Simple.Linear.Regression.Tutorial.R

vkoelling

Thu Jul 27 14:40:06 2017

```
# Vanessa Koelling, July 6, 2017. Simple linear regression example.

# needed libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

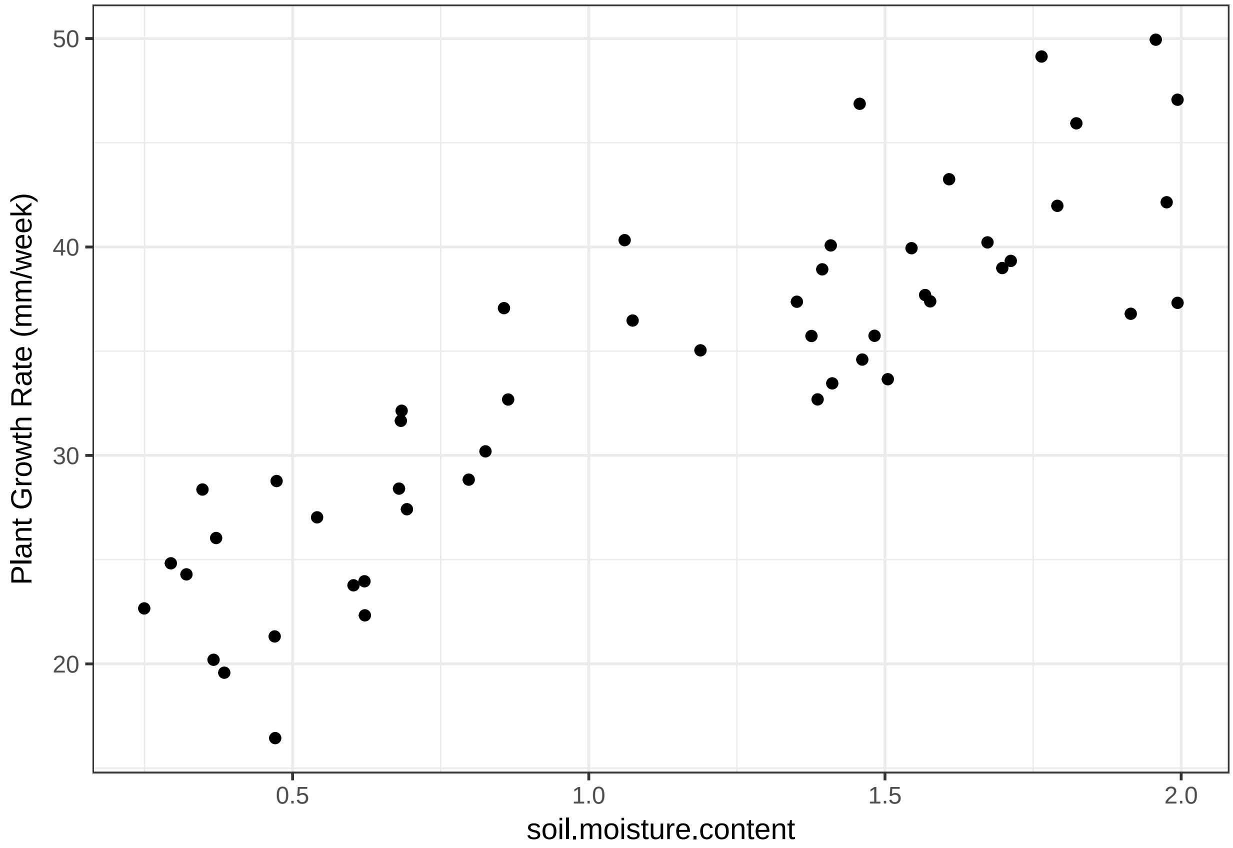
```
library(ggplot2)
library(readr)
library(ggfortify)

