Lecture 22: Linear Regression

STAT 324

Ralph Trane
University of Wisconsin–Madison

Spring 2020

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Linear Regression

From last time: we have to numerical variables, X and Y, and we want to find out if they are associated in any way (i.e. can we based on the value of one say something about the value of the other?).

Linear Regression: assume the relationship is linear with some random errors, i.e.

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

and that the errors are independent and normally distributed with mean 0, $\epsilon_i \sim N(0, \sigma^2)$.

Based on observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, our best guesses for β_0 and β_1 are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

From these estimates, we can find what are called the *fitted values*. These are simply the values of Y we would expect to see from the x's, if the model is correct. I.e.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

Linear Regression

Using the fitted and observed values, we can find the *residuals*. (These are basically the estimated errors.)

$$\hat{\epsilon}_i = y_i - \hat{y}_i$$

This makes is even more evident why we call the SSE the SSE (sum of squares error):

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \hat{\epsilon}_i^2$$

So far, this is a purely mathematical construct: a straight line through a cloud of points that minimizes the SSE. Turns out we can actually use this to do statistics!

In particular, we want to test $H_0 : \beta_1 = 0$ vs. $H_A : \beta_1 \neq 0$. That is, is there indeed a relationship between x and y, or did we just happen to see one in the sample we got?

Remember, if we go out and measure x and y in a new sample, we will get a new value of $\hat{\beta}_1$. We never really expect $\hat{\beta}_1$ to be *exactly* the same as β_1.

The question is, is $\hat{\beta}_1$ so far from 0 that we are ready to reject the idea that the true value β_1 is in fact 0?

Linear Regression

To do this, we need to know the distribution of $\hat{\beta}_1$. If we know the distribution of $\hat{\beta}_1$, we can find $P(\text{something more extreme than our observed } \hat{\beta}_1)$, and reject if less than α!

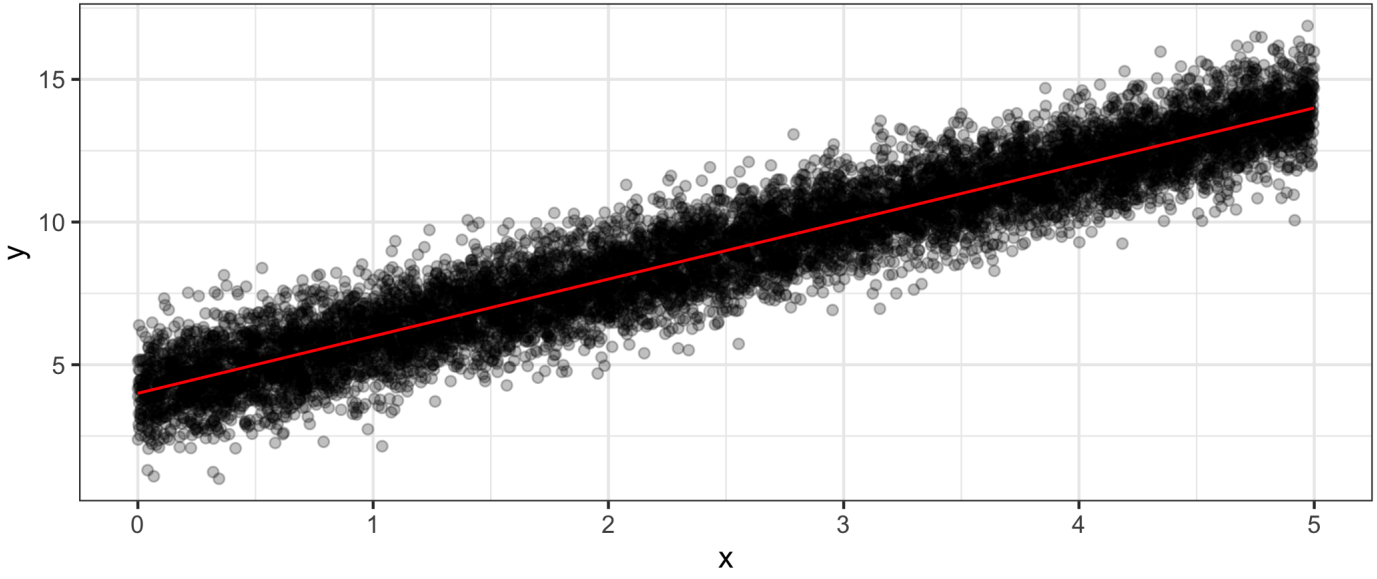
What happens if we go out and get many, many samples from a population where the linear relationship is indeed true?

```
library(tidyverse); library(distributions3); theme_set(theme_bw())

beta0 <- 4
beta1 <- 2

samp_size <- 10000

big_population <- tibble(x = random(Uniform(0, 5), n = samp_size),
                         y = beta0 + beta1*x + random(Normal(0, 1), n = samp_size))

many_many_samples <- tibble(i = 1:3000) %>%
  mutate(sample = map(i, ~sample_n(big_population, size = 35)),
         lin_mod = map(sample, ~lm(data = .x, y ~ x)),
         beta_hats = map(lin_mod, coef)) %>%
  unnest_wider(col = beta_hats) %>%
  rename(beta0_hat = `(Intercept)`, beta1_hat = x)

beta1_sd <- 1/sqrt((35-1)*var(big_population$x))
```

Linear Regression

First, let's take a look at the population data. Keep in mind, we know **for a fact** that the linear model is correct here.

