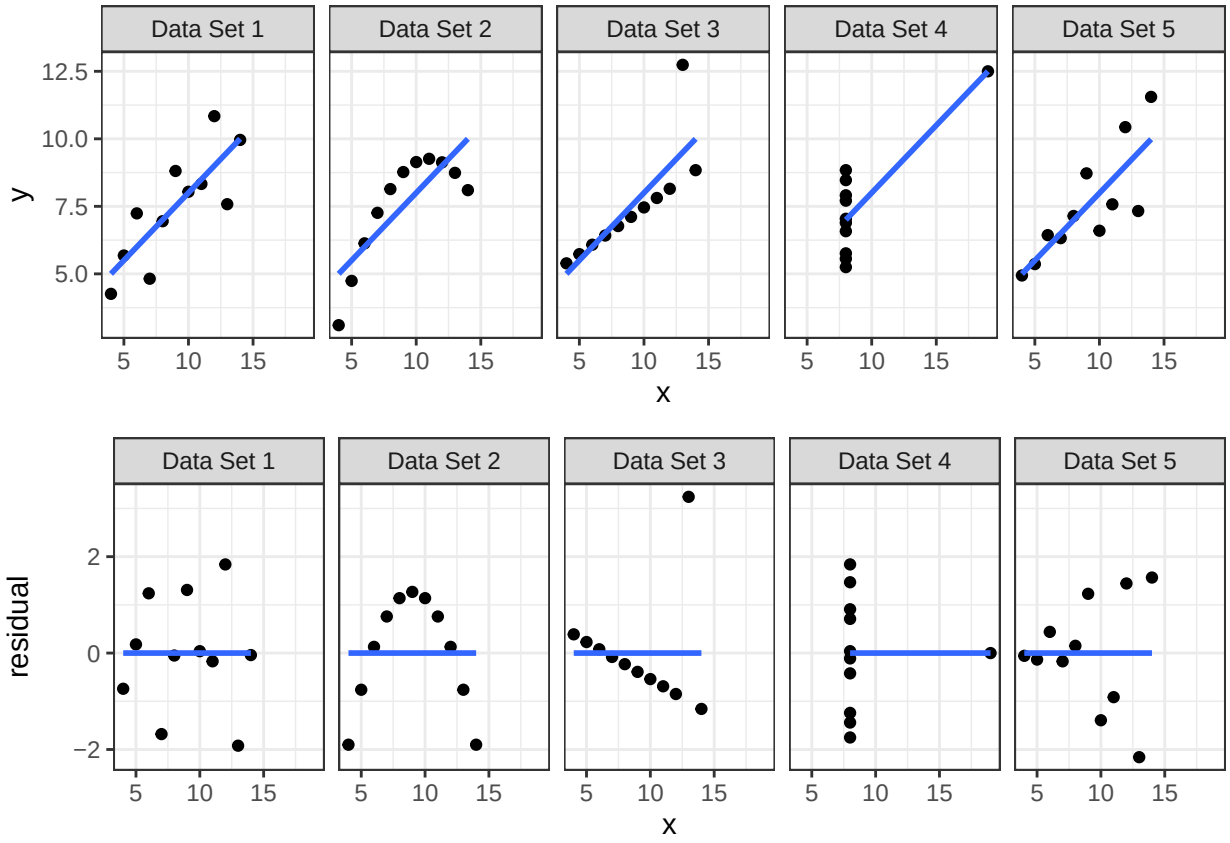
Chapter 11: Outliers and Influential Observations

