

HVPG TRT and Power Analysis

Contents

Aims	2
Libraries	2
Data	3
Study Names	3
Test-Retest Analysis	3
As if one study	3
Grouped by study	7
Are they different	20
Causes of differences	25
Skewness	25
Percentage Decomp	33
Days elapsed	34
Percentage Alcoholic	35
Multicentre	37
Combined Model	38
Mean Value	41
Figures	47
Power Analysis for a difference	49
True and Apparent Changes	54
Simulation	56
Plotting	59
Required Individuals	59
Analytical solution	60

Power Analysis for Difference in Differences	63
Simulation	63
Plotting	67
Required Individuals	69
Contour Plot	70
ICC Figures	79

Aims

Here, I will analyse the HVPG dataset, evaluate aspects of its test-retest reliability, and perform a power analysis for comparing treatments.

Libraries

```

library(tidyverse)
library(relfeas)
library(readxl)
library(hrbrthemes)
library(extrafont)
library(lme4)
library(lmerTest)
library(effsize)
library(broom)
library(RColorBrewer)
library(knitr)
library(cowplot)
library(progress)
library(kableExtra)
library(perm)
library(ggbeeswarm)
library(psych)
library(permuco)
library(metafor)
library(viridis)
library(pwr)

extrafont::loadfonts(quiet = T)

theme_set(theme_ipsum_rc())
nsims <- 1e4
overwrite <- FALSE

knitr::opts_chunk$set(fig.path = "figures/", dev="png", dpi=600,
                      warning=FALSE, message=FALSE)

set.seed(42)

```

Data

```
trt_tidy <- read_excel("../RawData/TEST_RETEST_FINA_DATABASE_TIDY.xlsx") %>%
  mutate(author = str_match(Description, "(^\\w*)[,2]") %>%
    select(-contains("<"))

trt_wide <- read_excel("../RawData/TEST_RETEST_FINA_DATABASE.xlsx") %>%
  select(New_Description, `Serial number`) %>%
  select(-contains("<"))

trt_studydemog <- read_excel("../RawData/FINAL_AGGREGATED_ICCs-2.xlsx") %>%
  select(Study, Perc_Alc, Perc_DecomP, Days=TIMEDAYS,
         Centre = CENTER,
         n_Patients = NUMBEROFPATIENTs) %>%
  mutate(Centre = ifelse(Centre == 1, yes = "Multi-centre", "Single-centre"))
```

Now let's add the new new description to the trt_tidy sheet

```
trt_tidy <- trt_tidy %>%
  left_join(trt_wide)
```

Study Names

```
studynames <- trt_tidy %>%
  select(Study = New_Description,
         Technique = `Technique - Balloon/ catheter`) %>%
  unique() %>%
  mutate(Technique = ifelse(Technique=="Wedged Catheter",
                            yes = "Wedged Catheter",
                            no = "Balloon-tipped Catheter")) %>%
  mutate(Technique = ifelse(is.na(Technique),
                            yes = "Balloon-tipped Catheter",
                            no = Technique)) %>%
  mutate(Catheter = ifelse(Technique=="Wedged Catheter",
                           yes="Wedged", no="Balloon tip"))
```

Test-Retest Analysis

As if one study

Now, we look at the data as if it were all one study, however we also divide by whether the study contains decompensated patients.

```
trt_all <- trt_tidy %>%
  filter(Description != "Spahr. Octreotide") %>%
  select(-Study) %>%
  rename(Study= New_Description) %>%
```

```

left_join(trt_studydemog) %>%
  mutate(decomp = ifelse(Perc_Decom >0 ,
                        yes="Includes Decompensated",
                        no = "Only Compensated")) %>%
  group_by(decomp) %>%
  nest() %>%
  mutate(trt = map(data, ~relfeas::trt(data = .x,
                                         values='PP',
                                         cases = "Serial number",
                                         rater = 'MEASUREMENT')$tidy)) %>%
  select(-data) %>%
  unnest(trt)

trt_all

## # A tibble: 2 x 14
## # Groups:   decomp [2]
##   decomp           mean     sd     cv   skew kurtosis    icc  icc_l  icc_u    wscv    sdd absvar signvar
##   <chr>          <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Includes Decompensated 17.3   4.83  0.279 0.163 -0.610 0.852 0.813 0.883 0.107  5.16  0.107 -0.0
## 2 Only Compensated      16.4   3.83  0.233 0.339 -0.394 0.837 0.784 0.877 0.0945 4.29  0.108  0.0

saveRDS(trt_all, "../Cluster/trt_all.rds")

kable(trt_all, digits=2)

```

decomp	mean	sd	cv	skew	kurtosis	icc	icc_l	icc_u	wscv	sdd	absvar	signvar
Includes Decompensated	17.32	4.83	0.28	0.16	-0.61	0.85	0.81	0.88	0.11	5.16	0.11	-0.01
Only Compensated	16.40	3.83	0.23	0.34	-0.39	0.84	0.78	0.88	0.09	4.29	0.11	0.01

I'll also create a tidy version of this to complement the individual studies.

```

trt_all_tidy <- trt_all %>%
  mutate(signvar_sd = signvar_sd * mean) %>%
  ungroup() %>%
  select(Patients = decomp,
         Mean = mean, CV = cv,
         WSCV = wscv, ICC = icc,
         SDD = sdd,
         "Change SD" = signvar_sd) %>%
  arrange(desc(Patients))

kable(trt_all_tidy, digits=2)

```

Patients	Mean	CV	WSCV	ICC	SDD	Change SD
Only Compensated	16.40	0.23	0.09	0.84	4.29	2.19
Includes Decompensated	17.32	0.28	0.11	0.85	5.16	2.63

So, overall, we estimate that our smallest detectable difference in an individual is 4.3mmHg for only compensated patients, and 5.2mmHg for studies including decompensated patients

```

trt_all_detailed <- trt_tidy %>%
  filter>Description != "Spahr. Octreotide") %>%
  select(-Study) %>%
  rename(Study= New_Description) %>%
  left_join(trt_studydemog) %>%
  mutate(decomp = ifelse(Perc_Decom >0 ,
                        yes="Includes Decompensated",
                        no = "Only Compensated")) %>%
  filter>Description != "Spahr. Octreotide") %>%
  group_by(decomp) %>%
  nest() %>%
  mutate(trt = map(data, ~relfeas::trt(data = .x,
                                         values='PP',
                                         cases = "Serial number",
                                         rater = 'MEASUREMENT' )))

trt_all_detailed$decomp[1]

```

```
## [1] "Includes Decompensated"
```

```
trt_all_detailed$trt[[1]]$sdd
```

```

## $value
## [1] 5.15761
##
## $lbound
## [1] 4.657455
##
## $ubound
## [1] 5.77906

```

```

trt_all$sdd_l <- trt_all_detailed$trt[[1]]$sdd$lbound
trt_all$sdd_u <- trt_all_detailed$trt[[1]]$sdd$ubound

```

```
trt_all_detailed$decomp[2]
```

```
## [1] "Only Compensated"
```

```
trt_all_detailed$trt[[2]]$sdd
```

```

## $value
## [1] 4.293093
##
## $lbound
## [1] 3.802727
##
## $ubound
## [1] 4.929798

```

```

trt_all$sdd_l[2] <- trt_all_detailed$trt[[2]]$sdd$lbound
trt_all$sdd_u[2] <- trt_all_detailed$trt[[2]]$sdd$ubound

trt_all_detailed$decomp[1]

```

```
## [1] "Includes Compensated"
```

```
trt_all_detailed$trt[[1]]$sddm
```

```

## $value
## [1] 0.2978477
##
## $lbound
## [1] 0.2656889
##
## $ubound
## [1] 0.3344565

```

```

trt_all$sddm <- trt_all_detailed$trt[[1]]$sddm$value*100
trt_all$sddm_l <- trt_all_detailed$trt[[1]]$sddm$lbound*100
trt_all$sddm_u <- trt_all_detailed$trt[[1]]$sddm$ubound*100

```

```
trt_all_detailed$decomp[2]
```

```
## [1] "Only Compensated"
```

```
trt_all_detailed$trt[[2]]$sddm
```

```

## $value
## [1] 0.2618087
##
## $lbound
## [1] 0.228803
##
## $ubound
## [1] 0.3000771

```

```

trt_all$sddm[2] <- trt_all_detailed$trt[[2]]$sddm$value*100
trt_all$sddm_l[2] <- trt_all_detailed$trt[[2]]$sddm$lbound*100
trt_all$sddm_u[2] <- trt_all_detailed$trt[[2]]$sddm$ubound*100

```

Now, let's prepare this as a table for below the forest plot

```

overall_n <- map_dbl(trt_all_detailed$data, nrow)

overall <- trt_all %>%
  ungroup() %>%
  rename(Study = decomp) %>%
  mutate(decomp = "Overall",
        Catheter = "Balloon tip",

```

```

      n = overall_n) %>%
arrange(desc(Study)) %>%
mutate(Study = ifelse(Study=="Includes Decompensated",
                      "Includes Decompensated*",
                      Study))

```

overall

```

## # A tibble: 2 x 22
##   Study           mean     sd    cv skew kurtosis    icc icc_l icc_u   wscv    sdd absvar sign
##   <chr>          <dbl>   <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Only Compensated    16.4   3.83  0.233  0.339   -0.394  0.837  0.784  0.877  0.0945  4.29   0.108  0.0
## 2 Includes Decompensated* 17.3   4.83  0.279  0.163   -0.610  0.852  0.813  0.883  0.107   5.16   0.107 -0.0

```

Grouped by study

```

trt_study <- trt_tidy %>%
  group_by(New_Description) %>%
  nest() %>%
  mutate(outcomes = map(data, ~ trt(
    data=.x, values='PP',
    cases = "Serial number", rater = 'MEASUREMENT' )))

trt_study <- trt_study %>%
  mutate(n = map_dbl(data, nrow))

tidytrt <- map_df(trt_study$outcomes, 'tidy') %>%
  mutate(n = trt_study$n) %>%
  mutate(Study = trt_study$New_Description) %>%
  select(Study, everything()) %>%
  left_join(studynames) %>%
  left_join(trt_studydemog) %>%
  mutate(decomp = ifelse(Perc_Decomp >0 ,
                        yes="Includes Decompensated",
                        no = "Only Compensated")) %>%
  mutate(decomp = fct_inorder(decomp)) %>%
  select(-n_Patients)

knitr::kable(tidytrt, digits = 2)

```

Study	mean	sd	cv	skew	kurtosis	icc	icc_1	icc_u	wscv	sdd	absvar	signvar	s
Abraldes 2008 (D)	20.28	4.43	0.22	0.12	-1.00	0.81	0.62	0.91	0.10	5.43	0.11	-0.03	
Abraldes 2008 (C)	18.47	2.91	0.16	0.28	-1.47	0.81	0.45	0.94	0.07	3.65	0.08	0.02	
Albillos 1995	19.60	3.82	0.20	0.14	-1.69	0.94	0.82	0.98	0.05	2.74	0.05	0.01	
Berzigotti 2010	18.62	2.21	0.12	0.05	-2.35	0.96	0.42	1.00	0.03	1.55	0.04	0.01	
Blei 1987	15.06	4.67	0.31	0.14	-1.53	0.97	0.85	0.99	0.06	2.49	0.06	0.05	
Debernardi 2007	14.59	1.76	0.12	0.49	-0.56	0.64	0.32	0.83	0.07	3.01	0.08	0.05	
Jayakumar 2013	22.72	3.11	0.14	0.18	-0.90	0.62	0.10	0.88	0.09	5.42	0.11	0.00	
Kimer 2017	16.18	4.46	0.28	0.69	0.10	0.63	0.32	0.82	0.17	7.62	0.19	-0.02	
Lebrec 2012	18.04	3.01	0.17	-1.10	-0.06	0.66	0.06	0.92	0.10	5.01	0.13	-0.06	
Merkel 2004	12.39	1.18	0.10	0.12	-0.26	0.87	0.63	0.96	0.04	1.22	0.04	0.02	
Moller 2000	15.38	5.45	0.35	-0.41	-1.51	0.96	0.87	0.99	0.07	3.10	0.07	0.02	
Reverter 2015	15.76	4.48	0.28	0.39	-1.06	0.95	0.90	0.98	0.06	2.76	0.07	-0.03	
Schepke 2001	18.25	3.85	0.21	0.06	-1.18	0.69	0.42	0.85	0.12	6.01	0.12	0.00	
Spahr 2007	17.69	2.96	0.17	-0.07	-1.32	0.14	-0.47	0.67	0.16	7.64	0.21	-0.07	
Schwarzer 2017	20.50	4.86	0.24	-0.06	-1.50	0.94	0.83	0.98	0.06	3.39	0.06	0.02	
Pomier 1987	15.42	5.50	0.36	1.45	0.52	0.95	0.54	0.99	0.09	3.71	0.09	0.10	
Hidaka 2011	14.75	3.71	0.25	0.27	-0.87	0.79	0.59	0.90	0.12	4.80	0.13	-0.06	
McCormick 1992	17.44	4.40	0.25	-0.18	-0.32	0.94	0.87	0.97	0.06	3.04	0.08	0.02	
Pozzi 2005	16.11	3.61	0.22	0.10	-1.14	0.78	0.41	0.93	0.11	4.89	0.14	0.08	
Garcia-Tsao 2020 (C)	16.63	4.00	0.24	0.24	-0.50	0.82	0.72	0.88	0.10	4.75	0.13	0.01	
Garcia-Tsao 2020 (D)	16.32	4.73	0.29	-0.76	-0.15	0.26	-0.19	0.72	0.27	12.17	0.35	-0.25	
Fukada 2014	17.28	4.15	0.24	0.32	-0.83	0.93	0.77	0.98	0.07	3.19	0.07	0.02	

Now let's make a table for the paper with the relevant things.

```

tidytrt_table <- tidytrt %>%
  mutate(signvar_sd = signvar_sd * mean) %>%
  select(decomp,
         Study,
         n,
         "Decompensated (%)" = Perc_Decomp,
         "Alcoholic (%)" = Perc_Alc,
         "Mean Days Elapsed" = Days,
         Mean = mean, CV = cv,
         WSCV = wscv, ICC = icc,
         SDD = sdd,
         "Change SD" = signvar_sd) %>%
  arrange(desc(decomp), Study)

decomp_change <- head(
  which(
    tidytrt_table$decomp == tail(tidytrt_table$decomp, 1)), 1 )

tidytrt_kable <- knitr::kable(tidytrt_table[,-1], digits=2) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows(head(tidytrt_table$decomp, 1), 1, decomp_change-1) %>%
  pack_rows(tail(tidytrt_table$decomp, 1), decomp_change, nrow(tidytrt_table))

tidytrt_kable

```

```
# save_kable(tidytrt_kable, file = "figures/tidytrt_kable.jpg")
```

ICC

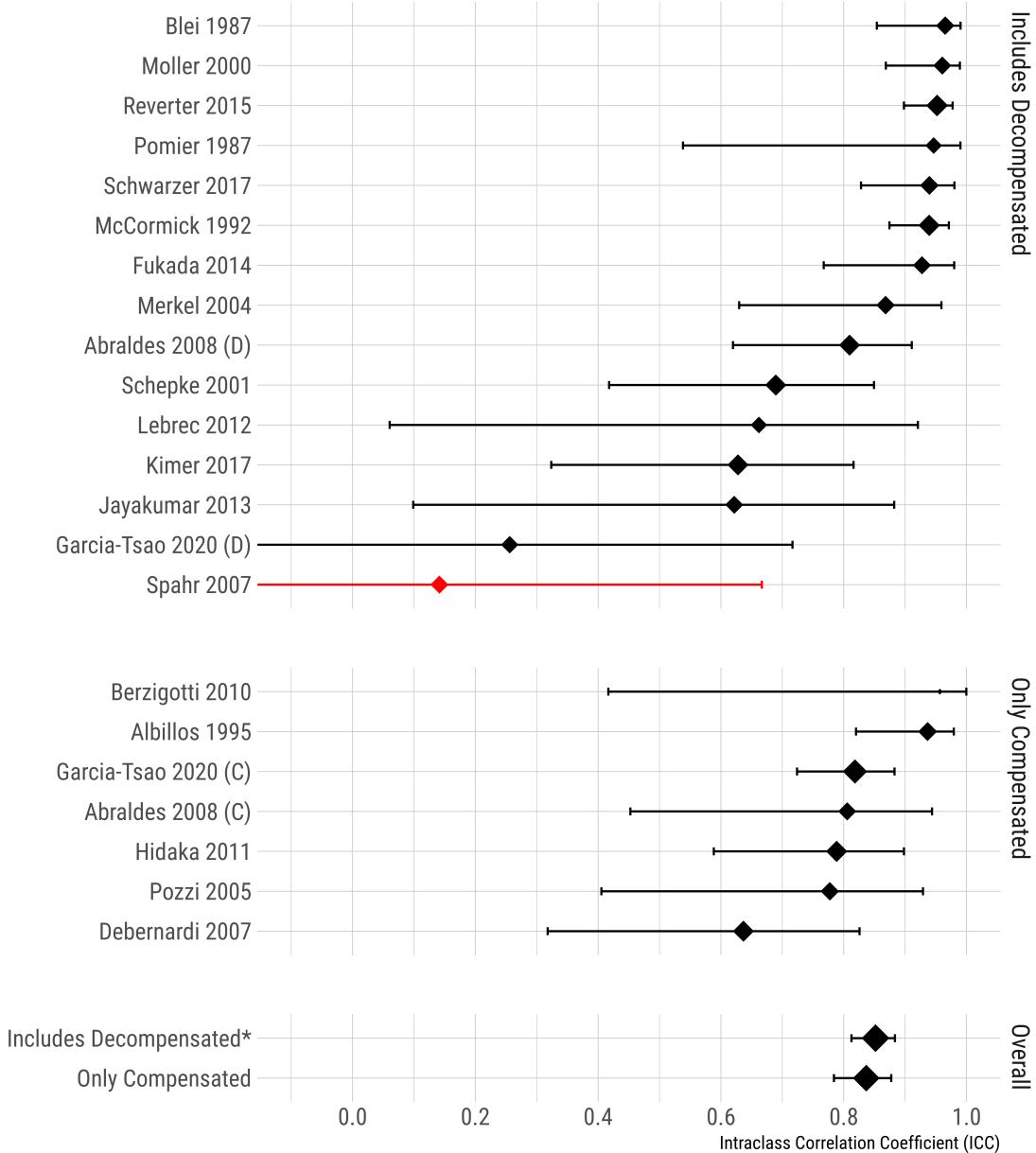
Let's have a look at the ICC as a measure of between-subject differentiability (i.e. reliability).

```
icc_out <- select(tidytrt, Study, icc, icc_l, icc_u, decomp, n) %>%
  left_join(studynames) %>%
  mutate(decomp = factor(decomp, levels=c(
    "Includes Decompensated", "Only Compensated"))) %>%
  arrange(decomp, icc) %>%
  bind_rows(overall) %>%
  mutate(Study = fct_inorder(Study))

ICCs <- ggplot(icc_out,aes(x=icc,y=Study,
                           colour=Catheter)) +
  #geom_rect(aes(xmin=0.8, xmax=1, ymin=-Inf, ymax=Inf),
  #          alpha = .1, fill="grey", colour="grey") +
  facet_grid(decomp~, scales="free", space="free") +
  geom_point(aes(size=log(n)), shape=18) +
  scale_x_continuous(breaks = seq(0, 1, by=0.2))+ 
  geom_errorbarh(aes(xmax = icc_u, xmin = icc_l), height = 0.2) +
  #geom_vline(xintercept = 1, linetype = "longdash") +
  labs(y="", x="Intraclass Correlation Coefficient (ICC)") +
  theme(text = element_text(size=20)) +
  ggtitle("ICC (95% CI)") +
  scale_colour_manual(values = c("black", "red")) +
  coord_cartesian(xlim = c(-0.1, 1)) +
  #ylim = c(0,12)) +
  guides(size = FALSE) +
  NULL

ICCs + theme(legend.position = "none")
```

ICC (95% CI)



This analysis constitutes a mega-analysis: we have all the original data and the overall estimates are performed using the original data estimates. However, in order to assess the study heterogeneity, I will run a classical meta-analysis. I will perform a Fisher's z transformation on the ICC values, perform a classic meta-analysis, and assess the heterogeneity.

```
dat <- icc_out %>%
  filter(Study != "Spahr 2007") %>%
  filter(decomp != "Overall") %>%
  filter(n > 4) %>%
```

```

escalc(measure="ZCOR", ri=icc, ni=n, data=., slab=Study)

res <- rma(yi, vi, data=dat)
res

##
## Random-Effects Model (k = 20; tau^2 estimator: REML)
##
## tau^2 (estimated amount of total heterogeneity): 0.1838 (SE = 0.0768)
## tau (square root of estimated tau^2 value):       0.4287
## I^2 (total heterogeneity / total variability):   81.58%
## H^2 (total variability / sampling variability):  5.43
##
## Test for Heterogeneity:
## Q(df = 19) = 91.7982, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval   ci.lb   ci.ub
##   1.2674  0.1091  11.6164  <.0001  1.0536  1.4813  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

SDD

Absolute Let's have a look at the study-by-study SDD, but in raw units

```

sdd_out <- map_df(trt_study$outcomes, "sdd") %>%
  mutate(Study = trt_study$New_Description) %>%
  rename(sdd=value, sdd_l = lbound, sdd_u = ubound) %>%
  left_join(studynames) %>%
  left_join(select(tidytrt, Study, decomp, n)) %>%
  left_join(trt_studydemog) %>%
  mutate(decomp = ifelse(Perc_Decom >0 ,
                        yes="Includes Decompensated",
                        no = "Only Compensated")) %>%
  mutate(decomp = factor(decomp, levels=c(
    "Includes Decompensated", "Only Compensated"))) %>%
  arrange(decomp, desc(sdd)) %>%
  bind_rows(overall) %>%
  mutate(Study = fct_inorder(Study))

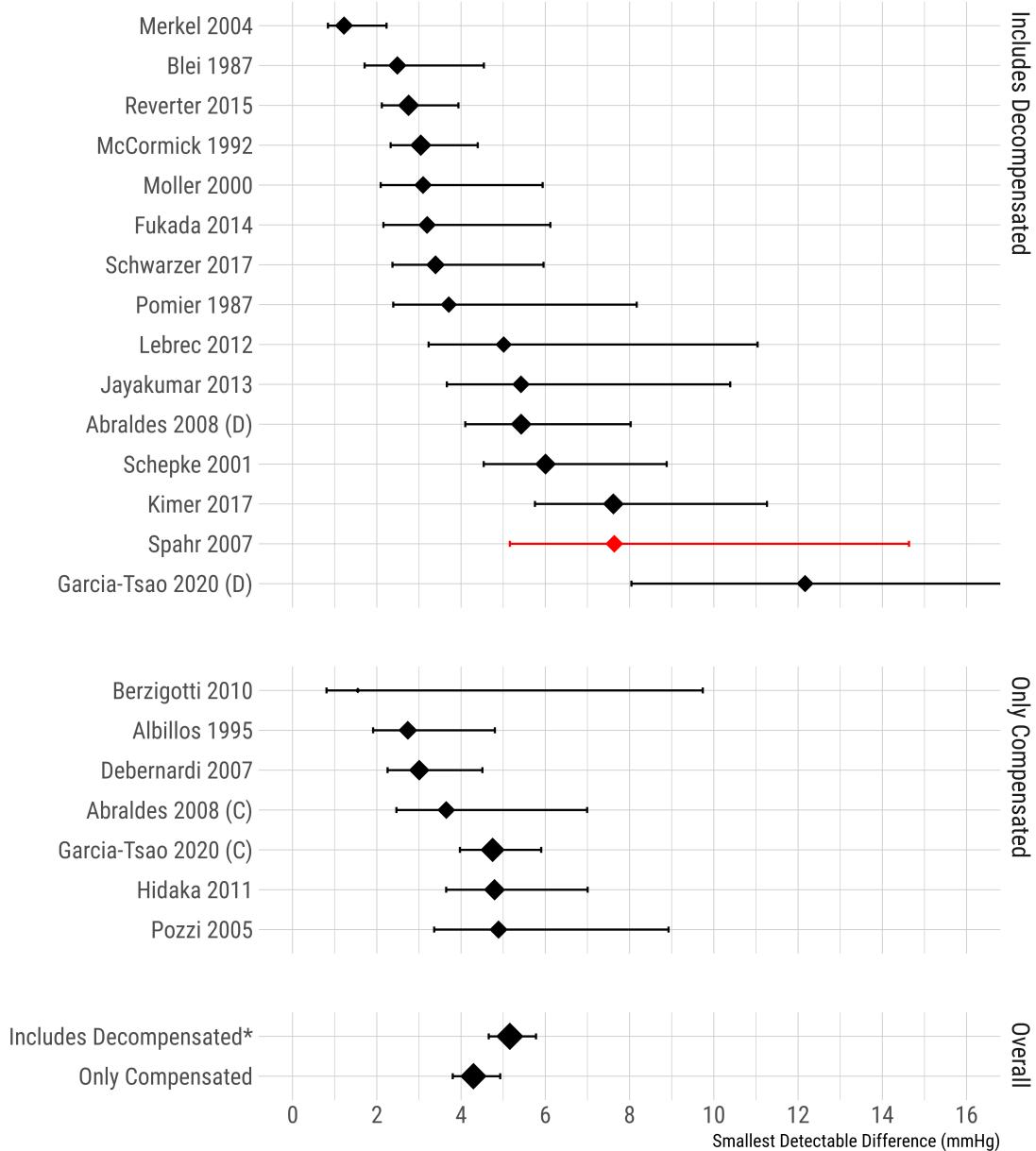
SDDs <- ggplot(sdd_out,aes(x=sdd,y=Study,
                           colour=Catheter)) +
  facet_grid(decomp~, scales="free", space="free") +
  geom_point(aes(size=log(n)), shape=18) +
  scale_x_continuous(breaks = seq(0, 20, by=2))+ 
  geom_errorbarh(aes(xmax = sdd_u, xmin = sdd_l), height = 0.15) +
  #geom_vline(xintercept = 1, linetype = "longdash") +
  labs(y="", x="Smallest Detectable Difference (mmHg)") +
  theme(text = element_text(size=20)) +

```

```
ggtitle("SDD (95% CI)") +
scale_colour_manual(values = c("black", "red")) +
guides(colour=FALSE, shape=FALSE) +
#scale_shape_manual(values = c(18, 19)) +
coord_cartesian(xlim=c(0,16)) +
guides(size = FALSE) +
NULL

SDDs  #+ annotate("rect", xmin = 0.75, xmax = 1, ymin="Spahr 2007", ymax="Blei 1987", alpha = .2, fill=
```

SDD (95% CI)



And here we run the meta-analysis again to assess heterogeneity. We can't use the SDD because it has asymmetric confidence intervals, and it cannot be Fisher's z transformed. So we can run a meta-analysis based on the average absolute variation to test the heterogeneity.

