統計学第 7 講/第 8 講

後藤　晶
akiragoto@meiji.ac.jp

目次

目次

前回の復習

関数とパッケージ

R においてよく使う計算式は**関数**として用意されています．必要な関数は**パッケージ**をインストールすることで，適宜追加することができます．

- パッケージ：機能を拡張するもの．
  - 研究者など，たくさんの開発者が自身の研究上・仕事上のニーズに応じて拡張パッケージを用意している．
  - これを使えば様々な分析や操作が便利になる．
  - 必要に応じて，様々なパッケージを追加していくイメージ

以下の段取りを踏むことで利用可能となる．

1. パッケージをインストールする．
2. パッケージを読み込む．

1. については，一度行うだけで良い．
2 については必要に応じて適宜実施する．

以下にはその一例を示す．

```
install.packages(“dplyr”, dependencies = TRUE)
パッケージをインストールする．一度実行するだけで良い．
```

```
library(dplyr)
パッケージを読み込む，使うときには必ず入力する
```

他のデータを読み込む

今は皆さんに手入力でデータを打ち込んで貰いました．今度は，皆さんには “csv ファイル” からデータを読み込んでもらおうと思います．R の標準のデータ形式以外の他の形式のファイルを読み込むことを「インポート」と言います．

RStudio を使ってもらうと，次の手順でデータを読み込むことができます．

- “Import Dataset” をクリックする．
  - “From Text (readr)…” をクリックする．
  - 何かをインストールするように案内されたら，素直にインストールする．
- “Browse” をクリックする
  - 読み込みたいデータを選んで “Open” をクリックする．
  - データに併せて，クリックしていく．
    * 今回の場合は “First Row as Names” にチェックを入れる．これは 1 行目が各行のデータ名を示しているためである．
- “Import” をクリックしてデータを読み込む．
- 完了

下のコンソールには 3 つのコードが書かれます．1 番目のコードは “readr” というパッケージを使うように，という指示をしています．2 番目のコードは “データを読み込んで，こんな名前にしておいて下さい” を示しており，3 番目のコードは “読み込んだデータを表示して下さい” を示している．

なお，このコード（特に上の 2 つ）は “>” を取り除いて上の “.R” ファイルに保存しておくと，次回以降便利です．

```
library(readr)
# パッケージreadrを使う
dataset <- read_csv("~/hogehoge/dataset.csv")
# datasetを読み込む
```

- R のコード内で “#” と書くとコメントアウト（コードとして扱わず，メモとして使える）

なお，この “hogehoge” は読み込んだデータを保存した場所を示しており，人によって異なるので注意してください．

データのダウンロード

- 「データの説明」ページにある「こちらからダウンロードしてください」というところから，csv ファイルをダウンロードしてください．

注意：この授業で取り扱うデータについて

このデータはゴトウが実施した 1926 人分のデータのうち，ランダムに選んだ 963 人分のデータです．まだ，データの中身は「データの概要」に記載してあるので，そちらを参考にしてください．

読み込んだデータの記述統計量を算出します．ここでは人々の主観的幸福度について記述統計量を算出します．

主観的幸福度とは：

主観的幸福度とは人が感じている幸福度を示したものです．ここでは「現在，あなたはどの程度幸せですか？「とても幸せ」を 10 点，「とても不幸せ」を 0 点とすると，何点くらいになると思いますか？」として尋ねたものです．

それでは，記述統計量を出してみましょう．特に，複数列あるデータの場合は $ を使って，「データセットの中のこのデータ列について平均値を出して下さい」というように指定してあげます．

- データは前回の授業資料からダウンロードできます．

平均・分散・標準偏差・度数など．

```
library(readr)
exdataset <- read_csv("../data/exdataset.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   F_SEX = col_character(),
##   F_GEN_2 = col_character(),
##   F_GEN = col_character(),
##   F_FGR = col_character(),
##   F_INK = col_character(),
```

```
##   F_INS = col_character(),
##   F_TAN = col_character(),
##   ARE = col_character(),
##   MAR = col_character(),
##   CHI = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
library(ggplot2)
```

```
# データを読み込めたら，以下をコピペしてください．
# いろいろなものの順番を理解しやすいように並べ替えます．

## Reordering exdataset$ARE
exdataset$ARE <- factor(exdataset$ARE, levels=c("Kanto", "Hokkaido", "Tohoku", "Chubu", "Kinki", "Ch

## Reordering exdataset$MAR
exdataset$MAR <- factor(exdataset$MAR, levels=c("NotMarried", "Married"))

## Reordering exdataset$CHI
exdataset$CHI <- factor(exdataset$CHI, levels=c("NoChild", "Child"))
```

