Module 3: The Linear Regression Model

Harvard University
Spring 2018
Jeff Gill

Linear Models

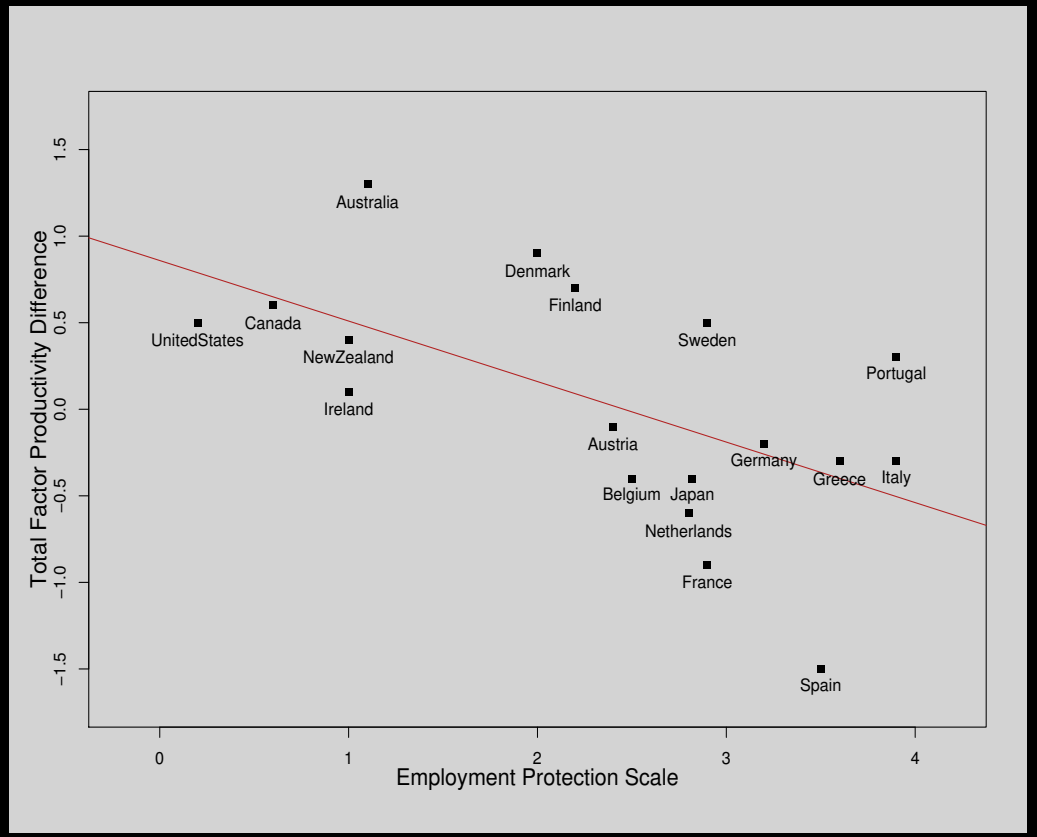
- A very common tool of social scientists is the so-called *linear regression model*, also sometimes called *OLS regression*.
- This is a method of looking at data and figuring out an underlying trend in the form of a straight line.
- We will not worry about any of the calculation details here, but we can think about the implications.
- What does this particular line mean? It means that for a specified change in the x-variable on the horizontal axis, we get a $\beta \times x$ change in the y-variable on the vertical axis.
- So this allows us to make claims about the trend in a linear fashion.

OECD Example

- The data are from the Organization for Economic Cooperation and Development (OECD) and highlight the relationship between
  - commitment to employment protection measured on an interval scale (0-4) indicating the quantity and extent of national legislation to protect jobs,
  - the total factor productivity difference in growth rates between 1980-1990 and 1990-1998 for 19 countries.
- For details, see The Economist, September 23, 2000.

OECD Example

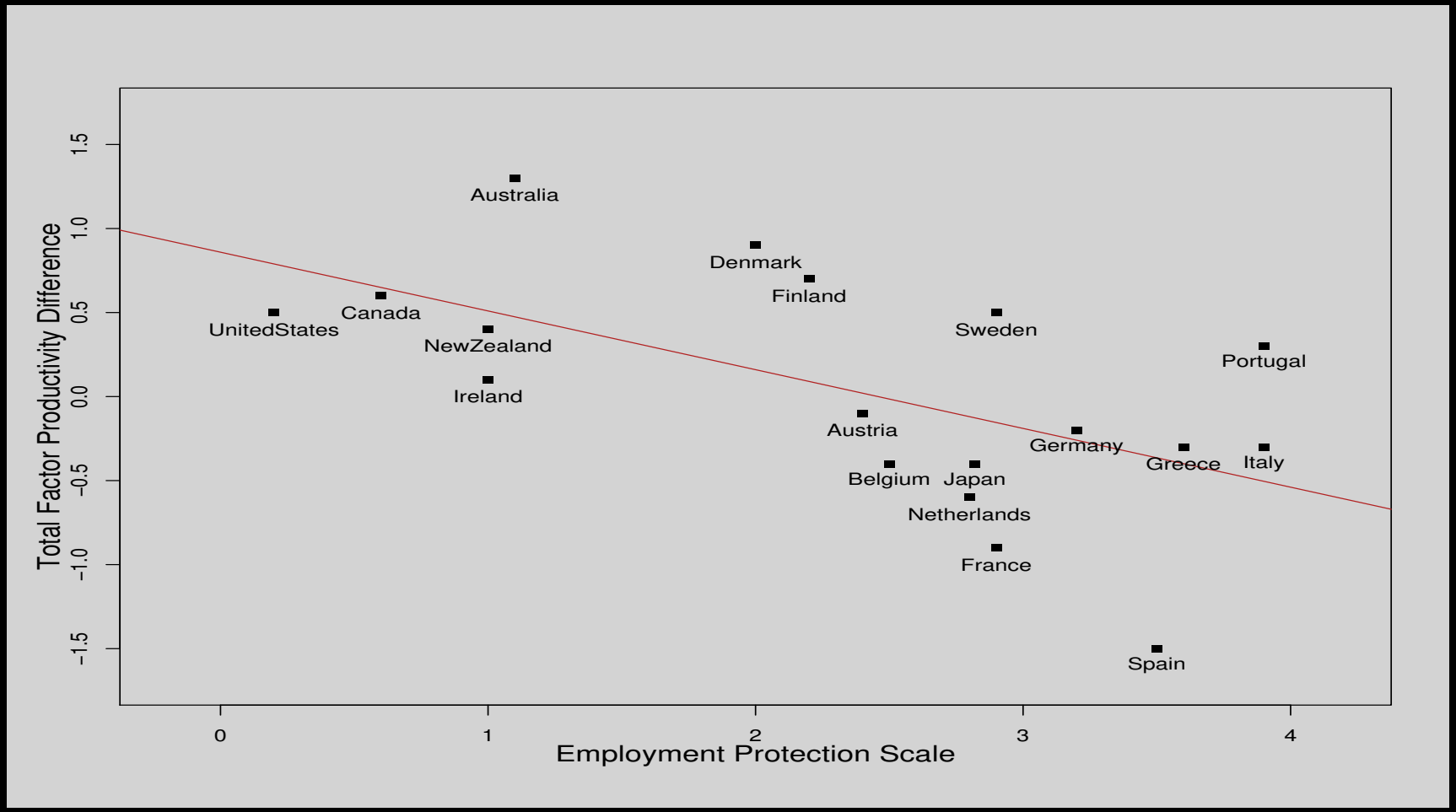
	Prot.	Prod.
United States	0.2	0.5
Canada	0.6	0.6
Australia	1.1	1.3
New Zealand	1.0	0.4
Ireland	1.0	0.1
Denmark	2.0	0.9
Finland	2.2	0.7
Austria	2.4	-0.1
Belgium	2.5	-0.4
Japan	2.6	-0.4
Sweden	2.9	0.5
Netherlands	2.8	-0.5
France	2.9	-0.9
Germany	3.2	-0.2
Greece	3.6	-0.3
Portugal	3.9	0.3
Italy	3.8	-0.3
Spain	3.5	-1.5



OECD Example

```
oecd <- read.table("http://people.hmdc.harvard.edu/~jgill/data/oecd.data",
        header=TRUE,row.names=1)

plot(oecd$Prot,oecd$Prod,xlim=c(-0.2,4.2),ylim=c(-1.7,1.7),pch=15,xlab="",ylab="")
x.y.fit <- lsfit(oecd$Prot,oecd$Prod)
abline(x.y.fit$coefficients,col="firebrick")
text(oecd$Prot,(oecd$Prod-0.1),dimnames(oecd)[[1]])
mtext(side=1,cex=1.3,line=2,"Employment Protection Scale")
mtext(side=2,cex=1.3,line=2,"Total Factor Productivity Difference")
```

OECD Example



OECD Example

```
oecd.fit <- lm(oecd$Prod~oecd$Prot)
summary(oecd.fit)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8591     0.3174   2.706   0.0156
oecd$Prot    -0.3496     0.1215  -2.878   0.0109
---
Residual standard error: 0.5761 on 16 degrees of freedom
Multiple R-Squared: 0.3411,     Adjusted R-squared: 0.2999
F-statistic: 8.284 on 1 and 16 degrees of freedom,      p-value: 0.01093
```

What Happens When it Doesn't Work?

