Lab 11

Stat C131A

In this lab, we will learn about PCA, `ggbiplot`, `coplot` and multiple regression.

A simple PCA example

This is a simple example for PCA from R Bloggers. The iris dataset is perhaps the best known dataset for classification. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

```
# Load data
data(iris)
head(iris, 3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

There are five variables in the dataset `iris`, where `Sepal.Length`, `Sepal.Width`, `Petal.Length` and `Petal.Width` are continuous and `Species` is categorical.

```
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

Excercise 1: PCA on the iris dataset

(a) Apply PCA to the four continuous variables. Be sure to transform these variables by taking the log and then converting those log values to standard units. **Note.** The centering and scaling may be done by the user or by changing the arguments of `prcomp()`. See the manual page `?prcomp()` for details.

```
# log transform on the four continuous variable
# log.ir <-
# the Species variable
# ir.species <-

# Apply PCA
# center and scale is used to standardize the variables prior to the application of PCA.
# iris.pca <-
```

(b) Create a scatter plot of the transformed data projected onto PC1 and PC2 with PC2 on the vertical axis and PC1 on the horizontal axis. Color the points on the scatter plot by species. Use `ggbiplot` to plot the weights of the linear combinations (loadings) when calculating principal components.

```
# Uncomment the following two lines to install "ggbiplot"
# library(devtools)
# install_github("vqv/ggbiplot")
```

```
library(ggbiplot)
# insert your code here
```

(c) Interpret the lines in the biplot: what to the lengths and directions tell you?

- The red arrows are vectors where the horizontal direction is given by the loading in PC1, and the vertical direction is given by the loading in PC2
- The loadings are the covariances between the original variable and the PC
- They may also be thought of as the coefficients on the centered and scaled components to be used in the prediction of the original variables. -If arrows point close together, then they contribute similarly to the PCs, and can be summarized in a lower dimension

Now, lets add a few missing values to the data.

```
iris2 <- iris
iris2[2,3] <- NA
iris2[5,2] <- NA
iris2[10,1] <- NA

# What species are missing?
# insert your code here
```

(d) Apply PCA as before, but make change to code to omit rows with missing values.

```
# Do PCA, but on iris2, the data with missing values
```

(e) Project the full data (no missing values) onto the principal components space computed from the data with missingness. Create a scatter plot of the full data projected onto the space formed by PC1 and PC2 (as calculated from the data with missingness). Can you identify the species corresponding to the rows with missing data?

