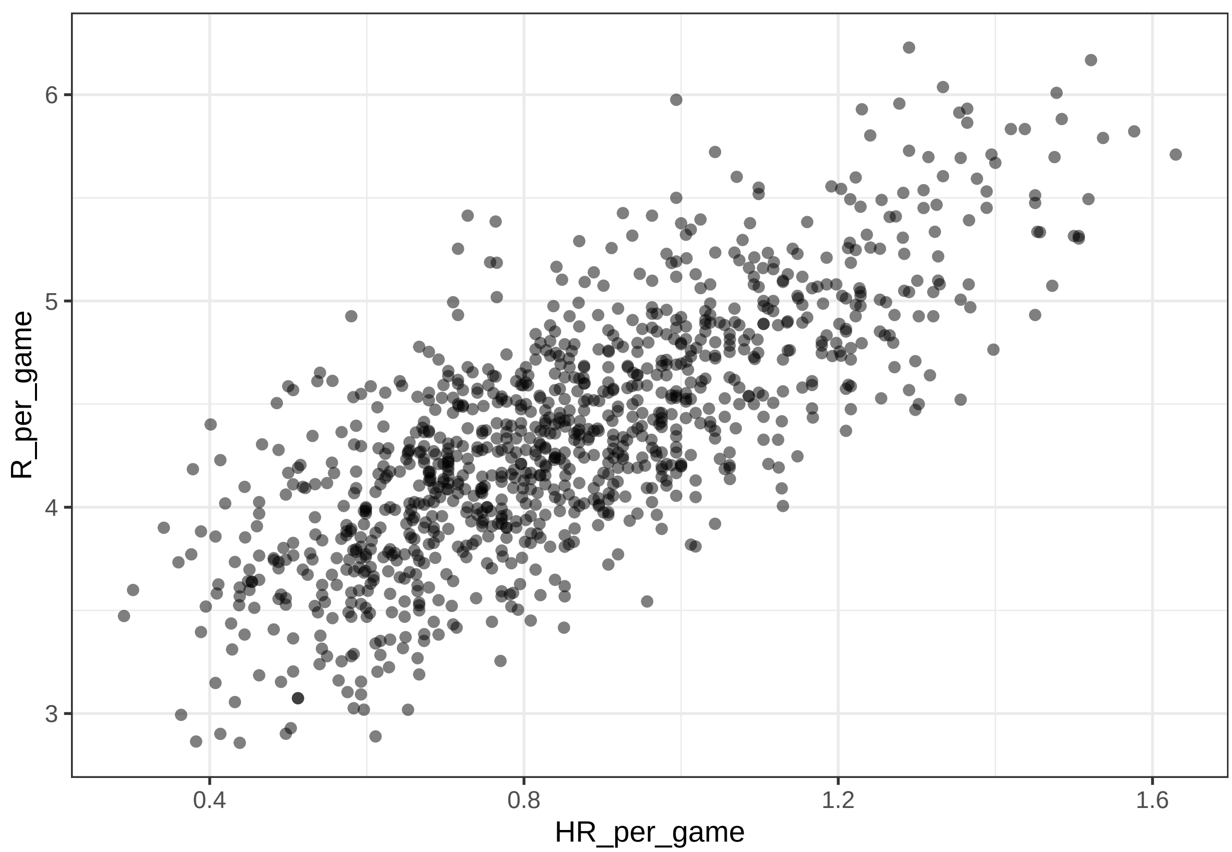
Linear Regression

Case study: Moneyball

```
library(tidyverse)
library(Lahman)
library(dslabs)
library(tinytex)
ds_theme_set()
```

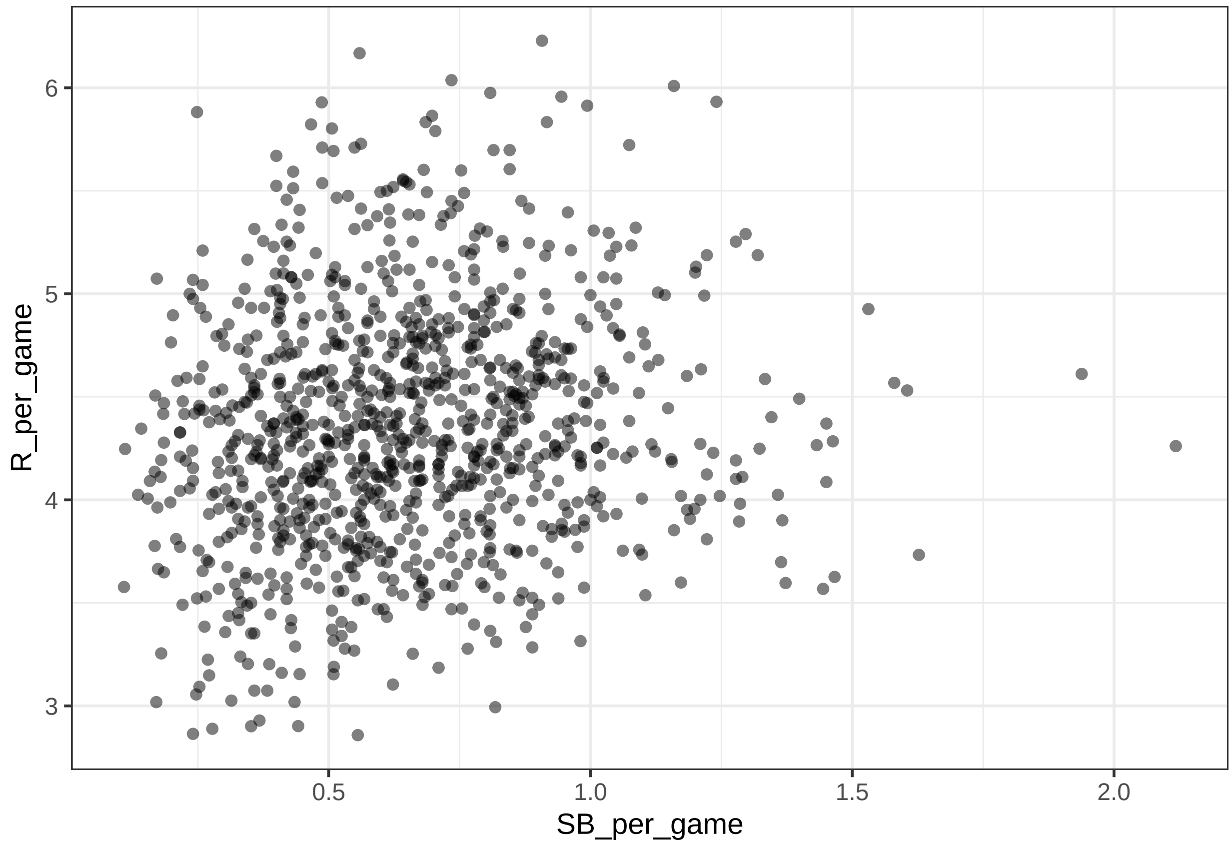
Plot relationship between home runs and runs per game (wins):

```
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(HR_per_game = HR / G,
         R_per_game = R / G) %>%
  ggplot(aes(HR_per_game, R_per_game)) +
  geom_point(alpha = 0.5)
```



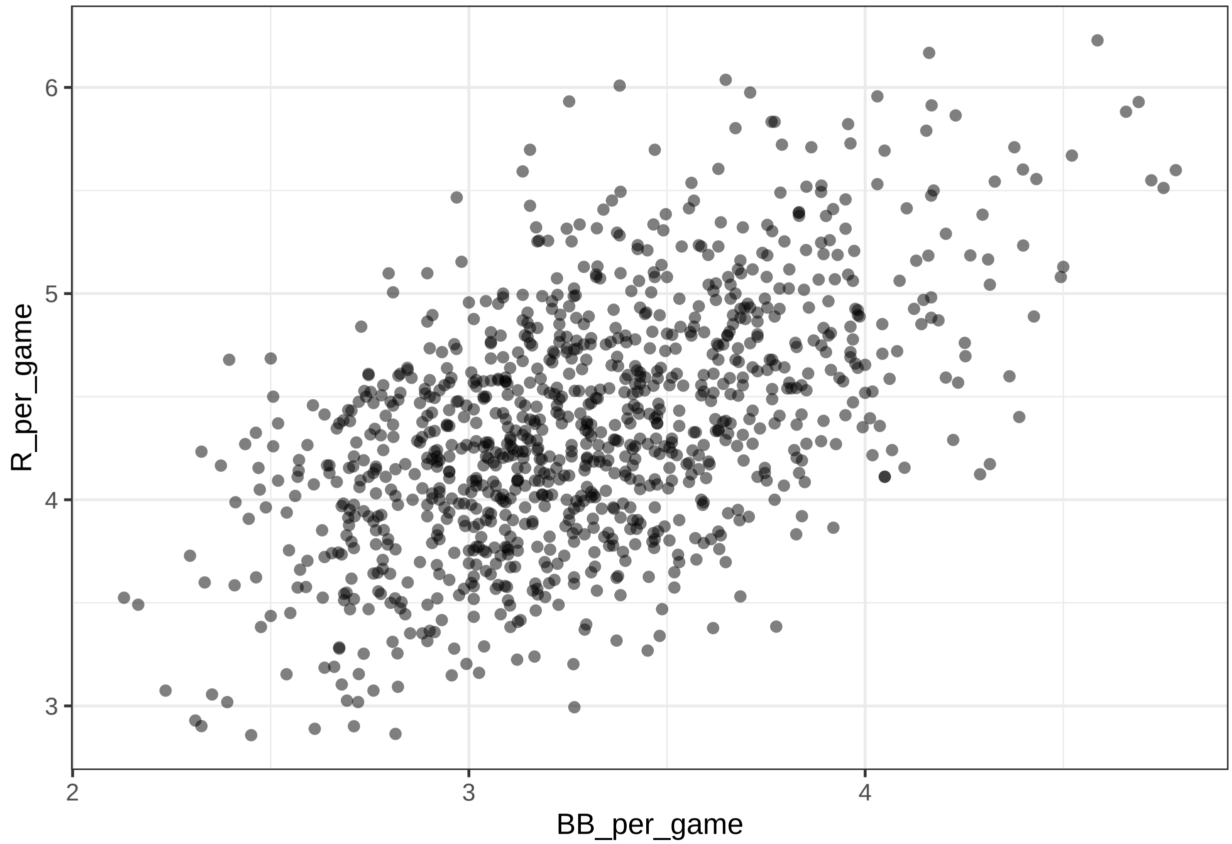
Plot relationship between stolen bases and runs per game (wins):

```r
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(SB_per_game = SB / G,
         R_per_game = R / G) %>%
  ggplot(aes(SB_per_game, R_per_game)) +
  geom_point(alpha = 0.5)
```



