Exam 2

Amy Fan

19 Nov 2023

```r
library(dplyr)
library(glmnet)
library(ggplot2)
```

II. Tree Rings in AZ[26pts]

The data for this exam are available on D2L (tree_rings.csv) and were originally gathered from multiple sources for the analysis in Heilman et al. (2022).1 There are a number of important variables, many of which have been standardized to be mean 0 with a sample variance 1.

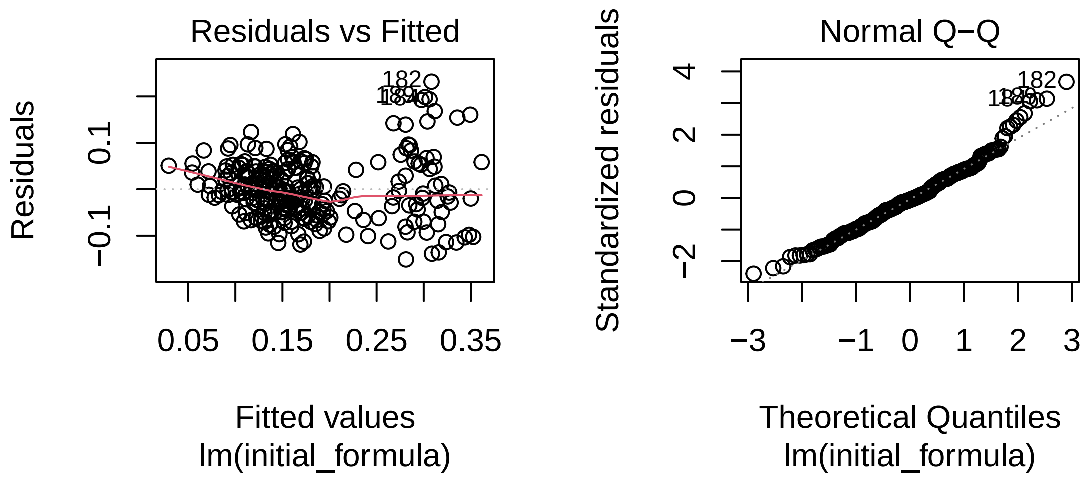
Variable Meaning: treeid: Individual tree identifier. year: Year measurement was made. SICOND: Site condition. Measure of the overall quality of the site for the tree (higher is better). Standardized. SDI: Stand density index. Measure of the density of trees in the vicinity (higher is more trees per area). Standardized. MAP: Mean annual precipitation. Standardized. MAT: Mean annual temperature. Standardized. width: Tree ring width in centimeters. Water_year_precip: Total precipitation for year corresponding to tree ring width. Standardized. Fall_spring_tmax: Maximum temperature through seasons of fall, winter, and spring immediately before tree ring width measurement.

As trees grow, they form "rings" in their main stem as they expand outward. Each ring corresponds to a single year, and the width of each ring is related to several important conditions that affect trees' capacity for growth, including weather and climate. Tree rings are studied for several important scientific reasons, including dendrochronology (for which the University of Arizona is well known). The goal of the following analyses is to better understand what effects several different variables have on tree ring width.

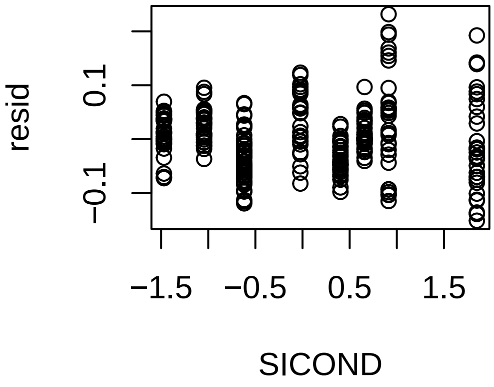
(5) [3 pts] Fit a linear regression model for tree ring width as a function of SICOND, SDI, MAP, MAT, Water_year_precip, and Fall_spring_tmax. Create appropriate diagnostic plots and comment on the validity of the assumptions of (i) constant variance and (ii) normally distributed residuals.

