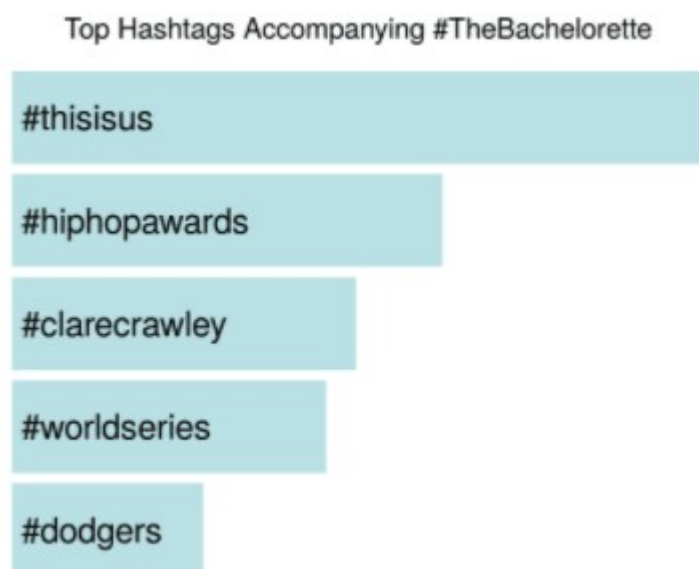
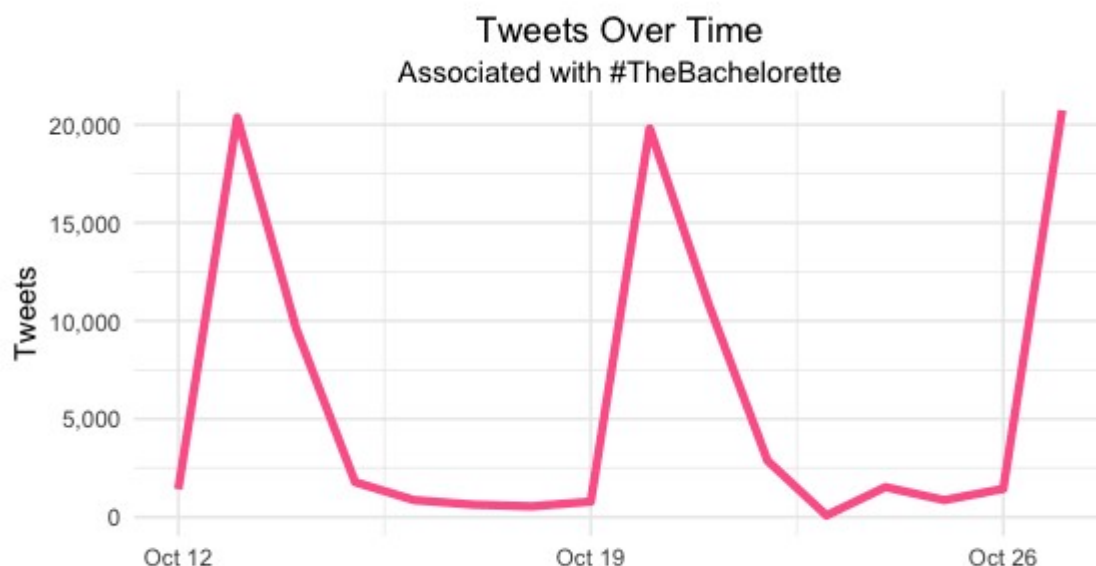


