Do Stats Make Competitive Pokemon?

Preamble

This research focuses on something dear to most people under 30's hearts: Pokemon. First and foremost, let's talk about what exactly Pokemon is. Pokemon is a game, where you capture and battle with Pokemon, who come with an assortment of types, abilities, moves, and stats. Let's look at an example of a Pokemon from our data:

```
## # A tibble: 1 x 15
##    ...1 dexno pokename Ability     hp   Atk   Def  SAtk  SDef   Spd generation
##   <dbl> <dbl> <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl>
## 1   270   245 Suicune  Pressure   100    75   115    90   115    85          2
## # ... with 4 more variables: Rate <dbl>, Real <dbl>, bst <dbl>, genno <fct>
```

This is our first Pokemon, Lotad: We can see it's stats above. The dexno is its number in the pokedex, the list that compiles all Pokemon. Its dexno is 270, meaning it hold the 270th position in the pokedex. Next, we see its ability, which is a special trait that has an impact in game - we won't look at this at all, but it will affect a Pokemon's strength. The next 5 numbers are its stats. HP is hitpoints, how much "life" it has. Atk is how it attacks using physical moves. Def is its defense against phsycial moves. SAtk and SDef are the special counterparts to these stats. lastly, Spd is how fast a Pokemon is. This decides turn order. We also have generation - which set of games the Pokemon came out in. Not all 898 Pokemon came out at once.

disclaimer: I scraped this data myself, and it is subpar :(please excuse any funky stuff going on that is unaddressed, especially abilities! I had a hard time parsing them.

So, if we are looking only at stats of Pokemon, as we are, we would think that we would want the highest stats. If my Pokemon does more damage to yours, then I win. So, based off of that, we would think that the best players would use the strongest Pokemon, right? I used Smogon's Usage Statistics, which detail which Pokemon competitive Pokemon players use. Sadly, the highest strongest Pokemon stats wise was not the most used. So, here's our research questions.

1) What does our favorite Pokemon's usage look like?

2) We know our Pokemon's usage, can we find Pokemon with similar strength?

3) Are newer Pokemon better competitvely?

For our second question, we finally get to what the usage data is. The usage rates are for "OU" (overused) Pokemon, but that doesn't really explain what's going on. A Pokemon "starts" its journey in the OU tier, where then it's usage in that tier decides which tier it actually ends up in. There are a bunch of tiers, but this research basically boils it down to untiered and OU. The tiering system was created to allow for weak Pokemon to be used competitively without being an absolute liability in the match. As such, we have a lot of Pokemon with very low usage, which causes us to have to use a nifty trick on the data in order for us to get an answer for our second question.

Varaibles

We've already gone over most of the variables, so we'll just cover the "meta" stats here, which refer to usage and transformations on base stats.

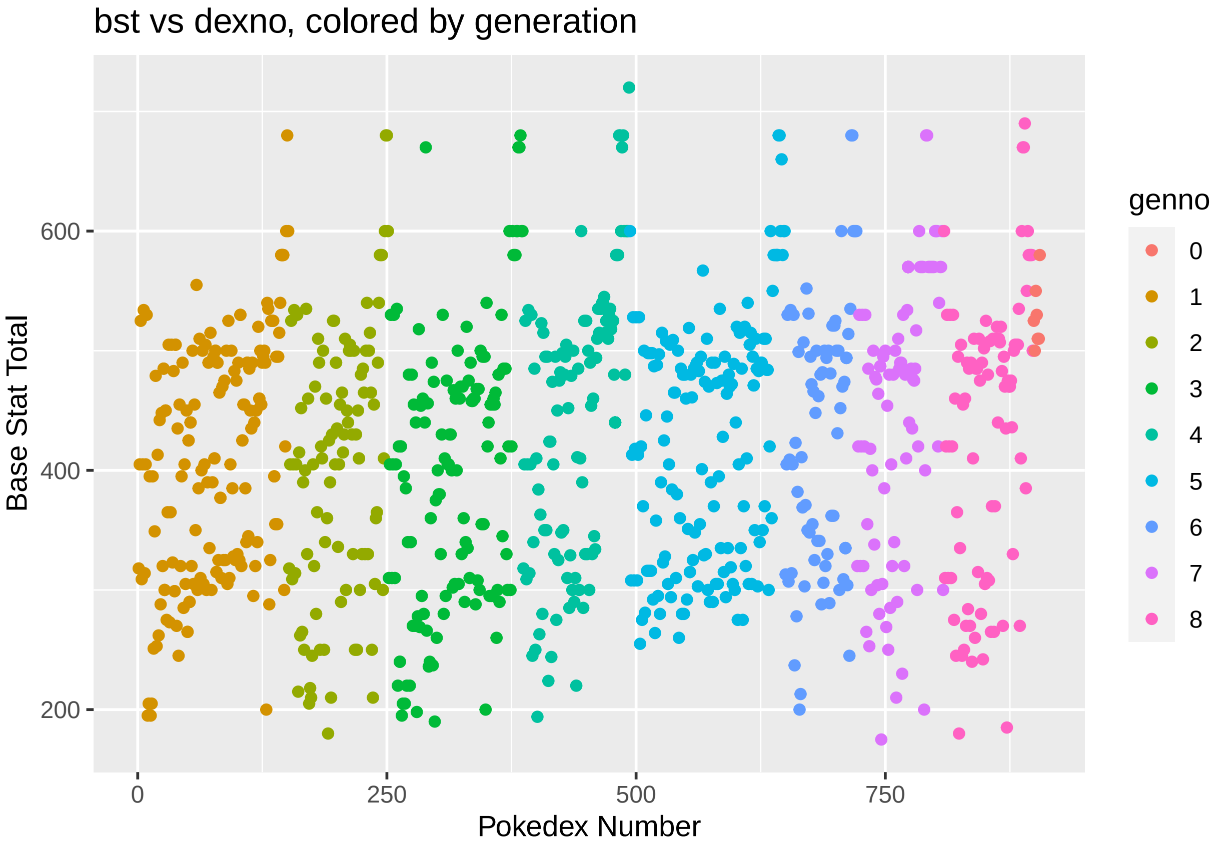
-bst: Base Stat Total. This is the additon of a Pokemon's stats. Lotad's bst, for instance, is 220, about half of the average bst. There's no units on this.

