✪ Psychological Statistics ✪

Week 09: *Linear Regression*

- Edited by Prof. **Changwei Wu**
- Graduate Institute of Mind, Brain and Consciousness (***GIMBC***), Taipei Medical University

In [ ]:
```
### [ Setup the working directory ]

setwd("/Users/wesley/[Course]/Python/R_Script")
getwd()
```

In [75]:
```
### [ Loading the required libraries ]

library("dplyr")
library("rstatix")
library("ggplot2")
library("car")
library("performance")
library("QuantPsyc")
library("boot")
```

---

(1) Linear Regression

[Hypothesis] Rat growth can be predicted by dietary tannin concentration. *(2-tailed)*

- Null hypothesis $\boldsymbol{H_0}$ **:** No linear relationship between rat growth and tannin concentration $\rightarrow \boldsymbol{\beta} = 0$
- Alternative hyp. $\boldsymbol{H_1}$ **:** There is (linear) relationship between rat growth and tannin concentration $\rightarrow \boldsymbol{\beta} \neq 0$

In [89]:
```
### [ Step.1 ] Load data

reg <- read.csv("tannin.csv", header = TRUE)
attach(reg)

reg %>% get_summary_stats(type = "mean_sd")
```

```
The following objects are masked from reg (pos = 3):

    growth, tannin
```

A tibble: 2 × 4

variable	n	mean	sd
<chr>	<dbl>	<dbl>	<dbl>
growth	9	6.889	3.689
tannin	9	4.000	2.739

In [15]:
```
### [ Step.2 ] Check assumptions

#----- Check the assumptions of regression after regression analysis -----#
```

In [90]:
```
### [ Step.3 (1)] Regression analysis: lm in Regression table

model <- lm(growth ~ tannin)
summary(model)
```

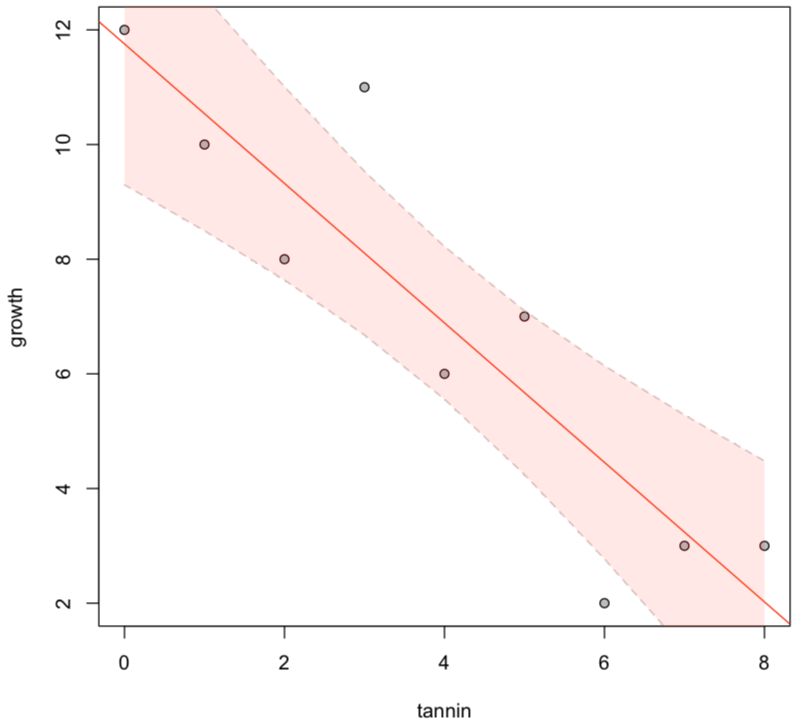
```
Call:
lm(formula = growth ~ tannin)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4556 -0.8889 -0.2389  0.9778  2.8944 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  11.7556     1.0408  11.295 9.54e-06 ***
tannin       -1.2167     0.2186  -5.565 0.000846 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.693 on 7 degrees of freedom
Multiple R-squared:  0.8157,	Adjusted R-squared:  0.7893 
F-statistic: 30.97 on 1 and 7 DF,  p-value: 0.0008461
```

