Classifying Plant Leaves

Ethan Scott

02/03/2023

Abstract

The goal of this report is to classify leaves into one of two categories based on their length and width. 32 leaves were collected for this report, 16 pear leaves and 16 cherry leaves. These leaves had their length and width recorded to the nearest millimeter. Using the 32 leaves a model was created to classify new leaves, this model will classify any new measurements of length and width as either a cherry leaf or a pear leaf.

Introduction

Leaf classification is an important task in the field of plant biology, as it helps in identifying different plant species. Traditional methods of leaf classification involve manual identification of leaves by experts. However, this approach can be time-consuming and may not be practical for large-scale studies. In recent years, machine learning algorithms have been developed for leaf classification, which can automate the process of identifying plant species. In this study, we used LDA and QDA models to classify leaves of two different plant species, pear and cherry, based on their physical characteristics of length and width.

The remainder of this report will show how this was accomplished. Firstly, it will cover the materials and methods used to collect and clean the data. Next, the models created and the results of the models will be discussed. Finally, the results of the models, the classification boundary of the models and the shortcomings of project will be discussed.

Materials and Methods

The materials needed are:

- 32 total leaves, 16 cherry leaves and 16 pear leaves.
- A ruler capable of measuring to the nearest millimeter.

Once these materials have been acquired we can begin to collect our data by:

1. Measure; to the nearest millimeter, the width of the leaf near middle and record this value.
2. Measure the length of the leaf and record the value to the nearest millimeter.
3. Label each leaf as Leaf_type=0 or Leaf_type=1, where 0 are cherry leaves and 1 are pear leaves.

For each leaf in this study, the length and width were measured to the nearest millimeter and recorded into a single data set with the leaf types being recorded as 0 if it is a cherry leaf and 1 if it is a pear leaf. This data was organized with the leaf type in the first column, the width in the second column and the length in the third column.

Once all the data was collected and organized, an linear discriminant analysis(LDA) model was created. To do this, the bi-variate normal distribution of the cherry leaves and the pear leaves using the 32 leaves we collected. To create these distributions, the method of moments was used to find the mean and covariance matrix for both of the leaf types. When creating our LDA model we assume the covariance matrices are equal so we used a pooled estimate of the covariance matrix for both distributions. Our LDA model has the classification rule where if the bi-variate normal of a cherry leaf is greater than the bi-variate normal of the pear leaf, given the length and width of the leaf, then our model will classify it as a cherry leaf, and if the bi-variate normal of the cherry leaf is smaller than that of the pear leaf, then our model will classify it as a pear leaf. Should the two bi-variate normal's be equal given the length and width of the leaf then it will be unclassified in out LDA model. Then with this model, we took the measurements of 3 new leaves to see how the model would classify them. A geometric view of our classification was also made on a graph to visualize the classification rule.

