# 1.Teoretiskafrågor

1.Beskriv kort hur en relationsdatabas fungerar.

En relationsdatabas är en datatyp som lagrar och ger tillgång till data som är organiserad i tabeller. Dessa tabeller har data ordnade i rader (som tillhandahåller entiteter) och kolumner (som innehåller entitetsattribut) genom en process som kallas normalisering. Varje rad innehåller en unik identifierare eller nyckel som länkar tabeller mellan dem för att skapa en relation. Så när du frågar en relationsdatabas används nyckeln för att hitta relevant data bland datauppsättningar. En relationsdatabas är därför ett verktyg för effektiv hantering av stora mängder strukturerad data som enkelt kan skapa relationer mellan olika typer av data och därmed svara på komplexa frågor och hjälpa till i deras analys. SQL används för att interagera med en relationsdatabas.

2.Vadmenas med ”CRUD” flödet?

Creat Read Update Delete

CRUD är en akronym som representerar de fyra grundläggande operationerna som används i databaser och andra informationslagringssystem för att manipulera data. CRUD står för: Create (Skapa) som syftar på att skapa eller lägga till ny data (till exempel lägga till nya rader i en tabell)

Read (Läs) som syftar på att hämta eller läsa data

Update (Uppdatera) som avser uppdatering eller ändring av befintlig data (till exempel att ändra värdena i en rad i en tabell)

Delete (Radera) som hänvisar till att radera data (till exempel att ta bort rader från en tabell)

3.Beskriv kort vad en ”leftjoin” och ”inner join” är. Varför använder man det?

Båda är SQL-anslutningstyper. De används för att kombinera rader från två eller flera tabeller baserat på en relaterad kolumn mellan dem. Specifikt returnerar ''left join'' alla rader från den vänstra (dvs första) tabellen och motsvarande rader från den högra (andra) tabellen. Om det inte finns någon matchning i den högra tabellen kommer raden från den vänstra tabellen fortfarande att returneras men alla kolumner från den högra tabellen kommer att ha nollvärden. Däremot returnerar en "inner join" endast de rader där det finns en matchning i båda tabellerna. Det vill säga, om en rad i en tabell inte har en motsvarande rad i den andra tabellen, så ingår inte denna rad i resultatet. ''Left join'' och ''inner join'' används och är mycket användbara eftersom de ger någon möjlighet att samla in och analysera data från olika tabeller som är relaterade till varandra, vilket är vad vi vet är vanligt i relationsdatabaser. På så sätt skapas mer flexibla och mer informativa datamängder som bättre hjälper till vid analys och beslutsfattande.

4.Beskriv kort vad indexering i SQL innebär.

SQL-indexering är en databasfunktion som förbättrar hastigheten för att hämta data från en databastabell. Det liknar ett register i en bok. Det är med andra ord en hjälpstruktur som möjliggör snabbare sökning och hämtning av data. Ett index skapas på en eller flera kolumner i en databastabell. Indexering minskar tiden som krävs för att söka i en tabell för att hitta specifika rader, vilket är mycket användbart i stora databaser med tusentals rader. Indexering är ett kraftfullt verktyg i SQL för att optimera frågor och förbättra prestandan för databasoperationer, men det måste användas strategiskt för att balansera hämtningsprestanda med kostnaden för indexunderhåll.

5.Beskriv kort vad en vy i SQL är.

