F179434

Usama Bin Haider

1/3/2022

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'purrr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## Warning: package 'forcats' was built under R version 4.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.1.2
```

```
list.files(path = "netflix_titles.csv")
```

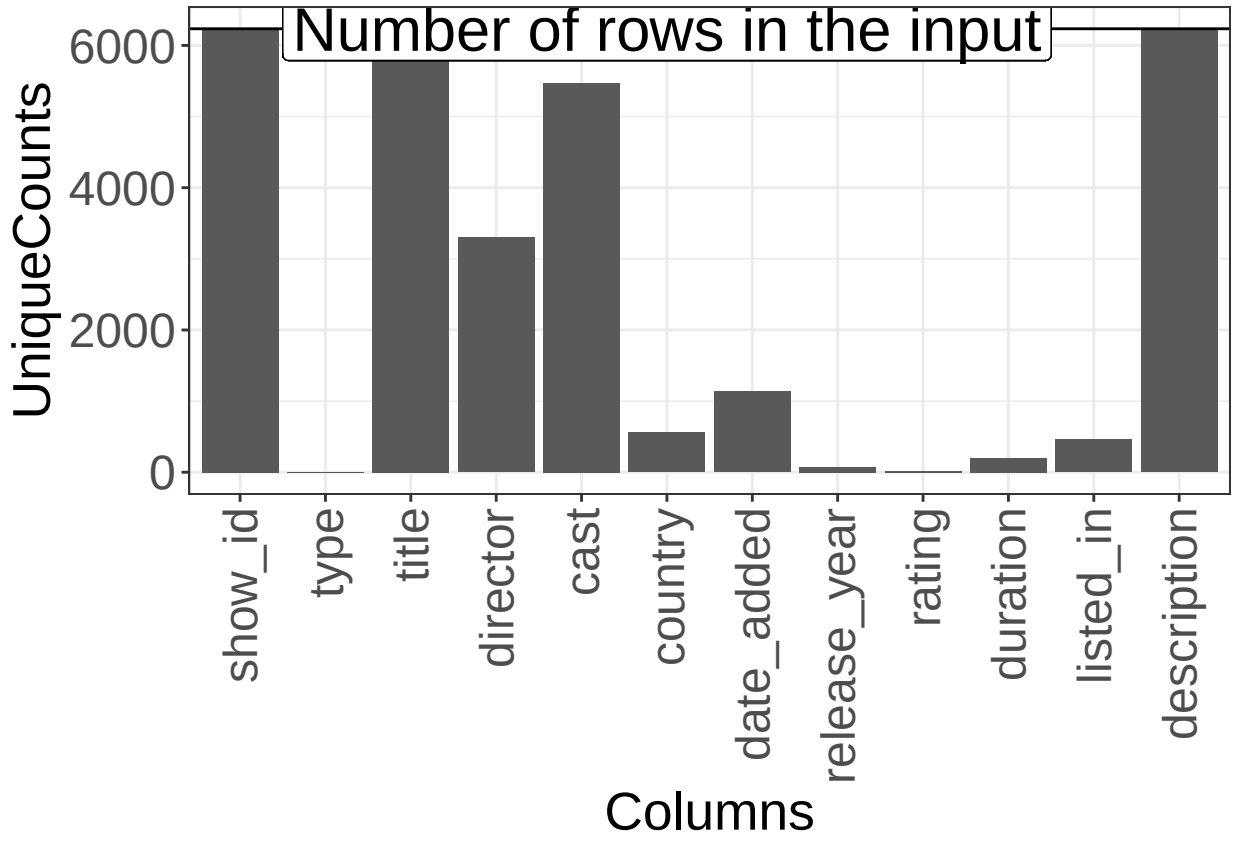
```
## character(0)
```

```
theme_custom_sk_90 <- theme_bw() +  theme(axis.text.x = element_text(size = 18, angle = 90, hjust = 1,
                                          axis.title = element_text(size = 20),strip.text  = element_te

# Change plot size to 8 x 3
options(repr.plot.width=12, repr.plot.height=8)


df_netflix <- read.csv("netflix_titles.csv")
df_netflix$date_added <- as.Date(df_netflix$date_added, format = "%B %d, %Y")
head(df_netflix)
```

