May 5, 2024

The results below are generated from an R script.

```
library(nlme)
library(lattice)
library(car)

## Warning:  package ’car’ was built under R version 4.3.3
## Loading required package:  carData

data<-read.csv("ptcp_gnometrans.csv")


data_no_g11<-data[data$generation <11, ]
data_no_g11<-data_no_g11[data_no_g11$condition =="SSL", ]
drops <- c("network_id","replication","cloned","algorithm","algorithm_description")
data_no_g11<-data_no_g11[ , !(names(data_no_g11) %in% drops)]
data_no_g11<-na.omit(data_no_g11)

n <- nrow(data_no_g11)

summary(mod.lm0<-lm(p_trans ~ 1,data=data_no_g11))

##
## Call:
## lm(formula = p_trans ~ 1, data = data_no_g11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54657 -0.21324  0.05343  0.25343  0.45343
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54657    0.01511   36.16   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3416 on 510 degrees of freedom

summary(mod.lm1<-lm(p_trans ~ generation,data=data_no_g11))

##
## Call:
## lm(formula = p_trans ~ generation, data = data_no_g11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55686 -0.20856  0.04913  0.25612  0.46111
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.558857   0.037921  14.738   <2e-16 ***
## generation  -0.001996   0.005650  -0.353    0.724
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3419 on 509 degrees of freedom
## Multiple R-squared:  0.0002452, Adjusted R-squared:  -0.001719
## F-statistic: 0.1249 on 1 and 509 DF,  p-value: 0.724
```

```
summary(mod.lm2<-lm(p_trans ~ s_demo+generation,data=data_no_g11))
```

```
##
## Call:
## lm(formula = p_trans ~ s_demo + generation, data = data_no_g11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65909 -0.19107  0.03265  0.23150  0.59797
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.430683   0.040626  10.601  < 2e-16 ***
## s_demo       0.235571   0.033696   6.991 8.62e-12 ***
## generation  -0.003582   0.005406  -0.663    0.508
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3269 on 508 degrees of freedom
## Multiple R-squared:  0.08799, Adjusted R-squared:  0.0844
## F-statistic: 24.51 on 2 and 508 DF,  p-value: 6.917e-11
```

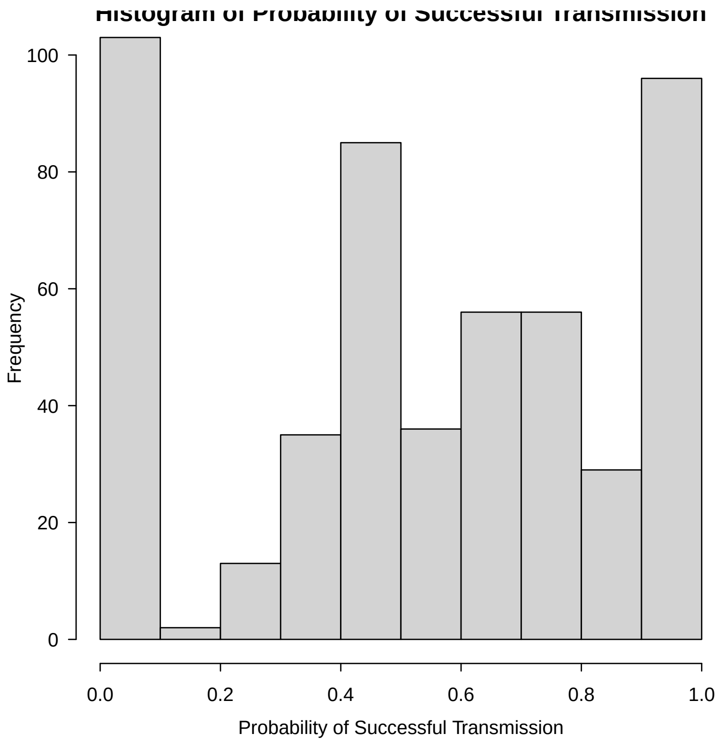
```
anova(mod.lm0,mod.lm1)
```

```
## Analysis of Variance Table
##
## Model 1: p_trans ~ 1
## Model 2: p_trans ~ generation
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    510 59.528
## 2    509 59.514  1  0.014598 0.1249  0.724
```