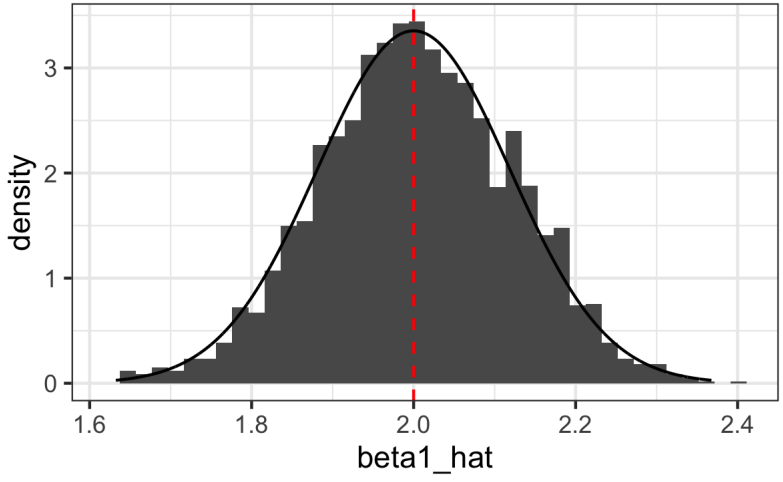
```
ggplot(big_population,
       aes(x = x, y = y)) +
  geom_point(alpha = 0.25) +
  geom_line(aes(y = beta0 + beta1*x),
            color = "red")
```



Linear Regression

Now, if we look at the distribution of the $\hat{\beta}_1$ values from the many, many samples we have, this is what we see. The black line is a curve for a $N\left(\beta_1, \frac{\sigma^2}{(n-1)\cdot s_x^2}\right)$, where $s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ is the sample variance of the x values.

```
ggplot(many_many_samples,
       aes(x = beta1_hat)) +
  geom_histogram(bins = 40,
                 aes(y = after_stat(density))) +
  geom_vline(xintercept = beta1, color = "red", linetype = "dashed") +
  geom_pdf(d = Normal(beta1, beta1_sd))
```



Linear Regression

It seems like $\hat{\beta}_1 \sim N$ with $E(\hat{\beta}_1) = \beta_1$!! So, how would we test $H_0 : \beta_1 = 0$ against $H_A : \beta_1 \neq 0$?

Using the good ol' one sample T-test!! Since $\hat{\beta}_1 \sim N$,

$$T = \frac{\hat{\beta}_1 - E(\hat{\beta}_1)}{\widehat{\text{SD}}(\hat{\beta}_1)} \sim t_{\text{df}}$$

for some appropriate number for the degrees of freedom.

IF H_0 is true, then $E(\hat{\beta}_1) = \hat{\beta}_1 = 0$. Now, we just need a good estimate of $\widehat{\text{SD}}(\hat{\beta}_1)$.

As hinted at on the previous slide, $\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{(n-1)\cdot s_x^2} = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$. So, we need a really good estimator for σ^2.

From ANOVA: MSE = $\frac{\text{SSE}}{N-t}$ is a good estimator for within group variance (i.e. variance of data).

Here, instead of t = number of groups, we use number of *parameters*.

So, a good estimator for σ^2 is MSE = $\frac{\text{SSE}}{n-2} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$.

This finally gives us a good estimator for $\text{SD}(\hat{\beta}_1)$:

$$\widehat{\text{SD}}(\hat{\beta}_1) = \sqrt{\frac{SSE/(n-2)}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

IF $H_0 : \beta_1 = 0$ is true, then

$$T = \frac{\hat{\beta}_1}{\widehat{\text{SD}}(\hat{\beta}_1)} \sim t_{n-2}.$$

So, we can test H_0 against any of the three alternatives as we always do!

For this test to be valid, we have to make some assumptions. If these assumptions are violated, T might not follow a t_{n-2} distribution, which means everything breaks! (In such a scenario, one could do a bootstrap to estimate the distribution of T, and use this to perform the test.)

We have actually implicitly stated the assumptions previously. All assumptions can be compactly written as

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad \epsilon \sim_{\text{iid}} N(0, \sigma^2).$$

Let's unpack:

1. The model is correct (i.e. Y_i's are in fact give as a straight line + noise)
2. Observations are independent (as always...)
3. The variance around the fitted line is constant
4. The random error around the fitted line is normal

Linear Regression

Let's take a look at some real data.

Sir Francis Galton (1822-1911) was interested in how children resemble their parents. One simple measure of this is height. So Galton (actually his disciple, Karl Pearson) measured the heights of father son pairs (in inches) at maturity. In the actual study, 1078 pairs were measured. For convenience, we will use a small subsample of $n = 50$.

As always, we need to address the assumptions before we can consider the results.

