

# Visualizing TPC Data (1999 vs. 2024)

Taylor Hatcher

2025-02-10

```
# Load Data
```

```
getwd()
```

```
## [1] "/Users/taylorhatcher/Desktop/Repos/WARP2024/Data"
```

```
tpcvis <- read.csv("PastPresentFilteredConstantTpc2024.csv")
```

## Historic Data Analysis

### Filter Historic Data

```
tpcvis_past <- tpcvis %>%  
  filter(time.per == "past")
```

### Log Scale Growth Rate

### Arithmetic Scale Growth Rate

### Combined Scale Comparison (Historic Data)

## Comparison of 1999 vs. 2024 Data

### Logarithmic Scale Comparison

### Arithmetic Scale Comparison

## Aggregated Data Analysis

### Logarithmic Scale Aggregation

### Arithmetic Scale Aggregation

```
# Filter na's from rgrlog out
```

```
tpcvis_cleancurrent <- tpcvis_current %>% filter(is.finite(rgrlog))
```

```
# Temp is a factor for analyzing
```

```
tpcvis_cleancurrent$temp <- as.factor(tpcvis_cleancurrent$temp)
```

## Analysis approach

1. Fit the mixed model with a quadratic time effect
2. Check model fit using summary & ANOVA
3. Simulate residuals with DHARMa
4. Run residual diagnostics
5. Compare models (linear vs. quadratic)
6. Refit the final model using REML
7. Visualize residuals and predictions using DHARMa

```
library(lme4)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

## Fit mixed model with a quadratic time effect—

```
mod.lmer <- lmer(rgrlog ~ poly(duration, 2) + temp + (1 | ID),
                 data = tpcvis_cleancurrent,
                 REML = FALSE)

summary(mod.lmer)
```

rgrlog= response variable, poly(duration, 2) - models duration as a quadratic polynomial, temp is a categorical fixed effect, (1 | ID) random intercept for individual

```
## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: rgrlog ~ poly(duration, 2) + temp + (1 | ID)
## Data: tpcvis_cleancurrent
##
##      AIC      BIC   logLik deviance df.resid
## -17939.2 -17875.1   8980.6 -17961.2     2502
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -5.2955 -0.5331  0.0256  0.4536  5.9891
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## ID      (Intercept)  4.339e-06  0.002083
```

```
## Residual          4.324e-05 0.006576
## Number of obs: 2513, groups: ID, 438
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    0.0059939  0.0003614  16.586
## poly(duration, 2)1 0.0001093  0.0066297   0.016
## poly(duration, 2)2 -0.1604871  0.0066818 -24.019
## temp17          0.0069691  0.0004602  15.143
## temp23          0.0150547  0.0004537  33.180
## temp29          0.0210419  0.0004712  44.653
## temp35          0.0250289  0.0005085  49.217
## temp40          0.0038415  0.0005672   6.772
## temp41         -0.0056523  0.0006952  -8.130
##
## Correlation of Fixed Effects:
##               (Intr) p(,2)1 p(,2)2 temp17 temp23 temp29 temp35 temp40
## ply(drt,2)1  0.016
## ply(drt,2)2 -0.030  0.000
## temp17       -0.612 -0.020  0.010
## temp23       -0.635 -0.021  0.009  0.499
## temp29       -0.593 -0.023  0.011  0.464  0.477
## temp35       -0.645 -0.018  0.007  0.437  0.454  0.420
## temp40       -0.596 -0.012  0.062  0.391  0.399  0.382  0.395
## temp41       -0.477 -0.005  0.142  0.316  0.324  0.307  0.311  0.296
```

```
anova(mod.lmer)
```

```
## Analysis of Variance Table
##               npar    Sum Sq Mean Sq F value
## poly(duration, 2)    2 0.016658  0.008329  192.63
## temp                6 0.200717  0.033453  773.67
```

## Simulate residuals

```
library(DHARMA)
```

```
## This is DHARMA 0.4.7. For overview type '?DHARMA'. For recent changes, type news(package = 'DHARMA')
```

