All\_Codes\_Fluazinam

Jajati Mandal

2024-01-08

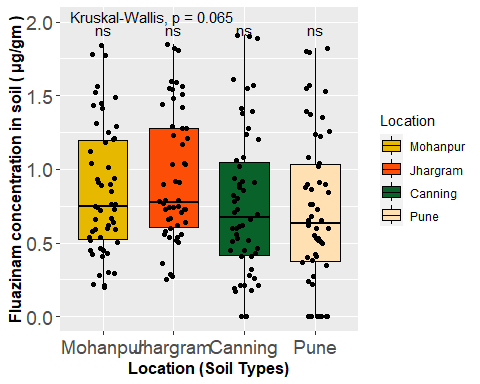
library("readxl")  
library("ggpubr")

## Loading required package: ggplot2

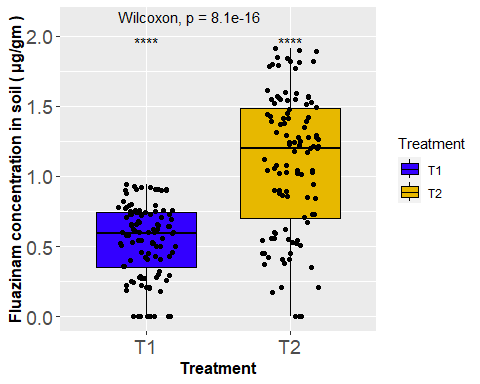
###SOIL STUDIES

Flua <- read\_excel("Flua.xlsx")  
Flua <- read\_excel("Flua.xlsx",sheet=11)  
  
A<-ggboxplot(Flua, x = "Location", y = "Conc", fill = "Location",add = "jitter",  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 xlab="Location (Soil Types)",ylab = "Fluazinam concentration in soil ( μg/gm )")+  
 theme\_gray()+  
 stat\_compare\_means(method = "kruskal", label.y = 2.0)+   
 stat\_compare\_means(label = "p.signif", method = "wilcox.test",   
 ref.group = ".all.")+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
  
  
B<-ggboxplot(Flua, x = "Treatment", y = "Conc", fill = "Treatment",add = "jitter",  
 palette = c("#3300FF","#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#C4D8F3"),  
 xlab="Treatment",ylab = "Fluazinam concentration in soil ( μg/gm )")+  
 theme\_gray()+  
 stat\_compare\_means(method = "wilcox", label.y = 2.10)+   
 stat\_compare\_means(label = "p.signif", method = "wilcox.test",   
 ref.group = ".all.")+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))

A

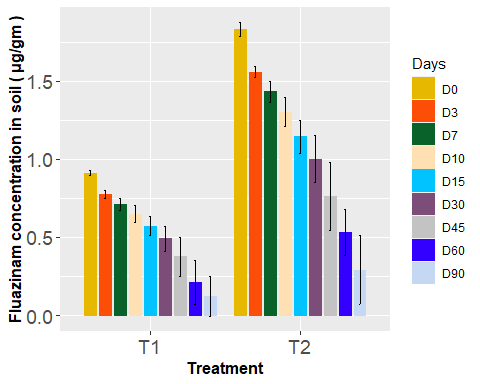


B

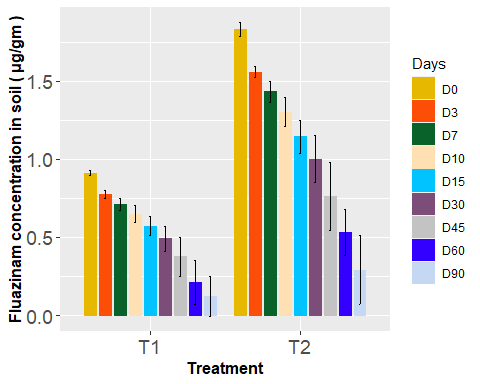


days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")  
  
Flua$Days <- factor(Flua$Days, levels = days\_order)  
  
C<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in soil ( μg/gm )",  
 add = "mean\_sd",add.params = list(color="black"),  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
  
  
days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")  
  
Flua$Days <- factor(Flua$Days, levels = days\_order)  
  
D<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in soil ( μg/gm )",  
 add = "mean\_sd",add.params = list(color="black"),  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
  
  
days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")  
  
Flua$Days <- factor(Flua$Days, levels = days\_order)  
  
E<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in soil ( μg/gm )",  
 add = "mean\_sd",add.params = list(color="black"),  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
  
days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")  
  
Flua$Days <- factor(Flua$Days, levels = days\_order)  
  
F<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in soil ( μg/gm )",  
 add = "mean\_sd",add.params = list(color="black"),  
 palette = c("#E7B800","#FFE0B3","#3300FF","#FC4E07","#09622A","#00C3FF",  
 "#7C4D79","#C3C3C3","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))

