SDS 439 - Homework 05

Due Mar 31, 1:00 pm

Pinewood Derby Revisited

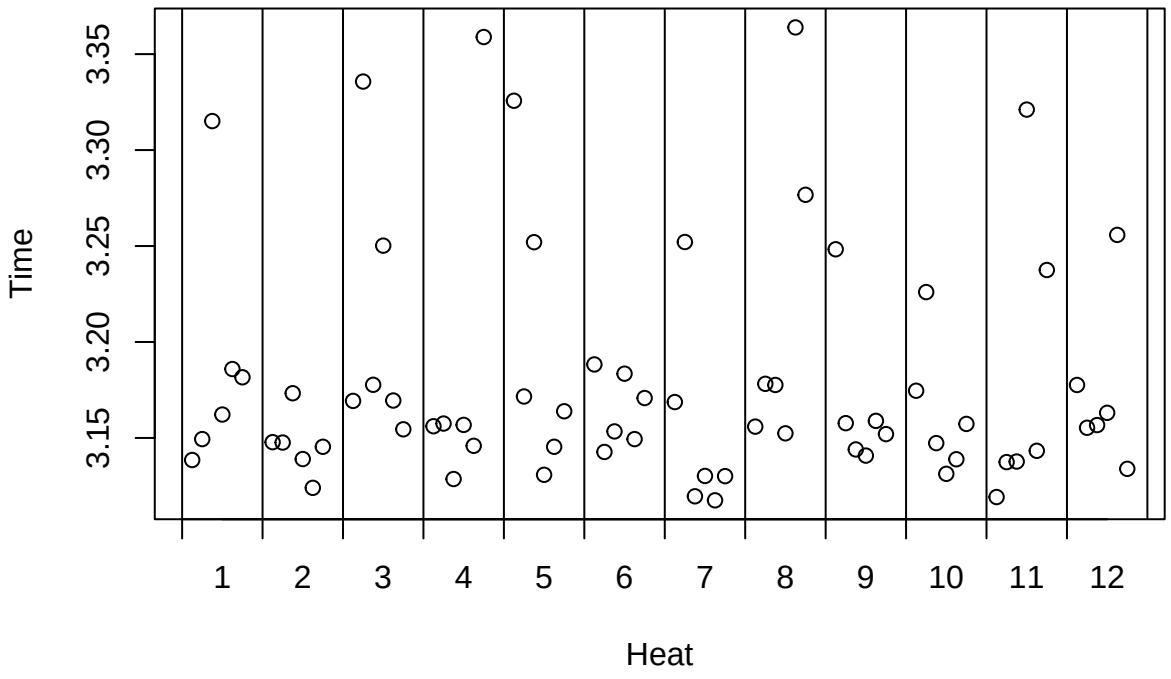
We return to the pinewood derby data, but this time with more complex models at our disposal.

1. To refamiliarize yourself with the data, make one plot of the data, showing some of the major features. This may be a plot from the previous assignment, but see if you can improve it in some way.

