HW8 - Simple Regression

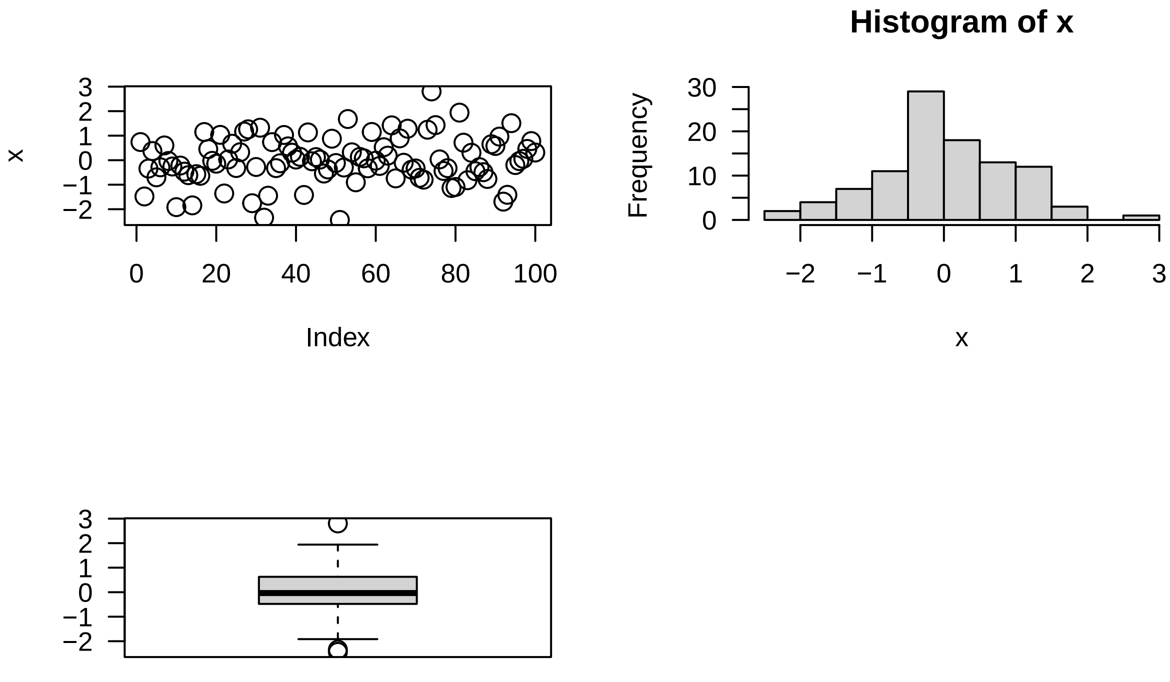
Madhu Peduri

July 5, 2021

0.0.1 1. Goodness of Fit

```r
x <- rnorm(100)
par(mfrow = c(2, 2))
plot(x, las = 1, cex = 1.5)
hist(x, las = 1, cex = 1.5)
boxplot(x, las = 1, cex = 1.5)
```

0.0.1.1 1.1 Generate a vector x containing 100 random numbers from a standard normal distribution and visualize the data:



```r
y1 <- 0 + 1 * x  ## A perfect linear association
y2 <- 0 + 1 * x + rnorm(length(x), mean = 0, sd = sqrt(1))  # Add a little N(0,1) noise (error)
```

0.0.1.2 1.2 To induce a linear relationship between x and a dependent variable y, generate the vector y using a linear transformation of the vector x:

0.0.1.3 1.3 Compute, y3,y4,y5,y6

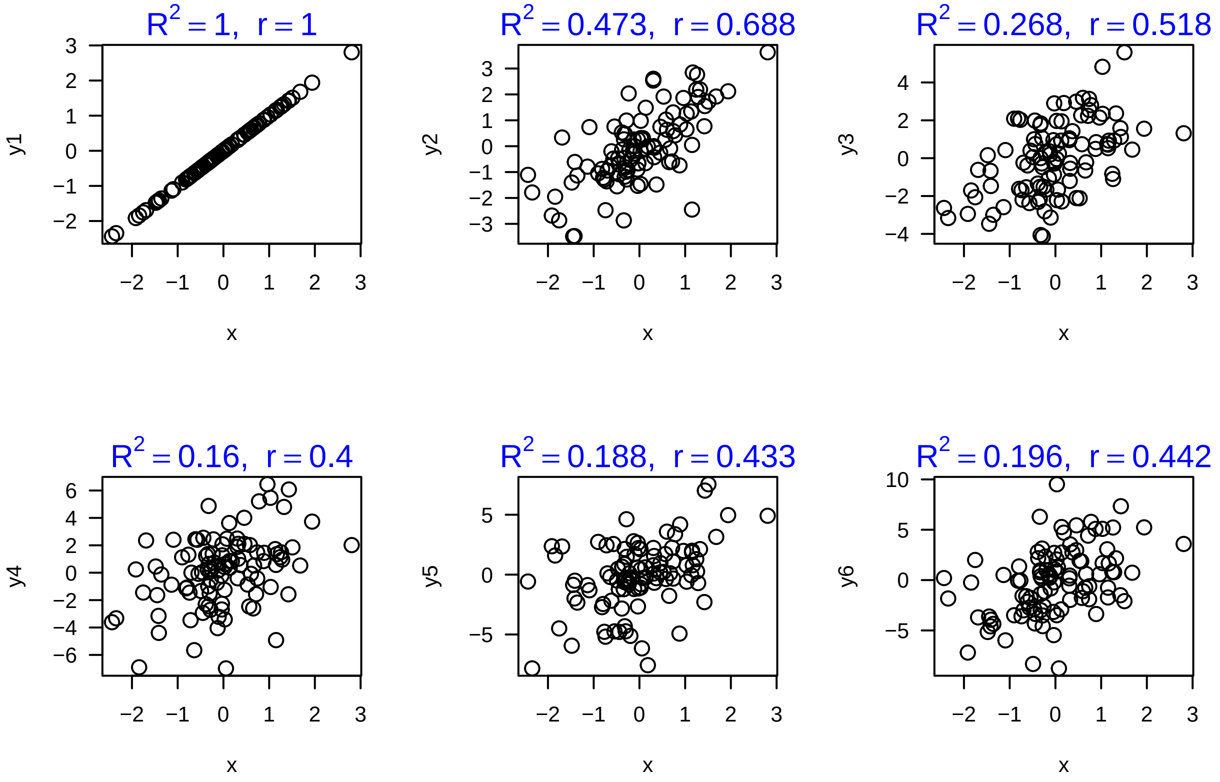
```r
y3 <- 0 + 1 * x + rnorm(length(x), mean = 0, sd = sqrt(3))
y4 <- 0 + 1 * x + rnorm(length(x), mean = 0, sd = sqrt(5))
y5 <- 0 + 1 * x + rnorm(length(x), mean = 0, sd = sqrt(7))
y6 <- 0 + 1 * x + rnorm(length(x), mean = 0, sd = sqrt(9))
```

0.0.1.4 $y_i = \beta_0 + \beta_1 x_i + \epsilon_i \; N(0, \sigma^2)$

```r
vlist <- list(y1, y2, y3, y4, y5, y6)
par(mfrow = c(2, 3))
nv <- c(1, 2, 3, 4, 5, 6)
rv <- c(1, 2, 3, 4, 5, 6)
sdv <- c(sqrt(0), sqrt(1), sqrt(3), sqrt(5), sqrt(7), sqrt(9))
R2v <- c(1, 2, 3, 4, 5, 6)
i = 0
for (yi in vlist) {
    i <- i + 1
    rv[i] <- cor(x, yi)
    R2v[i] <- summary(lm(yi ~ x))$r.squared
    plot(x, yi, las = 1, cex = 1.5, ylab = paste("y", nv[i],
        sep = ""))
    mtext(bquote(paste(R^2 == .(R2v[i]), ", ", ~r == .(rv[i]))),
        col = "blue")
}
```