```
##    show_id    type                                  title
## 1 81145628   Movie Norm of the North: King Sized Adventure
## 2 80117401   Movie              Jandino: Whatever it Takes
## 3 70234439 TV Show                      Transformers Prime
## 4 80058654 TV Show        Transformers: Robots in Disguise
## 5 80125979   Movie                             #realityhigh
## 6 80163890 TV Show                                  Apaches
##                   director
## 1 Richard Finn, Tim Maltby
## 2
## 3
## 4
## 5         Fernando Lebrija
## 6
##
## 1                                        Alan Marriott, Andrew Toth, Brian Dobson, Cole Howard, Jenni
## 2
## 3 Peter Cullen, Sumalee Montano, Frank Welker, Jeffrey Combs, Kevin Michael Richardson, Tania Gunadi
## 4                                                              Will Friedle, Darren Criss, Constance Zi
## 5           Nesta Cooper, Kate Walsh, John Michael Higgins, Keith Powers, Alicia Sanz, Jake Borelli,
```

```
## 6                                                                                                     Al
n, VerÃ³nica Echegui, LucÃa JimÃ©nez, Claudia Traisac
##                                    country date_added release_year   rating
## 1 United States, India, South Korea, China 2019-09-09         2019    TV-PG
## 2                           United Kingdom 2016-09-09         2016    TV-MA
## 3                            United States 2018-09-08         2013 TV-Y7-FV
## 4                            United States 2018-09-08         2016    TV-Y7
## 5                            United States 2017-09-08         2017    TV-14
## 6                                    Spain 2017-09-08         2016    TV-MA
##   duration                                                        listed_in
## 1   90 min                                Children & Family Movies, Comedies
## 2   94 min                                                  Stand-Up Comedy
## 3 1 Season                                                         Kids' TV
## 4 1 Season                                                         Kids' TV
## 5   99 min                                                         Comedies
## 6 1 Season Crime TV Shows, International TV Shows, Spanish-Language TV Shows
##
## 1         Before planning an awesome wedding for his grandfather, a polar bear king must take back a
## 2    Jandino Asporaat riffs on the challenges of raising kids and serenades the audience with a rous
## 3         With the help of three human allies, the Autobots once again protect Earth from the onslaug
## 4                       When a prison ship crash unleashes hundreds of Decepticons on Earth, Bumbleb
## 5 When nerdy high schooler Dani finally attracts the interest of her longtime crush, she lands in th
## 6            A young journalist is forced into a life of crime to save his father and family in this
```

```
unique_counts <- apply(df_netflix, MARGIN = 2, FUN = function(x) length(unique(x)))
unique_counts <- data.frame(Columns = names(unique_counts), UniqueCounts = unique_counts, stringsAsFact
unique_counts %>% ggplot(aes(x = Columns, y = UniqueCounts)) +
  geom_bar(stat = 'identity') +
  scale_x_discrete(limits = colnames(df_netflix)) +
  geom_hline(yintercept = nrow(df_netflix)) +
  geom_label(aes(x = 6, y = nrow(df_netflix), label = 'Number of rows in the input'), size = 8) +
  theme_custom_sk_90
```

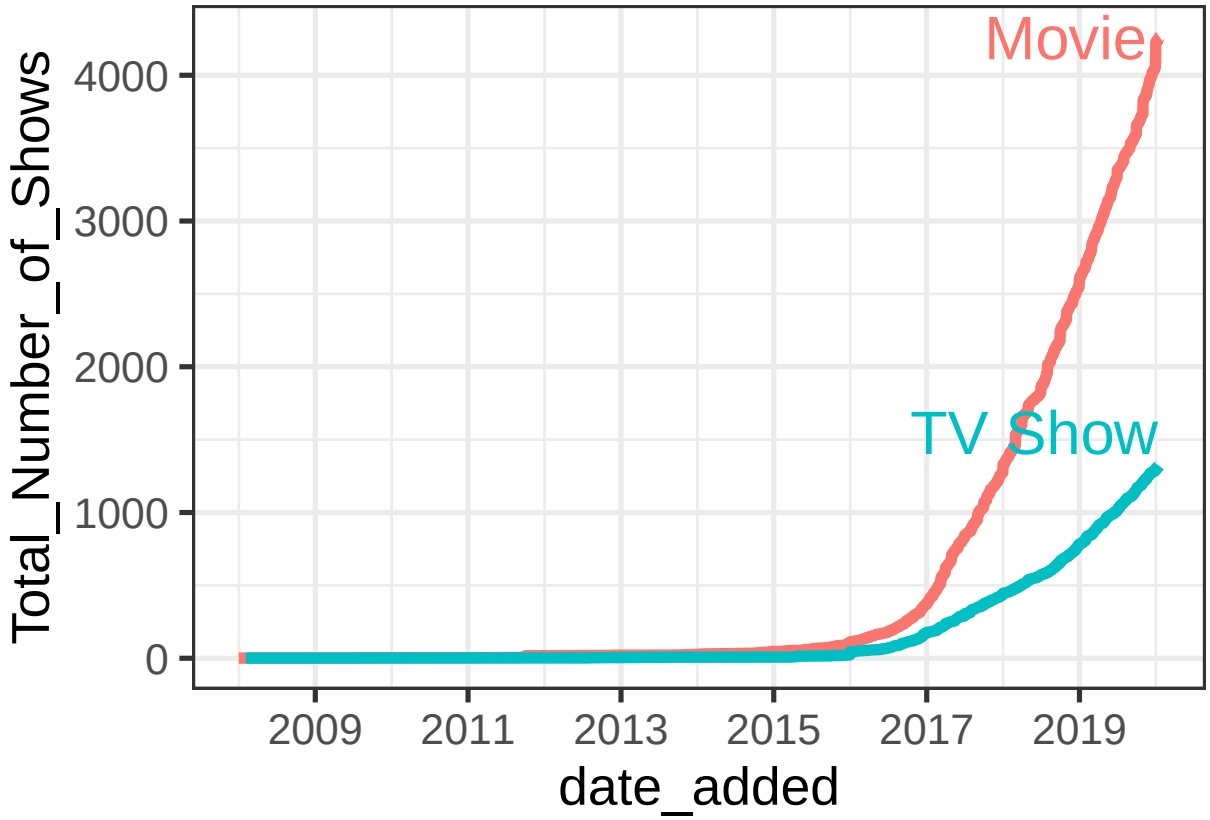



