Biehler et al

Tricia

2024-11-19

load libraries

```
library(ggplot2)
library(dplyr)
library(betareg)
library(emmeans)
library(MuMIn)
```

Import CSV

```
data <- read.csv("../datafiles/BiehlerDataBKA.csv", header = TRUE, sep = ",")

# Convert 'Sex', 'Month', 'Year', and 'Species' to factors
data$Sex <- factor(data$Sex)
data$Species <- factor(data$Species)
```

Mean and Standard Devation for BKA

```
# Calculate mean and SD for each species
summary_stats <- data %>%
  group_by(Species) %>%
  summarise(
    Mean = mean(Bacterial_Killing_Percentage, na.rm = TRUE),
    SD = sd(Bacterial_Killing_Percentage, na.rm = TRUE)
  )

# Print the summary statistics
print(summary_stats)
```

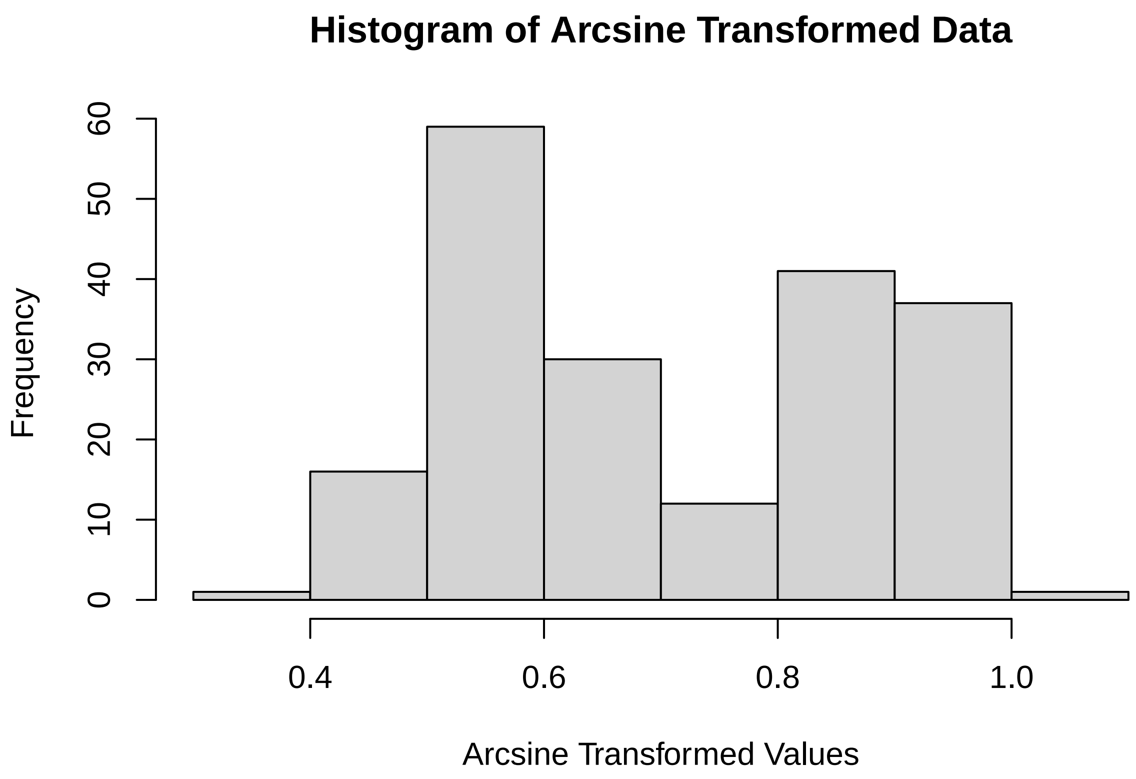
```
## # A tibble: 4 x 3
##   Species  Mean    SD
##   <fct>   <dbl> <dbl>
## 1 AMGO     26.3  4.76
## 2 HOFI     60.1  5.56
## 3 LEGO     27.6  6.12
## 4 PISI     33.4  7.38
```

