Lecture 23: Transformations and Weighted Least Squares

Pratheepa Jeganathan

11/13/2019

Recap

- What is a regression model?
- Descriptive statistics – graphical
- Descriptive statistics – numerical
- Inference about a population mean
- Difference between two population means
- Some tips on R
- Simple linear regression (covariance, correlation, estimation, geometry of least squares)
  - Inference on simple linear regression model
  - Goodness of fit of regression: analysis of variance.
  - F-statistics.
  - Residuals.
  - Diagnostic plots for simple linear regression (graphical methods).

Recap

- Multiple linear regression
  - Specifying the model.
  - Fitting the model: least squares.
  - Interpretation of the coefficients.
  - Matrix formulation of multiple linear regression
  - Inference for multiple linear regression
    - T-statistics revisited.
    - More F statistics.
    - Tests involving more than one β.
- Diagnostics – more on graphical methods and numerical methods
  - Different types of residuals
  - Influence
  - Outlier detection
  - Multiple comparison (Bonferroni correction)
  - Residual plots:
    - partial regression (added variable) plot,
    - partial residual (residual plus component) plot.

Recap

- Adding qualitative predictors
  - Qualitative variables as predictors to the regression model.
  - Adding interactions to the linear regression model.
  - Testing for equality of regression relationship in various subsets of a population
- ANOVA
  - All qualitative predictors.
  - One-way layout
  - Two-way layout

Transformation

Outline

- We have been working with *linear* regression models so far in the course.
- Some models are nonlinear, but can be *transformed* to a linear model (**CH** Chapter 6).
- We will also see that transformations can sometimes *stabilize* the variance making constant variance a more reasonable assumption (**CH** Chapter 6).
- Finally, we will see how to correct for unequal variance using a technique weighted least squares (WLS) (**CH** Chapter 7).

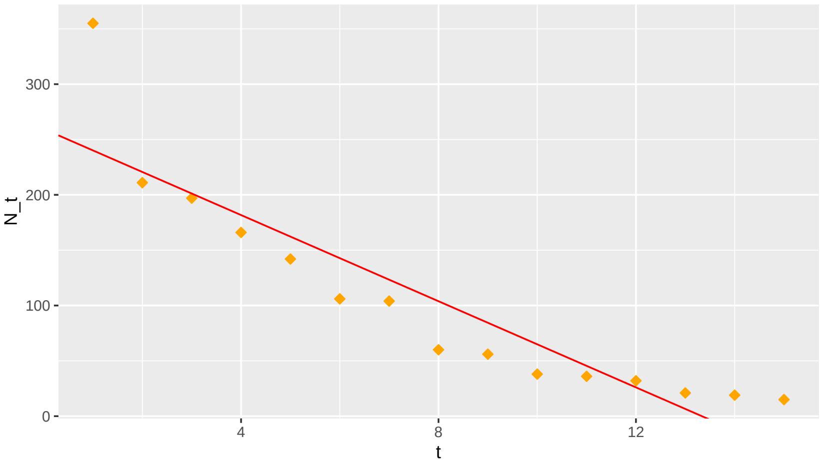
Bacterial colony decay (**CH** Chapter 6.3, Page 167)

- Here is a simple dataset showing the number of bacteria alive in a colony, n_t as a function of time t.
- A simple linear regression model is clearly not a very good fit.

