Fitting the Gamma model to Blood pressure data

David Steinsaltz

Distribution of BP measurements

Extract intervals for the digits

Suppose the fractions of digits 0,2,4,6,8 are b_0, b_2, b_4, b_6, b_8. Letting $B_0 = 0$ and $B_k = 10\sum_{j=0}^{k-1} b_{2j}$ for $k = 1, \ldots, 5$, we want to choose a positive a and place breaks at $-a + B_k$, so that measurements between $-a + B_k$ and $-a + B_{k+1}$ modulo 10 are assigned the final digit $2k$, for $k = 0, \ldots, 4$. We choose a to minimise the total distance of the intervals from the rounded value:

$$\sum_{k=0}^{4} \int_{-a+B_k}^{-a+B_{k+1}} |x - 2k| \mathrm{d}x = \frac{1}{2} \sum_{k=0}^{4} (-a + B_k - 2k)^2 + (-a + B_{k+1} - 2k)^2 ,$$

as long as $2k$ is in the appropriate interval. This is minimized at

$$a = \frac{1}{5}(B_1 + B_2 + B_3 + B_4 - 15) = \sum_{j=0}^{3} (8 - 2j) b_{2j} - 3.$$

Load data

```
digitbreaks <- function(bp){
  digit_table <- unname(table(unlist(bp)%%10))
  digits=digit_table/sum(digit_table)
  a <- sum(seq(8,0,by = -2)*digits) - 3 # Find the starting point for the first interval that minimises
  list(BP=bp,digit.breaks=c(0,cumsum(digits[-5]))*10-a)
}

# Combine the measures into a list, so they can be processed uniformly
#   First level, Systolic or Diastolic
#   Second level, Home or Clinic
#  Third level, BP and breaks
allBP <- list(Systolic=list(Clinic=digitbreaks(sys),Home=digitbreaks(sysH)),Diastolic=list(Clinic=digitl

  # Shift by the mean of the cumulative sums; apply transposes, so we transpose back

### Impute fractional parts.

impute <- function(d, breaks=c(1,3,5,7,9)){
  # Intervals defined relative to the centre
  right_breaks <- breaks-seq(0,8,by=2)
    left_breaks <- c(breaks[5]-10,breaks[1:4]-seq(2,8,by=2))
  if (!all(d%%2==0)){stop('Not all even digits')}
  else{
    if (is.null(dim(d))){ # Not an array
      d2 <- (d%%10)/2+1
      runif(length(d2))*(right_breaks[d2]-left_breaks[d2])+left_breaks[d2]+d
```

```
    }
    else{
      apply(d,2,function(dd) impute(dd,breaks))
    }
  }
}

# Input is a list BP=matrix of measures, digit.breaks=break points
imputeBP <- function(bp_with_breaks){
  breaks <- bp_with_breaks$digit.breaks
  d <- bp_with_breaks$BP
  list(BP=impute(d,breaks),digit.breaks=breaks)
}

allBP_imp <- lapply(allBP, function(BPtype) lapply(BPtype,function(BPplace) imputeBP(BPplace)))
```

Fit the BP distribution parameters

We suppose that each individual has BP measures $\tilde{y}_{ij}^l$ for $i = 1, \ldots, n$, $j = 1, \ldots, k$ (default $k = 3$), and $l = 1, 2$, which are rounded versions of

$$y_{ij}^l \sim \mathcal{N}\big(\mu_i^l, (\tau_i^l)^{-1}\big),$$

where

$$\begin{aligned}
\mu_i^1 &= (M_i + \Delta_i)/2, \\
\mu_i^2 &= (M_i - \Delta_i)/2, \\
M_i &\sim \mathcal{N}\big(m_M, \sigma_M^2\big) \text{ and } \Delta_i \sim \mathcal{N}\big(m_\Delta, \sigma_\Delta^2\big) \text{ independent}, \\
\tau_i^l &\sim \text{Gamma}(\alpha^l, \alpha^l/\theta^l).
\end{aligned}$$

(Note that α^l is the usual shape parameter, while θ^l is the expectation.)

