# CS_08

steppe

2/28/2022

This is a case study !

Many solutions are available!

'You can go your own way, go your own way!'

We made you panic and sweat!

```
?iris
library(kableExtra)
library(tidyverse)
```

```
data(iris)
knitr::kable(head(iris), booktab = T)
```

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

We will now create a table of the summary statistics of this species here.

```
a <- iris %>%  # use map and modify the outputs
  split(.$Species) %>%
  map(summary) %>%
  map(as.data.frame) %>%
  map(drop_na) %>%
  bind_rows() %>%
  separate(Freq, c('Statistic', 'Value'), ":") %>%
  filter(!str_detect(Var2, 'Species')) %>%
  mutate(Species = c(rep('setosa', times = 24),
                     rep('versicolor', times = 24),
                     rep('virginica', times = 24))) %>%
  select(-Var1) %>%
  rename(Variable = Var2) %>%
  filter(!str_detect(Statistic, 'Qu'))

iris_table <- iris %>% # use summarize
  group_by(Species) %>%
  summarise(across(where(is.numeric),
                   list(
                     'min.'= min,
```

Table 1: Summary Statistics of Iris Dataset

| | | Sepal | | Petal | |
|---|---|---|---|---|---|
| Species | Statistic | Length | Width | Length | Width |
| I. setosa | min. | 4.3 | 2.3 | 1.0 | 0.1 |
| | mean | 5.0 | 3.4 | 1.5 | 0.2 |
| | median | 5.0 | 3.4 | 1.5 | 0.2 |
| | max | 5.8 | 4.4 | 1.9 | 0.6 |
| I. versicolor | min. | 4.9 | 2.0 | 3.0 | 1.0 |
| | mean | 5.9 | 2.8 | 4.3 | 1.3 |
| | median | 5.9 | 2.8 | 4.3 | 1.3 |
| | max | 7.0 | 3.4 | 5.1 | 1.8 |
| I. virginica | min. | 4.9 | 2.2 | 4.5 | 1.4 |
| | mean | 6.6 | 3.0 | 5.6 | 2.0 |
| | median | 6.5 | 3.0 | 5.5 | 2.0 |
| | max | 7.9 | 3.8 | 6.9 | 2.5 |

```
                    max = max,
                    mean = mean,
                    median = median))
          ) %>%
  pivot_longer(-Species, values_to = 'Value') %>%
  separate(name, c('Tissue', 'Statistic'), sep = "_") %>%

  # here we work on styling the values for the table
  mutate(Tissue = str_replace(Tissue, '[.]', ' ')) %>%
  mutate(Species =  paste0('I. ', Species)) %>%
  mutate(Value = round(Value, digits = 1)) %>%

  pivot_wider(names_from = Tissue, values_from = Value) %>%
  mutate(Statistic=fct_relevel(Statistic, "min.", "mean", "median", 'max')) %>%
  arrange(Species, Statistic)

iris_table[c(2:4, 6:8, 10:12),1] <- ""
knitr::kable(iris_table,
             align = "llcccc",
             col.names = c('Species','Statistic','Length','Width','Length','Width'),
             caption = "Summary Statistics of Iris Dataset",
             booktabs = T) %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  add_header_above(., c(" " = 2, "Sepal" = 2, "Petal" = 2))
```

```
rm(a, iris_table)
```

```
model_summary <- summary(lm(Sepal.Length ~ Sepal.Width, data = iris))
model_summary_statistics <- tibble("intercept" = model_summary$coefficients[1,1],
                                   "slope"     = model_summary$coefficients[2,1],
                                   "p_value"   = model_summary$coefficients[2,4],
                                   "r_squared"  = model_summary$r.squared)

model_summary_statistics <- model_summary_statistics %>%
```

