marchmadness2017

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(stringr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(PlayerRatings)
```

```
## Warning: package 'PlayerRatings' was built under R version 3.3.2
```

```
inpath <- "C:/Users/jroberti/Git/mm2017/data/"
    #"C:/Users/Amy/Documents/GitHub/mm2017/data/"

reg <- read.csv(paste0(inpath, "RegularSeasonCompactResults.csv"), stringsAsFactors = FALSE)

team <- read.csv(paste0(inpath, "Teams.csv"), stringsAsFactors = FALSE)
seasons <- read.csv(paste0(inpath, "Seasons.csv"), stringsAsFactors = FALSE)

tourney <- read.csv(paste0(inpath, "TourneyCompactResults.csv"), stringsAsFactors = FALSE)

head(reg)
```

```
##   Season Daynum Wteam Wscore Lteam Lscore Wloc Numot
## 1   1985     20  1228     81  1328     64    N     0
## 2   1985     25  1106     77  1354     70    H     0
## 3   1985     25  1112     63  1223     56    H     0
## 4   1985     25  1165     70  1432     54    H     0
## 5   1985     25  1192     86  1447     74    H     0
## 6   1985     25  1218     79  1337     78    H     0
```

estimate ELO ratings

- Pilot 2016 season to see how to format and evaluate whether it makes sense...

```
reg2016 <- filter(reg, Season==2016)
# assign 1 for win, 0 for loss, 0.5 for draw to team in left-most column
```

```
reg2016$Outcome <- ifelse(reg2016$Wscore - reg2016$Lscore>0, 1, 0)

ranks2016 <- steph(select(reg2016, Daynum, Wteam, Lteam, Outcome), history=TRUE)

regSeason2016 <- merge(ranks2016$ratings, team, by.x="Player", by.y="Team_Id")
arrange(regSeason2016, desc(Rating)) %>% head(30)
```

```
##    Player   Rating Deviation Games Win Draw Loss Lag      Team_Name
## 1    1242 2754.596  95.56238    33  29    0    4   1         Kansas
## 2    1332 2673.208  93.10324    33  27    0    6   1         Oregon
## 3    1277 2667.766  95.93249    34  29    0    5   0    Michigan St
## 4    1314 2651.349  92.70579    34  28    0    6   1 North Carolina
## 5    1437 2647.263  99.05290    34  29    0    5   1      Villanova
## 6    1438 2641.196  93.15092    33  26    0    7   1       Virginia
## 7    1371 2627.514  93.21456    33  25    0    8   1     Seton Hall
## 8    1462 2623.723 100.60215    32  27    0    5   2         Xavier
## 9    1428 2623.709  91.84258    33  25    0    8   1           Utah
## 10   1452 2614.583  92.02405    34  26    0    8   1  West Virginia
## 11   1328 2594.269  95.97940    32  25    0    7   2       Oklahoma
## 12   1274 2594.019  93.54436    32  25    0    7   2       Miami FL
## 13   1246 2572.044  90.92303    34  26    0    8   0       Kentucky
## 14   1345 2570.630  94.41302    34  26    0    8   0         Purdue
## 15   1257 2568.278  94.67939    31  23    0    8   8     Louisville
## 16   1112 2567.659  93.14874    33  25    0    8   2        Arizona
## 17   1143 2562.720  92.48756    33  23    0   10   2     California
## 18   1231 2548.563  98.12021    32  25    0    7   2        Indiana
## 19   1401 2536.878  92.87844    34  26    0    8   0      Texas A&M
## 20   1400 2528.090  94.18473    32  20    0   12   3          Texas
## 21   1124 2526.720  92.67573    32  21    0   11   2         Baylor
## 22   1139 2522.096  96.73781    31  21    0   10   3         Butler
## 23   1344 2507.500  95.95060    33  23    0   10   2     Providence
## 24   1439 2503.714  93.31535    33  19    0   14   3  Virginia Tech
## 25   1386 2502.055  94.89349    34  27    0    7   0 St Joseph's PA
## 26   1181 2496.467  92.76778    33  23    0   10   3           Duke
## 27   1268 2493.114  97.21830    32  24    0    8   1       Maryland
## 28   1163 2492.299  90.62244    34  24    0   10   0    Connecticut
## 29   1323 2490.544  93.99831    32  21    0   11   2     Notre Dame
## 30   1235 2489.866  95.60378    32  21    0   11   3        Iowa St
```

- Villanova is in top 5 for entire country (they won 2016 tourney)
-

1 ranked Kansas made it to Final Four, lost to Villanova

- Miami FL is in top 15, and made it to Elite Eight
- Maryland is in top 30, and made it to Elite Eight
- Oregon is in top 3, and made it to Final Four
- Oklahoma is in top 15 and made it to Final Four
- Texas A&M made it to Elite Eight and is in top 20
- Duke is in top 30, made it to Elite Eight

