IDS702_Assignment1

Anshupriya Srivastava (NetID:as996)

9/4/2019

```
#Adding required libraries
library(ggplot2)
library(readr)
library(moderndive)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Question_1

```
#Reading the file

OldFaithful <- read_csv("OldFaithful.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Date = col_double(),
##   Interval = col_double(),
##   Duration = col_double()
## )
```

```
glimpse(OldFaithful)
```

```
## Observations: 107
## Variables: 4
## $ X1       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ Date     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,...
## $ Interval <dbl> 78, 74, 68, 76, 80, 84, 50, 93, 55, 76, 58, 74, 75, 8...
## $ Duration <dbl> 4.4, 3.9, 4.0, 4.0, 3.5, 4.1, 2.3, 4.7, 1.7, 4.9, 1.7...
```

```
summary(OldFaithful)
```

```
##        X1              Date          Interval       Duration
##  Min.   :  1.0   Min.   :1.000   Min.   :42.0   Min.   :1.700
##  1st Qu.: 27.5   1st Qu.:3.000   1st Qu.:59.0   1st Qu.:2.300
##  Median : 54.0   Median :5.000   Median :75.0   Median :3.800
##  Mean   : 54.0   Mean   :4.514   Mean   :71.0   Mean   :3.461
##  3rd Qu.: 80.5   3rd Qu.:6.000   3rd Qu.:80.5   3rd Qu.:4.300
##  Max.   :107.0   Max.   :8.000   Max.   :95.0   Max.   :4.900
```

Write down a regression model for predicting the interval between eruptions from the duration of the previous one. Make sure to use the right mathematical notation.

$Interval = \beta_0 + \beta_1 * Duration$

Fit the model to the data and interpret your results. In your answer, make sure you include the output from the regression model including the estimated intercept, slope, residual standard error, and R^2.

```
fit_interval <- lm(Interval ~ Duration, data = OldFaithful)
summary(fit_interval)
```

```
##
## Call:
## lm(formula = Interval ~ Duration, data = OldFaithful)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.644  -4.440  -1.088   4.467  15.652
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.8282     2.2618   14.96   <2e-16 ***
## Duration     10.7410     0.6263   17.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.683 on 105 degrees of freedom
## Multiple R-squared:  0.7369, Adjusted R-squared:  0.7344
## F-statistic: 294.1 on 1 and 105 DF,  p-value: < 2.2e-16
```

```
##Intercept: 33.8282
##Slope: 10.7410
##Residual standard error: 6.683 on 105 degrees of freedom
##Multiple R-squared:  0.7369,  Adjusted R-squared:  0.7344
```

Also, include the 95% confidence interval for the slope, and explain what the interval reveals about the relationship between duration and waiting time.

```
confint(fit_interval, level = 0.95)
```

```
##                 2.5 %   97.5 %
## (Intercept) 29.343441 38.31297
## Duration     9.499061 11.98288
```

A confidence level of 95% means that we are confident that 95% of the time the true value of the population parameter will lie with the confidence interval. We are trying to estimate the value of the interval between eruptions from the duration of the previous one.

***Based on 95% confidence level we can say that 95% of the time-**

Confidence interval of the intercept is (29.343442, 38.31297)

Confidence interval of the slope is (9.499061, 11.98288)

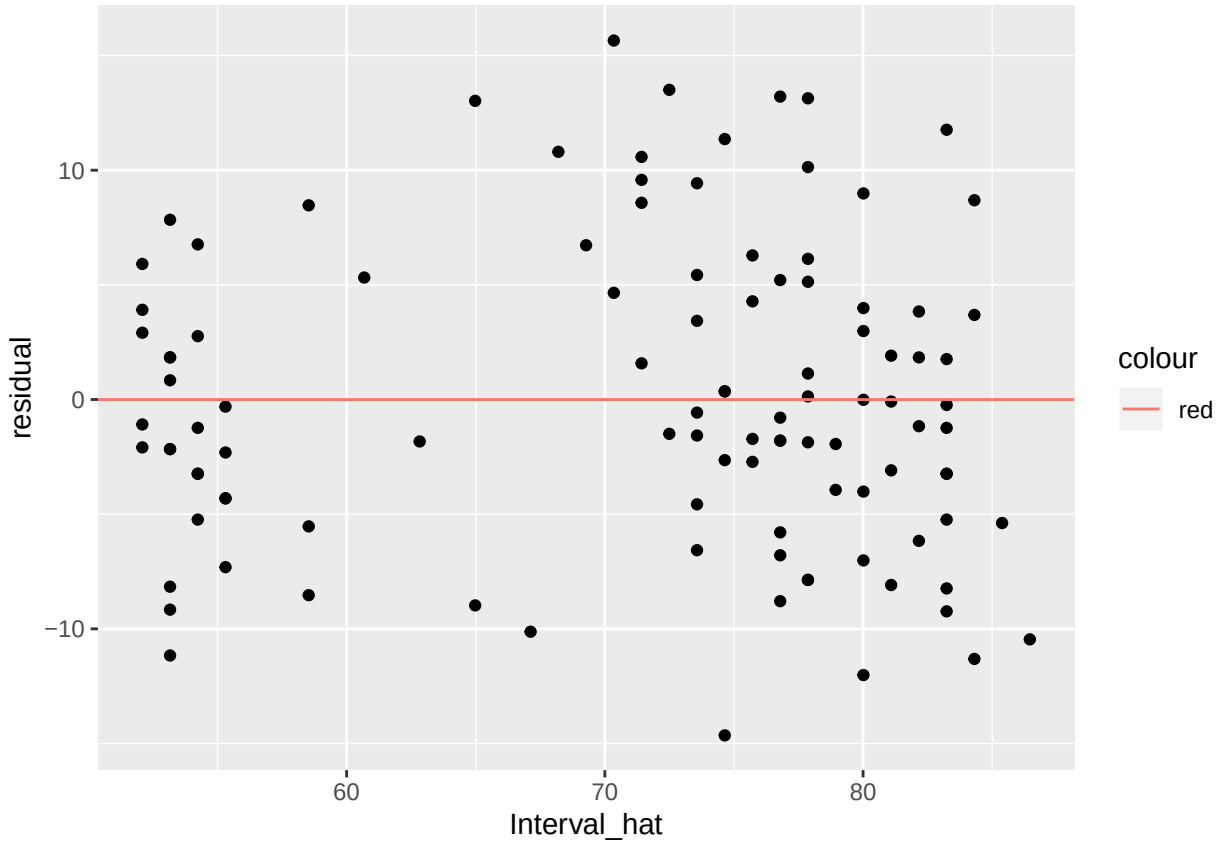
Slope determines that with every one unit increase in the duration, the interval between eruptions will also increase by (9.499061, 11.98288).

Describe in a few sentences whether or not you think the regression assumptions are plausible based on residual plots (you don't need to include the plots).