En vy i SQL är en virtuell tabell vars innehåll är baserat på resultaten av en SQL-fråga. En vy består av en uppsättning namngivna kolumner och rader med data. Raderna och kolumnerna med data kommer från tabeller eller andra SQL Server-vyer som refereras till i frågan som definierar vyn och skapas dynamiskt när frågan exekveras. Frågan som definierar vyn kan komma från en eller flera tabeller eller från andra vyer i en eller flera databaser. En vy i SQL är en virtuell tabell som representeras av en SQL-fråga. Det är i grunden ett lagrat frågeresultat som kan användas som om det vore en vanlig tabell. En vy skapas med en CREATE VIEW-sats följt av en SQL-fråga. Den här frågan definierar data som ska inkluderas i vyn. Vyn lagrar inte data fysiskt (det är en virtuell tabell). Vyer används för att förenkla komplexa frågor, dölja komplexitet för användare eller skydda känslig data genom att endast visa en delmängd av datan. De kan användas på exakt samma sätt som en vanlig tabell i SQL-frågor. Det vill säga, någon kan göra SELECT-förfrågningar från en vy. Genom att begränsa åtkomsten till specifika kolumner eller rader med data kan vyer bidra till förbättrad datasäkerhet. Vyer kan dölja komplexa SQL-frågor bakom ett enklare gränssnitt, vilket gör det lättare för användare att hämta data utan att förstå den underliggande komplexiteten. Vyer kan återanvändas i olika frågor och applikationer genom att tillhandahålla en form av återanvändning. Eftersom vyer inte lagrar data naturligt kan de ibland vara långsammare än att fråga direkt mot de faktiska tabellerna, särskilt för stora eller komplexa frågor. Man kan inte alltid använda INSERT, UPDATE eller DELETE i alla vyer. Sammanfattningsvis är SQL-vyer kraftfulla verktyg för att hantera och presentera data effektivt och säkert och erbjuder flexibilitet i databashantering.

6.Beskriv kort vad en lagrad procedur i SQL är.

En lagrad SQL-procedur är en samling SQL-satser lagrade i databasen. Det fungerar som en funktion som kan köras från databasen för att utföra en specifik uppgift eller serie av uppgifter. En lagrad procedur är ett paket av åtgärder som består av SQL-satser och valfri kontrollflödeslogik. Den definieras en gång och lagras i databasen. Skapat med CREATE PROCEDURE-satsen. Den kan innehålla en eller flera SQL-satser och kan även innehålla programmeringslogik som IF-satser, WHILE-loopar etc. Sammanfattningsvis är lagrade procedurer ett effektivt verktyg i SQL för att hantera komplexa databasoperationer, automatisera uppgifter och förbättra prestanda och säkerhet i databassystem. Lagrade procedurer är ett mycket bra sätt att kapsla in din SQL-kod och lagra den nära dina data i datalagret. Lagrade procedurer hjälper utvecklare att strukturera sina lösningar genom att dela upp kod i hanterbara enheter och underlätta bättre återanvändning av kod.

# 2.Programmeringsuppgift och Rapport

1. Deskriptiv sammanfattning

I databasen AdventureWorks 2022 finns det många tabeller som spänner över olika aspekter av ett företags verksamhet. Nyckeltabeller inkluderar:

Person.Address, Person.EmailAddress, Person.Person som innehåller kund- och anställds personuppgifter.

HumanResources.Employee, HumanResources.Department som relaterar till personalinformation och avdelningsdata.

Sales.Customer, Sales.SalesOrderHeader, Sales.SalesOrderDetail som involverar kundorder och försäljningstransaktioner.

Production.Product, Production.ProductReview som relaterat till produktdetaljer och kundrecensioner.

Purchasing.PurchaseOrderHeader, Purchasing.PurchaseOrderDetail som rör information om leverantör och inköpsorder.

Dessa tabeller ger bland annat en omfattande datauppsättning för olika typer av analyser som försäljningstrender, kundbeteende, anställdas prestationer och lagerhantering. Således innehåller databasen ett brett utbud av data relaterad till en tillverkare av cyklar och tillbehör, inklusive produktinventering, försäljning, inköp, mänskliga resurser och annat där en intresserad part kan göra försäljningsanalys, produktanalys, kundanalys, analys av den mänskliga potentialen och vad som rör medarbetarna men också vad som gäller lager, leverantörer och generellt leverantörskedjan.

