Module 4_4

Keaton Wilson

4/29/2020

Bringing it all together - how do multiple factors affect penguin nesting?

So, we're going to spend the next two classes focusing on what is probably the most used tool in statistics, and address the big question we've been dancing around for most of this module: **what factors predict penguin nesting, and can we build a model that allows to pick out sites that are the least destructive to the penguin population if we develop a road there**?

Visualizing the problem - Two variables

Let's begin by reading in our data and building a plot that explores some of the variables of interest.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------- tidyverse 1

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------ tidyverse_confli
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

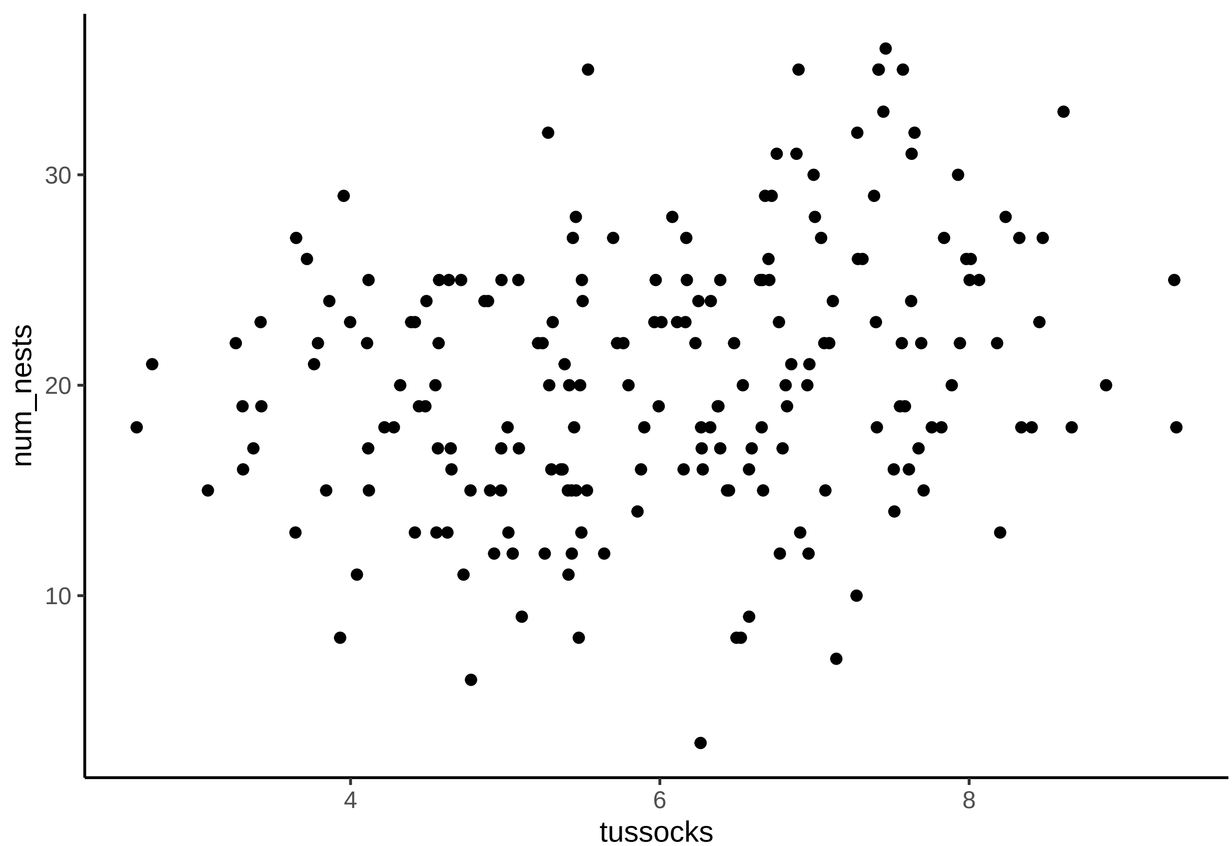
```
# importing the data
site_data = read_csv("https://tinyurl.com/yardhofj")
```

```
## Parsed with column specification:
## cols(
##   site_id = col_double(),
##   year = col_double(),
##   tussocks = col_double(),
##   dist_to_water = col_double(),
##   stone_size = col_double(),
##   num_nests = col_double()
## )
```

```r
glimpse(site_data)
```

```
## Rows: 200
## Columns: 6
## $ site_id       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ year          <dbl> 1971, 1971, 1971, 1971, 1971, 1971, 1971, 1971, 1971,...
## $ tussocks      <dbl> 4.219239, 3.932796, 6.263532, 4.116507, 4.118727, 7.1...
## $ dist_to_water <dbl> 27.86114, 25.33264, 28.93208, 28.39797, 28.75344, 28....
## $ stone_size    <dbl> 45.04550, 39.73190, 37.47425, 45.87984, 45.69094, 38....
## $ num_nests     <dbl> 18, 8, 3, 25, 15, 7, 15, 19, 13, 15, 25, 11, 23, 22, ...
```

