```
# 以下から、奇数偶数の好み実験結果の一部のデータをインポートしてください。
# http://lab.kenrikodaka.com/_download/class/2022_AppliedMedia/oddeven.csv


#Environmentパネルの「Import Dataset」から
#oddeven.csvのデータを、expdatという変数に読み込みます。

#Name：expdat（入力してください）
#Encoding:Automatic
#Heading:Yes
#Row names:Automatic
#Separator:Comma
#Decimal:Period
#Quote：Double
#Coomen:None
#na.stfings:NA
#Strings as factors:Unchecked（チェックは要りません）


#誕生日と奇数偶数の好みに関するアンケートの架空のデータ（1000人分）です。
#month（誕生月）・name（誕生日）・preference（奇数が好き：1, 偶数が好き：0）
#sex（男性：1, 女性：0）・$domhand（利き手が左：l , 利き手が右：0）
#age（年齢）

#最初の6行をちょっと出し
head(expdat)

#  month day preference sex domhand age
#1     8  29          1   1       0  19
#2     1  29          0   0       1  18
#3     3  19          0   1       0  18
#4    10   6          0   0       0  18
#5    10  10          1   0       0  18
#6     1  13          0   1       0  19

# 翻訳すると以下の様になります。
# 8月1日、奇数好き、男性、右利き、19歳
# 1月29日、偶数好き、女性、左利き、18歳
# 3月19日、偶数好き、男性、右利き、18歳
# 10月6日、偶数好き、女性、右利き、18歳
# 10月10日、奇数好き、女性、右利き、18歳
# 1月13日、偶数好き、男性、右利き、19歳




#----------------------------------------
# [準備1] データフレームから集計表をつくる
#----------------------------------------

# table関数を使うと、指定した変数に関する集計表（対応表）を
# 簡単に作ることができます。

# 「誕生月」と「好み」の関係の集計表（どちらも同じ結果となります。）
table(expdat[c("month","preference")])
table(expdat[c(1,3)])
#     preference
```

```
#month  0  1
#1  52 38
#2  42 18
# ...
#12 52 30

# 「day」と「好み」の集計表
table(expdat[c("day","preference")])
#    preference
#day   0  1
# 1  19 18
# 2  21  7
# ...
# 31 15 10

# 「性別」と「好み」の集計表
table(expdat[c("sex","preference")])
# preference
# sex   0   1
# 0 370 203
# 1 230 197


#----------------------------------------
# [準備２] ファクタ
#----------------------------------------

# 量的変数（0 1）を、量を持たないカテゴリ変数に変換します。
# 以下の例では、性別の01を"FEMALE" "MALE"に、
# 利き手の01を"RIGHT" "LEFT"に、
# 奇数偶数の好みの01を"EVEN" "ODD"の文字列カテゴリに割り当てます。
# このようなデータのタイプをファクタと呼びます。

expdat$sex = factor(expdat$sex,levels=0:1,labels=c("FEMALE","MALE"))
expdat$domhand =
factor(expdat$domhand,levels=0:1,labels=c("RIGHT","LEFT"))
expdat$preference =
factor(expdat$preference,levels=0:1,labels=c("EVEN","ODD"))