Transformation of data

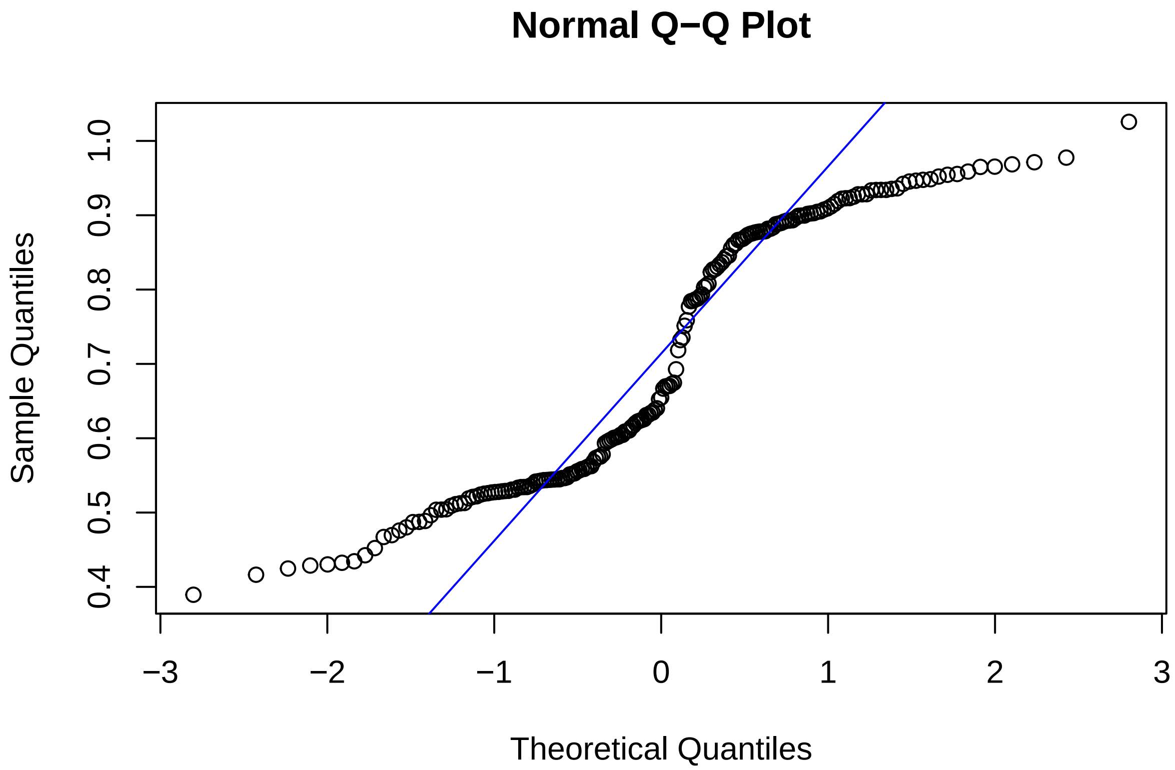
```r
# Convert percentage to proportion
data$Bacterial_Killing_Proportion <- data$Bacterial_Killing_Percentage / 100

# Apply arcsine square root transformation
data$Bacterial_Killing_Arcsine <- asin(sqrt(data$Bacterial_Killing_Proportion))

# Shapiro-Wilk test for normality
shapiro.test(data$Bacterial_Killing_Arcsine)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Bacterial_Killing_Arcsine
## W = 0.89719, p-value = 2.093e-10
```

```r
# Visual inspection with histogram and Q-Q plot
hist(data$Bacterial_Killing_Arcsine, main="Histogram of Arcsine Transformed Data", xlab="Arcsine Transf
```



```r
qqnorm(data$Bacterial_Killing_Arcsine)
qqline(data$Bacterial_Killing_Arcsine, col="blue")
```



