

Replication1

Bernard A. Coles IV

10/29/2019

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggpubr)
```

```
## Loading required package: ggplot2
## Loading required package: magrittr
```

```
library(ggplot2)
```

```
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:dplyr':
##
##   src, summarize
## The following objects are masked from 'package:base':
##
##   format.pval, units
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##   select
```

```
library(ppcor)
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:survival':
##
##      aml

## The following object is masked from 'package:lattice':
##
##      melanoma
```

```
CorFunc <- function(d,indices){
  R <- rcorr(d$X[indices],d$Y[indices])
  return(R$r[1,2])
}
```

```
# example of difference in significance but no difference in effect.
for (i in c(1:10)){ #
  print(i)
  # set.seed(1235) # for reproducibility
  set.seed(1234+i)
  A <- rnorm(20,0.5,1)# simulate random data
  tA <- t.test(A, mu=0)
  print(mean(A))
  print(tA$p.value)
}
```

```
## [1] 1
## [1] 0.5886905
## [1] 0.005039768
## [1] 2
## [1] 0.570992
## [1] 0.01676168
## [1] 3
## [1] 0.5471627
## [1] 0.05202813
## [1] 4
## [1] 0.4991654
## [1] 0.02134782
## [1] 5
## [1] 0.03927364
## [1] 0.8561865
## [1] 6
## [1] 0.5937153
## [1] 0.01069414
## [1] 7
## [1] 0.546981
## [1] 0.03120765
## [1] 8
## [1] 0.6472981
## [1] 0.007175651
## [1] 9
## [1] 0.3387416
## [1] 0.08353486
## [1] 10
## [1] 0.05170604
## [1] 0.8279797
```

```

set.seed(1237) # for reproducibility
A <- rnorm(20,0.5,0.9)# simulate random data
t.test(A, mu=0)

##
## One Sample t-test
##
## data: A
## t = 2.2834, df = 19, p-value = 0.03409
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.04521522 1.03967761
## sample estimates:
## mean of x
## 0.5424464

set.seed(1238) # for reproducibility
B <- rnorm(20,0.5,1.5)# simulate random data with larger variance
t.test(B, mu=0)

##
## One Sample t-test
##
## data: B
## t = 1.671, df = 19, p-value = 0.1111
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.1259521 1.1234482
## sample estimates:
## mean of x
## 0.498748

AB=as.matrix(c(A,B))
group = factor(rep(c('A','B'),each=20));
dt = data.frame(AB,group)
t.test(A,B,var.equal = TRUE)

##
## Two Sample t-test
##
## data: A and B
## t = 0.11455, df = 38, p-value = 0.9094
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.7285505 0.8159472
## sample estimates:
## mean of x mean of y
## 0.5424464 0.4987480

sumup = data.frame("Gr" = c('A','B'))

sumup<-cbind(sumup,Mean=with(dt, tapply(AB, group, mean)))
sumup<-cbind(sumup,SD=with(dt, tapply(AB, group, sd)))
sumup<-cbind(sumup,N=with(dt, tapply(AB, group, length)))
sumup<-cbind(sumup,SE =sumup$SD/sqrt(sumup$N)) # Calculate standard error of the mean
ciMult <- qt(.95/2 + .5, sumup$N-1)