- The New York Times Magazine, August 7, 2011, page 13.
- 24 countries: average survey review of restaurant service quality and a tipping index from three travel etiquette web sites.
- The data

Country	Quality	Tip	Country	Quality	Tip
Japan	4.4	0.00	Thailand	3.9	0.03
Canada	3.7	0.16	New_Zealand	3.7	0.07
UAE	3.6	0.10	Germany	3.6	0.08
USA	3.6	0.18	South_Africa	3.5	0.11
Australia	3.4	0.08	Argentina	3.4	0.10
Morocco	3.4	0.07	Turkey	3.4	0.08
India	3.3	0.10	Brazil	3.3	0.07
Vietnam	3.2	0.05	England	3.2	0.10
Greece	3.2	0.08	Spain	3.1	0.08
France	3.1	0.08	Italy	3.0	0.07
Egypt	3.0	0.08	Mexico	3.0	0.13
China	2.9	0.03	Russia	1.7	0.10

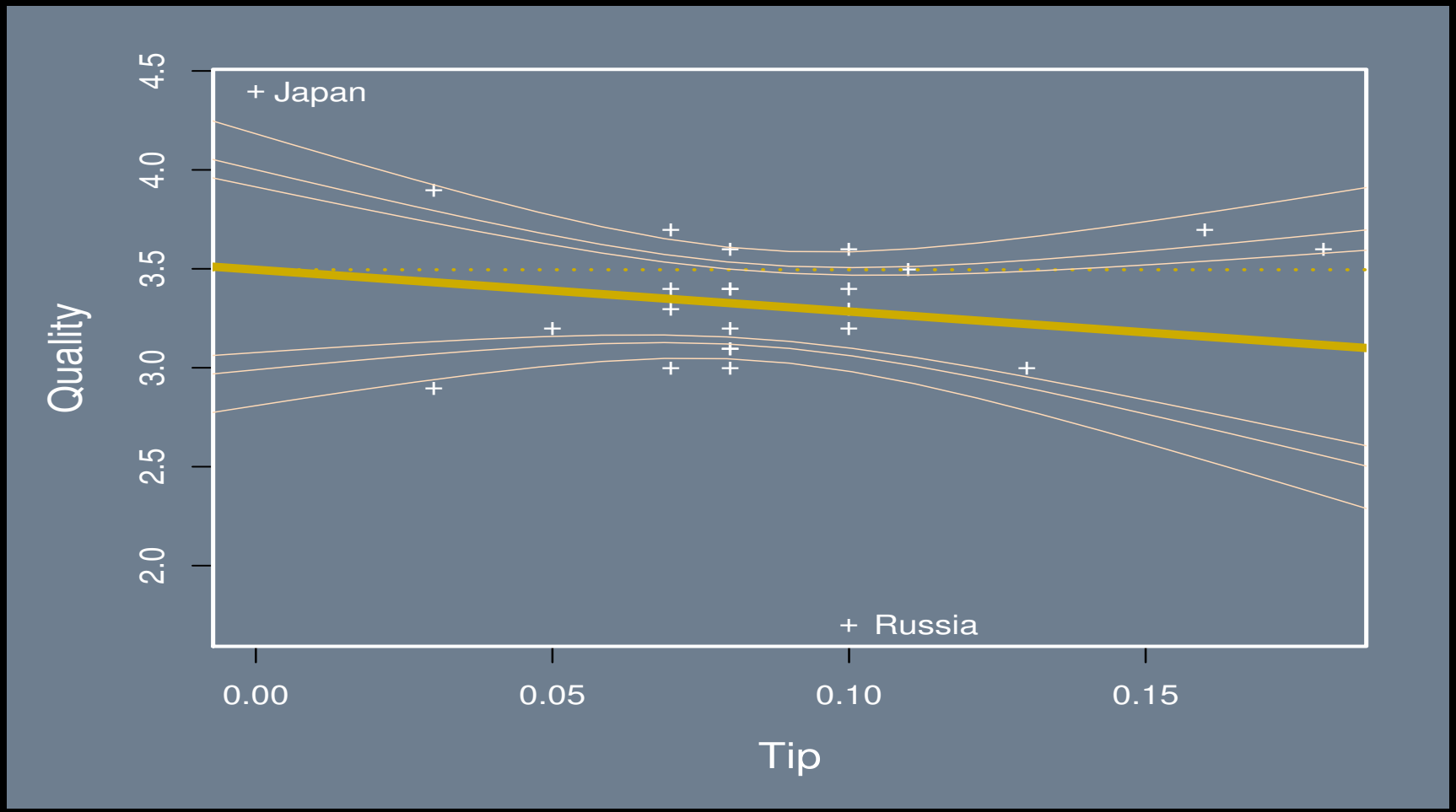
What Happens When it Doesn't Work?

```
service <- read.table("http://jeffgill.org/data/service.dat",
                      header=TRUE,row.names=1)
service.lm <- lm(Quality ~ Tip, data=service)
source("./Class.MLE/graph.summary.R")
graph.summary(service.lm)

Family: gaussian
Link function: identity

             Coef Std.Err. 0.95 Lower 0.95 Upper   CIs:  ZE+RO
(Intercept)  3.495    0.244      3.018      3.973              |o|
Tip         -2.113    2.632     -7.272      3.046  |-----o-----|

N: 24    Estimate of Sigma: 0.485
```

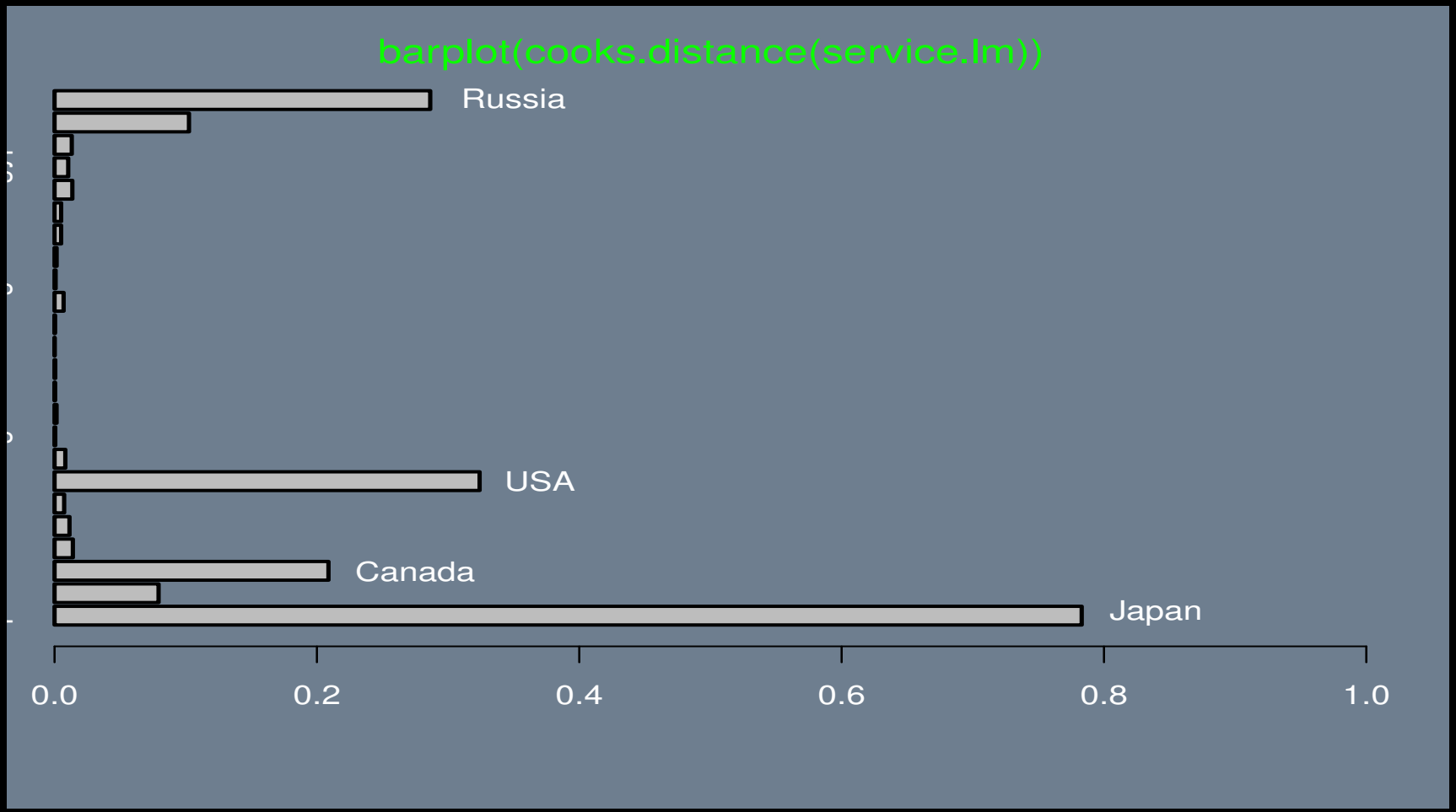
What Happens When it Doesn't Work?

```
par(mfrow=c(1,1),mar=c(5,5,2,2),lwd=2,col.axis="white",col.lab="white",
    col.sub="white",col="white",bg="slategray", cex.lab=1.3)
# PLOT POINTS AND REGRESSION LINES
plot(service$Tip, service$Quality, pch="+",xlab="Tip",ylab="Quality")
abline(service.lm,col="gold3",lwd=5)
abline(h=service.lm$coef[1],col="gold3",lty=3,lwd=2)
# ADD CONFIDENCE BOUNDS AT THREE LEVELS
ruler.df <- data.frame(Tip = seq(-0.1, 2,length=200))
for (k in c(0.99,0.95,0.90))  {
    confidence.interval <- predict(service.lm, ruler.df, interval="confidence",
        level=k)
    lines(ruler.df[,1],confidence.interval[,2],col="peachpuff",lwd=0.75)
    lines(ruler.df[,1],confidence.interval[,3],col="peachpuff",lwd=0.75)
}
# IDENTIFY POTENTIAL OUTLIERS
text(0.113,1.7,"Russia")
text(0.011,4.38,"Japan")
```

What Happens When it Doesn't Work?



How Influential is Japan?



