# Welcome to Nightvale Text Analysis

William Lovejoy

7/12/2022

As practice in scraping multiple webpages, and in analyzing multiple related texts, I decided to collect and analyze all transcripts for the “Welcome to Nightvale” podcast. The transcripts are available through their main site: link.

```
library(rvest)
library(tidyverse)
library(tidytext)
library(gridExtra)
library(cowplot)
```

As always, we load in our libraries. We’ll be using rvest for scraping, and the others for analysis and visualization.

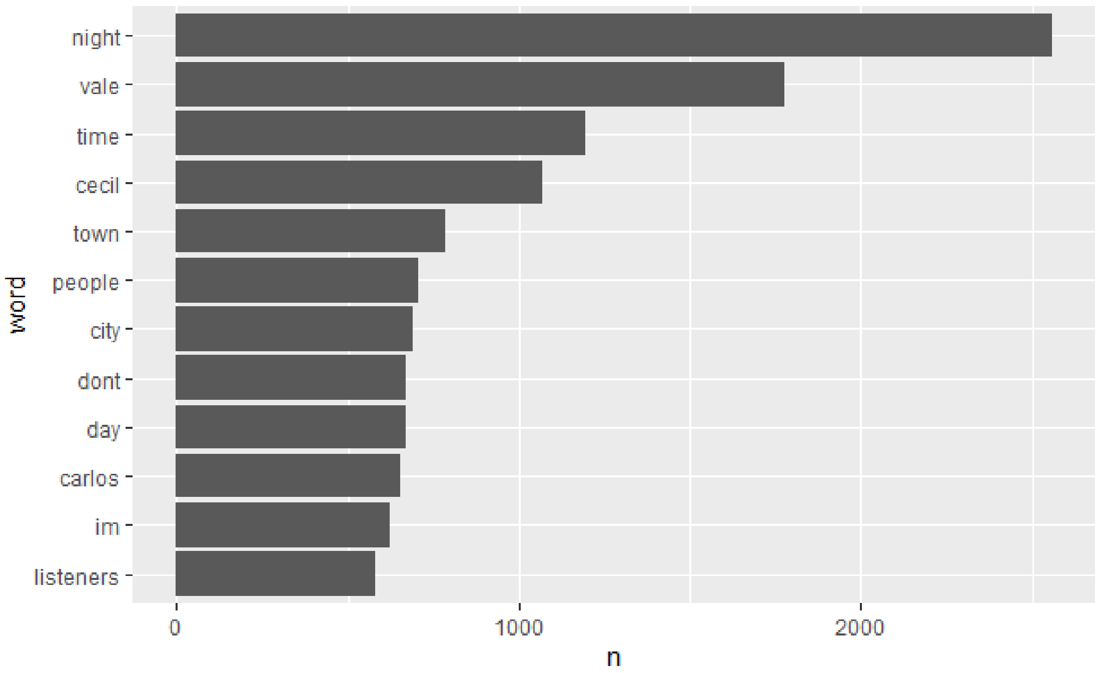
## Scraping Data

```
counter = 2012
new_urls = "http://www.nightvalepresents.com/welcome-to-night-vale-transcript
s?year=%s"
i = 1
w = 1
all_frames <- list()

while(counter < 2023){
  home <- read_html(sprintf(new_urls, counter))
  links <- home %>%
    html_nodes(".entry-title") %>%
    html_nodes("a") %>%
    html_attr("href") %>%
    xml2::url_absolute("https://www.nightvalepresents.com/welcome-to-night-va
le-transcripts")
  links <- rev(links)
  for(x in 1:length(links)){
    script <- read_html(links[x]) %>%
      html_nodes("p+ p") %>%
      html_text()
    df <- data.frame(Episode = i,
                     Relative_Episode = w,
                      Year = counter,
                      Transcript = seq(1:length(script)))
    df$Transcript <- script
    all_frames[[i]] <- df
    i = i+1
    w = w + 1
  }
  counter = counter + 1
  w = 1
}

nightvale <- bind_rows(all_frames)
```

This section loads in the base url for the podcast transcripts. However, this page itself doesn’t have the transcripts itself. Instead it contains links to individual pages, with one transcript per page. We use a while loop to go through the page for a given year and save the links into a list. It then uses a for loop to pass through that list, taking the year produced, the base episode number, the relative episode (starting from 1 each new year), and the transcript for the episode. It’s important to note that the transcripts are only available in paragraphs, so each row is a new paragraph. Once a dataframe is created, it’s added to a list. After exiting the loop, we can bind all the dataframes together into one large one to work on.

