Lecture 6 Notes

2024-02-01

Two quick notes

1. `as_factor()` (from `haven`) versus `to_character()` (from `labelled`)
2. Proportions

```
rm(list = ls())
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts t
o become errors
```

```
# This line opens the raw data from github
dat <- read_rds("https://github.com/jbisbee1/DS1000_S2024/raw/main/data/MI2020_ExitPoll.
rds")

dat %>%
  count(FAVTRUMP)
```

```
## # A tibble: 4 × 2
##     FAVTRUMP     n
##   <hvn_lbl_> <int>
## 1          1   220
## 2          2   385
## 3          9    10
## 4         NA   616
```

```
dat %>%
  mutate(FAVTRUMP_1 = as.character(haven::as_factor(FAVTRUMP)),
         FAVTRUMP_2 = labelled::to_character(FAVTRUMP)) %>%
  count(FAVTRUMP,
        FAVTRUMP_1,
        FAVTRUMP_2)
```

```
## # A tibble: 4 × 4
##   FAVTRUMP                                FAVTRUMP_1                FAVTRUMP_2     n
##   <dbl+lbl>                               <chr>                     <chr>      <int>
## 1  1 [Favorable]                          Favorable                 Favorable    220
## 2  2 [Unfavorable]                        Unfavorable               Unfavorab…   385
## 3  9 [[DON'T READ] Don’t know/refused] [DON'T READ] Don’t know… [DON'T RE…    10
## 4 NA                                      <NA>                      <NA>         616
```

```
dat <- dat %>%
  mutate(fav_trump_text = labelled::to_character(FAVTRUMP))
```

2. Proportions

```
dat %>%
  count(fav_trump_text)
```

```
## # A tibble: 4 × 2
##   fav_trump_text                      n
##   <chr>                           <int>
## 1 Favorable                         220
## 2 Unfavorable                       385
## 3 [DON'T READ] Don’t know/refused    10
## 4 <NA>                              616
```

```
# Manually
220 / (220+385+10)
```

```
## [1] 0.3577236
```

```
# Using code
dat %>%
  count(fav_trump_text) %>%
  filter(!is.na(fav_trump_text)) %>%
  mutate(totVoters = sum(n)) %>%
  mutate(prop = n / totVoters)
```

```
## # A tibble: 3 × 4
##   fav_trump_text                      n totVoters   prop
##   <chr>                           <int>     <int>  <dbl>
## 1 Favorable                         220       615 0.358
## 2 Unfavorable                       385       615 0.626
## 3 [DON'T READ] Don’t know/refused    10       615 0.0163
```

```
# Proportion of trump approvers by gender
dat %>%
  mutate(gender = labelled::to_character(SEX)) %>%
  count(gender,fav_trump_text) %>%
  filter(!is.na(fav_trump_text)) %>%
  group_by(gender) %>%
  mutate(totGender = sum(n)) %>%
  ungroup() %>%
  mutate(prop = n / totGender)
```

```
## # A tibble: 6 × 5
##   gender fav_trump_text                      n totGender   prop
##   <chr>  <chr>                           <int>     <int>  <dbl>
## 1 Female Favorable                         107       339 0.316
## 2 Female Unfavorable                       227       339 0.670
## 3 Female [DON'T READ] Don’t know/refused     5       339 0.0147
## 4 Male   Favorable                         113       276 0.409
## 5 Male   Unfavorable                       158       276 0.572
## 6 Male   [DON'T READ] Don’t know/refused     5       276 0.0181
```

```
view(dat)

dat %>%
  select(starts_with("LG"))
```

```
## # A tibble: 1,231 × 1
##    LGBT
##    <dbl+lbl>
##  1 NA
##  2  2 [No]
##  3  2 [No]
##  4 NA
##  5 NA
##  6  2 [No]
##  7  2 [No]
##  8  2 [No]
##  9 NA
## 10 NA
## # ℹ 1,221 more rows
```

