3 Regression

3.1 Scatter Plots

The best way of visualizing the relationship between two continuous random variables is through a scatterplot. (Just in passing, the analog for two categorical variables is the Contingency Table, also called the Confusion Matrix.) It can convey a great deal of information, including whether or not the relationship linear, and the extent of the strength of the relationship. Here, strength refers to the skinniness of the scatterplot. Let’s illustrate through an example: Pick 100 random $x$ values, and corresponding $y$ values that have some linear association with $x$ and change the amount of linear association by adding different amounts of “error” to $y$.

```r
par(mfrow = c(1, 2))
x <- runif(100, -1, 1)  # Take 100 points from a uniform distribution between
# -1 and 1.
hist(x)  # The shape and looks uniform.
error <- rnorm(100, 0, 0.1)  # Generate a normal variable (the error), with mu=0,
# sigma=0.1
hist(error)  # The shape looks normal.
```
y_1 <- 2 * x  # Perfect linear relation between x and y.
y_2 <- 2 * x + error  # With some error added to y.
y_3 <- 2 * x + rnorm(100, 0, 0.5)  # With more error added to y.
y_4 <- 2 * x + rnorm(100, 0, 1.0)

par(mfrow = c(2, 2))
plot(x, y_1, cex = 0.5)
plot(x, y_2, cex = 0.5)
plot(x, y_3, cex = 0.5)
plot(x, y_4, cex = 0.5)  # Note that too much noise makes it hard to see
                 # the linear relationship between x and y.
3.2 Correlation

To quantify the strength of the association between two continuous variables, Pearson's correlation coefficient (i.e., correlation), can be computed. It measures the 'amount of scatter' (i.e., skinniness) in a linear sense (but NOT about “the fit”).

```r
cor(x, y_1) # Check against the scatterplots, to get a feeling for r (correlation).
```

```
[1] 1
```

```r
cor(x, y_2)
```

```
[1] 0.9962
```

```r
cor(x, y_3)
```

```r
```
0.9013

cor(x, y_4)
[1] 0.7445

cor(y_4, x)  # r is symmetric.
[1] 0.7445

cor(y_4, x + 10)  # r is invariant under shifts.
[1] 0.7445

cor(x, 10 * y_4)  # r is invariant under scaling.
[1] 0.7445

3.2.1 Defects of Correlation

Pearson’s correlation coefficient, r, can become misleading in several situations.

```
set.seed(123)  # Set a seed to get reproducible results.
x <- runif(100, 0, 1)
error <- rnorm(100, 0, 0.5)
y <- 1 + 2 * x + error
x_1 <- rnorm(100, 0, 50)
y_1 <- rnorm(100, 0, 50)
x_2 <- 1000 + rnorm(100, 0, 50)
y_2 <- 1000 + rnorm(100, 0, 50)
plot(x, y, main = 'Without Outliers', cex = 0.5)
cor(x, y)
[1] 0.7562

# Effect of outliers:
x[101] <- 0.2  # Adding one outlier can artificially reduce r.
y[101] <- 8.0
plot(x, y, main = 'With Outlier (0.2, 8.0)', cex = 0.5)
cor(x, y)
[1] 0.516

x[101] <- 2.0  # A different outlier can artificially increase r.
y[101] <- 8.0
plot(x, y, main = 'With Outlier (2.0, 8.0)', cex = 0.5)
cor(x, y)
[1] 0.8129

# Clusters can also make r meaningless.
plot(x_1, y_1, main = 'Cluster 1', cex = 0.5)
cor(x_1, y_1)  # No correlation between x and y in cluster 1
```
**Without Outliers**

```
plot(x_2, y_2, main = 'Cluster 2', cex = 0.5)
cor(x_2, y_2)  # No correlation between x and y in cluster 2
```

**With Outlier (0.2, 8.0)**

```
x <- c(x_1, x_2)  # Combine/concatenate the 2 clusters.
y <- c(y_1, y_2)
plot(x, y, main = 'Combined Clusters', cex = 0.5)
cor(x, y)  # R incorrectly sees a correlation between x and y.
```

**With Outlier (2.0, 8.0)**

**Cluster 1**
The moral is this: Use $r$ to measure linear correlation, but always examine the data (e.g. with a scatterplot) to make sure things are okay.

