Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
# reading libaries
library(tidyverse)
library(ModelMetrics)
library(MASS)
# reading data
# note: data was obtained through a given docx,
# which I made into a google doc, then copy pasted to google sheets,
# then saved as a csv
# note: the data we were given is about 10% of the data they used,
# so our graphs will look slightly different
metasequoia <- read_csv("data/metasequoia_data.csv")
# data exploration
metasequoia %>%
  pivot_longer(col = c("height", "diameter"),
               names_to = "datatype",
               values_to = "values") %>%
  group_by(datatype) %>%
  summarise(mean = mean(values),
            max = max(values),
            min = min(values),
            sd = sd(values)) %>%
  t()
# models
metasequoia_model1 <- lm(height ~ diameter, data = metasequoia)
metasequoia_model2 <- lm(height ~ I(log(diameter)), data = metasequoia)
metasequoia_model3 <- lm(height ~ diameter + I(diameter^2), data = metasequoia)
metasequoia_model4 <- lm(height ~ I(diameter^2) + I(diameter^3), data = metasequoia)
metasequoia_model5 <- lm(height ~ I(diameter^-1) + I(diameter^2), data = metasequoia)
# Fig 2. Scatter diagram of the tree height and dbh of a single Metasequoia tree.
plot(height ~ diameter, data = metasequoia, main = "Scatterplot of Height and Diameter",
     xlab = "Diameter (at breast height in cm)", ylab = "Height (in meters)")
abline(a = 12.546, b = 0.264) # the paper's data's trendline
abline(metasequoia_model1, col = "red") # trendline for model 1
point_color <- rgb(0, 0, 0, alpha = 0.25)
par(mfrow = c(2, 3))
plot(height ~ diameter, data = metasequoia, main = "Height vs Diameter (Model 1)",
     xlab = "Diameter (at breast height in cm)", ylab = "Height (in meters)", col = point_color)
x <- with(metasequoia, seq(min(diameter), max(diameter), length.out=2000))
y <- predict(metasequoia_model1, newdata = data.frame(diameter = x))
lines(x, y, col = "red", lwd = 5)
plot(height ~ diameter, data = metasequoia, main = "Height vs Diameter (Model 2)",
     xlab = "Diameter (at breast height in cm)", ylab = "Height (in meters)", col = point_color)
x <- with(metasequoia, seq(min(diameter), max(diameter), length.out=2000))
y <- predict(metasequoia_model2, newdata = data.frame(diameter = x))
lines(x, y, col = "orange", lwd = 5)
plot(height ~ diameter, data = metasequoia, main = "Height vs Diameter (Model 3)",
     xlab = "Diameter (at breast height in cm)", ylab = "Height (in meters)", col = point_color)
x <- with(metasequoia, seq(min(diameter), max(diameter), length.out=2000))
y <- predict(metasequoia_model3, newdata = data.frame(diameter = x))
lines(x, y, col = "green", lwd = 5)
```

