

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Testování statistických hypotéz v psychometrii

Jakub Škrach

Bakalářská práce
2024

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jakub Škrach**
Osobní číslo: **I21242**
Studijní program: **B0688A140009 Informační technologie**
Téma práce: **Testování statistických hypotéz v psychometrii**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

Psychometrie je pomocná disciplína diferenciální psychologie a psychodiagnostiky, která užívá kvantitativní metody a pomáhá vytvářet a vyhodnocovat psychologické testy. Její náplní je hledání a rozvíjení matematických nástrojů vhodných k měření nejrůznějších stránek osobnosti. Cílem práce je nastudovat a naprogramovat statistické testy užívané ke zpracování testových a dotazníkových metod. Nejčastěji se v psychometrii užívají neparametrické metody jako je Mc Nemarův test, Cochranův Q test, Mann-Whitneyův test, Friedmannův test, Kruskal-Wallisův test, různé korelační koeficienty, faktorová analýza. Velká pozornost je věnována reliabilitě testů a validitě testů. V práci budou popsány veškeré aplikované metody a bude vysvětleno uživatelské prostředí aplikace.

Rozsah pracovní zprávy: **45**
Rozsah grafických prací: **5**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Eva Reiterová: Základy psychometrie, Vydavatelství Univerzity Palackého, Olomouc, 2003. ISBN 80-244-0717-5

Kerri A. Goodwin, C. James Goodwin: Research in Psychology: Methods and Design, Wiley; 8th edition, 2016. ISBN 978-1119330448.

Vedoucí bakalářské práce: **Mgr. Jaroslav Marek, Ph.D.**
Katedra matematiky a fyziky

Datum zadání bakalářské práce: **15. prosince 2023**

Termín odevzdání bakalářské práce: **10. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2024

Prohlášení autora

Prohlašuji:

Práci s názvem Testování statistických hypotéz v psychometrii jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15.08.2024

Jakub Škrach

Poděkování

Tímto bych chtěl velmi poděkovat Mgr. Jaroslavu Markovi, Ph.D. za jeho odborné vedení a cenné rady. Dále bych chtěl poděkovat rodině a svým kamarádům, kteří mě podporovali a pomáhali při studiu.

Anotace

V této bakalářské práci je zkoumána problematika statistických testů, výběrových korelačních koeficientů a faktorové analýzy s důrazem na jejich aplikaci v psychometrii. Nejprve jsou popsány cíle psychometrie a její historie, důraz je kladen na výklad základů testování statistických hypotéz, včetně popisu parametrických a neparametrických testů a jejich použití v praxi. V teoretické části jsou krátce popsány konkrétní statistické testy, jako jsou McNemarův test, Cochranův Q-test, Mann-Whitneyův U-test, Friedmanův test a Kruskal-Wallisův test. Dále jsou zde prezentovány různé výběrové korelační koeficienty a z metod mnohorozměrné statistické analýzy je studována faktorová analýza společně s analýzou hlavních komponent (PCA). V navazující části je prezentován návrh a implementace aplikace pro psychometrické testování, která umožňuje praktické využití uvedených statistických metod. Celá práce je zaměřena na propojení teoretických znalostí s jejich praktickou aplikací v oblasti psychometrie.

Klíčová slova

Statistické testy, korelační koeficienty, neparametrické testy, faktorová analýza, psychometrie.

Title

Statistical hypothesis testing in psychometrics.

Annotation

This bachelor thesis examines the issues of statistical tests, sample correlation coefficients and factor analysis with emphasis on their application in psychometrics. First, the goals of psychometrics and its history are described, and emphasis is placed on an explanation of the basics of statistical hypothesis testing, including a description of parametric and nonparametric tests and their use in practice. In the theoretical part, specific statistical tests such as the McNemar test, Cochran Q-test, Mann-Whitney U-test, Friedman test and Kruskal-Wallis test are briefly described. In addition, various sample correlation coefficients are presented, and among the multivariate statistical analysis methods, factor analysis is studied together with principal component analysis (PCA). In a follow-up section, the design and implementation of a psychometric testing application is presented, which enables the practical use of these statistical methods. The whole work is aimed at linking theoretical knowledge with its practical application in the field of psychometrics.

Keywords

Statistical tests, correlation coefficients, non-parametric tests, factor analysis, psychometrics.

Obsah	
Seznam zkratk a matematických symbolů	8
Seznam obrázků.....	10
Seznam tabulek.....	10
1 Úvod.....	11
1.1 Psychometrie	12
1.1.1 Historie a vývoj psychometrie.....	12
1.1.2 Psychometrické testy, klíčové aspekty a jejich význam.....	13
1.2 Testování statistických hypotéz.....	15
1.2.1 Základní pojmy.....	15
1.2.2 Chyby při testování hypotéz.....	17
1.2.3 Postup testování hypotéz	17
1.3 Neparametrické testy	18
1.3.1 Výhody a nevýhody neparametrických testů.....	18
1.3.2 Situace vhodné pro použití neparametrických testů	18
2 Teoretická část.....	20
2.1 McNemarův test	20
2.2 Cochranův Q-test.....	21
2.3 Mann-Whitneyův U-test.....	22
2.4 Friedmanův test	23
2.5 Kruskal-Wallisův test	24
2.6 Korelační koeficienty	25
2.6.1 Spearmanův korelační koeficient	25
2.6.2 Kendallův korelační koeficient.....	26
2.7 Faktorová analýza.....	27
2.7.1 Analýza hlavních komponent (PCA).....	28
3 Návrh a implementace aplikace	32
3.1 Požadavky na aplikaci	32
3.2 Architektura aplikace a použité technologie	33
3.2.1 .NET framework.....	33
3.2.2 Jazyk C#	34
3.2.3 XAML	34
3.2.4 WPF.....	35

3.2.5	OxyPlot.....	35
3.2.6	Visual Studio	36
3.2.7	Architektura MVVM	37
3.2.8	Math.NET Numerics	37
3.3	Uživatelské rozhraní	38
3.3.1	Hlavní okno	38
3.3.2	McNemarův test	40
3.3.3	Mann-Whitneyův test	42
3.3.4	Cochranův test	43
3.3.5	Friedmanův test	44
3.3.6	Kruskal-Wallisův test	45
3.3.7	Korelační koeficienty	46
3.3.8	Faktorová analýza.....	46
3.4	Implementace psychometrických testů.....	47
3.4.1	Výpočet kritické hodnoty pro χ^2 rozdělení	47
3.4.2	Výpočet hodnot pomocí inverzní funkce normálního rozdělení	48
3.4.3	Implementace McNemaraova testu	49
3.4.4	Implementace Mann-Whitneyho U-testu	51
3.4.5	Implementace Cochranova Q-testu	52
3.4.6	Implementace Friedmanova testu	54
3.4.7	Implementace Kurskal-Wallisova testu.....	55
3.4.8	Implementace Spearmanova korelačního koeficientu.....	57
3.4.9	Implementace Kendalova korelačního koeficientu	58
3.4.10	Implementace faktorové analýzy.....	60
	Závěr	62
	Literatura	63
	Příloha A – MIT Licence knihovny OxyPlot	65
	Příloha B – MIT Licence knihovny Math.NET Numerics.....	66

Seznam zkratk a matematických symbolů

př. n. K.	před narozením Krista
př. n. l.	před naším letopočtem
MMPI	Minnesota Multiphasic Personality Inventory
TAT	Tematický apercepční test
ROR	Rorschachův test
α	malé řecké alfa, hladina významnosti, chyba I. druhu
β	malé řecké beta, chyba II. druhu
χ^2	chi-kvadrát
Θ	velké theta, obecné označení skalárního parametru
Θ	velké tučné theta, obecné označení vektorového parametru
τ	malé tau, Kendallův korelační koeficient
Σ	sumační znak, velké řecké sigma
$\text{cov}(X, Y)$	kovariance náhodných veličin X a Y
$\text{var}(X)$	rozptyl veličiny X
ρ	malé rho, korelační koeficient
σ	malé sigma, směrodatná odchylka
CSV	Comma-separated values
GC	garbage collector
JSON	JavaScript Object Notation
SDK	software development kit
IoT	Internet of Things
XAML	Extensible Application Markup Language
UI	User interface
WPF	Windows Presentation Foundation
XML	Extensible Markup Language

λ	malé lambda, vlastní číslo
Λ	velké tučné lambda, matice faktorových zátěží
Π	symbol pro součin prvků, velké pí
API	Application programming interface
PNG	Portable Network Graphics
SVG	Scalable Vector Graphics
PDF	Portable document format
IDE	Integrated development environment
AI	Artificial intelligence
MIT	Massachusetts Institute of Technology
MVVM	Model–view–viewmodel
CDF	kumulativní distribuční funkce
$\Gamma(x)$	gama funkce
SVD	Singular value decomposition
PCA	Principal component analysis

Seznam obrázků

Obrázek 1 – Návrhový vzor MVVM, zdroj: (16).....	37
Obrázek 2 – Hlavní okno aplikace	39
Obrázek 3 – Okno s licenčními podmínkami	40
Obrázek 4 – Okno s grafy McNemarova testu	41
Obrázek 5 – Okno s grafy Mann-Whitneyho testu.....	42
Obrázek 6 – Okno s grafy Cochranova testu	43
Obrázek 7 – Okno s grafy Friedmanova testu	44
Obrázek 8 – Okno s grafy Kruskal-Wallisova testu	45
Obrázek 9 – Okno s grafy faktorové analýzy	47

Seznam tabulek

Tabulka 1 – Vztah mezi hypotézou a rozhodnutím, zdroj: (Blatná, strana 81).....	17
Tabulka 2 – Kontingenční tabulka pro McNemarův test, zdroj: (ANDĚL, strana 217)	20
Tabulka 3 – Dichotomická tabulka pro Cochranův Q-test, zdroj: (ANDĚL, strana 217)...	21

1 Úvod

Psychometrie, pomocný vědní obor psychologie, je disciplínou, která se zabývá kvantitativním měřením psychologických vlastností jedinců. Od svého vzniku prošla psychometrie dlouhým vývojem, během kterého se z pouhé teorie měření stala důležitou metodologií využívanou v různých oblastech, jako je klinická psychologie, výzkum či personalistika. Psychometrické testy, které jsou hlavním výstupem tohoto oboru, umožňují hodnocení širokého spektra vlastností, včetně inteligence, osobnostních rysů, schopností, postojů a dalších mentálních a behaviorálních charakteristik. Díky využití matematických a statistických metod tyto testy zajišťují spolehlivé a validní výsledky, které jsou klíčové pro objektivní hodnocení jedinců.

Tato bakalářská práce se zaměřuje na zkoumání vybraných statistických metod a jejich aplikací v psychometrii, přičemž hlavním cílem je propojit teoretické základy s praktickou implementací. V teoretické části jsou popsány základní koncepty testování statistických hypotéz, včetně rozdílů mezi parametrickými a neparametrickými testy, které mají v psychometrii své specifické využití. Práce se dále zaměřuje na konkrétní neparametrické statistické testy, jako jsou McNemarův test, Cochranův Q-test, Mann-Whitneyův U-test, Friedmanův test a Kruskal-Wallisův test, které jsou často využívány pro analýzu dat v psychologických výzkumech, kde nejsou splněny předpoklady pro použití parametrických metod.

Další část práce se věnuje výběrovým korelačním koeficientům, konkrétně Spearmanovu a Kendallovu koeficientu, které umožňují měřit sílu a směr vztahu mezi dvěma proměnnými na základě jejich pořadí. Tyto metody jsou důležité zejména v případech, kdy data nevyhovují předpokladům pro použití Pearsonova korelačního koeficientu. Práce se také zabývá faktorovou analýzou a analýzou hlavních komponent (PCA), které představují mocné nástroje pro redukci dimenzionality a identifikaci základních faktorů ovlivňujících psychometrická data.

Následně se práce zaměřuje na praktickou implementaci aplikace pro psychometrické testování, která integruje výše zmíněné statistické metody. Aplikace je vyvinuta s cílem usnadnit psychologům a výzkumníkům provádění komplexních analýz a interpretaci výsledků. Součástí této aplikace jsou mimo jiné moduly pro výpočet kritických hodnot, implementaci jednotlivých statistických testů a korelačních koeficientů, a také nástroje pro faktorovou analýzu. Výsledná aplikace je navržena tak, aby byla uživatelsky přívětivá a zároveň poskytovala přesné a spolehlivé výsledky, které jsou nezbytné pro odborné posouzení psychologických charakteristik.

Cílem této bakalářské práce je nejen představit teoretické základy a aplikace vybraných statistických metod v psychometrii, ale také ukázat, jak lze tyto metody efektivně využít v praxi prostřednictvím moderní softwarové aplikace. Výstupem této práce je proto nejen teoretický přínos v oblasti psychometrie, ale i praktický nástroj, který může být využit v reálných psychologických výzkumech a diagnostických postupech.

1.1 Psychometrie

Psychometrie je pomocný vědní obor psychologie. Psychometrie se věnuje měření psychologických vlastností jedince. Zkoumá zejména osobnostní charakteristiky, jako jsou inteligence, znaky osobnosti, schopnosti, postoje a další mentální a behaviorální charakteristiky. Cílem psychometrie je vyvinout, validovat a aplikovat nástroje a techniky, které umožňují kvantitativní měření a analýzu těchto vlastností. Takto psychometrie přispívá k objektivnímu a přesnému hodnocení jednotlivců, posuzuje normalitu a proměnlivost charakteristik a zkoumá závislosti mezi charakteristikami.

Psychometrické testy jsou hlavním nástrojem této disciplíny a používají se k hodnocení různých aspektů lidského chování a psychiky. Mezi nejčastější typy psychometrických testů patří testy inteligence, osobnostní testy, testy schopností a dovedností, dotazníky zaměřené na měření postojů a hodnot, a mnoho dalších.

Psychometrie se opírá o matematické a statistické metody. To zajišťuje, že vyvinuté testy jsou spolehlivé (tj. že poskytují konzistentní výsledky při opakovaném měření) a validní (tj. že skutečně měří to, co mají měřit). Další důležitou součástí psychometrie je standardizace testů, což znamená, že testy jsou prováděny stejným způsobem ve všech případech, aby byly výsledky srovnatelné.

Psychometrie má široké uplatnění v různých oblastech, včetně klinické a školní psychologie, personalistiky, vzdělávání a výzkumu trhu, kde se psychometrické testy používají například při diagnostice duševních poruch, výběru zaměstnanců, hodnocení výukových metod nebo analýze spotřebitelských preferencí.

1.1.1 Historie a vývoj psychometrie

Psychometrie, stejně jako psychodiagnostika, má dlouhou historii, ačkoliv její kořeny sahají až do starověku. První zmínky o individuálních rozdílech v psychických vlastnostech lze najít již ve starověkých literárních památkách, jako je Epos o Gilgamešovi a Bible. Praktické využití psychodiagnostiky začalo kolem roku 2200 př. n. K. v Číně, kde se používaly různé zkoušky pro výběr vládních úředníků. (URBÁNEK, strana 5)

V antické filozofii se Platón a Aristoteles zabývali individuálními rozdíly. Řecký lékař Řecký lékař Hippokrates v 5. století př. n. K. je známý svou teorií o čtyřech tělních tekutinách (krve, černé žluči, žluté žluči a hlenu), které podle něj určovaly lidský temperament a zdraví. Tuto teorii později rozvinul Galénos, který na základě této teorie popsal čtyři základní typy temperamentu: sangvinický, cholерický, melancholický a flegmatický. Aristotelův následovník Theofrastos vytvořil popisy třiceti typů osobností, ačkoliv tyto popisy nebyly empiricky podloženy. (URBÁNEK, strana 5)

Ve středověku byly psychické poruchy často spojovány s posedlostí d'áblem a nemocí vyháněním. Francouzská revoluce a práce Philippa Pinela přinesla změnu v postojích k duševním nemocem, což vedlo k rozvoji humánnějších metod léčby.

Ve druhé polovině 19. století se rozvinula psychometrie jako vědecká disciplína, což bylo ovlivněno především pracemi vědců jako Wilhelm Wundt, který založil v roce 1879 první psychologickou laboratoř, a Francise Galtona, který se zaměřil na měření individuálních rozdílů. Galton byl průkopníkem v oblasti vědeckého zkoumání mentálních schopností a vývoje prvních psychometrických metod. (URBÁNEK, strana 5-6)

James McKeen Cattell, který byl ovlivněn Galtonem, přispěl k rozvoji experimentální psychologie a vytvořil první testy mentálních schopností. Alfred Binet ve spolupráci Théodorem Simonem vyvinul Binet-Simonovu škálu, která byla základem pro moderní testy inteligence. Tato škála byla později adaptována Henrym Goddardem a revidována Lewisem Madisonem Termanem v USA jako Stanford-Binetova škála. (URBÁNEK, strana 6-8)

Ve 20. století došlo k dalšímu významnému rozvoji psychometrie, přičemž z tohoto období lze vybrat několik klíčových momentů. Prvním osobnostním dotazníkem byl v roce 1919 Woodworthův Personal Data Sheet, který byl v podstatě standardizovaným psychiatrickým interview. V roce 1921 následoval Rorschachův test (ROR), v roce 1935 Tematický apercepční test (TAT) a v roce 1943 Minnesota Multiphasic Personality Inventory (MMPI), který se stal základním nástrojem pro diagnostiku psychopatologií. (URBÁNEK, strana 8-9)

Od 20. let 20. století byly podnikány pokusy měřit nejen osobnostní rysy, ale i zájmy, postoje a hodnoty. Například Kurt Lewin se zabýval experimentálním výzkumem konfliktu, frustrace a aspirační úrovně. V tomto období se rozvíjely jak jednodimenzionální, tak vícefaktorové nástroje pro diagnostiku normálních i patologických rysů osobnosti. Ve 40. a 50. letech došlo k rozvoji aplikovaných oblastí spojených s konstrukcí, skóringem a interpretací psychometrických nástrojů. V 50. a 60. letech se pak výzkum zaměřil na specifické osobnostní proměnné, které vyžadovaly nové diagnostické metody, jako jsou rysy autoritářství, motivace k výkonu, závislost či nezávislost na poli, represe a senzitivita nebo místo kontroly. (URBÁNEK, strana 8-9)

Významným faktorem vývoje psychometrie byla také dostupnost výpočetní techniky, která umožnila zjednodušit a zpřístupnit psychodiagnostické postupy. Díky tomu mohly být psychometrické modely ověřovány na reálných datech a aplikovány při tvorbě nových diagnostických nástrojů. (URBÁNEK, strana 9)

1.1.2 Psychometrické testy, klíčové aspekty a jejich význam

Psychometrické testy jsou nástroje používané v psychologii a psychiatrii k měření a hodnocení různých psychologických vlastností, jako jsou schopnosti, osobnostní rysy, inteligence, postoje, emoce a další mentální a behaviorální charakteristiky. Tyto testy jsou navrženy tak, aby analyzovaly objektivní, standardizovaná a kvantitativní data. Výsledky testů pak mohou být použity k porovnání jednotlivců nebo skupin.

Psychometrické testy mají klíčové aspekty, na které je důležité se zaměřit, aby byla zajištěna jejich efektivita a věrohodnost:

Objektivita – Test je objektivní tehdy, když jeho výsledky jsou nezávislé na osobě, která test předkládá, a na osobě, která jej vyhodnocuje. To znamená, že instrukce a podmínky pro provedení testu musí být pro všechny testované osoby stejné. Důležité je zajistit, aby byly instrukce jasně pochopeny, a to buď prostřednictvím podrobných manuálů, nebo přesně formulovaných pokynů. Objektivitu také zajišťuje standardizované hodnocení, kde jsou odpovědi zařazovány do předem stanovených kategorií, a výsledky testu jsou hodnoceny bez subjektivního vlivu hodnotitele. Testy, které umožňují jasné a objektivní hodnocení, mají vyšší důvěryhodnost, protože minimalizují riziko zkreslení výsledků způsobeného lidským faktorem. (SVOBODA, strana 17-18)

Standardizace – Psychometrické testy jsou standardizovány, což znamená, že jsou navrženy tak, aby byly prováděny a vyhodnocovány stejným způsobem pro všechny účastníky. Standardizace zahrnuje nejen zajištění konzistentních podmínek testování, ale také normalizaci výsledků na základě velkých reprezentativních vzorků populace. To zajišťuje, že výsledky jsou srovnatelné napříč různými skupinami testovaných osob a mohou být interpretovány vzhledem k průměrným hodnotám a standardním odchylkám populace. (SVOBODA, strana 18-19)

Reliabilita – Reliabilita testu označuje jeho schopnost poskytovat konzistentní výsledky při opakovaném měření. Vysoce reliabilní test by měl produkovat stejné výsledky, pokud je administrace a podmínky testování stejné. Reliabilita testu může být měřena různými způsoby, včetně test-retest reliability (stability výsledků v čase), ekvivalence (korelací paralelních verzí testu) a vnitřní konzistence (homogenity položek). Vysoká reliabilita, obvykle nad 0.8, je klíčová pro to, aby byl test považován za spolehlivý. (SVOBODA, strana 19-20)

Validita – Validita testu se týká toho, zda test skutečně měří to, co má měřit. Například test inteligence by měl přesně hodnotit intelektové schopnosti, nikoli jiné vlastnosti, jako je například paměť nebo specifické dovednosti. Validita zahrnuje různé aspekty měření, včetně empirické (praktické) validity, která ověřuje, zda testované výsledky odpovídají reálným situacím, predikční (prognostické) validity, která zkoumá, jak dobře výsledky testu předpovídají budoucí chování, teoretické (konstrukční) validity, která určuje, zda test skutečně měří zamýšlený psychologický konstrukt, a inkrementální validity, která hodnotí, do jaké míry test přispívá k lepšímu pochopení nebo diagnostice ve srovnání s již existujícími metodami. (SVOBODA, strana 20-21)

Psychometrické testy mají mnoho využití v různých odvětvích a mohou být rozděleny do několika kategorií podle toho, jaké vlastnosti měří. Testy se dělí na výkonové testy a testy osobnosti.

