

```
## Rows: 35 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): race
## dbl (3): dist, climb, time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

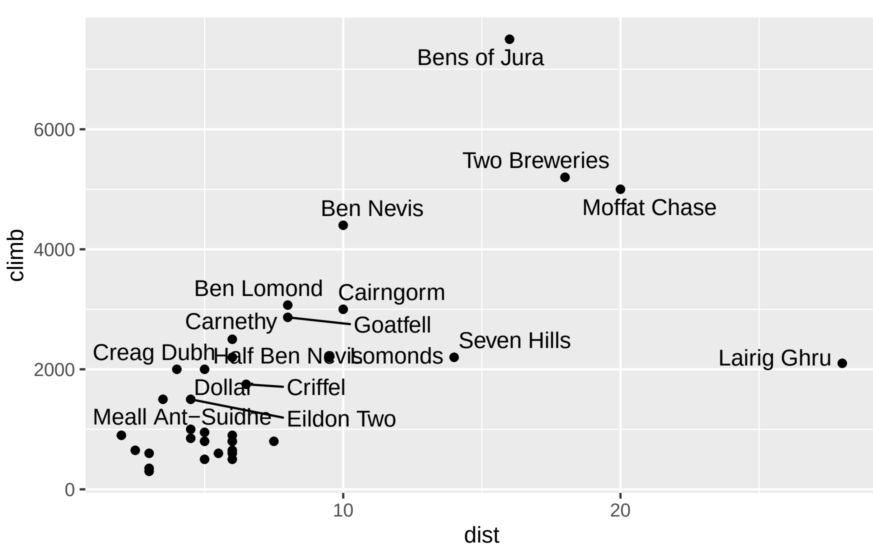
```r
hills
```

```
## # A tibble: 35 x 4
##    race          dist climb  time
##    <chr>        <dbl> <dbl> <dbl>
##  1 Greenmantle    2.5   650  16.1
##  2 Carnethy       6    2500  48.4
##  3 Craig Dunain   6     900  33.6
##  4 Ben Rha        7.5   800  45.6
##  5 Ben Lomond     8    3070  62.3
##  6 Goatfell       8    2866  73.2
##  7 Bens of Jura  16    7500 205.
##  8 Cairnpapple    6     800  36.4
##  9 Scolty         5     800  29.8
## 10 Traprain       6     650  39.8
## # ... with 25 more rows
```

- Obtain a scatter plot of `dist` vs `climb`. Identify the points that you believe will have high leverage. The package `ggrepel` is very cool for this.

```r
hills %>%
  ggplot(aes(dist, climb)) +
  geom_point() +
  geom_text_repel(aes(label = race))
```

```
## Warning: ggrepel: 18 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



- Calculate the leverage values from the design matrix for the model `{~ climb + dist` using the matrix expression given in lectures. Note the command `diag(H)` will extract the diagonal values of a square matrix `H`.

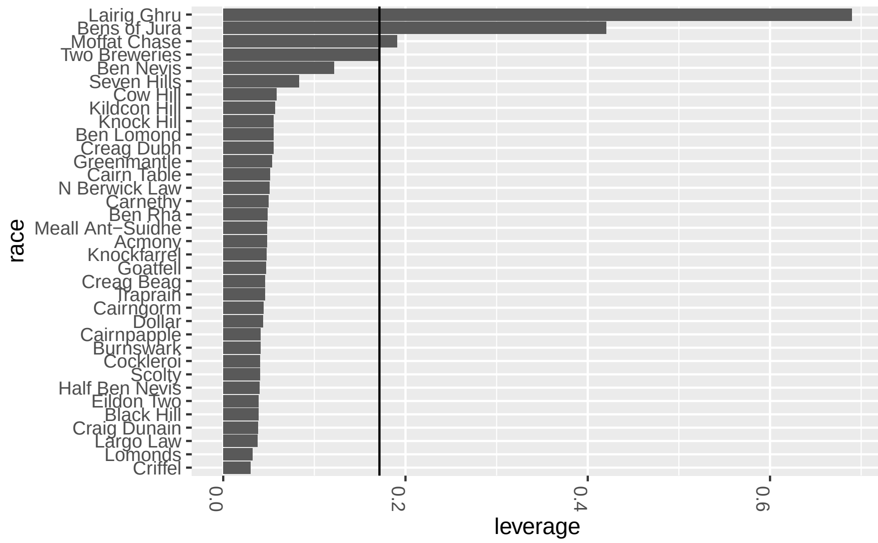
Identify the points with leverage greater than $2p/n$.

