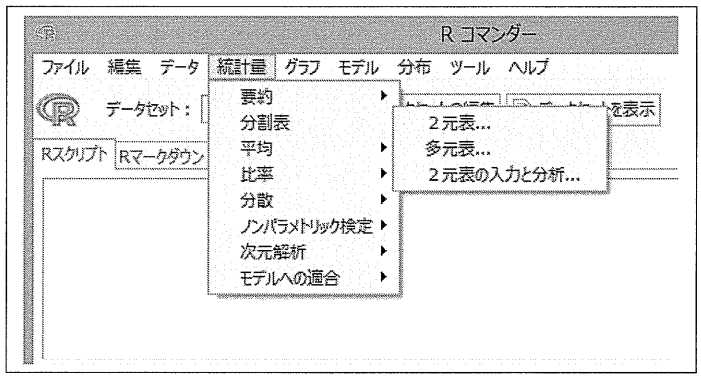
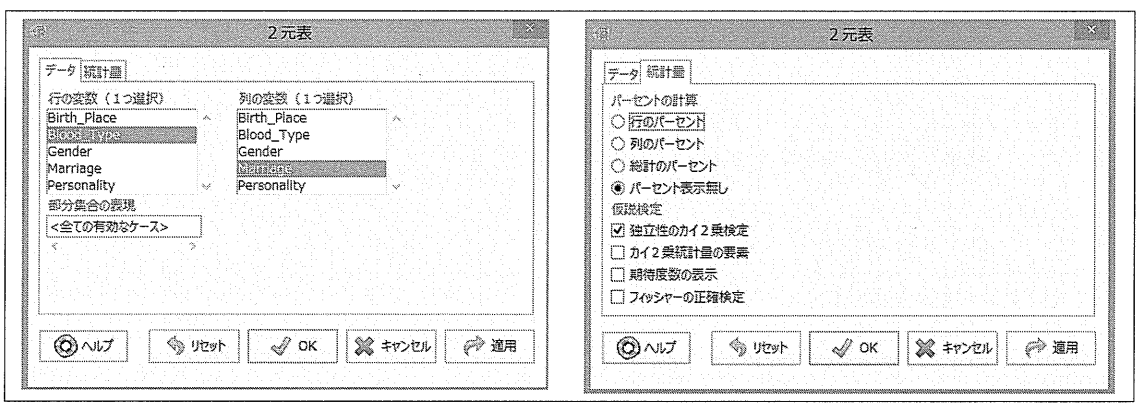


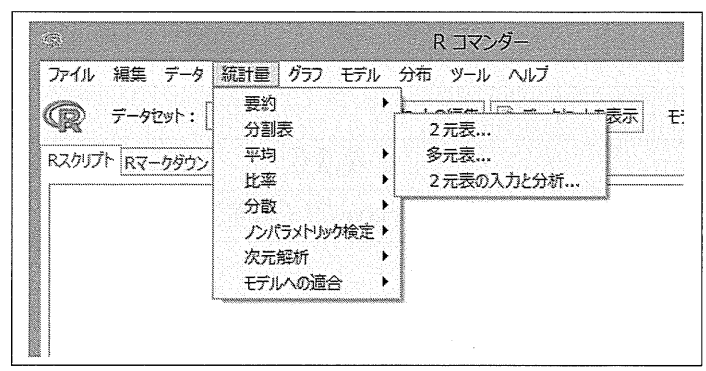


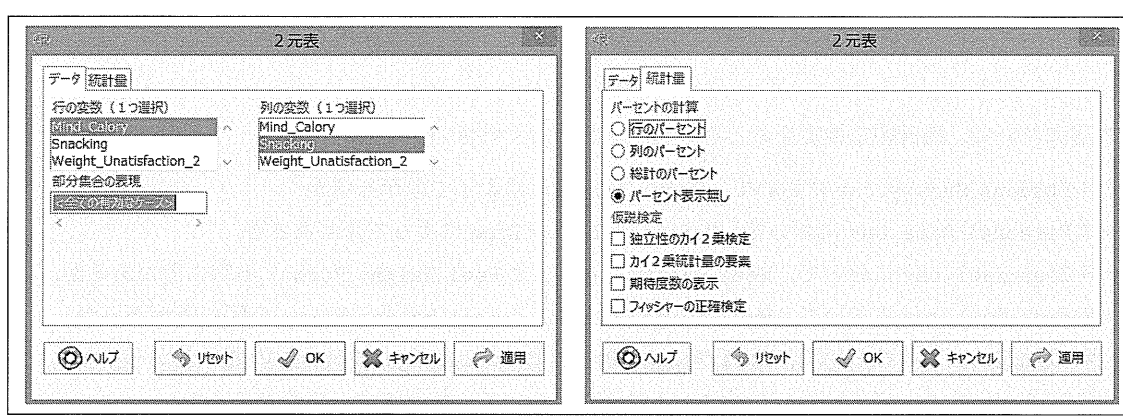
次のようにスクリプトが生成され，結果が表示されます．

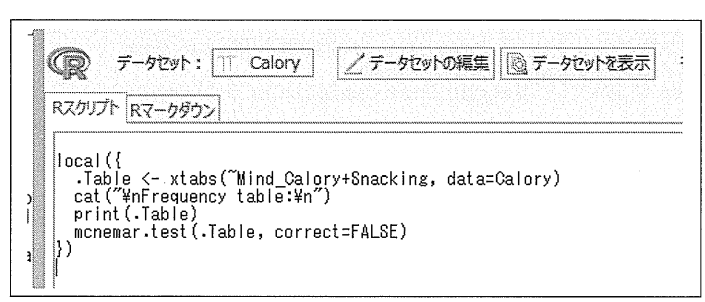
```
> local({
+   .Table <- xtabs(~Blood_Type+Marriage, data=Personal)
+   cat("
nFrequency table:
n")
+   print(.Table)
+   .Test <- chisq.test(.Table, correct=FALSE)
+   print(.Test)
+ })

Frequency table:
          Marriage
Blood_Type Married Single
        A       29     35
        AB      12     15
        B       22     18
        O       26     29

Pearson's Chi-squared test

data:  .Table
X-squared = 1.1175, df = 3, p-value = 0.7729
```