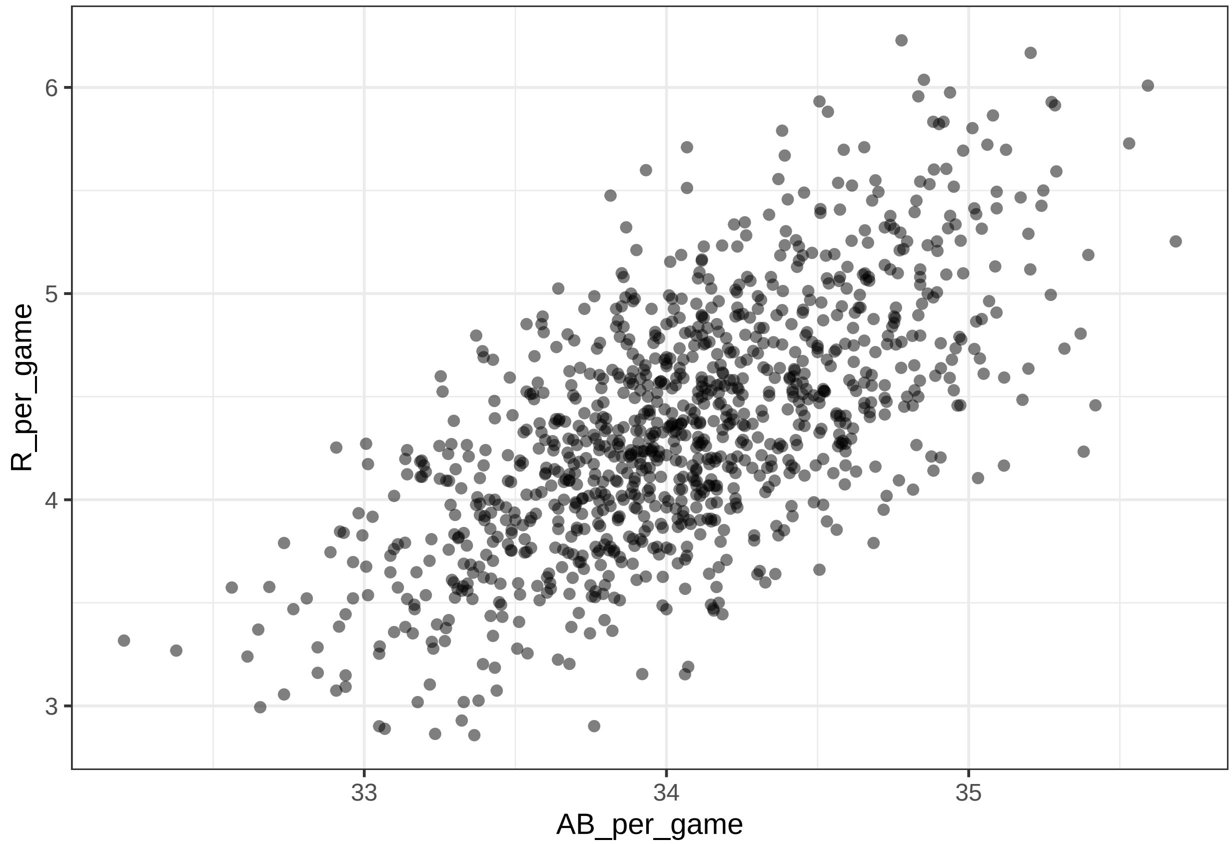
Plot relationship between base on balls and runs:

```r
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(BB_per_game = BB / G,
         R_per_game = R / G) %>%
  ggplot(aes(BB_per_game, R_per_game)) +
  geom_point(alpha = 0.5)
```



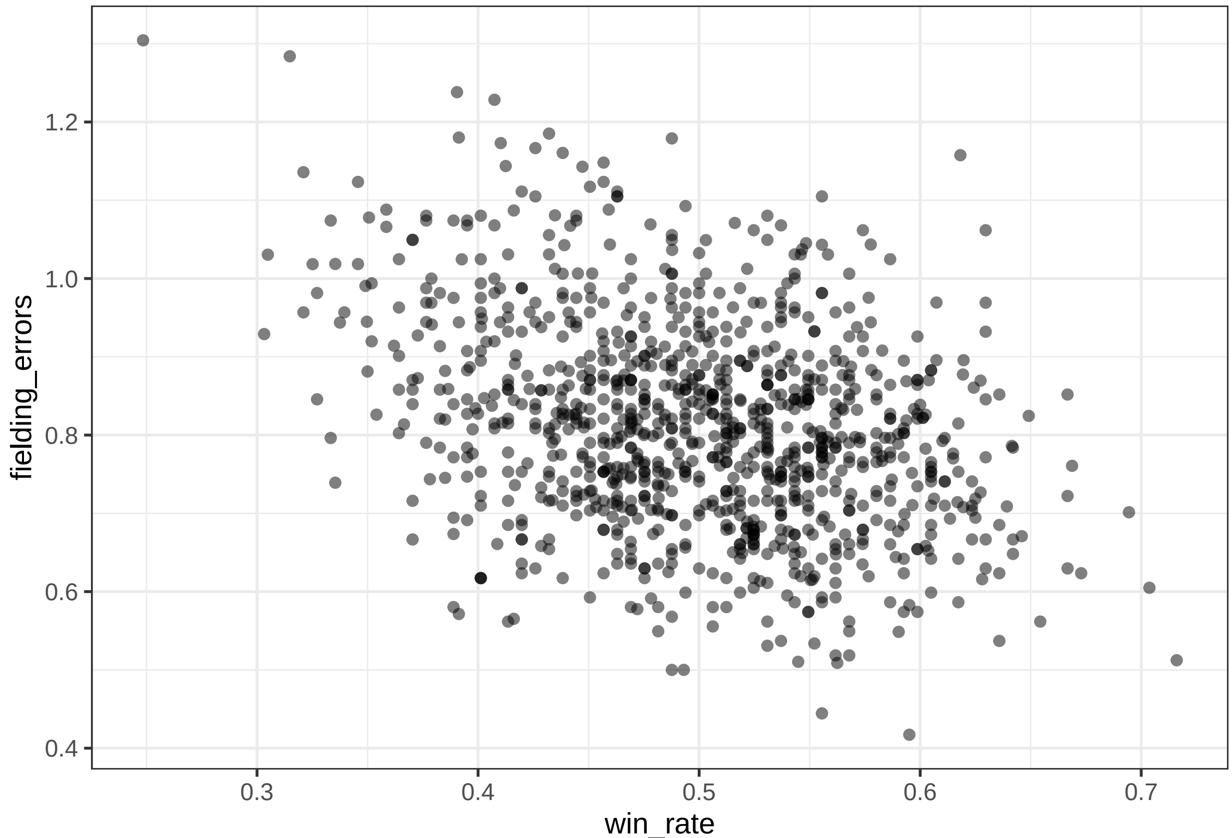
Plot relationship between at-bats per game and runs per game

```
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(AB_per_game = AB / G,
         R_per_game = R / G) %>%
  ggplot(aes(AB_per_game, R_per_game)) +
  geom_point(alpha = 0.5)
```



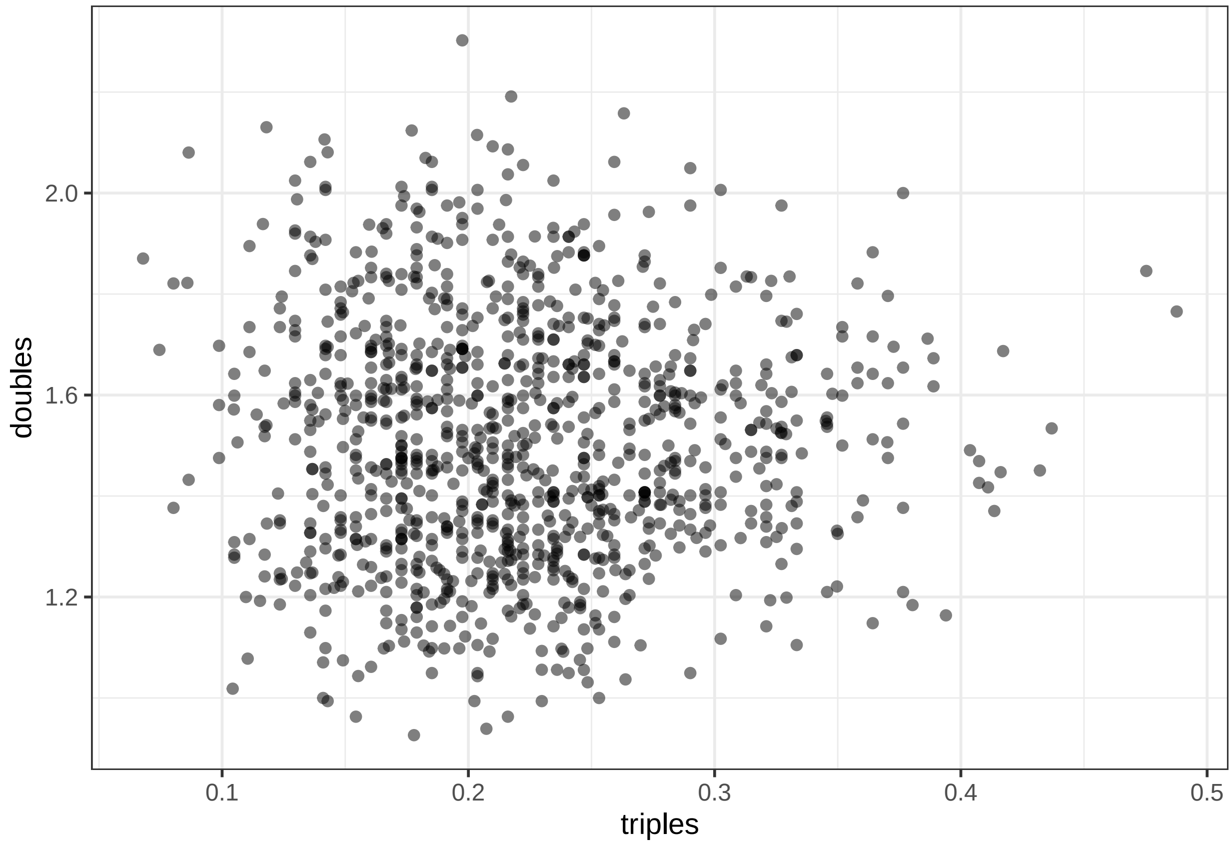
Plot relationship between win rate and fielding errors

```
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(win_rate = W / G,
         fielding_errors = E / G) %>%
  ggplot(aes(win_rate, fielding_errors)) +
  geom_point(alpha = 0.5)
```



