6. Linear Regression Analysis

6.1

Introduction to linear regression analysis
Linear regression analysis

- A statistical technique for estimating the relationships among variables.

- Examples:
  - The earning of a person is affected by many factors such as occupation, age, experience, educational attainment, motivation, and innate ability come to mind, and perhaps race and gender.
  - The fuel consumption of a particular model of cars in m.p.g presumably depends on the weight of the car, the horse power, the number of cylinders, and so on.

- Linear regression equation:
  \[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + e \]

Linear models

- The parameters enter linearly, but predictors do not have to be linear:
  \[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 \log X_3 + e \]

- Some non-linear models can be transformed to linearity
  \[ Y = \beta_0 X^{\beta} e \]
  \[ \log Y = \log \beta_0 + \beta_1 \log X + \log e \]
  Or, \[ Y^* = \beta_0^* + \beta_1 X^* + e^* \]
Classic assumptions

- The sample must be representative of the population, e.g. for the inference prediction.
- The predictors (independent variables) must be linearly independent, and are error-free.
- The error term is assumed to be a random variable with a zero mean conditional on the explanatory variables.
- The errors are uncorrelated, and the variance of the error is constant across observations (i.e., homoscedasticity)

Least squares estimation

- The model:
  \[ y_i = \alpha + \beta x_i + e_i \quad \text{where} \quad e_i \sim N\left(0, \sigma^2\right) \]

- The LS method is to find values of the unknown parameters that minimize the sum of squared residuals:
  \[ \sum_i \hat{e}_i^2 = \sum_i (y_i - \hat{y}_i)^2 = \sum_i (y_i - (\hat{\alpha} + \hat{\beta}x_i))^2 \]
Intercept and slope

Differentiating the right hand side with respect to each of the unknown parameters, and setting the first derivatives to zero.

\[ \hat{\alpha} = \bar{y} - \hat{\beta} \bar{x} \]
\[ \hat{\beta} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \]

Residual variance

\[ \hat{\sigma}^2 = \frac{\sum_i \left( y_i - \left( \hat{\alpha} + \hat{\beta} x_i \right) \right)^2}{n - 2} \]
Hypothesis test

- Under the normal assumption of the error terms, the estimator of the slope coefficient will be normally distributed with mean equal to $\beta$ and variance $se(\beta)$.
- A Student’s t-test can be used to test: $\beta=0$

$$t = \frac{\hat{\beta}}{se(\hat{\beta})}$$

where $se(\hat{\beta}) = \sqrt{\frac{\tilde{\sigma}^2}{\sum_i (x_i - \bar{x})^2}}$

6.2

Fitting a simple linear regression model
Data frame

> library(faraway)
> data(gala)
> # show the structure of the gala data frame
> str(gala)
'data.frame': 30 obs. of 7 variables:
$ Species : num 58 31 3 25 2 18 24 10 8 2 ...
$ Endemics : num 23 21 3 9 1 11 0 7 4 2 ...
$ Area : num 25.09 1.24 0.21 0.1 0.05 ...
$ Elevation: num 346 109 114 46 77 119 93 168 71 112 ...
$ Nearest : num 0.6 0.6 2.8 1.9 1.9 8 6 34.1 0.4 2.6 ...
$ Scruz : num 0.6 26.3 58.7 47.4 1.9 ...
$ Adjacent : num 1.84 572.33 0.78 0.18 903.82 ...

Model fitting using \texttt{lm()}

> species <- as.vector(gala$Species)
> elevation <- as.vector(gala$Elevation)
> lm.fit <- lm(species ~ elevation)
Brief output

```r
> lm.fit
Call:
lm(formula = species ~ elevation)
Coefficients:
(Intercept)     elevation
     11.3351       0.2008

> names(lm.fit)
[1] "coefficients" "residuals" "effects" "rank"
[5] "fitted.values" "assign" "qr" "df.residual"
[9] "xlevels" "call" "terms" "model"
```

Extracting result summaries

```r
> summary(lm.fit)
Call:
  lm(formula = species ~ elevation)
Residuals:
   Min      1Q  Median       3Q      Max
Coefficients:
             Estimate  Std. Error  t value  Pr(>|t|)
(Intercept) 11.33511   19.20529    0.590     0.56
Elevation    0.20079    0.03465   5.795 3.18e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 78.66 on 28 degrees of freedom
Multiple R-squared: 0.5454, Adjusted R-squared: 0.5291
F-statistic: 33.59 on 1 and 28 DF,  p-value: 3.177e-06
```
Results (1/4) - model fitted

