

# Multistep Model of Parkinson's

## R Setup

### Data import

```
ages_100 <- c("30", "35", "40", "45", "50", "55",
             "60", "65", "70", "75", "80",
             "85", "90", "95", "100+")

sim_df_ms_full <- read.csv("data/incidence_by_age_multistep_full.csv") %>%
  mutate(agecut5 = fct_relevel(agecut5, ages_100)) %>%
  mutate(agecut5_scaled = agecut5_log - min(agecut5_log)) %>%
  mutate(restriction = if_else(agecut5_numeric <= 80, "modelled", "unmodelled"))

sim_df_ms_sex_full <- read.csv("data/incidence_by_age_multistep_sex_full.csv") %>%
  mutate(agecut5 = fct_relevel(agecut5, ages_100)) %>%
  filter(agecut5_numeric >= 30) %>%
  mutate(agecut5_scaled = agecut5_log - min(agecut5_log)) %>%
  mutate(restriction = if_else(agecut5_numeric <= 80, "modelled", "unmodelled")) %>%
  mutate(sex_restriction = factor(sex) : factor(restriction)) %>%
  mutate(sex = factor(sex, levels=c("Male", "Female"))) %>%
  mutate(sex_restriction = factor(sex_restriction,
                                  levels=c("Male:modelled", "Female:modelled",
                                           "Male:unmodelled", "Female:unmodelled")))

sim_df_ms_model <- sim_df_ms_full %>%
  filter(agecut5_numeric <= 80)

sim_df_ms_sex_model <- sim_df_ms_sex_full %>%
  filter(agecut5_numeric <= 80)
```

### Common constants/code

```
# Colours for modeled/unmodeled data points
colours_modelled <- c("#000000", "#AAAAAA")

# Colours for male/female
colours_mf <- c("#009E73", "#D55E00")

# Colours for male/female modeled/unmodeled
colours_mf_modelled <- c("#009E73", "#D55E00", "#AAAAAA", "#AAAAAA")
```

### Model: Linear

```
mod_ms_linear <- stan_model('stan/multistep_linear.stan')

pars_linear <- c('m', 'c', 'sigma_sq')

data_list_linear <- list(m_prior_mean = 6,
```

```

m_prior_sd = 1,
N=nrow(sim_df_ms_model),
x = sim_df_ms_model$agecut5_scaled,
y_min = sim_df_ms_model$lower_log,
y_max = sim_df_ms_model$upper_log)

fit_linear <- sampling(mod_ms_linear, data = data_list_linear, seed = SEED,
                        pars = pars_linear, chains = 4, iter = 2000,
                        control = list(max_treedepth = 15,adapt_delta=0.99),
                        verbose=TRUE)

##  

## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_linear' NOW.  

##  

## COMPILING MODEL 'multistep_linear' NOW.  

##  

## STARTING SAMPLER FOR MODEL 'multistep_linear' NOW.  

print(fit_linear)

## Inference for Stan model: multistep_linear.  

## 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%   25%   50%   75% 97.5% n_eff Rhat  

## m       6.49    0.00 0.12   6.21   6.42   6.51   6.57  6.69   868 1.00  

## c      -0.60    0.00 0.09  -0.74  -0.66  -0.61  -0.55  -0.39   916 1.00  

## sigma_sq  0.01    0.00 0.01   0.00   0.00   0.01   0.01  0.02   813 1.01  

## lp__   -10.22    0.05 1.35 -13.67 -10.85 -9.91 -9.20 -8.64   750 1.00  

##  

## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:40 2020.  

## 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).

H_LINEAR = bridge_sampler(fit_linear)

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10

```

## Model: Broken-stick

```

mod_ms_bs <- stan_model('stan/multistep_brokenstick.stan')

pars_bs <- c('m_early','m_late','bp','c','sigma_sq')

# wider priors for estimation
data_list_bs <- list(m_early_prior_mean = 5,
                      m_early_prior_sd = 1,

```

```

m_late_prior_mean = 7,
m_late_prior_sd = 1,
bp_prior_mean = 0.5,
bp_prior_sd = 0.3,
N=nrow(sim_df_ms_model),
x = sim_df_ms_model$agecut5_scaled,
y_min = sim_df_ms_model$lower_log,
y_max = sim_df_ms_model$upper_log)

fit_bs <- sampling(mod_ms_bs, data = data_list_bs, seed = SEED,
                    pars = pars_bs, chains = 4, iter = 2000,
                    control = list(max_treedepth = 15,adapt_delta=0.99),
                    verbose=TRUE)

##  

## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_brokenstick' NOW.  

##  

## COMPILING MODEL 'multistep_brokenstick' NOW.  

##  

## STARTING SAMPLER FOR MODEL 'multistep_brokenstick' NOW.  

print(fit_bs)

## Inference for Stan model: multistep_brokenstick.
## 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%   25%   50%   75% 97.5% n_eff Rhat
## m_early    5.21    0.02 0.70   3.78   4.74   5.23   5.72  6.44  1296 1.00
## m_late     6.75    0.01 0.21   6.37   6.61   6.73   6.87  7.21  1327 1.00
## bp         0.41    0.00 0.11   0.22   0.34   0.41   0.48  0.65  1282 1.00
## c        -0.20    0.01 0.22  -0.62  -0.37  -0.20  -0.04  0.22  1165 1.00
## sigma_sq   0.01    0.00 0.01   0.00   0.00   0.01   0.01  0.02  1153 1.01
## lp__     -11.80    0.06 1.86 -16.39 -12.79 -11.37 -10.42 -9.44   865 1.00
##  

## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:42 2020.
## 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).

## Run in console
#pdf("plots/bs_pairs_diagnosis.pdf")
#pairs(fit_bs)
#dev.off()

H_BS = bridge_sampler(fit_bs)

## Warning: 14 of the 2000 log_prob() evaluations on the proposal draws produced
## -Inf/Inf.

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8

```

```

## Iteration: 9
## Iteration: 10
## Iteration: 11
bf_bs = bf(H_BS,H_LINEAR)

```

## Model: Sex Broken-stick

```

# Common slope and intercept by sex
mod_ms_bs_sex_common <- stan_model('stan/multistep_brokenstick.stan')
pars_bs_sex_common <- c('m_early','m_late','bp','c','sigma_sq')

# Allow intercept vary by sex
mod_ms_bs_sex_vc <- stan_model('stan/multistep_brokenstick_sex_intercept.stan')
pars_bs_sex_vc <- c('m_early','m_late','bp','c','c_sex','sigma_sq')

# Allow slope and intercept to vary by sex
mod_ms_bs_sex_vmc <- stan_model('stan/multistep_brokenstick_sex.stan')
pars_bs_sex_vmc <- c('m_early','m_late','bp','c','c_sex','m_sex','sigma_sq',
                      'c_female','m_early_female','m_late_female')

# data and priors - varying intercept/slope
data_list_bs_sex <- list(m_early_prior_mean = 5,
                          m_early_prior_sd = 1,
                          m_late_prior_mean = 7,
                          m_late_prior_sd = 1,
                          bp_prior_mean = 0.5,
                          bp_prior_sd = 0.3,
                          N=nrow(sim_df_ms_sex_model),
                          x = sim_df_ms_sex_model$agecut5_scaled,
                          sex = abs(as.numeric(factor(sim_df_ms_sex_model$sex))-2),
                          y_min = sim_df_ms_sex_model$lower_log,
                          y_max = sim_df_ms_sex_model$upper_log)

# Common slope and intercept to vary by sex
fit_bs_sex_common <- sampling(mod_ms_bs_sex_common, data = data_list_bs_sex,
                               seed = SEED,
                               pars = pars_bs_sex_common, chains = 4, iter = 2000,
                               control = list(max_treedepth = 20,adapt_delta=0.99),
                               verbose=TRUE)

## 
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_brokenstick' NOW.
## 
## COMPILING MODEL 'multistep_brokenstick' NOW.
## 
## STARTING SAMPLER FOR MODEL 'multistep_brokenstick' NOW.

print(fit_bs_sex_common)

## Inference for Stan model: multistep_brokenstick.
## 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%    25%    50%    75%  97.5% n_eff Rhat
## m_early    5.25     0.02 0.71   3.74   4.78   5.30   5.77   6.48  1542     1
## m_late     6.86     0.01 0.45   6.09   6.55   6.82   7.13   7.84  1888     1
## bp         0.46     0.00 0.14   0.22   0.35   0.45   0.55   0.75  1724     1

```

```

## c      -0.21    0.01 0.22  -0.64  -0.36  -0.21  -0.06  0.22  1776    1
## sigma_sq  0.09    0.00 0.03   0.04   0.06   0.08   0.10   0.17  2037    1
## lp__   -38.42    0.05 1.83 -42.79 -39.35 -38.07 -37.11 -35.93 1136    1
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:44 2020.
## 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).

# Allow intercept to vary by sex
fit_bs_sex_vc <- sampling(mod_ms_bs_sex_vc, data = data_list_bs_sex,
                           seed = SEED,
                           pars = pars_bs_sex_vc, chains = 4, iter = 2000,
                           control = list(max_treedepth = 20,adapt_delta=0.99),
                           verbose=TRUE)

##
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_brokenstick_sex_intercept' NOW.
##
## COMPILING MODEL 'multistep_brokenstick_sex_intercept' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_brokenstick_sex_intercept' NOW.
print(fit_bs_sex_vc)

## Inference for Stan model: multistep_brokenstick_sex_intercept.
## 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%   25%   50%   75% 97.5% n_eff Rhat
## m_early    5.18    0.02 0.66  3.76  4.73  5.21  5.69  6.32  1351    1
## m_late     6.75    0.00 0.15  6.49  6.65  6.74  6.84  7.07  1951    1
## bp        0.40    0.00 0.09  0.22  0.33  0.40  0.46  0.59  1654    1
## c       -0.49    0.01 0.21 -0.87 -0.65 -0.49 -0.33 -0.11  1348    1
## c_sex     0.53    0.00 0.04  0.45  0.50  0.53  0.56  0.61  2265    1
## sigma_sq  0.00    0.00 0.00  0.00  0.00  0.00  0.01  0.01  1911    1
## lp__   -14.90    0.05 1.88 -19.52 -15.95 -14.54 -13.47 -12.34 1210    1
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:47 2020.
## 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).

# Allow slope and intercept to vary by sex
fit_bs_sex_vmc <- sampling(mod_ms_bs_sex_vmc, data = data_list_bs_sex,
                            seed = SEED,
                            pars = pars_bs_sex_vmc, chains = 4, iter = 2000,
                            control = list(max_treedepth = 20,adapt_delta=0.99),
                            verbose=TRUE)

##
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_brokenstick_sex' NOW.
##
## COMPILING MODEL 'multistep_brokenstick_sex' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_brokenstick_sex' NOW.
print(fit_bs_sex_vmc)

## Inference for Stan model: multistep_brokenstick_sex.
```

```

## 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%   25%   50%   75% 97.5% n_eff
## m_early      5.13   0.02 0.65  3.78  4.70  5.15  5.59  6.28  1506
## m_late       6.74   0.00 0.19  6.38  6.61  6.73  6.86  7.12  1704
## bp           0.40   0.00 0.09  0.23  0.34  0.40  0.46  0.59  1734
## c            -0.46   0.01 0.23 -0.88 -0.63 -0.46 -0.29 -0.03 1366
## c_sex         0.50   0.00 0.17  0.16  0.38  0.50  0.61  0.86  1309
## m_sex         0.04   0.01 0.23 -0.43 -0.11  0.04  0.18  0.48  1301
## sigma_sq     0.01   0.00 0.00  0.00  0.00  0.00  0.01  0.01  1813
## c_female     0.04   0.00 0.21 -0.35 -0.11  0.04  0.20  0.41  1777
## m_early_female 5.17   0.02 0.65  3.81  4.73  5.20  5.64  6.32  1655
## m_late_female 6.77   0.00 0.18  6.44  6.65  6.77  6.88  7.15  2265
## lp__        -15.26  0.06 2.07 -20.23 -16.42 -14.87 -13.72 -12.40 1148
##
##          Rhat
## m_early      1
## m_late       1
## bp           1
## c            1
## c_sex         1
## m_sex         1
## sigma_sq     1
## c_female     1
## m_early_female 1
## m_late_female 1
## lp__         1
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:50 2020.
## 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).
## DIAGNOSTICS
## Run in console
#pdf("plots/sex_pairs_diagnosis.pdf")
#pairs(fit_bs_sex_vmc)
#dev.off()

H_BS_SEX_COMMON = bridge_sampler(fit_bs_sex_common)

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8

H_BS_SEX_VC = bridge_sampler(fit_bs_sex_vc)

## Warning: 5 of the 2000 log_prob() evaluations on the proposal draws produced
## -Inf/Inf.

## Iteration: 1
## Iteration: 2
## Iteration: 3

