Analysis of covariance - ANCOVA

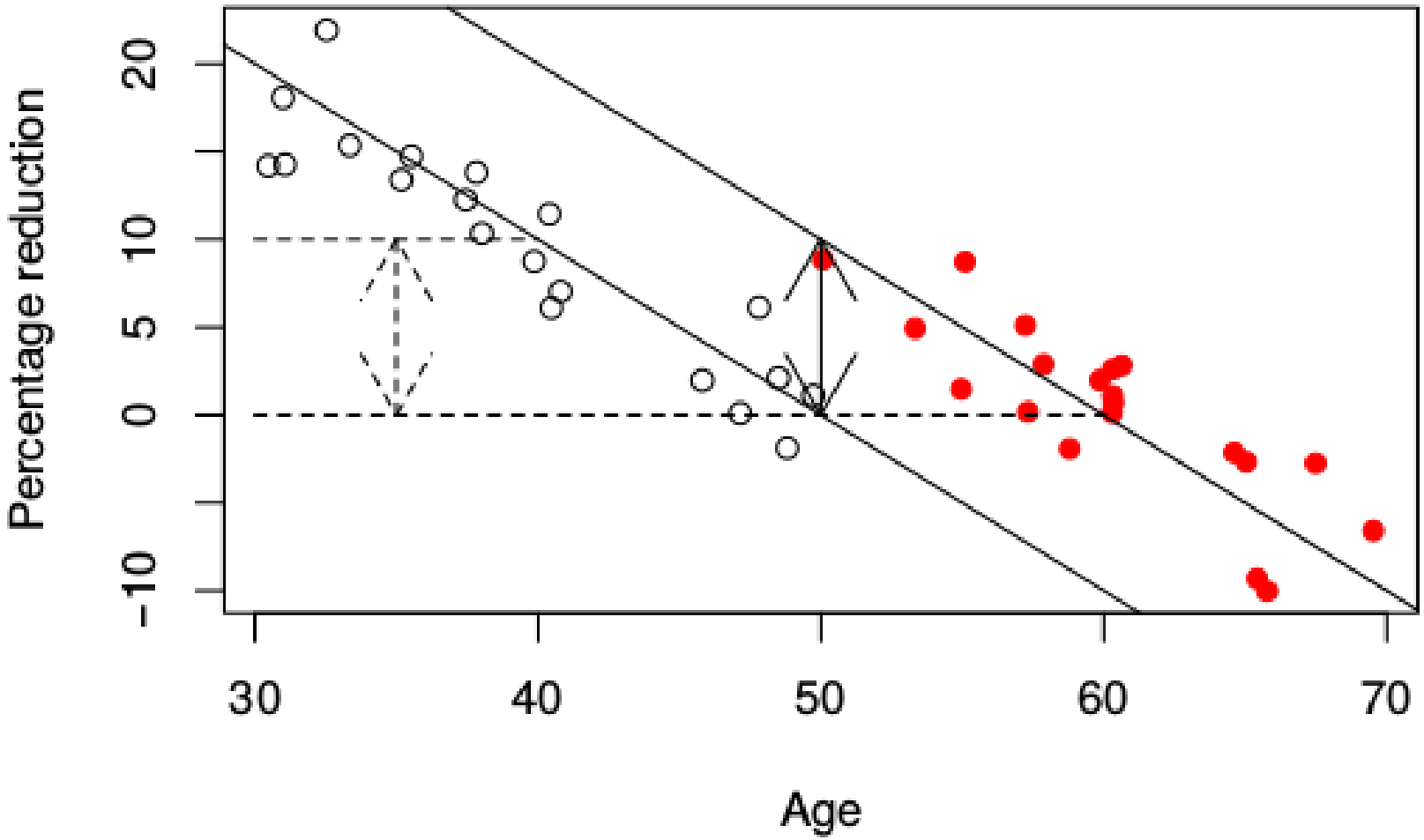
Based on a book by Julian J. Faraway

University of Iceland

Ancova

- Predictors that are qualitative in nature are sometimes called *categorical* or *factors*.
- Analysis of Covariance (ANCOVA) refers to regression problems where there is a mixture of quantitative and qualitative predictors.
- We use ANCOVA for evaluating whether population means of a dependent variable are equal across levels of a categorical independent variable often called a treatment, while controlling for the effects of other continuous variables that are not of primary interest, known as covariates.
- ANCOVA can also be used when there are more than two groups and more than one covariate.