```
sim_res <- simulateResiduals(fittedModel = mod.lmer)
```

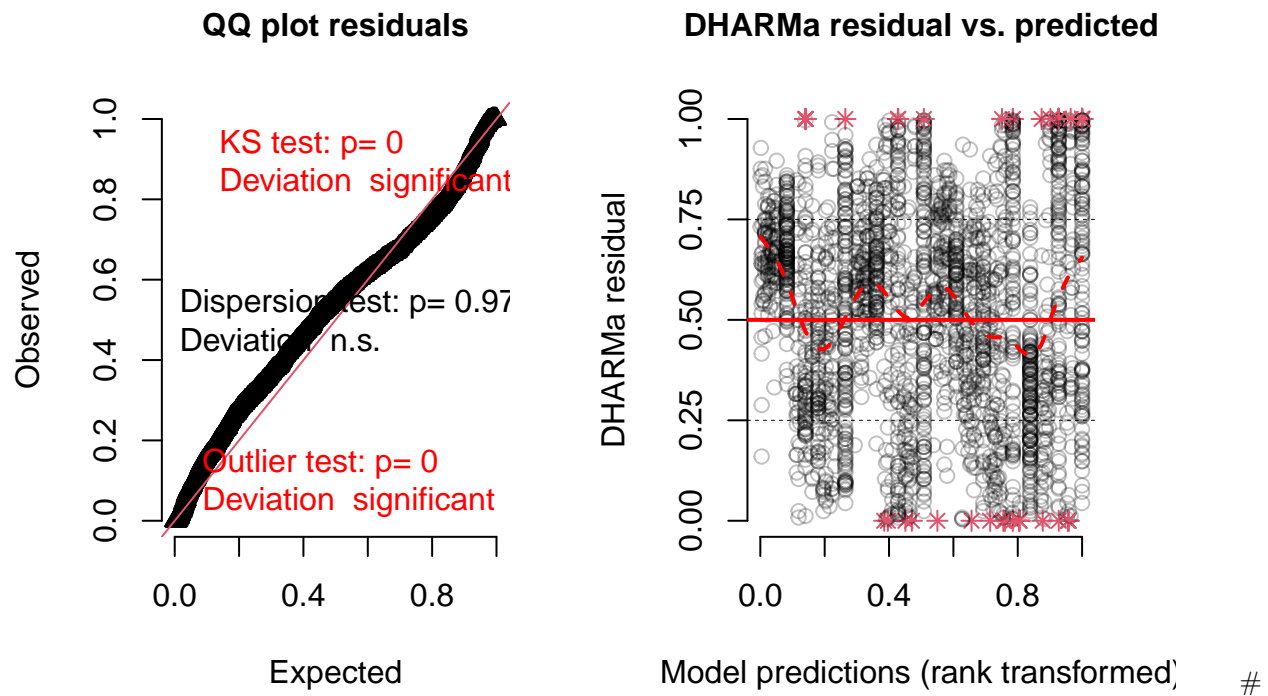
```
print(sim_res)
```

```
## Object of Class DHARMA with simulated residuals based on 250 simulations with refit = FALSE . See ?DHARMA
##
```

