# Lecture 16: Diagnostics in multiple linear regression

Pratheepa Jeganathan

10/28/2019

# Recap

- What is a regression model?
- Descriptive statistics – graphical
- Descriptive statistics – numerical
- Inference about a population mean
- Difference between two population means
- Some tips on R
- Simple linear regression (covariance, correlation, estimation, geometry of least squares)
  - Inference on simple linear regression model
  - Goodness of fit of regression: analysis of variance.
  - $F$-statistics.
  - Residuals.
  - Diagnostic plots for simple linear regression (graphical methods).

# Recap

- Multiple linear regression
  - Specifying the model.
  - Fitting the model: least squares.
  - Interpretation of the coefficients.
  - Matrix formulation of multiple linear regression
  - Inference for multiple linear regression
    - $T$-statistics revisited.
    - More $F$ statistics.
    - Tests involving more than one $\beta$.

# Outline

- Diagnostics – more on graphical methods and numerical methods
- Different types of residuals
- Influence
- Outlier detection
- Multiple comparison (Bonferroni correction)
- Residual plots:
  - partial regression (added variable) plot,
  - partial residual (residual plus component) plot.

# Data

- The dataset we will use is based on record times on Scottish hill races.

| Variable | Description |
|---|---|
| Time | Record time to complete course |
| Distance | Distance in the course |
| Climb | Vertical climb in the course |

- Time: response variable
- Distance and Climb: predictor variables

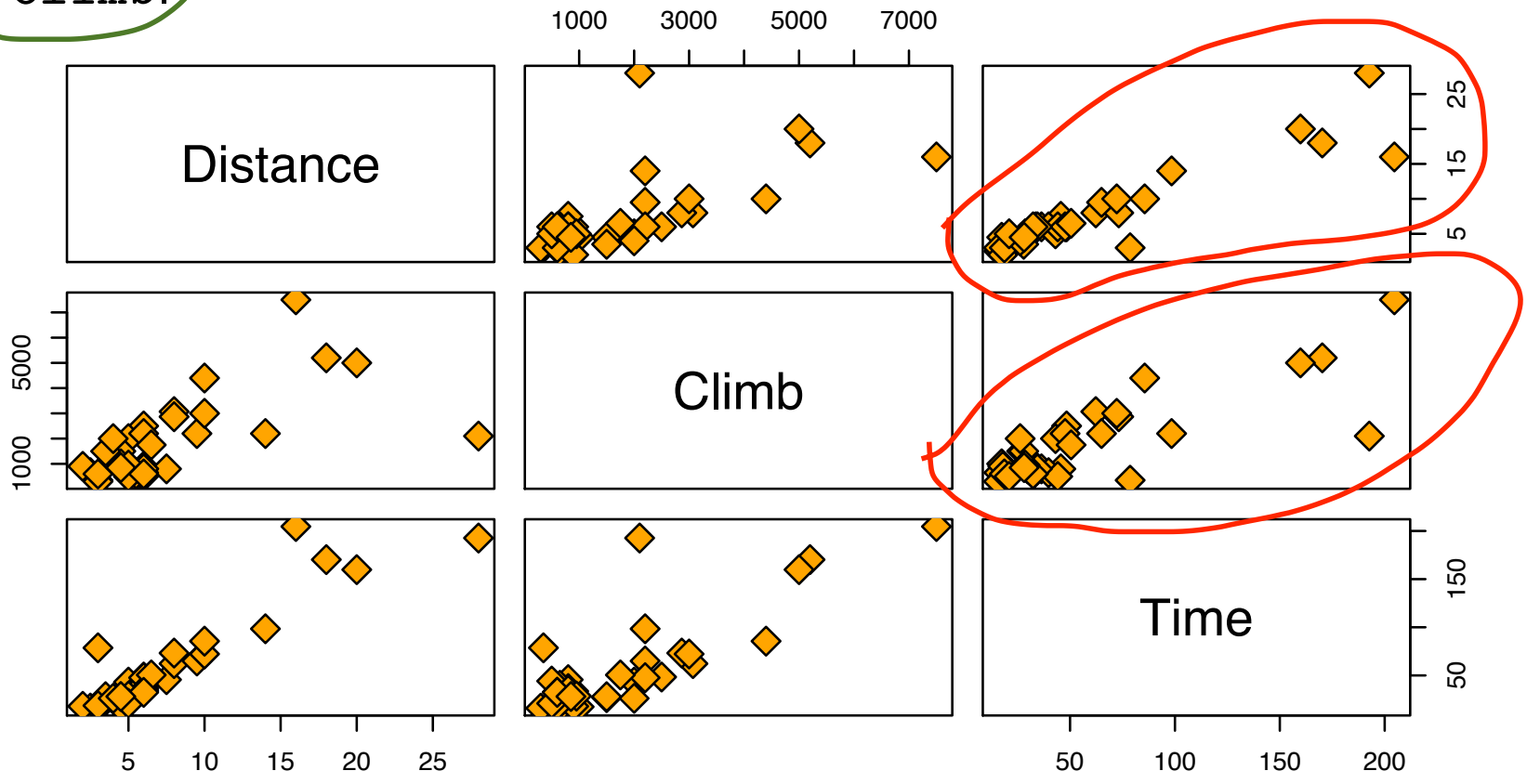
# Data

```
url = 'http://www.statsci.org/data/general/hills.txt'
races.table = read.table(url,
  header=TRUE, sep='\t')
head(races.table)
```

```
##           Race Distance Climb   Time
## 1 Greenmantle      2.5   650 16.083
## 2    Carnethy      6.0  2500 48.350
## 3 CraigDunain      6.0   900 33.650
## 4      BenRha      7.5   800 45.600
## 5   BenLomond      8.0  3070 62.267
## 6    Goatfell      8.0  2866 73.217
```

# Exploratory analysis

- As we'd expect, the time increases both with `Distance` and `Climb`.




Let's look at our multiple regression model.

# Multiple linear regression (MLR)

```
races.lm = lm(Time ~ Distance + Climb,
  data=races.table)
```

```
Call:
lm(formula = Time ~ Distance + Climb, data = races.table)

Residuals:
    Min      1Q  Median      3Q     Max
-16.215  -7.129  -1.186   2.371  65.121

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.992039   4.302734  -2.090   0.0447 *
Distance     6.217956   0.601148  10.343 9.86e-12 ***
Climb        0.011048   0.002051   5.387 6.45e-06 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.68 on 32 degrees of freedom
Multiple R-squared:  0.9191,    Adjusted R-squared:  0.914
F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16
```

- But is this a good model?

# Diagnostics

# What can go wrong?

- Regression function can be wrong: maybe regression function should have some other form (see diagnostics for simple linear regression).
- Model for the errors may be incorrect:
  - may not be normally distributed.
  - may not be independent.
  - may not have the same variance.
- Detecting problems is more *art* than *science*, i.e. we cannot *test* for all possible problems in a regression model.

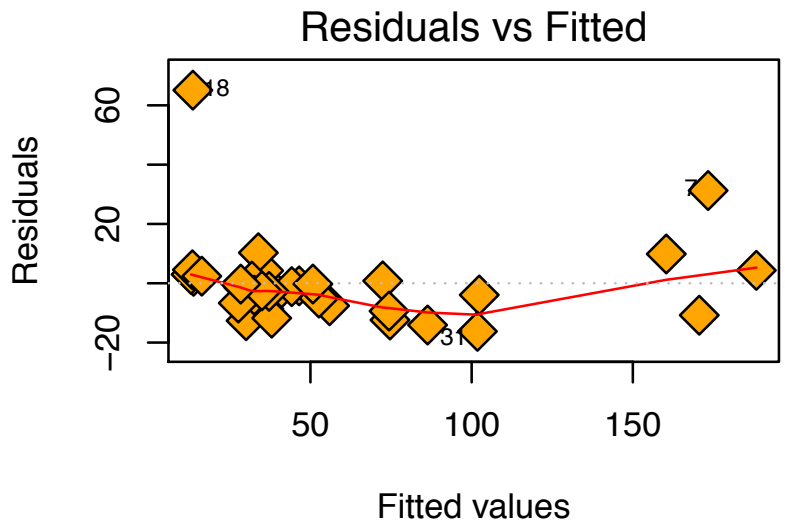
# Residual analysis

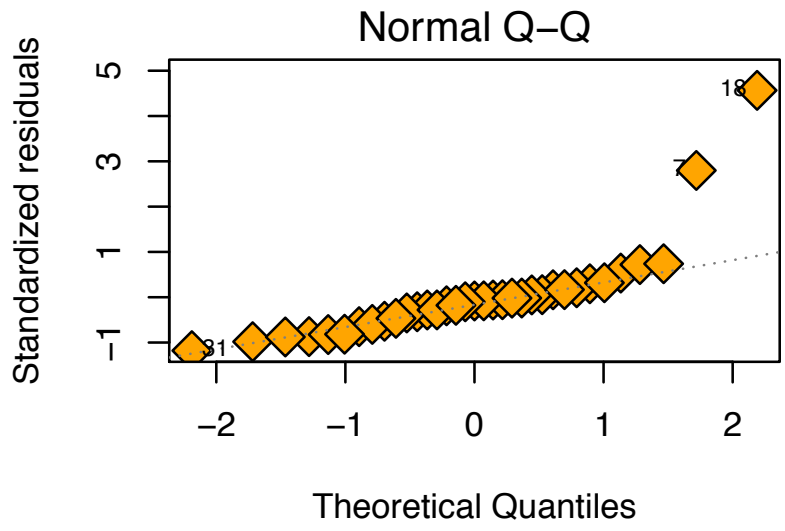
- Basic idea of diagnostic measures: if model is correct then residuals $e_i = Y_i - \widehat{Y}_i, 1 \leq i \leq n$ should look like a sample of (not quite independent) $N(0, \sigma^2)$ random variables.

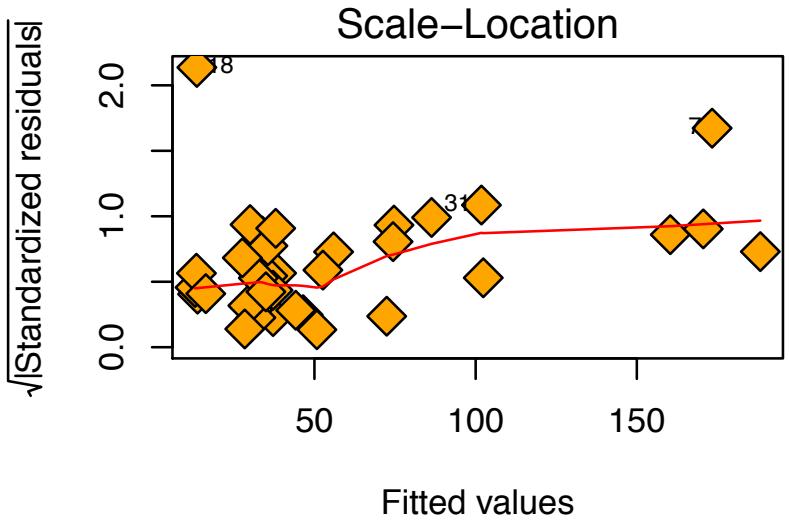
# Standard diagnostic plots

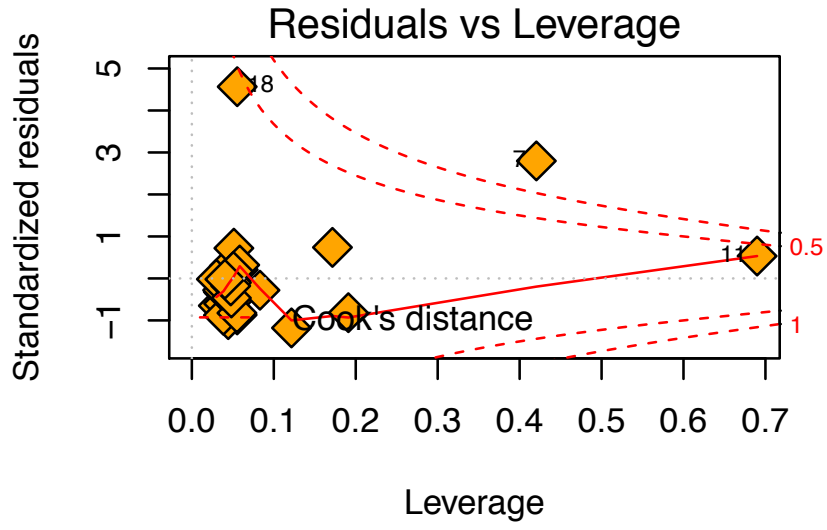
- `R` produces a set of standard plots for `lm` that help us assess whether our assumptions are reasonable or not.
  - We will go through each in some, but not too much, detail.
- As we see next, there are some quantities which we need to define in order to read these plots. We will define these first.

```
par(mfrow=c(2,2))
plot(races.lm, pch=23 ,bg='orange',cex=2)
```



# Possible problems & diagnostic checks

- Errors may not be normally distributed or may not have the same variance – `qqnorm` can help with this.
  - This may not be too important in large samples.
- Variance may not be constant. Can also be addressed in a plot of $X$ vs. $e$: *fan shape* or other trend indicate non-constant variance.
- Influential observations. Which points "affect" the regression line the most?
- Outliers: points where the model really does not fit!
  - Possibly mistakes in data transcription, lab errors, who knows? Should be recognized and (hopefully) explained.