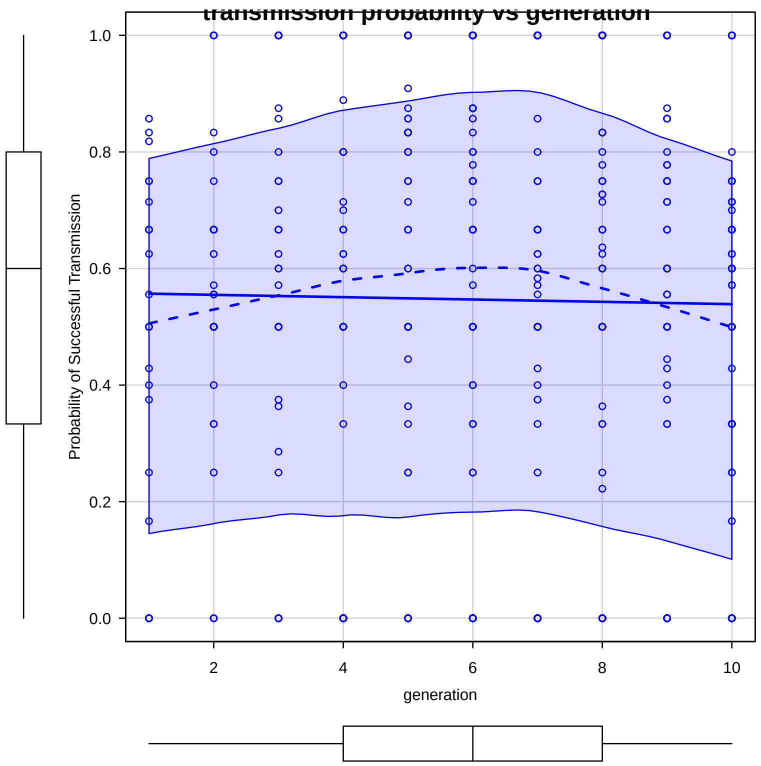
```
anova(mod.lm1,mod.lm2)
```

```
## Analysis of Variance Table
##
## Model 1: p_trans ~ generation
## Model 2: p_trans ~ s_demo + generation
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    509 59.514
## 2    508 54.290  1    5.2233 48.875 8.623e-12 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

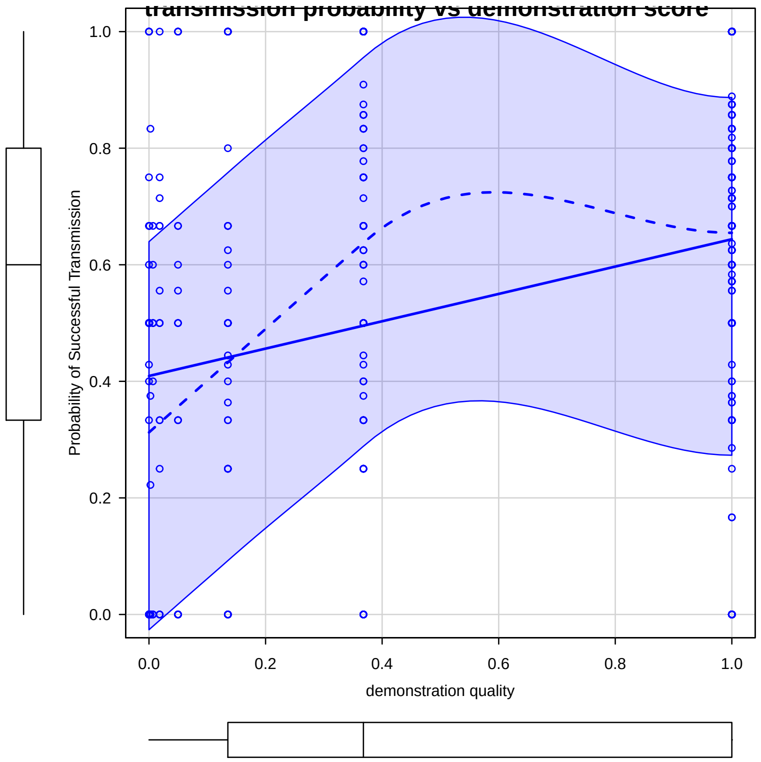
```
anova(mod.lm0,mod.lm2)

## Analysis of Variance Table
##
## Model 1: p_trans ~ 1
## Model 2: p_trans ~ s_demo + generation
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    510 59.528
## 2    508 54.290  2    5.2379 24.506 6.917e-11 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

anova.1_2<-anova(mod.lm1,mod.lm2)


hist(data_no_g11$p_trans,main = "Histogram of Probability of Successful Transmission",xlab = ’Probabilit
```



```
scatterplot(p_trans ~ generation, data=data_no_g11,xlab=’generation’,ylab = ’Probability of Successful T
title(main = "transmission probability vs generation ")
```