```
## Scaled residual values: 0.188 0.628 0.664 0.64 0.464 0.572 0.1 0.512 0.192 0.604 0.696 0.828 0.324 0
```

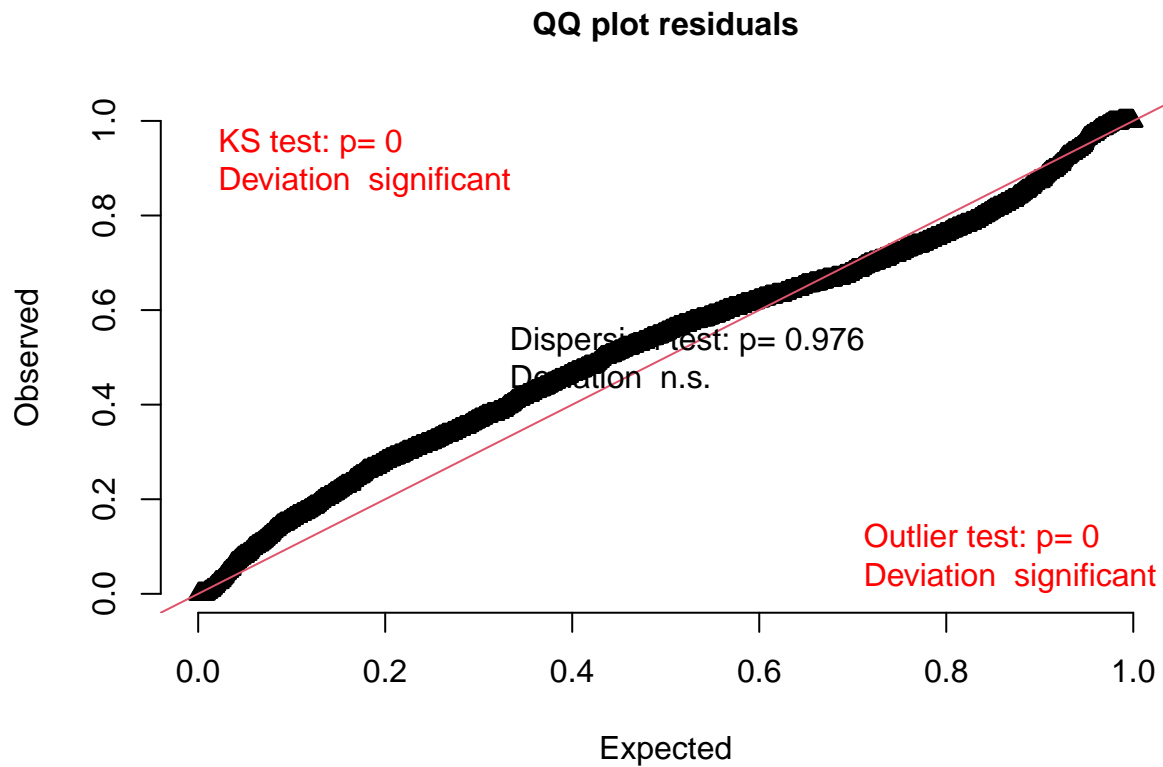
```
plot(sim_res)
```

## DHARMa residual



Check for uniformity

```
testUniformity(sim_res)
```



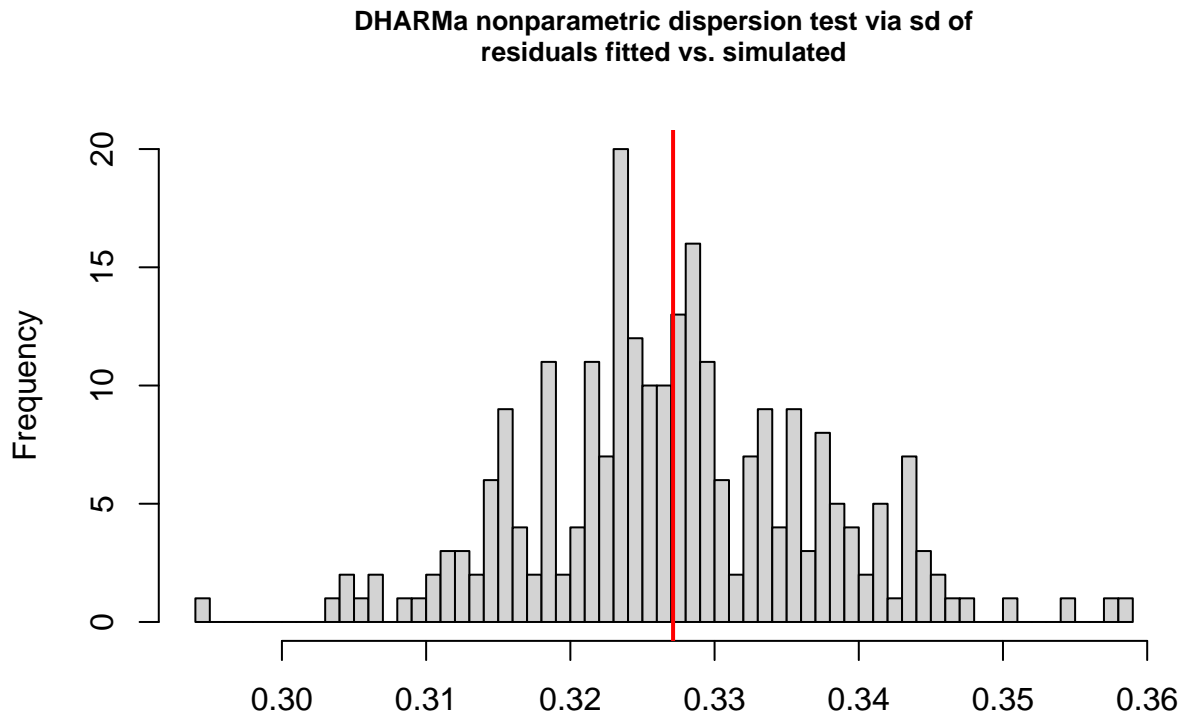
##

## One-sample Kolmogorov-Smirnov test

