Homework 3

STAT 431

Sebastian Nuxoll

10-06-2024

Data Clean up

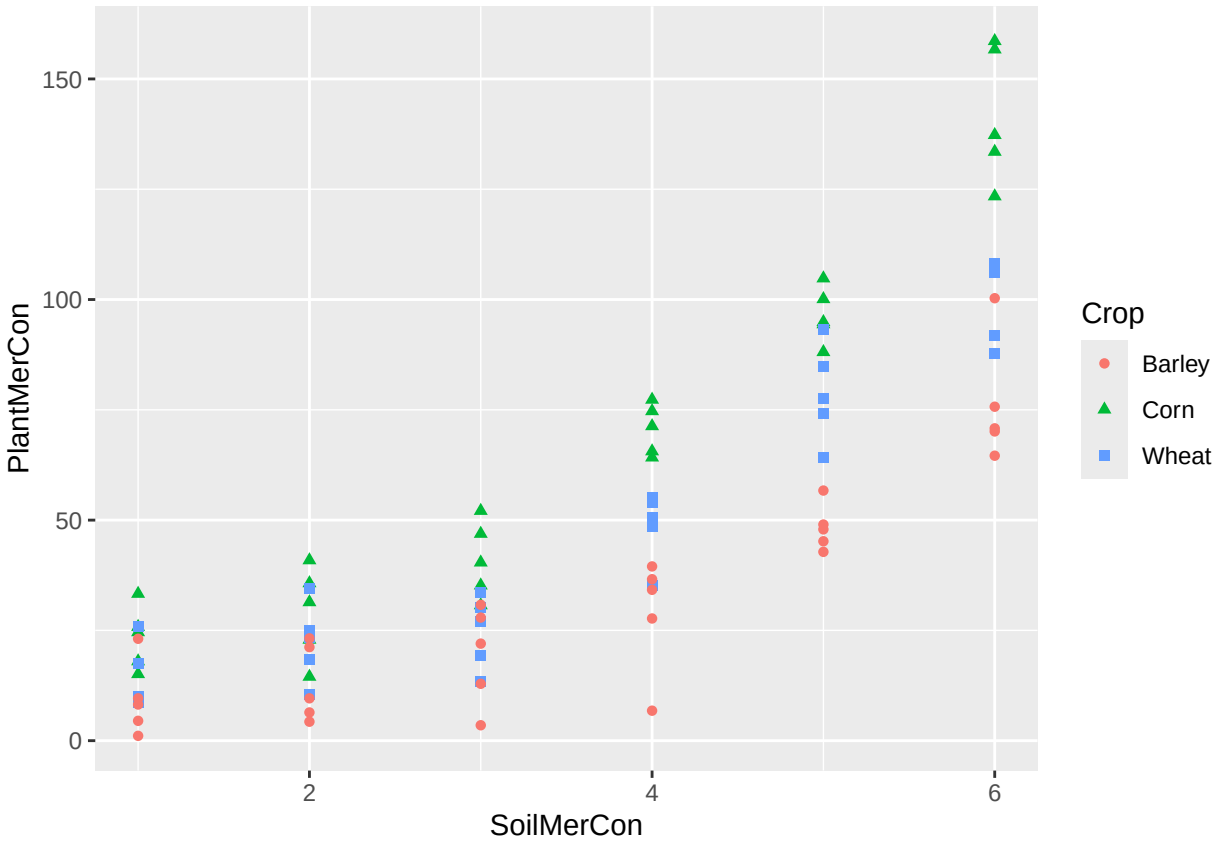
We are looking at a dataset from a .txt file that is formatted similar to a .csv file. We can read it in using `read.csv`, but this will result in some irregularities in the data frame. The column names are in single quotes, as are all of the entries in the `Crop` column. Both of these can be fixed with the `gsub` command. The `Crop` column is also imported as a list of strings, but for our purposes, it would be better as a list of factors. This can be accomplished with the `as.factor` command. We can now see the formatted data below:

```
dat <- read.csv("ex12-35.txt", check.names = F)
colnames(dat) <- gsub("'", "", colnames(dat))
dat$Crop <- as.factor(gsub("'", "", dat$Crop))
head(dat)
```

```
##   SoilMerCon  Crop PlantMerCon
## 1          1  Corn        33.3
## 2          1  Corn        25.8
## 3          1  Corn        24.6
## 4          1  Corn        15.1
## 5          1  Corn        18.0
## 6          1 Wheat        17.4
```

