Regression and Other Stories: Simulation

Simulation

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

Why simulation?

1. *Probability models mimic variation in the world and the tools of simulation help us better understand variation. Variation/randomness is not necessarily an intuitive process.*

2. *Simulation allows us to approximate the generative model of the data and pass uncertainty to estimates. In general, we only have one draw from the generative model (the dataset), but simulation allows us to understand variability from the generative model (CLT simulation)*

3. *Simulation can be used to generate probabilistic predictions.*

Simulation can be used for discrete or continuous outcomes, *but requires an assumed probability model*

- With discrete (binary) outcomes, *the binomial distribution is common:* `rbinom`

- With discrete (count) outcomes, *the Poisson or negative-binomial distributions are common*

- 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.

```
sun_prob <- 300 / 365
num_sims <- 100000

sunny_week <- mean(rbinom(num_sims,7,sun_prob) == 7)
```

The probability that a week has all sunny days is approximately 0.253.

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

```
not_sunny <- mean(rbinom(num_sims,3,1 - sun_prob) == 3)
```

The probability that the next three days are "not sunny" is approximately 0.006.

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

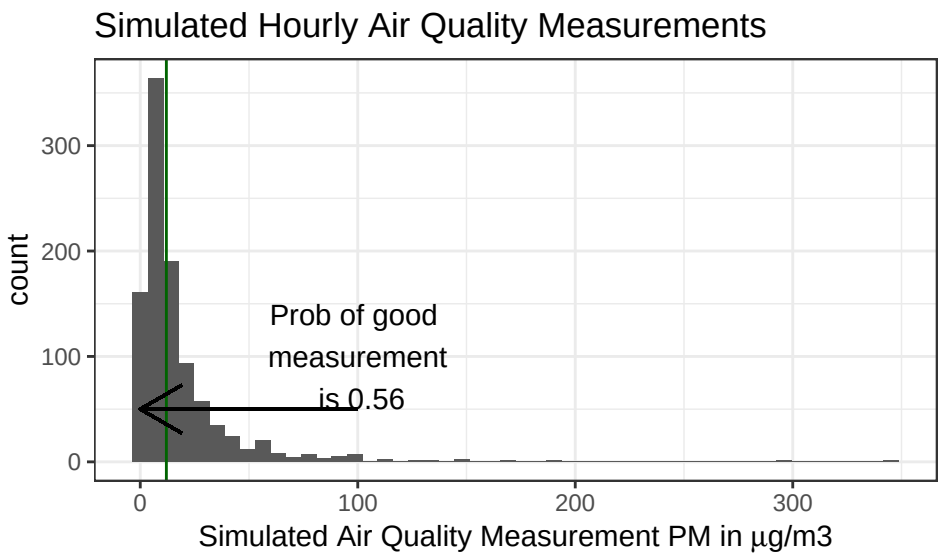
This simulation treats each day as an independent draw where the probability of a sunny day is 300/365 = 0.82. Based on my experiences in Bozeman, it tends to be sunnier in the summer and less sunny (snowy!) in the winter

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.

```
num_sims <- 1000
conc <- tibble(conc = rlnorm(num_sims, log(10), 1))
good_prob <- round(mean(conc < 12),2)

conc %>% ggplot(aes(x = conc)) + geom_histogram( bins = 50) +
  theme_bw() + xlab(expression(paste('Simulated Air Quality Measurement PM in ', mu, 'g/m3'))) +
  ggtitle('Simulated Hourly Air Quality Measurements') +
  geom_vline(xintercept = 12, color = 'darkgreen') +
  annotate('text', label = paste('Prob of good \n measurement \n is', good_prob), x = 100, y = 100) +
  geom_segment( x = 100, y = 50, yend = 50, xend = 0, arrow = arrow())
```



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

```
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 4.0.2
## Loading required package: xml2
## Warning: package 'xml2' was built under R version 4.0.2
##
## Attaching package: 'rvest'
## The following object is masked from 'package:purrr':
##
##     pluck
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

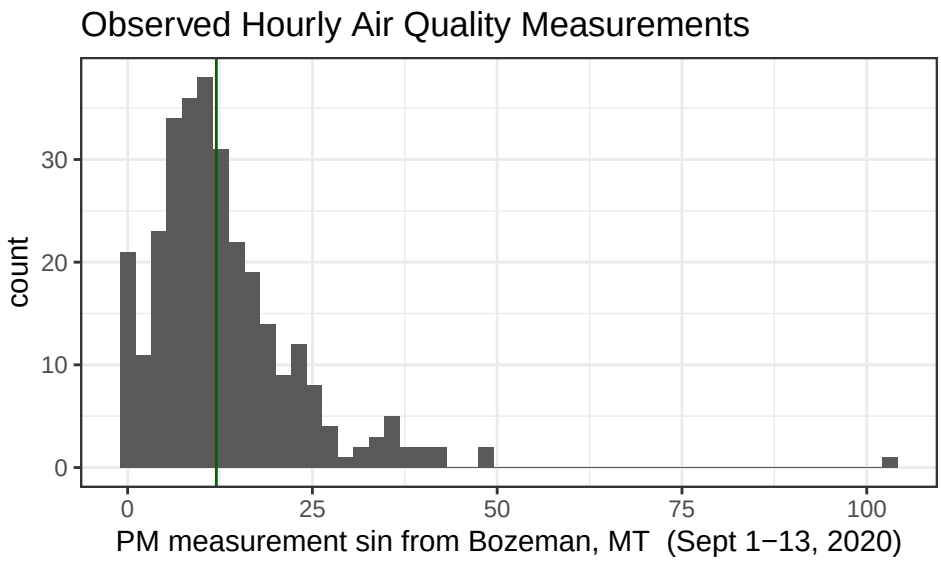
```
scrape_BZNPM <- function(days){
  # Scrapes hourly PM2.5 readings in Bozeman for September 2020
  # inputs: sequence of days
  # outputs: data frame that contains day, hour, and hourly average PM2.5 concentration.
  smoke.df <- data.frame(day = NULL, hour = NULL, conc = NULL)
  for (d in days){
    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)
```

```
## Warning in scrape_BZNPM(1:13): NAs introduced by coercion
```

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

conc %>% ggplot(aes(x = conc)) + geom_histogram( bins = 50) +
  theme_bw() + xlab('PM measurement sin from Bozeman, MT  (Sept 1-13, 2020)') +
  ggtitle('Observed Hourly Air Quality Measurements') +
  geom_vline(xintercept = 12, color = 'darkgreen')
```



Bootstrap

The bootstrap algorithm is a procedure to resample from the data to approximate a sampling distribution.

A sample of size n is drawn with replacement

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

A single bootstrap replicate

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

```
##   conc
## 1 16.1
## 2 19.0
## 3 20.0
## 4  5.0
## 5 13.5
## 6  3.8
```

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')
```