03 - Density Estimation

Parametric; Gaussian Mixture Models

SYS 4582/6018 | Spring 2019

03-density.pdf

Contents

1 Density Estimation Intro

1.1 Required R Packages

We will be using the R packages of:

- `tidyverse` for data manipulation and visualization
- `fitdistrplus` for parametric estimation
- `mixtools` for mixture modeling

```
library(tidyverse)     # install.packages("tidyverse")
library(fitdistrplus) # install.packages("fitdistrplus")
library(mixtools)      # install.packages("mixtools")
```

1.2 Distributions

For many problems, an optimal decision can be formulated if we know the **distribution** of the random variable. Often, only certain properties of the distribution (expected value, variance, quantiles) are needed to make decisions. Much of statistics is involved with estimation of the distributions or their properties.

Let X be a random variable of interest.

- The **cumulative distribution function (cdf)** is $F(x) = \Pr(X \leq x)$.
- For *discrete* random variables, the **probability mass function (pmf)** is $f(k) = \Pr(X = k)$.
  - $f(k) \geq 0, \sum_k f(k) = 1$
- For *continuous* random variables, the **probability density function (pdf)** is $f(x) = \frac{d}{dx}F(x)$.
  - $f(x) \geq 0, \int_{\infty}^{\infty} f(x) = 1$

A **parametric** distribution, $f(x; \boldsymbol{\theta})$ is one that is fully characterized by a set of parameters, $\boldsymbol{\theta}$. Examples include:

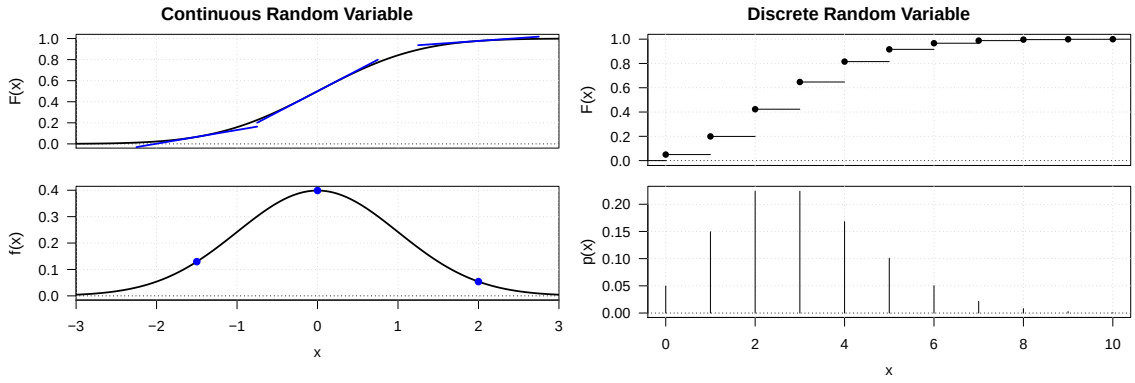
- Normal/Gaussian (mean-μ,standard deviation-σ)
- Poisson (rate-λ)
- Binomial (size-n, probability-p).
- There are also multivariate versions: Gaussian $N(\mu, \Sigma)$.

If we can model (assume) the random variable follows a specific parametric distribution, then we only need to estimate the parameter(s) to have the entire distribution characterized. The parameters are often of direct interest themselves (mean, standard deviation).

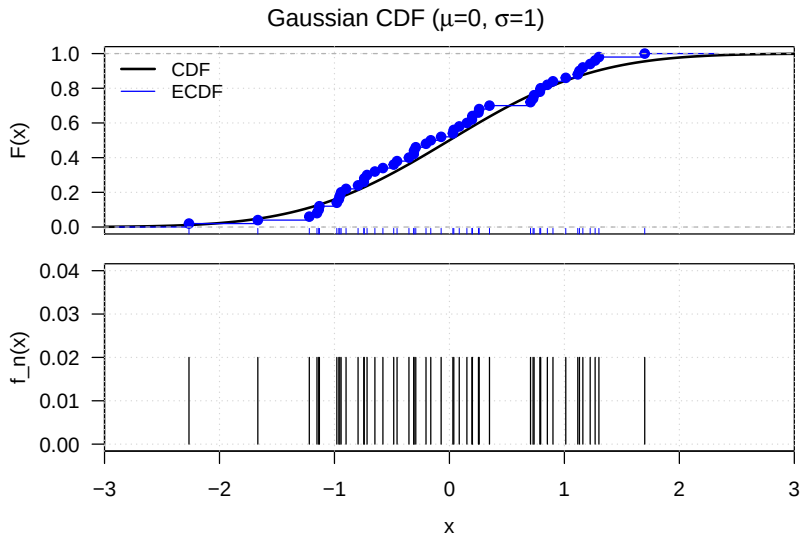
A distribution can also be estimated using **non-parametric** methods (e.g., histograms, kernel methods, splines). These approaches do not enforce a parametric family (which is essentially a type of prior knowledge), but let the data determine the shape of the density/pmf/cdf. As you might imagine more data is required for these methods to work well. Non-parametric approaches are excellent for exploratory data analysis, but can also be very useful for other types of modeling (e.g., classification, anomaly detection).