Univariate Analysis

- NBA data introduction

```
nba <- read_rds("https://github.com/jbisbee1/DS1000_S2024/raw/main/data/nba_players_2018.Rds")

nba
```

```
## # A tibble: 530 × 37
##    namePlayer    idPlayer slugSeason numberPlayerSeason isRookie slugTeam idTeam
##    <chr>            <dbl> <chr>                   <dbl> <lgl>    <chr>     <dbl>
##  1 LaMarcus Ald…   200746 2018-19                    12 FALSE    SAS      1.61e9
##  2 Quincy Acy      203112 2018-19                     6 FALSE    PHX      1.61e9
##  3 Steven Adams    203500 2018-19                     5 FALSE    OKC      1.61e9
##  4 Alex Abrines    203518 2018-19                     2 FALSE    OKC      1.61e9
##  5 Bam Adebayo    1628389 2018-19                     1 FALSE    MIA      1.61e9
##  6 Rawle Alkins   1628959 2018-19                     0 TRUE     CHI      1.61e9
##  7 Grayson Allen  1628960 2018-19                     0 TRUE     UTA      1.61e9
##  8 Deng Adel      1629061 2018-19                     0 TRUE     CLE      1.61e9
##  9 Jaylen Adams   1629121 2018-19                     0 TRUE     ATL      1.61e9
## 10 DeVaughn Ako…  1629152 2018-19                     0 TRUE     DEN      1.61e9
## # ℹ 520 more rows
## # ℹ 30 more variables: gp <dbl>, gs <dbl>, fgm <dbl>, fga <dbl>, pctFG <dbl>,
## #   fg3m <dbl>, fg3a <dbl>, pctFG3 <dbl>, pctFT <dbl>, fg2m <dbl>, fg2a <dbl>,
## #   pctFG2 <dbl>, agePlayer <dbl>, minutes <dbl>, ftm <dbl>, fta <dbl>,
## #   oreb <dbl>, dreb <dbl>, treb <dbl>, ast <dbl>, stl <dbl>, blk <dbl>,
## #   tov <dbl>, pf <dbl>, pts <dbl>, urlNBAAPI <chr>, n <int>, org <fct>,
## #   country <chr>, idConference <int>
```

```
view(nba)
```

Selecting variables

```
nba %>%
  select(pts)
```

```
## # A tibble: 530 × 1
##      pts
##    <dbl>
##  1  1727
##  2    17
##  3  1108
##  4   165
##  5   729
##  6    37
##  7   211
##  8    32
##  9   108
## 10     7
## # ℹ 520 more rows
```

```
nba$pts
```

```
##   [1] 1727   17 1108  165  729   37  211   32  108    7  760 1994  346  178  873
##  [16]    0  189    2  134  948   30  544  469  316  923 1159  415  209  699  494
##  [31]  779  284  465  292  151  829  181  596  606 2099  230  917  687  223   34
##  [46]   39 1241  741  236 1454  240  990  114  418 1700   17  207   93  136  684
##  [61]  597   19  190   55  964 1001  135  372  319  248  678  167  113  938  282
##  [76]  758   82  562  226  744 1114  965  229   10  172  455  401 1364    9  148
##  [91]    2    6 1478  853  488  413  509  421  512 1188   39  508 1881  194 1452
## [106]  951  581  318   36  430  157 1635  472  693  131    0   46    3  494 1370
## [121]  485 1143  387   19  779  190 1526 2027  287 1761  519    5    3   41  142
## [136]  706  897  288  240  420  513   40  129   22 1226  608  967  155 1399   15
## [151]  946 1346  753 2159  672   26  233 1103 1284 1090 1246  184  408  889  217
## [166]   23  675  946 1841  821  486  250   57    0  428 2818  460  737 1041 1361
## [181]    8   25  825   79  421  511  408  474   53  525   12  151 1420  191    0
## [196]  527 1695  294    9  115  925 1112   17  572  366  727  229   15  389  453
## [211]    0 1596  994  950   22  720  495  339 1505  459 1260  910   27  798   13
## [226]  201  428 1604  130   81   27  537  427    2  468  405  422  613    0  156
## [241] 1492  279 1308  322  485  963  534 1596 1208  854   86  547  541  158  225
## [256]  258  374 1009  704  156 2067  357  500    0  926   10    2  546   29   23
## [271]    6   29  139   20  115  974  195    2  897   75 1468  503  160  564  483
## [286]  337    4  883   51  811 1407  567  141   35  250   51  633  653 1829   16
## [301]   10   87 1046    6  873 1367  102  851   19  373  298  378  196  529  627
## [316]  235  241  156 1125  484  426  334  246  186  304  530  267  187  787  675
## [331]  991  570  906  229   80  637  446   11    5   81   98  815  516  423  742
## [346]  107 1372 1565  271 1209    2   30  284   12   95  424  917 1223  198  708
## [361]   47  112  485   50  190  864  709 1224   80 1712 1043   71    3 1337 1354
## [376]    0 1371   75   74  501  444  708   40  272  510  491    2 1243   66   48
## [391]   31  326  205  470 1680   97 1880   62  374   61  601  494  119  984    0
## [406]   40  695 1083 2102 1665  527  571  701   24  207   45 1675  663  140  772
## [421]   40  218  663   18   39  756 1498  887   39 1321   18    0  279  157 1024
## [436]    9  830  111   94  280 1549   81  497  459   63  863   14 1265  151  625
## [451]  561 1215  181  711  631  134  173  305  182  465   21  282  287  296  545
## [466]  389  114  384    6  310  118 1071  611 1644 1174   54  458  857  805  121
## [481]  583  128  619  454   23  758  915  600  266  136   67  707  430  231  840
## [496]  422  323   74  339  230  697    7  547  449  133   36  930 1047  250  780
## [511]  710  618  459  858  465  366  514  722   65  399  722   54  586   43   10
## [526]  763  653  172   46  525
```

```
summary(nba$pts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   115.0   419.0   516.2   759.5  2818.0
```