Beta regression as data are not normally distributed after arcsin transformation

```
# Fit a beta regression model for the proportion data
beta_bka_species <- betareg(Bacterial_Killing_Proportion ~ Species, data = data)

# View the summary of the model
summary(beta_bka_species)
```

```
##
## Call:
## betareg(formula = Bacterial_Killing_Proportion ~ Species, data = data)
##
## Quantile residuals:
##     Min      1Q  Median      3Q     Max
## -3.1051 -0.4407  0.0222  0.6436  2.8471
##
## Coefficients (mean model with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.02429    0.05124 -19.989  < 2e-16 ***
## SpeciesHOFI  1.43335    0.05803  24.701  < 2e-16 ***
## SpeciesLEGO  0.05717    0.06338   0.902    0.367
## SpeciesPISI  0.32771    0.07343   4.463 8.09e-06 ***
##
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)
## (phi)   63.794      6.383   9.994   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 281.8 on 5 Df
## Pseudo R-squared: 0.8618
## Number of iterations: 12 (BFGS) + 2 (Fisher scoring)
```

Post Hoc Comparison

```
# Obtain estimated marginal means for Species from the beta regression model
emmeans_species <- emmeans(beta_bka_species, ~ Species)

# Perform pairwise comparisons between all species
pairwise_comparisons <- pairs(emmeans_species, adjust = "fdr")

# Print pairwise comparisons with more significant digits
print(pairwise_comparisons)
```

```
##  contrast    estimate      SE  df z.ratio p.value
##  AMGO - HOFI  -0.3367 0.01190 Inf -28.268  <.0001
##  AMGO - LEGO  -0.0113 0.01240 Inf  -0.906  0.3651
##  AMGO - PISI  -0.0684 0.01530 Inf  -4.455  <.0001
##  HOFI - LEGO   0.3254 0.00991 Inf  32.823  <.0001
##  HOFI - PISI   0.2683 0.01340 Inf  20.046  <.0001
##  LEGO - PISI  -0.0571 0.01390 Inf  -4.122  <.0001
##
## P value adjustment: fdr method for 6 tests
```

```
# Convert pairwise comparisons to a data frame
pairwise_results_df <- as.data.frame(pairwise_comparisons)

# View the results
pairwise_results_df
```

```
##  contrast      estimate          SE  df z.ratio p.value
##  AMGO - HOFI -0.3366701 0.011910018 Inf -28.268  <.0001
##  AMGO - LEGO -0.0112617 0.012435274 Inf  -0.906  0.3651
##  AMGO - PISI -0.0683771 0.015347939 Inf  -4.455  <.0001
##  HOFI - LEGO  0.3254084 0.009914167 Inf  32.823  <.0001
##  HOFI - PISI  0.2682929 0.013383544 Inf  20.046  <.0001
##  LEGO - PISI -0.0571154 0.013856993 Inf  -4.122  <.0001
##
## P value adjustment: fdr method for 6 tests
```

Model Testing

```
# Fit a model with main effects only
main_effects_model <- betareg(Bacterial_Killing_Proportion ~ Species + Sex, data = data)
model_A <- betareg(Bacterial_Killing_Proportion ~ Species, data = data)
model_B <- betareg(Bacterial_Killing_Proportion ~ Sex, data = data)
model_C <- betareg(Bacterial_Killing_Proportion ~ Species * Sex, data = data)

# Example for comparing models using AIC
```

```r
AIC(main_effects_model, model_A, model_B, model_C)
```

```
##                    df       AIC
## main_effects_model  6 -549.6710
## model_A             5 -553.6223
## model_B             3 -160.8675
## model_C             9 -545.0486
```

