32.75/33, great work!

My first assignment

Ahmad Saquib Sina

February 6, 2019

Load all libraries

Read in data

```
HF = read_csv("C:/Users/Ahmad Saquib Sina/Desktop/HolzingerSwineford1939.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   gender = col_character(),
##   school = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
head(HF)
```

```
## # A tibble: 6 x 32
##      id gender grade  agey  agem school visual cubes paper flags general
##   <dbl> <chr>  <dbl> <dbl> <dbl> <chr>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1     1 Male       7    13     1 Paste~     20    31    12     3      40
## 2     2 Female     7    13     7 Paste~     32    21    12    17      34
## 3     3 Female     7    13     1 Paste~     27    21    12    15      20
## 4     4 Male       7    13     2 Paste~     32    31    16    24      42
## 5     5 Female     7    12     2 Paste~     29    19    12     7      37
## 6     6 Female     7    14     1 Paste~     32    20    11    18      31
## # ... with 21 more variables: paragrap <dbl>, sentence <dbl>, wordc <dbl>,
## #   wordm <dbl>, addition <dbl>, code <dbl>, counting <dbl>,
## #   straight <dbl>, wordr <dbl>, numberr <dbl>, figurer <dbl>,
## #   object <dbl>, numberf <dbl>, figurew <dbl>, deduct <dbl>,
## #   numeric <dbl>, problemr <dbl>, series <dbl>, arithmet <dbl>,
## #   paperrev <dbl>, flagssub <dbl>
```

Question one

```
typeof(HF$gender)
```

```
## [1] "character"
```

```
typeof(HF$sentence)
```

```
## [1] "double"
```

From the R, the variable type for gender is vector of characters as the storage type is character. On the other hand, the variable type for sentence is vector of decimals as the storage type is double.These variables are not the same type. The scale of measurement for the vector of characters is nominal which is not ordered. The scale of measurement for the vector of decimals is ratio.

OK

Not really. Ratio data means that ratios are meaningful, which isn't usually true for psychological variables. 0 doesn't mean the person actually is illiterate, for instance -- it's more of a floor than a true zero point. A person who has the score 8 doesn't really have " twice as much" as someone who has the score 4. However, someone 6 feet tall is *really* twice as tall as 3, because zero has a concrete interpretation, unlike here

Question two

```
summary(HF$sentence)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.00   14.00   18.00   17.36   21.00   28.00
```

The minimum scores on the sentence test is 4 and maximum scores is 28.

Question three

```
aggregate(sentence ~ school, data = HF, FUN = mean)
```

```
##        school sentence
## 1 Grant-White 18.84828
## 2     Pasteur 15.98077
```

The mean on the sentence test at Grand white is 18.84828 and the mean on the sentence test at Pasteur is 15.98077.

Question four

```
cor(HF$wordc, HF$sentence)
```

```
## [1] 0.6744079
```

The correlation between wordc and sentence test is 0.6744079

Quesiton five

```
subset(HF, id == 90)
```

```
## # A tibble: 1 x 32
##      id gender grade  agey  agem school visual cubes paper flags general
##   <dbl> <chr>  <dbl> <dbl> <dbl> <chr>   <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1    90 Male       7    14     5 Paste~     29    33     6    18      18
## # ... with 21 more variables: paragrap <dbl>, sentence <dbl>, wordc <dbl>,
## #   wordm <dbl>, addition <dbl>, code <dbl>, counting <dbl>,
## #   straight <dbl>, wordr <dbl>, numberr <dbl>, figurer <dbl>,
## #   object <dbl>, numberf <dbl>, figurew <dbl>, deduct <dbl>,
## #   numeric <dbl>, problemr <dbl>, series <dbl>, arithmet <dbl>,
## #   paperrev <dbl>, flagssub <dbl>
```

THe school for id=90 is pasteur and their gender is Male.

Question six

```
table(HF$gender)
```

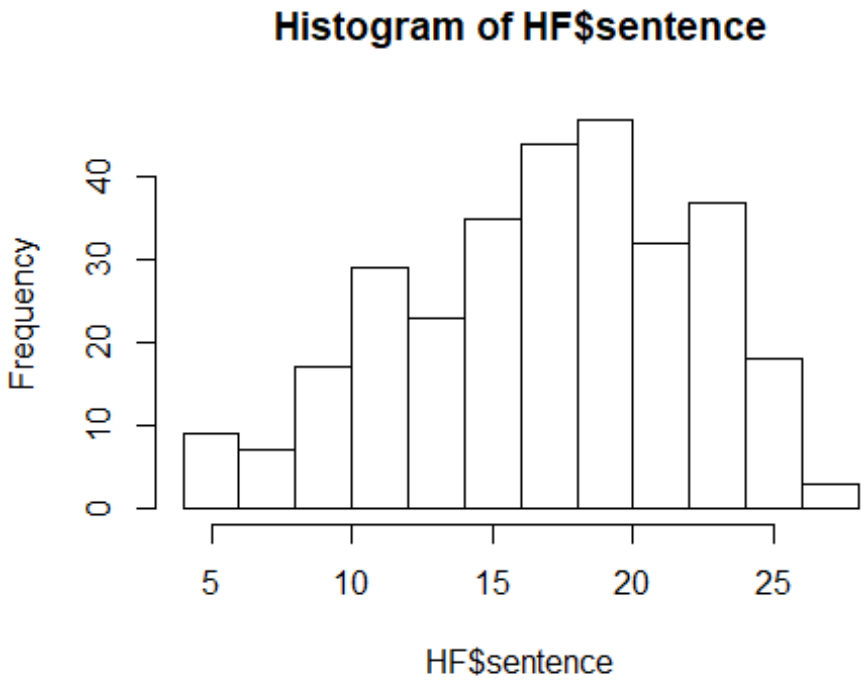
```
##
## Female   Male
##    155    146
```

Part 1: 6.5/6.5

Female are 155 and Male are 146.