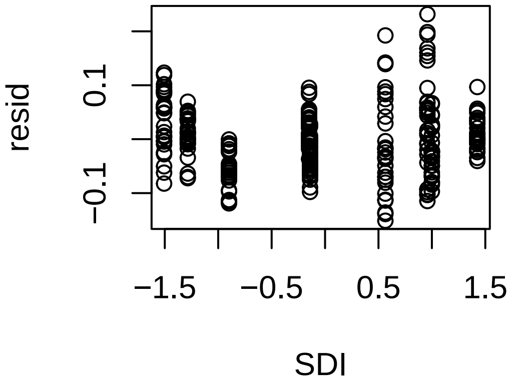
Variance appears to be consistent based on the residuals v. fitted and residuals v. predictor plots. There is a slight curve in the SDI, MAP, MAT v. residuals. The QQ plot shows a heavier upper tail which indicates nonnormality.

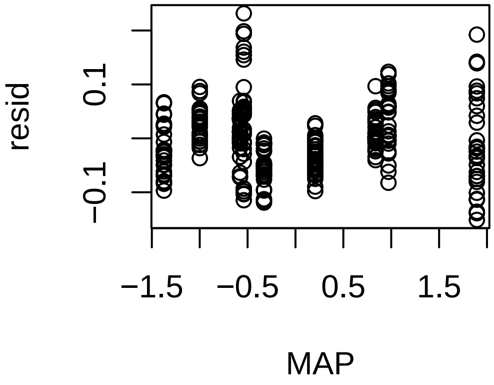
```r
treering <- read.csv('./Datasets/tree_rings.csv')
initial_formula <- width ~ SICOND + SDI + MAP + MAT + Water_year_precip + Fall_spring_tmax
tr_fit <- lm(initial_formula, data = treering)
plot(tr_fit, c(1,2)) # Fitted v. Residuals, QQ-plot
```

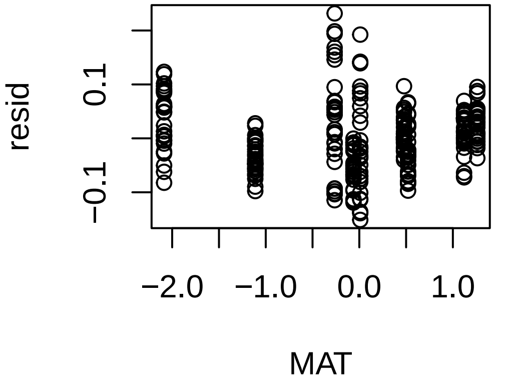


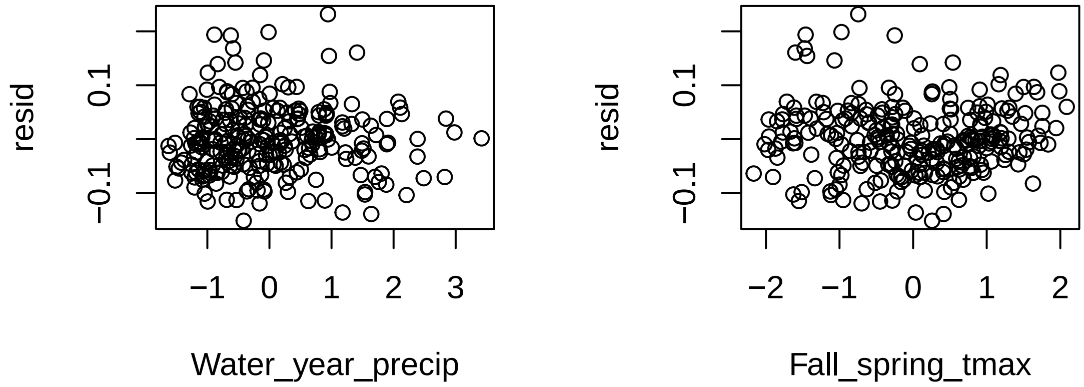
```
# plotting predictors v. residuals
plot_pvr <- function(df, model, cts_pred, cat_pred) {
  resid <- model$residuals
  if (length(cts_pred) != 0){
    for (pred in cts_pred){
    predictor <- df[[pred]]
    plot(predictor, resid, xlab = pred)
    }
  }
  if (length(cat_pred) != 0){
    for (pred in cat_pred) {
      plot(as.factor(df[[pred]]), resid, xlab = pred)
      }
    }
}

cts_pred <- c('SICOND', 'SDI', 'MAP', 'MAT', 'Water_year_precip','Fall_spring_tmax')
cat_pred <- c()
plot_pvr(treering, tr_fit, cts_pred, cat_pred)
```


