Fitting Linear Models

Fitting Models to Data Not Data to Models
Model Fitting Series - With Applications in R

Jesse Ghashti

October 8, 2025

Centre for Scholarly Communication
The University of British Columbia | Okanagan Campus | Syilx Okanagan Nation Territory

Welcome, feel free to get ready while we wait to start...

Install Required Packages

```r
packages <- c("dplyr", "mgcv", "ggplot2", "gratia", "faraway")
toInstall <- packages[!(packages %in%
                        installed.packages()[,"Package"])]
if(length(toInstall)) install.packages(toInstall)

library('dplyr')   # for data wrangling
library('mgcv')    # for modeling
library('ggplot2') # for fancy plots
library('gratia')  # for ggplot-based model graphics
library('faraway') # for datasets
```

Workshop Series Overview

Session	Topic	Date/Time
1	**Simple Linear Regression**	**Oct 7, 9:00 AM**
2	Fitting Linear Models in R	Oct 8, 10:30 AM
3	Multiple Linear Regression in R	Oct 16, 4:00 PM
4	Interaction Terms & Hierarchical Linear Models	Oct 21, 11:00 AM
5	Generalized Linear Models	Oct 23, 4:00 PM
6	Generalized Additive Models (GAMs)	Oct 28, 11:00 AM
7	Interpreting & Predicting from GAMs	Oct 29, 10:30 AM
8	Hierarchical GAMs	Nov 4, 12:00 PM
9	Penalized Models	Nov 18, 11:00 AM
10	Survival Models	Nov 25, 11:00 AM
11	Nonparametric Models	Dec 2, 11:00 AM

New Here?

<——————————-

New to R? Check out the Fundamentals of R series!

GitHub code for today's workshop

——————————>

Last Time (Workshop 1) — Quick Recap

Key Concepts

- Simple linear regression: $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$
- Least Squares estimates: $\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2}$, $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$
- Five assumptions: certainty in x, linearity, homoscedasticity, independence, normality

Today: Fitting & Diagnosing Linear Models in R

Today we will...

- Build a reusable diagnostics function
- Fit linear models to real datasets (`ChickWeight`, `prostate`, `women`, `state.x77`)
- Examine transformations and why they may fail
- Emphasize: *Fit the correct model to the data, not the data to the model*

In case you missed it, today we require...

```
library('dplyr')   # for data wrangling
library('mgcv')    # for modeling
library('ggplot2') # for fancy plots
library('gratia')  # for ggplot-based model graphics
library('faraway') # for datasets
theme_set(theme_classic(base_size = 15))
```

Diagnostics Function: Structure & Data Generation

We'll build a function that simulates data, fits a linear model, computes residuals, and plots the 5 assumption views.

Function diagnosing any issues with model assumption violations

```
plot_example_diagnostics <- function(N = 20, seed = as.numeric(Sys.time())) {
  a <- 0.3 + 0.7 / max(1, sqrt(N - 9))
  set.seed(seed)
  d0 <- tibble(x = runif(n = N),                                # predictor of Y
               mu = 4 - 3 * x,                                  # true mean of Y
               epsilon = rnorm(n = length(x), mean = 0, sd = 1), # Gaussian error
               Y = mu + epsilon,                                # values of Y
               mu_hat = predict(lm(Y ~ x)),
               e = Y - mu_hat)
```

Diagnostics Function: Plots (1) Certainty in x; (2) Linearity

Function continued...

```
cowplot::plot_grid(
  #' 1. *Certainty in x*: unlike Y, there is no error or uncertainty in x.
  ggplot(d0) +
    geom_errorbar(aes(x, ymin = Y - 1, ymax = Y + 1), color = 'grey') +
    geom_point(aes(x, Y), alpha = a) +
    labs(x = 'x',  y = 'Y',
         title = expression(E(Y)~'='~mu~but~E(x)~'='~x)),
  #' 2. *Linearity*: The relationship between X and the mean of Y is linear.
  ggplot(d0) +
    geom_line(aes(x, mu), col = 'red', lwd = 1, alpha = 0.5) +
    geom_smooth(aes(x, Y), lwd = 1, method = 'gam', formula = y ~ s(x),
                color = 'darkorange') +
    geom_point(aes(x, Y), alpha = a) +
    labs(x = 'x',  y = 'Y',
         title = expression(E(Y)~'='~mu~'='~beta[0]~+~beta[1]~x)),
```

Diagnostics Function: Plots (3)–(4) Homoscedasticity, Independence

Function continued...

