　さっき申し上げたとおり、「R」は全部無料です。特に臨床研究をされている方は、SPSSがかなり王道で使っていらっしゃると思いますが、少なくとも基本的なパッケージというか、そもそも解析に耐え得るようなパッケージを買うだけで20万や30万して、何か少し高度な解析をしたいと思えば一気に50万円ほどします。バージョンアップしたりパソコンを買い換えたりしたらまた買い直させられるというような感じで、とにかく高いわけです。

　総額で何百万とつぎ込んでいる研究機関はたくさんあると思いますが、それはもうやめにしたいということで、「R」にしました。「R」は今、もうだいぶ整ってきていて、10年ほど前なら「R」で統計解析したら「あにゃ？」という感じで、〓レギュラー〓にと言われることもあったのですが、今では「R」を使っても当たり前だという流れです。

　やはり欠点はあり、プログラムベースなのが一番の欠点です。ただ、プログラムベースというのは、MRの解析でも、今後GUIというのは一つ遅れていくので、プログラムベースに慣れていかないと研究していく上であまり太刀打ちできなくなってくるかなと思って、こちらをお薦めしているわけです。

　あと、「R」はもう世界中で使われているので、日本語・英語に限らず、大体Googleに聞けば解決します。結構高度な解析でも、「これがスタックした」とGoogleに入れたら解決して、それでも解決しなければ、筆者に言うと答えが返ってくることもあります。そういう意味でも使いやすいです。SPSSでももちろんGoogleで解決しますが、無料で解決できるというのが一番大きいというわけです。

00：15：02

　SPSSをやったことのある方は、こうやってクリックすれば何かやりたいのが見つかるわけです。ただ、これでうまくはいきますけれども、20万30万をこれにかけるのかという話になるわけです。「R」なら、何をしているのかがさっぱり分からなくて、コードを読み取っていかなくてはならなくて、他人の「R」のコードを見ても「これは何だ？」というような感じですが、慣れれば無料でできるわけです。

　では、最初に使ってみようということです。今、「R」をそのまま使っている人というのはそれほどいなくて、大体このRStudioをパッケージとして組み込んでインストールしてやっている人が多いです。これはまだRStudioですね。立ち上げてみましょう。

　皆さんは多分、R-introの所に「./Rintro/seminar script.R」というファイルがあると思います。それをクリックしてもらえば、きちんとインストールした人はRStudioが立ち上がるはずです。もしくは、RStudioをインストールして、ウィンドウズなどでは左上を選択してくれれば起動できます。

　起動したら、このような画面が出てくるはずです。これは見やすいようにやや拡大しています。出てきましたか。出てこない人は手を挙げたほうがいいですよ。もう行きます。

　こんな感じになります。色が違いますが、皆さんは色が白の場合が多いかもしれないです。00：17：01

00：00：00

小池：……いろいろな色に変えられますので、好きな色に変えてあげてください。別に変えなくても大丈夫です。フォントが見えないとか字が大きすぎるとかは、フォントの大きさが変えられるので、変えてみてください。ここまでいいですか。

　あと、事前の作業でまだインストールされていないlibraryがある場合は、最近のRStudioがアップデートされて、こういう感じで「これはインストールしていないぞ」というような「！」が表示されると思います。既にインストールパッケージでインストールしている人はこれが表示されないはずです。

　ただ、Wi-Fi環境がないとインストールできませんので、その場合はWi-Fi環境を備えていただくか、メールを送ったような〓エッジローム〓と「U Tokyo-Guest」で、エッジロームが入って登録されている人はエッジロームが使えますし、「U Tokyo-Guest」は多分、登録すればそのWi-Fi環境につながるはずですので、少し〓帯域〓は狭いですが、できます。ここまでは大丈夫そうですか。いいですか。

　Working directoryというのがあるにはあって、そのフォルダーの中でファイルが動いたり操作したりというのがあります。Working directoryが最初から〓そうなっていれば〓いいのですが、そうなっていないこともあります。大体ドキュメントになっているので、その場合はこの「setwd（“directory path”）」でdirectory pathを打ってくれればいいです。

　それでなれない人は、ここの右下に「More」というのがあるので、ここの「Set As Working Directory」がWorking Directoryになっていればいいです。なっていなければ、そこまで移動してもらって「Set As Working Directory」にすると、僕だったらここの「Directory」の所に移動します。Windowsのコンソールと同じですが、〓（dir）〓でやるとそこのファイルが出てきます。Windowsのコマンドラインを使っていた方は分かると思いますが、（dir）というこのファイルが出てきます。こういうふうに出てこない人はWorking directoryの設定が間違っているので、よろしくお願いします。これですね。

　ここまでで何か困ったことはありますか。ここに書いてありますが、たまに「一体ここはどこだ」というような場合があります。その場合はこれを押してもらうと、Windowsのマップでアベイラブルな形で、こんな感じで分かりやすいようにいつもの画面が表示されます。これで選んでもらっても構わないので、大丈夫です。迷子にならないでください。では、次に行きましょう。

　RStudioの基本ルールです。これも「当たり前だ」と言う人と、「ああ、そうなんだ」と言う人がいると思いますが、大文字と小文字が異なります。僕はSPSSも結構使っていたのですが、大文字と小文字は関係ないのです。恐らくプログラミングは大文字と小文字が異なるのが基本だと思います。そういうことがあります。

　大体は変数を代入します。例えばxやyなどに変数を代入する場合は、昔はこの〓「＜－」〓しかできなかったのですが、今は「＝」でも代入できます。

00：04：56

　ただ、「R」の場合は右側に変数を書いて右側の「＜－」を書くという方法もフレキシブルに取れるので、あまり「＝」はよほどではない限り、この「＜－」のほうが明示的で分かりやすいかと思います。

　基本的にこの左下の画面がこの「R」のもともとの画面になるわけですけれども、改行で実行されます。しかし、（ ）は閉じない限り実行されないのです。どういうことかというと、画面にあるようにやってもらえばいいのですが、例えばここで「1＋1」とやると、「2」と出てくるわけです。ただ、「1×（2＋）」とかは……、（ ）は自動で表示されてしまうのですが、こうすると「あれ？」となるわけです。それで、残りを書いてやると出てくるというかたちになります。

　大体何となく書いていって、この「＋」画面が出てきたらどこかで（ ）が閉じられていないと思ったほうがいいわけです。こちら側の画面で実際はバーッと書いていくのですが、よく見ると、後ろになってくるとこれぐらいならまだいいと思います。しかし、だんだん長くなってくると、「何だ、これ。随分長いぞ、長いぞ」となってくる場合があるわけです。そういう場合はこういう所で適当にリターンキーを押すとここで区切れて実行されないので、そうするといいと思います。大体カンマが後で区切るのが多いです。こんな感じで区切って押してやるときれいに（ ）の所でずれてくれるので、こういう感じでやっていくのがいいと思います。

　スペースは空けても空けなくても大丈夫です。見てもらえば分かりますが、＃を打つとそこから先はコメントになります。プログラムを書いてその後ろ側に＃を打ってこれは何かを明示してもらって構わなくて、だから、この後に＃を打って「これはライブラリーだ」などと書いてくれても動きます。

