Statistical Inference Course Project

Aurel Prosz

2017 november 18

Part 1: Simulation Exercise

Overview

This short simulation exercise investigates the exponential distribution and its relation to the Central Limit Theorem.
The exponential distribution is simulated in R with the **rexp()** function. Theoreticaly the mean of this distribution is 1/lambda and the standard deviation is also 1/lambda, where lambda is a distribution parameter.
In the following simulations the lambda parameter is set to 0.2 and averages and standard deviations from 40 exponential distribution is going to be investigated per simulation run. Considering the central limit theorem 10000 simulations is going to be enough to evaluate the mean and the standard deviation.
There are three main goals in this exercise:

- Show the sample mean and compare it to the theoretical mean of the distribution.
- Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
- Show that the distribution is approximately normal.

Simulations

First set the default parameters and variables:

```r
set.seed(10) #Set a seed
mns = NULL #vector containing means obtained from each simulations
sds = NULL #vector containing standard deviations obtained from each simulations
lambda = 0.2 #lambda parameter for the exponential distribution
sims <- 10000 #how many simulations will it run
numexp <- 40 #how many exponential distributions are going to be evaluated
#per simulation run
```

Next we construct a *for cycle* to run the simulations with the given parameters:

```r
for (i in 1 : sims) {
m <- mean(rexp(numexp, lambda))
s <- sd(rexp(numexp, lambda))
mns <- c(mns, m)
sds <- c(sds, s)
}
```

Sample Mean versus Theoretical Mean

Next we evaluate the sample mean from the simulations and compare it to the theoretical mean (which is 1/lambda):

```r
print(mean(mns)) #Sample mean
```

```
## [1] 5.007218
```

```r
print(1/lambda) #Theoretical mean
```

```
## [1] 5
```

The results indicates that the sample mean is a good estimate for the theoretical mean.

Sample Variance versus Theoretical Variance

Next we evaluate the sample variance from the simulations and compare it to the theoretical variance (which is (1/lambda))^2:

```r
print(mean(sds^2)) #Sample variance
```

```
## [1] 24.95536
```

```r
print((1/lambda)^2) #Theoretical variance
```

