Sunayu Tech Task
K-means Clustering: Image Compression

Michael Vatt

09 Dec 20

Contents


```
# Create a Function to Check for Installed Packages and Install if They Are Not Installed

install <- function(packages){
  new.packages <- packages[!(packages %in% installed.packages()[, "Package"])]
  if (length(new.packages))
    install.packages(new.packages, dependencies = TRUE, repos = "http://cran.us.r-project.org")
  sapply(packages, require, character.only = TRUE)
}

# Install

packages <- c("caTools", "car", "caret", "cluster", "Clustering", "corpus", "corrplot", "data.table",
                  "dendextend", "doParallel", "dplyr", "e1071", "factoextra", "FactoMineR", "fpc",
                  "GGally", "ggplot2", "ggthemes", "gridExtra", "imager", "jpeg", "kableExtra", "knitr"
                  "ldatuning", "magrittr", "mclust", "NbClust", "petro.One", "plotly", "plotrix", "png"
                  "qdap", "qdapTools", "quanteda", "randomForest", "readxl", "reshape", "RColorBrewer",
                  "rlist", "RWeka", "scales", "SentimentAnalysis", "sentimentr", "SnowballC", "stats",
                  "stm", "stringr", "syuzhet", "tensorflow", "tidyr", "tidytext", "tidyverse", "tm",
                  "topicmodels", "viridisLite", "wordcloud", "xlsx", "zoo")

install(packages)
```

```
# Call the installed packages

#library(plyr) # plyr is required to be loaded before dplyr or issues may arise
library(dplyr) # dplyr needed for efficient loading of loadApp()

loadApp <- function() {

my_library <- c("caTools", "car", "caret", "cluster", "Clustering", "corpus", "corrplot", "data.table",
                "dendextend", "doParallel", "dplyr", "e1071", "factoextra", "FactoMineR", "fpc",
                "GGally", "ggplot2", "ggthemes", "gridExtra", "imager", "jpeg", "kableExtra", "knitr"
                "ldatuning", "magrittr", "mclust", "NbClust", "petro.One", "plotly", "plotrix", "png"
                "qdap", "qdapTools", "quanteda", "randomForest", "readxl", "reshape", "RColorBrewer",
                "rlist", "RWeka", "scales", "SentimentAnalysis", "sentimentr", "SnowballC", "stats",
                "stm", "stringr", "syuzhet", "tensorflow", "tidyr", "tidytext", "tidyverse", "tm",
                "topicmodels", "viridisLite", "wordcloud", "xlsx", "zoo")

install.lib <- my_library[!my_library %>% installed.packages()]

for(lib in install.lib) install.packages(lib, dependencies = TRUE)

sapply(my_library, require, character = TRUE)

}

loadApp()
```

Image Compression

This data set includes the following variables:

Objective

This was a task issued by Sunayu as an evaluation of approach, style, ability, and performance. The R script will compress an image successfully by removing some information without losing the overall detail. Thus, the image will require less memory for storage, interpretation, and it still remains meaningful and informative.

Dataset

This algorithm too an image "Satellite.png" and compressed the image from 6.5MB at full color to several levels of clusters ranging from 2 to 128 - each level increased by a multiplication of 2.

```
split <- detectCores(TRUE)
cl <- makePSOCKcluster(split)
registerDoParallel(cl)
```

```
image <- readPNG("C:/My Desktop/k means/Images/Satellite.png") # Read image into R

class(image) # Ensure data was read in
```

```
## [1] "array"
```

```
original_dim <- dim(image) # Keep original dimensions
dim(image) <- c(dim(image)[1]*dim(image)[2],3) # Reshape dimensions
dim(image) # Verify it worked
```

```
## [1] 8294400       3
```

Compression Algorithm

Here we will compress the image into compressed images with clusters of 2, 4, 8, 16, 32, 64, and 128.

