SDS 439 - Homework 04

Due Mar 17, 1:00 pm

Pinewood Derby

This homework concerns analysis of times from the finals of a Pinewood Derby contest. Pinewood Derby is a competition in which people, usually children with the help of their parents, build cars from a kit that consists of a block of wood and wheels. The races take place on an inclined track in a series of heats that pit 6 cars at a time against each other.

The data are real, but the names of the children have been replaced with fake names. You can find it in `datasets/pinewood_derby.csv`

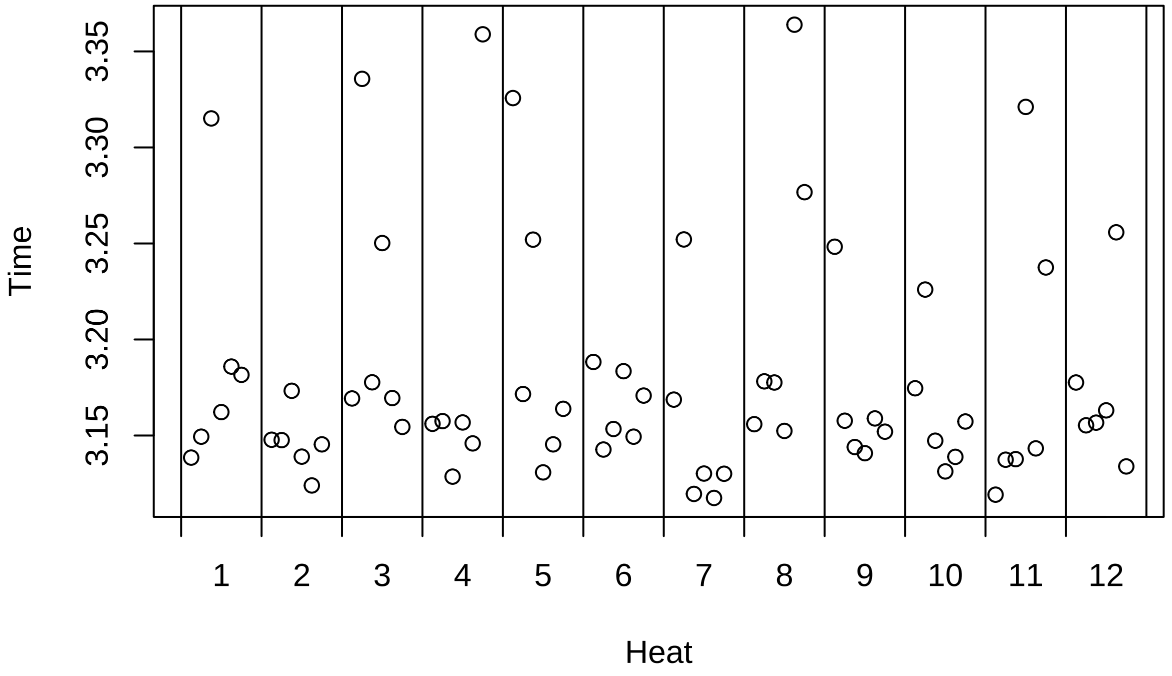
The final was contested in order to determine which racer had the fastest car. We are going to analyze the data with this goal in mind, using the times as the response, and racer as a factor covariate.

1. The first thing you should do in a data analysis is to try to find a way to look at the data. Read in the data and make one or two informative plots. Remember to label your axes, use legends where appropriate, use informative plotting symbols. Try to get multiple aspects of the data into your plot. Also, make sure your plot looks nice when knitted to a pdf.