```
reg$wdiff <- reg$Wscore - reg$Lscore
reg$ldiff <- reg$Lscore - reg$Wscore

wreg <- select(reg, Season, Daynum, Wteam, Wscore, Wloc, Numot, wdiff) %>% rename(team=Wteam,score=Wsco
lreg <- select(reg, Season, Daynum, Lteam, Lscore, Wloc, Numot, ldiff) %>% rename(team=Lteam,score=Lsco

outreg <- rbind(wreg,lreg)
outreg$outcome <- ifelse(outreg$diff > 0, "win", "loss")


### NEED TO TURN OFF PLYR if dplyr:: is not specified for summarise
#detach(package:plyr)
start <- Sys.time()
proc_reg <- group_by(outreg, Season, team) %>%
  ## need to make sure to use summarise from dplyr, not plyr
  dplyr::summarise(totwin=sum(str_count(outcome, "win")),  # count total wins for the season
                   totloss=sum(str_count(outcome, "loss")),
                   ## average win margin - filter out negatives (those are losses), can do stdev too wi
                   wdiff_avg=mean(ifelse(diff>0, as.numeric(diff), 0)),
                   ldiff_avg=mean(ifelse(diff<0, as.numeric(diff), 0)),## average loss margin
                   score_avg=mean(score),
                   score_sd=sd(score),
                   wdiff_sd=sd(ifelse(diff>0, as.numeric(diff),0)),
                   ldiff_sd=sd(ifelse(diff<0, as.numeric(diff),0))
                   )
end <- Sys.time()
end - start # takes about 2.5 seconds to run
```

```
## Time difference of 3.661007 secs
```

```
head(proc_reg)
```

```
## Source: local data frame [6 x 10]
## Groups: Season [1]
##
##   Season  team totwin totloss   wdiff_avg  ldiff_avg score_avg  score_sd
##    <int> <int>  <int>   <int>       <dbl>      <dbl>     <dbl>     <dbl>
## 1   1985  1102      5      19  2.08333333  -7.875000  63.08333  9.964793
## 2   1985  1103      9      14  2.95652174  -6.000000  61.04348 11.125230
## 3   1985  1104     21       9  9.23333333  -1.433333  68.50000 13.860761
## 4   1985  1106     10      14  3.95833333  -7.750000  71.62500 11.765138
## 5   1985  1108     19       6 10.52000000  -2.560000  83.00000 14.077168
## 6   1985  1109      1      23  0.04166667 -29.166667  53.83333 11.567070
## # ... with 2 more variables: wdiff_sd <dbl>, ldiff_sd <dbl>
```

Process Tournament Data

```
tourney$wdiff <- tourney$Wscore - tourney$Lscore
tourney$ldiff <- tourney$Lscore - tourney$Wscore