```
scatterplot(p_trans ~ s_demo, data=data_no_g11,xlab=’demonstration quality’,ylab = ’Probability of Succe
title(main = "transmission probability vs demonstration score ")
```



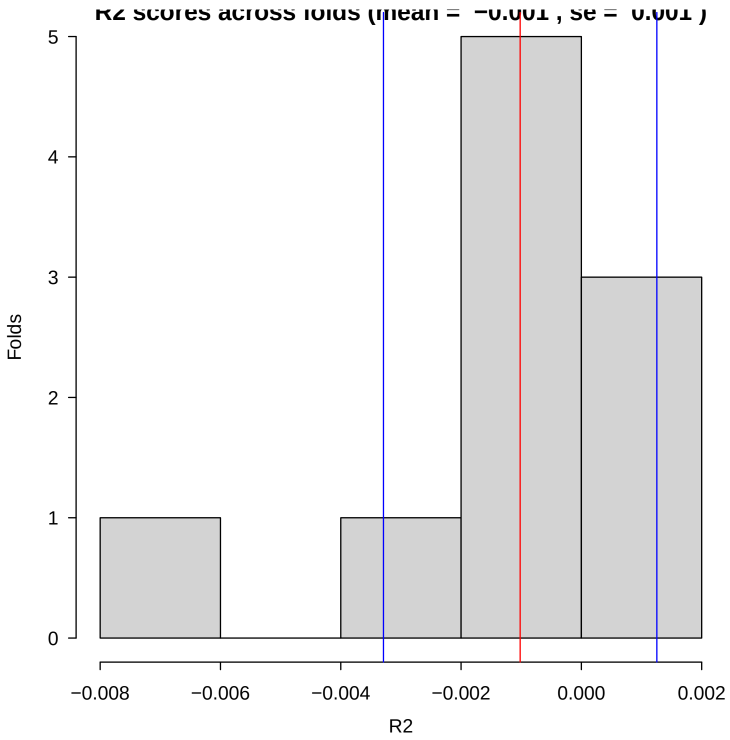
```
(R2.1_2 <- 1 - anova.1_2$RSS[2]/anova.1_2$RSS[1])
```

```
## [1] 0.08776623
```

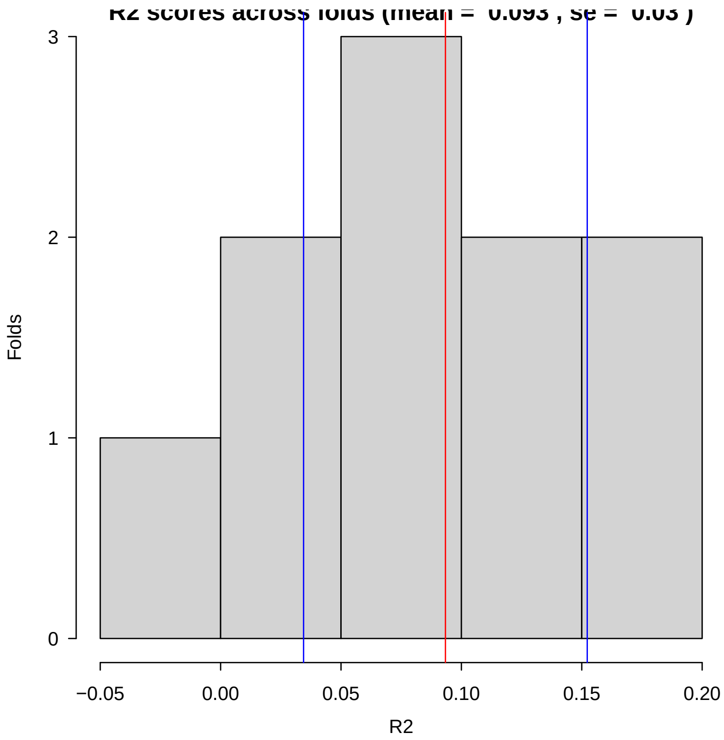
```
(R2.adj.1_2 <- 1 - (anova.1_2$RSS[2]/mod.lm2$df.residual)/(anova.1_2$RSS[1]/mod.lm1$df.residual))
```

