pancake

student_ID	month	score	subject	pancake_frac

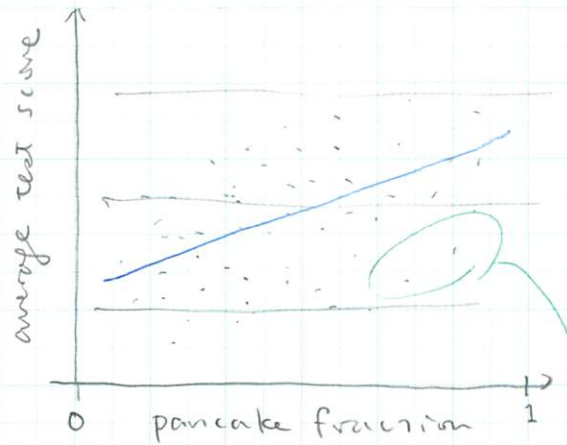
① select & filter relevant data.
② regression
③ scatter plot

```
main ← func(){
    data ← read_interim(folder_nm = , file_nm = master)
    data %>%
        choose_conditions(outcome_var = "average") %>%   (or move?)
        make_reg_table(
        make_scatter_plot(
}
```

↑ verb.

← どこまで分けて書くか.

```
choose_conditions ← fun(data_input, outcome_var){
    data_output ← data_input %>%
        dplyr::select(! dog flank )
        dplyr::filter(                )
    return(data_output)
}
```



alpha

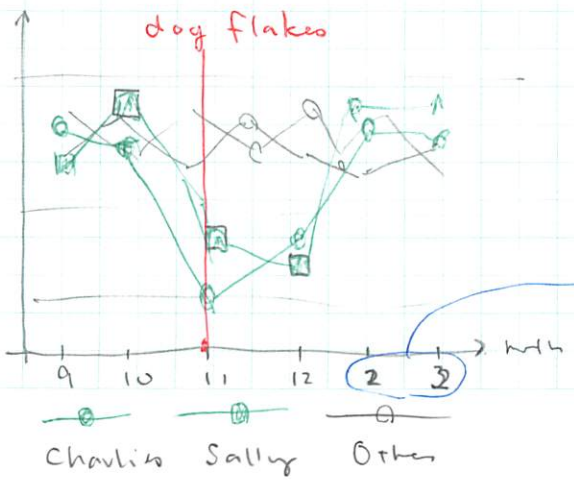
Title:		
OLS	FE	FE
(1)	(2)	(3)
$\hat{\beta}$		
R^2		
N		
FN:		

→ put in table text OR in the LaTeX?

dogflakes.

student_name	math	score	subject	dog flak

① read data
② Bron family group make.
③ plot & save



① 横係だけで十分
② ~~白~~ ~~黒~~) 白黒印刷
③ gray out unimport parts.
④ 係の太さ
⑤ 1月でなく

name	month	score	Bm
A	9		
A	10		
A	11		
	1		

Boolean

1. mutate "Bm family"
2. geom_line (if Bm == 0) "FALSE" in gray dots
3. geom_line (if Bm == 1) "TRUE" in ▭ solid.
4. geom_point (if name == Chuck) ◉
   Sally) ◉
5. background region 
6. put text "dog flakes"

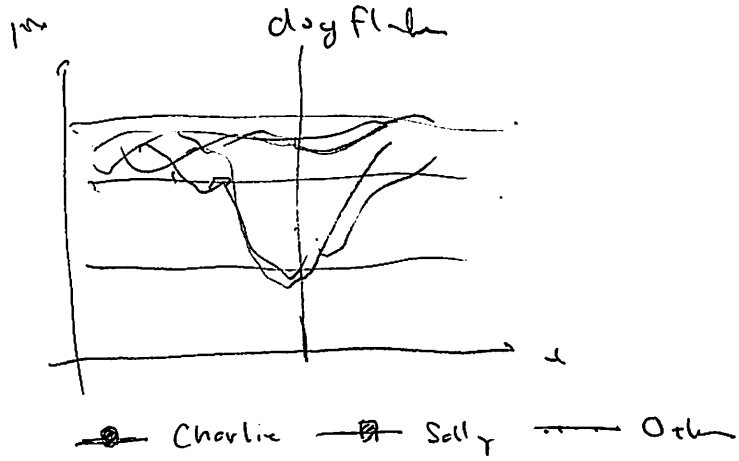
---

titles x = months

y = avg test sc

title = Dog Flake & Test S

3. dog flake 2 개.



month	student	test	subject
9			
10			
11			
12			
21			

rank: 1 2 3 4 5

?

require (ggplot2)