Case-influence statistics

- Least squares regression can be heavily impacted by outliers and influential observations, so it is important to identify such cases and deal with them appropriately, possibly through a transformation
- Sleuth3 defines as "numerical measures associated with the individual influence of each observation (each case)."
- Can help (1) identify influential observations that may not be apparent from a graphical inspection and (2) assign the overall influence of a case into what is unusual about its x value and what is unusual about its y value relative to the fitted model
- Statistics:
  - *leverage* for case i: $h_i = \frac{1}{(n-1)}\left[\frac{x_i - \bar{x}}{s_X}\right]^2 + \frac{1}{n}$; a measure of the distance of case i from the average
    - if $h_i > 2p/n$, flag case as having high potential for excessive influence
  - *studentized residual* for case i: $studres_i = \frac{res_i}{\hat{\sigma}\sqrt{1-h_i}}$; a residual divided by its estimated standard deviation
    - Some residuals naturally have less variation because of their leverages (h_i). In order to compare them and eliminate the leverage effect, we need to standardize them to put them on a standard scale: numbers of standard deviations. Roughly 95% of normally distributed observations fall within about 2 standard deviations of their mean (recall 68-99-99.7 Rule, also know as the Empirical Rule), so it is common to investigate studentized residuals that are larger than $|2|$.
  - *Cook's distance* for case i: $D_i = \sum_{j=1}^{n} \frac{(\hat{Y}_{j(i)} - \hat{Y}_j)^2}{p\hat{\sigma}^2}$; measures overall influence, meaning the effect omitting a particular case i has on the estimated regression coefficients
    - $\hat{Y}_j$: j^{th} fitted value in a fit using all cases
    - $\hat{Y}_{j(i)}$: j^{th} fitted value in a fit that excludes case i
    - p: number of regression parameters
    - $\hat{\sigma}^2$: estimated variance from the fit with all observations included
    - May investigate cases where $D_i \geq 1$

Anscombe's Data

Anscombe's quintet comprises five data sets that have nearly identical summary statistics ($\bar{x}$, s_x, $\bar{y}$, s_y, correlation between x and y (r), linear regression line, and coefficient of determination (R^2)), but look very different when they are plotted (see below). Statistician Francis Anscombe constructed these data sets in 1973 to illustrate the importance of graphing data before analyzing it and to demonstrate the effect of outliers and other influential points (the subject of this lecture) on statistical properties.



- For today, let's focus on Data Sets 3 and 4. We will see how to identify the problematic observations from the diagnostics.
- In data set 3, observation 3 is the one with a big Y!

```
anscombe$y3[3]
```

```
## [1] 12.74
```

- In data set 4, observation 8 is the one with a big X!

```
anscombe$x4[8]
```

```
## [1] 19
```

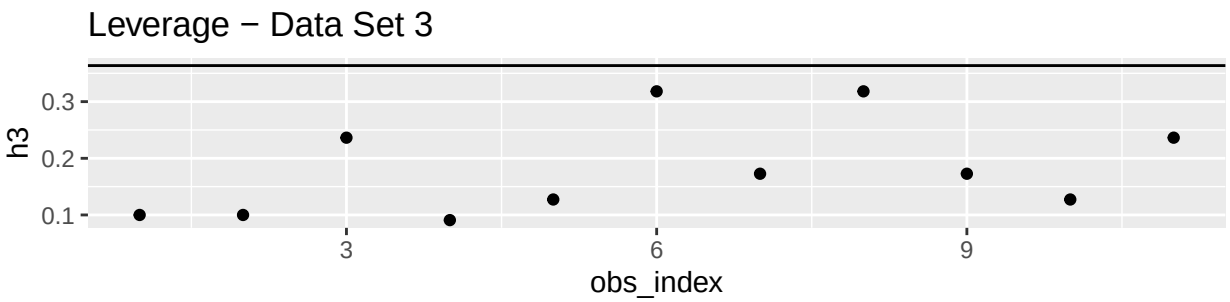
Data Set 3

- Every statistical software package will give you different plots by default. Here is my preferred option:

```
fit3 <- lm(y3 ~ x3, data = anscombe)
anscombe <- anscombe %>%
  mutate(
    obs_index = row_number(),
    h3 = hatvalues(fit3),
    studres3 = rstudent(fit3),
    D3 = cooks.distance(fit3)
  )
# 2p/n; p = 2 since we have beta_0 and beta_1 in our simple linear regression model
2 * 2 / nrow(anscombe)
```