Categorical versus Continuous

```
nba %>%
  select(namePlayer)
```

```
## # A tibble: 530 × 1
##    namePlayer
##    <chr>
##  1 LaMarcus Aldridge
##  2 Quincy Acy
##  3 Steven Adams
##  4 Alex Abrines
##  5 Bam Adebayo
##  6 Rawle Alkins
##  7 Grayson Allen
##  8 Deng Adel
##  9 Jaylen Adams
## 10 DeVaughn Akoon-Purcell
## # ℹ 520 more rows
```

```
mean(nba$namePlayer)
```

```
## Warning in mean.default(nba$namePlayer): argument is not numeric or logical:
## returning NA
```

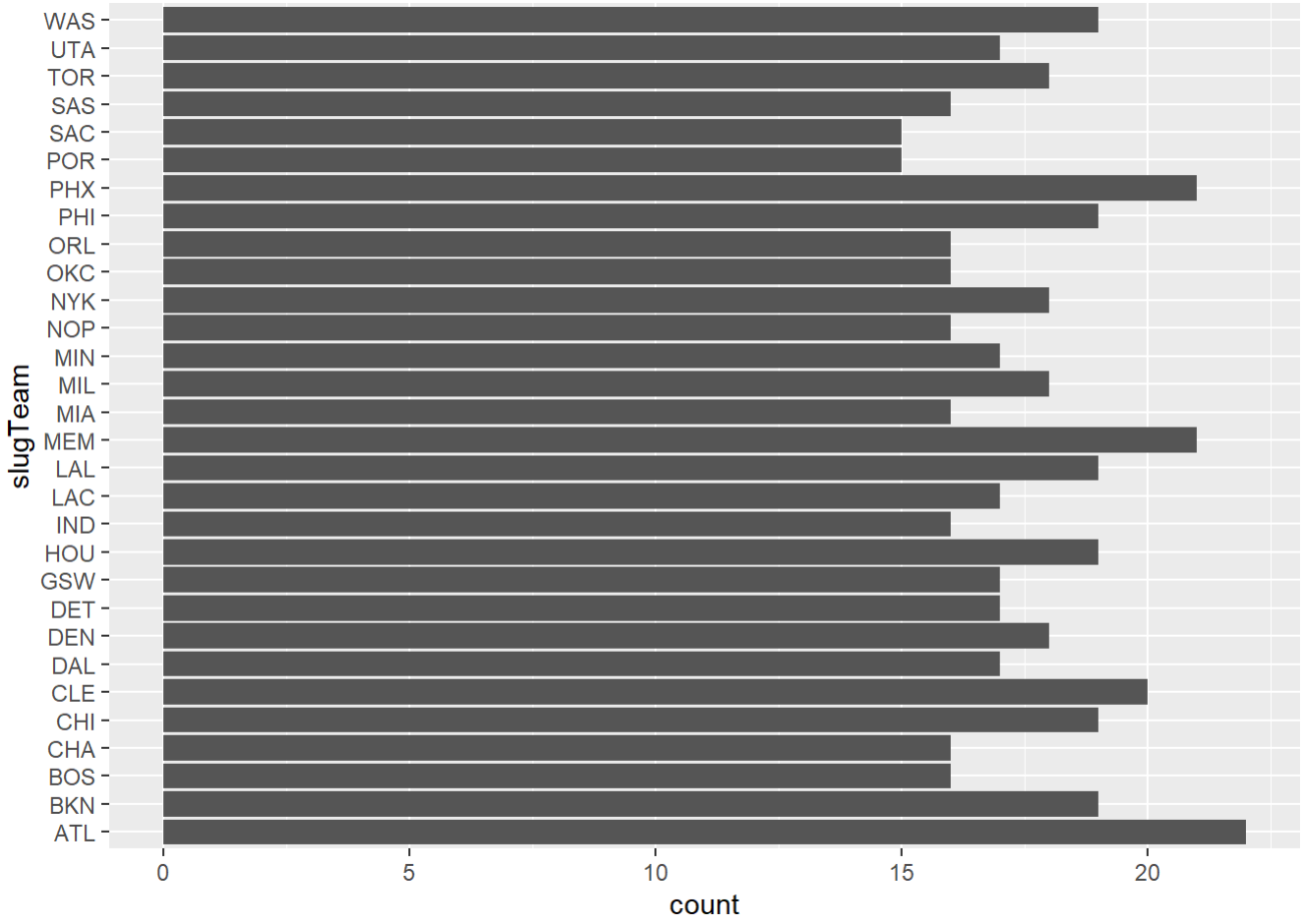
```
## [1] NA
```

```
nba %>%
  count(pts)
```