Everything would be easy if we knew the exact distributions of the random variables of interest. Unfortunately, this is usually never the case (but of course: flipping coins, drawing cards, and playing with urns is different). We must use data to estimate the aspects/parameters of a distribution necessary to make good decisions. And it is important to be mindful of the resulting uncertainty (bias, variance) in our estimation.

1.2.1 Random Variables



1.2.2 Empirical CDF and PDF



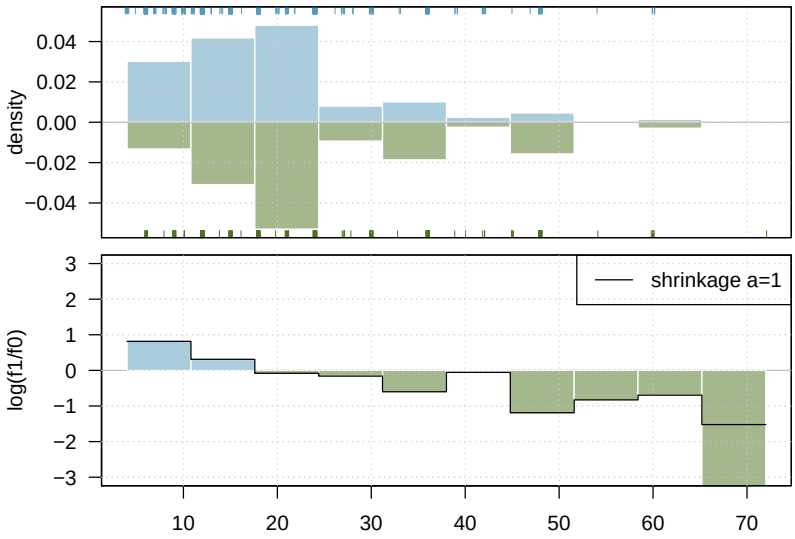
Data: $n = 50$ from $X \sim N(0, 1)$

1.3 Example: Default Classification

Density estimation can be useful in *classification problems*, where the goal is to determine which class a new observation belongs to.

Below are two *histogram* density estimates; one for customers of a German bank that have good credit (blue) and the other for customers who defaulted (green). If a new customer is observed to have $X = 5$, then the evidence favors them having good credit because $X = 5$ is more likely under customers with good credit.

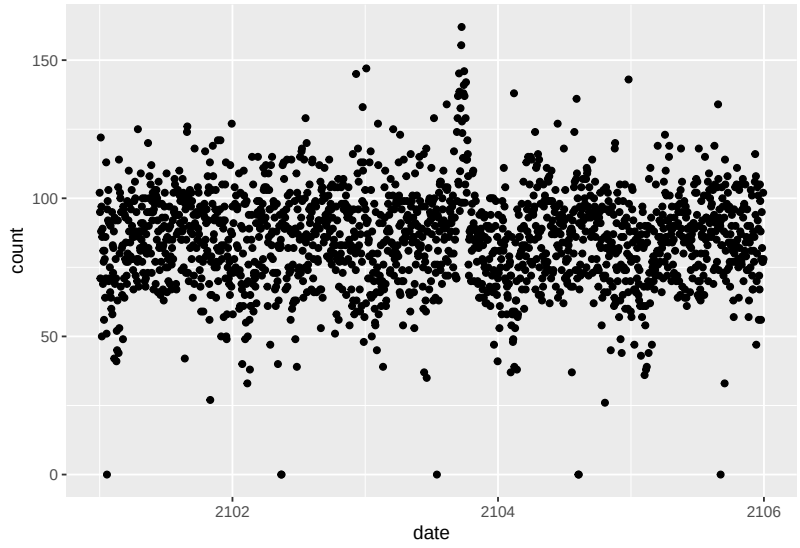
The bottom plot shows the corresponding log density ratio, which can help the bank make a decision on the customer's credit-worthiness.



1.4 Example: Disease Outbreak Detection

Density estimation can be useful in *anomaly detection systems*, where the goal is to (often quickly) determine the time point when observations starting coming from a new or different distribution.

Below is simulated disease outbreak data representing the number of cases of some reported symptoms in an Emergency Department. If we can estimate the distribution of the baseline, or normal counts, on each day, then we will be able to flag an anomaly whenever the observations become *unlikely*.