Question 7

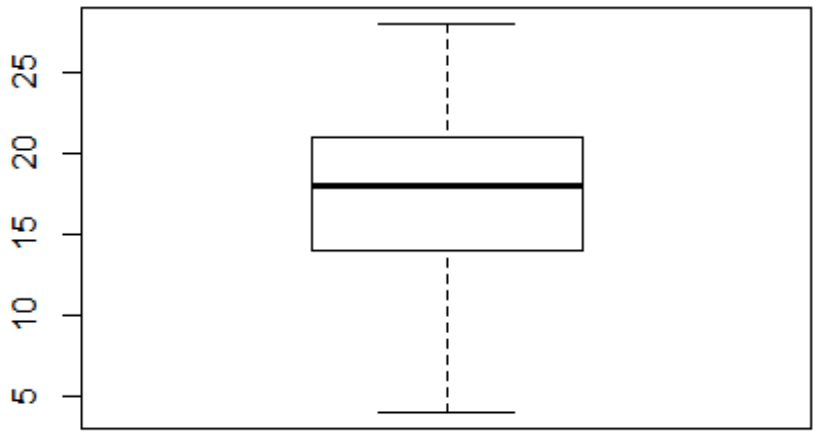
```
hist(HF$sentence)
```




```
stem(HF$sentence)

##
##   The decimal point is at the |
##
##    4 | 00
##    5 | 0
##    6 | 000000
##    7 | 000
##    8 | 0000
##    9 | 00000000
##   10 | 000000000
##   11 | 00000000000
##   12 | 0000000000000000000
##   13 | 0000000000000
##   14 | 0000000000
##   15 | 0000000000000
##   16 | 00000000000000000000
```

```
##    17 | 00000000000000000000000
##    18 | 0000000000000000000
##    19 | 000000000000000000000000000
##    20 | 00000000000000000
##    21 | 000000000000000
##    22 | 00000000000000000
##    23 | 0000000000000000000000000
##    24 | 00000000000
##    25 | 00000000000
##    26 | 0000000
##    27 | 00
##    28 | 0
```

```
boxplot(HF$sentence)
```

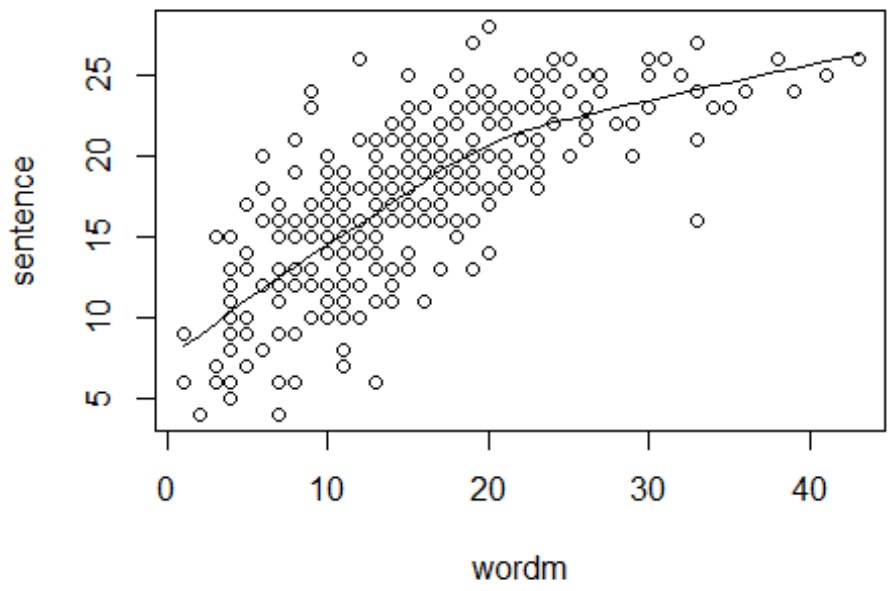


(a)From the histogram, it is visible that data in sentence are not normally discrtibuted. It is skewed to the left that is negative skewness. (b) It seems that there are no outliers from the stem and boxplot. (c) In my view, a transformation would be necessary to make it normal as data are not normally distributed. In my view, log transformation of data could be very helpful to make the skewed distribution into a more normal model.

But, we lose interpretability with a transformation, so it's always a tradeoff

Question 8

```
plot(sentence~wordm, data=HF)
lines(lowess(x = HF$wordm, y = HF$sentence))
```



The scatterplot shows that (a)the direction is postive, (b) the relationship is strong,all points are very close (c)the scatterplot is non-linear, the scatterplot is not linear as there is curvy in the middle and (d) there are outliers below and above the line and some outliers in the edge that are shown in the scatterplot.

-0.25 a bit vague -- which points are you considering outliers?