```r
plot(height ~ diameter, data = metasequoia, main = "Height vs Diameter (Model 4)",
     xlab = "Diameter (at breast height in cm)", ylab = "Height (in meters)", col = point_color)
x <- with(metasequoia, seq(min(diameter), max(diameter), length.out=2000))
y <- predict(metasequoia_model4, newdata = data.frame(diameter = x))
lines(x, y, col = "blue", lwd = 5)
plot(height ~ diameter, data = metasequoia, main = "Height vs Diameter (Model 5)",
     xlab = "Diameter (at breast height in cm)", ylab = "Height (in meters)", col = point_color)
x <- with(metasequoia, seq(min(diameter), max(diameter), length.out=2000))
y <- predict(metasequoia_model5, newdata = data.frame(diameter = x))
lines(x, y, col = "purple", lwd = 5)
#par(mfrow = c(2, 3))
# making residuals plot for model 1
plot(resid(metasequoia_model1) ~ predict(metasequoia_model1),
     main = "Residual Plot for Model 1", xlab = "Predicted Values", ylab = "Residuals")
abline(h = 0,col = "red",lty = 2)
# making residuals plot for model 2
plot(resid(metasequoia_model2) ~ predict(metasequoia_model2),
     main = "Residual Plot for Model 2", xlab = "Predicted Values", ylab = "Residuals")
abline(h = 0,col = "red",lty = 2)
# making residuals plot for model 3
plot(resid(metasequoia_model3) ~ predict(metasequoia_model3),
     main = "Residual Plot for Model 3", xlab = "Predicted Values", ylab = "Residuals")
abline(h = 0,col = "red",lty = 2)
# making residuals plot for model 4
plot(resid(metasequoia_model4) ~ predict(metasequoia_model4),
     main = "Residual Plot for Model 4", xlab = "Predicted Values", ylab = "Residuals")
abline(h = 0,col = "red",lty = 2)
# making residuals plot for model 5
plot(resid(metasequoia_model5) ~ predict(metasequoia_model5),
     main = "Residual Plot for Model 5", xlab = "Predicted Values", ylab = "Residuals")
abline(h = 0,col = "red",lty = 2)
#par(mfrow = c(2, 3))
# making qq plot for model 1
qqnorm(resid(metasequoia_model1), main = "Q-Q Plot for Model 1", col = "red")
qqline(resid(metasequoia_model1))
# making qq plot for model 2
qqnorm(resid(metasequoia_model2), main = "Q-Q Plot for Model 2", col = "red")
qqline(resid(metasequoia_model2))
# making qq plot for model 3
qqnorm(resid(metasequoia_model3), main = "Q-Q Plot for Model 3", col = "red")
qqline(resid(metasequoia_model3))
# making qq plot for model 4
qqnorm(resid(metasequoia_model4), main = "Q-Q Plot for Model 4", col = "red")
qqline(resid(metasequoia_model4))
# making qq plot for model 5
qqnorm(resid(metasequoia_model5), main = "Q-Q Plot for Model 5", col = "red")
qqline(resid(metasequoia_model5))
#par(mfrow = c(2, 3))
# predicted vs observed for model 1
plot(height ~ predict(metasequoia_model1), data = metasequoia,
     main = "Observed vs Predicted in Model 1", xlab = "Predicted", ylab = "Observed")
abline(a = 0, b = 1, col = "red")
# predicted vs observed for model 2
```

```r
plot(height ~ predict(metasequoia_model2), data = metasequoia,
     main = "Observed vs Predicted in Model 2", xlab = "Predicted", ylab = "Observed")
abline(a = 0, b = 1, col = "red")
# predicted vs observed for model 3
plot(height ~ predict(metasequoia_model3), data = metasequoia,
     main = "Observed vs Predicted in Model 3", xlab = "Predicted", ylab = "Observed")
abline(a = 0, b = 1, col = "red")
# predicted vs observed for model 4
plot(height ~ predict(metasequoia_model4), data = metasequoia,
     main = "Observed vs Predicted in Model 4", xlab = "Predicted", ylab = "Observed")
abline(a = 0, b = 1, col = "red")
# predicted vs observed for model 5
plot(height ~ predict(metasequoia_model5), data = metasequoia,
     main = "Observed vs Predicted in Model 5", xlab = "Predicted", ylab = "Observed")
abline(a = 0, b = 1, col = "red")
# calculating bias
mean((predict(metasequoia_model1) - metasequoia$height) / metasequoia$height) * 100
mean((predict(metasequoia_model2) - metasequoia$height) / metasequoia$height) * 100
mean((predict(metasequoia_model3) - metasequoia$height) / metasequoia$height) * 100
mean((predict(metasequoia_model4) - metasequoia$height) / metasequoia$height) * 100
mean((predict(metasequoia_model5) - metasequoia$height) / metasequoia$height) * 100
# calculating RMSE
# we want the lowest value which is model 4
rmse(metasequoia_model1) # can also use: rmse(metasequoia$height, predict(metasequoia_model1))
rmse(metasequoia_model2)
rmse(metasequoia_model3)
rmse(metasequoia_model4)
rmse(metasequoia_model5)
# calculating AIC
# we want the lowest value which is model 4
AIC(metasequoia_model1)
AIC(metasequoia_model2)
AIC(metasequoia_model3)
AIC(metasequoia_model4)
AIC(metasequoia_model5)
# calculating BIC
# we want the lowest value which is model 4
BIC(metasequoia_model1)
BIC(metasequoia_model2)
BIC(metasequoia_model3)
BIC(metasequoia_model4)
BIC(metasequoia_model5)
# calculating R^2adj
# we want the highest value which is model 4
summary(metasequoia_model1)$adj.r.squared
summary(metasequoia_model2)$adj.r.squared
summary(metasequoia_model3)$adj.r.squared
summary(metasequoia_model4)$adj.r.squared
summary(metasequoia_model5)$adj.r.squared
# calculating CIs
confint(metasequoia_model1, level = 1-0.05)
confint(metasequoia_model2, level = 1-0.05)
confint(metasequoia_model3, level = 1-0.05)
```

