Lab 3: Linear Regression

PSTAT 131/231, Fall 2023

Learning Objectives

- A very quick review of linear regression and it's practical considerations
- Fit logistic model using `lm()` and the related functions

The lm() function

```
library(tibble)
data(state)
statedata <- data.frame(state.x77, row.names = state.abb)
?state.x77
head(statedata)
```

```
##    Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## AL       3615   3624        2.1    69.05   15.1    41.3    20  50708
## AK        365   6315        1.5    69.31   11.3    66.7   152 566432
## AZ       2212   4530        1.8    70.55    7.8    58.1    15 113417
## AR       2110   3378        1.9    70.66   10.1    39.9    65  51945
## CA      21198   5114        1.1    71.71   10.3    62.6    20 156361
## CO       2541   4884        0.7    72.06    6.8    63.9   166 103766
```

```
# Can use the . to indicate to include all the other variables in your model
lmod <- lm(Life.Exp ~ ., statedata)
summary(lmod)
```

```
##
## Call:
## lm(formula = Life.Exp ~ ., data = statedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586  < 2e-16 ***
## Population   5.180e-05  2.919e-05   1.775   0.0832 .
## Income      -2.180e-05  2.444e-04  -0.089   0.9293
## Illiteracy   3.382e-02  3.663e-01   0.092   0.9269
## Murder      -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## HS.Grad      4.893e-02  2.332e-02   2.098   0.0420 *
## Frost       -5.735e-03  3.143e-03  -1.825   0.0752 .
## Area        -7.383e-08  1.668e-06  -0.044   0.9649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7448 on 42 degrees of freedom
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10
```

Hypothesis testing

```r
n <- dim(statedata)[1] # number of observations
p <- 7 # number of predictors

round(coefficients(summary(lmod)), 5)
```

```
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 70.94322    1.74798 40.58594  0.00000
## Population   0.00005    0.00003  1.77477  0.08318
## Income      -0.00002    0.00024 -0.08921  0.92934
## Illiteracy   0.03382    0.36628  0.09233  0.92687
## Murder      -0.30112    0.04662 -6.45900  0.00000
## HS.Grad      0.04893    0.02332  2.09788  0.04197
## Frost       -0.00574    0.00314 -1.82456  0.07519
## Area         0.00000    0.00000 -0.04426  0.96491
```

Let's double check `HS.Grad` t-value and p-value

```r
# summary output t - value
coefficients(summary(lmod))[6,3] # t - value
```

```
## [1] 2.097882
```

```r
t_value <- coefficients(summary(lmod))[6,1]/coefficients(summary(lmod))[6,2]

(coefficients(summary(lmod))[6,3]) == t_value
```

```
## [1] TRUE
```

```r
pt(q = -t_value, df = n - p - 1) * 2
```

```
## [1] 0.04197175
```

```r
round(coefficients(summary(lmod)), 5)
```


```
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 70.94322    1.74798 40.58594  0.00000
## Population   0.00005    0.00003  1.77477  0.08318
## Income      -0.00002    0.00024 -0.08921  0.92934
## Illiteracy   0.03382    0.36628  0.09233  0.92687
## Murder      -0.30112    0.04662 -6.45900  0.00000
## HS.Grad      0.04893    0.02332  2.09788  0.04197
## Frost       -0.00574    0.00314 -1.82456  0.07519
## Area         0.00000    0.00000 -0.04426  0.96491
```


```r
(pt(q = -t_value, df = n - p - 1) * 2) == coefficients(summary(lmod))[6,4]
```

```
## [1] TRUE
```

R^2

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}$$

```r
y <- statedata$Life.Exp # Response values
y_hat <- fitted(lmod) # Fitted Values
e <- y - y_hat # Residuals
y_bar <- mean(y)
SST <- sum((y - y_bar)^2)

r_2 <- 1 - (sum(e^2)/SST)
r_2
```

```
## [1] 0.7361563
```

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

```
## [1] 0.7361563
```

```r
r <- cor(y_hat,y)
r^2
```

```
## [1] 0.7361563
```

Notes on R^2

- Always between 0 and 1
- Can interpret as $R^2 \times 100$ percent of the variation in Y is explained by the variation in the predictors.

