Rechnernutzung in der Physik

Institut für Experimentelle Teilchenphysik
Institut für Theoretische Teilchenphysik
Interfakultatives Institut für Anwendungen der Informatik

Dr. Th. Kuhr, Prof. U. Husemann, Prof. M. Steinhauser
Mildenberger, Hoff, Hermann, Heck
<http://comp.physik.kit.edu>

WS2012/13 – Blatt 11

Prog.: Di, 15.01.2012 / Ausarb.: Fr., 18.01.2012

Zufallszahlen, Histogramme und Verteilungsdichten

Wie Sie es schon vom letzten Blatt kennen, gibt es auch zu dieser Übung eine Vorlage, `ex11.cc`.

Aufgabe 25: Korrelationen

Programmtestat

In dieser Aufgabe soll die Verteilung der Bin-Inhalte eines Histogramms sowie die Korrelation zwischen ihnen näher untersucht werden.

Verteilung von Bin-Inhalten

Füllen Sie ein Histogramm mit 5 Bins mit 100 gleichverteilten Zufallszahlen und wiederholen Sie das Experiment 10.000 mal (Schleife über den Teil Ihres Codes zum Füllen des Histogramms). Untersuchen Sie die Häufigkeitsverteilung der einzelnen Bin-Inhalte und deren Korrelationen, indem Sie sie nach jedem Experiment in ein- bzw. zweidimensionale Histogramme füllen.

Hinweis und Hilfe: Erzeugen Sie ein Histogramm für gleichverteilte Zufallszahlen (analog Aufgabe 23 oder 24 vom letzten Blatt) mit 5 Bins zwischen 0 und 1. Ein Histogramm mit Variablennamen `id` kann folgendermaßen zurückgesetzt werden:

```
id->Reset();
```

Erzeugen Sie jetzt je ein weiteres Histogramm, in das Sie in jedem Experiment den Inhalt n_i des Bins i füllen. Der Bin-Inhalt eines Histogramms kann mit der Methode `GetArray()` ausgelesen werden, die einen Zeiger auf ein Feld zurückliefert:

```
Float_t *content = id->GetArray();
```

Der Inhalt des Feldes ist wie folgt:

```
content[0]    number of underflows
content[1]    number of entries in bin 1
...
content[n]    number of entries in bin n (last bin)
content[n+1]  number of overflows
```

Welche Verteilung der Bin-Inhalte erwarten Sie? Vergleichen Sie, indem Sie die erwartete Verteilung in dasselbe Diagramm eintragen wie die Verteilung der Bin-Inhalte (*Tipp:* die ROOT-Klasse `TMath` anschauen).

Korrelation von Bin-Inhalten

Um Korrelationen zwischen den Bin-Inhalten zu untersuchen, füllt man einen *Scatterplot* mit den Inhalten zweier beliebiger Bins n_i und n_j (sagen wir Bin 2 und Bin 4). Dabei kann man die Histogramm-Klasse `TH2F` verwenden. Wenn sie gezeichnet wird (`Draw`), werden die Daten auf eine Weise dargestellt, die bei einer großen Zahl Bins äquivalent zu einem *Scatterplot* ist.

Hilfe: Das Erzeugen einer Instanz eines zweidimensionalen Histogramms geht so:

```
TH2F *hscat = new TH2F("hscat","some title"
                      NumberBinsX,XMin,XMax,NumberBinsY,YMin,YMax);
```

Füllen geht analog zum eindimensionalen Fall mit der Methode `Fill`: `id->Fill(x,y)`;

Die Korrelation bzw. Kovarianz lässt sich in ROOT ausgeben mit

```
id->GetCorrelationFactor() bzw. id->GetCovariance().
```

Stellen Sie den Zusammenhang der Einträge in den Bins 2 und 4 als Scatter-Plot dar, und

lassen Sie sich die Korrelation ausgeben. Wie groß ist die beobachtete Korrelation? Haben Sie eine Erklärung dafür?