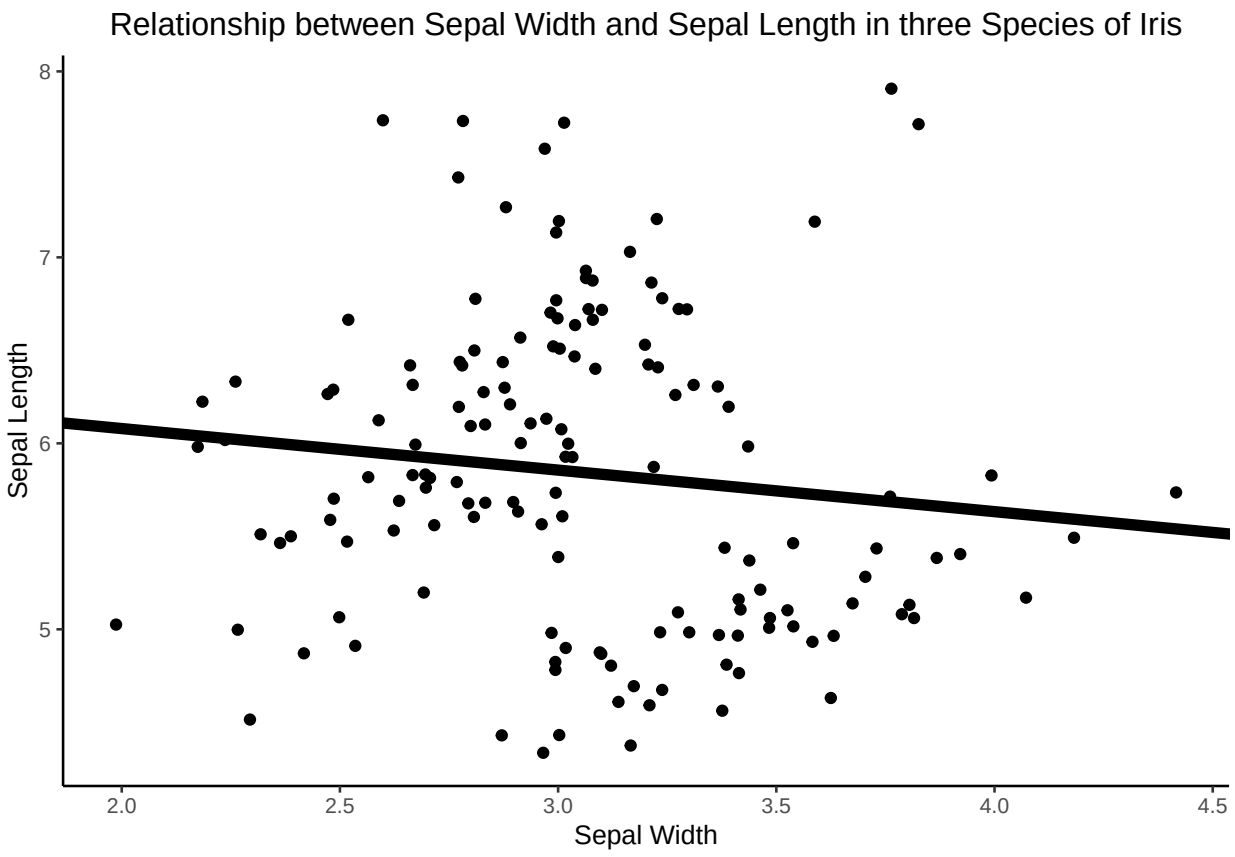
Table 2: Results of a linear model with one continuous predictor and response

| Intercept | Slope | p-Value | R$^2$ |
|---|---|---|---|
| 6.526 | -0.223 | 0.152 | 0.014 |

```
  mutate(across(where(is.numeric), ~ round(.x, digits = 3)))

knitr::kable(model_summary_statistics,
             col.names = c('Intercept', 'Slope','p-Value','R$^2$'),
             align = "cccc",
             caption = "Results of a linear model with one continuous predictor and response",
             booktabs = T
             ) %>%
  kable_classic(full_width = F, html_font = "Cambria")
```

We can very simply plot our linear model using the intercept and slope.

```
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length))+
  geom_jitter()+
  geom_abline(intercept = model_summary$coefficients[1,1],
              slope = model_summary$coefficients[2,1], size = 2) +
  theme_classic(base_size = 10) +
  labs(title = 'Relationship between Sepal Width and Sepal Length in three Species of Iris',
       y = 'Sepal Length', x = 'Sepal Width') +
  theme(plot.title = element_text(hjust = 0.5))
```



