R language and data analysis: Linear regression 2

Qiang Shen

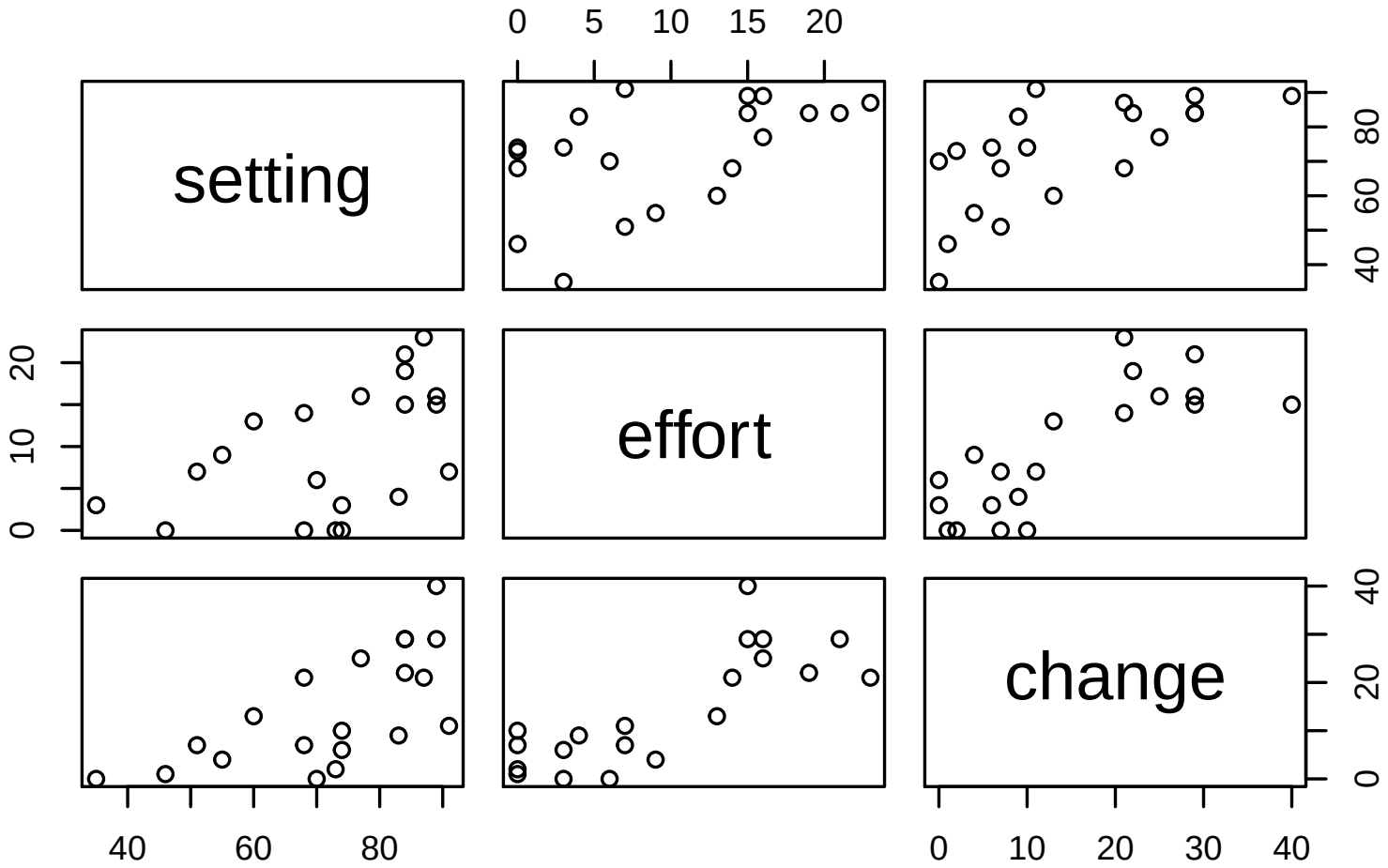
Jan.2, 2018

raw data

This data consist of observations on an index of social setting, an index of family planning effort, and the percent decline in the crude birth rate (CBR) between 1965 and 1975, for 20 countries in Latin America.

	setting	effort	change
Bolivia	46	0	1
Brazil	74	0	10
Chile	89	16	29
Colombia	77	16	25
CostaRica	84	21	29
Cuba	89	15	40

```
library(foreign)
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
pairs(fpe)
```



NULL model.