　これはチップスですが、該当部分をコメントしたい場合は「Ctrl＋Shift＋c」と押すとコメントになります。ここに書いてありますね。例えばこれを選んでもらって、「Ctrl＋Shift＋c」とやると、こんな感じです。こんな感じになるので、取りあえずバーンと打って、これを全部コメントといったときに「Ctrl＋Shift＋c」とやると、その部分が全部コメントになります。

　Saveのこの辺はいいとして、これを実行したいというときにどういうふうにすればいいか。この後、例えばこのdata1というものを呼んでもらうわけですが、これを実行するときに、もちろんこれをコピーしてここに貼り付けて実行してもらってもいいのです。それは無駄なので、Ctrl＋ReturnやAlt＋Returnを押すとできます。

　例えば、Alt＋Returnと押してもらうと、実行されてその場にいます。Ctrl＋Returnだと、実行されて次の行に行きます。だから、普通に書いていく場合は、Ctrl＋Returnをダーッと押して、何か最初のものを実行するというようなことをよくやるわけです。

　Library xlsxは使わないので飛ばしますが、このような感じで実行してみたいものをぼーっと押していくとCtrl＋Return、Ctrl＋Returnというのが出てきます。私はMacを使いませんが、Macの人は違うらしいので一応Macのことが書いてあります。よろしくお願いします。MacはAltがないわけですか。CtrlがCommandになっているのか、よく分かりませんが、そんな感じです。

　ここまでで何か分からない方はいらっしゃいますか。分からない方は適宜手を挙げてください。お願いします。

　では、データファイルを読み込んでみましょう。

00：10：01

　先ほどやったように、ここの行を持っていってCtrl＋Returnを押すと勝手に読み込まれます。読み込まれたものはここに表示されているので、ここに表示されていれば大丈夫です。もちろんこういうものは死んでも選ばないという人は、Fileから読み込みというかたちをすればいいです。

　ここでOpen fileとかいうのは、実はあまり良くなくて、Open Fileは「R」のファイルを読み込むことなので、csvとかxlsxとかSPSSのファイルを読み込むときは、「R」にとってはインポートする感じになって、Import Datasetになります。Import Datasetで今ならSPSSとSASとStataのファイルまで読み込めるらしいです。もちろんtextやExcelももちろん読み込めるというかたちになるので、これをやればいいわけです。

　皆さんにこれでやってもらうとこういうGUI画面が出てきて、Previewが出てきてというふうになります。結局、リターンするとここに何かそういうコマンドが表示されるだけなので、毎回選ぶのがだるいと思った場合は、まずこれで選んでからここに表示されたものをこちらにコピーして実行すればいいのではないかと思います。

　よろしいですか。ファイルは読み込めましたか。読み込めたら、最初にデータをチェックしてみましょう。

　このsummary（x）を実行してもらうとどうなるかということです。やはりidとジェンダーがあって、その後、var1～var30まで謎の変数がバーッと並んでいるという感じです。このhead（x）を実行すると、最初の5列が全部読み込めます。­GUI上でもこれを押せばデータベースは見られます。それほど難しいものではありません。だから、これを見れば確認できるということです。

　ただ、これぐらいの変数ならいいですけれども、多変量で縦が1,000で横が100とかいうデータを使う方がいらっしゃる場合は、これをやると結構重くなってしまうので、僕は結構summary やheadでさっさと確認することもあります。headも5が面倒なら例えば15とかで〓やれば〓、15行分表示されますので、このような感じになります。

　IDEに関しては、実はこのCheat Sheetというのが幾つかあります。実はこの〓HELP〓の所にCheat Sheetというのが隠れていて、これをクリックすると、大体そのRStudioのそもそものガイドラインや、これは何と読むのかいつも分からないのですが、〓Deep Laye〓やggplotといった所のCheat Sheetの表示が出てくるのです。残念ならが英語なので、誰か日本語に訳してほしいのですが、これを見たら何が●場合はどれを押せばいいのかが全部出てきます。RStudioに関してはこのような感じでやってくれればと思います。このHELPのCheat Sheetをダウンロードすると、ネットにつながっていればできます。

　さて、data 1は大変シンプルなデータセットで、僕が適当に作ったものです。50人いて、性差が25：25で、var1～var30までそれぞれ1からランダムに振っただけなので、平均すると50になるわけです。id、ジェンダー、平均が1～100まで50.5になります。そのような感じのものが並んでいます。見れば分かります。そのとおりになっているはずです。

00：15：03

　さて、これで少し慣れてもらおうというわけです。これは先ほどの簡便なデータのチェックですね。昨晩、PDFファイルで送ったスライドと全く同じですから、飛んでいて何が見たいのかが分からなくなったら、そちらもご参照ください。これは先ほどのステップどおりです。

　なぜ差があるのかを検定してみましょう。基本的にはこれを打てば大丈夫です。これですね。最初に取りあえずこれを実行してもらってもいいですか。動きません。大体統計検定をやるとき、「R」の場合は、var1は従属変数なので従属変数を左に置いて右側独立変数を置いて、あとはオプションを置くというのは、もうほとんどのパッケージで決まっています。ここの関数がt.testならt.test、ANOVAならAOVという感じです。

　後のスライドでその一覧表の半分作りかけをご覧になれば分かると思いますが、大体こんな感じで書けるというわけです。これはチルダで結び付けるわけです。ただ、これを実行するとエラーが出るはずです。そんなものはありません。「オブジェクト‘var1’がありません」と出るわけです。

　なぜかというと、「R」はオブジェクトを読み込んで、データを読み込んでいるのですが、これはxのわけです。きちんと住所を書かないと読み込んでくれないのです。例えば「東京都目黒区駒場3－8－1」という住所があったとして……。

00：17：02

00：00：00

小池：「目黒区駒場3－8－1」と書けば大体郵便は届きますが、「R」はその辺の融通が利かなくて、目黒区が東京だということは誰も見てくれないわけです。ですから、そこは平等に「東京都」から書かなければいけないのです。

　そのオブジェクトの中のdatasetの中の変数を選ぶ場合は、$でつなぎます。x$sexと書くと解析できます。これはここにないので、手打ちすれば出てくると思います。

　もう1個は、行列番号を指定するというものがあります。datasetで番地が1から順番に振られると思っていただいて、3列目と2列目を調べるわけです。だから、行の部分は全部なので何も書かずに「,」で打って列の部分だけを3と指定すると、全く同じものが出てきます。“［ ］”でくくると行列番号が指定できます。恐らく多変量の解析をするときは、はっきり言ってこちら側のほうが楽です。いちいち書いているのは大変なので、2番目の指定の方法は最初「何だ、これは」と思ったのですが、慣れてくるとこちらのほうが圧倒的に楽です。

　ベクトルを選びたければ、こういうのはあまり統計解析ではやらないですが、x［1,］、2列目はx［,2］と。1行目2列目の数字を取ってきたスカラーを打った場合は、x［1,2］という言葉で、〓マトランホウ〓とほぼ一緒です。マトランホウの場合は確かカギ括弧やカンマが違ったような気がしますが、同じようなものです。

