WK02 Data Exploration

1 데이터 탐색

1.1 참고 사이트:

- RStudio Cheat Sheets (https://rstudio.com/resources/cheatsheets/ (https://rstudio.com/resources/cheatsheets/)): 최신 치트시트
- Data Visualization Cheat Sheet (https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf (https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf))
- Data Transformation Cheat Sheet (https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf (https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf))
- ggplot2 사이트 (https://docs.ggplot2.org/current/ (https://docs.ggplot2.org/current/))

1.2 htwtbd 자료설명

- htwtbd00.csv: 2021년 온라인으로 수집한 연예인 신체계측자료
  - n = 84. 여자 42명(배우 40, 가수 1, 개그맨 1), 남자 42명(배우 40, 가수 1, 개그맨 1)

변수	설명
name	이름
gnd	성별{F, M}. 이진 판별분석시 타겟
byr	출생년도
ht	키(cm). 회귀분석시 타겟
wt	몸무게(kg)
bd	혈액형{A,AB,B,O}
a	분야{actor, singer, comedian}

- Model Lookup (https://topepo.github.io/caret/available-models.html (https://topepo.github.io/caret/available-models.html))
- install.packages("caret", dependencies=c("Depends", "Suggests"))

1.3 패키지

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## √ ggplot2 3.3.3     √ purrr   0.3.4
## √ tibble  3.1.0     √ dplyr   1.0.5
## √ tidyr   1.1.3     √ stringr 1.4.0
## √ readr   1.4.0     √ forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra) # ggplot 객체를 한 페이지에 표시. grid.arrange(..., nrow, ncol)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(scales) # 시각화 축조정 scale_x_xxx
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(skimr)  # 기초통계량 + 결측정보
library(naniar) # 결측 정보
```

```
##
## Attaching package: 'naniar'
```

```
## The following object is masked from 'package:skimr':
##
##     n_complete
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

1.4 읽기

```
# as.data.frame으로 안바꾸면 caret vs tidyverse 호환문제 때문에 경고 발생
# as.data.frame해도 문자변수를 factor화 하지 않음
DF <- as.data.frame(read_csv('D:/Github/Statics/DataMining/0321/htwtbd00.csv'))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   name = col_character(),
##   gnd = col_character(),
##   byr = col_double(),
##   ht = col_double(),
##   wt = col_double(),
##   bd = col_character(),
##   a = col_character(),
##   ftln = col_double()
## )
```

```
head(DF)
```

```
##     name gnd  byr  ht wt bd     a ftln
## 1 강소라   F 1990 168 NA  A actor   NA
## 2 김고은   F 1991 167 NA  B actor   NA
## 3 김민희   F 1982 170 49  A actor  240
## 4 김아중   F 1982 170 48  A actor   NA
## 5 김태리   F 1990 166 46  B actor   NA
## 6 김태희   F 1980 165 45  O actor   NA
```

```
dim(DF)
```

```
## [1] 84  8
```

```
str(DF) # (Old) sapply(DF, class)
```

```
## 'data.frame':    84 obs. of  8 variables:
##  $ name: chr  "강소라" "김고은" "김민희" "김아중" ...
##  $ gnd : chr  "F" "F" "F" "F" ...
##  $ byr : num  1990 1991 1982 1982 1990 ...
##  $ ht  : num  168 167 170 170 166 165 170 168 168 164 ...
##  $ wt  : num  NA NA 49 48 46 45 NA 45 48 NA ...
##  $ bd  : chr  "A" "B" "A" "A" ...
##  $ a   : chr  "actor" "actor" "actor" "actor" ...
##  $ ftln: num  NA NA 240 NA NA NA NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   name = col_character(),
##   ..   gnd = col_character(),
##   ..   byr = col_double(),
##   ..   ht = col_double(),
##   ..   wt = col_double(),
##   ..   bd = col_character(),
##   ..   a = col_character(),
##   ..   ftln = col_double()
##   .. )
```

1.5 전처리

- age: 나이계산
- 이산형 변수처리
    - 문자변수(gnd, bd)를 factor화
    - {0,1}로 코딩된 가변수는 그대로 숫자형으로 사용. factor화해도 되지만 해석시 유의

```
# factor로 다 바꿀 것. lm, rpart, rf은 안해도 무방, gbm에서 오류남

DF <- mutate(DF,
             age = 2021-byr,
             gnd = factor(gnd),
             bd = factor(bd),
             a = factor(a))
sapply(DF, class)
```

```
##        name         gnd         byr          ht          wt          bd
## "character"    "factor"   "numeric"   "numeric"   "numeric"    "factor"
##           a        ftln         age
##    "factor"   "numeric"   "numeric"
```

1.6 기초통계량/결측파악

- skimr::skim(data):summary()와 결측정보. group_by와 연결. pandas::describe()와 유사
- naniar::vis_miss(data): 변수별 결측비율 시각화
    - 주의: 출력물의 Missing(%)과 Present(%)는 완전 결측값 비율이 아니고, 전체 셀 중 결측의 비율임
    - sum(complete.cases(DF)): 완전 관측값 개수 반환
    - 원자료가 너무 크면 랜덤 추출(sample_n)해서 파악할 것
    - DF %>% dplyr::sample_frac(size=0.1) %>% vis_miss()
- naniar::miss_var_summary(data):: 변수별 결측비율 요약

```
skim(DF)
```

Data summary

Name	DF
Number of rows	84
Number of columns	9
________________________	
Column type frequency:	
character	1
factor	3
numeric	5
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1	2	3	0	84	0

