

```
#####
#####
##### グラフ4 (実践3) #####
##### ブッダの耳錯覚実験 #####
#####
#####

# 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/buddhaexp2024.csv"
source = read.csv(url)

str(source)

# 'data.frame':      16 obs. of  29 variables:
# $ SBJ      : int   1 2 3 4 5 6 7 8 9 10 ...
# $ Q1T0B0   : num   1 1.5 0 1 2 0 5 1 1.5 0.5 ...
# $ Q1T0B1   : num   3.5 4 5 3 3 1 5.5 2 4 3 ...
# $ Q1T1B0   : num   2.5 1.5 0 3 4 0 4.5 1 2.5 0 ...
# $ Q1T1B1   : num   5.5 4.5 6 3.5 5.5 1.5 5.5 3.5 5.5 3 ...
# $ Q2T0B0   : num   0 2.5 0 1.5 2 0 5 1 1.5 1.5 ...
# $ Q2T0B1   : num   2.5 3 3.5 0 4 0 5.5 1 3 4 ...
# $ Q2T1B0   : num   1.5 1.5 0 1.5 4 0 5 1.5 2 2.5 ...
# $ Q2T1B1   : num   3 4.5 4.5 3 5.5 0 5 1 4 6 ...
# $ Q3T0B0   : num   0 0.5 0 0.5 0 0 2 0 0 0 ...
# $ Q3T0B1   : num   2.5 1.5 0 1.5 0 0 1.5 1 1.5 0 ...
# $ Q3T1B0   : num   1.5 0 0 3 0.5 0 1 0 0 0 ...
# $ Q3T1B1   : num   3 0 0 3 0 0 1 0 1 0.5 ...
# $ Q4T0B0   : num   0 0 0 2.5 0 0 0 1.5 0 0 ...
# $ Q4T0B1   : num   0 0 0 2 0.5 0 0 0.5 1.5 0 ...
# $ Q4T1B0   : num   0 0 0 2.5 1.5 0 0 0.5 0 0 ...
# $ Q4T1B1   : num   0 0 0 2 0.5 0 0 0 1 0 ...
# $ E0T0B0   : num   4.25 13.15 -0.25 3.6 -2.1 ...
# $ E0T0B1   : num   15.45 14.15 3.05 8.2 -1.3 ...
# $ E0T1B0   : num   20.1 11.1 1.1 -1.5 1.4 4.15 5.2 -0.4 5.5 1.25 ...
# $ E0T1B1   : num   1.7 11.1 0.2 -0.2 3.3 0.75 6.4 -0.2 5 2.95 ...
# $ E1T0B0   : num   36.3 16.55 4.55 5.6 -1.95 ...
# $ E1T0B1   : num   49 55 55.5 9.1 40.9 ...
# $ E1T1B0   : num   32.8 41.4 4.8 5.2 30 ...
# $ E1T1B1   : num   50.2 46.4 77.3 6.8 75 ...
# $ ORDER   : int   0 1 0 0 1 0 1 1 0 1 ...

# SBJ 被験者のID (1-16)

# 変数名 [XX] [TaBb] の説明

# [XX] Q1-Q4はアンケートの設問に対する主観評価値 (0-6の7段階)
## (主観評価実験)
## Q1: 「自分の耳たぶが通常よりも伸びている感じがした。」
## Q2: 「何も無い空間に「見えない」皮膚が存在しているように感じた。」
## Q3: 「自分の耳全体が下に移動しているような感じがした。」
## Q4: 「自分の耳全体の面積が小さくなる感じがした。」
## (行動実験)
## E0: 耳上端の主観位置の錯覚前後の降下量 (cm)
## E1: 耳下端の主観位置の錯覚前後の降下量 (cm)

# [TaBb] Taは体験者側、Bbは実験者側の条件
## T0B0: (体験者が) 耳上端に触れない x (実験者が) 耳下端をつまむ (だけ)
## T0B1: (体験者が) 耳上端に触れない x (実験者が) 耳下端の下を引っ張る
## T1B0: (体験者が) 耳上端の上をつまむ x (実験者が) 耳下端をつまむ (だけ)
## T0B1: (体験者が) 耳上端の上をつまむ x (実験者が) 耳下端の下を引っ張る
```

```

# ORDER 行動実験の順序（今回の演習では、ORDERは無視します）
## 0：T1B1→T0B1（錯覚条件に関してT1が先、その後にT0）
## 1：T0B1→T1B1（錯覚条件に関してT0が先、その後にT1）

# データ数（16人のデータ）
n = nrow(source)
# [1] 16

#-----