　あとは、最初のうちはこの「attach」を使うと慣れます。アタッチ関数というのは、番地を無視していいと。xの中の変数を全てオブジェクトとして認めてもらうことですが、それを実行するとx以下の変数は既に独立したオブジェクトとして見てくれるようになります。そうすると、これができます。このattachですけれども、これを実行してもらいます。「detach」を実行するとdetachしてしまうので、これでこれを実行します。こんな感じになります。ここまでに何かおかしなことが起こっている人はいますか。大丈夫ですか。こんな感じで実行できます。

　見れば分かると思いますが、僕はvar.equalがTRUEではないほうを選んだので、welch検定になっているわけです。この辺は等分散性を仮定するとか仮定しないという話で、これは「R」の問題ではなく、統計解析の問題です。違いはまた調べておいてください。等分散性を仮定するデータであれば、var以降がTRUEになります。ジェンダーで男女だから等分散性は仮定されるので、上でも正解ですが、下のほうが検定は緩くなる感じです。そういうかたちになると思います。

　このままdetachしてもらってもいいですし、detachしてもらわないで動かしてもらっても構わないです。detachするとやはり動かなくなります。次に行きます。

　いろいろ検定すると思います。毎回変数を変えてやり続けるとなると大変なわけです。ですから、「for構文」というものを使えば繰り返しで検定を掛け続けてもらえます。

00：04：57

　僕は慣れましたが、for構文は「R」の場合、少し複雑らしく、あまり機能としてはよろしくないようです。上か下のどちらかの方法を使うことによって、今回のvar1～var30のfor構文を組むことができます。

　最初はどういうことかというと、for構文自体はここに書いてありますけれども、これを言うfunctionを作るわけです。こういう機能を作る場合は、t.test.sexという、これは何でもいいのですが、例えば＝f 1、2、3〓とかにして定義してくれればいいです。どういう定義かというと、こういうiという変数を持つ機能を作りますと。ここは一緒で、このiという先ほどの行列を指示します。例えば1が入ったら1を入れる、2を入れたら2を入れる、3を入れたら3を入れるという感じで、t.test.3と入れたらこれが実行されるというような実行関数を作ります。

　「print」だけは囲んでくれないといけないのが少し難点です。囲まないと多分、これが実行されて表示されないだけですが、こんな感じの関数を作ってもらいます。先ほどの変数を見てもらったと思いますが、実際ここに「32変数」と書いてありますね。32変数はなぜかというと、最初にidとsexが入っていたので、3列目～32列目までの変数が1～30まで並んでいるので、それを全部実行したければ、iを3～32まで入れるというのでぱっとやると、これで実行されるというわけです。やってみますか。

　これでパンと定義して、この後にfor構文をやると、皆さん、目にも留まらぬ感じで表示されて、何か変わったのかと思ったけれども、全部ガーッとやられているわけです。変数が30個の間は全部実行されています。こんな感じになります。

　もう1個あります。これが世紀の謎関数のeval（parse）関数という関数ですが、これは一種の呪文なので、意味を捉える必要はあまりないと考えてもらっていいです。このeval（parse（text=paste……と括弧でくくった所で、この括弧の間は何も入れないというふうな指定をすると、ここが実行されます。

　細かい説明を下のほうに書いておきましたので、意味を知りたい人は知ってくれていいです。このprint(t.test(x$var、var1、var2、var3、var4……と入れていくわけですよね。でも、このvar1は文字列なので、文字列の部分だけ数字でどんどん置き換えたい場合に、ここを括弧でくくるわけです。だからここでvar1、var2、var3、var4が入って来るのですが、これがpaste関数というもので文字列として規定されます。

　文字列として規定されるときに、普通は間をカンマで置くとかスペースを入れるとかになるのですが、このsep = ””と打つと、隙間は入れませんと。ここに隙間を入れると隙間ができてしまって変数が解読されなくなるので、隙間なしで設定すると、これに関してiが1～30、var1、var2、var3、var4、var5、var6、var7……と、これで実行できます。

　やってみますか。全く同じことが起こります。スタックしているとか、分けが分からないとか、ここまで付いていけない人はいますか。大丈夫ですか。

　どちらかでこういったことが実行できると思います。ですから、行列番号を指定してもらったほうがいいと思います。直感的にやりたい場合はこちらのほうがお薦めです。一回慣れると簡単です。

　30個も表示されてバババッと並んで、もちろんプログラムを書けばこの数字を全部抽出してきて表にするなどは簡単にできるのですが、取りあえずざっと見たい場合にはこれなら毎回画面を見たら絶対に見落としが発生するわけです。

00：10：03

　ですから、sink関数を使うといいと思います。sink関数というのは、出力結果の保存先を指定するわけです。デフォルトは〓コンソール〓になっているのであれですが、ここでファイルを指定します。ファイルの名前を書き換えても大丈夫ですが、これを実行します。その後、もう一回これを実行すると、何もエラーも出ずにただ点々が出ます。

　実際に中身を見てみましょう。ここに何かファイルができています。これは皆さんのWorking directoryです。テキストで表示されているわけです。全部できています。これは別にずっとやる限りは永遠にやられてしまいます。先ほど1＋1をやってもらったと思いますが、答えが表示されなくなります。どこへ行ったかというと、ここのテキストファイルに多分表示されています。これです。今、表示されていますね。こんな感じで、プリントされる関数が全部テキストに保管されてしまいます。

　さて、問題ですが、出力されたファイルを見て、var1～var30のどこに男女差があるのか検討してみましょう。どうぞ。

　1は違う。ここです。2は違う、3は違う、4は違う、5は違う、6は違う。惜しい。あった。ver14です。実はこのver14だけ、〓ドンブリカンバン〓で。そういうことではないですよね。やっている人はいますか。この辺は●と関係ないですが、これは大学の授業用に使っているものなので、Multiple testingを体感してもらう感じでやってもらっています。

　統計検定というのは、繰り返すと有意なものが必ず見つかるというものです。これは20変数ですが、30もやれば有意水準を満たすものがランダムでも発生してしまうことを示しているわけです。ですから、MRIの研究は、〓BonferroniやFWER〓、FDRを使うわけですが、こういったことがあるのでこれはやってはいけないというわけです。よろしいでしょうか。

　では、次の回に行きますが、休憩はなくてもいいですか。ゆっくりやっていきますので、適宜トイレとかは行っていただいて構わないです。

　さて、もう少し臨床研究に即したdatasetがあったほうが慣れると思いますので、data2を読み込んでいただければと思います。あと、sink関数は解除してもらったほうがいいです。標準化されないということが確実に起こります。いちいちテキストファイルを見てもしょうがないので、これは解除しておいてください。ここからスタートですね。54～55行目ぐらいからスタートです。

　どんな質問紙かというのは、添付のワードファイルに入れました。しかし、今回は「だから何だ」という話があるので、●でやられているかもしれませんが、イメージを膨らませるためにご覧になっていただければと思います。実際このデータは、とある調査でやったデータに関して、それぞれランダムに－1とか0とか1とかで完全に架空のデータになっているので、皆さんはご自由にお使いいただければと思います。

