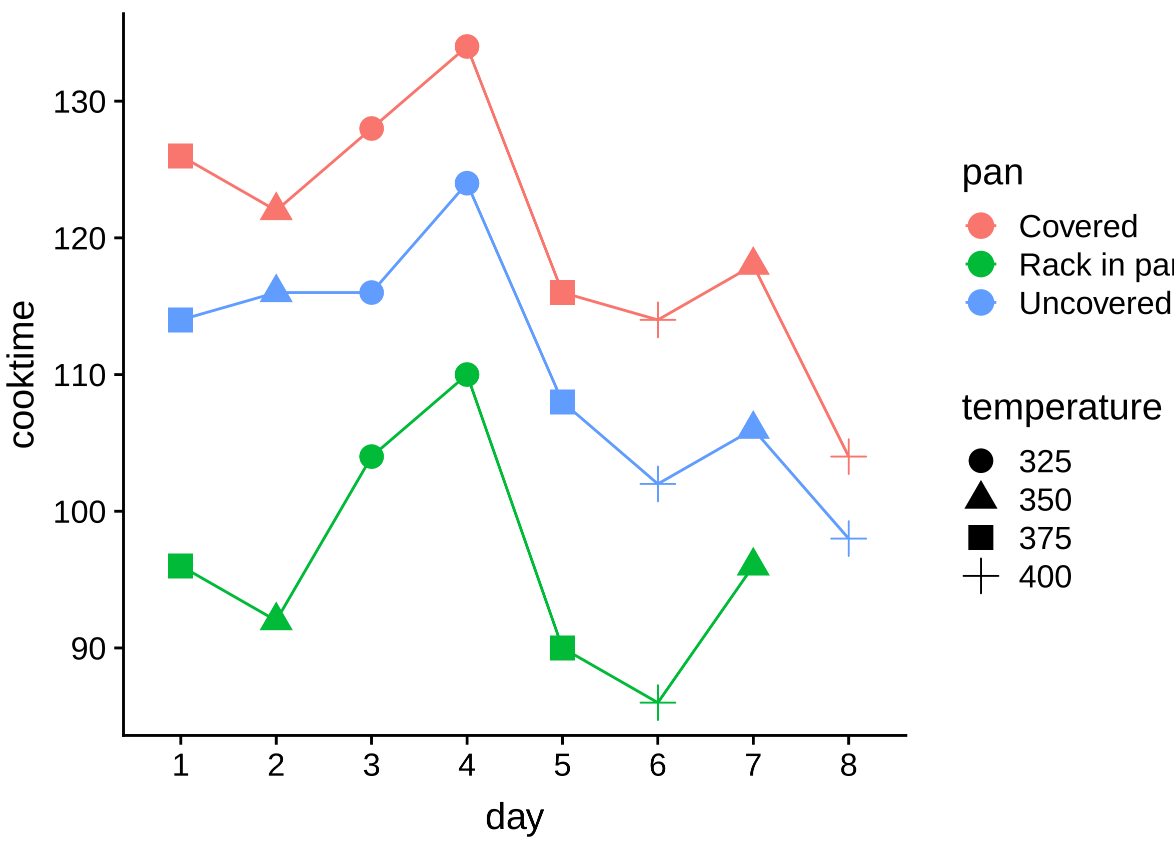
STAT850 HW10

Stewart Kerr

April 22, 2019

Problem 1

```
#Problem 1
potroast_num <- potroast <- read.csv("~/2019spring/STAT850/hw10/potroast.csv")
potroast$day = factor(potroast$day)
potroast$temperature = factor(potroast$temperature)

#Make a suitable plot to begin analysis
ggplot(potroast, aes(x = day, y = cooktime, color = pan)) + aes(group = pan, y = cooktime) + geom_point
```



Based on this plot, it appears that pan has the largest effect on cooking time. I would also say that temperature has an effect - the higher temperatures have a lower cooktime (within each pan treatment). There does not seem to be much of a strong day effect, but I still think it was good to use each pan on each day - more replications of pan is useful because that appears to have the largest effect on cooktime.