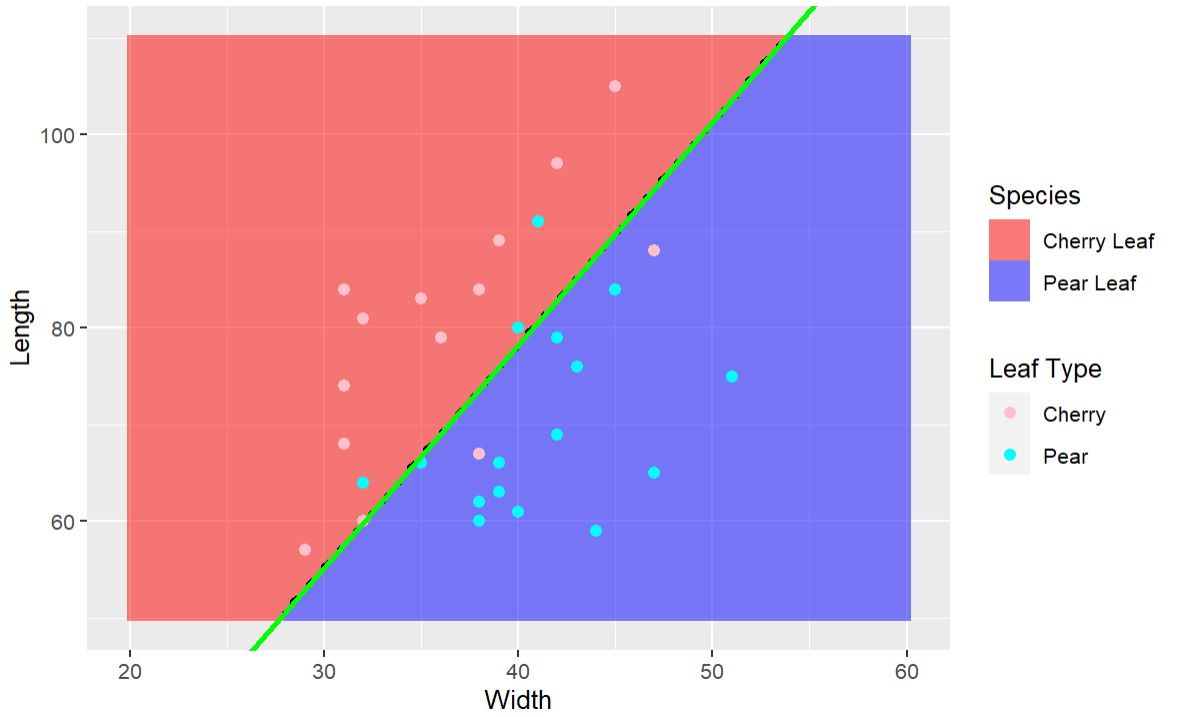
Once the LDA has been created we will also create a model where we do not assume the covariance matrices of both of our distributions to be equal, so we do not use a pooled covariance matrix to create our normal distributions. Instead we use the covariance matrix of each of the leaf types separately. This new model is a quadratic discriminant analysis(QDA) model. This new model uses the same classification rule as our LDA model. Once created we use the same 3 leaves that we used to test our LDA model, and test the results of our QDA model. We also graph a geometric view of this model to visualize the classification rule.

Results

Once our LDA model is created, we try to classify 3 new leaves to see how our model will classify them, below are the results of this classification.

Leaf #	Width(mm)	Length(mm)	LDA Classification
1	32	82	Cherry Leaf
2	38	52	Pear Leaf
3	40	76	Pear Leaf

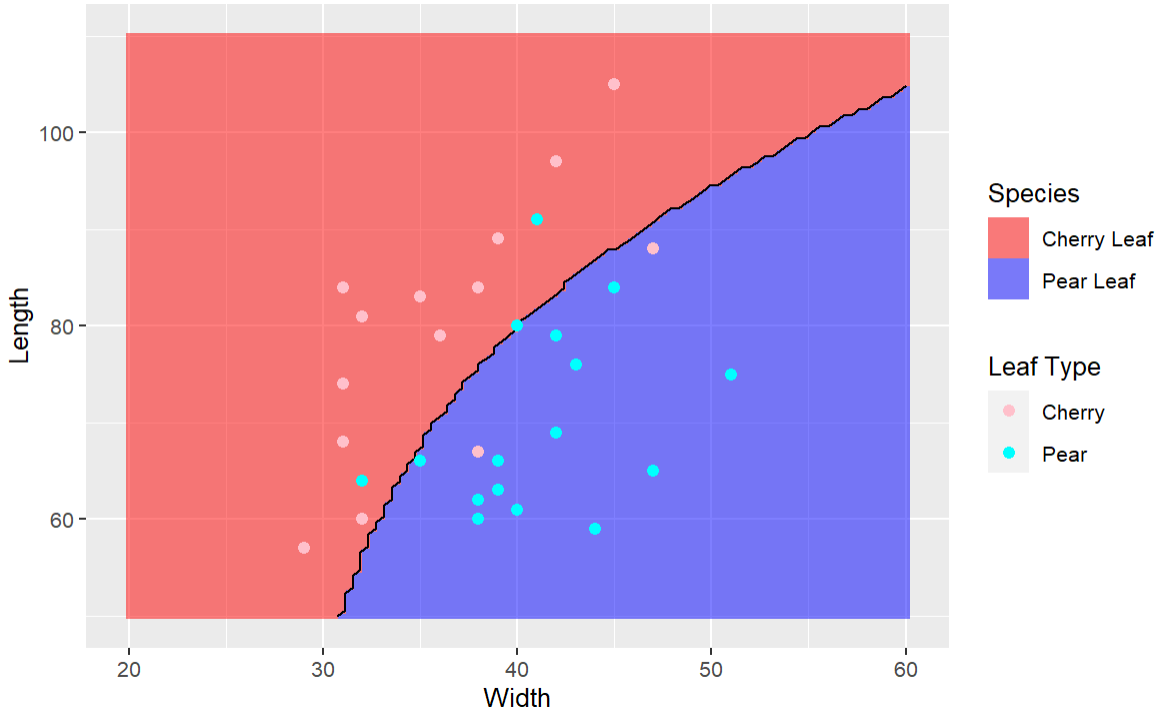
The classification rule for the LDA model was also plotted on top of all original 32 leaves. Below we can see the geometric interpretation of our model. We can also find that the equation for the line of our decision boundary is $y = -13.7932 + 2.303389x$.