Výkonové testy, známé také jako testy schopností, jsou jednou z nejstarších a nejpropracovanějších diagnostických metod v psychologii, zaměřenou na měření výkonu a schopností jednotlivců. Tyto testy hodnotí odpovědi probandů na základě jasných kritérií, jako je správnost nebo úspěšnost řešení úloh, což zajišťuje objektivitu výsledků. Výsledky jsou nezávislé na subjektivních snahách o zkreslení, což umožňuje spolehlivé měření obecně

inteligence nebo specifických dovedností, které jsou relevantní pro konkrétní činnosti. (SVOBODA, strana 46)

Testy osobnosti jsou diagnostické nástroje určené k měření a analýze různých charakteristik a vlastností osobnosti, které jsou projevy vnitřního světa jedince. Osobnost je v tomto kontextu chápána jako dynamická organizace psychických systémů, které určují, jak se člověk přizpůsobuje prostředí. Testy osobnosti mohou být zaměřeny na registrování vnějších projevů osobnosti, které jsou následně interpretovány za účelem identifikace skrytých rysů, vlastností a charakteristik jedince. Na základě těchto pozorování se osobnostní testy dělí na projektivní metody, objektivní testy osobnosti, dotazníky a posuzovací stupnice. Každá z těchto metod má své specifické využití při diagnostice osobnosti, přičemž projektivní testy zkoumají nevědomé procesy, zatímco dotazníky a objektivní testy poskytují kvantifikovatelné údaje o vědomých aspektech osobnosti. (SVOBODA, strana 148-149)

Psychometrické testy mají široké uplatnění v různých oblastech psychologie a příbuzných disciplínách. Jsou velice důležité pro:

- Diagnostiku: Pomáhají odborníkům identifikovat a klasifikovat psychické poruchy a poskytují empirická data pro rozhodnutí o léčbě.
- Výzkum: Umožňují vědcům testovat hypotézy a ověřovat teorie o lidském chování a osobnosti, čímž přispívají k teoretickému pokroku.
- Výběr zaměstnanců: Jsou důležitým nástrojem při hodnocení kandidátů na pracovní pozice, protože umožňují objektivní srovnání jejich schopností a osobnostních rysů.

1.2 Testování statistických hypotéz

Statistické hypotézy hrají klíčovou roli v psychometrii, kde jsou používány k testování předpokladů o psychologických vlastnostech a chování jedinců nebo skupin. V této kapitole se zaměříme na definici statistických hypotéz a jejich význam, které jsou nezbytné pro správnou interpretaci výsledků v psychometrických testech. Dále se budeme věnovat typům hypotéz, konkrétně rozlišení mezi nulovou a alternativní hypotézou, které tvoří základ pro rozhodovací procesy ve výzkumu psychometrických vlastností.

1.2.1 Základní pojmy

Testování statistických hypotéz představuje hlavní pilíře, pomocí nichž se na základě výsledků zjištěných náhodným výběrem ze základního souboru ověřuje určitý předpoklad, tedy hypotéza, o parametru nebo rozdělení základního souboru. V tomto kontextu vždy klademe proti sobě dvě hypotézy: nulovou a alternativní.

Nulová hypotéza (někdy též nazývána testovaná hypotéza), značená jako H_0 , představuje tvrzení, které je považováno za pravdivé, dokud není statisticky vyvráceno. Pokud označíme

Θ jako neznámý parametr základního souboru, můžeme vytvořit nulovou hypotézu, která tvrdí, že Θ je rovno určité hodnotě Θ_0 :

$$H_0 : \Theta = \Theta_0.$$

Alternativní hypotéza, označená jako H_1 , je tvrzením, které se snaží nulovou hypotézu popřít. Alternativní hypotéza může být buď dvoustranná, nebo jednostranná. Dvoustranná alternativní hypotéza tvrdí, že parametr základního souboru je různý od hodnoty nulové hypotézy. Zápis je tedy

$$H_1 : \Theta \neq \Theta_0.$$

V případě, že nulová hypotéza vymezuje obor hodnot parametru Θ nalevo od hodnoty Θ_0 , zapisujeme ji

$$H_0 : \Theta \leq \Theta_0.$$

Obdobně se postupuje pro

$$H_0 : \Theta \geq \Theta_0.$$

(BLATNÁ, strana 80)

(KAŇOKOVÁ, strana 346-347)

Statistický test je nástrojem, který nám umožňuje rozhodnout, zda na základě zjištěných dat můžeme zamítnout nulovou hypotézu ve prospěch alternativní hypotézy. Tento rozhodovací proces probíhá pomocí testového kritéria (testové či testovací statistiky) vypočteného z náhodného výběru. (BLATNÁ, strana 80)

Testové kritérium, které se používá při testování hypotéz, může mít různý tvar v závislosti na typu testu a hypotézy, kterou se snažíme ověřit. Každé testové kritérium se řídí určitým rozdělením, jako například standardizovaným normálním rozdělením, Studentovým rozdělením, χ^2 rozdělením, F-rozdělením a dalšími. Hodnoty testového kritéria se vypočítávají na základě dat získaných z náhodného výběru, a jelikož se jedná o statistickou veličinu, může nabývat různých hodnot. Tyto hodnoty se obvykle dělí do dvou oddělených oborů: oboru kritického („W“), kde nulovou hypotézu zamítáme, a oboru přijetí („A“), kde nulovou hypotézu nezamítáme. (BLATNÁ, strana 80) (NOVIČKOVÁ, strana 100-101)

Kritický obor představuje hodnoty testového kritéria, které naznačují, že alternativní hypotéza je pravdivá, a proto jsou tyto hodnoty neslučitelné s platností nulové hypotézy. Rozhodnutí o tom, zda zamítnout nebo nezamítnout nulovou hypotézu, se provádí na základě porovnání vypočítané hodnoty testového kritéria s kritickou hodnotou. Pokud hodnota testového kritéria spadá do kritického oboru, zamítáme nulovou hypotézu a přijímáme alternativní hypotézu. Naopak, pokud hodnota spadá do oboru přijetí, nulovou hypotézu nezamítáme. (BLATNÁ, strana 80)

1.2.2 Chyby při testování hypotéz

Při testování hypotéz vyvozujeme závěry na základě výsledků z náhodně vybraných dat, což s sebou nese určitou míru pravděpodobnosti. To znamená, že naše rozhodnutí může být buď správné, nebo můžeme udělat jednu ze dvou druhů chyb. Následující tabulka ilustruje možné vztahy mezi hypotézou a rozhodnutím, včetně rizika těchto chyb:

Nulová hypotéza H_0	Správná (pravdivá)	Nesprávná (nepravdivá)
Nezamítáme	Správné rozhodnutí $P = 1 - \alpha$	Chyba II. druhu $P = \beta$
Zamítáme	Chyba I. druhu $P = \alpha$	Správné rozhodnutí $P = 1 - \beta$

Tabulka 1 – Vztah mezi hypotézou a rozhodnutím, zdroj: (Blatná, strana 81)

Chyba I. druhu nastává, když nesprávně zamítneme pravdivou nulovou hypotézu H_0 . Pravděpodobnost této chyby se nazývá hladina významnosti a je označována jako α . Na druhou stranu, chyba II. druhu se vyskytuje, když nezamítneme nulovou hypotézu, která je ve skutečnosti nepravdivá. Pravděpodobnost této chyby je označena jako β .

Při výběru vhodného testu hraje důležitou roli síla testu, která je definována jako veličina $1 - \beta$, a představuje pravděpodobnost, že správně zamítneme nepravdivou nulovou hypotézu. Během testování hypotéz se snažíme minimalizovat pravděpodobnost chyby I. druhu tím, že volíme nízkou hladinu významnosti, například 0,05 nebo 0,01. Avšak snížení pravděpodobnosti chyby I. druhu zároveň zvyšuje pravděpodobnost chyby II. druhu. Zvýšením rozsahu výběru lze snížit pravděpodobnost obou typů chyb. Když je k dispozici více testů, dáváme přednost tomu, který při stejné hladině významnosti nabízí vyšší sílu testu, což znamená, že má nižší pravděpodobnost chyby II. druhu. (BLATNÁ, strana 81)

1.2.3 Postup testování hypotéz

Proces testování statistických hypotéz začíná stanovením nulové hypotézy H_0 a alternativní hypotézy H_1 . Dále se volí hladina významnosti α , která určuje pravděpodobnost chyby I. druhu. Následně se vybírá vhodný statistický test a testové kritérium s ohledem na povahu základního souboru a velikost výběru. Z výběrových dat se poté vypočítá hodnota testového kritéria, která se porovná s kritickou hodnotou. Na základě tohoto porovnání se rozhodne, zda nulovou hypotézu zamítáme nebo nezamítáme. Pokud hodnota testového kritéria padne do kritického oboru, nulovou hypotézu zamítáme, jinak ji nezamítáme.

V počítačových programech používaných pro testování hypotéz je k hodnotám testových kritérií často připojena vypočítaná p-hodnota, která představuje minimální hladinu významnosti, na jejímž základě lze zamítnout nulovou hypotézu. Pokud je tato p-hodnota menší než zvolená hladina významnosti α , zamítáme testovanou nulovou hypotézu H_0 . Naopak, pokud je p-hodnota větší nebo rovna hladině významnosti α , nulovou hypotézu nezamítáme. (BLATNÁ, strana 81)

1.3 Neparametrické testy

Při testování statistických hypotéz si můžeme zvolit z různých statistických testů, které se obecně dělí na parametrické a neparametrické. Volba mezi těmito dvěma typy závisí na různých kritériích, jako je povaha dat, předpoklady o jejich rozdělení a velikost výběru. V této kapitole se zaměříme na neparametrické testy. Prozkoumáme jejich výhody a nevýhody ve srovnání s parametrickými testy a dále se zaměříme na vhodné situace pro jejich použití.

1.3.1 Výhody a nevýhody neparametrických testů

Neparametrické testy nabízejí několik výhod oproti parametrickým testům, a to především tím, že nevyžadují dodržení tolika předpokladů pro jejich správné použití. Díky této flexibilitě mají neparametrické testy široké uplatnění a jsou méně citlivé na případné porušení teoretických předpokladů. Významnou vlastností těchto testů je skutečnost, že nejsou závislé na předpokladu normality rozdělení, což znamená, že je lze použít pro výběry z libovolného spojitého rozdělení. (BLATNÁ, strana 90)

Další výhodou je jejich vhodnost pro malé výběry, kde tradiční parametrické testy nemusí být spolehlivé. Předpokladem pro použití neparametrických testů je zpravidla pouze spojitost (někdy i symetrie) rozdělení, z něhož jsou data vybrána. Hlavním omezením těchto testů je však menší síla testu $1 - \beta$ ve srovnání s odpovídajícími parametrickými testy. (BLATNÁ, strana 90)

1.3.2 Situace vhodné pro použití neparametrických testů

Neparametrické testy jsou široce využívány v situacích, kdy klasické parametrické metody nejsou vhodné z důvodu nesplnění klíčových předpokladů. Jedním z hlavních důvodů pro jejich použití je skutečnost, že některé soubory dat nelze podřídít přísným požadavkům parametrických testů, jako je například normalita rozdělení, nezávislost měření, předpoklad stejné variability nebo známého poměru variací. (REITEROVÁ, strana 20)

V psychologickém výzkumu se neparametrické testy často používají ve třech hlavních situacích. První situací je, když jsou k dispozici pouze data klasifikační nebo pořadová, tedy data, která lze rozdělit do různých kategorií nebo seřadit podle určitého kritéria. Tyto typy dat nesplňují předpoklady parametrických testů, které obvykle vyžadují intervalová nebo poměrová data. (REITEROVÁ, strana 20)

Druhou situací je, když nelze zjistit, zda data splňují předpoklady pro použití parametrických testů, nebo pokud je prokázáno, že tyto předpoklady neplatí. V těchto případech jsou parametrické testy nevhodné a neparametrické metody poskytují alternativu, která umožňuje analyzovat data i bez znalosti přesného rozdělení. (REITEROVÁ, strana 20)

Třetí situací je, když máme k dispozici pouze malý počet dat. Parametrické testy jsou při práci s malými vzorky méně spolehlivé, neboť jejich předpoklady o rozdělení nemusí být přesně splněny. Naopak neparametrické testy jsou méně citlivé na velikost vzorku

a poskytují robustnější výsledky i v případech, kdy je počet pozorování omezený. (REITEROVÁ, strana 20)

Vzhledem k výhodám a nevýhodám neparametrických testů jsou užity ve chvíli, kdy nám jde o předběžné a orientační výsledky. Po jejich použití můžeme usoudit, zda má cenu pokračovat v další složitější a zdlouhavější analýze parametrickými testy. Parametrické testy volíme tedy pouze tehdy, jsou-li splněny veškeré podmínky pro jejich použití. (REITEROVÁ, strana 20)

2 Teoretická část

V této teoretické části se zaměřujeme na základní charakteristiky a použití neparametrických testů, které jsou často využívány v psychometrických analýzách, když data nesplňují předpoklady pro aplikaci parametrických testů. Tato kapitola postupně představí obecné definice těchto testů, vysvětlí jejich předpoklady a podmínky, za kterých je vhodné je použít, a detailně popíše postupy výpočtu, které jsou s nimi spojené. V rámci jednotlivých testů bude rozebrána jejich teoretická základna, interpretace výsledků a jejich možné aplikace.

2.1 McNemarův test

McNemarův test je neparametrický statistický test, který se používá k testování změn u párových kategoriálních dat. Typicky se používá k porovnání dvou skupin testovaných osob, například k hodnocení účinnosti léčby nebo k porovnání dvou diagnostických testů provedených na stejných subjektech. Tento test je vhodný zejména pro případy, kdy jsou data ve formě dichotomických odpovědí (např. ano/ne, pravda/nepravda). (REITEROVÁ, strana 21)

Test vyžaduje 2x2 kontingenční tabulku:

	Po zásahu pozitivní (+)	Po zásahu negativní (-)	Σ
Před pozitivní (+)	n_{11}	n_{12}	$n_{1.}$
Před negativní (-)	n_{21}	n_{22}	$n_{2.}$
Σ	$n_{.1}$	$n_{.2}$	n

Tabulka 2 – Kontingenční tabulka pro McNemarův test, zdroj: (ANDĚL, strana 217)

Buňky představují následující kategorie:

- n_{11} : počet subjektů s pozitivním výsledkem v obou podmínkách (např. před a po léčbě),
- n_{12} : počet subjektů s pozitivním výsledkem v první podmínce a negativním výsledkem ve druhé podmínce,
- n_{21} : počet subjektů s negativním výsledkem v první podmínce a pozitivním výsledkem ve druhé podmínce,
- n_{22} : počet subjektů s negativním výsledkem v obou podmínkách.

McNemarův test se zaměřuje na hodnoty n_{12} a n_{21} , které reprezentují neshody mezi dvěma podmínkami. Statistika McNemarova testu je poté vypočítána pomocí vzorce

$$\chi^2 = \frac{(n_{12} - n_{21})^2}{n_{12} + n_{21}}.$$

Test by se však neměl používat v případě, jsou-li očekávané četnosti příliš malé. V případě menších četností je vhodné použít pro výpočet χ^2 vzorec s Yatesovou korekcí

$$\chi^2 = \frac{(|n_{12} - n_{21}| - 1)^2}{n_{12} + n_{21}}$$

(REITEROVÁ, strana 21-22)

2.2 Cochranův Q-test

Cochranův Q-test představuje rozšíření McNemarova testu pro více než dva závislé výběry. Tento test se používá k testování významnosti změn mezi k závislými výběry a zjišťuje, zda se distribuce určitého jevu, například správných odpovědí, úspěchů apod., liší mezi jednotlivými výběry. Data jsou alternativní, tedy např. kladná nebo záporná odpověď za k různých podmínek. (REITEROVÁ, strana 23)

Nulová hypotéza testuje předpoklad, že distribuce kladných odpovědí je ve všech výběrech stejná, tj. pravděpodobnost kladné odpovědi je ve všech výběrech stejná. Získané údaje se uspořádají do tabulky o n řádcích a k sloupcích, přičemž do každého pole tabulky se запиše buď 1 pro úspěch či kladnou odpověď, nebo 0 pro opačný případ. (REITEROVÁ, strana 23)

Tabulka tedy vypadá takto:

	Léčba 1	Léčba 2	...	Léčba k	Σ
Skupina 1	X_{11}	X_{12}	...	X_{1k}	$u_{1.}$
Skupina 2	X_{21}	X_{22}	...	X_{2k}	$u_{2.}$
Skupina 3	X_{31}	X_{32}	...	X_{3k}	$u_{3.}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
Skupina n	X_{n1}	X_{n2}	...	X_{nk}	$u_{n.}$
Σ	$T_{.1}$	$T_{.2}$...	$T_{.k}$	N

Tabulka 3 – Dichotomická tabulka pro Cochranův Q-test, zdroj: (ANDĚL, strana 217)

Každá buňka v tabulce označená jako X_{ij} obsahuje binární hodnotu, kde 1 představuje úspěch (kladnou odpověď) a 0 neúspěch (zápornou odpověď). Index i označuje řádek (pozorování) a index j označuje sloupec (výběr).

Testová statistika se vypočítá pomocí testového kritéria Q :

$$Q = \frac{(k-1) \left[k \cdot \sum_{j=1}^k T_j^2 - \left(\sum_{j=1}^k T_j \right)^2 \right]}{k \cdot \sum_{i=1}^n u_i - \sum_{i=1}^n u_i^2},$$

kde:

- T_j je celkový počet kladných odpovědí (označených 1) v j -tém sloupci,

- u_i je celkový počet těchto odpovědí v i -tém řádku,
- n je počet řádků,
- k je počet sloupců.

Velikost Q má χ^2 rozdělení s $k - 1$ stupni volnosti a významnost výsledku lze určit pomocí tabulky kritických hodnot tohoto rozdělení. (REITEROVÁ, strana 23)

2.3 Mann-Whitneyův U-test

Mann-Whitneyův U-test se používá pro dva nezávislé výběry a je jedním z nejběžněji používaných a nejsilnějších neparametrických testů. Tento test vychází z pořadových hodnot a ověřuje, zda dva náhodné výběry pocházejí ze stejné populace nebo ze dvou identických základních souborů. Skóry obou výběrů jsou seřazeny do vzestupné posloupnosti. Testové kritérium pak zjišťuje počet inverzí, označovaných jako U . Pro každý prvek z první skupiny zjišťujeme, kolik pozorování z druhé skupiny je před ním v seřazeném pořadí. Tímto získáme dva údaje o počtu inverzí U a U' a jako testové kritérium se volí menší z těchto dvou hodnot. (REITEROVÁ, strana 25)

Pro výpočet U se seřadí skóry obou výběrů vzestupně podle velikosti do jedné posloupnosti, přiřadí se jim pořadová čísla, a tato čísla se následně vrátí zpět do původní skupiny. Poté se sečtou pořadová čísla pro každou skupinu zvlášť, označené jako T_1 a T_2 . Hodnoty U a U' jsou vypočítány jako:

$$U = n_1 \cdot n_2 + \left[\frac{n_1 \cdot (n_1 + 1)}{2} \right] - T_1,$$

$$U' = n_1 \cdot n_2 + \left[\frac{n_2 \cdot (n_2 + 1)}{2} \right] - T_2.$$

Tyto hodnoty spolu souvisejí vztahem

$$U + U' = n_1 \cdot n_2,$$

kde n_1 a n_2 jsou velikosti obou výběrů.