-Rate: this is the usage rate. This cannot be over 100 because it is a percentage. For ease of reading, .2 is 20 percent, but here we will call 20 percent 20.

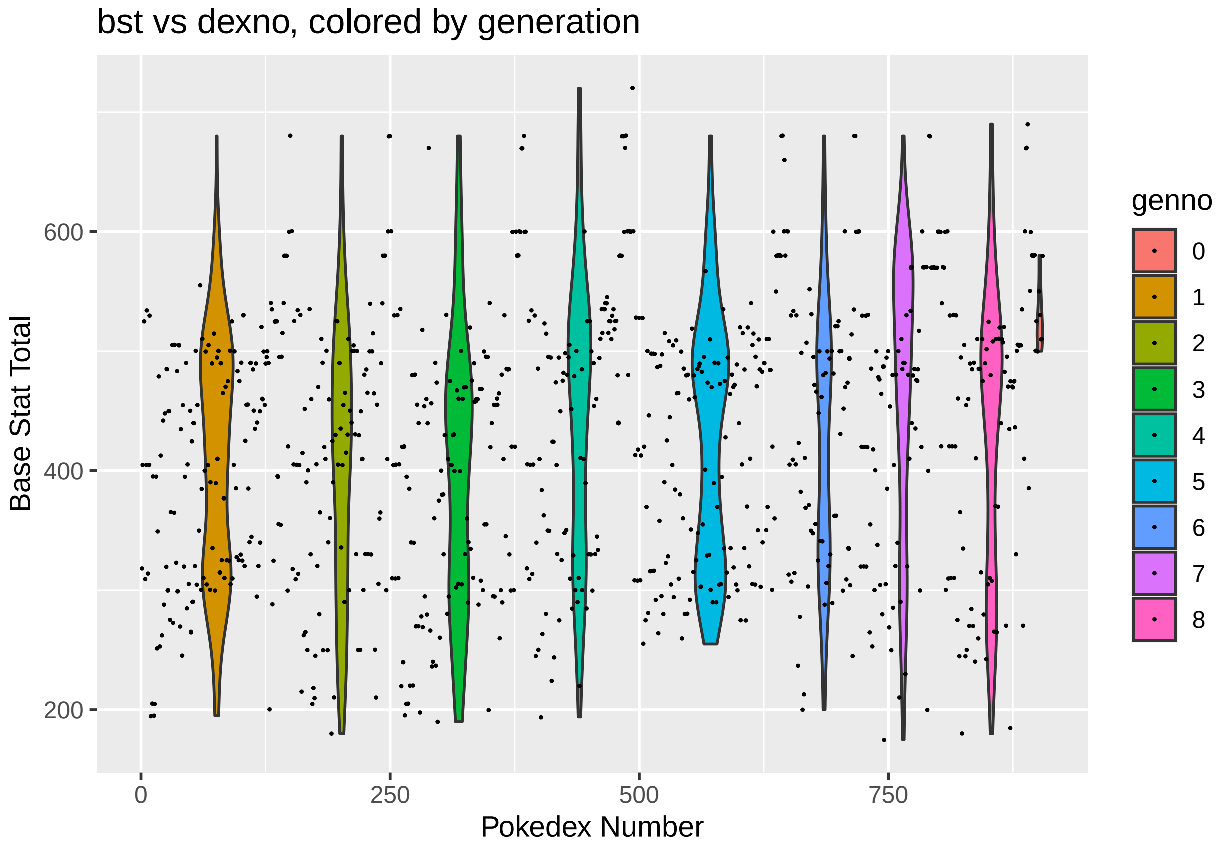
-Real: this is the number of battles a Pokemon has been brought to. It is a transformation of Rate.

-genno: this is the generation number, but as a factor. This was something I worked with and isn't too helpful, since it is just generation as a factor.

Each Generation's bst Distribution



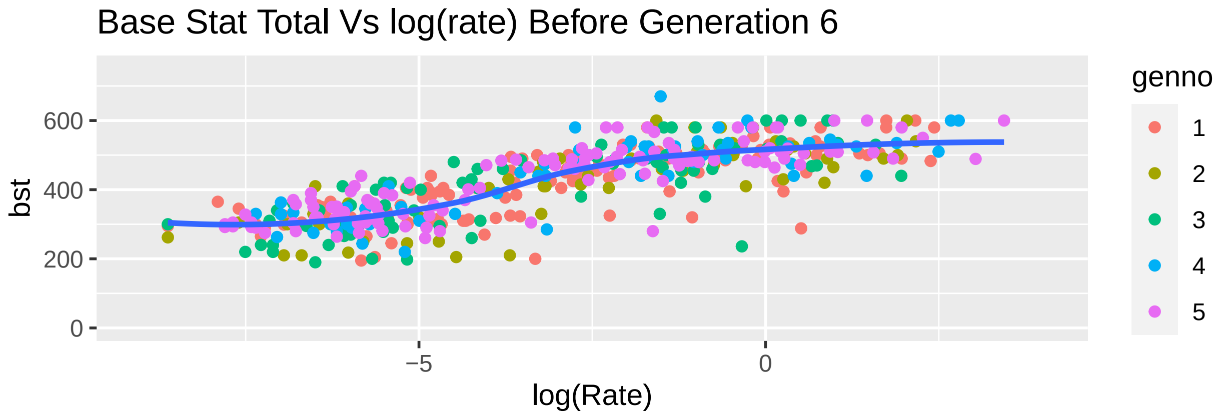
We can't glean a whole bunch of information in this plot, but we can tell a few things here that we can't in the next plot. First, we can see that some generations are "thin," while others are not. Our points are colored by which generation they belong to. So, lets look at generation 1. It is pretty wide, about twice as wide as generation 7. The range of the points on the x axis relate to the number of Pokemon introduced. We see that generations 1 and 5 have the most Pokemon added, so we'll remember to keep an eye on them for later. Next, we see that there are few Pokemon over 600 bst; those are legendary Pokemon, and often times they are banned from OU for being unfair, meaning they would have a very high usage rate. However, they aren't *always* banned, so later on when we are talking conclusions, they'll come up again. There are also a few Pokemon at 600, who aren't legendary but are strong, and they'll be a big part of this project going forwards.