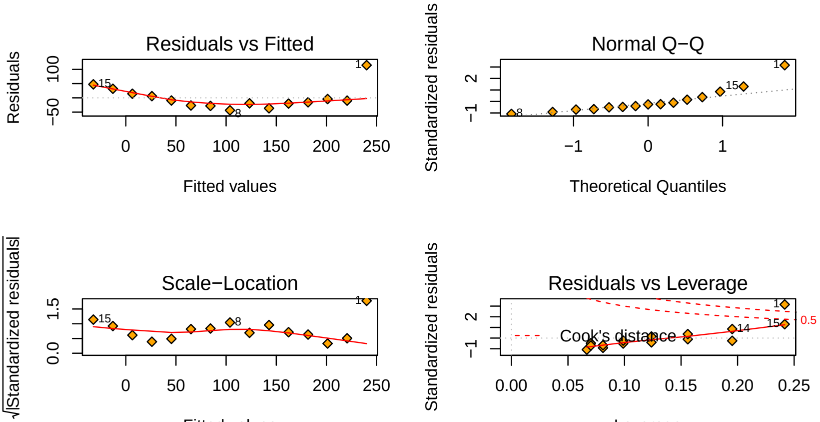
```
bacteria.table = read.table('http://stats191.stanford.edu/c
  header=T)
head(bacteria.table)
```

```
##   t N_t
## 1 1 355
## 2 2 211
## 3 3 197
## 4 4 166
## 5 5 142
## 6 6 106
```

Fitting (Bacterial colony decay)



Diagnostics (Bacterial colony decay)



Exponential decay model

- Suppose the expected number of cells grows like

$$E(n_t) = n_0 e^{\beta_1 t}, \qquad t = 1, 2, 3, \ldots$$

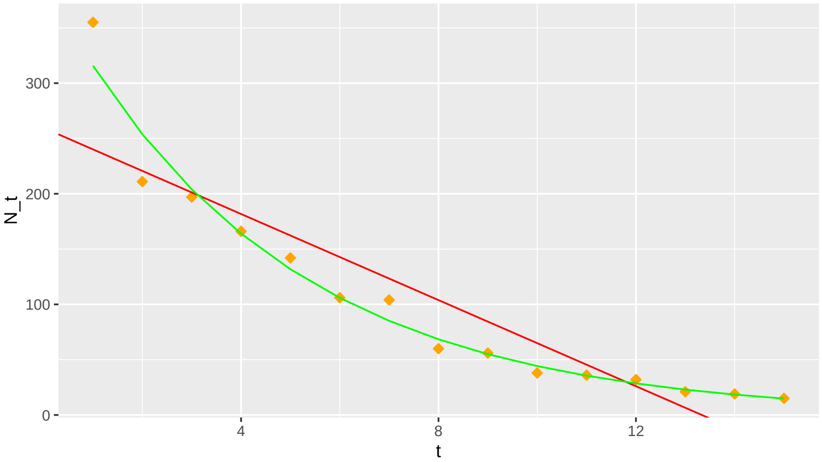
- If we take logs of both sides

$$\log E(n_t) = \log n_0 + \beta_1 t.$$

- A reasonable (?) model:

$$\log n_t = \log n_0 + \beta_1 t + \varepsilon_t, \qquad \varepsilon_t \overset{IID}{\sim} N(0, \sigma^2).$$

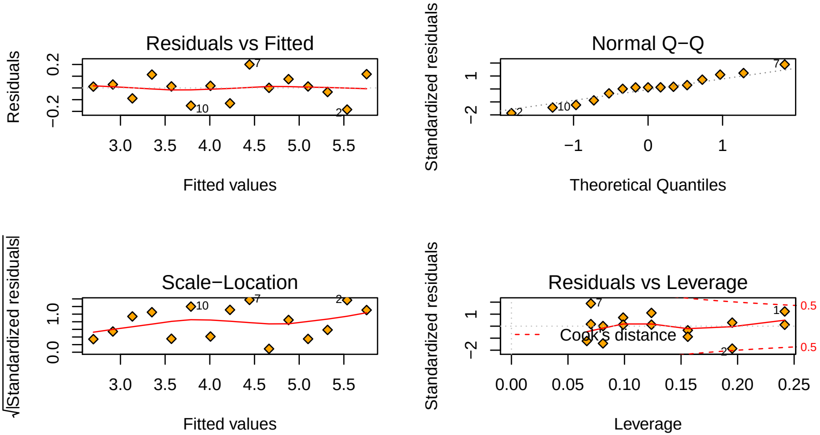
```r
bacteria.log.lm = lm(log(N_t) ~ t, bacteria.table)
df = cbind(bacteria.table,
  lm.fit = fitted(bacteria.lm),
  lm.log.fit = exp(fitted(bacteria.log.lm)))
p = p +
  geom_abline(intercept = bacteria.lm$coef[1],
  slope = bacteria.lm$coef[2],
    col = "red") +
  geom_line(data = df,
    aes(x = t, y = lm.log.fit),
    color = "green")
```



