Supplement for Lecture 1: Model Basics

Read in Dataset

```
# Loads a package needed to use the read_csv() function
# CSV file needs to be in the same folder as your RMD file
# Default working directory is same as directory of your RMD file

lego1 <- read_csv("lego.csv")
```

```
## Rows: 1304 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): Set_Name, Theme, Ages, Packaging, Weight, Availability, Size
## dbl (8): Item_Number, Pieces, Price, Amazon_Price, Year, Pages, Minifigures,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# You can specify the full path on your computer like below
#lego2 = read_csv("D:/Mario Documents/UNC/STOR 455/STOR455_WEBSITE/Supplement/Model Basics/lego.csv")

# Created Objects lego1 and lego2 using <- and =
```

Notice that objects `lego1` and `lego2` have been added to the Global Environment.

Preview Dataset

```
head(lego1) #Vertically (Top 6 Rows by Default)
```

```
## # A tibble: 6 x 15
##   Item_Number Set_Name         Theme Pieces Price Amazon_Price  Year Ages  Pages
##         <dbl> <chr>            <chr>  <dbl> <dbl>        <dbl> <dbl> <chr> <dbl>
## 1       41916 Extra Dots - Se~ DOTS     109  3.99         3.44  2020 Ages~    NA
## 2       41908 Extra Dots - Se~ DOTS     109  3.99         3.99  2020 Ages~    NA
## 3       11006 Creative Blue B~ Clas~     52  4.99         4.93  2020 Ages~    37
## 4       11007 Creative Green ~ Clas~     60  4.99         4.93  2020 Ages~    37
## 5       41901 Funky Animals B~ DOTS      33  4.99         4.99  2020 Ages~    NA
## 6       41902 Sparkly Unicorn~ DOTS      33  4.99         4.99  2020 Ages~    NA
## # i 6 more variables: Minifigures <dbl>, Packaging <chr>, Weight <chr>,
## #   Unique_Pieces <dbl>, Availability <chr>, Size <chr>
```

```
str(lego1)  #Horizontally
```

```
## spc_tbl_ [1,304 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Item_Number  : num [1:1304] 41916 41908 11006 11007 41901 ...
##  $ Set_Name     : chr [1:1304] "Extra Dots - Series 2" "Extra Dots - Series 1" "Creative Blue Bricks
##  $ Theme        : chr [1:1304] "DOTS" "DOTS" "Classic" "Classic" ...
```

```
##  $ Pieces       : num [1:1304] 109 109 52 60 33 33 33 33 33 33 ...
##  $ Price        : num [1:1304] 3.99 3.99 4.99 4.99 4.99 4.99 4.99 4.99 4.99 4.99 ...
##  $ Amazon_Price : num [1:1304] 3.44 3.99 4.93 4.93 4.99 4.99 4.99 4.99 4.99 4.99 ...
##  $ Year         : num [1:1304] 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ Ages         : chr [1:1304] "Ages_6+" "Ages_6+" "Ages_4+" "Ages_4+" ...
##  $ Pages        : num [1:1304] NA NA 37 37 NA NA NA NA NA NA ...
##  $ Minifigures  : num [1:1304] NA NA NA NA NA NA NA NA NA NA ...
##  $ Packaging    : chr [1:1304] "Foil pack" "Foil pack" "Box" "Box" ...
##  $ Weight       : chr [1:1304] NA NA NA NA ...
##  $ Unique_Pieces: num [1:1304] 6 6 28 36 10 9 9 12 10 9 ...
##  $ Availability : chr [1:1304] "Retail" "Retail" "Retail" "Retail" ...
##  $ Size         : chr [1:1304] "Small" "Small" "Small" "Small" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Item_Number = col_double(),
##   ..   Set_Name = col_character(),
##   ..   Theme = col_character(),
##   ..   Pieces = col_double(),
##   ..   Price = col_double(),
##   ..   Amazon_Price = col_double(),
##   ..   Year = col_double(),
##   ..   Ages = col_character(),
##   ..   Pages = col_double(),
##   ..   Minifigures = col_double(),
##   ..   Packaging = col_character(),
##   ..   Weight = col_character(),
##   ..   Unique_Pieces = col_double(),
##   ..   Availability = col_character(),
##   ..   Size = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
names(lego1) #Get Vector of Variable Names
```

```
##  [1] "Item_Number"   "Set_Name"      "Theme"         "Pieces"
##  [5] "Price"         "Amazon_Price"  "Year"          "Ages"
##  [9] "Pages"         "Minifigures"   "Packaging"     "Weight"
## [13] "Unique_Pieces" "Availability"  "Size"
```

How would you show the top 12 rows instead of the top 6? You can use `?head` to access the documentation for the `head()` function.