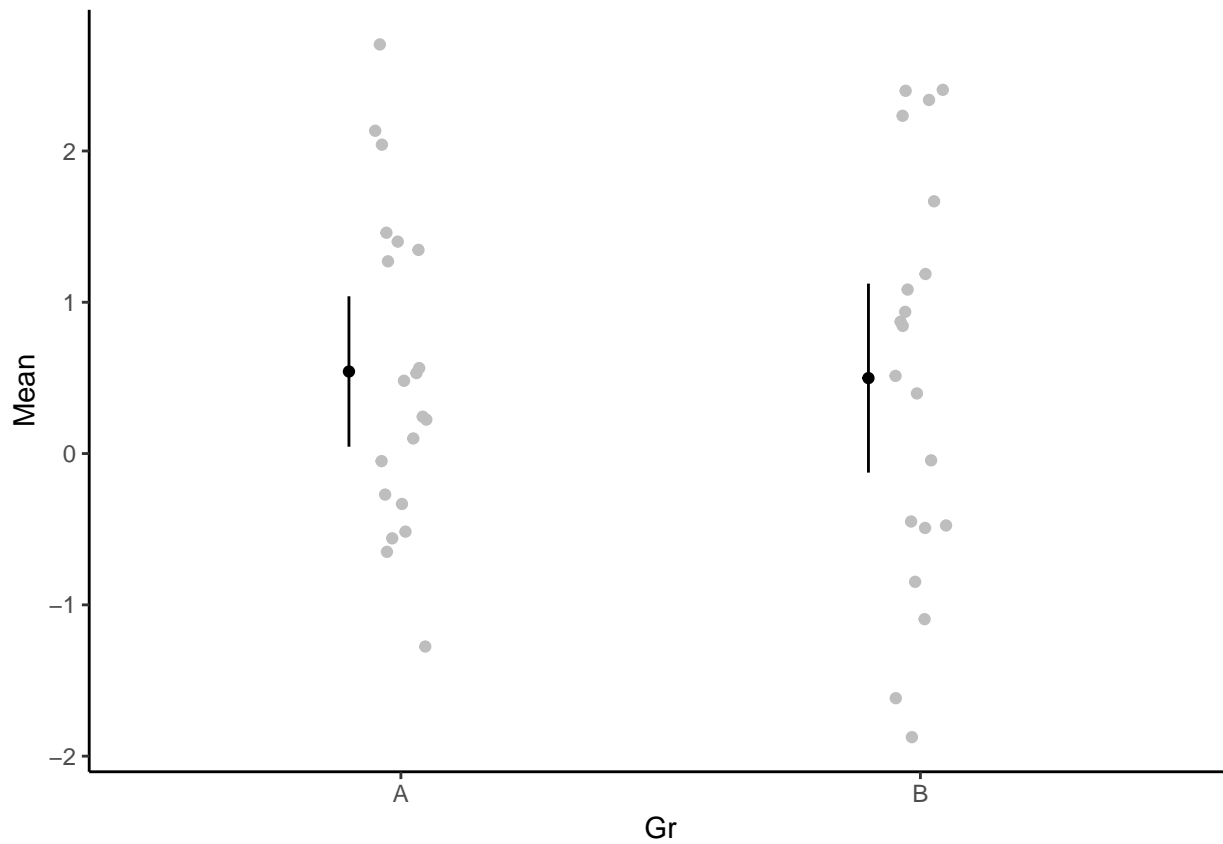
```

```

sumup<-cbind(sumup,CI = sumup$SE * ciMult)

EffNieuw <- ggplot(sumup, aes(x=Gr, y=Mean)) +
  geom_errorbar(aes(ymin=Mean-CI, ymax=Mean+CI), width=NA,position = position_nudge(x = -0.1)) +
  geom_point(position = position_nudge(x = -0.1)) +
  geom_jitter(data=dt,aes(x=group,y=AB),color = "gray",width = 0.05)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = element_
print(EffNieuw)

```



```

# figure on Nieuwenhuis with correlations
Rval <- 0.5
Sigma <- matrix(c(1,Rval,Rval,1),2,2)
for (i in c(10,26)){ #,11,8,18,22,24 --> other possible examples
  print(i)
  set.seed(1234+i) # for reproducibility
  D <- mvrnorm(n = 20, rep(0, 2), Sigma, empirical = FALSE)
  X <- D[,1]
  Y <- D[,2]
  Add <- data.frame(X,Y)
  R<-rcorr(Add$X,Add$Y,type=c("pearson"))
  RValue = R$r[1,2]
  print(R$r[1,2])
  print(R$p[1,2])

  # bootstrap CI
  cor.boot <- NULL
  N <- dim(Add)[1]

```

```

for (i in 1:1000) {
  idx <- sample.int(N, N, replace = TRUE)
  cor.boot[i] <- cor(Add[idx, ])[1,2]
}
Rbci0<- quantile(cor.boot, c(.025, .975))

