Stat 341 – Homework 04

Luca Seazzu

April 13, 2023

N1

```
fiji <- read_csv('https://sldr.netlify.app/data/fiji-filters.csv') |>
  mutate(household_annual_income = household_annual_income / 1000)
```

```
## Rows: 1006 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): town, time_point, water_source, season, severe_diarrhea_adults, sev...
## dbl (9): household_id, n_adults, n_kids, n_total, household_annual_income, m...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
n_grid = 500

grid_income_model <-
  crossing(
    mu = seq(from = 50, to = 65, length.out = n_grid),
    sigma = seq(from = 80, to = 95, length.out = n_grid)
    ) |>
  mutate(
    # based on: http://www.salaryexplorer.com/salary-survey.php?loc=72&loctype=1
    prior_mu = dnorm(mu, mean = 57.7, sd = 5),
    prior_sigma = dnorm(sigma, mean = 41, sd = 20)
  ) |>
  rowwise() |>
  mutate(
    logL = dnorm(
      fiji$household_annual_income,
      mean = mu,
      sd = sigma,
      log = TRUE
      ) |>
      sum()
    )|>
  ungroup() |>
  mutate(
    unscaled_ln_post = logL + log(prior_mu) + log(prior_sigma)
  ) |>
  mutate(
    scaled_posterior =
      exp(unscaled_ln_post - max(unscaled_ln_post)
```

```
        )
  )
glimpse(grid_income_model)
```

```
## Rows: 250,000
## Columns: 7
## $ mu                <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 5~
## $ sigma             <dbl> 80.00000, 80.03006, 80.06012, 80.09018, 80.12024, 80.~
## $ prior_mu          <dbl> 0.02437551, 0.02437551, 0.02437551, 0.02437551, 0.024~
## $ prior_sigma       <dbl> 0.002979735, 0.002971012, 0.002962307, 0.002953621, 0~
## $ logL              <dbl> -5975.169, -5975.065, -5974.961, -5974.857, -5974.753~
## $ unscaled_ln_post <dbl> -5984.699, -5984.598, -5984.497, -5984.396, -5984.295~
## $ scaled_posterior <dbl> 6.513992e-08, 7.210785e-08, 7.978891e-08, 8.825260e-0~
```

A.

Model Description

$$\text{household annual income} \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(\text{mean}_1 = 57.7, \text{sd}_1 = 5)$$

$$\sigma \sim \text{Normal}(\text{mean}_2 = 41, \text{sd}_2 = 20)$$

B.

```
n_grid = 500

grid_income_model <-
  crossing(
    mu = seq(from = 50, to = 65, length.out = n_grid),
    sigma = seq(from = 80, to = 95, length.out = n_grid)
    ) |>
  mutate(
    # based on: http://www.salaryexplorer.com/salary-survey.php?loc=72&loctype=1
    prior_mu = dnorm(mu, mean = 57.7, sd = 5),
    prior_sigma = dnorm(sigma, mean = 41, sd = 20)
  ) |>
  rowwise() |>
  mutate(
    likelihood = dnorm(
      fiji$household_annual_income,
      mean = mu,
      sd = sigma,
      ) |>
      sum()
    )|>
  ungroup() |>
  mutate(
    unscaled_post = likelihood + prior_mu + prior_sigma
  ) |>
```

```
  mutate(
    scaled_posterior =
      exp(unscaled_post - max(unscaled_post)
          )
  )
```

C.

Define the grid

```
grid_income_model2 <-
  crossing(
    mu = seq(from = 50, to = 65, length.out = n_grid),
    sigma = seq(from = 80, to = 95, length.out = n_grid)
    )
glimpse(grid_income_model2)
```

```
## Rows: 250,000
## Columns: 2
## $ mu    <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, ~
## $ sigma <dbl> 80.00000, 80.03006, 80.06012, 80.09018, 80.12024, 80.15030, 80.1~
```

Looks good so far.