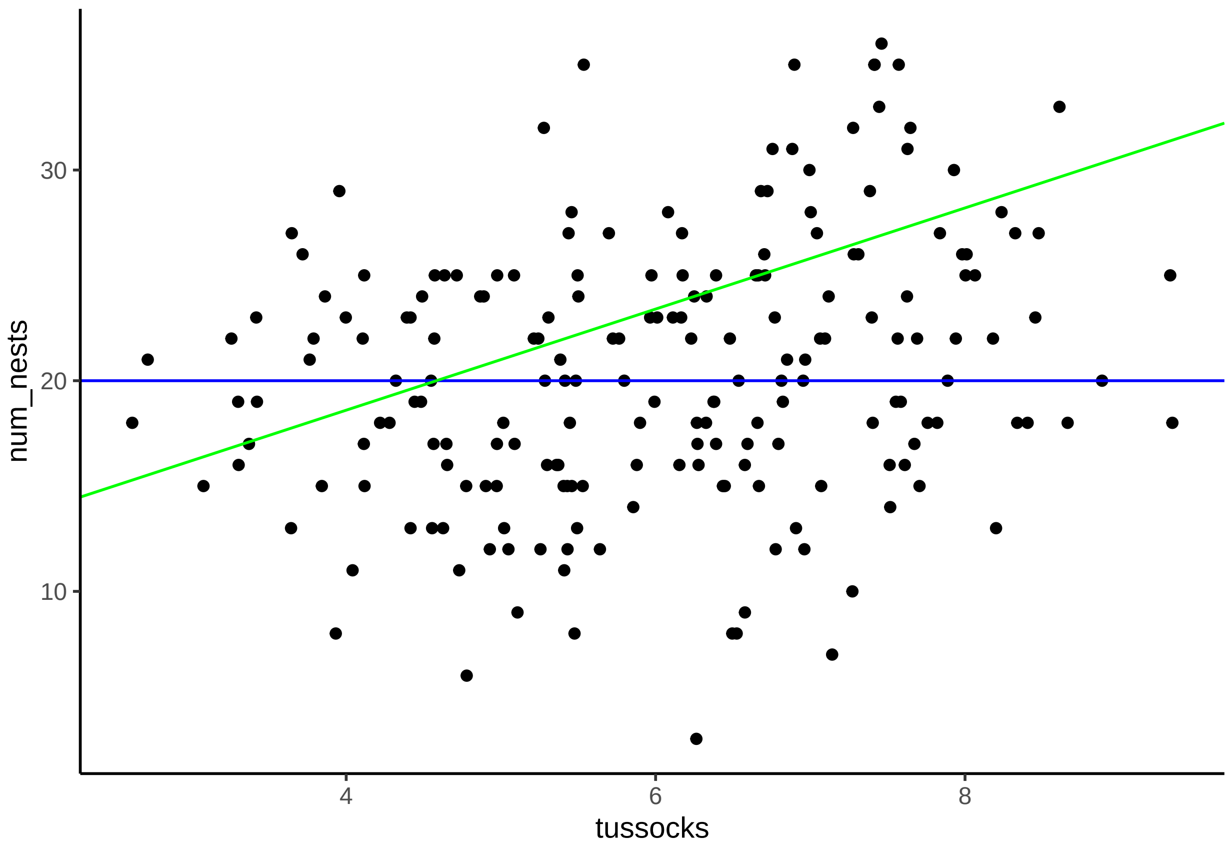
```r
# filtering so we just have the most current data set at our dispsoal
site_data_2011 = site_data %>%
  filter(year == 2011)

# plotting
ggplot(site_data, aes(x = tussocks, y = num_nests)) +
  geom_point() +
  theme_classic()
```



Ultimately, the goal of linear regression is to be able to draw a line through this cloud of points that best represents the pattern between the number of tussocks and the number of nests. This line represents a simple model of how we think the system works and allows us to make predictions about the number of nests at a given site given we know how many tussocks are at that site.

