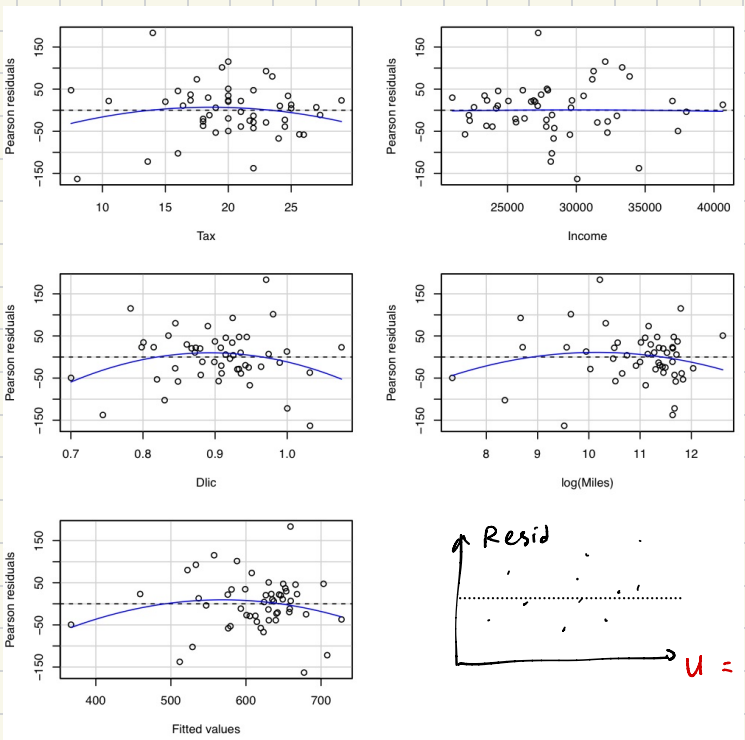
11/5 More on diagnostics + transformation

Testing for curvature

e.g. fuel consumption data (in 2001)

MLR model: $fuel = \beta_0 + \beta_1 tax + \beta_2 inc + \beta_3 dlic + \beta_4 \log(Miles) + \varepsilon$

```
> lmod <- lm(Fuel~Tax+Income+Dlic+log(Miles),fuel2001)
library(alr4)
residualPlots(lmod)
```



Blue curves assess whether there is a non-linear (quadratic) pattern in the scatter plot

```
> residualPlots(lmod)
             Test stat Pr(>|Test stat|)
Tax            -1.0767           0.28737
Income         -0.0840           0.93345
Dlic           -1.9219           0.06096
log(Miles)     -1.3473           0.18463
Tukey test     -1.4460           0.14818
```


U = predictor or combination of predictors

Intuition: If e.g. $y = \beta_0 + \beta_1 tax + \beta_2 income + \beta_3 dlic + \beta_4 \log(Miles) + \beta_5 dlic^2 + \varepsilon$

then residuals $y - \hat{y} \approx \beta_5 dlic^2 + \varepsilon$ will show a quadratic relation with dlic if $\beta_5 \neq 0$

To test whether the estimated curvature can be attributed random fluctuation, try:

1) Refitting the model, including u^2 as additional predictor
2) Test whether the true coefficient for u^2 is zero

e.g.

```
lmod2 <- lm(Fuel~Tax+Income+Dlic+I(Dlic^2)+log(Miles),fuel2001)
summary(lmod2)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.705e+03  9.858e+02  -1.730   0.0905 .
Tax         -4.636e+00  1.985e+00  -2.336   0.0240 *
Income      -5.392e-03  2.167e-03  -2.488   0.0166 *
Dlic         4.812e+03  2.262e+03   2.128   0.0389 *
I(Dlic^2)   -2.426e+03  1.262e+03  -1.922   0.0610 .
log(Miles)   1.928e+01  9.874e+00   1.952   0.0571 .
```

