Chapter 9.1 Linear regression explanatory variable fixed by experiment

Amy Hurford (ahurford@mun.ca)

2023-07-11

This imports the data into R from a website without needing to download (generally, click 'Raw' on the github website and copy url: https://raw.githubusercontent.com/ahurford/biol-4605-data/main/data/corn.csv)

```
data <- read.csv('https://raw.githubusercontent.com/ahurford/biol-4605-data/main/data/corn.csv', fill=TR
```

Give the variables shorter names

```
# Response variable
Pcorn = data$Pcorn
# Explanatory variable
Psoil = data$Psoil
```

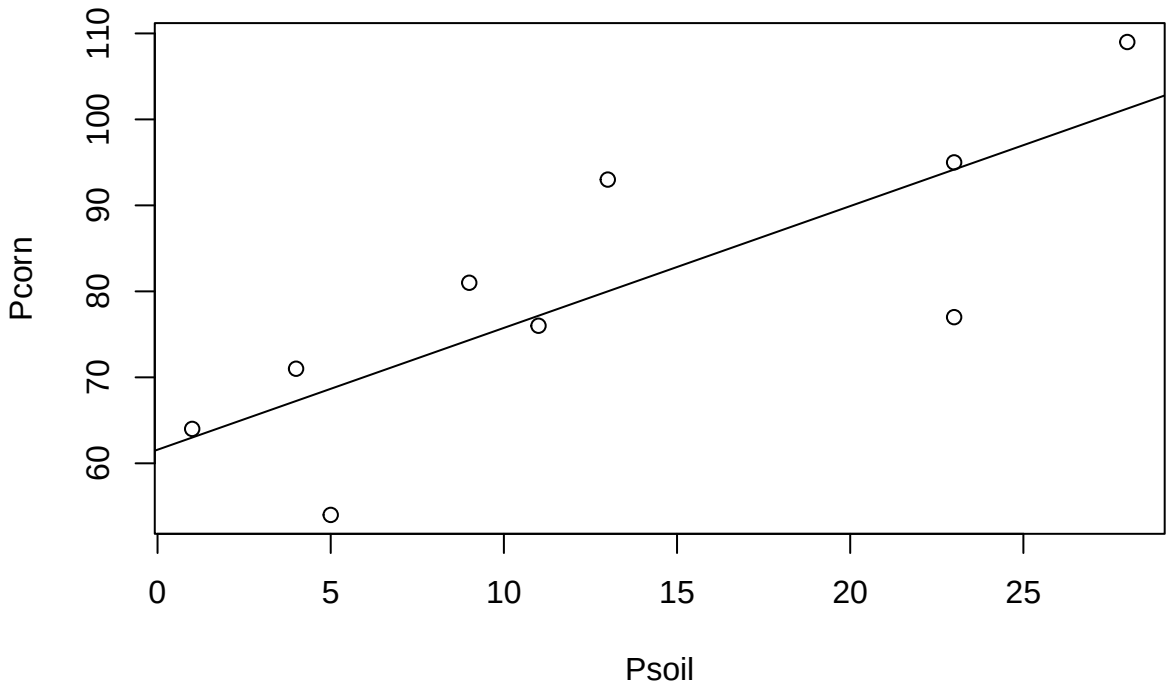
Do the linear regression:

```
reg <- lm(Pcorn~Psoil)
# see the results of your regression
summary(reg)
```

```
##
## Call:
## lm(formula = Pcorn ~ Psoil)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.169  -1.166   1.003   6.668  13.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  61.5804     6.2477   9.857 2.35e-05 ***
## Psoil         1.4169     0.3947   3.590  0.00886 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.69 on 7 degrees of freedom
## Multiple R-squared:  0.648,  Adjusted R-squared:  0.5977
## F-statistic: 12.89 on 1 and 7 DF,  p-value: 0.008859
```

Plot of the regression (line) and the data (open circles)

```
plot(Psoil,Pcorn)
abline(reg)
```



Calculate the likelihood ratio for the evidence supporting the linear regression relative to a null model that no matter the value of phosphorous in the soil, the phosporous in the corn is always equal to the mean recorded value.

```
data.eq_null = data.frame(Psoil = Psoil, Data = Pcorn, Model = rep(mean(Pcorn),length(Pcorn)), res = Pc
```

```
sum(data.eq_null$res)
```

```
## [1] 0
```

```
SS.total = sum(data.eq_null$res2)
```

