

Library Size 1e6

Loading Data

```
library(tximeta)
dir <- "/fs/cbcb-lab/rob/students/noor/Uncertainty/boot_gibbs/quant_output/Drosophilla/sim_1e6"
source("/fs/cbcb-lab/rob/students/noor/Uncertainty/boot_gibbs/R/helper_func.R")
load("/fs/cbcb-lab/rob/students/noor/Uncertainty/ase-sim/sim_1e6/sim_counts_matrix.rda") ##True Counts

fileNames <- paste("sample", sort(as.vector(outer(c("01", "02"), c("B", "GS"), paste, sep="_"))), sep="")
files <- file.path(dir, fileNames, "quant.sf")
coldata <- data.frame(files, names = fileNames, infType = rep(c("Boot", "GS"), times = 2), stringsAsFactors = F)
seSamp1 <- tximeta(coldata[c(1:2),])
seSamp2 <- tximeta(coldata[c(3:4),])

seSamp1 <- appTrueCounts(seSamp1, counts_matrix[,1])
seSamp2 <- appTrueCounts(seSamp2, counts_matrix[,2])
```

Computing size factors and confidence intervals

```
seSamp1 <- computeSizeFactors(seSamp1)

## [1] 1
## [1] 2

seSamp2 <- computeSizeFactors(seSamp2)

## [1] 1
## [1] 2

seSamp1 <- computeConfInt(seSamp1, sf = T)
seSamp2 <- computeConfInt(seSamp2, sf = T)
```

Coverages for the replicates

```
print(computeCoverage(counts_matrix[,1], seSamp1, list(seq(nrow(seSamp1))))) ###Replicate 1

##   sample_01_B sample_01_GS
## 1   0.8302623   0.9954509

print(computeCoverage(counts_matrix[,2], seSamp2, list(seq(nrow(seSamp2))))) ###Replicate 2

##   sample_02_B sample_02_GS
## 1   0.8298031   0.9948482
```

Plotting the coverage difference across bins of varying expression

```
nB = c(20,50)
pNb <- vector(mode = "list", length = 2)
```

```

for(i in seq_along(pNb))
  pNb[[i]] <- vector(mode = "list", length = length(nB))

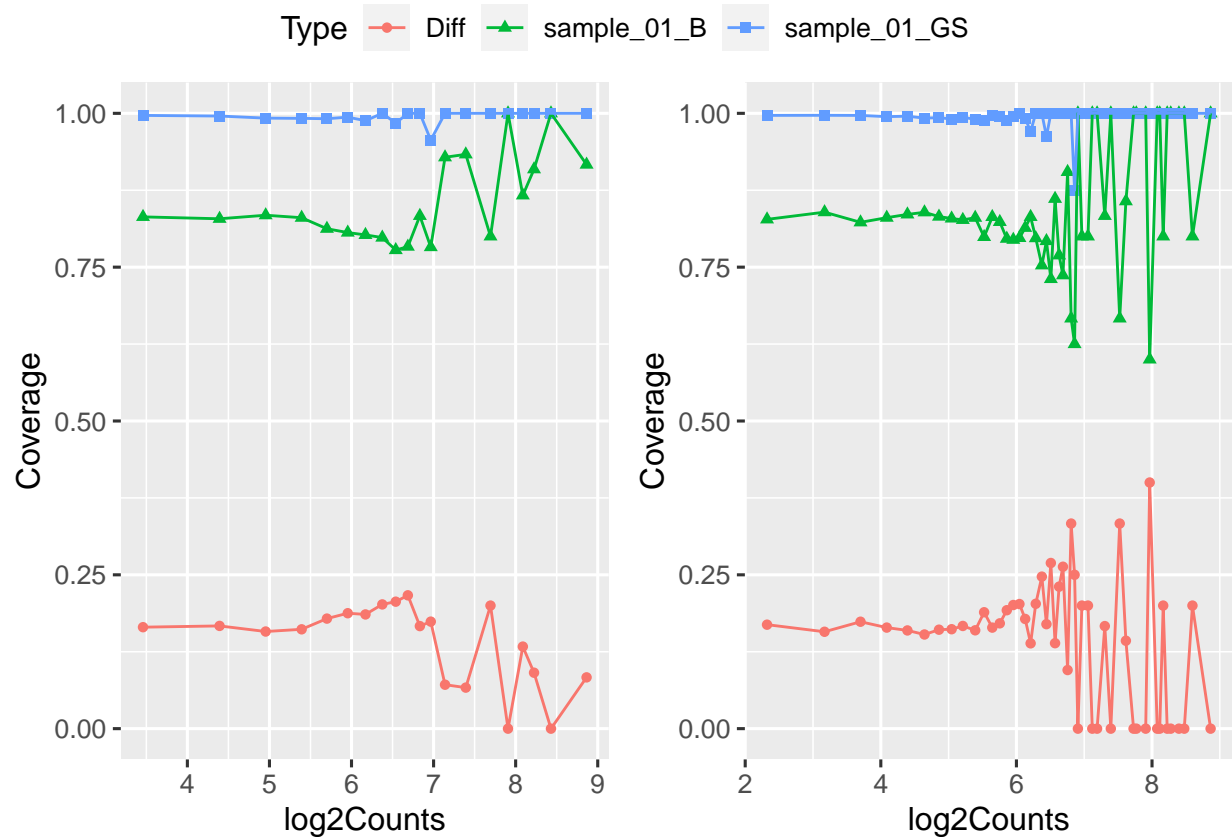
for(j in seq_along(pNb))
{
  i=1
  for(b in nB)
  {
    cInds <- extractBinInds(counts_matrix[,j], breaks = b)
    if(j==1)
      covDf <- createCovDf(seSamp1, counts_matrix[,j], cInds)
    else
      covDf <- createCovDf(seSamp2, counts_matrix[,j], cInds)
    covDf[(b*2+1):(b*3),] = covDf[1:b,]
    covDf[(b*2+1):(b*3),3] = abs(covDf[1:b,3] - covDf[(b*1+1):(b*2),3])
    covDf[,2] = as.character(covDf[,2])
    covDf[(b*2+1):(b*3),2] = "Diff"
    pNb[[j]][[i]] <- plotCovDf(covDf, line=T)
    #print(sort(covDf[(b*2+1):(b*3),3], decreasing=T)[1:20]*100)
    i=i+1
  }
}

p1 <- ggarrange(plotlist = pNb[[1]], common.legend = T)
p2 <- ggarrange(plotlist = pNb[[2]], common.legend = T)