We wish to estimate the eight parameters

$$(m_M, m_\Delta, \sigma_M^2, \sigma_\Delta^2, \alpha^1, \theta^1, \alpha^2, \theta^2)$$

We begin by assuming y_{ij}^l observed directly. We estimate by maximising the partial likelihood on the observations

$$\begin{aligned}
\bar{y}_{i+} &:= \frac{1}{2k} \sum_{j=1}^{k} y_{ij}^1 + y_{ij}^2, \\
\bar{y}_{i-} &:= \frac{1}{2k} \sum_{j=1}^{k} y_{ij}^1 - y_{ij}^2, \\
s_i^l &:= \frac{1}{k-1} \sum_{j=1}^{k} \Big(y_{ij}^l - \frac{1}{k} \sum_{j=1}^{k} y_{ij}^l \Big)^2.
\end{aligned}$$

Note that

$$(k-1) s_i^l \tau_i^l = \sum_{j=1}^{k} \Big(z_{ij}^l - \frac{1}{k} \sum_{j=1}^{k} z_{ij}^l \Big)^2.$$

where z_{ij}^l are i.i.d. standard normal is independent of τ_i^l, thus has a chi-squared distribution with $k-1$ degrees of freedom — hence $\frac{k-1}{2} \cdot s_i^l \tau_i^l$ is gamma distributed with parameters $(\frac{k-1}{2}, 1)$. Since $\frac{\alpha}{\theta} \tau_i^l$ is independent of

$s_i^l \tau_i^l$, with Gamma$(\alpha, 1)$ distribution, we see that $\frac{\theta(k-1)}{2\alpha} s_i^l$ is the ratio of two independent gamma random variables, hence has beta-prime distribution with parameters $\left(\frac{k-1}{2}, \alpha\right)$, so log partial likelihood

$$\ell_{\text{Beta}}(\alpha, \theta; s_\cdot^l) = n\alpha \log \frac{\alpha}{\theta} + n \log \Gamma\left(\alpha + \frac{k-1}{2}\right) - n \log \Gamma(\alpha) + \frac{k-1}{2} \sum_{i=1}^{n} \log s_i^l - \left(\alpha + \frac{k-1}{2}\right) \sum_{i=1}^{n} \log\left(s_i^l + \frac{\alpha}{\theta}\right).$$

The partial Fisher Information has entries

$$\begin{aligned}
-\frac{\partial^2 \ell}{\partial \alpha^2} &= n\psi_1(\alpha) - n\psi_1\left(\alpha + \frac{k-1}{2}\right) - \frac{n}{\alpha} + \sum_{i=1}^{n} \frac{2\theta s_i^l + \alpha - (k-1)/2}{(\theta s_i^l + \alpha)^2} \\
-\frac{\partial^2 \ell}{\partial \theta^2} &= -\frac{n\alpha}{\theta^2} + \frac{\alpha}{\theta^2}\left(\alpha + \frac{k-1}{2}\right) \sum_{i=1}^{n} \frac{2\theta s_i^l + \alpha}{(\theta s_i^l + \alpha)^2} \\
-\frac{\partial^2 \ell}{\partial \theta \partial \alpha} &= \frac{n}{\theta} - \frac{1}{\theta} \sum_{i=1}^{n} \frac{\alpha^2 + 2\alpha\theta s_i^l + \frac{k-1}{2}\theta s_i^l}{(\theta s_i^l + \alpha)^2}.
\end{aligned}$$