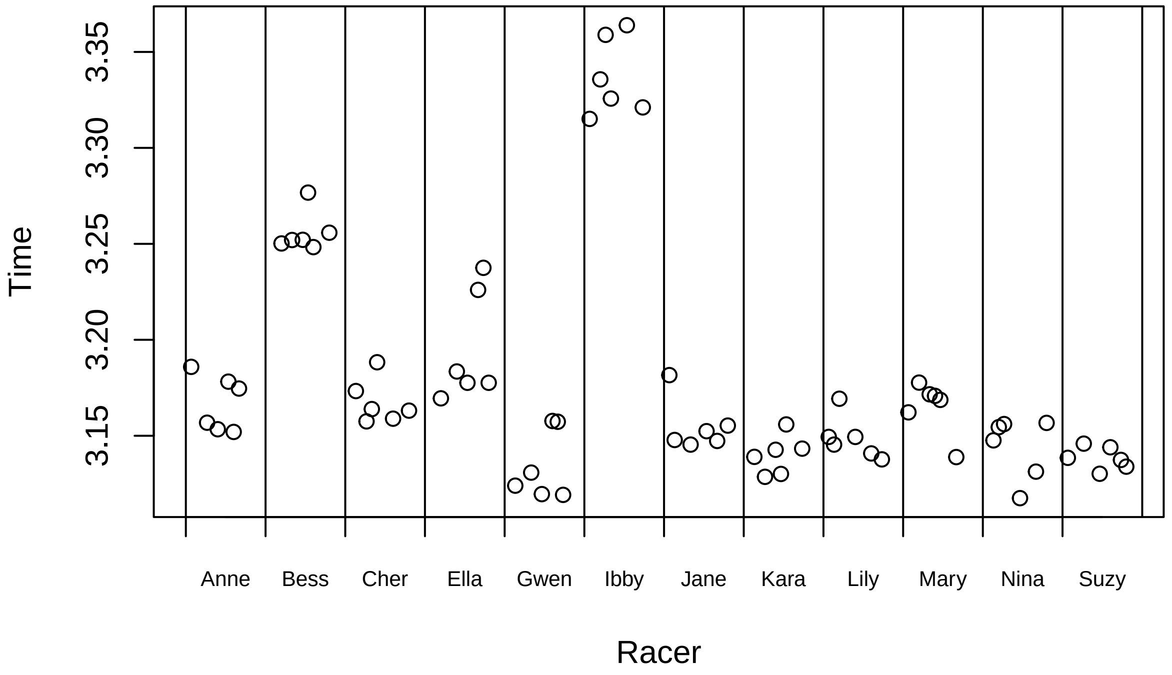
```r
derby <- read.csv("../datasets/pinewood_derby.csv")
head(derby)
```

```
##   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
```

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



2. To get started, create the design matrix X for the one-factor model that has racer as the factor covariate. Create the full design matrix without dropping any columns. Print out the first twelve rows.

```
X <- cbind( rep(1,nrow(derby)), model.matrix( ~ racer -1, data = derby ) )
X[1:12,]
```

```
##      racerAnne racerBess racerCher racerElla racerGwen racerIbby racerJane
## 1  1         0         0         0         0         0         0         0
## 2  1         0         0         0         0         0         0         0
## 3  1         0         0         0         0         0         0         0
## 4  1         0         0         0         0         0         0         1
## 5  1         1         0         0         0         0         0         0
## 6  1         0         0         0         0         0         1         0
## 7  1         0         0         0         0         1         0         0
## 8  1         0         0         0         0         0         0         0
## 9  1         0         0         0         0         0         0         0
## 10 1         0         0         0         0         0         0         0
## 11 1         0         0         0         0         0         0         1
## 12 1         0         0         1         0         0         0         0
##    racerKara racerLily racerMary racerNina racerSuzy
## 1          0         0         0         0         1
## 2          0         1         0         0         0
## 3          0         0         1         0         0
## 4          0         0         0         0         0
## 5          0         0         0         0         0
## 6          0         0         0         0         0
## 7          0         0         0         0         0
## 8          1         0         0         0         0
## 9          0         1         0         0         0
## 10         0         0         0         1         0
## 11         0         0         0         0         0
## 12         0         0         0         0         0
```

3. Calculate $X^T X$ and demonstrate that its inverse does not exist. This is actually a little tricky because the markdown will not knit if the code has an error. You can circumvent this by putting the problematic code inside of the `try` function, or you can use your matrix algebra knowledge to find a way to show that the inverse does not exist (e.g. look at eigenvalues).

```
XtX <- t(X) %*% X
try( solve(XtX) )
```

```
## Error in solve.default(XtX) :
##   system is computationally singular: reciprocal condition number = 1.89782e-18
```

```
ee <- eigen(XtX)
min(ee$values)
```

```
## [1] 7.105427e-14
```