Ancova



The framework

To put qualitative predictors into the $y = X\beta + \varepsilon$ form we need to code the qualitative predictors.

Let's consider a specific example:

$$
\begin{aligned}
y = &\quad \text{change in cholesterol level} \\
x = &\quad \text{age} \\
d = &\begin{cases} 0 & \text{did not take medication} \\ 1 & \text{took medication} \end{cases}
\end{aligned}
$$

A variety of linear models may be considered here.

The framework

Some possible models:

1. The same regression line for both groups:

$$y = \beta_0 + \beta_1 x + \varepsilon.$$

2. Separate regression lines for each group but with the same slope:

$$y = \beta_0 + \beta_1 x + \beta_2 d + \varepsilon.$$

   In this case β_2 represents the distance between the regression lines i.e. the effect of the drug.

3. Separate regression lines for each group y:

$$y = \beta_0 + \beta_1 x + \beta_2 d + \beta_3 x d + \varepsilon.$$

   Any interpretation of the effect of the drug will now depend on age also.

Cathedral example

The data for the example consist of x = nave height and y = total length in feet for English medieval cathedrals. Some are in the Romanesque (r) style and others are in the Gothic (g) style.

We wish to investigate how the length is related to height for the two styles.

```
library(faraway)
data(cathedral)
```

Cathedral example

```
str(cathedral)

## 'data.frame': 25 obs. of  3 variables:
##  $ style: Factor w/ 2 levels "g","r": 2 2 2 2 2 2 2 2 2 1 ...
##  $ x    : num  75 80 68 64 83 80 70 76 74 100 ...
##  $ y    : num  502 522 425 344 407 451 551 530 547 519 ...

head(cathedral)

##                style  x   y
## Durham             r 75 502
## Canterbury         r 80 522
## Gloucester         r 68 425
## Hereford           r 64 344
## Norwich            r 83 407
## Peterborough       r 80 451
```