where ψ_1 is the trigamma function.

```
  beta_prime_LL=function(alpha,theta,s,k=3){
    k1= (k-1)/2
    n<- length(s)
    alpha * n * log( alpha / theta ) - n * lbeta(k1 , alpha )+ (k1-1)* sum(log(s)) -
      (alpha + k1 )*sum(log(s+alpha/theta))
  }

beta_prime_gradient= function(alpha,theta,s,k=3){
  n<- length(s)
  k1 <- (k-1)/2
  d_a <- -n*log(alpha/theta)  - n*digamma(alpha) + n*digamma(alpha+ k1 ) - (alpha+ k1) * sum(log(s * th
  d_t <- -n*alpha/theta + (alpha+ k1 ) * alpha/theta * sum( 1 /(theta*s+ alpha))
  c(d_a,d_t)
}

# Fisher Information
beta_prime_FI=function(alpha,theta,s,k=3){
    n<- length(s)
    k1 <- (k-1)/2
    d_aa <- -n/alpha + sum(( 2*theta*s+alpha - k1 )/(theta*s+alpha)^2) -
      n*trigamma(alpha+k1 ) + n * trigamma(alpha)
    d_tt <- -n* alpha /theta^2 + ( alpha + k1 )*alpha/theta^2 * sum((alpha+2*theta*s)/
                                          (theta*s+alpha)^2)
    d_at <- sum((s^2*theta - s*k1) / (alpha+ theta*s)^2)
    matrix(c(d_aa,d_at,d_at,d_tt), 2, 2)
}

alphatheta <- function(BP){
  k <- dim(BP)[2]
  s <- (k-1)/2*apply(BP,1,var)
  LL <- function(gamma_params){
    -beta_prime_LL(gamma_params[1],gamma_params[2],s,k)
  }
  LL_grad <- function(gamma_params){
    -beta_prime_gradient(gamma_params[1],gamma_params[2],s,k)
```

```
  }
  # Using optim because constrOptim doesn't work
  #fit <- suppressWarnings( constrOptim(c(1,1),f = LL, grad = LL_grad, ui = diag(1,nrow = 2,ncol = 2),
  fit <- suppressWarnings( optim(par = c(1,1),fn = LL, gr = LL_grad ) )
  fisher_info <- beta_prime_FI(fit$par[1],fit$par[2],s,k)
  list( alpha = fit$par[1], theta = fit$par[2], variance = solve(fisher_info), LogLikelihood = -fit$valu
}
```

Let $(\hat{\alpha}^l, \hat{\beta}^l)$ be the maximum partial likelihood estimators. Conditioned on (τ_i^l) we have

$$\bar{y}_{i+} \sim \mathcal{N}\left(m_M, \sigma_M^2 + \frac{1}{4k}\left(\frac{1}{\tau_i^1} + \frac{1}{\tau_i^2}\right)\right),$$
$$\bar{y}_{i-} \sim \mathcal{N}\left(m_\Delta, \sigma_\Delta^2 + \frac{1}{4k}\left(\frac{1}{\tau_i^1} + \frac{1}{\tau_i^2}\right)\right).$$

We would then have MLEs

$$\hat{m}_M = \frac{1}{n}\sum_{i=1}^n \bar{y}_{i+},$$
$$\hat{m}_\Delta = \frac{1}{n}\sum_{i=1}^n \bar{y}_{i-},$$

which are approximately normally distributed, with means m_M and m_Δ respectively, and conditional on τ_i^l standard errors

$$\frac{\sigma_M^2}{n} + \frac{1}{4kn^2}\sum_{i=1}^n (\tau_i^1)^{-1} + (\tau_i^2)^{-1} \quad \text{and} \quad \frac{\sigma_\Delta^2}{n} + \frac{1}{4kn^2}\sum_{i=1}^n (\tau_i^1)^{-1} + (\tau_i^2)^{-1},$$

which we may approximate — with error on the order of $n^{-3/2}$ — replaceing the mean of $(\tau_i^l)^{-1}$ by its expected value $\beta^l/(\alpha^l - 1)$ to obtain

$$\mathrm{Var}(\hat{m}_M) \approx \frac{\sigma_M^2}{n} + \frac{1}{4kn}\left(\frac{\beta^1}{\alpha^1 - 1} + \frac{\beta^2}{\alpha^2 - 1}\right)$$
$$\mathrm{Var}(\hat{m}_\Delta) \approx \frac{\sigma_\Delta^2}{n} + \frac{1}{4kn}\left(\frac{\beta^1}{\alpha^1 - 1} + \frac{\beta^2}{\alpha^2 - 1}\right)$$

Finally, conditioned on the τ_i^l we have that the random variables $\bar{y}_{i+}$ are normal with variance

$$\sigma_M^2 + \frac{1}{4k}\left((\tau_i^1)^{-1} + (\tau_i^1)^{-1}\right),$$

so the unconditional variance is the expected value, or

$$\sigma_M^2 + \frac{1}{4k}\left(\frac{\beta^1}{\alpha^1 - 1} + \frac{\beta^2}{\alpha^2 - 1}\right).$$

This yields the estimators

$$\hat{\sigma}_M^2 = \frac{1}{n-1}\sum_{i=1}^n \left(\bar{y}_{i+} - n^{-1}\sum_{i=1}^n y_{i+}\right)^2 - \frac{1}{4k}\left(\frac{\hat{\beta}^1}{\hat{\alpha}^1 - 1} + \frac{\hat{\beta}^2}{\hat{\alpha}^2 - 1}\right),$$
$$\hat{\sigma}_\Delta^2 = \frac{1}{n-1}\sum_{i=1}^n \left(\bar{y}_{i-} - n^{-1}\sum_{i=1}^n y_{i-}\right)^2 - \frac{1}{4k}\left(\frac{\hat{\beta}^1}{\hat{\alpha}^1 - 1} + \frac{\hat{\beta}^2}{\hat{\alpha}^2 - 1}\right).$$