# グラフ作成用に2種類のデータフレームを整形する

# dat_Q：アンケート実験に関するデータフレーム
## rate（評価値）、stmt（質問項目）、etop（耳上端条件）、ebot（耳下端条件）

# dat_B：行動実験に関するデータフレーム
## drift（位置変化量）、epos（変化部位）、etop（耳上端条件）、ebot（耳下端条件）
#-----

# sourceをdat_にコピーします。
dat_ = source

# 実験参加者ID（長さ：16x16）を引き延ばす
v_sbj = rep(dat_[,1],16)

# アンケートの全ての評価値を格納したvRate（長さ：16x16）を作成
# 2列目（Q1T0B0）から17列目（Q4T1B1）をつなげる
v_rate = dat_[,2]
for(i in 3:17){
  v_rate = c(v_rate,dat_[,i])
}

v_stmt = c(rep("Q1",16*4),rep("Q2",16*4),rep("Q3",16*4),rep("Q4",16*4))
v_etop = rep(c(rep("T0",16*2),rep("T1",16*2)),4)
v_ebot = rep(rep(c(rep("B0",16),rep("B1",16)),2),4)

# アンケートに対するデータフレームを作成
dat_Q = data.frame(sbj = v_sbj, rate = v_rate, stmt = v_stmt, etop = v_etop, ebot = v_ebot)
dat_Q[1:5,]
#   sbj rate stmt etop ebot
# 1   1  1.0  Q1   T0   B0
# 2   2  1.5  Q1   T0   B0
# 3   3  0.0  Q1   T0   B0
# 4   4  1.0  Q1   T0   B0
# 5   5  2.0  Q1   T0   B0

# 実験参加者ID（長さ：16x8）を引き延ばす
v_sbj = rep(dat_[,1],8)

# 行動実験の全ての位置変化量を格納したvDrift（長さ：16x8）を作成
# 18列目（Q1T0B0）から25列目（Q4T1B1）をつなげる
v_drift = dat_[,18]
for(i in 19:25){
  v_drift = c(v_drift,dat_[,i])
}

v_epos = c(rep("TOP",16*4),rep("BOT",16*4))
v_etop = rep(c(rep("T0",16*2),rep("T1",16*2)),2)
v_ebot = rep(rep(c(rep("B0",16),rep("B1",16)),2),2)

# 行動実験（ドリフト）に対するデータフレームを作成
dat_D = data.frame(sbj = v_sbj, drift = v_drift, epos = v_epos,
                  etop = v_etop, ebot = v_ebot)
dat_D[1:5,]
#   sbj drift epos etop ebot
# 1   1  4.25  TOP   T0   B0
# 2   2 13.15  TOP   T0   B0
# 3   3 -0.25  TOP   T0   B0
# 4   4  3.60  TOP   T0   B0

```

# 5 5 -2.10 TOP T0 B0

```

#-----
#-----
# [Graph1_QuestionnaireBoxplot]
# アンケート結果をボックスプロットでまとめる
#-----
#-----

# [ggplot]
## interactionは、複数の変数の組み合わせをグラフの属性にマップできる
## interaction(etop,ebot)は「T0.B0」「T1.B0」「T0.B1」「T1.B1」の4つの要素を作る
gp = ggplot(dat_Q,aes(x=interaction(etop,ebot),
                        y=rate,fill=interaction(etop,ebot)))

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

gp = gp + scale_x_discrete(
  labels=c("T0B0","T1B0","T0B1","T1B1"))
gp = gp + scale_y_continuous(limits=c(0,6),breaks=0:6)
gp = gp + scale_fill_discrete(
  name = "CONDITION",
  labels=c("(T0B0) TOP:none, BOT:pinched",
            "(T1B0) TOP:self-pinched, BOT:pinched",
            "(T0B1) TOP:none, BOT:pulled",
            "(T1B1) TOP:self-pinched, BOT:pulled"))

gp = gp + labs(title = "Graph1_QuestionnaireBoxplot",
               x = "CONDITION", y = "RATE")

