StatR 502 Lecture 5: Simulation, Box-Cox, Stepwise Model Selection, and the Anatomy of a Package

Gregor Thomas

February 4, 2016

Announcements: $\{$.build = FALSE$\}$

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15

Tonight's Menu:

Quick Review

Things you should know off the top of your head

Announcements: $\{.\text{build} = \text{FALSE}\}$

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

Quick Review

Things you should know off the top of your head

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review

Quick Review

Things you should know off the top of your head

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)

Quick Review

Things you should know off the top of your head

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations

Quick Review

Things you should know off the top of your head

Announcements: $\{.build = \text{FALSE}\}$

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)

Quick Review

Things you should know off the top of your head

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise

Quick Review

Things you should know off the top of your head

Announcements: $\{.build = \text{FALSE}\}$

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other

Quick Review

Things you should know off the top of your head

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons

Quick Review

Things you should know off the top of your head

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

Announcements: $\{.build = FALSE\}$

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?

Announcements: $\{.build = FALSE\}$

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?
- Looking at a residuals plot (residuals vs fitted values), what pattern would make you think "I need to transform or try a Poisson GLM?"

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?
- Looking at a residuals plot (residuals vs fitted values), what pattern would make you think "I need to transform or try a Poisson GLM?"

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?
- Looking at a residuals plot (residuals vs fitted values), what pattern would make you think "I need to transform or try a Poisson GLM?"

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?
- Looking at a residuals plot (residuals vs fitted values), what pattern would make you think "I need to transform or try a Poisson GLM?"

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
  - Stepwise
  - Other
  - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?
- Looking at a residuals plot (residuals vs fitted values), what pattern would make you think "I need to transform or try a Poisson GLM?"

Announcements: {.build = FALSE}

- We'll talk Final Projects tonight. Project Proposals are due Sunday February 15
- Homework writeups

Tonight's Menu:

- Quick review
- Simulation for Prediction (briefly)
- Box-Cox Transformations
- Stepwise Regression (and the like)
    - Stepwise
    - Other
    - Multiple comparisons
- Projects and Packages

Quick Review

Things you should know off the top of your head

- AIC: which direction is better?
- What's the difference between AIC and BIC?
- What is the point of linear transformations of data?
- In logistic regression, what does the linear predictor predict?
- Looking at a residuals plot (residuals vs fitted values), what pattern would make you think "I need to transform or try a Poisson GLM?"

Simulation (briefly)

A naive approach to simulation...

A regression model has point-estimates for coefficients, but also standard errors. To simulate data from the model based only on the point-estimates would ignore the important information we know about their variances.

- This ignores the uncertainty in the model fitting, which is encoded in the standard errors for the coefficients.

A little better

A better procedure would be to draw a simulated coefficient from a normal distribution with mean = point estimate and sd = standard error...

Much better

This is easily accessible with `vcov(your_model)`, but the `sim()` function from `arm` makes it even easier.

Note that this is more or less the approach Nate Silver and the 538-blog in the popular election forecasting. Poll results, with margins of error, were compiled for each state, then simulations run to see the probability of outcomes in the aggregate.

Simulation (briefly)

A naive approach to simulation...

A regression model has point-estimates for coefficients, but also standard errors. To simulate data from the model based only on the point-estimates would ignore the important information we know about their variances.

- This ignores the uncertainty in the model fitting, which is encoded in the standard errors for the coefficients.

A little better

A better procedure would be to draw a simulated coefficient from a normal distribution with mean = point estimate and sd = standard error...

- But this still ignores the **covariance** of coefficient estimates.

Much better

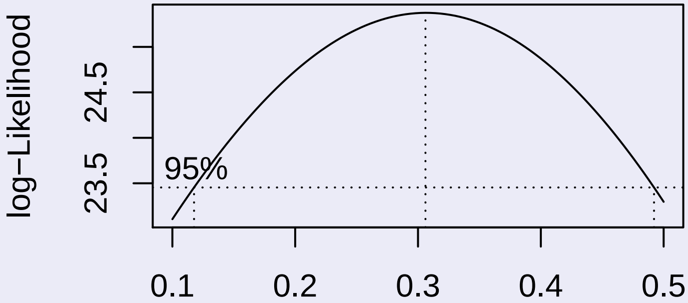
This is easily accessible with `vcov(your_model)`, but the `sim()` function from `arm` makes it even easier.
Note that this is more or less the approach Nate Silver and the 538-blog in the popular election forecasting. Poll results, with margins of error, were compiled for each state, then simulations run to see the probability of outcomes in the aggregate.

Box-Cox Transformations

Essentially, we're testing transforms of the form (y^lambda - 1) / lambda for lambda != 0 log(y) for lambda == 0 But the - 1 and / lambda parts are linear, so they don't change the model fit.