```

```

## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
H_BS_SEX_VMC = bridge_sampler(fit_bs_sex_vmc)

## Warning: 2 of the 2000 log_prob() evaluations on the proposal draws produced
## -Inf/Inf.

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10

bf(H_BS_SEX_VC,H_BS_SEX_COMMON)

## Estimated Bayes factor in favor of H_BS_SEX_VC over H_BS_SEX_COMMON: 209889691.01172
bf(H_BS_SEX_VMC,H_BS_SEX_VC)

## Estimated Bayes factor in favor of H_BS_SEX_VMC over H_BS_SEX_VC: 0.03939

```

## Model: Armitage-Doll

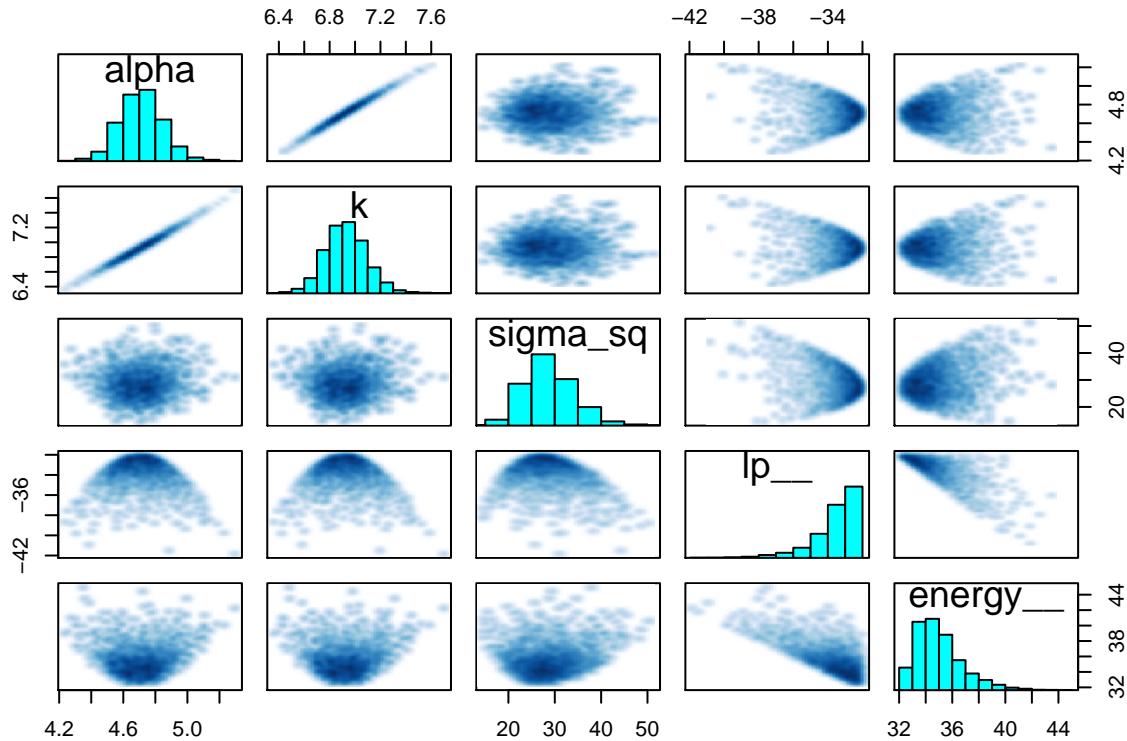
```

##
## Formula: middle ~ (alpha * agecut5_numeric)^(k - 1) * 1e+05
##
## Parameters:
##           Estimate Std. Error t value Pr(>|t|)
## alpha    0.0047029  0.0002309   20.37 7.72e-09 ***
## k       6.9154990  0.2815963   24.56 1.47e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.861 on 9 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 8.947e-06
##
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_ad' NOW.
##
## COMPIILING MODEL 'multistep_ad' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_ad' NOW.
##
## Inference for Stan model: multistep_ad.
## 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%   25%   50%   75% 97.5% n_eff Rhat
## alpha      4.71     0.00 0.14  4.45  4.62  4.71  4.80  4.99  891 1.00
## k          6.93     0.01 0.17  6.62  6.82  6.93  7.03  7.27  884 1.00
## sigma_sq  28.72     0.17 5.37 19.61 24.97 28.19 32.00 40.51  951 1.00
## lp__     -33.54     0.05 1.29 -37.07 -34.06 -33.18 -32.61 -32.12  760 1.01
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:53 2020.
## 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).

```



```

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6

```

## Model: Armitage-Doll Sex

```

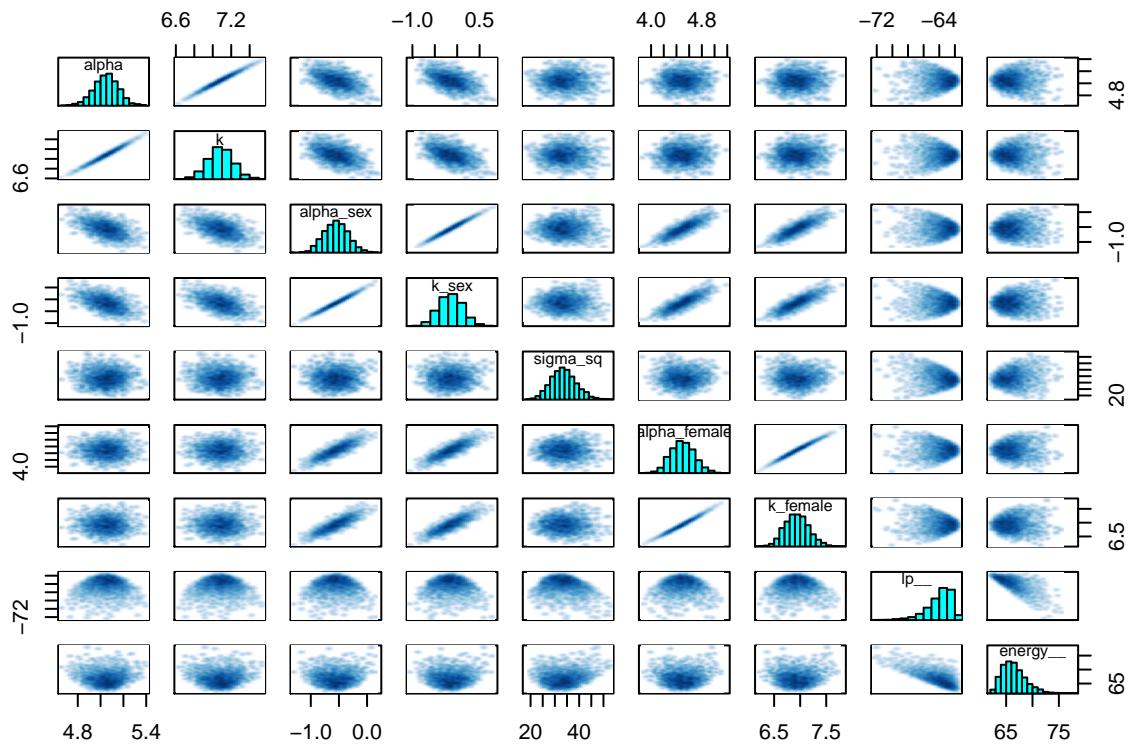
##
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_ad_sex' NOW.
##
## COMPIILING MODEL 'multistep_ad_sex' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_ad_sex' NOW.
##
## Inference for Stan model: multistep_ad_sex.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.

```

```

## 
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## alpha      5.05  0.00 0.11  4.83  4.98  5.05  5.12  5.26  1022 1.00
## k         7.08  0.00 0.14  6.82  6.99  7.08  7.17  7.35  1024 1.00
## alpha_sex -0.55  0.01 0.22 -0.96 -0.70 -0.55 -0.40 -0.12 1163 1.00
## k_sex     -0.15  0.01 0.27 -0.64 -0.33 -0.15  0.03  0.39 1153 1.00
## sigma_sq   33.64 0.12 5.25 24.04 29.99 33.37 37.04 44.51 1876 1.00
## alpha_female 4.50  0.00 0.19  4.15  4.36  4.49  4.62  4.87 1487 1.00
## k_female   6.94  0.01 0.23  6.52  6.77  6.93  7.09  7.41 1493 1.00
## lp__      -63.98 0.05 1.61 -68.13 -64.80 -63.63 -62.81 -61.89 1081 1.01
## 
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:15:57 2020.
## 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).

```



```

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6

```

## Model: Beta

```
# Lets try with nls to get an idea of parameter values to start with
```

```
nlm <- nls(middle ~ (alpha*agecut5_numeric)^(k-1)*(1-beta*agecut5_numeric)*100000, data=sim_df_ms_full, start =
```

```

summary(nlm)

##
## Formula: middle ~ (alpha * agecut5_numeric)^(k - 1) * (1 - beta * agecut5_numeric) *
##      1e+05
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## alpha 6.084e-03 4.199e-04 14.49 5.77e-09 ***
## beta  9.863e-03 8.292e-05 118.94 < 2e-16 ***
## k     7.202e+00 5.639e-01 12.77 2.41e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.39 on 12 degrees of freedom
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 4.371e-06

mod_ms_beta <- stan_model('stan/multistep_beta.stan')

pars_beta <- c('alpha','beta','k','sigma_sq')

data_list_beta <- list(
  alpha_prior_mean = 0,
  alpha_prior_sd = 0.01,
  beta_prior_mean = 0,
  beta_prior_sd = 0.01,
  k_prior_mean = 7,
  k_prior_sd = 2,
  N=nrow(sim_df_ms_full),
  x = sim_df_ms_full$agecut5_numeric,
  y = sim_df_ms_full$middle,
  y_min = sim_df_ms_full$lower,
  y_max = sim_df_ms_full$upper)

initf1 <- function() {
  list(alpha = 0.006, beta = 0.01, k = 7, sigma_sq = 80)
}

fit_beta <- sampling(mod_ms_beta, data = data_list_beta, seed = SEED,
                      pars = pars_beta, chains = 4, iter = 2000, init=initf1,
                      control = list(max_treedepth = 15,adapt_delta=0.99),
                      verbose=TRUE)

##
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_beta' NOW.
##
## COMPILE MODEL 'multistep_beta' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_beta' NOW.

print(fit_beta)

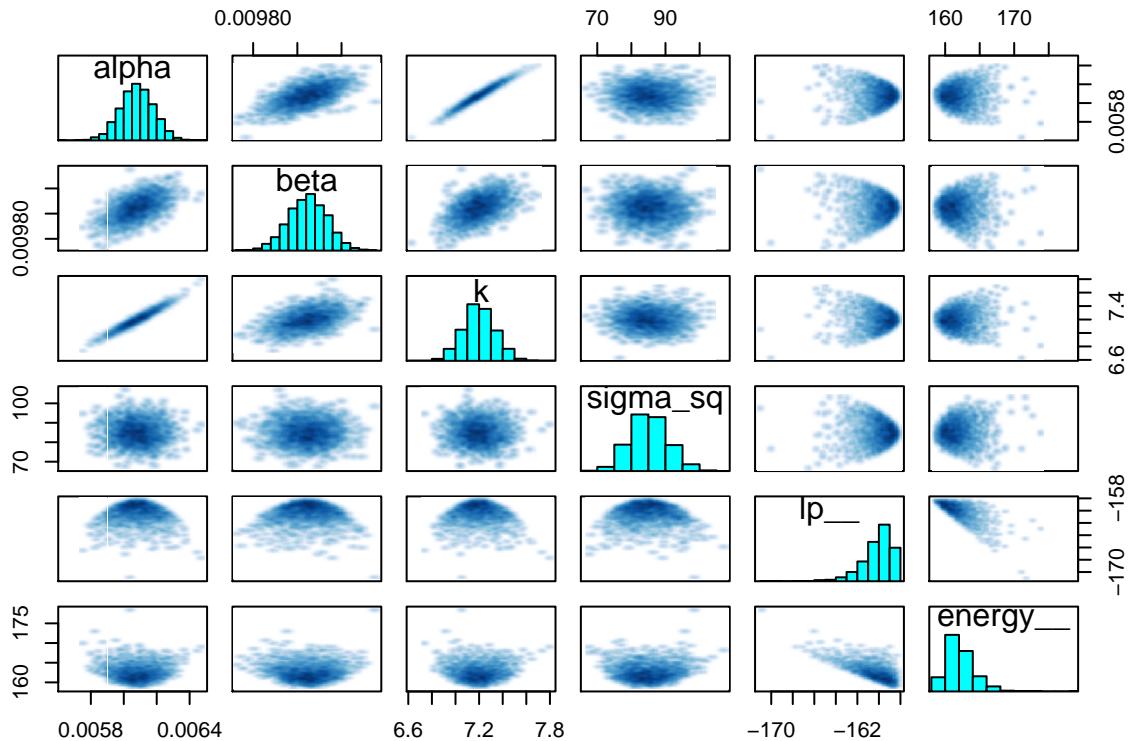
## Inference for Stan model: multistep_beta.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##

