Web Material
A guide to estimating the reference range from a meta-analysis using aggregate or individual participant data

Lianne Siegel, PhD, Mohammad Hassan Murad, MD, Richard D Riley, PhD,

Fateh Bazerbachi, MD, Zhen Wang, PhD, Haitao Chu, PhD, MD

Table of Contents

Web Appendix 1

Web Table 1. Methods for estimating the reference range.

1A. Frequentist Approach (Aggregate Data):

1) Estimate the pooled mean (μ_{RE}) and between-study variation (τ^2) using a frequentist random-effects model such as REML
2) $\hat{\sigma}^2 = \frac{\sum_{i=1}^{N}(n_i-1)s_i^2}{\sum_{i=1}^{N}(n_i-1)}$, where s_i^2 is the sample variance from study $i \in \{1, \dots, N\}$
3) Limits of the estimated reference range: $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N(\hat{\mu}_{RE}, \hat{\sigma}^2 + \hat{\tau}^2)$ distribution

1B. Frequentist Approach (Individual Participant Data):

1) Fit a frequentist random-effects model (linear mixed model) directly using the individual participant data
2) $\hat{\tau}^2$ = Estimated variance of the random effects
3) $\hat{\sigma}^2$ = Estimated residual variance
4) $\hat{\mu}_{RE}$ = Estimated pooled mean (fixed effect)
5) Limits of the estimated reference range: $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N(\hat{\mu}_{RE}, \hat{\sigma}^2 + \hat{\tau}^2)$ distribution

1C. Bayesian Approach (Aggregate Data):

1) $\bar{y}_i \sim N\left(\theta_i, \frac{\sigma^2}{n_i}\right), \theta_i \sim N(\mu_{RE}, \tau^2), (n_i - 1)s_i^2 \sim gamma\ (\frac{n_i-1}{2}, \frac{1}{2\sigma^2})$
2) Place N(0, 1000) prior on μ_{RE} and Uniform(0, 100) priors on σ and τ
3) Use MCMC sampler (such as JAGS, Stan, or WinBugs) to sample from posterior predictive distribution for a new individual: $y_{new} \sim N(\hat{\mu}_{RE}, \hat{\sigma}^2 + \hat{\tau}^2)$
4) Limits of the estimated reference range: $\alpha/2$ and 1 - $\alpha/2$ quantiles of y_{new} samples

1D. Bayesian Approach (Individual Participant Data):

1) $y_{ij} \sim N(\theta_i, \sigma^2), \theta_i \sim N(\mu_{RE}, \tau^2)$
2) Place N(0, 1000) prior on μ_{RE} and Uniform(0, 100) priors on σ and τ
3) Use MCMC sampler (such as JAGS, Stan, or WinBugs) to sample from posterior predictive distribution for a new individual: $y_{new} \sim N(\hat{\mu}_{RE}, \hat{\sigma}^2 + \hat{\tau}^2)$
4) Limits of the estimated reference range: $\alpha/2$ and 1 - $\alpha/2$ quantiles of y_{new} samples

1E. Empirical Approach (Aggregate Data):

1) Empirically estimate the pooled mean and total variance:
$$\hat{\mu}_{emp} = \frac{\sum_{i=1}^{N} n_i \bar{y}_i}{\sum_{i=1}^{N} n_i},\ \hat{\sigma}^2_{T,emp} = \frac{\sum_{i=1}^{N}(n_i-1)\, s_i^2}{\sum_{i=1}^{N}(n_i-1)} + \frac{\sum_{i=1}^{N}(n_i-1)\,(\bar{y}_i-\hat{\mu}_{emp})^2}{\sum_{i=1}^{N}(n_i-1)}$$
2) Limits of the estimated reference range: $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N\left(\hat{\mu}_{emp},\ \hat{\sigma}^2_{T,emp}\right)$ distribution

1F. Empirical Approach (Individual Participant Data):

1) Empirically estimate the pooled mean and total variance as the observed mean and variance of the pooled individual participant data:
$$\hat{\mu}_{emp} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i} y_{ij}}{\sum_{i=1}^{N} n_i},\ \hat{\sigma}^2_{T,emp} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n_i}(y_{ij}-\hat{\mu}_{emp})^2}{(\sum_{i=1}^{N} n_i)-1}$$
2) Limits of the estimated reference range: $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N\left(\hat{\mu}_{emp},\ \hat{\sigma}^2_{T,emp}\right)$ distribution

Web Table 2. Comparison of interpretations of intervals described in paper

Interval	Pooled Mean	95% Prediction Interval for a New Study	95% Reference Range (proposed)
Interpretation	**Frequentist 95% Confidence Interval (a, b):** "We are 95% confident that the mean across all studies is between a and b."	**Prediction interval (c,d):** "The **mean of a new study** (from the same overall target population) is expected to fall between c and d with 95% probability."	**Reference range (e,f):** "The **measurement of a new individual** is expected to fall between e and f with 95% probability."
	Bayesian 95% Credible Interval (a,b): "The true mean across all studies lies between a and b with 95% probability."		
Assumptions	Under a random-effects model: • Study means follow a normal distribution[a] • The means are exchangeable across studies • The studies included are representative of some superpopulation of interest	Under a random-effects model: • Study means follow a normal distribution • The means are exchangeable across studies • The studies included are representative of some superpopulation of interest	**Frequentist:** • Measurements within each study follow a normal distribution • Study means follow a normal distribution and are exchangeable • Constant within-study variance
			Bayesian: • Same as frequentist
			Empirical: • Measurements across all studies follow a normal distribution
Estimation	**Frequentist:** $\hat{\mu}_{RE} \pm t_{N-1;1-0.05/2}\, SE(\hat{\mu}_{RE})$ (where N = # of studies)	**Frequentist:** $\hat{\mu}_{RE} \pm t_{N-2;1-0.05/2}\sqrt{\widehat{Var}(\hat{\mu}_{RE}) + \hat{\tau}^2}$ [3]	**See Box 1**
	Bayesian: 2.5th and 97.5th quantiles of the posterior distribution of the pooled mean (μ_{RE})	**Bayesian:** 2.5th and 97.5th quantiles of the posterior predictive distribution of a new study: $N(\mu_{RE}, \tau^2)$, where μ_{RE} and τ^2 refer to their posterior distributions	

[a] The method proposed by DerSimonian and Laird[1] does not require the study means to be normally distributed but can underestimate the between-study variance, particularly when the number of studies is small [2]

Frequentist approach using a random-effects model

Aggregate data

The first method we proposed uses the results of a frequentist random-effects model, which assumes that the underlying study means in the meta-analysis follow a normal distribution with mean μ_{RE} and variance τ^2. We then add the additional assumptions that the data within each study are normally distributed and that the within-study variance (σ^2) is constant across studies. This implies that each of the individuals included in the meta-analysis are marginally distributed $N(\mu_{RE}, \sigma^2 + \tau^2)$. To estimate the reference range, first estimate the pooled mean across all studies (μ_{RE}) and the between-study variation (τ^2) using a frequentist random-effects model. We use the restricted maximum likelihood (REML) model implemented in the R package "meta" [4], but other methods and software could also be used. Next, use the pooled sample variance as an estimate of the common within-study variance (Box 1). Finally, the bounds of an (1-α)100% level reference range can be estimated by the $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N(\hat{\mu}_{RE}, \hat{\sigma}^2 + \hat{\tau}^2)$ distribution (Box 1A).

Individual participant data

If IPD are available, any of the methods based on aggregate data described in this guide can still be used by first aggregating the data by study to get the study means, standard deviations, and sample sizes. However, a frequentist random effects model (linear mixed model) can also be fit directly using the IPD without first aggregating. We use the R package "lme4," [5(p4)] but many other choices of software are available. Let $\hat{\tau}^2$ be the estimated variance of the random effects, $\hat{\sigma}^2$ be the estimated residual variance, and $\hat{\mu}_{RE}$ be the estimated pooled mean from the model.

Then, the bounds of an (1-α)100% level reference range can be estimated by the $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N(\hat{\mu}_{RE}, \hat{\sigma}^2 + \hat{\tau}^2)$ distribution (Box 1B).

Bayesian posterior predictive interval

Aggregate data

The second method we proposed uses the posterior predictive distribution of a new individual from a Bayesian random-effects model. This imposes the same distributional assumptions as with the frequentist approach. The sampling distributions of the study-means and standard deviations can be used to estimate the posterior distributions of μ_{RE}, σ^2, and τ^2 using Markov Chain Monte Carlo sampling (Box 1C). An (1-α)100% level reference range can then be estimated by the $\alpha/2$ and 1 - $\alpha/2$ quantiles of samples from a $N(\mu_{RE}, \sigma^2 + \tau^2)$ distribution, the posterior predictive distribution for a new individual. We place a N(0, 1000) prior on μ_{RE}, and Uniform(0,100) priors on σ and τ, as shown in Box 1C. The main difference between this and the frequentist methods is that the posterior predictive interval incorporates the uncertainty in the estimated parameters into the reference range, whereas the frequentist methods do not.

Individual participant data

However, if IPD are available, a Bayesian random effects model can also be fit directly on the individual observations, just as with the frequentist approach. Instead of using the sampling distributions for the study means and standard deviations, we can use the likelihood for an individual observation (Box 1D). We still place the same priors on each of the estimated parameters, and the resulting range has the same interpretation as the posterior predictive interval based on the aggregate data.

Empirical approach

Aggregate data

Finally, we proposed a simple empirical approach using aggregate data, which is similar to the method used by Conceição et al. [6] to estimate reference ranges for normal Subjective Postural Vertical (SPV) measurements. This does not make the same assumption about constant within-study variance, but still assumes the data are normally distributed. First, empirically estimate the pooled mean (μ_{emp}), weighting by the sample size in each study. This is equivalent to the mean estimate in the fixed effects model proposed by Laird and Mosteller [7] when weighting by sample size. Then, estimate the total variance both within and across studies (σ_T^2) (Box 1E). An (1-α)100% level reference range can then be estimated by the $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N(\hat{\mu}_{emp}, \hat{\sigma}_T)$ distribution.