Zusätzlich zu den zweidimensionalen Histogrammen gibt es in ROOT das sogenannte *profile histogram*, Klasse `TProfile`. Damit lässt sich die Korrelation deutlicher darstellen:

```
TProfile *hprof=new TProfile("hprof","title",NBinsX,XMin,XMax,YMin,YMax);
```

Füllen geht genau wie im Fall eines zweidimensionalen Histogramms.

Stellen Sie den Zusammenhang der Einträge in den Bins 2 und 4 als Profile-Histogramm dar, und erklären Sie die Grafik.

Aufgabe 26: Verteilung von Bin-Inhalten (2)

freiwillig

Modifizieren Sie das Histogramm aus Aufgabe 25 so, dass die gleichverteilten Zufallszahlen in ein Histogramm mit 100 Bins eingetragen werden. Füllen Sie in einer Schleife 10.000 mal je 2000 Zufallszahlen. Der Erwartungswert für jedes Bin ist nun wieder 20, genau wie oben. Vergleichen Sie die Verteilung der Bin-Inhalte durch Überlagerung geeignet normierter Verteilungen mit der Binomial- und der Poissonverteilung (*Tipp*: siehe Klasse `TMath`).

Aufgabe 27: Überprüfung der naiven Fehlerfortpflanzung

Ausarbeitung

Für die Messung einer Größe $Y = 0.5$ mit Messunsicherheit ± 0.1 gebe es eine Vorhersage einer als Standard anerkannten Theorie, $T = 1$ mit Unsicherheit ± 0.3 . Beide Unsicherheiten seien als gaußförmig angenommen (was bei der Theorie-Unsicherheit eine ziemlich gewagte Behauptung ist!). Zum Vergleich verwendet man gerne Verhältnisse, $r_1 = \frac{Y}{T}$ oder $r_2 = \frac{T}{Y}$, man könnte auch die Differenz $d = T - Y$ angeben. Von der Entdeckung „neuer Physik“ würde man sprechen, wenn r_1 oder r_2 statistisch signifikant von 1 bzw. d von 0 abweicht. Die Signifikanz, S , drückt man gerne als den auf die Unsicherheit normierten Abstand zwischen erwartetem und beobachtetem Wert aus.

Berechnen Sie mit Hilfe des Fehlerfortpflanzungsgesetzes Wert und Unsicherheit von r_1 , r_2 und d , und geben Sie die Signifikanzen S_1 , S_2 und S_d an.

Simulieren Sie nun diese Fälle in einer „Toy-Monte-Carlo“-Studie, indem Sie die Verteilungen von $r_{1,2}$ und d mit Hilfe von gaußverteilten Zufallszahlen bestimmen. Standard-normalverteilte Zufallszahlen erhalten Sie mit dem Befehl `x=gRandom->Gaus()`;

Vergleichen Sie jeweils die Verteilungen mit den Gaußverteilungen, die den Parametern aus der Fehlerfortpflanzung entsprechen, indem Sie sie jeweils in dasselbe Diagramm eintragen. Kommentieren Sie Ihre Beobachtungen. Welche der Größen r_1 , r_2 bzw. d würden Sie verwenden, um die Übereinstimmung zwischen Messung und theoretischer Erwartung zu quantifizieren?

Hinweis: Falls nötig, können Sie sich zur Lösung an der Funktion `erprop()` im Vorlagenmacro orientieren.

Aufgabe 28: Überprüfung der naiven Fehlerfortpflanzung II

freiwillig

Modernen Programmiersprachen wie Python kommt in der Datenanalyse eine immer größere Bedeutung zu, da sie für diese Anwendung streng typisierten Programmiersprachen wie C++ oft überlegen sind. Mit dieser Aufgabe haben Sie die Gelegenheit, Datenanalyse mit Python auszuprobieren. Wiederholen Sie die Aufgabe 27 mit Hilfe eines Python-Skripts. Welche Vorteile und welche Nachteile sehen Sie im Vergleich zu ROOT? Einige nützliche Funktionen werden in der Beispieldatei `ex11.py` angegeben.

Hinweis:

Mit dem Rechnernamen `fphctssh.physik.uni-karlsruhe.de` können Sie von überall aus mittels `ssh/scp` Programm auf einen Poolrechner zugreifen.