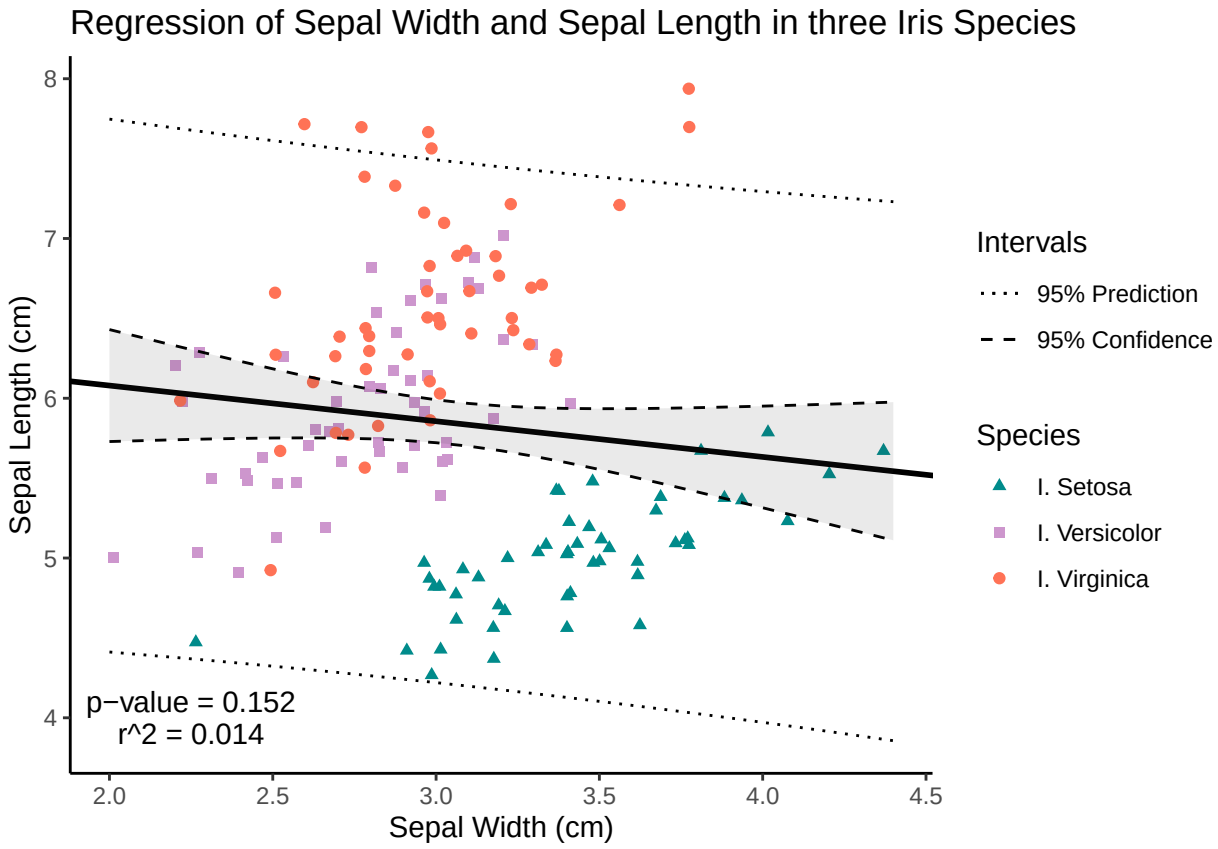
We can make a big overly fancy plot of these relationships like so below. The one wrench in the gear is that I cannot figure out how to reorder the fancy dual legend with the line legend below using guides! Perhaps you can?

```
model <- lm(Sepal.Length ~ Sepal.Width, data = iris)
conf_int <- as.data.frame(predict(model, interval = "confidence", level = 0.95))
conf_int <- cbind(conf_int, iris)

pred_int <- as.data.frame(predict(model, interval = "prediction", level = 0.95))
```

```
## Warning in predict.lm(model, interval = "prediction", level = 0.95): predictions on current data refe
```

```
pred_int <- cbind(pred_int, iris)

ggplot(conf_int, aes(x = Sepal.Width, y = Sepal.Length))+

  # I use jitter to show I used these data as continuous !
  geom_jitter(
    aes(
      shape=Species, color = Species)
    ) +

  # we can add our model here
  geom_abline(intercept = model_summary$coefficients[1,1],
              slope = model_summary$coefficients[2,1],
              size = 1) +

  # Confidence intervals are here
  geom_line(aes(y = lwr), color = "black", linetype = "dashed") +
  geom_line(aes(y = upr), color = "black", linetype = "dashed") +

    # this one is fake! only used to add the legend !
  geom_line(aes(y = lwr, linetype = "dashed"), color = "black") +
  # fill in the polyong here
  geom_ribbon(aes(ymin=lwr, ymax=upr), alpha=0.1) +

  # Prediction intervals are here
  geom_line(data = pred_int, aes(y = lwr), color = "black", linetype = "dotted") +
  geom_line(data = pred_int, aes(y = upr), color = "black", linetype = "dotted") +
    # this one is fake! only to add the legend !
  geom_line(data = pred_int, aes(y = lwr, linetype = "dotted"), color = "black") +


  # define and add a line legend here
  scale_linetype_manual(values = c("95% Prediction"  = "dotted",
                                   "95% Confidence"  = "dashed"),
                        name = "Intervals") +

  # modify the shapes and colours of these data; if you are interested.
  scale_colour_manual(name = "Species",
                      labels = c("I. Setosa", "I. Versicolor", "I. Virginica"),
                      values = c("cyan4", "plum3", "coral1")) +