# Types of residuals

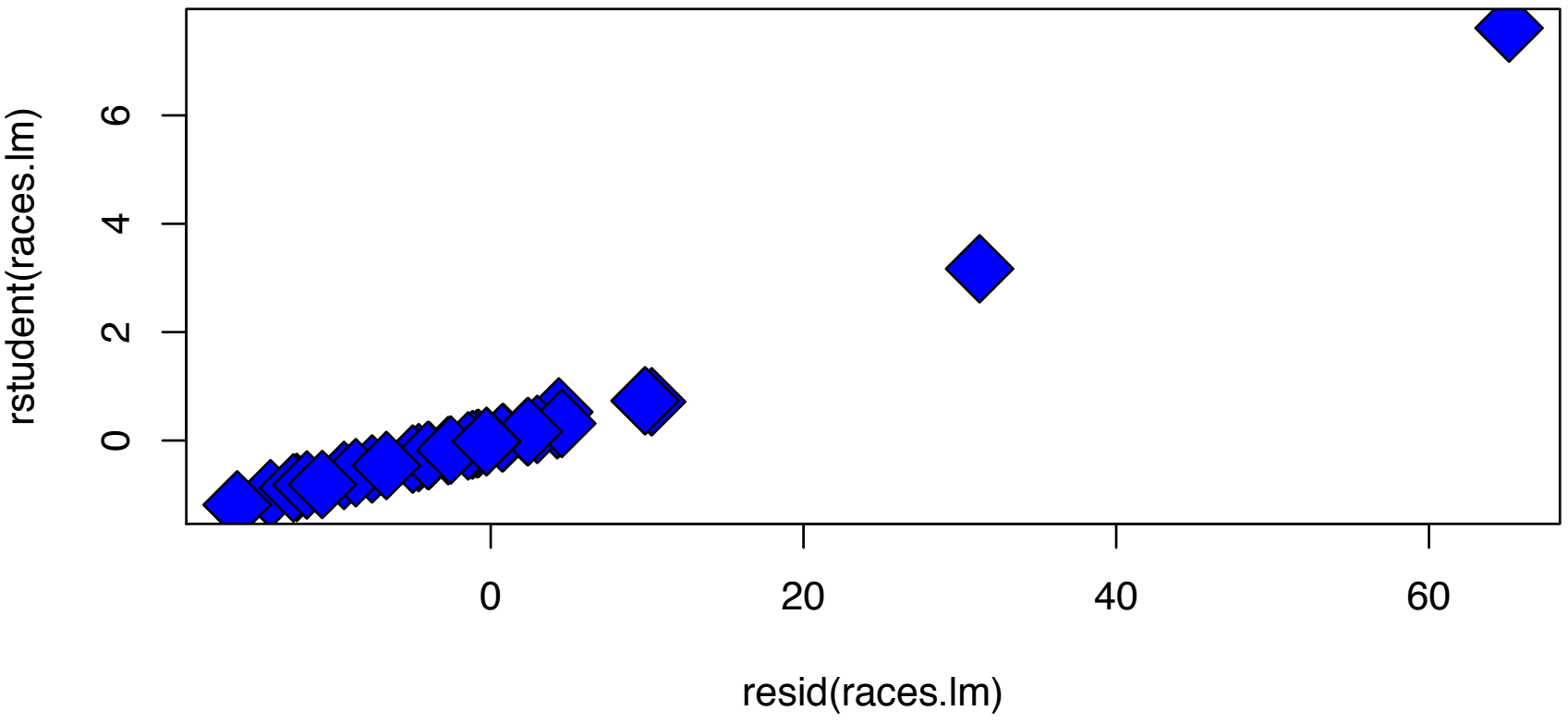
- Ordinary residuals: $e_i = Y_i - \widehat{Y}_i$. These measure the deviation of predicted value from observed value, but their distribution depends on unknown scale, $\sigma$.
- Internally studentized residuals (`rstandard` in R):

$$r_i = e_i / SE(e_i) = \frac{e_i}{\widehat{\sigma}\sqrt{1 - H_{ii}}}$$

- Above, $H$ is the "hat" matrix $H = X(X^T X)^{-1} X^T$.
- $r_i$'s are almost $t$-distributed, except $\widehat{\sigma}$ depends on $e_i$.

# Compare ordinary and studentized residuals

```
plot(resid(races.lm),
  rstudent(races.lm), pch=23, bg='blue', cex=3)
```



# Types of residuals

- Externally studentized residuals (`rstudent` in R):

$$t_i = \frac{e_i}{\widehat{\sigma_{(i)}}\sqrt{1 - H_{ii}}} \sim t_{n-p-2}.$$

$(n-1)-(p+1)$

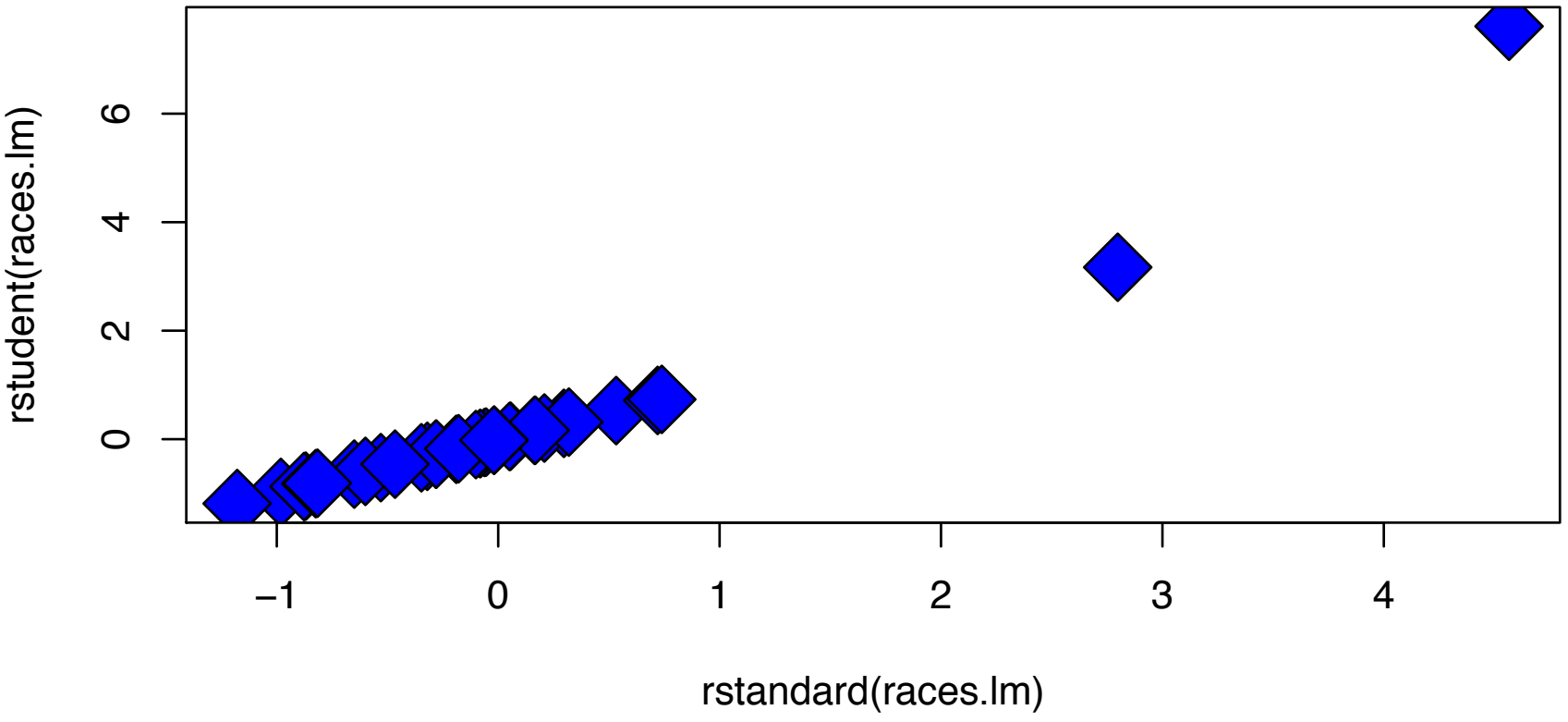
  - These are exactly $t$ distributed so we know their distribution and can use them for tests, if desired.

- The quantity $\hat{\sigma}^2_{(i)}$ is the mean squared error (MSE) of the model fit to all data except case $i$ (i.e. it has $n-1$ observations and $p$ predictors).

- Numerically, these residuals are highly correlated, as we would expect.

# Compare studentized and externally studentized residuals

```
plot(rstandard(races.lm),
  rstudent(races.lm), pch=23, bg='blue', cex=3)
```



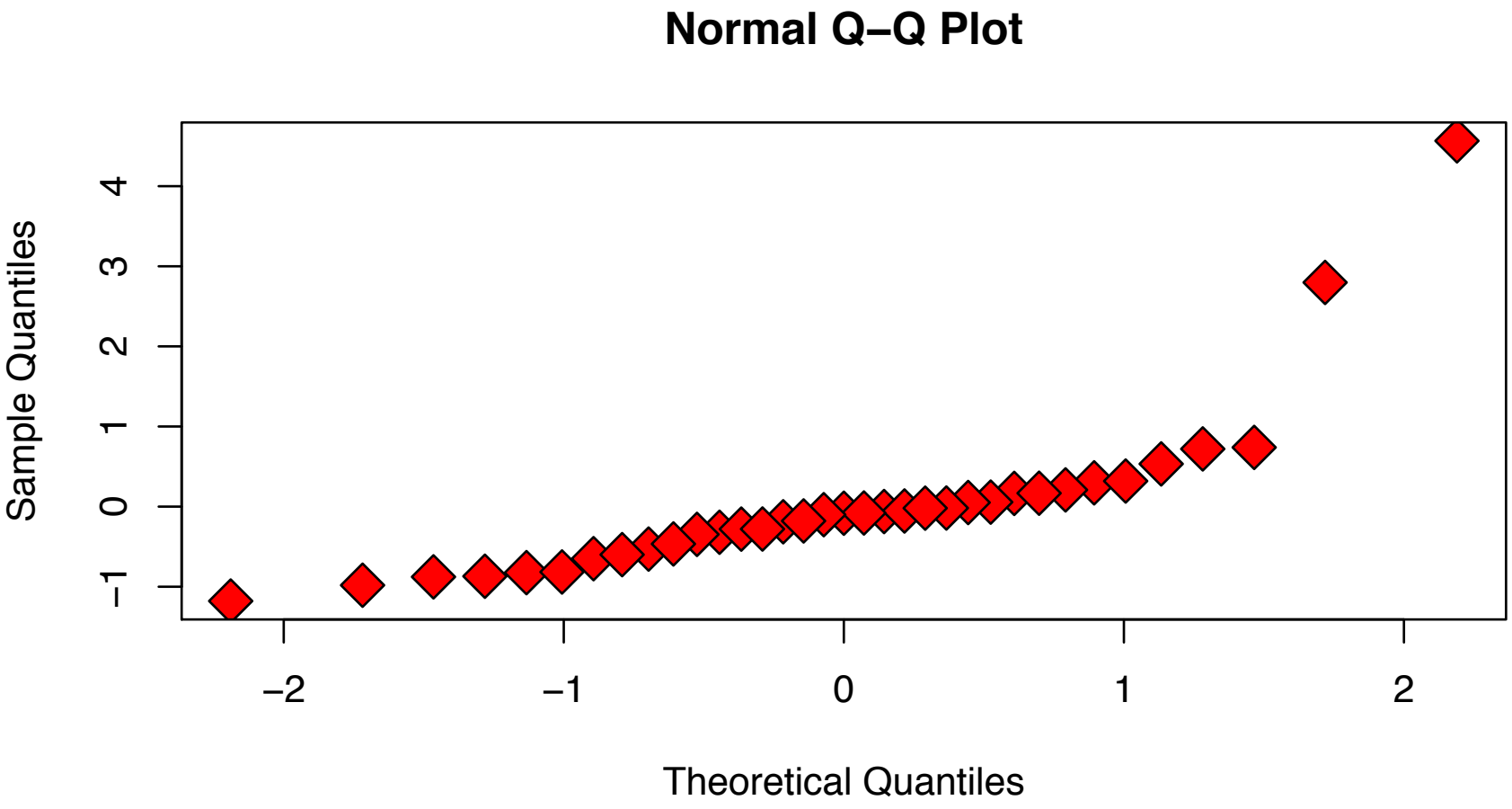
# Standard diagnostic plots

# Quantile plot for the residuals

- The first plot is the quantile plot for the residuals, that compares their distribution to that of a sample of independent normal.

```
qqnorm(rstandard(races.lm), pch=23,
  bg='red', cex=2)
```