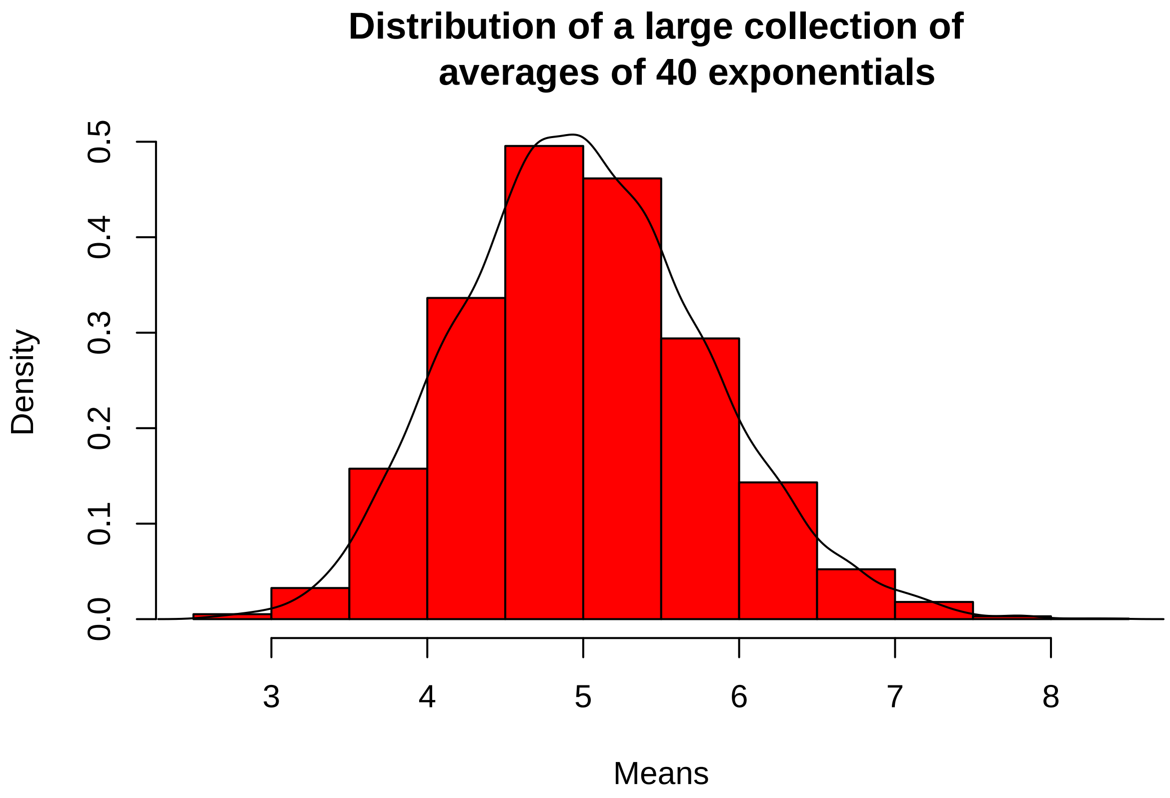
```
## [1] 25
```

The results indicates that the sample variance is a good estimate for the theoretical variance.

Distribution

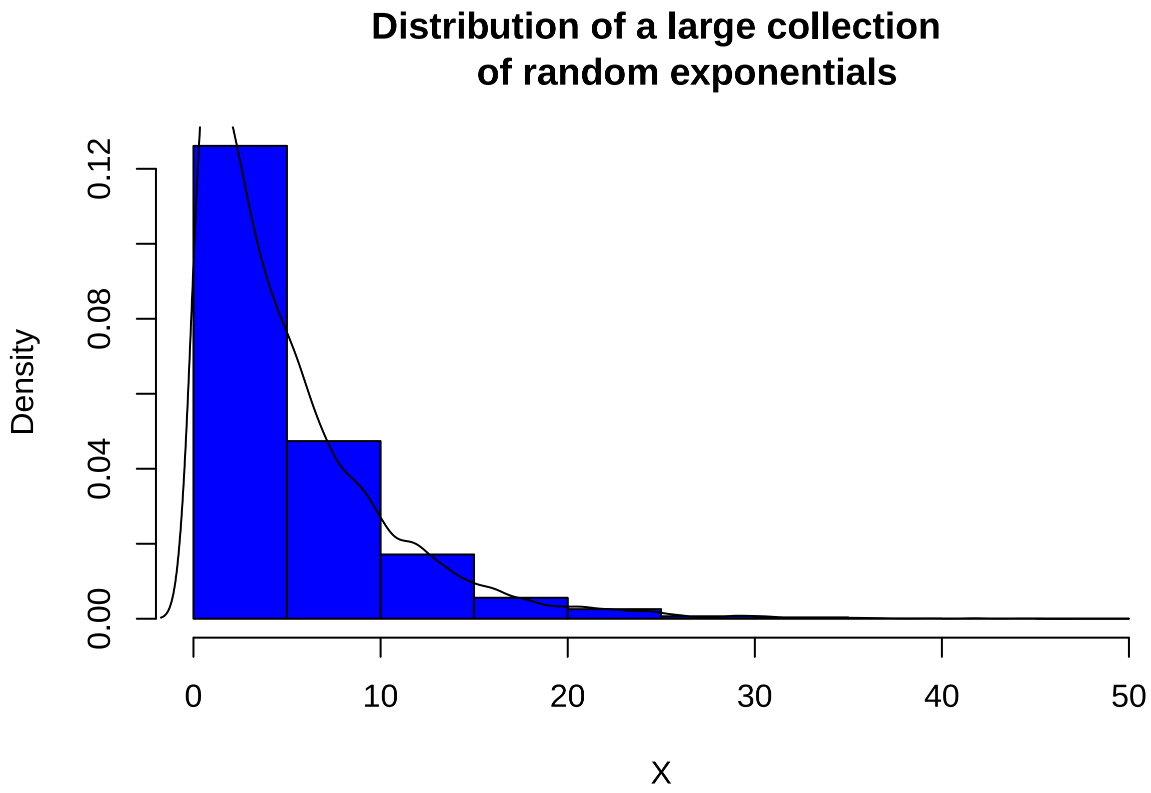
In this part we focus on the difference between the distribution of a large collection of random exponentials and the distribution of a large collection of averages of 40 exponentials.
We use the means from the previous parts and construct a histogram:

```r
hist(mns, col = "red", main = "Distribution of a large collection of
     averages of 40 exponentials",
     xlab = "Means", prob=TRUE)
lines(density(mns)) #Add Kernel Density Estimation
```



