卒論作成に必要なRのスキルのまとめ

一橋大学経済学部 　高久玲音

2022/4/1

昨年の第1期生は約半数程度がRをそこそこ使いこなして卒論を書いてた一方で、問題設定は良くとも詳細な分析までたどり着けない卒論もあった。卒論に必要だと思われるRのコーディングスキルはそれほど難易度的に高くないものも多いが、Rは経済学以外でも使われているため、経済学部の学生が卒論でよく使うコードがまとめられた文献も少ない印象。

そこで、高久先生が昨年度実施した大規模なアンケート調査を実際に解析してみて、卒論執筆までにどのようなコードが書ければ良いのかおさらいしてみよう。調査は2021年8月に約400万円をかけて実施されたweb調査で約3万人を対象に行われた。調査票は別途配っているので確認し、様々な分析を試してみましょう。

・vdata1.csv 8月6日から数日間行われた第一回調査。22000人程度

・vdata1.csv 8月20日前後に行われた追加調査。8000人程度

なお、調査はワクチン摂取の勧奨が65歳以上に対して優先されていたことに着目し、65歳の前後3歳の方を対象としています。実際の解析は65歳での摂取率のジャンプを利用した回帰不連続デザインですが、その他にもワクチンの摂取についての有用な情報が含まれていると思いますので、適宜解析すると良いでしょう。

データを読み込みます。

knitr::opts\_chunk$set(echo = TRUE)  
  
#install.packages("ggplot2")  
library(ggplot2)

## Warning: パッケージ 'ggplot2' はバージョン 4.1.3 の R の下で造られました

#install.packages("tidyverse")  
library(tidyverse)

## Warning: パッケージ 'tidyverse' はバージョン 4.1.3 の R の下で造られました

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.2 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1  
## v purrr 0.3.4

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

setwd("C:/Users/takak/Dropbox/lecture/2022/zemi/Vaccine")  
  
vdata1 <- read.csv("vdata1.csv")  
  
#データの先頭行をみてみましょう  
head(vdata1[1:10])

## no group sc1\_1 sc1\_2 sc1\_3 sc1\_4 sc1\_5 sc2\_1\_1 sc2\_1\_2 sc2\_2\_1  
## 1 1 1 1 1 0 0 0 1955 3 1967  
## 2 2 1 1 2 0 0 0 1959 8 1969  
## 3 3 1 1 0 1 0 0 1960 2 1960  
## 4 4 1 1 0 0 0 0 1954 9 1951  
## 5 5 1 1 0 0 0 0 1958 3 1963  
## 6 6 1 1 0 0 0 0 1954 8 1952

最近はパイプ%>%を用いてデータの受け渡しを行うのが主流となっています。 パイプはmagrittrパッケージの機能ですが、tidyverseと同時にインストールされています。データ前処理にはdplyr（読み：ディープライアール）というパッケージの関数を主に用います。 dplyrはtidyverseの一部なので、tidyverseがインストールされていればOKです。なお、MACは Control + Shift + m で %>% を出力できます. Windowsは Ctrl + Shift + m で %>% を出力できます（ショートカット）.覚えておくと便利です。

#データの先頭行をパイプを用いてみてみましょう  
  
vdata1[1:10] %>% head()

## no group sc1\_1 sc1\_2 sc1\_3 sc1\_4 sc1\_5 sc2\_1\_1 sc2\_1\_2 sc2\_2\_1  
## 1 1 1 1 1 0 0 0 1955 3 1967  
## 2 2 1 1 2 0 0 0 1959 8 1969  
## 3 3 1 1 0 1 0 0 1960 2 1960  
## 4 4 1 1 0 0 0 0 1954 9 1951  
## 5 5 1 1 0 0 0 0 1958 3 1963  
## 6 6 1 1 0 0 0 0 1954 8 1952