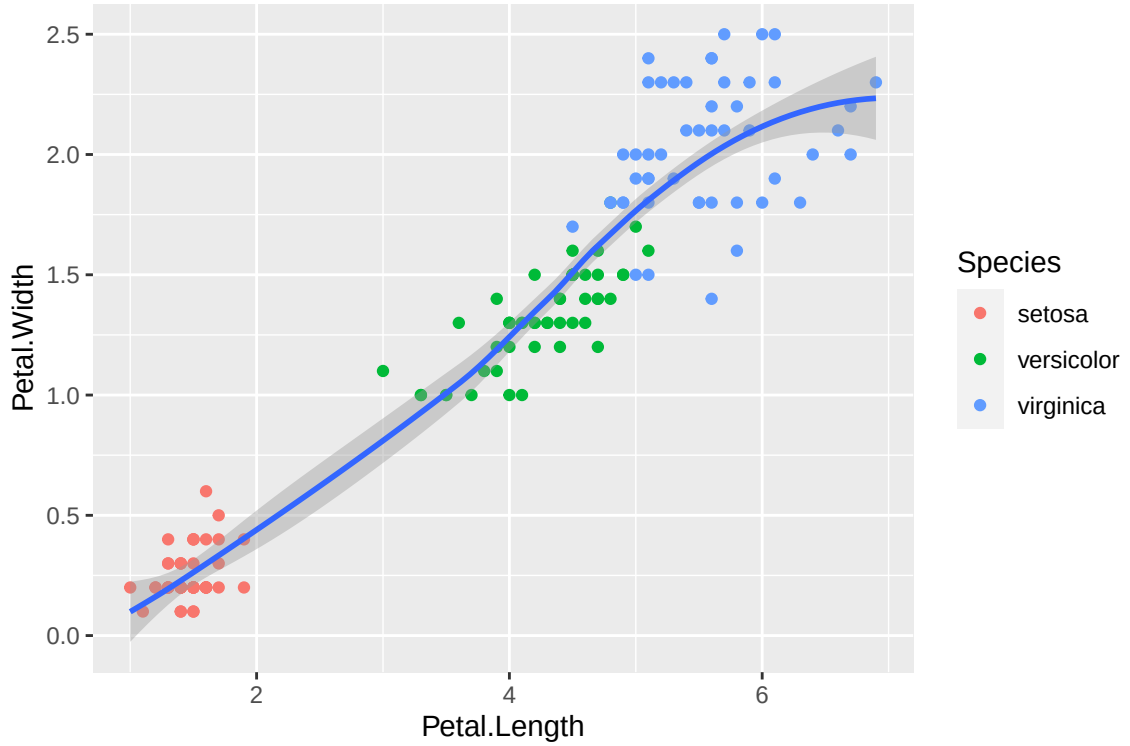
```
# Uncomment certain lines after finishing part(a)&(c)
# insert your code here

# Rotate data by doing the matrix multiplication
# dimensions: (150 x 4) x (4 x 4)
# iris.rot <- scale(log(as.matrix(iris[,1:4]))) %*% iris2.pca$rotation

# Copy vector of species labels where we can have Missing as a label
# ir.species_with_missing <- factor(ir.species, levels = c(levels(ir.species), "Missing"))

# Add missing label
# ir.species_with_missing[!complete.cases(iris2)] <- "Missing"

# Plot
```

Exercise 2: Conditioning plots on iris

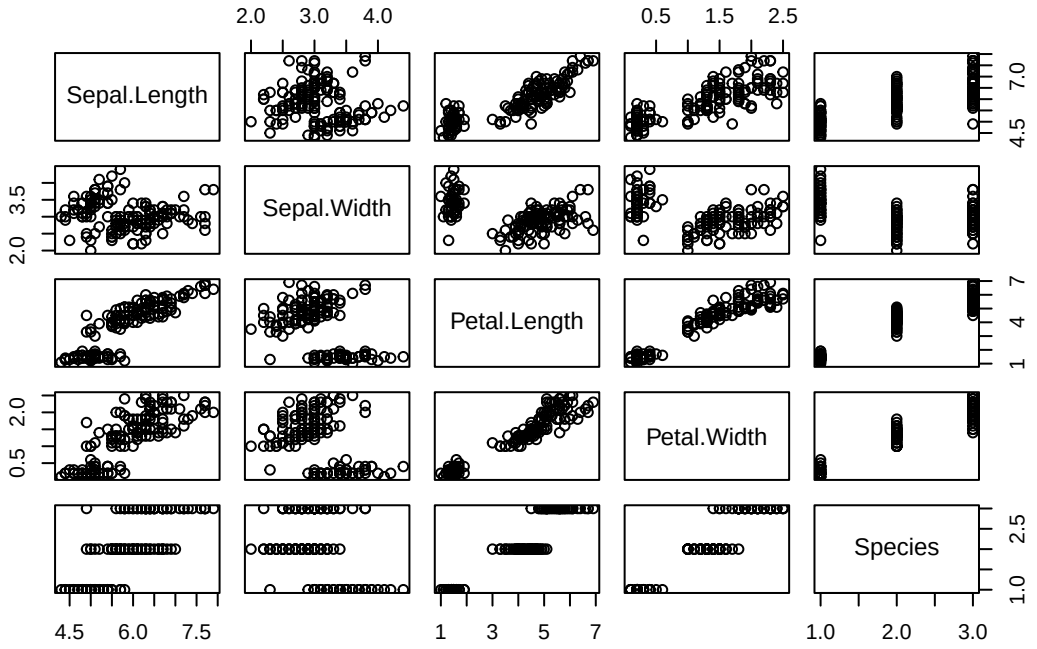
Explore the relationship between `Petal.Length` and `Petal.Width`.

```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width))+
  geom_point(aes(col = Species)) +
  geom_smooth(method = "loess")
```