```
df_by_date <- df_netflix %>% group_by(date_added,type) %>% summarise(addedToday = n()) %>%
  ungroup() %>% group_by(type) %>% mutate(Total_Number_of_Shows = cumsum(addedToday), label = if_else(da
```

```
## `summarise()` has grouped output by 'date_added'. You can override using the `.groups` argument.
```

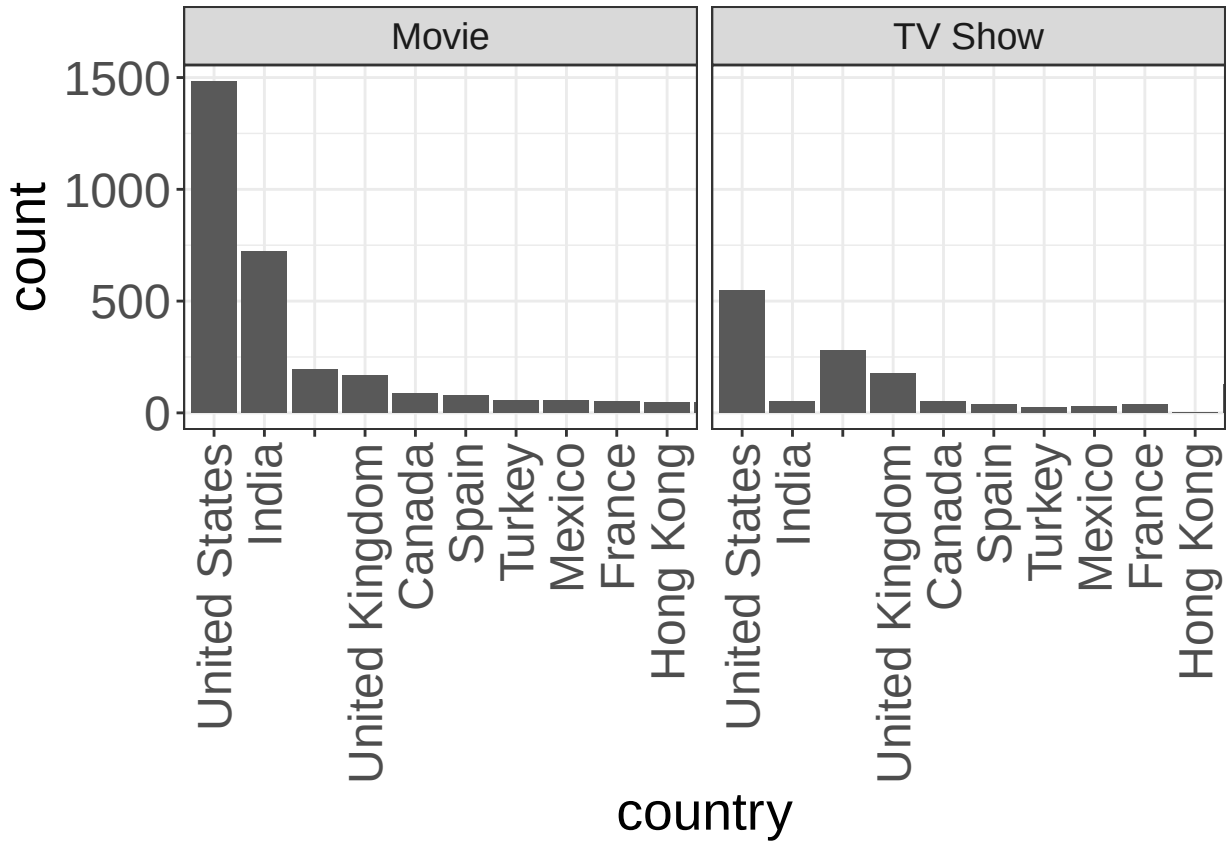
```
df_by_date  %>%
  ggplot(aes(x = date_added, y = Total_Number_of_Shows, color = type)) + geom_line(size = 2) +
  theme_bw(base_size = 20) +
  scale_x_date(date_breaks = '2 years', date_labels = "%Y") +
  theme(legend.position = 'none') +
  geom_text_repel(aes(label = label), size = 8,na.rm = TRUE, nudge_y = 100)
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```




```
df_netflix %>% group_by(type) %>% mutate(country = fct_infreq(country)) %>% ggplot(aes(x = country)) +
  geom_histogram(stat = 'count') + facet_wrap(~type, scales = 'free_x') +
  theme_custom_sk_90 + coord_cartesian(xlim = c(1,10)) + scale_x_discrete(labels = function(x){str_wrap
```