  scale_shape_manual(name = "Species",
                      labels = c("I. Setosa", "I. Versicolor", "I. Virginica"),
                      values = c(17, 15, 19)) +
```

```
  # Some style stuff is here
labs(title = 'Regression of Sepal Width and Sepal Length in three Iris Species',
     y = 'Sepal Length (cm)',
     x = 'Sepal Width (cm)') +
theme(plot.title = element_text(hjust = 0.5)) +

# we will add the p-value and r^2 to the plot - it i  s a little busy for my liking !
annotate(geom ="text", x=2.25, y=4.1,
         label= paste0("p-value = ", model_summary_statistics[,3])) +
annotate(geom = "text", x=2.25, y=3.9,
        label = paste0("r^2 = ", model_summary_statistics[,4])) +

# manually select our shapes
theme_classic()
```




```
rm(model, conf_int, pred_int)
```

```
p1 <- iris %>%
  mutate(Sepal.Length.Predict =
           ((model_summary$coefficients[2,1]) * Sepal.Width) + model_summary$coefficients[1,1],
         Sepal.Length.Predict.upr = Sepal.Length.Predict + 1.96*model_summary$sigma,
         Sepal.Length.Predict.lwr = Sepal.Length.Predict - 1.96*model_summary$sigma)

ggplot(data = iris, aes(x= Sepal.Width, y= Sepal.Length)) +
  geom_point() +
  geom_abline(slope = model_summary$coefficients[2,1],
```

```
              intercept = model_summary$coefficients[1,1]) +
  geom_ribbon(data = p1,
              aes(ymin=Sepal.Length.Predict.lwr,
                  ymax=Sepal.Length.Predict.upr),
              alpha=0.2) +
  labs(title = 'Relationship between Sepal Width and Sepal Length in three Iris Species',
       y = 'Sepal Length', x = 'Sepal Width') +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()

rm(p1)
```

```
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length))+
  geom_jitter()+
  geom_smooth(method = "lm", colour = "black", formula = y ~ x) +
  labs(title = 'Relationship between Sepal Width and Sepal Length in three Iris Species',
       y = 'Sepal Length', x = 'Sepal Width') +
  theme_classic(base_size = 10) +
  theme(plot.title = element_text(hjust = 0.5))
```

```
rm(iris, model_summary)
```

Citations:

"Anderson, Edgar (1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2–5."