Quick Intro to R

Contents

Welcome to R

In this very brief introduction to R, we work up to a few examples from two popular packages: *dplyr* for data wrangling and *ggplot2* for plotting. Additionally, we give two examples of some common statistical models. We will use this code later in an interactive Shiny application. These topics comprise some of R's greatest strengths: + Data wrangling + Plotting/visuals + Statistical model fitting + Interactive applications (Shiny) + All in a fantastic GUI (RStudio)

You can pull this entire repo down from GitHub using this url: https://github.com/milliman/ADA_IntroToR_Shiny.git

Packages

```
# install.packages(c("dplyr", "ggplot2"))
library(dplyr)
library(ggplot2)
```

```
sessionInfo()$R.version$version.string
```

```
## [1] "R version 3.4.4 (2018-03-15)"
```

```
sapply(sessionInfo()$otherPkgs, function(x) x$Version)
```

```
## ggplot2   dplyr
## "2.2.1" "0.7.4"
```

Objects

Scalars

```
1 + 1
```

```
## [1] 2
```

```
a <- 1 + 1
```

Vectors

We show various ways to make similar numeric vectors:

```
v1 <- 1:3
v2 <- c(1, 2, 3)
v3 <- seq(1, 3, by = 1)
v4 <- seq(1, 3, length.out = 3)
v1
```

```
## [1] 1 2 3
```

```
v2
```

```
## [1] 1 2 3
```

```
v3
```

```
## [1] 1 2 3
```

```
v4
```

```
## [1] 1 2 3
```

```
v1*v2
```

```
## [1] 1 4 9
```

Character and factor vectors

```
c1 <- c("one", "two", "three", "one", "two", "three")
c1
```

```
## [1] "one"   "two"   "three" "one"   "two"   "three"
```

```
f1 <- factor(c("one", "two", "three", "one", "two", "three"))
f1
```

```
## [1] one   two   three one   two   three
## Levels: one three two
```

Matrices

```
m1 <- matrix(data = c(1, 2, 3, 4, 5, 6, 7, 8, 9),
             nrow = 3,
             ncol = 3,
             byrow = FALSE)

m2 <- matrix(data = rnorm(n = 9, mean = 0, sd = 1),
             nrow = 3,
             ncol = 3,
             byrow = TRUE)
m1 * m2
```

```
##            [,1]        [,2]     [,3]
## [1,] -0.4045566  0.07426224 2.079391
## [2,]  1.3884093  2.37323044 5.893639
## [3,] -3.5391530 -5.66007191 5.504147
```

```
m1 %*% m2
```

```
##           [,1]      [,2]      [,3]
## [1,] -5.885762 -4.686267  7.524879
## [2,] -6.775832 -5.136401  9.170211
## [3,] -7.665901 -5.586535 10.815544
```

Data frame objects

```
d <- data.frame(A = 1:4,
                B = c("red", "blue", "yellow", "green"))
class(d)
```

```
## [1] "data.frame"
```

```
class(iris)
```

```
## [1] "data.frame"
```

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
```

```
##
##
```

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 ...
```

```
head(iris)
```

```
##   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
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Reading and writing data

CSV

```
write.csv(x = iris,
          file = "iris.csv",
          row.names = F)
mattysdf <- read.csv(file = "iris.csv")
```

RDS

```
saveRDS(object = iris,
        file = "iris.RDS")
mattysdf <- readRDS("iris.RDS")
```

Data wrangling

We turn to our favorite data wrangling package, dplyr.

Select specific columns

```
mattysdf %>%
  select(Petal.Length, Species) %>%
  head(5)
```

```
##   Petal.Length Species
## 1          1.4  setosa
```

```
## 2          1.4  setosa
## 3          1.3  setosa
## 4          1.5  setosa
## 5          1.4  setosa
```

Filter to specific rows

```
mattysnewdf <- mattysdf %>%
  filter(Species == "versicolor")
```

"Mutate" on new columns

```
mattysnewdf2 <- mattysdf %>%
  mutate(Sepal.Area = Sepal.Width*Sepal.Length,
         Petal.Area = Petal.Width*Petal.Length)
```

Pivot table