Så med den här databasen måste jag jobba som nyanställd på ett företag. För att få en global bild tog jag först en titt på all data så att jag sedan kunde välja de som jag ville undersöka och som jag tror att deras resultat har något att erbjuda företaget. Så genom SQL Server Management Studio började jag sätta SQL-queries först för att hitta vilka produkter som har flest recensioner. Så jag behövde analysera tabellen Production.ProductReview genom att gruppera efter produkt och år och förklarade Year-funktionen från ReviewDate och grupperade efter Year,ProductID och ProductName. Min andra nyfikenhet var att leta efter de produkter som har högst ‘’Average Rating’’. Jag extraherade året tillbaka från ReviewDate, beräknade AverageRating för varje produkt för varje år och sedan grupperar resultaten efter Year, ProductID och ProduktName (och sorterar i fallande ordning). Min nästa strävan var att hitta produkterna med högst försäljning per år. Jag tänkte samla in försäljningsdata från SalesOrderDetail, beräkna den totala försäljningen för varje produkt och sedan gruppera resultaten efter OrderDate och ProductID (i fallande ordning). Min närmaste tanke och fråga samtidigt var att se de produkter som haft störst försäljning under dessa 4 år till vilken produktkategori de tillhör och gjorde det genom att ansluta sig till tabellerna som hänvisar till produktförsäljningsinformation, produktinformation och kategoriinformation. Sedan undrade jag över något som antagligen berör de flesta företag och det är effekten av rabatter, om rabatter leder till ökad försäljning (vilket säkert inte alltid innebär större vinst). Så jag tittade också på tabellerna som hänvisar till rabatter, och undersökte först de tre produkter som hade högst försäljning åren 2011-2014. Sedan för dessa specifika produkter hittade jag de rabatterade och icke-rabatterade kvantiteterna sålda såväl som den rabatterade och icke-rabatterade försäljningen per år samt procentandelen av rabatten och jag fick idén att hitta försäljningsskillnaderna mellan rabatterade produkter och icke-rabatterade produkter.

2. Statistisk analys (konfidensintervall)

Den **statistiska analysen** jag utförde var i samband med försäljningsdata, för att jämföra försäljningsintäkterna för samma uppsättning produkter när de såldes med rabatt jämfört med när de såldes till fullt pris så utföras ett paired t-test med funktionen ttest\_rel från SciPy-biblioteket. Paired t-test användes eftersom det finns en en-till-en-koppling mellan DiscountedTotalSales och NonDiscountedTotalSales. Varje par består av försäljning av samma produkt eller kategori, en säljs till ett rabatterat pris och det andra till ett icke-rabatterat pris inom samma tidsram (samma år). För att utföra den statistiska analysen antar jag som en nollhypotes (H0) att det inte finns någon skillnad i genomsnittlig försäljningsintäkt mellan rabatterad försäljning och icke-rabatterad försäljning, det vill säga rabatter påverkar inte försäljningen. Eventuella observerade skillnader beror på en slumpmässig slump. Jag ställer också som en alternativ hypotes (H1) att det finns en signifikant skillnad i genomsnittlig försäljningsintäkt mellan rabatterad och icke-rabatterad försäljning, det vill säga att rabatter påverkar försäljningen. När det paired t-testet utförs undersöks och testas parade observationer (försäljning av samma produkt med och utan rabatt) för att se om deras medelskillnad skiljer sig statistiskt signifikant från noll. T-statistiken mäter storleken på skillnaden i förhållande till variansen i våra urvalsdata. Ett litet P-värde (typiskt ≤ 0.05) indikerar starka bevis mot H0, medan ett stort P-värde (> 0.05) indikerar svaga bevis mot H0. I mitt fall har jag resultatet: Paired T-statistic: -1.0670347994593568, P-value: 0.3642017359788525 När det gäller den Paired T-statistiken (= -1.0670347994593568) indikerar det negativa värdet att medelvärdet av den rabatterade försäljningsintäkten är lägre än medelvärdet av den icke-rabatterade försäljningsintäkten. Men t-statistiken ensam berättar inte om denna skillnad är statistiskt signifikant.