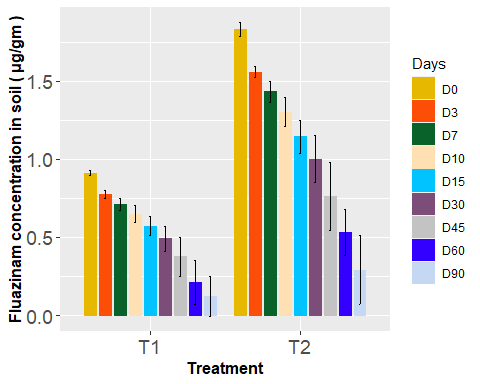
C



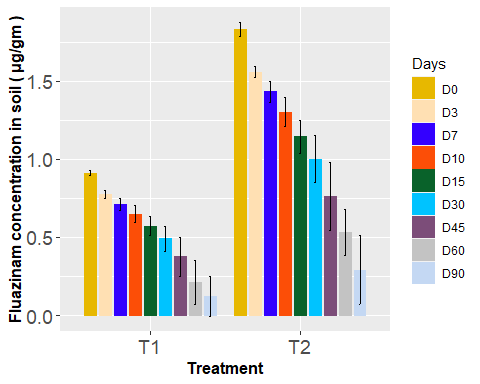
D



E

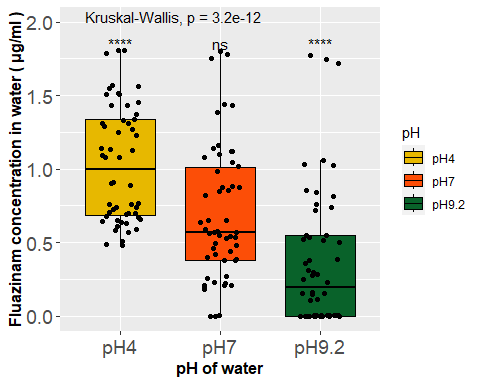


F

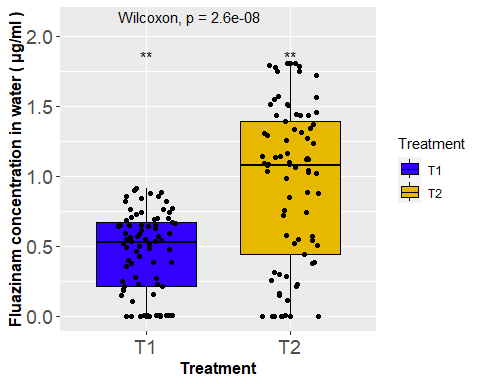


###WATER STUDIES

Flua <- read\_excel("Flua.xlsx")  
Flua <- read\_excel("Flua.xlsx",sheet=12)  
  
A<-ggboxplot(Flua, x = "pH", y = "Conc", fill = "pH",add = "jitter",  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 xlab="pH of water",ylab = "Fluazinam concentration in water ( μg/ml )")+  
 theme\_gray()+  
 stat\_compare\_means(method = "kruskal", label.y = 2.0)+   
 stat\_compare\_means(label = "p.signif", method = "wilcox.test",   
 ref.group = ".all.")+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
  
B<-ggboxplot(Flua, x = "Treatment", y = "Conc", fill = "Treatment",add = "jitter",  
 palette = c("#3300FF","#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#C4D8F3"),  
 xlab="Treatment",ylab = "Fluazinam concentration in water ( μg/ml )")+  
 theme\_gray()+  
 stat\_compare\_means(method = "wilcox", label.y = 2.10)+   
 stat\_compare\_means(label = "p.signif", method = "wilcox.test",   
 ref.group = ".all.")+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
A



B

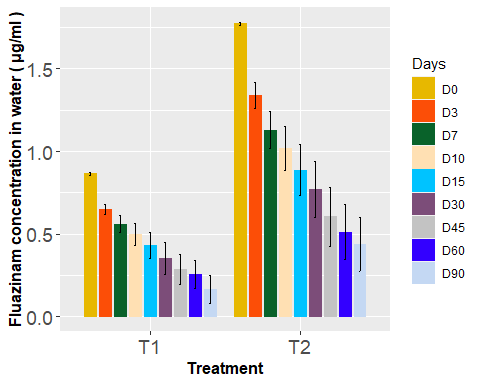


days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")  
  
Flua$Days <- factor(Flua$Days, levels = days\_order)  
  
C<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in water ( μg/ml )",  
 add = "mean\_se",add.params = list(color="black"),  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
  
days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")  
  
Flua$Days <- factor(Flua$Days, levels = days\_order)  
  
D<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in water ( μg/ml )",  
 add = "mean\_se",add.params = list(color="black"),  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))

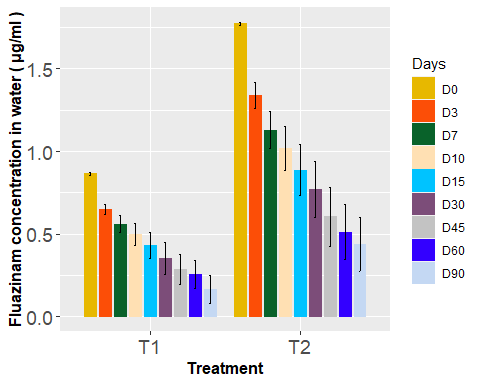
days\_order <- c("D0", "D3", "D7", "D10", "D15", "D30", "D45", "D60", "D90")

Flua$Days <- factor(Flua$Days, levels = days\_order)

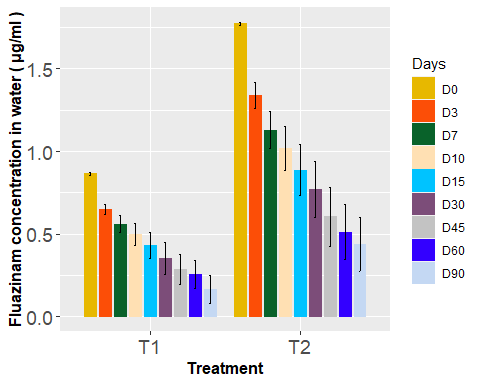
E<-ggbarplot(Flua, x = "Treatment", y = "Conc",  
 fill="Days", color = "Days",  
 xlab="Treatment",ylab = "Fluazinam concentration in water ( μg/ml )",  
 add = "mean\_se",add.params = list(color="black"),  
 palette = c("#E7B800","#FC4E07","#09622A","#FFE0B3","#00C3FF",  
 "#7C4D79","#C3C3C3","#3300FF","#C4D8F3"),  
 position = position\_dodge(0.9))+theme\_gray()+  
 theme(axis.text=element\_text(size=14),  
 axis.title=element\_text(size=12,face="bold"))  
C



D



E



SOIL STUDIES

#####################Dissipation Kinetics (First Order) T1 ##################

# Dissipation Kinetics (First Order) MOHANPUR  
library("readxl")  
Flua <- read\_excel("Flua.xlsx")  
Flua <- read\_excel("Flua.xlsx",sheet=1)  
  
t <- Flua$t  
c <- Flua$mqt  
  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.102208 -0.078996 0.006478 0.034456 0.122962   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.2136160 0.0401164 -5.325 0.00109 \*\*   
## t -0.0153675 0.0009824 -15.643 1.06e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.08506 on 7 degrees of freedom  
## Multiple R-squared: 0.9722, Adjusted R-squared: 0.9682   
## F-statistic: 244.7 on 1 and 7 DF, p-value: 1.055e-06

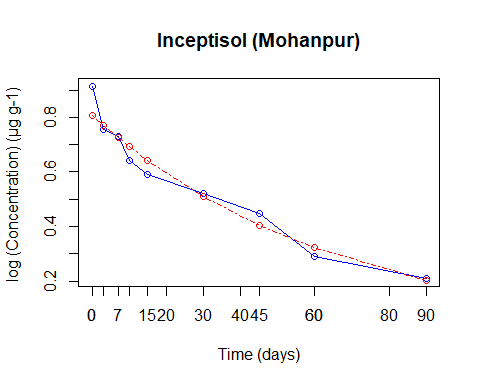
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.01536751

cat("Half-Life:", half\_life, "\n")