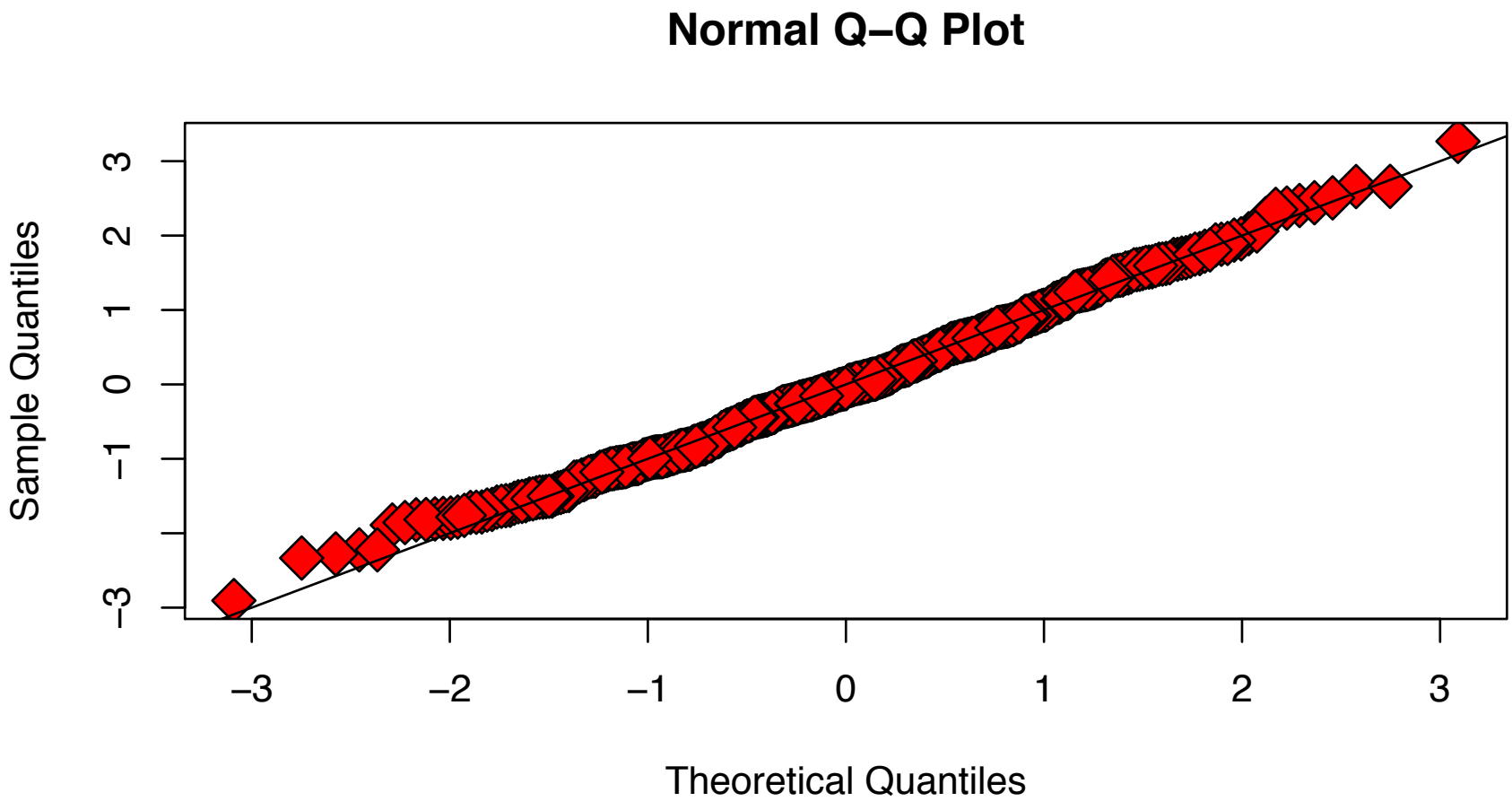
# Quantile plot for the residuals




- If the residuals were really normal we'd expect this plot to be roughly on the diagonal.

- For example, if we have a sample from a standard normal distirbution, the qq-plot looks like the following:

```
qqnorm(rnorm(500), pch=23,
  bg='red', cex=2)
abline(0, 1)
```



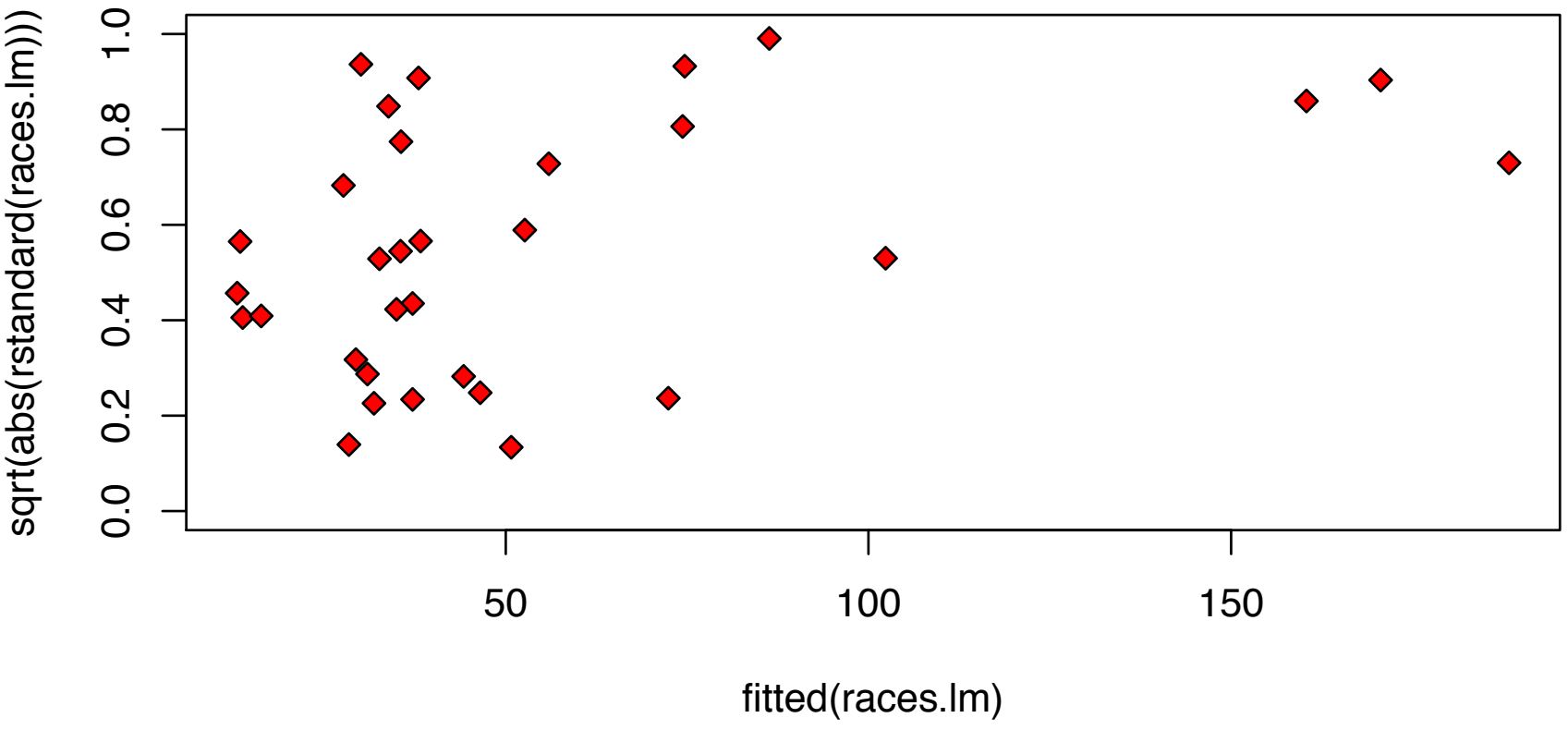
# Plots for addressing constant variance assumption

- Two other plots try address the constant variance assumptions.
  - If these plots have a particular shape (maybe the spread increases with $\hat{Y}$) then maybe the variance is not constant.

# Plots for addressing constant variance assumption

```
plot(fitted(races.lm),
  sqrt(abs(rstandard(races.lm))),
  pch=23, bg='red', ylim=c(0,1))
```

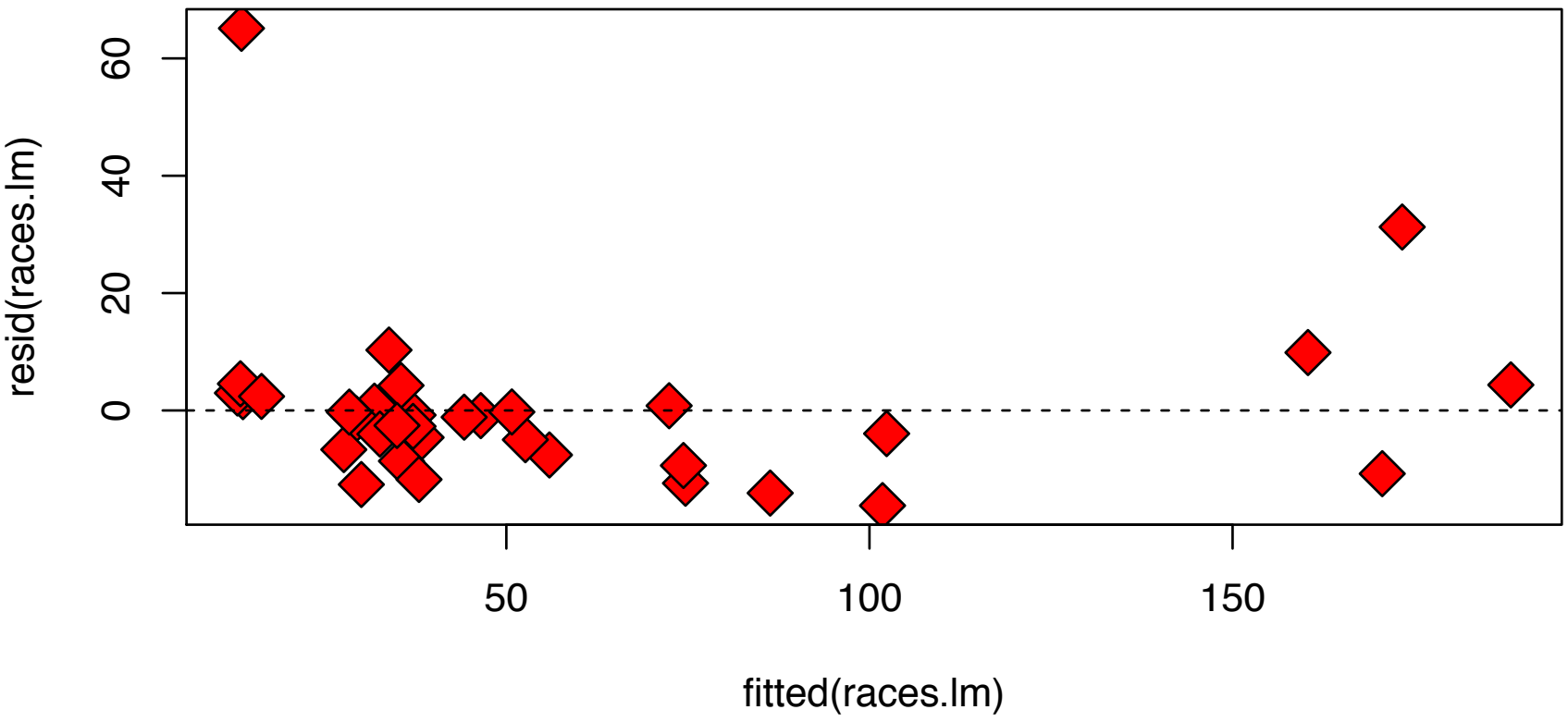
# Plots for addressing constant variance assumption



# Plots for addressing constant variance assumption

```
plot(fitted(races.lm),
  resid(races.lm), pch=23, bg='red', cex=2)
abline(h=0, lty=2)
```

# Plots for addressing constant variance assumption



# Influence of an observation

- Other plots provide an assessment of the `influence` of each observation.
- Usually, this is done by dropping an entire case $(y_i, x_i)$ from the dataset and refitting the model.
- In this setting, a$\cdot_{(i)}$ indicates $i$-th observation was not used in fitting the model.
- For example: $\widehat{Y}_{j(i)}$ is the regression function evaluated at the $j$-th observation predictors BUT the coefficients $(\widehat{\beta}_{0(i)}, \ldots, \widehat{\beta}_{p(i)})$ were fit after deleting $i$-th case from the data.