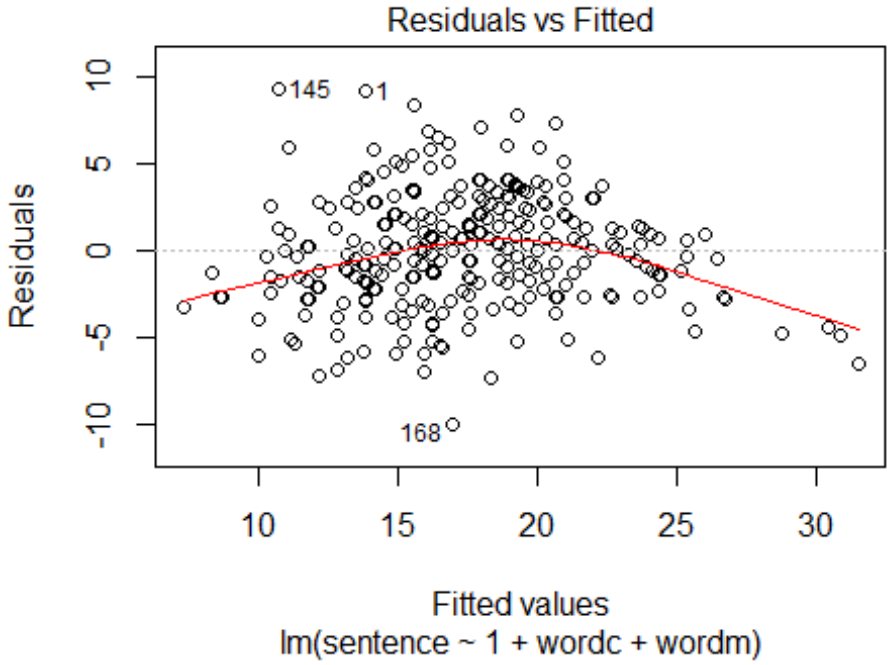
Multiple regression

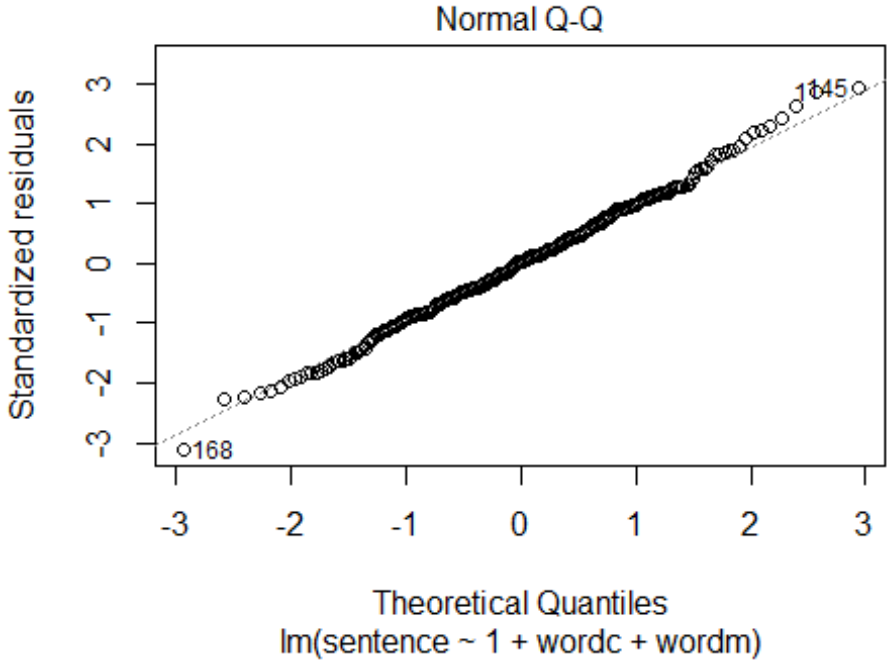
Part 2: 7.25/7.5

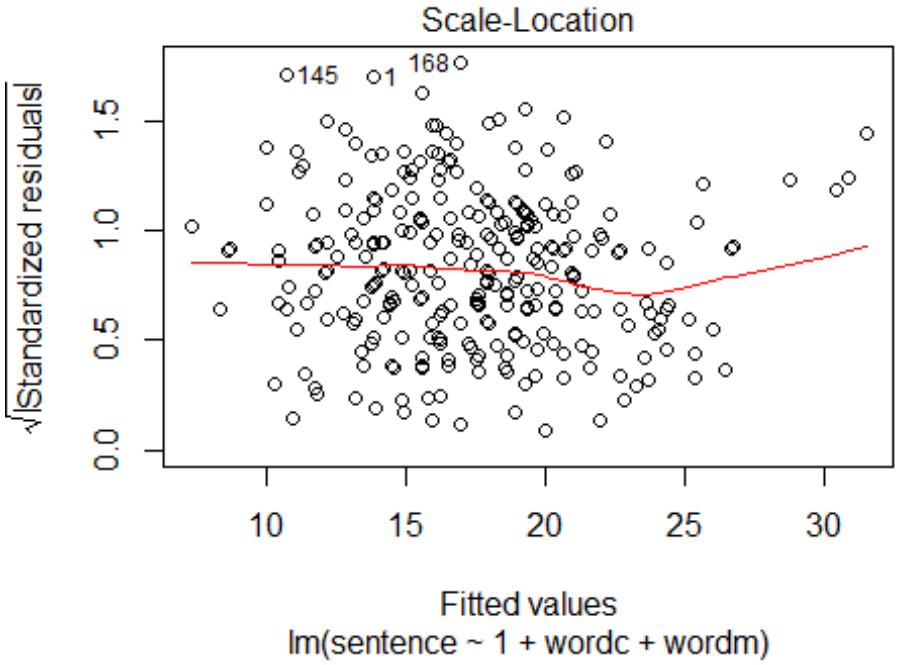
```
fit.lm=lm(sentence~1+wordc+wordm,data=HF)
fit.sem=sem(sentence~1+wordc+wordm,data=HF)

## Warning in lavaan::lavaan(model = sentence ~ 1 + wordc + wordm, data =
## HF, : lavaan WARNING: model seems to be a formula; please enclose the
model
## syntax between quotes
```

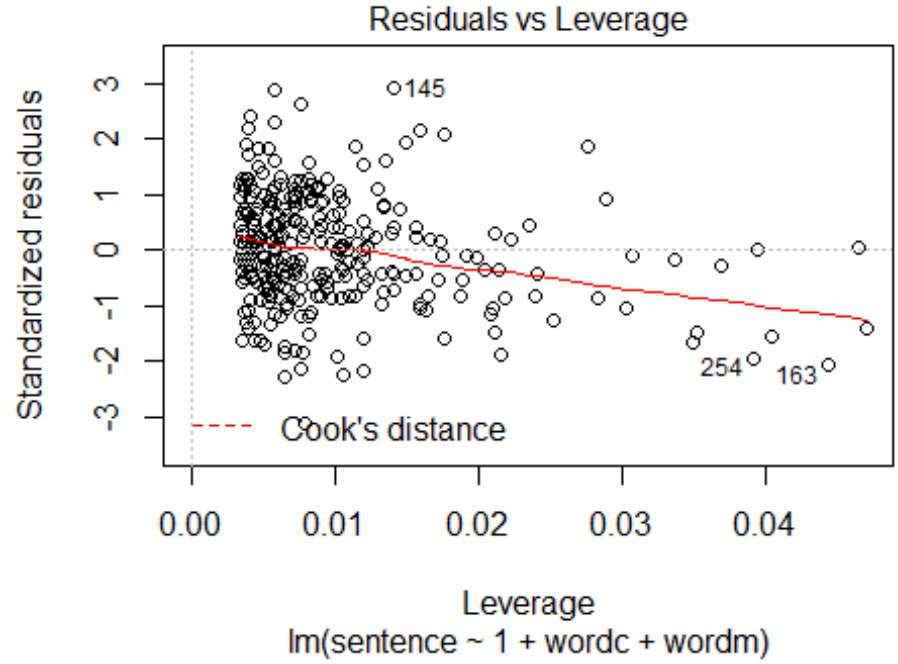
Question 9

```
# View modeling assumptions
plot(fit.lm)
```