§ 4.15
McNemar の検定

1. 【統計量】→【分割表】→【2 元表】を選びます.
2. 【データ】タブにおいて,【行の選択】で 1 つ変数を選択し,【列の選択】でも変数を 1 つ選びます.
3. 【統計量】タブにおいて,【パーセントの計算】で 1 つ選び,【仮説検定】ではすべてのチェックを外します.
4. 【OK】を押します.
5. その後で，表示された R スクリプトにおいて，図に示した行に `mcnemar.test(.Table, correct=FALSE)` と入力する.
6. 修正したスクリプトを実行する.







次のようにスクリプトが生成され，結果が表示されます.

```
> local(
+   .Table <- xtabs(~Mind_Calory+Snacking, data=Calory)
+   cat("¥nFrequency table:¥n")
+   print(.Table)
+   mcnemar.test(.Table, correct=FALSE)
+ )

Frequency table:
           Snacking
Mind_Calory No Yes
        No   6  16
        Yes  5  27

McNemar's Chi-squared test

data:  .Table
McNemar's chi-squared = 5.7619, df = 1, p-value = 0.01638
```

§ 4.16
Cochran の Q 検定

現行有用な関数は存在せず.

§ 4.17
課題

課題 1

以下の内容について，統計的に検定をおこない解釈を行ってください.

1. データ Student_Survey.csv において，Score が Smoke によって差があるか.
2. データ Personal_Data.csv において，Height が Blood_Type によって差があるか.
3. データ High_Heel.csv において，Heel_Height_Casual と Heel_Height_Formal の間に差があるか.
4. データ High_Heel.csv において，Heel_Height_Casual と Heel_Height_Party の間に差があるか.
5. データ High_Heel.csv において，Heel_Height_Party と Heel_Height_Formal の間に差があるか.
6. データ High_Heel.csv において，ハイヒールの高さについて，状況によって差があるか.
7. データ smoke03.csv において，smoke ごとに Score の分散が等しいか.
8. データ IceCream.csv において，MaxTemp と Qunatity の間に相関はあるか.
9. データ IceCream.csv において，Min と Qunatity の間に偏相関はあるか.
10. データ Calory_Diet.csv において，Weight_Unsatisfaction_5 が Snacking によって差があるか.
11. データ Student_Survey.csv において，の Extraversion が Blood_Type によって差があるかどうか.
12. データ High_Heel.csv において，Cheerfulness と Fashionable の間に差があるか.
13. データ Personal_Data.csv において，Marriage と Blood_Type の間に関連性はあるか
14. データ Calory_Diet.csv において，Mind_Calory と Snacking の間に関連性はあるか
15. データ Calory_Diet.csv において，Weight_Unsatisfaction_5 と Snacking の間に関連性はあるか
16. データ High_Heel.csv において，Cheerfulness と Fashionable の間に相関はあるか.

A

R と Rcmdr

§ A.1
R のインストール

R[*1] とは，数値計算をおこなうための言語・環境です.「R 言語」と呼ばれることもあります．R の特徴は次のようなものです.

1. オープンソースである．したがって，プログラムの設計図とも言うべきソースコードが公開されている．そのために誰でも，プログラムを改良することが出来る.
2. フリーソフトウェアであり，無料で手に入れることが出来る．ただし，結果に対する補償は一切ない.
3. 「パッケージ」と呼ばれる，R 言語のプログラムを配布用の形式にしたものが多数用意されている．このことにより，基本的なプログラムをインストールした後も，自ら必要な機能をパッケージをインストールするという形で追加し，拡張することができるようになっている.
4. 描画機能が優れている．出版に耐えうる水準の図を描くことが出来る.

これらの特長が，R の使用者を近年増やす要因になっています.

ここでは，まず R といくつかのパッケージをインストールし，R を使える環境を整えます．インストールするために，次の準備をしてください.

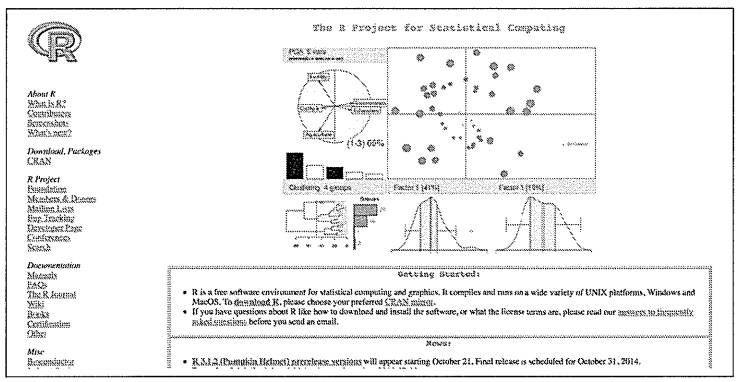
1. インターネットに接続できるよう各自設定してください．R の基本プログラムも，拡張用パッケージも CRAN が設定した世界中にあるいくつかのミラーサーバからダウンロードするようになっています.
2. PC 設定の administrator 権限で PC 環境の変更が出来るようにします．administrator 権限がないと新しいプログラムをインストールすることができません.
3. ウイルス対策ソフト等のダウンロード制限機能を設定しておいてください．ウイルス対策ソフトによっては，ネットワーク経由でダウンロードする実行形式を伴うファイルを制限するものがあります.

