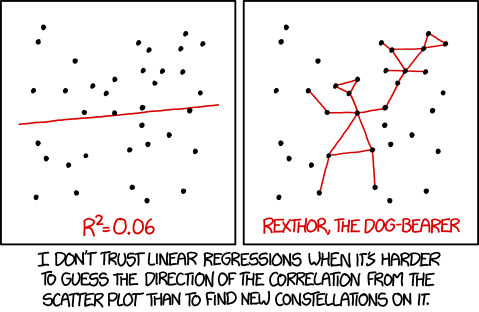
GLAMM Week 1

#### Devin Gamble, Raine Detmer, Ruby Harris-Gavin

#### 7/14/2020

Linear Models: Regression and Analysis of Variance (ANOVA)



This R-Markdown file serves as a practical companion to the GLAMM Week 1 slides on linear models, regression, and ANOVAs. Provided below is R code demonstrating how to build, analyze, and visualize linear models. We encourage you to work through these examples at your own pace and play around with the code to understand how it works! We assume that you are familiar with base-R language and have at least some familiarity with Tidyverse.

Contents

* Loading Packages & Data
* Linear Regression
* Assumptions of Linear Regression
* ANOVA
* Other Fun Stuff
* Try it out!

Load Data & Packages

library(tidyverse) #several tidy-friendly packages, including ggplot2  
library(janitor) #tools for formatting dataframe column names  
library(car) #Companion to applied regression package

Our Data

We’ll be working with the built-in ‘iris’ dataset. R. A. Fisher’s (& Edgar Anderson’s) ‘iris’ data includes measurements (in cm) of 50 *Iris* flowers for each of three species (N = 150) ([wiki page](https://en.wikipedia.org/wiki/Iris_flower_data_set)).

It has five variables: sepal\_length, sepal\_width, petal\_length, petal\_width, and species. The measurement variables are all numeric and the species variable is a factor.

#loads the data (built into R) as a dataframe  
iris <- iris %>%   
 clean\_names() #cleans up column names (needs 'janitor' package)

Run This line to view the data in RStudio:

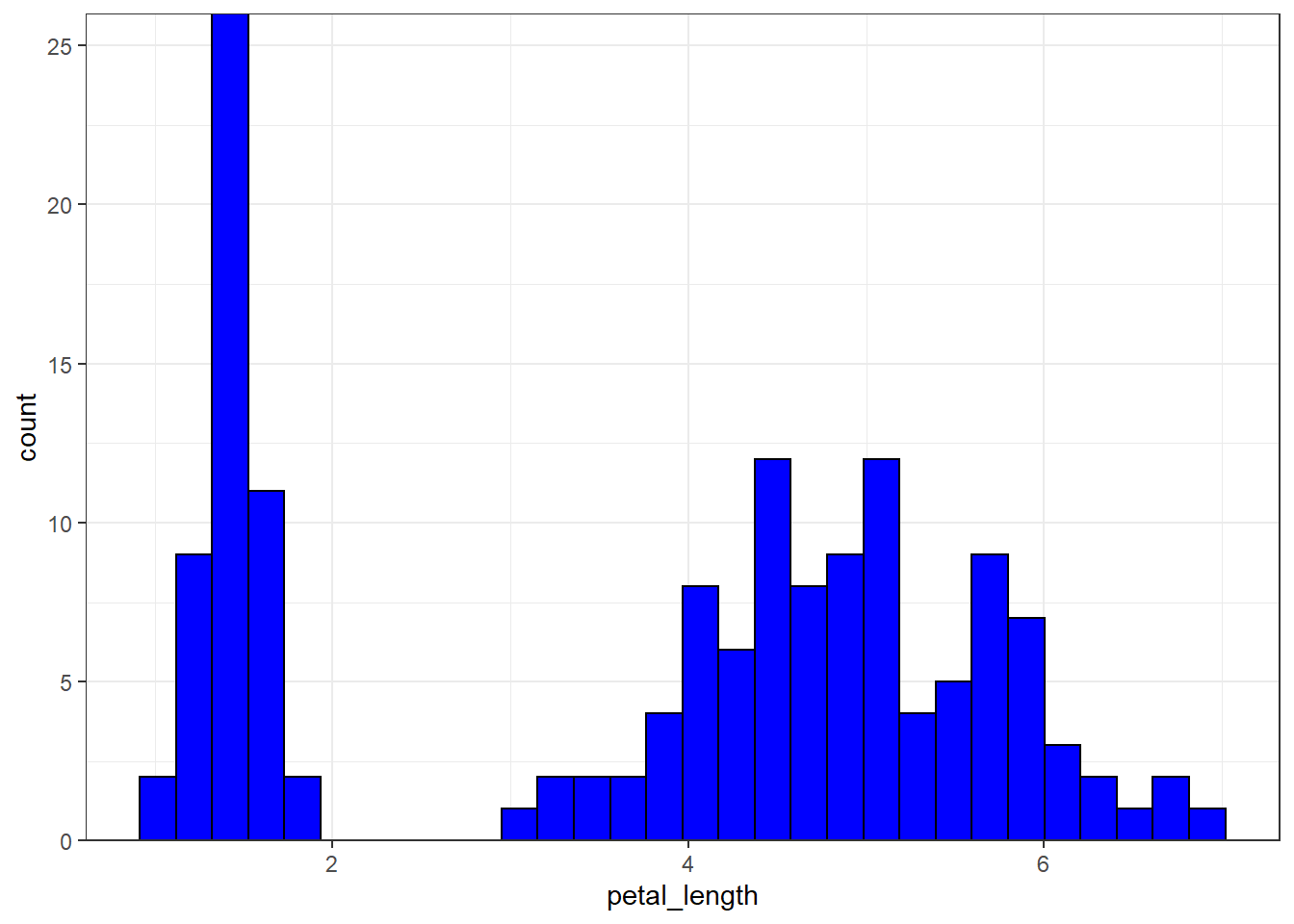
View(iris)

It’s usually a good idea to take a look at the structure of a dataset before jumping into analyses. Here are just a few things you can check out:

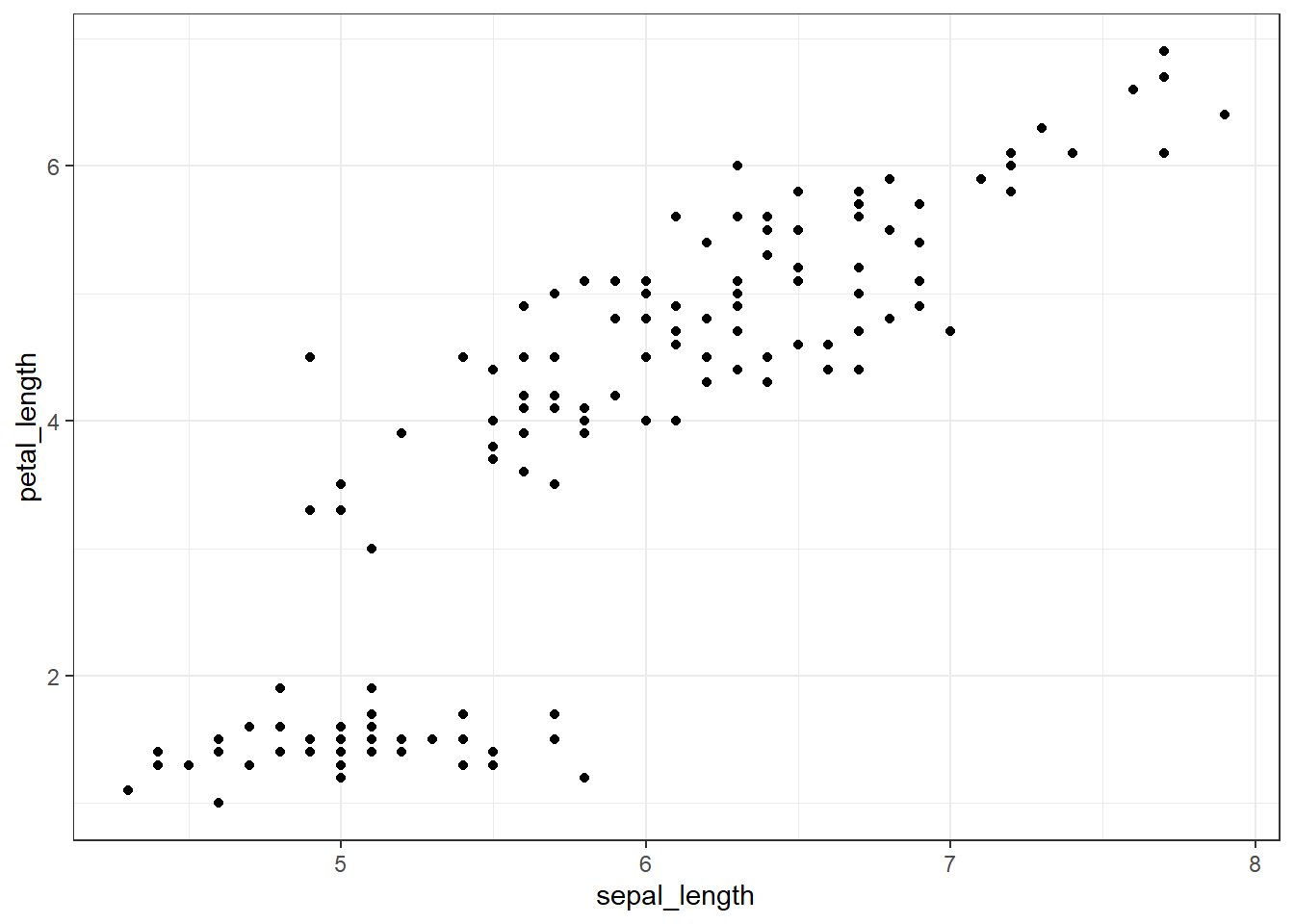
#Summary statistics  
summary(iris)

## sepal\_length sepal\_width petal\_length petal\_width   
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100   
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300   
## Median :5.800 Median :3.000 Median :4.350 Median :1.300   
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199   
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800   
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500   
## species   
## setosa :50   
## versicolor:50   
## virginica :50   
##   
##   
##

#Histograms for the distribution of numeric (continuous)` variables   
ggplot(data = iris, aes(x = petal\_length)) +  
 geom\_histogram(bins = 30, fill = "blue", color = "black") +   
 scale\_y\_continuous(expand = c(0,0)) +  
 theme\_bw()