```
##
## data: simulationOutput$scaledResiduals
## D = 0.082921, p-value = 1.998e-15
## alternative hypothesis: two-sided
```

## Check for overdispersion

```
testDispersion(sim_res)
```



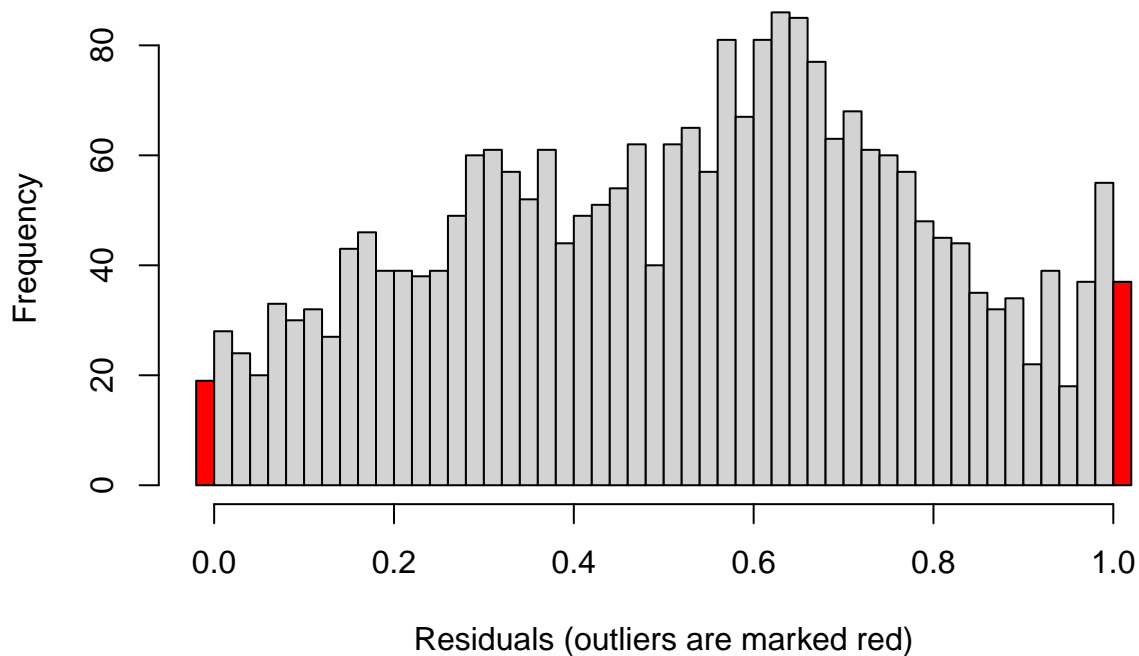
Simulated values, red line = fitted model. p-value (two.sided) = 0.976

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.99975, p-value = 0.976
## alternative hypothesis: two.sided
```