# Influence of an observation

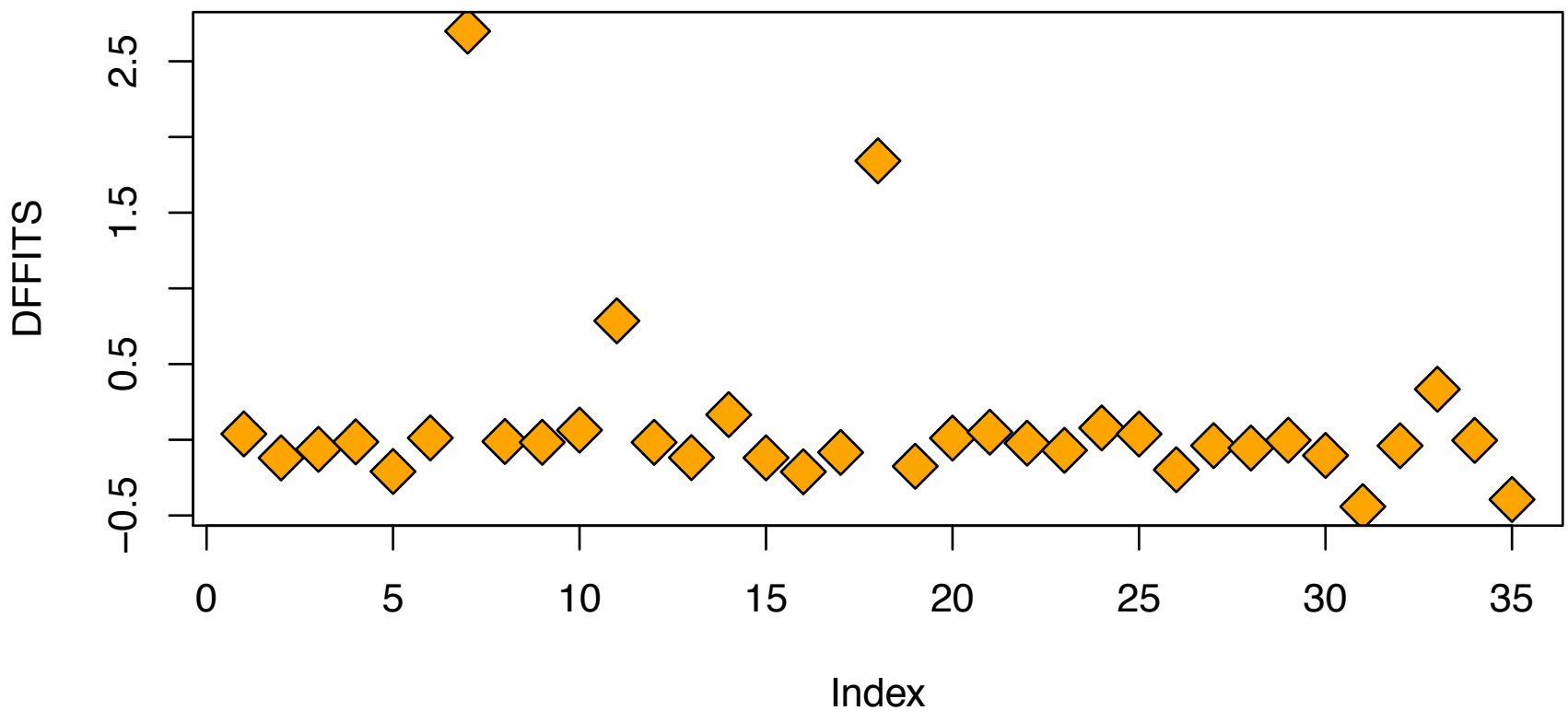
- Idea: if $\widehat{Y}_{j(i)}$ is very different than $\widehat{Y}_j$ (using all the data) then $i$ is an influential point, at least for estimating the regression function at $(X_{1,j}, \ldots, X_{p,j})$.
- Could also look at difference between $\widehat{Y}_{i(i)} - \widehat{Y}_i$, or any other measure.
- There are various standard measures of influence (DFFITS, Cook's distance, DFBETAS for $X_p$).

# DFFITS

- $$DFFITS_i = \frac{\widehat{Y}_i - \widehat{Y}_{i(i)}}{\widehat{\sigma}_{(i)}\sqrt{H_{ii}}}$$
- This quantity measures how much the regression function changes at the $i$-th case / observation when the $i$-th case / observation is deleted.
- For small/medium datasets: value of 1 or greater is "suspicious". For large dataset: value of $2\sqrt{(p+1)/n}$.
- `R` has its own standard rules similar to the above for marking an observation as influential.

# DFFITS plot

```
plot(dffits(races.lm), pch=23,
  bg='orange', cex=2, ylab="DFFITS")
```




- It seems that some observations had a high influence measured by *DFFITS*:

```
races.table[which(dffits(races.lm) > 0.5),]
```

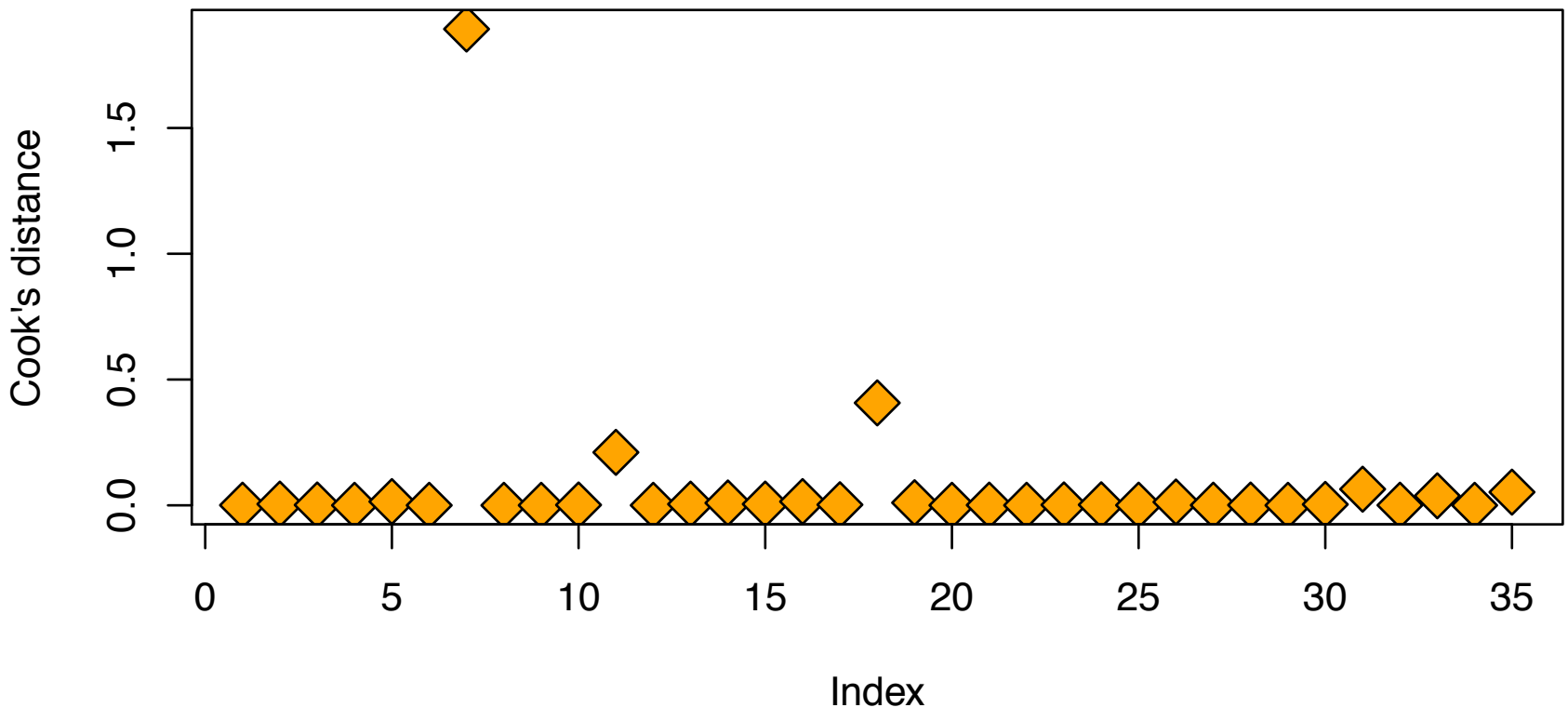
```
##           Race Distance Climb    Time
## 7  BensofJura       16  7500 204.617
## 11 LairigGhru       28  2100 192.667
## 18  KnockHill        3   350  78.650
```

- It is perhaps not surprising that the longest course and the course with the most elevation gain seemed to have a strong effect on the fitted values. What about `Knock Hill`? We'll come back to this later.

# Cook's distance

- Cook's distance measures how much the entire regression function changes when the $i$-th case is deleted.
- $$D_i = \frac{\sum_{j=1}^{n} (\widehat{Y}_j - \widehat{Y}_{j(i)})^2}{(p+1)\,\widehat{\sigma}^2}$$
- Should be comparable to $F_{p+1,n-p-1}$: if the "$p$-value" of $D_i$ is 50 percent or more, then the $i$-th case is likely influential: investigate further. (**CH**)
- Again, `R` has its own rules similar to the above for marking an observation as influential.
- What to do after investigation? No easy answer.

```
plot(cooks.distance(races.lm),
  pch=23, bg='orange', cex=2,
  ylab="Cook's distance")
```



```
races.table[which(cooks.distance(races.lm) > 0.1),]
```

```
##           Race Distance Climb    Time
## 7  BensofJura       16  7500 204.617
## 11 LairigGhru       28  2100 192.667
## 18  KnockHill        3   350  78.650
```