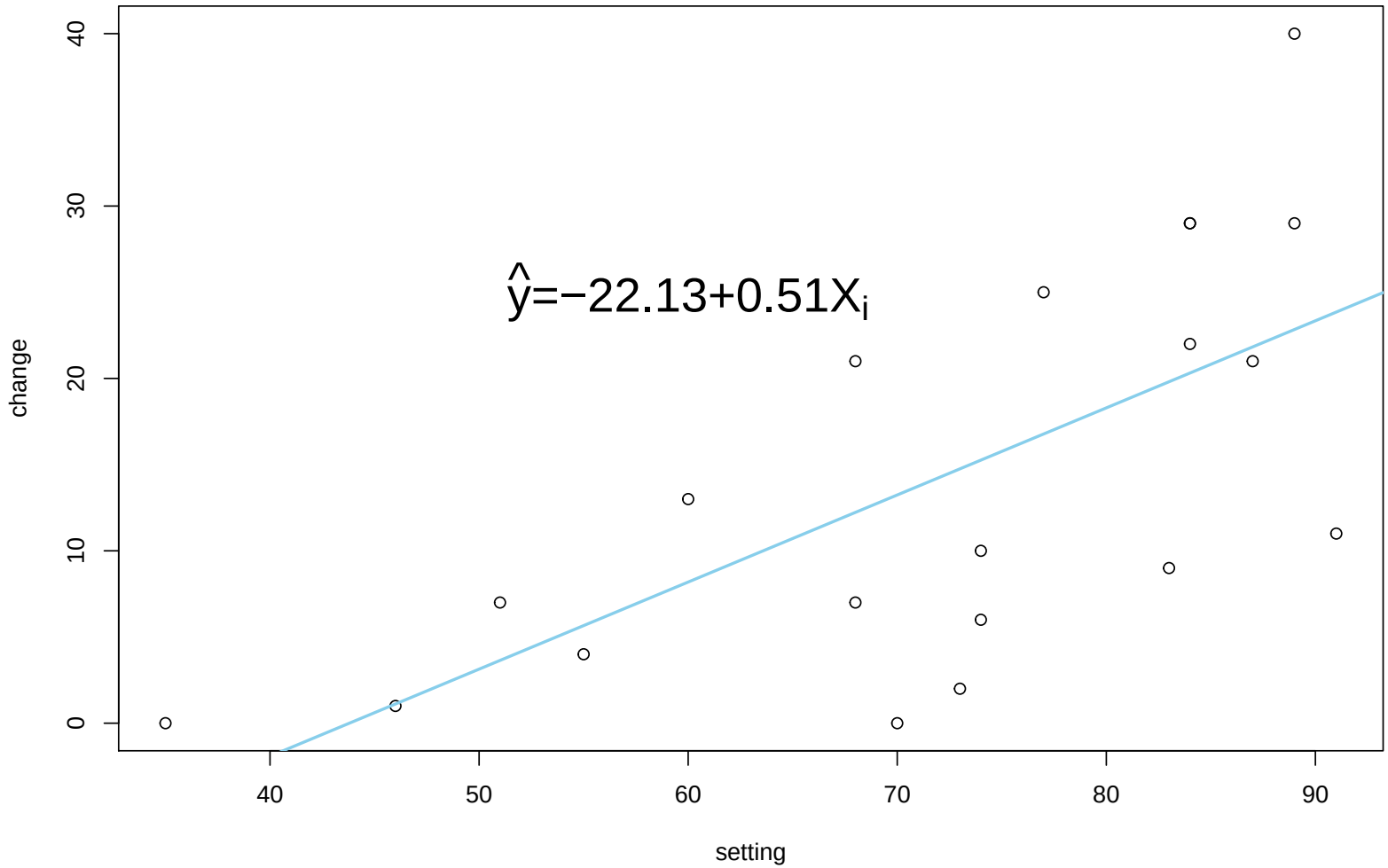
```
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
m0 = lm(change ~ 1, fpe)
library(knitr)
kable(summary(m0)$coef, digits=2)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	14.3	2.64	5.41	0

