

Global Crop Yield

Jo Hardin

9/1/2020

For a different project, I'm in need of a dataset to use to work through some inferential linear model ideas. Seems like maybe the crop yield dataset could be used?

Not sure how easy it is to create an inferential claim, but I'm going to try!

The Data

```
fertilizer <- readr::read_csv('cereal_crop_yield_vs_fertilizer_application.csv')
tractors <- readr::read_csv('cereal_yields_vs_tractor_inputs_in_agriculture.csv')
land_use <- readr::read_csv('land_use_vs_yield_change_in_cereal_production.csv')
arable_land <- readr::read_csv('arable_land_pin.csv')

key_crop_yields <- readr::read_csv('key_crop_yields.csv') %>%
  rename(wheat = `Wheat (tonnes per hectare)`,
         rice = `Rice (tonnes per hectare)`,
         maize = `Maize (tonnes per hectare)`,
         soybeans = `Soybeans (tonnes per hectare)`,
         potatoes = `Potatoes (tonnes per hectare)`,
         beans = `Beans (tonnes per hectare)`,
         peas = `Peas (tonnes per hectare)`,
         cassava = `Cassava (tonnes per hectare)`,
         barley = `Barley (tonnes per hectare)`,
         cocoa = `Cocoa beans (tonnes per hectare)`,
         bananas = `Bananas (tonnes per hectare)`)

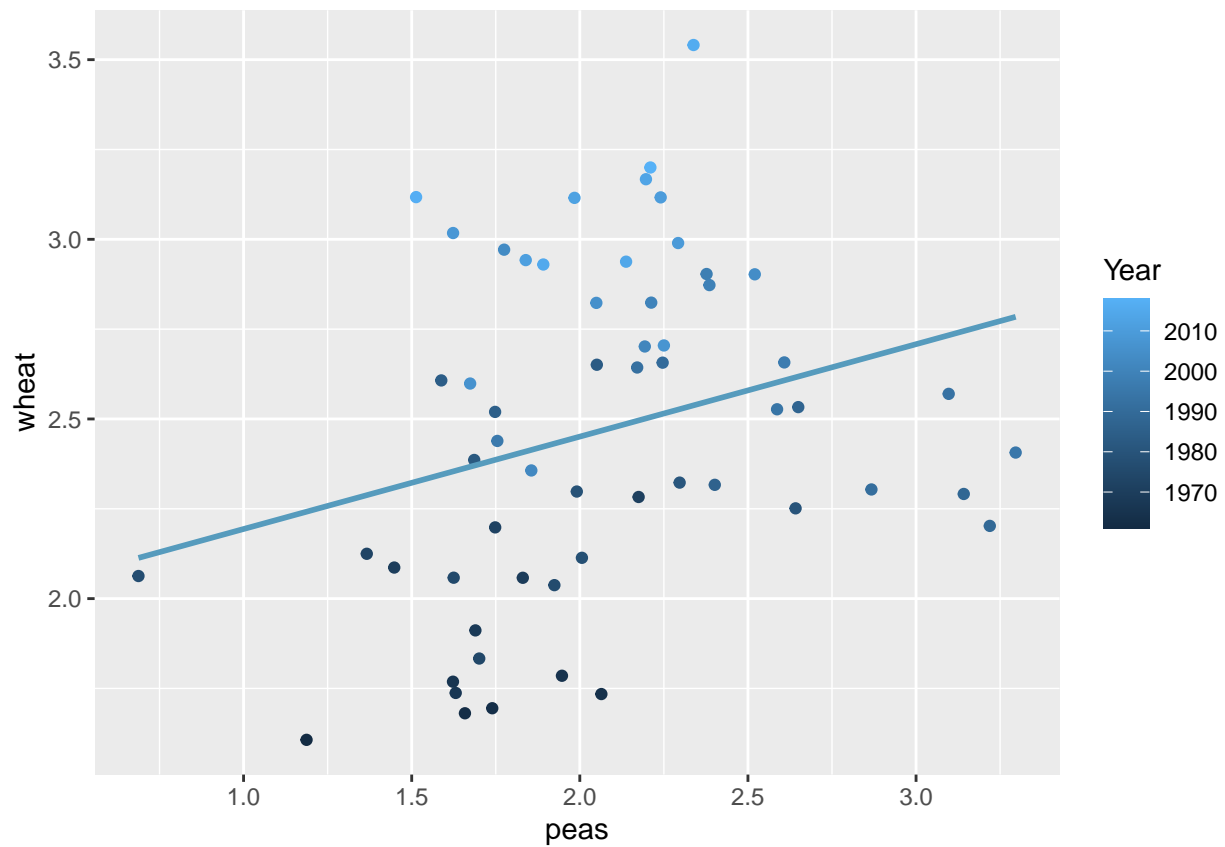
crops_USA <- key_crop_yields %>%
  filter(Code == "USA")
```