300
200
100
0
N_t
4
8
12
t

Diagnostics

```
par(mfrow=c(2,2))
plot(bacteria.log.lm, pch=23, bg='orange')
```



Logarithmic transformation

- This model is slightly different than original model:

$$E(\log n_t) \leq \log E(n_t)$$

  but may be approximately true.

- If $\varepsilon_t \sim N(0, \sigma^2)$ then

$$n_t = n_0 \cdot \gamma_t \cdot e^{\beta_1 t}.$$

- $\gamma_t = e^{\varepsilon_t}$ is called a log-normal $(0, \sigma^2)$ random variable.

Linearizing regression function

- We see that an exponential growth or decay model can be made (approximately) linear.
- Here are a few other models that can be linearized:
  - $y = \alpha x^{\beta}$, use $\tilde{y} = \log(y), \tilde{x} = \log(x)$;
  - $y = \alpha e^{\beta x}$, use $\tilde{y} = \log(y)$;
  - $y = x/(\alpha x - \beta)$, use $\tilde{y} = 1/y, \tilde{x} = 1/x$.
  - More in textbook.

Caveats

- Just because expected value linearizes, doesn't mean that the errors behave correctly.
- In some cases, this can be corrected using weighted least squares (more later).
- Constant variance, normality assumptions should still be checked.

Stabilizing variance

- Sometimes, a transformation can turn non-constant variance errors to "close to" constant variance. This is another situation in which we might consider a transformation.
- Example: by the "delta rule", if

$$\mathrm{Var}(Y) = \sigma^2 E(Y)$$

then

$$\mathrm{Var}(\sqrt{Y}) \simeq \frac{\sigma^2}{4}.$$

- In practice, we might not know which transformation is best. Box-Cox transformations offer a tool to find a "best" transformation.

Delta rule

The following approximation is ubiquitous in statistics.

- Taylor series expansion:

$$f(Y) = f(E(Y)) + \dot{f}(E(Y))(Y - E(Y)) + \dots$$

- Taking expectations of both sides yields:

$$\mathrm{Var}(f(Y)) \simeq \dot{f}(E(Y))^2 \cdot \mathrm{Var}(Y)$$

Delta rule

- So, for our previous example:

$$\mathrm{Var}(\sqrt{Y}) \simeq \frac{\mathrm{Var}(Y)}{4 \cdot E(Y)}$$

- Another example

$$\mathrm{Var}(\log(Y)) \simeq \frac{\mathrm{Var}(Y)}{E(Y)^2}.$$

Caveats

- Just because a transformation makes variance constant doesn't mean regression function is still linear (or even that it was linear)!
- The models are approximations, and once a model is selected our standard diagnostics should be used to assess adequacy of fit.
- It is possible to have non-constant variance but the variance stabilizing transformation may destroy linearity of the regression function.
  - *Solution:* try weighted least squares (WLS).

Weighted Least Squares (**CH** Chapter 7)

Correcting for unequal variance: weighted least squares

- We will now see an example in which there seems to be strong evidence for variance that changes based on `Region`.
- After observing this, we will create a new model that attempts to *correct* for this and come up with better estimates.
- *Correcting* for unequal variance, as we describe it here, generally requires a model for how the variance depends on observable quantities.

