

We’are just going to admit it: We *like the* [*bachelor series!*](https://abc.go.com/shows/the-bachelorette)I won’t apologize for it. And sometimes I like to have a little fun by creating weird, nerd fabulous graphs which are about silly things. This is one of those times. In this blog post, I try to identify once and for all which was the most dramatic season of the bachelor series. I will accomplish this by analyzing the relative popularity of the bachelor franchise seasons as measured by their google search traffic.

I will be using the [gtrendsR package](https://cran.r-project.org/web/packages/gtrendsR/gtrendsR.pdf" \t "_blank) to gather Google trend information, dplyr to format the data, ggplot2 to create the graphs, gganimate to create a graph animation and ggimage to create custom lollipop charts.

**Load all of the packages and install if necessary**

In my [latest blog post](https://www.littlemissdata.com/blog/iconmap), someone kindly suggested that I do an auto check to install all necessary packages before loading them. After a quick search, I found this code below to efficiently install and load packages in [Vikram Baliga’s Blog](https://www.vikram-baliga.com/blog/2015/7/19/a-hassle-free-way-to-verify-that-r-packages-are-installed-and-loaded).

#specify the packages of interest

packages = c("gtrendsR","tidyverse","gifski", "gganimate", "ggimage", "lubridate", "usethis")

#use this function to check if each package is on the local machine

#if a package is installed, it will be loaded

#if any are not, the missing package(s) will be installed and loaded

package.check <- lapply(packages, FUN = function(x) {

if (!require(x, character.only = TRUE)) {

install.packages(x, dependencies = TRUE)

library(x, character.only = TRUE)

}

})

**Set the color variables**

To ensure consistent and effective color formatting, I am setting the color variables up front.

pink <- "#FF8DC6"

blue <- "#56C1FF"

yellow <- "#FAE232"

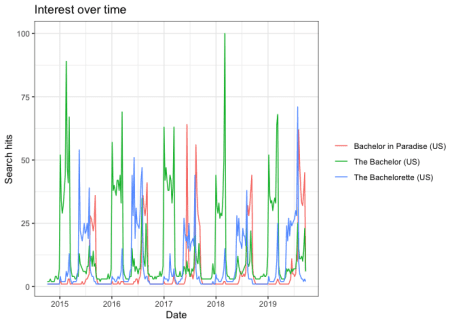
**Gather the ratings data**

Using the gtrendsR package, load weekly US ratings for “Bachelor in Paradise”, “The Bachelor” and “The Bachelorette”. The hits are calculated with a max of 100 to show the relative max hits across the time range and search subjects.

Plot the trends with the plot function.

bachTrends <- gtrends(c("Bachelor in Paradise", "The Bachelor", "The Bachelorette"), geo ="US")

plot(bachTrends)



**Transform the data**

Perform a little data transformation to get the trend information in a more desirable state. Filter to data that is from 2017 or later, then convert the hits field to numeric as the default is a character.

bachTrendsInterest <- bachTrends$interest\_over\_time

trends <- bachTrendsInterest %>%

filter(year(date)>2016) %>%

mutate(date = ymd(date),

hits = as.numeric(hits))

**Create the same plot with ggplot2**

Create the basic plot of relative search popularity by search criteria with the ggplot2 package. I am transitioning to ggplot2 in order to use the extensive ggplot2 features and complimentary packages like ggimage and gganimate.

#Frequency plot by keyword

p <- ggplot() +

geom\_line(data=trends, aes(x=date, y=hits, group=keyword, color = keyword)) +

scale\_color\_manual(values=c( yellow, blue, pink)) +

theme\_classic() +

theme(legend.position="bottom") +

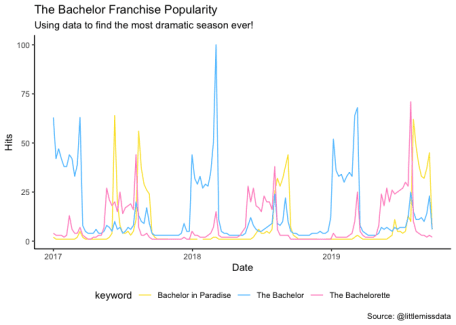
labs(title = "The Bachelor Franchise Popularity ",

subtitle = "Using data to find the most dramatic season ever!",

caption = "Source: @littlemissdata",

x = "Date", y = "Hits")

p



**Create an animation**

Take the basic plot created above and make an animation with it using the gganimate package. Thanks to [Sam Hunley](https://twitter.com/shunley42) for [sharing code and encouraging me to try gganimate!](https://twitter.com/shunley42/status/1174299457422340097)

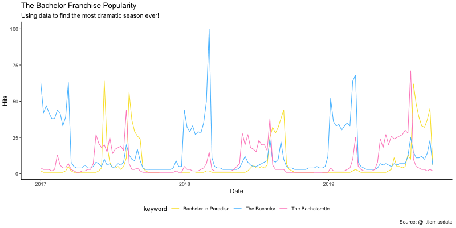
t <- p +

transition\_reveal(as.numeric(date))

gif <- animate(t, end\_pause = 25, width = 800, height = 400, fps = 8)

gif

anim\_save("Bachelor trends", gif)



**Bring in meta data about bachelor franchise shows**

We are going to bring in a data set which has the start dates for every single season of the bachelor franchise. We will then do some data munging to find the closest ratings date to the season start date. With this info we will join the bachelor season metadata to the ratings table.

