Week 2.2 - Data Exploration and Summary Statistics

Import Merged NBA Game Data

```
library(tidyverse)
```

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

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

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

```
library(fastDummies)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
NBA_Games = read.csv("~/Google Drive/Sports Analytics Moocs/MOOC 1 - Foundations of sports analytics/We
head(NBA_Games)
```

```
##      CITY     TEAM_NAME    TEAM_ID NICKNAME   STATE YEAR_FOUNDED SEASON_ID
## 1 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 2 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 3 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 4 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 5 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 6 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22018
##   TEAM_ABBREVIATION    GAME_ID GAME_DATE     MATCHUP WL MIN PTS FGM FGA FG_PCT
## 1               ATL 1521900072 7/12/2019   ATL @ SAS  W 201  80  27  79  0.342
## 2               ATL 1521900060 7/11/2019   ATL @ WAS  L 200  71  26  68  0.382
## 3               ATL 1521900042  7/9/2019 ATL vs. IND  W 202  87  31  60  0.517
## 4               ATL 1521900023  7/7/2019 ATL vs. MIN  L 178  60  18  62  0.290
## 5               ATL 1521900013  7/6/2019   ATL @ MIL  L 201  83  25  73  0.342
## 6               ATL   21801220 4/10/2019 ATL vs. IND  L 240 134  43 103  0.417
##   FG3M FG3A FG3_PCT FTM FTA FT_PCT OREB DREB REB AST STL BLK TOV PF PLUS_MINUS
## 1    9   32   0.281  17  20  0.850   13   23  36  14  15   3  12 24        8.0
## 2   12   29   0.414   7  10  0.700    9   28  37  19  10   8  22 25       -5.0
## 3    8   21   0.381  17  24  0.708    7   27  34  17   5   5  18 21       18.2
## 4    4   22   0.182  20  32  0.625    9   27  36   7   7  10  18 28      -24.0
## 5   10   32   0.313  23  26  0.885    9   30  39  13  11   6  13 21        2.0
## 6   17   41   0.415  31  38  0.816   22   39  61  29   5   7  17 25       -1.0
```

Explore the dataset

Qualitative (Categorical) vs. Quantitative (Numerical) Data

– To assess the variable type in R, we use the "str" command.