How do we choose this line? Let me illustrate the problem.

```r
ggplot(site_data, aes(x = tussocks, y = num_nests)) +
  geom_point() +
  geom_abline(slope = 0, intercept = 20, color = "blue") +
  geom_abline(slope = 2.4, intercept = 9, color = "green") +
  theme_classic()
```



Individual Exploration - Least Squares?

Spend 10 minutes searching the web. Your goal is to be able to explain how linear regression works. How do we figure out what the best line is? What is this process called, and how would you explain it to someone who has never taken a statistics course before?

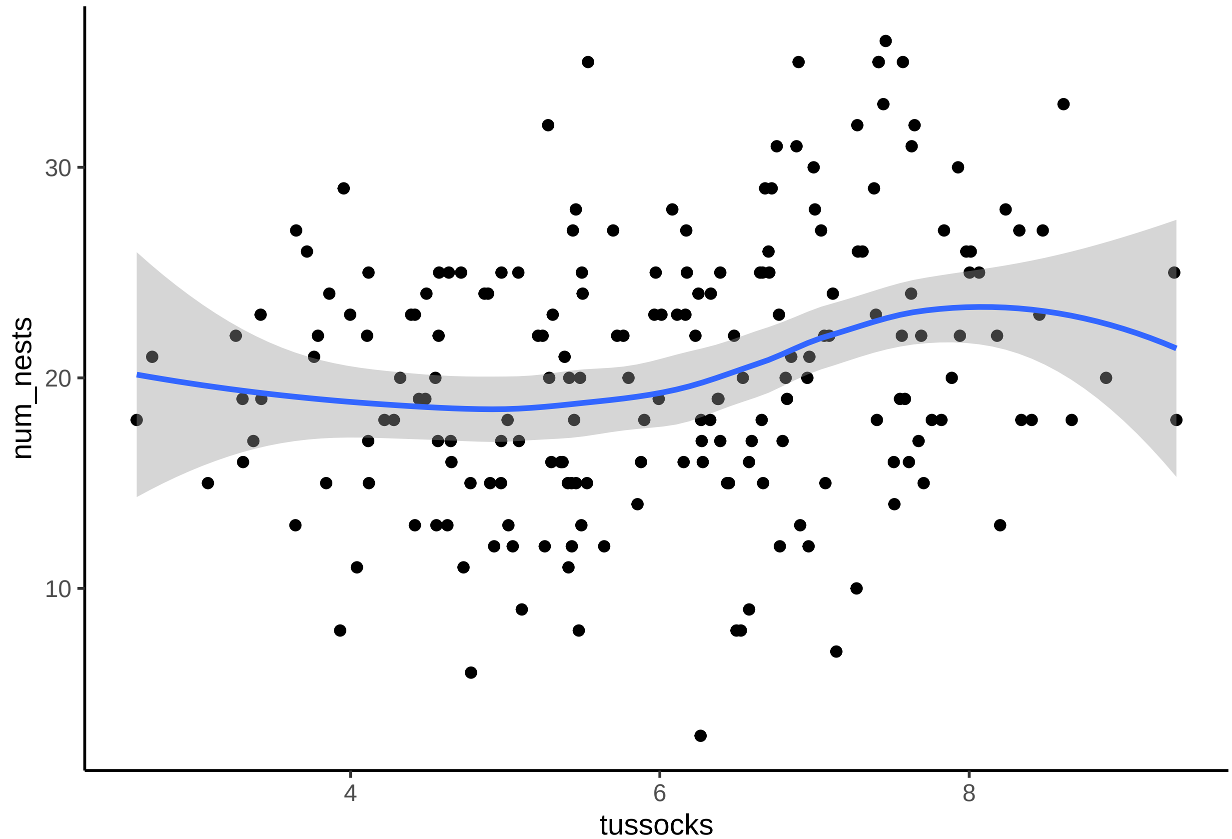
Be ready to report out!

Visualizing the regression line

So now that we have a basic idea of how this works, let's explore how we can visualize a linear regression and also dive into some nitty gritty about building a model and interpreting what R tells us about the model.