```
mattysnewdf2 %>%
  group_by(Species) %>%
  summarize(Sepal.Area.mean = mean(Sepal.Area),
            Sepal.Area.sd = sd(Sepal.Area),
            Petal.Area.mean = mean(Petal.Area),
            Petal.Area.sd = sd(Petal.Area))
```

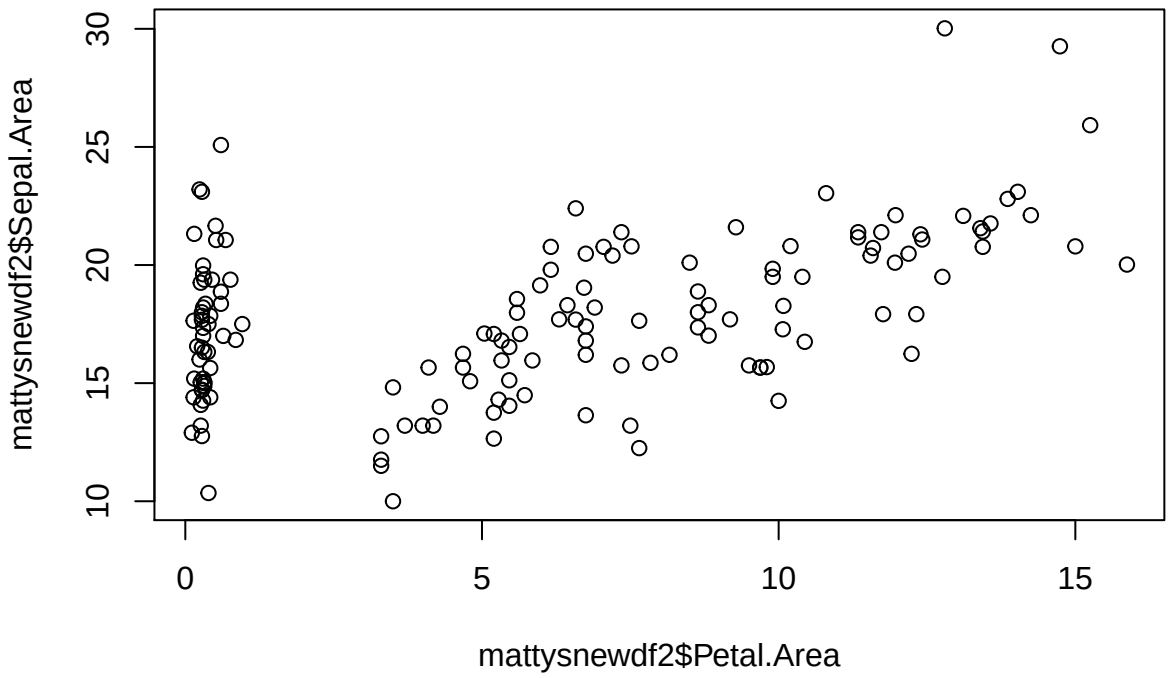
```
## # A tibble: 3 x 5
##      Species Sepal.Area.mean Sepal.Area.sd Petal.Area.mean Petal.Area.sd
##       <fctr>           <dbl>         <dbl>           <dbl>         <dbl>
## 1     setosa         17.2578      2.933775          0.3656     0.1811546
## 2 versicolor         16.5262      2.866882          5.7204     1.3684029
## 3  virginica         19.6846      3.458783         11.2962     2.1574124
```

Plotting

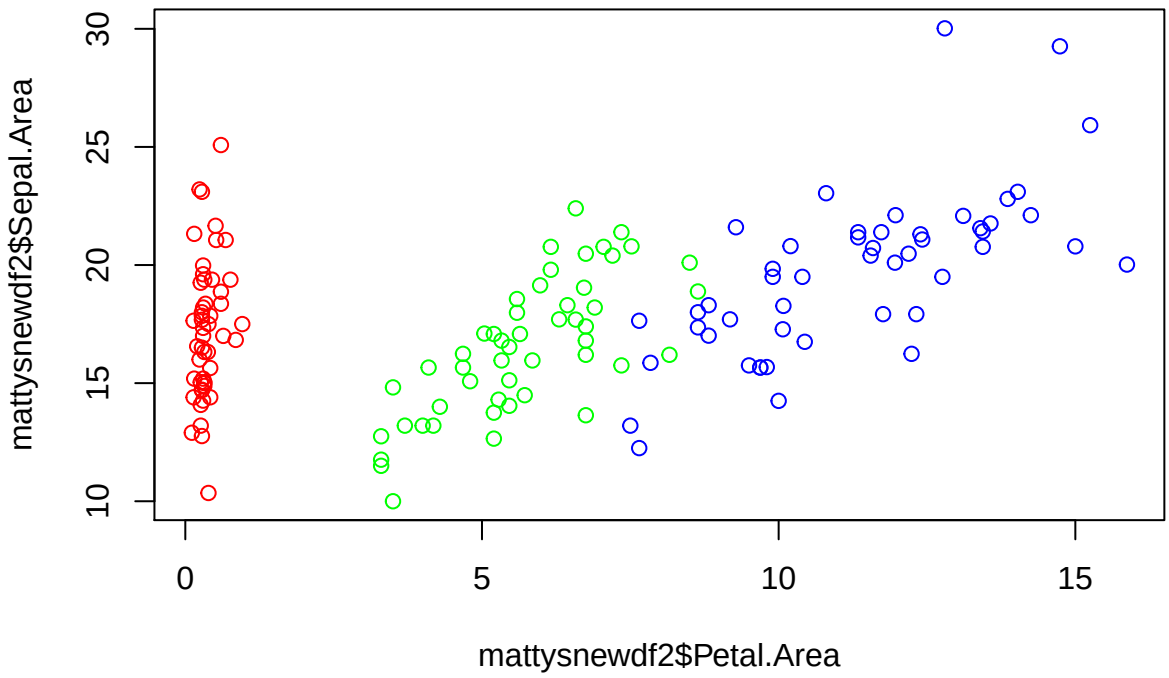
Base R

Scatter plot