wtourney <- select(tourney, Season, Daynum, Wteam, Wscore, Wloc, Numot, wdiff) %>% rename(team=Wteam,sc
ltourney <- select(tourney, Season, Daynum, Lteam, Lscore, Wloc, Numot, ldiff) %>% rename(team=Lteam,sc
```

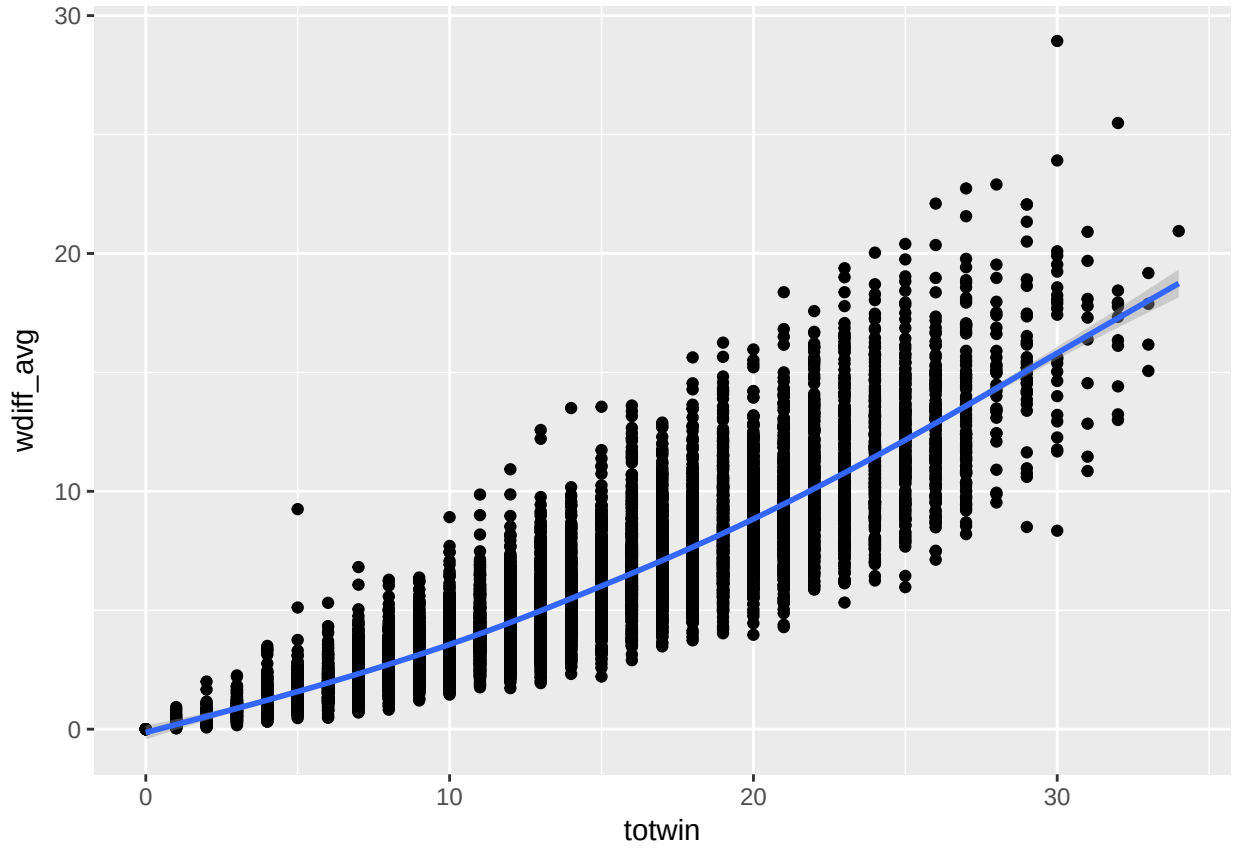
```
outtourney <- rbind(wtourney,ltourney)
outtourney$outcome <- ifelse(outtourney$diff > 0, "win", "loss")

proc_tourn <- group_by(outtourney, Season, team) %>%
  ## need to make sure to use summarise from dplyr, not plyr
  dplyr::summarise(totwin=sum(str_count(outcome, "win")),  # count total wins for the season
                   totloss=sum(str_count(outcome, "loss")),
                   ## average win margin - filter out negatives (those are losses), can do stdev too wi
                   wdiff_avg=mean(ifelse(diff>0, as.numeric(diff), 0)),
                   ldiff_avg=mean(ifelse(diff<0, as.numeric(diff), 0)),## average loss margin
                   score_avg=mean(score),
                   score_sd=sd(score),
                   wdiff_sd=sd(ifelse(diff>0, as.numeric(diff),0)),
                   ldiff_sd=sd(ifelse(diff<0, as.numeric(diff),0))
                   )

## rename "T_" == tournament data
names(proc_tourn) <- paste0("T_",names(proc_tourn))
```

```
ggplot(proc_reg, aes(totwin,wdiff_avg)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam'
```



Execute a merge

