Lecture 6- Predictions and Leverage

Peter Shaffery

2/2/2021

Prediction Intervals

Recall our basic MLR model for a single data point:

$$y_i = \vec{x_i}^T \vec{\beta} + \epsilon_i$$

Where by convention the first element of $\vec{x_i}$ is 1.

Let's say that we have fit our model by obtaining $\vec{\beta}$, but now we want to predict the value of y at a previously unobserved vector of independent variables, $\vec{x}_n$:

$$\hat{y_n} = \vec{x_n}^T \hat{\beta}$$

But what is the uncertainty in $\hat{y}_n$? How far off can we expect our prediction $\hat{y}_n$ to be from future observations?

Conceptual question: are the above two questions asking the same thing?

We've spent a lot of time thinking about uncertainty in $\vec{\beta}$, recall from last lecture:

$$\text{Var}[\hat{\beta}] = \sigma^2 (X^T X)^{-1}$$

You might also remember that $\hat{\beta}$ was Multivariate Normally Distributed,$\hat{\beta} \sim MVN(\vec{\beta}, \sigma^2 (X^T X)^{-1})$ and thus, by the properties of Multivariate Normals:

$$\hat{y}_n = \vec{x_n}^T \hat{\beta} \sim MVN(\vec{x_n}^T \vec{\beta}, \sigma^2 x_n^T (X^T X)^{-1} x_n)$$

Note two things:

1. Even though $\hat{y_n}$ is "Multivariate Normally Distributed", in this case the "Multivariate" dimension is 1, so $\hat{y_n}$ is just regular Normal distributed (and going forward I will not make this distinction explicitly)
2. The factor $x_n^T (X^T X)^{-1} x_n$ is very similar to something we saw Thursday. Can you think of where you might have seen something like this?

Thinking.

Thinking..

Thinking...

Thinking....

Thinking.....

It's *very* similar to our hat matrix $H = X(X^T)^{-1} X^T$

Indeed, we could have gone through this same process for $(y_i, \vec{x_i})$ and showed that:

$$\text{Var}[\hat{y}_i] = \sigma^2 x_i^T (X^T X)^{-1} x_i = \sigma^2 H_{ii}$$

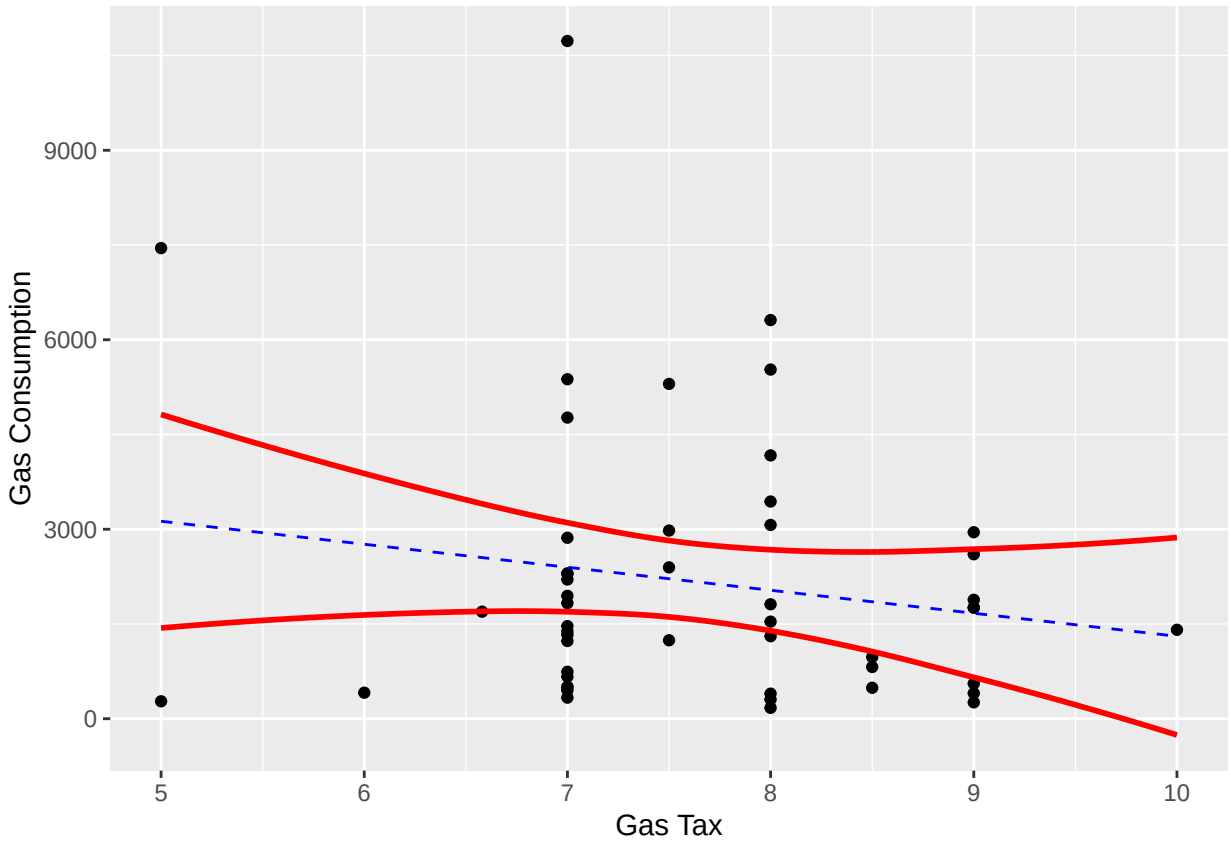
For a given choice of significance α, we can now compute "confidence intervals" for $\hat{y}_i$ (or $\hat{y}_n$):