Correlation and Regression

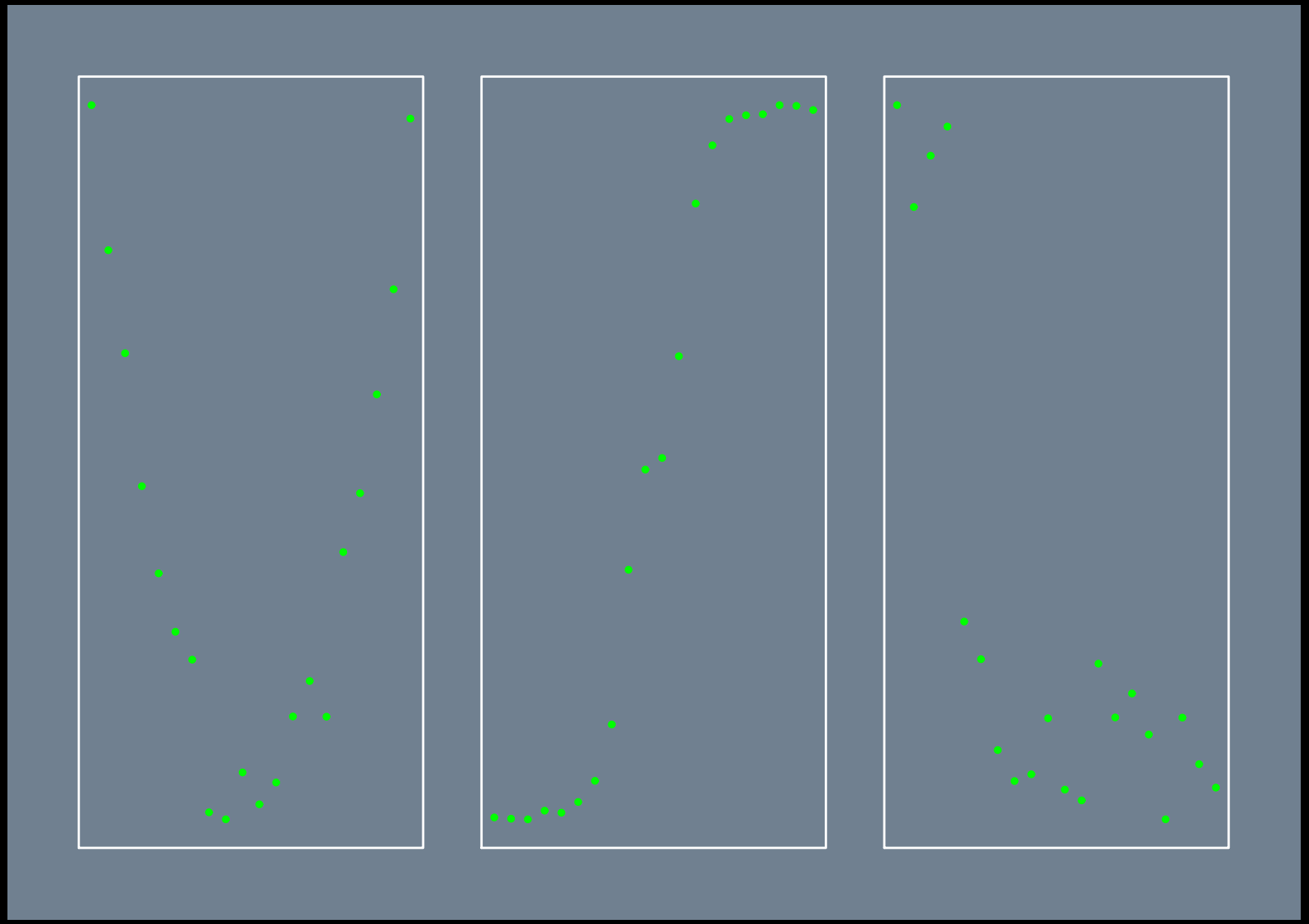
- Also, regression *is* correlation, since:

$$cor(X, Y) = \frac{s_X}{s_Y}\beta$$

- From our anaemia example picking `Age`:

```
coef(a.lm.out)[2]
0.1342515
apply(anaemia,2,sd)
        Hb        PCV        Age  Menapause
 2.4018852  7.8958683 15.7366485  0.5129892

cor(anaemia[,1],anaemia[,3])
[1] 0.8795875

(15.7366485/2.4018852) * 0.1342515
[1] 0.8795877
```

When Not To Use Correlation



Correlation: Tests of Significance

- Hypotheses: $H_0{:}\rho = 0$ versus $H_1{:}\rho \neq 0$.
- Test statistic:

$$t = r/SE(r), \qquad SE(r) = \sqrt{\frac{1-r^2}{n-2}}.$$

- HB and PCV from the anaemia data:

$$r = 0.6733745 \qquad SE(r) = \sqrt{\frac{1-0.6733745^2}{20-2}} = 0.1742551 \qquad t = 3.864304 \qquad p \approx 0.001.$$

- HB and Age from the anaemia data:

$$r = 0.8795875 \qquad SE(r) = \sqrt{\frac{1-0.6733745^2}{20-2}} = 0.1121323 \qquad t = 7.844191 \qquad p \approx 0.0001.$$

Gauss-Markov Assumptions for Classical Linear Regression

- Functional Form: $\underset{(n\times 1)}{\mathbf{Y}} = \underset{(n\times k)(k\times 1)}{\mathbf{X}\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\epsilon}}$ (recall $\mathbf{X}$ has a leading column of 1's)
- Mean Zero Errors: $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$
- Homoscedasticity: $\text{Var}[\boldsymbol{\epsilon}] = \sigma^2\mathbf{I}$
- Non-Correlated Errors: $\text{Cov}[\epsilon_i, \epsilon_j] = 0, \quad \forall i \neq j$
- Exogeneity of Explanatory Variables: $\text{Cov}[\epsilon_i, \mathbf{X}] = 0, \qquad \forall i$

  - Note that every one of these lines has $\boldsymbol{\epsilon}$ in it, meaning that these are assumptions about the underlying population values.

Other Considerations

- Requirements:
  - conformability of matrix/vector objects
  - $\mathbf{X}$ has full rank k, so $\mathbf{X}'\mathbf{X}$ is invertible (non-zero determinant, nonsingular)
  - identification condition: not all points lie on a vertical line.
- Freebee: eventual normality. . . $\boldsymbol{\epsilon}|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$.
- Toughness: the linear model is both *robust* to minor violations of the Gauss-Markov assumptions and *resistant* to outlying values.

Estimation With OLS, Scalar:

- Our goal is to minimize squared errors around the line determined by the unknown coefficients:

$$S(\boldsymbol{\beta}) = \min_{\hat{\beta}_0, \hat{\beta}_1} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

- Taking the derivatives with respect to each parameter separately, and set equal to zero:

$$\frac{dS(\boldsymbol{\beta})}{d\hat{\beta}_0} \equiv \sum_{i=1}^{n} (-2)(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \qquad \frac{dS(\boldsymbol{\beta})}{d\hat{\beta}_1} \equiv \sum_{i=1}^{n} (-2x_i)(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)$$

- Working on the intercept algebraically:

$$n\bar{y} - n\hat{\beta}_0 - \hat{\beta}_1 n\bar{x} = 0 \longrightarrow \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \hat{x}$$

  which depends on $\hat{\beta}$.

- Now get the slope estimate...

Estimation With OLS, Scalar:

$$\sum_{i=1}^{n}\left[x_i y_i - \hat{\beta}_0 x_i - \hat{\beta}_1 x_i^2\right] = 0$$

$$\sum_{i=1}^{n}\left[x_i y_i - (\bar{y} - \hat{\beta}_1 \bar{x}) x_i - \hat{\beta}_1 x_i^2\right] = 0$$

$$\sum_{i=1}^{n} x_i y_i - n\bar{y}\bar{x} + n\hat{\beta}_1 \bar{x}^2 - \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = 0$$

$$\sum_{i=1}^{n} x_i y_i - n\bar{y}\bar{x} + \hat{\beta}_1 \left(n\bar{x}^2 - \sum_{i=1}^{n} x_i^2\right) = 0$$

$$\hat{\beta}_1 \left(n\bar{x}^2 - \sum_{i=1}^{n} x_i^2\right) = n\bar{y}\bar{x} - \sum_{i=1}^{n} x_i y_i$$

$$\hat{\beta}_1 = \frac{n\bar{y}\bar{x} - \sum_{i=1}^{n} x_i y_i}{n\bar{x}^2 - \sum_{i=1}^{n} x_i^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Estimation With OLS, Matrix:

- Define the following function:

$$\begin{aligned} S(\boldsymbol{\beta}) &= \boldsymbol{\epsilon}'\boldsymbol{\epsilon} \\ &= (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \underset{(1\times n)(n\times 1)}{\mathbf{Y}'\mathbf{Y}} - \underset{(1\times n)(n\times k)(k\times 1)}{2\mathbf{Y}'\mathbf{X}\boldsymbol{\beta}} + \underset{(1\times k)(k\times n)(n\times k)(k\times 1)}{\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}} \end{aligned}$$

- Take the derivative of $S(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$:

$$\frac{\partial}{\partial \boldsymbol{\beta}} S(\boldsymbol{\beta}) = 0 - 2\underset{(k\times n)(n\times 1)}{\mathbf{X}'\mathbf{Y}} + \underset{(k\times n)(n\times k)(k\times 1)}{2\mathbf{X}'\mathbf{X}\boldsymbol{\beta}} \equiv 0,$$

  (think about what sign you would get by taking another derivative).

- So there exists a (minimizing) solution at some value $\hat{\boldsymbol{\beta}}$ (or notationally $\hat{\beta}$) of $\boldsymbol{\beta}$: $\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y}$ which is the Normal Equation.

- Premultiplying the Normal Equation by $(\mathbf{X}'\mathbf{X})^{-1}$, gives: $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, where we can call $\hat{\boldsymbol{\beta}}$ as $\hat{\beta}$ for notational convenience (this is where the requirement for $\mathbf{X}'\mathbf{X}$ to be nonsingular comes in).

OLS Estimator Notes

- Another way to express the OLS estimator is to say that we want the $\boldsymbol{\beta}$ that minimizes the *squared prediction error*:

$$S(\boldsymbol{\beta}) = \mathbb{E}[(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^2],$$

  which is a restatement of $S(\boldsymbol{\beta}) = \boldsymbol{\epsilon}'\boldsymbol{\epsilon}$.

- Sometimes the solution is called the *linear projection coefficient*:

$$\hat{\boldsymbol{\beta}} = \underset{\mathbf{b}\in\Re}{\text{argmin}} S(\mathbf{b}).$$

- And yet another expression for this quantity is:

$$\hat{\boldsymbol{\beta}} = \mathbb{E}(\mathbf{XY})\left(\mathbb{E}(\mathbf{XX}')\right)^{-1}$$

  meaning that:

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbb{E}(\mathbf{XY})\left(\mathbb{E}(\mathbf{XX}')\right)^{-1}$$

  from:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \;\longrightarrow\; \hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

Estimating From Sample Quantities

- From $\mathbb{E}[\mathbf{e}'\mathbf{e}|\mathbf{X}] = (n-k)\sigma^2$, we algebraically get an unbiased estimator:

$$\hat{\sigma}^2 = \frac{\mathbf{e}'\mathbf{e}}{n-k} = s^2,$$

  so that a *finite sample* estimator of $\text{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ is:

$$\widehat{\text{Var}}[\hat{\boldsymbol{\beta}}] = s^2(\mathbf{X}'\mathbf{X})^{-1}.$$

- The Wald-style traditional linear inference, for the kth coefficient is:

$$z_k = \frac{\hat{\boldsymbol{\beta}}_k - \beta_k^{\text{null}}}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})_k^{-1}}} \sim N(0,1),$$

  with the assumption that we know σ^2 (which we usually do not).