2 Parametric Density Estimation

2.1 Method of Moments Estimation (MOM)

- Let X be a random variable with *pdf/pmf* $f(x;\theta)$ parameterized by $\theta \in \Theta$.
- Let $D = \{X_1, X_2, \ldots, X_n\}$ be the observed data.
- *Method of Moments (MOM)* estimators match the sample moments to the theoretical moments
  - This works when the parameter(s) can be written as functions of the moments.
  - To estimate p parameters, use p moments.

- 1st moments
  - The 1st *theoretical* moment $E[X]$ is the mean.
  - The 1st *sample* moment is the sample mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} X_i$
  - If $g_1(\theta) = E[X]$, then set $g_1(\theta_{\text{MM}}) = \bar{x}$ and solve for θ_{MM}.
- 2nd (central) moments
  - The 2nd *theoretical* central moment $V(X) = E[(X-\mu)^2]$ is the variance.
  - The 2nd *sample* central moment is the sample variance $\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$.
  - If $g_2(\theta) = V(X)$, then set $g_2(\theta_{\text{MM}}) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$ and solve for θ_{MM}.
- Maximum Likelihood estimation usually produces a *better* estimate, so we will focus on the MLE approach.

2.2 Maximum Likelihood Estimation (MLE)

- Let X be a random variable with *pdf/pmf* $f(x;\theta)$ parameterized by $\theta \in \Theta$.
- Let $D = \{X_1, X_2, \ldots, X_n\}$ be the observed data.
- *Maximum Likelihood Estimation (MLE)* uses the value of θ that, well, maximizes the *likelihood*:

$$L(\theta) = P(X_1, X_2, \ldots, X_n; \theta)$$

  - written as a function of θ treating the observed data as known
- When the observations are *independent*, this becomes:

$$L(\theta) = \prod_{i=1}^{n} f(x_i;\theta)$$

- With log-likelihood of

$$\log L(\theta) = \sum_{i=1}^{n} \log f(x_i;\theta)$$

- The MLE becomes:

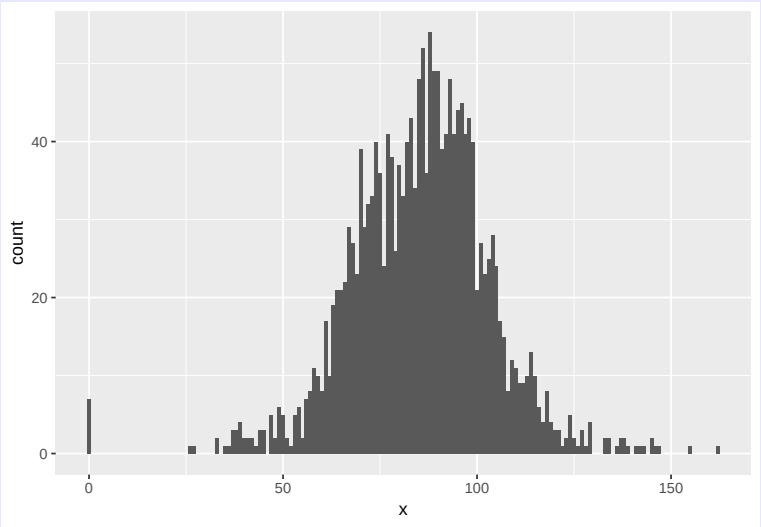
$$\begin{aligned}\theta_{\text{MLE}} &= \arg\max_{\theta\in\Theta} L(\theta) \\ &= \arg\max_{\theta\in\Theta} \log L(\theta)\end{aligned}$$

Your Turn #1

Estimate the baseline density of ED counts.

```
#-- Load Data
url = 'https://raw.githubusercontent.com/mdporter/SYS6018/master/data/ED-counts.csv'
x = readr::read_csv(url)$count

#-- empirical pmf
ggplot() + geom_bar(aes(x=x))
```



1. Use a Poisson model.
2. Use a Negative Binomial model.
3. Use a Gaussian model.
4. Based on our models, what is the probability that we would get more than > 150 or < 50 counts on a *regular* day?

