STAT\_615\_Lab\_04

Sunday Okechukwu

2023-02-07

# Question

1. Page 152 problem 3.24 a, b

Xi <- c(5, 8, 11, 7, 13, 12, 12, 6)  
Yi <- c(63, 67, 74, 64, 75, 69, 90, 60)  
  
  
my\_data <- as.tibble(data.frame(Xi, Yi))

## Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
## ℹ Please use `as\_tibble()` instead.  
## ℹ The signature and semantics have changed, see `?as\_tibble`.

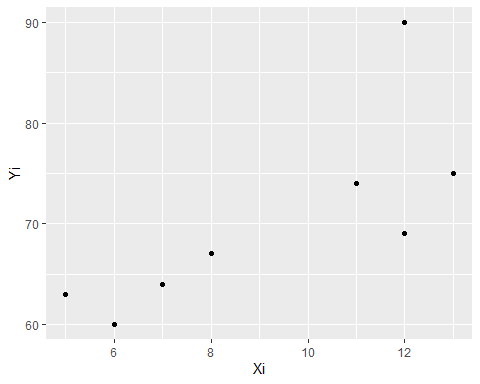
my\_data

## # A tibble: 8 × 2  
## Xi Yi  
## <dbl> <dbl>  
## 1 5 63  
## 2 8 67  
## 3 11 74  
## 4 7 64  
## 5 13 75  
## 6 12 69  
## 7 12 90  
## 8 6 60

model <- lm(Yi ~ Xi)  
summary(model)

##   
## Call:  
## lm(formula = Yi ~ Xi)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.6667 -3.0000 -0.6667 0.4167 13.3333   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 48.6667 7.8869 6.171 0.000832 \*\*\*  
## Xi 2.3333 0.8135 2.868 0.028487 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.683 on 6 degrees of freedom  
## Multiple R-squared: 0.5783, Adjusted R-squared: 0.508   
## F-statistic: 8.228 on 1 and 6 DF, p-value: 0.02849

ggplot(data = my\_data) +  
 geom\_point(mapping = aes(x = Xi , y = Yi))



my\_data%>%  
 mutate(yhat = 48.6667 + 2.3333\*Xi)%>%  
 mutate(residuals = Yi - (48.6667 + 2.3333\*Xi))-> my\_data1  
my\_data1

## # A tibble: 8 × 4  
## Xi Yi yhat residuals  
## <dbl> <dbl> <dbl> <dbl>  
## 1 5 63 60.3 2.67   
## 2 8 67 67.3 -0.333  
## 3 11 74 74.3 -0.333  
## 4 7 64 65.0 -1.00   
## 5 13 75 79.0 -4.00   
## 6 12 69 76.7 -7.67   
## 7 12 90 76.7 13.3   
## 8 6 60 62.7 -2.67

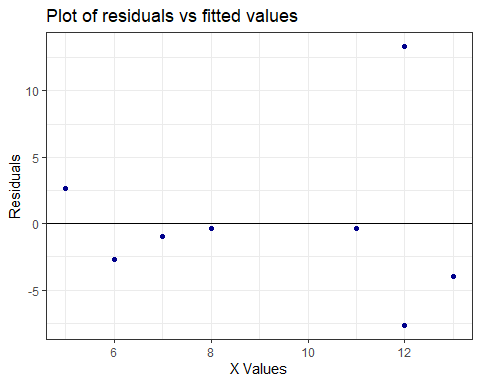
#Sum of the residuals is approx. zero  
sum(my\_data1$residuals)

## [1] 0.0022

# Question

* What does your residual plot show?

#Plot of residuals vs X values  
ggplot(my\_data1, aes(x = Xi, y = residuals)) +   
 geom\_point(color = "dark blue") +  
 geom\_hline(yintercept = 0) +  
 xlab("X Values") +  
 ylab("Residuals") +  
 ggtitle("Plot of residuals vs fitted values")+  
 theme\_bw()



# Answer

The residual plot shows the presence of an oulier. From the residual plot we can see that the residuals are not close to zero or the average value of the residuals will not be zero.