Using the delta method, and the fact that we see that the variance of $\hat{\beta}/(\hat{\alpha}-1)$ is approximately

$$\frac{\sigma_\beta^2}{(\hat{\alpha}-1)^2}+\frac{\hat{\beta}^2\sigma_\alpha^2}{(\hat{\alpha}-1)^4},$$

where σ_α and σ_β are the standard errors for $\hat{\alpha}$ and $\hat{\beta}$ respectively, so the standard errors for $\hat{\sigma}_M^2$ and $\hat{\sigma}_\Delta^2$ are approximately

$$\text{SE}\left(\hat{\sigma}_M^2\right)\approx\frac{1}{2k}\Big(\frac{8k^2\hat{\sigma}_M^2}{n}+\frac{\sigma_\beta^2}{(\hat{\alpha}^1-1)^2}+\frac{(\hat{\beta}^1)^2\sigma_\alpha^2}{(\hat{\alpha}^1-1)^4}+\frac{\sigma_\beta^2}{(\hat{\alpha}^2-1)^2}+\frac{(\hat{\beta}^2)^2\sigma_\alpha^2}{(\hat{\alpha}^2-1)^4}\Big)^{1/2},$$
$$\text{SE}\left(\hat{\sigma}_\Delta^2\right)\approx\frac{1}{2k}\Big(\frac{8k^2\hat{\sigma}_\Delta^2}{n}+\frac{\sigma_\beta^2}{(\hat{\alpha}^1-1)^2}+\frac{(\hat{\beta}^1)^2\sigma_\alpha^2}{(\hat{\alpha}^1-1)^4}+\frac{\sigma_\beta^2}{(\hat{\alpha}^2-1)^2}+\frac{(\hat{\beta}^2)^2\sigma_\alpha^2}{(\hat{\alpha}^2-1)^4}\Big)^{1/2}$$

Test the effect of imputation

```
number_imp <- 100 # This is the number of different imputations we'll simulate
alphas <- rep(0,number_imp)
thetas <- rep(0,number_imp)
d <- allBP$Systolic$Clinic
for (i in seq_len(number_imp)){
  d_imp <- imputeBP(d)$BP
  ab <- alphatheta(d_imp)
  alphas[i] <- ab$alpha
  thetas[i] <- ab$theta
}
```

Test whether parameters are being fit correctly

Simulate bootstrap data sets. Find the average parameter estimate, and compare to the "true" parameters. Also compare the average estimated SE to the "true" SE (which is the SD of the estimates). Systolic Parameter estimates Sim.average True Error m_M 123.0000 123.0000 -8.04e-05 m_Delta 1.3600 1.3400 8.29e-03 sigma2_M 377.0000 377.0000 1.67e-03 sigma2_Delta 46.4000 46.4000 2.59e-04 alpha_C 2.5600 2.5600 9.61e-04 theta_C 0.0753 0.0753 6.95e-04 alpha_H 2.1800 2.1700 3.09e-03 theta_H 0.1490 0.1490 4.17e-04

Systolic parameter SE Sim.average True SE Error m_M 0.161073498 0.146364417 0.10049629 m_Delta 0.057969806 0.056504033 0.02594103 sigma2_M 1.823935415 4.436930998 -0.58891959 sigma2_Delta 1.304045887 0.541363162 1.40881903 alpha_C 0.075236746 0.081118903 -0.07251279 theta_C 0.001038027 0.001130308 -0.08164248 alpha_H 0.057304615 0.058875556 -0.02668239 theta_H 0.002101338 0.002021784 0.03934860 Diastolic Parameter estimates Sim.average True Error m_M 72.300 72.300 -0.000149 m_Delta 1.090 1.100 -0.001840 sigma2_M 104.000 104.000 0.000177 sigma2_Delta 21.900 21.900 -0.001140 alpha_C 2.800 2.790 0.003810 theta_C 0.109 0.109 -0.002080 alpha_H 2.340 2.340 0.000358 theta_H 0.195 0.195 0.000527