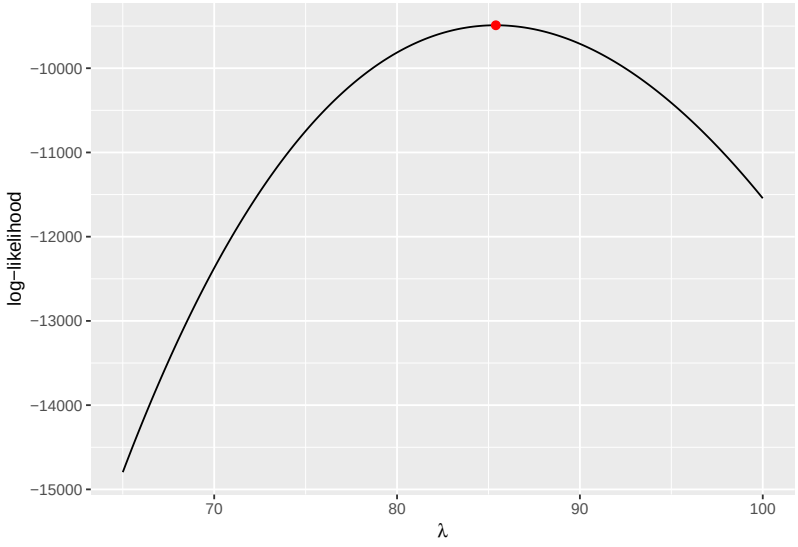
Note: Distribution Reference Sheet

2.2.1 Example: Poisson MLE

- Notation:
  - $X \sim Pois(\lambda)$
  - $\Pr(X = x) = f(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$ where $f(x)$ is a pmf and $x = \{0, 1, \ldots\}$.
  - $E[X] = V[X] = \lambda$

Grid Search:

- Calculate the log-likelihood over a range of λ values and choose the one that gives the maximum.

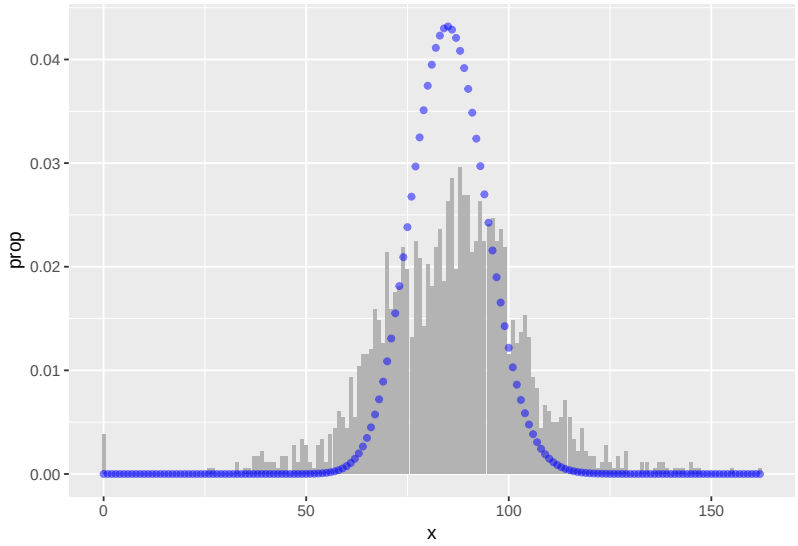


- The grid search gives $\hat{\lambda} = 85.402$
  - Searched 200 values between 65 and 100.

Use Calculus

Your Turn #2

Estimated pmf using Poisson MLE



2.2.2 Example: Negative Binomial

- The Negative Binomial distribution can help for modeling data with overdispersion
- Notation:
  - $X \sim NBin(\mu, r)$ (using *mean* paramaterization)
  - $\mu > 0, r > 0$
  - $E[X] = \mu, V[X] = \mu + \mu^2/r$

- Mean representation: https://en.wikipedia.org/wiki/Negative_binomial_distribution

$$\Pr(X = x; r, \mu) = \frac{\Gamma(r+x)}{x!\Gamma(r)} \left(\frac{r}{r+\mu}\right)^r \left(\frac{\mu}{r+\mu}\right)^x$$

- $n! = \Gamma(n-1)$
- $\Gamma(n) = \int_0^\infty e^{-x} x^{n-1} dx$

Your Turn #3

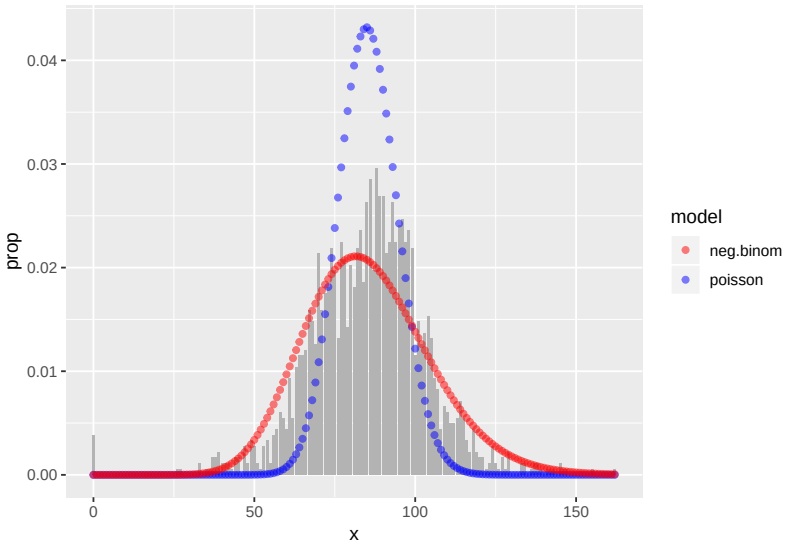
Use Method of Moments to estimate $\theta = (r, \mu)$.