När det gäller P-värdet (=0.3642017359788525) är detta p-värde större än de vanliga signifikansnivåerna på 0.05 (5%), vilket betyder att det inte finns tillräckligt med bevis för att förkasta nollhypotesen. Skillnaden i försäljningsintäkter mellan rabatterad och icke-rabatterad försäljning är med andra ord inte statistiskt signifikant vid vanliga trösklar. Det vill säga, i mitt fall, eftersom P-värdet är större än 0.05 så finns det inga statistiska bevis för att förkasta nollhypotesen, vilket innebär att det utifrån mina data inte finns några signifikanta bevis för att rabatter påverkar försäljningen. Detta bevisar dock inte att rabatter inte har någon effekt utan betyder kanske bara att med de data och testet jag gjorde kunde statistisk signifikans inte hittas.

Sammanfattningsvis och utifrån resultaten från t-testet kan man säga att det inte finns någon statistiskt signifikant skillnad i försäljningsintäkter mellan rabatterad och icke-rabatterad försäljning. Detta kan tyda på att prissättningsstrategin (rabatt kontra fullt pris) inte har en betydande inverkan på totala försäljningsintäkter, eller så kan det indikera att testets kraft är för låg för att upptäcka en skillnad, möjligen på grund av ett litet urval eller låg variation i data.

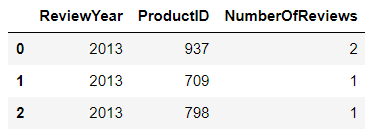
Nästa steg i studien var skapandet av ett **konfidensintervall**. Så med hjälp av SciPy Python-biblioteket beräknade jag ett konfidensintervall för den genomsnittliga totala försäljningen per beställning i kategorin "Bikes". För att beräkna konfidensintervallet initialt ställ in konfidensnivån lika med 0,95 vilket betyder att konfidensintervallet kommer att fånga det sanna populationsmedelvärdet på 95 %. Frihetsgrader beräknades sedan genom att ta antalet observationer och subtrahera en. Nödvändigt för att hitta konfidensintervallet är bestämningen av provmedelvärdet som beräknar den genomsnittliga totala försäljningen från kolumnen "TotalSales" som representerar data från kategorin "Bikes". Det är också nödvändigt att beräkna standardfelet för att uppskatta standardavvikelsen för provmedelvärdet. Detta är ett mått på hur mycket stickprovets medelvärde förväntas skilja sig från ett urval till ett annat. Så genom t-fördelningen beräknar alltså gränserna för konfidensintervallet givet konfidensnivån, frihetsgrader, urvalsmedelvärde och standardfel.

Det resulterande konfidensintervallet är ungefär (4985.20, 5320.90). Detta intervall ger ett intervall inom vilket företaget kan vara 95 % säker på den faktiska genomsnittliga totala försäljningen per order för cyklar. Det vill säga, om företaget upprepade gånger provade försäljningsdata från samma population och beräknade konfidensintervallet varje gång, skulle 95 % av dessa intervall innehålla det verkliga medelvärdet av den totala försäljningen per order. Intervallet indikerar inte att den genomsnittliga försäljningen per order kommer att ligga inom detta intervall för framtida försäljning. Det är snarare genomsnittet av populationen som urvalet togs från. Det är också värt att notera att konfidensintervallet endast ger en intervalluppskattning av det sanna medelvärdet. Det betyder inte att all eller större delen av försäljningen faller inom detta intervall. Men denna information från uppskattningar kan vara användbar för att sätta förväntningar på försäljningsresultat, utvärdera framgången för försäljningsstrategier och fatta affärsbeslut relaterade till prissättning, marknadsföring och lagerhantering.

3. Slutsatser och rekommendationer

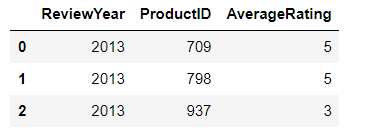
Både den data jag var tvungen att hantera och resultaten från deras analys tror jag var lika viktiga. Eftersom jag tog hand om olika frågor kommer jag att ta dem i ordning:

Till att börja med ville jag hitta de produkter som fick flest recensioner per år och jag hade några tabeller som data. Så jag behövde analysera tabellen Production.ProductReview genom att gruppera efter produkt och år. Efter att jag förklarade Year-funktionen från ReviewDate och grupperade efter Year,ProductID och ProductName överraskade resultatet mig eftersom jag såg att det finns recensioner endast för år 2013 och de flesta av dem hänvisar definitivt till produkterna med ProductID 937, 709 och 798 med 2 , 1 och 1 recensioner respektive. Resultaten kan också ses i tabellen nedan som presenteras i Jupyter.