# gp = gp + facet_grid(stmt ~ .)
gp = gp + facet_grid(. ~ stmt)

gp = gp + theme(
  aspect.ratio=3/2,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 25),
  axis.text.y = element_text(size = 25),
  #凡例の位置の調整、背景を加える
  legend.position=c(1,1.08),
  legend.justification=c(1,0),
  legend.background =
    element_rect(fill = "white", colour = "black"),
  legend.text = element_text(size = 30))

gp

```

```

#-----
#-----
# [Graph2_QuestionnaireBar]
# アンケート結果をバープロットでまとめる
#-----
#-----

#個々の条件の平均値・標準誤差を算出する

# まずはdat_Qの構造を復習
## stmt (4) ・etop (2) ・ebot (2) の4x2x2=16通り
dat_Q[1:3,]
#   sbj rate stmt etop ebot
# 1   1  1.0  Q1  T0  B0
# 2   2  1.5  Q1  T0  B0
# 3   3  0.0  Q1  T0  B0

# 例えばQ1、T0、B0の平均値と標準誤差は以下の手順で算出

## (1)Q1,T0,B0のrateを全て取り出す(16人分)
rates = dat_Q[dat_Q$stmt=="Q1" & dat_Q$etop=="T0" & dat_Q$ebot=="B0",c("rate")];rates
#[1] 1.0 1.5 0.0 1.0 2.0 0.0 5.0 1.0 1.5 0.5 3.5 0.0 1.0 0.0 2.0 0.5

## (2)平均値の算出：meanを使う
mean(rates) #[1] 1.28125

## (3)標準誤差：標準偏差 (sd) をサンプルサイズの平方根で割る
sd(rates) / sqrt(length(rates)) #[1] 0.3414697
### 以下も同じ (varは分散、sdは分散の平方根)
sqrt(var(rates)) / sqrt(length(rates)) #[1] 0.3414697

# 平均値・標準誤差の組を16通りの条件で一気に吐き出すために関数を定義

getMeanSE = function(Qn,Tn,Bn){
  rates = dat_Q[dat_Q$stmt==Qn & dat_Q$etop==Tn & dat_Q$ebot==Bn,c("rate")]
  mean = mean(rates)
  SE = sd(rates) / sqrt(length(rates))

  c(mean,SE)
}

# 列名だけ存在する空のデータフレームを作成
dat_stat = data.frame(mean=as.numeric(),se=as.numeric(),
                      stmt=as.character(),etop=as.character(),ebot=as.character())
dat_stat
# [1] mean se   stmt etop ebot
# <0 rows> (or 0-length row.names)

for(Qn in c("Q1","Q2","Q3","Q4")){
  for(Tn in c("T0","T1")){
    for(Bn in c("B0","B1")){
      mean.se = getMeanSE(Qn,Tn,Bn)
      add_data = data.frame(mean=mean.se[1],se=mean.se[2],stmt=Qn,etop=Tn,ebot=Bn)
      dat_stat = rbind(dat_stat,add_data)
    }
  }
}