resid
0.1
−0.1
−1.5
−0.5
0.5
1.5
SICOND


resid
0.1
−0.1
−1.5
−0.5
0.5
1.5
SDI


resid
0.1
−0.1
−1.5
−0.5
0.5
1.5
MAP

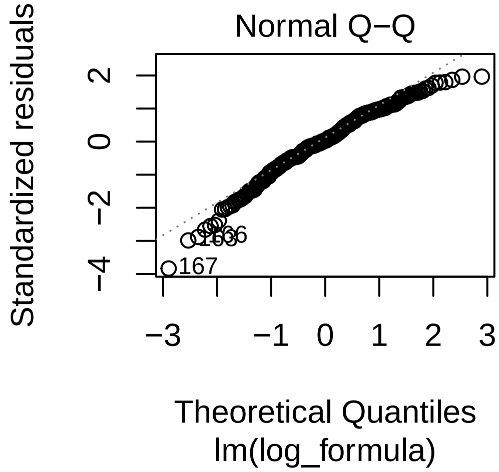

resid
0.1
−0.1
−2.0
−1.0
0.0
1.0
MAT



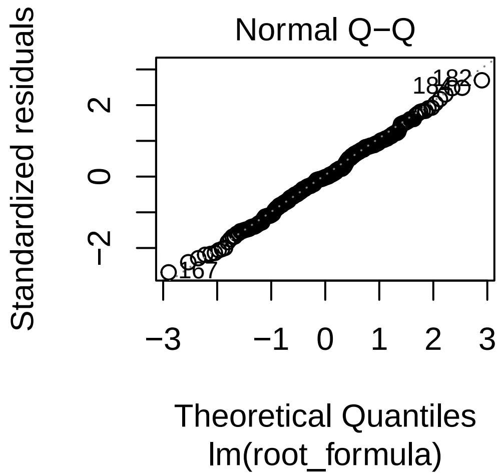
(6) [3 pts] Fit two more linear regression models with the same predictors, but this time for two transformed responses: (i) the log of tree ring width, and (ii) the square root of tree ring width. Which model appears to better satisfy the assumption of normally distributed residuals? Explain your conclusion.

The rootfit appears to have more normally distributed residuals which can be seen in the QQ plot. The log fit appears to have fatter tails on the QQ which indicate nonnormal residuals.

```r
log_formula <- log(width) ~ SICOND + SDI + MAP + MAT + Water_year_precip + Fall_spring_tmax
tr_logfit <- lm(log_formula, data = treering)

root_formula <- width^(1/2) ~ SICOND + SDI + MAP + MAT + Water_year_precip + Fall_spring_tmax
tr_rootfit <- lm(root_formula, data = treering)

plot(tr_logfit, 2)
```



```r
plot(tr_rootfit, 2)
```



(7) The following scatterplot shows the square root of tree ring width as a function of Water_year_precip with points colored according to SICOND. Simple linear regressions fit separately for each observed level of SICOND are shown in matching colors.

(a) [2 pts] Based on the figure, do you see evidence of a non-zero interaction effect between SICOND and Water_year_precip? What specific features visible in the figure led you to your conclusion?

There appears to be an interaction between SICOND and Water_year_precip since the slopes in the figure are not parallel. Higher SICOND values (Darker, warmer) tend to have negative slopes while lower SICOND values(Lighter, cooler) tend to have more positive slopes.