## Tokens

```
tidy_vale<- nightvale %>%
  unnest_tokens(word, Transcript) %>%
  mutate(word = gsub("’", "", word)) %>%
  anti_join(stop_words)

tidy_vale <- subset(tidy_vale, !grepl("[^a-zA-Z]", word))

tidy_vale %>%
  count(word, sort = TRUE) %>%
  filter(n > 500) %>%
  mutate(word = reorder(word, n))%>%
  ggplot(aes(x = n, y = word)) + geom_col()
```



```r
annual_bigrams <- nightvale %>%
  group_by(Year) %>%
  unnest_tokens(bigrams, Transcript, token = "ngrams", n = 2) %>%
  separate(bigrams, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word,
         is.na(word1) == FALSE,
         is.na(word2) == FALSE,
         !grepl("[^a-zA-Z]", word1),
         !grepl("[^a-zA-Z]", word1)) %>%
  count(word1, word2, sort = TRUE) %>%
  ungroup() %>%
  unite(bigrams, word1, word2, sep = " ")
```

Throughout the entire podcast, the two most important words are “night” and “vale”. Which makes sense, as they’re in the name of the show. Other key words in this list are “cecil” (which is the name of the narrator), “carlos” (who is a love interest for the narrator), and “listeners” (because the podcast is done as a radio show broadcast in the small southwestern town of Nightvale).

```r
nightvale_bigrams <- nightvale %>%
  unnest_tokens(bigrams, Transcript, token = "ngrams", n = 2) %>%
  separate(bigrams, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word,
         is.na(word1) == FALSE,
         is.na(word2) == FALSE,
         !grepl("[^a-zA-Z]", word1),
         !grepl("[^a-zA-Z]", word1)) %>%
  count(word1, word2, sort = TRUE) %>%
  unite(bigrams, word1, word2, sep = " ")

all_bg <- nightvale_bigrams %>%
  filter(n > 75) %>%
  ggplot(aes(x = n, y = reorder(bigrams, n))) + geom_col() +
  labs(title = "Bigram Counts in All Episodes", x = "n", y = "Bigram")
```

While our individual tokens gave us a glimpse at the more common topics, using bigrams helps us get a more specific idea of what the texts talk about. “night vale” is still the most common, but that’s not particularly surprising. Of more significant note are the phrases “city council”, “secret police”, and “dog park”. While the city council and Sheriff’s secret police are long term topics of the show, the dog park is less of one. It was part of the pilot episode, with the first mention reading:

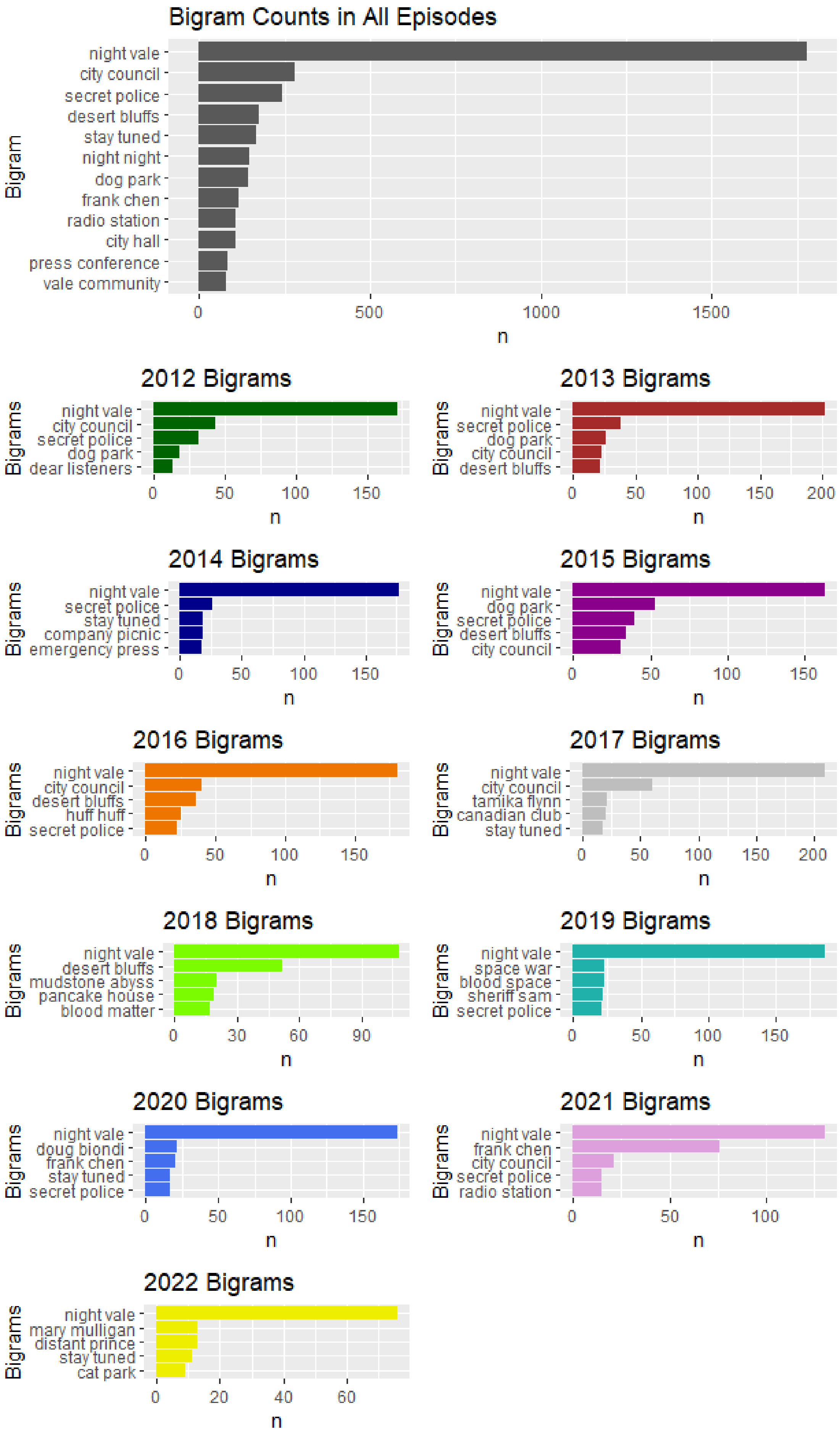
> The City Council announces the opening of a new dog park at the corner of Earl and Summerset, near the Ralphs. They would like to remind everyone that dogs are not allowed in the dog park. People are not allowed in the dog park. It is possible you will see hooded figures in the dog park. Do not approach them. Do not approach the dog park. The fence is electrified and highly dangerous. Try not to look at the dog park and especially do not look for any period of time at the hooded figures. The dog park will not harm you.