Plot triples per game vs doubles per game

```
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(triples = X3B / G,
         doubles = X2B / G) %>%
  ggplot(aes(triples, doubles)) +
  geom_point(alpha = 0.5)
```




```r
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(runs_per_game = R/G,
         AB_per_game = AB/G) %>%
  summarize(r = cor(runs_per_game, AB_per_game)) %>%
  pull(r)
```

```
## [1] 0.6580976
```

```r
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(wins_per_game = W/G,
         errors_per_game = E/G) %>%
  summarize(r = cor(wins_per_game, errors_per_game)) %>%
  pull(r)
```

```
## [1] -0.3396947
```

```r
Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(doubles = X2B/G,
         triples = X3B/G) %>%
  summarize(r = cor(doubles, triples)) %>%
  pull(r)
```

```
## [1] -0.01157404
```

Galton Genetics Assessment

Analyze mother and daughter heights from GaltonFamilies

```
set.seed(1989, sample.kind = 'Rounding')
```

```
## Warning in set.seed(1989, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
library(HistData)
```

```
data("GaltonFamilies")
```

```
female_heights <- GaltonFamilies %>%
  filter(gender == 'female') %>%
  group_by(family) %>%
  sample_n(1) %>%
  ungroup() %>%
  select(mother, childHeight) %>%
  rename(daughter = childHeight)

head(female_heights)
```

```
## # A tibble: 6 x 2
##   mother daughter
##    <dbl>    <dbl>
## 1   67       69
## 2   66.5     65.5
## 3   64       68
## 4   64       64.5
## 5   58.5     66.5
## 6   68       69.5
```

Calculate the mean and standard deviation of mothers' and daughters' heights. Calculate the correlation coefficient between mother and daughter heights.

```
mom_m <- mean(female_heights$mother)
mom_sd <- sd(female_heights$mother)
dau_m <- mean(female_heights$daughter)
dau_sd <- sd(female_heights$daughter)
rho <- cor(female_heights$mother, female_heights$daughter)

mom_m
```

```
## [1] 64.125
```

```
mom_sd
```

```
## [1] 2.289292
```

```r
dau_m
```

```
## [1] 64.28011
```

```r
dau_sd
```

```
## [1] 2.39416
```

```r
rho
```

```
## [1] 0.3245199
```

```r
summary(female_heights)
```

```
##      mother         daughter    
##  Min.   :58.00   Min.   :57.00  
##  1st Qu.:63.00   1st Qu.:63.00  
##  Median :64.00   Median :64.50  
##  Mean   :64.12   Mean   :64.28  
##  3rd Qu.:66.00   3rd Qu.:66.00  
##  Max.   :70.50   Max.   :70.50
```

```r
slope <- rho * dau_sd / mom_sd
slope
```

```
## [1] 0.3393856
```

```r
intercept <- dau_m - slope * mom_m
intercept
```

```
## [1] 42.51701
```

```r
variance <- rho^2*100
variance
```

```
## [1] 10.53132
```

```r
intercept + slope * 60
```

```
## [1] 62.88015
```

Linear Models

```r
galton_heights <- GaltonFamilies %>%
  filter(gender == 'male') %>%
  group_by(family) %>%
  sample_n(1) %>%
  ungroup() %>%
  select(father, childHeight) %>%
  rename(son = childHeight)
```

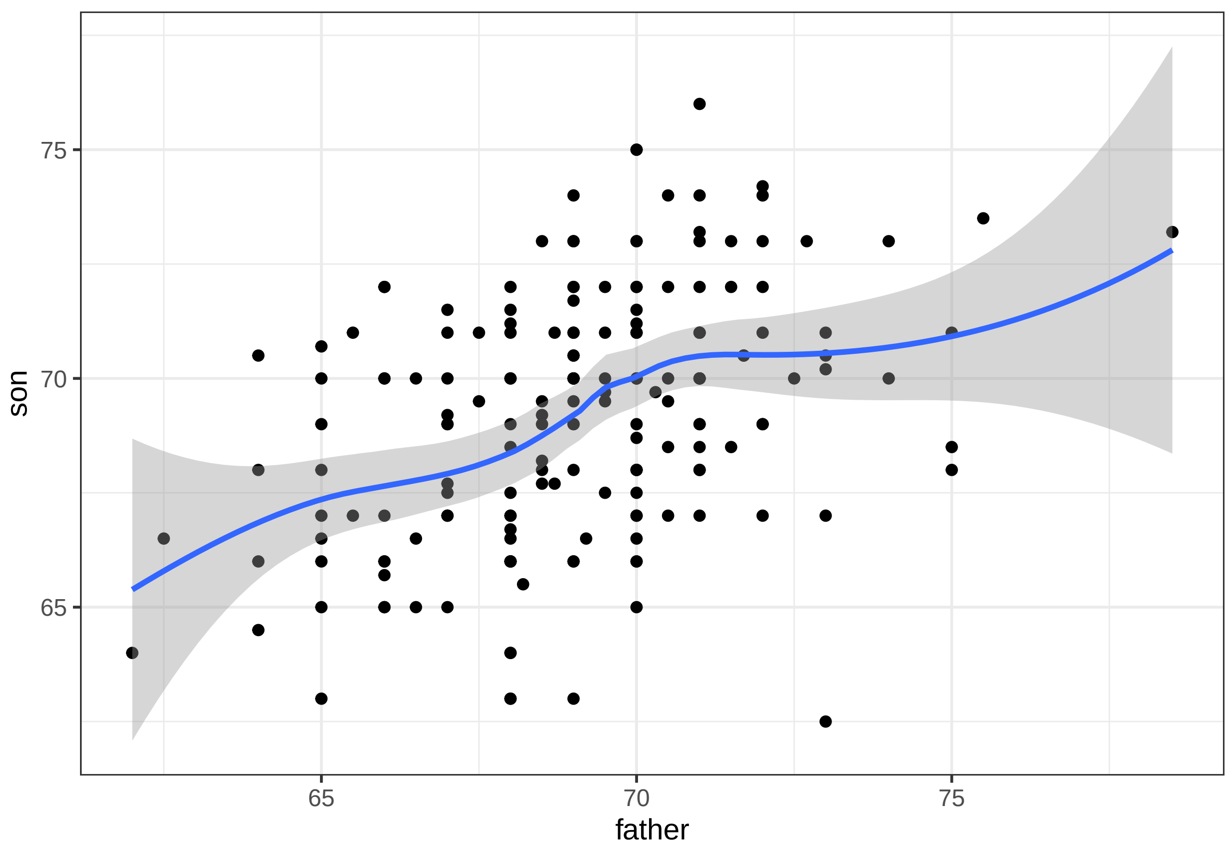
```r
rss <- function(beta0, beta1, data){
  resid <- galton_heights$son - (beta0 + beta1 * galton_heights$father)
  return(sum(resid^2))
}
```

```r
B <- 1000
N <- 50

lse <- replicate(B, {
  sample_n(galton_heights, N, replace = TRUE) %>%
    mutate(father = father - mean(father)) %>%
    lm(son ~ father, data = .)
})

# error with given function using %>% .$coef
```

```r
galton_heights %>%
  ggplot(aes(father, son)) +
  geom_point() +
  geom_smooth(mthod = 'lm')
```

```
## Warning: Ignoring unknown parameters: mthod
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



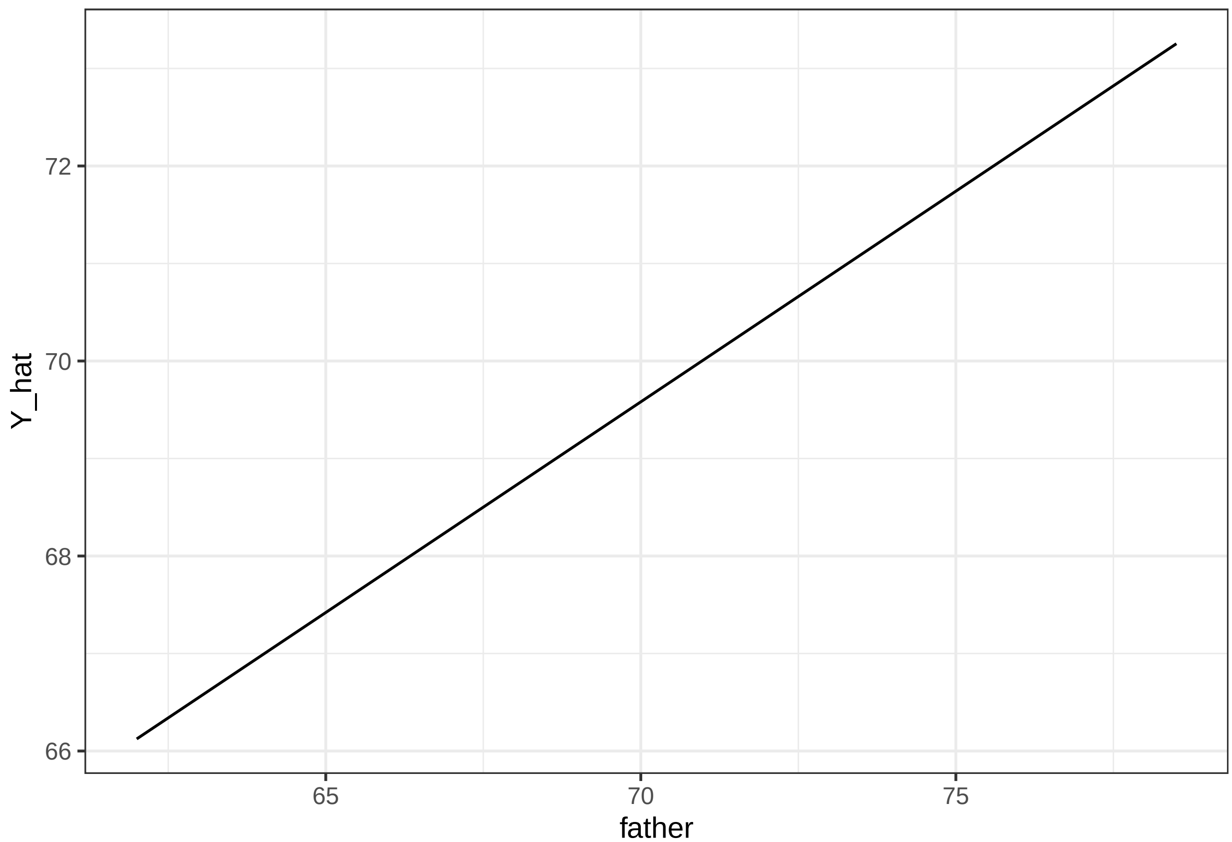
Predict 'y' directly:

```
fit <- galton_heights %>%
  lm(son ~ father, data = .)