## Half-Life: 45.10471

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Inceptisol (Mohanpur)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order)PUNE  
Flua <- read\_excel("Flua.xlsx",sheet=2)  
  
t <- Flua$t  
c <- Flua$pqt  
  
# Fit a first-order kinetic model and remove infinite values  
kinetic\_model <- lm(log(c) ~ t)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t)  
##   
## Residuals:  
## 1 2 3 4 5 6 7   
## 0.08602 -0.01318 -0.03513 -0.03999 -0.06822 0.09759 -0.02709   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.195092 0.038923 -5.012 0.00406 \*\*   
## t -0.026470 0.001791 -14.783 2.56e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07116 on 5 degrees of freedom  
## Multiple R-squared: 0.9776, Adjusted R-squared: 0.9732   
## F-statistic: 218.5 on 1 and 5 DF, p-value: 2.561e-05

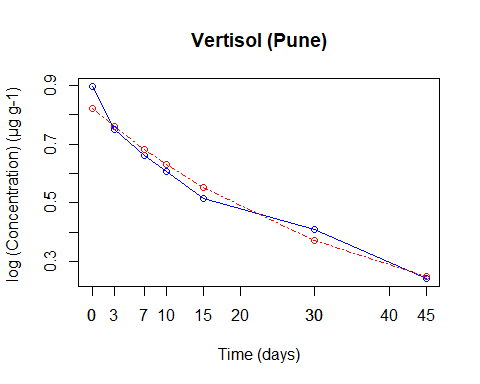
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.02646981

cat("Half-Life:", half\_life, "\n")

## Half-Life: 26.18633

# Plot the actual data  
plot(t,c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Vertisol (Pune)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)),col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) CANNING  
Flua <- read\_excel("Flua.xlsx",sheet=3)  
  
t <- Flua$t  
c <- Flua$cqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.119193 -0.020889 0.007285 0.030752 0.088022   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.167787 0.036575 -4.587 0.00374 \*\*   
## t -0.024463 0.001245 -19.654 1.12e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07145 on 6 degrees of freedom  
## Multiple R-squared: 0.9847, Adjusted R-squared: 0.9822   
## F-statistic: 386.3 on 1 and 6 DF, p-value: 1.125e-06

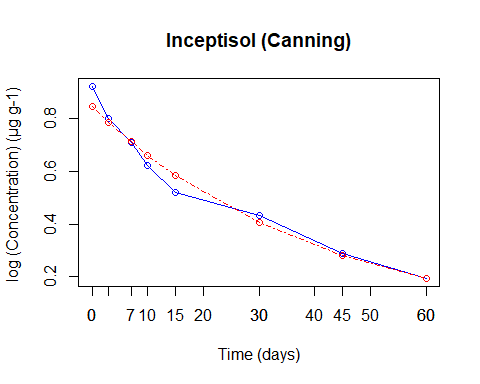
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.02446311

cat("Half-Life:", half\_life, "\n")

## Half-Life: 28.33439

# Plot the actual data  
plot(t,c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Inceptisol (Canning)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) JHARGRAM  
Flua <- read\_excel("Flua.xlsx",sheet=1)  
  
t <- Flua$t  
c <- Flua$jqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.08904 -0.03325 -0.01886 0.03973 0.11298   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.1684970 0.0328173 -5.134 0.00135 \*\*   
## t -0.0127353 0.0008036 -15.847 9.66e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06958 on 7 degrees of freedom  
## Multiple R-squared: 0.9729, Adjusted R-squared: 0.969   
## F-statistic: 251.1 on 1 and 7 DF, p-value: 9.66e-07

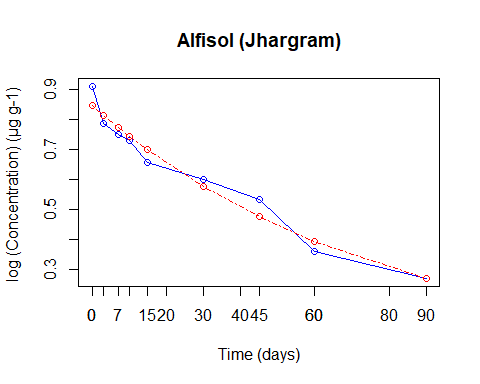
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.0127353

cat("Half-Life:", half\_life, "\n")

## Half-Life: 54.42722

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Alfisol (Jhargram)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



#####################Dissipation Kinetics (First Order) T2 ##################

# Dissipation Kinetics (First Order) MOHANPUR  
  
Flua <- read\_excel("Flua.xlsx",sheet=4)  
  
t <- Flua$t  
c <- Flua$mqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.085196 -0.070461 -0.009268 0.030329 0.110723   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.475210 0.035486 13.39 3.04e-06 \*\*\*  
## t -0.015014 0.000869 -17.28 5.35e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07524 on 7 degrees of freedom  
## Multiple R-squared: 0.9771, Adjusted R-squared: 0.9738   
## F-statistic: 298.5 on 1 and 7 DF, p-value: 5.346e-07

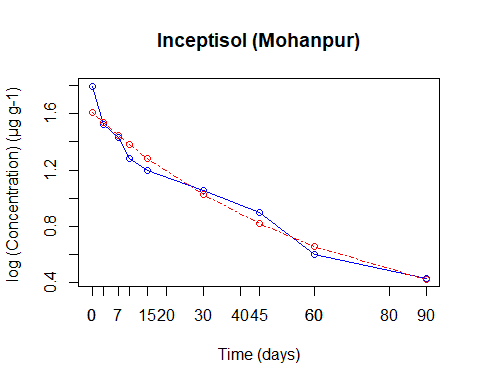
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.01501399

cat("Half-Life:", half\_life, "\n")

## Half-Life: 46.16674

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Inceptisol (Mohanpur)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order)PUNE  
Flua <- read\_excel("Flua.xlsx",sheet=5)  
  
t <- Flua$t  
c <- Flua$pqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.08943 -0.02759 -0.01049 0.02022 0.09932   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.512500 0.034103 15.03 5.47e-06 \*\*\*  
## t -0.025164 0.001161 -21.68 6.28e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06662 on 6 degrees of freedom  
## Multiple R-squared: 0.9874, Adjusted R-squared: 0.9853   
## F-statistic: 470.2 on 1 and 6 DF, p-value: 6.282e-07