インストールは以下の手順でおこないます.

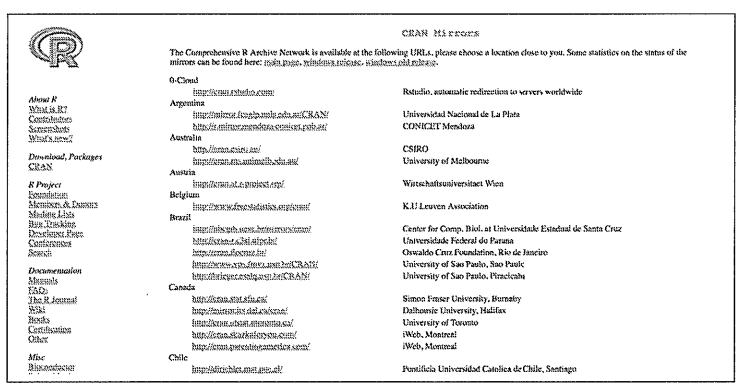
1. R-project のサイト `http://www.r-project.org` をブラウズします (step 1).
2. ホーム画面を経由して，ミラーサイトのうちの一つを選びます (step 2).
3. 【`Download R for Windows`】を選択します (step 3).
4. 【`install R for the first time`】を選択します (step 4).

[*1] `http://www.r-project.org`

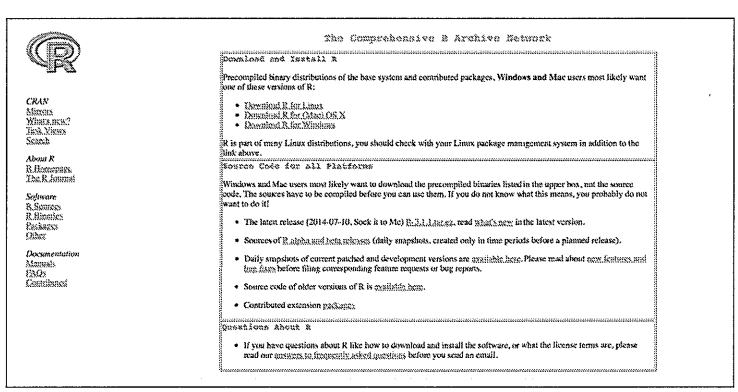
5. 【Download R 3.1.1.for Windows】を選択するとダウンロードが開始されます (step 5).
6. ダウンロードが完了したら，各自の PC に R をインストールします.



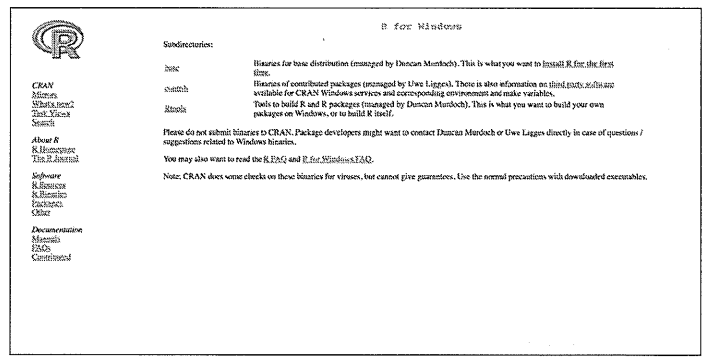
A.1 R project の WEB サイト (step 1)



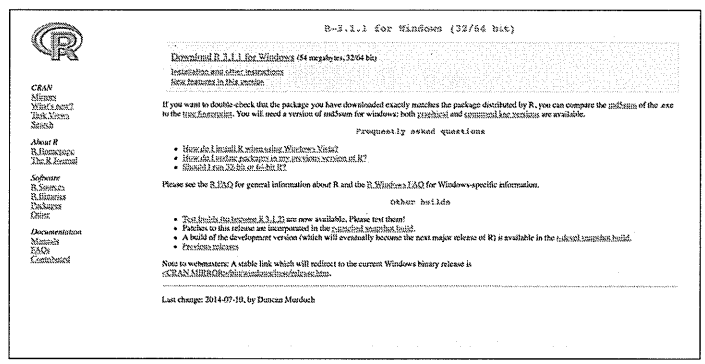
A.2 ミラーサイトの選択 (step 2)



A.3 CRAN の WEB ページ (step 3)



A.4 R のインストール画面 1 (step 4)



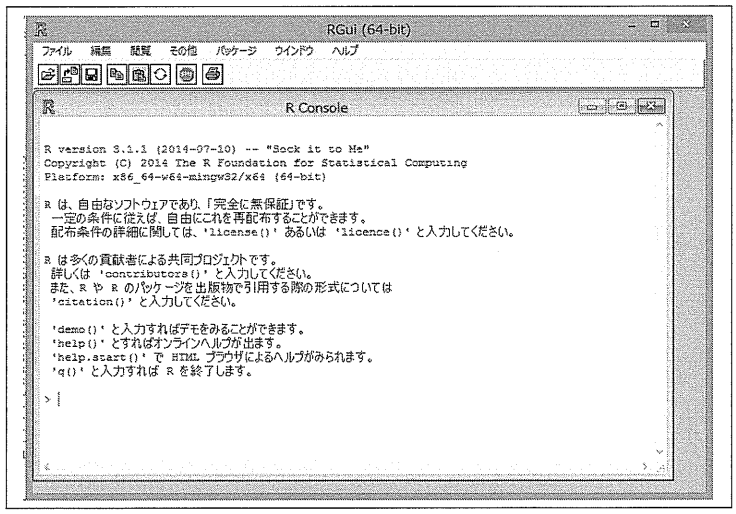
A.5 R のインストール画面 2 (step 5)