記述統計量をコードで算出する

平均値を算出してみる．

主観的幸福度 (SUB_HAP) の平均値

```
mean(exdataset$SUB_HAP)
```

```
## [1] 6.002077
```

分散を算出してみる．

主観的幸福度 (SUB_HAP) の分散

```
var(exdataset$SUB_HAP)
```

```
## [1] 5.503114
```

標準偏差を算出してみる.

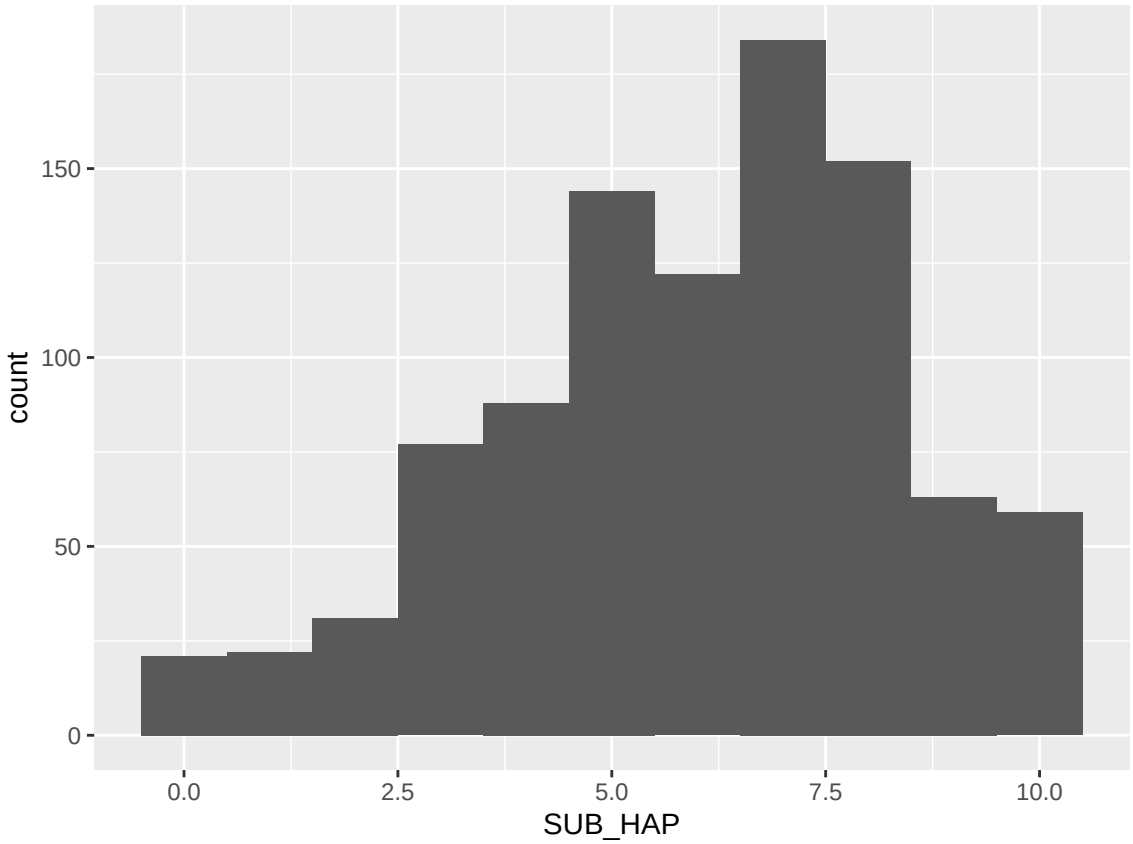
主観的幸福度 (SUB_HAP) の標準偏差

```
sd(exdataset$SUB_HAP)
```

```
## [1] 2.345872
```

主観的幸福度 (SUB_HAP) のヒストグラム

```
exdataset %>% ggplot(aes(x = SUB_HAP)) + geom_histogram(binwidth = 1.0)
```



運命 (SPN_UNM) の頻度を数えてみる.

