

```

1  ##ggplotの基本的使い方 データチェックのための最低限のcode知識
2  ## 機械学習関連のパッケージでもggplotは使われている##
3  # グラフ作成の目的と種類
4  # 1変量:分布の形(多峰性, 正規性, 外れ値, 規則性(系列性))の確認
5  # 連続量:ヒストグラム, 密度曲線, 箱ひげ図
6  # 離散量:棒グラフ, 帯(円)グラフ, ジッターグラフ
7  # 2変量:散布の形状:線形性, 外れ値
8  # 連続量と連続量:散布図
9  # 連続量と離散量/離散量と離散量:ジッター散布図, count散布図
10 #####
11 par(family = "Osaka") #Mac用
12 #install.packages("tidyverse", dep = T)
13 library(tidyverse) #ggplotを含む包括的パッケージ
14 #install.packages("ggplot2", dep = T)
15 library(ggplot2)
16 #####使用するデータ#####
17 data(mpg) # n=234
18 str(mpg)
19 library(MASS)
20 data("Cars93") # n=93
21 str(Cars93)
22 data("diamonds") # n=53940
23 str(diamonds)
24 data("economics") # 時系列データ
25 str(economics)
26 #install.packages("WDI")
27 library(WDI) # 時系列データ
28 wdi <- WDI(indicator="NY.GDP.PCAP.KD",
29            country=c("MX","CA","US"), # 3カ国
30            start=1960, end=2018) # 1960年~2018年
31 names(wdi) <- c("iso2c","country","GDP","year") # 変数名変更
32 str(wdi)
33 library(WDI)
34 gdp <- WDI(indicator="NY.GDP.PCAP.CD",
35            country=c("US","CA","GB","DE",
36                    "CN","JP","SG","IL"),
37            start=1960, end=2018) # 8カ国,1960年~2018年
38 names(gdp) <- c("iso2c","country","GDP","year") # 変数名変更
39 str(gdp)
40 #####
41 # 基本的書き方
42 # ggplot function + Layer (geom or stat function)
43 # aesthetic mappingはggplot, geom どちらに書いても可 (ここではggplot())
44 # data=, mapping=, x=, y= は省略可
45 gbase <- ggplot(data=mpg, mapping=aes(x=displ, y=hwy, color=class))
46 gbase + geom_point()
47 #Layer:グラフの種類 geom_xx, 統計的変換 stat_xx
48 gbase <- ggplot(data=mpg, mapping=aes(x=displ, y=hwy))
49 gbase + geom_point() + #geom_xx
50            stat_smooth() #stat_xx
51 #geom 関数にも統計的変換関数がある
52 gbase + geom_point() + geom_smooth()
53 # ここでは統計的変換はstat 関数を使用
54 ##### 最低限の審美的属性のcode #####
55 # 背景, x, y軸範囲, グラフタイトル, X, Y軸名,
56 # ポイントの形, 色, 文字サイズ, 座標線, 凡例位置
57 ## codeは分けてあるので必要なcodeをコピー ##
58 ##### 1変量:連続量 ヒストグラム#####
59 g1 <- ggplot(mpg, aes(hwy))
60 g1 + geom_histogram() # defaultのbinの数:30, y:count
61 g1 + geom_histogram(binwidth = 2, #binwidth(bin中)を2に, bins:数を指定
62                    fill = "red", #赤
63                    aes(y = ..density..)) + #yをdensityに, default:count
64 #xlim(10,55) + ylim(0.0, 0.15) + #X, Y軸の範囲
65 theme_bw(base_size=15, base_family="Osaka") + #目盛size, Mac文字化け用
66 #背景(白); default:theme_gray(), theme_classic()
67 labs(title="ヒストグラム", x="排気量", y="高速道路での燃費") + #ラベル
68 theme(plot.title = element_text(size = 20)) + #Title size
69 theme(plot.title = element_text(hjust=0.5)) + #Title position
70 theme(axis.title.y = element_text(size = 20)) + #y軸Title size
71 theme(axis.title.x = element_text(size = 20)) + #x軸Title size
72 geom_hline(yintercept=0, lty = 1, lwd=0.5, color="gray") #y=0に座標線
73 # Others
74 g1 + geom_freqpoly() #頻度折れ線:binsはdefault:bins = 30
75 g1 + geom_density() #密度曲線
76 #####
77 #グラフを重ねる
78 g2 <- ggplot(mpg, aes(hwy, fill=factor(cyl))) #fill or color:質的変数