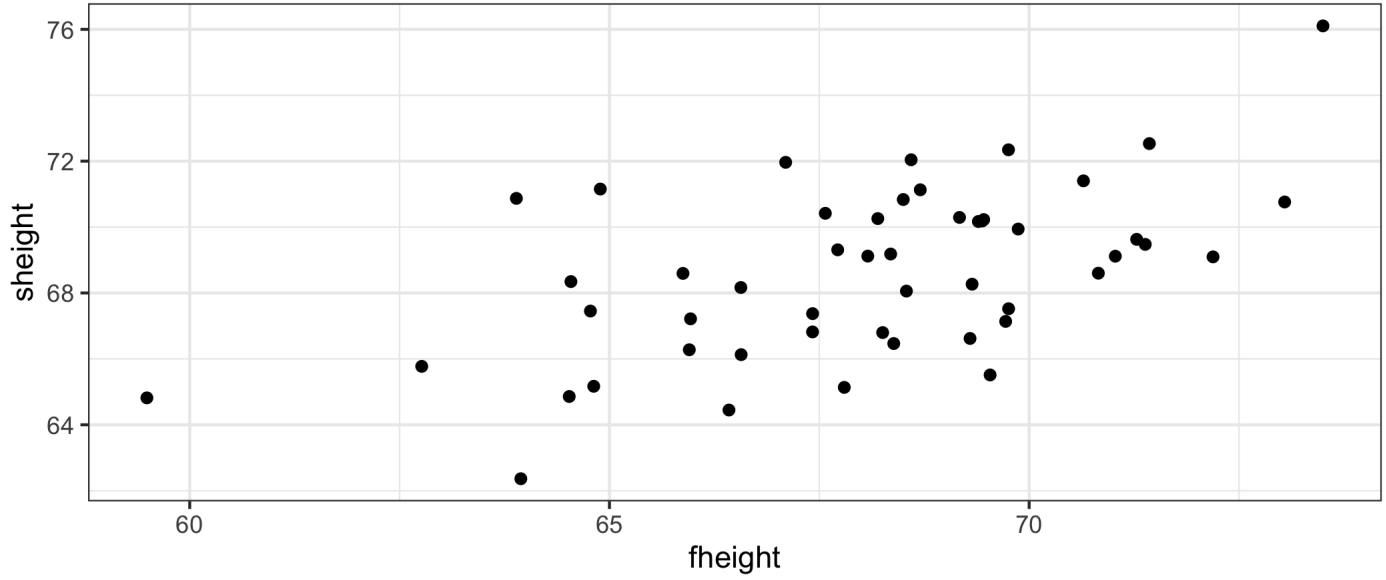
```
father_son_heights
```

```
# A tibble: 50 x 2
   fheight sheight
     <dbl>   <dbl>
 1    71.0    69.1
 2    68.5    70.8
 3    64.8    67.5
 4    67.4    66.8
 5    67.1    72.0
 6    69.9    69.9
 7    71.4    69.5
 8    68.6    72.0
 9    73.5    76.1
10    63.9    70.9
# … with 40 more rows
```

Linear Regression

Assumption 1: the model is correct. Let's look at a scatter plot:

```
ggplot(father_son_heights,
       aes(x = fheight, y = sheight)) +
  geom_point()
```



From this plot, it seems like a linear relationship (i.e. a straight line) with some random error might not be a bad model!

Linear Regression

We check assumptions 2-4 using residuals (since these all are about the residuals!). The easiest way to get these is to fit the model, then use the `broom` package to get the model summaries.

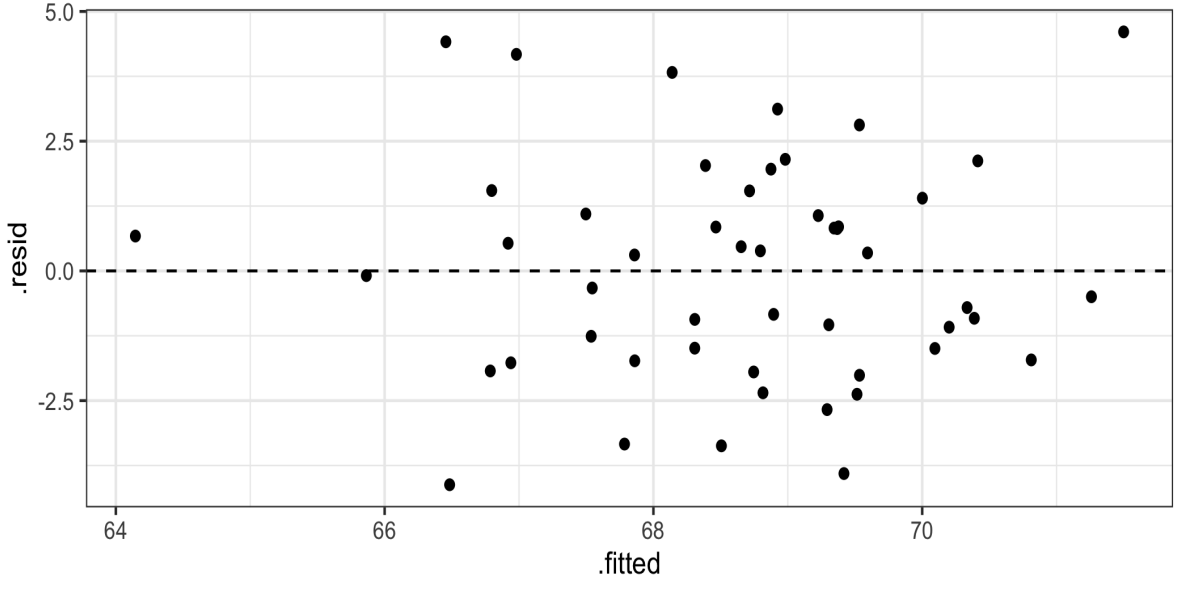
```
library(broom)
lin_mod <- lm(data = father_son_heights, sheight ~ fheight)

augment(lin_mod)
```

```
# A tibble: 50 x 9
   sheight fheight .fitted .se.fit .resid   .hat .sigma  .cooksd .std.resid
     <dbl>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>      <dbl>
 1    69.1    71.0    70.2   0.457 -1.08  0.0444   2.19 0.00607      -0.511
 2    70.8    68.5    68.9   0.311  1.96  0.0206   2.17 0.00877       0.913
 3    67.5    64.8    66.9   0.480  0.533 0.0489   2.19 0.00163       0.252
 4    66.8    67.4    68.3   0.315 -1.49  0.0210   2.18 0.00517      -0.694
 5    72.0    67.1    68.1   0.324  3.82  0.0224   2.12 0.0364        1.78
 6    69.9    69.9    69.6   0.371  0.346 0.0292   2.19 0.000394      0.162
 7    69.5    71.4    70.4   0.488 -0.913 0.0505   2.19 0.00496      -0.432
 8    72.0    68.6    68.9   0.313  3.12  0.0209   2.14 0.0225        1.45
 9    76.1    73.5    71.5   0.690  4.61  0.101    2.07 0.283         2.24
10    70.9    63.9    66.5   0.560  4.42  0.0667   2.09 0.158         2.11
# … with 40 more rows
```