Estimating From Sample Quantities

- Making the substitution gives:

$$t_{(n-k)} = \frac{\hat{\boldsymbol{\beta}}_k - \beta_k^{\text{null}}}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})^{-1}}} \times \frac{1}{\sqrt{\frac{(n-k)s^2}{\sigma^2}/(n-k)}} = \frac{\hat{\boldsymbol{\beta}}_k - \beta_k^{\text{null}}}{\sqrt{s^2(\mathbf{X}'\mathbf{X})^{-1}}}$$

- Typical (Wald) regression test:

$$H_0\text{: } \beta_k = 0 \qquad\qquad H_1\text{: } \beta_k \neq 0$$

making:

$$t_{(n-k)} = \frac{\hat{\boldsymbol{\beta}}_k - \beta_k^{\text{null}}}{\sqrt{s^2(\mathbf{X}'\mathbf{X})^{-1}}} = \frac{\hat{\boldsymbol{\beta}}_k}{SE(\beta_k)}$$

- Alternatives usually look like:

$$H_0\text{: } \beta_k < 7 \qquad\qquad H_1\text{: } \beta_k \geq 7$$

making:

$$t_{(n-k)} = \frac{\hat{\boldsymbol{\beta}}_k - 7}{SE(\beta_k)}$$

Summary Statistics

- $(1-\alpha)$ Confidence Interval for $\hat{\boldsymbol{\beta}}_k$:

$$\left[\hat{\boldsymbol{\beta}}_k - SE(\hat{\boldsymbol{\beta}}_k)t_{\alpha/2,df} : \hat{\boldsymbol{\beta}}_k + SE(\hat{\boldsymbol{\beta}}_k)t_{\alpha/2,df}\right]$$

- $(1-\alpha)$ Confidence Interval for σ^2:

$$\left[\frac{(n-k)s^2}{\chi^2_{1-\alpha/2}} : \frac{(n-k)s^2}{\chi^2_{\alpha/2}}\right]$$

- The F-statistic test for all but β_0 (the intercept) equal to zero:

$$F = \frac{SSR/(k-1)}{SSE/(n-k)} \sim F_{k-1,n-k} \text{ under the null.}$$

Properties of the Estimator, General

- The OLS estimator is consistent (converges in probability):

$$\underset{n\to\infty}{\text{plim}}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta},$$

  with Gauss-Markov assumption #5: $\text{Cov}[\epsilon_i, \mathbf{X}] = 0, \forall i$, and there is no perfect multicollinearity.

- The OLS estimate is optimal:

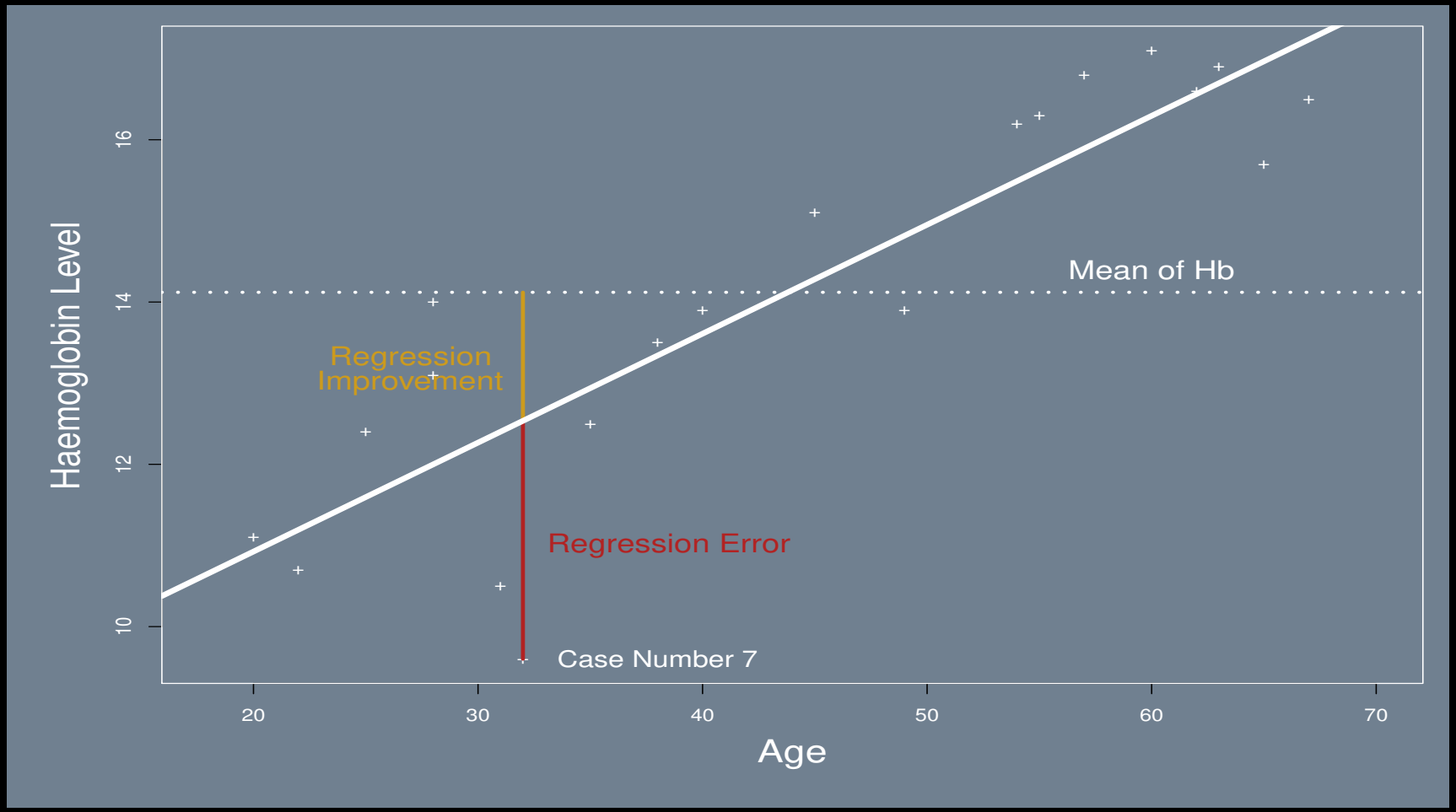
$$\text{Var}(\hat{\boldsymbol{\beta}}_{\text{OLS}}) \leq \text{Var}(\hat{\boldsymbol{\beta}}_{\text{All Other}})$$

  with Gauss-Markov assumptions #3 $\text{Var}[\boldsymbol{\epsilon}] = \sigma^2\mathbf{I}$, and #4 $\text{Cov}[\epsilon_i, \epsilon_j] = 0, \quad \forall i \neq j$.

- Given all of the Gauss-Markov assumptions and $\sigma^2 < \infty$, we say that $\hat{\boldsymbol{\beta}}$ is BLUE for $\boldsymbol{\beta}$ if calculated from OLS or MLE.

- Given sufficient sample size $\hat{\boldsymbol{\beta}}|\mathbf{X} \sim N(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$.

Fit & Decomposition, Illustration



Fit & Decomposition, Variability Definitions

- *Sum of Squares Total*, all the variability to obtain over the mean estimate,

$$\text{SST} = \sum_{i=1}^{n} (\mathbf{Y}_i - \bar{\mathbf{Y}})^2$$

- *Sum of Squares Regression*, the variability accounted for by the regression,

$$\text{SSR} = \sum_{i=1}^{n} (\hat{\mathbf{Y}}_i - \bar{\mathbf{Y}})^2$$

- *Sum of Squares Error*, the remaining variability not accounted for by the regression,

$$\text{SSE} = \sum_{i=1}^{n} (\hat{\mathbf{Y}}_i - \mathbf{Y}_i)^2$$

Total Magic!

- Adding Total Sum of Squares Regression to Total Sum of Squares Error:

$$\begin{aligned} SSR + SSE &= (\hat{\boldsymbol{\beta}}'\mathbf{X}'\hat{\mathbf{Y}} - n\mathbf{Y}'\mathbf{J}\mathbf{Y}) + (\mathbf{Y}'\mathbf{Y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\hat{\mathbf{Y}}) \\ &= \mathbf{Y}'\mathbf{Y} - n\mathbf{Y}'\mathbf{J}\mathbf{Y} \\ &= \text{SST} \end{aligned}$$

- Because in general sums of squares do not equal squares of sums, for example:

```
7^2 + 3^2
[1] 58
(7+3)^2
[1] 100
```

(except in unusual or pathological circumstances).

A Measure of Fit

- The "R-Square" or "R-Squared" measure:

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} = 1 - \frac{\mathbf{e}'\mathbf{e}}{\mathbf{Y}'\mathbf{M}^o\mathbf{Y}} = \frac{\hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{M}^o\mathbf{X}\hat{\boldsymbol{\beta}}}{\mathbf{Y}'\mathbf{M}^o\mathbf{Y}}$$

  where $M^o = \mathbf{I} - \frac{1}{n}\mathbf{i}\mathbf{i}'$, $\quad \mathbf{i} = c(1, 1, \ldots, 1)$.

- Note: $\mathbf{M}^o$ is idempotent and transforms means to deviances for the explanatory variables:

```
M.0 <- diag(3) - (1/3)*c(1,1,1)%*%t(c(1,1,1))
M.0
          [,1]     [,2]     [,3]
[1,]  0.66667 -0.33333 -0.33333
[2,] -0.33333  0.66667 -0.33333
[3,] -0.33333 -0.33333  0.66667
```

A Measure of Fit

- Also, there is another version that accounts for sample size and the number of explanatory variables (k):

$$R^2_{adj} = 1 - \frac{\mathbf{e}'\mathbf{e}/(n-k)}{\mathbf{Y}'M^o\mathbf{Y}/(n-1)}$$

  which is useful with small datasets.