79 g2 + geom_histogram(bins = 20, #binの数指定
80                    position = "nudge", #identity, nudge
81                    alpha = 0.6, #透過率
82                    aes(y = ..count..)) + #density
83 #xlim(10,55) + ylim(0.0, 0.15) + #X, Y軸の範囲
84 theme_bw(base_size=15, base_family="Osaka") + #目盛size, Mac文字化け用
85 #背景(白); default:theme_gray(), theme_classic()
86 labs(title = "層別ヒストグラム", x="高速道路での燃費", y="頻度") +
87 theme(plot.title = element_text(size = 20)) + #Title size
88 theme(plot.title = element_text(hjust=0.5)) + #Title position
89 theme(axis.title.y = element_text(size = 20)) + #y軸Title size
90 theme(axis.title.x = element_text(size = 20)) + #x軸Title size

```

```

91   theme(legend.position = "top") ##凡例位置:bottom, left, right, none(なし)
92   geom_hline(yintercept=0, lty = 2, lwd=0.5, color="gray") #y=0に座標線
93
94   #dataをサブセットに分ける:facet_wrap or facet_grid
95   g2 + geom_histogram(bins = 20, aes(y = ..count..)) +
96     facet_wrap(~ factor(cyl) , nrow = 2) +
97     labs(title = "層別ヒストグラム", x="高速道路での燃費", y="頻度") +
98     geom_hline(yintercept=0, lty = 2, lwd=0.5, color="gray") #y=0に座標線
99   ##### 1変量:離散量#####
100  g3 <- ggplot(mpg , aes(class)) #classを単純集計しグラフ化
101  g3 + geom_bar(stat = "count") #stat=countはデフォルト,省略可
102  g3 + geom_bar(stat = "count")+
103     stat_count(aes(label = ..count..),
104               geom = "text", vjust = 2, colour = "white") #度数の記入
105  g3 + geom_bar(fill="lightblue", #塗りつぶし色
106              colour = "blue") + #枠
107     geom_hline(yintercept=0, lty = 1, lwd=0.5, color="blue")
108  #縦/横棒
109  g4x <- ggplot(mpg , aes(x=class)) #縦棒:xを書かないときも縦棒
110  g4x + geom_bar()
111  g4y <- ggplot(mpg , aes(y=class)) #横棒:y=
112  g4y + geom_bar() +
113     stat_count(aes(label = ..count..),
114              geom = "text", hjust = 2, colour = "white") #度数の記入
115  ###層別:分割
116  g3 + geom_bar(stat = "count") +
117     facet_wrap(~ factor(cyl) , nrow = 2) +
118     stat_count(aes(label = ..count..),
119              geom = "text", vjust = -0.3, colour = "black")
120  ###層別:並び棒
121  g32 <- ggplot(mpg , aes(class , fill = factor(cyl)))
122  g32 + geom_bar(position = position_dodge(width = 0.7) ,
123              width = 0.6) +
124     stat_count(aes(label = ..count..),
125              geom = "text", vjust = -0.2, colour = "black",
126              position = position_dodge(0.7))
127  ###x別にyの値を積み上げる, stat = "identity"を指定する
128  data("ToothGrowth") #ToothGrowth dataを使用
129  #TOTAL
130  g4 <- ggplot(ToothGrowth , aes(x = supp , y = len))
131  g4 + geom_bar(stat = "identity",
132              fill = "lightblue" , color = "red") #積み上がりを確認
133  #層別:facet_wrap(~ dose , nrow = 2)
134  g4 + geom_bar(stat = "identity",
135              fill = "lightblue" , color = "red") +
136     facet_wrap(~ dose , nrow = 2) +
137     labs(title = "ビタミンC(0.5, 1.0, 2.0mg)の3用量レベル別",
138          x="オレンジジュース or アスコルビン酸", y="歯の長さ") +