```

```

##          mean se_mean    sd   2.5%   25%   50%   75% 97.5% n_eff
## alpha      0.01  0.00 0.00  0.01  0.01  0.01  0.01  0.01 1153
## beta       0.01  0.00 0.00  0.01  0.01  0.01  0.01  0.01 1594
## k          7.20  0.00 0.13  6.94  7.11  7.20  7.29  7.47 1172
## sigma_sq   85.16 0.12 5.52  74.90 81.31 85.00 88.91 95.96 1966
## lp__     -160.15 0.04 1.39 -163.64 -160.84 -159.86 -159.12 -158.42 1507
##             Rhat
## alpha       1
## beta        1
## k           1
## sigma_sq   1
## lp__        1
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:16:02 2020.
## 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).
pairs(fit_beta)

```



```
H_BETA = bridge_sampler(fit_beta)
```

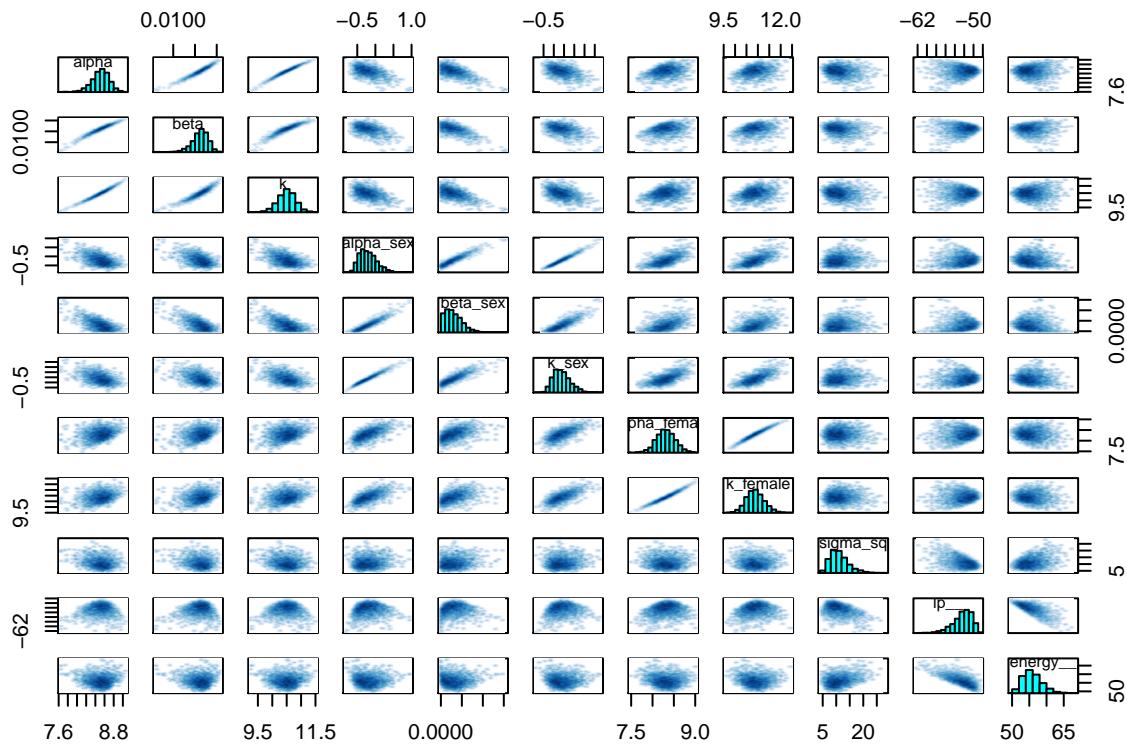
```

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5

```

## Model: Beta Sex

```
##  
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_beta_sex' NOW.  
##  
## COMPILING MODEL 'multistep_beta_sex' NOW.  
##  
## STARTING SAMPLER FOR MODEL 'multistep_beta_sex' NOW.  
  
## Inference for Stan model: multistep_beta_sex.  
## 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%   25%   50%   75% 97.5% n_eff Rhat  
## alpha      8.51    0.01 0.19   8.11   8.40   8.52   8.64   8.85   662 1.00  
## beta       0.01    0.00 0.00    0.01   0.01   0.01   0.01   0.01   635 1.00  
## k         10.50    0.01 0.31   9.85  10.30  10.51  10.70  11.08  704 1.00  
## alpha_sex -0.20    0.01 0.24  -0.57  -0.38  -0.23  -0.05   0.35  699 1.01  
## beta_sex   0.00    0.00 0.00    0.00   0.00   0.00   0.00   0.00  644 1.01  
## k_sex      0.39    0.02 0.43  -0.29   0.08   0.34   0.65   1.34  762 1.00  
## alpha_female 8.31    0.01 0.22   7.88   8.16   8.31   8.46   8.75  1149 1.00  
## k_female   10.89    0.01 0.40  10.13  10.62  10.87  11.14  11.73  1167 1.00  
## sigma_sq   11.15    0.09 3.39   5.99   8.66  10.68  13.03  19.20  1370 1.00  
## lp__      -52.40   0.07 2.08 -57.28 -53.55 -52.06 -50.87 -49.38  836 1.00  
##  
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:16:12 2020.  
## 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).
```



```

## Warning: 14 of the 2000 log_prob() evaluations on the proposal draws produced
## -Inf/Inf.

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10

```

## Model: Susceptibility

```

# Lets try with nls to get an idea of parameter values to start with

nlm <- nls(middle ~ alpha*agecut5_numeric^(k-1)*100000/(1+((1-C)/C)*exp(alpha/k*(agecut5_numeric^k-1))), d

summary(nlm)

## 
## Formula: middle ~ alpha * agecut5_numeric^(k - 1) * 1e+05/(1 + ((1 - C)/C) *
##           exp(alpha/k * (agecut5_numeric^k - 1)))
##
## Parameters:
##             Estimate Std. Error t value Pr(>|t|)
## alpha    1.200e-15  1.520e-15   0.789   0.445
## C        8.024e-02  2.246e-03  35.729  1.48e-13 ***
## k        8.200e+00  2.920e-01  28.081  2.58e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14 on 12 degrees of freedom
##
## Number of iterations to convergence: 33
## Achieved convergence tolerance: 2.017e-06

mod_ms_suscept <- stan_model('stan/multistep_susceptibility.stan')

pars_suscept <- c('alpha','C','k','sigma_sq')

data_list_suscept <- list(C_prior_mean = 5,
                           C_prior_sd = 5,
                           alpha_prior_mean = 0,
                           alpha_prior_sd = 10,
                           k_prior_mean = 7,
                           k_prior_sd = 2,
                           N=nrow(sim_df_ms_full),
                           x = sim_df_ms_full$agecut5_numeric,
                           y = sim_df_ms_full$middle,
                           y_min = sim_df_ms_full$lower,
                           y_max = sim_df_ms_full$upper)

init_suscept <- function() {
  list(alpha = 1, C = 5, k = 7, sigma_sq = 80)
}

```

```

}

fit_suscept <- sampling(mod_ms_suscept, data = data_list_suscept, seed = SEED,
                        pars = pars_suscept, chains = 4, iter = 2000,
                        init = init_suscept,
                        control = list(max_treedepth = 15,adapt_delta=0.99),
                        verbose=TRUE)

## 
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_susceptibility' NOW.
##
## COMPILING MODEL 'multistep_susceptibility' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_susceptibility' NOW.

print(fit_suscept)

## Inference for Stan model: multistep_susceptibility.
## 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%    25%    50%    75%  97.5% n_eff Rhat
## alpha      2.08     0.04  1.28   0.54   1.19   1.80   2.63   5.40  1072    1
## C         8.07     0.00  0.11   7.85   7.99   8.07   8.14   8.29  1305    1
## k         8.11     0.00  0.14   7.85   8.02   8.11   8.20   8.39  1078    1
## sigma_sq  45.03    0.13  5.41  35.21  41.25  44.81  48.57  55.78 1803    1
## lp__     -62.71     0.04  1.39 -66.24 -63.41 -62.40 -61.68 -60.92 1303    1
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:16:24 2020.
## 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).

summary(fit_suscept)

## $summary
##          mean   se_mean     sd    2.5%    25%    50%
## alpha      2.081815 0.039179839 1.282995  0.5392389  1.193651  1.795499
## C         8.067079 0.003020384 0.109130   7.8519989  7.993767  8.069164
## k         8.111873 0.004147213 0.136134   7.8527848  8.019597  8.107358
## sigma_sq  45.025049 0.127354703 5.408069  35.2095092  41.246230  44.805154
## lp__     -62.710318 0.038514858 1.390079 -66.2350004 -63.406137 -62.398196
##          75%    97.5%   n_eff   Rhat
## alpha      2.628081  5.400695 1072.321 1.002264
## C         8.140125  8.287528 1305.463 1.003382
## k         8.201420  8.385170 1077.509 1.002887
## sigma_sq  48.569274 55.781604 1803.244 1.001258
## lp__     -61.679687 -60.924585 1302.635 1.000584
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter      mean       sd    2.5%    25%    50%    75%
## alpha      2.093684 1.3802858  0.538945  1.167477  1.797439  2.581022
## C         8.067792 0.1142621   7.853162  7.990296  8.076368  8.144100
## k         8.112822 0.1384938   7.840447  8.023220  8.106299  8.207138

```

```

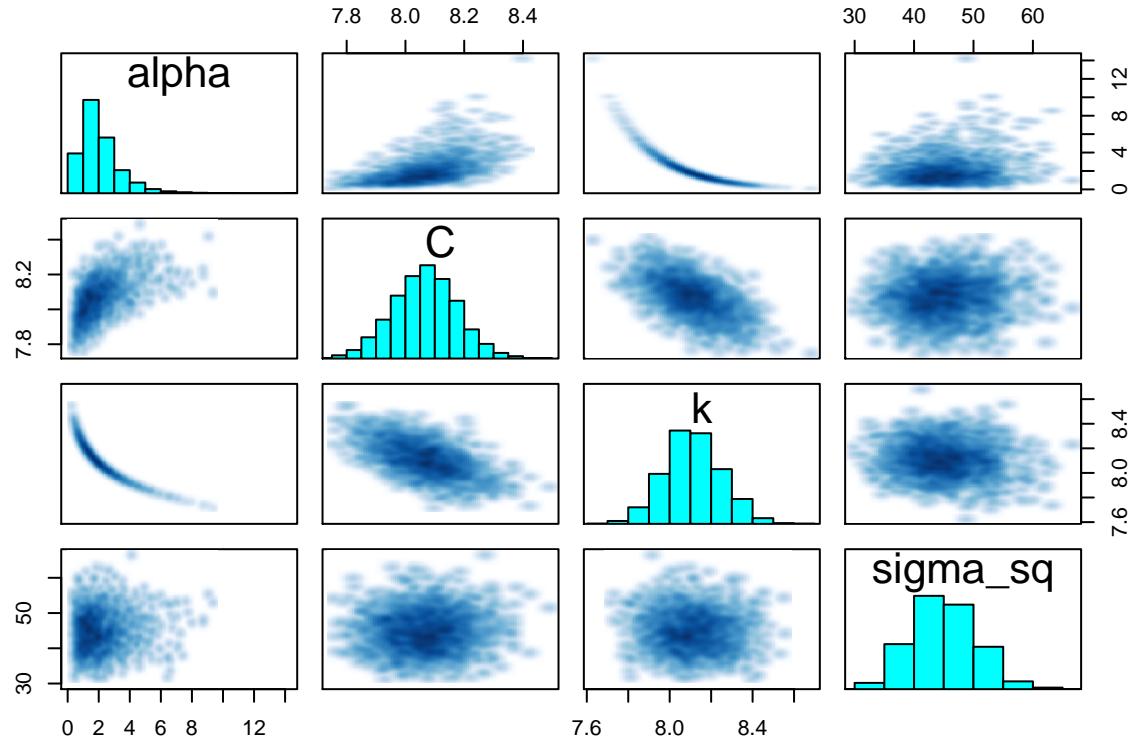
##   sigma_sq  45.110722 5.3792350  35.395907  41.243943  44.965590  48.842416
##   lp__     -62.732627 1.3987597 -66.147000 -63.403521 -62.409392 -61.705184
##       stats
## parameter    97.5%
## alpha        5.65894
## C            8.29165
## k             8.38381
## sigma_sq    55.19920
## lp__      -60.93106
##
## , , chains = chain:2
##
##       stats
## parameter    97.5%
## alpha        2.160689 1.3276493  0.544938  1.230762  1.790992  2.704708
## C            8.074448 0.1060640  7.851939  8.010119  8.077895  8.140433
## k             8.103344 0.1358151  7.839561  8.009751  8.106773  8.194770
## sigma_sq    45.332280 5.5322662  35.073604  41.543817  44.938530  48.793776
## lp__      -62.702747 1.4418870 -66.317680 -63.320017 -62.374462 -61.647994
##       stats
## parameter    97.5%
## alpha        5.687106
## C            8.290910
## k             8.379733
## sigma_sq    57.008280
## lp__      -60.923386
##
## , , chains = chain:3
##
##       stats
## parameter    97.5%
## alpha        2.006148 1.2331694  0.5004308  1.130323  1.786528  2.556852
## C            8.059005 0.1078391  7.8398162  7.987759  8.055518  8.130450
## k             8.120893 0.1379884  7.8656126  8.027199  8.109833  8.211595
## sigma_sq    45.077366 5.3460326  35.7464514  41.388825  44.687018  48.615211
## lp__      -62.697360 1.3785139 -66.1448485 -63.402903 -62.398196 -61.669865
##       stats
## parameter    97.5%
## alpha        5.185077
## C            8.287246
## k             8.402608
## sigma_sq    56.182834
## lp__      -60.923839
##
## , , chains = chain:4
##
##       stats
## parameter    97.5%
## alpha        2.066739 1.1782938  0.5772255  1.206438  1.806812  2.666874
## C            8.067071 0.1077890  7.8593262  7.994097  8.068732  8.143767
## k             8.110431 0.1317621  7.8701392  8.016866  8.105328  8.196396
## sigma_sq    44.579827 5.3527057  35.0301881  40.844941  44.399325  48.130723
## lp__      -62.708537 1.3410786 -65.9537491 -63.435713 -62.409365 -61.698327
##       stats
## parameter    97.5%
## alpha        4.926724

