**## Supplemental File S7. Coding for all graphs and confidence interval statistics.**

## Title of article: **Triplication: an important component of the modern scientific method.**

## Authors: **Jeremy S.C. Clark, Kamila Rydzewska, Konrad Podsiadło**

options(timeout=1000)

options(java.parameters = "-Xmx8000m")

ipak <- function(pkg){

new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]

if (length(new.pkg))

install.packages(new.pkg, dependencies = TRUE, repos="http://cran.r-project.org")

sapply(pkg, require, character.only = TRUE)

} ## end function

packages <- c("purrr", "broom", "DescTools", "Hmisc", "XNomial", "rlist", "plyr", "cowplot", "ggtext", "ggarrow", "ggforce", "forcats", "dplyr", "data.table", "ggplot2", "ggpattern", "tidyverse")

ipak(packages) ## References given at end of document.

## cbp2 <- c("#D55E00", "#56B4E9", "#009E73", "#E69F00", "#F0E442", "#0072B2", "#CC79A7", "#000000") ## color-blind with black

cbp2 <- c("#D55E00", "#56B4E9", "#00d99e", "#009E73", "#0072B2", "#5D3A9B", "#CC79A7", "#000000") ## color-blind with black

## dbp2 <- c("solid", "twodash", "longdash", "dotdash", "dashed")

dbp2 <- c("solid", "91", "519191", "71", "5191", "13")

linetypes <- function(n) {

types <- c("55", "75", "95", "1115", "111115", "11111115", "5158", "9198", "c1c8")

return(types[seq\_len(n)])

}

dev.new()

scales::show\_col(cbp2)

##

scale\_colour\_discrete <- function(...) {

ggplot2::scale\_colour\_manual(values = cbp2, ...)

}

scale\_fill\_discrete <- function(...) {

ggplot2::scale\_fill\_manual(values = cbp2, ...)

}

theme\_jack <- function (base\_size = 16, font = "Helvetica", base\_family = "", axisColor='#999999', textColor='black') {

theme\_classic(base\_size = base\_size, base\_family = base\_family) %+replace%

theme(

plot.title = element\_text(size=15, vjust=3),

axis.text.x = element\_text(colour = "black", family="Times", face=c('bold'), size = 18, vjust = 0.2),

axis.text.y = element\_text(colour = "black", family="Times", face=c('bold'), size = 18),

axis.title.x = element\_text(colour = "black", family="Times", face=c('bold'), vjust = -4, size = 20),

axis.title.y = element\_text(colour = "black", family="Times", angle=90, face=c('bold'), vjust= 5, size = 20),

panel.grid.minor = element\_blank(),

panel.grid.major = element\_blank(),

plot.background = element\_rect(fill="white", color = "white"),

panel.border = element\_blank(),

panel.background = element\_blank(),

plot.margin=unit(c(1,1.5,1.3,1.3),"cm")

)

}

theme\_set(theme\_jack())

## THE FOLLOWING CHANGES ggplot function !! adds 10% to y-axis. To revert to standard use ggplot <- function(...) ggplot2::ggplot(...) or ggplot2::ggplot(content). MUST define xxx and yyy in loop before using ggplot

ggplotFB <- function(...) ggplot2:::ggplot(...) + {if (all(xxx >= 0, na.rm = TRUE)) scale\_x\_continuous(expand = c(0, 0), limits = c(0, NA))} + {if (all(yyy >= 0, na.rm = TRUE)) scale\_y\_continuous(expand = c(0, 0), limits = c(0, NA))} + {if ((all(xxx >= 0, na.rm = TRUE)) & (all(yyy >= 0, na.rm = TRUE))) coord\_cartesian(xlim = c(0, max(xxx, na.rm = TRUE)), ylim = c(0, max(yyy, na.rm = TRUE) + max(yyy, na.rm = TRUE)/10))}

integer\_breaks <- function(n = 5, ...) {

fxn <- function(x) {

breaks <- floor(pretty(x, n, ...))

names(breaks) <- attr(breaks, "labels")

breaks

}

return(fxn)

}

int\_breaks\_drop\_zero <- function(n = 5, ...) { ## use -breaks=int\_breaks\_drop\_zero(n = 7))

fxn <- function(x) {

breaks <- floor(pretty(x, n, min.n = 0, ...))

breaks <- breaks[breaks != 0]

if (length(breaks) == 0) return(NULL)

names(breaks) <- attr(pretty(x, n, min.n = 0, ...), "labels")

breaks

}

return(fxn)

}

mypath <- "/Users/jsc/Documents/aaJ Clark MACBOOK 2025 PUM/9991 SUBMITTED R3/"

## mypath <- "C:\\Users\\jeremy.clark\\Documents\\aaJ Clark MACBOOK 2024 PUM\\999 SUBMITTED R3\\"

## **Figure 0.** Science type.

set.seed(567)

A <- c(13, rep(22, 3), rep(40, 4), rep(55.5, 2), 19.0, 39.5, 35.7, 60, 70)

x1 <- rep("A", length(A))

B <- c(20, rep(40, 2), rep(50, 3), rep(60, 3), rnorm(5, 80, 15), rep(80, 4), rep(90, 3), 100)

x2 <- rep("B", length(B))

df1 <- data.frame(x1, A)

colnames(df1) <- c("x", "y")

df2 <- data.frame(x2, B)

colnames(df2) <- c("x", "y")

dfType <- rbind(df1, df2)

dfType

plotType <- ggplot(data = dfType, aes(x = x, y = y)) + geom\_count(color = cbp2[[5]], na.rm = TRUE) + scale\_size\_area(max\_size = 15) +

xlab("Categorically distinct groups") + ylab(str\_wrap("Continuous (or ordinal) main-effect parameter", width = 25)) + scale\_y\_continuous(breaks = c(0, 25, 50, 75, 100, 120), labels = c("", "", "", "", "", ""), expand = c(0, 0), limits = c(0, 120)) +

theme(legend.position = "none")

dev.new()

plotType

ggsave("Figure0\_Science\_type.pdf", plot = plotType, device = "pdf", width = 6, height = 5, units = "in", dpi = 300, path = mypath)

collated <- fread(paste0(mypath, "S9\_Supplemental\_File\_S9\_Collated.csv"), header = TRUE)

collated <- as.data.frame(collated)

collated <- collated[1:80, ]

conclusion <- collated$Conclusion

table (conclusion)

## > table (conclusion)

## conclusion

## Double result (repetition) protocol

## 11

## Global (duplicated-methods) protocol

## 4

## Global (replication-error) protocol (s.d.)

## 6

## Global (replication-error) protocol (s.e.)

## 3

## Global (replication-error) protocol (variance)

## 1

## Global (triplicated-methods) protocol

## 33

## Non-replicated protocol

## 10

## Triple result (repetition) protocol

## 12

##

nrow(collated)

collated$con2 <- collated$Conclusion

collated$con2[collated$con2 == "Global (replication-error) protocol (s.d.)"] <- "Global (replication-error) protocol"

collated$con2[collated$con2 == "Global (replication-error) protocol (s.e.)"] <- "Global (replication-error) protocol"

collated$con2[collated$con2 == "Global (replication-error) protocol (variance)"] <- "Global (replication-error) protocol"

df <- as.data.frame(table(collated$con2))

df

df$group <- c("D", "G2", "G3e", "G3p", "NoR", "T")

neworder <- c("NoR", "G2", "D", "G3p", "G3e", "T")

df <- df %>% slice(match(neworder, group))

colnames(df) <- c("Var1", "Frequency", "group")

df$Proportion <- df$Frequency /80

df$Percentage <- paste0(round(df$Proportion \* 100, digits = 0), "%")

df$Percentage <- factor(df$Percentage, levels = unique(df$Percentage))

df$group <- factor(df$group, levels = c("NoR", "G2", "D", "G3e", "G3p", "T")) ## prevents alphabetical ordering by ggplot !

df$Protocol <- c("NoR", "G2", "D", "G3", "G3", "T")

df$Protocol <- factor(df$Protocol, levels = unique(df$Protocol))

## df$position <- c(1, 1, 2, 1, 2, 1)

## df$position <- factor(df$position, levels=c(1, 1, 2, 1, 2, 1))

## > df

## Var1 Frequency group Proportion Percentage Protocol

## 1 Non-replicated protocol 10 NoR 0.1250 12% NoR

## 2 Global (duplicated-methods) protocol 4 G2 0.0500 5% G2

## 3 Double result (repetition) protocol 11 D 0.1375 14% D

## 4 Global (triplicated-methods) protocol 33 G3p 0.4125 41% G3

## 5 Global (replication-error) protocol 10 G3e 0.1250 12% G3

## 6 Triple result (repetition) protocol 12 T 0.1500 15% T

##

##

cbp2B <- c("#56B4E9", "#56B4E9", "#56B4E9", "#56B4E9", "#0072B2", "#56B4E9")

names(cbp2B) <- df$group # Named vector for proper mapping

myPattern <- c("none", "none", "stripe", "none", "stripe", "stripe")

names(myPattern) <- df$group

myAngle <- c(45, 135, 45, 135, 0, 45)

names(myAngle) <- df$group

legendPattern <- c("none", "none", "stripe", "none", "stripe", "stripe")

names(legendPattern) <- df$group

Patterncolor <- c("#56B4E9", "#56B4E9", "#56B4E9", "#56B4E9", "#0072B2", "#56B4E9")

names(Patterncolor) <- df$group

myVjust <- c(4, 2, 4.5, 6, 4, 5.2)

legend\_groups <- c("G3p", "D", "G3e")

myLabels <- c("NoR or Global", "Result reps", "G3e")

plotProtocols <- ggplot(data = df, aes(x = Protocol, y = Proportion, fill = group, pattern = group)) + geom\_bar\_pattern(

stat = "identity", position = "stack", color = "black",

pattern\_fill = "black", pattern\_color = Patterncolor,

pattern\_angle = myAngle, pattern = myPattern

) +

scale\_fill\_manual(values = cbp2B, breaks = legend\_groups, labels = myLabels) +

scale\_pattern\_manual(values = myPattern, breaks = legend\_groups) +

guides(fill = guide\_legend(override.aes = list(pattern = legendPattern[legend\_groups], pattern\_angle = myAngle[legend\_groups], pattern\_color = c("#56B4E9", "#56B4E9", "#0072B2"))), pattern = guide\_legend(override.aes = list(pattern = legendPattern[legend\_groups], pattern\_angle = myAngle[legend\_groups], pattern\_color = "#0072B2"))) + labs(fill = "") + xlab("Protocol") + ylab("Prop. of studies (%)") +

scale\_y\_continuous(breaks = c(0, 0.2, 0.4, 0.6), labels = c("0", "20", "40", "60"), expand = c(0, 0), limits = c(0, 0.6)) + geom\_text(aes(label = Percentage), position = position\_stack(), vjust =myVjust, color="white", size=3.5) + theme(legend.position = "inside", legend.position.inside = c(0.25, 0.8))

## graphics.off()

## dev.off()

## dev.new(width=7, height=5)

## grid::grid.newpage(recording = TRUE)

plotProtocols

ggsave("FigureAA\_protocol\_props.pdf", plot = plotProtocols, device = "pdf", width = 6, height = 5, units = "in", dpi = 300, path = mypath)

##

collated$Num\_reps <- ifelse((grepl("Yes", collated$More\_or\_not)) & (!grepl("4", collated$More\_or\_not)), "Yes", ifelse(grepl("4", collated$More\_or\_not), "4", ifelse(grepl("no", collated$More\_or\_not), "no", NA)))

collated$exact <- collated$con2

collated$exact[(collated$Num\_reps == "4") & (collated$con2 == "Global (triplicated-methods) protocol")] <- "G4"

collated$exact[(collated$Num\_reps == "4") & (collated$con2 == "Triple result (repetition) protocol")] <- "T4"

collated$exact[(collated$Num\_reps == "Yes") & (collated$con2 == "Global (triplicated-methods) protocol")] <- "G5+"

collated$exact[(collated$Num\_reps == "Yes") & (collated$con2 == "Triple result (repetition) protocol")] <- "T5+"

collated$exact

collated$exactcode <- collated$exact

collated$exactcode[collated$exactcode == "Triple result (repetition) protocol"] <- "T"

collated$exactcode[collated$exactcode == "Double result (repetition) protocol"] <- "D"

collated$exactcode[collated$exactcode == "Global (replication-error) protocol"] <- "G3e"

collated$exactcode[collated$exactcode == "Global (triplicated-methods) protocol"] <- "G3p"

collated$exactcode[collated$exactcode == "Non-replicated protocol"] <- "NoR"

collated$exactcode[collated$exactcode == "Global (duplicated-methods) protocol"] <- "G2"

collated$exactcode <- factor(collated$exactcode, levels = c("NoR", "G2", "D", "G3e", "G3p", "T", "G4", "T4", "G5+", "T5+")) ## prevents alphabetical ordering by ggplot !

mygroup <- c("NoR", "G2", "D", "G3", "T", "G4", "T4", "G5+", "T5+")

dfexact <- as.data.frame(table(collated$exactcode))

dfexact

colnames(dfexact) <- c("Protocol", "Frequency")

dfexact$Proportion <- dfexact$Frequency /80

dfexact$Percentage <- paste0(round(dfexact$Proportion \* 100, digits = 0), "%")

dfexact$Percentage <- factor(dfexact$Percentage, levels = unique(dfexact$Percentage))

dfexact$group <- as.character(dfexact$Protocol)

dfexact$group[dfexact$group == "G3e"] <- "G3"

dfexact$group[dfexact$group == "G3p"] <- "G3"

dfexact$group <- factor(dfexact$group, levels = unique(dfexact$group))