- chr: qualitative variable – variables that are not in numerical form
- int: quantitative & continuous – real numbers that may not contain decimal points
- num: quantitative & continuous – real numbers that may contain decimal points

```
str(NBA_Games)
```

```
## 'data.frame':    18414 obs. of  32 variables:
##  $ CITY             : chr  "Atlanta" "Atlanta" "Atlanta" "Atlanta" ...
##  $ TEAM_NAME        : chr  "Atlanta Hawks" "Atlanta Hawks" "Atlanta Hawks" "Atlanta Hawks" ...
##  $ TEAM_ID          : int  1610612737 1610612737 1610612737 1610612737 1610612737 1610612737 161061273
##  $ NICKNAME         : chr  "Hawks" "Hawks" "Hawks" "Hawks" ...
##  $ STATE            : chr  "Atlanta" "Atlanta" "Atlanta" "Atlanta" ...
##  $ YEAR_FOUNDED     : int  1949 1949 1949 1949 1949 1949 1949 1949 1949 1949 ...
##  $ SEASON_ID        : int  22019 22019 22019 22019 22019 22018 22018 22018 22018 22018 ...
##  $ TEAM_ABBREVIATION: chr  "ATL" "ATL" "ATL" "ATL" ...
##  $ GAME_ID          : int  1521900072 1521900060 1521900042 1521900023 1521900013 21801220 21801202 2
##  $ GAME_DATE        : chr  "7/12/2019" "7/11/2019" "7/9/2019" "7/7/2019" ...
##  $ MATCHUP          : chr  "ATL @ SAS" "ATL @ WAS" "ATL vs. IND" "ATL vs. MIN" ...
##  $ WL               : chr  "W" "L" "W" "L" ...
##  $ MIN              : int  201 200 202 178 201 240 240 240 240 240 ...
##  $ PTS              : int  80 71 87 60 83 134 107 113 130 111 ...
##  $ FGM              : int  27 26 31 18 25 43 40 41 48 43 ...
##  $ FGA              : int  79 68 60 62 73 103 100 94 92 94 ...
##  $ FG_PCT           : num  0.342 0.382 0.517 0.29 0.342 0.417 0.4 0.436 0.522 0.457 ...
##  $ FG3M             : int  9 12 8 4 10 17 17 10 12 12 ...
##  $ FG3A             : int  32 29 21 22 32 41 45 39 36 34 ...
##  $ FG3_PCT          : num  0.281 0.414 0.381 0.182 0.313 0.415 0.378 0.256 0.333 0.353 ...
##  $ FTM              : int  17 7 17 20 23 31 10 21 22 13 ...
##  $ FTA              : int  20 10 24 32 26 38 19 31 28 20 ...
##  $ FT_PCT           : num  0.85 0.7 0.708 0.625 0.885 0.816 0.526 0.677 0.786 0.65 ...
##  $ OREB             : int  13 9 7 9 9 22 9 10 11 11 ...
##  $ DREB             : int  23 28 27 27 30 39 39 28 33 32 ...
##  $ REB              : int  36 37 34 36 39 61 48 38 44 43 ...
##  $ AST              : int  14 19 17 7 13 29 25 21 29 26 ...
##  $ STL              : int  15 10 5 7 11 5 2 16 7 13 ...
##  $ BLK              : int  3 8 5 10 6 7 3 4 7 2 ...
##  $ TOV              : int  12 22 18 18 13 17 11 14 11 11 ...
##  $ PF               : int  24 25 21 28 21 25 28 21 26 18 ...
##  $ PLUS_MINUS       : num  8 -5 18.2 -24 2 -1 -8 -36 8 -6 ...
```

In data analysis, we often convert categorical variable into dummy variable, if the observation belongs to the specified category, the dummy variable indicating the category would equal to 1, otherwise it equals to 0.

Convert a categorical variable to a dummy variable

The variable "WL" only carries two values, win or lose. We will create dummy variables to capture the categories.

We can use the "dummy_cols" function in the fastDummies library to convert a categorical variable to dummy variable. This function will also omit any missing value.

```r
dummy = dummy_cols(NBA_Games, select_columns = 'WL')
```

```r
colnames(dummy)
```

```
##  [1] "CITY"              "TEAM_NAME"         "TEAM_ID"
##  [4] "NICKNAME"          "STATE"             "YEAR_FOUNDED"
##  [7] "SEASON_ID"         "TEAM_ABBREVIATION" "GAME_ID"
## [10] "GAME_DATE"         "MATCHUP"           "WL"
## [13] "MIN"               "PTS"               "FGM"
## [16] "FGA"               "FG_PCT"            "FG3M"
## [19] "FG3A"              "FG3_PCT"           "FTM"
## [22] "FTA"               "FT_PCT"            "OREB"
## [25] "DREB"              "REB"               "AST"
## [28] "STL"               "BLK"               "TOV"
## [31] "PF"                "PLUS_MINUS"        "WL_L"
## [34] "WL_W"
```

Notice that two variables are created, WL_L and WL_W. WL_L=1 if the team lost and WL_L=0 if the team won. The original variable WL is deleted.

```r

NBA_Games = cbind(NBA_Games, dummy$WL_W)
head(NBA_Games)
```