Problem 2

a. The experimental design for this study is a split plot. The whole plot factor is temperature and the subplot factor is pan, there is also a blocking factor for day. For analysis, I will examine the fixed effects of temperature and pan (as well as their interaction).

```r
#Problem 2a
roast_fit <- lmer(cooktime ~ temperature*pan + (1|day), data = potroast)
anova(roast_fit)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                  Sum Sq Mean Sq NumDF  DenDF  F value    Pr(>F)
## temperature      223.51    74.5     3 3.9520   8.0084   0.03716 *
## pan             2300.23  1150.1     2 7.0793 123.6275 3.122e-06 ***
## temperature:pan   16.78     2.8     6 7.0511   0.3006   0.91797
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

An analysis of this data consistent with a standard split plot design was conducted above. We see that both F-tests for temperature and pan have small p-values, with pan having an extremely small p-value, indicating that both factors are likely influential on cooking time. Their interaction, however, has a very high p-value and does not appear to significantly affect cooking time. \

If this design were balanced, we would expect DF(WPError) $= t \times (b-1) = 4 \times (2-1) = 4$ denominator degrees of freedom for temperature and DF(SPError) $= t \times (b-1) \times (p-1) = 4 \times (2-1) \times (3-1) = 8$ denominator degrees of freedom for pan and the temperature/pan interaction term. In both cases, we have less approximate degrees of freedom with the unbalanced case due to the spoiled meat on day 8.

b.

```r
#Problem 2b
roast_fit <- lmer(cooktime ~ temperature*pan + (1|day), data = potroast)
lsmeans(roast_fit, pairwise ~ temperature)
```

```
## $lsmeans
##  temperature lsmean   SE   df lower.CL upper.CL
##  325          119.3 3.10 3.86    110.6      128
##  350          108.3 3.10 3.86     99.6      117
##  375          108.3 3.10 3.86     99.6      117
##  400           97.4 3.23 4.40     88.8      106
##
## Results are averaged over the levels of: pan
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##  contrast  estimate   SE   df t.ratio p.value
##  325 - 350     11.0 4.39 3.86 2.506   0.2021
##  325 - 375     11.0 4.39 3.86 2.506   0.2021
##  325 - 400     21.9 4.48 4.13 4.890   0.0255
##  350 - 375      0.0 4.39 3.86 0.000   1.0000
##  350 - 400     10.9 4.48 4.13 2.434   0.2081
##  375 - 400     10.9 4.48 4.13 2.434   0.2081
##
## Results are averaged over the levels of: pan
## P value adjustment: tukey method for comparing a family of 4 estimates
```

This t-test is approximate because we had to use Kenward-Roger approximation to get our degrees of freedom. We had to do this because we have unequal sample size for the 400 degree cooking temperature (from the spoiled pot-roast on the final day). Looking at our contrasts, we see that the cooking temperatures for 350, 375, and 400 do not differ significantly (at $\alpha = 0.05$) but 325 does differ significantly from 400. Also, it's interesting to see that temperatures 350 and 375 have the exact same average cooking time.

Problem 3

a. We only cooked at one temperature each day, so it wouldn't make sense for each day to have its own slope for temperature.

b. First, we'll fit the model where each day has its own intercept.

```
#Problem 3b
# Fit with each day having its own slope and intercept
potroast_num$day = factor(potroast_num$day)
roast_fit <- lmer(cooktime ~ temperature*pan + (1|day), data = potroast_num)
```

Now, we'll test for the interaction effect using a LRT with a chi-square null distribution.

