

Lesson 04

Christopher A. Swenson (chris@csvenson.com)

11/27/2021

Alcohol consumption and muscle strength Load the alcoholarm data. Fit a simple linear regression model with $y = \text{strength}$ and $x = \text{alcohol}$. Display model results. Display a scatterplot of the data with the simple linear regression line. Display a residual plot with fitted values on the horizontal axis. Display a residual plot with $x = \text{alcohol}$ on the horizontal axis.

```
alcoholarm <- read.table("./Data/alcoholarm.txt", header=T)
attach(alcoholarm)

model <- lm(strength ~ alcohol)
summary(model)

##
## Call:
## lm(formula = strength ~ alcohol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.7847 -2.5450 -0.1477  2.6359  7.4815
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.36954    1.20273   21.925 < 2e-16 ***
## alcohol     -0.29587    0.05105   -5.796 5.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.874 on 48 degrees of freedom
## Multiple R-squared:  0.4117, Adjusted R-squared:  0.3994
## F-statistic: 33.59 on 1 and 48 DF,  p-value: 5.136e-07

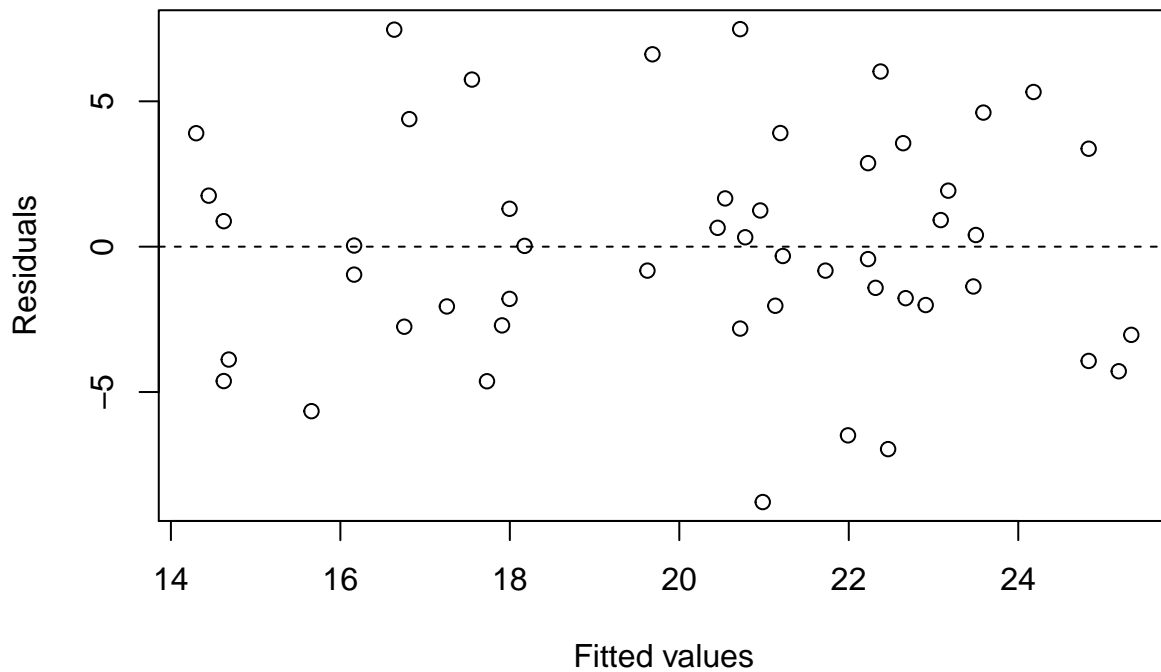
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) 26.36954    1.20273   21.925 < 2e-16 ***
# alcohol     -0.29587    0.05105   -5.796 5.14e-07 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 3.874 on 48 degrees of freedom
# Multiple R-squared:  0.4117, Adjusted R-squared:  0.3994
# F-statistic: 33.59 on 1 and 48 DF,  p-value: 5.136e-07

plot(x=alcohol, y=strength,
      xlab="Lifetime consumption of alcohol", ylab="Deltoid muscle strength",
```

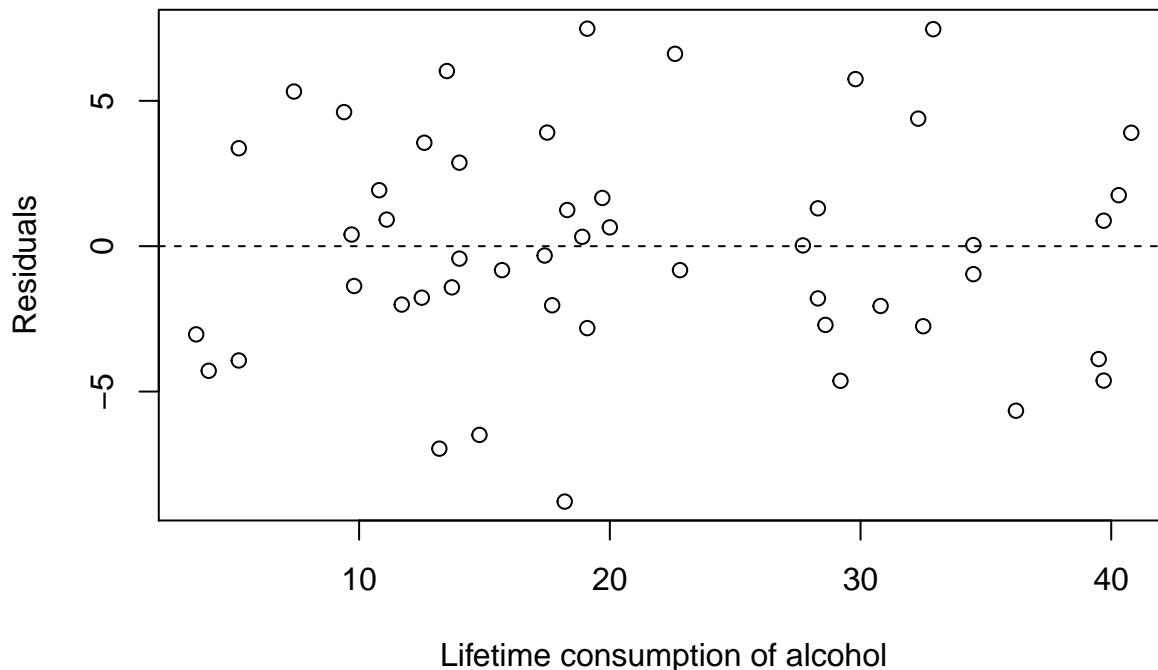
```
panel.last = lines(sort(alcohol), fitted(model)[order(alcohol)]))
```



```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
plot(x=alcohol, y=residuals(model),
     xlab="Lifetime consumption of alcohol", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(alcoholarm)
```

Blood pressure

Load the bloodpress data. Fit a simple linear regression model with $y = \text{BP}$ and $x = \text{Age}$, display model results, and display a scatterplot of the data with the simple linear regression line. Fit a simple linear regression model with $y = \text{BP}$ and $x = \text{Weight}$, display model results, and display a scatterplot of the data with the simple linear regression line. Fit a simple linear regression model with $y = \text{BP}$ and $x = \text{Duration}$, display model results, and display a scatterplot of the data with the simple linear regression line. Display a residual plot for the model using $x = \text{Age}$ with Weight on the horizontal axis. Fit a multiple linear regression model with $y = \text{BP}$, $x_1 = \text{Age}$, and $x_2 = \text{Weight}$. Display a residual plot for the model using $x_1 = \text{Age}$ and $x_2 = \text{Weight}$ with Duration on the horizontal axis.

```
bloodpress <- read.table("./Data/bloodpress.txt", header=T)
attach(bloodpress)

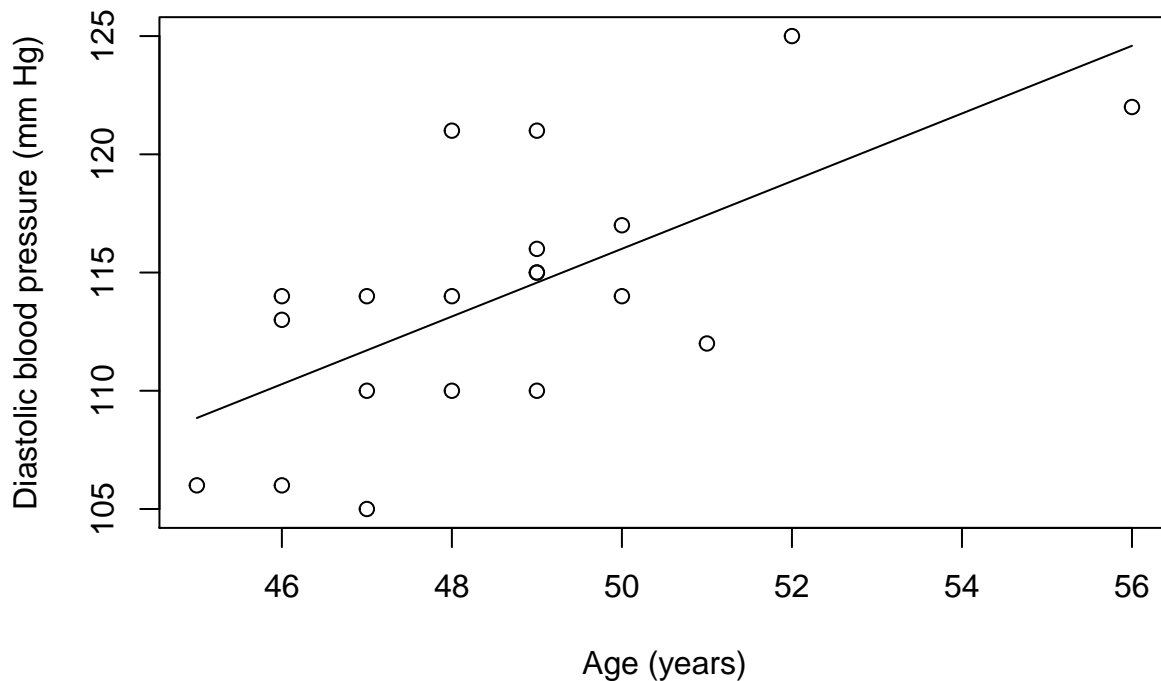
model.1 <- lm(BP ~ Age)
summary(model.1)

##
## Call:
## lm(formula = BP ~ Age)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.7104 -2.9217  0.4276  2.3973  7.8586
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  44.4545    18.7277   2.374  0.02894 *
## Age          1.4310     0.3849   3.718  0.00157 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.195 on 18 degrees of freedom
## Multiple R-squared:  0.4344, Adjusted R-squared:  0.403
## F-statistic: 13.82 on 1 and 18 DF,  p-value: 0.001574

# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  44.4545    18.7277   2.374  0.02894 *
# Age          1.4310     0.3849   3.718  0.00157 **
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Residual standard error: 4.195 on 18 degrees of freedom
# Multiple R-squared:  0.4344, Adjusted R-squared:  0.403
# F-statistic: 13.82 on 1 and 18 DF,  p-value: 0.001574
plot(x=Age, y=BP,
     xlab="Age (years)", ylab="Diastolic blood pressure (mm Hg)",
     panel.last = lines(sort(Age), fitted(model.1)[order(Age)]))
```