Individual participant data

If IPD are available, one could equivalently pool the data across studies and estimate the pooled mean (μ_{emp}) as the mean of these individual measurements. The total variance within and across studies could similarly be estimated as the variance of these pooled samples (σ_T^2) (Box 1F). Then, an (1-α)100% level reference range can be estimated by the $\alpha/2$ and 1 - $\alpha/2$ quantiles of a $N(\hat{\mu}_{emp}, \hat{\sigma}_T)$ distribution.

Web Appendix 2

Lognormally distributed data

In some cases, such as when a measurement cannot take on negative values, it may be more reasonable to assume that the data within each study follow a lognormal distribution. If IPD are available, the preferred approach would be to first log-transform the individual observations, estimate the reference interval, then exponentiate the resulting bounds. However, if only aggregate data are available, the observed means and standard deviations need to be transformed to the log scale. Suppose $Y = \{y_1, \dots, y_n\}$ denotes a set of continuous observations. Because, $\frac{1}{n}\sum_{i=1}^{n} \log(y_i) \neq \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i\right)$, the observed means and sample variances on the log-scale must be estimated. The method of moments estimators for the mean and variance of log(Y) are given in Box 2. These equations can be used to estimate the observed means and sample variances, which can then be used in each of the methods described to estimate the reference range. Finally, the resulting range can be exponentiated in order to return to the original scale. We note that when performing either of these transformations, the normality assumption for the study means now applies to the log-transformed data and should still be assessed using methods such as a normal Q-Q plot, as the transformed means may be skewed.

Let $\overline{y_i}$, s_i^2 , and n_i be the sample mean, sample variance, and sample size for study i, respectively. The method of moments estimators for the location and scale parameters of the log-normal distribution are given by*:*

$$\bar{y}_i^* = \log\left(\frac{\bar{y}_i}{\sqrt{1 + \frac{\frac{n_i - 1}{n_i} s_i^2}{\bar{y}_i^2}}}\right)$$

$$s_i^{2^*} = \log\left(1 + \frac{\frac{n_i - 1}{n_i} s_i^2}{\bar{y}_i^2}\right)$$

We can then use $\bar{y}_i^*$ and $s_i^{2^*}$ as estimates of the mean and sample standard deviation on the log scale when individual participant data are not available.

Note: We assume $s_i^2 = \frac{1}{n-1}\sum_{j=1}^{n_i}\left(y_{ij} - \bar{y}_i\right)^2$

Web Appendix 3

Estimating the Reference Range using Aggregate Data Methods

Code for Frequentist Approach

```
library(meta)
library(rjags)
library(coda)
library(data.table)
library(ggplot2)
library(readxl)
library(dplyr)
library(ggsci)

### Function for Frequentist-based Reference Range based on
### REML Takes in study means, study standard deviations, study
### sample sizes (n)
FreqFit = function(means, sds, n) {
    N = length(means)
    means.use = means
    sds.use = sds

    # fit random effects model using REML
    m.reml = metagen(means.use, sds.use/sqrt(n), comb.fixed = FALSE,
        comb.random = TRUE, method.tau = "REML", hakn = FALSE,
        prediction = TRUE)

    # pooled variance - assumes study populations drawn
    # independently
    sigma_hat = sqrt(sum((n - 1) * sds.use^2)/sum(n - 1))

    # Estimates from RE model
    tau_hat = m.reml$tau
    mu_hat = m.reml$TE.random

    # Estimate of total variance (within and between studies)
    total.var = sigma_hat^2 + tau_hat^2

    # Estimates for limits of reference range
    lower_limit = qnorm(0.025, mean = mu_hat, sd = sqrt(total.var))
    upper_limit = qnorm(0.975, mean = mu_hat, sd = sqrt(total.var))

    return(list(m.reml, lower_limit, upper_limit))
}
```

Code for Bayesian Approach

Model Specification:

```
cat(readLines("RefRangeRandomModel.txt"), sep = "\n")

model{
for (i in 1:length(y)){
    theta[i] ~ dnorm(mu, 1/tau^2)
    y[i] ~ dnorm(theta[i], n[i]/(sigma^2))

    #dist of sigma
    x[i] ~ dgamma((n[i]-1)/2, 1/(2*sigma^2))
    }

#posterior predictive interval
new ~ dnorm(mu, 1/(sigma^2 + tau^2))


# priors
mu ~ dnorm(0, 0.001)
tau ~ dunif(0,100)
sigma ~ dunif(0,100)
beta ~ dnorm(0, 0.001)

}
```

R Function:

```
### Function for Bayesian Reference Range Takes in study means,
### study standard deviations, study sample sizes (n) Fits
### Bayesian model using packages rjags and coda
BayesFit = function(means, sds, n, n.iter = 50000) {
    N = length(means)
    y = means
    sd = sds
    x = (n - 1) * (sd^2)

    data = list(y = y, x = x, n = n)
    Inits1 = list(.RNG.name = "base::Mersenne-Twister", .RNG.seed = 123)
    Inits2 = list(.RNG.name = "base::Mersenne-Twister", .RNG.seed = 124)
    Inits = list(Inits1, Inits2)

    jags.model = jags.model(file = "RefRangeRandomModel.txt",
        data = data, inits = Inits, n.chain = 2, quiet = T)

    burn.in = 5000
    update(jags.model, n.iter = burn.in)
    params = c("tau", "mu", "sigma", "new")
    samps = coda.samples(jags.model, params, n.iter = n.iter,
        .RNG.name = "base::Mersenne-Twister", .RNG.seed = 123)
```

```
    results = list(samps, summary(samps)$quantiles["new", "2.5%"],
        summary(samps)$quantiles["new", "97.5%"])

    return(results)
}
```

Code for Empirical Approach

```
# Fit empirical method to data takes in study means, sds, and
# sample sizes (n)
EmpFit = function(means, sds, n) {
    N = length(means)

    sigma_hat = sqrt(sum((n - 1) * sds^2)/sum(n - 1))
    mu_hat = sum(n * means)/sum(n)
    var.e.w = sum((n - 1) * (means - mu_hat)^2)/sum(n - 1)
    total.var = var.e.w + sigma_hat^2

    lower_limit = qnorm(0.025, mean = mu_hat, sd = sqrt(total.var))
    upper_limit = qnorm(0.975, mean = mu_hat, sd = sqrt(total.var))


    return(list(lower_limit, upper_limit))
}
```

Web Appendix 4

Data for Liver Stiffness Applied Example

Identifying Target Population and IPD Dataset

We first limit the dataset to the group of "healthy non-obese individuals" used by Bazerbachi et al.[8] in order to get the individual participant used later. Bazerbachi et al. [8] identified datasets through a systematic review of the literature and by contacting primary and corresponding authors. Those studies were published between 2006 and 2016, and some patients/datasets were provided by corresponding authors from unpublished work. Moreover, some studies/patients were excluded from this current analysis given data-sharing agreements (participants from a study conducted in the Netherlands). For breakdown and details regarding the original study, please refer to Bazerbach et al. [8].

```
# import data file
liver = read_excel("Liverstiffness_adults_final.xls")

# need to restrict to 'truly healthy non-obese individuals'
# BMI < 30, no hypertension, dyslipidemia, hepatic steatosis
# on ultrasound, diabetes liver$Dyslipidemia != 1 &
# liver$steatosis != 1 & liver$FHO_Diabetes!= 1 This gives
# dataset with 20 studies
healthy = subset(liver, bmi < 30 & is.na(bmi) == F & steatosis ==
    0 & DM == 0 & HTN == 0 & Dyslipidemia == 0 & is.na(stiff_measure) ==
    F)

# Because study labeled #4 only has four individuals, we will
# further exclude this study summary(factor(healthy$studyID))
healthy2 = subset(healthy, studyID != 4)

# this gives final dataset of 3648 individuals and 19 studies
# nrow(healthy2) length(unique(healthy2$studyID))

# we will also rename the study ID's so that they range from
# 1-19
healthy2$s = as.numeric(factor(as.numeric(as.factor(healthy2$studyID))))
```

Deriving Aggregate Dataset

We will first consider the situation where only aggregate data are available for the liver stiffness measurements from each study. We present a table with the observed study means, standard deviations, and sample sizes:

```
# summarize each study by mean, sd, and sample size
# (consistent with what would usually be available in
# practice)
summary.data = healthy2 %>% group_by(s) %>% summarise(mean = mean(stiff_measu
re),
    sd = sd(stiff_measure), n = n())

knitr::kable(summary.data, col.names = c("Study ID", "Mean",
    "SD", "n"), digits = 3, booktabs = T, longtable = F, caption = "Aggregate
Data for Liver Stiffness Example")
```

Web Table 3. Aggregate data for liver stiffness example

Study ID	Mean	SD	n
1	4.659	1.377	581
2	4.112	0.891	530
3	3.972	1.001	67
4	4.569	1.291	248
5	4.527	1.305	206
6	4.990	1.076	183
7	4.198	1.176	60
8	4.669	1.184	35
9	5.050	4.421	34
10	4.816	1.216	132
11	5.199	1.385	420
12	4.826	1.254	90
13	5.091	1.177	433
14	4.516	1.508	498
15	5.450	1.865	52
16	5.176	0.684	29
17	5.167	1.135	9
18	3.827	0.669	15
19	4.358	1.367	26