Data equations for the regression model

Rename the estimated coefficients with the symbols used in the notes

alpha <- unname(coef(reg)[1]) beta <- unname(coef(reg)[2]) fitted.values = alpha + beta*Psoil

Likelihood ratio

n = length(Pcorn) LR = (SS.res/SS.total)^(-n/2)

These are needed in the data.eq_reg table

The residuals

res = Pcorn-fitted.values # Residuals lagged lag1 = c(NA, head(res,-1))

In data.eq_reg table "prob" is hard to calculate - I had to write an new R function to do it!

(never mind about this part it is not very relevant)

inv.qnorm = function(x){ incr = 0.01 y = seq(-100,100,incr) pdf = dnorm(y, mean, sd) cdf = cumsum(pdf)*incr i = min(which(y>x)) val = cdf[i] }

This calculates "prob" which is a column in data.eq_reg

mean <- mean(res) sd<- sd(res) prob = sapply(res, inv.qnorm)

data.eq_reg = data.frame(Pcorn = Pcorn, Model = fitted.values, res = res, res2 = resˆ2, lag1 = lag1, prob = prob, rank = rank(res)) SS.res = sum(data.eq_reg$res2)

more plots (these were in Ch9.1xls - the plots in R are more informative

so these are commented out)

#plot(data.eq_reg$lag1, data.eq_reg$res) #plot(data.eq_reg$Model, data.eq_reg$res) #plot(data.eq_reg$rank, data.eq_reg$prob) # add regression line to last plot #mod = lm(data.eq_reg$prob data.eq_reg$rank) #abline(mod) # New plot to check for residuals #plot(Psoil,lag1-res)

A series of plots to check model assumptions.

plot(reg)

What do we do if the assumptions of the regression aren't met?

It may depend on how the assumptions are violated, *but* something you

can definitely do (and might want to do even if the assumptions are met)

is a randomization experiment. The randomization experiment only assumes

that each value of the response variable (Pcorn) is equally likely to have

been recorded for any of the values of the explanatory variable (Psoil).

This is a null hypothesis that Pcorn has no effect. This randomization

approach is computationally intensive but it makes few assumptions about

the residuals - i.e., nowhere in the randomization experiment do we assume

a normal distribution. In fact, our assumptions about the residuals arise

only from the recorded values of the response variable.

Randomization function - this defines the function generally and then you

just supply your own eta and y.

Conceptually, under the null hypothesis any value of Pcorn could occur

for each of the measured Psoil values (recall the null hypothesis is that

Psoil has no effect)

The function returns the F-value for 1000 randomizations

F.rand = function(eta, y){ result = replicate(10000,anova(lm(sample(y,length(y),TRUE)~eta))$F value[1]) }

Using the randomization function and calling the result 'rand'

This is 10000 random associations of the reported Pcorn values with the Psoil

values.

rand = F.rand(Psoil,Pcorn)

hist(log(rand))

Actual F-value from Corn data - this will be plotted in the figure as a vertical line. Recall the F-value for the data was 12.89

Note the x-axis is on a log scale so the extreme values do not overwhelm the plot.

F.data <- anova(lm(Pcorn~Psoil))$F value[1] lines(c(log(F.data),log(F.data)), c(0,3000), col = "red")

How many F-values from the randomization are bigger than the F-value

for the real data?

The code below finds which values of rand are bigger than the real value. The length function calculates how many rand values are bigger

than the real value (F.data) and this is expressed as a fraction by dividing by 10,000

p.rand = length(which(rand>F.data))/10000 # The code below adds the value of p.rand to the histogram text(-20, 3000, paste0("than data =",p.rand), cex=.8) text(-20, 3500, "fraction random bigger", cex=.8)

The likelihood ratio is >100. Therefore, the linear model with the positive

slope is a lot more likely given the data, than the model assuming no relationship (that the slope is zero).

FWIW this is another way of testing for normality of the residuals:

#mod = lm(data.eq_reg*prob* $data.eq_reg$rank) #plot(data.eq_reg$rank, data.eq_reg$prob) #abline(mod)

‘‘‘