```
## make keys to match the data between the two tables
proc_reg$key <- paste0(proc_reg$Season,"_",proc_reg$team)
proc_tourn$key <- paste0(proc_tourn$T_Season,"_",proc_tourn$T_team)

## the tournament results should be the left table, because the proc_reg table
## has results of ALL teams that played (i.e. even teams that didn't make it to the tourney)
model_dat <- merge(proc_tourn, proc_reg, by.x="key", by.y="key")

model_dat$win_pct <- model_dat$totwin / (model_dat$totwin + model_dat$totloss)
```

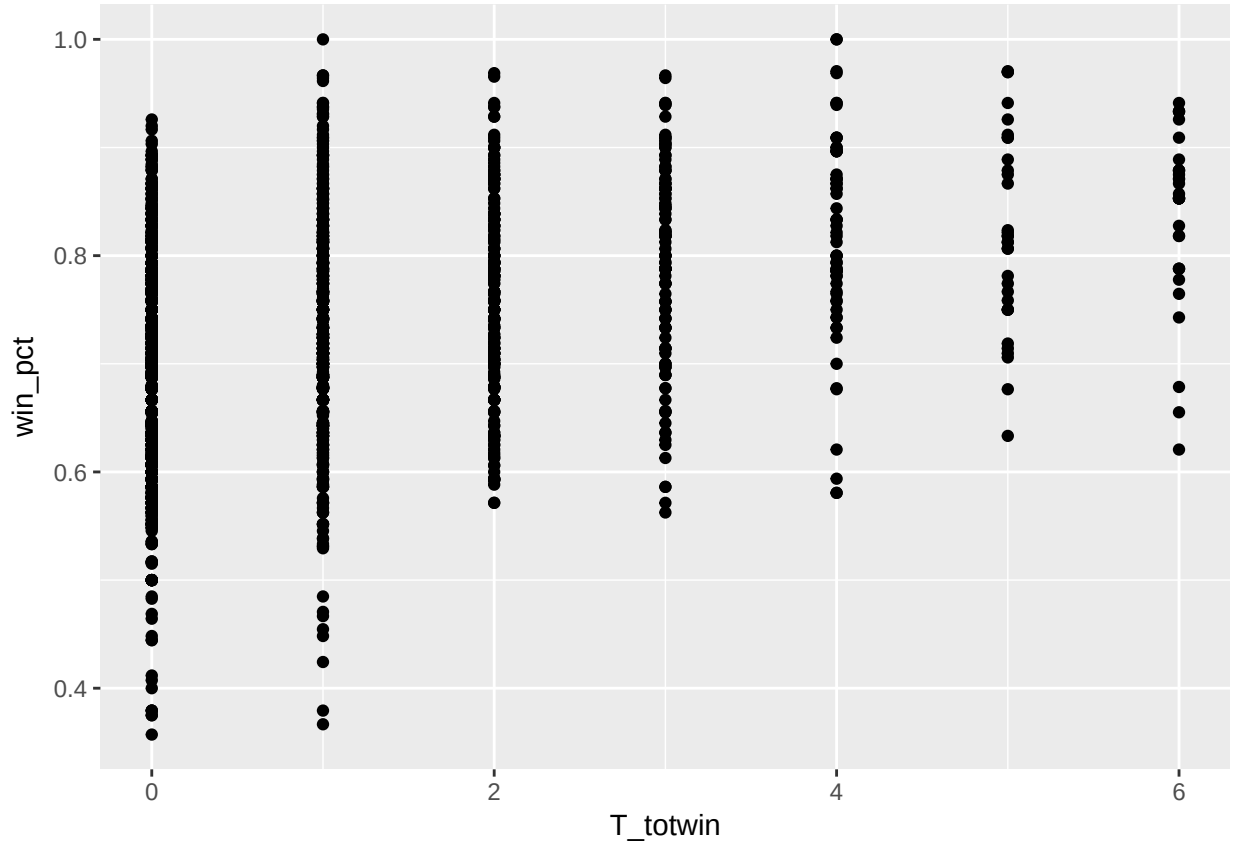
try a simple model