この結果は、前のChapterで見たhead(saving)と同じ結果です。 パイプを用いると、パイプの前のオブジェクトをパイプの後の関数の引数としてくれます。

パイプを用いた書き方は以下のとおりです。

・使用するオブジェクトを示す: vdata1

・パイプでつなぐ: %>%

・使用する関数を書く: head() mean() table()

別の例を見てみましょう。sc2\_1\_1は調査対象者の生年、sc2\_1\_2は生まれ月です。以下の例で確認してください。

#変数の平均を出力  
vdata1$sc2\_1\_1 %>% mean()

## [1] 1956.989

#変数の値の分布を出力  
vdata1$sc2\_1\_1 %>% table()

## .  
## 1954 1955 1956 1957 1958 1959 1960   
## 2089 3403 3469 3530 3980 4276 1253

vdata1$sc2\_1\_2 %>% table()

## .  
## 1 2 3 4 5 6 7 8 9 10 11 12   
## 2412 1913 2067 1698 1632 1533 1823 1893 1761 1777 1711 1780

★練習問題

1．パイプを用いて100の平方根を求めてください

2．パイプを用いて調査対象者の生年の中央値を求めてください

100 %>% sqrt()

## [1] 10

vdata1$sc2\_1\_1 %>% median()

## [1] 1957

変数の作成について。 まずは回答者が女性である場合に１を取る2値変数を作ってみましょう。変数の作成にはmutate()という関数を用います。また2つのカテゴリ（男性or 女性）しかない場合にはif\_elase関数を組み合わせて用いると変数が作成できます。sc3 = 2なら女性、１なら男性です。

vdata1 <- vdata1 %>% mutate(Fem = if\_else(sc3 == 2,1,0))  
mean(vdata1$Fem)

## [1] 0.2390455

次にワクチンの接種状況に関する変数（1.2回以上摂取、2. 1回以上摂取）を作成しましょう。本人の摂取状況の質問はq6\_2\_1で回答は以下です。

1 ２回接種した

2 １回接種しており、次の接種の予定も決まっている

3 １回接種したが、次の接種の予定が決まっていない

4 １回も接種していないが、最初の接種の予定は決まっている

5 接種券を受け取ったが１回も接種しておらず、接種の予定も決まっていない

6 １回も接種しておらず、接種の予定も決まっていない

7 あてはまるものはない

#最初に回答の分布を確認しましょう  
  
vdata1$q6\_2\_1 %>% table()

## .  
## 1 2 3 4 5 6 7   
## 13790 4106 78 1524 2249 155 98

#回答が7の場合には摂取状況が不明なのでNAに変換しましょう  
  
vdata1 <- vdata1 %>% mutate(q6\_2\_1\_r = ifelse(q6\_2\_1 == 7,NA,q6\_2\_1))  
vdata1$q6\_2\_1\_r %>% table()

## .  
## 1 2 3 4 5 6   
## 13790 4106 78 1524 2249 155

#2回以上摂取  
vdata1 <- vdata1 %>% mutate(TwoDose = ifelse(q6\_2\_1\_r == 1 ,1,0))  
vdata1$TwoDose %>% table()

## .  
## 0 1   
## 8112 13790

#欠損を変換した場合にはq6\_2\_1 ==   
  
#1回以上摂取  
vdata1 <- vdata1 %>% mutate(OneDose = ifelse(q6\_2\_1\_r <= 3,1,0))  
vdata1$Dose %>% table()

## < table of extent 0 >

★練習問題 ・大卒以上である場合に１、そうでない場合に０をとる2値変数をCollegeという変数名で、配偶者についてはSCollegeという変数名で作成してください。 学生の質問はq3\_1という変数で以下の定義です

1 中学校／卒業・修了

2 中学校／中退

3 私立高校／卒業・修了

4 私立高校／中退

5 国立・公立高校／卒業・修了

6 国立・公立高校／中退

7 専門学校／卒業・修了

8 専門学校／中退

9 短大・高専／卒業・修了

10 短大・高専／中退

11 私立大学／卒業・修了

12 私立大学／中退

13 国立大学／卒業・修了

14 国立大学／中退

15 公立大学（県立や市立など）／卒業・修了

16 公立大学（県立や市立など）／中退

17 大学院／卒業・修了

18 大学院／中退

vdata1 <- vdata1 %>% mutate(College = ifelse(q3\_1 == 11 | q3\_1 == 13 | q3\_1 == 15 | q3\_1 >= 17,1,0))  
vdata1$College %>% mean()