Confidence Intervals

Can calculate a confidence interval by entering values into formula:

$$\hat{\beta}_j \pm (t_{n-p-1}^{\alpha/2} \boldsymbol{SE}(\hat{\beta}_j))$$

```r
std_errors <- (coef(summary(lmod))[, "Std. Error"]) # Standard errors
Beta_hats <- (coefficients(lmod)) # estimates of coefficients
t_pct <- qt(p = 0.95, df = n - p - 1) # t-statistic for a 90% CI

CI_90 <-  tibble(Beta_j = names(Beta_hats),
              lower_bound = Beta_hats - t_pct*std_errors,
              upper_bound = Beta_hats + t_pct*std_errors) # 90% CIs
CI_90
```

```
## # A tibble: 8 x 3
##   Beta_j      lower_bound upper_bound
##   <chr>             <dbl>       <dbl>
## 1 (Intercept) 68.0        73.9
## 2 Population   0.00000271  0.000101
## 3 Income      -0.000433    0.000389
## 4 Illiteracy  -0.582       0.650
## 5 Murder      -0.380      -0.223
## 6 HS.Grad      0.00970     0.0882
## 7 Frost       -0.0110     -0.000448
## 8 Area        -0.00000288  0.00000273
```

Can also use the `confint` function

```r
?confint

confint(lmod, level = .90) # 90% CIs
```

```
##                         5 %         95 %
## (Intercept)  6.800321e+01  7.388324e+01
## Population   2.709162e-06  1.008916e-04
## Income      -4.329165e-04  3.893080e-04
## Illiteracy  -5.822450e-01  6.498856e-01
## Murder      -3.795370e-01 -2.227093e-01
## HS.Grad      9.700837e-03  8.815812e-02
## Frost       -1.102176e-02 -4.482386e-04
## Area        -2.879602e-06  2.731939e-06
```

```
confint(lmod, level = 0.95) # 95% CIs
```

```
##                       2.5 %        97.5 %
## (Intercept)  6.741567e+01  7.447078e+01
## Population  -7.101457e-06  1.107022e-04
## Income      -5.150751e-04  4.714666e-04
## Illiteracy  -7.053624e-01  7.730031e-01
## Murder      -3.952076e-01 -2.070387e-01
## HS.Grad      1.861199e-03  9.599776e-02
## Frost       -1.207830e-02  6.082932e-04
## Area        -3.440321e-06  3.292657e-06
```

Prediction interval for new observations

95% Prediction Interval for new observation

```
new_data <- data.frame(Population = 1000,
                       Income = 4000,
                       Illiteracy = 1.1,
                       Murder = 10.5,
                       HS.Grad = 57.8,
                       Frost = 55,
                       Area = 83000) # new data saved as a data frame in R

predict(lmod, newdata = new_data,
        level = 0.95, interval = 'predict')
```

```
##        fit      lwr      upr
## 1 70.28979 68.65569 71.92388
```

F - Test

- Testing a Subset of Parameters Equal 0

$$F = \frac{\frac{SSR_m - SSR_M}{(df_m - df_M)}}{\frac{SSR_M}{df_M}}$$

Global F Test

```
mod_M <- lm(Life.Exp ~ ., statedata) # Larger model with all the predictors
mod_m <- lm(Life.Exp ~ 1, statedata) # Smaller model with only intercept
anova(mod_m, mod_M) # Global F - Test
```

```
## Analysis of Variance Table
##
```

```
## Model 1: Life.Exp ~ 1
## Model 2: Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad +
##     Frost + Area
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     49 88.299
## 2     42 23.297  7    65.002 16.741 2.534e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(F_value <- ((88.299 - 23.297)/7)/(23.297/42))
```

```
## [1] 16.74087
```

```r
summary(mod_M)
```

```
##
## Call:
## lm(formula = Life.Exp ~ ., data = statedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586  < 2e-16 ***
## Population   5.180e-05  2.919e-05   1.775   0.0832 .
## Income      -2.180e-05  2.444e-04  -0.089   0.9293
## Illiteracy   3.382e-02  3.663e-01   0.092   0.9269
## Murder      -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## HS.Grad      4.893e-02  2.332e-02   2.098   0.0420 *
## Frost       -5.735e-03  3.143e-03  -1.825   0.0752 .
## Area        -7.383e-08  1.668e-06  -0.044   0.9649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7448 on 42 degrees of freedom
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10
```

Partial F Test

Want to test the null hypothesis that $\beta_{HS.Grad} = \beta_{Frost} = 0$

```r
mod_M <- lm(Life.Exp ~ ., statedata) # Larger model with all the predictors
mod_m <- lm(Life.Exp ~ Population +
                       Income +
                       Illiteracy +
                       Murder +
                       Area, statedata) # smaller model without HS.Grad and Frost

anova(mod_m, mod_M)
```

```
## Analysis of Variance Table
##
## Model 1: Life.Exp ~ Population + Income + Illiteracy + Murder + Area
## Model 2: Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad +
```

```
##     Frost + Area
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     44 29.303
## 2     42 23.297  2    6.0059 5.4137 0.008095 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now lets test the null hypothesis that $\beta_{Income} = \beta_{Area} = \beta_{Illiteracy} = 0$