Summarize AIC models

```r
summary(main_effects_model)
```

```
## 
## Call:
## betareg(formula = Bacterial_Killing_Proportion ~ Species + Sex, data = data)
## 
## Quantile residuals:
##     Min      1Q  Median      3Q     Max 
## -3.0805 -0.4503  0.0240  0.6821  2.8022 
## 
## Coefficients (mean model with logit link):
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.01196    0.05688 -17.791  < 2e-16 ***
## SpeciesHOFI  1.43249    0.05792  24.730  < 2e-16 ***
## SpeciesLEGO  0.05995    0.06343   0.945    0.345    
## SpeciesPISI  0.33998    0.07412   4.587  4.5e-06 ***
## SexMale     -0.01987    0.03953  -0.503    0.615    
## 
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)    
## (phi)   64.130      6.433   9.968   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1 
## 
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 280.8 on 6 Df
## Pseudo R-squared: 0.8625
## Number of iterations: 14 (BFGS) + 2 (Fisher scoring)
```

```r
summary(model_A)
```

```
## 
## Call:
## betareg(formula = Bacterial_Killing_Proportion ~ Species, data = data)
## 
## Quantile residuals:
##     Min      1Q  Median      3Q     Max 
## -3.1051 -0.4407  0.0222  0.6436  2.8471 
## 
## Coefficients (mean model with logit link):
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.02429    0.05124 -19.989  < 2e-16 ***
## SpeciesHOFI  1.43335    0.05803  24.701  < 2e-16 ***
## SpeciesLEGO  0.05717    0.06338   0.902    0.367    
## SpeciesPISI  0.32771    0.07343   4.463 8.09e-06 ***
## 
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)    
## (phi)   63.794      6.383   9.994   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1 
## 
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 281.8 on 5 Df
## Pseudo R-squared: 0.8618
## Number of iterations: 12 (BFGS) + 2 (Fisher scoring)
```

```
summary(model_B)
```

