The task is achieved in a couple of steps.

1. Identify the file path of each blog post in my local blog folder
2. For each post, extract the date and title of the blog post from the front matter, as well as the name and links to the image files.
3. Combine the extracted information into a character vector that can be copy-pasted to a gallery page.

I should state a couple of caveats/notes for anyone looking to emulate my approach.

* I take advantage of the fact that I use the same prefix for (almost) all  
  visualization that I generate with R—viz, as well as the same file  
  format—.png.
* I create headers and links from the titles of the blog posts  
  (via something like sprintf('## %s, [%s](%s)', ...) and I order everything according  
  to descending date and ascending line in the post.  
  This may not be what you would like for your gallery format.

**The Code**

library(tidyverse)

paths\_post\_raw <-

fs::dir\_ls(

'content/post/',

regexp = 'index[.]md$',

recurse = TRUE

) %>%

# Ignore the "\_index.md" at the base of the content/post directory.

# Would need to also ignore draft posts if there are draft posts.

str\_subset('\_index', negate = TRUE)

paths\_post\_raw[1:10]

## [1] "content/post/analysis-texas-high-school-academics-1-intro/index.md"

## [2] "content/post/analysis-texas-high-school-academics-2-competitions/index.md"

## [3] "content/post/analysis-texas-high-school-academics-3-individuals/index.md"

## [4] "content/post/analysis-texas-high-school-academics-4-schools/index.md"

## [5] "content/post/analysis-texas-high-school-academics-5-miscellaneous/index.md"

## [6] "content/post/cheat-sheet-rmarkdown/index.md"

## [7] "content/post/data-science-podcasts/index.md"

## [8] "content/post/dry-principle-make-a-package/index.md"

## [9] "content/post/gallery-visualizations/old/index.md"

## [10] "content/post/interval-data-nycflights13/index.md"

# Define some important regular expressions (or "regex"es).

# These regexes are probably applicable to most Hugo/blogdown setups.

rgx\_replace <- '(content\\/post\\/)(.\*)(\\/)(.\*)([.]png$)'

rgx\_title <- '^title[:]\\s+'

rgx\_date <- 'date[:]\\s+'

# This regex is particular to the way that I name and save my ggplots.

rgx\_viz <- '(^[!][\\[][\\]].\*)(viz.\*png)(.\*$)'

# Define a helper function for a common idiom that we will implement for extracting the ines of markdown that we

# want---those containing the title, data, and visualization---and trimming them just to the text that we want

# (i.e. removing "title:" and "date:" preceding the title and date in the YAML/TOML header, and

# removing the "![]" preceding an image).

str\_pluck <- function(x, pattern, replacement = '') {

x %>%

str\_subset(pattern) %>%

str\_replace\_all(pattern = pattern, replacement = replacement) %>%

str\_trim()

}

str\_pluck\_title <- purrr::partial(str\_pluck, pattern = rgx\_title)

str\_pluck\_date <- purrr::partial(str\_pluck, pattern = rgx\_date)

str\_pluck\_viz <- purrr::partial(str\_pluck, pattern = rgx\_viz, replacement = '\\2')

# Extract the title, date, and visualizations from each post.

# Note that there should be only one title and date pers post, but there are likely more than one visualization per post.

paths\_post <-

paths\_post\_raw %>%

as.character() %>%

tibble(path\_post = .) %>%

mutate(

lines = path\_post %>% purrr::map(read\_lines)

) %>%

mutate\_at(vars(path\_post), ~str\_remove\_all(., 'content|\\/index[.]md')) %>%

mutate\_at(

vars(lines),

list(

title = ~purrr::map\_chr(., str\_pluck\_title),

date = ~purrr::map\_chr(., str\_pluck\_date) %>% lubridate::ymd(),

viz = ~purrr::map(., str\_pluck\_viz)

)

) %>%

# viz` is a list item (because there may be more than one per post), so we need to `unnest()` it to return a "tidy" data frame.

unnest(viz) %>%

select(date, viz, title, path\_post)

paths\_post

## # A tibble: 71 x 4

## date viz title path\_post

##

## 1 2018-05-20 viz\_map\_bycomp~ An Analysis of Texas H~ /post/analysis-texas~

## 2 2018-05-20 viz\_map\_bycomp~ An Analysis of Texas H~ /post/analysis-texas~

## 3 2018-05-20 viz\_n\_bycomplv~ An Analysis of Texas H~ /post/analysis-texas~

## 4 2018-05-20 viz\_n\_bycomp-1~ An Analysis of Texas H~ /post/analysis-texas~

## 5 2018-05-20 viz\_n\_bycompco~ An Analysis of Texas H~ /post/analysis-texas~

## 6 2018-05-20 viz\_n\_bycompco~ An Analysis of Texas H~ /post/analysis-texas~

## 7 2018-05-20 viz\_persons\_st~ An Analysis of Texas H~ /post/analysis-texas~

## 8 2018-05-20 viz\_persons\_st~ An Analysis of Texas H~ /post/analysis-texas~

## 9 2018-05-20 viz\_persons\_st~ An Analysis of Texas H~ /post/analysis-texas~

## 10 2018-05-20 viz\_persons\_st~ An Analysis of Texas H~ /post/analysis-texas~

## # ... with 61 more rows

# Create the markdown lines for images (visualizations) for our gallery markdown output.

paths\_post\_md <-

paths\_post %>%

mutate(

label\_md = sprintf('![%s](%s/%s)', viz, path\_post, viz)