同じようなルールにすればいい？

```
main ← func(){
  data ← read_interim (folder_nm = "master")

  my_plot ← data %>%
    gen_month_order() %>%
    lay_basic( x_var = month_order,
               y_var = test), group = student

  my_plot + set_frame ()
          + set_coordinates ()
          + lay_geom(data)
          + lay_titles ()
          + lay_dogflakes ()

  save_plot (my_plot)
              ↑ my_
}
```

· classic
· serif
· hline

logical stream & title

```
lay_basic ← func(data,
                 x_var,
                 y_var ){
  ggplot2:: ggplot (data,
    mapping = ggplot2::aes ( x = x_var,
                             y = y_var))
}
```

```
degree_transparency = 0.9

alpha = 1 - degree_transparency
```

↑ can assign object w/o disrupting the series?

YES!!

can assign locally.

```
set_theme
ggplot2::theme_classic ( base_size:
                  font = "   ") +
  hline ( hlines, gray) +
  set_y_continuous (
      breaks = hlines,
      labels = hlines ) +
  set_x_continuous (
      breaks = c(     ),
      labels = c(     ))
```

ggplot2:: theme (axis.text.x =
element_text (angle = 45,
hjust = 1))

関数の中でのみ生きている。

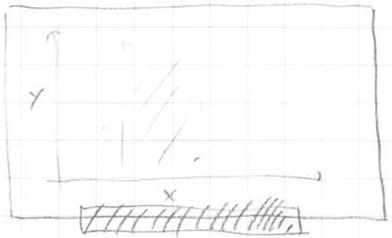
```
hlines ← c("  ")
seq (y_floor, y_ceiling, by = 10)
```

① 読み込む
② month order をつくる
③

9 >
+ 11
②

Charlie 19
Sally 18

図の論理



· 変数，変域，タイトル
· バックグラウンド

```
lay_geom (data)
```

① Charlie, Sally, Others で分ける
② ◎ 図 を point で plot
③ geom lines { orange, orange, gray

```
lay_titles (
```

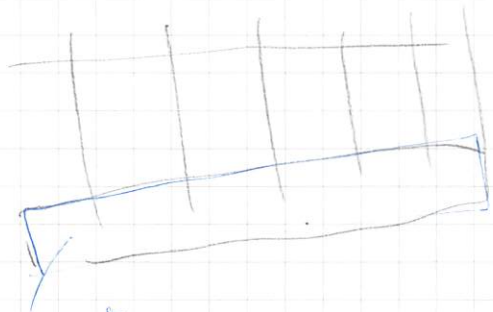
→ 異なる名前で呼んでいる

① define formula.

② loop through reg.

③ use tibble

rbind rows first



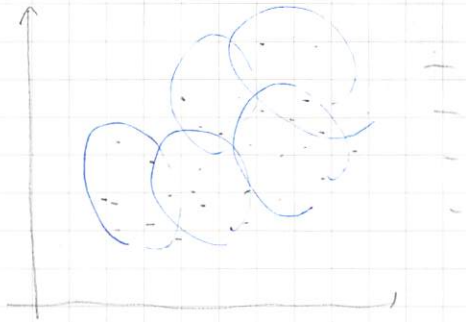
mutate.

↑ reg table

& data frame?

---

* use GitHub for PS instead?



① kable extra output

② reg fig.

as_image.. can output

→ how to add only R^2?

→ line width & color

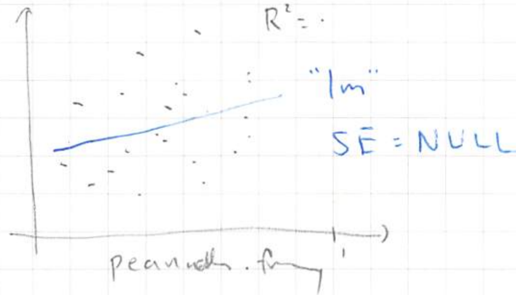
· several themes constantly?

```
basic_plot <- data %>%
    lay_basic(x_var =      )
              y_var =      )
              group =      )

[ ]_plot <- basic_plot %>%
    lay_frame() %>%
    lay_geom() %>%
    lay_title() %>%
    lay_R2(R2 = [ ]$R2)
```

avg	pancake

```
my_plot <- data %>%
    lay_basic(x_var = pancake_frac,
              y_var = test_sc)
              group = student

my_plot + lay_frame()
        + lay_geom(data)
        + lay_title()
        + lay_R2(R2 = 0.01)

save_my_plot(   )

lay_basic <- func(data, x_var, y_var) {
    require(ggplot2)

    ggplot(data,
       mapping = aes(x =   ,
                     y =   ))
}

lay_theme <- func() {
    theme_bw()          <- common
}
```

← how to keep log in Rmd?