　どんなデータなのかをサマリーで見てみます。230個のidがあって、ageは16歳～28歳ぐらいということです。ジェンダーがあって、iqがあって、－1が〓そこに〓入ります。WHO5というwellbeingの尺度があって、GHO12という、うつの尺度があって、SQという尺度があって0から……。これも－1が入ります。SSQのAというのは、助けを求めたときに助けてくれそうな人数を主観的に答えるもので、その人たちに対する満足度をBで答えるというふうにもってくるものです。0、1、2というのがあって、0年後、1年後、2年後という縦断研究です。

　説明に書いてありますが、blankをviewで見てみます。見てみると、幾つかが「NA」と表示されていますね。要は、縦断研究をしていると1年後はドロップアウトする人がいるわけです。ドロップアウトした場合は、blankとして〓キヨ〓されています。質問紙の「あるある」ですが、5個も10個も見ていれば答えをしないとか、〓2つ〓回答とか、よく分からない……。

00：17：02

00：00：00

小池：……があるのですが、それは－1になっているというわけです。こういうdatasetになります。

　欠損値が2つあります。headで見てくれればこんな感じになります。〓－5〓で、10の人は分からない。－1もあればNAもあります。

　RStudioではこの欠損値をまず処理しなければいけません。－1は欠損値だというのを、答えが下に書いてありますが、どうしてこのようなことになるかというと、上のほうに説明があります。

　比較演算子というのがあって、これはプログラムを書いている人は慣れていると思いますけれども、「＝」の場合と「一緒だ」という意味が違うので、こういうふうな感じで「＝＝」を書きます。sexが2の行を抜き出す。これを打ってくれれば分かるのですが、これが2の行を選んでくださいとなると、このような感じで2の●がバーッと出てきます。列に関しては指定していないので、全ての変数が出てきます。

　だから、逆にこのようにしてしまうと、多分2しか出てこないのではないか。出てこないです。すみません。こんな感じになります。〓＆条件〓などは＆で結んで、or条件だと〓縦の傍線〓で結び付けることができます。

　今回の全て前の中の－1があるものを全て抜き出せというのは、こういうふうな感じで、コンマを打たないです。コンマを打たないと、とてもデータがスカラーのものを全部代入するかたちになります。だから、これをやればいいと思うのですが、これだけ実行するとNAはこのような感じで山ほど出てきます。－1が増えています。NAもここに入ってしまうわけです。本当は－1だけ抜きたいのですが、NAも出てきてしまうわけです。これをNAとするということです。

　こう実行してもらってもう一回サマリーをしてもらうと、どうですか。前にあったWHO5やGHQ、iqの－1がミニマムというのがなくなっています。これで欠損値の処理が完了したというわけです。

　だから皆さん、臨床研究のdatasetをもらうときに大体欠損値が空白だったり、〓9999〓になったりとか、いろいろあると思います。そういう場合に、最初にこれを処理しておかないと「R」は分かってくれないので、そこだけ最初は注意です。その都度サマリーでデータがちゃんと整理できているかを確認したほうがいいです。そういうところがあります。

　何となくシートをもらって大体解析して、飛び地があって、グラフも作った段階で間違いに気付くといったことがありますが、簡単に打てるので取りあえずこれでざっと打つほうがいいと思います。いいでしょうか。

　お作法が結構あるので困るのですが、「R」の場合は変数の定義、論理値や文字列、整数、実数、複素数、順序なし因子、ベクトル、行列、データフレームやリストなどがあります。これが実は重要で、こちらまでは行列で、ほとんどベクトルで指定されていて、こちら側はベクトルのx、yは〓集合したら行列〓になるし、データフレームになるといろいろな型がはめられる。例えば、整数と順序なし因子で、ファクターが交じっているというのができるというわけです。

00：05：05

　大体皆さんは、この辺の整数や実数、順序なし因子しか使わない。あとは論理値と文字列、複素数以外この辺しか気にしなくていいと思いますが、少しやってみましょう。factorを少しやってみましょう。factorは後で説明しますが、実は鬼門なのです。

　例えば、性別は別に1と2ではないですね。こうすると、今のところyは整数になっていますね。〓int〓というのは全部integerなので、整数ですね。だから、●を読み込むときに全部整数だと判別されて、年齢も性別も全部整数だと思われているのですが、性別は整数ではなくてfactorです。間はありません。間はありませんと言うと怒られるかもしれないですが、統計上、間はなく、取りあえず間がない場合はas.factorとやると、このyのsexという変数をfactorにした上でyにsexを戻しますと。これだけ実行●です。

　「あるある」ですが、これを実行するとコンソールに共有されるわけで、レベル1になっていて、確かになっているけれども、これは実際問題、変数は関係ないので●。こうすると格納されたわけです。

　yのsexという変数は、factorですかという質問をする場合はisを使いますけれども、TRUEですね。「実数ですか」と言うと、「いや、違います」になります。「何ですか」と聞きたいから〓class〓。それでサマリーをもう一回やってみましょう。きれいにfactorを変えましょう。男が139人で女が〓95〓人です。このまま行くと、そもそもこれは男と女の比率が出てきていますが、何人かを正確に求めるには計算しないと出てこないというような使えない統計になってしまうので、こんな感じで出てきます。元に戻しておきます。元に戻っていないです。これは実数になっています。そうするとまた……。ここまでで付いてこられない方はいますか。大丈夫ですか。

　「R」を使ったことがある人は、大体このfactor変数に悩まされます。このfactor変数は少しポンコツなのです。どういうことかというと……、すみませんが、これをもう一回実行してもらっていいですか。直前にダウンロードしてくれた方はこれが入っていますが、昨日これをここに入れたほうが早いだろうというので入れてしまったので、ない場合はこれだけ実行してください。

　これだけを実行すると、もう一回yに、元のcsvの変数が戻るので、－1がミッシングの状態になっています。先ほどこれをNAにしてしまいました。NAにしたらこのトリックが証明できません。

　このssqb\_0というのがあって、これはもともと－1が入っているのですが、0、1、2、3、4、5、6という7段階になっているので、確かにカテゴリーとして見てもいいだろうと思います。精神疾患業界だと、特に自記式の質問紙でうつが4点以上をポジティブにするという〓0点に割ったり〓します。連続変数として解析するか、カテゴリーから解析するかというときに、factor変数を必ず使いますが、まずこれのプロフィールを見てみましょう。

　table関数というかsummary関数に似ていて、どういう変数があるかというのを並べてくれるわけです。－1が3点、1が3点、2が17点。別にこれは実数でもできるので、例えばiqのような連続変数でもできないことはないです。たくさんありますが、連続変数だとこうなります。小数点まで表示されているものはこれをやるとバーとなりますが、これでもまだ何とか見られます。〓素数分布〓を見るわけです。