```r
mod_M <- lm(Life.Exp ~ ., statedata) # Larger model with all the predictors
mod_m <- lm(Life.Exp ~ Population +
                       Murder +
                       HS.Grad +
                       Frost, statedata) # smaller model without Income, Area, and Illiteracy

anova(mod_m, mod_M)
```

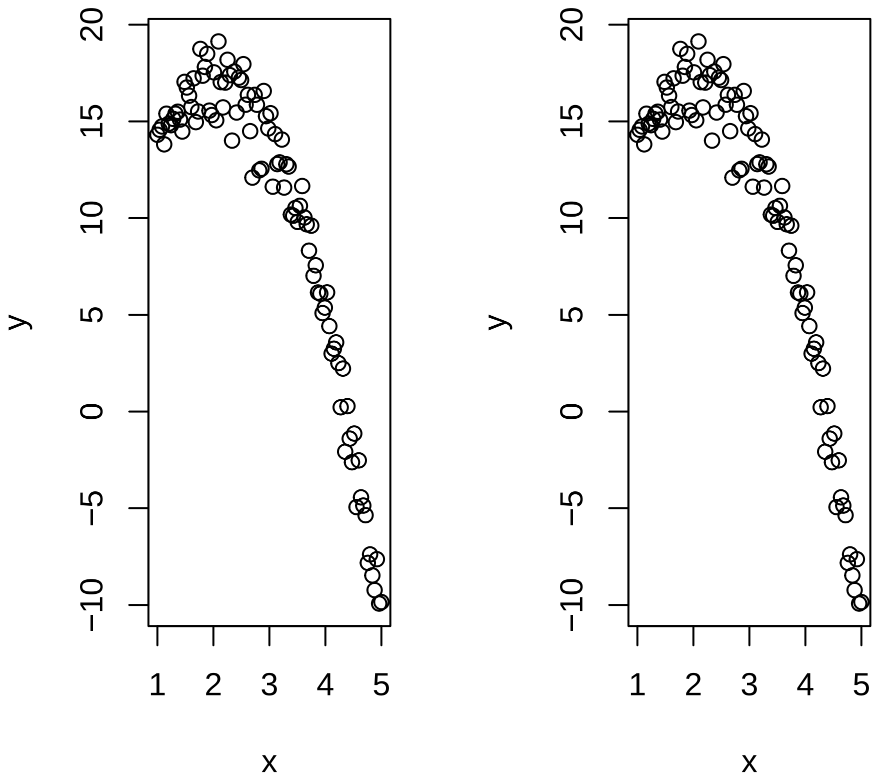
```
## Analysis of Variance Table
##
## Model 1: Life.Exp ~ Population + Murder + HS.Grad + Frost
## Model 2: Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad +
##     Frost + Area
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     45 23.308
## 2     42 23.297  3  0.010905 0.0066 0.9993
```

Polynomial Regression

Adding polynomial terms to our model with the I() function

```r
## Simulated data
n <- 100
x <-  seq(1, 5, length = n)
y <-  5 + 12 * x - 3 * x ^ 2 +
  rnorm(n, mean = 0, sd = sqrt(2))

# visualize data
par(mfrow = c(1,2))
# side note, the plot functions below do the same thing.
plot(x,y)
plot(y ~ x)
```



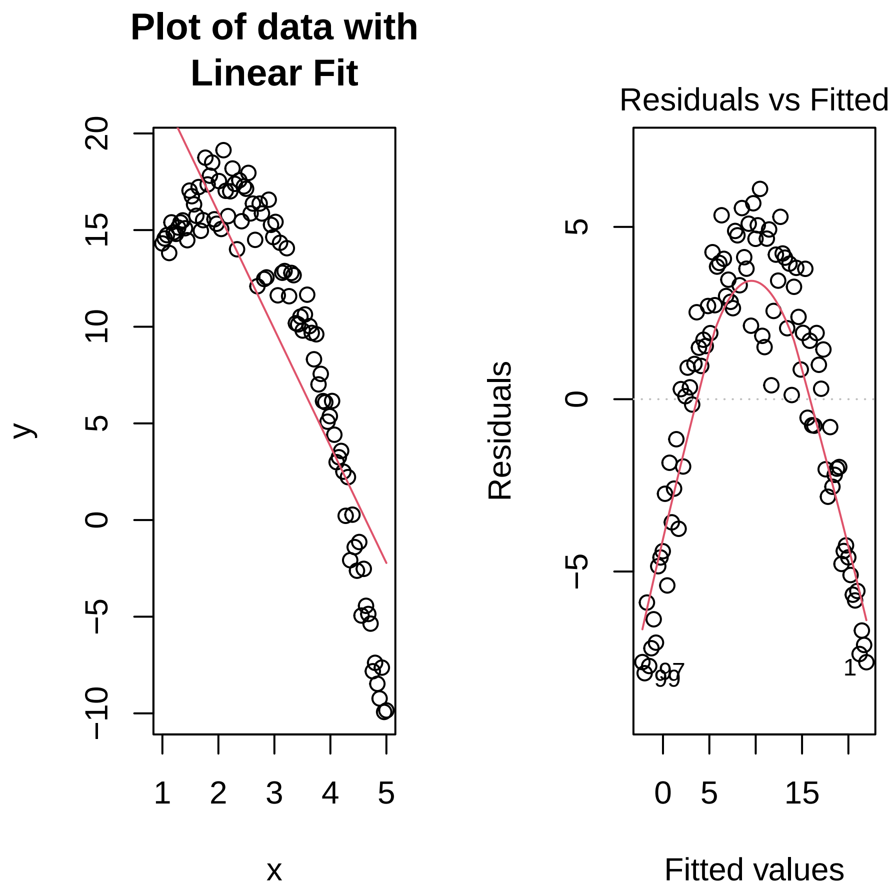