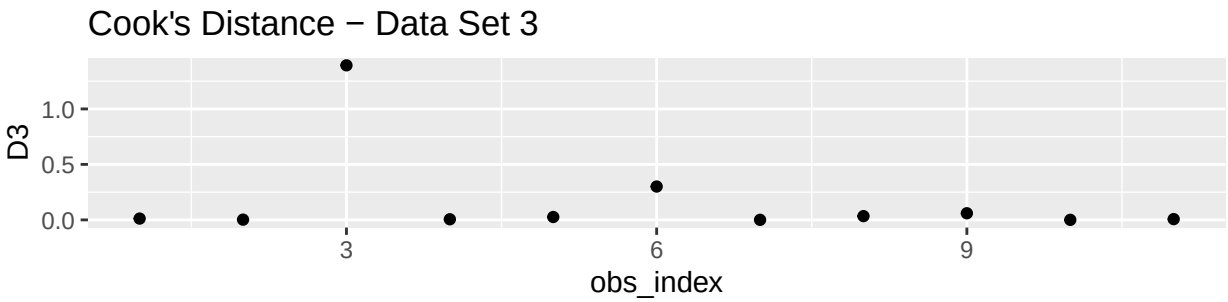
```
## [1] 0.3636364
```

```
ggplot(data = anscombe, mapping = aes(x = obs_index, y = h3)) +
  geom_point() +
  geom_hline(yintercept = 2 * 2 / nrow(anscombe)) +
  ggtitle("Leverage - Data Set 3")
```




```
ggplot(data = anscombe, mapping = aes(x = obs_index, y = studres3)) +
  geom_point() +
  ggtitle("Studentized Residuals - Data Set 3")
```


```
ggplot(data = anscombe, mapping = aes(x = obs_index, y = D3)) +
  geom_point() +
  ggtitle("Cook's Distance - Data Set 3")
```

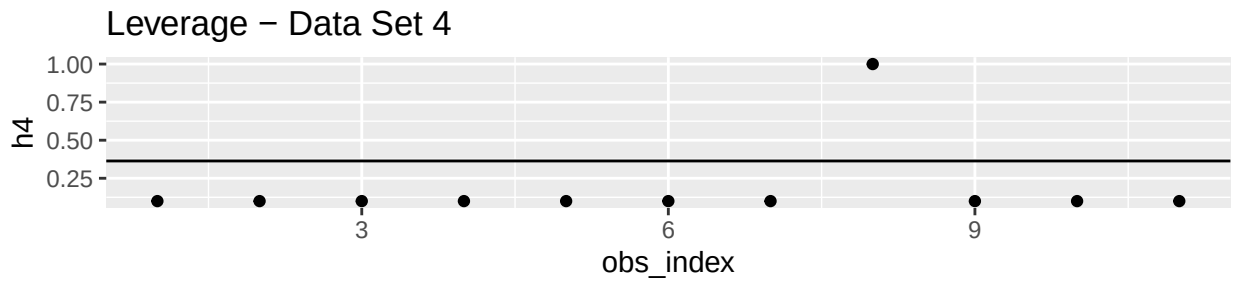

Cook's Distance − Data Set 3
D3
1.0
0.5
0.0
3
6
9
obs_index

Data Set 4

```
fit4 <- lm(y4 ~ x4, data = anscombe)
anscombe <- anscombe %>%
  mutate(
    obs_index = row_number(),
    h4 = hatvalues(fit4),
    studres4 = rstudent(fit4),
    D4 = cooks.distance(fit4)
  )
# 2p/n; p = 2 since we have beta_0 and beta_1 in our simple linear regression model
2 * 2 / nrow(anscombe)
```

