WalmartSales

Libraries and Data

```r
library(ggplot2)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.4     v tibble    3.3.0
v purrr     1.1.0     v tidyr     1.3.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.4.0 --
v broom        1.0.9     v rsample      1.3.1
v dials        1.4.2     v tune         2.0.1
v infer        1.0.9     v workflows    1.3.0
v modeldata    1.5.1     v workflowsets 1.1.1
v parsnip      1.3.3     v yardstick    1.3.2
v recipes      1.3.1
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
```

```r
library(patchwork)
library(tune)
library(vroom)
```

```

Attaching package: 'vroom'

The following object is masked from 'package:yardstick':

    spec

The following object is masked from 'package:scales':

    col_factor

The following objects are masked from 'package:readr':

    as.col_spec, col_character, col_date, col_datetime, col_double,
    col_factor, col_guess, col_integer, col_logical, col_number,
    col_skip, col_time, cols, cols_condense, cols_only, date_names,
    date_names_lang, date_names_langs, default_locale, fwf_cols,
    fwf_empty, fwf_positions, fwf_widths, locale, output_column,
    problems, spec
```

```r
library(dplyr)
library(embed)
library(kknn)


train <- vroom('./train.csv')
```

```
Rows: 421570 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl  (3): Store, Dept, Weekly_Sales
lgl  (1): IsHoliday
date (1): Date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
test <- vroom('./test.csv')
```

```
Rows: 115064 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl  (2): Store, Dept
lgl  (1): IsHoliday
date (1): Date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
features <- vroom("./features.csv")
```

```
Rows: 8190 Columns: 12
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl  (10): Store, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3, ...
lgl   (1): IsHoliday
date  (1): Date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
stores   <- vroom("./stores.csv")
```

```
Rows: 45 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Type
dbl (2): Store, Size

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

EDA Stuff

```r
# how many stores / depts / weeks?
train %>%
  summarise(
    n_rows   = n(),
    n_store  = n_distinct(Store),
    n_dept   = n_distinct(Dept),
    min_date = min(Date),
    max_date = max(Date)
  )
```

```
# A tibble: 1 x 5
  n_rows n_store n_dept min_date   max_date
   <int>   <int>  <int> <date>     <date>
1 421570      45     81 2010-02-05 2012-10-26
```

```r
# sales summary
train %>%
  summarise(
    mean_sales = mean(Weekly_Sales),
    median_sales = median(Weekly_Sales),
    min_sales = min(Weekly_Sales),
    max_sales = max(Weekly_Sales)
  )
```

```
# A tibble: 1 x 4
  mean_sales median_sales min_sales max_sales
       <dbl>        <dbl>     <dbl>     <dbl>
1     15981.        7612.    -4989.   693099.
```

```r
# check for negative sales
train %>%
  filter(Weekly_Sales < 0) %>%
  count()
```

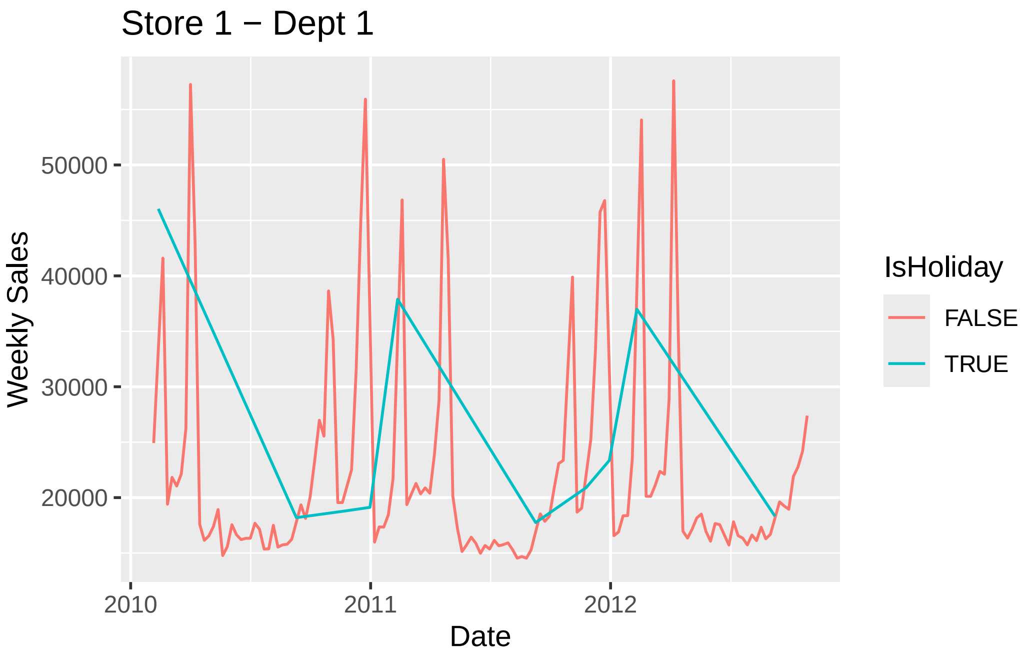
```
# A tibble: 1 x 1
      n
  <int>
1  1285
```