(REITEROVÁ, strana 25)

Mann a Whitney stanovili za předpokladu platnosti nulové hypotézy distribuci U , která může být použita pokud $n_1 \leq 20$ a $n_2 \leq 20$. Pokud jsou výběry větší než 20, testovací statistika U může být aproximována normálním rozdělením. V takovém případě se pro test významnosti používají tabulky kvantilů normálního rozdělení, kde je vypočtená hodnota statistiky z podle vzorce:

$$z = \frac{U - \frac{n_1 \cdot n_2}{2}}{\sqrt{\frac{n_1 \cdot n_2 (n_1 + n_2 + 1)}{12}}}$$

kteřá je pouřžita k určení, zda zamítnout či nezamítnout nulovou hypotézu. (REITEROVÁ, strana 25-26)

2.4 Friedmanův test

Friedmanův test je neparametrickou obměnou parametrické analýzy rozptylu pro ordinální data. Pracuje s k nezávislými výběry se stejným rozsahem. Typicky se jedná o výsledky stejných subjektů měřených za k různých podmínek. Tento test má výhodu v tom, že nemusíme brát v úvahu normalitu rozdělení, což je požadavek běžně kladený na parametrické testy. Neparametrický postup statistiky vychází s pořadových hodnot. Je tedy jednodušší, flexibilnější, a využívá se v širším spektru výzkumných situací. Jeho nevýhodou je však ztráta informací, která nastává, když místo původních naměřených hodnot používáme jejich pořadí. (REITEROVÁ, strana 28)

Základem testu je tabulka dvojného třídění s n řádky a k sloupci, kde n představuje výběrový rozsah pro všechny skupiny a k počet výběřů. Tento postup se označuje jako model náhodných bloků, kde se zkoumá vliv k ošetření, která jsou uplatněna na n blocích. Podle výsledků je potřeba rozhodnout, zda všechna ošetření mají stejný efekt. Pokud stejný efekt nemají, určuje se, které se liší. V každém řádku tabulky jsou hodnoty nahrazeny jejich pořadovým číslem od nejnižší po nejvyšší. Výpočet testové statistiky je založen na porovnání průměrných pořadí v jednotlivých sloupcích, kde testová statistika má rozdělení χ^2 s $k - 1$ stupni volnosti. Výpočet lze vyjádřit následovně:

$$Q = \frac{12}{n \cdot k(k + 1)} \sum_{j=1}^k \left(\sum_{i=1}^n R_{ij} \right)^2 - 3n(k + 1),$$

kde R_{ij} je pořadí v i -tém řádku a j -tém sloupci.

(REITEROVÁ, strana 28-29)

(ANDĚL, strana 233)

Pro kontrolu výpočtů se používá vztah $\sum_i \sum_j R_{ij} = \frac{n \cdot k(k+1)}{2}$. Za platnosti hypotézy má Q při $n \rightarrow \infty$ asymptoticky χ_{k-1}^2 rozdělení. Když Q dosáhne nebo přesáhne kritickou hodnotu $\chi_{k-1}^2(\alpha)$, hypotéza se zamítá. Pro malé vzorky je třeba využít speciální tabulky kritických hodnot. (ANDĚL, strana 233)

Jednotlivé rozdíly lze dále zkoumat pomocí Neményiho metody. Označením $\sum_i R_{ij} = R_j$ lze vytvořit tabulku hodnot $|R_j - R_m|$ pro všechna $j < m$. Pokud některá z hodnot $|R_j - R_m|$ je větší, nebo rovna kritické hodnotě, zamítá se hypotéza na zvolené hladině významnosti, která tvrdí, že j -té a m -té ošetření mají stejný efekt. (ANDĚL, strana 233)

2.5 Kruskal-Wallisův test

Kruskal-Wallisův test je, obdobně jako Friedmanův test, neparametrickou obdobou analýzy rozptylu. Test je zobecněním Wilcoxonova dvouvýběrového testu na případ k výběrů ($k \geq 3$). Tato metoda slouží k testování homogenity rozdělení sledovaného znaku v různých populacích, kde máme k dispozici nezávislé náhodné výběry. Podle nulové hypotézy pocházejí všechny výběry z téže populace, tj. že mají stejnou distribuční funkci. Obvykle tuto hypotézu testujeme parametrickou analýzou rozptylu, která však vyžaduje, aby výběry byly nezávislé, pocházely z normálně distribuovaných populací při dodržení homoskedasticity. Tyto předpoklady nejsou vždy splněny. Kruskal-Wallisův test nabízí výhodu, že zeslabuje předpoklady a umožňuje jednodušší výpočty. (REITEROVÁ, strana 30) (ANDĚL, strana 233)

V Kruskal-Wallisově testu pracujeme s nezávislými výběry n_1, n_2, \dots, n_k , kde n_i představuje velikost výběru z i -té populace a k je počet srovnávaných výběrů. Všechny n prvků z k výběrů seřadíme do rostoucí posloupnosti a následně se určí pořadí každého prvku. Označme pak T_i jako součet pořadí těch prvků, které patří do i -tého výběru ($i = 1, 2, \dots, k$). Jelikož musí platit $T_1 + \dots + T_k = \frac{n(n+1)}{2}$, můžeme ověřit, zda jsou T_i vypočtena správně. Testová statistika Q se počítá pomocí následujícího vzorce:

$$Q = \frac{12}{n(n+1)} \sum_{i=1}^k \frac{T_i^2}{n_i} - 3(n+1),$$

kde T_i představuje součet pořadových hodnot v j -tém výběru a n je součet rozsahů jednotlivých výběrů, se kterými pracujeme:

$$n = \sum_{i=1}^k n_i$$

a n_i je velikost i -tého výběru.

(REITEROVÁ, strana 30-31)

(ANDĚL, strana 233)

Za předpokladu, že rozsahy výběrů nejsou příliš malé (ne menší než 5), má veličina Q přibližně χ^2 rozdělení s $k-1$ stupni volnosti. Proto pro test významnosti můžeme použít standardní tabulky kritických hodnot χ^2 distribuce. (REITEROVÁ, strana 31)

V případě $Q \geq \chi_{k-1}^2(\alpha)$ zamítáme nulovou hypotézu na hladině významnosti α . Pokud je nulová hypotéza zamítnuta, znamená to, že rozdělení hodnot mezi výběry není homogenní. Je potřeba se zaměřit na identifikaci dvojic výběrů, které se od sebe významně liší. Pro tento účel se obvykle používají post-hoc testy, jako je Tukeyův nebo Scheffeho test. Pokud má

Kruskal-Wallisův test zamítnout hypotézu, porovnávají se hodnoty T_i , přičemž se zkoumá rozdíl mezi hodnotami. (ANDĚL, strana 233)

V případě, že všech k výběrů má stejný rozsah ($n_1 = n_2 = \dots = n_k = N$), lze Kruskal-Wallisův test doplnit pomocí Neményiho metody. Pokud je číslo $|T_i - T_j|$ větší, nebo rovno kritické hodnotě, zamítá se nulová hypotéza, že i -tý a j -tý výběr pochází ze stejného rozdělení, přičemž tento postup se aplikuje na všech $\frac{k(k+1)}{2}$ číslech $|T_i - T_j|$. (ANDĚL, strana 233)

2.6 Korelační koeficienty

Uvažujme dvě náhodné veličiny X a Y s konečnými druhými momenty, kde platí, že $\text{var } X > 0$ a $\text{var } Y > 0$. Tyto veličiny X a Y mohou být závislé nebo nezávislé. Pokud jsou závislé, lze jejich závislost kvantitativně ohodnotit. Protože tato závislost může mít různou povahu, je třeba najít optimální statistický ukazatel pro měření této závislosti. Nejčastěji se k tomu používá Pearsonův korelační koeficient, který je definován vzorcem:

$$\rho = \frac{\text{cov}(X, Y)}{\sqrt{(\text{var}(X) \cdot \text{var}(Y))}}$$

kde:

- $\text{cov}(X, Y)$ značí kovarianci náhodných veličin X a Y
- $\text{var } X$ a $\text{var } Y$ představují rozptyly jednotlivých veličin
- ρ značí Pearsonův korelační koeficient

Místo označení q se někdy používá symbol $\rho_{X,Y}$, aby bylo jasné, o které veličiny se jedná. Korelační koeficient je symetrický, což znamená, že platí $\rho_{X,Y} = \rho_{Y,X}$ (nezávisí na pořadí veličin X a Y). Pokud je $\rho_{X,Y} = 0$, znamená to, že veličiny jsou nezávislé, avšak opačné tvrzení obecně platí. (ANDĚL, strana 34)

Pearsonův korelační koeficient je parametrickou metodu pro měření lineární závislosti mezi dvěma náhodnými veličinami. Nicméně, v praxi se můžeme setkat s daty, která nesplňují předpoklady normálního rozdělení nebo kde vztah mezi proměnnými není lineární. V těchto případech se ukazují jako vhodnější neparametrické metody korelace, mezi které patří Spearmanův korelační koeficient a Kendallovo tau.

2.6.1 Spearmanův korelační koeficient

Spearmanův korelační koeficient se používá v situacích, kdy nelze aplikovat běžný korelační koeficient, což může být důsledkem porušení předpokladu normality nebo jiných důvodů. Někdy není možné zjistit hodnoty náhodných veličin přesně, a tak se používá pořadí těchto veličin. Pokud jsou pořadí hodnot veličin X_1, \dots, X_n a Y_1, \dots, Y_n podobná, můžeme předpokládat, že existuje závislost mezi X_i a Y_i , kde $i = 1, \dots, n$. Tento vztah měří Spearmanův korelační koeficient r_s . (ANDĚL, strana 234)

Pokud máme náhodný výběr $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$ ze spojitého dvourozměrného rozdělení, můžeme testovat hypotézu, že veličiny X_i a Y_i jsou nezávislé. Hodnoty veličin X_1, \dots, X_n se seřadí podle velikosti a stanoví se jejich pořadí R_1, \dots, R_n . Stejným způsobem se postupuje i pro veličiny Y_1, \dots, Y_n a jejich pořadí se označí jako Q_1, \dots, Q_n . Spearmanův korelační koeficient r_s se následně vypočítá podle vzorce:

$$r_s = 1 - \frac{6}{n \cdot (n^2 - 1)} \sum_{i=1}^n (R_i - Q_i)^2,$$

kde n značí počet dvojic pozorování (X_i, Y_i) , tedy velikost výběru. (ANDĚL, strana 234–235)

V tabulkách se většinou udávají kritické hodnoty pro r_s v případě, že $n \leq 30$. Pokud r_s překročí nebo je rovno tabulkovým kritickým hodnotám, hypotéza o nezávislosti je zamítnuta. Pro $n > 30$ platí, že se hypotéza zamítá v případě, že:

$$|r_s| \geq \frac{\Phi^{-1}\left(1 - \frac{\alpha}{2}\right)}{\sqrt{n-1}},$$

kde $\Phi^{-1}\left(1 - \frac{\alpha}{2}\right)$ značí hodnotu kvantilu standardního normálního rozdělení $N(0, 1)$ pro oboustranný test. Obdobným způsobem lze konstruovat jednostranný test. (ANDĚL, strana 235)

V mnoha příručkách se tento test často opírá pouze o hodnoty veličiny $D = \sum_{i=1}^n (R_i - Q_i)^2$, a jsou k tomu přizpůsobeny i tabulky kritických hodnot. Výpočet D se zdůvodňuje tím, že je o něco jednodušší než výpočet celého r_s . Když však známe r_s , poskytuje nám nejen informace o významnosti či nevýznamnosti testu, ale také o síle zkoumané statistické závislosti, což může být důležité při interpretaci výsledků. (ANDĚL, strana 235)

2.6.2 Kendallův korelační koeficient

Kendallův koeficient pořadové korelace slouží k měření síly a směru závislosti mezi dvěma proměnnými, které jsou vyjádřeny pořadovými hodnotami. Předpokládejme, že máme dvojice (X_i, Y_i) , kde hodnoty X_i jsou seřazeny v neklesající posloupnosti. Pro $i = 1, \dots, n-1$ označme k_i počet těch hodnot Y_j , kde platí, že $Y_j > Y_i$ a $j > i$. Pak je Kendallův koeficient pořadové korelace definován následujícím vztahem:

$$\tau = \frac{4}{n \cdot (n-1)} \sum_{i=1}^n k_i - 1.$$

(REIF, strana 166)

U tohoto koeficientu vždy platí $-1 \leq \tau \leq 1$ ve stejných situacích, jako u Spearmanova korelačního koeficientu, přičemž hodnota 1 znamená úplnou pozitivní pořadovou závislost a hodnota -1 úplnou negativní pořadovou závislost. Pokud hodnota τ dosahuje hodnoty

blízké nule, indikuje to slabou nebo žádnou závislost mezi pořadovými hodnotami. Kritické hodnoty τ , při kterých je možné zamítnout nulovou hypotézu nezávislosti, jsou určeny pomocí tabulek. Pro velké hodnoty n se statistika τ přibližuje normálnímu rozdělení se střední hodnotou 0 a rozptylem přibližně $\frac{4}{9n}$. (REIF, strana 166)

2.7 Faktorová analýza

Faktorová analýza je matematicko-statistická metoda, kterou využívají psychologové k řešení specifických klasifikačních problémů, zejména v psychologii osobnosti. Jejím hlavním cílem je redukovat velké množství vlastností, běžně užívaných při popisu osobnosti, na menší počet základních faktorů, které jsou zdrojem jejich vzájemných korelací. Korelují-li dva nebo více testů, pak tyto testy obsahují společný rozptyl, označovaný jako faktorový rozptyl, což je měřítko jejich společného vlivu. (REITEROVÁ, strana 67)

Zakladatelem faktorové analýzy je Charles Spearman, který předpokládal, že každá baterie testů inteligence může být vysvětlena ve svých účincích pomocí obecného faktoru působícího ve všech korelačních maticích a specifických faktorů, jejichž četnost odpovídá počtu proměnných (testů). Často se proto používá Thurstonův model vícenásobné faktorové analýzy, kde je každá proměnná (test) t_j , kde $j = 1, \dots, n$, popsána rovnicí:

$$t_j = a_{j1}F_1 + a_{j2}F_2 + \dots + a_{jm}F_m + a_j u_j,$$

kde F_1, F_2, \dots, F_m jsou společné faktory a u_j je jedinečný faktor proměnné t_j . (REITEROVÁ, strana 67)

Cílem faktorové analýzy je určení koeficientů a_{jm} (faktorových nábojů, vah) společného faktoru. Výchozími daty jsou korelace mezi proměnnými. U kteréhokoliv testu lze vyjádřit jeho celkový rozptyl jako součet rozptylu sdíleného s ostatními testy a jedinečného rozptylu, který není obsažen v žádném jiném testu:

$$\sigma^2 = h^2 + u^2 = 1,$$

kde h^2 je komunalita a u^2 jedinečnost testu. Komunalita h^2 je součet druhých mocnin všech faktorových nábojů v testu:

$$h_j^2 = a_{j1}^2 + a_{j2}^2 + \dots + a_{jm}^2.$$

Jestliže společné faktory vysvětlují proměnnou vyčerpávajícím způsobem, musí se h_j^2 blížit hodnotě 1. (REITEROVÁ, strana 67)

Pokud test měří pouze jeden faktor, označuje se jako faktorově čistý. Test, který je nasycen nebo nabit více než jedním faktorem, je označován jako faktorově komplexní. Jedním z výstupů faktorové analýzy je faktorová matice, což je tabulka koeficientů vyjadřujících vztahy mezi testy a faktory, které jsou jejich základem. Údaje v tabulce se nazývají faktorové náboje a mají rozpětí od -1 do $+1$, podobně jako korelační koeficienty. V posledním sloupci faktorové matice jsou zapsány komunality h^2 . K faktorové matici se dojde pomocí

speciálních metod faktorové analýzy z korelační matice, ve které je vypočítána Pearsonova korelace pro všechny testy. (REITEROVÁ, strana 68)

2.7.1 Analýza hlavních komponent (PCA)

Analýza hlavních komponent (PCA) je jednou z nejpoužívanějších metod pro redukcí dimenzionality dat. Tato metoda se zaměřuje na transformaci původních datových proměnných na nový soubor proměnných, tzv. hlavních komponent, které jsou navzájem ortogonální a maximalizují vysvětlený rozptyl dat.

Uvažujme, že máme p -rozměrný prostor E_p , ve kterém jsou data reprezentována body. Nad těmito body lze sestavit nadelipsoid, jehož hlavní osy odpovídají směrům maximálního rozptylu dat. Poloha hlavních os nadelipsoidu je určena tak, že maximalizujeme rozptyl dat v daném směru. Formálně se jedná o nalezení maxima funkce

$$f(\mathbf{u}) = \mathbf{u}^T \mathbf{u}$$

za podmínky

$$\mathbf{u}^T \mathbf{K}^{-1} \mathbf{u} = c,$$

kde \mathbf{K}^{-1} je inverzní matice ke kovariační matici \mathbf{K} vypočtené z dané matice dat, \mathbf{u} je sloupcový vektor souřadnic bodů $U = (u_1, u_2, \dots, u_p)$, \mathbf{u}^T je řádkový vektor transponovaný k vektoru \mathbf{u} a c je konstanta. (LUKASOVÁ, strana 26)

Při zpracování mnohorozměrných dat metodu PCA se data nejprve standardizují a pro standardizovaná data se určí kovarianční matice. Tuto matici budeme dále značit \mathbf{K} .

Problém nalezení směru hlavní osy nadelipsoidu, který maximalizuje rozptyl dat, lze formálně popsat jako problém nalezení vlastního vektoru \mathbf{u} odpovídajícího největšímu vlastnímu číslu λ matice \mathbf{K} , tak aby platila rovnice:

$$\mathbf{K} \mathbf{u} = \lambda \mathbf{u}.$$

Vektor \mathbf{u} určuje směr nejdelší osy nadelipsoidu a jeho norma je rovna $\lambda^{1/2}$. Druhý vlastní vektor, který je kolmý na první, odpovídá druhému největšímu vlastnímu číslu matice \mathbf{K} , a určuje směr druhé nejdelší osy nadelipsoidu. Tímto způsobem určujeme i další osy nadelipsoidu, přičemž všechny tyto osy jsou vzájemně kolmé a jejich délky odpovídají odmocninám z odpovídajících vlastních čísel matice. (LUKASOVÁ, strana 27)

Tento postup lze chápat i z geometrického hlediska jako otáčení soustavy souřadnic do nové soustavy, kde nové souřadnice jsou navzájem kolmé, což jsou hlavní komponenty. Tyto hlavní komponenty jsou navzájem nekorelované a přispívají k celkové variabilitě dat.