# これで準備完了!
# >dat_stat
#   mean      se stmt etop ebot
# 1 1.28125 0.3414697  Q1  T0  B0
# 2 3.28125 0.3817961  Q1  T0  B1
# 3 1.68750 0.4327119  Q1  T1  B0
# 4 4.59375 0.3169475  Q1  T1  B1
# 5 1.25000 0.3505947  Q2  T0  B0
# 6 2.46875 0.4017378  Q2  T0  B1
# 7 1.53125 0.3776814  Q2  T1  B0
# 8 3.53125 0.4017378  Q2  T1  B1
# 9 0.34375 0.1562500  Q3  T0  B0
# 10 1.09375 0.2858421  Q3  T0  B1
# 11 0.46875 0.2163559  Q3  T1  B0

```

```
# 12 0.84375 0.3119787 Q3 T1 B1
# 13 0.25000 0.1767767 Q4 T0 B0
# 14 0.28125 0.1511673 Q4 T0 B1
# 15 0.28125 0.1765925 Q4 T1 B0
# 16 0.21875 0.1366927 Q4 T1 B1
```

```
#-----
# [Graph2A_QuestionnaireBar]
# エラーバーなし
#-----

gp = ggplot(dat_stat,aes(x=interaction(etop,ebot), y=mean,
                             fill=interaction(etop,ebot)))
# geom_barは数え上げ、geom_colは数値をそのまま表示することに注意
## position_dodgeは横に並べるの意味
gp = gp + geom_col(position=position_dodge(width=0.9))

gp = gp + scale_x_discrete(labels=c("T0B0","T1B0","T0B1","T1B1"))
gp = gp + scale_y_continuous(limits=c(0,6),breaks=0:6)
gp = gp + scale_fill_discrete(
  name = "CONDITION",
  labels=c("(T0B0) Top:none, Bot:pinched",
            "(T1B0) Top:self-pinched, Bot:pinched",
            "(T0B1) Top:none, Bot:pulled",
            "(T1B1) Top:self-pinched, Bot:pulled"))

gp = gp + facet_grid(. ~ stmt)

gp = gp + labs(title="Graph2A_QuestionnaireBar",x="CONDITION",y="RATE")

gp = gp + theme(
  aspect.ratio=3/2,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 25),
  axis.text.y = element_text(size = 25),
  #凡例の位置の調整、背景を加える
  legend.position=c(1,1.08),
  legend.justification=c(1,0),
  legend.background =
    element_rect(fill = "white", colour = "black"),
  legend.text = element_text(size = 30))

gp

#-----
# [Graph2B_QuestionnaireBar]
# エラーバーあり
#-----

gp = ggplot(dat_stat,aes(x=interaction(etop,ebot), y=mean,
                             fill=interaction(etop,ebot)))

gp = gp + geom_col(position=position_dodge(width=0.9))

#[geom_errorbar]
## aes内のyminはエラーバーの下限ymaxは上限に対応
## yminは「平均値-標準誤差」、ymaxは「平均値+標準誤差」に対応
## widthは横幅 (1が棒グラフの横幅いっぱい)
gp = gp + geom_errorbar(aes(ymin = mean-se, ymax = mean+se), width=0.3)

gp = gp + scale_x_discrete(labels=c("T0B0","T1B0","T0B1","T1B1"))
gp = gp + scale_y_continuous(limits=c(0,6),breaks=0:6)
gp = gp + scale_fill_discrete(
  name = "CONDITION",
  labels=c("(T0B0) Top:none, Bot:pinched",
            "(T1B0) Top:self-pinched, Bot:pinched",
            "(T0B1) Top:none, Bot:pulled",
            "(T1B1) Top:self-pinched, Bot:pulled"))

gp = gp + facet_grid(. ~ stmt)

gp = gp + labs(title="Graph2B_QuestionnaireBar",x="CONDITION",y="RATE")
```

```
gp = gp + theme(  
  aspect.ratio=3/2,  
  text = element_text(face = "bold", size = 30),  
  axis.text.x = element_text(size = 25),  
  axis.text.y = element_text(size = 25),  
  #凡例の位置の調整、背景を加える  
  legend.position=c(1,1.08),  
  legend.justification=c(1,0),  
  legend.background =  
    element_rect(fill = "white", colour = "black"),  
  legend.text = element_text(size = 30))
```

gp

```

#-----
#-----
# [Graph3_EarPositionDrift]
# 錯覚前後の耳（上端・下端）の主観位置の変化
#-----
#-----

# まずはdat_Dの構造を復習
## drift：錯覚前後の位置の変化
## epos：耳が上端（TOP）か、下端（BOT）か
## etop：耳の上を被験者がつままない（T0）or つまむ（T1）
## ebot：耳の下を実験者がつまむ（B0）or 引っ張る（B1）
## order：（0）T0→T1、（1）T1→T0
str(dat_D)
