Atelier autour des probabilités et de la statistique
Journée de formation aux nouveaux programmes de terminales

Octobre 2012

Loi normale et calcul de probabilité

Dans une fabrique de peintures décoratives, une machine remplit automatiquement avec de la peinture des bidons de 51 centilitres.

Pour pouvoir être commercialisé, un bidon doit contenir au moins 48 centilitres de peinture.

La quantité de peinture en centilitres fournie par la machine peut être modélisée par une variable aléatoire X qui suit une loi normale de moyenne μ et d'écart-type $\sigma = 1{,}2$.

1) La machine est réglée sur $\mu = 50$.

   a) Quel est le pourcentage de bidons qui ne pourront pas être commercialisés ?

   b) Quel est le pourcentage de bidons qui risquent de déborder au moment du remplissage ?

   c) Quelle devrait être la contenance, au millilitre près, des bidons pour que moins de 10 % des bidons ne débordent ?

2) Réglage de la machine.

Quelle devrait être la quantité moyenne maximale de remplissage de sorte que moins de 10 % des bidons ne débordent ?

1) a) $P(X < 48) = 0{,}5 - P(48 < X < 50) \approx 0{,}048$

Donc environ 4,8 % des bidons ne pourront pas être commercialisés.

1) b) $P(X > 51) = 0{,}5 - P(50 < X < 51) \approx 0{,}202$

Donc environ 20,2 % des bidons vont déborder au cours du remplissage.

1) c) On utilise la fonction inverse normale qui détermine le réel t vérifiant $P(X < t) = \alpha$

Ici on souhaite déterminer b de sorte que $P(X > b) < 0{,}1$ donc $P(X < b) > 0{,}9$.

La calculatrice donne $t \approx 51{,}54$ de sorte que $P(X < t) = 0{,}9$.

Il faudrait donc que les bidons puissent contenir 51,6 centilitres de peinture.

2) Il s'agit de déterminer la valeur maximale de μ de sorte que

$$P(X > 51) < 0{,}1 \text{ donc } P(X < 51) > 0{,}9$$

On introduit la variable centrée réduite Z et on cherche t de sorte que $P(Z < t) = 0{,}9$;

on obtient $t \approx 1{,}28155$ ainsi μ doit vérifier l'inéquation $\dfrac{51-\mu}{\sigma} > 1{,}28155$

ce qui donne une valeur maximale de μ de $51 - 1{,}28155 \times 1{,}2$ donc de 49,5 au ml près.

Remarque : *pour accéder à la valeur de t, on peut proposer plusieurs méthodes :*

- *Méthodes-élève : tâtonnement calculatrice, tableur, geogebra (module « calcul probabilité »)...*
- *Méthode-prof : tables (réactiver pratiques d'étudiant mais hors-programme pour les élèves de terminale)*

Intervalle de fluctuation asymptotique et prise de décision

1) Déterminer l'intervalle de fluctuation asymptotique *J* au seuil de 95% pour la proportion de grains de riz marron.

$0{,}2 - 1{,}96\sqrt{\frac{0{,}2 \times 0{,}8}{690}} \approx 0{,}170\,15$ *et* $0{,}2 + 1{,}96\sqrt{\frac{0{,}2 \times 0{,}8}{690}} \approx 0{,}229\,84$

Aves des valeurs approchées à 10^{-2} *près (par défaut pour la première et excès pour la seconde), on prend* $J = [0{,}17 ; 0{,}23]$.

2) Calculer la proportion de grains de riz marron dans l'échantillon. Que peut-on en conclure ?

$f = \frac{140}{690}$ *c'est-à-dire* $f \approx 0{,}20$.

$f \in J$ *donc d'après la règle de décision choisie, on ne peut pas rejeter l'hypothèse selon laquelle les grains de riz représentent 20% de la totalité des grains de riz.*

3) Dans le même échantillon, il y avait 152 grains de riz noir pour une proportion annoncée de 20% et 125 grains de riz rouge, pour une proportion annoncée de 10%. Que peut-on conclure de ces résultats ?

Pour les grains de riz noir, $f' = \frac{152}{690} \approx 0{,}22$.