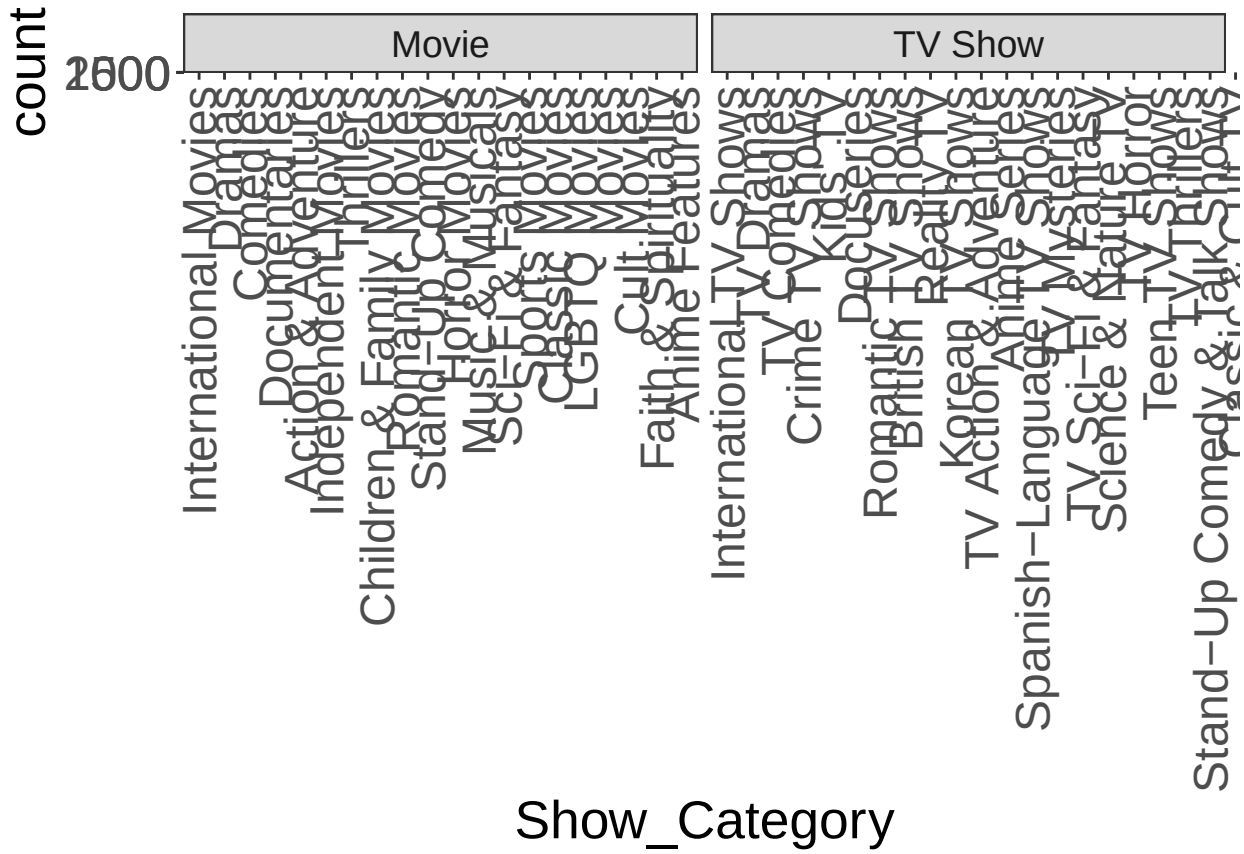
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
df_show_categories <- df_netflix %>%
  select(c('show_id','type','listed_in')) %>%
  separate_rows(listed_in, sep = ',') %>%
  rename(Show_Category = listed_in)
df_show_categories$Show_Category <- trimws(df_show_categories$Show_Category)
head(df_show_categories)
```