(b) [2 pts] Conduct an appropriate test to determine whether the null hypothesis of a zero interaction effect should be rejected at the level $/alpha = 0.01$ after accounting for all other predictors mentioned in (5). Report your p-value and conclusion

Because the test is conducted "after" other predictors have been accounted for, we do not conduct family-wise inference/bonferroni correction. Evaluated at alpha = 0.01, the interaction term appears to be significant (t test for interaction term has p-value 0.0058).

```r
inter_root_formula <- width^(1/2) ~ SDI + MAP + MAT + Water_year_precip*SICOND + Fall_spring_tmax
summary(lm(inter_root_formula, data = treering))
```

```
##
## Call:
## lm(formula = inter_root_formula, data = treering)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.179812 -0.045422  0.001639  0.054794  0.196621
##
```

```
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 0.404121   0.004438  91.054  < 2e-16 ***
## SDI                        -0.015441   0.011969  -1.290  0.19819    
## MAP                        -0.022301   0.013745  -1.622  0.10593    
## MAT                        -0.021092   0.008744  -2.412  0.01655 *  
## Water_year_precip           0.021150   0.005280   4.006 8.08e-05 ***
## SICOND                      0.101344   0.017581   5.764 2.32e-08 ***
## Fall_spring_tmax           -0.039835   0.008236  -4.836 2.27e-06 ***
## Water_year_precip:SICOND -0.012643   0.004547  -2.781  0.00582 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07045 on 259 degrees of freedom
## Multiple R-squared:  0.5854, Adjusted R-squared:  0.5742 
## F-statistic: 52.24 on 7 and 259 DF,  p-value: < 2.2e-16
```

(8) (a) [2 pts] Examine the marginal plots below of square root of tree ring width as a function of SDI, MAP, and MAT. For which of the three predictor variables do you see evidence that a quadratic fit would be more appropriate than a linear one? Your answer may include more than one variable. Explain what specifically you see in the figure(s) that led you to your conclusion.

There appears to possibly be a positive quadratic relationship for SDI because of the higher edges and dip in the middle values. MAT and MAP appear to possibly have negative quadratic relationships since the middle values are peaked higher than the edge values.

(b) [2 pts] Fit a regression model for the square root of tree ring width as a function of Fall_spring_tmax, Water_year_precip, SICOND, the interaction between SICOND and Water_year_precip, and both linear and quadratic effects for SDI, MAP, and MAT. For which quadratic effects is there sufficient evidence to reject a null hypothesis of zero effect at a familywise error rate of $\alpha = 0.05$?

There are 3 quadratic terms, so the corrected alpha = 0.01666667. The quadratic term for "SDI" and "MAT" is significant with a p-values of 0.000201 and 4.80e-16, respectively.

```
## [1] 0.01666667
```

```
formula_quad <- sqrt(width) ~ Fall_spring_tmax + Water_year_precip*SICOND +
  poly(SDI, 2) + poly(MAP, 2) + poly(MAT, 2)
summary(lm(formula_quad, data = treering))
```

```
##
## Call:
## lm(formula = formula_quad, data = treering)
##
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.191796 -0.033687 -0.001821  0.039659  0.162636 
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 0.404236   0.003782 106.893  < 2e-16 ***
## Fall_spring_tmax           -0.036305   0.008272  -4.389 1.67e-05 ***
## Water_year_precip         0.007926   0.004786   1.656 0.098909 .
## SICOND                    0.218626   0.022307   9.801  < 2e-16 ***
## poly(SDI, 2)1            -0.892691   0.243015  -3.673 0.000291 ***
## poly(SDI, 2)2             0.368484   0.097671   3.773 0.000201 ***
## poly(MAP, 2)1            -1.531274   0.275585  -5.556 6.90e-08 ***
## poly(MAP, 2)2            -0.083663   0.076353  -1.096 0.274223
## poly(MAT, 2)1             0.251435   0.165990   1.515 0.131066
## poly(MAT, 2)2             0.864642   0.099648   8.677 4.80e-16 ***
## Water_year_precip:SICOND -0.012149   0.003894  -3.120 0.002015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06001 on 256 degrees of freedom
## Multiple R-squared:  0.7027, Adjusted R-squared:  0.6911
## F-statistic:  60.5 on 10 and 256 DF,  p-value: < 2.2e-16
```