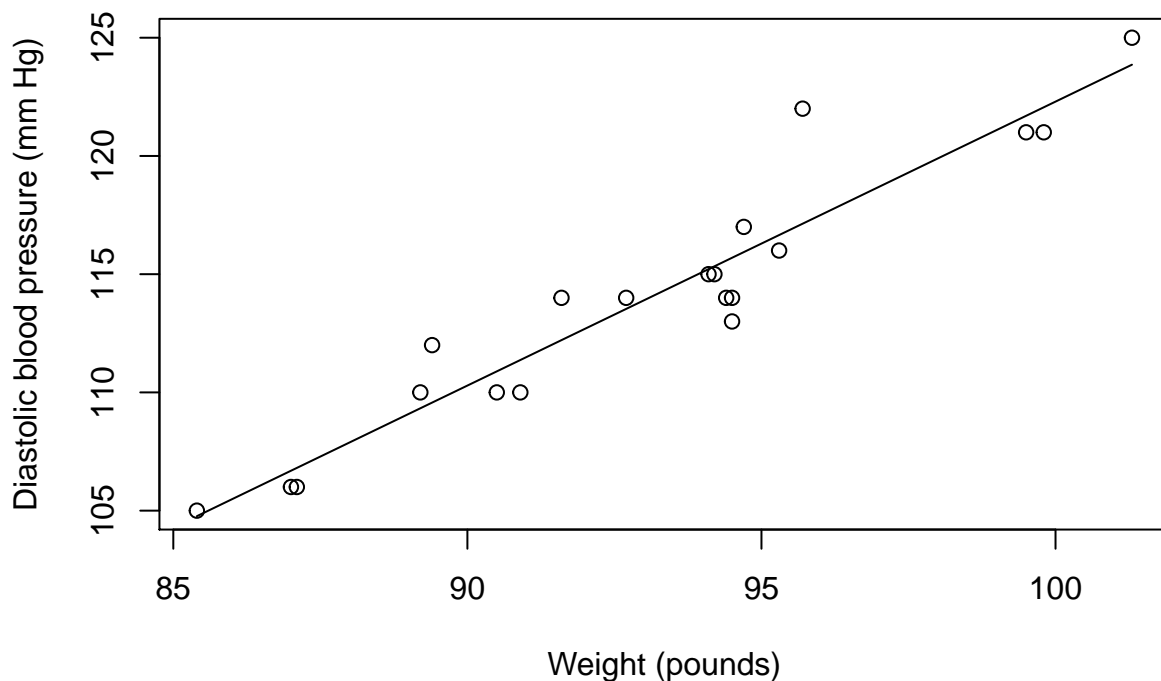


```
model.2 <- lm(BP ~ Weight)
summary(model.2)
```

```
##
## Call:
## lm(formula = BP ~ Weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6933 -0.9318 -0.4935  0.7703  4.8656
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.20531    8.66333   0.255   0.802
```

```
## Weight      1.20093    0.09297   12.917 1.53e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.74 on 18 degrees of freedom
## Multiple R-squared:  0.9026, Adjusted R-squared:  0.8972
## F-statistic: 166.9 on 1 and 18 DF,  p-value: 1.528e-10

# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  2.20531     8.66333   0.255   0.802
# Weight       1.20093     0.09297  12.917 1.53e-10 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 1.74 on 18 degrees of freedom
# Multiple R-squared:  0.9026, Adjusted R-squared:  0.8972
# F-statistic: 166.9 on 1 and 18 DF,  p-value: 1.528e-10
plot(x=Weight, y=BP,
     xlab="Weight (pounds)", ylab="Diastolic blood pressure (mm Hg)",
     panel.last = lines(sort(Weight), fitted(model.2)[order(Weight)]))
```



```
model.3 <- lm(BP ~ Dur)
summary(model.3)
```

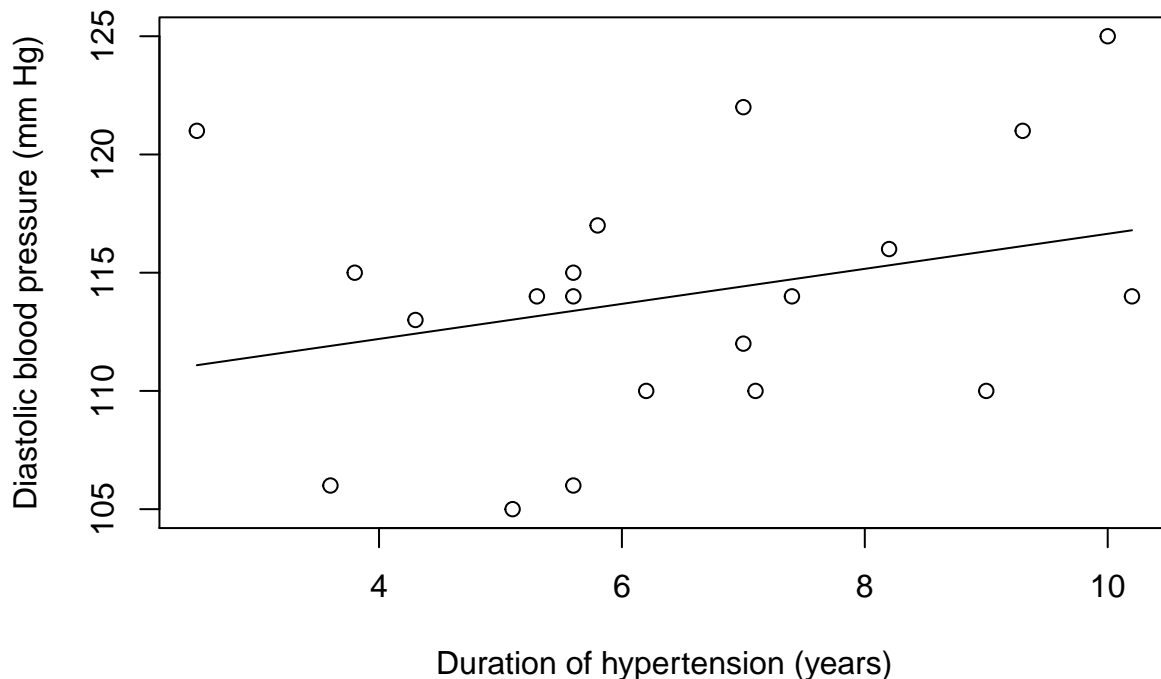
```
##
## Call:
## lm(formula = BP ~ Dur)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-8.0144	-3.9963	0.5968	3.0785	9.9124

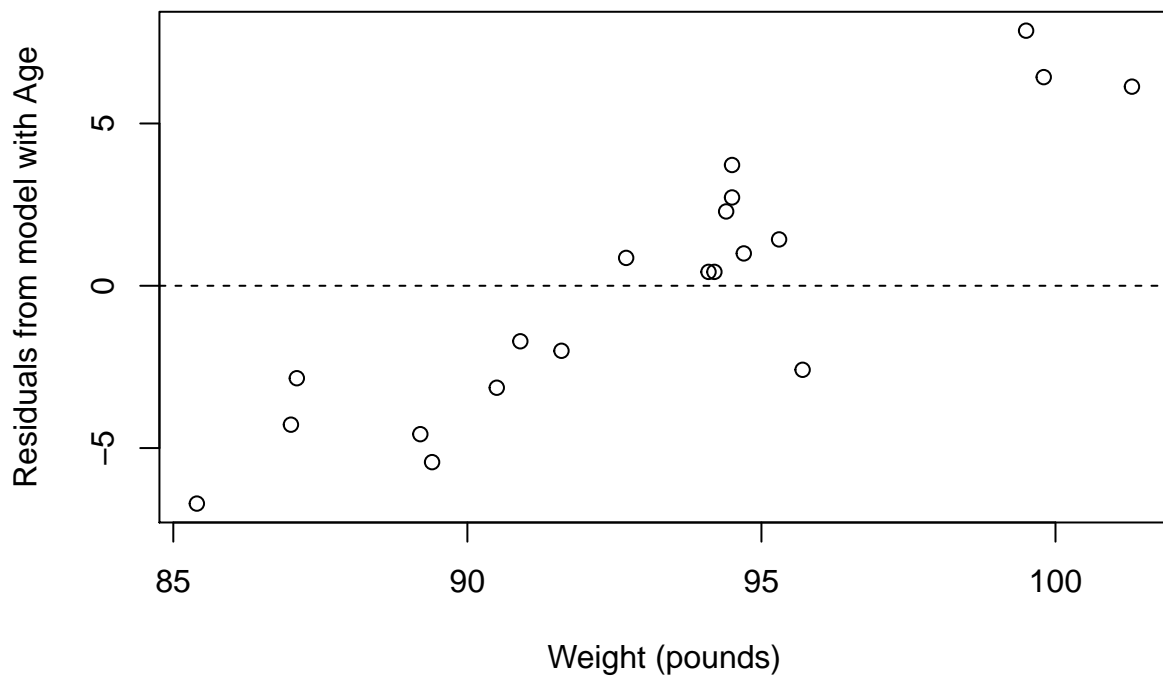
```
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 109.2350      3.8563  28.327  <2e-16 ***
## Dur          0.7411      0.5703   1.299    0.21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.333 on 18 degrees of freedom
## Multiple R-squared:  0.08575,    Adjusted R-squared:  0.03496
## F-statistic: 1.688 on 1 and 18 DF,  p-value: 0.2102
```

```
# Coefficients:
#           Estimate Std. Error t value Pr(>|t|)
# (Intercept) 109.2350      3.8563  28.327  <2e-16 ***
# Dur          0.7411      0.5703   1.299    0.21
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 5.333 on 18 degrees of freedom
# Multiple R-squared:  0.08575,    Adjusted R-squared:  0.03496
# F-statistic: 1.688 on 1 and 18 DF,  p-value: 0.2102
plot(x=Dur, y=BP,
     xlab="Duration of hypertension (years)",
     ylab="Diastolic blood pressure (mm Hg)",
     panel.last = lines(sort(Dur), fitted(model.3)[order(Dur)]))
```

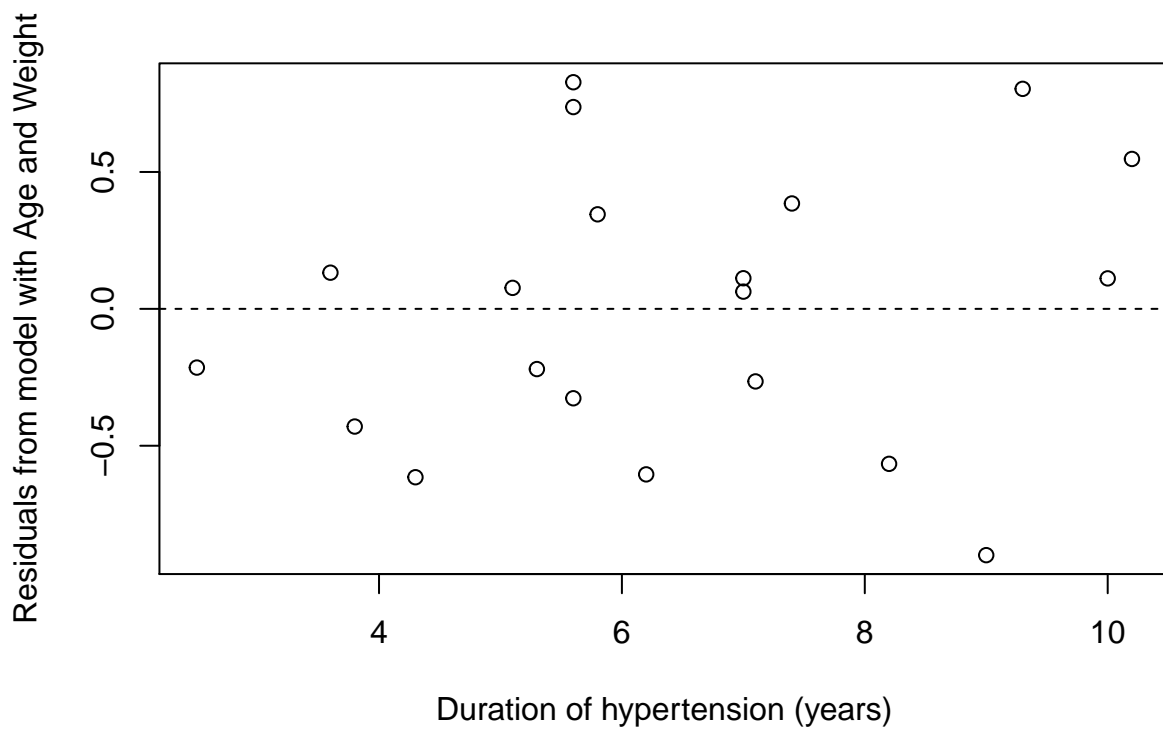


