Regression and Other Stories: Simulation

Simulation

Simulation is important element of applied statistics (as well as probability/ mathematical statistics).

Why simulation?

Simulation can be used for discrete or continuous outcomes,

- With discrete (binary) outcomes,

- With discrete (count) outcomes,

- With continuous outcomes, normal or lognormal are common, but there are others too.

Simulation Activity

According to the MSU promotional materials https://www.montana.edu/marketing/about-msu/bozeman/, Bozeman has 300 "days of sunshine".

1. Using this information, simulate the probability that a given week has all sunny days.

2. Using this information, simulate the probability that the next three days are "not sunny days."

3. What assumptions are you making with these simulations? Hint: do three consecutive "not sunny days" seem equally likely in the summer / winter?

Now let's consider a simulation for that captures the likelihood that an air quality measurement will fall in the good level (< 12 μg/m): https://svc.mt.gov/deq/todaysair/.

4. Assume the hourly particulate measurement can be approximated with a lognormal distribution with parameters mean (log 10) and sd of 1. Note these are the `meanlog` and `sdlog` parameters in `rlnorm`. Create a data visualization to accompany this figure.

5. Finally let's scrape actual air quality data from https://svc.mt.gov/deq/todaysair/. And create a data visualization.

```
library(rvest)
scrape_BZNPM <- function(days){
  # Scrapes hourly PM2.5 readings in Bozeman for September 2020
  # inputs: sequence of days
  # outputs: data frame that contains day, hour, and hourly average PM2.5 concentration.
  smoke.df <- data.frame(day = NULL, hour = NULL, conc = NULL)
  for (d in days){
    bzn_aq <- read_html(paste("http://svc.mt.gov/deq/todaysair/AirDataDisplay.aspx?siteAcronym=BH&targe
    daily.smoke <- cbind(rep(d,24),html_table(bzn_aq)[[1]][,1:2])
    colnames(daily.smoke) <- c('day','hour','conc')
    daily.smoke$conc[daily.smoke$conc == 'DU'] <- 'NA'
    daily.smoke$conc <- as.numeric(daily.smoke$conc)
    smoke.df <- bind_rows(smoke.df,daily.smoke)
  }
  return(smoke.df)
}


air.quality <- scrape_BZNPM(1:13)

conc <- air.quality %>%
  filter(!is.na(conc)) %>%
  select(conc)
```

Bootstrap

Example. Using the air quality data, we can implement the bootstrap.

A single bootstrap replicate

```
conc %>% sample_frac(1, replace = T) %>% head()
```

```
##   conc
## 1  5.5
## 2 12.7
## 3  6.3
## 4 15.1
## 5  0.6
## 6  9.7
```

Bootstrap replicates to simulate the sampling distribution of mean air quality measurement.

```
num_replicates <- 1000

boot_mean <- rep(0, num_replicates)

for (i in 1:num_replicates){
  boot_mean[i] <- conc %>% sample_frac(1, replace = T) %>% summarise(mean(conc)) %>% pull()
}

tibble(boot_mean = boot_mean) %>%
  ggplot(aes(x = boot_mean)) +
  geom_histogram( bins = 30) +
  theme_bw() + xlab('Mean Air Quality') +
  ggtitle('Bootstrap distribution of Mean Air Quality')
```