

```
#####
#####
##### グラフ3 (実践2) #####
##### フリーシャッター課題 #####
#####
#####

# 主要なライブラリを読み込みます

#install.packages(ggplot2)
library(ggplot2)

#install.packages(dplyr)
library(dplyr)

# まずは以下のCSVを読み込んでください。

url = "https://lab.kenrikodaka.com/_download/csv/frees shutter_230601.csv"
dat_230601 = read.csv(url)
head(dat_230601)

# seat a1 b1 b2 b3 seat_ix seat_iy
#1 192 2.2 1.5 2.7 5.6 12 16
#2 168 0.8 4.4 10.0 14.7 12 14
#3 59 0.6 0.6 0.8 0.9 11 5
#4 60 2.9 3.4 5.3 9.9 12 5
#5 89 6.2 0.3 10.2 18.1 5 8
#6 180 0.1 0.1 0.5 0.8 12 15

# 集団フリーシャッター実験
# 実験A (20秒間の間に1度だけシャッターを押す)
# 実験B (20秒間の間に最大で3回シャッターを押す)

#seat (座席番号)
#a1 (実験Aのシャッター時間)
#b1 (実験Bのシャッター時間:1回目)
#b2 (実験Bのシャッター時間:2回目)
#b3 (実験Bのシャッター時間:3回目)
#seat_ix (座席の左右位置:1-12)
#seat_iy (座席の奥行き位置:1-22)

# データ数 (119人のデータ)
n = nrow(dat_230601)
n
# [1] 119

# dat0を、以下のような形式に整理し直します。
## - 行数を4倍とし、新たに変数「exp」と「order」を用いて、
## - 個々の時間の、実験属性 (A or B) と順序 (1、2、3回目) を紐づけます。

seat0 = rep(dat_230601$seat,4) #座席番号 (参加者ID)
seatx = rep(dat_230601$seat_ix,4) #座席位置 (Y)
seaty = rep(dat_230601$seat_iy,4) #座席位置 (Y)
time0 = c(dat_230601$a1,dat_230601$b1,dat_230601$b2,dat_230601$b3) #シャッター時間
exp0 = factor(c(rep("Single",n),rep("Triple",3*n))) #実験の種類 (A or B)
order0 = factor(c(rep(1,n),rep(1,n),rep(2,n),rep(3,n))) #何回目のシャッターか

#データフレーム化します。
dat = data.frame(seat=seat0,seatx=seatx,seaty=seaty,
                 time=time0,exp=exp0,order=order0)

dat
# 以下のような構造となっています。
str(dat)
#'data.frame':      476 obs. of  4 variables:
#$ seat : int  192 168 59 60 89 180 223 213 214 191 ...
#$ seatx: int  16 14 5 5 8 15 19 18 18 16 ...
```

```
#$ seaty: int 12 12 11 12 5 12 7 9 10 11 ...
#$ time : num 2.2 0.8 0.6 2.9 6.2 0.1 14.8 0.7 1.9 0.3 ...
#$ exp : Factor w/ 2 levels "Single","Triple": 1 1 1 1 1 1 1 1 1 ...
#$ order: Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 ...
```

```
#-----
#
# [Graph1_ShutterHistogram]
# シャッターのヒストグラム
#-----
#-----
```

```
#-----
# Graph1A (Single実験)
#-----
```

```
# Single実験のデータフレームを取得
dat_si = dat[dat$exp=="Single",]
```

```
# [ggplot]
## x軸をtimeに設定
gp_a = ggplot(dat_si,aes(x=time))
```

```
# [geom_histogram] ヒストグラムの作成
## 一つのピンの幅を0.1sec、塗りつぶしを白、枠線を黒
gp_a = gp_a + geom_histogram(binwidth=0.1, fill="white",colour="black")
```

```
# Y軸 (Count) の最大を10とする
gp_a = gp_a + scale_y_continuous(limits=c(0,10),breaks=seq(0,10,by=2))
# X軸の範囲と目盛の位置を決める
gp_a = gp_a + scale_x_continuous(limits=c(0,20),breaks=seq(0,20,by=5))
```

```
# [labs]
## グラフのタイトル、XY軸のタイトル
gp_a = gp_a + labs(title = "Graph1A_ShutterHistogram", x = "Time[s]", y = "Count")
```

```
# [theme]
gp_a = gp_a + theme(
  # 縦横比を1:3
  aspect.ratio=1/3,