```r
# model without polynomial terms
fit <-  lm(y ~ x)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9504 -3.0155  0.9817  3.4476  6.1018
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.9807     1.1186   25.01   <2e-16 ***
## x            -6.0404     0.3475  -17.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.053 on 98 degrees of freedom
## Multiple R-squared:  0.7551, Adjusted R-squared:  0.7526
## F-statistic: 302.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
yhat <-  fitted(fit) # fitted values

plot(x, y, main = 'Plot of data with\nLinear Fit')
lines(x, yhat, col = 2)

plot(fit, which = 1)
```




```r
# model with quadratic term
fit_2 <-  lm(y ~ x + I(x ^ 2))
summary(fit_2)
```

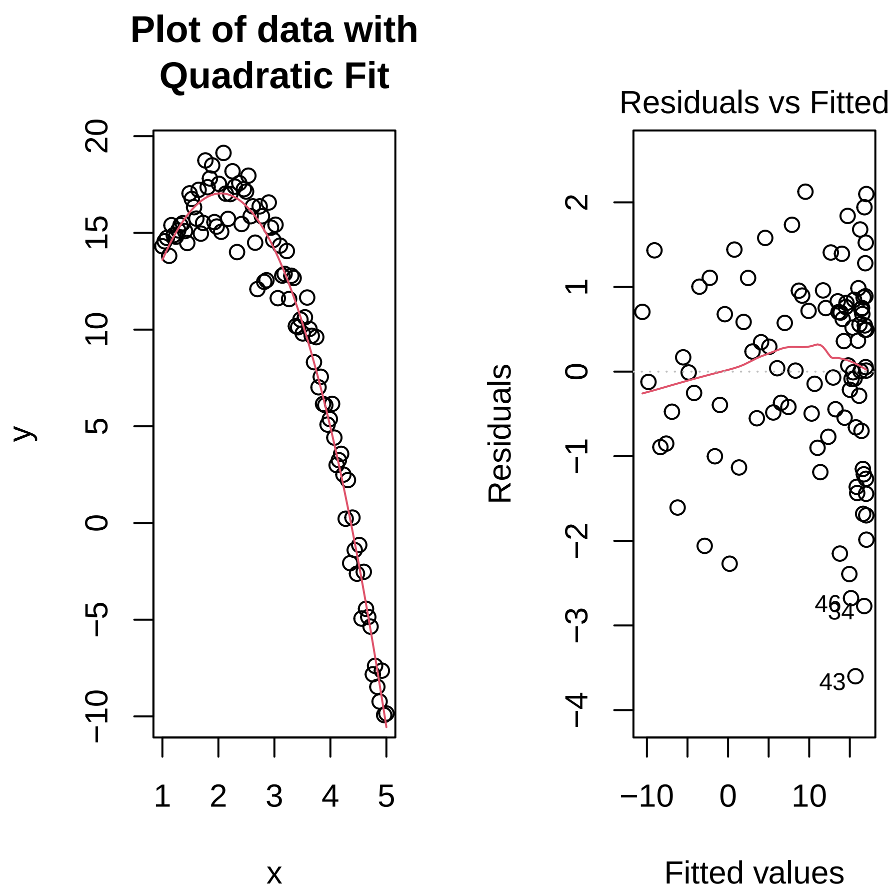
```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6001 -0.6692  0.0628  0.7751  2.1261
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.86213    0.80999   4.768 6.54e-06 ***
## x           12.90152    0.59112  21.825  < 2e-16 ***
```