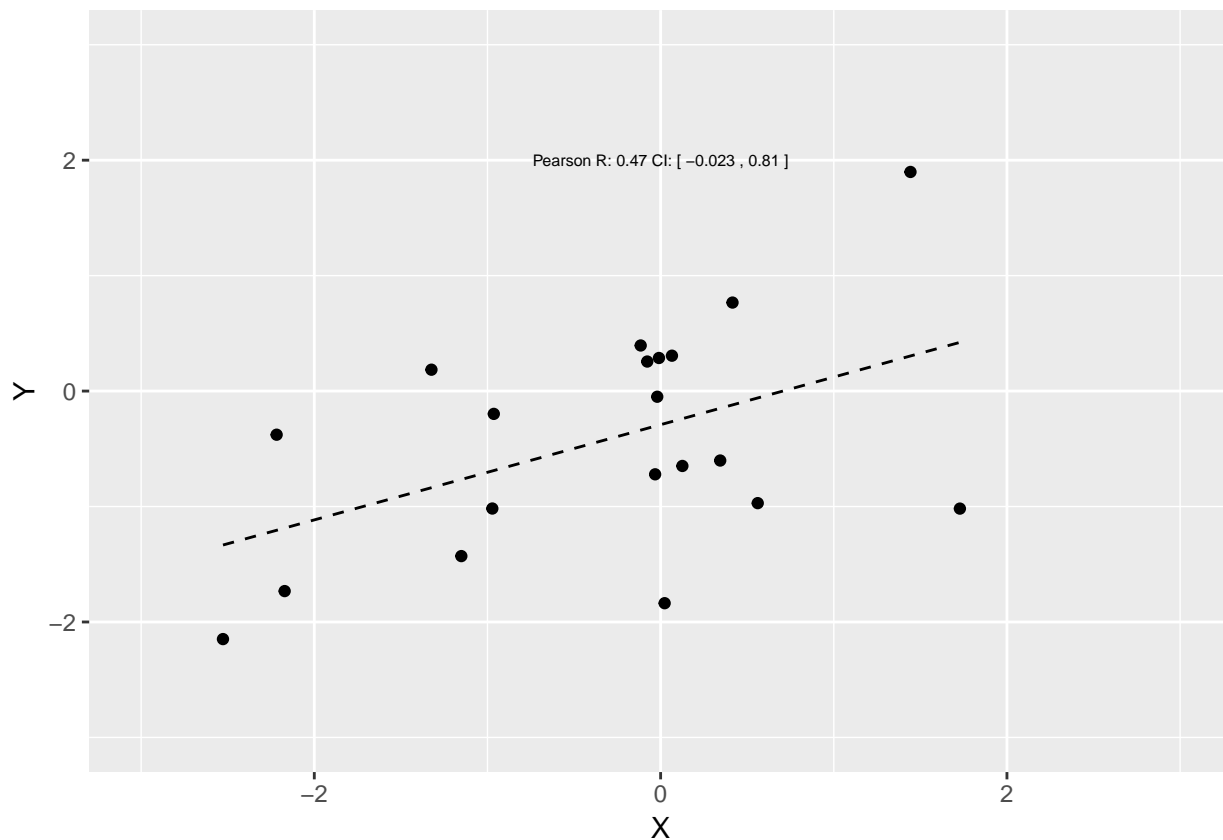
txt = toString(paste("Pearson R:", format(RValue,digits=2), "CI: [",format( Rbci0[1],digits=2),",",fo
NiewenCorr<-ggplot(Add, aes(X, Y)) +
  geom_point() +
  labs(x = "X", y = "Y") +
  geom_smooth(method="lm",color = "black",linetype = 2, size=0.5, se=FALSE) +
  coord_cartesian(xlim = c(-3,3),ylim = c(-3,3)) +
  annotate("text", x = 0, y = 2, label = txt,size=2)+
  theme(legend.position="none")
print(NiewenCorr)
}

```