```
dat <- trt_study %>%
  mutate(
    mean_abs_var = map_dbl(outcomes, ~mean(sqrt(.x$absvars))),
    se_abs_var = map_dbl(outcomes, ~sd(sqrt(.x$absvars)) / sqrt(length(.x$absvars)))
  ) %>%
```

```

filter(New_Description != "Spahr 2007") %>%
ungroup() %>%
mutate(
  yi = mean_abs_var,
  vi = se_abs_var^2) %>%
select(-data, -outcomes) %>%
escalc(measure="MN", yi=yi, vi=vi, ni=n, data=., slab=New_Description)

transf.sqrt <- function (xi, ...)
{
  zi <- sqrt(xi)
  zi[xi < 0] <- 0
  return(c(zi))
}

res <- rma(yi, vi, data=dat)
res

##
## Random-Effects Model (k = 21; tau^2 estimator: REML)
##
## tau^2 (estimated amount of total heterogeneity): 0.0016 (SE = 0.0010)
## tau (square root of estimated tau^2 value):      0.0405
## I^2 (total heterogeneity / total variability):   54.64%
## H^2 (total variability / sampling variability):  2.20
##
## Test for Heterogeneity:
## Q(df = 20) = 46.5104, p-val = 0.0007
##
## Model Results:
##
## estimate      se     zval    pval   ci.lb   ci.ub
##   0.2572  0.0128  20.1232 <.0001  0.2322  0.2823  ***
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Percentage Now the study-by-study SDD, but in percentages

```

sddm_out <- map_df(trt_study$outcomes, "sddm") %>%
  mutate(Study = trt_study$New_Description) %>%
  rename(sddm=value, sddm_l = lbound, sddm_u = ubound) %>%
  mutate(sddm=100*sddm, sddm_l = 100*sddm_l, sddm_u = 100*sddm_u) %>%
  left_join(studynames) %%
  left_join(select(tidytrt, Study, decomp, n)) %>%
  left_join(trt_studydemog) %>%
  mutate(decomp = ifelse(Perc_Decom >0 ,
                        yes="Includes Decompensated",
                        no = "Only Compensated")) %>%
  mutate(decomp = factor(decomp, levels=c(
    "Includes Decompensated", "Only Compensated")))) %>%
  arrange(decomp, desc(sddm)) %>%

```

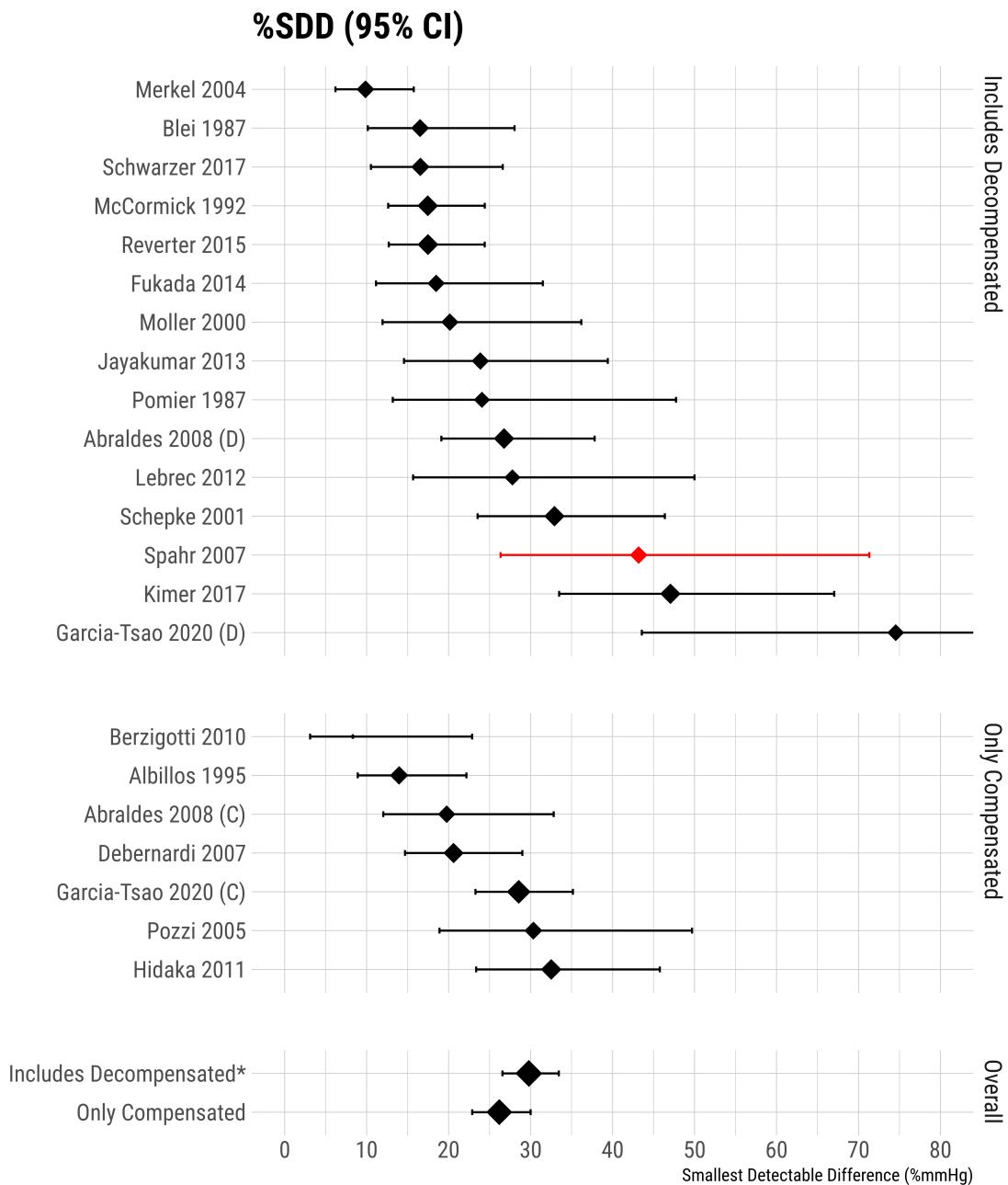
```

bind_rows(overall) %>%
  mutate(Study = fct_inorder(Study))

SDDms <- ggplot(sddm_out,aes(x=sddm,y=Study,
                                colour=Catheter)) +
  facet_grid(decomp~., scales="free", space="free") +
  geom_point(aes(size=log(n)), shape=18) +
  scale_x_continuous(breaks = seq(0, 80, by=10))+ 
  geom_errorbarh(aes(xmax = sddm_u, xmin = sddm_l), height = 0.15) +
  #geom_vline(xintercept = 1, linetype = "longdash") +
  labs(y="", x="Smallest Detectable Difference (%mmHg)") +
  theme(text = element_text(size=20)) +
  ggtitle("%SDD (95% CI)") +
  scale_colour_manual(values = c("black", "red")) +
  guides(colour=FALSE, shape=FALSE) +
  #scale_shape_manual(values = c(18, 19)) +
  coord_cartesian(xlim=c(0,80)) +
  guides(size = FALSE) +
  NULL

SDDms  #+ annotate("rect", xmin = 0.75, xmax = 1, ymin="Spahr 2007", ymax="Blei 1987", alpha = .2, fill

```



And here we run the meta-analysis to assess heterogeneity. We can't use the SDD% because it has asymmetric confidence intervals, and it cannot be Fisher's z transformed. So we can run a meta-analysis based on the average absolute percentage variation to test the heterogeneity.

```
dat <- trt_study %>%
  mutate(
    mean_abs_var = map_dbl(outcomes, ~mean(sqrt(.x$absvars / .x$means))),
    se_abs_var = map_dbl(outcomes, ~sd(sqrt(.x$absvars / .x$means)) /
      sqrt(length(.x$absvars)))
```

```

) %>%
filter(New_Description != "Spahr 2007") %>%
ungroup() %>%
mutate(
  yi = mean_abs_var,
  vi = se_abs_var^2)

res <- rma(yi, vi, data=dat)
res

## 
## Random-Effects Model (k = 21; tau^2 estimator: REML)
## 
## tau^2 (estimated amount of total heterogeneity): 0.0001 (SE = 0.0001)
## tau (square root of estimated tau^2 value):      0.0107
## I^2 (total heterogeneity / total variability):   60.82%
## H^2 (total variability / sampling variability):  2.55
## 
## Test for Heterogeneity:
## Q(df = 20) = 60.4345, p-val < .0001
## 
## Model Results:
## 
## estimate      se     zval    pval   ci.lb   ci.ub
## 0.0628  0.0033  18.8998  <.0001  0.0563  0.0694  ***
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

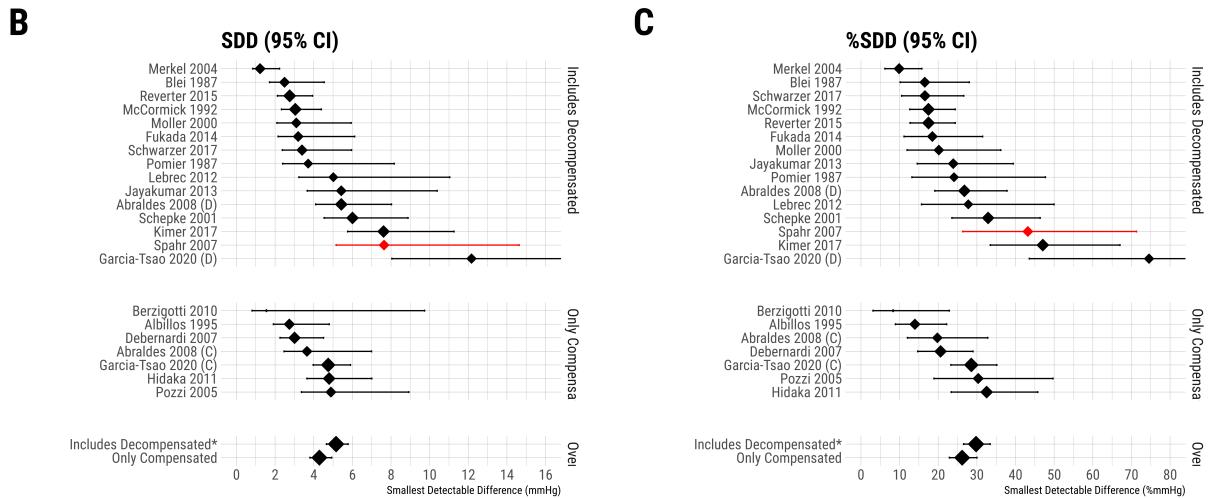
Figures together

```

SDD_together <- cowplot:::plot_grid(SDDs, SDDms, ncol = 2,
                                     labels = c("B", "C"),
                                     label_size = 30)

SDD_together

```



Alternatively, all in a row

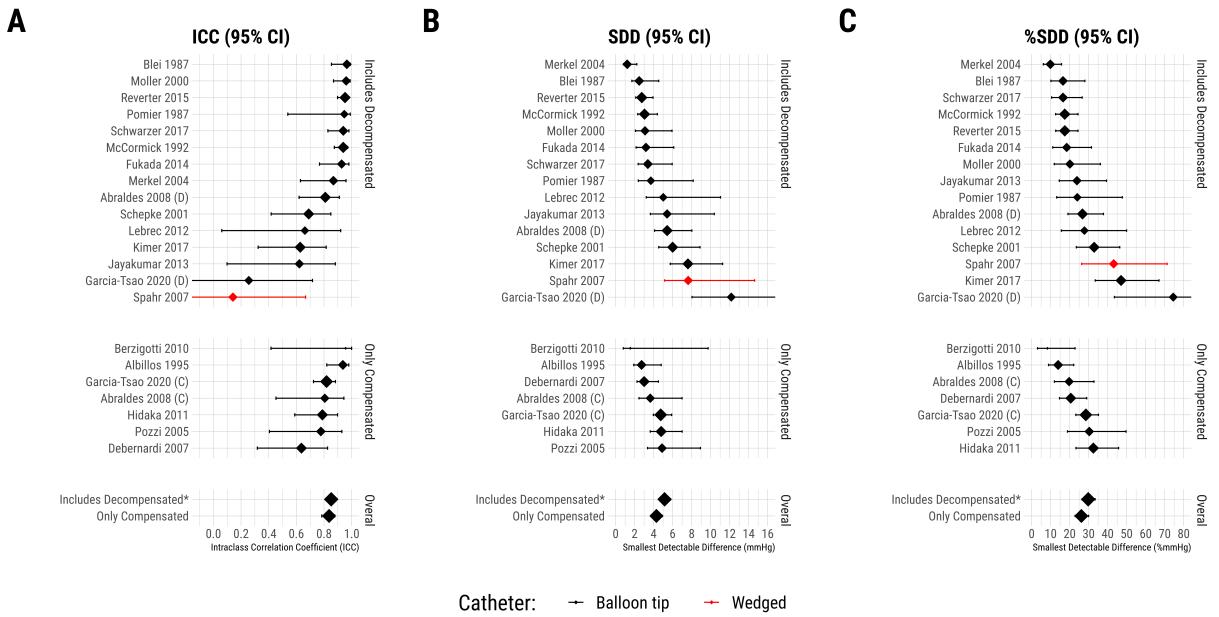
```
ICCs_row <- ICCs + guides(colour=FALSE, shape=FALSE)

forest_legend <- get_legend(ICCs +
  theme(legend.position="bottom") +
  scale_shape_discrete("Patients:") +
  scale_colour_manual("Catheter:",
    values=c("black", "red")))

forest_row_legendless <- plot_grid(ICCs_row, SDDs, SDDms,
  nrow=1, labels=c("A", "B", "C"),
  label_size=30)

forest_row <- plot_grid(forest_row_legendless, forest_legend,
  nrow=2, rel_heights = c(15,1))

forest_row
```



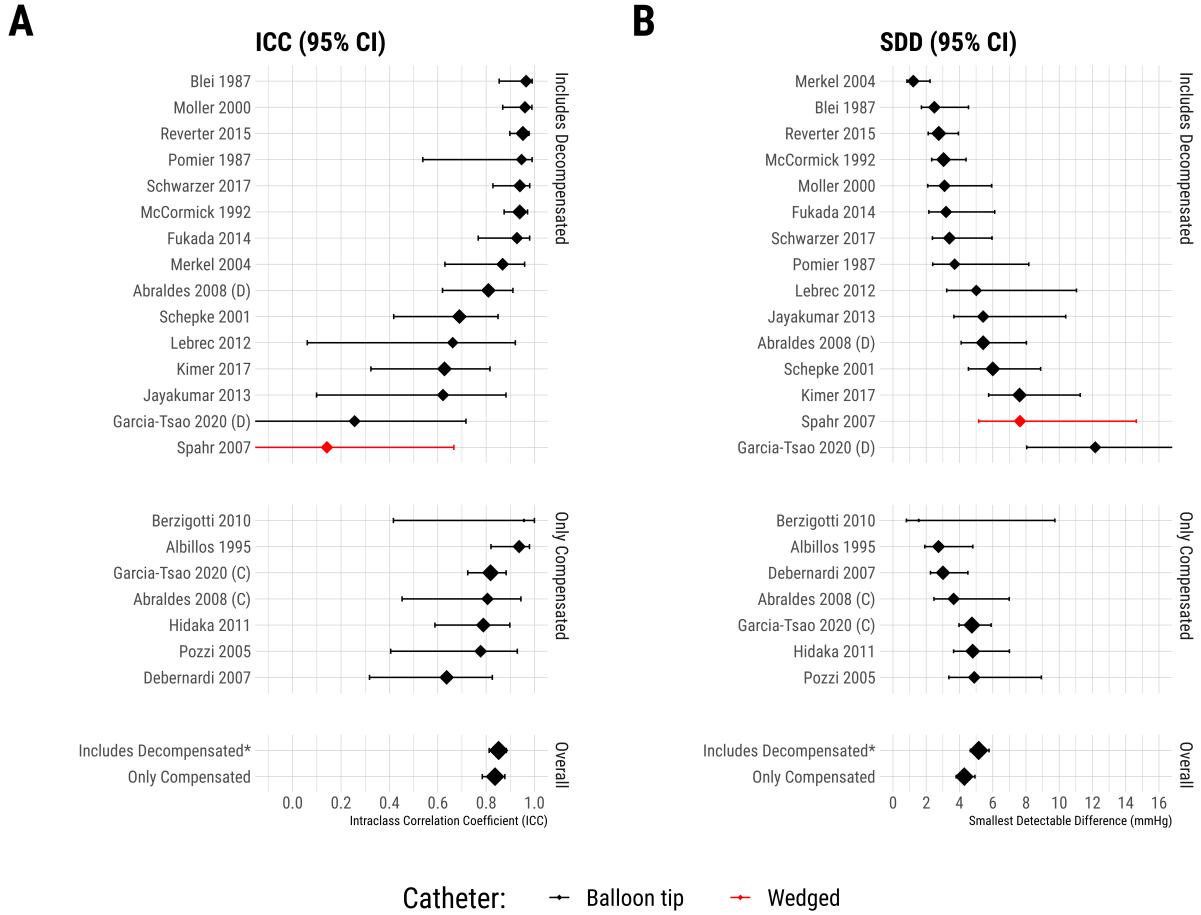
```
ggsave(forest_row, filename = "figures/forest_three_row.png", width = 16, height = 8)
```

Just ICC and SDD

```
forest_row2_legendless <- plot_grid(ICC_s_row, SDDs,
                                    nrow=1, labels=c("A", "B"),
                                    label_size=30)

forest_row2 <- plot_grid(forest_row2_legendless, forest_legend,
                         nrow=2, rel_heights = c(15,1))

forest_row2
```



Are they different

To test the difference between the groups, we'll bootstrap the difference. That means we take 1000 random samples from each group of the same size with replacement, and calculate the difference to get the null distribution. Then we compare the difference we see to the bootstrap distribution.

```
trt_compare <- trt_tidy %>%
  filter(Description != "Spahr. Octreotide") %>%
  select(-Study) %>%
  rename(Study= New_Description) %>%
  left_join(trt_studydemog) %>%
  mutate(decomp = ifelse(Perc_Decom > 0 ,
                        yes="Includes Decompensated",
                        no = "Only Compensated")) %>%
  group_by(decomp) %>%
  nest()
```

ICC

```
bootstrap_icc_single <- function(data) {
```

```

sample_ids <- unique(data$`Serial number`)
ids <- sample(sample_ids, length(sample_ids), replace=TRUE)

data_nested <- data %>%
  select(`Serial number`, MEASUREMENT, PP) %>%
  nest_by(`Serial number`)

boot_sample <- tibble(
  `Serial number` = ids
) %>%
  left_join(data_nested, by="Serial number") %>%
  select(-`Serial number`) %>%
  mutate(boot_serial = 1:n()) %>%
  unnest(data) %>%
  rlefeas::trt_widify("PP", "boot_serial", "MEASUREMENT")

boot_icc <- suppressMessages(
  suppressWarnings(
    psych::ICC(boot_sample[,c(2,3)])$results$ICC[2] )
  return(boot_icc)
}

bootstrap_icc <- function(n, data, colname=NULL) {

  out <- tibble(
    n = 1:n
  ) %>%
    mutate(boot_icc = map_dbl(n, ~bootstrap_icc_single(data)))

  if(!is.null(colname)) {
    colnames(out)[2] <- colname
  }

  return(out)
}

trt_compare_iccvals <- trt_compare %>%
  mutate(boot = map2(data, decomp, ~bootstrap_icc(10000, .x))) %>%
  select(-data) %>%
  unnest(boot)

trt_compare_icc <- trt_compare_iccvals %>%
  spread(decomp, boot_icc) %>%
  mutate(dif = `Only Compensated` - `Includes Decompenated`)

quantile(trt_compare_icc$dif, c(0.025, 0.5, 0.975)) # 95% CI

##          2.5%      50%      97.5%
## -0.09604300 -0.01736313  0.06649128

1-sum(trt_compare_icc$dif > 0)/10000 # one-sided p value

## [1] 0.6645

```

Not significant

SDD

```
bootstrap_sdd_single <- function(data) {  
  
  sample_ids <- unique(data$`Serial number`)  
  ids <- sample(sample_ids, length(sample_ids), replace=TRUE)  
  
  data_nested <- data %>%  
    select(`Serial number`, MEASUREMENT, PP) %>%  
    nest_by(`Serial number`)  
  
  boot_sample <- tibble(  
    `Serial number` = ids  
) %>%  
    left_join(data_nested, by="Serial number") %>%  
    select(-`Serial number`) %>%  
    mutate(boot_serial = 1:n()) %>%  
    unnest(data) %>%  
    relfeas::trt_widify("PP", "boot_serial", "MEASUREMENT")  
  
  boot_sdd <- suppressMessages(  
    suppressWarnings(  
      agRee::agree.sdd(as.matrix(boot_sample[,c(2,3)]))$value ) )  
  return(boot_sdd)  
}  
  
bootstrap_sdd_single <- function(data) {  
  
  sample_ids <- unique(data$`Serial number`)  
  ids <- sample(sample_ids, length(sample_ids), replace=TRUE)  
  
  data_nested <- data %>%  
    select(`Serial number`, MEASUREMENT, PP) %>%  
    nest_by(`Serial number`)  
  
  boot_sample <- tibble(  
    `Serial number` = ids  
) %>%  
    left_join(data_nested, by="Serial number") %>%  
    select(-`Serial number`) %>%  
    mutate(boot_serial = 1:n()) %>%  
    unnest(data) %>%  
    relfeas::trt_widify("PP", "boot_serial", "MEASUREMENT")  
  
  boot_sdd <- suppressMessages(  
    suppressWarnings(  
      agRee::agree.sdd(as.matrix(boot_sample[,c(2,3)]))$value ) )  
  return(boot_sdd)  
}
```

```

bootstrap_sdd <- function(n, data, colname=NULL) {

  out <- tibble(
    n = 1:n
  ) %>%
    mutate(boot_sdd = map_dbl(n, ~bootstrap_sdd_single(data)))

  if(!is.null(colname)) {
    colnames(out)[2] <- colname
  }

  return(out)
}

# bootstrap_sdd_single(trt_compare$data[[1]])
# bootstrap_sdd(100, trt_compare$data[[1]], "test")

```

And now we test

```

trt_compare_sddvals <- trt_compare %>%
  mutate(boot = map2(data, decomp, ~bootstrap_sdd(10000, .x))) %>%
  select(-data) %>%
  unnest(boot)

trt_compare_sdd <- trt_compare_sddvals %>%
  spread(decomp, boot_sdd) %>%
  mutate(dif = `Includes Decompensated` - `Only Compensated`)

quantile(trt_compare_sdd$dif, c(0.025, 0.5, 0.975)) # 95% CI

##          2.5%      50%     97.5%
## -0.2412147  0.8292606  2.0480528

1-sum(trt_compare_sdd$dif > 0)/10000 # one-sided p value

```

```
## [1] 0.0678
```

SDDM

Percentage SDD

```

bootstrap_sddm_single <- function(data) {

  sample_ids <- unique(data$`Serial number`)
  ids <- sample(sample_ids, length(sample_ids), replace=TRUE)

  data_nested <- data %>%
    select(`Serial number`, MEASUREMENT, PP) %>%
    nest_by(`Serial number`)
}

```