# clear the decks
rm(list = ls())

# import the data frame
plant_growth_rate <- read.csv("~/Desktop/R_Practice_Files/datasets/plant.growth.rate.csv")
# look at the data
glimpse(plant_growth_rate)
```

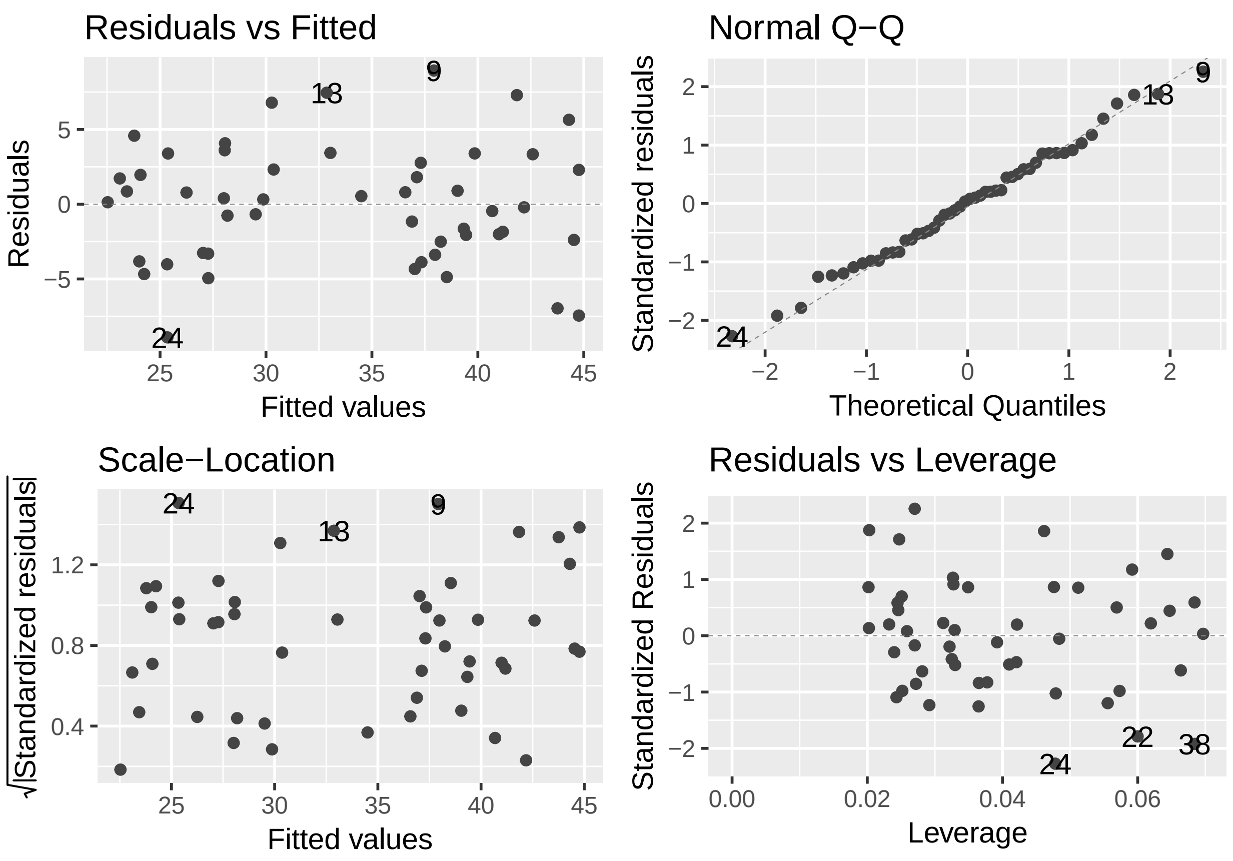
```
## Observations: 50
## Variables: 2
## $ soil.moisture.content <dbl> 0.4696876, 0.5413106, 1.6979915, 0.82557...
## $ plant.growth.rate     <dbl> 21.31695, 27.03072, 38.98937, 30.19529, ...
```

```
# explore the data in a scatter plot
ggplot(plant_growth_rate, aes(x = soil.moisture.content, y = plant.growth.rate)) + geom_point() + ylab(
```



```r
# fit the general linear model
model_pgr <- lm(plant.growth.rate ~ soil.moisture.content, data = plant_growth_rate)