Your Turn #4

Use Maximum likelihood to estimate $\theta = (r, \mu)$.

- Use R `fitdistrextra` package.

```
library(fitdistrplus)
opt = fitdist(data=x, distr="nbinom", method="mle")
nb.pars = opt$estimate
```



2.2.3 Example: Gaussian/Normal

- Data are non-negative integers, not continuous, so Gaussian is clearly "wrong". But as the famous saying goes:

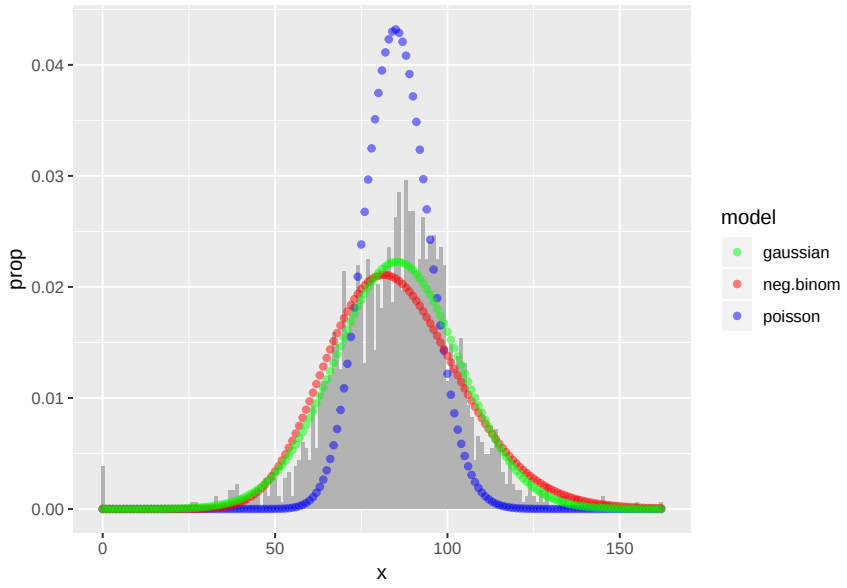
  "All models are wrong, but some are useful". - George E. P. Box

- Notation:
  - $X \sim N(\mu, \sigma)$
  - $\mu \in \mathbf{R}, \sigma > 0$
  - $E[X] = \mu, V[X] = \sigma^2$

Your Turn #5

Find the MLE for (μ, σ).

2.2.4 Comparison of Models



- Models:
  1. Poisson: $\lambda = 85.3817$
  2. Neg.Binom: $r = 25.6266, \mu = 85.3857$

3. Gaussian: $\mu = 85.3817, \sigma = 17.919$

Your Turn #6

Which model do you choose? Why?

2.3 Bayesian Estimation

In Bayesian analysis, the parameter(s) are *random variables*.

- In MOM and MLE, the parameters are assumed fixed, but unknown.

Prior knowledge, any information known about the parameter(s) *before the data are seen*, is captured in the *prior distribuation*.

- Let $g(\theta)$ be the (possibly multivariate) prior pmf/pdf

Bayes theory gives us the *posterior distribution*,

$$f(\theta|D) = \frac{P(D|\theta)g(\theta)}{\int_{\theta\in\Theta} P(D|\theta)g(\theta)\,\mathrm{d}\theta}$$

- $P(D|\theta) = P(X_1, X_2, \ldots, X_n) =$ *likelihood*
- $\int_{\theta\in\Theta} P(D|\theta)g(\theta)\,\mathrm{d}\theta = P(D)$ is the *normalizing constant* (not function of θ).
- $P(\theta|D)$ is the *posterior distribution*, which contains the updated knowledge about the parameter(s).

2.3.1 Point Estimation

1. Posterior Mean

$$\hat{\theta}_{\mathrm{PM}} = E[\theta|D] = \int_{\theta\in\Theta} \theta f(\theta|D)\,\mathrm{d}\theta$$

2. MAP (Maximum a posteriori)

$$\begin{aligned}\hat{\theta}_{\mathrm{MAP}} &= \arg\max_{\theta\in\Theta} f(\theta|D) \\ &= \arg\max_{\theta\in\Theta} P(D|\theta)g(\theta) \\ &= \arg\max_{\theta\in\Theta} \left(\log P(D|\theta) + \log g(\theta)\right)\end{aligned}$$

3 Mixture Models

3.1 Example: Old Faithful

The old faithful geyser in Yellowstone National Park is one of the most regular geysers in the park. The waiting time between eruptions is between 35 and 120 mins.