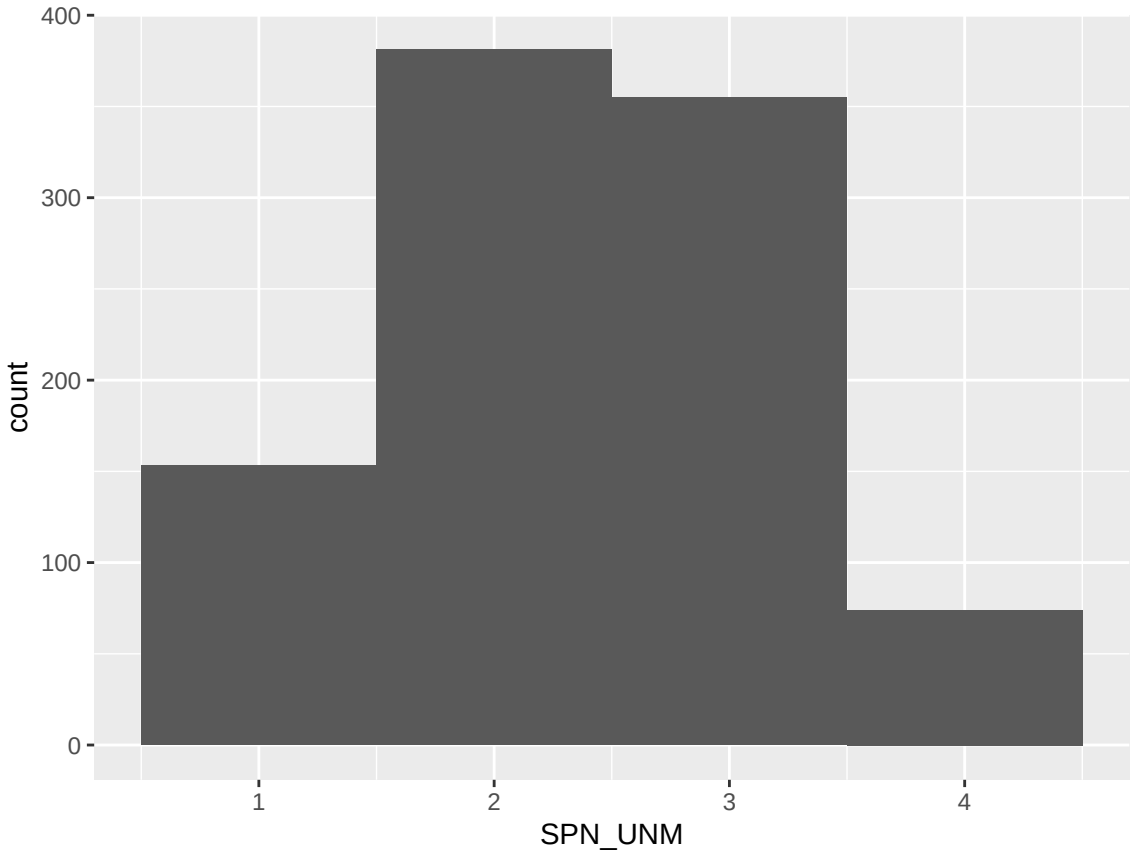
```
table(exdataset$SPN_UNM)
```

```
##
##   1   2   3   4
```

```
## 153 381 355  74
```

ついで運命 (SPN_UNM) のにヒストグラムも作ってみよう

```
exdataset%>% ggplot(aes(x = SPN_UNM)) +
  geom_histogram(binwidth = 1.0)
```



世代の頻度を数えてみる.

```
table(exdataset$F_GEN)
```

```
##
## 10dai 20dai 30dai 40dai 50dai 60dai 70dai
##     8   140   361   358    77    18     1
```

- クリックだけで図表を作ります.

```
install.packages("ggplotgui")
```

```
# 初回だけ必要，ggplotguiをインストールする
library(ggplotgui)
# パッケージggplotguiを使う
```