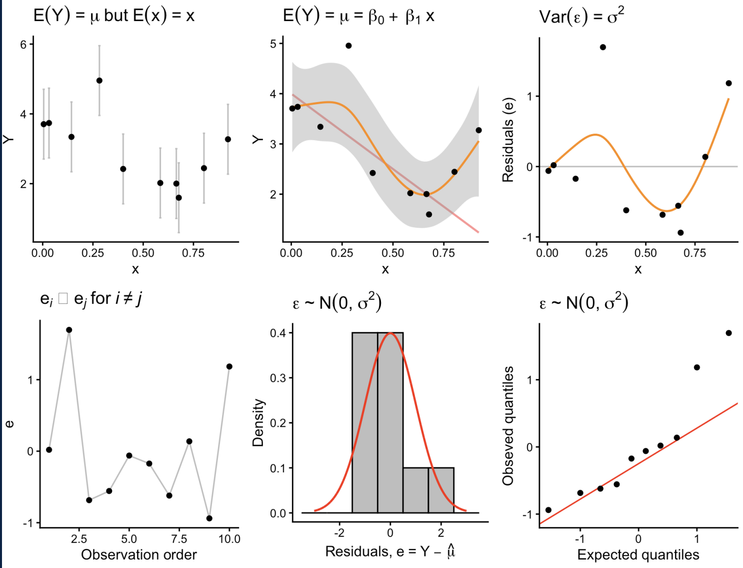
```r
#' 3. *Homoscedasticity*: The variance of the residuals is constant.
ggplot(d0) +
  geom_hline(yintercept = 0, color = 'grey') +
  geom_smooth(aes(x, e), col = 'darkorange', lwd = 1, method = 'gam',
              formula = y ~ s(x), se = FALSE) +
  geom_point(aes(x, e), alpha = a) +
  labs(x = 'x', y = 'Residuals (e)',
       title = expression(Var(epsilon)~'='~sigma^2)),
#' 4. *Independence*: residuals are independent of each other.
ggplot(d0) +
  geom_line(aes(seq(nrow(d0)), e), color = 'grey') +
  geom_point(aes(seq(nrow(d0)), e), alpha = a) +
  labs(x = 'Observation order', y = 'e',
       title = expression(e[italic(i)]~'\U2AEB'~e[italic(j)]~
                          'for'~italic(i)~'\U2260'~italic(j))),
```