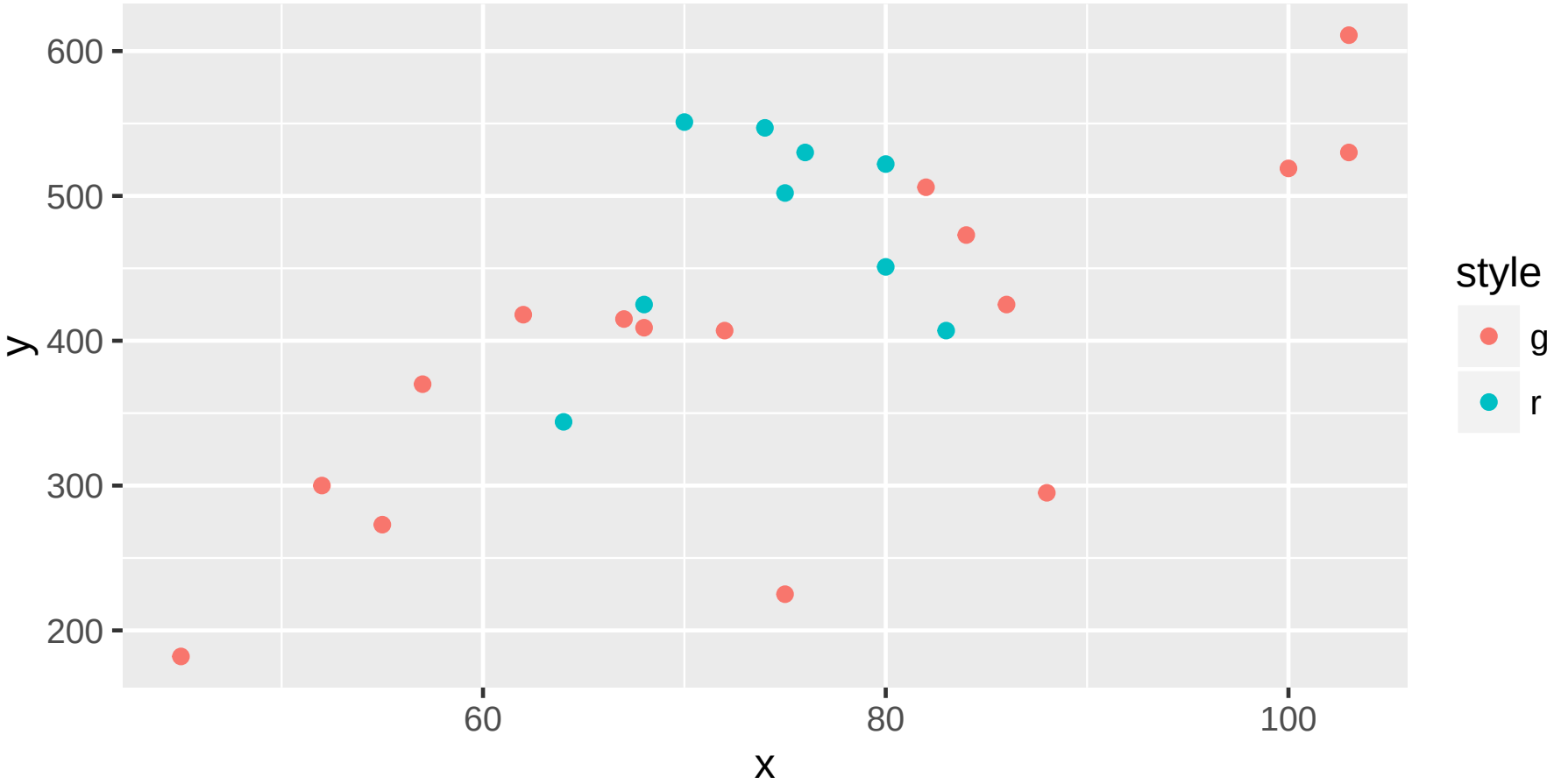
Cathedral example

```
library(plyr) # need to install first
ddply(cathedral, c("style"), summarize,
 mean = round(mean(y), 2),
 sd = round(sd(y), 2))

##   style   mean     sd
## 1     g 397.38 118.19
## 2     r 475.44  72.27
```

```
ddply(cathedral, c("style"), summarize,
 mean = round(mean(x), 2),
 sd = round(sd(x), 2))

##   style  mean    sd
## 1     g 74.94 18.36
## 2     r 74.44  6.21
```

Cathedral example

```
p<-ggplot(cathedral, aes(x=x, y=y, col=style)) + geom_point()
p
```