Check whether the points you identified on the scatter plot do have high leverage.

```
X <- model.matrix(~ climb + dist, data = hills)
H <- X %*% solve( t(X) %*% X ) %*% t(X)
hills <-
  hills %>%
  add_column(
    leverage = diag(H)
  )
hills
```

```
## # A tibble: 35 x 5
##    race          dist climb  time leverage
##    <chr>        <dbl> <dbl> <dbl>    <dbl>
##  1 Greenmantle    2.5   650  16.1   0.0538
##  2 Carnethy       6    2500  48.4   0.0495
##  3 Craig Dunain   6     900  33.6   0.0384
##  4 Ben Rha        7.5   800  45.6   0.0485
##  5 Ben Lomond     8    3070  62.3   0.0553
##  6 Goatfell       8    2866  73.2   0.0468
##  7 Bens of Jura  16    7500 205.    0.420
##  8 Cairnpapple    6     800  36.4   0.0410
##  9 Scolty         5     800  29.8   0.0403
## 10 Traprain       6     650  39.8   0.0457
## # ... with 25 more rows
```

```
p <- ncol(X)
n <- nrow(X)
hills %>%
  ggplot(aes(leverage, fct_reorder(race, leverage))) +
  geom_col() +
  theme(axis.text.x = element_text(angle = -90, hjust=0)) +
  geom_vline(xintercept = 2 * p / n) +
  labs(y = "race")
```



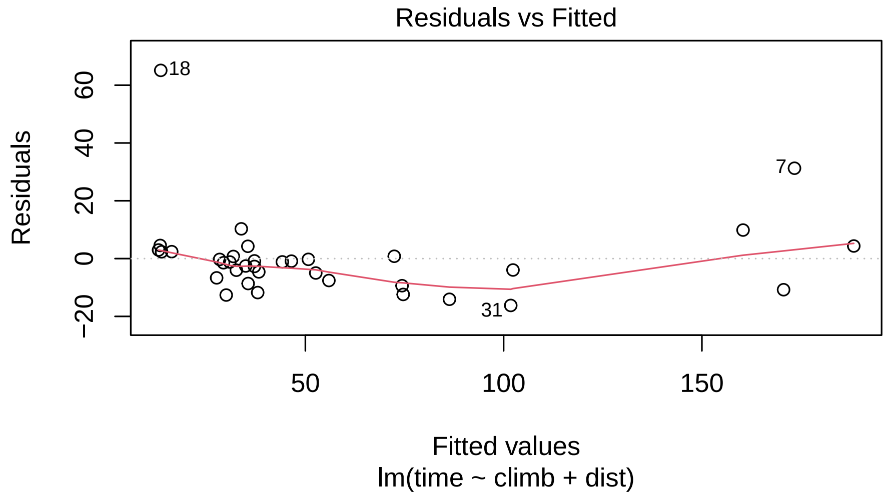
- Calculate the leverage values using the built-in `hatvalues()` function i n R and check that they agree with those calculated from the formula.
  (Note: R also provides functions, `cooks.distance()`, `rstudent()` and `rstandard()` to calculate Cook's distance, the studentized residuals and the standardized residuals, respectively.)

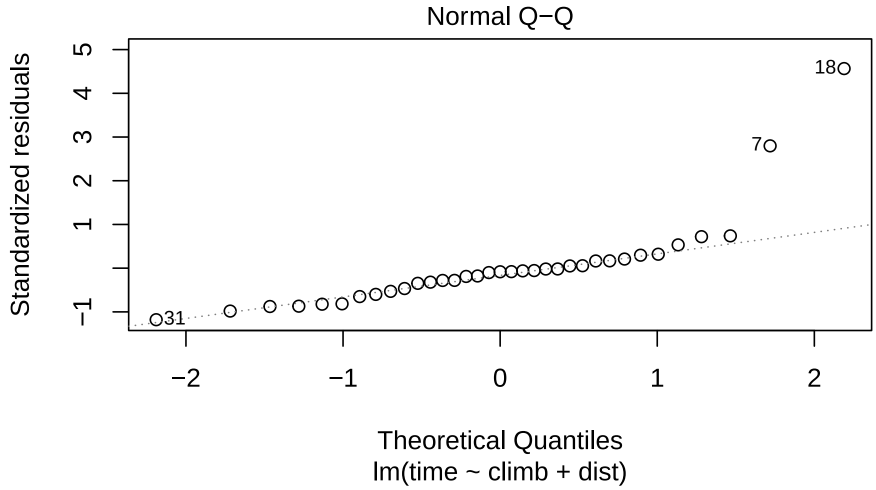
```
hills_lm <- lm(time ~ climb + dist, data = hills)
hills <-
  hills %>%
  add_column(
    r_leverage = hatvalues(hills_lm)
  )
hills
```