```
## # A tibble: 6 x 3
##    show_id type    Show_Category
##      <int> <chr>   <chr>
## 1 81145628 Movie   Children & Family Movies
## 2 81145628 Movie   Comedies
## 3 80117401 Movie   Stand-Up Comedy
## 4 70234439 TV Show Kids' TV
## 5 80058654 TV Show Kids' TV
## 6 80125979 Movie   Comedies
```

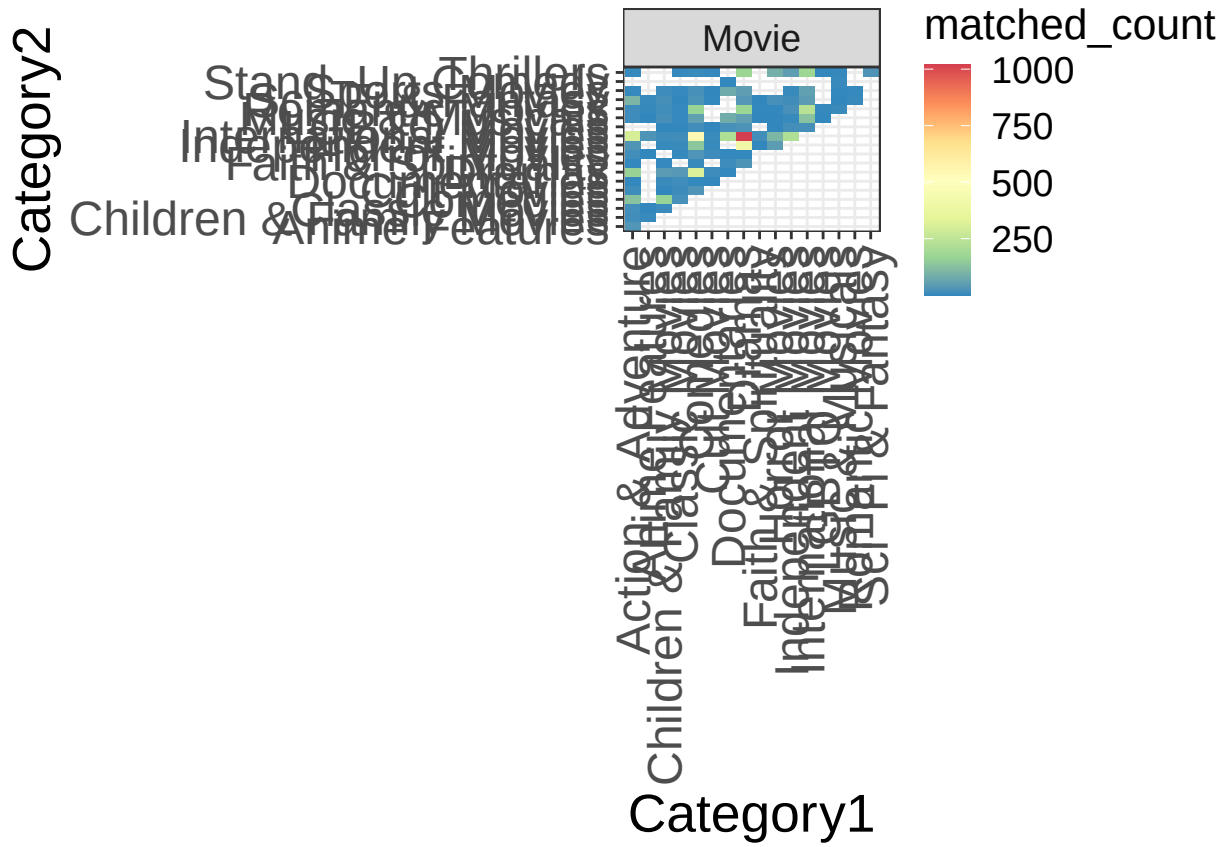