- Bivariate relationships for `lm(Y ~ X)`:

$$\frac{s_Y}{s_X}\text{cor}X,Y = \beta$$

$$R^2 = \text{cor}(X,Y)^2$$

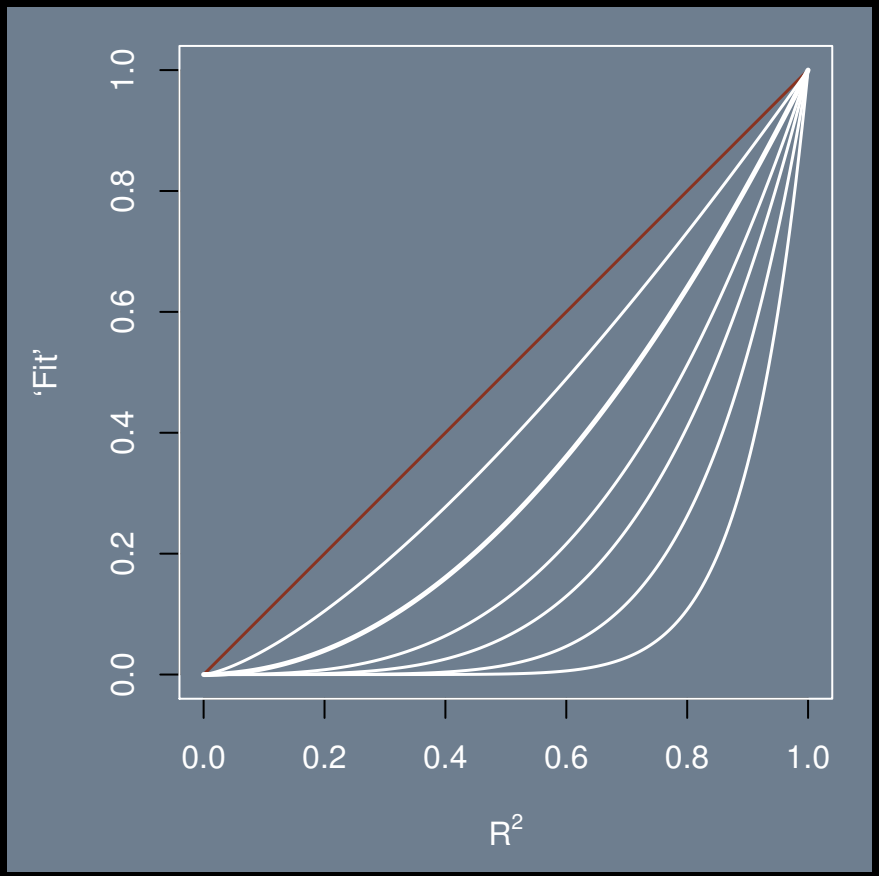
$$\frac{s_Y}{s_X}\sqrt{R^2} = \beta$$

$$\sqrt{R^2} = \frac{s_X}{s_Y}\beta$$

$$R^2 = \left(\frac{s_X}{s_Y}\beta\right)^2$$

Warnings about R^2

- There is not a *population* analog.
- It can never be reduced by adding more explanatory variables.
- It is a *quadratic* form in $[0:1]$ space.
- Therefore it does not have quite the meaning that people expect.



Linear Model Confidence Bands

- We want the predicted value of the outcome variable for $\mathbf{x}_i$ *in the sample*:

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i \qquad\qquad \hat{y}_i = \mathbf{x}_i\hat{\boldsymbol{\beta}}$$

- The variance at this point on the regression line, bivariate, is:

$$\text{Var}(\hat{b}|x_i) = s^2\left(\frac{1}{n} + \frac{(x_i - \bar{x})}{\sum(x_j - \bar{x})}\right).$$

- The variance at this point on the regression line, multivariate, is:

$$\begin{aligned}
\text{Var}[\mathbf{e}_i|\mathbf{X}, \mathbf{x}_i] &= \text{Var}[\mathbf{x}_i(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})] \\
&= \text{Var}[\mathbf{x}_i\boldsymbol{\beta}] + \text{Var}[\mathbf{x}_i\hat{\boldsymbol{\beta}}] \\
&= s^2 + \mathbf{x}_i\text{Var}[\hat{\boldsymbol{\beta}}]\mathbf{x}_i' \\
&= s^2 + \mathbf{x}_i(s^2(\mathbf{X}\mathbf{X})^{-1})\mathbf{x}_i' \\
&= s^2[1 + \mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i']
\end{aligned}$$

- The prediction interval (in the vertical direction) is created from

$$CI[\hat{y}] = \hat{y} \pm t_{\alpha/2}\sqrt{\text{Var}[e|\mathbf{X}, \mathbf{x}]}.$$

Linear Model Predictions/Forecasts

- The prediction interval (in the vertical direction) is created from

$$CI[\hat{y}^0] = \hat{y}^0 \pm t_{\alpha/2}\sqrt{\text{Var}[e^0|\mathbf{X}, \mathbf{x}^0]}$$

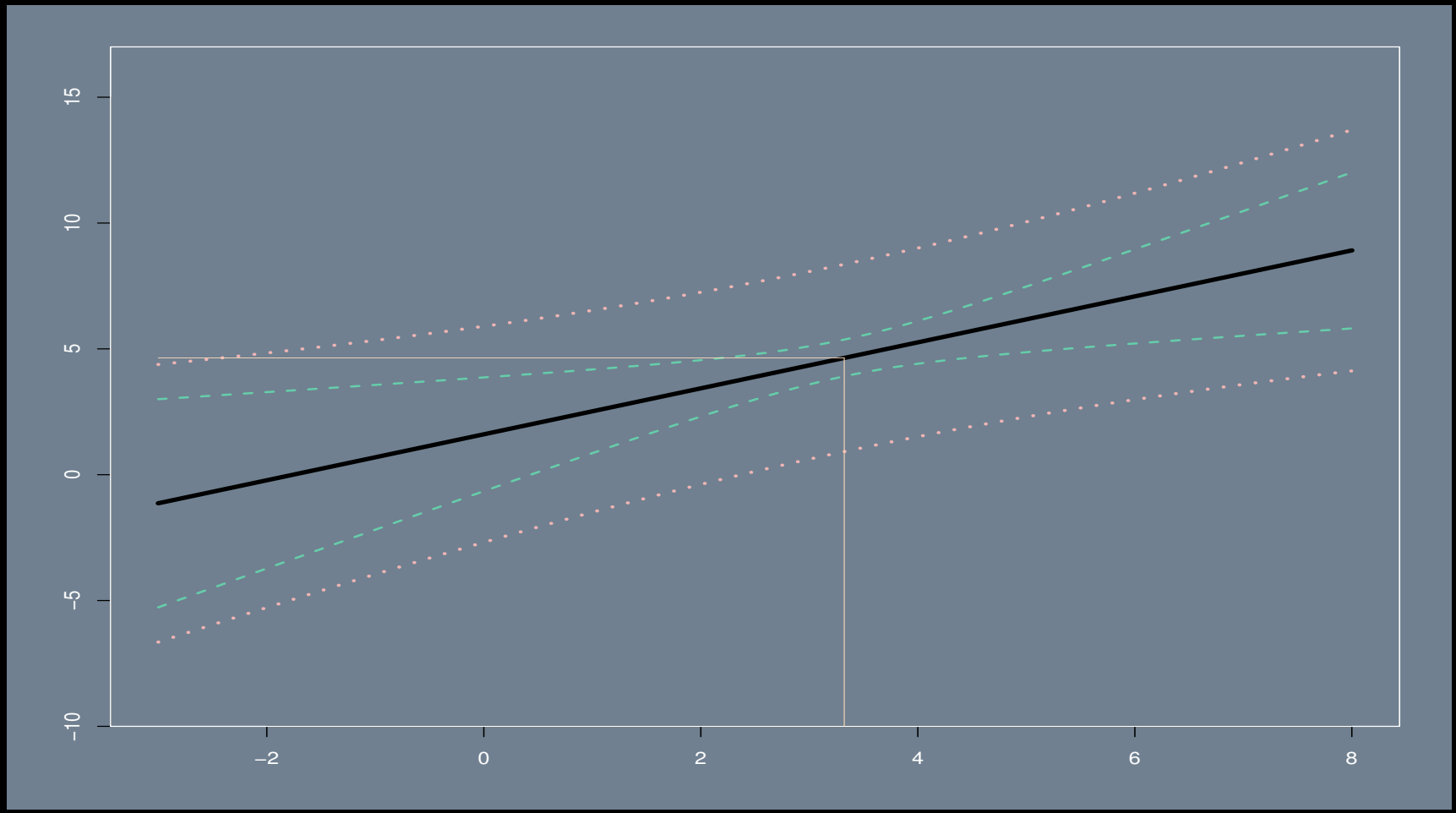
  which is higher than for some $\mathbf{x}$ value that was actually observed in the sample.

- Note that the value of $\mathbf{x}^0$ is buried in there, and like the CI for $\boldsymbol{\beta}$, it is smallest around $\bar{x}$.

- It is important to also distinguish between two interval estimates around the regression line: the CI for $\hat{y} = \mathbf{x}\boldsymbol{\beta}$ and the CI for $\hat{y}^0 = \mathbf{x}^0\boldsymbol{\beta}$.

- Where the prediction interval is always wider than the regression confidence interval.