## Test for differences between the proportions: G3e and G3p combined:

dfmultiB <- dfexact[!(dfexact$Protocol %in% "G3e") & !(dfexact$Protocol %in% "G3p"), ]

G3df <- data.frame(Protocol = "G3", Frequency = 33, Proportion = 0.1250 + 0.2875, Percentage = "41%", group = "G3")

dfmulti <- rbind(dfmultiB[1:3, ], G3df, dfmultiB[4:8, ])

dfmulti$equal <- rep(80 \* 1/nrow(dfmulti), nrow(dfmulti))

rownames(dfmulti) <- NULL

dfmulti

## Multinomial exact Monte-Carlo p-values:

## XNomial::xmonte(dfmulti$Frequency, dfmulti$equal, ntrials = 1e+07, statName = "Prob", histobins = F, histobounds = c(0, 0), showCurve = FALSE, detail = 2, safety = 1e+12)

## P value (Prob) = 0 ± 0 ## taken to be p<1e-07

## 1e+07 random trials

## Observed: 10 4 11 33 7 3 4 7 1

## Expected Ratio: 8.888889 8.888889 8.888889 8.888889 8.888889 8.888889 8.888889 8.888889 8.888889

## post-hoc binomial tests (from https://rcompanion.org/ Salvatore Mangnifico), and Wilson's confidence intervals:

dfbin <- data.frame(dfmulti$Frequency, rep(80, nrow(dfmulti)), rep(9, nrow(dfmulti)), rep(80, nrow(dfmulti)))

colnames(dfbin) <- c("Successes", "Total", "Numerator", "Denominator")

dfbin[4, "Numerator"] <- 8

dfbin

funp <- function (x){

stats::binom.test(x["Successes"], x["Total"], x["Numerator"]/x["Denominator"])$p.value

}

funest <- function (x){

DescTools::BinomCI(x["Successes"], x["Total"], conf.level = 0.95, sides = c("two.sided"), method = c("wilson"))[[1]]

}

funlwr <- function (x){

DescTools::BinomCI(x["Successes"], x["Total"], conf.level = 0.95, sides = c("two.sided"), method = c("wilson"))[[2]]

}

funupr <- function (x){

DescTools::BinomCI(x["Successes"], x["Total"], conf.level = 0.95, sides = c("two.sided"), method = c("wilson"))[[3]]

}

dfbin$p.value <- apply(dfbin, 1, funp)

dfbin$p.value <- signif(dfbin$p.value, 3)

dfbin$p.bonf <- dfbin$p.value \* 9 ## Bonferroni correction

dfbin$est <- apply(dfbin, 1, funest)

dfbin$lwr <- apply(dfbin, 1, funlwr)

dfbin$upr <- apply(dfbin, 1, funupr)

dfbin$group <- mygroup

dfbin

##Successes Total Numerator Denominator p.value p.bonf est lwr upr group

## 1 10 80 9 80 7.22e-01 6.498e+00 0.1250 0.069335576 0.2150280 NoR

## 2 4 80 9 80 7.77e-02 6.993e-01 0.0500 0.019613555 0.1216228 G2

## 3 11 80 9 80 4.77e-01 4.293e+00 0.1375 0.078547143 0.2296710 D

## 4 \*\*\* 33 80 8 80 2.66e-13 2.394e-12 0.4125 0.311058576 0.5219596 G3

## 5 7 80 9 80 5.96e-01 5.364e+00 0.0875 0.043032002 0.1697680 T

## 6 3 80 9 80 3.20e-02 2.880e-01 0.0375 0.012834569 0.1045472 G4

## 7 4 80 9 80 7.77e-02 6.993e-01 0.0500 0.019613555 0.1216228 T4

## 8 7 80 9 80 5.96e-01 5.364e+00 0.0875 0.043032002 0.1697680 G5+

## 9 \* 1 80 9 80 1.24e-03 1.116e-02 0.0125 0.002209976 0.0674627 T5+

## p \* <0.05, \*\* < 0.01, \*\*\* < 0.001

cbp2B <- c("#56B4E9", "#56B4E9", "#56B4E9", "#0072B2", "#56B4E9", "#56B4E9", "#56B4E9", "#56B4E9", "#56B4E9", "#56B4E9")

names(cbp2B) <- dfexact$Protocol

myPattern <- c("none", "none", "stripe", "stripe", "none", "stripe", "none", "stripe", "none", "stripe")

names(myPattern) <- dfexact$Protocol

myAngle <- c(45, 135, 45, 0, 135, 45, 90, 45, 90, 45)

names(myAngle) <- dfexact$Protocol

legendPattern <- c("none", "none", "stripe", "stripe", "none", "stripe", "none", "stripe", "none", "stripe")

names(legendPattern) <- dfexact$Protocol

myVjust <- c(4, 2, 4.5, 6, 6, 3, 2, 2, 2, 1)

names(myVjust) <- dfexact$Protocol

legend\_groups <- c("G3p", "D", "G3e")

myLabels <- c("NoR or Global", "Result reps", "G3e")

annotation1 <- data.frame(label = c("\*\*\*", "\*"), x = c(4.3, 9.2), y = c(0.45, 0.03))

plotProtocolexact <- ggplot(data = dfexact, aes(x = group, y = Proportion, fill = Protocol, pattern = Protocol)) +

geom\_bar\_pattern(stat = "identity", position = "stack", color = "black", pattern\_fill = "black", pattern\_color = "#0072B2", pattern\_angle = myAngle, pattern = myPattern) + scale\_fill\_manual(values = cbp2B, breaks = legend\_groups, labels = myLabels) + scale\_pattern\_manual(values = myPattern, breaks = legend\_groups) + guides(

fill = guide\_legend(override.aes = list(pattern = legendPattern[legend\_groups],

pattern\_angle = myAngle[legend\_groups],

pattern\_color = "#0072B2")),

pattern = guide\_legend(override.aes = list(pattern = legendPattern[legend\_groups],

pattern\_angle = myAngle[legend\_groups],

pattern\_color = "#0072B2"))) + labs(fill = "") + xlab("Protocol") + geom\_text(aes(label = Percentage), position = position\_stack(), vjust =myVjust, color="white", size=3.5) + theme(legend.position = "inside", legend.position.inside = c(0.86, 0.85)) + xlab("Protocol") + ylab("Prop. of studies (%)") + scale\_y\_continuous(breaks = c(0, 0.1, 0.2, 0.3, 0.4, 0.5), labels = c("0", "10", "20", "30", "40", "50"), expand = c(0, 0), limits = c(0, 0.55)) +

geom\_errorbar(data = dfbin, aes(x = group, ymin = lwr, ymax = upr), inherit.aes = FALSE, width = 0.4, color = cbp2[[1]], alpha = 0.9, linewidth = 1.3) + geom\_text(aes(label = Percentage), position = position\_stack(), vjust =myVjust, color="white", size=3.5) + theme(legend.position = "inside", legend.position.inside = c(0.86, 0.85)) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), inherit.aes = FALSE, color = cbp2[8], size=5, fontface="bold")

## graphics.off()

dev.new(width=7, height=5)

## grid::grid.newpage(recording = TRUE)

plotProtocolexact

ggsave("FigureAB\_protocol\_props.pdf", plot = plotProtocolexact, device = "pdf", width = 7, height = 5, units = "in", dpi = 300, path = mypath)

##

**## Associations within each group - exact**

dfassoc <- collated %>%

group\_by(collated$exact, collated$ASSOC\_EXP) %>% count()

dfassoc <- as.data.frame(dfassoc)

colnames(dfassoc) <- c("exact", "ASSOC\_EXP", "n")

dfassoc

## exact ASSOC\_EXP n

## 1 Double result (repetition) protocol Association 1

## 2 Double result (repetition) protocol Experimental 10

## 3 G4 Experimental 3

## 4 G5+ Association 1

## 5 G5+ Experimental 6

## 6 Global (duplicated-methods) protocol Experimental 4

## 7 Global (replication-error) protocol Association 2

## 8 Global (replication-error) protocol Experimental 8

## 9 Global (triplicated-methods) protocol Association 3

## 10 Global (triplicated-methods) protocol Experimental 20

## 11 Non-replicated protocol Association 2

## 12 Non-replicated protocol Experimental 8

## 13 T4 Experimental 4

## 14 T5+ Association 1

## 15 Triple result (repetition) protocol Association 1

## 16 Triple result (repetition) protocol Experimental 6

##

dfassoc$group <- dfassoc$exact

dfassoc$group[dfassoc$group == "Double result (repetition) protocol"] <- "D"

dfassoc$group[dfassoc$group == "Global (duplicated-methods) protocol"] <- "G2"

dfassoc$group[dfassoc$group == "Global (replication-error) protocol"] <- "G3e"

dfassoc$group[dfassoc$group == "Global (triplicated-methods) protocol"] <- "G3p"

dfassoc$group[dfassoc$group == "Non-replicated protocol"] <- "NoR"

dfassoc$group[dfassoc$group == "Triple result (repetition) protocol"] <- "T"

dfassoc$group[dfassoc$group == "G4"] <- "G4+"

dfassoc$group[dfassoc$group == "G5+"] <- "G4+"

dfassoc$group[dfassoc$group == "T4"] <- "T4+"

dfassoc$group[dfassoc$group == "T5+"] <- "T4+"

dfassoc$group <- factor(dfassoc$group, levels = c("NoR","G2", "D", "G3e", "G3p", "T", "G4+", "T4+"))

dfassoc

## exact ASSOC\_EXP n group

## 1 Double result (repetition) protocol Association 1 D

## 2 Double result (repetition) protocol Experimental 10 D

## 3 G4 Experimental 3 G4+

## 4 G5+ Association 1 G4+

## 5 G5+ Experimental 6 G4+

## 6 Global (duplicated-methods) protocol Experimental 4 G2

## 7 Global (replication-error) protocol Association 2 G3e

## 8 Global (replication-error) protocol Experimental 8 G3e

## 9 Global (triplicated-methods) protocol Association 3 G3p

## 10 Global (triplicated-methods) protocol Experimental 20 G3p

## 11 Non-replicated protocol Association 2 NoR

## 12 Non-replicated protocol Experimental 8 NoR

## 13 T4 Experimental 4 T4+

## 14 T5+ Association 1 T4+

## 15 Triple result (repetition) protocol Association 1 T

## 16 Triple result (repetition) protocol Experimental 6 T

##

## remove row 3:

dfassoc <- dfassoc[-3, ]

## change n for new row 4 to 9:

dfassoc[4, "n"] <- 9

dfassoc

tab <- dfassoc %>% group\_by(ASSOC\_EXP, group) %>%

summarise(n = sum(n), .groups = "drop") %>%

pivot\_wider(names\_from = group, values\_from = n, values\_fill = 0) %>% column\_to\_rownames("ASSOC\_EXP") %>% as.matrix()

print(tab)

fisher.res <- fisher.test(tab)

## Fisher's Exact Test for Count Data

##

## data: tab

## p-value = 0.9852

## alternative hypothesis: two.sided

##

cbp2 <- c("#D55E00", "#56B4E9", "#00d99e", "#009E73", "#0072B2", "#5D3A9B", "#CC79A7", "#000000") ## color-blind with black

dfassoc$myPattern[dfassoc$ASSOC\_EXP == "Association"] <- "stripe"

dfassoc$myPattern[dfassoc$ASSOC\_EXP == "Experimental"] <- "none"

dfassoc

myPattern <- dfassoc$myPattern

names(myPattern) <- dfassoc$group

myAngle <- rep(45, 15)

names(myAngle) <- dfassoc$group

## dev.new(width=10, height=5)

## grid::grid.newpage(recording = TRUE)

plotProtocolassoc <- ggplot(data = dfassoc, aes(x = group, y = n, fill = ASSOC\_EXP, pattern = ASSOC\_EXP)) + geom\_bar\_pattern(stat = "identity", position = "stack", color = "black", pattern\_color = cbp2[1], pattern\_fill = "black", pattern\_angle = myAngle, pattern = myPattern) +

guides(fill = guide\_legend(override.aes = list(pattern = myPattern[1:2], pattern\_angle = myAngle[1:2], pattern\_color = cbp2[1:2], pattern\_key\_scale\_factor = 0.8))) + labs(fill = "") + xlab("Protocol") + ylab("Number of studies") +

scale\_y\_continuous(breaks = c(0, 5, 10, 15, 20, 25), labels = c("0", "5", "10", "15", "20", "25"), expand = c(0, 0), limits = c(0, 25)) + theme(legend.position = "inside", legend.position.inside = c(0.87, 0.85))

dev.new()

plotProtocolassoc

## brew install --cask xquartz

ggsave("FigureB\_assoc.pdf", plot = plotProtocolassoc, device = "pdf", width = 7, height = 5, units = "in", dpi = 300, path = mypath)

