統計学第4/5講

明治大学情報ｺﾐｭﾆｹｰｼｮﾝ学部
後藤　晶
akiragoto@meiji.ac.jp

今日のお話

前回の続き

分散と標準偏差を手計算で算出してみよう

分子を計算する

分母を計算する

最後の計算

標準偏差を算出する

3種類の分散と標準偏差

前回の続き

平均値・分散・標準偏差とは？

- 平均値：全てのデータを足して割ったもの．一般的に代表値（データ全体を表している数値）として扱われる．
- 分散：平均値とそれぞれの値の差を求めて2乗して，合計したものをデータの個数で割ったもの．データの散らばり具合を示す数値であり，分散が大きければ大きいほど，データが散らばっていることを示す．
  - σ^2 という記号で表される．
  - $(\ \ ) = \Sigma\{(\ ) - (\ \ )\}^2/(\ \ )$
- 標準偏差：分散の平方根．通常の長さのばらつきを評価する際には同じ単位で理解したほうがわかりやすいために用いる．
  - σ という記号で表される．

その他，重要な指標

- 最小値：そのデータの中で最も小さい値
- 第一四分位数（25%パーセンタイル値）：最小値と中央値の間の中央値
- 中央値（第二四分位数）：データを大きい(小さい)順に並べたとき，真ん中の値のこと(median)．外れ値がある時に代表値として用いられる．
  - 奇数の場合：ちょうど真ん中が存在する．
  - 偶数の場合：真ん中の数字2つの平均値を中央値とする．

- 最頻値：データの中で最も多く出てくる値のこと(mode)．因子データの際に代表値として使われる．
- 第三四分位数（75%パーセンタイル値）：中央値と最大値の間の中央値
- 最大値：そのデータの中で最も大きい数
- 以下の2つは参考までに．
  - 平均偏差：「平均からの偏差」の絶対値の平均
  - 範囲：最大値から最小値の間．引き算で求められる．

Rにおける「関数」とは？

- 関数：頻繁に用いられるデータ操作方法や，標準的な統計計算をまとめてオブジェクトにしたもの．正式には「関数オブジェクト」
  - 簡単に計算できるように，先人たちがまとめたものだと理解すれば良い．
  - これを用いることで，簡単に計算ができる．

基本的な関数：

- sum：合計
- mean：平均値
- max：最大値
- min：最小値
- range：範囲（最大値-最小値）
- median：中央値
- var：不偏分散
- sd：標準偏差
- quantile：四分位点
- IQR：四分位範囲
- summary：要約統計量
- sqrt：平方根

基本的な関数：

- abs：絶対値
- round：値の丸め
- floor：値の切り捨て
- ceiling：値の切り上げ
- log：自然対数
- log10：10を底とする対数
- log2：2を底とする対数
- log1p：1を加算した自然対数
- exp：指数関数
- sin, cos, tan：三角関数
- asin, acos, atan：三角関数の逆関数

データセットを作ろう

7人の学生の体重が50, 60, 85, 70, 80, 67, 66kgであったする．このデータを変数名“weight”に代入する．

```
weight<-c(50, 60, 85, 70, 80, 67, 66)
```

記述統計量を色々出してみる.

```
sum(weight)/7
```

```
## [1] 68.28571
```

- sum()関数で合計を算出できる.

```
sum(weight)/length(weight)
```

```
## [1] 68.28571
```

- length()関数でデータの個数を数える.

```
mean(weight)
```

```
## [1] 68.28571
```

- 実はmean()という関数を使うと一発で出てしまう.

```
median(weight)
```

```
## [1] 67
```

- 中央値はmedian()という関数で出せる.

```
table(weight)
```

```
## weight
## 50 60 66 67 70 80 85
##  1  1  1  1  1  1  1
```

- 最頻値はtable()という関数を使って探し出す.
- ちなみに，“weight”の中に最頻値は存在していない．(全てが最頻値=1)

分散と標準偏差を手計算で算出してみよう

分散を算出する

- 定義通りの算出方法

$$\sigma^2 = \Sigma((\ ) - (\quad))^2/(\ \ )$$

- 簡便に算出する際に用いられる数式

$$\sigma^2 = \Sigma((the\ value\ of data)^2/(a\ number\ of\ data))$$

$$-(\Sigma((the\ value\ of data)/(a number\ of\ data))^2)$$

- 「2乗の平均」-「平均の2乗」
- ここでは，定義通りの数式から分母と分子に分けて話を進めていきましょう.

