

## Question 2

Gifty Osei

2024-11-01

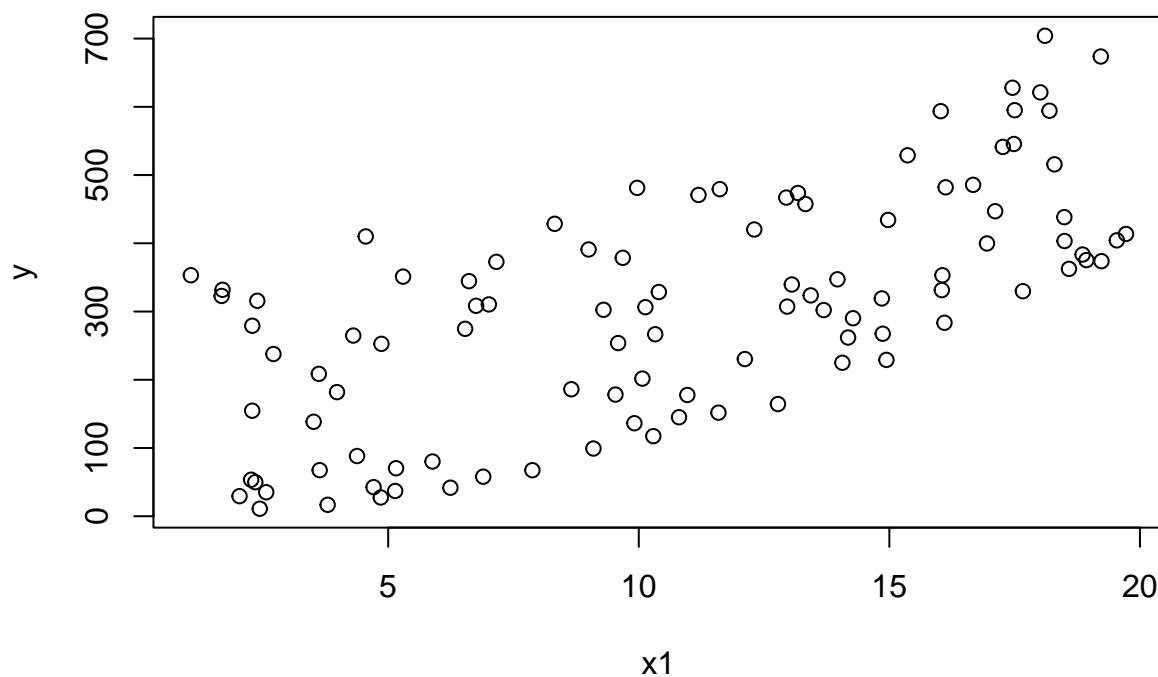
2b.

In general, a nonlinear trend can look fairly linear when data contain noise. For example, the following code plots data generated from  $y = f(x_1, x_2) = x_1^2 + x_2^2 + \epsilon$ , where  $\epsilon \sim N(0, 0.02^2)$ .

```
set.seed(390)
n <- 100
# simulate 20 values from uniform(0,1)
x1 = runif(n,1,20)
x2= runif(n,1,20)

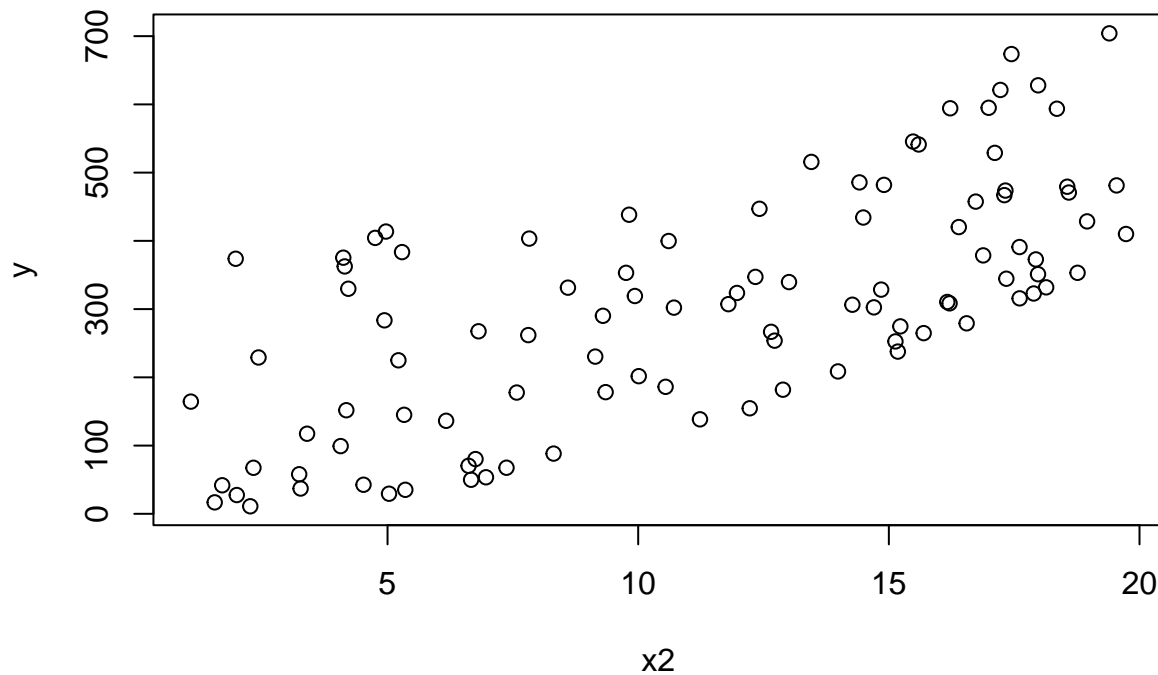
y = x1^2+x2*x2+rnorm(20,sd=0.02)
plot(y~x1, main = " Marginal plot of y vs x1", xlab = "x1", ylab = "y")
```

**Marginal plot of y vs x1**



```
plot(y~x2,main = " Marginal plot of y vs x2", xlab = "x2", ylab = "y")
```

## Marginal plot of y vs x2



We can see clearly from the marginal plots showing a linear trend but the overall model that generated the y values is nonlinear in x. We can see from the joint marginal plot that they are jointly nonlinear.