In [23]:
```
### [ Step.3 (2) ] Regression analysis: lm in ANOVA table

summary.aov(model)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
tannin       1  88.82   88.82   30.97 0.000846 ***
Residuals    7  20.07    2.87                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In [18]:
```
### [ Step.4 ] Adjusted R-squared

# ----- This has been shown in the Regression table -----#
```

In [21]:
```
### [ Step.5 (1) ] Visualization

par(mfrow=c(1,1))
plot(tannin,growth,pch=21,bg="gray")
abline(model,col="red")
lines(range, prdt[,2], col="gray80", lty=2)
lines(range, prdt[,3], col="gray80", lty=2)
polygon(c(rev(range), range), c(rev(prdt[ ,3]), prdt[ ,2]), col=rgb(1,0,0,0.1),
```



In [20]:

```
### [ Step.5 (2) ] Prediction through Regression model

#--- predict confidence interval by prediction----
confint(model)
range <- seq(min(reg$tannin), max(reg$tannin))
prdt <- predict(model, data.frame(x=range), interval='confidence')
prdt
```

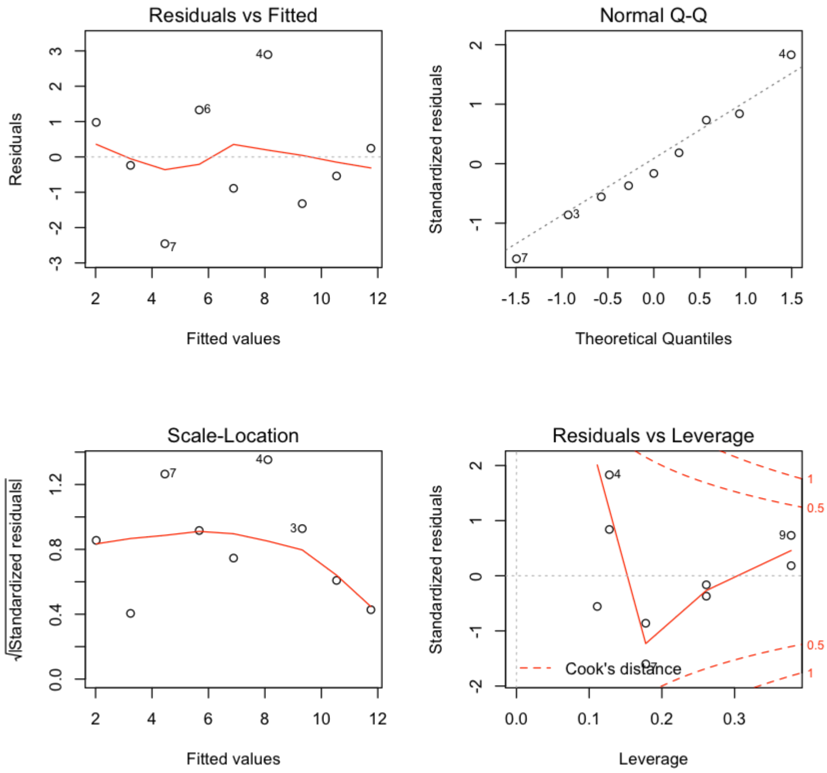
A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	9.294457	14.2166544
tannin	-1.733601	-0.6997325

A matrix: 9 × 3 of type dbl

	fit	lwr	upr
1	11.755556	9.2944567	14.216654
2	10.538889	8.4928046	12.584973
3	9.322222	7.6339224	11.010522
4	8.105556	6.6742297	9.536881
5	6.888889	5.5541707	8.223607
6	5.672222	4.2408964	7.103548
7	4.455556	2.7672557	6.143855
8	3.238889	1.1928046	5.284973
9	2.022222	-0.4388766	4.483321

In [92]:
```
### [ Step.2(1) ] Check assumptions after regression analysis

# → Visual inspection of assumptions: plot(model)
par(mfrow=c(2,2))
plot(model)
```



In [93]:
```
### [ Step.2(2) ] Check assumptions after regression analysis