Bonferroni correction $4(0.061) \rightsquigarrow p = 0.24 \rightsquigarrow$ Don't reject $H_0: \beta_{Dlic^2} = 0$

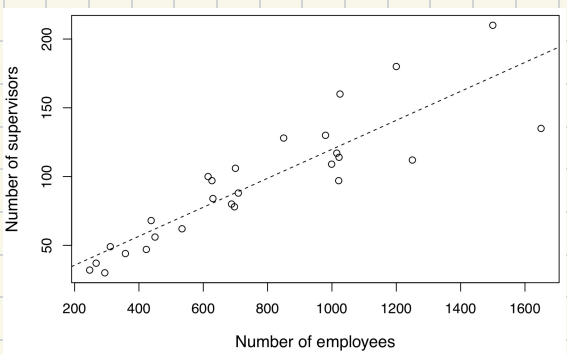
e.g. supervisor data

For $n = 27$ industrial establishments, record

- Number of supervised workers (X)
- Number of supervisors (Y)

```
> supervisor.df <- read.table("supervisor.txt",header=T)
> head(supervisor.df)
    X  Y
1 294 30
2 247 32
3 267 37
4 358 44
5 423 47
6 311 49
```

```
plot(supervisor.df$X,supervisor.df$Y,
     xlab="Number of employees",
     ylab="Number of supervisors",cex.lab=1.2)
abline(lm(Y~X,supervisor.df),lty=2)
```

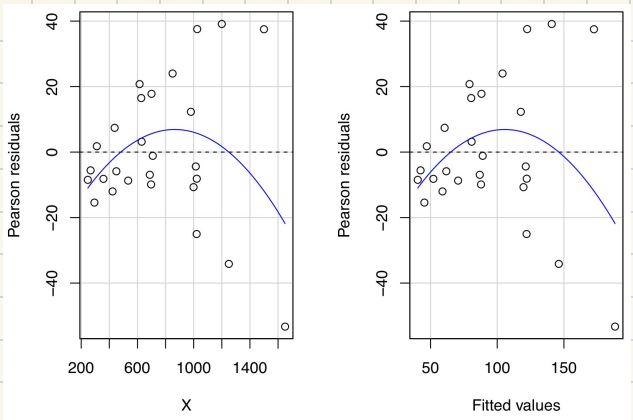


Fit SLR $y \sim x$

```
> lmod <- lm(Y~X,supervisor.df)
> summary(lmod)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.44806    9.56201   1.511    0.143
X            0.10536    0.01133   9.303 1.35e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.73 on 25 degrees of freedom
Multiple R-squared:  0.7759,    Adjusted R-squared:  0.7669
F-statistic: 86.54 on 1 and 25 DF,  p-value: 1.35e-09
```

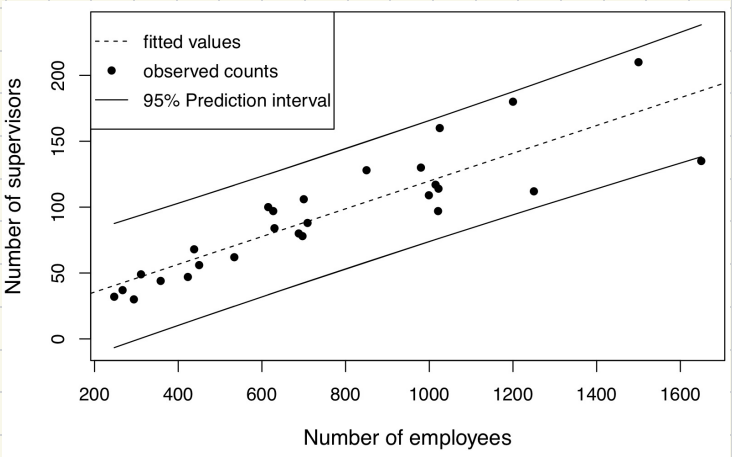
```
residualPlots(lmod)
```



Observations: 1) Slight non-linearity

2) Variance increases with X

Q: What happens if we ignore these issues?



