STA 108 Discussion 5: Diagnostic

Yuanyuan Li

Reference: Textbook Chapter 3.

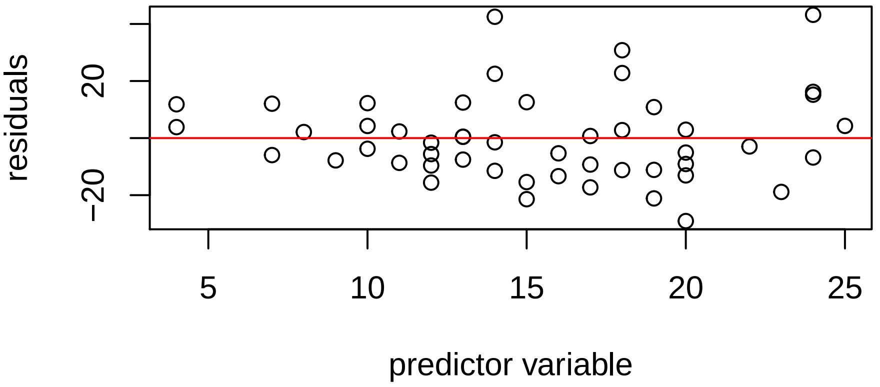
The goal of diagnostics is to examine the departures from the simple linear regression model with normal errors. Typical departures and corresponding diagnostic plots/tests:

- The regression line is not **linear** —— residual plots(residual against the predictor variable, or against the fitted values), lack-of-fit test.
- The error terms are not **normally** distributed —— histogram, boxplot/dot plot of residuals, normal probability plot (a.k.a. QQ plot), Shapiro-Wilk's test,correlation test for normality, etc.
- The error terms do not have **constant variance** —— residual plots, B-F test.
- The error terms are not **independent** —— residual against time, e.g.,P109.
- The model fits all but one or a few **outlier** observations. —— (semistudentized) residual plots, box plots, dot plots, stem-and-leaf plots, etc.
- Some important **predictors are missing** —— residual plots (residual against other possibly important predictors)

1. Residual plots

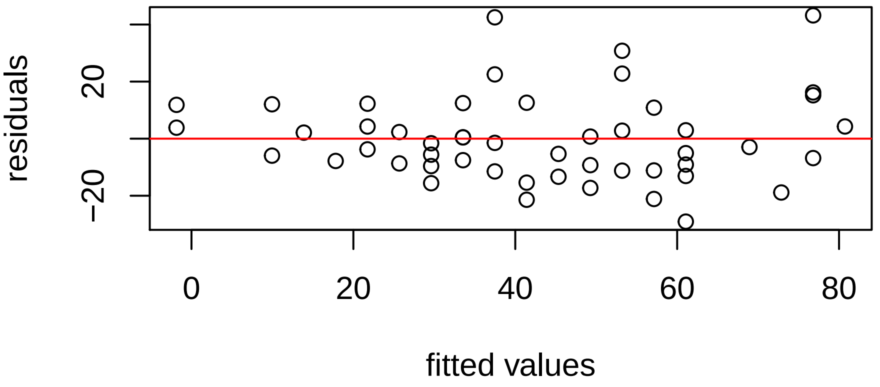
1.1 Against the predictor variable X

```r
fit = lm(dist~speed, data=cars)
residuals = fit$residuals#or Y-yhat
plot(x=cars$speed, y=residuals, xlab='predictor variable', ylab='residuals')
abline(h=0, col='red')
```



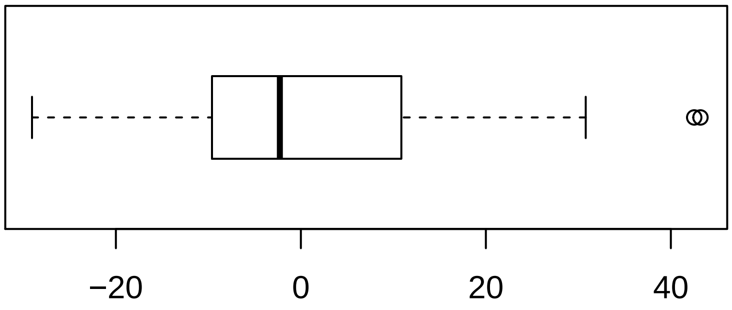
1.2 Equivalently, against the fitted values $\hat{Y}$

```
y_hat = fit$fitted.values
plot(x=y_hat, y=residuals, xlab='fitted values', ylab='residuals')
abline(h=0, col='red')
```