```
plot(x = mattysnewdf2$Petal.Area,
     y = mattysnewdf2$Sepal.Area)
```

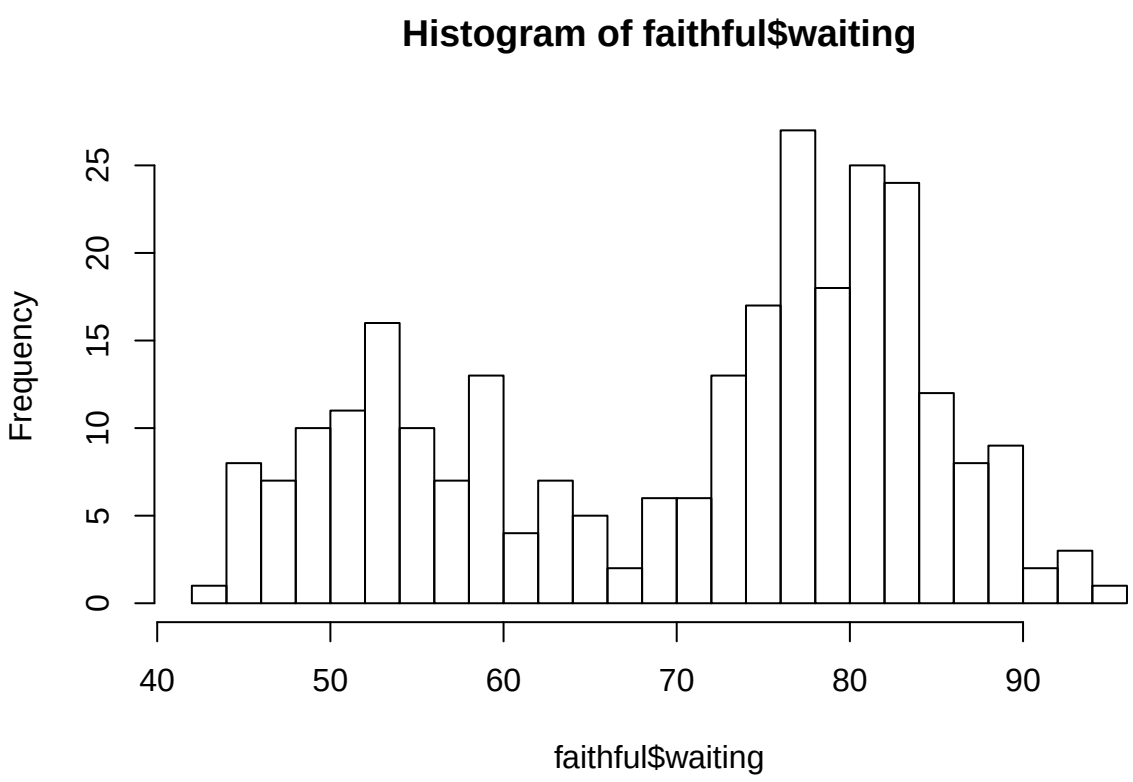