Subsetting the Data

Datasets are Just Fancy Matrices

```
dim(lego1)
```

```
## [1] 1304   15
```

This dataset has 1,304 observations or cases and 15 variables or covariates. The variables n and p are typically used to represent the sample size and number of variables. Therefore we would say that $n = 1304$ and $p = 15$. To subset the dataset we can use the double brackets to select/deselect rows or columns

```
lego1[c(1,5, 10, 15, 20),]
```

```
## # A tibble: 5 x 15
##   Item_Number Set_Name         Theme Pieces Price Amazon_Price  Year Ages  Pages
##         <dbl> <chr>            <chr>  <dbl> <dbl>        <dbl> <dbl> <chr> <dbl>
## 1       41916 Extra Dots - Se~ DOTS     109  3.99         3.44  2020 Ages~    NA
## 2       41901 Funky Animals B~ DOTS      33  4.99         4.99  2020 Ages~    NA
## 3       41917 Magic Forest Br~ DOTS      33  4.99         4.99  2020 Ages~    NA
## 4       11010 White Baseplate  Clas~      1  7.99         7.86  2020 Ages~     1
## 5       60239 Police Patrol C~ City      92  9.99         8.35  2019 Ages~    36
## # i 6 more variables: Minifigures <dbl>, Packaging <chr>, Weight <chr>,
## #   Unique_Pieces <dbl>, Availability <chr>, Size <chr>
```

```
lego1[,c(1,5, 15)]
```

```
## # A tibble: 1,304 x 3
##    Item_Number Price Size
##          <dbl> <dbl> <chr>
##  1       41916  3.99 Small
##  2       41908  3.99 Small
##  3       11006  4.99 Small
##  4       11007  4.99 Small
##  5       41901  4.99 Small
##  6       41902  4.99 Small
##  7       41903  4.99 Small
##  8       41911  4.99 Small
##  9       41912  4.99 Small
## 10       41917  4.99 Small
## # i 1,294 more rows
```

```
lego1[,c("Item_Number","Price","Size")]
```

```
## # A tibble: 1,304 x 3
##    Item_Number Price Size
##          <dbl> <dbl> <chr>
##  1       41916  3.99 Small
##  2       41908  3.99 Small
##  3       11006  4.99 Small
##  4       11007  4.99 Small
##  5       41901  4.99 Small
##  6       41902  4.99 Small
##  7       41903  4.99 Small
##  8       41911  4.99 Small
##  9       41912  4.99 Small
## 10       41917  4.99 Small
## # i 1,294 more rows
```

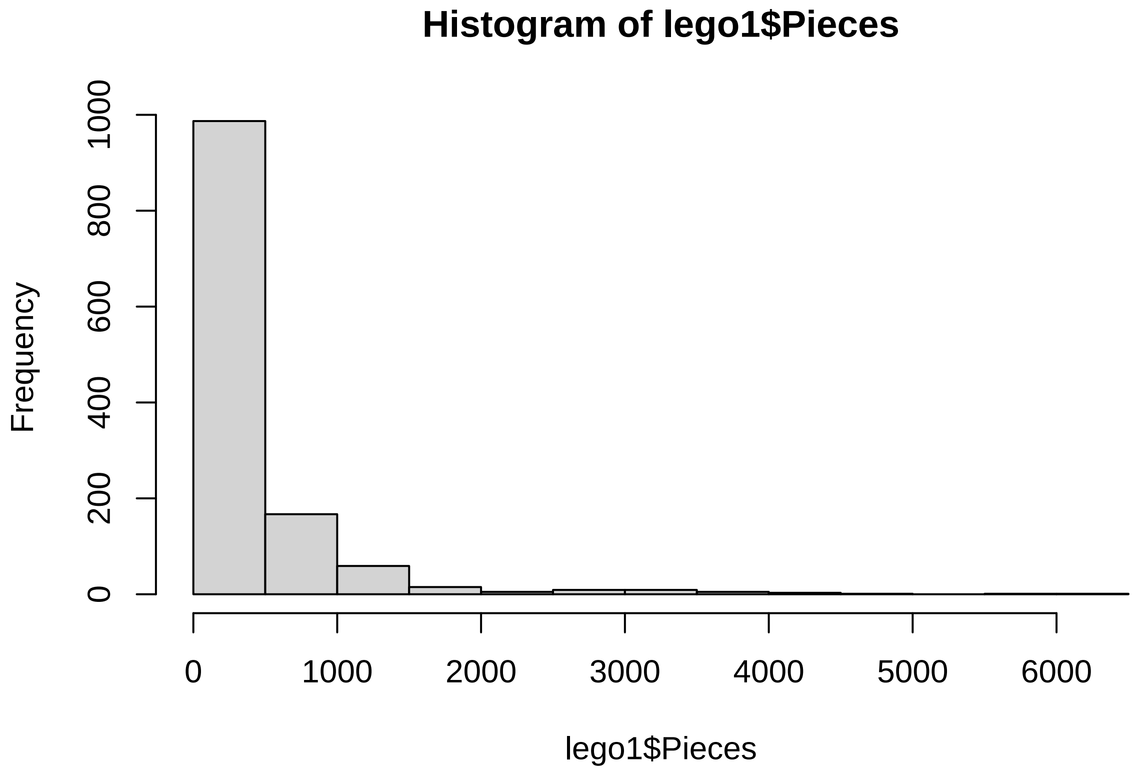
```
lego1[,names(lego1)[c(1,5,15)]]
```

```
## # A tibble: 1,304 x 3
##    Item_Number Price Size
##          <dbl> <dbl> <chr>
##  1       41916  3.99 Small
##  2       41908  3.99 Small
##  3       11006  4.99 Small
##  4       11007  4.99 Small
##  5       41901  4.99 Small
##  6       41902  4.99 Small
```

```
##  7       41903  4.99 Small
##  8       41911  4.99 Small
##  9       41912  4.99 Small
## 10       41917  4.99 Small
## # i 1,294 more rows
```