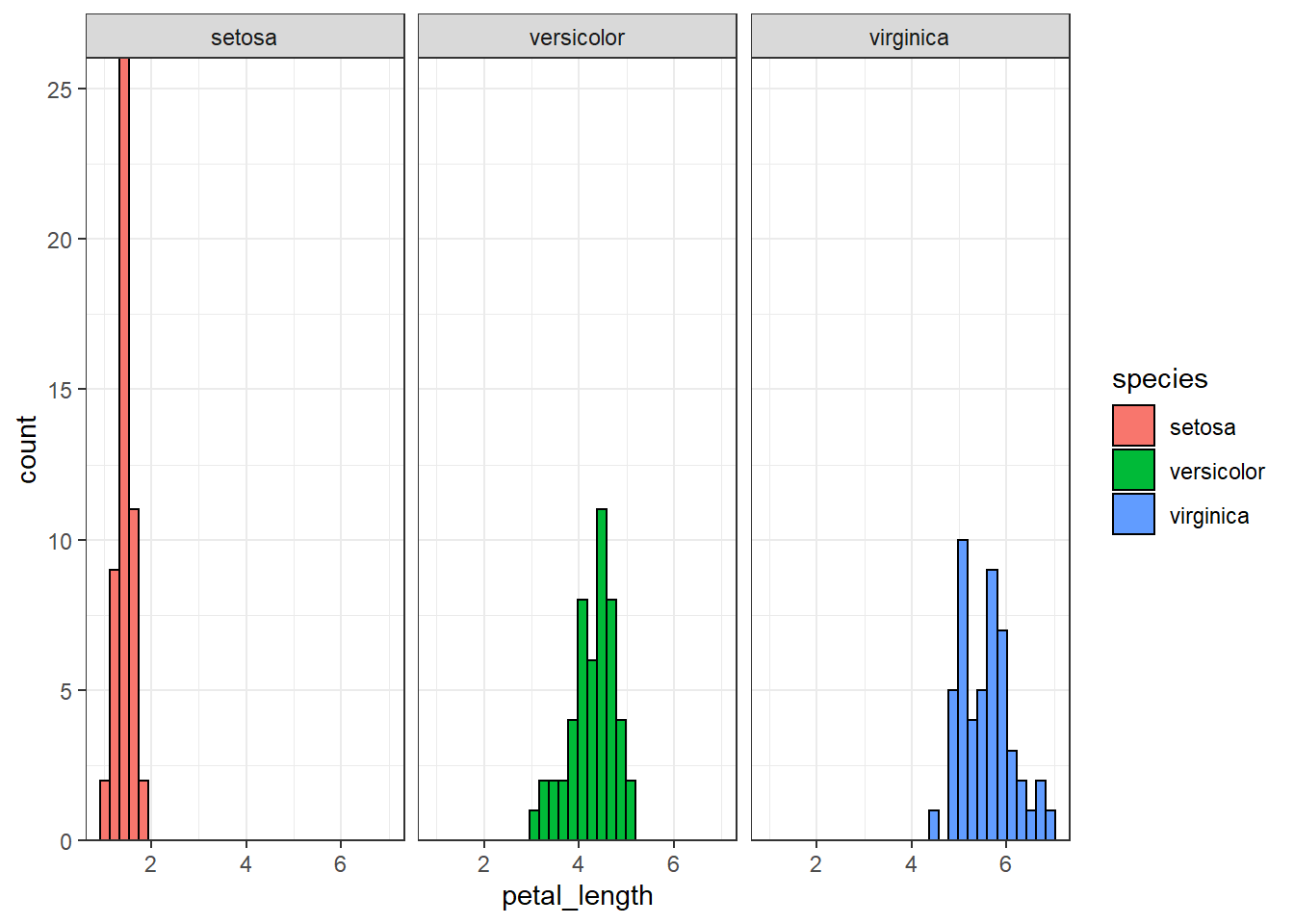


#Scatter plots - relationship between two numeric variables  
ggplot(data = iris, aes(x = sepal\_length, y = petal\_length)) +   
 geom\_point() +  
 theme\_bw()

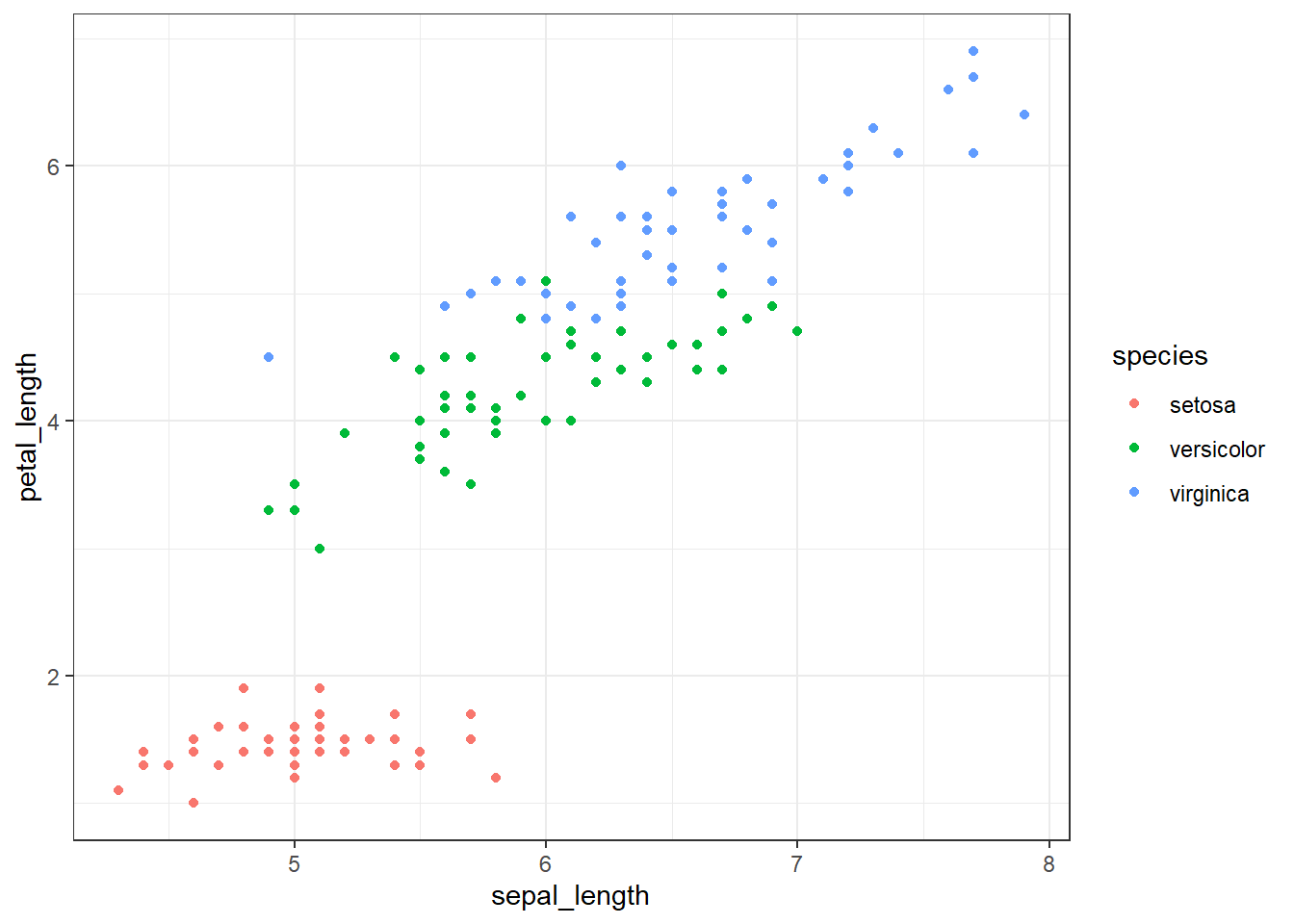


We can also plot these by species

ggplot(data = iris, aes(x = petal\_length, fill = species)) +  
 geom\_histogram(bins = 30, color = "black") +   
 scale\_y\_continuous(expand = c(0,0)) +  
 theme\_bw() +  
 facet\_wrap(~species) #splits up histograms by species



ggplot(data = iris, aes(x = sepal\_length, y = petal\_length)) +   
 geom\_point(aes(color = species), size = 1.5) +  
 theme\_bw()



*Note*: You can also use base-R functions like hist() and plot() or pairs() for these types of plots.

hist(iris$sepal\_length)  
  
plot(iris$sepal\_length, iris$petal\_length)  
  
pairs(iris[,1:4]) #Look at scatterplots for all variables [species excluded here]

Simple (OLS) Linear Regression

Say we want to understand whether a relationship exists between *sepal\_length* and *petal\_length*. We can build a linear model to do so!

iris\_lm1 <- lm(petal\_length ~ sepal\_length, data = iris)

The lm() function fits a linear model for us, where *petal\_length* is the dependent (response) variable and *sepal\_length* is the independent (explanatory) variable, or covariate.

This simple linear model is described by the formula:

\(Y\_i = \beta\_0 + \beta\_1 X\_i + \epsilon\_i\)         where \(\epsilon\_i\) ~ \(N(0, \sigma^2)\)

\(Y\) is our response variable, *petal\_length*,

\(X\) is our explanatory variable (or covariate), *sepal\_length*,

\(\beta\_0\) is the intercept,

\(\beta\_1\) is the parameter coefficient or slope, which describes the relationship between our explanatory and response variables.

\(\epsilon\) is the random error (residuals) term, which accounts for differences between observed and predicted \(Y\) values. \(\epsilon\) is assumed to be Normally distributed with mean 0 and variance \(\sigma^2\).

By fitting our data to a linear model, we are attempting to estimate two unknowns: the intercept \(\beta\_0\) and the coefficient \(\beta\_1\). Using linear regression, we can examine how the value of *petal\_length* depends on the value of *sepal\_length*.

The summary() function gives us an output table for a linear regression using the lm formula we made.

summary(iris\_lm1)

##   
## Call:  
## lm(formula = petal\_length ~ sepal\_length, data = iris)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.47747 -0.59072 -0.00668 0.60484 2.49512   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -7.10144 0.50666 -14.02 <2e-16 \*\*\*  
## sepal\_length 1.85843 0.08586 21.65 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8678 on 148 degrees of freedom  
## Multiple R-squared: 0.76, Adjusted R-squared: 0.7583   
## F-statistic: 468.6 on 1 and 148 DF, p-value: < 2.2e-16

This output gives:

- Summary statistics on the residuals of our model.

- Coefficients for our model’s intercept and slope: the parameter estimate, std error, t-value, and p-value.

- Residual standard error and degrees of freedom

- The \(R^2\) value, F-statistic & p-value for the overall model

Interpretation

*Intercept Estimate*: the average value of petal\_length when sepal\_length is 0. (Sometimes uninformative)

*Parameter Estimate* for sepal\_length, \(\hat\beta\_1\): for every 1 cm increase in sepal\_length, we can expect a 1.85843 cm increase in petal\_length. Since *p* < 0.05, we can say there is a significant effect of sepal\_length on petal\_length.

*Residual Standard Error*: the average amount the response variable deviates from the regression line, based on the degrees of freedom.

*The* \(R^2\) value (coeff. of determination) measures how well the model fits the data. Here, an \(R^2\) of 0.76 tells us that sepal\_length (or our model) explains 76% of variance in petal\_length.

*The F-statistic* and its p-value comes from an F-test that evaluates the overall fit of our model against a hypothetical null model. A large F & low p-value indicate that our model is significant (a relationship exists between \(X\) and \(Y\)). The F-statistic increases with the # of datapoints used.

Note: Magnitudes of the \(R^2\) and F-statitics are relative to the data and the models being fit.