# 'data.frame':      128 obs. of  6 variables:
#  $ sbj   : int   1 2 3 4 5 6 7 8 9 10 ...
#  $ drift: num  4.25 13.15 -0.25 3.6 -2.1 ...
#  $ epos  : chr  "TOP" "TOP" "TOP" "TOP" ...
#  $ etop  : chr  "T0" "T0" "T0" "T0" ...
#  $ ebot  : chr  "B0" "B0" "B0" "B0" ...
#  $ order: int   0 1 0 0 1 0 1 1 0 1 ...

#-- 前準備 --
# fillにマップされるeposの並び順を左からTOP->BOTにする
dat_D$epos = factor(dat_D$epos, levels=c("TOP", "BOT"))
#-----

# [ggplot]
gp = ggplot(dat_D, aes(x=interaction(etop,ebot),
                        y=drift, fill=epos))

# [geom_hline] y=0とy=50に横線を引く。solidは実線、dottedは点線
gp = gp + geom_hline(yintercept = 0, linetype="solid", linewidth=1)
gp = gp + geom_hline(yintercept = 50, linetype="dotted", linewidth=1)

# [geom_boxplot]ボックスプロットによる描画
gp = gp + geom_boxplot()
gp = gp + scale_x_discrete(
  limits=c("T0.B0", "T1.B0", "T0.B1", "T1.B1"),
  labels=c("T0B0", "T1B0", "T0B1", "T1B1"))

gp = gp + scale_x_discrete(
  limits=c("T0.B0", "T1.B0", "T0.B1", "T1.B1"),
  labels=c("T0B0", "T1B0", "T0B1", "T1B1"))

gp = gp + scale_y_continuous(limits=c(80, -5), breaks=seq(80, -10, by=-10), trans="reverse")

# 今回はbrewerパレットを使います（scale_fill_discreteと色の割り当てが異なる）。
# RColorBrewer::display.brewer.all() #色のパレットを見て、pairedを選ぶ。
gp = gp + scale_fill_brewer(
  name = "EAR'S POSITION", labels=c("TOP", "BOTTOM"), palette="Paired")
# gp = gp + scale_fill_discrete(
#   name = "EAR'S POSITION", labels=c("TOP", "BOTTOM"))

gp = gp + labs(title = "Graph3_EarPositionDrift",
               x = "CONDITION", y = "SUBJECTIVE EAR-LOCATION'S DRIFT(cm)")

gp = gp + theme(
  aspect.ratio=3/2,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 25),
  axis.text.y = element_text(size = 25),
  #凡例の位置の調整、背景を加える
  legend.position=c(0.05, 0.05),
  legend.justification=c(0, 0),
  legend.background =
    element_rect(fill = "white", colour = "black"),
  legend.text = element_text(size = 30))

gp

```



```

#-----
#-----
# [Graph4_EarSizeDeformation]
# 錯覚前後の耳のサイズの変化
#-----
#-----

# 最初に確認 (eposはGraph3でFactor化されていることに注意)
str(dat_D)