```
plot(x = mattysnewdf2$Petal.Area,
     y = mattysnewdf2$Sepal.Area,
     col = ifelse(mattysnewdf2$Species == "setosa", "red",
                  ifelse(mattysnewdf2$Species == "versicolor", "green", "blue")))
```

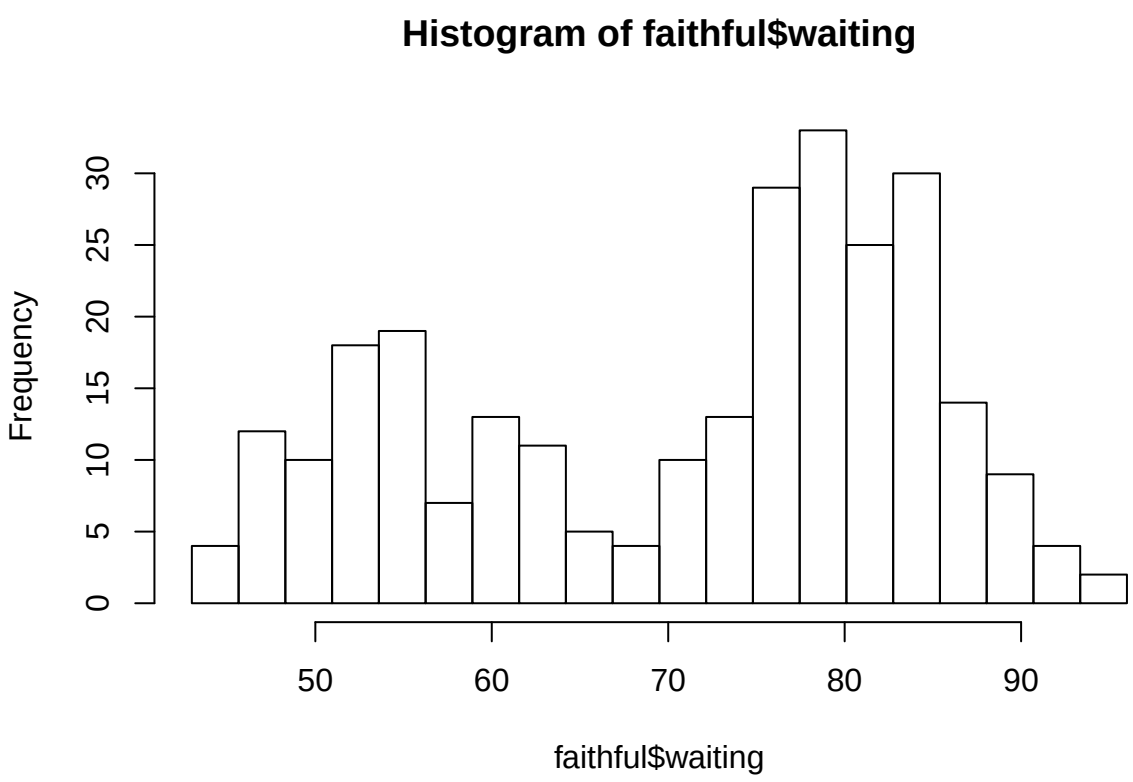


Histogram