More things

We can also extract certain terms of the output

summary(iris\_lm1)$r.squared

## [1] 0.7599546

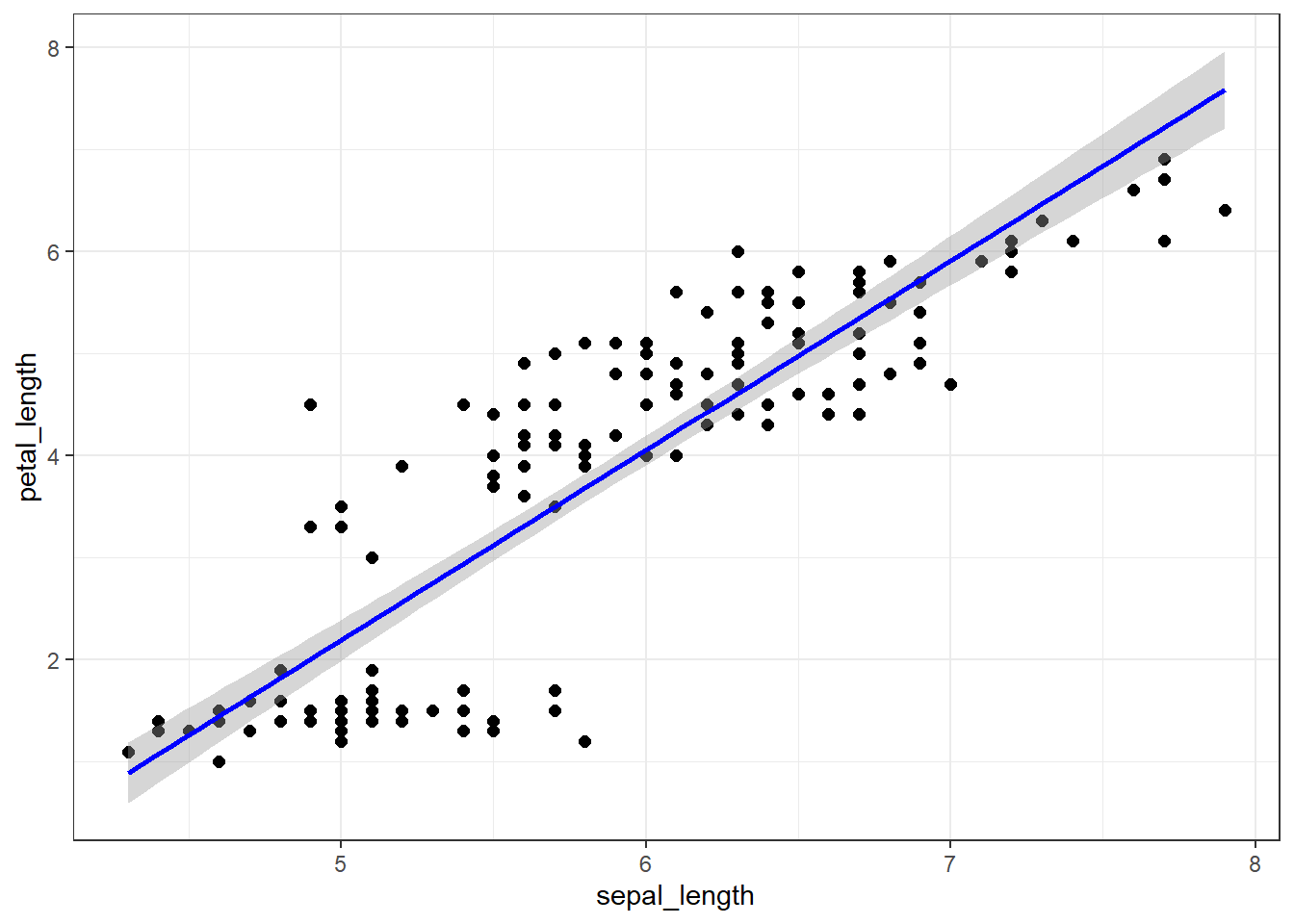
summary(iris\_lm1)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -7.101443 0.50666229 -14.01613 6.133586e-29  
## sepal\_length 1.858433 0.08585565 21.64602 1.038667e-47

#See ?summary.lm for more

Visualizing our model

ggplot(data = iris, aes(y = petal\_length, x = sepal\_length)) +  
 geom\_point(size = 2) +   
 geom\_smooth(method = "lm", color = "blue") + #Add a regression line (method = "lm") with confidence interval  
 theme\_bw()



Multivariate Linear Regression

Multiple linear regression is just an extension of simple linear regression with two or more covariates:

\(Y\_i = \beta\_0 + \beta\_1 X\_{1i} + \beta\_2 X\_{2i} + \beta\_3 X\_{3i} + ... + \beta\_{ji} X\_{ji}+ \epsilon\_i\)       where \(\epsilon\_i\) ~ \(N(0, \sigma^2)\)

Now we are attempting to estimate the intercept \(\beta\_0\) and \(\beta\_{j}\) coefficients for each of \(j\) covaraites.

To see how *sepal\_length* and *petal\_width* influence *petal\_length*, we can fit another linear model.

iris\_lm2 <- lm(petal\_length ~ sepal\_length + petal\_width, data = iris)  
  
summary(iris\_lm2)

##   
## Call:  
## lm(formula = petal\_length ~ sepal\_length + petal\_width, data = iris)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.15506 -0.21920 -0.02115 0.25986 1.35204   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.50714 0.33696 -4.473 1.54e-05 \*\*\*  
## sepal\_length 0.54226 0.06934 7.820 9.41e-13 \*\*\*  
## petal\_width 1.74810 0.07533 23.205 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4032 on 147 degrees of freedom  
## Multiple R-squared: 0.9485, Adjusted R-squared: 0.9478   
## F-statistic: 1354 on 2 and 147 DF, p-value: < 2.2e-16

Try it out: How would you interpret the output of this multivariate linear regression?

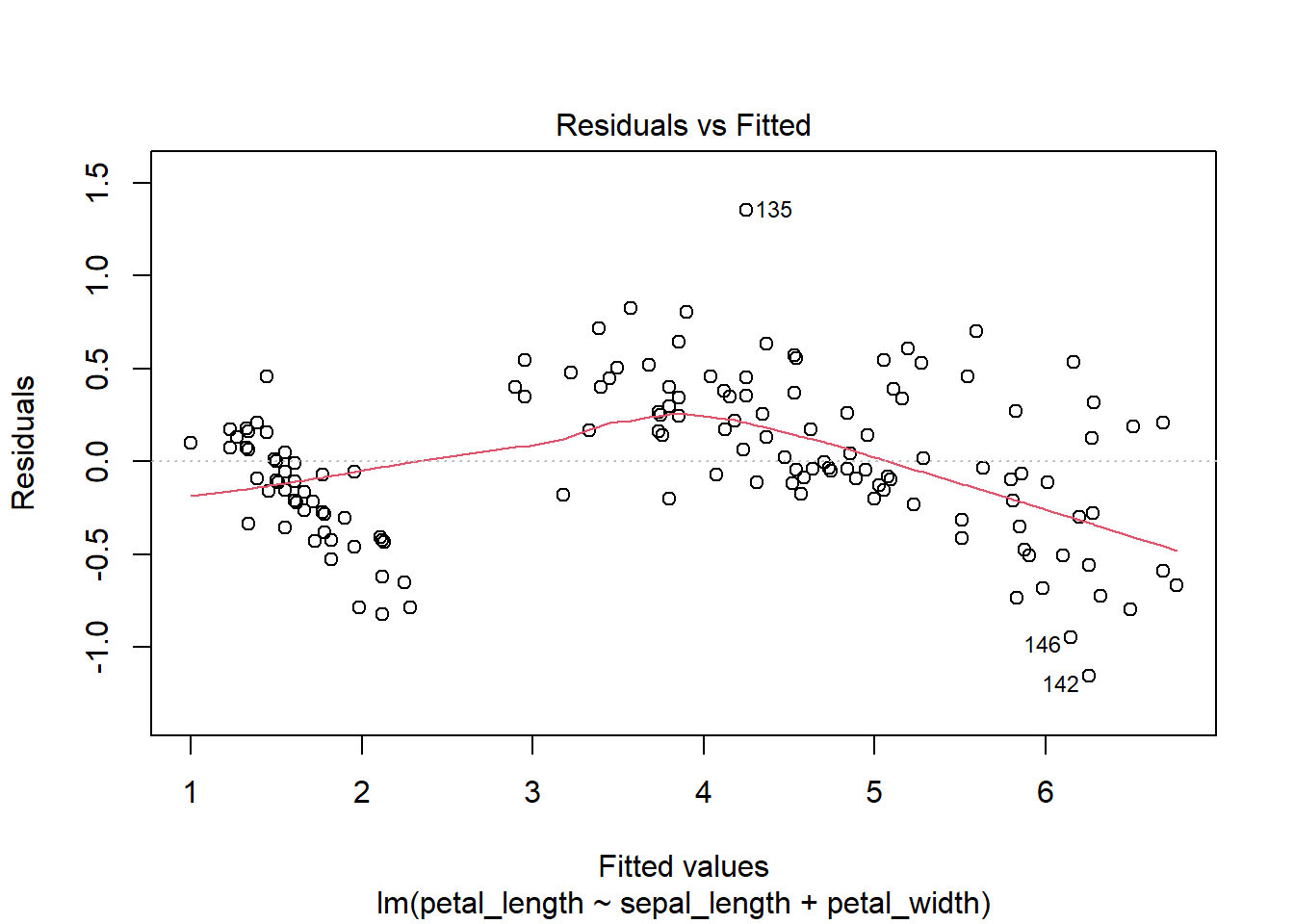
*Note*: The adjusted \(R^2\) adjusts for the number of variables in the model, whereas the regular \(R^2\) does not and increases with every covariate added to a model which may be problematic.

Assumptions of Linear Regression

Linear relationship between the covariates and response

We expect the relationship between the response variable and the covariate(s) to be linear. We can check this in scatterplots like those above, or by plotting our model’s residuals vs fitted \(Y\) values:

plot(iris\_lm2, 1)



The red line would be horizontal at 0 if there were no pattern, but it looks there may be some nonlinear trend in our model.

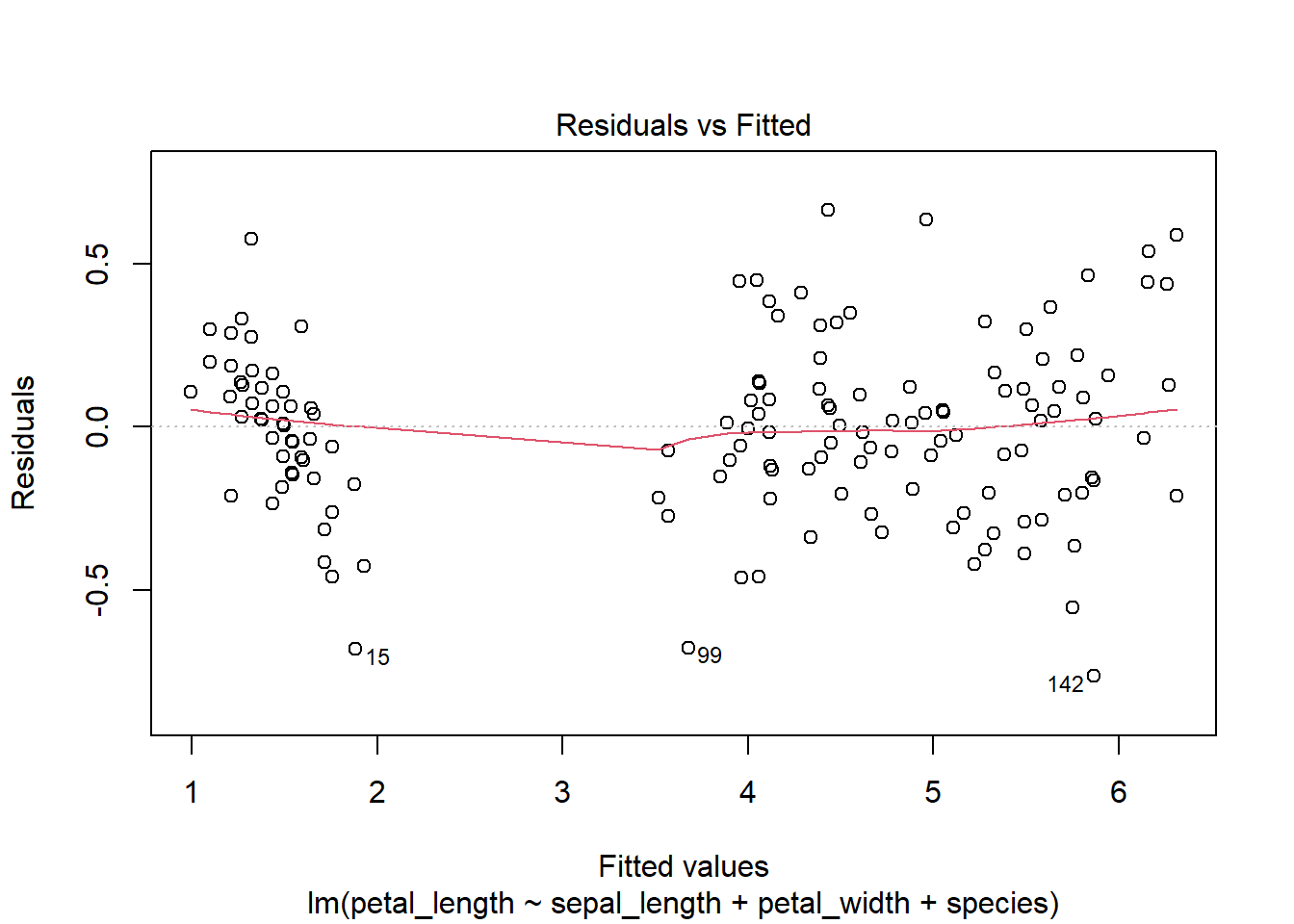
Fixes

- Add missing covariates

- Variable transformations (log(x), sqrt(x), x^2)