Correcting for unequal variance: weighted least squares (**CH** Chapter 7.4, Page 197)

Variable	**Description**
Y	Per capita education expenditure by state
X_1	Per capita income in 1973 by state
X_2	Proportion of population under 18
X_3	Proportion in urban areas
Region	Which region of the country are the states located in

lm

```
education.table = read.table('http://stats191.stanford.edu/
education.table$Region = factor(education.table$Region)
education.lm = lm(Y ~ X1 + X2 + X3, data=education.table)
```

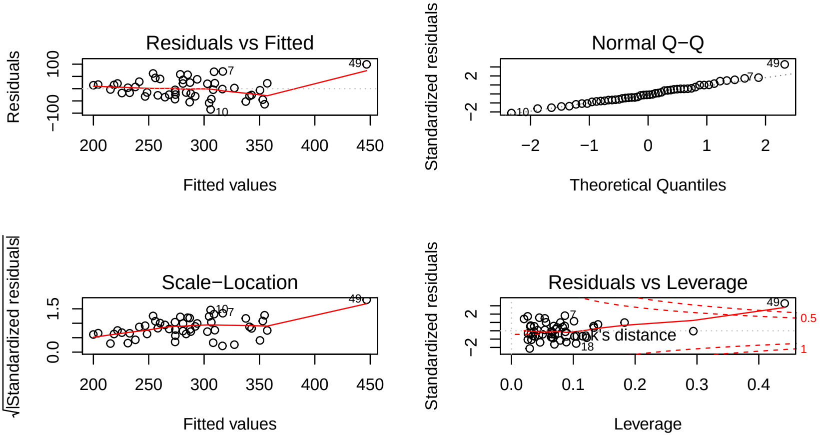
lm

```r
summary(education.lm)
```

```
Call:
lm(formula = Y ~ X1 + X2 + X3, data = education.table)

Residuals:
    Min      1Q  Median      3Q     Max 
-84.878 -26.878  -3.827  22.246  99.243 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.566e+02  1.232e+02  -4.518 4.34e-05
X1           7.239e-02  1.160e-02   6.239 1.27e-07
X2           1.552e+00  3.147e-01   4.932 1.10e-05
X3          -4.269e-03  5.139e-02  -0.083    0.934
               
(Intercept) ***
X1          ***
X2          ***
X3             
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 40.47 on 46 degrees of freedom
Multiple R-squared:  0.5913,	Adjusted R-squared:  0.5647 
F-statistic: 22.19 on 3 and 46 DF,  p-value: 4.945e-09
```

Diagnostics

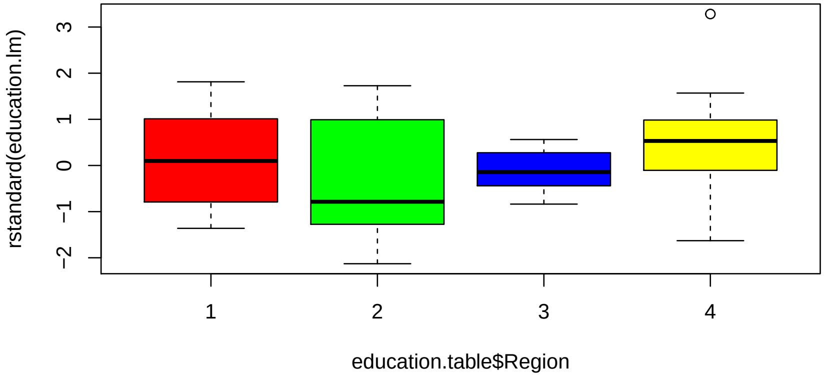
```r
par(mfrow=c(2,2))
plot(education.lm)
```



Diagnostics

- there is an outlier, let's drop the outlier and refit

```
boxplot(rstandard(education.lm) ~ education.table$Region,
  col=c('red', 'green', 'blue', 'yellow'))
```