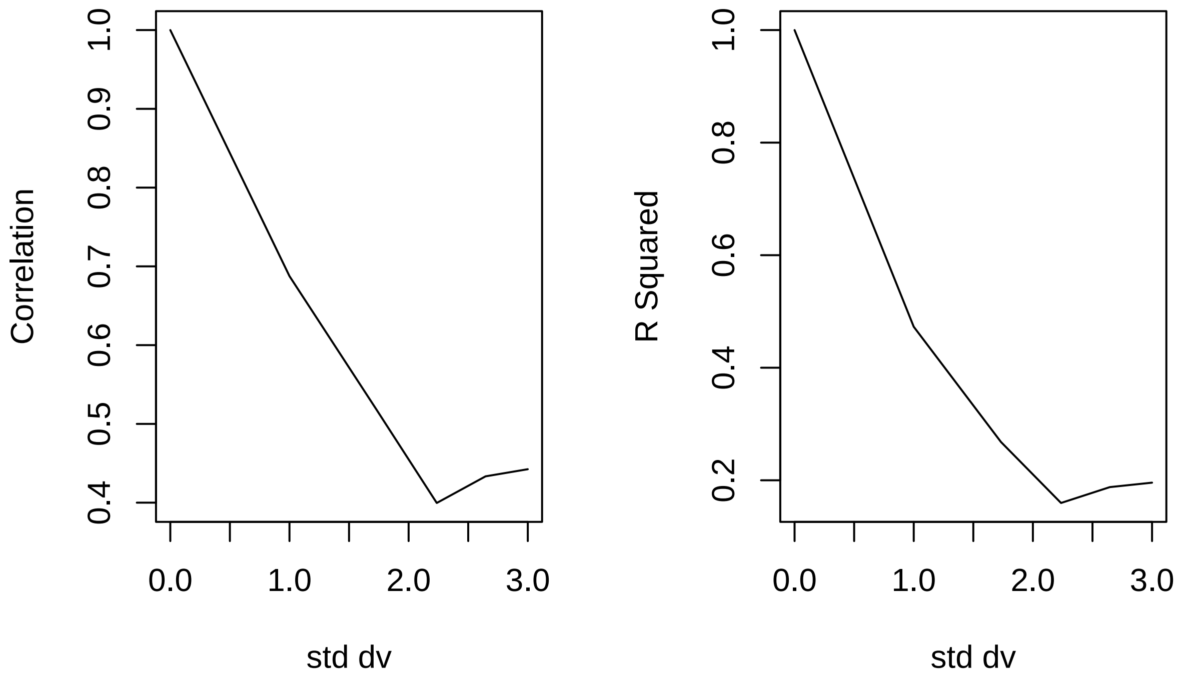
0.0.1.5 1.4 Generate scatterplots, r, and R2

```
## Warning in summary.lm(lm(yi ~ x)): essentially perfect fit: summary may be
## unreliable
```



```
par(mfrow = c(1, 2))
plot(sdv, rv, type = "l", xlab = "std dv", ylab = "Correlation")
plot(sdv, R2v, type = "l", xlab = "std dv", ylab = "R Squared")
```

0.0.1.6 1.5 Plot r and R2 versus standard deviation



0.0.1.7 Observation

- From above plots, we can say that as standard deviation of the noise increases, the coefficients of correlation and R-squared decreases.

0.0.2 2. Simple Linear Regression

```r
gscore <- read.csv("GPA.csv")
regmodel <- lm(gscore$GPA ~ gscore$ACT)
modelsumm <- summary(regmodel)
names(regmodel)
```

0.0.2.1 1. Write out the regression model that would explore the proposed relationship and state the model assumptions.