```
#Problem 3b
#Remove the interaction term
roast_fit_null <- update(roast_fit, . ~ . -temperature:pan, REML = FALSE)

#Perform the LRT
anova(roast_fit_null, roast_fit)
```

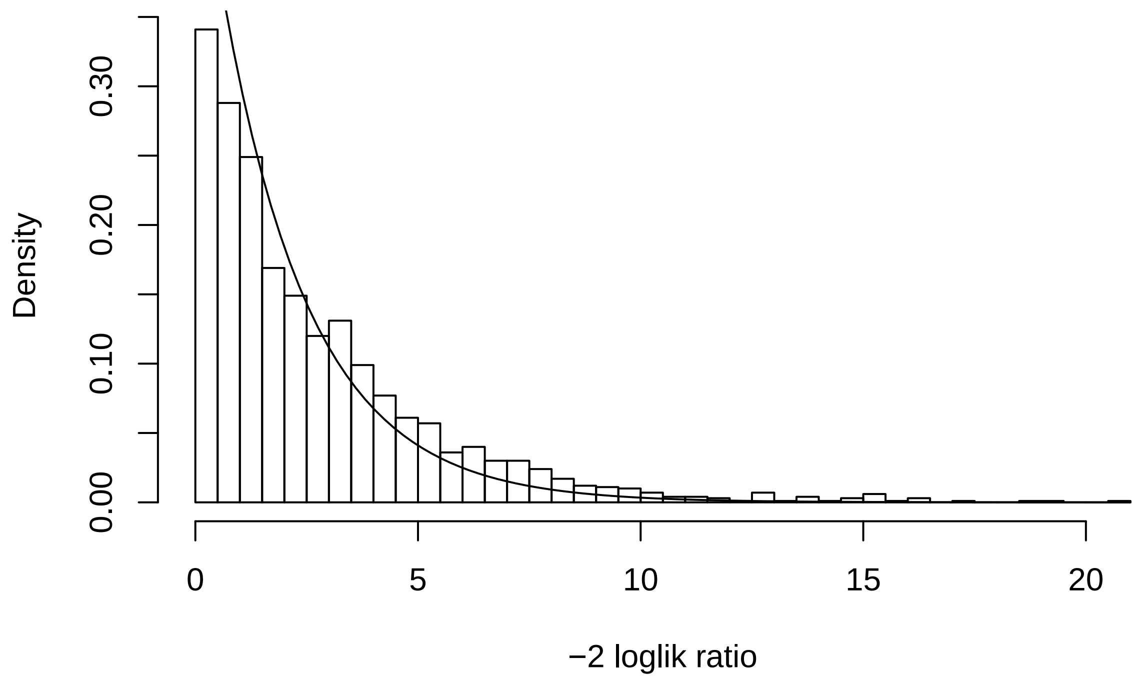
```
## refitting model(s) with ML (instead of REML)

## Data: potroast_num
## Models:
## roast_fit_null: cooktime ~ temperature + pan + (1 | day)
## roast_fit: cooktime ~ temperature * pan + (1 | day)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## roast_fit_null  6 133.21 140.02 -60.602   121.20
## roast_fit       8 135.38 144.47 -59.692   119.38 1.8209      2     0.4023
```

Then LRT via parametric bootstrap:

```
#Problem 3b
#Define likelihood ratio function
lr = function(m){
  #Add interaction back
  m.withint = with(m@frame, lmer(cooktime ~ temperature*pan + (1|day), REML=FALSE))
  #Calculate LRT statistic
  x2 = as.numeric(2*(logLik(m.withint) - logLik(m)))

  return(c(lr = x2))
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag')) {
  #Perform the bootstrap
  bag = bootMer(roast_fit_null, lr, nsim = 2000)
}
bag_df <- as.data.frame(bag)

#Now find the p-value
bag_df$lr[bag_df$lr<0] = 0.0
pval = mean(bag_df$lr >= bag$t0["lr"])
paste("The p-value for the parametric bootstrap test is: ",pval)
```

```
## [1] "The p-value for the parametric bootstrap test is:  0.505"
```

```r
#Plot the null distribution
hist(bag_df$lr, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=2), add = T, n = 100)
```



Lastly, we will perform an approximate f-test:

```r
#Problem 3b
anova(roast_fit)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                  Sum Sq Mean Sq NumDF   DenDF F value   Pr(>F)
## temperature     129.618 129.618     1  6.0299 20.0888 0.004132 **
## pan               7.018   3.509     2 11.0578  0.5438 0.595279
## temperature:pan   9.461   4.730     2 11.0696  0.7332 0.502331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In summary, we found a p-value of 0.4023 for the LRT based on the chi-square distribution, a p-value of 0.521 for the LRT based on a parametric bootstrap distribution, and a p-value of 0.502331 for the approximate F-test. The LRT based on the chi-square distribution uses infinite degrees of freedom, so it tends to underestimate the p-value (and is not accurate to report). The p-value for both the approximate F-test and the parametric bootstrap are fairly close, however, I will prefer to report the p-value for the parametric bootstrap because our design is not balanced (and thus the F-test is not exact). Thus, we have a p-value of 0.521 - indicating that interaction between temperature and pan does not seem to have a large influence on cooking time.

c. Now, I will test the effect of day using the same model as above. First, the LRT based on the chi-square test statistic:

```r
#Problem 3c

#Perform the LRT
ranova(roast_fit)
```

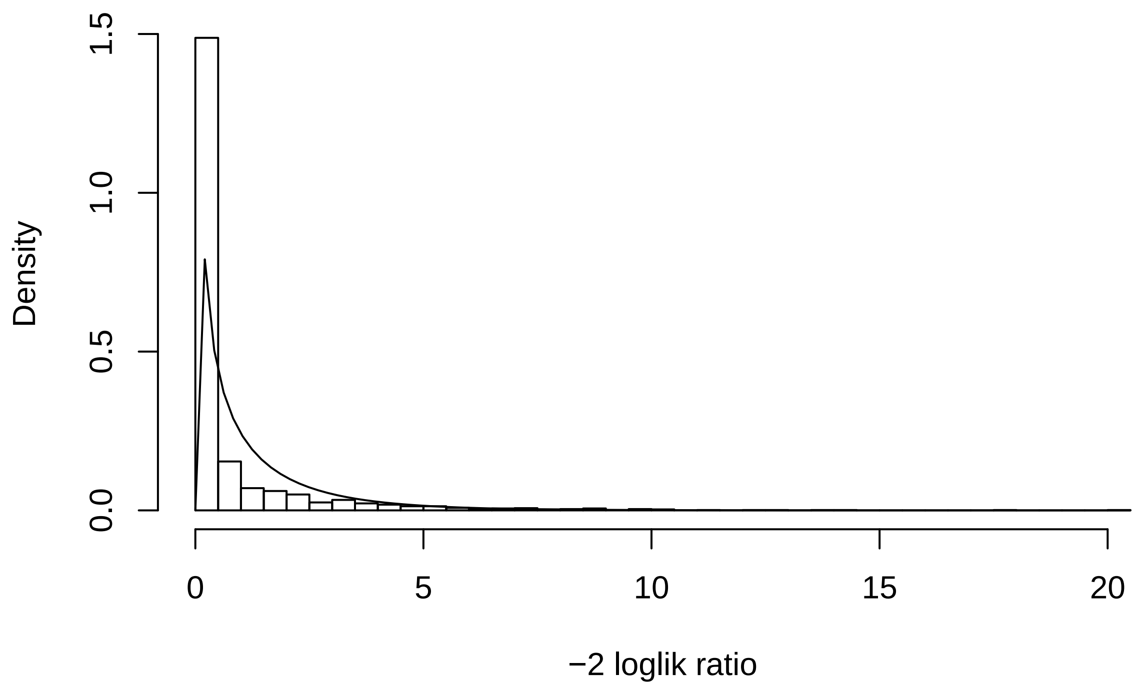
```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## cooktime ~ temperature + pan + (1 | day) + temperature:pan
##             npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>         8 -62.758 141.52
## (1 | day)      7 -67.652 149.30 9.7877  1   0.001757 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, the parametric bootstrap test:

```r
#Problem 3c
#Define bootstrap function
oneX2 = function(x){
  #Define null model to generate data
  null_model = lm(cooktime ~ temperature*pan, data = potroast_num)
  y = simulate(null_model)$sim_1

  #Make alternative model from data simulated by null
  alt_model = lmer(y ~ temperature*pan + (1|day), data = potroast_num, REML = FALSE)
  null_model_fit = lm(y ~ temperature*pan, data = potroast_num)

