Precept 5 Exercise

Bill Haarlow

Fall 2019

Objectives

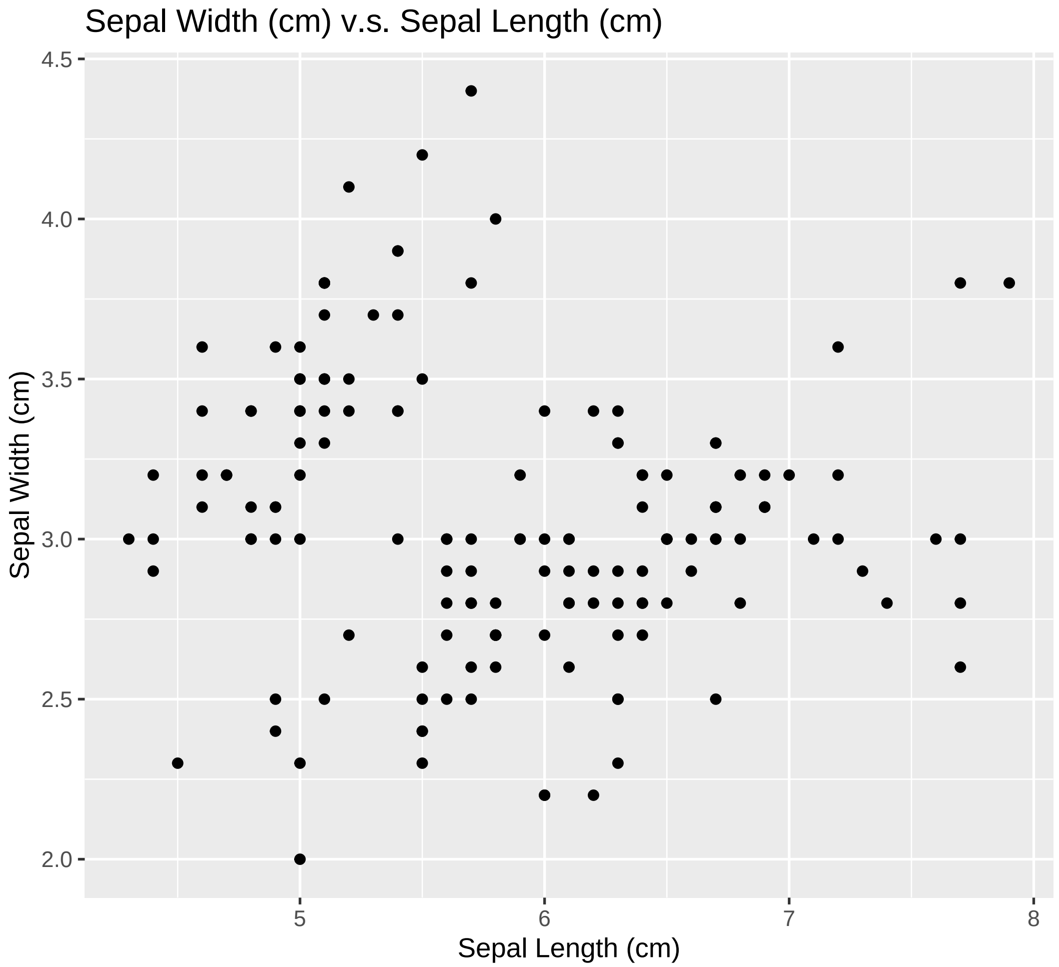
- Being able to fit linear regression models with R
- Being able to interpret the results of linear regressions
- Being able to verify some of the assumptions in linear regression

Demo

Example 1

a) Use the dataset `iris` and make a scatterplot of the two variables `Sepal.Length` and `Sepal.Width`. Does it look like a good idea to regress `Sepal.Width` on `Sepal.Length`?

```r
library(ggplot2)

# Generate scatterplot
ggplot(data=iris) +
  geom_point(mapping = aes(x=Sepal.Length, y=Sepal.Width)) +
  ggtitle("Sepal Width (cm) v.s. Sepal Length (cm)") +
  labs(x="Sepal Length (cm)", y="Sepal Width (cm)")
```



b) What is the correlation between `Sepal.Length` and `Sepal.Width`? What happens if you regress `Sepal.Width` on `Sepal.Length` anyway? Do you get better estimates than you would if you just used the mean of Sepal.Width as your predicted value? Compare the following two models:

$$y_i = \beta_0 + \epsilon_i$$
$$y_i = \beta_0 + \beta_l \cdot l_i + \epsilon_i$$

where l_i is the sepal length of a given plant. Add the regression line to your scatterplot. Think about the physical meaning of what your model is telling; does your model make sense?

```
# Get correlation
cor(x=iris$Sepal.Length, y=iris$Sepal.Width)
[1] -0.1175698

# Regress on a constant (i.e., include only the intercept in your model)
mod1 <- lm(Sepal.Width ~ 1, data=iris)
```