分子を計算する

体重の平均値をオブジェクトに入れる

- “mean_weight”というオブジェクトを作って，体重の平均値を入れます．

```
mean_weight <- mean(weight)
mean_weight
```

```
## [1] 68.28571
```

平均からの偏差を求めて，オブジェクトに入れる

- (データの値)-(weightの平均値)をして「平均からの偏差」を求めます．結果は“hensa_weight”に代入します．

```
hensa_weight <- weight - mean_weight
hensa_weight
```

```
## [1] -18.285714  -8.285714  16.714286   1.714286  11.7142
```

「平均からの偏差」を2乗する

- 「平均からの偏差」を2乗します．“hensa_weight2”というオブジェクトを作って代入をしましょう．
  - 2乗しないとで全部足すと，数字は0になります．
  - ただし，小数点以下を四捨五入しているので，ここでは完璧に0にはなりませんが，限りなく0に近くなります．

```
hensa_weight2 <- hensa_weight^2
hensa_weight2
```

```
## [1] 334.367347  68.653061 279.367347   2.938776 137.224
```

「平均からの偏差の2乗」を全部足してオブジェクトに入れる

- これらの5つの値を合計した「平均からの偏差の二乗和」を求めます.
  - “sum_hensa_weight2”という名前にしましょう．これで分子は完成です.

```
sum_hensa_weight2<-sum(hensa_weight2)
sum_hensa_weight2
```

```
## [1] 829.4286
```

分母を計算する

データの個数を数えてオブジェクトに入れる

- 今度は分母を算出します．分母はデータ数です，“length_weight”というオブジェクトに代入しましょう．

```
length_weight<-length(weight)
length_weight
```

```
## [1] 7
```

最後の計算

分散の算出

- 分散は「平均からの偏差の二乗和」/「データ数」ですから，以下の通りに求められます．
  - 分散は“vari_weight”というオブジェクトに入れましょう

```
vari_weight<-sum_hensa_weight2/length_weight
vari_weight
```

```
## [1] 118.4898
```

標準偏差を算出する

標準偏差を算出する

$$\sigma^2 = \Sigma((the\ value\ of data)^2/(a\ number\ of\ data))$$

$$-(\Sigma((the\ value\ of data)/(anumber\ of\ data))^2)$$

$$\sigma = \sqrt{\sigma^2}$$

- 標準偏差は分散の平方根です.
  - 平方根を求める関数は“sqrt()”であり，“hyohen_weight”というオブジェクトに入れてあげます.

ルートをとります

```
hyohen_weight <- sqrt(vari_weight)
hyohen_weight
```

```
## [1] 10.8853
```

分散の別解

$$\sigma^2 = \Sigma((the\ value\ of data)^2/(a\ number\ of\ data))$$

$$-(\Sigma((the\ value\ of data)/(anumber\ of\ data))^2)$$

```
sum(weight^2/length(weight))-sum(mean(weight)^2)
```

```
## [1] 118.4898
```

演習問題6：

- “weight”と同様に，変数名“height”について分散・標準偏差を計算してください.
  - ヒント：ちょっと書き換えるだけですぐいけます.
  - “ctrl + f”(macでは“cmd + f”)で置換ができます.
  - 基本方針は「極力手抜きをしましょう」です.

関数を使って分散と標準偏差を算出する.

- 分散と標準偏差はよく算出します. 当然Rでは関数が用意されています.

```
var(weight)
```

```
## [1] 138.2381
```

- 分散

```
sd(weight)
```

```
## [1] 11.75747
```

- 標準偏差
- 関数で求められるのは「不偏分散・不偏標準偏差」
- 手計算で求めたのは「標本分散・標本標準偏差」

2種類の分散と標準偏差

2種類の分散と標準偏差？？

「不偏分散・不偏標準偏差」と「標本分散・標本標準偏差」というものが出てきました．この話を理解するためには「母集団」と「標本」という話を理解する必要があります．ここでは簡単に，その2つの違いについてお話したいと思います．

私達が何かのデータを取る時は，全ての物事のデータを集めることが必ずしもできるとは限りません．例えば，「本学大学生1年生全員を対象としたアンケート」を実施すれば全てのデータを集めることができるかもしれませんが，「日本国民全てを対象としたアンケート」を集計するのは非常に困難です．

例えば，大学1年生の意見を調査することを目的として，1年生全員のデータをそのまま用いる分には問題ないのですが，「日本国民全てを対象としたアンケート」を実施するのはコストの面から考えても現実的ではありません．そのために，全体（母集団）の中から一部を取り出して（標本，サンプル），全体の意見・傾向を「推定」するという手法がとられるようにな

このような「推定」という手法を取る時に，“データ数”のままで分析するよりも“データ数-1”で計算してあげたほうがよりよい推定ができる，ということで“データ数-1”をするようになりました.