### 3.2.2 Example: Ecological Correlation

The following example illustrates another way in which the value of $r$ can be “artificially” increased, i.e., by averaging over things before computing $r$. For similar reasons, regression results (discussed in later sections) can be misleading as well.

```r
dat <- read.table('3_17_dat.txt', header = TRUE)
x <- dat[, 1]
y <- dat[, 2]
z <- dat[, 3]

plot(x, y)  # Making a scatter plot.
cor(x, y)   # Moderate correlation of 0.733 between the 9 pairs.
[1] 0.7329

xbar <- numeric(3)  # Allocating space for storing the time-averaged values of x.
ybar <- numeric(3)   # and of y.
xbar[1] <- mean(x[z == 1])  # This averages x values only when time = 1.
ybar[1] <- mean(y[z == 1])
xbar[2] <- mean(x[z == 2])  # USE UP-ARROW.
ybar[2] <- mean(y[z == 2])
xbar[3] <- mean(x[z == 3])
ybar[3] <- mean(y[z == 3])

plot(xbar,ybar)  # Scatterplot of the 3 averaged pairs,
cor(xbar,ybar)   # and their extreme correlation of 0.998 .
[1] 0.9985
```
You can see clearly how it is that averaging tends to increase $r$, by reducing the number of points and their scatter about a line. Looking at the last scatterplot of the original data, but with the three times colored differently, you can see why this magnification of $r$ is happening: Averaging the three pairs for each time, replaces the three points with a single point located in the "middle" of the three. In general, then, averaging tends to reduce the scatter, and hence the resulting $r$ (called the ecological correlation):

```r
plot(x[z == 1], y[z == 1], xlim = range(x), ylim = range(y))  # Scatterplot for time 1
points(x[z == 2], y[z == 2], col = 2)  # time 2 (USE UP-ARROW)
points(x[z == 3], y[z == 3], col = 3)  # time 3
points(xbar, ybar, col = 4)  # and the averaged data.

pdf("ecol.pdf")
plot(x[z == 1], y[z == 1], xlim = range(x), ylim = range(y), xlab = "x",
     ylab = "y", pch = 1, cex = 3)
points(x[z == 2], y[z == 2], col = 1, pch = 2, cex = 3)
points(x[z == 3], y[z == 3], col = 1, pch = 3, cex = 3)
dev.off()

pdf
2
```
3.3 OLS Regression on Simulated Data

Regression (or a line that fits the scatterplot) can be used for prediction. The function `lm()`, which stands for linear model, does runs a regression in R. It fits a curve through a scatterplot, or a surface through higher-dimensional data.

```r
rm(list = ls(all = TRUE))  # Start from a clean slate.
set.seed(123)  # Ensures reproducible results.
x <- runif(100, 0, 1)  # x is uniform between 0 and 1.
error <- rnorm(100, 0, 1)  # Error is normal with mean = 0, sigma = 1.
y <- 10 + 2*x + error  # The real/true line is y = 10 + 2x.
plot(x, y)  # Plot the scatterplot.
cor(x, y)  # Correlation between x and y.
[1] 0.4916
model.1 <- lm(y ~ x)  # Fitting the regression.
model.1  # Note that the estimated coefficients are pretty close to the true ones

Call:
  lm(formula = y ~ x)

Coefficients:
(Intercept)       x
   9.990       1.911
```

```r
abline(model.1)  # Superimposes the fit on the scatterplot.
```

To see what else is returned by `lm()`, use the following command:

```r
names(model.1)
```

```
[1]  "coefficients"  "residuals"  "effects"  "rank"
[5]  "fitted.values"  "assign"  "qr"  "df.residual"
[9]  "xlevels"  "call"  "terms"  "model"
```
3.4 OLS Regression on “Real” Data

```r
x <- c(72, 70, 65, 68, 70)  # Enter data into R.
y <- c(200, 180, 120, 118, 190)  # See 1.1 for alternative ways to enter data.
plot(x, y, cex = 0.5)
cor(x, y)

[1] 0.8692

model.1 <- lm(y ~ x)
abline(model.1)  # Draws the fit
model.1  # Returns the estimated intercept and slope.

Call:
  lm(formula = y ~ x)

Coefficients:
  (Intercept)       x
  -755.1           13.3

summary(model.1)

Call:
  lm(formula = y ~ x)
```
Residuals:
1 2 3 4 5
-1.46 5.11 11.54 -30.31 15.11

Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -755.11 | 272.53 | -2.77 | 0.070 |
| x | 13.29 | 3.95 | 3.37 | 0.044 * |

---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.9 on 3 degrees of freedom
Multiple R-squared: 0.791, Adjusted R-squared: 0.721
F-statistic: 11.3 on 1 and 3 DF, p-value: 0.0436

Example: Regression on Hail Data

In practice, two quantities called “divergence” and “rotate” are measured by Doppler radar, while hail size is measured directly, i.e., on the ground. But if we can relate hail size to divergence and rotate, then we can predict hail size from Doppler radar. In regression lingo, size is the response (or dependent) variable, and the others are predictors (or independent variables, or covariates).

dat <- read.table("hail.dat.txt", header=T)

plot(dat)
cor(dat)  # This shows the correlations between ALL the vars in the hail data

<table>
<thead>
<tr>
<th>Divergence</th>
<th>Rotational_velocity</th>
<th>Hail_size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Divergence</td>
<td>1.0000</td>
<td>0.5496</td>
</tr>
<tr>
<td>Rotational_velocity</td>
<td>0.5496</td>
<td>1.0000</td>
</tr>
<tr>
<td>Hail_size</td>
<td>0.5214</td>
<td>0.5386</td>
</tr>
</tbody>
</table>

size <- dat[, 3]  # Name the 3 columns in dat. Size is in 100th-of-an-inch.
rotate <- dat[, 2]
diverg <- dat[, 1]

model.1 <- lm(size ~ diverg)  # Regression of size and divergence.
plot(diverg, size)  # Viewing the scatterplot.
abline(model.1)  # Viewing the regression line.

model.2 <- lm(size ~ rotate)  # Regression on size and rotation.
plot(rotate, size)
abline(model.2)

Note that it looks like the line is not really going “through” the data; it seems like the line’s slope should be larger. The fit is in fact correct. The line that intuitively (or visually) goes “through” the scatterplot is NOT the regression line, but something else called the “sd line.”
# Decomposing SST into SS_explained and SS_unexplained:  
anova(model.1)

Analysis of Variance Table

Response: size

|    | Df | Sum Sq | Mean Sq | F value | Pr(>|F|) |
|----|----|--------|---------|---------|----------|
| diverg | 1  | 603829 | 603829  | 105     | <2e-16 *** |
| Residuals | 280 | 1616909 | 5775    |         |          |

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

summary(model.1)

Call:

    lm(formula = size ~ diverg)

Residuals:

    Min     1Q Median     3Q    Max
    -126.1   -50.9   -19.8    44.8   262.6

Coefficients:

    Estimate Std. Error  t value Pr(>|t|)
   (Intercept)  33.673      13.361    2.52   0.012 *
 diverg        3.417       0.334    10.23 <2e-16 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 76 on 280 degrees of freedom
Multiple R-squared:  0.272, Adjusted R-squared:  0.269
F-statistic: 105 on 1 and 280 DF,  p-value: <2e-16

# I.e., 27% of the variation in hail size can be attributed to (or explained  
# by the linear relation with) divergence. The typical deviation of hail  
# size about the regression line is 75.99 (100th-of-an-inch) ~ 0.76 (in) ~ 2 (cm).
3.5 Analysis of Variance (ANOVA) in Regression

ANOVA decomposes $SS_T$ (total sum of squares) into $SS_{explained}$ and $SS_{unexplained}$ (SSE).

\begin{align}
SS_{explained} &= \sum_{i=1}^{n} (y_i - \bar{y})^2 \\
SS_{unexplained} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\
SST &= \sum_{i=1}^{n} (y_i - \bar{y})^2
\end{align}