  # ペースを太字、サイズ36に
  text = element_text(face = "bold", size = 36),
  # 軸の目盛りのサイズ
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25),
  # 凡例関係の設定
  legend.text = element_text(size=20),
  legend.title = element_text(size=25))
```

```
gp_a #out Graph1A_ShutterHistogram
```

```
#-----
# Graph1B (Triple実験)
#-----
```

```
# Triple実験のデータフレームを取得
dat_tri = dat[dat$exp=="Triple",]
```

```
# [ggplot]
## x軸をtimeに設定
gp_b = ggplot(dat_tri,aes(x=time))
gp_b = gp_b + geom_histogram(binwidth=0.1, fill="white",colour="black")
gp_b = gp_b + scale_y_continuous(limits=c(0,10),breaks=seq(0,10,by=2))
gp_b = gp_b + scale_x_continuous(limits=c(0,20),breaks=seq(0,20,by=5))
gp_b = gp_b + labs(title = "Graph1B_ShutterHistogram", x = "Time[s]", y = "Count")
```

```

# [theme]
gp_b = gp_b + theme(
  aspect.ratio=1/3,
  text = element_text(face = "bold", size = 36),
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25))
gp_b #out Graph1B_ShutterHistogram

#-----
# Graph1C (Triple実験、順序別)
#-----

# [ggplot]
## x軸をtimeに設定、塗りつぶしをorderに設定
gp_c = ggplot(dat_tri,aes(x=time,fill=order))

# 積み上げ順序の反転 (factor順序：下から上へ)
## 1->2->3の順に積み上げるようにする
gp_c = gp_c + geom_histogram(binwidth=0.1,colour="black",
  position = position_stack(reverse = TRUE))

gp_c = gp_c + scale_y_continuous(limits=c(0,10),breaks=seq(0,10,by=2))
gp_c = gp_c + scale_x_continuous(limits=c(0,20),breaks=seq(0,20,by=5))
gp_c = gp_c + labs(title = "Graph1C_ShutterHistogram", x = "Time[s]", y = "Count")

# 塗りつぶしの対応を決める。
## orderはfactor (非数値) のため、limits=c(1,2,3)では不可
gp_c = gp_c + scale_fill_discrete(
  limits=c("1","2","3"),labels=c("1st","2nd","3rd"))

# [theme]
gp_c = gp_c + theme(
  aspect.ratio=1/3,
  text = element_text(face = "bold", size = 36),
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25))
gp_c #out Graph1C_ShutterHistogram

#-----
# Graph1D (Triple実験、順序別)
#-----

gp_d = ggplot(dat_tri,aes(x=time,fill=order))
gp_d = gp_d + geom_histogram(binwidth=0.1,colour="black")
gp_d = gp_d + scale_y_continuous(limits=c(0,10),breaks=seq(0,10,by=5))
gp_d = gp_d + scale_x_continuous(limits=c(0,20),breaks=seq(0,20,by=5))
gp_d = gp_d + scale_fill_discrete(
  limits=c("1","2","3"),labels=c("1st","2nd","3rd"))
gp_d = gp_d + labs(title = "Graph1D_ShutterHistogram", x = "Time[s]", y = "Count")
gp_d = gp_d + theme(
  aspect.ratio=1/3,
  text = element_text(face = "bold", size = 36),
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25))

# facetをorderで分割、factの背景なども整える
gp_d = gp_d + facet_grid(order ~ .)
gp_d = gp_d + theme(
  strip.text = element_text(size = 25),
  strip.background =
    element_rect(fill="gray",colour="black",size=1))

gp_d #out Graph1D_ShutterHistogram