本当はもう少し細かな数学的な議論があるのですが，入り込むと帰って来れなくなるのでここまでにしておこうと思います.
とりあえず，これからは「不偏分散・不偏標準偏差」が使われることが多い，とだけ覚えておいて下さい.

興味のある方はコチラをご参照ください.

関数と手計算が一致することを確認する.

分散

```
var(weight)
```

```
## [1] 138.2381
```

```
huhen_vari_weight<-sum_hensa_weight2/(length_weight-1)
huhen_vari_weight
```

```
## [1] 138.2381
```

関数と手計算が一致することを確認する.

標準偏差

```
sd(weight)
```

```
## [1] 11.75747
```

```
huhen_hyohen_weight<-sqrt(huhen_vari_weight)
huhen_hyohen_weight
```

```
## [1] 11.75747
```

データの取扱い方

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)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

```
## Parsed with column specification:
```

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.

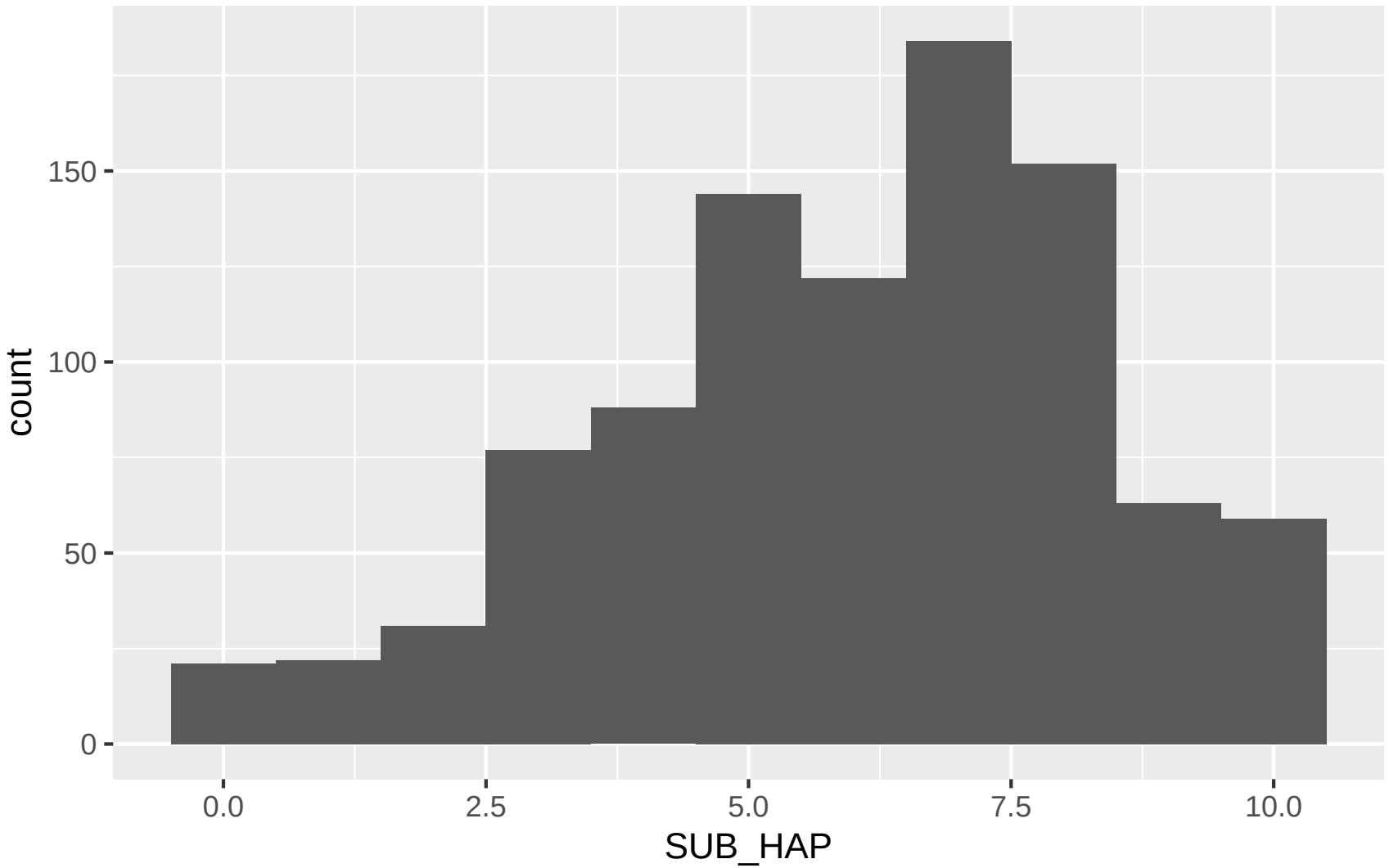
主観的幸福度(SUB_HAP)の標準偏差

```
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(bi
```