## Check for outliers

```
testOutliers(sim_res)
```

## Outlier test significant



```
##
## DHARMA outlier test based on exact binomial test with approximate
## expectations
##
## data:  sim_res
## outliers at both margin(s) = 56, observations = 2513, p-value =
## 2.587e-11
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
## 0.01687604 0.02884121
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                0.02228412
```

```
mod.lmer_linear <- lmer(rgrlog ~ duration + temp + (1 | ID),
                        data = tpcvis_cleancurrent, REML = FALSE)
```

```
anova(mod.lmer_linear, mod.lmer) # Compare linear vs. quadratic
```

```
## Data: tpcvis_cleancurrent
## Models:
## mod.lmer_linear: rgrlog ~ duration + temp + (1 | ID)
## mod.lmer: rgrlog ~ poly(duration, 2) + temp + (1 | ID)
##               npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## mod.lmer_linear   10 -17428 -17370 8724.2   -17448
## mod.lmer           11 -17939 -17875 8980.6   -17961 512.73  1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod.lmer_final <- lmer(rgrlog ~ poly(duration, 2) + temp + (1 | ID),
  data = tpcvis_cleancurrent,
  REML = TRUE) # Now using REML
```

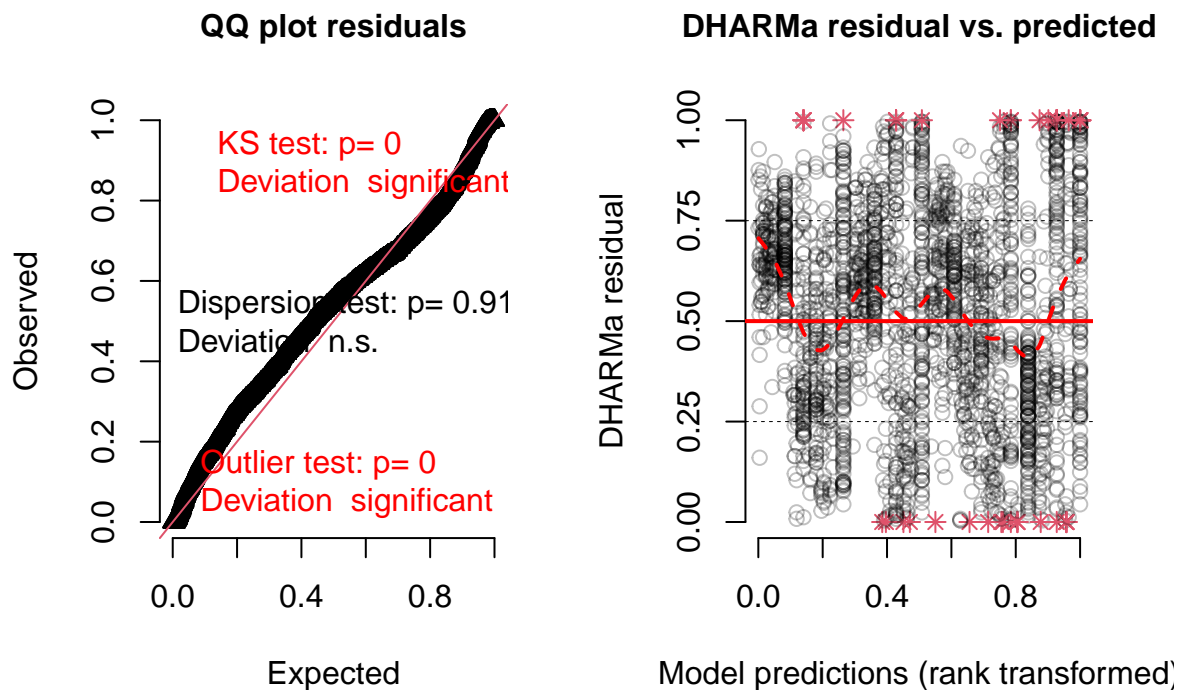
```
library(DHARMA)
```

## Simulate residuals

```
sim_res <- simulateResiduals(fittedModel = mod.lmer_final)

# Plot residual diagnostics
plot(sim_res)
```