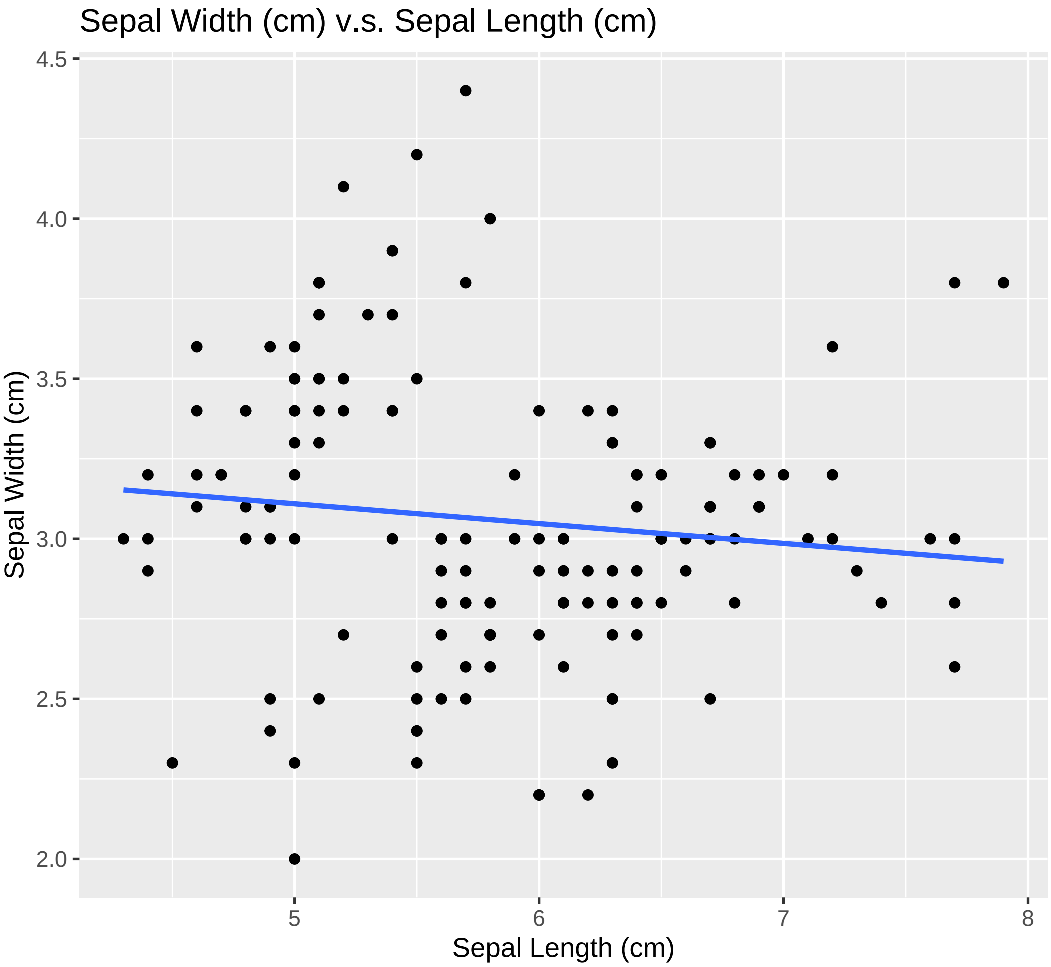
```
ls(mod1)  # Check what's in an lm object
 [1] "assign"        "call"          "coefficients"
 [4] "df.residual"   "effects"       "fitted.values"
 [7] "model"         "qr"            "rank"
[10] "residuals"     "terms"
class(mod1) # An object of class "lm" is a list; see `?lm`
[1] "lm"

############################################################

mod1$coefficients # This gives the estimate of the coefficient
(Intercept)
   3.057333

head(mod1$fitted.values) # `fitted.values` gives the estimated y-values
       1        2        3        4        5        6
3.057333 3.057333 3.057333 3.057333 3.057333 3.057333
length(mod1$fitted.values) # This should be of the same length as the y-vector
[1] 150

head(mod1$residuals)
          1           2           3           4           5
 0.44266667 -0.05733333  0.14266667  0.04266667  0.54266667
          6
 0.84266667
# `residuals` gives (estimated y-values - actual y-values)
length(mod1$residuals) # This should be of the same length as the y-vector
[1] 150

# What is the average of the residuals?
mean((mod1$residuals)^2)
[1] 0.1887129
# How is this compared with the variance of the Sepal.Width?
var(iris$Sepal.Width)
[1] 0.1899794


# Note that this is equivalent to using the
# mean of Sepal Width to estimate the sepal
# width value of a new observation.

mean(iris$Sepal.Width)
[1] 3.057333

############################################################

# We can get additional info of the model with `summary(mod1)`
```

```
summary(mod1)

Call:
lm(formula = Sepal.Width ~ 1, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.05733 -0.25733 -0.05733  0.24267  1.34267

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.05733    0.03559   85.91   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4359 on 149 degrees of freedom
ls(summary(mod1)) # see what info I can get from `summary(mod1)`
 [1] "adj.r.squared" "aliased"       "call"
 [4] "coefficients"  "cov.unscaled"  "df"
 [7] "r.squared"     "residuals"     "sigma"
[10] "terms"
head(summary(mod1)$residuals)
          1           2           3           4           5
 0.44266667 -0.05733333  0.14266667  0.04266667  0.54266667
          6
 0.84266667

summary(mod1)$r.squared # why is this zero?
[1] 0
summary(mod1)$adj.r.square
[1] 0
```