# このように変わります。
expdat$preference
#[1] ODD  EVEN EVEN EVEN ODD  EVEN EVEN EVEN EVEN EVEN EVEN EVEN EVEN EVEN
EVEN
# . . .
#[976] EVEN EVEN EVEN ODD  ODD  ODD  EVEN EVEN EVEN EVEN ODD  ODD  EVEN
EVEN ODD
#[991] EVEN ODD  EVEN ODD  ODD  EVEN EVEN EVEN ODD  ODD

# クラスはfactorになります。
class(expdat$preference)
#[1] "factor"

# factorの型は常にintegerです。
typeof(expdat$preference)
#[1] "integer"
```

```
# これは、factorが、各カテゴリを特定の整数インデックス（1,2,..）と対応させているためです。
# これを確認するには、ファクタを解除する「as.numeric」または「unclass」を使います。
as.numeric(expdat$preference)
#[1] 2 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2 2 1 1 1 1 1 1 2 1 2
2 1 1 1 1 2
#[42] 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 1 1 2 2 2 1 1 2 1 1 1 1 1 2 1 1 1 2 1
1 1 2 1 1 2 1
unclass(expdat$preference)
#[1] 2 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2 2 1 1 1 1 1 1 2 1 2
2 1 1 1 1 2
#[42] 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 1 1 2 2 2 1 1 2 1 1 1 1 1 2 1 1 1 2 1
1 1 2 1 1 2 1

# このようにファクタ化されたpreferenceは
# 1がEVEN、2がODDに対応していることに注意してください。

# もとに戻したければ、例えば以下のようにすれば大丈夫です。
expdat$preference = unclass(expdat$preference)-1
#[1] 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1
1 0 0 0 0 1
#[42] 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0
0 0 1 0 0 1 0

# Factorにすると、各カテゴリが何を意味するか一目瞭然となります。
table(expdat[c("sex","preference")])
# sex      EVEN ODD
# FEMALE  370 203
# MALE    230 197


# ついでに月もファクタの変えてみましょう。
# あらかじめ用意されている、便利な月の文字列ベクトルを使いましょう。
# month.abbはmonthのabbreviation（略語）の意味です。
month.abb
#[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov"
"Dec"

expdat$month = factor(expdat$month,levels=1:12,labels=month.abb)
expdat$month
#[1] Aug Jan Mar Oct Oct Jan Jul Aug Nov Apr Dec Jul Oct May Apr Mar Jan
Jul
#[19] Apr Apr Jan Jan Jul Apr Dec Nov Oct Jun May Dec Dec Aug Mar Dec Sep
Nov

table(expdat[c("month","preference")])
#       preference
#month  0  1
#Jan 52 38
#Feb 42 18
#...
#Dec 52 30

# もとの数字として扱いたい時は（as.xxxは「xxxとして」と読んでください）
as.numeric(expdat$month)
#[1]  8  1  3 10 10  1  7  8 11  4 12  7 10  5  4  3  1  7  4  4  1  1  7
4 12
```

```
#[26] 11 10  6  5 12 12  8  3 12  9 11 10  1 12  3  6  1  6  8 10  9 12  3
1  8

# unclassも同じ効果があります。
unclass(expdat$month)
#[1]  8  1  3 10 10  1  7  8 11  4 12  7 10  5  4  3  1  7  4  4  1  1  7
4 12
#[26] 11 10  6  5 12 12  8  3 12  9 11 10  1 12  3  6  1  6  8 10  9 12  3
1  8




########################################
########################################
####### [KAI] カイ二乗分布 ############
########################################
########################################


#-----------------------------------------
# [KAI.1] 適合度検定：サイコロの例題
#-----------------------------------------