## Question 2

a

```
# simulate data
n <- 100
x <- runif(n, 1, 10)
beta_0 <- 3
beta_1 <- 2
sigma <- 3

# Generate y values based on the original model
epsilon <- rnorm(n, mean = 0, sd = sigma * x)
y <- beta_0 + beta_1 * x + epsilon

# transformations
y_prime <- y / x
x_prime <- 1 / x

# Check the variance of y_prime
var_y_prime <- var(y_prime)
print(sqrt(var_y_prime))
```

```
## [1] 3.430146
```

We can see that variance of  $y'$  is very close to the initial  $\sigma$

c

This is equivalent to OLS estimate because using the weights on the original data is like doing a  $\frac{y}{x}$  transformation. We can see from the coefficients that there is not much change.

```
# wls model
fit_wls <- lm(y ~ x, weights = 1 / x^2)

# OLS model
fit_ols_trans <- lm(y_prime ~ x_prime)

# Compare the coefficients
summary(fit_wls)$coefficients
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept)  4.089974   2.2529295  1.815402  0.0725203966
## x            2.193610   0.6439473  3.406505  0.0009555446
```

```
summary(fit_ols_trans)$coefficients
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept)  2.193610   0.6439473  3.406505  0.0009555446
## x_prime      4.089974   2.2529295  1.815402  0.0725203966
```

### Question 3

```
library(faraway)
pander::pander(head(pipeline))
```

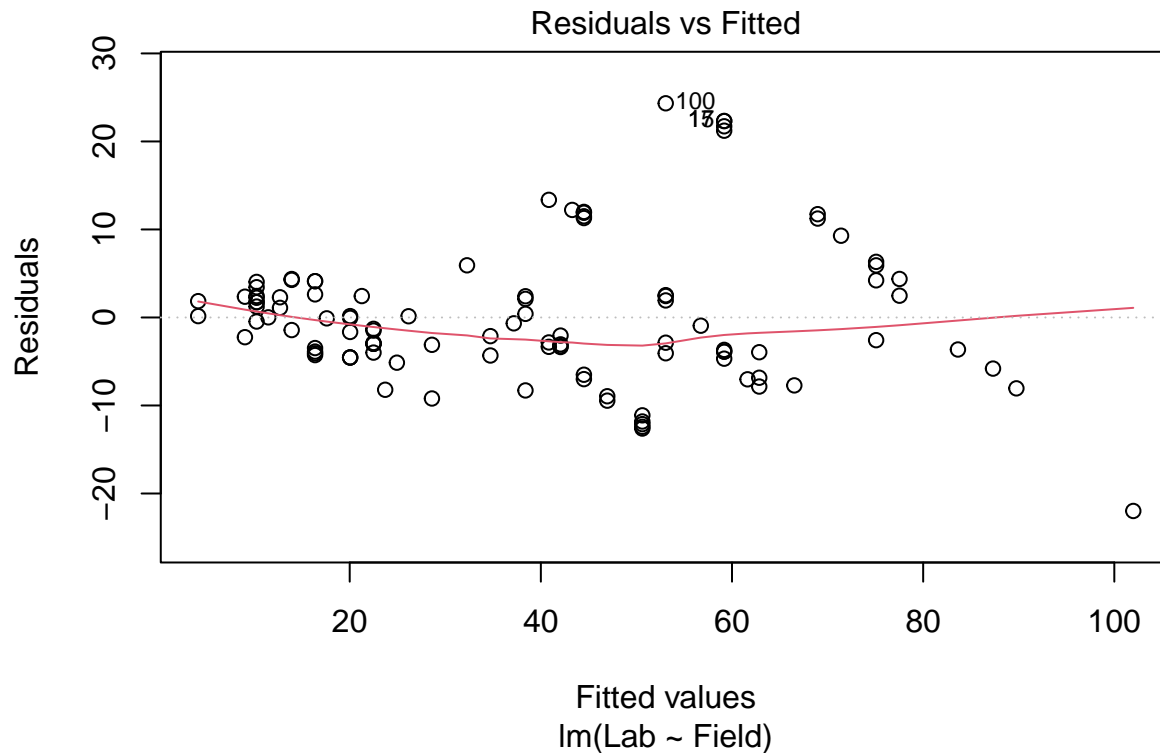
Field	Lab	Batch
18	20.2	1
38	56	1
15	12.5	1
20	21.2	1
18	15.5	1
36	39	1

a.