Fit a model without the outlier

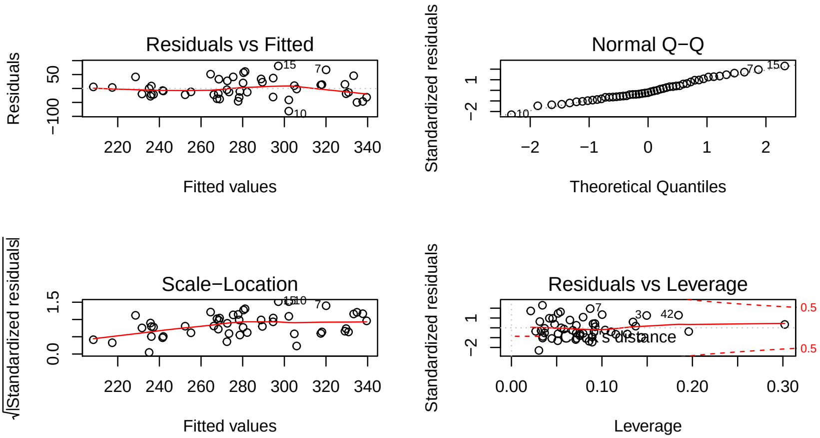
```
keep.subset = (education.table$STATE != 'AK')
education.noAK.lm = lm(Y ~ X1 + X2 + X3,
  subset=keep.subset,
  data=education.table)
```

Fit a model without the outlier

```r
summary(education.noAK.lm)
```

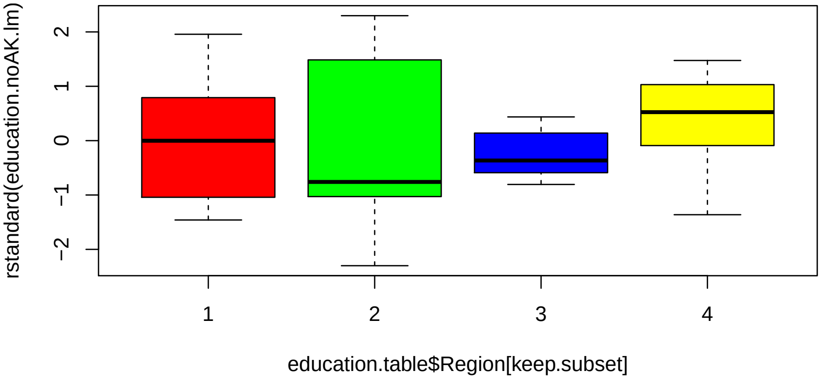
```
Call:
lm(formula = Y ~ X1 + X2 + X3, data = education.table, subset = keep.subset)

Residuals:
    Min      1Q  Median      3Q     Max
-81.128 -22.154  -7.542  22.542  80.890

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -277.57731  132.42286  -2.096 0.041724
X1             0.04829    0.01215   3.976 0.000252
X2             0.88693    0.33114   2.678 0.010291
X3             0.06679    0.04934   1.354 0.182591

(Intercept) *
X1          ***
X2          *
X3
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 35.81 on 45 degrees of freedom
Multiple R-squared:  0.4967,    Adjusted R-squared:  0.4631
F-statistic:  14.8 on 3 and 45 DF,  p-value: 7.653e-07
```

```r
par(mfrow=c(2,2))
plot(education.noAK.lm)
```



Diagnostics (refitted model)

```
par(mfrow=c(1,1))
boxplot(rstandard(education.noAK.lm) ~ education.table$Regi
  col=c('red', 'green', 'blue', 'yellow'))
```



Re-weighting observations

- If you have a reasonable guess of variance as a function of the predictors, you can use this to *re-weight* the data.
- Hypothetical example

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2 X_i^2).$$

- Setting $\tilde{Y}_i = Y_i / X_i$, $\tilde{X}_i = 1/X_i$, model becomes

$$\tilde{Y}_i = \beta_0 \tilde{X}_i + \beta_1 + \gamma_i, \gamma_i \sim N(0, \sigma^2).$$

Weighted Least Squares

- Fitting this model is equivalent to minimizing

$$\sum_{i=1}^{n} \frac{1}{X_i^2} \left(Y_i - \beta_0 - \beta_1 X_i\right)^2$$

- Weighted Least Squares

$$SSE(\beta, w) = \sum_{i=1}^{n} w_i \left(Y_i - \beta_0 - \beta_1 X_i\right)^2, \qquad w_i = \frac{1}{X_i^2}.$$

- In general, weights should be like:

$$w_i = \frac{1}{\mathrm{Var}(\varepsilon_i)}.$$

- Our education expenditure example assumes

$$w_i = W_{\texttt{Region[i]}}$$

Common weighting schemes

- If you have a qualitative variable, then it is easy to estimate weight within groups (our example today).
- "Often"
$$\mathrm{Var}(\varepsilon_i) = \mathrm{Var}(Y_i) = V(E(Y_i))$$
- Many non-Gaussian (non-Normal) models behave like this: logistic, Poisson regression.