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

```r
names(modelsumm)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
modelsumm
```

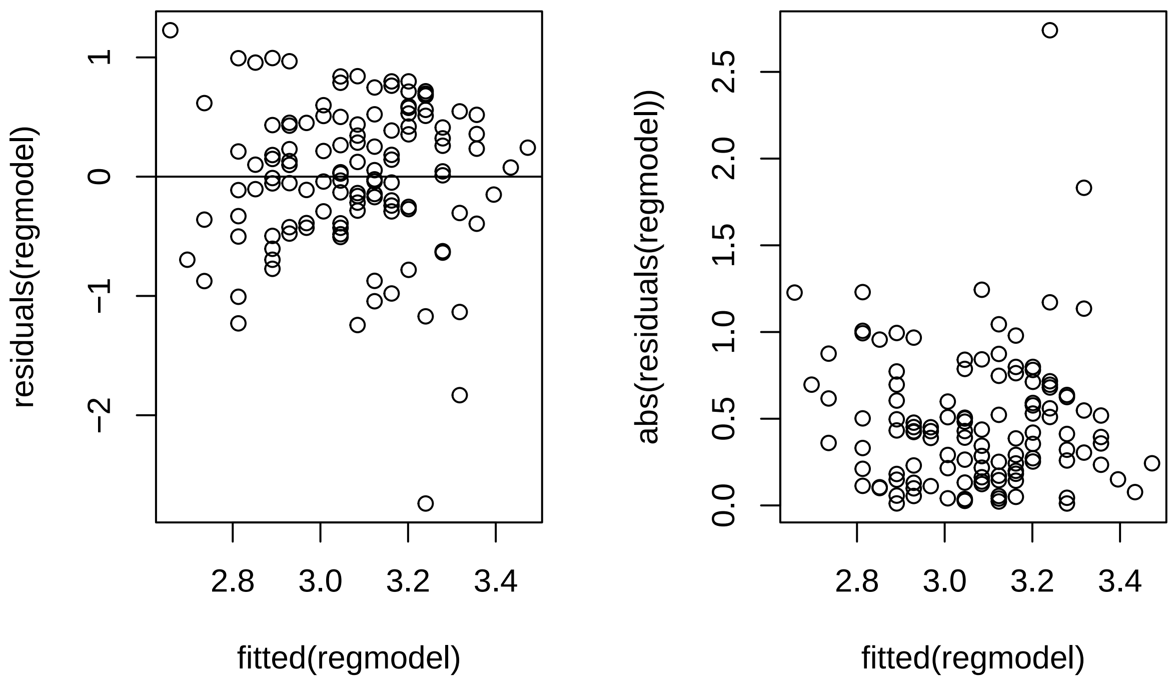
```
##
## Call:
## lm(formula = gscore$GPA ~ gscore$ACT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7400 -0.3383  0.0406  0.4406  1.2274
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1140     0.3209    6.59  1.3e-09 ***
## gscore$ACT    0.0388     0.0128    3.04   0.0029 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.623 on 118 degrees of freedom
## Multiple R-squared:  0.0726, Adjusted R-squared:  0.0648
## F-statistic: 9.24 on 1 and 118 DF,  p-value: 0.00292
```

0.0.2.2 Assumptions

- Homoscedasticity
- Normality
- Correlation

```
# Homoscedasticity
par(mfrow = c(1, 2))
plot(fitted(regmodel), residuals(regmodel))
abline(h = 0)
plot(fitted(regmodel), abs(residuals(regmodel)))
```

0.0.2.3 2. Testing Assumptions



```
summary(lm(abs(residuals(regmodel)) ~ fitted(regmodel)))
```

```
##
## Call:
## lm(formula = abs(residuals(regmodel)) ~ fitted(regmodel))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.464 -0.292 -0.057  0.210  2.267
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.49554    0.64949    0.76     0.45
## fitted(regmodel) -0.00696    0.21095   -0.03     0.97
##
## Residual standard error: 0.4 on 118 degrees of freedom
## Multiple R-squared:  9.23e-06,   Adjusted R-squared:  -0.00847
## F-statistic: 0.00109 on 1 and 118 DF,  p-value: 0.974
```