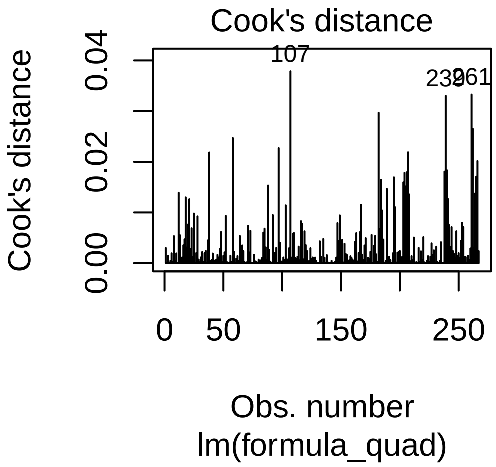
(9) [2 pts] Compute DFFITS and DFBETAS for the model in (8b). Is there evidence of influential outliers in the data? If so, for which observation indices? If not, explain your reasoning.

DFFITS revealed that 107, 261, 239, and 182 have the strongest influence. DFBETAS revealed that the greatest influence on parameters came from 207 and 265 affecting the linear MAP and intercept terms, respectively.

Investigating the DFBETAS for Obs. 107 which had the highest DFFITS and Cook's leverage, I found that DFBETAS was highest for SDI's quadratic term (DFBETAS = -0.23220194). However, this DFBETAS values was relatively small in comparison to other DFBETAS values.

Altogether, the lack of overlap between DFFITS and DFBETAS indicate that there are likely no influential outliers.

```r
# look at cook's plot to see obs to expect
plot(lm(formula_quad, data = treering), 4) # 107, 261, 239
```



```r
# DFFITS
getDFFITS <- function(model, data){
  n <- nrow(data)
  p <- length(model$coefficients)
  SSE <- sum(model$residuals^2)
  X <- model.matrix(model)
  H <- X %*% solve(t(X) %*% X) %*% t(X)

  e <- rep(0,n)
  t <- rep(0,n)
  DFFITS <- rep(0,n)

  for (i in 1:n) {
    e[i] <- model$residuals[i]
    t[i] <- e[i] * ( (n-p-1) / ( SSE*(1-H[i,i]) - e[i]^2 ) )^(1/2)
    DFFITS[i] <- t[i] * ( (H[i,i]) / (1 - H[i,i]) )^(1/2)
  }
  return(DFFITS)
}

tr_8bfit <- lm(formula_quad, data = treering)
tr_8b_DFFITS <- getDFFITS(tr_8bfit, treering)

posDFFITS_vals <- sort(tr_8b_DFFITS, decreasing = T)[1:2]
negDFFITS_vals <- sort(tr_8b_DFFITS, decreasing = F)[1:2]
DFFITS_i <- match(c(posDFFITS_vals,negDFFITS_vals),tr_8b_DFFITS)

DFFITS_comp <- data.frame(
  treering[DFFITS_i,
           c('width', 'Fall_spring_tmax','Water_year_precip', 'SICOND', 'SDI', 'MAP', 'MAT')
           ], DFFITS = tr_8b_DFFITS[DFFITS_i])
DFFITS_comp # 107, 239, 261, 182
```

```
##     width Fall_spring_tmax Water_year_precip     SICOND        SDI        MAP
## 239  0.49       -0.2500042        -0.6229344  1.8490893  0.5646962  1.8944387
## 182  0.54       -0.7456187         0.9433239  0.9130715  0.9579344 -0.5396904
## 107  0.05        1.6289074        -0.8589369 -0.0229463 -1.5064586  0.9712476
## 261  0.14        1.0227096        -1.0436816  1.8490893  0.5646962  1.8944387
##              MAT     DFFITS
## 239  0.009864167  0.6102076
## 182 -0.263732283  0.5791168
## 107 -2.086472213 -0.6578482
## 261  0.009864167 -0.6097398
```

```
# DFBETAS
getDFBETAS <- function(model,df){
  n <- nrow(df)
  p <- length(model$coefficients)
  X <- model.matrix(model)
  original_formula <- eval(model$call[[2]])

  DFBETAS <- as.data.frame(matrix(0, nrow=n, ncol=p, byrow=T))
  names(DFBETAS) <- names(model$coefficients)
  rownames(DFBETAS) <- 1:n

  for_c <- solve(t(X)%*%X)
  for (i in 1:n){
    diff_fit <- lm(original_formula, data = df[-c(i),])
    MSE_diff <- mean(diff_fit$residuals^2)

    for (k in 1:p) {
      b_k <- model$coefficients[[k]]
      b_k_diff <- diff_fit$coefficients[[k]]
      c_kk <- for_c[k,k]
      DFBETAS[i,k] <- (b_k - b_k_diff) / sqrt(MSE_diff * c_kk)
    }
  }
  return(DFBETAS)
}

tr_8b_DFBETA <- getDFBETAS(tr_8bfit, treering)

posDFBETA_val <- max(tr_8b_DFBETA)
negDFBETA_val <- -max(-tr_8b_DFBETA)
tr_8b_DFBETA <- data.frame(tr_8b_DFBETA, removed_obs = 1:nrow(treering))

# 207 (linear MAP term) and 265 (intercept term)
tr_8b_DFBETA %>%
  filter(if_any(.cols = everything(),
                .fns = ~ .x %in% c(posDFBETA_val, negDFBETA_val)) |
           row_number() == 107)
```

```
##   X.Intercept. Fall_spring_tmax Water_year_precip      SICOND poly.SDI..2.1
## 1   -0.1113932      -0.13409150         0.1233219 -0.15066816   0.139728046
## 2   -0.1681587      -0.15323467         0.1111087 -0.38417317   0.280435258
## 3   -0.4407517      -0.02095603        -0.2443851  0.02898446  -0.002302571
##   poly.SDI..2.2 poly.MAP..2.1 poly.MAP..2.2 poly.MAT..2.1 poly.MAT..2.2
## 1   -0.23220194      0.1477451   -0.15923564   -0.05619756   -0.13118016
## 2   -0.25853252      0.3786320    0.05172886   -0.24086783   -0.05952725
## 3    0.05497394     -0.1191754   -0.05493123   -0.02620135    0.06888591
##   Water_year_precip.SICOND removed_obs
## 1               0.01706619         107
## 2               0.14900475         207
## 3              -0.30507537         265
```