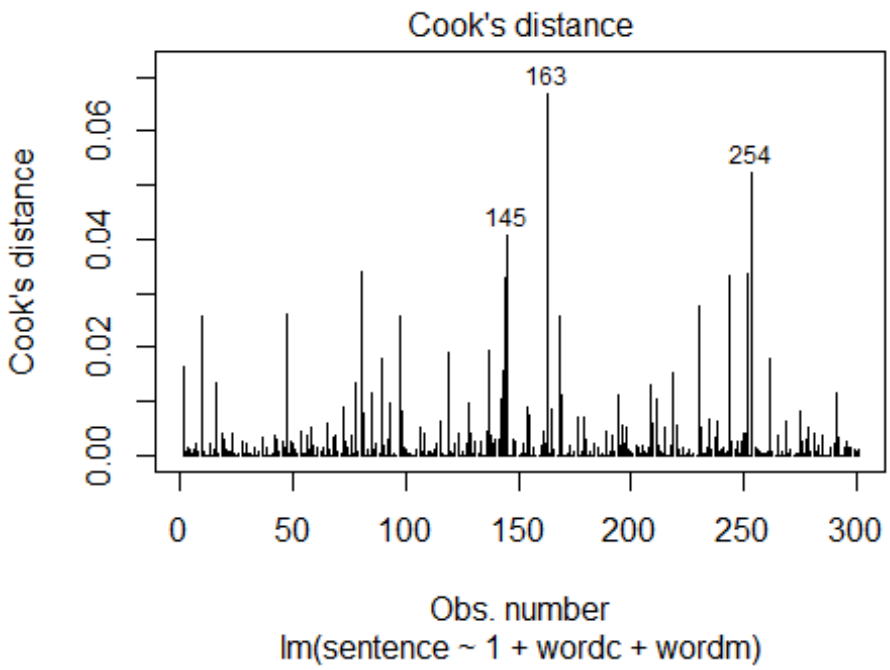

Residuals vs Fitted
Residuals
Fitted values
lm(sentence ~ 1 + wordc + wordm)
145
1
168
10
5
0
-5
-10
10
15
20
25
30


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(sentence ~ 1 + wordc + wordm)
145
168
3
2
1
0
-1
-2
-3

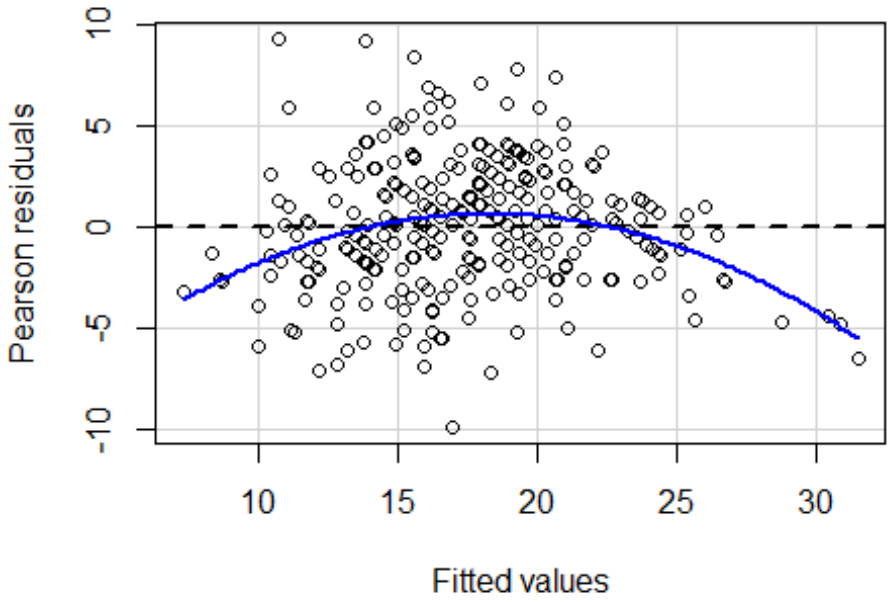





```r
plot(fit.lm, which = 4) # Cook's distance
```




```
residualPlot(fit.lm)
```



a) From the residual plot, The homogeneity of variance for residuals is likely OK; the fanning is not severe; there is some fanning in the edge of the residual b) Yes, there are outliers that are present in residuals versus fitted plot and those points are 1, 145, and 168 c) From the cook's distance, we have found the influential points and they are 145, 164, and 254.d) The normality assumption is not likely violated; the upper end of the distribution is not deviated from what would be expected from a normal distribution in the QQ-plot. The normality seems roughly normal.

Question 10

```
summary(fit.lm)

##
## Call:
## lm(formula = sentence ~ 1 + wordc + wordm, data = HF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9390 -2.0554  0.0971  2.1153  9.2959
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.08154    0.88526   3.481 0.000574 ***
## wordc        0.35141    0.04008   8.768  < 2e-16 ***
## wordm        0.33333    0.02966  11.239  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.205 on 298 degrees of freedom
## Multiple R-squared:  0.6171, Adjusted R-squared:  0.6146
## F-statistic: 240.2 on 2 and 298 DF,  p-value: < 2.2e-16

summary(fit.sem)

## lavaan 0.6-3 ended normally after 15 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                          4
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Model Fit Test Statistic                       0.000
##   Degrees of freedom                                 0
##   Minimum Function Value               0.0000000000000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
```

```
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   sentence ~
##     wordc             0.351    0.040    8.812    0.000
##     wordm             0.333    0.030   11.295    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .sentence          3.082    0.881    3.498    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .sentence         10.168    0.829   12.268    0.000
```

(a) The unstandadized estimates from the regression model for wordc and wordm are 0.35 and 0.33.(b) Yes, they are statiscally significant because for both of them p value is less than 0.001 as we are assuming alpha vale is 0.05 (c) interpretation: The interpretations are: each one unit increase or difference in word classification test is associated with a 0.35 unit predicted increase or difference in sentence test after controlling for the differecne of word meaning test. Also, each one unit difference or increase in word meaning test is associated with a 0.33 unit predicted increase or difference in sentence test after controlling for the difference of word classification test.

