Residuals – actual score – predicted
SE – how much our sample estimates will vary due to chance

Lab Assignment #3
Student Name: KEY

Instructions: Complete the tasks outlined in this document. You should upload your completed assignment online as a .pdf, along with your R file.

Problem 1 : Cool Problems

1A. Load the dataset "2cool4school.csv" into R. These data are from Berkeley and Stanford students who answered questions about their love for coffee, and were then rated on how cool they were by an unbiased observer.

Use R to print a *random* set of rows from this dataset and the number of participants in the dataset.

```
> cool <- read.csv("D://Dropbox/STATS/Data/2cool4school.csv") # new data!
> some(cool)
     cool1 cool2r cool3 cool4 cool5r cool6 cool7r cool8 coolness   SCH coffee.love SEX AGE
66      NA     NA    NA    NA     NA    NA     NA    NA       NA Stanf           8   1  19
1867     4      2     4     4      2     4      2     4    4.000 Stanf           6   2  21
2419     2      3     2     2      4     2      4     2    2.125   Cal           9   2  19
2537     5      1     5     5      2     5      2     4    4.625   Cal           8   1  21
2993     5      2     5     5      2     5      2     5    4.625 Stanf           9   1  18
3302    NA     NA    NA    NA     NA    NA     NA    NA       NA Stanf           3   2  20
3829     4      2     4     4      3     4      4     5    3.750   Cal           3   2  20
3944     4      4     4     4      3     4      3     4    3.500   Cal          10   1  22
4552     2      4     3     3      4     2      4     3    2.375 Stanf           7   1  20
4789     1      5     2     2      5     4      5     1    1.625 Stanf           9   1  20
> nrow(cool)
[1] 5738
```

1B. The variable SEX is reported as a numerical variable. Create a new variable in the dataset called sexF that is a copy of SEX, but is saved as a factor variable with three levels (1 = female; 2 = male; 3 = other). The variable SCH is a categorical factor with two levels. Create a variable called school that is a copy of SCH, and change the names of the levels so Cal = Public and Stanf = Private. Print summaries of these two new variables to show that you did the work correctly.

```
> summary(cool$SEX)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.503   2.000   3.000
> cool$sexF <- cool$SEX # creating the copy
> cool$sexF <- as.factor(cool$sexF)
> levels(cool$sexF)
[1] "1" "2" "3"
> levels(cool$sexF) <- c('female', 'male', 'other')
> summary(cool$sexF) # whoop!
female   male  other
  2879   2830     29
```

```
> summary(cool$SCH)
  Cal Stanf
 2870  2868
> cool$schoolF <- cool$SCH
> levels(cool$schoolF)
[1] "Cal"   "Stanf"
> levels(cool$schoolF) <- c('public', 'private')
> summary(cool$schoolF) # whoop!
 public private
   2870    2868
```