Této vlastnosti lze využít k zobrazení bodu prostoru E_p ($p > 2$) do prostoru E_2 . Pro toto zobrazení používáme pouze prvních dvou souřadnic (hlavních komponent). Tím je zachováno co nejvíce původní variability dat. Tento proces zahrnuje normalizaci vektorů na

jednotkovou normu, přičemž složky pak představují směrové kosiny odchylek nových os otočené soustavy od os původní soustavy. (LUKASOVÁ, strana 27)

Pro nalezení řešení maticové rovnice provedeme úpravu na tvar

$$(\mathbf{K} - \lambda \mathbf{I})\mathbf{u} = \mathbf{o},$$

kde \mathbf{I} je jednotková matice a \mathbf{o} je nulový vektor. Rovnice pak představuje soustavu homogenních lineárních rovnic:

$$(k_{11} - \lambda)\mathbf{u}_1 + k_{12}\mathbf{u}_2 + \dots + k_{1p}\mathbf{u}_p = 0,$$

$$k_{21}\mathbf{u}_1 + (k_{22} - \lambda)\mathbf{u}_2 + \dots + k_{2p}\mathbf{u}_p = 0,$$

⋮

$$k_{p1}\mathbf{u}_1 + k_{p2}\mathbf{u}_2 + \dots + (k_{pp} - \lambda)\mathbf{u}_p = 0.$$

Aby měla soustava netriviální řešení, musí být determinant roven nule. Vlastní číslo λ kovariační matice \mathbf{K} je tedy kořenem charakteristické rovnice matice \mathbf{K} :

$$|\mathbf{K} - \lambda \mathbf{I}| = 0,$$

což znamená

$$\begin{vmatrix} k_{11} - \lambda & k_{12} & \dots & k_{1p} \\ k_{21} & k_{22} - \lambda & \dots & k_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ k_{p1} & k_{p2} & \dots & k_{pp} - \lambda \end{vmatrix} = 0.$$

(LUKASOVÁ, strana 27-28)

Charakteristická rovnice matice \mathbf{K} je p -tého stupně a má tedy p kořenů ${}^1\lambda, {}^2\lambda, \dots, {}^p\lambda$, což jsou vlastní čísla matice \mathbf{K} . Při jejich dosazení do rovnice $(\mathbf{K} - \lambda \mathbf{I})\mathbf{u} = \mathbf{o}$ můžeme vypočítat vlastní vektory ${}^1\mathbf{u}, {}^2\mathbf{u}, \dots, {}^p\mathbf{u}$ odpovídající těmto vlastním číslům. (LUKASOVÁ, strana 28)

Sestavme nyní z vlastních čísel matice \mathbf{K} matici $\mathbf{\Lambda} = (\lambda_{ij})$ tak, že vlastní čísla tvoří prvky její hlavní diagonály, $\lambda_{kk} = {}^k\lambda$ (kde $k = 1, 2, \dots, p$), a pro její ostatní prvky λ_{ij} (pro $i, j = 1, 2, \dots, p$ a $i \neq j$) platí $\lambda_{ij} = 0$. Dále sestavíme matici \mathbf{U} tak, že normalizovaný k -tý vlastní vektor ${}^k\mathbf{u}$ tvoří její k -tý sloupec. Rovnici $\mathbf{K}\mathbf{u} = \lambda\mathbf{u}$ můžeme pak vyjádřit v maticovém tvaru jako:

$$\mathbf{K}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}.$$

(LUKASOVÁ, strana 28)

Protože pro matici \mathbf{U} platí $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$ (kde \mathbf{I} je ortonormální matice), dostáváme z rovnice $\mathbf{K}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}$ násobením zleva maticí \mathbf{U}^T rovnicí:

$$\Lambda = U^T K U,$$

z níž vyplývá, že matice U převádí matici K na diagonální tvar. Z této skutečnosti vyplývají dvě zajímavé vlastnosti vlastních čísel matice K :

1. $\text{tr}(\Lambda) = \text{tr}(K)$, což znamená, že součet vlastních čísel matice K je roven součtu jejich prvků hlavní diagonály.
2. $\prod_{i=1}^p \lambda_i = |K|$, což znamená, že součin vlastních čísel matice K je roven jejímu determinantu.

Nechť Z je matice dat. Pro každý znak vypočítáme střední hodnotu \bar{z}_j ($j = 1, 2, \dots, p$) a transformujeme hodnoty každého znaku na nový počátek – jeho střední hodnotu, čímž získáme matici Y , kde $(y_{ij}) = (z_{ij} - \bar{z}_j)$. Dále provedeme násobení:

$$\frac{1}{n} Y^T Y = K,$$

abychom získali kovariační matici K . Poté přistoupíme ke standardizaci dat, což znamená, že každý znak v matici Z je převeden na jednotkovou směrodatnou odchylku. Konkrétně se pro každý znak z_{ij} standardizovaná hodnota x_{ij} vypočítá jako:

$$x_{ij} = \frac{(z_{ij} - \bar{z}_j)}{s_j},$$

kde s_j je směrodatná odchylka znaku z_j . Tímto způsobem získáme matici X , která představuje standardizovaná data. Standardizace zajišťuje, že všechny proměnné mají průměr 0 a směrodatnou odchylku 1. Následně pomocí této standardizované matice X můžeme vypočítat korelační matici R podle následujícího vztahu:

$$\frac{1}{n} X^T X = R.$$

(LUKASOVÁ, strana 23)

Matice U_K vlastních vektorů kovariační matice vypočtené z matice dat Z transformuje matici Y do nových souřadnic (hlavních komponent) podle rovnice:

$$Y_t = Y U_K,$$

ve které Y_t značí výslednou matici transformovaných dat. (LUKASOVÁ, strana 28)

Podobně matice U_R vlastních vektorů korelační matice, kterou jsme vypočítali z datové matice Z , použijeme k transformaci matice standardizovaných dat X na nové souřadnice X_t podle následující rovnice:

$$X_t = X U_R.$$

Protože hlavní komponenty závisí na lineární transformaci dat, volíme v případě takových dat, jejichž znaky se nám zdají nesouměřitelné, raději druhý z uvedených způsobů transformace. Znovu je třeba připomenout, že pokud počítáme kovariační matici ze standardizovaných dat, je tato matice korelační. (LUKASOVÁ, strana 29)

3 Návrh a implementace aplikace

3.1 Požadavky na aplikaci

Funkční požadavky jsou specifické funkcionality, které by měl systém poskytovat, zatímco nefunkční požadavky se týkají toho, jak by měl systém tyto funkce poskytovat.

Funkční požadavky:

1. **Výběr testu:** Uživatel má možnost vybrat si mezi různými statistickými testy, jako je Mann-Whitney, McNemar, Kruskal-Wallis, atd.
2. **Vstup dat:** Uživatel má možnost zadávat data pro statistické testy ve formě dvou vzorků (ve formě řádků a sloupců).
 - Pro Mann-Whitneyho test je možné přidávat řádky (tedy velikost výběru), ale nikoliv sloupce (další výběry).
 - Pro McNemarův test není možné přidávat sloupce ani řádky, je dostupná pouze 2x2 tabulka
3. **Zobrazení popisků:** Po vybrání testu se u buněk zobrazují popisky pro intuitivní používání.
4. **Výpočet testu:** Program správně umí vypočítat hodnotu daného testu, kritickou hodnotu, p-hodnotu a rozhodnutí o hypotéze.
 - Program umožní uživateli zobrazit výsledky testu po výpočtu ve formě grafu.
5. **Omezení funkcionality:** Program správně omezí své funkcionality u specifických testů
 - Tlačítko pro přidávání sloupců je deaktivováno, pokud je vybrán Mann-Whitneyho test.
 - Tlačítko pro přidávání sloupců a řádků je deaktivováno, pokud je vybrán McNemarův test
6. **Ukládání a načítání dat:** Uživatel má možnost načítat data z CSV souborů a ukládat je zpět do CSV.

Nefunkční požadavky:

1. **Použitelnost:** Uživatelské rozhraní by mělo být intuitivní a snadno použitelné, s jasně označenými vstupy a výstupy.
2. **Výkonnost:** Výpočty statistických testů by měly být prováděny rychle, i při práci s většími datovými sadami.

3. **Spolehlivost:** Program by měl být stabilní a neměl by padat při nesprávných vstupech. Výsledky výpočtů by měly být přesné a spolehlivé.
4. **Kompatibilita:** Program by měl být kompatibilní s různými verzemi systému Windows a mělo by být možné jej snadno spustit na běžných uživatelských počítačích.
5. **Údržba:** Kód by měl být napsán tak, aby byl snadno rozšiřitelný a udržitelný, což zahrnuje přehledné komentáře a konzistentní styl kódování.

3.2 Architektura aplikace a použité technologie

3.2.1 .NET framework

.NET je bezplatná, multiplatformní a open-source vývojová platforma, která se používá k vytváření široké škály aplikací. Podporuje více programovacích jazyků, přičemž nejpobulárnější je C#. Platforma je navržena pro vysoký výkon a spolehlivost, poskytuje automatickou správu paměti pomocí garbage collectoru (GC), typovou bezpečnost a robustní zabezpečení. .NET zahrnuje runtime pro spouštění aplikačního kódu, knihovny pro užitečné funkce jako je parsování JSON, kompilátor a SDK s nástroji pro vytváření a sledování aplikací. (10)

Klíčové vlastnosti .NET zahrnují:

1. **Produktivita a výkon:** .NET je optimalizován pro produktivitu díky plně integrovanému přístupu, který zahrnuje runtime, knihovny, jazyk a nástroje. Podporuje jak statický, tak dynamický kód, což umožňuje širokou škálu vývojových scénářů.
2. **Bezpečnost a interoperabilita:** .NET je navržen tak, aby byl typově a paměťově bezpečný, využívá garbage collector ke správě paměti. Podporuje také interoperabilitu s nativním kódem a poskytuje levný a vysoce věrný přístup k surovým API a hardwarovým instrukcím.
3. **Přenositelnost a přizpůsobivost:** Aplikace napsané v .NET jsou přenosné na různé platformy a procesorové architektury, což činí .NET vhodným pro různé oblasti, jako je cloud computing, klientské aplikace a herní vývoj.
4. **Open Source a komunitně řízený vývoj:** .NET je spravován Microsoftem a komunitou, přičemž zdrojový kód je dostupný na GitHubu. Pravidelně se aktualizuje, aby byla zajištěna bezpečnost a spolehlivost.
5. **Varianty ekosystému:** Existuje několik implementací .NET přizpůsobených pro různé prostředí:
 - **.NET Framework:** Původní .NET, primárně pro Windows.

- **Mono:** Multiplatformní implementace, původně vyvinutá komunitou, podporující Android, iOS a WebAssembly.
- **.NET (Core):** Moderní, multiplatformní verze .NET, optimalizovaná pro cloud a kompatibilní s Linuxem, macOS a Windows.

Celkově je .NET univerzální platforma navržená tak, aby podporovala širokou škálu typů aplikací a vývojových potřeb, od cloudových služeb po mobilní aplikace. (10)

3.2.2 Jazyk C#

C# je objektově orientovaný programovací jazyk, který je součástí platformy .NET, což je bezplatné, multiplatformní a open-source vývojové prostředí. Tento jazyk je určen pro psaní aplikací, které mohou běžet na různých zařízeních, od zařízení pro Internet of Things (IoT) po cloudové servery a všechno mezi tím. C# je všeobecně použitelný jazyk, který je navržen tak, aby byl produktivní při psaní vysoce výkonného kódu. Miliony vývojářů po celém světě používají C#, což z něj činí nejoblíbenější jazyk na platformě .NET. (11)

C# je silně typovaný jazyk, což znamená, že každý deklarovaný typ proměnné je znám už při kompilaci. To pomáhá při odhalování chyb ještě před spuštěním programu. C# podporuje různá programovací paradigmat, včetně objektově orientovaného programování, a přebírá prvky i z jiných programovacích paradigmat, včetně funkcionálního programování. Díky nízké úrovně funkčním umožňuje psát efektivní kód bez potřeby používat nebezpečné praktiky. (11)

Jazyk C# také těží z automatické správy paměti poskytované .NET Runtime a široké podpory ve vývojovém ekosystému. Tento jazyk je vhodný jak pro začátečníky, tak pro pokročilé vývojáře, kteří chtějí psát specializované aplikace. C# je navíc součástí C rodiny jazyků, což znamená, že má podobnou syntaxi jako C, C++, JavaScript a Java. (11)

3.2.3 XAML

XAML (Extensible Application Markup Language) je deklarativní značkovací jazyk používaný v rámci .NET platformy, zejména pro tvorbu uživatelských rozhraní (UI) v aplikacích pro Windows Presentation Foundation (WPF). Jeho hlavním přínosem je schopnost oddělit vizuální složku aplikace od její logiky, což usnadňuje spolupráci mezi designéry a vývojáři. XAML umožňuje definovat objekty a jejich vlastnosti prostřednictvím značkovacího jazyka, přičemž tyto objekty jsou automaticky vytvářeny při běhu aplikace. (12)

Jako textový formát je XAML kompatibilní s XML, což umožňuje snadnou integraci s dalšími technologiemi a nástroji. Díky své strukturovanosti a intuitivnímu formátu je XAML ideálním nástrojem pro vytváření složitých uživatelských rozhraní, aniž by bylo nutné psát rozsáhlý kód v tradičním programovacím jazyce. (12)

XAML podporuje práci s atributy, což umožňuje jednoduché nastavení vlastností objektů, ale také nabízí pokročilé možnosti pro složitější scénáře pomocí markup extensions a type

converters. Tyto rozšíření umožňují přizpůsobit způsob, jakým jsou hodnoty přidělovány vlastnostem, což zvyšuje flexibilitu při tvorbě UI. (12)

Kromě toho XAML umožňuje definování událostí a jejich propojení s kódem, což usnadňuje implementaci interaktivních prvků aplikace. Pro pokročilejší použití nabízí XAML také připojené vlastnosti a události, které umožňují nastavovat a spravovat hodnoty na prvcích, které dané vlastnosti nebo události běžně neobsahují. (12)

3.2.4 WPF

Windows Presentation Foundation (WPF) je UI framework, který je nezávislý na rozlišení a využívá vektorový vykreslovací engine, optimalizovaný pro moderní grafický hardware. WPF poskytuje širokou sadu funkcí pro vývoj aplikací, které zahrnují Extensible Application Markup Language (XAML), ovládací prvky, datové vazby, rozvržení, 2D a 3D grafiku, animace, styly, šablony, dokumenty, média, text a typografii. WPF je součástí .NET, což umožňuje vytvářet aplikace, které využívají i další prvky API platformy .NET. (13)

Existují dvě implementace WPF: verze pro .NET, která je open-source a hostována na GitHubu, a verze pro .NET Framework 4, která je podporována Visual Studiem. I když je .NET technologie multiplatformní, WPF běží pouze na Windows. (13)

WPF umožňuje vývoj aplikací pomocí jak značkovacího jazyka XAML, tak kódu v pozadí (code-behind). XAML se používá k definování vzhledu aplikací, zatímco kód v pozadí implementuje jejich chování. Tato separace umožňuje efektivnější vývoj, kdy designéři mohou pracovat na vzhledu, zatímco vývojáři se soustředí na funkčnost. WPF také poskytuje systém příkazů, který odděluje uživatelské akce od kódu, který na ně reaguje, což umožňuje konzistentní implementaci akcí, jako jsou například operace kopírování, vystřihování a vkládání. (13)

Další klíčové aspekty WPF zahrnují systém rozvržení, který umožňuje flexibilní přizpůsobení rozložení UI, a výkonný engine pro datové vazby, který zjednodušuje integraci uživatelského rozhraní s daty. WPF rovněž podporuje rozsáhlé grafické a animační možnosti, včetně 2D a 3D grafiky, a poskytuje různé nástroje pro přizpůsobení vzhledu aplikací, jako jsou šablony ovládacích prvků, datové šablony a styly. V případě potřeby lze v rámci WPF vytvořit i vlastní ovládací prvky, což poskytuje vysokou míru flexibility při tvorbě specifických uživatelských rozhraní. (13)

3.2.5 OxyPlot

OxyPlot je open source knihovna pro generování grafů, licencovaná pod MIT licenci, která je známá svou vysokou mírou permisivity a umožňuje použití i v proprietárním softwaru. Tato knihovna je založena na platformě .NET a podporuje více platform díky svému přenosnému jádru, což usnadňuje opětovné použití kódu pro vykreslování grafů na různých platformách. (14)

Knihovna byla vyvinuta s cílem poskytnout jednoduchý a rozšiřitelný model pro vykreslování grafů. Uživatelé mohou snadno měnit vzhled grafů pomocí jednoduchých

vlastností, a pro pokročilejší úpravy je možné vytvářet nové podtřídy s plnou kontrolou nad chováním a vykreslováním. (14)

OxyPlot nabízí podporu pro různé typy grafů, včetně XY grafů, kartézských grafů, polárních grafů a koláčových grafů. Kromě toho knihovna umožňuje práci s více osami, různými typy sérií dat a anotacemi, které lze dále rozšiřovat o vlastní typy. Výstupy lze exportovat do různých formátů, jako jsou PNG, SVG a PDF. (14)

Jako open source projekt je OxyPlot vyvíjen komunitou a je k dispozici na GitHubu. Dokumentace a příklady jsou zahrnuty v repozitáři, což umožňuje uživatelům snadný přístup k praktickým ukázkám a integraci knihovny do vlastních projektů. (14)

OxyPlot se primárně zaměřuje na dvourozměrné souřadnicové systémy, což se odráží i v jeho názvu, kde 'xy' naznačuje toto zaměření. Mezi omezení patří nutnost manuální aktualizace grafů při změně dat, absence podpory pro animace a gradientové či šrafované štetce. (14)

Vzhledem k MIT licenci mohou uživatelé s knihovnou zacházet prakticky bez omezení, což zahrnuje možnost úprav, distribuce a použití v jakémkoli projektu. Licence je umístěna v příloze této práce. (14)

3.2.6 Visual Studio

Visual Studio je výkonný nástroj pro vývojáře, který umožňuje kompletní cyklus vývoje softwaru na jednom místě. Jedná se o integrované vývojové prostředí (IDE), které zahrnuje nástroje pro psaní, úpravy, ladění a sestavování kódu, a také pro nasazení aplikací. Visual Studio poskytuje kompilátory, nástroje pro automatické doplňování kódu, správu verzí a mnoho dalších funkcí, které zlepšují každý krok vývojového procesu. (15)

Jednou z hlavních výhod Visual Studia je podpora široké škály programovacích jazyků a možnost vývoje aplikací pro různé platformy. Uživatelé mohou vytvářet a ladit aplikace v jazycích jako C++, C#, JavaScript, TypeScript, Python a další. Visual Studio také nabízí AI-asistované nástroje, jako jsou GitHub Copilot a IntelliCode, které pomáhají psát kód rychleji a přesněji. (15)

Díky integrovaným nástrojům pro ladění a testování kódu umožňuje Visual Studio vývojářům snadno diagnostikovat problémy a zlepšovat výkon aplikací. Testovací nástroje zahrnují možnosti jednotkového testování a analýzu pokrytí kódu, což přispívá k vytvoření vysoce kvalitního softwaru. (15)

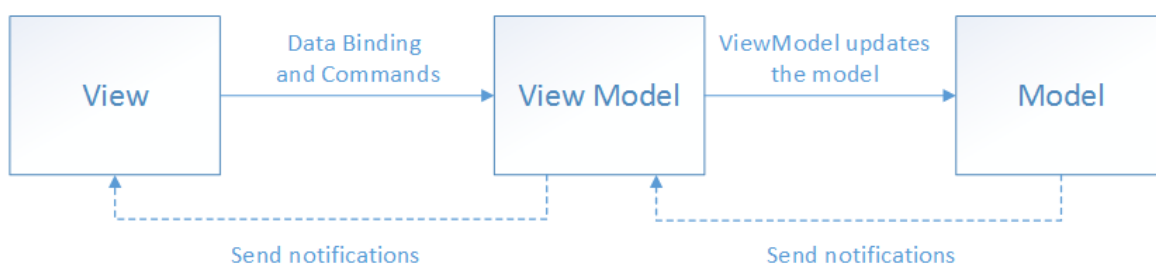
Visual Studio rovněž podporuje správu verzí pomocí Git, což umožňuje vývojářům spolupracovat na projektech a spravovat repozitáře přímo z IDE. Funkce jako Visual Studio Live Share umožňují real-time spolupráci, což zvyšuje efektivitu týmového vývoje. (15)

Existují tři edice Visual Studia: Community (bezplatná), Professional a Enterprise, které se liší funkcemi a jsou určeny pro různé typy uživatelů, od studentů po velké podniky. (15)

3.2.7 Architektura MVVM

Model-View-ViewModel (MVVM) je architektonický vzor, který slouží k oddělení obchodní logiky a prezentační logiky aplikace od jejího uživatelského rozhraní (UI). Tento vzor usnadňuje testování, údržbu a rozvoj aplikací a zároveň zvyšuje možnosti opětovného využití kódu. Díky MVVM mohou vývojáři a návrháři UI spolupracovat nezávisle na vývoji svých částí aplikace. (16)

MVVM se skládá ze tří hlavních komponent: modelu (Model), pohledu (View) a modelu pohledu (ViewModel). Model reprezentuje obchodní logiku a data aplikace, pohled zajišťuje strukturu a vzhled uživatelského rozhraní, a model pohledu slouží jako most mezi těmito dvěma komponentami, zajišťující logiku prezentace a stav. (16)



Obrázek 1 – Návrhový vzor MVVM, zdroj: (16)

Výhody použití MVVM zahrnují možnost provádění jednotkových testů na modelu a modelu pohledu bez nutnosti použití pohledu, možnost redesignu UI bez zásahu do modelu a modelu pohledu, a snadnější spolupráci mezi vývojáři a návrháři. MVVM také zajišťuje, že uživatelské rozhraní zůstane reaktivní díky použití asynchronních operací v modelu pohledu. (16)