Goals for inference in linear regression:

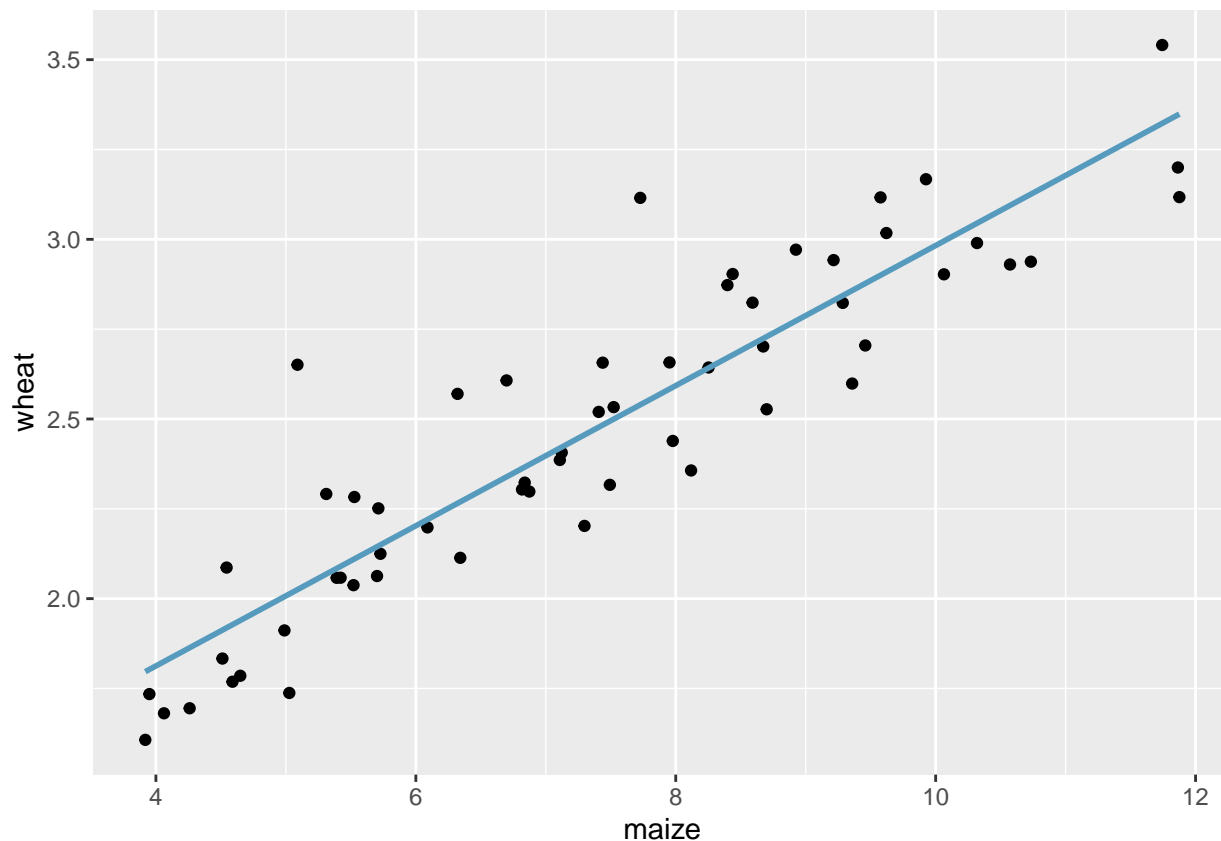
- make inferential claims about the model
- create confidence intervals for the slope

EDA

```
ggplot(crops_USA) +
  geom_point(aes(x = peas, y = wheat, color = Year)) +
  geom_smooth(aes(x = peas, y = wheat), method = "lm", se = FALSE, color = COL[1,1])
```



```
ggplot(crops_USA) +  
  geom_point(aes(x = maize, y = wheat)) +  
  geom_smooth(aes(x = maize, y = wheat), method = "lm", se = FALSE, color = COL[1,1])
```