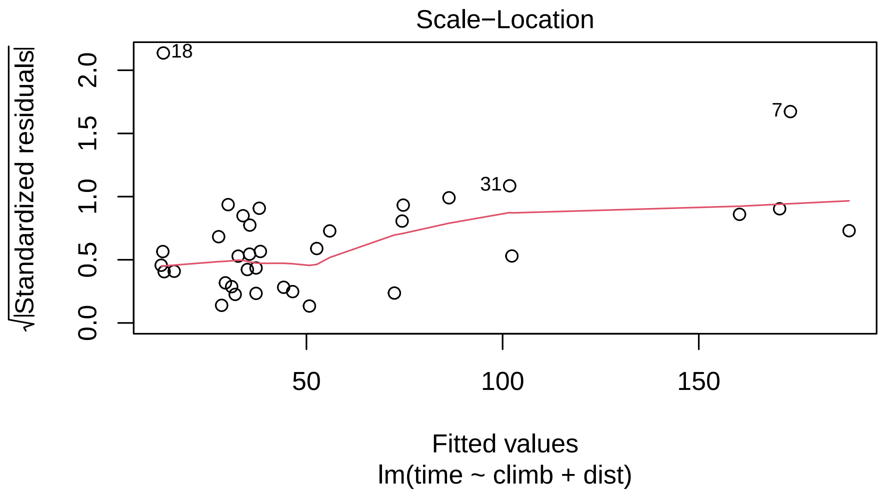
```
## # A tibble: 35 x 6
##    race          dist climb  time leverage r_leverage
##    <chr>        <dbl> <dbl> <dbl>    <dbl>      <dbl>
##  1 Greenmantle    2.5   650  16.1   0.0538     0.0538
##  2 Carnethy       6    2500  48.4   0.0495     0.0495
##  3 Craig Dunain   6     900  33.6   0.0384     0.0384
##  4 Ben Rha        7.5   800  45.6   0.0485     0.0485
##  5 Ben Lomond     8    3070  62.3   0.0553     0.0553
##  6 Goatfell       8    2866  73.2   0.0468     0.0468
##  7 Bens of Jura  16    7500 205.    0.420      0.420
##  8 Cairnpapple    6     800  36.4   0.0410     0.0410
##  9 Scolty         5     800  29.8   0.0403     0.0403
## 10 Traprain       6     650  39.8   0.0457     0.0457
## # ... with 25 more rows
```

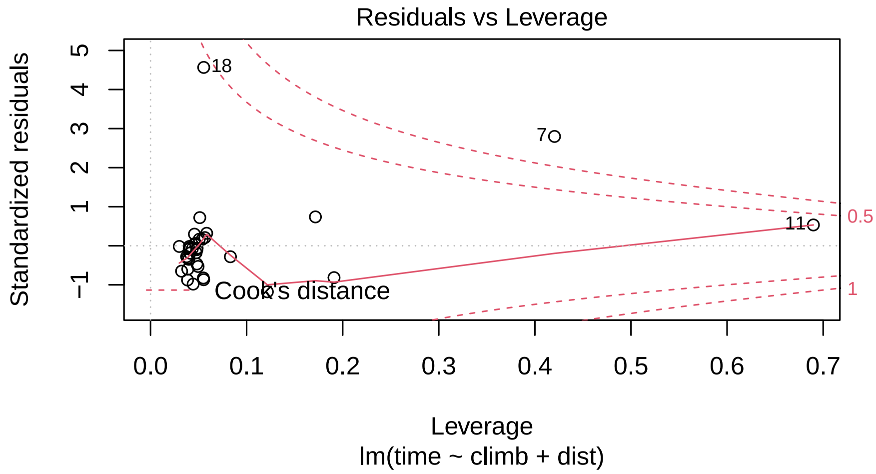
- Obtain the usual sequence of diagnostic plots from R.

```
plot(hills_lm)
```