再現可能性 (Reproducibility) の重要性

再現可能性に関する様々な議論と定義（Kulkarni, 2017）

Goodman による定義（Goodman et.al, 2016）：

- **方法の再現可能性** (Methods reproducibility)：反復可能性にもっとも近い．研究方法とデータに関する十分な情報が提供され、同じ手順を反復できるようになっていることを意味する．
- 結果の再現可能性 (Results reproducibility)：「方法の再現可能性」と密接に関連している．「元の実験と可能な限り同じ手順で，独立した実験を実施し，同じ結果を得ること」を意味する．
- 推論の再現可能性 (Inferential reproducibility)：先の 2 つの再現可能性とは異なる。別の研究から同じ推論が導かれることもあれば，同じデータから別の結果が推測されることもある．このため，推論の再現可能性とは「独立した再現実験もしくは元の研究の再分析から，質的に類似した結果を導くこと」を意味する．

Stodden による定義（Stodden, 2014）：

- 実証的再現可能性 (Empirical reproducibility)：物理的に実験を繰り返して実証する必要なすべての情報が提供されていることを意味する．この定義は，グッドマン氏の「方法の再現可能性」の定義に近い．
- **計算／統計的再現可能性** (Computational and statistical reproducibility)：研究における計算結果や分析結果を再び行うために欠かせないリソースが提供されていることを意味する．

Baker による定義（Baker, 2016）：

- **分析的反復** (Analytic replication)：単に元データを再分析して結果を再現すること．
- 直接的反復 (Direct replication)：元の実験と同じ条件，材料，方法を利用しようとすること．
- 体系的反復 (Systematic replication)：異なる実験条件で結果を再現しようとすること．（例えば，異なる細胞株やマウス株で実験を行うことなど．）
- 概念的反復 (Conceptual replication)：ある概念の一般的な正当性を示そうとすること．異なる有機体を使用する場合も含まれる．

再現可能なデータ分析とレポート作成のメリット（高橋, 2018）

信頼性の向上

- データ解析：得たデータを分析結果やグラフに変換すること
- 同じデータからいつでもどこでも誰でも同じ結果を得られる必要がある
  - 皆さんの分析結果がゴトウが授業でやっている結果と一致しなかったら不安になりませんか？
- 分析が再現できることは，その研究の信頼性が高いことを示している．
- 統計処理はあくまでも「プロセス」なので，決まった形式が存在している．同じ分析結果を出力するための技術は身につける必要がある．

間違いの検証

- 人間の作業には何らかの間違いが発生しがち．
- 特に，分析過程でコードのどこかに間違いが存在することがある．
- 再現可能なデータ分析を行うことで，間違いを探すことができる．
- **間違いは罪ではないが，「どこで間違ったかわからなくする」のは罪である．**
  - 間違ったことを責めるのではなく，どこに原因があるのかを探す & 見つけられることが重要．

作業効率の向上

- 作業の大半を自動化できており，作業時間を減少することができる．
- 間違いの検証にかかる時間も大幅に減少することが可能となる．
- 本当は R を使うと同じコードを使いまわしできる．
  - 必要に応じて過去に使ったコードを使う必要がある．

作業を進める際には以下のことを気をつけましょう．

- データソースを手で加工，整形していないか
  - どんなことをやったかわからなくなりがちなので，極力データは RStudio の上で加工するようにしましょう．
  - とはいえ，最初はいきなりこれも厳しいか．

- コピペを行っていないか
  - R のコードを R スクリプトにコピペする作業は除く
- コンソールに直接コマンドを入力していないか
  - R スクリプトを作成する際の動作確認やインストールするためのコマンドはコンソールに直接入力して良い

カテゴリーデータの分析

- 第 5 講：カテゴリーデータの分析
- 第 6 講：カテゴリーデータの分析

実証分析の手続き

実証分析とは：

- 実証分析：**客観的**にたくさんのケースにまたがって多量のデータを収集した上で，**統計的**な手法によってそれを分析しようとする方法（森田, 2014）.
  - ただし，個別具体的な事例に踏み込んでの議論には合わないが，**一般性・客観性のある議論には適している**.
  - 個別具体的な事例に踏み込んだ議論は分析者の主観的観点が含まれてしまうために，客観性に劣ってしまう.
  - いわゆる「質的研究」が抱える課題

データの分類

- 「データの分類」を改めて確認しましょう.

