ECS 132 Term Project

Steven Alvarado, Russell Chien, and Ruth Hailu

University of California, Davis

June 2023

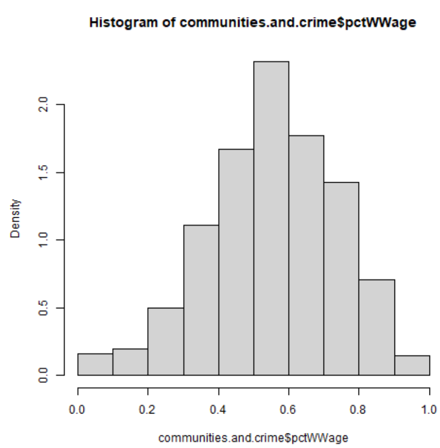
Chapter 1

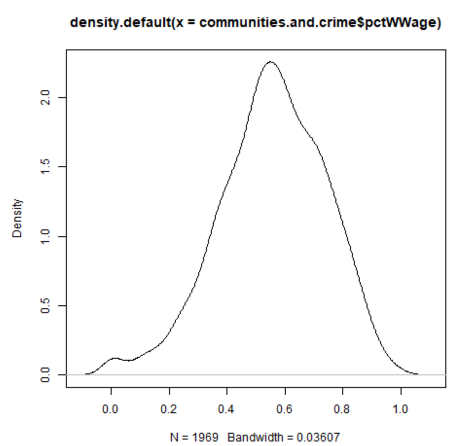
The Normal Family

1.1 Communities and Crime: pctWWage

Our group observed that the variable **pctWWage** of the Communities and Crime dataset seemed well-approximated by the normal family of continuous distributions. According to the UCI Machine Learning Repository, **pctWWage** is described as the percentage of households within the United States with wage or salary income in 1989.

1.2 Histogram and Density





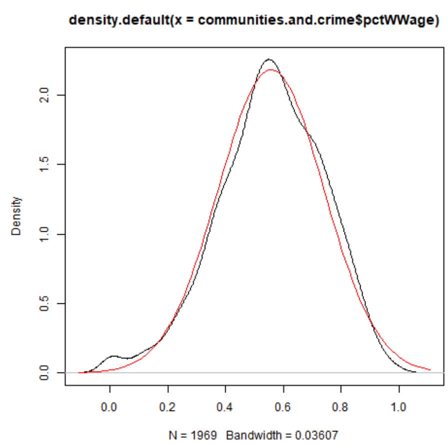
1.3 MLE and MM

To find the MLE of the normal family, we first defined our log likelihood function:

$$L(\mu, \sigma^2) = -n \log(2\pi) + \frac{\log(\sigma^2)}{2} - \frac{\sum(x-\mu)^2}{2\sigma^2} \tag{1.1}$$

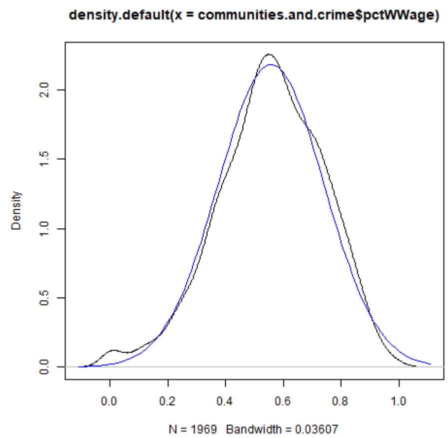
To find the MLE, we use R's built in mle() function with the negative log likelihood function.

```
z <- mle(minuslogl = ll, start = c(list(mean = 1),
   list(var = 1)))
```



To find the MM of the normal family, we used the following function to predict μ and σ^2.

```
  mm <- function(x) {
    mu <- mean(x)
    sigma <- sqrt(mean(x^2) - mu^2)
    return(c(mu, sigma))
  }
```



1.4 Conclusion

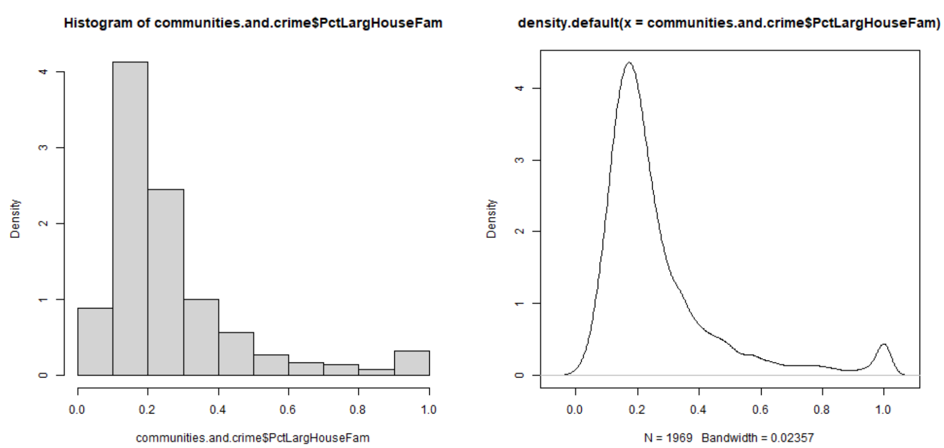
Chapter 2

The Exponential Family

2.1 Communities and Crime: PctLargHouseFam

For the exponential family of continuous distributions, we observed that the variable **PctLargHouseFam** was a suitable approximation. According to the UCI Machine Learning Repository, **PctLargHouseFam** is described as the percentage of family households with six or more family members.

2.2 Histogram and Density



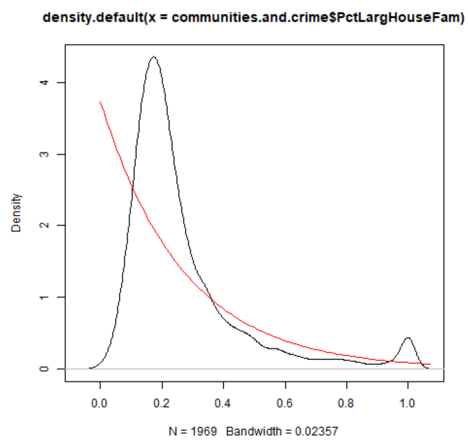
2.3 MLE and MM

To find the MLE of the exponential family, we first defined our log likelihood function:

$$L(\lambda) = n \log \lambda - \lambda \sum x \tag{2.1}$$

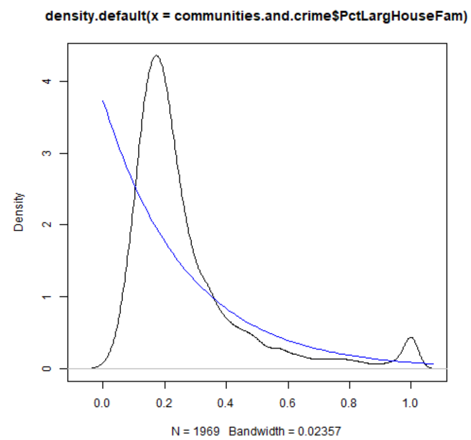
To find the MLE, we use R's built in mle() function with the negative log likelihood function.

```
z <- mle(minuslogl = ll, start = c(list(lambda =
   1)))
```



To find the MM of the exponential family, we used the following function to predict the value of λ:

```
mm <- function(x) {
  lambda <- 1 / mean(x)
  lambda
}
```



2.4 Conclusion

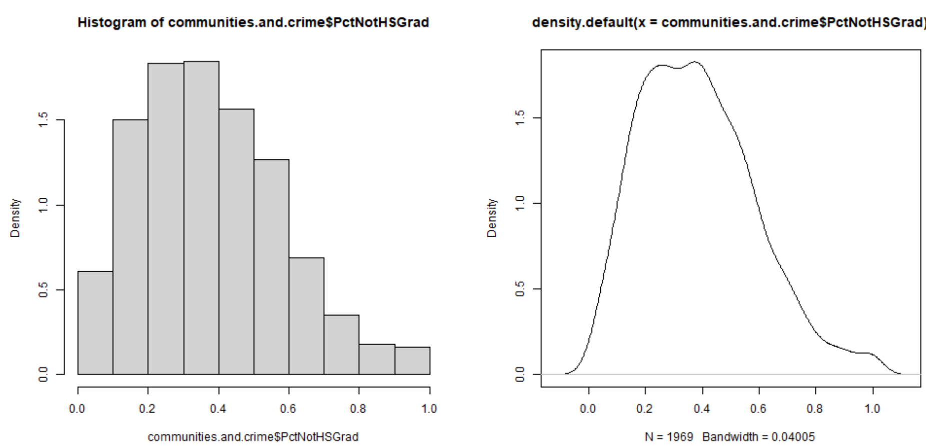
Chapter 3

The Gamma Family

3.1 Communities and Crime: PctNotHsGrad

We observed that the variable **PctNotHsGrad** of the Communities and Crime dataset seemed well-approximated by the gamma family of continuous distributions. According to the UCI Machine Learning Repository, **PctNotHsGrad** is described as the percentage of people 25 and over that are not high school graduates.

3.2 Histogram and Density



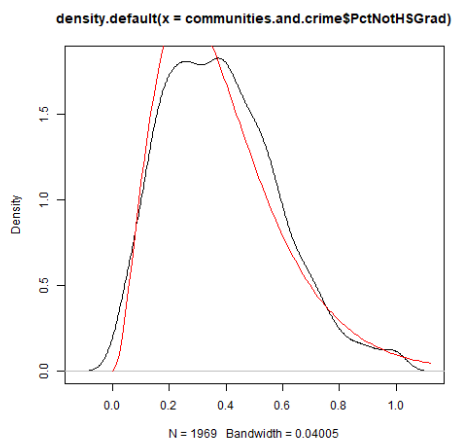
3.3 MLE and MM

To find the MLE of the gamma family, we first defined our log likelihood function:

$$L(k, \theta) = (k-1)\sum(\log x) - \sum(\frac{x}{\theta}) - nk\log(\theta) - n\log(\Gamma(k)) \tag{3.1}$$

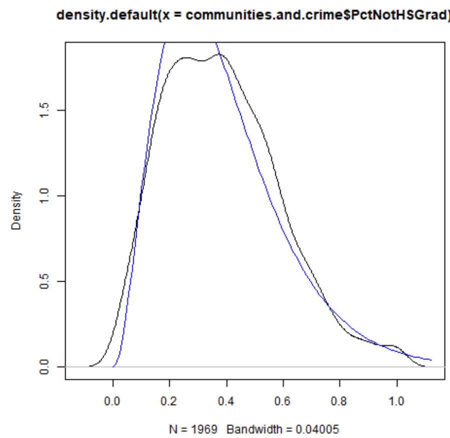
To find the MLE, we use R's built in mle() function with the negative log likelihood function.

```
z <- mle(minuslogl = ll, start = c(list(k = 1),
   list(theta = 1)))
```



To find the MM of the gamma family, we used the following function to predict the value of k and θ:

```
mm <- function(x) {
  mu <- mean(x)
  theta <- mean(x * log(x)) - mu * mean(log(x))
  k <- mu / theta
  return(c(k, theta))
}
```



3.4 Conclusion

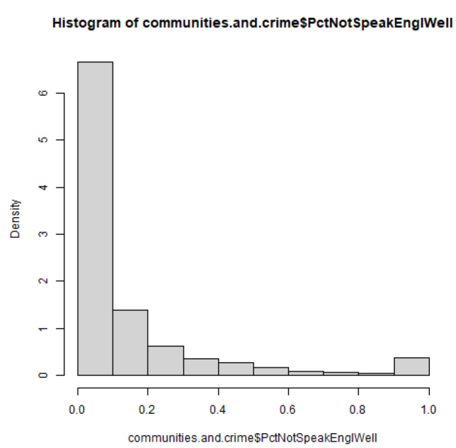
Chapter 4

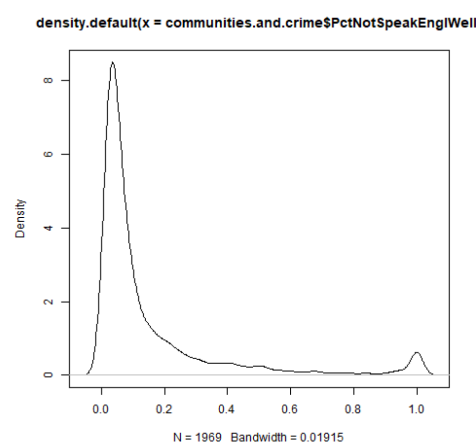
The Beta Family

4.1 Communities and Crime: PctNotSpeakEnglWell

For the beta family of continuous distributions, we observed that the variable **PctNotSpeakEnglWell** was a suitable approximation. According to the UCI Machine Learning Repository, **PctNotSpeakEnglWell** is described as the percentage of people who do not speak English well.

4.2 Histogram and Density





4.3 MLE and MM

To find the MLE of the beta family, we first had to scale our data so it was within the support of $(0, 1)$:

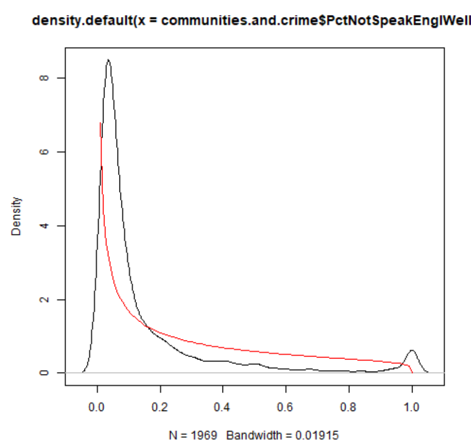
```
x[which(x == 0)] <- 0.0001
x[which(x == 1)] <- 0.9999
```

We then found the log likelihood function:

$$L(\alpha, \beta) = n\log\left(\Gamma(\alpha+\beta)\right) - n\log\left(\Gamma(\alpha)\right) - n\log\left(\Gamma(\beta)\right) + (\alpha-1)\sum(\log(x)) + (\beta-1)\sum(\log(1-x)) \quad (4.1)$$

To find the MLE, we use R's built in mle() function with the negative log likelihood function.

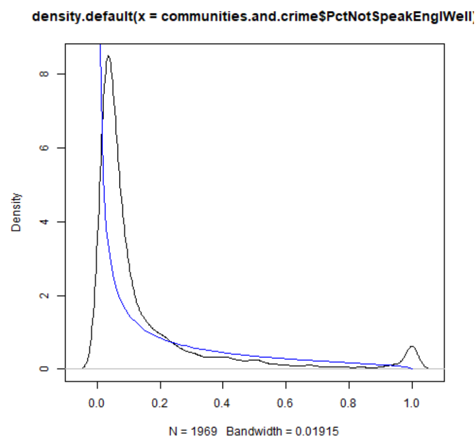
```
z <- mle(minuslogl = ll, start = c(list(alpha = 1),
  list(beta = 1)))
```



To find the MM of the beta family, we used the following function to estimate the α and β values:

```
mm <- function(x) {
  mu <- mean(x)
  var <- var(x)
  alpha <- mu * (mu * (1 - mu) / var - 1)
```

```
    beta <- (1 - mu) * (mu * (1 - mu) / var - 1)
    return(c(alpha, beta))
}
```



4.4 Conclusion