# Question

Omit case 7 from the data and obtain the estimated regression function based on the remaining seven cases. Compare this estimated regression function to that obtained in part (a). What can you conclude about the effect of case 7?

Xi <- c(5, 8, 11, 7, 13, 12, 6)  
Yi <- c(63, 67, 74, 64, 75, 69, 60)  
  
  
my\_data2 <- as.tibble(data.frame(Xi, Yi))  
my\_data2

## # A tibble: 7 × 2  
## Xi Yi  
## <dbl> <dbl>  
## 1 5 63  
## 2 8 67  
## 3 11 74  
## 4 7 64  
## 5 13 75  
## 6 12 69  
## 7 6 60

model2 <- lm(Yi ~ Xi)  
summary(model2)

##   
## Call:  
## lm(formula = Yi ~ Xi)  
##   
## Residuals:  
## 1 2 3 4 5 6 7   
## 1.8252 0.9612 3.0971 -0.4175 0.8544 -3.5243 -2.7961   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 53.0680 3.2136 16.514 1.49e-05 \*\*\*  
## Xi 1.6214 0.3448 4.702 0.00533 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.645 on 5 degrees of freedom  
## Multiple R-squared: 0.8156, Adjusted R-squared: 0.7787   
## F-statistic: 22.11 on 1 and 5 DF, p-value: 0.005327

# Answer

Thus we can see that the R-squared value in this case is 81.56% which means that the fitted regression equation is a better fit of the given data after deleting the 7th observation.

It is definitely much better than the R-Squared value obtained in part (a).

Thus the 7th observation acted as an outlier and influenced the regression equation and made it a bad fit in part(a).

1. Using your fitted regression function in part (b), obtain a 99 percent prediction interval for a new Y observation at X = 12. Does observation Y7 fall outside this prediction interval? What is the significance of this?

new\_data <- lm(Yi ~ Xi, data = my\_data2)  
  
new\_df <- data.frame(Xi = 12)  
predict(object = new\_data, newdata = new\_df, interval = "prediction") %>%   
cbind(new\_df)

## fit lwr upr Xi  
## 1 72.52427 64.73909 80.30945 12

We can see that the observation Y7=90 is falling outside the 99% P.I. Thus which means that Y7 is an outlier.

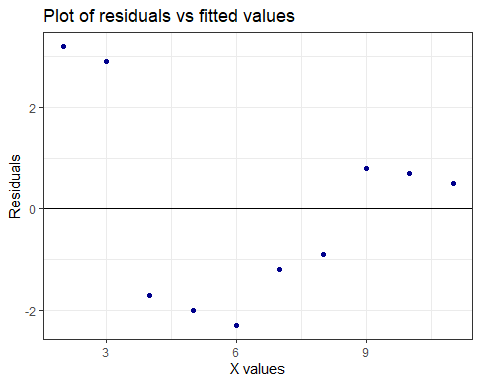
# Question

1. Plot the residuals ei against Xi. What problem appears to be present here? Might a transformation alleviate this problem?

xi <- c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)  
ei <- c(3.2, 2.9, -1.7, -2.0, -2.3, -1.2,-0.9, 0.8, 0.7, 0.5)  
data1 <- data.frame(xi, ei)  
data1

## xi ei  
## 1 2 3.2  
## 2 3 2.9  
## 3 4 -1.7  
## 4 5 -2.0  
## 5 6 -2.3  
## 6 7 -1.2  
## 7 8 -0.9  
## 8 9 0.8  
## 9 10 0.7  
## 10 11 0.5

ggplot(data1, aes(x = xi, y = ei)) +   
 geom\_point(color = "dark blue") +  
 geom\_hline(yintercept = 0) +  
 xlab("X values") +  
 ylab("Residuals") +  
 ggtitle("Plot of residuals vs fitted values")+  
 theme\_bw()