```

boot_sample <- tibble(
  `Serial number` = ids
) %>%
  left_join(data_nested, by="Serial number") %>%
  select(-`Serial number`) %>%
  mutate(boot_serial = 1:n()) %>%
  unnest(data) %>%
  relfeas::trt_widify("PP", "boot_serial", "MEASUREMENT")

boot_sddm <- suppressMessages(
  suppressWarnings(
    agRee::agree.sddm(as.matrix(boot_sample[,c(2,3)])$value) )
  return(boot_sddm)
}

bootstrap_sddm <- function(n, data, colname=NULL) {

  out <- tibble(
    n = 1:n
  ) %>%
    mutate(boot_sddm = map_dbl(n, ~bootstrap_sddm_single(data)))

  if(!is.null(colname)) {
    colnames(out)[2] <- colname
  }

  return(out)
}

# bootstrap_sddm_single(trt_compare$data[[1]])
# bootstrap_sddm(100, trt_compare$data[[1]], "test")

```

And now we test

```

trt_compare_sddmvals <- trt_compare %>%
  mutate(boot = map2(data, decomp, ~bootstrap_sddm(10000, .x))) %>%
  select(-data) %>%
  unnest(boot)

trt_compare_sddm <- trt_compare_sddmvals %>%
  spread(decomp, boot_sddm) %>%
  mutate(dif = `Includes Decompenated` - `Only Compensated`)

quantile(trt_compare_sddm$dif, c(0.025, 0.5, 0.975)) # 95% CI

##          2.5%      50%     97.5%
## -0.03102582  0.03489179  0.10583268

1-sum(trt_compare_sddm$dif > 0)/10000 # one-sided p value

```

```
## [1] 0.1557
```

Causes of differences

Here, we perform an exploratory analysis of which factors may contribute to differences

```
trt_wide <- trt_tidy %>%
  select(-Study) %>%
  rename(Study= New_Description) %>%
  spread(MEASUREMENT, PP) %>%
  rename(Meas1 = `1`,
        Meas2 = `2`) %>%
  mutate(change = Meas2 - Meas1,
        abschange=abs(change),
        meanval = (Meas1 + Meas2)/2) %>%
  left_join(studynames) %>%
  left_join(trt_studydemog) %>%
  mutate(Description = as.factor(Description)) %>%
  filter(Description != "Spahr. Octreotide") %>%
  mutate(decomp = ifelse(Perc_Decom == 0,
                        "Only Compensated",
                        "Includes Decompensated"))

trt_wide_c <- trt_wide %>%
  filter(Perc_Decom == 0)

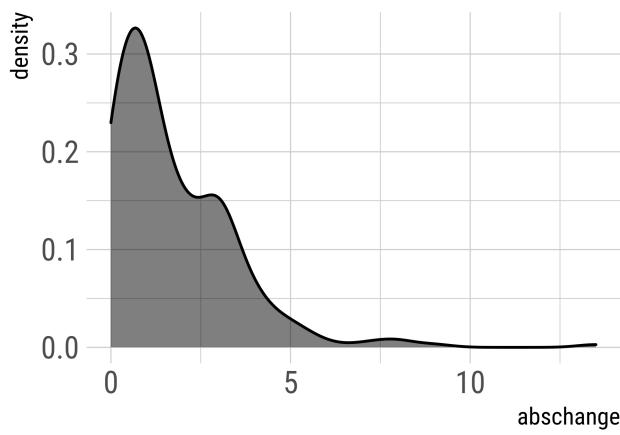
trt_wide_dc <- trt_wide %>%
  filter(Perc_Decom > 0)
```

Skewness

Absolute

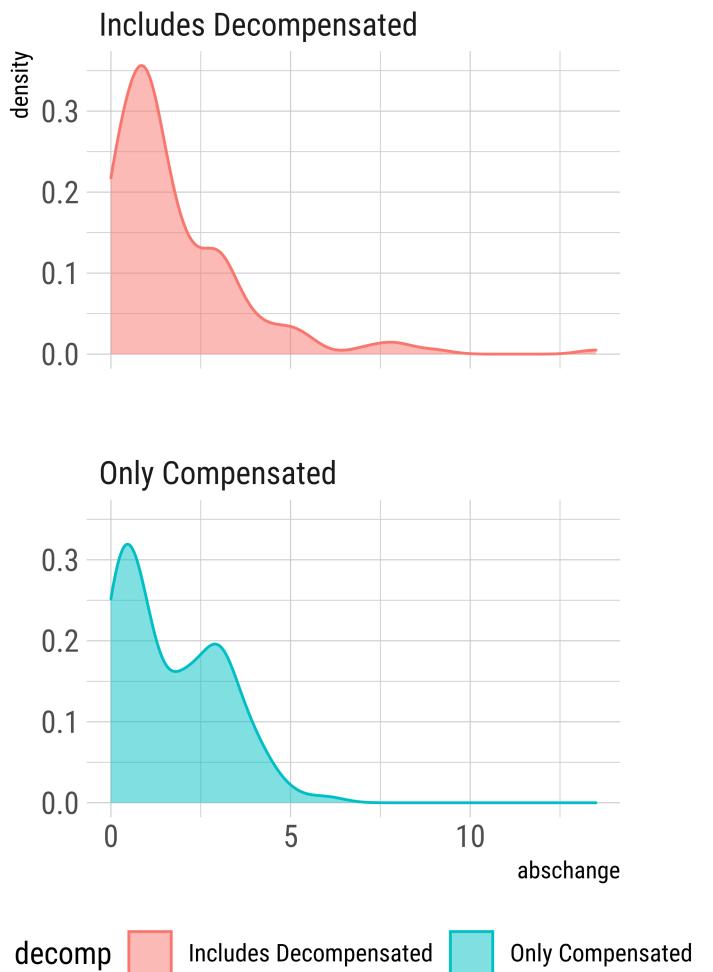
Let's just visualise our distribution.

```
ggplot(trt_wide, aes(x=abschange)) +
  geom_density(fill="black",alpha=0.5)
```



And between groups

```
ggplot(trt_wide, aes(x=abschange)) +  
  geom_density(aes(colour=decomp, fill=decomp), alpha=0.5) +  
  facet_wrap(decomp~, nrow=2) +  
  theme(legend.position="bottom")
```



And let's get some values for that too

```
psych::describe(trt_wide$abschange)
```

```
##      vars   n  mean    sd median trimmed  mad min  max range skew kurtosis   se
## X1     1 281 1.74 1.74      1     1.5 1.48   0 13.5 13.5 2.14     8.37 0.1
```

```
psych::describeBy(trt_wide$abschange, group=trt_wide$decomp)
```

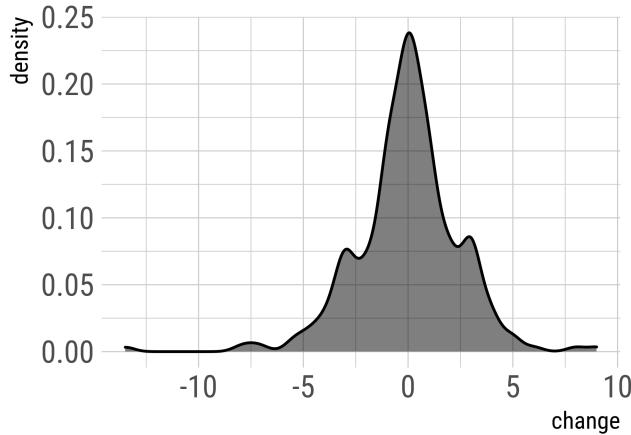
```
##
##  Descriptive statistics by group
##  group: Includes Decompensated
##      vars   n  mean    sd median trimmed  mad min  max range skew kurtosis   se
## X1     1 166 1.79 1.94      1     1.46 1.11   0 13.5 13.5 2.45     8.89 0.15
## -----
##  group: Only Compensated
```

```
##      vars     n   mean    sd median trimmed   mad min max range skew kurtosis    se
## X1      1 115 1.67 1.42      1     1.56 1.48    0    6      6 0.57    -0.64 0.13
```

Signed

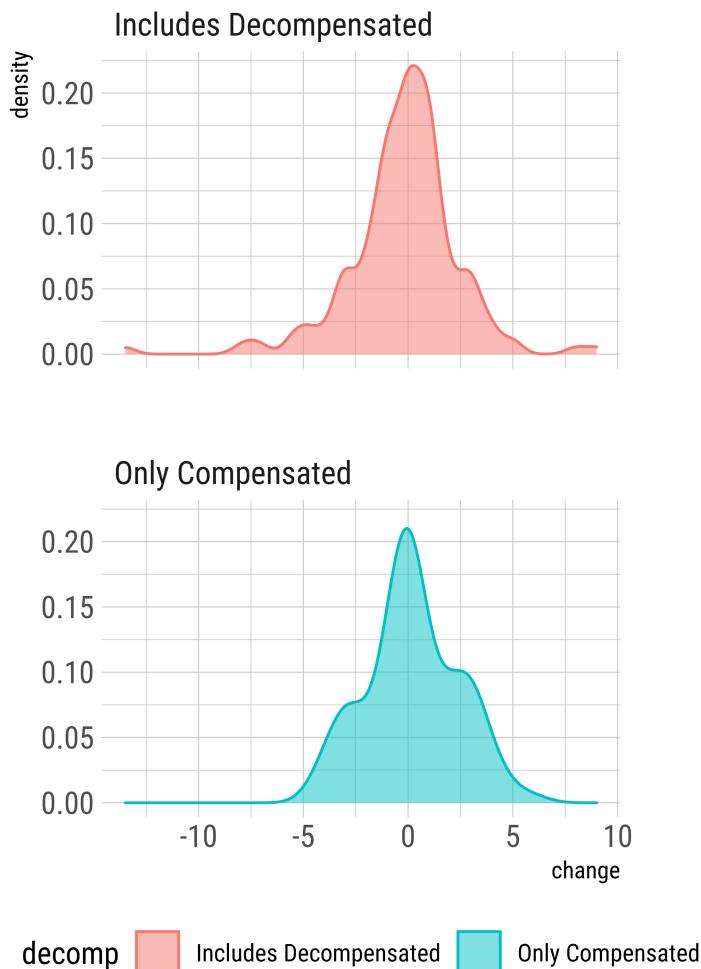
Let's just visualise our overall distribution.

```
ggplot(trt_wide, aes(x=change)) +
  geom_density(fill="black",alpha=0.5)
```



... and by group

```
ggplot(trt_wide, aes(x=change)) +
  geom_density(aes(colour=decomp, fill=decomp),alpha=0.5) +
  facet_wrap(decomp~., nrow=2) +
  theme(legend.position="bottom")
```



For signed differences, let's assess whether they differ from zero.

```
summary(lmer(change ~ 1 + (1|Study), data=trt_wide))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: change ~ 1 + (1 | Study)
##   Data: trt_wide
##
## REML criterion at convergence: 1304.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max 
## -5.1470 -0.4712 -0.0170  0.5292  3.7968 
##
## Random effects:
##   Groups    Name        Variance Std.Dev. 
##   Study     (Intercept) 0.2793   0.5285 
##   Residual             0.0000   0.0000 
```

```

##  Residual           5.8542   2.4195
## Number of obs: 281, groups: Study, 21
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -0.03655   0.19379 9.51195 -0.189   0.854

summary(lmer(change ~ 1 + (1|Study), data=trt_wide_c))

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: change ~ 1 + (1 | Study)
## Data: trt_wide_c
##
## REML criterion at convergence: 506.5
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.1852 -0.5692 -0.1075  0.7360  2.6627
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## Study    (Intercept) 0.1414   0.3761
## Residual            4.6911   2.1659
## Number of obs: 115, groups: Study, 7
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  0.2522    0.2659 3.4807  0.948   0.404

summary(lmer(change ~ 1 + (1|Study), data=trt_wide_dc))

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [lmerModLmerTest]
## Formula: change ~ 1 + (1 | Study)
## Data: trt_wide_dc
##
## REML criterion at convergence: 792
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -4.7455 -0.4164  0.0247  0.4964  3.5999
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## Study    (Intercept) 0.3612   0.601
## Residual            6.6449   2.578
## Number of obs: 166, groups: Study, 14
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -0.200     0.263  5.868  -0.76   0.476

```

Not significantly different from zero in either the combined sample or each group.

And some values

```

psych::describe(trt_wide$change)

##    vars   n  mean   sd median trimmed  mad   min max range skew kurtosis   se
## X1     1 281 -0.03 2.46      0    0.03 1.48 -13.5   9  22.5 -0.56    3.68 0.15

psych::describeBy(trt_wide$change, group=trt_wide$decomp)

##
## Descriptive statistics by group
## group: Includes Decompensated
##    vars   n  mean   sd median trimmed  mad   min max range skew kurtosis   se
## X1     1 166 -0.2 2.63      0   -0.1 1.48 -13.5   9  22.5 -0.75    4.57 0.2
## -----
## group: Only Compensated
##    vars   n  mean   sd median trimmed  mad   min max range skew kurtosis   se
## X1     1 115  0.22 2.19      0    0.22 1.48 -4.5    6  10.5  0.08   -0.4 0.2

```

Figures

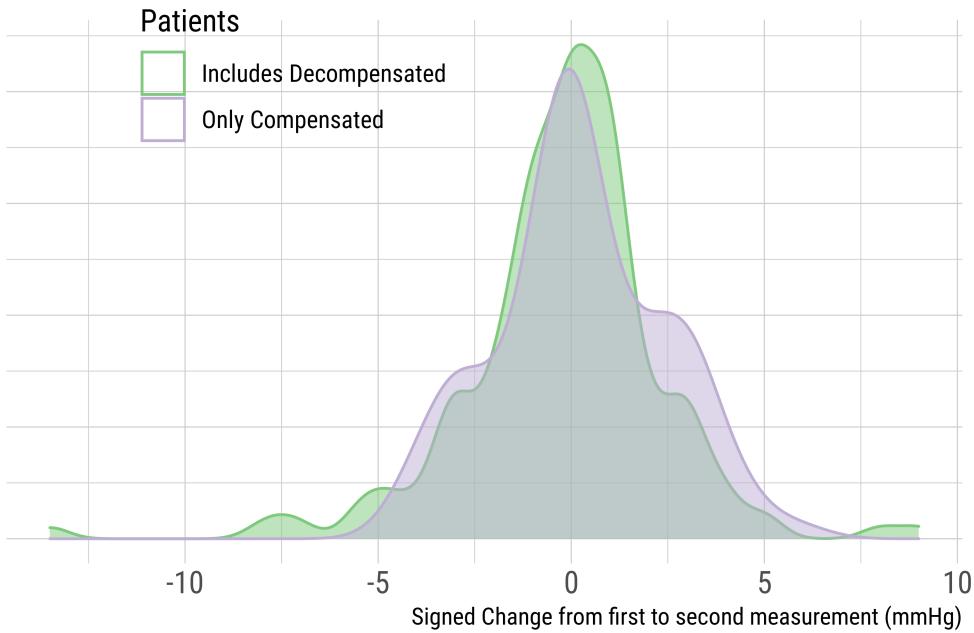
Let's make some combined figures for the paper

```

sign_change_distr <- trt_wide %>%
  mutate(decomp = as.factor(decomp)) %>%
  ggplot(aes(x=change, colour=decomp, fill=decomp, group=decomp)) +
  geom_density(aes(colour=decomp, fill=decomp), alpha=0.5) +
  theme(legend.position=c(0.3, 0.9)) +
  scale_color_brewer("Patients", type = "qual", palette = 1) +
  scale_fill_brewer("Patients", type = "qual", palette = 1) +
  guides(fill=FALSE) +
  labs(y = "",
       x = "Signed Change from first to second measurement (mmHg)") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y=element_blank())

sign_change_distr

```

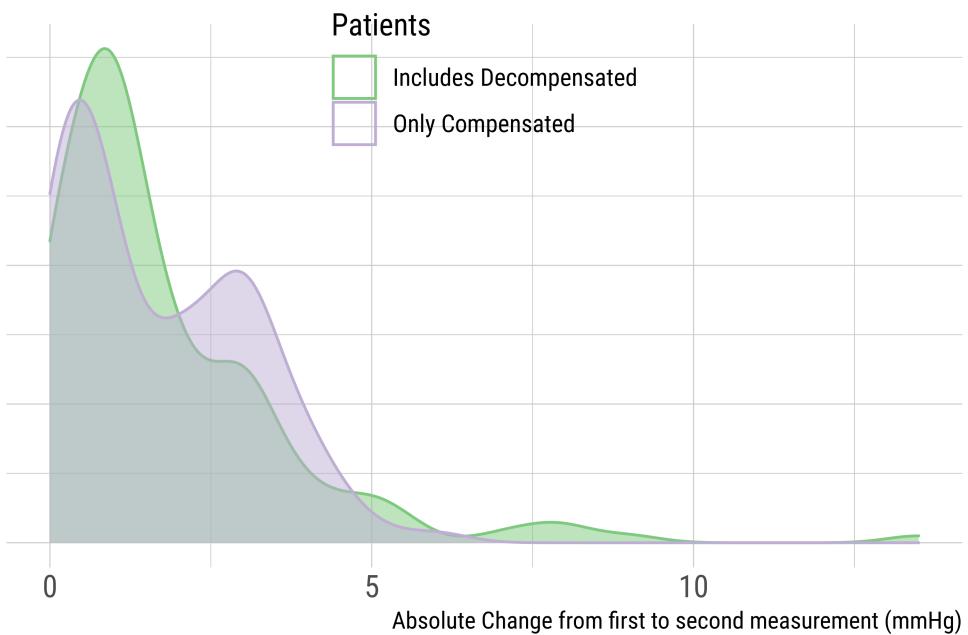


```

abs_change_distr <- trt_wide %>%
  mutate(decomp = as.factor(decomp)) %>%
  ggplot(aes(x=abschange, colour=decomp, fill=decomp, group=decomp)) +
  geom_density(aes(colour=decomp, fill=decomp), alpha=0.5) +
  theme(legend.position=c(0.5, 0.9)) +
  scale_color_brewer("Patients", type = "qual", palette = 1) +
  scale_fill_brewer("Patients", type = "qual", palette = 1) +
  guides(fill=FALSE) +
  labs(y = "",
       x = "Absolute Change from first to second measurement (mmHg)") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y=element_blank())

```

abs_change_distr



Percentage Decomp

```

permTREND(formula=abschange ~ Perc_Decompa, data=trt_wide,
           method="exact.mc")

##
##  Exact Permutation Test Estimated by Monte Carlo
##
## data:  x and y
## p-value = 0.032
## alternative hypothesis: true correlation of x and y is not equal to 0
## sample estimates:
## correlation of x and y
##                  0.1262568
##
## p-value estimated from 999 Monte Carlo replications
## 99 percent confidence interval on p-value:
##  0.01384991 0.05601331

permuco::lmpperm(abschange ~ Perc_Decompa, data=trt_wide)

## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
##          Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t) permutation Pr(>t) per
## (Intercept) 1.499246   0.153912    9.741           1.725e-19
## Perc_Decompa 0.005031   0.002367    2.126           3.439e-02               0.9812            0.019

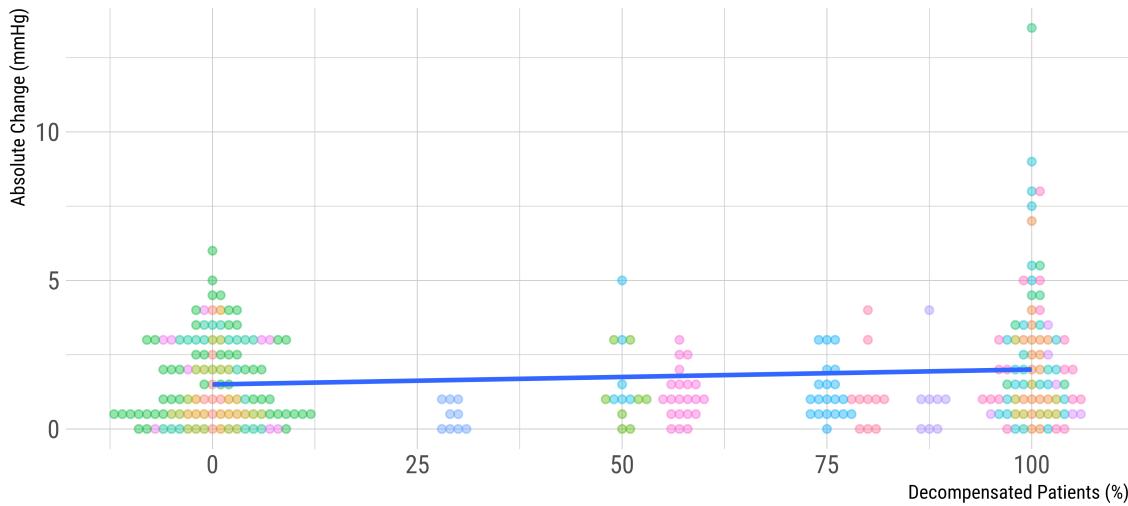
```

```

decomp_plot <- ggplot(trt_wide, aes(x=Perc_Decom, y=abschange)) +
  geom_beeswarm(aes(colour=as.factor(Study),
                     group=as.factor(Perc_Decom)),
                 alpha=0.4, cex=1) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se = FALSE) +
  labs(x="Decompensated Patients (%)",
       y="Absolute Change (mmHg)")

decomp_plot

```



Days elapsed

```
permuco::lmperrm(abschange ~ Days + Perc_Decom, data=trt_wide)
```

```

## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
##           Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t) permutation Pr(>t)
## (Intercept) 1.6031874 0.1994382 8.0385      2.624e-14
## Days        -0.0006133 0.0007477 -0.8202      4.128e-01      0.2076
## Perc_Decom  0.0041622 0.0025941  1.6045      1.097e-01      0.9524      0.0478

```

Now let's plot, after correcting for the effect of the patient groups.