```

```

##   C          8.269402
##   k          8.370048
## sigma_sq  55.019893
## lp__     -60.927572
pairs(fit_suscept,pars=pars_suscept)

```



```
H_SUSCEPT = bridge_sampler(fit_suscept)
```

```

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
bf(H_SUSCEPT,H_AD)

```

```
## Estimated Bayes factor in favor of H_SUSCEPT over H_AD: 0.00000
```

```
bf(H_BETA,H_AD)
```

```
## Estimated Bayes factor in favor of H_BETA over H_AD: 0.00000
```

```
bf(H_SUSCEPT,H_BETA)
```

```
## Estimated Bayes factor in favor of H_SUSCEPT over H_BETA: 100685273026986911994395880325120.00000
```

## Model: Susceptibility fixed C

```
# Iterate over values of C, to see how plausible alterative values of C are

mod_ms_suscept_fixed <- stan_model('stan/multistep_susceptibility_fixed_C.stan')

pars_suscept_fixed <- c('alpha', 'k', 'sigma_sq')

data_list_suscept_fixed <- list(C = 3,
                                 alpha_prior_mean = 0,
                                 alpha_prior_sd = 10,
                                 k_prior_mean = 7,
                                 k_prior_sd = 2,
                                 N=nrow(sim_df_ms_full),
                                 x = sim_df_ms_full$agecut5_numeric,
                                 y = sim_df_ms_full$middle,
                                 y_min = sim_df_ms_full$lower,
                                 y_max = sim_df_ms_full$upper)

init_suscept_fixed <- function() {
  list(alpha = 1, k = 7, sigma_sq = 80)
}

fit_suscept_fixed <- sampling(mod_ms_suscept_fixed,
                               data = data_list_suscept_fixed,
                               seed = SEED,
                               pars = pars_suscept_fixed,
                               chains = 4, iter = 2000,
                               init = init_suscept_fixed,
                               control = list(max_treedepth = 15,
                                              adapt_delta=0.99),
                               verbose=TRUE)

##  
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_susceptibility_fixed_C' NOW.  
##  
## COMPILING MODEL 'multistep_susceptibility_fixed_C' NOW.  
##  
## STARTING SAMPLER FOR MODEL 'multistep_susceptibility_fixed_C' NOW.  
H_SUSCEPT_FIXED = bridge_sampler(fit_suscept_fixed)

## Iteration: 1  
## Iteration: 2  
## Iteration: 3  
## Iteration: 4  
## Iteration: 5  
## Iteration: 6  
## Iteration: 7  
## Iteration: 8  
## Iteration: 9  
## Iteration: 10  
## Iteration: 11  
## Iteration: 12  
## Iteration: 13  
## Iteration: 14
```

```

bf(H_SUSCEPT_FIXED,H_AD)

## Estimated Bayes factor in favor of H_SUSCEPT_FIXED over H_AD: 0.00000

Model: Susceptibility Sex

mod_ms_suscept_sex <- stan_model('stan/multistep_susceptibility_sex.stan')

pars_suscept_sex <- c('alpha','C','k',
                      'alpha_sex','C_sex','k_sex',
                      'alpha_female','C_female','k_female','sigma_sq')

data_list_suscept_sex <- list(C_prior_mean = 5,
                                C_prior_sd = 5,
                                alpha_prior_mean = 0,
                                alpha_prior_sd = 10,
                                k_prior_mean = 7,
                                k_prior_sd = 2,
                                N=nrow(sim_df_ms_sex_full),
                                DEBUG=0,
                                x = sim_df_ms_sex_full$agecut5_numeric,
                                y = sim_df_ms_sex_full$middle,
                                sex = abs(as.numeric(factor(sim_df_ms_sex_full$sex))-1),
                                y_min = sim_df_ms_sex_full$lower,
                                y_max = sim_df_ms_sex_full$upper)

init_ms_suspect_sex <- function() {
  list(alpha = 2, C = 8, k = 8, alpha_sex=0, C_sex=0, k_sex=0, sigma_sq = 80)
}

fit_suscept_sex <- sampling(mod_ms_suscept_sex, data = data_list_suscept_sex,
                             seed = SEED, init = init_ms_suspect_sex,
                             pars = pars_suscept_sex, chains = 4, iter = 2000,
                             control = list(max_treedepth = 15,adapt_delta=0.99),
                             verbose=TRUE)

## 
## CHECKING DATA AND PREPROCESSING FOR MODEL 'multistep_susceptibility_sex' NOW.
##
## COMPILING MODEL 'multistep_susceptibility_sex' NOW.
##
## STARTING SAMPLER FOR MODEL 'multistep_susceptibility_sex' NOW.

## Warning: There were 12 divergent transitions after warmup. Increasing adapt_delta above 0.99 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## Warning: Examine the pairs() plot to diagnose sampling problems

print(fit_suscept_sex)

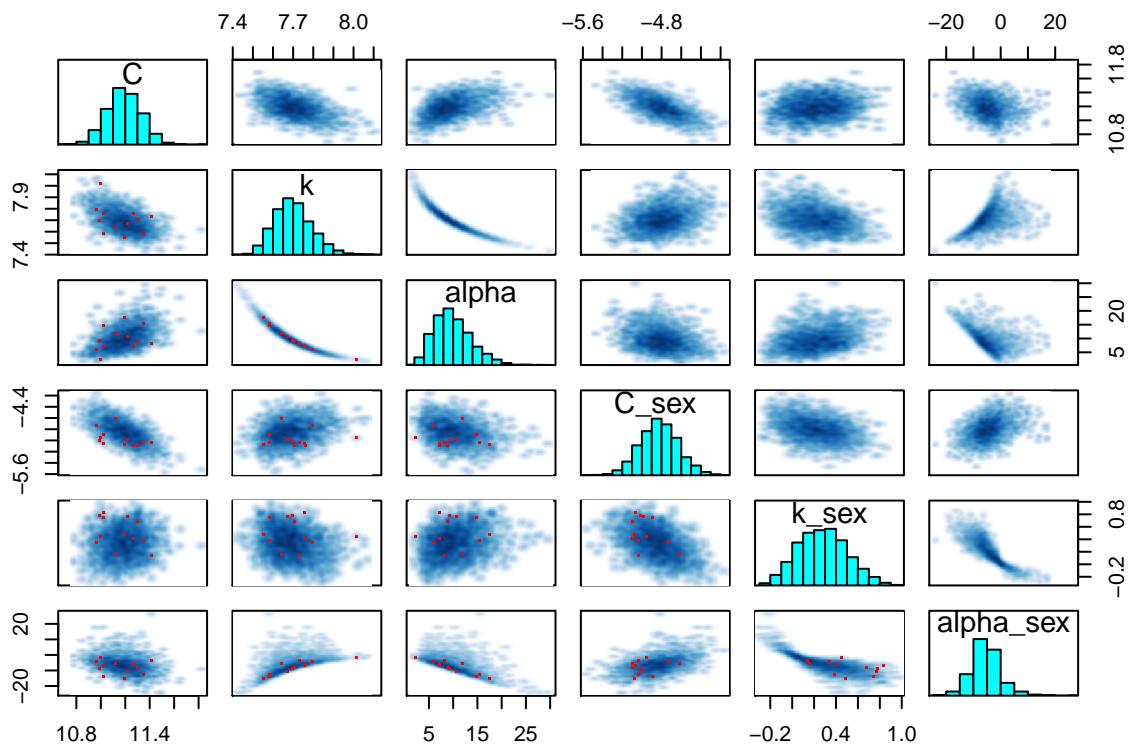
## Inference for Stan model: multistep_susceptibility_sex.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd   2.5%    25%    50%    75%  97.5% n_eff
## alpha      9.94   0.13  3.85   3.86   7.15   9.48  12.28  18.58   922
## C         11.19   0.00  0.14  10.92  11.10  11.19  11.28  11.45  1403
## k          7.70   0.00  0.09   7.54   7.63   7.69   7.76   7.90   963

```

```

## alpha_sex      -5.50    0.19  5.29   -15.36   -8.75   -5.72   -2.55    6.31    802
## C_sex         -4.82    0.01  0.19   -5.20   -4.95   -4.82   -4.70   -4.43   1341
## k_sex          0.28    0.01  0.22   -0.13    0.12    0.28    0.43    0.73    876
## alpha_female   4.44    0.14  4.26    0.49    1.54    3.01    5.95   16.23   962
## C_female      6.36    0.00  0.14    6.09    6.27    6.37    6.46    6.64   1957
## k_female       7.98    0.01  0.21    7.59    7.82    7.98    8.13    8.40   1026
## sigma_sq       83.32   0.12  5.62   72.70   79.48   83.11   87.14   94.80  2217
## lp__        -181.94   0.07  2.02  -186.80  -183.10  -181.65  -180.42  -178.99  933
##
## Rhat
## alpha        1.00
## C            1.01
## k            1.01
## alpha_sex    1.01
## C_sex        1.01
## k_sex        1.00
## alpha_female 1.00
## C_female     1.00
## k_female     1.00
## sigma_sq     1.00
## lp__         1.00
##
## Samples were drawn using NUTS(diag_e) at Thu Dec 17 10:17:20 2020.
## 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).
pairs(fit_suscept_sex,pars=c("C","k","alpha","C_sex","k_sex","alpha_sex"))

```



```

## Run in console
pdf("plots/suspect_sex_pairs_diagnosis.pdf")
pairs(fit_suspect_sex,pars=c("C","k","alpha","C_sex","k_sex","alpha_sex"))
dev.off()

## pdf
## 2
#H_SUSCEPT = bridge_sampler(fit_suscept)

#bf(H_SUSCEPT,H_AD)
#bf(H_BETA,H_AD)
#bf(H_SUSCEPT,H_BETA)

```

### Plot: Age-specific incidence non-log

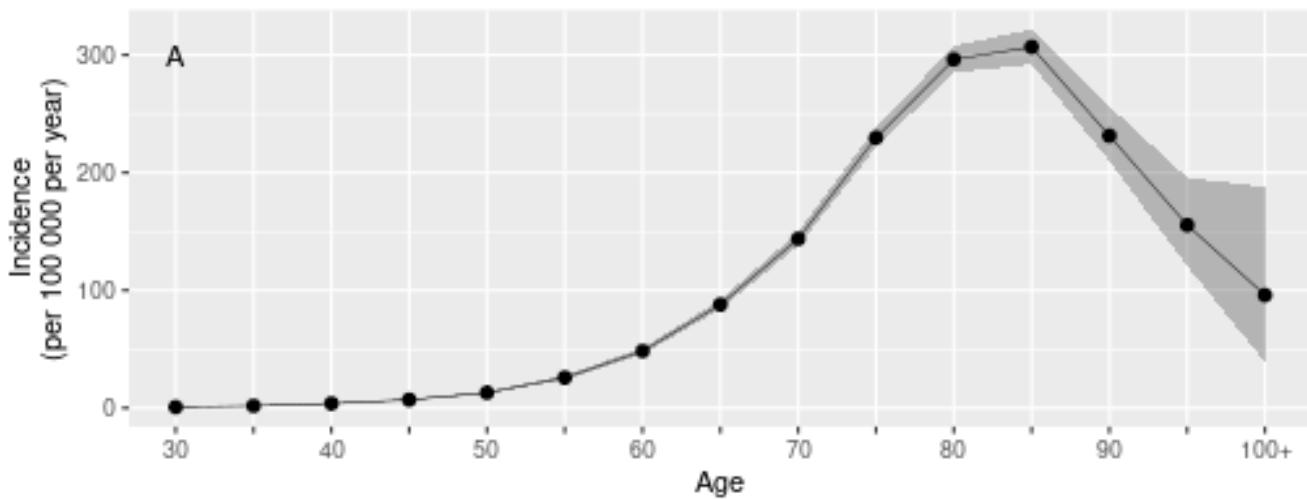
```

fig2_age_specific_incidence <- sim_df_ms_full %>%
  ggplot(aes(x=agecut5,y=middle,group="incidence"))+
  geom_ribbon(aes(ymin=lower,ymax=upper),alpha=0.3)+
  scale_x_discrete("Age", labels = c("30","","",
                                     "40","","" , "50","","",
                                     "60","","" , "70","","",
                                     "80","","" , "90","","",
                                     "100+"))

) +
  geom_line(size=0.2)+geom_point(size=2,fill="grey")+
  ylab("Incidence\n(per 100 000 per year)")+
  scale_shape_manual(values=c(21))+
  xlab("Age")+
  theme(legend.position = "none")+
  annotate("text",x=1,y=300,label="A")

plot(fig2_age_specific_incidence)