$$CI_{\hat{y}_i} = [\hat{y}_i - t_{\alpha/2,n-2}s_{\hat{y}_i}, \hat{y}_i - t_{\alpha/2,n-2}s_{\hat{y}_i}]$$

Recalling that $t_{\alpha/2,n-2}$ is the $1-\alpha/2$ percent quantile of the Student T distribution with $n-2$ degrees of freedom, and $s_{\hat{y}_i} = \sqrt{\text{Var}[\hat{y}_i]}$ is the standard error

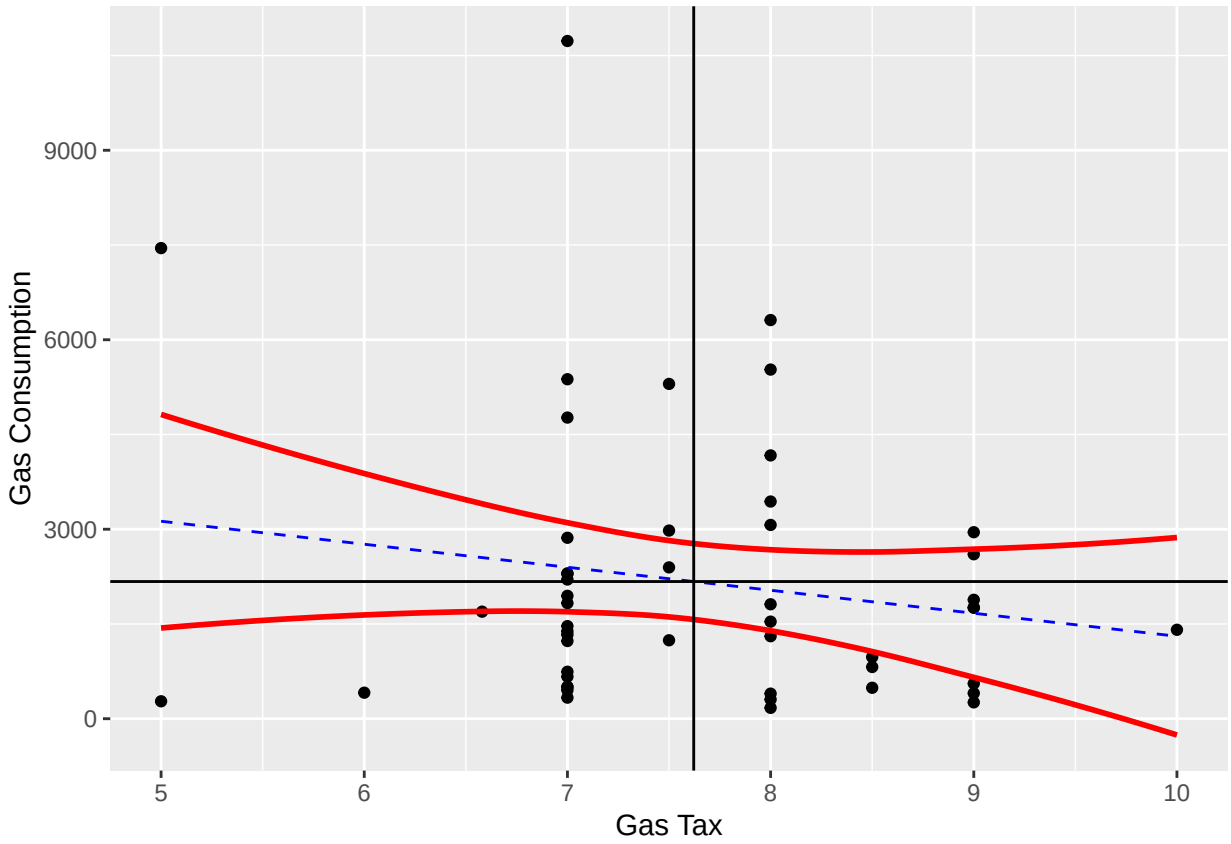
Let's go back to SLR so we can visualize what these prediction intervals look like

```
library(tidyverse)
library(magrittr)
library(pracma)

fuel = read.csv('../../data/fuel.csv') %>% drop_na

n = fuel %>% nrow
y = fuel$gas
X = fuel %>%
  select(c('tax')) %>%
  as.matrix %>%
  cbind(rep(1,n),.)
beta.hat = inv(t(X)%*%X)%*%t(X)
H = X%*%beta.hat
y.hat = H%*%y

resids = (diag(n)-H)%*%y
SSE = sum(resids^2)
sigma2.hat = SSE/(n-2)
var.y.hat = sigma2.hat*diag(H)
y.se = sqrt(var.y.hat)
y.ci = qt(.975,n-2)*y.se

plt.df = data.frame('gas'=y,
                    'tax'=fuel$tax,
                    'best_fit'=y.hat,
                    'best_fit_se'=y.se,
                    'y_hat_upper' = y.hat+y.ci,
                    'y_hat_lower' = y.hat-y.ci
                    )
plt = ggplot(plt.df,aes(x=tax)) +
  geom_point(aes(y=gas)) +
  geom_line(aes(y=best_fit),color='blue', linetype='dashed') +
  geom_smooth(aes(y=y_hat_upper), color='red') +
  geom_smooth(aes(y=y_hat_lower), color='red') +
  labs(x='Gas Tax',y='Gas Consumption')

plt
```