## [1] 0.5589545

vdata1 <- vdata1 %>% mutate(SCollege = ifelse(q3\_2 == 11 | q3\_2 == 13 | q3\_2 == 15 | q3\_2 >= 17,1,0))  
vdata1$SCollege %>% mean()

## [1] 0.3195455

★練習問題 世帯所得を調査した変数はq25で定義は以下です。「答えたくない」「分からない」という回答を除いて、中位所得以上の世帯が１となるような2値変数（MIncome）を作成してください。

1 世帯の収入はない

2 ５０万円未満

3 ５０万円以上１００万円未満

4 １００万円以上２００万円未満

5 ２００万円以上３００万円未満

6 ３００万円以上４００万円未満

7 ４００万円以上５００万円未満

8 ５００万円以上６００万円未満

9 ６００万円以上７００万円未満

10 ７００万円以上８００万円未満

11 ８００万円以上９００万円未満

12 ９００万円以上１，０００万円未満

13 １，０００万円以上１，２００万円未満

14 １，２００万円以上１，４００万円未満

15 １，４００万円以上１，６００万円未満

16 １，６００万円以上１，８００万円未満

17 １，８００万円以上２，０００万円未満

18 ２，０００万円以上

19 答えたくない

20 分からない

vdata1 <- vdata1 %>% mutate(MIncome = ifelse(q25 >= 19, NA, ifelse(q25 >= median(q25),1,0)))  
vdata1$MIncome %>% mean(,na.rm = TRUE)

## [1] 0.3625594

#vdata1$q25 < 19の範囲で計算されていることを確認  
mean(vdata1$MIncome[vdata1$q25 < 19])

## [1] 0.3625594

median(vdata1$q25[vdata1$q25 < 19])

## [1] 7

2つのカテゴリにしたい場合は上記のif\_elseが良いですが、3つ以上にしたい場合はcase\_whenが便利です。 case\_whenはcase\_when(条件A ~ 条件が成り立つ場合の値, 条件B ~ 条件が成り立つ場合の値…と書いていきます。

q13\_1は「２０２１年６月～現在において、下記の時間は、１日あたり（平均）どれくらいでしたか。／仕事や勉強でのＶＤＴ作業」という質問に対する回答で以下のカテゴリから選びます

1 なし（０時間）

2 １日あたり３０分未満

3 １日あたり３０分程度

4 １日あたり１時間

5 １日あたり２時間

6 １日あたり３時間

7 １日あたり４～５時間

8 １日あたり６～７時間

9 １日あたり８～９時間

10 １日あたり１０～１１時間

11 １日あたり１２時間以上

12 わからない

このままでは解析に使えませんので、分単位でおおよその投入時間を割り当ててみたいと思います。割り当てルールは以下のようにしましょう。

1 0 2 15 3 30 4 60 5 120 6 180 7 270 8 390 9 450 10 630 11 720 12 NA

vdata1 <- vdata1 %>% mutate(q13\_1\_r = case\_when(q13\_1 == 1 ~ 0,  
 q13\_1 == 2 ~ 15,  
 q13\_1 == 3 ~ 30,  
 q13\_1 == 4 ~ 60,  
 q13\_1 == 5 ~ 120,  
 q13\_1 == 6 ~ 180,  
 q13\_1 == 7 ~ 270,  
 q13\_1 == 8 ~ 390,  
 q13\_1 == 9 ~ 450,  
 q13\_1 == 10 ~ 630,  
 q13\_1 == 11 ~ 720)  
 )  
  
vdata1$q13\_1\_r %>% table()

