Lab 1

Sascha Strobl

7/8/2018

Problem 1

Taylor Series Expansion (Approximate Euler's number)

```
Taylor_expansion = function(n,x)
{a=1;for (i in 1:n){a=a+x^i/factorial(i)};a};

Taylor_expansion(100,1)
```

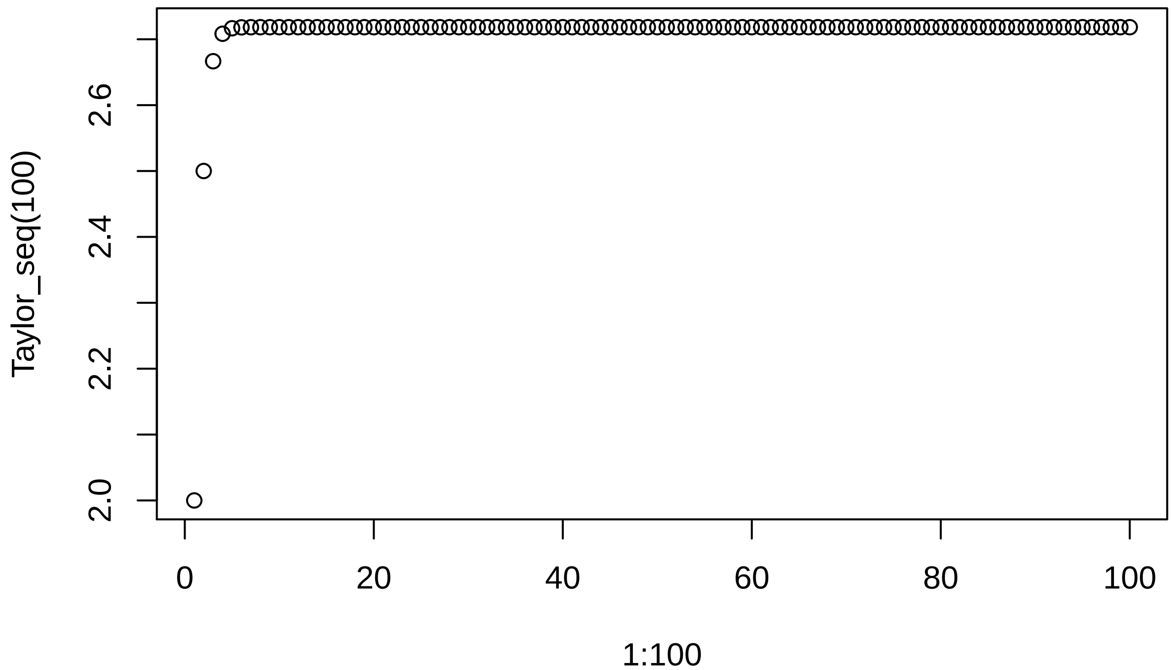
```
## [1] 2.718282
```

```
Taylor_seq = function (n)
{c=rep(0,n); for(i in 1:n){c[i]=Taylor_expansion(i,1)};
c
}

Taylor_seq(100)
```

```
##   [1] 2.000000 2.500000 2.666667 2.708333 2.716667 2.718056 2.718254
##   [8] 2.718279 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [15] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [22] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [29] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [36] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [43] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [50] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [57] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [64] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [71] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [78] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [85] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [92] 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282 2.718282
##  [99] 2.718282 2.718282
```

```
plot(1:100,Taylor_seq(100))
```