```
plot(x=Weight, y=residuals(model.1),
     xlab="Weight (pounds)", ylab="Residuals from model with Age",
     panel.last = abline(h=0, lty=2))
```



```
model.12 <- lm(BP ~ Age + Weight)

plot(x=Dur, y=residuals(model.12),
     xlab="Duration of hypertension (years)",
     ylab="Residuals from model with Age and Weight",
     panel.last = abline(h=0, lty=2))
```



```
detach(bloodpress)
```

Tread wear

Load the treadwear data. Fit a simple linear regression model with $y = \text{groove}$ and $x = \text{mileage}$. Display model results. Display a scatterplot of the data with the simple linear regression line. Display a residual plot with fitted values on the horizontal axis.

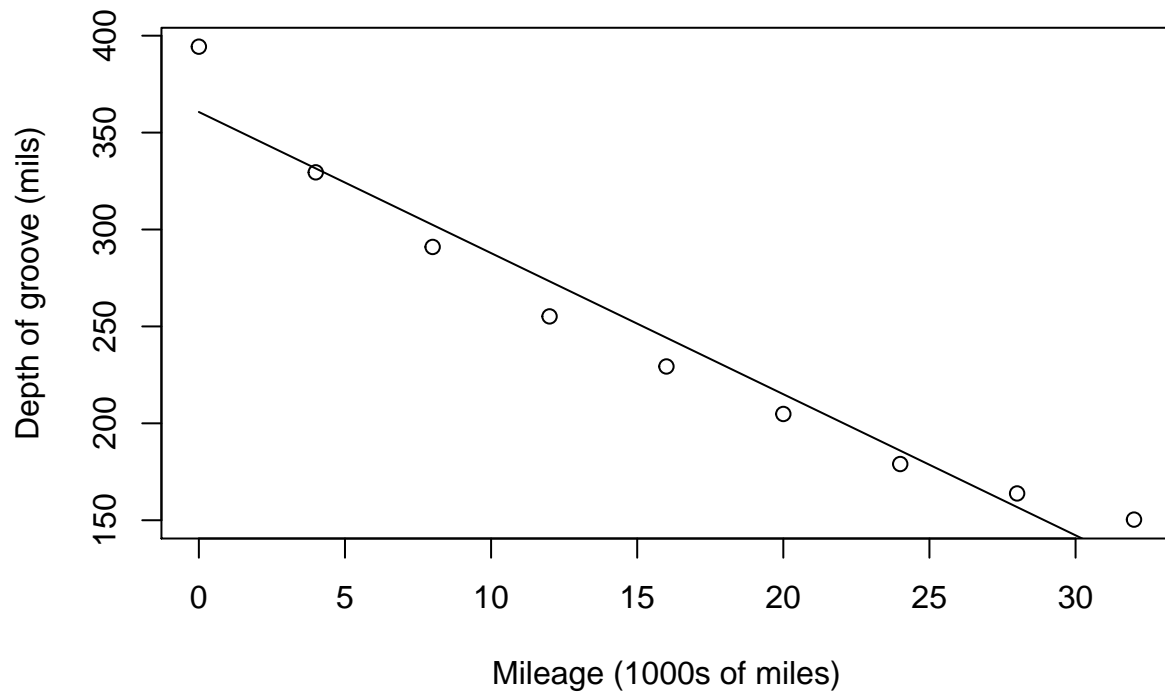
```
treadwear <- read.table("./Data/treadwear.txt", header=T)
attach(treadwear)

model <- lm(groove ~ mileage)
summary(model)

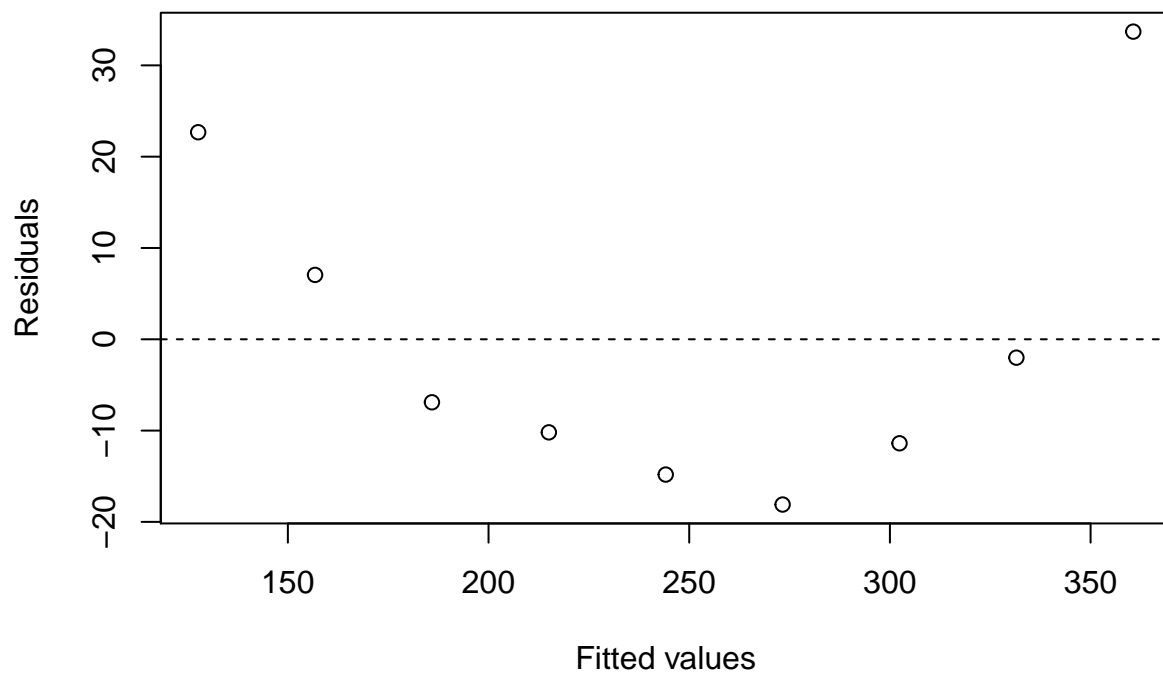
##
## Call:
## lm(formula = groove ~ mileage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.099 -11.392  -6.902   7.051  33.693
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  360.6367    11.6886   30.85 9.70e-09 ***
## mileage      -7.2806     0.6138  -11.86 6.87e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.02 on 7 degrees of freedom
## Multiple R-squared:  0.9526, Adjusted R-squared:  0.9458
## F-statistic: 140.7 on 1 and 7 DF, p-value: 6.871e-06

# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  360.6367    11.6886   30.85 9.70e-09 ***
# mileage      -7.2806     0.6138  -11.86 6.87e-06 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 19.02 on 7 degrees of freedom
# Multiple R-squared:  0.9526, Adjusted R-squared:  0.9458
# F-statistic: 140.7 on 1 and 7 DF, p-value: 6.871e-06

plot(x=mileage, y=groove,
     xlab="Mileage (1000s of miles)", ylab="Depth of groove (mils)",
     panel.last = lines(sort(mileage), fitted(model)[order(mileage)]))
```

```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(treadwear)
```

Plutonium

Load the `alphapluto` data. Fit a simple linear regression model with $y = \text{alpha}$ and $x = \text{pluto}$. Display model results. Display a scatterplot of the data with the simple linear regression line. Display a residual plot with

fitted values on the horizontal axis.

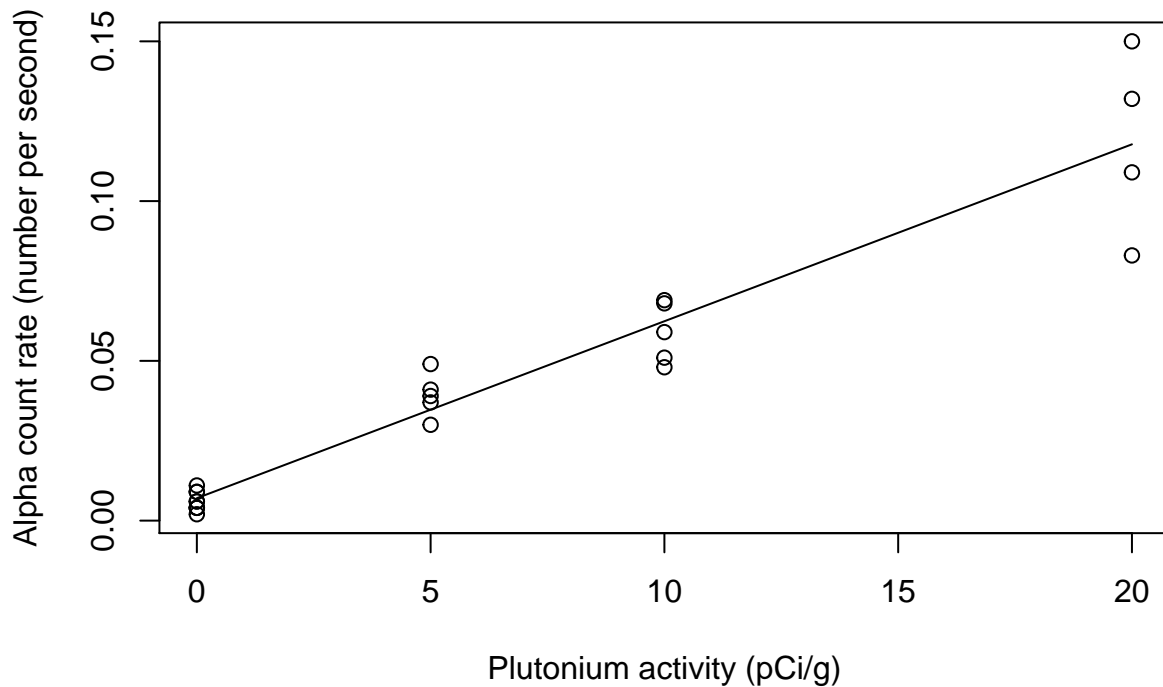
```
alphapluto <- read.table("./Data/alphapluto.txt", header=T)
attach(alphapluto)

model <- lm(alpha ~ pluto)
summary(model)

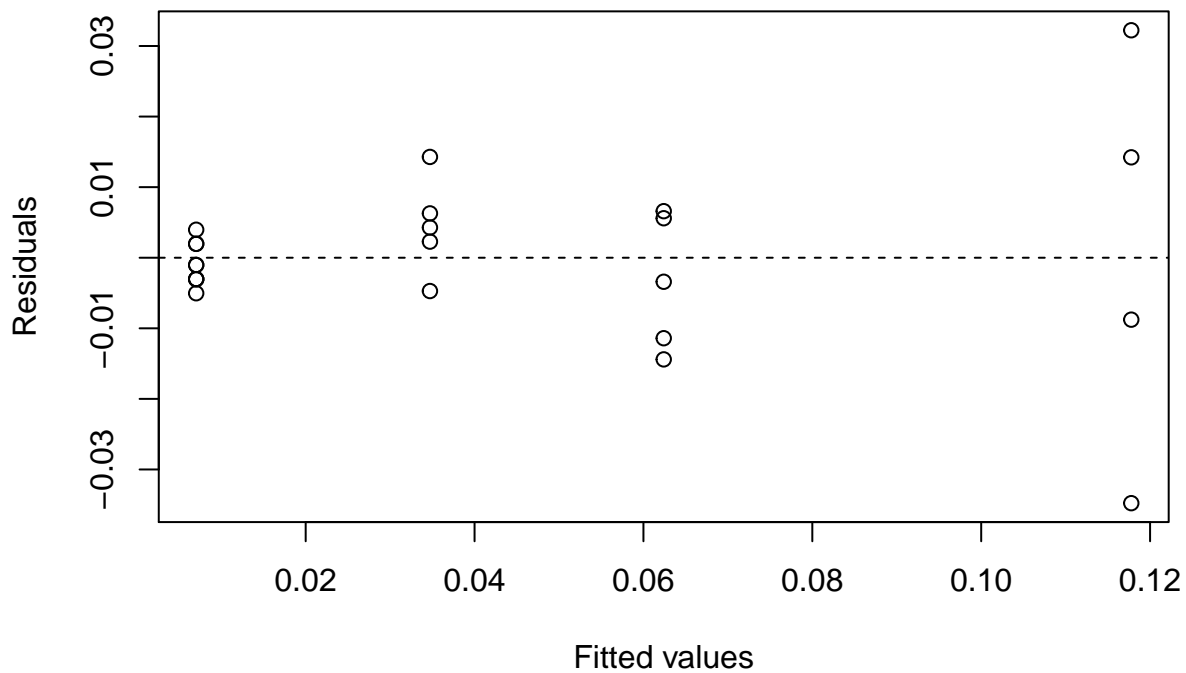
##
## Call:
## lm(formula = alpha ~ pluto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.034773 -0.004061 -0.001033  0.004939  0.032227
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0070331  0.0035988   1.954   0.0641 .
## pluto       0.0055370  0.0003659  15.133 9.08e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01257 on 21 degrees of freedom
## Multiple R-squared:  0.916, Adjusted R-squared:  0.912
## F-statistic: 229 on 1 and 21 DF, p-value: 9.077e-13

# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) 0.0070331  0.0035988   1.954   0.0641 .
# pluto       0.0055370  0.0003659  15.133 9.08e-13 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.01257 on 21 degrees of freedom
# Multiple R-squared:  0.916, Adjusted R-squared:  0.912
# F-statistic: 229 on 1 and 21 DF, p-value: 9.077e-13

plot(x=pluto, y=alpha,
     xlab="Plutonium activity (pCi/g)", ylab="Alpha count rate (number per second)",
     panel.last = lines(sort(pluto), fitted(model)[order(pluto)]))
```



```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(alphapluto)
```

Alcohol and tobacco

Load the `alcoholtobacco` data. Fit a simple linear regression model with $y = \text{Alcohol}$ and $x = \text{Tobacco}$. Display model results. Display a scatterplot of the data with the simple linear regression line. Display a

residual plot with fitted values on the horizontal axis. Refit the model excluding Northern Ireland. Display a scatterplot of the data excluding Northern Ireland with the simple linear regression line for the model excluding Northern Ireland. Display a standardized residual plot for the model fit to all the data with fitted values on the horizontal axis. Calculate the standardized residual for Northern Ireland.

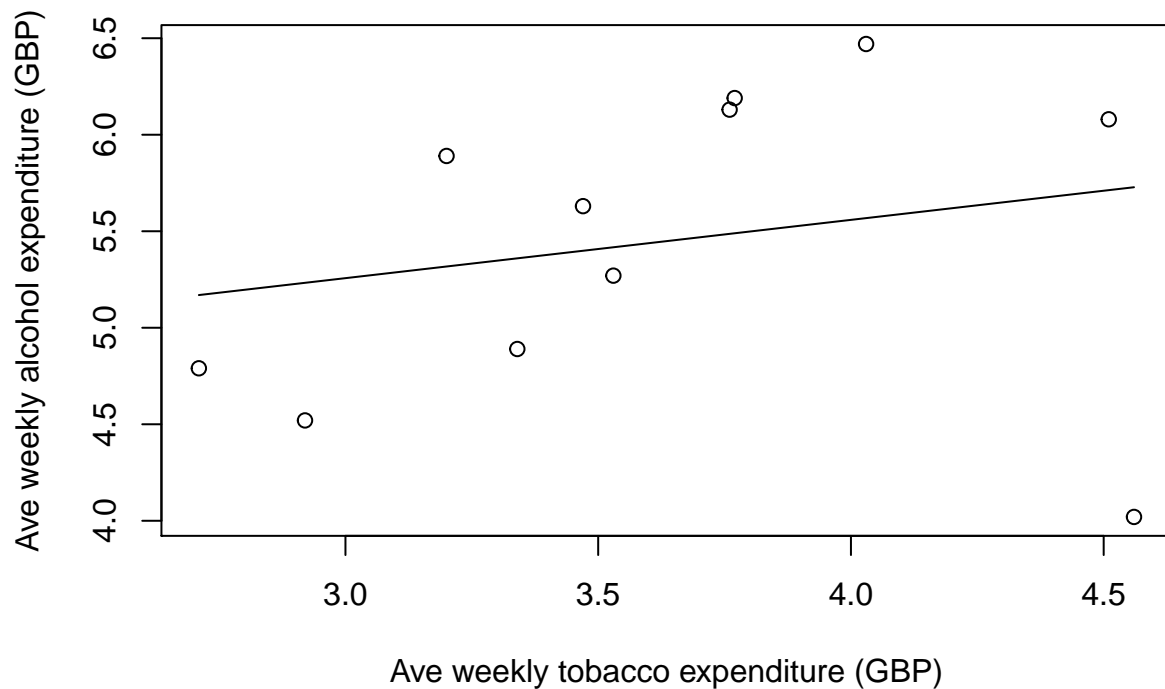
```
alcoholtobacco <- read.table("./Data/alcoholtobacco.txt", header=T)
attach(alcoholtobacco)

model.1 <- lm(Alcohol ~ Tobacco)
summary(model.1)

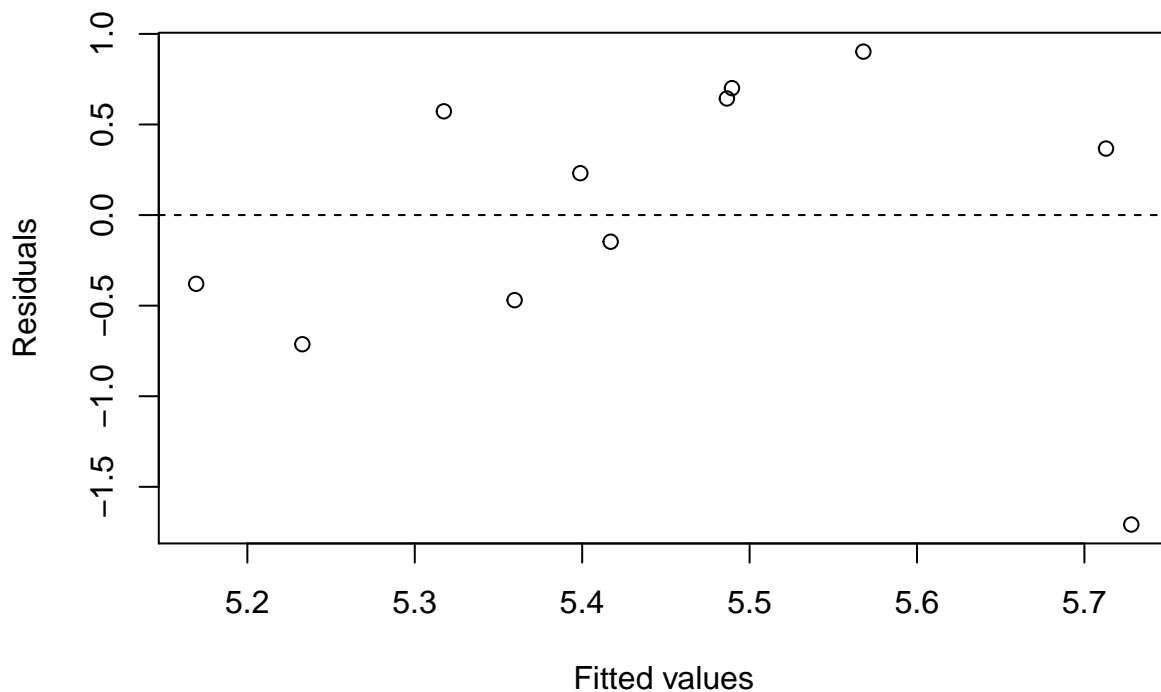
##
## Call:
## lm(formula = Alcohol ~ Tobacco)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7080 -0.4245  0.2311  0.6081  0.9020
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3512     1.6067   2.708  0.0241 *
## Tobacco       0.3019     0.4388   0.688  0.5087
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8196 on 9 degrees of freedom
## Multiple R-squared:  0.04998,    Adjusted R-squared:  -0.05557
## F-statistic: 0.4735 on 1 and 9 DF,  p-value: 0.5087

# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)   4.3512     1.6067   2.708  0.0241 *
# Tobacco       0.3019     0.4388   0.688  0.5087
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.8196 on 9 degrees of freedom
# Multiple R-squared:  0.04998,    Adjusted R-squared:  -0.05557
# F-statistic: 0.4735 on 1 and 9 DF,  p-value: 0.5087

plot(x=Tobacco, y=Alcohol,
     xlab="Ave weekly tobacco expenditure (GBP)",
     ylab="Ave weekly alcohol expenditure (GBP)",
     panel.last = lines(sort(Tobacco), fitted(model.1)[order(Tobacco)]))
```

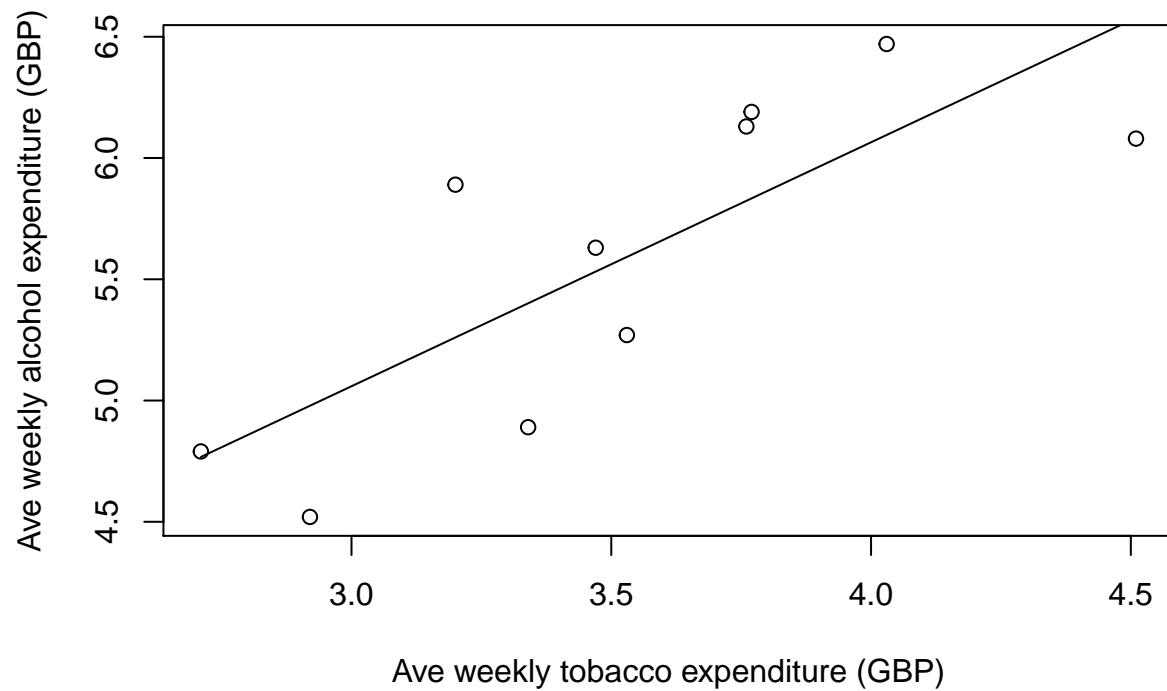


```
plot(x=fitted(model.1), y=residuals(model.1),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```