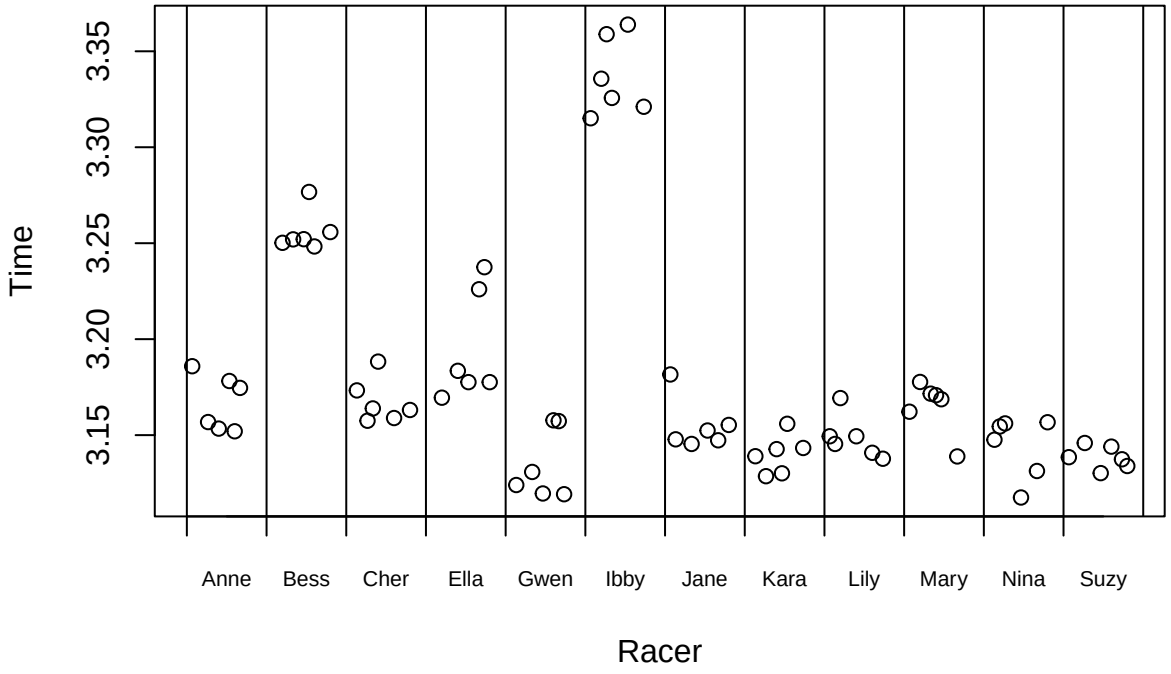
```
derby <- read.csv("../datasets/pinewood_derby.csv")
head(derby)
```

```
##   lane racer heat   time place
## 1    1  Suzy    1 3.1385     1
## 2    2  Lily    1 3.1494     2
## 3    4  Mary    1 3.1622     3
## 4    6  Jane    1 3.1816     4
## 5    5  Anne    1 3.1859     5
## 6    3  Ibby    1 3.3151     6
```

```
plot( derby$heat + derby$lane/8, derby$time, axes = FALSE, xlab = "Heat", ylab = "Time" )
axis(1, at = 1:12, labels = rep("",12 ) )
axis(1, at = (1:12) + 0.5, labels = 1:12, lwd.ticks = 0 )
box()
axis(2)
for(j in 1:13) abline( v = j )
```



```
plot( as.numeric(as.factor(derby$racer)) + derby$heat/15, derby$time, axes = FALSE, xlab = "Racer", ylab
axis(1, at = 1:12, labels = rep("",12 ) )
axis(1, at = (1:12) + 0.5, labels = levels(as.factor(derby$racer)), lwd.ticks = 0, cex.axis =0.7)
box()
axis(2)
for(j in 1:13) abline( v = j )
```



2. The official winner of the competition was determined by which racer had the lowest average time over their six runs. Use both `tapply` and `dplyr`'s `group_by` and `summarize` functions to calculate the average times for each racer.

```
tapply( derby$time, derby$racer, mean )
```

```
##     Anne     Bess     Cher     Ella     Gwen     Ibby     Jane     Kara
## 3.166817 3.255850 3.167500 3.195283 3.134767 3.336733 3.154967 3.139933
##     Lily     Mary     Nina     Suzy
## 3.148667 3.164983 3.143950 3.138317
```

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
derby |> group_by( racer ) |> summarize( avg = mean( time ) )
```

```
## # A tibble: 12 x 2
##    racer   avg
```

```
##    <chr> <dbl>
##  1 Anne   3.17
##  2 Bess   3.26
##  3 Cher   3.17
##  4 Ella   3.20
##  5 Gwen   3.13
##  6 Ibby   3.34
##  7 Jane   3.15
##  8 Kara   3.14
##  9 Lily   3.15
## 10 Mary   3.16
## 11 Nina   3.14
## 12 Suzy   3.14
```

3. Convert the racer column to a factor, and set its reference level to be the racer with the fastest average time. Then fit the one-factor model in racer and print the results. If you do it correctly, all of the coefficients should be positive. Explain why.