Fitting a linear model by regressing Lab on Field as;

```
fit_lm <- lm( Lab ~ Field, data = pipeline)

plot(fit_lm, which = 1)
```



We can see from the residual vs fitted plot that linearity assumption is checked but the residuals have a funnel like pattern. As x increases the variability increases.

b

```
i = order(pipeline$Field)
npipe = pipeline[i,]
ff = gl(12,9)[-108]
meanfield = unlist(lapply(split(npipe$Field,ff),mean))
varlab = unlist(lapply(split(npipe$Lab,ff),var))
```

```
# Remove last point
meanfield <- meanfield[-length(meanfield)]
varlab <- varlab[-length(varlab)]
```

```
# Log transform
log_meanfield <- log(meanfield)
log_varlab <- log(varlab)
```

```
# model
var_model <- lm(log_varlab ~ log_meanfield)
summary(var_model)
```

```
##
## Call:
## lm(formula = log_varlab ~ log_meanfield)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00477 -0.42268  0.05989  0.37854  0.93815
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.9352     1.0929  -1.771 0.110403
## log_meanfield    1.6707     0.3296   5.070 0.000672 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.657 on 9 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7118
## F-statistic: 25.7 on 1 and 9 DF, p-value: 0.0006723
```

```
# Extract coefficients for a0 and a1
## take exp because of the log
a0 <- exp(coef(var_model)[1])

## slope - a1
a1 <- coef(var_model)[2]

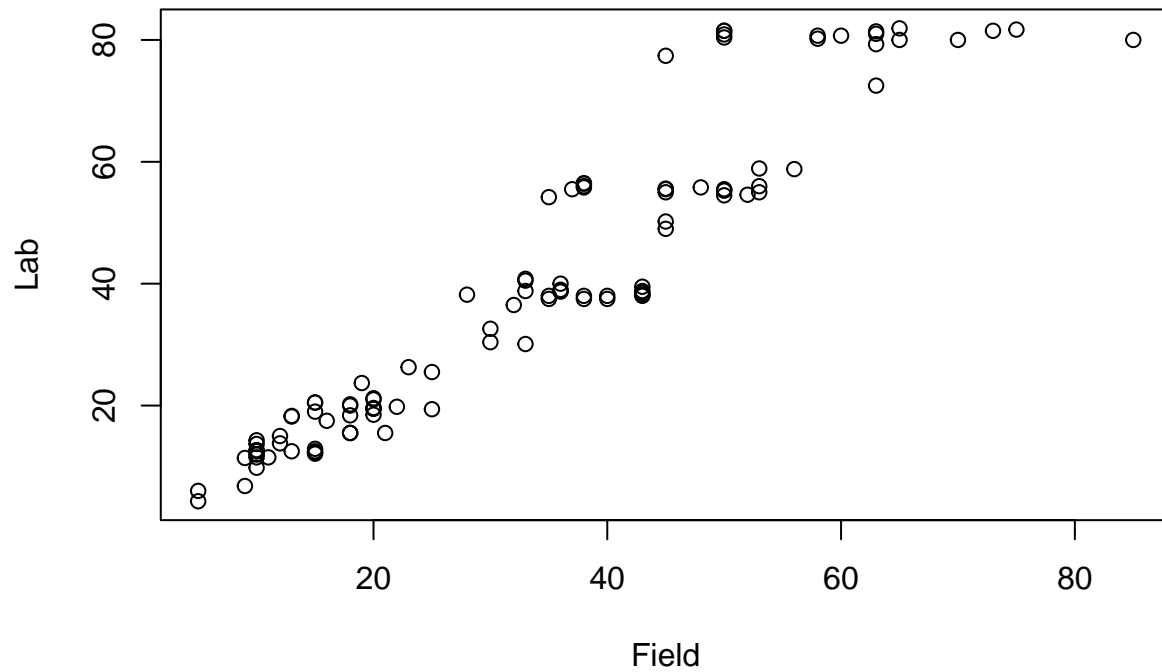
# WLS as the inverse
predicted_variance <- a0 * (pipeline$Field ^ a1)
weights <- 1 / predicted_variance

# Perform WLS regression of Lab on Field using the calculated weights
wls_model <- lm(Lab ~ Field, data = pipeline, weights = weights)
summary(wls_model)
```

```
##
## Call:
## lm(formula = Lab ~ Field, data = pipeline, weights = weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7432 -0.6719 -0.2493  0.5967  2.7275
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.05530     0.69765  -1.513   0.133
## Field        1.18963     0.03401  34.984 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9846 on 105 degrees of freedom
## Multiple R-squared:  0.921, Adjusted R-squared:  0.9202
## F-statistic: 1224 on 1 and 105 DF, p-value: < 2.2e-16
```

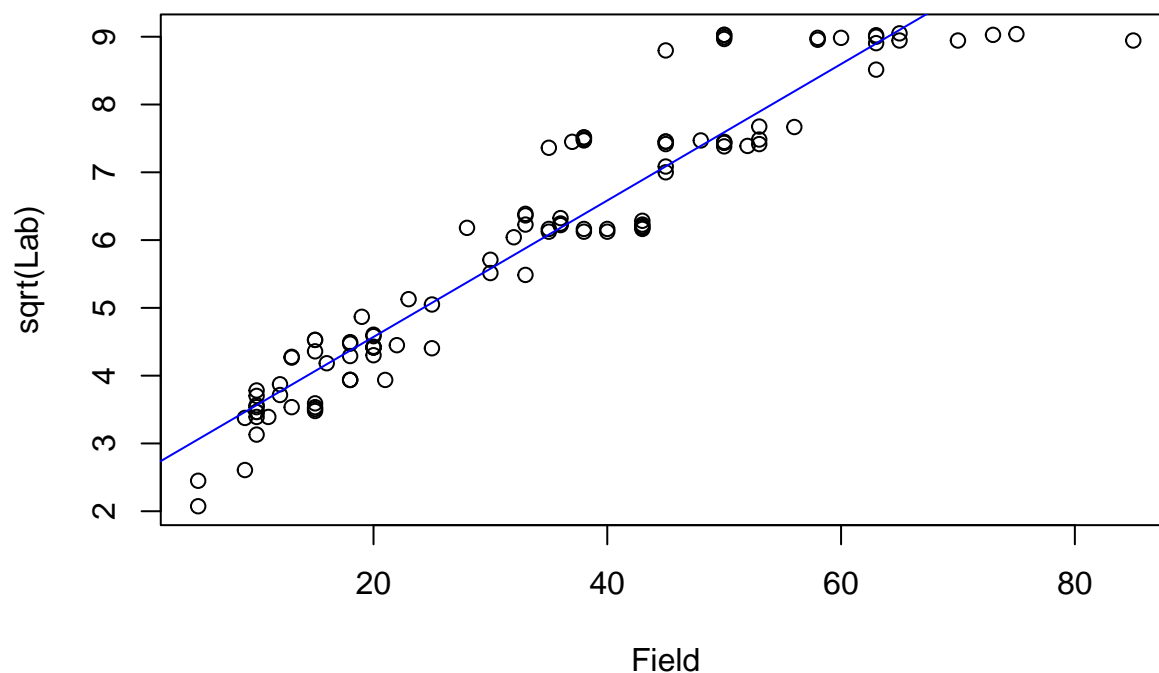
```
# Assuming the pipeline dataset is available with columns Field and Lab
# Initial Scatter Plot
plot(pipeline$Field, pipeline$Lab, main = "Original Plot of Lab vs Field",
     xlab = "Field", ylab = "Lab")
```

