

Implementierung der CL_s -Methode in ROOT zur statistischen Datenanalyse in der Suche nach neuer Physik am LHC

Bachelorarbeit in Physik

angefertigt am
Physikalischen Institut der Universität Bonn

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

August 2011

von Christopher Helmes

1. Gutachter: Prof. Dr. Klaus Desch
2. Gutachter: Dr. Philip Bechtle

Zusammenfassung

Die vorliegende Bachelorarbeit behandelt die CL_s -Methode zur Bestimmung von Konfidenzniveaus. Die Implementation der Methode im `ROOT`-Framework via `Roofit` und `Roostats` und deren Untersuchung auf Konsistenz der Ergebnisse bildet den Schwerpunkt der Arbeit. Als Anwendung des CL_s Formalismus werden hierfür abschließend die in [ATLAS11] veröffentlichten Daten untersucht.

Inhaltsverzeichnis

1. Einleitung	7
2. Bayesische und frequentistische Statistik	7
3. Likelihoodfunktionen	8
4. Hypothesentests	8
5. CL_s -Methode	11
6. Look-Elsewhere-Effekt	14
7. Implementierung in ROOT	15
8. Ergebnisse	21
9. Diskussion und Vergleich mit Ergebnissen der ATLAS-Kollaboration	22
10. Fazit	25
A. Ergebnisse des CL_s -Tests	27
Literatur	29
Abbildungsverzeichnis	29

1. Einleitung

In der Hochenergiephysik (HEP) sind experimentell gewonnene Daten im Allgemeinen statistisch verteilt. Meist liegen die Daten als Histogramme vor, deren einzelne Bins zählen, wie oft ein poissonverteiltes Ereignis in einer bestimmten Zeitspanne aufgetreten ist. Die Auswertung funktioniert ausschließlich über statistische Methoden. Aufgrund der großen Menge an Daten aus jedem einzelnen Experiment ist die computergestützte Auswertung unverzichtbar, zumal die angewandten Methoden außerdem sehr rechenintensiv sind. Zumeist erfolgt die Auswertung im Rahmen des vom CERN entwickelten C++-Frameworks `ROOT`, welches als Hilfsmittel eine Vielzahl physikalisch relevanter Fragestellungen bearbeiten kann. Die hier verwendeten Pakete `Roofit` und das darauf aufbauende `Roostats` stellen Klassen zur Verfügung, mit deren Hilfe sich statistische Modelle und Auswertungsmethoden sowie die vorhergehende Datenverarbeitung implementieren lassen. Die Abschnitte 2 bis 6 behandeln die theoretischen Grundlagen der angewendeten statistischen Methoden. In Abschnitt 7 wird die Implementierung des CL_s -Formalismus und des Look-Elsewhere-Effekts beschrieben und anschließend das Massenspektrum aus Datensätzen [ATLAS11] bezüglich dieser Größen in Abschnitt 7.2 untersucht. Die verwendeten Daten wurden auf der „Physics at the LHC“-Konferenz 2011 präsentiert und in [ATLAS11] ausgewertet. Die Diskussion der Ergebnisse findet sich in Abschnitt 9.

2. Bayesische und frequentistische Statistik

Aufbauend auf der mathematischen Definition von Wahrscheinlichkeit ([Bar96] p. 119ff.) existieren mehrere unterschiedliche Interpretationen derselben. Die beiden am häufigsten benutzten sind die Bayesische Statistik und die frequentistische Statistik. Nach [Bar96] beschreibt die frequentistische Statistik die Wahrscheinlichkeit, einen bestimmten Wert einer Observablen im Hinblick auf das Limit für viele Experimentwiederholungen zu messen. Wiederholt man ein Experiment N -mal und tritt das Ergebnis A M -mal auf, so heißt $\lim_{N \rightarrow \infty} \frac{M}{N} =: P(A)$ die Wahrscheinlichkeit von A . Die so erhaltene Wahrscheinlichkeit ist dann nicht nur allein von der Observablen sondern auch vom durchgeführten Experiment abhängig.

Die Bayesische Statistik fußt auf dem Theorem von Bayes ([Cow] p. 3) für bedingte Wahrscheinlichkeiten. Hierbei ist die Wahrscheinlichkeit, dass ein Ereignis eintritt, an Bedingungen geknüpft, was a priori Annahmen in das Experiment einbringt. Nach [Cow] beträgt die Wahrscheinlichkeit für das Eintreffen eines Ereignisses a , unter einer Bedingung b :

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)} \quad (1)$$

Die Wahrscheinlichkeiten $P(a)$ und $P(b)$ sind Integrale von Wahrscheinlichkeitsvertei-

lungen für a und b , die a priori angenommen werden. $P(a|b)$ bezeichnet eine a posteriori gewonnene Wahrscheinlichkeit. Interpretiert wird der Ausdruck 1 als Grad der Glaubwürdigkeit der Annahme, dass a unter gegebenem b zutrifft. Ist beispielsweise ein mögliches Resultat a durch ein vorgegebenes b verboten, so ist $P(a|b) = 0$ und der gesamte Ausdruck wird 0 ([Bar96] p. 123). Dieser Grad der Glaubwürdigkeit ist nicht mit der intuitiveren frequentistischen Interpretation von Wahrscheinlichkeit gleichzusetzen.

3. Likelihoodfunktionen

Kennt man die Wahrscheinlichkeitsverteilung $p(x; \mathbf{a})$ einer Variablen x und erhält aus einer Messung einen Datensatz mit N Messwerten $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$, die von einem Satz von k Parametern $\mathbf{a} = \{a_1, a_2, \dots, a_k\}$ abhängen, so definiert sich für einen bestimmten Satz von Parameterwerten die Likelihoodfunktion folgendermaßen:

$$\mathcal{L}(x_1, x_2, \dots, x_n; \mathbf{a}) = \prod_{i=1}^N p(x_i, \mathbf{a})$$

4. Hypothesentests

In der Teilchenphysik wird über die Entdeckung neuer Resonanzen in Massenspektren durch Hypothesentests entschieden. Hierfür werden zunächst zwei Hypothesen aufgestellt: die Null-Hypothese (\mathcal{H}_0) und eine alternative Hypothese (\mathcal{H}_1). Anschließend werden die beiden Hypothesen zur mathematischen Behandlung in einer zu definierenden Teststatistik quantifiziert. Die zu betrachtende Observable geht mit in die Quantifizierung ein. Weiterhin definiert man Kriterien für die Akzeptanz oder Zurückweisung der Nullhypothese, also Intervalle der Werte der Teststatistik, die zu dem einen oder anderen Ergebnis führen.

Als mögliche Teststatistik kann das logarithmische Verhältnis der Likelihood-Funktion unter der alternativen Hypothese und der Likelihood-Funktion unter der Null-Hypothese verwendet werden:

$$-2 \ln Q = -2 \ln \frac{\mathcal{L}(\mathcal{H}_1)}{\mathcal{L}(\mathcal{H}_0)} \quad (2)$$

Nach Festlegung der Hypothesen wird der beobachtete Wert der Teststatistik $-2 \ln Q_{\text{obs}}$ durch Anpassung der Likelihoodfunktionen an die Daten bestimmt. Da auch die Teststatistiken einer Verteilung unterliegen, kann man aus der Durchführung vieler (simulierter) Experimente und jeweiliger Bestimmung von $-2 \ln Q$ die Verteilung der Teststatistik für den Fall, dass die Null-Hypothese stimmt, erhalten. Gleiches kann für die alternative Hypothese durchgeführt werden.

Anschließend versucht man die Hypothese \mathcal{H}_0 zu widerlegen. Gelingt dies, bedeutet das nicht, dass die alternative Hypothese richtig ist. Bei Hypothesentests existieren vier Möglichkeiten, ein Ergebnis zu erhalten: Eine wahre Hypothese anzuerkennen, eine falsche Hypothese zurückzuweisen, eine falsche Hypothese als richtig anzuerkennen (Fehler erster Art) oder eine richtige Hypothese als falsch zurückzuweisen (Fehler zweiter Art). Um die Verträglichkeit einer Hypothese mit den Daten zu messen teilt man die Wahrscheinlichkeitsverteilung der Teststatistik in die Region der Akzeptanz, große Fläche, und die Region der Zurückweisung, kleine Fläche. Als Signifikanz α definiert man das Integral über die Region der Zurückweisung. Liegt $-2 \ln Q$ in der Akzeptanzregion, wird die zugrunde liegende Hypothese akzeptiert; fällt der Wert in die Zurückweisungsregion, ist die Hypothese verworfen. Da die Wahl der Regionen beliebig ist, hat man auf diese Weise Einfluss auf die Signifikanz eines Hypothesentests. Nach dem Neyman-Pearson-Lemma ist für eine gegebene Signifikanz das Verhältnis der Likelihoodfunktionen die Teststatistik, die die stärkste Zurückweisung erlaubt ([Cow] p. 50f). Die beiden Hypothesen können beispielsweise in der HEP die Hypothese einer „Nur-Untergrund“-Messung (\mathcal{H}_0) und einer „Signal+Untergrund“-Messung (\mathcal{H}_1) sein.