After which the narrator described it as a mysterious place where interns vanished, the mangled remains of prehistoric creatures could be found near, and that may or may not emit a static-y hum that is actually a coded message from an “unearthly voice” urging citizens to bring precious metals and toddlers to the dog park.

```r
nightvale_2012 <- annual_bigrams %>%
  filter(Year == "2012")

graph2012 <- nightvale_2012 %>%
  head(5)  %>%
  ggplot(aes(x= n, y = reorder(bigrams,n))) + geom_col(fill = "darkgreen") +
  labs(title = "2012 Bigrams", y = "Bigrams")
```

In order to graph the most important bigrams for each year of the podcast, we need to filter out each year, sort it, take the top handful (5 in this case), and create a saved graph. I did this for all 11 years of the show, then used grid.arrange to make the following plot.

```r
bigrams_by_year <- plot_grid(graph2012, graph2013, graph2014, graph2015, grap
h2016, graph2017,
             graph2018, graph2019, graph2020, graph2021, graph2022,
             nrow = 6, ncol = 2)

plot_grid(all_bg, bigrams_by_year, nrow = 2, rel_heights = c(1/4, 3/4))
```



Bigram Counts in All Episodes

2012 Bigrams

2013 Bigrams

2014 Bigrams

2015 Bigrams

2016 Bigrams

2017 Bigrams

2018 Bigrams

2019 Bigrams

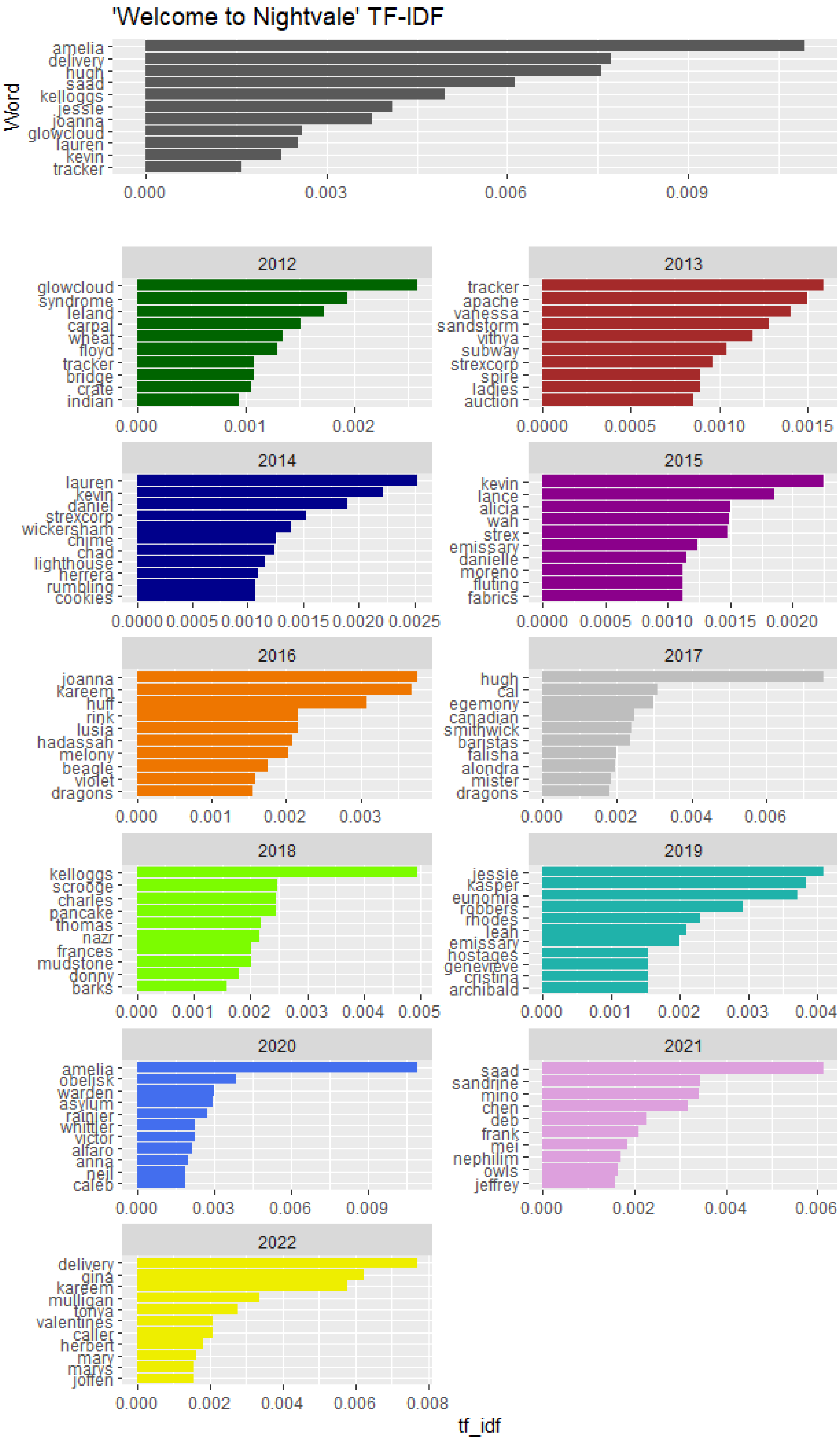
2020 Bigrams

2021 Bigrams

2022 Bigrams

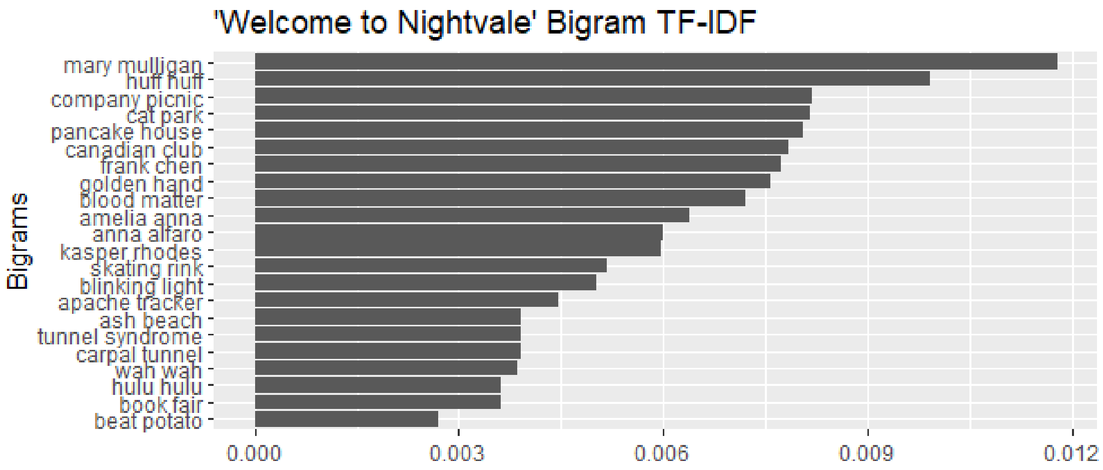
From this we can see that “Night vale” is the most common bigram every year, but that “dog park” only breaks into the top 5 in 2012, 2013, and 2015. We can also see the rise and fall of terms such as “Desert Bluffs”, the adjacent town to Nightvale which is owned and operated by “StrexCorp Synernists Incorporated”. It makes the top 5 in 2013, 2015, and 2016, but never after that. This is because in episode 83, aired in 2016, Desert Bluffs was forcibly annexed into Nightvale.

