STAT\_615\_HW\_02

Sunday Okechukwu

2023-02-23

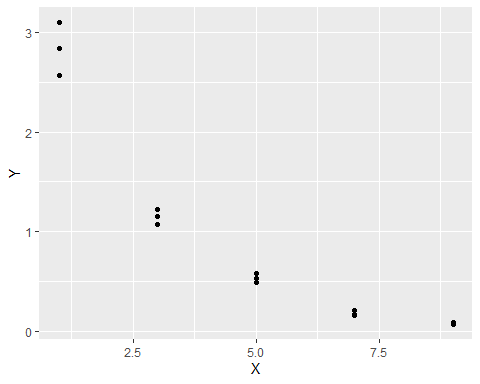
1. Page 150 problem 3.15 (a, b) Found in the online text book.

# Get the data   
X <- c(9.0,9.0,9.0,7.0,7.0,7.0,5.0,5.0,5.0,3.0,3.0,3.0,1.0,1.0,1.0)  
  
Y <- c(0.07,0.09,0.08,0.16,0.17,0.21,0.49,0.58,0.53,1.22,1.15,1.07,2.84,2.57,3.10)  
  
data\_1 <- as\_tibble(data.frame(X, Y))  
data\_1

## # A tibble: 15 × 2  
## X Y  
## <dbl> <dbl>  
## 1 9 0.07  
## 2 9 0.09  
## 3 9 0.08  
## 4 7 0.16  
## 5 7 0.17  
## 6 7 0.21  
## 7 5 0.49  
## 8 5 0.58  
## 9 5 0.53  
## 10 3 1.22  
## 11 3 1.15  
## 12 3 1.07  
## 13 1 2.84  
## 14 1 2.57  
## 15 1 3.1

qplot(X, Y, data = data\_1)

## Warning: `qplot()` was deprecated in ggplot2 3.4.0.



1a. Fit a linear regression function

model1 <- lm(Y ~ X)  
model1

##   
## Call:  
## lm(formula = Y ~ X)  
##   
## Coefficients:  
## (Intercept) X   
## 2.575 -0.324

summary(model1)

##   
## Call:  
## lm(formula = Y ~ X)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5333 -0.4043 -0.1373 0.4157 0.8487   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.5753 0.2487 10.354 1.20e-07 \*\*\*  
## X -0.3240 0.0433 -7.483 4.61e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4743 on 13 degrees of freedom  
## Multiple R-squared: 0.8116, Adjusted R-squared: 0.7971   
## F-statistic: 55.99 on 1 and 13 DF, p-value: 4.611e-06

1b. Perform the F test to determine whether or not there is lack of fit of a linear regression function; use alpha =.025. State the alternatives, decision rule, and conclusion. The hypothesis being tested is:

anova(model1)

## Analysis of Variance Table  
##   
## Response: Y  
## Df Sum Sq Mean Sq F value Pr(>F)   
## X 1 12.5971 12.597 55.994 4.611e-06 \*\*\*  
## Residuals 13 2.9247 0.225   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Answer

H0: β1 = 0

H1: β1 ≠ 0

The p-value from the output is 4.611e-06

Since the p-value (4.611e-06) is less than the significance level (0.025), we can reject the null hypothesis.

Therefore, we can conclude that the slope is significant and not equal to zero.

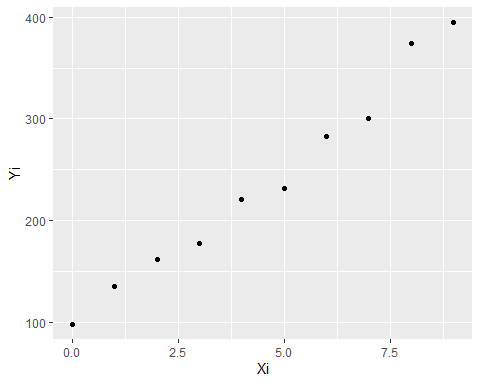
1. Page 151 problem 3.17 ( a – e); Found in the online text book.