00：10：07

　これに対してfactorをするとどうなるかというと、－1というものが3人、1というものが3人、2というものが17人、3というものが22人になります。当たり前ですが、全く同じ答えです。ただ、これを実数に戻すとどうなるかというと、test2というものに戻しましたと。それでtableをやってみると、おかしい、1、2、3、4、5、6、7となるわけです。

　どうしてかというと、このfactor因子というのは、これらの数字が少ないものからアルファベット順などに並べて順番どおりにfactor1、factor2、factor3、factor4、factor5と並べていくのです。ですから、－1はfactor1、1はfactor2で、2はfactor3でという感じで、数字だからといってこれはfactor 0にしようとか、factor xにしようとか、そんなことは言ってくれないわけです。ですから、factorを繰り返しまた実数に戻すと数字がおかしくなります。

　これは大問題で、前から大問題だと言っています。世界中の人が大問題と言っていて、世界中の人が諦めているので、諦めています。残念です。ですから、これだけ注意していただければと思います。もう一回読み込みますが、この場合は注意してくださいということです。

　どうしてこれほどギャーギャー言うかというと、read.csvやread.何とかとかは結構factorだと思ってfactorとして勝手に読み込むというふざけた性質があるのです。こちらで勝手に指定させてくれればいいのに、例えばヘルシーコントロールをhcとしていて、●はscとしていて、hc、hc、sc、sc、hc、hc、とバーッと並んでいるものは、文字列としてではなくてfactorとして読み込まれます。それを便利だと思う人もいるかもしれないですが、大きなお世話だと思う人もいるわけです。ここまで何か分からない人はいますか。大丈夫ですか。

　そのような感じで、read.csvというのはfactorと読み込んでしまう場合があるので、最初にここの変数を読み込むときにはviewだけで見るのはあれなので、ここの型もきちんとチェックしたほうがいいです。今回は、どうしてかは知らないけれども、全部実数が読み込まれていますが、そうではないときも多々あります。

　変数として、絶対factorとして読み込んでほしくない場合は、このstringsAsFacotrsという、なぜか大文字と小文字が乱立している指定●があり、これをくくってfalseとすると、必ずfactorとして読み込まれずに、そういう文字列として読み込まれたり、数値として読み込まれたりします。

　あとは、最初からデータ型を全て指定するという方法があります。このyという変数なら、idはidなので、カテゴリーでも何でもなくてただの文字列です。1、2、3、4、5、6、7、8という数字ですが、意味としては文字列なので、このcolClassesというのを打ってもらいます。cというのは行列を作る意味なので、character、最初は文字列だよ、次は数字だよ、factorは性別だよと。そこから先は全部連続変数なので、連続変数が15個並んでいますという感じです。repというのはリピートの意味なので、これを実行するとこういう行列ができます。

　例えば、これだけなら同じですが、こうするとこういう行列ができます。文字列の行列ができるので、これを全部打ってもらってcolClassesとやってくれてもいいですが、そのようなアホなことはしないです。

00：14：58

　こうすると……、何かおかしいな。データは18列ありますね。これでいいですね。こんな感じで読み込むと、よく分かっているdatasetに関してはこちらが確実です。

　少しやってみますか。実際問題、こういう「2年後のうつ症状という点数は、ベースラインと1年後のうつ症状のどちらから影響を受けているでしょうか」。本当はこういう解析をあまりしませんが、お試しでやってみようというわけです。lmというのは、一般線形モデルの関数です。重回帰分析をして、2年後のうつの症状が従属変数で、独立変数はどちらかということでこんな感じで＋で書き、data＝yと書けばできます。

　答えが書いてありますが、これだけあっても本当は出てこないです。

00：17：02

00：00：00

小池：……これでは不足しているので、関数が高度になればなるほど、大体summaryでググると欲しいデータが結構出てきます。これが欲しいですね。ほかは要らないです。

　そうするとどういうような答えが出てくるかというと、切片が1.5で、ghq12\_0が0.36というのが係数で、1は係数がこれで、0のほうが予測は大きくなったような感じです。突き詰めるとめちゃくちゃですから、あまり言わないでください。突き詰めると、そもそもghqの1年後は0年後から多大な影響を受けているので、本当にこれを示したければ、●を持ってこなければいけないのですが、これとこれが完全に独立だと思うならこうなるわけです。だから本当はWHOとかのほうがいいかもしれないですね。

　2年後のうつの症状は、ベースラインのうつの症状に影響しているのか、それともベースラインの活力、Wellbeing、生きていて幸せ、どちらでしょうか。完全な活動データではないので、それなりに反映すると思います。出ました。全然違います。やはりうつの人がうつなのは、Wellbeingは関係ないです。逆はいいです。逆は出ますか。

　逆は違う結果ですね。だから、やはり両方とも2年後にハッピーな人というのは、ベースラインのうつの症状が少なくて、かつ、ベースラインの時からハッピーだったようです。精神科業界では知られているのかどうかは知りませんが、僕の中ではGHQとWHOは似たような手法であっても結構相違が見られると思います。

　こうやって書いてもいいですし、こういうものは全部オブジェクトとして見ることができるので、この計算結果をまずresult1という所のオブジェクトに格納して、それをsummaryで見るという二段構えで表示することもできます。こちらのほうがお薦めです。なぜかというと、result1を見るときに、resultというのは〓ビッグオブジェクト〓で、いろいろな感じで抜き取られていくわけです。

　ですから、例えばresut1でどういうものが入っているかというと、〓因子〓係数から残差からeffectも●。とにかく何か知っているものを入れて〓ください〓。coefficientとやって出てくると、こんな感じで数値だけ出すことができます。そうすると、例えばresult1、result2、resutl3、result4、result5という感じでバーッとやって、このcoefficientだけ行列として並び替えたい場合は、ちょこちょことプログラムを書けばこれの表が簡単に作れるわけです。ここを何とするかが●けれども、こちらのほうがお薦めです。

　ここまで何か質問のある人はいますか。大丈夫ですか。だんだん疲れてきているので、そろそろ終わりにしたいと思います。

　主な統計解析手法はたくさんあります。こういうことをやっていたので、この辺は何が何かというのは全部分かりますが、皆さんは全て詳細に説明することはできますか。できなくて大丈夫だと思います。どうして名前がこれほど違うのかというのは、歴史の問題なので、名前が何でこうなのかというのはあまり記憶にしなくてもいいです。

　結局、こういう表を見れば、おのずと統計検定方法が出てくるわけです。皆さんはカイ二乗検定を知っていますか。t検定も知っていますか。この辺になると●。ANOVAは見たことがある。ANOVA●、repeated-measures ANOVA、ロジスティック、ロジスティックといろいろ言っていますが、結局、SPSSでやるときも何となく答えが出てきてしまうからみんな選ぶと。SPSSだと、これにこれを入れると動かないとか、ここには名義変数を入れますとかが出てるので間違えないですが、こういう表を見ておけば何を調べればいいのかが分かるわけです。

00：05：01

　例えばどんなことかというと、男女の間で抑うつの得点を知りたいときは何の統計をしますか。これはt検定ですよね。どうしてかというと、男女というのはBinaryで2変数、抑うつというのは連続変数なので、独立変数が名義変数2つで反復測定がなくて従属変数が連続変数でparametricなら、t検定になります。こんな感じで決まっているわけです。それ以外の方法はありません。