```
m1 <- lm(T_totwin ~ win_pct + wdiff_avg + ldiff_avg + wdiff_sd + ldiff_sd, data = model_dat)
summary(m1)
```

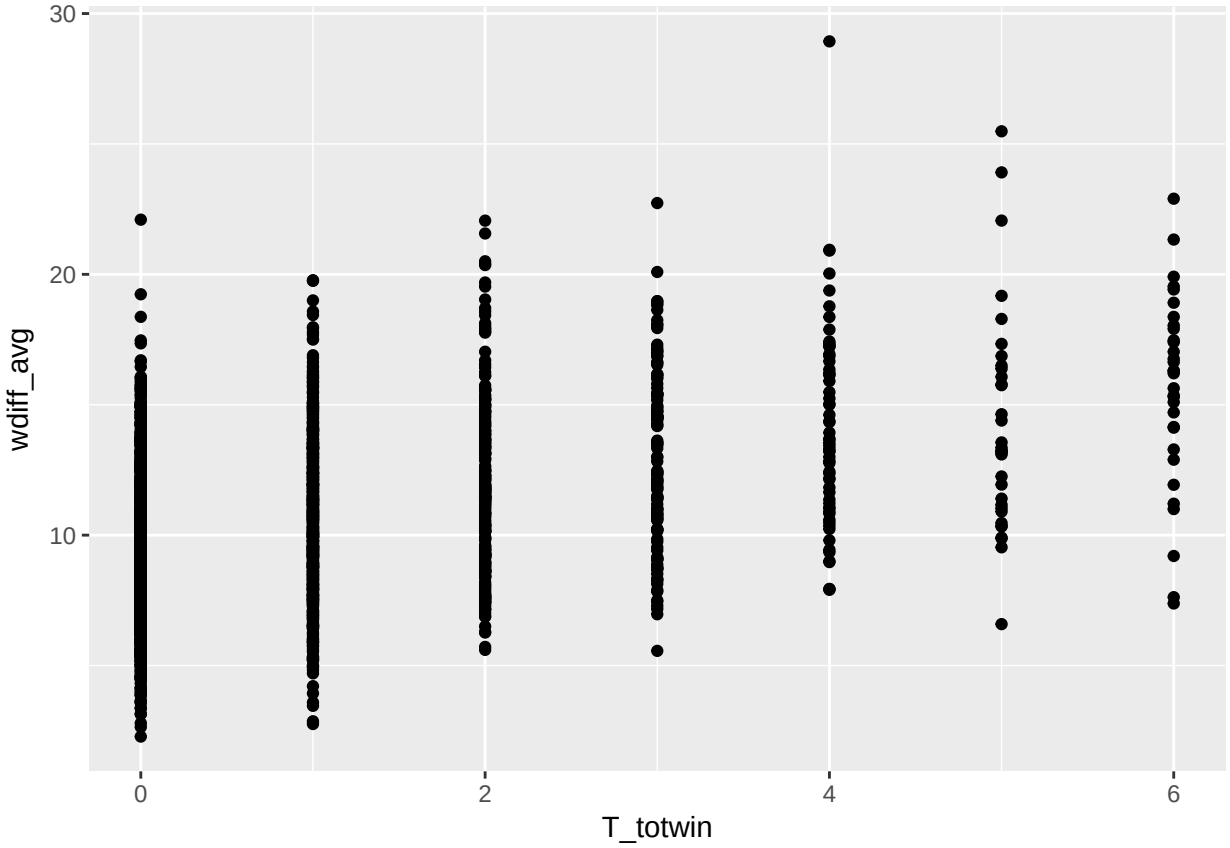
```
##
## Call:
## lm(formula = T_totwin ~ win_pct + wdiff_avg + ldiff_avg + wdiff_sd +
##     ldiff_sd, data = model_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7341 -0.7664 -0.2809  0.5174  5.6076
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.493944   0.523733  -2.852  0.00438 **
## win_pct      1.769504   0.684868   2.584  0.00984 **
## wdiff_avg    0.142103   0.020666   6.876  8.1e-12 ***
## ldiff_avg   -0.218296   0.073286  -2.979  0.00293 **