## Add lollipops

x <-read.csv("/Users/lgellis/Desktop/Files/Cloud/littlemissdata/gtrends/bachelorListing.csv", stringsAsFactors = FALSE)

# Turn the dates into proper dates.

#Ratings are only tracked on sundays so get the closest Sunday for ratings

x <-x %>%

mutate(startDate = ymd(as.Date(startDate, "%m/%d/%y")),

endDate = ymd(as.Date(endDate, "%m/%d/%y")),

ratingStartDate = floor\_date(startDate, "weeks"),

ratingEndDate = floor\_date(endDate, "weeks"))

x

#Ratings are typically highest at the beginning

x<-left\_join(x, trends, by = c("topic"= "keyword", "ratingEndDate"="date"))

**Get the images for each of the seasons**

I have some plans to layer on a lollipop graph with the image of the bachelor season and display it at the height of the number of hits for the last day of the season. As such, I need to assign an image to every single season.

x <-x %>%

mutate(Image = case\_when(season == "Nick Viall" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/Nick.png>",

season == "Arie Luyendyk Jr" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/Arie.png>",

season == "Colton Underwood" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/Colton.png>",

season == "Rachel Lindsay" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/Rachel.png>",

season == "Becca Kufrin" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/Becca.png>",

season == "Hannah Brown" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/Hannah.png>",

topic == "Bachelor in Paradise" ~ "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/gtrendsR/images/BIP.png>"))

**Create the combined chart**

Create a fun graph to display the relative ratings for each season by layering on a lollipop chart. The lollipop chart will represent the seasons and their relative search popularity for the last week of the season. Use the geom\_segment function to set the lollipop stem and the geom\_image function to set the lollipop circle with the image representing the season.

p <- ggplot() +

geom\_line(data=trends, aes(x=date, y=hits, group=keyword, color = keyword), size=1) +

scale\_color\_manual(values=c(yellow, blue, pink)) +

geom\_segment(data=x, aes(x=ratingEndDate,

xend=ratingEndDate,

y=0,

yend=hits,

color=topic), size=1) +

geom\_image(data=x, aes(x=ratingEndDate, y=hits, image=Image), size=0.105) +

theme\_classic() +

labs(title = "The Bachelor Franchise Popularity ",

subtitle = "Using data to find the most dramatic season ever!",

caption = "Source: @littlemissdata",

x = "Date", y = "Hits") +

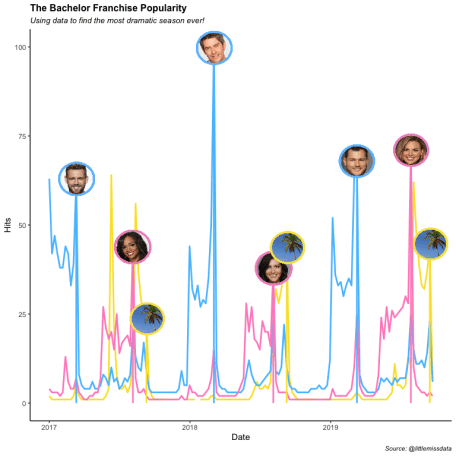
theme(legend.position="none",

plot.title = element\_text(size = 12, face = "bold"),

plot.subtitle = element\_text(size=10, face = "italic"),

plot.caption = element\_text(size = 8, face = "italic") )

p



**The Results Are in!**

Thank you for following along on this bachelor investigation with me.  Looking at the graph above, I think we can safely conclude that [Arie Luyendyk Jr’s](https://en.wikipedia.org/wiki/Arie_Luyendyk_Jr.) season was the most dramatic yet!

Please comment below if you enjoyed this blog, have questions, or would like to see something different in the future.  Note that the full code is available on my [github repo.](https://github.com/lgellis/MiscTutorial/tree/master/gtrendsR)

If you have trouble downloading the files or cloning the repo from github, please go to the [main page of the repo](https://github.com/lgellis/MiscTutorial) and select “Clone or Download” and then “Download Zip”. Alternatively or you can execute the following R commands to download the whole repo through R

use\_course("<https://github.com/lgellis/MiscTutorial/archive/master.zip>")