Because liver stiffness cannot be negative and we believe it may be skewed (see the individual participant data section for more detailed exploration), we first log-transform the data and then exponentiate the resulting bounds of the estimated reference ranges. Here, we assume only the aggregate data are available (means, sd's, and sample size), so we must use an approximation (described in Web Appendix 2) in order to do the transformation. This approximation relies on the assumption that the data follow a log-normal distribution.

```
# log transformation to get estimates of mean(log(liver
# stiffness)) and sd(log(liver stiffness)) summary.data is
# the data frame with the aggregate data for each study (skip
# this step if not log-transforming the data)
summary.data$logmean = with(summary.data, log(mean/sqrt(1 + ((n -
    1)/(n)) * sd^2/mean^2)))
summary.data$logsd = with(summary.data, sqrt(log(1 + ((n - 1)/(n)) *
    sd^2/mean^2)))
```

Web Appendix 5

Liver Stiffness Applied Example - Aggregate Data

Traditional Forest Plot with Study Means (CI's) and Pooled Mean (CI)

We estimate the 95% confidence intervals for the study means:

```
summary.data$Lower.mean = summary.data$mean - qt(0.975, summary.data$n -
    1) * summary.data$sd/sqrt(summary.data$n)
summary.data$Upper.mean = summary.data$mean + qt(0.975, summary.data$n -
    1) * summary.data$sd/sqrt(summary.data$n)
```

We estimate the pooled mean (and CI) using a frequentist random effects model (REML) implemented in the R package "meta".

```
# fit random effects model using REML
m.reml = metagen(summary.data$mean, summary.data$sd/sqrt(summary.data$n),
    comb.fixed = FALSE, comb.random = TRUE, method.tau = "REML",
    hakn = FALSE, prediction = TRUE)
m.reml

##                      95%-CI %W(random)
## 1  4.6594 [4.5474; 4.7714]        6.0
## 2  4.1115 [4.0357; 4.1873]        6.1
## 3  3.9716 [3.7318; 4.2114]        5.7
## 4  4.5694 [4.4087; 4.7300]        5.9
## 5  4.5272 [4.3490; 4.7054]        5.9
```

```
## 6  4.9902 [4.8343; 5.1460]        5.9
## 7  4.1983 [3.9007; 4.4960]        5.4
## 8  4.6686 [4.2764; 5.0608]        5.0
## 9  5.0500 [3.5640; 6.5360]        1.4
## 10 4.8159 [4.6084; 5.0234]        5.8
## 11 5.1988 [5.0663; 5.3313]        6.0
## 12 4.8256 [4.5665; 5.0846]        5.6
## 13 5.0905 [4.9797; 5.2014]        6.0
## 14 4.5163 [4.3838; 4.6487]        6.0
## 15 5.4500 [4.9431; 5.9569]        4.5
## 16 5.1759 [4.9270; 5.4247]        5.6
## 17 5.1667 [4.4254; 5.9080]        3.4
## 18 3.8267 [3.4880; 4.1653]        5.3
## 19 4.3577 [3.8321; 4.8833]        4.4
##
## Number of studies combined: k = 19
##
##                                        95%-CI     z p-value
## Random effects model 4.6673 [4.4634; 4.8712] 44.86       0
## Prediction interval         [3.7556; 5.5790]
##
## Quantifying heterogeneity:
##  tau^2 = 0.1759 [0.0882; 0.4133]; tau = 0.4194 [0.2970; 0.6429]
##  I^2 = 95.9% [94.6%; 96.8%]; H = 4.92 [4.31; 5.62]
##
## Test of heterogeneity:
##       Q d.f.  p-value
##  435.97   18 < 0.0001
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau

# make vector of results to add to dataset to make forest
# plot
pooled.mean = c(m.reml$TE.random)
pooled.mean.results = c("Pooled Mean", m.reml$TE.random, NA,
    nrow(healthy2), NA, NA, m.reml$lower.random, m.reml$upper.random)
```

Finally, we make the forest plot (Figure 2 in main paper):

```
pooled.mean.results2 = t(pooled.mean.results)
colnames(pooled.mean.results2) = names(summary.data)
#make sure studies are in correct order
forest1 = rbind(summary.data, pooled.mean.results2, use.names = F)[1:20,]
forest1$s = c(1:19,"Overall")
forest1$s = factor(forest1$s, levels = c(as.character(1:19), "Overall"))

intervals <- c(" Mean (95% CI)", paste0(sprintf("%.2f",
```

```
as.numeric(forest1$mean)), " ",
                    "(",sprintf("%.2f", as.numeric(forest1$Lower.mean)),
                    ", ", sprintf("%.2f", as.numeric(forest1$Upper.mean)), ")
"))
row_names <- cbind(forest1$s, intervals)

xticks1 <- seq(from = 3, to = 6, by =1)
xtlab <- signs(xticks1) # using minus signs for negative numbers instead of h
yphens for axis labels
attr(xticks1, "labels") <- xtlab

mean <- c(NA, forest1$mean)
lower <- c(NA, forest1$Lower.mean)
upper <- c(NA, forest1$Upper.mean)

fig2 <- forestplot(row_names, mean, lower, upper,
                   txt_gp = fpTxtGp(ticks = gpar(cex = 0.9),
                                    xlab  = gpar(cex = 0.9),
                                    label = gpar(cex = 0.9)),
                   hrzl_lines = list("2" = gpar(lwd = 1, columns = c(1,3), co
l = "#000044")),
                   align = "l",
                   fn.ci_norm = c(rep("fpDrawNormalCI",21)),
                   graph.pos = 2,
                   zero = 4.49,
                   boxsize = 0.2,
                   clip = c(1, 7),
                   xlab = "Liver Stiffnes (kPa)",
                   col=fpColors(line="black", zero="black", box="black"),
                   new_page = TRUE,
                   xticks = xticks1)
```

Estimate Reference Range using Aggregate Data Methods

```
set.seed(123)

# Frequentist AD method
freq.results = FreqFit(summary.data$logmean, summary.data$logsd,
    summary.data$n)
freq.range = exp(c(freq.results[[2]], freq.results[[3]]))

# Bayesian AD method
bayes.results = BayesFit(summary.data$logmean, summary.data$logsd,
    summary.data$n, n.iter = 1e+05)
summary(bayes.results[[1]])

##
## Iterations = 6001:106000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 1e+05
```

```
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean       SD  Naive SE Time-series SE
## mu    1.49686 0.024824 5.551e-05      6.273e-05
## new   1.49743 0.293070 6.553e-04      6.606e-04
## sigma 0.27354 0.003204 7.164e-06      8.456e-06
## tau   0.09981 0.021129 4.725e-05      8.298e-05
##
## 2. Quantiles for each variable:
##
##          2.5%     25%     50%    75%  97.5%
## mu    1.44763 1.48078 1.49695 1.5129 1.5460
## new   0.92402 1.30020 1.49644 1.6946 2.0725
## sigma 0.26735 0.27136 0.27350 0.2757 0.2799
## tau   0.06671 0.08489 0.09706 0.1116 0.1491
```

```
gelman.diag(bayes.results[[1]])
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## mu             1          1
## new            1          1
## sigma          1          1
## tau            1          1
##
## Multivariate psrf
##
## 1
```

```
bayes.range = exp(c(bayes.results[[2]], bayes.results[[3]]))

# Empirical method
emp.results = EmpFit(summary.data$logmean, summary.data$logsd,
    summary.data$n)
emp.range = exp(c(emp.results[[1]], emp.results[[2]]))

# Table with AD results
tbl2 = rbind(freq.range, bayes.range, emp.range)
rownames(tbl2) = c("Frequentist AD", "Bayesian AD", "Empirical AD")
colnames(tbl2) = c("Lower", "Upper")
knitr::kable(tbl2, digits = 2, caption = "Estimated 95% reference ranges for
liver stiffness measurement using each of the methods presented with aggregat
e data. The reference ranges were estimated on the log-scale and the resultin
g intervals were exponentiated")
```