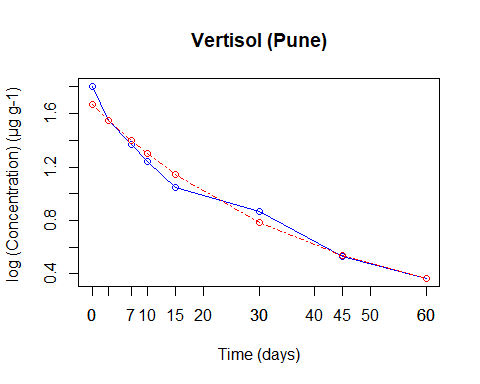
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.02516386

cat("Half-Life:", half\_life, "\n")

## Half-Life: 27.54534

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Vertisol (Pune)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) CANNING  
Flua <- read\_excel("Flua.xlsx",sheet=4)  
  
t <- Flua$t  
c <- Flua$cqt  
  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.117584 -0.033227 0.004337 0.053113 0.120177   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.5216767 0.0363359 14.36 1.89e-06 \*\*\*  
## t -0.0234966 0.0008898 -26.41 2.86e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07704 on 7 degrees of freedom  
## Multiple R-squared: 0.9901, Adjusted R-squared: 0.9886   
## F-statistic: 697.3 on 1 and 7 DF, p-value: 2.86e-08

# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

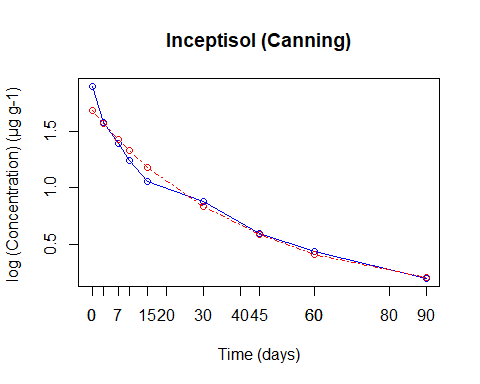
## Rate Constant: 0.0234966

cat("Half-Life:", half\_life, "\n")

## Half-Life: 29.49989

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Inceptisol (Canning)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)

# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) JHARGRAM  
Flua <- read\_excel("Flua.xlsx",sheet=4)  
  
t <- Flua$t  
c <- Flua$jqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.082570 -0.026860 -0.007132 0.045337 0.085304   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.5243034 0.0309564 16.94 6.13e-07 \*\*\*  
## t -0.0128182 0.0007581 -16.91 6.20e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06564 on 7 degrees of freedom  
## Multiple R-squared: 0.9761, Adjusted R-squared: 0.9727   
## F-statistic: 285.9 on 1 and 7 DF, p-value: 6.198e-07

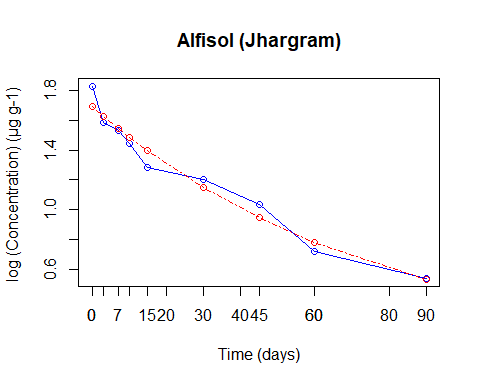
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.01281817

cat("Half-Life:", half\_life, "\n")

## Half-Life: 54.07536

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg g-1) ",   
 col = "blue", main = "Alfisol (Jhargram)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



#####################Dissipation Kinetics (Second Order) T1 ##################

# Dissipation Kinetics(Second Order)MOHANPUR  
Flua <- read\_excel("Flua.xlsx", sheet = 1)  
  
t <- Flua$t  
c <- Flua$mqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.53842 0.04680 0.07045 0.13016 0.26311   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.048090 0.125689 8.339 6.99e-05 \*\*\*  
## t 0.038425 0.003078 12.484 4.87e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2665 on 7 degrees of freedom  
## Multiple R-squared: 0.957, Adjusted R-squared: 0.9509   
## F-statistic: 155.9 on 1 and 7 DF, p-value: 4.874e-06

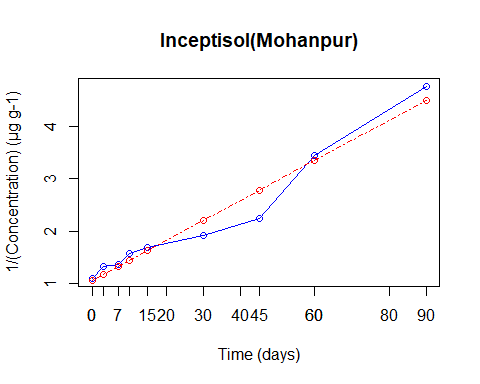
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.03842532

cat("Half-Life:", half\_life, "\n")

## Half-Life: 26.02451

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Inceptisol(Mohanpur)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order)PUNE  
Flua <- read\_excel("Flua.xlsx",sheet=2)  
  
t <- Flua$t  
c <- Flua$pqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## 1 2 3 4 5 6 7   
## 0.06373 0.09812 0.03501 -0.01549 -0.03452 -0.44947 0.30261   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.051511 0.136187 7.721 0.000582 \*\*\*  
## t 0.061233 0.006265 9.774 0.000191 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.249 on 5 degrees of freedom  
## Multiple R-squared: 0.9503, Adjusted R-squared: 0.9403   
## F-statistic: 95.53 on 1 and 5 DF, p-value: 0.0001907

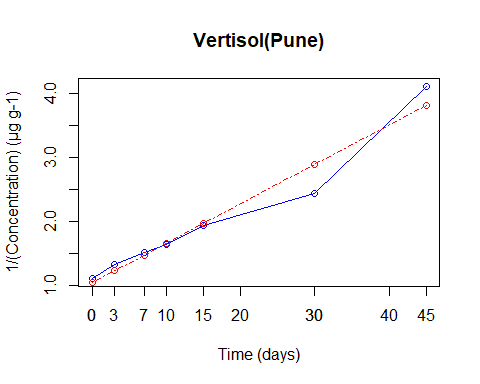
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.06123274

cat("Half-Life:", half\_life, "\n")

## Half-Life: 16.33113

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Vertisol(Pune)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order)CANNING  
Flua <- read\_excel("Flua.xlsx",sheet=3)  
  
t <- Flua$t  
c <- Flua$cqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.52293 -0.05214 0.03497 0.12496 0.45278   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.94162 0.16162 5.826 0.00112 \*\*   
## t 0.06297 0.00550 11.449 2.66e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3157 on 6 degrees of freedom  
## Multiple R-squared: 0.9562, Adjusted R-squared: 0.9489   
## F-statistic: 131.1 on 1 and 6 DF, p-value: 2.665e-05

