# STA 610 LAB 4

William Tirone

2023-09-20

## EDA

Since this is the same data, I am just copying my EDA from last time:

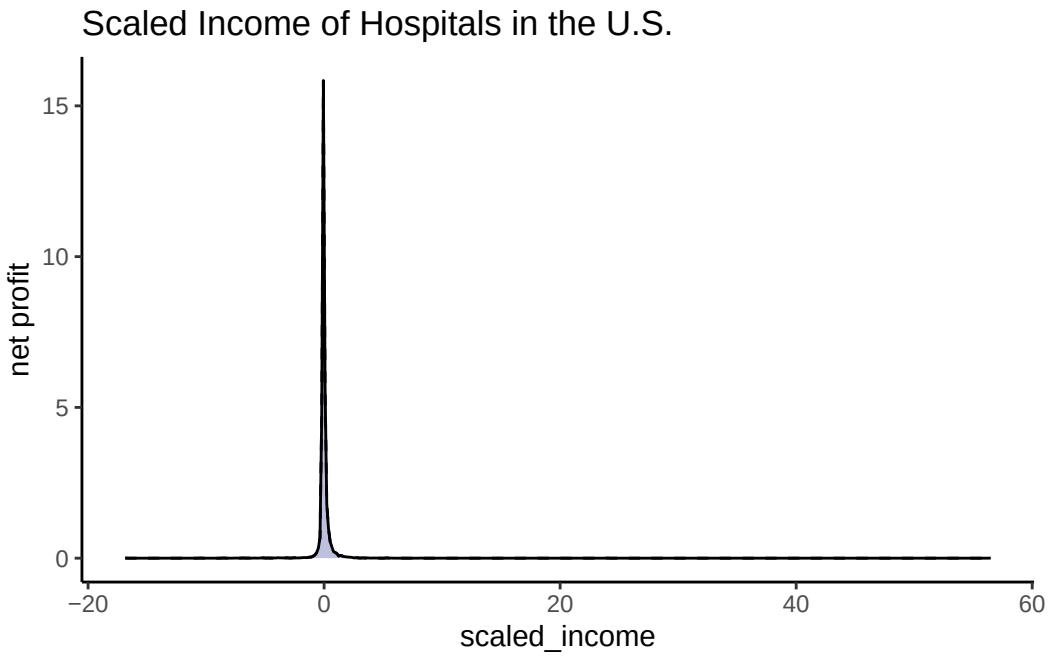
```
agg = hc2014 |>
  group_by(control) |>
  summarise(total_beds = sum(numbeds)) |>
  arrange(desc(total_beds))

agg[0:10,]
```

```
# A tibble: 10 x 2
   control                total_beds
   <fct>                       <dbl>
 1 Nonprofit-other            383324
 2 Corporation                166482
 3 Nonprofit-church           104345
 4 Gvmt-State                  46498
 5 Gvmt-County                 31339
 6 Gvmt-Hospital District      27978
 7 Gvmt-Other                  12480
 8 Partnership                 10317
 9 Gvmt-City-County             9225
10 Gvmt-City                    7935
```

We see a very sharp spike at 0, but also some pretty large outliers.

```
ggplot(hc2014, aes(scaled_income)) +
  geom_density(aes(y=..density..),color="black",linetype="dashed") +
  theme(legend.position="none") +
  geom_density(alpha=.25, fill="navy") +
  labs(title="Scaled Income of Hospitals in the U.S.",y="net profit") +
  theme_classic()
```

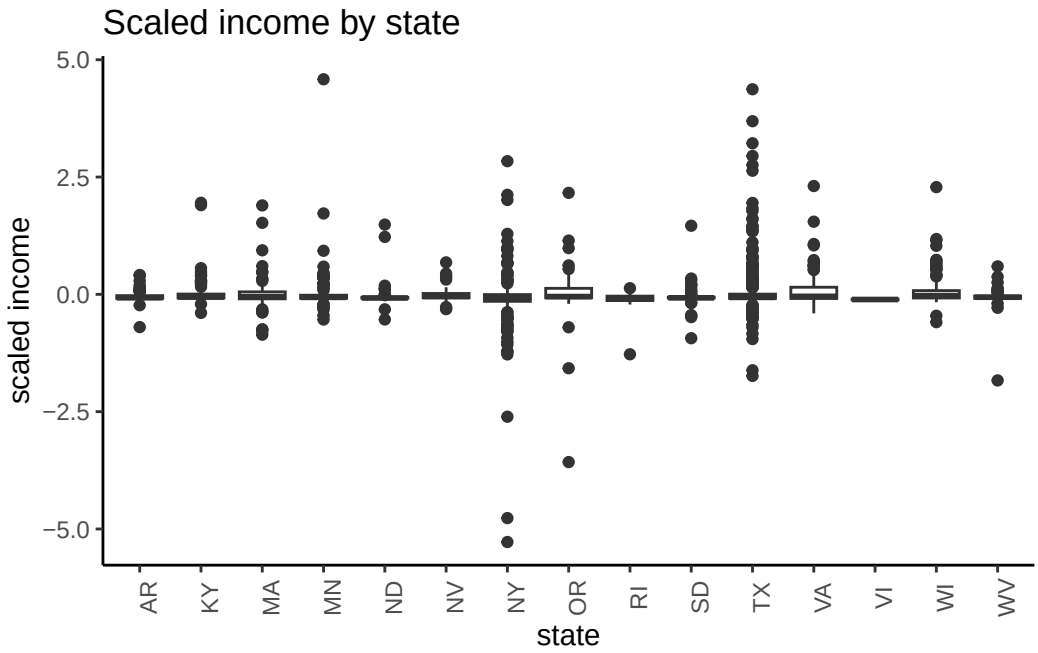
```
Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
```