**## Impact factors**

collated$Impact\_Factor <- collated$Impact\_Factor\_2017

collated$Impact\_Factor[collated$Impact\_Factor < 2] <- 1

collated$Impact\_Factor[(collated$Impact\_Factor >= 2) & (collated$Impact\_Factor <3)] <- 2

collated$Impact\_Factor[(collated$Impact\_Factor >= 3) & (collated$Impact\_Factor <4)] <- 3

collated$Impact\_Factor[collated$Impact\_Factor >= 4] <- 4

collated$exactcode <- collated$exact

collated$exactcode[collated$exactcode == "G4"] <- "G4+"

collated$exactcode[collated$exactcode == "T4"] <- "T4+"

collated$exactcode[collated$exactcode == "G5+"] <- "G4+"

collated$exactcode[collated$exactcode == "T5+"] <- "T4+"

collated$exactcode[collated$exactcode == "Triple result (repetition) protocol"] <- "T"

collated$exactcode[collated$exactcode == "Double result (repetition) protocol"] <- "D"

collated$exactcode[collated$exactcode == "Global (replication-error) protocol"] <- "G3e"

collated$exactcode[collated$exactcode == "Global (triplicated-methods) protocol"] <- "G3p"

collated$exactcode[collated$exactcode == "Non-replicated protocol"] <- "NoR"

collated$exactcode[collated$exactcode == "Global (duplicated-methods) protocol"] <- "G2"

collated$exactcode <- factor(collated$exactcode, levels = c("NoR", "G2", "D", "G3e", "G3p", "T", "G4+", "T4+")) ## prevents alphabetical ordering by ggplot !

collated$exactcode <- fct\_rev(collated$exactcode)

dfimpact <- collated %>%

group\_by(collated$Impact\_Factor, collated$exactcode) %>% count()

dfimpact <- as.data.frame(dfimpact)

colnames(dfimpact) <- c("Impact\_Factor", "Protocol", "n")

dfimpact$Impact\_Factor <- as.character(dfimpact$Impact\_Factor)

dfimpact

tabimp <- dfimpact %>% group\_by(Impact\_Factor, Protocol) %>%

pivot\_wider(names\_from = Protocol, values\_from = n, values\_fill = 0) %>% column\_to\_rownames("Impact\_Factor") %>% as.matrix()

print(tabimp)

## fisher.imp <- fisher.test(tabimp, simulate.p.value=TRUE, B = 1e+07)

## Fisher's Exact Test for Count Data with simulated

## p-value (based on 1e+07 replicates)

##

## data: tabimp

## p-value = 0.0526

## alternative hypothesis: two.sided

myPattern <- rep("none", 28)

names(myPattern) <- dfimpact$Protocol

myPattern <- replace(myPattern, names(myPattern) == "NoR", "none")

myPattern <- replace(myPattern, names(myPattern) == "G2", "stripe")

myPattern <- replace(myPattern, names(myPattern) == "D", "stripe")

myPattern <- replace(myPattern, names(myPattern) == "G3e", "stripe")

myPattern <- replace(myPattern, names(myPattern) == "G3p", "none")

myPattern <- replace(myPattern, names(myPattern) == "T", "stripe")

myPattern <- replace(myPattern, names(myPattern) == "G4+", "circle")

myPattern <- replace(myPattern, names(myPattern) == "T4+", "crosshatch")

myPattern

length(myPattern)

legendPattern <- c("NoR" = "none", "G2" = "stripe", "D" = "stripe", "G3e" = "stripe", "G3p" = "none", "T" = "stripe", "G4+" = "circle", "T4+" = "crosshatch")

legendPattern <- c("crosshatch", "circle", "stripe", "none", "stripe", "stripe", "stripe", "none") ## note only 8 and reverse order

myAngle <- rep(45, 28)

names(myAngle) <- dfimpact$Protocol

myAngle <- replace(myAngle, names(myAngle) == "NoR", 45)

myAngle <- replace(myAngle, names(myAngle) == "G2", 45)

myAngle <- replace(myAngle, names(myAngle) == "D", 115)

myAngle <- replace(myAngle, names(myAngle) == "G3e", 65)

myAngle <- replace(myAngle, names(myAngle) == "G3p", 45)

myAngle <- replace(myAngle, names(myAngle) == "T", 135)

myAngle <- replace(myAngle, names(myAngle) == "G4+", 45)

myAngle <- replace(myAngle, names(myAngle) == "T4+", 45)

legendAngle <- c(45, 45, 135, 45, 65, 115, 45, 45) ## note length 8 and reverse order !

length(myAngle)

myAngle

Patterncolor <- rep("#56B4E9", 28)

names(Patterncolor) <- dfimpact$Protocol

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "NoR", cbp2[8])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "G2", cbp2[1])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "D", cbp2[2])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "G3e", cbp2[3])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "G3p", cbp2[4])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "T", cbp2[5])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "G4+", cbp2[6])

Patterncolor <- replace(Patterncolor, names(Patterncolor) == "T4+", cbp2[7])

length(Patterncolor)

Patterncolor

legendpatterncolor <- c(cbp2[7], cbp2[6], cbp2[5], cbp2[4], cbp2[3], cbp2[2], cbp2[1], cbp2[8]) ## note only 8 and reverse order

## dev.new()

plotProtocolimpact <- ggplot(data = dfimpact, aes(x = Impact\_Factor, y = n, fill = Protocol, pattern = Protocol)) + geom\_bar\_pattern(stat = "identity", position = "stack", color = "black", pattern\_color = Patterncolor, pattern\_fill = "black", pattern\_angle = myAngle, pattern = myPattern) + guides(fill = guide\_legend(override.aes = list(pattern = legendPattern, pattern\_angle = legendAngle, pattern\_color = legendpatterncolor))) + labs(fill = "") +

scale\_y\_continuous(breaks = c(0, 5, 10, 15, 20), labels = c("0", "5", "10", "15", "20"), expand = c(0, 0), limits = c(0, 20)) +

scale\_x\_discrete(labels = c("1:<2", "2:<3", "3:<4", "4+")) +

ylab("Number of studies") +

xlab("Impact factor")

plot(plotProtocolimpact)

ggsave("FigureC\_impact.pdf", plot = plotProtocolimpact, device = "pdf", width = 6, height = 5, units = "in", dpi = 300, path = mypath)

**## Figure 5 - confidence intervals**

**## no systematic error**

## To find the confidence interval width around a global (combined) mean from several (k) samples of the same size with equal variance (s2). Mean values do not change the CI\_width. ## With equal variances the ratios of CI\_widths with two k values is not dependent on s.

## parameters for all (including plus systematic errors):

conf\_level <- 0.95

alpha <- 1 - conf\_level

k <- c(1, 2, 3, 4) ## number of groups

n <- 100 ## number in each group; larger n decreases differences between k2,k3,k4 (slightly !)

s <- 10 ## this is arbitrary ! changes CI\_width but doesn't change ratios of CI\_widths ## all groups have this s - this can have different values and will still give the same graph !

M <- 10 ## typical group mean - this can have different values and will still give the same graph ! used where systematic errors are added

SE <- s / sqrt(k \* n)

df <- k \* (n - 1) ## for t distribution

dist <- "normal" ## "normal" or "t" or "log\_normal"

myquantiles <- function (dist, df = NULL, alpha = 0.05) {

if (dist == "normal") {

quantiles <- qnorm(1 - alpha/2)

}

if (dist == "t") {

quantiles <- qt(1 - alpha/2, df)

}

if (dist == "log\_normal") {

quantiles <- qlnorm(1 - alpha/2)

}

return(quantiles)

}

mycrit <- function (dist, df = NULL, alpha = 0.05) {

if (dist == "normal") {

crit <- qnorm(1 - alpha/2)

}

if (dist == "t") {

crit <- qt(1 - alpha/2, df)

}

if (dist == "log\_normal") {

crit <- qlnorm(1 - alpha/2)

}

return(crit)

}

vals <- myquantiles(dist = dist, df = df, alpha = 0.05) ## df not used if normal

vals

CI\_width\_nosys\_nonAdj <- 2 \* vals \* SE

## 32.00848 25.85172 22.26758

CI\_width\_nosys <- 100 \* (CI\_width\_nosys\_nonAdj / CI\_width\_nosys\_nonAdj[[1]])

CI\_width\_nosys

## CI\_width\_nosys[[1]] / CI\_width\_nosys[[2]]

## CI\_width\_nosys[[2]] / CI\_width\_nosys[[3]]

## (CI\_width\_nosys[[1]] / CI\_width\_nosys[[2]]) - (CI\_width\_nosys[[2]] / CI\_width\_nosys[[3]])

**## Plus Uniform systematic error with range +-% of means (Uniform ±10%** and **t-distribution ±20%** give identical scaled patterns) When systematic error **scales with 1/k​** (as is realistic in many averaging scenarios), the CI narrows with more groups - **The scaled CI width curve will be identical from k >= 2**, regardless of whether the underlying distribution is uniform, normal, or log-normal.

**## Uniform distribution:** for ±20% of M:

sys\_var <- (0.20 \* M)^2 / 3 ## The variance of a continuous uniform distribution on [a,b] is known to be ((b−a)^2)/12 [Casella & Berger, 2002]. In the case of a symmetric distribution centered at zero over [−r,r], this simplifies to (r^2)/3.

sys\_var

var\_sys\_global <- sys\_var / k ## it is the 1/k here which gives the curve

var\_sys\_global

var\_sys\_global[1] <- 0 ## assumes syst. error cannot be detected for k = 1

var\_sampling <- s^2 / (k \* n)

var\_sampling

# Calculate SE including systematic error

SE\_total <- sqrt(var\_sampling + var\_sys\_global)

SE\_total

df\_total <- k \* (n - 1)

vals <- myquantiles(dist = dist, df = df\_total, alpha = 0.05) ## df not used if normal

vals

CI\_width\_1\_nonAdj <- 2 \* vals \* SE\_total

CI\_width\_1\_nonAdj

## Adjust to CI\_width\_1[[1]] as 100

CI\_width\_1 <- 100 \* CI\_width\_1\_nonAdj / CI\_width\_1\_nonAdj[[1]]

CI\_width\_1 ## with uniform distribution systematic error

# Ratios

CI\_width\_1[1] / CI\_width\_1[2]

CI\_width\_1[2] / CI\_width\_1[3]

**## Plus t-distribution systematic error scaled to % of mean**

**##**

df\_sys <- 10 # degrees of freedom for t-distribution

systematic\_scale <- 0.15 # % of group mean

## Std dev of systematic error (scaled to ±% of mean)

sigma\_sys <- M \* systematic\_scale

## Var(t\_df) = df / (df - 2), for df > 2 ## known variance for t distribution: Johnson NL, Kotz S, Balakrishnan N. Continuous Univariate Distributions. Vol. 2. 2nd ed. New York: Wiley; 1995.

var\_sys\_per\_group <- sigma\_sys^2 \* (df\_sys / (df\_sys - 2))

# Total variance contribution from systematic error across k groups

var\_sys\_global <- var\_sys\_per\_group / k

var\_sys\_global

var\_sys\_global[1] <- 0 ## assumes syst. error cannot be detected for k = 1

var\_sampling <- s^2 / (k \* n)

var\_sampling

# --- Total Standard Error (SE\_total) ---

SE\_total <- sqrt(var\_sampling + var\_sys\_global)

SE\_total

# Degrees of freedom for t-distribution

df <- k \* (n - 1)

crit <- mycrit(dist, df = df) ## df not used if normal

crit

# Compute CI width

CI\_width\_2\_nonAdj <- 2 \* crit \* SE\_total

CI\_width\_2\_nonAdj

## Adjust to CI\_width\_2\_nonAdj[[1]] as 100:

CI\_width\_2 <- 100 \* CI\_width\_2\_nonAdj / CI\_width\_2\_nonAdj[[1]]

CI\_width\_2 ## with t distribution systematic error

CI\_width\_2[1] / CI\_width\_2[2]

CI\_width\_2[2] / CI\_width\_2[3]

CI\_width\_1[1] / CI\_width\_1[2]

CI\_width\_1[2] / CI\_width\_1[3]

**## Plus normal distribution systematic error scaled to % of mean**

systematic\_scale <- 0.15

## Std dev of systematic error (scaled to ±% of mean)

sigma\_sys <- M \* systematic\_scale

var\_sys\_per\_group <- sigma\_sys^2

# Total variance contribution from systematic error across k groups

var\_sys\_global <- var\_sys\_per\_group / k

var\_sys\_global

var\_sys\_global[1] <- 0 ## assumes syst. error cannot be detected for k = 1

var\_sampling <- s^2 / (k \* n)

var\_sampling

# --- Total Standard Error (SE\_total) ---

SE\_total <- sqrt(var\_sampling + var\_sys\_global)

SE\_total

# Degrees of freedom for t-distribution

df\_total <- k \* (n - 1)

crit <- mycrit(dist, df = df\_total) ## df not used if normal

crit

# Compute CI width

CI\_width\_3\_nonAdj <- 2 \* crit \* SE\_total

CI\_width\_3\_nonAdj

## Adjust to CI\_width\_3\_nonAdj[[1]] as 100:

CI\_width\_3 <- 100 \* CI\_width\_3\_nonAdj / CI\_width\_3\_nonAdj[[1]]

CI\_width\_3 ## with normal distribution systematic error

## For text - using only nosys values relative to k = 2 NOT k = 1 !!

nosysRelk2 <- 100 \* CI\_width\_nosys\_nonAdj / CI\_width\_nosys\_nonAdj[[2]] ## [2] !