```
df_show_categories %>% mutate(Show_Category = fct_infreq(Show_Category)) %>%
  ggplot(aes(x = Show_Category)) +
  geom_bar() + scale_x_discrete() + facet_wrap(~type, scales = 'free_x') +
  theme_custom_sk_90 + theme() + coord_cartesian(xlim = c(1,20))
```



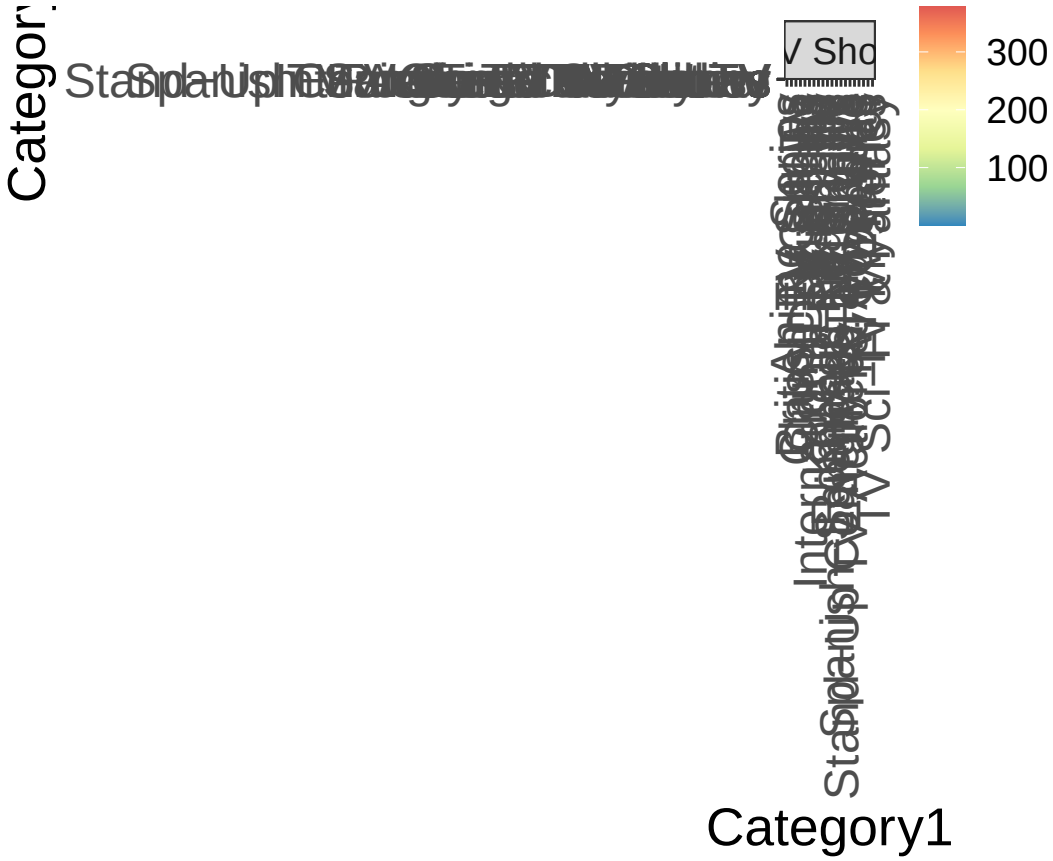