## .  
## 0 15 30 60 120 180 270 390 450 630 720   
## 11222 1163 679 1173 1105 1274 2126 1459 596 76 35

#欠損値（q13\_1 == 12）は無視して平均を算出  
vdata1$q13\_1\_r %>% mean( ,na.rm = TRUE)

## [1] 93.47738

q13\_1からq13\_9はすべて生活時間の質問で回答形式は同じです。q13\_1で行ったような変換作業を行うにはどうしたらいいでしょうか？変数名はq13\_1やq13\_9のように末尾の数字のみが変ることに着目してforを用いて、変換前の変数名（e.g., q13\_2）と変換後の変数名(e.g., q13\_2\_r)を作成します。変数名はpasteという関数をよく用います。paste(“q13”,“5”, sep = "\_") であればq13\_5という変数名を指します。printという関数を用いて変数名を出力してみましょう

for (n in 1:9){  
 var\_name = paste("q13",n, sep = "\_")   
 print(var\_name)  
}

## [1] "q13\_1"  
## [1] "q13\_2"  
## [1] "q13\_3"  
## [1] "q13\_4"  
## [1] "q13\_5"  
## [1] "q13\_6"  
## [1] "q13\_7"  
## [1] "q13\_8"  
## [1] "q13\_9"

paste(“q13”,1:9,sep = "\_")でq13\_1からq13\_9までの変数リストを作成し、末尾にrがつくように元の変数を加工します。q13\_1のケースであれば「.」にこの変数名が代入される形で計算が繰り返されます

vdata1 <- vdata1 %>% mutate\_at(c(paste("q13",1:9,sep = "\_")),  
 funs(r = case\_when(. == 1 ~ 0,  
 . == 2 ~ 15,  
 . == 3 ~ 30,  
 . == 4 ~ 60,  
 . == 5 ~ 120,  
 . == 6 ~ 180,  
 . == 7 ~ 270,  
 . == 8 ~ 390,  
 . == 9 ~ 450,  
 . == 10 ~ 630,  
 . == 11 ~ 720)))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

#q13\_9\_rを確認  
vdata1$q13\_9\_r %>% table()

## .  
## 0 15 30 60 120 180 270 390 450 630 720   
## 19365 640 315 340 202 96 67 35 16 7 17

vdata1$q13\_9 %>% table()

## .  
## 1 2 3 4 5 6 7 8 9 10 11 12   
## 19365 640 315 340 202 96 67 35 16 7 17 900

★練習問題 ・q21\_1からq21\_6は「Kessler Psychological Distress Scale（K6）」という鬱病のスクリーニングテストです。過去1か月間に「神経過敏に感じましたか」「絶望的だと感じましたか」「そわそわ，落ち着かなく感じましたか」「気分が沈み込んで，何が起こっても気が晴れないように感じましたか」「何をするのも骨折りだと感じましたか」「自分は価値のない人間だと感じましたか」という6項目の質問に対して、「まったくない」（0点）「少しだけ」（1点）「ときどき」（2点）「たいてい」（3点）「いつも」（4点）で回答。合計点数が9点以上の場合、心の健康が崩れている可能性が高いとされます。

ただこの調査では調査対象者は以下のように回答しています。

1 いつも

2 たいてい

3 ときどき

4 少しだけ

5 まったくない

このままでは合計スコアを算出できないので、「いつも」が「4」点、「まったくない」が「0」点になるように変数を加工してください。合計点をK6Scoreという変数名で作成し平均を計算してください。

vdata1 <- vdata1 %>% mutate\_at(c(paste("q21",1:6,sep = "\_")),  
 funs(r = case\_when(. == 1 ~ 4,  
 . == 2 ~ 3,  
 . == 3 ~ 2,  
 . == 4 ~ 1,  
 . == 5 ~ 0,)))  
  
vdata1 <- vdata1 %>% mutate(K6Score = q21\_1\_r + q21\_2\_r + q21\_3\_r + q21\_4\_r + q21\_5\_r + q21\_6\_r)  
vdata1$K6Score %>% mean()