In this case, we can see if adding the *species* variable to our model helps explain the nonlinear pattern we see in petal\_length:

iris\_lm3 <- lm(petal\_length ~ sepal\_length + petal\_width + species, data = iris)  
  
#summary(iris\_lm3) #unhash for regression output  
  
plot(iris\_lm3, 1)

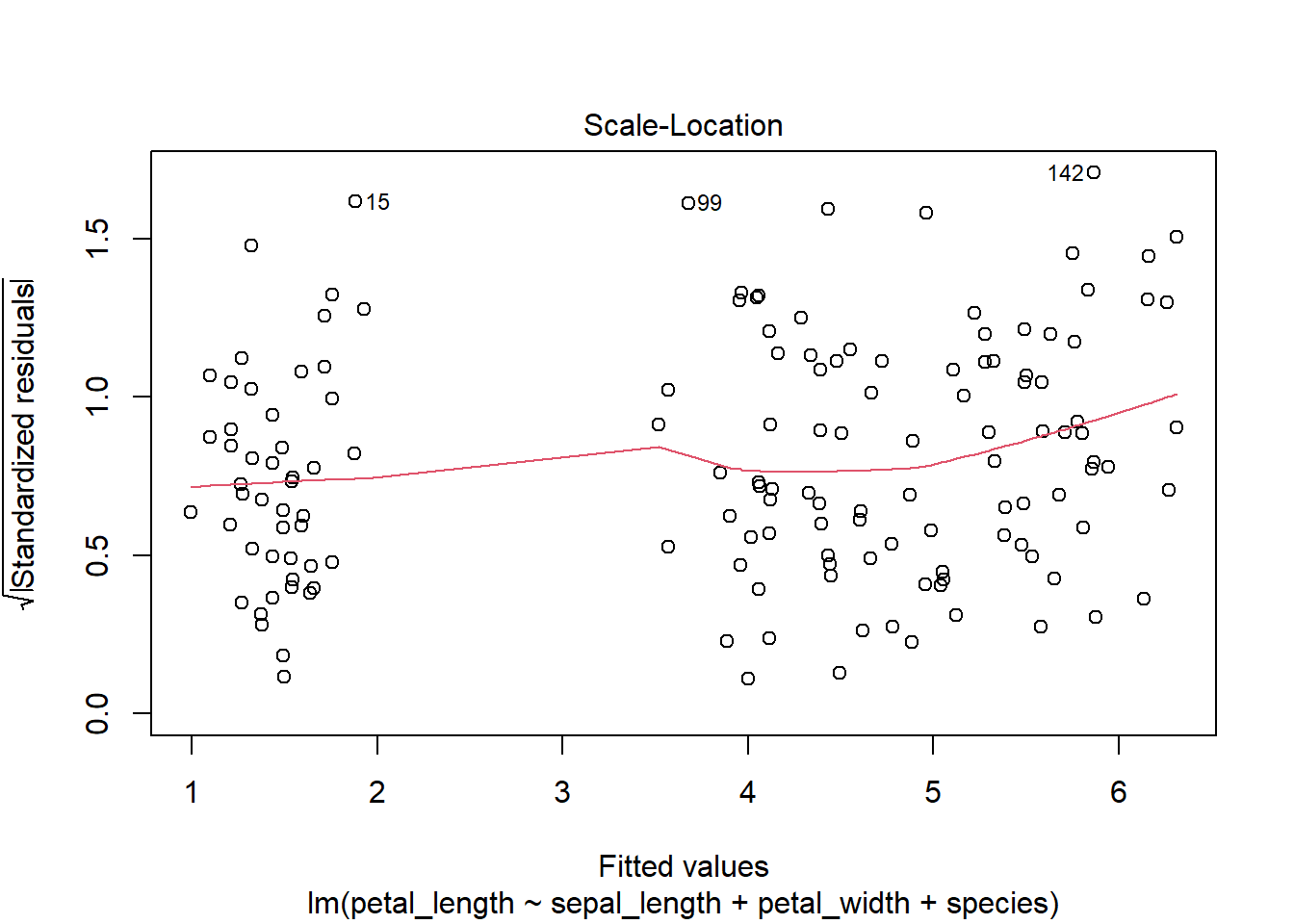


Much better, this looks like a linear relationship!

Homoscedasticity (Homogeneity) of Variance

We can check the assumption of equal variance by comparing the spread of residuals at different \(X\) values. Specificially, we can plot the model’s standardized residuals against the fitted values of \(Y\). The spread should be similar across all fitted values.

plot(iris\_lm3, 3)



The line is roughly horizontal with points spread similar distances from the line, suggesting that the variance of residuals across different values of \(X\) is fairly equal.

Fixes:

- Add explanatory covariates

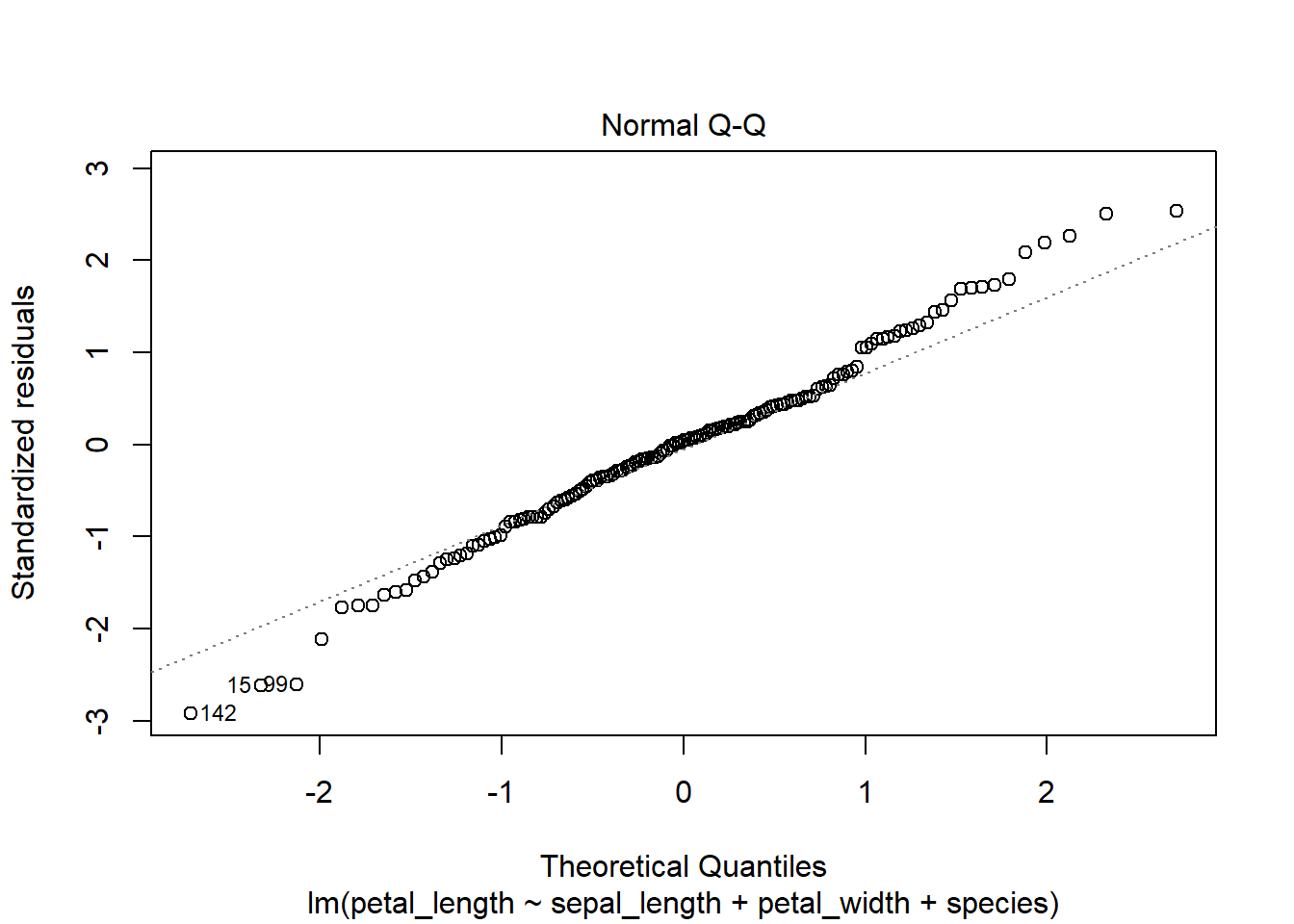
- Use a “mean-variance stabilizing” transformation on the response (log(y), sqrt(y))

Note: The lack of a pattern in model residuals against fitted values (as above) or against any covariate supports the assumption that residuals are independently distributed and uncorrelated.

Residuals are Normally distributed

We can check this with histograms (or density plots) and with Q-Q plots:

plot(iris\_lm3, 2)



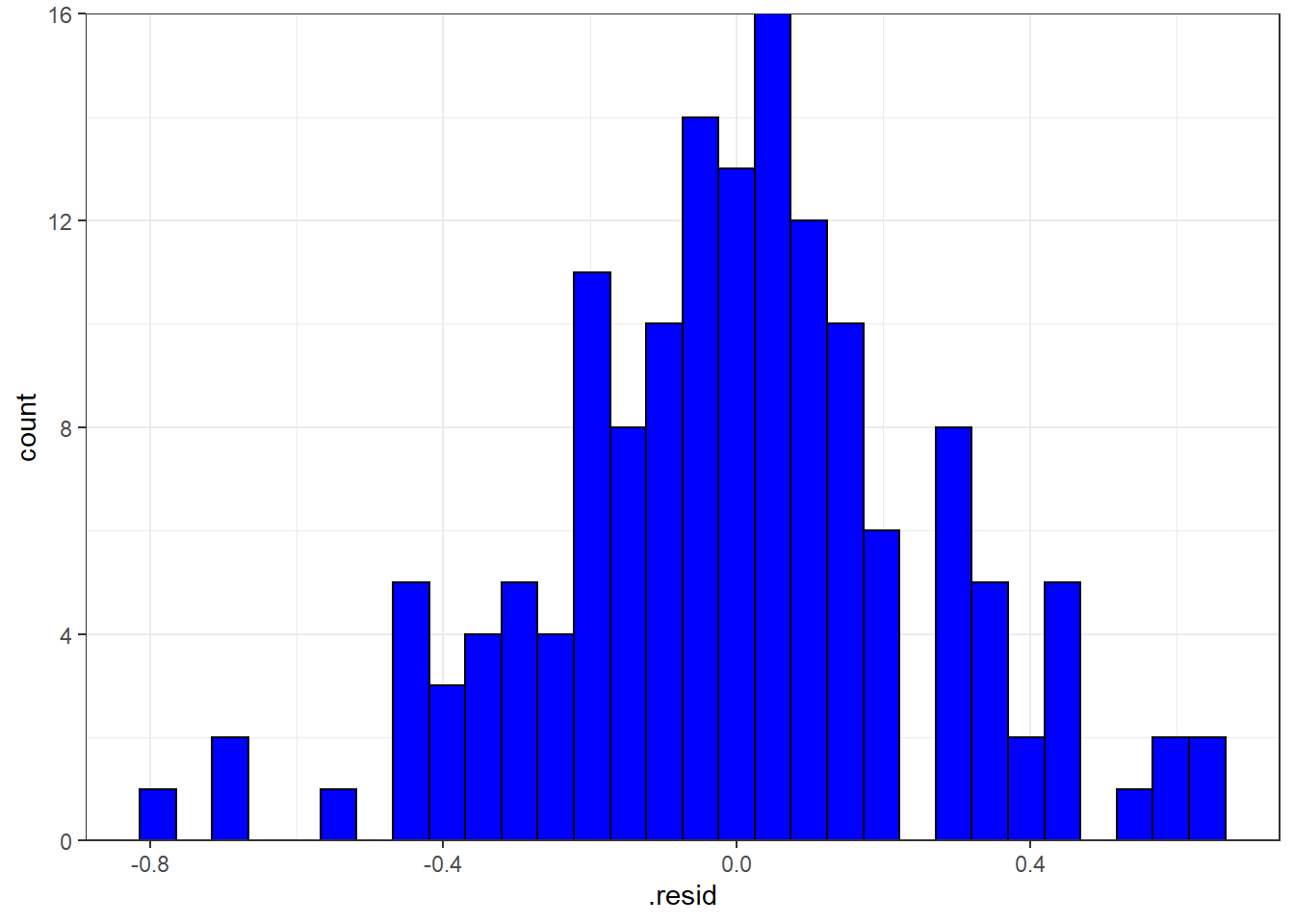
#The same plot is also produced when running the following two lines together  
qqnorm(iris\_lm3$residuals)  
qqline(iris\_lm3$residuals)

This plots the residuals (quantiles) from our model against theoretical (Normal) quantiles to assess normality. Since our points roughly stick to the 1:1 line, we can assume our residuals are normally distributed. Small deviations from normality are usually not a problem.