Xi <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)  
Yi <- c(98, 135, 162, 178, 221, 232, 283, 300, 374, 395)  
  
data\_2 <- as\_tibble(data.frame(Xi, Yi))  
data\_2

## # A tibble: 10 × 2  
## Xi Yi  
## <dbl> <dbl>  
## 1 0 98  
## 2 1 135  
## 3 2 162  
## 4 3 178  
## 5 4 221  
## 6 5 232  
## 7 6 283  
## 8 7 300  
## 9 8 374  
## 10 9 395

1. Prepare a scatter plot of the data. Does a linear relation appear adequate here?

qplot(Xi, Yi, data = data\_2)

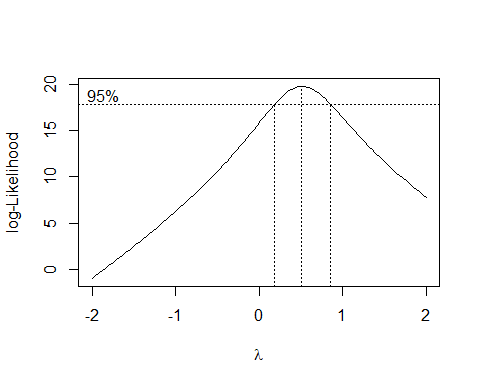


# Answer

Yes, a linear relation appear adequate

1. Use the Box-Cox procedure and standardization (3.36) to find an appropriate power transformation of Y. Evaluate SSE for A = 0.3, 0.4, 0.5, 0.6, 0.7. What transformation of Y is suggested?

box\_cox <- boxcox(Yi ~ Xi)



box\_cox

## $x  
## [1] -2.00000000 -1.95959596 -1.91919192 -1.87878788 -1.83838384 -1.79797980  
## [7] -1.75757576 -1.71717172 -1.67676768 -1.63636364 -1.59595960 -1.55555556  
## [13] -1.51515152 -1.47474747 -1.43434343 -1.39393939 -1.35353535 -1.31313131  
## [19] -1.27272727 -1.23232323 -1.19191919 -1.15151515 -1.11111111 -1.07070707  
## [25] -1.03030303 -0.98989899 -0.94949495 -0.90909091 -0.86868687 -0.82828283  
## [31] -0.78787879 -0.74747475 -0.70707071 -0.66666667 -0.62626263 -0.58585859  
## [37] -0.54545455 -0.50505051 -0.46464646 -0.42424242 -0.38383838 -0.34343434  
## [43] -0.30303030 -0.26262626 -0.22222222 -0.18181818 -0.14141414 -0.10101010  
## [49] -0.06060606 -0.02020202 0.02020202 0.06060606 0.10101010 0.14141414  
## [55] 0.18181818 0.22222222 0.26262626 0.30303030 0.34343434 0.38383838  
## [61] 0.42424242 0.46464646 0.50505051 0.54545455 0.58585859 0.62626263  
## [67] 0.66666667 0.70707071 0.74747475 0.78787879 0.82828283 0.86868687  
## [73] 0.90909091 0.94949495 0.98989899 1.03030303 1.07070707 1.11111111  
## [79] 1.15151515 1.19191919 1.23232323 1.27272727 1.31313131 1.35353535  
## [85] 1.39393939 1.43434343 1.47474747 1.51515152 1.55555556 1.59595960  
## [91] 1.63636364 1.67676768 1.71717172 1.75757576 1.79797980 1.83838384  
## [97] 1.87878788 1.91919192 1.95959596 2.00000000  
##   
## $y  
## [1] -0.9658193 -0.6937913 -0.4207767 -0.1467073 0.1284841 0.4048636  
## [7] 0.6824993 0.9614671 1.2418449 1.5237154 1.8071653 2.0922839  
## [13] 2.3791682 2.6679172 2.9586366 3.2514375 3.5464347 3.8437523  
## [19] 4.1435167 4.4458635 4.7509344 5.0588759 5.3698461 5.6840053  
## [25] 6.0015257 6.3225855 6.6473693 6.9760735 7.3088975 7.6460518  
## [31] 7.9877526 8.3342209 8.6856860 9.0423782 9.4045296 9.7723734  
## [37] 10.1461352 10.5260299 10.9122647 11.3050059 11.7044036 12.1105484  
## [43] 12.5234532 12.9430874 13.3692272 13.8015801 14.2396062 14.6824841  
## [49] 15.1292255 15.5781882 16.0275207 16.4746060 16.9160610 17.3480721  
## [55] 17.7653492 18.1622014 18.5318600 18.8666105 19.1587042 19.4003849  
## [61] 19.5840436 19.7042926 19.7574566 19.7414832 19.6586825 19.5123180  
## [67] 19.3082046 19.0538784 18.7569929 18.4254045 18.0668841 17.6880092  
## [73] 17.2946495 16.8919915 16.4838201 16.0736281 15.6638914 15.2565584  
## [79] 14.8532307 14.4548895 14.0624002 13.6762592 13.2967958 12.9242249  
## [85] 12.5585814 12.1998685 11.8479894 11.5028080 11.1641606 10.8318458  
## [91] 10.5056567 10.1853718 9.8707647 9.5616090 9.2576780 8.9587477  
## [97] 8.6646064 8.3750456 8.0898555 7.8088241