```

correct_for_decomp <- function(formula) {

  formula <- as.formula(formula)

  coefficients <- coef(permuco::lmperrm(formula,
                                            data=trt_wide))

```

```

predicted <- as.character(formula[2])

after_decomp_corr <- trt_wide[[predicted]] -
  trt_wide[["Perc_Decomps"]]* coefficients[which(names(coefficients)=="Perc_Decomps")]

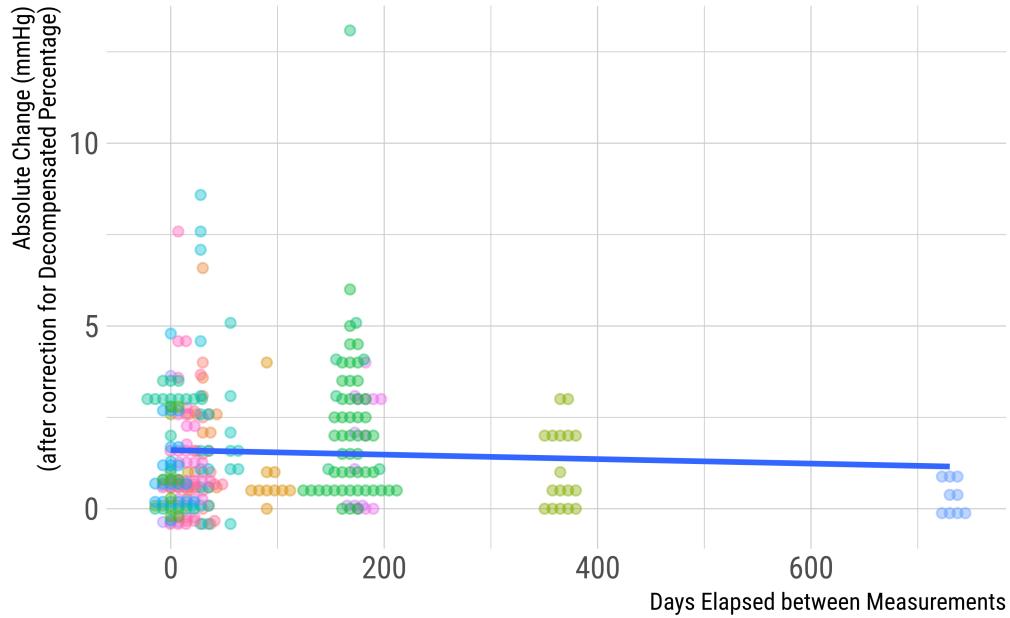
return(after_decomp_corr)

}

days_plot <- trt_wide %>%
  mutate(abschange_dccorr = correct_for_decomp(abschange ~ Days + Perc_Decomps)) %>%
  ggplot(aes(x=Days, y=abschange_dccorr)) +
  geom_beeswarm(aes(colour=Study, group=Study), alpha=0.4, cex=1) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se=FALSE) +
  labs(y="Absolute Change (mmHg)\n(after correction for Decompensated Percentage)",
       x="Days Elapsed between Measurements")

days_plot

```



Percentage Alcoholic

```
permuco::lmperm(abschange ~ Perc_Alc + Perc_Decomps, data=trt_wide)
```

```
## Table of marginal t-test of the betas
```

```

## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
##           Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t) permutation Pr(>t) per
## (Intercept)  1.94758   0.166155  11.721          4.711e-26
## Perc_Alc    -0.02579   0.004560  -5.655         3.869e-08      2e-04           1e+00
## Perc_Decom  0.01881   0.003313   5.677         3.447e-08      1e+00           2e-04

permTREND(formula=abschange ~ Perc_Alc, data=trt_wide_c,
           method="exact.mc")

##
##  Exact Permutation Test Estimated by Monte Carlo
##
## data: x and y
## p-value = 0.01
## alternative hypothesis: true correlation of x and y is not equal to 0
## sample estimates:
## correlation of x and y
##                  -0.2373197
##
## p-value estimated from 999 Monte Carlo replications
## 99 percent confidence interval on p-value:
##  0.001347329 0.025105152

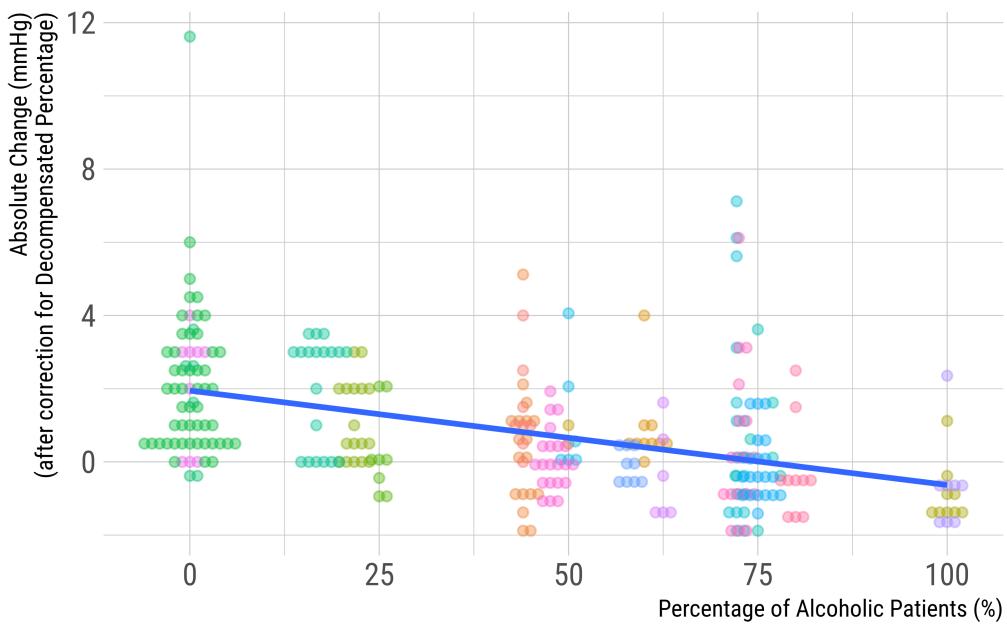
permTREND(formula=abschange ~ Perc_Alc, data=trt_wide_dc,
           method="exact.mc")

##
##  Exact Permutation Test Estimated by Monte Carlo
##
## data: x and y
## p-value = 0.006
## alternative hypothesis: true correlation of x and y is not equal to 0
## sample estimates:
## correlation of x and y
##                  -0.2099865
##
## p-value estimated from 999 Monte Carlo replications
## 99 percent confidence interval on p-value:
##  0.0002072893 0.0184986927

perc_alc_plot <- trt_wide %>%
  mutate(abschange_dccorr = correct_for_decomp(abschange ~ Perc_Alc + Perc_Decom)) %>%
  ggplot(aes(x=Perc_Alc, y=abschange_dccorr)) +
  geom_beeswarm(aes(colour=Study, group=Study), alpha=0.4, cex=1) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se=FALSE) +
  labs(y="Absolute Change (mmHg)\n(after correction for Decompensated Percentage)",
       x="Percentage of Alcoholic Patients (%)")

perc_alc_plot

```



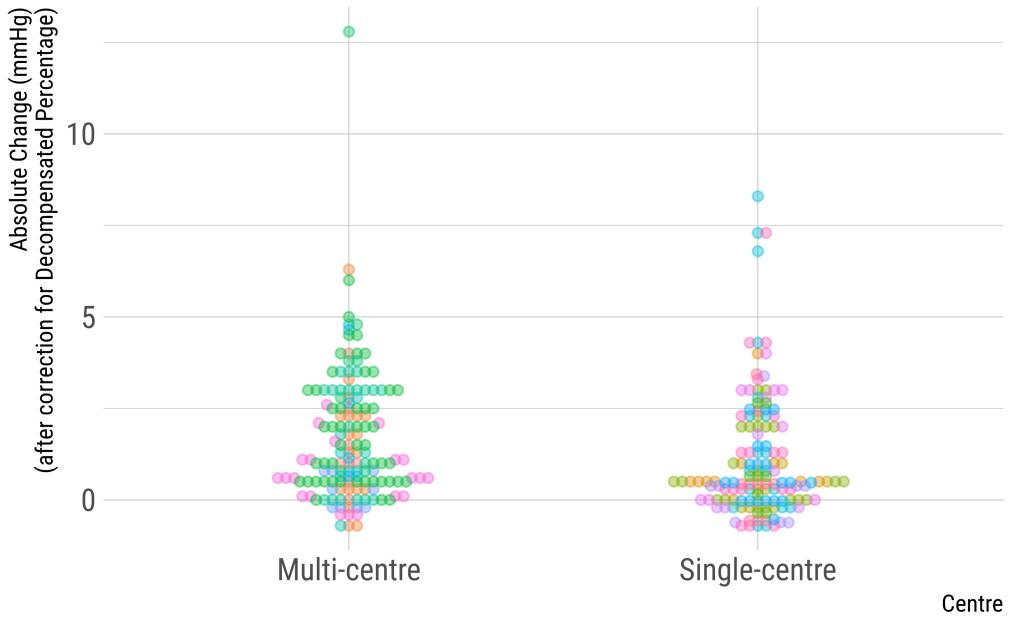
Multicentre

```
permuco::lmperm(abschange ~ Centre + Perc_Decom, data=trt_wide)
```

```
## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
##                               Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t) permutation
## (Intercept)           1.663139   0.165791 10.032          2.056e-20
## CentreSingle-centre -0.544349   0.216320 -2.516          1.242e-02          0.0062
## Perc_Decom            0.007055   0.002478  2.846          4.750e-03          0.9982
```

```
centre_plot <- trt_wide %>%
  mutate(abschange_dccorr = correct_for_decomp(abschange ~ Centre + Perc_Decom)) %>%
  ggplot(aes(x=Centre, y=abschange_dccorr)) +
  geom_beeswarm(aes(colour=Study, group=Study), alpha=0.4, cex=1) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se=FALSE) +
  labs(y="Absolute Change (mmHg)\n(after correction for Decompensated Percentage)")

centre_plot
```



Combined Model

```
permuco::lmperm(abschange ~ Perc_Decom + Centre + Perc_Alc, data=trt_wide)
```

```
## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
## Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t) permutation Pr(<=t)
## (Intercept) 1.94534 0.168789 11.52527 2.287e-25
## Perc_Decom 0.01884 0.003349 5.62673 4.493e-08 1.0000
## CentreSingle-centre 0.01892 0.236163 0.08012 9.362e-01 0.5338
## Perc_Alc -0.02599 0.005198 -4.99906 1.023e-06 0.0002
```

```
permuco::lmperm(abschange ~ Centre + Perc_Alc, data=trt_wide_c)
```

```
## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
## Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t) permutation Pr(<=t)
## (Intercept) 1.99513 0.170383 11.710 3.636e-21
## CentreSingle-centre -0.38265 0.310236 -1.233 2.200e-01 0.1070
## Perc_Alc -0.01289 0.007501 -1.718 8.858e-02 0.0474
```

```
permuco::lmperm(abschange ~ Centre + Perc_Alc, data=trt_wide_dc)
```

```
## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
```

```

##                               Estimate Std. Error t value parametric Pr(>|t|) permutation Pr(<t)
## (Intercept)            2.99299   0.440666   6.792           1.967e-10
## CentreSingle-centre  0.37572   0.372646   1.008           3.148e-01
## Perc_Alc              -0.02274  0.008097  -2.808           5.587e-03

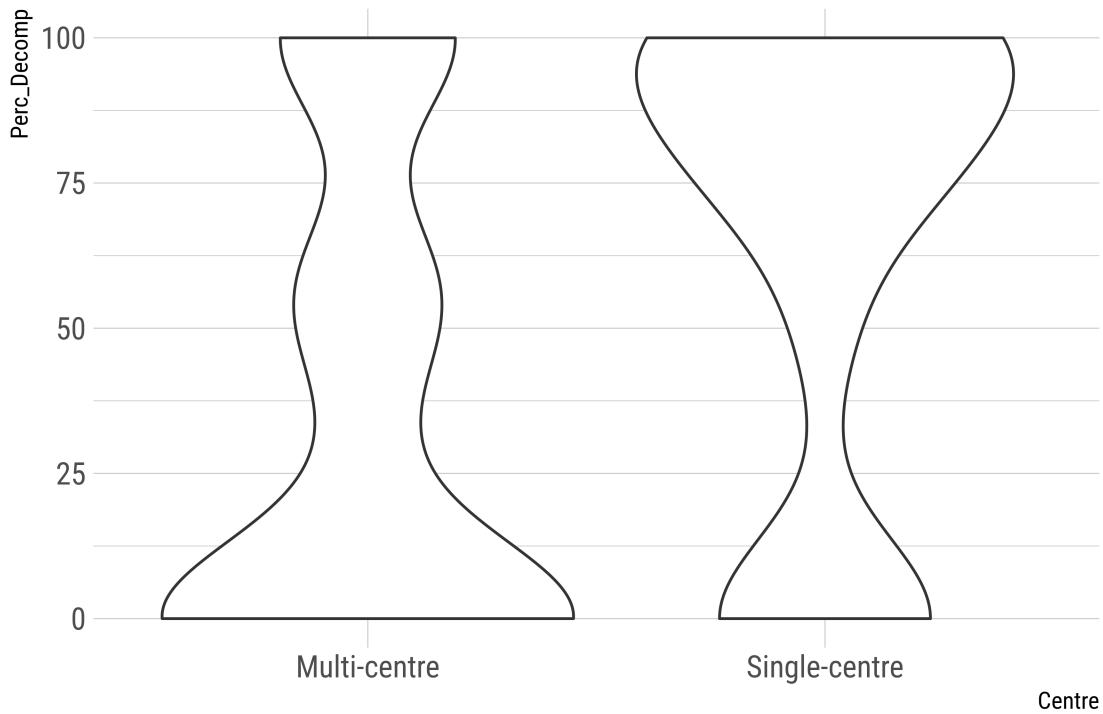
```

Let's examine why this might be.

```

trt_wide %>%
  ggplot(aes(y=Perc_Decom, x=Centre)) +
  geom_violin()

```



```

permTS(formula=Perc_Decom ~ as.factor(Centre), data=trt_wide,
       method="exact.mc")

```

```

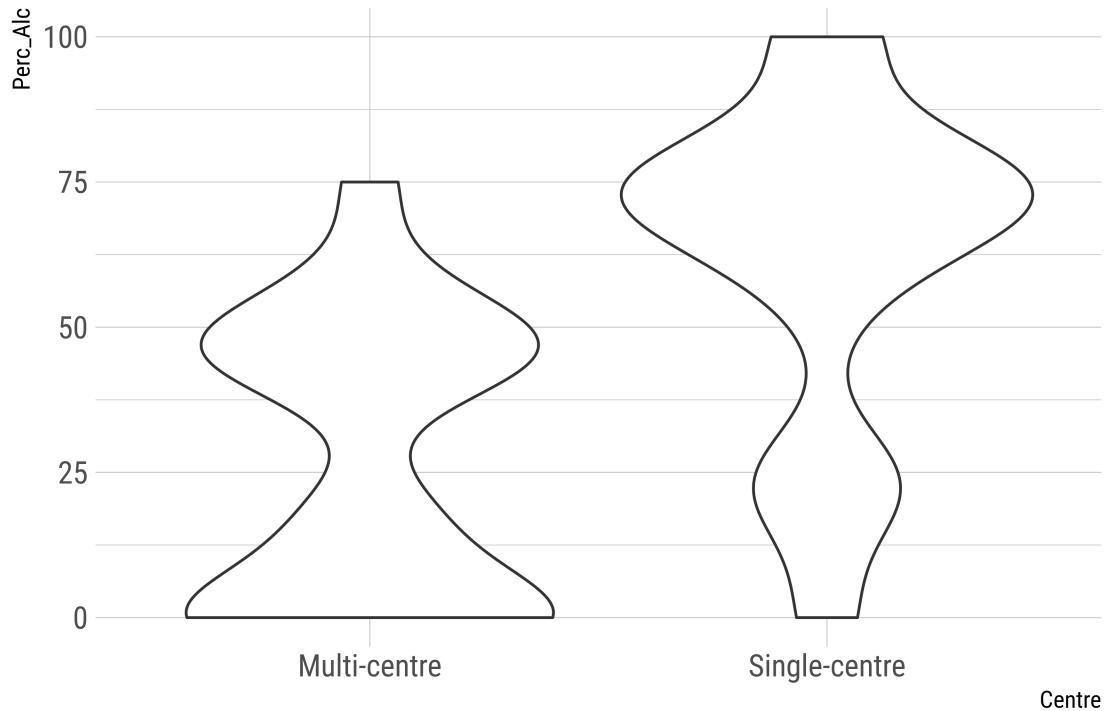
##
##  Exact Permutation Test Estimated by Monte Carlo
##
##  data:  Perc_Decom by as.factor(Centre)
##  p-value = 0.002
##  alternative hypothesis: true mean as.factor(Centre)=Multi-centre - mean as.factor(Centre)=Single-centre
##  sample estimates:
##  mean as.factor(Centre)=Multi-centre - mean as.factor(Centre)=Single-centre
##                                         -28.31913
##
##  p-value estimated from 999 Monte Carlo replications

```

```
## 99 percent confidence interval on p-value:
## 0.00000000 0.01057916
```

Most of the single-centre studies have high proportions of decompensated patients.

```
trt_wide %>%
  ggplot(aes(y=Perc_Alc, x=Centre)) +
  geom_violin()
```



```
permTS(formula=Perc_Alc ~ as.factor(Centre), data=trt_wide,
       method="exact.mc")
```

```
##
##  Exact Permutation Test Estimated by Monte Carlo
##
##  data:  Perc_Alc by as.factor(Centre)
##  p-value = 0.002
##  alternative hypothesis: true mean as.factor(Centre)=Multi-centre - mean as.factor(Centre)=Single-centre
##  sample estimates:
##  mean as.factor(Centre)=Multi-centre - mean as.factor(Centre)=Single-centre
##                                         -34.52329
##
##  p-value estimated from 999 Monte Carlo replications
##  99 percent confidence interval on p-value:
##  0.00000000 0.01057916
```

Similarly, most of the single-centre studies have high proportions of alcoholic patients.

Days

Reviewers were surprised at the lack of an effect of the number of days elapsed. Perhaps this was obscured by not having included the other confounding factors.

```
permuco::lmperm(abschange ~ Perc_Decom + Centre + Perc_Alc + Days, data=trt_wide)
```

```
## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
##           Estimate Std. Error   t value parametric Pr(>|t|) permutation Pr(<t) permutation
## (Intercept) 2.0833441  0.2090856  9.96407      3.545e-20
## Perc_Decom  0.0178489  0.0034640  5.15263      4.898e-07    1.0000
## CentreSingle-centre 0.0234628  0.2360917  0.09938      9.209e-01    0.5348
## Perc_Alc    -0.0262613  0.0052018 -5.04852      8.097e-07    0.0002
## Days        -0.0007942  0.0007107 -1.11745      2.648e-01    0.1308
```

This does not appear to be the case!

Mean Value

Absolute

```
permuco::lmperm(abschange ~ meanval + Perc_Decom, data=trt_wide)
```

```
## Table of marginal t-test of the betas
## Permutation test using freedman_lane to handle nuisance variables and 5000 permutations.
##           Estimate Std. Error   t value parametric Pr(>|t|) permutation Pr(<t) permutation
## (Intercept) 1.085870  0.422931   2.567      0.01077
## meanval     0.025623  0.024418   1.049      0.29494      0.8470    0.1532
## Perc_Decom  0.004602  0.002401   1.917      0.05630      0.9774    0.0228
```

```
permTREND(formula=abschange ~ meanval, data=trt_wide_c,
           method="exact.mc")
```

```
##
##  Exact Permutation Test Estimated by Monte Carlo
##
## data: x and y
## p-value = 0.654
## alternative hypothesis: true correlation of x and y is not equal to 0
## sample estimates:
## correlation of x and y
##                  -0.04578405
##
## p-value estimated from 999 Monte Carlo replications
## 99 percent confidence interval on p-value:
##  0.5770505 0.7316143
```

```

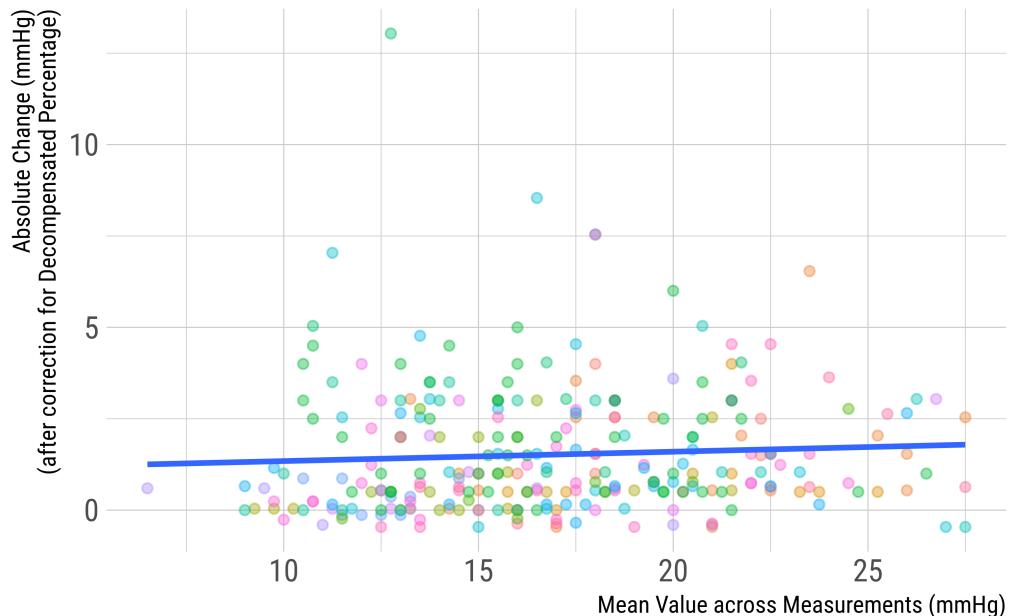
permTREND(formula=abschange ~ meanval, data=trt_wide_dc,
          method="exact.mc")

##
##  Exact Permutation Test Estimated by Monte Carlo
##
## data:  x and y
## p-value = 0.114
## alternative hypothesis: true correlation of x and y is not equal to 0
## sample estimates:
## correlation of x and y
##                  0.1302787
##
## p-value estimated from 999 Monte Carlo replications
## 99 percent confidence interval on p-value:
##  0.07792884 0.15503144

meanv_abs_plot <- trt_wide %>%
  mutate(abschange_dccorr = correct_for_decomp(abschange ~ meanval + Perc_DecomP)) %>%
  ggplot(aes(x=meanval, y=abschange_dccorr)) +
  geom_point(aes(colour=Study, group=Study), alpha=0.4) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se=FALSE) +
  labs(y="Absolute Change (mmHg)\n(after correction for Decompensated Percentage)",
       x="Mean Value across Measurements (mmHg)")

meanv_abs_plot

```



Signed

```
meanv_sign <- lmer(change ~ meanval + Perc_Decom + (1 | Study), data = trt_wide)
summary(meanv_sign)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: change ~ meanval + Perc_Decom + (1 | Study)
##   Data: trt_wide
##
## REML criterion at convergence: 1304
##
## Scaled residuals:
##    Min     1Q   Median     3Q    Max
## -4.8083 -0.6359  0.0657  0.5932  3.9860
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   Study    (Intercept) 0.4263   0.6529
##   Residual           5.5354   2.3527
## Number of obs: 281, groups: Study, 21
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept) -1.773365  0.658135 70.196412 -2.695 0.008813 **
## meanval      0.125777  0.034937 258.265538  3.600 0.000381 ***
## Perc_Decom  -0.007595  0.004939 10.005645 -1.538 0.155110
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) meanvl
## meanval   -0.860
## Perc_Decom -0.301 -0.110

meanv_sign_nocorr <- lmer(change ~ meanval + (1 | Study), data = trt_wide)
summary(meanv_sign_nocorr)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: change ~ meanval + (1 | Study)
##   Data: trt_wide
##
## REML criterion at convergence: 1297.6
##
## Scaled residuals:
##    Min     1Q   Median     3Q    Max
## -4.8493 -0.6062  0.0470  0.5689  3.9289
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   Study    (Intercept) 0.505    0.7106
##   Residual           5.530    2.3515
## Number of obs: 281, groups: Study, 21
```

```

## 
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|) 
## (Intercept) -2.1062    0.6338 161.0550 -3.323 0.001102 ** 
## meanval      0.1215    0.0349 258.7515  3.481 0.000586 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Correlation of Fixed Effects:
##          (Intr) 
## meanval -0.938

```

```

meanv_sign_c <- lmer(change ~ meanval + (1 | Study), data = trt_wide_c)
summary(meanv_sign_c)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: change ~ meanval + (1 | Study)
##   Data: trt_wide_c
## 
## REML criterion at convergence: 505.3
## 
## Scaled residuals:
##       Min     1Q   Median     3Q    Max 
## -1.95680 -0.66000 -0.09292  0.72674  2.50831 
## 
## Random effects:
##   Groups   Name        Variance Std.Dev. 
##   Study    (Intercept) 0.1447   0.3804  
##   Residual            4.5220   2.1265  
##   Number of obs: 115, groups: Study, 7 
## 
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|) 
## (Intercept) -1.82528    0.95247  64.26834 -1.916   0.0598 . 
## meanval      0.12622    0.05556 102.76959   2.272   0.0252 * 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Correlation of Fixed Effects:
##          (Intr) 
## meanval -0.961

```

```

meanv_sign_dc <- lmer(change ~ meanval + (1 | Study), data = trt_wide_dc)
summary(meanv_sign_dc)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: change ~ meanval + (1 | Study)
##   Data: trt_wide_dc
## 
## REML criterion at convergence: 789.3
## 
## Scaled residuals:
##       Min     1Q   Median     3Q    Max 
## -1.95680 -0.66000 -0.09292  0.72674  2.50831 
## 
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|) 
## (Intercept) -1.82528    0.95247  64.26834 -1.916   0.0598 . 
## meanval      0.12622    0.05556 102.76959   2.272   0.0252 * 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Correlation of Fixed Effects:
##          (Intr) 
## meanval -0.961

```

```

## -4.4469 -0.5597  0.1251  0.5181  3.7263
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## Study    (Intercept) 0.6818   0.8257
## Residual           6.2470   2.4994
## Number of obs: 166, groups: Study, 14
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -2.33529   0.83332 100.93794 -2.802 0.00608 **
## meanval      0.12351   0.04499 153.92550  2.745 0.00677 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## meanval -0.932

correct_for_decomp_lmer <- function(formula) {

  formula <- as.formula(formula)

  coefficients <- fixef(lmer(formula, data=trt_wide))

  predicted <- as.character(formula[2])