*Note*: A model’s residuals will usually be normally distributed when the \(Y\) variable is normally distributed. A ‘U’ shaped Q-Q plot indicates that the data is right (positively) or left (negatively)-skewed. An ‘S’ shape can indicate that the data is under or over-dispersed.

We can also check a histogram of the residuals.

#Checking the distribution of residuals with a histogram  
ggplot(data = iris\_lm3, aes(x = .resid)) +   
 geom\_histogram(fill = "blue", color = "black") +  
 scale\_y\_continuous(expand = c(0,0)) +  
 theme\_bw()



Look’s pretty good!

Fixes:

- Transform problematic covariates

- Add in missing covarites that may explain more variation in the response

- Non-normal modeling approaches

Note: Another option is the Shapiro-Wilk’s test of Normality:

shapiro.test(iris\_lm3$residuals)  
#If p > 0.5, we can assume normality (distribution of residuals not different from normal distribution)

No Multicollinearity

Covaraites in a linear model should not be correlated with one another. We can check this by looking at correlations or at VIFs (variance inflation factors) among covariates:

cor(iris[,1:4]) #excludes species column

## sepal\_length sepal\_width petal\_length petal\_width  
## sepal\_length 1.0000000 -0.1175698 0.8717538 0.8179411  
## sepal\_width -0.1175698 1.0000000 -0.4284401 -0.3661259  
## petal\_length 0.8717538 -0.4284401 1.0000000 0.9628654  
## petal\_width 0.8179411 -0.3661259 0.9628654 1.0000000

vif(iris\_lm2) #model without species variable

## sepal\_length petal\_width   
## 3.021401 3.021401

vif(iris\_lm3) #model with species

## GVIF Df GVIF^(1/(2\*Df))  
## sepal\_length 3.024496 1 1.739108  
## petal\_width 16.215838 1 4.026889  
## species 14.075725 2 1.936947

VIF values > 2 (or > 4 according to some) usually indicate some multicollinearity, such that two variables may be highly related and redundant in explaining the response variable. This can lead to erratic changes in coefficient estimates in regression. Including *petal\_width* could be a problem in our model.

Fixes

- Remove covariates with the highest VIFs

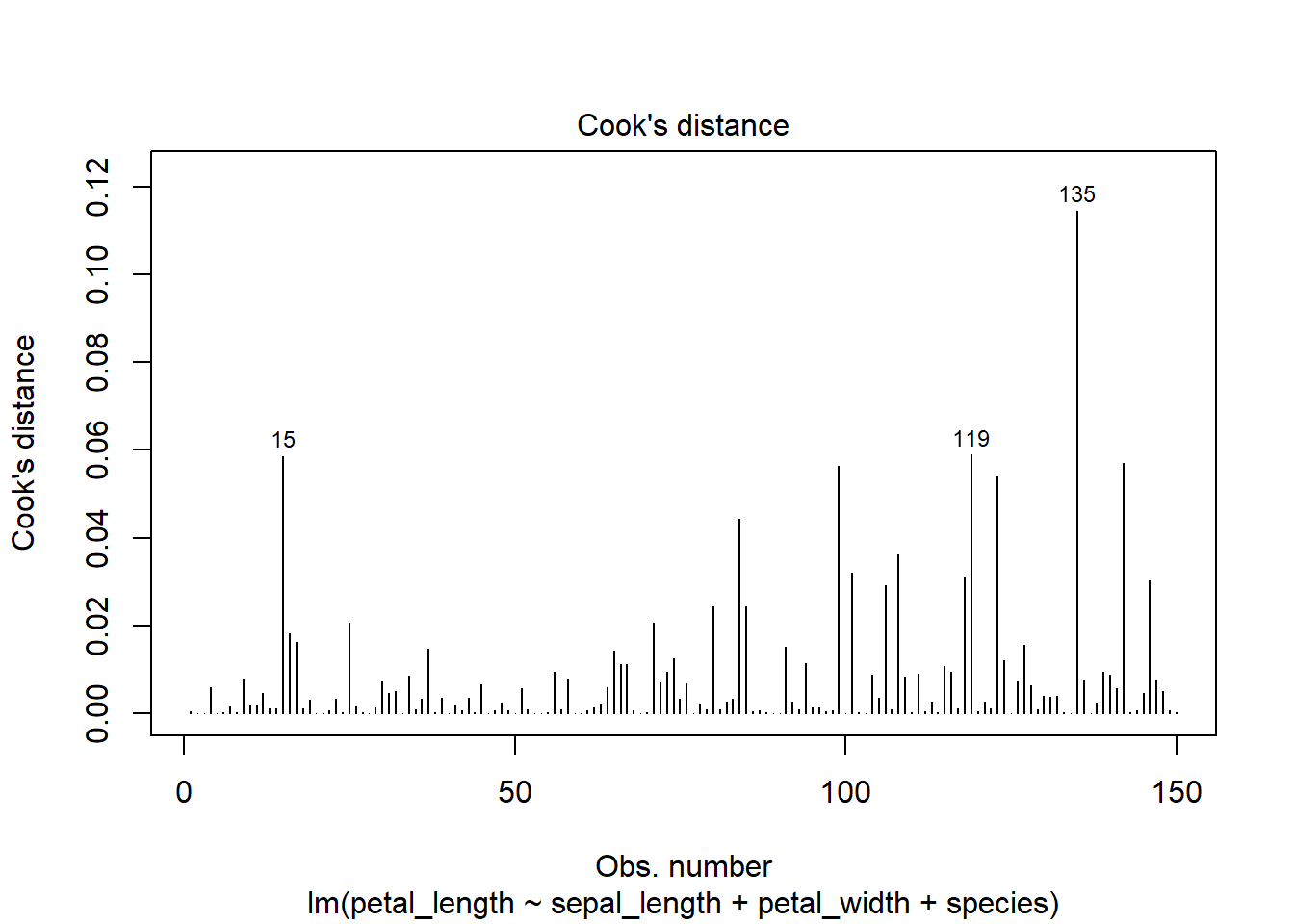
- Choose fewer highly correlated pairs of covariates

Other things to check

Influential Observations/Outliers

We typically don’t expect any single observations to have significant influence on the relationship between covariates and the response variable. We can use a plot of “Cook’s Distance” to look for points/outliers that could have exaggerated influence on fitted \(Y\) values:

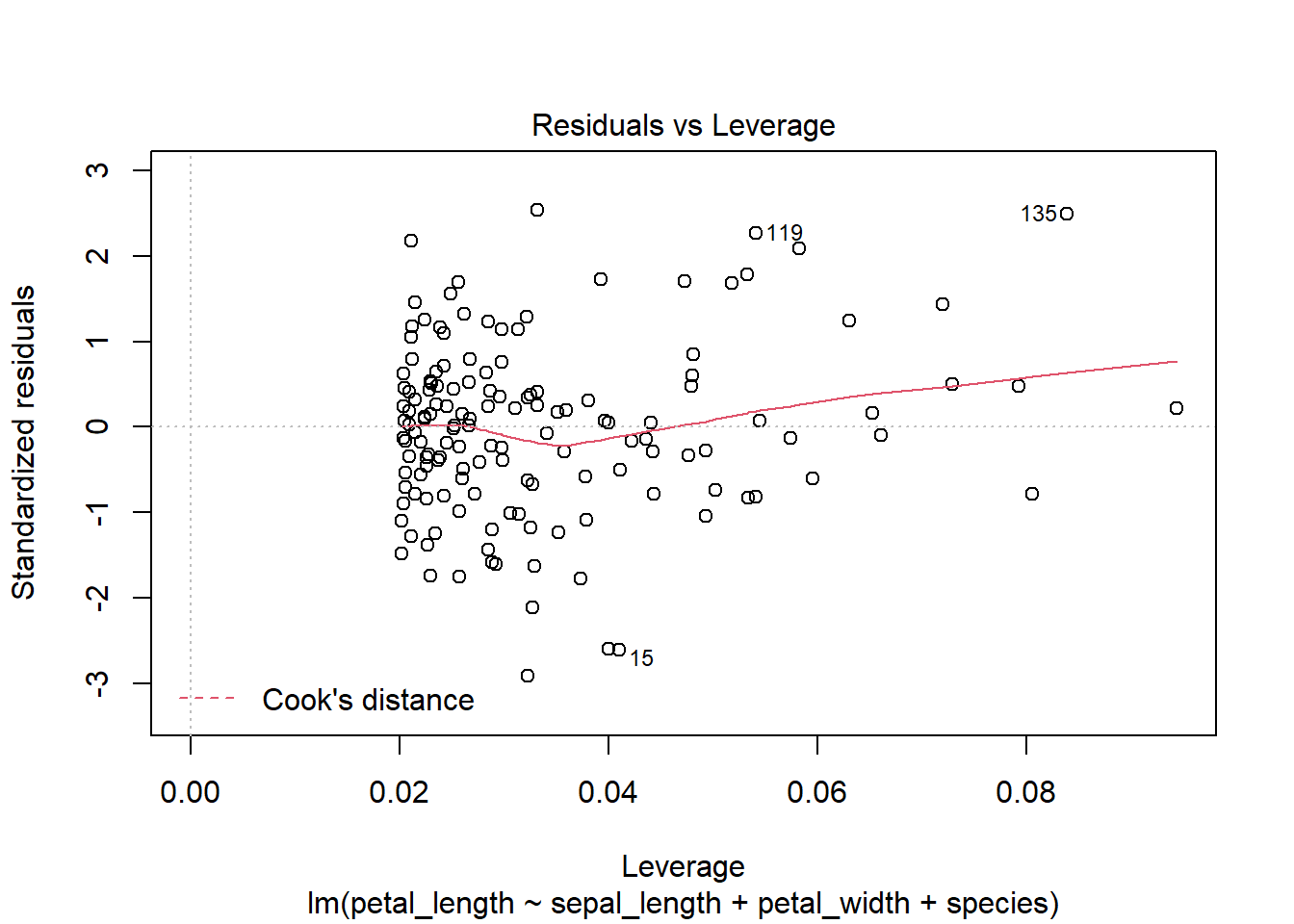
plot(iris\_lm3, 4)



Typically, an acceptable threshold for Cook’s D or influence is 4/N, so observations above .0266 (4/150) in our case may be influencing the fit of our model.

Plotting residuals vs leverage can also help to identify overly influential observations:

plot(iris\_lm3, 5)



There may be one or two observations of large influence (outliers and high leverage), but all points are within the mean Cook’s distance (red dashed lines - not shown) so they shouldn’t be problematic in our model.

Fixes

- Points of high influence/leverage can be removed from the data if they have large effects on the regression

No Autocorrelation

This is usually not a problem unless you are working with time-series data. Relates to the assumption that residuals are independent.

You can check for autocorrelation by plotting the residuals against residuals -1 (lag1) to look for a trend. You can also use the Durbin-Watson test. One possible fix is to add lag1 as a covariate to reduce the effects of autocorrelation in your model.