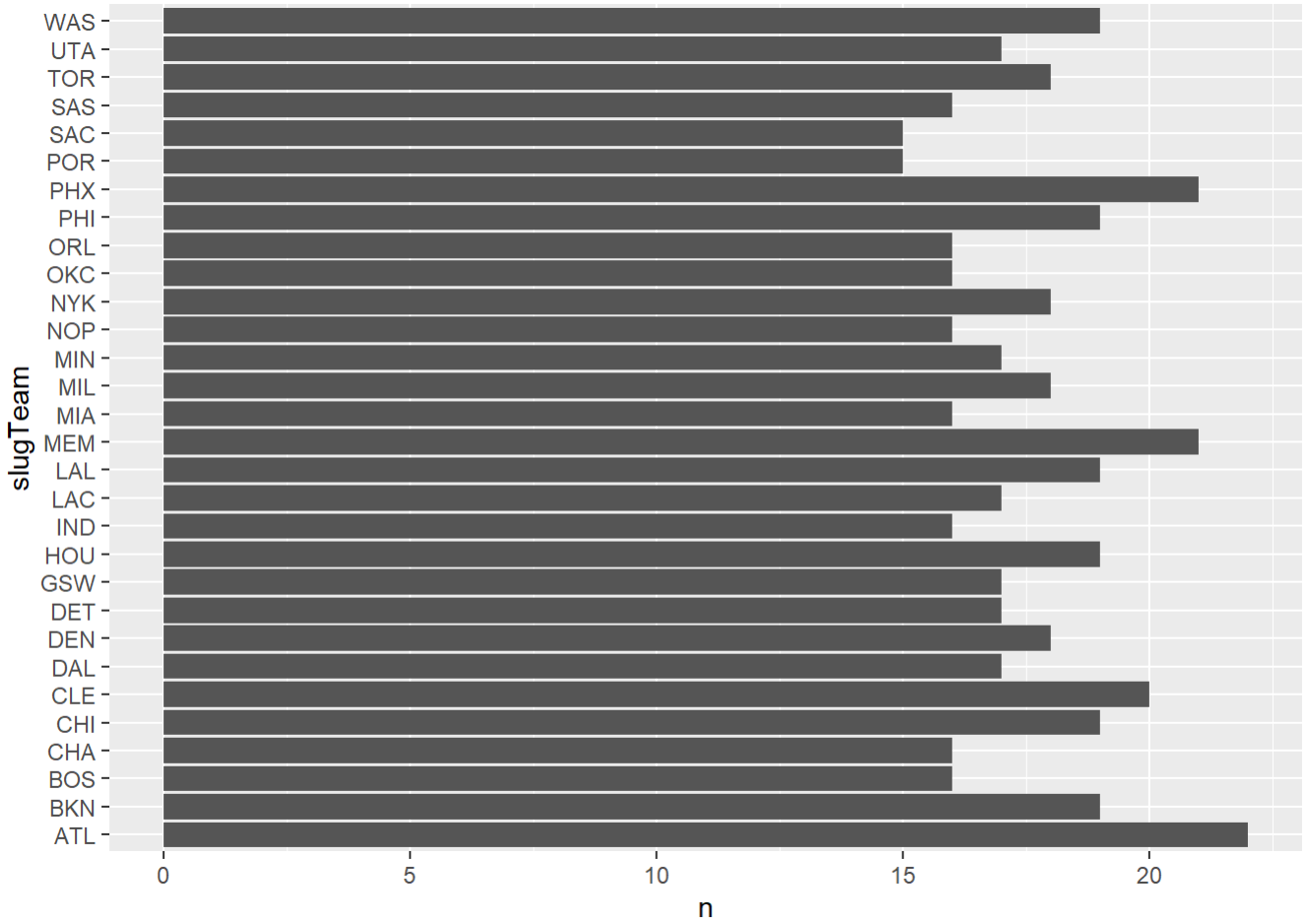
```
## # A tibble: 419 × 2
##      pts     n
##    <dbl> <int>
##  1     0    10
##  2     2     7
##  3     3     3
##  4     4     1
##  5     5     2
##  6     6     4
##  7     7     2
##  8     8     1
##  9     9     3
## 10    10     4
## # ℹ 409 more rows
```

```
# Visualization
# nba %>%
#   ggplot(aes(x = namePlayer)) +
#   geom_histogram()

nba %>%
  ggplot(aes(y = slugTeam)) +
  geom_bar()
```