Question 11

Some results are same using the sem and lm function, however, some are not same. For example, the values of estimates and standard errors are same for both function.From the lm function, t value is obtained. However, sem function gives z value. Also, residual standatrd error is 3.205 on 298 degrees of freedom for lm fundtion. However, the degrees of freedom from sem function is 0 (zero). From lm function, we get multiple and adjusted R squaured value. On the other hand, sem function does not give multiple and adjusted R squared value.

Part 3: 8.5/8.5

Question 12

```
summary(fit.sem, standardized = TRUE)
```

```
## lavaan 0.6-3 ended normally after 15 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                          4
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Model Fit Test Statistic                       0.000
##   Degrees of freedom                                 0
##   Minimum Function Value               0.0000000000000
```

```
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   sentence ~
##     wordc             0.351    0.040    8.812    0.000    0.351    0.386
##     wordm             0.333    0.030   11.295    0.000    0.333    0.495
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .sentence          3.082    0.881    3.498    0.000    3.082    0.598
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .sentence         10.168    0.829   12.268    0.000   10.168    0.383
```

The standardized estimates that is the beta weights for wordc and wordm are 0.386 and 0.495. interpretation: The interpretations are each one standard deviation increase or difference in word classification test is associated with a 0.386 standard deviation predicted increase or difference in sentence test after controlling for the difference of word meaning test. Also, each one standard deviation increase or difference in word meaning test is associated with a 0.495 standard deviation predicted increase or difference in sentence test after controlling for the difference of word classification test.

Question 13

```
# Confidence intervals
confint(fit.lm)
```

```
##                 2.5 %    97.5 %
## (Intercept) 1.3393857 4.8236986
## wordc       0.2725404 0.4302791
## wordm       0.2749611 0.3916922
```

```
summary(fit.sem, ci = TRUE)
```

```
## lavaan 0.6-3 ended normally after 15 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                          4
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Model Fit Test Statistic                       0.000
##   Degrees of freedom                                 0
```

```
##   Minimum Function Value                0.0000000000000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   sentence ~
##     wordc             0.351    0.040    8.812    0.000    0.273    0.430
##     wordm             0.333    0.030   11.295    0.000    0.275    0.391
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .sentence          3.082    0.881    3.498    0.000    1.355    4.808
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .sentence         10.168    0.829   12.268    0.000    8.543   11.792
```

(a)From the regression model, The lower confidence interval for wordc and wordm are 0.273 and 0.275 respectively. The upper confidence interval for wordc and wordm are 0.430 and 0.392 respectively. The interpretations are that each one unit difference or increase in word classification test is associated with a difference or increase in sentence test between 0.273 and 0.430 units after controlling for the differences of word meaning test. Also, each one unit difference in word meaning test is associated with a difference or increses in sentence test between 0.275 and 0.392 units after controlling fro the differences of word classification test.b) we have found that p value is less than 0.05. If p value is less than 0.05, then confidence interval does not include the null hypothesis. Also, the null hypothesis is equal to zero. Here, we have found that confidence intervals for both word and wordm do not contain 0. Therefore, for both wordc and wordm, confidence intervals do not include null hypothesis so that the results are statistically significant.

Question 14

```r
# Obtain fit measures from lavaan
summary(fit.sem, fit.measures = TRUE)
```

```
## lavaan 0.6-3 ended normally after 15 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                          4
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Model Fit Test Statistic                       0.000
##   Degrees of freedom                                 0
```

```
##   Minimum Function Value               0.0000000000000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             288.970
##   Degrees of freedom                                2
##   P-value                                       0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                   1.000
##   Tucker-Lewis Index (TLI)                      1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -776.147
##   Loglikelihood unrestricted model (H1)      -776.147
##
##   Number of free parameters                         4
##   Akaike (AIC)                               1560.293
##   Bayesian (BIC)                             1575.121
##   Sample-size adjusted Bayesian (BIC)        1562.436
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.000
##   90 Percent Confidence Interval          0.000  0.000
##   P-value RMSEA <= 0.05                           NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   sentence ~
##     wordc             0.351    0.040    8.812    0.000
##     wordm             0.333    0.030   11.295    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .sentence          3.082    0.881    3.498    0.000
##
## Variances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .sentence         10.168    0.829   12.268    0.000
```

(a) The unusual about the fit measures for the fit.sem model is degrees of freedom is zero. For, model test basedline model the degress of freedom is 2 which is very small. (b) The fit statistics for RMSEA, TLI, and CFI are 0,1,and 1 respectively.and (b) the degrees of freedom of this model is 0 however, for the model test basedline model the degress of freedom is 2. In Structural equation modeling, we specify that variables exist in the matrix (i.e., they have variances) and we make no other claims about them at all. No means, no regression paths, no intercepts, no correlations or covariances, no correlated residuals.Also, Number of observations available for model estimation and Number of observations used to estimate parameters might be equal therefore, the degrees of freedom is zero. However, for the model test baseline model there might some differences between them so that the degrees of freedom is 2.