# Answer

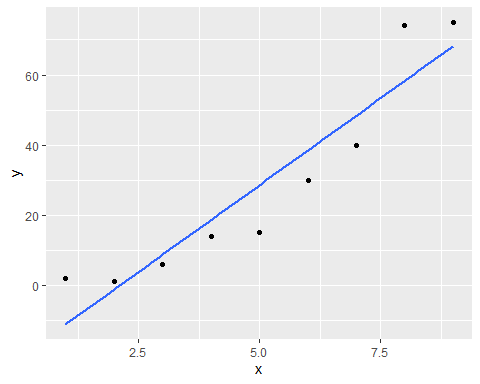
There is a clear pattern to the distribution of the residuals, which are positive for more extreme values of X and negative for values closer to the median. A nonlinear transformation such as quadratic or logarithmic might alleviate the problem. Also it seems that the residuals have a heteroscedasticity problem. The absolute value of residuals decreases as X increases. If this is the case, a nonlinear transformation might not be helpful. Instead we may use weighted least squares (WLS) to fix the problem.

x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)  
y <- c(2, 1, 6, 14, 15, 30, 40, 74, 75)  
data3 <- data.frame(x,y)  
data3

## x y  
## 1 1 2  
## 2 2 1  
## 3 3 6  
## 4 4 14  
## 5 5 15  
## 6 6 30  
## 7 7 40  
## 8 8 74  
## 9 9 75

ggplot(data = data3) +  
 geom\_point(mapping = aes(x = x , y = y)) +  
 geom\_smooth(method = lm,mapping = aes(x = x , y = y), se=FALSE)

## `geom\_smooth()` using formula = 'y ~ x'



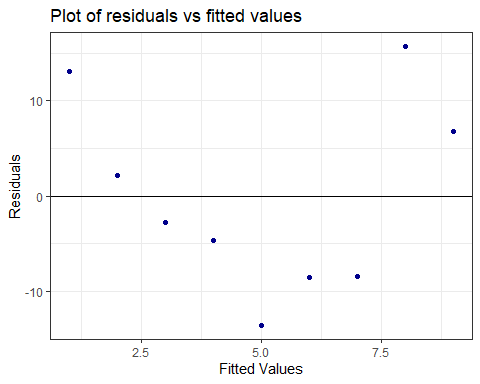
model3 <- lm(y ~ x)  
summary(model3)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.556 -8.389 -2.722 6.778 15.694   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -21.028 7.881 -2.668 0.032087 \*   
## x 9.917 1.401 7.081 0.000197 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.85 on 7 degrees of freedom  
## Multiple R-squared: 0.8775, Adjusted R-squared: 0.86   
## F-statistic: 50.14 on 1 and 7 DF, p-value: 0.000197

data3%>%  
 mutate(yhat = -21.028 + 9.917\*x)%>%  
 mutate(residuals = y - (-21.028 + 9.917\*x))-> my\_data4  
my\_data4

## x y yhat residuals  
## 1 1 2 -11.111 13.111  
## 2 2 1 -1.194 2.194  
## 3 3 6 8.723 -2.723  
## 4 4 14 18.640 -4.640  
## 5 5 15 28.557 -13.557  
## 6 6 30 38.474 -8.474  
## 7 7 40 48.391 -8.391  
## 8 8 74 58.308 15.692  
## 9 9 75 68.225 6.775

ggplot(my\_data4, aes(x = x, y = residuals)) +   
 geom\_point(color = "dark blue") +  
 geom\_hline(yintercept = 0) +  
 xlab("Fitted Values") +  
 ylab("Residuals") +  
 ggtitle("Plot of residuals vs fitted values")+  
 theme\_bw()



# Answer

The pattern in the residual plot is not random and shows clear structure, it suggests that the linear regression model is not appropriate for the data. In this case, the linear relationship between the predictor and response variables may not hold, or there may be additional factors affecting the response that are not captured by the predictor variables in the model. In these situations, a more complex regression model, such as non-linear regression or polynomial regression, may be more appropriate.