Diastolic parameter SE Sim.average True SE Error m_M 0.085064446 0.085958370 -0.010399501 m_Delta 0.040351804 0.042001563 -0.039278529 sigma2_M 1.090234531 1.261955955 -0.136075608 sigma2_Delta 0.885905177 0.289383004 2.061358697 alpha_C 0.088028909 0.085441769 0.030279573 theta_C 0.001476194 0.001468476 0.005255872 alpha_H 0.064903765 0.074687367 -0.130994073 theta_H 0.002722848 0.002934787 -0.072216250

Multiple imputation for the real data

```
number_impute <- 10
all_variables_est <- c('m_M','m_Delta','sigma2_M','sigma2_Delta','alpha_C','theta_C','alpha_H','theta_H
```

```
all_variables_SE <- c( setNames(vapply(all_variables_est,function(N) paste0(N,'.SE'),'x'),NULL))
all_variables_cov <- c('Covar_C','Covar_H')
all_variables <- c(all_variables_est, all_variables_SE, all_variables_cov)
impute_results <- lapply(allBP, function(BPtype)
  setNames(data.frame( array(0,dim = c(number_impute,length(all_variables))) ), all_variables))

suppressWarnings(remove('Delta','Gamma'))

## Make containers for the results, one each for systolic and diastolic
for(i in seq_len(number_impute)){
    allBP_imp <- lapply(allBP, function(BPtype) lapply(BPtype,function(BPplace) imputeBP(BPplace)))
      # quantifying over allBP gives Systolic vs diastolic
      # Next level gives Home vs Clinic
      # Next level has BPs and digit breaks
    for (BPtype in BP_type_names){
      spe <- all_BP_parameters(allBP_imp[[BPtype]]) # simulated parameter estimates
      attach(spe$Gamma)
      attach(spe$Normal)
      gamma_SE_C <- sqrt(diag(Clinic$variance))
      gamma_SE_H <- sqrt(diag(Home$variance))

  impute_results[[BPtype]][i,all_variables_est] <- c(M$m, Delta$m, M$sigma2, Delta$sigma2,
                                  Clinic$alpha,Clinic$theta, Home$alpha,Home$theta)
  impute_results[[BPtype]][i,all_variables_SE] <- c(M$m.std.error,Delta$m.std.error, M$sigma2.std.error
                                   gamma_SE_C,gamma_SE_H)
  impute_results[[BPtype]][i,all_variables_cov] <- c( Clinic$variance[1,2], Home$variance[1,2] )
    detach(spe$Gamma)
    detach(spe$Normal)
    }
}

total_M_I_results <- lapply(impute_results, function(result)
{
  M_I_estimates = apply(result[,all_variables_est],2,mean)
                M_I_std_error = apply(result[,all_variables_est], 2, sd)
                M_I_covariance = c('Clinic' = cov(result[,'alpha_C'],result[,'theta_C']), 'Home' = cov(
  c(list(estimates= M_I_estimates, std_errors = c(sqrt(apply(result[,all_variables_SE]^2, 2 , mean ) + I
})

total_M_I_results <- lapply(total_M_I_results, function(result) c(result, list(Correlation = setNames(c
                                                                                'Home' = resu
)
```

Now compute the combined variance

For a parameter like α we estimate the variance of $\hat{\alpha}$ by

$$\mathrm{Var}(\hat{\alpha}) = \mathbb{E}\big[\mathrm{Var}\left(\hat{\alpha} \mid I\right)\big] + \mathrm{Var}\left(\mathbb{E}\left[\hat{\alpha} \mid I\right]\right).$$

Here I represents the randomly imputed fractional part. We can estimate the first term by averaging the estimated variance (from Fisher Information) over all random imputations. We estimate the second term by the variance of the α estimates over imputations. Note that this is not quite right, since what we really want the variance of is $\alpha_0(I)$ — effectively, the "true'' parameter consistent with the imputation. This is a plug-in

estimate, as is the Fisher Information estimate of the variance.