Variable type: factor

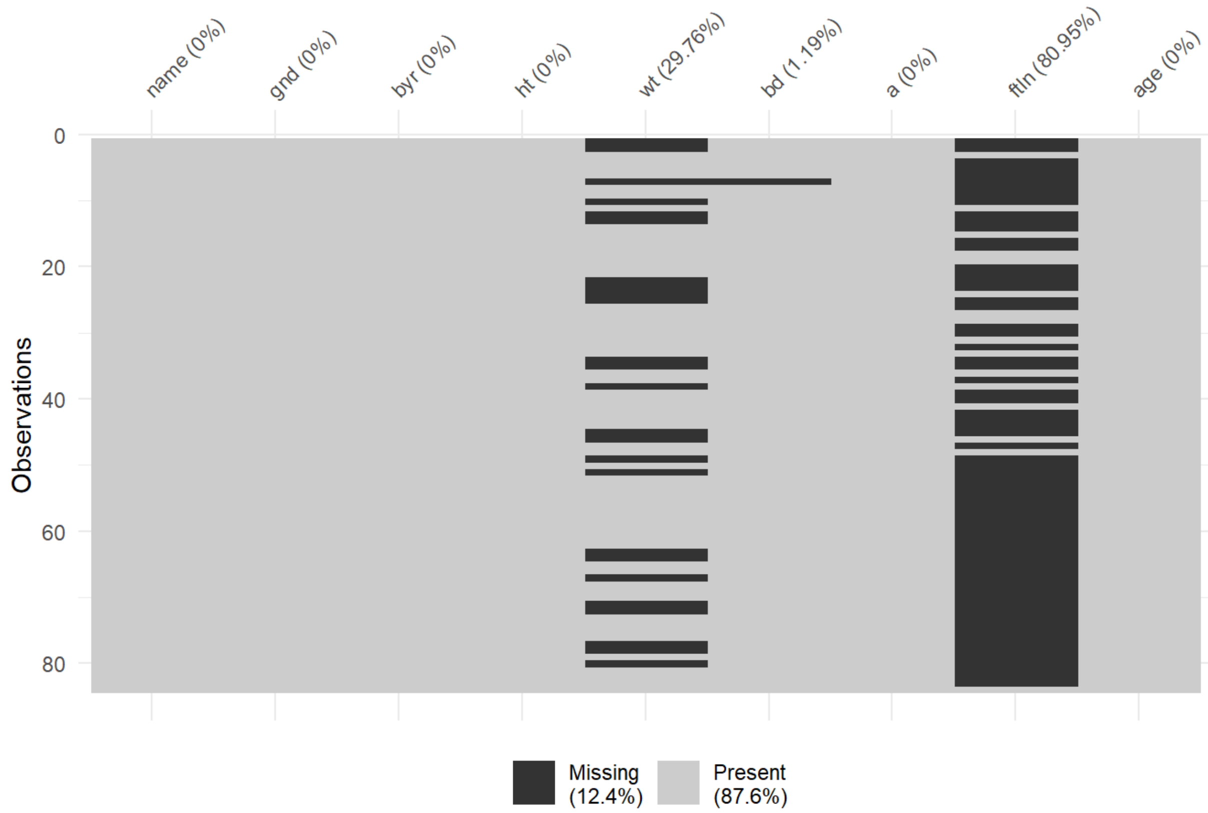
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gnd	0	1.00	FALSE	2	F: 42, M: 42
bd	1	0.99	FALSE	4	B: 26, A: 25, O: 21, AB: 11

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
a	0	1.00	FALSE	3	act: 80, com: 2, sin: 2

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
byr	0	1.00	1982.65	7.40	1967	1978.00	1982	1989.00	1994	▃▂▇▅▇
ht	0	1.00	173.33	8.68	149	166.75	172	181.00	188	▁▂▇▅▆
wt	25	0.70	59.27	13.14	43	47.50	60	70.00	100	▇▂▅▁▁
ftln	68	0.19	245.31	23.06	215	235.00	240	246.25	310	▂▇▁▁▁
age	0	1.00	38.35	7.40	27	32.00	39	43.00	54	▇▅▇▂▃

```
# 변수별 결측비율, Missing=결측셀/전체셀, Present=비결측셀/전체셀
naniar::vis_miss(DF)
```



```
naniar::miss_var_summary(DF)
```

```
## # A tibble: 9 x 3
##   variable n_miss pct_miss
##   <chr>     <int>    <dbl>
## 1 ftln         68    81.0 
## 2 wt           25    29.8 
## 3 bd            1     1.19
## 4 name          0     0   
## 5 gnd           0     0   
## 6 byr           0     0   
## 7 ht            0     0   
## 8 a             0     0   
## 9 age           0     0   
```

```
# 완전 관측값 비율 = 15%
# 회귀분석계통 분석방법을 그대로 적용하면 전체 자료의 15%만 사용하게 됨
sum(complete.cases(DF))/nrow(DF)*100  # prop_complete_case(DF)
```

```
## [1] 15.47619
```

1.7 탐색

1.7.1 단변량 탐색

- 연속형 변수의 탐색
  - 수치요약: 평균, 표준편차
  - 시각화: 히스토그램, density(커널분포추정), 상자그림, rug

```
summary(DF)
```

```
##      name           gnd         byr            ht             wt        
##  Length:84          F:42   Min.   :1967   Min.   :149.0   Min.   : 43.00  
##  Class :character   M:42   1st Qu.:1978   1st Qu.:166.8   1st Qu.: 47.50  
##  Mode  :character          Median :1982   Median :172.0   Median : 60.00  
##                            Mean   :1983   Mean   :173.3   Mean   : 59.27  
##                            3rd Qu.:1989   3rd Qu.:181.0   3rd Qu.: 70.00  
##                            Max.   :1994   Max.   :188.0   Max.   :100.00  
##                                                           NA's   :25      
##     bd            a           ftln           age       
##  A   :25   actor   :80   Min.   :215.0   Min.   :27.00  
##  AB  :11   comedian: 2   1st Qu.:235.0   1st Qu.:32.00  
##  B   :26   singer  : 2   Median :240.0   Median :39.00  
##  O   :21                 Mean   :245.3   Mean   :38.35  
##  NA's: 1                 3rd Qu.:246.2   3rd Qu.:43.00  
##                          Max.   :310.0   Max.   :54.00  
##                          NA's   :68                     
```

```
# summarize_if(.tbl, .predicate:logical, .funs:list, ...)
# summarize_at(, tbl, .vars_vector, .fybs:list, ...)
summarize_if(DF, is.numeric, list(mn=mean, sd=sd), na.rm=TRUE)
```

```
##     byr_mn    ht_mn    wt_mn  ftln_mn   age_mn   byr_sd    ht_sd    wt_sd
## 1 1982.655 173.3333 59.27119 245.3125 38.34524 7.398297 8.683641 13.13698
##    ftln_sd   age_sd
## 1 23.05564 7.398297
```

```
summarize_at(DF, c('ht', 'wt'), list(mn=mean, sd=sd), na.rm=TRUE)
```

```
##      ht_mn    wt_mn    ht_sd    wt_sd
## 1 173.3333 59.27119 8.683641 13.13698
```

```
DF%>% dplyr::select_if(is.numeric) %>% skim()
```

