SemiMech

First we load the library.

The main function exported is called `semimech:generate_samples()` which takes a `start` argument, the beginning of the time series. `dates` a sequence of dates to forecast, `region` a flag for which geographic resolution to forecast (right now only US states), `target` a string ("hosp","cases","deaths") representing the target (right now set to hospitalizations) and `inc_dat` a data frame of disease incidence data (right now must be the result of a call to get_hhs.)

Get the data!

```
library(stringr)
hhs <- get_hhs()
```

Define the forecast dates.

```
dates <- c(as.Date("2021-07-12"))
```

Get the samples.

```
samples <- semimech::generate_samples("2020-08-01",dates,"states",hhs)
```

Plot the samples

```
library(ggplot2)

median_mat <- matrix(NA,nrow=length(names(samples)),ncol=30)
upper_ci_mat <- matrix(NA,nrow=length(names(samples)),ncol=30)
lower_ci_mat <- matrix(NA,nrow=length(names(samples)),ncol=30)

row_idx <- 1
for (key in names(samples)){
  median_mat[row_idx,] <- colMeans(samples[[key]])
  upper_ci_mat[row_idx,] <- apply(samples[[key]],2,function(x){quantile(x,probs=.975)})
  lower_ci_mat[row_idx,] <- apply(samples[[key]],2,function(x){quantile(x,probs=.025)})

  row_idx <- row_idx + 1
}


ci_mat_df <- data.frame(median_ = c(median_mat),upper_ci=c(upper_ci_mat),lower_ci=c(lower_ci_mat),st




p <- ggplot(ci_mat_df,aes(x=date,y=median_)) + geom_line() +
  geom_point(data=hhs[hhs$date > dates-30 & hhs$date < dates+30,],aes(x=as.Date(date),y=previous_day


ggsave(filename=paste0("../figs/",dates[1],".png"),plot=p,height = 10,width = 12)
```

Generate submission file.

```
semimech::generate_submission_file(dates,samples,"UT-test","sub_files")
```

```
## Rows: 51
## Columns: 4
## $ fips       <int> 1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20~
## $ state_full <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "California", "~
## $ state      <chr> "AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "DC", "FL",~
## $ alphacount <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ~
```

Let's do a full years worth of eval.

```
date_seq <- seq(as.Date("2021-07-12"),as.Date("2021-07-12")+50*7,by="week")
for (date_idx in 1:length(date_seq)){
  samples <- semimech::generate_samples("2020-08-01",date_seq[date_idx],"states",hhs)
  semimech::generate_submission_file(date_seq[date_idx],samples,"UT-test","sub_files")
}
```

Let's evaluate with respect to baseline and ensemble. **Note** Before you evaluate you must copy the files from `sub_files` and placed in the `data_processed/UT-test` folder in a cloned version of `https://github.com/reichlab/covid19-forecast-hub` .

```
library(covidHubUtils)
fcasts <- load_forecasts(
  source = "local_hub_repo",
  models =c("UT-test"),
  targets = c(paste0(rep(1:28), " day ahead inc hosp")),
  data_processed_subpath = "data-processed/",
  hub_repo_path = "/Users/gcg799/covid19-forecast-hub/",
  hub = "US")

fcasts_baseline <- load_forecasts(
  source = "local_hub_repo",
  models =c("COVIDhub-baseline","COVIDhub-ensemble"),date_window_size = 3,
  dates = unique(fcasts$forecast_date),locations = unique(fcasts$location),

  targets = c(paste0(rep(1:28), " day ahead inc hosp")),
  data_processed_subpath = "data-processed/",
  hub_repo_path = "/Users/gcg799/covid19-forecast-hub/",
  hub = "US")


truth <- load_truth(hub = "US")
truth <- truth[truth$target_variable == "inc hosp",]

scores <- score_forecasts(rbind(fcasts,fcasts_baseline),truth = truth,metrics = c("abs_error","wis",
scores_common_location <- unique(scores[scores$model == "UT-test",]$location)
scores_common_forecast_date <- unique(scores[scores$model == "UT-test",]$forecast_date)

summ_scores <- scores[scores$location %in%scores_common_location &
              scores$forecast_date %in% scores_common_forecast_date,] %>% group_by(model) %>% summar


print (summ_scores)
```

```
## # A tibble: 3 x 4
##   model             coverage_90   wis abs_error
```

```
##   <chr>                  <dbl> <dbl>     <dbl>
## 1 COVIDhub-baseline      0.906  54.0      78.8
## 2 COVIDhub-ensemble      0.875  34.5      52.9
## 3 UT-test                0.848  38.8      57.0
```