```



```

ggsave("plots/multistep-incidence-by-age.pdf",width=5,height=3)
ggsave("plots/multistep-incidence-by-age.png",width=5,height=3)

```

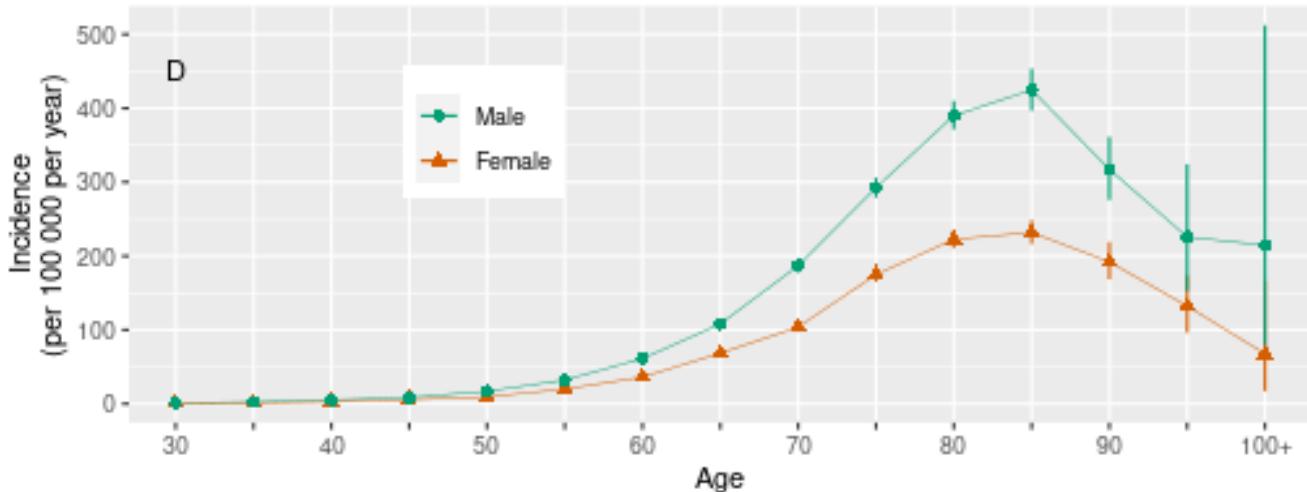
### Plot: Age-specific incidence by sex non-log

```

fig2_age_sex_specific_incidence <- sim_df_ms_sex_full %>%
  ggplot(aes(x=agecut5,y=middle,colour=sex,group=sex,shape=sex))+ 
  geom_errorbar(aes(ymin=lower,ymax=upper), width=0) +
  scale_x_discrete("Age", labels = c("30","", "40","", "50","", "60","", "70","", "80","", "90","", "100+"))
))+
  geom_line(size=0.2)+geom_point(size=2,fill="grey")+
  ylab("Incidence\n(per 100 000 per year)")+
  #scale_shape_manual(values=c(21))+
  scale_color_manual(values = colours_mf)+
  xlab("Age") +
  theme(legend.position = c(0.3,0.7),
        legend.title = element_blank()) +
  annotate("text",x=1,y=450,label="D")

plot(fig2_age_sex_specific_incidence)

```



```

ggsave("plots/multistep-incidence-by-sex-age.pdf",width=5,height=3)
ggsave("plots/multistep-incidence-by-sex-age.png",width=5,height=3)

```

## Plot: Linear model

```

# Extract fitted values from model
values_l = summary(fit_linear)$summary

start=0.0
end = max(sim_df_ms_model$agecut5_scaled)
m = values_l[1,1]
c = values_l[2,1]

fig2_log_linear <- sim_df_ms_full %>%
  ggplot(aes(x=agecut5_scaled,y=middle_log,
             group=type,colour=restriction))+
  geom_errorbar(aes(ymin=lower_log,ymax=upper_log), width=0) +

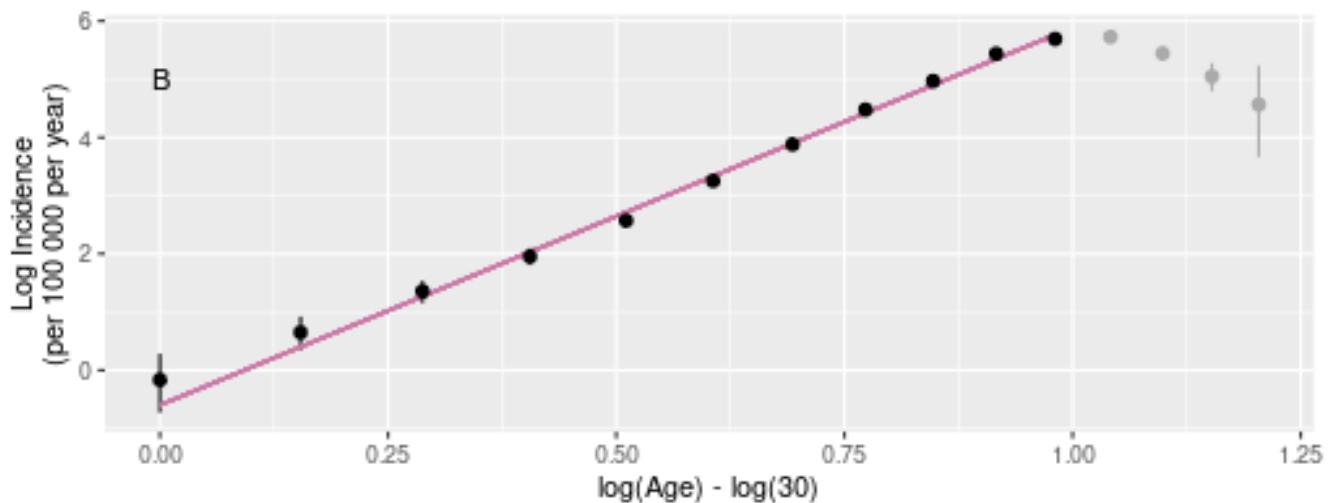
```

```

ylab("Log Incidence\n(per 100 000 per year)") +
xlab("log(Age) - log(30)") +
theme(legend.position = "none") +
geom_segment(x=start,y=c+m*start,
             xend=end,yend=c+m*end,colour="#CC79A7")+
geom_point(size=2) +
scale_colour_manual(values = colours_modelled)+
annotate("text",x=0,y=5,label="B")

plot(fig2_log_linear)

```



```

ggsave("plots/multistep-log-relationship.pdf",width=6,height=4)
ggsave("plots/multistep-log-relationship.png",width=6,height=4)

```

### Plot: Broken-stick model

```

# Extract fitted values from model
values_bs = summary(fit_bs)$summary

m_early=values_bs[1,1]
m_late=values_bs[2,1]
bp=values_bs[3,1]
c=values_bs[4,1]

offset = 0.25

x1 = start
y1 = c+m_early*x1

x2 = bp
y2 = c+m_early*x2

#x2a = bp+offset
#y2a = c+m_early*x2a

x2a = end
y2a = c+m_early*x2a

#x2b = bp-offset

```

```

#y2b = y2-m_late*offset

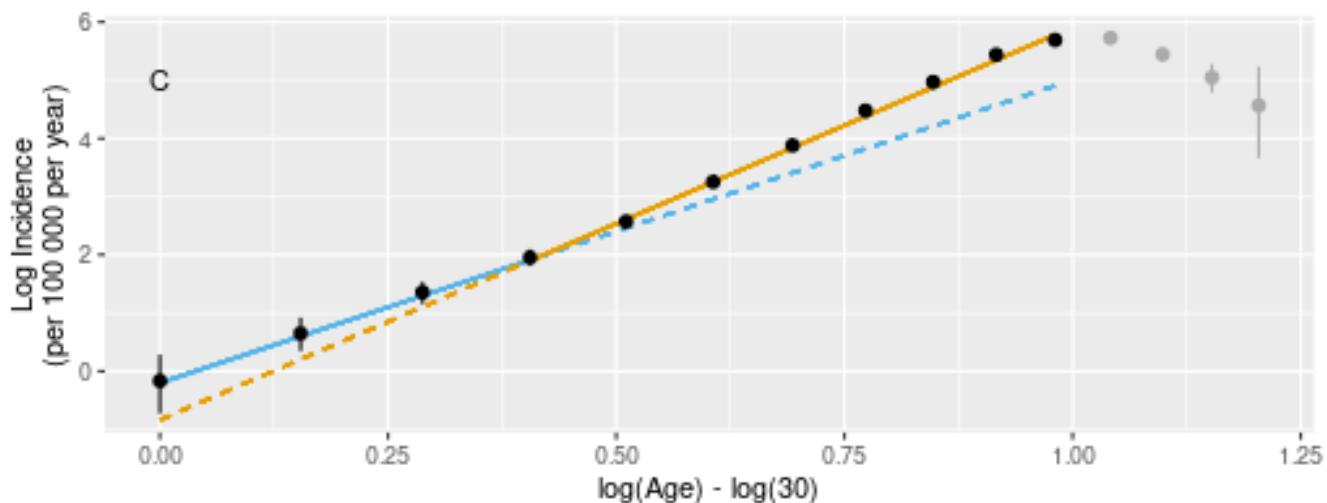
x2b = start
y2b = y2-m_late*(bp-start)

x3 = end
y3 = y2+(end-bp)*m_late

fig2_log_broken_stick <- sim_df_ms_full %>%
  ggplot(aes(x=agecut5_scaled,y=middle_log,
             group=type,colour=restriction))+ 
  geom_errorbar(aes(ymin=lower_log,ymax=upper_log), width=0)+ 
  ylab("Log Incidence\n(per 100 000 per year)")+ 
  xlab("log(Age) - log(30)")+ theme(legend.position = "none")+
  geom_segment(x=x1,y=y1,
               xend=x2,yend=y2,colour="#56B4E9")+
  geom_segment(x=x2,y=y2,
               xend=x2a,yend=y2a,colour="#56B4E9",linetype="dashed")+
  geom_segment(x=x2b,y=y2b,
               xend=x2,yend=y2,colour="#E69F00",linetype="dashed")+
  geom_segment(x=x2,y=y2,
               xend=x3,yend=y3,colour="#E69F00")+
  geom_point(size=2) +
  scale_colour_manual(values = colours_modelled)+
  annotate("text",x=0,y=5,label="C")

plot(fig2_log_broken_stick)

```



```

ggsave("plots/multistep-log-relationship-bs.pdf",width=6,height=4)
ggsave("plots/multistep-log-relationship-bs.png",width=6,height=4)

```

## Plot-HRC: Broken-stick model

```

# Extract fitted values from model
values_bs = summary(fit_bs)$summary

m_early=values_bs[1,1]
m_late=values_bs[2,1]
bp=values_bs[3,1]

```

```

c=values_bs[4,1]

offset = 0.25

x1 = start
y1 = c+m_early*x1

x2 = bp
y2 = c+m_early*x2

x2a = bp+offset
y2a = c+m_early*x2a

x2b = bp-offset
y2b = y2-m_late*offset

x3 = end
y3 = y2+(end-bp)*m_late

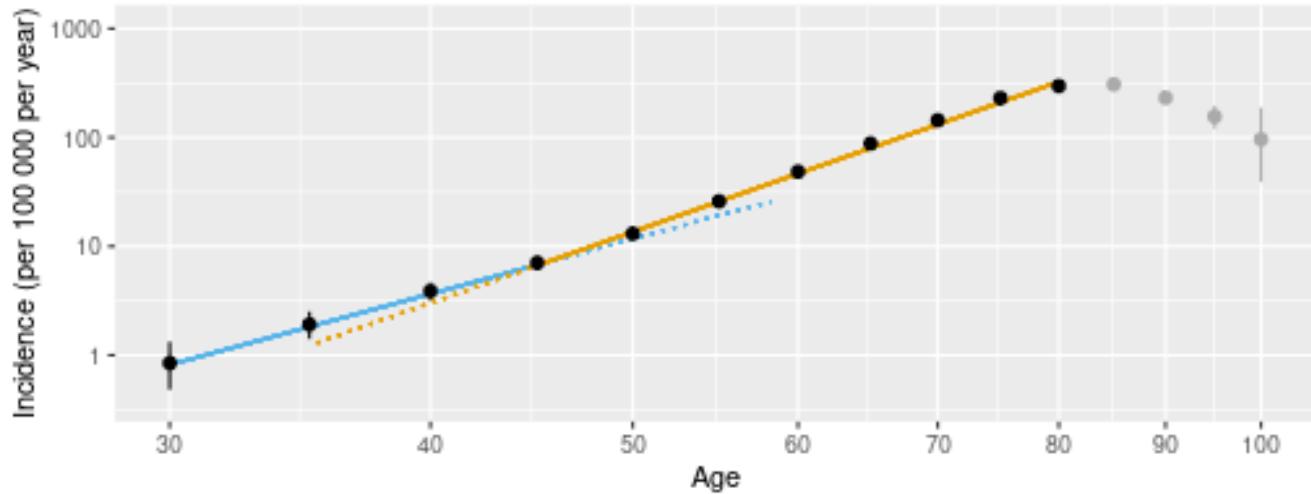
x_axis <- data.frame(age_label=seq(30,100,by=10)) %>%
  mutate(age_x = log(age_label)-log(30))

y_axis <- data.frame(inc_label=c(1,10,100,1000)) %>%
  mutate(inc_y=log(inc_label))

fig2_log_broken_stick_HRC <- sim_df_ms_full %>%
  ggplot(aes(x=agecut5_scaled,y=middle_log,
             group=type,colour=restriction))+ 
  geom_errorbar(aes(ymin=lower_log,ymax=upper_log), width=0)+ 
  scale_x_continuous(breaks = x_axis$age_x,
                     labels = x_axis$age_label,
                     name = "Age")+
  scale_y_continuous(breaks = y_axis$inc_y,
                     labels = y_axis$inc_label,
                     limits = c(-1,7),
                     name = "Incidence (per 100 000 per year)")+
  theme(legend.position = "none")+
  geom_segment(x=x1,y=y1,
               xend=x2,yend=y2,colour="#56B4E9")+
  geom_segment(x=x2,y=y2,
               xend=x2a,yend=y2a,colour="#56B4E9",linetype="dotted")+
  geom_segment(x=x2b,y=y2b,
               xend=x2,yend=y2,colour="#E69F00",linetype="dotted")+
  geom_segment(x=x2,y=y2,
               xend=x3,yend=y3,colour="#E69F00")+
  geom_point(size=2) +
  scale_colour_manual(values = colours_modelled)

plot(fig2_log_broken_stick_HRC)