We can attach the "WL_W" dummy variable back to our NBA_Games dataset using the cbind function.

```
##      CITY     TEAM_NAME    TEAM_ID NICKNAME   STATE YEAR_FOUNDED SEASON_ID
## 1 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 2 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 3 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 4 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 5 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 6 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22018
##   TEAM_ABBREVIATION    GAME_ID GAME_DATE     MATCHUP WL MIN PTS FGM FGA FG_PCT
## 1               ATL 1521900072 7/12/2019   ATL @ SAS  W 201  80  27  79  0.342
## 2               ATL 1521900060 7/11/2019   ATL @ WAS  L 200  71  26  68  0.382
## 3               ATL 1521900042  7/9/2019 ATL vs. IND  W 202  87  31  60  0.517
## 4               ATL 1521900023  7/7/2019 ATL vs. MIN  L 178  60  18  62  0.290
## 5               ATL 1521900013  7/6/2019   ATL @ MIL  L 201  83  25  73  0.342
## 6               ATL   21801220 4/10/2019 ATL vs. IND  L 240 134  43 103  0.417
##   FG3M FG3A FG3_PCT FTM FTA FT_PCT OREB DREB REB AST STL BLK TOV PF PLUS_MINUS
## 1    9   32   0.281  17  20  0.850   13   23  36  14  15   3  12 24        8.0
## 2   12   29   0.414   7  10  0.700    9   28  37  19  10   8  22 25       -5.0
## 3    8   21   0.381  17  24  0.708    7   27  34  17   5   5  18 21       18.2
## 4    4   22   0.182  20  32  0.625    9   27  36   7   7  10  18 28      -24.0
## 5   10   32   0.313  23  26  0.885    9   30  39  13  11   6  13 21        2.0
## 6   17   41   0.415  31  38  0.816   22   39  61  29   5   7  17 25       -1.0
##   dummy$WL_W
## 1          1
## 2          0
## 3          1
## 4          0
## 5          0
## 6          0
```

```
NBA_Games = NBA_Games %>% rename('WIN' = 'dummy$WL_W')
head(NBA_Games)
```

Rename “WL_W” to “WIN”

```
##      CITY     TEAM_NAME    TEAM_ID NICKNAME   STATE YEAR_FOUNDED SEASON_ID
## 1 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 2 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 3 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 4 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 5 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22019
## 6 Atlanta Atlanta Hawks 1610612737    Hawks Atlanta         1949     22018
##   TEAM_ABBREVIATION    GAME_ID GAME_DATE     MATCHUP WL MIN PTS FGM FGA FG_PCT
## 1               ATL 1521900072 7/12/2019   ATL @ SAS  W 201  80  27  79  0.342
## 2               ATL 1521900060 7/11/2019   ATL @ WAS  L 200  71  26  68  0.382
## 3               ATL 1521900042  7/9/2019 ATL vs. IND  W 202  87  31  60  0.517
## 4               ATL 1521900023  7/7/2019 ATL vs. MIN  L 178  60  18  62  0.290
## 5               ATL 1521900013  7/6/2019   ATL @ MIL  L 201  83  25  73  0.342
## 6               ATL   21801220 4/10/2019 ATL vs. IND  L 240 134  43 103  0.417
##   FG3M FG3A FG3_PCT FTM FTA FT_PCT OREB DREB REB AST STL BLK TOV PF PLUS_MINUS
## 1    9   32   0.281  17  20  0.850   13   23  36  14  15   3  12 24        8.0
## 2   12   29   0.414   7  10  0.700    9   28  37  19  10   8  22 25       -5.0
## 3    8   21   0.381  17  24  0.708    7   27  34  17   5   5  18 21       18.2
## 4    4   22   0.182  20  32  0.625    9   27  36   7   7  10  18 28      -24.0
## 5   10   32   0.313  23  26  0.885    9   30  39  13  11   6  13 21        2.0
## 6   17   41   0.415  31  38  0.816   22   39  61  29   5   7  17 25       -1.0
##   WIN
## 1   1
## 2   0
## 3   1
## 4   0
## 5   0
## 6   0
```

Working with date variable

In sports, we often have to work with date and time data.

```
typeof(NBA_Games$GAME_DATE)
```

```
## [1] "character"
```

The date variable is originally stored as a character In this case, each date is treated equally without ordering.

```
NBA_Games$GAME_DATE = mdy(NBA_Games$GAME_DATE)
head(NBA_Games$GAME_DATE)
```