Data summary

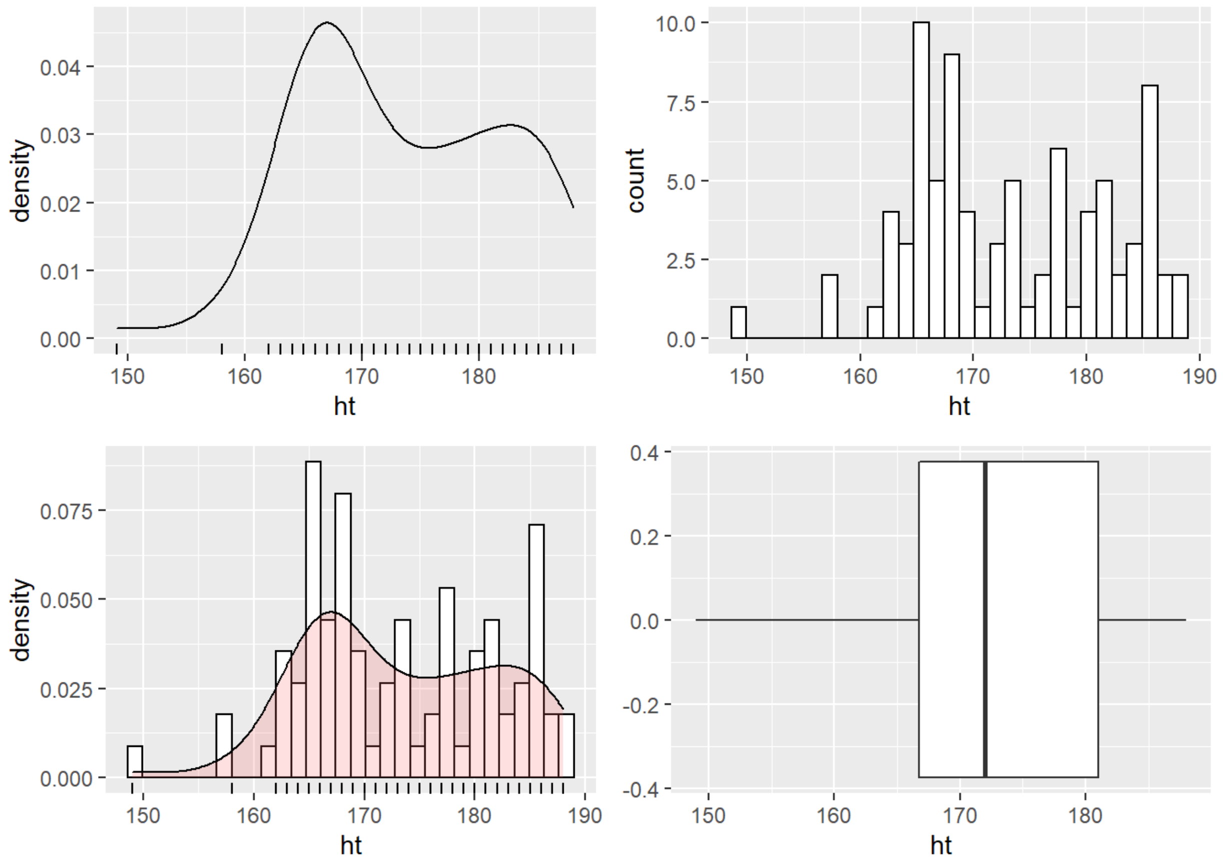
Name	Piped data
Number of rows	84
Number of columns	5
_______________________	
Column type frequency:	
numeric	5
________________________	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
byr	0	1.00	1982.65	7.40	1967	1978.00	1982	1989.00	1994	▃▂▇▅▇
ht	0	1.00	173.33	8.68	149	166.75	172	181.00	188	▁▃▇▅▆
wt	25	0.70	59.27	13.14	43	47.50	60	70.00	100	▇▂▅▁▁
ftln	68	0.19	245.31	23.06	215	235.00	240	246.25	310	▂▇▁▁▁
age	0	1.00	38.35	7.40	27	32.00	39	43.00	54	▇▅▇▂▃

```
g1 <- ggplot(DF, aes(x=ht)) + geom_density() + geom_rug()
g2 <- ggplot(DF, aes(x=ht)) + geom_histogram(color='black', fill='white')
g3 <- ggplot(DF, aes(x=ht)) + geom_histogram(aes(y=..density..), color='black', fill='white') + geom_den
sity(alpha=0.2, fill='#FF6666') + geom_rug()
g4 <- ggplot(DF, aes(x=ht)) + geom_boxplot()
grid.arrange(g1, g2, g3, g4, nrow=2, ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- 이산형 변수의 탐색
  - 수치요약: 빈도, 상대빈도
  - 시각화: 막대그래프(barplot)

```
DF %>% dplyr::select_if(is.factor) %>% skim()
```

Data summary

Name	Piped data
Number of rows	84
Number of columns	3
________________________	
Column type frequency:	
factor	3
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gnd	0	1.00	FALSE	2	F: 42, M: 42
bd	1	0.99	FALSE	4	B: 26, A: 25, O: 21, AB: 11
a	0	1.00	FALSE	3	act: 80, com: 2, sin: 2