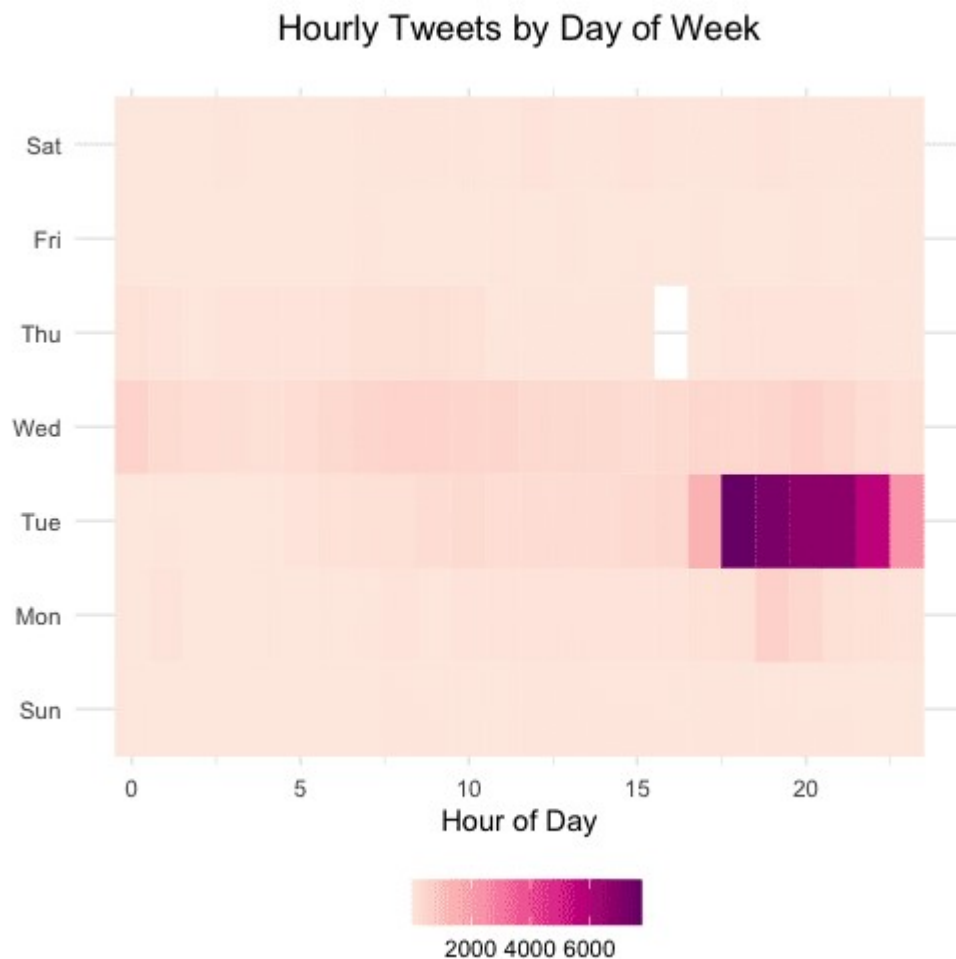
Those who were looking for entertainment last night may not have been satisfied if they decided to watch The Bachelorette. Upon analyzing Twitter data, it is clear that there were conflicts of interest amongst TV watchers in the US. The top hashtags trending with #TheBachelorette reflect the notion that people weren't necessarily tuned in last night. The top hashtags mentioned were: This is Us, Hip Hop Awards, and The World Series.



Despite the fact there were more interesting things on TV, we still saw tremendous Bachelorette activity on Twitter. In analyzing the data, it became clear that no one cares about the Bachelorette unless the episode is on the air. On most days, there are less than 200 tweets per hour tagged with #TheBachelorette and that increases to roughly 20,000 tweets per hour when the show is on.



That plot leads nicely into visualizing the discrepancy between the time the show is aired vs. all other times. As you'll notice, the time leading up to the show sees the most activity, and then it slows down until the end of the show. There is another interesting thing that stands out (ever so slightly). Monday evenings indicate some build up for the show and then people recap on Wednesdays.

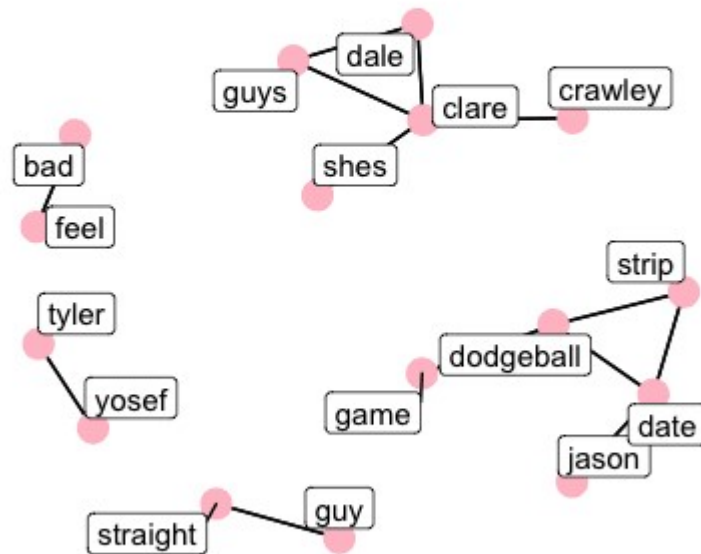


You may be asking, “if this has been such a boring season, why are people talking about it on Twitter?” This is a completely reasonable question. If we analyze the text and find the most highly correlated words, we realize that it’s all about Dale from South Dakota (and sometimes Clare). If you were to only see this one chart, it would tell you a lot:

1. Dale is the center of attention
2. Dale gets the most time with Clare
3. There was a strip dodgeball game
4. Yosef and Tyler are either best buds or worst enemies
5. People “feel bad” either watching this show or for certain contestants

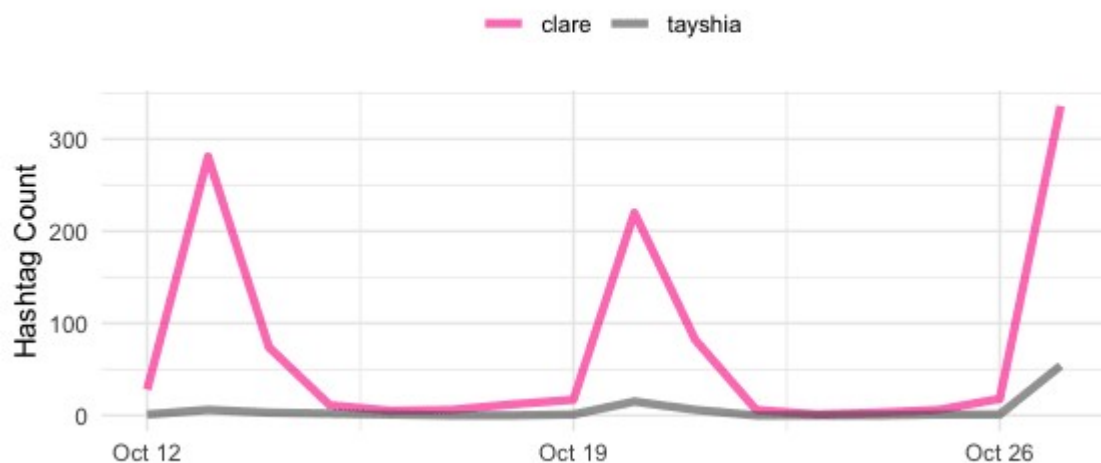
## Twitter Talk

What's the conversation?

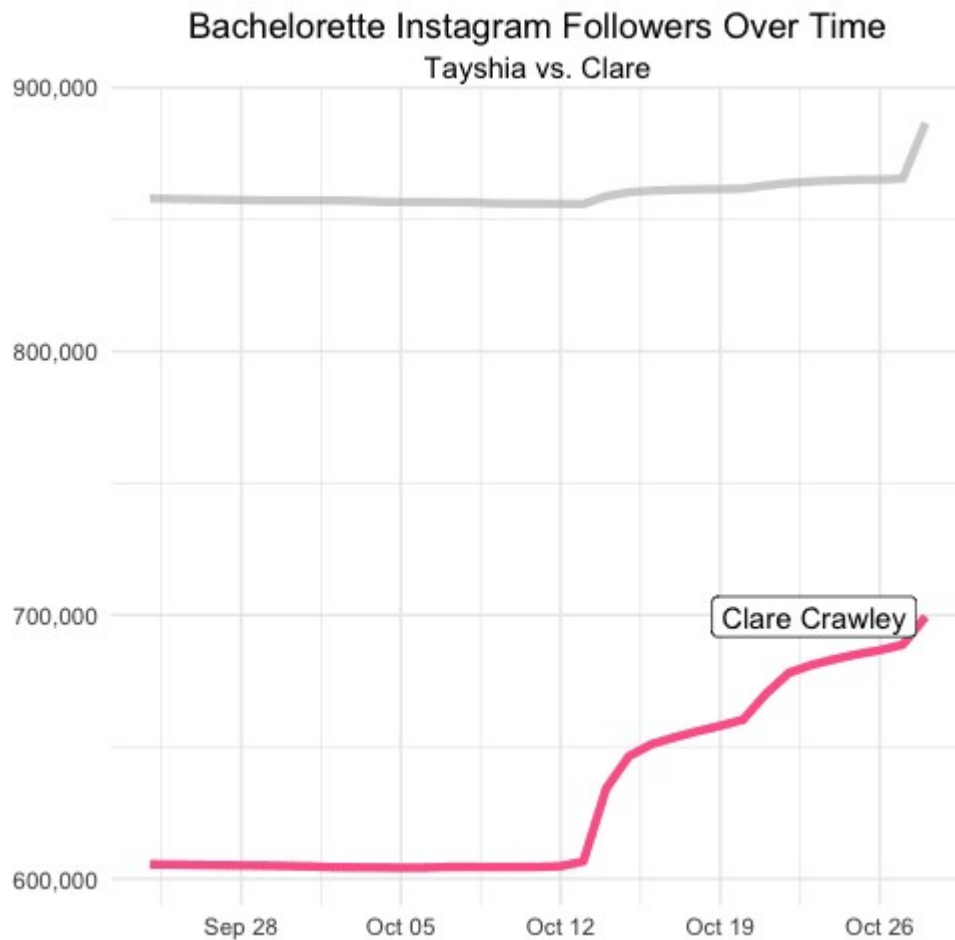


ABC has made it clear that Tayshia is going to replace Clare as The Bachelorette. This less than juicy tidbit was not only leaked, but immediately confirmed by ABC. Returning to Twitter, we can see that Tayshia is slowly entering the conversation. She has started appearing in more and more Tweets as time passes.