```

```

#-----

```

```

#-----
# [Graph2_ShutterBox]
# シャッター時間のボックスプロット
#-----
#-----

# [ggplot]
## x軸をexp、y軸にtime、塗りつぶしをorderに設定
gp = ggplot(dat,aes(x=exp,y=time,fill=order))

# [geom_boxplot]ボックスプロットによる描画
gp = gp + geom_boxplot()

gp = gp + scale_x_discrete(name="Experiment")
gp = gp + scale_y_continuous(name="Time [sec]")

gp = gp + scale_fill_discrete(
  limits=c("1","2","3"),labels=c("1st","2nd","3rd"))

gp = gp + labs(title = "Graph2_ShutterBox",
               x = "Experiment", y = "Time[s]")

gp = gp + theme(
  aspect.ratio=3/2,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25),
  #凡例の位置の調整、背景を加える
  legend.position=c(1.05,1.00),
  legend.justification=c(0,1),
  legend.background =
    element_rect(fill = "white", colour = "black"))

gp

#-----
#-----
# [Graph3_ShutterComma]
# コンマ何秒でシャッターを押していたか
#-----
#-----

# オリジナルのデータは476行
nrow(dat)
#[1] 476

# datには欠損値が42ある。
sum(is.na(dat$time))

# 欠損値を除いたdat_を作成 (行数434)
dat_ = dat[is.na(dat$time)==FALSE,]
nrow(dat_)
# [1] 434

#コンマN秒に相当するtime2属性を追加したデータフレームdat2を作成
## roundは四捨五入、digitsは桁の指定
dat2 = mutate(dat_,time2 = round(10*(time - floor(time)),digits=1))

## 正しく変換できていることを確認
dat2[,c("time","time2")]
#   time time2
# 1   2.2     2
# 2   0.8     8
# 3   0.6     6
# 4   2.9     9
# 5   6.2     2

```

```

# 6    0.1    1
# 7   14.8    8
# 8    0.7    7
# 9    1.9    9
# ...

# expごとのtime2の集計
table(dat2$time2,dat2$exp)
#   Single Triple
# 0     17     42
# 1      5     25
# 2     16     32
# 3     13     32
# 4     10     35
# 5     11     28
# 6     11     20
# 7     12     31
# 8     14     33
# 9     10     37

# このテーブルをdata.frame化する
dat_comma = as.data.frame(table(dat2$time2,dat2$exp))
colnames(dat_comma) = c("n","exp","count");dat_comma
#   n   exp count
# 1  0 Single   17
# 2  1 Single    5
# 3  2 Single   16
# 4  3 Single   13
# 5  4 Single   10
# 6  5 Single   11
# 7  6 Single   11
# 8  7 Single   12
# 9  8 Single   14
# 10 9 Single   10
# 11 0 Triple   42
# 12 1 Triple   25
# 13 2 Triple   32
# 14 3 Triple   32
# 15 4 Triple   35
# 16 5 Triple   28
# 17 6 Triple   20
# 18 7 Triple   31
# 19 8 Triple   33
# 20 9 Triple   37

#-----
# Graph3A (X軸を0:9の順で)
#-----

# number属性をfactorからintegerに変更します。
class(dat_comma$n)
#[1] "factor"
## - 文字列にしてから整数へ
## - !!) 直接as.integerをするとファクタのIDが対象となります
dat_comma$n = as.integer(as.character(dat_comma$n))
class(dat_comma$n)
#[1] "integer"

gp_a = ggplot(dat_comma,aes(x=n,y=count,colour=exp))
gp_a = gp_a + geom_line(size=1.5)
gp_a = gp_a + geom_point(size=10)
gp_a = gp_a + scale_x_continuous(limits=c(-0.5,9.5),breaks=0:9)
gp_a = gp_a + scale_y_continuous(limits=c(0,50),breaks=seq(0,50,by=10))
gp_a = gp_a + labs(title="Graph3A_ShutterComma",x="Comma.N",y="Count")
#g3 = g3 + scale_colour_discrete()
gp_a = gp_a + theme(
  aspect.ratio=3/4,
  text = element_text(face = "bold", size = 36),
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25)
)
gp_a #out Graph3A_ShutterComma

#-----

```

```

# Graph3B (X軸を6,7,8,9,0,1,2,3,4,5の順で)
#-----

# n = c(0,1,2,3,4,5,6,7,8,9)を
# n2 = c(4,5,6,7,8,9,0,1,2,3)にマッピング
## n2の0:9が、nの6:9,0:5に対応していることに注意

dat_comma$n
dat_comma = mutate(dat_comma,
                    n2 = if_else(n>=6,n-6,n+4))

gp_b = ggplot(dat_comma,aes(x=n2,y=count,colour=exp))
gp_b = gp_b + geom_vline(xintercept = 4, linetype = "dotted",
                          size=1.5,colour="black")

gp_b = gp_b + geom_line(size=1.5)
gp_b = gp_b + geom_point(size=10)

# ラベルをずらしてつけるのがポイント
gp_b = gp_b + scale_x_continuous(
  limits=c(-0.5,9.5),breaks=0:9,
  labels=c(6:9,0:5));
gp_b = gp_b + scale_y_continuous(limits=c(0,50),breaks=seq(0,50,by=10))

gp_b = gp_b + labs(title="Graph3B_ShutterComma",x="Comma.N",y="Count");
gp_b = gp_b + theme(
  aspect.ratio=3/4,
  text = element_text(face = "bold", size = 36),
  axis.text.x = element_text(size = 30),
  axis.text.y = element_text(size = 25)
)
gp_b #out Graph3B_ShutterComma

```

```

#-----
#-----
# [Graph4_Correlation]
# シャッターとシャッターの時間差の相関 (Triple実験)
#-----
#-----

# Triple実験のデータフレームを取得
dat_tri = dat[dat$exp=="Triple",]

# 以下、dat_として参照します。
dat_ = dat_tri;

#席順に並べる (昇順に)
dat_ = arrange(dat_,seat)
#arrange(dat_,desc(seat)) #降順
dat_ = dat_[c("seat","seatx","seaty","time","order")];

dat_[1:6,]
# seat seatx seaty time exp order
# 1 4 1 4 3.2 Triple 1
# 2 4 1 4 6.9 Triple 2
# 3 4 1 4 10.1 Triple 3
# 4 5 1 5 NA Triple 1
# 5 5 1 5 NA Triple 2
# 6 5 1 5 NA Triple 3

# orderごとのデータを切り出す
dat_t1 = dat_[dat_$order==1,]; dat_t1
dat_t2 = dat_[dat_$order==2,]; dat_t2
dat_t3 = dat_[dat_$order==3,]; dat_t3

```

```

# これらを列でバインド
dat_ =
  cbind(dat_t1[,c("seat","seatx","seaty","time")],
        dat_t2[,c("time")],
        dat_t3[,c("time")])

# 列名を変更
colnames(dat_) = c("seat","seatx","seaty","time1","time2","time3")
dat_[1:6,]
#   seat seatx seaty time1 time2 time3
# 1     4     1     4   3.2   6.9  10.1
# 4     5     1     5    NA    NA    NA
# 7     9     1     9   0.5   3.6  19.5
# 10    13    2     1   3.4  10.0  13.0
# 13    28    3     4   5.0   9.0  10.2
# 16    36    3    12   2.7   6.1   9.0

# シャッター1からシャッター2の時間差をtime21
# シャッター2からシャッター3の時間差をtime32として属名を追加
dat_ = mutate(dat_,time21 = time2-time1)
dat_ = mutate(dat_,time32 = time3-time2)
dat_[1:6,]
#   seat seatx seaty time1 time2 time3 time21 time32
# 1     4     1     4   3.2   6.9  10.1    3.7    3.2
# 4     5     1     5    NA    NA    NA     NA     NA
# 7     9     1     9   0.5   3.6  19.5    3.1   15.9
# 10    13    2     1   3.4  10.0  13.0    6.6    3.0
# 13    28    3     4   5.0   9.0  10.2    4.0    1.2
# 16    36    3    12   2.7   6.1   9.0    3.4    2.9

#-----
# Graph4A (time1 と time21の相関図)
#-----

gp_a = ggplot(dat_, aes(x = time1, y = time21))

# [geom_point] 散布図 (同一行のxとyの関係)
gp_a = gp_a + geom_point(size=8,shape=21)

# [geom_smooth] 線形回帰直線
gp_a = gp_a + geom_smooth(method="lm", formula='y~x')

gp_a = gp_a + scale_x_continuous(limits=c(0,10),breaks=c(0,5,10))
gp_a = gp_a + scale_y_continuous(limits=c(0,10),breaks=c(0,5,10))

# [geom_segment] 任意の直線
# (x,y) to (xend,yend)の座標間に線を引く
gp_a = gp_a + geom_segment(x=0,y=0,xend=10,yend=10,
                           size=0.5,linetype="dotted")

gp_a = gp_a + labs(title = "Graph4A_Correlation1",
                   x = "Time1", y = "Time1 to Time2")
gp_a = gp_a + theme(
  aspect.ratio=1/1,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 20),
  axis.text.y = element_text(size = 20))

gp_a #out Graph4A_Correlation1

#-----
# Graph4B (time21 と time32の相関図)
#-----

gp_b = ggplot(dat_, aes(x = time21, y = time32))
gp_b = gp_b + geom_point(size=8,shape=21)
gp_b = gp_b + geom_smooth(method="lm", formula='y~x')
gp_b = gp_b + scale_x_continuous(limits=c(0,10),breaks=c(0,5,10))
gp_b = gp_b + scale_y_continuous(limits=c(0,10),breaks=c(0,5,10))
gp_b = gp_b + geom_segment(x=0,y=0,xend=10,yend=10,
                           size=0.5,linetype="dotted")
gp_b = gp_b + labs(title = "Graph4B_Correlation2",
                   x = "Time1 to Time2", y = "Time2 to Time3")
gp_b = gp_b + theme(