```
library(psych);
kable(describe(fpe$change),digits=2)
```

	vars	n	mean	sd	median	trimmed	mad	min	max
X1	1	20	14.3	11.81	10.5	13.56	14.83	0	40

fitted plot



resudial

$$\epsilon = y - \hat{y}$$

```
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
m0 = lm(change ~ 1, fpe)
m1 = lm(change ~ setting,fpe)
names(m1)
m1$model
m1$fitted
m1$resid
all.equal(m1$model[,1]-m1$fitted,m1$residu)
```

F value

$$ESS = TSS - RSS$$

$$F = \frac{\mathrm{ESS}/k}{\mathrm{RSS}/(n-k-1)}$$

```
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
m1=lm(change~setting,fpe)
knitr::kable(anova(m1))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
setting	1	1201.078	1201.0776	14.91896	0.0011409
Residuals	18	1449.122	80.5068	NA	NA

```
rss <- function(lmfit) {sum(resid(lmfit)^2)}
aov_result<-anova(m1)
aov_result[1,2]/aov_result[1,1]
```

hypothesis testing and p value.

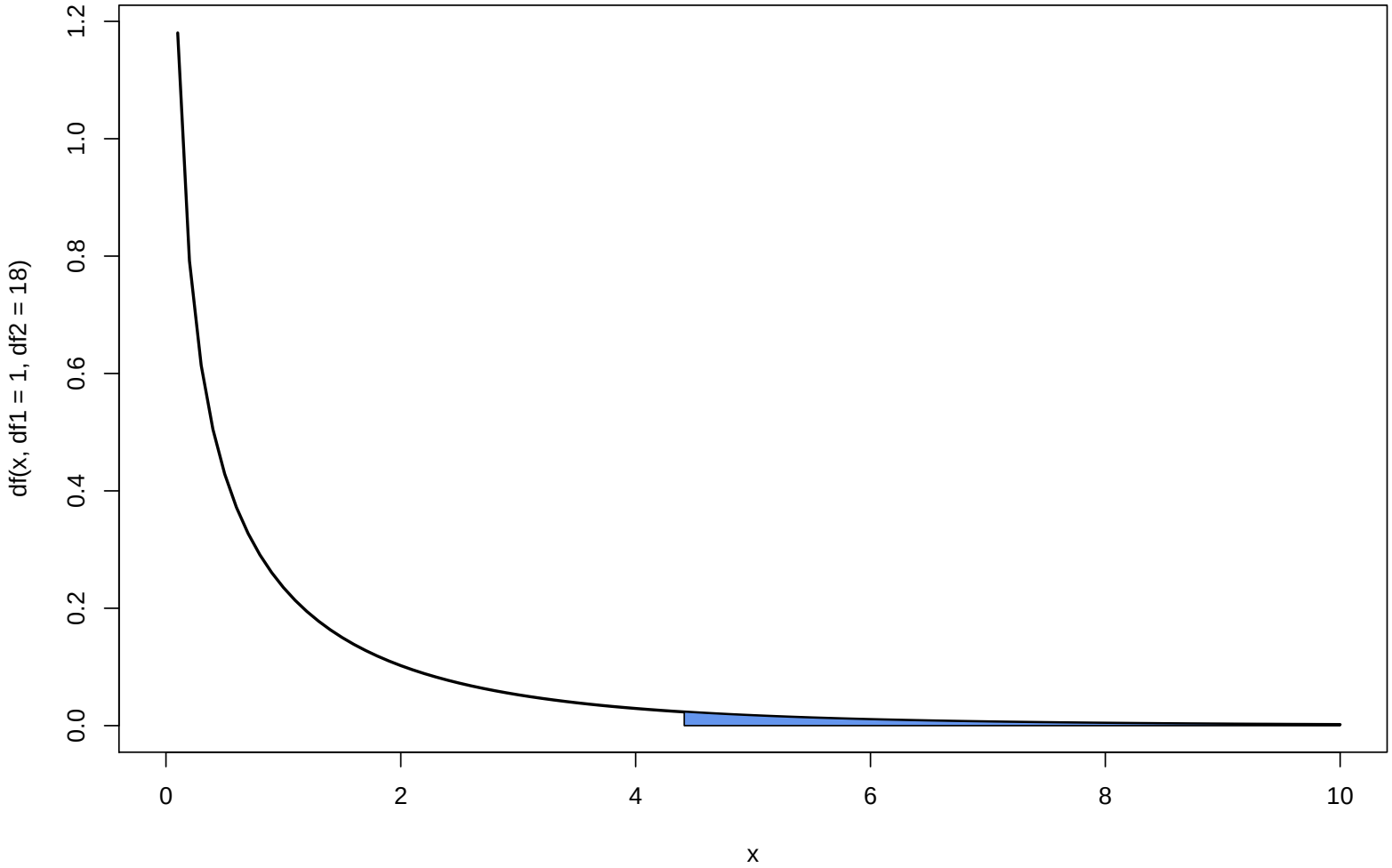
```
qf(.95, 1,18)
```

```
[1] 4.413873
```

```
1-pf(14.9,1,18)
```

```
[1] 0.001147126
```