量的／質的 | データの名称 | 測定尺度 | 直接できる演算 | 主な代表値 | | | | 量的データ | 比率データ | 比率尺度 | ＋− ×÷ | 各種平均量的データ | 間隔データ | 間隔尺度 | ＋− | 算術平均質的データ | 順位データ | 順位尺度 | ＞＝ | 中央値質的データ | カテゴリデータ | 名義尺度 | 度数カウント | 最頻値

（参考：入門統計学-検定から多変量解析・実験計画法まで-(栗原伸一)）

仮説とは

これから統計的な手法を学ぶ上で，大事なことは「仮説検証」という考え方です．統計学は「対立仮説」と「帰無仮説」の 2 つの仮説を元に考えていきます.

帰無仮説と対立仮説

- 対立仮説：一番主張したいこと，H_1
  - ゴトウは若い．
  - ゴトウはイケメンである．
  - カレーは飲み物である．
  - 授業は楽しい．
- 帰無仮説：主張したいことではないこと，H_0
  - ゴトウは若いとはいえない．
  - ゴトウはイケメンであるとはいえない．
  - カレーは飲み物であるとはいえない．
  - 授業は楽しいとはいえない．
- 対立仮説：一番主張したいことです．
  - 統計学ではこの「対立仮説」を「採択」するためにあーでもない，こーでもないとひたすら戦います．一方，この対立仮説が採択されなかった場合には，「帰無仮説」が採択されることになります．
- 「帰無仮説」の各項目を見てみると，いずれも煮え切らない態度でイライラするかもしれません．しかし，統計学では実は対立仮説が選ばれなかった場合には，この煮え切らないイライラする結論しか出せないのです．

  昨今では，この煮え切らないイライラする姿勢は良くない！ということで「ベイズ統計学」という手法であったり，「効果量」という概念を用いて分析・検討を行うことがあります．この点についてはこの授業の中では触れられないので，ご了承ください．でも，ちょっとやってみたい！って方がいればやってみましょう．

まずは「対立仮説」と「帰無仮説」という考え方を理解して下さい．その上で，統計分析を行うときにはその仮説にあわせて手法を考えることになります．

途中で対立仮説や帰無仮説を「選ぶ」or「採択する」という表現が出てきました．統計学では，この選んだり採択する基準として「p 値」というものを使います．正確には，「平均や標準偏差などを計算する」→「t 値や z 値を算出する」→「p 値を算出する」という手順を踏むことになります．

この授業では，基本的な考え方を理解してもらった上で，実際に関連する数値を見て分析・考えていくという流れを追いますが，一部には時間の都合上，説明が端的になってしまう部分もあります．その際は，各自で統計学に関する教科書を覗いていただければ幸いです．

分析の方法による仮説の作り方

※ ここでは基本的な分類のみを説明しています．

関係を明らかにする分析手法

- 回帰分析：A という変数と B という変数の間に相関があるか否か
  - 応答変数：量的変数
  - 説明変数：量的変数
- χ 二乗検定：A 群と B 群の間が独立しているか否か
  - 応答変数：質的変数
  - 説明変数：質的変数

差異を明らかにする分析手法

- t 検定：A 群と B 群の間に差があるか否か
  - 応答変数：量的変数
  - 説明変数：質的変数 (2 値データ)
- 分散分析：A 群と B 群と C 群と... の間に差があるか否か
  - 応答変数：量的変数
  - 説明変数：質的変数 (3 つ以上のデータ)

差異を一定にしたまま関係を明らかにする分析手法 or 関係を一定にしたまま差異を明らかにする分析手法

- 重回帰分析
  - 応答変数：量的変数
  - 説明変数：質的変数複数 or 量的変数複数 or 量的変数＆質的変数 etc...

クロス集計

クロス集計：

- クロス集計表：複数の質問項目を組み合わせて集計する手法
  - ex. 朝食を食べているか否か × 深夜アルバイトしているか否かなど.
  - 企業の中でも基本的な統計手法としてよく用いられている.

  - 2 つの質的変数間の関連性である「連関」を示す.

組み合わせの数をカウントする.