Live Streaming Webcam with eruption predictions

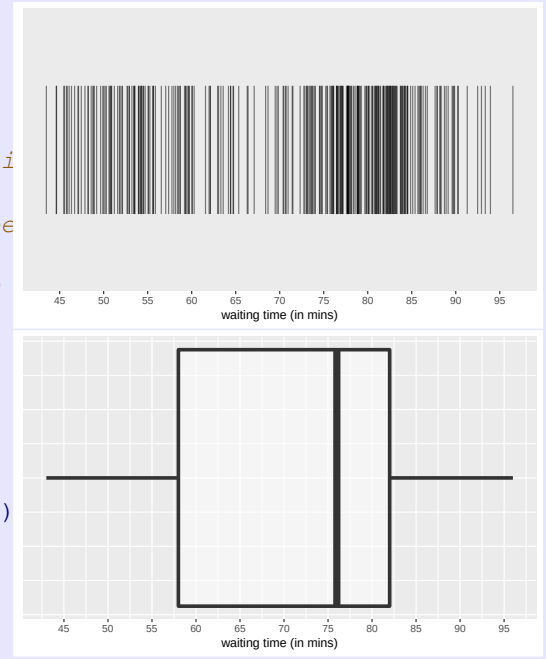
Because the nearby Yellowstone Lodge is nice and warm in the winter, and serves good ice cream in the summer, you may be distracted from stepping outside to watch the eruption. Let's see if we can determine the best time to go out and watch.

Your Turn #7 : Old Faithful

The data, summary statistics, and plots below represent a sample of *waiting times*, the time (in min) between Old Faithful eruptions.

```
#-- Load the Old Faithful data
wait = datasets::faithful$waiting

#-- Calculate summary stats
length(wait)              # sample si
#> [1] 272
summary(wait)             # six numbe
#>    Min. 1st Qu.  Median    Mean
#>    43.0    58.0    76.0    70.9
mean(wait)                # mean
#> [1] 70.9
sd(wait)
#> [1] 13.59
median(wait)
#> [1] 76
quantile(wait, probs=c(.25,.50,.75))
#> 25% 50% 75%
#>  58  76  82
```

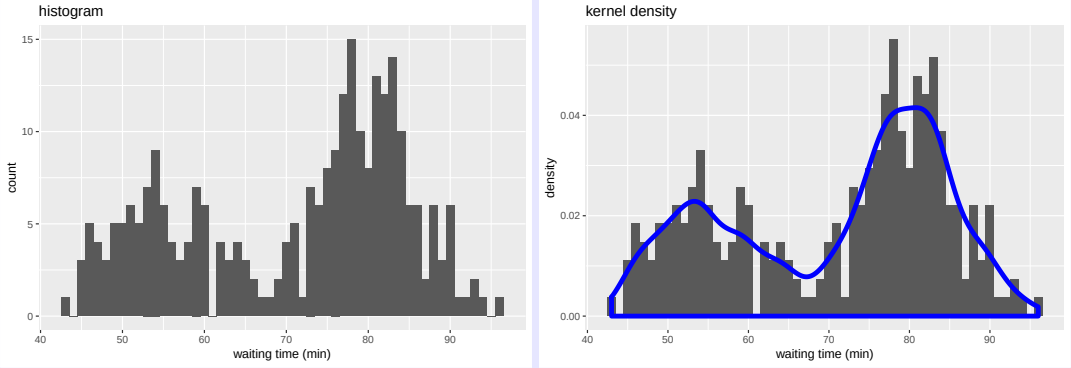



```
#-- Put data into a data.frame/tibble for use with ggplot
wait.df = tibble(wait)

#-- Make a ggplot object
pp = ggplot(wait.df, aes(x=wait)) + xlab("waiting time (min)")

#-- Histogram
pp + geom_histogram(binwidth = 1) + ggtitle("histogram")

#-- overlay kernel density plot
pp + geom_histogram(binwidth = 1, aes(y=stat(density))) +  # *density* histogram
  geom_density(bw=2, size=2, color="blue") + ggtitle("kernel density")
```



1. What can you say about the shape of the distribution?
2. Would a Gaussian (i.e., Normal) Distribution be a good choice for modeling the distribution of these data?
3. What would you recommend?

3.2 Finite Mixture Models

Mixture models combine several parametric models to produce a more complex, yet easy to interpret distribution.