　BinaryでBinaryなら、これはカイ二乗検定です。ありなしと男女で比べているので、2×2のカイ二乗検定になるわけです。年齢層は3カテゴリー以上でcontinuousなら、これはANOVAと決まっているわけです。〓連続変数〓と連続変数の関係を知りたければ、これはPearson’s●「R」だと、こういう感じで出てきます。continuous、continuousだからPearson’sと覚えてもらってもいいですが、実際問題〓どういうふうに〓使ったらどう●というのがいちいち決まるわけです。

　回答のほうは、SPSSに慣れている人に関しては、SPSSとの対応表があったほうがいいだろうと、僕自身が思ったので、こつこつ作ったものがあります。一部は埋まっていないのですが、こういうものがあるので、もし興味のある人は「これはこうだ」とか、「これはこう書いたほうがいい」とか、「これはどうするの？」というのを教えてくれれば、Rに関しては僕がお答えできると思います。大体何があればいいかが書いてあるので分かりやすいです。

　あと、後半は臨床研究のデータの整理というのでもう一回説明します。皆さんはExcelで処理するときに、count\_ifやsum\_ifやvlookupなどを使って、100×100ぐらいのフェーズになって、Excelの場合は10メガぐらいなっていて開くのに10秒ほどかかるそうです。結構大変ですよね。それで、大体間違える。データ整理やExcelでやっても可視化できていていいですけれども、変数が途中でコピーできていないとか、隣のものをコピーしていると思っていたら1個ずれていたとか、大体間違えるので、可能であればこういうプログラムベースでやったほうが確実です。

　ただ、count\_ifとかvlookupなどに慣れている人はどうすればいいかというのは、幾つかこの後に紹介しますが、「R」で作っている人も昔はSPSSなどExcelを使っていて、こういう変数とか〓関数〓があったらいいなというのを勝手に開発するのです。Libraryの〓SPSS〓とはまさにそうで、同じようなことを入力すれば同じようなことができるという楽なパッケージが幾つかあります。それに関して慣れてくれれば、ここはプログラムベースで行けるということになります。

　ここまでに何か質問はありますか。最後の休憩の後にこのEZRを紹介したいと思いますので、RStudioはまた立ち上げるのでこのまま置いておいてもらってEZRを立ち上げてみてください。EZRは結構立ち上げに時間がかかります。

　作った人は不思議な人で、自治医科大学の血液内科の先生なのです。訳が分からないです。なぜか統計ソフトをGUIベースで全部無料で作ってしまったという変わった人です。お会いしたこともないですが、お礼を言っておいてください。

　EZRに関してはもう書籍が幾つか出ているので、そちらを見てもらったほうがいいと思います。SPSS〓ライト〓で統計解析ができます。データのインポートでこれを選んでもらって、datasetというファイルになっていますが、これでOKとしてもらうと、さっきのデータを読み込む前に……。これを表示すると、もう読み込めているのが分かります。すみません、大きくできないので皆さんの画面でやってください。

　一部は実行できないものがあります。

00：10：02

　例えば先ほどのdatasetで－1ならNAだというのは、探したけれどもなくて、こういうふうに書いてもらって「実行」とすると、実行されて表示すると全部NAになっています。

　統計解析をしたい場合、例えば先ほどのt検定……。「目的変数は1つ選択」、当たり前です。この辺が多分GUIの便利なところで、比較する部分のデータで、〓両側〓で「はい」で、棒グラフを出せとやります。これでできました。差はなしです。差はなしというのは、精神科業界ではおかしいです。女の人のほうがghqでは高いというところがあります。こんな感じで誰にでもできます。SPSSを買うお金がなくてもプログラムは死んでも書きたくないという人にはこれをお薦めします。

　●はどこにあるのかな。あった。〓線型●〓、これですね。目的変数はghqにして、説明変数には今後はSSQも入れてみましょう。ghqだと●。これで〓トウセイ＝が通ればいいわけですね。できました。GHQとWHOは説明して、SSQは説明しそうだけれども説明しませんでした。こんな感じで出てきます。

　あとは、このscriptの所に何をやったかが全部表示されているので、そういうものを見るとRの勉強になるというか、これがメインになります。これは適宜遊んで好き勝手やっていただければいいと思います。

　休憩の前にお話ししましたが、統計解析の手法はいろいろあります。研究〓スタッフ〓の皆さんは分かると思いますが、統計に掛けるまでがとても時間がかかるわけです。一説には統計に掛ける前に9割の時間が費やされていて、統計を掛けるのが1割で、それで結果が出なくて惨敗するということを皆さんも理解してやっていらっしゃると思います。

　解析に移る前に前処理というものを必ずしなければいけないです。これはMRIの研究でも何でもそうですが、いきなり解析できるような研究はありません。幾つかやりましが、やってないものとしては、縦に長いデータです。ダーッとたくさんあるデータだと重複している場合が取り方によっては発生するので、2回入力してしまったりします。そういうのがあるので駄目だというのと、欠損値や外れ値の処理です。あとは正規性や離散傾向、カテゴライズするのか、連続変数●しなければいけないということです。

　その後にデータの意味付けをしなければいけないわけです。変数の定義や変数の作成をして、機械学習のときは正規化をします。あとは、MRの研究でも横に長いデータは次元削減して、PCAなりICAなりを掛けないといけないです。そうしてデータをきれいにした上で、datasetを解析datasetに組み換える必要があるかどうかは分からないですが、一部組み換える必要があります。こういうデータサイエンティストみたいな仕事をしなければいけないわけです。

　今回、前処理については、R Markdownでも可視化を行います。こちらのほうが圧倒的にリポートとして使えるのでいいのですが、R Markdownは毎回〓手軽に〓●できるわけではないので、うまく使い分けていただいて、データを可視化しつつ処理してもらえればいいと思います。

　先ほどと同じdatasetを使います。datasetの可視化は、先ほどから示しているようにsummaryやheadやtableを使うとか、午後にするR Markdownを使ってきれいにリポーティングするとかがあります。あとは、散布図や相関行列をしたいという場合があります。このままの状態で行きますけれども、プリパレーションのほうでこれを実行してもらえばいいと思います。

　散布図や相関行列というのは、SPSSで相関行列はバーッと出てくるけれども、〓グラフ〓はあまり出てこないです。こちらだと結構簡単に出てくるわけです。ここを見れば誰でも分かります。2列目から6列目にしたというのは、もう見れば分かるとおり、全部にすると「バラララ……」となってしまうので、分かりやすいものだけ●。

　どうですか。これでざっと見る限り、年齢とかはあまり関係なさそうですね。全部〓Pearson‘s R〓なので、この辺はもう少し目視で確認した後は、きちんと●しなければいけません。

　WHOとGHQは先ほど性質が違うと言っていましたが、〓プロセクショナル〓で相関を取ると－0.6という感じで、結構相関係数としては高いわけです。ただ、●は違うというのは先ほど例でもお示ししたとおりです。