We add the estimated reference ranges to a forest plot along with the prediction interval for the mean of a new study to obtain Figure 3 in the main paper:

```
# 95% prediction interval for new study mean
lower.predict = m.reml$lower.predict
upper.predict = m.reml$upper.predict
predict.results = c("Overall", pooled.mean, NA, nrow(healthy2),
    NA, NA, lower.predict, upper.predict)

# 95% prediction intervals for each study
predict.ind.results = list()
for (i in 1:19) {
    lower.predict.ind = summary.data$mean[i] - qt(0.975, summary.data$n[i] -
        1) * sqrt(1 + 1/summary.data$n[i]) * summary.data$sd[i]
    upper.predict.ind = summary.data$mean[i] + qt(0.975, summary.data$n[i] -
        1) * sqrt(1 + 1/summary.data$n[i]) * summary.data$sd[i]
    predict.ind.results[[i]] = c(i, summary.data$mean[i], NA,
        summary.data$n[i], NA, NA, lower.predict.ind, upper.predict.ind)
}

predict.ind = do.call("rbind", predict.ind.results)

# reference ranges
ref = c("95% Reference Range", pooled.mean, NA, nrow(healthy2),
    NA, NA)
refrange.results = matrix(c(predict.results, ref, freq.range,
    ref, bayes.range, ref, emp.range), nrow = 4, byrow = T)
colnames(predict.ind) = names(forest1)
colnames(refrange.results) = names(forest1)
forest2 = rbind(forest1[1:19, ], predict.ind, forest1[20, ],
    refrange.results, use.names = F)[1:43, ]
forest2$IntervalType = factor(c(rep("95% CI for Study Mean",
    19), rep("95% Prediction Interval", 19), "95% CI for Pooled Mean",
    "95% Prediction Interval for New Study Mean", "Frequentist AD Reference R
ange",
    "Bayesian AD Reference Range", "Empirical AD Reference Range"),
    levels = rev(c("95% CI for Study Mean", "95% Prediction Interval",
        "95% CI for Pooled Mean", "95% Prediction Interval for New Study Mean
",
        "Frequentist AD Reference Range", "Bayesian AD Reference Range",
        "Empirical AD Reference Range")))

int.types <- c("           Mean (CI)", "           PI")
l_col <- c("      Study and Interval Type", "       1", int.types,
    "       2", int.types, "       3", int.types, "       4",
    int.types, "       5", int.types, "       6", int.types,
    "       7", int.types, "       8", int.types, "       9",
    int.types, "       10", int.types, "       11", int.types,
```

```
    "       12", int.types, "       13", int.types, "       14",
    int.types, "       15", int.types, "       16", int.types,
    "       17", int.types, "       18", int.types, "       19",
    int.types, "       Overall", "           Pooled Mean (CI)",
    "           PI for New Study Mean", "           Frequentist AD RR",
    "           Bayesian AD RR", "           Empirical AD RR")

r_col <- c()
for (i in 1:19) {
    ci <- paste0(sprintf("%.2f", forest2$mean[i]), " (", sprintf("%.2f",
        forest2$Lower.mean[i]), ", ", sprintf("%.2f", forest2$Upper.mean[i]),
        ")")
    pi <- paste0("        (", sprintf("%.2f", forest2$Lower.mean[i +
        19]), ", ", sprintf("%.2f", forest2$Upper.mean[i + 19]),
        ")")
    ints <- c(NA, ci, pi)
    r_col <- c(r_col, ints)
}

r_col <- c("         Estimate", r_col, NA, paste0(sprintf("%.2f",
    forest2$mean[39]), " (", sprintf("%.2f", forest2$Lower.mean[39]),
    ", ", sprintf("%.2f", forest2$Upper.mean[39]), ")"), paste0("        (",
    sprintf("%.2f", forest2$Lower.mean[40:43]), ", ", sprintf("%.2f",
        forest2$Upper.mean[40:43]), ")"))

row_names <- cbind(l_col, r_col)

xticks2 <- seq(from = 2, to = 10, by = 2)
xtlab2 <- signs(xticks2)  # using minus signs for negative numbers instead of
hyphens for axis labels
attr(xticks2, "labels") <- xtlab2

mean <- c(NA, rep(c(NA, 1, 2), 19))
mean[which(mean == 1)] <- forest2[forest2$IntervalType == "95% CI for Study M
ean",
    ]$mean
mean[which(mean == 2)] <- forest2[forest2$IntervalType == "95% Prediction Int
erval",
    ]$mean
mean <- c(mean, NA, forest2$mean[39:43])

lower <- c(NA, rep(c(NA, 1, 2), 19))
lower[which(lower == 1)] <- forest2[forest2$IntervalType == "95% CI for Study
Mean",
    ]$Lower.mean
lower[which(lower == 2)] <- forest2[forest2$IntervalType == "95% Prediction I
nterval",
    ]$Lower.mean
lower <- c(lower, NA, forest2$Lower.mean[39:43])
```

```
upper <- c(NA, rep(c(NA, 1, 2), 19))
upper[which(upper == 1)] <- forest2[forest2$IntervalType == "95% CI for Study
Mean",
    ]$Upper.mean
upper[which(upper == 2)] <- forest2[forest2$IntervalType == "95% Prediction I
nterval",
    ]$Upper.mean
upper <- c(upper, NA, forest2$Upper.mean[39:43])

fig3 <- forestplot(row_names, mean, lower, upper, txt_gp = fpTxtGp(ticks = gp
ar(cex = 0.9),
    xlab = gpar(cex = 0.9), label = gpar(cex = 0.9)), hrzl_lines = list(`2` =
gpar(lwd = 1,
    columns = c(1, 3), col = "#000044")), align = "l", fn.ci_norm = c(rep("fp
DrawNormalCI",
    64)), graph.pos = 2, zero = forest2$mean[39], boxsize = 0.2,
    clip = c(1, 7), xlab = "Liver Stiffnes (kPa)", col = fpColors(line = "bla
ck",
        zero = "black", box = "black"), new_page = TRUE, xticks = xticks2)
fig3
```

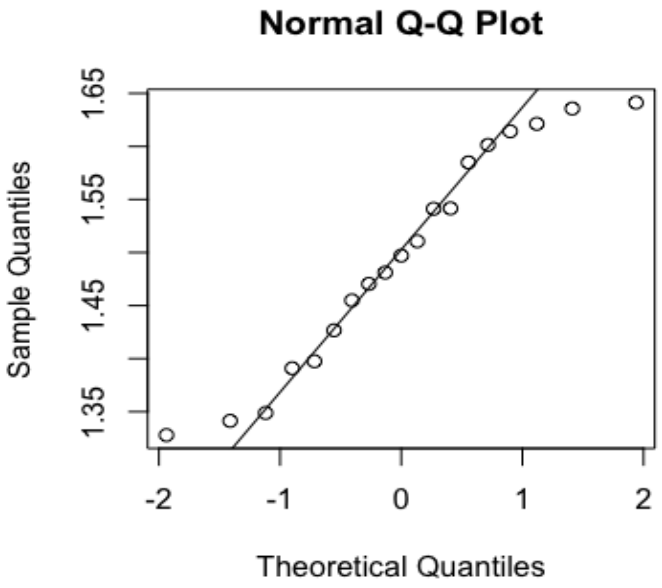
Assessing Model Assumptions

The frequentist and Bayesian approaches require the assumption that the study means are normally distributed and that the variances within each study are equal (any differences are due to sampling variability). We use normal QQ plots in order to assess the normality of the study means and a forest plot of the study standard deviations (and CI's) in order to assess the equal variance assumption. Additionally, both methods also assume that the data within each study are normally distributed (and the empirical method assumes that the overall distribution is normal), but it is not possible to assess these assumptions with only aggregate data available. See the section on individual participant data for assessment of these assumptions.

Because we are first estimating the reference ranges on the log-scale and then exponentiating the resulting intervals, we assess these assumptions based on the log-transformed means and standard deviations.

```
## normal QQ plot of log-transformed study means
qqnorm(summary.data$logmean)
qqline(summary.data$logmean)
```