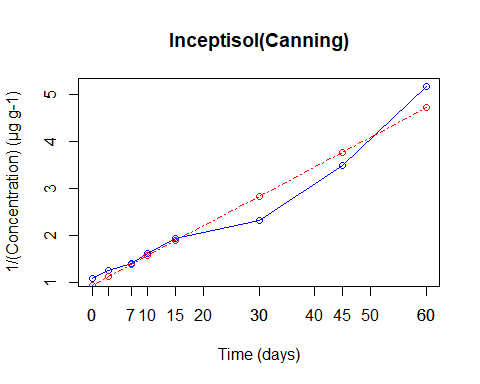
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.06296691

cat("Half-Life:", half\_life, "\n")

## Half-Life: 15.88136

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Inceptisol(Canning)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order)JHARGRAM  
Flua <- read\_excel("Flua.xlsx",sheet=1)  
  
t <- Flua$t  
c <- Flua$jqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.40988 0.03714 0.05410 0.08482 0.19461   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.060672 0.095394 11.12 1.06e-05 \*\*\*  
## t 0.027205 0.002336 11.65 7.77e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2023 on 7 degrees of freedom  
## Multiple R-squared: 0.9509, Adjusted R-squared: 0.9439   
## F-statistic: 135.6 on 1 and 7 DF, p-value: 7.773e-06

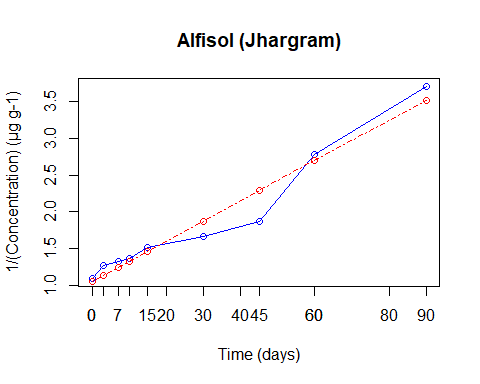
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.02720472

cat("Half-Life:", half\_life, "\n")

## Half-Life: 36.75832

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Alfisol (Jhargram)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



#####################Dissipation Kinetics (Second Order) T2 ##################

# Dissipation Kinetics(Second Order) MOHANPUR  
Flua <- read\_excel("Flua.xlsx", sheet = 4)  
  
t <- Flua$t  
c <- Flua$mqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.24735 0.02087 0.02698 0.06356 0.13988   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.529638 0.059990 8.829 4.83e-05 \*\*\*  
## t 0.018603 0.001469 12.663 4.43e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1272 on 7 degrees of freedom  
## Multiple R-squared: 0.9582, Adjusted R-squared: 0.9522   
## F-statistic: 160.4 on 1 and 7 DF, p-value: 4.429e-06

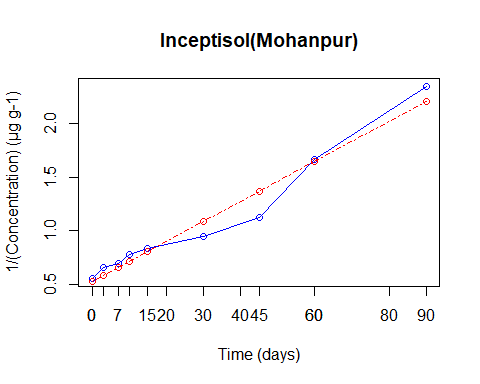
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.01860258

cat("Half-Life:", half\_life, "\n")

## Half-Life: 26.87799

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Inceptisol(Mohanpur)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order)PUNE  
Flua <- read\_excel("Flua.xlsx", sheet = 5)  
  
t <- Flua$t  
c <- Flua$pqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.32485 -0.03649 0.01748 0.08342 0.23277   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.462897 0.090064 5.14 0.00214 \*\*   
## t 0.033860 0.003065 11.05 3.27e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1759 on 6 degrees of freedom  
## Multiple R-squared: 0.9531, Adjusted R-squared: 0.9453   
## F-statistic: 122.1 on 1 and 6 DF, p-value: 3.273e-05

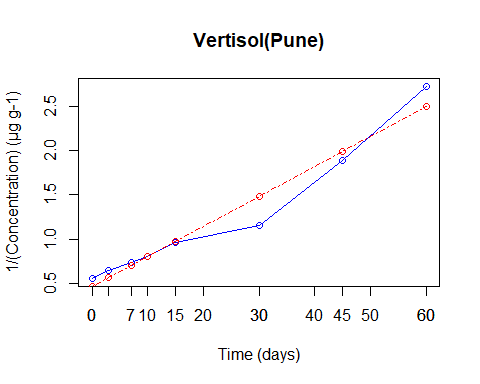
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.03386011

cat("Half-Life:", half\_life, "\n")

## Half-Life: 14.76664

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Vertisol(Pune)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order)CANNING  
Flua <- read\_excel("Flua.xlsx", sheet = 4)  
  
t <- Flua$t  
c <- Flua$cqt  
  
# Calculate 1/C  
c <- 1 / c  
  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.6334 -0.4483 0.1125 0.2525 0.8181   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.247605 0.237569 1.042 0.331951   
## t 0.044656 0.005818 7.676 0.000119 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5037 on 7 degrees of freedom  
## Multiple R-squared: 0.8938, Adjusted R-squared: 0.8786   
## F-statistic: 58.92 on 1 and 7 DF, p-value: 0.0001186

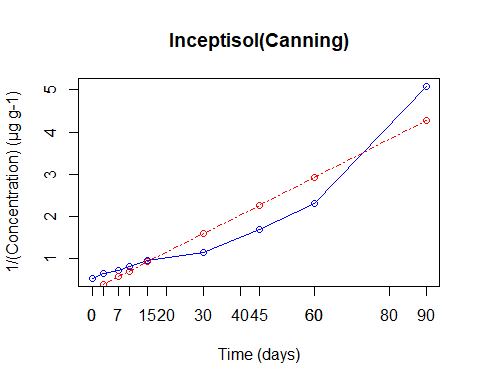
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.04465612

cat("Half-Life:", half\_life, "\n")

