Linear models

Your Name Here

Month Day, 2020

Contents

Linear modeling

Q0: The dataset

A dataset called `trees` comes packaged with R and is always available in your workspace. This dataset contains measurements of felled timber from 31 cherry trees:

- `Height` (in feet)
- `Volume` (in cubic ft)
- `Girth` (actually it's the tree diameter, measured 4'6" above the ground (in inches)

Check out the dataset to get a feel for what it looks like.

```
## dataset ##########################################################################
# inspect the dataset

---
---
```

```
## Error: <text>:3:1: unexpected input
## 2: # inspect the dataset
## 3: _
##    ^
```

Q1: Create linear models of Volume vs. predictor variables

We are interested in predicting the tree Volume from the measured height and diameter. First, let's create the simplest possible linear models, using one or both predictive variables.

Note: Remember that there are two equivalent expressions for specifying the formula: you can either refer to the columns of the columns directly using the `$` notation, or you can specify the data to be used with `data=` and just use the column names in the formula. The latter is a bit easier to type and to read.

```
## lm ###############################################################################
# create linear models & check summaries

# Height only
```

```
tree_lm1 = lm(___ ~ ___, data = ___)
summary(___)

# Girth only
tree_lm2 = ___
---

# Height + Girth
tree_lm3 = ___
---
```

```
## Error: <text>:5:15: unexpected input
## 4: # Height only
## 5: tree_lm1 = lm(_
##                  ^
```

What do the models tell you about how much of the variation in Volume are explained by Height and Girth? You should consider the following in your evaluation:

- Residuals – How big are they?
- Coefficients –
- R-squared – How much of the variation in the data are explained by the Explanatory variable? Does this look encouraging?

Which model seems best at first glance, and why? Is there anything troubling about these models? Why or why not?

Based on the statistics from the linear regression, how would you evaluate the goodness-of-fit? Do the data violate any of the basic assumptions for linear modeling?

Q2: Simple plots of Response vs. Explanatory variables

Before deciding upon a model, we need to check whether any of the basic assumptions of linear models are violated. What are these assumptions?

Let's draw some simple plots to get an intial feel for how well the models fit the assumptions of linear models.

First, explore the data by superimposing a smoothened line over the datapoints using `geom_smooth()`.

Note: The default method for smoothing is called ***LOESS****, which stands for "locally estimated scatterplot smoothing". This method uses locally weighted least-squares across a sliding window to compute the best fit for each datapoint. The "smoothing parameter", a.k.a. the "bandwidth", determines how much of the data is used at each step.*

Next, remake the plots to show the best-fit regression line using `stat_smooth()` with the "lm" method. This will automatically show the 95% CI around the regression line.

```
# load the ggplot library
library(ggplot2)

##############################################################################
## 1) create a plot of Volume vs. Height using ggplot

# a) simple plot -- show the data points + a smoothened line
ggplot(data = ___, aes(___, ___)) +
    ___() +  # add data points
    ___()    # add smoothening line
```

```
# b) plot with linear regression line
ggplot(___, ___) +
    ___() +  # add data points
    # add regression line, in RED
    ___(___, ___)

################################################################################
## 2) plot Volume vs. Girth using ggplot

# a) simple plot -- datapoints and smoothened line
---

# b) plot datapoints with linear regression line, in RED
---
```

```
## Error: <text>:8:15: unexpected input
## 7: # a) simple plot -- show the data points + a smoothened line
## 8: ggplot(data = _
##                  ^
```

How do these plots look to you? Do you notice anything funny about the LOESS curve for Volume vs. Girth?

Q3: Diagnostic residual plots

You can assess the quality of any linear model using a variety of plots that show different properties of the **residuals** (the distance from each data point to the fitted line).

Types of Diagnostic Plots

Residuals vs. Fitted

This plot looks at the distance of the data points to the predicted line. There are several things to look for on this plot:

- **Small residuals?** Ideally, the points should be close to the dotted line (where the residual is 0). Small residuals reflect a good fit to the data.
- **Uniform residuals?** The magnitude of the residuals should be about the same across the range of measured x-values (equal variance).
- **Outliers?** Outliers are numbered accoring to their index in the data frame to call attention to them.
- **Linear relationship?** If the relationship is linear, then the fitted line should be relatively straight. If this is not the case, then applying some transformations to the predictor and/or response variables may help make the data more linear.

Normal QQ plot of residuals

This plot is a quantile-quantile plot that shows whether the Residuals are normally distributed. Again, the best case scenario is that the points fall on the dashed line.

Scale-location plot

This plot shows the **variance** in the residuals to the regression line. Ideally, the line should be horizontal, indicating that the variance is evenly distributed across all the datapoints. **Homoscedasticity**, or equal variance, is an assumption of certain statistical tests.

Residuals vs. Leverage

Cook's distance is calculated by removing a specific data point and checking how much the model has changed. The greater the Cook's distance, the more influential that point is on the model. The dashed horizontal line corresponds to the ideal situation in which each datapoint has little effect on the overall model. Outliers can be very disruptive, so this is a good way to determine if there are indeed outliers that have a strong negative impact on the linear model.

Examine the residuals plots

Here we will use a `ggplot2` companion package called `ggfortify`, which takes the normal base R graphics and converts them into ggplot graphs.