Web Figure 1. Normal Q-Q plot of log-transformed study means

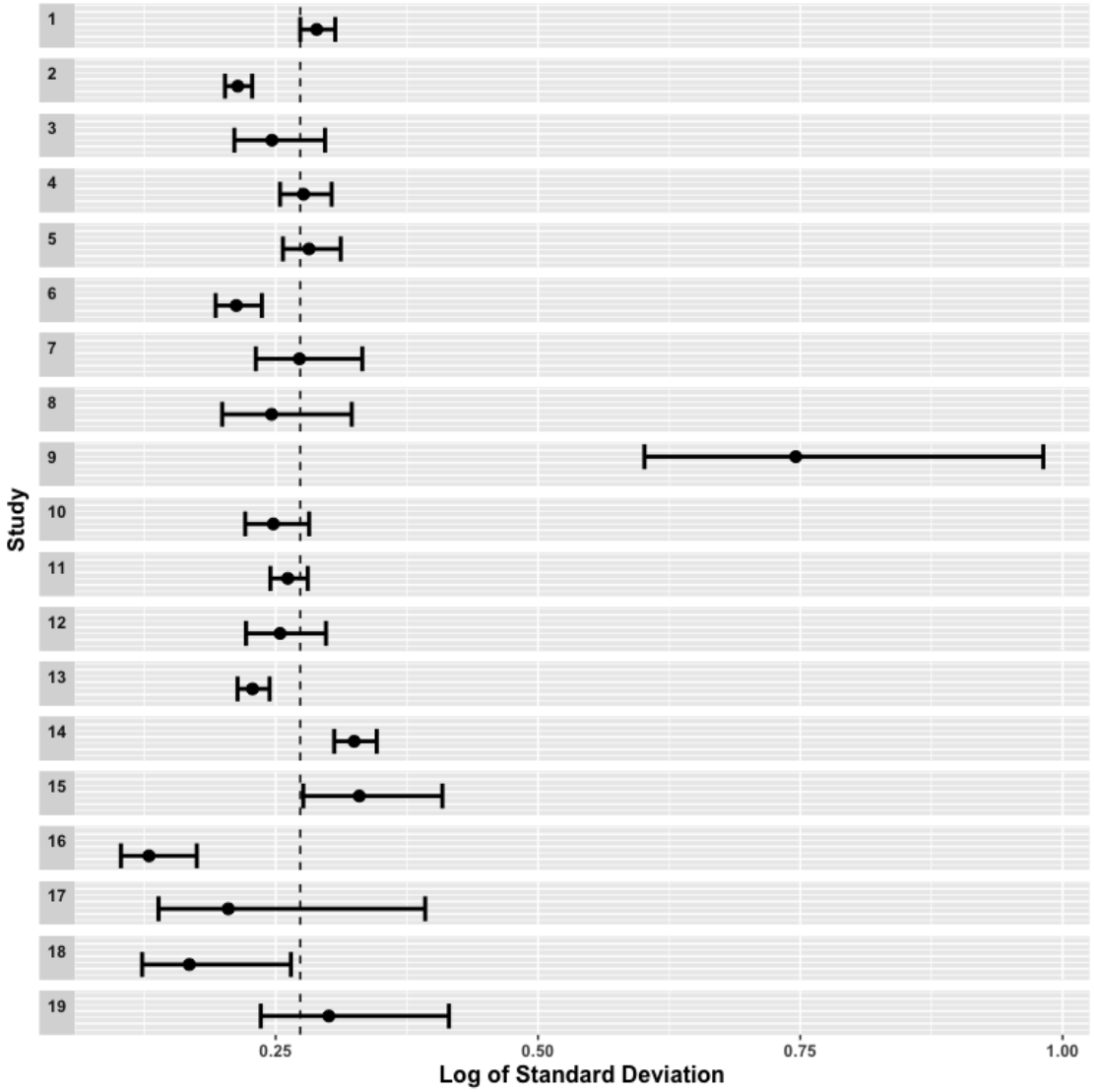


Based on the normal QQ plot (above) there are no clear violations to the assumption that the log-transformed study means are normally distributed.

```
# forest plot of log transformed study standard deviations
# 95% CI's for standard deviation of log liver stiffness
lower.sd = sqrt(summary.data$logsd^2 * (summary.data$n - 1)/qchisq(0.025,
    summary.data$n - 1, lower.tail = F))
upper.sd = sqrt(summary.data$logsd^2 * (summary.data$n - 1)/qchisq(0.025,
    summary.data$n - 1, lower.tail = T))
sd.dat = as.data.frame(cbind(summary.data$s, summary.data$logsd,
    lower.sd, upper.sd))
names(sd.dat) = c("s", "logsd", "lower.sd", "upper.sd")
pooled.sd = sqrt(sum((summary.data$n - 1) * summary.data$logsd^2)/sum(summary
.data$n -
    1))

p3 = ggplot(sd.dat,
            aes(x = logsd,
                xmin = lower.sd,
                xmax = upper.sd))+
  geom_hline(yintercept = pooled.sd, linetype=2)+
  xlab('Study')+ ylab("Log of Standard Deviation")+
  facet_wrap(~s,strip.position="left",nrow=20) +
    geom_errorbar(aes(y = logsd,
                      ymin= lower.sd,
                      ymax= upper.sd),
                  width=1,cex=1)+
```

```
  geom_point(aes(y = sd.dat$logsd), size = 2.5)+
  theme(plot.title=element_text(size=16,face="bold"),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_text(face="bold"),
        axis.title=element_text(size=12,face="bold"),
        legend.title = element_blank(),
        strip.text.y.left = element_text(hjust=0,vjust = 1,angle=0,face="bold
"))+
  coord_flip() + guides(fill = guide_legend(reverse = TRUE))  +
  guides(color = guide_legend(reverse = TRUE))
p3
```

Web Figure 2. Forest plot showing standard deviations of the log of liver stiffness measurements for each study and corresponding 95% confidence intervals. The observed standard deviations in studies 9 and 16 look as though they may differ from the others.



Based on the forest plot above, it seems as though the standard deviation for Study #9 may be different, as well as possibly Study #16. While the simulation results shown by Siegel et al.[9] suggest that these models may be robust to small differences in the study standard deviations, we conduct a sensitivity analysis where we remove the two aformentioned studies and re-estimate the reference ranges in order to compare the results.

Sensitivity Analysis for Equal Within-Study Standard Deviations Assumption

```
summary.data2 = subset(summary.data, !(s %in% c(9, 16)))
# Frequentist AD method
freq.results2 = FreqFit(summary.data2$logmean, summary.data2$logsd,
    summary.data2$n)
freq.range2 = exp(c(freq.results2[[2]], freq.results2[[3]]))
```

```r
# Bayesian AD method
bayes.results2 = BayesFit(summary.data2$logmean, summary.data2$logsd,
    summary.data2$n, n.iter = 1e+05)
summary(bayes.results2[[1]])
```

```
##
## Iterations = 6001:106000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 1e+05
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean       SD  Naive SE Time-series SE
## mu    1.49822 0.024940 5.577e-05      6.239e-05
## new   1.49890 0.284781 6.368e-04      6.364e-04
## sigma 0.26612 0.003152 7.049e-06      8.276e-06
## tau   0.09473 0.021393 4.784e-05      8.498e-05
##
## 2. Quantiles for each variable:
##
##          2.5%     25%     50%    75%  97.5%
## mu    1.44855 1.48232 1.49825 1.5142 1.5476
## new   0.94221 1.30708 1.49871 1.6903 2.0588
## sigma 0.26005 0.26396 0.26609 0.2682 0.2724
## tau   0.06193 0.07961 0.09173 0.1064 0.1451
```

```r
gelman.diag(bayes.results2[[1]])
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## mu             1          1
## new            1          1
## sigma          1          1
## tau            1          1
##
## Multivariate psrf
##
## 1
```

```
bayes.range2 = exp(c(bayes.results2[[2]], bayes.results2[[3]]))

# Empirical method
emp.results2 = EmpFit(summary.data2$logmean, summary.data2$logsd,
    summary.data2$n)
emp.range2 = exp(c(emp.results2[[1]], emp.results2[[2]]))

# Table with AD results
tbl2 = rbind(freq.range2, bayes.range2, emp.range2)
rownames(tbl2) = c("Frequentist AD", "Bayesian AD", "Empirical")
colnames(tbl2) = c("Lower", "Upper")
knitr::kable(tbl2, digits = 2, caption = "Results of Sensitivity Analysis Rem
oving Studies 9 and 16")
```

Web Table 4. Results of sensitivity analysis removing studies 9 and 16

	Lower	Upper
Frequentist AD	2.58	7.74
Bayesian AD	2.57	7.84
Empirical	2.61	7.75

We can see that the estimated reference ranges after excluding these studies are similar to the original results.

Web Appendix 6

Estimating the Reference Range with Individual Participant Data

R Code for Frequentist Approach

If individual participant data are available, rather than fitting a model with the aggregate data, we can directly fit a generalized linear mixed model on the individual participant data in a one-step approach.

```
library(lme4)
library(ggeffects)
# used to extract fixed effect
library(nlme)
# used to get variance estimates
library(insight)

FreqIPD = function(y, s) {
    # fit LMM directly on IPD
    freq.lmer = lmer(y ~ 1 + (1 | s))

    # get estimates of mu, sigma2, and tau2
    lmer.mu = fixed.effects(freq.lmer)[1]
    lmer.sigma2 = get_variance_residual(freq.lmer)
    lmer.tau2 = get_variance_random(freq.lmer)

    # reference range estimated by exponentiated 2.5th and 97.5th
    # quantiles of N(mu, sigma^2 + tau^2) distribution
    freq.ipd.lower = qnorm(0.025, mean = lmer.mu, sd = sqrt(lmer.tau2 +
        lmer.sigma2))
    freq.ipd.upper = qnorm(0.975, mean = lmer.mu, sd = sqrt(lmer.tau2 +
        lmer.sigma2))
    freq.ipd.results = c(freq.ipd.lower, freq.ipd.upper)
    return(list(freq.lmer, freq.ipd.results))
}
```

R Code for Bayesian Approach

Alternatively, we can use a similar one-step approach but when estimating the Bayesian posterior predictive interval:

Model Specification

```
cat(readLines("RefRangeLMM.txt"), sep = "\n")

model{
#uses vector of obs (y) and vector of study assignments (s)
for (i in 1:length(y)){
    beta0[i] ~ dnorm(mu, 1/tau^2)
    y[i] ~ dnorm(beta0[s[i]], 1/sigma^2)
```

```
}

#posterior predictive interval for new indiv.
new ~ dnorm(mu, 1/(sigma^2 + tau^2))

#posterior predictive interval for new study
new_study ~ dnorm(mu, 1/tau^2)

#priors
mu ~ dnorm(0, 0.001)
tau ~ dunif(0, 100)
sigma ~ dunif(0, 100)
}
```

R Function

```
### Estimates Bayesian posterior predictive interval using IPD
### (i.e. one step approach) Also estimates Bayesian posterior
### predictive interval for mean of a new study
BayesLMM = function(data = data, studyID = studyID, outcome = outcome,
    iterations = 50000, burn.in = 5000, n.thin = 2) {
    s = as.matrix(data[, studyID])[, 1]
    s = as.numeric(s)
    y = as.matrix(data[, outcome])[, 1]

    Inits1 = list(.RNG.name = "base::Mersenne-Twister", .RNG.seed = 123)
    Inits2 = list(.RNG.name = "base::Mersenne-Twister", .RNG.seed = 124)
    Inits = list(Inits1, Inits2)
    data = list(y = y, s = s)
    jags.model = jags.model(file = "RefRangeLMM.txt", data = data,
        inits = Inits, n.chain = 2, quiet = T)
    burn.in = 5000
    update(jags.model, n.iter = burn.in)
    params = c("tau", "mu", "sigma", "new", "new_study")
    samps = coda.samples(jags.model, params, n.iter = iterations,
        .RNG.name = "base::Mersenne-Twister", .RNG.seed = 123)
    results = list(samps, summary(samps)$quantiles["new", "2.5%"],
        summary(samps)$quantiles["new", "97.5%"])
    return(results)
}
```

Web Appendix 7

Liver Stiffness Applied Example - Individual Participant Data

We now demonstrate how to estimate the reference range using IPD in the applied example, as well as how IPD can allow for more detailed assessment of the model assumptions. We also note that with IPD, we can directly log transform the individual liver

stiffness measurements for each patient, rather than relying on the approximation used previously with the aggregate data.