4.1. χ^2 -Test

Die Frage, die in Hypothesentests in der HEP am häufigsten auftaucht, ist die nach der Güte der Anpassung einer Theorie an die Daten. Einer der wichtigsten Tests für diese Güte ist der χ^2 -Test.

Ausgegangen wird von einer Messung mit N bekannten Werten x_i und gemessenen Werten $y_i \pm \sigma_i$, wobei σ_i die Messfehler der y_i darstellen. Eine zu untersuchende Funktion verspricht den wahren Wert der Observablen y_i für gegebenes x_i zu liefern. Liegen die Werte x_i exakt vor, funktioniert der Test folgendermaßen:

$$\chi^2 = \sum_i^N \frac{(y_i - f(x_i))^2}{\sigma_i^2} \quad (3)$$

gibt in jedem Summanden $\chi_i^2 \approx 1$, wenn die y_i ca. $1\sigma_i$ von der Vorhersage $f(x_i)$ entfernt liegen und die σ_i unkorreliert sind. Für eine mit 1σ verträgliche Hypothese folgt $\chi^2 \approx N$. Das Integral über die Wahrscheinlichkeitsverteilung p von χ^2

$$p(\chi^2; N) = \frac{2^{-\frac{N}{2}}}{\Gamma\left(\frac{N}{2}\right)} \chi^{N-2} \exp\left[-\frac{\chi^2}{N}\right] \quad (4)$$

$$P(\chi^2; N) = \int_{\chi^2}^{\infty} p(\chi'^2; N) d\chi'^2 \quad (5)$$

heißt χ^2 -Wahrscheinlichkeit ([Bar96] p. 150) und gibt an, mit welcher Wahrscheinlichkeit der beobachtete χ^2 -Wert für eine vorgegebene Anzahl Datenpunkte N auftritt. Kleine

Werte von P korrespondieren mit großen Werten von χ^2 und liefern dadurch einen Hinweis darauf, dass die Hypothese $f(x_i)$ nicht gut zu den gemessenen Daten passt. Da das Integral vom experimentell erhaltenen Wert χ^2 bis ∞ läuft, ist ersichtlich, dass der χ^2 -Test einseitig ist. Das bedeutet, dass große Werte von P , also kleine Werte von χ^2 , keine genaue Aussage über Akzeptanz oder Ablehnung der Funktion zulassen.

Nach Wilks Theorem [Wil] gilt

$$\lim_{N \rightarrow \infty} \chi^2 = -2 \ln Q, \quad (6)$$

wobei Q das Verhältnis zweier Likelihoodfunktionen für zusammengesetzte Hypothesen ist. Gleichung 6 rechtfertigt die Wahl von $-2 \ln Q$ als Teststatistik für Hypothesentests.

4.2. Konfidenzintervall und Konfidenzniveau

Eine Teststatistik t unterliegt, in gleicher Weise wie eine Observable, einer Wahrscheinlichkeitsverteilung $p(t)$. Ein Wert von t wird aus einem Experiment erhalten, was allerdings nicht immer Schlüsse auf $p(t)$ zulässt. Die Durchführung genügend vieler realer Experimente um eine große Stichprobe von t zu erhalten, ist im Allgemeinen sehr zeitaufwändig. In diesem Fall nimmt man Simulationen dieser Experimente.

Hat man $p(t)$ bestimmt, so ist es ein Leichtes diese von $-\infty$ bis t_+ , einem oberen Wert der Teststatistik, zu integrieren. Es muss $t_{\text{obs}} \leq t_+$ gelten. Dann beschreibt das Ergebnis

$$P(t_{\text{obs}} \leq t_+) = \int_{-\infty}^{t_+} p(t) dt = C_+$$

die Wahrscheinlichkeit, dass $t_{\text{obs}} < t_+$ bei der Durchführung eines Experiments ist. Gleichfalls kann das Integral von einer unteren Grenze t_- bis ∞ berechnet werden, wobei $t_- \leq t_{\text{obs}}$ ist.

$$P(t_{\text{obs}} \geq t_-) = \int_{t_-}^{\infty} p(t) dt = C_-$$

Die Wahrscheinlichkeiten C_+ und C_- heißen oberes und unteres einseitiges Konfidenzniveau. Durch die Wahl von t_+ und t_- werden die Konfidenzniveaus analog zur Signifikanz (vgl. Abschnitt 4) kontrolliert und man erhält eine große Konfidenz C_+ ($\max[C_+] = 1$) für große Werte von t_+ und für C_- genau umgekehrt. Mit diesen Definitionen kann man außerdem ein beidseitiges Konfidenzintervall bestimmen. Es ist das Integral von $p(t)$ in den Grenzen t_- und t_+ :

$$P(t_- \leq t_{\text{obs}} \leq t_+) = \int_{t_-}^{t_+} p(t) dt = C$$

4.3. Wahrscheinlichkeitsverteilungen in Hypothesentests

Eine Möglichkeit die Hypothesen \mathcal{H}_0 und \mathcal{H}_1 zu quantifizieren ist die Definition über Wahrscheinlichkeitsverteilungen. In der HEP wird hierbei eine Wahrscheinlichkeitsverteilung für den Untergrund (\mathcal{H}_0) und eine Wahrscheinlichkeitsverteilung für ein mögliches Signal gewählt. Eine mögliche Normierung der beiden Hypothesen ist dann jeweils die gesamte Anzahl der im Experiment registrierten Ereignisse N_{ges} . Dies resultiert in erweiterten Wahrscheinlichkeitsverteilungen, in denen das Integral der Verteilungen über den beobachteten Bereich N_{ges} ist. Je nachdem wie die Wahrscheinlichkeitsverteilungen im Einzelnen festgelegt werden, existieren ein oder mehrere Parameter, welche für die physikalische Aussage nicht interessant sind, aber für die korrekte Berechnung der Teststatistik mit berücksichtigt werden müssen. Solche Parameter heißen Nuisance-Parameter. Als Beispiel betrachte man hierfür folgende Hypothesen:

- Wahrscheinlichkeitsverteilung für \mathcal{H}_0

$$h_0(x; \kappa) = \exp[-\kappa x] \quad (7)$$

- Wahrscheinlichkeitsverteilung für \mathcal{H}_1

$$h_1(x; \sigma, x_0, \kappa) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-x_0)^2}{2\sigma^2}\right] + \exp[-\kappa x] \quad (8)$$

Wenn in diesem Fall der Hypothesentest für ein festes x_0 durchgeführt wird, müssen die Nuisance-Parameter σ und κ trotzdem berücksichtigt werden. Da diese aber nicht in unmittelbarem Zusammenhang mit x_0 stehen, sind sie für den Hypothesentest nicht von physikalischem Interesse. Um die möglichen Werte der Nuisance-Parameter während der Bestimmung der Verteilung der Teststatistik für die einzelnen simulierten Experimente bestimmen zu können, ist es hilfreich eine A-Priori-Wahrscheinlichkeitsverteilung anzugeben, gemäß der die Nuisance-Parameter fluktuieren. Solche Wahrscheinlichkeitsverteilungen sind beispielsweise aus Kontrollmessungen zu erhalten. Die Einführung von Nuisance-Parametern führt zu einer Verbreiterung der Verteilung der Teststatistik $p(t)$.

5. CL_s -Methode

Eine Form des Hypothesentests stellt die CL_s -Methode dar. Bezeichne $\mathcal{L} = \prod_{i=1}^N f(x_i; a)$ die Likelihoodfunktion einer Variablen x unter dem Parameter a , deren Wahrscheinlichkeitsverteilung durch $f(x; a)$ beschrieben wird. Das Verhältnis

$$Q = \frac{\mathcal{L}(x, s+b)}{\mathcal{L}(x, b)} \quad (9)$$

der Likelihoodfunktionen für die Annahme, dass Signal+Untergrund gemessen wurde und der Likelihoodfunktion für die Annahme, dass nur Untergrund gemessen wurde, ist Ausgangspunkt für alle weiteren Betrachtungen. Aus experimentellen Daten kann ein Wert Q_{obs} für Gleichung 9 von \mathcal{H}_0 und \mathcal{H}_1 bestimmt werden. Die Parameter der gewählten Hypothesen werden hierzu an die Daten angepasst. Dieses Verhältnis unterliegt der Wahrscheinlichkeitsverteilung $p(Q)$, so dass die Wahrscheinlichkeit dafür, einen Wert von Q zu finden, der kleiner als Q_{obs} ist, durch

$$P(Q \leq Q_{\text{obs}}) = \int_{-\infty}^{Q_{\text{obs}}} p(Q) dQ \quad (10)$$

gegeben ist. Dies entspricht der Definition eines Konfidenzniveaus für die Hypothese, dass Signal + Untergrund gemessen wurde ($p(Q) = \frac{dP_{s+b}}{dQ}$), bzw. für die Hypothese, dass nur Untergrund gemessen wurde ($p(Q) = \frac{dP_b}{dQ}$). In den beiden Fällen ergibt sich also:

$$CL_{s+b} = P_{s+b}(Q \leq Q_{\text{obs}}) = \int_{-\infty}^{Q_{\text{obs}}} \frac{dP_{s+b}}{dQ} dQ \quad (11)$$

$$CL_b = P_b(Q \leq Q_{\text{obs}}) = \int_{-\infty}^{Q_{\text{obs}}} \frac{dP_b}{dQ} dQ \quad (12)$$