- 使用するパッケージ

```r
# install.packages("dplyr") # 最初のみ
library(dplyr)

# install.packages("tidyr") # 最初のみ
library(tidyr)
```

地域ごとに子どもがいる人の数を数える．

- 2 つの手法
  - dplyr の group_by 関数を使う方法
  - table 関数を使う方法
  - 現在では前者がメインの手法だが，念のために後者の方法についても紹介する．
  - 今の御時世の最先端の関数を使っている

dplyr の group_by 関数を使う方法

```r
tablea<-exdataset %>%
  group_by(ARE, CHI) %>% # 地域ごとにまとめる関数
  tally() %>% # 地域ごとに数える関数
  spread(ARE, n) # 数えた結果を地域ごとにまとめる関数
tablea
```

```
## # A tibble: 2 x 9
##   CHI     Kanto Hokkaido Tohoku Chubu Kinki Chugoku Shikoku Kyushu
##   <fct>   <int>    <int>  <int> <int> <int>   <int>   <int>  <int>
## 1 NoChild   192       14     35    68    79      31      10     39
## 2 Child     184       21     29    80    86      34       8     53
```

dplyr の count 関数を使う方法

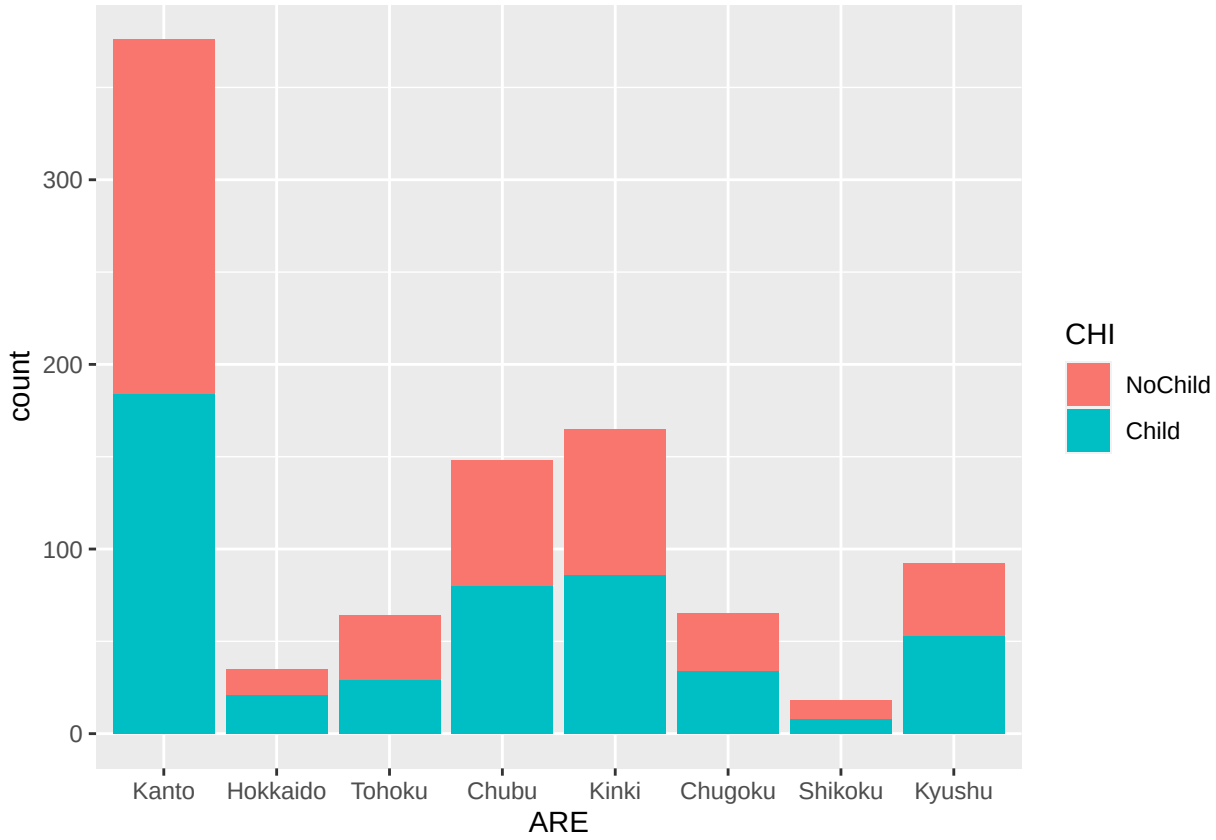
```r
tableb<-exdataset %>%
  count(ARE, CHI) %>%
  spread(ARE, n)
tableb
```

```
## # A tibble: 2 x 9
##   CHI     Kanto Hokkaido Tohoku Chubu Kinki Chugoku Shikoku Kyushu
```

```
##   <fct>   <int>    <int>  <int> <int> <int>   <int>   <int>  <int>
## 1 NoChild   192       14     35    68    79      31      10     39
## 2 Child     184       21     29    80    86      34       8     53
```