```

## [1] 10
## [1] 0.474046
## [1] 0.03472206

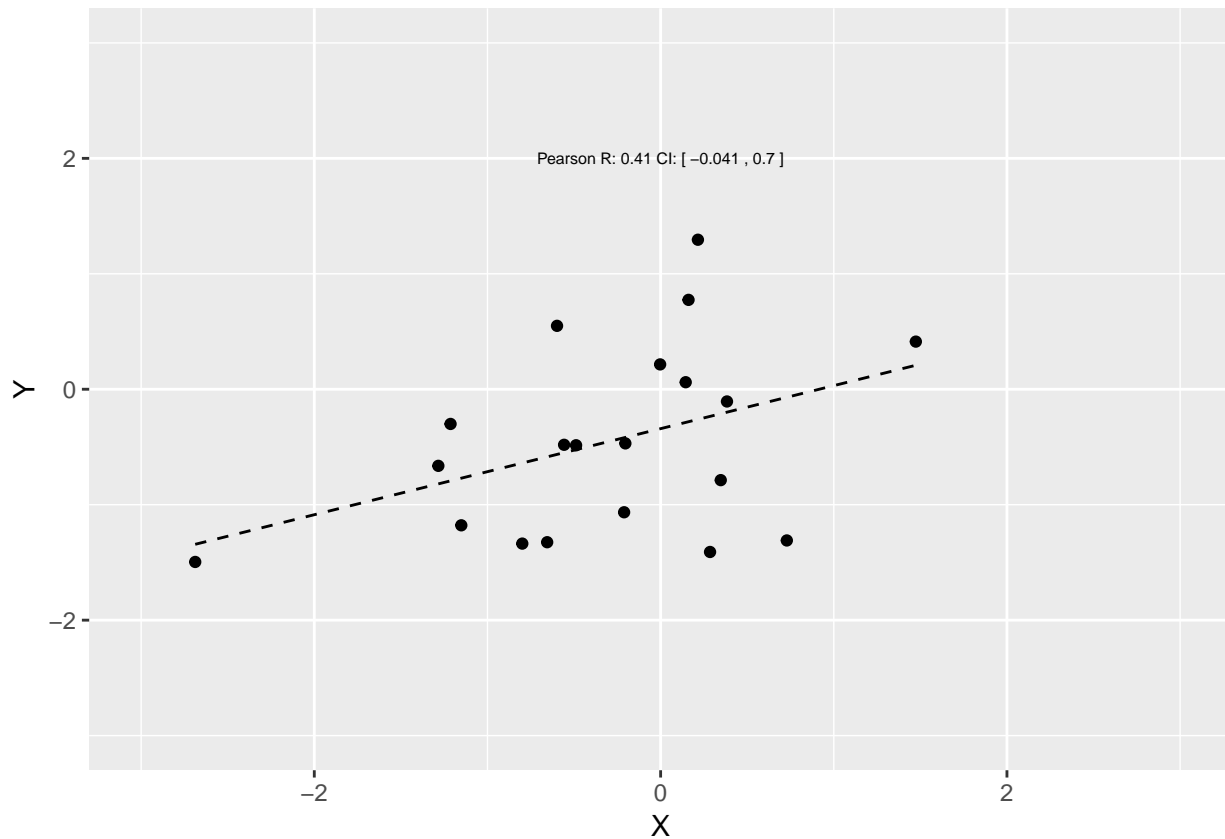
```



```

## [1] 26
## [1] 0.4082099
## [1] 0.07397

```



```
#####
# figure on underpower
#####
P=NA;R=NA;Cond=NA;
LowP = data.frame(P,R,Cond)
HighP= data.frame(P,R,Cond)