It can be clearly seen that the distribution is nearly normal.
Take a look at the large collection of random exponentials:

```
distr <- rexp(sims, 0.2)
hist(distr, col = "blue", main = "Distribution of a large collection
     of random exponentials",
     xlab = "X", prob=TRUE)
lines(density(distr)) #Add Kernel Density Estimation
```



```
mean(distr)
```

```
## [1] 4.972049
```

It can be seen that this second distribution is not gaussian, but the means are equal. The explanation is that according to the Central Limit Theorem the sampling averages are distributed normaly as the simulation size converges to infinity.

Part 2: Basic Inferential Data Analysis

Overview

In this part of the exercise I am going to present a short exploratory, and a more detailed inference analysis of the built-in ToothGrowth dataset which contain information about the growth of tooth of animals from certain supplements. ## Basic exploratory analysis

Basic information about the ToothGrowth dataset

ToothGrowth {datasets} R Documentation
The Effect of Vitamin C on Tooth Growth in Guinea Pigs

Description

The response is the length of odontoblasts (cells responsible for tooth growth) in 60 guinea pigs. Each animal received one of three dose levels of vitamin C (0.5, 1, and 2 mg/day) by one of two delivery methods, (orange juice or ascorbic acid (a form of vitamin C and coded as VC).

Usage

ToothGrowth
Format

A data frame with 60 observations on 3 variables.

[,1] len numeric Tooth length
[,2] supp factor Supplement type (VC or OJ).
[,3] dose numeric Dose in milligrams/day
Source

C. I. Bliss (1952) The Statistics of Bioassay. Academic Press.

References

McNeil, D. R. (1977) Interactive Data Analysis. New York: Wiley.

Crampton, E. W. (1947) The growth of the odontoblast of the incisor teeth as a criterion of vitamin C intake of the guinea pig. The Journal of Nutrition 33(5): 491–504. http://jn.nutrition.org/content/33/5/491.full.pdf

Basic data processing before the real analysis

```r
require(ggplot2) #Load ggplot2 functions
```

```
## Loading required package: ggplot2
```

```r
require(dplyr) #Load dplyr functions
```

```
## Loading required package: 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
```

```r
require(reshape2) #Load reshape2 functions
```

```
## Loading required package: reshape2
```

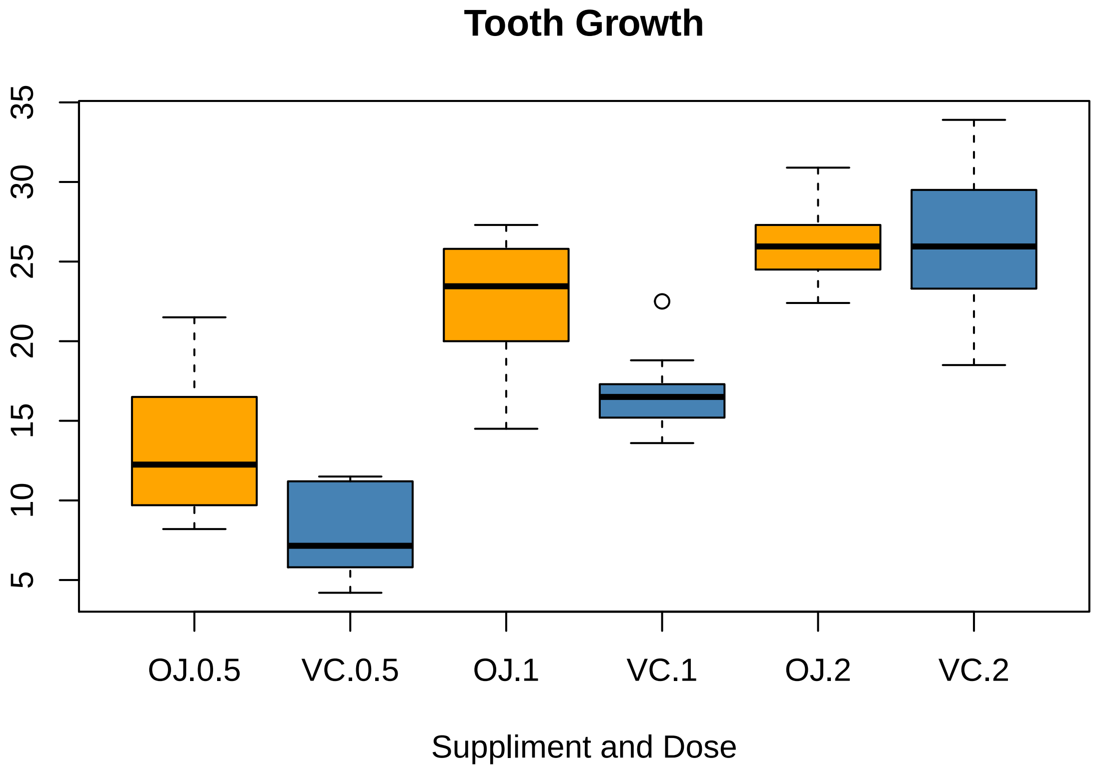
```r
data(ToothGrowth) #Load the dataset

tg.agg <- aggregate(ToothGrowth, by=list(ToothGrowth$supp, ToothGrowth$dose),
                    FUN=mean, na.rm=TRUE)
```