This compression algorithm caused the original image to go from 6.5MB to:

- 2 clusters: 262KB
- 4 clusters: 459KB
- 8 clusters: 647KB
- 16 clusters: 1.05MB
- 32 clusters: 1.54MB
- 64 clusters: 1.94MB
- 128 clusters: 2.57MB

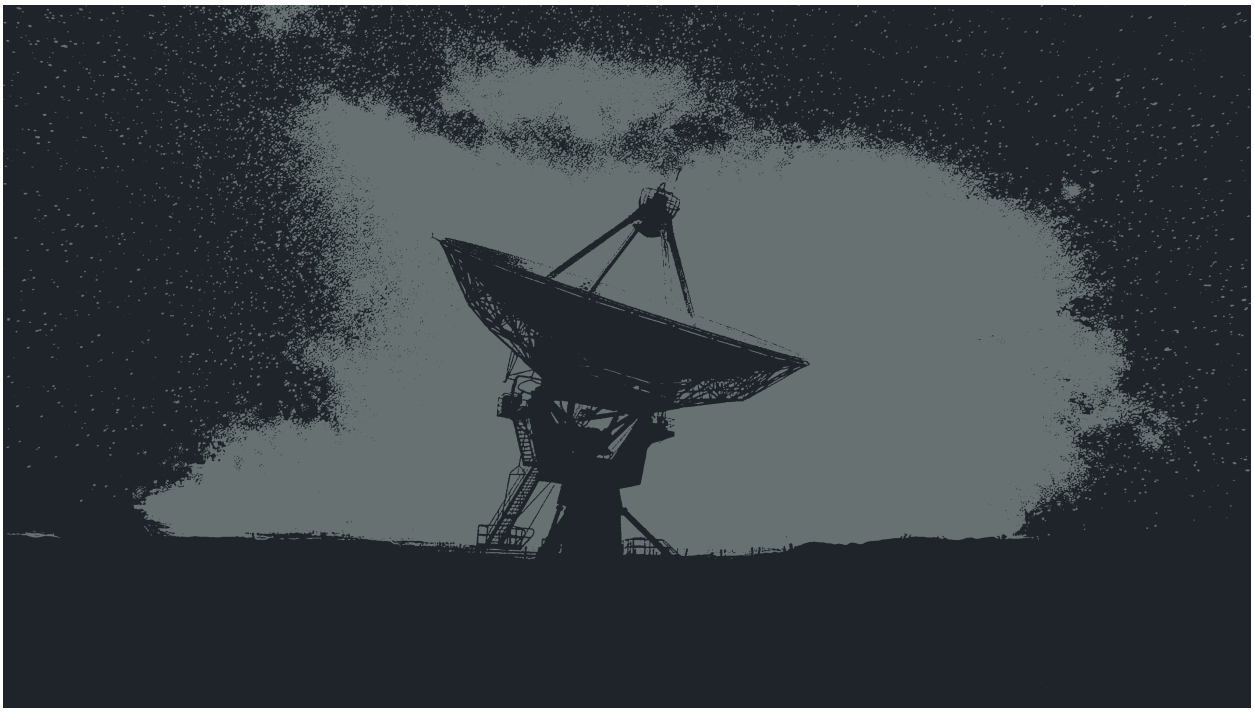
Here is the original photo for comparison:

```
include_graphics("C:/My Desktop/k means/Images/Satellite.png") # Output original image
```

Now here are the compressed images:

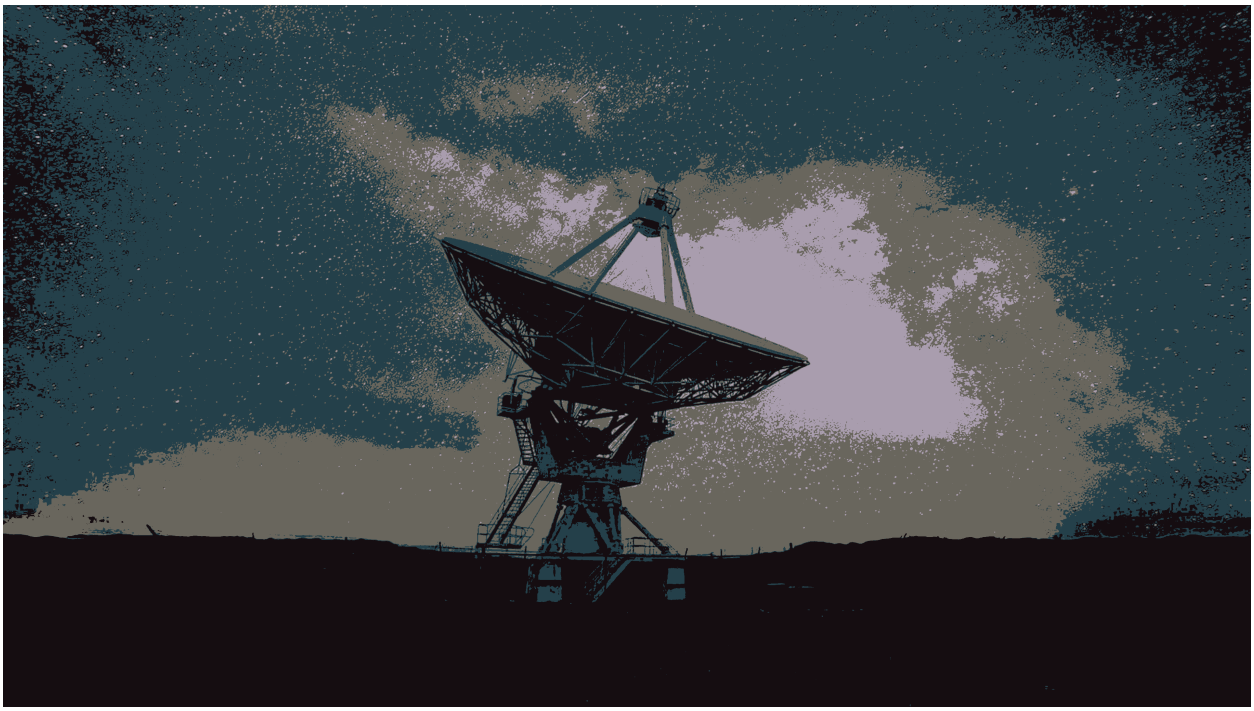
Compressed image with 2 clusters

```
# Compress with 2 centers

n <- 2
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-2.png") # Output compressed image
```