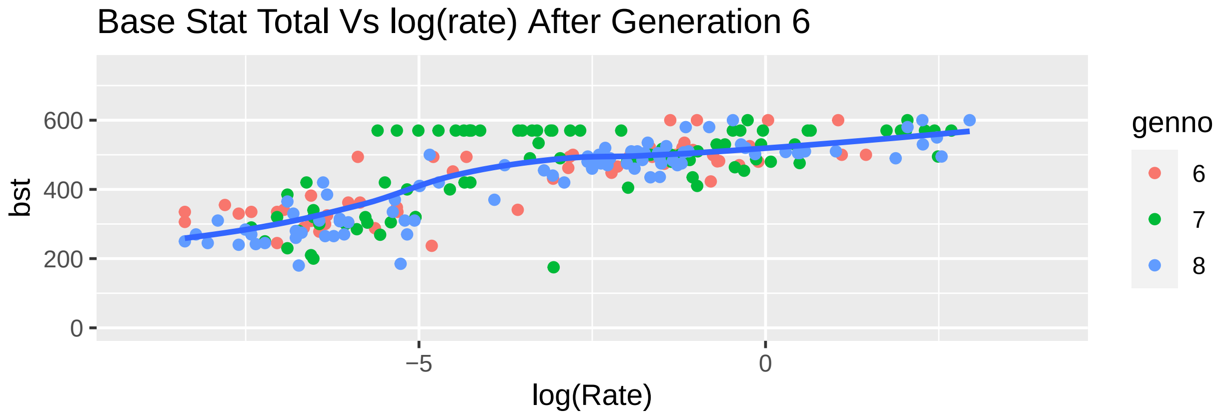


Here, we see a similar idea as the last, but in a violin plot (even the colors are the same!). This is a nice complementary plot because we get to see the distribution of base stat total in each generation and put that into conversation with the other generations. We can see a general trend, where the top is super skinny, the ~75th quartile being the widest part of the distribution, the ~50th quartile and the ~25th quartile being a little less wider, and then the bottom being skinny. These plots represent total number through, not percentage! Because of this we cannot take the plot completely at face value. We see than gens 1 and 5 have some pretty similar distributions, except that gen 5 doesn't go anywhere near as low as the rest of them! Interestingly enough, gen 5 is known as when excessive "power creep" began, which is where there are lots of strong Pokemon, so the weak ones get phased out of competitive play. We also see that generation 2 has a disproportionate amount of good Pokemon, and generation 7 takes that feeling to the next level. In Generation 6, the game started to reuse the bad Pokemon, opting to forgo creating more bad ones. Look in the appendix for the actual base stat total distributions.

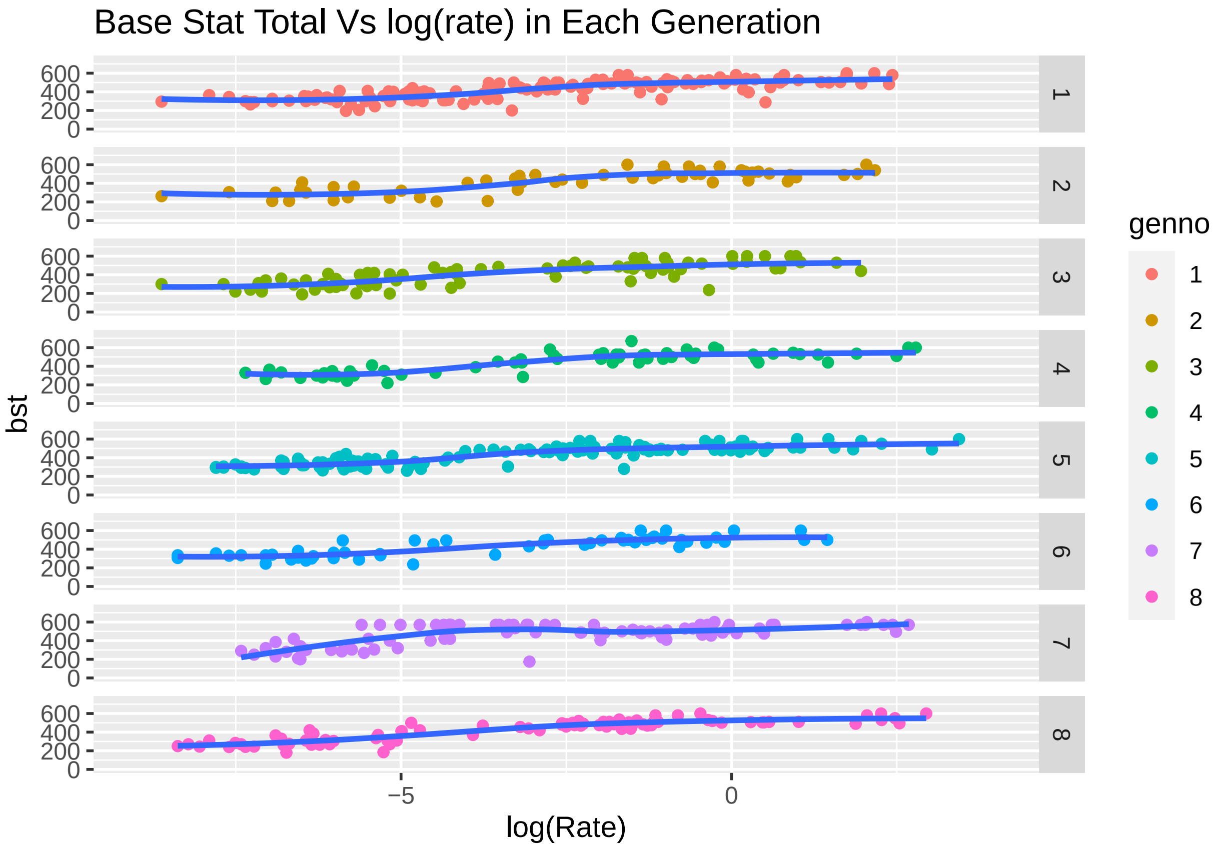
Before Generation 6 and After Generation 6