Diagnostics Function: Plots (5) Normality

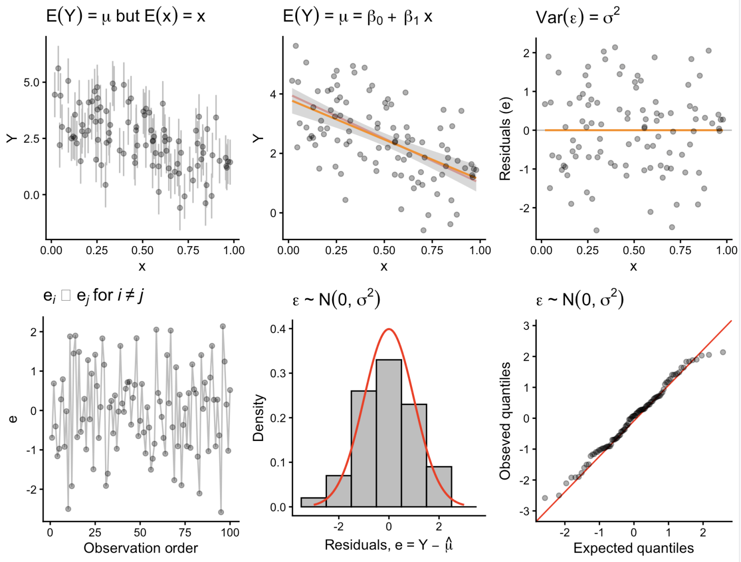
Function continued...

```r
    #' 5. *Normality*: errors follow a Gaussian distribution.
    ggplot(d0, aes(e)) +
      geom_histogram(aes(y = after_stat(density)), color = 'black',
                     fill = 'grey', binwidth = 1) +
      geom_line(aes(x, dens), color = 'red', lwd = 1,
                tibble(x = seq(-3, 3, by = 0.001),
                       dens = dnorm(seq(-3, 3, by = 0.001)))) +
      labs(x = expression('Residuals,'~e~'='~Y~-~hat(mu)),
           y = 'Density',
           title = expression(epsilon~'~'~N('0,'~sigma^2))),
    ggplot(d0, aes(sample = e)) +
      geom_qq_line(color = 'red') +
      geom_qq(color = 'black', alpha = a) +
      labs(x = 'Expected quantiles',
           y = 'Obseved quantiles',
           title = expression(epsilon~'~'~N('0,'~sigma^2)))
}
```

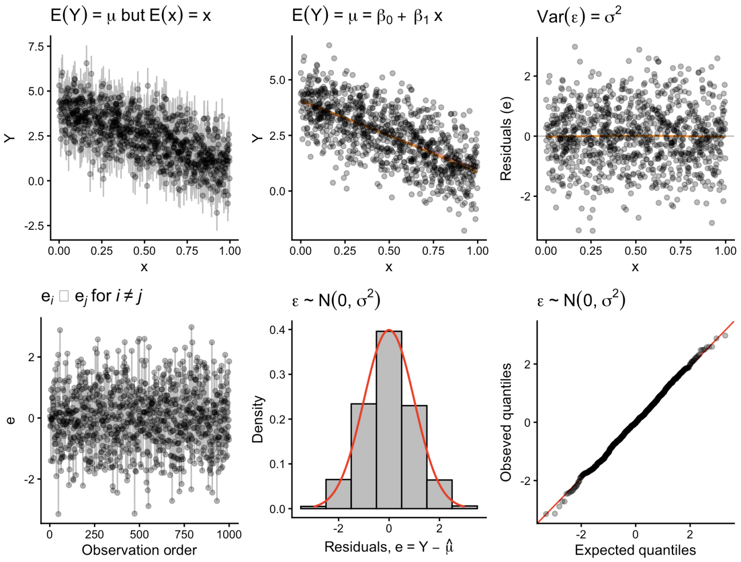
Running the Diagnostics Function ($N = 10$)



Running the Diagnostics Function ($N = 100$)



Running the Diagnostics Function ($N = 100$)

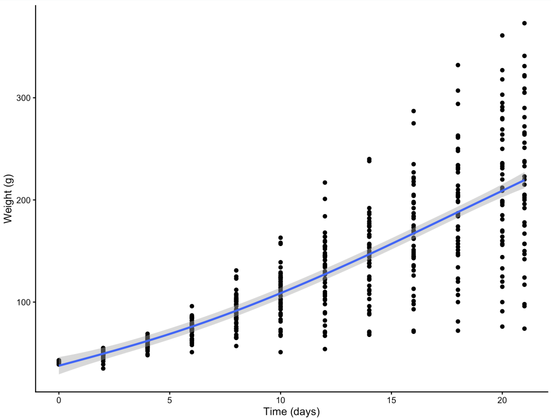


Linear Models in R: `ChickWeight`

ChickWeight Data

```r
?ChickWeight
m_cw <- gam(formula = weight ~ Time,
            family = gaussian(), #linear model
            data = ChickWeight,
            method = 'ML')
# ML = most likely coefficients given data
```