We can now easily plot the data:

```
library(ggplot2)
ggplot(dat, aes(x = SoilMerCon, y = PlantMerCon, color = Crop, shape = Crop)) +
  geom_point()
```



Building Model

We use `model.matrix` to create a design matrix for a model where each crop has its own intercept and slope. The matrix encodes how each crop and the soil mercury concentration contribute to predicting the plant mercury concentration. The hat matrix maps observed responses (plant mercury concentration) to the predicted responses in linear regression. We calculate it using the formula $H = X(X^TX)^{-1}X^T$, where X is the design matrix.

```
x <- model.matrix(~ Crop * SoilMerCon, data = dat)
x_hat <- x %*% solve(t(x) %*% x) %*% t(x)
```

The predicted plant mercury concentration values are calculated by multiplying the hat matrix by the observed response vector (plant mercury concentration). This gives the predictions from the model. The parameter estimates are calculated using the formula $\beta = (X^TX)^{-1}X^TY$.

```
y_hat <- x_hat %*% dat$PlantMerCon
beta <- solve(t(x) %*% x) %*% t(x) %*% dat$PlantMerCon
```

```
rss <- sum((dat$PlantMerCon - y_hat)^2)
tss <- sum((dat$PlantMerCon - mean(dat$PlantMerCon))^2)
regss <- tss-rss

df <- nrow(dat) - length(beta)
r2 <- rss/df
```

We calculate the residual sum of squares (RSS), total sum of squares (TSS), and regression sum of squares (RegSS). These quantities are used to assess how well the model fits the data. RSS (1.4925195×10^4) measures the unexplained variance and TSS (1.2371524×10^5) measures the total variance in the data, so RegSS

(1.0879005×10^5) is the variance explained by the model. The R^2 value is RegSS divided by the degrees of freedom, which is 177.6808889