For these next two plots, we logged the Rate variable, allowing us to make a linear regression. Because of this transformation, we had to exclude Pokemon with 0 usage, which happens to be the abysmally weak and the exceedingly strong. The overly strong ones are banned due to being unfair, as previously mentioned, and they cause a little curving in our LOB. If these strong pokemon were not banned, we'd probably see a straighter, positive line, meaning that as bst increases, log(rate) does as well. Or, for easier reading, as as rate increases, bst increases, but at a progressively slower rate.


Base Stat Total Vs log(rate) Before Generation 6


Base Stat Total Vs log(rate) After Generation 6

Here, we see the two different groups (before and after gen 6) being compared, and we see pretty much the same thing. However, we see a slight difference at about log(rate) = -5: before generation 6, Pokemon with this base stat total (on average!) were used less than they were after generation 6. Does this mean that worse Pokemon were more competitively viable? No. Look around bst = 600, and we see a LOT of bst ~600 with a low usage rate, compared to where we would expect to see them (high usage). Our conclusion from these plots is actually that good Pokemon were LESS viable than before, meaning that the competitive scene was far more difficult, and that you'd need to use certain Pokemon in order to be competitive.



This is a plot with each generation separated so that we can look how each one preforms competitively. The right the line goes, the better the generation (or just the top end) is.