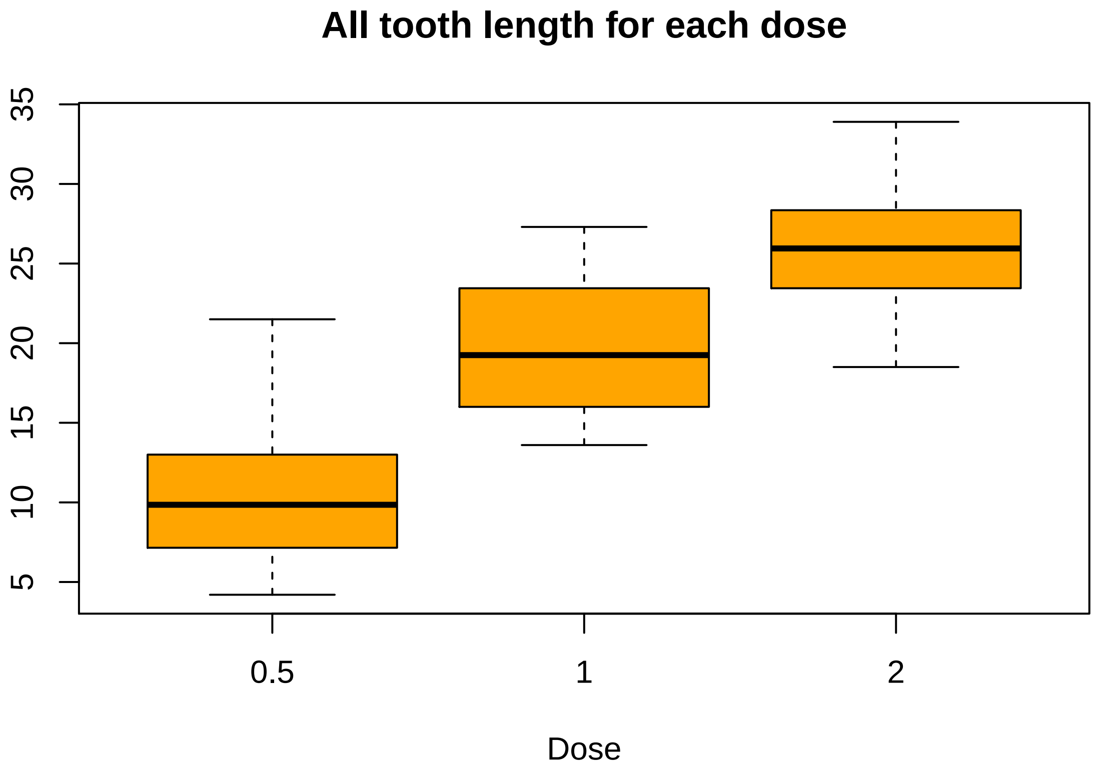
```
# Aggregate the data by supplement type and dose.
# Plot the results which are function of the supplement and the dose
g1 <- ggplot(tg.agg, aes(tg.agg$Group.2, tg.agg$len)) +
  geom_bar(aes(fill = tg.agg$Group.1), position = "dodge", stat="identity")

g1 <- g1+scale_fill_manual(values=c("orange", "steelblue"))

boxplot(len~supp*dose, data=ToothGrowth, notch=FALSE,
col=(c("orange","steelblue")),main="Tooth Growth", xlab="Suppliment and Dose")
```



```
# Aggregate the data by dose.
tg.agg.all <- aggregate(ToothGrowth, by=list(ToothGrowth$dose),FUN=mean,
                        na.rm=TRUE)
# Plot the results which are function of the dose
boxplot(len~dose, data=ToothGrowth, notch=FALSE,
col="orange",main="All tooth length for each dose", xlab="Dose")
```