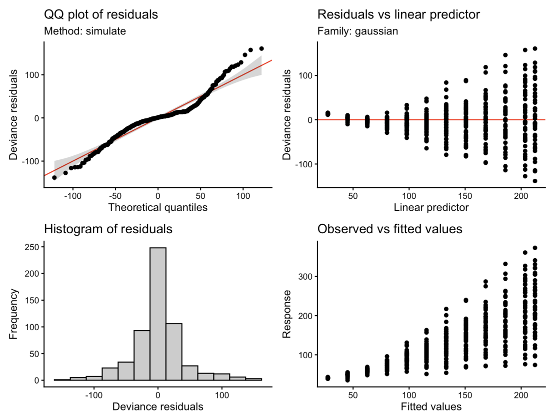
```r
ggplot(ChickWeight, aes(Time, weight)) +
  geom_point() +
  geom_smooth( method = 'gam',
               formula = y ~ s(x)) +
  labs( x = 'Time (days)',
        y = 'Weight (g)')
```



Diagnostics for `ChickWeight`

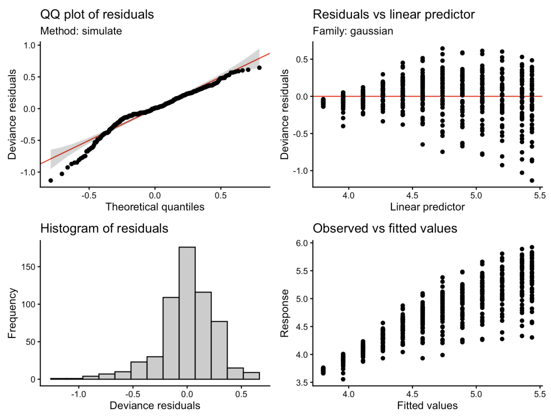
```r
appraise(model = m_cw,
         method = 'simulate',
         n_simulate = 1e4)
```

Watch for nonlinearity, non-constant variance, and structure (e.g., growth curves) that violate simple linearity.



Log Transform ≠ Magic Fix (I)

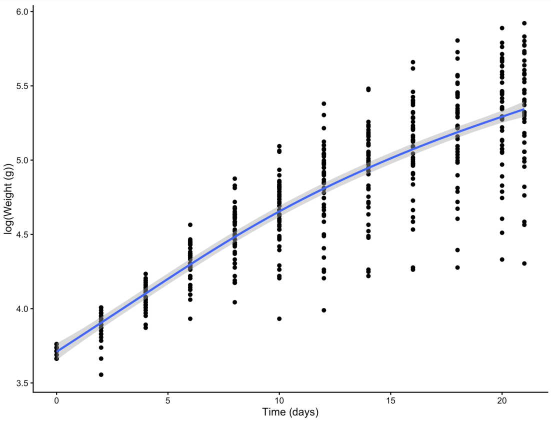
```r
min(ChickWeight$weight)
m_log <- gam(formula = log(weight) ~ Time,
             family = gaussian(),
             data = ChickWeight,
             method = 'ML')
appraise(m_log, method = 'simulate',
         n_simulate = 1e4)
```



Log Transform ≠ Magic Fix (II)

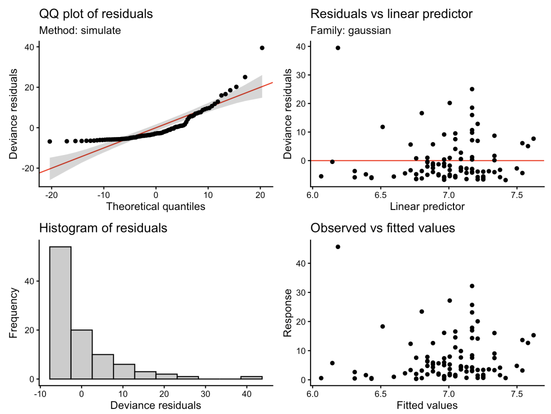
```r
# plot the data
ggplot(ChickWeight, aes(Time,
                        log(weight))) +
  geom_point() +
  geom_smooth(method = 'gam',
              formula = y ~ s(x)) +
  labs(x = 'Time (days)',
       y = 'log(Weight (g))')
```

Transformations can change the question and introduce bias.