nosysRelk2

diff1\_nosys <- nosysRelk2[1] - nosysRelk2[2]

diff2\_nosys <- nosysRelk2[2] - nosysRelk2[3]

diff3\_nosys <- nosysRelk2[3] - nosysRelk2[4]

diffs\_nosys <- c(diff1\_nosys, diff2\_nosys, diff3\_nosys)

diffs\_nosys

## 42.29623 18.51736 10.98871

## for comparison - not put onto graph:

uniRelk2 <- 100 \* CI\_width\_1\_nonAdj / CI\_width\_1\_nonAdj[[2]] ## [2] !

uniRelk2

diff1\_uni <- uniRelk2[1] - uniRelk2[2]

diff2\_uni <- uniRelk2[2] - uniRelk2[3]

diff3\_uni <- uniRelk2[3] - uniRelk2[4]

diffs\_uni <- c(diff1\_uni, diff2\_uni, diff3\_uni)

diffs\_uni

## -3.060688 18.517364 10.988709

ttRelk2 <- 100 \* CI\_width\_2\_nonAdj / CI\_width\_2\_nonAdj[[2]] ## [2] !

ttRelk2

diff1\_tt <- ttRelk2[1] - ttRelk2[2]

diff2\_tt <- ttRelk2[2] - ttRelk2[3]

diff3\_tt <- ttRelk2[3] - ttRelk2[4]

diffs\_tt <- c(diff1\_tt, diff2\_tt, diff3\_tt)

diffs\_tt

## -27.12334 18.51736 10.98871

normRelk2 <- 100 \* CI\_width\_3\_nonAdj / CI\_width\_3\_nonAdj[[2]] ## [2] !

normRelk2

diff1\_norm <- normRelk2[1] - normRelk2[2]

diff2\_norm <- normRelk2[2] - normRelk2[3]

diff3\_norm <- normRelk2[3] - normRelk2[4]

diffs\_norm <- c(diff1\_norm, diff2\_norm, diff3\_norm)

diffs\_norm

## -21.06825 18.51736 10.98871

percinc1 <- paste0(round(diff1\_nosys, 1), "%")

percinc1

percinc2 <- paste0(round(diff2\_nosys, 1), "%")

percinc2

percinc3 <- paste0(round(diff3\_nosys, 1), "%")

percinc3

CI\_width\_1 <- c(CI\_width\_1) ## with uniform distribution systematic error

CI\_width\_2 <- c(CI\_width\_2) ## with t distribution systematic error

CI\_width\_3 <- c(CI\_width\_3) ## with normal distribution systematic error

## TEMP !!

CI\_width\_1 <- c(CI\_width\_1)

CI\_width\_2 <- c(CI\_width\_2)

CI\_width\_3 <- c(CI\_width\_3)

xxx <- c("NoR", "D", "T", "T4")

xxx <- factor(xxx, levels = c("NoR", "D", "T", "T4"))

dfCI <- data.frame(xxx, CI\_width\_nosys, CI\_width\_1, CI\_width\_2, CI\_width\_3)

colnames(dfCI) <- c("Protocol", "CI\_width\_nosys", "CI\_width\_1", "CI\_width\_2", "CI\_width\_3")

cbp2 <- c("#D55E00", "#56B4E9", "#009E73", "#E69F00", "#F0E442", "#0072B2", "#CC79A7", "#000000")

mylabels <- c("No sys.", "Uniform", "Normal", "t")

mylabels <- factor(mylabels, levels = c("No sys.", "Uniform", "Normal", "t"))

ybreaks <- c(0, 50, 100, 140)

ylabels <- paste0(ybreaks)

ylabels

annotation1 <- data.frame(x = c(1.7, 2.4, 3.4), y = c(109, 102, 85), label = c("a", percinc2, percinc3))

annotation2 <- data.frame(x = c(1.3, 2.4, 3.4), y = c(81, 60, 49), label = c(percinc1, percinc2, percinc3))

annotation3 <- data.frame(x = c(1.39), y = c(121), label = c("b"))

dev.new()

plotCI <- ggplot(data = dfCI) +

geom\_point(aes(x = Protocol, y = CI\_width\_1, color = "Uniform"), size = 3) +

geom\_point(aes(x = Protocol, y = CI\_width\_2, color = "t"), size = 4, shape = 3) +

geom\_point(aes(x = Protocol, y = CI\_width\_3, color = "Normal"), size = 5, shape = 5) +

geom\_point(aes(x = Protocol, y = CI\_width\_nosys, color = "No sys."), size = 4, shape = 7) + scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 142)) + ylab("Relative C.I. width (%)") + scale\_x\_discrete(labels=c("D"="2", "T"="3", "T4"="4")) + xlab("Number of replicates (k)") + scale\_colour\_manual(name="Distribution", values = c(cbp2[8], cbp2[8], cbp2[1], cbp2[6]), breaks = mylabels, labels = mylabels) + ## breaks give correct label order

theme(legend.title = element\_text(family = "serif"), legend.position = "inside", legend.position.inside = c(0.2, 0.20), legend.box.background = element\_rect(colour = "black")) +

geom\_arrow\_segment(x = c(1, 2, 3, 10), y = c(CI\_width\_3[[1]], CI\_width\_3[[2]], CI\_width\_3[[3]], CI\_width\_3[[4]]), xend = c(2, 3, 4, 10), yend = c(CI\_width\_3[[2]], CI\_width\_3[[3]], CI\_width\_3[[4]], CI\_width\_3[[1]]), resect\_head = 2, resect\_fins = 3.5) +

geom\_arrow\_segment(x = c(1), y = 100, xend = c(1), yend = c(130), resect\_head = 5, resect\_fins = 3.5, linetype = "dashed", linewidth = 0.7) +

geom\_arrow\_segment(x = c(1, 2, 3, 10), y = c(CI\_width\_2[[1]], CI\_width\_2[[2]], CI\_width\_2[[3]], CI\_width\_2[[4]]), xend = c(2, 3, 4, 10), yend = c(CI\_width\_2[[2]], CI\_width\_2[[3]], CI\_width\_2[[4]], CI\_width\_2[[1]]), resect\_head = 2, resect\_fins = 3.5, linetype = "dashed", linewidth = 0.7) +

geom\_arrow\_segment(x = c(1, 2, 3, 10), y = c(CI\_width\_nosys[[1]], CI\_width\_nosys[[2]], CI\_width\_nosys[[3]], CI\_width\_nosys[[4]]), xend = c(2, 3, 4, 10), yend = c(CI\_width\_nosys[[2]], CI\_width\_nosys[[3]], CI\_width\_nosys[[4]], CI\_width\_nosys[[1]]), resect\_head = 2, resect\_fins = 3.5, linetype = "dashed", linewidth = 0.7) +

geom\_arrow\_segment(x = c(1, 2, 3, 10), y = c(CI\_width\_1[[1]], CI\_width\_1[[2]], CI\_width\_1[[3]], CI\_width\_1[[4]]), xend = c(2, 3, 4, 10), yend = c(CI\_width\_1[[2]], CI\_width\_1[[3]], CI\_width\_1[[4]], CI\_width\_1[[1]]), resect\_head = 2, resect\_fins = 3.5, linetype = "dashed", linewidth = 0.7) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation2, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation3, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold")

one\_slice <- data.frame(start = 0.1, end = 0.77\*pi, r0 = 0, r = 10)

p\_circle <- ggplot(one\_slice) +

ggforce::geom\_arc\_bar(aes(x0 = 0, y0 = 100, r0 = 0, r = 10, start = start, end = end),

fill = cbp2[2], alpha = 0.5, colour = cbp2[2], linewidth = 1, n = 180) +

coord\_fixed(xlim = c(0, 4), ylim = c(0, 140), clip = "off") +

theme\_void()

library(grid)

plotCIplus <- cowplot::ggdraw() +

cowplot::draw\_plot(plotCI) +

draw\_plot(p\_circle, x = -0.187, y = 0.015, width = 1, height = 1)

dev.new(width = 6, height = 7)

print(plotCIplus)

ggsave("FigureD\_CI.pdf", plot = plotCIplus, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

## ## for main parameter t distribution, n = 100:

## > diff1

## [1] 18.51736

## > diff2

## [1] 10.98871

## for main parameter t distribution, n = 1000000:

## > diff1

## [1] 18.35036

## > diff2

## [1] 10.93898

## ## for main parameter normal or log\_normal distribution:

## > diff1

## [1] 18.35034

## > diff2

## [1] 10.93898

##

**## Figure 5B - confidence intervals with true values at k=1:**

mylabels <- c("t", "Normal", "Uniform", "No sys.")

mylabels <- factor(mylabels, levels = c("t", "Normal", "Uniform", "No sys."))

trueProInc1 <- 1 + diff1\_nosys / 100

trueProInc1

trueCI <- trueProInc1\* c(CI\_width\_nosys[[2]], CI\_width\_1[[2]], CI\_width\_3[[2]], CI\_width\_2[[2]])

names(trueCI) <- c("No sys.", "Uniform", "Normal", "t")

trueCI

CI\_width\_nosystrue <- CI\_width\_nosys ## no sys.

CI\_width\_nosystrue[[1]] <- trueCI[["No sys."]]

CI\_width\_nosystrue

CI\_width\_1true <- CI\_width\_1 ## with uniform distribution systematic error

CI\_width\_1true[[1]] <- trueCI[["Uniform"]]

CI\_width\_1true

CI\_width\_2true <- CI\_width\_2 ## with t distribution systematic error

CI\_width\_2true[[1]] <- trueCI[["t"]]

CI\_width\_2true

CI\_width\_3true <- CI\_width\_3 ## with normal distribution systematic error

CI\_width\_3true[[1]] <- trueCI[["Normal"]]

CI\_width\_3true

Protocol <- dfCI[ , "Protocol"]

Protocol

dfCItrue <- data.frame(Protocol, CI\_width\_nosystrue, CI\_width\_1true, CI\_width\_2true, CI\_width\_3true)

dfCItrue

ybreaks <- c(0, 50, 100, 150, 200)

ylabels <- paste0(ybreaks)

ylabels

annotation1 <- data.frame(x = c(1.5, 2.4, 3.4), y = c(109, 94, 78), label = c("a", percinc2, percinc3))

annotation2 <- data.frame(x = c(1.3, 1.3, 2.4, 3.4), y = c(81, 140, 60, 49), label = c(percinc1, percinc1, percinc2, percinc3))

annotation3 <- data.frame(x = c(1.39), y = c(121), label = c("b"))

plotCItrue <- ggplot(data = dfCI) +

geom\_point(aes(x = Protocol, y = CI\_width\_nosys, color = "No sys."), size = 4, shape = 7) +

geom\_point(aes(x = Protocol, y = CI\_width\_1, color = "Uniform"), size = 3) +

geom\_point(aes(x = Protocol, y = CI\_width\_2, color = "t"), size = 4, shape = 3) +

geom\_point(aes(x = Protocol, y = CI\_width\_3, color = "Normal"), size = 5, shape = 5) +

geom\_point(data = dfCItrue[1, ], aes(x = Protocol, y = CI\_width\_nosystrue, color = "No sys."), size = 4, shape = 7) +

geom\_point(data = dfCItrue[1, ], aes(x = Protocol, y = CI\_width\_1true, color = "Uniform"), size = 3) +

geom\_point(data = dfCItrue[1, ], aes(x = Protocol, y = CI\_width\_2true, color = "t"), size = 4, shape = 3) +

geom\_point(data = dfCItrue[1, ], aes(x = Protocol, y = CI\_width\_3true, color = "Normal"), size = 5, shape = 5) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 200)) + ylab("Relative C.I. width (%)") + scale\_x\_discrete(labels=c("D"="2", "T"="3", "T4"="4")) + xlab("Number of replicates (k)") + scale\_colour\_manual(name="Distribution", values = c(cbp2[8], cbp2[8], cbp2[1], cbp2[6]), breaks = mylabels, labels = mylabels) + ## breaks give correct label order

theme(legend.title = element\_text(family = "serif"), legend.position = "inside", legend.position.inside = c(0.8, 0.8), legend.box.background = element\_rect(colour = "black")) + discrete\_scale("linetype", palette = linetypes) +

geom\_segment(x = c(1, 2, 3, 10), y = c(CI\_width\_2[[1]], CI\_width\_2[[2]], CI\_width\_2[[3]], CI\_width\_2[[4]]), xend = c(2, 3, 4, 10), yend = c(CI\_width\_2[[2]], CI\_width\_2[[3]], CI\_width\_2[[4]], CI\_width\_2[[1]]), linetype = "twodash", linewidth = 0.7) + **## t lines**

geom\_arrow\_segment(x = c(2, 3, 10, 20), y = c(CI\_width\_3[[2]], CI\_width\_3[[3]], CI\_width\_3[[4]], CI\_width\_3[[1]]), xend = c(3, 4, 10, 20), yend = c(CI\_width\_3[[3]], CI\_width\_3[[4]], CI\_width\_3[[1]], CI\_width\_3[[2]]), resect\_head = 2, resect\_fins = 3.5, linetype = "solid") + **## Normal arrows**