Computing residuals

We define the deviance for an individual i with observations (Y_i) given the hyperparameters $h = (m_M, m_\Delta, \sigma_M^2, \sigma_\Delta^2, \alpha^H, \theta^H, \alpha^C, \theta^C)$

$$D = \sum_{i=1}^{n} \log \mathbb{P}\left\{ \mathbf{Y}_i \,|\, \text{hyperparameters} = h \right\}.$$

Since the $\mathbf{Y}_i$ are independent conditioned on h,

$$\begin{aligned} D &= \sum_{i=1}^{n} \log \mathbb{E}_h \left[\mathbb{P}\left\{ \mathbf{Y}_i \,|\, M_i, \Delta_i, \tau_i^C, \tau_i^H \right\} \right] \\ &\approx \sum_{i=1}^{n} \log \frac{1}{R} \sum_{r=1}^{R} \left[\mathbb{P}\left\{ \mathbf{Y}_i \,|\, M_{i,r}, \Delta_{i,r}, \tau_{i,r}^C, \tau_{i,r}^H \right\} \right] \frac{\pi_h(M_{i,r}, \Delta_{i,r}, \tau_{i,r}^C, \tau_{i,r}^H)}{q(M_{i,r}, \Delta_{i,r}, \tau_{i,r}^C, \tau_{i,r}^H \,|\, h,\, \mathbf{Y}_i)}, \end{aligned}$$

where $(M_{i,r}, \Delta_{i,r}, \tau_{i,r}^C, \tau_{i,r}^H)$ are independent samples from a distribution q that may depend on $\mathbf{Y}_i$ and h, and π_h is the true density of those individual parameters given hyperparameters h.

Check empirical variances

We simulate new variances from the inferred model, and compare it with QQ plots. The first thing we do is to compare the empirical variances (with imputed fractional parts) to the beta-prime distribution.

```
# Compare to beta-prime distribution
params <- convert_param1(BP_parameters)
dbetaprime <- function(x, A=1, B= BP_parameters$Systolic$Gamma$Clinic$alpha)
  {
    x^(A-1)*(1+x)^(-A-B)/ beta(A,B)
}

all_s2k <- NULL
all_s2k_sim <- NULL
all_s2k_sim_noimp <- NULL
all_SD <-NULL
all_HC <- NULL

oversample <- 1

bp_sim <- BP_simulation2(params$Normal,params$Gamma,
                         num_indiv= N, num_sim = oversample) # Get oversample sets of simulations;
    #We're going to take 1/oversample of them, to get a more accurate
#    estimate of the distribution, particularly at the upper end

for(which_bp in BP_type_names){
#    s2k <- apply(bp_sim[[which_bp]])
    for(i in seq_len(oversample)){
      allBP_imp <- lapply(allBP, function(BPtype) lapply(BPtype,function(BPplace) imputeBP(BPplace)))
    for ( where_bp in BP_place_names){

      s2k <- apply(allBP_imp[[which_bp]][[where_bp]]$BP,1,var) /
                          params$Gamma[[which_bp]]['beta',where_bp]
      s2k_sim <- apply(bp_sim[[i]][[which_bp]][[where_bp]]$BP,1,var) /
```

```
                          params$Gamma[[which_bp]]['beta',where_bp]
      all_s2k %<>% c(sort(s2k))
      all_s2k_sim %<>% c(sort(s2k_sim)[seq(from = ceiling(oversample/2), by = oversample, length.out = l
      all_SD %<>% c(rep(which_bp,length(s2k)))
      all_HC %<>% c(rep(where_bp,length(s2k)))
}
    ggplot(tibble(Variance = s2k[s2k<4]), aes(x=Variance)) +
      geom_histogram(alpha=0.2, position="identity", aes(y = ..density..), binwidth = .04) +
      labs(title =  paste(which_bp,where_bp, ': Histogram of Variances' )) +
      stat_function(fun = function(xx) dbetaprime(xx,A =1,
                           B = params$Gamma[[which_bp]]['alpha',where_bp]))
  }
}

for(which_bp in BP_type_names){
  for(where_bp in BP_place_names){
    all_s2k[all_SD == which_bp & all_HC == where_bp] <- sort(all_s2k[all_SD == which_bp & all_HC == whe
    all_s2k_sim[all_SD == which_bp & all_HC == where_bp] <- sort(all_s2k_sim[all_SD == which_bp & all_H
  }
}
# Need to sort the variances within simulation groups.
```

