P8130 Biostatistical Methods Homework 5

Emil Hafeez (eh2928)

11/13/2020

Problem 1

```
#Read the CSV data into a dataframe
antibodies_df <- read.csv("./data/Antibodies.csv")

# Make AgeCategory, Smell, and Gender appopriate datatypes. Helps to ensure we know all unique values,
unique(antibodies_df$AgeCategory)
    antibodies_df <- antibodies_df %>%
        mutate(AgeCategory = factor(AgeCategory, labels = c("18-30", "31-50", "51+") ))
unique(antibodies_df$Smell)
    antibodies_df <- antibodies_df %>%
        mutate(Smell = factor(Smell, levels = c("Normal","Altered", "Unanswered/Others")))
    antibodies_df = antibodies_df %>% filter(Smell != "Unanswered/Others")
unique(antibodies_df$Gender)
    antibodies_df <- antibodies_df %>%
        mutate(Gender = factor(Gender, levels = c("Male","Female")))
```

In order to assess the difference in IgM levels between the two smell factor groups (Normal vs Altered) given non-normal distributions, we opt for a non-parametric test called the Wilcoxon Rank-Sum test. It's the nonparametric equivalent of the two-sample independent t-test, and examines if the medians of the two populations are equal versus not equal:

H_0 = the medians of the two groups' IgM levels are equal, and H_A = the medians of the two groups' IgM levels are not equal. The decision rule is that we reject H_0 = if $T > z_{1-(\alpha/2)}$.

The test statistic is computed, with a continuity correction, one of two ways. We first combine the data from the two groups, order the values from lowest to highest, assign ranks to the individual values (1 to n), and if ties, assign the average rank. Then, select a group and compute the sum of ranks T_1 for the first group, and then use the appropriate test statistic formula.

With no ties (referring to two equally ranked values once the values are listed), the test statistic is

$$T = \frac{\left|T_1 - \frac{n_1(n_1+n_2+1)}{2}\right| - \frac{1}{2}}{\sqrt{(n_1 n_2/12)(n_1+n_2+1)}}$$

and with ties, the test statistic is

$$T = \frac{\left|T_1 - \frac{n_1(n_1+n_2+1)}{2}\right| - \frac{1}{2}}{\sqrt{(n_1 n_2/12)\left[(n_1+n_2+1) - \sum_{i=1}^{g} t_i\left(t_i^2-1\right)/(n_1+n_2)(n_1+n_2-1)\right]}}$$

where t_i refers to the number of observations with the same absolute value in the i^{th} group and g is the number of tied groups.

In our case, the test statistic calculated by R is slightly different, since it does not by default add the $n_1(n_1+1)/2$ term (and is denoted by W).

The p-value under the normal approximation, with n_1 and $n_2 \geq 10$ is described by $2 * [1 - \Phi(T)]$.

```
antibodies_df2 =
  antibodies_df %>%
  pivot_wider(
        names_from = Smell,
        values_from = Antibody_IgM
    )

wilcox.test(antibodies_df2$Normal, antibodies_df2$Altered, mu = 0)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  antibodies_df2$Normal and antibodies_df2$Altered
## W = 5836, p-value = 0.01406
## alternative hypothesis: true location shift is not equal to 0
```

In context, and ignoring missing values and the unanswered smell category, we find evidence to reject the null hypothesis and conclude that the true location shift between the Normal and Altered smell categories is not equal to zero (in other words, the median IgM values are different for the two groups).

Problem 2

Let's come back to this.