・そのコードが何をやってるのか
がわからない

```
scale_color_manual()
```
← カラーパレットの使用.

ggrepel → ラベル付

```
scale_x_continuous(limits = c( , ))+
labs(
```

いい例をたくさん見ること

```
my_plot <- ggplot(data = ,
                  mapping = aes(x = ,
                                y = ))
```

5つのステップ.
① data
② mapping
③ geom
④ coordinate & Ticks
⑤ labels.

```
my_plot <- geom_point(          ) +
```

```
ggsave(here(   , .pdf),
       plot = my_plot,
       height = ,
       width = ,
       units = " ")
```

path = "../"

共通要因

↑ [illegible]

~~my_ggs~~ save_figure(

ggsave

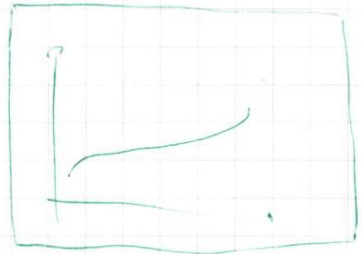
とpngの両方.

dpi = 300,

facet
coord_flip.

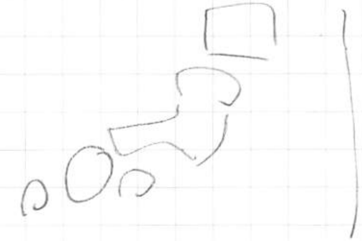
+ interactiveな可視化.
→ 言いたいものを伝えているか.

• プロットの論理的構造を考える



```
scale_x_continuous( breaks = c(1, 2, 3, 4, 5, 6 ),
                    labels = c("Sept", "Oct",    , Feb"   . )

annotate( geom = "rect",
          xmin =      ,
          xmax =     ,
          ymin =
          ymax =
          fill = "red",
          alpha = . ) +
annotate( geom = "text",
          x =
          y =   ,
          label = "dog_flat",
          hjust = 0 ).

geom_text( mapping = aes( label = chamm ))
```

・なぜ試してみないと分からないか



適切な
詳細レベル

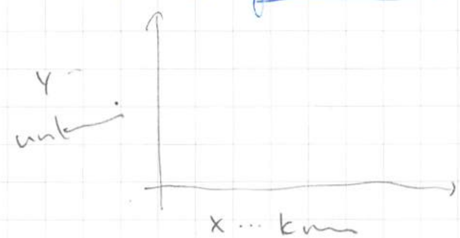
* subtitle

も覚えずに調べられるように

⇒ tidyverse的な書き方をすればいい。

main()のところは、
コンテクストに合った書き方

* check grammar of graphics



customizeする.

```
xlim(1.5, 4.5) +
ylim( 6,  8) +
geom_hline(yintercept = c(   ), col=  )
```

```
geom_smooth( method = "lm",
             se = FALSE)
```

```
theme_classic( base_size = 20,
               base_family = "serif")
```

```
scale_colour_manual(values = c("   "  =  "   ",
                               "   "  =  "   "))
```

フォントの指定.

- 色をつくる
- 名前は?

2つの書き方がある.

・自分なりのカラーを

記法と読みやすさ.

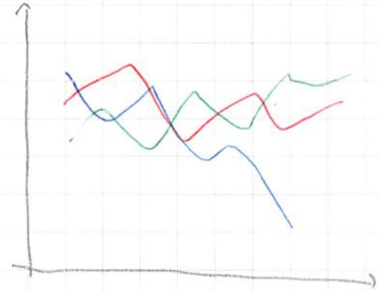
geom_line (color = "gray 70",
mapping = aes (group = student))

↑ if student == "Charlie" ?

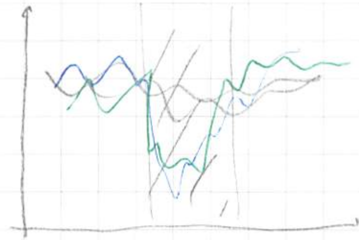
mapping = aes (
geom_line (colour = c ("
"
"
"gray 70"
:
:
), } ごとに決める.

geom_point (((,

・いつ mapping = aes ( ) を使うか



・ただ異なるパターンを
示したい



・色がいいことを伝えている

in drawing
自分のため
他人のため

list(

"OLS" = estimatr::lm_robust(
test = frac ~ ,
cluster = , fe: studi, re-term,
data =)

"FE" :

---

① reshape pivot_wider(

→ cannot remove redundancy in the estimate writing.

• flag to-do.

①

• can always solve at the end
⇒ 何とか

• 焦ることでミス
⇒