Modeling

I made a couple models, but just tested generation and bst as variables, to predict a Pokemons usage if we know its bst.

```
## 
## Call:
## lm(formula = log(Rate) ~ bst, data = logdex)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.4716 -1.1289 -0.0938  1.0293  6.7192 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.196e+01  2.726e-01  -43.88   <2e-16 ***
## bst          2.077e-02  6.185e-04   33.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.737 on 692 degrees of freedom
## Multiple R-squared:  0.6196, Adjusted R-squared:  0.6191
## F-statistic:  1127 on 1 and 692 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = log(Rate) ~ bst + generation, data = logdex)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2157 -1.1474 -0.0783  0.9906  6.6141
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.158e+01  2.875e-01  -40.29  < 2e-16 ***
## bst          2.095e-02  6.143e-04   34.10  < 2e-16 ***
## generation  -1.052e-01  2.746e-02   -3.83  0.00014 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.72 on 691 degrees of freedom
## Multiple R-squared:  0.6275, Adjusted R-squared:  0.6264
## F-statistic: 582.1 on 2 and 691 DF,  p-value: < 2.2e-16
```

```
## Analysis of Variance Table
##
## Model 1: log(Rate) ~ bst + generation
## Model 2: log(Rate) ~ bst
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    691 2045.4
## 2    692 2088.8 -1   -43.432 14.672 0.0001395 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

nested model

We'll take all of this output and talk about it one by one. So, our first output, for the nested model: we have coefficient for bst $\beta_1 = .0207$, which means that for each bst point we add, the log(rate) will go up .00207. We've got a really small p-value, meaning that this coefficient is statistically significant, and we should trust it. Next, we see the R-Squared being .6196, meaning that bst can only account for about 62% of our data, or that we have a decent predictor, but nothing to write home about. We are missing information to help us predict, most likely ability information.