2) Pred. interval is too wide for small X, and barely covers Y for large X (too narrow)

1) If the mean function is non-linear, then PI is not centered correctly

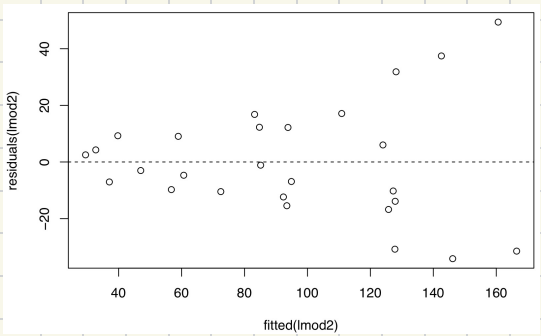
```
pred.int <- predict(lmod,supervisor.df,interval="prediction",level=0.95)
lower <- pred.int[,2]
upper <- pred.int[,3]
plot(supervisor.df$X,supervisor.df$Y,pch=16,
     xlab="Number of employees",
     ylab="Number of supervisors",cex.lab=1.2,ylim=c(min(lower),max(upper)))
points(supervisor.df$X,lower,type="l",lty=1,lwd=1)
points(supervisor.df$X,upper,type="l",lty=1,lwd=1)
abline(lm(Y~X,supervisor.df),lty=2)
legend("topleft",c("fitted values","observed counts","95% Prediction interval"),
       lty=c(2,NA,1),pch=c(NA,16,NA),cex=1)
```

~> Try $Y \sim X + X^2$ & check residual plot

```
lmod2 <- lm(Y~X+I(X^2),supervisor.df)
summary(lmod2)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.365e+01  1.792e+01  -0.762 0.453518
X            1.862e-01  4.564e-02   4.080 0.000431 ***
I(X^2)      -4.669e-05  2.561e-05  -1.824 0.080704 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.78 on 24 degrees of freedom
Multiple R-squared:  0.8031,    Adjusted R-squared:  0.7867
F-statistic: 48.96 on 2 and 24 DF,  p-value: 3.386e-09
```

```
plot(fitted(lmod2),residuals(lmod2))
abline(h=0,lty=2)
```



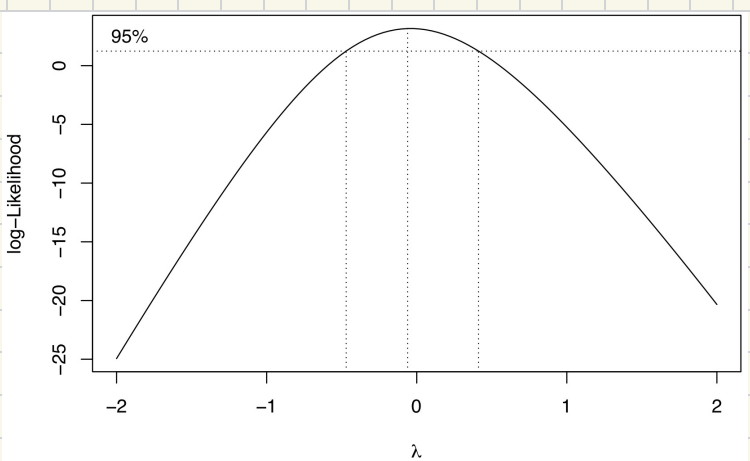
Non-linearity disappears, but

variance is still non-constant ~> try a transformation for y

Box-Cox method can help choose which one

Recall: Choose λ to make $y^\lambda = X\beta + \varepsilon$ as close to normal with const. variance as possible

```
library(MASS)
boxcox(lmod2)
```