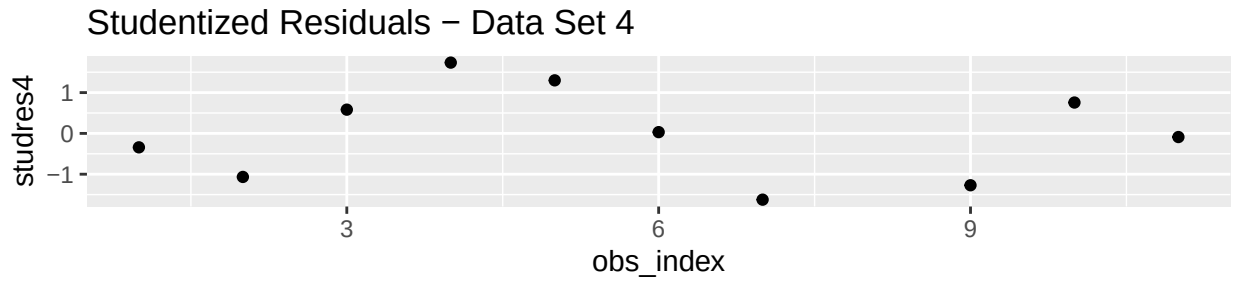
```
## [1] 0.3636364
```

```
ggplot(data = anscombe, mapping = aes(x = obs_index, y = h4)) +
  geom_point() +
  geom_hline(yintercept = 2 * 2 / nrow(anscombe)) +
  ggtitle("Leverage - Data Set 4")
```



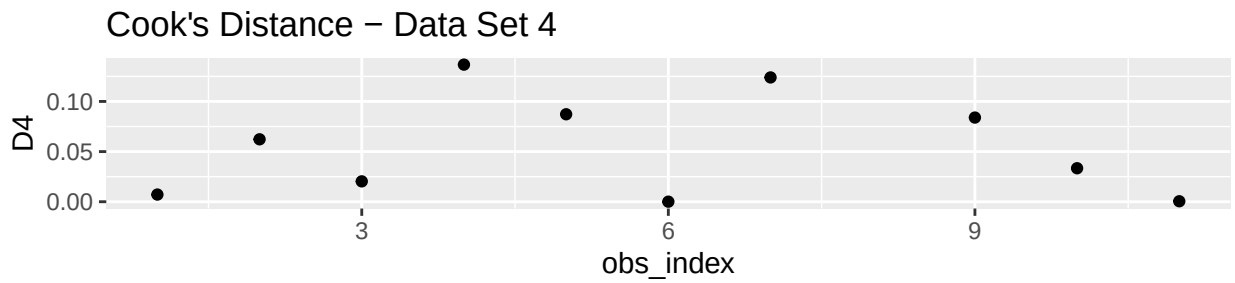

```
ggplot(data = anscombe, mapping = aes(x = obs_index, y = studres4)) +
  geom_point() +
  ggtitle("Studentized Residuals - Data Set 4")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(data = anscombe, mapping = aes(x = obs_index, y = D4)) +
  geom_point() +
  ggtitle("Cook's Distance - Data Set 4")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



The lower two plots look OK… what's up with that warning?

```
anscombe$h4
```

```
##   1   2   3   4   5   6   7   8   9  10  11
## 0.1 0.1 0.1 0.1 0.1 0.1 0.1 1.0 0.1 0.1 0.1
```