Once this was done, we take the same steps to test our QDA model. Below are the results of the classification done by the QDA model.

Leaf #	Width(mm)	Length(mm)	QDA Classification
1	32	82	Cherry Leaf
2	38	52	Pear Leaf
3	40	76	Pear Leaf

The classification rule for the QDA model was also plotted on top of all original 32 leaves. Below we can see the geometric interpretation of our model.



Discussion

From the results of these two models, we can see that they both classify our test leaves in the same way, the first leaf in both models was classified as a cherry leaf, where the second and third leaves were classified as pear leaves. When we look at the geometric interpretation of both models, we can see that the LDA model has a straight line as a classification, where the classification for QDA has a curved line. One of the shortcomings of this project is that the measurements for the length and width of the leaves are measured by hand with a ruler, and only to the nearest millimeter. This means that each measurement will be slightly inaccurate which could lead to the models misclassifying some of the leaves. The ruler used to take the measurements may also be flawed and lead to poor measurements. For the LDA model we also assume the covariance matrices are the same when this may not be the case for the leaves in our data set. For both models we are also assuming that our predictors follow a multivariate normal distribution.

Conclusion

In conclusion, for this project 2 models were created to classify cherry leaves and pear leaves, these models were LDA and QDA. To create these models we used 32 total leaves, 16 cherry and 16 pear leaves. For each of these models we tested 3 new leaf measurements to see how they would be classified and each of the models classified the leaves in the same way. A graph of the decision boundary was also made for each of these models. From these graphs we can see that the decision boundary for the LDA model has a linear classification rule where the decision boundary for the QDA model has a curved classification rule.