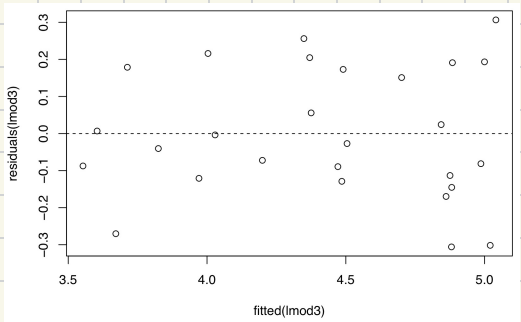
95% CI contains 0 and excludes 1 (a transform is needed)

~> Choose $\lambda = 0$ (log-transform) so $\tilde{y} = \log(y)$

```
> lmod3 <- lm(log(Y)~X+I(X^2),supervisor.df)
> summary(lmod3)
```

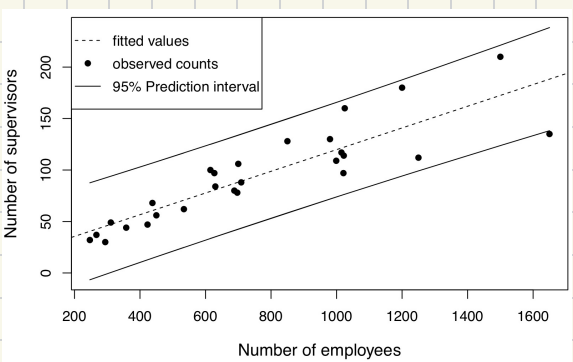
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.852e+00  1.566e-01  18.205 1.50e-15 ***
X           3.113e-03  3.989e-04   7.803 4.90e-08 ***
I(X^2)     -1.102e-06  2.238e-07  -4.925 5.03e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1817 on 24 degrees of freedom
Multiple R-squared:  0.8857,    Adjusted R-squared:  0.8762
F-statistic: 92.98 on 2 and 24 DF,  p-value: 4.976e-12
```