Let's regress Sepal.Width on Sepal.Length and see what happens.

```
# Simple linear regression
mod2 <- lm(Sepal.Width ~ Sepal.Length, data=iris)
summary(mod2)

Call:
lm(formula = Sepal.Width ~ Sepal.Length, data = iris)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1095 -0.2454 -0.0167  0.2763  1.3338

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.41895    0.25356   13.48   <2e-16 ***
```

```
Sepal.Length -0.06188    0.04297   -1.44    0.152
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4343 on 148 degrees of freedom
Multiple R-squared:  0.01382,   Adjusted R-squared:  0.007159
F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519

# Add the regression line to the scatterplot
# Preceptors: please explain what
# `geom_smooth(method = "lm", se=F)` mean
ggplot(data=iris, mapping = aes(x=Sepal.Length, y=Sepal.Width)) +
  geom_point() +
  geom_smooth(method = "lm", se=F) +
  labs(x="Sepal Length (cm)", y="Sepal Width (cm)",
       title = "Sepal Width (cm) v.s. Sepal Length (cm)")
```



c) Can you improve the model by adding in a second explanatory variable for each species using either of the following models:

$$y_i = \beta_l \cdot l_i + \beta_{ver} \cdot \mathbf{1}_{ver} + \beta_{vir} \cdot \mathbf{1}_{vir} + \beta_0 + \epsilon_i$$
$$y_i = \beta_l \cdot l_i + \beta_{set} \cdot \mathbf{1}_{set} + \beta_{ver} \cdot \mathbf{1}_{ver} + \beta_{vir} \cdot \mathbf{1}_{vir} + \epsilon_i$$

where l_i is the sepal length of a given plant and `set`, `ver`, and `vir` refer to the respective species. How should you interpret the output from each of these models? Are the results any better than the previous model? Use the R^2 and R^2_{adj} in your argument. (Preceptors: please explain the meanings of the coefficients and how the meanings are different for the two models.)