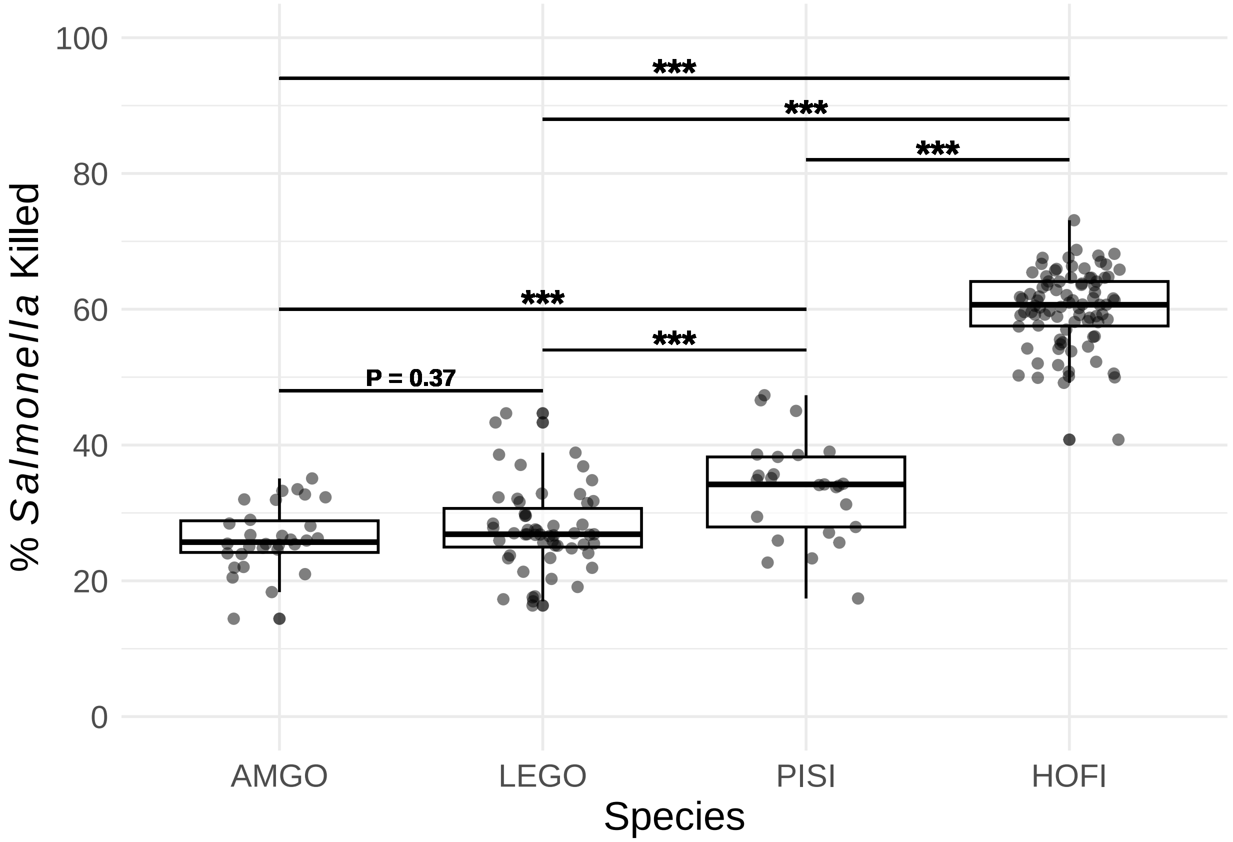
```
## 
## Call:
## betareg(formula = Bacterial_Killing_Proportion ~ Sex, data = data)
## 
## Quantile residuals:
##     Min      1Q  Median      3Q     Max 
## -2.0585 -0.8762 -0.2497  0.9821  1.9359 
## 
## Coefficients (mean model with logit link):
##             Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.19860    0.07601  -2.613  0.00898 **
## SexMale     -0.15832    0.09622  -1.645  0.09988 . 
## 
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)    
## (phi)   8.4850     0.8124   10.45   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1 
## 
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 83.43 on 3 Df
## Pseudo R-squared: 0.01405
## Number of iterations: 12 (BFGS) + 1 (Fisher scoring)
```

```
summary(model_C)
```

```
## 
## Call:
## betareg(formula = Bacterial_Killing_Proportion ~ Species * Sex, data = data)
## 
## Quantile residuals:
##     Min      1Q  Median      3Q     Max 
## -3.1145 -0.4662 -0.0136  0.6470  2.7569 
## 
## Coefficients (mean model with logit link):
##                     Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -1.05758    0.08473 -12.482  < 2e-16 ***
## SpeciesHOFI          1.47016    0.09433  15.585  < 2e-16 ***
## SpeciesLEGO          0.12139    0.11370   1.068 0.285680    
## SpeciesPISI          0.43288    0.11292   3.833 0.000126 ***
## SexMale               0.05220    0.10598   0.493 0.622346
## SpeciesHOFI:SexMale -0.05819    0.11924  -0.488 0.625551
## SpeciesLEGO:SexMale -0.09300    0.13710  -0.678 0.497580
## SpeciesPISI:SexMale -0.16706    0.15030  -1.112 0.266350
##
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)
## (phi)   64.583      6.479   9.968   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 281.5 on 9 Df
## Pseudo R-squared: 0.8635
## Number of iterations: 18 (BFGS) + 2 (Fisher scoring)
```