```r
derby$racer <- as.factor( derby$racer )
derby$racer <- relevel( derby$racer, ref = "Gwen" )
m1 <- lm( time ~ racer, data = derby )
summary(m1)
```

```
##
## Call:
## lm(formula = time ~ racer, data = derby)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.026450 -0.010004 -0.002675  0.007633  0.042217
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.134767   0.006365 492.490  < 2e-16 ***
## racerAnne   0.032050   0.009002   3.560 0.000732 ***
## racerBess   0.121083   0.009002  13.451  < 2e-16 ***
## racerCher   0.032733   0.009002   3.636 0.000576 ***
## racerElla   0.060517   0.009002   6.723 7.42e-09 ***
## racerIbby   0.201967   0.009002  22.437  < 2e-16 ***
## racerJane   0.020200   0.009002   2.244 0.028532 *
## racerKara   0.005167   0.009002   0.574 0.568135
## racerLily   0.013900   0.009002   1.544 0.127808
## racerMary   0.030217   0.009002   3.357 0.001373 **
## racerNina   0.009183   0.009002   1.020 0.311738
## racerSuzy   0.003550   0.009002   0.394 0.694705
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01559 on 60 degrees of freedom
## Multiple R-squared:  0.9419, Adjusted R-squared:  0.9312
## F-statistic: 88.39 on 11 and 60 DF,  p-value: < 2.2e-16
```

```r
# The intercept is the estimated average time for Gwen, which has to be a
# positive number. The other coefficients are differences between each
# racer and Gwen. Since Gwen has the fastest car, these differences should
# be positive.
```

4. Show that the estimated expected time for each racer is equal to the sample averages that you computed in a previous question.

```
x1 <- tapply( derby$time, derby$racer, mean )
x2 <- rep(NA, length(x1))
names(x2) <- names(x1)
x2["Gwen"] <- m1$coefficients["(Intercept)"]
for(j in 2:length(x2)){
    nm <- names(x2)[j]
    x2[ names(x2)[j] ] <- m1$coefficients["(Intercept)"] + m1$coefficients[ paste0("racer",nm) ]
}
data.frame( x1, x2, diff = x2 - x1 )
```

```
##            x1       x2         diff
## Gwen 3.134767 3.134767 8.881784e-16
## Anne 3.166817 3.166817 8.881784e-16
## Bess 3.255850 3.255850 8.881784e-16
## Cher 3.167500 3.167500 8.881784e-16
## Ella 3.195283 3.195283 8.881784e-16
## Ibby 3.336733 3.336733 4.440892e-16
## Jane 3.154967 3.154967 8.881784e-16
## Kara 3.139933 3.139933 4.440892e-16
## Lily 3.148667 3.148667 8.881784e-16
## Mary 3.164983 3.164983 8.881784e-16
## Nina 3.143950 3.143950 4.440892e-16
## Suzy 3.138317 3.138317 2.220446e-15
```

5. Now fit a two-factor model to the times, with racer and lane as the two factors. Keep the fastest racer as the reference level and print the summary. How do you interpret the intercept in this model? Note that the intercept is a slightly different number and has a slightly different interpretation than that in the previous model.

```
derby$lane <- as.factor(derby$lane)
m2 <- lm( time ~ racer + lane, data = derby )
summary(m2)
```

```
##
## Call:
## lm(formula = time ~ racer + lane, data = derby)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.024386 -0.008582 -0.001653  0.008185  0.032647
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.136619   0.007158 438.188  < 2e-16 ***
## racerAnne    0.032050   0.008505   3.768 0.000403 ***
## racerBess    0.121083   0.008505  14.236  < 2e-16 ***
## racerCher    0.032733   0.008505   3.849 0.000312 ***
## racerElla    0.060517   0.008505   7.115 2.44e-09 ***
## racerIbby    0.201967   0.008505  23.746  < 2e-16 ***
## racerJane    0.020200   0.008505   2.375 0.021062 *
## racerKara    0.005167   0.008505   0.607 0.546036
```

```
## racerLily    0.013900   0.008505   1.634 0.107907
## racerMary    0.030217   0.008505   3.553 0.000791 ***
## racerNina    0.009183   0.008505   1.080 0.284972
## racerSuzy    0.003550   0.008505   0.417 0.678015
## lane2        0.003433   0.006014   0.571 0.570403
## lane3       -0.007250   0.006014  -1.206 0.233168
## lane4       -0.009050   0.006014  -1.505 0.138095
## lane5       -0.005967   0.006014  -0.992 0.325484
## lane6        0.007717   0.006014   1.283 0.204836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01473 on 55 degrees of freedom
## Multiple R-squared:  0.9524, Adjusted R-squared:  0.9386
## F-statistic: 68.83 on 16 and 55 DF,  p-value: < 2.2e-16
```

```
# The intercept is the estimated expected time for Gwen in lane 1, since
# Gwen and lane 1 are the reference levels for the two factors.
```

6. Show that the coefficients for the racers in the two-factor model are exactly the same as they were in the one-factor model. This means that, after accounting for lane, the relative differences among the racers is estimated to be the same. This is because the structure of the pinewood derby is set up so that each racer appears in each lane exactly once. If you take an experimental design course, you will learn that this is an example of a balanced experiment with respect to racer and lane. In a balanced experiment, the estimated effects in a two-factor model can be calculated by taking differences of simple averages.