```
df_unique_categories <- df_show_categories %>% group_by(type,Show_Category) %>%  summarise()
```

```
## `summarise()` has grouped output by 'type'. You can override using the `.groups` argument.
```

```
df_category_correlations_movies <- data.frame(expand_grid(type = 'Movie',
                                                          Category1 = subset(df_unique_categories, type
                                                          Category2 = subset(df_unique_categories, type

df_category_correlations_TV <-      data.frame(expand_grid(type = 'TV Show',
                                                           Category1 = subset(df_unique_categories, typ
                                                           Category2 = subset(df_unique_categories, typ

df_category_correlations <- rbind(df_category_correlations_movies,df_category_correlations_TV)
df_category_correlations$matched_count <- apply(df_category_correlations, MARGIN = 1,FUN = function(x)
  length(intersect(subset(df_show_categories, type == x['type'] & Show_Category == x['Category1'])$show
                   subset(df_show_categories, type == x['type'] & Show_Category == x['Category2'])$show

df_category_correlations <- subset(df_category_correlations, (as.character(Category1) < as.character(Ca
# Change plot size to 8 x 3
options(repr.plot.width=14, repr.plot.height=10)

ggplot(subset(df_category_correlations, type == 'Movie'), aes(x = Category1, y = Category2, fill = matc
  geom_tile() + facet_wrap( ~type, scales = 'free') + theme_custom_sk_90 + scale_fill_distiller(palette
  theme(legend.text = element_text(size = 14), legend.title = element_text(size = 16))
```




```
ggplot(subset(df_category_correlations, type == 'TV Show'), aes(x = Category1, y = Category2, fill = ma
  geom_tile() + facet_wrap( ~type, scales = 'free') + theme_custom_sk_90 + scale_fill_distiller(palette
  theme(legend.text = element_text(size = 14), legend.title = element_text(size = 16))
```



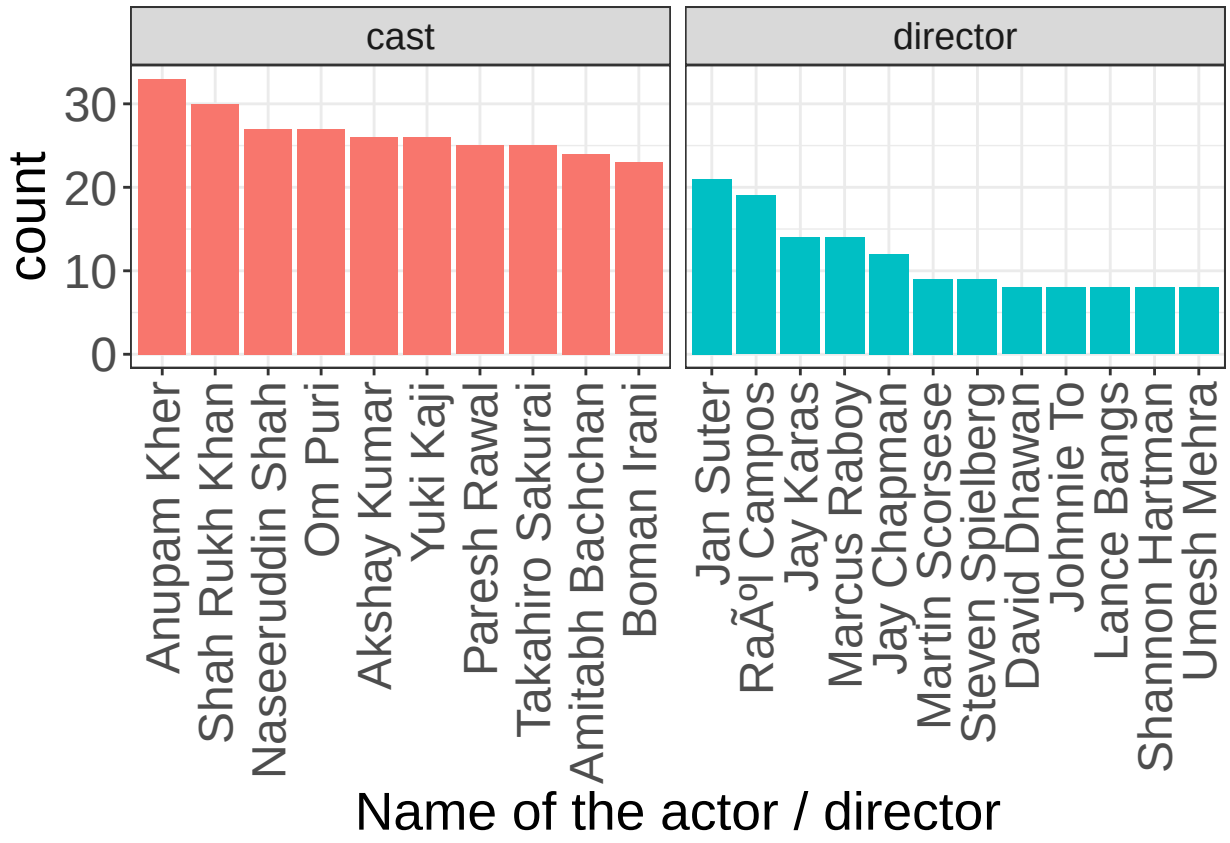