References

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing,

Vienna, Austria. URL <https://www.R-project.org/.>

Wickham H, Bryan J (2022). _readxl: Read Excel Files_. R package version 1.4.1,
<https://CRAN.R-project.org/package=readxl>.
H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Appendix

```
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
#0 shows cherry leaves and 1 shows pear leaves
leaves = read_xlsx("C:\\Users\\ethan\\Downloads\\Stat 517\\Project #4\\Leaf_data.xlsx")
leaves
```

```
## # A tibble: 32 x 3
##    Leaf_type Width Length
##        <dbl> <dbl>  <dbl>
##  1         0    29     57
##  2         0    32     60
##  3         0    31     68
##  4         0    41     91
##  5         0    38     67
##  6         0    45    105
##  7         0    31     74
##  8         0    47     88
##  9         0    31     74
## 10         0    32     81
## # ... with 22 more rows
```

```
#Finding means and var-cov matrices
leaf0=subset(leaves, Leaf_type==0)
leaf1=subset(leaves, Leaf_type==1)

#mean of cherry leaves
width0.mean = mean(leaf0$Width)
length0.mean = mean(leaf0$Length)
#mean of pear leaves
width1.mean = mean(leaf1$Width)
length1.mean = mean(leaf1$Length)
#covariance matrices of both leaf types
cov0 = cov(leaf0[,2:3])
cov1 = cov(leaf1[,2:3])
# Calculate the pooled variance-covariance matrix
#This is for LDA
pooled_cov <- ((16 - 1) * cov0 + (16 - 1) * cov1) / (16 + 16 - 2)
```

```
#multivariate normal cherry leaves
fa = function(x,y){
  (1/(2*pi*sqrt(det(pooled_cov))))*exp(-0.5*t(c(x-width0.mean, y-length0.mean))%*%solve(pooled_cov)%*%c
```

```
}
#multivariate normal of pear leaves
fb = function(x,y){
  (1/(2*pi*sqrt(det(pooled_cov))))*exp(-0.5*t(c(x-width1.mean, y-length1.mean))%*%solve(pooled_cov)%*%c
}
```

```
#function to classify leaves with LDA
classify_lda <- function(width, length) {
  lambda <- fa(width, length) / fb(width, length)
  if (lambda > 1) {
    return("Cherry Leaf")
  } else if (lambda < 1) {
    return("Pear Leaf")
  } else {
    return("Undetermined")
  }
}
#New leaves to classify
classify_lda(32,82)
```

```
## [1] "Cherry Leaf"
```

```
classify_lda(38,52)
```

```
## [1] "Pear Leaf"
```

```
classify_lda(40,76)
```

```
## [1] "Pear Leaf"
```

```
# Calculate the coefficients for the decision boundary line using your decision rule
A <- solve(pooled_cov) %*% c(width1.mean - width0.mean, length1.mean - length0.mean)
B <- -0.5 * (t(c(width0.mean, length0.mean)) %*% solve(pooled_cov) %*% c(width0.mean, length0.mean) -
             t(c(width1.mean, length1.mean)) %*% solve(pooled_cov) %*% c(width1.mean, length1.mean))

#Intercept of decision boundary
(B / A[2])
```

```
##          [,1]
## [1,] -13.7932
```

```
#Slope of decison boundary
-A[1] / A[2]
```