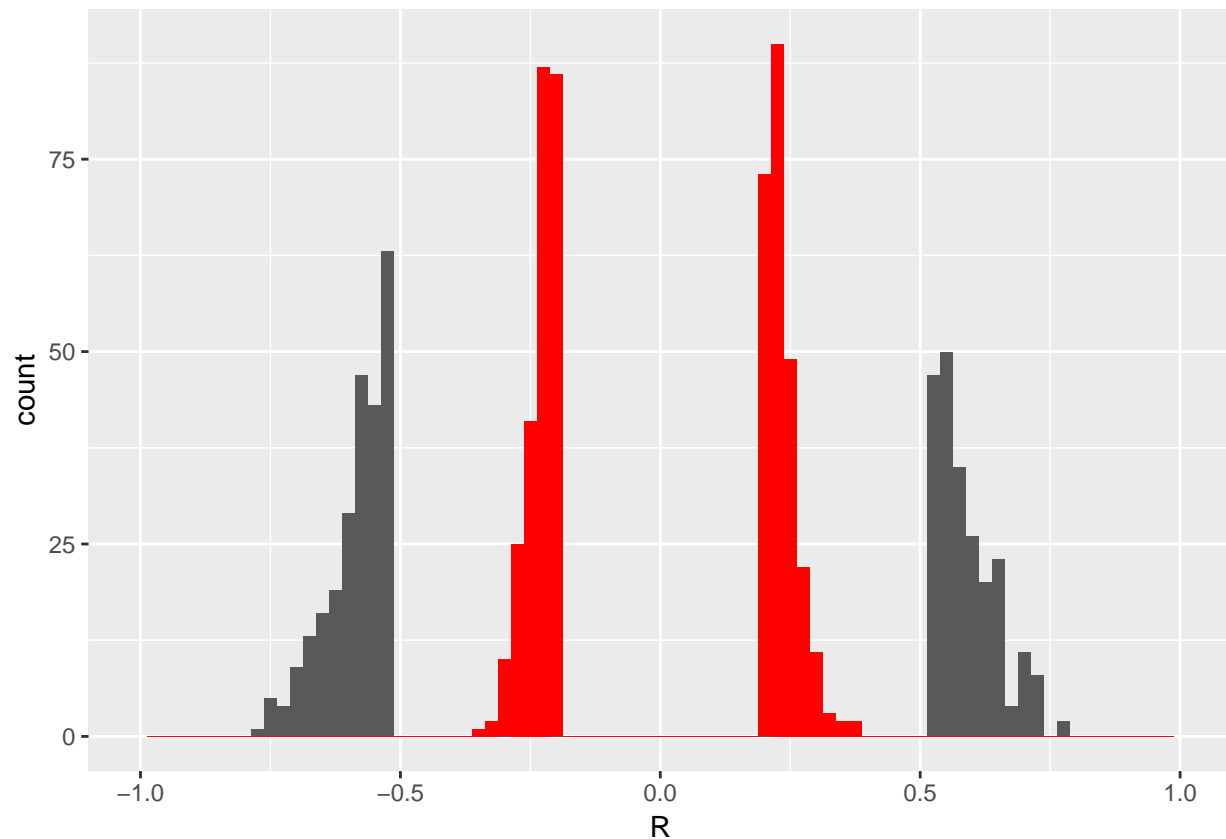
for (i in 1:10000){
  A <- rnorm(15,0.5,1)
  B <- rnorm(15,0.5,1)
  R<-rcorr(A,B)
  LowP <- rbind(LowP,c(R$P[1,2], R$r[1,2],R$P[1,2]<0.05))

  A <- rnorm(100,0.5,1)
  B <- rnorm(100,0.5,1)
  R<-rcorr(A,B)
  HighP <- rbind(HighP,c(R$P[1,2], R$r[1,2],R$P[1,2]<0.05))
}
LowP$Cond = factor(LowP$Cond)
HighP$Cond = factor(HighP$Cond)

L2 <-filter(LowP, Cond == "1")
R2 <-filter(HighP, Cond == "1")
ggplot(L2, aes(x=R)) +
  geom_histogram(aes(x=R),binwidth=.025)+
  geom_histogram(data=R2,aes(x=R),binwidth=.025,fill="red")+
  xlim(-1, 1)
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```
#####  
# figure on correlation  
#####  
  
set.seed(1234) # for reproducibility  
PP = list()  
  