```
range( m1$coefficients[2:12] - m2$coefficients[2:12] )
```

```
## [1] -4.336809e-19  0.000000e+00
```

7. Use the `anova` function to perform an F-test for whether `lane` significantly improves the model and interpret the results. The easiest way to do this is to put both the reduced model and the full model into the `anova` function as arguments.

```
anova(m1,m2)
```

```
## Analysis of Variance Table
##
## Model 1: time ~ racer
## Model 2: time ~ racer + lane
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     60 0.014585
## 2     55 0.011936  5 0.0026496 2.4419 0.04541 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# The p-value is 0.045, so we have moderate evidence that there are
# differences among the lanes.
```

8. Now fit a three-factor additive model with racer, lane, and heat as the three factors and show that the coefficients for the racers change. Why do you think this happens?

```
derby$heat <- as.factor(derby$heat)
m3 <- lm( time ~ racer + lane + heat, data = derby )
summary(m3)
```

```
##
```

```
## Call:
## lm(formula = time ~ racer + lane + heat, data = derby)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.025497 -0.007769 -0.001361  0.007031  0.034510
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.144164   0.010268 306.224  < 2e-16 ***
## racerAnne    0.025924   0.009365   2.768  0.00822 **
## racerBess    0.119479   0.009222  12.956  < 2e-16 ***
## racerCher    0.031265   0.009217   3.392  0.00148 **
## racerElla    0.055060   0.009365   5.879 5.08e-07 ***
## racerIbby    0.196467   0.009365  20.978  < 2e-16 ***
## racerJane    0.015953   0.009226   1.729  0.09080 .
## racerKara    0.002116   0.009226   0.229  0.81966
## racerLily    0.010299   0.009226   1.116  0.27039
## racerMary    0.028093   0.009226   3.045  0.00392 **
## racerNina    0.007866   0.009217   0.853  0.39810
## racerSuzy    0.002095   0.009222   0.227  0.82131
## lane2        0.003433   0.006237   0.550  0.58480
## lane3       -0.007250   0.006237  -1.162  0.25136
## lane4       -0.009050   0.006237  -1.451  0.15389
## lane5       -0.005967   0.006237  -0.957  0.34399
## lane6        0.007717   0.006237   1.237  0.22259
## heat2       -0.007378   0.009365  -0.788  0.43505
## heat3       -0.002372   0.009222  -0.257  0.79820
## heat4       -0.002634   0.009226  -0.285  0.77664
## heat5       -0.009288   0.009226  -1.007  0.31961
## heat6       -0.003088   0.009217  -0.335  0.73924
## heat7       -0.015886   0.009365  -1.696  0.09689 .
## heat8        0.005972   0.009217   0.648  0.52042
## heat9       -0.006872   0.009226  -0.745  0.46035
## heat10      -0.001894   0.009226  -0.205  0.83829
## heat11      -0.003951   0.009222  -0.428  0.67042
## heat12      -0.007198   0.009365  -0.769  0.44626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01528 on 44 degrees of freedom
## Multiple R-squared:  0.9591, Adjusted R-squared:  0.934
## F-statistic: 38.19 on 27 and 44 DF,  p-value: < 2.2e-16
```

```
range( m1$coefficients[2:12] - m3$coefficients[2:12] )
```

```
## [1] 0.001317826 0.006125531
```

```
# The experiment is balanced with respect to racer and lane, but it's not
# balanced with respect to racer and heat, since there are 12 racers
# but only 6 compete in each heat. If one of the heats runs particularly slow
# (due to wind or some other cause), then the 6 racers in that heat will tend to
# have slower average times than the 6 who didn't race in that heat. The model
# with racer, lane, and heat attempts to account for this possibility. Luckily,
# slow (or fast) heats don't appear to be an issue, since Gwen still comes out
```

```
# on top after accounting for heat, which you can see by noting that all of
# the racer coefficients are still positive, though slightly different than before.
```