## wdiff_sd     0.003513   0.019525   0.180  0.85725
## ldiff_sd    -0.169568   0.035324  -4.800  1.7e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.179 on 2076 degrees of freedom
## Multiple R-squared:  0.2206, Adjusted R-squared:  0.2187
## F-statistic: 117.5 on 5 and 2076 DF,  p-value: < 2.2e-16
```

try some viz for tourney data

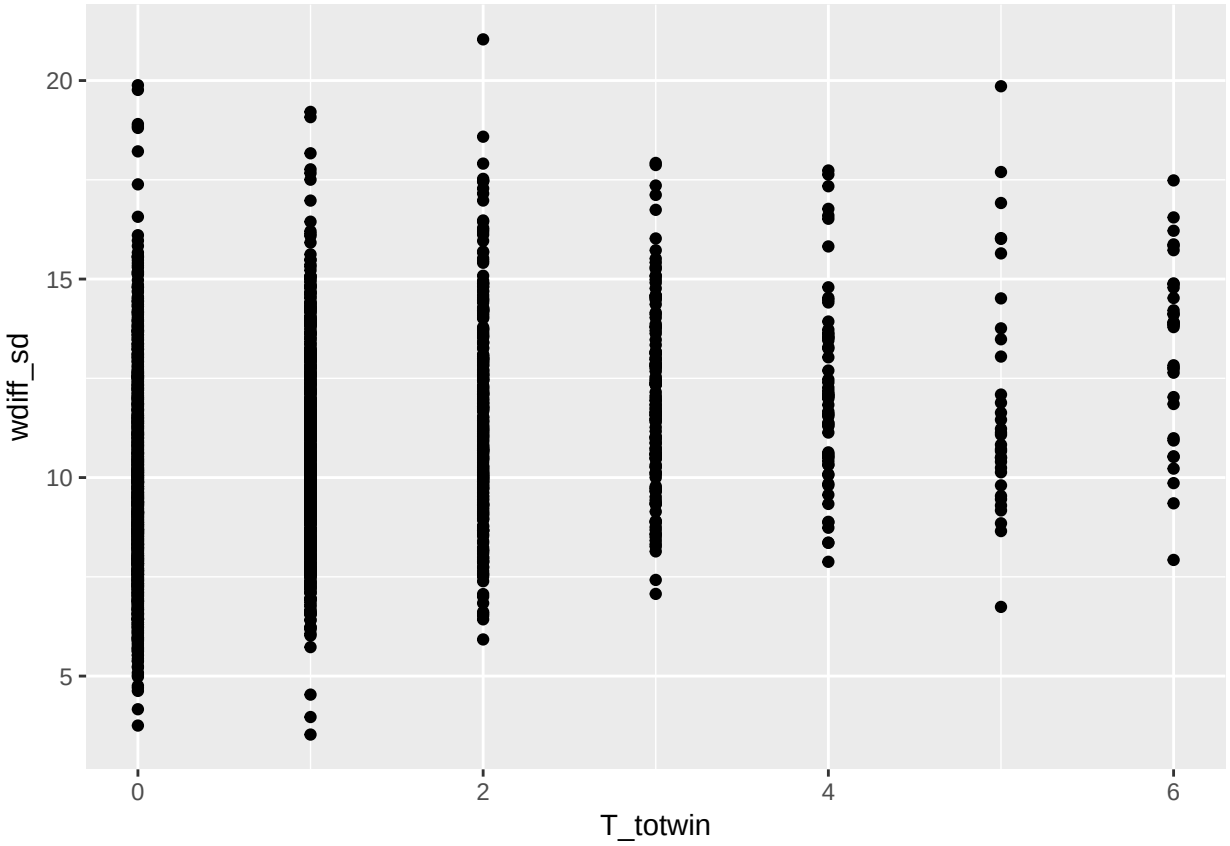
```
ggplot(model_dat, aes(T_totwin,win_pct)) + geom_point()
```




```
ggplot(model_dat, aes(T_totwin,wdiff_avg)) + geom_point()
```




```
ggplot(model_dat, aes(T_totwin,wdiff_sd)) + geom_point()
```


20
15
10
5
wdiff_sd
0
2
4
6
T_totwin