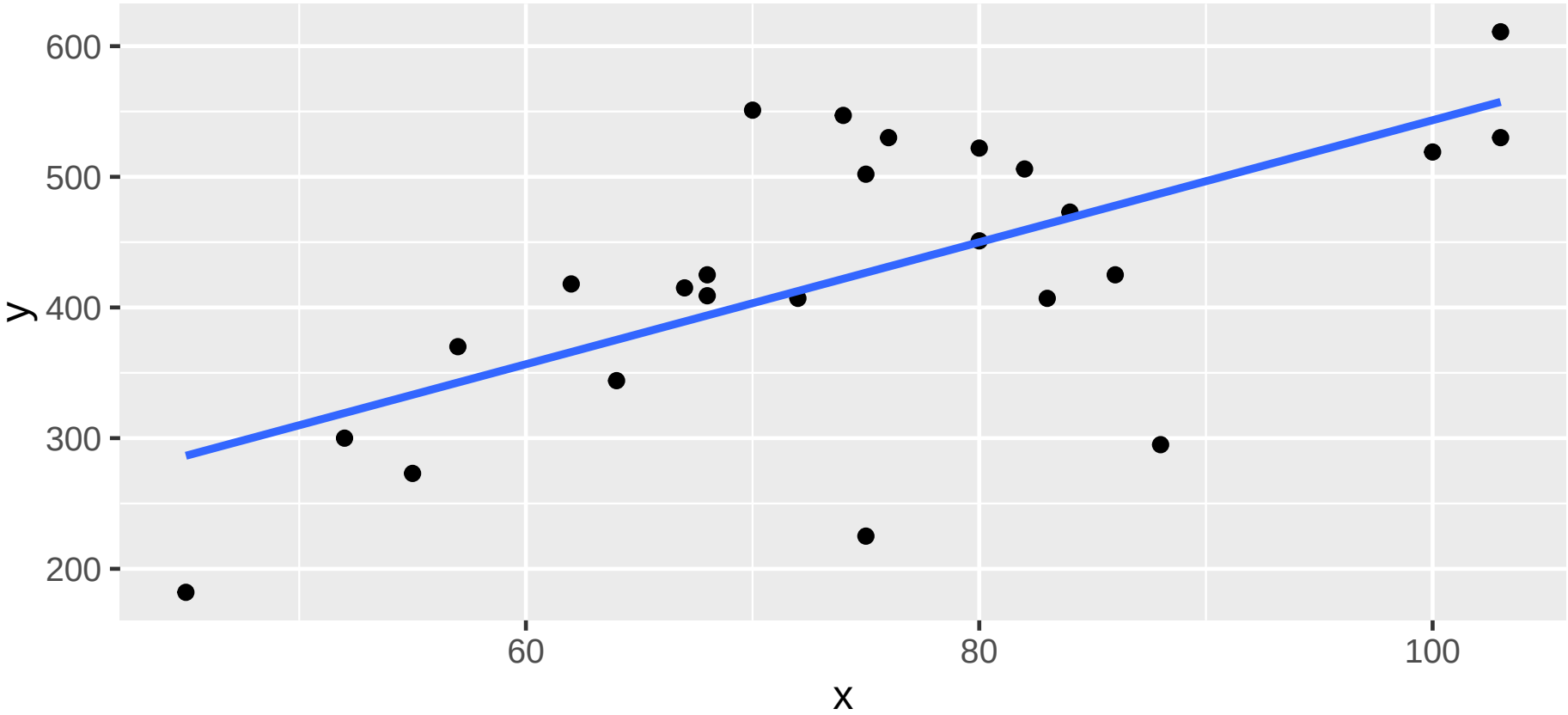
Cathedral example - model 1

```
fit.1<-lm(y~x,data=cathedral)
summary(fit.1)
```

```
##
## Call:
## lm(formula = y ~ x, data = cathedral)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -201.601  -31.241    4.378   52.097  147.745
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   76.420     89.258   0.856 0.400739
## x              4.669      1.172   3.985 0.000584 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.83 on 23 degrees of freedom
## Multiple R-squared:  0.4084, Adjusted R-squared:  0.3827
## F-statistic: 15.88 on 1 and 23 DF,  p-value: 0.0005838
```

Cathedral example - model 1

```
p1 <- ggplot(cathedral, aes(x = x, y = y)) + geom_point()
p1<- p1 + stat_smooth(method="lm",se=F)
p1
```