hypothesis testing and p value.



R square

$$R^2 = 1 - \frac{ESS}{TSS}$$

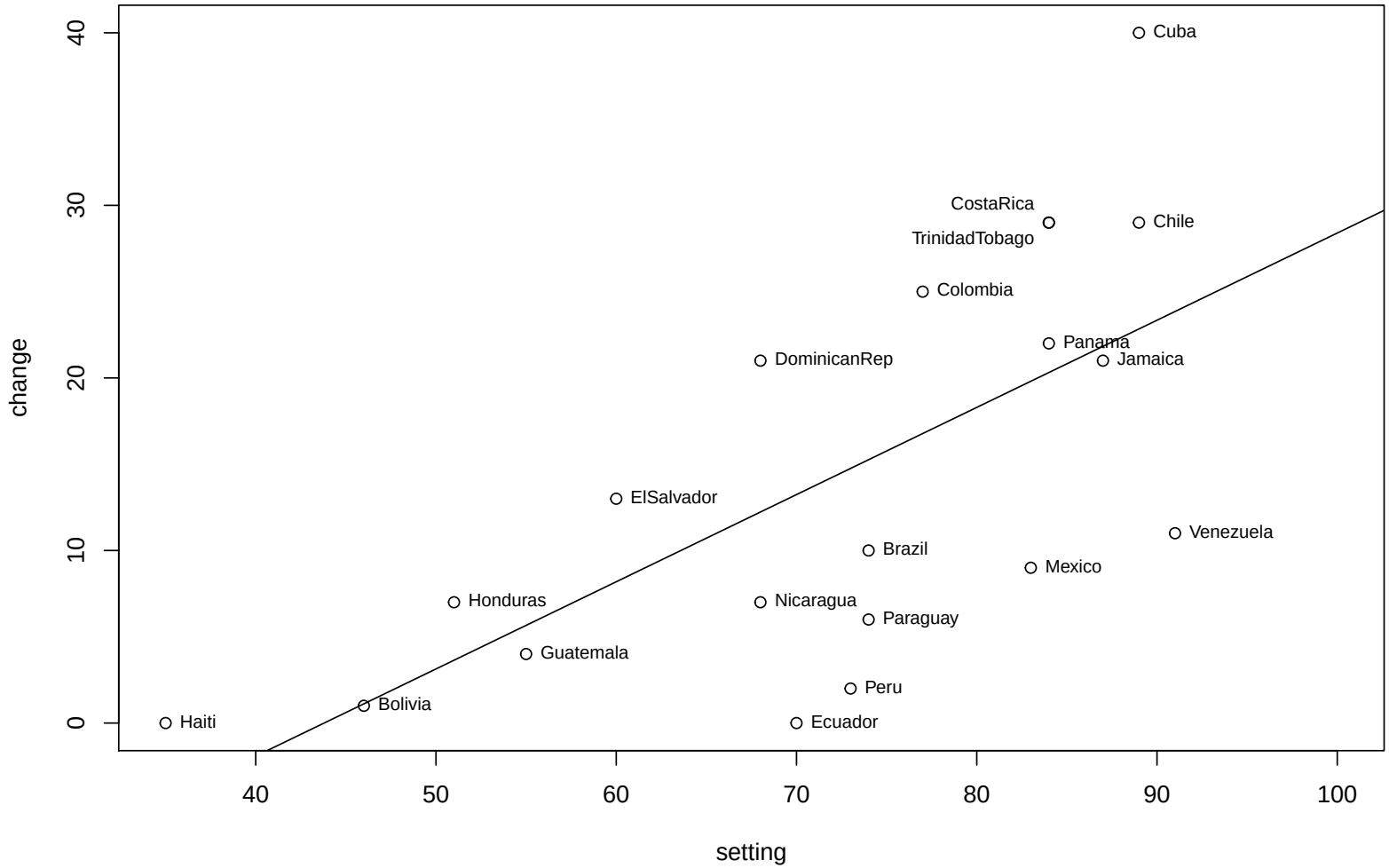
```
library(foreign);fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
m1=lm(change~setting,fpe)
knitr::kable(anova(m1))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
setting	1	1201.078	1201.0776	14.91896	0.0011409
Residuals	18	1449.122	80.5068	NA	NA