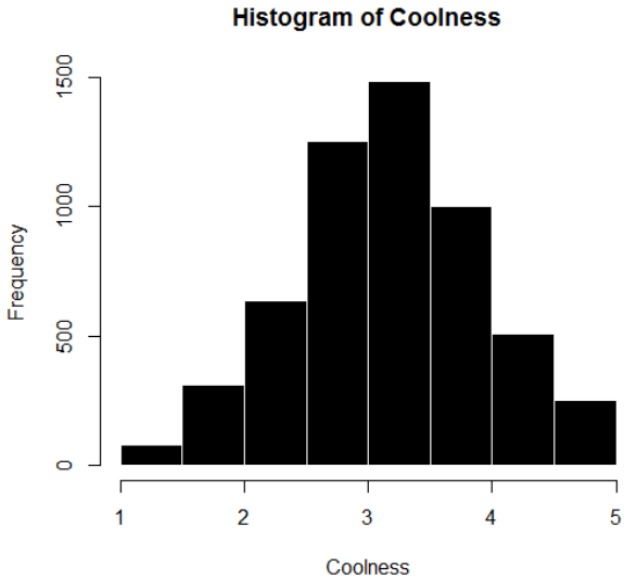
1C. After cleaning the variables, use R to print the descriptive statistics for the participants in the sample, and then describe the sample as you would for a research paper: (Report the sample size and descriptive statistics for age and sex of participants. If ethnicity was measured, which it is not in these data, that variable is usually summarized as well.)

```
> describe(cool)
           vars    n  mean   sd median trimmed  mad min  max range  skew kurtosis   se
cool1         1 5616  3.70 1.15   4.00    3.81 1.48   1  5.0   4.0 -0.76    -0.36 0.02
cool2r        2 5598  3.33 1.10   4.00    3.36 1.48   1  5.0   4.0 -0.45    -0.71 0.01
cool3         3 5610  3.56 1.01   4.00    3.60 1.48   1  5.0   4.0 -0.43    -0.44 0.01
cool4         4 5594  3.58 1.00   4.00    3.63 1.48   1  5.0   4.0 -0.47    -0.40 0.01
cool5r        5 5602  3.17 1.16   3.00    3.19 1.48   1  5.0   4.0 -0.23    -0.95 0.02
cool6         6 5598  3.31 1.10   4.00    3.33 1.48   1  5.0   4.0 -0.34    -0.74 0.01
cool7r        7 5602  3.51 1.06   4.00    3.57 1.48   1  5.0   4.0 -0.66    -0.37 0.01
cool8         8 5602  3.69 0.99   4.00    3.77 1.48   1  5.0   4.0 -0.68    -0.05 0.01
coolness      9 5518  3.23 0.75   3.25    3.23 0.74   1  5.0   4.0 -0.06    -0.20 0.01
SCH*         10 5738  1.50 0.50   1.00    1.50 0.00   1  2.0   1.0  0.00    -2.00 0.01
coffee.love  11 5738  5.45 2.64   5.00    5.44 2.97   1 10.0   9.0  0.02    -1.15 0.03
SEX          12 5738  1.50 0.51   1.00    1.50 0.00   1  3.0   2.0  0.10    -1.71 0.01
AGE          13 5738 21.12 4.28  20.00   20.49 1.48  18 64.8  46.8  6.08    46.60 0.06
sexF*        14 5738  1.50 0.51   1.00    1.50 0.00   1  3.0   2.0  0.10    -1.71 0.01
```

Participants (N = 5738) were students who volunteered for research. Students were recruited from a large public school (N = 2870) and a large private school (N = 2868) on the West Coast. On average, participants were 21.1 years old (SD = 4.28) and 50.2% participants identified as Female, 49.3% Male, and 0.5% marked Other.

1D. The variable 'coolness' comes from a scale comprised of eight questions ('cool1 – cool8') that doesn't really exist, but for the sake of this assignment pretend it was written by Drs. Catperson & Gomi in 1951. In these data, the observer rated participants using this scale from 1 = Strongly Disagree to 5 = Strongly Agree. (The questions were things like, "I think this person is cool") Some of these questions were reverse-scored – this was fortunately accounted for in the variable names (e.g., 'cool2r' = "this person is not cool they are a fool"). Create a scale that is the average of these eight items (to make sure you did this correctly compare your scale to the variable 'coolness'). Then, use R to determine whether the scale is reliable. Finally, report the descriptive statistics for this variable as you would for a research paper, and print a histogram of this variable worthy of presentation in the best academic journal (or your Final Project).

```
> cool.df <- with(cool, # with() adds the dataset to all the following objects (it's a shortcut)
+                 data.frame(cool1, (6-cool2r), cool3, cool4,
+                            (6-cool5r), cool6, (6-cool7r), cool8))
> cool$coolness2 <- rowMeans(cool.df) # calculating the scale
> head(cbind(cool$coolness, cool$coolness2)) # same values
      [,1]  [,2]
[1,] 3.250 3.250
[2,] 3.250 3.250
[3,] 3.250 3.250
[4,] 3.000 3.000
[5,] 3.375 3.375
[6,] 4.250 4.250


> alpha(cool.df)

Reliability analysis
Call: alpha(x = cool.df)

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd
      0.85      0.85    0.86      0.42 5.8 0.0054  3.2 0.75

 lower alpha upper     95% confidence boundaries
0.84 0.85 0.86

 Reliability if an item is dropped:
              raw_alpha std.alpha G6(smc) average_r S/N alpha se
cool1              0.83      0.83    0.83      0.41 4.9   0.0062
X.6...cool2r.      0.84      0.84    0.84      0.43 5.2   0.0061
cool3              0.84      0.84    0.83      0.42 5.1   0.0061
cool4              0.83      0.83    0.83      0.42 5.0   0.0061
X.6...cool5r.      0.82      0.82    0.82      0.40 4.7   0.0064
cool6              0.85      0.85    0.85      0.45 5.6   0.0058
X.6...cool7r.      0.84      0.84    0.84      0.43 5.2   0.0061
cool8              0.83      0.83    0.83      0.40 4.8   0.0063

 Item statistics
                n raw.r std.r r.cor r.drop mean   sd
cool1         5616  0.74  0.73  0.68   0.62  3.7 1.15
X.6...cool2r. 5598  0.68  0.68  0.62   0.57  2.7 1.10
cool3         5610  0.68  0.69  0.64   0.57  3.6 1.01
cool4         5594  0.69  0.71  0.66   0.59  3.6 1.00
X.6...cool5r. 5602  0.78  0.77  0.75   0.69  2.8 1.16
cool6         5598  0.61  0.60  0.50   0.47  3.3 1.10
X.6...cool7r. 5602  0.68  0.67  0.62   0.56  2.5 1.06
cool8         5602  0.75  0.76  0.72   0.66  3.7 0.99

Non missing response frequency for each item
                1    2    3    4    5 miss
cool1         0.05 0.15 0.12 0.43 0.26 0.02
X.6...cool2r. 0.11 0.43 0.19 0.21 0.06 0.02
cool3         0.02 0.14 0.25 0.41 0.17 0.02
cool4         0.02 0.14 0.24 0.42 0.17 0.03
X.6...cool5r. 0.11 0.36 0.21 0.24 0.08 0.02
cool6         0.06 0.21 0.23 0.39 0.12 0.02
X.6...cool7r. 0.14 0.49 0.16 0.17 0.04 0.02
cool8         0.02 0.12 0.19 0.48 0.19 0.02
```