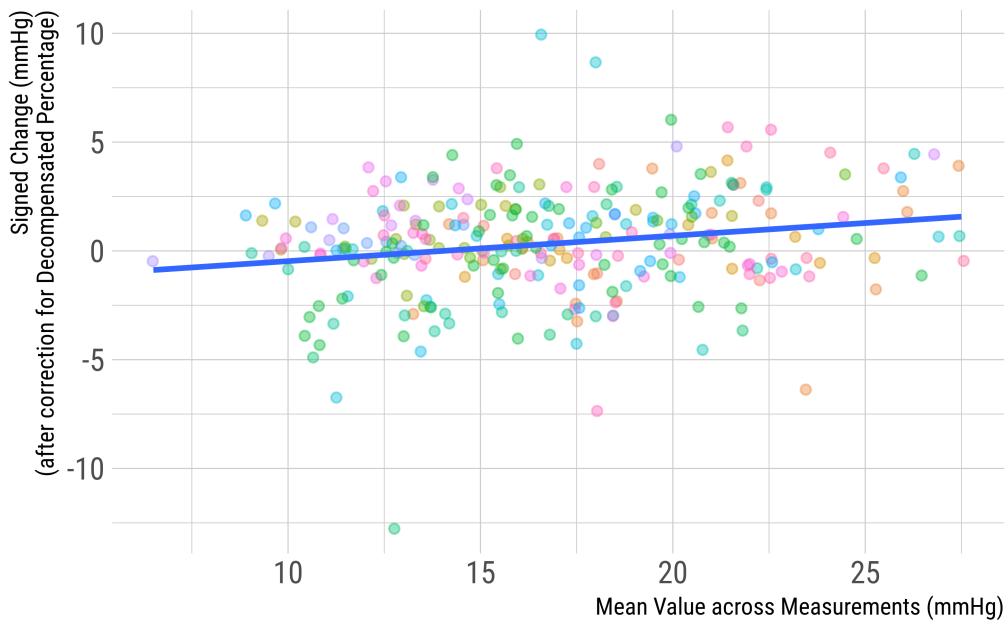
  after_decomp_corr <- trt_wide[[predicted]] -
    trt_wide[["Perc_DecomP"]] * coefficients[which(names(coefficients)== "Perc_DecomP")]

  return(after_decomp_corr)
}

meanv_signed_plot <- trt_wide %>%
  mutate(change_dccorr = correct_for_decomp_lmer("change ~ meanval + Perc_DecomP + (1 | Study)")) %>%
  ggplot(aes(x=meanval, y=change_dccorr)) +
  geom_jitter(aes(colour=Study, group=Study), alpha=0.4, height = 0.2) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se=FALSE) +
  labs(y="Signed Change (mmHg)\n(after correction for Decompensated Percentage)",
       x="Mean Value across Measurements (mmHg)")

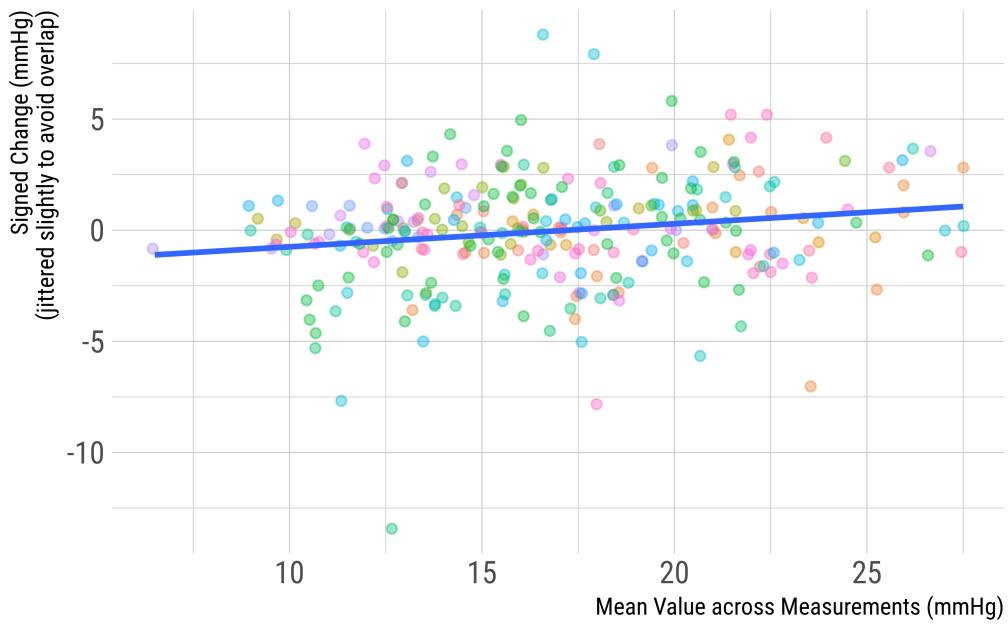
meanv_signed_plot

```



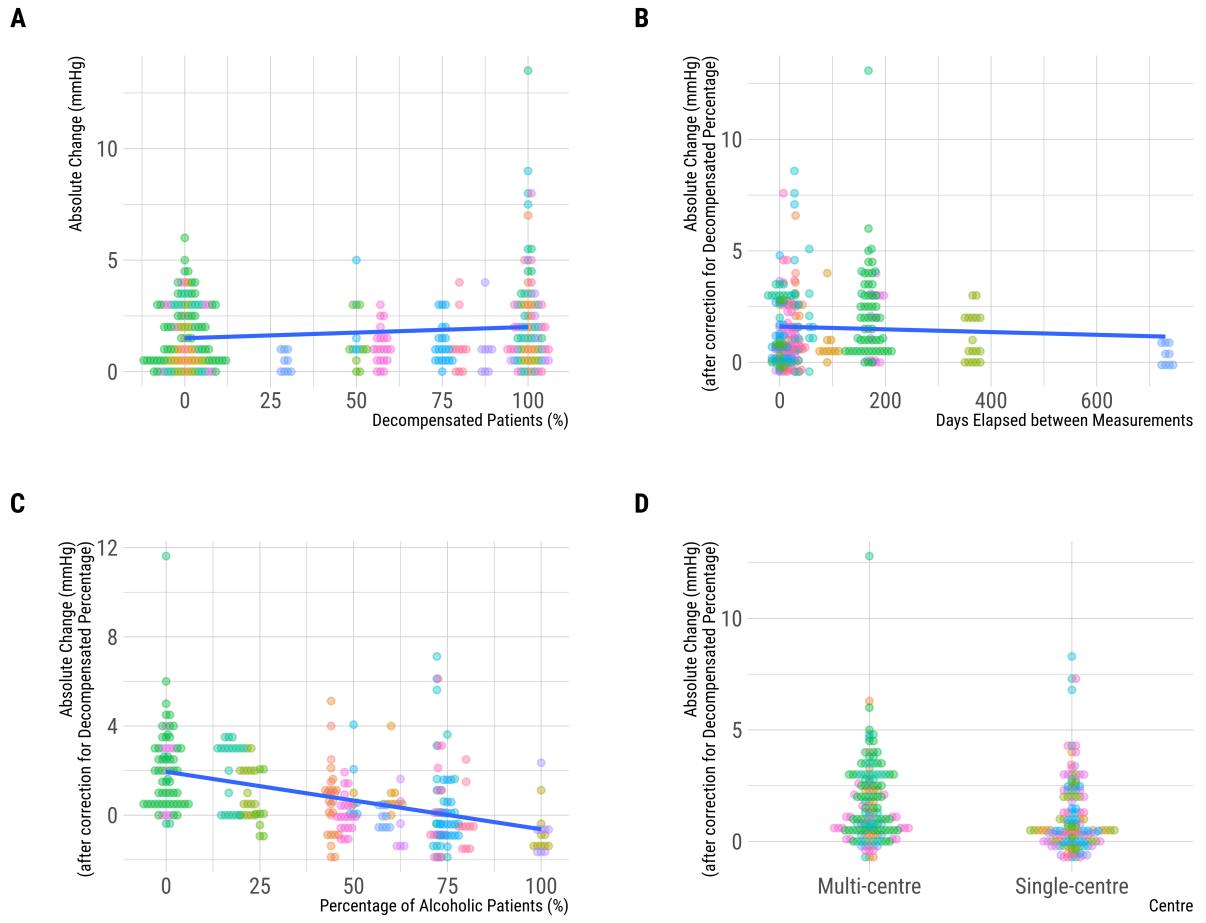
```
meanv_signed_plot_nocorr <- ggplot(data=trt_wide, aes(x=meanval, y=change)) +
  geom_jitter(aes(colour=Study, group=Study), alpha=0.4, height = 0.2) +
  guides(colour=FALSE) +
  geom_smooth(method="lm", se=FALSE) +
  labs(y="Signed Change (mmHg)\n(jittered slightly to avoid overlap)",
       x="Mean Value across Measurements (mmHg)")

meanv_signed_plot_nocorr
```

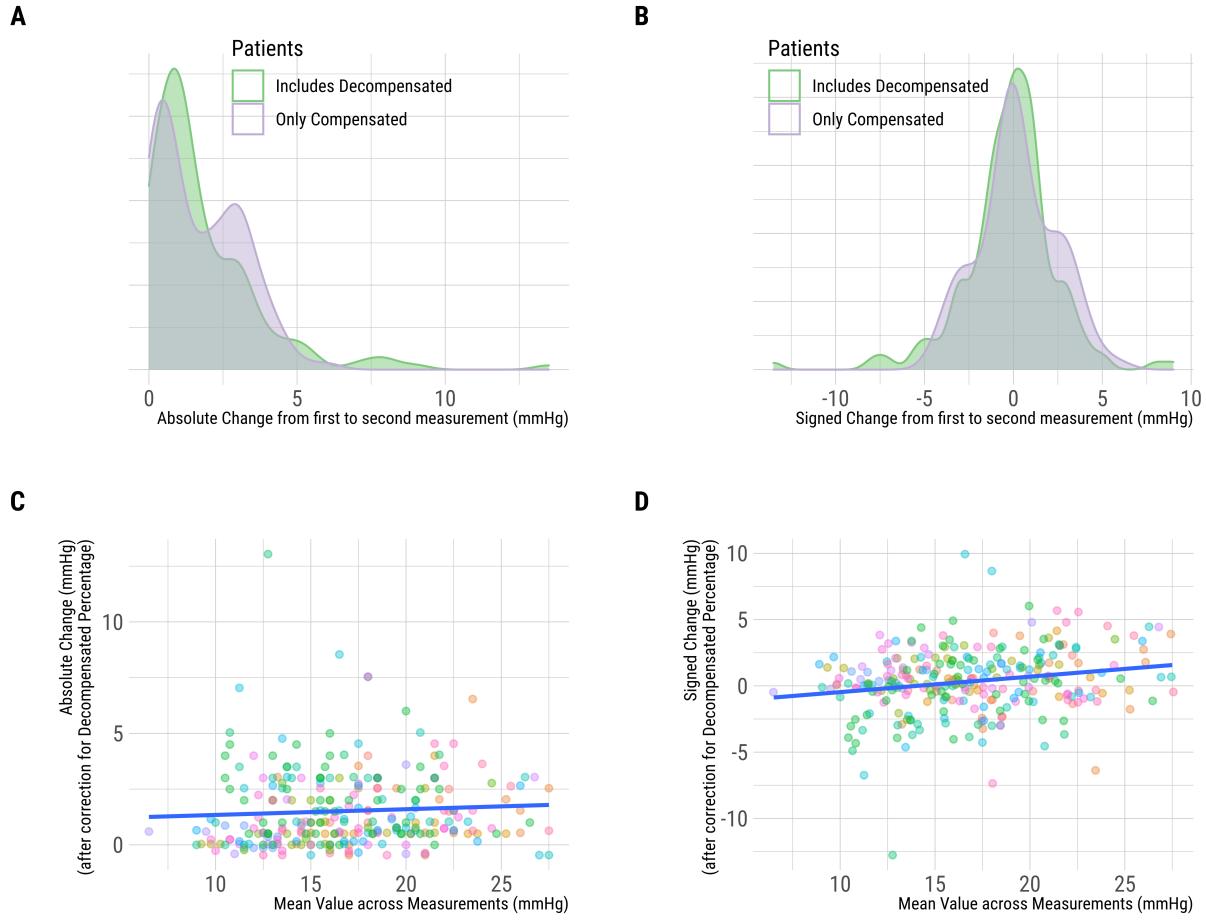


Figures

```
cowplot::plot_grid(decomp_plot, days_plot,
                    perc alc_plot, centre_plot, align = "hv",
                    ncol = 2, labels = "AUTO")
```



```
cowplot::plot_grid(abs_change_distr, sign_change_distr,
  meanv_abs_plot, meanv_signed_plot,
  align = "hv",
  ncol = 2, labels = "AUTO")
```



Power Analysis for a difference

Now, the core thing we want to do here is to perform a power analysis for examining within-individual effects.

One way of doing this is to use the `signvar_sd` column of the `tidytrt` object. This is the standard deviation of the signed changes, and hence, if we assume a change after an intervention, this is the SD we could imagine being true, and thus, the effect size, the Cohen's Dz, is equal to difference / `signvar_sd`. However, this method makes an assumption that everyone changes by exactly the same amount: the effect (before accounting for error) is completely uniform. This may be the case, but this is the most optimistic scenario. We should be taking into consideration the possibility of heterogeneous effects.

First, let's make a little plot to show what I mean by homogeneous and heterogeneous effects.

```
set.seed(1234)

trt_all_comp <- trt_all[2,]

wscv=trt_all_comp$wscv
meanval=trt_all_comp$mean
cv=trt_all_comp$cv
icc=trt_all_comp$icc
```

```

sd_true <- sqrt(icc * (cv * meanval)^2)

n <- 20
delta <- 2

# Homogeneous
cv_delta <- 0

pre_true <- rnorm(n, meanval, sd_true)
pre_meas <- pre_true + rnorm(n, 0, meanval*wscv)

post_true <- pre_true - rnorm(n, delta, cv_delta*delta)
post_meas <- post_true + rnorm(n, 0, meanval*wscv)

hom_true <- tibble::tibble(
  ID = rep(1:n, times=2),
  Outcome = c(pre_true, post_true),
  PrePost = rep(c("Pre", "Post"), each=n),
  Effects = "Homogeneous Effects",
  MeasuredTrue = "True Values"
)

hom_measured <- tibble::tibble(
  ID = rep(1:n, times=2),
  Outcome = c(pre_meas, post_meas),
  PrePost = rep(c("Pre", "Post"), each=n),
  Effects = "Homogeneous Effects",
  MeasuredTrue = "Measured Values"
)

# hom_difference <- tibble::tibble(
#   ID = rep(1:n, times=2),
#   Outcome = c(post_meas-pre_),
#   PrePost = rep(c("Pre", "Post"), each=n),
#   Effects = "Homogeneous",
#   MeasuredTrue = "Difference"
# )

# Heterogeneous
cv_delta <- 0.5

#pre_true <- rnorm(n, meanval, abs(cv*meanval)) # Use same as above
#pre_meas <- pre_true + rnorm(n, 0, abs(meanval*wscv))

post_true <- pre_true - rnorm(n, delta, abs(cv_delta*delta))
post_meas <- post_true + rnorm(n, 0, abs(meanval*wscv))

het_true <- tibble::tibble(
  ID = rep(1:n, times=2),
  Outcome = c(pre_true, post_true),
  PrePost = rep(c("Pre", "Post"), each=n),
  Effects = "Heterogeneous Effects",

```

```

MeasuredTrue = "True Values"
)

het_measured <- tibble::tibble(
  ID = rep(1:n, times=2),
  Outcome = c(pre_meas, post_meas),
  PrePost = rep(c("Pre", "Post"), each=n),
  Effects = "Heterogeneous Effects",
  MeasuredTrue = "Measured Values"
)

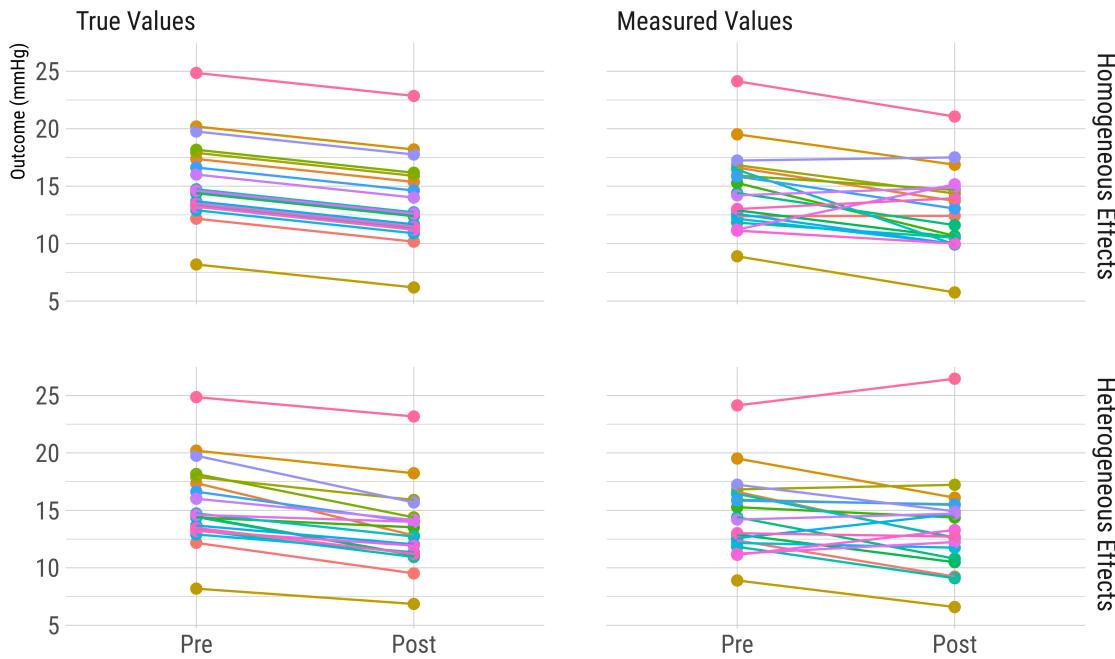
# Plot
effects <- bind_rows(hom_true, hom_measured, het_true, het_measured) %>%
  mutate(MeasuredTrue = fct_inorder(MeasuredTrue),
        Effects = fct_inorder(Effects),
        PrePost = fct_inorder(PrePost),
        ID = as.factor(ID))

ggplot(effects, aes(x=PrePost, y=Outcome, colour=ID, group=ID)) +
  geom_point(size=2) +
  geom_line() +
  facet_grid(Effects~MeasuredTrue) +
  labs(y="Outcome (mmHg)",
       colour="Values",
       x=NULL,
       title="Homogeneous and Heterogeneous Effects",
       subtitle="Homogeneous effects imply that the true underlying change is the same\nin all individuals",
       guides(colour=FALSE)

```

Homogeneous and Heterogeneous Effects

Homogeneous effects imply that the true underlying change is the same in all individuals (hence parallel lines in the true values)



So, to summarise, we have underlying true values, and measured values after accounting for measurement error. The change from before to after the intervention can either be homogeneous (everyone has exactly the same effect), or heterogeneous (effect sizes differ, and some even get harmed by the intervention - about 2.5% as I've chosen the SD of the intervention effect as 50% of the mean effect, so 0 effect is 2 SDs away from the mean effect size, which is approximately 2.5%). Then, the measured values appear to show more people getting worse after treatment, but this is just due to measurement error.

```

heterogen_cv <- 0.5

annotations <- tibble(
  x=c(-3, 0),
  text=c(paste0(round(100*pnorm(1/heterogen_cv)), "% experience improvements"),
         paste0(100-round(100*pnorm(1/heterogen_cv)), "% experience worsening")),
  colour = c("#61b096", "#bd7969")
)

heterogen_effects <- tibble(Effect=c(-3, 1))

ggplot(heterogen_effects, aes(x=Effect)) +
  geom_area(stat="function", fun = dnorm, fill="#61b096", xlim=c(-3, 0),
            args = list(mean = -1, sd=heterogen_cv), alpha=0.7) +
  geom_area(stat="function", fun = dnorm, fill="#bd7969", xlim=c(0, 1),
            args = list(mean = 1, sd=heterogen_cv), alpha=0.7) +
  annotate("text", x=-2.5, y=0.7,

```

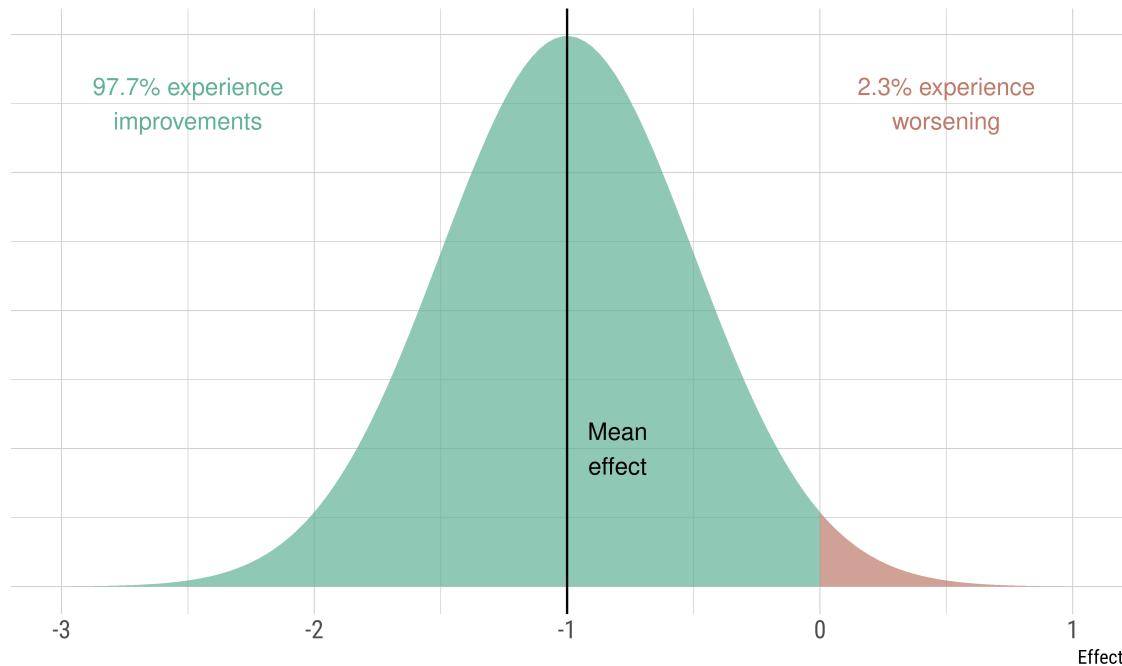
```

label=paste0(round(100*pnorm(1/heterogen_cv), 1),
             "% experience\nimprovements"),
colour = "#61b096", hjust=0.5) +
annotate("text", x=0.5, y=0.7,
         label=paste0(100-round(100*pnorm(1/heterogen_cv), 1),
                     "% experience\nworsening"),
         colour = "#bd7969", hjust=0.5) +
annotate("text", x=-0.8, y=0.2,
         label="Mean\neffect", hjust=0.5) +
theme(axis.title.y=element_blank(),
      axis.text.y=element_blank(),
      axis.ticks.y=element_blank()) +
geom_vline(xintercept = -1) +
labs(title="Heterogeneous Effects",
     subtitle=paste0("Using a 50% CV for effect heterogeneity implies that some ",
                   "participants\nmay benefit more and others may even have ",
                   "worsening."))

```

Heterogeneous Effects

Using a 50% CV for effect heterogeneity implies that some participants may benefit more and others may even have worsening.



This would imply that for all different sizes of effect, that only 2.3% experience a true worsening. However, when we measure the values, there are also several individuals who will exhibit an apparent worsening (increases from first to second measurement), when their true underlying values exhibited improvements. Let's calculate what fraction of individuals this would be.

True and Apparent Changes

Here, we can also observe the percentage of individuals who show apparent 10% and 20% changes from baseline.

