Basic Course on R: Linear Regression

Elizabeth Ribble*

20-24 May 2019

Contents

*emcclel3@msudenver.edu

Most of the following examples use the data "R_data_January2015.csv" which contains variables on mothers whose babies are either intellectually disabled or developmentally normal.

```
babies <- read.csv("R_data_January2015.csv",header=T,row.names=1)
names(babies)
```

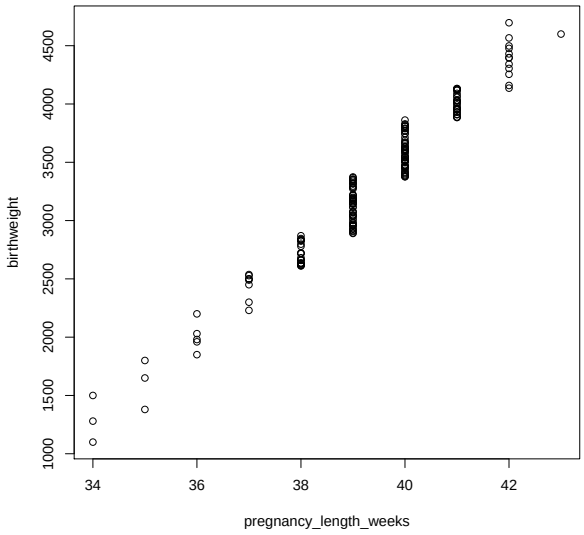
```
##  [1] "Status"                "iodine_deficiency"
##  [3] "BMI"                   "educational_level"
##  [5] "alcohol"               "smoking"
##  [7] "medication"            "birthweight"
##  [9] "pregnancy_length_weeks" "pregnancy_length_days"
## [11] "SAM"                   "SAH"
## [13] "homocysteine"          "cholesterol"
## [15] "HDL"                   "triglycerides"
## [17] "vitaminB12"            "folicacid_serum"
## [19] "folicacid_erys"
```

```
attach(babies)
```

1 Linear Regression Basics

There is a high positive correlation between birth weight and gestational age, but this says nothing about predictive power of the variables. We would like to explain how gestational age influences changes in birth weight.

```
plot(pregnancy_length_weeks,birthweight)
```



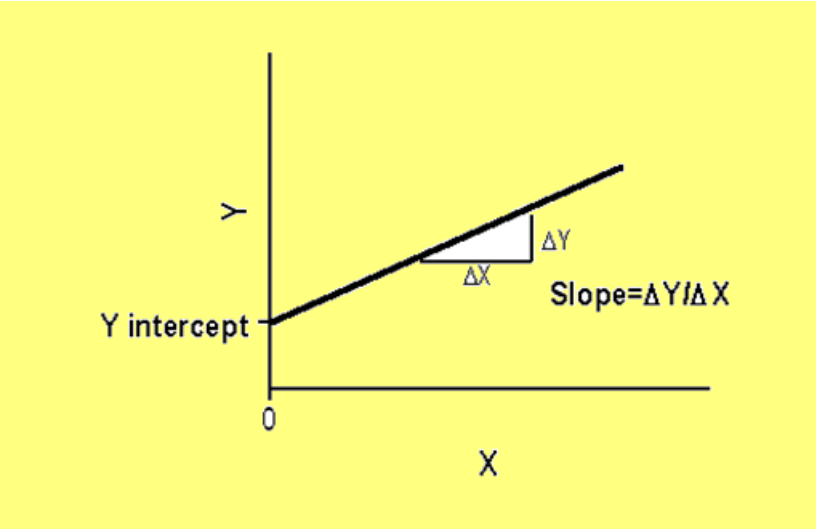

```
cor(pregnancy_length_weeks,birthweight)

## [1] 0.9785784
```

We'll quantify this relationship using linear regression, distinguishing between an independent, or predictor or explanatory, variable (gestational age) and a dependent, or response or outcome, variable (birth weight). Simple linear regression uses the following model:

$$\text{response}_i = (\text{model}_i) + \text{error}_i$$
$$y_i = b_0 + b_1 X_i + \epsilon_i$$

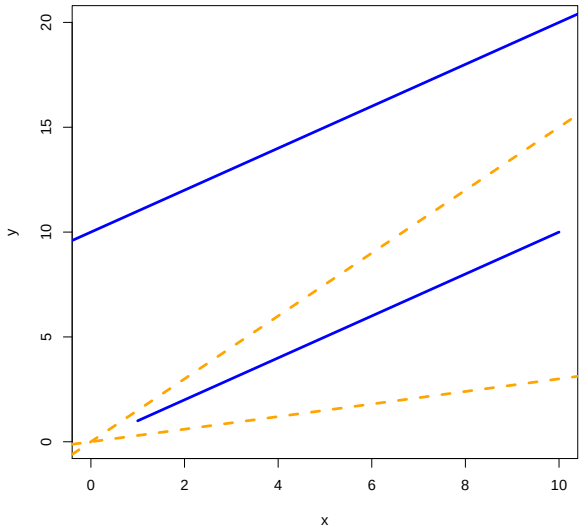
where $1 \leq i \leq n$, model is a straight line, and error is remaining variation which cannot be explained by the model.