```
anscombe$studres4
```

```
##           1           2           3           4           5           6
## -0.34104165 -1.06669299  0.58216636  1.73514504  1.30031318  0.03136768
##           7           8           9          10          11
## -1.62381807         NaN -1.27046922  0.75677904 -0.08931624
```

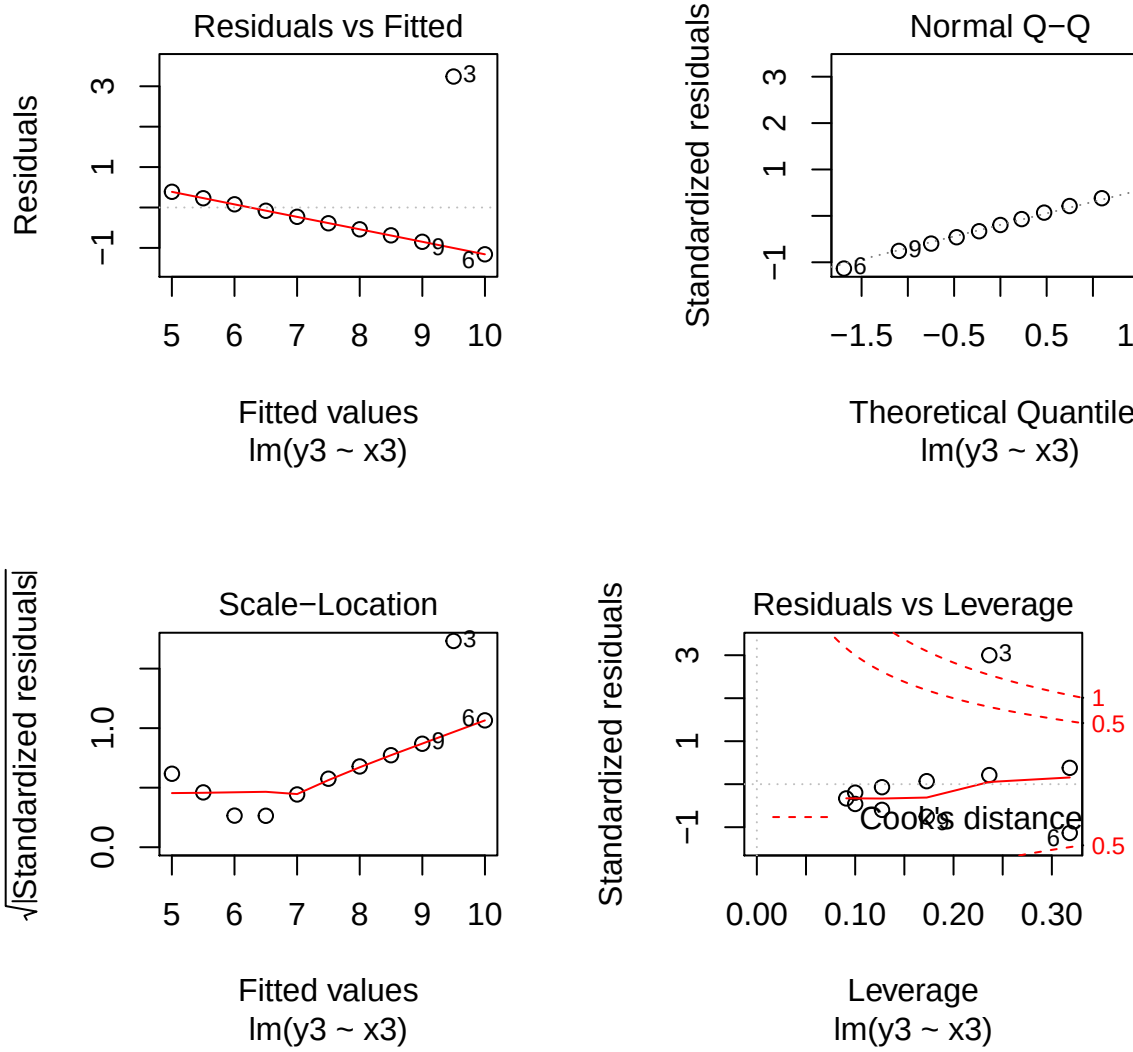
```
anscombe$D4
```

```
##            1            2            3            4            5
## 7.165166e-03 6.225950e-02 2.032144e-02 1.367179e-01 8.723799e-02
##            6            7            8            9           10
## 6.148813e-05 1.239465e-01          NaN 8.394407e-02 3.340334e-02
##           11
## 4.980902e-04
```

R Code: Default Plots

You can get a set of different diagnostic plots more easily, but I find the plot involving Cook's distance and Leverage less intuitive:

```
plot(fit3)
```



Note: to get the plots to all show up in the knitted pdf, I had to set figure height and width in the code chunk declaration:

'`markdown{r, fig.height = 4, fig.width = 4}`