# 'data.frame':      128 obs. of  6 variables:
# $ sbj  : int   1 2 3 4 5 6 7 8 9 10 ...
# $ drift: num  4.25 13.15 -0.25 3.6 -2.1 ...
# $ epos : Factor w/ 2 levels "TOP","BOT": 1 1 1 1 1 1 1 1 1 1 ...
# $ etop : chr  "T0" "T0" "T0" "T0" ...
# $ ebot : chr  "B0" "B0" "B0" "B0" ...
# $ order: int   0 1 0 0 1 0 1 1 0 1 ...

# 耳のサイズの(下方向)変化(dsize)はebotからetopを引いたもの
# dsize = ebot - etopとして、新たな属性dsizeを追加しdat_D2とする

# 特定の参加者(id)の、TnBn条件におけるサイズ変化を計算する
getSizeDeformation = function(id,Tn,Bn){
  drift_eartop = dat_D[dat_D$subj==id & dat_D$etop==Tn
    & dat_D$ebot==Bn & dat_D$epos=="TOP",c("drift")]
  drift_earbot = dat_D[dat_D$subj==id & dat_D$etop==Tn
    & dat_D$ebot==Bn & dat_D$epos=="BOT",c("drift")]

  dsize = drift_earbot - drift_eartop
  dsize
}

# [使い方]
## 参加者1のT1B1におけるサイズ変化は48.5cm
getSizeDeformation(1,"T1","B1") #[1] 48.5
## 参加者13のT1B0におけるサイズ変化は3.35cm
getSizeDeformation(13,"T1","B0") #[1] 3.35

#全ての被験者、条件を一気に計算し、データフレームに流し込む

## (1) まずは空のデータフレームを作成
dat_dsize = data.frame(sbj=as.numeric(),dsize=as.numeric(),
  etop=as.character(),ebot=as.character())
# > dat_dsize
# [1] sbj dsize etop ebot
# <0 rows> (or 0-length row.names)

## (2) 繰り返し文で一気に計算
## - rbind(追加元DF、追加するDF)による行の結合を行います
for(id in 1:16){
  for(Tn in c("T0","T1")){
    for(Bn in c("B0","B1")){

      dsize = getSizeDeformation(id,Tn,Bn)
      print(dsize)
      dat_dsize = rbind(dat_dsize,
        data.frame(sbj=id,dsize=dsize,etop=Tn,ebot=Bn))
    }
  }
}

# 準備完了!
dat_dsize

str(dat_dsize)
# 'data.frame':      64 obs. of  4 variables:

```

```

# $ sbj : int 1 1 1 1 2 2 2 2 3 3 ...
# $ dsize: num 32.1 33.6 12.7 48.5 3.4 ...
# $ etop : chr "T0" "T0" "T1" "T1" ...
# $ ebot : chr "B0" "B1" "B0" "B1" ...

# [ggplot]
gp = ggplot(dat_dsize,aes(x=interaction(etop,ebot),
                                y=dsize,fill=interaction(etop,ebot)))

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

gp = gp + scale_x_discrete(
  labels=c("T0B0","T1B0","T0B1","T1B1"))
gp = gp + scale_y_continuous(limits=c(0,100),breaks=seq(0,100,by=10))
gp = gp + scale_fill_discrete(
  name = "CONDITION",
  labels=c("(T0B0) top:none, bot:pinched",
            "(T1B0) top:self-pinched, bot:pinched",
            "(T0B1) top:none, bot:pulled",
            "(T1B1) top:self-pinched, bot:pulled"))

gp = gp + labs(title = "Graph4_EarSizeDeformation",
               x = "CONDITION", y = "DEFORMATION SIZE(cm)")

gp = gp + theme(
  aspect.ratio=3/2,
  text = element_text(face = "bold", size = 30),
  axis.text.x = element_text(size = 25),
  axis.text.y = element_text(size = 25),
  #凡例の位置の調整、背景を加える
  legend.position=c(0.02,0.98),
  legend.justification=c(0,1),
  legend.background =
    element_rect(fill = "white", colour = "black"),
  legend.text = element_text(size = 25))

gp

```

```

#-----
#-----
# 【確認課題】
#-----
#-----

#-----
# [Work6_EarSizeDeformationBar]

# Graph4_EarSizeDeformationを
# エラーバー付きの棒グラフにしてください。
# 締切は11月中とします。
# ファイル名は「2250xx_work6.R」としてください。
#-----

#-----
# 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_dsize
dat_dsize[1:3,]

# 平均値・標準誤差の組を4通りの条件で一気に吐き出すために関数を定義（上書き）
getMeanSE = function(Tn,Bn){
  ## 16人分の変形量をベクトルで取り出す
  dsizes = dat_[dat_$etop==Tn & dat_$ebot==Bn,c("dsize")]
  mean = mean(dsizes)
  SE = sd(dsizes) / sqrt(length(dsizes))

  c(mean,SE)
}

# 以降は[Graph2_QuestionnaireBar] を参考にすすめてください。

```

```

#-----
#-----
# (補講：パイプ演算子、統計量の要約)