$SS_{explained}$ is converted to a proportion called R squared (a.k.a. coefficient of determination). It measures the proportion of the variability in $y$ that is explained by $x$. It’s a measure of goodness-of-fit.

```r
x <- c(72, 70, 65, 68, 70) # Enter data into R.
y <- c(200, 180, 120, 118, 190) # See 1.1 for alternative ways to enter data.
plot(x, y) # Plot the scatterplot.
cor(x, y) # Correlation between x and y.

[1] 0.8892

model.1 <- lm(y ~ x) # Fitting the regression.
anova(model.1) # Note that SS_explained = 4942 and SSE = 1309

Analysis of Variance Table

Response: y

<table>
<thead>
<tr>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
<td>4942</td>
<td>4942</td>
<td>11.3</td>
</tr>
<tr>
<td>Residuals</td>
<td>3</td>
<td>1309</td>
<td>436</td>
<td></td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

summary(model.1) # R-squared = 0.7906

Call:
    lm(formula = y ~ x)

Residuals:
        1         2         3         4         5
    -1.46      5.11     11.54    -30.31    15.11

Coefficients:

|                | Estimate | Std. Error | t value | Pr(>|t|) |
|----------------|----------|------------|---------|---------|
| (Intercept)    | -755.11  | 272.53     | -2.77   | 0.070   |
| x              |    13.29 |     3.95   | 3.37    | 0.044 * |

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 20.9 on 3 degrees of freedom
Multiple R-squared: 0.791, Adjusted R-squared: 0.721
F-statistic: 11.3 on 1 and 3 DF, p-value: 0.0436

# The R^2 is reported as "Multiple R-squared".

# Note that the R-squared from summary() agrees with 1-(SSE/SST):
1 - (1308.9 / (4942.3 + 1308.9))

[1] 0.7906

\[ SS_{\text{unexplained}} \] (SSE) is converted to a standard deviation (of errors), and denoted as \( se \). This standard deviation of errors is also called standard deviation about regression. Either way it is reported as “Residual standard error 20.9”. Note that it is equal to \( \sqrt{\frac{SSE}{n-2}} \).

\[ \sqrt{1308.9 / (5 - 2)} \]

[1] 20.89

In sum, R-squared and \( se \) together tell you how good the model is. R-squared tells you what percent of the variance in \( y \) can be attributed to \( x \), and \( se \) tells you the typical error, i.e., deviation of data from the line.

To get \( R^2 \) (and nothing else), use the following command:

\[ \text{summary(model.1)} \] \$ r.squared

[1] 0.7906

# Do the following to check that R's SSE really is Sum of Squared Errors:
\( y \_hat \leftarrow \text{predict(model.1)} \) # This is a quick way of getting \( y \_hat \).
\( y \_hat \) # To see the predictions.

1 2 3 4 5
201.5 174.9 108.5 148.3 174.9

34
3.6 Visual Assessment of Goodness-of-Fit

One way to access the goodness of fit is to examine the scatterplot of predicted $y$ versus actual $y$.

```r
y_hat <- predict(model.1)

# Alternatively, you can use the predictions stored in lm() itself:
y_hat <- model.1$fitted.values

# Here is the scatterplot of predicted size vs. actual size:
plot(y, y_hat, cex = 0.5)
abline(0, 1, col = "red")  # Add a diagonal line.
abline(h = mean(y))        # Add a horizontal line at the mean of y (i.e., size).
```

If the model were good, this scatterplot would be symmetrically spread about the red line. But, clearly our model is not good. This scatterplot (of predicted vs. actual) is often a great way of visualizing how well the model is doing. For example, we see that for smaller hail size (i.e., small $x$ value), the predictions of size are all above the diagonal indicating that the model over-predicts the size of small hail. Looking at larger $x$ values, it’s clear that the model under-predicts the size of large hail.

If a model is completely useless, then the predictions will be symmetrically spread about the horizontal line at the mean of $y$. Here, we can see that our model is nearly (but not completely) useless. This kind of model diagnosis can help in coming up with a better model.

Another visual assessment tool is the residual plot. This plot checks different facet of “goodness” (or quality) than the above plot.