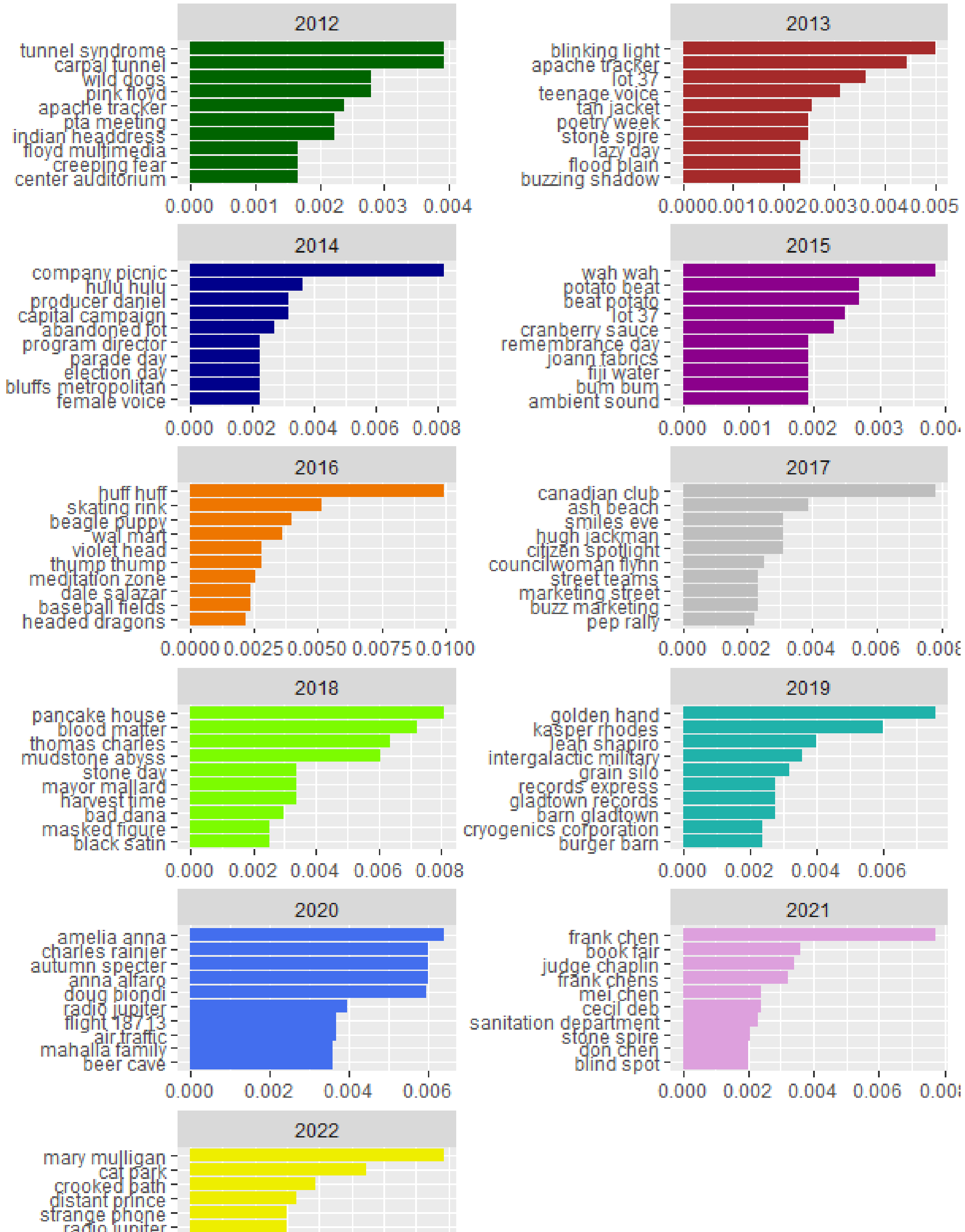
```r
nightvale_tf <- nightvale %>%
  unnest_tokens(word, Transcript) %>%
  mutate(word = gsub("’", "", word)) %>%
  anti_join(stop_words) %>%
  add_count(Year, name = "Total_Words") %>%
  group_by(Year, Total_Words) %>%
  count(word, sort = TRUE)

nightvale_tf <- subset(nightvale_tf, !grepl("[^a-zA-Z]", word))


nightvale_tf_idf <- nightvale_tf %>%
  bind_tf_idf(word, Year, n) %>%
  arrange(desc(tf_idf))

yearly_tf <- nightvale_tf_idf %>%
  group_by(Year) %>%
  top_n(10) %>%
  ungroup() %>%
  facet_bar(y = word, x = tf_idf, by = Year, nrow = 6)

all_tf <- nightvale_tf_idf %>%
  slice_max(11, with_ties = FALSE) %>%
  ggplot(aes(x = tf_idf, y = reorder(word, tf_idf))) + geom_col() +
  labs(title = "'Welcome to Nightvale' TF-IDF",x = "", y = "Word")

plot_grid(all_tf, yearly_tf, nrow = 2, rel_heights = c(1/6, 5/6))
```