Since the GAME_DATE variable is currently in the month/day/year format, we can use the “mdy” command from the lubridate package to convert the character variable to a date variable.

```
## [1] "2019-07-12" "2019-07-11" "2019-07-09" "2019-07-07" "2019-07-06"
## [6] "2019-04-10"
```

Descriptive and Summary Analyses

Summarize numerical data

We can use the “summary()” command to calculate summary statistics. This will return basic summary statistics for all the numerical variables which include the average, min and max, median, and the first and third quartiles of the values of the variable. For date variable, it summarizes the start and end dates of the dataset. You will see that for non-numerical variables, it only provides the number of values in the variable.

```
summary(NBA_Games)
```

```
##      CITY            TEAM_NAME            TEAM_ID            NICKNAME
##  Length:18414       Length:18414       Min.   :1.611e+09   Length:18414
##  Class :character   Class :character   1st Qu.:1.611e+09   Class :character
##  Mode  :character   Mode  :character   Median :1.611e+09   Mode  :character
##                                        Mean   :1.611e+09
##                                        3rd Qu.:1.611e+09
##                                        Max.   :1.611e+09
##     STATE            YEAR_FOUNDED    SEASON_ID     TEAM_ABBREVIATION
##  Length:18414       Min.   :1946   Min.   :12013   Length:18414
##  Class :character   1st Qu.:1949   1st Qu.:22014   Class :character
##  Mode  :character   Median :1970   Median :22015   Mode  :character
##                     Mean   :1970   Mean   :22651
##                     3rd Qu.:1980   3rd Qu.:22017
##                     Max.   :2002   Max.   :42018
##     GAME_ID            GAME_DATE            MATCHUP               WL
##  Min.   :1.130e+07   Min.   :2013-03-08   Length:18414       Length:18414
##  1st Qu.:2.140e+07   1st Qu.:2014-12-03   Class :character   Class :character
##  Median :2.160e+07   Median :2016-03-28   Mode  :character   Mode  :character
##  Mean   :1.210e+08   Mean   :2016-05-21
##  3rd Qu.:2.180e+07   3rd Qu.:2017-12-26
##  Max.   :1.622e+09   Max.   :2019-07-15
##       MIN             PTS             FGM             FGA
##  Min.   :170.0   Min.   : 47.0   Min.   :15.00   Min.   : 46.00
##  1st Qu.:239.0   1st Qu.: 93.0   1st Qu.:34.00   1st Qu.: 79.00
##  Median :240.0   Median :102.0   Median :38.00   Median : 84.00
##  Mean   :239.1   Mean   :102.4   Mean   :37.93   Mean   : 83.94
##  3rd Qu.:241.0   3rd Qu.:112.0   3rd Qu.:42.00   3rd Qu.: 89.00
##  Max.   :341.0   Max.   :168.0   Max.   :61.00   Max.   :129.00
##      FG_PCT            FG3M             FG3A          FG3_PCT
##  Min.   :0.2170   Min.   : 0.000   Min.   : 3.00   Min.   :0.0000
##  1st Qu.:0.4140   1st Qu.: 6.000   1st Qu.:20.00   1st Qu.:0.2860
##  Median :0.4520   Median : 9.000   Median :25.00   Median :0.3490
##  Mean   :0.4525   Mean   : 9.033   Mean   :25.63   Mean   :0.3506
##  3rd Qu.:0.4890   3rd Qu.:11.000   3rd Qu.:30.00   3rd Qu.:0.4170
##  Max.   :0.6840   Max.   :28.000   Max.   :70.00   Max.   :0.8420
##       FTM             FTA              FT_PCT          OREB
##  Min.   : 1.00   Min.   : 1.00   Min.   :0.1430   Min.   : 0.0
##  1st Qu.:13.00   1st Qu.:18.00   1st Qu.:0.6920   1st Qu.: 8.0
##  Median :17.00   Median :23.00   Median :0.7650   Median :10.0
##  Mean   :17.52   Mean   :23.14   Mean   :0.7583   Mean   :10.4
##  3rd Qu.:21.00   3rd Qu.:28.00   3rd Qu.:0.8330   3rd Qu.:13.0
##  Max.   :52.00   Max.   :64.00   Max.   :1.0000   Max.   :38.0
##       DREB            REB             AST             STL
##  Min.   :11.00   Min.   :17.00   Min.   : 2.00   Min.   : 0.000
##  1st Qu.:29.00   1st Qu.:39.00   1st Qu.:18.00   1st Qu.: 6.000
##  Median :33.00   Median :43.00   Median :22.00   Median : 8.000  
##  Mean   :32.82   Mean   :43.21   Mean   :22.08   Mean   : 7.789  
##  3rd Qu.:37.00   3rd Qu.:48.00   3rd Qu.:26.00   3rd Qu.:10.000  
##  Max.   :56.00   Max.   :81.00   Max.   :47.00   Max.   :23.000  
##       BLK              TOV              PF          PLUS_MINUS     
##  Min.   : 0.000   Min.   : 2.00   Min.   : 7.00   Min.   :-61.00000  
##  1st Qu.: 3.000   1st Qu.:11.00   1st Qu.:18.00   1st Qu.: -9.00000  
##  Median : 5.000   Median :14.00   Median :20.00   Median :  0.00000  
##  Mean   : 4.818   Mean   :13.96   Mean   :20.73   Mean   :  0.04403  
##  3rd Qu.: 6.000   3rd Qu.:16.00   3rd Qu.:24.00   3rd Qu.:  9.00000  
##  Max.   :19.000   Max.   :35.00   Max.   :45.00   Max.   : 62.00000  
##       WIN        
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.4999  
##  3rd Qu.:1.0000  
##  Max.   :1.0000
```

```

summary(NBA_Games$PTS)
```