Call:
\texttt{lm(formula = Species ~ Elevation)}

> \texttt{lm.fit$call}
\texttt{lm(formula = species ~ elevation)}

Results (2/4) – residual variance

Residuals:
\begin{tabular}{ccccccc}
 & Min & 1Q & Median & 3Q & Max \\
\hline
\end{tabular}

> \texttt{round(quantile(lm.fit$residuals),3)}
\begin{tabular}{cccccc}
 & 0% & 25% & 50% & 75% & 100% \\
\hline
\end{tabular}

Question: Do residuals follow a normal distribution?
Q-Q plot of residuals

Results (3/4) – regression coefficients

Coefficients:

|                  | Estimate | Std. Error | t value | Pr(>|t|) |
|------------------|----------|------------|---------|----------|
| (Intercept)      | 11.33511 | 19.20529   | 0.590   | 0.56     |
| Elevation        | 0.20079  | 0.03465    | 5.795   | 3.18e-06 *** |

---

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

> lm.fit$coef

(Intercept)  elevation

11.3351132  0.2007922
Compute intercept and slope

```r
> slope <- cov(species, elevation) / var(elevation)
> slope
[1] 0.2007922
> intercept <- mean(species) - slope * mean(elevation)
> intercept
[1] 11.33511
```

Results (4/4) – model fitting

Residual standard error: 78.66 on 28 degrees of freedom
Multiple R-squared: 0.5454, Adjusted R-squared: 0.5291
F-statistic: 33.59 on 1 and 28 DF, p-value: 3.177e-06

```r
> sqrt(sum(lm.fit$residuals^2)/(30-2))
[1] 78.66154
> R2 <- 1 - sum(lm.fit$residuals^2)/sum((species - mean(species))^2)
> R2
[1] 0.5453625
> 1 - (1 - R2) * (30 - 1) / (30 - 2)
[1] 0.5291255
```
\[ R^2 \text{ - A measure of Goodness of fit} \]

- Coefficient of determination or percentage of variance explained

\[
R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{\text{RSS}}{\text{Total SS(corrected for mean)}}
\]

where: \( 0 \leq R^2 \leq 1 \)

---

Scatterplot & regression line
6.3

A beginner’s program for linear regression analysis

The model

\[ y_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ik}\beta_k + e_i \]

\[ y = X\beta + e \]

where:

\[ x = \begin{pmatrix}
1 & x_{11} & \ldots & x_{1k} \\
1 & x_{21} & \ldots & x_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{nk} & \ldots & x_{nk}
\end{pmatrix}, \quad y = (y_1, y_2, \ldots, y_n)' \]

\[ \beta = (\beta_0, \beta_1, \ldots, \beta_k)' \]

\[ e = (e_1, e_2, \ldots, e_n)' \]
Least squares estimation

- The least squares estimate of $\beta$ minimizes
  \[ e'e = (y - X\hat{\beta})' (y - X\hat{\beta}) \]
  \[ = y'y - 2\hat{\beta}'X'y + \hat{\beta}'X'X\hat{\beta} \]

- Differentiating the right hand side with respect to $\hat{\beta}$ setting to zero
  \[ X'X\hat{\beta} = X'y \quad \rightarrow \quad \hat{\beta} = (X'X)^{-1} X'y \]

Expectation and variance of $\hat{\beta}$

- $E(\hat{\beta}) = (X'X)^{-1} X'E(y) = (X'X)^{-1} X'X\beta = \beta$
- $\text{var}(\hat{\beta}) = (X'X)^{-1} X' \text{var}(y) X (X'X)^{-1}$
  \[ = (X'X)^{-1} X'X (X'X)^{-1} \sigma^2 = (X'X)^{-1} \sigma^2 \]
- $se(\hat{\beta}_i) = \sqrt{(X'X)^{-1}_{ii} \hat{\sigma}^2}$
Residual variance

\[ \hat{\sigma}^2 = \frac{\hat{e}'\hat{e}}{n - p} \]
\[ = \frac{(y - X\hat{\beta})'(y - X\hat{\beta})}{n - p} \]

Linear regression using \textit{lm}

\begin{verbatim}
> lmm.fit<-lm(Species~Area+Elevation+Nearest+Scruz+Adjacent,data=gala)
> summary(lmm.fit)
Call:
  lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent, 
     data = gala)
Residuals:
    Min      1Q  Median      3Q     Max 
-111.679  -34.898   -7.862   33.460  182.584

\end{verbatim}
Linear regression using \textit{lm} (cont.)