full model

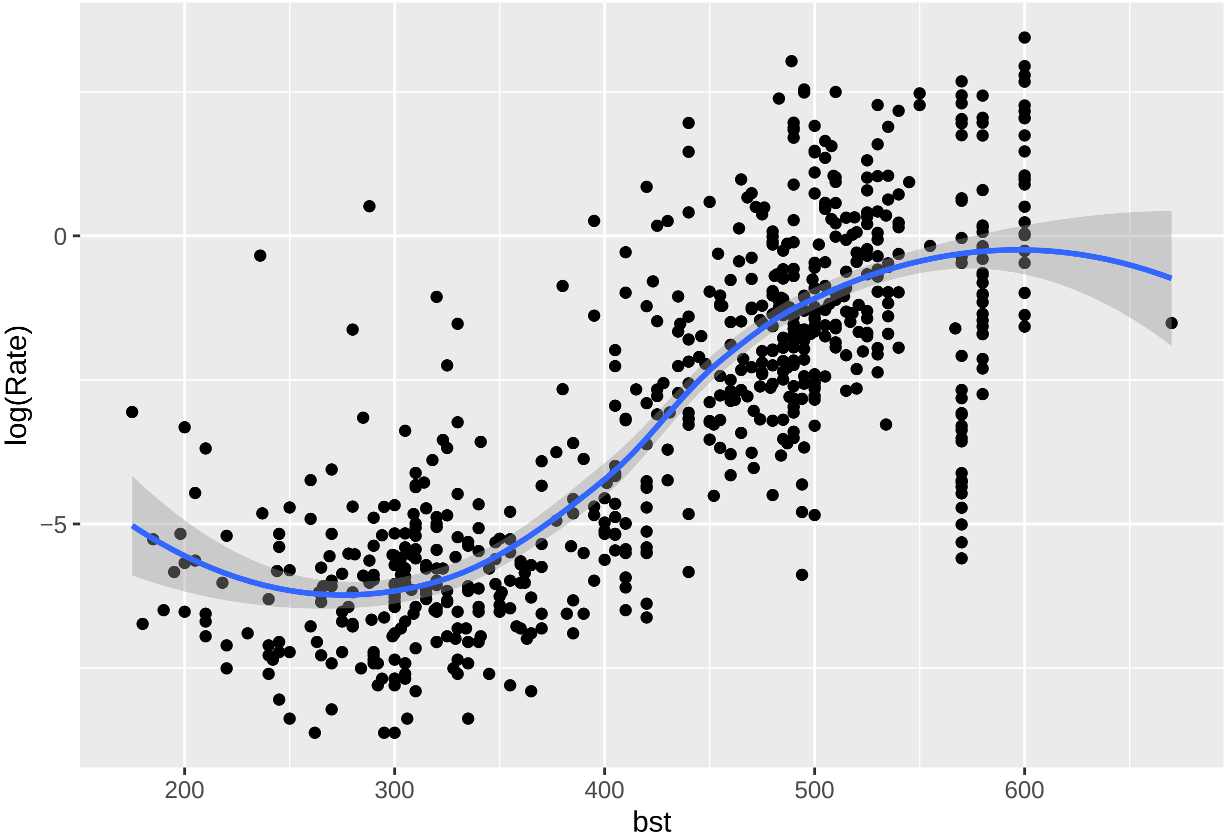
On our full model, we have that our coefficient for for bst is $\beta_2 = .02095$, meaning that for every point our bst increases, the log(rate) will go up by .02095. Again, we've got a very small p-value, so we can trust to keep this coefficient/variable in the model. Next, we have the coefficient for generation, $\beta_3 = -.1052$, meaning that for each newer generation, the log(rate) actually gets lower. Holding bst constant, it means that older Pokemon of the same strength are better. Lastly, our R-Squared is .6275, meaning that almost 63% of our data is accounted for, again making it a decent predictor. We now also have an Adjusted R-Squared, which tells us the quality of the variables we added; since the adjusted is lower than the regular r-squared, the variable was not great. However, the second model has a higher regualr/adjusted R-Squared, so we'll just use that one for predicting. Again, this would be better if we had more information, namely ability.

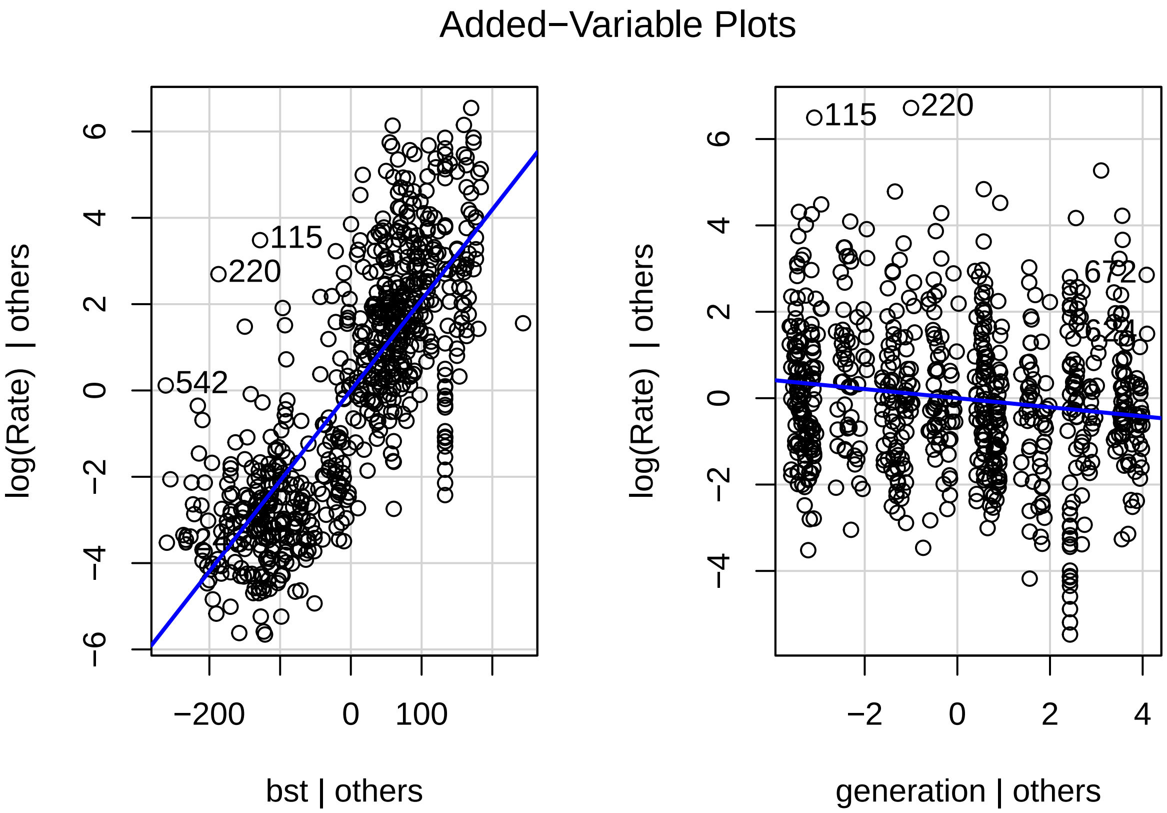
ANOVA test

Here, we tested the two models against each other, and assumed that they predict equally as well. We ran a hypothesis test using this, and found (with a low p-value) that we should definetly use the full model, as opposed to the nested version.

Visualization of Our Full Model

This has a big drop at the start because using bad Pokemon is a gimmick, but it's funny if you win with them, so people use them. It's not that these Pokemon are good, but they are popular. Further, we see that the strongest Pokemon is used as much as a bst 500 Pokemon, meaning there's something besides bst that matters.



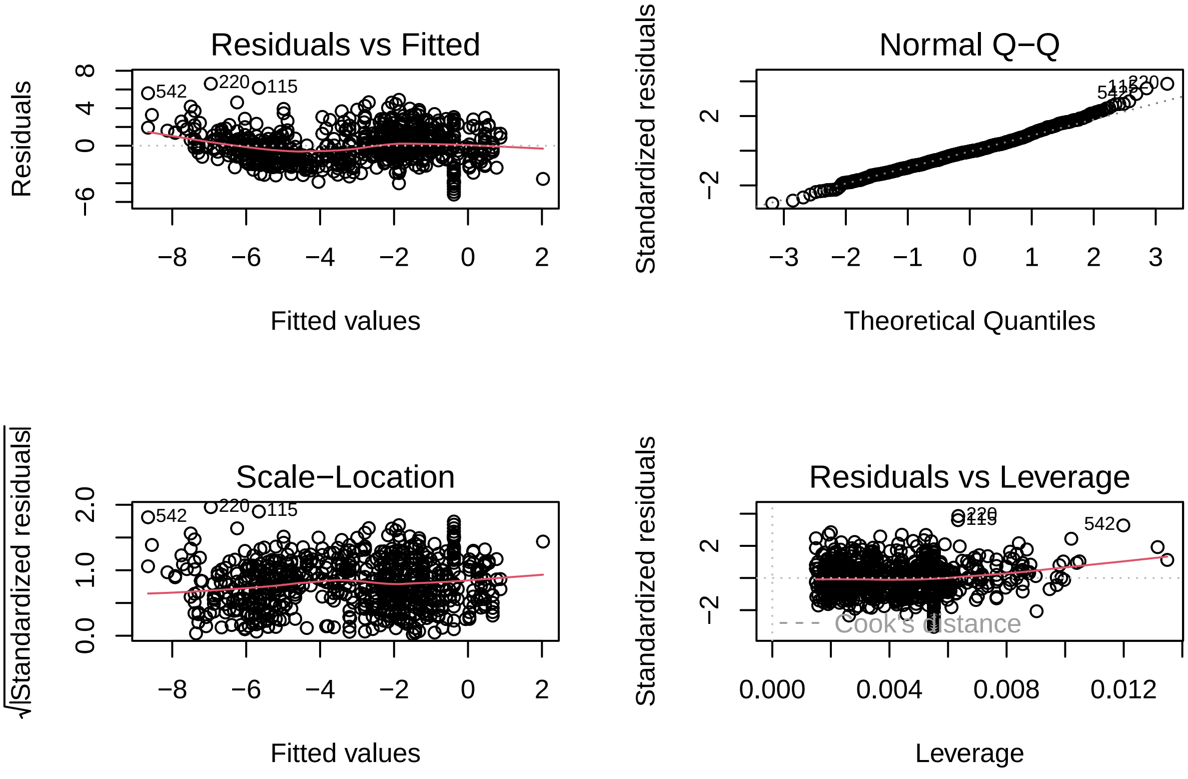


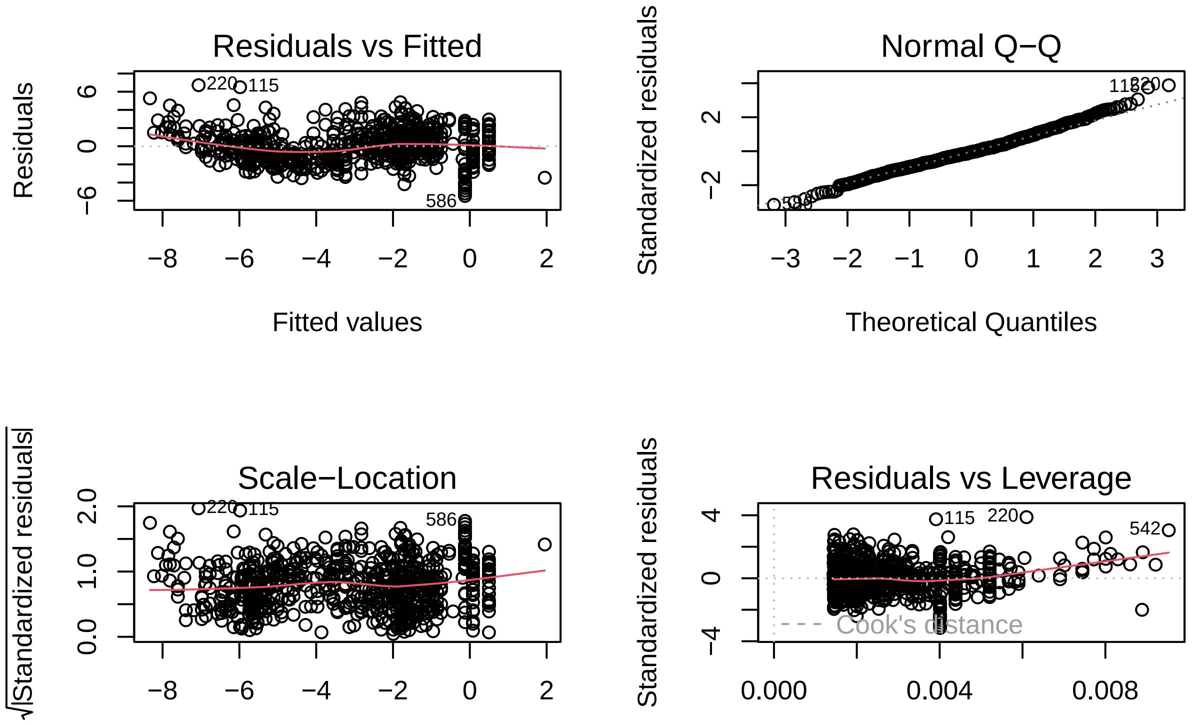
Another Way to See the Effect of Variables