```
table(DF$gnd)
```

```
##
##  F  M
## 42 42
```

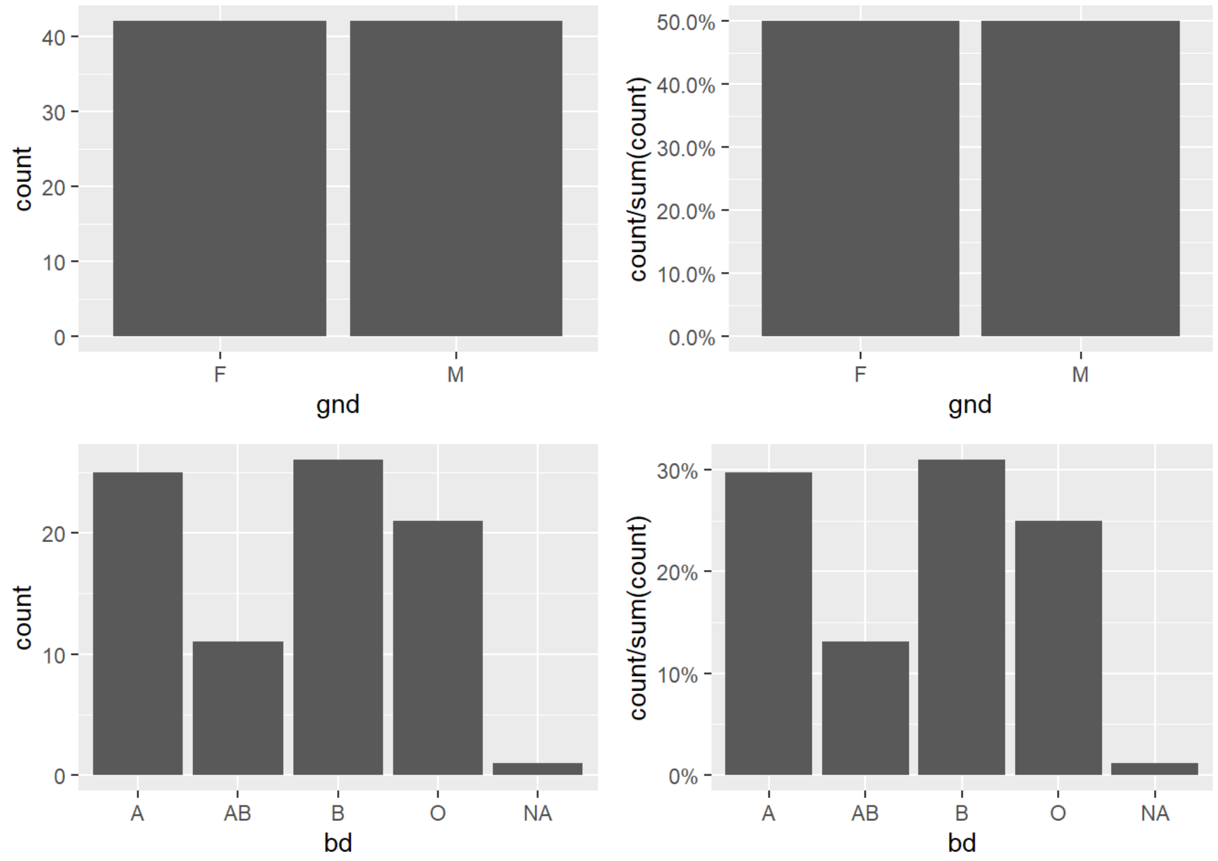
```
xtabs(~bd, data=DF)
```

```
## bd
##  A AB  B  O
## 25 11 26 21
```

```
xtabs(~a, data=DF)
```

```
## a
##    actor comedian   singer
##       80        2        2
```

```
g1 <- ggplot(DF, aes(x=gnd)) + geom_bar()
g2 <- ggplot(DF, aes(x=gnd)) + geom_bar(aes(y=..count../sum(..count..))) + scale_y_continuous(labels=per
cent)
g3 <- ggplot(DF, aes(x=bd)) + geom_bar()
g4 <- ggplot(DF, aes(x=bd)) + geom_bar(aes(y=..count../sum(..count..))) + scale_y_continuous(labels=perc
ent)
grid.arrange(g1, g2, g3, g4, nrow=2, ncol=2)
```



1.7.2 이변량 탐색

- 연속 ~ 이산

```
DF %>% group_by(gnd) %>% dplyr::select_if(is.numeric) %>% skim()
```

Data summary

Name	Piped data
Number of rows	84
Number of columns	6
_______________________	
Column type frequency:	
numeric	5
________________________	
Group variables	gnd

Variable type: numeric

skim_variable	gnd	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
byr	F	0	1.00	1985.10	6.22	1970	1981.25	1985.5	1990.00	1994	▁▁▆▆▇
byr	M	0	1.00	1980.21	7.74	1967	1975.25	1979.5	1987.50	1994	▆▅▇▃▅
ht	F	0	1.00	166.12	4.33	149	164.25	166.5	168.00	173	▁▁▂▇▂
ht	M	0	1.00	180.55	5.22	168	177.00	181.0	185.00	188	▁▂▇▇▇
wt	F	13	0.69	47.55	3.55	43	45.00	47.0	49.00	60	▇▇▃▁▁
wt	M	12	0.71	70.60	7.93	55	65.50	70.0	72.75	100	▃▇▃▁▁
ftln	F	29	0.31	236.15	9.61	215	235.00	240.0	240.00	250	▁▂▆▇▆
ftln	M	39	0.07	285.00	22.91	265	272.50	280.0	295.00	310	▇▇▁▁▇
age	F	0	1.00	35.90	6.22	27	31.00	35.5	39.75	51	▇▆▆▁▁
age	M	0	1.00	40.79	7.74	27	33.50	41.5	45.75	54	▅▃▇▅▆

```
DF %>%
  group_by(gnd) %>%
  summarize_at(c('ht', 'wt'), list(mn=mean, sd=sd), na.rm=TRUE)
```

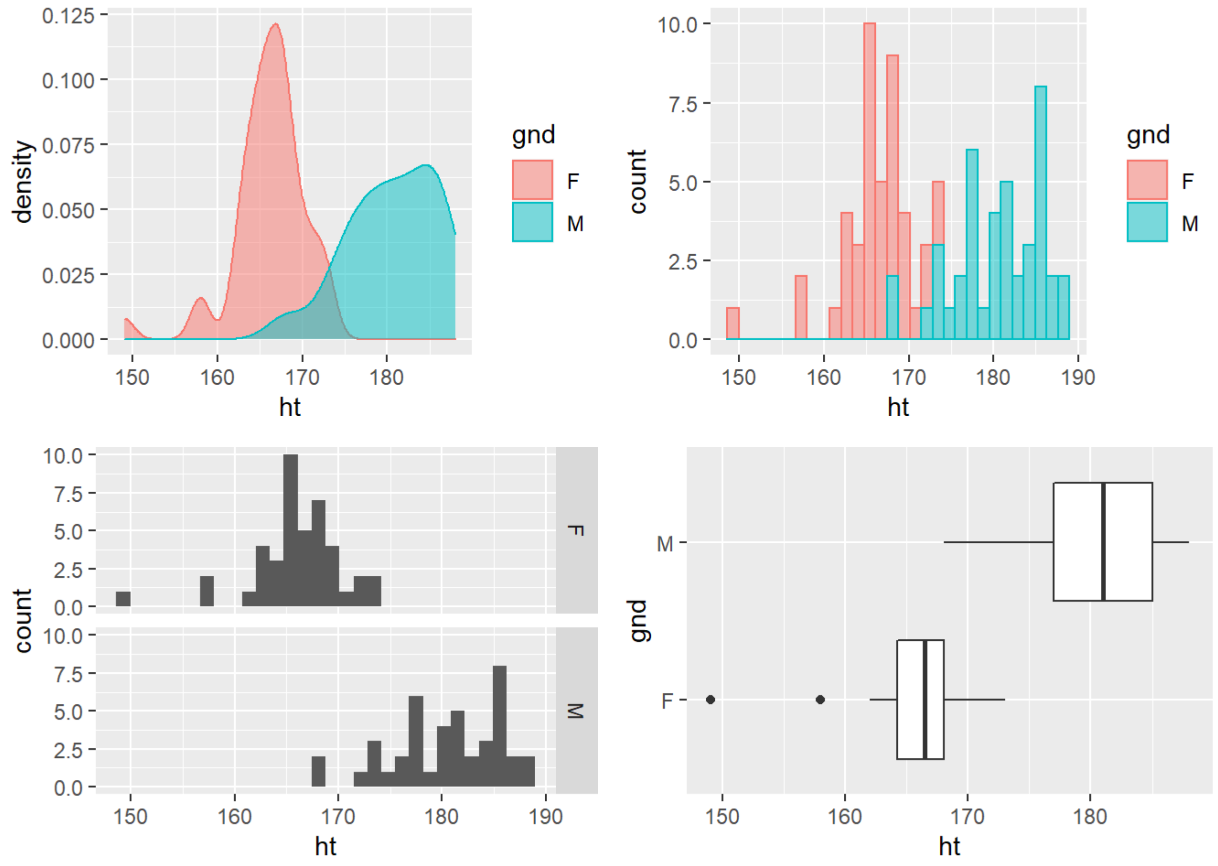
```
## # A tibble: 2 x 5
##   gnd   ht_mn wt_mn ht_sd wt_sd
##   <fct> <dbl> <dbl> <dbl> <dbl>
## 1 F      166.  47.6  4.33  3.55
## 2 M      181.  70.6  5.22  7.93
```