```
plot(fitted(lmod3),residuals(lmod3))
abline(h=0,lty=2)
```

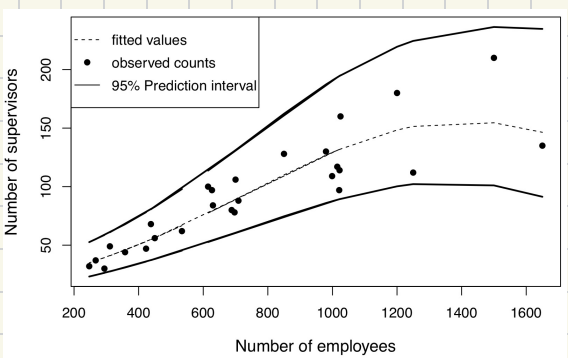


⇝ Issues of non-linearity + non-const. variance are much alleviated

Before



After



R code for right plot:

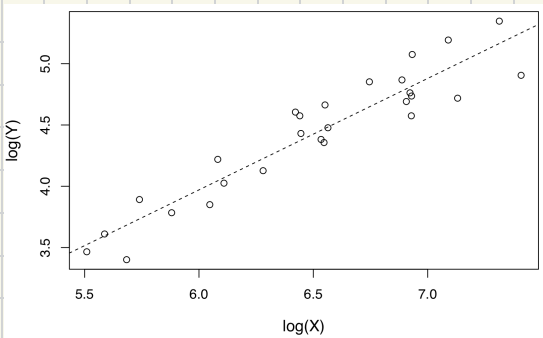
```
pred.int <- predict(lmod3,data.frame(X=supervisor.df$X),interval="prediction",level=0.95)
fitted.y <- exp(pred.int[,1])
lower <- exp(pred.int[,2])
upper <- exp(pred.int[,3])
plot(supervisor.df$X,fitted.y,
     type="l",lty=2,xlab="Number of employees",
     ylab="Number of supervisors",cex.lab=1.2,ylim=c(min(lower),max(upper)))
points(supervisor.df$X,lower,type="l",lty=1,lwd=2)
points(supervisor.df$X,upper,type="l",lty=1,lwd=2)
points(supervisor.df$X,supervisor.df$Y,pch=16)
legend("topleft",c("fitted values","observed counts","95% Prediction interval"),
       lty=c(2,NA,1),pch=c(NA,16,NA),cex=1)
```

Minor complaint: After the transformation, the fitted mean function (dashed line) is not monotone in X, i.e. more employees can lead to fewer supervisors... (non-intuitive)

Alternative transformation : $\tilde{x} = \log x$, $\tilde{y} = \log y$

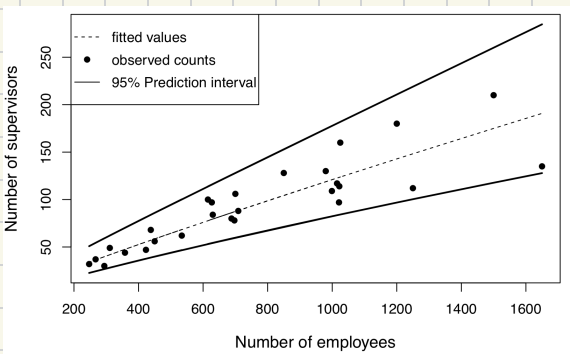
```
lmod4 <- lm(log(Y)~log(X),supervisor.df)
summary(lmod4)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.48458    0.43544  -3.409  0.00221 **
log(X)       0.90920    0.06673  13.625 4.51e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1814 on 25 degrees of freedom
Multiple R-squared:  0.8813,    Adjusted R-squared:  0.8766
F-statistic: 185.6 on 1 and 25 DF,  p-value: 4.508e-13