　正規性は担保されていますか。●は担保されているような、されていないような感じです。ジェンダーはカテゴライズしなければいけないので、そもそも正規性はないです。iqも正規性はないと言ってもいいような気もしないでもないですが、正規性の検定をすると「ある」ということになります。WHOはありそうですね。これはきれいです。GHQ12というのは、その性質上ありません。これは正規性が担保できないので、検定を行う人は正規性をやらないような解析をしなければいけないというわけです。こんな感じです。

　先ほどのものでも、これでも十分分かるのですが、こういうきれいなパッケージがあります。

00：05：02

　これはもう皆さんにインストールしてもらっているので、このLibraryから行きます。警告は無視してください。バージョンがまだアップデートできていないです。

　〓このGGペアーズ〓というのを〓Gギャリー〓のLibraryの中で使うと、少し時間はかかります。全部の編集をやる必要はなかったです。これは〓「ア」〓と書いていますが、〓NコルY〓というのはyの変数のコラムだから、列の数を調べているわけです。これは結構有効です。どうしてかというと、いちいちここを見て32とか打っていると、変数を増やせば33になるかたちでバーンとやると連続性という感じになるので、大体この3からスタートして最後までやりたければ、3からNコルYとやるとこのYの列数は●できます。きれいな画像が出てきます。

　見方はいろいろありますが、男女で分けて、男女の分布があって、それぞれバーッとありますね。今回のdatasetはあまり男女の差がないですよね。男女ごとに分布が●なっていて、この辺はすごく分かりづらいですけれども、こちらが分布で、こちらが相関の分布になっていて、ここに〓コジレーション〓のPearson’s Rの値が出てきます。男と女でそれぞれ値が違うということをお示ししています。こんな感じですね。これでは少し分かりづらいので、ほとんど3でない所が〓いいですかね〓。

　これは分かりやすいです。数字が小さいですね。このような感じになっていて、これでdatasetの目星を付けることができるわけです。男女差がある所はありますか。この辺はありそうですね。この点だと少し分かりづらいのですが、もう少し連続値の場合はこの辺が分かりやすいかと思います。どうしてもこの辺の正規性が全部保たれていないので、点の打ち方がおかしくなっています。正規性が保たれていればとてもきれいになります。

　あと、これに近いものがあります。〓コンレイムズ〓とやると、こうやって変数の名前が出てきます。いろいろとやりたいと思ったことは大体Googleに聞けば全て答えが返ってくるので、こういうことをやりたいと思えば、できる・できないと考える前にGoogleに「何々したいな」と書けば、大体答えが返ってきます。

　summary toolsです。summary toolsの〓ds〓summaryはとても速いですね。こんな感じで自動的にどういう変数かというのをある程度●もらって、カテゴライズできそうならこういうふうな感じで●作ってもらって、連続変数〓互換〓から、2チャンネルみたいですが、テキストでグラフを作ってくれて、大体こんな感じで分布がある、ミッシングはこのぐらいという感じで、表にしてくれます。

　これももちろんsummaryと〓ヘンドル〓とtableとかを駆使してやってもらうのもいいかと思いますが、こんな感じでざっと見て、取りあえずそのあとにこの辺はこれでというのをやっていきたいという人は、こういうものを使ってもいいかなと思います。

　もう一個、論文を書くときにメインなのはtableoneというRパッケージがあります。tableoneで最初からお作法としてここに全部このまま貼り付けてもいいのですが、長いからここにまとめてあります。

00：10：08

　こういう変数定義をして、このCreate Table Oneというtable1というオブジェクトに持ってきます。別にこのバーの右側に全部書いてもいいですが、結構みんなはtableでこれを表示したい、これは表示したくないと、少しモディファイします。このtableのバーの中に、factorを●指定してあげます。それで、strataはジェンダーとやって、〓data y〓と、こんな感じでやると……。table1をこうやってすると……、すみません、これもこちらで落とします。

　table1はこれですね。こちらが切れてしまっていますが、多分皆さんのものはフォントが小さいので全部横側にうつると思います。このような感じでtable1はできています。これをExcelで開いてもらうと、あとは少し変型するだけで表は完成です。Excelでいちいち手打ちしないほうがいいと思います。大体小数点を間違えます。この1と2がずれてしまっていますが、これはご愛嬌（あいきょう）で、こんな感じです。

　これで統合できそうです。このまま統合していいかどうかは教えませんが、こんな感じです。これは簡単です。Create Table Oneパッケージは日本人が作りました。これです。

　もう一個あります。ここまで何か質問のある人はいますか。これはただの紹介です。少し汚くなってきたので、これを一回全部消します。消す場合はrmというのを使います。lsとやると、これまでに使ったオブジェクトのリストが出てきます。FactoVarsとかいろいろと読み込みましたね。これを消す場合は、rmを使って、リストの右側にそのオブジェクトの名前を入れればいいです。オブジェクトの名前はこれで承認させるので、これで全部消えます。消えました。なくなりました。

　次は、このsource(“y.pretreatment.R”)をこのまま実行していただいていいですかね。そうすると、このyがここに表示されています。これでもう一回●と。これは先ほどと全く同じです。この(“y.pretreatment.R”)というのは何を開いているかというと、ただこれが書いてあるだけなのです。これは先ほど説明したジェンダーだけをfactorにしようということでここを丸ごとコピー＆ペーストしてきたのです。

　前処理のときというのは、大体皆さん前処理で、datasetできれいにバーッと作るというところで取りあえず仕事が1つ終わりますよね。そこから後にいろいろやりたいというと統計とはまた別の作業になるので、1つのRのファイルをずっと500行や1,000行書いていると、一体どこまでが前処理で一体どこから統計解析とか、自分の〓お試し〓した変数の処理なのかが分からなくなります。だから、ここまで確定となればそのRファイルは一回閉じてもらって保存しておいてください。最初から最後までバーッと動けば前処理が動くようにきれいに整えておいたら、そのsource関数でそのファイルを最初から最後まで実行すればいいのです。

00：14：58

　そうすると、最初に統計解析ソフトのRの〓バイ〓の所にこのsourceというのを書いておけば、そこから自分のやりたい統計やさらにやりたい処理といったものができます。そうやって完全にファイルを分けていくことが非常に大事です。そうしないと、500行や1,000行などになって、ここまで実行したらリターンキーが壊れるまでリターンキーを押し続けるとか、そんなばかなことになります。

　ここに〓rm〓とかsourceとかがあり、このsourceというボタンを押すとこのファイルが最初から最後までバーッと実行されます。だから押しません。こちらは押していいです。もう一回やります。これを取りあえず消して、ここでsourceと押すと全部実行されます。同じです。こんな感じです。こちらでsourceボタンを押した人はそこからずっと動いていますが、待っていてください。

　データの確認からデータを意味付けしなければなりません。要は、iqもそうですけれども、最初に上げたうつの指標というのは自記式のうつの不安の尺度で、最低点が0点で最高点が12点です。「あるある」の問題として、被験者によって何でも「はい、はい」と付けて12点満点の人がいるわけです。一生懸命考えて、やはりうつのようだと思って4～5点の人がいるわけです。