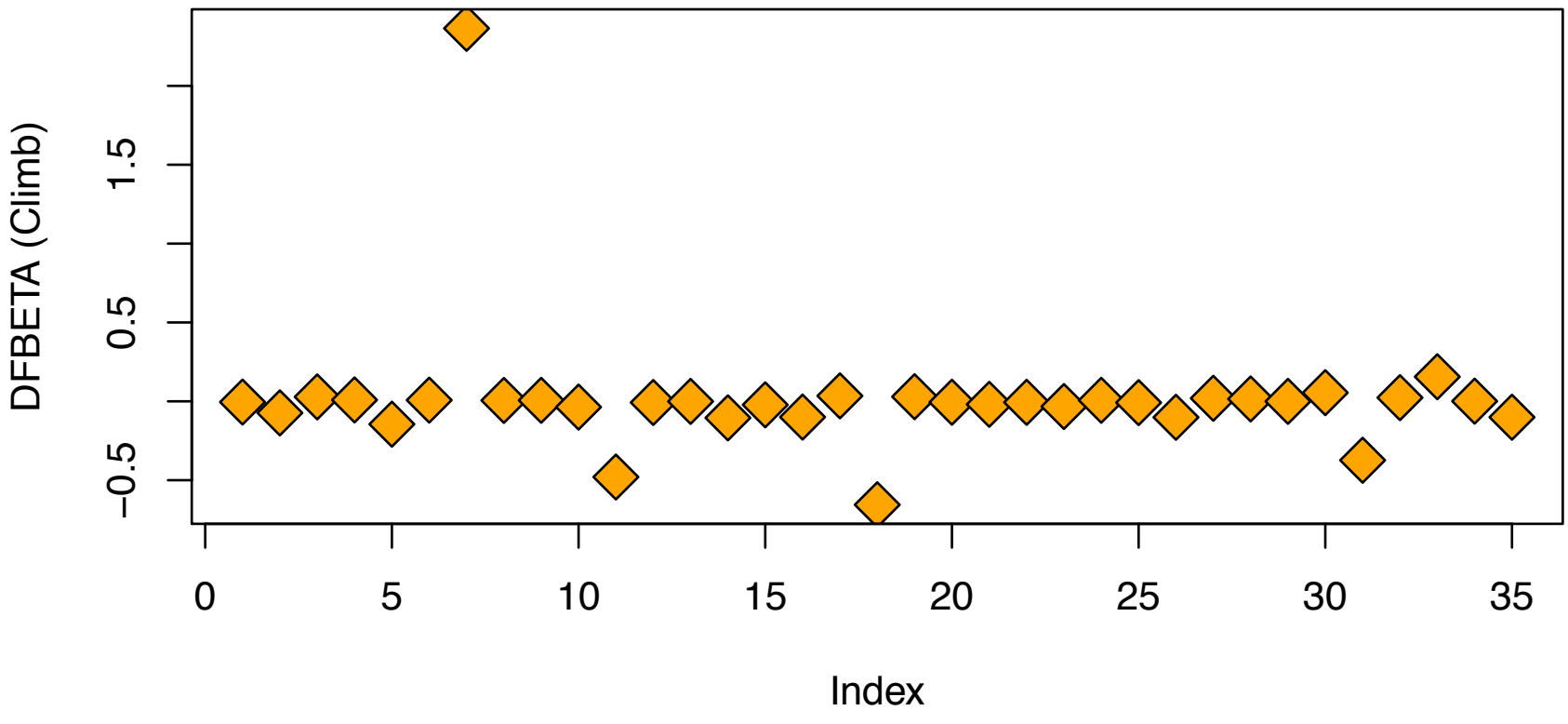
- Again, the same 3 races.
- This is not surprising as both *DFFITS* and Cook's distance measure changes in fitted values. The difference is that one measures the influence on one fitted value, while the other measures the influence on the entire vector of fitted values.

# DFBETAS

- This quantity measures how much the coefficients change when the $i$-th case is deleted.
- $$DFBETAS_{j(i)} = \frac{\widehat{\beta}_j - \widehat{\beta}_{j(i)}}{\sqrt{\widehat{\sigma}^2_{(i)}(X^T X)^{-1}_{jj}}}.$$
- For small/medium datasets: absolute value of 1 or greater is "suspicious". For large dataset: absolute value of $2/\sqrt{n}$.

# DFBETAS for $X_1$

```
plot(dfbetas(races.lm)[,'Climb'],
  pch=23, bg='orange',
  cex=2, ylab="DFBETA (Climb)")
```



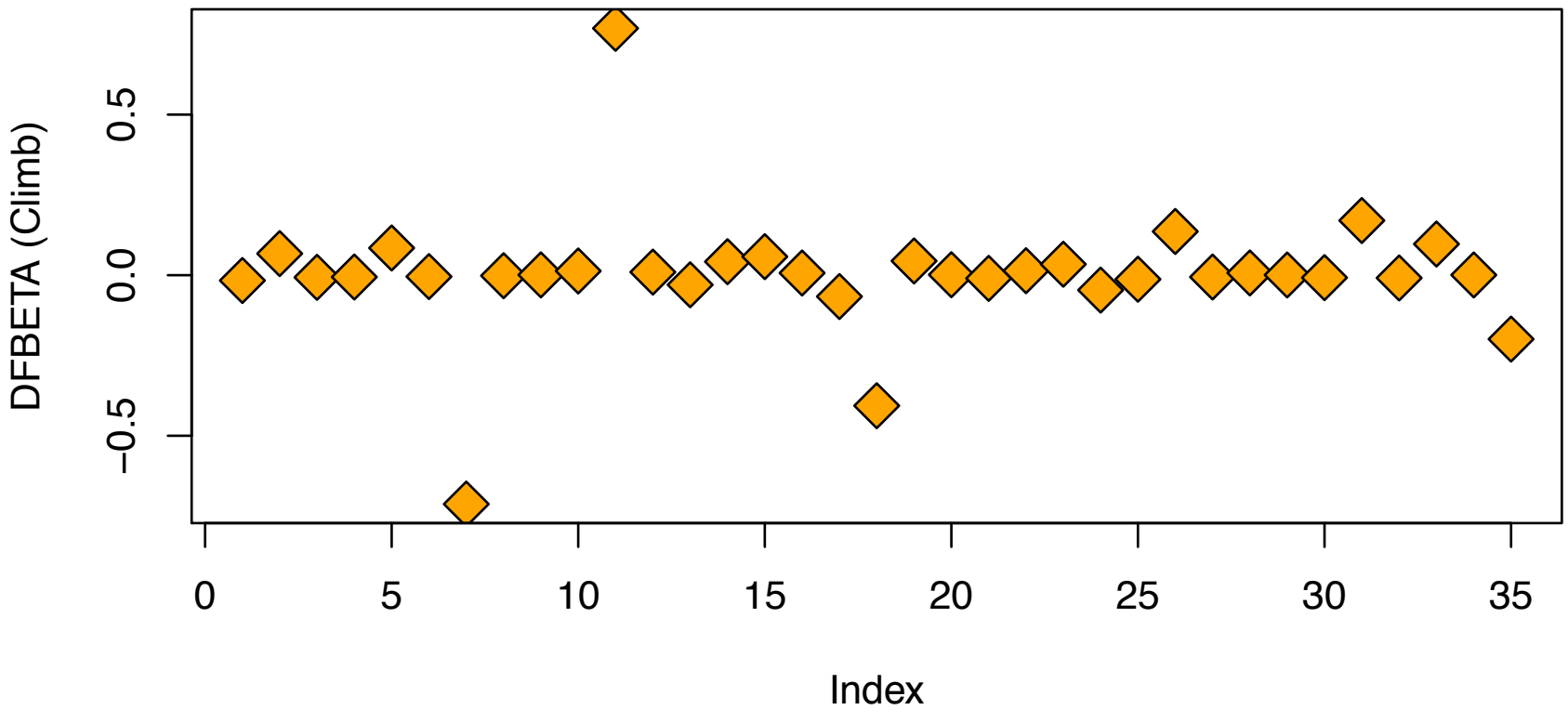
# DFBETAS for $X_1$

```
races.table[which(abs(dfbetas(races.lm)[,'Climb']) > 1),]
```

```
##          Race Distance Climb    Time
## 7 BensofJura       16  7500 204.617
```

# DFBETAS for $X_2$

```
plot(dfbetas(races.lm)[,'Distance'],
  pch=23, bg='orange', cex=2,
  ylab="DFBETA (Climb)")
```



# DFBETAS for $X_2$

```
races.table[which(abs(dfbetas(races.lm)[,
  'Distance']) > 0.5),]
```

```
##           Race Distance Climb    Time
## 7  BensofJura       16  7500 204.617
## 11 LairigGhru       28  2100 192.667
```

# Outliers

- The essential definition of an *outlier* is an observation pair $(Y, X_1, \ldots, X_p)$ that does not follow the model, while most other observations seem to follow the model.
- Outlier in *predictors*: the $X$ values of the observation may lie outside the "cloud" of other $X$ values.
  - This means you may be extrapolating your model inappropriately.
  - The values $H_{ii}$ can be used to measure how "outlying" the $X$ values are.
- Outlier in *response*: the $Y$ value of the observation may lie very far from the fitted model.
  - If the studentized residuals are large: observation may be an outlier.

# Outliers

- The races at `Bens of Jura` and `Lairig Ghru` seem to be outliers in *predictors* as they were the highest and longest races, respectively.
- How can we tell if the `Knock Hill` result is an outlier?
  - It seems to have taken much longer than it should have so maybe it is an outlier in the *response*.

# Outlying $X$ values

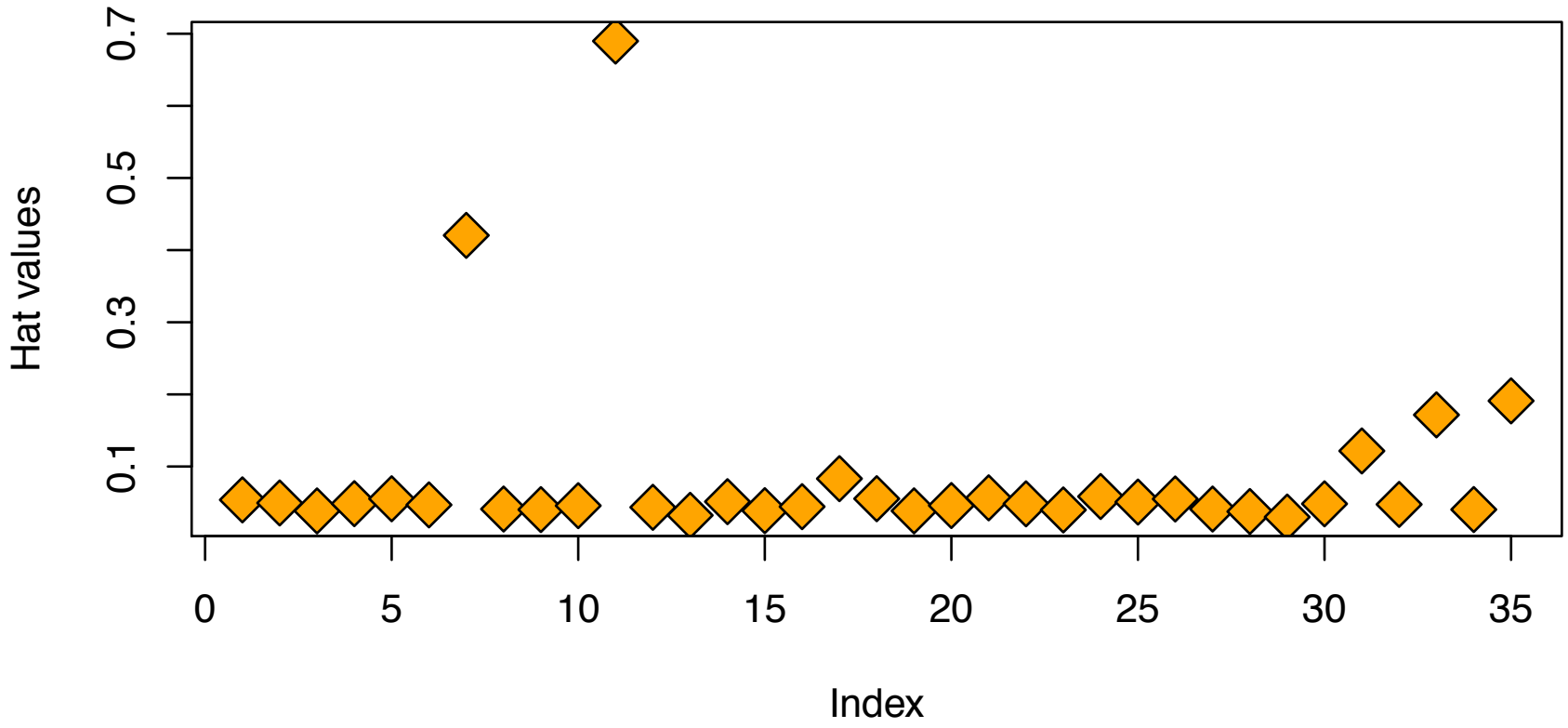
- One way to detect outliers in the *predictors*, besides just looking at the actual values themselves, is through their leverage values, defined by

$$\text{leverage}_i = H_{ii} = (X(X^T X)^{-1} X^T)_{ii}.$$

- Not surprisingly, our longest and highest courses show up again.
  - This at least reassures us that the leverage is capturing some of this "outlying in $X$ space".

# Outlying $X$ values

```
plot(hatvalues(races.lm), pch=23,
  bg='orange', cex=2, ylab='Hat values')
```



# Outlying $X$ values

```
races.table[which(hatvalues(races.lm) > 0.3),]
```

```
##           Race Distance Climb    Time
## 7  BensofJura       16  7500 204.617
## 11 LairigGhru       28  2100 192.667
```

# Outliers in the response

- We will consider a crude outlier test that tries to find residuals that are "larger" than they should be.
- Since `rstudent` are $t$ distributed, we could just compare them to the $T$ distribution and reject if their absolute value is too large.
- Doing this for every observation results in $n$ different hypothesis tests.
- This causes a problem: if $n$ is large, if we "threshold" at $t_{1-\alpha/2,n-p-2}$ we will get many outliers by chance even if model is correct.
- In fact, we expect to see $n \cdot \alpha$ "outliers" by this test. Every large data set would have outliers in it, even if model was entirely correct!