# [%>%]
# [group_by(data_frame, 列属性1, 列属性2, ...)]
# [summarize(arg1 = mean(計測値属性), arg2 = sd(計測値属性), ...)]

# パイプ演算子を使って、
# 属性毎の統計値をデータフレームとして一気に出力する
#-----
#-----

# group_by関数を使うためにはdplyrライブラリが必要です。

#install.packages(dplyr)
library(dplyr)

#-----
# パイプ演算子 %>%
#-----

# アンケート実験の結果
dat_Q
#   sbj rate stmt etop ebot
# 1     1  1.0  Q1  T0  B0
# 2     2  1.5  Q1  T0  B0
# 3     3  0.0  Q1  T0  B0
# ...
# 198   6  0.0  Q4  T0  B0
# 199   7  0.0  Q4  T0  B0
# 200   8  1.5  Q4  T0  B0

# (復習) 新しい列属性 (sbj2) を追加する方法
mutate(dat_Q, sbj2 = sbj-1)
#   sbj rate stmt etop ebot sbj2
# 1     1  1.0  Q1  T0  B0     0
# 2     2  1.5  Q1  T0  B0     1
# 3     3  0.0  Q1  T0  B0     2
# ...

# 同じことを、異なる記法 (パイプ演算子) でも表現できます。

dat_Q %>%
  mutate(sbj2 = sbj-1)
#   sbj rate stmt etop ebot sbj2
# 1     1  1.0  Q1  T0  B0     0
# 2     2  1.5  Q1  T0  B0     1
# 3     3  0.0  Q1  T0  B0     2
# ...

# (1)と(2)は同じ意味です。

# (1)
# dat %>%
#   func(arg1, arg2, ...)

# (2)
# func(dat, arg1, arg2, ...)

# 階層はいくらでも深くできます。
# (この場合、明らかに(1)の方が視認性が高くなります)

# (1)
# dat %>%
#   func1(arg1, arg2, ...) %>%
#   func2(arg1, arg2, ...)

# (2)
# fun2(func1(dat, arg1, arg2, ...), arg1, arg2, ...)

```

```

#-----
# group_by, summarize
#-----

dat_Q
#   sbj rate stmt etop ebot
# 1   1  1.0  Q1  T0  B0
# 2   2  1.5  Q1  T0  B0
# 3   3  0.0  Q1  T0  B0
# ...

# dat_Qの列属性の構造は以下の通り
# sbj : 1,2,3,...,16 (16人)
# stmt: Q1,Q2,Q3,Q4 (4種類: 質問の種類)
# etop: T0,T1 (2種類: 耳上端条件)
# ebot: B0,B1 (2種類: 耳下端条件)

# stmtごとの平均値の出力 (4条件を無視)

## group_byで、グループ属性 (stmt) を指定し、
## その結果をsummarizeにパイプし、
## その中で、計算したい統計値 (mean:平均値) を指定する

dat_Q %>%
  group_by(stmt) %>%
  summarize(
    mean = mean(rate)
  )

# A tibble: 4 × 2
#   stmt mean
#   <chr> <dbl>
# 1 Q1    2.71
# 2 Q2    2.20
# 3 Q3    0.688
# 4 Q4    0.258

# stmt x etop x ebotごとの平均値の出力 (全16グループ)

dat_Q %>%
  group_by(stmt,etop,ebot) %>%
  summarize(
    mean = mean(rate)
  )

# # A tibble: 16 × 4
# # Groups:   stmt, etop [8]
#   stmt etop ebot mean
#   <chr> <chr> <chr> <dbl>
# 1 Q1    T0    B0    1.28
# 2 Q1    T0    B1    3.28
# 3 Q1    T1    B0    1.69
# 4 Q1    T1    B1    4.59
# 5 Q2    T0    B0    1.25
# 6 Q2    T0    B1    2.47
# 7 Q2    T1    B0    1.53
# 8 Q2    T1    B1    3.53
# 9 Q3    T0    B0    0.344
# 10 Q3   T0    B1    1.09
# 11 Q3   T1    B0    0.469
# 12 Q3   T1    B1    0.844
# 13 Q4   T0    B0    0.25
# 14 Q4   T0    B1    0.281
# 15 Q4   T1    B0    0.281
# 16 Q4   T1    B1    0.219

## 標準誤差も合わせて求める

### sd(rate)はrateの標準偏差、n()はサンプル数

dat_Q %>%