Below, it looks like with a random sample of states, we have a pretty significant spread among them. New York has some hospitals with pretty large negative scaled profits and Texas has some with very positive values.

```
set.seed(1000)
sample_state <- sample(unique(hc2014$state),15,replace=F)
ggplot(hc2014[is.element(hc2014$state, sample_state),],
       aes(x=state, y=scaled_income)) +
  geom_boxplot() +
  labs(title="Scaled income by state",
       x="state",y="scaled income") + theme_classic() +
```

```
theme(legend.position="none",axis.text.x = element_text(angle = 90))
```




However, last time we saw that there wasn't much difference between income and scaled income, so we will just proceed with regular income here and see what happens.

## Model

Since I am using the Gibb's sampler approach, I will use the shortened version of the model:

$$y_{ij} = \mu_0 + \alpha_j + \epsilon_{ij}$$

# Gibb's Sampler

Here, I'm adapting code from the lecture slides but have replaced info from the data with calculated values from our data. For the hyperparemeters, I just set some random values and hoped they worked out.

```
# Data summaries
J <- 55

ybar <- grouped |>
```

```r
  summarise(mean(netincome)) |>
  pull()

# had some NA values, so just replacing these with a value
s_j_sq <- grouped |>
  summarise(v = var(netincome)) |>
  replace_na(list(v = 8.449697e+15)) |>
  pull()

n <- grouped |>
  summarise(n()) |>
  pull()

# Hyperparameters for the priors
mu_0 <- 1000000
gamma_0_sq <- 10000
eta_0 <- 5
tau_0_sq <- 5
alpha <- 500
a <- 3
b <- 3

# Grid values for sampling nu_0_grid
nu_0_grid <- 1:5000

# Initial values for Gibbs sampler
theta <- ybar # Theta vector for all the mu_j's
sigma_sq <- s_j_sq
mu <- mean(theta)
tau_sq <- var(theta)
nu_0 <- 1
sigma_0_sq <- 100

# First, set the number of iterations and burn-in, then set the seed
n_iter <- 5000
burn_in <- 0.3 * n_iter

# Set null matrices to save samples
SIGMA_SQ <- THETA <- matrix(nrow = n_iter, ncol = J)
OTHER_PAR <- matrix(nrow = n_iter, ncol = 4)
```

```r
# Now, to the Gibbs sampler
for (s in 1:(n_iter + burn_in)) {

  # Update the theta vector (all the mu_j's)
  tau_j_star <- 1 / (n / sigma_sq + 1 / tau_sq)
  mu_j_star <- tau_j_star * (ybar * n / sigma_sq + mu / tau_sq)
  theta <- rnorm(J, mu_j_star, sqrt(tau_j_star))

  # Update the sigma_sq vector (all the sigma_sq_j's)
  nu_j_star <- nu_0 + n
  theta_long <- rep(theta, n)
  nu_j_star_sigma_j_sq_star <- nu_0 * sigma_0_sq + c(by((hc2014[, "netincome"] - theta_lon
  sigma_sq <- 1 / rgamma(J, (nu_j_star / 2), (nu_j_star_sigma_j_sq_star / 2))

  # Update mu
  gamma_n_sq <- 1 / (J / tau_sq + 1 / gamma_0_sq)
  mu_n <- gamma_n_sq * (J * mean(theta) / tau_sq + mu_0 / gamma_0_sq)
  mu <- rnorm(1, mu_n, sqrt(gamma_n_sq))

  # Update tau_sq
  eta_n <- eta_0 + J
  eta_n_tau_n_sq <- eta_0 * tau_0_sq + sum((theta - mu)^2)
  tau_sq <- 1 / rgamma(1, eta_n / 2, eta_n_tau_n_sq / 2)

  # Update sigma_0_sq
  sigma_0_sq <- rgamma(1, (a + J * nu_0 / 2), (b + nu_0 * sum(1 / sigma_sq) / 2))

  # Update nu_0
  log_prob_nu_0 <- (J * nu_0_grid / 2) * log(nu_0_grid * sigma_0_sq / 2) -
    J * lgamma(nu_0_grid / 2) +
    (nu_0_grid / 2 + 1) * sum(log(1 / sigma_sq)) -
    nu_0_grid * (alpha + sigma_0_sq * sum(1 / sigma_sq) / 2)
  nu_0 <- sample(nu_0_grid, 1, prob = exp(log_prob_nu_0 - max(log_prob_nu_0)))

  # Save results only past burn-in
  if (s > burn_in) {
    THETA[(s - burn_in), ] <- theta
    SIGMA_SQ[(s - burn_in), ] <- sigma_sq
    OTHER_PAR[(s - burn_in), ] <- c(mu, tau_sq, sigma_0_sq, nu_0)
  }
}
```

```
colnames(OTHER_PAR) <- c("mu", "tau_sq", "sigma_0_sq", "nu_0")
colnames(THETA) = hc2014 |> distinct(state) |> pull()

THETA = data.frame(THETA)
```