The parameters b_0 and b_1 are the intercept and slope of a straight line, respectively:



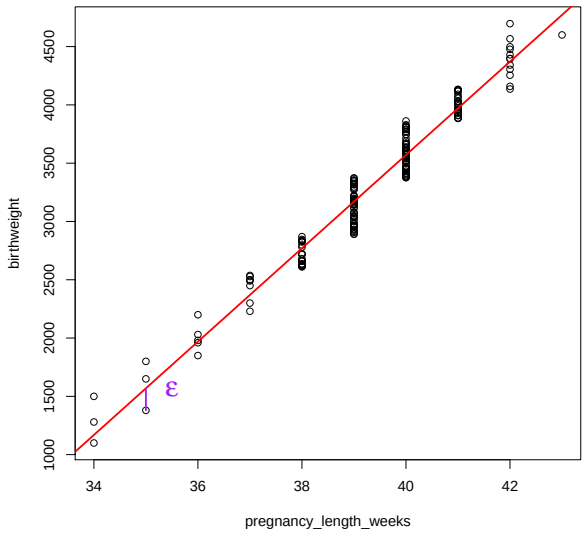
Note that intercept and slope represent different things:

```
plot(1:10, 1:10, type="l", col="blue", lwd=3,
     ylim=c(0,20), xlim=c(0,10), xlab="x", ylab="y")
abline(0, 1.5, lty=2, col="orange", lwd=3)
abline(0, .3, lty=2, col="orange", lwd=3)
abline(10, 1, lty=1, col="blue", lwd=3)
```



The error term ϵ_i denotes the residuals: the differences between observed values and the fitted line.

```r
plot(pregnancy_length_weeks,birthweight)
abline(-12441.4, 400.3, col="red", lwd=2)
lines(c(35,35), c(1380,1568.921), lwd=2, col="purple")
text(35.5, 1560, expression(epsilon), cex=2, col="purple")
```



The b_0 and b_1 of the one straight line that best fits the data is estimated via the method of least squares. The "best" line is the one that has the lowest sum of squared residuals. The command to get these estimates in R is `lm`:

```r
lm1 <- lm(birthweight ~ pregnancy_length_weeks)
lm1

##
## Call:
## lm(formula = birthweight ~ pregnancy_length_weeks)
##
## Coefficients:
##            (Intercept)  pregnancy_length_weeks
##                -12441.4                   400.3
```

and that's how I knew the line in the above plot should have intercept -12441.4 and slope 400.3! Also, the point on the line for 35 weeks is $-12441.4 + 400.3 * 35 = 1568.921$, which can be more accurately provided by

```
predict(lm1)[pregnancy_length_weeks==35]

##       12       73      141
## 1568.921 1568.921 1568.921
```

You can also use `predict` to predict y for x's that are not already in your data:

```
predict(lm1, newdata=data.frame(pregnancy_length_weeks=c(seq(25,50,5))))

##          1          2          3          4          5          6
## -2434.0284  -432.5538  1568.9207  3570.3952  5571.8698  7573.3443
```

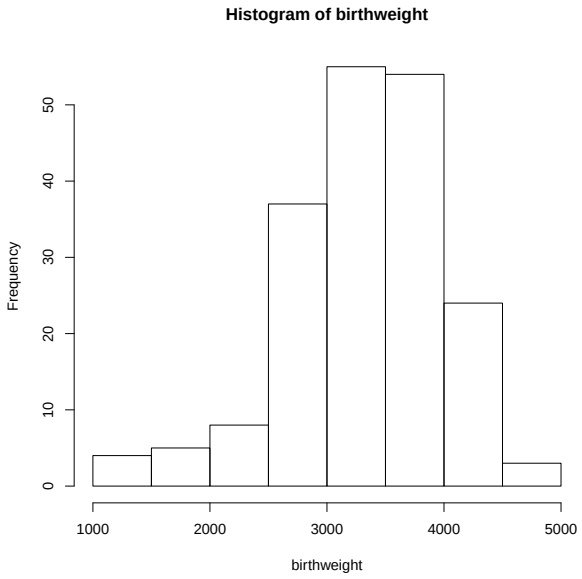
but watch out for extrapolating (predicting outside the range of your data) - clearly we can't have negative birth weights!