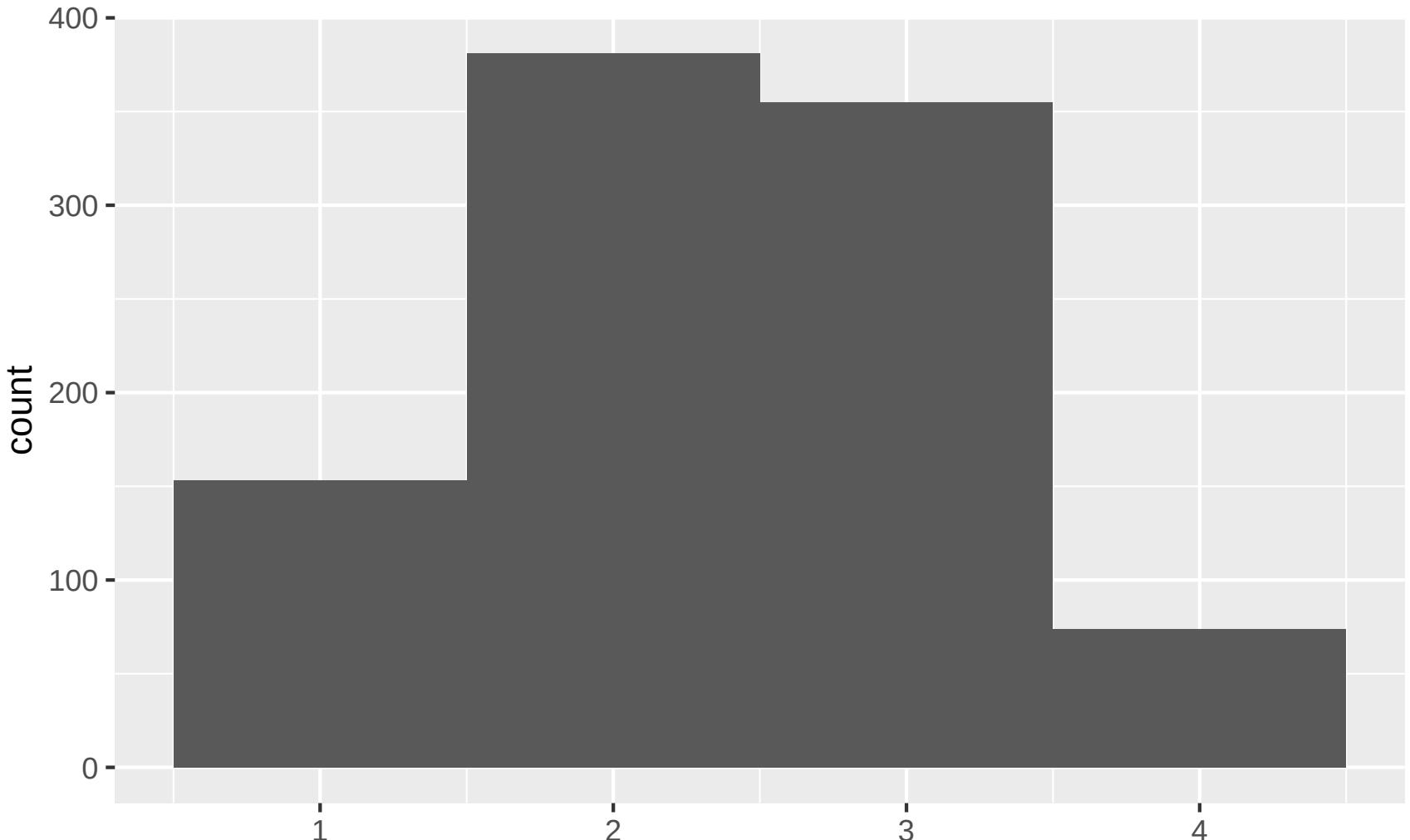
運命(SPN_UNM)の頻度を数えてみる.

```
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

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
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…

相関係数とは

概要

相関係数とは，数値データ同士の関連性を探る指標です．相関係数の絶対値が0に近いと2つの変数同士には線形関係がないことを示します．

- $|r|=1.00$：完全に相関がある
- $0.70<|r|<1.00$：高い相関がある
- $0.40<|r|<0.70$：中程度の相関がある
- $0.20<|r|<0.40$：低い相関がある
- $0.00<|r|<0.20$：ほとんど相関がない
- $|r|=0.00$：完全に無相関である．

概要

ちなみに，この「相関の強さ」について分野によって評価が異なります．例えば，社会科学研究では高い相関が認められることは少ないです．今回の基準で中程度の相関や低い相関で議論をすることもあります．

この辺は分野によって異なりますので，ご承知おきください．

- 次のスライドからは同じ記述統計量の散布図を見てもらって，相関係数を確認することの重要性を感じてもらいます．

パッケージの読み込み