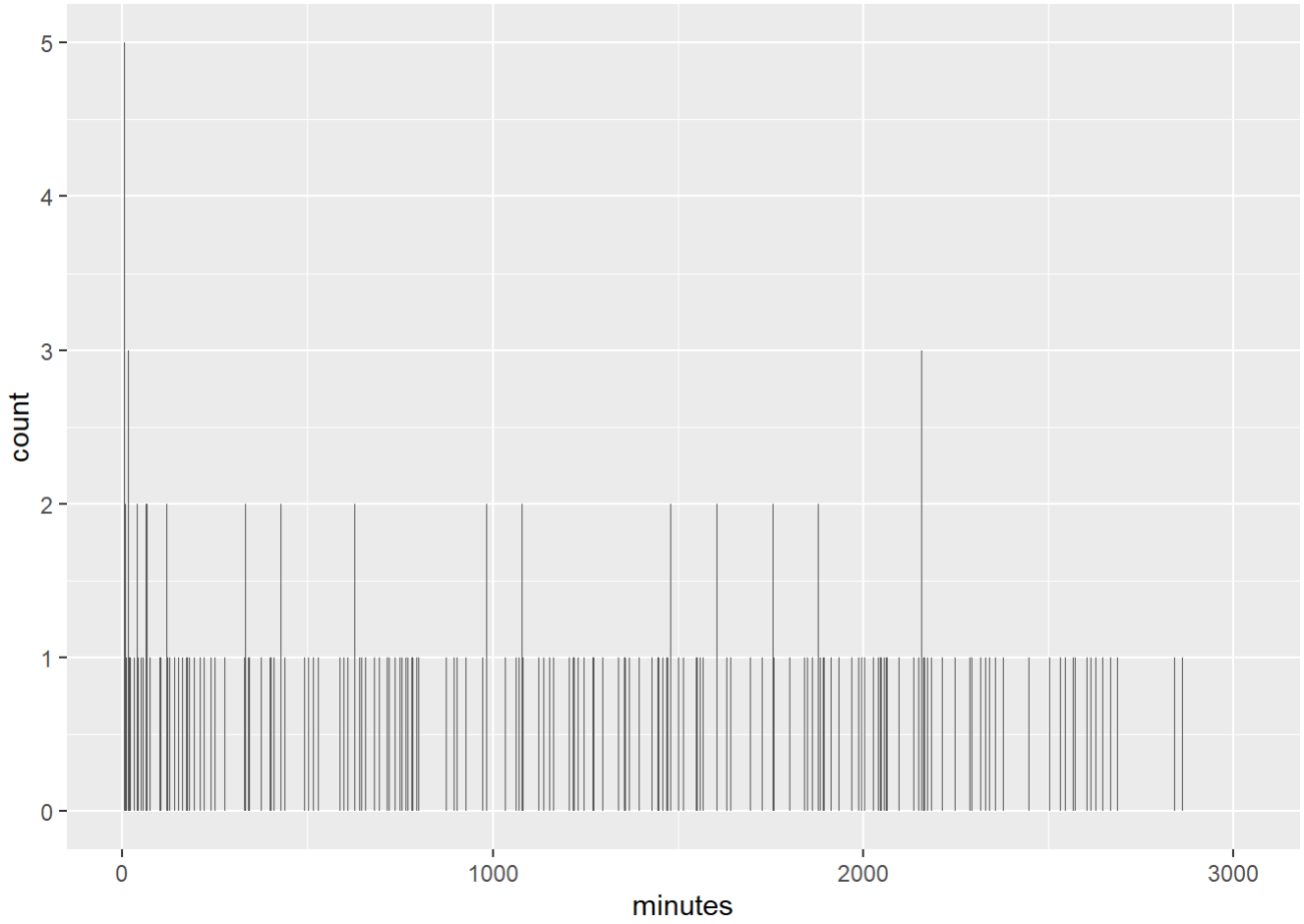



```
nba %>%
  count(slugTeam) %>%
  ggplot(aes(x = n,y = slugTeam)) +
  geom_bar(stat = 'identity')
```




```
# nba %>%
#   ggplot(aes(x = slugTeam)) +
#   geom_histogram()

nba %>%
  ggplot(aes(x = minutes)) +
  geom_bar()
```