geom\_segment(x = c(1), y = c(CI\_width\_3[[1]]), xend = c(2), yend = c(CI\_width\_3[[2]]), linetype = "solid", linewidth = 1) + **## Normal line**

geom\_segment(x = c(1), y = 100, xend = c(1), yend = c(130), linetype = "dashed", linewidth = 0.7) + **## vertical arrow**

geom\_arrow\_segment(x = c(2, 3, 10, 20), y = c(CI\_width\_nosys[[2]], CI\_width\_nosys[[3]], CI\_width\_nosys[[4]], CI\_width\_nosys[[4]]), xend = c(3, 4, 10, 20), yend = c(CI\_width\_nosys[[3]], CI\_width\_nosys[[4]], CI\_width\_nosys[[1]], CI\_width\_nosys[[1]]), resect\_head = 2, resect\_fins = 3.5, linetype = "solid", linewidth = 0.7) + **## No sys. arrows**

geom\_arrow\_segment(x = c(2), y = c(CI\_width\_nosys[[2]]), xend = c(1), yend = c(CI\_width\_nosys[[1]]), resect\_head = 2, resect\_fins = 3.5, linetype = "solid", linewidth = 0.7) + **## No sys. arrows**

geom\_arrow\_segment(x = c(2, 3, 10, 20), y = c(CI\_width\_1[[2]], CI\_width\_1[[3]], CI\_width\_1[[4]], CI\_width\_1[[4]]), xend = c(3, 4, 10, 20), yend = c(CI\_width\_1[[3]], CI\_width\_1[[4]], CI\_width\_1[[1]], CI\_width\_1[[1]]), resect\_head = 2, resect\_fins = 3.5, linetype = "solid", linewidth = 0.7) + **## Uniform arrows**

geom\_segment(x = c(1), y = c(CI\_width\_1[[1]]), xend = c(2), yend = c(CI\_width\_1[[2]]), linetype = "solid", linewidth = 0.7) + **## Uniform line**

geom\_segment(data = dfCItrue, x = c(1), y = c(dfCItrue[1, "CI\_width\_2true"]), xend = c(2), yend = c(dfCItrue[2, "CI\_width\_2true"]), linetype = "twodash", linewidth = 0.7) + **## twodash line t**

geom\_arrow\_segment(data = dfCItrue, x = c(2), y = c(dfCItrue[2, "CI\_width\_1true"]), xend = c(1), yend = c(dfCItrue[1, "CI\_width\_1true"]), resect\_head = 2, resect\_fins = 3.5, linetype = "19", linewidth = 0.7) + **## dotted arrow Uniform**

geom\_arrow\_segment(data = dfCItrue, x = c(2), y = dfCItrue[2, "CI\_width\_3true"], xend = c(1), yend = c(dfCItrue[1, "CI\_width\_3true"]), resect\_head = 2, resect\_fins = 3.5, linetype = "19", linewidth = 0.7) + **## dotted arrow Normal**

geom\_segment(data = dfCItrue, x = c(2), y = 0, xend = c(2), yend = dfCItrue[2, "CI\_width\_nosystrue"], linetype = "19", linewidth = 0.7, color = cbp2[1]) + **## vertical line k=2**

geom\_segment(data = dfCItrue, x = c(3), y = 0, xend = c(3), yend = dfCItrue[3, "CI\_width\_nosystrue"], linetype = "19", linewidth = 0.7, color = cbp2[1]) + **## vertical line k=3**

geom\_segment(data = dfCItrue, x = c(4), y = 0, xend = c(4), yend = dfCItrue[4, "CI\_width\_nosystrue"], linetype = "19", linewidth = 0.7, color = cbp2[1]) + **## vertical line k=4**

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation2, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

annotate("text", x = 1.8, y = 25, label = "sqrt(1/2)", parse = TRUE, color = cbp2[1],

fontface = "bold") +

annotate("text", x = 2.8, y = 25, label = "sqrt(2/3)", parse = TRUE, color = cbp2[1],

fontface = "bold") +

annotate("text", x = 3.8, y = 25, label = "sqrt(3/4)", parse = TRUE, color = cbp2[1],

fontface = "bold")

one\_slice <- data.frame(start = 0.05, end = 0.75\*pi, r0 = 0, r = 10) ## edge angles of arc

p\_circle <- ggplot(one\_slice) +

ggforce::geom\_arc\_bar(aes(x0 = 0, y0 = 100, r0 = 0, r = 10, start = start, end = end),

fill = cbp2[2], alpha = 0.5, colour = cbp2[2], linewidth = 1, n = 180) +

coord\_fixed(xlim = c(0, 4), ylim = c(0, 200), clip = "off") +

theme\_void()

library(grid)

plotCIplus <- cowplot::ggdraw() +

cowplot::draw\_plot(plotCItrue) +

draw\_plot(p\_circle, x = -0.21, y = 0.037, width = 1, height = 1) ## point of arc adjust

plotwidth <- 7

plotheight <- 9

dev.new(width = plotwidth, height = plotheight)

print(plotCIplus)

ggsave("FigureDB\_CI.pdf", plot = plotCIplus, device = "pdf", width = plotwidth, height = plotheight, units = "in", dpi = 300, path = mypath)

**## Plus t-distribution systematic error - WITH VARYING N:**

set.seed(123)

conf\_level <- 0.95

alpha <- 1 - conf\_level

k <- c(2, 3, 4)

n <- c(2, 10, 33, 100, 333, 10000)

kdf <- data.frame(matrix(k, ncol = length(k), nrow = length(n), byrow = TRUE))

head(kdf)

ndf <- kdf

for (i in 1:length(k)) {

ndf[ , i] <- n

}

colnames(ndf) <- k

colnames(kdf) <- k

head(ndf)

head(kdf)

s <- 50 ## all groups have this s - this can have different values and will still give the same graph !

M <- 10 ## typical group mean - this can have different values and will still give the same graph !

df\_sys <- 30 # degrees of freedom for t-distribution

systematic\_scale <- 0.10 # 10% of group mean

## Std dev of systematic error (scaled to ±10% of mean)

sigma\_sys <- M \* systematic\_scale

## Var(t\_df) = df / (df - 2), for df > 2

var\_sys\_per\_group <- sigma\_sys^2 \* (df\_sys / (df\_sys - 2))

var\_sys\_global <- var\_sys\_per\_group / k

var\_sys\_global

var\_sampling <- s^2 / (kdf \* ndf)

head(var\_sampling, 100)

SE\_total <- var\_sampling

for (i in 1:nrow(SE\_total)) {

SE\_total[i, ] <- sqrt(var\_sampling[i, ] + var\_sys\_global)

}

head(SE\_total)

df\_total <- kdf \* (ndf - 1)

head(df\_total)

crit <- df\_total; ## temp for structure

for (i in 1:ncol(df\_total)) {

for (j in 1:nrow(df\_total)) {

crit[j, i] <- qt(1 - alpha/2, df\_total[j, i])

}  
}  
head(crit)

CI\_width\_n\_nonAdj <- 2 \* crit \* SE\_total

head(CI\_width\_n\_nonAdj)

## Adjust to CI\_width\_n\_nonAdj[ , 1] as 100:

CI\_width\_n <- CI\_width\_n\_nonAdj

for (i in 1:ncol(CI\_width\_n)) {

CI\_width\_n[ , i] <- CI\_width\_n\_nonAdj[ , i] / CI\_width\_n\_nonAdj[ , 1]

}

head(CI\_width\_n)

CI\_width\_n <- CI\_width\_n \* 100

head(CI\_width\_n)

CI\_width\_n <- as.data.frame(t(CI\_width\_n))

colnames(CI\_width\_n) <- n

CI\_width\_n <- cbind(k, CI\_width\_n)

head(CI\_width\_n)

n\_char <- as.character(n)

n\_char <- factor(n\_char, levels = unique(n\_char))

mylabels <- paste0(n\_char)

mylabels[6] <- "10000 or normal"

dev.new()

plotCInA <- ggplot(data = CI\_width\_n) +

geom\_point(aes(x = k, CI\_width\_n[ , 2], color = n\_char[[1]]), size = 3, shape = 1) +

geom\_point(aes(x = k, y = CI\_width\_n[ , 3], color = n\_char[[2]]), size = 6, shape = 3) +

geom\_point(aes(x = k, y = CI\_width\_n[ , 4], color = n\_char[[3]]), size = 4, shape = 17) +

geom\_point(aes(x = k, y = CI\_width\_n[ , 5], color = n\_char[[4]]), size = 4, shape = 4) +

geom\_point(aes(x = k, y = CI\_width\_n[ , 6], color = n\_char[[5]]), size = 4, shape = 18) +

geom\_point(aes(x = k, y = CI\_width\_n[ , 7], color = n\_char[[6]]), size = 4, shape = 6) +

geom\_line(aes(x = k, y = CI\_width\_n[ , 2], color = n\_char[[1]])) +

geom\_line(aes(x = k, y = CI\_width\_n[ , 3], color = n\_char[[2]])) +

geom\_line(aes(x = k, y = CI\_width\_n[ , 4], color = n\_char[[3]])) +

geom\_line(aes(x = k, y = CI\_width\_n[ , 5], color = n\_char[[4]])) +

geom\_line(aes(x = k, y = CI\_width\_n[ , 6], color = n\_char[[5]])) +

geom\_line(aes(x = k, y = CI\_width\_n[ , 7], color = n\_char[[6]])) + scale\_x\_continuous(breaks = c(2, 3, 4), labels = c("2", "3", "4"), expand = c(0, 0), limits = c(1.8, 4.2)) + scale\_y\_continuous(breaks = c(0, 20, 40, 60, 80, 100), labels = c("0", "20", "40", "60", "80", "100"), expand = c(0, 0), limits = c(0, 105)) + ylab("Relative C.I. width (%)") + xlab("Number of replicates") +

scale\_colour\_manual(name = "Sample size, one rep", values = c(cbp2[1], cbp2[8], cbp2[7], cbp2[6], cbp2[5], cbp2[4]), limits = c(n\_char[[1]], n\_char[[2]], n\_char[[3]], n\_char[[4]], n\_char[[5]], n\_char[[6]]), labels = mylabels) + theme(legend.title = element\_text(family = "serif"), legend.position = "inside", legend.position.inside = c(0.29, 0.27), legend.box.background = element\_rect(colour = "black"))

## Add label to plot:

plotCIn <- cowplot::ggdraw(plotCInA) + draw\_figure\_label(label = "A.", position = c("top.left"), size = 25, fontface = "bold")

plot(plotCIn)

ggsave("FigureE\_CIn.pdf", plot = plotCIn, device = "pdf", width = 6, height = 6, units = "in", dpi = 300, path = mypath)

**## Plus t-distribution systematic error - Gain versus N:**

conf\_level <- 0.95

alpha <- 1 - conf\_level

k <- c(2, 3, 4)

n <- seq(from = 5, to = 335, by = 5)

kdf <- data.frame(matrix(k, ncol = length(k), nrow = length(n), byrow = TRUE))

head(kdf)

ndf <- kdf

for (i in 1:length(k)) {

ndf[ , i] <- n

}

colnames(ndf) <- k

colnames(kdf) <- k

head(ndf)

head(kdf)

s <- 50 ## all groups have this s - this can have different values and will still give the same graph !

M <- 10 ## typical group mean - this can have different values and will still give the same graph !

df\_sys <- 30 # degrees of freedom for t-distribution

systematic\_scale <- 0.10 # 10% of group mean

## Std dev of systematic error (scaled to ±10% of mean)

sigma\_sys <- M \* systematic\_scale

## Var(t\_df) = df / (df - 2), for df > 2

var\_sys\_per\_group <- sigma\_sys^2 \* (df\_sys / (df\_sys - 2))

var\_sys\_global <- var\_sys\_per\_group / k

var\_sys\_global

var\_sampling <- s^2 / (kdf \* ndf)

head(var\_sampling, 100)

SE\_total <- var\_sampling

for (i in 1:nrow(SE\_total)) {

SE\_total[i, ] <- sqrt(var\_sampling[i, ] + var\_sys\_global)

}

head(SE\_total)

df\_total <- kdf \* (ndf - 1)

head(df\_total)

crit <- df\_total;

for (i in 1:ncol(df\_total)) {

for (j in 1:nrow(df\_total)) {

crit[j, i] <- qt(1 - alpha/2, df\_total[j, i])

}  
}  
head(crit)

CI\_width\_ng\_nonAdj <- 2 \* crit \* SE\_total

head(CI\_width\_ng\_nonAdj)

## Adjust to CI\_width\_ng\_nonAdj[ , 1] as 100:

CI\_width\_ng <- CI\_width\_ng\_nonAdj

for (i in 1:ncol(CI\_width\_ng)) {

CI\_width\_ng[ , i] <- CI\_width\_ng\_nonAdj[ , i] / CI\_width\_ng\_nonAdj[ , 1]

}

head(CI\_width\_ng)

CI\_width\_ng$diff1 <- 100 \* (CI\_width\_ng[ , 1] - CI\_width\_ng[ , 2])

CI\_width\_ng$diff2 <- 100 \* (CI\_width\_ng[ , 2] - CI\_width\_ng[ , 3])

CI\_width\_ng$Gain <- 100 \* (CI\_width\_ng$diff1 - CI\_width\_ng$diff2) / CI\_width\_ng$diff2