```
apparent_effects <- function(n, delta, cv_delta, wscv=trt_all$wscv,
                               mean=trt_all$mean, cv=trt_all$cv, icc=trt_all$icc,
                               decomp) {

  wscv <- wscv[decomp]
  mean <- mean[decomp]
  cv   <- cv[decomp]

  var <- (cv*mean)^2
  sd_true <- sqrt(var * icc)

  pre_true <- rnorm(n, mean, sd_true)

  pre_meas <- pre_true + rnorm(n, 0, mean*wscv)

  post_true <- pre_true - rnorm(n, delta, cv_delta*delta)

  post_meas <- post_true + rnorm(n, 0, mean*wscv)

  measured <- tibble::tibble(
    ID = rep(1:n, times=2),
    Outcome = c(pre_meas, post_meas),
    PrePost = rep(c("Pre", "Post"), each=n)
  ) %>%
    spread(PrePost, Outcome)

  true <- tibble::tibble(
    ID = rep(1:n, times=2),
    Outcome = c(pre_true, post_true),
    PrePost = rep(c("Pre", "Post"), each=n)
  ) %>%
    spread(PrePost, Outcome)

  out <- list()

  out$apparent_worse <- round(100*with(measured, mean(Post > Pre)),1)
  out$apparent_10     <- round(100*with(measured, mean((Pre-Post)/Pre > 0.1)),1)
  out$apparent_20     <- round(100*with(measured, mean((Pre-Post)/Pre > 0.2)),1)
  out$true_worse     <- round(100*with(true, mean(Post > Pre)),1)
  out$true_10          <- round(100*with(true, mean((Pre-Post)/Pre > 0.1)),1)
  out$true_20          <- round(100*with(true, mean((Pre-Post)/Pre > 0.2)),1)

  return(out)
}

if(!file.exists("../DerivedData/percdifs.rds") || overwrite) {
```

```

measured_percs <- tidyverse::crossing(
  delta = seq(0, 3, by=0.5),
  cv_delta = c(0, 0.5),
  decomp=c(1,2)
) %>%
  mutate(condition = 1:n()) %>%
  group_by(condition) %>%
  nest() %>%
  mutate(res = map(data, ~apparent_effects( n=1e7,
                                             delta=.x$delta,
                                             cv_delta = .x$cv_delta,
                                             decomp=.x$decomp))) %>%
  ungroup()

saveRDS(measured_percs, "../DerivedData/percdifs.rds")

}

measured_percs <- readRDS("../DerivedData/percdifs.rds")

measured_percs_summary <- measured_percs %>%
  mutate(apparent_worse = map_dbl(res, "apparent_worse"),
         apparent_10 = map_dbl(res, "apparent_10"),
         apparent_20 = map_dbl(res, "apparent_20"),
         true_worse = map_dbl(res, "true_worse"),
         true_10 = map_dbl(res, "true_10"),
         true_20 = map_dbl(res, "true_20")) %>%
  select(-res) %>%
  unnest(data) %>%
#  mutate(true_worse = round(100*(1-pnorm(1/cv_delta)),1)) %>%
  arrange(decomp, delta, cv_delta) %>%
  mutate(decomp = ifelse(decomp==1,
                        trt_all$decomp[1],
                        trt_all$decomp[2]),
         cv_delta = ifelse(cv_delta==0,
                           "Homogeneous",
                           "Heterogeneous")) %>%
  select(decomp, condition, delta, cv_delta,
         true_worse, apparent_worse,
         true_10, apparent_10,
         true_20, apparent_20) %>%
  rename("Patients" = decomp,
         "Apparent Worsening (%)" = apparent_worse,
         "True Worsening (%)" = true_worse,
         "True 10%+ Improvement (%)" = true_10,
         "Apparent 10%+ Improvement (%)" = apparent_10,
         "True 20%+ Improvement (%)" = true_20,
         "Apparent 20%+ Improvement (%)" = apparent_20,
         "Effects" = cv_delta,
         "Difference" = delta) %>%
  select(-condition)

decomp_change <- head(

```

```

which(
  measured_percs_summary$Patients ==
    trt_all$decomp[2]), 1)

knitr::kable(measured_percs_summary[,-1]) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows(trt_all$decomp[1], 1, decomp_change-1) %>%
  pack_rows(trt_all$decomp[2], decomp_change, nrow(measured_percs_summary))

appchange <- knitr::kable(measured_percs_summary[,-1]) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows(trt_all$decomp[1], 1, decomp_change-1) %>%
  pack_rows(trt_all$decomp[2], decomp_change, nrow(measured_percs_summary))

appchange

# save_kable(appchange, file = "figures/appchange.jpg")

```

So, for a true effect of about 2mmHg in compensated patients, it will appear as if 12-16% would appear to worsen. This fits with clinical experience.

Simulation

Note: these are no longer run as we instead make use of the analytical solutions

```

HVPG_dif_sim <- function(n, delta, cv_delta, wscv=trt_all$wscv,
                           mean=trt_all$mean, cv=trt_all$cv, icc=trt_all$icc,
                           decomp = 1) {

  wscv <- wscv[decomp]
  mean <- mean[decomp]
  cv   <- cv[decomp]

  var <- (cv*mean)^2
  sd_true <- sqrt(var * icc)

  pre_true <- rnorm(n, mean, sd_true)

  pre_meas <- pre_true + rnorm(n, 0, mean*wscv)

  post_true <- pre_true - rnorm(n, delta, cv_delta*delta)

  post_meas <- post_true + rnorm(n, 0, mean*wscv)

  measured <- tibble::tibble(
    ID = rep(1:n, times=2),
    Outcome = c(pre_meas, post_meas),
    PrePost = rep(c("Pre", "Post"), each=n)
  )

```

```

d <- effsize::cohen.d(measured$Outcome, measured$PrePost,
                      paired=TRUE)$estimate

dz <- effsize::cohen.d(measured$Outcome, measured$PrePost,
                      paired=TRUE, within=FALSE)$estimate

test <- t.test(pre_meas, post_meas, alternative = "greater",
                paired = T)

# Note: one-sided p value

testout <- broom::tidy(test)

out <- mutate(testout, d = d, dz = dz)

return(out)

}

```

Now, we set up the simulation parameters for various scenarios.

```

difsimpars <- tidyr::crossing(
  n=seq(5, 100, by = 5),
  delta = seq(1,3, by=0.5),
  cv_delta = c(0, 0.5),
)

```

And now we run them

```

if(!file.exists(paste0("../DerivedData/difsims_decomp_",
                      nsims, ".rds")) || overwrite) {

  pb <- progress_bar$new(total = nrow(difsimpars))

  difsims <- difsimpars %>%
    mutate(sim = 1:nrow(difsimpars)) %>%
    group_by(sim) %>%
    nest(params = c(n, delta, cv_delta)) %>%
    mutate(output = map(params,
                        ~{pb$tick();
                          bind_rows(purrr::rerun(nsims,
                                                 HVPG_dif_sim(.x$n, .x$delta,
                                                 .x$cv_delta,
                                                 decomp=1))))})

  saveRDS(difsims, paste0("../DerivedData/difsims_decomp_", nsims, ".rds"))

}

if(!file.exists(paste0("../DerivedData/difsims_comp_",
                      nsims, ".rds")) || overwrite) {

  pb <- progress_bar$new(total = nrow(difsimpars))

```

```

difsims <- difsimpars %>%
  mutate(sim = 1:nrow(difsimpars)) %>%
  group_by(sim) %>%
  nest(params = c(n, delta, cv_delta)) %>%
  mutate(output = map(params,
    ~{pb$tick();
      bind_rows(purrr::rerun(nsims,
        HVPG_dif_sim(.x$n, .x$delta,
        .x$cv_delta,
        decomp=2))))})

saveRDS(difsims, paste0("../DerivedData/difsims_comp_", nsims, ".rds"))

}

```

And extract the results

```

difsims_decomp <- readRDS(
  paste0("../DerivedData/difsims_decomp_", nsims, ".rds"))

difsims_decomp_res <- difsims_decomp %>%
  ungroup() %>%
  mutate(power = map_dbl(output, ~mean(.x$p.value < 0.05))) %>%
  unnest(params) %>%
  mutate(delta = as.factor(delta)) %>%
  mutate(Effects = ifelse(cv_delta==0, "Homogeneous Effects",
    "Heterogeneous Effects")) %>%
  mutate(Effects = fct_inorder(Effects)) %>%
  mutate(decomp = trt_all$decomp[1])

difsims_comp <- readRDS(
  paste0("../DerivedData/difsims_comp_", nsims, ".rds"))

difsims_comp_res <- difsims_comp %>%
  ungroup() %>%
  mutate(power = map_dbl(output, ~mean(.x$p.value < 0.05))) %>%
  unnest(params) %>%
  mutate(delta = as.factor(delta)) %>%
  mutate(Effects = ifelse(cv_delta==0, "Homogeneous Effects",
    "Heterogeneous Effects")) %>%
  mutate(Effects = fct_inorder(Effects)) %>%
  mutate(decomp = trt_all$decomp[2])

difsims_res <- bind_rows(difsims_comp_res, difsims_decomp_res)

```

Plotting

```
ggplot(difsims_res, aes(x=n, y=power, colour=delta)) +  
  geom_point() +  
  geom_line() +  
  facet_grid(decomp~Effects) +  
  coord_cartesian(ylim=c(0.5, 1)) +  
  scale_color_brewer(type = "qual", palette = 2) +  
  annotate("rect", ymin = 0.8, ymax = 1, xmin=0,  
          xmax=100, alpha = .4, fill="grey") +  
  labs(y="Power", x="Sample Size",  
       colour="Intervention\\nEffect (mmHg)")
```

Required Individuals

And how many people do we need for each scenario?

```
difsims_80power <- difsims_res %>%  
  arrange(power) %>%  
  filter(power > 0.8) %>%  
  select(decomp, delta, Effects, n) %>%  
  group_by(delta, Effects, decomp) %>%  
  slice(1) %>%  
  arrange(decomp, delta) %>%  
  rename("Patients" = decomp,  
        "Difference (mmHg)" = delta,  
        "80% Power" = n)  
  
difsims_90power <- difsims_res %>%  
  arrange(power) %>%  
  filter(power > 0.9) %>%  
  select(decomp, delta, Effects, n) %>%  
  group_by(delta, Effects, decomp) %>%  
  slice(1) %>%  
  arrange(decomp, delta) %>%  
  rename("Patients" = decomp,  
        "Difference (mmHg)" = delta,  
        "90% Power" = n)  
  
difsims_power <- left_join(difsims_80power, difsims_90power)  
  
decomp_change <- head(  
  which(  
    difsims_power$Patients ==  
    trt_all$decomp[2]), 1)  
  
# kable(difsims_power[,-1]) %>%  
#   kable_styling("striped", full_width = F) %>%  
#   pack_rows(trt_all$decomp[1], 1, decomp_change-1) %>%  
#   pack_rows(trt_all$decomp[2], decomp_change, nrow(difsims_80power))
```

Analytical solution

```
difsims_ana <- function(Patients, Difference, Effects, Power,
                           trt_all=trt_all) {

  Effects <- as.character(Effects)
  Difference <- as.numeric(Difference)

  heterogen <- ifelse(Effects=="Homogeneous Effects",
                      0, 0.5)

  decomp = ifelse(Patients=="Includes Decompensated", 1, 2)

  trt_all_pat <- trt_all[decomp,]

  signvar_sd <- sqrt(
    (trt_all_pat$signvar_sd*trt_all_pat$mean)^2 +
    (heterogen*Difference)^2)

  dz <- Difference/signvar_sd

  ceiling(pwr::pwr.t.test(d=dz, sig.level = 0.05, power = Power,
                          alternative = "greater", type = "paired")$n)

}

difsims_power_ana <- difsims_power %>%
  rename(Difference = `Difference (mmHg)` %>%
  mutate(Difference = as.numeric(as.character(Difference)),
         Effects= as.character(Effects)) %>%
  group_by(Patients, Difference, Effects) %>%
  mutate("80% Power"= pmap_dbl(list(Patients, Difference,
                                      Effects), difsims_ana, Power=0.8,
                                      trt_all = trt_all),
        "90% Power"= pmap_dbl(list(Patients, Difference,
                                      Effects), difsims_ana, Power=0.9,
                                      trt_all = trt_all))

kable(difsims_power_ana[,-1]) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows(trt_all$decomp[1], 1, decomp_change-1) %>%
  pack_rows(trt_all$decomp[2], decomp_change, nrow(difsims_power_ana))

difpower <- kable(difsims_power_ana[,-1]) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows(trt_all$decomp[1], 1, decomp_change-1) %>%
  pack_rows(trt_all$decomp[2], decomp_change, nrow(difsims_power_ana))

difpower
```

```
# save_kable(difpower, "figures/difpower.jpg")
```

Contour Plots

```
difsims Ana contour <- function(Patients, Difference, Effects, n,
                                trt_all=trt_all) {

  Effects <- as.character(Effects)
  Difference <- as.numeric(Difference)

  heterogen <- ifelse(Effects=="Homogeneous Effects",
                      0, 0.5)

  decomp = ifelse(Patients=="Includes Decompensated", 1, 2)

  trt_all_pat <- trt_all[decomp,]

  signvar_sd <- sqrt(
    (trt_all_pat$signvar_sd*trt_all_pat$mean)^2 +
    (heterogen*Difference)^2)

  dz <- Difference/signvar_sd

  pwr::pwr.t.test(d=dz, sig.level = 0.05, n = n,
                   alternative = "greater", type = "paired")$power

}

contour_dat <- tidyverse::crossing(Difference = seq(0.3,3,length.out=2701),
                                    n = 5:80,
                                    Effects = c("Homogeneous Effects",
                                               "Heterogeneous Effects"),
                                    Patients = c("Includes Decompensated",
                                               "Only Compensated"))
```

Run it

```
contour_power <- contour_dat %>%
  mutate(Test = 1:n()) %>%
  group_by(Test) %>%
  mutate(Power=pmap_dbl(list(Patients, Difference, Effects, n),
                        difsims Ana contour, trt_all=trt_all)) %>%
  ungroup()

saveRDS(contour_power, "../DerivedData/contour_power.rds")
```

```
contour_power <- readRDS("../DerivedData/contour_power.rds")
```

```
library(viridis)
```

```
contour_power <- contour_power %>%
```

```

mutate(Power_cut = cut(Power, breaks = seq(0,1, length.out = 11))) %>%
  mutate(Power = ifelse(Power == 1, 0.99, Power)) # Note below to explain this

contour_het <- contour_power %>%
  filter(Effects!="Homogeneous Effects")

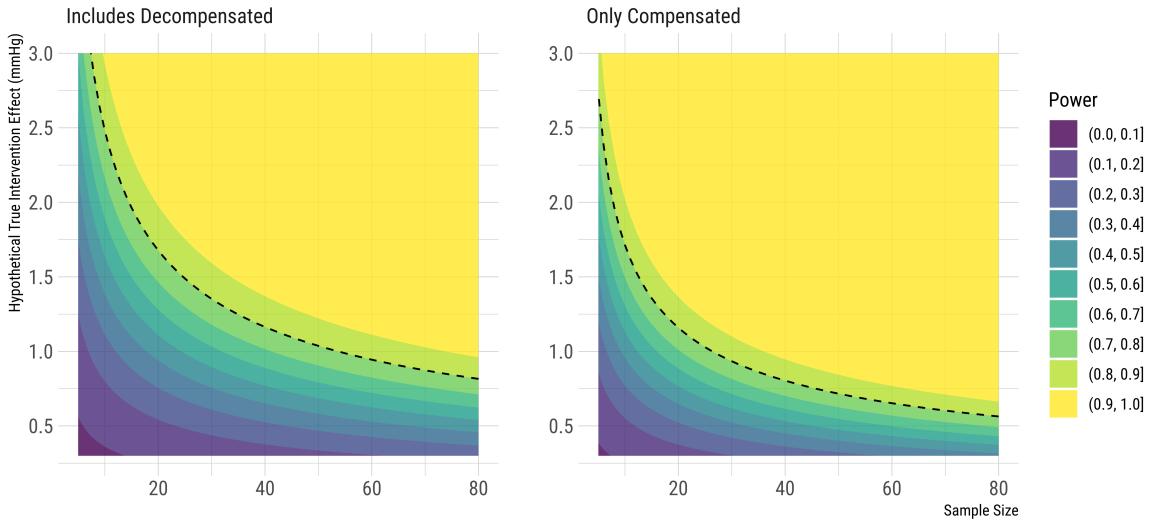
contour_hom <- contour_power %>%
  filter(Effects=="Homogeneous Effects")

homplot <- ggplot(contour_hom, aes(x=n, y=Difference, z=Power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1, by=0.1)) +
  facet_wrap(.~Patients, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical True Intervention Effect (mmHg)") +
  scale_y_continuous(breaks = seq(0.5, 3, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed")

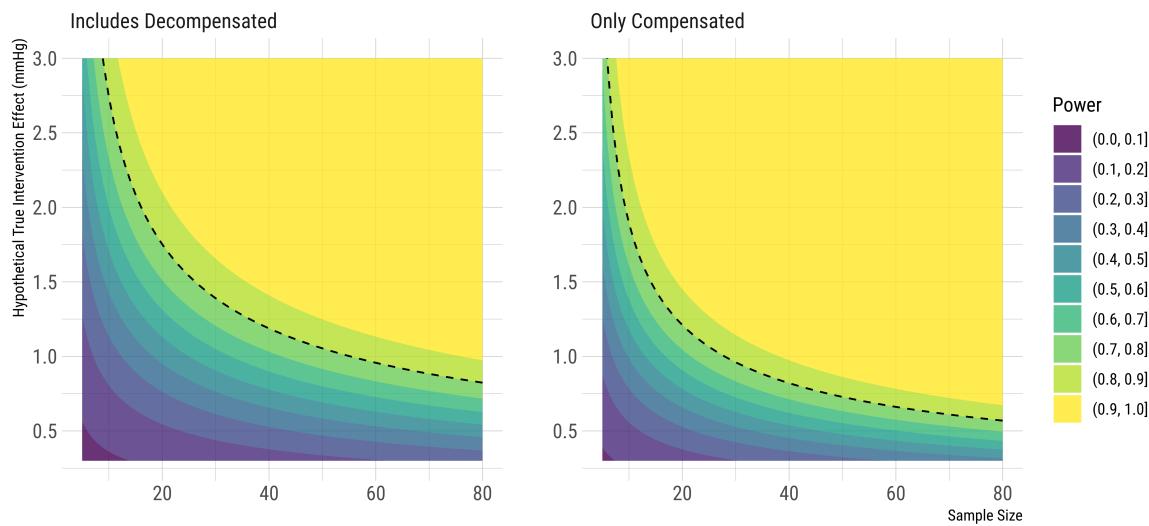
hetplot <- ggplot(contour_het, aes(x=n, y=Difference, z=Power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_wrap(.~Patients, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical True Intervention Effect (mmHg)") +
  scale_y_continuous(breaks = seq(0.5, 3, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed")

homplot

```



```
hetplot
```



```
ggsave(homplot, height=5, width=10, filename = "figures/Dif_hom_contour.png")
ggsave(hetplot, height=5, width=10, filename = "figures/Dif_het_contour.png")

ggsave(homplot, height=5, width=10, filename = "figures/Dif_hom_contour.jpg",
       dpi = 600)
ggsave(hetplot, height=5, width=10, filename = "figures/Dif_het_contour.jpg",
       dpi = 600)

# +
#   xlab("Minor allele frequency")+
#   ylab("Hypothetical effect size")+
#   scale_x_log10(expand = c(0, 0), position = "bottom") +
#   scale_y_continuous(expand = c(0, 0))+
#   geom_line(data = power.80, col = "black")+
#   theme_classic(base_size = 12)
```

The strange line of code where I convert those values = 1 to 0.99 is to prevent the graph from showing strangely. I think this has to do with floating point accuracy. Some of the values equal to 1, are rounded in the computer number system to a little bit above 1, and then the plot makes them white. So by setting them to 0.99, they're still the same colour, but the plot is filled correctly.

Power Analysis for Difference in Differences

Simulation

```
HVPG_difindif_sim <- function(n, delta1, delta2, cv_delta,
                                wscv=trt_all$wscv,
```

```

        mean=trt_all$mean,
        cv=trt_all$cv,
        decomp ) {

wscv <- wscv[decomp]
mean <- mean[decomp]
cv   <- cv[decomp]

var <- (cv*mean)^2
sd_true <- sqrt(var * icc)

pre_true1 <- rnorm(n, mean, sd_true)
pre_true2 <- rnorm(n, mean, sd_true)

pre_meas1 <- pre_true1 + rnorm(n, 0, mean*wscv)
pre_meas2 <- pre_true2 + rnorm(n, 0, mean*wscv)

post_true1 <- pre_true1 - rnorm(n, delta1, cv_delta*delta1)
post_true2 <- pre_true2 - rnorm(n, delta2, cv_delta*delta2)

post_meas1 <- post_true1 + rnorm(n, 0, mean*wscv)
post_meas2 <- post_true2 + rnorm(n, 0, mean*wscv)

measured <- tibble::tibble(
  ID = rep(1:n, times=2),
  Pre = c(pre_meas1, pre_meas2),
  Post = c(post_meas1, post_meas2),
  Diff = Post - Pre,
  Group = rep(c("A", "B"), each=n)
)

d <- effsize::cohen.d(measured$Diff, measured$Group,
                      paired=FALSE)$estimate

mod <- lm(Post ~ Pre + Group, data=measured)

testout <- broom::tidy(mod) %>%
  filter(term=="GroupB") %>%
  select(-term) %>%
  mutate(p.value = pt(statistic, mod$df, lower.tail = FALSE))
# Note: using a one-sided p value

out <- mutate(testout, d=d)

return(out)
}

```

Now, we set up the simulation for various scenarios.

```

difindifsimpars <- tidyrr::crossing(
  n = c( seq(5, 100, by = 5), seq(110, 200, by=10)),
  delta1 = seq(1,3, by = 0.5),

```

```

delta2 = seq(0, 2, by = 0.5),
cv_delta = c(0, 0.5),
) %>%
mutate(deltadif = delta1 - delta2) %>%
filter(delta1 > delta2)

```

And we run it

```

# if(!file.exists(paste0("../DerivedData/difindifsims_decomp_",
#                   nsims, ".rds")) || overwrite) {
#
#   pb <- progress_bar$new(total = nrow(difindifsimpars))
#
#   difindifsims <- difindifsimpars %>%
#     mutate(sim = 1:nrow(difindifsimpars)) %>%
#     group_by(sim) %>%
#     nest(params = c(n, delta1, delta2, cv_delta)) %>%
#     mutate(output = map(params, .progress = TRUE,
#                         ~{pb$tick();
#                           bind_rows(purrr::rerun(nsims,
#                                         HVPG_difindif_sim(.x$n, .x$delta1,
#                                         .x$delta2, .x$cv_delta,
#                                         decomp = 1))))})
#
#   saveRDS(difindifsims,
#           paste0("../DerivedData/difindifsims_decomp_", nsims, ".rds"))
#
# }

```

```

# if(!file.exists(paste0("../DerivedData/difindifsims_comp_",
#                   nsims, ".rds")) || overwrite) {
#
#   pb <- progress_bar$new(total = nrow(difindifsimpars))
#
#   difindifsims <- difindifsimpars %>%
#     mutate(sim = 1:nrow(difindifsimpars)) %>%
#     group_by(sim) %>%
#     nest(params = c(n, delta1, delta2, cv_delta)) %>%
#     mutate(output = map(params,
#                         ~{pb$tick();
#                           bind_rows(purrr::rerun(nsims,
#                                         HVPG_difindif_sim(.x$n, .x$delta1,
#                                         .x$delta2, .x$cv_delta,
#                                         decomp = 2))))})
#
#   saveRDS(difindifsims,
#           paste0("../DerivedData/difindifsims_comp_", nsims, ".rds"))
#
# }

```

```

# difindifsims_decomp <- readRDS(paste0("../DerivedData/difindifsims_decomp_", nsims, ".rds"))
#
# difindifsims_decomp_res <- difindifsims_decomp %>%

```

```

# ungroup() %>%
# mutate(power = map dbl(output, ~mean(.x$p.value < 0.05))) %>%
# unnest(params) %>%
# mutate(deltadif = delta1 - delta2,
#        delta1 = as.factor(delta1),
#        delta2 = as.factor(delta2),
#        deltatadif = as.factor(deltadif)) %>%
# #filter(deltadif != 0.5) %>%
# mutate(Effects = ifelse(cv_delta==0, "Homogeneous Effects",
#                         "Heterogeneous Effects")) %>%
# mutate(Effects = fct_inorder(Effects)) %>%
# mutate(decomp = trt_all$decomp[1])
#
#
# difindifsims_comp <- readRDS(paste0("../DerivedData/difindifsims_comp_", nsims, ".rds"))
#
# difindifsims_comp_res <- difindifsims_comp %>%
#   ungroup() %>%
#   mutate(power = map dbl(output, ~mean(.x$p.value < 0.05))) %>%
#   unnest(params) %>%
#   mutate(deltadif = delta1 - delta2,
#         delta1 = as.factor(delta1),
#         delta2 = as.factor(delta2),
#         deltatadif = as.factor(deltadif)) %>%
# #filter(deltadif != 0.5) %>%
# mutate(Effects = ifelse(cv_delta==0, "Homogeneous Effects",
#                         "Heterogeneous Effects")) %>%
# mutate(Effects = fct_inorder(Effects)) %>%
# mutate(decomp = trt_all$decomp[2])
#
#
# difindifsims_res <- bind_rows(difindifsims_decomp_res,
#                                 difindifsims_comp_res)

```

Now run in the cluster, and extract the results

```

difindifsimfiles <- list.files("../Cluster/", pattern = "\\\d.rds",
                                full.names = T)

a <- as.numeric(str_match(difindifsimfiles, pattern = "_comp_(\\d*)\\.rds")[,2])
which(!(1:500 %in% a))

b <- as.numeric(str_match(difindifsimfiles, pattern = "_decomp_(\\d*)\\.rds")[,2])
which(!(1:500 %in% b))

difindifsims <- tibble(file = difindifsimfiles) %>%
  mutate(decomp = ifelse(str_detect(file, "_decomp_"),
                        "Includes Decompsated",
                        "Only Compensated"),
         batch = as.numeric(str_match(file, "(\\d+).rds")[,2]),
         sim_add = 50*(batch-1)) %>%

```

```

rowwise() %>%
  mutate(data = map(file, readRDS))

difindifsims <- difindifsims %>%
  ungroup() %>%
  unnest(data) %>%
  ungroup() %>%
  unnest(params) %>%
  unnest(output)

difindifsims_res <- difindifsims %>%
  group_by(decomp, deltadif, n, delta1, delta2, cv_delta) %>%
  summarise(
    power = mean(p.value < 0.05),
    d = mean(d)
  ) %>%
  ungroup() %>%
  mutate(Effects = ifelse(cv_delta==0, "Homogeneous Effects",
                         "Heterogeneous Effects")) %>%
  mutate(delta1 = as.factor(delta1),
         delta2 = as.factor(delta2),
         deltadif = as.factor(deltadif))

saveRDS(difindifsims_res, "../DerivedData/difindifsims_res.rds")

difindifsims_res <- readRDS("../DerivedData/difindifsims_res.rds")

difindifsims_decomp_res <- filter(difindifsims_res,
                                    decomp=="Includes Decompensated")

difindifsims_comp_res <- filter(difindifsims_res,
                                 decomp=="Only Compensated")

```

Plotting

Here we have the size of the effect of the better intervention as the