Y_hat <- predict(fit, se.fit = TRUE)
names(Y_hat)
```

```
## [1] "fit"            "se.fit"         "df"             "residual.scale"
```

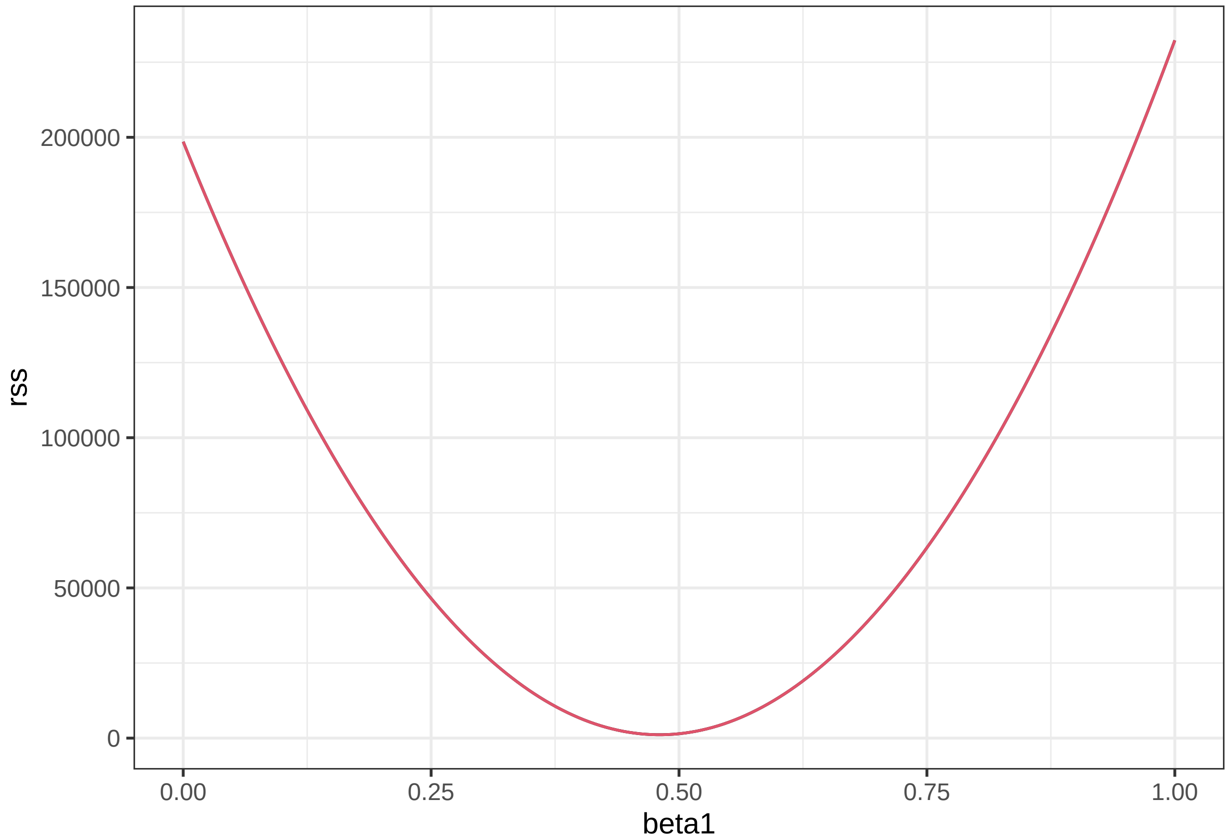
Plot best fit line:

```
galton_heights %>%
  mutate(Y_hat = predict(lm(son ~ father, data = .))) %>%
  ggplot(aes(father, Y_hat)) +
  geom_line()
```



Plot RSS with B0 fixed at 25:

```
beta1 <- seq(0, 1, len = nrow(galton_heights))
results <- data.frame(beta1 = beta1,
                      rss = sapply(beta1, rss, beta0 = 36))

results %>% ggplot(aes(beta1, rss)) +
  geom_line() +
  geom_line(aes(beta1, rss), col = 2)
```




```r
Teams <- Teams %>%
  filter(yearID %in% 1961:2001)

q3 <- Teams %>%
  mutate(R_g = R/G,
         BB_g = BB/G,
         HR_g = HR/G) %>%
  lm(R_g ~ BB_g + HR_g, data = .)
summary(q3)
```

```
##
## Call:
## lm(formula = R_g ~ BB_g + HR_g, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.87325 -0.24507 -0.01449  0.23866  1.24218
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.74430    0.08236   21.18   <2e-16 ***
## BB_g         0.38742    0.02701   14.34   <2e-16 ***
## HR_g         1.56117    0.04896   31.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3484 on 1023 degrees of freedom
## Multiple R-squared:  0.6503, Adjusted R-squared:  0.6496
## F-statistic: 951.2 on 2 and 1023 DF,  p-value: < 2.2e-16
```

```r
set.seed(1989, sample.kind = 'Rounding')
```

```
## Warning in set.seed(1989, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```r
options(digits = 3)
```

```r
female_heights <- GaltonFamilies %>%
  filter(gender == 'female') %>%
  group_by(family) %>%
  sample_n(1) %>%
  ungroup() %>%
  select(mother, childHeight) %>%
  rename(daughter = childHeight)
```

Fit a linear regression model predicting the mothers' heights using daughters' heights.

```r
q7 <- lm(mother ~ daughter, data = female_heights)
summary(q7)
```

```
##
## Call:
## lm(formula = mother ~ daughter, data = female_heights)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.659 -1.211 -0.211  1.496  7.176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  44.1785     4.4105   10.02  < 2e-16 ***
## daughter      0.3103     0.0686    4.53  1.1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.17 on 174 degrees of freedom
## Multiple R-squared:  0.105,  Adjusted R-squared:   0.1
## F-statistic: 20.5 on 1 and 174 DF,  p-value: 1.11e-05
```

```r
predict(q7, female_heights[1,'daughter'])
```

```
##    1
## 65.6
```

```r
female_heights[1,'mother']
```

```
## # A tibble: 1 x 1
##   mother
##    <dbl>
## 1     67
```

Want to assess the stability of BB and singles metrics. Want to generate two tables: one for 2002 and another for average of 1999-2001 seasons. Want to define per plate appearance statistics, keeping only players with more than 100 plate appearances.

Create 2002 table:

```r
bat_02 <- Batting %>%
  filter(yearID == 2002) %>%
  mutate(pa = AB + BB,
         singles = (H - X2B - X3B - HR) / pa,
         bb = BB / pa) %>%
  filter(pa >= 100) %>%
  select(playerID, singles, bb)
```

1999-2001:

```r
bat_99_01 <- Batting %>%
  filter(yearID %in% 1999:2001) %>%
  mutate(pa = AB + BB,
         singles = (H - X2B - X3B - HR) / pa,
         bb = BB / pa) %>%
  filter(pa >= 100) %>%
  select(playerID, yearID, singles, bb)

sum_99_01 <- bat_99_01 %>%
  group_by(playerID) %>%
  summarise(mean_singles = mean(singles),
            mean_bb = mean(bb),
            .groups = 'drop')

sum(sum_99_01$mean_bb > 0.2)
```

```
## [1] 3
```

Use inner_join() to combine bat_02 with the rate averages.

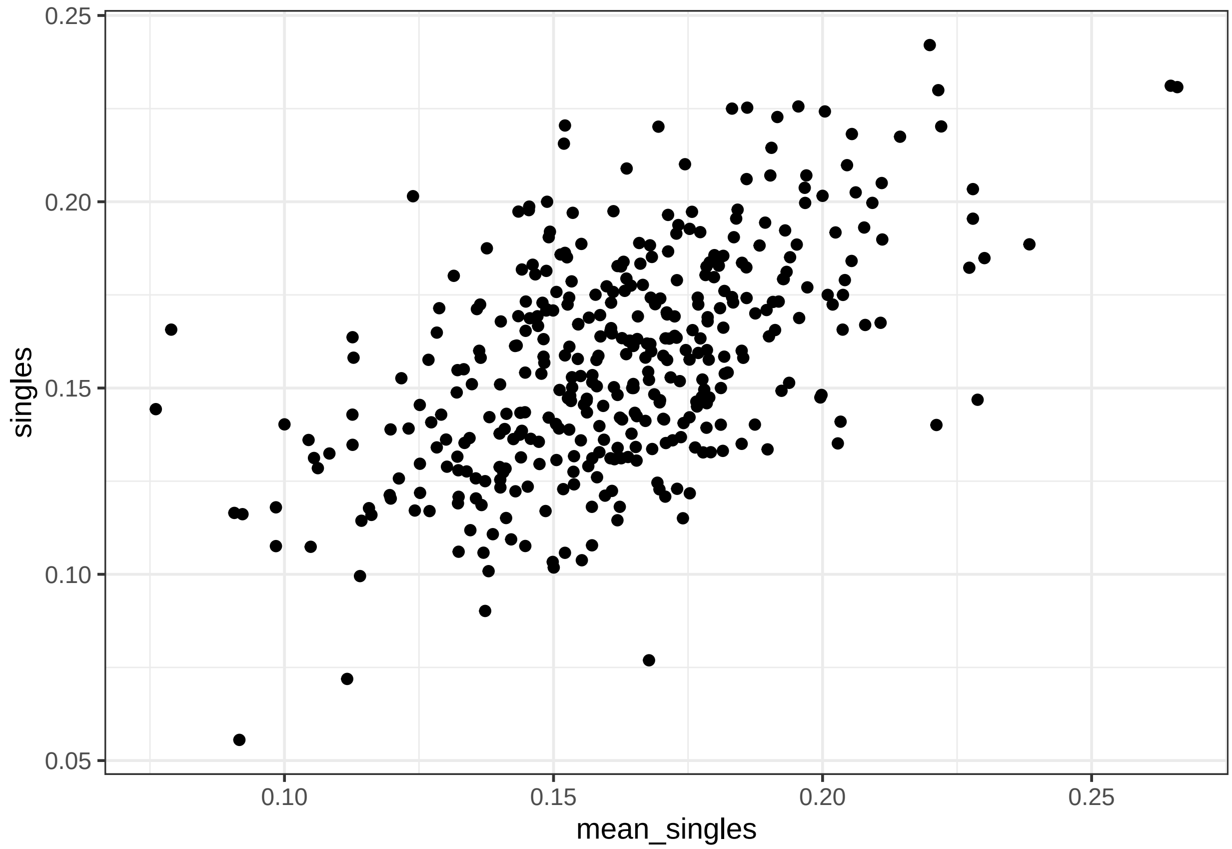
```r
bat <- inner_join(bat_02, sum_99_01, by = 'playerID')