Define the priors

```
grid_income_model2 <- grid_income_model2 |>
  mutate(
    # based on: http://www.salaryexplorer.com/salary-survey.php?loc=72&loctype=1
    prior_mu = dnorm(mu, mean = 57.7, sd = 5),
    prior_sigma = dnorm(sigma, mean = 41, sd = 20)
  )
glimpse(grid_income_model2)
```

```
## Rows: 250,000
## Columns: 4
## $ mu          <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ sigma       <dbl> 80.00000, 80.03006, 80.06012, 80.09018, 80.12024, 80.15030~
## $ prior_mu    <dbl> 0.02437551, 0.02437551, 0.02437551, 0.02437551, 0.02437551~
## $ prior_sigma <dbl> 0.002979735, 0.002971012, 0.002962307, 0.002953621, 0.0029~
```

Prior_mu and prior_sigma looks really small and looks the same as before.

Compute the likelihood

```
grid_income_model2 <- grid_income_model2 |>
  mutate(
    likelihood = dnorm(
      fiji$household_annual_income,
      mean = mu,
      sd = sigma,
      ) |>
      sum()
  )
glimpse(grid_income_model2)
```

```
## Rows: 250,000
## Columns: 5
```

```
## $ mu          <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ sigma       <dbl> 80.00000, 80.03006, 80.06012, 80.09018, 80.12024, 80.15030~
## $ prior_mu    <dbl> 0.02437551, 0.02437551, 0.02437551, 0.02437551, 0.02437551~
## $ prior_sigma <dbl> 0.002979735, 0.002971012, 0.002962307, 0.002953621, 0.0029~
## $ likelihood  <dbl> 944.016, 944.016, 944.016, 944.016, 944.016, 944.016, 944.~
```

Likelihood is almost a thousand as a positive number whereas the Log likelihood was around -6000.

Compute the posterior

```
grid_income_model2 <- grid_income_model2 |>
  mutate(
    unscaled_post = likelihood + prior_mu + prior_sigma)
glimpse(grid_income_model2)
```

```
## Rows: 250,000
## Columns: 6
## $ mu            <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, ~
## $ sigma         <dbl> 80.00000, 80.03006, 80.06012, 80.09018, 80.12024, 80.150~
## $ prior_mu      <dbl> 0.02437551, 0.02437551, 0.02437551, 0.02437551, 0.024375~
## $ prior_sigma   <dbl> 0.002979735, 0.002971012, 0.002962307, 0.002953621, 0.00~
## $ likelihood    <dbl> 944.016, 944.016, 944.016, 944.016, 944.016, 944.016, 94~
## $ unscaled_post <dbl> 944.0434, 944.0434, 944.0434, 944.0434, 944.0433, 944.04~
```

Appears the priors have little effect on the unscaled posterior because the likelihood is so high and the priors are so small.

Scale the posterior

```
grid_income_model2 <- grid_income_model2 |>
    mutate(
    scaled_posterior =
      exp(unscaled_post - max(unscaled_post)
          ))
glimpse(grid_income_model2)
```

```
## Rows: 250,000
## Columns: 7
## $ mu               <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 5~
## $ sigma            <dbl> 80.00000, 80.03006, 80.06012, 80.09018, 80.12024, 80.~
## $ prior_mu         <dbl> 0.02437551, 0.02437551, 0.02437551, 0.02437551, 0.024~
## $ prior_sigma      <dbl> 0.002979735, 0.002971012, 0.002962307, 0.002953621, 0~
## $ likelihood       <dbl> 944.016, 944.016, 944.016, 944.016, 944.016, 944.016,~
## $ unscaled_post    <dbl> 944.0434, 944.0434, 944.0434, 944.0434, 944.0433, 944~
## $ scaled_posterior <dbl> 0.9460944, 0.9460862, 0.9460779, 0.9460697, 0.9460615~
```

Scaled posterior is anow a value just under 1 around .95 whereas on the log scale it was around 0.