lambda <- box\_cox$x[which.max(box\_cox$y)]   
lambda

## [1] 0.5050505

SSE code

lambda\_seq <- seq(0.3, 0.7, by = 0.1)  
  
for (i in seq\_along(lambda\_seq)) {  
 λ <- lambda\_seq[i]  
 p <- (1/(λ \*(prod(Yi)^(1/length(Yi)))^(λ - 1)))\*(Yi^λ-1)  
   
 sse\_i <- anova(lm(p~Xi))  
 sse <- sse\_i$`Sum Sq`[2]  
   
 print(str\_glue("SSE for λ={λ}: {sse}"))  
}

## SSE for λ=0.3: 1099.70927132159  
## SSE for λ=0.4: 967.908780410858  
## SSE for λ=0.5: 916.404798150164  
## SSE for λ=0.6: 942.44976495254  
## SSE for λ=0.7: 1044.2384001476

The suggested transformation is when λ = 0.5050505. This makes sense since SSE has the least value when lambda = 0.5.

2c. Use the transformation Y′=sqr(Y) and obtain the estimated linear regression function for the transformed data.

Yi <- sqrt(Yi)  
Yi.lm <- lm(Yi ~ Xi)   
Yi.lm

##   
## Call:  
## lm(formula = Yi ~ Xi)  
##   
## Coefficients:  
## (Intercept) Xi   
## 10.261 1.076

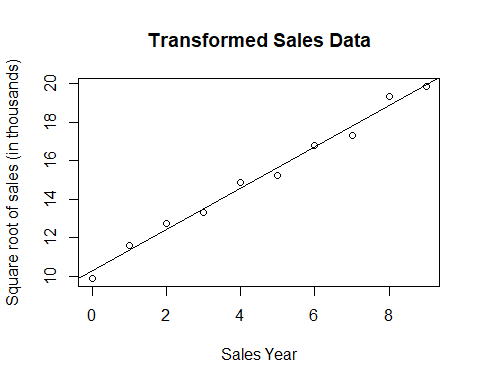
Yhat =10.26093 + 1.076X

resid <- residuals(Yi.lm)  
resid

## 1 2 3 4 5 6   
## -0.36143656 0.28172678 0.31440703 -0.14814273 0.29997018 -0.41084412   
## 7 8 9 10   
## 0.10392174 -0.47446579 0.46781397 -0.07295049

2d. Plot the estimated regression line and the transformed data. Does the regression line appear to be a good fit to the transformed data?

plot(Xi, Yi, xlab = "Sales Year", ylab = "Square root of sales (in thousands)", main = "Transformed Sales Data")  
abline(Yi.lm)

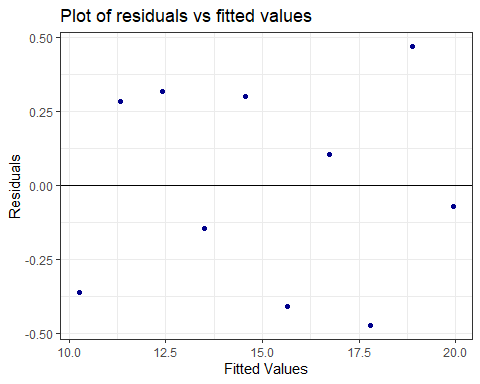


# Answer

The regression line appears to be a good fit for the transformed data.