Linear Regression

Assumption 2: independence. This is still hard to check using data. One thing we can do, though, is to look at the residuals, and make sure we do not see any patterns when plotted against the fitted values.

```
ggplot(augment(lin_mod),
       aes(x = .fitted, y = .resid)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point()
```



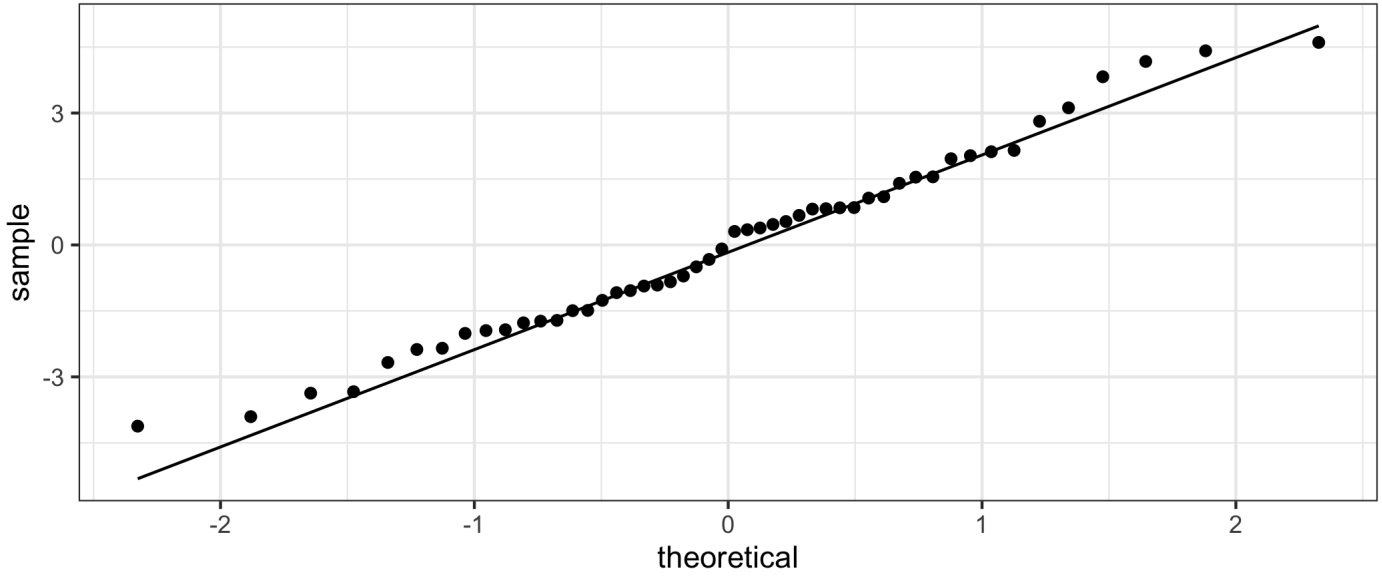
We want this figure to look as random as possible. No patterns. Looks good here!

This also takes care of assumption 3: equal variance. If the variance is not equal, then we would see a pattern in the plot above (for example, a trumpet shape, or a horizontal hour-glass).

Linear Regression

Assumption 4: normal residuals. We use a QQ-plot to assess this. This looks remarkably good!

```
ggplot(augment(lin_mod),
       aes(sample = .resid)) +
  geom_qq() + geom_qq_line()
```



(Might not be that surprising -- heights are often used as an example for a normally distributed variable for a reason...)

Linear Regression

We have checked all assumptions, so we can now finally take a look at the results from the linear regression:

```
summary(lin_mod)
```

```
Call:
lm(formula = sheight ~ fheight, data = father_son_heigh

Residuals:
    Min      1Q  Median      3Q     Max
-4.1206 -1.6603  0.1081  1.3256  4.6061

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  32.9222     7.7030   4.274 9.06e-05 ***
fheight       0.5249     0.1131   4.639 2.72e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.

Residual standard error: 2.169 on 48 degrees of freedom
Multiple R-squared:  0.3096,    Adjusted R-squared:  0.
F-statistic: 21.52 on 1 and 48 DF,  p-value: 2.722e-05
```