DHARMA residual



```
library(ggplot2)
```

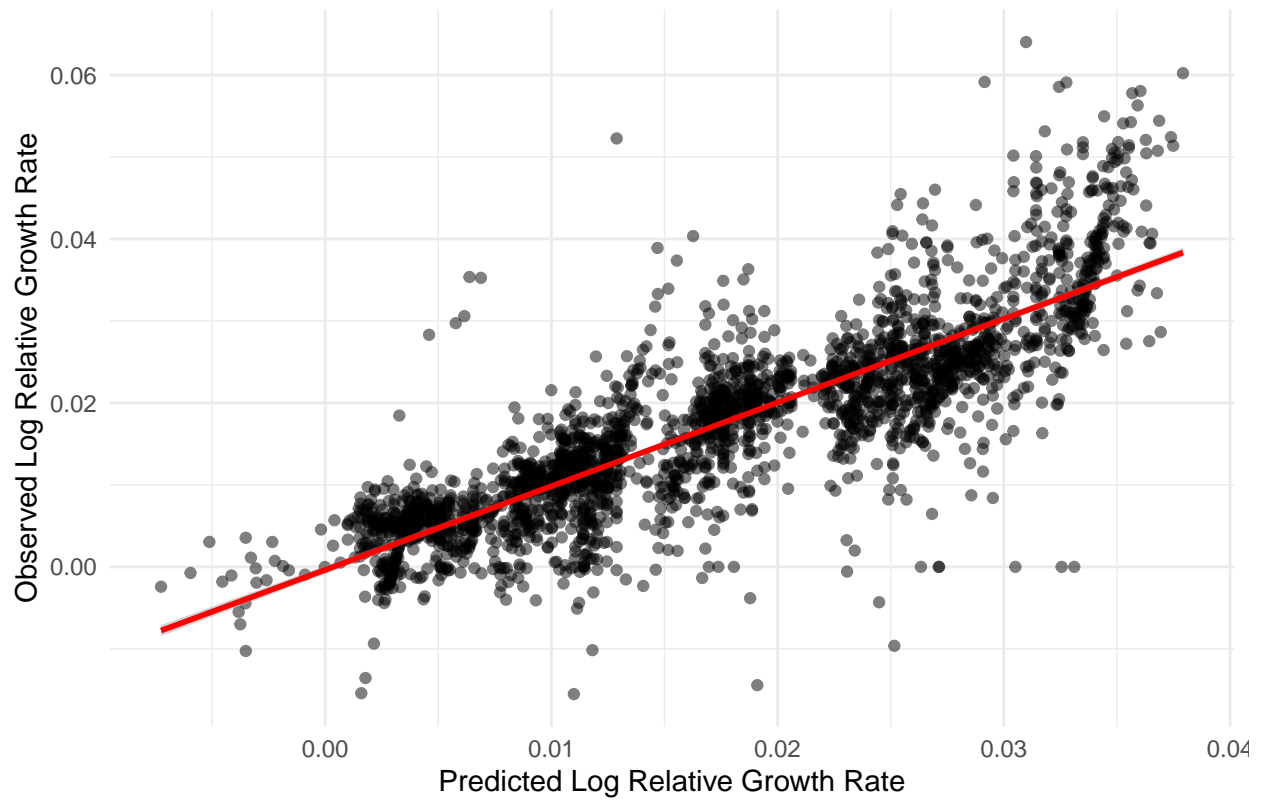
## Extract predictions

```
tpcvis_cleancurrent$predicted <- predict(mod.lmer_final)

ggplot(tpcvis_cleancurrent, aes(x = predicted, y = rgrlog)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Observed vs. Predicted Values",
    x = "Predicted Log Relative Growth Rate",
    y = "Observed Log Relative Growth Rate") +
  theme_minimal()
```

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

### Observed vs. Predicted Values



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.