Coefficients:

|            | Estimate | Std. Error | t value   | Pr(>|t|) |
|------------|----------|------------|-----------|----------|
| (Intercept)| 7.068221 | 19.154198  | 0.369     | 0.715351 |
| Area       | -0.023938| 0.022422   | -1.068    | 0.296318 |
| Elevation  | 0.319465 | 0.053663   | 5.953     | 3.82e-06 *** |
| Nearest    | 0.009144 | 1.054136   | 0.009     | 0.993151 |
| Scruz      | -0.240524| 0.215402   | -1.117    | 0.275208 |
| Adjacent   | -0.074805| 0.017700   | -4.226    | 0.000297 *** |

---

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

Residual standard error: 60.98 on 24 degrees of freedom
Multiple R-squared: 0.7658,  Adjusted R-squared: 0.7171
F-statistic: 15.7 on 5 and 24 DF,  p-value: 6.838e-07

Set up \( y \), \( X \), \( n \) and \( k \)

\[
> \# \text{ vector } y \text{ and matrix } X \\
> y \leftarrow \text{ gala$Species} \\
> X \leftarrow \text{ cbind}(1, \text{ gala[,c(1,2)]}) \\
> X \leftarrow \text{ as.matrix}(X) \\
> n \leftarrow \text{ length}(X[,1]) \\
> k \leftarrow \text{ length}(X[1,])
\]
Construct matrix $XX = X'X$

```r
> XX <- t(X) %*% X
> round(XX, 1)
```

```
   1     Area   Elevation Nearest   Scruz  Adjacent
1 30.0 7851.3 11041.0  301.80 1709.30    7832.95
```

Construct vector $Xy = X'y$

```r
> Xy <- t(X) %*% y
> Xy
```

```
   [,1]
1 2557.0
```

```
   Area   Elevation Nearest   Scruz  Adjacent
2444013.2 1976099.0  25054.6 106983.1    742829.8
```
Compute LS estimates of $\beta$

```r
> beta <- solve(XX, Xy)
> beta

[,1]
1    7.068220709
Area -0.023938338
Elevation 0.319464761
Nearest 0.009143961
Scruz -0.240524230
Adjacent -0.074804832
```

Predicted values & residual variance

```r
> predict <- X %*% beta

> residual <- y - predict

> sigma <- sqrt(sum(residual^2)/(n-k))
> sigma

[1] 60.97519
```
Standard deviation of $\beta$'s

```r
> xxi <- solve(xx)
> round(sqrt(diag(xxi))*60.975,4)
```

<table>
<thead>
<tr>
<th>Area</th>
<th>Elevation</th>
<th>Nearest</th>
<th>Scruz</th>
<th>Adjacent</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.15414</td>
<td>0.0224</td>
<td>0.0537</td>
<td>1.0541</td>
<td>0.2154</td>
</tr>
</tbody>
</table>

A user-defined function

```r
lm.reg <- function(X, y) {
  n <- length(X[,1])
  k <- length(X[1,])
  XX <- t(X) %*% X
  Xy <- t(X) %*% y
  beta <- solve(XX, Xy)
  predict <- X %*% beta
  resid <- y - predict
  sigma <- sqrt(sum(residual^2)/(n-k))
  out <- list(XX=XX, Xy=Xy, beta=beta,
               predict=predict, resid=resid, sigma=sigma)
}
```
Using this function

```r
> gala.fit<-lm.reg(X,y)
> gala.reg$beta
  [,1]
1  7.068220709
Area -0.023938338
Elevation 0.319464761
Nearest  0.009143961
Scruz  -0.240524230
Adjacent -0.074804832
> gala.reg$sigma
[1] 60.97519
```

Attentions

- This procedure is recommended for beginners in R programming
- The actual algorithm used in the generic function `lm` is very different from (and more efficient than) what is shown here.
6.4

Handling residuals that have non-constant variance and are correlated

Non-constant, correlated variance

- Residuals can have non-constant variance and are correlated.
  \[ \text{var}(e) = \Sigma = \sigma^2 \Sigma^* \]
  
  Where \( \sigma^2 \) is not known, but \( \Sigma^* \) is.