```
rp_OldFaithful <- get_regression_points(fit_interval)
rp_OldFaithful
```

```
## # A tibble: 107 x 5
##       ID Interval Duration Interval_hat residual
##    <int>    <dbl>    <dbl>        <dbl>    <dbl>
##  1     1       78      4.4         81.1   -3.09
##  2     2       74      3.9         75.7   -1.72
##  3     3       68      4           76.8   -8.79
##  4     4       76      4           76.8   -0.792
##  5     5       80      3.5         71.4    8.58
##  6     6       84      4.1         77.9    6.13
##  7     7       50      2.3         58.5   -8.53
##  8     8       93      4.7         84.3    8.69
##  9     9       55      1.7         52.1    2.91
## 10    10       76      4.9         86.5  -10.5
## # ... with 97 more rows
```

```
ggplot(rp_OldFaithful, aes(x = Interval_hat, y = residual)) + geom_point() +
  geom_hline(aes(yintercept = sum(residual), col = "red"))
```



Residual plot shows the residuals on the y-axis and the predicted values on the x-axis. When this plot shows a random pattern it supports a linear model. Residual is the difference between the observed and the calculated value. The sum of all residuals should be equal to zero. When the residual shows a pattern there is a high tendency that the sum might not add upto zero. Thus we know that a regression model may not be the best for the given data set.

I think that regression assumptions are plausible based on residual plots because the sum of the residuals is roughly equal to 0. This indicates a good fit.

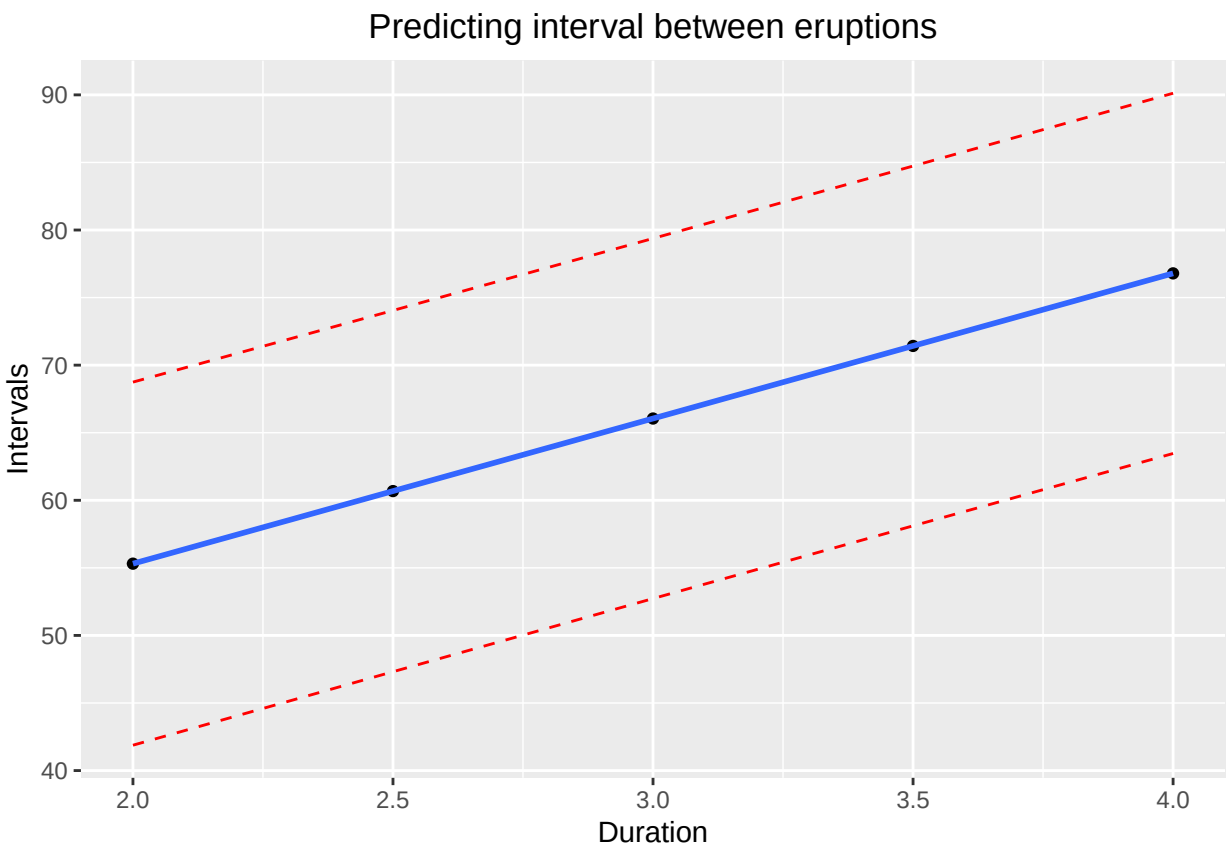
Construct 95% prediction intervals for the waiting time until the next eruption if the duration of the previous one was 2 minutes, 2.5 minutes, 3 minutes, 3.5 minutes and 4 minutes. Present your answer as a single plot.