  #Return the LRT statistic
  x2 = as.numeric(2*(logLik(alt_model) - logLik(null_model_fit)))
  x2 = ranova(alt_model)$LRT[2]

  return(x2)
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag_c')) {
  #Perform the bootstrap
  bag_c = unlist(parallel::mclapply(1:2000,oneX2))
}
bag_df <- as.data.frame(bag_c)

#Now find the p-value
bag_df$bag_c[bag_df$bag_c<0] = 0.0
pval = mean(bag_df$bag_c >= 9.7877)
paste("The p-value for the parametric bootstrap test is: ",pval)
```

```
## [1] "The p-value for the parametric bootstrap test is:  0.0065"
```

```r
#Plot the null distribution
hist(bag_df$bag_c, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=1), add = T, n = 100)
```



For these tests, the hypothesis are $H_0 : \sigma^2_{Day} = 0$ vs. $H_1 : \sigma^2_{Day} \neq 0$. Both of our tests have p-values that are very small, indicating that our data suggests there is a significant effect on day. For the LRT based on the chi-square test statistic, we got a p-value of 0.001757. We know that the LRT for random effect usually gives a p-value that is too *high*. The parametric bootstrap yielded a p-value of 0.0065. However, the bootstrap is not particularly trustworthy because we have a spike at zero (which is a bound) - that might be way the p-value is larger than expected. In this case, I would choose to report the p-value of the LRT test with the chi-square test statistic.

d.

```r
#Problem 3d
#No interaction
roast_fit <- lmer(cooktime ~ temperature + pan + (1|day), data = potroast_num)
summary(roast_fit)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: cooktime ~ temperature + pan + (1 | day)
##    Data: potroast_num
##
## REML criterion at convergence: 118.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6423 -0.3458 -0.0465  0.5718  1.4659
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
```

```
##  day      (Intercept) 18.60    4.313
##  Residual             6.34    2.518
## Number of obs: 23, groups:  day, 8
##
## Fixed effects:
##                  Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)     215.50810   21.07336   6.05630  10.227 4.80e-05 ***
## temperature      -0.26278    0.05793   6.04277  -4.536  0.00388 **
## panRack in pan  -25.92381    1.32236  13.06036 -19.604 4.53e-11 ***
## panUncovered     -9.75000    1.25897  12.97062  -7.744 3.23e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tmprtr pnRcip
## temperature -0.996
## panRackinpn -0.062  0.034
## panUncoverd -0.030  0.000  0.476
```

For the confidence interval, we're interested in estimating the effect of cooking time when going from a covered steak to an uncovered steak. In this parameterization, the intercept is the covered pan, thus we're interested in making a confidence interval for the "panUncovered" fixed effect. From the summary table, you can see that our estimate for the panUncovered term is -9.75, the standard error is 1.322, and the approximate degrees of freedom is 13.06. The 95% confidence interval is thus:

$$CI = \hat{\beta} \pm t_{\alpha/2,\text{adf}} \times SE(\hat{\beta}) = -9.75 \pm t_{0.025,13.06} \times 1.322 = -9.75 \pm 2.86 = [-12.61, -6.89]$$

Note that on an exam, I would use 13 degrees of freedom instead of the approximated 13.06 degrees of freedom. Therefore, the 95% confidence interval based on our data for the change in cooking time going from a covered pan to an uncovered pan is [-12.61,-6.89] minutes - that is, the uncovered pan on average cooks between 6.89 and 12.61 minutes faster than the covered pan.

To predict the cooking time for a new roast cooked at 340 degrees using the rack in pan method we add the fixed effects of the intercept, temperature, and rack in pan. We do not need to add in random effects because they have mean of zero. Thus, the prediction for this new roast's cooking time is:

$$\text{Cook Time} = \beta_0 + 340\beta_1 + \beta_2 = 215.50810 + 340 \times -0.26278 + -25.92381 = 100.24 \text{ minutes}$$

Where β_0 is the intercept term, β_1 is the temperature slope, and β_2 is the rack in pan cooking method term. Thus, for this roast we would predict a cooking time of 100.24 minutes.

Problem 4

a.