# effect of outlier  
X <- rnorm(20,0,1)  
Y <- rnorm(20,0,1)  
Z <- rep(1,20)  
Dcor = data.frame(X,Y,Z)  
for (i in c(1,3,5)){  
  print(i)  
  Add <- rbind(Dcor,c(i,i,2))  
  R<-rcorr(Add$X,Add$Y,type=c("pearson"))  
  RValue = R$r[1,2]  
  print(R$p[1,2])  
  
  # bootstrap CI  
  cor.boot <- NULL  
  N <- dim(Add)[1]  
  for (i in 1:1000) {  
    idx <- sample.int(N, N, replace = TRUE)
```

```

    cor.boot[i] <- cor(Add[idx, ])[1,2]
  }
  Rbci<- quantile(cor.boot, c(.025, .975))

  txt = toString(paste("Pearson R:", format(RValue,digits=2), "CI: [",format( Rbci[1],digits=2),",",format(
PP[[length(PP) + 1]]<-ggplot(Add, aes(X, Y),fill=factor(Z)) +
  scale_color_manual(values=c("black", "red")) +
  scale_fill_manual(values=c(NA, "red")) +
  geom_point(aes(colour = factor(Z),fill=factor(Z)),shape=21) +
  labs(x = "X", y = "Y") +
  geom_smooth(method="lm",color = "black",linetype = 2, size=0.5) +
  coord_cartesian(xlim = c(-2.5,6.5),ylim = c(-2.5,6.5)) +
  annotate("text", x = 0, y = 6, label = txt,size=2)+
  theme(legend.position="none")
}

```

```

## [1] 1
## [1] 0.5595559

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

## Warning in cor(Add[idx, ]): the standard deviation is zero

```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

## Warning in cor(Add[idx, ]): the standard deviation is zero
## Warning in cor(Add[idx, ]): the standard deviation is zero
## Warning in cor(Add[idx, ]): the standard deviation is zero
## Warning in cor(Add[idx, ]): the standard deviation is zero
# effect of subgroups
set.seed(1234) # for reproducibility
QQ = list()
Xbase <- rnorm(20,0,1)
Ybase <- rnorm(20,0,1)
Z <- c(rep(0,10),rep(1,10))

for (i in c(0,2,4)){
  print(i)
  X = Xbase+i*Z
  Y = Ybase+i*Z
  Add = data.frame(X,Y,Z)
  R<-rcorr(Add$X,Add$Y)
  RCor = R$r[1,2]
  print(R$p[1,2])
  # bootstrap CI
  cor.boot <- NULL
  N <- dim(Add)[1]
  for (i in 1:1000) {
    idx <- sample.int(N, N, replace = TRUE)
    cor.boot[i] <- cor(Add[idx, ])[1,2]
  }
  Rbci2<- quantile(cor.boot, c(.025, .975))

  txt = toString(paste("Pearson R:", format(RCor,digits=2), "CI: [",format( Rbci2[1],digits=2),",",format(
QQ[[length(QQ) + 1]]<-ggplot(Add, aes(X, Y),fill=factor(Z)) +
  scale_color_manual(values=c("black", "red")) +
  scale_fill_manual(values=c(NA, "red")) +
  geom_point(aes(colour = factor(Z),fill=factor(Z)),shape=21) +
  labs(x = "X", y = "Y") +
  geom_smooth(method="lm",color = "black",linetype = 2, size=0.5) +
  coord_cartesian(xlim = c(-2.5,6.5),ylim = c(-2.5,6.5)) +
  annotate("text", x = 0, y = 6, label = txt,size=2)+
  theme(legend.position="none")
}

## [1] 0
## [1] 0.2378304
## [1] 2
## [1] 0.05013606
## [1] 4
## [1] 2.337339e-05

ggarrange(PP[[1]],PP[[2]],PP[[3]],QQ[[1]],QQ[[2]],QQ[[3]],ncol = 3, nrow = 2,labels = c("A","B","C","D")

```