- This is saying that the correlation and relative variance between the residuals are known, but their absolute scales are not.
Generalized least squares method

- GLS estimates of $\beta$ minimize
  \[ (y - X\hat{\beta})'\Sigma^{-1}(y - X\hat{\beta}) \]

- Leading to:
  \[ \hat{\beta} = (X'\Sigma^{-1}X)^{-1}X'y \]

An alternative form

- Let $\Sigma = LL'$, where $L$ is a lower triangular matrix obtained using the Choleski decomposition.
  \[ y = X\beta + \varepsilon \quad \Rightarrow \quad L^{-1}y = L^{-1}X\beta + L^{-1}\varepsilon \]
  \[ \text{var}(L^{-1}\varepsilon) = L^{-1}\text{var}(\varepsilon) L^{-1}' = L^{-1}\sigma^2 LL'L^{-1}' = \sigma^2 I \]

- Estimates of $\beta$
  \[ \hat{\beta} = (X'L^{-1}L^{-1}X)^{-1}X'L^{-1}L^{-1}y = (X'\Sigma^{-1}X)^{-1}X'\Sigma^{-1}y \]
  \[ \text{var}(\hat{\beta}) = (X'\Sigma^{-1}X)^{-1}\sigma^2 \]
Using the `gls` function (1)

```r
> gala$id <- 1:30
> gls.fit<-gls(Species~Area+Elevation+Nearest+Scruz+Adjacent, 
+ correlation=corAR1(form=~id),data=gala)
> summary(gls.fit)
```

```
......
AIC       BIC            logLik
350.8986   360.323    -167.4493
Correlation Structure: AR(1)
Formula: ~id
Parameter estimate(s):
  Phi
  -0.3032535
```

Using the `gls` function (2)

```
Coefficients:
                Value  Std.Error  t-value  p-value
(Intercept)     5.106188  16.330163  0.312684  0.7572
Area        -0.032109    0.019156   -1.676153   0.1067
Elevation    0.347527   0.045386    7.657136   0.0000
Nearest     -0.445732   1.035401   -0.430492   0.6707
Scruz       -0.259952   0.204432   -1.271579   0.2157
Adjacent   -0.075218   0.016127   -4.664189   0.0001
```
Using the `gls` function (3)

Correlation:

<table>
<thead>
<tr>
<th></th>
<th>(Intr)</th>
<th>Area</th>
<th>Elevtn</th>
<th>Nearst</th>
<th>Scruz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>0.350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elevation</td>
<td>-0.597</td>
<td>-0.761</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nearest</td>
<td>-0.032</td>
<td>0.078</td>
<td>-0.280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scruz</td>
<td>-0.407</td>
<td>-0.026</td>
<td>0.199</td>
<td>-0.634</td>
<td></td>
</tr>
<tr>
<td>Adjacent</td>
<td>0.218</td>
<td>0.336</td>
<td>-0.590</td>
<td>0.408</td>
<td>-0.339</td>
</tr>
</tbody>
</table>

Using the `gls` function (4)

Standardized residuals:

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Q1</th>
<th>Med</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1.9050</td>
<td>-0.5301</td>
<td>-0.2582</td>
<td>0.6274</td>
<td>2.7722</td>
</tr>
</tbody>
</table>

Residual standard error: 60.58686

Degrees of freedom: 30 total; 24 residual
Auto-correlation of order 1

- Assume the residuals take a simple autoregressive form with autocorrelation given by $\rho$

\[ \mathcal{E}_{i+1} = \rho \mathcal{E}_i + \delta_i \]

where

\[ \delta_i \sim N(0, \tau^2) \]

- Then,

\[ \Sigma_{ij} = \rho^{|i-j|} \]

An step-by-step GLS

```r
# obtain the GLS estimate of beta
> X<-model.matrix(lmm.fit)
> sigma<-diag(30)
> sigma<--(-0.3032)^abs(row(sigma)-col(sigma))
> sigmai<-solve(sigma)
> XX.new<-t(X) %*% sigmai %*% X
> Xy.new<-t(X) %*% sigmai %*% gala$Species
> beta<-solve(XX.new,Xy.new)
```
GLS estimates of beta

```r
> beta

[,1]
(Intercept)  5.10635409
Area          -0.03210771
Elevation    0.34752357
Nearest     -0.44565076
Scruz        -0.25994948
Adjacent   -0.07521757
```

Standard deviations

```r
> # calculate the standard deviation of beta
> resid<-gala$Species - X %*% beta
> sigma<-sqrt(sum(resid^2)/lmm.fit$df)
> XX.inv<-solve(XX.new)
> round(sqrt(diag(XX.inv))*sigma,4)

(Intercept)  Area   Elevation  Nearest   Scruz    Adjacent
16.7127     0.0196  0.0464    1.0596    0.2092   0.0165
```