```
## I(x^2)       -3.15699    0.09706 -32.525  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.181 on 97 degrees of freedom
## Multiple R-squared:  0.9794, Adjusted R-squared:  0.979
## F-statistic:  2309 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
yhat_2 <-  fitted(fit_2) # fitted values

plot(x, y, main = 'Plot of data with\nQuadratic Fit')
lines(x, yhat_2, col = 2)

plot(fit_2, which = 1)
```



Interactions

If relationship between Y and $X_1, ..., X_p$ is not additive, then can add interaction terms.

Interaction between 2 continuous variable

```r
data(state)
statedata <- data.frame(state.x77, row.names = state.abb)
```

```r
head(statedata)
```

```
##    Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## AL       3615   3624        2.1    69.05   15.1    41.3    20  50708
## AK        365   6315        1.5    69.31   11.3    66.7   152 566432
## AZ       2212   4530        1.8    70.55    7.8    58.1    15 113417
## AR       2110   3378        1.9    70.66   10.1    39.9    65  51945
## CA      21198   5114        1.1    71.71   10.3    62.6    20 156361
## CO       2541   4884        0.7    72.06    6.8    63.9   166 103766
```

```r
mod1 <- lm(Income ~ Frost + Murder + Frost:Murder,
           data = statedata)

# can also write the model like this:
mod1 <- lm(Income ~ Frost * Murder, data = statedata)
summary(mod1)
```

```
##
## Call:
## lm(formula = Income ~ Frost * Murder, data = statedata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -960.91 -405.10  -15.52  260.83 1574.23
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5579.0804   526.0379  10.606 6.06e-14 ***
## Frost          -7.9379     3.8403  -2.067  0.04439 *
## Murder       -153.8532    51.8564  -2.967  0.00476 **
## Frost:Murder    1.2266     0.4288   2.861  0.00634 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 564.2 on 46 degrees of freedom
## Multiple R-squared:  0.2085, Adjusted R-squared:  0.1569
## F-statistic: 4.039 on 3 and 46 DF,  p-value: 0.01245
```

p-value is less than a threshold value of $\alpha = 0.05$, thus there is significant evidence that the interaction term between `Frost` and `Murder` is a significant predictor of `Income`.

Qualitative Predictor

```r
library(faraway)
head(teengamb)
```

```
##   sex status income verbal gamble
## 1   1     51   2.00      8    0.0
## 2   1     28   2.50      8    0.0
## 3   1     37   2.00      6    0.0
## 4   1     28   7.00      4    7.3
## 5   1     65   2.00      8   19.6
## 6   1     61   3.47      6    0.1
```

This *teengamb* dataset is a survey about teenage gambling in Britain. The *sex* is 0 for male and 1 for female.

The *status* is socioeconomic status score based on parents' occupation, *income* is income in pounds per week, *verbal* is verbal score in words out of 12 correctly difined, and *gamble* is expenditure on gambling in pounds per year. In this dataset, *sex* is qualitative, and the rest are quantitative.
Now we use *gamble* as the response and the rest as predictors to fit a MLR model to the data:

```
mod=lm(gamble~factor(sex)+status+income+verbal,data=teengamb)
summary(mod)
```

```
##
## Call:
## lm(formula = gamble ~ factor(sex) + status + income + verbal,
##     data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.55565   17.19680   1.312   0.1968
## factor(sex)1 -22.11833    8.21111  -2.694   0.0101 *
## status         0.05223    0.28111   0.186   0.8535
## income         4.96198    1.02539   4.839 1.79e-05 ***
## verbal        -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

```
contrasts(factor(teengamb$sex))
```

```
##   1
## 0 0
## 1 1
```

Here, the factor level 0 (male) for sex is the baseline level. The regression coefficient of $factor(sex)1$ is -22.11833, which should be interpreted like holding all other predictors fixed, on average one female spends 22.11833 pounds per year less than one male on gambling.
Next we fit a model to predict *gamble* using *sex* and *income* as well as an interaction term between them.

```
mod2=lm(gamble~factor(sex)*income,teengamb)
summary(mod2)
```

```
##
## Call:
## lm(formula = gamble ~ factor(sex) * income, data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.522  -4.860  -1.790   6.273  93.478
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          -2.6596     6.3164  -0.421  0.67580
## factor(sex)1          5.7996    11.2003   0.518  0.60724
## income                6.5181     0.9881   6.597 4.95e-08 ***
## factor(sex)1:income  -6.3432     2.1446  -2.958  0.00502 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.98 on 43 degrees of freedom
## Multiple R-squared:  0.5857, Adjusted R-squared:  0.5568
## F-statistic: 20.26 on 3 and 43 DF,  p-value: 2.451e-08
```

Our fitted model is now:

$$\hat{gamble} = -2.6596 + 5.7996 * I(sex = 1(female)) + 6.5181 * income - 6.3432 * income * I(sex = 1(female))$$

And in this model, for a female, if *income* increases by 1 unit, then the *gamble* will increase by $6.5181 - 6.3432 = 0.1749$.