If you want to quickly check some of these linear regression assumptions, you can use plot(lm)

plot(iris\_lm1)

In Conclusion:

Normality, a linear relationship, Homoscedasticity, and absence of Multicollinearity (independence) are often the most important assumptions to check for. However, minor deviations from these assumptions usually don’t invalidate a model. Aside from the listed potential fixes to more serious violations, more complex modeling approaches (e.g. assuming non-normal distributions) can be useful.

Models with Interactions

In some cases, you may be interested in whether there may be non-independence among explanatory variables, which could interact to influence a response variable. We can include interactions in linear models by adding an interaction term (: instead of +, or a \* to specify an interaction term with main effects):

iris\_lm\_int <- lm(petal\_length ~ petal\_width\*sepal\_length, data = iris)  
  
summary(iris\_lm\_int)

##   
## Call:  
## lm(formula = petal\_length ~ petal\_width \* sepal\_length, data = iris)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.99588 -0.24329 0.00355 0.29735 1.24780   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.24804 0.59586 -5.451 2.08e-07 \*\*\*  
## petal\_width 2.97115 0.35836 8.291 6.74e-14 \*\*\*  
## sepal\_length 0.87551 0.11667 7.504 5.60e-12 \*\*\*  
## petal\_width:sepal\_length -0.22248 0.06384 -3.485 0.00065 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3888 on 146 degrees of freedom  
## Multiple R-squared: 0.9525, Adjusted R-squared: 0.9515   
## F-statistic: 975.4 on 3 and 146 DF, p-value: < 2.2e-16

A significant parameter estimate for an interaction term indicates that these two variables co-dependently influence the value of the response variable. Interpretations of this coefficient, as well as main effects, can be a little trickier and may be easier with some graphical methods.

ANOVA

Analysis of Variance (ANOVA) is a statistical analysis used to test whether three or more different groups have different means. Like regression, we can also model the effects some explanatory variable has on a response variable, only the explanatory variable is categorical, not numeric.

One-Way ANOVA

A one-way (single-factor) ANOVA tests whether differences in group means exist for a single factor (using an F-test). In the case of our *Iris* data, for example, we can test whether differences in petal\_length exist among the three species:

iris\_lm4 <- lm(petal\_length ~ species, data = iris)  
  
anova(iris\_lm4)

## Analysis of Variance Table  
##   
## Response: petal\_length  
## Df Sum Sq Mean Sq F value Pr(>F)   
## species 2 437.10 218.551 1180.2 < 2.2e-16 \*\*\*  
## Residuals 147 27.22 0.185   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#summary.aov(iris\_lm4)  
#anova() is the same as summary.aov(), but slightly different than the base-R aov()

The anova() function runs a one-way ANOVA on our linear model. The output provides the degrees of freedom, the sum of squares (SS) and the Mean Squares (MS) for the species factor and the residuals, as well as an F statistic and p-value for species.

A significant p-value indicates that *one or more* significant differences exist between group means.

When used on a linear model built for ANOVA, the summary() function provides an output table similar to those above with our linear regression models. The difference is that with factors/categorical variables, this output uses the intercept as the reference level (species 1) and tests other levels (species 2 & 3) against the reference level. (It does not test the significance of the species factor overall). These “treatment contrasts” are the default in R.

summary(iris\_lm4)

##   
## Call:  
## lm(formula = petal\_length ~ species, data = iris)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.260 -0.258 0.038 0.240 1.348   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.46200 0.06086 24.02 <2e-16 \*\*\*  
## speciesversicolor 2.79800 0.08607 32.51 <2e-16 \*\*\*  
## speciesvirginica 4.09000 0.08607 47.52 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4303 on 147 degrees of freedom  
## Multiple R-squared: 0.9414, Adjusted R-squared: 0.9406   
## F-statistic: 1180 on 2 and 147 DF, p-value: < 2.2e-16

Tukey’s Test

Since ANOVAs only test whether there a significant differences exists among groups, we can use a Tukey Honest Significant Difference (HSD) test to determine which group means are significantly different from one another:

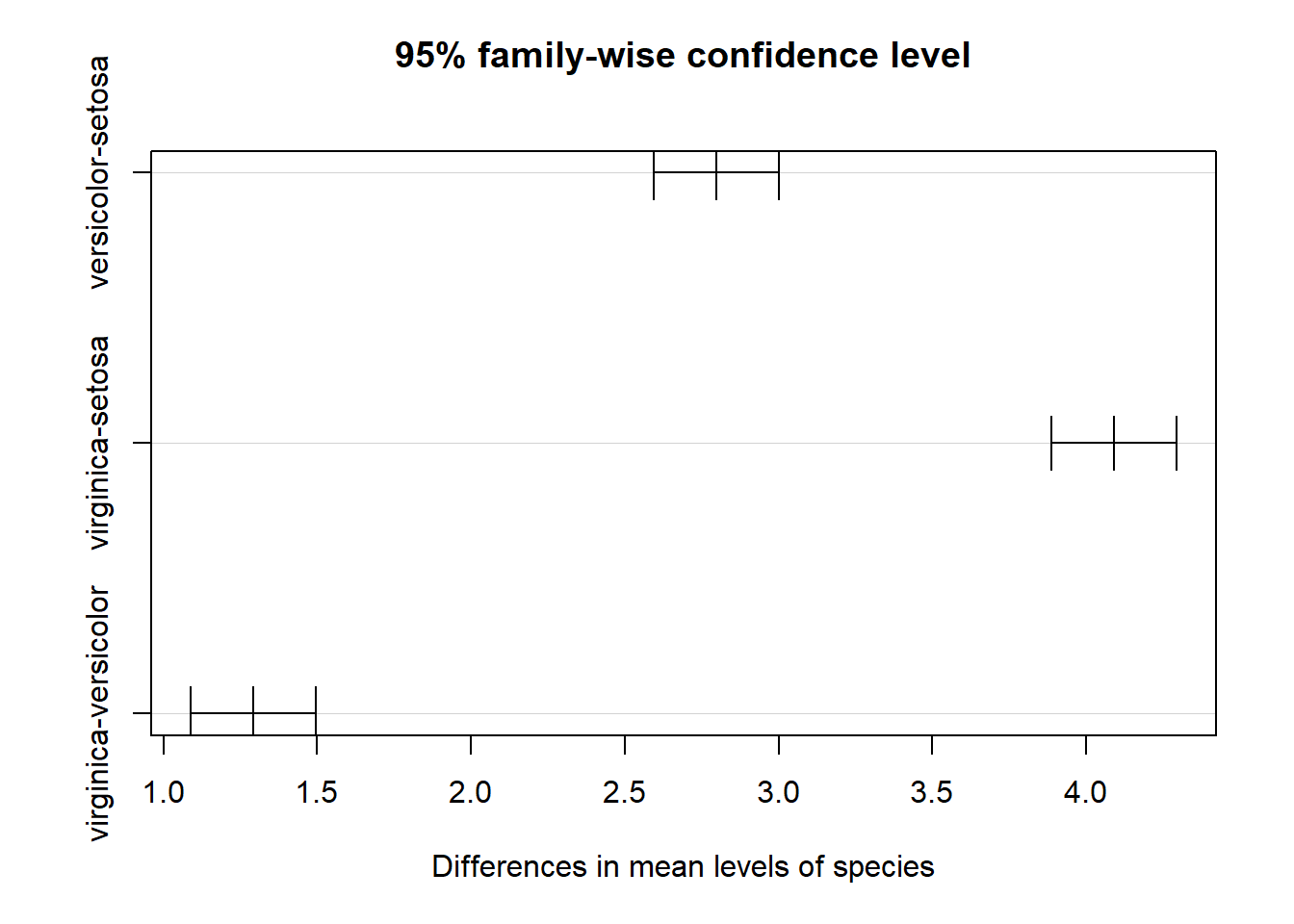
iris\_lm4a <- aov(petal\_length ~ species, data = iris)   
#aov is similar to the anova function, but compatible with Tukeys HSD test  
  
TukeyHSD(iris\_lm4a)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = petal\_length ~ species, data = iris)  
##   
## $species  
## diff lwr upr p adj  
## versicolor-setosa 2.798 2.59422 3.00178 0  
## virginica-setosa 4.090 3.88622 4.29378 0  
## virginica-versicolor 1.292 1.08822 1.49578 0

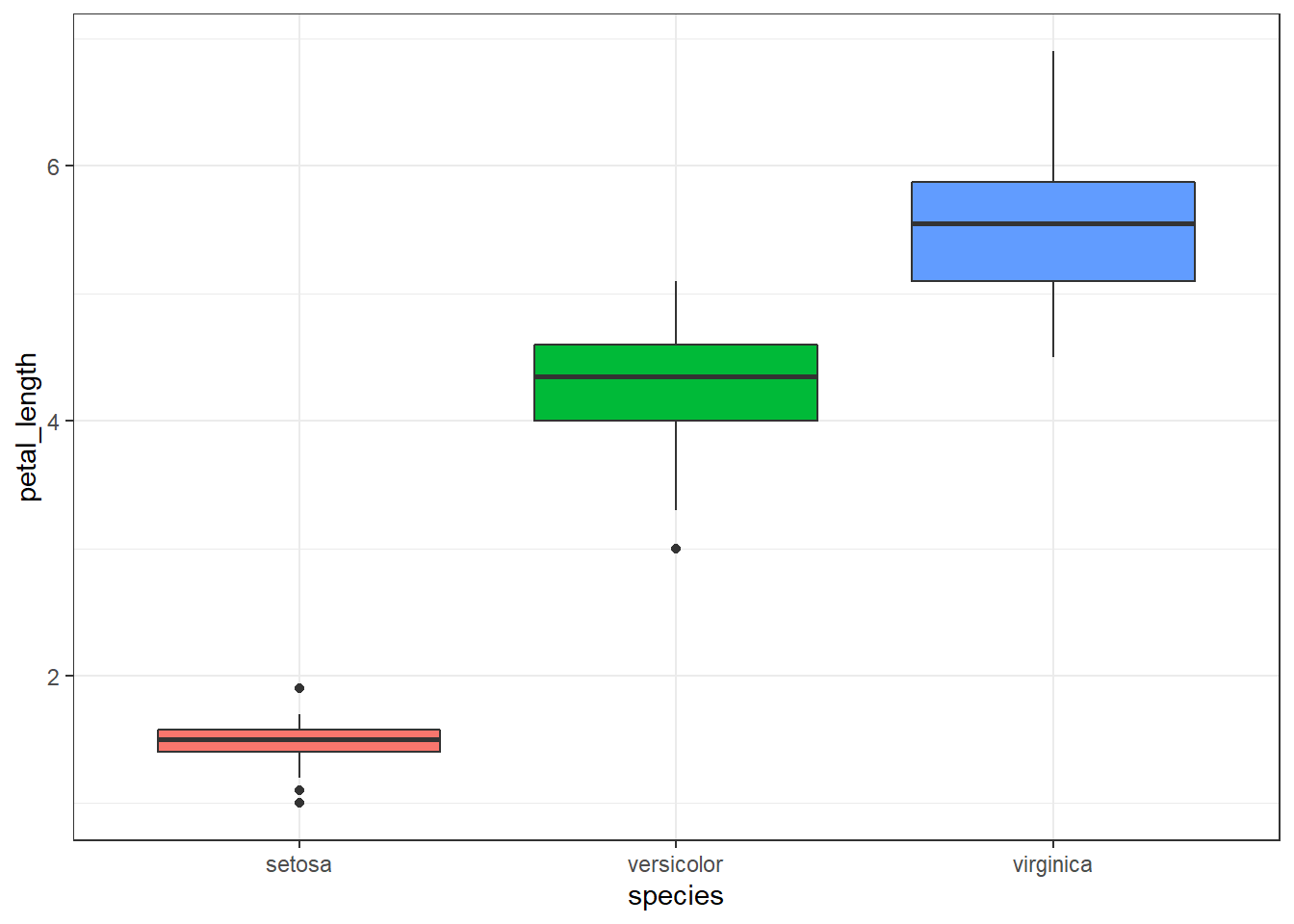
The output shows mean-differences between each species pair. Since p < 0.05 for all pairs, we can say that each species mean petal\_length is significantly different from one another.