```
hist(x = faithful$waiting,
     breaks = 20)
```



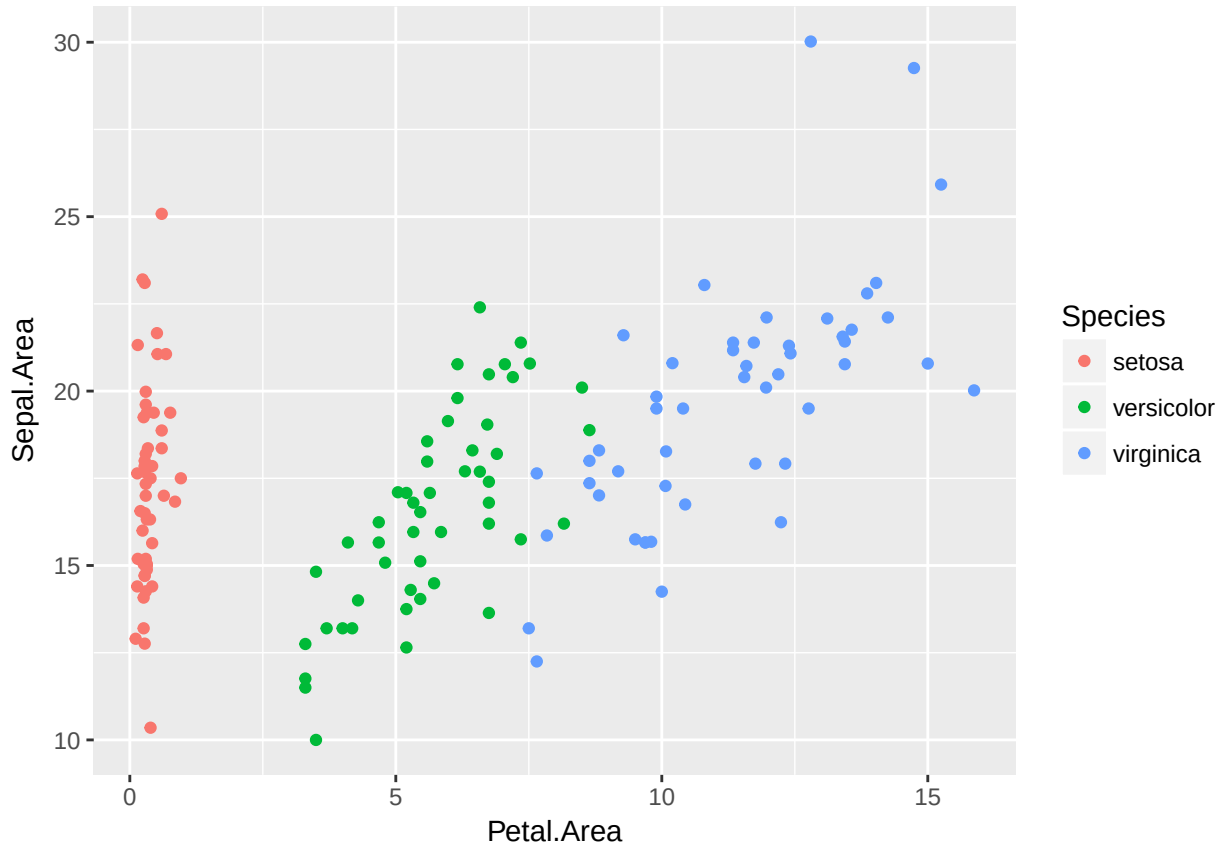