Estimating the Reference Ranges Using IPD

Here, we show how to estimate the reference ranges using the one-step approaches described previously.

```
## Refit Frequentist LMM on log-transformed data to satisfy
## within-study/overall normality assumption for frequentist
## reference range method Frequentist LMM (IPD)
freq.ipd2 = FreqIPD(log(healthy2$stiff_measure), healthy2$s)
freq.ipd.range = exp(freq.ipd2[[2]])

## Bayesian posterior predictive interval
healthy2$log.lsm = log(healthy2$stiff_measure)
bayes.ipd = BayesLMM(data = healthy2, studyID = "s", outcome = "log.lsm",
    iterations = 1e+05)
summary(bayes.ipd[[1]])

##
## Iterations = 6001:106000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 1e+05
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean       SD  Naive SE Time-series SE
## mu        1.50527 0.023147 5.176e-05      5.837e-05
## new       1.50612 0.278981 6.238e-04      6.305e-04
## new_study 1.50534 0.097686 2.184e-04      2.192e-04
## sigma     0.26142 0.003057 6.835e-06      7.964e-06
## tau       0.09276 0.019669 4.398e-05      7.668e-05
##
## 2. Quantiles for each variable:
##
##              2.5%    25%     50%    75%  97.5%
## mu        1.45918 1.4904 1.50531 1.5202 1.5511
## new       0.96041 1.3190 1.50637 1.6942 2.0558
## new_study 1.31121 1.4428 1.50532 1.5679 1.6992
## sigma     0.25551 0.2594 0.26140 0.2635 0.2675
## tau       0.06205 0.0789 0.09014 0.1037 0.1386

gelman.diag(bayes.ipd[[1]])
```

```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## mu                 1          1
## new                1          1
## new_study          1          1
## sigma              1          1
## tau                1          1
##
## Multivariate psrf
##
## 1
```

```
bayes.ipd.range = exp(c(bayes.results[[2]], bayes.results[[3]]))

# Empirical method - IPD
mean.ov = mean(healthy2$log.lsm)
sd.ov = sd(healthy2$log.lsm)
lower.ov = qnorm(0.025, mean = mean.ov, sd = sd.ov)
upper.ov = qnorm(0.975, mean = mean.ov, sd = sd.ov)
emp.range.ipd = exp(c(lower.ov, upper.ov))
```

The aggregate data approaches used previously are still valid when individual participant data are available; however, we no longer need to use the aggregate data log-transformation. Because of this, we re-estimate the AD reference ranges when directly taking the log of the individual measurements.

```
# Frequentist AD method
freq.results2 = FreqFit(summary.ipd$logmean, summary.ipd$logsd,
    summary.ipd$n)
freq.range2 = exp(c(freq.results2[[2]], freq.results2[[3]]))

# Bayesian AD method
bayes.results2 = BayesFit(summary.ipd$logmean, summary.ipd$logsd,
    summary.ipd$n, n.iter = 1e+05)
summary(bayes.results2[[1]])

##
## Iterations = 6001:106000
## Thinning interval = 1
## Number of chains = 2
```

```
## Sample size per chain = 1e+05
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean       SD  Naive SE Time-series SE
## mu    1.50522 0.023144 5.175e-05      5.825e-05
## new   1.50647 0.279514 6.250e-04      6.250e-04
## sigma 0.26144 0.003081 6.889e-06      8.140e-06
## tau   0.09265 0.019541 4.370e-05      7.620e-05
##
## 2. Quantiles for each variable:
##
##          2.5%     25%     50%    75%  97.5%
## mu    1.45903 1.49031 1.50527 1.5202 1.5510
## new   0.95888 1.31809 1.50678 1.6958 2.0532
## sigma 0.25548 0.25935 0.26141 0.2635 0.2676
## tau   0.06209 0.07879 0.09007 0.1036 0.1382
```

```
gelman.diag(bayes.results2[[1]])
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## mu             1          1
## new            1          1
## sigma          1          1
## tau            1          1
##
## Multivariate psrf
##
## 1
```

```
bayes.range2 = exp(c(bayes.results2[[2]], bayes.results2[[3]]))

# Empirical method - AD
emp.results2 = EmpFit(summary.ipd$logmean, summary.ipd$logsd,
    summary.ipd$n)
emp.range2 = exp(c(emp.results2[[1]], emp.results2[[2]]))
```

Because we have IPD available, we can compare these ranges to the 2.5th and 97.5th quantiles of the pooled data (ignoring study assignment):

```
quants = quantile(healthy2$stiff_measure, probs = c(0.025, 0.5,
    0.975))
quants
```

```
##    2.5%     50%   97.5%
## 2.70000 4.50000 7.49125
```

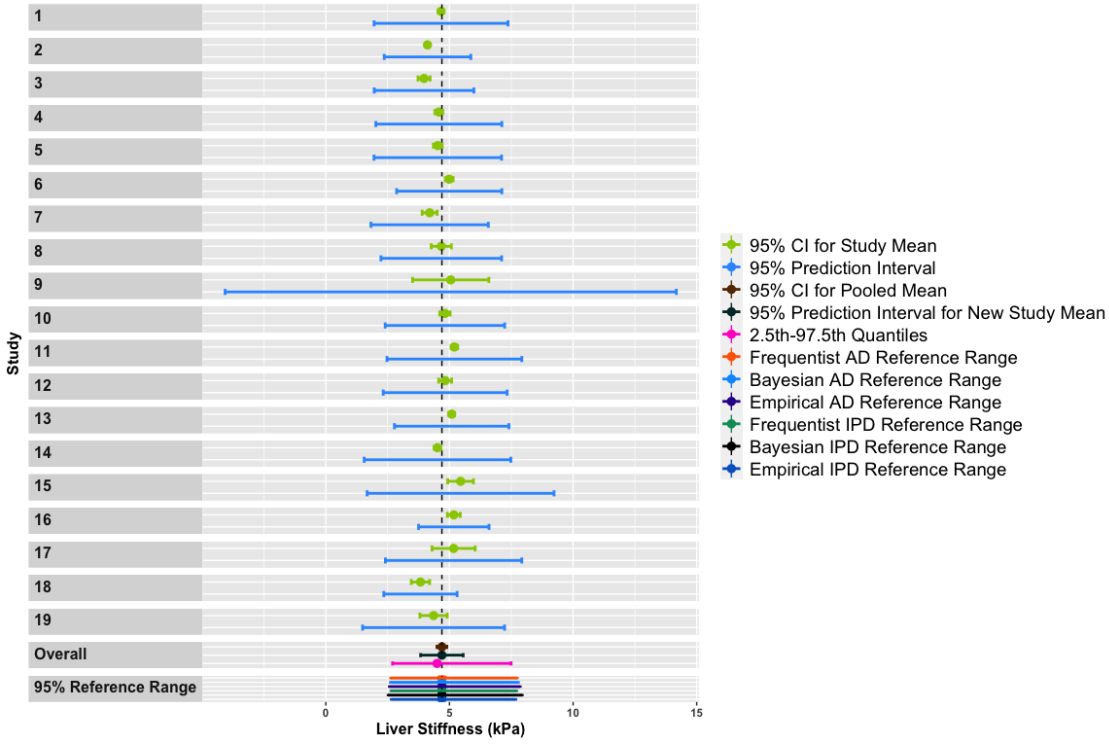
We can see that these quantiles are similar to the estimated reference ranges, with the reference ranges being slightly wider in order to accommodate the possibility of more extreme results from other (unobserved) studies.

We present a forest plot showing the study means (95% CI's), 95% prediction intervals for a new individual from each study population, the pooled mean (95% CI), 95% prediction interval for a new study mean, observed quantiles, and estimated reference ranges:

```
# 95% prediction interval for new study mean# Based on
# results of LMM
N = length(unique(healthy2$studyID))
freq.SEmu = coef(summary(freq.ipd[[1]]))[2]
lmer.tau2 = get_variance_random(freq.ipd[[1]])
predict.lower.ipd = pooled.mean.ipd - qt(0.975, N - 2) * sqrt(lmer.tau2 +
    freq.SEmu^2)
predict.upper.ipd = pooled.mean.ipd + qt(0.975, N - 2) * sqrt(lmer.tau2 +
    freq.SEmu^2)
predict.results = c("Overall", pooled.mean.ipd, NA, nrow(healthy2),
    NA, NA, predict.lower.ipd, predict.upper.ipd)

# add 2.5th, 50th, and 97.5th quantiles to plot
quant.results = c("Overall", quants[2], NA, nrow(healthy2), NA,
    NA, quants[1], quants[3])

# 95% prediction intervals for individuals in each study
predict.ind.results = list()
for (i in 1:19) {
    lower.predict.ind = summary.ipd$mean[i] - qt(0.975, summary.ipd$n[i] -
        1) * sqrt(1 + 1/summary.ipd$n[i]) * summary.ipd$sd[i]
    upper.predict.ind = summary.ipd$mean[i] + qt(0.975, summary.ipd$n[i] -
        1) * sqrt(1 + 1/summary.ipd$n[i]) * summary.ipd$sd[i]
    predict.ind.results[[i]] = c(i, NA, NA, summary.ipd$n[i],
        summary.ipd$mean[i], NA, lower.predict.ind, upper.predict.ind)
}

predict.ind = do.call("rbind", predict.ind.results)

# reference ranges
ref = c("95% Reference Range", pooled.mean.ipd, NA, nrow(healthy2),
    NA, NA)
refrange.results = matrix(c(predict.results, quant.results, ref,
    freq.range2, ref, bayes.range2, ref, emp.range, ref, freq.ipd.range,
    ref, bayes.ipd.range, ref, emp.range.ipd), nrow = 8, byrow = T)
colnames(predict.ind) = names(forest1)
colnames(refrange.results) = names(forest1)
forest4 = rbind(forest3[1:19, c(1:2, 4, 7, 8)], predict.ind[,
    c(1:2, 4, 7, 8)], forest3[20, c(1:2, 4, 7, 8)], refrange.results[,
```

```
    c(1:2, 4, 7, 8)], use.names = F)[1:47, ]
forest4$IntervalType = factor(c(rep("95% CI for Study Mean",
    19), rep("95% Prediction Interval", 19), "95% CI for Pooled Mean",
    "95% Prediction Interval for New Study Mean", "2.5th-97.5th Quantiles",
    "Frequentist AD Reference Range", "Bayesian AD Reference Range",
    "Empirical AD Reference Range", "Frequentist IPD Reference Range",
    "Bayesian IPD Reference Range", "Empirical IPD Reference Range"),
    levels = rev(c("95% CI for Study Mean", "95% Prediction Interval",
        "95% CI for Pooled Mean", "95% Prediction Interval for New Study Mean
",
        "2.5th-97.5th Quantiles", "Frequentist AD Reference Range",
        "Bayesian AD Reference Range", "Empirical AD Reference Range",
        "Frequentist IPD Reference Range", "Bayesian IPD Reference Range",
        "Empirical IPD Reference Range")))

#forest plot with ref range results
#forest plot
p5 = ggplot(forest4,
           aes(x = IntervalType, y =
                 as.numeric(as.character(mean)),
               ymin = as.numeric(as.character(Lower.mean)),
               ymax = as.numeric(as.character(Upper.mean))))+
  geom_pointrange(aes(col=IntervalType))+
  geom_hline(yintercept = pooled.mean.ipd, linetype=2)+
  xlab('Study')+ ylab("Liver Stiffness (kPa)")+
  geom_errorbar(aes(ymin=as.numeric(as.character(Lower.mean)),
                    ymax=as.numeric(as.character(Upper.mean)),
                    color = IntervalType, fill = IntervalType),
                    width=0.5,cex=1)+
  facet_wrap(~s,strip.position="left",nrow=28,scales = "free_y") +
  theme(plot.title=element_text(size=16,face="bold"),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_text(face="bold"),
        axis.title=element_text(size=12,face="bold"),
        legend.title = element_blank(),
        legend.text = element_text(size = 14),
        strip.text.y.left = element_text(hjust=0,vjust = 1,
                                         angle=0,face="bold", size=12))+
 coord_flip() + guides(fill = guide_legend(reverse = T)) +
  guides(color = guide_legend(reverse = T)) + scale_color_manual(position = "
right",values = c("#0066CC", "#000000", "#009966", "#330099", "#0099FF", "#FF
6600", "#FF00CC", "#003333", "#663300", "#3399FF", "#99CC00")) + scale_fill_m
anual(position = "right", values = c("#0066CC", "#000000", "#009966", "#33009
9", "#0099FF", "#FF6600", "#FF00CC", "#003333", "#663300", "#3399FF", "#99CC0
0"))
p5
```