```
## [1] 0.08597049
```

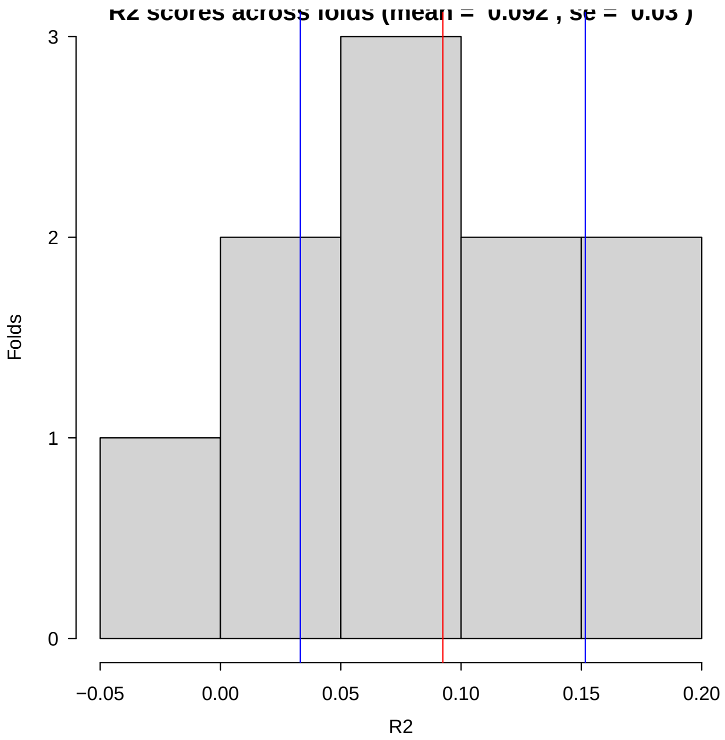
```
n.folds <- 10
folds <- cut(seq(1,n),breaks=n.folds,labels=FALSE)
folds <- sample(folds, replace = FALSE)


MSE.0 <- array(data=0, dim = n.folds)
MSE.1 <- array(data=0, dim = n.folds)
MSE.2 <- array(data=0, dim = n.folds)


#Cross Validation
for(i in 1:n.folds){
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- data_no_g11[testIndexes, ]
  trainData <- data_no_g11[-testIndexes, ]

  model.0.cv <- lm(p_trans ~ 1, data = trainData)
  model.1.cv <- lm(p_trans ~ generation, data = trainData)
  model.2.cv <- lm(p_trans ~ s_demo+generation, data = trainData)

  pred.0 <- predict(model.0.cv, newdata = testData)
  pred.1 <- predict(model.1.cv, newdata = testData)
  pred.2 <- predict(model.2.cv, newdata = testData)

  MSE.0[i] <- sum((testData$p_trans - pred.0)^2)/nrow(testData)
  MSE.1[i] <- mean((testData$p_trans - pred.1)^2)
  MSE.2[i] <- mean((testData$p_trans - pred.2)^2)

}

# Now we can calculate all of the cross-validated R2 - notice that these are arrays that have the size o

R2.cv.0_1 <- 1 - MSE.1/MSE.0
R2.cv.1_2 <- 1 - MSE.2/MSE.1
R2.cv.0_2 <- 1 - MSE.2/MSE.0

# Get means and SEs
R2.cv.m.0_1 <- mean(R2.cv.0_1)
R2.cv.m.1_2 <- mean(R2.cv.1_2)
R2.cv.m.0_2 <- mean(R2.cv.0_2)

R2.cv.se.0_1 <- sqrt(sum((R2.cv.0_1 - R2.cv.m.0_1)^2)/(n.folds-1))*sqrt(1/n.folds + 1/(n.folds-1))
R2.cv.se.1_2 <- sqrt(sum((R2.cv.1_2 - R2.cv.m.1_2)^2)/(n.folds-1))*sqrt(1/n.folds + 1/(n.folds-1))
R2.cv.se.0_2 <- sqrt(sum((R2.cv.0_2 - R2.cv.m.0_2)^2)/(n.folds-1))*sqrt(1/n.folds + 1/(n.folds-1))

hist(R2.cv.0_1, xlab="R2", ylab="Folds",
     main=paste("R2 scores across folds (mean = ",
                  round(R2.cv.m.0_1, 3), ", se = ", round(R2.cv.se.0_1, 3), ")"))
abline(v=R2.cv.m.0_1, col="red")
abline(v=R2.cv.m.0_1 - 1.96*R2.cv.se.0_1, col="blue")
abline(v=R2.cv.m.0_1 + 1.96*R2.cv.se.0_1, col="blue")
```