Now, before we make inference on or, actually, even find and use this line, we **must** check that the following assumptions hold, otherwise we will not obtain trustworthy results:

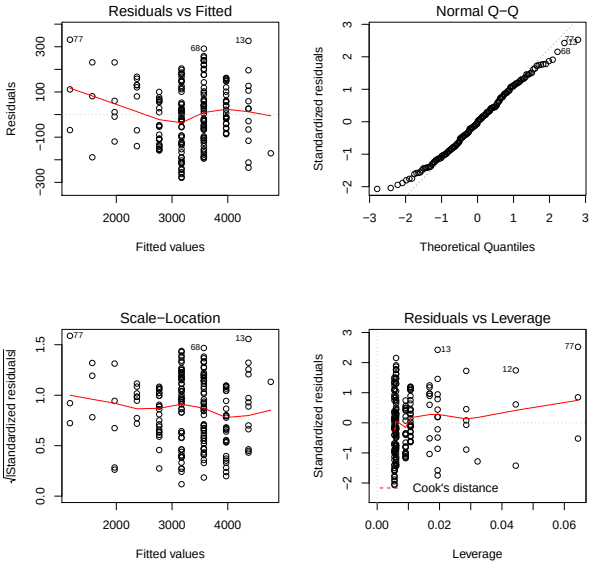
- relationship between x and y can be described by a straight line
- outcomes y are independent
- variance of residuals is constant across values of x
- residuals follow a normal distribution

To get diagnostic plots in R, we can check a histogram of the data and additionally plot our model to check that the residuals are normal and homoscedastic (have constant variance) across the weeks:

```
hist(birthweight)
```

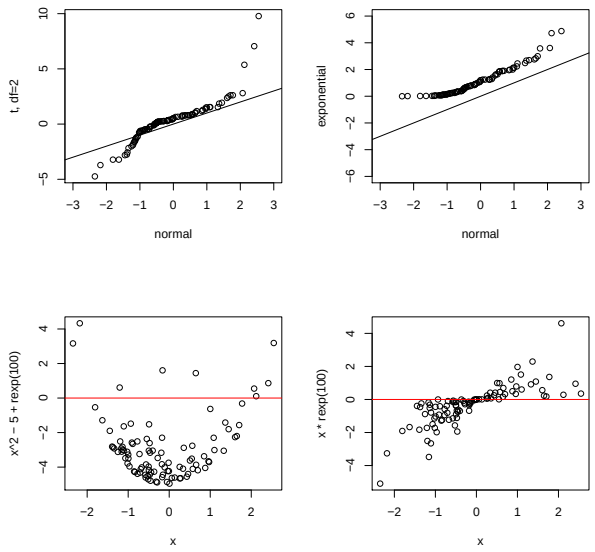



```
par(mfrow=c(2,2))
plot(lm1)
```



The top left plot tells us if our residuals are homoscedastic, and the top right plot displays a quantile-quantile (QQ) plot to check for normality. Here are examples of bad QQ plots and heteroscedasticity:

```
set.seed(1234)
par(mfrow=c(2,2))
x <- sort(rnorm(100))
y1 <- sort(rt(100,2))
plot(x, y1, xlim=c(-3,3), xlab="normal", ylab="t, df=2")
abline(0, 1)
y2 <- sort(rexp(100))
plot(x, y2, xlim=c(-3,3), ylim=c(-6,6), xlab="normal", ylab="exponential")
abline(0, 1)
plot(x, x^2-5+rexp(100))
abline(0, 0, col="red")
plot(x, x*rexp(100))
abline(0, 0, col="red")
```



However, the assumptions reasonably hold for our baby data, so we'll go ahead and use the model fit to make inference on the slope b_1.

Actually, R has already done the inference...we just need to extract it from the model:

```
summary(lm1)

##
## Call:
## lm(formula = birthweight ~ pregnancy_length_weeks)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -280.1 -106.7   -2.9  101.2  331.4
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -12441.401    242.033  -51.40   <2e-16 ***
## pregnancy_length_weeks    400.295      6.142   65.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 135.7 on 188 degrees of freedom
## Multiple R-squared:  0.9576,Adjusted R-squared:  0.9574
## F-statistic:  4248 on 1 and 188 DF,  p-value: < 2.2e-16
```

R has performed a one-sample t-test on the intercept b_1 and slope b_0 to determine if they are each statistically significantly different from 0. The probabilities are quite small, so we can reject the null hypothesis that they are equal to 0 and conclude that birthweight significantly increases, on average, by 400 grams per every additional week of gestation. The intercept is (usually) unimportant and we don't really care that it is different from 0. If the p-value for the slope is not small (e.g. greater than 0.05) then we would say "we do not have enough evidence to reject the null hypothesis that the slope is 0."