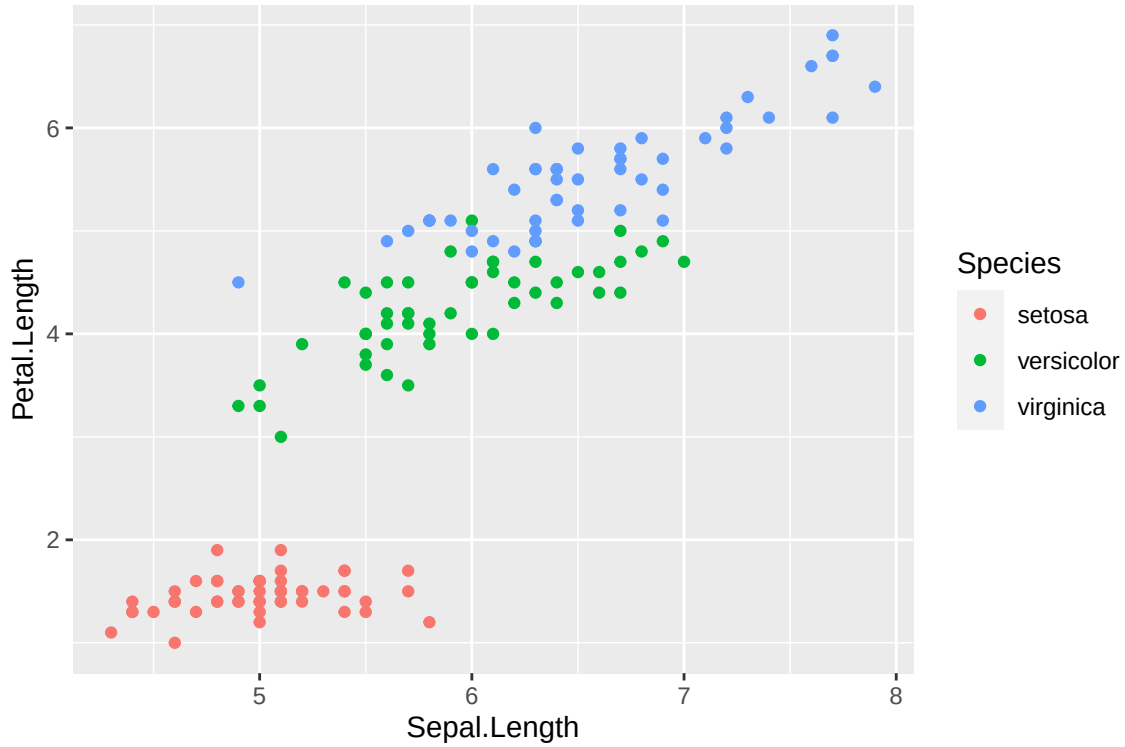
```
## `geom_smooth()` using formula 'y ~ x'
```




```
pairs(iris)
```




```
ggplot(iris) +
  geom_point(aes(x = Sepal.Length, y = Petal.Length, col = Species))
```



(a) Create a conditioning plot to visualize how `Petal.Length` and `Petal.Width` vary with `Sepal.Width`.

```
# insert your code here
```

Multiple regression with diamond price data

This is a very large data set showing various factors of over 50,000 diamonds including price, cut, color, clarity, etc. We are interested in diamond price `price` and how different factors influence it.