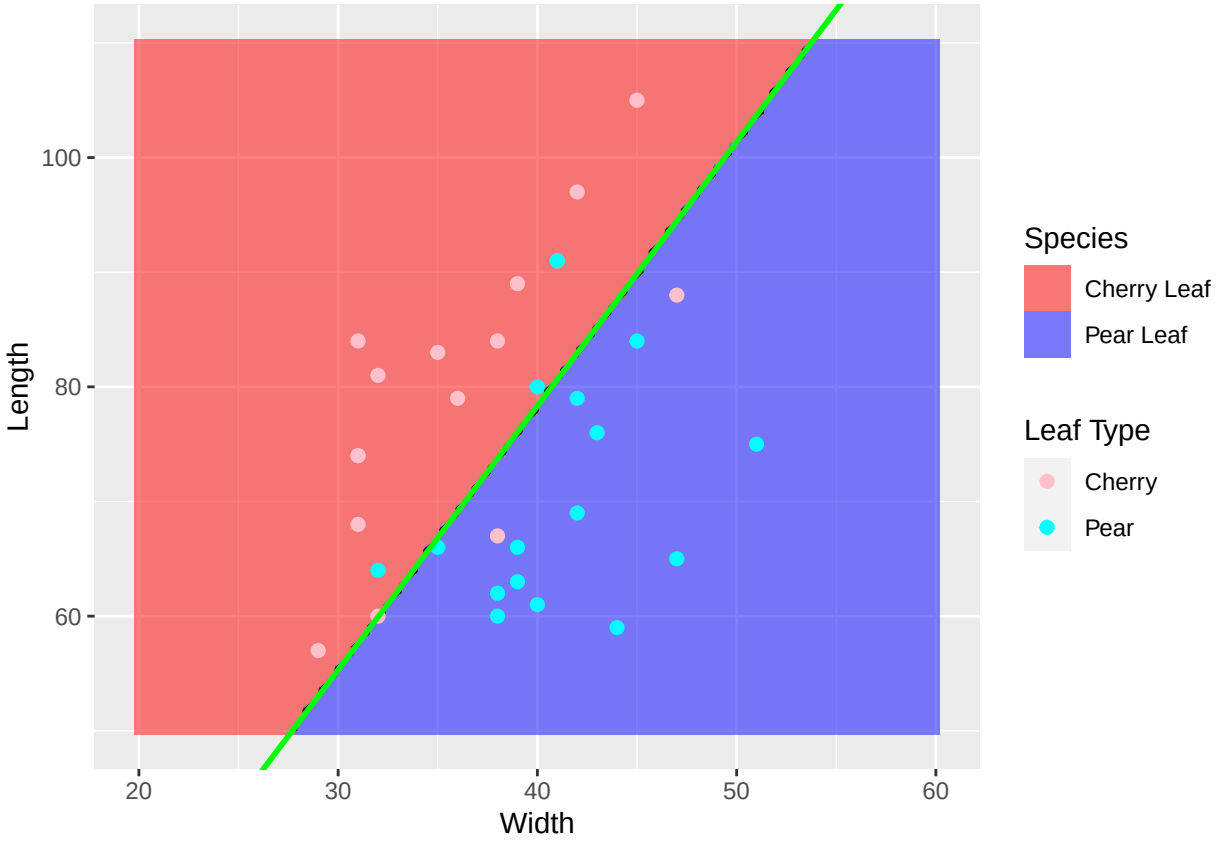
```
## [1] 2.303389
```

```
#plot of decision boundary
library(ggplot2)

# Create a dataframe with a grid of points to plot the decision boundary
```

```r
x <- seq(20, 60, length.out = 100)
y <- seq(50, 110, length.out = 100)
grid <- expand.grid(Width = x, Length = y)
grid$Species <- apply(grid, 1, function(row) classify_lda(row["Width"], row["Length"]))

# Plot of decision boundary for LDA with original data
ggplot() +
  geom_raster(data = grid, aes(x = Width, y = Length, fill = Species), alpha = 0.5) +
  geom_contour(data = grid, aes(x = Width, y = Length, z = ifelse(Species == "Cherry Leaf", 1, -1)),
               breaks = 0, color = "black") +
  geom_point(data = leaves, aes(x = Width, y = Length, color = factor(Leaf_type)), size = 2) +
  geom_abline(slope = -A[1] / A[2], intercept =(B / A[2]), color = "green", size = 1)+
  scale_fill_manual(values = c("red", "blue", "grey"), na.value = "grey") +
  scale_color_manual(values = c("pink", "cyan"), name = "Leaf Type", labels = c("Cherry", "Pear")) +
  labs(x = "Width", y = "Length")
```




```r
#now assuming covariance matrices are not equal
#this results in QDA
fa1 = function(x,y){
  (1/(2*pi*sqrt(det(cov0))))*exp(-0.5*t(c(x - width0.mean, y - length0.mean)) %*% solve(cov0) %*% c(x -
}

fb1 = function(x,y){
  (1/(2*pi*sqrt(det(cov1))))*exp(-0.5*t(c(x - width1.mean, y - length1.mean)) %*% solve(cov1) %*% c(x -
}
```

```r
#function to classify leaves with QDA
classify_qda <- function(width, length) {
  lambda <- fa1(width, length) / fb1(width, length)
  if (lambda > 1) {
    return("Cherry Leaf")
  } else if (lambda < 1) {
    return("Pear Leaf")
  } else {
    return("Undetermined")
  }
}
#New leaves to classify
classify_qda(32,82)
```

```
## [1] "Cherry Leaf"
```

```r
classify_qda(38,52)
```