```
df_netflix %>% select(c('show_id','cast','director')) %>%
  gather(key = 'role', value = 'person', cast, director) %>%
  filter(person != "") %>% separate_rows(person, sep = ',') -> df_show_people

df_show_people$person <- trimws(df_show_people$person)
head(df_show_people)
```

```
## # A tibble: 6 x 3
##    show_id role  person
##      <int> <chr> <chr>
## 1 81145628 cast  Alan Marriott
## 2 81145628 cast  Andrew Toth
## 3 81145628 cast  Brian Dobson
## 4 81145628 cast  Cole Howard
## 5 81145628 cast  Jennifer Cameron
## 6 81145628 cast  Jonathan Holmes
```

```
df_people_freq<- df_show_people %>% group_by(person,role) %>%
  summarise(count = n()) %>% arrange(desc(count))
```

```
## `summarise()` has grouped output by 'person'. You can override using the `.groups` argument.
```

```
df_people_freq %>% group_by(role) %>% top_n(10,count) %>% ungroup() %>% ggplot(aes(x = fct_reorder(perso
  geom_bar(stat = 'identity') + scale_x_discrete() + facet_wrap(~role, scales = 'free_x') +
  theme_custom_sk_90 + theme(legend.position = 'none') + labs(x = 'Name of the actor / director')
```

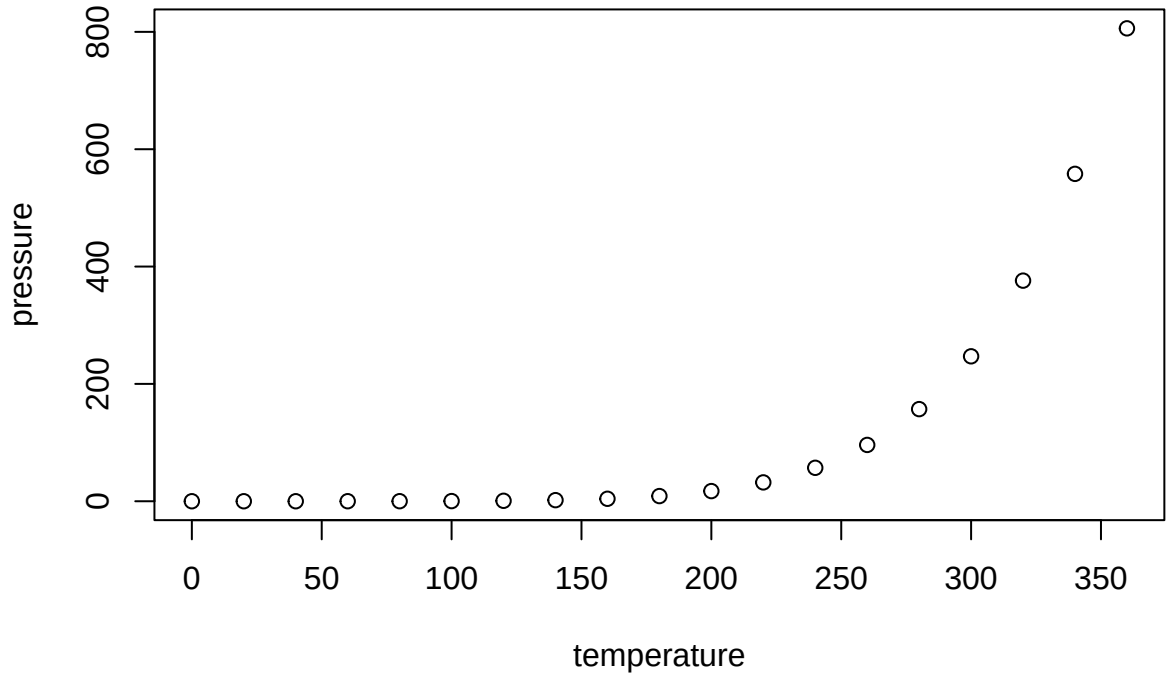



```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.