## Original Plot of Lab vs Field



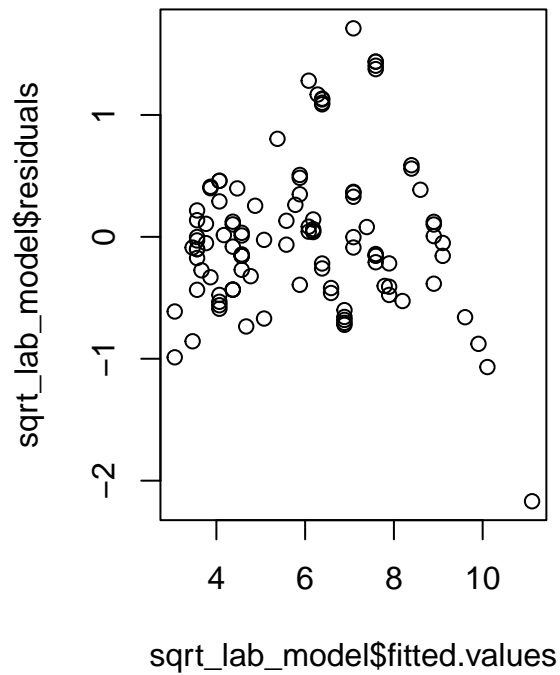
```
# 1. Square Root Transformations
# Square root of Lab vs. Field
sqrt_lab_model <- lm(sqrt(Lab) ~ Field, data = pipeline)
plot(pipeline$Field, sqrt(pipeline$Lab), main = "Square Root Transformation of Lab vs Field",
     xlab = "Field", ylab = "sqrt(Lab)")
abline(sqrt_lab_model, col = "blue")
```

## Square Root Transformation of Lab vs Field

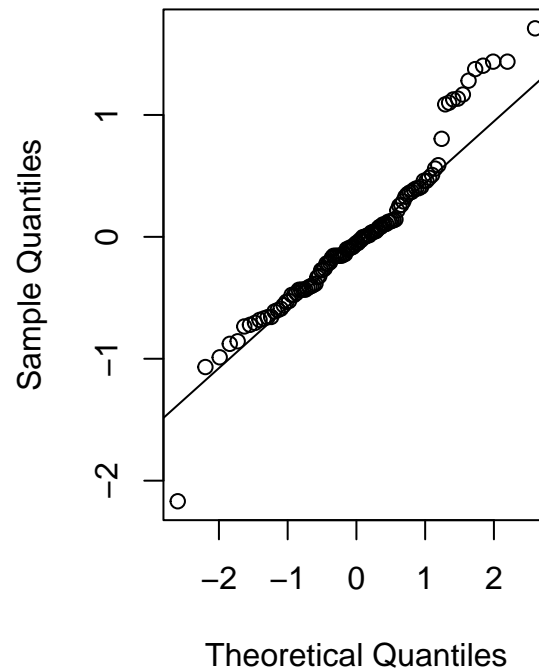


```
# Diagnostic plots for sqrt(Lab) vs Field
par(mfrow = c(1, 2))
plot(sqrt_lab_model$fitted.values, sqrt_lab_model$residuals, main = "Residuals vs Fitted (sqrt(Lab))")
qqnorm(sqrt_lab_model$residuals, main = "QQ Plot of Residuals (sqrt(Lab))")
qqline(sqrt_lab_model$residuals)
```

Residuals vs Fitted (sqrt(Lab))



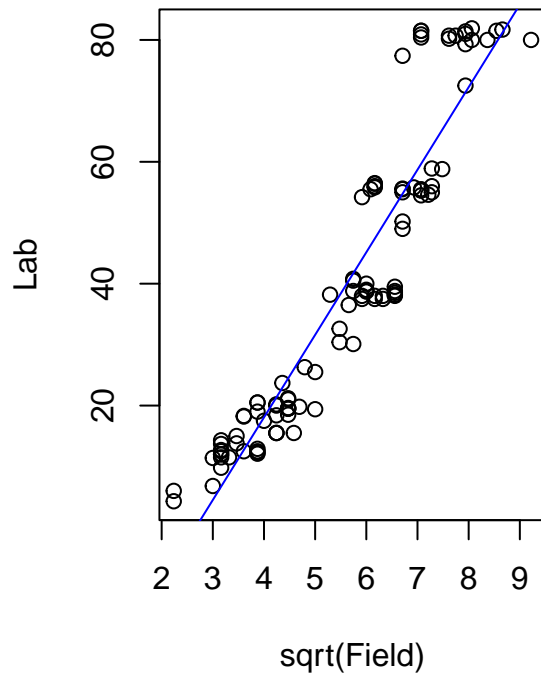
QQ Plot of Residuals (sqrt(Lab))



```
# Square root of Field vs Lab
sqrt_field_model <- lm(Lab ~ sqrt(Field), data = pipeline)
plot(sqrt(pipeline$Field), pipeline$Lab, main = "Square Root Transformation of Field vs Lab",
      xlab = "sqrt(Field)", ylab = "Lab")
abline(sqrt_field_model, col = "blue")
# Diagnostic plots for Lab vs sqrt(Field)
par(mfrow = c(1, 2))
```