This package defines the function `autoplot()`, which automatically creates one or more standard plots that are tailored to specific object classes. Here we will apply `autoplot` to our alternative linear models.

```
## diagnostics ##############################################################################
# load the ggfortigy library
# install.packages("ggfortify") -- uncomment if you haven't already loaded the package
library(___)

# draw the residual plots for the three linear models above
---
---
---
```

```
## Error: <text>:4:9: unexpected input
## 3: # install.packages("ggfortify") -- uncomment if you haven't already loaded the package
## 4: library(_
##            ^
```

Examine the plots for the three models and discuss the following:

Model 1: Volume ~ Height

Comment specifically on the uniformity of variation as revealed by the Residuals vs. Fitted and Scale-Location plots. What do the patterns reveal?

Models 2 & 3: Volume ~ Girth and Volume ~ Height + Girth

Comment specifically on the Residuals vs. Fitted plot. What shape is the LOESS line? Can you think of any functions that resemble this pattern?

Also consider the Scale-Location and Residuals vs. Leverage plots. What do you learn from these?

Finally, discuss what you think the main dependencies in this dataset are, i.e. how influential each predictor is. The way you think about this will determine the downstream analyses that you will perform.

Q4: Data transformation

The above plots reveal that linear models of the raw data might not provide the best picture of the relationship of Height and Girth with Volume.

Reflecting upon the exercises above, and *your own knowledge of volumetric measurements*, can you think of a way to transform the data that might improve the model? Explain your reasoning. If you have a particular mathematical formula in mind, please include it in your answer.

Hint: You may want to play around with a few transformations on your own to see what might work best (do not include your exploratory analysis here). Since this is not a trick question, stick to common transformations

like log / power functions and quadratics (square / square root). You can also just answer this question based on first principles.

Below, create a linear model using the transformation you think works best. Since Height made little contribution to the model, just use Girth as the Predictor variable.

Note: Please transform the Response variable rather than the Predictor variable(s) in the linear model. This will allow you to predict Volume from any set of new measured values for Girth, which you will be asked to do below. Making predictions on new data, based on models generated from training data, is a common task in data analysis.

```
## transformation ========================================================================
# a) make a linear model and look at the summary
model = ___
---

# b) make a ggplot of the **transformed** Response variable (y-axis) vs. Girth (x-axis)
# for the y-axis, just write in the function you used to transform the Volume
ggplot(___) +

# show the datapoints and the regression line as before
---
---


# c) draw the diagnostic plots for the residuals of this linear model
---
```

```
## Error: <text>:3:9: unexpected input
## 2: # a) make a linear model and look at the summary
## 3: model = _
##            ^
```

How have your stats and diagnostic plots changed? What features of the particular transformation you have chosen do you think correct the problems you saw before?

Q5: Prediction and Confidence Intervals

First, apply the `predict.lm()` function to your best linear model to predict the Volume across a range of values for Girth. Read the documentation for this function, and include lower and upper *Prediction Intervals* in the model.

For the predictions, generate 20 random numbers chosen from a uniform distribution that spans the range of values in the original dataset.

```
## predict ===============================================================================
# generate 20 Girth measurements randomly sampled across the previously measured range
samples = ___

# store these in a data frame as a column named "Girth"
test_set = ___

# make the predictions for the test set using the model above
# include the prediction intervals for each point
pred_data = data.frame(predict.lm(___, newdata = ___, interval = ___))

# add the predictions to the test dataset by adding new columns to the data frame
```

```
# pick a name for the prediction column that represents the data transformation you are using
# e.g. VolumeFunction (where Function is some abbreviation of the transform)
test_set = cbind(test_set,
                 ___ = pred_data$fit,
                 lwr = pred_data$lwr,
                 upr = pred_data$upr)
test_set
```

```
## Error: <text>:3:11: unexpected input
## 2: # generate 20 Girth measurements randomly sampled across the previously measured range
## 3: samples = _
##              ^
```

Now, use `ggplot2` to plot the following:

- the original data
- the regression line including lower and upper confidence intervals
- the prediction intervals for the model
- the predicted values for the test data

```
## final plot #################################################################################
# a complicated ggplot command!
ggplot() +

    # **regression line** for the model using the original **trees dataset** -- in DARKGRAY
    # (for the y-axis, write in the function you used to transform the Volume)
    stat_smooth(method=___, data = ___, aes(x=___, y=___), ___) +

    # **data points** from the original **trees dataset** -- in MIDNIGHTBLUE
    # (for the y-axis, write in the function you used to transform the Volume)
    geom_point(data = ___, aes(x=___, y=___), ___) +

    # **data points** for the **test dataset** -- in RED
    # (use Girth and add the column name you used for the transformed data)
    geom_point(data = ___, aes(x=___, y=___), ___) +

    # **prediction intervals** from the test data -- as RED DASHED LINES
    # (use the Girth and the columns containing the lower and upper prediction bounds)
    geom_line(data = ___, aes(x=___, y=___), ___, ___) +
    geom_line(data = ___, aes(x=___, y=___), ___, ___) +

    # **plot title** and axis labels
    # (for the y-label, write in the function you used to transform the Volume)
    ggtitle("trees Volume prediction -- training data (black) + test data (red)") +
    xlab("___") + ylab("___")
```

```
## Error: <text>:7:24: unexpected input
## 6:     # (for the y-axis, write in the function you used to transform the Volume)
## 7:     stat_smooth(method=_
##                           ^
```

And voilà! The End. ;-)