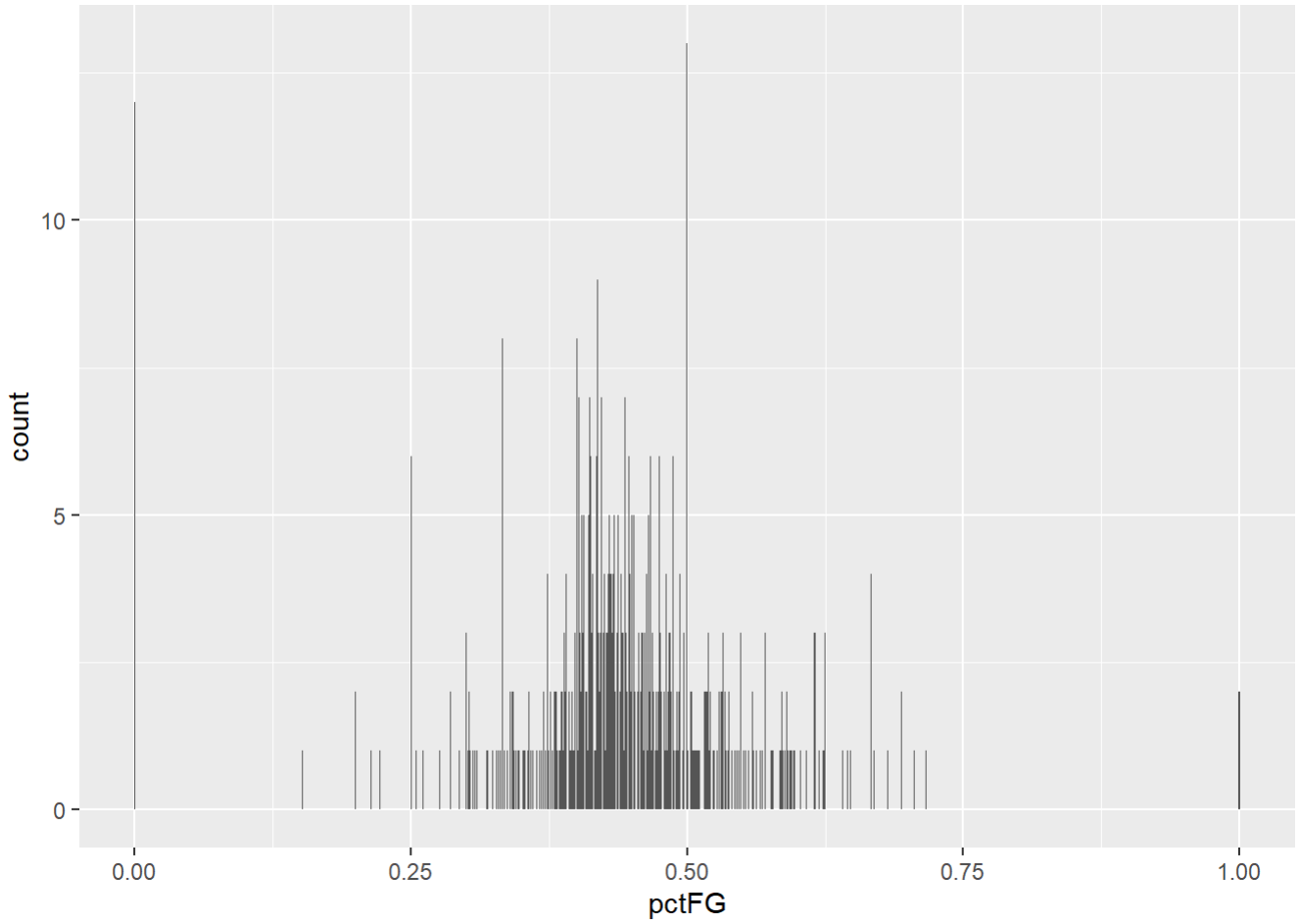
Applying the process

1. Look
2. Create (a plot)
3. Evaluate

```
nba %>%
  select(pctFG)
```

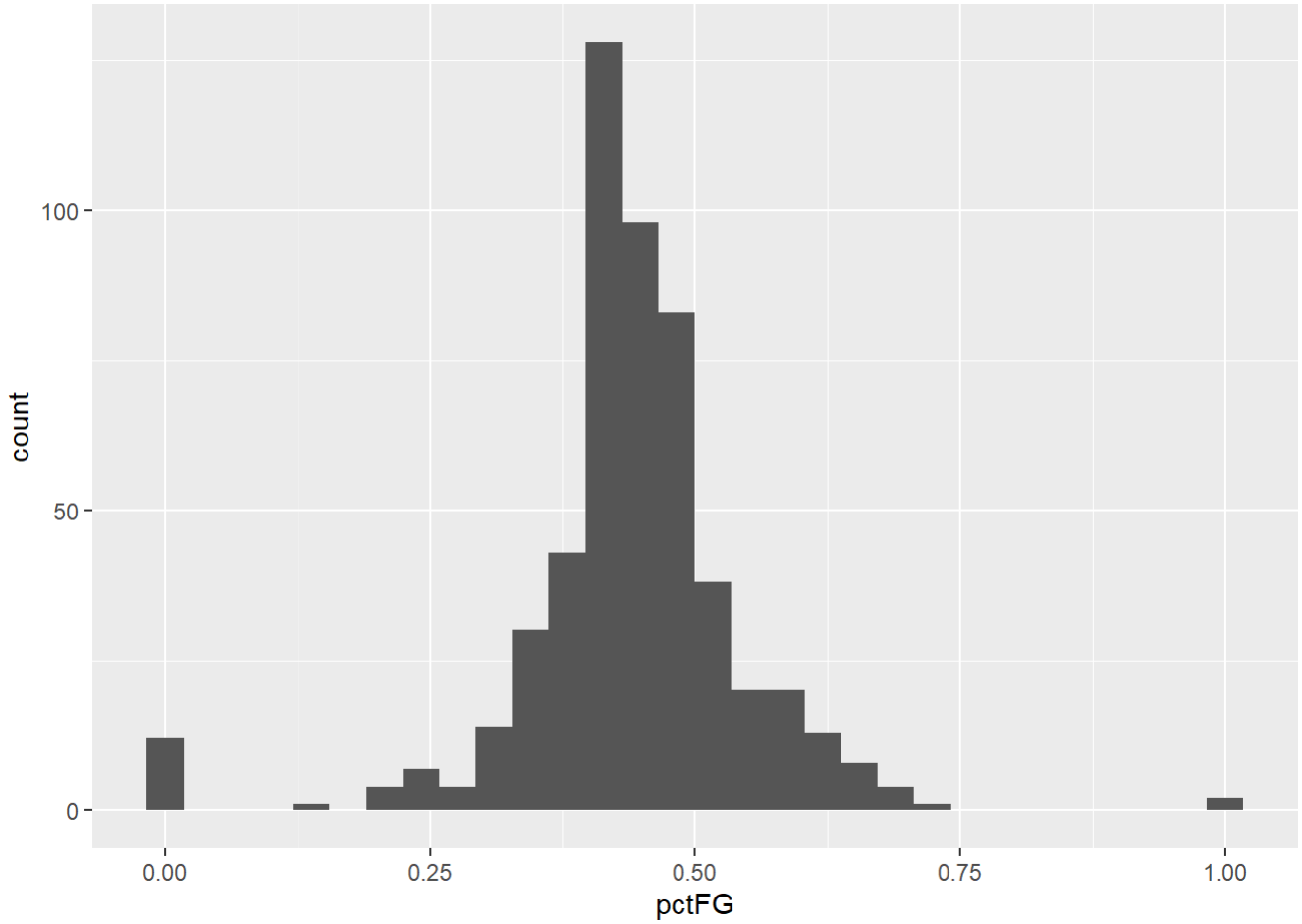
```
## # A tibble: 530 × 1
##    pctFG
##    <dbl>
##  1 0.519
##  2 0.222
##  3 0.595
##  4 0.357
##  5 0.576
##  6 0.333
##  7 0.376
##  8 0.306
##  9 0.345
## 10 0.3
## # ℹ 520 more rows
```

```
nba %>%
  ggplot(aes(x = pctFG)) +
  geom_bar()
```




```
nba %>%
  ggplot(aes(x = pctFG)) +
  geom_histogram()
```