```r
confint(metasequoia_model4, level = 1-0.05) # this one
confint(metasequoia_model5, level = 1-0.05)
#Data processing
sequoia = read.csv("data/metasequoia_data.csv")
sequoia$log.diameter <- log10(sequoia$diameter)
sequoia$squared.diameter <- (sequoia$diameter)^2
sequoia$cubic.diameter <- (sequoia$diameter)^3
sequoia$diameter.to.the.power.of.negativeone <- (sequoia$diameter)^-1

#Model Selection
full.model = lm(height ~ diameter + squared.diameter + cubic.diameter +
                  log.diameter + diameter.to.the.power.of.negativeone, data = sequoia)
empty.model = lm(height ~ 1,data = sequoia)

n = nrow(sequoia)

forward.model.AIC = stepAIC(empty.model, scope = list(lower = empty.model,
              upper= full.model), k = 2,direction = "forward",trace = FALSE)
forward.model.BIC = stepAIC(empty.model,  scope = list(lower = empty.model,
              upper= full.model), k = log(n),trace=FALSE,direction = "forward")
backward.model.AIC = stepAIC(full.model, scope = list(lower = empty.model,
              upper= full.model), k = 2,direction = "backward",trace = FALSE)
backward.model.BIC = stepAIC(full.model,  scope = list(lower = empty.model,
              upper= full.model), k = log(n),trace=FALSE,direction = "backward")
FB.model.AIC = stepAIC(empty.model, scope = list(lower = empty.model,
              upper= full.model), k = 2,direction = "both",trace = FALSE)
FB.model.BIC = stepAIC(empty.model,  scope = list(lower = empty.model,
              upper= full.model), k = log(n),trace=FALSE,direction = "both")
BF.model.AIC = stepAIC(full.model, scope = list(lower = empty.model,
              upper= full.model), k = 2,direction = "both",trace = FALSE)
BF.model.BIC = stepAIC(full.model,  scope = list(lower = empty.model,
              upper= full.model), k = log(n),trace=FALSE,direction = "both")
model4 = lm(height ~  squared.diameter + cubic.diameter, data = sequoia)
#Calculating AIC
AIC(forward.model.AIC)
AIC(forward.model.BIC)
AIC(backward.model.AIC)
AIC(backward.model.BIC)
AIC(FB.model.AIC)
AIC(FB.model.BIC)
AIC(BF.model.AIC)
AIC(BF.model.BIC)
AIC(model4)
#Calculating BIC
BIC(forward.model.AIC)
BIC(forward.model.BIC)
BIC(backward.model.AIC)
BIC(backward.model.BIC)
BIC(FB.model.AIC)
BIC(FB.model.BIC)
BIC(BF.model.AIC)
BIC(BF.model.BIC)
BIC(model4)
```

```r
#New Best Models
best.AIC.model = backward.model.AIC
best.BIC.model = forward.model.BIC
model4 = lm(height ~  squared.diameter + cubic.diameter, data = sequoia)
summary(best.AIC.model)
summary(best.BIC.model)
summary(model4)
sequoia$ei = best.AIC.model$residuals
sequoia$yhat = best.AIC.model$fitted.values

ei = best.AIC.model$residuals
the.SWtest = shapiro.test(ei)
the.SWtest

Group = rep("Lower",nrow(sequoia))
Group[sequoia$height < median(sequoia$height)] = "Upper"
Group = as.factor(Group)
sequoia$Group = Group
the.FKtest= fligner.test(sequoia$ei, sequoia$Group)
the.FKtest