Das hiermit nach [Rea00] hergeleitete Verhältnis

$$CL_s = \frac{CL_{s+b}}{CL_b} \quad (13)$$

gibt demnach eine Näherung an ein Konfidenzniveau für die Signalthypothese an. Es ist zu beachten, dass das CL_s nicht mit den in Abschnitt 4.2 beschriebenen Konfidenzniveaus übereinstimmt. Die Motivation, diese Methode dennoch zu verwenden liegt in der Möglichkeit auch bei im Vergleich zum Untergrund kleinen Signalstärken gültige Aussagen über das Experiment zu treffen. Außerdem liefert die Methode kein Ergebnis in Bereichen, wo das Experiment keine Aussagen treffen kann [Rea00].

Generell sind die Wahrscheinlichkeitsverteilungen der Teststatistik nicht bekannt. Wie in 4.2 angesprochen, helfen simulierte Experimente diesem Umstand ab. Dabei werden die Parameter der Teststatistik zunächst verändert und Zufallsdaten erzeugt. Anhand dieser Daten ist es möglich, weitere Werte der Teststatistik zu erhalten.

Es ist noch einmal zu betonen, dass das resultierende Verhältnis der beiden Wahrscheinlichkeiten CL_{s+b} und CL_b weder eine Wahrscheinlichkeit noch ein Konfidenzniveau für eine der beiden Hypothesen darstellt. Dies wirft die Frage auf, wie dieses Verhältnis zu interpretieren ist. Betrachtet man kleine Werte von CL_s erkennt man, dass $CL_{s+b} \leq CL_b$ und demnach die Konfidenz in \mathcal{H}_0 größer ist als in \mathcal{H}_1 . Kleine Werte von CL_s geben eine größere Sicherheit, Fehler zweiter Art zu vermeiden, also

eine Signalmessung fälschlicherweise zu verwerfen [Rea02]. Je höher die Trennung der Verteilungen der Hypothesen ist, desto mehr nähert sich CL_s an CL_{s+b} an, wie die Abbildungen 1 bis 3 anhand willkürlich gewählter Wahrscheinlichkeitsverteilungen mit unterschiedlichen Signalstärken demonstrieren. Die Signalstärke ist das Verhältnis der mittleren Signalereignisse zu den mittleren Untergrundeignissen.

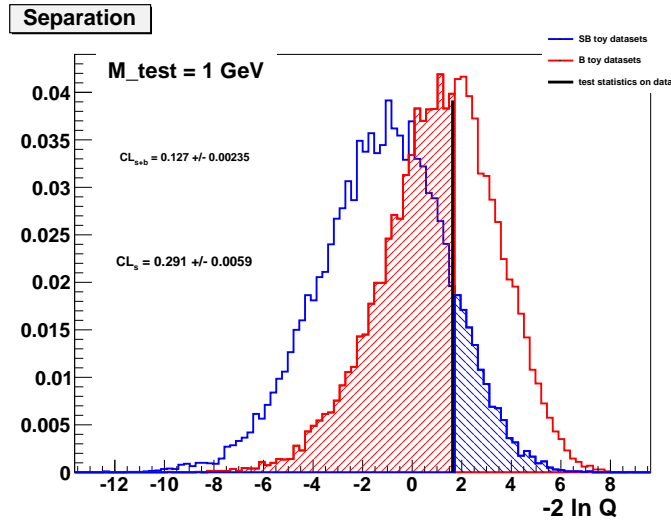


Abbildung 1: Verteilungen der Teststatistik $-2 \ln Q$ für $m_H = 1 \text{ GeV}$, Signalstärke: 0.125

Eine gute Separation ist nur für sich deutlich in den Daten abzeichnende Signale erreichbar. Im hier behandelten Fall kleiner Signale liegen demnach die Verteilungen $p(t)$ (Abschnitt 4.2) für \mathcal{H}_0 und \mathcal{H}_1 fast übereinander.

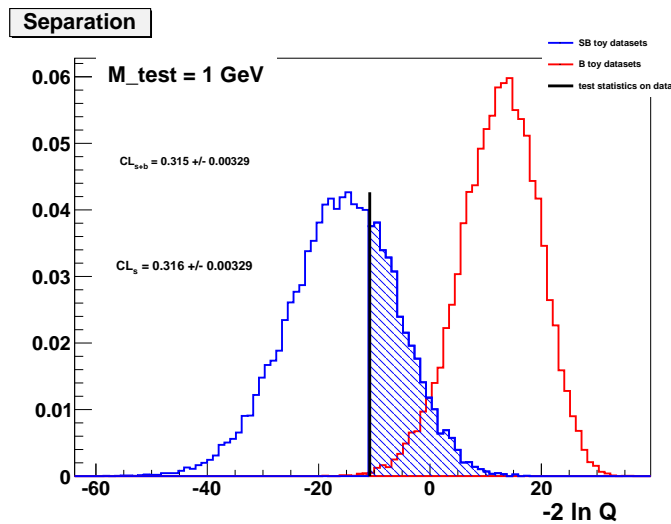


Abbildung 2: Verteilungen der Teststatistik $-2 \ln Q$ für $m_H = 1 \text{ GeV}$, Signalstärke: 0.5

Wenn man frequentistische Konfidenzniveaus CL_{s+b} betrachtet schlagen sich höhere experimentelle Unsicherheiten in einer Verbreiterung der Verteilung der Teststatistik für \mathcal{H}_1 nieder. Bei festem Wert für die Teststatistik kann sich so der Wert für CL_{s+b} erhöhen,

was einer früheren Akzeptanz der Hypothese \mathcal{H}_1 gleichkommt, obwohl die Verbreiterung systematischen Ursachen entspringt. Die CL_s -Methode führt unter diesen Bedingungen zu einer Verringerung der Sensitivität, da sie die Konfidenz in \mathcal{H}_0 mit berücksichtigt, ein weiterer Vorteil gegenüber der sonst gebräuchlichen CL_{s+b} -Methode.

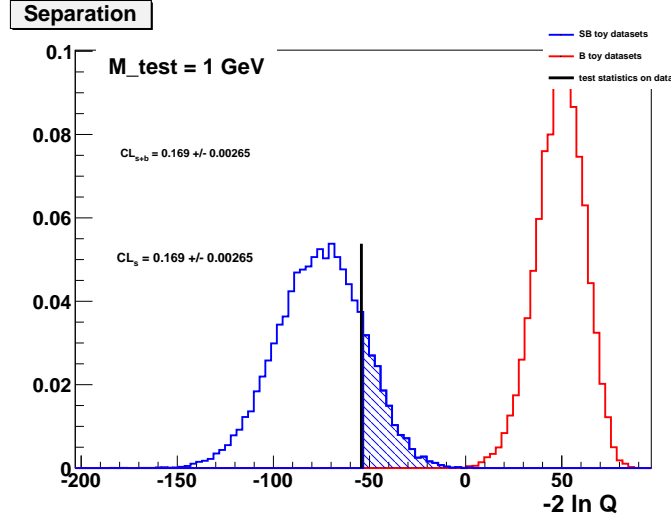


Abbildung 3: Verteilung der Teststatistik $-2 \ln Q$ für $m_H = 1 \text{ GeV}$, Signalstärke: 1.25

Bis hierhin waren alle Aussagen über die CL_s -Methode frequentistischer Natur. Wie in Abschnitt 4 bereits dargelegt, gehen in die Bestimmungen der Verteilungen $p(t)$ A-Priori-Wahrscheinlichkeitsverteilungen ein, die eine Verbreiterung von $p(t)$ verursachen. Da diese Wahrscheinlichkeitsverteilungen vor Bestimmung von CL_s , zum Beispiel aus Messungen in der Nähe der zu untersuchenden Signalregion, festgelegt werden, stellen sie eine bayesische Annahme im Sinne von Abschnitt 2 dar. Aufgrund der frequentistischen und bayesischen Züge der CL_s -Methode wird diese oft als Hybrid zwischen den beiden statistischen Interpretationen angesehen.