Observe that the width of the CIs changes with x, why is this?

```
plt + geom_hline(yintercept=mean(y)) + geom_vline(xintercept=mean(X[,2]))
```



Well, recall how we got to this point: we accounted for uncertainty in the slope parameter β_1. As the slope increases or decreases smoothly across it's CI, the line of best fit will sweep out a curve.

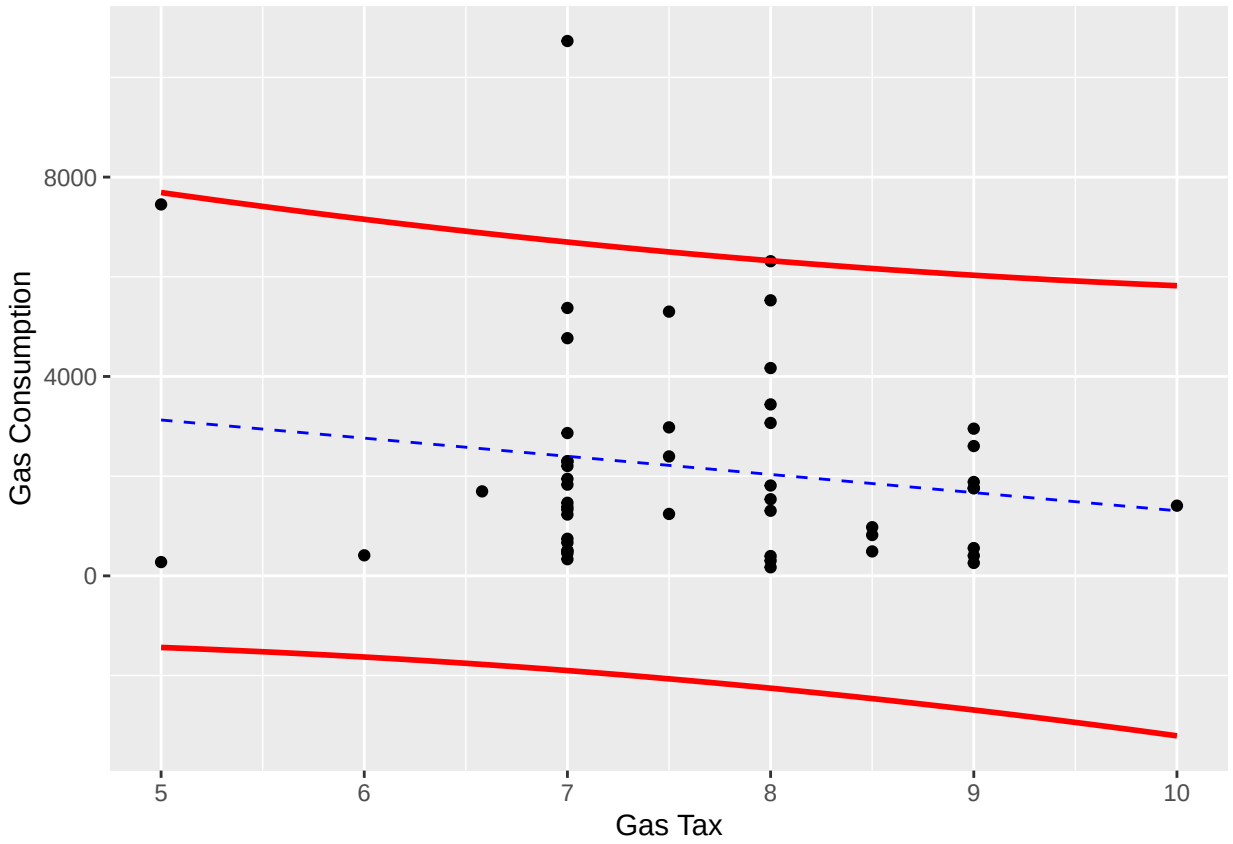
Put another way, predictions made far outside our dataset will have more uncertainty than those in the middle of the dataset