```

```

    aspect.ratio=1/1,
    text = element_text(face = "bold", size = 30),
    axis.text.x = element_text(size = 20),
    axis.text.y = element_text(size = 20))

gp_b #out Graph4B_Correlation2

#install.packages(gridExtra)
library(gridExtra)
grid.arrange(gp_a, gp_b, nrow=1) #1行で並べる

#-----
# [Graph5_DirectionEffect]
# 隣接空間（方向）の影響
# 前の人に影響されるか？後ろの人に影響されるか？
#-----

# Single実験のデータを取り出す。
dat_si = dat[dat$exp=="Single",]
dat_ = dat_si

# 注目するのは、座席の位置と時間のみなので、それだけを切り出します。
dat_ = dat_[,c("seatx", "seaty", "time")]
dat_[1:6,]

# seatx seaty time
# 1    16    12  2.2
# 2    14    12  0.8
# 3     5    11  0.6
# 4     5    12  2.9
# 5     8     5  6.2
# 6    15    12  0.1

# まずは隣接関係にあるものたちのシャッター時間を算出する関数を作成します。
## 引数は、自分の座席のxy座標です。
## timeForward：前方（最大）3人の平均シャッター時間の算出
## timeNeighbor：両隣（最大）2人の平均シャッター時間の算出
## timeBackward：後方（最大）3人の平均シャッター時間の算出

timeForward = function(sx,sy){
  forward = dat_[dat_$seatx>=sx-1 & dat_$seatx<=sx+1 & dat_$seaty==sy-1,]
  mean(forward$time) #平均値（引数が空の場合NA）
}
timeNeighbor = function(sx,sy){
  neighbor = dat_[(dat_$seatx==sx-1 | dat_$seatx==sx+1) & dat_$seaty==sy,]
  mean(neighbor$time) #平均値（引数が空の場合NA）
}
timeBackward = function(sx,sy){
  backward = dat_[dat_$seatx>=sx-1 & dat_$seatx<=sx+1 & dat_$seaty==sy+1,]
  mean(backward$time) #平均値（引数が空の場合NA）
}

# dat_の各行に、
# 3方向区間のシャッター時間（time.f, time.n, time.b）を付加します。
#-----

## mutateはなぜかうまくいかない、、、。（なぜだ？）
# mutate(dat_,time.f = timeforward(seatx,seaty))
# mutate(dat_,time.n = timeneighbor(seatx,seaty))
# mutate(dat_,time.b = timebackward(seatx,seaty))

```



```

## dat_と同じ行数のベクトルを3つ作成
time.f = vector("double",nrow(dat_))
time.n = vector("double",nrow(dat_))
time.b = vector("double",nrow(dat_))

## それぞれに、3方向区間の撮影時間を代入
for(i in 1:nrow(dat_)){
  seatx = dat_[i,c("seatx")]
  seaty = dat_[i,c("seaty")]
  time.f[i] = timeForward(seatx,seaty)
  time.n[i] = timeNeighbor(seatx,seaty)
  time.b[i] = timeBackward(seatx,seaty)
}

## データフレームの属性を新たに追加
dat_$time.f = time.f
dat_$time.n = time.n
dat_$time.b = time.b

## 最初の6行を確認します。
## NaNは対象区間に誰もいない場合と対応します。
dat_[1:6,]
#   seatx seaty time   time.f time.n time.b
# 1    16    12  2.2  0.300000  1.15   NaN
# 2    14    12  0.8  20.000000  9.60   NaN
# 3     5     11  0.6      NaN     NaN  2.45
# 4     5     12  2.9  0.600000  2.00   NaN
# 5     8     5  6.2  8.566667 13.50 15.80
# 6    15    12  0.1 10.150000  1.50   NaN

#-----

# 3区間（前方・隣接・後方）の撮影時間と比べて
# 早くシャッターを押したか（EARLY）か遅く押したか（LATE）
# mutate関数で、新たに属性を追加し並べる
dat_ = mutate(dat_, order.f = if_else(time - time.f>=0,"LATE","EARLY"))
dat_ = mutate(dat_, order.n = if_else(time - time.n>=0,"LATE","EARLY"))
dat_ = mutate(dat_, order.b = if_else(time - time.b>=0,"LATE","EARLY"))

# 最初の20行を、1-3,7-9列目のみを確認
dat_[1:20,c(1:3,7:9)]
#   seatx seaty time order.f order.n order.b
# 1    16    12  2.2   LATE   LATE   <NA>
# 2    14    12  0.8  EARLY  EARLY  <NA>
# 3     5     11  0.6   <NA>  <NA>  EARLY
# 4     5     12  2.9   LATE   LATE  <NA>
# 5     8     5  6.2  EARLY  EARLY  EARLY
# 6    15    12  0.1  EARLY  EARLY  <NA>
# 7    19     7 14.8   LATE   LATE   LATE
# 8    18     9  0.7   <NA>  EARLY  EARLY
# 9    18    10  1.9  EARLY  <NA>  EARLY
# 10   16    11  0.3  EARLY  <NA>  EARLY
# 11   17     9  7.4   LATE   LATE   LATE
# 12   16     9  9.8   LATE   LATE  EARLY
# 13    6     7  2.5   <NA>  EARLY  EARLY
# 14    6     8  6.1  EARLY  LATE   LATE
# 15    7    10  3.0   LATE   LATE  EARLY
# 16    7     9  3.7   LATE   LATE   LATE
# 17   13    12 19.1   LATE   LATE  <NA>
# 18   17    12  2.2   LATE   LATE  <NA>
# 19   11     9  9.7   LATE   LATE  EARLY
# 20   11    10 16.8   LATE   LATE   LATE

## (例)
### - 席(8,5)は前方・両隣・後方よりも早くシャッターを押している。
# 5     8     5  6.2  EARLY  EARLY  EARLY
### - 席(7,10)は前方・両隣よりも遅くシャッターを押し、
### - 後方よりも早くシャッターを押している。
# 15    7    10  3.0   LATE   LATE  EARLY

# 前方との比較：ほぼ変わらない
table(dat_$order.f)

```

```

# EARLY LATE
# 43 44

# 両隣との比較：ほぼ変わらない
table(dat_$order.n)
# EARLY LATE
# 43 44

# 後方との比較：後方よりも早く押す傾向
table(dat_$order.b)
# EARLY LATE
# 52 36

# グラフにするために、新たなデータフレームを作成

dat_dir = rbind(
  data.frame(direction = "FORWARD", dorder = dat_$order.f),
  data.frame(direction = "NEIGHBOR", dorder = dat_$order.n),
  data.frame(direction = "BACKWARD", dorder = dat_$order.b))

# NAはとる
dat_dir = dat_dir[is.na(dat_dir$dorder)==FALSE,]
dat_dir

# direction dorder
# 1 FORWARD LATE
# 2 FORWARD EARLY
# 4 FORWARD LATE
# ...
# 120 NEIGHBOR LATE
# 121 NEIGHBOR EARLY
# 123 NEIGHBOR LATE
# ...
# 241 BACKWARD EARLY
# 243 BACKWARD EARLY
# 245 BACKWARD LATE

## x軸をgenderに、枠線と塗りつぶしをpreferenceに設定
gp = ggplot(dat_dir,aes(x=direction, fill=dorder))
gp = gp + geom_bar(position=position_dodge(width=0.9), colour="black")
gp = gp + scale_x_discrete(limits=c("FORWARD","NEIGHBOR","BACKWARD"))
gp = gp + scale_y_continuous(limits=c(0,60))
gp = gp + scale_fill_discrete(limits=c("EARLY","LATE"),name="ORDER")
gp = gp + labs(title = "Graph5_DirectionEffect",
               x = "DIRECTION", y = "COUNT")
gp = gp + theme(
  aspect.ratio=2/3,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 20),
  axis.text.y = element_text(size = 20))

gp #out Graph5_DirectionEffect

# (発展) カイ二乗分布
## p>0.05で、方向による影響は統計的には認められない
chisq.test(dat_dir$direction, dat_dir$dorder, correct=F)
# data: df_dir$direction and df_dir$dorder
# X-squared = 2.1902, df = 2, p-value = 0.3345