## [1] 3.992182

元の数字にマイナス１を書けて５を足せば同じ変換をしたことになるので、以下のように書いてもいいでしょう。

vdata1 <- vdata1 %>% mutate\_at(c(paste("q21",1:6,sep = "\_")),  
 funs(r = .\*(-1) + 5))  
  
vdata1 <- vdata1 %>% mutate(K6Score = q21\_1\_r + q21\_2\_r + q21\_3\_r + q21\_4\_r + q21\_5\_r + q21\_6\_r)  
vdata1$K6Score %>% mean()

## [1] 3.992182

★練習問題 q27\_1からq27\_6は他者への信頼に関する質問です。「あなたの地域の人は機会があればあなたを利用しようとしている」といった意見に対して以下の中から回答を選びます。

1 完全にそう思う

2 そう思う

3 どちらかといえばそう思う

4 どちらともいえない

5 どちらかといえばそう思わない

6 そう思わない

7 全くそう思わない

つまり、他者を信頼する人ほど数字が大きくなるように回答しています。しかしq27\_5だけ「あなたの地域の人は基本的には正直だ」という意見で、他者を信頼しない人ほど数字が大きくなるように回答しています。q27\_5の質問のみ、1から7の回答を逆にしたあとに、すべの回答を足し合わせて信頼スコア（TrustScore）を作成してください。

vdata1 <- vdata1 %>% mutate(q27\_5\_r = q27\_5\*-1 + 8)  
vdata1 <- vdata1 %>% mutate(TrustScore = q27\_1 + q27\_2 +q27\_3 + q27\_4 + q27\_5\_r + q27\_6)

群別の属性の確認

群別の属性（e.g., 男女別のK6スコアの平均）はtapplyを用いて以下のように書くことができます。

tapply(vdata1$K6Score, vdata1$Fem, mean)

## 0 1   
## 3.805687 4.585853

パイプを用いて同じ演算をすると以下のようになります。

vdata1 %>% group\_by(Fem) %>% summarize(mean\_K6Score = mean(K6Score,na.rm = TRUE))

## # A tibble: 2 x 2  
## Fem mean\_K6Score  
## <dbl> <dbl>  
## 1 0 3.81  
## 2 1 4.59

男女別生年別のK6スコアも以下のように計算できます。

vdata1 %>% group\_by(Fem,sc2\_1\_1) %>% summarize(mean\_K6Score = mean(K6Score,na.rm = TRUE))

## `summarise()` has grouped output by 'Fem'. You can override using the `.groups` argument.

## # A tibble: 14 x 3  
## # Groups: Fem [2]  
## Fem sc2\_1\_1 mean\_K6Score  
## <dbl> <int> <dbl>  
## 1 0 1954 3.59  
## 2 0 1955 3.62  
## 3 0 1956 3.66  
## 4 0 1957 3.77  
## 5 0 1958 3.92  
## 6 0 1959 4.05  
## 7 0 1960 4.04  
## 8 1 1954 4.21  
## 9 1 1955 4.81  
## 10 1 1956 4.51  
## 11 1 1957 4.51  
## 12 1 1958 4.50  
## 13 1 1959 4.74  
## 14 1 1960 4.78

★練習問題 q10は「新型コロナウイルスのワクチンの有効性については多くの研究が行われています。１００人がファイザーやモデルナのワクチンを２回接種した場合、そのうち何人が新型コロナウイルス感染症の重症化の危険から守られると思いますか。／人」という質問に対する回答です。正解は95％です。

・男女、成年別に回答の平均値を求めてください。

・男女、学歴別に回答の平均値を求めてください。

・都道府県別に回答の平均値を求めてください。

vdata1$q10\_1 %>% table()