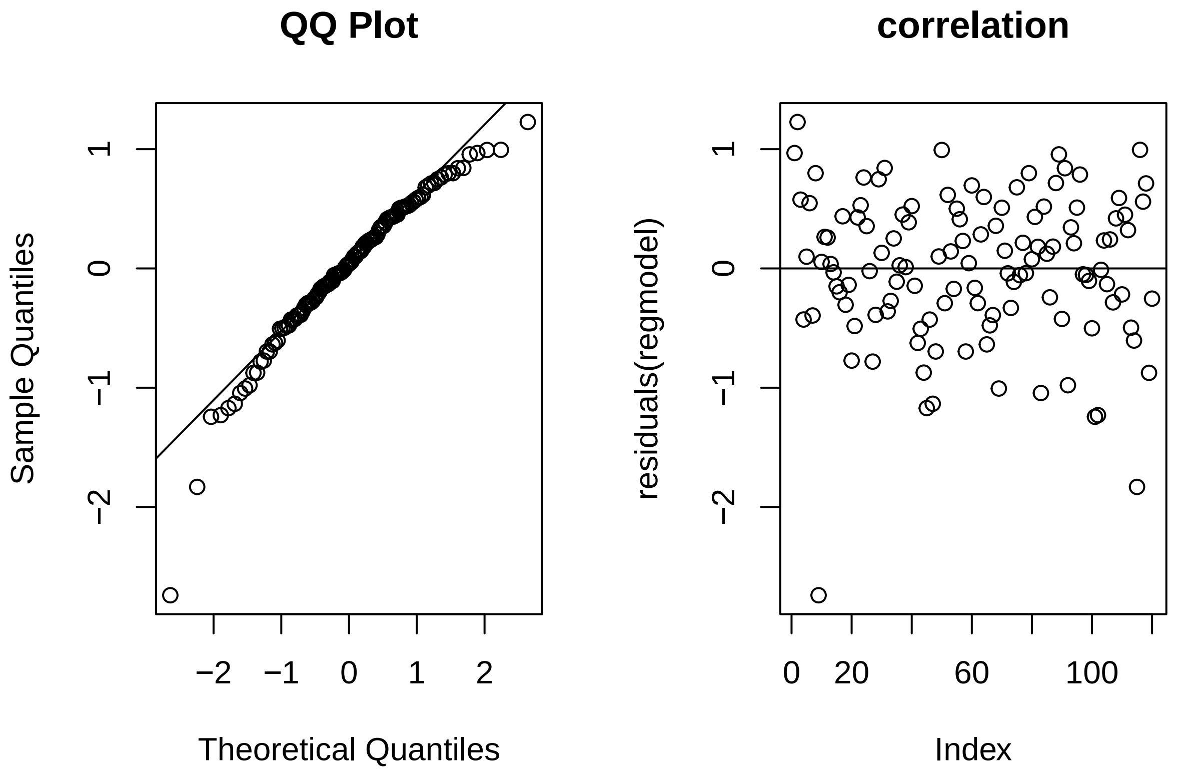
```
ncvTest(regmodel)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.639, Df = 1, p = 0.4
```

```
# Normality
qqnorm(residuals(regmodel), main = "QQ Plot")
qqline(residuals(regmodel))
# ggqqplot(residuals(regmodel))
shapiro.test(residuals(regmodel))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(regmodel)
## W = 1, p-value = 3e-04
```

```
# Correlation
plot(residuals(regmodel), main = "correlation")
abline(h = 0)
```



```
# summary(lm(residuals(regmodel)[-1] ~
# -1+residuals(regmodel)[-541]))
dwtest(regmodel)
```

```
##
##  Durbin-Watson test
##
## data:  regmodel
## DW = 2, p-value = 0.2
## alternative hypothesis: true autocorrelation is greater than 0
```

0.0.2.4 Observations Homoscedasticity - By observing the plot between fitted and residuals, they look like non-constant variance plots. P-value $= 0.4$ (0.05) from ncvTest suggest non-constant variance.