Box Cox

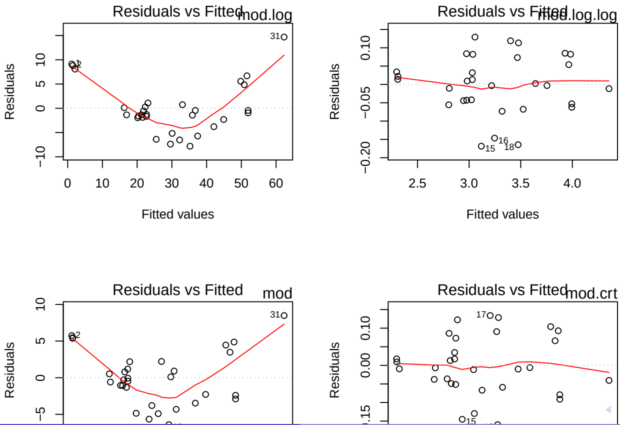
```
boxcox(Volume ~ Height + Girth, data = trees,
       lambda = seq(0.1, 0.5, length = 50))
```



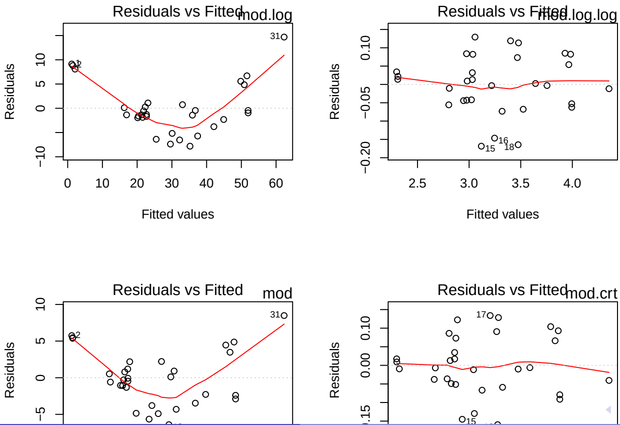
```r
mod.log = lm(Volume ~ log(Height) + log(Girth), data = trees)
mod.log.log = lm(log(Volume) ~ log(Height) + log(Girth), data = trees)
mod = lm((Volume) ~ (Height) + (Girth), data = trees)
mod.crt = lm((Volume)^(1/3) ~ (Height) + (Girth), data = trees)
```

```
library(arm)
display(mod.log.log)

## lm(formula = log(Volume) ~ log(Height) + log(Girth), data = trees)
##             coef.est coef.se
## (Intercept) -6.63     0.80
## log(Height)  1.12     0.20
## log(Girth)   1.98     0.08
## ---
## n = 31, k = 3
## residual sd = 0.08, R-Squared = 0.98

display(mod.crt)

## lm(formula = (Volume)^(1/3) ~ (Height) + (Girth), data = trees)
##             coef.est coef.se
## (Intercept) -0.09     0.18
## Height       0.01     0.00
## Girth        0.15     0.01
## ---
## n = 31, k = 3
## residual sd = 0.08, R-Squared = 0.98
```

```r
par(mfrow = c(2, 2))
plot(mod.log, which = 1)
mtext("mod.log", adj = 1)
plot(mod.log.log, which = 1)
mtext("mod.log.log", adj = 1)
plot(mod, which = 1)
mtext("mod", adj = 1)
plot(mod.crt, which = 1)
mtext("mod.crt", adj = 1)
```



```
par(mfrow = c(2, 2))
plot(mod.log, which = 1)
mtext("mod.log", adj = 1)
plot(mod.log.log, which = 1)
mtext("mod.log.log", adj = 1)
plot(mod, which = 1)
mtext("mod", adj = 1)
plot(mod.crt, which = 1)
mtext("mod.crt", adj = 1)
```



Automated Model Selection

Model Selection

Many people want to take the analyst out of analysis as much as possible. This is a good goal, but one we're still a good ways from achieving.
Model selection is **the** major problem in modern statistics.
Most of the problem is due to the lack of a perfect metric for model performance. If you're willing to focus solely on predictive accuracy, for example, then you can do pretty well with machine learning algorithms (things like random forests, which you'll see next quarter). Unfortunately, to get an *interpretable* model, especially a **causal** model, rather than black box that makes pretty good predictions, there is no silver bullet, and there are problems with selection and bias if you dredge your data too thoroughly.
All that said, let's look at a couple methods for automatically searching the "model space".

Step-wise Model Selection

Stepwise model selection (forward selection) starts with a simple model, and adds in predictors one at a time, choosing the best to add based on AIC (or BIC or similar).
Backward selection starts with a complex model and removes terms one at a time.
These are implemented in `MASS::stepAIC`. Setting `direction="both"` will consider adding or removing terms at each step.
(quick demo)

More thorough searches

You could try all combinations of variables, or all combinations with certain restrictions. The `leaps` package (not just a "step", haha) offers one implementation, another is in `bestglm`, but beware

Automated Model Selection

Model Selection

Many people want to take the analyst out of analysis as much as possible. This is a good goal, but one we're still a good ways from achieving.
Model selection is **the** major problem in modern statistics.
Most of the problem is due to the lack of a perfect metric for model performance. If you're willing to focus solely on predictive accuracy, for example, then you can do pretty well with machine learning algorithms (things like random forests, which you'll see next quarter). Unfortunately, to get an *interpretable* model, especially a **causal** model, rather than black box that makes pretty good predictions, there is no silver bullet, and there are problems with selection and bias if you dredge your data too thoroughly.
All that said, let's look at a couple methods for automatically searching the "model space".

