Tyler Schappe

ST540 Midterm II

Introduction. Nike recently released a new line of shoe called Vaporfly and we are interested in knowing whether it significantly reduces the time it takes elite athletes to run a marathon, and whether that effect varies by gender, runner, or marathon course. Our dataset consists of 1618 observations of different athletes running different marathons with and without the Vaporfly shoes.

Methods. I chose to fit several multilevel statistical models of varying complexity, all of which have a core set of common. The core parameters include an overall intercept, which represents the mean marathon time for a male without Vaporfly shoes with a mean age, and a slope parameter for the interaction between the continuous age predictor and gender in order to allow the slope to vary by gender. The slope parameters represent the change in mean time for each change of one standard deviation of age for men and for women. All of the parameters discussed so far were considered fixed effects because the grouping variable 'gender' contains the only two possible values (admittedly a simplification of the definition of gender) and does therefore not represent a sample of possible values of gender as a random effect would.

In addition to the aforementioned fixed effects, all models included 'nested' random intercepts consisting of 'marathon' and 'runner' main effects, and 'vaporfly:marathon' and 'vaporfly:runner' interaction effects. We are entering these as random effects in the model because we believe that both the 'marathon' and 'runner' grouping variables contain observations that do not exhaust the population of marathons and runners. The main random effects enable adjustments to overall intercept and allow for correlation of observations within marathons and within runners, which is reasonable to expect. To defend parameterizing these effects as nested, a counterexample is perhaps useful: Had we not included the interaction terms, the parameterization would imply that the 'vaporfly' effect applies equally across all marathons and similarly across all runners, which is probably unrealistic and also does not allow the model to address goals two and three of the study. Thus, the 'vaporfly:marathon' and 'vaporfly:runner' effects are additional adjustments to the overall intercept that capture the effect of wearing or not wearing the Vaporfly shoes in a particular marathon and for a particular runner.

Four additional complexities were included in the models that I fit. First, a random slope effect was included that enables the relationship between age and time to vary for each marathon; this also involved allowing for a correlation between the slope and intercept parameters across marathons using a covariance matrix prior. Second, the assumption of homoscedasticity of errors was relaxed by fitting separate error variance parameters for each marathon. Third, the response variable, time in minutes to complete the marathon, was log-transformed because errors from initial models appeared to be left-skewed. Fourth, the errors were assumed to follow a t-distribution instead of a normal to allow for more extreme errors; this model was the most complex of all fit. Below is a listing of models fit in this study:

Model 1

$$\begin{aligned} time_{ijklm} \sim Normal(&\beta_1 + \beta_2 Vapor + \beta_3 Female + \beta_4 Age + \beta_5 Vapor{:}Female + \beta_6 Age{:}Female \\ &+ Marathon_j + Runner_k + Vapor{:}Marathon_l + Vapor{:}Runner_m, \sigma^2) \end{aligned}$$

Model 2

$$\begin{aligned} time_{ijklm} \sim Normal(&\beta_1 + \beta_2 Vapor + \beta_3 Female + \beta_4 Age + \beta_5 Vapor{:}Female + \beta_6 Age{:}Female \\ &+ Marathon_j + Runner_k + Vapor{:}Marathon_l + Vapor{:}Runner_m, \sigma_j^2) \end{aligned}$$

Model 3

$$\begin{aligned} time_{ijklm} \sim Normal(&\beta_1 + \beta_2 Vapor + \beta_3 Female + \beta_4 Age + \beta_5 Vapor{:}Female + \beta_6 Age{:}Female \\ &+ Marathon_j + Age{:}Marathon_j + Runner_k + Vapor{:}Marathon_l \\ &+ Vapor{:}Runner_m, \sigma^2) \end{aligned}$$

Model 4

$$\begin{aligned} time_{ijklm} \sim Normal(&\beta_1 + \beta_2 Vapor + \beta_3 Female + \beta_4 Age + \beta_5 Vapor{:}Female + \beta_6 Age{:}Female \\ &+ Marathon_j + Age{:}Marathon_j + Runner_k + Vapor{:}Marathon_l \\ &+ Vapor{:}Runner_m, \sigma_j^2) \end{aligned}$$

Model 5

$$\begin{aligned} \log(time_{ijklm}) \sim Normal(&\beta_1 + \beta_2 Vapor + \beta_3 Female + \beta_4 Age + \beta_5 Vapor{:}Female \\ &+ \beta_6 Age{:}Female + Marathon_j + Age{:}Marathon_j + Runner_k + Vapor{:}Marathon_l \\ &+ Vapor{:}Runner_m, \sigma_j^2) \end{aligned}$$

Model 6

$$\begin{aligned} \log(time_{ijklm}) \sim Student\_t(&\eta, \beta_1 + \beta_2 Vapor + \beta_3 Female + \beta_4 Age + \beta_5 Vapor{:}Female \\ &+ \beta_6 Age{:}Female + Marathon_j + Age{:}Marathon_j + Runner_k + Vapor{:}Marathon_l \\ &+ Vapor{:}Runner_m, \sigma_j^2) \end{aligned}$$

Priors. Because the nested random effects were included in all models, the number of parameters for all models approached the total number of observations and, therefore, shrinkage priors were used for all random effects with common hyper-variance parameters within grouping variables.

$$Marathon_j \sim Normal(0, \sigma_j^2)$$
$$Runner_k \sim Normal(0, \sigma_k^2)$$
$$Vapor{:}Marathon_l \sim Normal(0, \sigma_l^2)$$
$$Vapor{:}Runner_m \sim Normal(0, \sigma_m^2)$$
$$\sigma_j^2, \sigma_k^2, \sigma_l^2, \sigma_m^2 \sim Cauchy(0, 2)$$

Additional priors were needed for models that included a random slope effect for age by marathon. Note that the covariance matrix was modeled using a Cholesky decomposition into correlation (L) and volatilities (τ) matrices; the LKJ prior with parameter value 1 implies a uniform density on L.

$$(Marathon_j, Age{:}Marathon_j) \sim MultiNormal(\begin{matrix}0\\0\end{matrix}, \Sigma)$$

$$\Sigma = \Lambda\Lambda^T \qquad \Lambda = \tau L \qquad \tau = \begin{pmatrix}\sigma_{m\_int} & 0\\ 0 & \sigma_{m\_slope}\end{pmatrix} \qquad L = \begin{pmatrix}1 & 0\\ \rho & \sqrt{1-\rho^2}\end{pmatrix}$$

$$L \sim LKJ(1) \qquad \sigma_{m\_int}^2, \sigma_{m\_slope}^2 \sim Cauchy(0, 2)$$

Computation. All models were coded in the Stan language and posterior distributions were sampled using the Stan statistical analysis platform via the Hamiltonian Markov Chain (HMC) algorithm in R using package ‘rstan’. Sampling was performed with 3 independent chains each with 3000 iterations of warmup and 8000 sampling iterations. Convergence was checked by ensuring that the Gelman-Rubin statistic for all parameters was between 0.9 and 1.1 and the effective sample size was > 1000 for all parameters; trace plots were visually inspected for proper mixing of chains.

Model comparisons. Model 6 scored the best WAIC value (Table 1), but posterior predictive checks indicated that model 5 achieved superior Bayesian p-values for the mean and median. Despite the better WAIC, I considered model 6 to have failed the posterior predictive checks and I therefore selected model 5 as the final model.

Model	**1**	**2**	**3**	**4**	**5**	**6**
WAIC	6909	6791	6912	6780	-4480	-4526

Table 1. WAIC values for all 6 candidate models.

Model	Bayesian p-values					MSE quantiles	MAD quantiles
	Mean	Median	Min	Max	Range		
5	0.244	0.540	0.574	0.001	0.000	[18.4, 22.1]	[3.1,3.5]
6	0.000	0.017	0.699	0.001	0.000	[18.8,22.5]	[2.9, 3.2]

Table 2. Bayesian p-values, MSE and MAD quantiles from 8000 PPD samples

Results. The fixed effects terms involved in the Vaporfly:gender interaction directly address the question of whether the Vaporfly shoes have an effect for men and women. Table 3 gives the 95% HDIs for each of these parameters and shows that for men, wearing Vaporfly shoes reduces the median time by between 0.6% and 2.7%; there is no significant effect for women because the 95% HDI includes zero. To determine whether the effect of Vaporfly shoes varies by runner or by marathon course, I performed a bootstrapping procedure by fitting two reduced models, one without any Vaporfly:marathon effects and one without Vaporfly:runner effects, in addition to the full model (model 5) with 10 bootstrap replicates each and calculated WAIC each time. Time constraints prohibited additional replicates. This gave me a sampling distribution of WAIC and corresponding quantiles for each model. This method was intended to mimic a likelihood ratio test except using WAIC, which includes likelihood information. The quantiles in Table 4 indicate that dropping all Vaporfly:marathon effects does not reduce the fit, and therefore I conclude that the Vaporfly effect does not vary by marathon course. In contrast, dropping the Vaporfly:runner effects does reduce WAIC, and so the Vaporfly effects vary by runner.

Parameter	Mean	Median	2.5% HDI	97.5% HDI
Vaporfly:Men	-0.0172	-0.0171	-0.0279	-0.00621
Vaporfly:Women	0.0014	0.0015	-0.0131	0.159

Table 3. Posterior means, medians, and 95% HDIs for the effect of wearing Vaporfly shoes for both men and women.

Bootstrap quantile	Full (model 5)	No Vaporfly:marathon	No Vaporfly:runner
2.5%	-4480.9	-4480.4	-4472.6
97.5%	-4479.0	-4477.1	-4458.8

Table 4. 2.5% and 97.5% quantiles of bootstrap WAIC values for the full model (model 5) and reduced models without either Vaporfly:marathon or Vaporfly:runner interaction effects.

Appendix. Stan code of final model (model 5).

```
data {
  int<lower=1> N;                              //Number of observations
  int<lower=1> P;                              //Number of predictors in X
  int<lower=1> M;                              //Number of marathons
  int<lower=1> Mp;                             //Number of marathon predictors
  int<lower=1> R;                              //Number of runners
  int<lower=1> VR;                             //Number of levels for vaporfly:runner
  int<lower=1> VM;                             //Number of levels for vaporfly:marathon
  int<lower=1, upper=M> marathon[N];           //Index of marathons
  int<lower=1, upper=R> runner[N];             //Index of runners
  int<lower=1, upper=VR> vaporfly_runner[N];   //Index of vaporfly:runner
  int<lower=1, upper=VM> vaporfly_marathon[N]; //Index of vaporfly:marathon
  matrix[N,P] X;                               //Fixed effects model matrix
  matrix[N,Mp] Z_m;                            //Marathon-level random effects model matrix
  vector[N] y;                                 //Response variable
}

transformed data {
  vector[N] logy;                              //Log-transformed response variable

  logy = log(y);
}

parameters {
  vector[P] beta;                              //Vector of fixed-effects coefficients
  real<lower=0> sigma_r;                       //Runner-level variance
  real<lower=0> sigma_vr;                      //Vaporfly:runner-level variance
  real<lower=0> sigma_vm;                      //Vaporfly:marathon-level variance
  vector<lower=0>[Mp] sigma_m;                 //Vector for marathon-level variances (intercept and
slope for age)
  cholesky_factor_corr[Mp] L_m;                //Correlation vector (Cholesky decomposition) of
marathon-level intercept and slope
  real<lower=0> sigma_e[M];                    //Error variance -- separate for each marathon
  vector[R] r_raw;                             //Vector of unscaled runner-level random intercepts
  vector[VR] vr_raw;                           //Vector of unscaled vaporfly:runner-level random
intercepts
  vector[VM] vm_raw;                           //Vector of unscaled vaporfly:marathon-level random
intercepts
  vector[Mp] m_raw[M];                         //Array of vectors of unscaled marathon-level random
intercepts and slopes
}

transformed parameters {
  vector[R] r;                                 //Vector of scaled runner-level random intercepts
  vector[VR] vr;                               //Vector of scaled vaporfly:runner-level random
intercepts
  vector[VM] vm;                               //Vector of scaled vaporfly:marathon-level random
intercepts
  vector[Mp] m[M];                             //Array of vectors of scaled marathon-level random
intercepts and slopes

  {
    matrix[Mp, Mp] Sigma_m;                    //Marathon-level intercept and slope covariance matrix
    r = sigma_r * r_raw;                       //Create scaled runner-level random intercepts
    vr = sigma_vr * vr_raw;                    //Create scaled vaporfly:runner-level random intercepts
    vm = sigma_vm * vm_raw;                    //Create scaled vaporfly:marathon-level random
intercepts

    Sigma_m = diag_pre_multiply(sigma_m, L_m);  //Create marathon=level intercept and slope covariance
matrix using Cholesky decomposition
    for (a in 1:M) {
      m[a] = Sigma_m * m_raw[a];               //Create scaled marathon-level random intercepts and
slopes
    }
  }
}

model {
  r_raw ~ std_normal();                        //Standard normal prior on unscaled runner-level random
intercepts. Has effect of r ~ Normal(0, r_sigma).
  vr_raw ~ std_normal();                       //Standard normal prior on unscaled vaporfly:runner-
level random intercepts. Has effect of vr ~ Normal(0, vr_sigma).
```

```
  vm_raw ~ std_normal();                        //Standard normal prior on unscaled vaporfly:marathon-
level random intercepts. Has effect of vm ~ Normal(0, vm_sigma).
  for (b in 1:M) {
    m_raw[b] ~ std_normal();                    //Standard normal prior on unscaled marathon-level
random intercepts. Has effect of m ~ MultivariateNormal(c(0,0), Sigma_m).
    sigma_e[b] ~ cauchy(0,2);                   //Cauchy prior on error variance.
  }
  L_m ~ lkj_corr_cholesky(1);                   //LKJ prior on Cholesky correlation vector. Imposes
uniform prior on correlation between intercept and slope.
  sigma_r ~ cauchy(0,2);                        //Cauchy prior on runner-level hypervariance parameter.
  for (c in 1:Mp) {
    sigma_m[c] ~ cauchy(0,2);                   //Independent Cauchy priors on marathon-level intercept
and slope hypervariance parameters.
  }
  sigma_vr ~ cauchy(0,2);                       //Cauchy prior on vaporfly:runner-level hypervariance
parameter.
  sigma_vm ~ cauchy(0,2);                       //Cauchy prior on vaporfly:marathon-level hypervariance
parameter.

  //Likelihood                                  //Normal likelihood function with mixed effects linear
predictor and homoscedastic error variance assumption.
  for (i in 1:N) {
     logy[i] ~ normal(X[i,]*beta + Z_m[i,]*m[marathon[i]] + r[runner[i]] + vr[vaporfly_runner[i]] +
vm[vaporfly_marathon[i]], sigma_e[marathon[i]]);
  }
}

generated quantities {
  vector[N] residuals;                          //Vector for residuals
  vector[N] yhat;                               //Vector for estimated linear predictor
  vector[N] yhat_ppd;                           //Vector for posterior predictive distribution
  vector[N] log_lik;                            //Vector for log-likelihood values for WAIC calculation

  for (j in 1:N) {
    //Samples for the linear predictor yhat
    yhat[j] = exp(X[j,]*beta + Z_m[j,]*m[marathon[j]] + r[runner[j]] + vr[vaporfly_runner[j]] +
vm[vaporfly_marathon[j]]);
    //Samples for the posterior predictive distribution, which includes error uncertainty
    yhat_ppd[j] = normal_rng(exp(X[j,]*beta + Z_m[j,]*m[marathon[j]] + r[runner[j]] +
vr[vaporfly_runner[j]] + vm[vaporfly_marathon[j]]), sigma_e[marathon[j]]);
    //Samples of log-likelihood for WAIC calculation
    log_lik[j] = normal_lpdf(logy[j] | (X[j,]*beta + Z_m[j,]*m[marathon[j]] + r[runner[j]] +
vr[vaporfly_runner[j]] + vm[vaporfly_marathon[j]]), sigma_e[marathon[j]]);
  }

  residuals = y - yhat;                         //Define residuals as y - hat

}
```