```
rss <- function(lmfit) {sum(resid(lmfit)^2)}
m0=lm(change~1,fpe)
1-rss(m1)/rss(m0)
```

```
[1] 0.4528026
```

plot


change
setting
0
10
20
30
40
40
50
60
70
80
90
100
Cuba
CostaRica
Chile
TrinidadTobago
Colombia
Panama
DominicanRep
Jamaica
ElSalvador
Venezuela
Brazil
Mexico
Honduras
Nicaragua
Paraguay
Guatemala
Peru
Bolivia
Haiti
Ecuador

plot

```
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
with(fpe,plot(setting,change,xlim=c(35,100)))
abline(coef(m1))
adj <- data.frame( pos=rep(4,nrow(fpe)),
                   jit=0, row.names=row.names(fpe))
adj[c("CostaRica","TrinidadTobago"),"pos"] <- 2
adj[c("CostaRica","TrinidadTobago"),"jit"] <- c(1,-1)
text(fpe$setting, fpe$change+adj$jit,
     row.names(fpe), pos=adj$pos, cex=0.75)
# dev.print(png,"fig.png",width=600,height=480)
```

multiple regression

```
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
m2 <- with(fpe,lm(change ~ setting + effort))
knitr::kable(summary(m2)$coe,digits=2)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-14.45	7.09	-2.04	0.06
setting	0.27	0.11	2.51	0.02
effort	0.97	0.23	4.30	0.00

```r
fpe<-read.dta('data/effort.dta')
row.names(fpe) <- fpe$country;fpe$country<-NULL