Checking Error Assumptions

Constant Variance (Homoscedasticity)

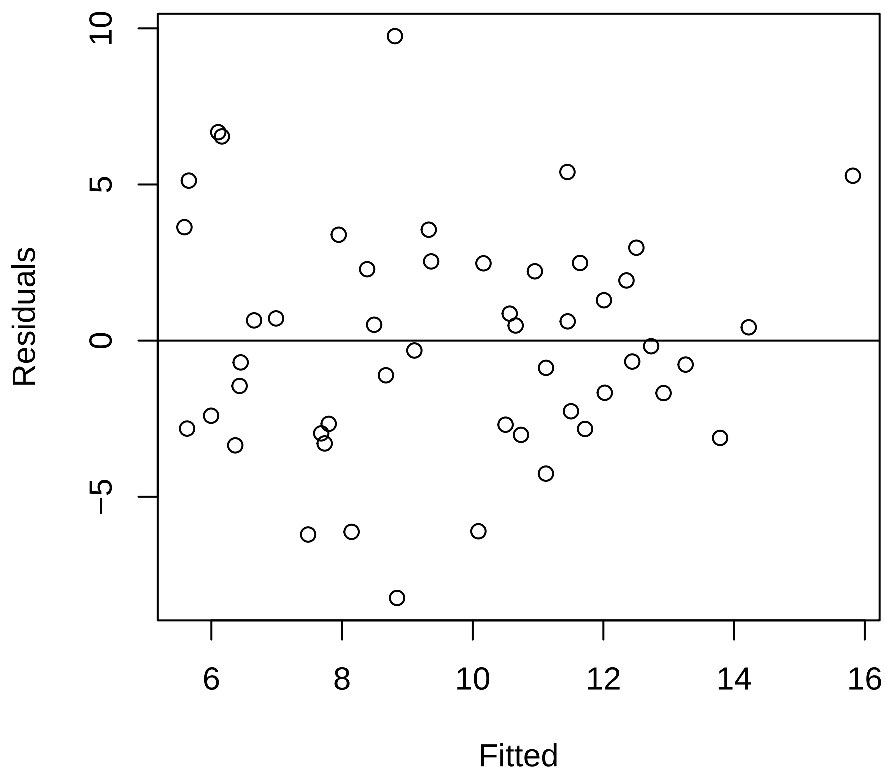
If everything is well, we should see constant symmetrical variation. Nonconstant variance (heteroscedasticity) or nonlinear pattern indicates that the constant variance assumption is questionable.

```r
library(faraway)
head(savings)
```

```
##              sr pop15 pop75     dpi ddpi
## Australia 11.43 29.35  2.87 2329.68 2.87
## Austria   12.07 23.32  4.41 1507.99 3.93
## Belgium   13.17 23.80  4.43 2108.47 3.82
## Bolivia    5.75 41.89  1.67  189.13 0.22
## Brazil    12.88 42.19  0.83  728.47 4.56
## Canada     8.79 31.72  2.85 2982.88 2.43
```

In this dataset, *sr* is the saving rate (personal saving divided by disposable income), $pop15$ is the percent population under age of 15, $pop75$ is the percent population over age of 75, *dpi* is the per-capita disposable income in dollars, *ddpi* is the percent growth rate of *dpi*.

```r
lmod=lm(sr~pop15+pop75+dpi+ddpi, data=savings)
plot(fitted(lmod),residuals(lmod),xlab='Fitted',ylab='Residuals')
abline(h=0)
```