Coolness was measured using the 8-item scale developed by Catperson & Gomi (1951). This scale included items such as, "I think this person is cool" and "this person is not cool they are a fool" (reverse-scored). In this study, observer ratings of participants using this scale were reliable (α = .85), and averaged 3.2 (SD = .75).

1E. Use R to print the z-score for the oldest person in the dataset and the youngest person in the dataset. What do these z-scores tell you about the shape of the distribution of age (and why?). Then, calculate the z-score of 1234th person in the dataset "by hand" in R.

```
> max(scale(cool$AGE))
[1] 10.20972
> min(scale(cool$AGE))
[1] -0.7300494
>
> (cool$AGE[1234] - mean(cool$AGE))/sd(cool$AGE)
[1] -0.02881427
> scale(cool$AGE)[1234]
[1] -0.02881427
```

The fact that the maximum z-score is 10 standard deviations above the mean, whereas the minimum z-score is just .7 standard deviations below the mean suggests that the distribution is not normally distributed, and that there is positive skew (since all the low ages are within one standard deviation from the mean). A quick look at the histogram supports this intuition.

1F. Build a linear model to predict variability in coolness (the DV) from variability in which school participants attended (the IV = schoolF). Print this model, and then type out a description of what the following statistics describe: intercept and schoolFprivate (estimate, standard error, t-value, pvalue), R-squared, F-statistics.

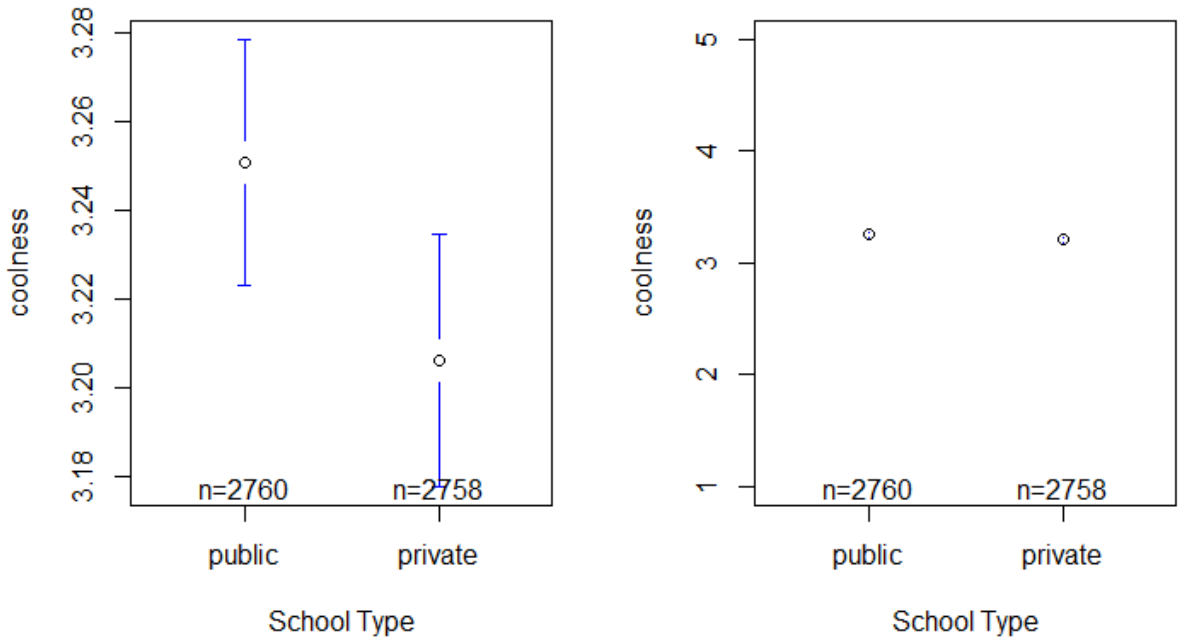
```
Call:
lm(formula = coolness ~ schoolF, data = cool)

Residuals:
     Min       1Q   Median       3Q      Max
-2.25063 -0.45595 -0.00063  0.49937  1.79405

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.25063    0.01432 226.967   <2e-16 ***
schoolFprivate -0.04469    0.02026  -2.206   0.0274 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7524 on 5516 degrees of freedom
  (220 observations deleted due to missingness)
Multiple R-squared:  0.0008814, Adjusted R-squared:  0.0007003
F-statistic: 4.866 on 1 and 5516 DF,  p-value: 0.02743
```