What if we didn't re-weight?

- Our (ordinary) least squares estimator with design matrix X is

$$\hat{\beta} = \hat{\beta}_{OLS} = (X^T X)^{-1} X^T Y = \beta + (X^T X)^{-1} X^T \epsilon.$$

- Our model says that $\varepsilon | X \sim N(0, \sigma^2 X)$ so

$$\begin{aligned} E[(X^T X)^{-1} X^T \epsilon] &= E[(X^T X)^{-1} X^T \epsilon | X] \\ &= 0 \end{aligned}$$

So the **OLS estimator is unbiased.**

- Variance of $\hat{\beta}_{OLS}$ is

$$\mathrm{Var}((X^T X)^{-1} X^T \epsilon) = \sigma^2 (X^T X)^{-1} X^T V X (X^T X)^{-1},$$

where $V = \mathrm{diag}(X_1^2, \ldots, X_n^2)$.

Two-stage procedure

- Suppose we have a hypothesis about the weights, i.e. they are constant within Region, or they are something like

$$w_i^{-1} = \mathrm{Var}(\epsilon_i) = \alpha_0 + \alpha_1 X_{i1}^2.$$

- We pre-whiten:

  1. Fit model using OLS (Ordinary Least Squares) to get initial estimate $\widehat{\beta}_{OLS}$
  2. Use predicted values from this model to estimate w_i.
  3. Refit model using WLS (Weighted Least Squares).
  4. If needed, iterate previous two steps.

Example (two-stage procedure)

- Let's use $w_i^{-1} = \text{Var}(\epsilon_i)$.

```
# Weight vector for each observation
educ.weights = 0 * education.table$Y
for (region in levels(education.table$Region)) {
  # remove the outlier Alaska
  subset.region = (education.table$Region[
    keep.subset] == region)

  educ.weights[subset.region] = 1.0/(sum(resid(
    education.noAK.lm)[
    subset.region]^2) /sum(subset.region))
}
```

Example (two-stage procedure)

- Weights for the observations in each `Region`

```
unique(educ.weights)
```

```
## [1] 0.0006891263 0.0004103443 0.0040090885 0.0010521628
```

Weighted least squares regression

- Here is our new model.
  - Note that the scale of the estimates is *unchanged*.
  - Numerically the estimates are similar.
  - What changes most is the `Std. Error` column.

```
education.noAK.weight.lm =lm(Y ~ X1 + X2 + X3,
                              weights=educ.weights,
                              subset=keep.subset,
                              data=education.table)
```