```r
#Problem 4a
roast_fit_null <- lmer(cooktime ~ temperature + pan + (1|day), data = potroast_num, REML = FALSE)
roast_fit_alt <- lmer(cooktime ~ as.factor(temperature) + pan + (1|day), data = potroast_num, REML = FA

#LRT with ChiSq test statistic
anova(roast_fit_null, roast_fit_alt)
```

```
## Data: potroast_num
## Models:
## roast_fit_null: cooktime ~ temperature + pan + (1 | day)
## roast_fit_alt: cooktime ~ as.factor(temperature) + pan + (1 | day)
```

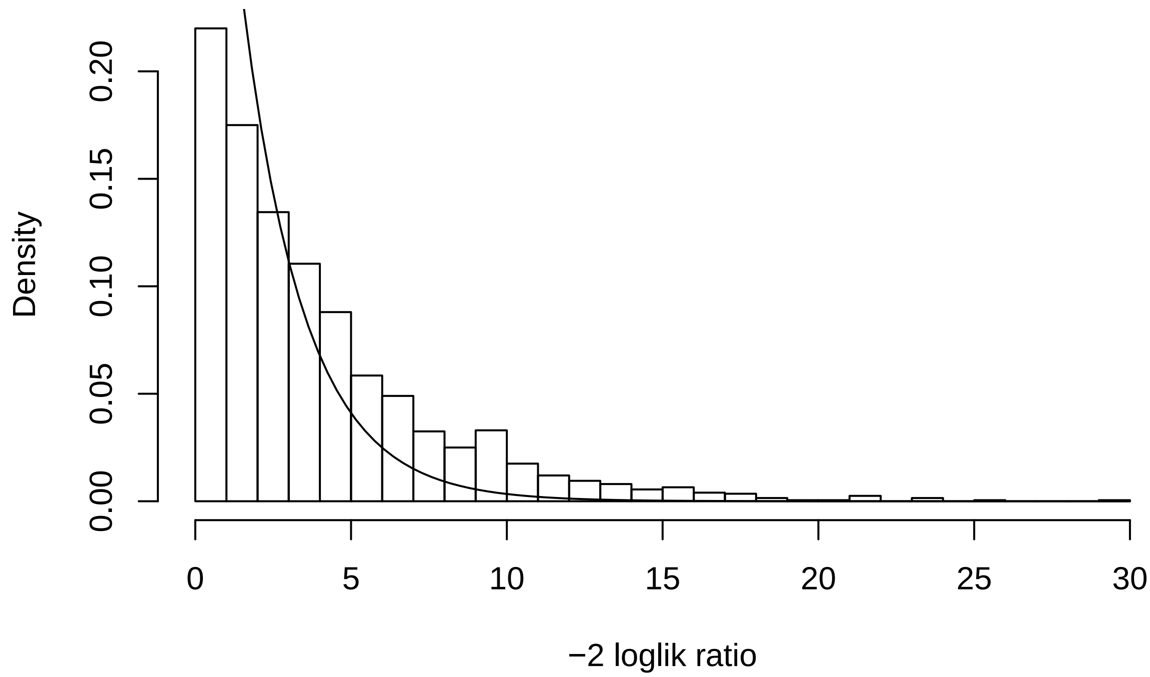
```
##                 Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## roast_fit_null  6 133.21 140.02 -60.602    121.2
## roast_fit_alt   8 133.30 142.38 -58.649    117.3 3.9059      2     0.1419
```

b.

```r
#Problem 4b
#Define likelihood ratio function
lr = function(m){
  #Change to categorical
  m.cat = with(m@frame, lmer(cooktime ~ as.factor(temperature) + pan + (1|day), REML=FALSE))
  #Calculate LRT statistic
  x2 = as.numeric(2*(logLik(m.cat) - logLik(m)))

  return(c(lr = x2))
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag_4b')) {
  #Perform the bootstrap
  bag_4b = bootMer(roast_fit_null, lr, nsim = 2000)
}
bag_df <- as.data.frame(bag_4b)

#Now find the p-value
bag_df$lr[bag_df$lr<0] = 0.0
pval = mean(bag_df$lr >= bag_4b$t0["lr"])
paste("The p-value for the parametric bootstrap test is: ",pval)
```

```
## [1] "The p-value for the parametric bootstrap test is:  0.3715"
```

```r
#Plot the null distribution
hist(bag_df$lr, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=2), add = T, n = 100)
```



c.