N2

```
set.seed(19)
phones <- read_csv('https://osf.io/download/r8s6n/') |>
  rename(participant_id = pp,
         program = faculty,
         phone_use = total_a20,
         percent_private_phone_use = privateUse,
         use = smartphoneUse,
```

```
        ) |>
  mutate(fomo_score = (fomo1 + fomo2 + fomo3)) |>
  select(participant_id,
         program,
         age,
         gender,
         fatigue,
         fomo_score,
         boredom,
         phone_use,
         percent_private_phone_use,
         use,) |>
  mutate(phone_frequency = case_when(use == 1 ~ 'never',
                                     use ==2 ~ 'once daily',
                                     use == 3 ~ 'several times a day',
                                     use == 4 ~ 'once an hour',
                                     use == 5 ~ 'several times an hour',
                                     use == 6 ~ 'every few minutes')) |>
  select(-use) |>
  drop_na(phone_use) |>
  filter(phone_use > 0)
```

```
## Rows: 3234 Columns: 26
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): gender, faculty
## dbl  (19): pp, age, yearPhD, day, time, fatigue, boredom, total_b10, total_a...
## time  (5): startWork, endWork, startBreak, endBreak, exactTime
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

A. Fit

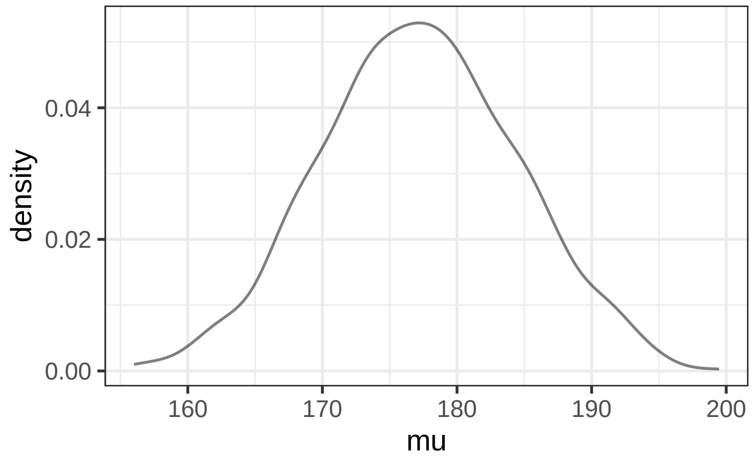
```
model_descrip <- alist(
  phone_use ~ dnorm(mu, sigma),
  mu ~ dnorm(mean = 120, sd =120),
  sigma ~ dnorm(mean = 300, sd = 100)
)

quap_phone_model <- quap(flist = model_descrip, data = phones)
```

B.

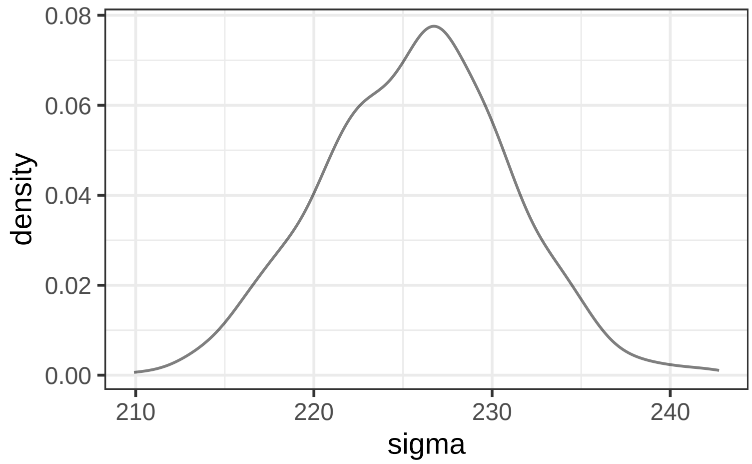
```
quap_phone_model <- quap(flist = model_descrip,
                          data = phones)
quap_phone_post_sample <- extract.samples(quap_phone_model, n = 1000)
```

```
gf_dens(~mu, data = quap_phone_post_sample)
```

```
## Warning: `stat(density)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
```