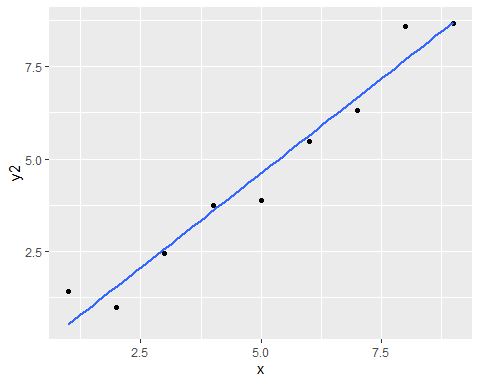
3b) Perform a square root transformation on y , plot your new data and discuss your now plot. Is it closer to being more appropriate for a linear regression model ?

my\_data4$y2 <- sqrt(my\_data4$y)  
my\_data4

## x y yhat residuals y2  
## 1 1 2 -11.111 13.111 1.414214  
## 2 2 1 -1.194 2.194 1.000000  
## 3 3 6 8.723 -2.723 2.449490  
## 4 4 14 18.640 -4.640 3.741657  
## 5 5 15 28.557 -13.557 3.872983  
## 6 6 30 38.474 -8.474 5.477226  
## 7 7 40 48.391 -8.391 6.324555  
## 8 8 74 58.308 15.692 8.602325  
## 9 9 75 68.225 6.775 8.660254

ggplot(data = my\_data4) +  
 geom\_point(mapping = aes(x = x , y = y2)) +  
 geom\_smooth(method = lm,mapping = aes(x = x , y = y2), se=FALSE)

## `geom\_smooth()` using formula = 'y ~ x'



# Answer

The scatter plot shows that linear regression is closer to being more appropriate after the transformation

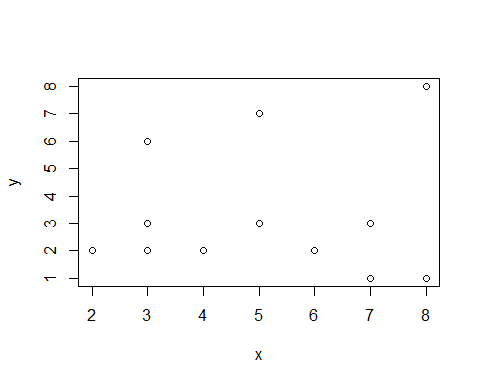
# Question

1. For the bivariate data given below, the residual plots suggest problems involving non-normality or non-constant error variance or both. Use Box – Cox method as indicated in class to produce a lamda power transformation that will best normalize the data. Your work should include all graphs and plots that support your final answer. Use a series of steps and reasoning demonstrated in class.

x <- c(7, 7, 8, 3, 2, 4, 4, 6, 6, 7, 5, 3, 3, 5, 8)  
y <- c(1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 6, 7, 8)  
  
data5 <- as.tibble(data.frame(x,y))  
data5

## # A tibble: 15 × 2  
## x y  
## <dbl> <dbl>  
## 1 7 1  
## 2 7 1  
## 3 8 1  
## 4 3 2  
## 5 2 2  
## 6 4 2  
## 7 4 2  
## 8 6 2  
## 9 6 2  
## 10 7 3  
## 11 5 3  
## 12 3 3  
## 13 3 6  
## 14 5 7  
## 15 8 8

plot(y ~ x)