Základem správného použití MVVM je pochopení, jak správně rozdělit kód aplikace do jednotlivých tříd a jak mají tyto třídy interagovat. Tento vzor je široce podporován v rámci .NET komunity a existuje mnoho frameworků, které usnadňují jeho implementaci, jako například .NET Community MVVM Toolkit, ReactiveUI nebo Prism Library. Tyto frameworky poskytují další funkcionality, jako jsou vlastní příkazy, navigační pomůcky nebo integrace s UI platformami. (16)

3.2.8 Math.NET Numerics

Math.NET Numerics představuje jednu z nejpoužívanějších knihoven pro numerické výpočty v oblasti vědy, techniky a každodenního použití. Nabízí širokou škálu metod a algoritmů, které zahrnují speciální funkce, lineární algebru, pravděpodobnostní modely, generování náhodných čísel, interpolace, integrace, regresní analýzy, optimalizační problémy a mnoho dalšího. (17)

Tato knihovna je součástí iniciativy Math.NET a vznikla sloučením dvou projektů, dnAnalytics a Math.NET Iridium, čímž nahradila oba tyto projekty. Knihovna je dostupná zdarma pod licencí MIT a podporuje různé platformy včetně Microsoft .NET 5.0, .NET 4.6.1

a vyšší, a .NET Standard 2.0 a vyšší. Vedle čistě spravované implementace nabízí také podporu pro nativní hardwarovou optimalizaci, což zvyšuje její výkon. (17)

Pro snadné začlenění do projektů jsou k dispozici balíčky NuGet, například tato knihovna jako jádrový balíček, nebo MathNet Numerics FSharp jako volitelné rozšíření pro F#. Vývojáři pracující v jazyce C# mohou tuto knihovnu jednoduše přidat do svého projektu pomocí balíčku NuGet. Knihovna je navržena tak, aby dobře spolupracovala s Visual Studiem a dalšími IDE. (17)

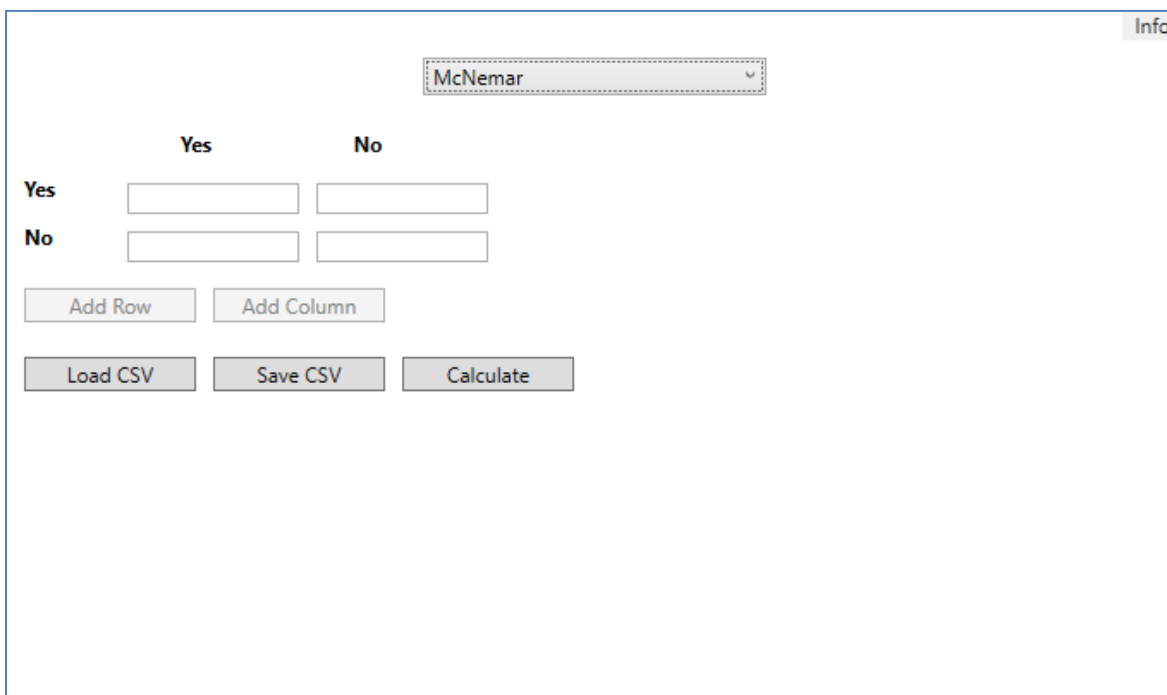
Knihovna je také kompatibilní s jazykem Visual Basic, což z ní činí univerzální nástroj pro numerické výpočty napříč různými .NET jazyky. Navíc je možné ji používat i na platformách mimo Windows, například v prostředí Linux s pomocí Mono. Díky své flexibilitě, výkonu a široké podpoře různých platforem a programovacích jazyků se knihovna stala nepostradatelným nástrojem pro vývojáře pracující s numerickými daty a algoritmy. (17)

3.3 Uživatelské rozhraní

Uživatelské rozhraní (UI) aplikace zajišťuje interakci mezi uživatelem a systémem. V této kapitole se zaměříme na popis hlavních komponent uživatelského rozhraní naší aplikace, které byly navrženy s důrazem na intuitivnost, přehlednost a efektivitu. Cílem bylo vytvořit rozhraní, které uživatelům umožní snadný a rychlý přístup ke všem funkcím aplikace, a to i bez předchozího technického zázemí. Následující části kapitoly se podrobně věnují jednotlivým prvkům rozhraní, jejich funkcionalitě, použitému designu a způsobu, jakým přispívají k celkovému uživatelskému zážitku. Aplikace je vyvinuta pomocí technologie WPF (Windows Presentation Foundation) a grafy, které jsou součástí aplikace, jsou vytvářeny pomocí knihovny OxyPlot.

3.3.1 Hlavní okno

Hlavní okno aplikace je klíčovým uživatelským rozhraním, kde se odehrává většina interakcí mezi uživatelem a aplikací. Toto okno bylo navrženo tak, aby uživatelům poskytlo co nejintuitivnější a nejefektivnější prostředí pro provádění různých statistických testů, práci s daty a zobrazení výsledků analýz.



Obrázek 2 – Hlavní okno aplikace

Jednou z prvních funkcí, kterou uživatel v hlavním okně nalezne, je rozbalovací seznam (ComboBox) umístěný na začátku aplikace. Tento seznam umožňuje uživateli vybrat jeden z dostupných statistických testů: Mann-Whitneyho test, McNemarův test, Cochranův Q test, Friedmanův test, Kruskal-Wallisův test, Spearmanův korelační koeficient, Kendallův korelační koeficient a faktorová analýza. Po výběru konkrétního testu se rozhraní aplikace dynamicky přizpůsobí tak, aby zobrazovalo pouze relevantní vstupy a možnosti pro daný test. Například výběrem McNemarova testu se zobrazí tabulka ve formátu 2x2, která je pro tento test charakteristická. Tento flexibilní přístup zajišťuje, že uživatelé vždy pracují s rozhraním, které odpovídá jejich aktuálním potřebám.

Pod výběrem testu se nachází tabulka, do které uživatelé mohou zadávat potřebná data. Tato tabulka je navržena s ohledem na variabilitu požadavků jednotlivých testů a dynamicky se mění podle vybraného testu. Pro testy, jako je McNemar, se zobrazí jednoduchá tabulka o dvou řádcích a dvou sloupcích, zatímco pro složitější testy, jako je Kruskal-Wallis nebo Friedmanův test, se zobrazí tabulky s více řádky a sloupci. Tato tabulka umožňuje přidávání nových řádků a sloupců pomocí tlačítek „Add Row“ a „Add Column“, což poskytuje uživatelům možnost přizpůsobit strukturu dat přesně podle svých potřeb. Například u testu Cochran's Q může uživatel snadno přidat další sloupce pro zadání nových proměnných. Popisky sloupců a řádků se dynamicky přizpůsobují vybranému testu, což zajišťuje, že uživatelé vždy vědí, kde a jaká data mají zadávat.

V dolní části hlavního okna se nachází tlačítka pro načítání a ukládání dat ve formátu CSV. Tlačítko „Load CSV“ umožňuje uživatelům rychle načíst data z externího souboru, což je užitečné zejména při práci s velkými datovými sadami nebo při opětovném použití dříve zadaných dat. Naopak tlačítko „Save CSV“ poskytuje možnost uložit aktuální data do

souboru, což umožňuje snadné zálohování nebo sdílení dat mezi různými uživateli nebo zařízeními. Tyto funkce zvyšují flexibilitu a efektivitu práce s aplikací, protože uživatelé nejsou omezeni pouze na ruční zadávání dat, ale mohou využívat i existující datové soubory.

Jednou z hlavních funkcí, které toto okno nabízí, je tlačítko „Calculate“. Toto tlačítko je umístěno vedle tlačítek pro načítání a ukládání dat a slouží k provedení zvoleného statistického testu na zadaných datech. Po stisknutí tlačítka aplikace provede potřebné výpočty a výsledky se okamžitě zobrazí v dolní části okna. U všech testů (mimo korelačních koeficientů) aplikace navíc otevře nové okno, ve kterém jsou zobrazeny výsledky a pomocné výpočty v přehledných grafech, včetně rozhodnutí o zamítnutí hypotézy. To uživatelům pomáhá s vizualizací výsledných hodnot.

Poslední komponentou v hlavním okně je v levém horním rohu tlačítko „Info“, které má za úkol zobrazit informace o licenci použitých knihoven. Po kliknutí na toto tlačítko se otevře nové okno, ve kterém je zobrazena MIT licence. Tato licence je zobrazena kvůli knihovnám OxyPlot a Math.NET Numerics, které jsou v aplikaci použity. Zobrazení licence v samostatném okně umožňuje uživatelům snadno přistupovat k právním informacím o použitých knihovnách bez zbytečného narušení jejich práce s aplikací. Tímto způsobem je zajištěna transparentnost a splnění licenčních požadavků.

This project includes software from the following open source projects, which are licensed under the MIT License:

Copyright (c) 2002 Math.NET
Copyright (c) 2014 OxyPlot contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Obrázek 3 – Okno s licenčními podmínkami

3.3.2 McNemarův test

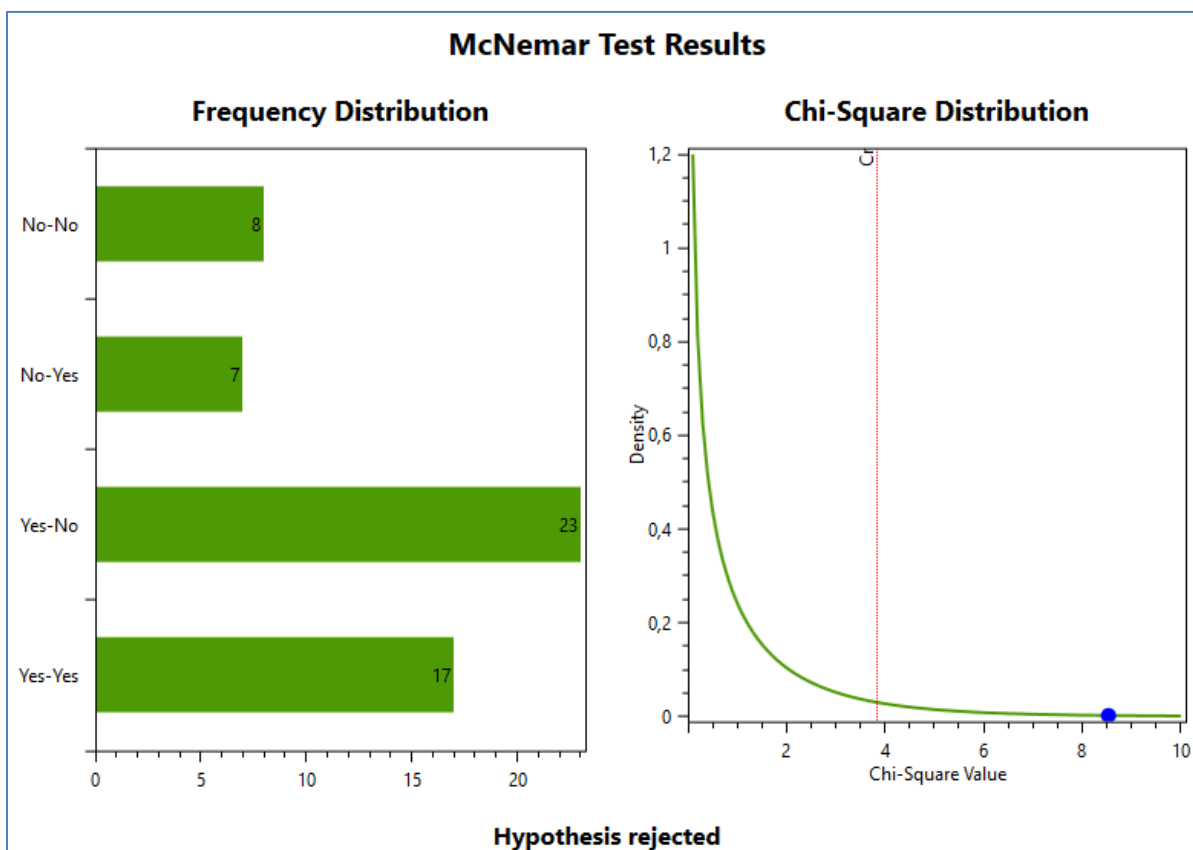
Po výběru McNemaraova testu z nabídky se automaticky upraví uživatelské rozhraní tak, aby odpovídalo požadavkům tohoto konkrétního testu. Tabulka, do které uživatelé zadávají data, se omezí na 2 řádky a 2 sloupce, což je formát nezbytný pro provedení McNemaraova testu. Nad sloupci a vedle řádků tabulky jsou umístěny popisky „Yes“ a „No“, které označují různé

odpovědi nebo kategorie pro analyzované proměnné. V tomto nastavení se také deaktivují tlačítka pro přidání nových řádků a sloupců, čímž se zamezí chybám při zadávání dat. Uživatelé mohou do této tabulky zadat čtyři hodnoty, které reprezentují četnost pro různé kombinace dvou kategorií.

Pod tabulkou je zobrazeno tlačítko „Calculate“, které po stisknutí provede McNemarův test na zadaných datech – provede výpočet χ^2 hodnoty. Pokud jsou frekvence příliš nízké, uživatel je upozorněn na nutnost zadat vyšší hodnoty. Výsledek testu je zobrazen v textovém bloku ve spodní části hlavního okna, kde je uvedeno, zda byla nulová hypotéza zamítnuta.

Po provedení výpočtu se automaticky otevře nové okno, kde jsou zobrazeny grafy a další vizualizace související s výsledky testu. Toto okno obsahuje dva hlavní grafy – jeden zobrazující χ^2 rozdělení s vyznačenou kritickou hodnotou a druhý zobrazující četnosti jednotlivých kategorií v podobě sloupcového grafu. První graf ilustruje pozici vypočtené χ^2 hodnoty v porovnání s kritickou hodnotou, což pomáhá uživateli vizualizovat rozhodovací proces týkající se zamítnutí nebo nezamítnutí nulové hypotézy.

Sloupcový graf na druhém místě znázorňuje četnosti různých kombinací kategorií, což umožňuje rychlé porovnání počtu případů v jednotlivých skupinách. Pod grafy je text, který oznamuje rozhodnutí o zamítnutí nebo nezamítnutí nulové hypotézy. Implementace tohoto okna v rámci aplikace zajišťuje, že uživatelé mají přístup k důležitým informacím potřebným k interpretaci výsledků McNemarova testu.



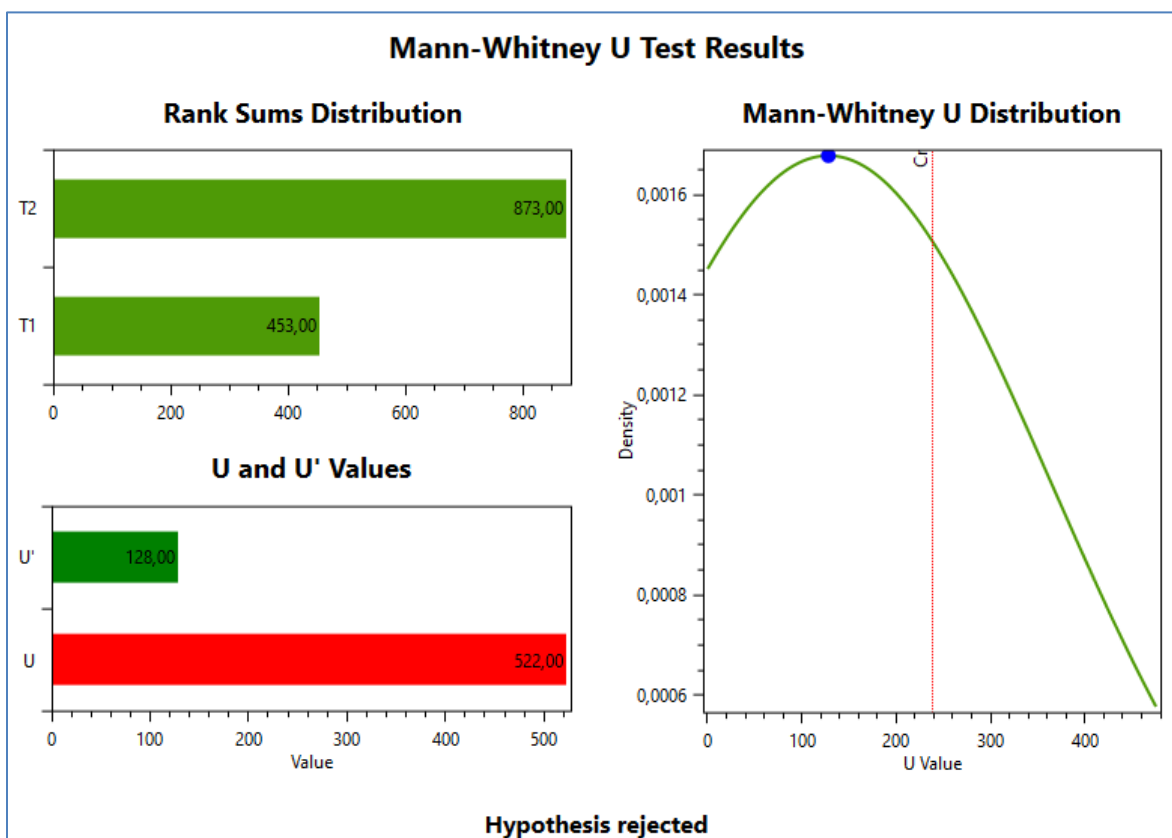
Obrázek 4 – Okno s grafy McNemarova testu

3.3.3 Mann-Whitneyův test

Při výběru Mann-Whitneyho testu se uživatelské rozhraní dynamicky přizpůsobuje podle potřeb uživatele, kdy se zobrazí tabulka o třech řádcích a dvou sloupcích. Popisky sloupců jsou nastaveny na "Sample 1" a "Sample 2", zatímco popisky řádků zůstávají nepojmenované. Uživatelé nemohou přidávat další sloupce, ale mohou přidávat nové řádky pro vložení dat. Tlačítko pro přidání nového sloupce je v tomto případě deaktivováno, což zajišťuje, že struktura tabulky zůstane vhodná pro tento typ testu.

Po zadání dat a kliknutí na tlačítko „Calculate“ se otevře nové okno, které obsahuje tři klíčové grafy, které pomáhají uživateli lépe pochopit výsledky testu. Prvním zobrazeným grafem je sloupcový graf, který vizualizuje součty ranků pro oba porovnávané vzorky ($T1$ a $T2$). Tento graf umožňuje uživateli snadno porovnat, jak si oba vzorky vedly z hlediska jejich ranků. Druhým grafem je sloupcový graf zobrazující hodnoty U a U' , které jsou klíčové pro interpretaci výsledků testu. Tento graf jasně ukazuje, která z těchto hodnot je vybrána jako U , což je důležité pro pochopení toho, která hodnota rozhoduje o zamítnutí či přijetí nulové hypotézy. Třetí graf je spojnicový graf, který zobrazuje rozložení hodnoty U v kontextu jejího normálního rozdělení. Tento graf poskytuje vizuální přehled o tom, kde se nachází hodnota U ve vztahu ke kritické hodnotě.

Ve spodní části okna je zobrazen textový blok s výsledkem hypotézy. Textový blok poskytuje jednoduchý a přehledný výstup, kde se jasně uvádí, zda byla nulová hypotéza zamítnuta, či nikoliv.