We can summarize a single variable by specifying the variable.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    47.0    93.0   102.0   102.4   112.0   168.0
```

We can also calculate individual statistics by using the mean(), median(), sd().

- Calculate mean of a numerical variable

```
mean(NBA_Games$FGM)
```

```
## [1] 37.93087
```

- Calculate median of a numerical variable

```
median(NBA_Games$FGM)
```

```
## [1] 38
```

- Calculate standard deviation of a numerical variable

```
sd(NBA_Games$FGM)
```

```
## [1] 5.664956
```

Self Test

1. Find the mean of field goals attempted;
2. Find the median of 3-point field goals made;
3. Find the standard deviation of the number of rebounds

```
mean(NBA_Games$FGA)
```

```
## [1] 83.94119
```

```
median(NBA_Games$FG3M)
```

```
## [1] 9
```

```
sd(NBA_Games$REB)
```

```
## [1] 6.726882
```

We can also calculate the summary statistics of a variable based on another variable, usually based on a different categorical variable.

- Calculate means by groups using "groupby" command.

```
NBA_Games %>% group_by(WL) %>% summarise_all(list(mean), na.rm = TRUE) %>% ungroup()
```

```
## Warning in mean.default(CITY, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(CITY, na.rm = TRUE): argument is not numeric or logical:
## returning NA

## Warning in mean.default(TEAM_NAME, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(TEAM_NAME, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(NICKNAME, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(NICKNAME, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(STATE, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(STATE, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(TEAM_ABBREVIATION, na.rm = TRUE): argument is not
## numeric or logical: returning NA

## Warning in mean.default(TEAM_ABBREVIATION, na.rm = TRUE): argument is not
## numeric or logical: returning NA

## Warning in mean.default(MATCHUP, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## Warning in mean.default(MATCHUP, na.rm = TRUE): argument is not numeric or
## logical: returning NA