```



```
ggsave("plots/multistep-log-relationship-bs-HRC.pdf", width=6, height=4)
ggsave("plots/multistep-log-relationship-bs-HRC.png", width=6, height=4)
```

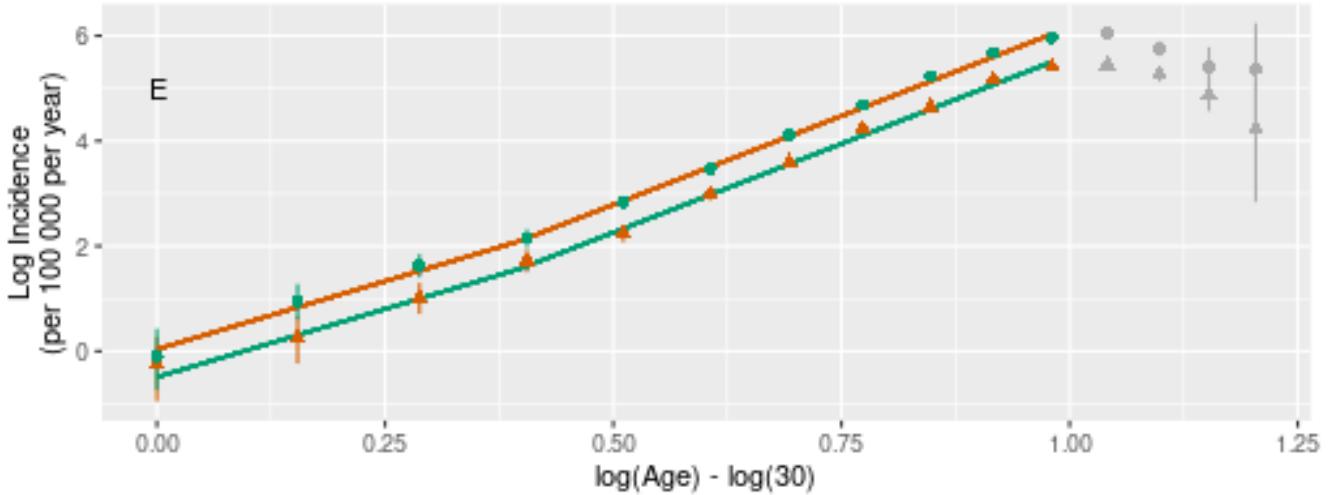
### Plot: Broken-stick model by sex

```
# Extract fitted values from model
values_bs = summary(fit_bs_sex_vc)$summary

m_early=values_bs[1,1]
m_late=values_bs[2,1]
bp=values_bs[3,1]
c=values_bs[4,1]
c_sex=values_bs[5,1]

fig2_log_sex <- sim_df_ms_sex_full %>%
  filter(agecut5_numeric >= 30) %>%
  ggplot(aes(x=agecut5_scaled,y=middle_log,group=sex,
             shape=sex,colour=sex_restriction))+
  geom_errorbar(aes(ymin=lower_log,ymax=upper_log), width=0)+
  ylab("Log Incidence\n(per 100 000 per year)")+
  xlab("log(Age) - log(30)") +
  #theme(legend.position = c(0.8,0.2),
  #      legend.title = element_blank()) +
  theme(legend.position = "none") +
  scale_color_manual(values = colours_mf_modelled)+
  geom_segment(x=start,y=c+m_early*start,
               xend=bp,yend=c+m_early*bp,colour=colours_mf[1])+
  geom_segment(x=bp,y=c+m_early*bp,
               xend=end,yend=c+m_early*bp+(end-bp)*m_late,colour=colours_mf[1])+
  geom_segment(x=start,y=c+c_sex+m_early*start,
               xend=bp,yend=c+c_sex+m_early*bp,colour=colours_mf[2])+
  geom_segment(x=bp,y=c+c_sex+m_early*bp,
               xend=end,yend=c+c_sex+m_early*bp+
                 (end-bp)*m_late,colour=colours_mf[2])+
  guides(colour=FALSE) +
  geom_point(size=2) +
  annotate("text",x=0,y=5,label="E")
```

```
print(fig2_log_sex)
```



```
ggsave("plots/multistep-log-relationship-sex.pdf",width=6,height=4)
ggsave("plots/multistep-log-relationship-sex.png",width=6,height=4)
```

### Plot-HRC: broken-stick model by sex

```
# Extract fitted values from model
values_bs = summary(fit_bs_sex_vc)$summary

m_early=values_bs[1,1]
m_late=values_bs[2,1]
bp=values_bs[3,1]
c=values_bs[4,1]
c_sex=values_bs[5,1]

x_axis <- data.frame(age_label=seq(30,100,by=10)) %>%
  mutate(age_x = log(age_label)-log(30))

y_axis <- data.frame(inc_label=c(1,10,100,1000)) %>%
  mutate(inc_y=log(inc_label))

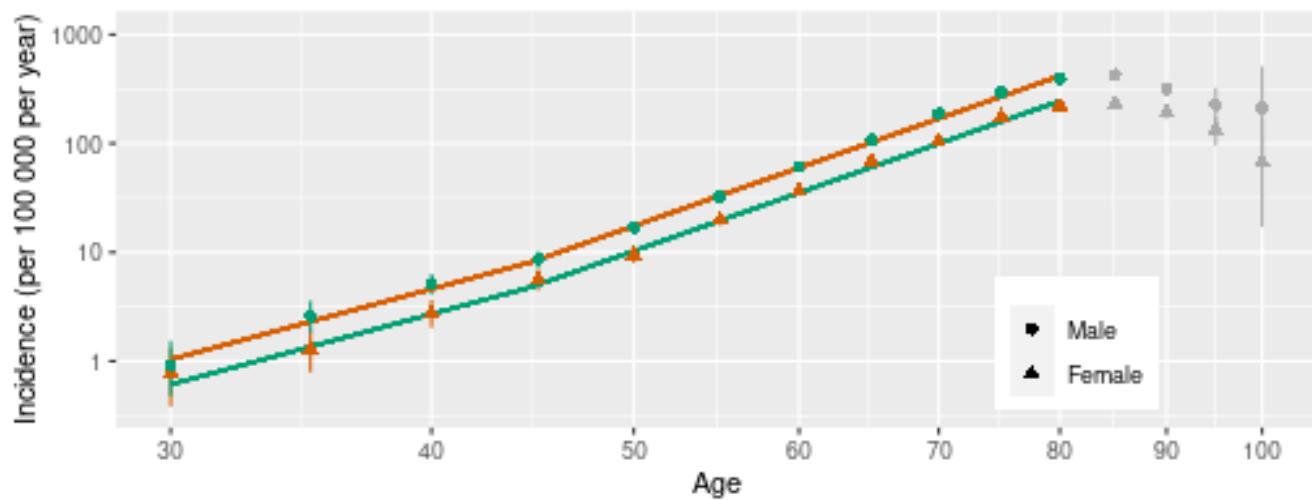
fig2_log_sex_HRC <- sim_df_ms_sex_full %>%
  mutate(sex = factor(sex,levels=c("Male","Female"))) %>%
  filter(agecut5_numeric >= 30) %>%
  ggplot(aes(x=agecut5_scaled,y=middle_log,group=sex,
             shape=sex,colour=sex_restriction))+ 
  geom_errorbar(aes(ymin=lower_log,ymax=upper_log), width=0)+ 
  scale_x_continuous(breaks = x_axis$age_x,
                     labels = x_axis$age_label,
                     name = "Age")+
  scale_y_continuous(breaks = y_axis$inc_y,
                     labels = y_axis$inc_label,
                     limits = c(-1,7),
                     name = "Incidence (per 100 000 per year)")+
  theme(legend.position = c(0.8,0.2),
        legend.title = element_blank())+
  guides(colour=FALSE) +
```

```

scale_color_manual(values = colours_mf_modelled)+
geom_segment(x=start,y=c+m_early*start,
             xend=bp,yend=c+m_early*bp,colour=colours_mf[1])++
geom_segment(x=bp,y=c+m_early*bp,
             xend=end,yend=c+m_early*bp+(end-bp)*m_late,colour=colours_mf[1])++
geom_segment(x=start,y=c+c_sex+m_early*start,
             xend=bp,yend=c+c_sex+m_early*bp,colour=colours_mf[2])++
geom_segment(x=bp,y=c+c_sex+m_early*bp,
             xend=end,yend=c+c_sex+m_early*bp+(end-bp)*m_late,colour=colours_mf[2])++
geom_point(size=2)

print(fig2_log_sex_HRC)

```



```

ggsave("plots/multistep-log-relationship-sex-HRC.pdf",width=6,height=4)
ggsave("plots/multistep-log-relationship-sex-HRC.png",width=6,height=4)

```

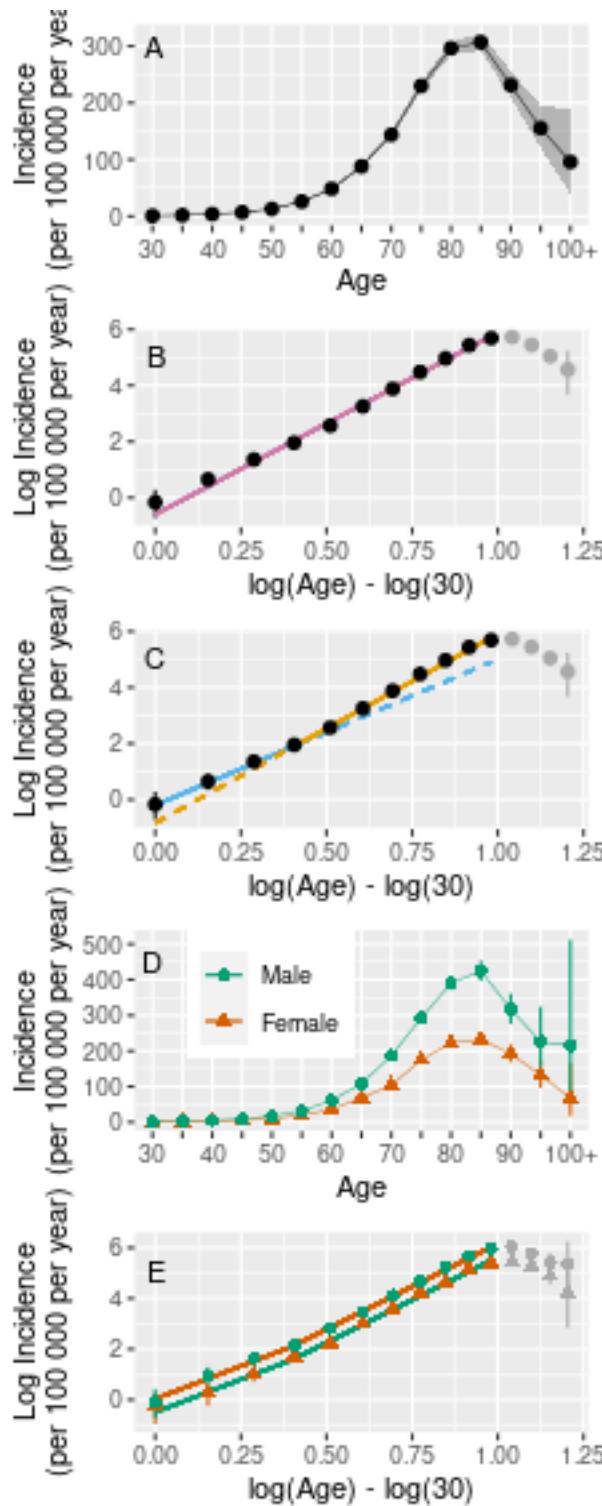
## Plot: Figure 2

```

figure_2 = cowplot::plot_grid(fig2_age_specific_incidence,
                             fig2_log_linear,
                             fig2_log_broken_stick,
                             fig2_age_sex_specific_incidence,
                             fig2_log_sex,
                             align = 'v',
                             axis = 'bt', ncol = 1)

plot(figure_2)