Subsetting Based Off Values

```
hist(lego1$Pieces)
```



```
lego1$Pieces>1000
```

```
##    [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
##  [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [469] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [481] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [505] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [517] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [529] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [553] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [565] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [577] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [589] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [613] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [637] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [673] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [685] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [697] FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
##  [709]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
##  [721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
##  [733] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
##  [745] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [757] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [769]  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [781] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
##  [793] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
##  [805] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
##  [817] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
```

```
##  [829] FALSE FALSE FALSE FALSE FALSE    NA FALSE    NA FALSE FALSE FALSE    NA
##  [841] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [853] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [865] FALSE    NA FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE    NA
##  [877] FALSE FALSE FALSE FALSE FALSE FALSE    NA    NA FALSE FALSE FALSE FALSE
##  [889] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE    NA    NA FALSE FALSE
##  [901] FALSE    NA FALSE FALSE FALSE    NA FALSE FALSE FALSE FALSE FALSE    NA
##  [913] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [925] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [937] FALSE FALSE FALSE FALSE    NA FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [949] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [961] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [973] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [985] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [997] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1009] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1021] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1033] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1045] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1057] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1069] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1081] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1093] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1105] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1117] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1129] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## [1141]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
## [1153] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
## [1165]  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1177]  TRUE  TRUE  TRUE    NA FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1189] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [1201]    NA FALSE FALSE FALSE FALSE FALSE FALSE FALSE    NA    NA    NA    NA
## [1213]    NA    NA FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE    NA FALSE
## [1225] FALSE FALSE FALSE FALSE FALSE FALSE    NA FALSE    NA    NA    NA  TRUE
## [1237] FALSE FALSE FALSE FALSE    NA    NA    NA    NA    NA    NA    NA    NA
## [1249] FALSE FALSE    NA FALSE    NA    NA    NA FALSE FALSE  TRUE FALSE  TRUE
## [1261] FALSE FALSE  TRUE    NA FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## [1273] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1285]    NA FALSE    NA FALSE FALSE FALSE FALSE FALSE    NA FALSE  TRUE  TRUE
## [1297] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
```

```r
large_lego = lego1[lego1$Pieces>2000,]

small_lego = lego1[lego1$Pieces< 15,]

middle_lego=lego1[lego1$Pieces>15 & lego1$Pieces<2000,]

extreme_lego=lego1[lego1$Pieces<15 & lego1$Pieces>2000,]
```

Mathematical/Statistical Functions

Mathematical Calculations

```
#Scalars
a=3
b=4

a+b
```

```
## [1] 7
```

```
a-b
```

```
## [1] -1
```

```
a*b
```

```
## [1] 12
```

```
a/b
```

```
## [1] 0.75
```

```
a^2
```