## # A tibble: 2 x 33
##   WL     CITY TEAM_NAME TEAM_ID NICKNAME STATE YEAR_FOUNDED SEASON_ID
##   <chr> <dbl>     <dbl>   <dbl>    <dbl> <dbl>        <dbl>     <dbl>
## 1 L        NA        NA  1.61e9       NA    NA        1969.    22663.
## 2 W        NA        NA  1.61e9       NA    NA        1970.    22640.
## # ... with 25 more variables: TEAM_ABBREVIATION <dbl>, GAME_ID <dbl>,
## #   GAME_DATE <date>, MATCHUP <dbl>, MIN <dbl>, PTS <dbl>, FGM <dbl>,
## #   FGA <dbl>, FG_PCT <dbl>, FG3M <dbl>, FG3A <dbl>, FG3_PCT <dbl>, FTM <dbl>,
## #   FTA <dbl>, FT_PCT <dbl>, OREB <dbl>, DREB <dbl>, REB <dbl>, AST <dbl>,
## #   STL <dbl>, BLK <dbl>, TOV <dbl>, PF <dbl>, PLUS_MINUS <dbl>, WIN <dbl>
```

- Calculate the mean of a single (points in this example) variable by group.

```
NBA_Games %>% group_by(WL) %>% summarise(mean_PTS = mean(PTS, na.rm = TRUE)) %>% ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   WL    mean_PTS
##   <chr>    <dbl>
## 1 L         96.8
## 2 W        108.
```

Summarize date variable

- We can find some basic statistics of the date variable. The describe() function returns the first and the last dates.