6. Look-Elsewhere-Effekt

Betrachtet man in Hypothesentests nur kleine Signale im Vergleich zum vorhandenen Untergrund, kann es passieren, dass man ein Signal dort sieht, wo es eigentlich nur eine Untergrundfluktuation gegeben hat. In diesem Fall erkennt man die alternative Hypothese an, obwohl das Experiment ein reines Untergrundspektrum liefert, es liegt also ein Fehler erster Art vor. Dieser Effekt heißt Look-Elsewhere-Effekt. Die Quantifizierung dieses Effekts kann folgendermaßen geschehen: Zunächst wird gemäß der angenommenen Untergrundverteilung ein Pseudodatensatz i für ein mögliches Experiment generiert. Anschließend wird das Konfidenzniveau für den Untergrund CL_b berechnet und dieser Schritt für möglichst viele Werte des zu testenden Parameters mit einem Pseudodatensatz wiederholt. Trägt man nun $1 - CL_b$, die Wahrscheinlichkeit bei einem Experiment keine reine Untergrundmessung zu haben, gegen den zugehörigen Wert des Parameters auf, so

findet man gewisse Parameterwerte, die ein vorgegebenes Konfidenzlevel (beispielsweise 2σ der Normalverteilung) unterschreiten. Die Anzahl der Unterschreitungen in diesem Durchlauf wird in $n_{\text{low},i}$, genauso wie die Anzahl aller untersuchten Werte $n_{\text{all},i}$ festgehalten. Dieses Verfahren wendet man auf eine große Anzahl von Pseudodatensätzen an. Die Summe über alle D untersuchten Datensätze

$$\sum_{i=1}^D \frac{n_{\text{low},i}}{n_{\text{all},i}} \quad (14)$$

ist die geforderte Quantifizierung des Look-Elsewhere-Effekts, die Wahrscheinlichkeit aus D durchgeführten Experimenten $M = \sum_{i=1}^D n_{\text{low},i}$ Fehler erster Art zu erhalten.

7. Implementierung in ROOT

Die aufeinander aufbauenden Umgebungen `Roofit` und `RooStats` des Frameworks `ROOT` stellen fertige Funktionen bereit, um die oben beschriebenen Aufgaben auszuführen. Die im Zuge des Hypothesentests zu ermittelnde Teststatistik benötigt für die jeweilige Hypothese, „Untergrund“ oder „Untergrund+Signal“, eine anpassbare Wahrscheinlichkeitsverteilung. Zu diesen Wahrscheinlichkeitsverteilungen gehören neben den Daten aus dem Experiment die zu testenden Parameter (Observablen) und die Nuisance Parameter. Für Observablen und Nuisance Parameter stellt `Roofit` den Variablentyp `RooRealVar` zur Verfügung. Innerhalb der Variablendeklaration wird der Wertebereich des Objekts festgelegt. Gleichzeitig kann die Variable mit einem bestimmten Wert initialisiert werden. Hat man Observablen und Parameter deklariert, kann mit ihnen die Wahrscheinlichkeitsverteilung definiert werden. Die hierfür zuständige Klasse `RooAbsPdf` implementiert verschiedene Standardverteilungen wie Normalverteilung, Breit-Wigner-Verteilung oder Landau-Verteilung, bietet zudem aber auch die Möglichkeit eigene Wahrscheinlichkeitsverteilungen zu definieren. Überdies können Wahrscheinlichkeitsverteilungen unter anderem gefaltet (`RooProdPdf`) und addiert (`RooAddPdf`), sowie mit beliebiger Normierung versehen werden (`RooExtendPdf`). Hierfür hilfreich sind die Variablentypen `RooArgList`, eine geordnete Liste von `Roofit`-Objekten und `RooArgSet` eine Menge von `Roofit`-Objekten. Per `RooDataSet` und `RooDataHist` werden Werte aus `ASCII`-Dateien oder `ROOT`-Histogrammen in den Workspace importiert. Dabei wird außerdem ein Bezug zwischen der Histogrammvariable und der Observablen aus dem Workspace hergestellt.

`RooStats` liefert mit der Klasse `HybridCalculatorOriginal` den nötigen Rahmen, um den Hypothesentest nach der CL_s -Methode auszuführen. Der Hypothesentest wird durch den Constructor `HybridCalculatorOriginal` deklariert und mit den experimentellen Daten, der Nullhypothese und der alternativen Hypothese sowie gegebenenfalls Nuisance-Parametern und deren A-Priori-Wahrscheinlichkeitsverteilungen initialisiert. Basierend auf den angegebenen Hypothesen wird nachfolgend eine Teststatistik festgelegt. Unterstützt werden neben der oben beschriebenen, $-2 \ln Q$, auch die Teststatistiken „Anzahl

der Ereignisse“ und „Profile Likelihood“, welche mit `SetTestStatistic` gewählt werden. Die Funktion `SetNumberOfToys` bestimmt die Anzahl der durchgeführten Monte-Carlo-Experimente zur Bestimmung der Verteilung der Teststatistik für die beiden Hypothesen. `UseNuisance` schaltet die Berücksichtigung von Nuisance Parametern an oder aus, mit `UseNuisance(true)` werden die Nuisance-Parameter für jedes Monte-Carlo-Experiment mitvariiert. Durch `SetGenerateBinned` wird festgelegt, ob die Pseudodaten aus den Monte-Carlo-Experimenten kontinuierlich oder gebinned generiert werden, was einen Unterschied in der Geschwindigkeit der Generierung hervorruft.

Sind alle gewünschten Einstellungen an das Objekt `HybridCalculatorOriginal` übergeben, erfolgt der komplette Hypothesentest dadurch, dass `HybridCalculatorOriginal` an die Funktion `GetHypoTest` übergeben wird. Die Ergebnisse speichert eine Variable des Typs `HybridResult`. Mit Hilfe verschiedener Funktionen können die Ergebnisse des Hypothesentests ausgelesen werden. `HybridCalculatorOriginal` bietet zudem eine eigene Klasse, um die erhaltenen Ergebnisse darzustellen: `HybridPlot`. Die Initialisierung und die Funktion `Draw` geben die Verteilung der jeweiligen Hypothese sowie den Wert der realen Teststatistik in einem Histogramm aus. Per `DumpToFile` wird dieser Plot in einer `.root`-Datei gesichert. Die Umsetzung der CL_s -Methode und der Quantisierung des Look-Elsewhere-Effekts wird im nächsten Abschnitt beschrieben.

7.1. CL_s -Methode

Mit den oben aufgeführten Funktionen kann nun die CL_s -Methode numerisch implementiert werden. Zu diesem Zweck ist der endgültige Code abschnittsweise aufgeführt und unter dem zugehörigen Abschnitt erläutert. Die untersuchten Daten aus [ATLAS11] sind in Abb. 4 zu finden.

```
double higgs_hyp;
Double_t mgg_min = 100;
Double_t mgg_max = 150;
char cl[200];
char m_test[30];

int ntoys = 50000, i; //ToyExperiments used for every CL_s run
RooRealVar mgg("mgg", "observed Energy", mgg_min, mgg_max);
higgs_hyp = atoi(argv[1]);

//construct hypotheses, abstract to change parameters

//background hypothesis
RooRealVar a0("a0", "coefficient 0", -0.03, -0.5, 0.);
RooExponential exp_bg("exp_bg", "background distribution", mgg, a0);
RooRealVar bg_yield("bg", "expected background", 700, 0, 1000);
RooExtendPdf bg_ext_pdf("bg_ext_pdf", "extended background pdf", exp_bg, ←
    bg_yield, "mgg");

//signal hypothesis
```



```

RooRealVar sigma_s("sigma_s", "width of seen signal", 2, 0, 5);
RooRealVar mean_s("mean_s", "mean_of_signal(mass)", higgs_hyp, 100, 150);
RooGaussian gauss_s("gauss_s", "gaussianlike signal", mgg, mean_s, sigma_s);
RooRealVar sig_yield("sig_yield", "expected signal", 7, 0, 50);

```

```
//signal+background hypothesis
```

```

RooAddPdf model("model", "model for signal+background", ←
  RooArgList(exp_bg, gauss_s), RooArgList(bg_yield, sig_yield));

```

Zunächst werden die für die Teststatistik notwendigen Wahrscheinlichkeitsverteilungen definiert. Hierbei wird auch die Normierung durch erweiterte Wahrscheinlichkeitsverteilungen festgelegt. Für den Untergrund wird ein exponentieller Abfall gewählt, der auf die Gesamtereignisse im untersuchten Histogramm normiert ist. Für die Verteilung von Signal+Untergrund wird zu dem Untergrund ein normalverteiltes Signal addiert. Die Erwartungen für die Ereignisse eines Signals und die Ereignisse aus dem Untergrund geben die Skalierungen der Verteilung an.

```
//prior pdfs of nuisance parameters
```

```

RooGaussian bg_yield_prior("bg_yield_prior", "", bg_yield, ←
  RooConst(900.), RooConst(175.));
RooGaussian a0_prior("a0_prior", "", a0, RooConst(-0.03), RooConst(0.005));
RooGaussian ←
  sigma_s_prior("sigma_s_prior", "", sigma_s, RooConst(3.), RooConst(1));
RooGaussian sig_yield_prior("sig_yield_prior", "", sig_yield, ←
  RooConst(40.), RooConst(2.));
RooProdPdf prior_pdf("prior_pdf", "all the nuisance parameters", ←
  RooArgSet(bg_yield_prior, a0_prior, sigma_s_prior, sig_yield_prior));

