Eine Ferienwohnanlage hat 80 Wohnungen. Erfahrungsgemäß werden 15% der Buchungen wieder storniert. Der Besitzer nimmt für die Pfingstferien 95 Buchungen an.

a) Berechnen Sie die Wahrscheinlichkeit, dass zu viele Buchungen angenommen wurden.

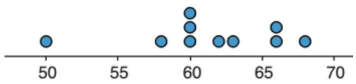
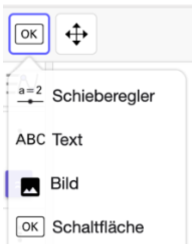
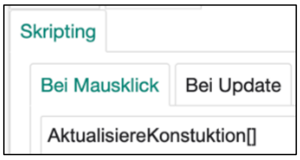
b) Bestimmen Sie die Anzahl an Buchungen, die der Besitzer für die Pfingstferien annehmen sollte, damit die Wahrscheinlichkeit, dass zu viele Buchungen angenommen wurden, kleiner als 5% ist.

Entnommen aus: Lambacher Schweizer. Mathematik Qualifikationsphase. Leistungskurs. S: 286. Niedersachsen. 2018

Dieser Aufgabentyp findet sich in vielen Schulbüchern und auch in Abituraufgaben. Wie sinnvoll ist es eigentlich, mit einer Binomialverteilung zu modellieren. Nennen Sie Gründe, die dagegensprechen.

Simulation und exakte Berechnung mit GeoGebra:

1. Simulation

Bin=Folge(ZufallszahlBinomialverteilt(n, p), i, 1, w) Bin = Folge(ZufallszahlBinomialverteilt(100, 0.6), i, 1, 10) = {62, 60, 63, 68, 66, 50, 60, 60, 66, 58}	w Realisationen einer binomialverteilten Zufallsvariablen mit den Parametern n und p. Die Ausgabe wird in der Liste mit Namen *Bin* gespeichert, wobei der Name der Liste beliebig wählbar ist.
DotPlot(Bin, dy) l1 = DotPlot(Bin, 0.2) = {(50, 0.2), (58, 0.2), (60, 0.2), (60, 0.4), (60, 0.6) 	PunktePlot der Werte aus Liste *Bin*. Mit der Variablen dy (Name beliebig wählbar) wird der Abstand übereinanderliegender Punkte gesteuert (optional).
AktualisiereKonstruktion[] 	Definition einer Schaltfläche, um damit auf Knopfdruck eine Neuberechnung einzuleiten und somit neue (Pseudo)-Zufallszahlen zu erzeugen. Es muss unter *Skripting*|*Bei Mausklick* der Befehl *AktualisiereKonstruktion[]* eingegeben werden. Tipp: Nachdem im Menü *OK Schaltfläche* ausgewählt wurde, muss im Grafikfenster eine Stelle ausgewählt werden, z.B. per Mausklick. Es geht auch mit dem Tastenbefehl *crl R (bzw. cmd R).*

Bemerkungen zur Arbeit mit Listen:

1. Aus absolut wird relativ –mit einem Befehl

   Um die zugehörigen relativen Trefferanzahlen zu erhalten, muss der folgende Befehl eingegeben werden:

$$l3 = \frac{\text{Folge(ZufallszahlBinomialverteilt(100, 0.6), i, 1, 10)}}{100}$$

≈{0.53, 0.65, 0.59, 0.65, 0.56, 0.62, 0.56, 0.65, 0.6, 0.6}

Listen sind in Programmiersprachen effektive Objekte. Operationen mit Listenelementen müssen nicht einzeln erfolgen. So muss hier nicht mit einer Schleife jedes einzelne Listenelement durch die Stichprobenanzahl dividiert werden. In unserem Beispiel liefert die Eingabe *Bin/100* das Gleiche, da die absoluten Trefferanzahlen in der Liste *Bin* gespeichert sind.
Sollen nur relative Trefferanzahlen erzeugt werden, kann man auch folgendermaßen vorgehen:

$$h_n = \text{Folge}\left(\frac{\text{ZufallszahlBinomialverteilt}(100, 0.6)}{100}, i, 1, 10\right)$$

$$= \{0.5, 0.59, 0.61, 0.52, 0.56, 0.62, 0.67, 0.6, 0.61, 0.57\}$$

2. Das Zählen leicht gemacht - mit booleschen Variablen
   Um in der Liste h_n die Anzahl an Elementen zu erhalten, deren Wert größer als 0.6 ist, kann zuerst der Befehl $h_n > 0.6$ eingegeben werden. Hiermit wird eine neue Liste mit den booleschen Werten *false* und *true* erzeugt.

$$l4 = h_n > 0.6$$

$$= \{\text{false, false, true, false, false, true, true, false, true, false}\}$$