head(bat)
```

```
##    playerID singles     bb mean_singles mean_bb
## 1 abernbr01   0.180 0.0512        0.178  0.0816
## 2 abreubo01   0.148 0.1538        0.153  0.1557
## 3 agbaybe01   0.118 0.0787        0.162  0.1153
## 4 alfoned01   0.197 0.1123        0.161  0.1228
## 5 alicelu01   0.160 0.1190        0.168  0.1001
## 6 alomaro01   0.182 0.0881        0.186  0.1222
```

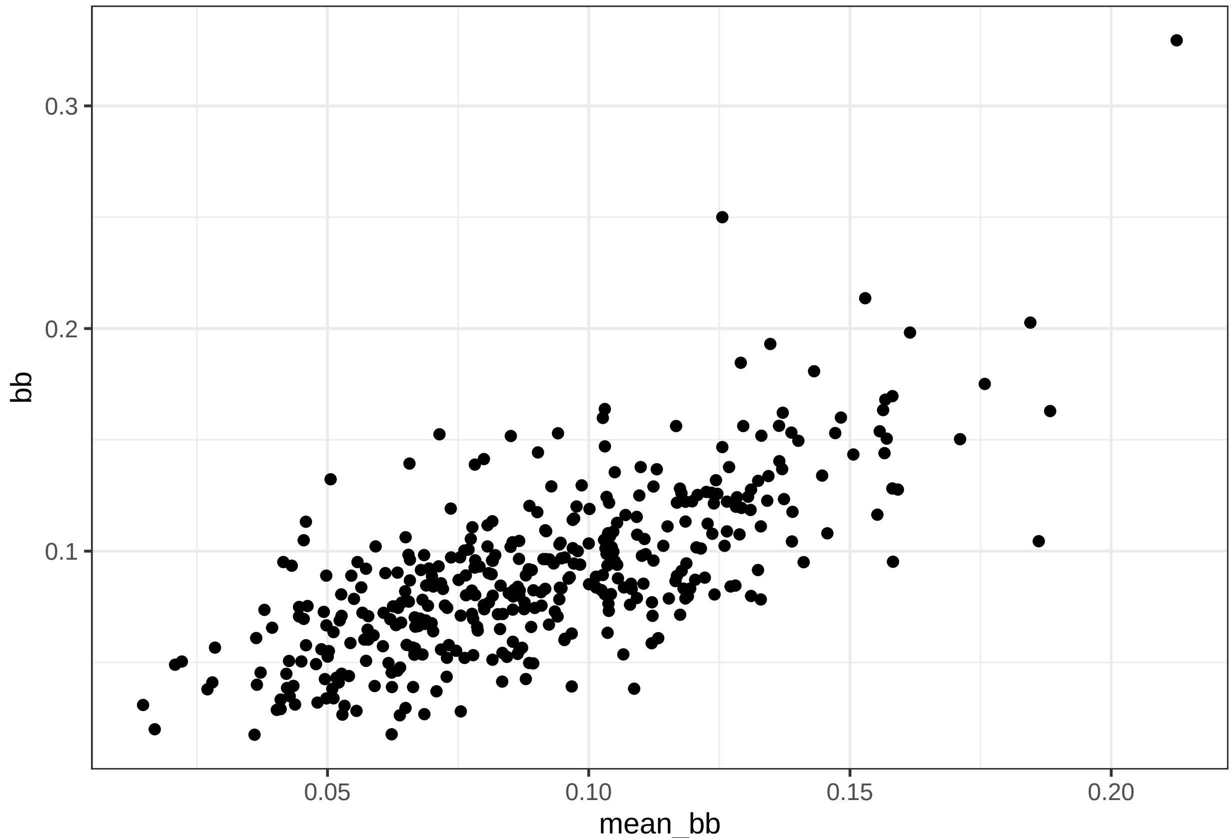
```r
cor(bat$bb, bat$mean_bb)
```

```
## [1] 0.717
```

```r
bat %>% ggplot(aes(mean_singles, singles)) +
  geom_point()
```




```r
bat %>% ggplot(aes(mean_bb, bb)) +
  geom_point()
```



Fit a linear model to predict 2002 singles given 1999-2001 mean_singles:

```
q12 <- lm(singles ~ mean_singles, data = bat)
summary(q12)
```

```
##
## Call:
## lm(formula = singles ~ mean_singles, data = bat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08380 -0.01673 -0.00108  0.01666  0.06894
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.06206    0.00742    8.37  1.1e-15 ***
## mean_singles  0.58813    0.04511   13.04  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0247 on 390 degrees of freedom
## Multiple R-squared:  0.304,  Adjusted R-squared:  0.302
## F-statistic:  170 on 1 and 390 DF,  p-value: <2e-16
```

```r
q12b <- lm(bb ~ mean_bb, data = bat)
summary(q12b)
```

```
##
## Call:
## lm(formula = bb ~ mean_bb, data = bat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06738 -0.01695 -0.00136  0.01527  0.13777
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01548    0.00391    3.96    9e-05 ***
## mean_bb      0.82905    0.04076   20.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0265 on 390 degrees of freedom
## Multiple R-squared:  0.515,  Adjusted R-squared:  0.514
## F-statistic:  414 on 1 and 390 DF,  p-value: <2e-16
```

tibbles, do, and broom

```r
set.seed(1, sample.kind = 'Rounding')
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
galton <- GaltonFamilies %>%
  group_by(family, gender) %>%
  sample_n(1) %>%
  ungroup() %>%
  gather(parent, parentHeight, father:mother) %>%
  mutate(child = ifelse(gender == 'female',
                        'daughter',
                        'son')) %>%
  unite(pair, c('parent', 'child'))

galton
```

```
## # A tibble: 710 x 8
##    family midparentHeight children childNum gender childHeight pair
##    <fct>            <dbl>    <int>    <int> <fct>        <dbl> <chr>
##  1 001               75.4        4        2 female        69.2 fath~
##  2 001               75.4        4        1 male          73.2 fath~
##  3 002               73.7        4        4 female        65.5 fath~
##  4 002               73.7        4        2 male          72.5 fath~
##  5 003               72.1        2        2 female        68   fath~
##  6 003               72.1        2        1 male          71   fath~
```

```
##  7 004                72.1        5        5 female        63   fath~
##  8 004                72.1        5        2 male          68.5 fath~
##  9 005                69.1        6        5 female        62.5 fath~
## 10 005                69.1        6        1 male          72   fath~
## # ... with 700 more rows, and 1 more variable: parentHeight <dbl>
```

Group by 'pair' and summarize the number of observations in each group.

```
galton %>%
  group_by(pair) %>%
  summarise(n = n(), .groups = 'drop')
```

```
## # A tibble: 4 x 2
##   pair                n
##   <chr>           <int>
## 1 father_daughter   176
## 2 father_son        179
## 3 mother_daughter   176
## 4 mother_son        179
```

```
galton %>%
  group_by(pair) %>%
  summarise(cc = cor(childHeight, parentHeight), .groups='drop') %>%
  arrange(desc(cc))
```

```
## # A tibble: 4 x 2
##   pair               cc
##   <chr>           <dbl>
## 1 father_son      0.430
## 2 father_daughter 0.401
## 3 mother_daughter 0.383
## 4 mother_son      0.343
```

```
library(broom)
```

```
galton %>%
  group_by(pair) %>%
  do(tidy(lm(childHeight ~ parentHeight, data = .), conf.int = TRUE))
```

```
## # A tibble: 8 x 8
## # Groups:   pair [4]
##   pair        term      estimate std.error statistic  p.value conf.low conf.high
##   <chr>       <chr>        <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 father_dau~ (Interce~   40.1      4.16        9.65 6.50e-18   31.9      48.3
## 2 father_dau~ parentHe~    0.345    0.0599      5.77 3.56e- 8    0.227     0.464
## 3 father_son  (Interce~   38.6      4.84        7.98 1.81e-13   29.1      48.2
## 4 father_son  parentHe~    0.443    0.0700      6.33 1.94e- 9    0.305     0.581
## 5 mother_dau~ (Interce~   38.9      4.62        8.41 1.46e-14   29.7      48.0
## 6 mother_dau~ parentHe~    0.394    0.0720      5.47 1.56e- 7    0.252     0.536
## 7 mother_son  (Interce~   44.9      5.02        8.94 4.96e-16   35.0      54.8
## 8 mother_son  parentHe~    0.381    0.0784      4.86 2.59e- 6    0.226     0.535
```

Building a baseball team

Regression with BB, singles, doubles, triples, and HR

```
fit <- Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(BB = BB / G,
         singles = (H - X2B - X3B - HR) / G,
         doubles = X2B / G,
         triples = X3B / G,
         HR = HR / G,
         R = R / G) %>%
  lm(R ~ BB + singles + doubles + triples + HR, data = .)

coefs <- tidy(fit, conf.int = TRUE)
coefs
```

```
## # A tibble: 6 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)   -2.77     0.0862     -32.1 4.76e-157   -2.94     -2.60
## 2 BB             0.371    0.0117      31.6 1.87e-153    0.348     0.394
## 3 singles        0.519    0.0127      40.8 8.67e-217    0.494     0.544
## 4 doubles        0.771    0.0226      34.1 8.44e-171    0.727     0.816
## 5 triples        1.24     0.0768      16.1 2.12e- 52    1.09      1.39
## 6 HR             1.44     0.0243      59.3 0.           1.40      1.49
```

```
Team_A <- 2*0.371 + 4*0.519 + 0.771 + 1.443
Team_B <- 0.371 + 6*0.519 + 2*0.771 + 1.24

Team_A >= Team_B
```

```
## [1] FALSE
```

Fit a multivariate linear regression model to obtain the effects of BB and HR on Runs in 1971. Use the tidy() function in the broom package to obtain the results in a dataframe.