? This isn't true, nor relevant

Question 15

```
# Obtain the residuals and predicted values
resid(fit.lm)
```

```
##           1           2           3           4           5           6
##  9.18750259 -1.81249741 -1.29843961  1.87229924 -1.86666658 -1.86674687
##           7           8           9          10          11          12
##  3.00681158  2.13327320  6.90846555  2.13333342 -0.47915070 -0.90281281
##          13          14          15          16          17          18
## -3.30847206  0.46661992  2.43053391  5.92644833  1.48468300 -3.05542837
##          19          20          21          22          23          24
## -2.69596787 -2.51525678 -1.44304461 -1.40681809  3.80002685 -1.14584413
##          25          26          27          28          29          30
## -0.56954639 -1.55146323 -2.77637125 -1.47917077 -3.12772083 -0.74016480
##          31          32          33          34          35          36
## -1.49721378  0.25979505  2.00677143  0.79996663  1.29594128  4.50276615
##          37          38          39          40          41          42
## -0.53336001 -3.51529693 -0.64175856  0.74579747 -1.62373562  3.57507869
##          43          44          45          46          47          48
##  2.85413580 -2.14582406  2.30395184  2.83609279 -5.98315598 -1.16390721
##          49          50          51          52          53          54
## -2.74020495  3.02485458 -1.44306469  0.69148779 -2.68601571  0.64741130
##          55          56          57          58          59          60
## -1.32655521  4.20558574 -1.75824796  5.16947965 -2.05544844  0.25979505
##          61          62          63          64          65          66
##  3.43051383  2.07908397 -2.12778105 -0.58756932  5.85417594 -2.62369548
##          67          68          69          70          71          72
##  0.09710690 -5.20003337 -5.18197029 -1.49717363  0.96261464 -5.88480995
##          73          74          75          76          77          78
## -4.18197029  2.69154801 -0.18201043 -3.79443433  1.44857691 -3.96507283
##          79          80          81          82          83          84
## -2.14580398 -5.28029624 -5.77633111  0.41243068 -1.74022502  0.11516998
##          85          86          87          88          89          90
## -6.79443433 -1.39878746  2.83607272  1.37626438 -7.14586420  3.18752266
```

```
##          91          92          93          94          95          96
##  0.07111355  2.48464285 -3.66787234 -0.47917077  1.43049376 -0.53333994
##          97          98          99         100         101         102
##  6.57513891  7.74577740 -2.79441426 -2.23611938 -0.84866372 -1.21811652
##         103         104         105         106         107         108
##  0.74573725 -0.88472966 -0.88462929  5.83613294  1.69156809 -3.77631104
##         109         110         111         112         113         114
##  0.44865721  1.25981512 -1.97510527  1.81802971 -2.90285296  2.69150787
##         115         116         117         118         119         120
## -4.81251749  1.39432746  0.09718719 -0.38879516 -6.93903933 -1.42490124
##         121         122         123         124         125         126
##  1.07904382 -4.18195021  3.69158816  0.16943951  1.81808993  0.06096067
##         127         128         129         130         131         132
##  3.13335350  4.90844547  4.00679150 -1.14586420  2.98874850  0.85421609
##         133         134         135         136         137         138
## -2.38881523 -0.84862357  0.35820130  6.07910404  6.22372911  4.83613294
##         139         140         141         142         143         144
## -2.67794493  3.78194370  3.34011815 -5.10971797 -4.67784457 -4.41675017
##         145         146         147         148         149         150
##  9.29594128  0.78194370  0.02487466 -3.79441426 -2.34453800  0.44851669
##         151         152         153         154         155         156
## -0.55130265  4.07908397 -1.58756932 -5.88480995 -5.90287303  0.72767417
##         157         158         159         160         161         162
##  2.85417594  0.04289759  0.09718719  3.46661992  4.07914419  4.07910404
##         163         164         165         166         167         168
## -6.50714586  1.06098074  2.95250190  3.09716712 -0.20003337 -9.93901926
##         169         170         171         172         173         174
## -7.29040898 -0.28031632  0.06108111 -1.07351152  3.72769424  0.16960009
##         175         176         177         178         179         180
## -1.88472966 -2.67782450  0.76380033 -1.11969020  5.07914419  2.61118477
##         181         182         183         184         185         186
## -0.60567254  2.11525027  0.44859699 -2.84866372 -2.95706227  1.11525027
##         187         188         189         190         191         192
##  1.50278622 -0.62369548 -5.23613945  0.20554559 -1.29044913  4.02485458
##         193         194         195         196         197         198
##  2.41245075 -3.26229338 -1.71413131 -2.73207395 -2.65982164  3.48478336
##         199         200         201         202         203         204
## -2.18199036  1.39430738  1.81802971  2.76388062 -1.68595549 -1.32653514
##         205         206         207         208         209         210
##  4.11525027  1.67346486  1.53893245  3.09714704 -6.18201043 -3.36272152
##         211         212         213         214         215         216
##  7.35822137  3.76382040  2.41243068  1.02493487  4.13325313  0.48472314
##         217         218         219         220         221         222
## -0.41683046  2.85415587 -5.06552103 -2.63176626 -1.86674687  0.18754273
##         223         224         225         226         227         228
## -2.95706227  0.44859699 -1.58758939 -2.27234590 -0.84864364 -1.88474973
##         229         230         231         232         233         234
## -0.53333994 -4.73203381 -3.09153445 -0.90279274  0.67354515 -3.55144316
##         235         236         237         238         239         240
## -5.58760947  2.41239053  3.06108111  7.07908397  2.02489473  2.06106103
```

```
##         241         242         243         244         245         246
##  3.76384048  0.97070549  1.26782568  5.91637574 -2.67796501 -2.01127158
##         247         248         249         250         251         252
##  2.85423616 -1.29034876  3.69154801  5.48470307  3.67344479 -4.85857573
##         253         254         255         256         257         258
## -0.27222546 -6.16374662  1.41253105 -2.01127158  1.50282637  0.61931577
##         259         260         261         262         263         264
##  1.44857691  1.39436760  1.24978268  8.43045361 -1.34461829  0.35826152
##         265         266         267         268         269         270
## -2.58748903  0.32211529  0.97074564  0.72773439 -3.39876738 -1.20003337
##         271         272         273         274         275         276
##  1.65536164 -0.29038891 -0.67796501  1.39432746 -3.36274159 -4.53333994
##         277         278         279         280         281         282
##  1.06100081  3.53895253 -5.49719370  2.15137643  3.41237046  1.30401206
##         283         284         285         286         287         288
##  3.18752266  0.34013822 -3.62375570  0.44853677 -0.34451793  0.46661992
##         289         290         291         292         293         294
## -2.64175856  3.44853677  5.07902375 -4.23619967  0.46663999  0.41241061
##         295         296         297         298         299         300
## -2.88478988 -3.64177863  2.09710690  2.50274608 -1.32645485 -1.34463836
##         301
## -1.11971027
```

```
resid(fit.sem)
```

```
## $type
## [1] "raw"
##
## $cov
##          sentnc wordc wordm
## sentence 0
## wordc    0      0
## wordm    0      0     0
##
## $mean
## sentence    wordc    wordm
##        0        0        0
```