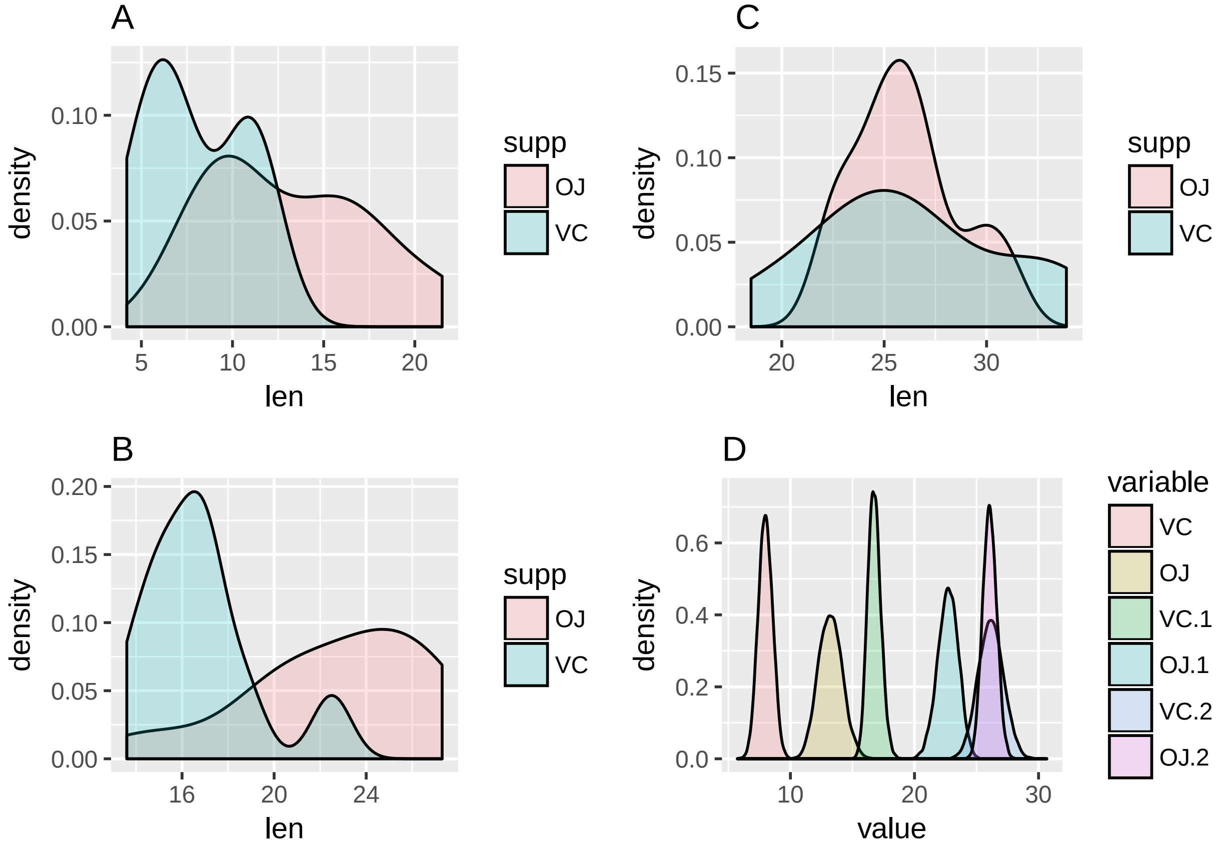
Next we utilize some more advanced data processing and visualising steps:

```
#Plot multiple data
g2 <- ggplot(data = ToothGrowth[ToothGrowth$dose == 0.5,],
             aes(len, fill = supp)) +
  geom_density(alpha = 0.2) + ggtitle("A")
g3 <- ggplot(data = ToothGrowth[ToothGrowth$dose == 1,],
             aes(len, fill = supp)) +
  geom_density(alpha = 0.2) + ggtitle("B")
g4 <- ggplot(data = ToothGrowth[ToothGrowth$dose == 2,],
             aes(len, fill = supp)) +
  geom_density(alpha = 0.2) + ggtitle("C")

#Utilize the bootstrap analysis
binded <- data.frame(nrow = 10000)
for (i in unique(ToothGrowth$dose)){
B = 10000
n = nrow(ToothGrowth[ToothGrowth$dose == i,])
boot.samples1 = matrix(sample(ToothGrowth[ToothGrowth$dose == i &
                                            ToothGrowth$supp == "VC" ,]$len,
                              size = B * n, replace = TRUE),B, n)
boot.samples2 = matrix(sample(ToothGrowth[ToothGrowth$dose == i &
                                            ToothGrowth$supp == "OJ" ,]$len,
                              size = B * n, replace = TRUE),B, n)
boot.statistics1 = apply(boot.samples1, 1, mean)
boot.statistics2 = apply(boot.samples2, 1, mean)
binded <- cbind(binded, data.frame("VC" = boot.statistics1,
                                              "OJ" = boot.statistics2))
}
binded <- binded[,-1]
melted <- melt(binded)
```

```
## No id variables; using all as measure variables
```

```
g5 <- ggplot(data = melted, aes(value, fill = variable)) +
  geom_density(alpha = 0.2)  + ggtitle("D")
multiplot(g2, g3, g4, g5, cols=2) #Plot the bootstrapped results
```