Variable	Description
price	price in US dollars ($326–$18,823)
carat	weight of the diamond (0.2–5.01)
cut	quality of the cut (Fair, Good, Very Good, Premium, Ideal)
color	diamond colour, from J (worst) to D (best)
clarity	how clear the diamond is (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
length.in.mm	length in mm (0–10.74)
width.of.mm	width in mm (0–58.9)
depth.in.mm	depth in mm (0–31.8)
depth	total depth percentage $ = z / mean(x, y) = 2 * z / (x+ y) (43–79)$
table	width of top of diamond relative to widest point (43–95)

```
diamonds <- read.csv("diamonds.csv")
head(diamonds)
```

```
##   carat       cut color clarity depth table price length.in.mm width.of.mm
## 1  0.23     Ideal     E     SI2  61.5    55   326         3.95        3.98
## 2  0.21   Premium     E     SI1  59.8    61   326         3.89        3.84
## 3  0.23      Good     E     VS1  56.9    65   327         4.05        4.07
## 4  0.29   Premium     I     VS2  62.4    58   334         4.20        4.23
## 5  0.31      Good     J     SI2  63.3    58   335         4.34        4.35
## 6  0.24 Very Good     J    VVS2  62.8    57   336         3.94        3.96
##   depth.in.mm
## 1        2.43
## 2        2.31
## 3        2.31
## 4        2.63
## 5        2.75
## 6        2.48
```

Exploratory Data Analysis before Regression

First, to better understand the relationships between the variables, we will generate scatter plots. Create scatter plots between the response variable (`price`) and all the continuous variables. The function `is.numeric()` might be helpful to check whether a variable is numeric. For example:

```
vec1 = 1:10
vec2 = as.character(1:10)
vec1
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
vec2
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```
# The following line of code would return TRUE
is.numeric(vec1)
```

```
## [1] TRUE
```

```
# The following line of code would return FALSE
is.numeric(vec2)
```

```
## [1] FALSE
```

```
# The following line of code would return TRUE
is.character(vec2)
```