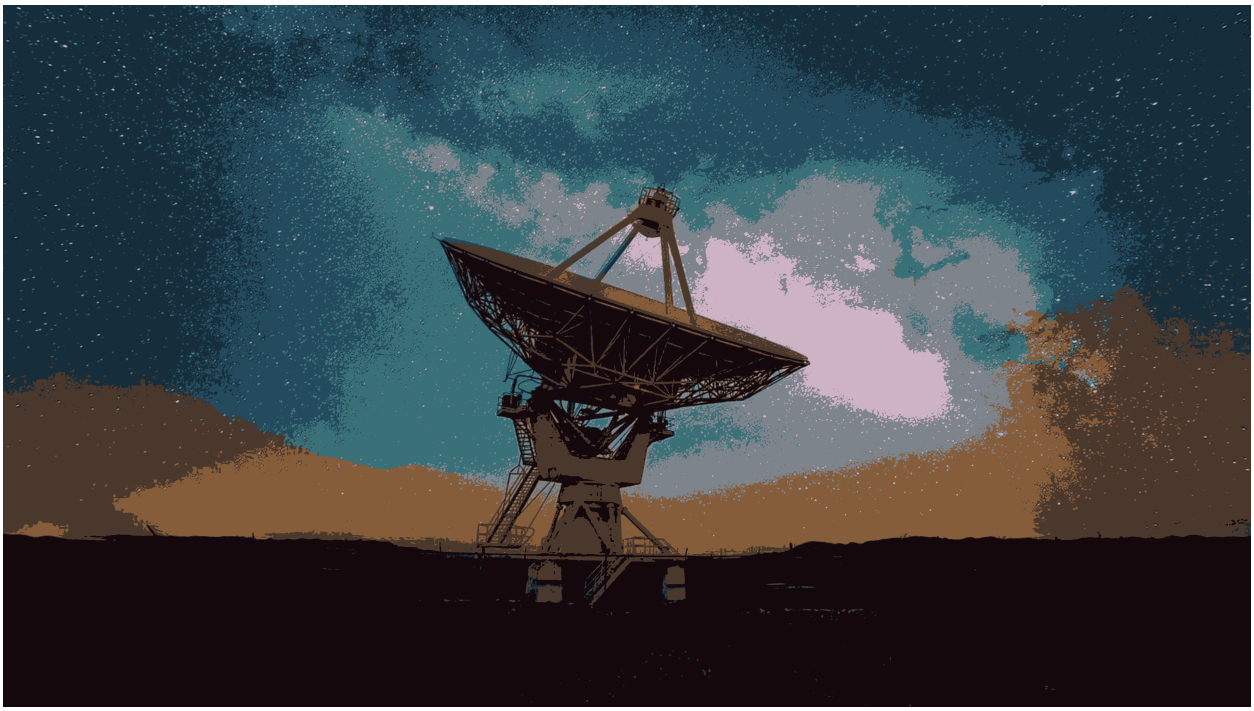
Compressed image with 4 clusters

```
# Compress with 4 centers

n <- 4
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-4.png") # Output compressed image
```



Compressed image with 8 clusters

```
# Compress with 8 centers

n <- 8
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-8.png") # Output compressed image
```