The code below is trying to find crops that have natural positive and negative correlations. Generally, yield is a function of population size, so I found the percent of a particular crop over all crop yield (summed over time) for a given country. Also, I filtered out countries that didn't have data for most of the crops.

```
library(ggpubr)
library(naniar)

mw <- ggplot(crops_USA) +
  geom_point(aes(x = maize, y = wheat))

crops_country <- key_crop_yields %>%
  filter(!is.na(Code)) %>% # remove continents, etc.
  group_by(Code) %>% # to sum and percent by country
  summarize(
    swheat = sum(wheat, na.rm = TRUE),
    srice = sum(rice, na.rm = TRUE),
    smaize = sum(maize, na.rm = TRUE),
    ssoybeans = sum(soybeans, na.rm = TRUE),
    spotatoes = sum(potatoes, na.rm = TRUE),
    sbeans = sumbeans, na.rm = TRUE),
    speas = sum(peas, na.rm = TRUE),
    scassava = sum(cassava, na.rm = TRUE),
    sbarley = sum(barley, na.rm = TRUE),
    scocoa = sum(cocoa, na.rm = TRUE),
    sbananas = sum(bananas, na.rm = TRUE),
    pwheat = 100*swheat / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + sbarley + scocoa + sbananas),
    price = 100*srice / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + sbarley + scocoa + sbananas),
    pmaize = 100*smaize / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + sbarley + scocoa + sbananas)
```

```

    psoybeans = 100*ssoybeans / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    ppotatoes = 100*spotatoes / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    pbeans = 100*sbeans / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    ppeas = 100*speas / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    pcassava = 100*scassava / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    pbarley = 100*sbarley / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    pcocoa = 100*scocoa / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
    pbananas = 100*sbananas / (swheat + srice + smaize + ssoybeans + spotatoes + sbeans + speas + scassava)
  ) %>%
  replace_na_all(condition = ~.x == 0)%>%
  mutate(nmiss = rowSums(is.na(.))/2) %>%
  filter(nmiss <= 5) # filter out countries that don't have data for most of the crops.

sb <- ggplot(crops_country) +
  geom_point(aes(x = psoybeans, y = pbananas)) +
  xlim(c(0,6)) +
  xlab("% soybeans") +
  ylab("% bananas")

sc <- ggplot(crops_country) +
  geom_point(aes(x = psoybeans, y = pcassava)) +
  xlim(c(0,6)) +
  xlab("% soybeans") +
  ylab("% cassava")

mc <- ggplot(crops_country) +
  geom_point(aes(x = pmaize, y = pcassava)) +
  xlim(c(0,15)) +
  xlab("% maize") +
  ylab("% cassava")

peb <- ggplot(crops_country) +
  geom_point(aes(x = ppotatoes, y = pbananas)) +
  xlim(c(0,60)) +
  xlab("% potatoes") +
  ylab("% bananas")

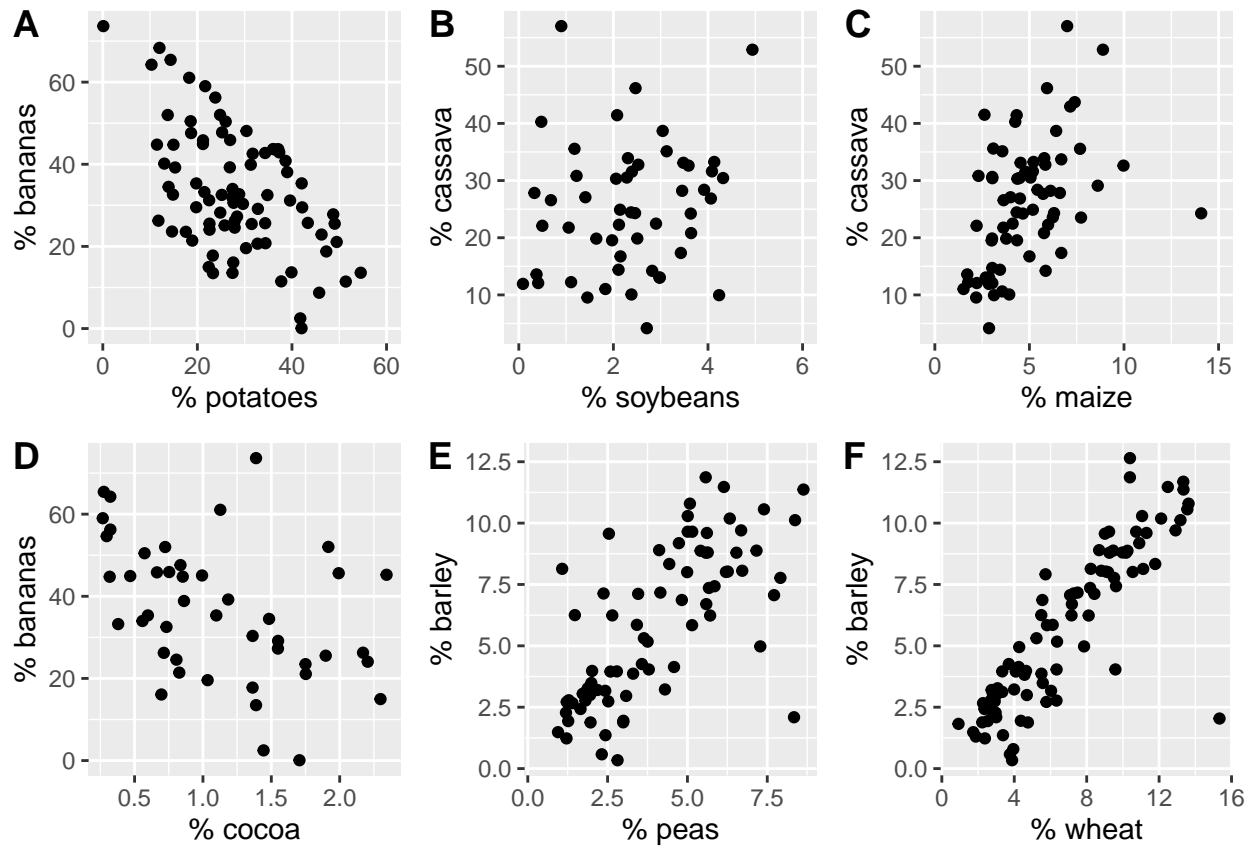
cb <- ggplot(crops_country) +
  geom_point(aes(x = pcocoa, y = pbananas)) +
  xlab("% cocoa") +
  ylab("% bananas")

wb <- ggplot(crops_country) +
  geom_point(aes(x = pwheat, y = pbarley)) +
  xlab("% wheat") +
  ylab("% barley")

pob <- ggplot(crops_country) +
  geom_point(aes(x = ppeas, y = pbarley)) +
  xlab("% peas") +
  ylab("% barley")