# Outliers in the response

- Let's sample some data from our model to convince ourselves that this is a real problem.

```
X = rnorm(100)
Y = 2 * X + 0.5 + rnorm(100)
alpha = 0.1
cutoff = qt(1 - alpha / 2, 97)
sum(abs(rstudent(lm(Y~X))) > cutoff)
```

```
## [1] 10
```

# Outliers in the response

```
# Bonferroni correction
X = rnorm(100)
Y = 2 * X + 0.5 + rnorm(100)
cutoff = qt(1 - (alpha / 100) / 2, 97)
sum(abs(rstudent(lm(Y~X))) > cutoff)
```

```
## [1] 0
```

# Multiple comparisons

- This problem we identified is known as *multiple comparisons* or *simultaneous inference.*
- When performing many tests (say $m$) each at level $\alpha$, we expect at least $\alpha m$ rejections even when *all* null hypotheses are true!
- In outlier detection, we are performing $m = n$ hypothesis tests, but might still like to control the probability of making *any* false positive errors.
- The reason we don't want to make errors here is that we don't want to throw away data unnecessarily.
- One solution: Bonferroni correction, threshold at $t_{1-\alpha/(2*n),n-p-2}$.

# Bonferroni correction

- Dividing $\alpha$ by $n$, the number of tests, is known as a *Bonferroni* correction.
- If we are doing many $t$ (or other) tests, say $m \gg 1$ we can control overall false positive rate at $\alpha$ by testing each one at level $\alpha/m$.
- In this case $m = n$, but other times we might look at a different number of tests.

# Bonferroni correction

- Essentially the *union bound* for probability.
- **Proof:** when the model is correct, with studentized residuals $T_i$:

$$\begin{aligned} P\left(\text{at least one false positive}\right) &= P\left(\cup_{i=1}^{m} |T_i| \geq t_{1-\alpha/(2*m),n-p-2}\right) \\ &\leq \sum_{i=1}^{m} P\left(|T_i| \geq t_{1-\alpha/(2*m),n-p-2}\right) \\ &= \sum_{i=1}^{m} \frac{\alpha}{m} = \alpha. \end{aligned}$$

- Let's apply this to our data. It turns out that `KnockHill` is a known error.

# Example (Bonferroni correction)

```
n = nrow(races.table)
cutoff = qt(1 - 0.05 / (2*n),
  (n - 4))
races.table[which(abs(rstudent(races.lm)) > cutoff),]
```

```
##          Race Distance Climb  Time
## 18 KnockHill        3   350 78.65
```

# Example (Bonferroni correction)

- The package car has a built in function to do this test.

```
library(car)
outlierTest(races.lm)
```

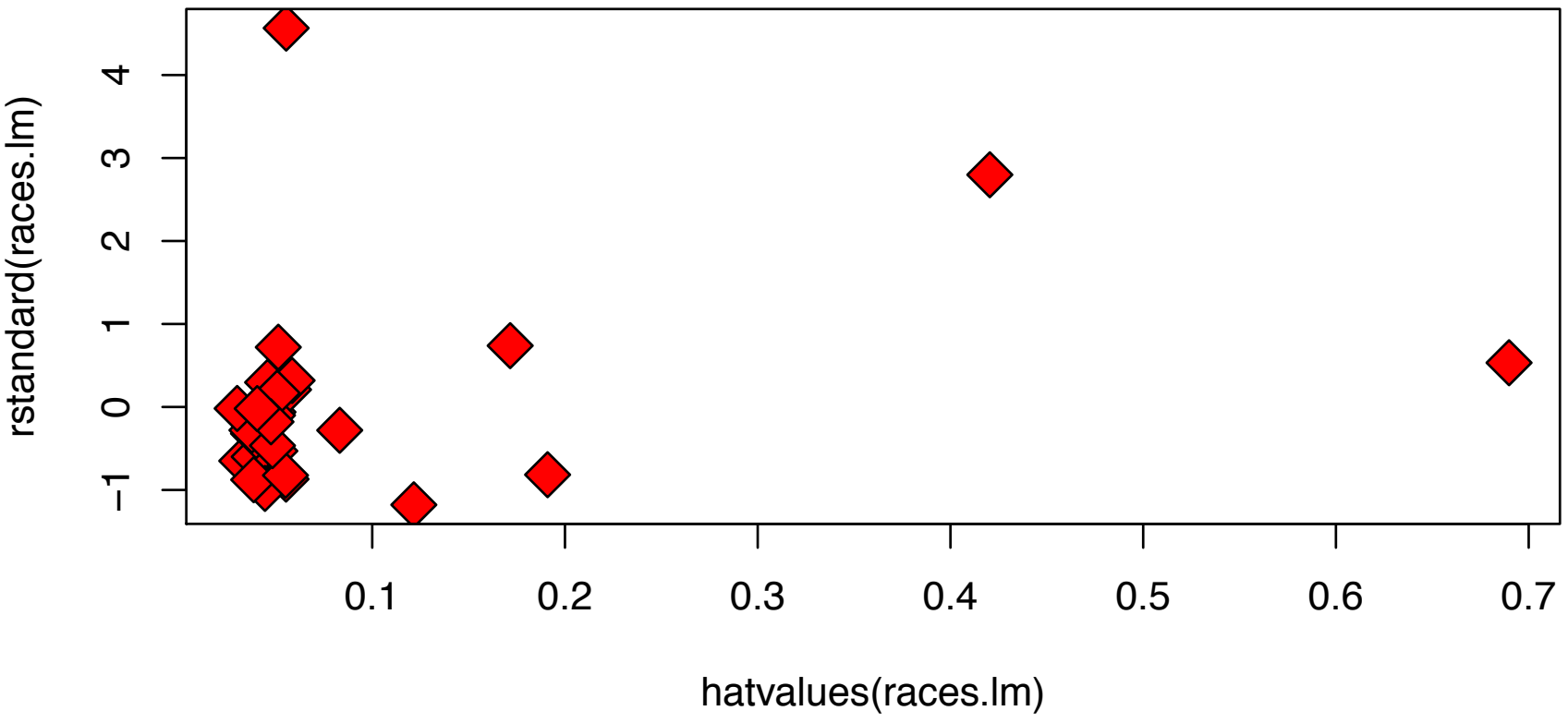
```
##    rstudent unadjusted p-value Bonferroni p
## 18 7.610845         1.3973e-08   4.8905e-07
```

# Influential observation - leverage

- The last plot that R produces is a plot of residuals against leverage.
- Points that have high leverage and large residuals are particularly influential.

# Example (leverage versus residuals)

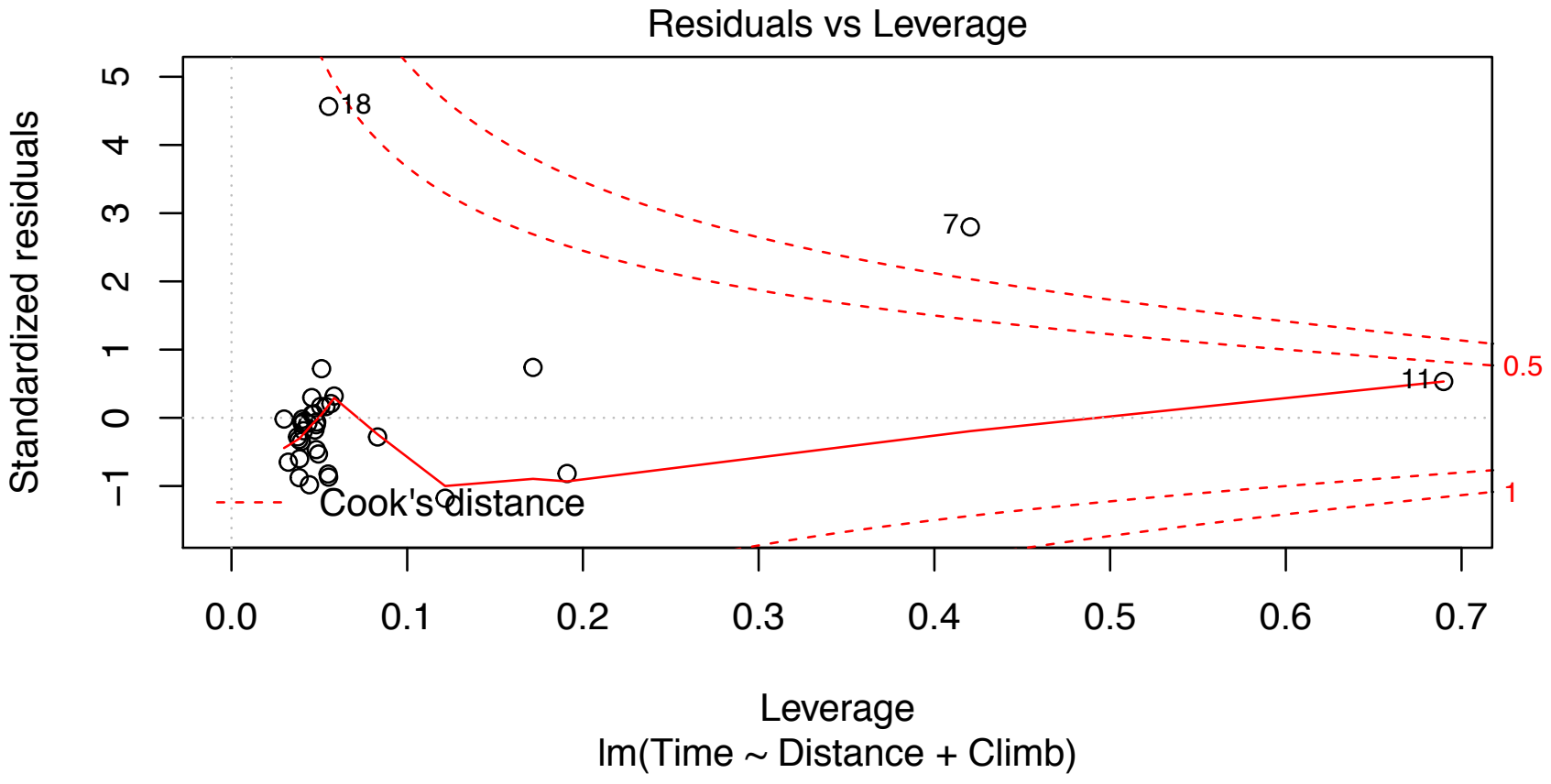
```
plot(hatvalues(races.lm), rstandard(races.lm),
  pch=23, bg='red', cex=2)
```



# Exampple (leverage versus residuals)

- R will put the IDs of cases that seem to be influential in these (and other plots).
  - Not surprisingly, we see our usual three suspects.

```
plot(races.lm, which=5)
```



# Influence measures

- As mentioned above, `R` has its own rules for flagging points as being influential.
- To see a summary of these, one can use the `influence.measures` function.