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

```
log(a) #Defaults to natural log
```

```
## [1] 1.098612
```

```
exp(a) # e^a
```

```
## [1] 20.08554
```

```
#Vectors
a=c(1,2,3,4)
b=c(2,3,4,5)

a+b
```

```
## [1] 3 5 7 9
```

```
a-b
```

```
## [1] -1 -1 -1 -1
```

```
a*b
```

```
## [1]  2  6 12 20
```

```
a/b
```

```
## [1] 0.5000000 0.6666667 0.7500000 0.8000000
```

```
a^2
```

```
## [1]  1  4  9 16
```

```
log(a)
```

```
## [1] 0.0000000 0.6931472 1.0986123 1.3862944
```

```
exp(a)
```

```
## [1]  2.718282  7.389056 20.085537 54.598150
```

```
#Combination (Based off Exercise 0.15: Roller Coasters)
yint=54
slope=7.6
TypeCode=c(0,1) #0=Wooden & 1=Steel

TopSpeed = yint+slope*TypeCode
TopSpeed
```

```
## [1] 54.0 61.6
```

Statistical Functions

```
mean(lego1$Amazon_Price,na.rm=T)
```

```
## [1] 57.8232
```

```
median(lego1$Amazon_Price,na.rm=T)
```

```
## [1] 37.325
```

```
sd(lego1$Amazon_Price,na.rm=T)
```

```
## [1] 66.26777
```

```
var(lego1$Amazon_Price,na.rm=T)
```

```
## [1] 4391.417
```

```
IQR(lego1$Amazon_Price,na.rm=T)
```

```
## [1] 50
```

```
quantile(lego1$Amazon_Price,na.rm=T)
```

```
##      0%     25%     50%     75%    100% 
##   3.440  19.950  37.325  69.950 699.950
```

```
quantile(lego1$Amazon_Price,c(0.05,0.1,0.9,0.95),na.rm=T)
```

```
##       5%      10%      90%      95% 
##   9.0350  13.9250 128.9700 179.9825
```

```
unique(lego1$Theme)
```

```
##  [1] "DOTS"                 "Classic"              "DUPLO"               
##  [4] "Friends"              "Disney"               "City"                
##  [7] "Unikitty!"            "NINJAGO"              "Star Wars"           
## [10] "Minecraft"            "Marvel"               "Creator 3-in-1"      
## [13] "Batman"               "THE LEGO MOVIE 2"     "Technic"             
## [16] "Speed Champions"      "BrickHeadz"           NA                    
## [19] "LEGO Frozen 2"        "LEGO Super Mario"     "Harry Potter"        
## [22] "Hidden Side"          "Trolls World Tour"    "Minions"             
## [25] "Powerpuff Girls"      "Jurassic World"       "Overwatch"           
## [28] "Spider-Man"           "Juniors"              "DC"                  
## [31] "Architecture"         "Ideas"                "Creator Expert"      
## [34] "LEGO Art"             "Powered UP"           "Stranger Things"     
## [37] "Monkie Kid"           "Xtra"                 "Minifigures"         
## [40] "LEGO Brick Sketches" "LEGO Education"
```

```
table(lego1$Theme)
```

```
##
##         Architecture               Batman           BrickHeadz                 City
##                   11                   16                   43                  101
##              Classic       Creator 3-in-1       Creator Expert                   DC
##                   21                   38                   15                   12
##               Disney                 DOTS                DUPLO              Friends
##                   46                   18                   53                  103
##         Harry Potter          Hidden Side                Ideas              Juniors
##                   27                   19                   12                   12
##       Jurassic World             LEGO Art  LEGO Brick Sketches       LEGO Education
##                   20                    4                    2                    5
##        LEGO Frozen 2     LEGO Super Mario               Marvel            Minecraft
##                    8                   17                   50                   26
##          Minifigures              Minions           Monkie Kid              NINJAGO
##                    8                    2                   12                   78
##            Overwatch           Powered UP      Powerpuff Girls      Speed Champions
##                    8                   14                    2                   18
##           Spider-Man            Star Wars     Stranger Things              Technic
##                    2                  119                    1                   38
##      THE LEGO MOVIE 2    Trolls World Tour             Unikitty!                 Xtra
##                   28                    9                    8                    8
```

```
table(lego1$Size,lego1$Availability)
```

```
##
##         Educational LEGO exclusive LEGOLAND exclusive Not sold Promotional
##   Large           0              0                  0        0           0
##   Small           4             64                  0        0           3
##
##         Retail Retail - limited
##   Large     34                0
##   Small    766               65
```

```
na_rm_lego1 = na.omit(lego1)
cor(na_rm_lego1$Amazon_Price,na_rm_lego1$Price)
```

```
## [1] 0.86657
```

```
lm(Amazon_Price ~ Price, data=na_rm_lego1)
```

```
##
## Call:
## lm(formula = Amazon_Price ~ Price, data = na_rm_lego1)
##
## Coefficients:
## (Intercept)        Price
##       9.219        1.151
```