Obrázek 5 – Okno s grafy Mann-Whitneyho testu

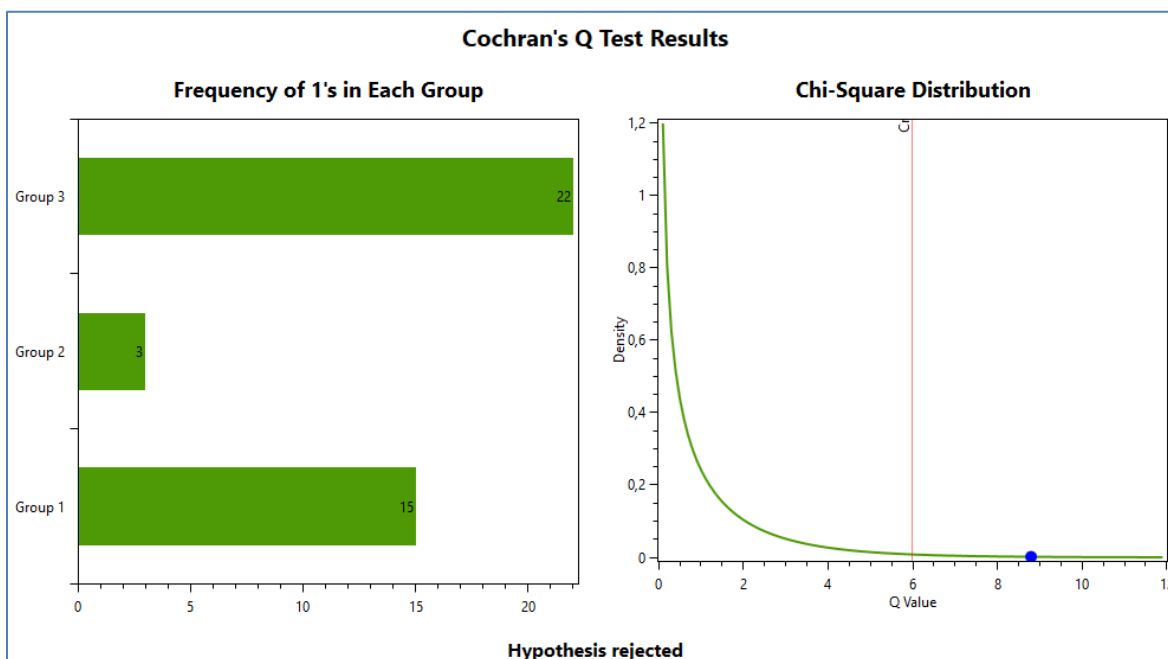
3.3.4 Cochranův test

Uživateli je zobrazena při zvolení Cochranova testu tabulka, která je specificky navržena pro zadávání dichotomických dat (tj. hodnoty 0 a 1). V případě, že je zadána jiná hodnota se vypíše upozornění na nesprávná data. Tato tabulka je automaticky nastavena na rozměr 3x3, což je minimální požadavek pro tento typ testu, avšak uživatelé mohou dle potřeby přidávat další řádky a sloupce.

Sloupce jsou označeny jako "Treatment 1", "Treatment 2", "Treatment 3" atd., a jsou dynamicky generovány na základě počtu sloupců. Tyto popisky pomáhají uživateli orientovat se v tom, jaká data má do jednotlivých buněk tabulky vložit. Řádky jsou označeny jako "Block 1", "Block 2", "Block 3" atd., a každý z těchto bloků reprezentuje jednu jednotku (nebo skupinu) v experimentu, kde byly provedeny různé léčby nebo intervence.

Po zadání dat a kliknutí na tlačítko „Calculate“ se otevře nové okno, ve kterém jsou zobrazeny 2 grafy a textový blok s rozhodnutím o přijetí nebo zamítnutí hypotézy. Prvním grafem je sloupcový graf, který zobrazuje frekvenci výskytu hodnoty "1" v jednotlivých skupinách. Každý sloupec představuje jednu skupinu, a díky označení kategorií ("Group 1", "Group 2" atd.) je uživateli poskytnut intuitivní přehled o rozložení hodnot v datech.

Druhým grafem je spojnicový graf, který zobrazuje rozdělení hodnoty Q ve vztahu k její kritické hodnotě na základě χ^2 rozdělení. Tento graf pomáhá uživatelům vizuálně pochopit, jak se vypočítaná hodnota Q vztahuje k teoretickému rozdělení, a kde se nachází ve vztahu ke kritické hodnotě. Důležité body, jako je samotná hodnota Q a kritická hodnota, jsou v grafu zřetelně zvýrazněny, což usnadňuje interpretaci, zda byla nulová hypotéza zamítnuta či nikoliv.



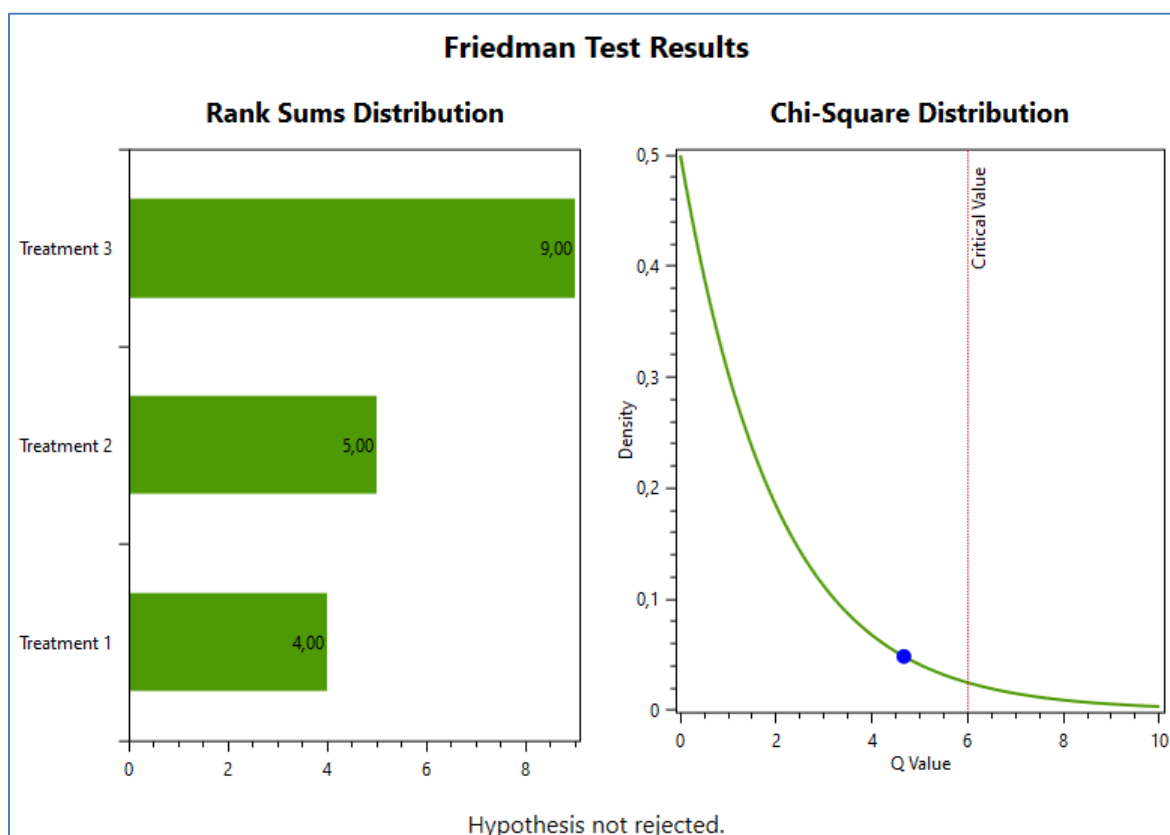
Obrázek 6 – Okno s grafy Cochranova testu

3.3.5 Friedmanův test

Při použití Friedmanova testu v aplikaci uživatel zadává data do tabulky, která je automaticky přizpůsobena tak, aby odpovídala požadavkům tohoto testu. Tabulka je přednastavena na rozměr 3x3. Popisky sloupců jsou automaticky generovány a označeny jako "Treatment 1", "Treatment 2", "Treatment 3" atd. Tyto popisky odkazují na různé experimentální podmínky nebo léčby, které byly použity během experimentu. Popisky řádků jsou označeny jako "Block 1", "Block 2", "Block 3" atd. Každý řádek představuje jeden blok v experimentu, což může být například skupina subjektů nebo jednotlivá pozorování, kde byly všechny léčby aplikovány.

Když jsou všechna data správně zadána, uživatel může spustit výpočet pomocí tlačítka "Calculate". Po provedení výpočtu se zobrazí výsledky testu v novém okně, kde jsou prezentovány pomocí dvou grafů. Prvním grafem je sloupcový graf zobrazující rozložení součtů hodnot (Rank Sums) pro jednotlivé léčby. Druhým grafem je spojnicový graf χ^2 rozdělení, na kterém je vykreslena hodnota testovací statistiky Q a kritická hodnota. Červená svislá čára označuje kritickou hodnotu, která slouží jako hranice pro rozhodování o zamítnutí nebo nezamítnutí nulové hypotézy.

Na spodní části okna je zobrazen textový blok, který informuje uživatele o tom, zda byla hypotéza zamítnuta nebo ne, což představuje konečný výsledek statistického testu.



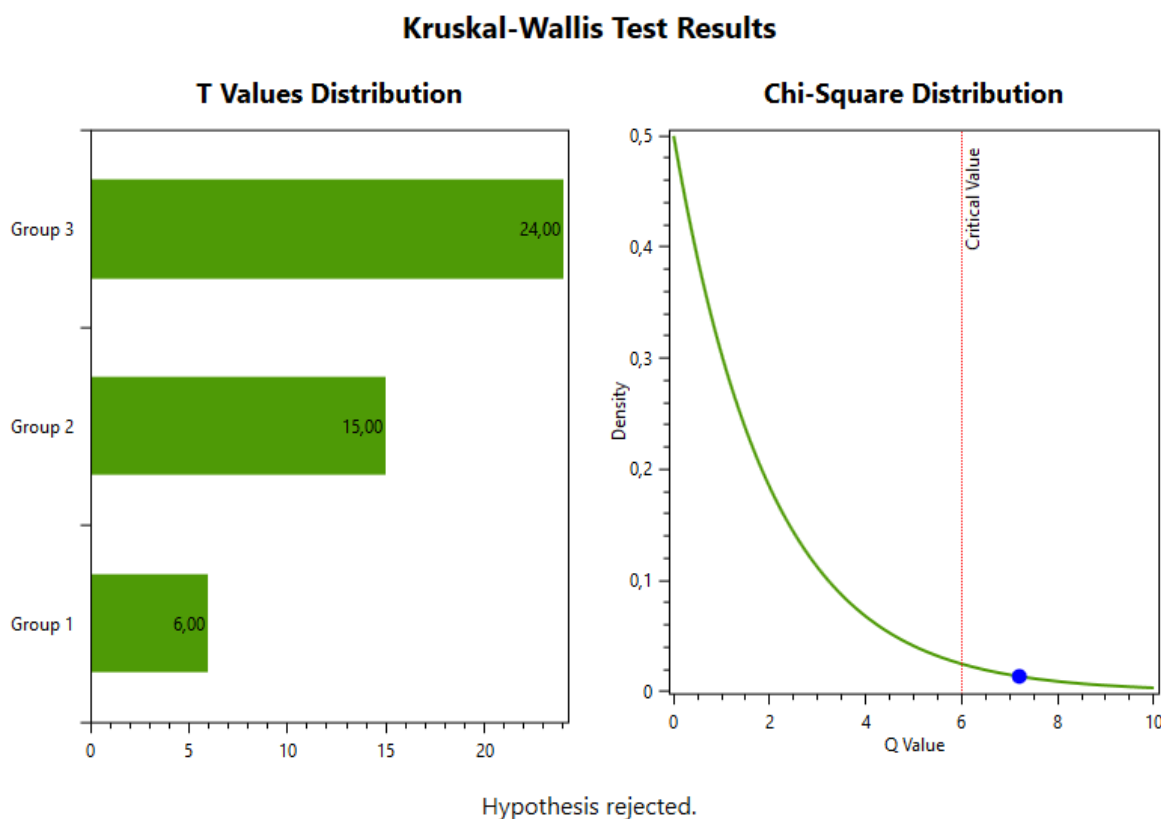
Obrázek 7 – Okno s grafy Friedmanova testu

3.3.6 Kruskal-Wallisův test

Při použití Kruskal-Wallisova testu v aplikaci je uživateli k dispozici tabulka, která se automaticky přizpůsobí požadavkům tohoto testu. Po zvolení tohoto testu je tabulka přednastavena na velikost 3x3, ale uživatel může přidat další řádky a sloupce podle potřeby. Popisky sloupců jsou označeny jako "Group 1", "Group 2", "Group 3" atd., kde každá skupina představuje různé kategorie nebo skupiny pozorování, které jsou porovnávány mezi sebou. Popisky řádků jsou označeny jako "Observation 1", "Observation 2", "Observation 3" atd., přičemž každý řádek představuje jednotlivé pozorování v rámci dané skupiny.

Když jsou všechna data správně zadána, uživatel může spustit výpočet pomocí tlačítka "Calculate". Po provedení výpočtu se výsledky testu zobrazí v novém okně, kde jsou prezentovány dvěma grafy. Prvním grafem je sloupcový graf, který zobrazuje hodnoty T pro každou skupinu. Druhým grafem je spojnicový graf χ^2 rozdělení, na kterém je zobrazena hodnota testovací statistiky Q a kritická hodnota. Tento graf umožňuje uživateli snadno určit, zda lze nulovou hypotézu zamítnout či nikoliv, přičemž kritická hodnota je označena červenou svislou čarou.

Na spodní části okna se nachází textový blok, který zobrazuje výsledek testu, konkrétně zda byla hypotéza zamítnuta nebo ne, čímž uživateli poskytuje jasný závěr provedené analýzy.



Obrázek 8 – Okno s grafy Kruskal-Wallisova testu

3.3.7 Korelační koeficienty

Při použití Spearmanova a Kendallova korelačního koeficientu v aplikaci zadává uživatel data do tabulky, která je přizpůsobena tak, aby odpovídala požadavkům těchto testů. Tabulka je přednastavena na rozměr 5x2. Popisky sloupců jsou označeny jako "X" a "Y", kde každý sloupec představuje jednu z proměnných, které se porovnávají mezi sebou. Popisky řádků nejsou konkrétně pojmenovány, protože každý řádek představuje jedno pozorování dvojice hodnot (X, Y).

Po zadání všech potřebných dat může uživatel spustit výpočet pomocí tlačítka "Calculate". Na rozdíl od jiných statistických testů v aplikaci, výsledky Spearmanova a Kendallova korelačního koeficientu nejsou zobrazeny v samostatném okně, protože vizualizace výsledků není v tomto případě nutná. Místo toho se výsledek zobrazí přímo v hlavním okně aplikace. Výsledek testu zahrnuje vypočtenou hodnotu korelačního koeficientu (ρ pro Spearmanův test a τ pro Kendallův test) a kritickou hodnotu.

Aplikace také informuje uživatele, zda byla nulová hypotéza zamítnuta nebo ne, což představuje konečný výsledek statistického testu. Tento výsledek je zobrazen v textovém bloku přímo pod tabulkou, což uživateli umožňuje rychle a snadno interpretovat výsledky bez nutnosti přecházet do jiného okna.

3.3.8 Faktorová analýza

Při použití faktorové analýzy v aplikaci je uživateli k dispozici tabulka, která se automaticky přizpůsobí požadavkům této analýzy. Po zvolení této metody je tabulka přednastavena na velikost 3x3, kde uživatel může zadat potřebná data. Řádky i sloupce může uživatel přidávat. Popisky řádků jsou označeny jako "Variable 1", "Variable 2", "Variable 3" atd., což odpovídá různým proměnným v analýze. Popisky sloupců jsou označeny jako "Observation 1", "Observation 2" atd., kde každá "Observation" představuje různá pozorování pro jednotlivé proměnné.

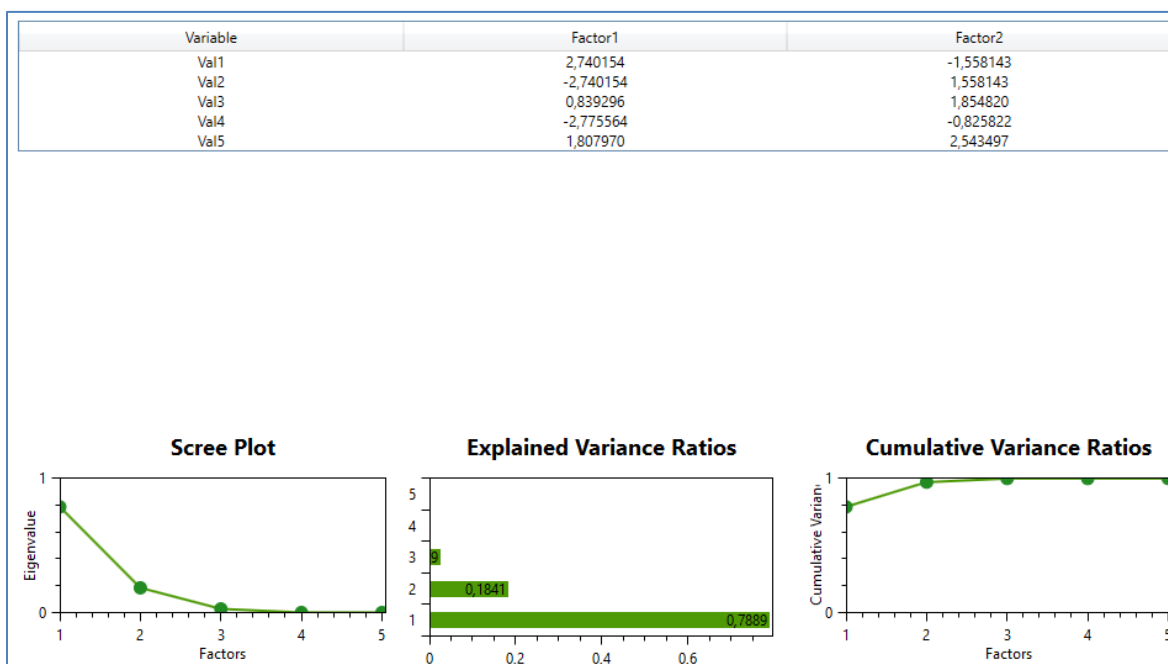
Když jsou všechna data správně zadána, uživatel může spustit výpočet pomocí tlačítka "Calculate". Po provedení výpočtu se výsledky faktorové analýzy zobrazí v novém okně, které poskytuje podrobný přehled výsledků analýzy.

První část okna obsahuje tabulku faktorových zátěží (Factor Loadings). Tato tabulka zobrazuje, jak jednotlivé proměnné přispívají k různým faktorům. Řádky tabulky jsou označeny jako "Variable 1", "Variable 2" atd., a sloupce jako "Factor 1", "Factor 2" atd. Tabulka poskytuje přehled o tom, jak významně každá proměnná přispívá k identifikovaným faktorům.

Další částí okna je Scree Plot, což je spojnicový graf, který zobrazuje vlastní čísla (eigenvalues) jednotlivých faktorů. Tento graf pomáhá určit, kolik faktorů by mělo být zahrnuto v analýze. V grafu je vizuálně zřetelný bod, ve kterém dochází k významnému poklesu eigenvalues, což indikuje optimální počet faktorů pro analýzu.

Následuje graf vysvětleného rozptylu (Explained Variance Ratios), který je zobrazen jako sloupcový graf. Tento graf ukazuje procentuální podíl rozptylu, který je vysvětlen jednotlivými faktory. Sloupce představují jednotlivé faktory a délka každého sloupce odpovídá množství rozptylu, který daný faktor vysvětluje.

Posledním grafem je graf kumulativního rozptylu (Cumulative Variance Ratios), který je opět zobrazen jako spojnicový graf. Tento graf ukazuje kumulativní podíl rozptylu, který zahrnuje více faktorů. Jak se zvyšuje počet faktorů, které jsou zahrnuty do analýzy, křivka v grafu ukazuje, jak roste celkové procento vysvětleného rozptylu.



Obrázek 9 – Okno s grafy faktorové analýzy

3.4 Implementace psychometrických testů

V této kapitole se zaměříme na implementaci psychometrických testů v rámci vyvíjené aplikace. Aplikace integruje několik vybraných psychometrických metod, které uživatelům umožňují analyzovat data a získávat relevantní statistické výsledky. V následujících částech této kapitoly bude podrobně pro každý test popsán způsob implementace v kódu a jakým způsobem byly jednotlivé metody integrovány do uživatelského rozhraní aplikace. Nejprve si zde však popíšeme způsoby výpočtu kritických hodnot, které jsou nezbytné pro rozhodnutí, zda zamítnout nebo nezamítnout nulovou hypotézu. Cílem je poskytnout komplexní přehled o tom, jakým způsobem byly tyto testy navrženy a implementovány, aby co nejlépe sloužily potřebám uživatelů. Veškeré neparametrické testy jsou implementovány jako jednostranné s hladinou významnosti 5 %.