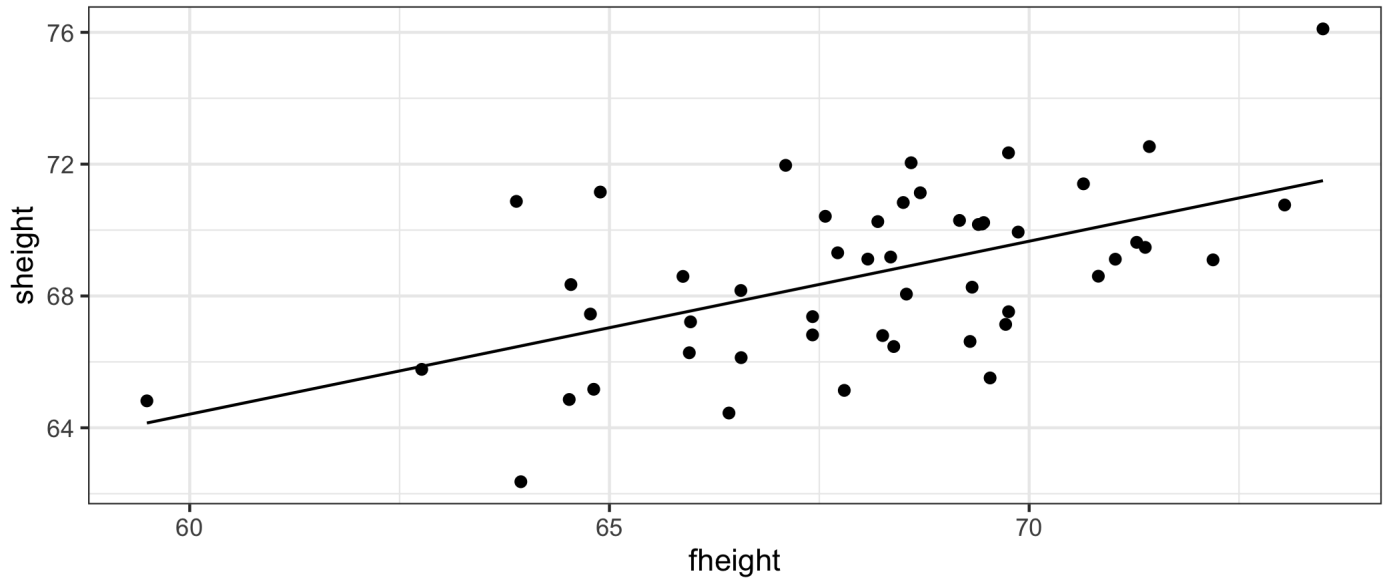
Things we notice:

- $\hat{\beta}_1$ is positive
  - i.e. taller men tend to have taller sons
- we reject $H_0 : \beta_1 = 0$ in favor of $H_A : \beta_1 \neq 0$
  - the relationship seems to be significant, and not simply due to random chance

Linear Regression

Plot the model with the data:

```
ggplot(augment(lin_mod),
       aes(x = fheight, y = sheight)) +
  geom_point() +
  geom_line(aes(y = .fitted))
```



Looks fairly convincing to me!

Linear Regression

Since we know that $T = \frac{\hat{\beta}_1}{\widehat{\text{SD}}(\hat{\beta}_1)} \sim t_{n-2}$, we can find confidence intervals for the slope!

A similar result can be obtained for the intercept $\hat{\beta}_0$.

Using the `tidy` function from the `broom` package gives us a nice looking summary of our linear regression:

```
tidy(lin_mod, conf.int = TRUE, conf.level = 0.95)
```

```
# A tibble: 2 x 7
  term        estimate std.error statistic   p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)   32.9       7.70       4.27 0.0000906   17.4      48.4
2 fheight        0.525     0.113      4.64 0.0000272    0.297     0.752
```

We see that the 95% CI for $\hat{\beta}_1$ does NOT contain 0, which aligns with the fact the we rejected the null hypothesis $H_0 : \beta_1 = 0$.

Linear Regression

From our model, we can say things like

- taller fathers tend to have taller sons
  - from $\hat{\beta}_1 > 0$
- a 6 foot tall father is expected to have a son of height 70.71
  - $E(y|x = 72) = \hat{y}|x = 72 = 32.92 + 0.52 \cdot 72$

When making predictions like above, we need to be careful that we don't generalize to parts of the population we do not have data on. For example, using the model to say that 7 feet tall fathers tend to have sons that are 77.01 is invalid. We have no data about fathers taller than 73.5inches (6 feet, 1.5 in). This is just as nonsensical as using the model to predict daughters' heights based on their mothers' heights...

In other words, the model is only valid in a domain where we have data.

Linear Regression

We can use the model to make predictions, as we just saw. But this prediction is only correct if our estimates of β_0 and β_1 are correct. So we would like a way to make predictions that include our uncertainty about β_0 and β_1. I.e. we would like to come up with *prediction intervals*.

(Note: x^* simply indicates a new value of x in contrast to a value of x that is in our data.)

There are two different kinds of predictions we can make:

- expected value: do we want to predict the *average* height of a son whose father is x^* inches?
- single observation: do we want to predict the *observed* height of a son whose father is x^* inches tall?

For the former, we need to worry about uncertainty in β_0 and β_1 only.

For the latter, we also have to worry about the residual -- i.e. the random noise that is associated with a single observation. Hence, intervals for predicting a single observation will be wider than intervals predicting expected value.

Expected value

The estimate of the expected value is simply $\hat{y}|x^*$, i.e. the point on the line that corresponds to the value x: $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x^*$.

Our prediction interval for the expected value will be of the form $\hat{y}|x^* \pm t_{\alpha/2,n-2}\widehat{\mathrm{SD}}(\hat{y}|x^*)$, where

$$\widehat{\mathrm{SD}}(\hat{y}|x^*) = \hat{\sigma}\sqrt{1/n + \frac{(x^* - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}.$$

We need a good estimator for $\hat{\sigma}$, which will be the same as before: $\frac{SSE}{n-2} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}$.