```
DF %>%
  group_by(gnd) %>%
  summarize_if(is.numeric, list(mn = mean, sd=sd), na.rm=TRUE)
```

```
## # A tibble: 2 x 11
##   gnd   byr_mn ht_mn wt_mn ftln_mn age_mn byr_sd ht_sd wt_sd ftln_sd age_sd
##   <fct>  <dbl> <dbl> <dbl>   <dbl>  <dbl>  <dbl> <dbl> <dbl>   <dbl>  <dbl>
## 1 F      1985.  166.  47.6    236.   35.9   6.22  4.33  3.55    9.61   6.22
## 2 M      1980.  181.  70.6    285    40.8   7.74  5.22  7.93   22.9    7.74
```

```
g1 <- ggplot(DF, aes(x=ht, col=gnd, fill=gnd)) + geom_density(alpha=0.5)
g2 <- ggplot(DF, aes(x=ht, col=gnd, fill=gnd)) + geom_histogram(alpha=0.5)
g3 <- ggplot(DF, aes(x=ht)) + geom_histogram() + facet_grid(gnd~.)
g4 <- ggplot(DF, aes(x=gnd, y=ht)) + geom_boxplot() + coord_flip()
grid.arrange(g1, g2, g3, g4, nrow=2, ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
t.test(ht~gnd, data=DF, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  ht by gnd
## t = -13.784, df = 82, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.51091 -12.34623
## sample estimates:
## mean in group F mean in group M
##        166.1190        180.5476
```

```
summary(aov(ht~bd, data=DF))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## bd           3     70   23.21   0.297  0.828
## Residuals   79   6178   78.20
## 1 observation deleted due to missingness
```

- 연속 ~ 연속

```
# cor(DF[,sapply(DF, is.numeric)], use='pairwise.complete.obs')
R <- cor(DF%>% select_if(is.numeric), use='pairwise.complete.obs')
R
```

```
##             byr         ht         wt       ftln        age
## byr   1.0000000 -0.2659908 -0.3483837 -0.3646281 -1.0000000
## ht   -0.2659908  1.0000000  0.8110682  0.8912221  0.2659908
## wt   -0.3483837  0.8110682  1.0000000  0.8381219  0.3483837
## ftln -0.3646281  0.8912221  0.8381219  1.0000000  0.3646281
## age  -1.0000000  0.2659908  0.3483837  0.3646281  1.0000000
```

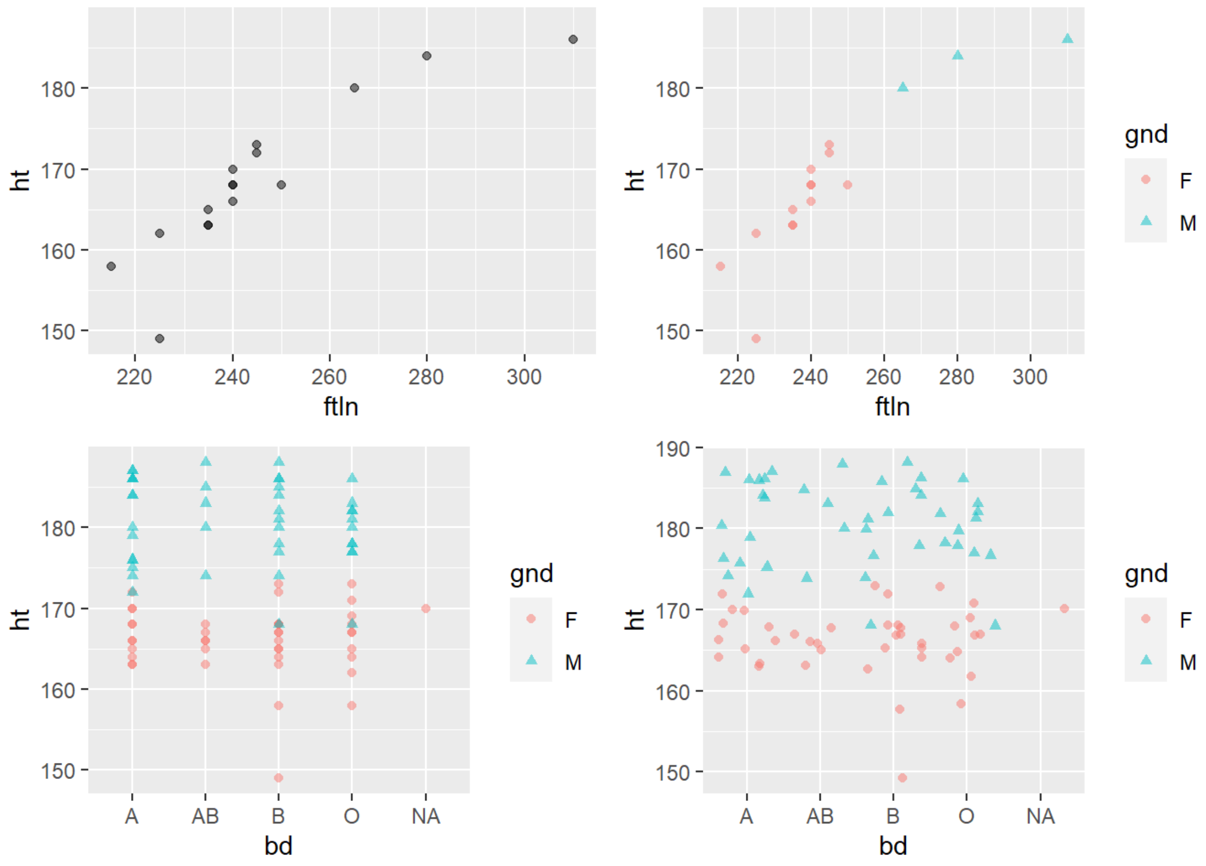
```
sort(R['ht',], decreasing=TRUE)
```