```



⇝ An increase in X by 1% is associated with an increase in y by 0.91 % (monotonic)

Original scale:



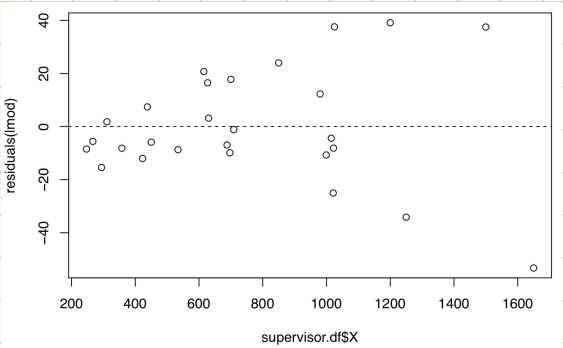
⇝ Transformation can improve the fit, but can be hard to interpret (beyond the special case of log-transform)

Remember to report results (e.g. prediction intervals) on the original scale of the data.

If no appropriate transformation can be found, we may try weighted least squares (next time)

Weighted least squares (WLS)

Recall: original residual plot for supervisor data



⇝ The vertical spread of points increases as X increases

Consider: $\sigma(x) = cx$ for some $c>0$

Idea: Observations with smaller x have lower noise ⇝ give them more weight than the noisier observations

define this to be proportional to precision = $1/\sigma^2 \propto 1/x^2$

In R:

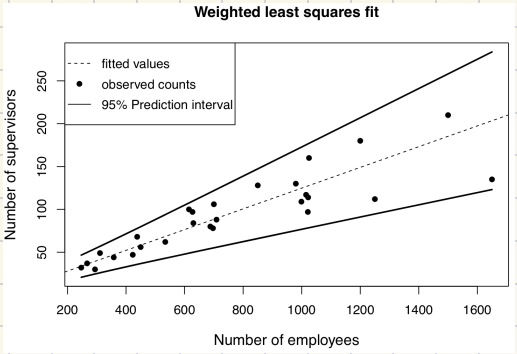
```
weighted.lmod <- lm(Y~X,supervisor.df,weights=1/X^2)
summary(weighted.lmod)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.803296   4.569745   0.832    0.413
X           0.120990   0.008999  13.445 6.04e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02266 on 25 degrees of freedom
Multiple R-squared:  0.8785,    Adjusted R-squared:  0.8737
F-statistic: 180.8 on 1 and 25 DF,  p-value: 6.044e-13
```

Interpretation: For an additional 100 employees, we would need to hire ≈ 12 supervisors on average

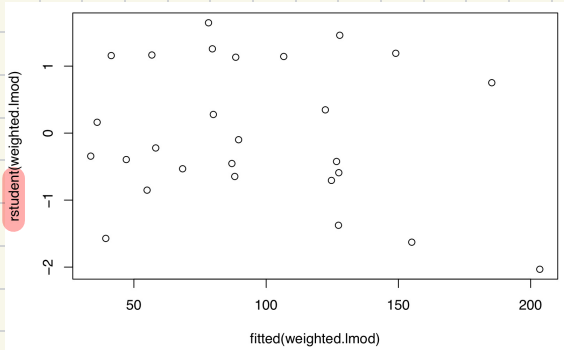
⇝ Coefficients in WLS have the usual (additive) interpretation

Prediction with WLS:




```
pred.int <- predict(weighted.lmod,supervisor.df,interval="prediction",
                    level=0.95,weights=1/supervisor.df$X^2)
lower <- pred.int[,2]
upper <- pred.int[,3]
plot(supervisor.df$X,supervisor.df$Y,pch=16,
     xlab="Number of employees",main="Weighted least squares fit",
     ylab="Number of supervisors",cex.lab=1.2,ylim=c(min(lower),max(upper)))
points(supervisor.df$X,lower,type="l",lty=1,lwd=2)
points(supervisor.df$X,upper,type="l",lty=1,lwd=2)
abline(weighted.lmod,lty=2)
legend("topleft",c("fitted values","observed counts","95% Prediction interval"),
       lty=c(2,NA,1),pch=c(NA,16,NA),cex=1)
```

```
plot(fitted(weighted.lmod),rstudent(weighted.lmod))
```



Observations: 1) WLS fitted line is close to (but slightly steeper than) the OLS fitted line (and PIs are much different)

2) We plot studentized residuals to assess WLS fit (not raw)

3) R summary for WLS in same format as for OLS, but $\hat{\sigma}$ is much smaller for WLS (would seem to contradict nearly equal fitted lines)

Q: What do the numbers mean for WLS?

Theory for WLS

Suppose $y_i = \beta^T x_i + \varepsilon_i$ where $\varepsilon_i \sim N(0, \sigma_i^2)$ independently and $\sigma_i^2 = \sigma^2 / w_i$ can now be different for each $i = 1, \ldots, n$

Assume $w_1, \ldots, w_n$ are known in advance

Q: Can this ↑ information be used to improve upon OLS? Yes!

Weighted RSS: $RSS_w(\beta) = \sum_{i=1}^{n} w_i (y_i - \beta^T x_i)^2$ $\quad (w_i \propto 1/\sigma_i^2)$

Definition/Claim: $\hat{\beta}^{WLS} = \underset{b_0,\dots,b_p}{\operatorname{argmin}} RSS_w(b) = (X^T W X)^{-1} X^T W y$

↑ HW problem

where $W = \operatorname{diag}(w_1, \dots, w_n) = \begin{pmatrix} w_1 & & 0 \\ & w_2 & \\ 0 & & \ddots \\ & & & w_n \end{pmatrix}$

$w_i \propto 1/\sigma_i^2 \Rightarrow$ Obs. with higher σ_i^2 get lower weight (de-emphasized)

i.e. y_i is a less precise measurement of $\beta^T x_i$

Claim: $y \sim N(X\beta, \sigma^2 W^{-1}) \Rightarrow \hat{\beta}^{WLS} \sim N(\beta, \sigma^2 (X^T W X)^{-1})$

i.e. $y_i \sim N(\beta^T x_i, \sigma^2/w_i)$ independently for $i = 1, \dots, n$

Proof: $E[\hat{\beta}^{WLS}] = E[(X^T W X)^{-1} X^T W y]$

$= (X^T W X)^{-1} X^T W \underbrace{E[y]}_{X\beta}$ (linearity)

$= (X^T W X)^{-1} (X^T W X) \beta = \beta$ so $\hat{\beta}^{WLS}$ is unbiased

For a matrix M, $Cov(My) = M\, Cov(y)\, M^T$ (matrix analogue for $Var(cZ) = c^2 Var(Z)$)

Let $M = (X^T W X)^{-1} X^T W$

$$\Rightarrow \text{Cov}(\hat{\beta}^{WLS}) = M \underbrace{\text{Cov}(y)}_{\sigma^2 W^{-1}} M^T$$

$$= (X^T W X)^{-1} X^T W \, \sigma^2 W^{-1} W X (X^T W X)^{-1}$$

$$= \sigma^2 (X^T W X)^{-1} X^T W X (X^T W X)^{-1} = \sigma^2 (X^T W X)^{-1}$$

□

$\rightsquigarrow$ Inference for β_j via $se(\hat{\beta}_j^{WLS}) = \hat{\sigma} \sqrt{(X^T W X)^{-1}_{jj}}$

where $\hat{\sigma}^2 = \frac{1}{n-p-1} \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2$ $\quad$ $\hat{\sigma}$ residual std error

↳ Intuition: $y_i - \hat{\beta}^T x_i \approx y_i - \beta^T x_i \sim N(0, \sigma^2/w_i)$

$\Rightarrow E[(y_i - \beta^T x_i)^2] = \sigma^2/w_i$

Main result: $\dfrac{\hat{\beta}_j^{WLS} - \beta_j}{se(\hat{\beta}_j^{WLS})} \sim$ Student-t with $df = n-p-1$

$\Rightarrow$ 95% CI: $\hat{\beta}_j^{WLS} \pm qt(0.975, n-p-1)\, se(\hat{\beta}_j^{WLS})$

$\Rightarrow$ Level α test of $H_0: \beta_j = c$

1) Compute $T_j = \dfrac{\hat{\beta}_j^{WLS} - c}{se(\hat{\beta}_j^{WLS})}$ $\quad$ 2) Reject if $|T_j| > qt(1-\frac{\alpha}{2}, n-p-1)$

e.g. Supervisor data, CI for β_1

```
> weighted.lmod <- lm(Y~X,supervisor.df,weights=1/X^2)
> confint(weighted.lmod,"X")
      2.5 %    97.5 %
X 0.1024573 0.1395233
```

Approximately 10 to 14 new supervisors are needed for an additional 100 workers

```
weighted.lmod <- lm(Y~X,supervisor.df,weights=1/X^2)
summary(weighted.lmod)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.803296   4.569745   0.832    0.413
X           0.120990   0.008999  13.445 6.04e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02266 on 25 degrees of freedom
Multiple R-squared:  0.8785,    Adjusted R-squared:  0.8737
F-statistic: 180.8 on 1 and 25 DF,  p-value: 6.044e-13
```

correspond to analogous (weighted) quantities

$$S^w_{yy} = \sum_{i=1}^{n} w_i (y_i - \bar{y}_w)^2 \quad \text{where} \quad \bar{y}_w = \frac{1}{\sum_j w_j} \sum_i w_i y_i$$

$$(= \bar{y} \text{ if } w_i \equiv 1)$$

$$SS^w_{reg} = \sum_{i=1}^{n} w_i (\hat{y}_i - \bar{y}_w)^2 = S^w_{yy} - RSS_w(\hat{\beta}^{WLS})$$

$$R^2 = SS^w_{reg} / S^w_{yy}, \quad F = \frac{SS^w_{reg}/p}{RSS_w/(n-p-1)}$$

e.g.