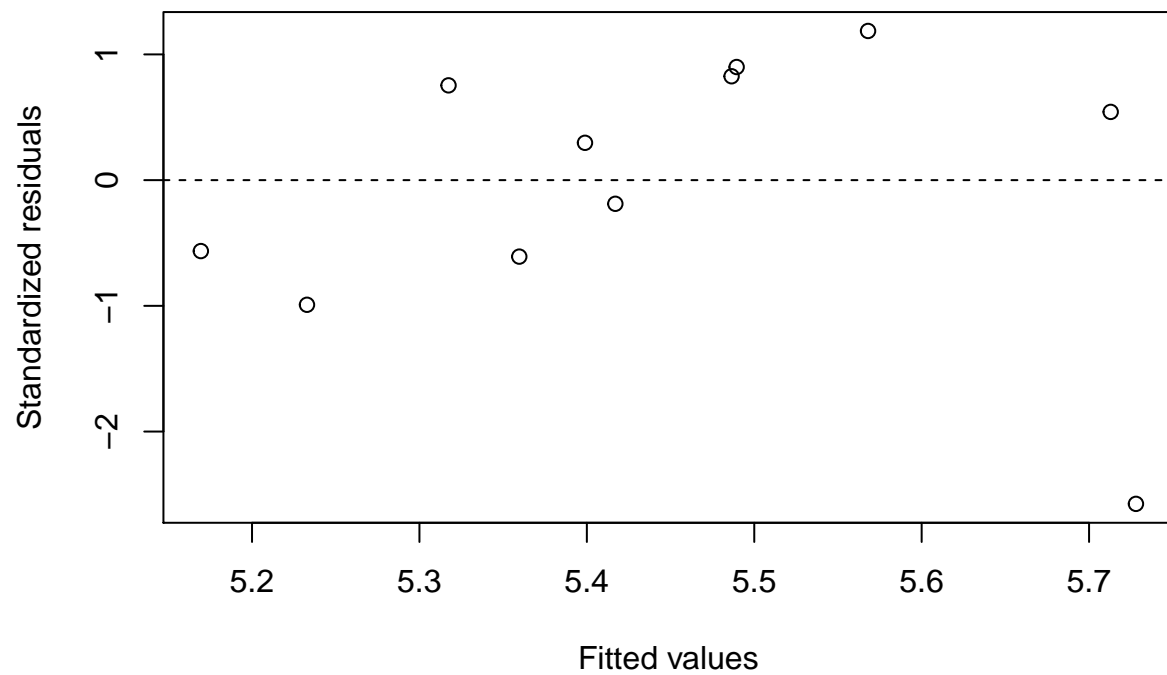


```
model.2 <- lm(Alcohol ~ Tobacco, subset=Region!="NorthernIreland")

plot(x=Tobacco[Region!="NorthernIreland"], y=Alcohol[Region!="NorthernIreland"],
     xlab="Ave weekly tobacco expenditure (GBP)",
     ylab="Ave weekly alcohol expenditure (GBP)",
     panel.last = lines(sort(Tobacco), fitted(model.2)[order(Tobacco)]))
```



```
plot(x=fitted(model.1), y=rstandard(model.1),
     xlab="Fitted values", ylab="Standardized residuals",
     panel.last = abline(h=0, lty=2))
```



```
rstandard(model.1)[Region=="NorthernIreland"] # -2.575075
```

```
##      11
## -2.575075
```

```
detach(alcoholtobacco)
```

Anscombe data

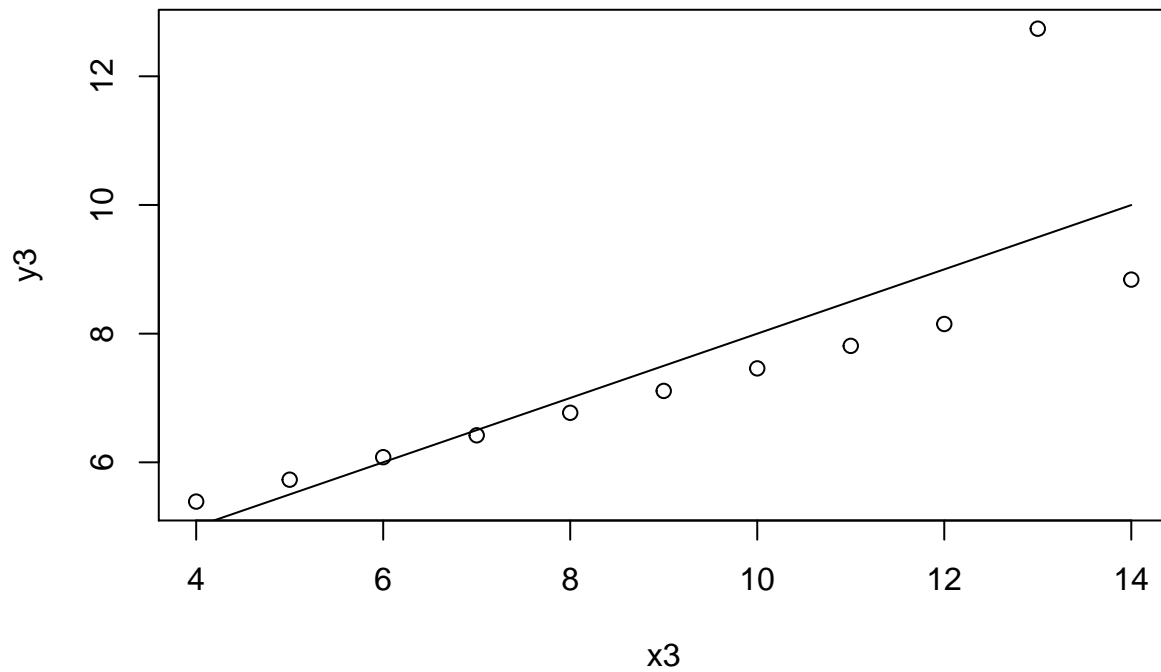
Load the anscombe data. Fit a simple linear regression model with $y = y3$ and $x = x3$. Display model results. Display a scatterplot of the data with the simple linear regression line. Display a residual plot with fitted values on the horizontal axis.

```
anscombe <- read.table("./Data/anscombe.txt", header=T)
attach(anscombe)

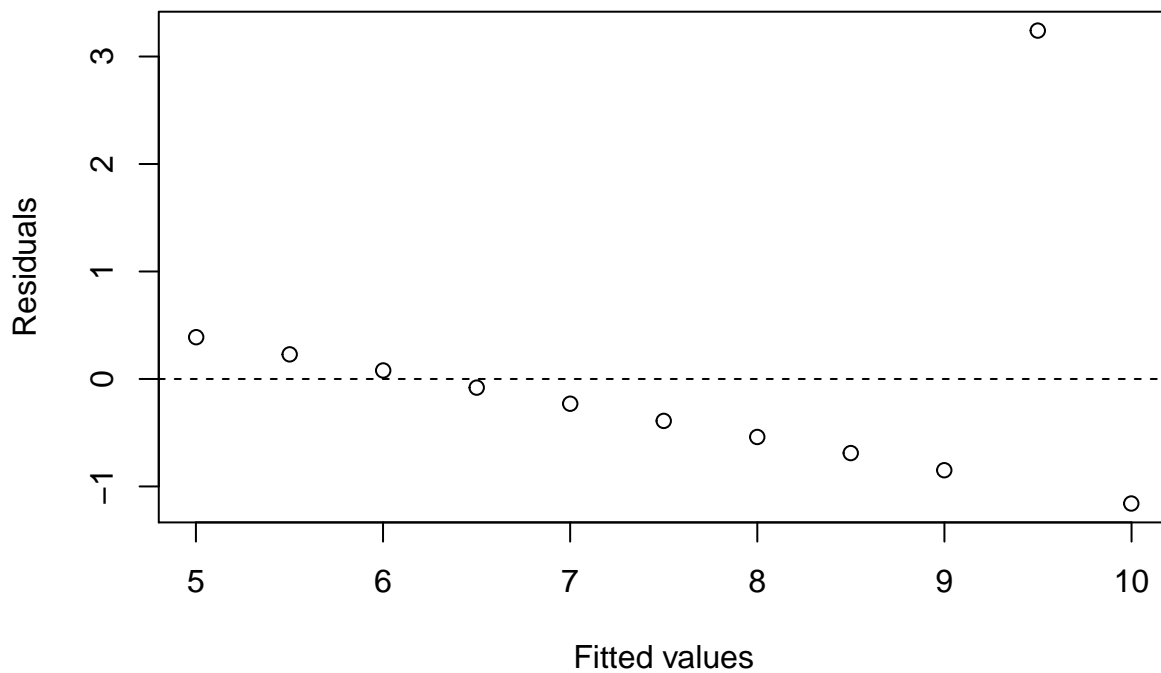
model <- lm(y3 ~ x3)
summary(model)

##
## Call:
## lm(formula = y3 ~ x3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x3            0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)   3.0025     1.1245   2.670  0.02562 *
# x3            0.4997     0.1179   4.239  0.00218 **
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 1.236 on 9 degrees of freedom
# Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
# F-statistic: 17.97 on 1 and 9 DF, p-value: 0.002176

plot(x=x3, y=y3,
      panel.last = lines(sort(x3), fitted(model)[order(x3)]))
```



```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(anscombe)
```

Skin cancer mortality

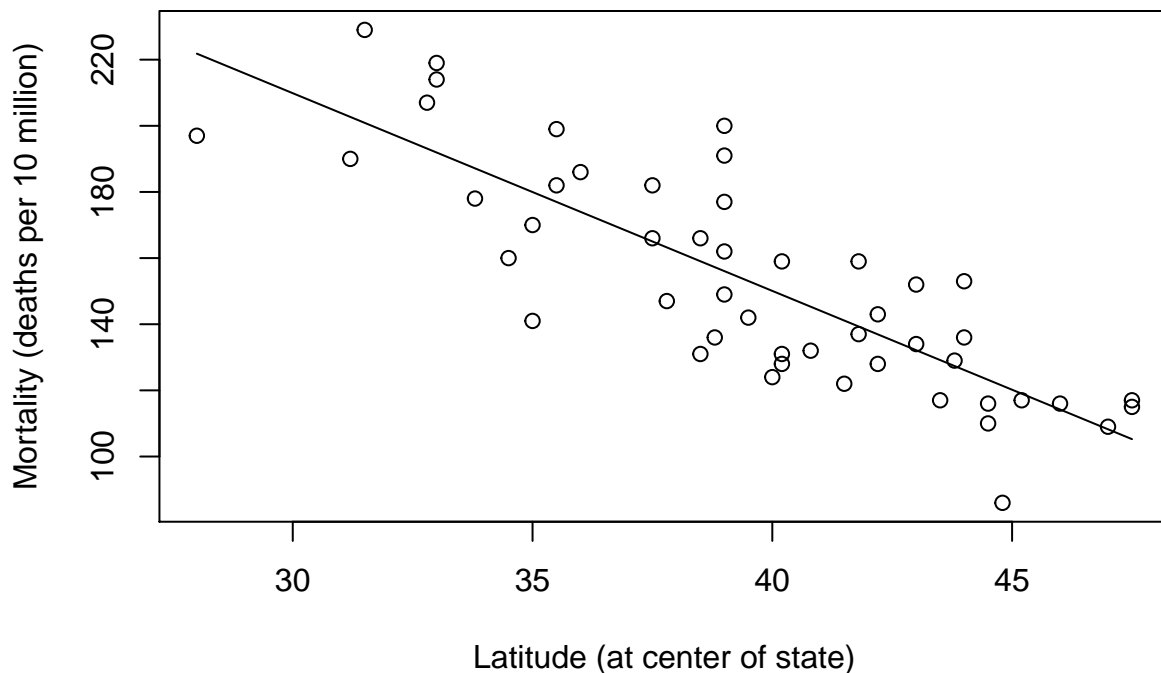
Load the skin cancer data. Fit a simple linear regression model with $y = \text{Mort}$ and $x = \text{Lat}$. Display a scatterplot of the data with the simple linear regression line.


```
skincancer <- read.table("./Data/skincancer.txt", header=T)
attach(skincancer)

model <- lm(Mort ~ Lat)

plot(x=Lat, y=Mort,
      xlab="Latitude (at center of state)", ylab="Mortality (deaths per 10 million)",
      main="Skin Cancer Mortality versus State Latitude",
      panel.last = lines(sort(Lat), fitted(model)[order(Lat)]))
```