'Welcome to Nightvale' TF-IDF
Word
amelia
delivery
hugh
saad
kelloggs
jessie
joanna
glowcloud
lauren
kevin
tracker
0.000
0.003
0.006
0.009
2012
glowcloud
syndrome
leland
carpal
wheat
floyd
tracker
bridge
crate
indian
0.000
0.001
0.002
2013
tracker
apache
vanessa
sandstorm
vithya
subway
strexcorp
spire
ladies
auction
0.0000
0.0005
0.0010
0.0015
2014
lauren
kevin
daniel
strexcorp
wickersham
chime
chad
lighthouse
herrera
rumbling
cookies
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
2015
kevin
lance
alicia
wah
strex
emissary
danielle
moreno
fluting
fabrics
0.0000
0.0005
0.0010
0.0015
0.0020
2016
joanna
kareem
huff
rink
lusia
hadassah
melony
beagle
violet
dragons
0.000
0.001
0.002
0.003
2017
hugh
cal
egemony
canadian
smithwick
baristas
falisha
alondra
mister
dragons
0.000
0.002
0.004
0.006
2018
kelloggs
scrooge
charles
pancake
thomas
nazr
frances
mudstone
donny
barks
0.000
0.001
0.002
0.003
0.004
0.005
2019
jessie
kasper
eunomia
robbers
rhodes
leah
emissary
hostages
genevieve
cristina
archibald
0.000
0.001
0.002
0.003
0.004
2020
amelia
obelisk
warden
asylum
rainier
whittier
victor
alfaro
anna
neil
caleb
0.000
0.003
0.006
0.009
2021
saad
sandrine
mino
chen
deb
frank
mei
nephilim
owls
jeffrey
0.000
0.002
0.004
0.006
2022
delivery
gina
kareem
mulligan
tonya
valentines
caller
herbert
mary
marys
joffen
0.000
0.002
0.004
0.006
0.008
tf_idf

While simple counts are helpful, the term frequency-inverse document frequency is a better measure of how important a word is to a document or collection. This lets us see that in 2012, “Amelia” was the most relevant word overall in the entire collection, but that she was only the most relevant in 2020 when looking at each year.

```r
nightvale_bigrams_tf <- nightvale %>%
  unnest_tokens(bigrams, Transcript, token = "ngrams", n = 2) %>%
  separate(bigrams, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word,
         is.na(word1) == FALSE,
         is.na(word2) == FALSE,
         !grepl("[^a-zA-Z]", word1),
         !grepl("[^a-zA-Z]", word1)) %>%
  unite(bigrams, word1, word2, sep = " ") %>%
  add_count(Year, name = "Total_Words") %>%
  group_by(Year, Total_Words) %>%
  count(bigrams, sort = TRUE)

nightvale_bigrams_tf<- nightvale_bigrams_tf %>%
  bind_tf_idf(bigrams, Year, n) %>%
  arrange(desc(tf_idf))

all_bgs_tf <- nightvale_bigrams_tf %>%
  slice_max(tf_idf, n= 2, with_ties = FALSE) %>%
  ggplot(aes(x = tf_idf, y = reorder(bigrams, tf_idf))) + geom_col() +
  labs(title = "'Welcome to Nightvale' Bigram TF-IDF", x = "",
       y = "Bigrams")

yearly_bgs_tf <- nightvale_bigrams_tf %>%
  group_by(Year) %>%
  slice_max(tf_idf, n= 10, with_ties = FALSE) %>%
  ungroup() %>%
  facet_bar(y = bigrams, x = tf_idf, by = Year, nrow = 6)

plot_grid(all_bgs_tf, yearly_bgs_tf, nrow = 2, rel_heights = c(1/4, 3/4))
```