```
> wlmod2 <- lm(Y~X+I(X^2),supervisor.df,weights=1/X^2)
> anova(weighted.lmod,wlmod2)
Analysis of Variance Table

Model 1: Y ~ X
Model 2: Y ~ X + I(X^2)
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1     25 0.012842
2     24 0.011598  1 0.0012444 2.5751 0.1216
```

$H_0: E[y_i] = \beta_0 + \beta_1 x_i$

$H_1: E[y_i] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$

$\rightsquigarrow$ don't reject H_0

Hat matrix: $H = X(X^T W X)^{-1} X^T W \Rightarrow Hy = X\hat{\beta}^{WLS} = \hat{y}$

Still have $e = y - \hat{y}$, $e_i \sim N(0, \sigma_i^2 (1 - h_{ii}))$ $\quad \sigma_i^2 = \frac{\sigma^2}{w_i}$

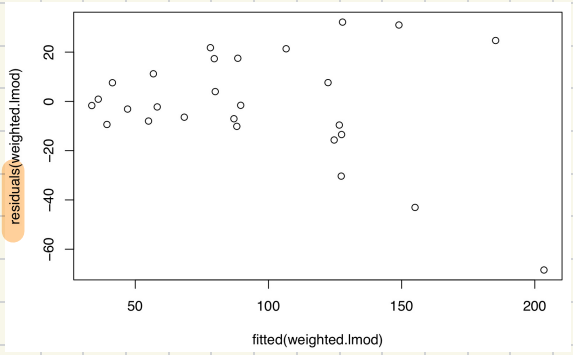
leverage = i^{th} diagonal entry of H

```
> weights <- 1/supervisor.df$X^2
> sum(residuals(weighted.lmod)*weights)
[1] -7.665648e-20
```

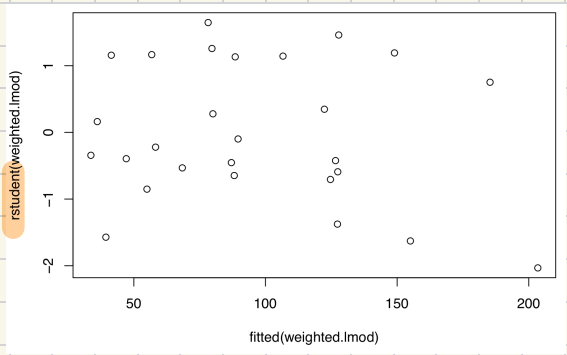
~ Need to standardize e_i for the residual plot to resemble constant variance

```
weighted.lmod <- lm(Y~X,supervisor.df,weights=1/X^2)
plot(fitted(weighted.lmod),residuals(weighted.lmod))
plot(fitted(weighted.lmod),rstudent(weighted.lmod))
```

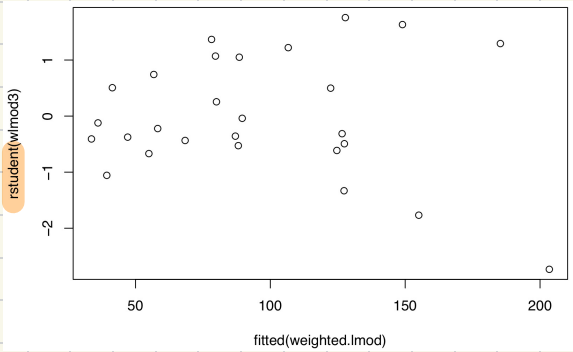
Raw residuals



Studentized residuals




```
wlmod3 <- lm(Y~X,supervisor.df,weights=1/X)
plot(fitted(weighted.lmod),rstudent(wlmod3))
```



Not as effective as $w \propto 1/x^2$ for removing non-const. variance