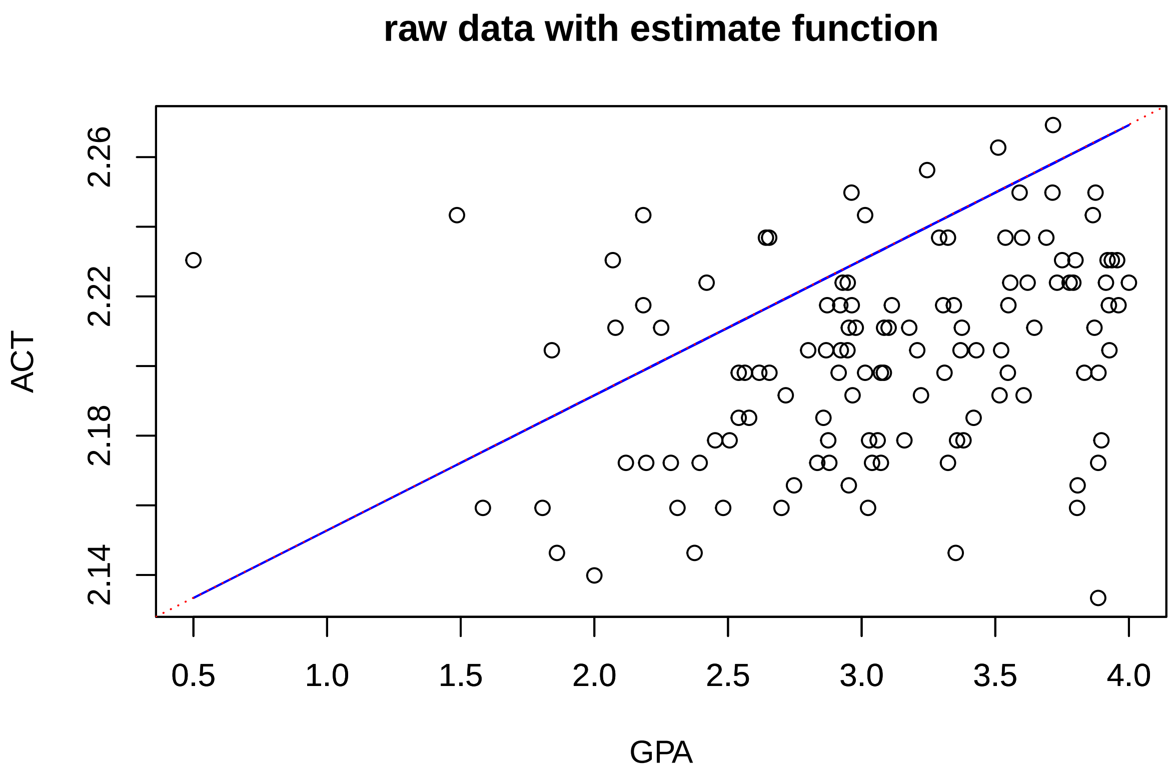
Normality - By observing the plots of qqnorm, we can say that all points are along the reference lines and that suggests the normality. However, Shapiro-wilk test gives a p-value < 0.05 significant value. This shows non-normality nature of the distribution of residuals. Normality is the least worrisome assumption.

Correlation - By observing the correlation plot, we can say th residuals are uncorrelated. dwtest has p-value 0.2 (>0.05) greater than significant value also suggest uncorrelation

0.0.2.5 3. Intercepts Intercept: $\beta_0 = 2.1140$
Y-intercept: $\beta_1 = 0.0388$
Regression function: $GPA = 2.1140 + 0.0388 * ACT$

```r
gscore$predict <- lapply(gscore$GPA, function(x) {
    2.114 + 0.0388 * x
})
plot(gscore$GPA, gscore$ACT, xlab = "GPA", ylab = "ACT", yaxt = "n")
par(new = TRUE)
plot(gscore$GPA, gscore$predict, type = "l", col = "blue", xlab = "",
    ylab = "")
abline(regmodel, col = "red", lty = 3)
title("raw data with estimate function")
```

0.0.2.6 4. Plot raw data



0.0.2.7 5. Percentage of variation

For every 1 mark increase in ACT score, the GPA function increases by 0.04%.

```r
confint(regmodel)
```

0.0.2.8 6. Confidence interval

```
##              2.5 % 97.5 %
## (Intercept) 1.4786 2.7495
## gscore$ACT  0.0135 0.0641
```

0.0.2.9 Observations

- 95% confidence interval for $\beta_1 = 0.04$
- (95% CI: 0.04, 0.01)
- From the confidence interval (2.5 to 97.5) above, we can say zero is not included in it. This says that there is evidence of a linear relationship between predictor GPA and response ACT in the sample.