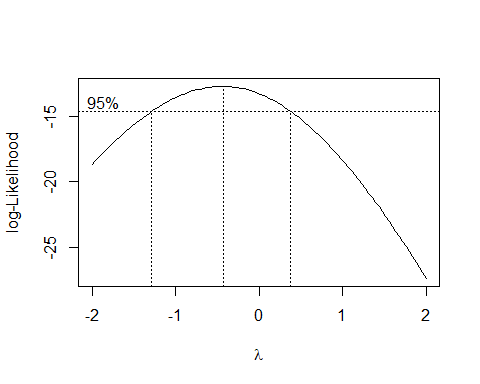
model <- lm(y~x)   
model

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Coefficients:  
## (Intercept) x   
## 3.000e+00 2.709e-16

summary(model)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2 -1 -1 0 5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 3.000e+00 1.717e+00 1.747 0.104  
## x 2.709e-16 3.101e-01 0.000 1.000  
##   
## Residual standard error: 2.287 on 13 degrees of freedom  
## Multiple R-squared: 5.8e-32, Adjusted R-squared: -0.07692   
## F-statistic: 7.541e-31 on 1 and 13 DF, p-value: 1

box\_cox <- boxcox(y ~ x)



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] -18.65840 -18.38718 -18.12041 -17.85823 -17.60076 -17.34814 -17.10051  
## [8] -16.85800 -16.62075 -16.38893 -16.16266 -15.94210 -15.72741 -15.51873  
## [15] -15.31623 -15.12006 -14.93038 -14.74735 -14.57113 -14.40187 -14.23974  
## [22] -14.08490 -13.93749 -13.79768 -13.66562 -13.54145 -13.42532 -13.31737  
## [29] -13.21774 -13.12656 -13.04395 -12.97004 -12.90493 -12.84873 -12.80154  
## [36] -12.76344 -12.73452 -12.71485 -12.70448 -12.70347 -12.71187 -12.72969  
## [43] -12.75698 -12.79373 -12.83995 -12.89562 -12.96075 -13.03528 -13.11919  
## [50] -13.21243 -13.31495 -13.42667 -13.54753 -13.67744 -13.81633 -13.96408  
## [57] -14.12062 -14.28581 -14.45957 -14.64177 -14.83229 -15.03100 -15.23778  
## [64] -15.45251 -15.67504 -15.90525 -16.14300 -16.38816 -16.64058 -16.90014  
## [71] -17.16669 -17.44011 -17.72025 -18.00699 -18.30018 -18.59971 -18.90544  
## [78] -19.21724 -19.53499 -19.85857 -20.18785 -20.52272 -20.86306 -21.20875  
## [85] -21.55968 -21.91575 -22.27685 -22.64287 -23.01372 -23.38928 -23.76947  
## [92] -24.15420 -24.54336 -24.93687 -25.33465 -25.73660 -26.14264 -26.55271  
## [99] -26.96670 -27.38455

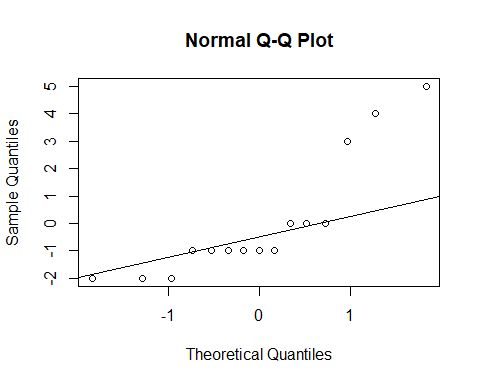
lambda <- box\_cox$x[which.max(box\_cox$y)]   
lambda

## [1] -0.4242424

new\_model <- lm(((y^lambda-1)/lambda) ~ x)   
new\_model

##   
## Call:  
## lm(formula = ((y^lambda - 1)/lambda) ~ x)  
##   
## Coefficients:  
## (Intercept) x   
## 1.00564 -0.06264

qqnorm(model$residuals)   
qqline(model$residuals)



qqnorm(new\_model$residuals)   
qqline(new\_model$residuals)