One final point: we have so far only account for uncertainty in $\hat{y_n}$, but this does not account for the *natural randomness* in y. A simple way to do this would be to use the standard error given:

$$\tilde{s}_{y_n} = \sqrt{\text{Var}[\hat{y_n}] + \hat{\sigma^2}}$$

This would give us the following plot:

```
y.se = sqrt(var.y.hat + sigma2.hat)
y.ci = qt(.975,n-2)*y.se

plt.df['y_hat_upper'] = y.hat+y.ci
plt.df['y_hat_lower'] = y.hat-y.ci

plt = ggplot(plt.df,aes(x=tax)) +
  geom_point(aes(y=gas)) +
  geom_line(aes(y=best_fit),color='blue', linetype='dashed') +
  geom_smooth(aes(y=y_hat_upper), color='red') +
  geom_smooth(aes(y=y_hat_lower), color='red') +
  labs(x='Gas Tax',y='Gas Consumption')

plt
```



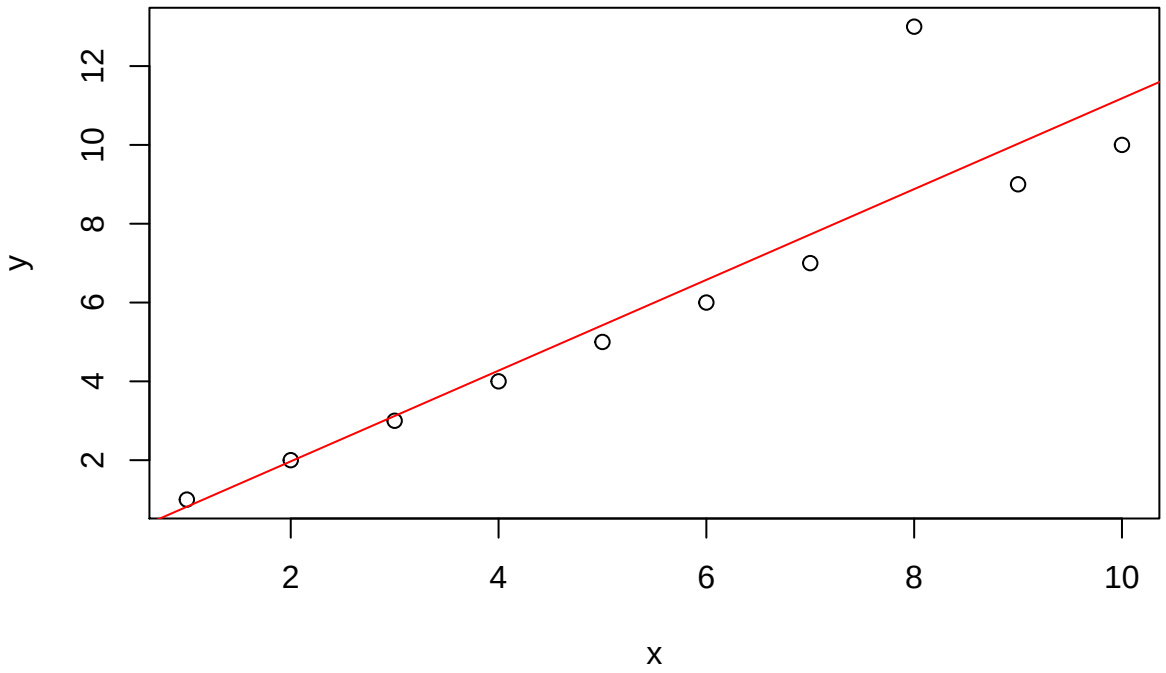
This prediction interval adequately captures the variability of the data, but is so wide that it's not useful. Typically you will just the first type of prediction interval.

Leverage and Outliers

Outliers

Let's look at a **bad** model:

```
n=10
x = 1:n
y = x
y[8] = y[8]+5
mod = lm(y~x)
plot(x,y)
abline(mod,col='red')
```




```
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1818 -0.6894 -0.3485 -0.0076  4.1212
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3333     1.0964  -0.304 0.768859
## x             1.1515     0.1767   6.517 0.000185 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.605 on 8 degrees of freedom
## Multiple R-squared:  0.8415, Adjusted R-squared:  0.8217
## F-statistic: 42.47 on 1 and 8 DF,  p-value: 0.0001848
```

From a statistics perspective this model might seem rather good, high R^2 and low p-value. But from a "just look at the freaking plot!" perspective, the model is terrible. If you cover up the point at $x = 8$ then it's clear that the relationship between x and y is perfectly linear, with unit slope.