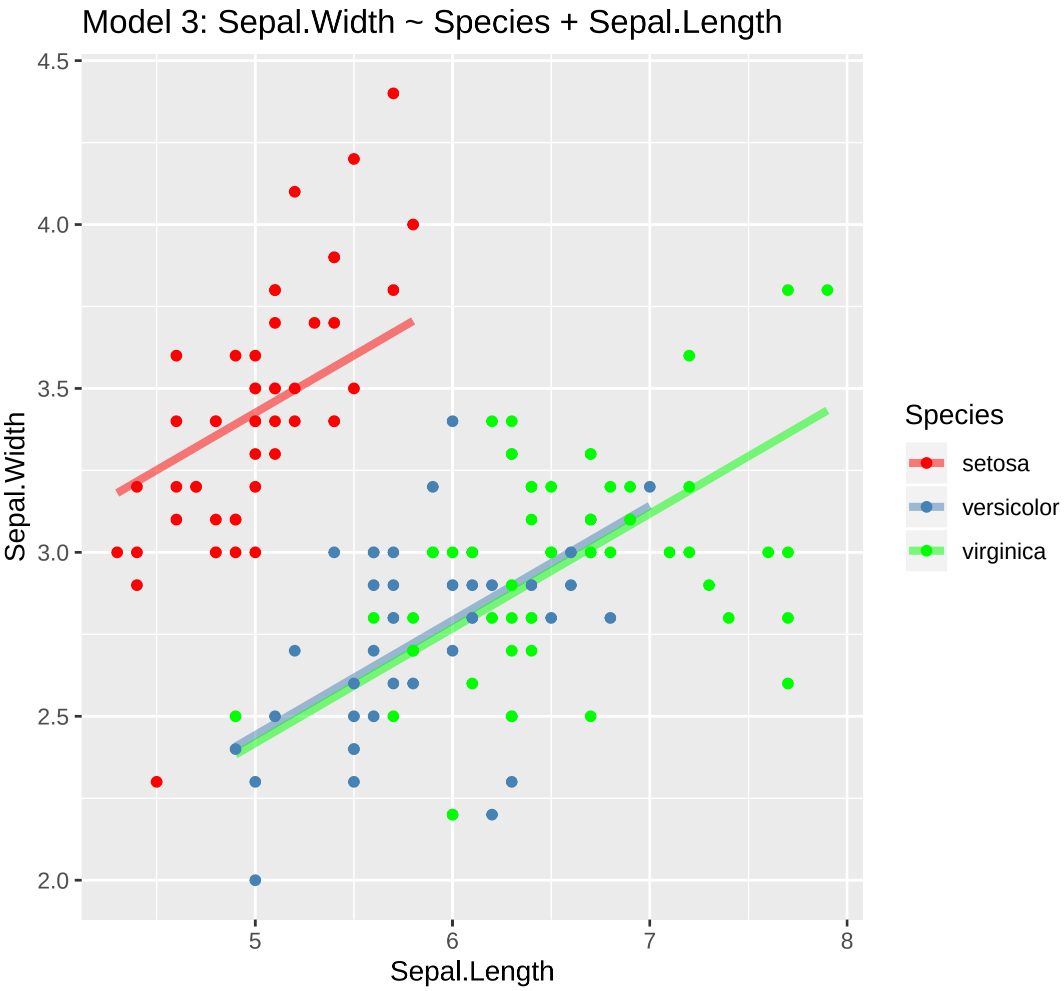
```r
# Multiple regression
mod3 <- lm(Sepal.Width ~ Sepal.Length + Species, data=iris)
summary(mod3)
```

```
Call:
lm(formula = Sepal.Width ~ Sepal.Length + Species, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.95096 -0.16522  0.00171  0.18416  0.72918 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1.67650    0.23536   7.123 4.46e-11 ***
Sepal.Length       0.34988    0.04630   7.557 4.19e-12 ***
Speciesversicolor -0.98339    0.07207 -13.644  < 2e-16 ***
Speciesvirginica  -1.00751    0.09331 -10.798  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.289 on 146 degrees of freedom
Multiple R-squared:  0.5693,    Adjusted R-squared:  0.5604 
F-statistic: 64.32 on 3 and 146 DF,  p-value: < 2.2e-16

mod4 <- lm(Sepal.Width ~ Sepal.Length + Species + 0, data=iris)
summary(mod4)

Call:
lm(formula = Sepal.Width ~ Sepal.Length + Species + 0, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.95096 -0.16522  0.00171  0.18416  0.72918 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
Sepal.Length        0.3499     0.0463   7.557 4.19e-12 ***
Speciessetosa       1.6765     0.2354   7.123 4.46e-11 ***
Speciesversicolor   0.6931     0.2779   2.494   0.0137 *  
Speciesvirginica    0.6690     0.3078   2.174   0.0313 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.289 on 146 degrees of freedom
Multiple R-squared:  0.9915,    Adjusted R-squared:  0.9912 
F-statistic:  4245 on 4 and 146 DF,  p-value: < 2.2e-16
```

