Final Exam for *Advanced topics in statistical modeling*
ADSAI PhD Course 2021-2022
Due on: Monday, 19th September

Instructors

Nicola Torelli (DEAMS, University of Trieste, `nicola.torelli@deams.units.it`)
Leonardo Egidi (DEAMS, University of Trieste, `legidi@units.it`)
Francesco Pauli (DEAMS, University of Trieste, `francesco.pauli@deams.units.it`)

Assignment 1

The paper Efron & Morris, (1975) uses in Table 1 the small baseball data set of Efron and Morris (1975) drawn from the 1970 Major League Baseball season from both leagues (data stored in the file `EfronMorrisBB.txt`). Give a look also at the paper Efron & Morris (1977) for a rather less technical treatment. The data separates the outcome from the initial 45 at-bats from the rest of the season. A batting average is defined, of course, simply as the number of hits divided by the number of times at bat; it is always a number between 0 and 1.

	FirstName	LastName	Hits	At.Bats	RemainingAt.Bats	RemainingHits
1	Roberto	Clemente	18	45	367	127
2	Frank	Robinson	17	45	426	127
3	Frank	Howard	16	45	521	144
4	Jay	Johnstone	15	45	275	61
5	Ken	Berry	14	45	418	114
6	Jim	Spencer	14	45	466	126
7	Don	Kessinger	13	45	586	155
8	Luis	Alvarado	12	45	138	29
9	Ron	Santo	11	45	510	137
10	Ron	Swaboda	11	45	200	46
11	Rico	Petrocelli	10	45	538	142
12	Ellie	Rodriguez	10	45	186	42
13	George	Scott	10	45	435	132
14	Del	Unser	10	45	277	73
15	Billy	Williams	10	45	591	195
16	Bert	Campaneris	9	45	558	159
17	Thurman	Munson	8	45	408	129
18	Max	Alvis	7	45	70	14

- Use the following code to access the table elements:

```
N <- dim(df)[1]
K <- df$At.Bats
```

```
y <- df$Hits
K_new <- df$RemainingAt.Bats;
y_new <- df$RemainingHits;
```

where N is the number of items (players). Then for each item n, K_n is the number of initial trials (at-bats), y_n is the number of initial successes (hits), K_new_n is the remaining number of trials (remaining at-bats), and y_new_n is the number of successes in the remaining trials (remaining hits). The remaining data can be used to evaluate the predictive performance of our models conditioned on the observed data. That is, we will "train" on the first 45 at bats and see how well our various models do at predicting the rest of the season.

- Assume the following complete-pooling binomial model: $p(y_n|\theta) = \mathsf{Bin}(y_n|K_n, \theta)$. Fit the model by using the `glm` function. Then, specify some possible priors for θ and fit the model by using `rstan`. Interpret the results and compare the distinct fits.

- Assume now a no-pooling model, which involves a separate chance-of-success parameter $\theta_n \in [0, 1]$ for each item n. The prior on each θ_n is uniform, $p(\theta_n) = \mathsf{Uniform}(\theta_n|0, 1)$. In such a way, the likelihood is $p(y_n|\theta_n) = \mathsf{Bin}(y_n|K_n, \theta_n)$. Fit the model by using `rstan`. Interpret and discuss.

- Assume now a multilevel model, where the players are assumed to belong to a population of players (one group for each distinct player). Fit the model by using the `glmer` function of the package `lme4`. Then, specify some possible priors for θ and fit the model by using `rstan`. Interpret the results and compare the distinct fits. In case of a poor fit evaluate possible model's reparametrizations. Do all players have the same chance of success?

- Plot the observed vs estimated chances of success under the models.

- Check your Bayesian models, for instance by using Graphical Posterior Predictive Checks and using Bayesian p-values.

- Prediction: the question arises as to how well these models predict a player's performance for the rest of the season based on their initial 45 at bats. Thus, make future held-out data predictions based on the posterior predictive distribution.

- Plot the held-out predictions for the considered models, by acknowledging prediction's uncertainty.

- Which is the best calibrated model in the sense of having roughly the right number of actual values fall within the prediction intervals?

- Check predictive accuracy of the proposed models with some suited metrics.

- We usually recommend fitting simulated data. For all or some of the models, generate data according to the prior and test whether the fitted model recovers the parameter values within their appropriate intervals.

- Generate fake data according to the pooling, no-pooling, and one of the hierarchical models. Fit the model and consider the coverage of the posterior 80% intervals.

- How sensitive is the basic no-pooling model to the choice of a prior? Consider using knowledge of baseball to provide a weakly informative prior for θ_n. How, if at all, does this affect posterior inference?

- Compute the James-Stein estimator for the batting averages abilities and then compare it with the Bayesian estimator from the multilevel model and the maximum likelihood estimation obtained through `glmer`. What is your comment? What about the degree of shrinkage of the proposed methods? Interpret and discuss. Hint: you could use the squared loss to compare distinct estimators.

- Apply a variant of the JS estimator as proposed by Efron & Morris (1975). Compute and comment.

- Realize a simple Shiny App to visualize the estimates of the baseball players and the corresponding predictions under the different models.

- Reverse approach. Consider the proportions of correctly successful pancreatic surgeries obtained from ten prominent US hospitals. Suppose you get the following JS estimates about the intrinsic hospital abilities:

New York	0.64
Seattle	0.41
Chicago	0.73
Miami	0.55
St Louis	0.49
New Orleans	0.81
Denver	0.75
Detroit	0.69
Boston	0.71
Houston	0.79

  Now, construct one (or more!) datasets consistent with these estimates. Fit a multilevel model on these *reverse* data. Compute the credibility intervals for the hospital abilities and check the model calibration at, say, the 80% level. Compute these final results with the JS estimates above.

Assignment 2

Consider the `mcycle` data in the `MASS` package: the data measure the acceleration of the rider's head, against time, in a simulated motorcycle crash.

- Plot the acceleration against time, and use gam to fit a univariate smooth to the data, selecting the smoothing parameter by GCV. Plot the resulting smooth, with partial residuals, but without standard errors.

- Use `lm` and `poly` to fit a polynomial to the data, with approximately the same degrees of freedom as was estimated by `gam`. Use `termplot` to plot the estimated polynomial and partial residuals. Note the substantially worse fit achieved by the polynomial, relative to the penalized regression spline fit.

- It's possible to overstate the importance of penalization in explaining the improvement of the penalized regression spline, relative to the polynomial. Use `gam` to refit an unpenalized thin plate regression spline to the data, with basis dimension the same as that used for the polynomial, and again produce a plot for comparison with the previous two results.

- Redo part (c) using an unpenalized cubic regression spline. You should find a fairly clear ordering of the acceptability of the results for the 4 models tried - what is it?

- Now plot the model residuals against time, and comment.
- Experiment with the order of penalty used in the smooth. Does increasing it affect the model fit?