While this isn't a "problem" in a statistical sense, from the perspective of an applied modeler it's undesirable. Not only are missing the obvious truth, but we're also getting way worse SSE than we could have without the wonky datapoint (called an **outlier**):

```
n=10
x = 1:n
y = x
y[8] = y[8]+5
resids.outlier = lm(y~x) %>% resid
```

```
SSE.outlier = sum(resids.outlier^2)

resids.no.outlier = lm(y[-8]~x[-8]) %>% resid
SSE.no.outlier = sum(resids.no.outlier^2)

print(c(SSE.outlier,SSE.no.outlier))
```

```
## [1] 2.060606e+01 2.034552e-30
```

We have already seen a quantitative measure of outlier-ness: *standardized residuals*.

Recalling that in MLR

$$\hat{\epsilon} = (I - H)\vec{y}$$

You can see that (similar to $\hat{y}_i$):

$$\text{Var}[\hat{\epsilon}_i] = \sigma^2(1 - H_{ii})$$

Thus we can scale the residuals to have variance 1 by:

$$\tilde{\epsilon}_i = \frac{\hat{\epsilon}}{\sigma\sqrt{1 - H_{ii}}}$$

Thus when $\tilde{\epsilon_i}$ are "large" (roughly > 2), we might consider them to be outliers in the y-direction.

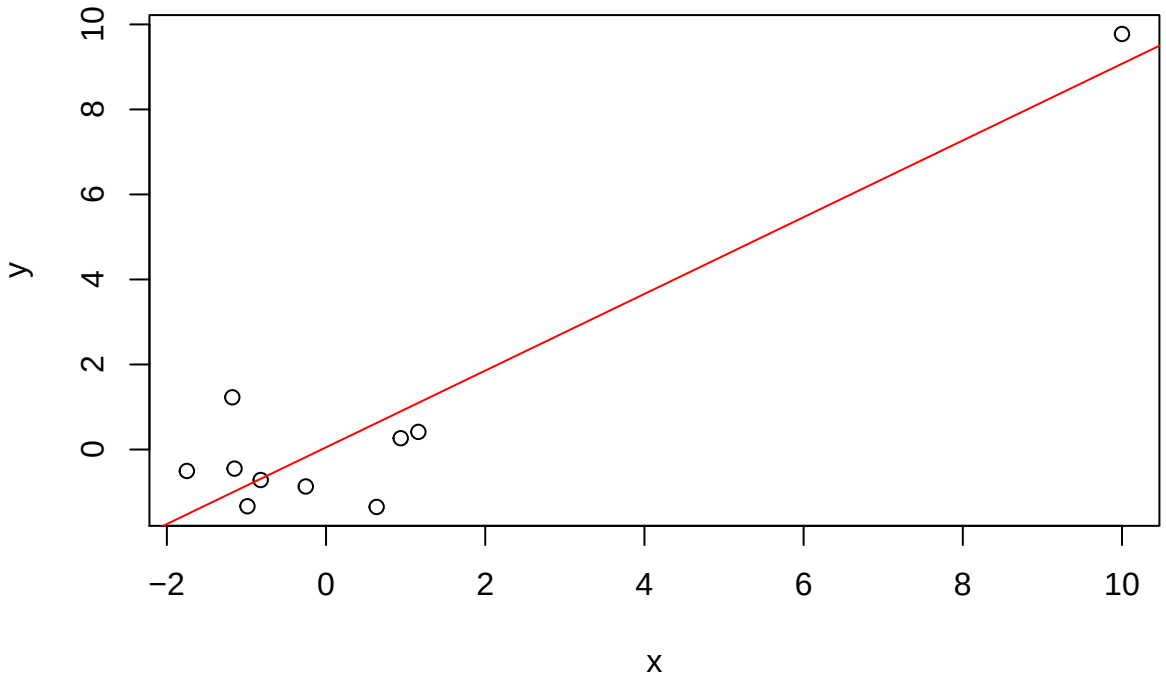
Generally outliers in the y-direction are not seen as a huge problem. It is recommended that if you believe that your data contains outliers, you perform the regression with and without them and present both results.

On the other hand, outliers in the x-direction can be a *major* problem

Leverage

Let's look at a **terrible** model:

```
n=10
x = rnorm(n-1)
x[n] = 10
y = rnorm(n-1)
y[n] = rnorm(1,x[n])
mod = lm(y~x)
plot(x,y)
abline(mod,col='red')
```




```
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9750 -0.6697 -0.2608  0.6593  2.2404
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0505     0.4006   0.126    0.903
## x             0.9023     0.1207   7.474  7.1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.242 on 8 degrees of freedom
## Multiple R-squared:  0.8747, Adjusted R-squared:  0.8591
## F-statistic: 55.86 on 1 and 8 DF,  p-value: 7.099e-05
```