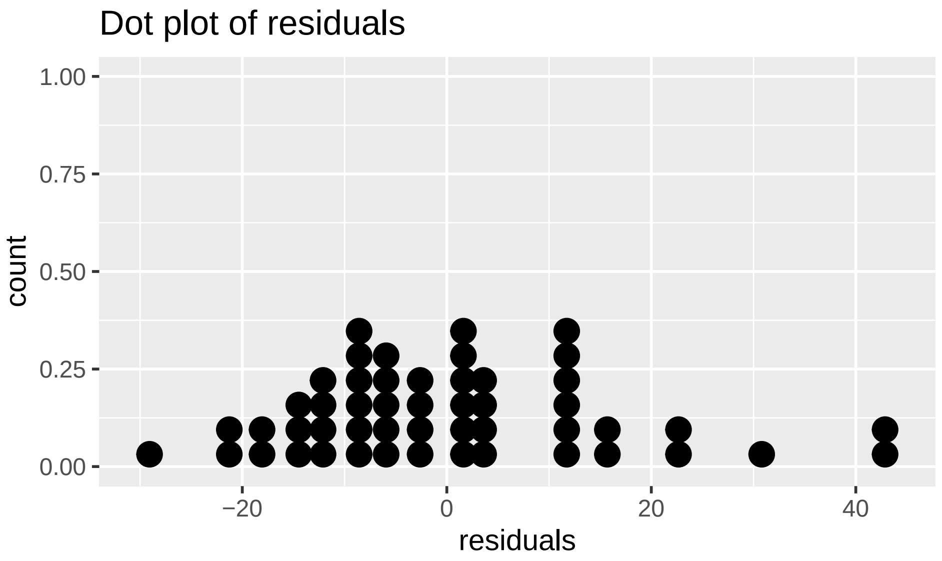
1.3 Box plot of residuals

```
boxplot(residuals,horizontal =T)
```



1.4 Dot plot

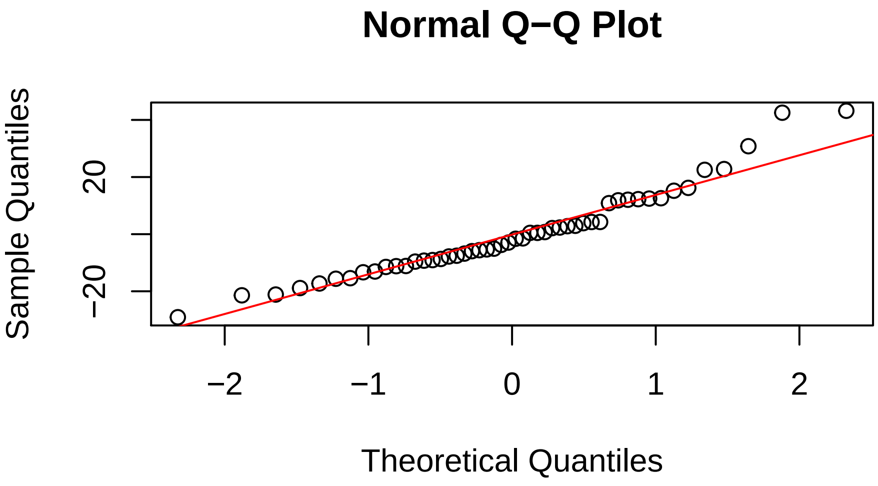
```
library(ggplot2)#install.packages("ggplot2") to install it
ggplot() + aes(residuals) + geom_dotplot() + xlab("residuals") + ggtitle("Dot plot of residuals")
```



1.5 Normal probability plot of residuals

Here each residual is plotted against its expected value under normality. A plot that is nearly linear suggests agreement with normality, whereas a plot that departs substantially from linearity suggests that the error distribution is not normal.

```
qqnorm(residuals)
qqline(residuals, col='red') # theoretical line
```



2. Diagnostic Tests

2.1 Test for normality

Shapiro-Wilk's test

H_0: The data is normally distributed. v.s. H_a: The distribution is non-normal.