```

```
//setting nuisance parameters
```

```
RooArgSet nuisance_parameters(bg_yield, a0, sigma_s, sig_yield);
```

Der Erwartungswert für die Anzahl der Ereignisse im Untergrund ist gleichzeitig als Nuisance-Parameter berücksichtigt und die A-Priori-Wahrscheinlichkeitsverteilung dafür eine Normalverteilung in der Gesamtzahl der erwarteten Ereignisse im Untergrund. Weiterhin gehen die Signalfbreite `sigma_s` und die erwartete Signalereignisse `sig_yield` sowie `a_0` aus der Untergrundverteilung ebenfalls als normalverteilte Nuisance-Parameter ein.

```
//getting data of experiment
```

```

mgg.setBins(25);
TFile* file = new TFile("mass_plots_isol5_CaloPointing.root");
TTree* tree = (TTree*) gDirectory->Get("mass_tree");
RooDataSet measure_2("measure_2", "Higgs Mass Search", tree, mgg);

```

```

RooDataHist* impData_2 = measure_2.binnedClone();

```

Der Import der Daten geschieht aus dem Tree des Rootfiles über ein `RooDataSet` in ein `RooDataHist`, sodass ein `RooPlot` des Histogramms `impData_2` identisches ergibt wie in Abb. 4. Um dies zu erreichen sind 25 Bins nötig. Ein Blick in die Statistik des Histogramms enthüllt 926 Einträge, so dass der Initialisierungswert von 700 Ereignissen für die Normierung `bg_yield` in einem möglichen Bereich von 0 bis 1000 Ereignissen in Ordnung ist. Da das Signal als klein angenommen wird ist hier die Anzahl der Gesamtereignisse auf 7 mit einem möglichen Bereich von 0 bis 50 Ereignissen gesetzt.

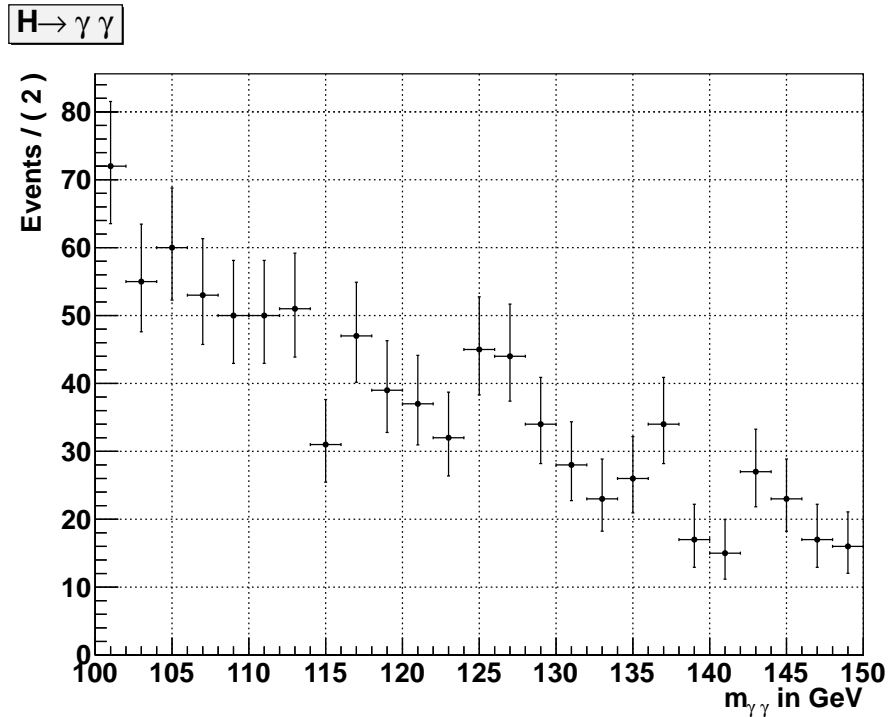


Abbildung 4: Daten nach den Schnitten aus [ATLAS11] für den Übergang eines theoretischen Higgs-Bosons $\mathcal{H} \rightarrow \gamma\gamma$

```

HybridCalculatorOriginal myHybridCalc(*impData_2, model, bg_ext_pdf, ←
    &nuisance_parameters, &prior_pdf);

```

```

//for 2*lnQ

```

```

myHybridCalc.SetTestStatistic(1);
myHybridCalc.SetNumberOfToys(ntoys);
myHybridCalc.UseNuisance(true);
myHybridCalc.SetGenerateBinned(true);

```

```

std::cout << "Model before computation" << endl;
myHybridCalc.PrintMore("v");

```

Im darauf folgenden Abschnitt wird der `HybridCalculatorOriginal` mit den Daten, den Wahrscheinlichkeitsverteilungen, den Nuisance-Parametern sowie deren Verteilungen initialisiert und die benötigten Optionen übergeben. Zur schnelleren Berechnung der Monte-Carlo-Experimente wird die Generierung der Daten für diese Experimente gebinned vorgenommen. Des Weiteren wählen wir $-2 \ln Q$ als Teststatistik sowie die Berücksichtigung der Nuisance-Parameter und der A-Priori-Wahrscheinlichkeitsverteilungen

```
HybridResult* myHybridResult = myHybridCalc.GetHypoTest();

if(!myHybridResult){
    std::cerr << "HybridResult not established" << std::endl;
} //end if

// recover and display the results
double clsb_data = myHybridResult -> CLsplusb();
double clb_data = myHybridResult -> CLb();
double cls_data = myHybridResult -> CLs();
double data_significance = myHybridResult -> Significance();
double min2lnQ_data = myHybridResult -> GetTestStat_data();
double clsb_error = myHybridResult -> CLsplusbError();
double clb_error = myHybridResult -> CLbError();
double cls_error = myHybridResult -> CLsError();
std::cout << "Model after Computation" << endl;
sprintf(cl, "CL_sb = %g, CL_b = %g, CL_s = %g", clsb_data, clb_data, ←
        cls_data);
sprintf(m_test, "M_test = %g GeV", mean_s.getVal());

TCanvas* c = new TCanvas("canvas1", "CL_S-Methode", 1024, 768);
c -> cd();
c -> SetBorderMode(0);
c -> SetFillColor(10);
HybridPlot* myHybridPlot = myHybridResult -> GetPlot(savename, ←
        "Verteilung von -2 ln Q", 100);
myHybridPlot -> Draw();
```

Im vorletzten Abschnitt erfolgt der Aufruf des Hypothesentests mit 50000 Monte-Carlo-Experimenten, sowie die Ausgabe der Ergebnisse, also CL_s , die bestimmten Integrale für CL_b und CL_{s+b} , der Wert der Teststatistik, $-2 \ln Q_{\text{obs}}$ aus den importierten Daten und die aus den Daten bestimmte Signifikanz für die Nullhypothese.

```
sprintf(workspacename, "workspace_50000toys_%d.root", higgs_hyp);

//instantate Workspace

RooWorkspace* w = new RooWorkspace("w", "CLs");

//fill contents into workspace
```

```
w -> import(model);
w -> import(prior_pdf);
w -> import(*impData_2);

//write workspace to file

w -> writeToFile(workspacename);
```

Daran anschließend werden das Modell für die „Signal+Untergrund“-Hypothese, was die Variablen und alle Wahrscheinlichkeitsverteilungen mit einschließt, die A-Priori-Wahrscheinlichkeitsverteilung und die experimentellen Daten in einen `RoWorkspace` geschrieben um sie später noch verwenden zu können. Die Sicherung der Histogramme aus dem Hypothesentest erfolgt in einer `ROOT`-Datei.