(a) No, the residuals values differ for these two types of models. From the lm, we get residuals for each individual data. However, from SEM, we get the residual covariance matrix (b) From residual (fit.sem), we get the residual covariance matrix. SEM is known as analysis of covariance structures.These residuals are actually residual covariance matrix from fit.sem

Question 16

```
predict(fit.lm)
```

```
##         1         2         3         4         5         6         7
## 13.812497 13.812497  8.298440 16.127701 17.866667 13.866747 20.993188
##         8         9        10        11        12        13        14
```

```
## 14.866727 16.091534 17.866667 14.479151 18.902813 19.308472 15.533380
##        15        16        17        18        19        20        21
## 19.569466 11.073552 14.515317 13.055428 23.695968 17.515257 11.443045
##        22        23        24        25        26        27        28
## 14.406818 19.199973 13.145844 15.569546 15.551463 11.776371 13.479171
##        29        30        31        32        33        34        35
## 15.127721 13.740165 15.497214 11.740205 18.993229 16.200033 10.704059
##        36        37        38        39        40        41        42
## 14.497234 16.533360 15.515297 21.641759 20.254203 18.623736 13.424921
##        43        44        45        46        47        48        49
## 12.145864 14.145824 19.696048 14.163907  9.983156 14.163907 11.740205
##        50        51        52        53        54        55        56
## 18.975145 10.443065 17.308512  8.686016 13.352589 19.326555 13.794414
##        57        58        59        60        61        62        63
## 13.758248 16.830520 12.055448 11.740205 18.569486 17.920916 12.127781
##        64        65        66        67        68        69        70
## 18.587569 14.145824 20.623695 14.902893 16.200033 15.181970 17.497174
##        71        72        73        74        75        76        77
## 11.037385 14.884810 15.181970 20.308452 13.182010 12.794434 17.551423
##        78        79        80        81        82        83        84
##  9.965073 15.145804 11.280296 13.776331 18.587569 10.740225 13.884830
##        85        86        87        88        89        90        91
## 12.794434 24.398787 13.163927 18.623736 12.145864 14.812477 10.928886
##        92        93        94        95        96        97        98
## 12.515357 11.667872 13.479171 17.569506 17.533340 16.424861 19.254223
##        99       100       101       102       103       104       105
## 13.794414 20.236119 13.848664 16.218117 17.254263 18.884730 23.884629
##       106       107       108       109       110       111       112
## 16.163867 21.308432 14.776311 21.551343 12.740185 20.975105 15.181970
##       113       114       115       116       117       118       119
## 16.902853 18.308492 12.812517 17.605673 18.902813 11.388795 15.939039
##       120       121       122       123       124       125       126
## 14.424901 15.920956 16.181950 22.308412 14.830560 18.181910 15.939039
##       127       128       129       130       131       132       133
## 18.866647 15.091555 19.993208 12.145864 22.011252 16.145784 10.388815
##       134       135       136       137       138       139       140
## 15.848624 19.641799 18.920896 16.776271 16.163867 20.677945 19.218056
##       141       142       143       144       145       146       147
## 19.659882 11.109718 25.677845 30.416750 10.704059 19.218056 19.975125
##       148       149       150       151       152       153       154
## 13.794414 24.344538 14.551483 23.551303 17.920916 18.587569 14.884810
##       155       156       157       158       159       160       161
## 15.902873 18.272326 14.145824 16.957102 18.902813 15.533380 20.920856
##       162       163       164       165       166       167       168
## 18.920896 31.507146 16.939019 18.047498 17.902833 16.200033 16.939019
##       169       170       171       172       173       174       175
## 18.290409 10.280316 21.938919 13.073512 19.272306 22.830400 18.884730
##       176       177       178       179       180       181       182
## 26.677824 16.236200 25.119690 20.920856 10.388815 17.605673 17.884750
##       183       184       185       186       187       188       189
```

```
## 18.551403 13.848664 18.957062 17.884750 15.497214 20.623695 19.236139
##       190       191       192       193       194       195       196
## 11.794454 16.290449 18.975145 19.587549  7.262293 19.714131 26.732074
##       197       198       199       200       201       202       203
## 22.659822 19.515217 14.181990 16.605693 15.181970 20.236119 11.685955
##       204       205       206       207       208       209       210
## 20.326535 17.884750 20.326535 14.461068 16.902853 13.182010 19.362722
##       211       212       213       214       215       216       217
## 20.641779 17.236180 18.587569 22.975065 13.866747 16.515277 26.416830
##       218       219       220       221       222       223       224
## 13.145844 21.065521  8.631766 13.866747 15.812457 18.957062 18.551403
##       225       226       227       228       229       230       231
## 17.587589 