## .  
## 0 1 2 3 4 5 6 7 8 9 10 12 13 15 18 20   
## 549 295 154 75 13 207 11 14 35 40 368 2 2 19 2 101   
## 23 25 30 33 35 37 38 39 40 45 50 51 52 55 59 60   
## 1 14 179 10 8 1 3 3 58 5 1164 1 1 8 2 466   
## 62 64 65 66 67 68 69 70 74 75 76 77 78 79 80 81   
## 2 5 53 6 2 6 2 1184 1 231 4 3 12 2 2681 1   
## 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97   
## 4 3 3 470 11 5 47 15 4964 22 72 163 237 3813 255 321   
## 98 99 100   
## 966 1200 1423

vdata1 %>% group\_by(Fem,sc2\_1\_1) %>% summarize(mean\_prob = mean(q10\_1,na.rm = TRUE))

## `summarise()` has grouped output by 'Fem'. You can override using the `.groups` argument.

## # A tibble: 14 x 3  
## # Groups: Fem [2]  
## Fem sc2\_1\_1 mean\_prob  
## <dbl> <int> <dbl>  
## 1 0 1954 80.1  
## 2 0 1955 81.9  
## 3 0 1956 79.9  
## 4 0 1957 80.4  
## 5 0 1958 79.2  
## 6 0 1959 79.3  
## 7 0 1960 77.7  
## 8 1 1954 78.9  
## 9 1 1955 77.0  
## 10 1 1956 77.5  
## 11 1 1957 76.0  
## 12 1 1958 75.5  
## 13 1 1959 75.8  
## 14 1 1960 74.2

vdata1 %>% group\_by(Fem,College) %>% summarize(mean\_prob = mean(q10\_1,na.rm = TRUE))

## `summarise()` has grouped output by 'Fem'. You can override using the `.groups` argument.

## # A tibble: 4 x 3  
## # Groups: Fem [2]  
## Fem College mean\_prob  
## <dbl> <dbl> <dbl>  
## 1 0 0 76.2  
## 2 0 1 82.0  
## 3 1 0 75.1  
## 4 1 1 79.6

tapply(vdata1$q10\_1, vdata1$sc4, mean)

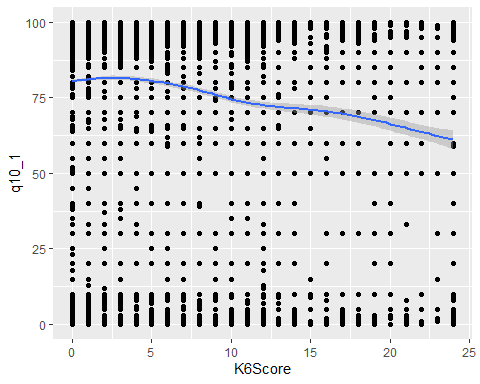
## 1 2 3 4 5 6 7 8   
## 78.37182 74.26144 79.17308 76.88757 81.28571 79.29801 78.20000 78.34271   
## 9 10 11 12 13 14 15 16   
## 79.51575 80.03759 79.20775 79.12214 80.06538 80.57640 77.29769 81.16129   
## 17 18 19 20 21 22 23 24   
## 78.70807 77.50893 82.61458 75.78233 78.19883 77.10369 78.52739 77.76552   
## 25 26 27 28 29 30 31 32   
## 79.81659 79.42828 79.33599 79.78348 80.46104 77.55769 78.76923 81.03409   
## 33 34 35 36 37 38 39 40   
## 80.60959 78.55604 77.74725 81.25843 81.57931 75.93500 78.46154 78.20544   
## 41 42 43 44 45 46 47   
## 73.77381 81.91176 81.55556 75.53704 74.88095 79.03922 71.50000

Data Visualization

K6スコアが高い人ほどワクチンの有効性をしんじなくなるのか散布図で確認してみましょう。

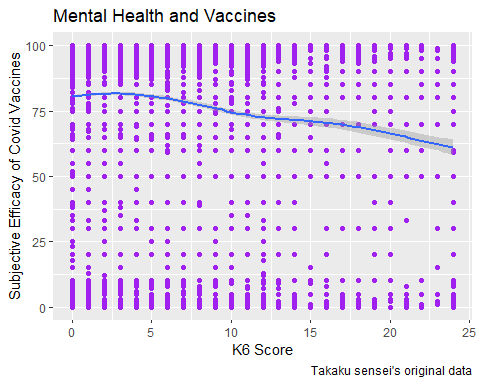
p <- ggplot(data = vdata1, mapping = aes(x = K6Score, y =q10\_1))  
p + geom\_point() + geom\_smooth()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