In the plots above, we have a comparison that is pretty interesting. The plots show that if we hold all variables except for the one on the bottom constant, how the log(rate) will change. In a simpler sense, it allows us to compare Pokemon of either the same generation (left) or the same bst (right) to each other. For example, if the Pokemon are from generation 1, but have different bsts, we can see that the one with the higher bst will have a higher usage rate. In the other plot (right), if the Pokemon have the same bst but different generations, the one that was earlier will be better!

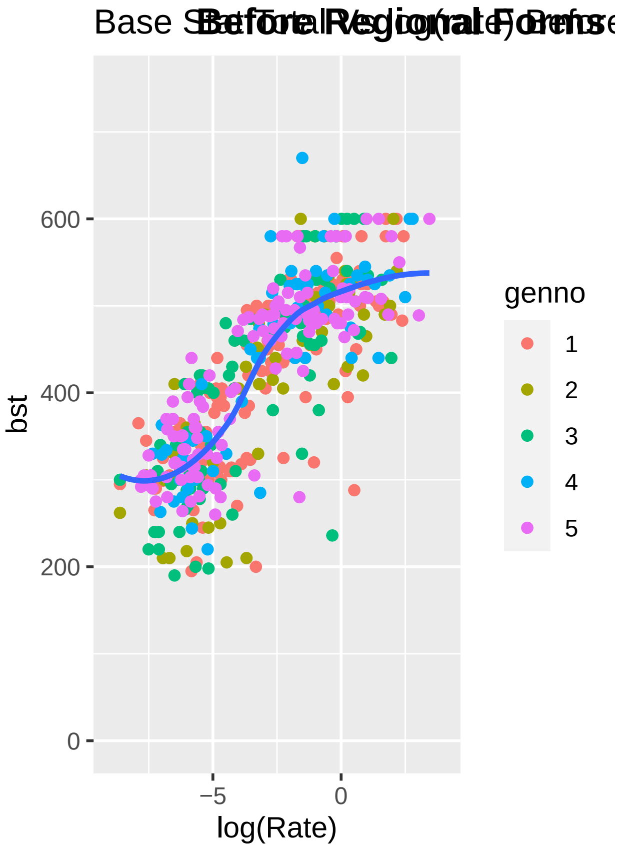
APPENDIX

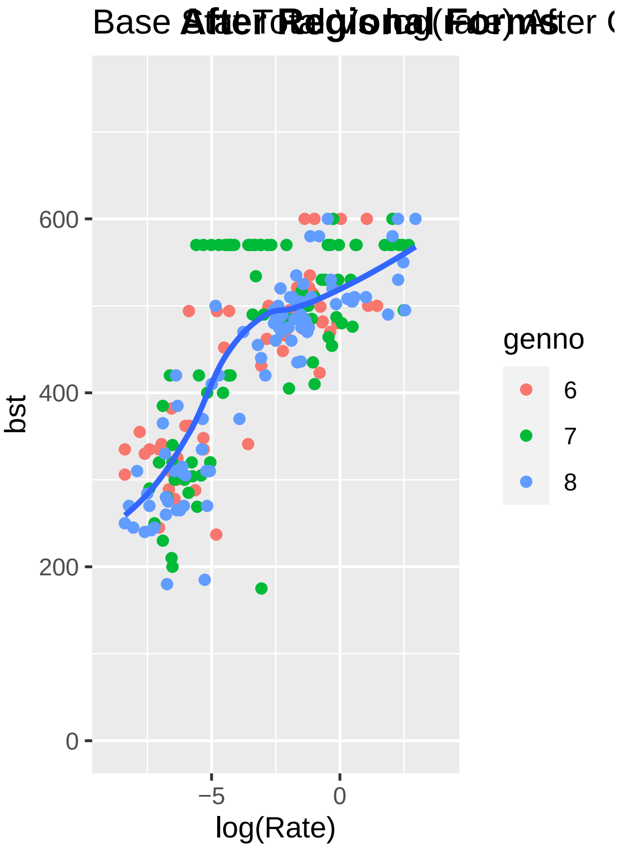
```
##
## Call:
## lm(formula = log(Rate) ~ bst + generation, data = logdex)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2157 -1.1474 -0.0783  0.9906  6.6141
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.158e+01  2.875e-01  -40.29   < 2e-16 ***
## bst          2.095e-02  6.143e-04   34.10   < 2e-16 ***
## generation  -1.052e-01  2.746e-02   -3.83  0.00014 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.72 on 691 degrees of freedom
## Multiple R-squared:  0.6275, Adjusted R-squared:  0.6264
## F-statistic: 582.1 on 2 and 691 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = log(Rate) ~ bst, data = logdex)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4716 -1.1289 -0.0938  1.0293  6.7192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.196e+01  2.726e-01  -43.88   <2e-16 ***
## bst          2.077e-02  6.185e-04   33.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.737 on 692 degrees of freedom
## Multiple R-squared:  0.6196, Adjusted R-squared:  0.6191
## F-statistic:  1127 on 1 and 692 DF,  p-value: < 2.2e-16
```

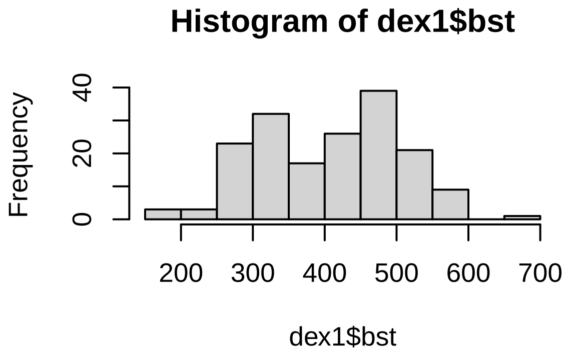


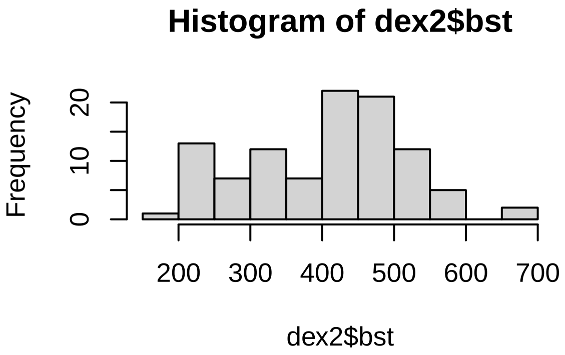


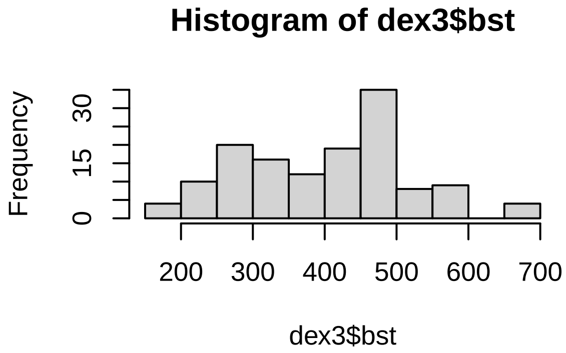
```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