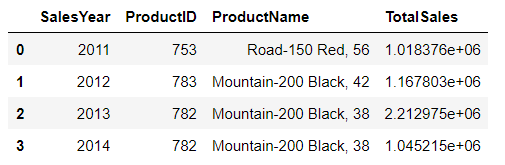
Jag tycker att företaget borde vara medvetet om detta som ett viktigt resultat för att ta hand om att leta efter recensioner på sina produkter, för på så sätt kan de förbättras eller uppgraderas.

Sedan hittade jag de högst ‘’AverageRating’’ produkterna ett år efter att jag extraherade året tillbaka från ReviewDate, beräknade ‘’AverageRating’’ för varje produkt för varje år och sedan grupperar resultaten efter år, productid och produktnamn (och sorterar i fallande ordning). Resultaten är vad som kan ses i tabellen nedan:

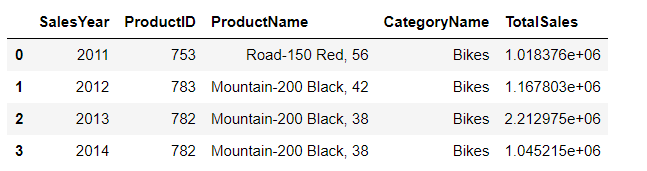


Så kan man märker att finns först bara data för 2013 (något som redan var känt från ovanstående studie) och sedan att produkterna med högst ‘’AverageRating’’ på 5 är de med productid 709 och 798, medan 937 följer med ett ‘’AverageRating’’ på 3. Det vill säga man märker att produkten med flest recensioner har det lägsta genomsnittliga betyget av de med få recensioner och det högsta genomsnittliga betyget. Detta kan vara ett viktigt resultat för företaget för även om det inte finns tillräckligt med data för ett mer tillförlitligt resultat kan detta visa att det finns några klagomål om 937-produkten som fick ett lägre genomsnittligt betyg, så det kan behöva lite uppmärksamhet och förbättring från företaget.

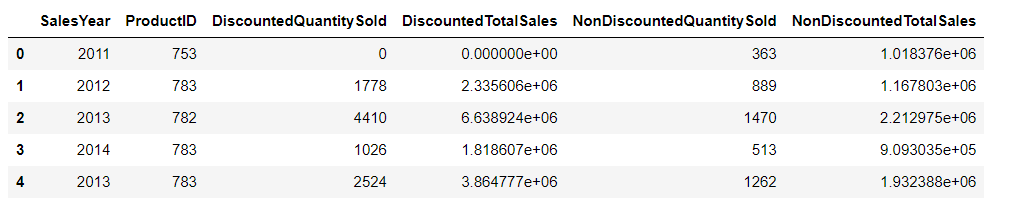
Till min nästa fråga om produkterna med högst försäljning per år samlade jag in försäljningsdata från SalesOrderDetail, beräknade den totala försäljningen för varje produkt och sedan grupperade resultaten efter OrderDate och ProductID (i fallande ordning). Ovanstående procedur resulterade i produkten 753 för 2011, produkten 783 för 2012 och produkten 782 för de två följande år 2013 och 2014.