```
summary(NBA_Games$GAME_DATE)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2013-03-08" "2014-12-03" "2016-03-28" "2016-05-21" "2017-12-26" "2019-07-15"
```

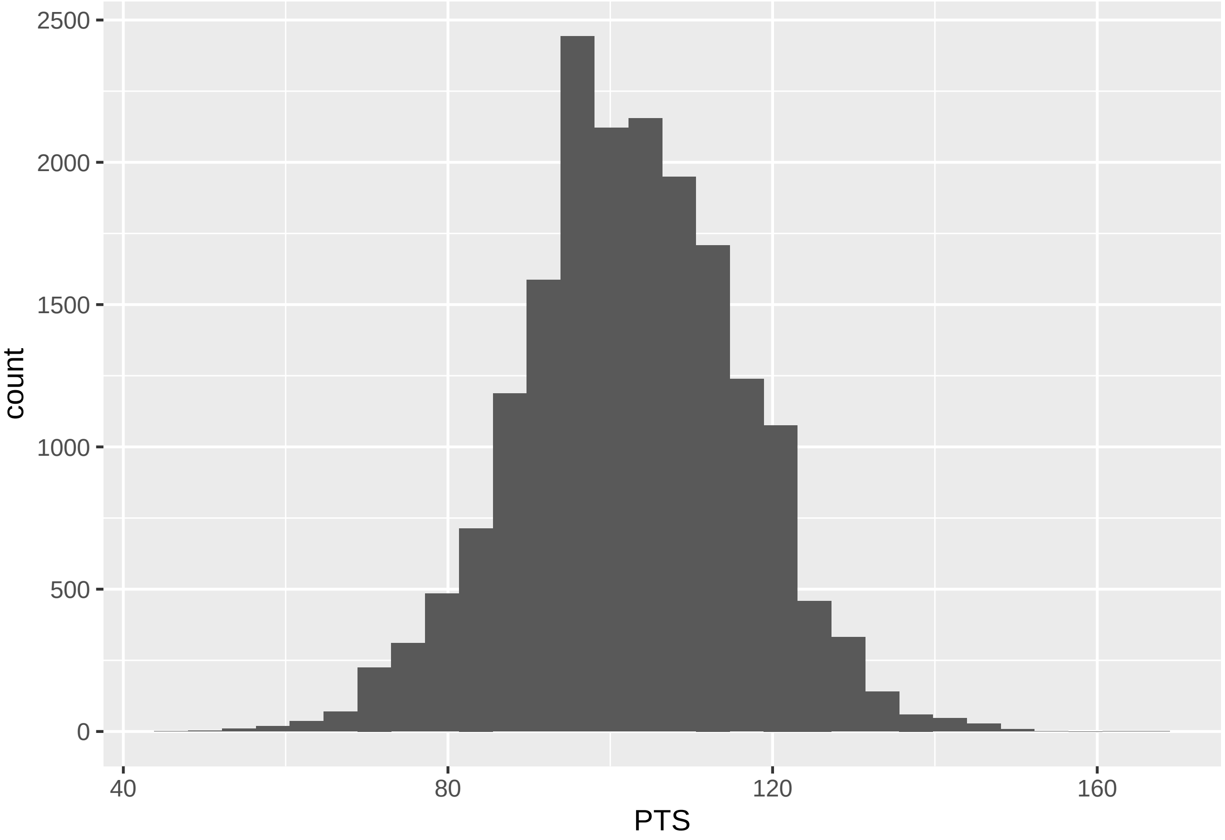
Visualizing data

Histogram

– We can visualize the distribution of a variable using a histogram.

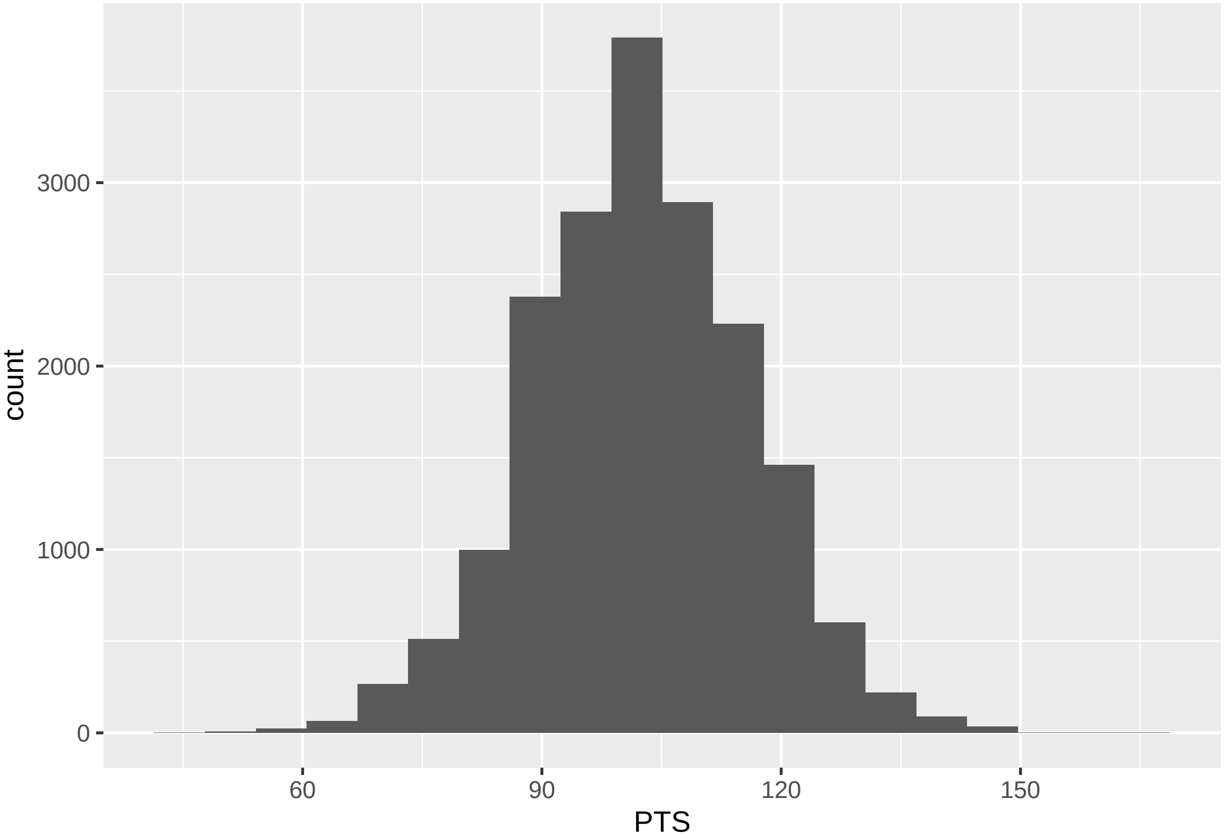
```
ggplot(NBA_Games, aes(x=PTS)) + geom_histogram()
```

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