参考：ggplot2 で可視化する方法

```
library(ggplot2)
exdataset%>%
  ggplot(aes(x=ARE, fill=CHI),stat="count")+geom_bar()
```




```
# aes() 内で変数名を指定する
# が，後ほど紹介する esquisse() を使うとより容易にヒストグラムを作成できる．
```

xtabs 関数を使う方法

```
tablec<-xtabs(~ CHI, exdataset)
tablec
```

```
## CHI
## NoChild   Child
##     468     495
```

```
tabled<- xtabs(~ ARE, exdataset)
tabled
```

```
## ARE
##    Kanto Hokkaido   Tohoku    Chubu    Kinki  Chugoku  Shikoku   Kyushu
##      376       35       64      148      165       65       18       92
```

table 関数を使う方法

```
tablee<-xtabs(~ ARE + CHI, exdataset)
tablee
```

```
##           CHI
## ARE        NoChild Child
##   Kanto        192   184
##   Hokkaido      14    21
##   Tohoku        35    29
##   Chubu         68    80
##   Kinki         79    86
##   Chugoku       31    34
##   Shikoku       10     8
##   Kyushu        39    53
```

- 行のパーセント表示

```
tableg<-prop.table(tablee, 1)
tableg
```

```
##           CHI
## ARE          NoChild     Child
##   Kanto    0.5106383 0.4893617
##   Hokkaido 0.4000000 0.6000000
##   Tohoku   0.5468750 0.4531250
##   Chubu    0.4594595 0.5405405
##   Kinki    0.4787879 0.5212121
##   Chugoku  0.4769231 0.5230769
##   Shikoku  0.5555556 0.4444444
```

```
##   Kyushu   0.4239130 0.5760870
```

- 列のパーセント表示

```
tableh<-prop.table(tablee, 2)
tableh
```

```
##           CHI
## ARE           NoChild      Child
##   Kanto    0.41025641 0.37171717
##   Hokkaido 0.02991453 0.04242424
##   Tohoku   0.07478632 0.05858586
##   Chubu    0.14529915 0.16161616
##   Kinki    0.16880342 0.17373737
##   Chugoku  0.06623932 0.06868687
##   Shikoku  0.02136752 0.01616162
##   Kyushu   0.08333333 0.10707071
```

もっと細かいクロス集計表を出してみよう

```
xtabs(~ CHI + ARE +F_SEX, exdataset)
```

```
## , , F_SEX = female
##
##          ARE
## CHI       Kanto Hokkaido Tohoku Chubu Kinki Chugoku Shikoku Kyushu
##   NoChild   103        7     19    30    39      17        5     22
##   Child     117       11     16    49    56      24        4     33
##
## , , F_SEX = male
##
##          ARE
## CHI       Kanto Hokkaido Tohoku Chubu Kinki Chugoku Shikoku Kyushu
##   NoChild    88        7     16    37    39      14        5     16
##   Child      65       10     13    31    30      10        4     20
##
## , , F_SEX = other
##
##          ARE
## CHI       Kanto Hokkaido Tohoku Chubu Kinki Chugoku Shikoku Kyushu
```

```
##   NoChild     1        0      0     1     1       0       0      1
##   Child       2        0      0     0     0       0       0      0
```

連関係数を出力しよう

- 連関係数：クラメール連関係数 V
  - 下限が 0，上限が 1 で完全な連関に近づくにつれて 1 に近い値を取る．

```
#install.packages('vcd', dependencies = T)：初回のみ
library(vcd)
```

```
## Loading required package: grid
```

```
assocstats(tablee)
```

```
##                     X^2 df P(> X^2)
## Likelihood Ratio 5.1570  7  0.64082
## Pearson          5.1408  7  0.64278
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.073
## Cramer's V        : 0.073
```