Skin Cancer Mortality versus State Latitude



```
detach(skincancer)
```

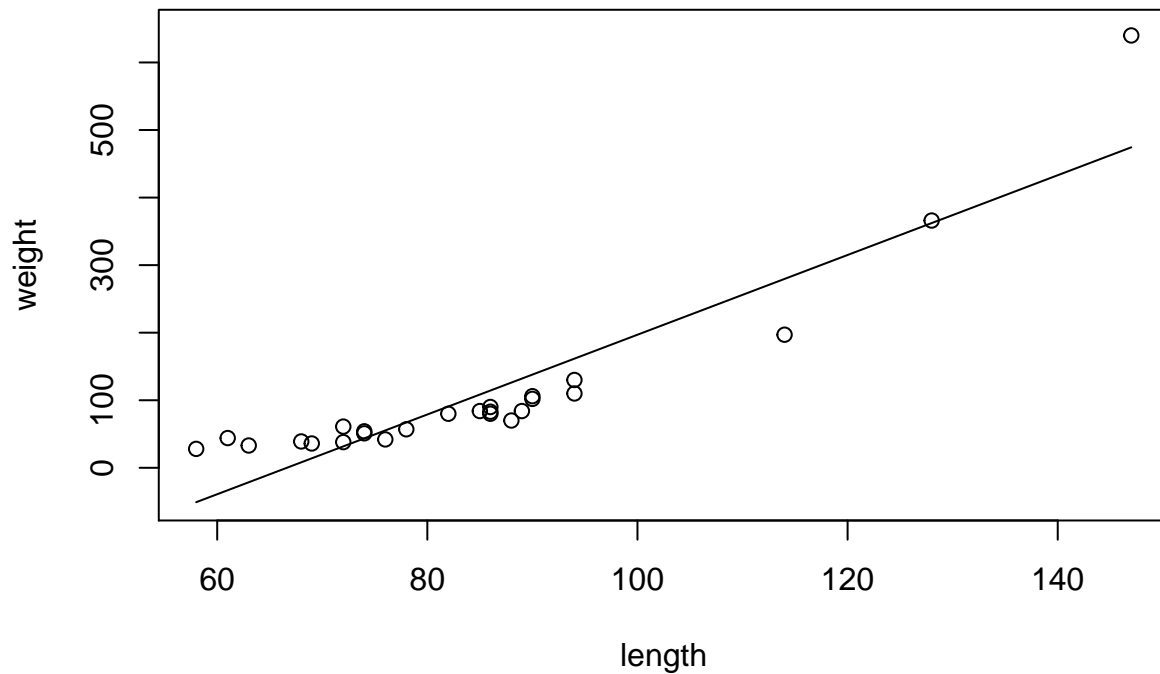
Alligators

Load the alligator data. Fit a simple linear regression model with $y = \text{weight}$ and $x = \text{length}$. Display a scatterplot of the data with the simple linear regression line.

```
alligator <- read.table("./Data/alligator.txt", header=T)
attach(alligator)

model <- lm(weight ~ length)

plot(x=length, y=weight, ylim=c(-50, 650),
      panel.last = lines(sort(length), fitted(model)[order(length)]))
```



```
detach(alligator)
```

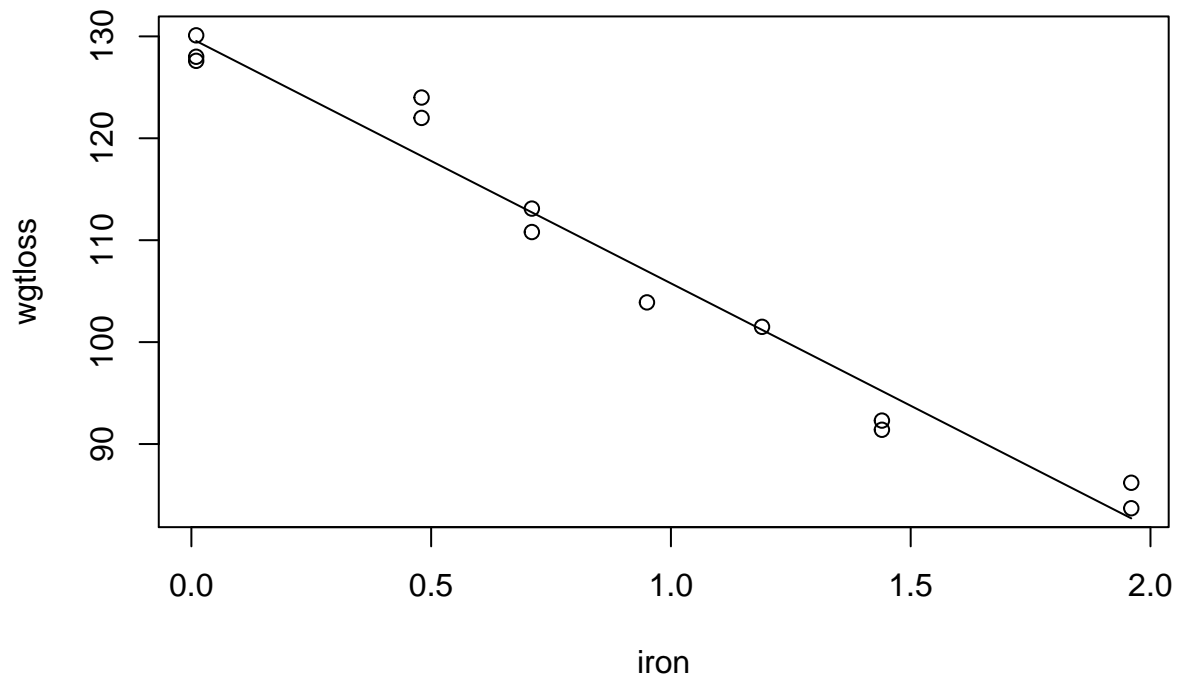
Alloy corrosion

Load the corrosion data. Fit a simple linear regression model with $y = \text{wgtloss}$ and $x = \text{iron}$. Display a scatterplot of the data with the simple linear regression line.

```
corrosion <- read.table("./Data/corrosion.txt", header=T)
attach(corrosion)

model <- lm(wgtloss ~ iron)

plot(x=iron, y=wgtloss,
      panel.last = lines(sort(iron), fitted(model)[order(iron)]))
```



```
detach(corrosion)
```

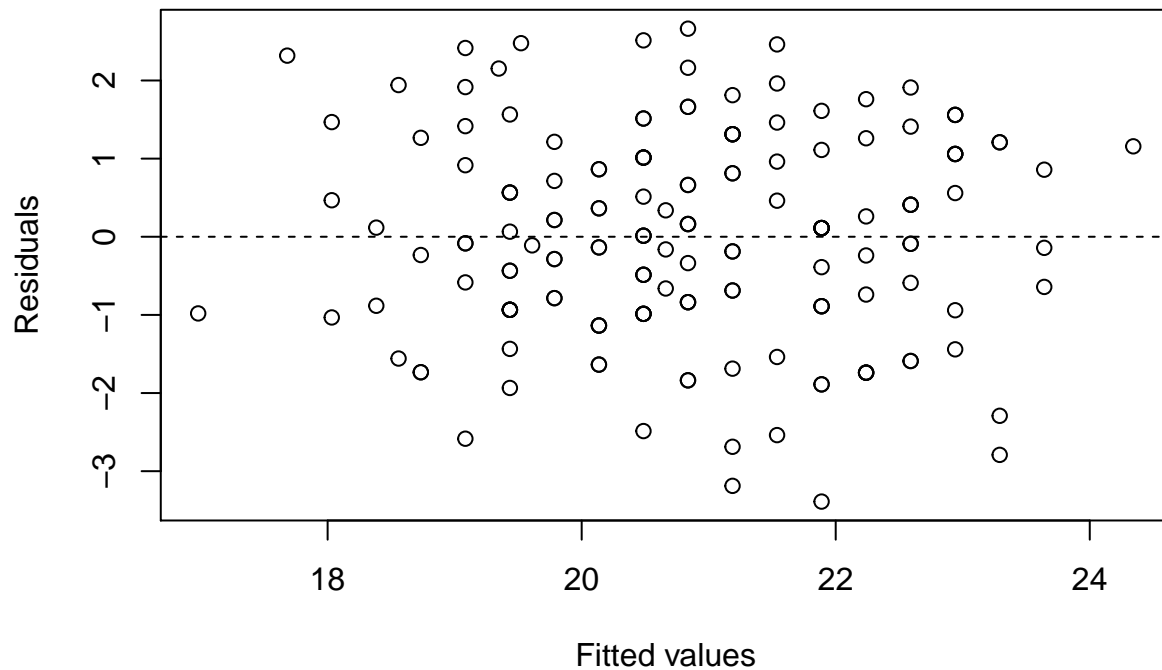
Handcode and height

Load the handheight data. Fit a simple linear regression model with $y = \text{HandSpan}$ and $x = \text{Height}$. Display a residual plot with fitted values on the horizontal axis.

```
handheight <- read.table("./Data/handheight.txt", header=T)
attach(handheight)
```

```
model <- lm(HandSpan ~ Height)
```

```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(handheight)
```

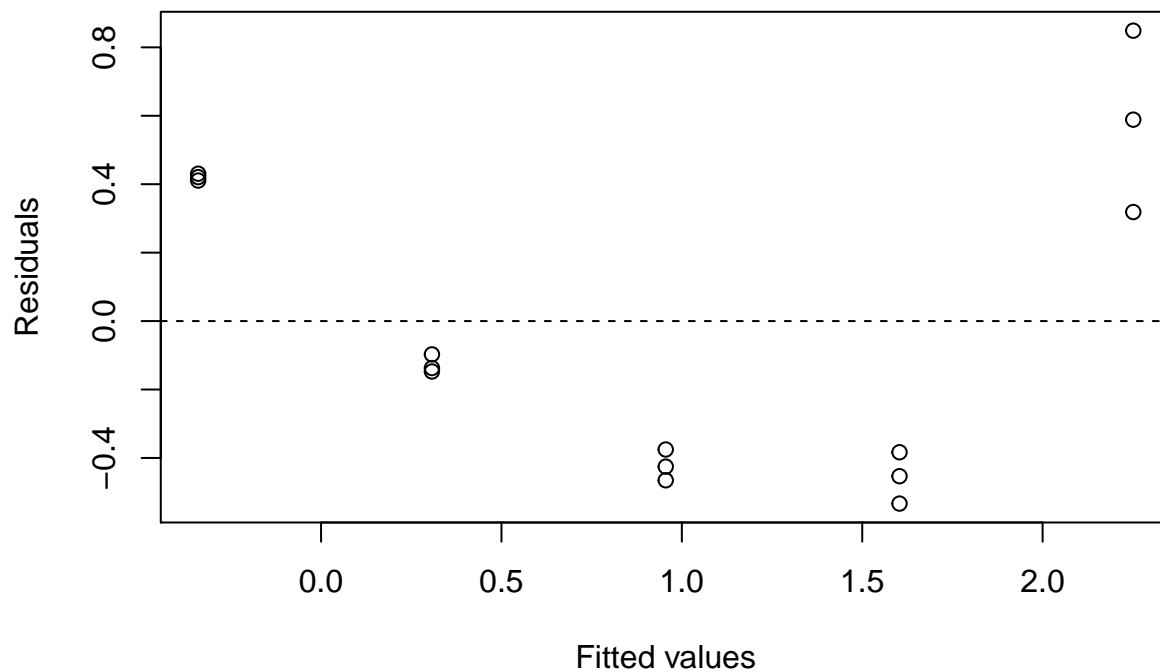
Chemical solution concentration

Load the solconc data. Fit a simple linear regression model with $y = y$ (concentration) and $x = x$ (time). Display a residual plot with fitted values on the horizontal axis.

```
solconc <- read.table("./Data/solutions_conc.txt", header=T)
attach(solconc)

model <- lm(y ~ x)

plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(solconc)
```