```

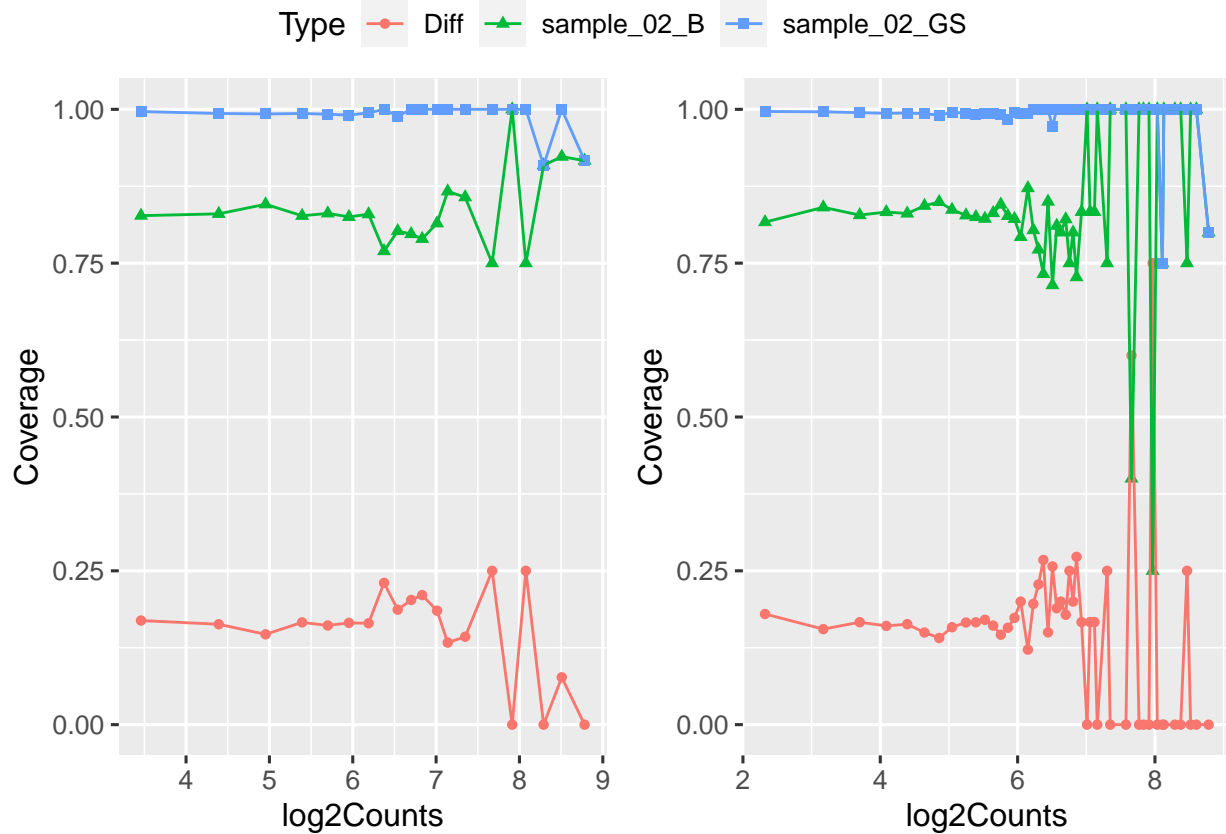
Coverage for Replicate 1 across 20 and 50 bins

```
print(p1)
```



Coverage for Replicate 2 across 20 and 50 bins

```
print(p2)
```



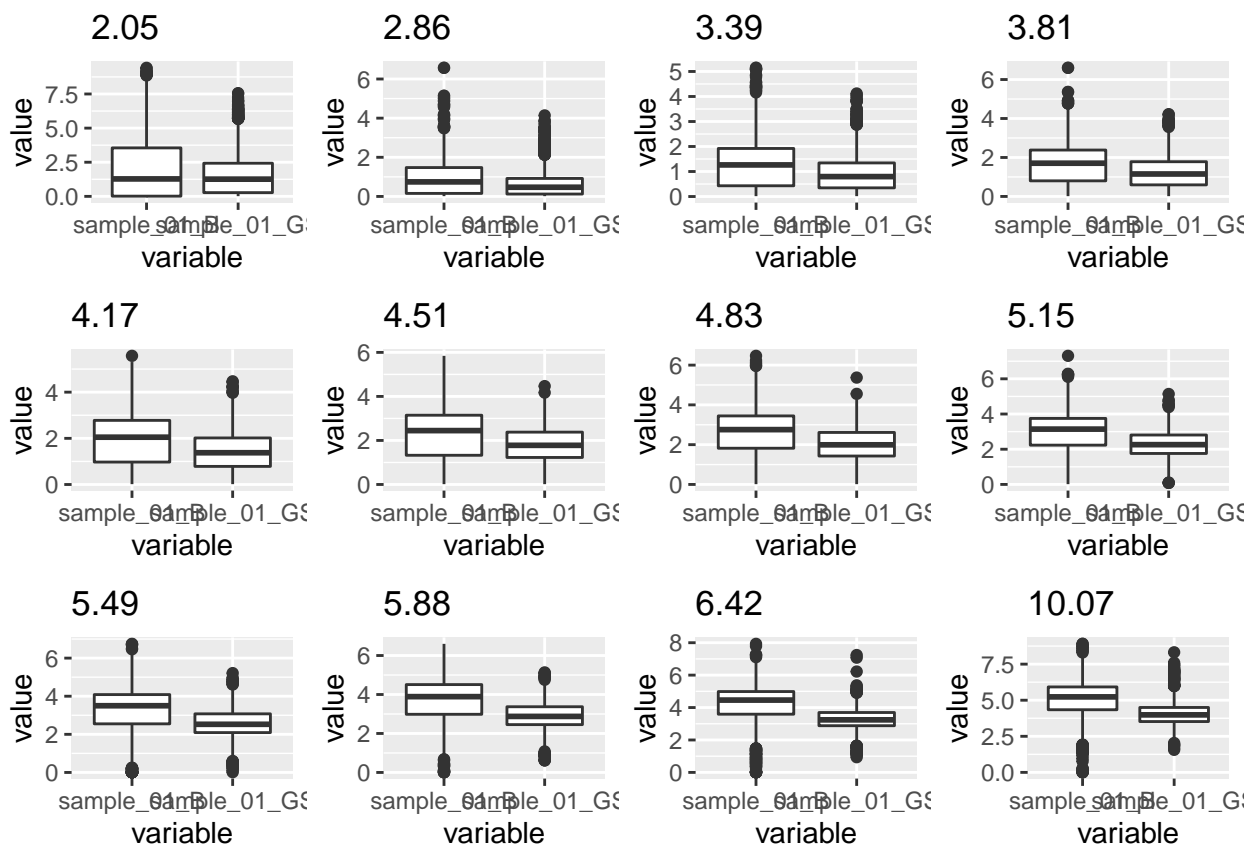
Boxplots

Inf RV

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="infRV", nbreaks = 12, type = "BP")
```