```
library(datasauRus)
```

相関係数で比較をしてみる.

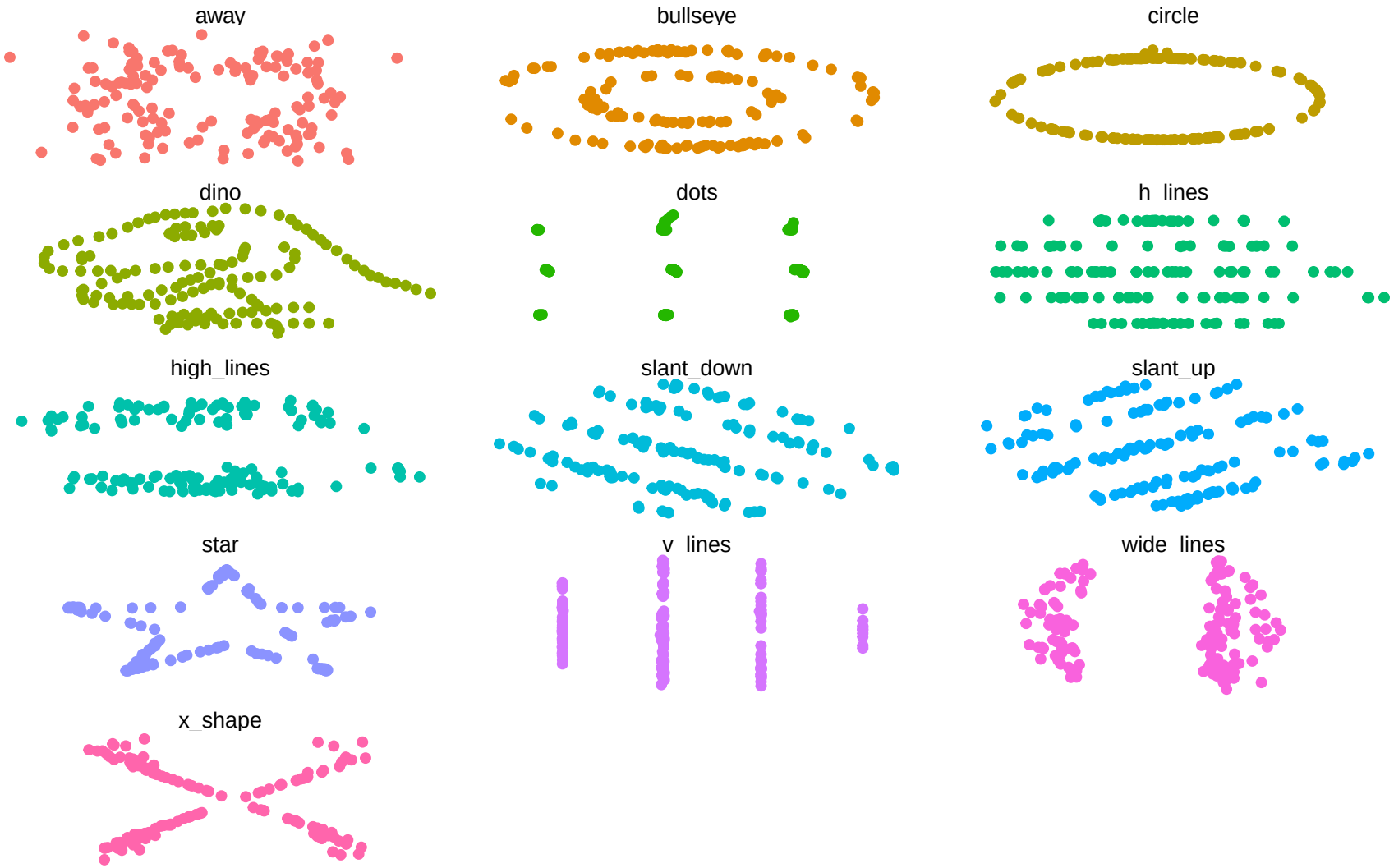
dataset	平均値	標準偏差	標本数	標準誤差
away	54.27	16.77	142	1.407
bullseye	54.27	16.77	142	1.407
circle	54.27	16.76	142	1.406
dino	54.26	16.77	142	1.407
dots	54.26	16.77	142	1.407
h_lines	54.26	16.77	142	1.407
high_lines	54.27	16.77	142	1.407
slant_down	54.27	16.77	142	1.407
slant_up	54.27	16.77	142	1.407
star	54.27	16.77	142	1.407
v_lines	54.27	16.77	142	1.407
wide_lines	54.27	16.77	142	1.407
x_shape	54.26	16.77	142	1.407

相関係数で比較をしてみる.