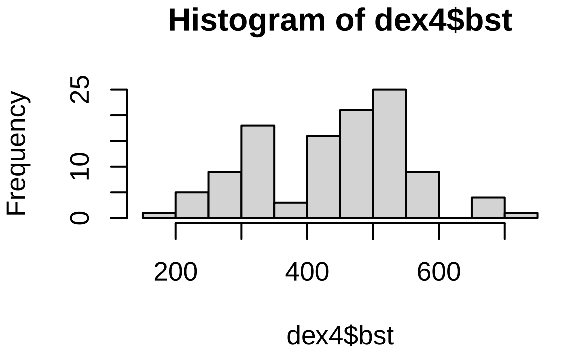

Before Regional Forms
bst
600
400
200
0
−5
0
log(Rate)
genno
1
2
3
4
5

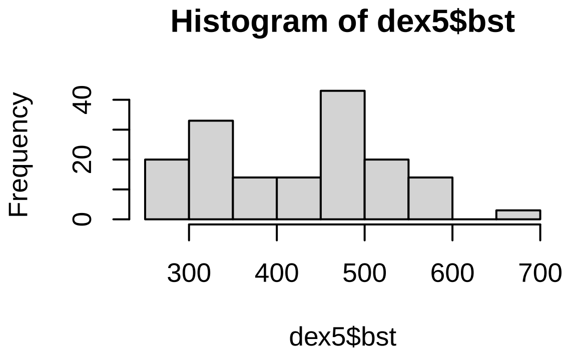

After Regional Forms
bst
600
400
200
0
−5
0
log(Rate)
genno
6
7
8

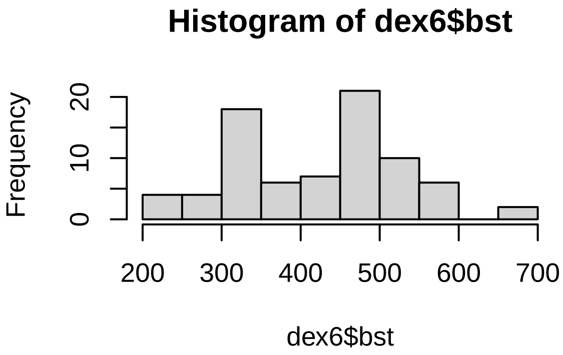

Histogram of dex1$bst
Frequency
0
20
40
200
300
400
500
600
700
dex1$bst

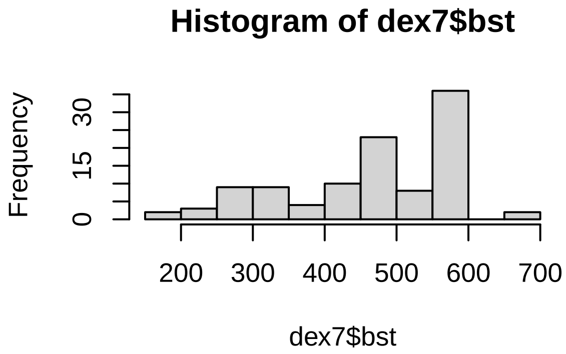

Histogram of dex2$bst
Frequency
0
10
20
200
300
400
500
600
700
dex2$bst

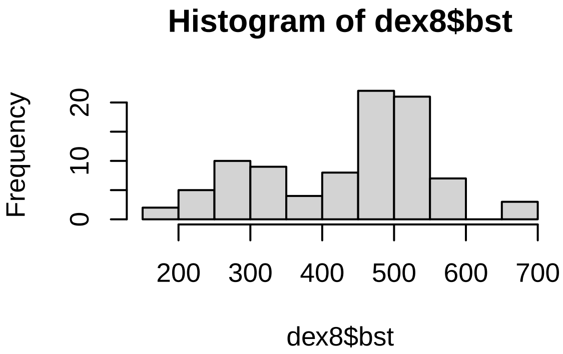

Histogram of dex3$bst
Frequency
0
15
30
200
300
400
500
600
700
dex3$bst


Histogram of dex4$bst
Frequency
0
10
25
200
400
600
dex4$bst


Histogram of dex5$bst
Frequency
0
20
40
300
400
500
600
700
dex5$bst


Histogram of dex6$bst
Frequency
0
10
20
200
300
400
500
600
700
dex6$bst


Histogram of dex7$bst
Frequency
0
15
30
200
300
400
500
600
700
dex7$bst


Histogram of dex8$bst
Frequency
0
10
20
200
300
400
500
600
700
dex8$bst