```
Teams %>%
  filter(yearID == 1971) %>%
  lm(R ~ BB + HR, data = .) %>%
  tidy(conf.int = TRUE)
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)  257.      112.         2.31 0.0314   25.3      489.
## 2 BB             0.414     0.210      1.97 0.0625   -0.0237     0.852
## 3 HR             1.30      0.431      3.01 0.00673   0.399      2.19
```

Repeat above to find effects for every year from 1961 - 2018

```r
Teams %>%
  filter(yearID %in% 1961:2018) %>%
  group_by(yearID) %>%
  do(tidy(lm(R ~ BB + HR, data = .), conf.int = TRUE))
```

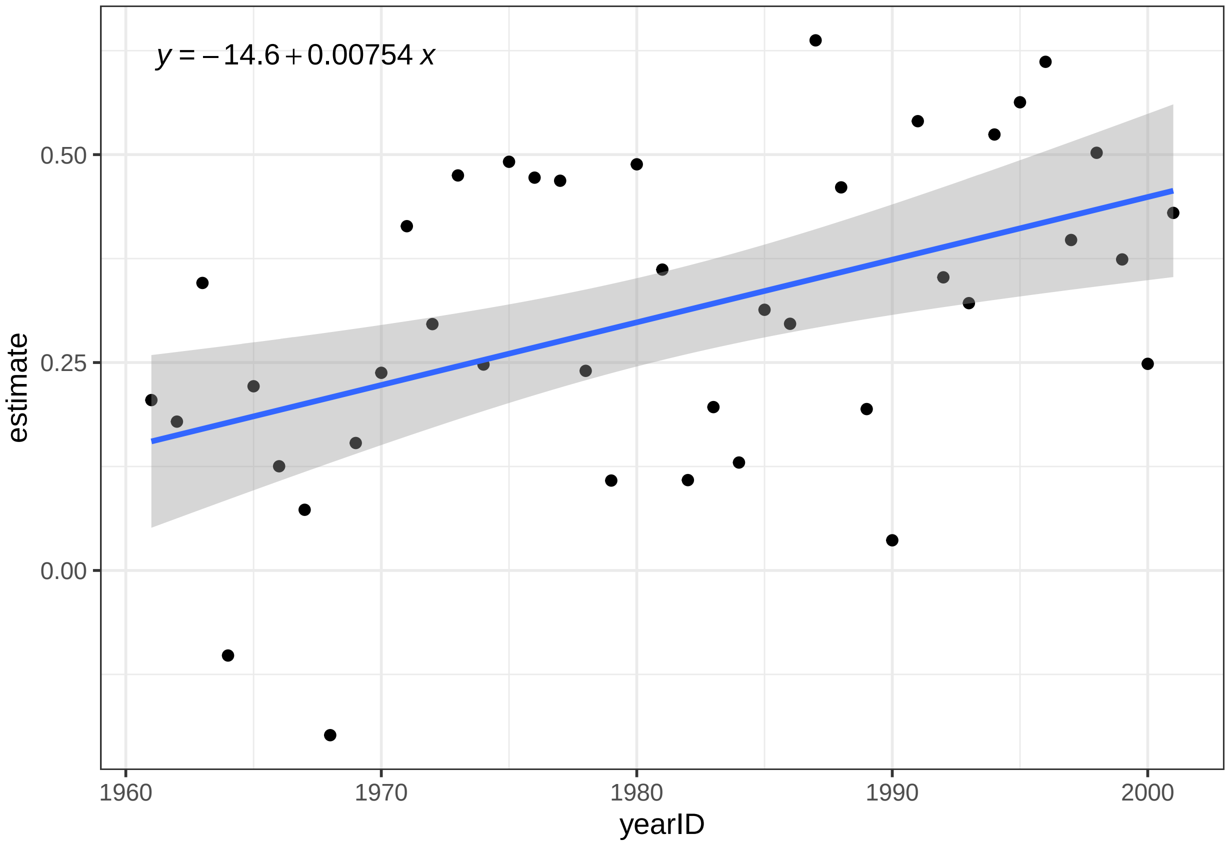
```
## # A tibble: 123 x 8
## # Groups:   yearID [41]
##    yearID term        estimate std.error statistic  p.value conf.low conf.high
##     <int> <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1   1961 (Intercept)  455.       89.8       5.07  0.000139 263.       646.
##  2   1961 BB             0.205     0.156     1.32  0.208     -0.127     0.536
##  3   1961 HR             0.999     0.300     3.33  0.00455    0.360     1.64
##  4   1962 (Intercept)  448.      149.        3.01  0.00789  134.       762.
##  5   1962 BB             0.179     0.283     0.632 0.536     -0.418     0.776
##  6   1962 HR             1.18      0.504     2.34  0.0316     0.117     2.24
##  7   1963 (Intercept)  281.      118.        2.38  0.0293    31.9      530.
##  8   1963 BB             0.346     0.242     1.43  0.171     -0.164     0.855
##  9   1963 HR             1.42      0.299     4.75  0.000186   0.790     2.05
## 10   1964 (Intercept)  512.      113.        4.54  0.000293 274.       751.
## # ... with 113 more rows
```

Make a scatter plot for effect of BB on runs over time with trend line

```r
library(ggpmisc)
```

```
##
## Attaching package: 'ggpmisc'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```r
Teams %>%
  filter(yearID %in% 1961:2018) %>%
  group_by(yearID) %>%
  do(tidy(lm(R ~ BB + HR, data = .), conf.int = TRUE)) %>%
  filter(term == 'BB') %>%
  ggplot(aes(x = yearID, y = estimate)) +
  geom_point() +
  geom_smooth(method = 'lm', formula = y~x) +
  stat_poly_eq(formula = y ~ x,
               aes(label = paste(..eq.label.., sep = '~~~')),
               parse = TRUE)
```



Fit a linear model on the results from above to determine the effect of year on impact of BB

```
data("Teams")
q11 <- Teams %>%
  filter(yearID %in% 1961:2018) %>%
  group_by(yearID) %>%
  do(tidy(lm(R ~ BB + HR, data = .), conf.int = TRUE)) %>%
  filter(term == 'BB')

tidy(summary(lm(estimate ~ yearID, data = q11)))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept) -6.75      2.57        -2.62 0.0112
## 2 yearID       0.00355   0.00129      2.75 0.00807
```

Average number of team plate appearances per game

```
pa_per_game <- Batting %>%
  filter(yearID == 2002) %>%
  group_by(teamID) %>%
  summarise(pa_per_game = sum(AB + BB) / max(G), .groups = 'drop') %>%
  pull(pa_per_game) %>%
  mean

pa_per_game
```

```
## [1] 38.7
```

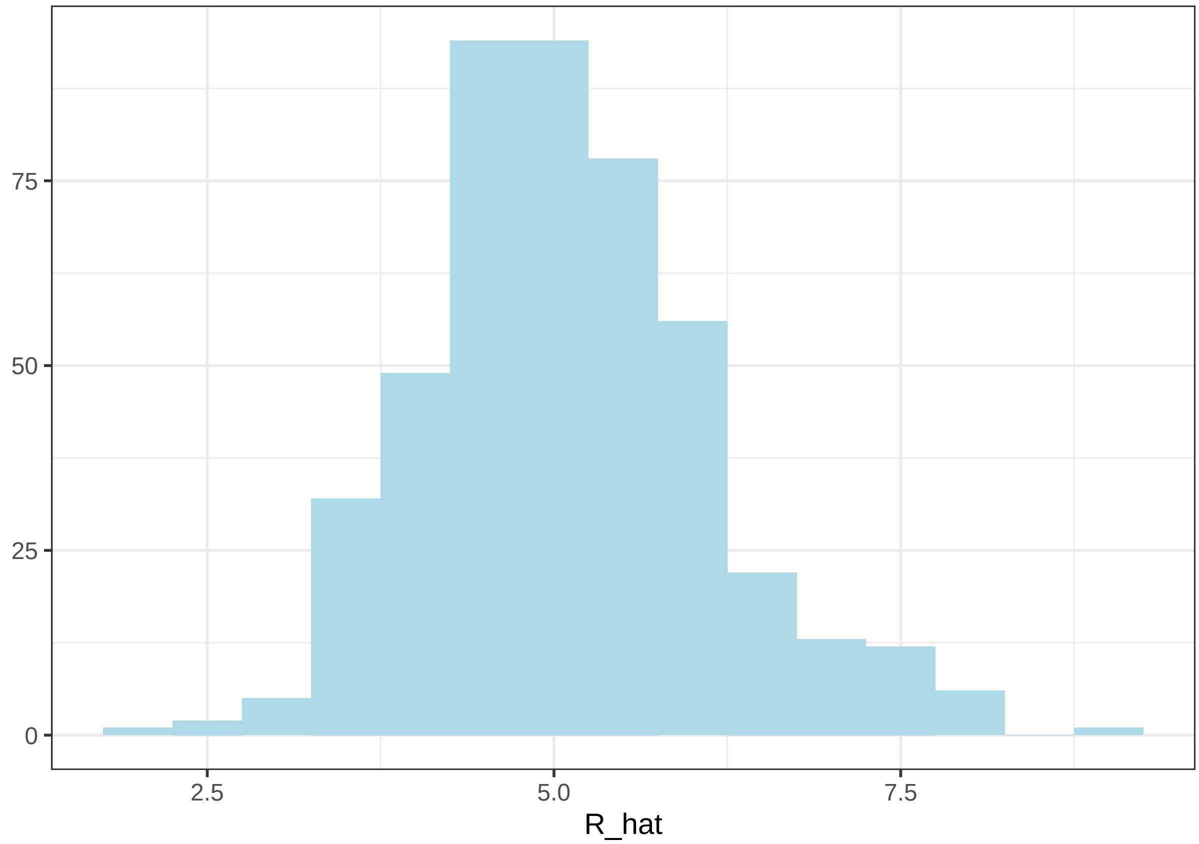
Per-plate rates for players available in 2002 using prior data

```r
players <- Batting %>%
  filter(yearID %in% 1999:2001) %>%
  group_by(playerID) %>%
  mutate(PA = BB + AB) %>%
  summarise(G = sum(PA) / pa_per_game,
            BB = sum(BB) / G,
            singles = sum(H - X2B - X3B - HR) / G,
            doubles = sum(X2B) / G,
            triples = sum(X3B) / G,
            HR = sum(HR) / G,
            AVG = sum(H) / sum(AB),
            PA = sum(PA),
            .groups = 'drop') %>%
  filter(PA >= 300) %>%
  select(-G) %>%
  mutate(R_hat = predict(fit, newdata = .))

head(players)
```

```
## # A tibble: 6 x 9
##   playerID     BB singles doubles triples    HR   AVG    PA R_hat
##   <chr>     <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <int> <dbl>
## 1 abbotje01  3.28    6.21    2.03   0.113 0.565 0.252   343  4.20
## 2 abbotku01  2.41    5.87    1.93   0.241 1.13  0.252   482  4.59
## 3 abernbr01  3.16    6.91    1.99   0.117 0.585 0.270   331  4.52
## 4 abreubo01  6.03    5.91    2.39   0.478 1.45  0.313  2025  7.08
## 5 agbaybe01  4.53    6.31    1.89   0.223 1.30  0.284  1044  5.80
## 6 alexama02  2.26    6.39    1.48   0.492 0.393 0.240   394  3.71
```

Plot player-specific predicted runs

```r
qplot(R_hat, data = players,
      geom = 'histogram',
      binwidth = 0.5,
      fill = I("lightblue"))
```



Add 2002 salaries to each player