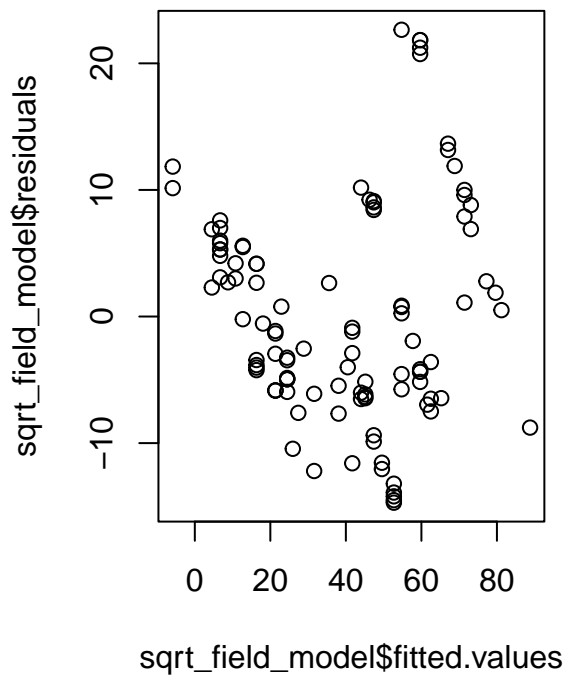


## Square Root Transformation of Field

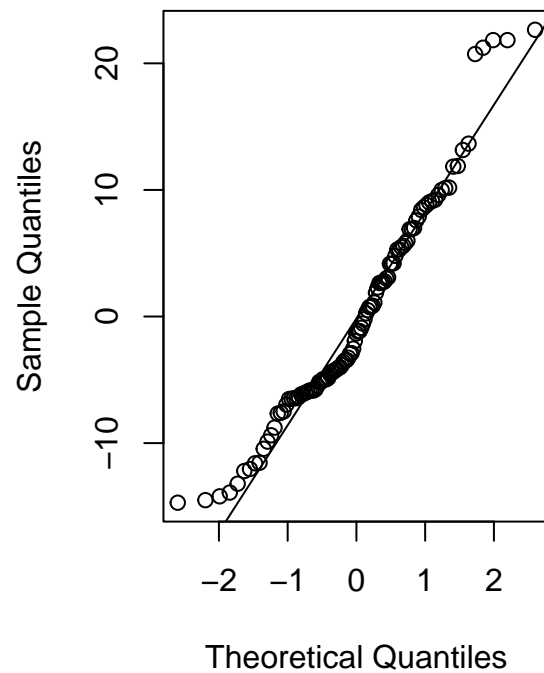


```
plot(sqrt_field_model$fitted.values, sqrt_field_model$residuals, main = "Residuals vs Fitted (sqrt(Field))",
     qqnorm(sqrt_field_model$residuals, main = "QQ Plot of Residuals (sqrt(Field))"),
     qqline(sqrt_field_model$residuals))
```