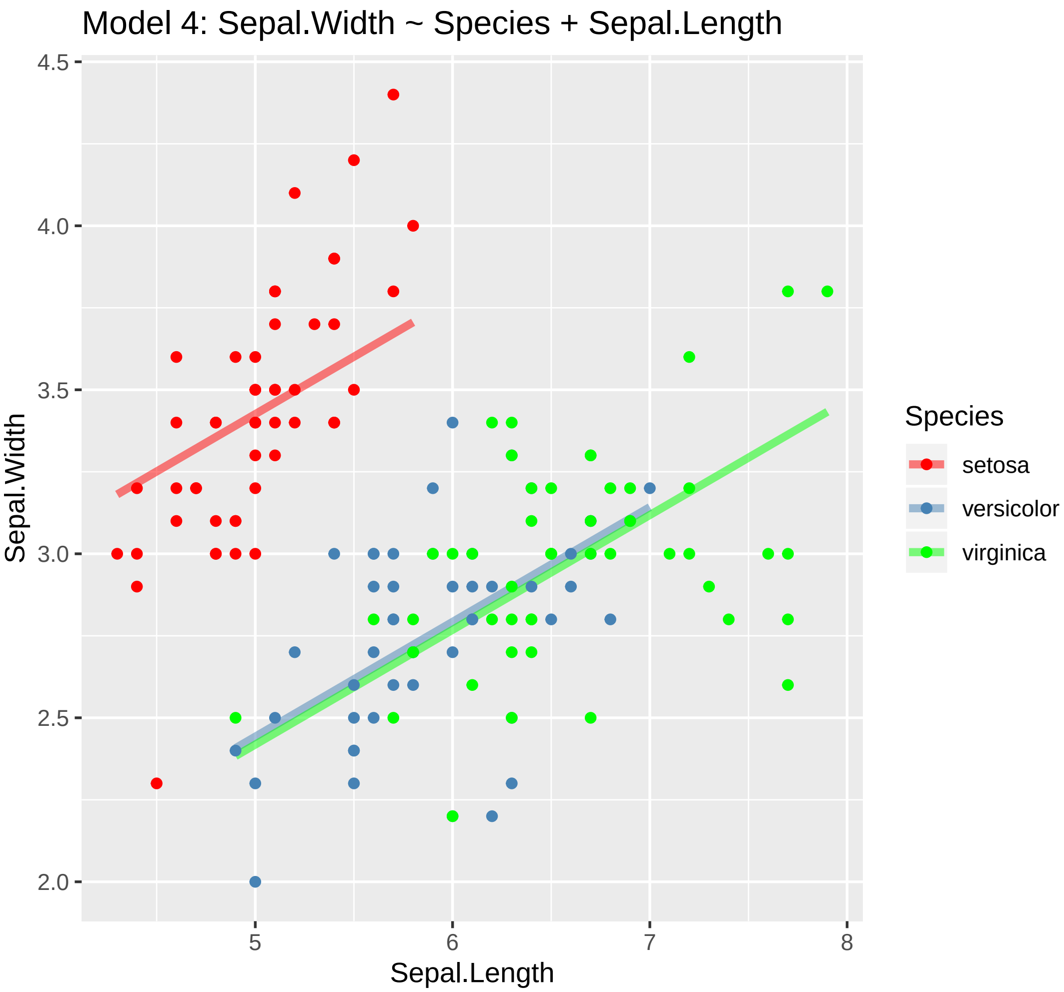
Graphical interpretation of what these models mean; note that both model 3 and 4 give the same set of lines.

```
# Plot data with the predicted models
# Prepare data frame for the graphs
```

```
fits <- data.frame(Sepal.Length=iris$Sepal.Length,
                   Sepal.Width=iris$Sepal.Width,
                   mod3=mod3$fitted.values,
                   mod4=mod4$fitted.values,
                   Species=iris$Species)

# Add the lines defined by the fitted mod3 to the scatterplot
ggplot(data = fits) +
  geom_line(aes(x=Sepal.Length, y=mod3, color=Species), size=1.5, alpha=0.5) +
  geom_point(aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
  scale_color_manual(values = c("red", "steelblue", "green")) +
  labs(x="Sepal.Length", y="Sepal.Width",
  title = "Model 3: Sepal.Width ~ Species + Sepal.Length")
```



```
# Add the lines defined by the fitted mod4 to the scatterplot
ggplot(data = fits) +
  geom_line(aes(x=Sepal.Length, y=mod4, color=Species), size=1.5, alpha=0.5) +
  geom_point(aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
  scale_color_manual(values = c("red", "steelblue", "green")) +
  labs(x="Sepal.Length", y="Sepal.Width",
  title = "Model 4: Sepal.Width ~ Species + Sepal.Length")
```