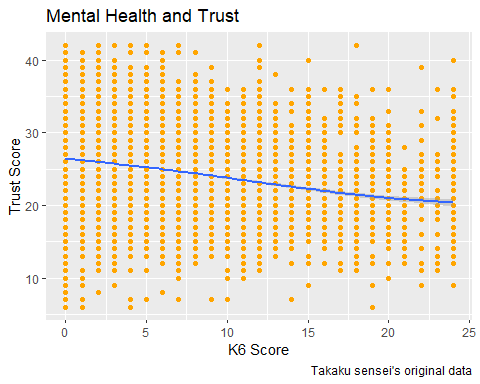
#ラベルなどを着けて加工  
p <- ggplot(data = vdata1, mapping = aes(x = K6Score, y =q10\_1))  
p + geom\_point(color="purple") + geom\_smooth() +  
 labs(x = "K6 Score", y = "Subjective Efficacy of Covid Vaccines",  
 title = "Mental Health and Vaccines", caption = "Takaku sensei's original data")

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



p <- ggplot(data = vdata1, mapping = aes(x = K6Score, y =TrustScore))  
p + geom\_point(color="orange") + geom\_smooth()+  
 labs(x = "K6 Score", y = "Trust Score",  
 title = "Mental Health and Trust", caption = "Takaku sensei's original data")

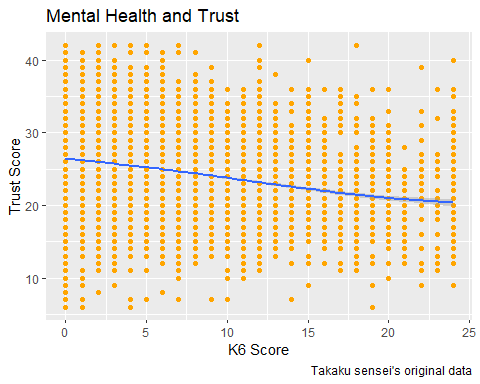
## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



作成した図表を保存してみましょう。図の保存はggsaveというコマンドで以下のように行うことができます。

library(ggplot2)  
library(tidyverse)  
  
p <- ggplot(data = vdata1, mapping = aes(x = K6Score, y =TrustScore))  
p + geom\_point(color="orange") + geom\_smooth()+  
 labs(x = "K6 Score", y = "Trust Score",  
 title = "Mental Health and Trust", caption = "Takaku sensei's original data")

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



#setwdの場所に保存する場合  
#ggsave(filename = "my\_figure.pdf")  
  
#指定したフォルダの下にfigureというフォルダがあってそこに保存したい場合にはhereというパッケージをインストールして使うと便利  
install.packages("here", repos = "https://cran.r-project.org/")

## パッケージを 'C:/Users/takak/OneDrive/Documents/R/win-library/4.1' 中にインストールします   
## ('lib' が指定されていないため)

## package 'here' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\takak\AppData\Local\Temp\RtmpcfAuZF\downloaded\_packages

library(here)

## Warning: パッケージ 'here' はバージョン 4.1.3 の R の下で造られました

## here() starts at C:/Users/takak/Dropbox/lecture/2022/zemi/Vaccine

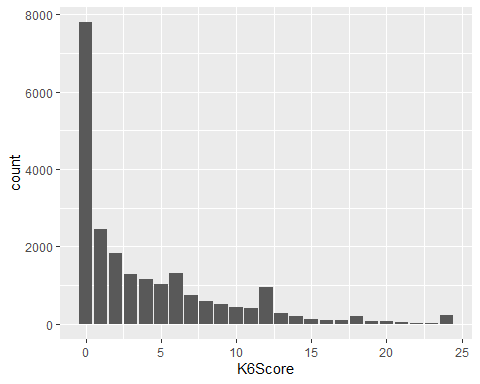
ggsave(here("figure", "my\_figure.pdf"))