4. Fit a model to the race times, using racer as the factor covariate. Print out the summary of the model fit.

```
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
```

```
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)  3.1668167  0.0063651 497.526  < 2e-16 ***
## racerBess    0.0890333  0.0090017   9.891 3.23e-14 ***
## racerCher    0.0006833  0.0090017   0.076 0.939742    
## racerElla    0.0284667  0.0090017   3.162 0.002455 ** 
## racerGwen   -0.0320500  0.0090017  -3.560 0.000732 ***
## racerIbby    0.1699167  0.0090017  18.876  < 2e-16 ***
## racerJane   -0.0118500  0.0090017  -1.316 0.193038    
## racerKara   -0.0268833  0.0090017  -2.986 0.004082 ** 
## racerLily   -0.0181500  0.0090017  -2.016 0.048252 *  
## racerMary   -0.0018333  0.0090017  -0.204 0.839304    
## racerNina   -0.0228667  0.0090017  -2.540 0.013683 *  
## racerSuzy   -0.0285000  0.0090017  -3.166 0.002428 ** 
## ---
## 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
```

5. Which column did R drop and how do you know?

```
# R removed the column for "Anne". You can tell by looking at which name
# is missing from the coefficients.
m1$coefficients
```

```
##   (Intercept)     racerBess     racerCher     racerElla     racerGwen 
##  3.1668166667  0.0890333333  0.0006833333  0.0284666667 -0.0320500000 
##     racerIbby     racerJane     racerKara     racerLily     racerMary 
##  0.1699166667 -0.0118500000 -0.0268833333 -0.0181500000 -0.0018333333 
##     racerNina     racerSuzy 
## -0.0228666667 -0.0285000000
```

```
sort( unique( derby$racer ) )
```

```
##  [1] "Anne" "Bess" "Cher" "Ella" "Gwen" "Ibby" "Jane" "Kara" "Lily" "Mary"
## [11] "Nina" "Suzy"
```

6. Using words, supported by code if necessary, answer the following questions about the results:

(a) How do you interpret the intercept?

(b) How do you interpret the racerBess coefficient and what can you conclude from the t-test for the racerBess coefficient?

(c) What is the size of the error after accounting for racer, and how does it compare in magnitude to the sizes of the racer effects?

(d) Which racer is estimated to have the fastest car?

(e) Which racer is estimated to have the second fastest car?

(f) Do we have enough evidence to say that the fastest estimated car is faster than the second-fastest? Do a t-test to answer this question.

```
# Intercept is the estimated expected time for Anne, since R dropped the column for Anne

# racerBess is the estimated expected time for Bess minus the estimated expected time
# for Anne (the reference racer). Since the coefficient is positive, and the
# p-value for this t-test is very small, we can conclude that Bess has a slower
# car than Anne.

# sigmahat = 0.01559. This is roughly the size of some of the differences in
# estimated expected time.

# Gwen is has the most negative coefficient, so it has the fastest estimated
# expected time

# The second most negative coefficient belongs to Suzy

# We can get the estimated difference between Gwen and Suzy, using the following
# linear combination
d <- rep(0, length(m1$coefficients) )
d[ c(5,12) ] <- c(1,-1)
# make sure they line up:
cbind( m1$coefficients, d )
```

```
##                            d
## (Intercept)  3.1668166667  0
## racerBess    0.0890333333  0
## racerCher    0.0006833333  0
## racerElla    0.0284666667  0
## racerGwen   -0.0320500000  1
## racerIbby    0.1699166667  0
## racerJane   -0.0118500000  0
## racerKara   -0.0268833333  0
## racerLily   -0.0181500000  0
## racerMary   -0.0018333333  0
## racerNina   -0.0228666667  0
## racerSuzy   -0.0285000000 -1
```

```
# calculate the difference
est <- t(d) %*% m1$coefficients
est
```

```
##          [,1]
## [1,] -0.00355
```

```
# calculate the standard error:
se <- sqrt( t(d) %*% vcov(m1) %*% d )
se
```

```
##             [,1]
## [1,] 0.009001657
```

```r
# calculate the t-statistic and p-value
tstat <- est/se
tstat
```

```
##            [,1]
## [1,] -0.3943718
```

```r
pval <- 2*pt( -abs(tstat), m1$df.residual )
pval
```

```
##           [,1]
## [1,] 0.6947048
```

```r
# the pvalue is 0.69, which means we don't have enough evidence to reject the
# hypothesis that Gwen and Suzy has cars with the same speed.
```

7. What else would you want to do in order to investigate which racers have the fastest cars? (You don't need to do it, just give some ideas)

```r
# there are various things to consider, including accounting for the speed of
# the lanes, accounting for how fast or slow different heats were,
# considering the effects of the outliers. Since we can't detect a difference
# between the fastest cars, we might want to run more heats.
```