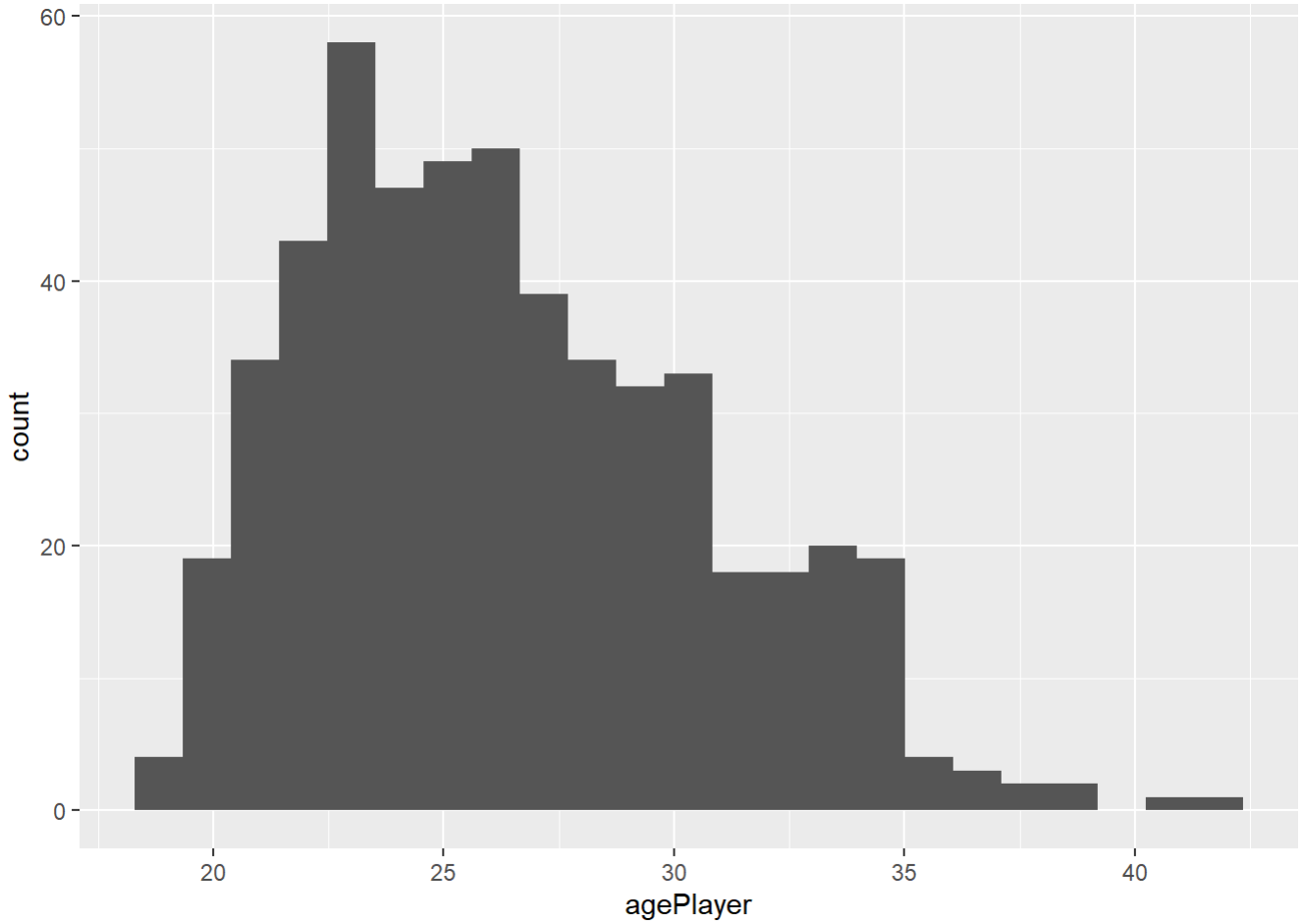
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