```
#Problem 4c
#Approximate F test
anova(roast_fit_null)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##              Sum Sq Mean Sq NumDF   DenDF F value    Pr(>F)
## temperature  150.53  150.53     1  8.0644  27.386 0.0007698 ***
## pan         2455.94 1227.97     2 15.0313 223.396 6.616e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(roast_fit_alt)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                         Sum Sq Mean Sq NumDF   DenDF F value    Pr(>F)
## as.factor(temperature)  270.31    90.1     3  7.9523  16.385  0.000912 ***
## pan                    2453.97  1227.0     2 15.0372 223.125 6.627e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For these tests, we have the following hypotheses: H_0 : Relationship between cooking time and temperature is linear vs. H_1 : Relationship between cooking time and temperature is not linear. To help determine this dependence, we performed three tests. A LRT with a chi-square test statistic was performed and a p-value of 0.1419 was found. The LRT for fixed effects is asymptotic and tends to give a p-value that is too small, thus I do not think this p-value is best to report. Next, we performed a parametric bootstrap test based on the null and alternative hypothesis. We found a p-value of 0.3715. This procedure performs a simulation under the null hypothesis and estimates the log likelihood under the null distribution - so I think this p-value is pretty good to report. The last test was an F-test performed on the model with temperature as a numeric feature. This F-test examines the hypothesis $H_0 : \beta_T = 0$ vs. $H_1 : \beta_T \neq 0$. The p-value is very small which indicates there is an effect of temperature. However, it's not really telling us whether it's more appropriate for temperature to be a categorical predictor instead of a numeric predictor, thus this p-value doesn't really answer our question. In fact, if you perform an F-test on the model with temperature as a categorical predictor, you also get an extremely low p-value.

Based on the discussion above, I think the p-value that's most appropriate to report is p = 0.3715 from the parametric bootstrap test. Our p-value is relatively large, thus our data is good evidence for the null hypothesis, suggesting that temperature does likely has a linear effect.