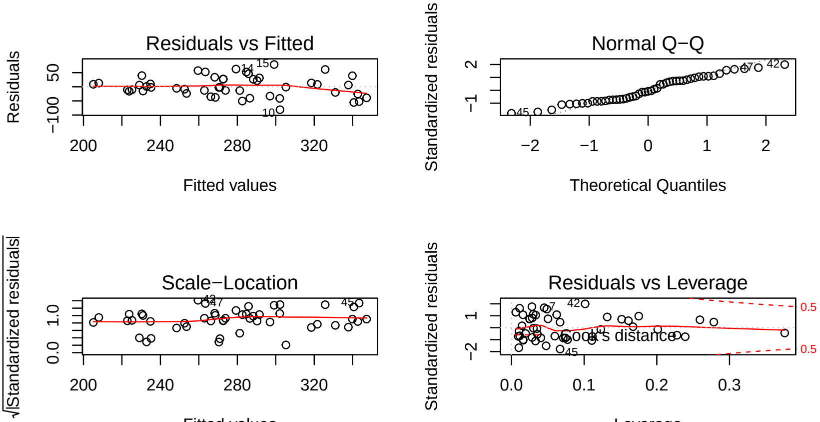
Weighted least squares regression

```r
summary(education.noAK.weight.lm)
```

```
Call:
lm(formula = Y ~ X1 + X2 + X3, data = education.table, subset = keep.subset,
    weights = educ.weights)

Weighted Residuals:
     Min       1Q   Median       3Q      Max
-1.69882 -0.71382 -0.07928  0.79298  1.86328

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.181e+02  7.833e+01  -4.060 0.000193
X1           6.245e-02  7.867e-03   7.938 4.24e-10
X2           8.791e-01  2.003e-01   4.388 6.83e-05
X3           2.981e-02  3.421e-02   0.871 0.388178

(Intercept) ***
X1          ***
X2          ***
X3
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.984 on 45 degrees of freedom
Multiple R-squared:  0.7566,    Adjusted R-squared:  0.7404
F-statistic: 46.63 on 3 and 45 DF,  p-value: 7.41e-14
```

Diagnostics

```
par(mfrow=c(2,2))
plot(education.noAK.weight.lm)
```

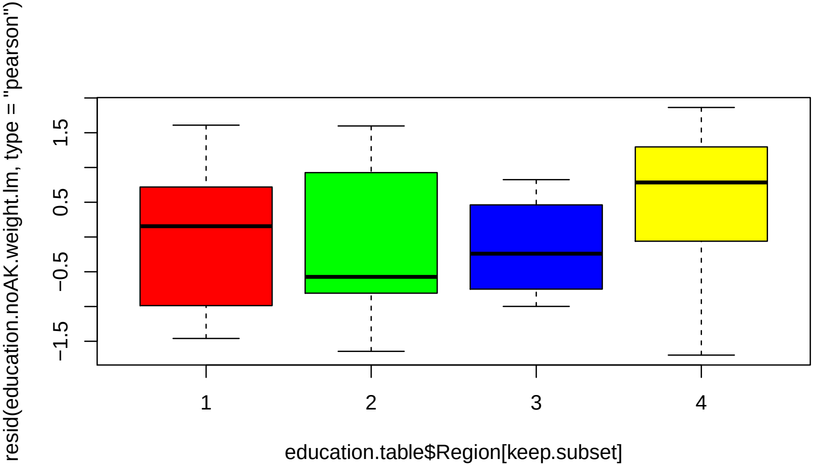


Diagnostics

- Let's look at the boxplot again. It looks better, but perhaps not perfect.

```
par(mfrow=c(1,1))
boxplot(resid(education.noAK.weight.lm,
              type='pearson') ~
        education.table$Region[keep.subset],
      col=c('red', 'green', 'blue', 'yellow'))
```

Diagnostics



Unequal variance: effects on inference

- So far, we have just mentioned that things *may* have unequal variance, but not thought about how it affects inference.
- In general, if we ignore unequal variance, our estimates of variance are not very good. The covariance has the "sandwich form" we saw above

n by n diagonal matrix

$$\mathrm{Cov}(\widehat{\beta}_{OLS}) = (X'X)^{-1}(X'W^{-1}X)(X'X)^{-1}.$$

with $W = \mathrm{diag}(1/\sigma_i^2)$.

- ** If our `Std. Error` is incorrect, so are our conclusions based on t-statistics!**
- In the education expenditure data example, correcting for weights seemed to make the t-statistics larger. ** This will not always be the case!**

Unequal variance: effects on inference

- Weighted least squares estimator

$$\hat{\beta}_{WLS} = (X^T W X)^{-1} X^T W Y$$

- If we have the correct weights, then

$$\text{Cov}(\widehat{\beta}_{WLS}) = (X^T W X)^{-1}.$$

Efficiency

- The efficiency of an unbiased estimator of β is 1 / variance.
- Estimators can be compared by their efficiency: the more efficient, the better.
- The other reason to correct for unequal variance (besides so that we get valid inference) is for efficiency.