139     geom_hline(yintercept=0, lty = 2, lwd=0.5, color="gray")
140  ###積み上げ棒
141  g5 <- ggplot(mpg , aes(class , fill = drv))
142  g5 + geom_bar(position = "stack", colour = 1) +
143     stat_count(aes(label = ..count..),
144              geom = "text", hjust = 0.5,
145              position = position_stack(vjust = 0.5) #値を棒の中心に
146              , colour = "black" , size = 5)
147  ##帯グラフ(縦,横)
148  library(scales) #y軸を%表記に
149  g5 + geom_bar(position = "fill") +
150     scale_y_continuous(labels = percent) +
151     theme(axis.title.y=element_blank()) #y軸のタイトルを削除
152  library(dplyr)
153  drv.t <- mpg %>% group_by(class , drv) %>% summarise(n=n()) %>%
154     mutate(per = paste0(round(100 * n/sum(n), 1) , "%"))
155  drv.t #構成比
156  ##### X:離散量, Y:連続量 #####
157  g6 <- ggplot(mpg , aes(drv , hwy))
158  g6 + geom_jitter() #jitter plot
159     stat_summary(fun = "median",
160                colour = "red", size = 2, geom = "point")
161  g6 + geom_boxplot() #box plot
162     stat_summary(fun = "mean",
163                colour = "red", size = 2, geom = "point")
164  g6 + geom_violin() #violin plot:確率密度曲線を縦にし左右を合わせる
165     stat_summary(fun = "mean",
166                colour = "red", size = 2, geom = "point")
167  #3変数,
168  g7 <- ggplot(mpg , aes(class , hwy , fill = drv)) #colour=drv (箱:白抜き)
169  g7 + geom_boxplot() +
170     theme(legend.position="top")
171  ##### x,y:連続変量 #####
172  ##散布図
173  gp <- ggplot(mpg , aes( displ , hwy))
174  gp + geom_point()
175  gp + geom_point(pch = 20, size = 2, color = "red") + # point:サイズ,形,色
176     xlim(1.5, 7.0) + ylim(10.0 , 45.0) + #X,Y軸の範囲
177     theme_bw(base_size=15, base_family="Osaka") + #目盛size, Mac文字化け用
178     #背景(白); default:theme_gray(), theme_classic()
179     labs(title = "散布図", x="排気量", y="高速道路での燃費") +
180     theme(plot.title = element_text(size = 20)) + #Title size

```

```

181   theme(plot.title = element_text(hjust=0.5)) + #Title position
182   theme(axis.title.y = element_text(size = 20)) + #y軸Title size
183   theme(axis.title.x = element_text(size = 20)) #x軸Title size
184   #theme(legend.position = "top") + #凡例位置:bottom,left,right,none(なし)
185   #geom_hline(yintercept=0,lty = 2,lwd=0.5,color="gray") #y=0に座標線
186 #層別 aesthetic attributes 層別の表現:色,形,サイズ,濃淡
187 gco <- ggplot(mpg, aes( displ , hwy , colour = factor(cyl))) #colour:色
188 gco + geom_point()
189 gsp <- ggplot(mpg, aes( displ , hwy , shape = factor(cyl))) #shape:形
190 gsp + geom_point()
191 gsz <- ggplot(mpg , aes( displ , hwy , size = cyl))#size:ポイントサイズ
192 gsz + geom_point()
193 gap <- ggplot(mpg , aes( displ , hwy , alpha = cyl))#alpha:濃淡
194 gap + geom_point()
195 #dataをサブセットに分ける facet_wrap facet_grid & 出力パターン
196 gp + geom_point() +
197   facet_wrap(~class,nrow=2) #classでサブセット作成:nrow(作図)
198 gp + geom_point() +
199   facet_wrap(~class,ncol=2) #classでサブセット作成:ncol(作図)
200 gp + geom_point() +
201   facet_grid(~class) #classでサブセットを作成:~ class(作図)