Single observation

The estimate is the same ($\hat{y} = \hat{\beta}_0 + \hat{\beta}_1x^*$), and the general form of the prediction interval is the same, but the standard deviation changes, because we now have to factor in uncertainty from the residuals.

Our prediction interval for the expected value will be of the form $\hat{y}|x^* \pm t_{\alpha/2,n-2}\widehat{\text{SD}}(\hat{y}|x^*)$, where

$$\widehat{\text{SD}}(\hat{y}|x^*) = \hat{\sigma}\sqrt{1 + 1/n + \frac{(x^* - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}.$$

We need a good estimator for $\hat{\sigma}$, which will be the same as before: $\frac{SSE}{n-2} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}$.

Notice how similar this is to the prediction interval for the expected value. Only difference is adding 1 in the square root.

Linear Regression

Example: using the `predict` function, we can get both the estimate (`fit`), and prediction intervals for either the expected value (`interval = "confidence"`), or a single predicted value (`interval = "prediction"`).

```
new_xs <- tibble(fheight = c(60, 68, 72))
```

```
predict(lin_mod, newdata = new_xs,
        interval = "confidence") %>%
  as_tibble() %>% # needed so I can muta
  mutate(width = upr-lwr)
```

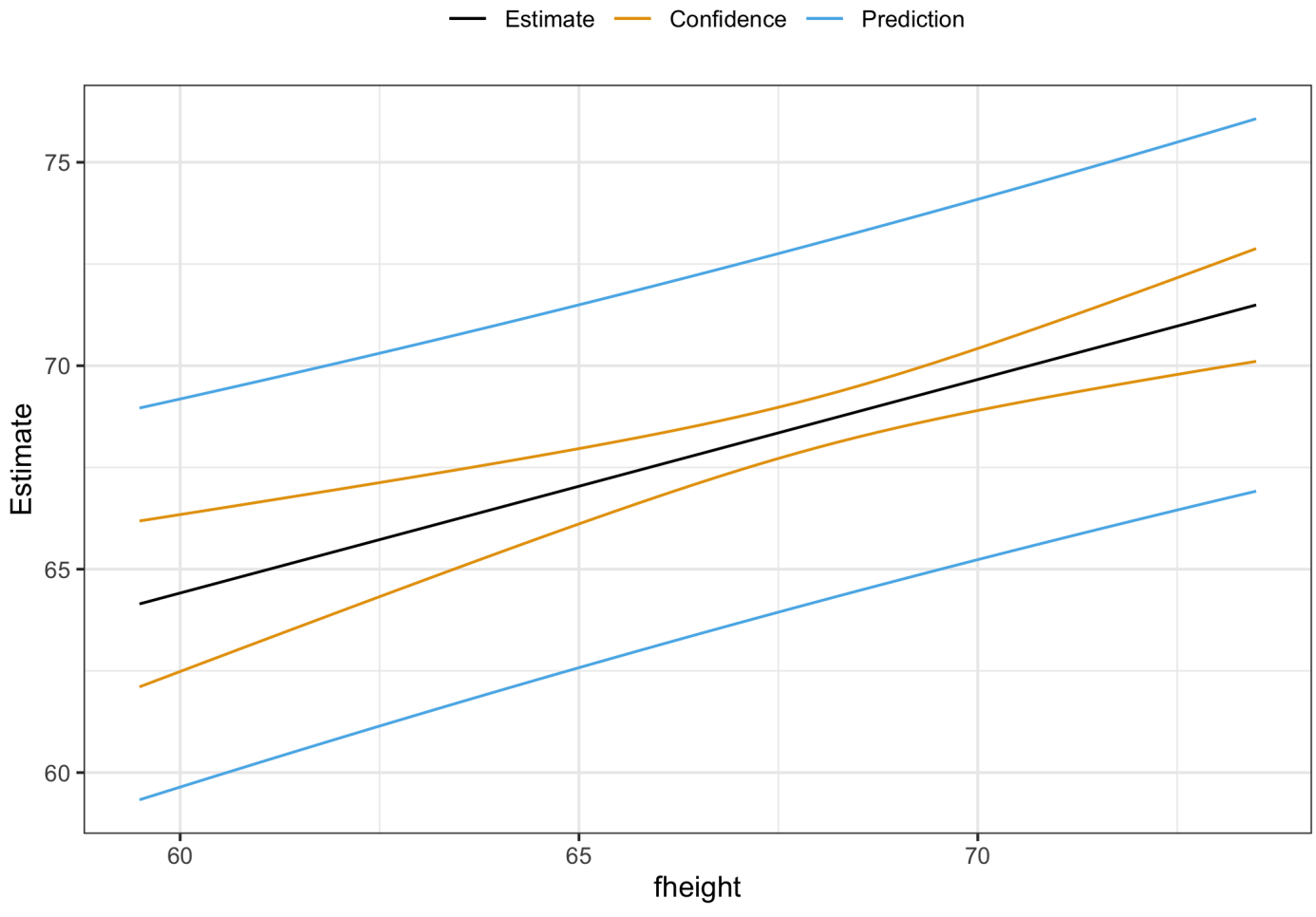
```
# A tibble: 3 x 4
    fit   lwr   upr width
  <dbl> <dbl> <dbl> <dbl>
1  64.4  62.5  66.3  3.86
2  68.6  68.0  69.2  1.23
3  70.7  69.6  71.8  2.19
```

```
predict(lin_mod, newdata = new_xs,
        interval = "prediction") %>%
  as_tibble() %>% # needed so I can muta
  mutate(width = upr-lwr)
```

```
# A tibble: 3 x 4
    fit   lwr   upr width
  <dbl> <dbl> <dbl> <dbl>
1  64.4  59.6  69.2  9.54
2  68.6  64.2  73.0  8.81
3  70.7  66.2  75.2  8.99
```

Notice how the width is not constant: when we are closer to the average of the data, we are more confident in our predictions, which is seen in more narrow intervals.