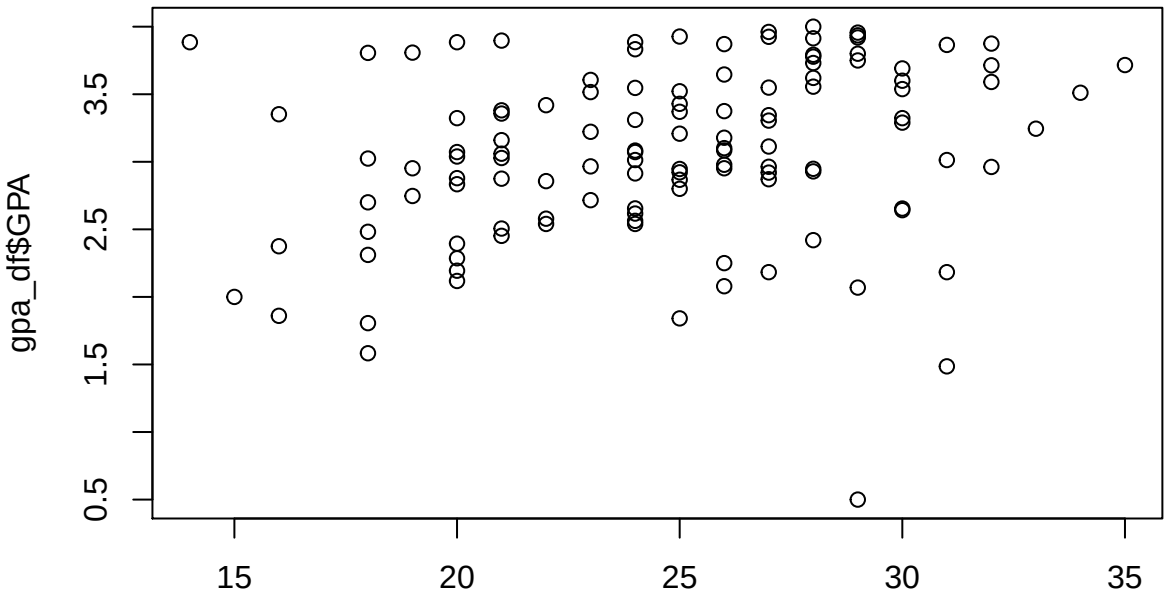
Problem 3

Problem 3 Part 1

Load the data and plot it.

```
gpa_df <- read.csv("./data/GPA.csv")

plot(gpa_df$ACT, gpa_df$GPA)
```



Test whether a linear association exists between ACT score (x) and GPA at the end of freshman year (Y).

Let $\alpha = 0.05$ and let β_1 represent the true slope to be estimated.

The null hypothesis is $H_0 : \beta_1 = \beta_{10}$ where $\beta_{10} = 0$. The alternative hypothesis is $H_A : \beta_1 \neq \beta_{10}$. In context, testing $H_0 : \beta_1 = 0$ examines whether a student's GPA at the end of freshman year can be predicted from the ACT test score.

The test statistic follows the t distribution with n-2 degrees of freedom, such that

$t = \frac{\widehat{\beta_1} - \beta_{10}}{se\left(\widehat{\beta_1}\right)} \sim t_{n-2}$, under $H_0 = \frac{0.03883 - 0}{0.01277} = 3.040$ using degrees of freedom $= n = 120, df = n - 2$

The corresponding critical value is fixed by $t_{n-2,1-(\alpha/2)}$ and the decision rule is that we reject H_0 if $|t| > t_{n-2,1-(\alpha/2)}$ and fail to reject H_0 if $|t| \leq t_{n-2,1-(\alpha/2)}$. As such, the critical value is $t_{118,0.975} = 1.980272$.

Therefore, $|t| > t_{118,0.975}$ using the 5% significance level, we find evidence to reject the null hypothesis and conclude that there is a significant linear association between students' ACT scores and GPA at the end of freshman year.

```
reg_admit<-lm(gpa_df$GPA~gpa_df$ACT)

# Summarize regression
summary(reg_admit)
```

```
##
## Call:
## lm(formula = gpa_df$GPA ~ gpa_df$ACT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74004 -0.33827  0.04062  0.44064  1.22737
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.11405    0.32089   6.588  1.3e-09 ***
## gpa_df$ACT   0.03883    0.01277   3.040  0.00292 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6231 on 118 degrees of freedom
## Multiple R-squared:  0.07262,    Adjusted R-squared:  0.06476
## F-statistic:  9.24 on 1 and 118 DF,  p-value: 0.002917
```

```
tidy(reg_admit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic       p.value
##   <chr>          <dbl>     <dbl>     <dbl>         <dbl>
## 1 (Intercept)   2.11      0.321       6.59 0.00000000130
## 2 gpa_df$ACT    0.0388    0.0128      3.04 0.00292
```

```
glance(reg_admit)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1    0.0726        0.0648 0.623      9.24 0.00292     1  -113.  231.  239.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
# Regression objects
names(reg_admit)
```