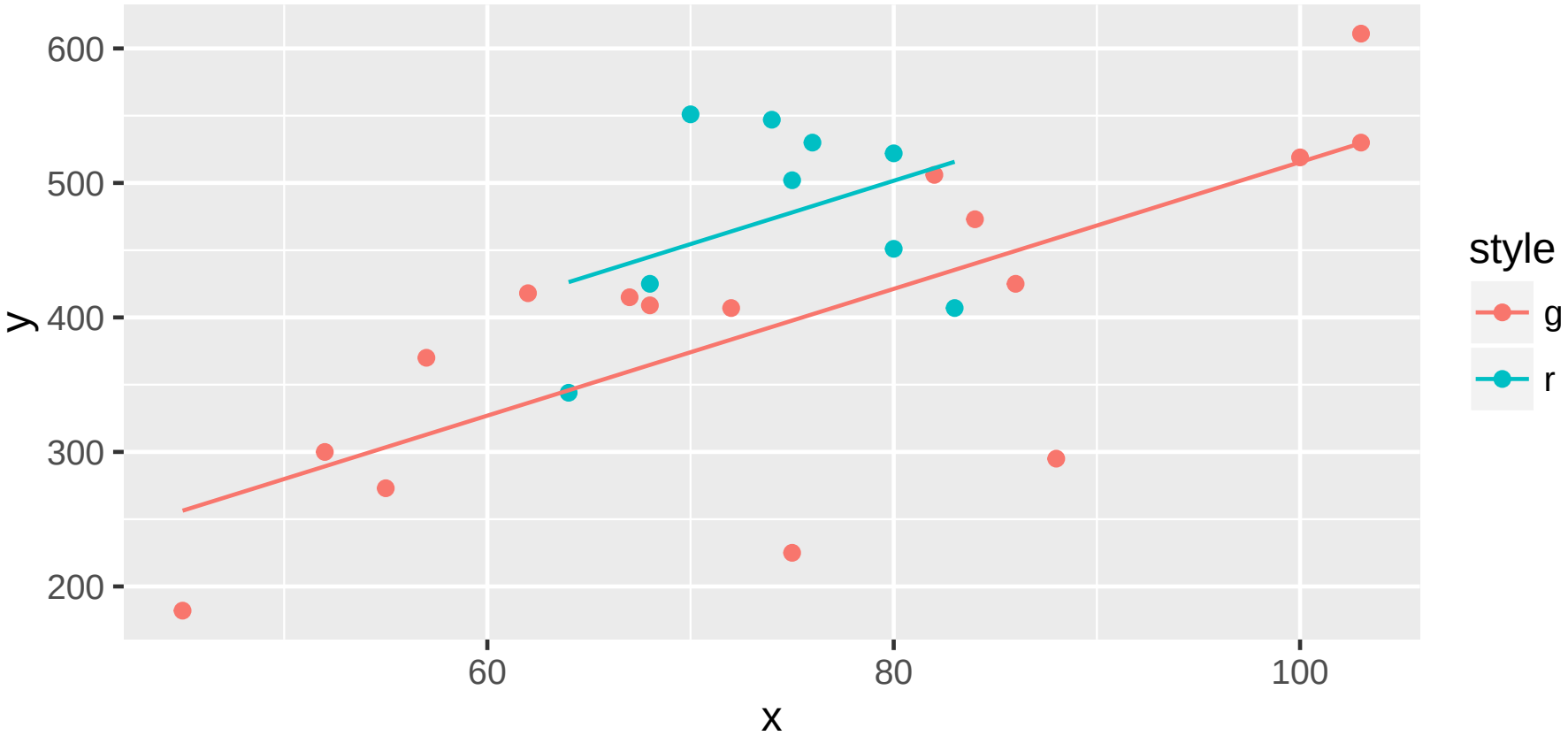


Cathedral example - model 2

```
fit.2<-lm(y~x + style,data=cathedral)
summary(fit.2)


##
## Call:
## lm(formula = y ~ x + style, data = cathedral)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -172.67  -30.44   20.38   55.02   96.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.298     81.648   0.543   0.5929
## x              4.712      1.058   4.452   0.0002 ***
## styler        80.393     32.306   2.488   0.0209 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77.53 on 22 degrees of freedom
## Multiple R-squared:  0.5384, Adjusted R-squared:  0.4964
## F-statistic: 12.83 on 2 and 22 DF,  p-value: 0.0002028
```

Cathedral example - model 2

```
p2 <- ggplot(data = cbind(cathedral, pred = predict(fit.2)),
             aes(x = x, y = y, color = style))
p2 <- p2 + geom_point() + geom_line(aes(y = pred))
p2
```