## Saving 5 x 4 in image  
## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

棒グラフとヒストグラム

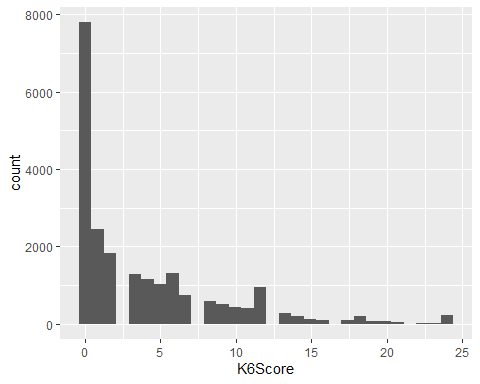
K6スコアの分布を棒グラフとヒストグラムで書いてみよう。二つのグラフはggplotの箇所は全く同じで、プロットタイプを指定するgeom\_以下が変るだけ。

#棒グラフ  
p <- ggplot(data = vdata1, mapping = aes(x = K6Score))  
p + geom\_bar()



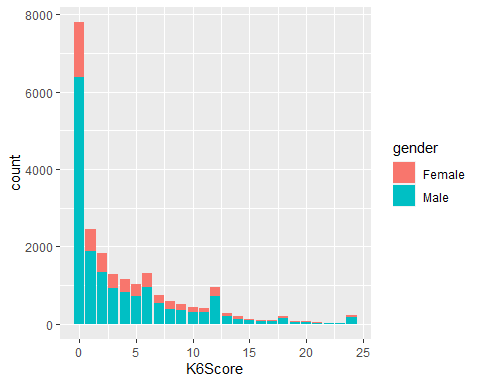
#ヒストグラム  
p <- ggplot(data = vdata1, mapping = aes(x = K6Score))  
p + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



棒グラフで男女の内訳がわかるように記載してみよう。この時、Femは現在女性ならば１、男性ならば０が格納されているが、このままではグラフを作成してくれない。文字列に変換してから作図すると上手く作成できる。

vdata1 <- vdata1 %>% mutate(gender = ifelse(Fem == 1, "Female", "Male"))  
  
p <- ggplot(data = vdata1, mapping = aes(x = K6Score, fill = gender))  
p + geom\_bar()



データセットによっては初めかgenderのように文字列が記載されているものがある。データの型を確認する際にはstr(df)と入力する。今試みにstr(vdata1$変数名)と入力すると以下が出力される。

str(vdata1$sc1\_1)

## int [1:22000] 1 1 1 1 1 1 1 1 1 1 ...

str(vdata1$TrustScore)

## num [1:22000] 20 17 19 20 24 26 25 37 23 23 ...

str(vdata1$gender)

## chr [1:22000] "Male" "Male" "Male" "Male" "Male" "Female" "Female" "Male" ...

intと書かれているのが型です。

これは整数(integer)を表します。

以下の型はすべて数値として扱われます。

int: integer, 整数

dbl: double, 実数

num: numeric, 数値

chr: character, 文字列

genderは文字列として認識されています。文字列として認識されている変数を数値に直す必要がある場合もあります。その場合にはas\_factor()を用いることで、数値解析に用いることができます。

vdata1 <- vdata1 %>% mutate(gender\_factor = as\_factor(gender))  
  
vdata1$gender\_factor %>% table()

## .  
## Male Female   
## 16741 5259

str(vdata1$gender\_factor)

## Factor w/ 2 levels "Male","Female": 1 1 1 1 1 2 2 1 1 1 ...

vdata1$Fem %>% table()

## .  
## 0 1   
## 16741 5259

str(vdata1$Fem)

## num [1:22000] 0 0 0 0 0 1 1 0 0 0 ...