The *multiplot()* function was used to plot the results, see appendix for more information
Legends:
A: Dose level 0.5
B: Dose level 1
C: Dose level 2
D: Results of the bootstrapping

In the following part we are going to use t-tests to answer some questions about the relation of the dosage and supplement types.

Question 1: Does increasing the dose affect the length of the tooth?

At 0.95 significance level does increasing dose results in bigger tooth?

```
t <- vector()
```

```
t <- c(t, t.test(ToothGrowth[ToothGrowth$dose == 0.5,]$len,
                 ToothGrowth[ToothGrowth$dose == 1 ,]$len,
                 var.equal=FALSE, paired=FALSE)[3])
t <- c(t, t.test(ToothGrowth[ToothGrowth$dose == 1,]$len,
                 ToothGrowth[ToothGrowth$dose == 2 ,]$len,
                 var.equal=FALSE, paired=FALSE)[3])
t <- c(t, t.test(ToothGrowth[ToothGrowth$dose == 0.5,]$len,
                 ToothGrowth[ToothGrowth$dose == 2 ,]$len,
                 var.equal=FALSE, paired=FALSE)[3])
t
```

```
## $p.value
## [1] 1.268301e-07
##
## $p.value
## [1] 1.90643e-05
##
## $p.value
## [1] 4.397525e-14
```

Conclusion

All of the obtained t-values are significant, we can conclude that increasing the dosage results in bigger toothgrowth.

Question 2: Are there any significant differences between the supplements with fixed dose values?

```
t <- vector()
for (i in unique(ToothGrowth$dose)){
  t <- c(t, t.test(ToothGrowth[ToothGrowth$dose == i & ToothGrowth$supp == "VC" ,]$len,
                 ToothGrowth[ToothGrowth$dose == i & ToothGrowth$supp == "OJ" ,]$len,
                 var.equal=FALSE, paired=FALSE)[3])

}

t
```

```
## $p.value
## [1] 0.006358607
##
## $p.value
## [1] 0.001038376
##
## $p.value
## [1] 0.9638516
```

Conclusion

The difference between the two supplement is significant at 0.5 and 1.0 dosage levels, but not significant at the 2.0 dosage level.

Appendix

multiplot() function

The function was obtained from the following site: *http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/*