Raw traffic count data

Count (value) and validity (suspect), specific to a given station, direction, traffic type

Value & suspect pair for another station, direction, traffic type

Dummy example data, `df_raw`:

datetime	<station>.<direction1>.<trafficType1>.**value**	<station>.<direction1>.<trafficType1>**.suspect**	<station>.<direction1>.<trafficType2>**.value**	<station>.<direction1>.<trafficType2>**.suspect**	…
<text>	<integer>	<code>	<integer>	<code>	…
<text>	<integer>	<code>	<integer>	<code>	…

Formatting

1) Drop columns where every observation is missing
2) Pivot the dataframe longer
3) Remove any rows where values are missing
4) Remove observations which are invalid

```
df_for <- df_raw |>
    ① dplyr::select(where(~ !all(is.na(.)))) |>
    ② pivot_longer(cols = -datetime,
          names_to = c("station", ".value"),
          names_pattern = "^(.*)\\.(suspect|value)$") |>
    ③ drop_na(value) |>
    ④ filter(suspect %in% c("ABV","IO2","IO3","IO4","IO5","IO6","VAL"))
```

`df_for`:

datetime	station	value	suspect
<text>	<station>.<direction1>.<trafficType1>	<integer>	<code>
<text>	<station>.<direction1>.<trafficType2>	<integer>	<code>
<text>	…	<integer>	<code>

Preprocessing

1) Format datetime into POSIXct. Obtain additional temporal info, calculate traffic type
2) Filter out observations not in season, not during the day (7am-6pm), and traffic that is not bicycle or pedestrian

```
df_pre <- df_for |>
    ①mutate(datetime = lubridate::parse_date_time(datetime),
            TrafficTime = lubridate::hour(datetime) +
                          lubridate::minute(datetime)/60,
            Date = as.Date(datetime),
            Month = lubridate::month(datetime),
            TrafficType = case_when(grepl('.Ped', station) ~ 'Ped',
                                    grepl('.Bic', station) ~ 'Bic',
                                    TRUE ~ 'Other')) |>
    ②filter(Month %in% forecasting_season_months[[region]],
            TrafficTime %in% seq(from = 7, to = 17.75, by = 0.25),
            TrafficType != 'Other')
```

Summarizing by time of day, and merging with other data

1) Obtain the time of day (TOD)
2) Group by TOD/day/traffic
3) Calculate inclusion criteria `keep` and `keep2` for each group. These are based on `n_obs`, `n_dir`, `n_unique`. Filter by these criteria
4) Summarize the traffic count for each group*
5) Join with other dataframes

```
morning <- seq(from = 7, to = 10.75, by = 0.25) # 7am-11am
midday <- seq(from = 11, to = 13.75, by = 0.25) # 11am-2pm
afternoon <- seq(from = 14, to = 17.75, by = 0.25) #2pm-6pm
h_timeofday <- list('Morning' = 4, 'Midday' = 3, 'Afternoon' = 4)

df_fin <- df_pre |>
    ① mutate(TimeOfDay = case_when(TrafficTime %in% morning ~ 'Morning',
                                   TrafficTime %in% midday ~ 'Midday',
                                   TrafficTime %in% afternoon ~ 'Afternoon',
             Hours = h_timeofday[[TimeOfDay]]) |>
    ② group_by(Date, TimeOfDay, TrafficType) |>
    ③ mutate(n_obs = n(), # total observations
             n_dir = n_distinct(station), # distinct directions per station
             n_unique = n_distinct(TrafficTime), # distinct time observations
             keep = n_obs == Hours * 4 * n_dir, # completeness check
             keep2 = n_obs == n_unique * n_dir) |> # uniqueness check
      filter(keep, keep2) |>
    ④ summarise(TrafficCount = sum(TrafficCount)*|>
    ⑤ left_join(aqi_dates_df, by = c('Date')) |>
      left_join(weather_df, by = c('Date','TimeOfDay'))
```

*Stations with at least 1 non-integer count observation (`TrafficCount %%1 !=0`) were filtered out after step 4.