```
gf_dens(~sigma, data = quap_phone_post_sample)
```



Based on these graphs, the prior could've been something more like mu ~ norm(175, 15) and sigma ~ norm(225, 10)

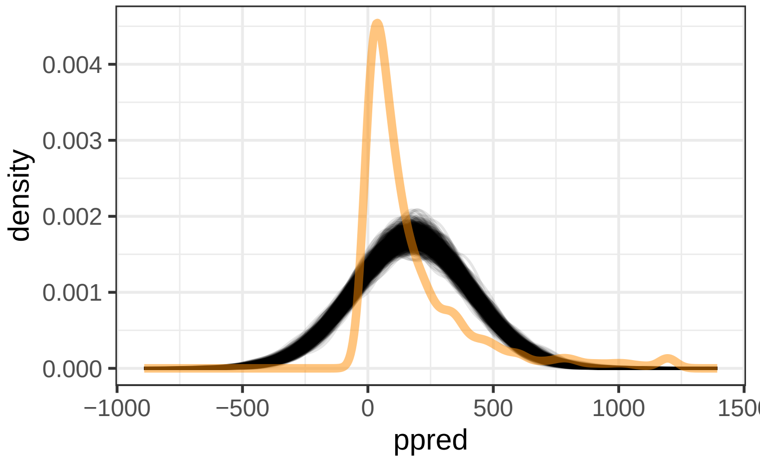
C.

```
phone_ppred <- quap_phone_post_sample |>
  # add row numbers to "label" each sampled combo of mu & sigma
  mutate(row_num = c(1:n())) |>
  # work one row (one mu, sigma combination) at a time
  rowwise() |>
  # simulate a dataset for each row (= each mu, sigma combo)
  mutate(ppred = list(rnorm(nrow(phones),
                            mean = mu,
                            sd = sigma))) |>
  # keep only the row-ids and the simulated data
  select(row_num, ppred)
```

```
phone_ppred <- phone_ppred |>
  unnest(cols = ppred)
```

```
gf_dens(~ppred, group = ~row_num,
        data = phone_ppred,
```

```
        alpha = 0.1) |>
  # overlay actual data
  gf_dens(~phone_use,
          data = phones,
          inherit = FALSE,
          color = 'darkorange',
          linewidth = 1.5)
```



This looks like the posterior is way too wide and isn't as centered near the real data because the priors weren't as informative as they could've been.

N3

A.

A quick websearch says that employees spend about 56 minutes on their phone during the 8 hour work day and that works out to about 140 seconds per 20 minutes. I think a standard deviation of 270 seconds is fair since it is about 2 and a half minutes per 20 minutes.

B.

```
gamma_params(mean = 140, sd = 30)
```

```
## # A tibble: 1 x 6
##   shape  rate scale  mode  mean    sd
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21.8 0.156  6.43  134.   140    30
```

$$\text{phone use} \sim \text{Gamma}(\alpha, \lambda)$$

$$\alpha \sim \text{Norm}(\text{mean} = 22, \text{sd} = 2)$$

$$\lambda \sim \text{Gamma}(\text{mean} = .15, \text{sd} = 0.05)$$

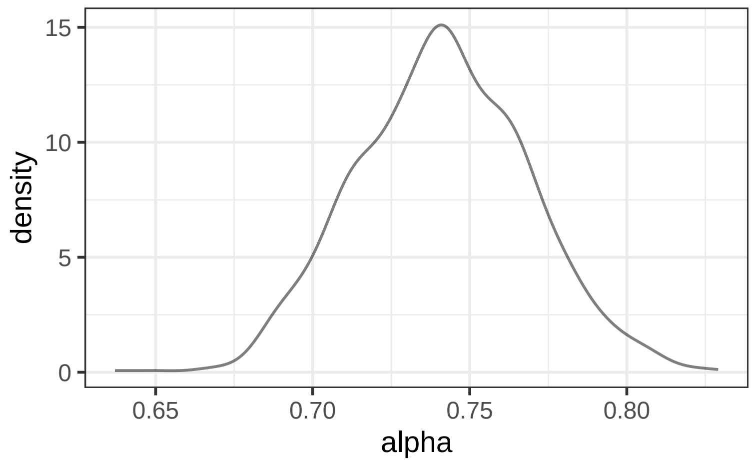
For alpha I did a norm prior of mean = 22 and sd = 2 because gamma_params put out 21.78 for a mean of 140 which would be my normal prior and for lambda I did mean = .15 with sd = 0.05 since gamma params put out a rate of .156 for a normal sd of 30.

B.

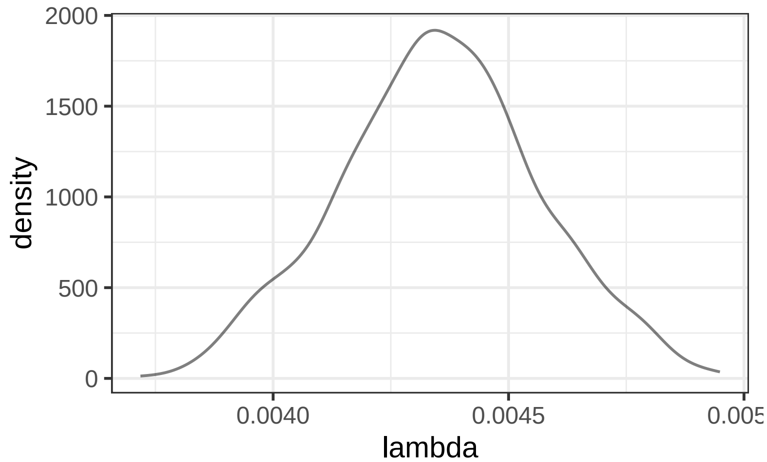
```r
model_descrip_g <- alist(
  phone_use ~ dgamma(alpha, lambda),
  alpha ~ dnorm(mean = .75, sd = .1),
  lambda ~ dnorm(mean = .1, sd = .005)
)