```
datasaurus<-datasaurus_dozen %>%
             ggplot(aes(x=x, y=y, colour=dataset))+
               geom_point()+
               theme_void()+
               theme(legend.position = "none")+
               facet_wrap(~dataset, ncol=3)
```

相関係数で比較をしてみる.

`datasaurus`



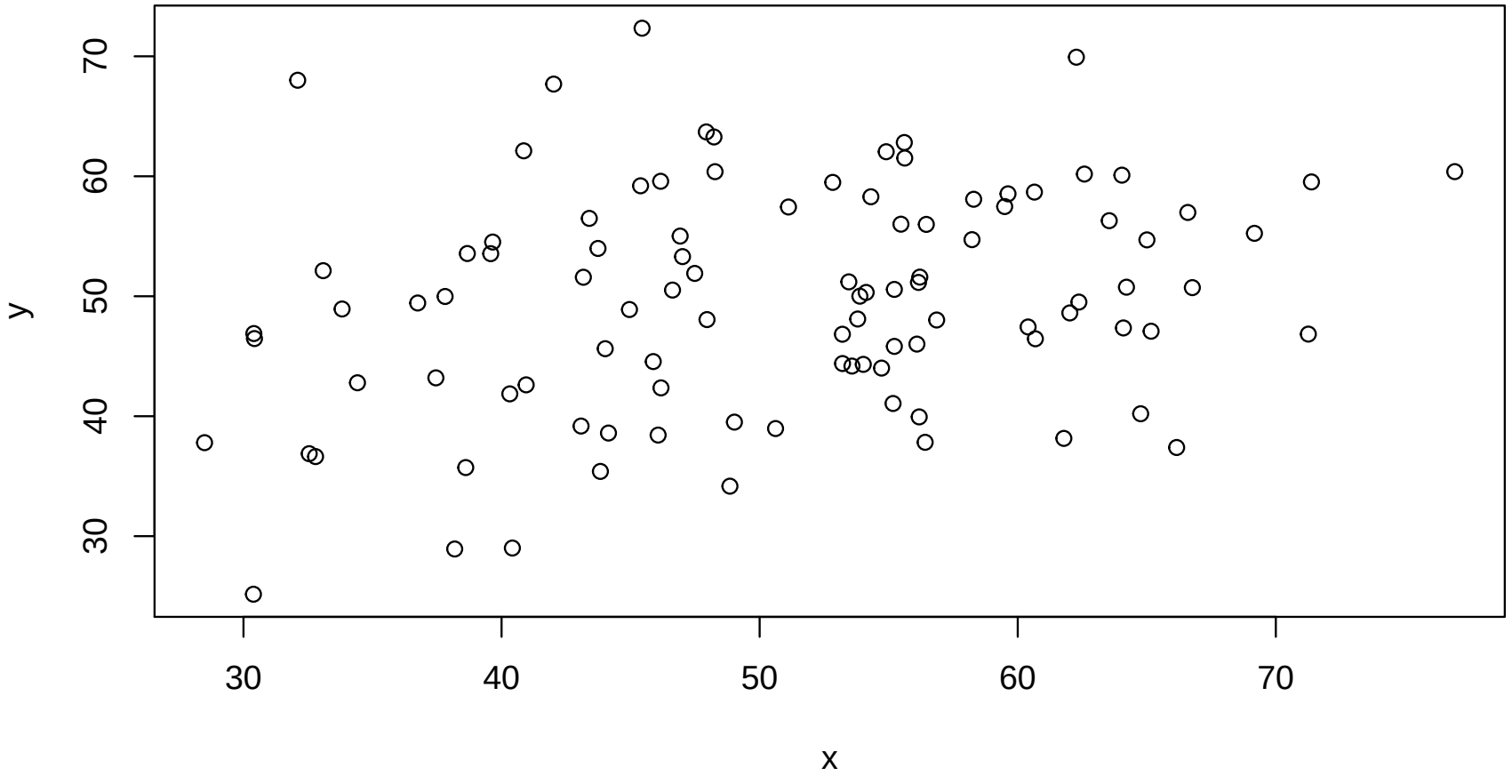
相関係数を出してみる

- 乱数で比較をしてみましょう.
  - 平均50，標準偏差10のデータを100個×2を作ります.
  - さらに，xとyを足して2で割ります.