Donc $f' \in J$ *et on ne peut pas rejeter l'hypothèse selon laquelle les grains de riz noir représentent 20% de la totalité des grains de riz.*

Pour les grains de riz rouge, $\left[0{,}1 - 1{,}96\sqrt{\frac{0{,}1 \times 0{,}9}{690}} ; 0{,}1 + 1{,}96\sqrt{\frac{0{,}1 \times 0{,}9}{690}}\right] \approx [0{,}07 ; 0{,}13]$

Or $f' = \frac{125}{690} \approx 0{,}18$ *et* $f' \in [0{,}07 ; 0{,}13]$ *donc on rejette l'hypothèse selon laquelle les grains de riz rouge représentent 10% de la totalité des grains de riz.*

4)

```
VARIABLES
  p EST_DU_TYPE NOMBRE
  f EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  borne_inf EST_DU_TYPE NOMBRE
  borne_sup EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE p
 LIRE n
  LIRE f
  borne_inf PREND_LA_VALEUR p-1.96*sqrt(p*(1-p)/n)
  borne_sup PREND_LA_VALEUR p+1.96*sqrt(p*(1-p)/n)
  SI (f>=borne_inf ET f<=borne_sup) ALORS
    DEBUT_SI
    AFFICHER "Hypothèse acceptée"
    FIN_SI
    SINON
      DEBUT_SINON
      AFFICHER "Hypothèse rejetée"
      FIN_SINON
FIN_ALGORITHME
```

Loi uniforme en géométrie

Cas 1 : P (S) = (π / 6) / (π / 2) = 1 / 3

```
VARIABLES
  k EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  c EST_DU_TYPE NOMBRE
  t EST_DU_TYPE NOMBRE
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  d EST_DU_TYPE NOMBRE
  f EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  //n est le nombre de simulations
  //c est le compteur du nombre de succès
  //t est le réel associé au point M
  //x est l'abscisse de M
  //y est l'ordonnée de M
  //d est la distance JM
  //f est la fréquence de succès pour n simulations
  LIRE n
  c PREND_LA_VALEUR 0
  POUR k ALLANT_DE 1 A n
    DEBUT_POUR
    t PREND_LA_VALEUR Math.PI*random()/2
    x PREND_LA_VALEUR cos(t)
    y PREND_LA_VALEUR sin(t)
    d PREND_LA_VALEUR sqrt(x*x+(y-1)*(y-1))
    SI (d>1) ALORS
      DEBUT_SI
      c PREND_LA_VALEUR c+1
      TRACER_POINT (x,y)
      TRACER_POINT (x,0)
      TRACER_POINT (0,y)
      FIN_SI
      SINON
        DEBUT_SINON
        TRACER_POINT (x,y)
        TRACER_POINT (x,0)
        TRACER_POINT (0,y)
        FIN_SINON
    FIN_POUR
  f PREND_LA_VALEUR c/n
  AFFICHER "nombre de simulations = "
  AFFICHER n
  AFFICHER "fréquence de succès = "
  AFFICHER f
FIN_ALGORITHME
```

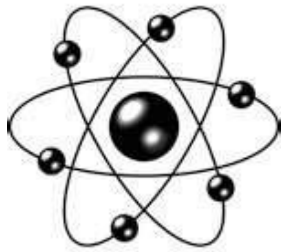
Cas 2 : P (S) = 1 – $\sqrt{3}$ / 2

```
VARIABLES
  k EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  c EST_DU_TYPE NOMBRE
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  d EST_DU_TYPE NOMBRE
  f EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  //n est le nombre de simulations
  //c est le compteur du nombre de succès
  //x est l'abscisse de M
  //y est l'ordonnée de M
  //d est la distance JM
  //f est la fréquence de succès pour n simulations
  LIRE n
  c PREND_LA_VALEUR 0
  POUR k ALLANT_DE 1 A n
    DEBUT_POUR
    x PREND_LA_VALEUR random()
    y PREND_LA_VALEUR sqrt(1-x*x)
    d PREND_LA_VALEUR sqrt(x*x+(y-1)*(y-1))
    SI (d>1) ALORS
      DEBUT_SI
      c PREND_LA_VALEUR c+1
      TRACER_POINT (x,y)
      TRACER_POINT (x,0)
      TRACER_POINT (0,y)
      FIN_SI
      SINON
        DEBUT_SINON
        TRACER_POINT (x,y)
        TRACER_POINT (x,0)
        TRACER_POINT (0,y)
        FIN_SINON
    FIN_POUR
  f PREND_LA_VALEUR c/n
  AFFICHER "nombre de simulations = "
  AFFICHER n
  AFFICHER "fréquence de succès = "
  AFFICHER f
FIN_ALGORITHME
```