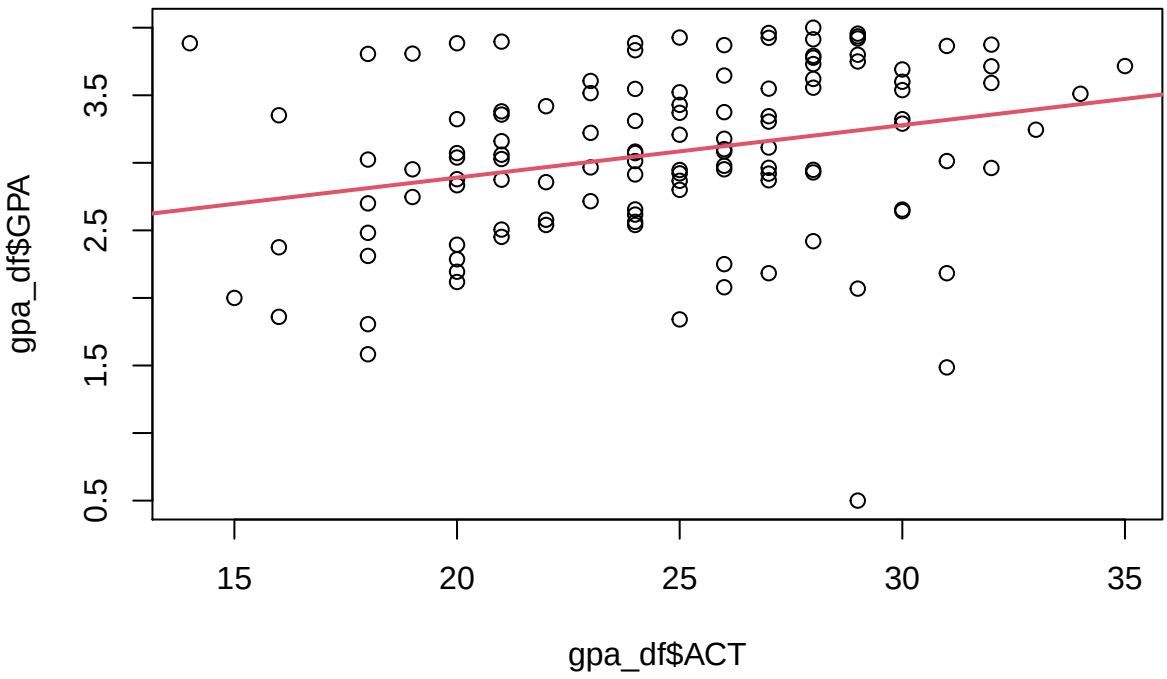
```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
# Get fitted.values
reg_admit$fitted.values
```

```
##        1        2        3        4        5        6        7        8
## 2.929419 2.657629 3.201209 2.968246 2.929419 3.317690 3.356517 3.162382
##        9       10       11       12       13       14       15       16
## 3.240036 3.123555 3.045900 3.278863 3.045900 3.045900 3.395344 3.162382
##       17       18       19       20       21       22       23       24
## 3.084727 3.317690 3.084727 2.890592 3.045900 2.929419 3.201209 3.162382
##       25       26       27       28       29       30       31       32
## 3.201209 3.123555 3.201209 2.968246 3.123555 2.929419 3.084727 2.735283
##       33       34       35       36       37       38       39       40
## 3.201209 3.123555 2.968246 3.045900 2.929419 3.278863 3.162382 3.123555
##       41       42       43       44       45       46       47       48
## 3.123555 3.278863 3.045900 3.123555 3.240036 3.045900 3.317690 2.696456
##       49       50       51       52       53       54       55       56
## 2.851765 2.812938 3.162382 2.735283 3.162382 3.123555 3.045900 3.278863
##       57       58       59       60       61       62       63       64
## 2.929419 2.890592 3.278863 3.240036 3.084727 3.007073 3.084727 3.007073
##       65       66       67       68       69       70       71       72
## 3.278863 2.929419 3.045900 3.356517 2.812938 3.007073 2.890592 3.007073
##       73       74       75       76       77       78       79       80
## 2.812938 2.812938 3.240036 2.890592 3.007073 3.123555 3.201209 3.434172
##       81       82       83       84       85       86       87       88
## 2.890592 2.890592 3.123555 3.356517 3.084727 3.162382 3.162382 3.240036
##       89       90       91       92       93       94       95       96
## 2.851765 2.929419 3.045900 3.162382 3.084727 2.812938 3.240036 3.045900
```

```
##       97       98       99      100      101      102      103      104
## 3.162382 2.929419 2.851765 2.812938 3.084727 2.812938 2.890592 3.356517
##      105      106      107      108      109      110      111      112
## 3.045900 3.472999 3.084727 3.201209 3.201209 3.084727 2.968246 3.278863
##      113      114      115      116      117      118      119      120
## 2.890592 2.890592 3.317690 2.890592 3.240036 3.201209 2.735283 3.201209
```

```
# Scatterplot and regression line overlaid
plot(gpa_df$ACT, gpa_df$GPA)
abline(reg_admit,lwd=2,col=2)
```



Problem 3 Part 2

The basic regression model follows the form $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ and the estimated regression model is given by $\widehat{Y} = \widehat{\beta_0} + \widehat{\beta_1} X_i, i = 1, 2, 3...n$. In our context, this means that the estimated GPA value is equal to the sum of the intercept β_0 and the estimated beta one times the student's ACT score, such that $\widehat{GPA} = 2.11405 + 0.03883 \cdot ACT$.

Problem 3 Part 3

The 95% confidence interval for the true slope β_1 is given by $\widehat{\beta_1} \pm t_{n-2,1-(\alpha/2)} \cdot se(\widehat{\beta_1})$

where $se(\widehat{\beta_1}) = \sqrt{MSE/\Sigma_{i=1}^{n}}(X_i - \overline{X})^2$