# → Identify outliers: influence.measures(model)
influence.measures(model)
```

```
Influence measures of
         lm(formula = growth ~ tannin) :

   dfb.1_  dfb.tnnn   dffit cov.r  cook.d   hat inf
1  0.1323 -1.11e-01  0.1323 2.167 0.01017 0.378   *
2 -0.2038  1.56e-01 -0.2058 1.771 0.02422 0.261
3 -0.3698  2.40e-01 -0.3921 1.323 0.08016 0.178
4  0.7267 -3.24e-01  0.8981 0.424 0.24536 0.128
5 -0.1011 -1.55e-17 -0.1864 1.399 0.01937 0.111
6  0.0635  1.13e-01  0.3137 1.262 0.05163 0.128
7  0.0741 -5.29e-01 -0.8642 0.667 0.27648 0.178
8  0.0256 -6.86e-02 -0.0905 1.828 0.00476 0.261
9 -0.2263  4.62e-01  0.5495 1.865 0.16267 0.378   *
```

~ *Report* ~

- **The rat growth was negatively associated with tannin concentration ($p < 0.001$).**
- The regression model is y = -1.22*x+11.76, which explained 79% of the total variation in rat growth.

---

(2) Multiple Regression ¶

[Hypothesis]* Album sales can be affected by 3 factors: advert, airplay and attractiveness.** ***(2-tailed)

- Null hypothesis H_0 **:** No prominent relations can be observed among album sales and the 3 factors (all βs = 0).
- Alternative hyp. H_1 **:** The album sales can be predicted by at least one of the 3 factors (βs ≠ 0).

In [4]:
```
### [ Step.1 ] Data Loading

# → Loading the dataset
album <- read.delim("AlbumSales.dat", header = TRUE)
album %>% head(4)
```

A data.frame: 4 × 4

	adverts	sales	airplay	attract
	<dbl>	<int>	<int>	<int>
1	10.256	330	43	10
2	985.685	120	28	7
3	1445.563	360	35	7
4	1188.193	270	33	7

In [28]:
```
### [ Step.2 ] Check assumptions

#----- Check the assumptions of regression after regression analysis -----#
```

In [9]:
```
### [ Step.3 ] Analyses of multiple regression models

#--- Regressor (1): adverts ---#
albumSales.1<-lm(sales ~ adverts, data = album)
summary(albumSales.1)
# confint(albumSales.1)
```

```
Call:
lm(formula = sales ~ adverts, data = album)

Residuals:
     Min       1Q   Median       3Q      Max 
-152.949  -43.796   -0.393   37.040  211.866 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.341e+02  7.537e+00  17.799   <2e-16 ***
adverts     9.612e-02  9.632e-03   9.979   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 65.99 on 198 degrees of freedom
Multiple R-squared:  0.3346,	Adjusted R-squared:  0.3313 
F-statistic: 99.59 on 1 and 198 DF,  p-value: < 2.2e-16
```

In [8]:
```
### [ Step.3 ] Analyses of multiple regression models

#--- Regressor (2): adverts + attractiveness ---#
#albumSales.2<-lm(sales ~ adverts + attract, data = album)
# summary(albumSales.2)
# confint(albumSales.2)

#--- Regressor (3): adverts + attractiveness + airplay ---#
albumSales.3<-lm(sales ~ adverts + attract + airplay, data = album)
summary(albumSales.3)
# confint(albumSales.3)
```

```
Call:
lm(formula = sales ~ adverts + attract + airplay, data = album)

Residuals:
     Min       1Q   Median       3Q      Max
-121.324  -28.336   -0.451   28.967  144.132

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -26.612958  17.350001  -1.534    0.127
adverts       0.084885   0.006923  12.261  < 2e-16 ***
attract      11.086335   2.437849   4.548 9.49e-06 ***
airplay       3.367425   0.277771  12.123  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.09 on 196 degrees of freedom
Multiple R-squared:  0.6647,    Adjusted R-squared:  0.6595
F-statistic: 129.5 on 3 and 196 DF,  p-value: < 2.2e-16
```

In [6]:
```
### [ Step.4 ] Effect size: R square (coefficient of determination)

#---- To compare the residual between models (simple vs. multiple regression)

