Write a full-length graduate level academic research paper about a project that uses Bayesian methods to quantify lineup protection in major league baseball. Break the paper into the following sections, and go into heavy details on each section:

Abstract

Introduction

* Discuss concept of lineup protection and whether or not it seems to exist – both anecdotally and from prior research studies (see references at end)
* Discuss why quantifying lineup protection could be useful for a team – lineup construction, player performance projection, etc.

Data

* Pitch level data from Statcast
* Used <https://saberpowers.github.io/sabRmetrics/> to scrape data

Stabilized xwOBA model

* Explain what xwOBA is (Statcast metric “expected weighted on-base average”) and why it is a good proxy of batter talent
* Creating a stabilized xwOBA for all batters using a hierarchical model – explain why it is a good step (shrinkage for players with fewer plate appearances, aging, positional differences)
* Outputs will be used as inputs for both current batter and on-deck batter in protection model
* stan data below, write out full model structure in correct notation

## Stabilized xwOBA with Age Curve and Positional Adjustments -----------------------

# 1) Stan Model

stan\_code\_wOBA <- "

data {

int<lower=1> J; // number of player‐seasons

vector[J] y; // historical xwOBA

int<lower=0> N[J]; // historical PA

matrix[J,4] A\_basis; // B‐spline basis evaluated at age

int<lower=1> pos[J]; // position index in 1..9

int<lower=1> K\_pos; // 9

int<lower=1> K\_basis; // 4

}

parameters {

real<lower=0,upper=1> mu; // global intercept

real<lower=0> tau; // between‐player SD

real<lower=0> sigma; // measurement noise scale

matrix[K\_pos,K\_basis] gamma; // age‐position spline coefs

vector[J] eta; // player‐level deviations

}

transformed parameters {

vector[J] theta;

// construct theta\_j = mu + f\_{pos[j]}(age[j]) + eta[j]

for (j in 1:J) {

// dot product of the 4‐vector A\_basis[j] with gamma[pos[j], ]

theta[j] = mu + dot\_product( A\_basis[j], gamma[pos[j]] ) + eta[j];

}

}

model {

// Priors

mu ~ normal(0.320, 0.05); // league-average prior (0.320)

tau ~ cauchy(0, 0.05);

sigma ~ cauchy(0, 0.05);

to\_vector(gamma) ~ normal(0, 0.1);

eta ~ normal(0, tau);

// Likelihood (if N[j]=0, no data term → pure prior)

for (j in 1:J) {

if (N[j] > 0)

y[j] ~ normal(theta[j], sigma / sqrt(N[j]));

}

}

generated quantities {

vector[J] y\_rep;

for (j in 1:J) {

if (N[j] > 0) {

// realistic posterior‐predictive draw

y\_rep[j] = normal\_rng(theta[j], sigma / sqrt(N[j]));

} else {

// no plate appearances → just return the latent theta (or league mean)

y\_rep[j] = theta[j];

}

}

}

"

* show convergence plots, posterior predictive checks, and other model validations

Lineup Protection Model

* stan data below, write out full model structure in correct notation
* show convergence plots, posterior predictive checks, and other model validations
* Ran model below to predict probability of a pitch inside strike zone given multiple criteria:

## Predicting Binary Outcome in\_strike\_zone -----------------------------------------

stan\_code\_in\_strike\_zone <- "

data {

int<lower=1> N; // total number of pitches

int<lower=0,upper=1> y[N]; // 1 if pitch is in zone

vector[N] woba; // current batter’s wOBA

vector[N] woba\_on\_deck; // on‑deck batter’s wOBA

int<lower=0,upper=3> balls[N]; // balls in count

int<lower=0,upper=2> strikes[N]; // strikes in count

int<lower=1> N\_pitchers; // total unique pitchers

int<lower=1> N\_batters; // total unique batters

int<lower=1,upper=N\_pitchers> pitcher\_id[N];

int<lower=1,upper=N\_batters> batter\_id[N];

}

parameters {

real alpha;

real beta\_woba;

real beta\_woba\_on\_deck;

real beta\_balls;

real beta\_strikes;

real beta\_int\_balls; // interaction: balls × woba\_on\_deck

real beta\_int\_strikes; // interaction: strikes × woba\_on\_deck

}

model {

// Priors

alpha ~ normal(0, 2);

beta\_woba ~ normal(0, 1);

beta\_woba\_on\_deck ~ normal(0, 1);

beta\_balls ~ normal(0, 1);

beta\_strikes ~ normal(0, 1);

beta\_int\_balls ~ normal(0, 1);

beta\_int\_strikes ~ normal(0, 1);

// Likelihood

for (i in 1:N) {

real nu = alpha

+ beta\_woba \* woba[i]

+ beta\_woba\_on\_deck \* woba\_on\_deck[i]

+ beta\_balls \* balls[i]

+ beta\_strikes \* strikes[i]

+ beta\_int\_balls \* (balls[i] \* woba\_on\_deck[i])

+ beta\_int\_strikes \* (strikes[i] \* woba\_on\_deck[i]);

y[i] ~ bernoulli\_logit(nu);

}

}

"

* I ran a MCMC sampler and got the below results:

Inference for Stan model: anon\_model.

4 chains, each with iter=2000; warmup=1000; thin=1;

post-warmup draws per chain=1000, total post-warmup draws=4000.

mean se\_mean sd 2.5% 50% 97.5% n\_eff Rhat

alpha -0.31 0 0.01 -0.32 -0.31 -0.29 2297 1

beta\_woba -0.01 0 0.00 -0.01 -0.01 0.00 5179 1

beta\_woba\_on\_deck 0.00 0 0.00 -0.01 0.00 0.01 3738 1

beta\_balls 0.22 0 0.01 0.20 0.22 0.23 2749 1

beta\_strikes -0.32 0 0.01 -0.33 -0.32 -0.31 2150 1

beta\_int\_balls 0.00 0 0.00 -0.01 0.00 0.00 4309 1

beta\_int\_strikes 0.00 0 0.00 0.00 0.00 0.01 4117 1

Samples were drawn using NUTS(diag\_e) at Sun Apr 20 14:59:17 2025.

For each parameter, n\_eff is a crude measure of effective sample size,

and Rhat is the potential scale reduction factor on split chains (at

convergence, Rhat=1).

* The below plot shows how the strike probability varies by count and on-deck wOBA, with a league-average batter at bat

A white grid with blue lines

AI-generated content may be incorrect.

Discussion and Future Steps

Cite these sources where appropriate in the paper:

<https://mlbresearch.blogspot.com/2007/04/protection-exists.html#:~:text=more%20pitches%20in%20the%20strike,outs%2C%20runners%2C%20ballpark%2C%20and%20year>

<https://www.si.com/mlb/juan-soto-admits-batting-mets-lineup-different-yankees-aaron-judge>

<https://blogs.fangraphs.com/players-view-does-lineup-protection-exist/>

<https://tht.fangraphs.com/pitching-around-batters/>