```
players <- Salaries %>%
  filter(yearID == 2002) %>%
  select(playerID, salary) %>%
  right_join(players, by = 'playerID')

head(players)
```

```
##    playerID  salary   BB singles doubles triples    HR   AVG   PA R_hat
## 1 anderga01 5000000 1.63    6.91    2.20   0.134 1.608 0.292 2024  5.61
## 2 eckstda01  280000 2.67    8.31    1.61   0.124 0.248 0.285  625  4.29
## 3 erstada01 6250000 3.25    7.43    1.80   0.225 0.882 0.291 2065  5.24
## 4 fabrejo01  500000 2.57    6.18    1.25   0.278 0.556 0.228  558  3.50
## 5 fullmbr01 4000000 2.42    6.00    2.53   0.134 1.586 0.282 1441  5.65
## 6   gilbe01  400000 2.82    6.47    1.86   0.320 0.897 0.266  605  4.76
```

Add defensive position

```
position_names <- c('G_p','G_c','G_1b',
                    'G_2b','G_3b','G_ss',
                    'G_lf','G_cf','G_rf')

temp_tab <- Appearances %>%
  filter(yearID == 2002) %>%
```

```r
  group_by(playerID) %>%
  summarise_at(position_names, sum) %>%
  ungroup()

pos <- temp_tab %>%
  select(position_names) %>%
  apply(., 1, which.max)  # get the position the player played most often
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(position_names)` instead of `position_names` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
players <- data_frame(playerID = temp_tab$playerID,
                      POS = position_names[pos]) %>%
  mutate(POS = str_to_upper(str_remove(POS, 'G_'))) %>%
  filter(POS != 'P') %>%
  right_join(players, by = 'playerID') %>%
  filter(!is.na(POS) & !is.na(salary))
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```r
head(players)
```

```
## # A tibble: 6 x 11
##   playerID  POS    salary    BB singles doubles triples    HR   AVG    PA R_hat
##   <chr>     <chr>   <int> <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <int> <dbl>
## 1 abernbr01 2B     215000  3.16    6.91    1.99  0.117  0.585 0.270   331  4.52
## 2 abreubo01 RF    6333333  6.03    5.91    2.39  0.478  1.45  0.313  2025  7.08
## 3 agbaybe01 LF     600000  4.53    6.31    1.89  0.223  1.30  0.284  1044  5.80
## 4 alfoned01 3B    6200000  4.81    6.31    2.15  0.0625 1.44  0.293  1860  6.10
## 5 alicelu01 2B     800000  3.55    7.16    1.65  0.387  0.419 0.273  1201  4.62
## 6 alomaro01 2B    7939664  4.73    7.20    2.22  0.331  1.23  0.323  1991  6.62
```

Top 10 players:

```r
players <- Master %>%
  select(playerID, nameFirst, nameLast, debut) %>%
  mutate(debut = as.Date(debut)) %>%
  right_join(players, by = 'playerID') %>%
  select(nameFirst, nameLast, POS, debut, salary, R_hat) %>%
  arrange(desc(R_hat)) %>%
  top_n(10)
```

```
## Selecting by R_hat
```

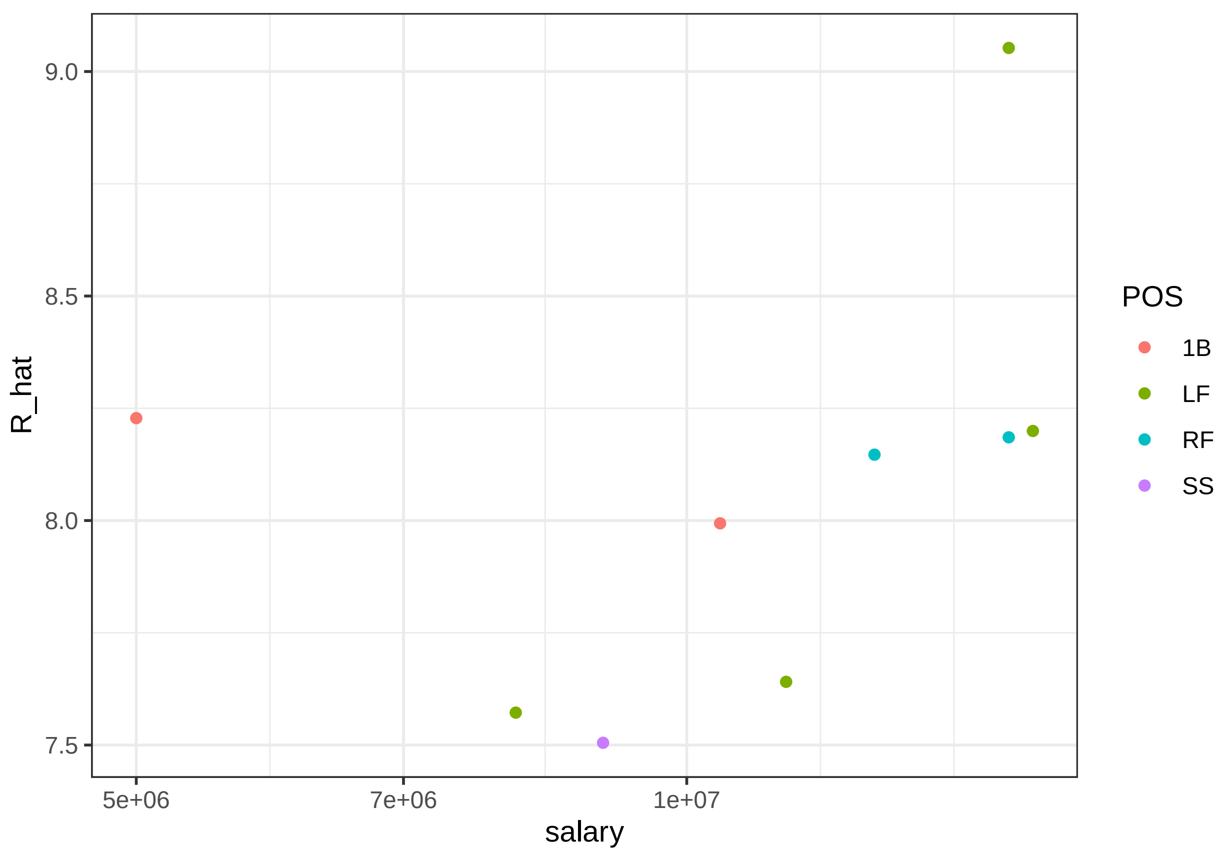
```r
players
```

```
##     nameFirst     nameLast POS      debut    salary R_hat
## 1       Barry        Bonds  LF 1986-05-30 15000000  9.05
## 2        Todd       Helton  1B 1997-08-02  5000000  8.23
## 3       Manny      Ramirez  LF 1993-09-02 15462727  8.20
## 4       Sammy         Sosa  RF 1989-06-16 15000000  8.19
## 5       Larry       Walker  RF 1989-08-16 12666667  8.15
## 6       Jason       Giambi  1B 1995-05-08 10428571  7.99
## 7     Chipper        Jones  LF 1993-09-11 11333333  7.64
## 8       Brian        Giles  LF 1995-09-16  8063003  7.57
## 9      Albert       Pujols  LF 2001-04-02   600000  7.54
## 10      Nomar Garciaparra  SS 1996-08-31  9000000  7.51
```

Remake plot without rookie players

```r
library(lubridate)
```

```r
players %>%
  filter(year(debut) < 1998) %>%
  ggplot(aes(salary, R_hat, color = POS)) +
  geom_point() +
  scale_x_log10()
```



Only showing data from top 10

Assessment

```
data("Teams")
Teams_small <- Teams %>%
  filter(yearID %in% 1961:2001) %>%
  mutate(avg_attendance = attendance/G)
```

```
Teams_small %>%
  mutate(R_g = R / G,
         HR_g = HR / G) %>%
  do(tidy(lm(avg_attendance ~ HR_g, data = .)))
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    3783.      502.      7.53 1.12e-13
## 2 HR_g           8113.      566.     14.3  1.14e-42
```

Use number of wins to predict avg_attendance; do not normalize for number of games.

```
Teams_small %>%
  do(tidy(lm(avg_attendance ~ yearID, data = .)))
```

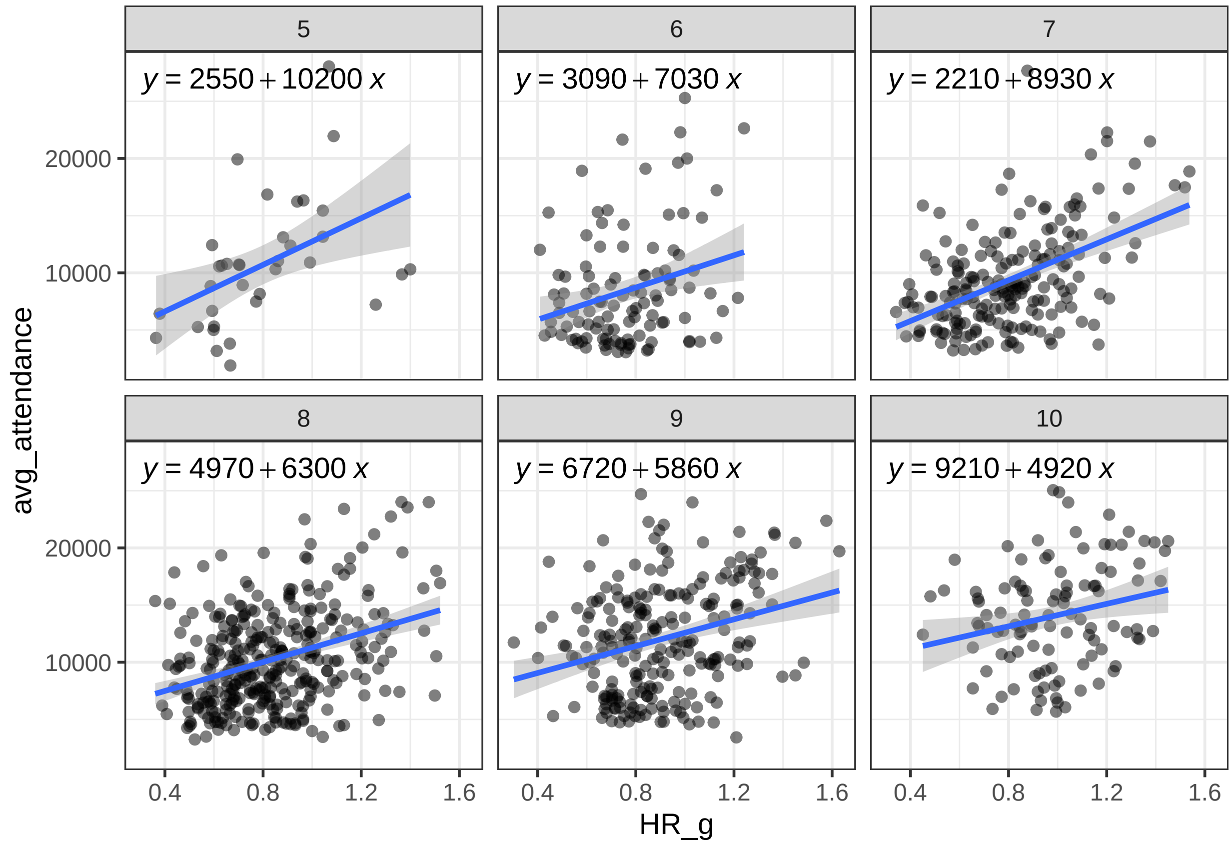
```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -473937.   20632.      -23.0 1.63e-94
## 2 yearID          244.      10.4      23.5 5.90e-98
```

```
Teams_small <- Teams_small %>%
  mutate(R_g = R / G,
         HR_g = HR / G)

cor(Teams_small$HR_g, Teams_small$W)
```

```
## [1] 0.274
```

Q3 Stratify Teams_small by wins: divide number of wins by 10 and then round to the nearest integer. Keep only strata 5 - 10, which have 20 or more data points.