```
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
```

```
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
```

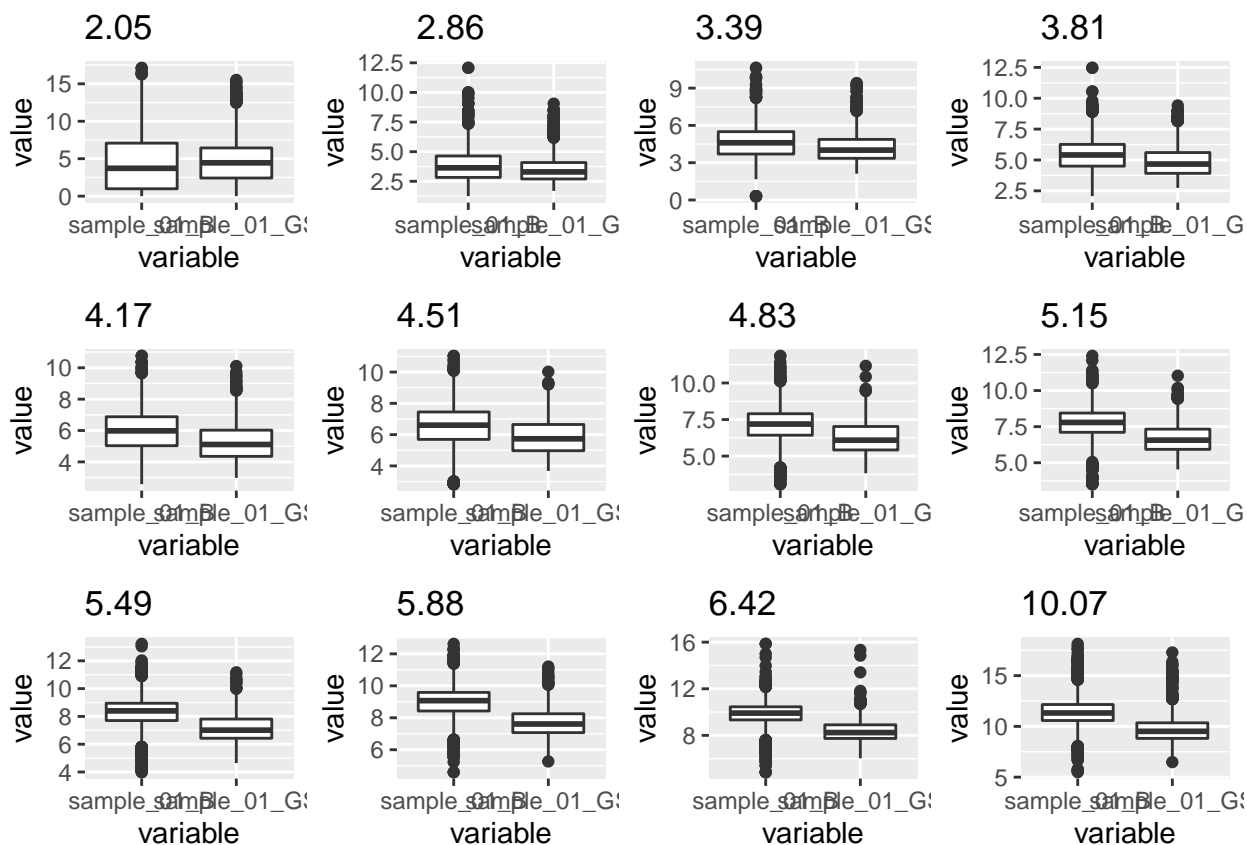


Variance

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="variance", nbreaks = 12, type = "BP")
```

```
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
```

```
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
```

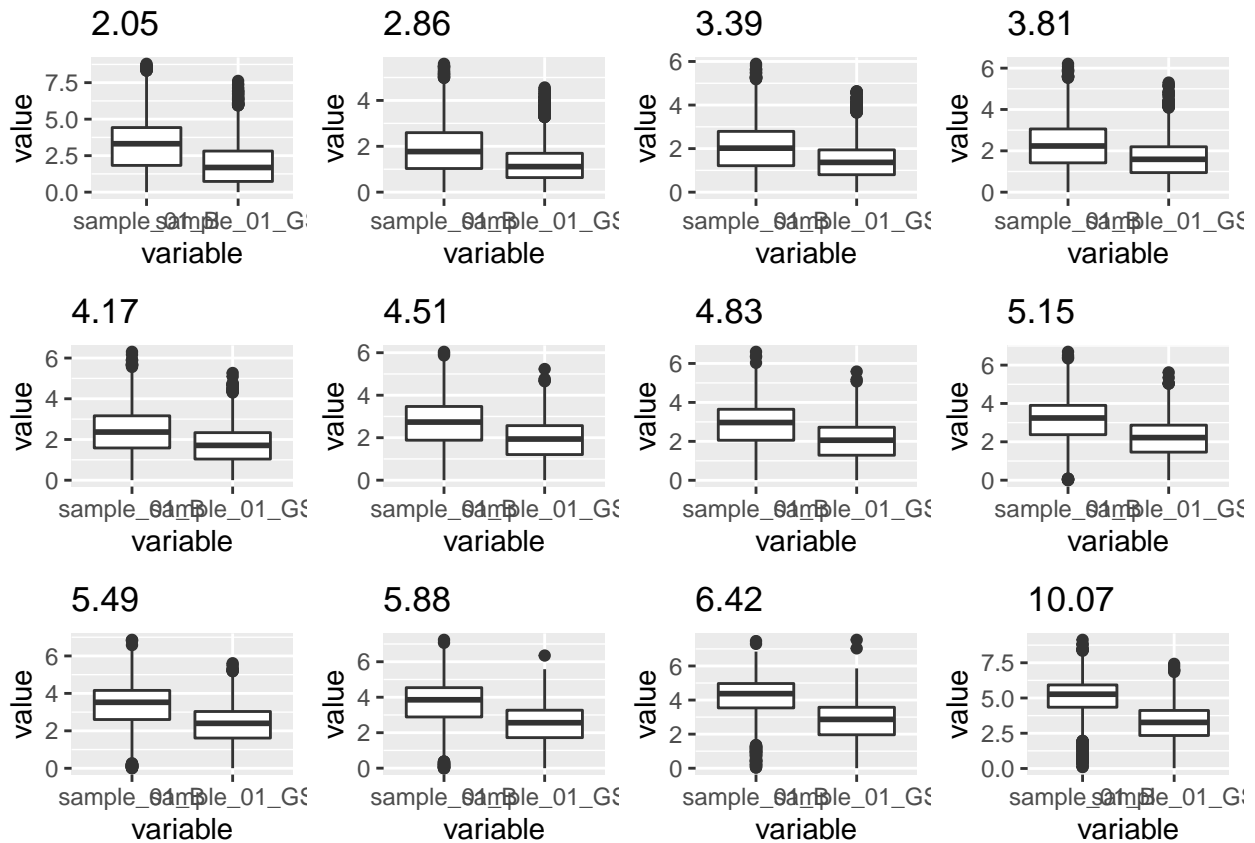


Bias

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="bias", nbreaks = 12, type = "BP")
```