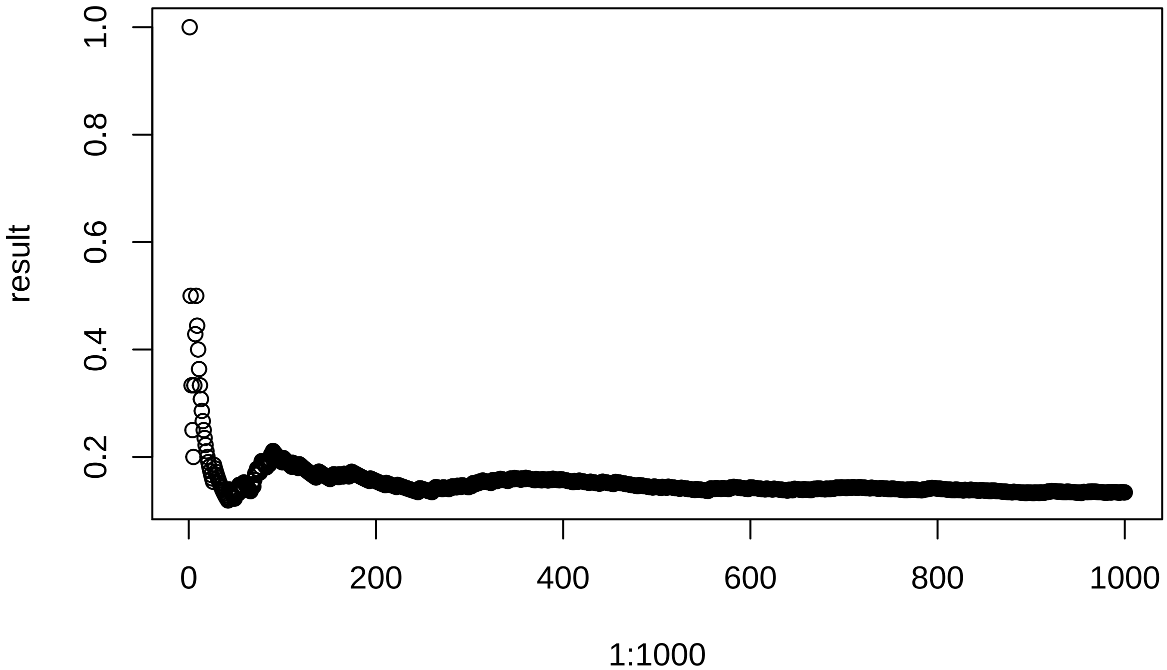
Problem 2

2 Dice have to equal 6

```
n=1000;
sum=0;
result = numeric(n)
for (i in 1:n){
  x1=sample(1:6,1,replace=T);
  x2=sample(1:6,1,replace=T);
  x=x1+x2;
  if (x==6){a=1} else{a=0}; sum=sum+a;
  result[i]=sum/i;}
mean(result)
```

```
## [1] 0.1515779
```

```
plot(1:1000, result)
```