```
hist(x = faithful$waiting,
     breaks = seq(min(faithful$waiting), max(faithful$waiting), length.out = 20 + 1))
```



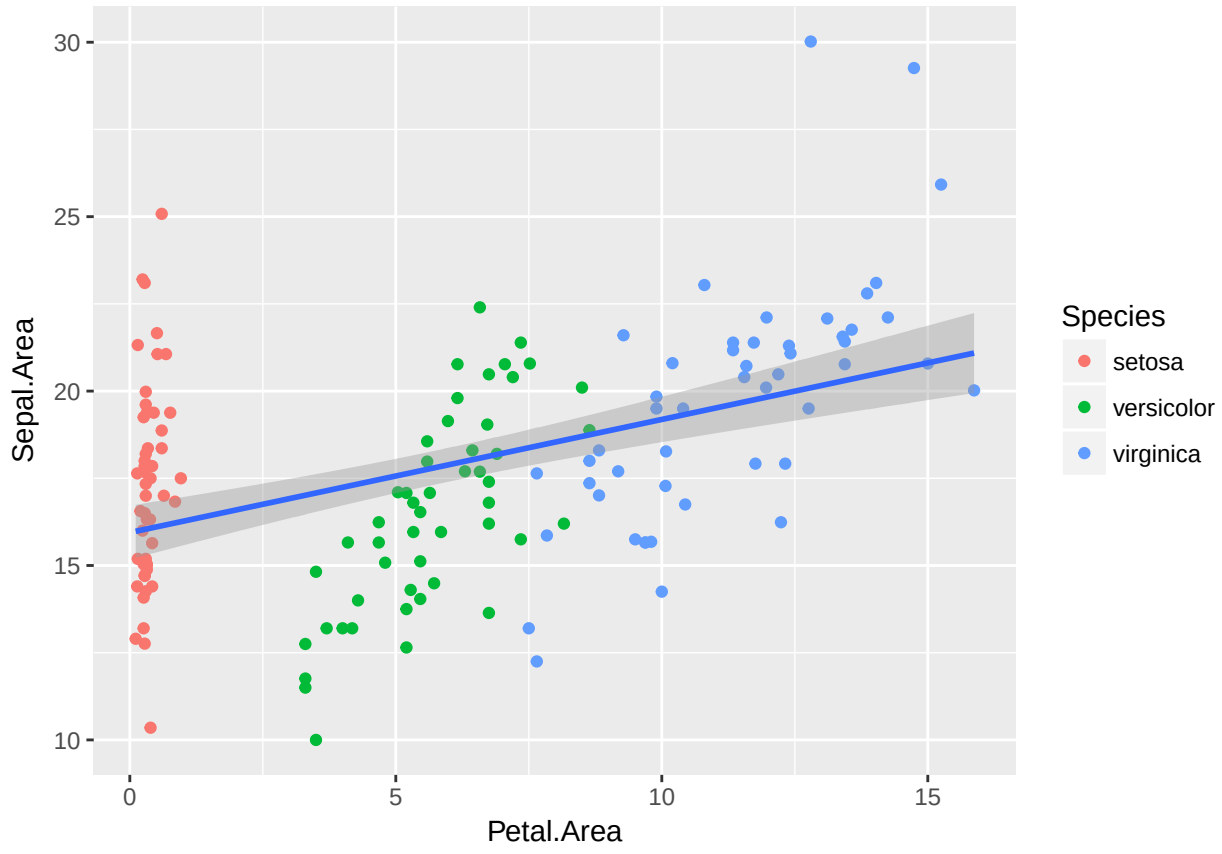
ggplot

Scatter plot