```
Teams_small %>%
  mutate(r_r = round(W/10)) %>%
  filter(r_r %in% 5:10) %>%
  ggplot(aes(HR_g, avg_attendance)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = 'lm', formula = y~x) +
  stat_poly_eq(formula = y ~ x,
               aes(label = paste(..eq.label.., sep = '~~~')),
               parse = TRUE) +
  facet_wrap(~r_r)
```



Q4 Fit a multivariate regression determining the effects of runs per game, home runs per game, wins, and year on average attendance. Use original Teams_small W col (not strata)

```
q4 <- Teams_small %>%
  do(tidy(lm(avg_attendance ~ R_g + HR_g + W + yearID, data = .)))
q4
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -456674.   21815.     -20.9   3.00e-81
## 2 R_g             322.     331.       0.972 3.31e- 1
## 3 HR_g           1798.     690.       2.61  9.24e- 3
## 4 W               117.       9.88    11.8   2.79e-30
## 5 yearID          230.      11.2     20.6   7.10e-79
```

Q5 Suppose a team averaged 5 runs per game, 1.2 home runs per game, and won 80 games in a season. What would the team's avg attendance be in 1960?

```
q5 <- lm(avg_attendance ~ R_g + HR_g + W + yearID,
         data = Teams_small)

q5_test <- data.frame(R_g = 5, HR_g = 1.2, W = 80, yearID = 2002)

predict(q5, q5_test)
```

```
##     1
## 16149
```

Q6 Use the model from q4 to predict average attendance for teams in 2002 in the original Teams dataframe. What is the correlation between the predicted attendance and actual?

```r
model <- lm(avg_attendance ~ R_g + HR_g + W + yearID,
            data = Teams_small)

Teams2002 <- Teams %>%
  filter(yearID == 2002) %>%
  mutate(R_g = R / G,
         HR_g = HR / G)

Teams2002 <- Teams2002 %>%
  mutate(pred_attend = predict(model, Teams2002))
```

```r
cor(Teams2002$attendance, Teams2002$pred_attend)
```

```
## [1] 0.519
```

Assessment #2

```r
library(dslabs)
data("research_funding_rates")
```

```r
head(research_funding_rates)
```

```
##          discipline applications_total applications_men applications_women
## 1  Chemical sciences                122               83                 39
## 2  Physical sciences                174              135                 39
## 3            Physics                 76               67                  9
## 4         Humanities                396              230                166
## 5 Technical sciences                251              189                 62
## 6  Interdisciplinary                183              105                 78
##   awards_total awards_men awards_women success_rates_total success_rates_men
## 1           32         22           10                26.2              26.5
## 2           35         26            9                20.1              19.3
## 3           20         18            2                26.3              26.9
## 4           65         33           32                16.4              14.3
## 5           43         30           13                17.1              15.9
## 6           29         12           17                15.8              11.4
##   success_rates_women
## 1                25.6
## 2                23.1
## 3                22.2
## 4                19.3
## 5                21.0
## 6                21.8
```

```r
sum(research_funding_rates$applications_women) - sum(research_funding_rates$awards_women)
```

```
## [1] 1011
```

```r
two_by_two <- research_funding_rates %>%
  select(-discipline) %>%
  summarize_all(funs(sum)) %>%
  summarise(yes_men = awards_men,
            no_men = applications_men - awards_men,
            yes_women = awards_women,
            no_women = applications_women - awards_women) %>%
  gather %>%
  separate(key, c('awarded','gender')) %>%
  spread(gender, value)
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```r
two_by_two
```

```
##   awarded  men women
## 1      no 1345  1011
## 2     yes  290   177
```

```r
two_by_two$men[2] / sum(two_by_two$men) *100
```

```
## [1] 17.7
```

```r
two_by_two$women[2] / sum(two_by_two$women) *100
```

```
## [1] 14.9
```

Run a chi-squared test on the two-by-two to determine whether the difference in the two success rates is significant.

```r
two_by_two %>%
  select(-awarded) %>%
  chisq.test() %>%
  tidy()
```

```
## # A tibble: 1 x 4
##   statistic p.value parameter method
##       <dbl>   <dbl>     <int> <chr>
## 1      3.81  0.0509         1 Pearson's Chi-squared test with Yates' continuity~
```

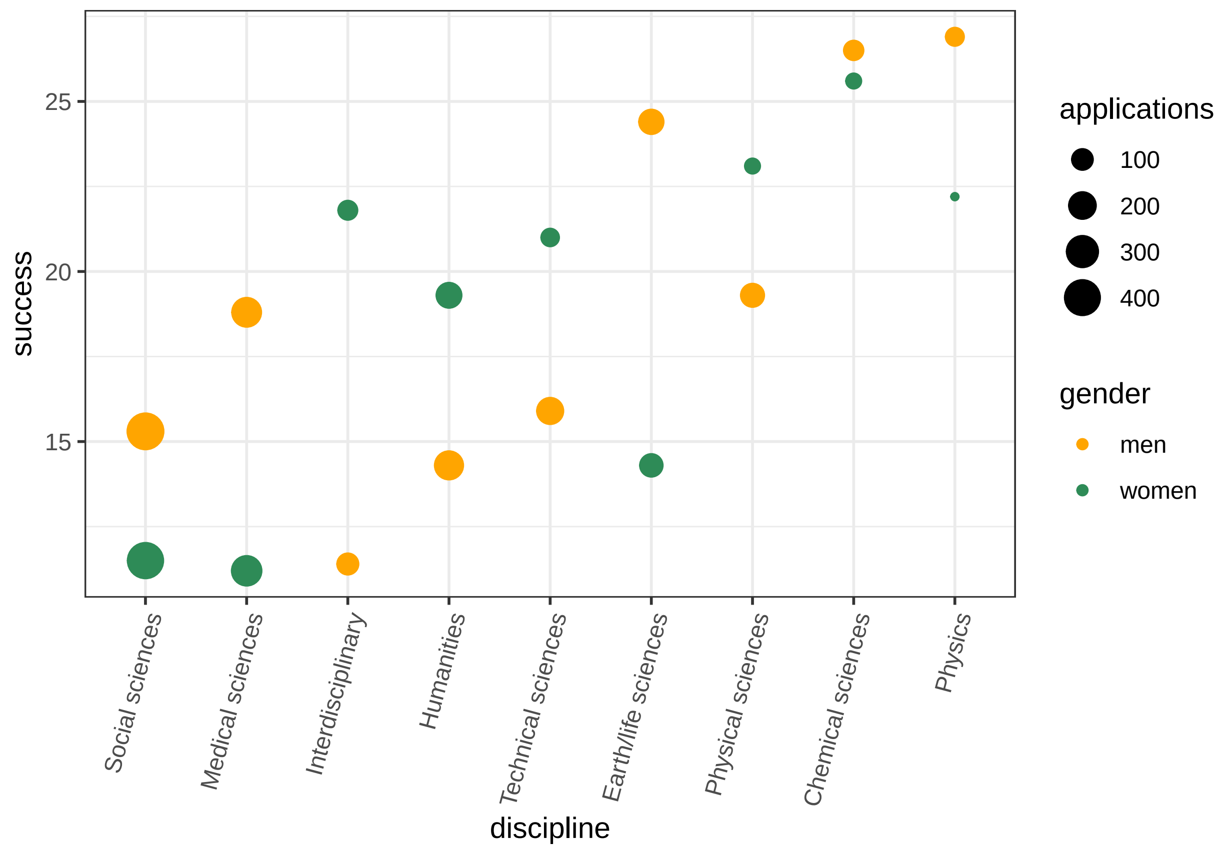
```
dat <- research_funding_rates %>%
  mutate(discipline = reorder(discipline, success_rates_total)) %>%
  rename(success_total = success_rates_total,
         success_men = success_rates_men,
         success_women = success_rates_women) %>%
  gather(key, value, -discipline) %>%
  separate(key, c('type','gender')) %>%
  spread(type, value) %>%
  filter(gender != 'total')

dat
```

```
##              discipline gender applications awards success
## 1       Social sciences    men          425     65    15.3
## 2       Social sciences  women          409     47    11.5
## 3      Medical sciences    men          245     46    18.8
## 4      Medical sciences  women          260     29    11.2
## 5     Interdisciplinary    men          105     12    11.4
## 6     Interdisciplinary  women           78     17    21.8
## 7            Humanities    men          230     33    14.3
## 8            Humanities  women          166     32    19.3
## 9    Technical sciences    men          189     30    15.9
## 10   Technical sciences  women           62     13    21.0
## 11 Earth/life sciences    men          156     38    24.4
## 12 Earth/life sciences  women          126     18    14.3
## 13    Physical sciences    men          135     26    19.3
## 14    Physical sciences  women           39      9    23.1
## 15    Chemical sciences    men           83     22    26.5
## 16    Chemical sciences  women           39     10    25.6
## 17              Physics    men           67     18    26.9
## 18              Physics  women            9      2    22.2
```

To check if this is a case of Simpson's paradox, plot the success rate vs disciplines, which have been ordered by overall success, with colors to denote the genders and size to denote the number of applications. In which fields do men have a higher success rate than women?

```
library(RColorBrewer)
```

```
dat %>% ggplot(aes(discipline, success, color = gender, size = applications)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 75,
                                   hjust = 1)) +
  scale_color_manual(values = c("women" = 'seagreen',
                                "men" = 'orange'))
```



25
20
15
success
Social sciences
Medical sciences
Interdisciplinary
Humanities
Technical sciences
Earth/life sciences
Physical sciences
Chemical sciences
Physics
discipline
applications
100
200
300
400
gender
men
women