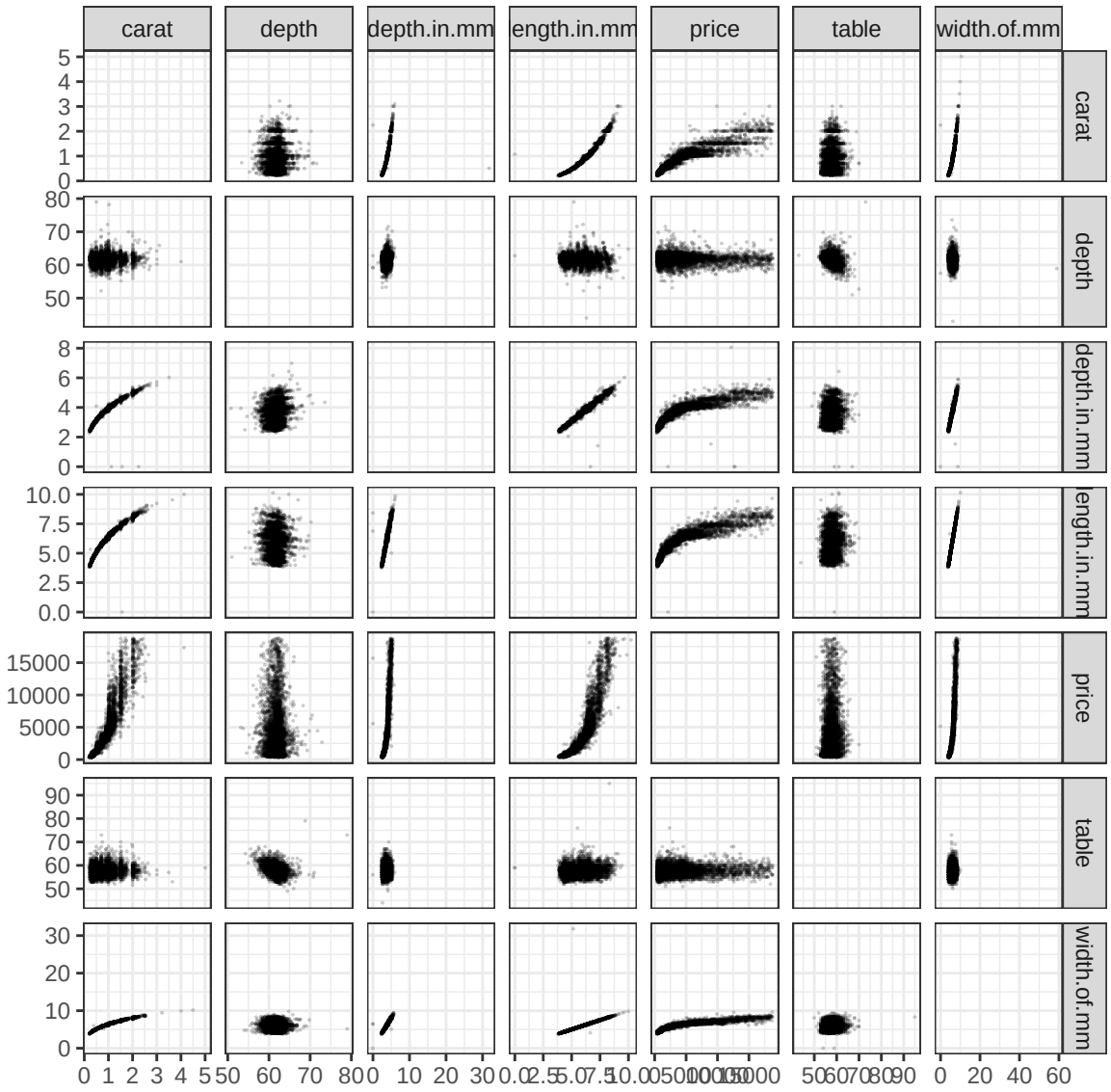
```
## [1] TRUE
```

The function `which()` can help you to locate the column indices of the numeric vectors. (In fact, function `which()` is a super useful function in R.)

```
# function `which` give the TRUE indices of a logical object
which(c(TRUE, FALSE, TRUE, FALSE, TRUE))
```

```
## [1] 1 3 5
```

```
# Insert your code here, use for loop to loop through each numerical variable
diamonds %>% select(where(is.numeric)) -> diamonds.select
apply(cbind(combn(1:7, 2),
                        combn(7:1, 2)), 2,  function(x) {
        pair <- diamonds.select[,x]
        pair$x.name <- names(pair)[1]
        pair$y.name <- names(pair)[2]
        names(pair)[1:2] <- c("x", "y")
        return(pair[sample(1:nrow(pair), 5000),])
    }
) %>% do.call(rbind, .) -> diamonds.pairs


diamonds.pairs %>% ggplot(
    aes(x = x, y = y)
) + geom_point(size = 0.01, shape = 1, alpha = 0.2) +
    facet_grid(y.name ~ x.name, scales = "free") +
    theme_bw() + theme(axis.title = element_blank())
```



Multiple regression with continuous variable

Exercise 3 Fit the model and Calculate the Statistics

(a) Fit a linear model to price with all the continuous variables as explanatory variables. Print the summary of your model.

```
# Insert you code here, save your model as `fit`
```

(b) Calculate the fitted values.

```
# Insert you code here, save your results as `fitted.value`
```

(c) Calculate the residuals, the residual sum of squares (RSS), and the total sum of squares (TSS) using the `fitted.value()` from the above chunk.

```
# Insert you code here
# RSS <-
# TSS <-
```

(d) Calculate the R-square (R^2) using RSS and TSS. What is the interpretation of R^2?

```
# Insert you code here, save your results as `Rsq`
# Rsq <-
```

Exercise 4 Think deeper. Is the model reasonable?

(a) Using the fitted model, we can write the estimated model formula. How do we interpret this equation? *Hint.* Use `summary()` on the model object.

```
# Insert your answer
```

By looking at the p-values, we know that the coefficients we estimated are significant except for that of `depth.in.mm`. Take `length.in.mm` for example, the coefficient -1315.668 tells us that for a unit increase in `length.in.mm` is associated with a reduction in price of $1315.67 dollars on average.

Isn't that weird? Fit another multivariate model, but this time, drop `length.in.mm`, `width.of.mm` and `depth.in.mm`? Is that a good idea? *Hint.* Check the correlation between `length.in.mm`, `width.of.mm`, `depth.in.mm` and `carat`.

```
# insert your code here
```

(b) Plot the residuals from the restricted model from part (a) against the variables and calculate their correlations. Can you find any problem in your model by looking at these scatter plots? If you're asked to add some terms to improve the model, what will you do? (*Hint.* Consider the scatter plot in Exercise 1: are the relationships linear?)

```
# insert your code here
```