CI\_width\_ng$n <- n

head(CI\_width\_ng)

class(CI\_width\_ng)

nrow(CI\_width\_ng)

ncol(CI\_width\_ng)

dev.new()

plotGainA <- ggplot(data = CI\_width\_ng) +

geom\_line(aes(x = n, y = Gain)) + scale\_y\_continuous(breaks = c(0, 10, 30, 50, 70, 90, 100), labels = c("0", "10", "30", "50", "70", "90", "100"), expand = c(0, 0), limits = c(0, 100)) + ylab("Gain (%)") + xlab("Sample size (n) per replicate") + coord\_cartesian(xlim = c(0, 300), clip = 'off')

plotGainA

plotGain <- cowplot::ggdraw(plotGainA) + draw\_figure\_label(label = "B.", position = c("top.left"), size = 25, fontface = "bold")

plotGain

ggsave("FigureF\_Gain.pdf", plot = plotGain, device = "pdf", width = 6, height = 6, units = "in", dpi = 300, path = mypath)

mymin <- CI\_width\_ng[CI\_width\_ng$Gain == min(CI\_width\_ng$Gain), ]

mymin

## 2 3 4 diff1 diff2 Gain n

## 67 1 0.8160029 0.7064656 18.39971 10.95372 67.97677 335

**## SYSTEMATIC ERRORS FROM DATA:**

**## REAL WORLD DATA**

## 17 Sotelo Montes Table 2 Region 1 Sp\_MOE-Tan

## Statistically significant differences between means shown - but variances not given.

## 28 Flores\_Fernandez Table 2 Undisturbed control plots - s.d. not given

## 40 Koehler emergence rates Table 4

## Statistically significant differences between means shown - but variances not given

## Brzostowski et al. 2017 Table 1 Seed yield

## Statistically significant differences between means shown - but variances not given

## Wang et al. 2017 Fig 8 D

## Results were repeated three times - but representative results shown

## Ruxton GD. The unequal variance t-test is an underused alternative to Student’s t-test and the Mann–Whitney U test. Behav Ecol. 2006;17(4):688–90. doi:10.1093/beheco/ark016

## SE is sp / sqrt(n \* k) ## SE is NOT sp / sqrt(n/k) = sp \* sqrt(k/n) which would be for analyzing differences

## The degrees of freedom for unequal variances were estimated using the Welch–Satterthwaite approximation (as noted by the International Organization for Standardization: ISO Guide to the Expression of Uncertainty in Measurement):

## M. Ballico (2000), Limitations of the Welch–Satterthwaite approximation for measurement uncertainty calculations, Metrologia, 37(1): 61–64 .

## Welch BL. The generalization of ‘Student’s’ problem when several different population variances are involved. Biometrika. 1947;34(1-2):28–35. doi:10.1093/biomet/34.1-2.28

**## First part assumes NO SYSTEMATIC ERRORS FROM DATA:**

global\_mean\_CI\_nosys <- function(means, ss, s\_method = c("sd", "SE"), ns, alpha = 0.05) {

s\_method <- match.arg(s\_method)

k <- length(means)

total\_n <- sum(ns)

weights <- ns / total\_n

global\_mean <- sum(weights \* means)

if (s\_method == "sd") {

sds <- ss

var\_total <- sum((sds^2 / ns) \* (weights^2))

var\_i <- (weights^2 \* sds^2) / ns

} else if (s\_method == "SE") {

SE <- ss

var\_total <- sum((weights^2) \* (SE^2))

var\_i <- (weights^2 \* SE^2)

}

SE\_total <- sqrt(var\_total)

df\_num <- var\_total^2

df\_den <- sum(var\_i^2 / (ns - 1)) ## Welch–Satterthwaite df correction - differing variances

df <- df\_num / df\_den

t\_crit <- qt(p = (1 - alpha/2), df)

CI\_lower <- global\_mean - t\_crit \* SE\_total

CI\_upper <- global\_mean + t\_crit \* SE\_total

CI\_width <- 2\*t\_crit \* SE\_total

CI\_expected <- qt((1-alpha/2), df = sum(ns-1)) / sqrt(k)

return(c(CI\_width = CI\_width, CI\_expected = CI\_expected))

}

## C.I. width ratios relative to k = 2

ybreaks <- c(0, 100, 200)

ylabels <- paste0(ybreaks)

ylabels

**## A. Tonk et al. 2017 Fig 1. (a) top panel Landraces TR1 to TR7**

Tonk\_label <- c("TR1", "TR2", "TR3", "TR4", "TR5", "TR6", "TR7")

Tonk\_mean <- c(34, 52, 46, 56, 39, 49, 23.5)

Tonk\_sd <- c(8.75, 15.25, 12.5, 25, 10.75, 15.25, 4.5)

Tonk\_n <- c(100, 100, 100, 100, 100, 100, 100)

Tonk\_raw\_df <- data.frame(Tonk\_label, Tonk\_mean, Tonk\_sd, Tonk\_n)

Tonk\_raw\_df

colnames(Tonk\_raw\_df) <- c("label", "mean", "sd", "n")

Tonk\_raw\_df

k <- 1:7

n <- 100

Tonk\_CIdata <- list();

for (i in 1:nrow(Tonk\_raw\_df)) {

Tonk\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Tonk\_mean[1:i], ss = Tonk\_sd[1:i], s\_method = "sd", ns = Tonk\_n[1:i])

}

Tonk\_CIdf <- as.data.frame(list.rbind(Tonk\_CIdata))

Tonk\_CIdf

Tonk\_actual\_ratio <- 100 \* signif(Tonk\_CIdf$CI\_width / Tonk\_CIdf[2, "CI\_width"], 4)

Tonk\_actual\_ratio

Tonk\_expected\_ratio <- 100 \* signif(Tonk\_CIdf$CI\_expected / Tonk\_CIdf[2, "CI\_expected"], 4)

Tonk\_expected\_ratio

Tonkdf <- data.frame(k, Tonk\_actual\_ratio, Tonk\_expected\_ratio)

colnames(Tonkdf) <- c("k", "Tonk\_Act", "Tonk\_Exp")

Tonkdf

plotRealTonk <- ggplot(data = Tonkdf) +

geom\_point(aes(x = k, y = Tonk\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Tonk\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = c(1, 3, 5, 7), expand = c(0, 0), limits = c(1, max(k) + 0.5)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Tonk\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Tonk\_Exp, color = "Expected"))

dev.new()

plotRealTonk

ggsave("FigureG\_Tonk\_real.pdf", plot = plotRealTonk, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## B. Kichigina et al. 2017 Fig 5(a) - Al sensitive genotypes**

Kich\_label <- c("3654", "8473", "2759", "1903")

Kich\_mean <- c(2.4, 3.9, 1.4, 7)

Kich\_SE <- c(0.35, 1.1, 0.15, 1.1)

Kich\_n <- c(5, 5, 5, 5)

Kich\_raw\_df <- data.frame(Kich\_label, Kich\_mean, Kich\_SE, Kich\_n)

Kich\_raw\_df

colnames(Kich\_raw\_df) <- c("label", "mean", "SE", "n")

Kich\_raw\_df

k <- 1:4

n <- 5

Kich\_CIdata <- list();

for (i in 1:nrow(Kich\_raw\_df)) {

Kich\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Kich\_mean[1:i], ss = Kich\_SE[1:i], s\_method = "SE", ns = Kich\_n[1:i])

}

Kich\_CIdf <- as.data.frame(list.rbind(Kich\_CIdata))

Kich\_CIdf

Kich\_actual\_ratio <- 100 \* signif(Kich\_CIdf$CI\_width / Kich\_CIdf[2, "CI\_width"], 4)

Kich\_actual\_ratio

Kich\_expected\_ratio <- 100 \* signif(Kich\_CIdf$CI\_expected / Kich\_CIdf[2, "CI\_expected"], 4)

Kich\_expected\_ratio

Kichdf <- data.frame(k, Kich\_actual\_ratio, Kich\_expected\_ratio)

colnames(Kichdf) <- c("k", "Kich\_Act", "Kich\_Exp")

Kichdf

plotRealKich <- ggplot(data = Kichdf) +

geom\_point(aes(x = k, y = Kich\_Act), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Kich\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = k, expand = c(0, 0), limits = c(1, max(k) + 0.2)) + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") + theme(axis.title.y = element\_blank()) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Kich\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Kich\_Exp, color = "Expected"))

dev.new()

plotRealKich

ggsave("FigureG\_Kich\_real.pdf", plot = plotRealKich, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## C. Carlucci et al. 2017 Table 5 Ilyonectria liriodendri**

Carl\_label <- c("BF144", "BF47", "BF12")

Carl\_mean <- c(8.2, 10.8, 11.8)

Carl\_sd <- c(3.1, 1.1, 1.9)

Carl\_n <- c(10, 10, 10)

Carl\_raw\_df <- data.frame(Carl\_label, Carl\_mean, Carl\_sd, Carl\_n)

Carl\_raw\_df

colnames(Carl\_raw\_df) <- c("label", "mean", "sd", "n")

Carl\_raw\_df

## N is a guess, but note the patterns of relative C.I. do not change substantially from N=2 to N=10000 in terms of Actual being above Expected for k=1 (both are shifted upwards for N=2 or downwards slightly for N=10000)

k <- 1:3

n <- 10

Carl\_CIdata <- list();

for (i in 1:nrow(Carl\_raw\_df)) {

Carl\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Carl\_mean[1:i], ss = Carl\_sd[1:i], s\_method = "sd", ns = Carl\_n[1:i])

}

Carl\_CIdf <- as.data.frame(list.rbind(Carl\_CIdata))

Carl\_CIdf

Carl\_actual\_ratio <- 100 \* signif(Carl\_CIdf$CI\_width / Carl\_CIdf[2, "CI\_width"], 4)

Carl\_actual\_ratio

Carl\_expected\_ratio <- 100 \* signif(Carl\_CIdf$CI\_expected / Carl\_CIdf[2, "CI\_expected"], 4)

Carl\_expected\_ratio

Carldf <- data.frame(k, Carl\_actual\_ratio, Carl\_expected\_ratio)

colnames(Carldf) <- c("k", "Carl\_Act", "Carl\_Exp")

Carldf

Carldf <- data.frame(k, Carl\_actual\_ratio, Carl\_expected\_ratio)

colnames(Carldf) <- c("k", "Carl\_Act", "Carl\_Exp")

Carldf

plotRealCarl <- ggplot(data = Carldf) +

geom\_point(aes(x = k, y = Carl\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Carl\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = k, expand = c(0, 0), limits = c(1, max(k) + 0.2)) + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") + theme(axis.title.y = element\_blank()) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Carl\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Carl\_Exp, color = "Expected"))

dev.new()

plotRealCarl

ggsave("FigureG\_Carl\_real.pdf", plot = plotRealCarl, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## D. Lawrence2017 - FIG 1 NATIVE**

Lawr\_label <- c("Gardener Marsh", "Pheasant Branch Slough", "Prairie Wolf")

Lawr\_mean <- c(6.2, 5.3, 14.2)

Lawr\_SE <- c(2.1, 3, 3.9)

Lawr\_n <- c(5, 5, 5)

Lawr\_raw\_df <- data.frame(Lawr\_label, Lawr\_mean, Lawr\_SE, Lawr\_n)

Lawr\_raw\_df

colnames(Lawr\_raw\_df) <- c("label", "mean", "SE", "n")

Lawr\_raw\_df

k <- 1:3

n <- 5

Lawr\_CIdata <- list();

for (i in 1:nrow(Lawr\_raw\_df)) {

Lawr\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Lawr\_mean[1:i], ss = Lawr\_SE[1:i], s\_method = "SE", ns = Lawr\_n[1:i])

}

Lawr\_CIdf <- as.data.frame(list.rbind(Lawr\_CIdata))

Lawr\_CIdf

Lawr\_actual\_ratio <- 100 \* signif(Lawr\_CIdf$CI\_width / Lawr\_CIdf[2, "CI\_width"], 4)

Lawr\_actual\_ratio

Lawr\_expected\_ratio <- 100 \* signif(Lawr\_CIdf$CI\_expected / Lawr\_CIdf[2, "CI\_expected"], 4)

Lawr\_expected\_ratio

Lawrdf <- data.frame(k, Lawr\_actual\_ratio, Lawr\_expected\_ratio)

colnames(Lawrdf) <- c("k", "Lawr\_Act", "Lawr\_Exp")

Lawrdf

plotRealLawr <- ggplot(data = Lawrdf) +

geom\_point(aes(x = k, y = Lawr\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Lawr\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = k, expand = c(0, 0), limits = c(1, max(k) + 0.2)) + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") +

geom\_line(aes(x = k, y = Lawr\_Act, color = "Actual")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Lawr\_Exp, color = "Expected"))

dev.new()

plotRealLawr

ggsave("FigureG\_Lawr\_real.pdf", plot = plotRealLawr, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## E. Zhang et al. 2017 Fig 8 A Control**

Zhan\_label <- c("Merlot", "Syrah", "Concord")

Zhan\_mean <- c(20, 23.5, 27.2)

Zhan\_SE <- c(1, 1, 1.4)

Zhan\_n <- c(10, 10, 10)

Zhan\_raw\_df <- data.frame(Zhan\_label, Zhan\_mean, Zhan\_SE, Zhan\_n)