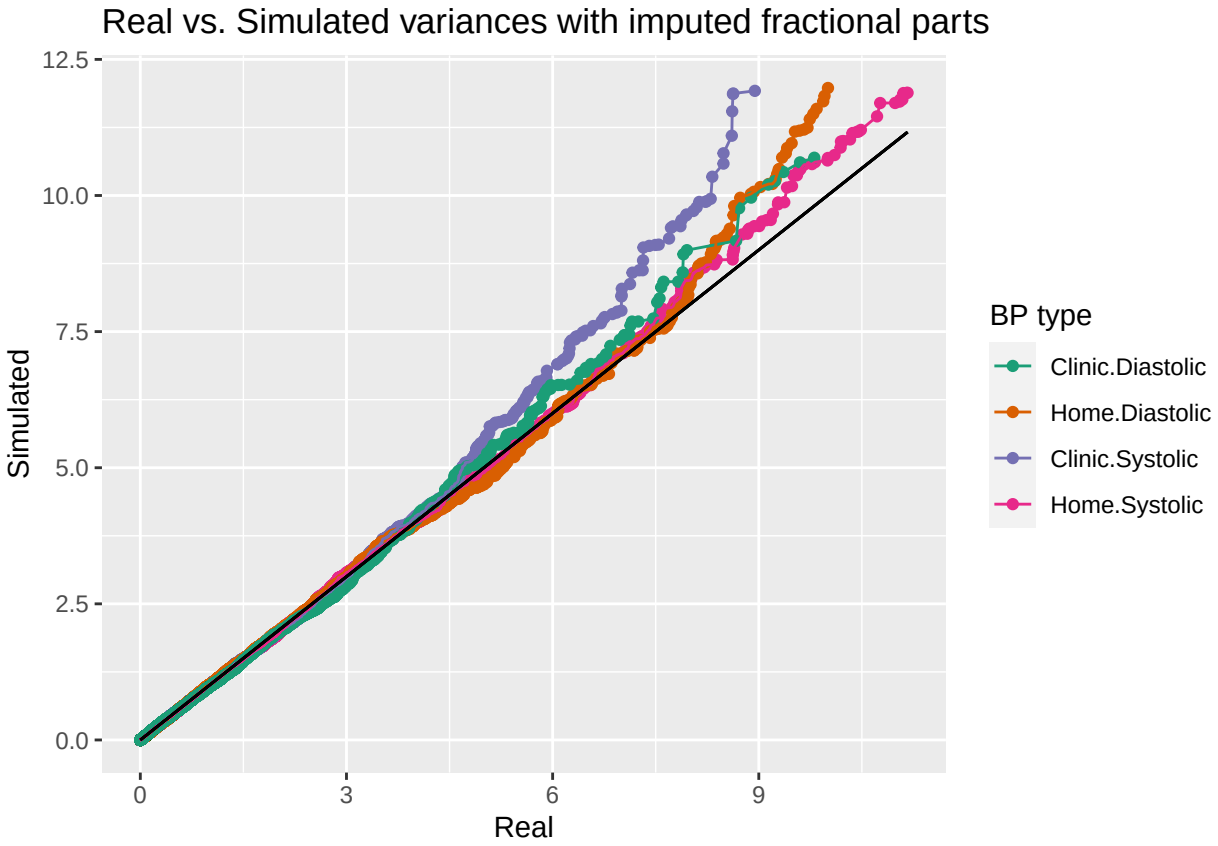
Now we compare the distribution of observed variances to the distribution of simulated variances.

```
all_s2k.tbl <- subset(tibble(Real = all_s2k, Simulated = all_s2k_sim,
                      HC = factor(all_HC),
                      DS = factor(all_SD)),
                      pmax(Real, Simulated) < 12 )

s2k.plot <- ggplot(all_s2k.tbl, aes(x=Real, y=Simulated, colour=interaction(HC,DS),
  group=interaction(HC, DS))) +
  geom_point() + geom_line() +  stat_function(fun = identity , color = 'black' ) + labs( colour = 'BP ty
  scale_color_brewer(palette="Dark2")

s2k.plot
```

```
## Warning: Multiple drawing groups in `geom_function()`. Did you use the correct
## `group`, `colour`, or `fill` aesthetics?
```



Estimating the mortality parameters. We divide the individuals up into three races $r_i \in \{\mathrm{B}, \mathrm{W}, \mathrm{M}\}$ and two sexes $s_i \in \{\mathrm{M}, \mathrm{F}\}$. For each of these we model the mortality rate at age t as $h_i(t) = B_{r_i s_i} \mathrm{e}^{\theta_{r_i s_i} t}$. So we need to estimate