- Intercept :
  - the estimate is the predicted value of coolness for the intercept ('when all Xs = zero'), which in this case would be the predicted value of coolness for public school students.
  - The standard error describes the variability we'd expect in this estimate of the population due to sampling error.
  - The t-value compares this estimate to zero in units of standard error.
  - The p-value reports the probability of finding an estimate this far away from zero, given the amount of variability we'd expect there to be due to sampling. This probability is very low (less than .0000000000000002) so we expect that public school students are different from a population of students with zero coolness.
- schoolFprivate
  - this estimate is the effect of going to a private school RELATIVE to the intercept. So private school students are predicted to be .04469 cool points LESS COOL than public school students.
  - The standard error describes the variability we'd expect in this estimate of the "TRUE" effect of going to private school relative to public school (i.e., the population) due to sampling error.
  - The t-value compares this estimated effect to zero (no effect) in units of standard error.
  - The p-value reports the probability of finding an estimate this far away from zero, given the amount of variability we'd expect there to be due to sampling. This probability is small (less than .05) so we expect that private school students are different from a population of public school students.
- R-squared : this describes the percentage of variation in coolness that is explained by variation in private vs. public school attendance (.00088 or .09% of the variation in

coolness is explained by differences in school affiliation. This is a small effect.)

- **F-statistic : this reports the probability that this reduction in variation is due to chance (i.e., is error). This probability is low (less than .05) and thus we can reject the null hypothesis and say that a model accounting for school is better than a model with no IV.**

1G. Graph the results of this model in R. For a challenge worth zero points extra credit, graph your model as a bar chart with standard error bars. Double the extra credit if you use ggplot2. Cite your sources if you borrow someone's code.



I've plotted two graphs here; the graph on the left uses the default y-axis … this illustrates the small (but statistically significant) difference. The graph on the right uses a y-axis range that reflects the full scale of coolness from 1-5. Same data, VERY different interpretations!

1I. Use the objects contained within the model in 1F to calculate the following statistics "by hand":

- The estimated coolness for Cal students, and the 95% Confidence Intervals for this estimate.
- The estimated coolness for Stanford students, and the 95% Confidence Intervals for this estimate.

```
> objects(mod1F)
 [1] "assign"        "call"          "coefficients"  "contrasts"     "df.residual"   "effects"
 [7] "fitted.values" "model"         "na.action"     "qr"            "rank"          "residuals"
[13] "terms"         "xlevels"
> mod1F$coefficients[1] # estimate for cal students
(Intercept)
   3.250634
> mod1F$coefficients[1] + mod1F$coefficients[2] # estimate for stanford students
(Intercept)
   3.205946
>
> confint(mod1F)[1,] # confidence interval for the intercept (cal students)
   2.5 %   97.5 %
3.222557 3.278711
>
> confint(mod1F)[2,] # confidence interval for the EFFECT of being from stanford.
       2.5 %       97.5 %
-0.084401636 -0.004973804
> confint(mod1F)[1,] + confint(mod1F)[2,] # confidence interval for the ESTIMATE of stanford.
   2.5 %   97.5 %
3.138156 3.273737
```