# Influence measures (in R)

```
influence.measures(races.lm)
```

| | dfb.1_ <dbl> | dfb.Dstn <dbl> | dfb.Clmb <dbl> | dffit <dbl> | cov.r <dbl> | cook.d <dbl> | hat <dbl> | inf <fctr> |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.037811462 | -0.0166142583 | -0.0047435625 | 0.038617999 | 1.15946791 | 5.127185e-04 | 0.05375572 | |
| 2 | -0.059579714 | 0.0672153961 | -0.0733958853 | -0.119560402 | 1.12694345 | 4.875401e-03 | 0.04946414 | |
| 3 | -0.048576860 | -0.0067065451 | 0.0280327646 | -0.063095302 | 1.13289525 | 1.365422e-03 | 0.03840444 | |
| 4 | -0.007664971 | -0.0056751901 | 0.0087636598 | -0.013674117 | 1.15557120 | 6.433010e-05 | 0.04848872 | |
| 5 | -0.050460528 | 0.0847092735 | -0.1450046113 | -0.209472340 | 1.08370625 | 1.474139e-02 | 0.05527121 | |
| 6 | 0.003484456 | -0.0043160647 | 0.0075759389 | 0.012209989 | 1.15360029 | 5.129264e-05 | 0.04680469 | |
| 7 | -0.890654684 | -0.7127735478 | 2.3646184862 | 2.699090776 | 0.81780209 | 1.893349e+00 | 0.42043463 | * |
| 8 | -0.008442784 | -0.0016484093 | 0.0055619075 | -0.011150263 | 1.14667200 | 4.277564e-05 | 0.04103328 | |
| 9 | -0.014368912 | 0.0009131396 | 0.0061606560 | -0.016631781 | 1.14533663 | 9.515950e-05 | 0.04025783 | |
| 10 | 0.047034115 | 0.0130569237 | -0.0365191836 | 0.063994414 | 1.14312971 | 1.405255e-03 | 0.04570891 | |
| 11 | -0.301182091 | 0.7687159937 | -0.4798493184 | 0.785688287 | 3.45248137 | 2.105214e-01 | 0.68981613 | * |
| 12 | -0.011491649 | 0.0096557210 | -0.0074877550 | -0.016715572 | 1.14921244 | 9.612212e-05 | 0.04345357 | |
| 13 | -0.031729063 | -0.0299106792 | -0.0007066754 | -0.117700687 | 1.09223183 | 4.703839e-03 | 0.03231875 | |
| 14 | 0.118031242 | 0.0420335396 | -0.1048840576 | 0.166101911 | 1.10391065 | 9.339448e-03 | 0.05126338 | |
| 15 | -0.100376388 | 0.0577007540 | -0.0223168727 | -0.119202733 | 1.10615460 | 4.834282e-03 | 0.03877135 | |
| 16 | -0.018520294 | 0.0067888268 | -0.0998617172 | -0.211352135 | 1.05013369 | 1.490749e-02 | 0.04436257 | |
| 17 | 0.011963729 | -0.0665049703 | 0.0344553620 | -0.083367689 | 1.19081472 | 2.385559e-03 | 0.08313942 | |
| 18 | 1.758274832 | -0.4065452697 | -0.6559341889 | 1.842374528 | 0.04932992 | 4.071560e-01 | 0.05535523 | * |
| 19 | -0.158890179 | 0.0443113962 | 0.0294135680 | -0.174838362 | 1.06346131 | 1.026539e-02 | 0.03850209 | |
| 20 | 0.008658369 | 0.0014243902 | -0.0059464022 | 0.011018523 | 1.15257413 | 4.177135e-05 | 0.04590867 | |
| 21 | 0.047765462 | -0.0100187391 | -0.0191985978 | 0.050317950 | 1.16113850 | 8.700051e-04 | 0.05657466 | |
| 22 | -0.018888912 | 0.0138562806 | -0.0064653159 | -0.022336402 | 1.15460132 | 1.716152e-04 | 0.04825780 | |
| 23 | -0.041306482 | 0.0340969664 | -0.0330224386 | -0.069613005 | 1.13261824 | 1.661162e-03 | 0.03977381 | |
| 24 | 0.074833295 | -0.0463850912 | 0.0064278105 | 0.078393718 | 1.15705550 | 2.107872e-03 | 0.05842537 | |
| 25 | 0.036911463 | -0.0126332955 | -0.0082568154 | 0.038084608 | 1.15566363 | 4.986386e-04 | 0.05072281 | |
| 26 | -0.137724315 | 0.1361238983 | -0.1013060816 | -0.197816078 | 1.09137481 | 1.317865e-02 | 0.05499644 | |
| 27 | -0.029204736 | -0.0057020716 | 0.0192393928 | -0.038570272 | 1.14314393 | 5.113116e-04 | 0.04103328 | |
| 28 | -0.047641080 | 0.0069360885 | 0.0149895347 | -0.054458683 | 1.13452136 | 1.017978e-03 | 0.03758135 | |
| 29 | -0.002137967 | 0.0006466224 | -0.0003281076 | -0.003091995 | 1.13382999 | 3.289579e-06 | 0.02992818 | |
| 30 | -0.085315881 | -0.0077051500 | 0.0548379624 | -0.103619059 | 1.13232031 | 3.669350e-03 | 0.04824732 | |
| 31 | 0.020993820 | 0.1701241625 | -0.3736338993 | -0.441381238 | 1.09600056 | 6.412250e-02 | 0.12158212 | |
| 32 | -0.028579099 | -0.0086935116 | 0.0232754469 | -0.039310491 | 1.15127772 | 5.311898e-04 | 0.04746275 | |
| 33 | -0.158227428 | 0.0970139844 | 0.1557016520 | 0.333844863 | 1.26094323 | 3.769491e-02 | 0.17158482 | |

1–33 of 35 rows

Previous 1 2 Next

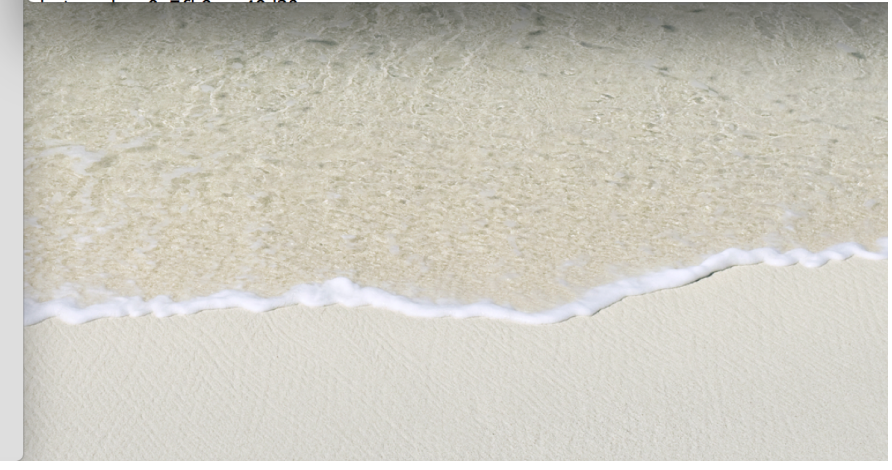
# Influence measures (in R)

- While not specified in the documentation, the meaning of the asterisks can be found by reading the code.
- The function `is.influential` makes the decisions to flag cases as influential or not.
- We see that the `DFBETAS` are thresholded at 1.
- We see that `DFFITS` is thresholded at $3 * sqrt((p+1)/(n-p-1))$.
- Etc.

# influence.measures() code

## `influence.measures`

```
function (model, infl = influence(model))
{
    is.influential <- function(infmat, n) {
        d <- dim(infmat)
        k <- d[[length(d)]] - 4L
        if (n <= k)
            stop("too few cases i with h_ii > 0), n < k")
        absmat <- abs(infmat)
        r <- if (is.matrix(infmat)) {
            cbind(absmat[, 1L:k] > 1, absmat[, k + 1] > 3 * sqrt(k/(n -
                k)), abs(1 - infmat[, k + 2]) > (3 * k)/(n -
                k), pf(infmat[, k + 3], k, n - k) > 0.5, infmat[,
                k + 4] > (3 * k)/n)
        }
        else {
            c(absmat[, , 1L:k] > 1, absmat[, , k + 1] > 3 * sqrt(k/(n -
                k)), abs(1 - infmat[, , k + 2]) > (3 * k)/(n -
                k), pf(infmat[, , k + 3], k, n - k) > 0.5, infmat[,
                , k + 4] > (3 * k)/n)
        }
        attributes(r) <- attributes(infmat)
        r
    }
    p <- model$rank
    e <- weighted.residuals(model)
    s <- sqrt(sum(e^2, na.rm = TRUE)/df.residual(model))
    mqr <- qr.lm(model)
    xxi <- chol2inv(mqr$qr, mqr$rank)
    si <- infl$sigma
    h <- infl$hat
    is.mlm <- is.matrix(e)
    cf <- if (is.mlm)
        aperm(infl$coefficients, c(1L, 3:2))
    else infl$coefficients
    dfbetas <- cf/outer(infl$sigma, sqrt(diag(xxi)))
    vn <- variable.names(model)
    vn[vn == "(Intercept)"] <- "1_"
    dimnames(dfbetas)[[length(dim(dfbetas))]] <- paste0("dfb.",
        abbreviate(vn))
    dffits <- e * sqrt(h)/(si * (1 - h))
    if (any(ii <- is.infinite(dffits)))
        dffits[ii] <- NaN
    cov.ratio <- (si/s)^(2 * p)/(1 - h)
    cooks.d <- if (inherits(model, "glm"))
        (infl$pear.res/(1 - h))^2 * h/(summary(model)$dispersion *
            p)
    else ((e/(s * (1 - h)))^2 * h)/p
    infmat <- if (is.mlm) {
        dns <- dimnames(dfbetas)
        dns[[3]] <- c(dns[[3]], "dffit", "cov.r", "cook.d", "hat")
        a <- array(dfbetas, dim = dim(dfbetas) + c(0, 0, 3 +
            1), dimnames = dns)
        a[, , "dffit"] <- dffits
        a[, , "cov.r"] <- cov.ratio
        a[, , "cook.d"] <- cooks.d
        a[, , "hat"] <- h
        a
    }
    else {
        cbind(dfbetas, dffit = dffits, cov.r = cov.ratio, cook.d = cooks.d,
            hat = h)
    }
    infmat[is.infinite(infmat)] <- NaN
    is.inf <- is.influential(infmat, sum(h > 0))
    ans <- list(infmat = infmat, is.inf = is.inf, call = model$call)
    class(ans) <- "infl"
    ans
}
```



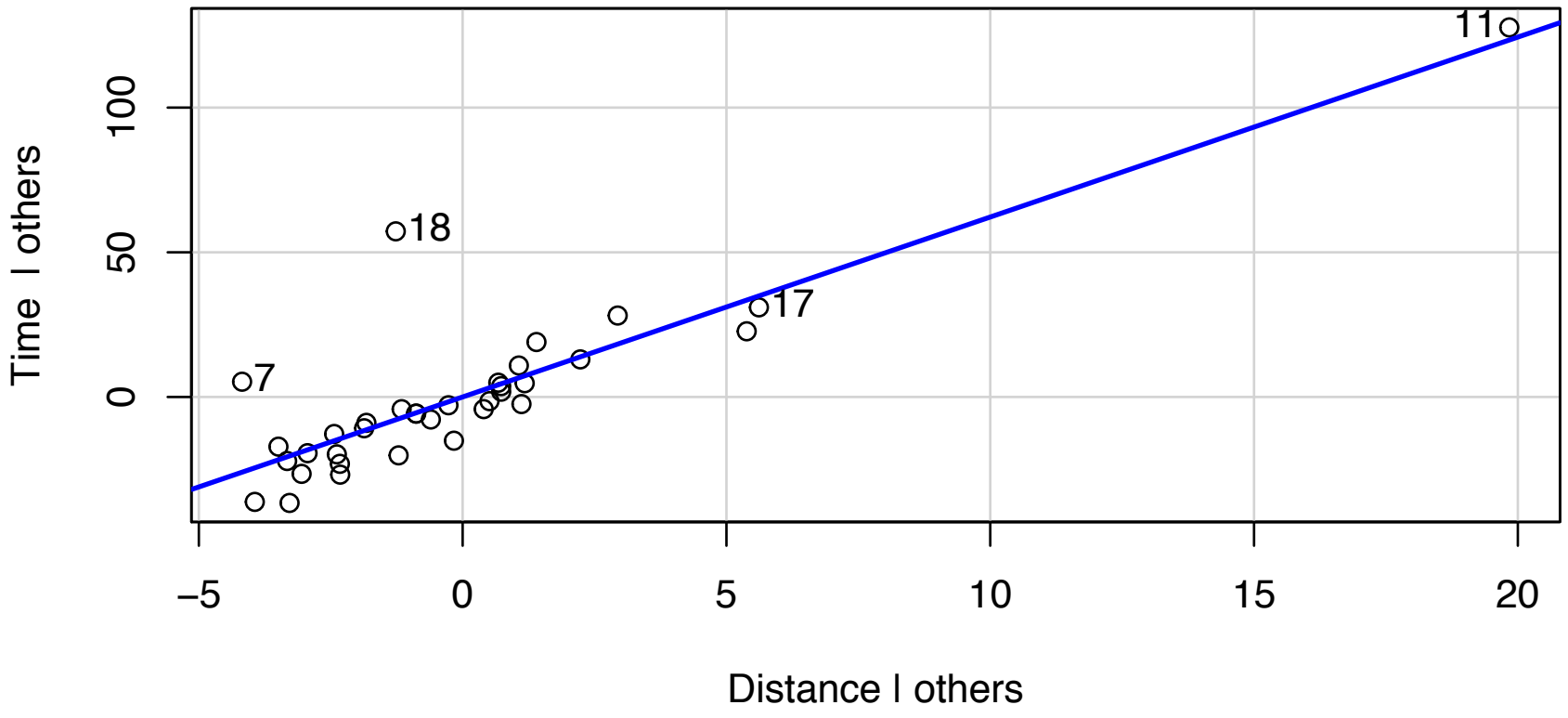
# Problems in the regression function

# Added variable plots

- The plots can be helpful for finding influential points, outliers.
- The functions can be found in the `car` package.
- Procedure:
  - Let $\tilde{e}_{X_j,i}, 1 \leq i \leq n$ be the residuals after regressing $X_j$ onto all columns of $X$ except $X_j$;
  - Let $e_{X_j,i}$ be the residuals after regressing $Y$ onto all columns of $X$ except $X_j$;
  - Plot $\tilde{e}_{X_j}$ against $e_{X_j}$.
  - If the (partial regression) relationship is linear this plot should look linear.