d) Now calculate the correlation within each species. Then, replace *sepal length* with the interaction term, *sepal length* × *species*, in model 4; find the regression line predicting the *sepal width* of a iris with given *sepal length* and *species* values. That is; find the fitted model:

$$\begin{aligned}\hat{y} = \hat{\beta}_{set} \cdot \mathbf{1}_{set} + \hat{\beta}_{ver} \cdot \mathbf{1}_{ver} + \hat{\beta}_{vir} \cdot \mathbf{1}_{vir} \\ + \hat{\beta}_{l\cdot set} \cdot l_i \cdot \mathbf{1}_{set} + \hat{\beta}_{l\cdot ver} \cdot l_i \cdot \mathbf{1}_{ver} + \hat{\beta}_{l\cdot vir} \cdot l_i \cdot \mathbf{1}_{vir}\end{aligned}$$

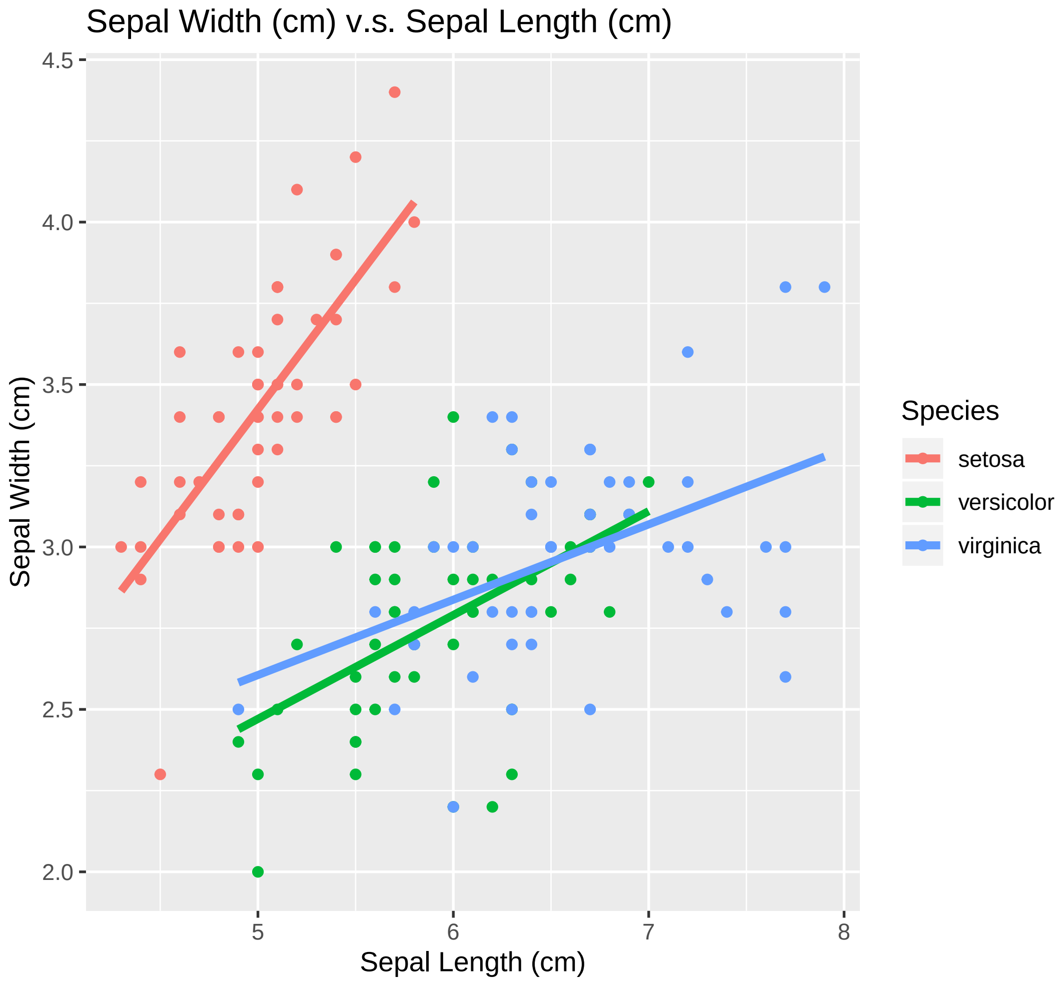
Preceptors and students: you do not need to worry about the details in the code chunk below. You just need to know that within-group-correlations between 'Sepal.Length' and 'Sepal.Width' are 0.74, 0.53, 0.46 for the species setosa, versicolor and virginica, respectively.

```
# Get correlations
corr.spec <- by(iris[,c('Sepal.Length', 'Sepal.Width')],
               INDICES= iris$Species,
               FUN=cor)
corr.spec
iris$Species: setosa
             Sepal.Length Sepal.Width
Sepal.Length    1.0000000   0.7425467
Sepal.Width     0.7425467   1.0000000
-----------------------------------------------
iris$Species: versicolor
             Sepal.Length Sepal.Width
Sepal.Length    1.0000000   0.5259107
Sepal.Width     0.5259107   1.0000000
-----------------------------------------------
iris$Species: virginica
             Sepal.Length Sepal.Width
Sepal.Length    1.0000000   0.4572278
Sepal.Width     0.4572278   1.0000000

# What by() does:
?by

# Careful how you use the output:
class(corr.spec)
[1] "by"
class(corr.spec[1])
[1] "list"
class(corr.spec[[1]])
[1] "matrix"
# Extacting out the within-species
# correlation between 'Sepal.Length' and 'Sepal.Width'
sapply(corr.spec, FUN=function(x){x[1,2]})
    setosa versicolor  virginica
 0.7425467  0.5259107  0.4572278
```