§ A.2
Rcmdr のインストール

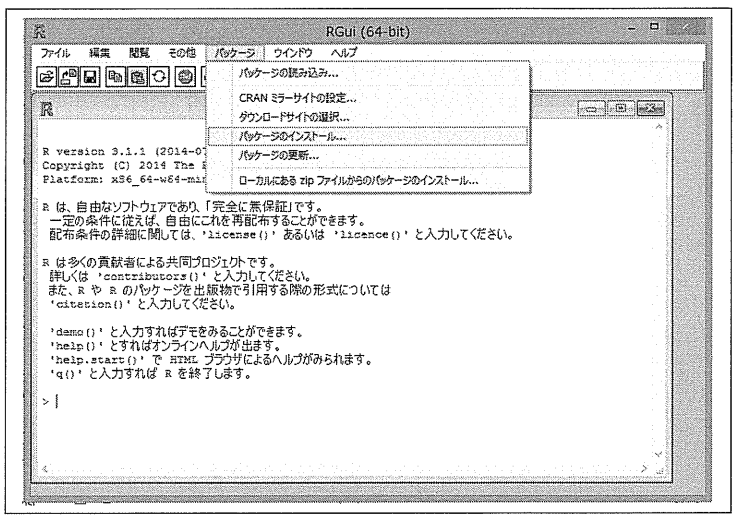
R Commander(Rcmdr) は，R の機能を GUI 方式で利用可能とするパッケージです．データ処理，グラフ作成，統計分析をメニューから選択しながら実行できるようにするものです．R はライン形式のソフトウェアですから，コンピュータが苦手な場合には「操作がとても難しい」というイメージを持ってしまうのが短所の一つです．しかし，拡張用パッケージの一つである Rcmdr をインストールすることによって，多くの商用の統計ソフトウェアと同じく直感的な操作が可能になります．

Rcmdr のインストールの流れは次のようになります．

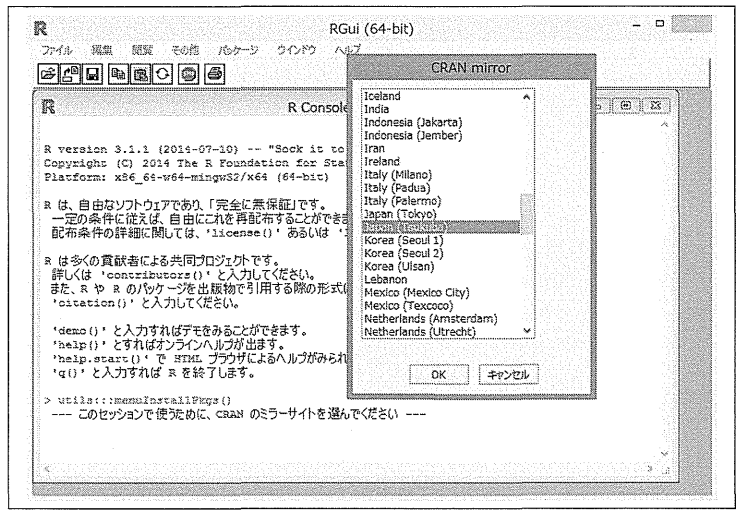
1. R を起動させます．(step 1)
2. 【パッケージ】→【パッケージのインストール】を選びます．(step 2)
3. ダウンロードサイトを一つ選びます．この例では，【Japan(Tsukuba)】 選んでいます．(step 3)
4. 【Rcmdr】を選択します．【OK】を押すとダウンロードが始まります．(step 4)
5. インストール完了後に，コンソール画面に`> library(Rcmdr)` と打ち，エンターキーを押します．Rcmdr が起動します．(step 5)
6. 他の足りない packages もインストールするように促される場合もありますので，その場合は【はい (Y)】を押して，必要な packages も入れるようにして下さい．パッケージのダウンロードが完了すると Rmcdr は使えるようになります．



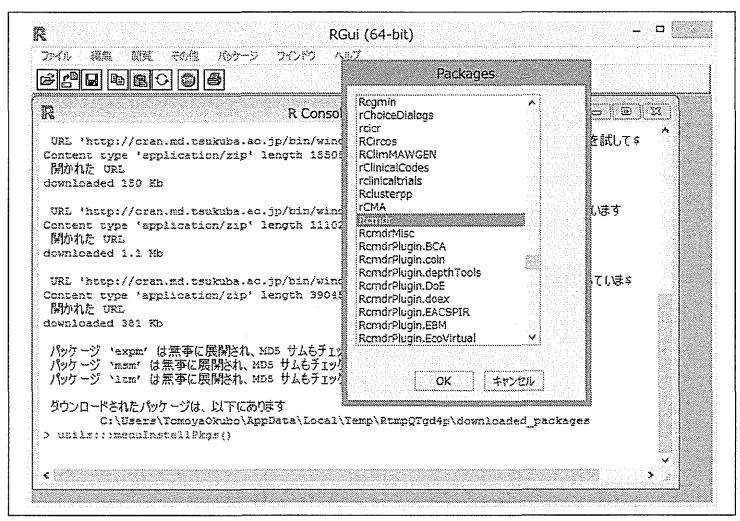
A.6 R の起動 (step 1)