quap_phone_model_g <- quap(flist = model_descrip_g, data = phones)
```

```r
quap_phone_model_g <- quap(flist = model_descrip_g,
                          data = phones)
quap_phone_post_sample_g <- extract.samples(quap_phone_model_g, n = 1000)
```

```r
gf_dens(~alpha, data = quap_phone_post_sample_g)
```




```r
gf_dens(~lambda, data = quap_phone_post_sample_g)
```



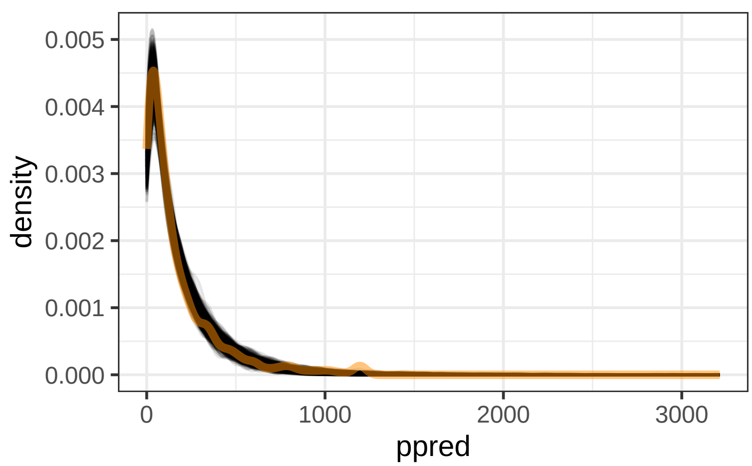
My priors were terrible so I corrected them to be shape ~ norm(mean = .75, sd = 1) and rate ~ norm(mean = 1, sd = 0.005)

D

```
phone_ppred_g <- quap_phone_post_sample_g |>
  # add row numbers to "label" each sampled combo of mu & sigma
  mutate(row_num = c(1:n())) |>
  # work one row (one mu, sigma combination) at a time
  rowwise() |>
  # simulate a dataset for each row (= each mu, sigma combo)
  mutate(ppred = list(rgamma(nrow(phones),
                              shape = alpha,
                              rate = lambda))) |>
  # keep only the row-ids and the simulated data
  select(row_num, ppred)
```

```
phone_ppred_g <- phone_ppred_g |>
  unnest(cols = ppred)
```

```
gf_dens(~ppred, group = ~row_num,
        data = phone_ppred_g,
        alpha = 0.1) |>
  # overlay actual data
  gf_dens(~phone_use,
          data = phones,
          inherit = FALSE,
          color = 'darkorange',
          linewidth = 1.5)
```



The Gamma model looks much better and the dark orange line and simulated data look very close together. For analyzing real data, this is much better than the normal model that was produced.