Code

```
## ----include = FALSE------------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(knitr)
library(car)
library(ggplot2)
library(MASS)
library(lme4)
library(lmerTest)
library(tidyr)
library(cowplot)
library(multcomp)
library(lsmeans)

set.seed(1104)              # make random results reproducible

this_file <- "kerr_stat850_hw10.Rmd"  # used to automatically generate code appendix

## ------------------------------------------------------------------------
#Problem 1
potroast_num <- potroast <- read.csv("~/2019spring/STAT850/hw10/potroast.csv")
potroast$day = factor(potroast$day)
potroast$temperature = factor(potroast$temperature)

#Make a suitable plot to begin analysis
ggplot(potroast, aes(x = day, y = cooktime, color = pan)) + aes(group = pan, y = cooktime) + geom_point

## ------------------------------------------------------------------------
#Problem 2a
roast_fit <- lmer(cooktime ~ temperature*pan + (1|day), data = potroast)
anova(roast_fit)

## ---- warning=FALSE, error=FALSE, message=FALSE--------------------------
#Problem 2b
roast_fit <- lmer(cooktime ~ temperature*pan + (1|day), data = potroast)
lsmeans(roast_fit, pairwise ~ temperature)

## ------------------------------------------------------------------------
#Problem 3b
```

```r
# Fit with each day having its own slope and intercept
potroast_num$day = factor(potroast_num$day)
roast_fit <- lmer(cooktime ~ temperature*pan + (1|day), data = potroast_num)

## ---------------------------------------------------------------------------
#Problem 3b
#Remove the interaction term
roast_fit_null <- update(roast_fit, . ~ . -temperature:pan, REML = FALSE)

#Perform the LRT
anova(roast_fit_null, roast_fit)

## ---- warning=FALSE, error=FALSE, message=FALSE--------------------------
#Problem 3b
#Define likelihood ratio function
lr = function(m){
  #Add interaction back
  m.withint = with(m@frame, lmer(cooktime ~ temperature*pan + (1|day), REML=FALSE))
  #Calculate LRT statistic
  x2 = as.numeric(2*(logLik(m.withint) - logLik(m)))

  return(c(lr = x2))
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag')) {
  #Perform the bootstrap
  bag = bootMer(roast_fit_null, lr, nsim = 2000)
}
bag_df <- as.data.frame(bag)

#Now find the p-value
bag_df$lr[bag_df$lr<0] = 0.0
pval = mean(bag_df$lr >= bag$t0["lr"])
paste("The p-value for the parametric bootstrap test is: ",pval)

#Plot the null distribution
hist(bag_df$lr, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=2), add = T, n = 100)

## ---------------------------------------------------------------------------
#Problem 3b
anova(roast_fit)

## ---------------------------------------------------------------------------
#Problem 3c

#Perform the LRT
ranova(roast_fit)

## ---- warning=FALSE, error=FALSE, message=FALSE--------------------------
#Problem 3c
```

```
#Define bootstrap function
oneX2 = function(x){
  #Define null model to generate data
  null_model = lm(cooktime ~ temperature*pan, data = potroast_num)
  y = simulate(null_model)$sim_1

  #Make alternative model from data simulated by null
  alt_model = lmer(y ~ temperature*pan + (1|day), data = potroast_num, REML = FALSE)
  null_model_fit = lm(y ~ temperature*pan, data = potroast_num)

  #Return the LRT statistic
  x2 = as.numeric(2*(logLik(alt_model) - logLik(null_model_fit)))
  x2 = ranova(alt_model)$LRT[2]

  return(x2)
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag_c')) {
  #Perform the bootstrap
  bag_c = unlist(parallel::mclapply(1:2000,oneX2))
}
bag_df <- as.data.frame(bag_c)

#Now find the p-value
bag_df$bag_c[bag_df$bag_c<0] = 0.0
pval = mean(bag_df$bag_c >= 9.7877)
paste("The p-value for the parametric bootstrap test is: ",pval)

#Plot the null distribution
hist(bag_df$bag_c, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=1), add = T, n = 100)

## ----------------------------------------------------------------------------
#Problem 3d
#No interaction
roast_fit <- lmer(cooktime ~ temperature + pan + (1|day), data = potroast_num)
summary(roast_fit)

## ----------------------------------------------------------------------------
#Problem 4a
roast_fit_null <- lmer(cooktime ~ temperature + pan + (1|day), data = potroast_num, REML = FALSE)
roast_fit_alt <- lmer(cooktime ~ as.factor(temperature) + pan + (1|day), data = potroast_num, REML = FA

#LRT with ChiSq test statistic
anova(roast_fit_null, roast_fit_alt)

## ---- warning=FALSE, error=FALSE, message=FALSE--------------------------
#Problem 4b
#Define likelihood ratio function
lr = function(m){
  #Change to categorical
```

```
  m.cat = with(m@frame, lmer(cooktime ~ as.factor(temperature) + pan + (1|day), REML=FALSE))
  #Calculate LRT statistic
  x2 = as.numeric(2*(logLik(m.cat) - logLik(m)))

  return(c(lr = x2))
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag_4b')) {
  #Perform the bootstrap
  bag_4b = bootMer(roast_fit_null, lr, nsim = 2000)
}
bag_df <- as.data.frame(bag_4b)

#Now find the p-value
bag_df$lr[bag_df$lr<0] = 0.0
pval = mean(bag_df$lr >= bag_4b$t0["lr"])
paste("The p-value for the parametric bootstrap test is: ",pval)

#Plot the null distribution
hist(bag_df$lr, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=2), add = T, n = 100)


## -----------------------------------------------------------------------
#Problem 4c
#Approximate F test
anova(roast_fit_null)
anova(roast_fit_alt)

## ----code = readLines(purl(this_file, documentation = 1)), echo = T, eval = F----
## # this R markdown chunk generates a code appendix
```