## Hourly Tweets Over Time Containing Bachelorette Name in Hashtag



The race to the top (bottom?) of the influencer food chain is on. Clare (pink line) has made significant follower gains since the show began. She increased her Instagram followers from roughly 600K to 700K. You'll also note that Tayshia (gray line) gained followers over that time. However, she had about 850K followers to begin with and increased sharply at the end of last night's episode.



As always, please feel free to play with the data yourself at <https://stoltzmaniac.shinyapps.io/TheBacheloretteApp/> where you can take advantage of some fancy algorithms to determine the emotions of the faces in each post the contestant made public on Instagram.

We'll be doing some analysis after next week's show, hope to see you then. The code for the plots is below and the data is available upon request by using our contact page.

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(stringr)
library(lubridate)

GLOBAL_DATA = get_database_data()

# Tweets over Time
GLOBAL_DATA$tweet_hashtags_raw %>%
  as_tibble() %>%
  filter(datetime >= '2020-10-12') %>%
  filter(datetime <= '2020-10-28') %>%
  mutate(date = as_date(datetime)) %>%
  group_by(date) %>%
  summarize(tweet_count = sum(tweet_count)) %>%
  ggplot(aes(x = date, y = tweet_count)) +
  geom_line(col = "#ff6699", size = 1.5) +
```

```

theme_minimal() +
scale_y_continuous(label = scales::comma) +
xlab('') +
ylab('Hashtag Count') +
ggtitle("Hourly Tweets Over Time", subtitle = "Associated with
#TheBachelorette") +
theme(legend.position = 'top', legend.direction = "horizontal",
      legend.title = element_blank(),
      plot.title = element_text( hjust = 0.5, vjust = -1),
      plot.subtitle = element_text( hjust = 0.5, vjust = -1))

```

```

GLOBAL_DATA$tweet_text_raw %>%
  as_tibble() %>%
  filter(group >= '2020-10-12') %>%
  filter(group <= '2020-10-28') %>%
  mutate(day_of_week = wday(group, label = TRUE, abbr = TRUE),
         hour_of_day = hour(group)) %>%
  group_by(day_of_week, hour_of_day) %>%
  summarize(tweet_count = n()) %>%
  ggplot(aes(x = hour_of_day, y = day_of_week, col = tweet_count, fill =
tweet_count)) +
  geom_tile() +
  scale_fill_distiller(palette = 'RdPu', direction = 1) +
  scale_color_distiller(palette = 'RdPu', direction = 1) +
  xlab('Hour of Day') +
  ylab('') +
  labs(fill = "Tweets", col = "Tweets") +
  ggtitle("Hourly Tweets by Day of Week", subtitle = "") +
  theme_minimal() +
  theme(legend.position = 'bottom', legend.direction = "horizontal",
        legend.title = element_blank(),
        plot.title = element_text( hjust = 0.5, vjust = -1),
        plot.subtitle = element_text( hjust = 0.5, vjust = -1))

```

```

GLOBAL_DATA$tweet_hashtags_raw %>%
  as_tibble() %>%
  filter(datetime >= '2020-10-12') %>%
  filter(datetime <= '2020-10-28') %>%
  mutate(lower_text = tolower(hashtag),
         bachelor = as.integer(str_detect(lower_text, 'bachelor')),
         tayshia = as.integer(str_detect(lower_text, 'tayshia')),
         clare = as.integer(str_detect(lower_text, 'clare')),
         tayshia_clare = tayshia + clare) %>%
  filter(!bachelor, tayshia_clare > 0) %>%
  select(datetime, tayshia, clare) %>%

```

