

# linear regression model and factor analysis for erosion

Yingmai Chen, Yan Wang

2023-11-26

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.1      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v lubridate  1.9.2      v tibble    3.2.1
## v purrr      1.0.2      v tidyr     1.3.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
erosion<-read.csv("erosion.csv")
```

linear regression for all variables

```
library(readr)
data <- read_csv("erosion.csv")
```

```
## Rows: 31 Columns: 10
## -- Column specification -----
## Delimiter: ","
## chr (1): Bluff
## dbl (9): Orientation (deg), RR (m/yr), ErosionVulnerability, Wave Height for...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
names(data) <- gsub(" ", "_", names(data))
names(data) <- gsub("\\(", "_(", names(data))
names(data) <- gsub("\\)", "_)", names(data))
names(data) <- gsub("/", "_per_", names(data))
names(data) <- gsub("%", "percent", names(data))
if ("RR (m/yr)" %in% names(data)) {
  names(data)[names(data) == "RR (m/yr)"] <- "RR_m_per_yr"
}
predictors <- setdiff(names(data), c("Bluff", "RR_m_per_yr"))
```

```
target <- "RR_m_per_yr"
model_formula <- as.formula(paste(target, "~", paste(predictors, collapse = " + ")))
model <- lm(model_formula, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = model_formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41347 -0.11112 -0.01882  0.06728  0.59329
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.237e-01  2.250e-01   0.550   0.5881
## Orientation_deg    6.972e-05  3.878e-04   0.180   0.8590
## ErosionVulnerability 1.722e-01  6.738e-02   2.556   0.0180 *
## Wave_Height_for_NNE_wind_15_m_per_s_m 8.558e-01  4.808e-01   1.780   0.0889 .
## Max_Wave_Height_m   -4.089e-01  3.976e-01  -1.028   0.3150
## Mud_percent        -8.982e-03  1.799e-02  -0.499   0.6226
## BaseEl_m           5.934e-03  6.560e-02   0.090   0.9287
## BluffEl_m          -1.531e-02  9.174e-03  -1.668   0.1094
## Seawall            -1.032e-01  1.254e-01  -0.823   0.4193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2088 on 22 degrees of freedom
## Multiple R-squared:  0.7008, Adjusted R-squared:  0.5921
## F-statistic: 6.442 on 8 and 22 DF,  p-value: 0.0002397
```

```
predictions <- predict(model, data)
mse <- mean((data[[target]] - predictions)^2)
rsquared <- summary(model)$r.squared
cat("Mean Squared Error (MSE):", mse, "\n")
```

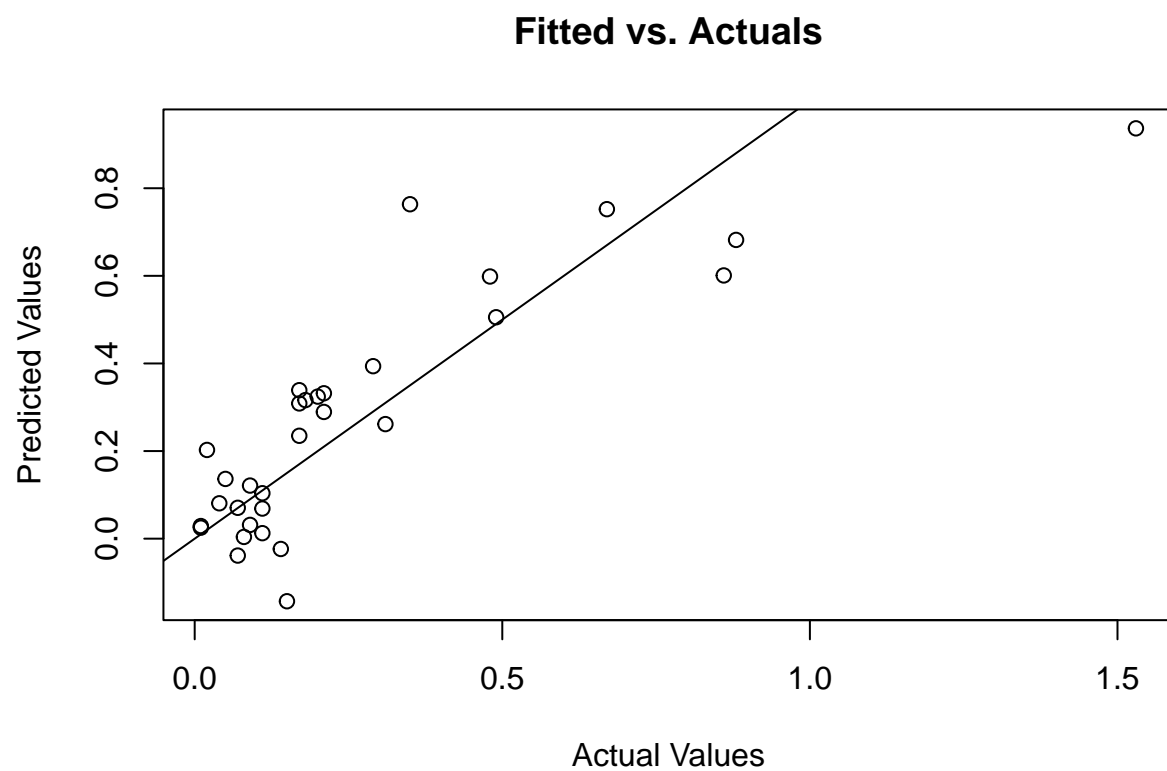
```
## Mean Squared Error (MSE): 0.03094682
```

```
cat("R-squared:", rsquared, "\n")
```

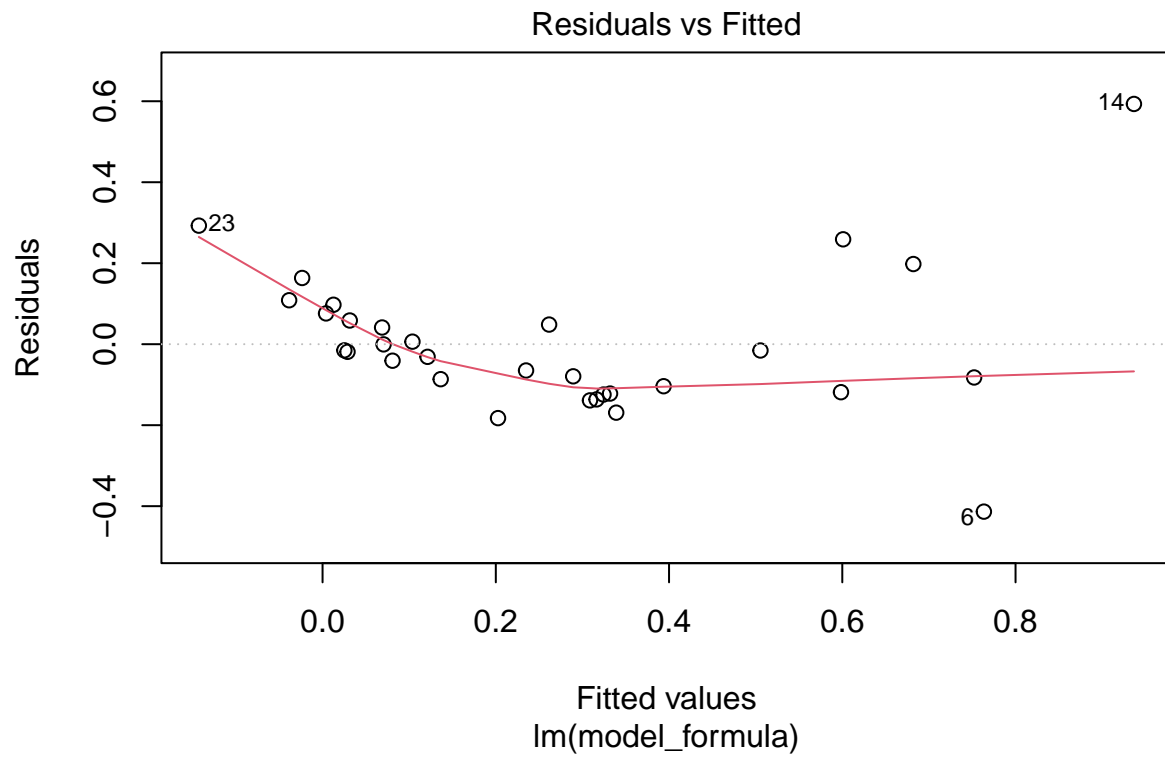
```
## R-squared: 0.7008402
```

## check

```
#Check Linearity
plot(data[[target]], predictions, main = "Fitted vs. Actuals",
      xlab = "Actual Values", ylab = "Predicted Values")
abline(0, 1)
```

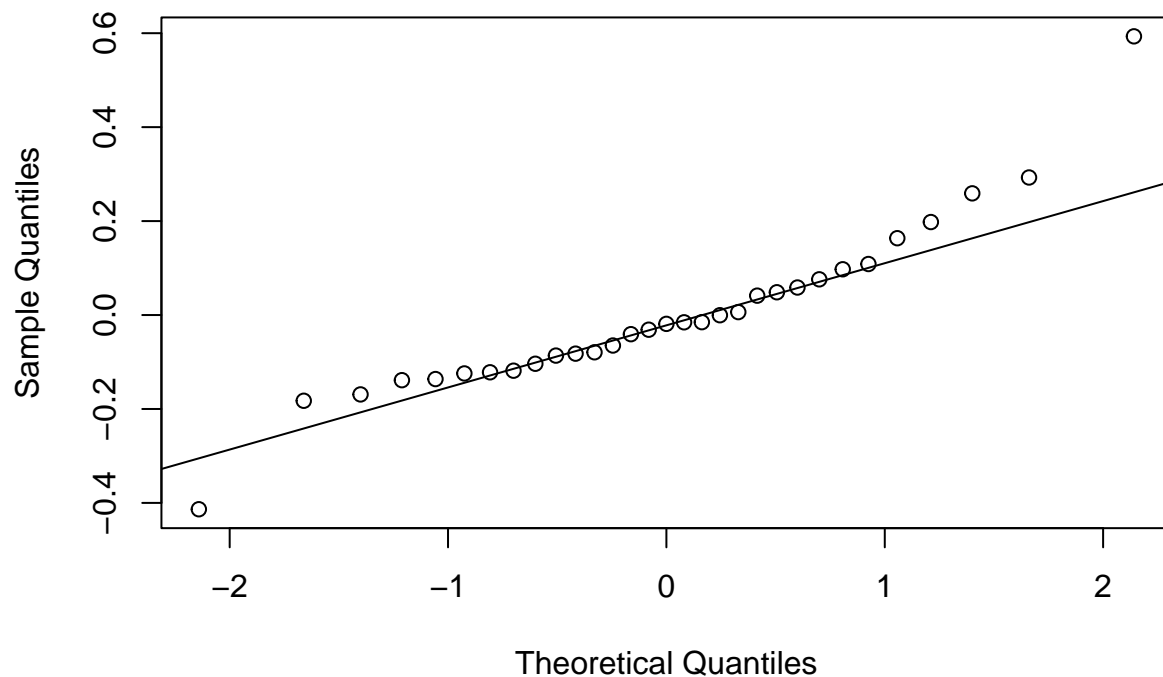


```
#Check Homoscedasticity  
plot(model, which = 1)
```



```
#Check Normal Distribution of Residuals  
qqnorm(model$residuals)  
qqline(model$residuals)
```

## Normal Q-Q Plot



linear regression for single variable

```
data <- read.csv("erosion.csv")
model1 <- lm(`RR..m.yr.` ~ `Mud....`, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = model_formula, data = data)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.41347	-0.11112	-0.01882	0.06728	0.59329

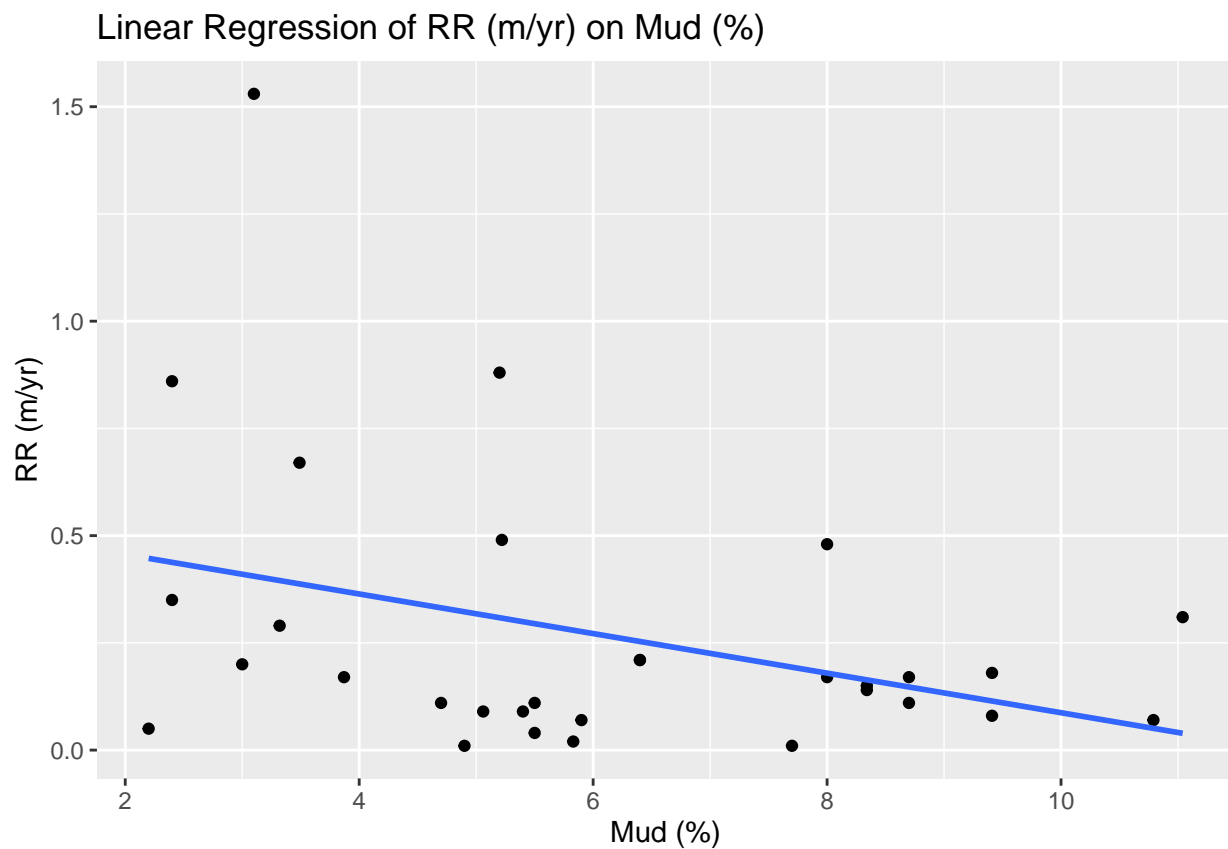
```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.237e-01	2.250e-01	0.550	0.5881
Orientation_deg	6.972e-05	3.878e-04	0.180	0.8590
ErosionVulnerability	1.722e-01	6.738e-02	2.556	0.0180 *
Wave_Height_for_NNE_wind_15_m_per_s_m	8.558e-01	4.808e-01	1.780	0.0889 .
Max_Wave_Height_m	-4.089e-01	3.976e-01	-1.028	0.3150
Mud_percent	-8.982e-03	1.799e-02	-0.499	0.6226
BaseEl_m	5.934e-03	6.560e-02	0.090	0.9287
BluffEl_m	-1.531e-02	9.174e-03	-1.668	0.1094

```
## Seawall -1.032e-01 1.254e-01 -0.823 0.4193
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2088 on 22 degrees of freedom
## Multiple R-squared: 0.7008, Adjusted R-squared: 0.5921
## F-statistic: 6.442 on 8 and 22 DF, p-value: 0.0002397
```

```
ggplot(data, aes(x = `Mud....`, y = `RR..m.yr.`)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Linear Regression of RR (m/yr) on Mud (%)",
       x = "Mud (%)",
       y = "RR (m/yr)")
```

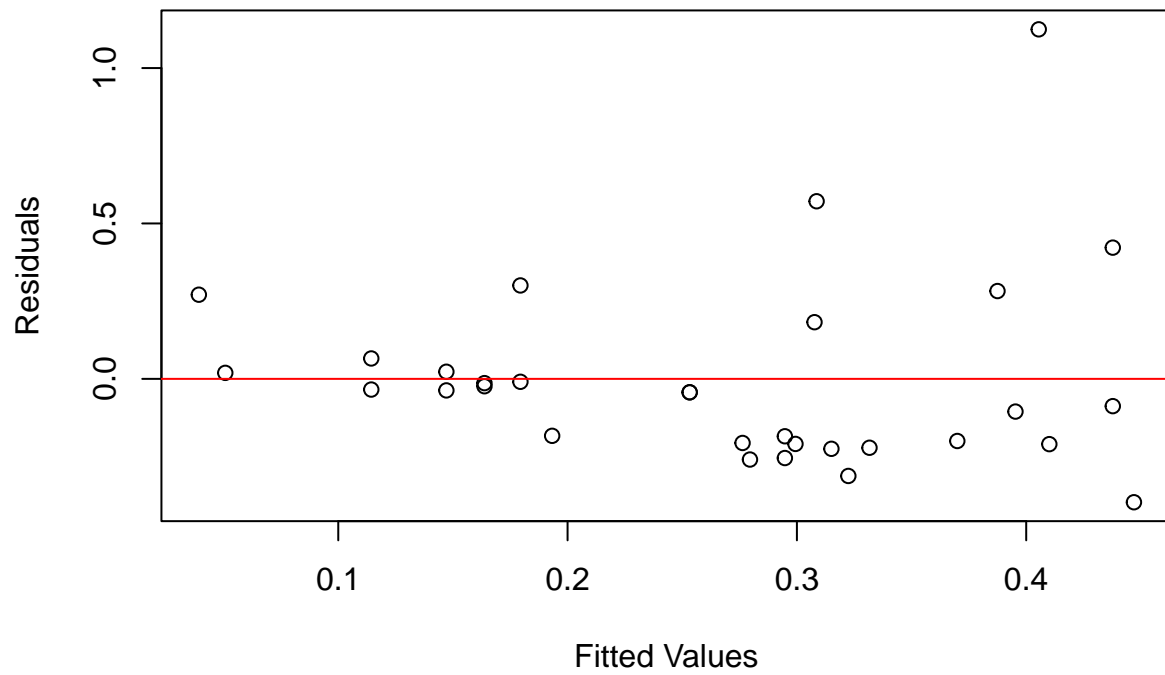
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
### check
```

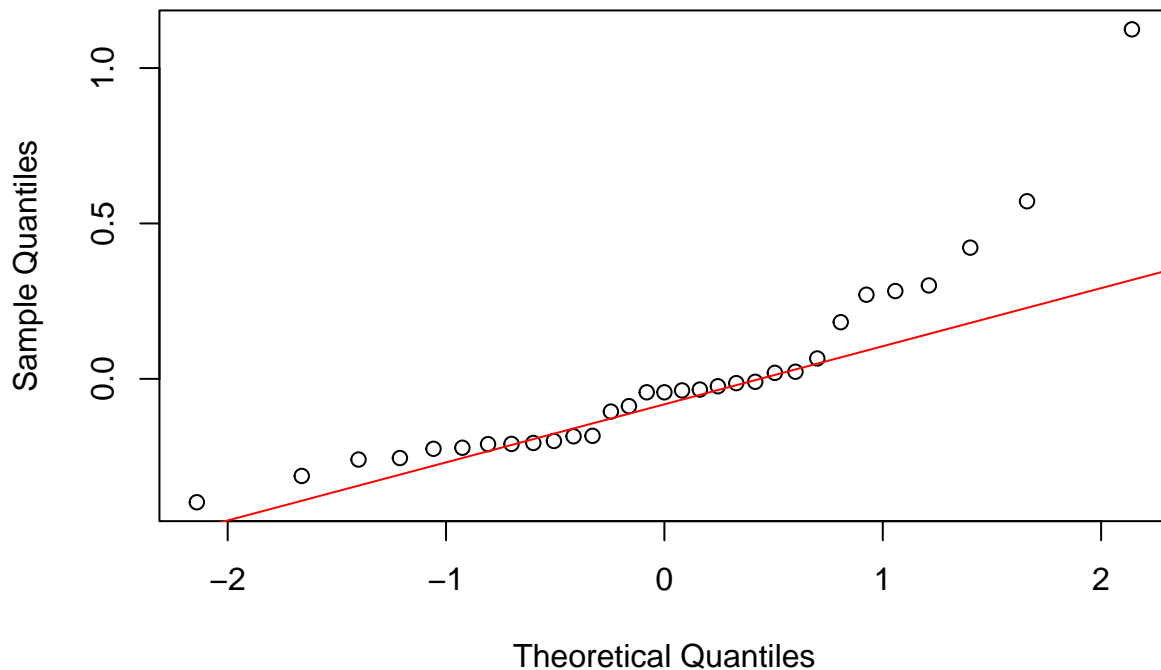
```
# Plotting Residuals vs Fitted Values
plot(model1$fitted.values, model1$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```

## Residuals vs Fitted



```
# Plotting Normal Q-Q plot for Residuals  
qqnorm(model1$residuals)  
qqline(model1$residuals, col = "red")
```

## Normal Q-Q Plot



```
model2 <- lm(`RR..m.yr.` ~ `Wave.Height.for.NNE.wind.15.m.s..m.`, data = data)
summary(model2)
```

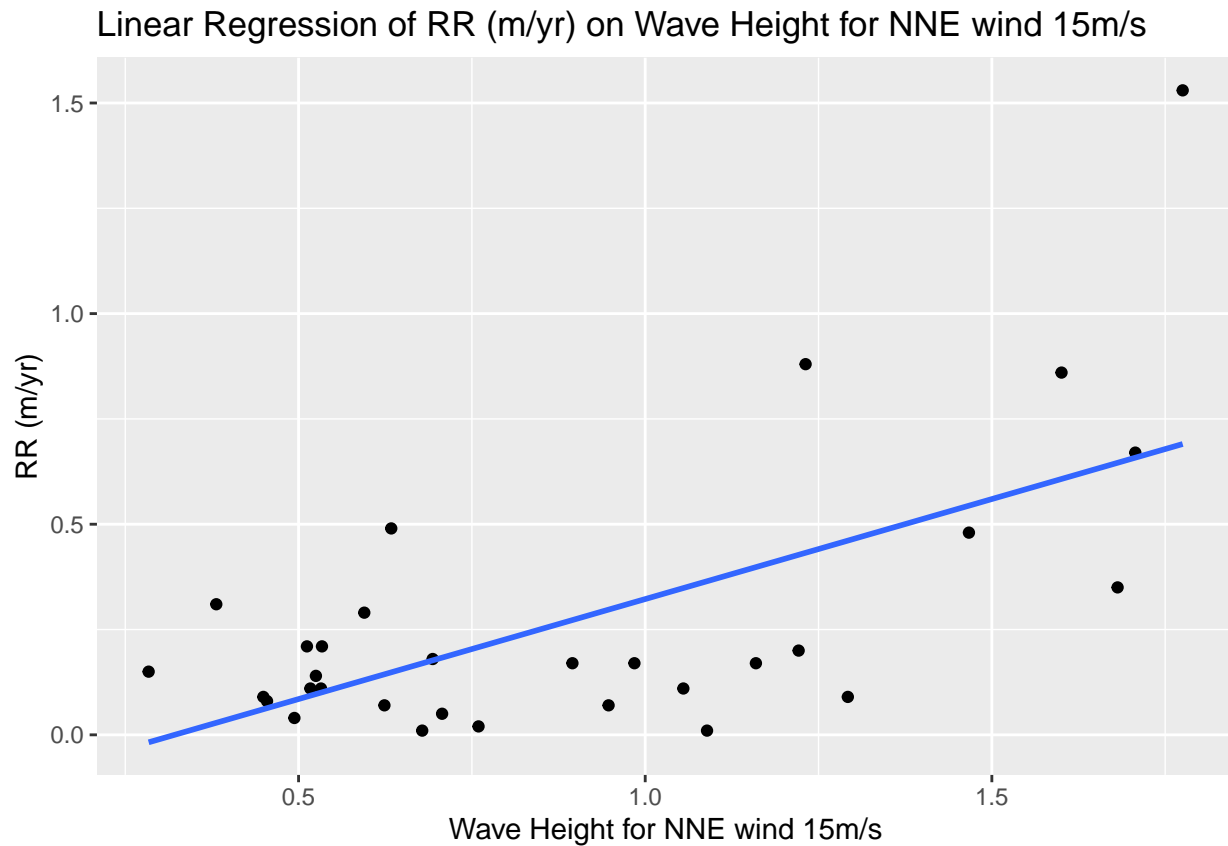
```
##
## Call:
## lm(formula = RR..m.yr. ~ Wave.Height.for.NNE.wind.15.m.s..m.,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37108 -0.17386  0.00325  0.11422  0.83960
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.1525     0.1069  -1.427  0.164314
## Wave.Height.for.NNE.wind.15.m.s..m.  0.4748     0.1086   4.371  0.000145 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2582 on 29 degrees of freedom
## Multiple R-squared:  0.3971, Adjusted R-squared:  0.3763
## F-statistic: 19.1 on 1 and 29 DF, p-value: 0.000145
```

```
ggplot(data, aes(x = `Wave.Height.for.NNE.wind.15.m.s..m.`, y = `RR..m.yr.`)) +
  geom_point() +
```



```
geom_smooth(method = "lm", se = FALSE) +
labs(title = "Linear Regression of RR (m/yr) on Wave Height for NNE wind 15m/s",
      x = "Wave Height for NNE wind 15m/s",
      y = "RR (m/yr)")
```

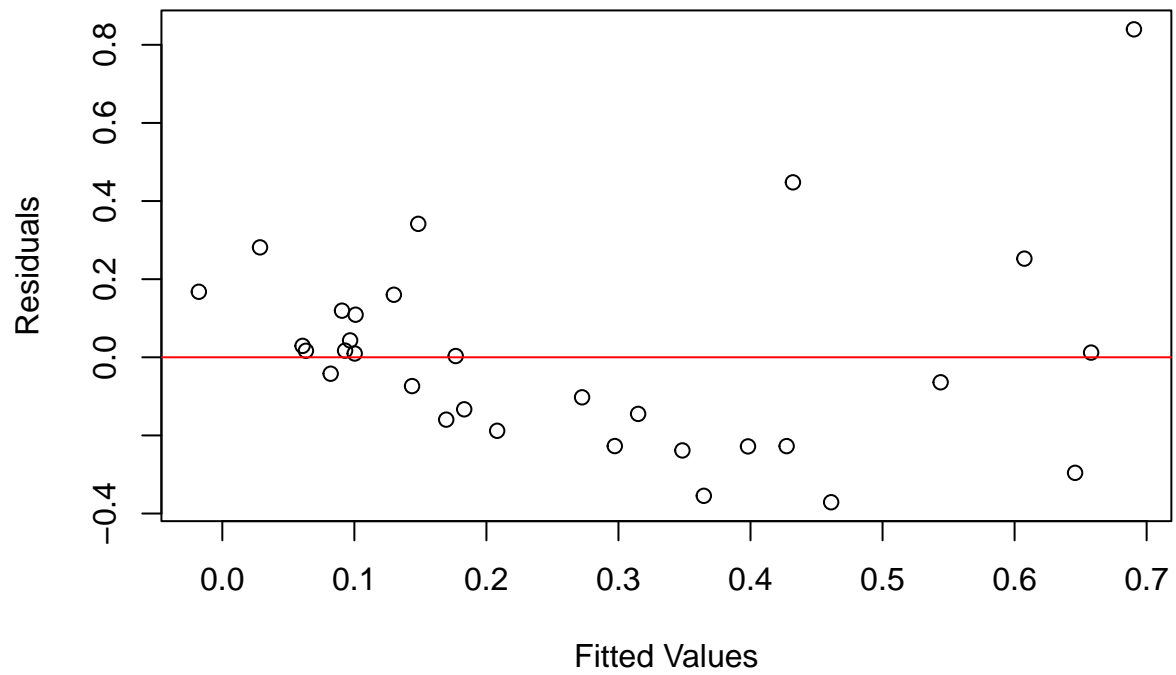
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
### check
```

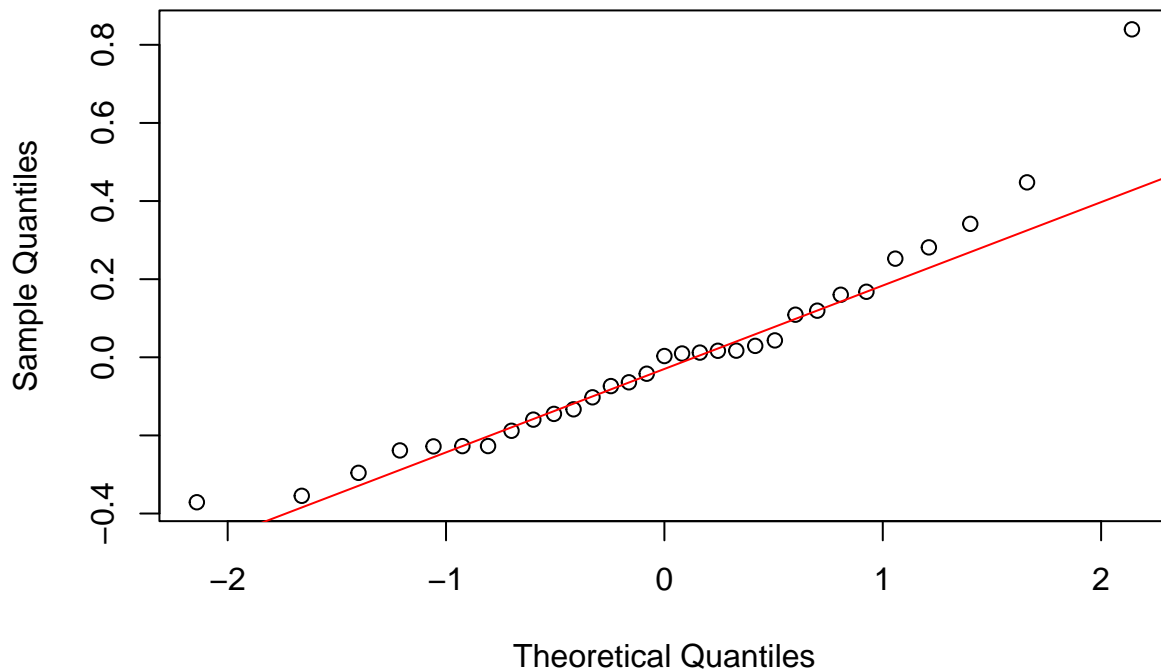
```
# Residuals vs Fitted Values Plot
plot(model2$fitted.values, model2$residuals,
      xlab = "Fitted Values", ylab = "Residuals",
      main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```

## Residuals vs Fitted



```
# Normal Q-Q Plot for Residuals  
qqnorm(model2$residuals)  
qqline(model2$residuals, col = "red")
```

## Normal Q-Q Plot



```
model3 <- lm(`RR..m.yr.` ~ `Max.Wave.Height..m.`, data = data)
summary(model3)
```

```
##
## Call:
## lm(formula = RR..m.yr. ~ Max.Wave.Height..m., data = data)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.37423	-0.17220	-0.03261	0.09062	0.93464

```
##
## Coefficients:
```

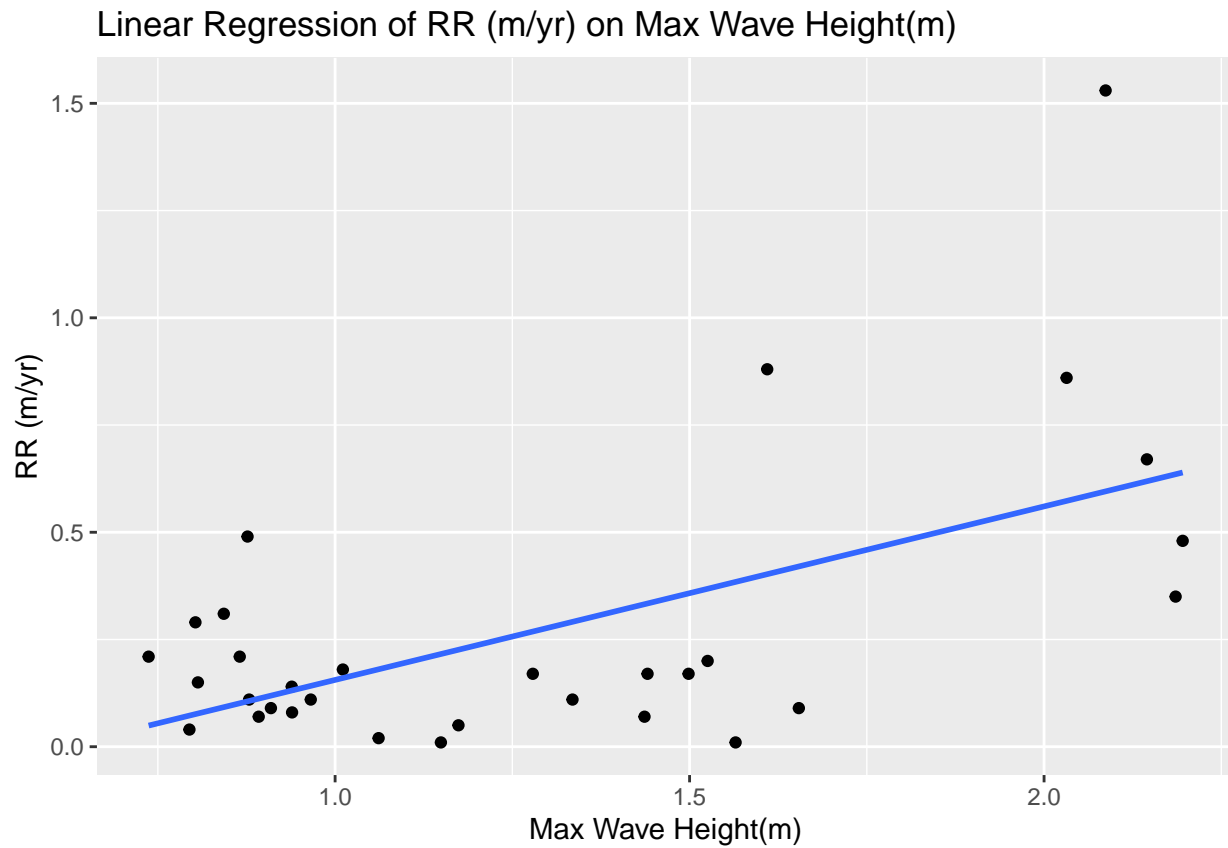
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.2488	0.1440	-1.727	0.094781 .
Max.Wave.Height..m.	0.4045	0.1060	3.815	0.000659 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2713 on 29 degrees of freedom
## Multiple R-squared:  0.3342, Adjusted R-squared:  0.3112
## F-statistic: 14.56 on 1 and 29 DF,  p-value: 0.0006589
```

```
ggplot(data, aes(x = `Max.Wave.Height..m.`, y = `RR..m.yr.`)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
```

```
labs(title = "Linear Regression of RR (m/yr) on Max Wave Height(m)",
     x = "Max Wave Height(m)",
     y = "RR (m/yr)")
```

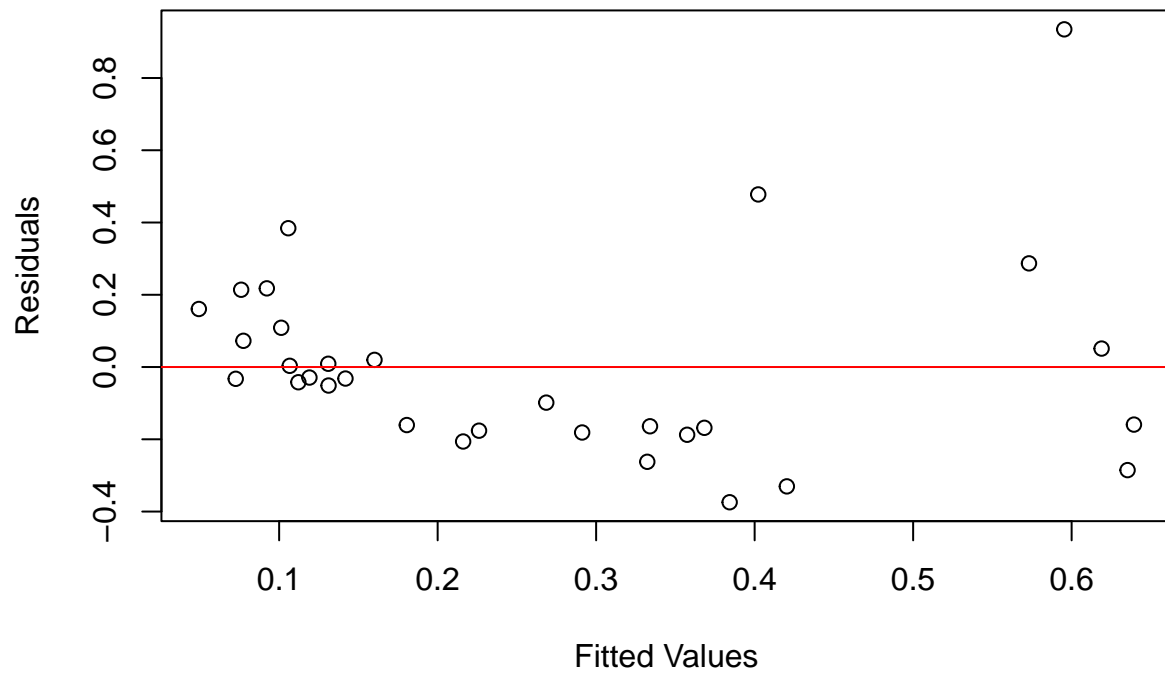
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
### check
```

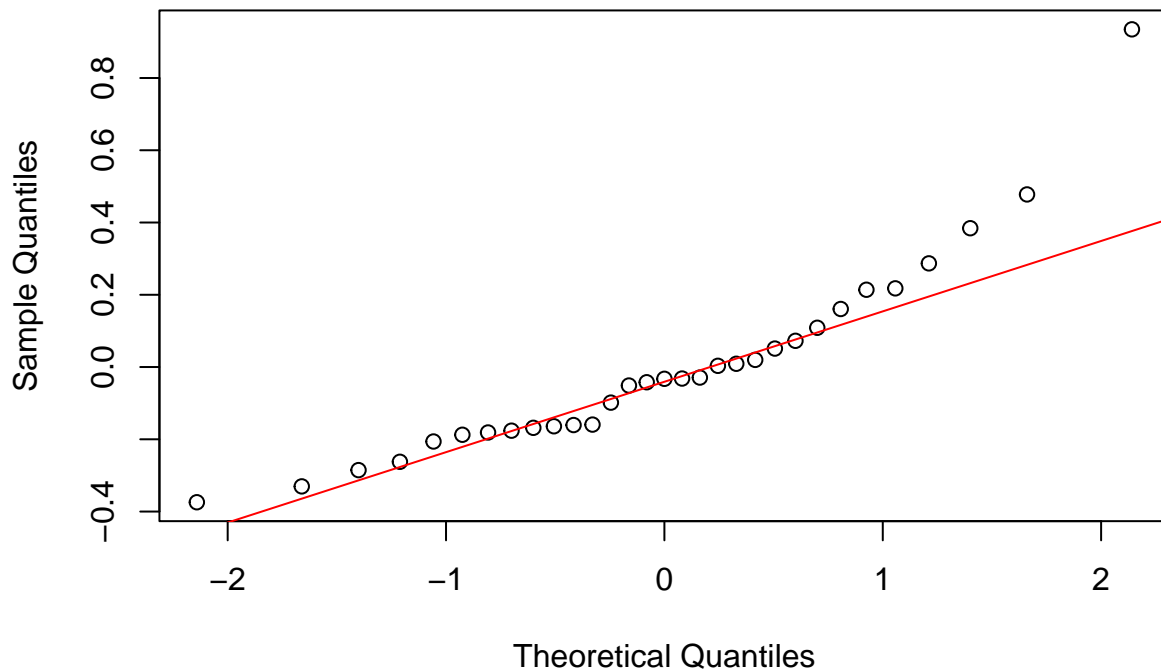
```
# Residuals vs Fitted Values Plot
plot(model3$fitted.values, model3$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```

## Residuals vs Fitted



```
# Normal Q-Q Plot for Residuals  
qqnorm(model3$residuals)  
qqline(model3$residuals, col = "red")
```

## Normal Q-Q Plot



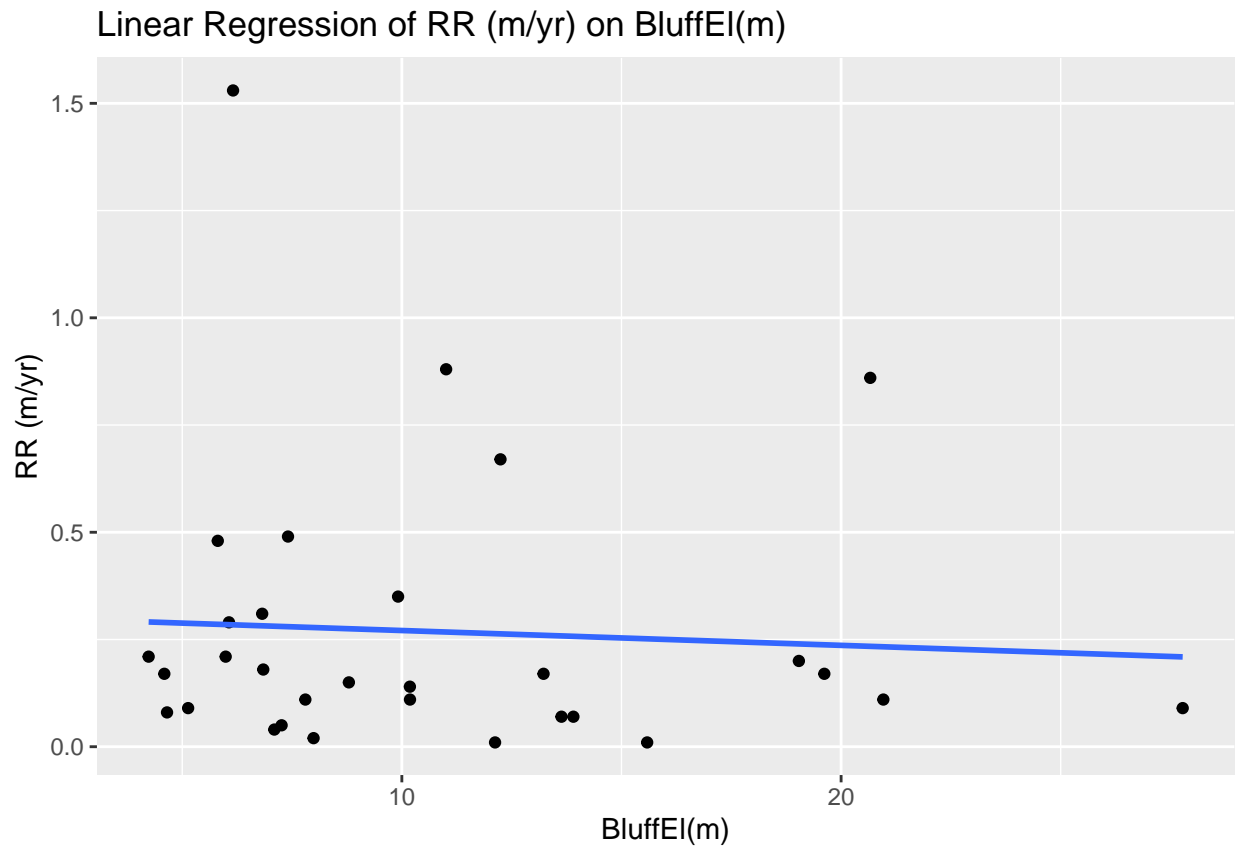
```
model4 <- lm(`RR..m.yr.` ~ `BluffEl..m.`, data = data)
summary(model4)
```

```
##
## Call:
## lm(formula = RR..m.yr. ~ BluffEl..m., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25788 -0.18788 -0.11943  0.01677  1.24578
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.305516   0.126477   2.416   0.0222 *
## BluffEl..m. -0.003459   0.010393  -0.333   0.7416
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3319 on 29 degrees of freedom
## Multiple R-squared:  0.003806,    Adjusted R-squared:  -0.03055
## F-statistic: 0.1108 on 1 and 29 DF,  p-value: 0.7416
```

```
ggplot(data, aes(x = `BluffEl..m.`, y = `RR..m.yr.`)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
```

```
labs(title = "Linear Regression of RR (m/yr) on BluffEl(m)",
     x = "BluffEl(m)",
     y = "RR (m/yr)")
```

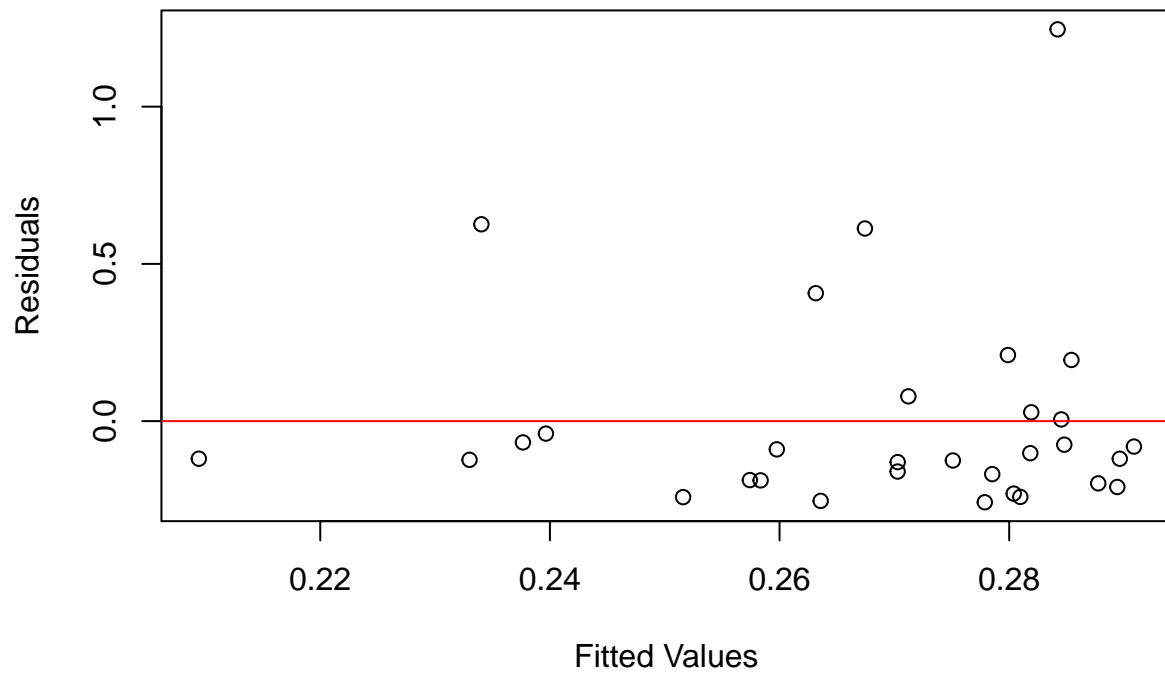
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
### check
```

```
# Residuals vs Fitted Values Plot
plot(model4$fitted.values, model4$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```

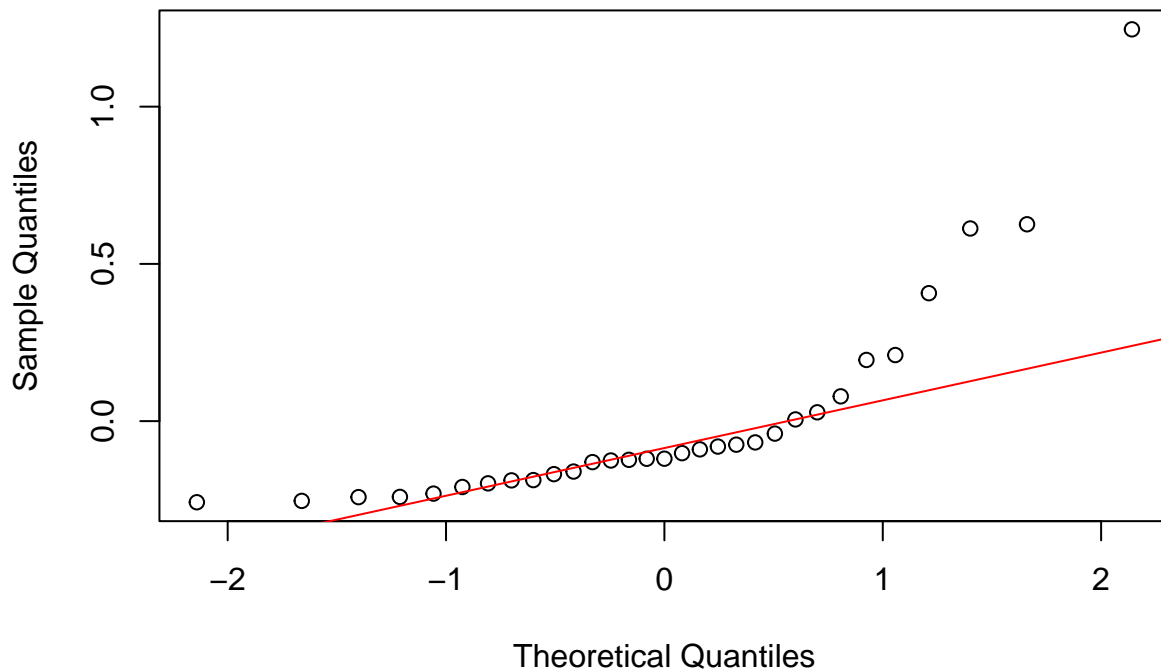
## Residuals vs Fitted



```
# Normal Q-Q Plot for Residuals  
qqnorm(model4$residuals)  
qqline(model4$residuals, col = "red")
```



## Normal Q-Q Plot



```
model5 <- lm(`RR..m.yr.` ~ `Orientation..deg.`, data = data)
summary(model5)
```

```
##
## Call:
## lm(formula = RR..m.yr. ~ Orientation..deg., data = data)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.31138	-0.17868	-0.10807	0.03997	1.22952

```
##
## Coefficients:
```

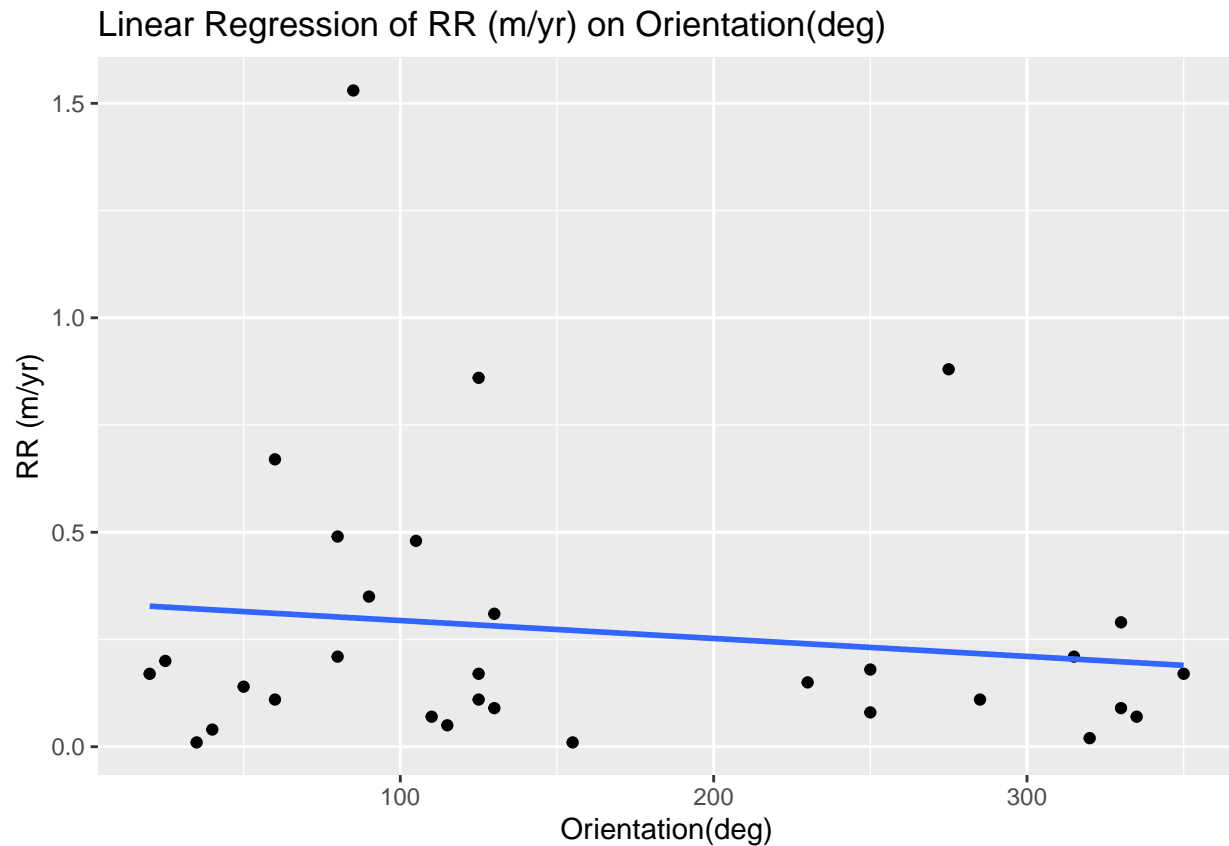
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.3360062	0.1066345	3.151	0.00376 **
Orientation..deg.	-0.0004180	0.0005485	-0.762	0.45219

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3293 on 29 degrees of freedom
## Multiple R-squared:  0.01963,    Adjusted R-squared:  -0.01417
## F-statistic: 0.5807 on 1 and 29 DF,  p-value: 0.4522
```

```
ggplot(data, aes(x = `Orientation..deg.`, y = `RR..m.yr.`)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
```

```
labs(title = "Linear Regression of RR (m/yr) on Orientation(deg)",
     x = "Orientation(deg)",
     y = "RR (m/yr)")
```

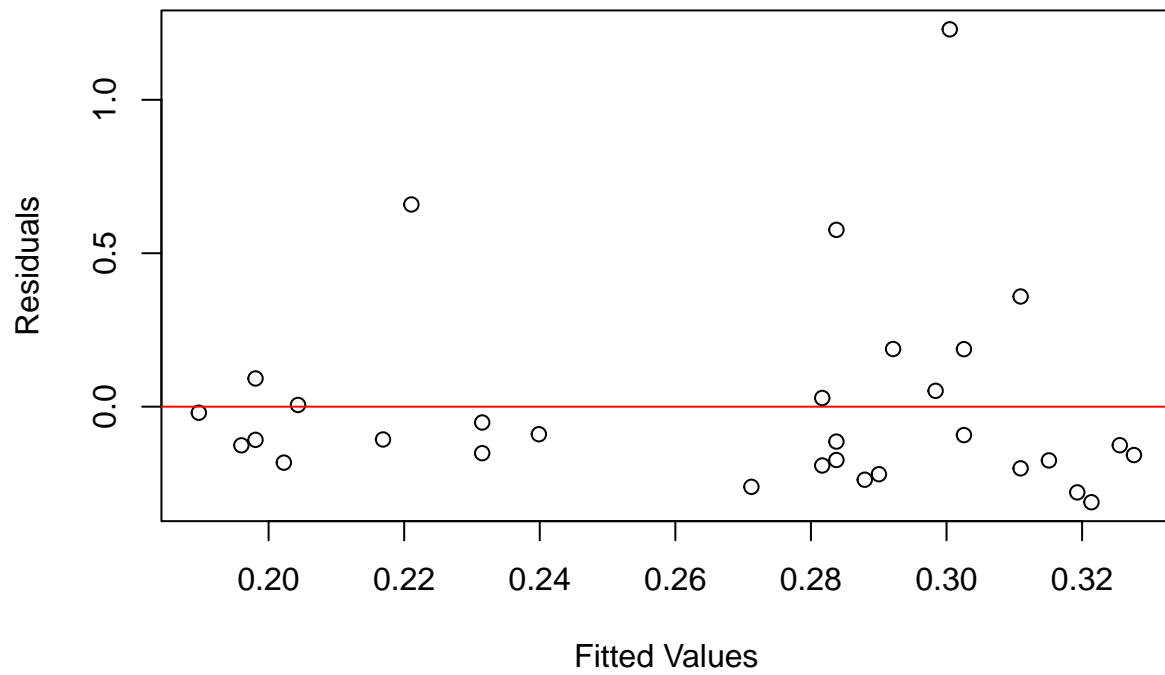
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
### check ‘
```

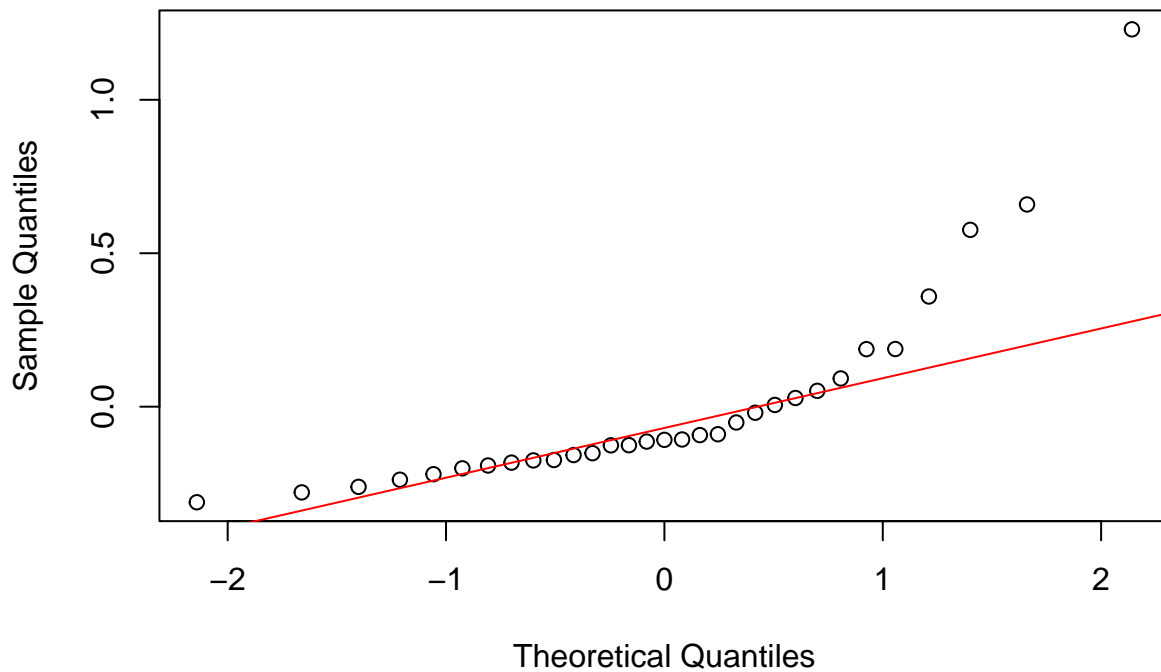
```
# Residuals vs Fitted Values Plot
plot(model5$fitted.values, model5$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```

## Residuals vs Fitted



```
# Normal Q-Q Plot for Residuals  
qqnorm(model5$residuals)  
qqline(model5$residuals, col = "red")
```

## Normal Q-Q Plot



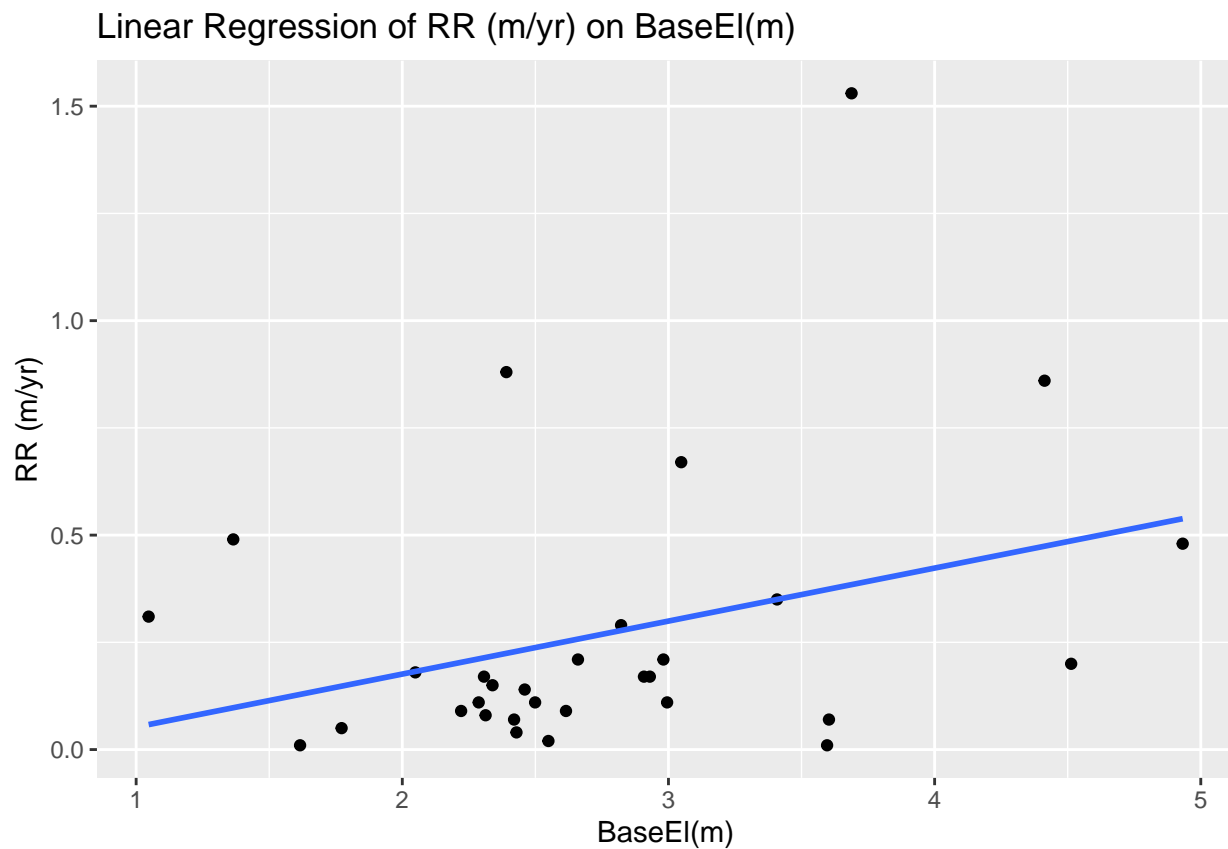
```
model6 <- lm(`RR..m.yr.` ~ `BaseEl..m.`, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = model_formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41347 -0.11112 -0.01882  0.06728  0.59329
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.237e-01  2.250e-01   0.550   0.5881
## Orientation_deg  6.972e-05  3.878e-04   0.180   0.8590
## ErosionVulnerability  1.722e-01  6.738e-02   2.556   0.0180 *
## Wave_Height_for_NNE_wind_15_m_per_s_m  8.558e-01  4.808e-01   1.780   0.0889 .
## Max_Wave_Height_m -4.089e-01  3.976e-01  -1.028   0.3150
## Mud_percent    -8.982e-03  1.799e-02  -0.499   0.6226
## BaseEl_m        5.934e-03  6.560e-02   0.090   0.9287
## BluffEl_m     -1.531e-02  9.174e-03  -1.668   0.1094
## Seawall       -1.032e-01  1.254e-01  -0.823   0.4193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.2088 on 22 degrees of freedom
## Multiple R-squared:  0.7008, Adjusted R-squared:  0.5921
## F-statistic: 6.442 on 8 and 22 DF,  p-value: 0.0002397
```

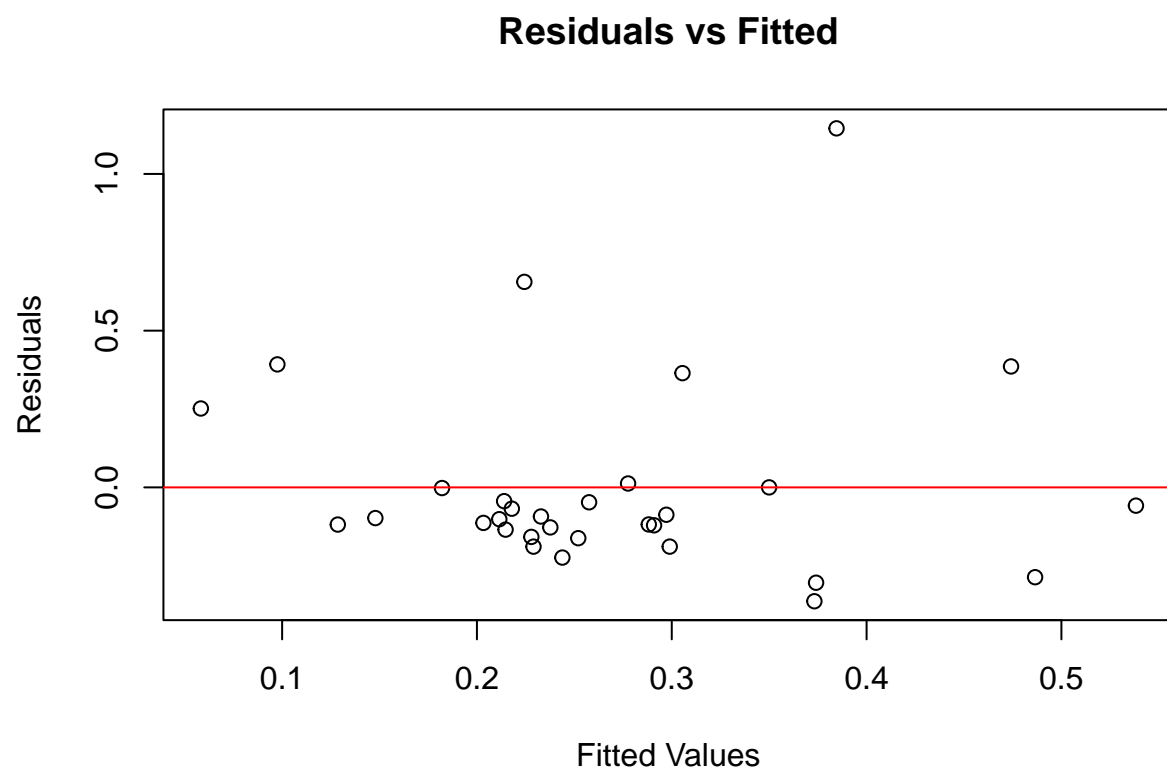
```
ggplot(data, aes(x = `BaseEl..m.`, y = `RR..m.yr.`)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Linear Regression of RR (m/yr) on BaseEl(m)",
       x = "BaseEl(m)",
       y = "RR (m/yr)")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



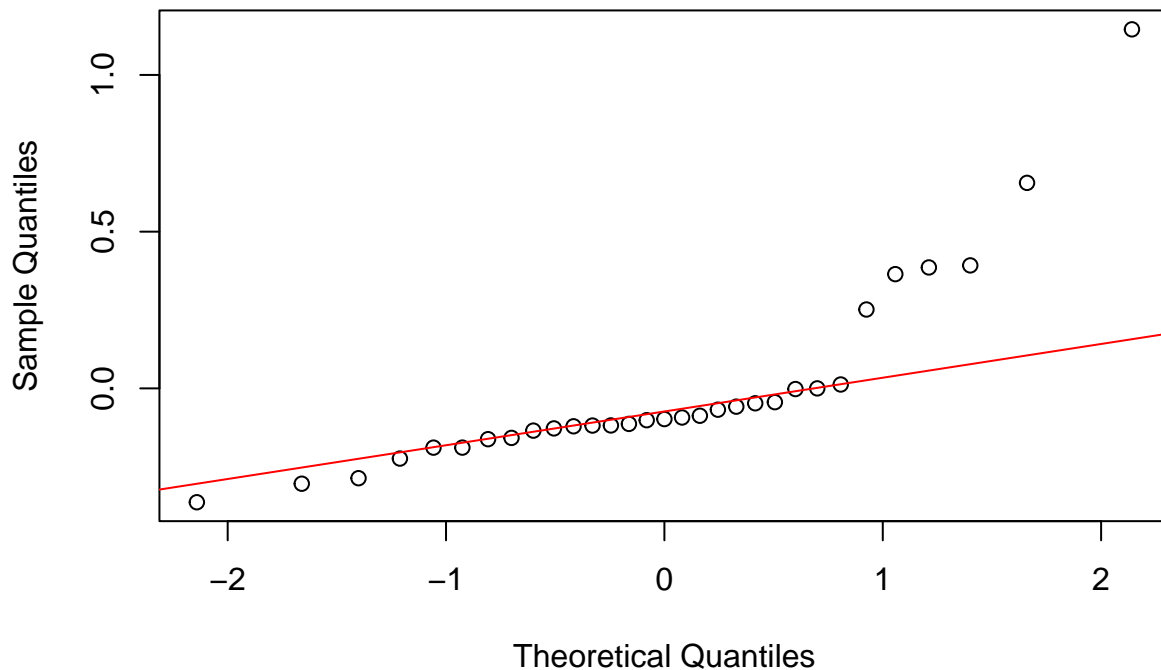
```
### check
```

```
# Residuals vs Fitted Values Plot
plot(model6$fitted.values, model6$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```



```
# Normal Q-Q Plot for Residuals  
qqnorm(model6$residuals)  
qqline(model6$residuals, col = "red")
```

## Normal Q-Q Plot

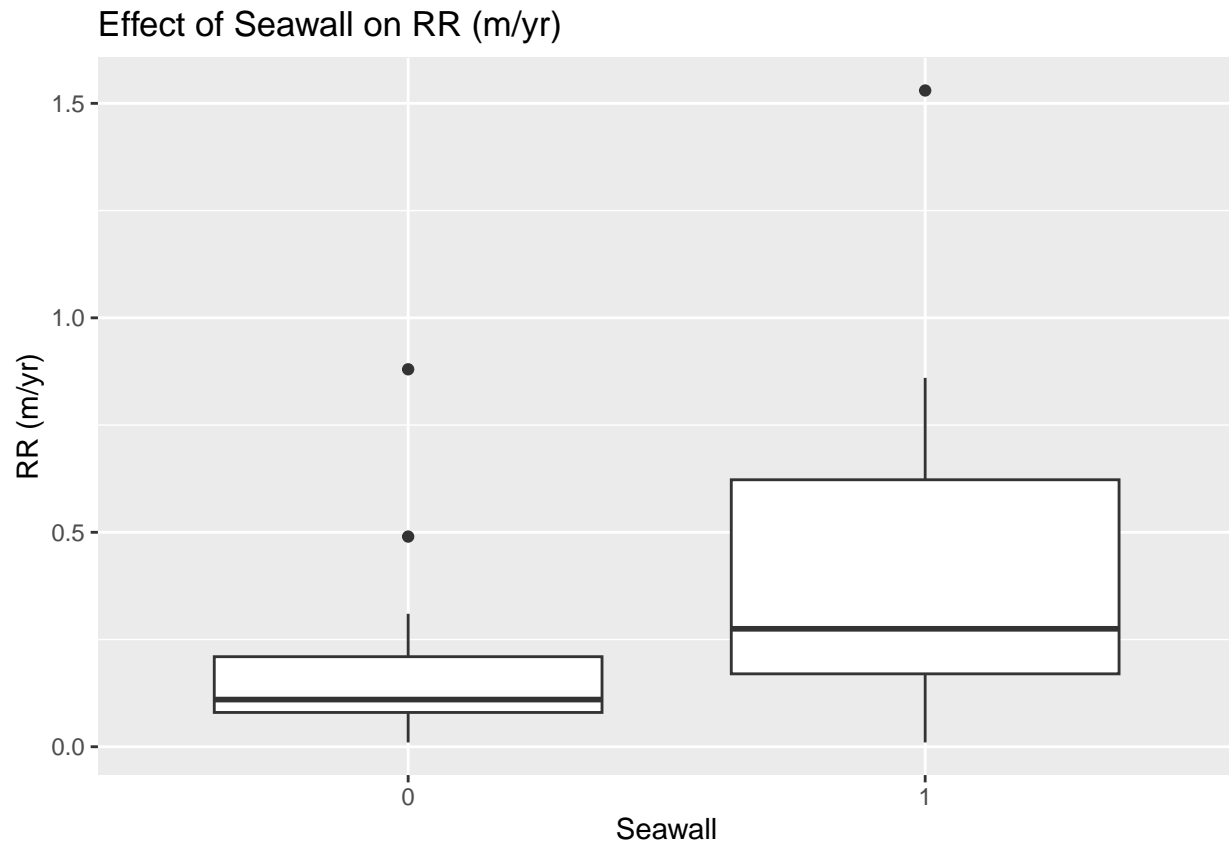


factor analysis for nominal variable

```
data$Seawall <- factor(data$Seawall)
model <- lm(`RR..m.yr.` ~ Seawall, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = RR..m.yr. ~ Seawall, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44100 -0.13643 -0.07143  0.02879  1.07900
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.18143    0.06676   2.717  0.0110 *
## Seawall1     0.26957    0.11755   2.293  0.0293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3059 on 29 degrees of freedom
## Multiple R-squared:  0.1535, Adjusted R-squared:  0.1243
## F-statistic: 5.259 on 1 and 29 DF, p-value: 0.02927
```

```
ggplot(data, aes(x = Seawall, y = `RR..m.yr.`)) +
  geom_boxplot() +
  labs(title = "Effect of Seawall on RR (m/yr)", x = "Seawall", y = "RR (m/yr)")
```



```
data$ErosionVulnerability <- factor(data$ErosionVulnerability)
model <- lm(`RR..m.yr.` ~ ErosionVulnerability, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = RR..m.yr. ~ ErosionVulnerability, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38625 -0.03333 -0.00667  0.03333  0.83375
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.07667    0.05241   1.463   0.155
## ErosionVulnerability1 0.12333    0.08886   1.388   0.176
## ErosionVulnerability2 0.61958    0.08886   6.973 1.4e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.203 on 28 degrees of freedom
```



```
## Multiple R-squared:  0.6403, Adjusted R-squared:  0.6146  
## F-statistic: 24.92 on 2 and 28 DF,  p-value: 6.072e-07
```

```
ggplot(data, aes(x = ErosionVulnerability, y = `RR..m.yr.`)) +  
  geom_boxplot() +  
  labs(title = "Effect of ErosionVulnerability on RR (m/yr)", x = "ErosionVulnerability", y = "RR (m/yr)"
```