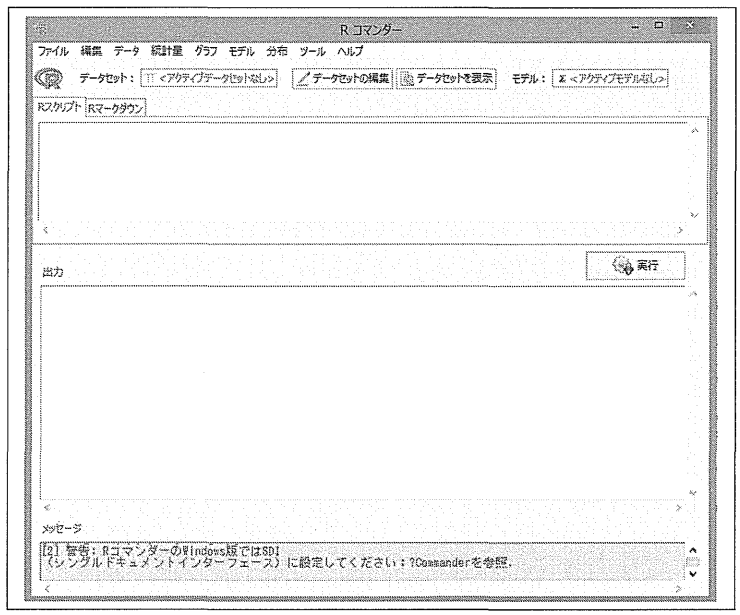
A.7 パッケージのインストール (step 2)



A.8 ミラーサイトの選択 (step 3)



A.9 Rcmdr パッケージの選択 (step 4)



A.10 Rcmdr の起動直後 (step 5)

§ A.3
Rcmdr の起動と終了

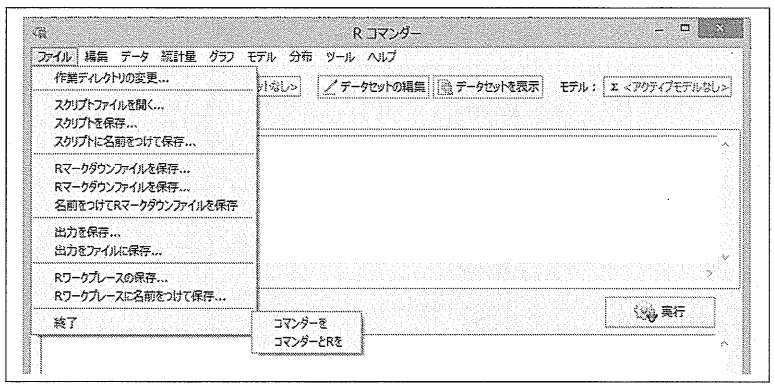
Rcmdr を起動するには，まず，R を起動した上で，

```
> library(Rcmdr)
```

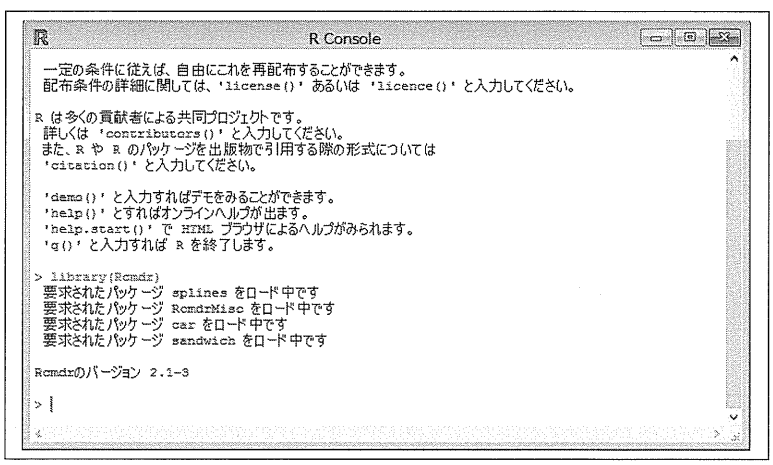
とコンソール画面に打ち込みます.

一方，Rcmdr を終了する場合は以下のようにします.

1. メニューの【ファイル】から【終了】を選択します. (step 1)
2. 【コマンダーを】を選択した場合は，Rcmdr のみが終了し【コマンダーと R を】を選択した場合は，R と Rcmdr が終了します.
3. 【OK】を選択後,【スクリプトファイルを保存？】【アウトプットファイルを保存？】と聞かれるので，それぞれ【Yes】または【No】を選択します.
4. R から再び Rcmdr を再起動する場合は，`Commander()` と R のコンソール画面に打ち込みます. R が起動されます. (step 2)



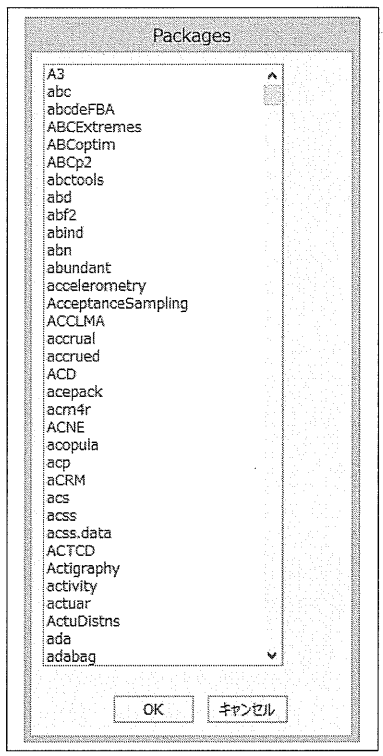
A.11 Rcmdr の終了画面 (step 1)



A.12 R のコンソール画面からの Rcmdr の起動 (step 2)

§ A.4 パッケージのインストール

R において，一連の関数をまとめて読み込めるようにしたものをパッケージと呼びます．必要なパッケージは事前にダウンロードしてインストールしておく必要があります．先ほどの Rcmdr もその「パッケージ」の一つなのです．したがって，パッケージが必要になった場合のインストール方法は Rcmdr のインストール方法を参考に行ってください．