7.2. Look-Elsewhere-Effekt

Da der in Abschnitt 7.1 beschriebene Algorithmus ebenfalls die Bestimmung von CL_b ausführt, wird dieser Teil aus dem obigen Makro übernommen. Die Anzahl der Monte-Carlo-Experimente pro Testmasse ist zur schnelleren Bearbeitung auf 500 reduziert. Für die Untersuchung von je 100 Testmassen wird ein Histogramm nach der vorgegebenen Untergrundverteilung generiert, der Beschleunigung halber mit einer Konstanten $a_0 = 0.003$. Die Histogramme für die Monte-Carlo-Experimente werden gebinned generiert, was die Ausführung ebenso beschleunigt wie das Konstanthalten der Nuisance-Parameter. Nach jeder Untersuchung eines Datenpunktes in einem Histogramm wird dieser zusammen mit dem Wert für die Teststatistik ($-2 \ln Q$) in temporäre Arrays (`R` für $1 - CL_b$ und `Q` für $-2 \ln Q$) geschrieben und verglichen ob der Wert $1 - CL_b$ kleiner als ein vorgegebener Wert $c = 0.0028$, zugehörig zu einer einseitigen 2σ -Umgebung für eine Normalverteilung der Teststatistik, ist. Bei einer Unterschreitung von c wird ein Zähler „yes“ um 1 erhöht, bei Nichtunterschreitung ein Zähler „no“ um 1. Zusätzlich werden die Testmassen, für die c unterschritten wird, ebenfalls in ein Array geschrieben (`M`). Die Werte der Arrays `R` und `Q` werden in einem `TGraph`-Objekt $1 - CL_b$ gegen $-2 \ln Q$ gezeichnet. Anschließend wird mit dem nächsten Histogramm fortgefahren. Nach 1000 Histogrammen aus Pseudodaten stellt das Array `M` genügend Testmassen, deren $1 - CL_b$ -Wert unterhalb von c liegt, bereit, um ein Histogramm zu füllen. Aus diesem Histogramm sollte die Verteilung der Testmassen, welche verstärkt dem Look-Elsewhere-Effekt unterliegen, ersichtlich werden. Das Verhältnis der Zähler

$$TF = \frac{\text{yes}}{\text{yes} + \text{no}} \quad (15)$$

ist dann die Wahrscheinlichkeit für das Auftreten des Look-Elsewhere-Effekts in dem untersuchten Massenbereich.

8. Ergebnisse

Die CL_s -Methode wurde für den Bereich von 100 GeV bis 150 GeV in Schritten von 1 GeV durchgeführt. Mit der in Abschnitt 7.1 dargestellten Konfiguration und den akkumulierten Daten aus [ATLAS11] ergab sich für $1 - CL_b$ in Abhängigkeit von der hypothetischen Higgsmasse m_H Abb. 5 und für CL_s gegen m_H Abb. 6:

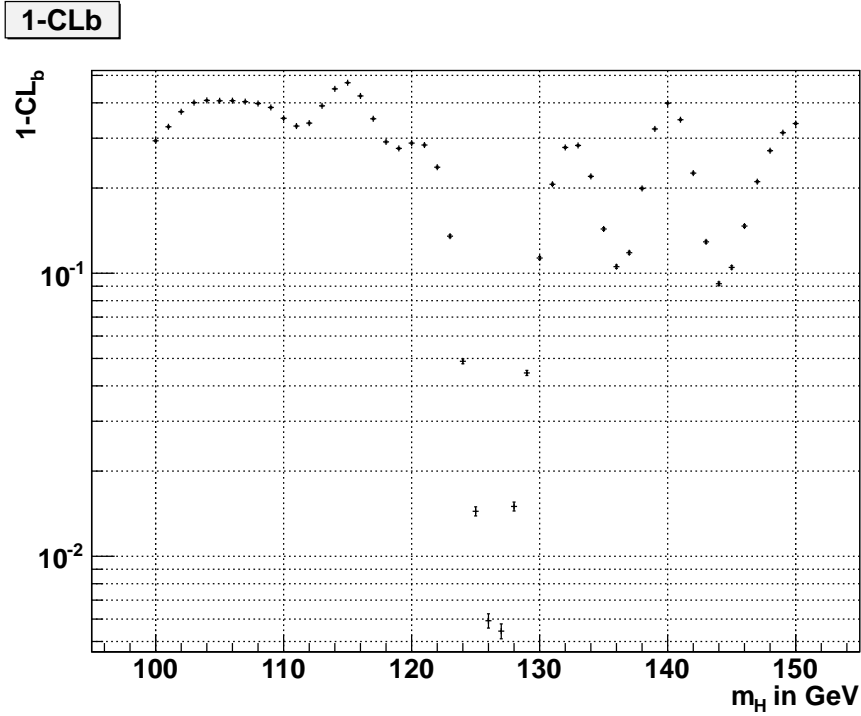


Abbildung 5:

$1 - CL_b$ aufgetragen gegen die hypothetische Higgsmasse m_H für die Daten aus Abb. 4

Für exemplarisch ausgesuchte, hypothetische Higgsmassen nahmen die Verteilungen der Teststatistik die in Abb. 7 und 8 dargestellten Formen an. Die Werte für diese Analyse sind in Tabelle 2 im Anhang 10 zu finden.

Die angegebenen Fehler berechnen sich unter der Annahme einer binomialen Varianz für CL_{s+b} und CL_b und der Fehler für CL_s gemäß der Fehlerfortpflanzung nach Gauß:

$$\sigma_{CL_{s+b}} = \sqrt{\frac{CL_{s+b}(1 - CL_{s+b})}{n_{\text{toys}}}} \quad (16)$$

$$\sigma_{CL_b} = \sqrt{\frac{CL_b(1 - CL_b)}{n_{\text{toys}}}} \quad (17)$$

$$\sigma_{CL_s} = CL_s \sqrt{\left(\frac{\sigma_{CL_{s+b}}}{CL_{s+b}}\right)^2 + \left(\frac{\sigma_{CL_b}}{CL_b}\right)^2} \quad (18)$$

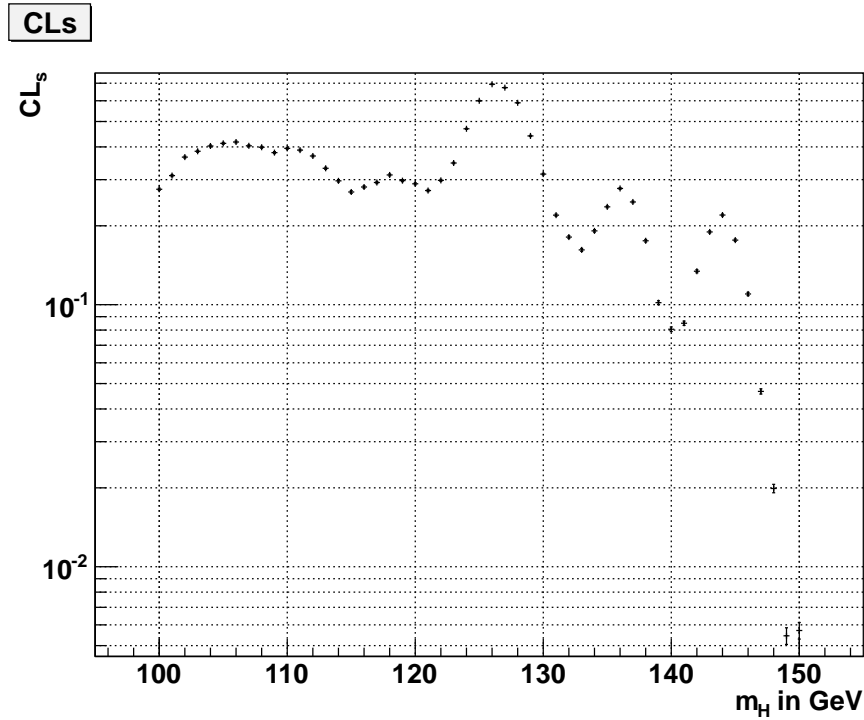


Abbildung 6: CL_s gegen die hypothetische Higgsmasse m_H für die Daten aus Abb. 4

Die Laufzeit des kompilierten Programms für eine Testmasse konnte von durch die Wahl gebinnter Datengenerierung in den Monte-Carlo-Experimenten von ca. 75 min auf ca. 45 min gesenkt werden. Für die Quantifizierung des Look-Elsewhere-Effekts liegen keine Ergebnisse vor. Das geschriebene Makro lief fehlerfrei, war aber nicht in der Lage, mit der in Abschnitt 7.2 beschriebenen Vorgehensweise in absehbarer Zeit Ergebnisse zu liefern. Eine Parallelisierung des Programms hätte auf 100 Rechnern immer noch eine Ausführdauer von etwa 70 Tagen erfordert. Weil die verwendete Version von ROOT nicht mit der Option zum Profiling kompiliert war, konnten die Zugriffszeiten der einzelnen aufgerufenen Funktionen nicht genau verfolgt werden. Die Verwendung von PROOF, einer Klasse zur automatischen Parallelisierung von ROOT-Makros, ist in der verwendeten Version von ROOT (Ver. 5.28.00) für den HybridCalculatorOriginal nicht implementiert.