# サイコロを60回振った時の各目が以下のようになる。
obs1 = c(5,8,10,20,7,10) #観測事象１
obs2 = c(15,8,14,6,8,9)  #観測事象２

# それぞれの目の出現確率が1/6のとき、期待値は？
expected = c(10,10,10,10,10,10)

# 観測値と期待値が対応するデータフレームを作成します。
dat = data.frame(OBS1 = obs1, OBS2 = obs2, EX = expected)


# 観測事象１と２はそれぞれ、
# 偏りのないサイコロからの出力と言えるか？
# 期待値からのズレを基にして検証する。


# （OBS1の場合）
# 1の出現数の期待値からズレを以下のように標準化します。
# ズレ：5（1の出現数）−10（期待値） = −5
# 二乗して正にする (5−10)∗(5−10) = 25
# 期待値で割ることで正規化：{(5−10)∗(5−10)}/10 = 2.5
# これが1の期待値とのズレを表す統計量

# 同じ操作を1から6まで全て行い加算したものをkai2.OBS1とする
kai2.OBS1 = sum((dat$OBS1 - dat$EX)^2 / dat$EX)
#[1] 13.8

# （OBS2の場合）
# OBS2に対しても同様に計算する
kai2.OBS2 = sum((dat$OBS2 - dat$EX)^2 / dat$EX)
#[1] 6.6
```

```
# このようにして計算される「期待値からのズレの二乗和」を
# 「カイ二乗値（χ^2）」と呼びます。


# 「カイ二乗値」は、Rでは
# chisq.test（観測値ベクトル, p=期待値の確率分布）で求められます。
# 2つ目の引数：期待値の確率分布の総和は1となっている必要があります。
# 今回は, 期待値の確率分布はc(1/6,1/6,1/6,1/6,1/6,1/6)となるので、
result.OBS1 = chisq.test(dat$OBS1,p = rep(1/6,times=6))
result.OBS1$statistic #返り値の名前属性$statisticがχ^2に対応
#X-squared
#     13.8

#F2についても同様に
result.OBS2 = chisq.test(dat$OBS2,p = rep(1/6,times=6))
result.OBS2$statistic
#X-squared
#       6.6


# この統計量（kai2）が大きな値をとるほど、ランダム事象とのズレが大きく、
# 偶然では起こりにくい事象であることがわかります。
# それでは、実際に、サイコロの出力が完全にランダムなときに、
# kai2がどのような分布をとるかを調べましょう。

# ここでは、多数のランダム試行のシミュレーション（モンテカルロ法）により、
# おおよその「当たり」をつけてみます。

# 60回サイコロを振った時のkai.2を算出する関数（getDiceKai2）を作ります。

getDiceKai2 = function(){

  # 60回サイコロを振りベクトルに展開します。
  dice.60 = sample(1:6,60,replace=TRUE);

  # 各出目の出現数のベクトルを作ります。
  dice.6 = vector("integer",6)
  for(i in 1:6){
    dice.6[i] = length(which(dice.60==i))
  }
  # 各出目の期待値は
  dice.ex = c(10,10,10,10,10,10)
  # 期待値からのズレの統計量を各出目毎に加算
  sum((dice.6 - dice.ex)^2  / dice.ex);
}

set.seed(17) #乱数を固定します（同じ出力とするため）。
# getDiceKai2()を100000回実行し、
# その統計量を集めます。
simulation = replicate(100000,getDiceKai2())

#最小値・下位25%・中央値・平均・上位25%・最大値
summary(simulation)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0.000   2.600   4.400   5.017   6.600  31.200

# ヒストグラムの計算
```

```
# X軸は、0から最大値より大きな値（35）まで,
# 0.2刻みでベクトルを計算
h = hist(simulation,breaks = seq(0,35,by=0.2))

h$breaks #横軸の刻み 0.0 0.2, ... 34.8 35
h$count #横軸に対応する頻度数（Frequency）222  573  815 ...
h$density #横軸に対応する確率密度 0.01110 0.02865 0.04075 ...

# hist関数はデフォルトでは頻度を縦軸に出力
# 縦軸を確率密度にするには、引数でfreq=FALSEを指定
h = hist(simulation,breaks = seq(0,35,by=0.2),freq=FALSE)

# getDiceKai2が13.8(kai2.obs1)より大きくなる確率は？
sum(h$density[h$breaks>=13.8]*0.2,na.rm = T)#NAを無視
#0.01711
# → kai2.F1程度の偏りは1.7%程度でしか生じない
# → 有意な偏りが存在する。

# getDiceKai2が18（kai2.obs2）より大きくなる確率は？
sum(h$density[h$breaks>=6.6]*0.2,na.rm = T) #NAを無視
#0.24703
# → kai2.F2以上の偏りはおよそ24.7%の確率で生じる。
# → 有意な偏りが存在しない。



# このような確率を「p値」と呼ぶ。
# 「p値」もまた、Rの「chisq.test」で正確な値を自動的に算出できる。

result.F1 = chisq.test(dat$F1,p = rep(1/6,times=6))