17.272346 14.848644 17.884750 17.533340 28.732034 16.091534
##       232       233       234       235       236       237       238
## 19.902793 24.326455 16.551443 16.587609 16.587609 21.938919 17.920916
##       239       240       241       242       243       244       245
## 20.975105 20.938939 18.236160 24.029295 21.732174 20.083624 19.677965
##       246       247       248       249       250       251       252
## 21.011272 17.145764 21.290349 20.308452 15.515297 19.326555 30.858576
##       253       254       255       256       257       258       259
## 23.272225 22.163747 23.587469 21.011272 17.497174 25.380684 17.551423
##       260       261       262       263       264       265       266
## 19.605632 23.750217 15.569546 20.344618 22.641738 22.587489 23.677885
##       267       268       269       270       271       272       273
## 26.029254 21.272266 25.398767 16.200033 19.344638 19.290389 19.677965
##       274       275       276       277       278       279       280
## 17.605673 18.362742 17.533340 17.938999 15.461047 16.497194 15.848624
##       281       282       283       284       285       286       287
## 15.587630 22.695988 14.812477 20.659862 17.623756 15.551463 25.344518
##       288       289       290       291       292       293       294
## 15.533380 21.641759 15.551463 14.920976 16.236200 16.533360 17.587589
##       295       296       297       298       299       300       301
## 15.884790 20.641779 14.902893 13.497254 24.326455 19.344638 24.119710
```

```
lavPredict(fit.sem)
```

```
##
##   [1,]
##   [2,]
##   [3,]
##   [4,]
##   [5,]
##   [6,]
##   [7,]
##   [8,]
##   [9,]
##  [10,]
##  [11,]
##  [12,]
##  [13,]
```

```
##  [14,]
##  [15,]
##  [16,]
##  [17,]
##  [18,]
##  [19,]
##  [20,]
##  [21,]
##  [22,]
##  [23,]
##  [24,]
##  [25,]
##  [26,]
##  [27,]
##  [28,]
##  [29,]
##  [30,]
##  [31,]
##  [32,]
##  [33,]
##  [34,]
##  [35,]
##  [36,]
##  [37,]
##  [38,]
##  [39,]
##  [40,]
##  [41,]
##  [42,]
##  [43,]
##  [44,]
##  [45,]
##  [46,]
##  [47,]
##  [48,]
##  [49,]
##  [50,]
##  [51,]
##  [52,]
##  [53,]
##  [54,]
##  [55,]
##  [56,]
##  [57,]
##  [58,]
##  [59,]
##  [60,]
##  [61,]
##  [62,]
##  [63,]
```

```
##  [64,]
##  [65,]
##  [66,]
##  [67,]
##  [68,]
##  [69,]
##  [70,]
##  [71,]
##  [72,]
##  [73,]
##  [74,]
##  [75,]
##  [76,]
##  [77,]
##  [78,]
##  [79,]
##  [80,]
##  [81,]
##  [82,]
##  [83,]
##  [84,]
##  [85,]
##  [86,]
##  [87,]
##  [88,]
##  [89,]
##  [90,]
##  [91,]
##  [92,]
##  [93,]
##  [94,]
##  [95,]
##  [96,]
##  [97,]
##  [98,]
##  [99,]
## [100,]
## [101,]
## [102,]
## [103,]
## [104,]
## [105,]
## [106,]
## [107,]
## [108,]
## [109,]
## [110,]
## [111,]
## [112,]
## [113,]
```

```
## [114,]
## [115,]
## [116,]
## [117,]
## [118,]
## [119,]
## [120,]
## [121,]
## [122,]
## [123,]
## [124,]
## [125,]
## [126,]
## [127,]
## [128,]
## [129,]
## [130,]
## [131,]
## [132,]
## [133,]
## [134,]
## [135,]
## [136,]
## [137,]
## [138,]
## [139,]
## [140,]
## [141,]
## [142,]
## [143,]
## [144,]
## [145,]
## [146,]
## [147,]
## [148,]
## [149,]
## [150,]
## [151,]
## [152,]
## [153,]
## [154,]
## [155,]
## [156,]
## [157,]
## [158,]
## [159,]
## [160,]
## [161,]
## [162,]
## [163,]
```

```
## [164,]
## [165,]
## [166,]
## [167,]
## [168,]
## [169,]
## [170,]
## [171,]
## [172,]
## [173,]
## [174,]
## [175,]
## [176,]
## [177,]
## [178,]
## [179,]
## [180,]
## [181,]
## [182,]
## [183,]
## [184,]
## [185,]
## [186,]
## [187,]
## [188,]
## [189,]
## [190,]
## [191,]
## [192,]
## [193,]
## [194,]
## [195,]
## [196,]
## [197,]
## [198,]
## [199,]
## [200,]
## [201,]
## [202,]
## [203,]
## [204,]
## [205,]
## [206,]
## [207,]
## [208,]
## [209,]
## [210,]
## [211,]
## [212,]
## [213,]
```

```
## [214,]
## [215,]
## [216,]
## [217,]
## [218,]
## [219,]
## [220,]
## [221,]
## [222,]
## [223,]
## [224,]
## [225,]
## [226,]
## [227,]
## [228,]
## [229,]
## [230,]
## [231,]
## [232,]
## [233,]
## [234,]
## [235,]
## [236,]
## [237,]
## [238,]
## [239,]
## [240,]
## [241,]
## [242,]
## [243,]
## [244,]
## [245,]
## [246,]
## [247,]
## [248,]
## [249,]
## [250,]
## [251,]
## [252,]
## [253,]
## [254,]
## [255,]
## [256,]
## [257,]
## [258,]
## [259,]
## [260,]
## [261,]
## [262,]
## [263,]
```

```
## [264,]
## [265,]
## [266,]
## [267,]
## [268,]
## [269,]
## [270,]
## [271,]
## [272,]
## [273,]
## [274,]
## [275,]
## [276,]
## [277,]
## [278,]
## [279,]
## [280,]
## [281,]
## [282,]
## [283,]
## [284,]
## [285,]
## [286,]
## [287,]
## [288,]
## [289,]
## [290,]
## [291,]
## [292,]
## [293,]
## [294,]
## [295,]
## [296,]
## [297,]
## [298,]
## [299,]
## [300,]
## [301,]
```

Yes, like the residuals, the predicted values are also differenct for these two types of mdoels. There is no predicted values for fit.sem model.When the lavPredict(fit.sem) runs, it does not give any predicted values because the main purpose of the lavPredict() function is to predict estimated values for the latent variables in the model ('factor scores'). It does not predict future values of dependent variables as in the regression framework. Therefore, it looks like this type.

... and it is blank because there are no latent variables

Part 4: 10.5/10.5