Zhan\_raw\_df

colnames(Zhan\_raw\_df) <- c("label", "mean", "SE", "n")

Zhan\_raw\_df

k <- 1:3

n <- 10

Zhan\_CIdata <- list();

for (i in 1:nrow(Zhan\_raw\_df)) {

Zhan\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Zhan\_mean[1:i], ss = Zhan\_SE[1:i], s\_method = "SE", ns = Zhan\_n[1:i])

}

Zhan\_CIdf <- as.data.frame(list.rbind(Zhan\_CIdata))

Zhan\_CIdf

Zhan\_actual\_ratio <- 100 \* signif(Zhan\_CIdf$CI\_width / Zhan\_CIdf[2, "CI\_width"], 4)

Zhan\_actual\_ratio

Zhan\_expected\_ratio <- 100 \* signif(Zhan\_CIdf$CI\_expected / Zhan\_CIdf[2, "CI\_expected"], 4)

Zhan\_expected\_ratio

Zhan\_actual\_ratio <- c(152.3, 100, 91.62)

Zhan\_expected\_ratio <- c(152.3, 100, 79.74)

Zhandf <- data.frame(k, Zhan\_actual\_ratio, Zhan\_expected\_ratio)

colnames(Zhandf) <- c("k", "Zhan\_Act", "Zhan\_Exp")

Zhandf

plotRealZhan <- ggplot(data = Zhandf) +

geom\_point(aes(x = k, y = Zhan\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Zhan\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = k, expand = c(0, 0), limits = c(1, max(k) + 0.2)) + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") + theme(axis.title.y = element\_blank()) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Zhan\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Zhan\_Exp, color = "Expected"))

dev.new()

plotRealZhan

ggsave("FigureG\_Zhan\_real.pdf", plot = plotRealZhan, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## F. Driver et al. 2017 Fig 9B, Total PA, High P**

Driv\_label <- c("GPD3-OE1", "GPD3-OE2", "GPD3-OE3")

Driv\_mean <- c(1.8, 2.6, 3)

Driv\_SE <- c(0.4, 0.6, 1.3)

Driv\_n <- c(3, 3, 3)

Driv\_raw\_df <- data.frame(Driv\_label, Driv\_mean, Driv\_SE, Driv\_n)

Driv\_raw\_df

colnames(Driv\_raw\_df) <- c("label", "mean", "SE", "n")

Driv\_raw\_df

k <- 1:3

n <- 3

Driv\_CIdata <- list();

for (i in 1:nrow(Driv\_raw\_df)) {

Driv\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Driv\_mean[1:i], ss = Driv\_SE[1:i], s\_method = "SE", ns = Driv\_n[1:i])

}

Driv\_CIdf <- as.data.frame(list.rbind(Driv\_CIdata))

Driv\_CIdf

Driv\_actual\_ratio <- 100 \* signif(Driv\_CIdf$CI\_width / Driv\_CIdf[2, "CI\_width"], 4)

Driv\_actual\_ratio

Driv\_expected\_ratio <- 100 \* signif(Driv\_CIdf$CI\_expected / Driv\_CIdf[2, "CI\_expected"], 4)

Driv\_expected\_ratio

Drivdf <- data.frame(k, Driv\_actual\_ratio, Driv\_expected\_ratio)

colnames(Drivdf) <- c("k", "Driv\_Act", "Driv\_Exp")

Drivdf

plotRealDriv <- ggplot(data = Drivdf) +

geom\_point(aes(x = k, y = Driv\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Driv\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = k, expand = c(0, 0), limits = c(1, max(k) + 0.2)) + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") + theme(axis.title.y = element\_blank()) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Driv\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Driv\_Exp, color = "Expected"))

dev.new()

plotRealDriv

ggsave("FigureG\_Driv\_real.pdf", plot = plotRealDriv, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## G. Zhou et al. 2017 Fig 3A Relative transcription level**

Zhou\_label <- c("RG2", "RG6", "RG14", "RG20")

Zhou\_mean <- c(0.8, 0.9, 0.8, 2.1)

Zhou\_sd <- c(0.3, 0.35, 0.5, 0.5)

Zhou\_n <- c(3, 3, 3, 3)

Zhou\_raw\_df <- data.frame(Zhou\_label, Zhou\_mean, Zhou\_sd, Zhou\_n)

Zhou\_raw\_df

colnames(Zhou\_raw\_df) <- c("label", "mean", "sd", "n")

Zhou\_raw\_df

k <- 1:4

n <- 3

Zhou\_CIdata <- list();

for (i in 1:nrow(Zhou\_raw\_df)) {

Zhou\_CIdata[[i]] <- global\_mean\_CI\_nosys(means = Zhou\_mean[1:i], ss = Zhou\_sd[1:i], s\_method = "sd", ns = Zhou\_n[1:i])

}

Zhou\_CIdf <- as.data.frame(list.rbind(Zhou\_CIdata))

Zhou\_CIdf

Zhou\_actual\_ratio <- 100 \* signif(Zhou\_CIdf$CI\_width / Zhou\_CIdf[2, "CI\_width"], 4)

Zhou\_actual\_ratio

Zhou\_expected\_ratio <- 100 \* signif(Zhou\_CIdf$CI\_expected / Zhou\_CIdf[2, "CI\_expected"], 4)

Zhou\_expected\_ratio

Zhoudf <- data.frame(k, Zhou\_actual\_ratio, Zhou\_expected\_ratio)

colnames(Zhoudf) <- c("k", "Zhou\_Act", "Zhou\_Exp")

Zhoudf

plotRealZhou <- ggplot(data = Zhoudf) +

geom\_point(aes(x = k, y = Zhou\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Zhou\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = k, expand = c(0, 0), limits = c(1, max(k) + 0.2)) + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

theme(legend.position = "none") + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Zhou\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Zhou\_Exp, color = "Expected"))

dev.new()

plotRealZhou

ggsave("FigureG\_Zhou\_real.pdf", plot = plotRealZhou, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## All Real data:**

mylist <- list(Tonkdf, Kichdf, Carldf, Lawrdf, Zhandf, Drivdf, Zhoudf)

mylist

plotlist <- list(plotRealTonk, plotRealKich, plotRealCarl, plotRealLawr, plotRealZhan, plotRealDriv, plotRealZhou)

**## Grid of seven graphs:**

O <- ggplot(data = Tonkdf) +

geom\_point(aes(x = k, y = Tonk\_Act, color = "Actual"), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Tonk\_Exp, color = "Expected"), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") + scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1])) +

geom\_line(aes(x = k, y = Tonk\_Act, color = "Actual")) +

geom\_line(aes(x = k, y = Tonk\_Exp, color = "Expected"))

legend <- get\_legend(O)

graph7 <- plot\_grid(plotlist[[1]], plotlist[[2]], plotlist[[3]], plotlist[[4]], plotlist[[5]], plotlist[[6]], plotlist[[7]], NULL, ncol=3, nrow=3, align="hv", rel\_widths = c(1, 1, 1, 1, 1, 1, 1, 1, 1), labels = c('A', 'B', 'C', 'D', 'E', 'F', 'G', "", ""))

dev.new(width = 9, height = 9)

graph7legend <- graph7 + draw\_grob(legend, 2/3, 0, 1/3, 0.5)

graph7legend

ggsave("FigureI\_real.pdf", plot = graph7legend, device = "pdf", width = 9, height = 7, units = "in", dpi = 300, path = mypath)

**## Second part WITH SYSTEMATIC ERROR ESTIMATED EMPIRICALLY FROM MEANS:**

**## two methods - Range using max - min or Variance of group means**

## Bessel correction gives standard unbiased estimator of the variance of a sample of means and between-group variance corresponding to systematic variability here.

## Sokal RR, Rohlf FJ. Biometry: The Principles and Practice of Statistics in Biological Research. 4th ed. New York: W.H. Freeman; 2012.

## Montgomery DC. Design and Analysis of Experiments. 10th ed. Hoboken, NJ: Wiley; 2020.

global\_mean\_CI\_sys <- function(means, ss, s\_method = c("sd", "SE"), ns, alpha = 0.05, sys\_method = c("var", "range")) {

sys\_method <- match.arg(sys\_method)

s\_method <- match.arg(s\_method)

k <- length(means)

total\_n <- sum(ns)

weights <- ns / total\_n

global\_mean <- sum(weights \* means)

if (s\_method == "sd") {

sds <- ss

var\_sampling <- sum((sds^2 / ns) \* (weights^2))

var\_i <- (weights^2 \* sds^2) / ns

} else if (s\_method == "SE") {

SE <- ss

var\_sampling <- sum((weights^2) \* (SE^2))

var\_i <- (weights^2 \* SE^2)

}

if (sys\_method == "var") {

if (k > 1) {

var\_sys <- sum((means - global\_mean)^2) / (k - 1) ## Bessel correction

} else {

var\_sys <- 0

}

} else if (sys\_method == "range") {

delta <- max(means) - min(means)

var\_sys <- (delta / 2)^2

}

var\_total <- var\_sampling + var\_sys

SE\_total <- sqrt(var\_total)

df <- var\_sampling^2 / sum((var\_i^2) / (ns - 1)) ## Welch's adjustment

t\_crit <- qt(1 - alpha / 2, df)

CI\_lower <- global\_mean - t\_crit \* SE\_total

CI\_upper <- global\_mean + t\_crit \* SE\_total

CI\_sys\_width <- 2\*t\_crit \* SE\_total

CI\_expected <- qt((1-alpha/2), df = sum(ns-1)) / sqrt(k)

return(c(CI\_sys\_width = CI\_sys\_width, CI\_expected = CI\_expected, var\_sampling = var\_sampling, var\_sys = var\_sys))

}

**## A2. Tonk - Systematic error estimate**

k <- 1:7

xbreaks <- c(1, 3, 5, 7)

xlabels <- paste0(xbreaks)

n <- 100

Tonk\_CIdatasysvar <- list();

for (i in 1:nrow(Tonk\_raw\_df)) {

Tonk\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Tonk\_mean[1:i], ss = Tonk\_sd[1:i], s\_method = "sd", ns = Tonk\_n[1:i], sys\_method = "var")

}

Tonk\_CIdfsysvar <- as.data.frame(list.rbind(Tonk\_CIdatasysvar))

Tonk\_CIdfsysvar

colnames(Tonk\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Tonk\_CIdfsysvar

xxx <- Tonk\_CIdfsysvar

xxx

Tonk\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Tonk\_actual\_ratiosysvar

Tonk\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Tonk\_expected\_ratiosys

Tonkdfsys <- data.frame(k, Tonk\_actual\_ratiosysvar, Tonk\_expected\_ratiosys)

colnames(Tonkdfsys) <- c("k", "Tonk\_Actsysvar", "Tonk\_Exp")

Tonkdfsys

Tonkdfsys$Tonk\_Act <- Tonkdf$Tonk\_Act

Tonkdfsys

annotation1 <- data.frame(x = c(4), y = c(200), label = paste0("n = ", n))

annotation1

plotRealTonksys <- ggplot(data = Tonkdfsys) +

geom\_point(aes(x = k, y = Tonk\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Tonk\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Tonk\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Tonk\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Tonk\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Tonk\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") + theme(legend.position = "none")

dev.new()

plotRealTonksys

ggsave("FigureH\_Tonk\_real.pdf", plot = plotRealTonksys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## B2. Kich - Systematic error estimate**

k <- 1:4

n <- 5

xbreaks <- k

xlabels <- paste0(xbreaks)

Kich\_CIdatasysvar <- list();

for (i in 1:nrow(Kich\_raw\_df)) {

Kich\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Kich\_mean[1:i], ss = Kich\_SE[1:i], s\_method = "SE", ns = Kich\_n[1:i], sys\_method = "var")

}

Kich\_CIdfsysvar <- as.data.frame(list.rbind(Kich\_CIdatasysvar))

Kich\_CIdfsysvar

colnames(Kich\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Kich\_CIdfsysvar

xxx <- Kich\_CIdfsysvar

xxx

Kich\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Kich\_actual\_ratiosysvar

Kich\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Kich\_expected\_ratiosys

Kichdfsys <- data.frame(k, Kich\_actual\_ratiosysvar, Kich\_expected\_ratiosys)

colnames(Kichdfsys) <- c("k", "Kich\_Actsysvar", "Kich\_Exp")

Kichdfsys

Kichdfsys$Kich\_Act <- Kichdf$Kich\_Act

Kichdfsys

annotation1 <- data.frame(x = c(2), y = c(200), label = paste0("n = ", n))

annotation1

plotRealKichsys <- ggplot(data = Kichdfsys) +

geom\_point(aes(x = k, y = Kich\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Kich\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Kich\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Kich\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Kich\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Kich\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

theme(legend.position = "none")

dev.new()

plotRealKichsys

ggsave("FigureH\_Kich\_real.pdf", plot = plotRealKichsys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## C2. Carl - Systematic error estimate**

k <- 1:3

n <- 10

xbreaks <- k

xlabels <- paste0(xbreaks)

Carl\_CIdatasysvar <- list();

for (i in 1:nrow(Carl\_raw\_df)) {

Carl\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Carl\_mean[1:i], ss = Carl\_sd[1:i], s\_method = "sd", ns = Carl\_n[1:i], sys\_method = "var")

}

Carl\_CIdfsysvar <- as.data.frame(list.rbind(Carl\_CIdatasysvar))