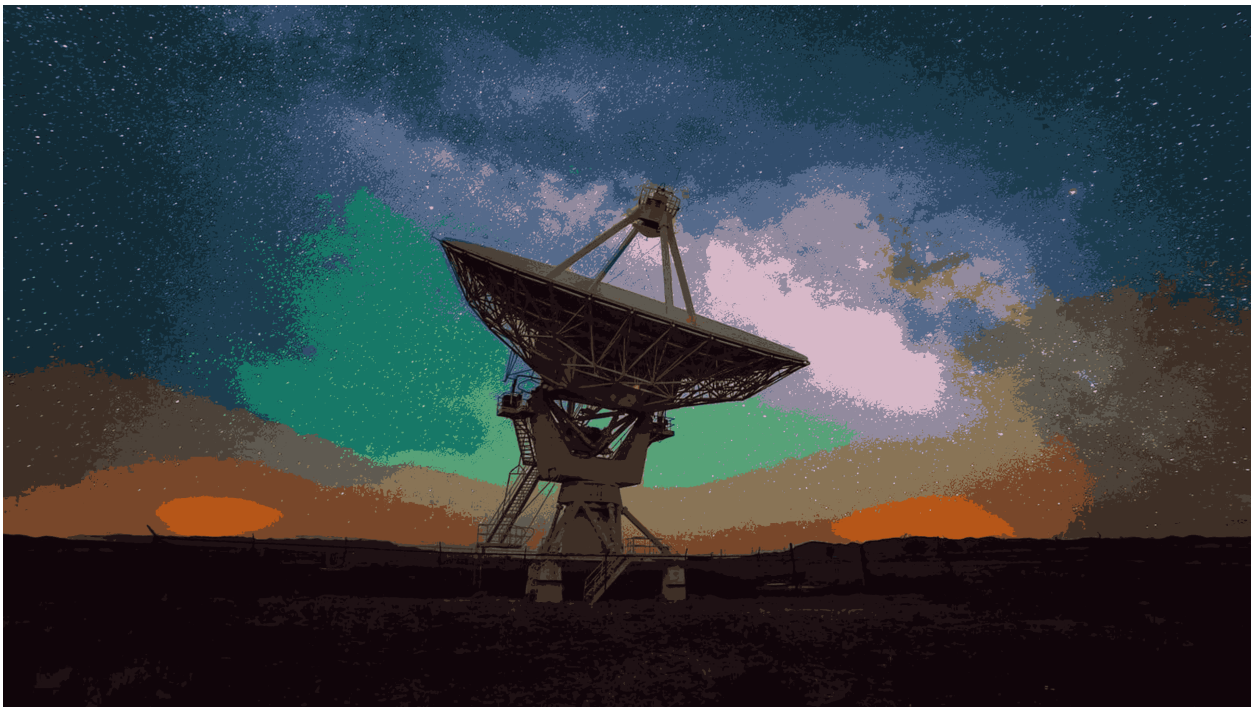


Compressed image with 16 clusters

```
# Compress with 16 centers

n <- 16
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 414720000)
```

```
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-16.png") # Output compressed image
```



Compressed image with 32 clusters

```
# Compress with 32 centers

n <- 32
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 414720000)
```

```
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-32.png") # Output compressed image
```

Compressed image with 64 clusters

```
# Compress with 64 centers

n <- 64
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 414720000)
```

```
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-64.png") # Output compressed image
```

Compressd image with 128 clusters

```
# Compress with 128 centers

n <- 128
kmeans_image <- kmeans(image, centers = n, iter.max = 100) # Run kmeans clustering algorithm
img <- kmeans_image$centers[kmeans_image$cluster,] # Retrieve colors
dim(img) <- original_dim # Reshape back to original dimensions
include_graphics("C:/My Desktop/k means/Images/Satellite-compressed-128.png") # Output compressed image
```

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] zoo_1.8-8              xlsx_0.6.4.2           wordcloud_2.6
##  [4] viridisLite_0.3.0      topicmodels_0.2-11     tm_0.7-7
##  [7] NLP_0.2-0              forcats_0.5.0          purrr_0.3.4
## [10] readr_1.3.1            tibble_3.0.1           tidyverse_1.3.0
## [13] tidytext_0.2.6         tidyr_1.1.2            tensorflow_2.2.0
## [16] syuzhet_1.0.4          stringr_1.4.0          stm_1.3.5
## [19] SnowballC_0.7.0        sentimentr_2.7.1       SentimentAnalysis_1.3-3
## [22] scales_1.1.1           RWeka_0.4-43           rlist_0.4.6.1
## [25] reshape_0.8.8          readxl_1.3.1           randomForest_4.6-14
## [28] quanteda_2.1.1         qdap_2.4.3             RColorBrewer_1.1-2
```