**Residuals vs Fitted (sqrt(Field))**

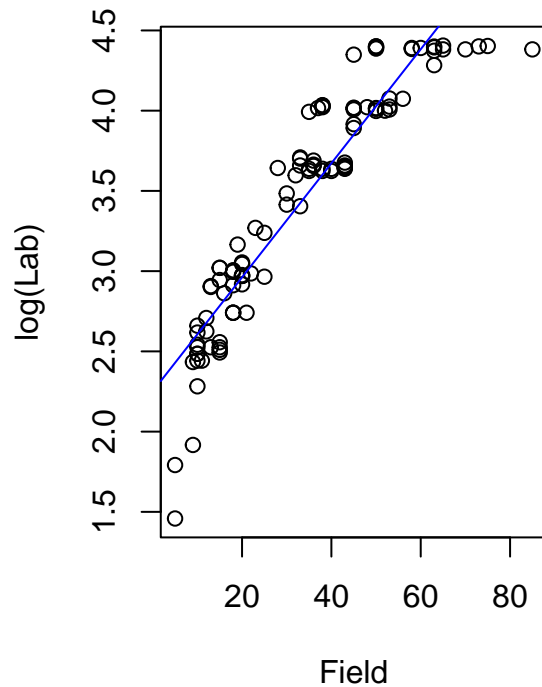


**QQ Plot of Residuals (sqrt(Field))**



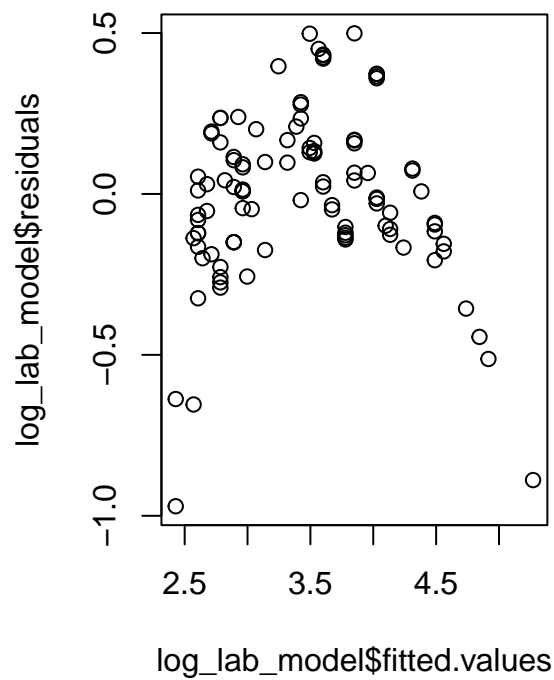
```
# 2. Log Transformations
# Log of Lab vs Field
log_lab_model <- lm(log(Lab) ~ Field, data = pipeline)
plot(pipeline$Field, log(pipeline$Lab), main = "Log Transformation of Lab vs Field",
     xlab = "Field", ylab = "log(Lab)")
abline(log_lab_model, col = "blue")
# Diagnostic plots for log(Lab) vs Field
par(mfrow = c(1, 2))
```

## Log Transformation of Lab vs Fie

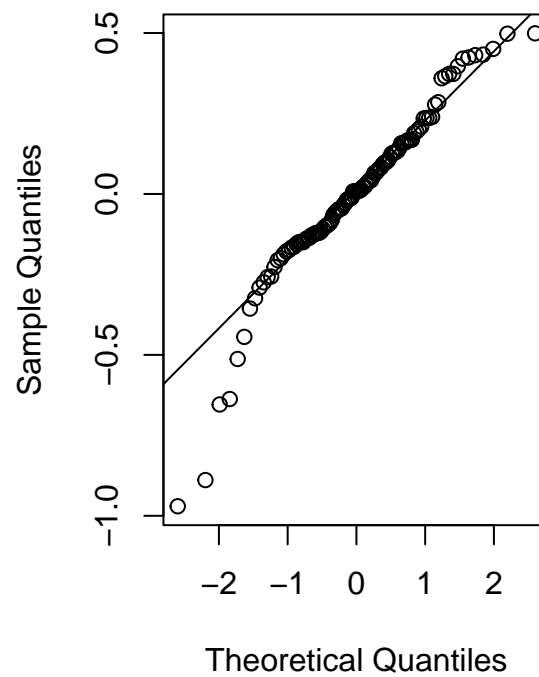


```
plot(log_lab_model$fitted.values, log_lab_model$residuals, main = "Residuals vs Fitted (log(Lab))")
qqnorm(log_lab_model$residuals, main = "QQ Plot of Residuals (log(Lab))")
qqline(log_lab_model$residuals)
```

**Residuals vs Fitted (log(Lab))**

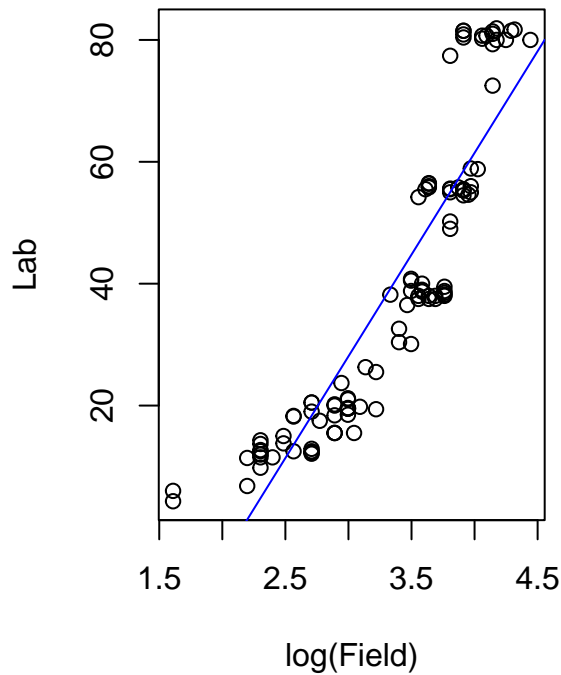


**QQ Plot of Residuals (log(Lab))**



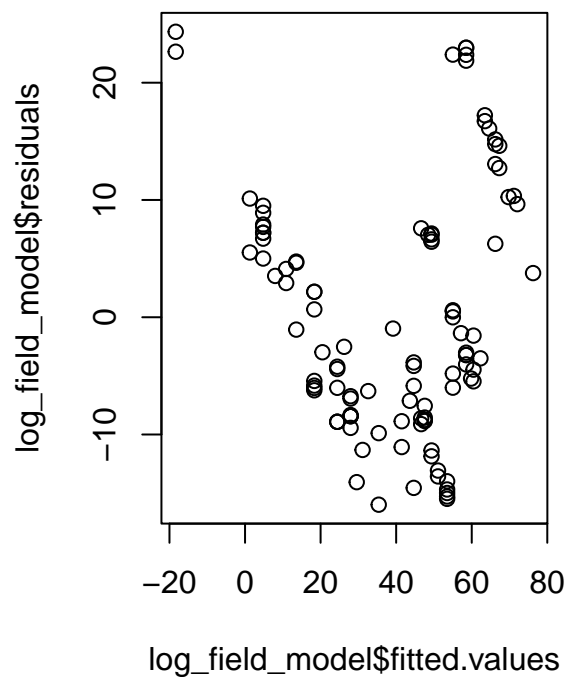
```
# Log of Field vs Lab
log_field_model <- lm(Lab ~ log(Field), data = pipeline)
plot(log(pipeline$Field), pipeline$Lab, main = "Log Transformation of Field vs Lab",
     xlab = "log(Field)", ylab = "Lab")
abline(log_field_model, col = "blue")
# Diagnostic plots for Lab vs log(Field)
par(mfrow = c(1, 2))
```

