1651 Final

Max Bauer

2024-04-21

1)

```r
# Read data from Bat.dat file
bat_data <- read.table("Bat.dat", header = TRUE)

# Filter out players with at-bats >= 10 in either period
filtered_data <- bat_data[bat_data$AB.1 > 10 & bat_data$AB.2 > 10, ]

# Count the remaining players
n <- nrow(filtered_data)

# Print the number of players remaining
print(n)
```

```
## [1] 491
```

There are 491 players left in the analysis

2)

```r
# Define function to update Beta parameters
update_beta_params <- function(alpha, beta, H, N) {
  alpha_new <- alpha + H
  beta_new <- beta + N - H
  return(list(alpha = alpha_new, beta = beta_new))
}

# Define function to estimate pi
estimate_pi <- function(alpha, beta, H, N) {
  return((alpha + H) / (alpha + beta + N))
}

# Set prior parameters
alpha_prior <- 1
beta_prior <- 1
```

```r
# Update prior with data from the first period
filtered_data <- bat_data[bat_data$AB.1 > 10 & bat_data$AB.2 > 10, ]
n <- nrow(filtered_data)

posterior_alpha <- alpha_prior + filtered_data$H.1
posterior_beta <- beta_prior + filtered_data$AB.1 - filtered_data$H.1

# Estimate pi
estimated_pi <- estimate_pi(posterior_alpha, posterior_beta, filtered_data$H.1, filtered_data$AB.1)

# Print estimated pi
print(estimated_pi)
```

```
##   [1] 0.28947368 0.18229167 0.15445026 0.08823529 0.17632850 0.20138889
##   [7] 0.20575221 0.21500000 0.21078431 0.28723404 0.09375000 0.22820513
##  [13] 0.22701149 0.12187500 0.15517241 0.18103448 0.23033708 0.20050761
##  [19] 0.02777778 0.19690265 0.16046512 0.19021739 0.20238095 0.07812500
##  [25] 0.15543071 0.14730290 0.20445344 0.17210145 0.18327402 0.11875000
##  [31] 0.13888889 0.16863905 0.15522876 0.25403226 0.19281046 0.17553191
##  [37] 0.14705882 0.18429487 0.08208955 0.10869565 0.15945946 0.13404255
##  [43] 0.16231343 0.18000000 0.16137566 0.16836735 0.13429752 0.18669528
##  [49] 0.12162162 0.15885417 0.23863636 0.07575758 0.19349315 0.18339768
##  [55] 0.19247788 0.13362069 0.14000000 0.22431507 0.16098485 0.10606061
##  [61] 0.09687500 0.17119565 0.20848708 0.10606061 0.04166667 0.11073826
##  [67] 0.19190141 0.16666667 0.16920152 0.16233766 0.11025641 0.25735294
##  [73] 0.13398693 0.21428571 0.20256410 0.11904762 0.07758621 0.18790850
##  [79] 0.12337662 0.14824121 0.18322981 0.20312500 0.14900662 0.17801858
##  [85] 0.23437500 0.06818182 0.16071429 0.15000000 0.13671875 0.15196078
##  [91] 0.14028777 0.06944444 0.29807692 0.11458333 0.16958042 0.18141593
##  [97] 0.19057377 0.22286822 0.31904762 0.19230769 0.15040650 0.15441176
## [103] 0.11858974 0.14377682 0.21590909 0.20068027 0.10000000 0.20535714
## [109] 0.19175627 0.17892157 0.15476190 0.13265306 0.13076923 0.18800000
## [115] 0.02500000 0.12857143 0.11538462 0.22321429 0.14130435 0.16798419
## [121] 0.17932489 0.17049808 0.09689922 0.10606061 0.15924658 0.16515837
## [127] 0.18421053 0.18750000 0.19162996 0.17300380 0.17248062 0.21061093
## [133] 0.16741071 0.16283525 0.19866071 0.22151899 0.19257951 0.24074074
## [139] 0.31481481 0.19966997 0.12240664 0.09482759 0.21535581 0.05769231
## [145] 0.17469880 0.12068966 0.20078740 0.13461538 0.15245902 0.31060606
## [151] 0.16346154 0.35714286 0.15714286 0.18611111 0.15145228 0.16847826
## [157] 0.15000000 0.18703704 0.09545455 0.17321429 0.06000000 0.17857143
## [163] 0.27702703 0.20258621 0.20270270 0.19666667 0.16926070 0.20659722
## [169] 0.15878378 0.16995074 0.16233766 0.19705882 0.14402174 0.18512111
## [175] 0.14593301 0.12626263 0.17086331 0.17338710 0.21153846 0.17615658
## [181] 0.29525862 0.20270270 0.19473684 0.14705882 0.21516393 0.14804469
## [187] 0.17173913 0.15700483 0.01785714 0.17741935 0.04411765 0.10937500
## [193] 0.19696970 0.15671642 0.17584746 0.19543147 0.16666667 0.17222222
## [199] 0.17982456 0.08974359 0.15400844 0.10652174 0.23106061 0.19620253
## [205] 0.12264151 0.21644295 0.14024390 0.24722222 0.17005076 0.10576923
## [211] 0.01724138 0.26923077 0.04385965 0.16851852 0.05555556 0.18000000
## [217] 0.18498168 0.11904762 0.20172414 0.18452381 0.14319249 0.18750000
## [223] 0.14215686 0.32500000 0.14000000 0.13759690 0.12500000 0.21217105
## [229] 0.25000000 0.24479167 0.18093385 0.13563830 0.18512111 0.15605096
## [235] 0.18871595 0.17067308 0.10818713 0.18430657 0.22590361 0.23226950
```

```
## [241] 0.15829146 0.18235294 0.16787004 0.15666667 0.18993506 0.10064935
## [247] 0.20041322 0.18037975 0.16136364 0.16761364 0.01724138 0.18262411
## [253] 0.20762712 0.11923077 0.16282895 0.27663230 0.19722222 0.12962963
## [259] 0.07352941 0.16421569 0.19444444 0.16666667 0.22469636 0.31418919
## [265] 0.17792793 0.20990566 0.17137097 0.16871166 0.26190476 0.18944099
## [271] 0.10937500 0.15310078 0.27083333 0.19193548 0.20588235 0.10384615
## [277] 0.15671642 0.19162996 0.10937500 0.12337662 0.24358974 0.16666667
## [283] 0.20833333 0.09210526 0.08208955 0.11781609 0.17984190 0.15931373
## [289] 0.23161765 0.20648464 0.15405405 0.16666667 0.19285714 0.38888889
## [295] 0.17803030 0.17307692 0.21604938 0.13709677 0.07777778 0.20600000
## [301] 0.12500000 0.21474359 0.15099010 0.16887417 0.18085106 0.18956044
## [307] 0.01851852 0.17924528 0.07142857 0.15566038 0.08677686 0.20344828
## [313] 0.10747664 0.15042373 0.19201521 0.18525180 0.17233010 0.09259259
## [319] 0.35465116 0.17763158 0.01666667 0.01851852 0.20198675 0.14534884
## [325] 0.12608696 0.18581081 0.15404040 0.17123288 0.19303797 0.15277778
## [331] 0.22500000 0.03571429 0.25000000 0.10087719 0.23888889 0.18686869
## [337] 0.11363636 0.10294118 0.18669528 0.22602740 0.03571429 0.14367816
## [343] 0.17315175 0.14646465 0.07894737 0.17716535 0.10416667 0.08984375
## [349] 0.26587302 0.06521739 0.17015707 0.20833333 0.22941176 0.17657343
## [355] 0.13380282 0.10240964 0.10869565 0.16136364 0.17658730 0.31666667
## [361] 0.18196721 0.17441860 0.16533865 0.22103004 0.16452991 0.10000000
## [367] 0.18421053 0.22019868 0.10080645 0.08593750 0.08823529 0.20036101
## [373] 0.19117647 0.19597070 0.19288390 0.04687500 0.15909091 0.12025316
## [379] 0.14717742 0.16193182 0.15371622 0.12916667 0.20138889 0.16567164
## [385] 0.19260700 0.17741935 0.17175573 0.27941176 0.22482014 0.14532020
## [391] 0.22696246 0.21003717 0.28461538 0.19230769 0.20629371 0.18222892
## [397] 0.21153846 0.16666667 0.14077670 0.16727941 0.06818182 0.26000000
## [403] 0.18627451 0.12937063 0.19155844 0.18122271 0.08928571 0.18750000
## [409] 0.16250000 0.17741935 0.20833333 0.15543071 0.30952381 0.02631579
## [415] 0.19964029 0.32666667 0.21875000 0.20175439 0.14473684 0.19461078
## [421] 0.14473684 0.18137255 0.20562771 0.11538462 0.16735537 0.15833333
## [427] 0.17483660 0.26785714 0.20992366 0.13461538 0.13945578 0.19598765
## [433] 0.21134021 0.16511628 0.21356784 0.22549020 0.19611307 0.15193370
## [439] 0.18454259 0.21634615 0.07746479 0.05232558 0.22321429 0.12500000
## [445] 0.10937500 0.20138889 0.21914894 0.13750000 0.20220588 0.15509259
## [451] 0.20135747 0.18750000 0.18382353 0.19957082 0.12500000 0.12295082
## [457] 0.27227723 0.11931818 0.21195652 0.16237113 0.28991597 0.16396761
## [463] 0.04411765 0.13636364 0.01785714 0.28461538 0.11290323 0.16782007
## [469] 0.21721311 0.20196078 0.11805556 0.09134615 0.18131868 0.17713004
## [475] 0.32500000 0.18888889 0.18902439 0.13035714 0.17161017 0.10784314
## [481] 0.16666667 0.13745020 0.25000000 0.19056604 0.15000000 0.15833333
## [487] 0.17105263 0.22798742 0.18055556 0.09615385 0.20454545
```

We'll assume a Beta prior distribution for p_i, which is defined as: $\text{Prior}(p_i \mid \alpha, \beta) = \frac{B(\alpha,\beta)}{p_i^{\alpha-1}(1-p_i)^{\beta-1}}$ Where $B(\alpha, \beta)$ is the Beta function and α and β are the shape parameters of the Beta distribution. These parameters represent our prior beliefs about the distribution of p_i. Choosing specific values for α and β reflects our prior knowledge or assumptions about the players' hitting abilities.

We'll update the prior distribution using the observed data from the first period. According to Bayes' theorem, the posterior distribution is proportional to the product of the prior distribution and the likelihood function. Since H_{i1} follows a binomial distribution with parameters N_{i1} and p_i, the likelihood function is given by: $\text{Likelihood}(H_{i1} \mid N_{i1}, p_i) = \binom{N_{i1}}{H_{i1}} p_i^{H_{i1}}(1-p_i)^{N_{i1}-H_{i1}}$

Multiplying the prior and likelihood, we obtain the unnormalized posterior distribution: $\text{Posterior}(p_i \mid N_{i1}, H_{i1}, \alpha, \beta) \propto p_i^{\alpha+H_{i1}-1}(1-p_i)^{\beta+N_{i1}-H_{i1}-1}$ This posterior distribution is also a Beta distribution with

updated parameters: $\text{Posterior}(p_i \mid N_{i1}, H_{i1}, \alpha, \beta) = \frac{1}{B(\alpha+H_{i1}, \beta+N_{i1}-H_{i1})} p_i^{\alpha+H_{i1}-1}(1-p_i)^{\beta+N_{i1}-H_{i1}-1}$

We can obtain estimates of p_i from the posterior distribution. The mean of the posterior distribution is given by: $\hat{p_i} = \frac{\alpha+\beta}{\alpha+H_{i1}}$

3)

```r
# Transform observed data to Xij
transform_to_X <- function(H, N) {
  return(asin(sqrt((H + 0.25) / (N + 0.5))))
}

# Define likelihood function
likelihood <- function(X, theta, N) {
  return(dnorm(X, mean = theta, sd = sqrt(1 / (4 * N))))
}

# Update Beta parameters
update_beta_params <- function(alpha, beta, H, N) {
  alpha_new <- alpha + H
  beta_new <- beta + N - H
  return(list(alpha = alpha_new, beta = beta_new))
}

# Estimate pi from theta
estimate_pi <- function(theta) {
  return(sin(theta)^2)
}

# Set prior parameters
alpha_prior <- 1
beta_prior <- 1

# Update prior with transformed data
X <- transform_to_X(filtered_data$H.1, filtered_data$AB.1)
posterior_alpha <- alpha_prior + sum(filtered_data$H.1)
posterior_beta <- beta_prior + sum(filtered_data$AB.1) - sum(filtered_data$H.1)

# Estimate theta
estimated_theta <- mean(X)

# Estimate pi
estimated_pi <- estimate_pi(estimated_theta)

# Print estimated pi
print(estimated_pi)
```

```
## [1] 0.1607521
```

Transform Data: We'll apply the transformation $X_{ij} = \arcsin\left(\sqrt{\frac{H_{ij}+\frac{1}{4}}{N_{ij}+\frac{1}{2}}}\right)$ to transform the observed data H_{ij} and N_{ij} into X_{ij} for each player i and period j.

Define Likelihood: We'll model X_{i1} using a normal distribution with parameters θ_i and σ_{i1}^2, where $\theta_i = \arcsin\left(\sqrt{p_i}\right)$ and $\sigma_{i1}^2 = \frac{1}{4N_{i1}}$.

Define Prior for p_i: We'll use the same Beta prior distribution as in the previous question.

Update Prior with Data: We'll update the prior distribution using the observed data X_{i1} to obtain the posterior distribution of p_i.

Estimate p_i: We'll derive estimates of p_i from the posterior distribution, similar to the previous question, by transforming the estimated θ_i back to obtain estimated $p_i = \sin^2(\theta_i)$.

4)

```
# Estimate Hi2/Ni2 from estimated pi (Step 2)
estimated_Hi2_Ni2_2 <- estimate_pi(estimated_pi)

# Estimate Hi2/Ni2 from estimated theta (Step 3)
estimated_theta <- asin(sqrt(estimated_pi))
estimated_Hi2_Ni2_3 <- sin(estimated_theta)^2

# Print estimated Hi2/Ni2 from both approaches
print(estimated_Hi2_Ni2_2)
```

```
## [1] 0.02561942
```

```
print(estimated_Hi2_Ni2_3)
```

```
## [1] 0.1607521
```

We first estimate $\frac{H_{i2}}{N_{i2}}$ using the estimated values of p_i obtained from step 2 (using the Bayesian method). Then, we estimate $\frac{H_{i2}}{N_{i2}}$ using the estimated values of θ_i obtained from step 3 (using the Bayesian hierarchical model). Finally, we print the estimated values of $\frac{H_{i2}}{N_{i2}}$ from both approaches.

5

```
# Calculate estimated pi based on data from the first period (tilde_pi)
tilde_pi <- filtered_data$H.1 / filtered_data$AB.1

# Calculate MSE using maximum likelihood estimates
MLE_MSE <- sum((filtered_data$H.2 / filtered_data$AB.2 - tilde_pi)^2) / n

# Calculate MSE using estimates from Step 2
Bayesian_2_MSE <- sum((filtered_data$H.2 / filtered_data$AB.2 - estimated_Hi2_Ni2_2)^2) / n

# Calculate MSE using estimates from Step 3
Bayesian_3_MSE <- sum((filtered_data$H.2 / filtered_data$AB.2 - estimated_Hi2_Ni2_3)^2) / n

# Print MSEs
print(MLE_MSE)
```

```
## [1] 0.01266148
```

```
print(Bayesian_2_MSE)
```

```
## [1] 0.05287898
```

```
print(Bayesian_3_MSE)
```

```
## [1] 0.01193305
```

We calculate $\tilde{p}_i$ based on the data from the first period. We compute the MSE using the estimated values $\tilde{p}_i$ and the actual values $\frac{Hi2}{Ni2}$ for each approach: maximum likelihood estimates, estimates from Step 2 (Bayesian method), and estimates from Step 3 (Bayesian hierarchical model). Finally, we print the MSEs for comparison.

The variance in Mean Squared Errors (MSEs) among the estimators arises from various factors. Model complexity is heightened in Bayesian methods from steps 2 and 3 due to prior distributions and hierarchical modeling, potentially increasing MSEs if assumptions or priors are ill-suited or misspecified. The choice of priors in Bayesian methods impacts estimates, with uninformed or divergent priors potentially biasing estimates, unlike Maximum Likelihood Estimation (MLE). Incorporating data structure in the hierarchical model from step 3 may improve accuracy and lower MSEs compared to MLE and the Bayesian method from step 2. Bayesian methods, particularly hierarchical modeling, tend to be more robust to outliers and small samples, enhancing stability and reducing MSEs. These differences highlight the trade-offs between model complexity, priors, modeling flexibility, and robustness. The Bayesian hierarchical model from step 3 strikes a balance among these factors, offering more accurate estimates and lower MSEs compared to both MLE and the simpler Bayesian method from step 2.

6)

```
# Split the dataset into nonpitchers and pitchers
nonpitchers_data <- filtered_data[filtered_data$Pitcher == 0, ]
pitchers_data <- filtered_data[filtered_data$Pitcher == 1, ]

# Define function to estimate pi
estimate_pi <- function(alpha, beta, H, N) {
  return((alpha + H) / (alpha + beta + N))
}

# Perform Bayesian analysis for nonpitchers (steps 2-5)
# For simplicity, let's assume the same prior parameters for both subgroups
nonpitchers_posterior_alpha <- alpha_prior + nonpitchers_data$H.1
nonpitchers_posterior_beta <- beta_prior + nonpitchers_data$AB.1 - nonpitchers_data$H.1
nonpitchers_estimated_pi <- estimate_pi(nonpitchers_posterior_alpha, nonpitchers_posterior_beta, nonpit
nonpitchers_estimated_theta <- asin(sqrt(nonpitchers_estimated_pi))
nonpitchers_estimated_Hi2_Ni2_3 <- sin(nonpitchers_estimated_theta)^2
nonpitchers_MSE <- sum((nonpitchers_data$H.2 / nonpitchers_data$AB.2 - nonpitchers_estimated_Hi2_Ni2_3)

# Perform Bayesian analysis for pitchers (steps 2-5)
pitchers_posterior_alpha <- alpha_prior + pitchers_data$H.1
pitchers_posterior_beta <- beta_prior + pitchers_data$AB.1 - pitchers_data$H.1
```

```r
pitchers_estimated_pi <- estimate_pi(pitchers_posterior_alpha, pitchers_posterior_beta, pitchers_data$H
pitchers_estimated_theta <- asin(sqrt(pitchers_estimated_pi))
pitchers_estimated_Hi2_Ni2_3 <- sin(pitchers_estimated_theta)^2
pitchers_MSE <- sum((pitchers_data$H.2 / pitchers_data$AB.2 - pitchers_estimated_Hi2_Ni2_3)^2) / nrow(pi

# Compare MSEs
print("Nonpitchers MSE:")
```

```
## [1] "Nonpitchers MSE:"
```

```r
print(nonpitchers_MSE)
```

```
## [1] 0.0117451
```

```r
print("Pitchers MSE:")
```

```
## [1] "Pitchers MSE:"
```

```r
print(pitchers_MSE)
```

```
## [1] 0.01340889
```

We split the dataset into non-pitchers and pitchers. We perform Bayesian analysis separately for each subgroup (steps 2-5) and compute the MSEs. Finally, we compare the MSEs between non-pitchers and pitchers.

The Mean Squared Error (MSE) for nonpitchers is slightly lower than that for the combined dataset (Step 5), indicating potentially more accurate predictions of $\frac{Hi2}{Ni2}$. This could be due to nonpitchers having a more uniform skill set, allowing for better estimation when modeled separately. Conversely, the MSE for pitchers is slightly higher, suggesting that separate modeling might yield slightly less accurate predictions of $\frac{Hi2}{Ni2}$ due to their specialized skills and smaller sample size. Heterogeneity in skill levels between pitchers and nonpitchers affects modeling effectiveness, with nonpitchers likely benefiting more from separate modeling due to their more consistent skill distribution. The smaller sample size for pitchers leads to increased uncertainty and higher MSEs, while the choice of prior distributions in Bayesian methods also impacts accuracy. Overall, while subgroup modeling may offer some advantages, the differences in MSEs are minor, requiring consideration of model complexity, sample size, and skill level heterogeneity when deciding whether to model subgroups separately.