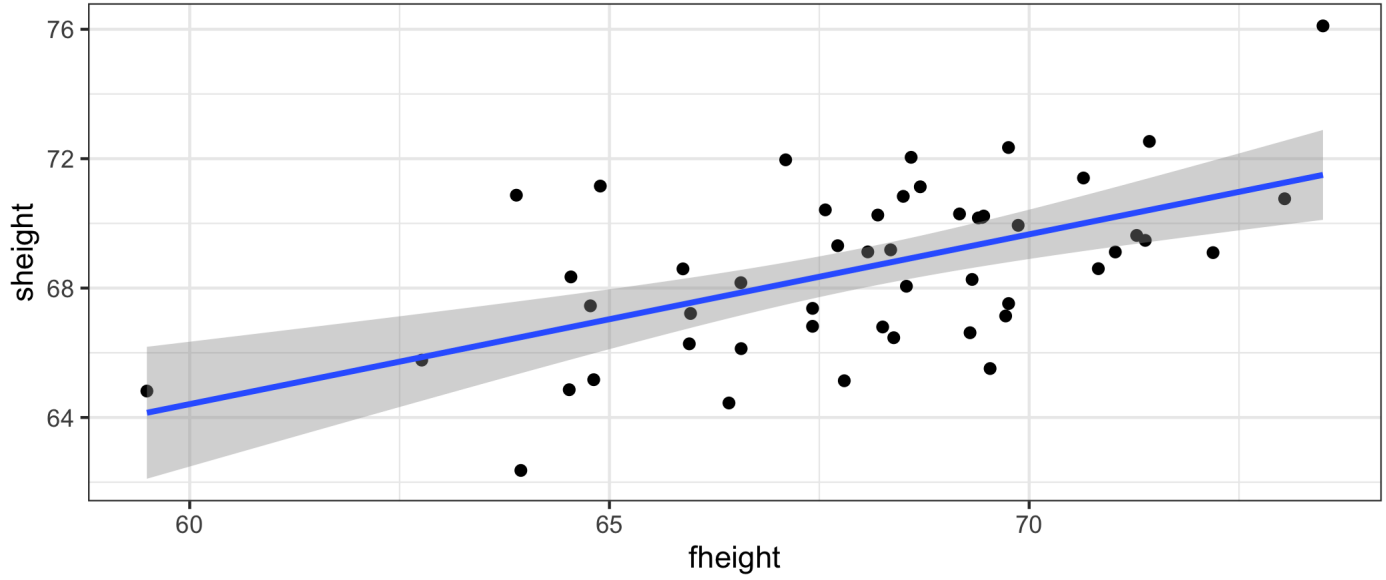
Linear Regression



Linear Regression

Neat shortcut to plot data with linear model and prediction interval for expected value (i.e. "confidence"):

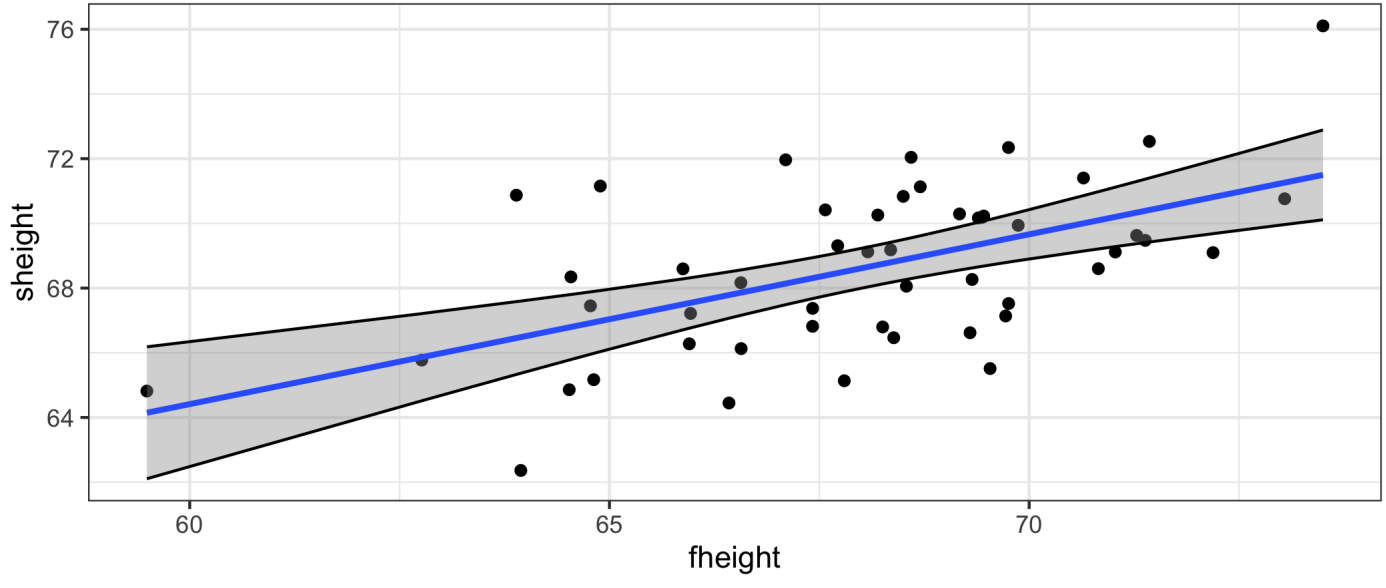
```
ggplot(data = father_son_heights,
       aes(x = fheight, y = sheight)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Linear Regression

Overlay manually calculated limits:

```
with_man_preds <- bind_cols(father_son_heights,
  predict(lin_mod, newdata = father_son_heights, interval = "confidence") %>% as_tibb

ggplot(with_man_preds,
       aes(x = fheight, y = sheight)) +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_line(aes(y = lwr)) + geom_line(aes(y = upr))
```



Linear Regression

Unfortunately, there is one more thing we need to talk about in regards to linear regression: R^2.

Unfortunately, because this is a terrible metric that gets misused all the time...

The idea: R^2 measures the variability in the data that is explained by the model.

$$R^2 = \frac{\text{SSTotal} - \text{SSE}}{\text{SSTotal}}$$

This is a metric that ranges from 0 (no variability explained by the model) to 1 (all variability explained by the model).

Here, $\text{SSTotal} = \sum_{i=1}^{n} (y_i - \bar{y})^2$.

Linear Regression

Using R^2 to measure how much of the variability can be explained by the model makes sense.

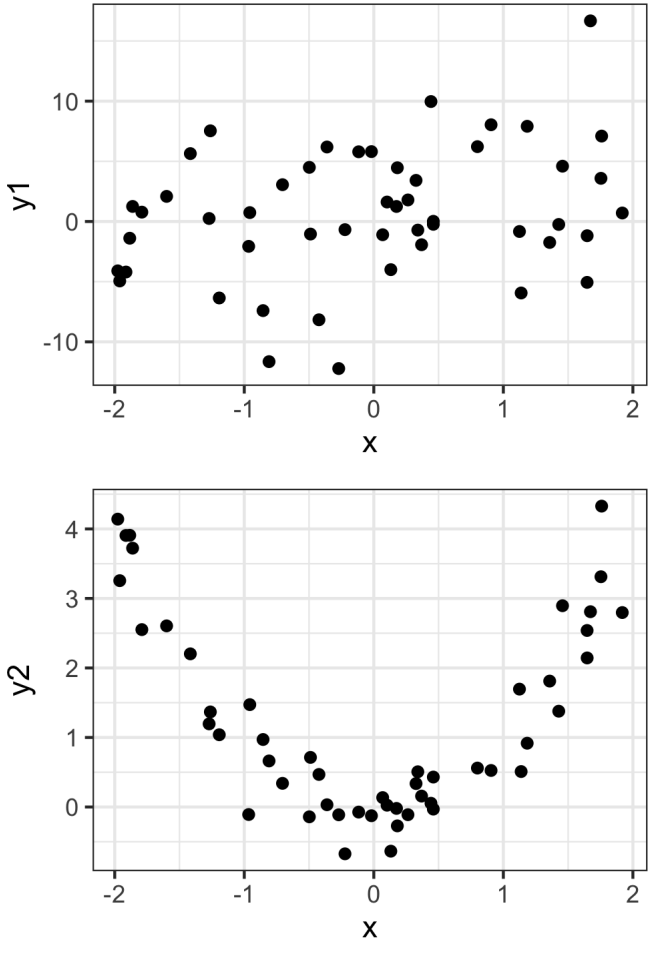
Using R^2 to measure how much of the variability in y can be explained by x makes sense **ONLY IF** the true model is in fact linear.

- This is a common way of using R^2: as a measure of the strength of the association between x and y. This is generally a **bad idea**!

Using R^2 to justify your model is a bad idea! It does not help check any assumptions, and it is poorly correlated with correctness of the model. (I.e. perfect model could have low R^2, and awful model could have high R^2.)

Linear Regression

```
samp_size <- 50
different_models <- tibble(x = random(Ur
                           y1 = 1 + x +
                           y2 = x^2 + ra
                           e1 = random(N
                           e2 = random(N
                           e3 = random(N

library(patchwork)

plot1 <- ggplot(different_models,
       aes(x = x, y = y1)) +
  geom_point()

plot2 <- ggplot(different_models,
         aes(x = x, y = y2)) +
    geom_point()

plot1 + plot2 + plot_layout(nrow = 2)
```



Linear Regression

```
lm1 <- lm(data = different_models,
          y1 ~ x)
summary(lm1)$r.squared
```

```
[1] 0.08403316
```

```
lm2 <- lm(data = different_models,
          y2 ~ x)
summary(lm2)$r.squared
```

```
[1] 0.01258555
```

```
lm3 <- lm(data = different_models,
          y1 ~ x + e1 + e2 + e3)
summary(lm3)$r.squared
```

```
[1] 0.09467117
```

Linear Regression

With this in mind:

```
summary(lin_mod)
```

```
Call:
lm(formula = sheight ~ fheight, data = father_son_heights)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1206 -1.6603  0.1081  1.3256  4.6061

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  32.9222     7.7030   4.274 9.06e-05 ***
fheight       0.5249     0.1131   4.639 2.72e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.169 on 48 degrees of freedom
Multiple R-squared:  0.3096,    Adjusted R-squared:  0.2952
F-statistic: 21.52 on 1 and 48 DF,  p-value: 2.722e-05
```

IF the true model is indeed linear, the height of the father explains about 31% of the variability of height of sons.