Web Figure 3. 95% confidence interval for each study mean, 95% frequentist prediction interval for a new individual's liver stiffness by study, 95% confidence interval for the pooled mean, 95% prediction interval for a new study mean, 2.5th and 97.5th quantiles of pooled data, and estimated 95% reference range.



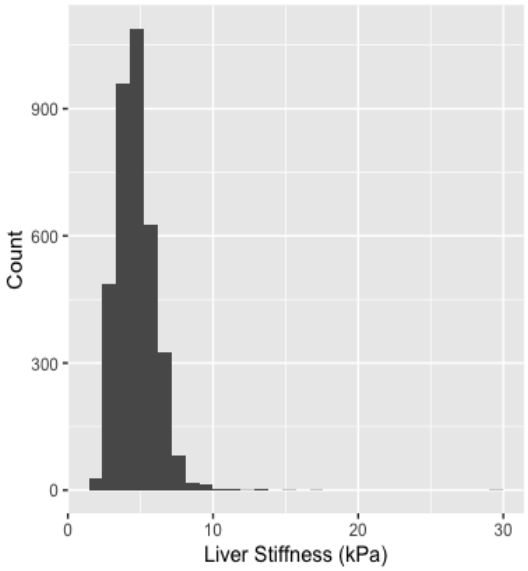
Further Investigation of Modeling Assumptions with IPD

Unlike when only aggregate data are available, the within study and overall normality assumptions can be assessed using histograms.

We first plot the histograms without log transformation and show that the observed data within and across studies looks skewed:

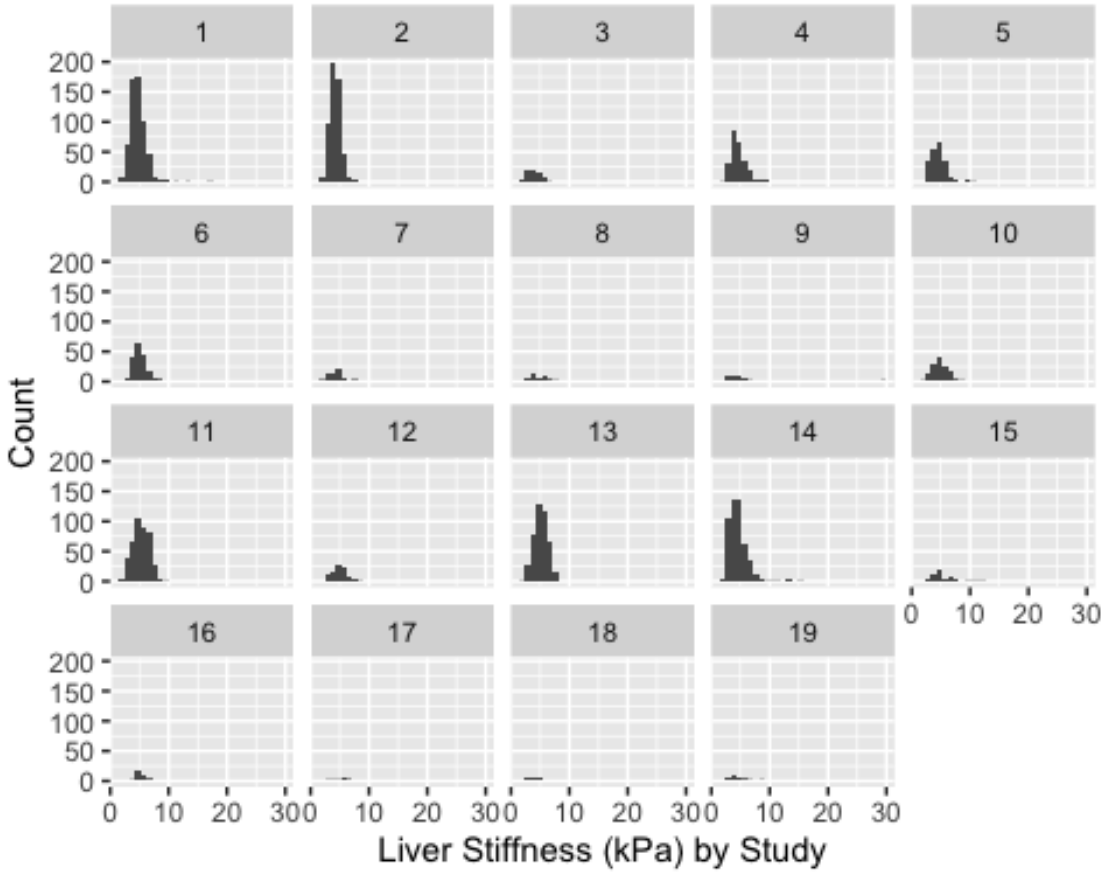
```
h = ggplot(data = healthy2, aes(x = stiff_measure)) + geom_histogram() +
    xlab("Liver Stiffness (kPa)") + ylab("Count")
h
```

Web Figure 4. Histogram of liver stiffness pooled across studies without log-transformation




```
# ggsave('LogHist.pdf', height=4, width=4, dpi = 600)

h2 = ggplot(data = healthy2, aes(x = stiff_measure)) + geom_histogram() +
    xlab("Liver Stiffness (kPa) by Study") + ylab("Count") +
    facet_wrap(~s)
h2
```

Web Figure 5. Histograms of liver stiffness by study without log-transformation

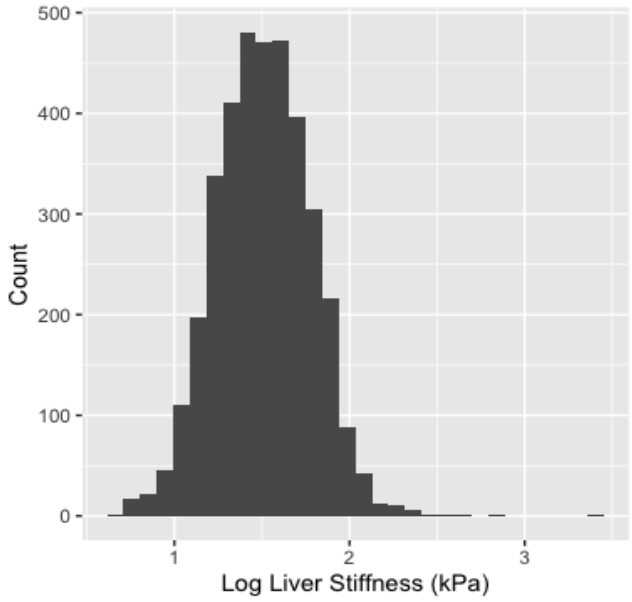



```
# ggsave('LogHistStudies.pdf', height=8, width=8, dpi = 600)
```

We now show that the normality assumptions appear more reasonable for the log-transformed data:

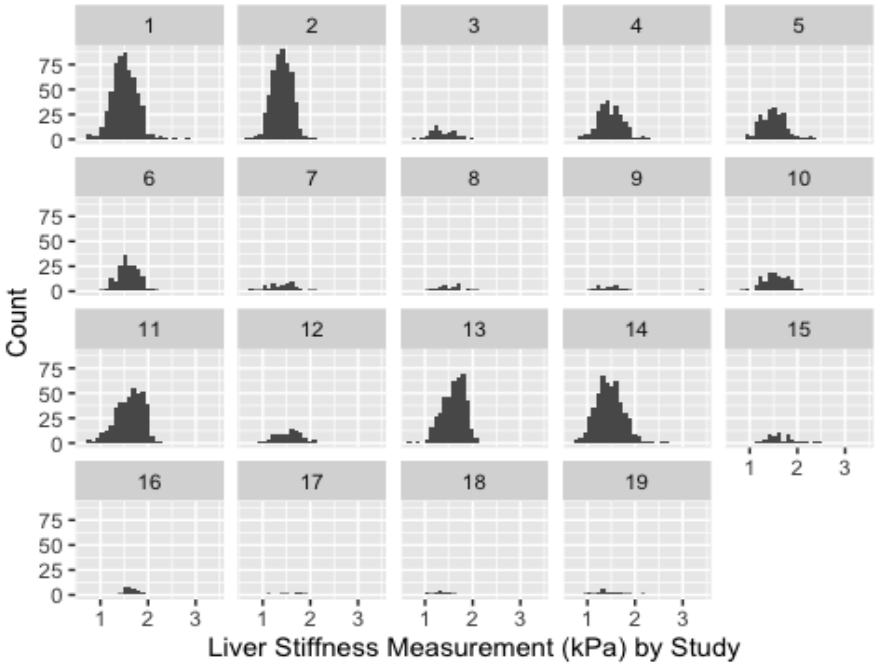
```
h3 = ggplot(data = healthy2, aes(x = log.lsm)) + geom_histogram() +
    xlab("Log Liver Stiffness (kPa)") + ylab("Count")
h3
```