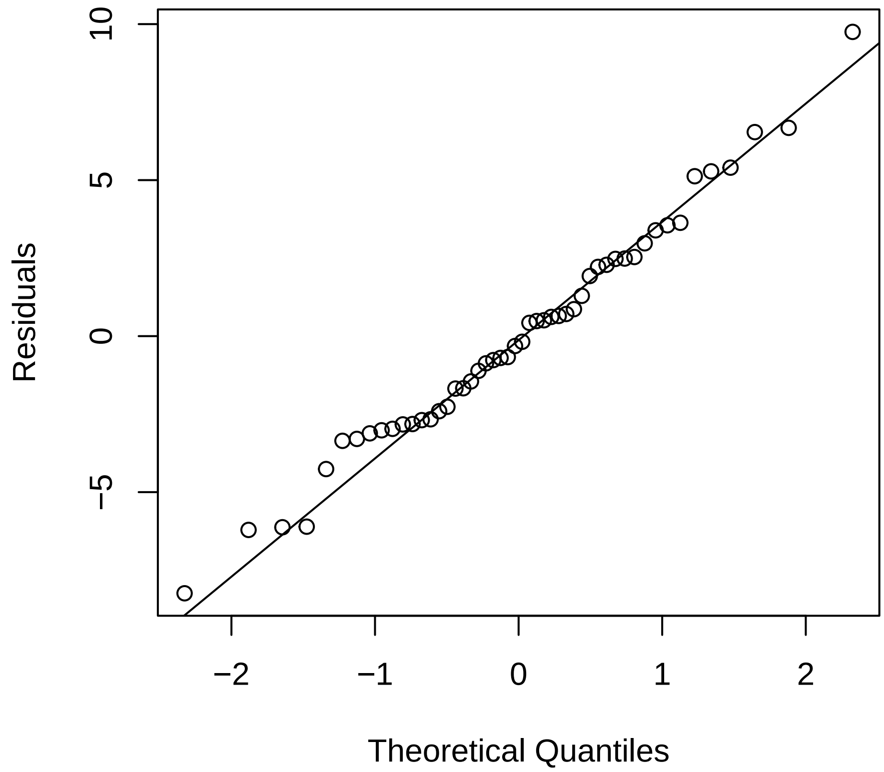


Everything seems alright in this plot. From this plot, no special pattern occurs and we can confirm that the constant variance assumption is satisfied.

Normality

We can use QQ plot or Shapiro-Wilk test to check normality.

```
qqnorm(residuals(lmod),ylab='Residuals',main='')
qqline(residuals(lmod))
```



Normal residuals should follow the line approximately. Here the residuals look normal.
Or we may use the Shapiro-Wilk test. Shapiro-Wilk test is a formal test for normality.

```
shapiro.test(residuals(lmod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmod)
## W = 0.98698, p-value = 0.8524
```

The null hypothesis is that residuals are normal. Since the p-value is very large in this case, we fail to reject the null hypothesis and conclude that the residuals are normal.

Correlated Errors

In general, it's difficult to check for correlated errors since there are just too many possible patterns of correlations that may occur. We do not have enough information to make any reasonable check. But some types of data have a structure which suggests where to look for problems. It's a temporal data. For the residuals versus year plot, if the errors were uncorrelated, we should expect to see a random scatter of points above and below the zero line. However in that plot, we see long sequence of points above(blue box) or below(red box) the line. This is an indication of positive serial correlation. For the successive pairs of residuals plot, we can see a positive correlation, which indicates positive serial correlation.

Finding Unusual Observations

Leverage

A high-leverage point is extreme in the predictor space. It has the potential to influence the fit, but does not necessarily do so. It is important to first identify such points. Deciding what to do about them can be difficult.

Recall that H_{ii} is the leverage of x_i. And $\sum_i H_{ii} = p + 1$, this can be easily proved using linear algebra knowledge. So the average value for leverage is $\frac{p+1}{n}$. A rough rule is that leverages with more than $2 - 3$ times of $\frac{p+1}{n}$.

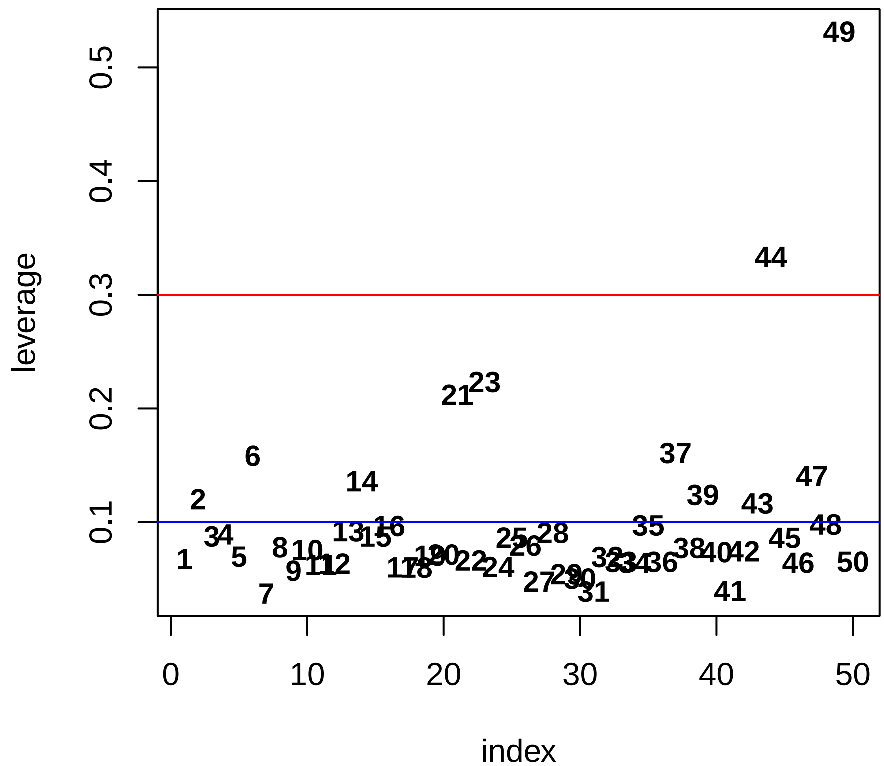
```
lev=hatvalues(lmod)
head(lev)
```

```
##  Australia    Austria    Belgium    Bolivia     Brazil     Canada
## 0.06771343 0.12038393 0.08748248 0.08947114 0.06955944 0.15840239
```

```
sum(lev)==4+1
```

```
## [1] TRUE
```

```
n=nrow(savings)
p=4
dat=data.frame(index=seq(n),leverage=lev)
plot(leverage~index,col="white",data=dat,pch=NULL)
text(leverage~index,labels = index,data=dat,cex=0.9,font=2)
abline(h=(p+1)/n,col ="blue")
abline(h=3*(p+1)/n,col="red")
```