The correlation between `Sepal.Length` and `Sepal.Width` is much stronger within each group compare to that for the entire dataset. As a result it will be good to use different lines to make predictions for each group.

```
# Fit the model; "+ 0" means do not include beta_0
# (i.e., no overall y-intercept for the model); you can also use "-1"
# to achieve the same result
mod5 <- lm(Sepal.Width ~ Species + Sepal.Length:Species + 0, data=iris)
summary(mod5)
```

```
Call:
lm(formula = Sepal.Width ~ Species + Sepal.Length:Species + 0,
    data = iris)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.72394 -0.16327 -0.00289  0.16457  0.60954 

Coefficients:
                               Estimate Std. Error t value
Speciessetosa                  -0.56943    0.55386  -1.028
Speciesversicolor               0.87215    0.44906   1.942
Speciesvirginica                1.44631    0.40491   3.572
Speciessetosa:Sepal.Length      0.79853    0.11037   7.235
Speciesversicolor:Sepal.Length  0.31972    0.07537   4.242
Speciesvirginica:Sepal.Length   0.23189    0.06118   3.790
                               Pr(>|t|)    
Speciessetosa                  0.305622    
Speciesversicolor              0.054072 .  
Speciesvirginica               0.000482 ***
Speciessetosa:Sepal.Length     2.55e-11 ***
Speciesversicolor:Sepal.Length 3.95e-05 ***
Speciesvirginica:Sepal.Length  0.000221 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2723 on 144 degrees of freedom
Multiple R-squared:  0.9925,    Adjusted R-squared:  0.9922 
F-statistic:  3190 on 6 and 144 DF,  p-value: < 2.2e-16

# Plot data with the predicted model
ggplot(data=iris,
       mapping = aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
  geom_point() +
  geom_smooth(method = "lm", se=F, size = 1.5, alpha=.5) +
  ggtitle("Sepal Width (cm) v.s. Sepal Length (cm)") +
  labs(x="Sepal Length (cm)", y="Sepal Width (cm)")
```