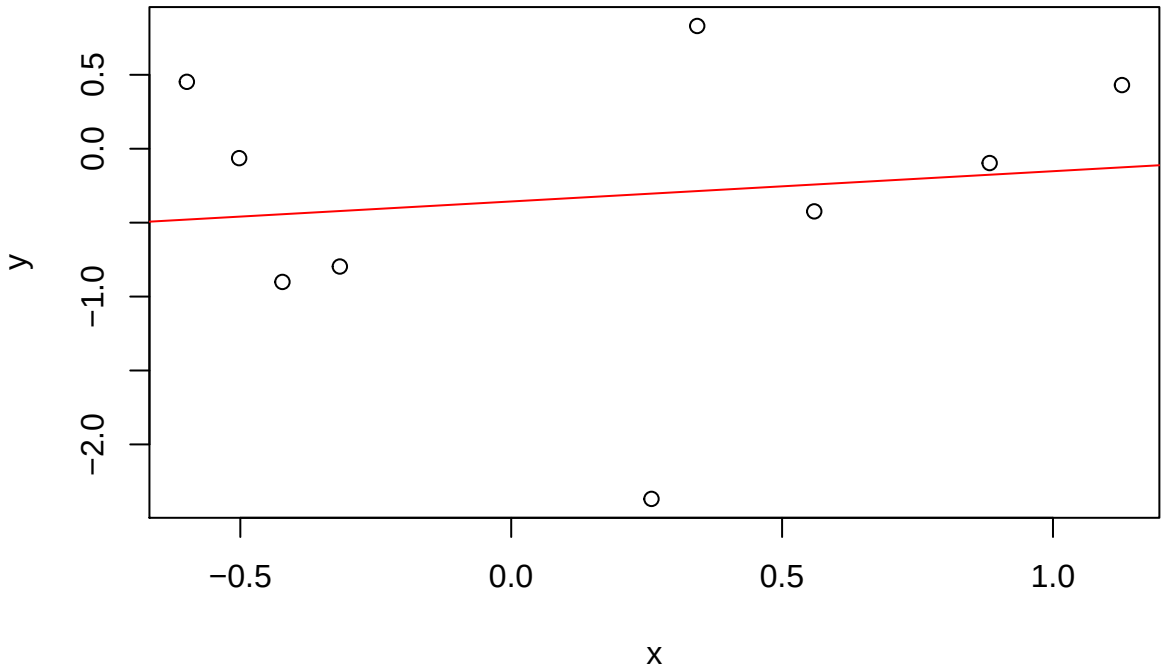
What's causing this?

One way to understood SLR (or MLR) is to imagine that each datapoint is connected to line by a spring. The best-fit line is then what you would get if you let this setup come to a physical equilibrium.

Following this physical intuition, we see that the point at $x = 10$ has a high amount of *mechanical leverage*. Because the "fulcrum" is so large, it exerts **undue influence** on the best fit line. If we remove this point, the issue is resolved:

```
n=10
x = rnorm(n-1)
y = rnorm(n-1)
mod = lm(y~x)
plot(x,y)
```

```
abline(mod,col='red')
```




```
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.06483 -0.37541  0.07913  0.55651  1.11662
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3566     0.3490  -1.022    0.341
## x             0.2046     0.5648   0.362    0.728
##
## Residual standard error: 1.017 on 7 degrees of freedom
## Multiple R-squared:  0.0184,    Adjusted R-squared:  -0.1218
## F-statistic: 0.1312 on 1 and 7 DF,  p-value: 0.7278
```

In analogy with the physical intuition, such points are said to have **high leverage**. In this SLR example, it was easy to visually determine the presence of a high-leverage. But in MLR (or even in some SLR cases), such a visual diagnostic might be challenging.

One way we can get around this, is to consider the derivative:

$$\frac{d\hat{y}_i}{dy_j} = \text{Change in prediction j due to change in data i}$$

This is actually easy to compute, recall:

$$\hat{\vec{y}} = H\vec{y}$$

And so:

$$\hat{y}_i = H_{i\cdot}\vec{y}$$
$$\hat{y}_i = \sum_j H_{ij} y_j$$

A little Calc 1 gets you that:

$$\frac{d\hat{y}_i}{dy_j} = H_{ij}$$

The hat matrix therefore encodes information about leverage values! This leads us to classic definition of "leverage" $\frac{d\hat{y}_i}{dy_i} = H_{ii}$

This definition of leverage is cool, but it doesn't give us a sense of the "overall" effect of a single datapoint $\vec{x}_i$.

A popular measure of "overall" leverage is called Cook's distance (or "Cook's D"). This is calculated using $\hat{y_{j}}_{(i)}$, the predicted value of the j^{th} observation of y, if we ignore the i^{th} data point during fitting. For the i^{th} datapoint Cook's Distance is calculated:

$$D_i = \frac{\sum_{j=1} (\hat{y}_j - \hat{y}_{j(i)})}{k\hat{\sigma}^2}$$

Where k is the number of independent variables in the model (equiv. the dimension of $\vec{x}_i$).