Illustrative example

- Suppose

$$Z_i = \mu + \varepsilon_i, \qquad \varepsilon_i \sim N(0, i^2 \cdot \sigma^2), 1 \le i \le n.$$

- Three **unbiased** estimators of μ:

$$\widehat{\mu}_1 = \frac{1}{n}\sum_{i=1}^{n} Z_i$$

$$\widehat{\mu}_2 = \frac{1}{\sum_{i=1}^{n} i^{-2}}\sum_{i=1}^{n} i^{-2} Z_i$$

$$\widehat{\mu}_3 = \frac{1}{\sum_{i=1}^{n} i^{-1}}\sum_{i=1}^{n} i^{-1} Z_i$$

Illustrative example

- The estimator $\widehat{\mu}_2$ will always have lower variance, hence tighter confidence intervals.
- The estimator $\widehat{\mu}_3$ has incorrect weights, but they are "closer" to correct than the naive mean's weights which assume each observation has equal variance.

Illustrative example

```
ntrial = 1000   # how many trials will we be doing?
nsample = 20   # how many points in each trial
sd = c(1:20)   # how does the variance change
mu = 2.0

get.sample = function() {
  return(rnorm(nsample)*sd + mu)
}

unweighted.estimate = numeric(ntrial)
weighted.estimate = numeric(ntrial)
suboptimal.estimate = numeric(ntrial)
```

Illustrative example

- Let's simulate a number of experiments and compare the three estimates.

```
for (i in 1:ntrial) {
  cur.sample = get.sample()
  unweighted.estimate[i] = mean(cur.sample)
  weighted.estimate[i] = sum(cur.sample/sd^2) / sum(1/sd^2
  suboptimal.estimate[i] = sum(cur.sample/sd) / sum(1/sd)
}
```

Illustrative example

- Compute $\mathrm{SE}(\hat{\mu}_i)$

```r
data.frame(mean(unweighted.estimate),
           sd(unweighted.estimate))
```

```
##   mean.unweighted.estimate. sd.unweighted.estimate.
## 1                  1.991227                2.590985
```

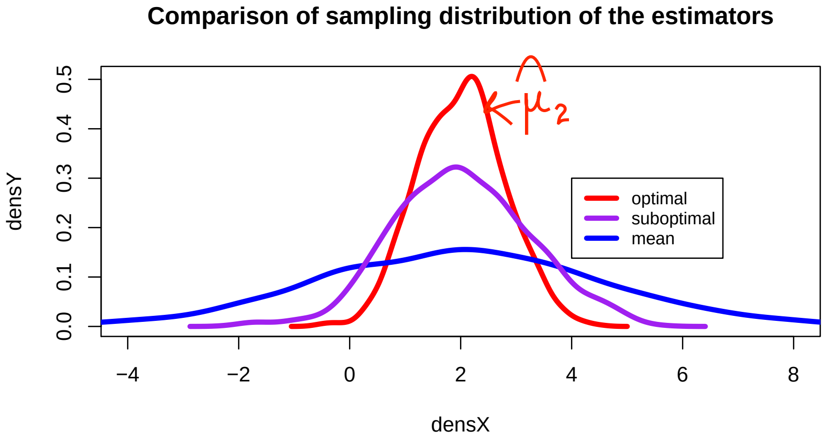
```r
data.frame(mean(weighted.estimate),
           sd(weighted.estimate))
```

```
##   mean.weighted.estimate. sd.weighted.estimate.
## 1                2.014456             0.7788919
```

```r
data.frame(mean(suboptimal.estimate),
           sd(suboptimal.estimate))
```

```
##   mean.suboptimal.estimate. sd.suboptimal.estimate.
## 1                  2.016207                1.227501
```

Illustrative example



Reference

- **CH** Chapter 6 and Chapter 7
- Lecture notes of Jonathan Taylor .