```
x <- rnorm(100, 50, 10)
y <- rnorm(100, 50, 10)
z <- (x+y) / 2
```

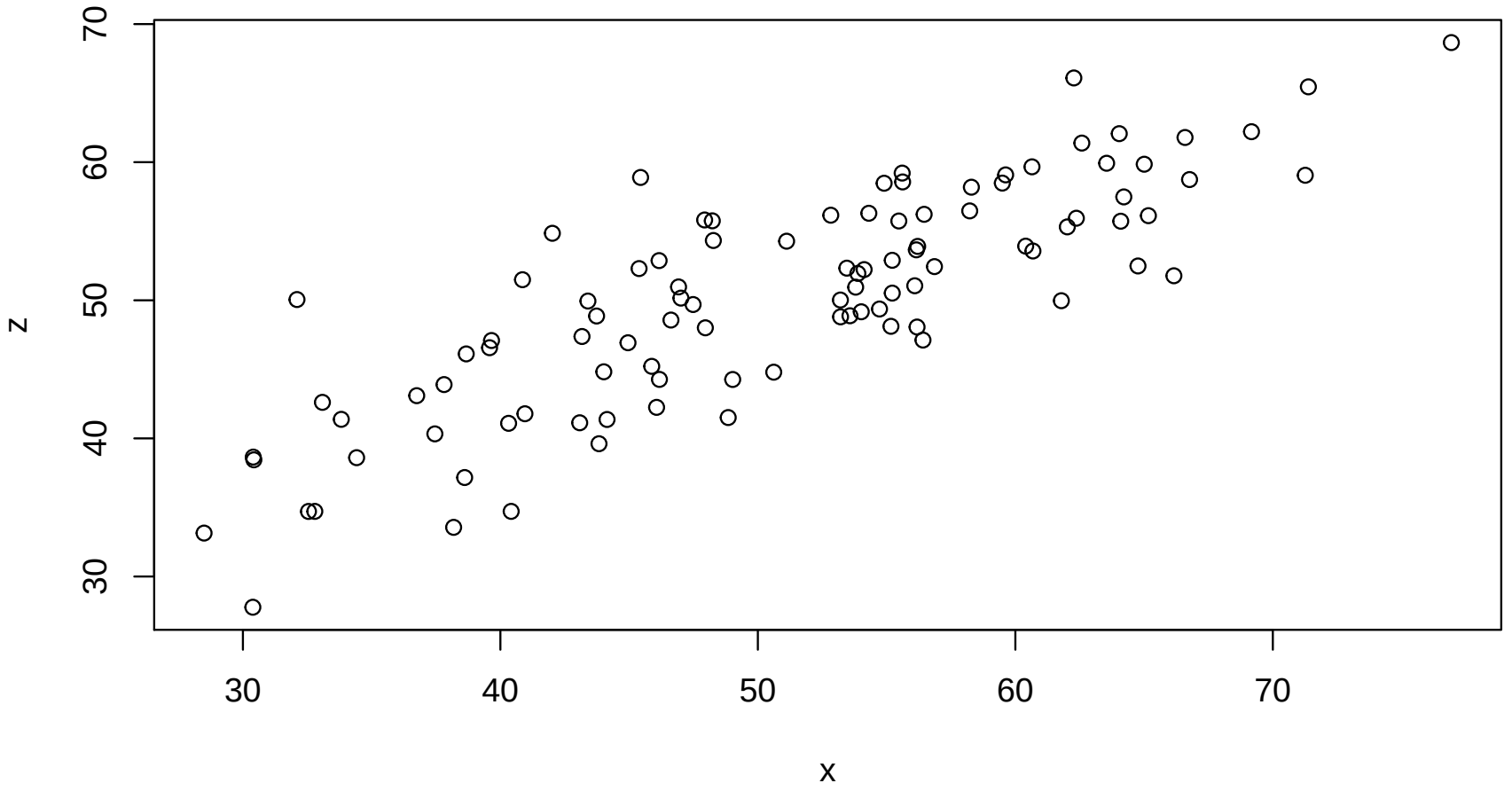
相関係数を出してみる