Linear Model Predictions/Forecasts



Linear Model Predictions/Forecasts

- The R code for these intervals can be produced by:

```
X <- rnorm(25,3,1); Y <- X + rnorm(25,2,2)
ruler <- data.frame(X = seq(-3, 8,length=200))

confidence.interval <- predict(lm(Y ~ X), ruler, interval="confidence")
predict.interval <- predict(lm(Y ~ X), ruler, interval="prediction")

par(mar=c(1,1,1,1),oma=c(3,3,1,1),mfrow=c(1,1),col.axis="white",col.lab="white",
   col.sub="white",col="white",bg="slategray")
# REGRESSION LINE
plot(ruler[,1], confidence.interval[,1], type="l",lwd=4,ylim=c(-9,16),col="black")
# UPPER AND LOWER CONFIDENCE INTERVALS
lines(ruler[,1],confidence.interval[,2], lwd=2, lty=2, col="aquamarine3")
lines(ruler[,1],confidence.interval[,3], lwd=2, lty=2, col="aquamarine3")
# UPPER AND LOWER PREDICTION INTER ALS
lines(ruler[,1],predict.interval[,2], lwd=3, lty=3, col="rosybrown2")
lines(ruler[,1],predict.interval[,3], lwd=3, lty=3, col="rosybrown2")
segments(mean(X),-10,mean(X),mean(Y), lwd=0.5, col="peachpuff")
segments(-3,mean(Y),mean(X),mean(Y),  lwd=0.5, col="peachpuff")
```

Example: Poverty Among the Elderly, Europe

- Governments often worry about the economic condition of senior citizens for political and social reasons.
- Typically in a large industrialized society, a substantial portion of these people obtain the bulk of their income from government pensions.
- An important question is whether there is enough support through these payments to provide subsistence above the poverty rate.
- To see if this is a concern, the European Union (EU) looked at this question in 1998 for the (then) 15 member countries with two variables:
  1. the median (EU standardized) income of individuals age 65 and older as a percentage of the population age 0–64,
  2. the percentage of all age groups with income below 60% of the median (EU standardized) income of the national population.

Example: Poverty Among the Elderly, Europe

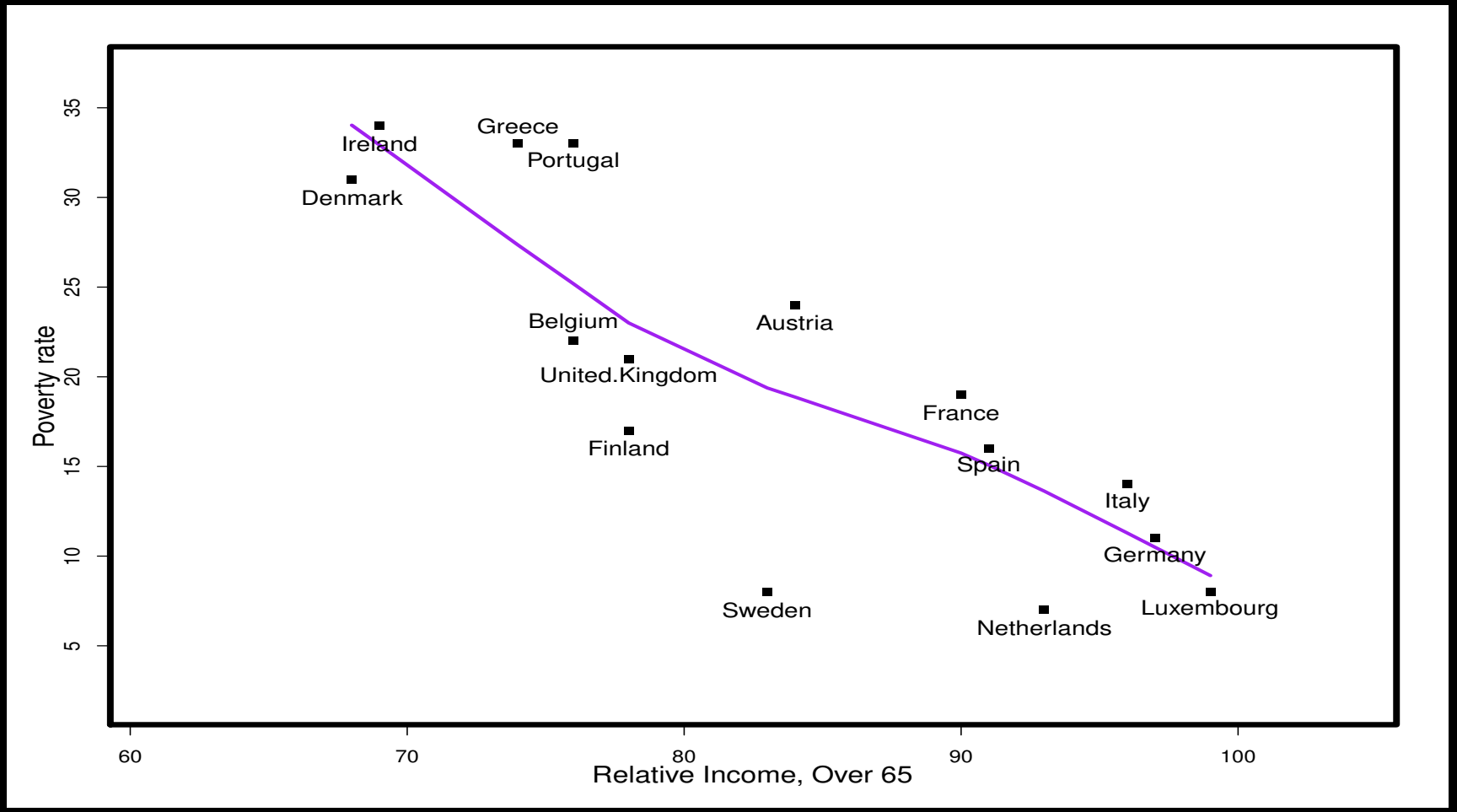
- The data from the European Household Community Panel Survey are:

Nation	Over 65 Relative Income	Total Poverty Rate
Netherlands	93.00	7.00
Luxembourg	99.00	8.00
Sweden	83.00	8.00
Germany	97.00	11.00
Italy	96.00	14.00
Spain	91.00	16.00
Finland	78.00	17.00
France	90.00	19.00
United.Kingdom	78.00	21.00
Belgium	76.00	22.00
Austria	84.00	24.00
Denmark	68.00	31.00
Portugal	76.00	33.00
Greece	74.00	33.00
Ireland	69.00	34.00

Example: Poverty Among the Elderly, Europe

```
eu.pov <- read.table("http://jeffgill.org/data/inc.pov.dat",row.names=1)
names(eu.pov) <- c("relative income", "poverty rate")
eu.pov <- eu.pov[-1,]  # FIRST LINE IS "EU.Average  89  17"

par(mar=c(4,4,2,2),lwd=5,bg="white")
plot(eu.pov,pch=15,xlab="",ylab="",ylim=c(2,37),xlim=c(61,104))
lines(lowess(eu.pov),col="purple",lwd=3)
text.loc <- cbind(eu.pov[,1],(eu.pov[,2]-1))
text.loc[14,2] <- text.loc[14,2] +2
text.loc[10,2] <- text.loc[10,2] +2
text(text.loc,dimnames(eu.pov)[[1]],cex=1.2)
mtext(side=1,cex=1.3,line=2,"Relative Income, Over 65")
mtext(side=2,cex=1.3,line=2,"Poverty rate")
```

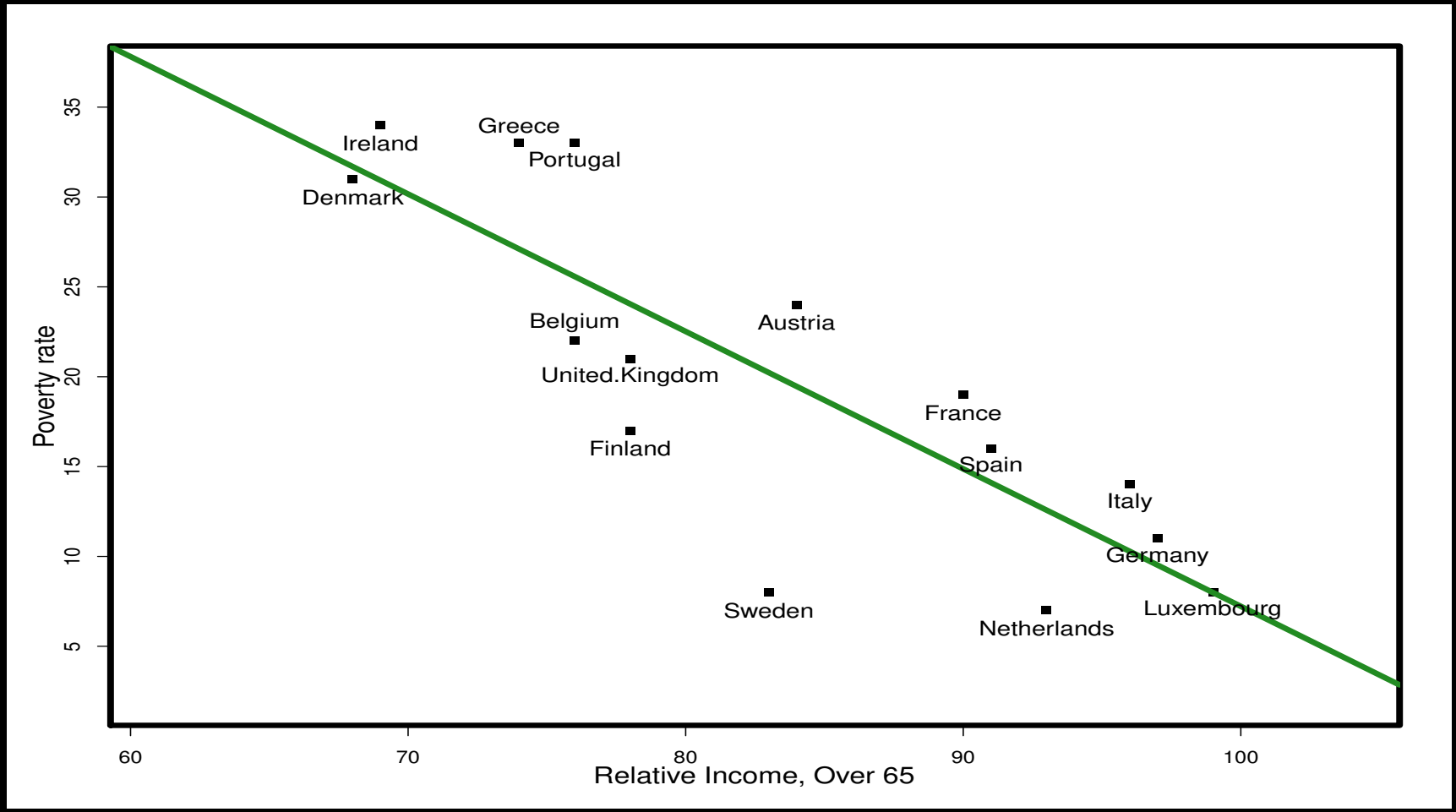
Example: Poverty Among the Elderly, Europe