(10) [3 pts] Use the glmnet R package to obtain penalized regression coefficient estimates for the model in (8b) based on a LASSO penalty of $\lambda = 0.002$. Which variables have coefficient estimates of zero? Do you agree they should be removed from the model? Why or why not?

The linear SDI term has a 0 estimate. However, since the quadratic SDI term is significant, the linear term should be kept in the model.

```
X <- model.matrix(formula_quad, data = treering)
coef(glmnet(X, sqrt(treering$width)), s = 0.002)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                   s1
## (Intercept)                0.405501690
## (Intercept)                .
## Fall_spring_tmax         -0.055479182
## Water_year_precip         0.008886399
## SICOND                    0.117778269
## poly(SDI, 2)1             .
## poly(SDI, 2)2             0.076664500
## poly(MAP, 2)1            -0.328244433
## poly(MAP, 2)2            -0.062590692
## poly(MAT, 2)1            -0.268423314
## poly(MAT, 2)2             0.684998048
## Water_year_precip:SICOND -0.006689806
```

(11) [3 pts] Obtain estimates for both the main effects of SICOND and Water_year_precip and their interaction based on the model in (8b). Give an interpretation of the coefficients in the context of the application.

From previous results: Water_year_precip 0.008886399 SICOND 0.117778269 Water_year_precip:SICOND -0.006689806

Interpretation: As precipitation increases by a unit and site condition is at 0, width increases by 0.008886399. As site condition increases by a unit and precipitation is at 0, width increases by 0.117778269. However, both of their main effects interfere with each other. Site condition's unit effect on width when precipitation also increases by a unit is 0.1110885. Conversely, the positive effect of precipitation when site condition increases by a unit is 0.002196593.