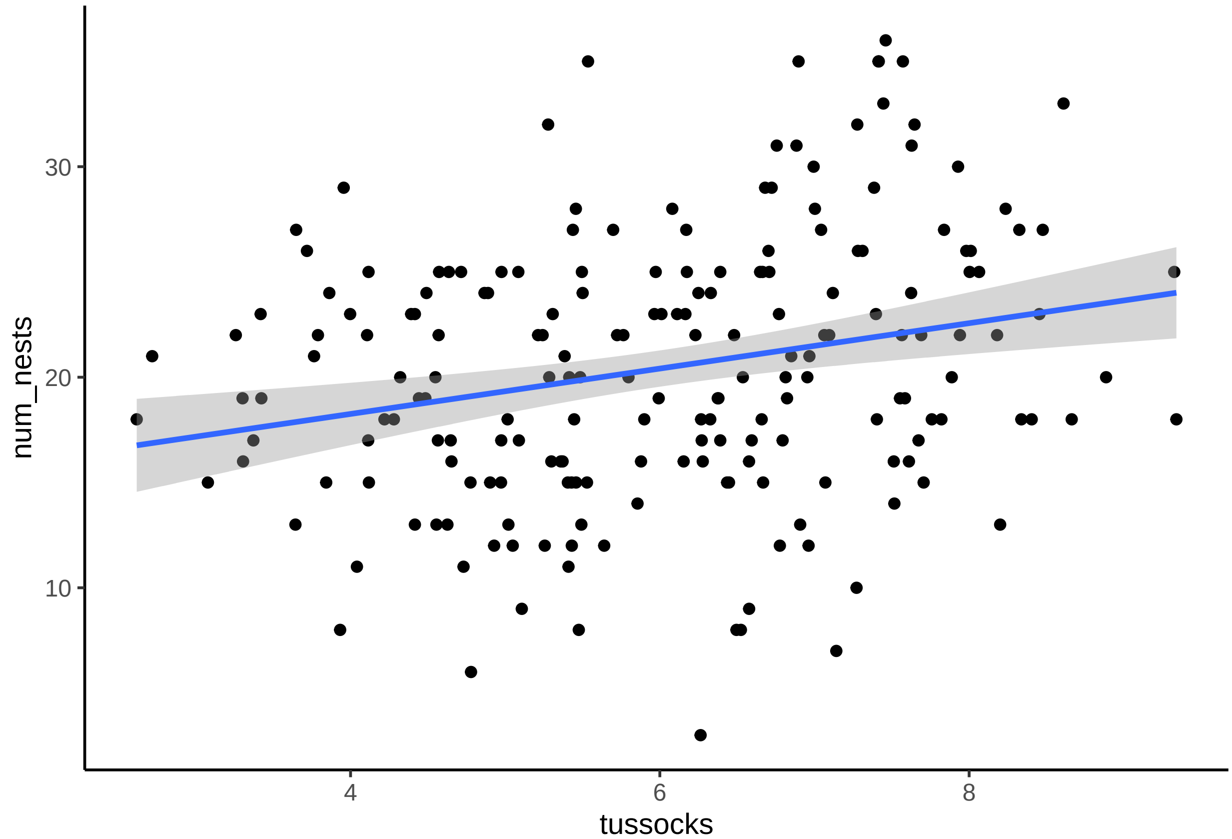
```r
# do a regular smooth first
ggplot(site_data, aes(x = tussocks, y = num_nests)) +
  geom_point() +
  geom_smooth() +
  theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```r
ggplot(site_data, aes(x = tussocks, y = num_nests)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Building a real model and exploring/interpreting

So the plot above is great, but... it's all happening within ggplot. We can't extract any useful information about the regression there. For that, we have to use some built-in code to generate a model (think of it as the equivalent of the t.test material we did earlier).

```
# specifying our model
tussock_mod = lm(num_nests ~ tussocks, data = site_data)
summary(tussock_mod)
```

```
##
## Call:
## lm(formula = num_nests ~ tussocks, data = site_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6952  -4.3233  -0.0771   4.2301  15.0891
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.9446     1.8756   7.435 3.09e-12 ***
## tussocks      1.0778     0.3031   3.556 0.000472 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 6.177 on 198 degrees of freedom
## Multiple R-squared:  0.06002,    Adjusted R-squared:  0.05527
## F-statistic: 12.64 on 1 and 198 DF,  p-value: 0.0004717
```

Group discussion and explanation of output

Spend 10 minutes in your groups discussing the following questions:
1. Can you explain what the following components are and how to interpret them:
a. Estimate for Intercept and tussocks
b. Pr(>|t|)
c. Adjusted R-squared
d. p-value
2. What is the overall conclusion about this model?

This is a big lift! Try and divide and conquer within your groups, and remember that Google is your friend.

As always - be ready to report out!