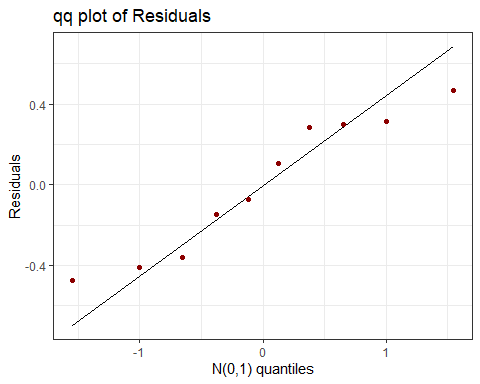
2e. Obtain the residuals and plot them against the fitted values. Also prepare a normal probability plot. What do your plots show?

ggplot(Yi.lm, aes(x = .fitted, y = .resid)) +   
 geom\_point(color = "dark blue") +  
 geom\_hline(yintercept = 0) +  
 xlab("Fitted Values") +  
 ylab("Residuals") +  
 ggtitle("Plot of residuals vs fitted values")+  
 theme\_bw()



Points to the error in the linear regression lining up well with the differences between the expected values and the observed values. Additionally, the sum of the residuals is zero which supports the use of this transformation on the data for linear regression analysis.

#qq plot of residuals   
ggplot(Yi.lm, mapping = aes(sample = Yi.lm$residuals)) +   
 stat\_qq() +  
 stat\_qq\_line() +  
 geom\_qq( color = "dark red") +  
 ggtitle("qq plot of Residuals") +  
 ylab("Residuals") +  
 xlab("N(0,1) quantiles") +  
 theme\_bw()



The Q-Q plot helps us determine if the standardized residuals from the linear model are normally distributed. The points do not perfectly line up along the line y=x; however they appear to generally follow this line; therefore, we conclude that the residuals are normally distributed.

1. Page 201 problem 5.1; Found in the online text book.

#Matrix definitions  
  
A <- matrix(c(1, 4,  
 2, 6,  
 3, 8) , ncol = 2, byrow = TRUE)  
  
B <- matrix(c(1, 3,  
 1, 4,  
 2, 5) , ncol = 2, byrow = TRUE)  
  
C <- matrix(c(3, 8, 1,  
 5, 4, 0), ncol = 3, byrow = TRUE)

A + B

A + B

## [,1] [,2]  
## [1,] 2 7  
## [2,] 3 10  
## [3,] 5 13

#dimension is 3 \* 2

A - B

A - B

## [,1] [,2]  
## [1,] 0 1  
## [2,] 1 2  
## [3,] 1 3

#dimension is 3 \* 2

AC

A %\*% C

## [,1] [,2] [,3]  
## [1,] 23 24 1  
## [2,] 36 40 2  
## [3,] 49 56 3

#dimension is 3 \* 3

AB^-1

A %\*% t(B)

## [,1] [,2] [,3]  
## [1,] 13 17 22  
## [2,] 20 26 34  
## [3,] 27 35 46

#dimension is 3 \* 3

B^-1(A)

t(B) %\*% A

## [,1] [,2]  
## [1,] 9 26  
## [2,] 26 76

#dimension is 2 \* 2

1. Call the mtcars data set and produce a linear model that uses hp (horse power) to predict mpg (miles per gallon).Using techniques and R code demonstrated in class, produce a prediction interval, and a confidence interval for the response variable mpg at a fixed mpg value of .21. (follow the coding sequence closely of the examples shown in class.)

data("mtcars")  
mtcars

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

mpg <- mtcars$mpg  
hp <- mtcars$hp  
fit <- lm(mpg ~ hp)  
fit

##   
## Call:  
## lm(formula = mpg ~ hp)  
##   
## Coefficients:  
## (Intercept) hp   
## 30.09886 -0.06823

Let’s find the confidence interval

new\_data <- lm(mpg ~ hp, data = mtcars)  
  
new\_df <- data.frame(hp = 62)  
  
  
predict(object = new\_data, newdata = new\_df, interval = "confidence") %>%   
cbind(new\_df)

## fit lwr upr hp  
## 1 25.86871 23.63082 28.10659 62

Let’s find the prediction interval

new\_data <- lm(mpg ~ hp, data = mtcars)  
  
new\_df <- data.frame(hp = 62)  
  
  
predict(object = new\_data, newdata = new\_df, interval = "prediction") %>%   
cbind(new\_df)

## fit lwr upr hp  
## 1 25.86871 17.66822 34.06919 62