Residuals vs Fitted
Residuals
Fitted values
lm(time ~ climb + dist)
18
7
31

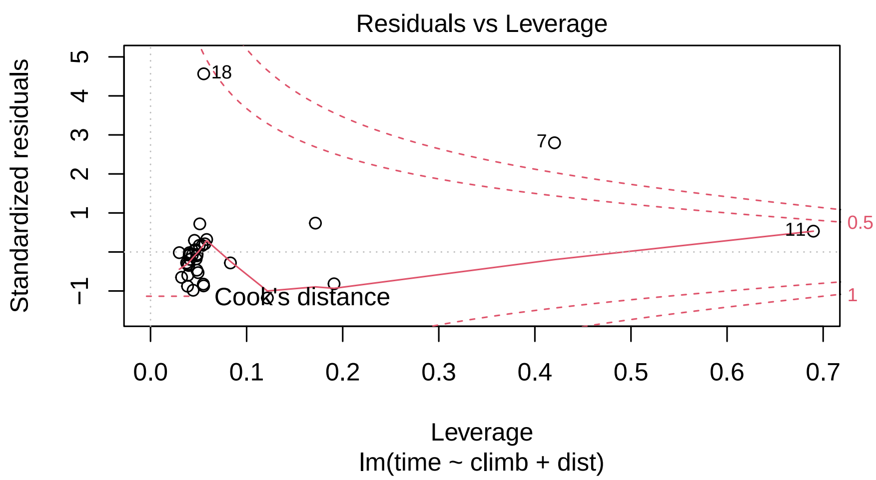

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(time ~ climb + dist)
18
7
31


Scale−Location
√|Standardized residuals|
Fitted values
lm(time ~ climb + dist)
18
7
31



- Based on the residuals vs leverage plot, identify the most influential point.

```
plot(hills_lm, which = 5)
```




```
hills %>% slice(7)
```

```
## # A tibble: 1 x 6
##   race          dist climb  time leverage r_leverage
##   <chr>        <dbl> <dbl> <dbl>    <dbl>      <dbl>
## 1 Bens of Jura    16  7500  205.    0.420      0.420
```

- Identify the points with the largest residual and the highest leverage, and comment.

```
augment(hills_lm) %>%
  add_column(
    race = hills$race
  ) %>%
  filter(
    .resid == max(.resid) | .hat == max(.hat) | .cooksd == max(.cooksd)
  ) %>%
  select(race, .resid, .hat, .cooksd)
```

```
## # A tibble: 3 x 4
##   race          .resid   .hat .cooksd
##   <chr>          <dbl>  <dbl>   <dbl>
## 1 Bens of Jura   31.3  0.420    1.89
## 2 Lairig Ghru     4.36 0.690    0.211
## 3 Knock Hill     65.1  0.0554   0.407
```

So `Knock Hill` has largest residual, but leverage is small, while `Lairig Ghru` has the largest leverage, but small residual.

- Fit the same model to the data with the most influential point removed.

```
hills_lm2 <- lm(time ~ climb + dist,
                data = hills %>% filter(race != "Bens of Jura"))
```

- Calculate Cook's distance according to the formula

$$D_i^2 = \frac{\left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{(i)}\right)^T \left[\hat{\text{Var}}(\hat{\boldsymbol{\beta}})\right]^{-1} \left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{(i)}\right)}{p} = \frac{\left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{(i)}\right)^T (X^T X) \left(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{(i)}\right)}{p s_e^2}.$$

Check that your value agrees with that produced by the built-in `cooks.distance` function.

```
beta <- coef(hills_lm)
beta_i <- coef(hills_lm2)
X <- model.matrix(hills_lm)
se2 <- glance(hills_lm)$sigma^2
p <- ncol(X)
t(beta - beta_i) %*% t(X) %*% X %*% (beta-beta_i) / (p*se2)
```

```
##          [,1]
## [1,] 1.893349
```