result.F1$p.value
#[1] 0.01693102
# 0.01711でうまく近似できていたことがわかる

result.F2 = chisq.test(dat$F2,p = rep(1/6,times=6))
result.F2$p.value
#[1] 0.2521282
#0.24703でうまく近似できていたことがわかる


# ちょうど5%の境界を求めるには、例えば以下のようにする

kai2 = max(simulation) #最大のシミュレーション値から始める。

repeat{
  # 統計量がkai2より大きくなる確率は？
  ratio = sum(h$density[h$breaks>kai2]*0.2,na.rm = T)
  # ratioが5%を初めて上回るところで
  # kai2をコンソールにプリントして、繰り返し計算を終了
  if(ratio>=0.05){
    print(kai2)
    break;
  }else{
    # それ以外は、kai2を0.01低くして再実行
    kai2 = kai2 - 0.01
  }
}
#11
```

```
# 以上より、観測するカテゴリが「6」のとき、
# カイ二乗の統計量が11付近より大きい場合,
# 偶然以上の偏りを持っていると結論できる。



#----------------------------------------
# [KAI.2] 適合度検定（男女別、奇数と偶数の好み）
#----------------------------------------

# 女性の偶数好き:370、奇数好き:203
preference = c(370,203)
# 好みがランダムの時の期待値は共に総数の半分
expected = c(0.5*sum(preference),0.5*sum(preference))

# 定義通り計算すると
kai2 = sum((preference - expected)^2 / (0.5 * sum(preference)))
#48.6719

# Rの関数を使うと
result = chisq.test(c(370,203),p = c(0.5,0.5))
result$statistic
#X-squared
# 48.6719


# モンテカルロ法でχ^2の分布をみます。

getPrefKai2 = function(){
  odd.sample = sum(sample(c(0,1),573,replace=T))
  even.sample = 573 - odd_sample
  result = chisq.test(c(even.sample,odd.sample),p = c(0.5,0.5))
  kai2 = result$statistic
  unname(kai2) #名前属性を消して返す
}

set.seed(17) #乱数を固定します。
simulation = replicate(100000,getPrefKai2())
summary(simulation)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0.00000 0.02788 0.11388 0.25295 0.33562 5.43327

# χ^2の分布のピークは0付近にあることに注意。
# カテゴリ数（自由度）によってカイ二乗分布のピークは変わります。
h = hist(simulation,seq(0,6,by=0.1),probability = T)

# summary関数より,100000回ランダムサンプリングして、
# 最大のカイ二乗値が5.4付近にあります。
# つまり、χ^2>48は、100000回に1回も起きないほどの異常な偏り

# 実際、p値は1/(10^12)付近（1000億回に一回程度）
result = chisq.test(c(370,203),p = c(0.5,0.5))
result$p.value
#[1] 3.025703e-12

# 以上より、女性は有意に偶数を好む傾向にある。
```

```r
# 男性の偏りも以下のように検証できる。
# 男性の偶数好き:230、奇数好き:197

result = chisq.test(c(230,197),p = c(0.5,0.5))
result$statistic #統計量は2.55
# X-squared
# 2.550351
result$p.value #p値>0.11
# [1] 0.1102697

# すなわち、230/197程度の偏りは、100回に11回程度は生じる。
# 男性が有意に偶数を好む傾向にあるとは言えない。



#----------------------------------------
# [KAI.3] 適合度検定（統計量と自由度）
#----------------------------------------

# カテゴリ数（自由度+1）が異なると
# χ^2分布がどのように変わるかを確認します。


# hのn番目の要素がカテゴリ数「n」のx^2分布を持つようにします。
# カテゴリ数1は意味を持たないため、h[[1]]は未定義となります。
h = list()

# カテゴリ数をnとする
for(n in 2:10){

  obs = vector("integer",n)   # 観測ベクトル
  exp = rep(1/n,times=n)  # 期待値ベクトル
  result = vector("double",10000) #kai^2の全サンプル（10000）

  # 各カテゴリ数nで10000回、kai^2値のサンプルを集めます。
  for(i in 1:10000){

    # 1からnのいずれかのカテゴリを500回ランダムに生成します。
    t = sample(1:n,500,replace=T)
    # 1からnの各事象の生起回数を（観測値として）obsベクトルにまとめます。