AIC(albumSales.1, albumSales.3)
anova(albumSales.1, albumSales.3)
```

A data.frame: 2 × 2

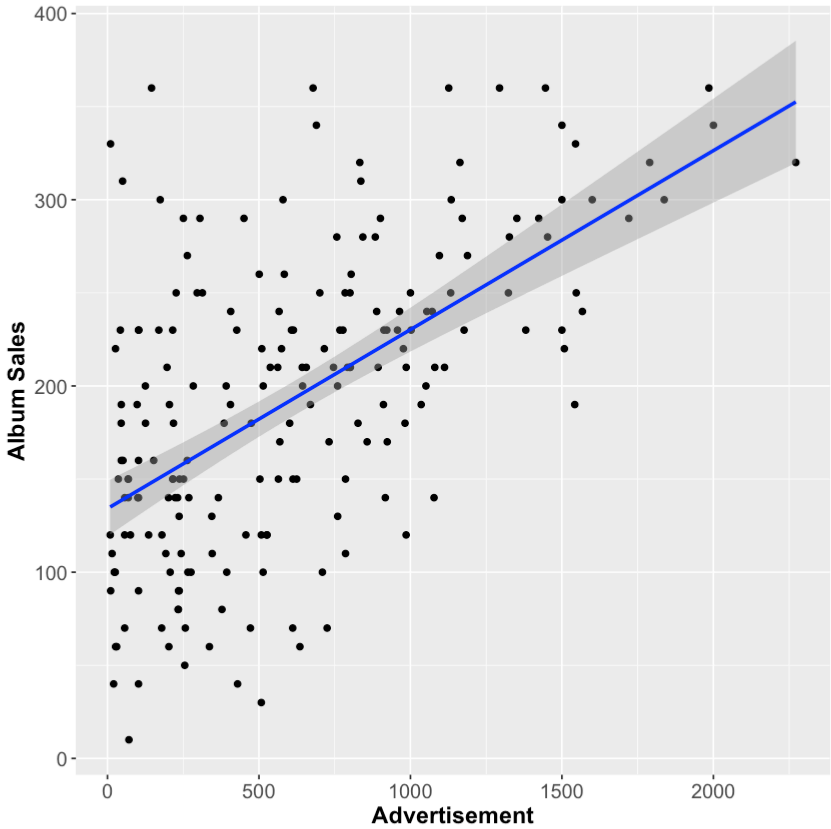
	df	AIC
	<dbl>	<dbl>
albumSales.1	3	2247.375
albumSales.3	5	2114.337

A anova: 2 × 6

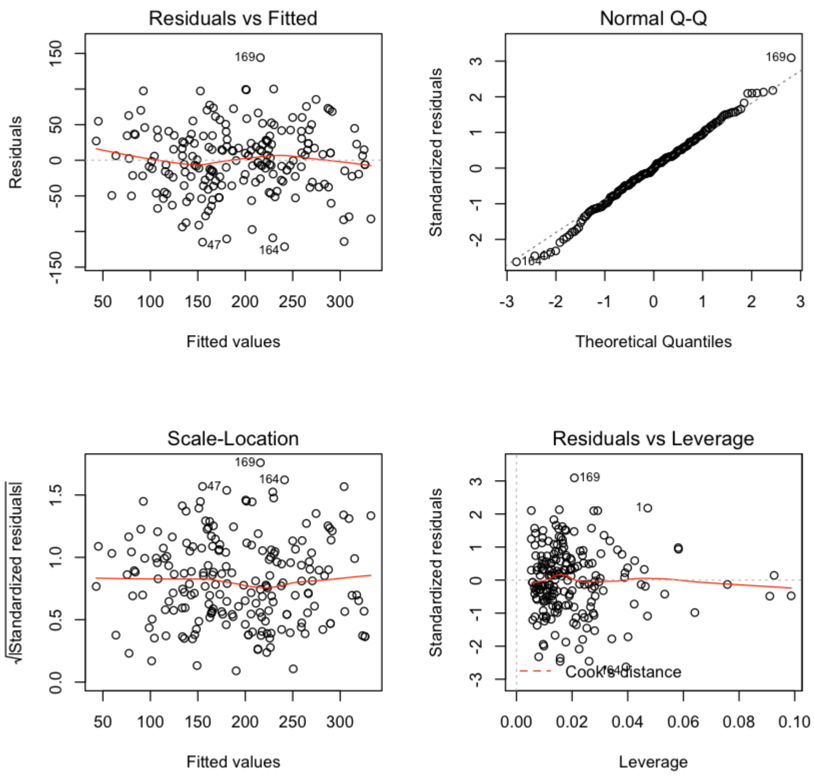
	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	198	862264.2	NA	NA	NA	NA
2	196	434574.6	2	427689.6	96.44738	6.879395e-30

In [53]:
```
### [ Step.5 ] Visualization: Scatter plot of the data (NOT lm model!)