An equivalent formula for D_i (that is a pain to derive) is:

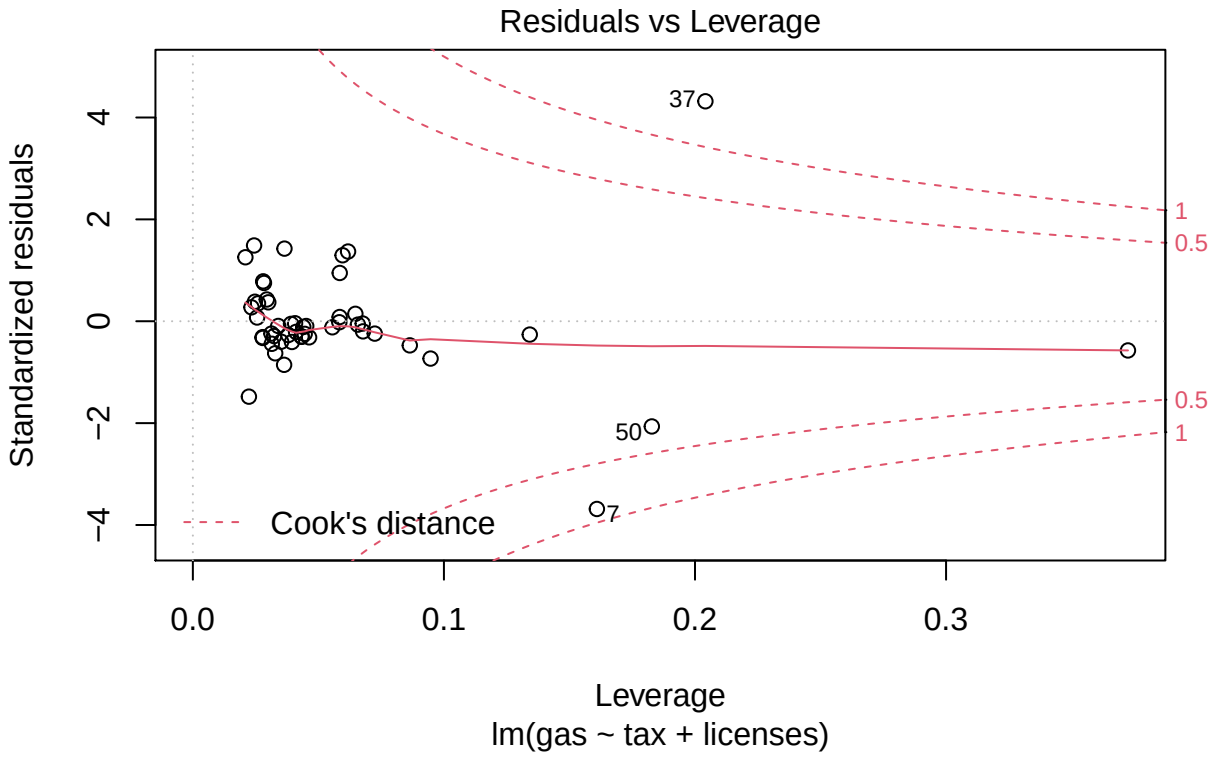
$$D_i = \frac{\tilde{\epsilon}_i}{k} \frac{H_{ii}}{1 - H_{ii}}$$

When $|D_i|$ is "large" then the i^{th} datapoint has a lot of influence, and you should consider excluding it. There is not really a good "cutoff" for $|D_i|$, but if it's values around 1 are usually considered "big".

Finding Outliers in Practice

R makes identifying outliers quite easy:

```
mod = lm(gas~tax+licenses,data=fuel)
plot(mod,which=5)
```




```
mod.no.outliers = lm(gas~tax+licenses,data=fuel[-c(7,37,50),])
summary(mod)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1087.28   -99.33   -36.28   105.39  1240.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 902.04055  362.90691   2.486   0.0165 *
## tax         -96.04517   46.12069  -2.082   0.0428 *
## licenses      0.87770    0.01958  44.835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 322.1 on 47 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9769
## F-statistic:  1037 on 2 and 47 DF,  p-value: < 2.2e-16
```

```
summary(mod.no.outliers)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses, data = fuel[-c(7, 37, 50),
##     ])
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -471.15  -95.76  -55.78   80.38  445.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 448.56287  261.25015   1.717    0.093 .
## tax         -37.07875   33.01597  -1.123    0.268
## licenses      0.88127    0.01333  66.122   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 194.7 on 44 degrees of freedom
## Multiple R-squared:  0.9902, Adjusted R-squared:  0.9898
## F-statistic:  2223 on 2 and 44 DF,  p-value: < 2.2e-16
```