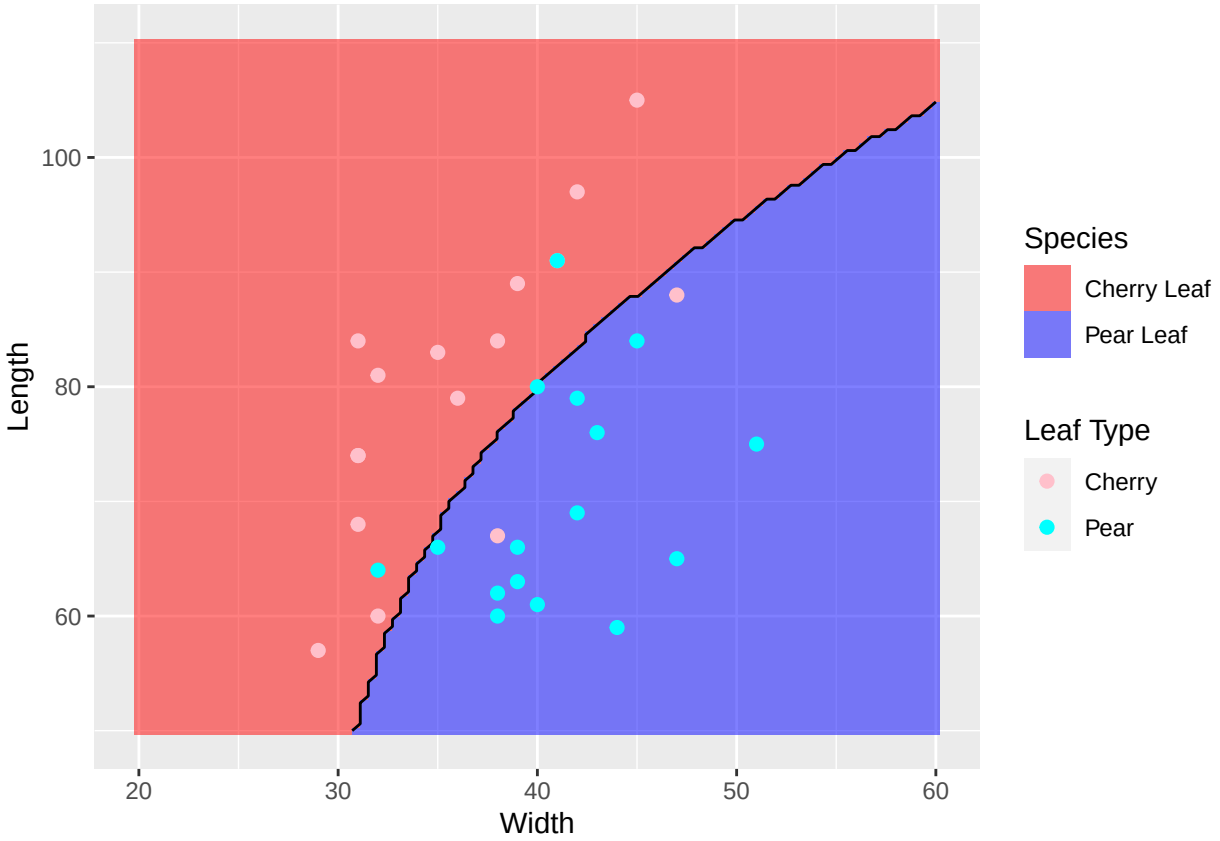
```
## [1] "Pear Leaf"
```

```r
classify_qda(40,76)
```

```
## [1] "Pear Leaf"
```

```r
# Create a dataframe with a grid of points to plot the decision boundary
x <- seq(20, 60, length.out = 100)
y <- seq(50, 110, length.out = 100)
grid <- expand.grid(Width = x, Length = y)
grid$Species <- apply(grid, 1, function(row) classify_qda(row["Width"], row["Length"]))

# Plot of decision boundary for QDA with original data
ggplot() +
  geom_raster(data = grid, aes(x = Width, y = Length, fill = Species), alpha = 0.5) +
  geom_contour(data = grid, aes(x = Width, y = Length, z = ifelse(Species == "Cherry Leaf", 1, -1)),
               breaks = 0, color = "black") +
  geom_point(data = leaves, aes(x = Width, y = Length, color = factor(Leaf_type)), size = 2) +
  scale_fill_manual(values = c("red", "blue", "grey"), na.value = "grey") +
  scale_color_manual(values = c("pink", "cyan"), name = "Leaf Type", labels = c("Cherry", "Pear")) +
  labs(x = "Width", y = "Length")
```




```
#citations
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2022). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2022},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```

```
citation("readxl")
```

```
##
## To cite package 'readxl' in publications use:
##
##   Wickham H, Bryan J (2022). _readxl: Read Excel Files_. R package
##   version 1.4.1, <https://CRAN.R-project.org/package=readxl>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {readxl: Read Excel Files},
##     author = {Hadley Wickham and Jennifer Bryan},
##     year = {2022},
##     note = {R package version 1.4.1},
##     url = {https://CRAN.R-project.org/package=readxl},
##   }
```

```
citation("ggplot2")
```

```
##
## To cite ggplot2 in publications, please use:
##
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2016},
##     isbn = {978-3-319-24277-4},
##     url = {https://ggplot2.tidyverse.org},
##   }
```