  group_by(stmt,etop,ebot) %>%

```

```

summarize(
  mean = mean(rate),
  sd = sd(rate),
  n = n(),
  se = sd / sqrt(n))

# stmt etop ebot mean sd n se
# <chr> <chr> <chr> <dbl> <dbl> <int> <dbl>
# 1 Q1 T0 B0 1.28 1.37 16 0.341
# 2 Q1 T0 B1 3.28 1.53 16 0.382
# 3 Q1 T1 B0 1.69 1.73 16 0.433
# 4 Q1 T1 B1 4.59 1.27 16 0.317
# 5 Q2 T0 B0 1.25 1.40 16 0.351
# 6 Q2 T0 B1 2.47 1.61 16 0.402
# 7 Q2 T1 B0 1.53 1.51 16 0.378
# 8 Q2 T1 B1 3.53 1.61 16 0.402
# 9 Q3 T0 B0 0.344 0.625 16 0.156
# 10 Q3 T0 B1 1.09 1.14 16 0.286
# 11 Q3 T1 B0 0.469 0.865 16 0.216
# 12 Q3 T1 B1 0.844 1.25 16 0.312
# 13 Q4 T0 B0 0.25 0.707 16 0.177
# 14 Q4 T0 B1 0.281 0.605 16 0.151
# 15 Q4 T1 B0 0.281 0.706 16 0.177
# 16 Q4 T1 B1 0.219 0.547 16 0.137

# 出力結果をデータフレームに変換するには
# 以下のようにすればよい。

tmp = dat_Q %>%
  group_by(stmt,etop,ebot) %>%
  summarize(
    mean = mean(rate),
    sd = sd(rate),
    n = n(),
    se = sd / sqrt(n))

dat_stat = as.data.frame(tmp)
# > dat_stat
# stmt etop ebot mean sd n se
# 1 Q1 T0 B0 1.28125 1.3658788 16 0.3414697
# 2 Q1 T0 B1 3.28125 1.5271842 16 0.3817961
# 3 Q1 T1 B0 1.68750 1.7308476 16 0.4327119
# ...

# 同様にdat_D関連の統計値を要約する

dat_D
# sbj drift epos etop ebot
# 1 1 4.250 TOP T0 B0
# 2 2 13.150 TOP T0 B0
# 3 3 -0.250 TOP T0 B0
# ...
# 126 14 41.100 BOT T1 B1
# 127 15 42.600 BOT T1 B1
# 128 16 39.000 BOT T1 B1

## drift量の統計量

dat_D %>%
  group_by(epos,etop,ebot) %>%
  summarize(
    mean = mean(drift),
    sd = sd(drift),
    n = n(),
    se = sd / sqrt(n)
  ) %>%
  as.data.frame()

# epos etop ebot mean sd n se
# 1 BOT T0 B0 8.240937 10.012798 16 2.5031994
# 2 BOT T0 B1 32.172187 19.936722 16 4.9841805
# 3 BOT T1 B0 14.928125 13.671990 16 3.4179976
# 4 BOT T1 B1 36.621875 25.877306 16 6.4693266
# 5 TOP T0 B0 1.678125 3.902904 16 0.9757259

```

# 6	TOP	T0	B1	4.028125	4.982870	16	1.2457174
# 7	TOP	T1	B0	4.421875	6.185762	16	1.5464406
# 8	TOP	T1	B1	3.265625	3.781478	16	0.9453694