A.13 インストール可能なパッケージ (の一部)

§ A.5 パッケージの読み込み

パッケージはインストールしただけでは使える状態にはなっていません．インストールしたパッケージを利用可能にするためには，

```
> library(MASS)
```

のように R のコンソール画面に打ち込み，呼び出しておく必要があります．このパッケージの読み込み作業は，R を立ち上げる度におこなう必要があります．したがって，通常は後の節「自作関数の読み込み」にあるように，よく使うパッケージの使用宣言を並べたファイルを作成しておき，その作成されたファイルを読み込む作業だけをする (すなわち，そのファイルの中で使用宣言されている library() が使えるようになる) という方法を使います．

この例では，MASS という多数の数学関数が組み込まれているパッケージを呼び出しました．他に，lme4 というインストール済みのパッケージを呼び出すには library(lme4) と R コンソール画面に打ち込む必要があります．

すでにインストールされているパッケージの一覧は

```
> library()
```

で表示されます．したがって，この時に表示されないパッケージは library(****) で呼び出すことが出来ません．

また，すでに呼び出されている (利用可能な) パッケージは

```
> search()
[1] ".GlobalEnv"        "package:MASS"      "package:methods"   "package:stats"
[5] "package:graphics"  "package:utils"     "Autoloads"         "package:base"
```

のように表示することができます.

§ A.6 Rによる単純計算

ここでは，Rを使って実際にいくつかの計算をしてみたいと思います．このテキストの方針として，細かな方法は説明はしません,「習うより慣れよ」「論よりRUN」の精神で進めていきます.

次のようにRコンソールに入力してみて下さい．なお入力するのは"> "となっている行のみです．他の行は「返り値」と言って，Rから返される計算結果です.

```
> 1+1
[1] 2
> 10-3
[1] 7
> 5*6
[1] 30
> 6/2
[1] 3
> 10^2
[1] 100
> sqrt(100)
[1] 10
> 1+1/2
[1] 1.5
```

§ A.7 Rによるオブジェクトへの付値

Rでは，文字変数に数値や文字列を付与することが出来ます．付与するための記号は"<-"です.

Rでは，この"<-"を多用します．"<-"が刺さった文字列をオブジェクトと呼びます．オブジェクトはRを終了させると消えてしまいます．したがって，Rを使うときはテキストエディタを開いて，その上でプログラムを書くように心がけます．実行するときはエディタからコピーしてRコンソールに張り付けるようにします.

```
> A <- 50
> A + 5
[1] 55
> B <- log(100)
> A / B
[1] 10.85736
> A * 30
[1] 1500
> C <- "apple"
> C
[1] "apple"
> A + C
 以下にエラー A + C :  二項演算子の引数が数値ではありません
```

§ A.8
Rにおけるパスの設定

ここでは，データやファイルのやり取りをおこなうパスの設定方法について説明をします．
そもそも，パスとは

"C:/folder1/folder2/data/"

のようにディレクトリ構造を示したものです．ここで，/はフォルダの区切りを表しています．
これまでは，R(Rcmdr) でデータを読み込む際には，

```
read.table("C:/folder1/folder2/data/Smoke01.csv", header=TRUE, sep=",", na.strings="", dec=".")
```

のように指定していました．しかし，毎回このようにデータファイルのある場所まで指定するのは手間がかかります．
そこで，

```
setwd("C:/folder1/folder2/data/")
```

のように一度宣言しておくことで，ファイルのやりとりをする際には () で示したパスを自動的に参照してくれるようになります．R のスクリプト上ではデータの読み書きを頻繁に指定する場合がありますので，あらかじめ，パスを設定しておくことはスクリプトの単純化・効率化につながります．
具体的には，以下のように指定しておきます．

```
setwd("C:/folder1/folder2/data/")
read.table("Smoke01.csv", header=TRUE, sep=",", na.strings="", dec=".")
```

上記のように，データファイルのあるフォルダまでを指定します．
例えば，デスクトップ上に R_統計というフォルダを作成しておき，

```
setwd("C:/Users/ユーザー名/Desktop/R_統計/")
```

と宣言しておき，データはそのデスクトップ上のフォルダに入れておくことが考えられます．

パスの書き方がわからない場合は，一度，Rcmdr でデータの読み込みを行えば，そのときのスクリプトが表示されるのでそれを利用します．

§ A.9
オプション・自作関数の読み込み

R では，オプションによっていくつかの基本的な機能を変更できます．

たとえば，エラーを返さないように設定するには，

```
> options(show.error.messages=FALSE)
```

のようにします．

また，R のコンソール画面で一行に表示できる文字数を変更するには

```
> options(width=40)
```

のように width オプションの値を変更することによって達成されます．なお，R のデフォルト値は width=80 となっています．

他にも，R では，自分で作成しておいたファイルを読み込ますこともできます．

たとえば，

```
> source("C:/R/mysource.txt")
```

のようにすれば，自分で作っておいた関数などがすべて読み込まれます．

先述のように，R を立ち上げるごとに宣言するものは，テキストファイルに用意しておき，そのファイルを読み込ますという方が効率が良いです．

例えば，次のように mysource.txt の中を書いておきます．

```
# library の読み込み
  library(Rcmdr) # R commandar の読み込み
  library(MASS) # MASS package の読み込み

# パスの設定
  setwd("X:/*******/data/")

# 表示桁数の設定
  options(digits=3)
```

R において，#は「#以下の文字列 (ただし，当該行のみ) は読み込まない」というルールがあります．そのルールを利用して，#をコメントを入れる際に使うことができます．上記のファイルでは,「library の読み込み」や「パスの設定」といった文字列がプログラムコードではなく，単なるコメントとして取り扱われています．

このように読み込むのも一つの方法ですが，もっと単純なのは，先ほどの myrource.txt の内容をコピーして，R の

コンソールにペーストしてしまうことです.

