My tidymodels notes

Matt Nickodemus

2021-01-17

Contents

Chapter 1

Preduction

Where in the world does this shit go? This goes in front of the introduction, so I guess this is called the preduction? Maybe. Well, why the hell not?

Chapter 2

Introduction

These are my notes on the tidymodels metapackage. These notes are copied from several other sources, so nothing here should be considered by me. This is just my collection of notes, thoughts, and things.

Chapter 3

Literature

Here is a review of existing methods.

Chapter 4

Clustering

These are my notes from the tutorial on K-means clustering in the tidymodels framework. They have some really nice pictures for this process, so I thought I would copy them and make them into a book.

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --

## v broom     0.7.3      v recipes   0.1.15
## v dials     0.0.9      v rsample   0.0.8
## v dplyr     1.0.2      v tibble    3.0.4
## v ggplot2   3.3.2      v tidyr     1.1.2
## v infer     0.5.3      v tune      0.1.2
## v modeldata 0.1.0      v workflows 0.2.1
## v parsnip   0.1.4      v yardstick 0.0.7
## v purrr     0.3.4

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
```

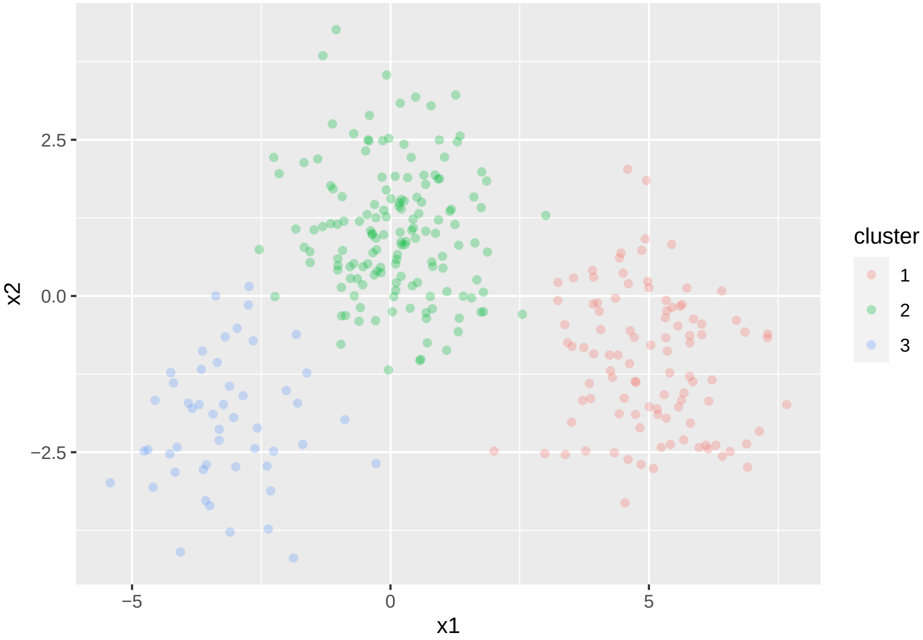
Here we are going to create a fake data set to do our clustering on.

```
set.seed(27)

centers <- tibble(
  cluster = factor(1:3),
  num_points = c(100, 150, 50),  # number points in each cluster
  x1 = c(5, 0, -3),              # x1 coordinate of cluster center
  x2 = c(-1, 1, -2)              # x2 coordinate of cluster center
)

labelled_points <-
  centers %>%
  mutate(
    x1 = map2(num_points, x1, rnorm),
    x2 = map2(num_points, x2, rnorm)
  ) %>%
  select(-num_points) %>%
  unnest(cols = c(x1, x2))

ggplot(labelled_points, aes(x1, x2, color = cluster)) +
  geom_point(alpha = 0.3)
```



```
points <-
  labelled_points %>%
  select(-cluster)

kclust <- kmeans(points, centers = 3)
kclust
```

```
## K-means clustering with 3 clusters of sizes 148, 51, 101
##
## Cluster means:
##             x1        x2
## 1  0.08853475  1.045461
## 2 -3.14292460 -2.000043
## 3  5.00401249 -1.045811
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
## [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 298.9415 108.8112 243.2092
##  (between_SS / total_SS =  82.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```