```
plot(x, y)
```



相関係数を出してみる

```
plot(x, z)
```



（不偏）共分散

- 2種類のデータの関係を示す指標であり，2つの変数の偏差の積の平均を計算する.
- 共分散が大きいほど関係性が強い，，，と言えるが，ちょっと不安がある.
- 「2つの変数の関係の強さ」と「単位」の影響を受けてしまうため，標準偏差の積で割ってあげる必要がある.
- N-1で割ると不偏共分散，Nで割ると標本共分散

$$s_{xy} = \Sigma((dev\ of\ x) * (dev\ of\ y))/(num\ of\ N-1)$$

不偏共分散を算出する

```
x_hensa <- x-mean(x)
y_hensa <- y-mean(y)
goukeixy <- sum(x_hensa * y_hensa)
kyobunsanxy <- goukeixy/(length(x)-1)
kyobunsanxy
```

```
## [1] 26.8348
```

- 演習問題
  - xとzについて，不偏共分散を算出してみよう．

関数で不偏共分散を求める

```
cov(x, y)
```

```
## [1] 26.8348
```

```
cov(x, z)
```

```
## [1] 71.79436
```

相関係数を出してみる

- xとyの相関係数：

$$r = \frac{s_{xy}}{s_x s_y}$$

- s_{xy}：xとyの共分散
- s_x：xの標準偏差
- s_y：yの標準偏差

相関係数を算出する

```
soukanxy <- kyobunsanxy/(sd(x)*sd(y))
soukanxy
```

```
## [1] 0.2659038
```

- 演習問題
  - xとzについて，相関係数を算出してみよう．

関数で相関係数を求める

```
cor(x, y)
```

```
## [1] 0.2659038
```

```
cor(x, z)
```

```
## [1] 0.8278751
```

次回の案内：

次回の案内：

- 今度こそ色々グラフを作ります．
  - とはいえ，今日も作りましたが，

Rでデータを扱う時に注意すべきこと

Rでデータを扱う時に注意すべきこと

- 日本語は使わずにローマ字を使用する.
- コメントアウト（コードではなく，関係ないメモを入れること）をするときは半角の「#」から始める.
  - メモする内容は全角でもよい.
- ファイル名およびパスには決して全角の文字（ひらがな，カタカナ，漢字，全角スペースなど）を入れてはいけない.
  - 半角英数字だけにする.
- 慌てずに落ち着いて操作すれば，決して難しくない.
  - 1つずつ落ち着いて作業することを心がける.
- 「わからない」ことを恐れない
  - 周りの友人に聞いたり，教員に確認したりしよう.