```r
features %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  t()
```

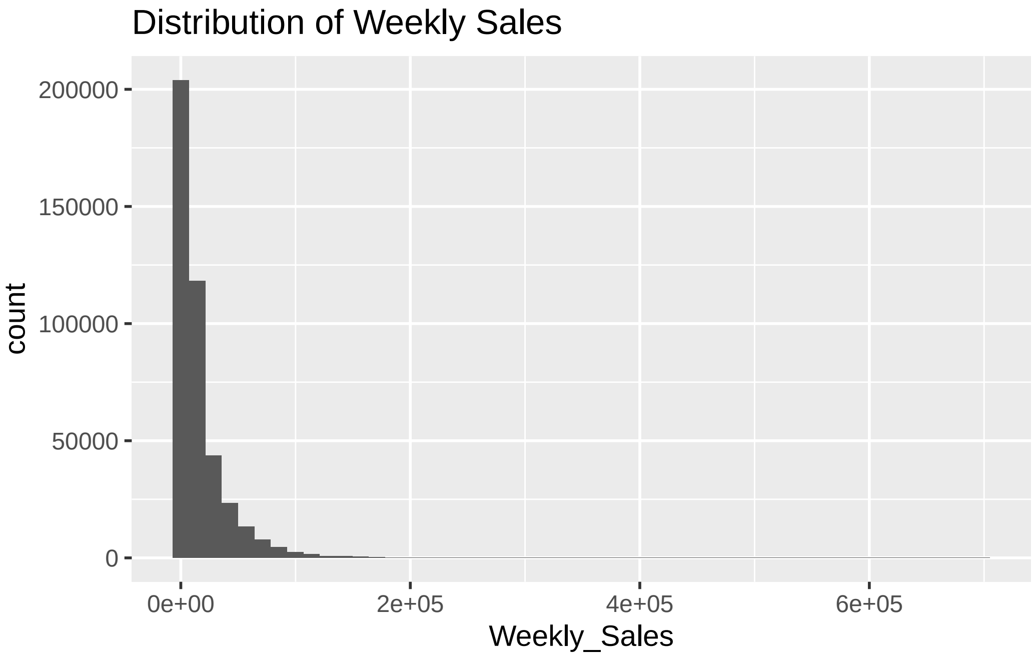
```
             [,1]
Store           0
Date            0
Temperature     0
Fuel_Price      0
MarkDown1    4158
MarkDown2    5269
MarkDown3    4577
MarkDown4    4726
MarkDown5    4140
CPI           585
Unemployment  585
IsHoliday       0
```

```r
store_example <- 1
dept_example  <- 1

train %>%
  filter(Store == store_example, Dept == dept_example) %>%
  ggplot(aes(x = Date, y = Weekly_Sales, color = IsHoliday)) +
  geom_line() +
  labs(title = paste("Store", store_example, "- Dept", dept_example),
       y = "Weekly Sales", x = "Date")
```



```
ggplot(train, aes(x = Weekly_Sales)) +
  geom_histogram(bins = 50) +
  labs(title = "Distribution of Weekly Sales")
```



We've got some issues, not super clear from the code I was able to produce above.

BUT
We need to figure out a good way to join data together (join features to train on store and date)
There are some stores that only have like 5 data points to use in predicting

And there are holes in the data where we need to figure out what to do with. maybe imputation or something.