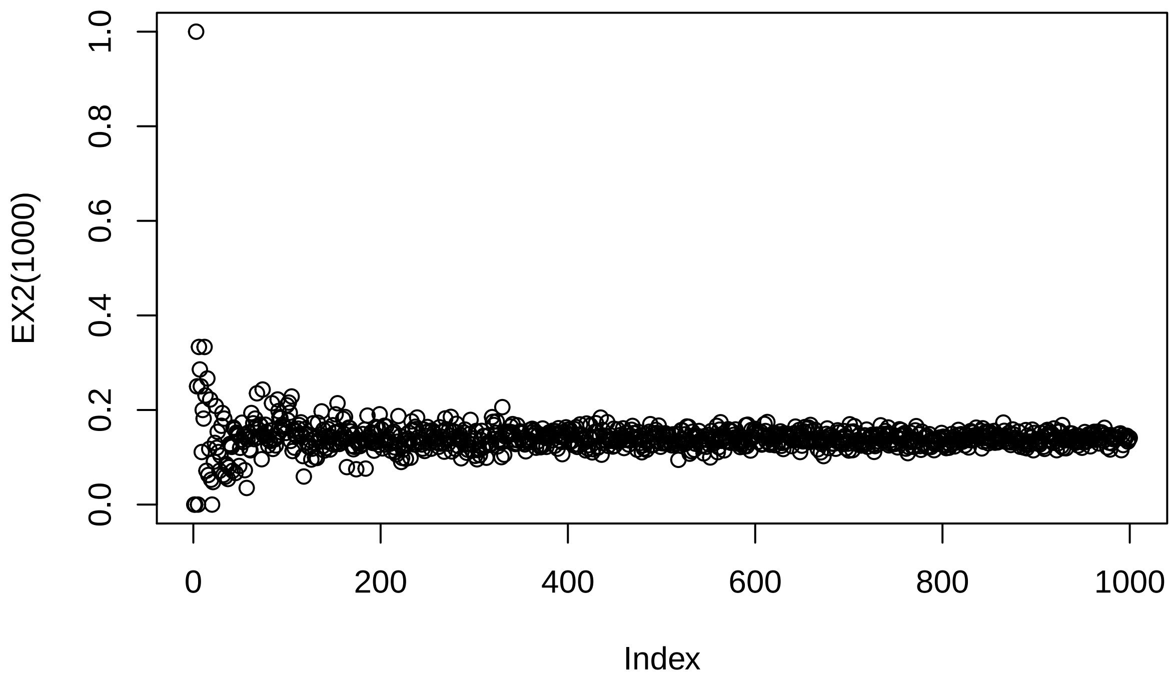


Create a function to compute average number of the events "sum equals 6".

```
EX2 = function (n)
{u=numeric(n);
for(i in 1:n){
  # Outcome (number of 6) of the first die among i rolls.
  X1=sample(1:6,i,replace=TRUE)[1:i];
  # Outcome (number of 6) of the second die among i rolls.
  X2=sample(1:6,i,replace=TRUE)[1:i];
  # Number of the events "sum equals 6" over the total number of rolls i.
  u[i]=sum((X1+X2)==6)/i}; u}
```

Use figure to illustrate the law of large numbers.

```
plot(EX2(1000))
```



Problem 3

Median, mean and is the variance of the mean or median lower

a)

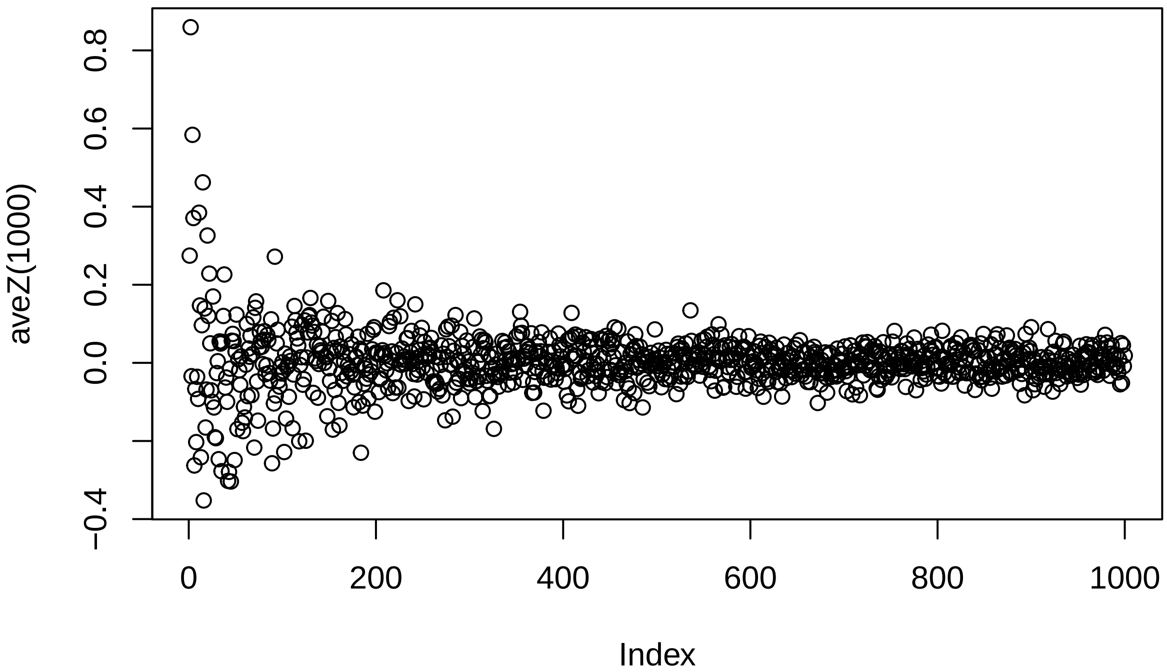
```
n=5
rnorm(n)
```

```
## [1] -0.4203260  1.1411076 -0.1236702 -1.5450058 -0.3237903
```