```
##        ht      ftln        wt       age       byr
##  1.0000000 0.8912221 0.8110682 0.2659908 -0.2659908
```

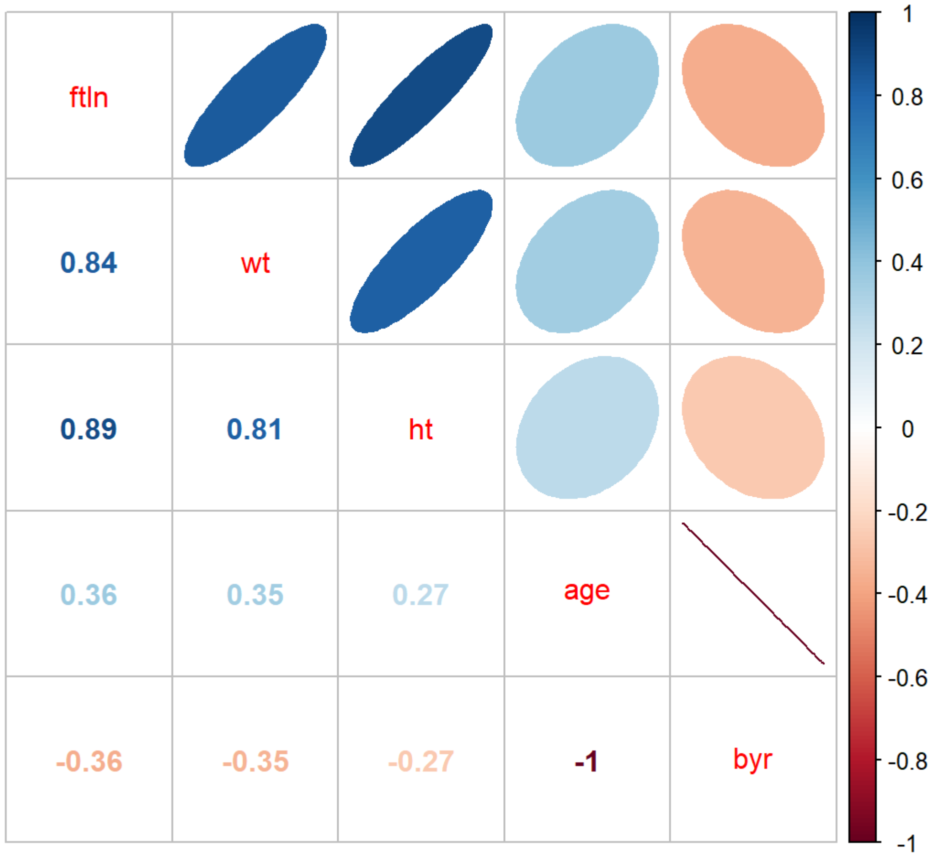
```
g1 <- ggplot(DF, aes(x=ftln, y=ht)) + geom_point(alpha=0.5)
g2 <- ggplot(DF, aes(x=ftln, y=ht, color=gnd, shape=gnd)) + geom_point(alpha=0.5)
g3 <- ggplot(DF, aes(x=bd, y=ht, color=gnd, shape=gnd)) + geom_point(alpha=0.5)
g4 <- ggplot(DF, aes(x=bd, y=ht, color=gnd, shape=gnd)) + geom_jitter(alpha=0.5)
grid.arrange(g1, g2, g3, g4, nrow=2, ncol=2)
```

```
## Warning: Removed 68 rows containing missing values (geom_point).

## Warning: Removed 68 rows containing missing values (geom_point).
```



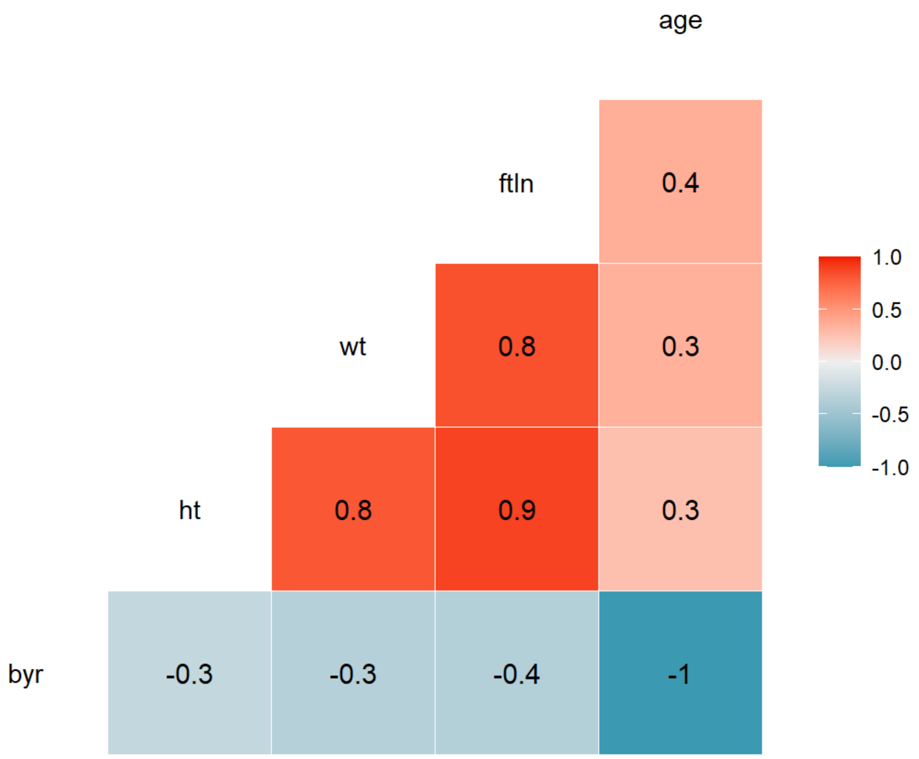
```
corrplot::corrplot.mixed(R, upper='ellipse', order='FPC')
```



```
library(GGally) # ggcorr, ggparis
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggcorr(DF %>% select_if(is.numeric), geom = 'tile', label=TRUE)
```



```
ggpairs(DF,
        columns=c('ht', 'ftln', 'wt'),
        lower=list(continuous=wrap('points', alpha=0.05, col='blue')),
        diag=list(continuous='barDiag'))  # diag=list(continous='densityDiag')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 68 rows containing missing values
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 25 rows containing missing values
```

```
## Warning: Removed 68 rows containing missing values (geom_point).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

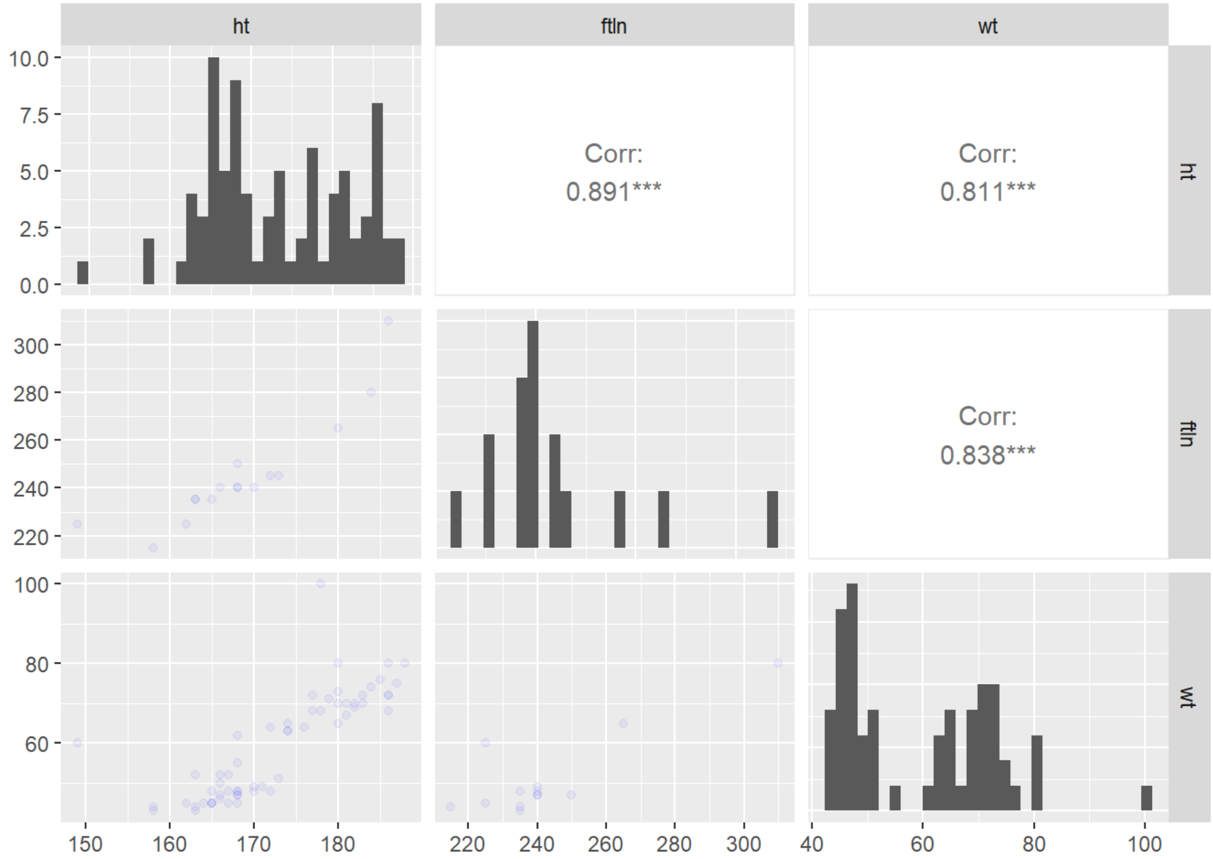
```
## Warning: Removed 68 rows containing non-finite values (stat_bin).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 71 rows containing missing values
```

```
## Warning: Removed 25 rows containing missing values (geom_point).
```

```
## Warning: Removed 71 rows containing missing values (geom_point).
```

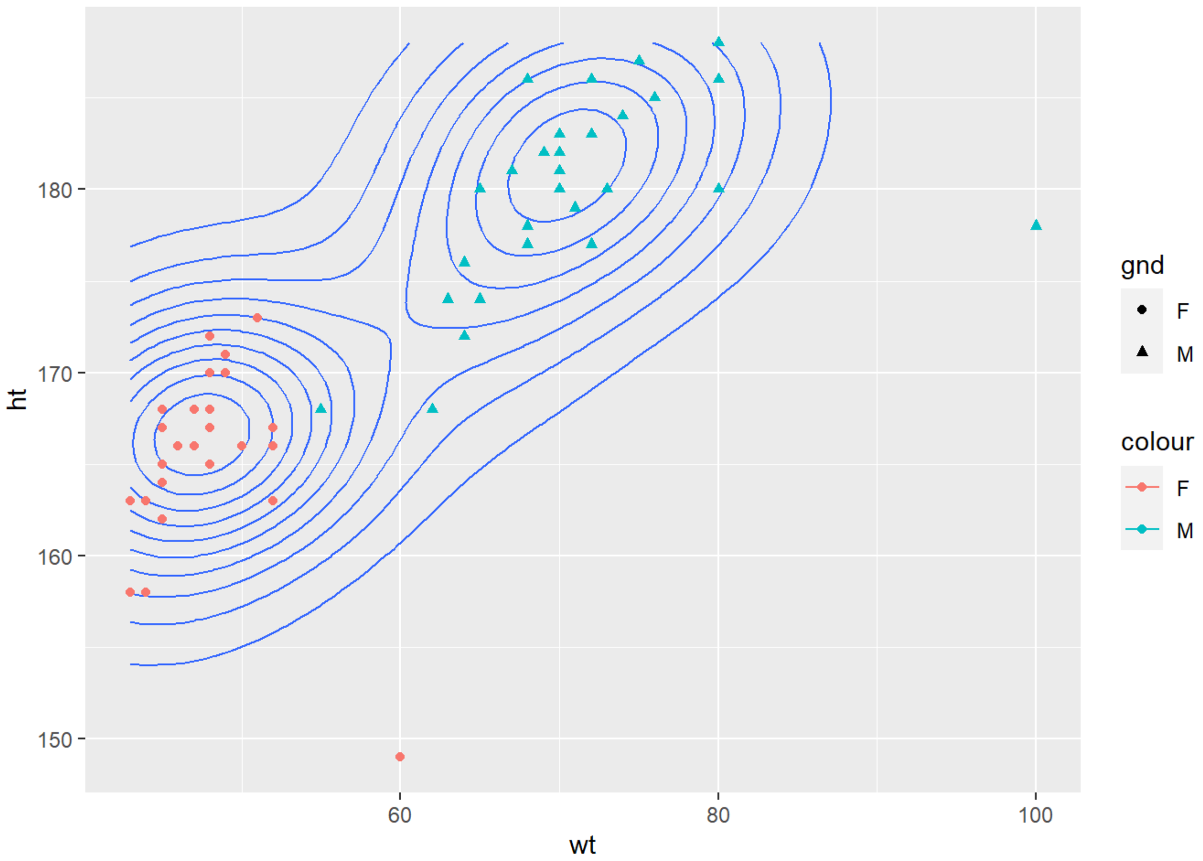
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 25 rows containing non-finite values (stat_bin).
```