    for(ii in 1:n){
      obs[ii] = sum(t==ii)
    }
    # i番目のサンプルに、現在のobsでの統計量を登録します。
    # unnameは単に名前属性を消すためです。
    result[i] = unname(chisq.test(obs,p = exp)$statistic)
  }
  # リストhのn番目の要素にkai^2のベクトルを代入します。
  h[[n]] = result
}

# nを2から10に変えてヒストグラムを観察してください。
n = 10
hist(h[[n]],seq(0,50,by=0.1),probability=T)

# カテゴリ数が増えるに従って、
# ピークがn付近に寄ることがわかるはずです。
```

```
# 一般に統計量は、自由度によって分布の形状が変化します。
# 逆に言えば、自由度が同じであれば、
# サイコロであれ、奇数偶数の好みであれ、
# 同一の尺度で期待値からのズレを計測することが可能となります。


#-----------------------------------------
# [KAI.4] 適合度検定（誕生日、奇数と偶数の好み）
#-----------------------------------------

# 応用です。

# 2月生まれ

expdat$sex = factor(expdat$sex,levels=0:1,labels=c("FEMALE","MALE"))
expdat$domhand =
factor(expdat$domhand,levels=0:1,labels=c("RIGHT","LEFT"))
expdat$preference =
factor(expdat$preference,levels=0:1,labels=c("EVEN","ODD"))
expdat$month = factor(expdat$month,levels=1:12,labels=month.abb)

# 月ごとの集計は以下でベクトル化できます。
obs = table(expdat$month);
#Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
#90  60 100  90  79  86  98  87  69  82  77  82

# このサンプルは実際の誕生日の分布と比べて偏りがあると言えるでしょうか。
# 期待値ベクトルは、各月の日数を365日で割ることでつくることができます。
exp = c(31,28,31,30,31,30,31,31,30,31,30,31) / 365

chisq.test(obs,p=exp)
# Chi-squared test for given probabilities
#data:  obs
#X-squared = 12.658, df = 11, p-value = 0.3163

# カイ二乗値は12.658です。
# カテゴリ数が12のため、12.66はそれほど大きな値ではありません。
# 実際、p値は0.3163のため、10回に3回程度は、
# 観測値より大きな偏りがあることになります。
# 以上より、観測値に有意な偏りは存在するとは言えません。

# 次に誕生日の日にちが偶数の人の数は
sum(expdat$day %% 2 == 0)
#[1] 489

# この480人の奇偶の好みは
expdat$preference[expdat$day %% 2 == 0]
#[1] EVEN ODD  EVEN EVEN EVEN EVEN EVEN ODD  ODD  ODD  EVEN EVEN EVEN EVEN
EVEN
#[16] EVEN EVEN EVEN EVEN EVEN EVEN EVEN ODD  EVEN EVEN EVEN EVEN ODD
EVEN EVEN
#...

# このうち、偶数・奇数が好きな人の数は,,
even = sum(expdat$preference[expdat$day %% 2 == 0] == "EVEN")
odd = sum(expdat$preference[expdat$day %% 2 == 0] == "ODD")
c(even, odd)
```

```
#[1] 306 183

# 適合度検定
chisq.test(c(even,odd),p=c(0.5,0.5))
#X-squared = 30.939, df = 1, p-value = 2.663e-08

# 誕生日が偶数日の集団は有意に偶数好きが多い。


# 誕生日が奇数の場合は？
even = sum(expdat$preference[expdat$day %% 2 == 1] == "EVEN")
odd = sum(expdat$preference[expdat$day %% 2 == 1] == "ODD")
c(even, odd)
#[1] 294 217

# 適合度検定
chisq.test(c(even,odd),p=c(0.5,0.5))
#X-squared = 11.603, df = 1, p-value = 0.0006585

# 誕生日が奇数日の集団も有意に偶数好きが多い。




# 誕生月も誕生日も奇数の場合は？
odd.double = expdat$day %% 2 == 1 & as.numeric(expdat$month) %% 2 == 1
even = sum(expdat$preference[odd.double] == "EVEN")
odd = sum(expdat$preference[odd.double] == "ODD")
c(even, odd)
#[1] 146 121

# 適合度検定
chisq.test(c(even,odd),p=c(0.5,0.5))
#X-squared = 2.3408, df = 1, p-value = 0.126

# 誕生月も誕生日も奇数の集団の好みに有意な差は存在しない。
```