## Log Transformation of Field vs Lab

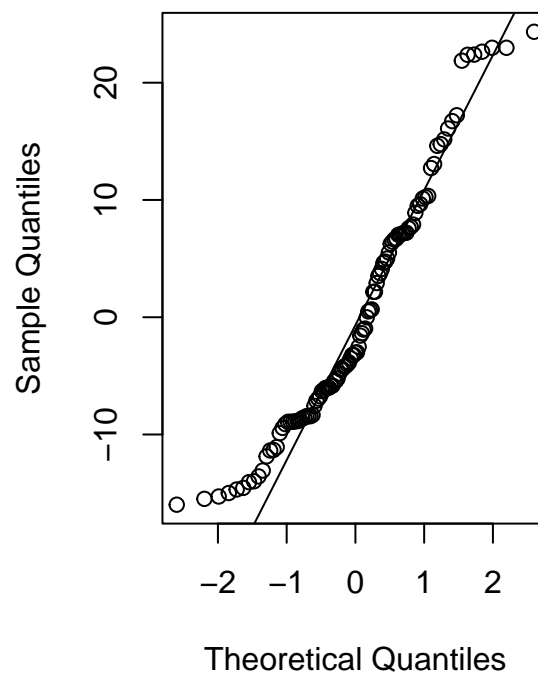


```
plot(log_field_model$fitted.values, log_field_model$residuals, main = "Residuals vs Fitted (log(Field))")
qqnorm(log_field_model$residuals, main = "QQ Plot of Residuals (log(Field))")
qqline(log_field_model$residuals)
```

### Residuals vs Fitted (log(Field))



### QQ Plot of Residuals (log(Field))

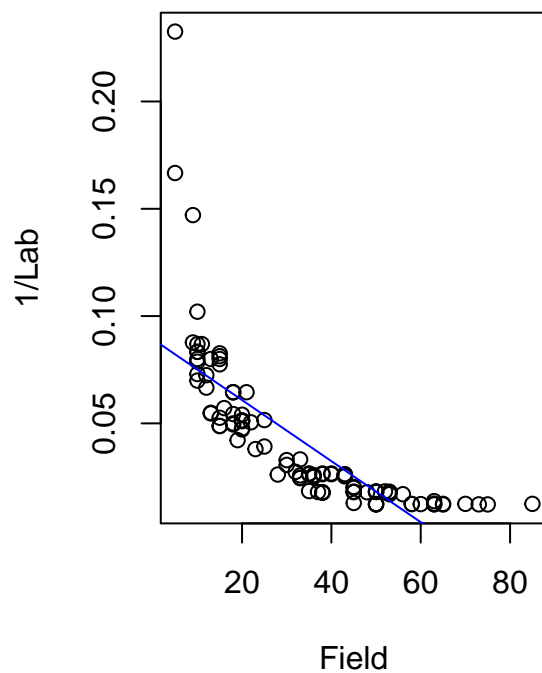


```

# 3. Inverse Transformations
# Inverse of Lab vs Field
inv_lab_model <- lm(I(1/Lab) ~ Field, data = pipeline)
plot(pipeline$Field, 1/pipeline$Lab, main = "Inverse Transformation of Lab vs Field",
     xlab = "Field", ylab = "1/Lab")
abline(inv_lab_model, col = "blue")
# Diagnostic plots for 1/Lab vs Field
par(mfrow = c(1, 2))

```

## Inverse Transformation of Lab vs F

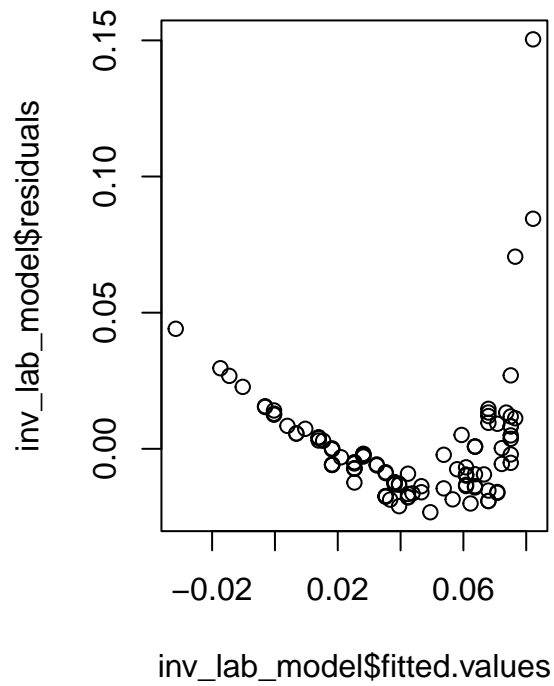


```

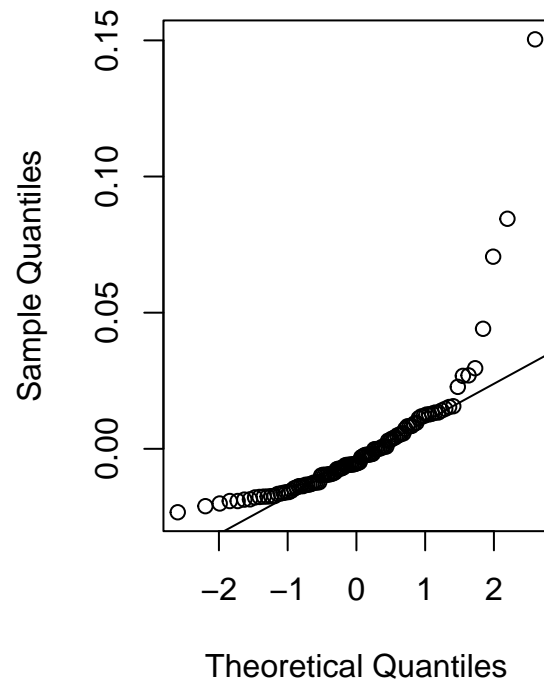
plot(inv_lab_model$fitted.values, inv_lab_model$residuals, main = "Residuals vs Fitted (1/Lab)")
qqnorm(inv_lab_model$residuals, main = "QQ Plot of Residuals (1/Lab)")
qqline(inv_lab_model$residuals)