Now, how good does the model actually fit our data? How well does x predict y? In a previous lecture, I told you that the square of Pearson's correlation coefficient, r, is a measure of goodness of fit. It is the proportion of variance in y that can be explained by the model (so, x). In our example, r^2 is:

```
cor(pregnancy_length_weeks,birthweight)^2
```

```
## [1] 0.9576157
```

```
summary(lm1)$r.squared
```

```
## [1] 0.9576157
```

which means that 96% of the variability in birth weight can be explained by gestational age.

2 Multiple Linear Regression

Simple linear regression models one y on one x. If we have multiple predictor variables, we use multiple linear regression to determine if the variability in y can be explained by this set of variables. In addition to the assumptions required for a valid simple linear regression, we now include that the covariates have no perfect multicollinearity, that is there is no strong correlation between the multiple x's. The model is

$$y = b_0 + b_1 X_1 + b_2 X_2 + ... + b_k X_k + \epsilon_i$$

In R, the addition of an extra variable is quite straightforward:

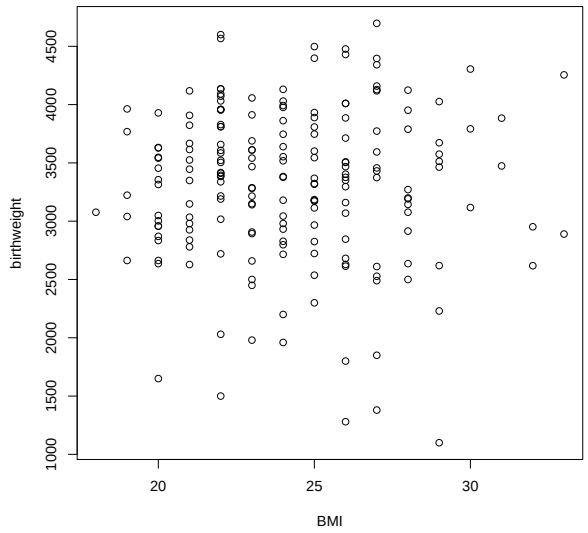
```
cor(pregnancy_length_weeks, BMI)

## [1] 0.01068054

lm2 <- lm(birthweight ~ pregnancy_length_weeks + BMI)
lm2stats <- summary(lm2)
lm2stats

##
## Call:
## lm(formula = birthweight ~ pregnancy_length_weeks + BMI)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -279.39 -105.78   -1.85  105.25  333.50
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -12377.339    253.444 -48.837   <2e-16 ***
## pregnancy_length_weeks    400.351      6.147  65.133   <2e-16 ***
## BMI                        -2.738      3.190  -0.858    0.392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 135.8 on 187 degrees of freedom
## Multiple R-squared:  0.9578,Adjusted R-squared:  0.9573
## F-statistic:  2121 on 2 and 187 DF,  p-value: < 2.2e-16
```

Since the 2 variables are uncorrelated, we can add BMI to the model. We see that it does not significantly predict birth weight, but gestational age still does. We use the adjusted r^2 to check goodness of fit:

```
lm2stats$adj.r.squared

## [1] 0.9573304
```

So adding BMI does not help explain any of the variability in birth weight (since r^2 was previously already 0.96). This is also confirmed by visualization:

```
plot(BMI,birthweight)
```



Note that the `summary` function returns a lot of information. If, for example, you wanted to extract only the p-values you could do the following:

```
names(lm2stats)

##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"

lm2stats$coef[,4]

##            (Intercept) pregnancy_length_weeks                     BMI
##          2.179929e-108          1.815592e-130           3.918778e-01
```

We can predict birthweights with new data:

```r
predict(lm2, newdata=data.frame(pregnancy_length_weeks=32,
                                BMI=30))
```

```
##       1
## 351.771
```

```r
## same as
lm2coefs <- lm2$coef
lm2coefs[1] + lm2coefs[2]*32 + lm2coefs[3]*30
```

```
## (Intercept)
##     351.771
```

But we cannot forget to check assumptions!

```r
par(mfrow=c(2,2))
plot(lm2)
```