Boxplot of BKA

```
# Reorder the species factor levels
data$Species <- factor(data$Species, levels = c("AMGO", "LEGO", "PISI", "HOFI"))

# Create box plot with individual points
BKAAssay <- ggplot(data, aes(x=Species, y=Bacterial_Killing_Percentage)) +
  geom_boxplot(color="black", alpha=0.7) +  # Box plot
  geom_jitter(width=0.2, size=1.5, alpha=0.5) +  # Jittered points
  labs(x="Species",
       y=expression("% " * italic(Salmonella) * " Killed")) +
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))  +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
    axis.title.x = element_text(size = 15),
    axis.title.y = element_text(size = 15),
    axis.text.x = element_text(size = 12),  # Bold x-axis labels
    axis.text.y = element_text(size = 12)  # Bold y-axis numbers
  )  +
  # Add line for significance between AMGO and HOFI
  geom_segment(aes(x=1, xend=4, y=94, yend=94), color="black", linewidth=.5) +
  # Add text for significance label "***"
  geom_text(aes(x=2.5, y=95, label="***"), color="black", size=5, fontface="bold") +
  # Add line for significance between LEGO and HOFI
  geom_segment(aes(x=2, xend=4, y=88, yend=88), color="black", linewidth=.5) +
  # Add text for significance label "***"
  geom_text(aes(x=3, y=89, label="***"), color="black", size=5, fontface="bold") +
  # Add line for significance between PISI and HOFI
  geom_segment(aes(x=3, xend=4, y=82, yend=82), color="black", linewidth=.5) +
  # Add text for significance label "***"
  geom_text(aes(x=3.5, y=83, label="***"), color="black", size=5, fontface="bold") +
  # Add line for significance between PISI and AMGO
  geom_segment(aes(x=1, xend=3, y=60, yend=60), color="black", linewidth=.5) +
  # Add text for significance label "***"
  geom_text(aes(x=2, y=61, label="***"), color="black", size=5, fontface="bold") +
  # Add line for significance between PISI and LEGO
  geom_segment(aes(x=2, xend=3, y=54, yend=54), color="black", linewidth=.5) +
  # Add text for significance label "***"
  geom_text(aes(x=2.5, y=55, label="***"), color="black", size=5, fontface="bold") +
  # Add line for significance between AMGO and LEGO
  geom_segment(aes(x=1, xend=2, y=48, yend=48), color="black", linewidth=.5) +
  # Add text for significance label "***"
  geom_text(aes(x=1.5, y=50, label="P = 0.37"), color="black", size=3, fontface="bold")

print(BKAAssay)
```