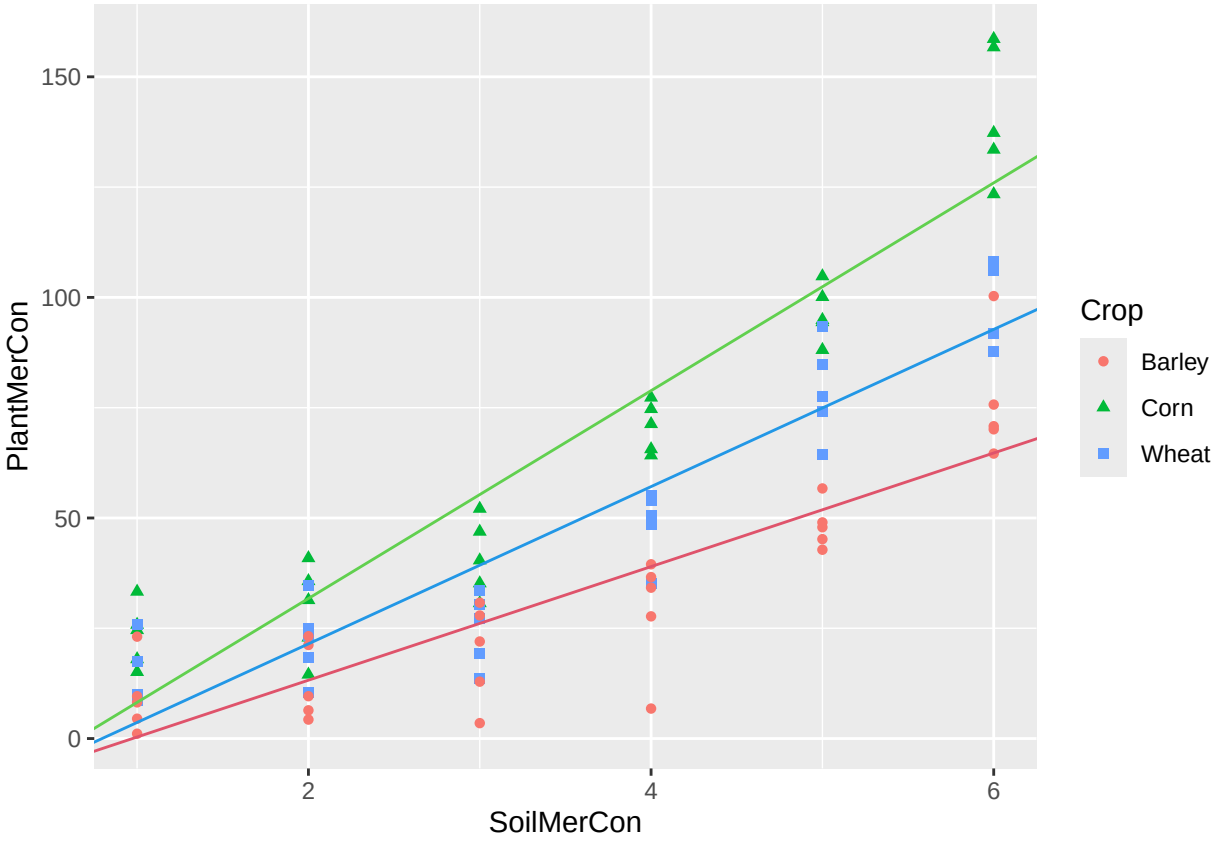
```
betase <- sqrt(diag(solve(t(x) %*% x) * r2))
tvals <- beta / betase
pvals <- 2 * pt(-abs(tvals), df)

f <- regss / (length(beta)-1) / r2
pf <- pf(f, length(beta)-1, df, lower.tail = F)
```

The t-values are calculated for each parameter to test whether the parameter is significantly different from zero. The t-value for `SoilMerCon` is 9.0361599, which is quite high, suggesting that the primary correlation is that more mercury in the soil means more mercury in the plant. The t-values for corn and wheat by themselves (-0.3608418 and -0.2101513 repectively) are close to zero, meaning that with no mercury in the soil, there isn't significantly more mercury in corn or wheat than barley. The t-values for wheat and corn along `SoilMerCon` (5.2984271 and 2.4535636 repectively) are reasonably large, meaning wheat absorbs more mercury from the soil than barley, and corn absorbs more than both of them.

The p-values correspond to the probability of observing a t-value as extreme as the one calculated, assuming the null hypothesis is true (i.e., the coefficient is zero). We can see that the smaller p-values correspond with larger t-values. The F-value is used to test whether all the slopes are equal to zero. The p-value for the F-value assesses the significance of the F-value. The p-value of F is $4.6937082 \times 10^{-37}$ which is incredibly small and provides a lot of evidence that there is a underlying relationship.

With this analysis, we can find the equations for fit lines for each of our crops. For barley, we have $y = 12.8765714x + -12.528$. For corn, we have $y = 23.5542857x + -15.36$. For wheat, we have $y = 17.8211429x + -14.1773333$. We can see how these lines compare to our data in the plot below:



Building Model w/ `lm()`

```
model <- lm(PlantMerCon ~ Crop * SoilMerCon, data = dat)
redmodel <- lm(PlantMerCon ~ SoilMerCon + Crop:SoilMerCon, data = dat)
anovares <- anova(redmodel, model)
```

We can create two different models with the `lm` function. One has different slopes and intercepts for each crop, while the other has only one intercept that is used for all the crops. We can compare these plots using `anova` to see if the distinct intercepts provide a significant improvement. Our anova gives us a p-value of 0.9364667, which is enough to warrant using the more complicated model with multiple intercepts. We compare out model to our data with an 85% prediction interval in the plot below: