Between-centre differences for COVID-19 ICU mortality from early data in England
ANALYSIS SUPPLEMENT

Zhaozhi Qian[1], Ahmed M. Alaa[2], Mihaela van der Schaar[1], and Ari Ercole[3,*]
[1]University of Cambridge Centre for Mathematical Sciences, Wilberforce Rd, Cambridge CB3 0WA, UK
[2]University of California, Los Angeles, CA 90095, USA
[3]University of Cambridge Division of Anaesthesia, Addenbrooke's Hospital, Hills Road, Cambridge CB2 0QQ, UK
[*]Corresponding author

This file contains the code from the analysis as canonical documentation of the pre-processing and modelling assumptions.

Final model results

Fixed effects

Fixed effect	Log hazard ratio	Hazard ratio for mortality	S.E. of log HR	*p*
Age (years)				
50-60	0.53	1.69	0.10	<0.0001
60-70	1.04	2.83	0.09	<0.0001
70-80	1.49	4.44	0.10	<0.0001
>80	1.66	5.24	0.14	<0.0001
Male sex	0.19	1.21	0.06	0.0022
Chronic respiratory disease	0.26	1.29	0.09	0.0037
Asthma requiring treatment	-0.01	0.99	0.09	0.87
Chronic heart disease	0.32	1.38	0.08	0.0001
Chronic renal disease	0.22	1.25	0.10	0.026
Chronic liver disease	0.17	1.18	0.20	0.41
Chronic neurological disease	0.06	1.06	0.14	0.67
Diabetes	0.12	1.13	0.07	0.067
Immunosuppressive treatment	0.15	1.16	0.13	0.27
Immunosuppressive disease	0.48	1.62	0.16	0.0020
Obesity	0.19	1.20	0.07	0.013
Hypertension	0.10	1.11	0.06	0.11
Mean APACHE-II score (for site)	0.08	1.09	0.11	0.45

Random effects

Random effects Intercept: Std Dev 0.88 Variance 0.77
*p*_value for random effect significance: 2.2e-16

Site number (random effect- arbitrary/ordered)	Hazard ratio estimate for mortality	Lower 95% CI	Upper 95% CI
1	0.048	0.020	0.120
2	0.086	0.033	0.224
3	0.097	0.037	0.255
4	0.147	0.052	0.417
5	0.238	0.076	0.743
6	0.242	0.125	0.467
7	0.259	0.121	0.553
8	0.273	0.123	0.605
9	0.282	0.087	0.914
10	0.424	0.141	1.271
11	0.478	0.125	1.823
12	0.481	0.127	1.827
13	0.498	0.265	0.935
14	0.527	0.194	1.430
15	0.551	0.175	1.736
16	0.575	0.351	0.944
17	0.588	0.238	1.452
18	0.616	0.403	0.942
19	0.648	0.283	1.485
20	0.669	0.287	1.561
21	0.712	0.407	1.248
22	0.749	0.220	2.554
23	0.757	0.448	1.280
24	0.761	0.296	1.954
25	0.772	0.472	1.264
26	0.786	0.415	1.489
27	0.808	0.476	1.371
28	0.812	0.342	1.928
29	0.865	0.479	1.562
30	0.889	0.246	3.213
31	0.894	0.647	1.235
32	0.899	0.444	1.820
33	0.913	0.384	2.173
34	0.957	0.510	1.795
35	1.008	0.420	2.417
36	1.018	0.681	1.521
37	1.025	0.617	1.702
38	1.029	0.583	1.818
39	1.052	0.672	1.647
40	1.055	0.622	1.790
41	1.069	0.568	2.013
42	1.084	0.532	2.207
43	1.091	0.478	2.488
44	1.119	0.755	1.657
45	1.148	0.620	2.127
46	1.166	0.719	1.891
47	1.173	0.666	2.067
48	1.190	0.518	2.735
49	1.192	0.744	1.911
50	1.192	0.749	1.898
51	1.208	0.571	2.557
52	1.226	0.684	2.197
53	1.258	0.797	1.986
54	1.280	0.670	2.445
55	1.284	0.730	2.258
56	1.289	0.712	2.333
57	1.310	0.931	1.843
58	1.355	0.775	2.370
59	1.379	0.662	2.872
60	1.419	0.967	2.081
61	1.445	0.876	2.385
62	1.447	0.948	2.209
63	1.454	0.841	2.514
64	1.456	0.937	2.262
65	1.460	0.696	3.062
66	1.527	0.681	3.424
67	1.553	0.880	2.742
68	1.629	0.803	3.304
69	1.633	0.637	4.187
70	1.688	0.350	8.131
71	1.725	0.919	3.238
72	1.727	1.066	2.799
73	1.749	0.355	8.622
74	1.792	1.168	2.751
75	1.801	1.229	2.640
76	1.840	0.516	6.555
77	1.845	1.216	2.799
78	1.895	1.238	2.899
79	1.992	1.079	3.676
80	1.993	1.204	3.298
81	2.080	1.158	3.734
82	2.168	1.568	2.997
83	2.302	1.431	3.705
84	2.368	1.404	3.995
85	2.371	1.501	3.744
86	2.488	1.749	3.538
87	2.497	1.784	3.495
88	2.521	1.889	3.364
89	2.542	1.589	4.066
90	2.654	1.843	3.821
91	3.170	2.066	4.864
92	4.922	3.307	7.326

Sensitivity analysis

To ensure model robustness against temporal bias, we conducted a sensitivity analysis excluding patients with less than 7 days follow-up.

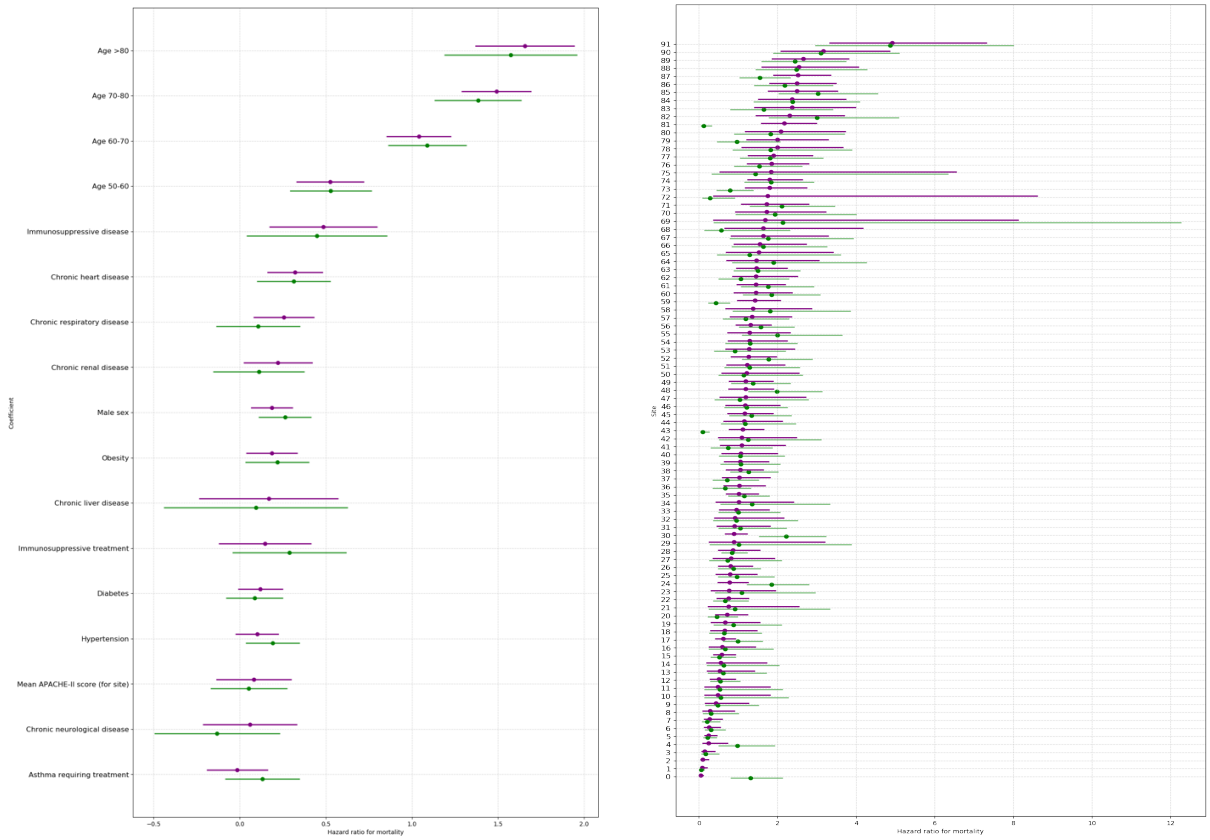

Figure S1. Model (purple) and sensitivity analysis with patients with <7days follow up excluded (green). No substantial temporal ascertainment bias is seen in the fixed effects (left panel) or random effects (right panel).

Full analysis code

Libraries used (latest versions as of 4/2020)

```
> library(ggplot2)
> library(dplyr)
> library(reshape2)
> library(coxme)
> library(tidyr)
```

Data cleaning specification

```
> update_date = '22/5/2020'
> update_dt = as.Date('22/5/2020', format="%d/%m/%Y")
> dat = read.csv(file='data/chess-2020-05-24.csv', stringsAsFactors = FALSE)

> names(dat) = tolower(names(dat))

> # table(dat$hospitaladmissiondate) # '%m/%d/%Y'
> # table(dat$finaloutcomedate)

> dat = dat %>%
+     mutate(dateupdated = gsub( " .*$", "", dateupdated)) %>%
+     mutate_at(.vars=c('dateupdated', 'hospitaladmissiondate', 'dateadmittedicu', 'dateleavingicu', 'finaloutcomedate'),
+               .funs = function(x) as.Date(x, format="%Y-%m-%d"))

> dt = dat$dateupdated
> dt[is.na(dt)] = update_dt
> dat$dateupdated = dt

> dat = dat %>%
+     mutate(hospital_ttc = as.numeric(dateupdated - hospitaladmissiondate),
+            icu_ttc = as.numeric(dateupdated - dateadmittedicu),
+            outcome_time = as.numeric(finaloutcomedate - hospitaladmissiondate),
+            time_to_icu = as.numeric(dateadmittedicu - hospitaladmissiondate),
+            icu_duration = as.numeric(dateleavingicu - dateadmittedicu)
+     ) %>%
+     mutate(finaloutcome = ifelse(finaloutcome == "Transfered", "", finaloutcome)) %>%
+     mutate_at(.vars=c('dateupdated', 'hospitaladmissiondate', 'dateadmittedicu', 'finaloutcomedate', 'dateleavingicu'),
+               .funs = as.character) %>%
+     mutate_all(.funs=function(x) x = ifelse((x == 'Unknown') | (x == 'No') | (x == 'Borderline'), '', x)) %>%
+     mutate(sex = ifelse(sex == 'Male', 'Male', '')) %>%
+     mutate(
+       death_ind = finaloutcome == 'Death',
+       icu_ind = !is.na(time_to_icu),
+       discharge_ind = finaloutcome == 'Discharged',
+       death_time = coalesce(outcome_time, hospital_ttc) ,
+       icu_time = coalesce(time_to_icu, hospital_ttc),
+       discharge_time = coalesce(outcome_time, hospital_ttc),
+     ) %>% group_by(caseid) %>% filter(row_number(caseid) == 1) %>% ungroup() %>%
+     filter(wasthepatientadmittedtoicu == 'Yes')

> nrow(dat) - length(unique(dat$caseid))
[1] 0
> sum(is.na(dat$dateupdated))
[1] 0

> sum(is.na(dat$hospitaladmissiondate))
```

```
[1] 0

> sum(is.na(dat$hospital_ttc))
[1] 0

> sum(is.na(dat$death_time))
[1] 0

> sum(is.na(dat$outcome_time))
[1] 1524

> # 5062 icu patients
> nrow(dat)
[1] 5062

> # death 1547
> dat %>%
+    filter(finaloutcome == 'Death') %>%
+    summarise(n=n())
# A tibble: 1 x 1
     n
  <int>
1  1547

> # ICU discharge 1618
> dat %>%
+    filter(!is.na(dateleavingicu) & finaloutcome != 'Death') %>%
+    summarise(n=n())
# A tibble: 1 x 1
     n
  <int>
1  1618

> dat = dat %>%
+    mutate(
+       ageyear = ifelse((is.na(ageyear) | ageyear == 0), median(ageyear, na.rm=TRUE), ageyear),
+       sex = replace_na(sex, 'Male'),
+
+    )
> dat$age = dat$ageyear
> dat$ageyear = (dat$ageyear - mean(dat$ageyear)) / sd(dat$ageyear )
> nrow(dat)
[1] 5062

> ### trusts
> # trusts = unique(dat$trustname)
> # write.csv(trusts, file='data/letter_trusts.csv')

> trust_scores = read.csv('data/letter_trusts_linked.csv', stringsAsFactors = FALSE)
> apache = read.csv('data/apache.csv', stringsAsFactors = FALSE)
> apache = apache %>%
+    group_by(hospname) %>%
+    summarise(apache = median(apache))

> dat2 = dat %>%
+    inner_join(trust_scores, by='trustname')%>%
+    inner_join(apache, by='hospname')

> dat2$apache_mean = (mean(dat2$apache) - dat2$apache) / sd(dat2$apache)

> dat2$agecut = cut(dat2$age, c(0, 50, 60, 70, 80, 999))
```

```
> table(dat2$agecut)

 (0,50]  (50,60]  (60,70]  (70,80] (80,999]
   1145     1676     1301      707      144
```

Define and fit Cox model

```
> model = coxme(Surv(death_time , death_ind) ~
+               agecut +
+               sex +
+               chronicrespiratory+
+               asthmarequiring+
+               chronicheart +
+               chronicrenal +
+               chronicliver +
+               chronicneurological +
+               isdiabetes +
+               immunosuppressiontreatment +
+               immunosuppressiondisease +
+               obesityclinical +
+               hypertension +
+               apache_mean +
+               (1|trustcode), dat2 )

> summary(model)
Cox mixed-effects model fit by maximum likelihood
  Data: dat2
  events, n = 1508, 4973
  Iterations= 10 66
                NULL Integrated   Fitted
Log-likelihood -12152.62  -11521.08 -11382.4

                      Chisq    df p      AIC     BIC
Integrated loglik  1263.08 18.00 0      1227.08 1131.35
 Penalized loglik  1540.44 92.64 0      1355.16 862.46

Model:  Surv(death_time, death_ind) ~ agecut + sex + chronicrespiratory +
    asthmarequiring + chronicheart + chronicrenal + chronicliver +
    chronicneurological + isdiabetes + immunosuppressiontreatment +
    immunosuppressiondisease + obesityclinical + hypertension +
    apache_mean + (1 | trustcode)
Fixed coefficients
```

	coef	exp(coef)	se(coef)	z	p
agecut(50,60]	0.52556267	1.6914103	0.09853076	5.33	9.6e-08
agecut(60,70]	1.04035339	2.8302170	0.09403432	11.06	0.0e+00
agecut(70,80]	1.49044437	4.4390676	0.10128174	14.72	0.0e+00
agecut(80,999]	1.65566567	5.2365646	0.14476477	11.44	0.0e+00
sexMale	0.18682293	1.2054138	0.06104706	3.06	2.2e-03
chronicrespiratoryYes	0.25562348	1.2912664	0.08815163	2.90	3.7e-03
asthmarequiringYes	-0.01447620	0.9856281	0.08923171	-0.16	8.7e-01
chronicheartYes	0.32175738	1.3795500	0.08084859	3.98	6.9e-05
chronicrenalYes	0.22207150	1.2486607	0.09999441	2.22	2.6e-02
chronicliverYes	0.16810960	1.1830663	0.20216825	0.83	4.1e-01
chronicneurologicalYes	0.05913716	1.0609207	0.13684876	0.43	6.7e-01
isdiabetesYes	0.12027089	1.1278023	0.06565021	1.83	6.7e-02
immunosuppressiontreatmentYes	0.14770605	1.1591721	0.13452276	1.10	2.7e-01
immunosuppressiondiseaseYes	0.48452881	1.6234099	0.15673605	3.09	2.0e-03
obesityclinicalYes	0.18591816	1.2043237	0.07480174	2.49	1.3e-02
hypertensionYes	0.10072133	1.1059684	0.06262017	1.61	1.1e-01
apache_mean	0.08230137	1.0857830	0.10979409	0.75	4.5e-01

```
Random effects
 Group            Variable         Std Dev          Variance
trustcode         Intercept        0.8769697        0.7690758

> # fixed effects
> coeff_main = fixef(model)
> sd_main = sqrt(diag(vcov(model)))

> df_fixed = data.frame(effect = names(coeff_main),
+                  coeff = coeff_main,
+                  coeff_sd = sd_main,
+                  hr = exp(coeff_main),
+                  hr_lower = exp(coeff_main - 2 * sd_main),
+                  hr_upper = exp(coeff_main + 2 * sd_main),
+                  row.names=NULL)

> write.csv(df_fixed, file = 'letter_fixed_effect_updated.csv')

> # random effects
> coeff_rand = ranef(model)$trustcode
> sd_rand = sqrt(diag(model$variance)[1:length(coeff_rand)])

> df_random = data.frame(effect = names(coeff_rand),
+                   coeff = coeff_rand,
+                   coeff_sd = sd_rand,
+                   hr = exp(coeff_rand),
+                   hr_lower = exp(coeff_rand - 2 * sd_rand),
+                   hr_upper = exp(coeff_rand + 2 * sd_rand),
+                   row.names=NULL)
> write.csv(df_random, file = 'letter_random_effect_updated.csv')
```

Test random effects

```
> fit1 <- coxph(Surv(death_time , death_ind) ~
+               ageyear +
+               sex +
+               chronicrespiratory+
+               asthmarequiring+
+               chronicheart +
+               chronicrenal +
+               chronicliver +
+               chronicneurological +
+               isdiabetes +
+               immunosuppressiontreatment +
+               immunosuppressiondisease +
+               obesityclinical +
+               hypertension,
+             dat2)
> print(anova(fit1,model))
Analysis of Deviance Table
 Cox model: response is  Surv(death_time, death_ind)
 Model 1: ~ageyear + sex + chronicrespiratory + asthmarequiring + chronicheart +        chronicrenal + chronicliver +
chronicneurological + isdiabetes +        immunosuppressiontreatment + immunosuppressiondisease + obesityclinical +
hypertension
 Model 2: ~agecut + sex + chronicrespiratory + asthmarequiring + chronicheart +        chronicrenal + chronicliver +
chronicneurological + isdiabetes +        immunosuppressiontreatment + immunosuppressiondisease + obesityclinical +
hypertension + apache_mean + (1 | trustcode)
  loglik  Chisq Df P(>|Chi|)
1 -11809
```

2 -11521 576.46 5 < 2.2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Proportional odds assumption

```
> ## test random effect

> fit1 <- coxph(Surv(death_time , death_ind) ~
+               ageyear +
+               sex +
+               chronicrespiratory+
+               asthmarequiring+
+               chronicheart +
+               chronicrenal +
+               chronicliver +
+               chronicneurological +
+               isdiabetes +
+               immunosuppressiontreatment +
+               immunosuppressiondisease +
+               obesityclinical +
+               hypertension,
+             dat2)
> print(anova(fit1,model))
Analysis of Deviance Table
 Cox model: response is  Surv(death_time, death_ind)
 Model 1: ~ageyear + sex + chronicrespiratory + asthmarequiring + chronicheart +      chronicrenal + chronicliver +
chronicneurological + isdiabetes +      immunosuppressiontreatment + immunosuppressiondisease + obesityclinical +
hypertension
 Model 2: ~agecut + sex + chronicrespiratory + asthmarequiring + chronicheart +      chronicrenal + chronicliver +
chronicneurological + isdiabetes +      immunosuppressiontreatment + immunosuppressiondisease + obesityclinical +
hypertension + apache_mean + (1 | trustcode)
  loglik  Chisq Df P(>|Chi|)
1 -11809
2 -11521 576.46  5 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> # p_value for random effect is 2.2e-16

> cox_model = coxph(Surv(death_time , death_ind) ~
+                 ageyear +
+                 sex +
+                 chronicrespiratory +
+                 asthmarequiring +
+                 chronicheart +
+                 chronicrenal +
+                 chronicliver +
+                 chronicneurological +
+                 isdiabetes +
+                 immunosuppressiontreatment +
+                 immunosuppressiondisease +
+                 obesityclinical +
+                 apache_mean, dat2)
>
> cox.zph(cox_model, transform='identity')
                                    chisq  df       p
ageyear                            11.685  1 0.00063
sex                                 2.616  1 0.10578
chronicrespiratory                  0.038  1 0.84552
asthmarequiring                     3.237  1 0.07201
chronicheart                        0.932  1 0.33444
chronicrenal                        0.215  1 0.64310
chronicliver                        0.163  1 0.68654
chronicneurological                 0.751  1 0.38604
isdiabetes                          0.403  1 0.52547
```

immunosuppressiontreatment	0.279	1	0.59724
immunosuppressiondisease	1.122	1	0.28953
obesityclinical	0.234	1	0.62876
apache_mean	1.616	1	0.20368
GLOBAL	21.631	13	**0.06135**

Sensitivity analysis (restrict to patients with only 7 days follow-up)

```
> # sensitivity analysis
>
> model = coxme(Surv(death_time , death_ind) ~
+               agecut +
+               sex +
+               chronicrespiratory+
+               asthmarequiring+
+               chronicheart +
+               chronicrenal +
+               chronicliver +
+               chronicneurological +
+               isdiabetes +
+               immunosuppressiontreatment +
+               immunosuppressiondisease +
+               obesityclinical +
+               hypertension +
+               apache_mean +
+
+               (1|trustcode), dat2 %>% filter(death_time > 7))
>
> summary(model)
Cox mixed-effects model fit by maximum likelihood
  Data: dat2 %>% filter(death_time > 7)
  events, n = 979, 4037
  Iterations= 22 137
                NULL Integrated    Fitted
Log-likelihood -7699.123  -7288.402 -7166.196

                Chisq    df p    AIC    BIC
Integrated loglik  821.44 18.00 0 785.44 697.48
 Penalized loglik 1065.85 88.61 0 888.64 455.67

Model:  Surv(death_time, death_ind) ~ agecut + sex + chronicrespiratory +     asthmarequiring + chronicheart + chronicrenal
+ chronicliver +       chronicneurological + isdiabetes + immunosuppressiontreatment +       immunosuppressiondisease +
obesityclinical + hypertension +      apache_mean + (1 | trustcode)
Fixed coefficients
                                  coef             exp(coef)        se(coef)         z        p
agecut(50,60]                     0.52836784       1.6961616        0.11900812       4.44     9.0e-06
agecut(60,70]                     1.08862495       2.9701871        0.11419653       9.53     0.0e+00
agecut(70,80]                     1.38314764       3.9874329        0.12631705       10.95    0.0e+00
agecut(80,999]                    1.57287387       4.8204817        0.19319242       8.14     4.4e-16
sexMale                           0.26286351       1.3006492        0.07688069       3.42     6.3e-04
chronicrespiratoryYes             0.10721566       1.1131743        0.12166589       0.88     3.8e-01
asthmarequiringYes                 0.13149276      1.1405296        0.10865313       1.21     2.3e-01
chronicheartYes                   0.31319465       1.3677877        0.10721581       2.92     3.5e-03
chronicrenalYes                   0.11072709       1.1170900        0.13286081       0.83     4.0e-01
chronicliverYes                   0.09363496       1.0981588        0.26678907       0.35     7.3e-01
chronicneurologicalYes            -0.13119391      0.8770477        0.18274115       -0.72    4.7e-01
isdiabetesYes                     0.08648545       1.0903355        0.08280806       1.04     3.0e-01
immunosuppressiontreatmentYes     0.28781337       1.3335084        0.16557883       1.74     8.2e-02
immunosuppressiondiseaseYes       0.44786594       1.5649689        0.20403409       2.20     2.8e-02
```

obesityclinicalYes	0.21865868	1.2444065	0.09261895	2.36	1.8e-02
hypertensionYes	0.19094188	1.2103891	0.07863300	2.43	1.5e-02
apache_mean	0.05333766	1.0547857	0.11157739	0.48	6.3e-01

Random effects

Group	Variable	Std Dev	Variance
trustcode	Intercept	0.8877288	0.7880625

```
> coeff_rand = ranef(model)$trustcode
> sd_rand = sqrt(diag(model$variance)[1:length(coeff_rand)])

> df_random = data.frame(effect = names(coeff_rand),
+                        coeff = coeff_rand,
+                        coeff_sd = sd_rand,
+                        hr = exp(coeff_rand),
+                        hr_lower = exp(coeff_rand - 2 * sd_rand),
+                        hr_upper = exp(coeff_rand + 2 * sd_rand),
+                        row.names=NULL)
> write.csv(df_random, file = 'letter_random_effect_sensitivity.csv')
```