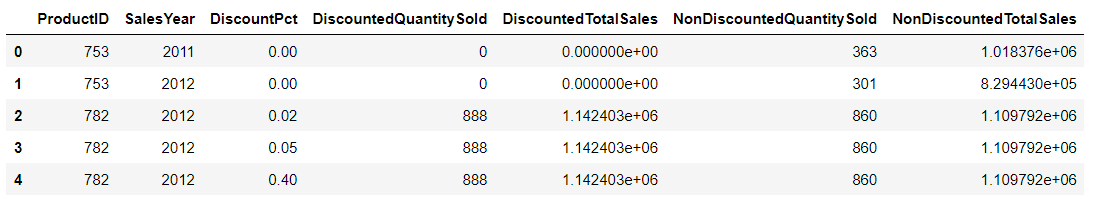
Man märker att under två år i rad har samma produkt den högsta försäljningen, vilket är mycket intressant eftersom detta kan betyda att kunderna är mycket nöjda med detta företags produkt och att det verkligen är en mycket bra konkurrenskraftig produkt och även att företagets marknadsföringsstrategi är den rätta och kanske användbar för dess tillämpning på andra produktföretag för att bli samma konkurrenskraftiga. Den närmaste tanken var att hitta de kategorier som produkterna med högst försäljning per år hörde till enligt föregående tabell, det vill säga produkterna med produkt-ID 753, 783 och 782. Genom att ansluta sig till tabellerna som hänvisar till produktförsäljningsinformation, produktinformation och kategoriinformation men endast för produkterna med productid 753, 783 och 782 märker man att de alla tillhör kategorin cyclar ("Bikes") vilket också betyder mycket för företaget och eventuella avdelning av sina produkter som den måste investera och kanske utveckla. Det är viktigt för ett företag att känna till sina styrkor.



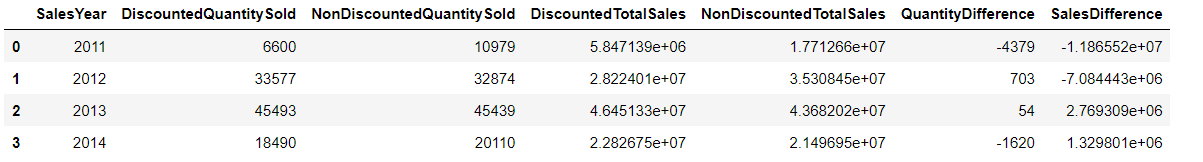
Sedan studerade jag något som antagligen berör de flesta företag och det är effekten av rabatter, om rabatter leder till ökad försäljning (vilket säkert inte alltid innebär större vinst). Så jag tittade också på tabellerna som hänvisar till rabatter och undersökte först de tre produkter som hade högst försäljning åren 2011-2014. Man märker att produkten 753 inte hade rabatt medan den hade de högsta priserna för 2011 vilket betyder att folk föredrog den och köpte den utan rabatt.



De andra två produkterna som fanns på rea hade dock ökad försäljning vid första anblicken. Sedan för dessa specifika produkter hittade jag de rabatterade och icke-rabatterade kvantiteterna sålda såväl som den rabatterade och icke-rabatterade försäljningen per år samt procentandelen av rabatten. Man kan se att uppgifterna behöver undersökas ytterligare eftersom exakt samma kvantiteter av en anledning presenteras med olika rabattprocent och det observeras också att rabatterna har gett nästan samma försäljning vilket gör att rabatterna inte ökat försäljningen drastiskt.