## Half-Life: 11.19667

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Inceptisol(Canning)")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order)JHARGRAM  
Flua <- read\_excel("Flua.xlsx", sheet = 4)  
  
t <- Flua$t  
c <- Flua$jqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.17835 0.01585 0.02499 0.04279 0.09126   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.531594 0.043668 12.17 5.77e-06 \*\*\*  
## t 0.013655 0.001069 12.77 4.18e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09259 on 7 degrees of freedom  
## Multiple R-squared: 0.9588, Adjusted R-squared: 0.953   
## F-statistic: 163.1 on 1 and 7 DF, p-value: 4.185e-06

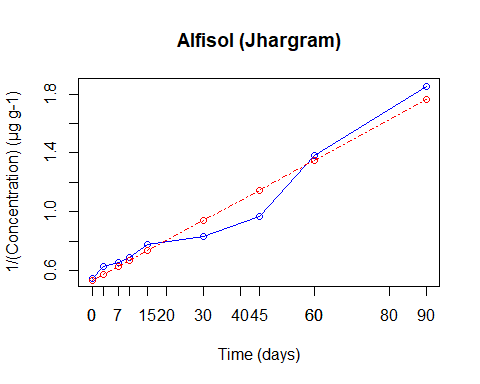
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.01365547

cat("Half-Life:", half\_life, "\n")

## Half-Life: 36.61536

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg g-1)",   
 col = "blue", main = "Alfisol (Jhargram)")  
  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



WATER STUDIES

#####################Dissipation Kinetics (First Order) T1 ##################

# Dissipation Kinetics (First Order) WATER pH 4  
  
Flua <- read\_excel("Flua.xlsx",sheet=6)  
  
t <- Flua$t  
c <- Flua$Fqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.05989 -0.03091 -0.01162 0.01714 0.14171   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.2470727 0.0306129 -8.071 8.62e-05 \*\*\*  
## t -0.0052465 0.0007497 -6.998 0.000212 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06491 on 7 degrees of freedom  
## Multiple R-squared: 0.875, Adjusted R-squared: 0.8571   
## F-statistic: 48.98 on 1 and 7 DF, p-value: 0.0002118

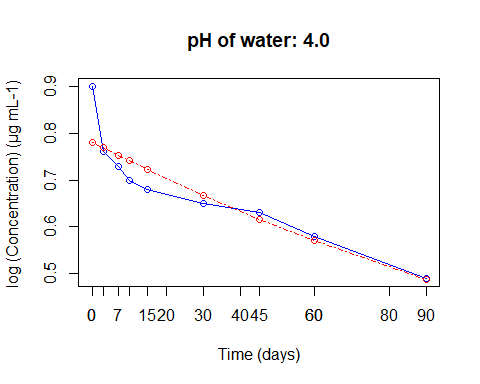
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.005246486

cat("Half-Life:", half\_life, "\n")

## Half-Life: 132.1165

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg mL-1) ",   
 col = "blue", main = "pH of water: 4.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) WATER pH 7  
Flua <- read\_excel("Flua.xlsx",sheet=7)  
  
t <- Flua$t  
c <- Flua$Sqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.07453 -0.06493 -0.04734 0.05427 0.16487   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.327392 0.050131 -6.531 0.000616 \*\*\*  
## t -0.022886 0.001706 -13.415 1.06e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09794 on 6 degrees of freedom  
## Multiple R-squared: 0.9677, Adjusted R-squared: 0.9624   
## F-statistic: 180 on 1 and 6 DF, p-value: 1.063e-05

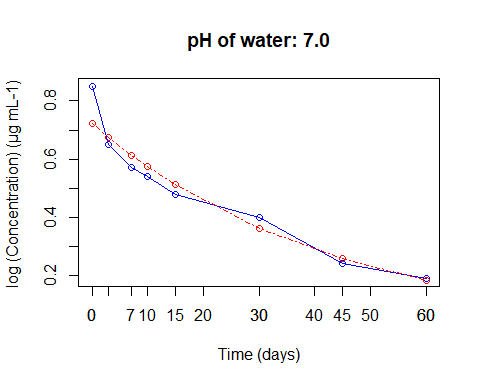
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.02288575

cat("Half-Life:", half\_life, "\n")

## Half-Life: 30.28729

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg mL-1) ",   
 col = "blue", main = "pH of water: 7.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) WATER pH 9.0  
Flua <- read\_excel("Flua.xlsx",sheet=8)  
  
t <- Flua$t  
c <- Flua$Nqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## 1 2 3 4 5   
## 0.025883 -0.063720 0.054522 -0.011958 -0.004727   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.200237 0.038155 -5.248 0.013473 \*   
## t -0.117410 0.004359 -26.932 0.000112 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.05121 on 3 degrees of freedom  
## Multiple R-squared: 0.9959, Adjusted R-squared: 0.9945   
## F-statistic: 725.3 on 1 and 3 DF, p-value: 0.0001123

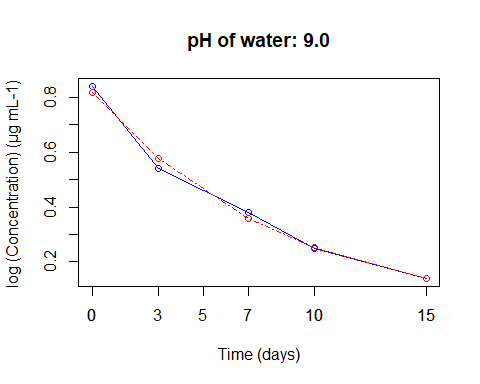
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.1174099

cat("Half-Life:", half\_life, "\n")

## Half-Life: 5.903651

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg mL-1) ",   
 col = "blue", main = "pH of water: 9.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



#####################Dissipation Kinetics (First Order) T2 ##################

# Dissipation Kinetics (First Order) WATER pH 4  
Flua <- read\_excel("Flua.xlsx",sheet=9)  
  
t <- Flua$t  
c <- Flua$Fqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.077364 -0.038098 -0.009556 0.009309 0.137933   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.4498535 0.0326843 13.764 2.52e-06 \*\*\*  
## t -0.0048256 0.0008004 -6.029 0.000527 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0693 on 7 degrees of freedom  
## Multiple R-squared: 0.8385, Adjusted R-squared: 0.8155   
## F-statistic: 36.35 on 1 and 7 DF, p-value: 0.0005267

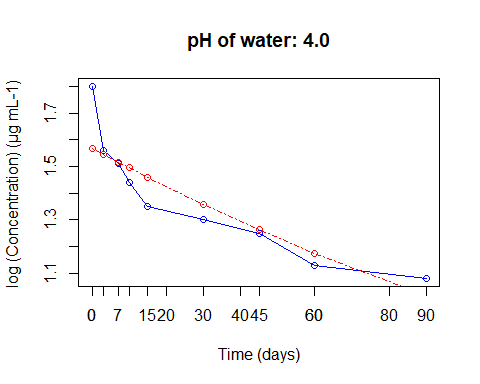
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.004825637

cat("Half-Life:", half\_life, "\n")