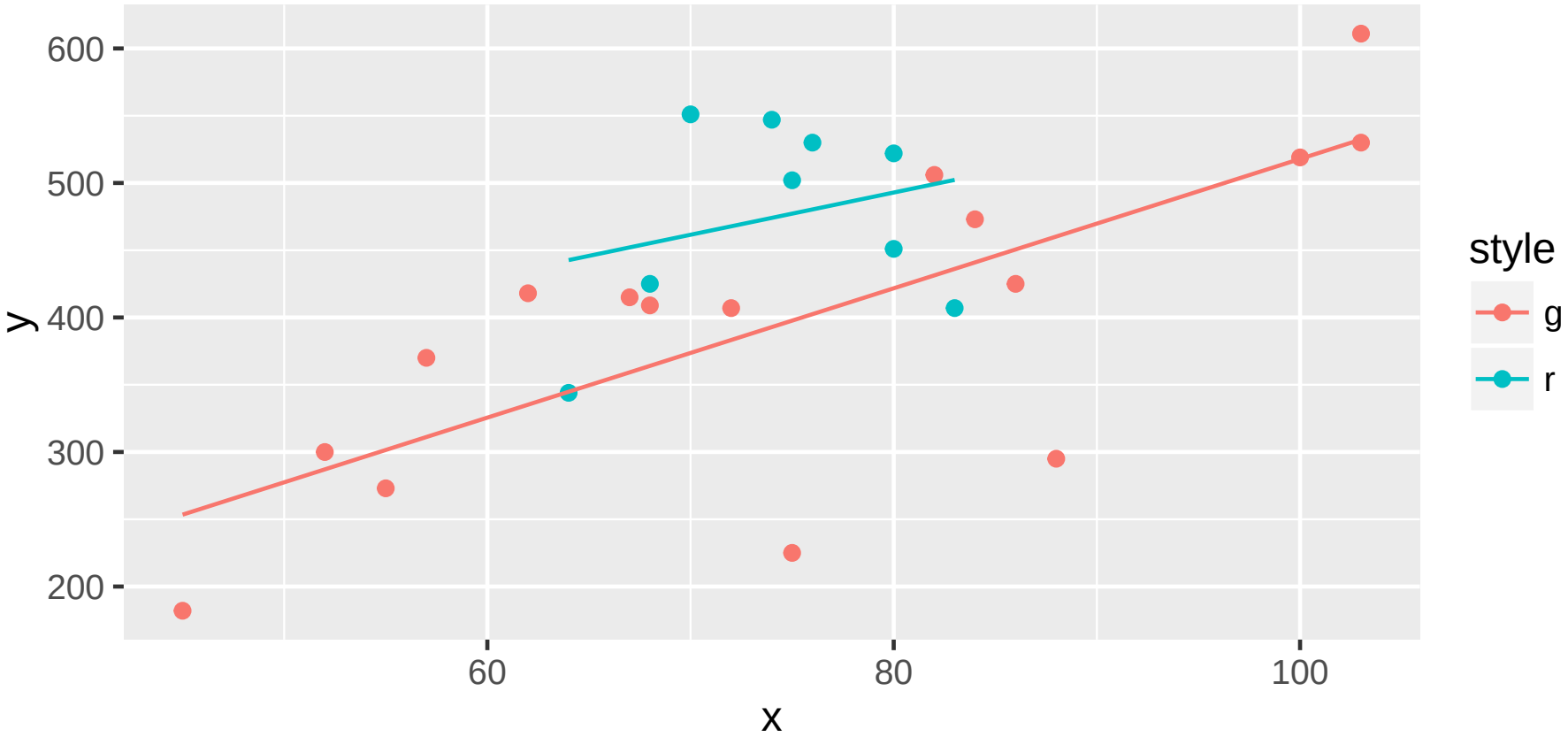
Cathedral example - model 3

```
fit.3<-lm(y ~ x*style,data=cathedral)
summary(fit.3)
```

```
##
## Call:
## lm(formula = y ~ x * style, data = cathedral)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -172.68  -30.22   23.75   55.78   89.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.111     85.675   0.433 0.669317
## x              4.808      1.112   4.322 0.000301 ***
## styler       204.722    347.207   0.590 0.561733
## x:styler      -1.669      4.641  -0.360 0.722657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.11 on 21 degrees of freedom
## Multiple R-squared:  0.5412, Adjusted R-squared:  0.4757
## F-statistic: 8.257 on 3 and 21 DF,  p-value: 0.0008072
```

Cathedral example - model 3

```
p3 <- ggplot(data = cbind(cathedral, pred = predict(fit.3)),
             aes(x = x, y = y, color = style))
p3 <- p3 + geom_point() + geom_line(aes(y = pred))
p3
```



Coding qualitative predictors

- There is no unique coding for a two-level factor - there are even more choices with multi-level predictors.
- For a k-level predictor, $k-1$ dummy variables are needed for the representation.
- One parameter is used to represent the overall mean effect or perhaps the mean of some reference level and so only $k-1$ variables are needed rather than k.
- These dummy variables cannot be exactly collinear but otherwise there is no restriction.
- The choice should be based on convenience.

Treatment coding

Consider a 4 level factor that will be coded using 3 dummy variables. This table describes the treatment coding:

		Dummy coding		
		1	2	3
	1	0	0	0
levels	2	1	0	0
	3	0	1	0
	4	0	0	1

Treatment coding

- Treatment coding treats level one as the standard level to which all other levels are compared to
- If there is a control group, that one would be appropriate for this level.