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

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.94509, p-value = 0.02152
```

Correlation test for normality

The test is conducted by calculating the coefficient of correlation r between the residuals e_i and their expected values under normality. A high value of the correlation coefficient is indicative of normality.

H_0: The data is normally distributed (ρ=1). v.s. H_a: The distribution is non-normal ($\rho < 1$).

```
res <- qqnorm(residuals, plot.it = FALSE)
#correlation between observed residuals and expected residuals
r=cor(res$x, res$y)#res$x is expected residuals, res$y is observed residuals
r
```

```
## [1] 0.9718331
```

```
n=length(residuals)
n
```

```
## [1] 50
```

Given $\alpha = 0.05$, from Table B.6 that the critical value for $n = 50$ is .977. Since $r = 0.972 < 0.977$, we conclude H_a that the data do not have a normal distribution.

2.2 B-F Test for constant variance

1. Divide the data into two parts according to residual pattern or just using the mean of X.
2. Compute $d_{i1} = |\hat{\epsilon}_{i1} - m(\hat{\epsilon}_1)|$, $d_{i2} = |\hat{\epsilon}_{i2} - m(\hat{\epsilon}_2)|$, where $m(\hat{\epsilon}_1)$, $m(\hat{\epsilon}_2)$ are the medians for the two groups.
3. Compute test statistic

$$t_{BF} = \frac{\bar{d}_1 - \bar{d}_2}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

   where $\bar{d}_1 = n_1^{-1} \sum_{i=1}^{n_1} d_{i1}$, and

$$s^2 = \frac{\sum_{i=1}^{n_1}(d_{i1} - \bar{d}_1)^2 + \sum_{i=1}^{n_2}(d_{i2} - \bar{d}_2)^2}{n-2}.$$

4. Test $H_0 : \sigma_1^2 = \sigma_2^2$ v.s. $H_a : \sigma_1^2 \neq \sigma_2^2$.Since $t_{BF} \sim t_{n-2}$ under H_0. Given α, reject H_0 when $|t_{BF}| > t_{n-2;1-\alpha/2}$.

```
group1 <- cars$speed < 15#Divide the data into two parts according to residual pattern
group2 <- !group1
d1 <- abs(residuals[group1] - median(residuals[group1]))
d2 <- abs(residuals[group2] - median(residuals[group2]))
n <- length(residuals)
n1 <- length(d1)
n2 <- length(d2)
s <- sqrt((sum((d1-mean(d1))^2)+sum((d2-mean(d2))^2))/(n - 2))
t.stat <- (mean(d1)-mean(d2))/s/sqrt(1/n1+1/n2)
t.stat
```

```
## [1] -1.383369
```

```
alpha=0.01
t.crit <- qt(1-alpha/2, df = n-2)
t.crit
```

```
## [1] 2.682204
```

Since $|t| = 1.38 < 2.68$, we fail to reject H_0, so the residuals have constant variance.

2.3 F test for lack-of-fit

Recall that simple linear regression model(2.1 or 1.24) can be written as

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i,\ i = 1, \cdots, n.$$

If we denote the different X levels in the data as $X_1, \cdots, X_c$(or X_j, $j = 1, \cdots, c$), and the number of replicates for the j-th level of X as n_j. The total number of observations

$$n = \sum_{j=1}^{c} n_j.$$

Then the observed value of the response variable for the i-th replicate for the j-th level of X can be denoted as Y_{ij}, where$j = 1, \cdots, c$,$i = 1, \cdots, n_j$. The model 2.1 can also be expressed as

$$\textbf{Reduced model:}\ \ Y_{ij} = \beta_0 + \beta_1 X_j + \varepsilon_{ij}.$$

This model is very simple, because the mean function $E(Y_{ij}) = \beta_0 + \beta_1 X_j$ only involves two parameters: β_0, β_1. Let us drop this linear relation assumption, and denote the mean function as $E(Y_{ij}) = \mu_j$, $j = 1, \cdots, c$, which involves c parameters. We get

$$\textbf{Full model:}\quad Y_{ij} = \mu_j + \varepsilon_{ij}.$$

The lack-of-fit test is testing the appropriateness of a linear regression relation, the hypotheses are

$$H_0 : E(Y) = \beta_0 + \beta_1 X, \quad v.s.\ \ H_a : E(Y) \neq \beta_0 + \beta_1 X.$$

Under H_0, we get the reduced model above. While under H_a, we get above full model. The test statistic is

$$F^* = \frac{(SSE_R - SSE_F)/(df_R - df_F)}{SSE_F/df_F}.$$

The p-value for this test is $P(F_{df_R-df_F,\ df_F} > F^*)$. The critical value for a given α is $F_{1-\alpha;\ df_R-df_F,\ df_F}$.

The result of anova function gives the ANOVA table. The first row and second row corresponds to the reduced model and the full model, repectively. The first column "Res.Df" shows the degrees of freedom associated with SSE in the two models (i.e. $df_R = n - 2$ and $df_F = n - c$). The second column "RSS" shows the sum of squared error of the two models (i.e. SSE_R and SSE_F).The third column is the difference between the degree of freedoms($df_R - df_F = c - 2$).The fourth column is $SSE_R - SSE_F$. Next are F statistics and p-value.