```

  pivot_longer(cols = c(tayshia, clare), names_to = 'hashtag', values_to =
'tweet_count') %>%
  #mutate(date = floor_date(datetime, unit = 'hours')) %>%
  mutate(date = as_date(datetime)) %>%
  group_by(hashtag, date) %>%
  summarize(tweet_count = sum(tweet_count), .groups = 'drop') %>%
  ggplot(aes(x = date, y = tweet_count, col = hashtag, fill = hashtag)) +
  geom_line(size = 1.5) +
  #geom_col(position = 'dodge') +
  scale_fill_manual(values = alpha(c("#FF1493", "#5a5a5a"), .6)) +
  scale_color_manual(values = alpha(c("#FF1493", "#5a5a5a"), .6)) +
  theme_minimal() +
  scale_y_continuous(label = scales::comma) +
  xlab('') +
  ylab('Hashtag Count') +
  ggtitle("Hourly Tweets Over Time", subtitle = "Containing Bachelorette
Name in Hashtag") +
  theme(legend.position = 'top', legend.direction = "horizontal",
        legend.title = element_blank(),
        plot.title = element_text( hjust = 0.5, vjust = -1),
        plot.subtitle = element_text( hjust = 0.5, vjust = -1))

```

```

GLOBAL_DATA$insta_followers %>%
  filter(suitor %in% c('tayshiaaa', 'clarecrawley')) %>%
  drop_na() %>%
  ggplot(aes(x = datetime, y = follower_count, group = name)) +
  geom_line(col = '#ff6699', fill = '#ff6699', size = 1.5) +
  gghighlight::gghighlight(name == 'Clare Crawley', label_key = name,
use_group_by = TRUE) +
  theme_minimal() +
  labs(x = '', y = '', title = "Bachelorette Instagram Followers Over Time",
subtitle = "Tayshia vs. Clare") +
  scale_y_continuous(label = scales::comma) +
  theme(legend.position = 'top', legend.direction = "horizontal",
        legend.title = element_blank(),
        plot.title = element_text( hjust = 0.5, vjust = -1),
        plot.subtitle = element_text( hjust = 0.5, vjust = -1))

```

```

twitter_data <- format_tweets(GLOBAL_DATA$tweet_text_raw %>%
dplyr::rename(created_at = group, clean_text = group_series_0, y = value))

twitter_correlations = twitter_correlations(twitter_data, min_usage = 1000)

twitter_correlations %>%
  influential::graph_from_data_frame() %>%

```

```

    ggraph::ggraph(layout = "fr") +
    ggraph::geom_edge_link(ggplot2::aes(edge_alpha = correlation),
show.legend = FALSE) +
    ggraph::geom_node_point(color = "pink", size = 5) +
    ggraph::geom_node_text(ggplot2::aes(label = name), repel = TRUE, size =
6) +
    ggplot2::theme_void()

```

```

clean_the_text = function(text){
  text = str_replace_all(text, 'biden', '')
  text = str_replace_all(text, 'trump', '')
  text = str_replace_all(text, 'pratt', '')
  text = str_replace_all(text, 'chris', '')
  text = str_replace_all(text, 'ruffalo', '')
  text = str_replace_all(text, 'twitch', '')
  text = str_replace_all(text, 'tory', '')
  text = str_replace_all(text, 'bachelorette', '')
  text = str_replace_all(text, 'bachelor', '')
  text = str_replace_all(text, 'mark', '')
  text = str_replace_all(text, 'chris', '')
  text = str_replace_all(text, 'harrison', '')
  return(text)
}

```

```

twitter_data <- format_tweets(GLOBAL_DATA$tweet_text_raw %>%
                                rename(created_at = group, clean_text =
group_series_0, y = value) %>%
                                mutate(clean_text =
clean_the_text(clean_text)))

```

```

twitter_correlations = twitter_correlations(twitter_data, min_usage = 1000)

twitter_correlations %>%
  filter(correlation > abs(0.061)) %>%
  filter(!item1 %in% c('bachelorette', 'season', 'clares', 'im', 'tonight',
'watching', 'episode', 'bachelorette', 'tonight', 'watch')) %>%
  filter(!item2 %in% c('bachelorette', 'season', 'clares', 'im', 'tonight',
'watching', 'episode', 'bachelorette', 'tonight', 'watch')) %>%
  #influential::graph_from_data_frame() %>%
  ggraph::ggraph(layout = "igraph", algorithm = 'kk') +
  ggraph::geom_edge_link() +
  ggraph::geom_node_point(color = "pink", size = 5) +
  ggraph::geom_node_label(ggplot2::aes(label = name), repel = TRUE, size =
4) +
  ggraph::theme_graph() +
  ggtitle("Twitter Talk", subtitle = "What's the conversation?") +
  theme(legend.position = 'top', legend.direction = "horizontal",
        legend.title = element_blank(),

```

```
plot.title = element_text( hjust = 0.5, vjust = -1),  
plot.subtitle = element_text( hjust = 0.5, vjust = -1))
```

...