```
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
```

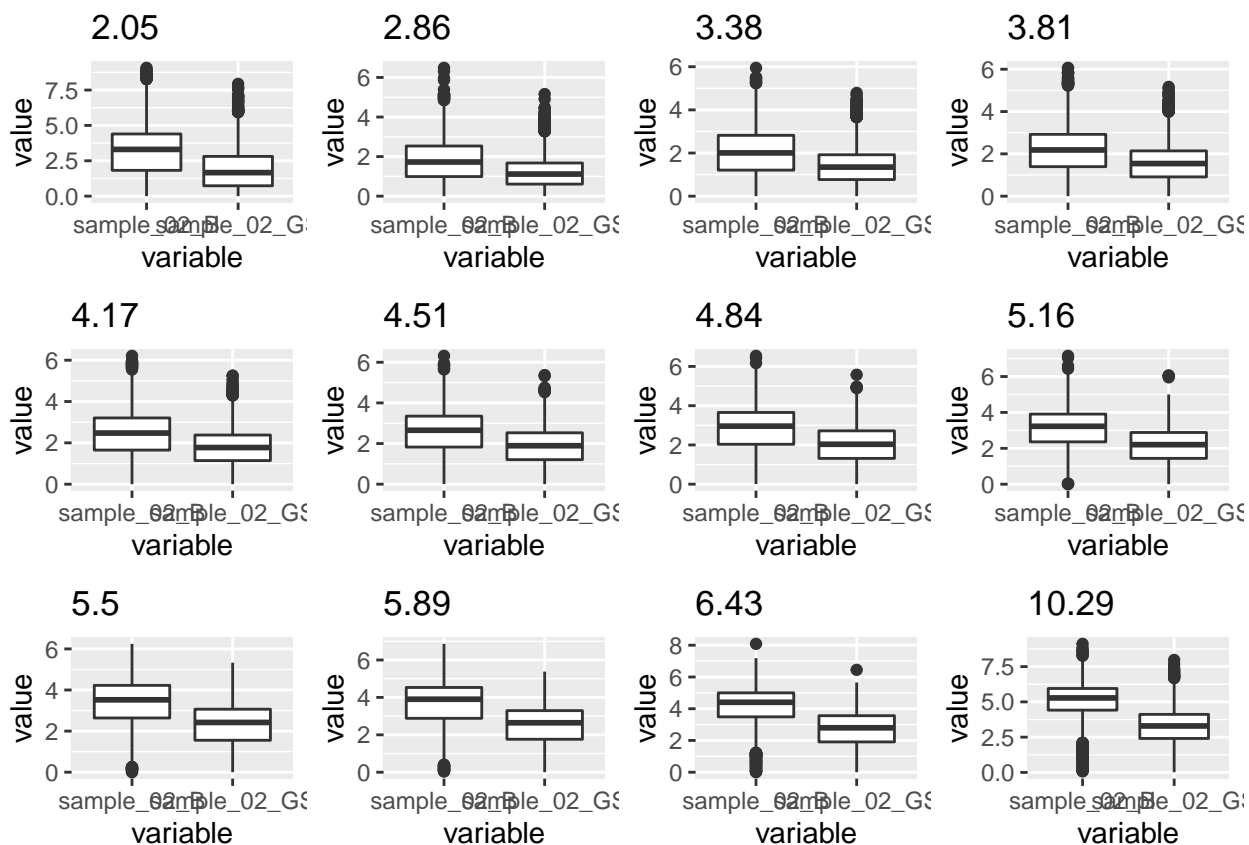
```
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
```



```
### Replicate 2
pL <- plotSummary(seSamp2, summQuant="bias", nbreaks = 12, type = "BP")
```

```
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
```

```
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
```

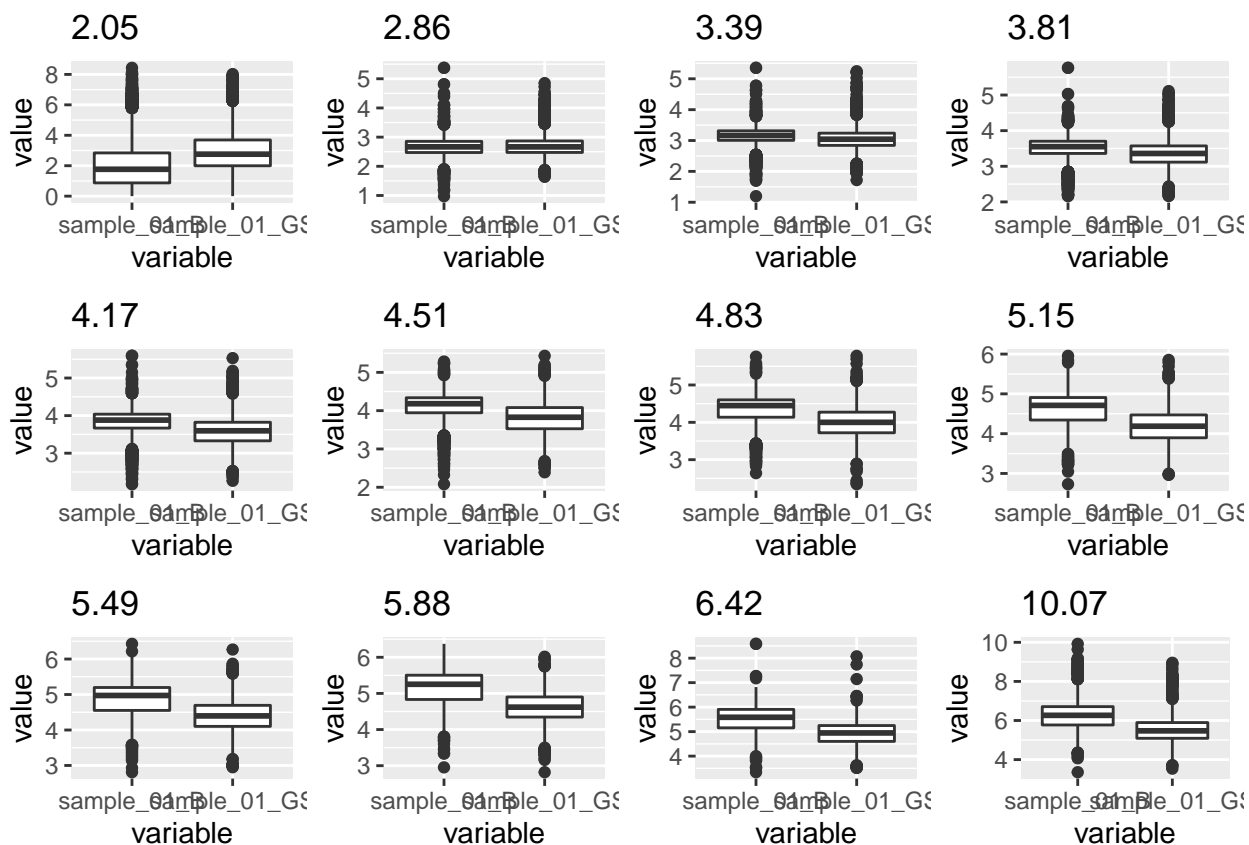


Mean

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="mean", nbreaks = 12, type = "BP")
```

```
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
```

```
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
```