e) What happens if you use the * operator in lieu of the : operator in the model?

```
# Regress model
mod6 <- lm(Sepal.Width ~ Sepal.Length*Species + 0, data=iris)
summary(mod6)

Call:
lm(formula = Sepal.Width ~ Sepal.Length * Species + 0, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.72394 -0.16327 -0.00289  0.16457  0.60954 

Coefficients:
                               Estimate Std. Error t value
Sepal.Length                     0.7985     0.1104   7.235
Speciessetosa                   -0.5694     0.5539  -1.028
```

```
Speciesversicolor                0.8721     0.4491   1.942
Speciesvirginica                 1.4463     0.4049   3.572
Sepal.Length:Speciesversicolor  -0.4788     0.1337  -3.582
Sepal.Length:Speciesvirginica   -0.5666     0.1262  -4.490
                               Pr(>|t|)    
Sepal.Length                   2.55e-11 ***
Speciessetosa                  0.305622    
Speciesversicolor              0.054072 .  
Speciesvirginica               0.000482 ***
Sepal.Length:Speciesversicolor 0.000465 ***
Sepal.Length:Speciesvirginica  1.45e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2723 on 144 degrees of freedom
Multiple R-squared:  0.9925,    Adjusted R-squared:  0.9922 
F-statistic:  3190 on 6 and 144 DF,  p-value: < 2.2e-16
```

Exercises

Question 1

Using the dataset `possum` in the package `openintro`, we will generate a linear model to predict possum lengths.

```
library(openintro)
Please visit openintro.org for free statistics
materials

Attaching package: 'openintro'
The following object is masked from 'package:ggplot2':

    diamonds
The following objects are masked from 'package:datasets':

    cars, trees
```

a) Use the ggpairs() function to investigate the pairwise relationship between the variables. We want to use one of the variables (other than `totalL` of course) in the dataset to predict the total length of a possum. Which variable will you use? Explain why.

b) Regress `totalL` on the variable that you selected in part (a). Report the R^2 value. Also, plot the scatterplot for `totalL` v.s. the selected variable with the regression line superimposed.

You can use the following code for making the labels

```
labs(x="Head Length", y="Total Length",
title = "Total Length (cm) vs. Head Length (cm)")
```

c) Add the variable `sex` into the model and refit the model by fill in the contents for `lm()` below.

```
# Fit model;
mod12<- lm()
```

The code in the following code chunk will plot the scatterplot and use different colors for different genders of the possums. Add the predicted regression lines that you made with `mod12` to your graph.

```
# Fit model
summary(mod12)


fit12 = data.frame(possum, mod12 = mod12$fitted.values)

# Plot data
ggplot(data=fit12) +
  geom_line(aes(x=headL, y=mod12, color=sex), size= 1.5) +
  geom_point(aes(x=headL, y=totalL, color=sex)) +
  labs(x="Head Length", y="Total Length",
  title = "Total Length (cm) vs. Head Length (cm)")
```

Now modify your model by adding the interaction term between `sex` and `headL` to the last model.

```
# Model;
mod13  = lm()
summary(mod13)
```

The code in the following code chunk will plot the scatterplot with the new predicted regression lines add to it.

```
fit13 = data.frame(possum, mod13 = mod13$fitted.values)

# Plot
ggplot(data=fit13) +
  geom_line(aes(x=headL, y=mod13, color=sex), size= 1.5) +
  geom_point(aes(x=headL, y=totalL, color=sex)) +
  labs(x="headL", y="totalL",
  title = "totalL ~ headL+sex")
```

d) With the simpler model (i.e., `mod12`) in part (c), predict the total length of a possum if the possum is female and has head length 92.4 mm; what about a male possum with head length 86.4 mm.

e) Assume that your model already has an intercept term, if a factor variable with 3 categories (i.e., with 3 levels) is added to the syntax in the R code, how would this affect the mathematical form of your model? E.g., if the original model is

```
mod.origin <- lm(y ~ 1, data=some.data.frame)
```

and the updated model is

```
mod.with.factor <- lm(y ~ f.var, data=some.data.frame)
```

where `f.var` is a factor vector with 3 levels, what will the mathematical forms of the original and the updated models be? Write out these models in their mathematical forms (instead of typing them out, you can write them out on a piece of paper to save time).

How would adding the factor variable affect the prediction line? Hint: Look at the result of `mod12` in part c.

f) How would adding the interaction term between the factor in part f and a continuous variable affect the prediction line (just explain this in words; you do not need to provide the mathematical model)? E.g., now the model would be

```
mod.with.factor <- lm(y ~ f.var:cont_var, data=some.data.frame)
```

Hint: Look at the result of `mod13` in part c.