9. Diskussion und Vergleich mit Ergebnissen der ATLAS-Kollaboration

Der in Abb. 5 deutlich zu erkennende Abfall für $1 - CL_b$ lässt zunächst auf eine kleine Wahrscheinlichkeit schließen, unter der Annahme der Untergrundhypothese den gleichen Wert der Teststatistik zu erhalten. Für die Masse $m_H = 127 \text{ GeV}$ bedeutet dies eine anscheinend schlechte Kompatibilität mit der Untergrundhypothese. Der hohe Wert für

Verteilung von $-2 \ln Q$

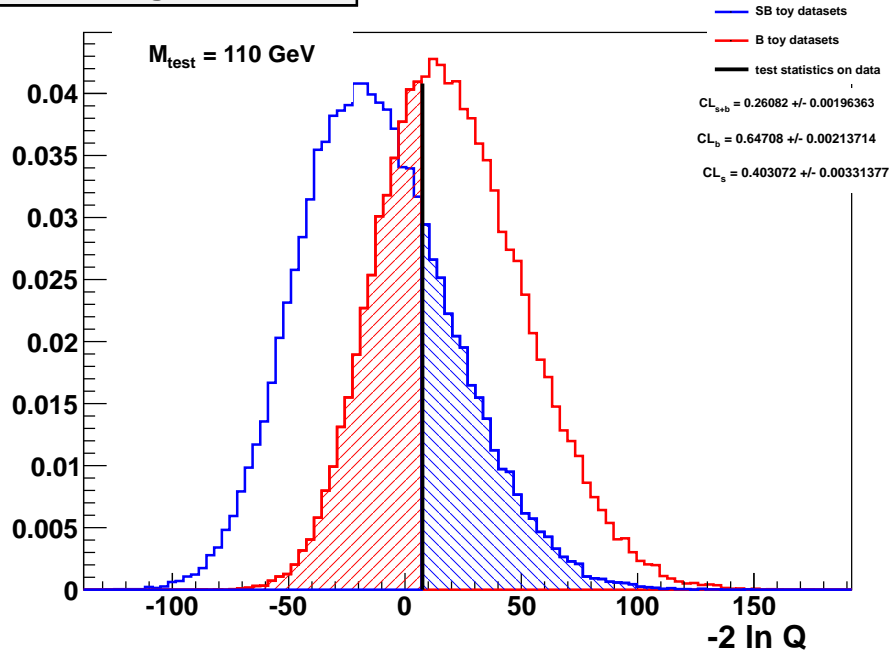


Abbildung 7:
Verteilungen
der Teststatis-
tik $-2 \ln Q$ für
 $m_H = 110 \text{ GeV}$

Verteilung von $-2 \ln Q$

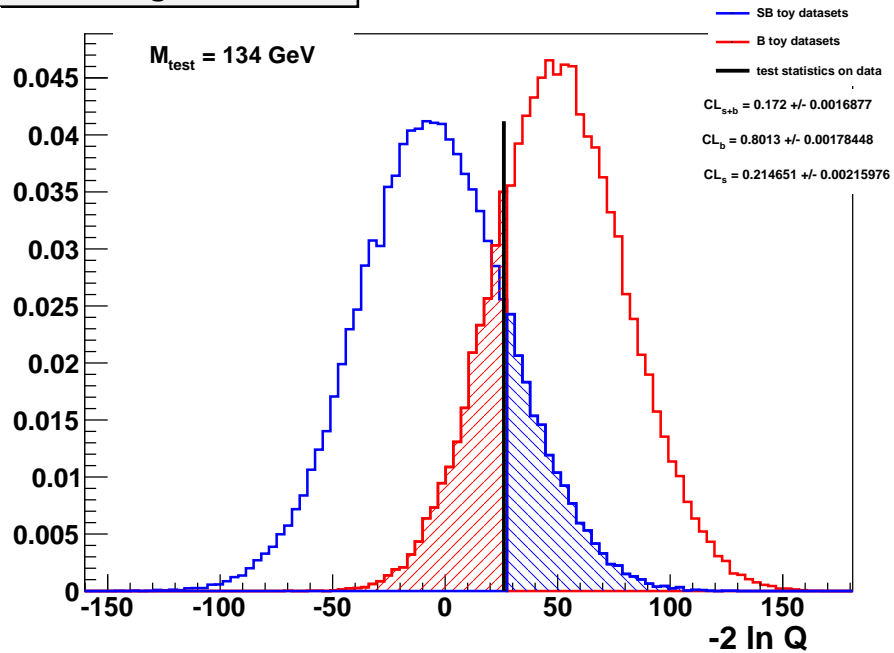


Abbildung 8:
Verteilungen
der Teststatis-
tik $-2 \ln Q$ für
 $m_H = 134 \text{ GeV}$

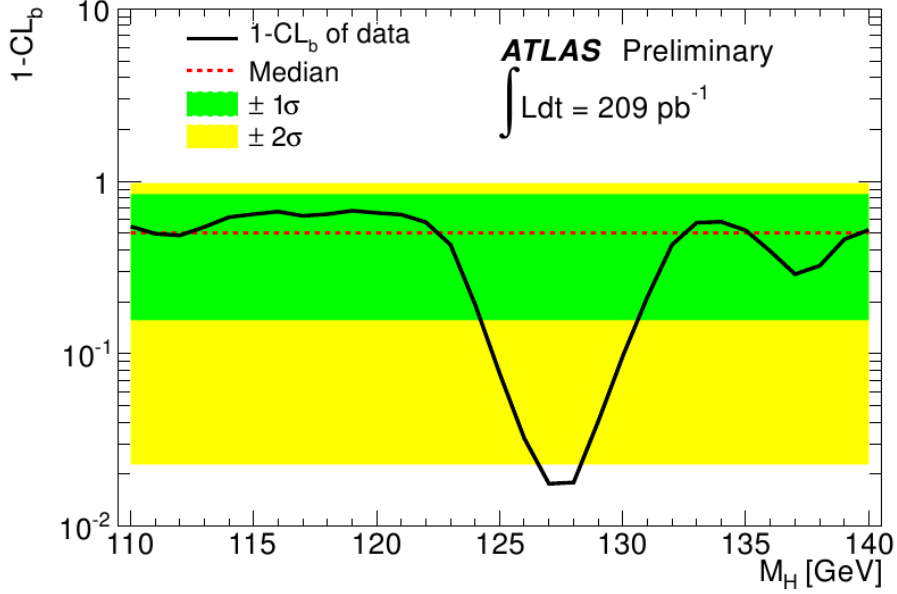


Abbildung 9: $1 - CL_b$ für den Bereich von m_H von 110 GeV bis 140 GeV, Quelle: [ATLAS11]

CL_s bei 127 GeV stützt diese Aussage.

Ein Vergleich mit den in [ATLAS11] veröffentlichten Auswertungen (siehe Abb. 9) liefert eine strukturell ähnliche Beobachtung. Die Werte für CL_s wurden nicht veröffentlicht und der Abfall als Fluktuation im Untergrund behandelt. Laut [ATLAS11] liegt die Wahrscheinlichkeit für eine solche Fluktuation durch den Look-Elsewhere-Effekt in dem untersuchten Massenbereich von 110 GeV bis 140 GeV bei ca. 30 %.

Nach einer Glättung der Werte für $1 - CL_b$ durch Mittelung mit den beiden benachbarten Einträgen in Abb. 5, sind die offensichtlichen Abweichungen zu Abb. 9 in dem betrachteten Bereich geringer (vgl. Abb. 10).

Einen quantitativen Vergleich ermöglichen die in [ATLAS11] aufgeführten Testmassen mit den zugehörigen Werten für $1 - CL_b$. Zusammen mit den aus dem ROOT-Makro erhaltenen Werten für diese Testmassen sind sie in Tab. 1 aufgeführt.

m_H (GeV)	110	115	120	130	140
$1 - CL_b$ (%) [ATLAS11]	55	65	65	10	52
$1 - CL_b$ (%)	35.270	47.110	28.824	11.318	39.874

Tabelle 1: Vergleich der veröffentlichten Werte aus [ATLAS11] und den mit ROOT bestimmten

Die direkte Gegenüberstellung zeigt, dass eine Übereinstimmung höchstens für den kleinsten Wert von $1 - CL_b$ bei $m_H = 130$ GeV angenommen werden kann. Zu der schlechten Übereinstimmung ist zu bemerken, dass die Wahrscheinlichkeitsverteilung für das Signal