Prostate Cancer Data

```r
?prostate
prostate$cavol <- exp(prostate$lcavol)
m_pc <- gam(formula = cavol ~ age,
            family = gaussian(),
            data = prostate,
            method = 'ML')
appraise(m_pc, method = 'simulate',
         n_simulate = 1e4)
```

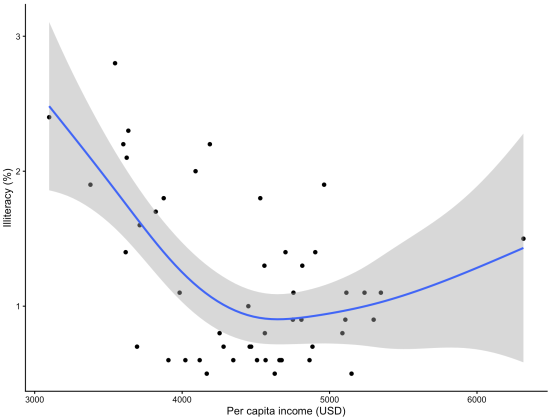


Interpretation: Check linearity and variance patterns; consider link/response choice if diagnostics misbehave.

Income vs Illiteracy (US States, 1970s)

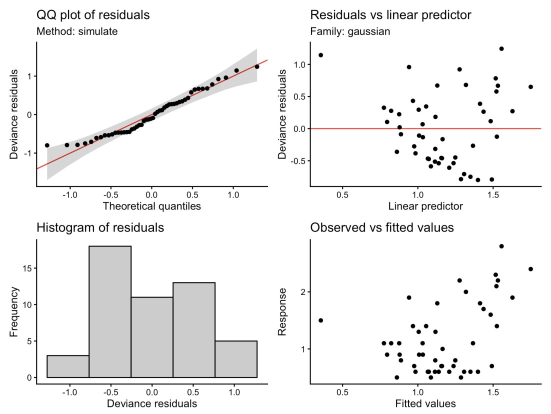
```r
?state.x77 # see notes in details
states <- as.data.frame(state.x77)
# Income: per capita income (1974)
# Illiteracy: illiteracy (1970, % population)

ggplot(states, aes(Income, Illiteracy)) +
  geom_point() +
  geom_smooth(method = 'gam',
              formula = y ~ s(x)) +
  labs(x = 'Per capita income (USD)',
       y = 'Illiteracy (%)')
```



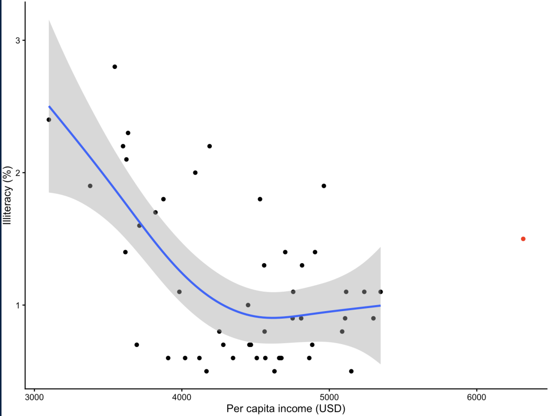
Income vs Illiteracy (US States, 1970s)

```r
m_ii <- gam(formula = Illiteracy ~ Income,
            family = gaussian(),
            data = states,
            method = 'ML')
appraise(m_ii, method = 'simulate',
         n_simulate = 1e4, n_bins = 5)
```