scatter <- ggplot(album, aes(adverts, sales))
scatter + geom_point() +
    geom_smooth(method = "lm", se = TRUE, colour = "Blue") +
    labs(x="Advertisement", y="Album Sales") +
    theme(axis.text=element_text(size=12),
          axis.title=element_text(size=14, face="bold"))
```

```
`geom_smooth()` using formula 'y ~ x'
```



In [10]:
```
### [ Step.2 ] Check assumptions via visual inspection

par(mfrow=c(2,2))
plot(albumSales.3)
```



~ *Report* ~

- **The total album sales are positively associated with the advertising budget ($\beta_{Standardized}$ = 0.51, p < .001), the plays on radio ($\beta_{Standardized}$ = 0.51, p < .001), and the attractiveness ($\beta_{Standardized}$ = 0.19, p < .001). This multiple regression model explains 66% of total variance in album sales.**

---

(3) Assumptions in Regression Analysis

Example: Album Sales (model 3 → albumSales.3)

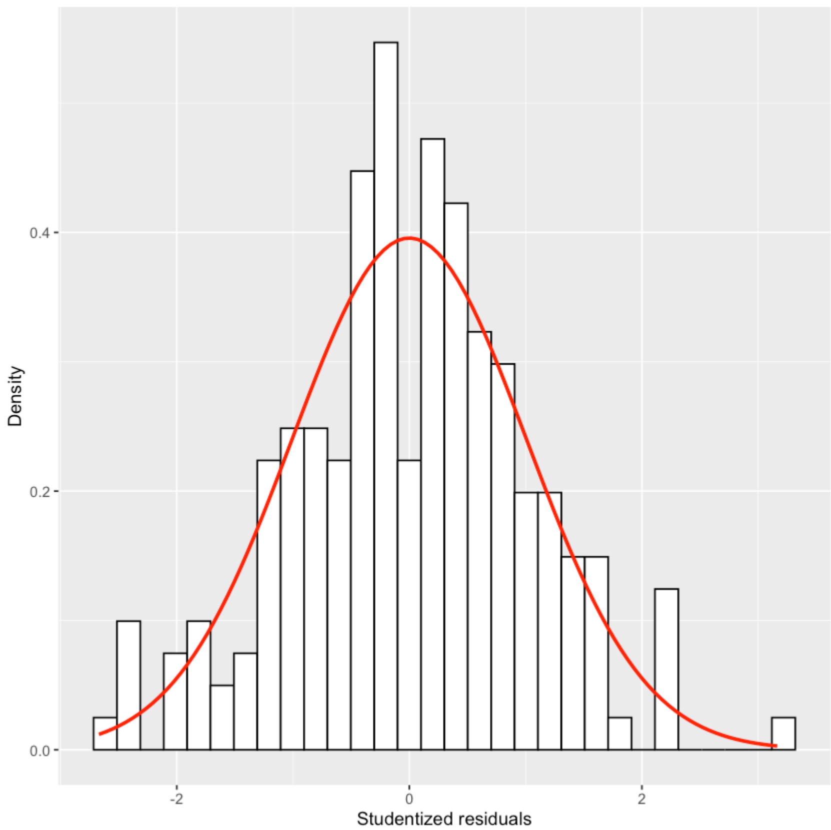
In [95]:

```r
#--- (3.1) Outliers / Influential cases ---#
# → influence.measures {stats}

#influence.measures(albumSales.3)
performance::check_outliers(albumSales.3)