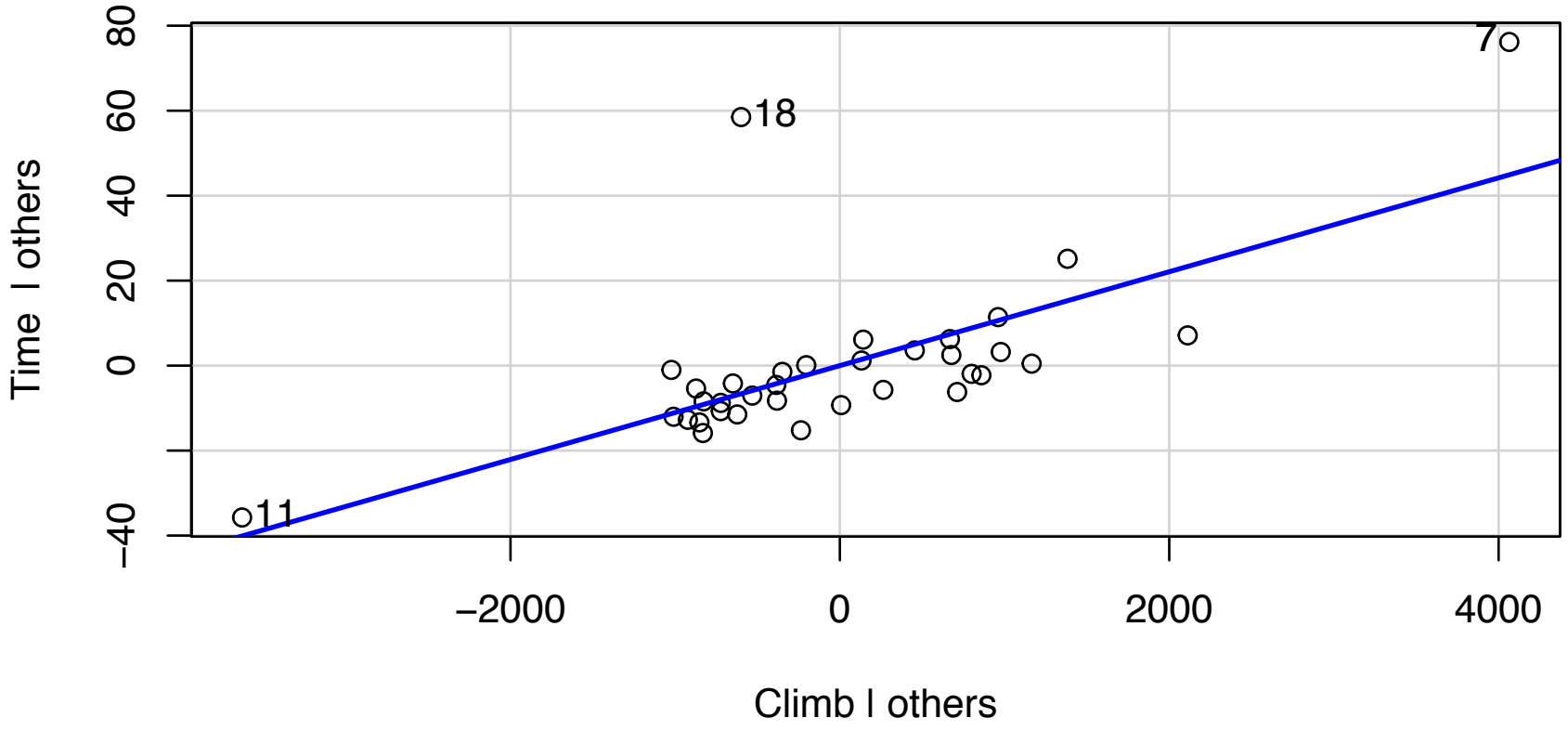
# Example (Added variable plots)

```
avPlots(races.lm, 'Distance')
```



# Example (Added variable plots)

```
avPlots(races.lm, 'Climb')
```

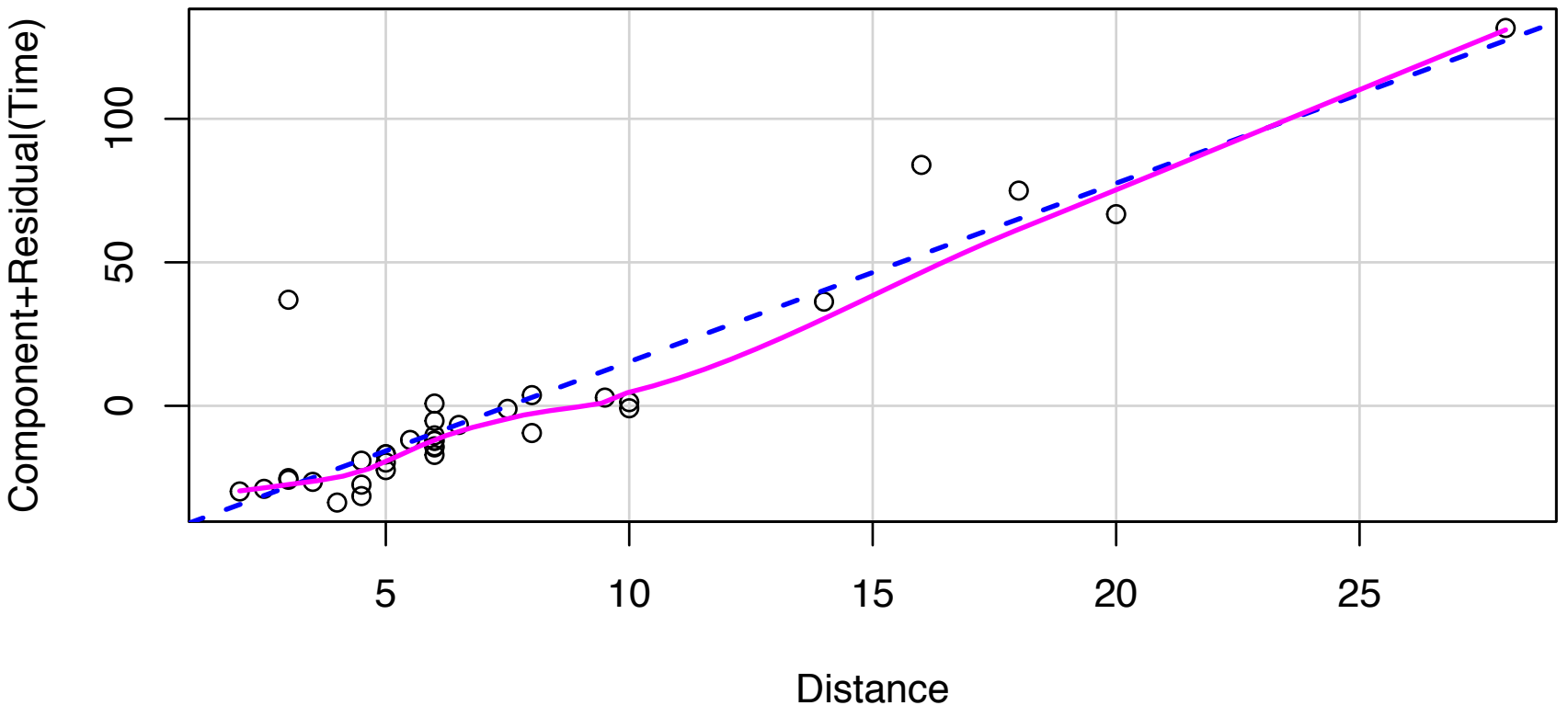


# Component + residual plots

- Similar to added variable, but may be more helpful in identifying nonlinear relationships.
- Procedure: plot $X_{ij}, 1 \leq i \leq n$ vs. $e_i + \widehat{\beta}_j \cdot X_{ij}, 1 \leq i \leq n$.
- The violet line is a non-parametric smooth of the scatter plot that may suggest relationships other than linear.

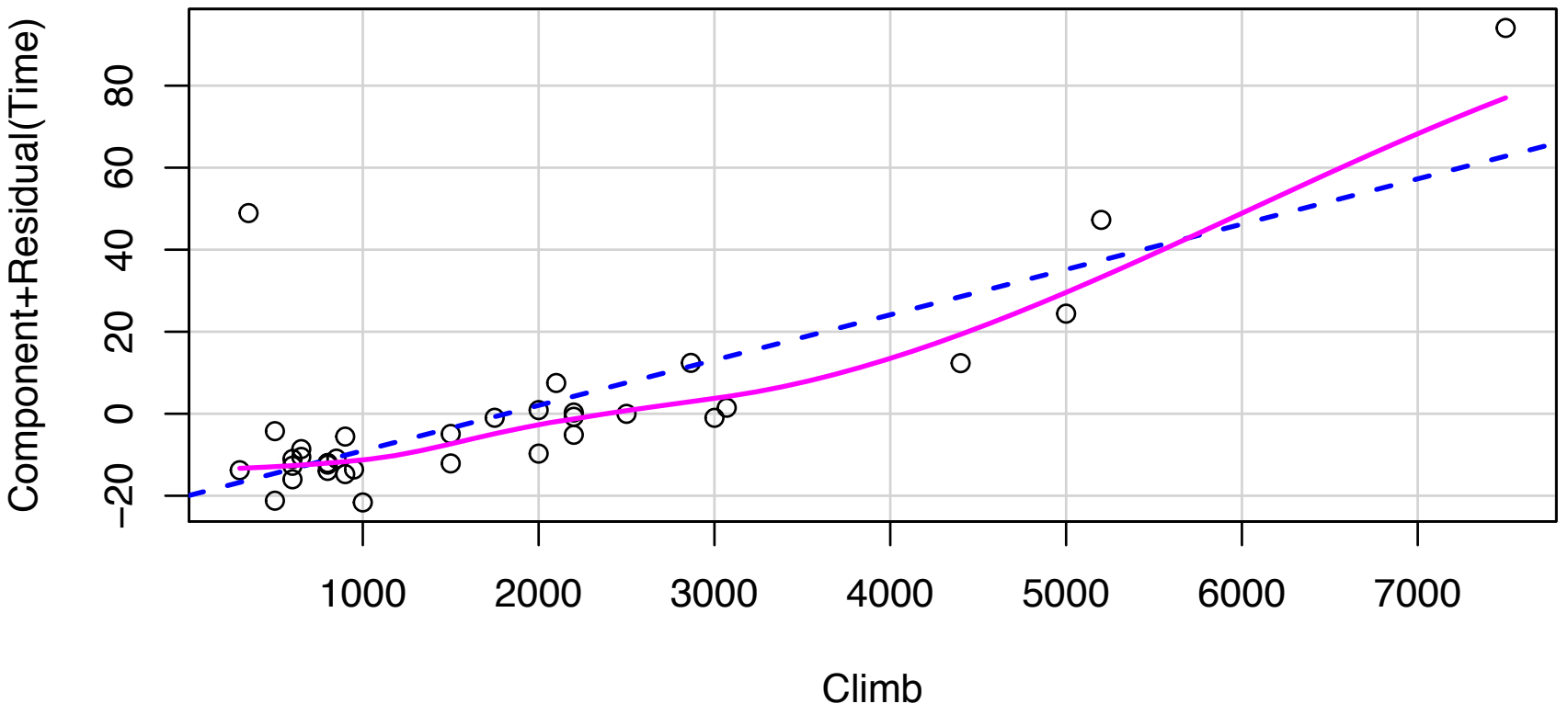
# Example (Component + residual plots)

```
crPlots(races.lm, 'Distance')
```



# Example (Component + residual plots)

```
crPlots(races.lm, 'Climb')
```



# Reference

- **CH**: Chapter 4.
- Lecture notes of Jonathan Taylor .