## Half-Life: 143.6385

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg mL-1) ",   
 col = "blue", main = "pH of water: 4.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) WATER pH 7  
Flua <- read\_excel("Flua.xlsx",sheet=9)  
  
t <- Flua$t  
c <- Flua$Sqt  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.11562 -0.06771 -0.02378 0.03691 0.18211   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.39451 0.04817 8.189 7.85e-05 \*\*\*  
## t -0.02112 0.00118 -17.906 4.18e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1021 on 7 degrees of freedom  
## Multiple R-squared: 0.9786, Adjusted R-squared: 0.9756   
## F-statistic: 320.6 on 1 and 7 DF, p-value: 4.184e-07

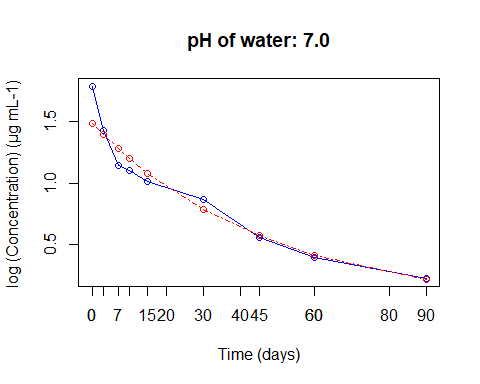
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.0211233

cat("Half-Life:", half\_life, "\n")

## Half-Life: 32.81435

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg mL-1) ",   
 col = "blue", main = "pH of water: 7.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics (First Order) WATER pH 9.0  
Flua <- read\_excel("Flua.xlsx",sheet=10)  
t <- Flua$t  
c <- Flua$Nqt  
  
  
# Fit a first-order kinetic model  
kinetic\_model <- lm(log(c) ~ t, data = Flua)  
  
# Print the summary of the model  
summary(kinetic\_model)

##   
## Call:  
## lm(formula = log(c) ~ t, data = Flua)  
##   
## Residuals:  
## 1 2 3 4 5 6   
## 0.26074 -0.01414 -0.02711 -0.13441 -0.27526 0.19017   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.298872 0.135543 2.205 0.092138 .   
## t -0.081839 0.009269 -8.829 0.000908 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.223 on 4 degrees of freedom  
## Multiple R-squared: 0.9512, Adjusted R-squared: 0.939   
## F-statistic: 77.95 on 1 and 4 DF, p-value: 0.0009083

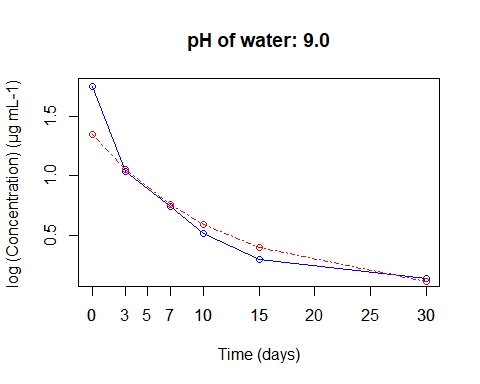
# Extract the slope (negative of the rate constant) and intercept from the model  
rate\_constant <- -coef(kinetic\_model)[2]  
  
# Calculate half-life  
half\_life <- log(2) / rate\_constant  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.08183866

cat("Half-Life:", half\_life, "\n")

## Half-Life: 8.46968

# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "log (Concentration) (µg mL-1) ",   
 col = "blue", main = "pH of water: 9.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, exp(predict(kinetic\_model)), col = "red",type="o", lty = 4, lwd=1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



#####################Dissipation Kinetics (Second Order) T1 ##################

Flua <- read\_excel("Flua.xlsx",sheet=6)  
  
t <- Flua$t  
c <- Flua$Fqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.15929 -0.04312 0.02055 0.04150 0.07598   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.2703962 0.0377941 33.614 5.34e-09 \*\*\*  
## t 0.0082811 0.0009255 8.947 4.43e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.08014 on 7 degrees of freedom  
## Multiple R-squared: 0.9196, Adjusted R-squared: 0.9081   
## F-statistic: 80.06 on 1 and 7 DF, p-value: 4.431e-05

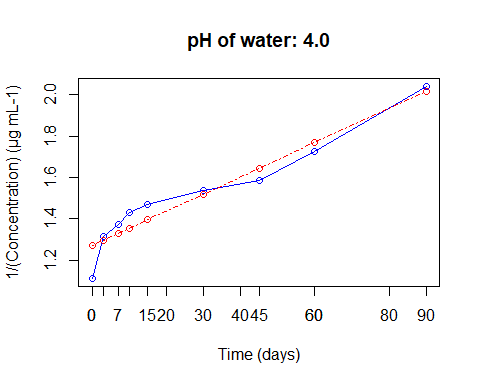
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.008281056

cat("Half-Life:", half\_life, "\n")

## Half-Life: 120.7575

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg mL-1)",   
 col = "blue", main = "pH of water: 4.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order) pH 7.0  
Flua <- read\_excel("Flua.xlsx",sheet=7)  
  
t <- Flua$t  
c <- Flua$Sqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.60694 -0.00832 0.06379 0.14358 0.21858   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.169297 0.144368 8.099 0.00019 \*\*\*  
## t 0.064588 0.004913 13.147 1.2e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.282 on 6 degrees of freedom  
## Multiple R-squared: 0.9664, Adjusted R-squared: 0.9609   
## F-statistic: 172.8 on 1 and 6 DF, p-value: 1.195e-05

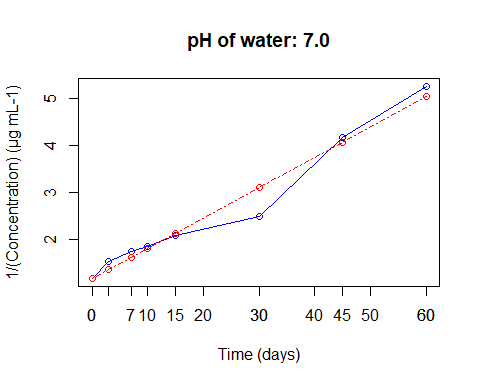
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.06458796

cat("Half-Life:", half\_life, "\n")