```

difindifplot_decomp <-
  ggplot(difindifsims_decomp_res, aes(x=n, y=power, colour=delta2)) +
  geom_point() +
  geom_line() +
  facet_grid(delta1~Effects) +
  coord_cartesian(ylim=c(0.5, 1)) +
  annotate("rect", ymin = 0.8, ymax = 1, xmin=0, xmax=200,
           alpha = .4, fill="grey") +
  scale_color_brewer(type = "qual", palette = 2) +
  labs(y="Power", x="Sample Size",
       colour="Reference\nIntervention\nEffect (mmHg)",
       title=trt_all$decomp[1]) +
  theme(plot.title = element_text(hjust = 0.5))

difindif_legend <- cowplot::get_legend(difindifplot_decomp)

```

```

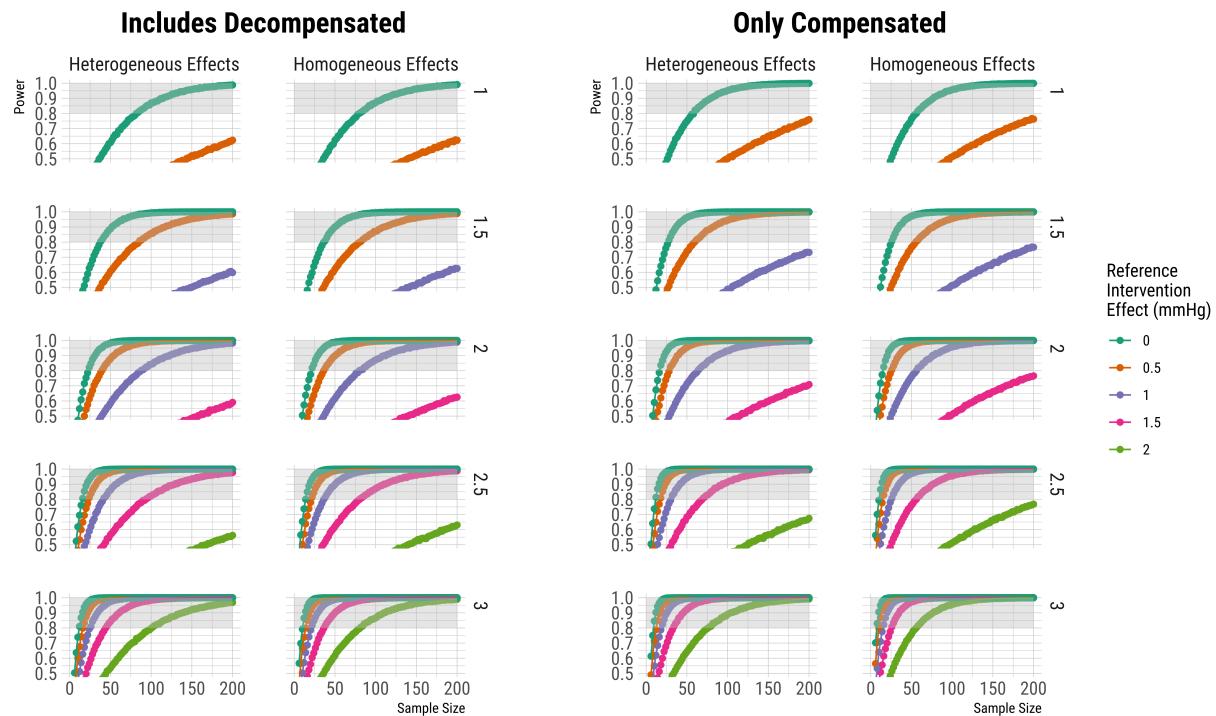
difindifplot_decomp <- difindifplot_decomp +
  guides(colour=FALSE)

difindifplot_comp <-
  ggplot(difindifsims_comp_res, aes(x=n, y=power, colour=delta2)) +
  geom_point() +
  geom_line() +
  facet_grid(delta1~Effects) +
  coord_cartesian(ylim=c(0.5, 1)) +
  annotate("rect", ymin = 0.8, ymax = 1, xmin=0, xmax=200,
           alpha = .4, fill="grey") +
  scale_color_brewer(type = "qual", palette = 2) +
  labs(y="Power", x="Sample Size",
       colour="Reference\nIntervention\nEffect (mmHg)",
       title=trt_all$decomp[2]) +
  theme(plot.title = element_text(hjust = 0.5)) +
  guides(colour=FALSE)

difindifplot <- cowplot::plot_grid(
  difindifplot_decomp, difindifplot_comp, difindif_legend,
  nrow = 1, rel_widths = c(3,3,0.5)
)

difindifplot

```



Required Individuals

And how many people do we need for each scenario?

```
difindifsims_80power <- difindifsims_res %>%
  arrange(power) %>%
  filter(power > 0.8) %>%
  select(decomp, deltadif, delta1, delta2, Effects, n) %>%
  group_by(delta1, deltadif, Effects, decomp) %>%
  slice(1) %>%
  arrange(decomp, desc(deltadif), desc(delta1), Effects) %>%
  rename("Patients" = decomp,
        "Intervention 1 Effect (mmHg)" = delta1,
        "Intervention 2 Effect (mmHg)" = delta2,
        "Difference (mmHg)" = deltadif,
        "80% Power" = n)

difindifsims_90power <- difindifsims_res %>%
  arrange(power) %>%
  filter(power > 0.9) %>%
  select(decomp, deltadif, delta1, delta2, Effects, n) %>%
  group_by(delta1, deltadif, Effects, decomp) %>%
  slice(1) %>%
  arrange(decomp, desc(deltadif), desc(delta1), Effects) %>%
  rename("Patients" = decomp,
        "Intervention 1 Effect (mmHg)" = delta1,
        "Intervention 2 Effect (mmHg)" = delta2,
        "Difference (mmHg)" = deltadif,
        "90% Power" = n)

difindifsims_power <- left_join(difindifsims_80power, difindifsims_90power) %>%
  mutate(`90% Power` = as.character(`90% Power`)) %>%
  mutate(`90% Power` = ifelse(is.na(`90% Power`),
                               ">200", `90% Power")) %>%
  filter(`Difference (mmHg)` != 0.5)

decomp_change <- head(
  which(
    difindifsims_power$Patients ==
    trt_all$decomp[2]), 1)

kable(difindifsims_power[,-1]) %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows(trt_all$decomp[1], 1, decomp_change-1) %>%
  pack_rows(trt_all$decomp[2], decomp_change, nrow(difindifsims_power))

difinditable_decomp <- difindifsims_power %>%
  filter(Patients=="Includes Decompensated") %>%
  ungroup() %>%
  select(-Patients) %>%
  kable() %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows("Includes Decompensated", 1, decomp_change-1)
```

```

difinditable_comp <- difindifsims_power %>%
  filter(Patients=="Only Compensated") %>%
  ungroup() %>%
  select(-Patients) %>%
  kable() %>%
  kable_styling("striped", full_width = F) %>%
  pack_rows("Only Compensated", 1, decomp_change=1)

difinditable_decomp

```

```
difinditable_comp
```

```
# save_kable(difinditable_decomp, "figures/difindifpower_dc.jpg")
# save_kable(difinditable_comp, "figures/difindifpower_c.jpg")
```

Contour Plot

First let's look at different deltadif values.

```

difindifsims_res_hom <- difindifsims_res %>%
  filter(cv_delta==0) %>%
  mutate(delta1 = as.numeric(as.character(delta1)),
         delta2 = as.numeric(as.character(delta2)),
         deltadif = as.numeric(as.character(deltadif))) %>%
  mutate(power = ifelse(power == 1, 0.99, power))

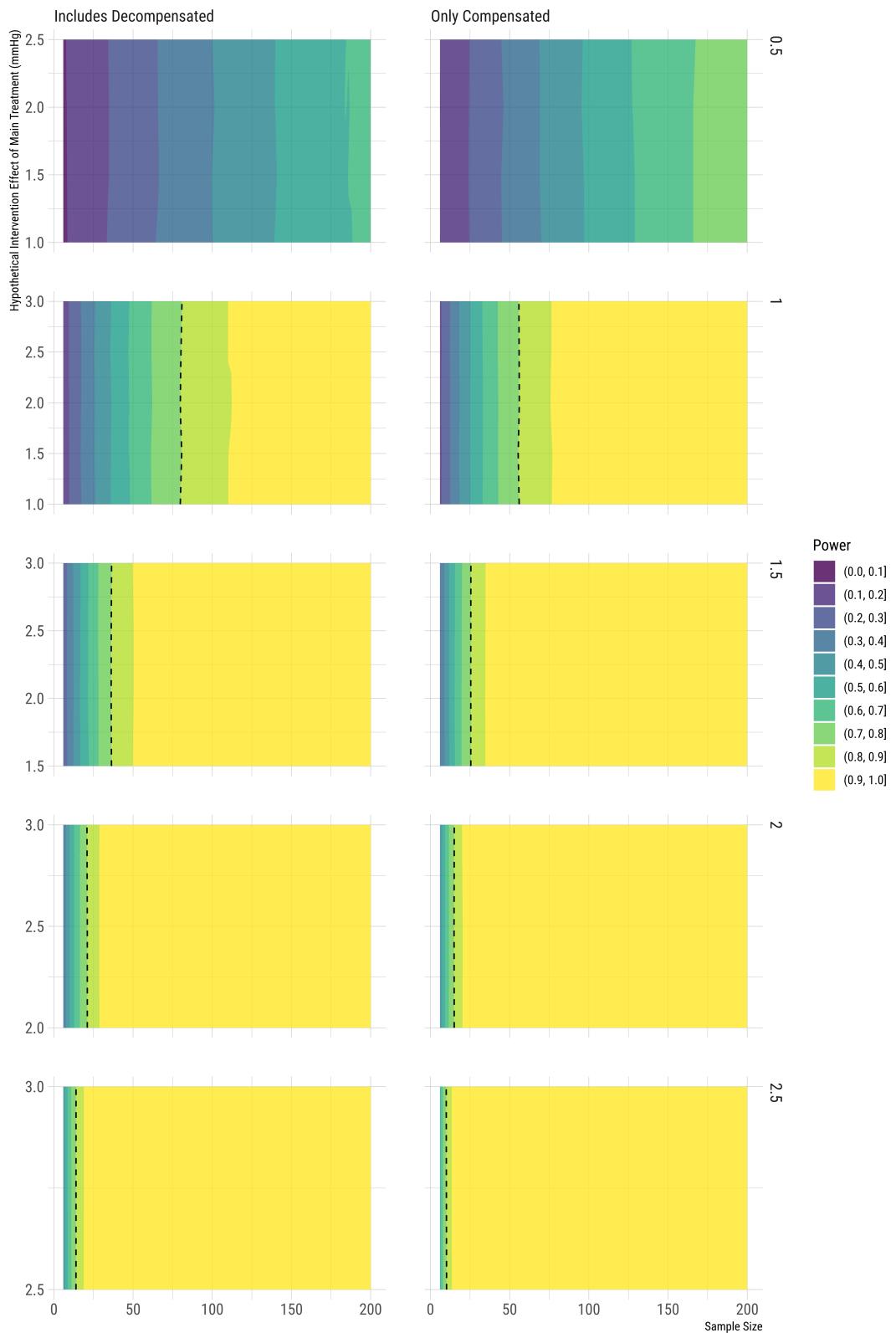
difindifsims_res_het <- difindifsims_res %>%
  filter(cv_delta!=0) %>%
  mutate(delta1 = as.numeric(as.character(delta1)),
         delta2 = as.numeric(as.character(delta2)),
         deltadif = as.numeric(as.character(deltadif))) %>%
  mutate(power = ifelse(power == 1, 0.99, power))

difindif_cont1_hom <- difindifsims_res_hom %>%
  filter(deltadif!=3) %>%
  ggplot(aes(x=n, y=delta1, z=power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_grid(deltadif~decomp, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical Intervention Effect of Main Treatment (mmHg)",
       subtitle = "Comparison of Differences of Effects between Interventions") +
  scale_y_continuous(breaks = seq(0.5, 5, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed")

difindif_cont1_hom

```

Comparison of Differences of Effects between Interventions



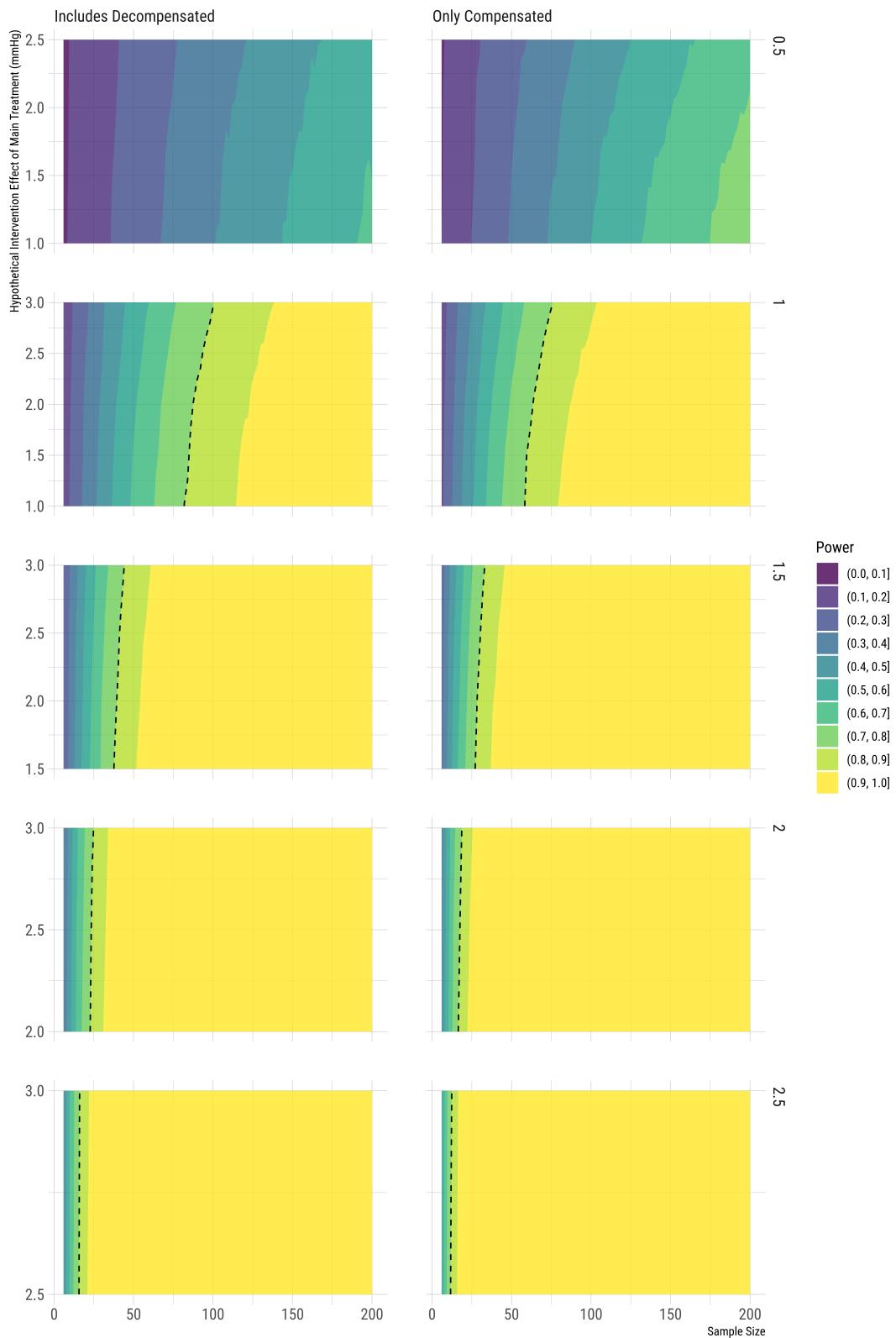
```

difindif_cont1_het <- difindifsims_res_het %>%
  filter(deltadif!=3) %>%
  ggplot(aes(x=n, y=delta1, z=power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_grid(deltadif~decomp, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical Intervention Effect of Main Treatment (mmHg)",
       subtitle = "Comparison of Differences of Effects between Interventions") +
  scale_y_continuous(breaks = seq(0.5, 5, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed")

difindif_cont1_het

```

Comparison of Differences of Effects between Interventions



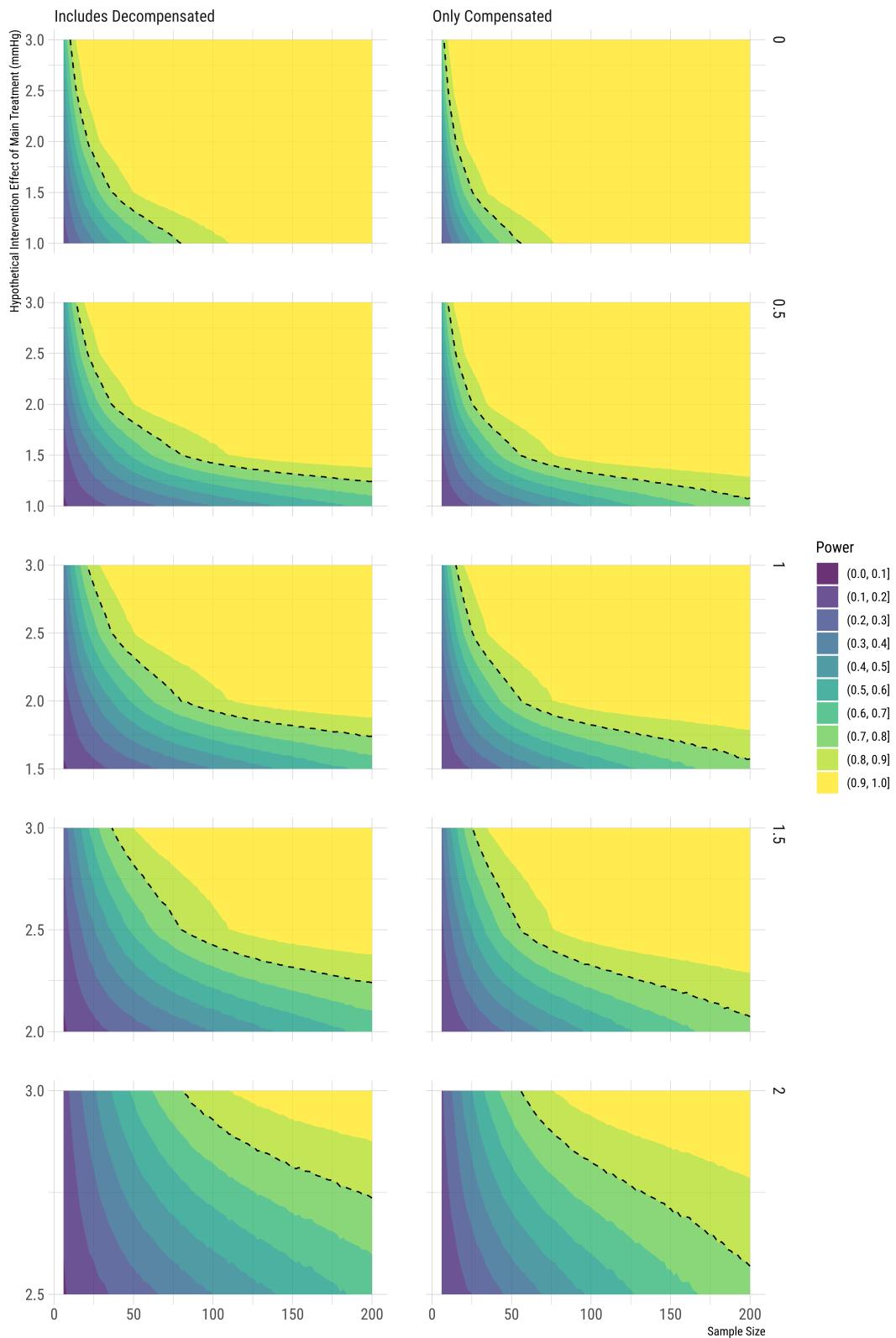
This is

helpful to visualise, though probably not for the paper. With homogeneous effects, the determinant of the power is the difference in intervention effects; it's a straight line. With heterogeneous effects, the line is not straight

```
difindif_cont2_hom <- ggplot(difindifsims_res_hom, aes(x=n, y=delta1, z=power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_grid(delta2~decomp, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical Intervention Effect of Main Treatment (mmHg)",
       subtitle = "Comparisons Grouped by Intervention Effects of the Reference Treatment (mmHg)") +
  scale_y_continuous(breaks = seq(0.5, 5, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed")

difindif_cont2_hom
```

Comparisons Grouped by Intervention Effects of the Reference Treatment (mmHg)



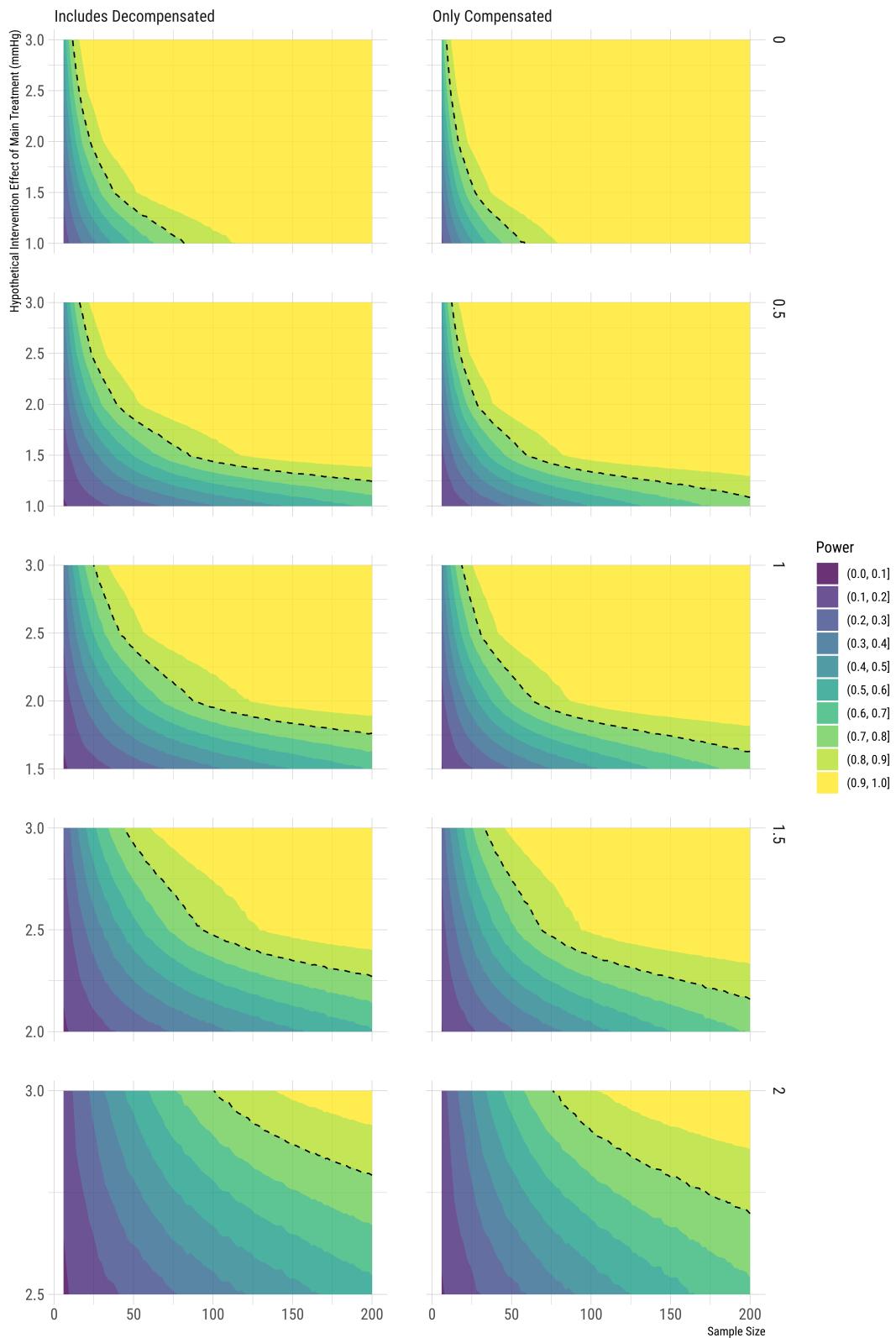
```

difindif_cont2_het <- ggplot(difindifsims_res_het, aes(x=n, y=delta1, z=power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_grid(delta2~decomp, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical Intervention Effect of Main Treatment (mmHg)",
       subtitle = "Comparisons Grouped by Intervention Effects of the Reference Treatment (mmHg)") +
  scale_y_continuous(breaks = seq(0.5, 5, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed")