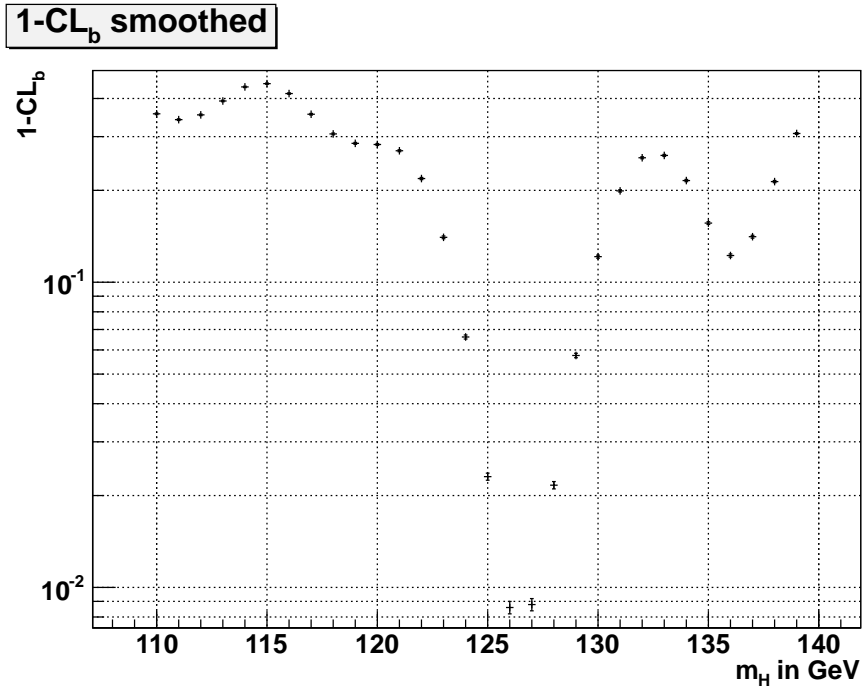


Abbildung 10: $1 - CL_b$ gegen m_H nach Glättung durch Mittelung mit den nächsten Nachbarn aus Abb. 5

aus [ATLAS11] aus einer mit einer Crystall-Ball-Funktion gefalteten Normalverteilung besteht. Das hier gewählte Signalmodell war eine einfache Normalverteilung. Des Weiteren waren Form und Initialisierungswerte der A-Priori-Wahrscheinlichkeitsverteilungen nicht bekannt. Unter diesen, nicht optimalen, Voraussetzungen ist das Wiedererkennen der Struktur aus [ATLAS11] für den Verlauf von $1 - CL_b$ ein guter Hinweis darauf, dass das Makro funktioniert.

10. Fazit

Resümierend lässt sich festhalten, dass die CL_s -Methode gut in ROOT zu implementieren ist. Obwohl die Methode stabil funktioniert, ist die lange Ausführdauer zu bemängeln. Ohne genau zu wissen, worin die Ursache dafür liegt lässt sich dieses Manko leider nicht beheben. Tatsächlich sind während der Zeit, in der diese Arbeit entstand, einige Verbesserungen an der Klasse vorgenommen worden. Insbesondere soll die Geschwindigkeit der HybridCalculator-Methoden sowie die Unterstützung von PROOF verbessert worden sein. Schaut man sich das Zusammenwirken der Klassen RooStats und RooFit an, fällt besonders das einfache Sichern der Resultate auf. Tests können dank des Konzepts des Workspace einfach gesichert und zur weiteren Verwendung genutzt werden.

Die Einarbeitung in das Framework ROOT, war für mich persönlich aufwändig, da ich die Implementierung ohne Kenntnisse objektorientierter Programmiersprachen begann. Oft

verliert man die Übersicht in der nicht gerade ausgefeilten Dokumentation. Online ist in den Referenzen und den Tutorials noch am meisten Hilfe zu finden. Funktionsumfang und Kombinationsmöglichkeiten sind in ROOT einfach zu groß, um den Überblick über diese sich schnell entwickelnde Software zu behalten.

A. Ergebnisse des CL_s -Tests

m_H (GeV)	$1 - CL_b$ (%)	$\Delta(1 - CL_b)$ (%)	CL_s	ΔCL_s	CL_{s+b} (%)	ΔCL_{s+b} (%)
100	29.372	0.204	0.276	0.003	19.470	0.177
101	32.924	0.210	0.311	0.003	20.852	0.182
102	37.206	0.216	0.366	0.003	22.964	0.188
103	40.066	0.219	0.385	0.003	23.068	0.188
104	40.800	0.220	0.403	0.004	23.872	0.191
105	40.682	0.220	0.413	0.004	24.478	0.192
106	40.712	0.220	0.418	0.004	24.756	0.193
107	40.394	0.219	0.404	0.004	24.076	0.191
108	39.776	0.219	0.399	0.003	24.016	0.191
109	38.528	0.218	0.380	0.003	23.384	0.189
110	35.270	0.214	0.395	0.003	25.564	0.195
111	33.114	0.210	0.389	0.003	26.010	0.196
112	33.912	0.212	0.369	0.003	24.408	0.192
113	39.028	0.218	0.331	0.003	20.208	0.180
114	44.814	0.222	0.297	0.003	16.382	0.166
115	47.110	0.223	0.269	0.003	14.236	0.156
116	42.302	0.221	0.281	0.003	16.226	0.165
117	35.138	0.214	0.293	0.003	18.976	0.175
118	29.114	0.203	0.313	0.003	22.162	0.186
119	27.606	0.200	0.297	0.003	21.526	0.184
120	28.824	0.203	0.289	0.003	20.588	0.181
121	28.380	0.202	0.273	0.003	19.522	0.177
122	23.690	0.190	0.298	0.003	22.744	0.187
123	13.520	0.153	0.347	0.002	30.028	0.205
124	4.886	0.096	0.469	0.002	44.614	0.222
125	1.442	0.053	0.600	0.002	59.132	0.220
126	0.592	0.034	0.694	0.002	68.954	0.207
127	0.544	0.033	0.673	0.002	66.960	0.210
128	1.500	0.054	0.589	0.002	58.046	0.221
129	4.444	0.092	0.441	0.002	42.094	0.221
130	11.318	0.142	0.315	0.002	27.934	0.201
131	20.624	0.181	0.220	0.002	17.430	0.170
132	27.824	0.200	0.181	0.002	13.072	0.151
133	28.280	0.201	0.162	0.002	11.606	0.143
134	21.988	0.185	0.191	0.002	14.920	0.159
135	14.334	0.157	0.236	0.002	20.238	0.180
136	10.550	0.137	0.278	0.002	24.836	0.193
137	11.816	0.144	0.247	0.002	21.744	0.184

m_H (GeV)	$1 - CL_b$ (%)	$\Delta(1 - CL_b)$ (%)	CL_s	ΔCL_s	CL_{s+b} (%)	ΔCL_{s+b} (%)
138	19.934	0.179	0.175	0.002	14.038	0.155
139	32.348	0.209	0.102	0.002	6.886	0.113
140	39.874	0.219	0.081	0.002	4.846	0.096
141	34.862	0.213	0.085	0.002	5.534	0.102
142	22.580	0.187	0.134	0.002	10.386	0.136
143	12.910	0.150	0.189	0.002	16.502	0.166
144	9.184	0.129	0.220	0.002	19.962	0.179
145	10.486	0.137	0.176	0.002	15.784	0.163
146	14.680	0.158	0.110	0.002	9.374	0.130
147	21.064	0.182	0.047	0.001	3.684	0.084
148	27.098	0.199	0.020	0.001	1.450	0.053
149	31.374	0.208	0.005	0.000	0.374	0.027
150	33.802	0.212	0.006	0.000	0.378	0.027

Tabelle 2: Ergebnisse der CL_s -Analyse für vorgegebenes m_H im Bereich von 100 GeV bis 150 GeV

Literatur

- [ATLAS11] ATLAS Collaboration, **Search for the Higgs Boson in the Diphoton Channel with the ATLAS Detector Using 209 pb⁻¹ of 7 TeV Data Taken in 2011**, ATLAS-CONF-2011-085
- [Bar96] R. J. Barlow, **Statistics: a Guide to the Use Of Statistical Methods in the Physical Sciences**, John Wiley & Sons, 1996
- [Cow] G. Cowan, **Statistical Data Analysis**, Oxford Science Publications, 2004
- [Rea00] A. L. Read, **Modified Frequentist Analysis of Search Results (the CL_s Method)**, CERN Workshop on Confidence Limits 2000, CERN 2000-005 81-99
- [Rea02] A. L. Read, **Presentation of Search Results: the CL_s Technique**, J. Phys. G: Nucl. Part. Phys. 28 (2002) 2693-2704
- [Wil] S. S. Wilks, **The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses**, Ann. Math. Statist. Volume 9, Number 1 (1938), 60-62.

Abbildungsverzeichnis

1.	$-2 \ln Q$, $m_H = 1$ GeV, Signalstärke: 0.125	13
2.	$-2 \ln Q$, $m_H = 1$ GeV, Signalstärke: 0.5	13
3.	$-2 \ln Q$, $m_H = 1$ GeV, Signalstärke: 1.25	14
4.	Daten: $\mathcal{H} \rightarrow \gamma\gamma$ nach Schnitten	18
5.	$1 - CL_b$ gegen m_H	21
6.	CL_s gegen m_H	22
7.	$-2 \ln Q$ bei $m_H = 110$ GeV	23
8.	$-2 \ln Q$ bei $m_H = 134$ GeV	23
9.	$1 - CL_b$ für 110 GeV bis 140 GeV	24
10.	$1 - CL_b$ gegen m_H geglättet	25

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate kenntlich gemacht habe.

Bonn, den.....

Unterschrift.....