(album$residuals <- resid(albumSales.3))
(album$standardized.residuals <- rstandard(albumSales.3))
(album$studentized.residuals <- rstudent(albumSales.3))
#album$cooks.distance <- cooks.distance(albumSales.3)
#album$dfbeta <- dfbeta(albumSales.3)
#album$dffit <- dffits(albumSales.3)
#album$leverage <- hatvalues(albumSales.3)
#album$covariance.ratios <- covratio(albumSales.3)
```

1.11010994482881 **128:** 0.656607013889276 **129:** 0.2974872573205 **130:** -0.483668639077895 **131:** 0.85983538026204 **132:** -0.156179790092755 **133:** 1.24452498411241 **134:** -0.211317997934779 **135:** -0.589943700082878 **136:** 0.225205565975611 **137:** 0.685232694533605 **138:** -0.477460922026658 **139:** 0.143072936397227 **140:** -0.0677014279834269 **141:** -0.805293001522956 **142:** 0.143102235064092 **143:** 0.274814216186551 **144:** -0.482569225908232 **145:** -1.10152767667313 **146:** -1.13011501392191 **147:** -0.335981918229487 **148:** 1.52539226298683 **149:** -1.09574827681366 **150:** -1.20634243360992 **151:** 0.742067177512104 **152:** -1.53060555013669 **153:** 0.519567878561133 **154:** 0.512962439684182 **155:** -1.4539307347172 **156:** -1.94342315839657 **157:** -0.294916371010763 **158:** 0.272356616455765 **159:** 0.798067968869423 **160:** -0.321517637782994 **161:** -1.11116664336877 **162:** -0.987764384207508 **163:** 1.41557801539676 **164:** -2.62881409071654 **165:** 1.31567147719999 **166:** 1.50069802350441 **167:** -1.99787653846771 **168:** -0.611677438597921 **169:** 3.09333296653134 **170:** 0.623366598443562 **171:** 0.729793655211063 **172:** 0.60564501902147 **173:** -0.161682835459264 **174:** 0.914724974902887 **175:** -0.811427609861406 **176:** -1.36960220340534 **177:** 1.54943858224622 **178:** -0.594882546550402 **179:** 0.510067233456521 **180:** -0.0561504531582476 **181:** 0.141479935737933 **182:** -1.18618778616627 **183:** -0.726661105021802 **184:** -0.133768780423567 **185:** 0.220232036085593 **186:** 0.390668838406333 **187:** 0.321803175424325 **188:** 0.294604439694965 **189:** 1.07279383440115 **190:** -0.144030084642013 **191:** -1.15300012964483 **192:** -0.767281943778982 **193:**

In [65]:
```r
#--- (3.2) Normality of residuals ---#

album$fitted <- albumSales.3$fitted.values
album$residuals <- resid(albumSales.3)
album$studentized.residuals <- rstudent(albumSales.3)

shapiro.test(album$residuals)
shapiro.test(album$studentized.residuals)