Carl\_CIdfsysvar

colnames(Carl\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Carl\_CIdfsysvar

xxx <- Carl\_CIdfsysvar

xxx

Carl\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Carl\_actual\_ratiosysvar

Carl\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Carl\_expected\_ratiosys

Carldfsys <- data.frame(k, Carl\_actual\_ratiosysvar, Carl\_expected\_ratiosys)

colnames(Carldfsys) <- c("k", "Carl\_Actsysvar", "Carl\_Exp")

Carldfsys

Carldfsys$Carl\_Act <- Carldf$Carl\_Act

Carldfsys

annotation1 <- data.frame(x = c(2), y = c(200), label = paste0("n = ", n))

annotation1

plotRealCarlsys <- ggplot(data = Carldfsys) +

geom\_point(aes(x = k, y = Carl\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Carl\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Carl\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Carl\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Carl\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Carl\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

theme(legend.position = "none")

dev.new()

plotRealCarlsys

ggsave("FigureH\_Carl\_real.pdf", plot = plotRealCarlsys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## D2. Lawr - Systematic error estimate**

k <- 1:3

n <- 5

xbreaks <- k

xlabels <- paste0(xbreaks)

Lawr\_CIdatasysvar <- list();

for (i in 1:nrow(Lawr\_raw\_df)) {

Lawr\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Lawr\_mean[1:i], ss = Lawr\_SE[1:i], s\_method = "SE", ns = Lawr\_n[1:i], sys\_method = "var")

}

Lawr\_CIdfsysvar <- as.data.frame(list.rbind(Lawr\_CIdatasysvar))

Lawr\_CIdfsysvar

colnames(Lawr\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Lawr\_CIdfsysvar

xxx <- Lawr\_CIdfsysvar

xxx

Lawr\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Lawr\_actual\_ratiosysvar

Lawr\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Lawr\_expected\_ratiosys

Lawrdfsys <- data.frame(k, Lawr\_actual\_ratiosysvar, Lawr\_expected\_ratiosys)

colnames(Lawrdfsys) <- c("k", "Lawr\_Actsysvar", "Lawr\_Exp")

Lawrdfsys

Lawrdfsys$Lawr\_Act <- Lawrdf$Lawr\_Act

Lawrdfsys

annotation1 <- data.frame(x = c(2), y = c(200), label = paste0("n = ", n))

annotation1

plotRealLawrsys <- ggplot(data = Lawrdfsys) +

geom\_point(aes(x = k, y = Lawr\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Lawr\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Lawr\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 265)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Lawr\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Lawr\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Lawr\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

theme(legend.position = "none")

dev.new()

plotRealLawrsys

ggsave("FigureH\_Lawr\_real.pdf", plot = plotRealLawrsys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## E2. Zhan - Systematic error estimate**

k <- 1:3

n <- 10

xbreaks <- k

xlabels <- paste0(xbreaks)

Zhan\_CIdatasysvar <- list();

for (i in 1:nrow(Zhan\_raw\_df)) {

Zhan\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Zhan\_mean[1:i], ss = Zhan\_SE[1:i], s\_method = "SE", ns = Zhan\_n[1:i], sys\_method = "var")

}

Zhan\_CIdfsysvar <- as.data.frame(list.rbind(Zhan\_CIdatasysvar))

Zhan\_CIdfsysvar

colnames(Zhan\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Zhan\_CIdfsysvar

xxx <- Zhan\_CIdfsysvar

xxx

Zhan\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Zhan\_actual\_ratiosysvar

Zhan\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Zhan\_expected\_ratiosys

Zhandfsys <- data.frame(k, Zhan\_actual\_ratiosysvar, Zhan\_expected\_ratiosys)

colnames(Zhandfsys) <- c("k", "Zhan\_Actsysvar", "Zhan\_Exp")

Zhandfsys

Zhandfsys$Zhan\_Act <- Zhandf$Zhan\_Act

Zhandfsys

annotation1 <- data.frame(x = c(2), y = c(200), label = paste0("n = ", n))

annotation1

plotRealZhansys <- ggplot(data = Zhandfsys) +

geom\_point(aes(x = k, y = Zhan\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Zhan\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Zhan\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 260)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Zhan\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Zhan\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Zhan\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

theme(legend.position = "none")

dev.new()

plotRealZhansys

ggsave("FigureH\_Zhan\_real.pdf", plot = plotRealZhansys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## F2. Driv - Systematic error estimate**

k <- 1:3

n <- 3

xbreaks <- k

xlabels <- paste0(xbreaks)

Driv\_CIdatasysvar <- list();

for (i in 1:nrow(Driv\_raw\_df)) {

Driv\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Driv\_mean[1:i], ss = Driv\_SE[1:i], s\_method = "SE", ns = Driv\_n[1:i], sys\_method = "var")

}

Driv\_CIdfsysvar <- as.data.frame(list.rbind(Driv\_CIdatasysvar))

Driv\_CIdfsysvar

colnames(Driv\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Driv\_CIdfsysvar

xxx <- Driv\_CIdfsysvar

xxx

Driv\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Driv\_actual\_ratiosysvar

Driv\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Driv\_expected\_ratiosys

Drivdfsys <- data.frame(k, Driv\_actual\_ratiosysvar, Driv\_expected\_ratiosys)

colnames(Drivdfsys) <- c("k", "Driv\_Actsysvar", "Driv\_Exp")

Drivdfsys

Drivdfsys$Driv\_Act <- Drivdf$Driv\_Act

Drivdfsys

annotation1 <- data.frame(x = c(2), y = c(200), label = paste0("n = ", n))

annotation1

plotRealDrivsys <- ggplot(data = Drivdfsys) +

geom\_point(aes(x = k, y = Driv\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Driv\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Driv\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 260)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Driv\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Driv\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Driv\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

theme(legend.position = "none")

dev.new()

plotRealDrivsys

ggsave("FigureH\_Driv\_real.pdf", plot = plotRealDrivsys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## G2. Zhou - Systematic error estimate**

k <- 1:4

n <- 3

xbreaks <- k

xlabels <- paste0(xbreaks)

Zhou\_CIdatasysvar <- list();

for (i in 1:nrow(Zhou\_raw\_df)) {

Zhou\_CIdatasysvar[[i]] <- global\_mean\_CI\_sys(means = Zhou\_mean[1:i], ss = Zhou\_sd[1:i], s\_method = "sd", ns = Zhou\_n[1:i], sys\_method = "var")

}

Zhou\_CIdfsysvar <- as.data.frame(list.rbind(Zhou\_CIdatasysvar))

Zhou\_CIdfsysvar

colnames(Zhou\_CIdfsysvar) <- c("CI\_sys\_widthvar", "CI\_expected", "var\_sampling", "var\_sys\_var")

Zhou\_CIdfsysvar

xxx <- Zhou\_CIdfsysvar

xxx

Zhou\_actual\_ratiosysvar <- 100 \* signif(xxx$CI\_sys\_widthvar / xxx[2, "CI\_sys\_widthvar"], 4)

Zhou\_actual\_ratiosysvar

Zhou\_expected\_ratiosys <- 100 \* signif(xxx$CI\_expected / xxx[2, "CI\_expected"], 4)

Zhou\_expected\_ratiosys

Zhoudfsys <- data.frame(k, Zhou\_actual\_ratiosysvar, Zhou\_expected\_ratiosys)

colnames(Zhoudfsys) <- c("k", "Zhou\_Actsysvar", "Zhou\_Exp")

Zhoudfsys

Zhoudfsys$Zhou\_Act <- Zhoudf$Zhou\_Act

Zhoudfsys

annotation1 <- data.frame(x = c(2), y = c(200), label = paste0("n = ", n))

annotation1

ybreaks <- c(0, 100, 200, 300, 400)

ylabels <- paste0(ybreaks)

plotRealZhousys <- ggplot(data = Zhoudfsys) +

geom\_point(aes(x = k, y = Zhou\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Zhou\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Zhou\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 400)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Zhou\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Zhou\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Zhou\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold") +

theme(legend.position = "none")

dev.new()

plotRealZhousys

ggsave("FigureH\_Zhou\_real.pdf", plot = plotRealZhousys, device = "pdf", width = 6, height = 7, units = "in", dpi = 300, path = mypath)

**## All Real data: Systematic:**

mylistsys <- list(Tonkdfsys, Kichdfsys, Carldfsys, Lawrdfsys, Zhandfsys, Drivdfsys, Zhoudfsys)

mylistsys

plotlistsys <- list(plotRealTonksys, plotRealKichsys, plotRealCarlsys, plotRealLawrsys, plotRealZhansys, plotRealDrivsys, plotRealZhousys)

**## Grid of seven graphs:**

Osys <- ggplot(data = Tonkdfsys) +

geom\_point(aes(x = k, y = Tonk\_Actsysvar), size = 3, shape = 1) +

geom\_point(aes(x = k, y = Tonk\_Act), size = 3, shape = 3) +

geom\_point(aes(x = k, y = Tonk\_Exp), size = 3, shape = 2) +

scale\_y\_continuous(breaks = ybreaks, labels = ylabels, expand = c(0, 0), limits = c(0, 230)) + ylab("C.I.(%)") +

scale\_x\_continuous(breaks = xbreaks, labels = xlabels, expand = c(0, 0), limits = c(1, max(k) + 0.2)) +

scale\_colour\_manual(name="Ratio", values = c(cbp2[8], cbp2[1], cbp2[6], cbp2[5])) +

scale\_linetype\_manual(name="Ratio", values=c("Actualsys" = "dotted", "Actual" = "dashed", "Expected" = "solid")) + theme(axis.title.x = element\_text(vjust = -2)) +

geom\_line(aes(x = k, y = Tonk\_Actsysvar, color = "Actualsys", linetype = "Actualsys")) +

geom\_line(aes(x = k, y = Tonk\_Act, color = "Actual", linetype = "Actual")) +

geom\_line(aes(x = k, y = Tonk\_Exp, color = "Expected", linetype = "Expected")) +

geom\_text(data = annotation1, aes(x=x, y=y, label=label), color = cbp2[8], size=5, fontface="bold")

legendsys <- get\_legend(Osys)

graph7sys <- plot\_grid(plotlistsys[[1]], plotlistsys[[2]], plotlistsys[[3]], plotlistsys[[4]], plotlistsys[[5]], plotlistsys[[6]], plotlistsys[[7]], NULL, ncol=3, nrow=3, align="hv", rel\_widths = c(1, 1, 1, 1, 1, 1, 1, 1, 1), labels = c('A', 'B', 'C', 'D', 'E', 'F', 'G', "", ""))

dev.new(width = 9, height = 9)

graph7legendsys <- graph7sys + draw\_grob(legendsys, 2/3, 0, 1/3, 0.5)

graph7legendsys

ggsave("FigureJ\_realsys.pdf", plot = graph7legendsys, device = "pdf", width = 9, height = 9, units = "in", dpi = 300, path = mypath)

**## References:** purrr (1), broom (2), DescTools (3), Hmisc (4), XNomial (5), rlist (6), plyr (7), cowplot (8), ggtext (9), ggarrow (10), ggforce (11), forcats (12), dplyr (13), data.table (14), ggplot2 (15), ggpattern (16), tidyverse (17) DO NOT SCAN DOCUMENT ! (transfer to temp. word file)

## 1. Hadley W, Lionel H. purrr: Functional Programming Tools. 2025

## 2. David R, Alex H, Simon C. broom: Convert Statistical Objects into Tidy Tibbles. 2025

## 3. Andri S. DescTools: Tools for Descriptive Statistics. 2025

## 4. Frank EHJ. Hmisc: Harrell Miscellaneous. 2025

## 5. Bill E. XNomial: Exact Goodness-of-Fit Test for Multinomial Data with Fixed Probabilities. 2024

## 6. Kun R. rlist: A Toolbox for Non-Tabular Data Manipulation. 2021

## 7. Hadley W. The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software. 2011;40:1-29.

## 8. Claus OW. cowplot: Streamlined Plot Theme and Plot Annotations for ‘ggplot2’. 2024

## 9. Claus OW, Brenton MW. ggtext: Improved Text Rendering Support for ‘ggplot2’. 2022

## 10. Teun VDB. ggarrow: Arrows for ‘ggplot2’. 2024

## 11. Thomas LP. ggforce: Accelerating ‘ggplot2’. 2025

## 12. Hadley W. forcats: Tools for Working with Categorical Variables (Factors). 2023

## 13. Hadley W, Romain F, Lionel H, Kirill M, Davis V. dplyr: A Grammar of Data Manipulation. 2023

## 14. Tyson B, Matt D, Arun S, Jan G, Michael C, Toby H, Benjamin S, Ivan K. data.table: Extension of `data.frame`. 2025

## 15. Hadley W. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2016

## 16. Mike FC, Trevor LD, ggplot2 A. ggpattern: ‘ggplot2’ Pattern Geoms. 2025

## 17. Hadley W, Mara A, Jennifer B, Winston C, Lucy DM, Romain F, Garrett G, Alex H, Lionel H, Jim H, Max K, Thomas LP, Evan M, Stephan MB, Kirill M, Jeroen O, David R, Dana PS, Vitalie S, Kohske T, Davis V, Claus W, Kara W, Hiroaki Y. Welcome to the tidyverse. Journal of Open Source Software. 2019;4:1686.

##