#B
sequoia$ei = best.BIC.model$residuals
sequoia$yhat = best.BIC.model$fitted.values

ei = best.BIC.model$residuals
the.SWtest = shapiro.test(ei)
the.SWtest

Group = rep("Lower",nrow(sequoia))
Group[sequoia$height < median(sequoia$height)] = "Upper"
Group = as.factor(Group)
sequoia$Group = Group
the.FKtest= fligner.test(sequoia$ei, sequoia$Group)
the.FKtest

#C
sequoia$ei = model4$residuals
sequoia$yhat = model4$fitted.values

ei = model4$residuals
the.SWtest = shapiro.test(ei)
the.SWtest

Group = rep("Lower",nrow(sequoia))
Group[sequoia$height < median(sequoia$height)] = "Upper"
Group = as.factor(Group)
sequoia$Group = Group
the.FKtest= fligner.test(sequoia$ei, sequoia$Group)
the.FKtest

qqnorm(best.AIC.model$residuals)
qqline(best.AIC.model$residuals)
```

```r
qqnorm(best.BIC.model$residuals)
qqline(best.BIC.model$residuals)

qqnorm(model4$residuals)
qqline(model4$residuals)
#Removing Outliers
sequoia$residuals = residuals(best.AIC.model)
sequoia$std_residuals = rstandard(best.AIC.model)

threshold = 2
outliers = sequoia[abs(sequoia$std_residuals) > threshold, ]

new.data1 <- sequoia[abs(sequoia$std_residuals) <= threshold, ]

sequoia$residuals = residuals(best.BIC.model)
sequoia$std_residuals = rstandard(best.BIC.model)

threshold = 2
outliers = sequoia[abs(sequoia$std_residuals) > threshold, ]

new.data2 <- sequoia[abs(sequoia$std_residuals) <= threshold, ]

sequoia$residuals = residuals(model4)
sequoia$std_residuals = rstandard(model4)

threshold = 2
outliers = sequoia[abs(sequoia$std_residuals) > threshold, ]

new.data3 <- sequoia[abs(sequoia$std_residuals) <= threshold, ]
#Re-model using the new dataset
best.AIC.model$coefficients
best.BIC.model$coefficients
model4$coefficients

model.a = lm(height ~ diameter + squared.diameter + log.diameter, data = new.data1)
model.b = lm(height ~ diameter + cubic.diameter, data = new.data2)
model.c = lm(height ~ squared.diameter + cubic.diameter, data = new.data3)
#SW Test
#A
new.data1$ei = model.a$residuals
new.data1$yhat = model.a$fitted.values

ei = model.a$residuals
the.SWtest = shapiro.test(ei)
the.SWtest

#B
new.data2$ei = model.b$residuals
new.data2$yhat = model.b$fitted.values

ei = model.b$residuals
the.SWtest = shapiro.test(ei)
the.SWtest
```

```
#C
new.data3$ei = model.c$residuals
new.data3$yhat = model.c$fitted.values

ei = model.c$residuals
the.SWtest = shapiro.test(ei)
the.SWtest
#FK Test
#A
Group = rep("Lower",nrow(new.data1))
Group[new.data1$height < median(new.data1$height)] = "Upper"
Group = as.factor(Group)
new.data1$Group = Group
the.FKtest= fligner.test(new.data1$ei, new.data1$Group)
the.FKtest

#B
Group = rep("Lower",nrow(new.data2))
Group[new.data2$height < median(new.data2$height)] = "Upper"
Group = as.factor(Group)
new.data2$Group = Group
the.FKtest= fligner.test(new.data2$ei, new.data2$Group)
the.FKtest

#C
Group = rep("Lower",nrow(new.data3))
Group[new.data3$height < median(new.data3$height)] = "Upper"
Group = as.factor(Group)
new.data3$Group = Group
the.FKtest= fligner.test(new.data3$ei, new.data3$Group)
the.FKtest
#Quality test of new models
AIC(model.a)
AIC(model.b)
AIC(model.c)

BIC(model.a)
BIC(model.b)
BIC(model.c)

rmse(model.a)
rmse(model.b)
rmse(model.c)

summary(model.a)$adj.r.squared
summary(model.b)$adj.r.squared
summary(model.c)$adj.r.squared
model.a$coefficients
model.b$coefficients
model.c$coefficients
alpha = 0.05
the.CIs = confint(model.b,level = 1-alpha)
the.CIs
```

```r
test.stuff = summary(model.b)$coefficients
summary(model.b)$coefficients
model.b
```