```
## [31] qdapTools_1.3.5         qdapRegex_0.7.2         qdapDictionaries_1.0.7
## [34] png_0.1-7               plotrix_3.7-8           plotly_4.9.2.1
## [37] petro.One_0.2.3         NbClust_3.0             mclust_5.4.6
## [40] ldatuning_1.0.2         knitr_1.30              kableExtra_1.2.1
## [43] jpeg_0.1-8.1            imager_0.42.3           magrittr_1.5
## [46] gridExtra_2.3           ggthemes_4.2.0          GGally_2.0.0
## [49] fpc_2.2-8               FactoMineR_2.3          factoextra_1.0.7
## [52] e1071_1.7-3             dplyr_1.0.0             doParallel_1.0.15
## [55] iterators_1.0.12        foreach_1.5.0           dendextend_1.14.0
## [58] data.table_1.12.8       corrplot_0.84           corpus_0.10.1
## [61] Clustering_1.6          cluster_2.1.0           caret_6.0-86
## [64] ggplot2_3.3.2           lattice_0.20-41         car_3.0-10
## [67] carData_3.0-4           caTools_1.18.0
##
## loaded via a namespace (and not attached):
##   [1] ClusterR_1.2.2       prabclus_2.3-2       ModelMetrics_1.2.2.2
##   [4] stopwords_2.0        bit64_4.0.5          rpart_4.1-15
##   [7] RCurl_1.98-1.2       generics_0.0.2       openNLP_0.2-7
##  [10] usethis_1.6.3        gama_1.0.3           RSQLite_2.2.0
##  [13] openNLPdata_1.5.3-4  chron_2.3-56         bit_4.0.4
##  [16] tokenizers_0.2.1     webshot_0.5.2        xml2_1.3.2
##  [19] lubridate_1.7.9      assertthat_0.2.1     viridis_0.5.1
##  [22] gower_0.2.1          amap_0.8-18          xfun_0.17
##  [25] hms_0.5.3            rJava_0.9-12         evaluate_0.14
##  [28] DEoptimR_1.0-8       fansi_0.4.1          dbplyr_1.4.4
##  [31] igraph_1.2.5         DBI_1.1.0            htmlwidgets_1.5.1
##  [34] apcluster_1.4.8      stats4_3.6.3         ellipsis_0.3.1
##  [37] backports_1.1.10     RcppParallel_5.0.2   vctrs_0.3.2
##  [40] abind_1.4-5          withr_2.2.0          robustbase_0.93-6
##  [43] lazyeval_0.2.2       crayon_1.3.4         recipes_0.1.12
##  [46] pkgconfig_2.0.3      slam_0.1-47          nlme_3.1-148
##  [49] nnet_7.3-14          rlang_0.4.7          diptest_0.75-7
##  [52] lifecycle_0.2.0      lexicon_1.2.1        modelr_0.1.8
##  [55] cellranger_1.1.0     bmp_0.3              tiff_0.1-5
##  [58] Matrix_1.2-18        reprex_0.3.0         base64enc_0.1-3
##  [61] whisker_0.4          bitops_1.0-6         advclust_0.4
##  [64] pROC_1.16.2          blob_1.2.1           venneuler_1.1-0
##  [67] leaps_3.1            memoise_1.1.0        plyr_1.8.6
##  [70] compiler_3.6.3       pvclust_2.2-0        clue_0.3-57
##  [73] cli_2.0.2            janeaustenr_0.1.5    MASS_7.3-51.6
##  [76] tidyselect_1.1.0     stringi_1.4.6        yaml_2.2.1
##  [79] ggrepel_0.8.2        grid_3.6.3           fastmatch_1.1-0
##  [82] tools_3.6.3          rio_0.5.16           rstudioapi_0.11
##  [85] foreign_0.8-75       prodlim_2019.11.13   scatterplot3d_0.3-41
##  [88] digest_0.6.25        pracma_2.2.9         lava_1.6.7
##  [91] proto_1.0.0          Rcpp_1.0.5           broom_0.7.0
##  [94] gender_0.5.4         httr_1.4.2           readbitmap_0.1.5
##  [97] kernlab_0.9-29       colorspace_1.4-1     rvest_0.3.6
## [100] XML_3.99-0.3         fs_1.5.0             reticulate_1.16
## [103] splines_3.6.3        RWekajars_3.9.3-2    xlsxjars_0.6.1
## [106] flexmix_2.3-17       xtable_1.8-4         gmp_0.6-0
## [109] jsonlite_1.7.1       timeDate_3043.102    flashClust_1.01-2
## [112] modeltools_0.2-23    ipred_0.9-9          R6_2.4.1
## [115] gsubfn_0.7           pillar_1.4.4         htmltools_0.5.0
```

```
## [118] glue_1.4.1           class_7.3-17         codetools_0.2-16
## [121] utf8_1.1.4           sqldf_0.4-11         curl_4.3
## [124] tfruns_1.4           gtools_3.8.2         zip_2.1.1
## [127] openxlsx_4.1.5       survival_3.2-3       textclean_0.9.3
## [130] rmarkdown_2.3        munsell_0.5.0        haven_2.3.1
## [133] reshape2_1.4.4       gtable_0.3.0
```

```
## Time difference of 5.792444 mins
```