b) Plot the sequence to illustrate the law of large numbers.

```
aveZ = function (n){
  m=numeric(n);for(i in 1:n){m[i]=mean(rnorm(i))
  }
  m
}

plot(aveZ(1000))
```



c) Variance of the mean is lower than the variance of the median

Mean:

```
n=1000
b=0
for (i in 1:n){
  a=median(rnorm(100));
  b=b+a};
mean1=b/n;
mean1
```

```
## [1] 0.00326291
```

Median:

```
n=1000
for (i in 1:n){
  a[i]=median(rnorm(100));
  v=var(a);};
v
```

```
## [1] 0.01563109
```

```
n=1000
for (i in 1:n){
  a[i]=mean(rnorm(100));
  v2=var(a)};

v2
```

```
## [1] 0.01002843
```

Problem 4

Linear Regression and prediction

A Walmart supermarket has 15 cashiers (but they are not all in service). The table below describes the corresponding customer average waiting time VS the number of cashiers in service.

```
cashiers=c(3, 4, 5, 6, 8, 10, 12);
wtime=c(16, 12, 9.6, 7.9, 6, 4.7, 4);
lm.fit=lm(wtime~cashiers);
summary(lm.fit);
```

```
##
## Call:
## lm(formula = wtime ~ cashiers)
##
## Residuals:
##        1        2        3        4        5        6        7
##  2.73744 -0.05374 -1.24493 -1.73612 -1.21850 -0.10088  1.61674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.8890     1.6753  10.081 0.000164 ***
## cashiers     -1.2088     0.2233  -5.413 0.002910 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.798 on 5 degrees of freedom
## Multiple R-squared:  0.8543, Adjusted R-squared:  0.8251
## F-statistic: 29.31 on 1 and 5 DF,  p-value: 0.00291
```

```
confint(lm.fit);
```

```
##                 2.5 %     97.5 %
## (Intercept) 12.582615 21.1953590
## cashiers    -1.782811 -0.6348098
```

```
newdata = data.frame(cashiers=15)
predict(lm.fit, newdata, se.fit = TRUE)
```

```
## $fit
##         1
## -1.243172
##
## $se.fit
## [1] 1.941153
##
## $df
## [1] 5
##
## $residual.scale
## [1] 1.798291
```

```
predict(lm.fit, newdata, interval = "prediction")
```

```
##         fit       lwr      upr
## 1 -1.243172 -8.045228 5.558884
```

Problem 5

a) Predict new data with Discriminant analysis (QDA)

```r
library(MASS);
Gender=c(0,1, 1, 1, 0, 0); # 0 = male; 1 = female
lweight=c(5.00, 4.70, 4.40, 5.12, 4.30, 5.44);
qda.fit=qda(Gender~lweight)
newdata = data.frame(lweight=4.9)
qda.pred=predict(qda.fit, newdata)
qda.pred
```

```
## $class
## [1] 1
## Levels: 0 1
##
## $posterior
##           0         1
## 1 0.4095252 0.5904748
```

b) K-nearest neighbors

```r
library(class);
train.X=lweight;
test.X=c(4.9);
train.Direction=Gender;
set.seed(1);
knn.pred=knn(train.X,test.X,train.Direction,k=4);
knn.pred
```

```
## [1] 1
## Levels: 0 1
```