# export tiff with 300dpi

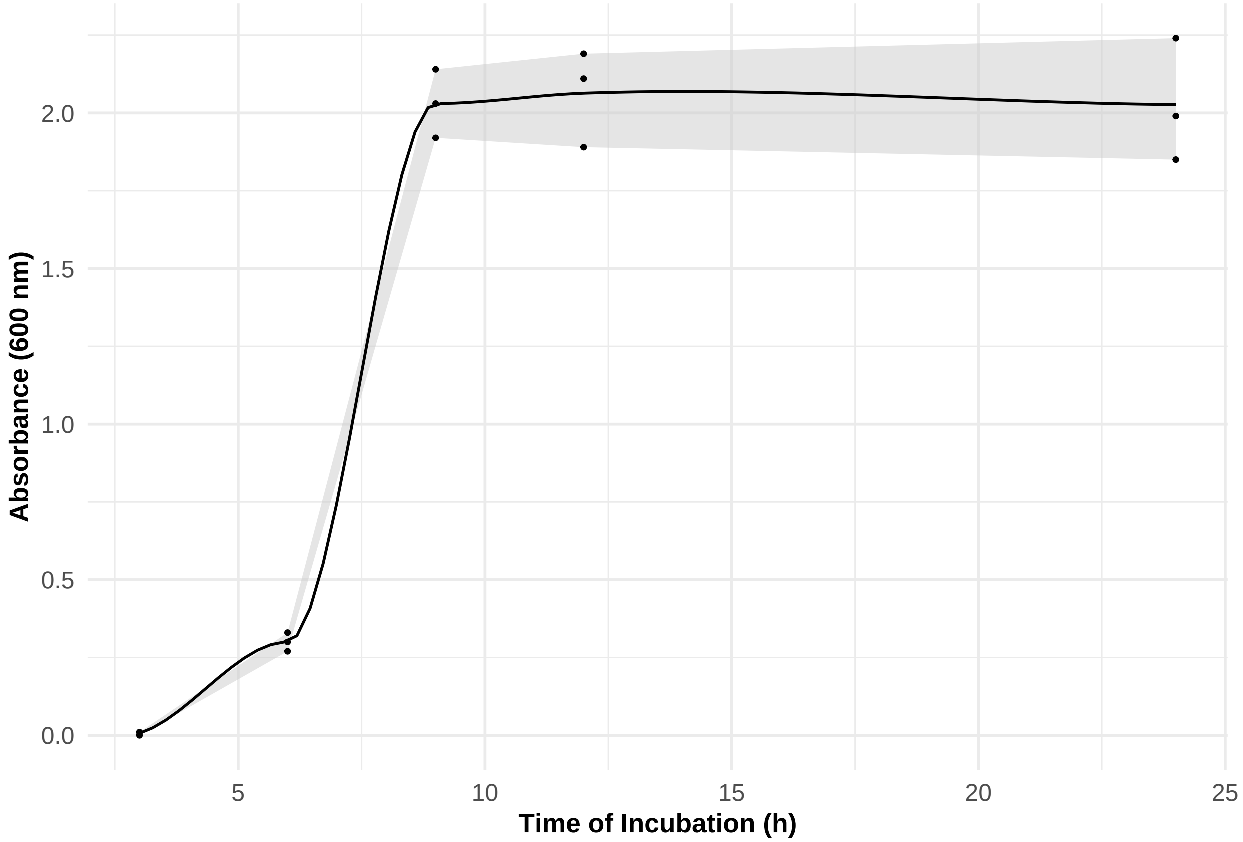
```
ggsave(
filename="../figures/BKAAssay.tiff",
plot = BKAAssay,
width = 200,
height = 200,
units = c("mm"),
dpi = 300,
bg = "white"
)
```

Import Growth Curve Data

```
growthcurvedata <- read.csv("../datafiles/GrowthCurve.csv", header = TRUE, sep = ",")
```

Growth Curve

```
# Calculate mean absorbance for each time point
average_data <- growthcurvedata %>%
  group_by(Time.h.) %>%
  summarise(
    Mean = mean(A600),
    SD = sd(A600),
    Min = min(A600),
    Max = max(A600)
  )

# Plot with smoothed line and shaded area representing the standard deviation
growthcurve <- ggplot() +
  geom_ribbon(data = average_data, aes(x = Time.h., y = Mean, ymin = Min, ymax = Max),
              fill = "grey80", alpha = 0.5) +
  geom_smooth(data = average_data, aes(x = Time.h., y = Mean), method = "loess",
              formula = y ~ x, color = "black", se = FALSE,
              size = 0.5) +
  geom_point(data = growthcurvedata, aes(x = Time.h., y = A600), color = "black", size = 0.5) +
  labs(x = "Time of Incubation (h)",
       y = "Absorbance (600 nm)") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 10),
    axis.title.x = element_text(face = "bold", size = 10),
    axis.title.y = element_text(face = "bold", size = 10)
  )

print(growthcurve)
```



# print growth curve info

```
print(average_data)
```

```
## # A tibble: 5 x 5
##   Time.h.    Mean      SD   Min   Max
##     <int>   <dbl>   <dbl> <dbl> <dbl>
## 1       3 0.00667 0.00577  0     0.01
## 2       6 0.3     0.03     0.27  0.33
## 3       9 2.03    0.110    1.92  2.14
## 4      12 2.06    0.155    1.89  2.19
## 5      24 2.03    0.198    1.85  2.24
```

export tiff with 300dpi

```
ggsave(
filename="../figures/growthcurve.tiff",
plot = growthcurve,
width = 200,
height = 100,
units = c("mm"),
dpi = 300,
bg = "white"
)
```