Web Figure 6. Histogram of pooled log-transformed liver stiffness measurements across studies




```
h4 = ggplot(data = healthy2, aes(x = log.lsm)) + geom_histogram() +
    xlab("Liver Stiffness Measurement (kPa) by Study") + ylab("Count") +
    facet_wrap(~s)
h4
```

Web Figure 7. Histograms of log-transformed liver stiffness measurement by study

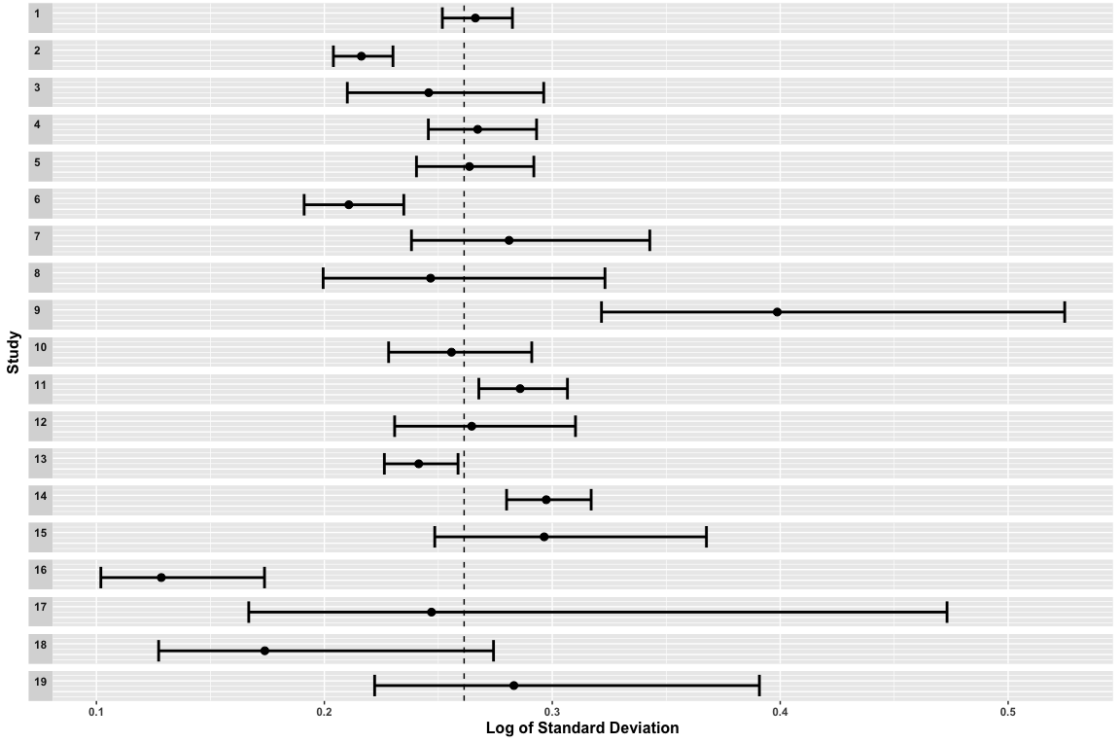


We can also assess the equal within-study variances assumption when directly log-transforming the liver stiffness measurements rather than relying on the aggregate data approximation:

```
## confidence intervals for standard deviations
lower.sd = sqrt(summary.ipd$logsd^2 * (summary.ipd$n - 1)/qchisq(0.025,
    summary.ipd$n - 1, lower.tail = F))
upper.sd = sqrt(summary.ipd$logsd^2 * (summary.ipd$n - 1)/qchisq(0.025,
    summary.ipd$n - 1, lower.tail = T))

sd.dat = as.data.frame(cbind(summary.ipd$s, summary.ipd$logsd,
    lower.sd, upper.sd))
names(sd.dat) = c("s", "logsd", "lower.sd", "upper.sd")
pooled.sd = sqrt(sum((summary.ipd$n - 1) * summary.ipd$logsd^2)/sum(summary.i
pd$n -
    1))

## forest plot
p6 = ggplot(sd.dat,
            aes(x = logsd,
                xmin = lower.sd,
                xmax = upper.sd))+
  geom_hline(yintercept = pooled.sd, linetype=2)+
  xlab('Study')+ ylab("Log of Standard Deviation")+
  facet_wrap(~s,strip.position="left",nrow=20) +
    geom_errorbar(aes(y = logsd,
                      ymin= lower.sd,
                      ymax= upper.sd),
```

```
                width=1,cex=1)+
  geom_point(aes(y = sd.dat$logsd), size = 2.5)+
  theme(plot.title=element_text(size=16,face="bold"),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_text(face="bold"),
        axis.title=element_text(size=12,face="bold"),
        legend.title = element_blank(),
        strip.text.y.left = element_text(hjust=0,vjust = 1,angle=0,face="bold
"))+
  coord_flip() + guides(fill = guide_legend(reverse = TRUE))  +
  guides(color = guide_legend(reverse = TRUE))
p6
```

Web Figure 8. Forest plot of study standard deviations (95% CI's) with log-transformation based on IPD



We see similar results as when we used the aggregate data approximation for the log transformation, though the results from Study 16 are slightly less extreme. We will again conduct a sensitivity analysis for the IPD reference ranges where we remove Studies 9 and 16 and compare the estimated reference ranges to the previous results.

Sensitivity Analysis for Equal Within-Study Standard Deviations Assumption (IPD)

```
## Remove studies 9 and 16 and re-estimate IPD reference
## ranges
healthy3 = subset(healthy2, !(s %in% c(9, 16)))

# Frequentist IPD
freq.ipd.sa = FreqIPD(log(healthy3$stiff_measure), healthy3$s)

# Bayesian IPD
bayes.ipd.sa = BayesLMM(data = healthy3, studyID = "s", outcome = "log.lsm",
    iterations = 1e+05)

# Empirical IPD
mean.ov.sa = mean(healthy3$log.lsm)
sd.ov.sa = sd(healthy3$log.lsm)
lower.ov.sa = qnorm(0.025, mean = mean.ov.sa, sd = sd.ov.sa)
upper.ov.sa = qnorm(0.975, mean = mean.ov.sa, sd = sd.ov.sa)
emp.range.ipd.sa = c(lower.ov.sa, upper.ov.sa)


tbl4 = exp(rbind(freq.ipd.sa[[2]], c(bayes.ipd.sa[[2]], bayes.ipd.sa[[3]]),
    emp.range.ipd.sa))
colnames(tbl4) = c("Lower", "Upper")
rownames(tbl4) = c("Frequentist IPD", "Bayesian IPD", "Empirical IPD")
knitr::kable(tbl4, caption = "IPD Reference Ranges Removing Studies 9 and 16"
,
    digits = 2)
```

Web Table 5. IPD reference ranges (kPa) removing studies 9 and 16

	Lower	Upper
Frequentist IPD	2.62	7.67
Bayesian IPD	2.60	7.75
Empirical IPD	2.64	7.67

We can see that like with the aggregate data approaches, removing Studies 9 and 16 does not meaningfully change the results.

References

1. DerSimonian R, Laird N. Meta-analysis in clinical trials. *Controlled Clinical Trials*. 1986;7(3):177-188. doi:10.1016/0197-2456(86)90046-2

2. Cornell JE, Mulrow CD, Localio R, et al. Random-Effects Meta-analysis of Inconsistent Effects: A Time for Change. *Ann Intern Med*. 2014;160(4):267-270. doi:10.7326/M13-2886

3. Schwarzer G, Carpenter JR, Rücker G. *Meta-Analysis with R*. Springer International Publishing; 2015. doi:10.1007/978-3-319-21416-0

4. Schwarzer G. meta: An R package for meta-analysis. *R News*. 2007;7(3):40-45.

5. Bates D, Mächler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*. 2015;67(1):1-48. doi:10.18637/jss.v067.i01

6. Conceição LB, Baggio JAO, Mazin SC, Edwards DJ, Santos × TEG. Normative data for human postural vertical: A systematic review and meta-analysis. *PLoS One; San Francisco*. 2018;13(9):e0204122. doi:http://dx.doi.org.ezp2.lib.umn.edu/10.1371/journal.pone.0204122

7. Laird NM, Mosteller F. Some Statistical Methods for Combining Experimental Results. *International Journal of Technology Assessment in Health Care*. 1990;6(01):5-30. doi:10.1017/S0266462300008916

8. Bazerbachi F, Haffar S, Wang Z, et al. Range of Normal Liver Stiffness and Factors Associated With Increased Stiffness Measurements in Apparently Healthy Individuals. *Clinical Gastroenterology and Hepatology*. 2019;17(1):54-64.e1. doi:10.1016/j.cgh.2018.08.069

9. Siegel L, Murad MH, Chu H. Estimating the reference range from a meta-analysis. *Research Synthesis Methods*. 2021;12(2):148-160. doi:10.1002/jrsm.1442