3.4.1 Výpočet kritické hodnoty pro χ^2 rozdělení

Implementace výpočtu kritické hodnoty pro χ^2 rozdělení je rozdělena do 4 metod ve třídě `StatisticsHelper`. Metoda `double ChiSquareCriticalValue(int df, double`

alpha) slouží k nalezení kritické hodnoty χ^2 pro zadaný počet stupňů volnosti (`int df`) a hladinu významnosti (`double alpha`). K výpočtu se používá numerická metoda bisekce (dichotomického dělení), která hledá kritickou hodnotu iterativně. Tato metoda začíná s počátečními hodnotami intervalu `double low = 0.0` a `double high = 100.0`. Během iterací se opakovaně vypočítává střední hodnota tohoto intervalu (`double mid`) a pomocí metody `double ChiSquareCDF(double x, int df)` se vypočítává kumulativní distribuční funkce (CDF) pro tuto hodnotu. Pokud je hodnota CDF menší než $1 - \alpha$, střední hodnota se stane novým dolním limitem intervalu, jinak se stane novým horním limitem. Tento proces pokračuje, dokud rozdíl mezi horním a dolním limitem nepřesáhne stanovenou toleranci ($1e-10$), a výsledná hodnota střední hodnoty je vrácena jako odhad kritické hodnoty.

Metoda `ChiSquareCDF` slouží k výpočtu kumulativní distribuční funkce pro χ^2 rozdělení s daným počtem stupňů volnosti. Tato metoda převádí problém výpočtu CDF χ^2 rozdělení na výpočet CDF gama rozdělení, což je jednodušší úloha. K tomu využívá metodu `double GammaCDF(double x, double k)`, která je volána s parametry $x / 2.0$ a $df / 2.0$, což odpovídá parametrům gama rozdělení. Tímto způsobem je výpočet kumulativní distribuční funkce χ^2 rozdělení efektivně transformován na problém výpočtu kumulativní distribuční funkce gama rozdělení.

Metoda `GammaCDF` je zodpovědná za samotný výpočet kumulativní distribuční funkce pro gama rozdělení. Tento výpočet probíhá pomocí sumace nekonečné řady, kde každý člen řady je vypočítán pomocí rekurzivního vztahu. V počátečním kroku se nastaví počáteční hodnota součtu (`double sum`) a prvního členu řady (`double term`) na $1.0 / k$. Následně se v iterativním procesu počítají další členy řady, které jsou postupně přičítány k hodnotě `sum`. Tento proces pokračuje, dokud nedojde ke konvergenci, což znamená, že přidávání dalších členů řady již neovlivňuje konečný výsledek v rámci stanovené přesnosti. Po dokončení sumace se celý výsledek vynásobí exponenciální funkcí, která obsahuje logaritmické výrazy. Tyto logaritmické výrazy jsou počítány pomocí metody `double GammaLog(double x)`, což zajišťuje přesnost výpočtu.

Metoda `GammaLog` slouží k výpočtu přirozeného logaritmu gama funkce ($\Gamma(x)$), což je speciální funkce používaná v mnoha oblastech matematiky, včetně statistiky. Tato metoda využívá Lanczosovu aproximaci, což je numerická metoda, která poskytuje přesnou aproximaci logaritmu gama funkce pro různé hodnoty `double x`. Při výpočtu se používá několik koeficientů, které jsou aplikovány v rámci sumace a poté kombinovány s logaritmickými operacemi, což vede k výsledné hodnotě logaritmu gama funkce.

3.4.2 Výpočet hodnot pomocí inverzní funkce normálního rozdělení

Postup výpočtu inverzní funkce k normálnímu rozdělení pomocí zadaného kódu začíná metodou `double NormalInverseCDF(double low, double high, double p)`, která má za cíl nalézt hodnotu, která odpovídá zadané pravděpodobnosti `double p`. Tato metoda využívá bisekční metodu k iterativnímu hledání správné hodnoty v daném intervalu mezi `double low` a `double high`.

Proces začíná výpočtem střední hodnoty (`double mid`) intervalu, která se vypočítá jako průměr hodnot `low` a `high`. Následně se opakovaně provádí kontrola, zda kumulativní distribuční funkce normálního rozdělení (`NormalCDF`) pro tuto střední hodnotu dává výsledek menší než zadaná pravděpodobnost `p`. Pokud je výsledek menší, aktualizuje se dolní hranice intervalu (`low`) na hodnotu `mid`. V opačném případě se aktualizuje horní hranice intervalu (`high`) na hodnotu `mid`. Tento postup se opakuje, dokud rozdíl mezi horní a dolní hranicí intervalu není dostatečně malý (méně než zadaná tolerance $1e-6$).

Pro výpočet kumulativní distribuční funkce normálního rozdělení (`NormalCDF`) se používá metoda `double NormalCDF(double x)`, která implementuje výpočet této funkce. Tato metoda se opírá o chybovou funkci `Erf`, která je klíčová pro aproximaci hodnot kumulativní distribuční funkce normálního rozdělení.

Metoda `double Erf(double x)` slouží k výpočtu chybové funkce, která je nezbytnou součástí výpočtu kumulativní distribuční funkce. V kódu je chybová funkce implementována pomocí dobře známé aproximace, která využívá sadu koeficientů (`double a1`, `double a2`, `double a3`, `double a4`, `double a5`) a proměnné `double p`, aby přesněji approximovala chybovou funkci. Vstupní hodnota `double x` je nejprve upravena na absolutní hodnotu a výpočet pokračuje aplikací polynomické aproximace. Výsledná hodnota `double y` je následně vynásobena znaménkem původní hodnoty `x`, aby se zajistilo, že chybová funkce správně vrátí kladnou nebo zápornou hodnotu podle vstupu.

Celkově tedy proces začíná výpočtem kumulativní distribuční funkce normálního rozdělení pomocí `NormalCDF`, která volá chybovou funkci `Erf` pro aproximaci. Poté `NormalInverseCDF` iterativně hledá hodnotu, která odpovídá zadané pravděpodobnosti `p`, pomocí bisekční metody. Tímto způsobem se vypočítá hledaná kvantilová hodnota normálního rozdělení, což je důležité při výpočtu kritických hodnot v statistických testech.

3.4.3 Implementace McNemarova testu

Implementace McNemarova testu v aplikaci je zajištěna pomocí metody `double McNemarTest(int[,] table)`, která je součástí třídy `StatisticsModel`. Tato metoda přijímá jako vstup dvourozměrné pole `int[,] table`, které reprezentuje 2x2 kontingenční tabulku, v níž jsou uloženy četnosti jednotlivých kategorií (n_{11} , n_{12} , n_{21} , n_{22}).

Prvním krokem v metodě `McNemarTest` je rozdělení hodnot z tabulky do jednotlivých proměnných:

- `int a` reprezentuje počet subjektů s pozitivním výsledkem v obou podmínkách (n_{11}),
- `int b` reprezentuje počet subjektů s pozitivním výsledkem v první podmínce a negativním výsledkem ve druhé podmínce (n_{12}),
- `int c` reprezentuje počet subjektů s negativním výsledkem v první podmínce a pozitivním výsledkem ve druhé podmínce (n_{21}),
- `int d` reprezentuje počet subjektů s negativním výsledkem v obou podmínkách (n_{22}).

Následuje kontrola, zda jsou četnosti dostatečné pro provedení testu. Pokud součet a a d (n_{11} a n_{22}) je menší než 5, metoda vyvolá výjimku s informací, že četnosti jsou příliš malé pro spolehlivé provedení testu.

Pokud jsou četnosti vhodné, metoda pokračuje samotným výpočtem statistiky McNemarova testu. Pokud je součet hodnot b a c (n_{12} a n_{21}) menší než 10, použije se vzorec s Yatesovou korekcí, který zohledňuje malé četnosti. Jinak se použije standardní vzorec pro výpočet χ^2 , který porovnává rozdíl mezi hodnotami b a c s jejich součtem.

V rámci metody `void Calculate()` je poté tabulka s daty získána a zkontrolována, zda má správný rozměr 2x2. Po úspěšné kontrole se zavolá metoda `double McNemarTest(int[,] table)`, která vrátí hodnotu statistiky χ^2 . Následně se vypočítá kritická hodnota χ^2 pro jeden stupeň volnosti pomocí metody `v` v pomocné třídě `StatisticsHelper` a rozhodne se, zda zamítnout nulovou hypotézu na základě porovnání statistiky χ^2 a kritické hodnoty.

Výsledky testu jsou poté zobrazeny uživateli formou textového výstupu, který zahrnuje vypočítanou hodnotu χ^2 , kritickou hodnotu a rozhodnutí o zamítnutí nebo nezamítnutí nulové hypotézy. Pro lepší vizualizaci výsledků je otevřeno nové okno `McNemarTestView`, kde jsou data a výsledky testu zobrazeny graficky.

Metoda pro výpočet McNemarova testu pak vypadá takto:

```
public double McNemarTest(int[, ] table)
{
    int a = table[0, 0];

    int b = table[0, 1];
    int c = table[1, 0];
    int d = table[1, 1];

    if ((a + d) / 2 < 5)
    {
        throw new ArgumentOutOfRangeException("Četnosti jsou příliš malé");
    }

    if (b + c < 10)
    {
        return Math.Pow(Math.Abs(b - c) - 1, 2) / (b + c); // Yatesova
korekce
    }
    else
    {
        return Math.Pow(b - c, 2) / (b + c); // McNemarův test
    }
}
```

3.4.4 Implementace Mann-Whitneyho U-testu

V implementaci Mann-Whitneyho U-testu v aplikaci se nejprve zajišťuje správná struktura a velikost dat. Po zvolení tohoto testu jsou data načtena z tabulky, která musí obsahovat právě dva sloupce `double[] sample1` a `double[] sample2`, z nichž každý reprezentuje jeden z analyzovaných vzorků. Pokud některý ze vzorků obsahuje méně než tři hodnoty, je uživatel upozorněn na potřebu většího vzorku pro provedení testu, a výpočet není proveden.

Hlavní část výpočtu testu probíhá v metodě `double MannWhitneyTest(double[] sample1, double[] sample2, out double U1, out double U2, out double T1, out double T2)`, kde jsou data z obou vzorků nejprve sloučena do jednoho pole `double[] allSamples` a poté seřazena podle velikosti. Pro zachování správnosti výpočtů se v případě shodných hodnot použijí průměrné pořadové hodnoty. Poté se sečtou pořadové hodnoty pro jednotlivé vzorky, což vede k získání součtů pořadí označovaných jako `double T1` pro první vzorek a `double T2` pro druhý vzorek.

Na základě těchto součtů jsou vypočítány hodnoty `double U1` a `double U2`, které reprezentují počet inverzí potřebných k tomu, aby se prvky jednoho vzorku dostaly před prvky druhého vzorku v seřazeném pořadí. Menší z těchto dvou hodnot se pak použije jako výsledná hodnota testové statistiky `double U`.

Implementace získání kritické hodnoty v Mann-Whitneyho testu probíhá dvěma způsoby, v závislosti na velikosti vzorků. Pro menší vzorky, konkrétně pokud velikost obou vzorků je menší nebo rovna 20, se použijí předem definované tabulkové hodnoty, které poskytují kritické hodnoty přímo na základě velikosti vzorků. Tento přístup je vhodný pro menší vzorky, kde jsou tabulky spolehlivým zdrojem kritických hodnot.

Pro větší vzorky, kde jeden ze vzorků je větší než 20, se k určení kritické hodnoty používá normální aproximace. V tomto případě se nejprve vypočítá průměrná hodnota U (`double meanU`) a směrodatná odchylka U (`double sdU`). Kritická hodnota je pak určena pomocí z -skóre, které se vypočítá jako inverzní hodnota k normálnímu rozdělení. Tento výpočet se provede pomocí funkce `double NormalInverseCDF(double low, double high, double p)`, která určí hodnotu z -skóre pro jednostranný test na hladině významnosti 0,05. Kritická hodnota se pak získá jako rozdíl průměrné hodnoty U a produktu z -skóre a směrodatné odchylky.

Po výpočtu se výsledek testu zobrazí včetně všech důležitých hodnot, jako jsou U , $U1$, $U2$ a kritická hodnota. Pokud je nulová hypotéza zamítnuta, uživatel je o tom informován. Kromě toho je také otevřeno nové okno, kde jsou výsledky graficky vizualizovány.

Metoda pro výpočet Mann-Whitneyho U-testu testu pak vypadá takto:

```
public double MannWhitneyTest(double[] sample1, double[] sample2, out double
U1, out double U2, out double T1, out double T2)
{
    int n1 = sample1.Length;
    int n2 = sample2.Length;

    double[] allSamples = new double[n1 + n2];
    Array.Copy(sample1, allSamples, n1);
    Array.Copy(sample2, 0, allSamples, n1, n2);

    // Rangování s průměrnými hodnotami pro duplikáty
    var ranks = allSamples.Select((value, index) => new { Value = value,
Index = index })
        .OrderBy(x => x.Value)
        .Select((x, i) => new { x.Index, Rank = i + 1 })
        .OrderBy(x => x.Index)
        .Select(x => x.Rank).ToArray();

    T1 = ranks.Take(n1).Sum();
    T2 = ranks.Skip(n1).Take(n2).Sum();

    // Výpočet U a U'
    U1 = n1 * n2 + (n1 * (n1 + 1)) / 2.0 - T1;
    U2 = n1 * n2 + (n2 * (n2 + 1)) / 2.0 - T2;

    double U = Math.Min(U1, U2);
    return U;
}
```

3.4.5 Implementace Cochranova Q-testu

Implementace Cochranova Q-testu v zadaném kódu začíná získáním dat z tabulky, která je součástí uživatelského rozhraní aplikace. Data jsou organizována ve dvourozměrném poli `int[][] cochranData`, kde každý prvek reprezentuje buď 1 pro úspěch, nebo 0 pro neúspěch. Kód následně provede kontrolu, zda jsou všechna vstupní data binární, tedy že obsahují pouze hodnoty 0 nebo 1. Pokud data neodpovídají těmto požadavkům, uživateli je vrácena zpráva, že data musí být binární, a výpočet se zastaví.

Následně se provádí samotný výpočet testové statistiky `double Q`. Nejprve se vypočítají součty úspěchů (binárních hodnot 1) pro každý sloupec (`int[] Tj`) a řádek (`int[] ui`) v datové tabulce. Tyto hodnoty jsou klíčové pro další výpočet, protože určují, jak se jednotlivé skupiny nebo podmínky liší v počtu úspěchů.

Poté se na základě vypočítaných součtů T_j a u_i spočítá hodnota testového kritéria Q . Tento výpočet zahrnuje součet čtverců hodnot T_j a jejich celkový součet, dále také součet čtverců hodnot u_i . Tato statistika Q je vypočtena podle specifikovaného vzorce a slouží k porovnání

s kritickou hodnotou, aby bylo možné rozhodnout, zda zamítnout nebo nezamítnout nulovou hypotézu.

Kritická hodnota není získána z tabulky, ale je vypočítána na základě χ^2 rozdělení s použitím zadané hladiny významnosti a stupně volnosti ($k - 1$). K tomu se používá metoda `double ChiSquareCriticalValue(int df, double alpha)`, která v kódu provádí výpočet této hodnoty pro zadaný počet stupňů volnosti a hladinu významnosti.

Poté se hodnota Q porovná s touto vypočítanou kritickou hodnotou. Pokud je hodnota Q větší než kritická hodnota, nulová hypotéza se zamítá. Výsledek tohoto porovnání, spolu s hodnotou Q a kritické hodnoty, je zobrazen uživateli v podobě textové zprávy, která jasně informuje o tom, zda byla hypotéza zamítnuta či nikoliv. Nakonec je otevřeno nové okno, které zobrazí výsledky testu graficky, což uživateli poskytuje vizuální interpretaci statistických výpočtů.

Metoda pro výpočet Cochranova Q-testu pak vypadá takto:

```
public double CochranQTest(int[][] data)
{
    int n = data.Length; // Počet řádků (skupin)
    int k = data[0].Length; // Počet sloupců (léčeb)

    // Výpočet  $T_{.j}$  a  $u_{i.}$ 
    int[] Tj = new int[k];
    int[] ui = new int[n];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < k; j++)
        {
            Tj[j] += data[i][j];
            ui[i] += data[i][j];
        }
    }

    // Výpočet  $Q$ 
    int sumTjSquared = Tj.Sum(t => t * t);
    int sumTj = Tj.Sum();
    int sumUiSquared = ui.Sum(u => u * u);
    int sumUi = ui.Sum();

    double Q = (k - 1.0) * (sumTjSquared - Math.Pow(sumTj, 2) / k) / (k *
sumUi - sumUiSquared);

    return Q;
}
```

3.4.6 Implementace Friedmanova testu

Implementace Friedmanova testu v daném kódu začíná načtením dat z tabulky, kde jsou jednotlivé hodnoty uspořádány v matici typu `double[][]` s n řádky a k sloupci. Každý řádek této matice představuje jeden blok (například subjekt nebo experiment), zatímco každý sloupec reprezentuje různé podmínky nebo ošetření, která jsou aplikována na bloky. Uživatel zadává data do tabulky, a pokud je počet řádků nebo sloupců menší než 3, což je minimální požadavek pro provedení Friedmanova testu, výpočet je přerušen a uživateli je zobrazena chybová zpráva.

V dalším kroku se pro každý řádek dat vypočítávají pořadí hodnot v rámci tohoto řádku. Tento proces zahrnuje seřazení hodnot v každém řádku od nejnižší po nejvyšší a přiřazení pořadových čísel. V kódu se k tomu používá konstrukce LINQ, která pro každý řádek vytváří kolekci objektů obsahujících původní index a pořadí hodnoty. Tato pořadí jsou následně uložena do dvourozměrné matice `double[,] ranks`, kde každý prvek představuje pořadí konkrétní hodnoty v příslušném řádku a sloupci.

Následuje výpočet průměrných pořadí pro každý sloupec. Tento krok spočívá v součtu všech pořadí ve sloupci napříč všemi řádky. Tyto součty pořadí (`double[] rankSums`) jsou klíčové pro výpočet testové statistiky `double Q`. Každý sloupec představuje jedno ošetření nebo podmínku, a součty pořadí poskytují informace o tom, jak se jednotlivá ošetření liší mezi sebou.

Samotný výpočet testové statistiky Q probíhá podle vzorce, který zahrnuje součet čtverců hodnot `rankSums`, normalizovaný podle počtu bloků a podmínek. Tento vzorec zohledňuje rozdíly mezi průměrnými pořadími v jednotlivých sloupcích, což je základní myšlenka Friedmanova testu – testuje se, zda různé podmínky (ošetření) mají stejný efekt, nebo zda existují mezi nimi statisticky významné rozdíly.

Jakmile je hodnota Q vypočítána, je porovnána s kritickou hodnotou. Kritická hodnota je vypočítána pomocí funkce `double GetFriedmanCriticalValue(int n, int k)`, která využívá χ^2 rozdělení se stupni volnosti $k - 1$. Pokud je hodnota testové statistiky Q větší než tato kritická hodnota, zamítá se nulová hypotéza, což znamená, že alespoň jedna z podmínek se statisticky významně liší od ostatních.

Na závěr se výsledek testu zobrazí uživateli ve formě textového výstupu, který obsahuje vypočtenou hodnotu Q , kritickou hodnotu a rozhodnutí, zda byla nulová hypotéza zamítnuta či nikoliv. Kromě toho se otevře nové okno, ve kterém jsou výsledky testu vizualizovány pomocí grafů.