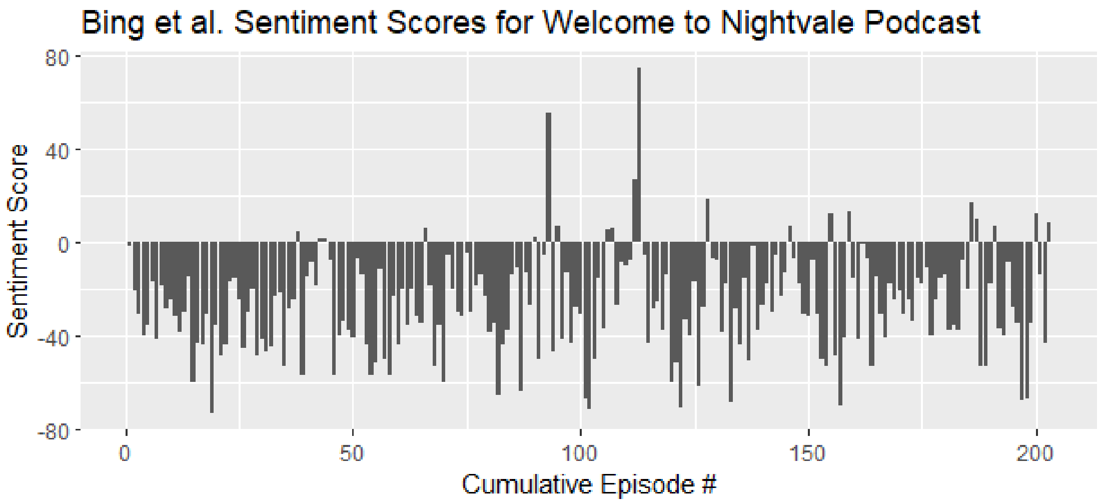

'Welcome to Nightvale' Bigram TF-IDF
Bigrams
mary mulligan
huff huff
company picnic
cat park
pancake house
canadian club
frank chen
golden hand
blood matter
amelia anna
anna alfaro
kasper rhodes
skating rink
blinking light
apache tracker
ash beach
tunnel syndrome
carpal tunnel
wah wah
hulu hulu
book fair
beat potato
0.000
0.003
0.006
0.009
0.012