```



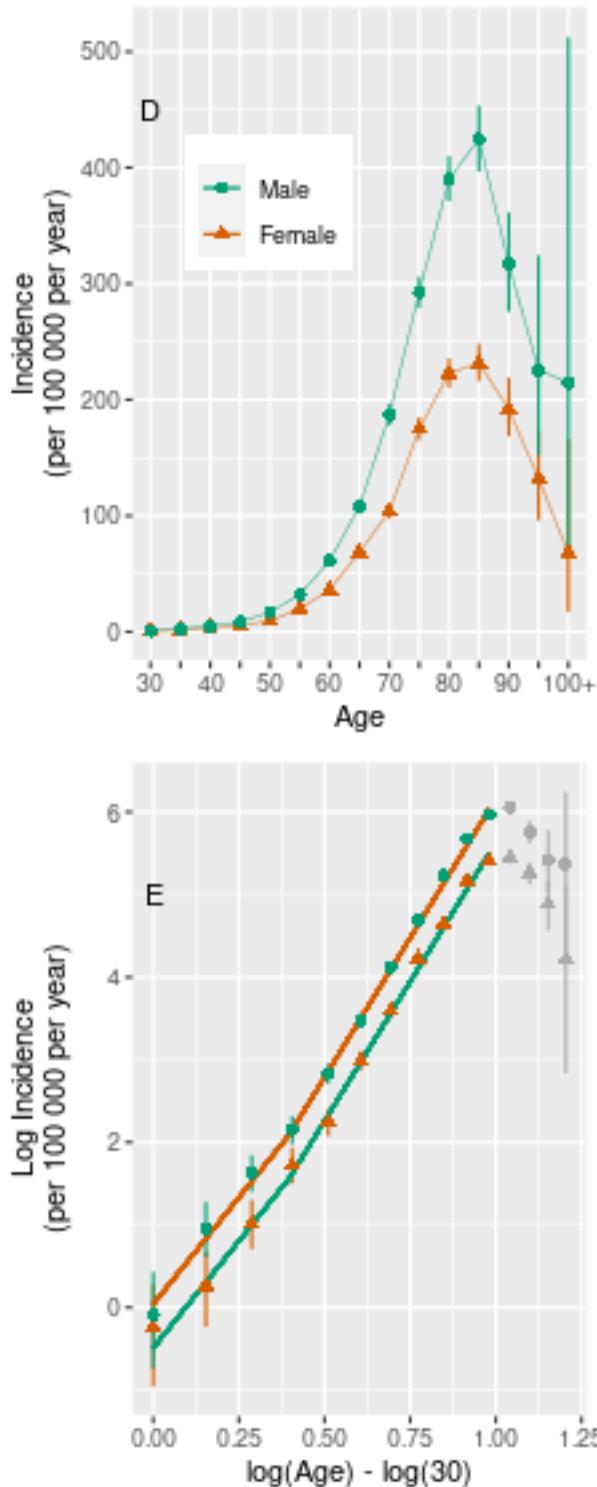
```
ggsave("plots/multistep-figure-2.pdf", width=4, height=12)
ggsave("plots/multistep-figure-2.png", width=4, height=12)
```

Plot: Figure 3

```
figure_3 = cowplot::plot_grid(fig2_age_sex_specific_incidence,
                             fig2_log_sex,
                             align = 'v',
```

```
axis = 'bt', ncol = 1)

plot(figure_3)
```



```
ggsave("plots/multistep-figure-3.pdf",width=4,height=6)
ggsave("plots/multistep-figure-3.png",width=4,height=6)
```

## Plot: Armitage-Doll model

```
# Extract fitted values from model
values_ad = summary(fit_ad)$summary
alpha=values_ad[1,1]
k=values_ad[2,1]

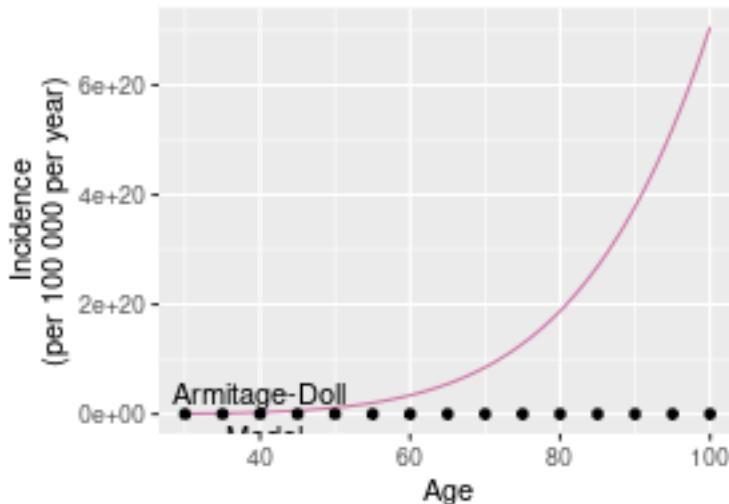
simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = (alpha*age)^(k-1)*100000)

fig4_ad <- ggplot(simdf,aes(x=age,y=predinc))+geom_line(colour="#CC79A7")+
  geom_point(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle))+  

  geom_errorbar(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle,ymin=lower,ymax=upper), width=0)+  

  annotate("text",x=40,y=1000,label="Armitage-Doll\n Model")+
  xlab("Age")+ylab("Incidence\n(per 100 000 per year)")

plot(fig4_ad)
```



```
ggsave("plots/multistep-ad-fit.pdf",width=5,height=3)
```

## Plot: Armitage-Doll model by Sex

```
# Extract fitted values from model
values_ad_sex = summary(fit_ad_sex)$summary

alpha=values_ad_sex[1,1]/1e3
k=values_ad_sex[2,1]

alpha_sex=values_ad_sex[3,1]/1e3
k_sex=values_ad_sex[4,1]

alpha1 <- alpha + alpha_sex
k1 <- k + k_sex

simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = (alpha*age)^(k-1)*100000)

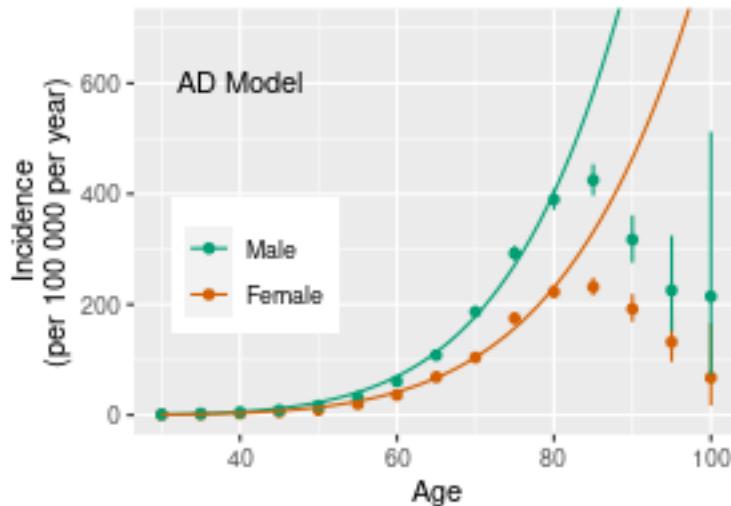
simdf_female <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = (alpha1*age)^(k1-1)*100000)
```

```

fig4_ad_sex <- sim_df_ms_sex_full %>%
  mutate(sex = factor(sex, levels=c("Male", "Female"))) %>%
  ggplot(aes(x=agecut5_numeric, y=middle, colour=sex)) +
  geom_point() + geom_errorbar(aes(ymin=lower, ymax=upper), width=0) +
  scale_color_manual(values = colours_mf) +
  geom_line(data=simdf, aes(x=age, y=predinc), colour=colours_mf[1]) +
  geom_line(data=simdf_female, aes(x=age, y=predinc), colour=colours_mf[2]) +
  annotate("text", x=40, y=600, label="AD Model") +
  xlab("Age") + ylab("Incidence\n(per 100 000 per year)") +
  coord_cartesian(ylim=c(0, 700)) +
  theme(legend.position = c(0.2, 0.4),
        legend.title = element_blank())

plot(fig4_ad_sex)

```



```

ggsave("plots/multistep-ad-sex.pdf", width=5, height=3)
ggsave("plots/multistep-ad-sex.png", width=5, height=3)

```

## Plot: Beta model

```

# Fit for beta (from nls)
alpha = 6.084e-03
beta = 9.863e-03
k = 7.202

# Extract fitted values from model
values_beta = summary(fit_beta)$summary

alpha=values_beta[1,1]
beta=values_beta[2,1]
k=values_beta[3,1]

#values_beta = summary(fit_beta)$c_summary
#chain = 2

#alpha=values_beta[1,1,chain]

```

```

#beta=values_beta[2,1,chain]
#k=values_beta[3,1,chain]

simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = (alpha*age)^(k-1)*(1-beta*age)*100000)

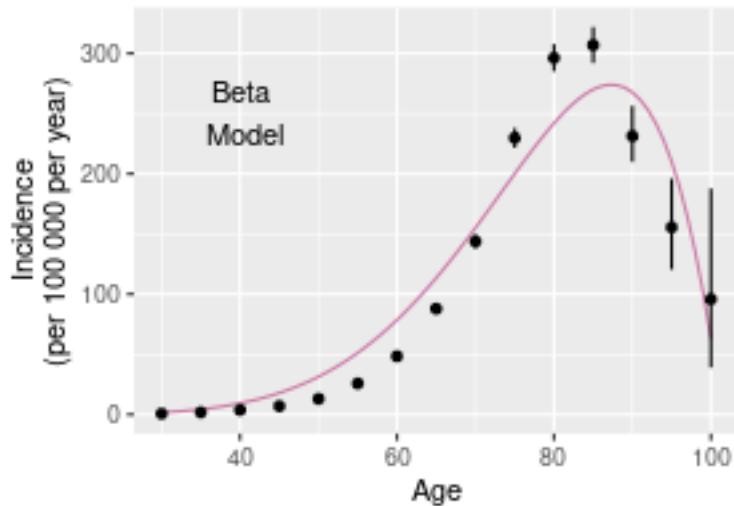
fig4_beta <- ggplot(simdf,aes(x=age,y=predinc))+geom_line(colour="#CC79A7")+
  geom_point(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle))+  

  geom_errorbar(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle,ymin=lower,ymax=upper), width=0)+  

  annotate("text",x=40,y=250,label="Beta\n Model")+
  xlab("Age")+ylab("Incidence\n(per 100 000 per year)")

plot(fig4_beta)

```



```
ggsave("plots/multistep-beta-fit.pdf",width=5,height=3)
```

### Plot: Beta model by Sex

```

# Extract fitted values from model
values_beta_sex = summary(fit_beta_sex)$summary

alpha=values_beta_sex[1,1]/1e3
beta=values_beta_sex[2,1]
k=values_beta_sex[3,1]

alpha_sex=values_beta_sex[4,1]/1e3
beta_sex=values_beta_sex[5,1]
k_sex=values_beta_sex[6,1]

alpha1 <- alpha + alpha_sex
k1 <- k + k_sex
beta1 <- beta + beta_sex

simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = (alpha*age)^(k-1)*(1-beta*age)*100000)

simdf_female <- data.frame(age=seq(30,100)) %>%