```
ggplot(DF, aes(x=wt, y=ht)) + geom_density2d() + geom_point(aes(col=gnd, shape=gnd))
```

```
## Warning: Removed 25 rows containing non-finite values (stat_density2d).
```

```
## Warning: Removed 25 rows containing missing values (geom_point).
```



- 이산 ~ 이산

```
g1 <- ggplot(DF, aes(x=bd, fill=gnd)) + geom_bar()
g2 <- ggplot(DF, aes(x=bd, fill=gnd)) + geom_bar(aes(y=..count../sum(..count..)))

# Or
tb <- table(DF$gnd, DF$bd)
tb <- xtabs(~bd+gnd, data=DF)
df <- data.frame(tb)
df
```

```
##    bd gnd Freq
## 1   A   F   11
## 2  AB   F    6
## 3   B   F   14
## 4   O   F   10
## 5   A   M   14
## 6  AB   M    5
## 7   B   M   12
## 8   O   M   11
```

```
g3 <- ggplot(df, aes(x=gnd, y=Freq)) + geom_bar(aes(fill=bd), stat='identity')

tb <- prop.table(xtabs(~gnd+bd, data=DF), 1)
tb
```

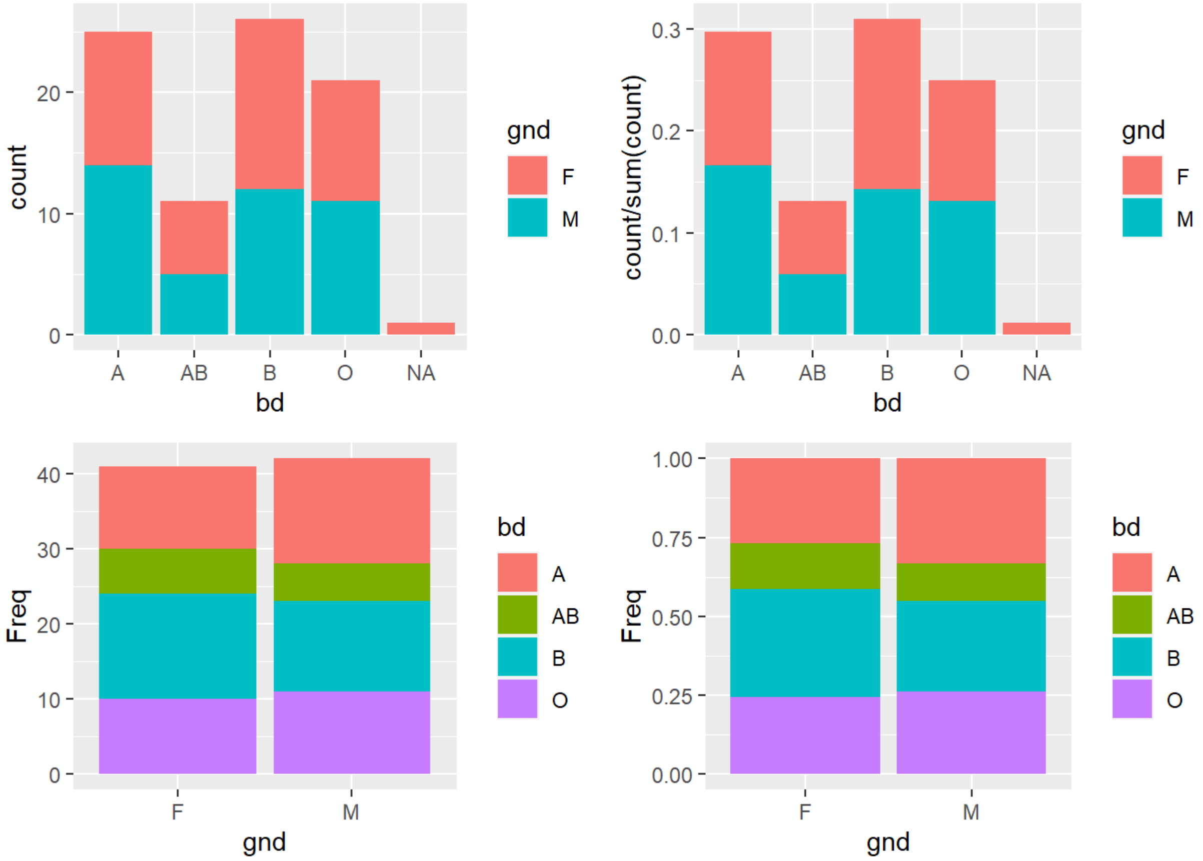
```
##    bd
## gnd         A        AB         B         O
##   F 0.2682927 0.1463415 0.3414634 0.2439024
##   M 0.3333333 0.1190476 0.2857143 0.2619048
```

```
df <- data.frame(tb)
df
```

```
##   gnd bd      Freq
## 1   F  A 0.2682927
## 2   M  A 0.3333333
## 3   F AB 0.1463415
## 4   M AB 0.1190476
## 5   F  B 0.3414634
## 6   M  B 0.2857143
## 7   F  O 0.2439024
## 8   M  O 0.2619048
```

```
g4 <- ggplot(df, aes(x=gnd, y=Freq)) + geom_bar(aes(fill=bd), stat='identity')

grid.arrange(g1, g2, g3, g4, nrow=2, ncol=2)
```



```
chisq.test(xtabs(~gnd+bd, data=DF), correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  xtabs(~gnd + bd, data = DF)
## X-squared = 0.64042, df = 3, p-value = 0.8871
```