difindif_cont2_het

```

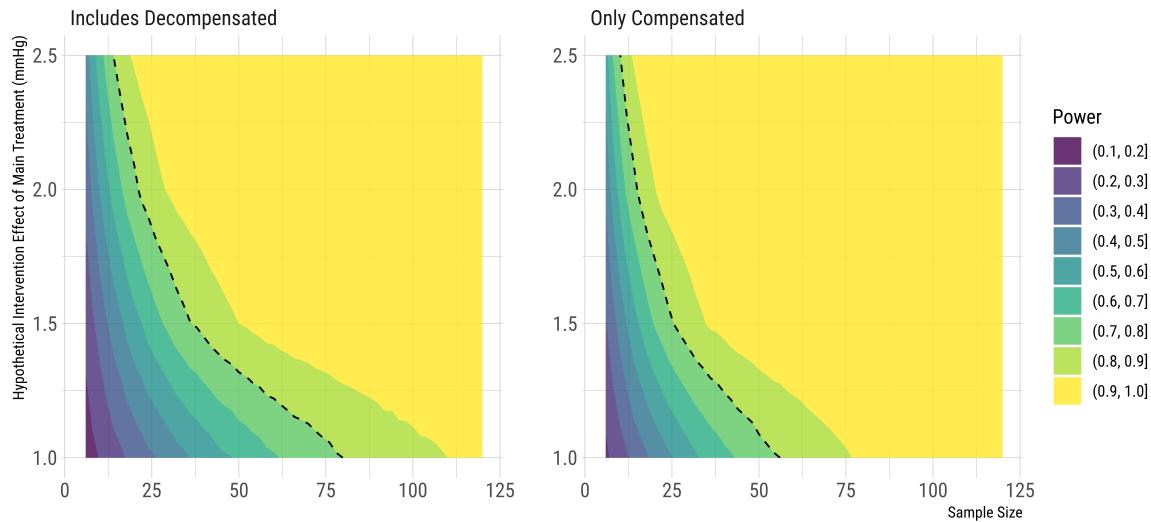
Comparisons Grouped by Intervention Effects of the Reference Treatment (mmHg)



And just looking against placebo (i.e. delta2=0)

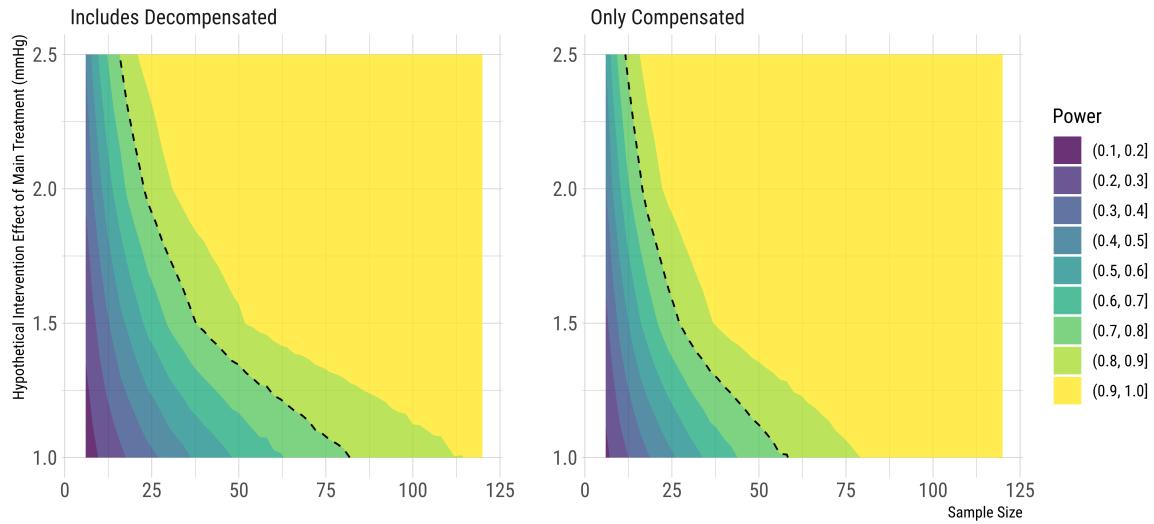
```
difindif_cont3_hom <- difindifsims_res_hom %>%
  filter(deltadif!=3) %>%
  filter(delta2==0) %>%
  ggplot(aes(x=n, y=delta1, z=power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_wrap(~decomp, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical Intervention Effect of Main Treatment (mmHg)") +
  scale_y_continuous(breaks = seq(0.5, 5, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed") +
  xlim(c(5,120))

difindif_cont3_hom
```



```
difindif_cont3_het <- difindifsims_res_het %>%
  filter(deltadif!=3) %>%
  filter(delta2==0) %>%
  ggplot(aes(x=n, y=delta1, z=power)) +
  geom_contour_filled(alpha=0.8, breaks=seq(0,1.1, by=0.1)) +
  facet_wrap(~decomp, scales = "free") +
  theme_ipsum_rc() +
  labs(x = "Sample Size",
       y = "Hypothetical Intervention Effect of Main Treatment (mmHg)") +
  scale_y_continuous(breaks = seq(0.5, 5, by = 0.5)) +
  scale_fill_viridis("Power", discrete = T) +
  geom_contour(breaks=0.8, colour="black", linetype="dashed") +
  xlim(c(5,120))

difindif_cont3_het
```



```
ggsave(difindif_cont3_hom, height=5, width=10, filename = "figures/Difindif_hom_contour.png")
ggsave(difindif_cont3_het, height=5, width=10, filename = "figures/Difindif_het_contour.png")

ggsave(difindif_cont3_hom, height=5, width=10, filename = "figures/Difindif_hom_contour.jpg",
       dpi = 600)
ggsave(difindif_cont3_het, height=5, width=10, filename = "figures/Difindif_het_contour.jpg",
       dpi = 600)
```

ICC Figures

```
colours <- c("#61b096", "#bd7969")

make_distributions <- function(icc) {

  sd_true <- 1
  var_true <- sd_true^2

  var_tot <- var_true / icc
  sd_tot <- sqrt(var_tot)

  sd_error <- sqrt(var_tot - var_true)

  distrib <- ggplot(data.frame(x=c(-5,5)), aes(x=x)) +
    stat_function(fun = dnorm,
                 colour = "black", size = 1.5, args = list(mean = 0, sd=sd_tot),
                 geom = "line") +
    stat_function(fun = dnorm,
                 colour = colours[1], size = 1, args = list(mean = 0, sd=sd_true),
                 geom = "line") +
    stat_function(fun = dnorm,
```

```

        colour = colours[2], size = 1, args = list(mean = 0, sd=sd_error),
        geom = "line") +
geom_vline(xintercept = 0, linetype="dashed") +
theme(axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      axis.text.y=element_blank(),
      axis.ticks.y=element_blank(),
      axis.text.x=element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank()) +
annotate("text", x=2.5, y=0.5,
         label="Error\nVariance",
         colour = colours[2], hjust=0.5) +
annotate("text", x=-2.5, y=0.5,
         label="True\nVariance",
         colour = colours[1], hjust=0.5) +
coord_cartesian(ylim=c(0,0.7))

return(distrib)
}

make_piecharts <- function(icc) {

  sd_true <- 1
  var_true <- sd_true^2

  var_tot <- var_true / icc
  sd_tot <- sqrt(var_tot)

  var_error <- var_tot - var_true

  var_pie <- data.frame(Variance = c("True", "Error"),
                         Values = c(var_true, var_error))
  var_pie$Variance <- forcats::fct_inorder(var_pie$Variance)

  pie <- ggplot(var_pie, aes(x="", y=Values, fill=Variance))+ 
  geom_bar(width = 1, stat = "identity") +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  coord_polar("y", start=0) +
  scale_fill_manual(values = colours) +
  labs(x="", y="") +
  guides(fill=FALSE) +
  #labs(title=paste0("ICC=", icc)) +
  NULL

  return(pie)
}

```

```
}

make_distrib_and_pie <- function(icc) {

  dist <- make_distributions(icc)
  pie <- make_piecharts(icc)

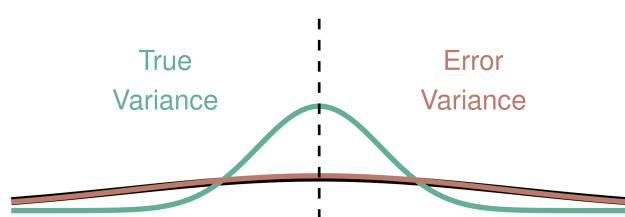
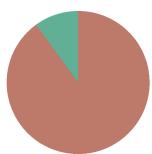
  outplot <- cowplot::plot_grid(pie, dist, rel_widths = c(1,2)) +
    draw_figure_label(paste0("ICC=", icc))

  return(outplot)
}

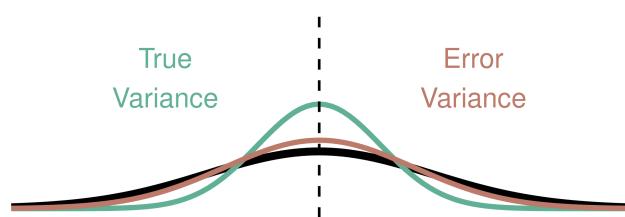
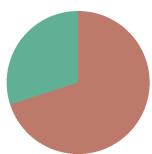
comparison_charts <- tibble(
  icc = c(0.1, 0.3, 0.7, 0.9)
) %>%
  mutate(chart = purrr::map(icc, make_distrib_and_pie))

plot_grid(plotlist = comparison_charts$chart, ncol=1)
```

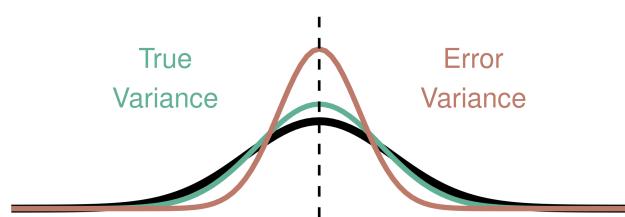
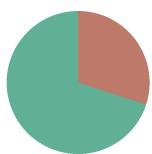
ICC=0.1



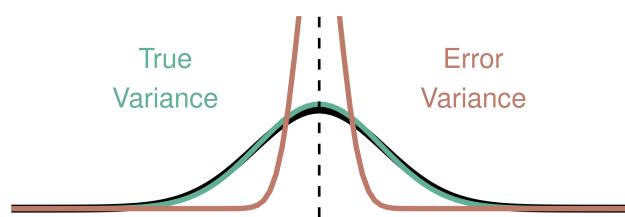
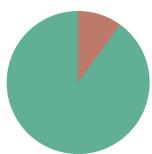
ICC=0.3



ICC=0.7



ICC=0.9



Study	n	Decompensated (%)	Alcoholic (%)	Mean Days Elapsed	Mean	CV	WSCV	ICC
Only Compensated								
Abraldes 2008 (C)	16	0.0	44.0	30.00	18.47	0.16	0.07	0.8
Albillos 1995	20	0.0	60.0	90.00	19.60	0.20	0.05	0.9
Berzigotti 2010	4	0.0	50.0	16.00	18.62	0.12	0.03	0.9
Debernardi 2007	34	0.0	21.7	365.00	14.59	0.12	0.07	0.6
Garcia-Tsao 2020 (C)	100	0.0	0.0	168.00	16.63	0.24	0.10	0.8
Hidaka 2011	38	0.0	16.7	0.01	14.75	0.25	0.12	0.7
Pozzi 2005	18	0.0	0.0	182.50	16.11	0.22	0.11	0.7
Includes Decompensated								
Abraldes 2008 (D)	36	100.0	44.0	30.00	20.28	0.22	0.10	0.8
Blei 1987	18	100.0	100.0	0.04	15.06	0.31	0.06	0.9
Fukada 2014	16	50.0	25.0	0.04	17.28	0.24	0.07	0.9
Garcia-Tsao 2020 (D)	14	100.0	0.0	168.00	16.32	0.29	0.27	0.2
Jayakumar 2013	16	100.0	75.0	56.00	22.72	0.14	0.09	0.6
Kimer 2017	36	100.0	72.2	28.00	16.18	0.28	0.17	0.6
Lebrec 2012	12	50.0	50.0	0.04	18.04	0.17	0.10	0.6
McCormick 1992	40	75.0	75.0	0.01	17.44	0.25	0.06	0.9
Merkel 2004	18	29.0	57.7	730.00	12.39	0.10	0.04	0.8
Moller 2000	16	87.5	100.0	0.02	15.38	0.35	0.07	0.9
Pomier 1987	12	100.0	62.5	172.50	15.42	0.36	0.09	0.9
Reverter 2015	42	57.0	47.6	15.00	15.76	0.28	0.06	0.9
Schepke 2001	36	100.0	72.5	7.00	18.25	0.21	0.12	0.6
Schwarzer 2017	20	80.0	80.0	28.00	20.50	0.24	0.06	0.9
Spahr 2007	16	75.0	62.5	90.00	17.69	0.17	0.16	0.1

Difference	Effects	True Worsening (%)	Apparent Worsening (%)	True 10%+ Improvement (%)	Apparent
Includes Decompensated					
0.0	Homogeneous	0.0	50.0	0.0	
0.0	Heterogeneous	0.0	50.0	0.0	
0.5	Homogeneous	0.0	42.5	0.3	
0.5	Heterogeneous	2.3	42.5	0.8	
1.0	Homogeneous	0.0	35.2	5.0	
1.0	Heterogeneous	2.3	35.4	13.7	
1.5	Homogeneous	0.0	28.5	30.1	
1.5	Heterogeneous	2.3	29.2	39.5	
2.0	Homogeneous	0.0	22.3	72.7	
2.0	Heterogeneous	2.3	23.8	59.7	
2.5	Homogeneous	0.0	17.1	95.8	
2.5	Heterogeneous	2.3	19.6	71.9	
3.0	Homogeneous	0.0	12.7	99.8	
3.0	Heterogeneous	2.3	16.1	79.1	
Only Compensated					
0.0	Homogeneous	0.0	50.0	0.0	
0.0	Heterogeneous	0.0	50.0	0.0	
0.5	Homogeneous	0.0	40.9	0.1	
0.5	Heterogeneous	2.3	41.1	0.4	
1.0	Homogeneous	0.0	32.4	3.4	
1.0	Heterogeneous	2.3	32.8	14.8	
1.5	Homogeneous	0.0	24.7	34.6	
1.5	Heterogeneous	2.3	25.8	43.3	
2.0	Homogeneous	0.0	18.1	84.7	
2.0	Heterogeneous	2.3	20.3	63.3	
2.5	Homogeneous	0.0	12.7	99.3	
2.5	Heterogeneous	2.3	16.1	74.6	
3.0	Homogeneous	0.0	8.5	100.0	
3.0	Heterogeneous	2.3	12.9	81.1	

Difference	Effects	True Worsening (%)	Apparent Worsening (%)	True 10%+ Improvement (%)	Apparent
Includes Decompensated					
0.0	Homogeneous	0.0	50.0	0.0	
0.0	Heterogeneous	0.0	50.0	0.0	
0.5	Homogeneous	0.0	42.5	0.3	
0.5	Heterogeneous	2.3	42.5	0.8	
1.0	Homogeneous	0.0	35.2	5.0	
1.0	Heterogeneous	2.3	35.4	13.7	
1.5	Homogeneous	0.0	28.5	30.1	
1.5	Heterogeneous	2.3	29.2	39.5	
2.0	Homogeneous	0.0	22.3	72.7	
2.0	Heterogeneous	2.3	23.8	59.7	
2.5	Homogeneous	0.0	17.1	95.8	
2.5	Heterogeneous	2.3	19.6	71.9	
3.0	Homogeneous	0.0	12.7	99.8	
3.0	Heterogeneous	2.3	16.1	79.1	
Only Compensated					
0.0	Homogeneous	0.0	50.0	0.0	
0.0	Heterogeneous	0.0	50.0	0.0	
0.5	Homogeneous	0.0	40.9	0.1	
0.5	Heterogeneous	2.3	41.1	0.4	
1.0	Homogeneous	0.0	32.4	3.4	
1.0	Heterogeneous	2.3	32.8	14.8	
1.5	Homogeneous	0.0	24.7	34.6	
1.5	Heterogeneous	2.3	25.8	43.3	
2.0	Homogeneous	0.0	18.1	84.7	
2.0	Heterogeneous	2.3	20.3	63.3	
2.5	Homogeneous	0.0	12.7	99.3	
2.5	Heterogeneous	2.3	16.1	74.6	
3.0	Homogeneous	0.0	8.5	100.0	
3.0	Heterogeneous	2.3	12.9	81.1	

Difference	Effects	80% Power	90% Power
Includes Decompensated			
1.0	Homogeneous Effects	45	61
1.0	Heterogeneous Effects	46	63
1.5	Homogeneous Effects	21	28
1.5	Heterogeneous Effects	22	30
2.0	Homogeneous Effects	13	17
2.0	Heterogeneous Effects	14	19
2.5	Homogeneous Effects	9	11
2.5	Heterogeneous Effects	10	14
3.0	Homogeneous Effects	7	9
3.0	Heterogeneous Effects	8	11
Only Compensated			
1.0	Homogeneous Effects	32	43
1.0	Heterogeneous Effects	33	45
1.5	Homogeneous Effects	15	20
1.5	Heterogeneous Effects	17	22
2.0	Homogeneous Effects	9	12
2.0	Heterogeneous Effects	11	14
2.5	Homogeneous Effects	7	9
2.5	Heterogeneous Effects	8	11
3.0	Homogeneous Effects	5	7
3.0	Heterogeneous Effects	7	9

Difference	Effects	80% Power	90% Power
Includes Decompensated			
1.0	Homogeneous Effects	45	61
1.0	Heterogeneous Effects	46	63
1.5	Homogeneous Effects	21	28
1.5	Heterogeneous Effects	22	30
2.0	Homogeneous Effects	13	17
2.0	Heterogeneous Effects	14	19
2.5	Homogeneous Effects	9	11
2.5	Heterogeneous Effects	10	14
3.0	Homogeneous Effects	7	9
3.0	Heterogeneous Effects	8	11
Only Compensated			
1.0	Homogeneous Effects	32	43
1.0	Heterogeneous Effects	33	45
1.5	Homogeneous Effects	15	20
1.5	Heterogeneous Effects	17	22
2.0	Homogeneous Effects	9	12
2.0	Heterogeneous Effects	11	14
2.5	Homogeneous Effects	7	9
2.5	Heterogeneous Effects	8	11
3.0	Homogeneous Effects	5	7
3.0	Heterogeneous Effects	7	9

Difference (mmHg)	Intervention 1 Effect (mmHg)	Intervention 2 Effect (mmHg)	Effects	80% Power
Includes Decompensated				
3	3	0	Heterogeneous Effects	12
3	3	0	Homogeneous Effects	12
2.5	3	0.5	Heterogeneous Effects	18
2.5	3	0.5	Homogeneous Effects	16
2.5	2.5	0	Heterogeneous Effects	16
2.5	2.5	0	Homogeneous Effects	14
2	3	1	Heterogeneous Effects	26
2	3	1	Homogeneous Effects	22
2	2.5	0.5	Heterogeneous Effects	24
2	2.5	0.5	Homogeneous Effects	22
2	2	0	Heterogeneous Effects	24
2	2	0	Homogeneous Effects	22
1.5	3	1.5	Heterogeneous Effects	46
1.5	3	1.5	Homogeneous Effects	38
1.5	2.5	1	Heterogeneous Effects	42
1.5	2.5	1	Homogeneous Effects	38
1.5	2	0.5	Heterogeneous Effects	40
1.5	2	0.5	Homogeneous Effects	38
1.5	1.5	0	Heterogeneous Effects	38
1.5	1.5	0	Homogeneous Effects	38
1	3	2	Heterogeneous Effects	102
1	3	2	Homogeneous Effects	82
1	2.5	1.5	Heterogeneous Effects	94
1	2.5	1.5	Homogeneous Effects	82
1	2	1	Heterogeneous Effects	88
1	2	1	Homogeneous Effects	80
1	1.5	0.5	Heterogeneous Effects	86
1	1.5	0.5	Homogeneous Effects	82
1	1	0	Heterogeneous Effects	82
1	1	0	Homogeneous Effects	80
Only Compensated				
3	3	0	Heterogeneous Effects	10
3	3	0	Homogeneous Effects	8
2.5	3	0.5	Heterogeneous Effects	14
2.5	3	0.5	Homogeneous Effects	10
2.5	2.5	0	Heterogeneous Effects	12
2.5	2.5	0	Homogeneous Effects	12
2	3	1	Heterogeneous Effects	20
2	3	1	Homogeneous Effects	16
2	2.5	0.5	Heterogeneous Effects	18
2	2.5	0.5	Homogeneous Effects	16
2	2	0	Heterogeneous Effects	18
2	2	0	Homogeneous Effects	16
1.5	3	1.5	Heterogeneous Effects	34
1.5	3	1.5	Homogeneous Effects	26
1.5	2.5	1	Heterogeneous Effects	32
1.5	2.5	1	Homogeneous Effects	26
1.5	2	0.5	Heterogeneous Effects	30
1.5	2	0.5	Homogeneous Effects	26
1.5	1.5	0	Heterogeneous Effects	28
1.5	1.5	0	Homogeneous Effects	26
1	3	2	Heterogeneous Effects	76
1	3	2	Homogeneous Effects	56
1	2.5	1.5	Heterogeneous Effects	70
1	2.5	1.5	Homogeneous Effects	56
1	2	1	Heterogeneous Effects	64
1	2	1	Homogeneous Effects	58

Difference (mmHg)	Intervention 1 Effect (mmHg)	Intervention 2 Effect (mmHg)	Effects	80% Power
Includes Decompensated				
3	3	0	Heterogeneous Effects	12
3	3	0	Homogeneous Effects	12
2.5	3	0.5	Heterogeneous Effects	18
2.5	3	0.5	Homogeneous Effects	16
2.5	2.5	0	Heterogeneous Effects	16
2.5	2.5	0	Homogeneous Effects	14
2	3	1	Heterogeneous Effects	26
2	3	1	Homogeneous Effects	22
2	2.5	0.5	Heterogeneous Effects	24
2	2.5	0.5	Homogeneous Effects	22
2	2	0	Heterogeneous Effects	24
2	2	0	Homogeneous Effects	22
1.5	3	1.5	Heterogeneous Effects	46
1.5	3	1.5	Homogeneous Effects	38
1.5	2.5	1	Heterogeneous Effects	42
1.5	2.5	1	Homogeneous Effects	38
1.5	2	0.5	Heterogeneous Effects	40
1.5	2	0.5	Homogeneous Effects	38
1.5	1.5	0	Heterogeneous Effects	38
1.5	1.5	0	Homogeneous Effects	38
1	3	2	Heterogeneous Effects	102
1	3	2	Homogeneous Effects	82
1	2.5	1.5	Heterogeneous Effects	94
1	2.5	1.5	Homogeneous Effects	82
1	2	1	Heterogeneous Effects	88
1	2	1	Homogeneous Effects	80
1	1.5	0.5	Heterogeneous Effects	86
1	1.5	0.5	Homogeneous Effects	82
1	1	0	Heterogeneous Effects	82
1	1	0	Homogeneous Effects	80

Difference (mmHg)	Intervention 1 Effect (mmHg)	Intervention 2 Effect (mmHg)	Effects	80% Power
Only Compensated				
3	3	0	Heterogeneous Effects	10
3	3	0	Homogeneous Effects	8
2.5	3	0.5	Heterogeneous Effects	14
2.5	3	0.5	Homogeneous Effects	10
2.5	2.5	0	Heterogeneous Effects	12
2.5	2.5	0	Homogeneous Effects	12
2	3	1	Heterogeneous Effects	20
2	3	1	Homogeneous Effects	16
2	2.5	0.5	Heterogeneous Effects	18
2	2.5	0.5	Homogeneous Effects	16
2	2	0	Heterogeneous Effects	18
2	2	0	Homogeneous Effects	16
1.5	3	1.5	Heterogeneous Effects	34
1.5	3	1.5	Homogeneous Effects	26
1.5	2.5	1	Heterogeneous Effects	32
1.5	2.5	1	Homogeneous Effects	26
1.5	2	0.5	Heterogeneous Effects	30
1.5	2	0.5	Homogeneous Effects	26
1.5	1.5	0	Heterogeneous Effects	28
1.5	1.5	0	Homogeneous Effects	26
1	3	2	Heterogeneous Effects	76
1	3	2	Homogeneous Effects	56
1	2.5	1.5	Heterogeneous Effects	70
1	2.5	1.5	Homogeneous Effects	56
1	2	1	Heterogeneous Effects	64
1	2	1	Homogeneous Effects	58
1	1.5	0.5	Heterogeneous Effects	60
1	1.5	0.5	Homogeneous Effects	56
1	1	0	Heterogeneous Effects	60
1	1	0	Homogeneous Effects	58