0.0.2.10 7. Test linear association

- Null hypothesis of linear regression $H_0 : \beta 1 = 0$ that says, that predictor has not effect on the output.
- Alternate hypothesis $H_1 : \beta_1 \neq 0$ that says, there is a linear relation between predictor and the output.
- We can see from the confidence interval do not include zero. That implies at the evidence of linear relation ship between Predictor GPA and output ACT score.

0.0.2.11 8. p-value of linear regression

- We can see from the summary of regression model, that p-value = 0.00292. This is less than the significant value 0.01. Using this we can say that alternate hypothesis is true and shows evidence of linear relationship between GPA and ACT score.

0.1 Document Information.

All of the statistical analyses in this document will be performed using R version 4.1.0 (2021-05-18). R packages used will be maintained using the packrat dependency management system.

```r
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
```

```
##  [1] lmtest_0.9-38      zoo_1.8-9          PairedData_1.1.1   mvtnorm_1.1-2     
##  [5] gld_2.6.2          ggpubr_0.4.0       car_3.0-11         carData_3.0-4     
##  [9] mnormt_2.0.2       vcd_1.4-8          epiDisplay_3.5.0.1 nnet_7.3-16       
## [13] foreign_0.8-81     Hmisc_4.5-0        Formula_1.2-4      survival_3.2-11   
## [17] lattice_0.20-44    MASS_7.3-54        ggplot2_3.3.5      rmarkdown_2.8     
## [21] knitr_1.33        
## 
## loaded via a namespace (and not attached):
##  [1] tidyr_1.1.3         splines_4.1.0       tmvnsim_1.0-2      
##  [4] highr_0.9           lmom_2.8            latticeExtra_0.6-29
##  [7] cellranger_1.1.0    yaml_2.2.1          pillar_1.6.1       
## [10] backports_1.2.1     glue_1.4.2          digest_0.6.27      
## [13] RColorBrewer_1.1-2  ggsignif_0.6.2      checkmate_2.0.0    
## [16] colorspace_2.0-1    htmltools_0.5.1.1   Matrix_1.3-3       
## [19] pkgconfig_2.0.3     broom_0.7.8         haven_2.4.1        
## [22] purrr_0.3.4         scales_1.1.1        jpeg_0.1-8.1       
## [25] openxlsx_4.2.4      rio_0.5.27          proxy_0.4-26       
## [28] htmlTable_2.2.1     tibble_3.1.2        generics_0.1.0     
## [31] ellipsis_0.3.2      withr_2.4.2         magrittr_2.0.1     
## [34] crayon_1.4.1        readxl_1.3.1        evaluate_0.14      
## [37] fansi_0.5.0         class_7.3-19        rstatix_0.7.0      
## [40] forcats_0.5.1       tools_4.1.0         data.table_1.14.0  
## [43] hms_1.1.0           formatR_1.11        lifecycle_1.0.0    
## [46] stringr_1.4.0       munsell_0.5.0       cluster_2.1.2      
## [49] zip_2.2.0           e1071_1.7-7         compiler_4.1.0     
## [52] rlang_0.4.11        rstudioapi_0.13     htmlwidgets_1.5.3  
## [55] base64enc_0.1-3     gtable_0.3.0        abind_1.4-5        
## [58] curl_4.3.1          R6_2.5.0            gridExtra_2.3      
## [61] dplyr_1.0.7         utf8_1.2.1          stringi_1.6.1      
## [64] Rcpp_1.0.6          vctrs_0.3.8         rpart_4.1-15       
## [67] png_0.1-7           tidyselect_1.1.1    xfun_0.23
```