## Half-Life: 15.48276

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg mL-1)",   
 col = "blue", main = "pH of water: 7.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order) pH 9.0  
Flua <- read\_excel("Flua.xlsx",sheet=8)  
  
t <- Flua$t  
c <- Flua$Nqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## 1 2 3 4 5   
## 0.53593 0.03639 -0.73177 -0.52427 0.68373   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.65455 0.53835 1.216 0.31099   
## t 0.38697 0.06151 6.291 0.00811 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7226 on 3 degrees of freedom  
## Multiple R-squared: 0.9295, Adjusted R-squared: 0.9061   
## F-statistic: 39.58 on 1 and 3 DF, p-value: 0.008112

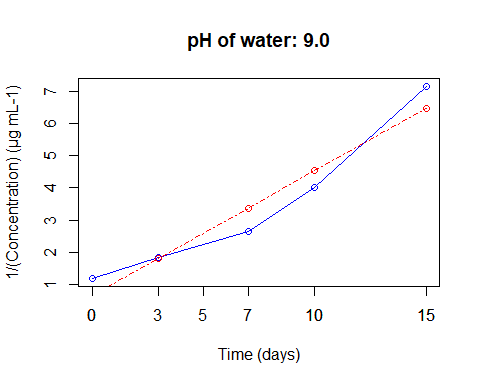
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 1)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.3869719

cat("Half-Life:", half\_life, "\n")

## Half-Life: 2.584167

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg mL-1)",   
 col = "blue", main = "pH of water: 9.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



#####################Dissipation Kinetics (Second Order) T2 ##################

# Dissipation Kinetics(Second Order) pH 4.0  
  
Flua <- read\_excel("Flua.xlsx",sheet=9)  
  
t <- Flua$t  
c <- Flua$Fqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.080762 -0.006222 0.000430 0.023613 0.049773   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.6363178 0.0198785 32.010 7.51e-09 \*\*\*  
## t 0.0036433 0.0004868 7.484 0.000139 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.04215 on 7 degrees of freedom  
## Multiple R-squared: 0.8889, Adjusted R-squared: 0.873   
## F-statistic: 56.02 on 1 and 7 DF, p-value: 0.0001391

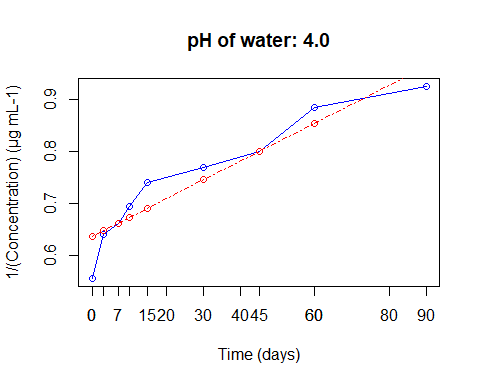
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.003643349

cat("Half-Life:", half\_life, "\n")

## Half-Life: 137.2364

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg mL-1)",   
 col = "blue", main = "pH of water: 4.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order) pH 7.0  
Flua <- read\_excel("Flua.xlsx",sheet=9)  
t <- Flua$t  
c <- Flua$Sqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.4295 -0.2352 0.1009 0.1659 0.4564   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.422736 0.146703 2.882 0.0236 \*   
## t 0.038541 0.003593 10.728 1.34e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3111 on 7 degrees of freedom  
## Multiple R-squared: 0.9427, Adjusted R-squared: 0.9345   
## F-statistic: 115.1 on 1 and 7 DF, p-value: 1.344e-05

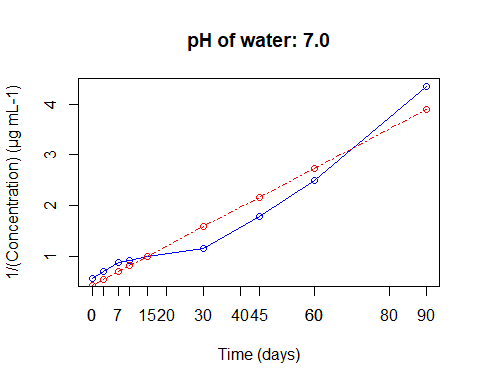
# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.03854133

cat("Half-Life:", half\_life, "\n")

## Half-Life: 12.97309

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg mL-1)",   
 col = "blue", main = "pH of water: 7.0")  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)



# Dissipation Kinetics(Second Order) pH 9.0  
Flua <- read\_excel("Flua.xlsx",sheet=10)  
  
t <- Flua$t  
c <- Flua$Nqt  
  
# Calculate 1/C  
c <- 1 / c  
  
# Fit a linear model to the transformed data  
linear\_model <- lm(c ~ t, data = Flua)  
  
# Print the summary of the model  
summary(linear\_model)

##   
## Call:  
## lm(formula = c ~ t, data = Flua)  
##   
## Residuals:  
## 1 2 3 4 5 6   
## 0.4626 0.1775 -0.3331 -0.4366 -0.1518 0.2814   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.10881 0.24488 0.444 0.679775   
## t 0.22509 0.01675 13.441 0.000177 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4029 on 4 degrees of freedom  
## Multiple R-squared: 0.9783, Adjusted R-squared: 0.9729   
## F-statistic: 180.7 on 1 and 4 DF, p-value: 0.0001772

# Extract the slope and intercept from the linear model  
slope <- coef(linear\_model)[2]  
intercept <- coef(linear\_model)[1]  
  
# Calculate the rate constant for the second-order reaction  
rate\_constant <- slope  
  
# Calculate the half-life for the second-order reaction  
half\_life <- 1 / (rate\_constant \* 2)  
  
# Print the rate constant and half-life  
cat("Rate Constant:", rate\_constant, "\n")

## Rate Constant: 0.2250879

cat("Half-Life:", half\_life, "\n")

## Half-Life: 2.221354

# Plot 1/C against time  
# Plot the actual data  
plot(t, c, type = "o", xlab = "Time (days)", ylab = "1/(Concentration) (µg mL-1)",   
 col = "blue", main = "pH of water: 9.0")  
  
# Add the fitted line (predicted values) to the plot  
lines(t, predict(linear\_model), col = "red", type = "o", lty = 4, lwd = 1)  
  
# Customize time axis breaks and labels  
axis.breaks <- c(0, 3, 7, 10, 15, 30, 45, 60, 90)  
axis.labels <- axis.breaks  
  
# Set breaks and labels on the time axis  
axis(side = 1, at = axis.breaks, labels = axis.labels)