- The R^2 coefficient
- The F-test

```
> actual <- mod1F$model$coolness
> predicted <- mod1F$fitted.values
> error <- mod1F$residuals
>
> ## R^2...by hand.
> SSmod <- sum(error^2) # errors in our model
> SStotal <- sum((actual - mean(actual))^2) # errors in the baseline model
> (SStotal-SSmod)/SStotal # this is Rsquared!! PRETTY NEAT.
[1] 0.0008813949
>
> ## The F-test...by hand.
> dfmod <- length(coef(mod1F))-1
> dfres <- mod1F$df.residual
> F <- ((SStotal - SSmod)/dfmod)/(SSmod/dfres)
> F # same thing as my model output!
[1] 4.866063
> 1-pf(F, df1 = 1, df2 = dfres) # and looking up the probability.
[1] 0.02743099
```

1J. Report the results of this model as you would for a research paper. Make sure to include whether this effect is considered a) statistically significant and b) theoretically meaningful.

I tested a model predicting ratings of coolness from whether participants attended a public or private school. The results of my test suggest a small, but statistically significant effect of type of school (b = -.04, 95% CI = [-.084, -.005]). Students at public schools were rated as significantly more cool (M = 3.25, 95% CI = [3.22, 3.28]) than were students at private schools (M = 3.21, 95% CI = [3.17, 3.25]). Although this difference was statistically significant, the overall variance in coolness explained by type of school was very small (R^2 = .09%) and our power to detect this population difference was relatively low (power = 56%). Thus, I do not hold much confidence in these results.

1K. Now, split the dataset into two separate datasets – one for students who attend the public school, and one for students who attended the private school. (Hint : use indexing to identify rows in the data that match the requirements you need.) Then, use the t.test() function to test for the difference in coolness between these two groups.

```
> C <- cool[cool$SCH == "Cal",] # subsetting the data
> S <- cool[cool$SCH == "Stanf",]
>
> head(C)
    cool1 cool2r cool3 cool4 cool5r cool6 cool7r cool8 coolness SCH coffee.love SEX AGE   sexF schoolF
718     5      2     4     4      2     4      2     4    4.125 Cal           8   2  20   male  public
719     4      3     3     4      4     3      3     4    3.250 Cal           5   1  22 female  public
720     5      4     4     4      2     4      4     5    3.750 Cal           9   2  21   male  public
721     4      4     3     3      2     2      3     3    3.000 Cal           5   2  19   male  public
722     4      4     4     4      3     4      4     4    3.375 Cal           1   1  18 female  public
723     4      2     5     4      1     5      2     5    4.500 Cal           2   1  22 female  public
> head(S)
  cool1 cool2r cool3 cool4 cool5r cool6 cool7r cool8 coolness   SCH coffee.love SEX AGE   sexF schoolF
1     4      4     5     4      3     2      4     4    3.250 Stanf           6   2  23   male private
2     4      4     4     4      4     4      4     4    3.250 Stanf           7   1  21 female private
3     4      3     4     3      3     3      4     4    3.250 Stanf           1   2  22   male private
4     4      4     4     4      4     2      4     4    3.000 Stanf           7   3  20  other private
5     3      4     4     4      3     4      3     4    3.375 Stanf           5   2  18   male private
6     4      1     4     4      2     4      2     5    4.250 Stanf           5   2  19   male private
>
> t.test(S$coolness, C$coolness) # same thing as model output.

        Welch Two Sample t-test

data:  S$coolness and C$coolness
t = -2.2059, df = 5513.4, p-value = 0.02743
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.08440194 -0.00497350
sample estimates:
mean of x mean of y
 3.205946  3.250634
```

1L. FINALLY, show that R did its work correctly (and you understand what a t-test really is) by calculating this statistic by hand in R. Note : you will need to remove the missing variables when calculating the sample sizes of the two datasets. Otherwise, your t-test result by hand will differ from the model output. (It's okay if you can't figure out how to remove missing variables, but try!)

```
> mS <- mean(S$coolness, na.rm = T)
> mC <- mean(C$coolness, na.rm = T)
>
> varS <- var(S$coolness, na.rm = T)
> varC <- var(C$coolness, na.rm = T)
>
> nC <- sum(!is.na(C$coolness)) # removing the na data
> nS <- sum(!is.na(S$coolness))
>
> dfC <- nC-1
> dfS <- nS-1
>
> sPool <- (dfC*varC + dfS*varS)/(dfC + dfS)
> sePool <- sqrt((sPool / nC)+(sPool / nS))
> meanDiff <- mS - mC
> t <- meanDiff / sePool
> t # WHOOP!
[1] -2.205916
>
```