# visual inspection
resid.hist <- ggplot(album, aes(studentized.residuals)) +
    geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
    labs(x="Studentized residuals", y = "Density")
resid.hist + stat_function(fun = dnorm,
                args= list(mean=mean(album$studentized.residuals, na.rm=TRUE),
                sd = sd(album$studentized.residuals, na.rm=TRUE)),
                colour = "red", size = 1)
```

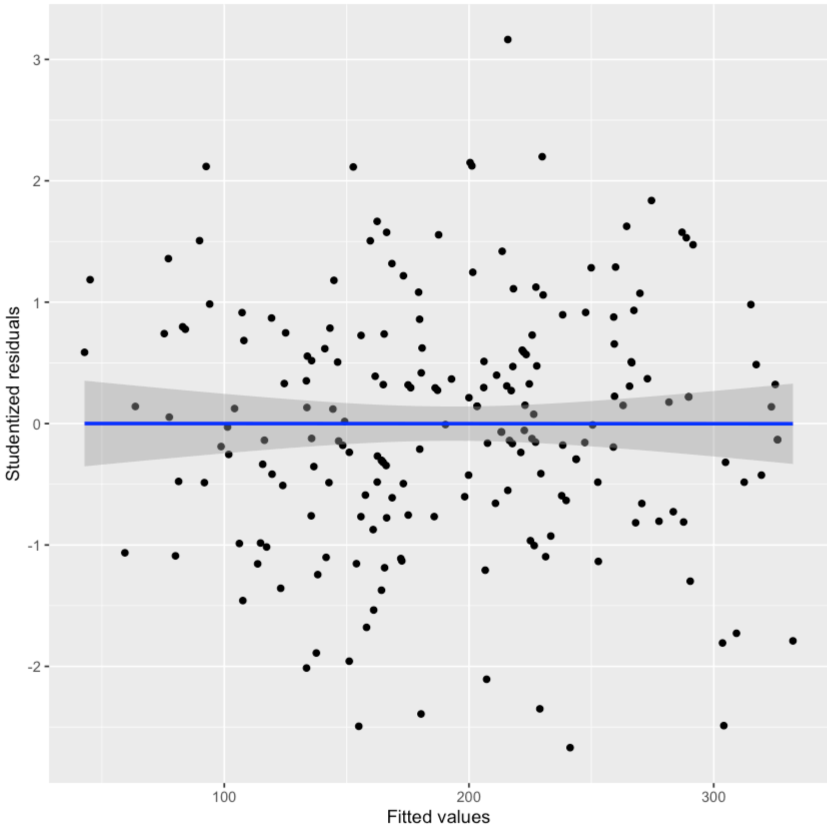
```
        Shapiro-Wilk normality test

data:  album$residuals
W = 0.99483, p-value = 0.7253


        Shapiro-Wilk normality test

data:  album$studentized.residuals
W = 0.99465, p-value = 0.6975

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In [47]:
```
#--- (3.3) Homoscedasticity ---#

# → check_heteroscedasticity {performance}
# reporting the p-value. p-value < 0.05 indicates a non-constant variance (hete
performance::check_heteroscedasticity(albumSales.3) %>% round(3)

# visual inspection
resid.scatter <- ggplot(album, aes(fitted, studentized.residuals))
resid.scatter + geom_point() +
    geom_smooth(method = "lm", colour = "Blue") +
    labs(x="Fitted values", y="Studentized residuals")
```

0.582

```
`geom_smooth()` using formula 'y ~ x'
```



In [49]:
```
#--- (3.4) Multicollinearity ---#

# → variance inflation factor (VIF) {car}
performance::check_collinearity(albumSales.3)

#--- Standardized parameter estimates with the lm.beta() function---
QuantPsyc::lm.beta(albumSales.3) %>% round(3)
```

A check_collinearity: 3 × 3

	Term	VIF	SE_factor
	<chr>	<dbl>	<dbl>
1	adverts	1.014593	1.007270
2	attract	1.038455	1.019046
3	airplay	1.042504	1.021031

adverts: 0.511 **attract:** 0.192 **airplay:** 0.512

In [48]:
```
#--- (3.5) Autocorrelation ---#

# → Durbin-Watson Test (D-W Test) {car}
car::dwt(albumSales.3)
#car::durbinWatsonTest(albumSales.3)

# D-W Test returns the p-value. A p-value < 0.05 indicates autocorrelated resi
performance::check_autocorrelation(albumSales.3) %>% round(3)
```

```
 lag Autocorrelation D-W Statistic p-value
   1       0.0026951      1.949819   0.682
 Alternative hypothesis: rho != 0
```

0.764

> ***What if the normality assumption was violated?***
>
> - Robust regression through **Bootstrapping**

In [79]:
```
bootReg <- function(formula, data, indices)
{
d <- data
fit <- lm(formula, data = d)
return(coef(fit))
}
```

In [80]:
```
BootResults <- boot(statistic = bootReg,
                    formula = sales ~ adverts + airplay + attract,
                    data = album, R = 5000)
```

In [85]:
```
summary(BootResults)
```

A summary.boot: 4 × 5

	R	original	bootBias	bootSE	bootMed
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	5000	-26.61295836	0	0	-26.61295836
2	5000	0.08488483	0	0	0.08488483
3	5000	3.36742517	0	0	3.36742517
4	5000	11.08633520	0	0	11.08633520