§ A.10
課題

課題 1 次のパッケージを各自で使えるようにしてみて下さい.

1. lme4
2. psych
3. polycor

なお, 使えるようになったかの確認は

```
> search()
```

でパッケージが表示されているか見ることによっておこなってください.

課題 2 Rと Rcmdr を一度終了した上で, 再度起動させ, Rcmdr まで使えるように準備してください.

研究計画書作成の説明とまとめ

聖路加国際大学研究センター
臨床疫学センター
高橋　理

研究デザインの3つの要素

- 理論的デザイン：研究テーマ・RQ
- データ収集デザイン：デザイン法
- 解析デザイン：統計手法（後程）

研究計画書の構造

表 1-1　研究プロトコールのアウトライン

構成要素	目的	
研究テーマ	どういうテーマを研究しようとしているか？	臨床知識経験
研究の意義(背景)	なぜその研究テーマが重要か？	理論的デザイン
研究デザイン	どのように研究を実施するか？	
研究期間		
研究のタイプ		
対象者	どのような研究対象者をどのように獲得するか？	
選択基準		臨床疫学
サンプリング方法		データ収集デザイン
観察因子(変数)	どのような因子(変数)を測定するか？	
予測因子		
交絡因子		
アウトカム		
統計学的事項	研究の規模はどれくらいで，データをどのように解析するか？	
仮説		医療統計
サンプルサイズの推定		統計手法デザイン
解析方法		

2015/11/21　3

Hulley et al., *Designing Clinical Research*, 2007.

Type of Clinical Question

臨床現場での課題	例題	EBMの分野
患者臨床情報の解釈・疾患の予測	この患者の症状と兆候を最もよく説明する疾患は？	診断についての研究
疾患の説明・リスク因子	なぜこの患者にこの病気に罹ったのか？	因果関係についての研究 予防についての研究
疾患の経過の予測	治療しない時の病状は？ 治療するとよくなるのか？	予後のついての研究 治療についての研究
医療行為の決断	他の治療法を選択するべきか？	決断分析・費用効果についての研究
医療行為の遂行・経過	その後の経過の頻度は？	観察頻度についての研究