```
predict_df <- data.frame("Duration" = c(2, 2.5, 3, 3.5, 4))
pred_interval <- predict(fit_interval, newdata=predict_df, interval="prediction",
                         level = 0.95)
pred_interval
```

```
##        fit      lwr      upr
## 1 55.31015 41.87495 68.74535
## 2 60.68064 47.31512 74.04616
## 3 66.05112 52.72668 79.37557
## 4 71.42161 58.10936 84.73385
## 5 76.79209 63.46310 90.12108
```

```
new_df = cbind(predict_df, pred_interval)
#new_df

ggplot(new_df, aes(x = Duration, y = fit)) + geom_point() +
  geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
  geom_line(aes(y=upr), color = "red", linetype = "dashed")+
  geom_smooth(method=lm, se=TRUE) + labs(y = "Intervals") +
  ggtitle("Predicting interval between eruptions") +
  theme(plot.title = element_text(hjust = 0.5))
```



Question_2

```
Respiratory <- read_csv("Respiratory.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Age = col_double(),
##   Rate = col_double()
## )
```

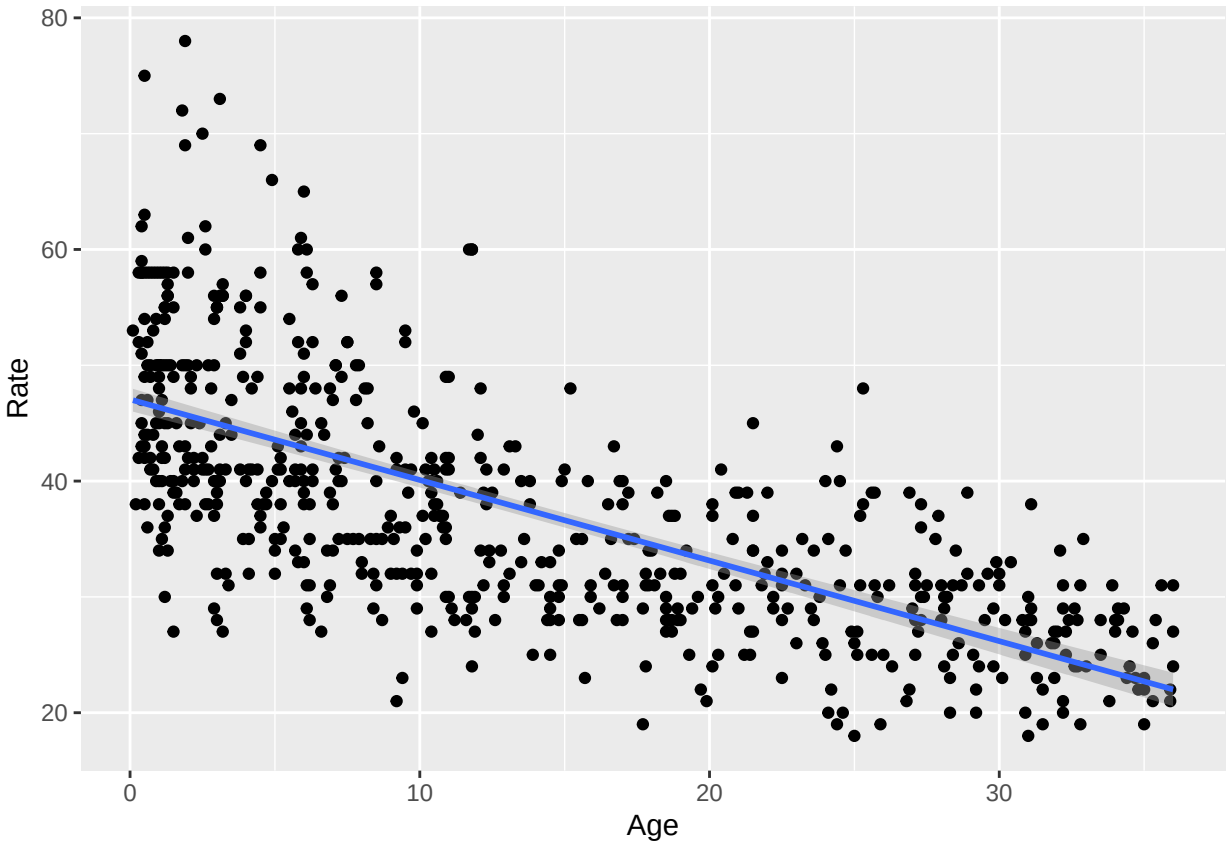
```
summary(Respiratory)
```

```
##        X1              Age              Rate
##  Min.   :  1.0   Min.   : 0.10   Min.   :18.00
##  1st Qu.:155.2   1st Qu.: 3.80   1st Qu.:30.00
##  Median :309.5   Median :10.55   Median :36.50
##  Mean   :309.5   Mean   :13.39   Mean   :37.74
##  3rd Qu.:463.8   3rd Qu.:22.00   3rd Qu.:44.00
##  Max.   :618.0   Max.   :36.00   Max.   :78.00
```

Analyze the data and include a useful plot that a physician could use to assess a normal range of respiratory rate for children of any age between 0 and 3.

```
Respiratory["Rate_log10"] = log10(Respiratory$Rate)

ggplot(data = Respiratory, aes(x = Age, y = Rate)) + geom_point() +
  labs(y = "Rate") + stat_smooth(method = lm)
```



The rate v/s age graph shows a linear trend.

Include the output of the regression that predicts respiratory rates from age. Also, is there enough evidence that the model assumptions are reasonable for this data? You should consider transformations (think log transformations etc) for both variables if you think the original relationship is nonlinear.

```
fit_respiratory = lm(Rate_log10~Age, data = Respiratory)
summary(fit_respiratory)
```

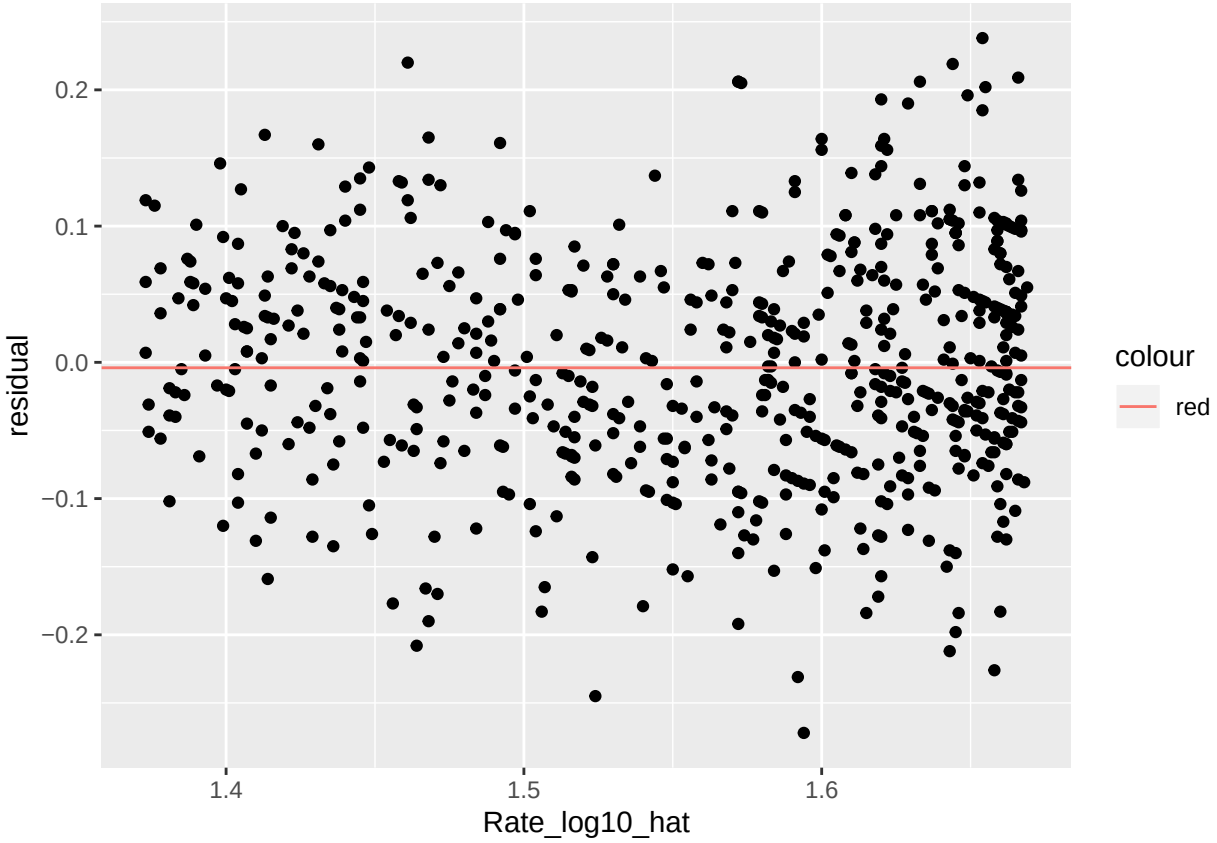
```
##
## Call:
## lm(formula = Rate_log10 ~ Age, data = Respiratory)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.271744 -0.057330 -0.001746  0.058581  0.237866
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.6699138  0.0054841  304.50   <2e-16 ***
## Age         -0.0082555  0.0003195  -25.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0853 on 616 degrees of freedom
## Multiple R-squared:  0.5201, Adjusted R-squared:  0.5193
## F-statistic: 667.6 on 1 and 616 DF,  p-value: < 2.2e-16
```

There are three things we can use to evaluate model fit:

1. R-Squared: The value of R-squared ranges from 0 to 1. 0 indicates that the proposed model is not a good fit whereas 1 indicates a perfect fit. Our model has R-squared: 0.5201 and Adjusted R-squared: 0.5193. We can say that age has a moderate effect on Rate.
2. F-Test: The result of an F-test help in determining whether or not we accept our null hypothesis. Our model is rejecting the null hypothesis which suggests that there can be a relationship between rate and age.
3. Residual Standard Error: Residual Standard Error helps determine how close the observed points are to the predicted values. The value for our model is 0.0853 which is very low.

All these factors show a good fit.

```
rp <- get_regression_points(fit_respiratory)
ggplot(data = rp, aes(x = Rate_log10_hat, y = residual)) + geom_point() + geom_hline(aes(yintercept = s
```



Apart from R-squared, F-Test and RMSE a residual plot is also useful to identify if our model fits the data well. In this case, we can see that the sum of errors is roughly centered around 0. So, the model is a good fit.

Demonstrate the usefulness of the model by providing 95% prediction intervals for the rate for three individual children: a 1 month old, an 18 months old, and a 29 months old.

```
#Assuming every month has 30 days
age_1 = log10(1 * 30)
age_2 = log10(18 * 30)
age_3 = log10(29 * 30)
rate_predict_df <- data.frame("Age" = c(age_1, age_2, age_3 ))

rate_pred_interval <- predict(fit_respiratory,
                      newdata=rate_predict_df, interval="prediction",
```

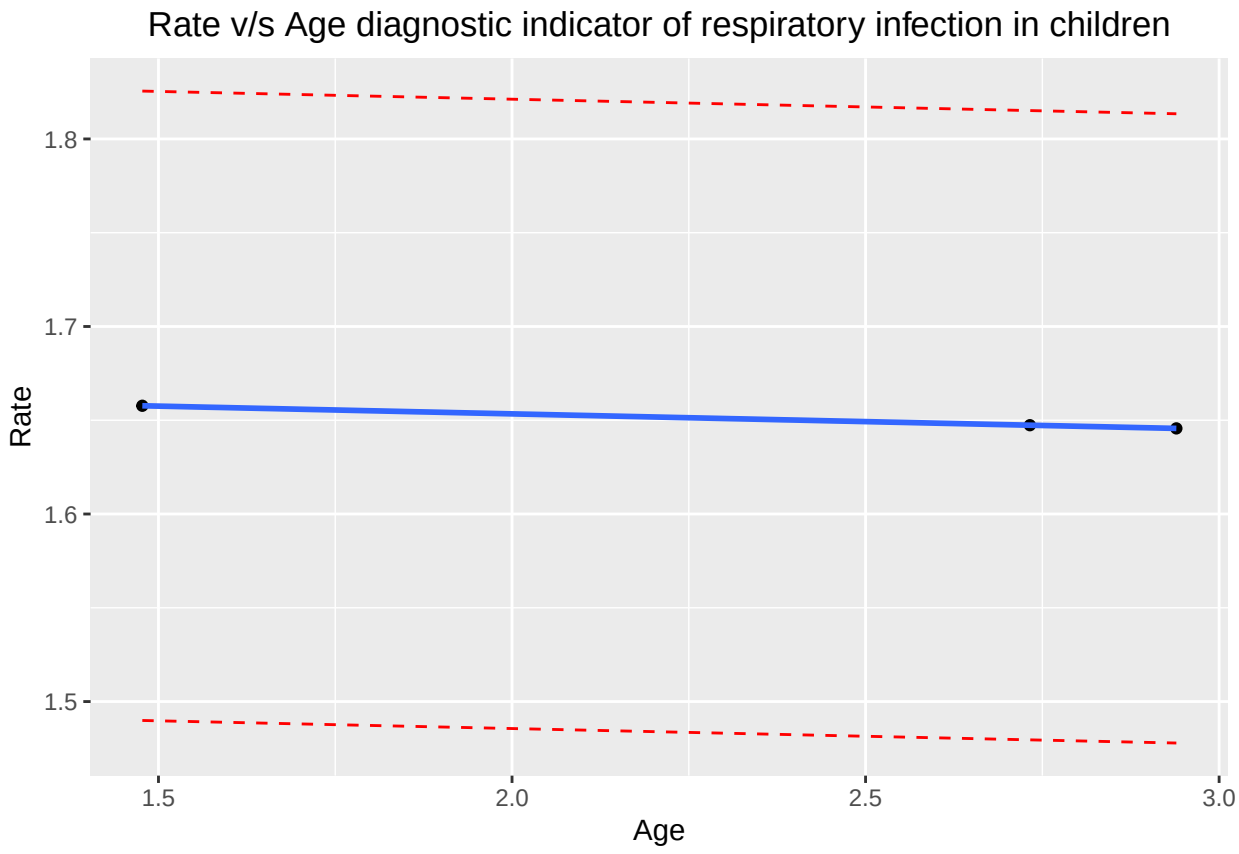
```
                      level = 0.95)
rate_pred_interval
```

```
##        fit      lwr      upr
## 1 1.657719 1.489912 1.825526
## 2 1.647357 1.479583 1.815130
## 3 1.645647 1.477878 1.813415
```

```
new_respiratory_df = cbind(rate_predict_df, rate_pred_interval)
new_respiratory_df
```

```
##        Age      fit      lwr      upr
## 1 1.477121 1.657719 1.489912 1.825526
## 2 2.732394 1.647357 1.479583 1.815130
## 3 2.939519 1.645647 1.477878 1.813415
```

```
ggplot(new_respiratory_df, aes(x = Age, y = fit)) + geom_point() +
  geom_line(aes(y=lwr), color = "red", linetype = "dashed") +
  geom_line(aes(y=upr), color = "red", linetype = "dashed") +
  geom_smooth(method=lm, se=TRUE) + labs(y = "Rate") +
  ggtitle("Rate v/s Age diagnostic indicator of respiratory infection in children") +
  theme(plot.title = element_text(hjust = 0.5))
```



Question_4

```
Elections <- read_csv("Elections.csv")
```

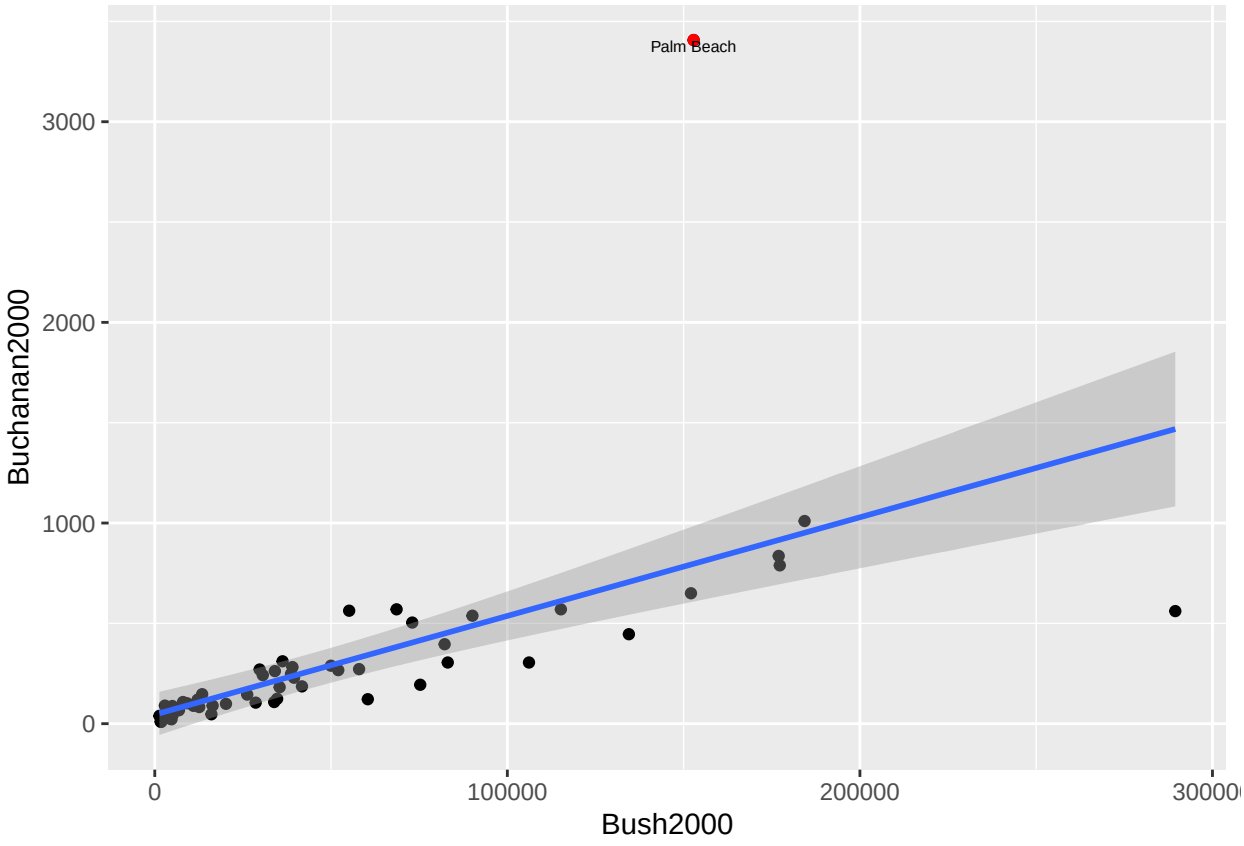
```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   County = col_character(),
##   Buchanan2000 = col_double(),
##   Bush2000 = col_double()
## )
```

```
options(scipen = 999)

palm_beach_county <- subset(Elections, County == "Palm Beach")

ggplot(data = Elections, aes(x = Bush2000, y = Buchanan2000)) +
    geom_point() +
    geom_point(data = palm_beach_county , colour="red") +
    geom_text(data = palm_beach_county, label="Palm Beach", vjust=1, size = 2) +
    stat_smooth(method = lm)
```



Buchanan has generally recieved smaller number of votes. Most of the data points are concentrated within the range of 0 to 1000 for him. Palm Beach is the only county where he seems to have recieved more than 3000 votes. There is a chance that he recieved more than the expected number of votes in Palm Beach County.

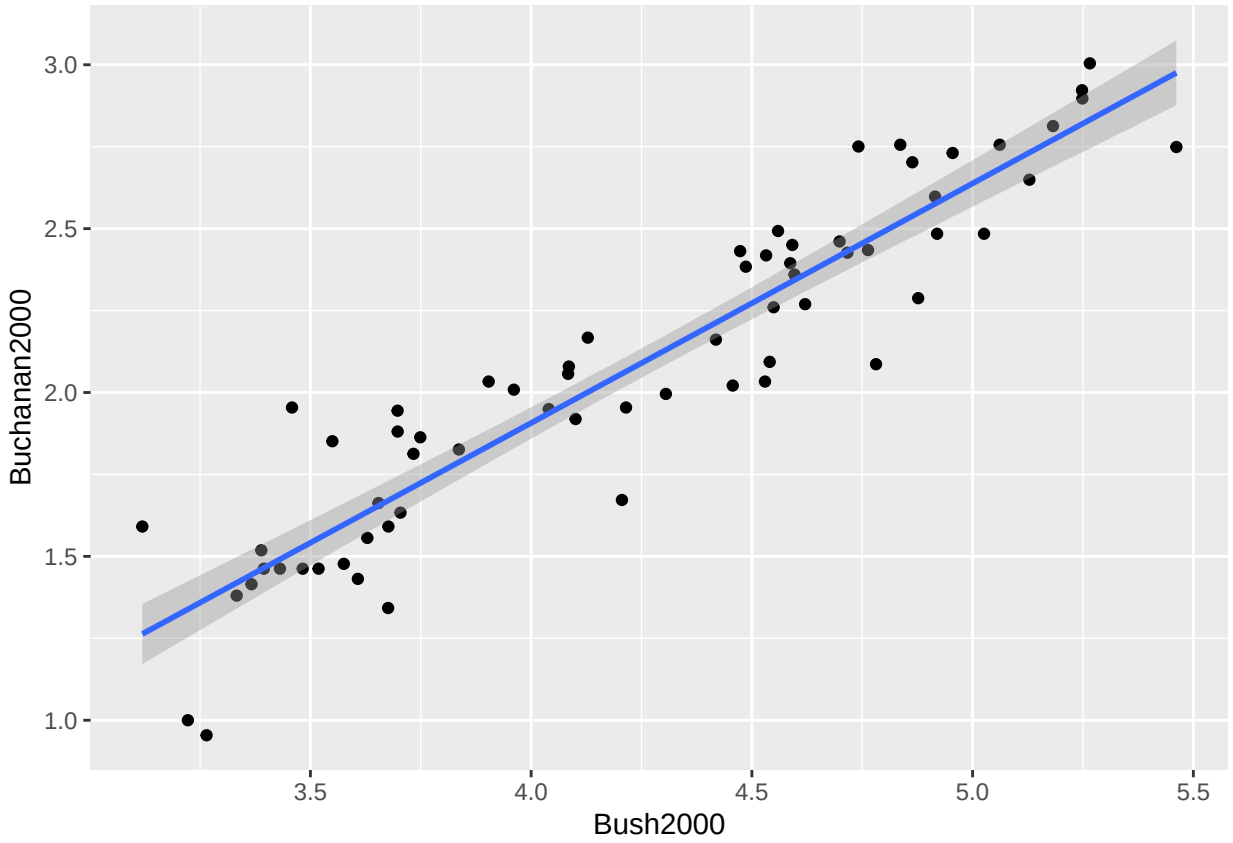
Analyze the data without Palm Beach County results to obtain an equation for predicting Buchanan votes from Bush votes. You should consider transformations (think log transformations etc) for both variables if you think the original relationship is nonlinear.

```
Elections_new = Elections[-c(67),]
Elections_new["Buchanan2000_log10"] = log10(Elections_new$Buchanan2000)
```

```
Elections_new["Bush2000_log10"] = log10(Elections_new$Bush2000)

Elections_new["Buchanan2000_log10"] = log10(Elections_new$Buchanan2000)
```

```
ggplot(data = Elections_new, aes(x = Bush2000_log10, y = Buchanan2000_log10)) +
    geom_point() + stat_smooth(method = lm) + labs(x = "Bush2000", y = "Buchanan2000")
```



Equation for predicting Buchanan's votes from Bush's votes -

$Buchanan2000 = \beta_0 + \beta_1 * Bush2000$ [Assuming all values in base 10]

```
fit_votes = lm(Buchanan2000_log10~Bush2000_log10, data = Elections_new)
```

Include the output from the final regression model that you used, as well as evidence that the model fits the assumptions reasonably well.

```
summary(fit_votes)
```

```
##
## Call:
## lm(formula = Buchanan2000_log10 ~ Bush2000_log10, data = Elections_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41532 -0.09223  0.01087  0.12204  0.44323
##
## Coefficients:
##                Estimate Std. Error t value            Pr(>|t|)
## (Intercept)    -1.01689    0.15392  -6.607        0.00000000907 ***
```

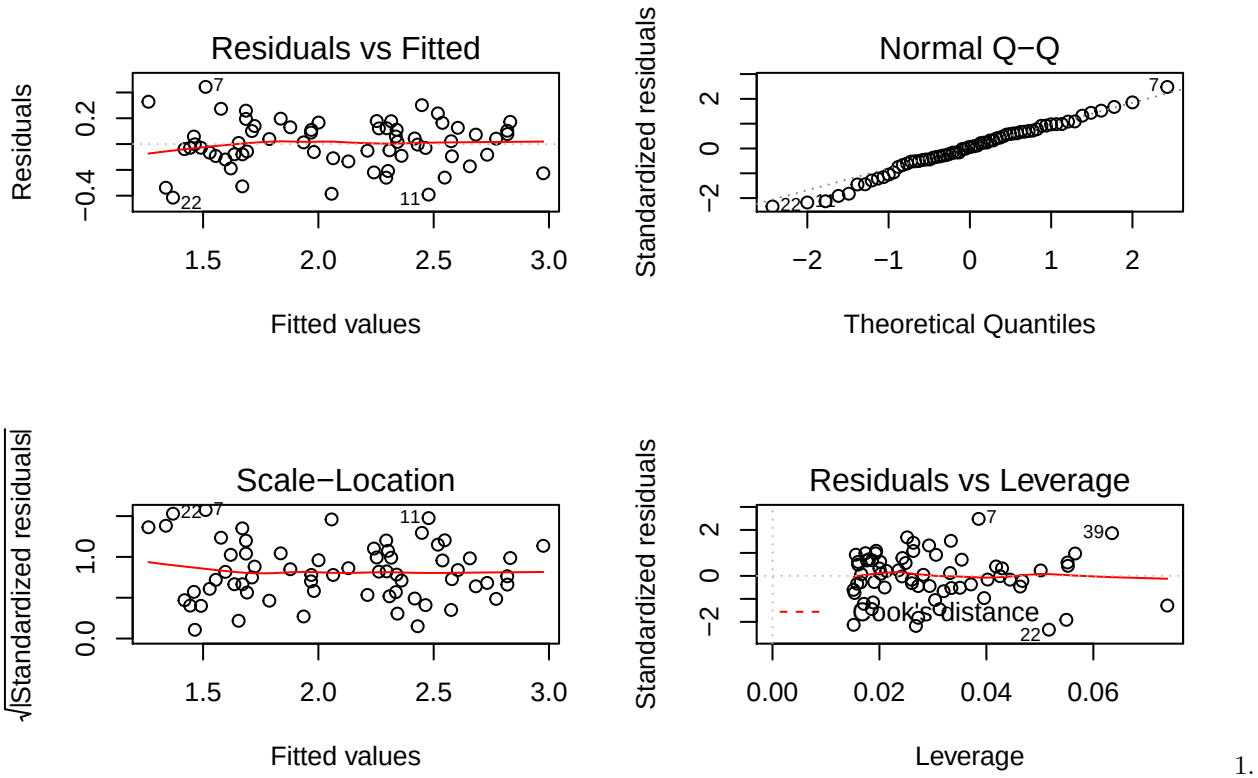
```
## Bush2000_log10  0.73096    0.03597  20.323 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1823 on 64 degrees of freedom
## Multiple R-squared:  0.8658, Adjusted R-squared:  0.8637
## F-statistic:   413 on 1 and 64 DF,  p-value: < 0.00000000000000022
```

There are three things we can use to evaluate model fit:

1. R-Squared: The value of R-squared ranges from 0 to 1. 0 indicates that the proposed model is not a good fit whereas 1 indicates a perfect fit. Our model has R-squared: 0.8658 and Adjusted R-squared: 0.8637. We can say that Bush's votes have a high effect on Buchanan's votes.
2. F-Test: The result of an F-test help in determining whether or not we accept our null hypothesis. Our model is rejecting the null hypothesis which indicates a relationship between the votes recieved by the two candidates.
3. Residual Standard Error: Residual Standard Error helps determine how close the observed points are to the predicted values. The value for our model is 0.1823 which is quite low.

```
#rp <- get_regression_points(fit_votes)

par(mfrow = c(2, 2))
plot(fit_votes)
```



1. Residual v/s Fitted: Shows Linearity 2. Normal Q-Q: Shows that it's normal 3. RMSE and Fitted values: It's roughly centered around 0. 4. Leverage: Three points 7, 22, 39 have a very high leverage.

Apart from R-squared, F-Test and RMSE a residual plot is also useful to identify if our model fits the data well. In this case, we can see that the sum of errors is roughly centered around 0. So, the model is a good fit.

Obtain a 95% prediction interval for the number of Buchanan votes in Palm Beach from this result, assuming the relationship is the same in this county as in the others. If it is assumed that Buchanan's actual count contains a number of votes intended for Gore, what can be said about the likely size of this number from the prediction interval?

```
Bush_val = log10(152846)

predict_vote_df <- data.frame("Bush2000_log10" = Bush_val)

pred_vote_interval <- predict(fit_votes, newdata=predict_vote_df, interval="prediction", level = 0.95)

pred_vote_interval
```

```
##        fit      lwr      upr
## 1 2.772598 2.399328 3.145869
```

```
fit_value = 2.772598
Buchanan_true_predicted_val = 10^2.772598
Buchanan_range_min = 10^2.399328
Buchanan_range_max = 10^3.145869

paste0("Buchanan Range at 95% confidence level: ", "(", Buchanan_range_min, ", ",
       Buchanan_range_max, ")")
```

```
## [1] "Buchanan Range at 95% confidence level: (250.800270173054, 1399.16521661096)"
```

```
paste0("Buchanan_true_predicted_val at 95% confidence level: ", Buchanan_true_predicted_val)
```

```
## [1] "Buchanan_true_predicted_val at 95% confidence level: 592.376743585725"
```

By using a linear model we have concluded that Buchanan received around 592 votes at a 95% confidence level. We can use the values calcualated for Buchanan and create an assumption for Gore.

If and only if the additional votes were meant for Gore, he should ideally receive at least 251 votes and at most 1399 more votes than he actually received.