```

```
ggpubr::ggarrange( peb, sc, mc, cb, pob, wb,
  labels = c("A", "B", "C", "D", "E", "F"),
  ncol = 3, nrow = 2)
```



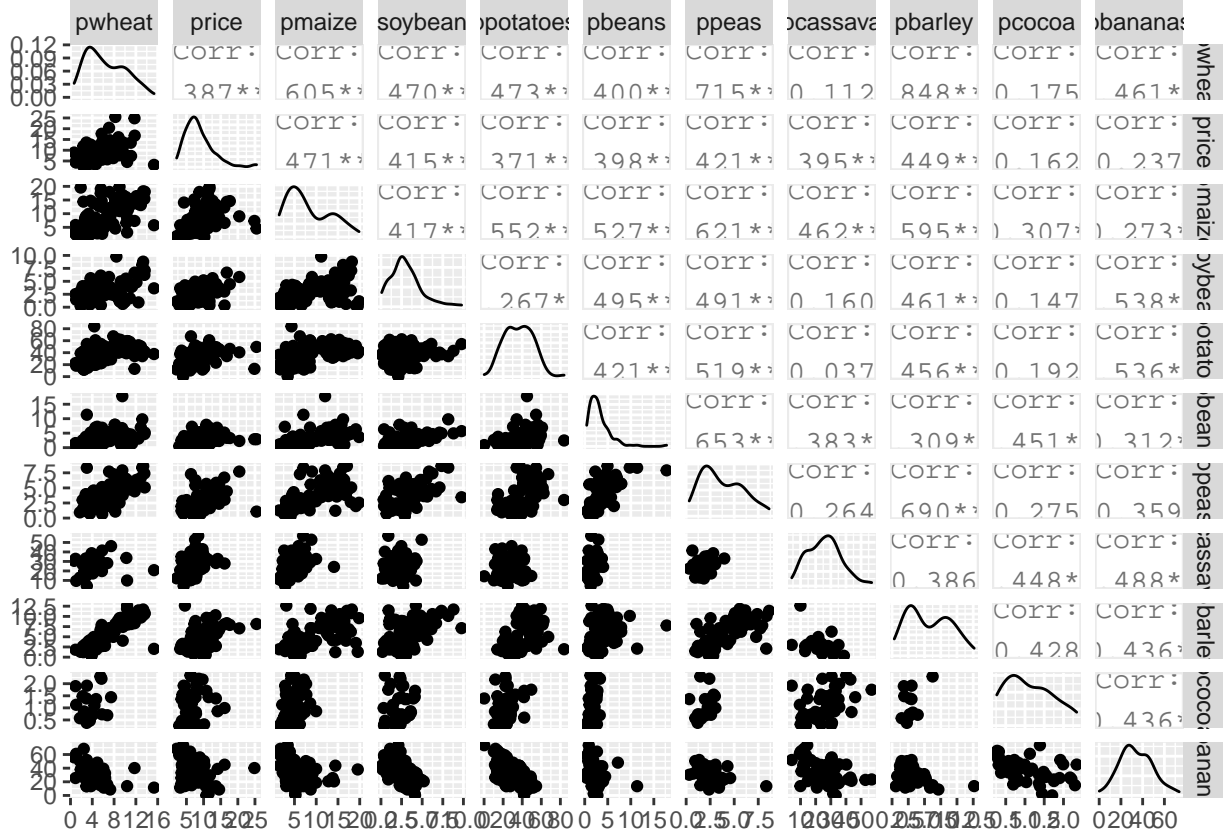
```
temptbl <- tribble(
  ~variable, ~col1, ~col2, ~corval,
  "A", "potatoes", "bananas", round(cor(crops_country$ppotatoes, crops_country$pbananas, use = "pairwise.complete.obs"), 2),
  "B", "soybeans", "cassava", round(cor(crops_country$psoybeans, crops_country$pcassava, use = "pairwise.complete.obs"), 2),
  "C", "maize", "cassava", round(cor(crops_country$pmaize, crops_country$pcassava, use = "pairwise.complete.obs"), 2),
  "D", "cocoa", "bananas", round(cor(crops_country$pcocoa, crops_country$pbananas, use = "pairwise.complete.obs"), 2),
  "E", "peas", "barley", round(cor(crops_country$ppeas, crops_country$pbarley, use = "pairwise.complete.obs"), 2),
  "F", "wheat", "barley", round(cor(crops_country$pwheat, crops_country$pbarley, use = "pairwise.complete.obs"), 2)
)

temptbl %>%
  kable(caption = "Correlation of percentage of total yield across different crops.",
        col.names = c("Graph", "x-variable", "y-variable", "correlation")) %>%
  kable_styling()

library(GGally)
ggpairs(crops_country[,13:23])
```