We can see from this plot that two observations(with index number 44 and 49) are potentially high leverage observations.

Outliers

An outlier is a point that does not fit the current model well. Here we consider the standardized residuals

$$r_i = \frac{y_i - \hat{y}_i}{\hat{\sigma}\sqrt{1 - H_{ii}}}$$

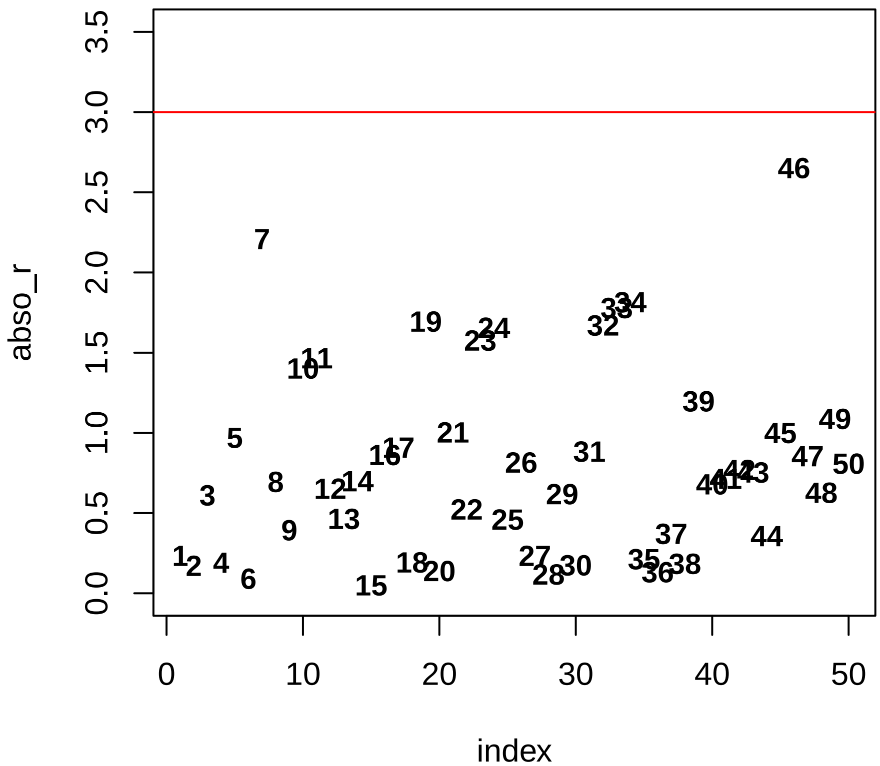
The rule of thumb is that observations with absolute value of standardized residuals greater than or equal to 3 are considered as outliers.

```
r=rstandard(lmod)
which(abs(r)>=3)
```

```
## named integer(0)
```

In this case, no outliers. Or we may also use the plot to check:

```
dat2=data.frame(index=seq(length(r)),abso_r=abs(r))
plot(abso_r~index,col="white",data=dat2,pch=NULL,ylim=c(0,3.5))
text(abso_r~index,labels = index,data=dat2,cex=0.9,font=2)
abline(h=3,col="red")
```



Again we see no points with absolute value of standardized residuals greater than or equal to 3. Our conclusion remains the same that there is no outlier.

Influential Observations

An influential point is one whose removal from the dataset would cause a large change in the fit. An influential point may or may not be an outlier and may or may not have large leverage, but it will tend to have at least one of these two properties. We usually use Cook's distance.

$$D_i = \frac{1}{p+1} r_i^2 \frac{H_{ii}}{1-H_{ii}}$$

One rule of thumb is that observations with Cook's distance greater than $4/n$ is influential.

```
d=cooks.distance(lmod)
which(d>4/n)
```

```
##   Japan Zambia  Libya
##      23     46     49
```