```r
reduced = lm(dist ~ speed, data = cars)
full = lm(dist ~ 0 + as.factor(speed), data = cars)
knitr::kable(anova(reduced, full))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
48	11353.521				
31	6764.783	17	4588.738	1.23695	0.2948374

```r
nlevels(factor(cars$speed))#c
```

```
## [1] 19
```

```r
n#n
```

```
## [1] 50
```

Since p-value is larger than 0.05 (given $\alpha = 0.05$), we fail to reject H_0, the regression function is linear.

3. Remedial Measures

3.1 Transformation of X: For nonlinear association

Check Figure 1.

3.2 Transformation of Y : For nonnormality/unequal variance

Box-Cox Transformation

```r
#Defince a function that calculate box-cox transformation; x,y are the variables.
box_trans = function(lambda,x,y){
  n = length(y)
  K2= prod(y)^(1/n)
  K1 = 1/(lambda* K2^(lambda - 1))
  if (lambda != 0){
    W = K1*(y^lambda-1)
  } else{
    W = K2*log(y)
  }
```

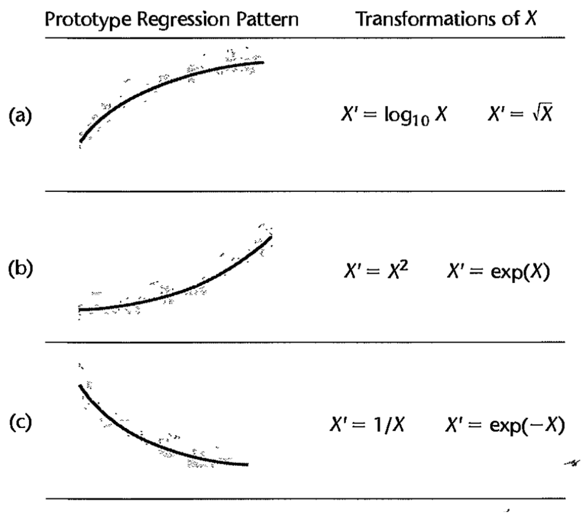


Figure 1: Transformation of X

```
  SSE = anova(lm(W~x))[2,2]
  return(SSE)
}

lambda = seq(-1, 3, by=1)
SSE = sapply(lambda, function(l)box_trans(l, cars$speed,cars$dist))
knitr::kable(t(data.frame(lambda, SSE)))#kabel function is in knitr package
```

lambda	-1.0	0.0	1.00	2.0	3.0
SSE	266474.4	11266.9	11353.52	43298.7	231743.2

Find the λ that can minimize SSE, then use the regression model

$$Y_i^\lambda = \beta_0 + \beta_1 X_i + \varepsilon_i.$$

Specially, if choose $\lambda = 0$, $Y_i^\lambda = \log(Y_i)$.

The algorithm above only works for positive response data. If the response variable includes some non-positive values, we can add a constant to make it positive. For example, assume the minimum value in the response variable is -2, we can add 2.5 to all responses, then run Box-cox transformation for the positive response.

```
SSE = sapply(lambda, function(l)box_trans(l, cars$speed,cars$dist+2.5))
knitr::kable(t(data.frame(lambda, SSE)))#kabel function is in knitr package
```

lambda	-1.0	0.000	1.00	2.00	3.0
SSE	72566.7	9362.519	11353.52	38155.36	177173.7

Find the λ that can minimize SSE, then use the regression model

$$(Y_i + 2.5)^\lambda = \beta_0 + \beta_1 X_i + \varepsilon_i.$$