Table 1: Correlation of percentage of total yield across different crops.

Graph	x-variable	y-variable	correlation
A	potatoes	bananas	-0.54
B	soybeans	cassava	0.16
C	maize	cassava	0.46
D	cocoa	bananas	-0.44
E	peas	barley	0.69
F	wheat	barley	0.85



```
temp <- cor(crops_country[,13:23], use = "pairwise.complete.obs")
diag(temp) <- NA
temp %>%
  quantile(na.rm = TRUE)
```

```
##          0%          25%          50%          75%         100%
## -0.5379375  0.1501472  0.3975110  0.4706361  0.8476391
temp
```

```
##          pwheat          price          pmaize          psoybeans          ppotatoes          pbeans
## pwheat          NA  0.3870334  0.6052646  0.4703257  0.47255763  0.3996283
## price  0.3870334          NA  0.4707395  0.4151908  0.37077250  0.3975110
## pmaize  0.6052646  0.4707395          NA  0.4169392  0.55154020  0.5272872
## psoybeans 0.4703257  0.4151908  0.4169392          NA  0.26707097  0.4950204
## ppotatoes 0.4725576  0.3707725  0.5515402  0.2670710          NA  0.4209704
## pbeans  0.3996283  0.3975110  0.5272872  0.4950204  0.42097040          NA
```

```
## ppeas      0.7150342  0.4214549  0.6210329  0.4914803  0.51887085  0.6533319
## pcassava   0.1116904  0.3948280  0.4620643  0.1603862  0.03726667  0.3833298
## pbarley    0.8476391  0.4494305  0.5949681  0.4611248  0.45560077  0.3092119
## pcocoa     0.1752966  0.1622816  0.3071385  0.1467342  0.19181424  0.4508198
## pbananas   -0.4614798 -0.2374004 -0.2732579 -0.5379375 -0.53592716 -0.3116874
##           ppeas    pcassava    pbarley    pcocoa    pbananas
## pwheat     0.7150342  0.11169041  0.8476391  0.1752966 -0.4614798
## price      0.4214549  0.39482801  0.4494305  0.1622816 -0.2374004
## pmaize     0.6210329  0.46206428  0.5949681  0.3071385 -0.2732579
## psoybeans   0.4914803  0.16038617  0.4611248  0.1467342 -0.5379375
## ppotatoes   0.5188708  0.03726667  0.4556008  0.1918142 -0.5359272
## pbeans      0.6533319  0.38332979  0.3092119  0.4508198 -0.3116874
## ppeas       NA      0.26378967  0.6898022  0.2753218 -0.3588672
## pcassava    0.2637897      NA -0.3860976  0.4482220 -0.4879015
## pbarley     0.6898022 -0.38609755      NA  0.4284665 -0.4355787
## pcocoa      0.2753218  0.44822200  0.4284665      NA -0.4356990
## pbananas    -0.3588672 -0.48790150 -0.4355787 -0.4356990      NA
```

SLR

```
crops_USA %>%
  lm(wheat ~ peas, .) %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)    1.94      0.253     7.67 2.73e-10
## 2 peas           0.257     0.119     2.16 3.50e- 2
```