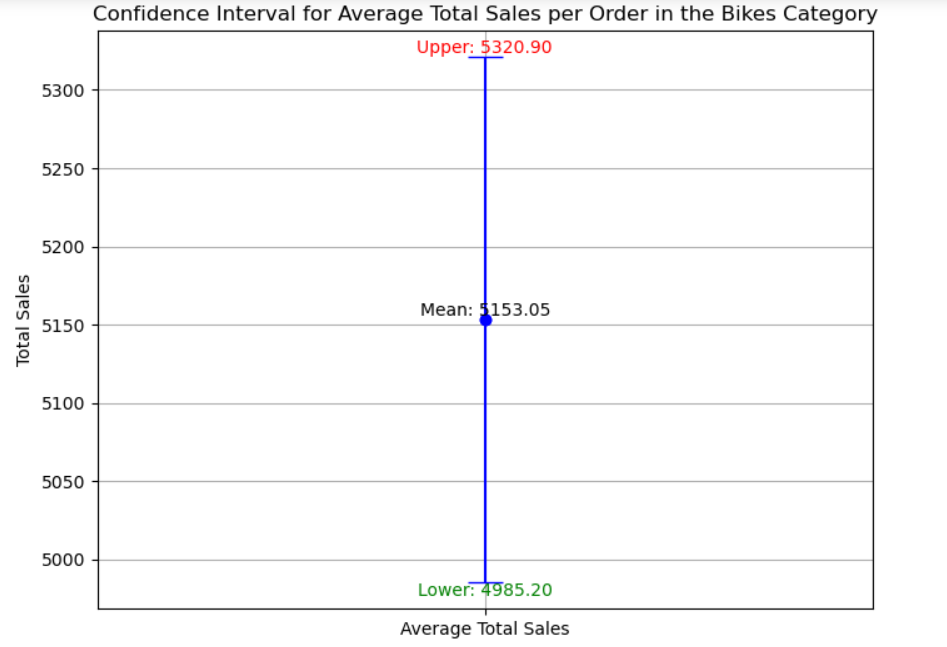


Sedan fick jag idén att hitta försäljningsskillnaderna mellan rabatterade produkter och icke-rabatterade produkter, också mycket viktigt för företaget att veta om rabatterna är av största vikt eller att bestämma prispolicyn eller marknadsföringen därefter.



Resultaten är intressanta och lite olika från år till år. Statistisk analys visar oss verkligen en bättre bild. Ett parat t-test genomfördes sedan för att jämföra försäljningsintäkterna för samma uppsättning produkter när de såldes med rabatt jämfört med när de såldes till fullt pris. Med en nollhypotes (H0) att det inte finns någon skillnad i genomsnittlig försäljningsintäkt mellan rabattförsäljning och icke-rabattförsäljning och som en alternativ hypotes (H1) att det finns en signifikant skillnad i genomsnittlig försäljningsintäkt mellan rabattförsäljning och icke-rabattförsäljning vi drar slutsatsen att den genomsnittliga intäkten från rabatterad försäljning är lägre än den genomsnittliga intäkten från icke-rabatterad försäljning och att det inte finns tillräckligt med bevis för att förkasta nollhypotesen så finns inte några signifikanta bevis för att rabatter påverkar försäljningen. Detta resultat är viktigt för företaget att först och främst känna till för fördelningen av sina resurser, eftersom om rabatterna inte ökar försäljningen avsevärt så kan företaget investera resurser i en strategi som inte är effektiv. Att förstå detta kan hjälpa till att omdirigera ansträngningar och resurser mot mer effektiva försäljningsstrategier. Också om rabatter inte leder till en proportionell ökning av försäljningsvolymen eller lockar nya kunder kan företaget offra vinster utan avkastning på denna investering. Dessutom är det fördelaktigt för företaget att känna till detta resultat för sin lagerhantering eftersom utan en ökning av försäljningen på grund av rabatter kan lagerhanteringen påverkas, vilket leder till överskottslager och en ökning av innehavskostnaderna. Att förstå ineffektiviteten med rabattering kan leda till effektivare lagerhantering.

Den sista delen av studien var att hitta ett konfidensintervall. Jag valde detta för den genomsnittliga totala försäljningen per beställning i kategorin cyklar ("Bikes") med Pythons SciPy-bibliotek. Resultatet visas i diagrammet nedan:



Resultatet visade att vi kan vara 95 % säkra på att den verkliga genomsnittliga totala försäljningen per beställning för cyklar kategorin ("Bikes") ligger någonstans mellan 4985.20 och 5320.90 vilket är viktigt för företaget att veta antingen för att veta vad man kan förvänta sig i form av genomsnittliga intäkter från en enda order eller för att bestämma prissättning och marknadsföringsstrategier. Detta intervall kan fungera som en referenspunkt. Om företaget till exempel introducerar en ny prismodell eller strategi kan de mäta dess inverkan på detta konfidensintervall för att se om intervallet har förändrats avsevärt. Men intervall hjälper också till att prognostisera efterfrågan och hantera lagernivåer och är i allmänhet ett nyckelelement i intäktsprognoser.