m2 <- with(fpe,lm(change ~ setting + effort))
knitr::kable(anova(m2))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
setting	1	1201.0776	1201.07756	29.42097	0.0000456
effort	1	755.1168	755.11677	18.49695	0.0004841
Residuals	17	694.0057	40.82386	NA	NA

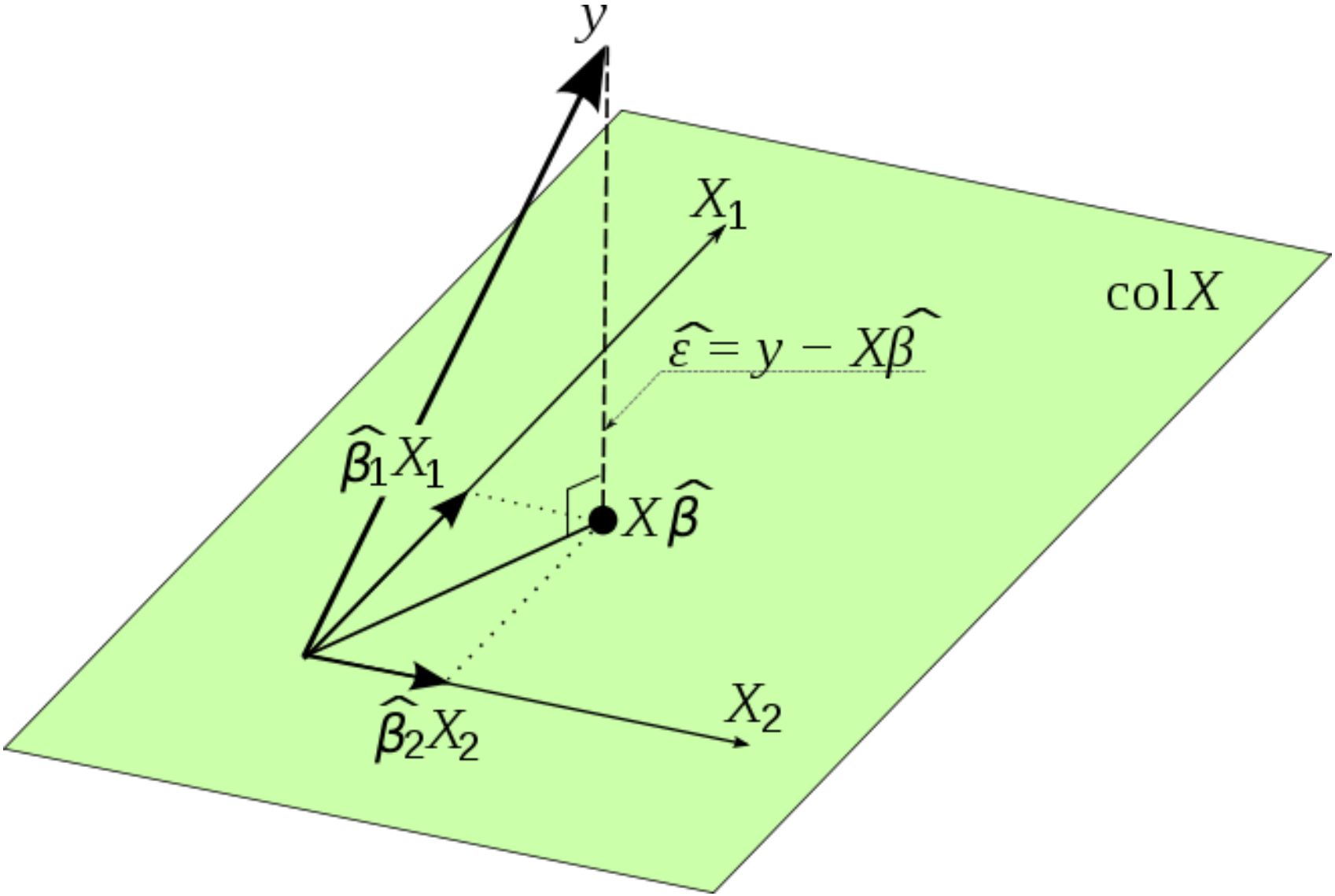
Frisch–Waugh–Lovell (FWL) theorem

$$Y = X_1\beta_1 + X_2\beta_2 + \mu$$

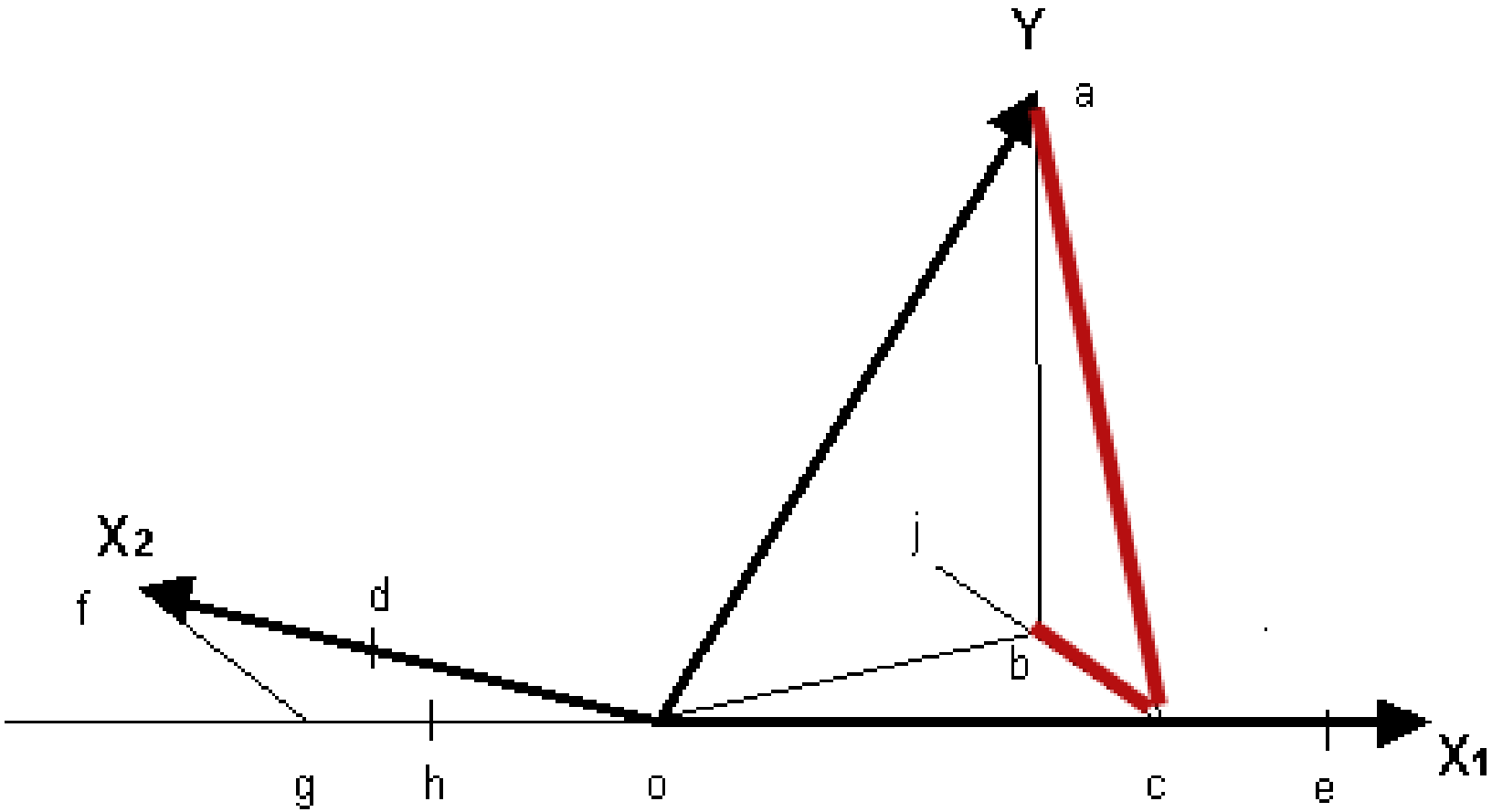
$$M_{X_1} Y = M_{X_1} X_2 \beta_2 + M_{X_1} \mu$$

$$M_{X_1} = I - X_1 (X_1' X_1)^{-1} X_1'$$

Frisch–Waugh–Lovell (FWL) theorem



Frisch–Waugh–Lovell (FWL) theorem



Frisch–Waugh–Lovell (FWL) theorem