- R assigns levels to a factor in alphabetical order by default.
- We can change the standard level with the `relevel()` function.
- Treatment coding is the default choice for R.

Coding categorical variables in R

- As stated before, the treatment coding is the default coding.
- We can change the coding of a of a single factor with `contrast()`.
- If we for example wan to change the coding of a categorical variable `catvar` to sum coding we do it with

```
contrasts(catvar)<-'contr.sum'
```

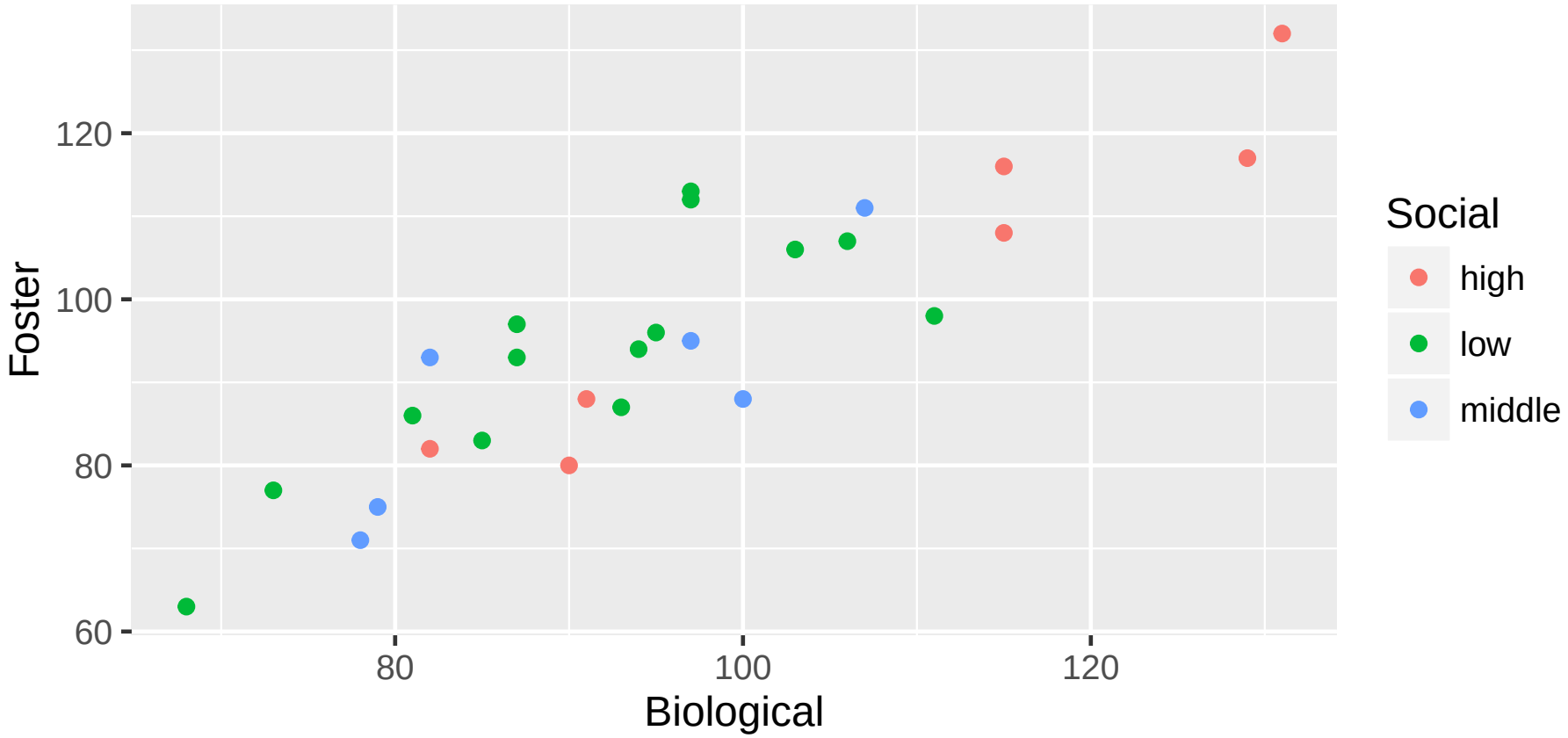
Twins example

The data consist of IQ scores for identical twins, one raised by foster parents, the other by the natural parents. We also know the social class of natural parents (high, middle or low). We are interested in predicting the IQ of the twin with foster parents from the IQ of the twin with the natural parents and the social class of natural parents:

```
data(twins)
str(twins)

## 'data.frame': 27 obs. of  3 variables:
##  $ Foster    : num  82 80 88 108 116 117 132 71 75 93 ...
##  $ Biological: num  82 90 91 115 115 129 131 78 79 82 ...
##  $ Social    : Factor w/ 3 levels "high","low","middle": 1 1 1 1 1 1 1 3
```