Example: Poverty Among the Elderly, Europe

```
par(mar=c(4,4,2,2),lwd=5,bg="white")
plot(eu.pov,pch=15,xlab="",ylab="",ylim=c(2,37),xlim=c(61,104))
x.y.fit <- lm(eu.pov[,2] ~ eu.pov[,1])
abline(x.y.fit$coefficients,col="forest green")
text.loc <- cbind(eu.pov[,1],(eu.pov[,2]-1))
text.loc[14,2] <- text.loc[14,2] +2
text.loc[10,2] <- text.loc[10,2] +2
text(text.loc,dimnames(eu.pov)[[1]],cex=1.2)
mtext(side=1,cex=1.3,line=2,"Relative Income, Over 65")
mtext(side=2,cex=1.3,line=2,"Poverty rate")

coefficients(x.y.fit)
(Intercept) eu.pov[, 1]
   83.69279    -0.76469
```

Example: Poverty Among the Elderly, Europe



Example: Poverty Among the Elderly, Europe

```
summary(x.y.fit)
Call:
lm(formula = eu.pov[, 2] ~ eu.pov[, 1])

Residuals:
    Min      1Q  Median      3Q     Max
-12.224  -3.312   1.482   3.923   7.424

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.6928    12.2526   6.831 1.21e-05
eu.pov[, 1]  -0.7647     0.1458  -5.246 0.000158

Residual standard error: 5.611 on 13 degrees of freedom
Multiple R-Squared: 0.6792,     Adjusted R-squared: 0.6545
F-statistic: 27.52 on 1 and 13 DF,  p-value: 0.0001580
```

More Details on Linear Models

- In this example the slope of the line is -0.7647, which tells us how much the poverty rate changes for a one unit increase in income.
- The intercept is 83.6928, which is what poverty would be for zero income.
- The standard error is a measure of how reliable these estimates are. One common rule of thumb is to see if the standard error is half the coefficient estimates or less.
- R-Squared tells us how much of the variance of the outcome variable can be explained by the explanatory variable.
- F-statistic tells us how well "significant" the model fit is.

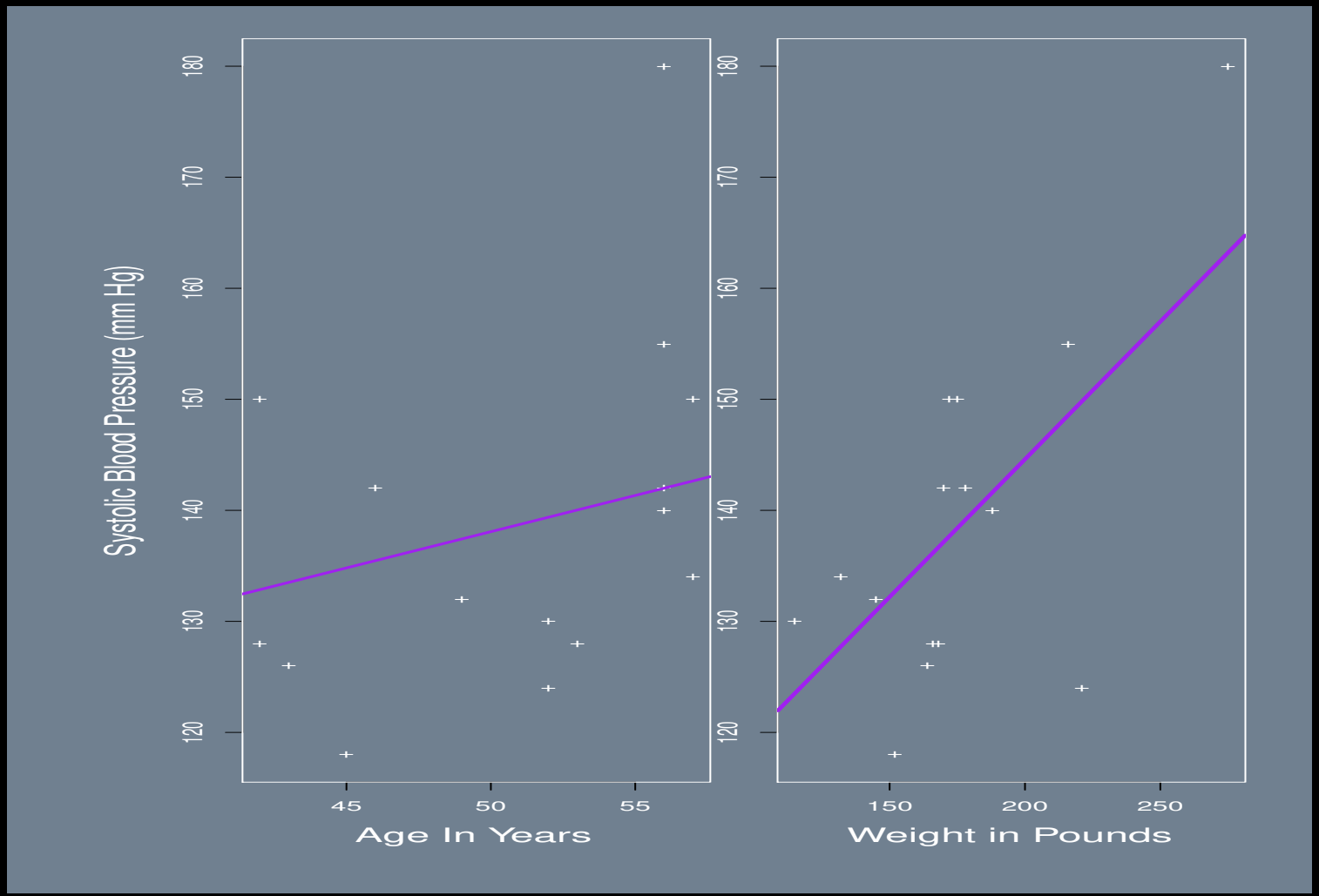
Example: Linear Model For Systolic Blood Pressure (mm Hg = millimeters of mercury)

```
sbp <- data.frame(
   "pressure" = c(132,155,130,142,150,128,126,118,180,124,150,134,140,142,128),
   "age.years" = c(49,56,52,46,57,42,43,45,56,52,42,57,56,56,53),
   "weight.lb" = c(145,216,115,170,172,166,164,152,275,221,175,132,188,178,168) )
sbp.out <- lm(pressure ~ age.years + weight.lb, data=sbp)
summary(sbp.out)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  61.8444    28.4664    2.17    0.051
age.years     0.6526     0.5632    1.16    0.269
weight.lb     0.2480     0.0839    2.96    0.012

Residual standard error: 11.7 on 12 degrees of freedom
Multiple R-squared: 0.52,        Adjusted R-squared: 0.44
F-statistic:  6.5 on 2 and 12 DF,  p-value: 0.0122
```

Example: Linear Model For Systolic Blood Pressure

```
attach(sbp)
par(mar=c(1,1,1,1),oma=c(6,6,1,1),mfrow=c(1,2),col.axis="white",col.lab="white",
   col.sub="white",col="white",bg="slategray")
plot(age.years,pressure,pch="+")
mtext(side=1,"Age In Years",line=3,cex=1.5);
mtext(side=2,"Systolic Blood Pressure (mm Hg)",line=3,cex=1.5)
abline(a=sbp.out$coef[1]+sbp.out$coef[3]*mean(sbp$weight.lb),b=sbp.out$coef[2],
   col="purple",lwd=3)
plot(weight.lb,pressure,pch="+")
mtext(side=1,"Weight in Pounds",line=3,cex=1.5)
abline(a=sbp.out$coef[1]+sbp.out$coef[2]*mean(sbp$age.years),b=sbp.out$coef[3],
   col="purple",lwd=3)
detach(sbp)
```

Example: Linear Model For Systolic Blood Pressure



Example: Linear Model For Systolic Blood Pressure

```
N <- 200
ruler.df <- data.frame(age.years = seq(min(sbp$age.years),max(sbp$age.years),length=N),
                weight.lb = seq(min(sbp$weight.lb),max(sbp$weight.lb),length=N) )
confidence.interval <- predict(sbp.out, ruler.df, interval="confidence")
predict.interval <- predict(sbp.out, ruler.df, interval="prediction")
par(mar=c(3,1,1,1),oma=c(4,4,1,1),mfrow=c(1,1),col.axis="white",col.lab="white",
   col.sub="white",col="white",bg="slategray")
plot(ruler.df[,1], confidence.interval[,1], type="l",lwd=4, col="black")
lines(ruler.df[,1],confidence.interval[,2], lwd=2, lty=2, col="aquamarine3")
lines(ruler.df[,1],confidence.interval[,3], lwd=2, lty=2, col="aquamarine3")
lines(ruler.df[,1],predict.interval[,2], lwd=3, lty=3, col="rosybrown2")
lines(ruler.df[,1],predict.interval[,3], lwd=3, lty=3, col="rosybrown2")
mtext(side=3,outer=FALSE,"Age In Years",line=1.03)
mtext(side=2,outer=FALSE,"Contribution to Systolic Blood Pressure",line=2.5)
```