- natural representation when data come from different clusters/groups

$$f(x) = \sum_{k=1}^{K} \pi_k \, f_k(x)$$

- $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{K} \pi_k = 1$
- $f_k(x)$ is a parametric pdf/pmf

3.3 Univariate Gaussian Mixture Model

Consider a two-component ($K = 2$) mixture of Gaussian distributions:

$$\begin{aligned} f(x;\theta) &= \pi f_1(x;\theta_1) + (1-\pi) f_2(x;\theta_2) \\ &= \pi \mathcal{N}(x;\mu_1,\sigma_1) + (1-\pi)\mathcal{N}(x;\mu_2,\sigma_2) \end{aligned}$$

- $0 \leq \pi \leq 1$
- $\theta = (\pi, \mu_1, \sigma_1, \mu_2, \sigma_2)$

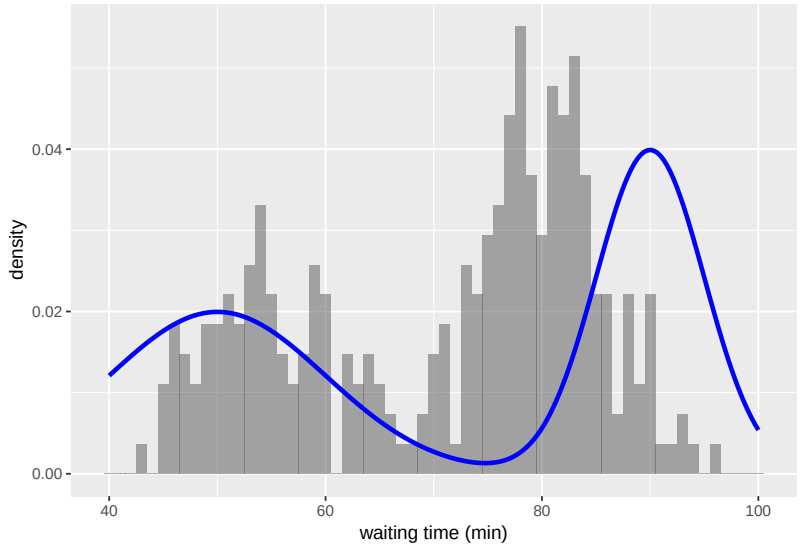
3.3.1 Example: Old Faithful

```
#-- Function to calculate Gaussian mixture pdf
dnmix <- function(theta1, theta2, w=.5, x.seq=seq(-4, 4, length=100)){
  f1 = dnorm(x.seq, mean=theta1[1], sd=theta1[2])
  f2 = dnorm(x.seq, mean=theta2[1], sd=theta2[2])
  fmix = f1*w + f2*(1-w)
  return(fmix)
}

#-- Set parameters
theta1 = c(mu=50, sigma=10)        # parameters for component 1
theta2 = c(mu=90, sigma=5)         # parameters for component 2
w = .5                             # mixture weight

#-- Make data for plotting
x.seq = seq(40, 100, length=200)
f = dnmix(theta1, theta2, w, x.seq)
data.mix = tibble(x.seq, f)

#-- Make plot
pp + geom_histogram(binwidth = 1, aes(y=stat(density)), alpha=.5) +
  geom_line(data=data.mix, aes(x=x.seq, y=f), color="blue", size=1.25)
```



Your Turn #8

What parameters do you suggest? Modify the code above to help you decide.

3.4 EM Algorithm

The details are found in the assigned reading Gaussian Mixture Models: 11.1-11.3, so we will just cover the basics.

Notation:

- Let $g_i \in \{1, 2, \ldots, K\}$ be the (unknown) group/component identifier.
  - π_k is prior probability that any observation is from component k
- The data $D = \{X_1, X_2, \ldots, X_n\}$
- The **responsibilities** are the posterior probability that event i came from component k:

$$\begin{aligned} r_{ik} &= \Pr(g_i = k | D, \theta) \\ &= \frac{P(D|g_i = k, \theta_k)\pi_k}{\sum_{j=1}^{K} P(D|g_i = j, \theta_j)\pi_j} \end{aligned}$$

- The responsibilities are weights: $\sum_{k=1}^{K} r_{ik} = 1 \; \forall i$, which represent the probability that events come from the components, conditional on the parameters θ.
- *EM* stands for *Expectation-Maximization*
  - *E-step*: calculate the responsibilities
  - *M-step*: estimate parameters using new responsibilities as weights
  - Iterate until convergence

3.4.1 Algorithm

1. Initiate θ
2. Repeat until convergence:
   - E-step: update r_{ik}, using θ, for $i = 1, 2, \ldots, n$ and $k = 1, \ldots, K$.
   - M-step: update θ using r_{ik}

For Gaussian components:

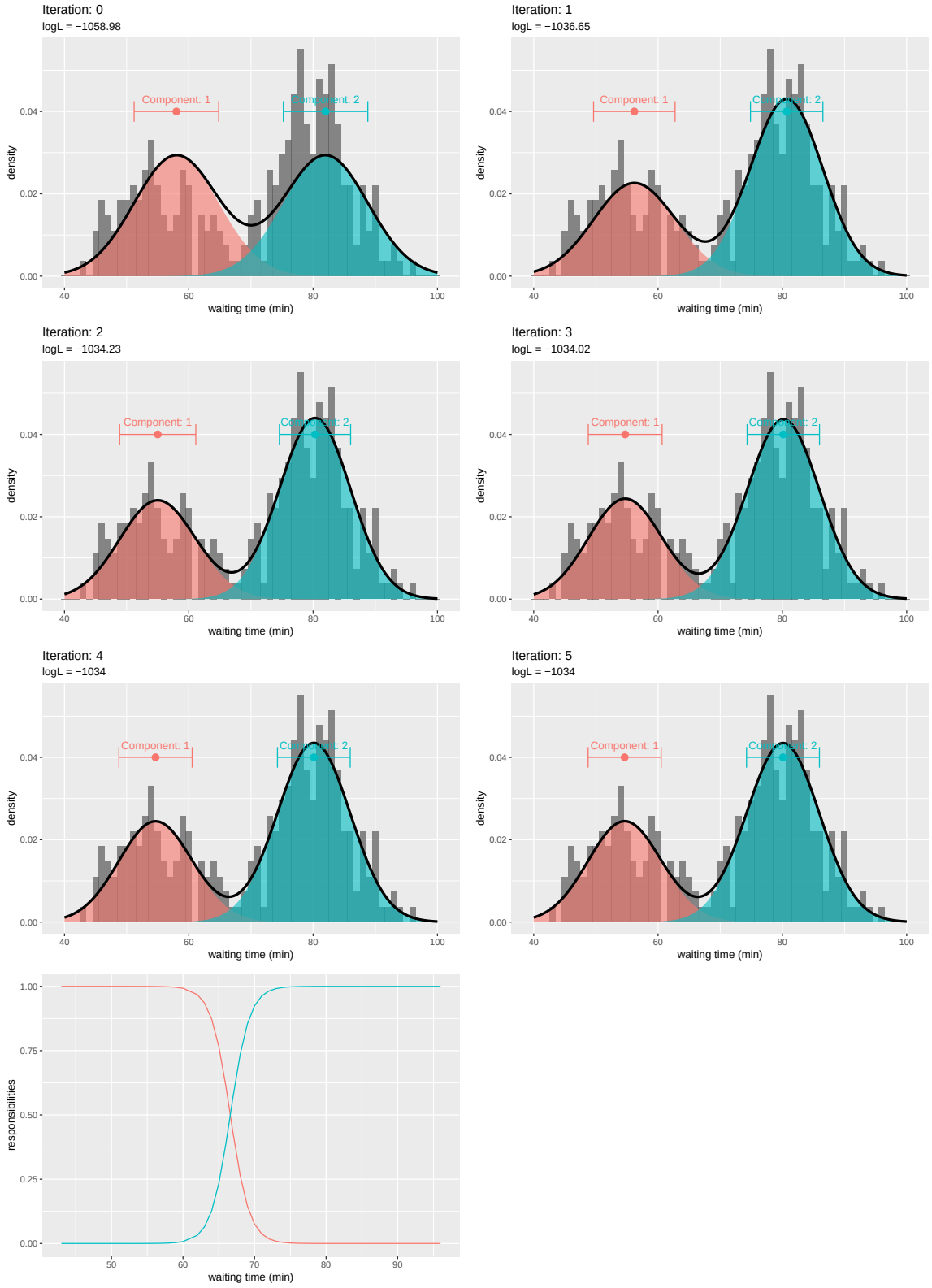
- Estimate like usual, except with *weighted* observations:

$$\begin{aligned} n_k &= \sum_{i=1}^{n} r_{ik} \\ \pi_k &= n_k / n \\ \mu_k &= \frac{1}{n_k} \sum_{i=1}^{n} r_{ik} x_i \\ \sigma_k &= \frac{1}{n_k} \sum_{i=1}^{n} r_{ik} (x_i - \mu_k)^2 \end{aligned}$$

3.4.2 R package `mixtools`

```
library(mixtools)
gauss_mix = normalmixEM(wait, k=2)  # 2 component gaussian mixture
#> number of iterations= 30

(w = gauss_mix$lambda)        # prior probabilities (pi)
#> [1] 0.3609 0.6391
(mu = gauss_mix$mu)           # component means
#> [1] 54.61 80.09
(sigma = gauss_mix$sigma)     # component standard deviations
#> [1] 5.871 5.868
r = gauss_mix$posterior       # responsibiliites matrix
```


Iteration: 0
logL = −1058.98
Component: 1
Component: 2
density
waiting time (min)
Iteration: 1
logL = −1036.65
Component: 1
Component: 2
density
waiting time (min)
Iteration: 2
logL = −1034.23
Component: 1
density
waiting time (min)
Iteration: 3
logL = −1034.02
Component: 1
density
waiting time (min)
Iteration: 4
logL = −1034
Component: 1
density
waiting time (min)
Iteration: 5
logL = −1034
Component: 1
density
waiting time (min)
responsibilities
waiting time (min)