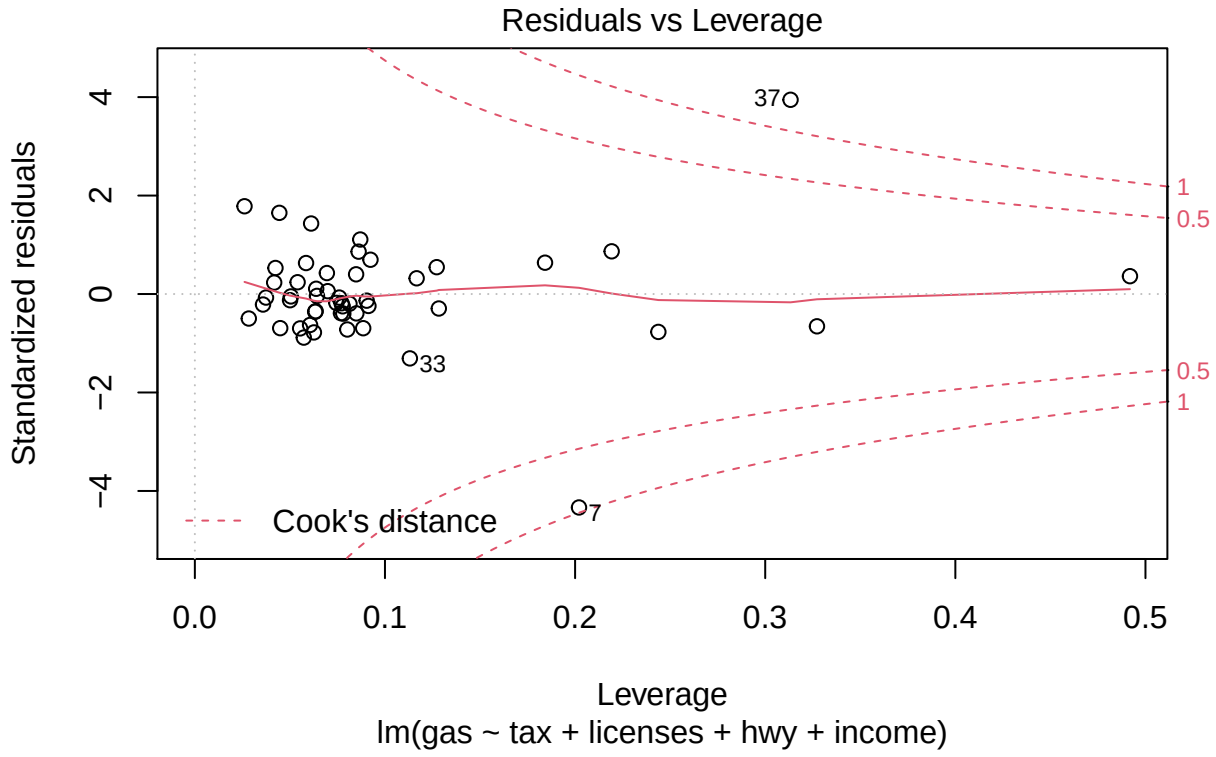
In this case removing the outliers has a pretty serious effect on the output of our model. The decision to include or exclude them therefore needs to beq considered carefully. Was there a possible error in the data process? Are there any factors which differentiate the outliers from the other data points?

```
fuel[c(7,37,50),]
```

```
##      pop tax licenses income   hwy  gas state
## 7  18366   8     8278   5319 11868 6312    NY
## 37 11649   5     6595   4045 17782 7451    TX
## 50   801   5      519   4995   602  276    HI
```

In this case- not really! One way to "stabilize" our results againt outliers in this example is to include more variables in the model:

```
mod = lm(gas~tax+licenses+hwy+income,data=fuel)
plot(mod,which=5)
```



```r
mod.no.outliers = lm(gas~tax+licenses+hwy+income,data=fuel[-c(7,36),])

summary(mod)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses + hwy + income, data = fuel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1031.00  -101.36   -33.43   107.58   870.39
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1262.12973  514.47732   2.453  0.01810 *
## tax          -46.79292   42.61022  -1.098  0.27798
## licenses       0.85530    0.02456  34.826  < 2e-16 ***
## hwy            0.04420    0.01698   2.603  0.01246 *
## income        -0.21610    0.07362  -2.935  0.00523 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266.2 on 45 degrees of freedom
## Multiple R-squared:  0.9855, Adjusted R-squared:  0.9842
## F-statistic: 764.8 on 4 and 45 DF,  p-value: < 2.2e-16
```

```r
summary(mod.no.outliers)
```

```
##
## Call:
## lm(formula = gas ~ tax + licenses + hwy + income, data = fuel[-c(7,
##     36), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -411.38 -132.84  -11.43   87.13  721.70
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 712.89605  414.64306   1.719 0.092752 .
## tax         -17.98421   33.63766  -0.535 0.595649
## licenses      0.86701    0.01917  45.223  < 2e-16 ***
## hwy           0.05699    0.01333   4.275 0.000104 ***
## income       -0.15452    0.05830  -2.650 0.011204 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 205.8 on 43 degrees of freedom
## Multiple R-squared:  0.991,  Adjusted R-squared:  0.9902
## F-statistic:  1183 on 4 and 43 DF,  p-value: < 2.2e-16
```

From this we might conclude that tax has no effect