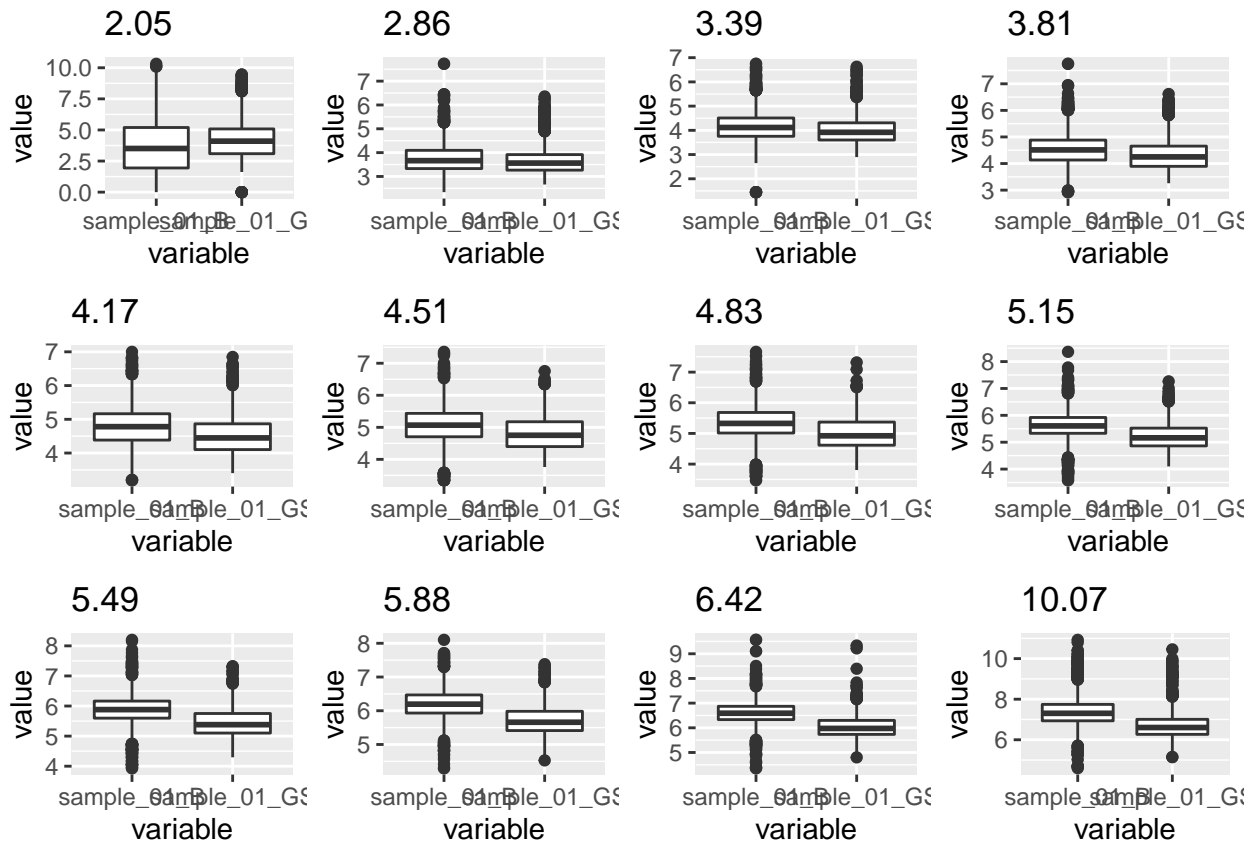



Width

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="Width", nbreaks = 12, type = "BP")
```

```
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
## Using as id variables
```

```
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
```



Width with Counts

```
df1 <- createWidthDf(seSamp1)
df2 <- createWidthDf(seSamp2)

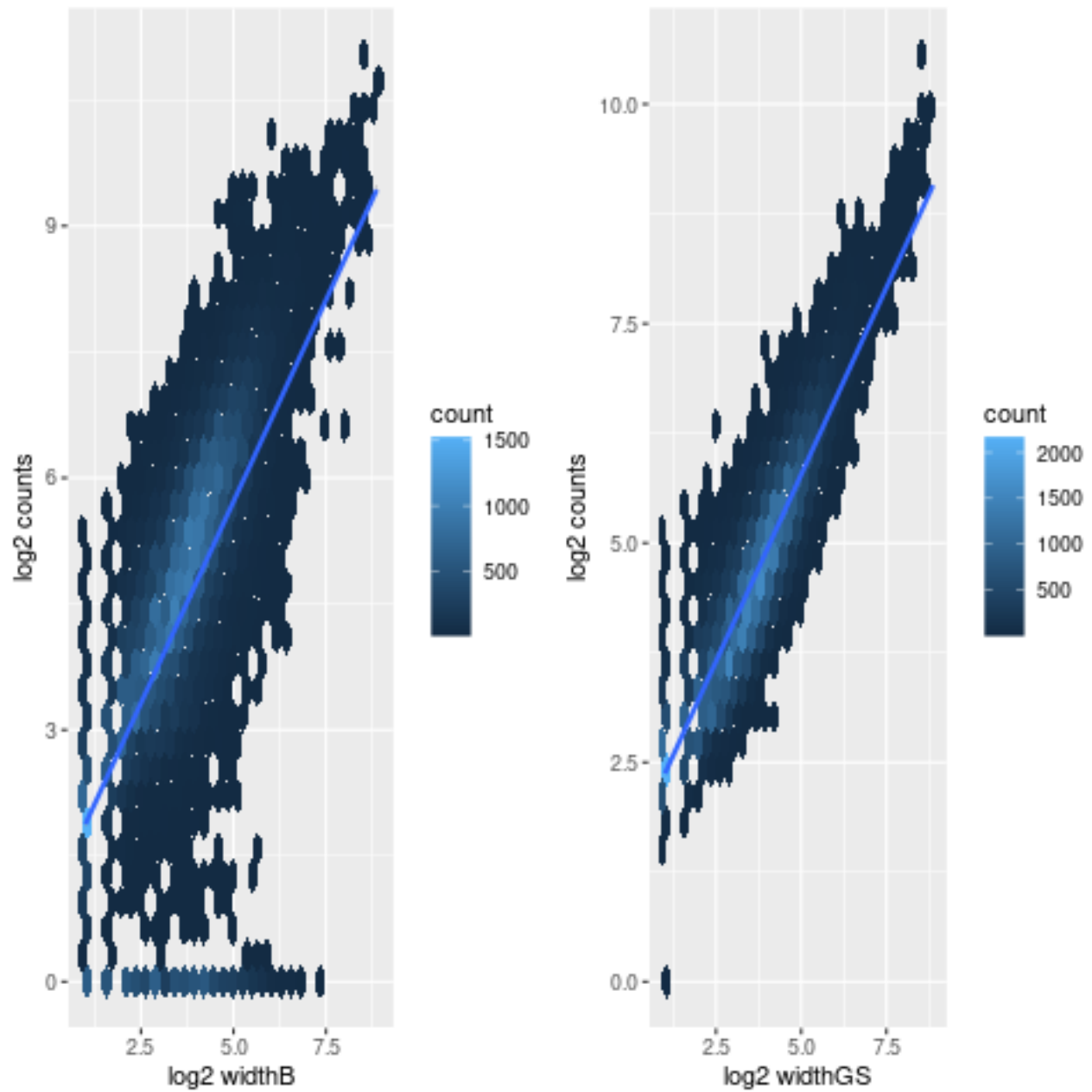
### Log count with log width
p1 <- plotWidthDf(df1)

## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'

png("width_1.png")
p1[[1]]
dev.off()

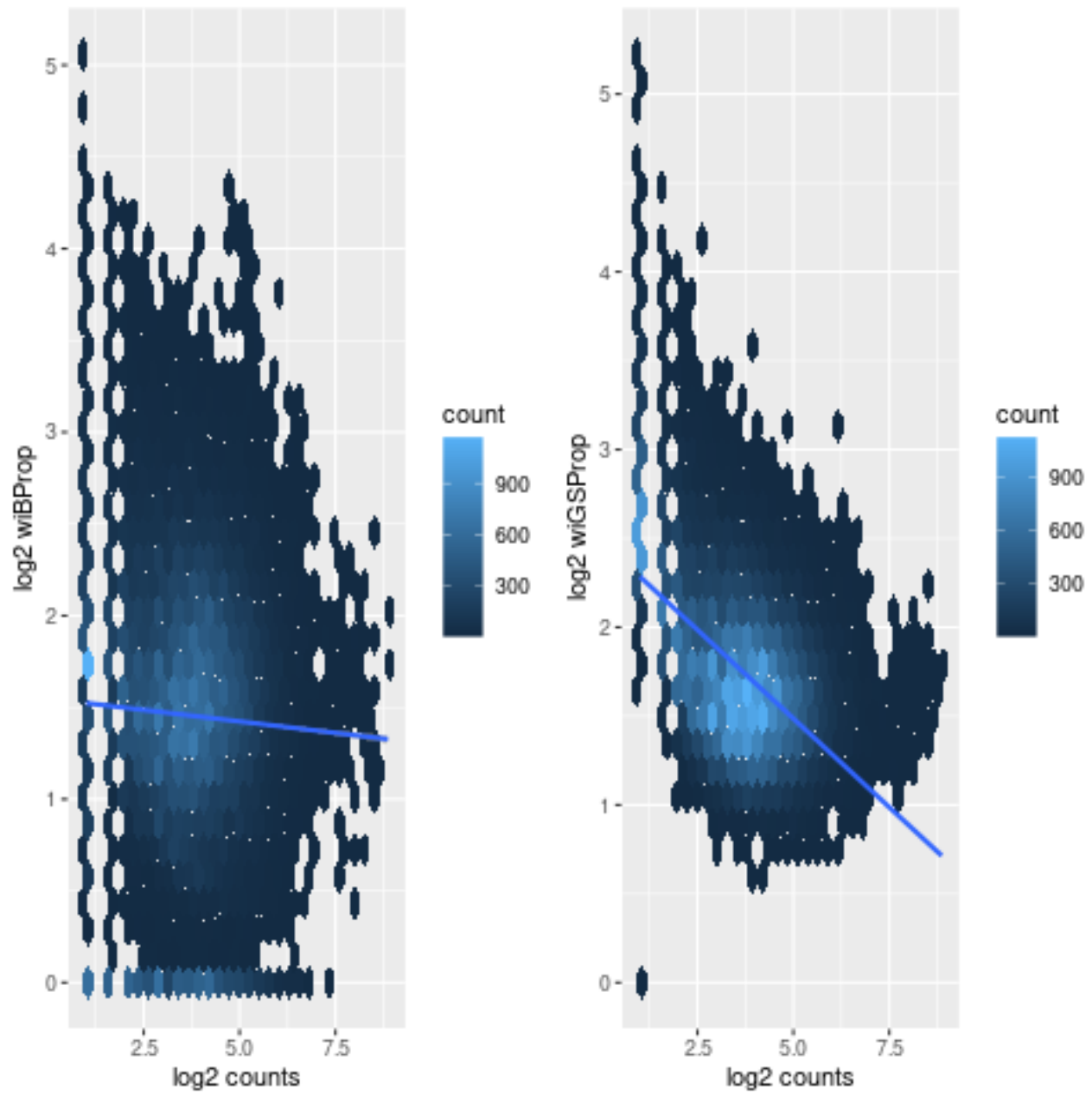
## pdf
## 2

knitr::include_graphics("width_1.png")
```



```
### Log count with log width/counts
png("width_2.png")
p1[[2]]
dev.off()

## pdf
## 2
knitr::include_graphics("width_2.png")
```



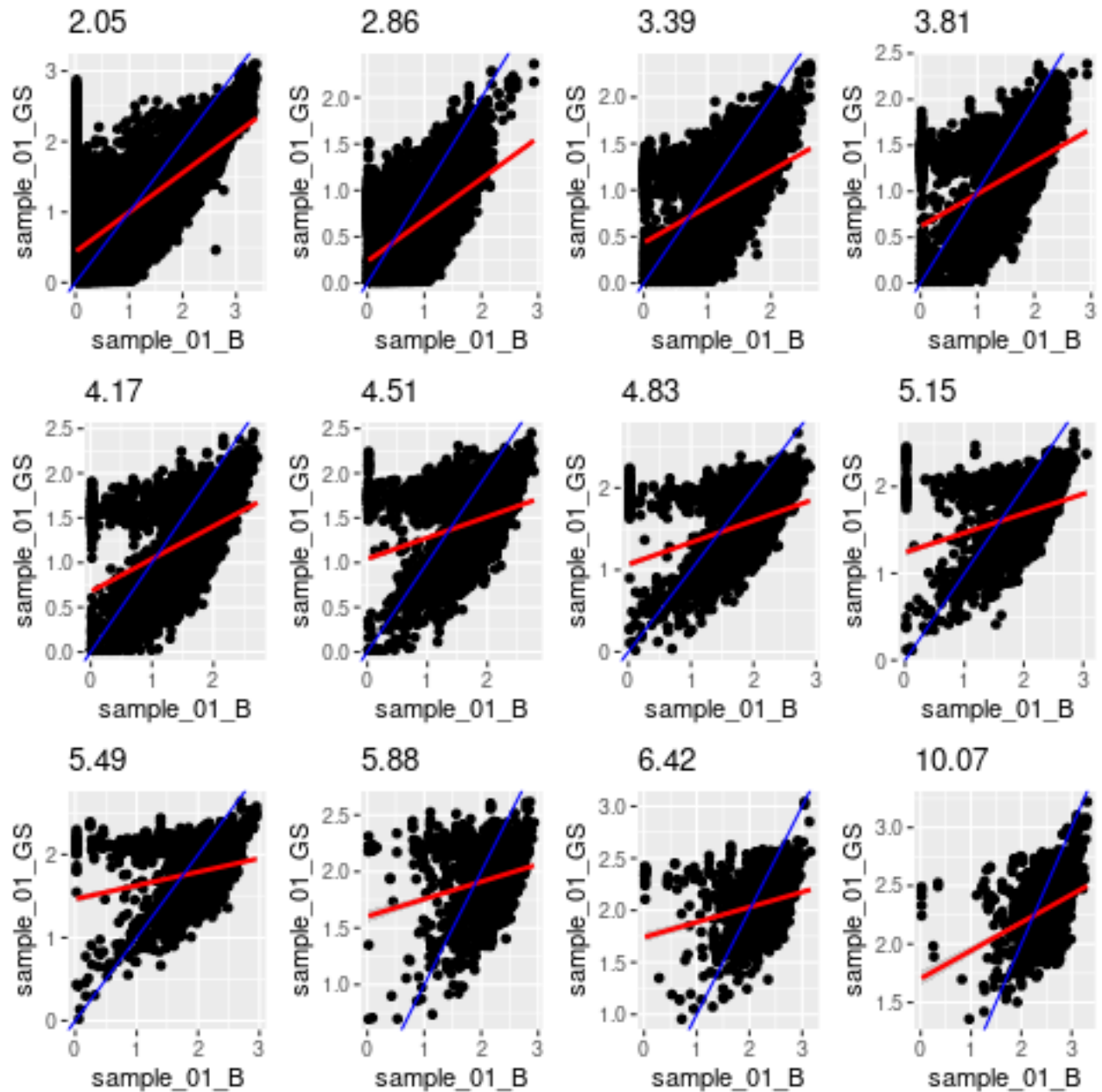
Scatter Plots

Inferential Variance

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="infRV", nbreaks = 12)
png("Inf.png")
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
dev.off()
```

```
## pdf
## 2
```

```
knitr::include_graphics("Inf.png")
```