- 今回の場合は，地域と子供の有無にほとんど連関は認められませんでした．

χ^2 検定を行いましょう

2 種類の χ^2 検定

- χ^2 検定はその目的に応じて 2 種類ある
- 適合度検定：観測度数が理論比率にもとづいて得られるかどうかを検証する仮説検定
- 独立性検定：複数の特性の間に関連があるかどうかを調べる仮説検定

適合度検定：

普通のサイコロを振ったときに，各目が等しい確率で出る．

あるサイコロを振ったとき，以下のような結果が得られた．このサイコロは「普通のサイコロ」であろうか？それとも，「普通ではないサイコロ」ではないだろうか？

- 1:40
- 2:21
- 3:40
- 4:90
- 5:50
- 6:70

適合度検定

- 対立仮説：観測された頻度分布と期待される頻度分布に差がある．
- 帰無仮説：観測された頻度分布と期待される頻度分布に差があるとは言えない．

```
psy <- c(40, 21, 40, 90, 50, 70)
# サイコロの出た目
the_psy <- c(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)
# サイコロの目の理論値
```

適合度検定の実施

```
chisq.test(psy, p = the_psy)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  psy
## X-squared = 58.28, df = 5, p-value = 2.754e-11
```

- p 値は有意水準を大きく下回るために帰無仮説を棄却し，対立仮説を採択する．
- このサイコロは「普通ではないサイコロ」である．

独立性の検定

- 性別と旅行の好みについて，以下のクロス表が得られた場合の変数 A および B の独立性の検定を行う．

	旅行好き	どちらともいえない	旅行嫌い
男性	70	50	60
女性	40	30	20

独立性検定

- 対立仮説：性別と旅行の好みに関連性がある
- 帰無仮説：性別と旅行の好みに関連性があるとは言えない（独立である）

独立性検定

```
ryoko_seibetsu <- matrix(c(70, 50, 60, 40, 30, 20),
                         nrow = 2, byrow = T)
# 行列をオブジェクトにしまう.
```

独立性検定

```
chisq.test(ryoko_seibetsu)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ryoko_seibetsu
## X-squared = 3.5795, df = 2, p-value = 0.167
```

- p 値は有意水準より大きいために帰無仮説を採択する.
- 性別と旅行の好みに関連性があるとは言えない（独立である）

χ^2 検定

- 対立仮説：居住地域と子供の有無は独立ではない（連関がある）
- 帰無仮説：居住地域と子供の有無は独立である（連関があるとは言えない）

```
chitest.tablee<-chisq.test(tablee)
chitest.tablee
```

```
##
##  Pearson's Chi-squared test
##
## data:  tablee
## X-squared = 5.1408, df = 7, p-value = 0.6428
```

- 検定の結果，p 値が.05 以上なので，対立仮説を採択できず，帰無仮説を採択する．

χ^2 検定

- レポートにまとめる時には，こんな書き方をします．

  χ^2 検定を行った結果，居住地域と子供の有無は独立であることがわかった (=5.1408, df=7, p=.64)．

- もし，χ^2 検定で p 値が.05 以下であった場合，残差分析を行います．
  - どのセルで有意な逸脱が生じたのかを検討する．
  - 標準化残差が 1.96 以上であれば，5% 水準で有意な逸脱があったと評価する．

```
chitest.tablee$stdres
```

```
##           CHI
## ARE           NoChild      Child
##   Kanto     1.2252616 -1.2252616
##   Hokkaido -1.0367594  1.0367594
##   Tohoku    1.0087797 -1.0087797
##   Chubu    -0.7017295  0.7017295
##   Kinki    -0.2030909  0.2030909
##   Chugoku  -0.1513129  0.1513129
##   Shikoku   0.5961874 -0.5961874
##   Kyushu   -1.2524729  1.2524729
```

- もしくは，以下の計算で p 値を算出しても良い．

```
pnorm(abs(chitest.tablee$stdres), lower.tail = FALSE) * 2
```

```
##           CHI
## ARE           NoChild     Child
##   Kanto    0.2204767 0.2204767
##   Hokkaido 0.2998480 0.2998480
##   Tohoku   0.3130803 0.3130803
##   Chubu    0.4828479 0.4828479
##   Kinki    0.8390640 0.8390640
##   Chugoku  0.8797289 0.8797289
##   Shikoku  0.5510501 0.5510501
##   Kyushu   0.2103976 0.2103976
```