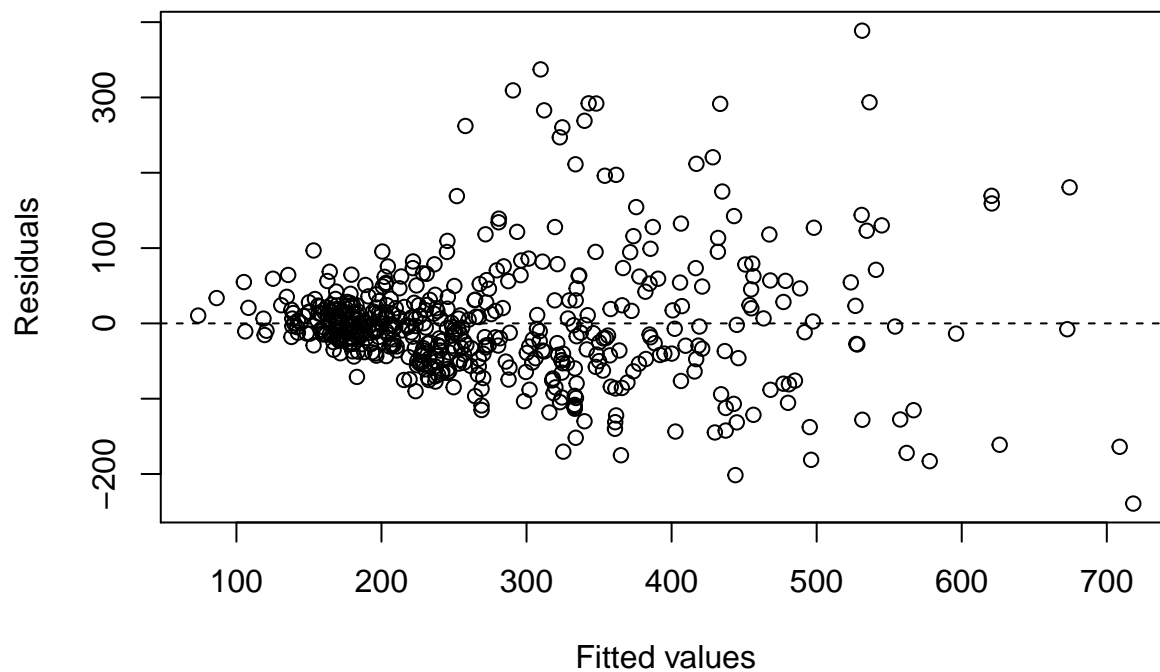
Real estate sales

Load the realestate data. Fit a simple linear regression model with $y = \text{SalePrice}$ and $x = \text{Sqfeet}$. Display a residual plot with fitted values on the horizontal axis.

```
realestate <- read.table("./Data/realestate_sales.txt", header=T)
attach(realestate)

model <- lm(SalePrice ~ SqFeet)

plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
detach(realestate)
```

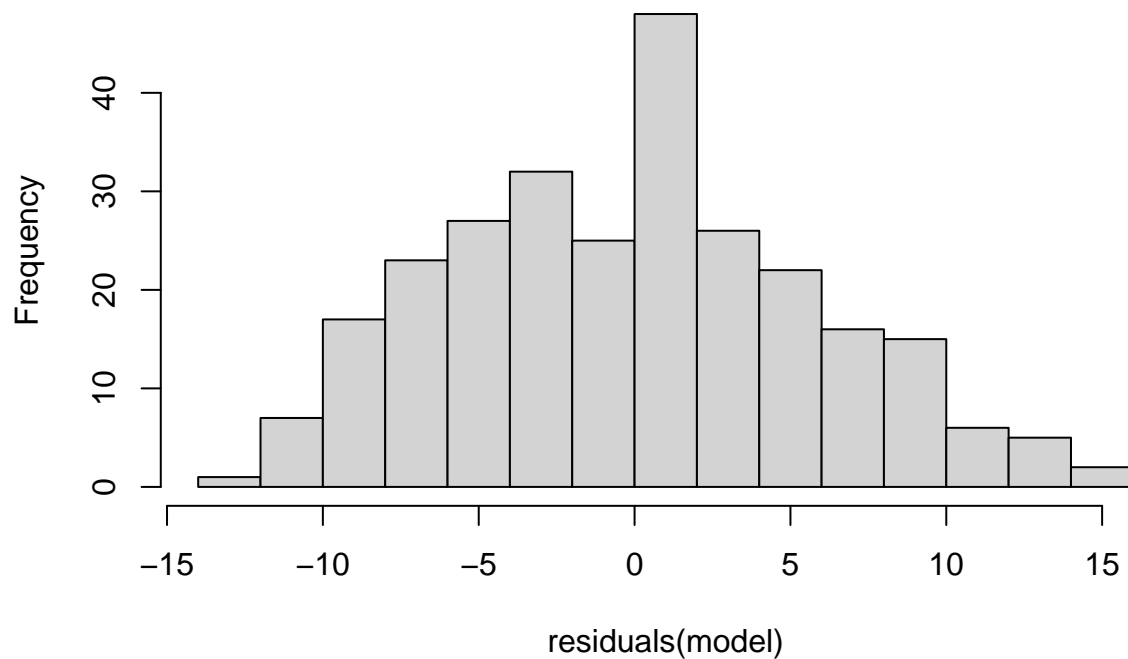
Old Faithful geyser eruptions

Load the oldfaithful data. Fit a simple linear regression model with $y = \text{waiting}$ and $x = \text{eruption}$. Display a histogram and normal probability plot of the residuals.

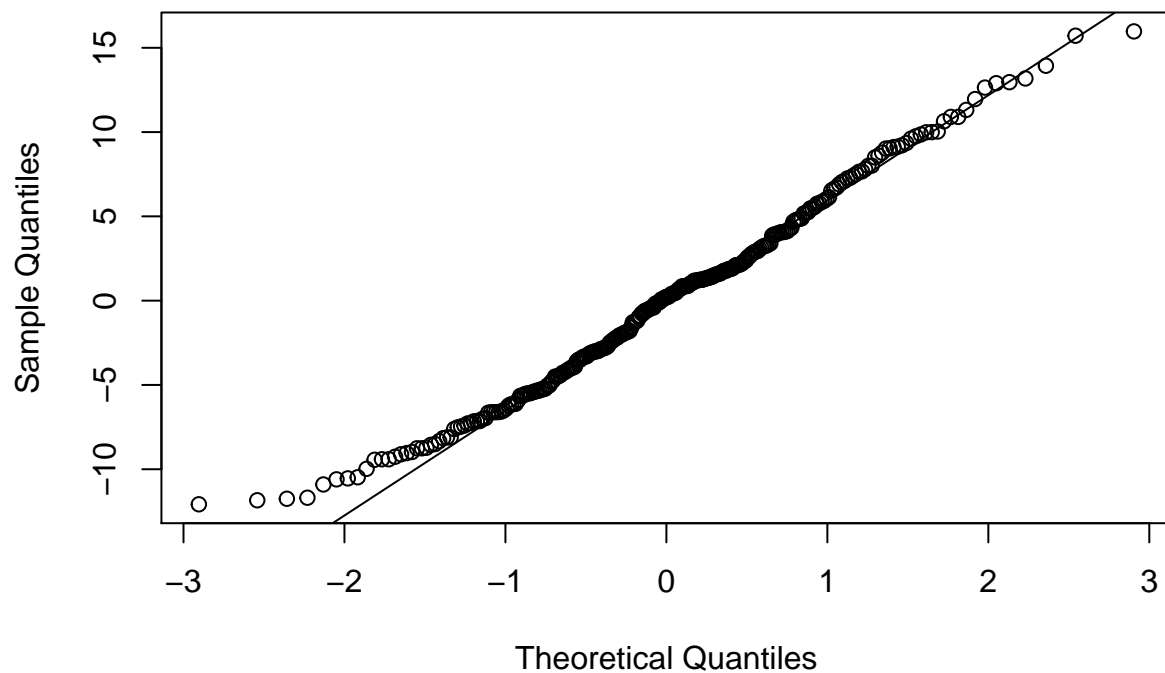
```
oldfaithful <- read.table("./Data/oldfaithful.txt", header=T)
attach(oldfaithful)
```

```
model <- lm(waiting ~ eruption)
```

```
hist(residuals(model), main="", breaks=12)
```



```
qqnorm(residuals(model), main="")
qqline(residuals(model))
```



```
detach(oldfaithful)
```

Hospital infection risk

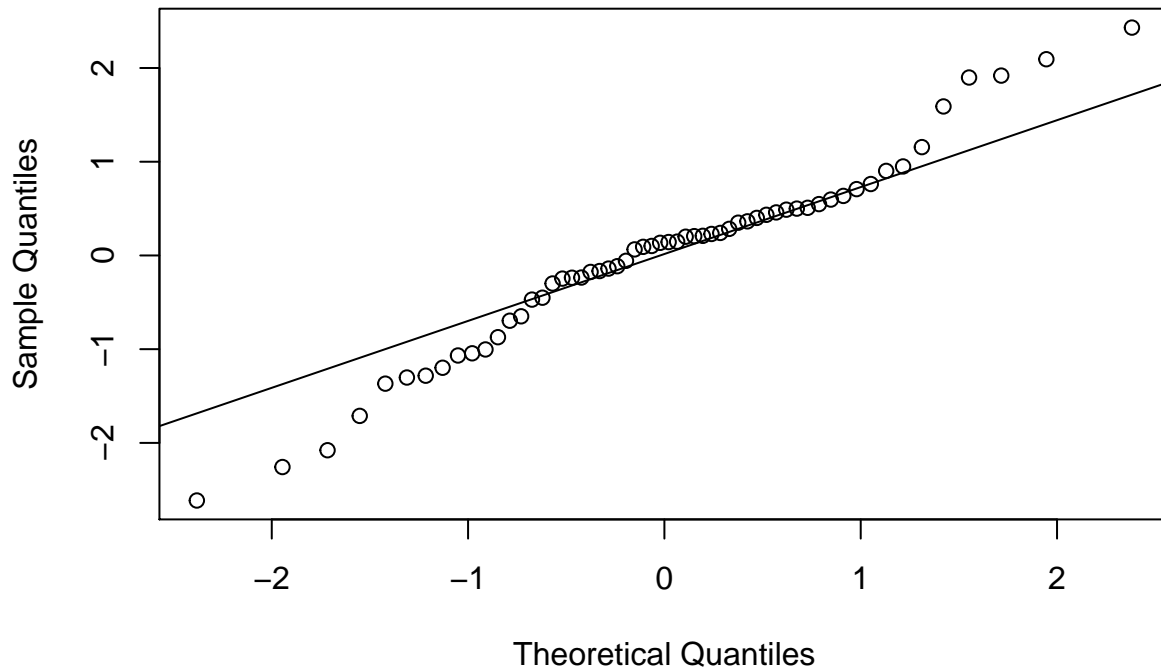
Load the `infectionrisk` data. Select only hospitals in regions 1 or 2. Fit a simple linear regression model with $y = \text{InfctRsk}$ and $x = \text{Stay}$. Display a normal probability plot of the residuals.

```
infectionrisk <- read.table("../Data/infectionrisk.txt", header=T)
infectionrisk <- infectionrisk[infectionrisk$Region==1 | infectionrisk$Region==2, ]
```

```
attach(infectionrisk)

model <- lm(InfctRsk ~ Stay)

qqnorm(residuals(model), main="")
qqline(residuals(model))
```