202 gp + geom_point() +
203   facet_grid(class ~.) #classでサブセットを作成:class ~.(作図)
204 ### 回帰直線の追加#####
205 #geom_smooth or stat_smooth
206 gp + geom_point() +
207   stat_smooth(method = "lm", #指定可:lm,glm,gam,loess,rlm
208             level=0.99 , se = T, #level:信頼区間,se=F;信頼区間無し
209             colour = "black", size = 1) #線の色と太さ
210 #層別
211 gco <- ggplot(mpg , aes( displ , hwy , colour = factor(cyl))) #colour:色
212 gco + geom_point() +
213   stat_smooth(method = "lm" , se = F)
214
215 #ロジスティクス曲線への当てはめ
216 data("mtcars")
217 gl <- ggplot(mtcars , aes(wt , am)) #プロットの重なりがあるのでJitterで
218 gl + geom_point(position=position_jitter(width=0.25,height=0.05)) +
219   stat_smooth(method = "glm" , #確率分布をmethod.argsに書く
220             method.args = list(family = binomial(link = "logit")),
221             level=0.95 , se = T,colour = "red", size = 1)
222 #ロジスティクス回帰分析のcode
223 out <- glm(am ~ wt, family = binomial(link = "logit"), mtcars)
224 summary(out)
225 #平均値に座標
226 gp + geom_point() +
227   geom_hline(yintercept=mean(mpg$cty) , lty = 2 , lwd=0.2) +
228   geom_vline(xintercept=mean(mpg$displ), lty = 2 , lwd=0.2)
229 #平滑化線
230 gp + geom_point() + geom_smooth()
231 gpd <- ggplot(diamonds , aes( carat , price)) #diamonds data
232 gpd + geom_point() + geom_smooth()
233 #対数変換
234 gp + geom_point() +
235   scale_x_log10() + scale_y_log10() +
236   geom_smooth(method = "lm", se = T,
237             colour = "red", size = 1)
238 gpd + geom_point() + # diamonds data
239   scale_x_log10() + scale_y_log10() +
240   geom_smooth(method = "lm", se = T,
241             colour = "red", size = 1)
242 ## Jitter plot
243 ggp <- ggplot(mpg , aes(drv , hwy))
244 ggp + geom_jitter(size=2)
245 ggp + geom_jitter(width = 0.2 , size=2) + #jitter幅狭く
246   stat_summary(fun = "mean" , color = "red" , geom = "point")
247 ##text
248 ggc <- ggplot(Cars93 ,
249             aes(EngineSize , MPG.city , label = Model)) #text出力項目
250 ggc + geom_text(size = 2.5) #geom_text:pchの代わりにtext
251 library(ggrepel)
252 ggc + geom_point() + geom_text_repel(size = 3) #引出線で
253
254 ggplot(Cars93 , aes(EngineSize , MPG.city)) +
255   geom_text(aes(label = Model, family = "Osaka"),
256           size=3, show.legend = F)
257 #####折れ線グラフ#####
258 library(dplyr)
259 canada <- wdi %>% filter(country == "Canada" )
260 #基本
261 g.l <- ggplot(canada , aes(year , GDP))
262 g.l + geom_line(linetype = "dashed" , #"twodash", "dotted"
263             color = "red" , lwd = 0.7 ) +
264   labs(title="GDP:Canada",x="1960-2018年",y="GDP(US$) ")
265 #ラインタイプで分ける
266 g.lg <- ggplot(wdi , aes(year , GDP , group = country))
267 g.lg + geom_line(aes(linetype = country)) +
268   geom_point(aes(shape = country)) +
269   labs(title="GDP:US/CA/MEX",x="1960-2018",y="GDP(US$) ") +
270   scale_x_continuous(breaks=seq(1960,2020,5)) + #x軸:5年間隔

```

```

271     scale_y_continuous(breaks=seq(0,60000,5000)) #y軸:$5000間隔