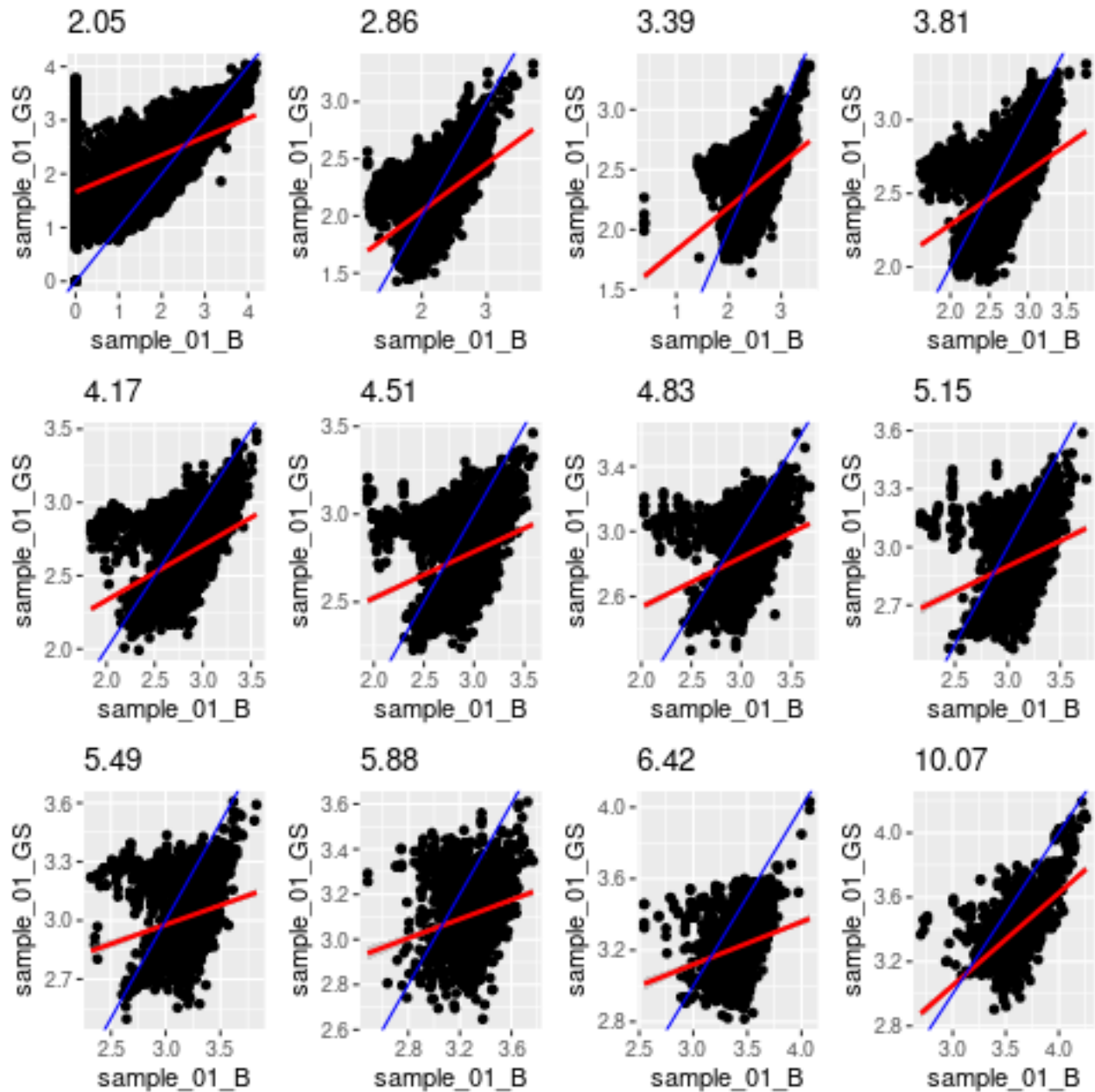


Variance

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="variance", nbreaks = 12)
png("var1.png")
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
dev.off()
```