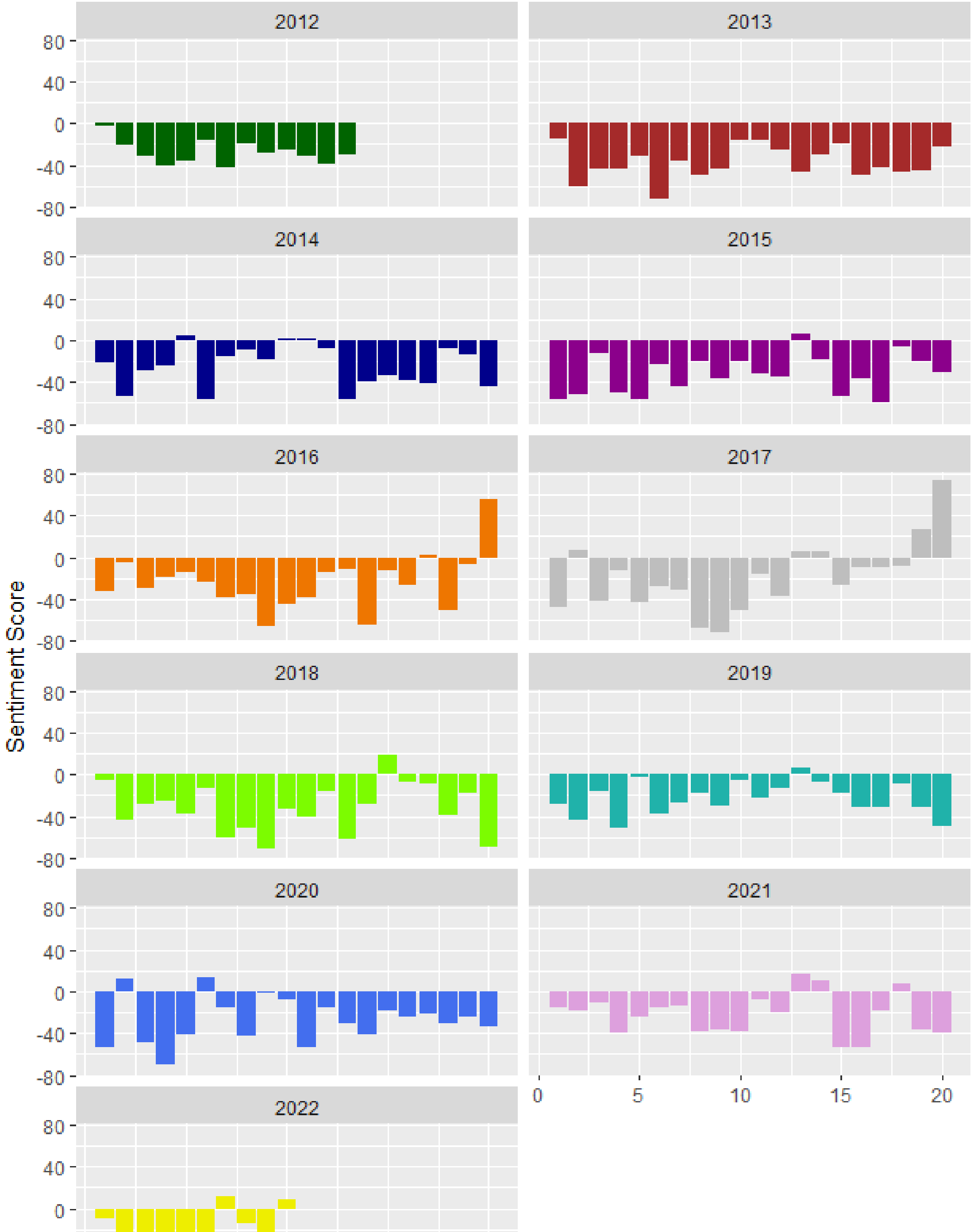

2012
tunnel syndrome
carpal tunnel
wild dogs
pink floyd
apache tracker
pta meeting
indian headdress
floyd multimedia
creeping fear
center auditorium
0.000 0.001 0.002 0.003 0.004
2013
blinking light
apache tracker
lot 37
teenage voice
tan jacket
poetry week
stone spire
lazy day
flood plain
buzzing shadow
0.0000.0010.0020.0030.0040.005
2014
company picnic
hulu hulu
producer daniel
capital campaign
abandoned lot
program director
parade day
election day
bluffs metropolitan
female voice
0.000 0.002 0.004 0.006 0.008
2015
wah wah
potato beat
beat potato
lot 37
cranberry sauce
remembrance day
joann fabrics
fiji water
bum bum
ambient sound
0.000 0.001 0.002 0.003 0.00
2016
huff huff
skating rink
beagle puppy
wal mart
violet head
thump thump
meditation zone
dale salazar
baseball fields
headed dragons
0.0000 0.0025 0.0050 0.0075 0.0100
2017
canadian club
ash beach
smiles eve
hugh jackman
citizen spotlight
councilwoman flynn
street teams
marketing street
buzz marketing
pep rally
0.000 0.002 0.004 0.006 0.008
2018
pancake house
blood matter
thomas charles
mudstone abyss
stone day
mayor mallard
harvest time
bad dana
masked figure
black satin
0.000 0.002 0.004 0.006 0.008
2019
golden hand
kasper rhodes
jean shapiro
intergalactic military
grain silo
records express
gladtown records
barn gladtown
cryogenics corporation
burger barn
0.000 0.002 0.004 0.006
2020
amelia anna
charles rainjer
autumn specter
anna alfaro
doug biondi
radio jupiter
flight 18713
air traffic
mahalla family
beer cave
0.000 0.002 0.004 0.006
2021
frank chen
book fair
judge chaplin
frank chens
mei chen
cecil deb
sanitation department
stone spire
don chen
blind spot
0.000 0.002 0.004 0.006 0.008
2022
mary mulligan
cat park
crooked path
distant prince
strange phone
radio jupiter