　では、その4～5点の人と何でも「はい、はい」と付けて12点の人はうつの重症度が3倍かというとそんなばかなことはなくて、むしろ4～5点の人が重症であったりするわけです。

00：17：01

00：00：00

小池：……自記式の質問紙の〓業界〓というのは結構カテゴライズするわけです。4点以上をうつ病のスクリーニング対象とします、それ以外はスクリーニング対象外とします、というような感じで、カテゴライズしていることが多いです。ほかの業界でも恐らく同じようなことが起こっているだろうというわけです。

　先ほど例に挙げたSPSSというのはExcel SPSSなのですが、ExcelやSPSSで便利だったコードをそのまま「R」でも実行できるようなパッケージングです。読み込むと結構いろいろ入っているわけです。どういうふうにしたいかというと、例えばSPSSならrecode関数をご存じの人はいますか。みんな知らない？　ヤバい（笑）。

　例えばこういう0～19歳を1として、20～49歳を2として、50～99歳を3として、それ以外は欠損値とするというようなコードを書くと、そのとおりに1、2、3と全部置き換えてくれるのです。これだとageの情報が消えてしまうので、agecategoryというふうに持っていきたい場合は、recode age(-1=sysmis)という感じで、into agecategoryと打って、最後にピリオドを打ちます。これを実行すればSPSSならagecategoryという新しい変数が簡単に作れます。

　こういうものが同じようにできないかというので、「R」も普通のパッケージでは結構3～4行書かなければいけないし、％が多く入って少し気持ち悪いですが、このように1行でできるというわけです。

　例えば、ここが1でここが0の人は、抑うつが1で不安も1の人は「うつ重症」というカテゴリーを新しく作りたい場合は、cout\_ifやsum\_if、あとは〓ウイルカフ〓とかそういうことをやっていくわけですが、そういうのもこんな感じでできるのです。幾つかあります。例えばこれなら、全く同じもので0～100、IQが100以下の人を0にして、100以上の人を1にする。これでもできるし、この〓ifエー〓スタンスは普通の「R」の中に入っていますが、「R」ではIQが100未満であれば0、それ以外は1というExcelのIF関数みたいなこともできるし、ほかにもいろいろあるわけです。やってみますか。

　こんな感じで表示されます。どういうことかというとhi IQが0とコーディングされた人は〓50〓 人いて、100未満の人は12人しかいなくて、100以上の人は216人いるというのがきちんと表示されているのが分かります。やり方は無限大にあるので、慣れた方法でとにかくカテゴリー化していくとか、●の処理をしていくことが大事なわけです。

　次に行きます。〓144と199〓だと言いましたが、最後、変数を処理した後に解析データセットを作らなければいけないです。言っていることの意味が分からないという話があるかもしれないですが、特に縦断研究をする場合にリピートの●、ああいうことをするときは、ずっと横に並んでいるのは解析できないのです。縦持ちに変えなければいけないです。横持ち、縦持ちは何のことだという話ですが、こういうことです。

　皆さんがデータとして保管しているかたちは、横持ちのことが多いです。大体の臨床研究のデータは最初にIDが書いてあって、そこから変数を打っていきます。では、1年後はどうするかというと、まさか縦に書かないで、そこのIDの人に対して横に書いて〓ミエル〓などとは、Excelナンバーの列、caxとかめちゃくちゃな列数になっている表があります。このままでは実は縦断研究のデータを解析できないので、こういう感じで縦持ちの保存形式にしなければいけません。

00：05：03

　どういうことかというと、例えばWHO-5というのは0時点と1時点の2時点ありますよね。IDが1の人というのはWHO-5の0時点、1時点、2時点とか、01……、こんな感じで保管しなければ、SPSSも「R」も各〓測定〓の解析ができないというのはあります。ですから、これにしなければならないわけです。

　実は、これも便利なファイルがあって、先ほど言ったように、dplyr、まだ読み方が分からないのですが、こういうパッケージングがあります。このCheat Sheetを見ると、縦持ちと横持ちに変換することが書いてあるわけです。これをこのとおりに実行すれば、縦持ちのファイルと横持ちのファイルに置き換えることができます。

　まずは、全てのデータがあると困るので、こういうふうにシュリンクするというわけです。これをシュリンクしてy2というdatasetを作る。id、age、ジェンダー、iq、WHO5\_0、1、2という感じで切り取ることができます。Excelではいちいち消さなければいけないので、いちいち読み込むのは大変です。しかし、「R」に持ってきた場合は、「y2」と今は名前を書いていますが、これまで整理したdatasetのオブジェクトとそこからさらにシュリンクさせたオブジェクトは名前を変えれば簡単に切ったり貼り付けたりできます。

　もちろん縦に長いほうを、その中で解析データセットを抽出したい場合も、こちら側の行で条件を付けると縦に関しても省けます。こんな感じで、まず解析データセットを作るというわけです。

　このcheat sheetを見ていただいて、gatherというファイルを使うと横持ちが縦持ちになるわけです。横持ちのy2というファイルをtimeという関数で縦持ちにして、その名前を新しくWHO-5にします。実際、そのtimeというのはどんなファイルがあるか、5～7行目にあるというので、〓y35〓を作ると。こうすると、timeはここの01になって、WHO-5はこういう感じになります。これでようやく開始ができるということです。

　先ほどの表を見たい場合は、orderという並び替えの関数で並び替えると表が出てくるので、0、1、2、0、1、2とidで並び替えます。解析上はこのファイルで解析しようが、このまま使おうが、解析はできますが、チェックするときは並び替えたほうが早いと思います。

　なぜかこのdplyrとtidyrというのは日本語のCheat Sheetがあって、Googleで「Cheat Sheet」で調べれば日本語版が出てくるので、ぜひこれを活用ください。

　もちろん先ほど言った、このデータだけ抽出するとか、変数に置き換えるというのも、実はこの中でもそれなりにできますので、見ながらやってください。どんな方法でもいいと思いますが、やっていただければと思います。

　これが最後のシートです。皆さんが経験している多くの研究データは、「縦持ちのデータですか、横持ちのデータですか」「横持ちのデータ多いです」というのは実は間違っていて、どちらでもないのです。めちゃくちゃです。僕も共同研究でデータをもらうことがありますけれども、大体はめちゃくちゃです。途中で入力方法が変わったり、途中で欠損値が発生したり、途中で欠損値の入力方法が変わっていたり、途中で大文字が小文字になっていたり、「あるある」は数字が全角になっていたりします。それで読み込めないものばかりです。

00：10：00

　どういう方法を使ってもらってもいいですが、自分のdatasetだけではなくて他人のdatasetもどれだけ素早く可視化して、どれだけ間違いをチェックするかという作業は最初に絶対にやらなければいけないです。そういった所でもこういう「R」を使うとそのまま統計に直結するので〓トラップ〓だというところがあります。

　方法はいろいろあると思いますが、これは一例です。皆さんが便利なように組み換えていってもらって、そのまま統計のほうで間違いを〓公表〓しないというような方法を作っていただければと思います。

00：10：45