Metoda pro výpočet Friedmanova testu pak vypadá takto:

```
public double FriedmanTest(double[][] data, out double[] rankSums)
{
    int n = data.Length; // Počet řádků (bloků)
    int k = data[0].Length; // Počet sloupců (podmínek)

    // Nahradíme hodnoty v každém řádku jejich pořadím
    double[,] ranks = new double[n, k];
    for (int i = 0; i < n; i++)
    {
        // Získáme aktuální řádek
        var row = Enumerable.Range(0, k).Select(j => new { Value = data[i][j],
Index = j })
            .OrderBy(x => x.Value)
            .Select((x, rank) => new { x.Index, Rank = rank + 1
});

        // Uložíme pořadí zpět do matice ranks
        foreach (var item in row)
        {
            ranks[i, item.Index] = item.Rank;
        }
    }

    // Výpočet průměrného pořadí pro každý sloupec
    rankSums = new double[k];
    for (int j = 0; j < k; j++)
    {
        for (int i = 0; i < n; i++)
        {
            rankSums[j] += ranks[i, j];
        }
    }

    // Výpočet Q
    double sumRankSumsSquared = rankSums.Sum(rs => rs * rs);
    double Q = (12.0 / (n * k * (k + 1))) * sumRankSumsSquared - 3 * n * (k +
1);

    return Q;
}
```

3.4.7 Implementace Kruskal-Wallisova testu

Implementace Kruskal-Wallisova testu v tomto kódu začíná načtením dat z tabulky, která je součástí uživatelského rozhraní aplikace. Data jsou organizována ve dvourozměrném poli `double[][] kruskalData`, kde každý prvek představuje jednu skupinu pozorování (např.

různé populace nebo experimentální podmínky). Kód následně zajistí, aby všechna data byla platná čísla a poté jsou předána metodě `double KruskalWallisTest(double[][] data, out double[] T)`, která se stará o hlavní výpočet.

V metodě `KruskalWallisTest` je nejprve vypočítán celkový počet pozorování `int n` jako součet velikostí jednotlivých výběrů, které jsou reprezentovány proměnnou `int k`, což je počet skupin. Následně se všechna data seřadí vzestupně podle hodnoty a každému prvku je přiřazeno pořadí. Tento proces se realizuje pomocí LINQ konstrukce, která kombinuje data ze všech skupin do jednoho seznamu, seřadí je a přiřadí pořadí. Tato pořadí jsou pak zpětně rozdělena mezi původní skupiny.

Poté se pro každou skupinu vypočítá součet pořadových hodnot, označený jako `double[] T`, což je klíčový prvek při výpočtu testové statistiky Q . Tato pole `T` obsahují součty pořadí pro každou skupinu a slouží k určení, jak se skupiny liší v rozdělení hodnot.

Hodnota testové statistiky `double Q` se následně vypočítá podle vzorce zahrnujícího součet čtverců součtů pořadových hodnot `T`, normalizovaných podle velikostí jednotlivých skupin. Výsledná hodnota Q je měřítkem rozdílů mezi skupinami – čím vyšší je Q , tím větší jsou rozdíly mezi skupinami.

Po výpočtu Q se tato hodnota porovná s kritickou hodnotou, která je vypočítána pomocí metody `double ChiSquareCriticalValue(int df, double alpha)` s použitím χ^2 rozdělení s $k - 1$ stupni volnosti. Pokud je hodnota Q vyšší než kritická hodnota, nulová hypotéza se zamítá, což znamená, že alespoň jedna ze skupin se statisticky významně liší od ostatních.

Výsledky testu jsou zobrazeny uživateli ve formě textového výstupu, který obsahuje hodnotu Q , kritickou hodnotu a informaci o tom, zda byla nulová hypotéza zamítnuta. Kromě toho se otevře nové okno `KruskalWallisTestView`, kde jsou výsledky testu vizualizovány pomocí grafů.

Metoda pro výpočet Kurskal-Wallisova testu pak vypadá takto:

```
public double KruskalWallisTest(double[][] data, out double[] T)
{
    int k = data.Length; // Počet výběrů
    int n = data.Sum(arr => arr.Length); // Celkový počet pozorování
    var rankedData = data.SelectMany(d => d)
        .OrderBy(x => x)
        .Select((x, i) => new { Value = x, Rank = i + 1 })
        .ToList();

    T = new double[k];
    for (int i = 0; i < k; i++)
    {
        T[i] = data[i].Sum(value => rankedData.First(r => r.Value ==
value).Rank);
    }

    double Q = (12.0 / (n * (n + 1))) * T.Select((ti, i) => (ti * ti) /
data[i].Length).Sum() - 3 * (n + 1);

    return Q;
}
```

3.4.8 Implementace Spearmanova korelačního koeficientu

Implementace Spearmanova korelačního koeficientu v tomto kódu začíná načtením dat z tabulky, která je součástí uživatelského rozhraní aplikace. Data jsou organizována ve dvou jednorozměrných polích typu `double[]`, označených jako `columnX` a `columnY`, kde každý prvek představuje hodnotu jednoho z analyzovaných proměnných. Kód následně zajišťuje, aby oba sloupce měly stejný počet platných hodnot, a poté jsou tato data předána metodě `double SpearmanRankCorrelation(double[] x, double[] y, out double[] rankX, out double[] rankY)`, která provádí hlavní výpočet Spearmanova korelačního koeficientu.

V metodě `SpearmanRankCorrelation` se nejprve spočítá pořadí pro každé pole hodnot zvlášť, a to pomocí pomocné metody `private static double[] GetRanks(double[] values)`. Tato metoda seřadí vstupní hodnoty v poli `values` a přiřadí jim odpovídající pořadí. Pořadí hodnot jsou uložena do výstupních parametrů `rankX` a `rankY`, které jsou poté použity k výpočtu rozdílů mezi odpovídajícími pořadími v polích `rankX` a `rankY`.

Následně se vypočítá součet čtverců rozdílů pořadí mezi jednotlivými prvky, což je označeno jako `double sumD2`. Tento součet čtverců rozdílů je klíčový pro výpočet Spearmanova korelačního koeficientu r_s . Následně se podle vzorce vypočítá hodnota r_s , která je pak návratovou hodnotou metody `SpearmanRankCorrelation`.

Po výpočtu hodnoty r_s se tato hodnota porovná s kritickou hodnotou, která je určena pomocí metody `double GetSpearmanCriticalValue(int n, double significanceLevel = 0.05)`. Tato metoda vrací kritickou hodnotu pro Spearmanův korelační koeficient na

základě velikosti výběru n a zadané hladiny významnosti. Pokud je počet pozorování n menší nebo roven 30, kritická hodnota se získá z předdefinovaného pole tabulkových hodnot. Pro výběry s větším počtem pozorování se kritická hodnota vypočítá pomocí normalizační aproximace, kde se použije kvantil standardního normálního rozdělení, určený funkcí `NormalInverseCDF`.

Pokud je absolutní hodnota Spearmanova koeficientu r_s větší nebo rovna kritické hodnotě, nulová hypotéza o nezávislosti veličin je zamítnuta. Výsledky testu, zahrnující hodnotu r_s , kritickou hodnotu a rozhodnutí o zamítnutí nebo nezamítnutí nulové hypotézy, jsou následně zobrazeny uživateli ve formě textového výstupu.

Metoda pro výpočet Spearmanova korelačního koeficientu pak vypadá takto:

```
public double SpearmanRankCorrelation(double[] x, double[] y, out double[]
rankX, out double[] rankY)
{
    int n = x.Length;

    // Seřazení a přiřazení pořadí
    rankX = GetRanks(x);
    rankY = GetRanks(y);

    // Výpočet rozdílů pořadí a jejich druhých mocnin
    double sumD2 = 0;
    for (int i = 0; i < n; i++)
    {
        double d = rankX[i] - rankY[i];
        sumD2 += d * d;
    }

    // Výpočet Spearmanova koeficientu
    double rs = 1 - (6 * sumD2) / (n * (n * n - 1));
    return rs;
}
```

3.4.9 Implementace Kendallova korelačního koeficientu

Implementace Kendallova korelačního koeficientu v tomto kódu začíná načtením dat z tabulky. Data jsou organizována ve dvou jednorozměrných polích typu `double[]`, označených jako `kendallX` a `kendallY`, kde každý prvek představuje hodnotu jednoho z analyzovaných proměnných. Kód zajišťuje, aby oba sloupce měly stejný počet platných hodnot. Pokud tato podmínka není splněna, výpočet se zastaví a uživateli je zobrazena odpovídající chybová zpráva. Pokud jsou data validní, jsou předána metodě `double KendallTau(double[] x, double[] y)`, která provádí hlavní výpočet koeficientu Kendallova korelačního koeficientu.

Metoda `KendallTau` začíná inicializací proměnné `int n`, která určuje délku pole `x`, tedy počet prvků v těchto dvou polích. Následně jsou vytvořeny dvojice hodnot z polí `x` a `y`, přičemž každá dvojice je reprezentována objektem s vlastnostmi `x` a `y`. Tento proces je realizován pomocí metody `Zip`, která kombinuje odpovídající prvky z obou polí. Tyto dvojice jsou poté seřazeny podle hodnoty `x` pomocí metody `OrderBy`, což zajistí, že data budou uspořádána podle hodnot první proměnné `x`. Výsledkem tohoto kroku je seznam seřazených dvojic, uložený do seznamu `var pairedData`.

Následuje výpočet proměnné `int sum_k`. Pro každý prvek v seznamu `pairedData` se prochází všechny následující prvky v seznamu. Pokud hodnota `y` pro prvek s indexem `j` je větší než hodnota `y` pro prvek s indexem `i`, je zvýšena hodnota proměnné `int k_i`. Tento proces se opakuje pro každý prvek s indexem `i`, a hodnoty `k_i` jsou následně přičítány do celkové proměnné `sum_k`.

Nakonec se hodnota Kendallova korelačního koeficientu `double tau` vypočítá podle vzorce. Tento koeficient vyjadřuje míru závislosti mezi dvěma proměnnými na základě pořadí hodnot. Výsledná hodnota `tau` je vrácena jako výsledek metody.

Po výpočtu hodnoty `tau` je tato hodnota porovnána s kritickou hodnotou, která je určena pomocí metody `double GetKendallTauCriticalValue(int n)`. Pokud je počet pozorování `n` menší nebo roven 30, kritická hodnota je získána z předdefinovaného pole tabulkových hodnot. Pro větší vzorky se používá metoda `double GetKendallTauCriticalValueForLargeSample(int n, double significanceLevel)`, která vrací kritickou hodnotu vypočtenou pomocí normalizační aproximace, využívající kvantil normálního rozdělení.

Pokud je absolutní hodnota koeficientu větší nebo rovna kritické hodnotě, nulová hypotéza o nezávislosti veličin je zamítnuta. Výsledky testu, zahrnující hodnotu `tau`, kritickou hodnotu a rozhodnutí o zamítnutí nebo nezamítnutí nulové hypotézy, jsou následně zobrazeny uživateli ve formě textového výstupu.

Metoda pro výpočet Kendallova korelačního koeficientu pak vypadá takto:

```
public double KendallTau(double[] x, double[] y)
{
    int n = x.Length;

    // Seřadit dvojice (X, Y) podle hodnot X
    var pairedData = x.Zip(y, (xi, yi) => new { X = xi, Y = yi })
        .OrderBy(pair => pair.X)
        .ToList();

    int sum_k = 0;

    // Výpočet počtu k_i pro každý i
    for (int i = 0; i < n - 1; i++)
    {
        int k_i = 0;
        for (int j = i + 1; j < n; j++)
        {
            if (pairedData[j].Y > pairedData[i].Y)
            {
                k_i++;
            }
        }
        sum_k += k_i;
    }

    // Výpočet Kendallova tau
    double tau = (4.0 * sum_k) / (n * (n - 1)) - 1;
    return tau;
}
```

3.4.10 Implementace faktorové analýzy

Implementace faktorové analýzy v kódu začíná metodou `PerformPCA`, která provádí analýzu hlavních komponent (PCA) nad vstupními daty. Tato metoda přijímá dvourozměrné pole `double[,] data`, které představuje původní data s pozorováními ve sloupcích a jednotlivými proměnnými v řádcích. Dále přijímá volitelný parametr `double varianceThreshold`, který určuje minimální kumulativní vysvětlený rozptyl pro výběr počtu faktorů (hlavních komponent).

Nejprve se v metodě `PerformPCA` spočítají rozměry dat, kde `int n` je počet proměnných a `int p` je počet pozorování. Poté se z těchto dat vypočítá kovarianční matice. Vstupní data jsou nejprve upravena tak, že se od každého prvku odečte průměr hodnot v daném sloupci, což vytvoří tzv. `meanAdjustedData`. Kovarianční matice se následně spočítá jako matice vnitřních součinů normalizovaných podle počtu pozorování.

Pro výpočet vlastních čísel a vlastních vektorů kovarianční matice se používá metoda SVD (Singular Value Decomposition), kterou umožňuje knihovna Math.NET Numerics. Tato metoda vrací spektrální hodnoty matice jako diagonální matice a vlastní vektory jako matice sloupců. Hodnoty vlastních čísel se uloží do pole `double[] eigenValues` a vlastní vektory do pole `double[,] eigenVectors`. Vlastní čísla jsou seřazena v sestupném pořadí a odpovídající vlastní vektory jsou rovněž přeuspořádány tak, aby odpovídaly tomuto pořadí.

Dále jsou vypočteny podíly vysvětleného rozptylu jednotlivých hlavních komponent (`double[] explainedVarianceRatios`), což jsou normalizované vlastní hodnoty. Z těchto podílů se následně počítá kumulativní podíl vysvětleného rozptylu (`double[] cumulativeVarianceRatios`), který ukazuje, jaký podíl celkového rozptylu dat je vysvětlený prvními n hlavními komponentami.

Na základě zadaného prahu kumulativního vysvětleného rozptylu (parametr `varianceThreshold`) se určí počet faktorů (`int numFactors`), které budou použity pro další analýzu. Tento počet faktorů odpovídá minimálnímu počtu hlavních komponent potřebných k dosažení stanoveného prahu vysvětleného rozptylu.

Nakonec jsou původní data transformována do prostoru hlavních komponent. Toto je provedeno tak, že se původní data vynásobí maticí vlastních vektorů odpovídajících vybraným hlavním komponentám. Výsledkem je nová matice `double[,] transformedData`, která obsahuje transformovaná data ve sníženém rozměru.

Výstupem metody `PerformPCA` jsou transformovaná data, podíly vysvětleného rozptylu, kumulativní podíly vysvětleného rozptylu a počet vybraných faktorů, kde mimo transformovaných dat jsou tyto hodnoty vypsány do textového bloku v hlavním okně. Kromě toho se otevře nové okno `FactorAnalysisView`, kde jsou výsledky vizualizovány pomocí tabulky a grafů.

Závěr

Tato bakalářská práce byla pro mě významným krokem, během kterého jsem se naučil mnoho nového jak v teoretické, tak praktické rovině. Výsledkem mého úsilí je funkční aplikace, která zpracovává psychometrická data pomocí různých statistických metod. Aplikace umožňuje uživatelům efektivně provádět analýzy, které jsou klíčové pro objektivní hodnocení psychologických vlastností jedinců. Vytvoření tohoto nástroje bylo náročným úkolem, který si vyžádal hluboké porozumění mnoha statistickým metodám, stejně jako znalost programování a softwarového inženýrství.

Jednou z nejnáročnějších částí této práce bylo nastudování a implementace faktorové analýzy. Tato metoda, která je ve srovnání s ostatními použitými statistickými nástroji výrazně složitější, vyžadovala důkladné pochopení teorie i technických aspektů. Implementace této analýzy mi poskytla cenné zkušenosti, které považuji za významný přínos této práce.

Vývoj této aplikace mě také naučil, jak efektivně využívat různé programovací knihovny a technologie, jako je OxyPlot a Math.NET Numerics, což mi poskytlo praktické dovednosti, které budu moci využít v dalších projektech. Práce na této aplikaci mě donutila přemýšlet o architektuře softwaru a o tom, jak navrhnout systém, který je flexibilní a snadno rozšiřitelný. Díky tomu je aplikace navržena tak, aby umožňovala jednoduchou integraci nových statistických testů a funkcionalit v budoucnu.

Kromě technických aspektů jsem si také uvědomil důležitost uživatelského rozhraní, které hraje klíčovou roli v tom, jak snadno a efektivně mohou uživatelé aplikaci používat. Ačkoli jsem se soustředil na funkčnost aplikace, uvědomuji si, že uživatelské rozhraní by mohlo být modernizováno a zpřístupněno širší škále uživatelů.

Aplikace má velký potenciál pro další rozvoj a vylepšení. V budoucnu by mohla zahrnovat nejen rozšíření o další statistické testy, ale také o možnosti jako volba hladiny významnosti nebo práce s oboustrannou alternativní hypotézou. Tyto funkce by dále zvýšily flexibilitu a použitelnost aplikace pro psychometrická testování.

Práce na této aplikaci mi poskytla nejen hlubší porozumění teoretickým konceptům statistiky a psychometrie, ale také praktické zkušenosti, které mě připravily na další výzvy v oblasti softwarového vývoje. Věřím, že aplikace, kterou jsem vytvořil, může sloužit jako užitečný nástroj v oblasti psychometrického výzkumu a že její další rozvoj přinese ještě větší hodnotu pro odborníky, kteří v této oblasti pracují.

V neposlední řadě jsem získal cenné zkušenosti, které se týkají nejen vývoje softwaru, ale také projektového řízení a řešení problémů. Naučil jsem se, jak plánovat a organizovat svou práci tak, aby bylo možné efektivně dosáhnout cíle. Jsem přesvědčen, že dovednosti, které jsem během této práce získal, mi budou užitečné v mé budoucí kariéře, ať už se rozhodnu pokračovat ve vývoji softwaru, nebo se zaměřím na další výzkumné projekty.

Literatura

1. URBÁNEK, Tomáš. Základy psychometriky. Brno: Masarykova univerzita, Filozofická fakulta, 2002. ISBN 80-210-2797-5. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:aa7b3ef0-c111-11ea-b7a2-005056827e51>
2. SVOBODA, Mojmír. Psychologická diagnostika dospělých. Praha: Portál, 1999. ISBN 80-7178-327-7. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:a71baef0-003a-11e4-9789-005056827e52>
3. BLATNÁ, Dagmar. Statistika a pravděpodobnost. Praha: Bankovní institut, 2003. ISBN 80-7265-059-9. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:1f41bda0-d415-11e2-8f90-005056827e51>
4. KAŇOKOVÁ, Jara. Teorie statistiky pro řízení a plánování: celost. vysokošk. učebnice pro stud. ekon. fakult. Praha: SNTL, 1989, s. 347. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:90630620-d350-11e3-85ae-001018b5eb5c>
5. NOVOVIČOVÁ, Jana. Pravděpodobnost a matematická statistika. Praha: ČVUT, Dopravní fakulta, 1999. ISBN 80-01-01980-2. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:bff844a0-f836-11e6-8830-005056827e51>
6. REITEROVÁ, Eva a Univerzita Palackého. Základy psychometrie. Olomouc: Univerzita Palackého, 2003. ISBN 80-244-0717-5. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:39e968a0-eabe-11e2-9923-005056827e52>
7. ANDĚL, Jiří. Matematická statistika. Praha: SNTL, 1985. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:13f491f0-e11b-11e5-a3e0-005056827e51>
8. REIF, Jiří a Západočeská univerzita. Metody matematické statistiky. Plzeň: Západočeská univerzita, 2000. ISBN 80-7082-593-6. Dostupné také z: <https://www.digitalniknihovna.cz/mzk/uuid/uuid:02ef2130-7309-11ee-9a22-5ef3fc9bb22f>
9. LUKASOVÁ, Alena a Jana ŠARMANOVÁ. Metody shlukové analýzy. Praha: Státní nakladatelství technické literatury, 1985. Dostupné také z:

<https://www.digitalniknihovna.cz/mzk/uuid/uuid:31e5c060-dc38-11e8-a5a4-005056827e52>

10. MICROSOFT. *Introduction to .NET*. Online. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/core/introduction>. [cit. 2024-08-12].
11. MICROSOFT. *A tour of the C# language*. Online. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview>. [cit. 2024-08-12].
12. MICROSOFT. *XAML overview (WPF .NET)*. Online. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0>. [cit. 2024-08-12].
13. MICROSOFT. *Desktop Guide (WPF .NET)*. Online. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-8.0>. [cit. 2024-08-15].
14. OXYPLOT. *Welcome to OxyPlot's documentation!*. Online. Dostupné z: <https://oxyplot.readthedocs.io/en/latest/>. [cit. 2024-08-15].
15. MICROSOFT. *What is Visual Studio?* Online. Dostupné z: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>. [cit. 2024-08-15].
16. MICROSOFT. *Model-View-ViewModel (MVVM)*. Online. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>. [cit. 2024-08-16].
17. MATH.NET. *Math.Net Numerics*. Online. Dostupné z: <https://numerics.mathdotnet.com/>. [cit. 2024-08-18].

Příloha A – MIT Licence knihovny OxyPlot

MIT License

Copyright (c) 2014 OxyPlot contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Příloha B – MIT Licence knihovny Math.NET Numerics

MIT License

Copyright (c) 2002-2022 Math.NET

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.