We can use a base-R function to plot the results of the TukeyHSD test, or ggplot() to compare groups in a boxplot (medians).

plot(TukeyHSD(iris\_lm4a))



ggplot(data = iris, aes(x = species, y = petal\_length)) +  
 geom\_boxplot(aes(fill = species)) +  
 theme\_bw() +   
 theme(legend.position = "none")



*Note*: If you compare the differences between means in the TukeyHSD plot and the group-level estimates (means) in the summary() output, you’ll see that they are the same!

Two- or Multi-way ANOVA

Like linear regression models with multiple covariates, we can run ANOVAs on linear models with more than one factor.

Here, let’s create a new factor in our iris dataset, based on the width of sepals:

iris2 <- iris %>%   
 mutate(sep\_width\_f = as.factor(  
 case\_when(sepal\_width < 3 ~ "narrow",  
 sepal\_width >= 3 ~ "wide")))  
  
class(iris2$sep\_width\_f)

## [1] "factor"

#View(iris2) #unhash to see the new dataframe

Now we have a new variable, a factor called sep\_width\_f with two levels, “narrow” and “wide”.

If we’re interested in whether mean petal\_length differs by species and by plants with narrow versus wide sepals, we can run a two-way ANOVA:

iris\_lm5 <- lm(petal\_length ~ species + sep\_width\_f, data = iris2)  
  
anova(iris\_lm5)

## Analysis of Variance Table  
##   
## Response: petal\_length  
## Df Sum Sq Mean Sq F value Pr(>F)   
## species 2 437.10 218.551 1273.465 < 2.2e-16 \*\*\*  
## sep\_width\_f 1 2.17 2.166 12.622 0.0005138 \*\*\*  
## Residuals 146 25.06 0.172   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The ANOVA table indicates that there are significant differences in mean petal length among species and among plants with narrow versus wide sepals.

Try it out

Which sepal widths have greater petal lengths?

How would you visualize these differences in a graph?

Assumptions and diagnostics

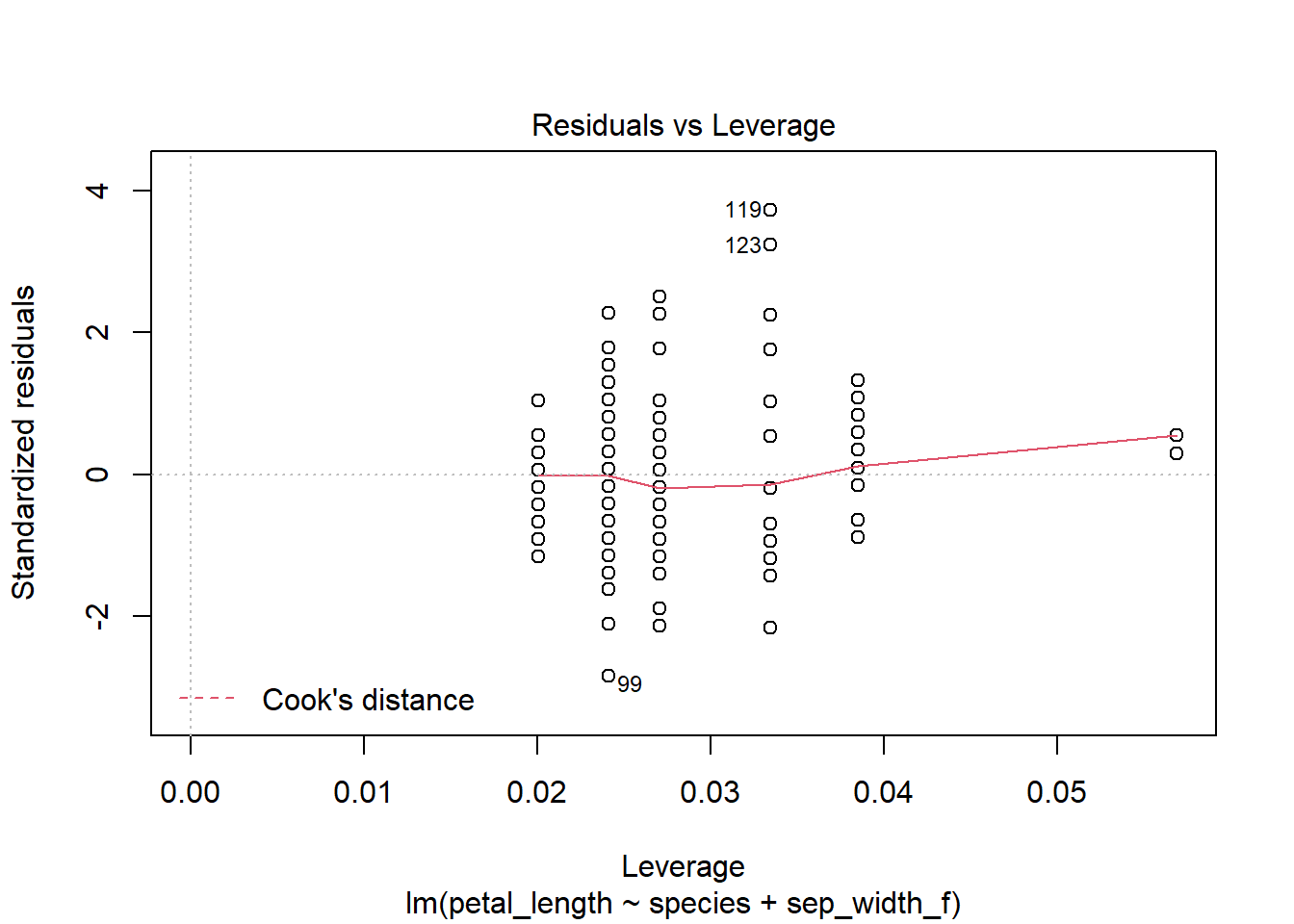
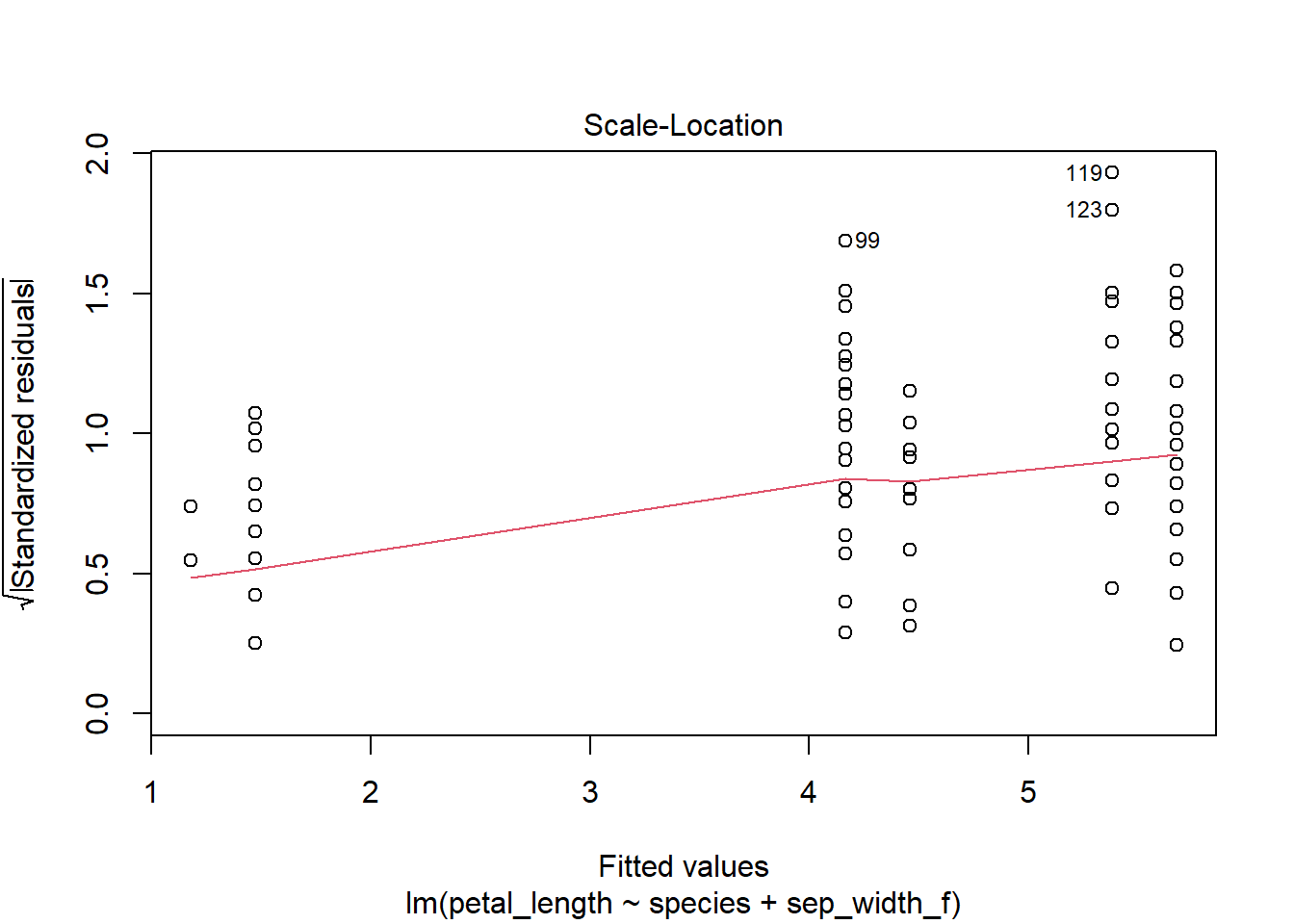
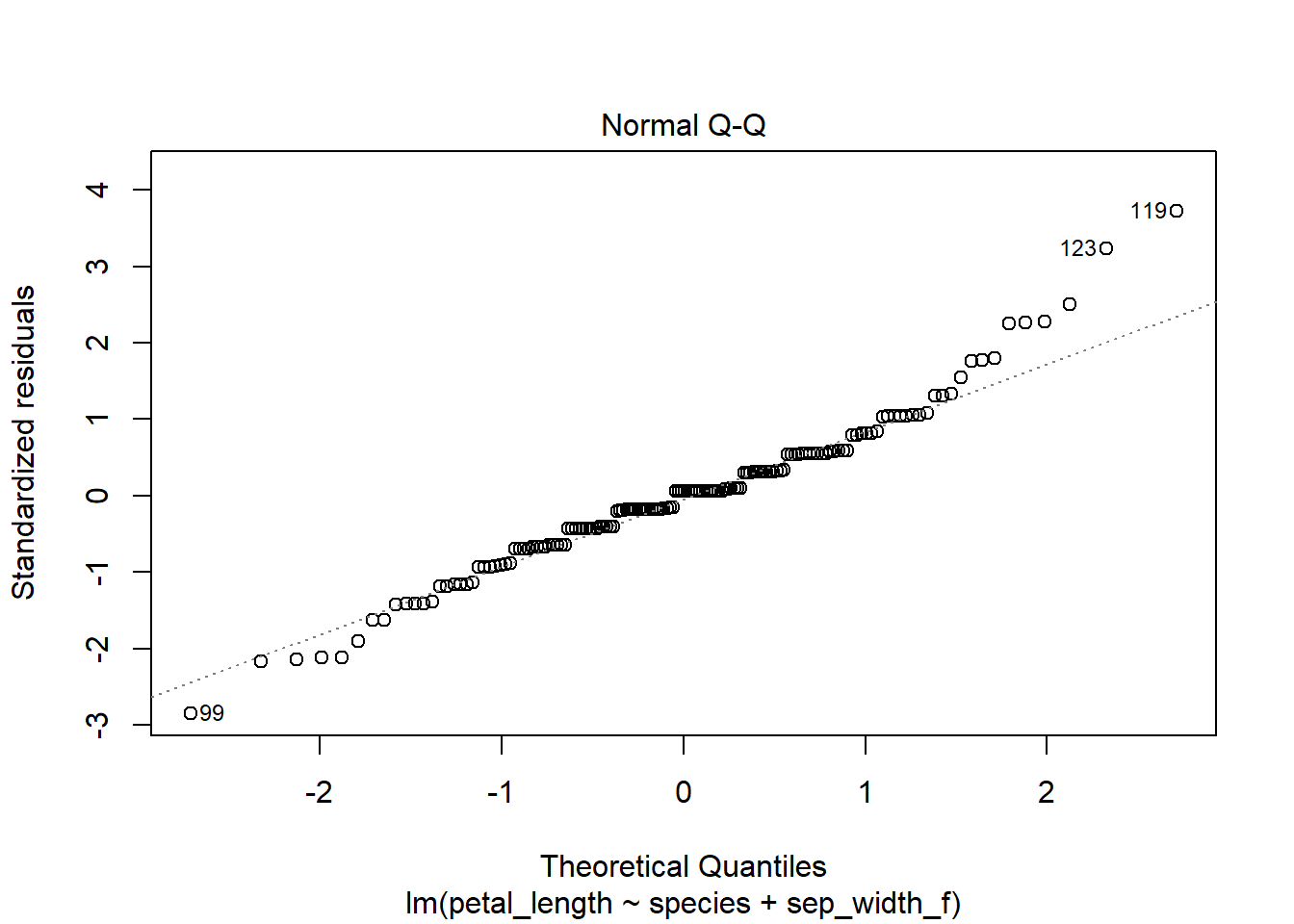
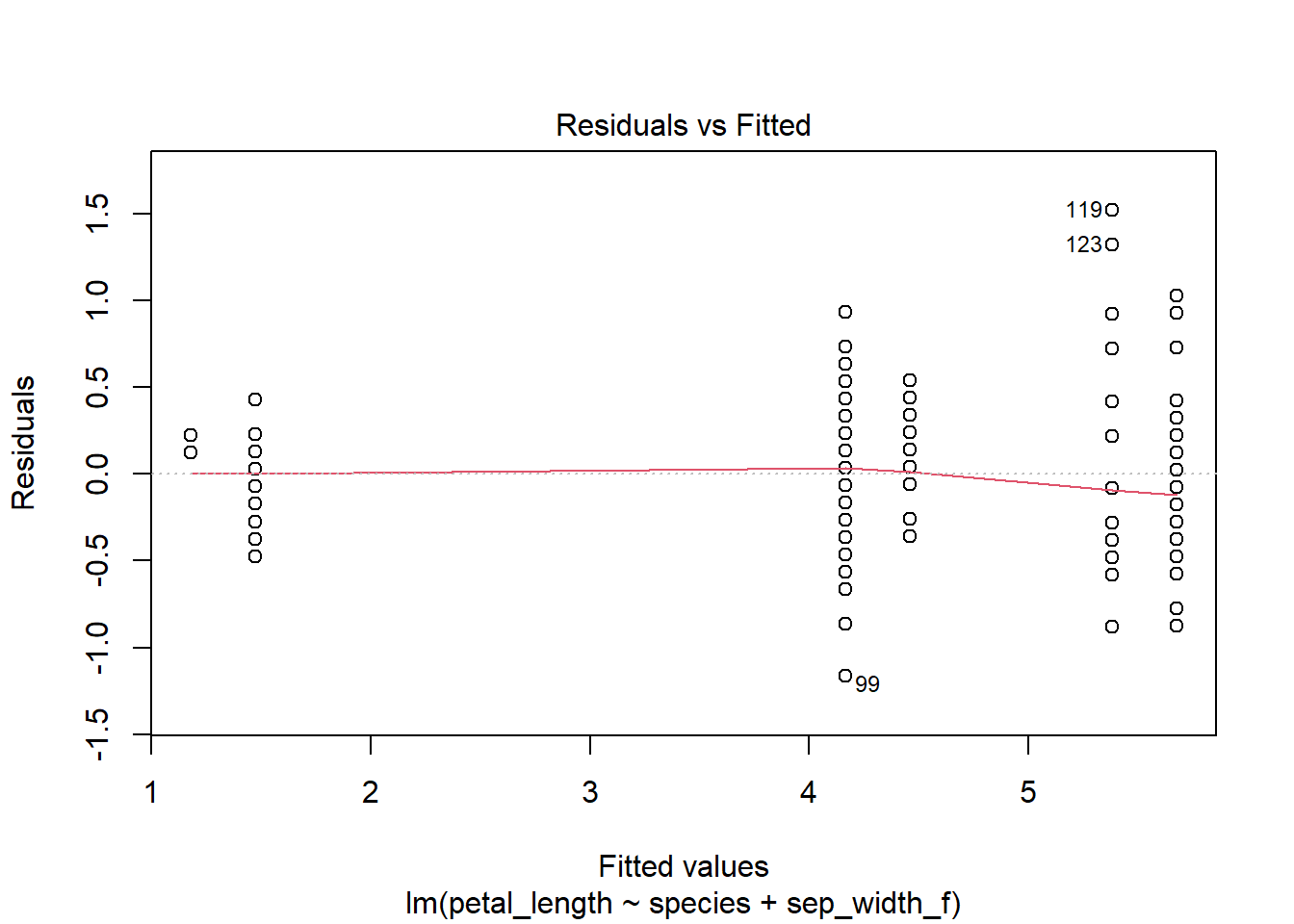
Assumptions of ANOVA with LMs are very similar to those of linear regression. These include:

- Homoscedasiticity (Equal variances among groups)

- Normality and independence of residuals

We can check these assumptions using the series of diagnostic plots like we did before:

plot(iris\_lm5)



*Note*

- The leveneTest() funcation evaluates the assumption of equal variance among groups.

- The Kruskal-Wallis test is a non-parametric alternative to ANOVA when these assumptions are not met (kruskal.test())

Sum of Squares (SS) Types

The type of Sum of squares used can determine the accuracy and interpretability of your ANOVA when you have multiple covariates, interactions between covariates, and when data are unbalanced (unequal # observations across groups).

Type I SS (Sequential)

Type I SS partitions the overall SS sequentially among covariates; the order of explanatory variables in a model matters. Type I SS can lead to different results with different covariate orders when data are unbalanced.

Type III SS (Partial)

Type III SS partitions SS for all nested models simultaneously, adjusting for covariate order and interactions. It is more appropriate to use when significant interactions exist.

*Notes*:

- Type II SS is recommended when you have multiple covariates but significant interactions are not present

- It is usually not worth interpreting main effects when interactions are present

- These contrasts and SS types also apply to linear regression!

- When data are balanced (factors are orthogonal) Types I, II, and III SS give the same results

To run an ANOVA with Type III SS, it’s recommended you change the type of contrasts in R. Instead of comparing factor levels to a reference level (as in treatment contrasts), sum-to-zero (orthogonal) contrasts compare each level to the overall mean. This helps with interpreting the model.

options(contrasts = c("contr.sum", "contr.poly")) #"Sum-to-zero contrasts  
#The default contrasts setting is treatment contrasts in R

Unlike anova() and aov(), the Anova() function from the car package allows us to test Type II and Type III SS:

iris\_lm5a <- lm(petal\_length ~ species\*sep\_width\_f, data = iris2)  
  
Anova(iris\_lm5a, type = 3, contrasts=list(species=contr.sum, sep\_width\_f=contr.sum))

## Anova Table (Type III tests)  
##   
## Response: petal\_length  
## Sum Sq Df F value Pr(>F)   
## (Intercept) 3.645 1 21.2069 8.986e-06 \*\*\*  
## species 42.692 2 124.1916 < 2.2e-16 \*\*\*  
## sep\_width\_f 0.026 1 0.1520 0.6972   
## species:sep\_width\_f 0.306 2 0.8902 0.4128   
## Residuals 24.750 144   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Alternative to previous contrast option code: Specifying contrast sums in Anova() for sig. interaction, unbalanced data  
  
Anova(iris\_lm5a, type = 2)

## Anova Table (Type II tests)  
##   
## Response: petal\_length  
## Sum Sq Df F value Pr(>F)   
## species 387.45 2 1127.1045 < 2.2e-16 \*\*\*  
## sep\_width\_f 2.17 1 12.6029 0.0005207 \*\*\*  
## species:sep\_width\_f 0.31 2 0.8902 0.4128135   
## Residuals 24.75 144   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#anova(iris\_lm5a)   
#Our data are well balanced, so the Type I anova gives roughly the same results

In the case of a linear model with species and sep\_width\_f, the interaction term is not significant so a Type II ANOVA would be more appropriate than Type III, *if* are data were unbalanced.

For a far better explanation of SS Types in ANOVA, see: [Anova – Type I/II/III SS explained](http://md.psych.bio.uni-goettingen.de/mv/unit/lm_cat/lm_cat_unbal_ss_explained.html)

ANCOVA

An Analysis of Covariance (ANCOVA) compares means of a response variable among 2+ groups while accounting for variability of a numerical variable(s). It can be one-way multi-way.

It makes the same assumptions as a regular ANOVA, *in addition* to linearity between the covariate and response for each level of the factor(s), homogeneity of regression slopes across factor levels.

You can learn more here: [ANCOVA in R](https://www.datanovia.com/en/lessons/ancova-in-r/)

Other Stuff

*You may need to install some of the packages used in the following code chunks*

Standardized outputs

When doing linear regression analysis, you may have explanatory variables that are really different in their measurement units. This can make interpreting their relative effects on a response tricky (though this isn’t a huge issue in the iris dataset).

Standardizing (and mean-centering) input variables can make interpretation and comparison among their effects on a response much simpler. The jtools package and the following functions are one method for standardizing parameter estimates.

library(jtools)  
  
summ(iris\_lm3, scale = TRUE) #the scale argument centers & standardizes the explanatory variables coefficients   
  
  
  
#Visualize differences in parameter estimates  
plot\_summs(iris\_lm3, scale = TRUE)  
  
#We can even compare parameter estimates from multiple models  
plot\_summs(iris\_lm3, iris\_lm2, model.names = c("iris\_3", "iris\_2"), scale = TRUE)   
  
  
#some compatibility with ggplot  
  
#see ?plot\_summs for more

*Note*: These functions are applicable to LMs as well as more complex models.

Visualizing Regressions

The visreg package and function are really useful for when you want to quickly and cleanly plot the results of a regression model. It parses out the regression coefficients of each explanatory variable and plots them against the response.

library(visreg)  
  
visreg(iris\_lm2)  
  
visreg(iris\_lm3)   
  
#You can also visualize models built for ANOVA  
visreg(iris\_lm5)

Visreg has functionality and arguments for all kinds of models beyond LMs, and is compatible (kind of?) with ggplot2.

Visualizing Interactions

The interactions package and interact\_plot() is one option for visualizing interaction terms in a linear regression.

library(interactions) #You may need to install this package  
  
interact\_plot(iris\_lm\_int, pred = "petal\_width", modx = "sepal\_length")

I’m pretty sure there are ways to do this in ggplot, too!

Sources

[STHDA](http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/)

[Kyle Edward’s Lecture 1](https://drive.google.com/file/d/0B5QCuGKrabF5ZTBFcVFTbmRRUkk/view)

[Zuur et al. Chapter 2](https://link.springer.com/book/10.1007/978-0-387-87458-6)

Try it out!

Ready to try your own linear regression or ANOVA?

Here’s a built-in datasets on the features of 32 cars from 1974:

data(mtcars)  
  
mtcars  
  
?mtcars

Linear Regression

Choose a response variable and one or more explanatory variable. Build a linear model and conduct a linear regression in R.

What do you find?

Does your model meet the assumptions of linear regression?

Build another model. Can you find any significant interactions among variables?

ANOVA

Here’s some data on an experiment studying how vitamin C affects tooth growth in 60 Guinea pigs. To run an ANOVA on ‘dose’, we can change it from a numeric variable to a factor.

ToothGrowth <- ToothGrowth %>%   
 clean\_names()%>%   
 mutate(dose = as\_factor(dose))  
  
View(ToothGrowth)  
  
?ToothGrowth

Using this or other data, create a linear model.

Run an ANOVA on your model. What do you find?

Does your model meet the assumptions of ANOVA?

Build a model with dose and supplement to run a two-way ANOVA. What do you find?