```
y=fpe$change;x1=fpe$setting;x2=fpe$effort
r1 = residuals(with(fpe,lm(y ~ x1)))
r2 = residuals(with(fpe,lm(x2 ~ x1)))
# ols
coef(with(fpe,lm(y ~ x1 + x2)))
```

```
(Intercept)          x1          x2
-14.4510978   0.2705885   0.9677137
```

```
# fwl ols
coef(with(fpe,lm(r1 ~ -1 + r2)))
```

```
       r2
0.9677137
```

dummy variable 1

```
setting.g <- cut(fpe$setting,
  breaks=c(min(fpe$setting),70,80,max(fpe$setting)),
  right=FALSE,include.lowest=TRUE,
  labels=c("Low","Medium","High"))
fpe$setting.g=setting.g
write.dta(fpe,'fpe_dummy.dta',convert.factors = 'string')
data.frame(min = tapply(fpe$setting, setting.g, min),
   max = tapply(fpe$setting, setting.g, max))
tapply(fpe$change, setting.g, mean)
```

dummy variable 1: R vs. stata

```
m1g <- lm(fpe$change ~ setting.g)
m1g

settingMedium <- as.numeric(setting.g == "Medium")
settingHigh   <- as.numeric(setting.g == "High")
lm(fpe$change ~ settingMedium + settingHigh)
```

Wald test

$$W = \hat{\alpha} \hat{var}^{-1}(\hat{\alpha}) \hat{\alpha}$$

```
setting.g <- cut(fpe$setting,
  breaks=c(min(fpe$setting),70,80,max(fpe$setting)),
  right=FALSE,include.lowest=TRUE,
  labels=c("Low","Medium","High"))
m1g <- lm(fpe$change ~ setting.g)
b = coef(m1g)[-1]
V = vcov(m1g)[-1,-1]
W = t(b) %*% solve(V) %*% b
W
```

```
          [,1]
[1,] 13.93447
```

```
c(W, W/2)
```

dummy variable 2

```
effort.g <- cut(fpe$effort,
   breaks=c(min(fpe$effort),5,15,max(fpe$effort)),
   right=FALSE, include.lowest=TRUE,
   labels=c("Weak","Moderate","Strong"))
m2g <- lm(fpe$change ~ setting.g + effort.g)
```

anova result

```
effort.g <- cut(fpe$effort,
   breaks=c(min(fpe$effort),5,15,max(fpe$effort)),
   right=FALSE, include.lowest=TRUE,
   labels=c("Weak","Moderate","Strong"))
m2g <- lm(fpe$change ~ setting.g + effort.g)
knitr::kable(anova(m2g))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
setting.g	2	1193.7857	596.89286	15.58761	0.0002176
effort.g	2	882.0226	441.01129	11.51683	0.0009320
Residuals	15	574.3917	38.29278	NA	NA

Symbols in R formula

Symbol	Usage
~	Separates response variables on the left from the explanatory variables on the right. For example, a prediction of y from x, z, and w would be coded `y ~ x + z + w`.
+	Separates predictor variables.
:	Denotes an interaction between predictor variables. A prediction of y from x, z, and the interaction between x and z would be coded `y ~ x + z + x:z`.
*	A shortcut for denoting all possible interactions. The code `y ~ x * z * w` expands to `y ~ x + z + w + x:z + x:w + z:w + x:z:w`.
^	Denotes interactions up to a specified degree. The code `y ~ (x + z + w)^2` expands to `y ~ x + z + w + x:z + x:w + z:w`.
.	A placeholder for all other variables in the data frame except the dependent variable. For example, if a data frame contained the variables `x, y, z, and w`, then the code `y ~ .` would expand to `y ~ x + z + w`.
-	A minus sign removes a variable from the equation. For example, `y ~ (x + z + w)^2 - x:w` expands to `y ~ x + z + w + x:z + z:w`.
-1	Suppresses the intercept. For example, the formula `y ~ x -1` fits a regression of y on x, and forces the line through the origin at x=0.