```
mattysnewdf2 %>%
  ggplot(mapping = aes(x = Petal.Area, y = Sepal.Area)) +
  geom_point(mapping = aes(color = Species))
```

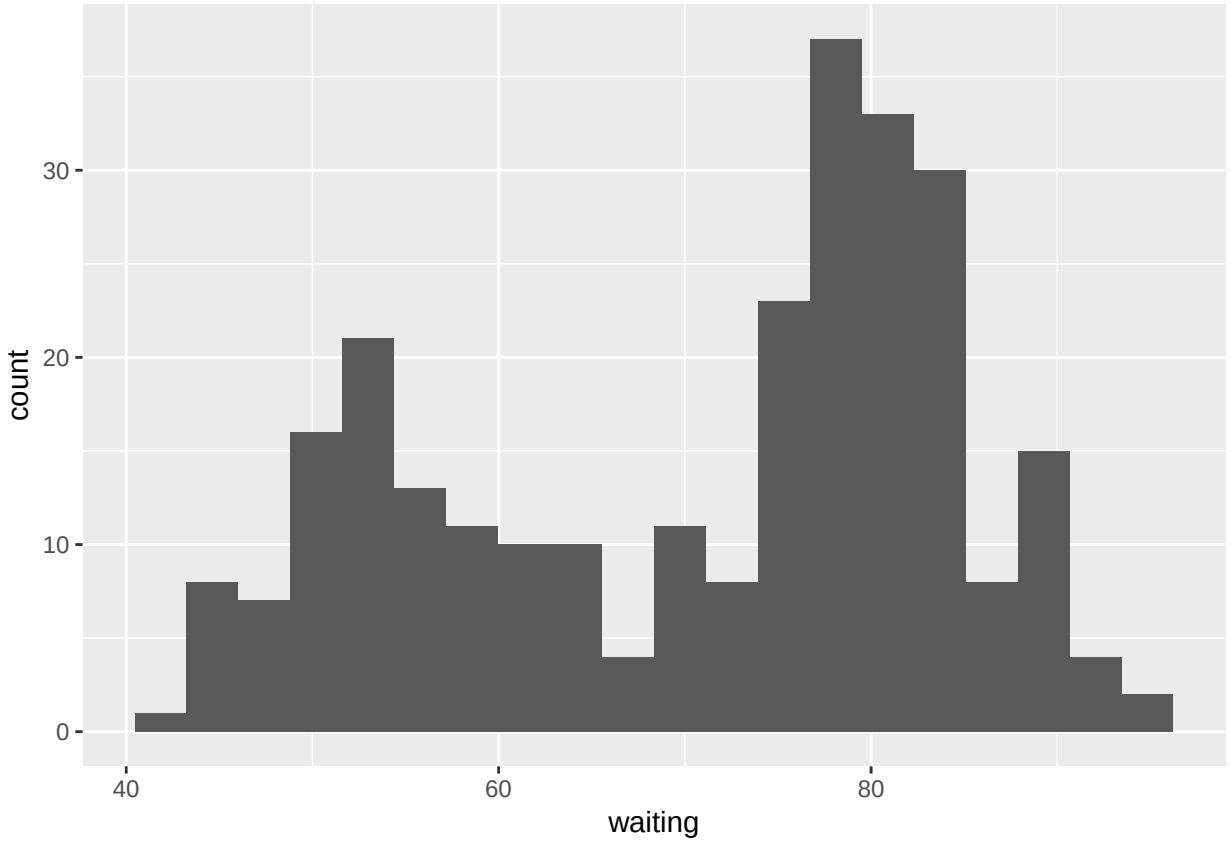