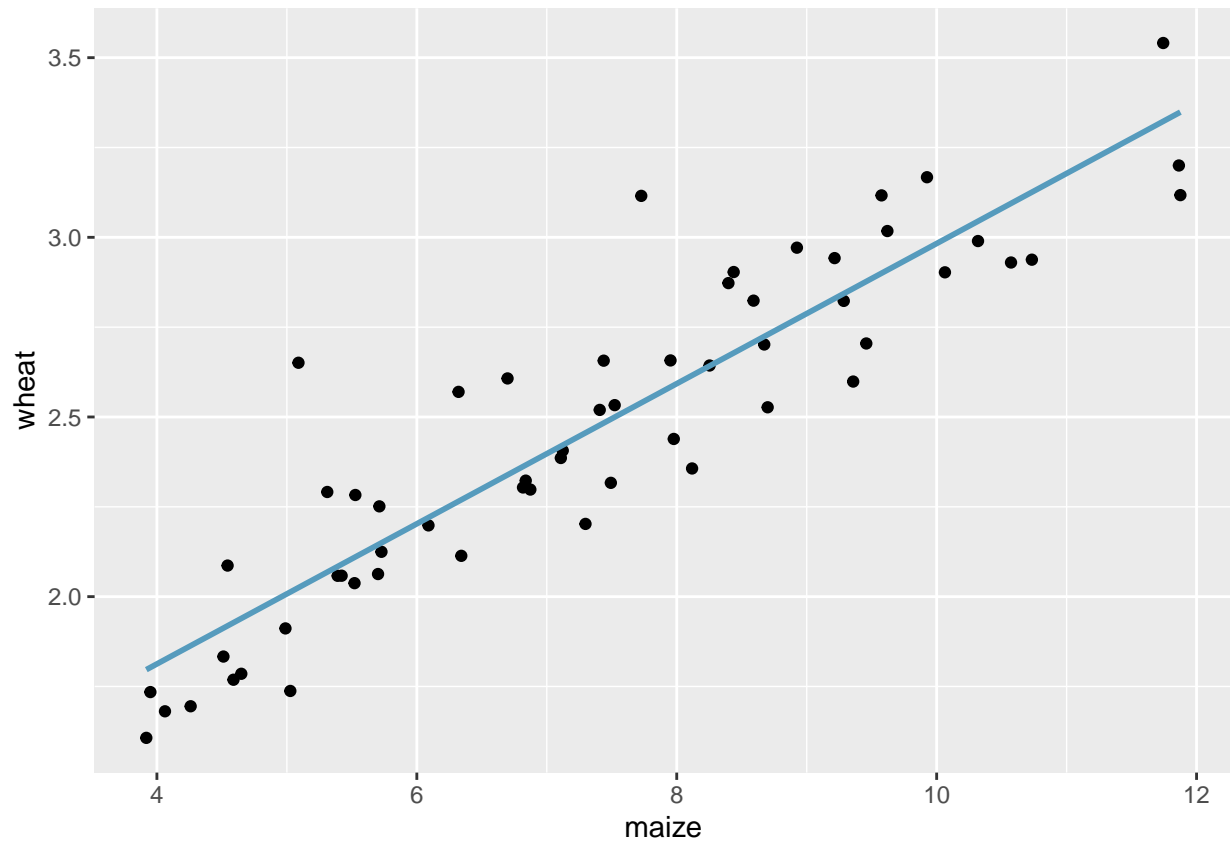
```
crops_USA %>%
  lm(wheat ~ maize, .) %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)    1.03     0.0912    11.3 4.13e-16
## 2 maize          0.195     0.0119    16.4 5.04e-23
```