Cas 3 : P (S) = 1 / 2

```
VARIABLES
  k EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  c EST_DU_TYPE NOMBRE
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  d EST_DU_TYPE NOMBRE
  f EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  //n est le nombre de simulations
  //c est le compteur du nombre de succès
  //x est l'abscisse de M
  //y est l'ordonnée de M
  //d est la distance JM
  //f est la fréquence de succès pour n simulations
  LIRE n
  c PREND_LA_VALEUR 0
  POUR k ALLANT_DE 1 A n
    DEBUT_POUR
    y PREND_LA_VALEUR random()
    x PREND_LA_VALEUR sqrt(1-y*y)
    d PREND_LA_VALEUR sqrt(x*x+(y-1)*(y-1))
    SI (d>1) ALORS
      DEBUT_SI
      c PREND_LA_VALEUR c+1
      TRACER_POINT (x,y)
      TRACER_POINT (x,0)
      TRACER_POINT (0,y)
      FIN_SI
      SINON
        DEBUT_SINON
        TRACER_POINT (x,y)
        TRACER_POINT (x,0)
        TRACER_POINT (0,y)
        FIN_SINON
    FIN_POUR
  f PREND_LA_VALEUR c/n
  AFFICHER "nombre de simulations = "
  AFFICHER n
  AFFICHER "fréquence de succès = "
  AFFICHER f
FIN_ALGORITHME
```

Désintégration radioactive et simulation



Un nombre N_0 d'atomes radioactifs identiques ont chacun la probabilité *p* de se désintégrer au cours d'une unité de temps.

Objectif : simuler ces désintégrations, et tracer l'histogramme des fréquences de désintégration par unité de temps. Tracer la courbe de la fonction de densité associée.

Algorithme de simulation

Dans le morceau d'algorithme ci-contre :

- la variable `N` représente le nombre d'atomes radioactifs à un instant donné ;
- `Nnew` compte les atomes « survivants » après une unité de temps ;
- *p* est la probabilité de désintégration d'un atome.

```
POUR compteur ALLANT_DE 1 A N
   DEBUT_POUR
   SI (random()<1-p) ALORS
      DEBUT_SI
      Nnew PREND_LA_VALEUR Nnew+1
      FIN_SI
   FIN_POUR
```

Compléter cet algorithme de manière à tracer l'histogramme des fréquences de désintégration par unité de temps, sur une durée à votre convenance.

```
VARIABLES
  N EST_DU_TYPE NOMBRE
  Nnew EST_DU_TYPE NOMBRE
  compteur EST_DU_TYPE NOMBRE
  p EST_DU_TYPE NOMBRE
  t EST_DU_TYPE NOMBRE
  N0 EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  //Initialisation du temps
  t PREND_LA_VALEUR 0
  //Nombre d'atomes radioactifs originel
  LIRE N0
  //N est le nombre d'atomes non encore désintégrés
  N PREND_LA_VALEUR N0
  //Probabilité de désintégration d'un atome par unité de temps
  LIRE p
  //Tant qu'il reste des atomes radioactifs :
  //potentiellement, cet algorithme pourrait ne jamais se terminer
  TANT_QUE (N>0) FAIRE
    DEBUT_TANT_QUE
    //Nnew compte les atomes non désintégrés en 1 unité de temps
    Nnew PREND_LA_VALEUR 0
    //Chaque atome radioactif se désintègre avec la probabilité p
    POUR compteur ALLANT_DE 1 A N
      DEBUT_POUR
      SI (random()<1-p) ALORS
        DEBUT_SI
        Nnew PREND_LA_VALEUR Nnew+1
        FIN_SI
      FIN_POUR
    //Tracé du rectangle de l'histogramme
    TRACER_SEGMENT (t,(N-Nnew)/N0)->(t+1,(N-Nnew)/N0)
    TRACER_SEGMENT (t+1,0)->(t+1,(N-Nnew)/N0)
    //Pour l'étape suivante, N devient le nb d'atomes non désintégrés
    N PREND_LA_VALEUR Nnew
    //Incrémentation du temps
    t PREND_LA_VALEUR t+1
    FIN_TANT_QUE
FIN_ALGORITHME
```