```
hist(R2.cv.1_2, xlab="R2", ylab="Folds",
     main=paste("R2 scores across folds (mean = ",
                  round(R2.cv.m.1_2, 3), ", se = ", round(R2.cv.se.1_2, 3), ")"))
abline(v=R2.cv.m.1_2, col="red")
abline(v=R2.cv.m.1_2 - 1.96*R2.cv.se.1_2, col="blue")
abline(v=R2.cv.m.1_2 + 1.96*R2.cv.se.1_2, col="blue")
```



```r
hist(R2.cv.0_2, xlab="R2", ylab="Folds",
     main=paste("R2 scores across folds (mean = ",
                round(R2.cv.m.0_2, 3), ", se = ", round(R2.cv.se.0_2, 3), ")"))
abline(v=R2.cv.m.0_2, col="red")
abline(v=R2.cv.m.0_2 - 1.96*R2.cv.se.0_2, col="blue")
abline(v=R2.cv.m.0_2 + 1.96*R2.cv.se.0_2, col="blue")
```



```
# Print some results
print(sprintf(’Model 1 R2=%.2f R2adj=%.2f R2cv=%.2f +- %.3f’, summary(mod.lm1)$r.squared, summary(mod.lm

## [1] "Model 1 R2=0.00 R2adj=-0.00 R2cv=-0.00 +- 0.002"

print(sprintf(’Model 2 R2=%.2f R2adj=%.2f R2cv=%.2f +- %.3f’, summary(mod.lm2)$r.squared, summary(mod.lm

## [1] "Model 2 R2=0.09 R2adj=0.08 R2cv=0.09 +- 0.059"

print(sprintf(’Model 1vs2 R2=%.2f R2adj=%.2f R2cv=%.2f +- %.3f’, R2.1_2, R2.adj.1_2, R2.cv.m.1_2, 1.96*

## [1] "Model 1vs2 R2=0.09 R2adj=0.09 R2cv=0.09 +- 0.059"

mod.lme<-lme(p_trans ~ s_demo+generation, random = ~ 1 |participant_id , data=data_no_g11,method = ’ML’)
summary(mod.lme)

## Linear mixed-effects model fit by maximum likelihood
##   Data: data_no_g11
##        AIC      BIC    logLik
##   314.4823 335.6641 -152.2411
##
## Random effects:
##  Formula: ~1 | participant_id
##         (Intercept)  Residual
## StdDev:   0.3051967 0.1144488
##
## Fixed effects:  p_trans ~ s_demo + generation
##                  Value  Std.Error  DF   t-value p-value
## (Intercept)  0.4306833 0.04062627 508 10.601103  0.0000
## s_demo       0.2355712 0.03369611 508  6.991051  0.0000
## generation  -0.0035821 0.00540638 508 -0.662560  0.5079
##  Correlation: 
##            (Intr) s_demo
## s_demo     -0.451       
## generation -0.799 -0.042
## 
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max 
## -0.7099922 -0.2058287  0.0351722  0.2493769  0.6441548 
## 
## Number of Observations: 511
## Number of Groups: 511
```

```
compareCoefs(mod.lm2, mod.lme)
```

```
## Warning in compareCoefs(mod.lm2, mod.lme):  models to be compared are of different classes
```

```
## Calls:
## 1: lm(formula = p_trans ~ s_demo + generation, data = data_no_g11)
## 2: lme.formula(fixed = p_trans ~ s_demo + generation, data = data_no_g11, random = ~1 
##   | participant_id, method = "ML")
## 
##              Model 1  Model 2
## (Intercept)   0.4307   0.4307
## SE            0.0406   0.0405
##                              
## s_demo        0.2356   0.2356
## SE            0.0337   0.0336
##                              
## generation  -0.00358 -0.00358
## SE           0.00541  0.00539
## 
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22631)
## 
## Matrix products: default
## 
## 
## locale:
## [1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8   
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.utf8    
## 
## time zone: America/Los_Angeles
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
```

```
## [1] car_3.1-2      carData_3.0-5  lattice_0.21-8 nlme_3.1-163
##
## loaded via a namespace (and not attached):
##  [1] digest_0.6.33     fastmap_1.1.1     xfun_0.40         abind_1.4-5
##  [5] knitr_1.44        htmltools_0.5.6   rmarkdown_2.24    tinytex_0.46
##  [9] cli_3.6.1         grid_4.3.1        compiler_4.3.1    highr_0.10
## [13] rstudioapi_0.15.0 tools_4.3.1       evaluate_0.21     yaml_2.3.7
## [17] rlang_1.1.1
```

```
Sys.time()
```

```
## [1] "2024-05-05 19:39:19 PDT"
```