As precipitation increases, better site conditions have less of a positive effect on tree ring width.

```
0.117778269 - 0.006689806 # site condition effect with interaction
```

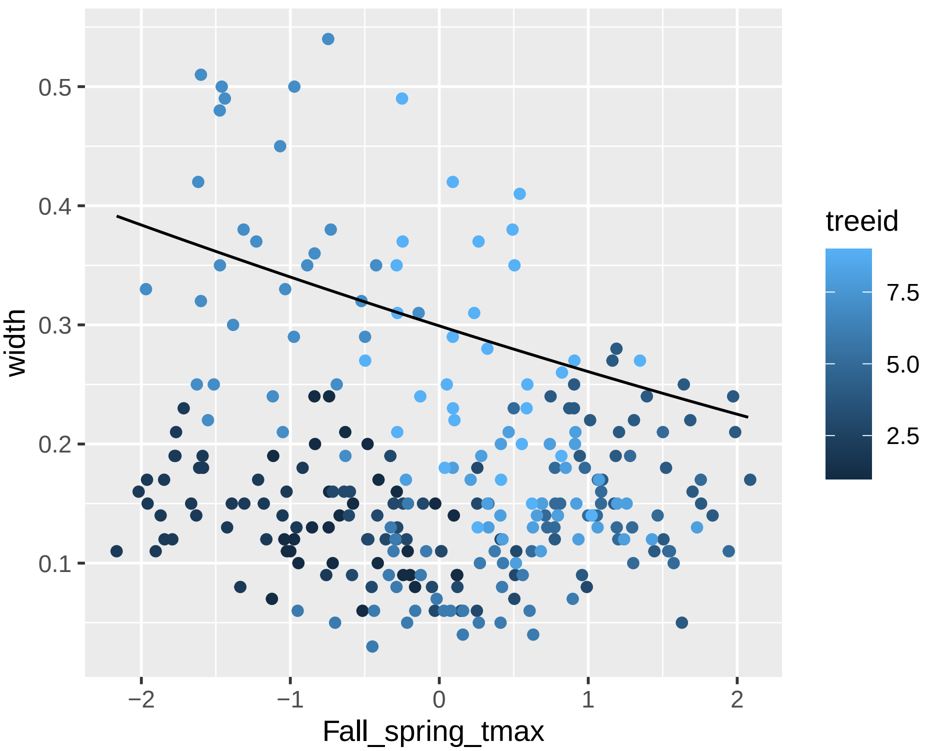
```
## [1] 0.1110885
```

```
0.008886399 - 0.006689806 # precipitation effect with interaction
```

```
## [1] 0.002196593
```

(12) [4 pts] Make a scatter plot of tree ring width as a function of Fall_spring_tmax. Color each point according to treeid. Add a curve that shows the predicted tree ring width across a range of values for Fall_spring_tmax with all other predictors set to the mean observed values for the tree with treeid = 7.

```
predictors <- c('Water_year_precip', 'SICOND', 'SDI', 'MAP', 'MAT')
tree7_subset <- treering %>%
  filter(treeid == 7)
fst_grid <- seq(min(treering$Fall_spring_tmax),max(treering$Fall_spring_tmax), by = .04)
tree7_mean_values <- data.frame(t(colMeans(tree7_subset[, predictors], na.rm = TRUE)))
tree7_mean_values_rep <- tree7_mean_values[rep(1, each = length(fst_grid)),]

newdata <- data.frame(tree7_mean_values_rep, Fall_spring_tmax = fst_grid)

predictions <- as.data.frame(cbind(newdata, predict(tr_8bfit, newdata = newdata)))
colnames(predictions)[7] <- 'fit'

ggplot(treering) +
  geom_point(aes(x = Fall_spring_tmax, y = width, color = treeid)) +
  geom_line(aes(x = Fall_spring_tmax, y = fit^2), data = predictions)
```



width
0.5
0.4
0.3
0.2
0.1
−2
−1
0
1
2
Fall_spring_tmax
treeid
7.5
5.0
2.5