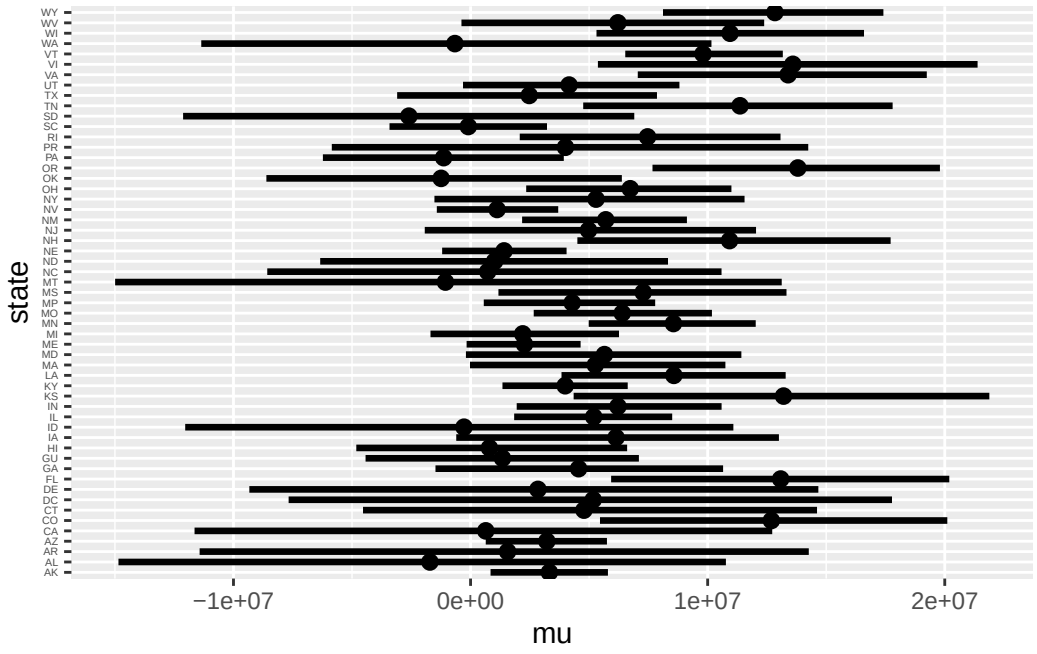
We have 55 + states and territories, but here is a small selection of them to view what `THETA` looks like:

```
THETA[, 1:5] |>
  head(5) |>
  kable()
```

| AL | AK | AZ | AR | CA |
|---|---|---|---|---|
| -6668031.5 | 4825700 | 4009072 | 3929035.0 | -2371221 |
| -1274376.3 | 1611160 | 5876031 | -8201309.2 | 3646816 |
| 5179050.0 | 3911948 | 5647149 | -8534262.7 | -3102231 |
| -177121.9 | 1635712 | 7233335 | -338294.6 | 4228613 |
| -1119764.6 | 2392002 | 3076061 | 4074214.7 | -3452417 |

And most importantly, the plot of our estimates along with uncertainty quantification. The mean netincome is all over the place and there are also huge amounts of uncertainty based on state. It looks like Montana has a very wide range, while Arizona has a very narrow range.

```
THETA |>
  gather() |>
  rename(state = key,
         mu = value) |>
  group_by(state) |>
  median_qi(.width = 0.95) |>
  ggplot(aes(y = state, x = mu, xmin = .lower, xmax = .upper)) +
  geom_pointinterval() +
  theme(axis.text.y = element_text(hjust = 1, size = 4))
```




And in fact, looking back at our data, there are fewer hospitals in Montana, so this emphasizes the need to share data across groups since we have fewer observations in some states.

```
grouped |>
  filter(state %in% c('MT', 'AZ')) |>
  summarise(count = n())
```

```
# A tibble: 2 x 2
  state count
  <chr> <int>
1 AZ      112
2 MT       63
```