Twins example

```
ggplot(twins,aes(x=Biological,y=Foster, color=Social)) + geom_point()
```



Twins example - different slopes and intercepts

```
fit.twins.1<-lm(Foster~Biological*Social,data=twins)
summary(fit.twins.1)


##
## Call:
## lm(formula = Foster ~ Biological * Social, data = twins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.479  -5.248  -0.155   4.582  13.798
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.872044  17.808264  -0.105    0.917
## Biological               0.977562   0.163192   5.990 6.04e-06 ***
## Sociallow                9.076654  24.448704   0.371    0.714
## Socialmiddle             2.688068  31.604178   0.085    0.933
## Biological:Sociallow    -0.029140   0.244580  -0.119    0.906
## Biological:Socialmiddle -0.004995   0.329525  -0.015    0.988
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.921 on 21 degrees of freedom
## Multiple R-squared:  0.8041, Adjusted R-squared:  0.7574
## F-statistic: 17.24 on 5 and 21 DF,  p-value: 8.31e-07
```

Twins example - same slopes different intercepts

```
fit.twins.2<-lm(Foster~Biological+Social,data=twins)
summary(fit.twins.2)
```

```
##
## Call:
## lm(formula = Foster ~ Biological + Social, data = twins)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8235  -5.2366  -0.1111   4.4755  13.6978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.6076    11.8551  -0.051    0.960
## Biological    0.9658     0.1069   9.031 5.05e-09 ***
## Sociallow     6.2264     3.9171   1.590    0.126
## Socialmiddle  2.0353     4.5908   0.443    0.662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.571 on 23 degrees of freedom
## Multiple R-squared:  0.8039, Adjusted R-squared:  0.7784
## F-statistic: 31.44 on 3 and 23 DF,  p-value: 2.604e-08
```

Twins example - same slopes different intercepts

We can compare the two models with the `anova()` function:

```
anova(fit.twins.1,fit.twins.2)

## Analysis of Variance Table
##
## Model 1: Foster ~ Biological * Social
## Model 2: Foster ~ Biological + Social
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     21 1317.5
## 2     23 1318.4 -2  -0.93181 0.0074 0.9926
```

Twins example - same line for both groups

```
fit.twins.3<-lm(Foster~Biological,data=twins)
summary(fit.twins.3)


##
## Call:
## lm(formula = Foster ~ Biological, data = twins)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3512  -5.7311   0.0574   4.3244  16.3531
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.20760    9.29990   0.990    0.332
## Biological   0.90144    0.09633   9.358  1.2e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.729 on 25 degrees of freedom
## Multiple R-squared:  0.7779, Adjusted R-squared:  0.769
## F-statistic: 87.56 on 1 and 25 DF,  p-value: 1.204e-09
```

Twins example

We can also feed the `anova()` funciton with one lm-object. Then we get the sequential tests - using type I sums of squares.

```
anova(fit.twins.1)

## Analysis of Variance Table
##
## Response: Foster
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## Biological         1 5231.1  5231.1 83.3823 9.279e-09 ***
## Social             2  175.1    87.6  1.3958    0.2697
## Biological:Social  2    0.9     0.5  0.0074    0.9926
## Residuals         21 1317.5    62.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Twins example

We can also use the `Anova()` funciton from the `car` package. Then we get the marginal tests - using type II sums of squares.

```
library(car)
Anova(fit.twins.1)

## Anova Table (Type II tests)
##
## Response: Foster
##                   Sum Sq Df F value    Pr(>F)
## Biological        4674.7  1 74.5132 2.382e-08 ***
## Social             175.1  2  1.3958    0.2697
## Biological:Social    0.9  2  0.0074    0.9926
## Residuals         1317.5 21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```