) %>%

select(title, date, path\_post, label\_md)

paths\_post\_md

## # A tibble: 71 x 4

## title date path\_post label\_md

##

## 1 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_map\_bycomplvl\_~

## 2 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_map\_bycomplvl\_~

## 3 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_n\_bycomplvl-1.~

## 4 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_n\_bycomp-1.png~

## 5 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_n\_bycompcomplv~

## 6 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_n\_bycompcomplv~

## 7 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_persons\_stats\_~

## 8 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_persons\_stats\_~

## 9 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_persons\_stats\_~

## 10 An Analysis of Tex~ 2018-05-20 /post/analysis-tex~ ![viz\_persons\_stats\_~

## # ... with 61 more rows

# Create the "main" data frame with the titles and dates in columns alongside the image column.

# In "tidy data" terminology, images are the "observations" \*and `title` and `date` are separate variables).

content\_gallery\_raw <-

paths\_post\_md %>%

group\_by(title, date, path\_post) %>%

# Add a "placeholder" line for the title of the post.

do(add\_row(., .before = 0)) %>%

ungroup() %>%

# In the first case, create the H2 markkdown heading line for the name of the post.

# In the second case, use the image markdown line created above.

mutate\_at(

vars(label\_md),

~case\_when(

is.na(.) ~ sprintf('## %s, [%s](%s)', dplyr::lead(date), dplyr::lead(title), dplyr::lead(path\_post)),

TRUE ~ .

)

) %>%

# Impute the `title` and `date` values to go with the image values.

fill(title, .direction = 'up') %>%

fill(date, .direction = 'up') %>%

arrange(date) %>%

# Number the posts in order of descending date.

mutate(idx\_intragrp = dense\_rank(sprintf('%s, %s', date, title))) %>%

group\_by(title, date) %>%

# Number the images within each post. (This isn't completely necessary. It's only used for sorting.)

mutate(idx\_intergrp = row\_number()) %>%

ungroup() %>%

select(idx\_intragrp, idx\_intergrp, date, label\_md) %>%

arrange(desc(idx\_intragrp), idx\_intergrp)

content\_gallery\_raw

## # A tibble: 90 x 4

## idx\_intragrp idx\_intergrp date label\_md

##

## 1 19 1 2019-06-29 ## 2019-06-29, [Text Parsing and T~

## 2 19 2 2019-06-29 ![viz\_toc\_n\_1yr\_tree.png](/post/te~

## 3 19 3 2019-06-29 ![viz\_content\_section\_n.png](/post~

## 4 19 4 2019-06-29 ![viz\_toc\_content\_n1.png](/post/te~

## 5 19 5 2019-06-29 ![viz\_sents\_section\_n.png](/post/t~

## 6 19 6 2019-06-29 ![viz\_sents\_section\_n\_yr.png](/pos~

## 7 19 7 2019-06-29 ![viz\_sents\_section\_sim.png](/post~

## 8 19 8 2019-06-29 ![viz\_words\_section\_tfidf.png](/po~

## 9 19 9 2019-06-29 ![viz\_words\_tfidf.png](/post/text-~

## 10 18 1 2019-01-27 ## 2019-01-27, [Summarizing rstudi~

## # ... with 80 more rows

# Create the final markdown output.

content\_gallery <-

content\_gallery\_raw %>%

select(label\_md) %>%

mutate(idx = row\_number()) %>%

# Add a blank line between the end of one section's last image

# and the next sections H2 header.

group\_by(idx) %>%

do(add\_row(., label\_md = '', .before = 0)) %>%

ungroup()

content\_gallery

## # A tibble: 180 x 2

## label\_md idx

##

## 1 "" NA

## 2 ## 2019-06-29, [Text Parsing and Text Analysis of a Periodic Repo~ 1

## 3 "" NA

## 4 ![viz\_toc\_n\_1yr\_tree.png](/post/text-parsing-analysis-periodic-re~ 2

## 5 "" NA

## 6 ![viz\_content\_section\_n.png](/post/text-parsing-analysis-periodic~ 3

## 7 "" NA

## 8 ![viz\_toc\_content\_n1.png](/post/text-parsing-analysis-periodic-re~ 4

## 9 "" NA

## 10 ![viz\_sents\_section\_n.png](/post/text-parsing-analysis-periodic-r~ 5

## # ... with 170 more rows

content\_copypaste <- content\_gallery %>% pull(label\_md)

# Copy paste this to the markdown file for the gallery page.

# It's probably possible to do this a bit more programmatically (i.e. without

# "manually" copying into a markdown file, but oh well

clipr::write\_clip(content\_copypaste)