

# Data 607 - Assignment 9

Shamecca Marshall

2023-11-05

## Assignment - Web APIs

The New York Times web site provides a rich set of APIs, as described here: <https://developer.nytimes.com/apis> You'll need to start by signing up for an API key. Your task is to choose one of the New York Times APIs, construct an interface in R to read in the JSON data, and transform it into an R DataFrame.

## Libraries

```
# load libraries
library("httr")
library("jsonlite")
library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library("kableExtra")

## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##   group_rows

library("stringr")
library("ggplot2")
library("tidyverse")

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v forcats   1.0.0      v readr     2.1.4
## v lubridate 1.9.3      v tibble   3.2.1
## v purrr     1.0.2      v tidyr    1.3.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()      masks stats::filter()
## x purrr::flatten()     masks jsonlite::flatten()
## x kableExtra::group_rows() masks dplyr::group_rows()
## x dplyr::lag()         masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

## Connecting API

I will be using the Top stories API, filtered for the following section: - Fashion

Let's first read in the data from the API:

```
# API Key
apikey <- "Jb1d6yqz14VulbGuWbH0BbCMSPjfowxg"

# Get the URL
theURL <- paste("https://api.nytimes.com/svc/topstories/v2/fashion.json?api-key=", apikey)

fashionstories <- GET(theURL)

# Get status code
fashionstories$status_code
```

```
## [1] 200
```

```
summary(fashionstories)
```

```
##           Length Class      Mode
## url           1 -none-   character
## status_code   1 -none-   numeric
## headers       22 insensitive list
## all_headers    1 -none-   list
## cookies        7 data.frame list
## content       68025 -none-   raw
## date           1 POSIXct   numeric
## times          6 -none-   numeric
## request        7 request   list
## handle         1 curl_handle externalptr
```

```
fashion_stories <- content(fashionstories, as = "text")
```

```
## No encoding supplied: defaulting to UTF-8.
```

Data Frame Conversion

```
fashion_stories2 <- fromJSON(fashion_stories, flatten = TRUE)
fashion_stories2 <- data.frame(fashion_stories2$results, stringsAsFactors = FALSE)
```

```
#Get column names
colnames(fashion_stories2)
```

```
## [1] "section"           "subsection"         "title"
## [4] "abstract"          "url"                "uri"
## [7] "byline"            "item_type"          "updated_date"
## [10] "created_date"      "published_date"     "material_type_facet"
## [13] "kicker"            "des_facet"          "org_facet"
## [16] "per_facet"         "geo_facet"          "multimedia"
```

```
## [19] "short_url"
#Rename columns
colnames(fashion_stories2) <- c("Section","Subsection", "Title", "Abstract", "URL", "URI", "Byline", "I

#Drop columns not needed
fashion_stories3 <- fashion_stories2[, -c(12:18)]
```

## Data Analysis

```
#Count of Section
Section <- fashion_stories3%>%
  group_by(Section)%>%
  summarise(num=n())%>%
  arrange(desc(num))
head(Section)
```

```
## # A tibble: 6 x 2
##   Section      num
##   <chr>      <int>
## 1 fashion     13
## 2 style       12
## 3 t-magazine   4
## 4 travel       2
## 5 arts         1
## 6 nyregion     1
```

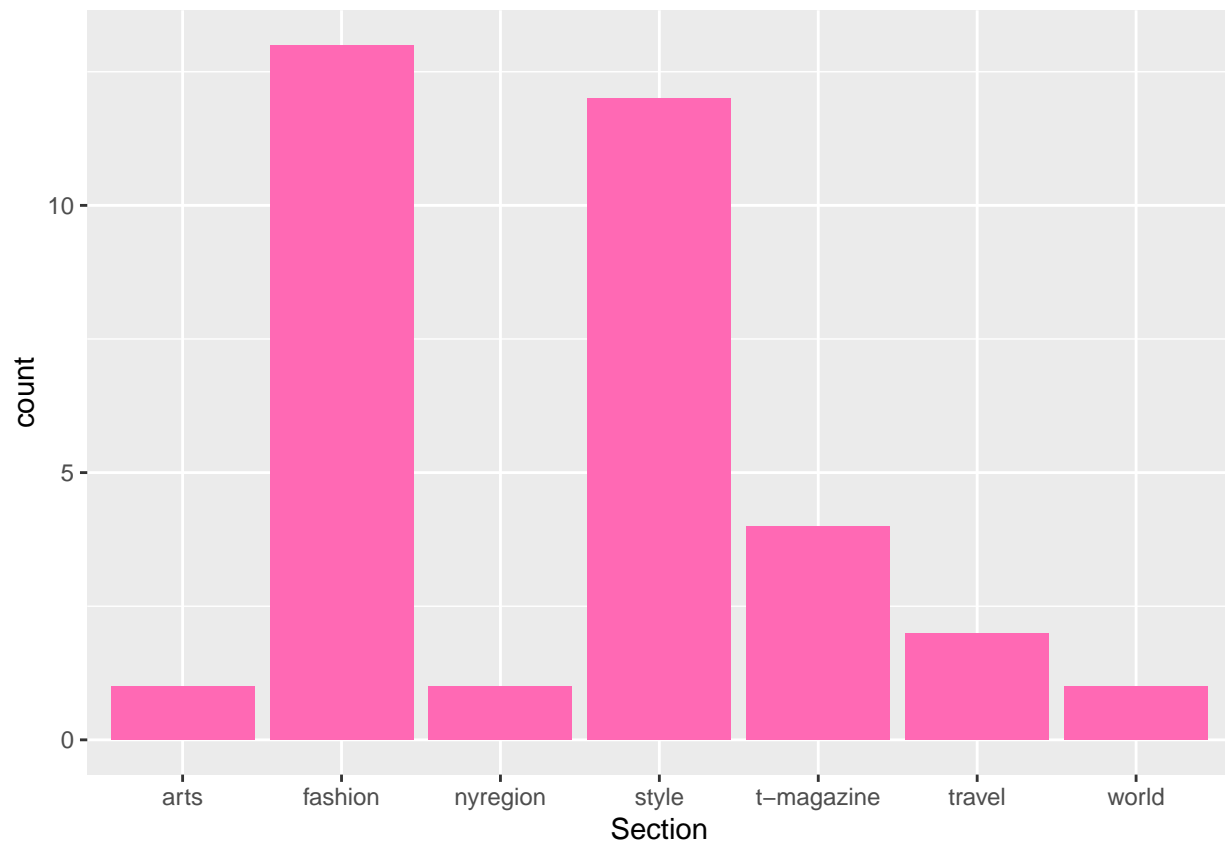
```
#Count of Subsection
Subsection <- fashion_stories3%>%
  group_by(Subsection)%>%
  summarise(num=n())%>%
  arrange(desc(num))
head(Subsection)
```

```
## # A tibble: 3 x 2
##   Subsection      num
##   <chr>          <int>
## 1 ""             32
## 2 "africa"        1
## 3 "design"         1
```

## Plots

The bar plot below shows that plenty of the articles belong to the fashion section followed by style and t-magazine.

```
# Bar Plot for Section
fashion_stories3 %>%
  ggplot(aes(x = Section)) +
  geom_bar(fill = "hotpink")
```



Within the next bar plot there is an unknown subsection that has the highest count followed by Africa and design.

```
# Bar Plot for Subsection  
fashion_stories3 %>%  
ggplot(aes(x = Subsection)) +  
  geom_bar(fill = "hotpink4")
```