272 #色で分ける
273 g.lc <- ggplot(wdi , aes(year , GDP , color = country))
274 g.lc + geom_line() + geom_point() +
275     labs(title="GDP:US/CA/MEX",x="1960-2018",y="GDP(US$)") +
276     scale_x_continuous(breaks=seq(1960,2020,5)) + #x軸:5年間隔
277     scale_y_continuous(breaks=seq(0,60000,5000)) #y軸:$5000間隔
278 ##### x,y:離散量#####
279 par(family="Osaka")
280 ggplot(diamonds , aes(cut , color)) +
281     geom_count(alpha=0.3,color = "blue") +
282     theme_bw(base_family="Osaka") +
283     scale_size_area(max_size = 20) +
284     labs(title = "離散量*離散量" , x = "Cut" , y = "Color")
285 table(diamonds$cut,diamonds$color)
286 ##比率で作図
287 ggplot(diamonds, aes(x = cut, y = clarity)) +
288     geom_count()+scale_size_area(max_size = 10)#groupなし:度数
289 ggplot(diamonds, aes(x = cut, y = clarity)) +
290     geom_count(aes(size=after_stat(prop),group=1))+#group:比率計算ベース
291     scale_size_area(max_size = 10) #1:全体,x=cut,y=clarity
292 ## ##組み合わせる 箱ひげ図とJitter #####
293 ggplot(mpg , aes(driv , hwy)) +
294     geom_boxplot() +
295     geom_jitter(width = 0.25) +
296     stat_summary(fun = "mean" , color = "red" , geom = "point")
297 ## 特定の層だけ色をつける
298 library(gghighlight)
299 ggplot(mpg,aes(displ,hwy,colour = factor(cyl))) +
300     geom_point(size=2) +
301     gghighlight() +
302     facet_wrap(~ cyl , ncol = 2)
303 ##複数のグラフを1シートに
304 p <- ggplot(mpg , aes(driv , hwy)) +geom_point(size=1.5)
305 j <- ggplot(mpg , aes(driv , hwy)) +geom_jitter(width = 0.25,size=1.5)
306 b <- ggplot(mpg , aes(driv , hwy)) +geom_boxplot()
307 v <- ggplot(mpg , aes(driv , hwy)) +geom_violin()
308 library("cowplot")
309 plot_grid(p , j , b , v, align = "h")# align軸を揃える
310 library("ggpubr")
311 ggarrange(p , j , b , v, labels = LETTERS) #グラフにアルファベット付記
312 library("Rmisc")
313 multiplot( p , j , b , v , cols = 2 )
314 library("gridExtra")
315 grid.arrange(p , j , b , v , ncol = 2) #nrow
316 ##ソートしてグラフ化
317 ggplot( mpg , aes(reorder(class , hwy , FUN = median),hwy)) +
318     geom_boxplot() +
319     labs(title = "散布図" , subtitle = "(中央値でソート)" ,
320         x = "Class",y = "高速燃費") +
321     theme_bw(base_size = 20) +
322     theme_gray (base_family = "Osaka")
323 #横
324 ggplot( mpg , aes(reorder(class , hwy , FUN = median),hwy)) +
325     geom_boxplot() +
326     labs(title = "散布図" , subtitle = "(中央値でソート)" ,
327         x = "Class",y = "高速燃費") +
328     theme_bw(base_size = 20) +
329     theme_gray (base_family = "Osaka") +
330     coord_flip()
331
332 ###平均値のグラフ
333 library(dplyr)
334 g.df <- group_by(mpg, class)#dataをclass別に:グループ別集計に便利
335 mean <- summarize(g.df, n = n() , mean = mean(hwy) )
336 mean #平均値をclass別に算出,data.frame形式で出力
337 #平均値の棒グラフと全体平均(赤破線)
338 g.m <- ggplot(mean , aes(class , mean))
339 g.m + geom_bar(stat = "identity") +
340     ggtitle("高速道での燃費:Class別平均値と全体平均値(赤線)") +
341     geom_hline(yintercept=(mean(mpg$hwy)),lty = 2,lwd=0.5,color="red")
342

```