Der Wert *false* wird intern als 0, der Wert *true* als 1 identifiziert. In der Stochastik hat man übrigens dafür die Indikatorfunktion eingeführt. Nun muss nur noch die Summe gebildet werden. Dies geschieht mit dem Befehl *Summe(l4)*.
Dies kann auch in einem Schritt geschehen:

$$\text{Anzahl} = \text{Summe}(h_n > 0.6)$$

$$= 4$$

Um binomialverteilte Zufallszahlen zu erzeugen, kann neben der in GeoGebra definierten Funktion *ZufallszahlBinomialverteil(n,p)* auch so vorgegangen werden:

$$H = \text{Summe}(\text{Folge}(\text{random}() < 0.6, i, 1, 100))$$

$$= 68$$

$$H_n = \text{Folge}(H, i, 1, 10)$$

$$= \{54, 52, 58, 60, 57, 67, 61, 54, 63, 60\}$$

$$\text{BinZufall} = \text{Folge}(\text{Summe}(\text{Folge}(\text{random}() < 0.6, i, 1, 100)), i, 1, 10)$$

$$= \{62, 65, 63, 63, 59, 63, 63, 61, 70, 57\}$$

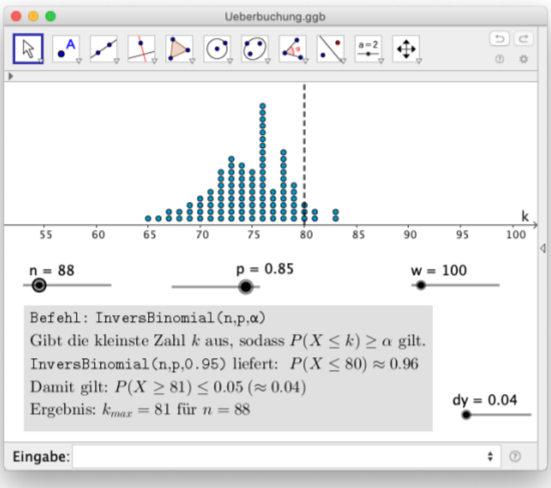
Entweder in zwei Schritten oder wie in der letzten Zeile mit einem Befehl. Der Befehl *random()* liefert eine Zufallszahl zwischen 0 und 1, *random()* < 0.6 entweder *false* (0) oder *true* (1). Der Folge-Befehl realisiert damit eine zufällige 0-1-Folge, die dann nur noch aufsummiert wird.
Warum nicht so die Formel für die Binomialverteilung einführen – ohne Baumdiagramme: $H = 68$ bedeutet, dass bei 100 Versuchen 68-mal eine 1 (Treffer) und 32-mal eine 0 (Niete) erschienen ist. Denken wir uns diese 0-1-Folge als 100-Tupel, so kann man nun fragen, wie viele solcher Tupel es überhaupt gibt. Es gibt insgesamt $\binom{100}{68}$ verschiedene

Tupel mit 68 Einsen und 32 Nullen. Damit ergibt sich sofort die Wahrscheinlichkeit, so ein Tupel zu erhalten:

$$P(X = 68) = \binom{100}{68} p^{68}(1-p)^{32}.$$

2. Berechnungen

InversBinomial(n,p,alpha)	Gibt die kleinste Zahl k aus, sodass $$P(X \leq k) \geq \alpha$$ gilt. X ist eine binomialverteilte Zufallsvariable. Es gilt: $$P(X \leq k) \geq \alpha \Leftrightarrow P(X \geq k+1) < 1-\alpha$$ In dem Beispiel wird ein n mit $P(X \geq 81) \leq 0{,}05$ gesucht (mehr Buchungen als Wohnungen). InversBinomial(n,p,0.95) liefert $k = 80$. Es muss somit der Wert für n gesucht werden, damit InversBinomial den Wert 80 liefert. Für $n = 88$ ergibt sich $P(X \geq 81) \leq 0{,}05$ (≈ 0.04).



Der *InversBinomial()*-Befehl ist schon speziell und nicht einfach zu verstehen. Man kann den Wert für n auch durch sinnvolles Probieren erhalten. Sind Prognoseintervalle (Sigma-Umgebungen) bekannt, kann die rechte Grenze des zugehörigen 90%-Prognoseintervalle dienen. Dieser Ansatz liefert die Bestimmungsgleichung

$$n \cdot 0{,}85 + 1{,}64\sqrt{n \cdot 0{,}85 \cdot (1-0{,}85)} = 81,$$

die graphisch durch Schnittpunktbestimmung gelöst werden kann (Ergebnis: 88,801…). Mit

$$1 - 1 - P_{89}(X \leq 80) = P_{89}(X \geq 81) = 0{,}0684 \ldots > 0{,}05 \text{ und } P_{88}(X \geq 81) = 0{,}0368 \ldots < 0{,}05$$

ergibt sich $n = 88$.

Das sinnvolle Probieren kann auch automatisiert werden:

```
l1 = Folge(1 − Binomial(n, 0.85, 80, true) < 0.05, n, 1, 100)
 = {true, true, true, true, true, true, true, true, true, true, true, true, true, true,

a = Summe(l1)
 = 88

l2 = {1 − Binomial(88, 0.85, 80, true), 1 − Binomial(89, 0.85, 80, true)}
 = {0.04, 0.07}
```

Bemerkung: *1-Binomial(88, 0.85, 80, true)* liefert den Wert für $P(X \geq 81)$; $n = 88$; $p = 0{,}85$.

Simulationen fördern in diesem Fall ein tieferes Verständnis für solche Problemstellungen. Dieser Weg erleichtert das Finden des folgenden Ansatzes:

Gesucht wird der größte Wert für n, sodass $P(X \geq 81) \leq 0{,}05$ gilt. Da in GeoGebra die Funktion InversBinomial existiert, kann der Wert sofort bestimmt werden.

Werden mehrere Simulationen für $n = 90, 89, \ldots, 85$ durchgeführt, so kann schon erahnt werden, dass $n = 88$ ein sinnvoller Wert ist.
Weiterhin wird der Fokus stärker auf die Variabilität des Zufalls gelegt. Die Anzahl an Simulationen sollte auch erhöht werden, um die Auswirkungen auf die Verteilungen zu erleben.