Step-wise Model Selection

Stepwise model selection (forward selection) starts with a simple model, and adds in predictors one at a time, choosing the best to add based on AIC (or BIC or similar).
Backward selection starts with a complex model and removes terms one at a time.
These are implemented in `MASS::stepAIC`. Setting `direction="both"` will consider adding or removing terms at each step.
(quick demo)

More thorough searches

You could try all combinations of variables, or all combinations with certain restrictions. The `leaps` package (not just a "step", haha) offers one implementation, another is in `bestglm`, but beware

Automated Model Selection

Model Selection

Many people want to take the analyst out of analysis as much as possible. This is a good goal, but one we're still a good ways from achieving.

Model selection is **the** major problem in modern statistics.

Most of the problem is due to the lack of a perfect metric for model performance. If you're willing to focus solely on predictive accuracy, for example, then you can do pretty well with machine learning algorithms (things like random forests, which you'll see next quarter). Unfortunately, to get an *interpretable* model, especially a **causal** model, rather than black box that makes pretty good predictions, there is no silver bullet, and there are problems with selection and bias if you dredge your data too thoroughly.

All that said, let's look at a couple methods for automatically searching the "model space".

Step-wise Model Selection

Stepwise model selection (forward selection) starts with a simple model, and adds in predictors one at a time, choosing the best to add based on AIC (or BIC or similar).

Backward selection starts with a complex model and removes terms one at a time.

These are implemented in `MASS::stepAIC`. Setting `direction="both"` will consider adding or removing terms at each step.

(quick demo)

More thorough searches

You could try all combinations of variables, or all combinations with certain restrictions. The `leaps` package (not just a "step", haha) offers one implementation, another is in `bestglm`, but beware

Automated Model Selection

Model Selection

Many people want to take the analyst out of analysis as much as possible. This is a good goal, but one we're still a good ways from achieving.
Model selection is **the** major problem in modern statistics.
Most of the problem is due to the lack of a perfect metric for model performance. If you're willing to focus solely on predictive accuracy, for example, then you can do pretty well with machine learning algorithms (things like random forests, which you'll see next quarter). Unfortunately, to get an *interpretable* model, especially a **causal** model, rather than black box that makes pretty good predictions, there is no silver bullet, and there are problems with selection and bias if you dredge your data too thoroughly.
All that said, let's look at a couple methods for automatically searching the "model space".

Step-wise Model Selection

Stepwise model selection (forward selection) starts with a simple model, and adds in predictors one at a time, choosing the best to add based on AIC (or BIC or similar).
Backward selection starts with a complex model and removes terms one at a time.
These are implemented in `MASS::stepAIC`. Setting `direction="both"` will consider adding or removing terms at each step.
(quick demo)

More thorough searches

You could try all combinations of variables, or all combinations with certain restrictions. The `leaps` package (not just a "step", haha) offers one implementation, another is in `bestglm`, but beware

Automated Model Selection

Model Selection

Many people want to take the analyst out of analysis as much as possible. This is a good goal, but one we're still a good ways from achieving.
Model selection is **the** major problem in modern statistics.
Most of the problem is due to the lack of a perfect metric for model performance. If you're willing to focus solely on predictive accuracy, for example, then you can do pretty well with machine learning algorithms (things like random forests, which you'll see next quarter). Unfortunately, to get an *interpretable* model, especially a **causal** model, rather than black box that makes pretty good predictions, there is no silver bullet, and there are problems with selection and bias if you dredge your data too thoroughly.
All that said, let's look at a couple methods for automatically searching the "model space".

Step-wise Model Selection

Stepwise model selection (forward selection) starts with a simple model, and adds in predictors one at a time, choosing the best to add based on AIC (or BIC or similar).
Backward selection starts with a complex model and removes terms one at a time.
These are implemented in `MASS::stepAIC`. Setting `direction="both"` will consider adding or removing terms at each step.
(quick demo)

More thorough searches

You could try all combinations of variables, or all combinations with certain restrictions. The `leaps` package (not just a "step", haha) offers one implementation, another is in `bestglm`, but beware

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()
```

```
## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```

Choosing Nonlinear Transformations

Box-Cox

Final Projects

- Build a (small) R package, that helps you to
- conduct an analysis on a data set using some of the methods learned this quarter.

Your data set will be included in the package, and your analysis will be written up as package vignette.

Anatomy of an R Package

R packages aren't all that special, just a collection of files. Nothing has to be compiled, the typical way of distributing package is just to compress the files together in a `.tar.gz` "tarball".

Library vs Package

A *library* is a folder on your computer where you store packages. A *package* is a folder that has a specific structure so R knows how to make its functions and documentation available to you when you load it.
You can see libraries on your search path with `.libPaths()` (also used for adding a new library location).

```
.libPaths()

## [1] "C:/rlib"
```