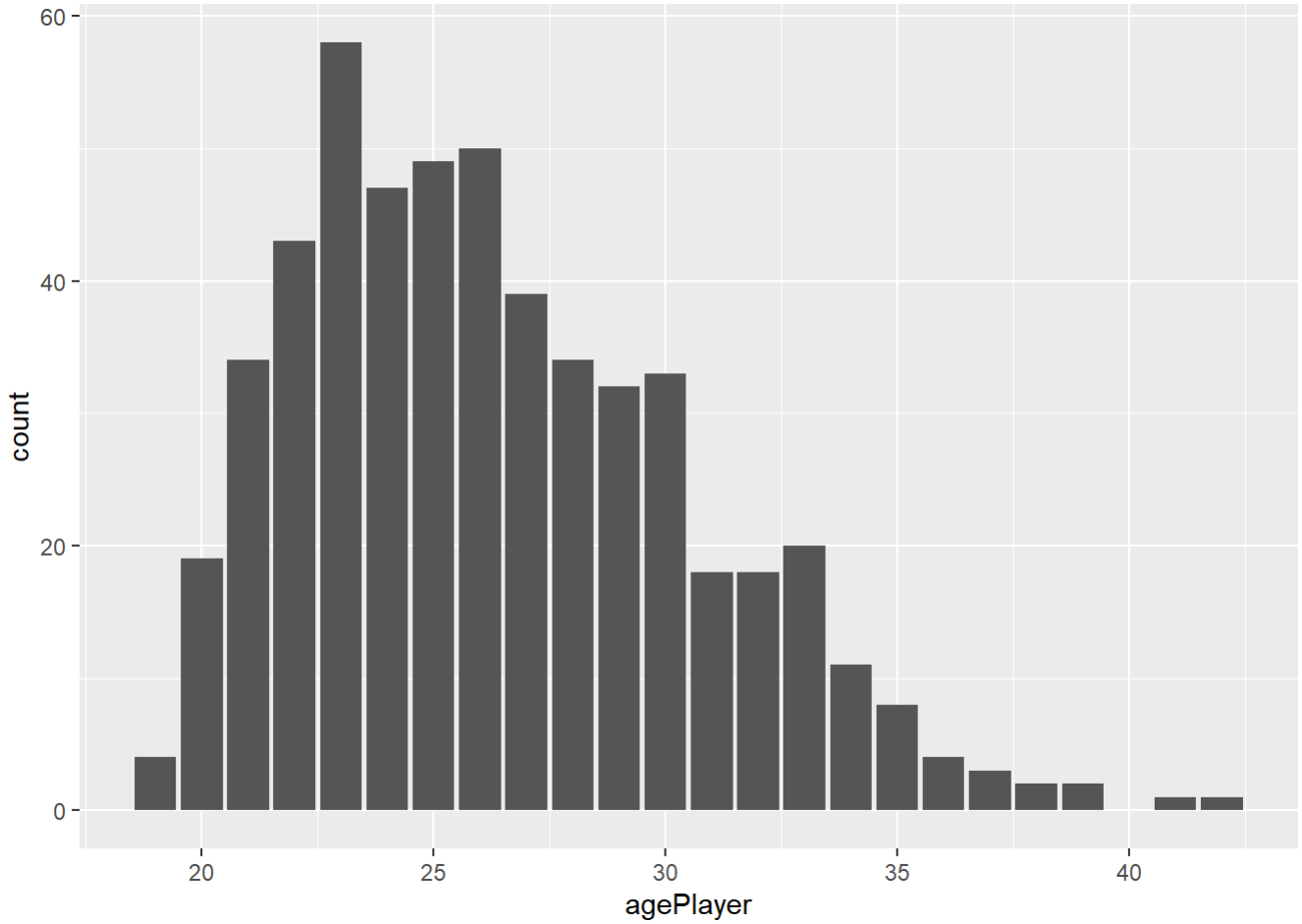

```
# Looking at age
nba %>%
  select(agePlayer)
```

```
## # A tibble: 530 × 1
##    agePlayer
##        <dbl>
##  1        33
##  2        28
##  3        25
##  4        25
##  5        21
##  6        21
##  7        23
##  8        22
##  9        23
## 10        26
## # ℹ 520 more rows
```

```
nba %>%
  ggplot(aes(x = agePlayer)) +
  geom_histogram(bins = 23)
```




```
nba %>%
  ggplot(aes(x = agePlayer)) +
  geom_bar()
```


60
40
20
0
count
20
25
30
35
40
agePlayer