We can specify the number of bins in a histogram; different numbers of bins may give us slightly different graphs.

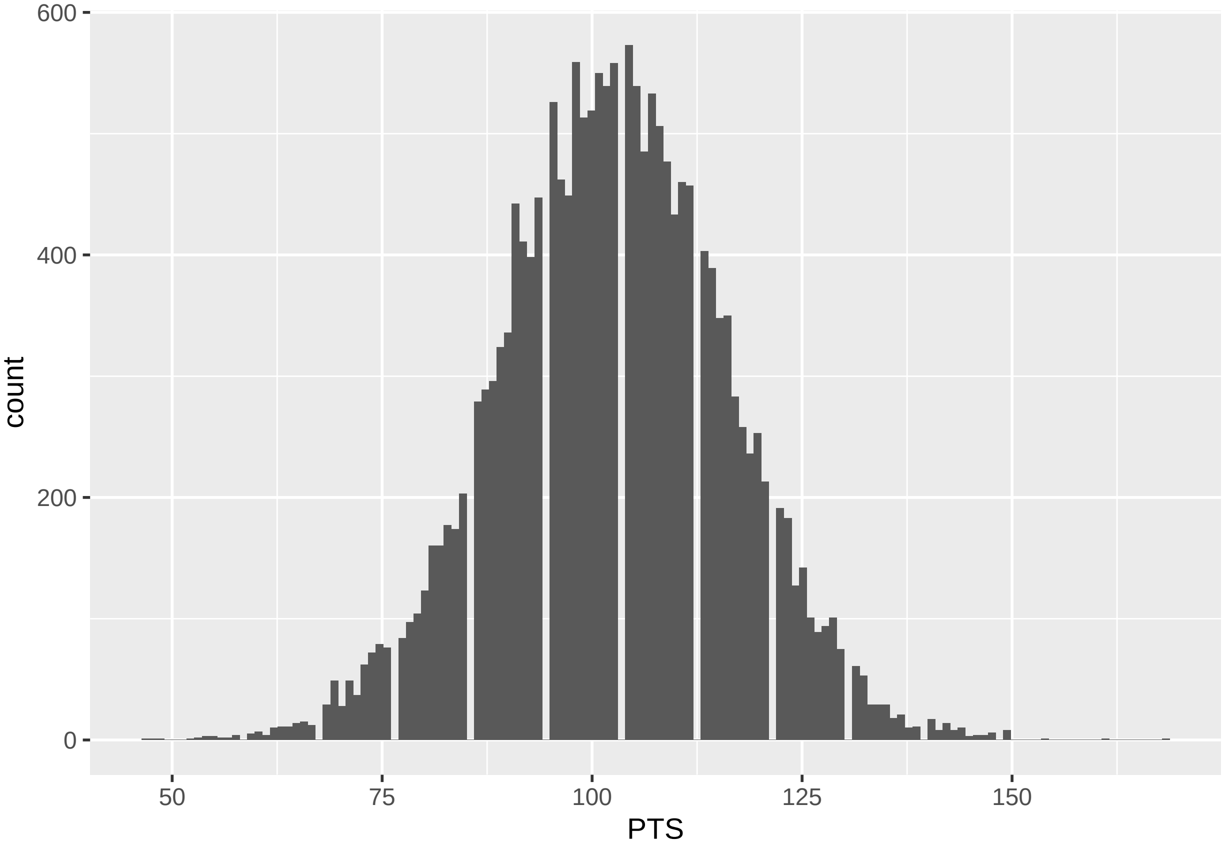
```r
ggplot(NBA_Games, aes(x=PTS)) + geom_histogram(bins=20)
```



For visual appeal, sometimes it may be helpful to add space between bins.

For example, we can narrow the bin to 0.9 width.

```
ggplot(NBA_Games, aes(x=PTS)) + geom_histogram(bins=20, binwidth = 0.9)
```



Save edited dataset

```
write.csv(NBA_Games, "NBA_Games2.csv", row.names=FALSE)
```