We can get a more detailed view by looking at bigrams, and by grouping our data by year. This let's us see how various characters are.

## Sentiments

```
tidy_vale_sentiments <- tidy_vale %>%
  inner_join(get_sentiments("bing")) %>%
  count(index = Episode, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)

all_s <- tidy_vale_sentiments %>%
  ggplot(aes(x = index, y = sentiment)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Bing et al. Sentiment Scores for Welcome to Nightvale Podcast
",
       x = "Cumulative Episode #", y = "Sentiment Score")


annual_sentiments <- tidy_vale %>%
  group_by(Year) %>%
  inner_join(get_sentiments("bing")) %>%
  count(index = Relative_Episode, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)

year_s <- annual_sentiments %>%
  ggplot(aes(x = index, y = sentiment, fill = factor(Year), group = Year)) +
  geom_col(show.legend = FALSE) +
  scale_fill_manual(values = colors) +
  labs(x = "Episode #", y = "Sentiment Score") +
  facet_wrap(~Year, nrow = 6)

plot_grid(all_s, year_s, nrow = 2,  rel_heights = c(1/4, 3/4))
```


Bing et al. Sentiment Scores for Welcome to Nightvale Podcast



We can see here that the overall sentiment of the show is negative. It makes sense when looking at the overall themes and genres of the show though. The podcast is classified as horror fiction, meaning overall darker tones. It also edges into the genre of cosmic horror, which functions as a subgenere of horror fiction that empahsizes the horror of the incomprehensible and unknowable instead of gore or shock. Works in this genre tend to emphasize themes of cosmic dread, forbidden/dangerous knowledge, madness, and non human-influences on religion, superstition, humanity, and fate. All of these are topics that we affix a negative sentiment to, so it seems obvious that a radio show that uses all of these themes would contain words with a larger negative sentiment.