2015/11/21　7

PECO：CQ→RQ

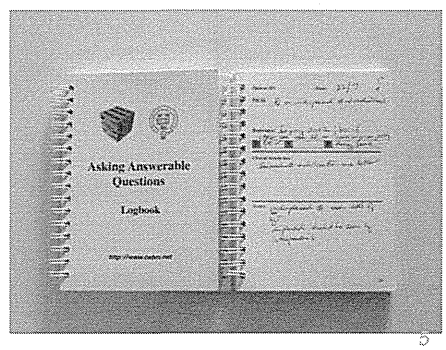
P: patient / population

E/I: exposure/intervention

C: comparison

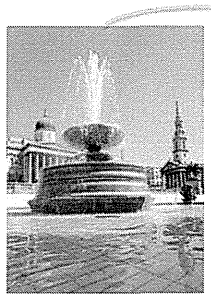
O: outcome

T: Type of question



2015/11/21 5

Idea⇒CQ ⇒ Research Questions



臨床現場

興味・情熱

PECO：臨床的意義/新規性

実行可能性

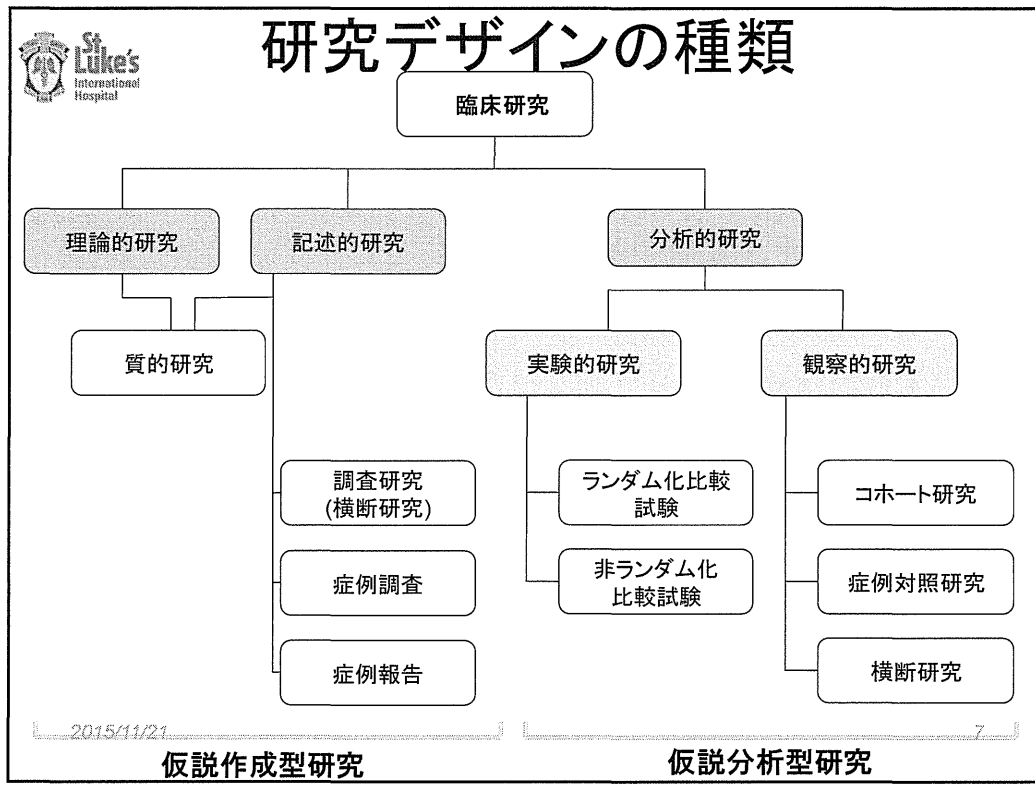
Idea 100

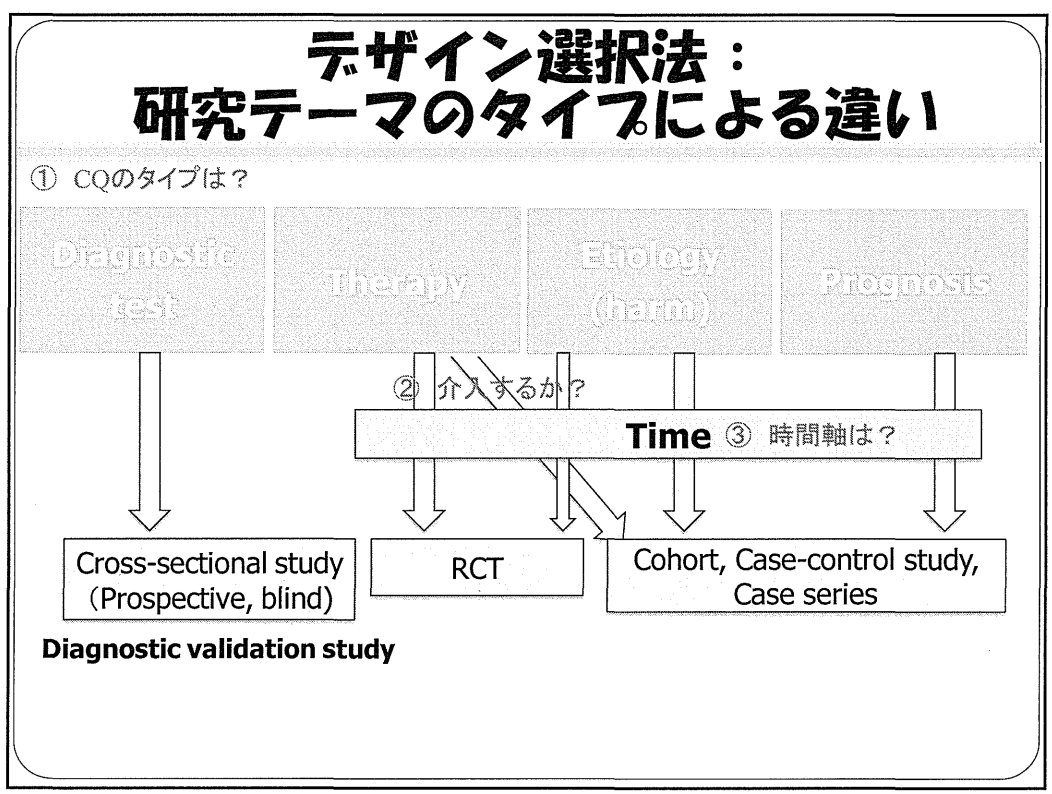
CQ 20

RQ 5

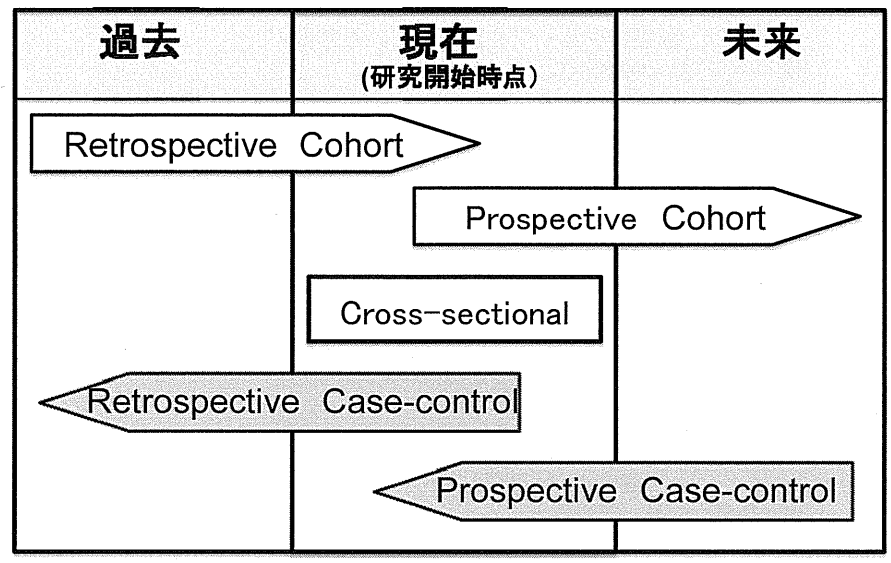
計画書作成開始

2015/11/21


研究デザインの種類
St Luke's International Hospital
臨床研究
理論的研究
記述的研究
分析的研究
質的研究
実験的研究
観察的研究
調査研究
(横断研究)
症例調査
症例報告
ランダム化比較
試験
非ランダム化
比較試験
コホート研究
症例対照研究
横断研究
2015/11/21
7
仮説作成型研究
仮説分析型研究


デザイン選択法：
研究テーマのタイプによる違い
① CQのタイプは？
Diagnostic test
Therapy
Etiology (harm)
Prognosis
② 介入するか？
Time ③ 時間軸は？
Cross-sectional study
(Prospective, blind)
RCT
Cohort, Case-control study,
Case series
Diagnostic validation study

データコレクションの時期



エビデンス・レベル
Evidence Level