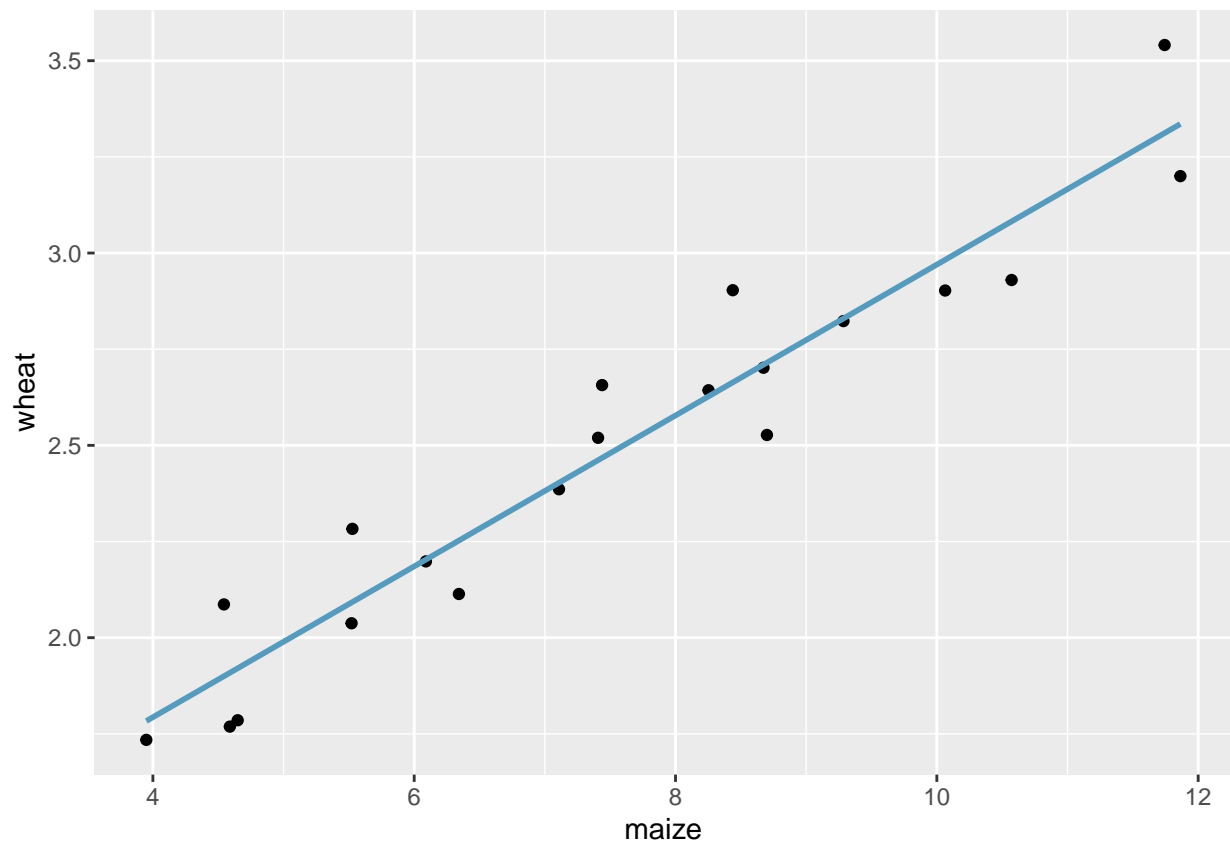
Sampling distribution

```
set.seed(4747)
crops2 <- crops_USA %>%
  sample_n(size=20)
crops3 <- crops_USA %>%
  sample_n(size=20)
crops_many <- crops_USA %>%
  rep_sample_n(size = 20, replace = FALSE, reps = 50)
```

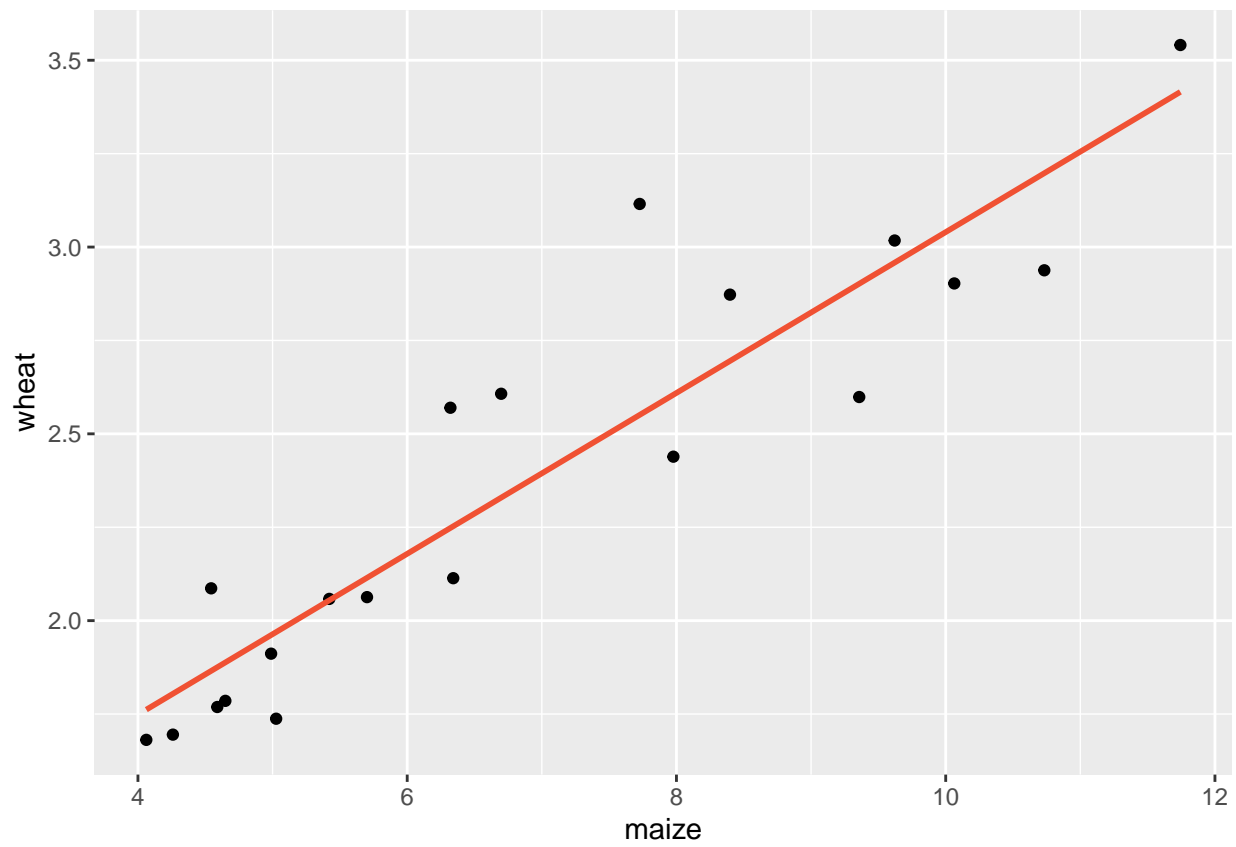
```
ggplot(crops_USA) +
  geom_point(aes(x = maize, y = wheat)) +
  geom_smooth(aes(x = maize, y = wheat), method = "lm", se = FALSE, color = COL[1,1])
```