```
# More cooler scatter plot
mattysnewdf2 %>%
  ggplot(mapping = aes(x = Petal.Area, y = Sepal.Area)) +
  geom_point(mapping = aes(color = Species)) +
  geom_smooth(method = "lm")
```

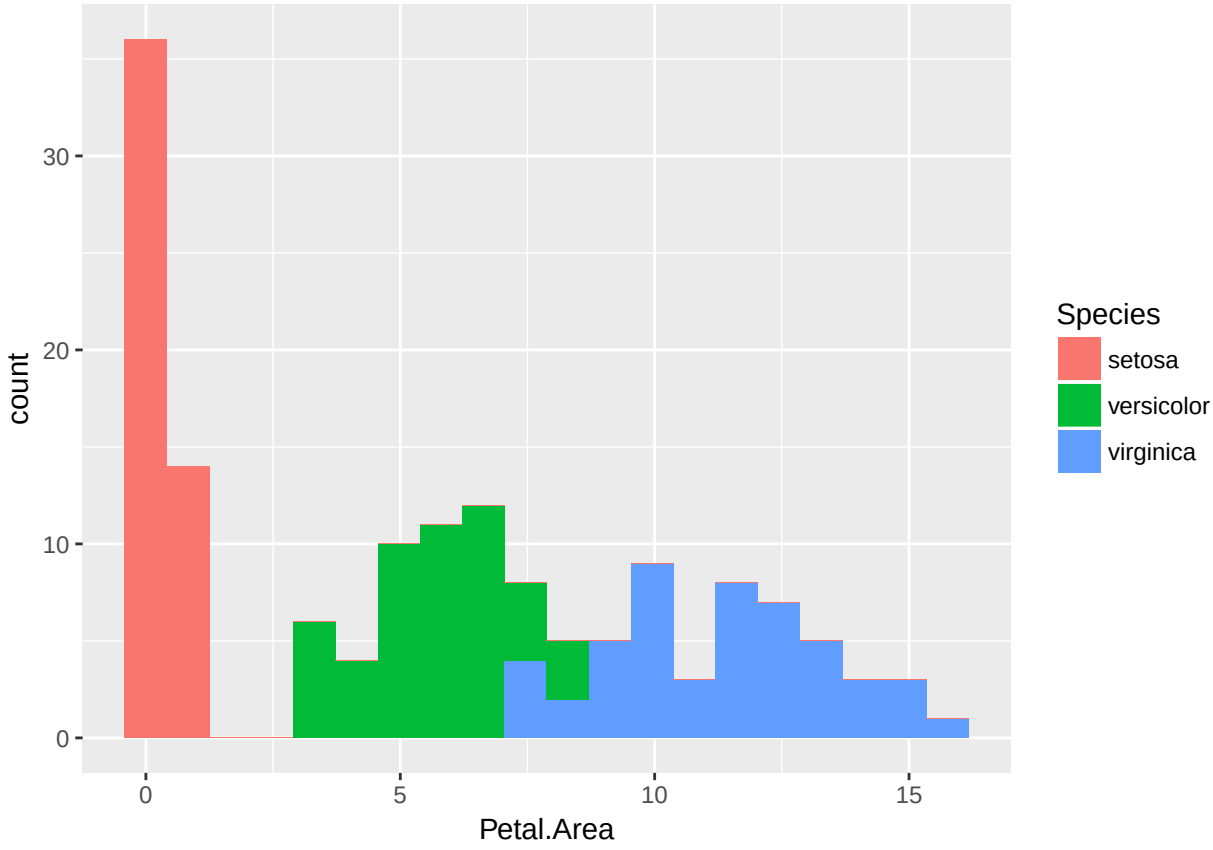


Histogram

```
faithful %>%
  ggplot(mapping = aes(x = waiting)) +
  geom_histogram(bins = 20)
```




```
# More betterer histogram
mattysnewdf2 %>%
  ggplot(mapping = aes(x = Petal.Area)) +
  geom_histogram(mapping = aes(fill = Species),
                 bins = 20)
```



Statistical models

R has a plethora of functions to create statistical models. Below we share two of the classics.

OLS

```
faithful.model <- lm(formula = eruptions ~ waiting,
                     data = faithful)
summary(faithful.model)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

Logistic GLM

```
iris.model <- glm(formula = versicolor ~ Sepal.Area + Petal.Area,
                  data = mattysnewdf2 %>%
                    mutate(versicolor = ifelse(Species == "versicolor", 1, 0)),
                  family = binomial(link = "logit"))
summary(iris.model)
```

```
##
## Call:
## glm(formula = versicolor ~ Sepal.Area + Petal.Area, family = binomial(link = "logit"),
##     data = mattysnewdf2 %>% mutate(versicolor = ifelse(Species ==
##         "versicolor", 1, 0)))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5134  -0.9092  -0.6955   1.1504   2.0085
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.25729    1.12538   2.894 0.003799 **
## Sepal.Area  -0.25166    0.07117  -3.536 0.000406 ***
## Petal.Area   0.07684    0.04679   1.642 0.100513
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.95  on 149  degrees of freedom
## Residual deviance: 176.03  on 147  degrees of freedom
## AIC: 182.03
##
## Number of Fisher Scoring iterations: 4
```