```
## pdf
## 2
```

```
knitr::include_graphics("var1.png")
```

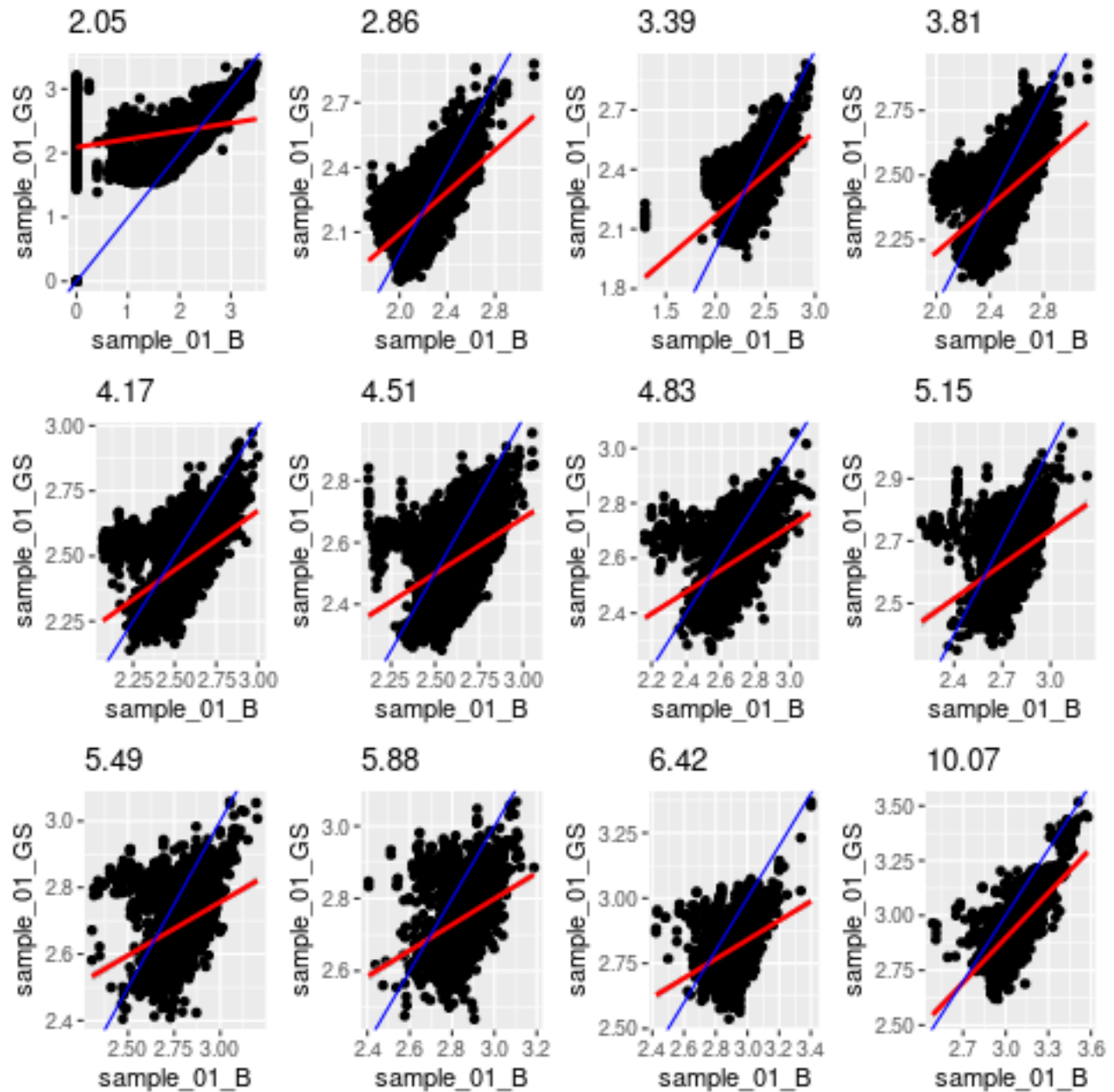


Width

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="Width", nbreaks = 12)
png("width1.png")
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
dev.off()
```

```
## pdf
## 2
```

```
knitr::include_graphics("width1.png")
```

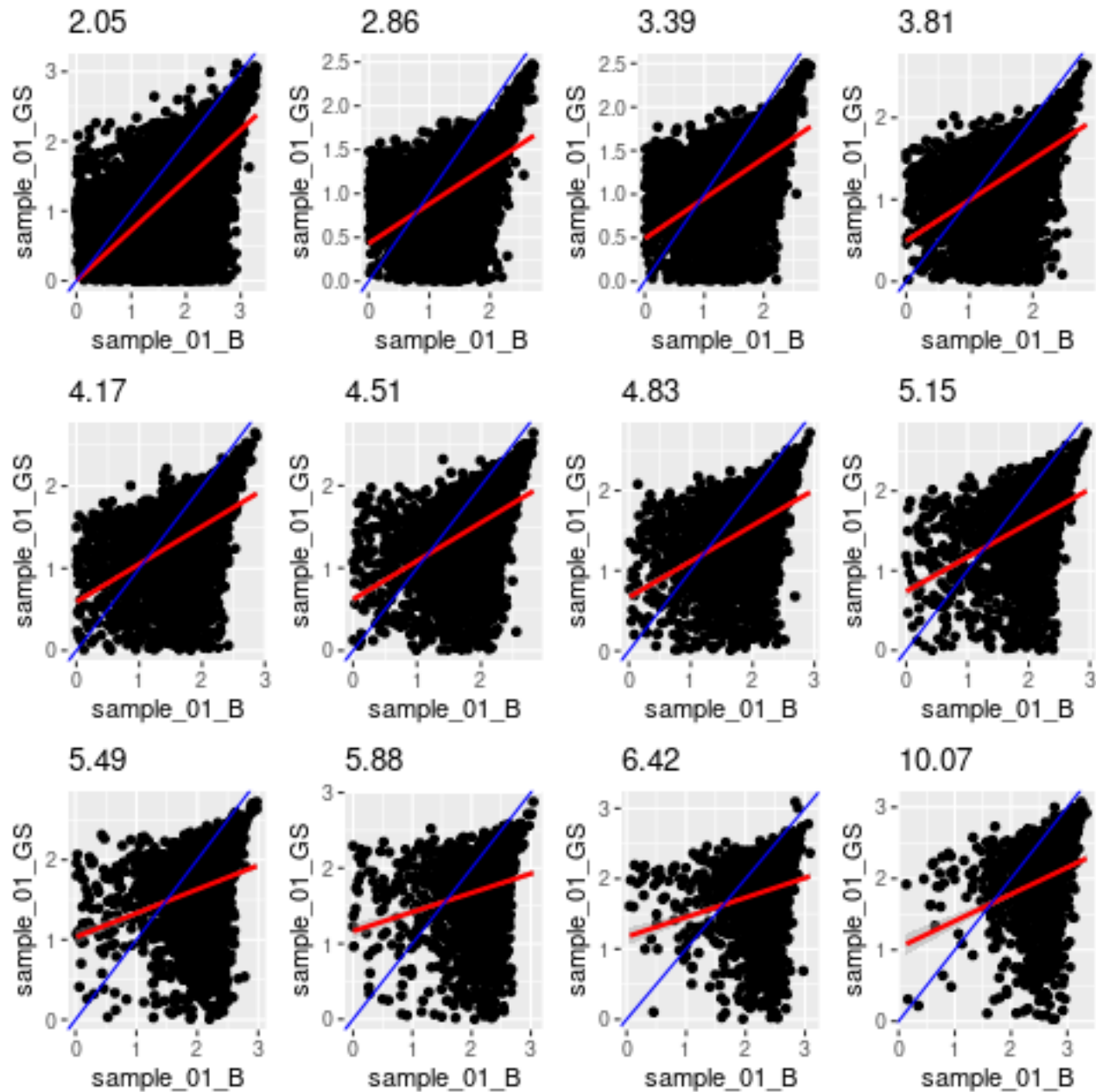


bias

```
### Replicate 1
pL <- plotSummary(seSamp1, summQuant="bias", nbreaks = 12)
png("bias1.png")
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
dev.off()
```

```
## pdf
## 2
```

```
knitr::include_graphics("bias1.png")
```



```
### Replicate 2
pL <- plotSummary(seSamp2, summQuant="bias", nbreaks = 12)
png("bias2.png")
ggarrange(plotlist = pL, nrow = 3, ncol = 4)
dev.off()
```

```
## pdf
## 2
```



```
knitr::include_graphics("bias2.png")
```