Vad jag skulle rekommendera är att företaget ska försöka få recensioner av sina produkter från sina kunder med hjälp av olika strategier antingen genom att skicka lite e-post eller meddelande eller genom att använda något incitament som en liten rabatt eller belöningspoäng eller genom att använda sociala medier nätverk. Jag skulle också föreslå företaget att särskilt studera det faktum att rabatter inte nämnvärt påverkar försäljningen (i kategorin cyklar). Kanske har någon felaktig eller otillräcklig rabattstrategi tillämpats så måste ändras den och hittas mer effektiva sätt att öka försäljningen och förbättra det ekonomiska resultatet. Om samma resultat fortsätter kanske företaget ska få en konkurrensfördel genom att särskilja andra faktorer som produktkvalitet, productsegenskaper eller kundservice.

4. Executivesummary

Jag anser att alla slutsatser som kom från dataanalysen är viktiga. Man kan inte bortse från det faktum att det inte finns några produktrecensioner, att produkten med flest recensioner är den med lägst ‘’AverageRating’’ eller att de produkter som hade högst försäljning var alla i cykelkategorin indikerad med den starka symbolen i företaget. Men det viktigaste tycker jag är resultatet att rabatterna inte verkar påverka försäljningen i denna kategori. Detta sågs även initialt, i studiens första fas, där produkten med ID 753 hade högst försäljning för år 2011 trots att det inte fanns någon rabatt. Efter en statistisk analys av urvalet visade resultatet att rabatter inte nämnvärt påverkar försäljningen. Jag tror att detta är en mycket viktig punkt och av avgörande betydelse för ett företag att känna till eftersom det påverkar hela dess verksamhet, från de strategier det kommer att följa, fördelningen av dess resurser, förvaltningen av dess reserver till lönsamhet, d.v.s. dess existens.

5. Datum för muntlig presentation

Jag skickade den inspelade filen fredagen den 5 januari.

# 3.Reflektion på eget arbete

1.Utmaningar du haft under arbetet samt hur du hanterat dem.

Att lära mig SQL var för mig en mycket intressant kurs och något mycket användbart i det jag vill ägna mig åt som Data Scientist. Utmaningarna jag mötte i den här kursen var definitivt många. För det första var tiden en utmaning för mig, den tidsperiod jag hade för att genomföra den här kursen, om man tar hänsyn till den mänskliga faktorn. Jag tror att en månad inte räcker, speciellt om någon vill fördjupa sig i SQL och om han aldrig tidigare kommit i kontakt med detta programmeringsspråk. Det tar tid att förstå begreppen i SQL då vissa begrepp är mer komplexa. Och naturligtvis är det ännu svårare och mer tidskrävande att tillämpa teoretiska begrepp i praktiken och att lösa vissa uppgifter. För att hantera dessa svårigheter (förutom den tid då jag var väldigt stressad) letade jag efter ytterligare material som till exempel STACK OVERFLOW och några onlineseminarier om YOUTUBE för en bättre förståelse.

2.Vilket betyg du anser att du skall ha och varför.

Betyget jag förväntar mig ligger någonstans mellan G och VG. Sanningen är att jag efter mitt deltagande i de onlineprogrammerade kurserna och det personliga arbetet med det befintliga och extra materialet upptäcker att mina färdigheter har förbättrats och jag tror att jag lyckas lösa och korrekt dokumentera de uppgifter som har tilldelats mig genom att exportera korrekt och dokumenterade logiska resultat och slutsatser.

3.Tips du hade ”gett till dig själv” i början av kursen nu när du slutfört den.

Om jag skulle börja kursen nu med mer erfarenhet och kunskap om resultatet så skulle jag börja från första minut med att leta efter material på internet och jämföra tillämpningen av begreppen i olika övningar och på så sätt stärka mitt kritiska tänkande som jag anser att vara en mycket viktig och värdefull resurs i denna hantering av programmering och specifikt med databaser och SQL. Det vill säga, det räcker inte med att man tillämpar kommandona, utan att man vet hur man läser och analyserar dem, det vill säga för att kunna förstå vad som döljer sig bakom dem och vad de egentligen betyder, samt om de går att ge resultat när de väljs ut att implementeras. Och det är definitivt nödvändigt att fördjupa vissa termer och begrepp eftersom man genom att använda dem kan dra mer korrekta och säkra slutsatser.