# produces four plots critical to evaluating your data analysis
# 1) residuals vs. fitted: evaluates whether or not a line is appropriate to fit to the data
# 2) normal Q-Q: evaluates the assumption of normality of the residuals
# 3) scale-location: evaluates the assumption of equal variance
# 4) residuals vs. leverage: evaluates leverage to detect outliers and influential data points
autoplot(model_pgr, smooth.colour = NA) # the smooth.colour = NA argument eliminates unnecessary lines
```



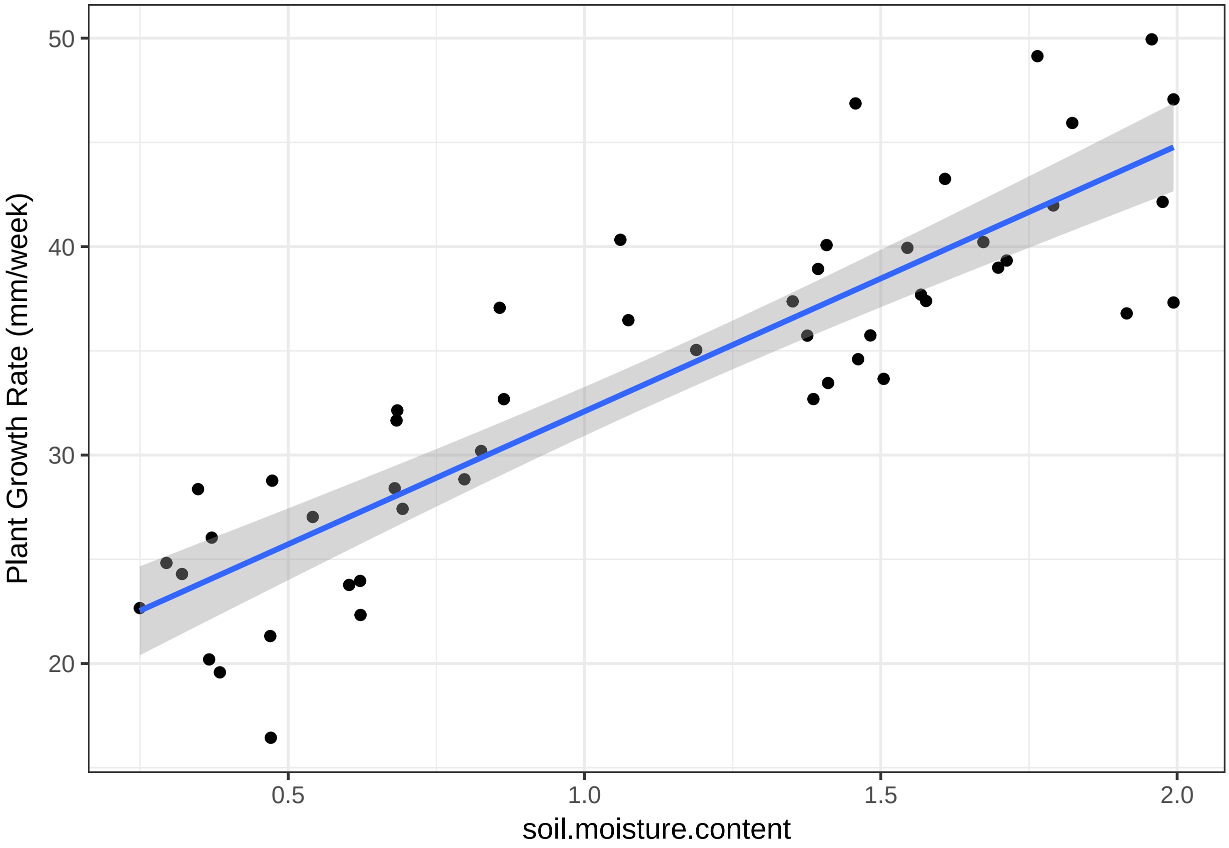
```
# calculate the sums-of-squares table
anova(model_pgr)
```

```
## Analysis of Variance Table
##
## Response: plant.growth.rate
##                       Df  Sum Sq Mean Sq F value    Pr(>F)
## soil.moisture.content  1 2521.15 2521.15  156.08 < 2.2e-16 ***
## Residuals             48  775.35   16.15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# calculate the y-intercept and slope of the regression line
summary(model_pgr)
```

```
##
## Call:
## lm(formula = plant.growth.rate ~ soil.moisture.content, data = plant_growth_rate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9089 -3.0747  0.2261  2.6567  8.9406
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             19.348      1.283   15.08   <2e-16 ***
## soil.moisture.content   12.750      1.021   12.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.019 on 48 degrees of freedom
## Multiple R-squared:  0.7648, Adjusted R-squared:  0.7599
## F-statistic: 156.1 on 1 and 48 DF,  p-value: < 2.2e-16
```

```
# produce a scatterplot with the regression line
# the geom_smooth(method = 'lm') adds the regression line; not appropriate for more complicated models
ggplot(plant_growth_rate, aes(x = soil.moisture.content, y = plant.growth.rate)) + geom_point() + geom_
```


Plant Growth Rate (mm/week)
soil.moisture.content