```

**Residuals vs Fitted (1/Lab)**

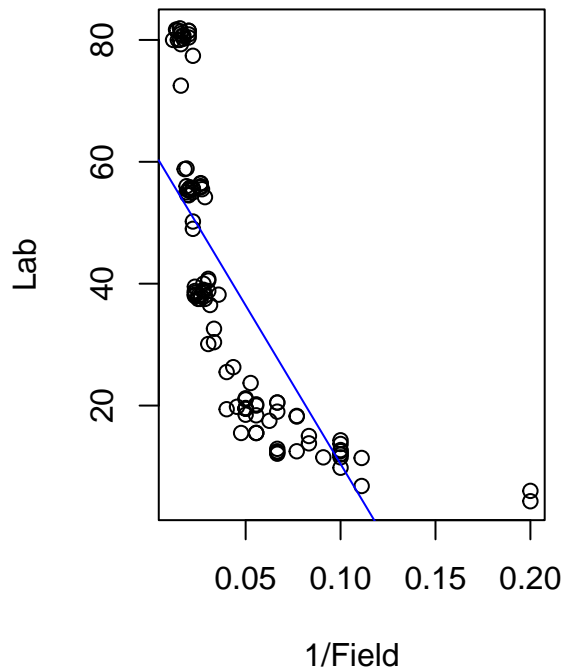


**QQ Plot of Residuals (1/Lab)**



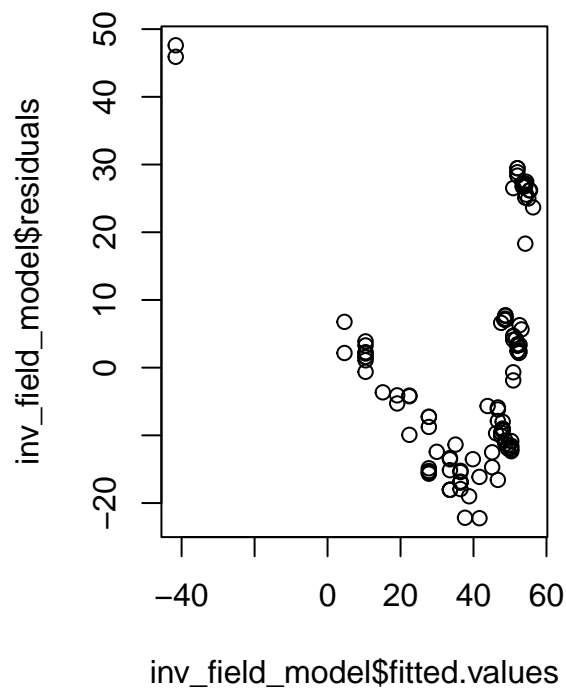
```
# Inverse of Field vs Lab
inv_field_model <- lm(Lab ~ I(1/Field), data = pipeline)
plot(1/pipeline$Field, pipeline$Lab, main = "Inverse Transformation of Field vs Lab",
     xlab = "1/Field", ylab = "Lab")
abline(inv_field_model, col = "blue")
# Diagnostic plots for Lab vs 1/Field
par(mfrow = c(1, 2))
```

## Inverse Transformation of Field vs

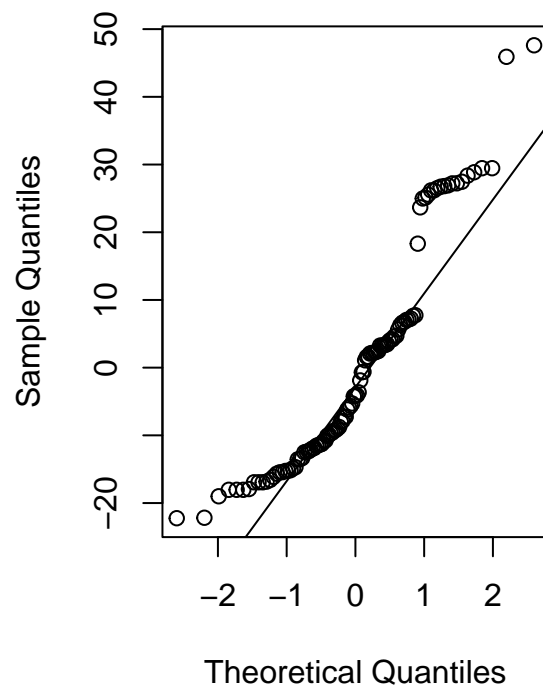


```
plot(inv_field_model$fitted.values, inv_field_model$residuals, main = "Residuals vs Fitted (1/Field)")
qqnorm(inv_field_model$residuals, main = "QQ Plot of Residuals (1/Field)")
qqline(inv_field_model$residuals)
```

**Residuals vs Fitted (1/Field)**



**QQ Plot of Residuals (1/Field)**



```
# Reset plotting layout  
par(mfrow = c(1, 1))
```