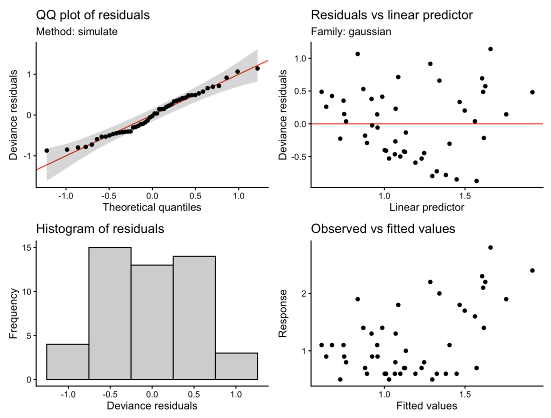
Handling Leverage/Outliers

```
# drop the state with the highest income
ggplot(filter(states, Income < 6000),
              aes(Income, Illiteracy)) +
  geom_point() +
  geom_point(data = filter(states, Income > 6000),
             color = 'red') +
  geom_smooth(method = 'gam',
              formula = y ~ s(x)) +
  labs(x = 'Per capita income (USD)',
       y = 'Illiteracy (%)')
```



Handling Leverage/Outliers

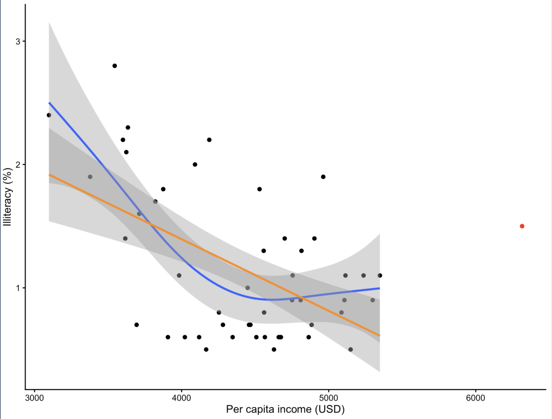
```r
m_ii2 <- gam(formula = Illiteracy ~ Income,
             family = gaussian(),
             data = states,
             subset = Income < 6000,
             method = 'ML')
appraise(m_ii2, method = 'simulate',
         n_simulate = 1e4, n_bins = 5)
```



Compare Smooth vs Linear Trend (Restricted Data)

```
ggplot(filter(states, Income < 6000),
              aes(Income, Illiteracy)) +
  geom_point() +
  geom_point(data = filter(states,
                           Income > 6000),
                           color = 'red') +
  geom_smooth(method = 'gam',
              formula = y ~ s(x)) +
  geom_smooth(method = 'gam',
              formula = y ~ x,
              color = 'darkorange') +
  labs(x = 'Per capita income (USD)',
       y = 'Illiteracy (%)')
```

Question: Is a strictly linear trend adequate, or do we choose the smooth term?



Interpreting `summary()` Output

```r
# interpret linear model summaries ----
# coefficients, df, SE,
# t statistics, p-values, R^2, R^2_adj,
# statistical significance
summary(m_ii2)

# For a single slope the F statistic equals
# the squared t statistic.
# *This does not hold for multiple
# t statistics at once.*

# Generally, the F statistic compares two
# models and assesses whether the addition
# of at least one term in the larger model
# (not in the  simpler model) is significant.
```

```
Family: gaussian
Link function: identity

Formula:
Illiteracy ~ Income

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.7177902  0.6054266   6.141 1.65e-07 ***
Income      -0.0005809  0.0001366  -4.252 9.97e-05 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


R-sq.(adj) =  0.262   Deviance explained = 27.8%
-ML = 37.145  Scale est. = 0.27801   n = 49
```

Question: Is a strictly linear trend adequate, or does a smooth term capture meaningful curvature?

Key Takeaways

- Visual diagnostics are essential: check all five assumptions
- Transformations change the target; use with care (Jensen's)
- Outliers/leverage can dominate inference — inspect, justify, document
- Choose models that match data-generating mechanisms

What’s Next?

Additional Resources:

- *An Introduction to Statistical Learning* (James et al.)
- *Linear Models with R* (Faraway)

Additional Questions? Book an Appointment!

Next Workshop:

Multiple Linear Regression
October 16, 4:00 PM
– Variable Selection
– Multicollinearity
– and more!