```

```

#-----
# 【確認課題】
#-----

```

```

#-----
# 【Work5_CommaDistribution】
# 20秒間で10回押すフリーシャッター実験の結果を用いて、
# 「コンマN秒」の分布を作成してください。
# この際、前半1~5回目のシャッターを「1ST」
# 後半6~10回目のシャッターを「2ND」として、

```

```

# 前後半別で分布を重ねて示してください。

# 締切は11月中とします。
# ファイル名は「2250xx_work5.R」としてください。
#-----

# [10times フリーシャッター実験]
# 20秒の間にシャッターを10回押してください実験の結果（24年7月11日）

url = "https://lab.kenrikodaka.com/_download/csv/freeshutter_240711.csv"
dat_240711 = read.csv(url)

# 参加者は109人
nrow(dat_240711)
#[1] 109

# 最初と最後の6行
dat_240711[c(1:6,104:109),]
#   seatx seaty  t1  t2  t3  t4  t5  t6  t7  t8  t9  t10
# 1     9     2  0.3 1.1 1.4 1.6 1.7 2.1 2.6 2.7 3.6 5.1
# 2    12     9  0.4 1.2 1.6 2.3 3.0 13.3 14.0 15.0 16.2 19.0
# 3     4     4  1.2 2.9 4.9 6.8 9.0 10.7 12.5 15.0 17.2 19.2
# 4    10     2  6.0 6.5 6.6 6.9 7.2 7.5 7.7 7.9 8.2 8.4
# 5     6    14  1.3 2.7 3.2 8.2 8.9 11.2 13.3 17.3 17.7 18.0
# 6     9     6  1.1 1.5 2.1 2.4 3.5 4.0 5.4 7.7 9.8 11.1
# 104    9     1  1.1 2.4 3.8 5.5 6.9 8.3 9.2 10.8 12.1  NA
# 105   12     4  1.2 2.4 4.2 5.2 6.2 7.4 8.2 9.4 10.5 11.5
# 106   12    15  2.3 3.4 4.5 6.7 10.3 11.8 16.9 18.9 20.1 21.0
# 107   10    15  0.9 1.6 2.2 3.5 4.6 5.2 5.9 6.5 8.1 9.9
# 108   12    15  2.0 3.6 4.6 5.7 10.2 11.5 16.7 17.9 18.7 19.9
# 109   11     2  2.9 3.1 3.6 4.0 5.2 14.2 15.7 16.0 16.7 18.8

## seatx : 座席の水平方向（生徒視点で左から1,2,.....）
## seaty : 座席の奥行き方向（教卓に近い方から1,2,...）
## t1, t2, t3, ... : 1回目のシャッター時間, 2回目のシャッター時間, ...
## NAは欠損データ（例えば、104行目のt10）

#-----
# 途中までヒント（必ずしも、以下の方法に従う必要はありません）
#-----

# 以下、dat_として扱います。
dat_ = dat_240711

# t1~t10について、少数第一位の数字を求めます。

comma1 = round(10*(dat_$t1-floor(dat_$t1)),digits=1)
comma2 = round(10*(dat_$t2-floor(dat_$t2)),digits=1)
comma3 = round(10*(dat_$t3-floor(dat_$t3)),digits=1)
comma4 = round(10*(dat_$t4-floor(dat_$t4)),digits=1)
comma5 = round(10*(dat_$t5-floor(dat_$t5)),digits=1)
comma6 = round(10*(dat_$t6-floor(dat_$t6)),digits=1)
comma7 = round(10*(dat_$t7-floor(dat_$t7)),digits=1)
comma8 = round(10*(dat_$t8-floor(dat_$t8)),digits=1)
comma9 = round(10*(dat_$t9-floor(dat_$t9)),digits=1)
comma10 = round(10*(dat_$t10-floor(dat_$t10)),digits=1)

# これ以降は、Graph3Aを参考にすすめてください。

```