Investment Data Example

```
# Data:
#   title 'Grunfeld's Investment Models Fit with Autoregressive Errors';
#   REF: Maddala, G.S. (1977), Econometrics, New York: McGraw-Hill, 290-281
#   label gei = 'Gross investment GE'
#         gec = 'Lagged Capital Stock GE'
#         gef = 'Lagged Value of GE shares';

ge <- read.table("http://people.hmdc.harvard.edu/~jgill/ge.stock.dat",header=T)
ge
```

	year	GEI	GEC	GEF
1	1935	33.1	1170.6	97.8
2	1936	45.0	2015.8	104.4
3	1937	77.2	2803.3	118.0
4	1938	44.6	2039.7	156.2
5	1939	48.1	2256.2	172.6
6	1940	74.4	2132.2	186.6
7	1941	113.0	1834.1	220.9
8	1942	91.9	1588.0	287.8

```
 9  1943  61.3 1749.4 319.9
10 1944  56.8 1687.2 321.3
11 1945  93.6 2007.7 319.6
12 1946 159.9 2208.3 346.0
13 1947 147.2 1656.7 456.4
14 1948 146.3 1604.4 543.4
15 1949  98.3 1431.8 618.3
16 1950  93.5 1610.5 647.4
17 1951 135.2 1819.4 671.3
18 1952 157.3 2079.7 726.1
19 1953 179.5 2371.6 800.3
20 1954 189.6 2759.9 888.9
```

```
ge.fit <- lm(GEF ~ year + GEI + GEC,data=ge)
summary(ge.fit)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-106.865  -26.781    6.366   21.399  103.660
```

```
        Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
        (Intercept) -8.447e+04  8.704e+03  -9.704 4.17e-08
        year         4.364e+01  4.490e+00   9.719 4.08e-08
        GEI         -3.565e-01  5.738e-01  -0.621    0.543
        GEC          2.088e-02  3.654e-02   0.571    0.576
        ---

        Residual standard error: 59.11 on 16 degrees of freedom
        Multiple R-Squared: 0.9532,     Adjusted R-squared: 0.9444
        F-statistic: 108.5 on 3 and 16 DF,  p-value: 7.581e-11

stem(residuals(ge.fit))
  -1 | 10
  -0 | 7
  -0 | 33321
   0 | 00112223
   0 | 559
   1 | 0

cooks.distance(ge.fit)
```

```
           1            2            3            4            5            6
6.864125e-01 4.787739e-02 2.564946e-02 4.201220e-03 3.965464e-03 8.205478e-03
           7            8            9           10           11           12
9.406674e-03 1.908396e-06 7.575185e-03 9.285874e-02 6.585631e-02 2.954182e-01
          13           14           15           16           17           18
2.294942e-02 3.698788e-03 2.308765e-02 4.218240e-03 3.663559e-05 3.097720e-03
          19           20
5.382212e-02 4.908950e-01

dfbetas(ge.fit)
    (Intercept)         year          GEI          GEC
1   0.966404810 -0.954752118  0.490539972 -1.337400884
2   0.165624623 -0.165265212 -0.023551318  0.083518661
3   0.091833014 -0.093256385  0.007371475  0.224908375
4  -0.002080003  0.002161565 -0.053468279  0.047111184
5   0.026735710 -0.026569876  0.069138730 -0.078830120
6  -0.040433750  0.040582206  0.016174031 -0.070862351
7  -0.158900316  0.158609815 -0.151982987  0.083132041
8   0.001364322 -0.001350728  0.001215412 -0.001833322
9   0.089979967 -0.090598603  0.119162499 -0.007963343
10  0.416384185 -0.418631027  0.492031068 -0.031276689
```

```
11  0.263147755 -0.263424657  0.286936780 -0.191345627
12 -0.798429113  0.800963315 -0.999416937  0.102596918
13 -0.157023091  0.156240616 -0.226093175  0.196291021
14  0.045601989 -0.045221171  0.077716863 -0.082522470
15 -0.175761410  0.177278100 -0.127135111 -0.121727104
16 -0.103646177  0.104056224 -0.087061799 -0.007906530
17 -0.006385140  0.006406534 -0.002353795 -0.001762486
18 -0.040764386  0.040628230  0.002337600  0.011387333
19 -0.091422153  0.089693634  0.081260781  0.152721750
20 -0.412141977  0.403826574  0.023546392  0.952675632
```

Multicollinearity Issues

- Sometimes we try to put two explanatory variables into the model that have the same information: height in feet, height in inches.
- More typically: 2 explanatory variables are highly but not perfectly correlated: ideology, tolerance, civic engagement, etc. measured in slightly different ways.
- This redundant information can cause serious estimation problems: in the worst case no parameter estimates are possible, and the model is now *not identified*.

- Symptoms for models that were able to be estimated: (1) software warnings, (2) small changes in data give large changes in parameter estimates, (3) coefficients have large standard errors and poor t-statistics even if F-statistics and R^2 are okay, (4) coefficients seem illogical (obviously wrong sign, huge magnitude).

Multicollinearity Remedies

- Respecify model (if reasonable): add/drop variables, add data cases that break the pattern, restrict the range of some variables, combine variables possibly with PCA.
- Center explanatory variables, or standardize.
- Create a new variable that is a weighted combination of highly correlated variables and use it to replace both (two variables to one variable in the model).
- Ridge regression (add a little bias):

$$\hat{\boldsymbol{\beta}} = [\mathbf{X}'\mathbf{X} + \mathbf{R}\mathbf{I}]^{-1}\mathbf{X}'\mathbf{Y}$$

  such that the [ ] part barely inverts, and can involve a penalty function.
- R packages that do this: `ridge`, `bigRR`, `genridge`, `parcor`, and more.
- See also: Jeff Gill and Gary King (SMR 2004), "What to do When Your Hessian is Not Invertible: Alternatives to Model Respecification in Nonlinear Estimation."

Simple Ridge Regression Example

```
anaemia <- read.table("http://jeffgill.org/data/anaemia.dat",
                      header=TRUE,row.names=1)
library(MASS)
a.lm3.out <- lm.ridge(Hb ~ Age + Menapause + I(Age+rnorm(nrow(anaemia))),
                      data=anaemia)
a.lm3.out$GCV
0.07936452

cbind(a.lm3.out$coef, sqrt(a.lm3.out$scales))
                                [,1]    [,2]
Age                          1.06565 3.91640
Menapause                    0.29350 0.70711
I(Age + rnorm(nrow(anaemia))) 0.73817 3.89488

summary(lm(Hb ~ Age + Menapause,data=anaemia))$coef[2:3,1:2]
           Estimate Std. Error
Age         0.11716   0.035881
Menapause   0.60002   1.100703
```

Dealing with Heteroscedasticity: The *TWEED* Dataset

- TWEED = Terrorism in Western Europe: Events Data (Jan Oskar Engene)
- Contains information on events related to internal (domestic) terrorism in 18 countries in Western Europe.
- The time period covered is 1950 to 2004.
- By focusing on internal terrorism, the TWEED data set only includes events initiated by agents originating in the West European countries.
- Terrorism data is characterized by observable and latent groupings/clusters.
- So there is likely to be extra heteroscedasticity in the linear model from the presence of these groups.

Dealing With Heteroscedascity From Group Effects

- What if there is heterogeneity in the standard errors from a group definition.
- This does not bias the coefficient estimates but will affect the estimated standard errors.
- Suppose there are M groups, with modified degrees of freedom for the model now equal to

$$df_{\text{robust}} = \frac{M}{(M-1)}\frac{(N-1)}{(N-K)}$$

- Cluster-Robust standard errors (White, Huber, etc.) adjust the variance-covariance matrix with a "sandwich estimation" approach:

$$VC^* = f_{\text{robust}} \underbrace{(\mathbf{X}'\mathbf{X})^{-1}}_{\text{bread}} \underbrace{(\mathbf{U}'\mathbf{U})}_{\text{meat}} \underbrace{(\mathbf{X}'\mathbf{X})^{-1}}_{\text{bread}}$$

  where:

  - $\mathbf{U}$ is an $M \times k$ matrix,
  - such that each row is produced by $\mathbf{X}_m * \mathbf{e}_m$ for group/cluster m, the element-wise product of the $N_m \times k$ matrix of observations in group m,
  - and the N_m-length $\mathbf{e}_m$ corresponding residuals vector.

Dealing With Heteroscedascity From Group Effects

- A non-clustered, non-robusted model:

```
tweed <- read.table("http://jeffgill.org/data/tweed2.dat",header=TRUE)
tweed.lm <- lm(I(killed+injured)~year+arrests+factor(attitude), data=tweed)
summary(tweed.lm)
                                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                                    252.070    213.799    1.18    0.239
year                                            -0.123      0.108   -1.15    0.253
arrests                                          5.359      2.188    2.45    0.015
factor(attitude)Ethnic/regionalist Separatist -6.373      7.137   -0.89    0.373
factor(attitude)Left wing extremist Other     -6.788      3.631   -1.87    0.063
factor(attitude)Right wing extremist Other    -4.710      3.632   -1.30    0.196

Residual standard error: 12.4 on 283 degrees of freedom
Multiple R-squared: 0.0374, Adjusted R-squared: 0.0204
F-statistic:  2.2 on 5 and 283 DF,  p-value: 0.0546
```

- The reference group for the factor is `Ethnic/regionalist Irredentist`.

Dealing With Heteroscedascity From Group Effects

- A model with robust standard errors:

```
lapply(c("sandwich","lmtest","plm"),library, character.only=TRUE)
source("Class.Multilevel/clx.R")    # http://jeffgill.org/slides/clx.R
M <- length(table(tweed$attitude))
new.df <- tweed.lm$df / (tweed.lm$df - (M -1))
clx(tweed.lm, new.df, tweed$attitude)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	252.0696	112.4921	2.24	0.026
year	-0.1233	0.0566	-2.18	0.030
arrests	5.3585	6.6622	0.80	0.422
factor(attitude)Ethnic/regionalist Separatist	-6.3725	0.3234	-19.70	< 2e-16
factor(attitude)Left wing extremist Other	-6.7882	0.7866	-8.63	4.5e-16
factor(attitude)Right wing extremist Other	-4.7096	0.0379	-124.22	< 2e-16

Dealing With Heteroscedascity From Group Effects

- `Stata` calculates Huber-White standard errors differently by using the same coefficient estimates as the regular linear model results, but scaling the variance covariance matrix by the degrees of freedom:

$$VC^{\texttt{Stata}} = \frac{M}{(M-1)}\frac{(N-1)}{(N-K)} \times f_{\text{robust}}(\mathbf{X}'\mathbf{X})^{-1}(\mathbf{U}'\mathbf{U})(\mathbf{X}'\mathbf{X})^{-1}$$

- This is easily obtained in `R` with the following:

```
library(sandwich)
hw.se <- sqrt(diag(vcovHC(tweed.lm,type="HC1")))
cbind(tweed.lm$coef,hw.se)
```

```
(Intercept)                                    252.06960 219.06530
year                                            -0.12333   0.11027
arrests                                          5.35855   6.04840
factor(attitude)Ethnic/regionalist Separatist   -6.37254   3.05351
factor(attitude)Left wing extremist Other       -6.78818   3.14835
factor(attitude)Right wing extremist Other      -4.70957   3.41273
```