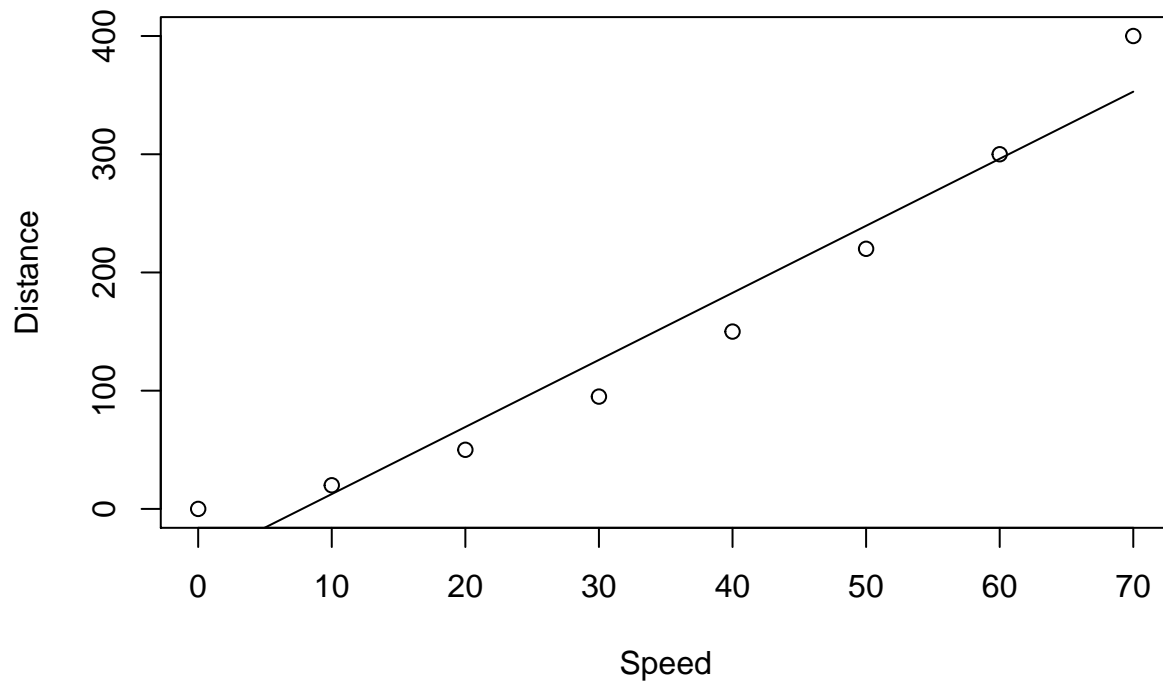
```
detach(infectionrisk)
```

Car stopping distances

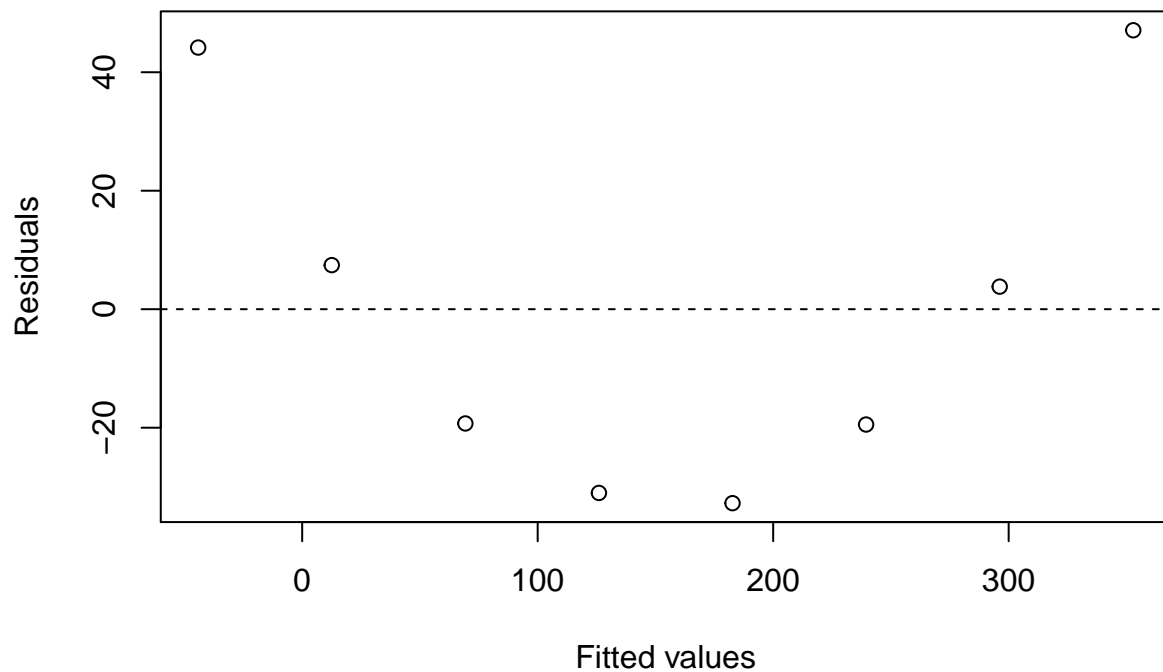
Load the carstopping data. Fit a simple linear regression model with $y = \text{Distance}$ and $x = \text{Speed}$. Display a scatterplot of the data with the simple linear regression line. Display a residual plot with fitted values on the horizontal axis. Create a new response variable equal to $\sqrt{\text{Distance}}$. Fit a simple linear regression model with $y = \sqrt{\text{Distance}}$ and $x = \text{Speed}$. Display a scatterplot of the data with the simple linear regression line. Display a residual plot with fitted values on the horizontal axis. Use the model to predict Distance for Speed = 10, 20, 30, and 40.

```
carstopping <- read.table("../Data/carstopping.txt", header=T)
attach(carstopping)

model <- lm(Distance ~ Speed)
plot(x=Speed, y=Distance,
     panel.last = lines(sort(Speed), fitted(model)[order(Speed)]))
```

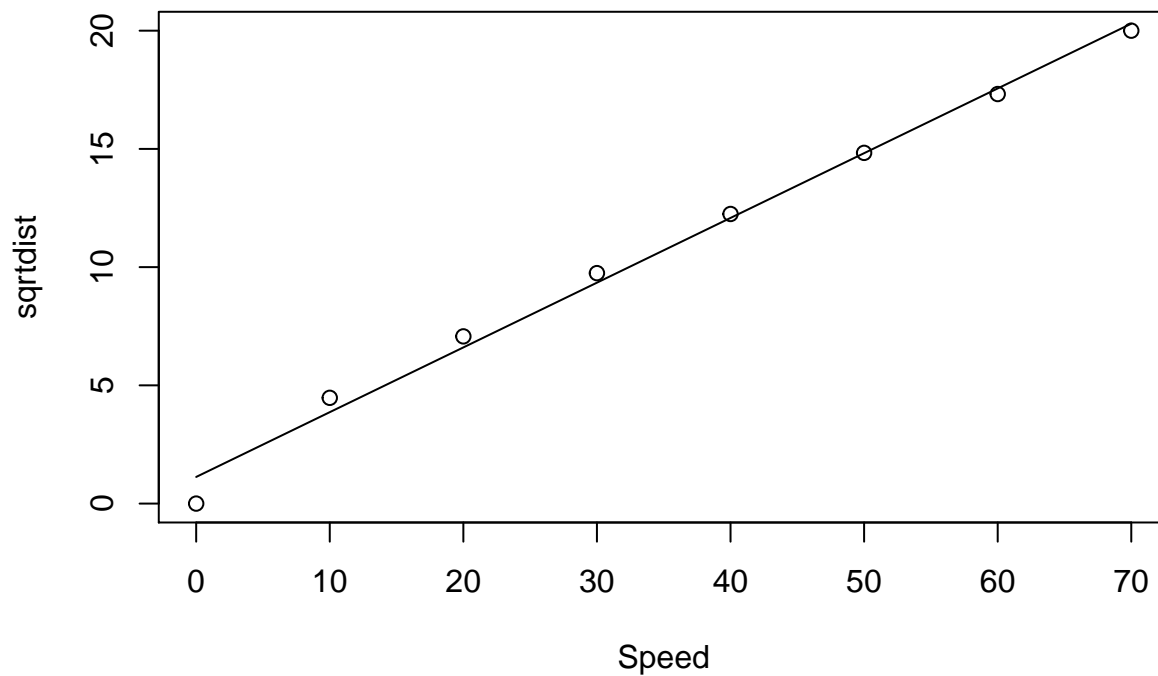



```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```

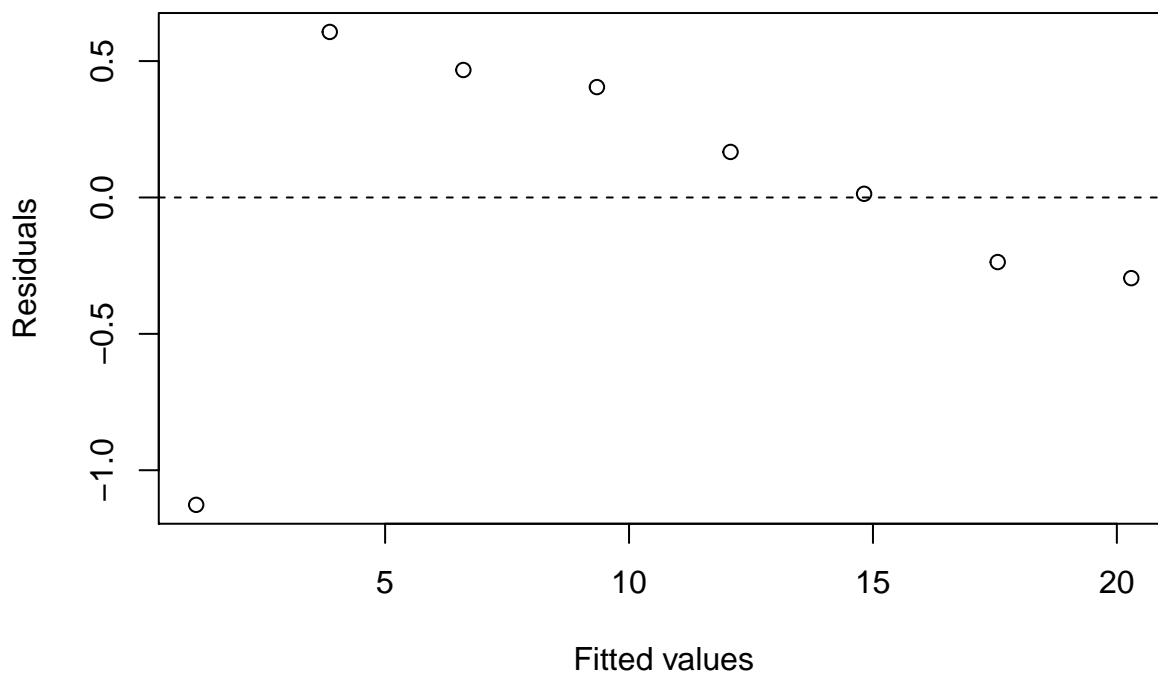


```
sqrtldist <- sqrt(Distance)

model <- lm(sqrtldist ~ Speed)
plot(x=Speed, y=sqrtldist,
     panel.last = lines(sort(Speed), fitted(model)[order(Speed)]))
```



```
plot(x=fitted(model), y=residuals(model),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```



```
predict(model, interval="prediction",
        newdata=data.frame(Speed=c(10, 20, 30, 40)))^2
```

```
##      fit      lwr      upr
## 1 14.94029  4.822718 30.63031
## 2 43.60856 24.980919 67.39202
## 3 87.27462 60.361600 119.13514
```

```
## 4 145.93846 110.411414 186.41301
```

```
#      fit      lwr      upr  
# 1 11.86090  3.93973 24.03997  
# 2 35.63671 20.42935 55.04771  
# 3 72.17067 49.44080 99.18664  
# 4 121.46277 90.63292 156.79793
```

```
detach(carstopping)
```