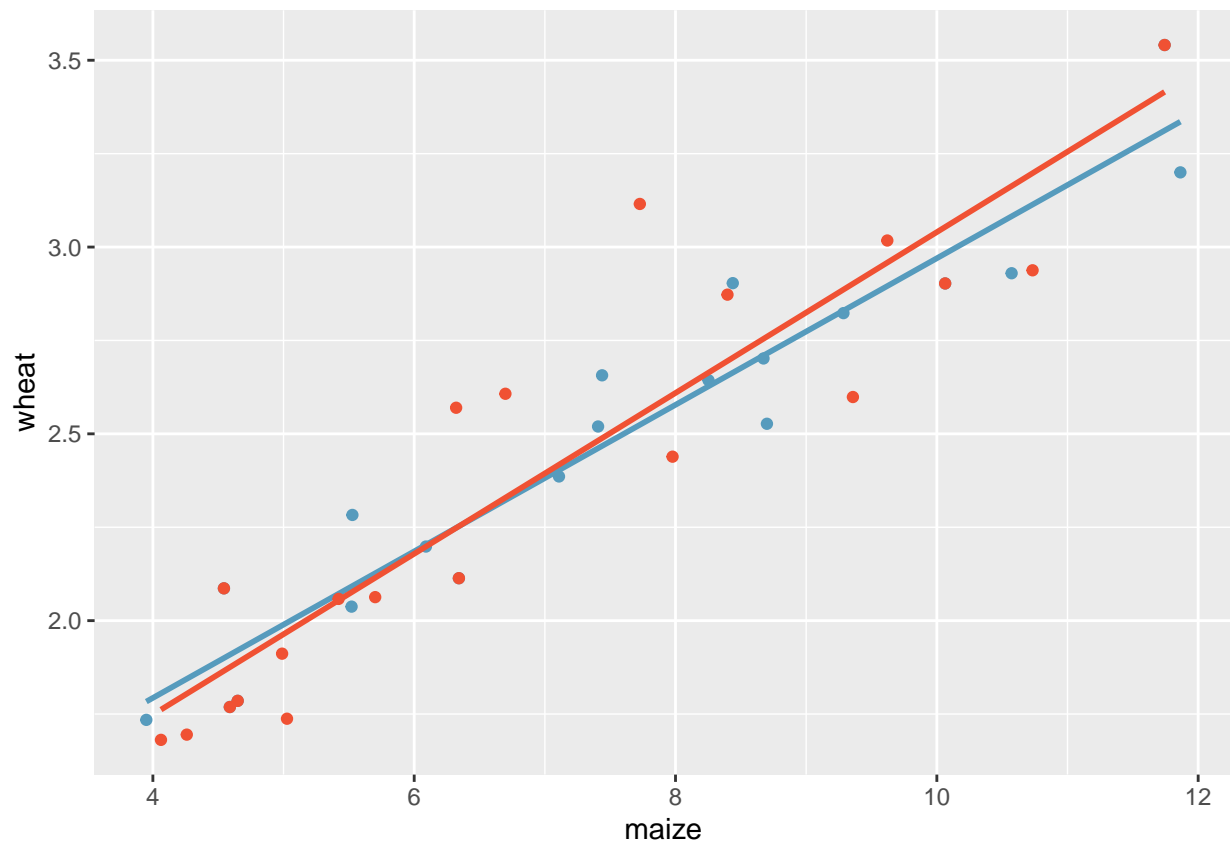
A subset of size 20 items shows a similar positive trend between maize and wheat, despite having fewer observations on the plot.



A second sample of size 20 also shows a positive trend!



But the line is slightly different!



That is, there is variability in the regression line from sample to sample. The concept of the sampling variability is something you've seen before, but in this lesson, you will focus on the variability of the line instead of the variability of a single statistic.

maize vs. wheat for annual crop yield in the US (n=20)

