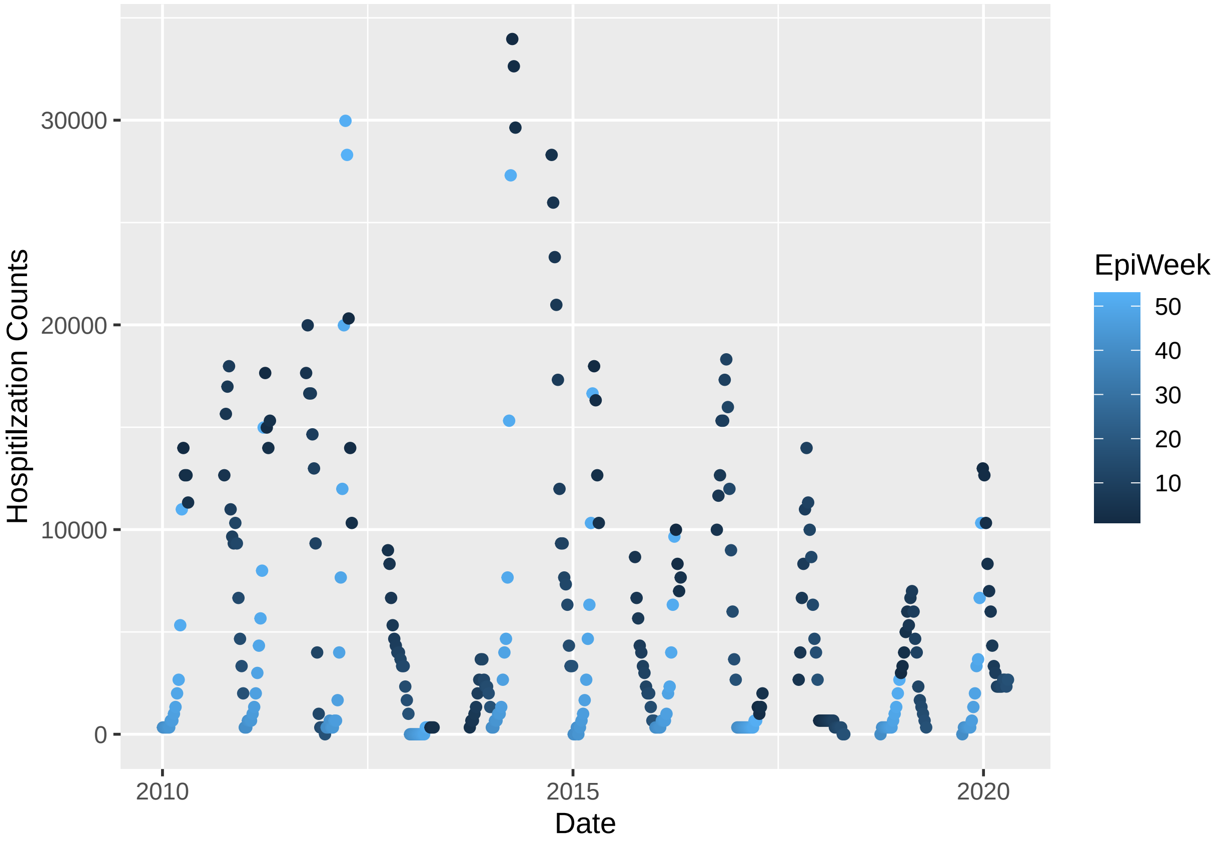
Historical Severity Statistics

SAJ

1/7/2022

```
# pull data from cdcfluview
hosp <- cdcfluview::hospitalizations(surveillance_area = "flusurv",
                                     region="all")

# visualize data to get oriented
hosp %>%
  filter(age_label == "Overall") %>%
  filter(year %>% between(2010, 2021)) %>%
  mutate("counts" = as.numeric(weeklyrate)*3330) %>%
  dplyr::select(c("wk_start", "counts", "year_wk_num")) %>%
  rename(EpiWeek = year_wk_num) %>%
  ggplot(aes(x = sort(wk_start), y = as.numeric(counts))) +
  geom_point(aes(color = EpiWeek)) +
  xlab("Date") +
  ylab("Hospitilzation Counts")
```



```
# for every epiweek, calculate fivenum based on count dist
hospStats <-
  hosp %>%
    filter(age_label == "Overall") %>%
    filter(year %>% between(2010, 2021)) %>%
    mutate("counts" = as.numeric(weeklyrate)*3300) %>%
    dplyr::select(c("wk_start", "counts", "year_wk_num")) %>%
    rename(EpiWeek = year_wk_num) %>%
    group_by(EpiWeek) %>%
    summarise(Minimum = min(counts),
              LowerHinge = IQR(counts, 0.25),
              Median = median(counts),
              UpperHinge = IQR(counts, 0.75),
              Mean = mean(counts))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
hospStats
```

```
## # A tibble: 31 x 6
##    EpiWeek Minimum LowerHinge Median UpperHinge  Mean
##      <int>   <dbl>      <dbl>  <dbl>      <dbl> <dbl>
##  1       1     330      15510  12870      15510 11910
##  2       2     330      12210  12540      12210 10530
##  3       3     330       9900  10230       9900  9270
##  4       4     330       7260   8250       7260  8850
##  5       5     660       6930   8250       6930  9030
##  6       6     660       8910   6600       8910  9060
##  7       7     660       9735   6600       9735  8790
##  8       8     660      12045   6930      12045  8730
##  9       9     660       7920   5940       7920  7560
## 10      10     660       8250   4620       8250  7290
## # ... with 21 more rows
```

```
# # distribution fit tests
# # create vector of counts
# CountsDF <-
#   hosp %>%
#     filter(age_label == "Overall") %>%
#     filter(year %>% between(2010, 2019)) %>%
#     #rename(EpiWeek = year_wk_num) %>%
#     mutate("counts" = as.numeric(rate)*100000)
# CountsVec <- CountsDF$counts
#
# # see what dist might fit
# descdist(CountsVec, discrete = F)
#
# # transform the counts to fit beta range (0,1)
# # note: adding tiny value to allow fitdist to not calculate inf/-inf
# range01 <- function(x){(x - min(x) + 0.001)/(max(x) - min(x) + 0.002)}
#
# # fit dists
# fit.beta <- fitdist(range01(CountsVec), "beta")
# fit.norm <- fitdist(CountsVec, "norm")
#
# # first look at norm, doesn't look great
# plot(fit.norm)
#
# # now look at beta, definitely more plausible
# plot(fit.beta)
```