```

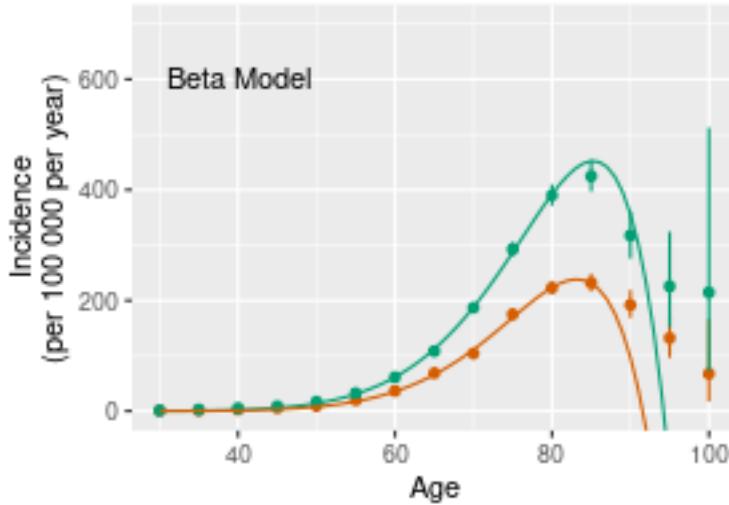
```

mutate(predinc = (alpha1*age)^(k1-1)*(1-beta1*age)*100000)

fig4_beta_sex <- sim_df_ms_sex_full %>%
  mutate(sex = factor(sex, levels=c("Male","Female"))) %>%
  ggplot(aes(x=agecut5_numeric,y=middle,colour=sex))+
  geom_point() + geom_errorbar(aes(ymin=lower,ymax=upper), width=0) +
  scale_color_manual(values = colours_mf) +
  geom_line(data=simdf, aes(x=age,y=predinc), colour=colours_mf[1]) +
  geom_line(data=simdf_female, aes(x=age,y=predinc), colour=colours_mf[2]) +
  annotate("text",x=40,y=600,label="Beta Model") +
  xlab("Age") + ylab("Incidence\n(per 100 000 per year)") +
  coord_cartesian(ylim=c(0, 700)) +
  theme(legend.position = "none")

plot(fig4_beta_sex)

```



```

ggsave("plots/multistep-beta-sex.pdf",width=5,height=3)
ggsave("plots/multistep-beta-sex.png",width=5,height=3)

```

## Plot: Susceptibility

```

# nls
#alpha = 1.2e-15
#C = 8.024e-02
#k = 8.2

# stan
alpha = 2.03e-15
C = 8.06e-02
k = 8.11

# Extract fitted values from model
values_suscept = summary(fit_suscept)$summary

alpha=values_suscept[1,1]/1e15
C=values_suscept[2,1]/100
k=values_suscept[3,1]

```

```

simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = alpha*age^(k-1)*100000/(1+((1-C)/C)*exp(alpha/k*(age^(k-1)))))

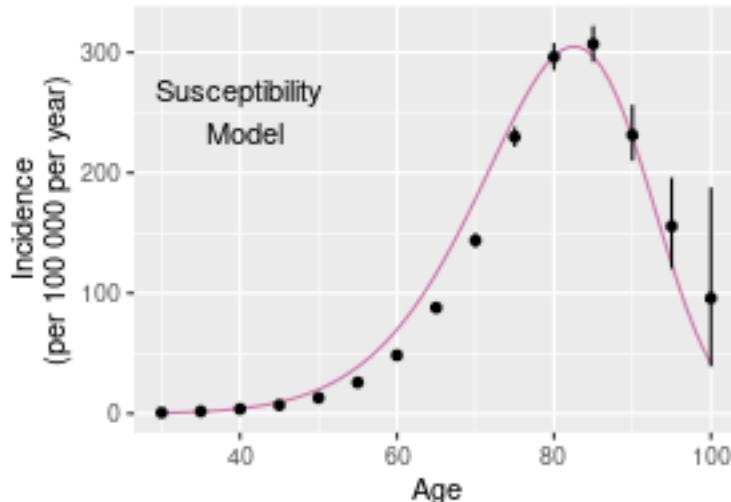
fig4_susceptibility <- ggplot(simdf,aes(x=age,y=predinc))+geom_line(colour="#CC79A7")+
  geom_point(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle))+  

  geom_errorbar(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle,ymin=lower,ymax=upper), width=0)+  

  annotate("text",x=40,y=250,label="Susceptibility\n Model")+
  xlab("Age")+ylab("Incidence\n(per 100 000 per year)")

plot(fig4_susceptibility)

```



```
ggsave("plots/multistep-susceptibiliy.pdf",width=5,height=3)
```

## Plot: Susceptibility Fixed

```

# Extract fitted values from model
values_suscept = summary(fit_suscept_fixed)$summary

alpha=values_suscept[1,1]/1e15
k=values_suscept[2,1]
C=data_list_suscept_fixed$C/100

simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = alpha*age^(k-1)*100000/(1+((1-C)/C)*exp(alpha/k*(age^(k-1)))))

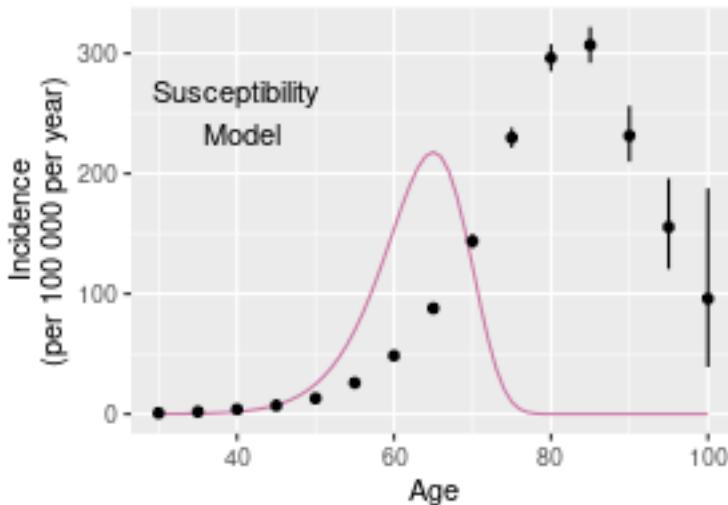
fig4_susceptility_fixed <- ggplot(simdf,aes(x=age,y=predinc))+geom_line(colour="#CC79A7")+
  geom_point(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle))+  

  geom_errorbar(data=sim_df_ms_full,aes(x=agecut5_numeric,y=middle,ymin=lower,ymax=upper), width=0)+  

  annotate("text",x=40,y=250,label="Susceptibility\n Model")+
  xlab("Age")+ylab("Incidence\n(per 100 000 per year)")

plot(fig4_susceptility_fixed)

```



```
ggsave("plots/multistep-susceptibiliy-fixed-C-03.pdf",width=5,height=3)
```

Plot: Susceptibility model by Sex

```
# nls
#alpha = 1.2e-15
#C = 8.024e-02
#k = 8.2

# stan
alpha = 2.03e-15
C = 8.06e-02
k = 8.11

# Extract fitted values from model
values_suscept_sex = summary(fit_suscept_sex)$summary

alpha=values_suscept_sex[1,1]/1e15
C=values_suscept_sex[2,1]/100
k=values_suscept_sex[3,1]

alpha_sex=values_suscept_sex[4,1]/1e15
C_sex=values_suscept_sex[5,1]/100
k_sex=values_suscept_sex[6,1]

alpha1 <- (alpha + alpha_sex)*0.8
C1 <- C + C_sex
k1 <- k + k_sex

simdf <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = alpha*age^(k-1)*100000/(1+((1-C)/C)*exp(alpha/k*(age^(k-1)))))

simdf_female <- data.frame(age=seq(30,100)) %>%
  mutate(predinc = alpha1*age^(k1-1)*100000/(1+((1-C1)/C1)*exp(alpha1/k1*(age^(k1-1)))))

fig4_susceptibility_sex <- sim_df_ms_sex_full %>%
  mutate(sex = factor(sex,levels=c("Male","Female")))) %>%
```

```

ggplot(aes(x=agecut5_numeric,y=middle,colour=sex))+  

  geom_point() + geom_errorbar(aes(ymin=lower,ymax=upper), width=0) +  

  scale_color_manual(values = colours_mf) +  

  geom_line(data=simdf,aes(x=age,y=predinc),colour=colours_mf[1]) +  

  geom_line(data=simdf_female,aes(x=age,y=predinc),colour=colours_mf[2]) +  

  annotate("text",x=40,y=600,label="Susceptibility\n Model") +  

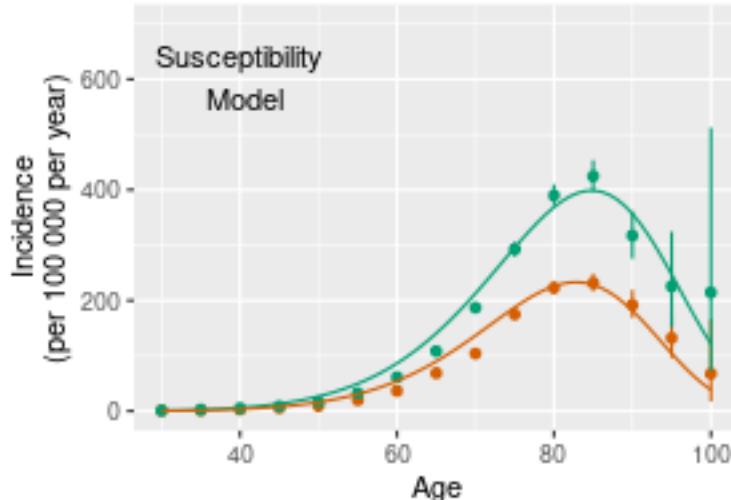
  xlab("Age") + ylab("Incidence\n(per 100 000 per year)") +  

  coord_cartesian(ylim=c(0, 700)) +  

  theme(legend.position = "none")

plot(fig4_susceptibility_sex)

```



```

ggsave("plots/multistep-susceptibiliy-sex.pdf",width=5,height=3)  

ggsave("plots/multistep-susceptibiliy-sex.png",width=5,height=3)

```

Plot: Figure 4

```

figure_4 = cowplot::plot_grid(fig4_ad_sex,  

                             fig4_beta_sex,  

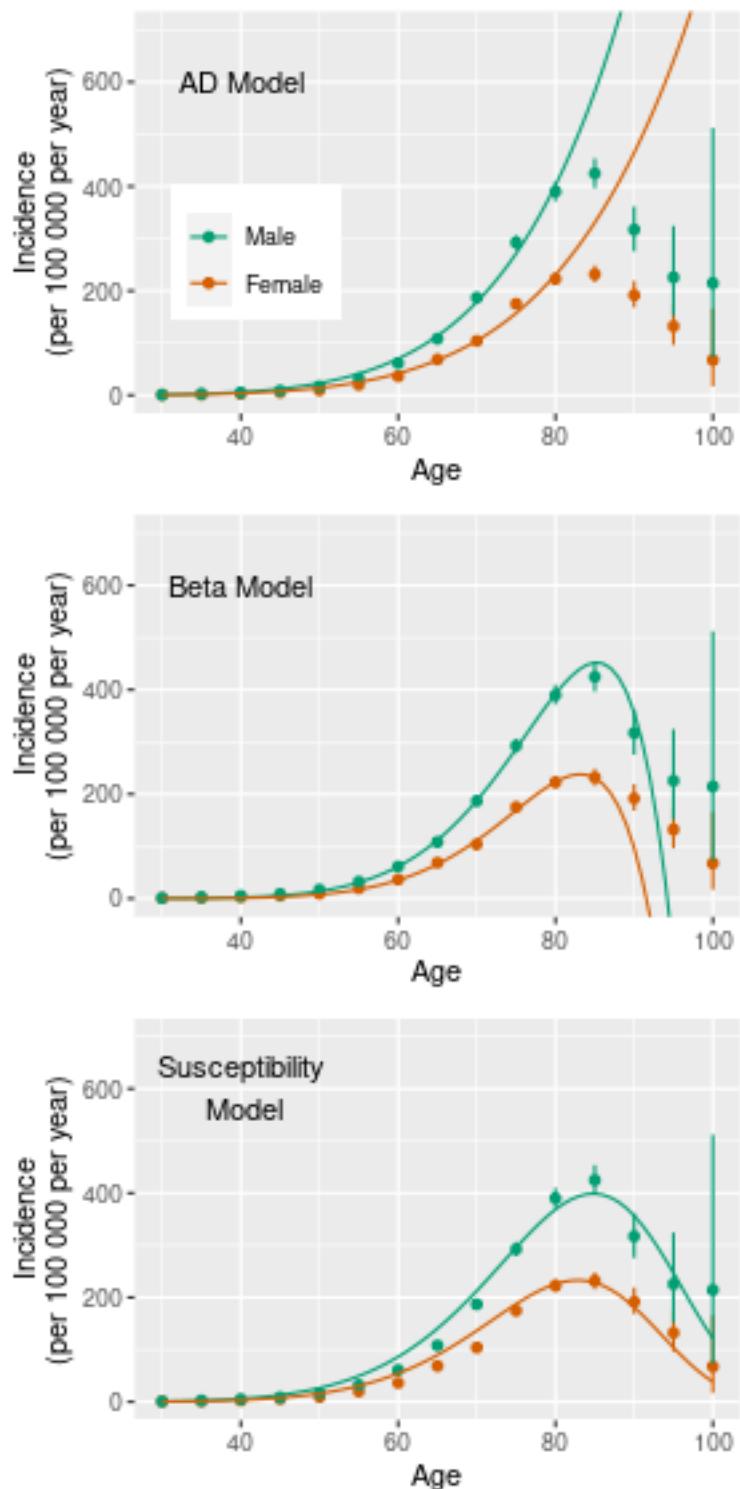
                             fig4_susceptibility_sex,  

                             align = 'v',  

                             axis = 'bt', ncol = 1)

plot(figure_4)

```



```
ggsave("plots/multistep-figure-4.pdf", width=4, height=8)
ggsave("plots/multistep-figure-4.png", width=4, height=8)
```