R

Prisprediktion av Volkswagenbilar med Regressionsanalys

Gustav Jeansson

EC Utbildning

Kunskapskontroll – R

2025-04

Abstract

The market for used cars is large and growing, making it valuable for both buyers and sellers to understand the factors driving pricing. This study aims to develop and evaluate regression models for predicting the selling price of used Volkswagen cars in Sweden, utilizing data manually collected from the online marketplace Blocket.

Regression analysis, a technique from statistics and machine learning, was employed to model the relationship between car attributes and price. Data on selling price, mileage, model year, horsepower, fuel type, and other features was gathered and subjected to extensive cleaning to handle errors, missing values, and outliers. The dataset was split into training (60%), validation (20%), and test (20%) sets.

Three linear regression models were developed and compared: a standard linear model, a log-linear model (using the logarithm of price), and a log-linear model incorporating interaction terms (specifically between model year and mileage, and horsepower and fuel type). Model performance was primarily assessed using the Root Mean Squared Error (RMSE) on the validation data, with predictions from log-linear models back-transformed to the original price scale. Diagnostic plots and Variance Inflation Factor (VIF) were used to evaluate model assumptions and multicollinearity.

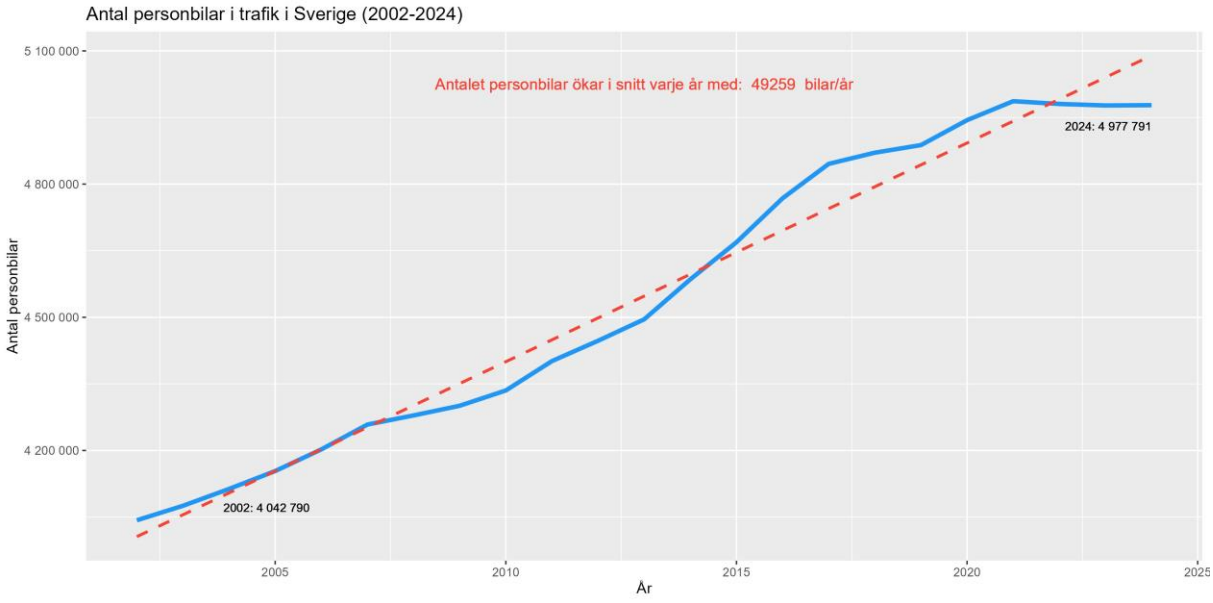
The log-linear model with interaction terms demonstrated the best predictive performance on the validation set. This model achieved a final RMSE of approximately 22,700 SEK on the independent test set, which represents around 12% of the average car price in the cleaned dataset. Diagnostic analysis indicated that the log transformation improved the linearity, homoskedasticity, and normality assumptions compared to the linear model, although some heteroskedasticity persisted, and interaction terms introduced high multicollinearity (high VIF values) which complicates coefficient interpretation but did not hinder predictive accuracy. Key price drivers identified included model year, mileage, horsepower, seller type, and specific car models, with evidence of interaction effects modifying the impact of mileage by age and horsepower by fuel type.

In conclusion, a regression model capable of providing reasonable price estimations for used Volkswagen cars was developed. While prediction intervals for individual cars were relatively wide, reflecting inherent variability, the model serves as a useful tool for price indication and could form the basis for a web-based estimation service.

Innehållsförteckning

1 Inledning

Marknaden för begagnade bilar är stor, det finns t.ex. 144 000 bilar till salu på blocket. Att förstå vad som driver priset på en begagnad bil är värdefullt både för köpare och säljare. Statistiska centralbyrån (SCB) visar att antalet personbilar i trafik i Sverige har ökat stadigt under de senaste åren. Ett rimligt antagande är att begagnatmarknaden kommer fortsätta att växa och till en följd av det kan det vara bra att skapa en modell som kan förutsäga bilpriser på begagnade bilar.



Figur 1: Antal personbilar i trafik i Sverige (2004-2024) (SCB, statistikdatabasen).

Regressionsmodellering som är en teknik inom statistik och maskininlärning, kan användas för att skapa modeller som predikterar ett pris baserat på en bils egenskaper. Genom statistisk inferens kan vi dessutom få insikt i vilka faktorer som signifikant påverkar priset.

Syftet med denna rapport är att utveckla och utvärdera regressionsmodeller för att prediktera försäljningspriset på begagnade Volkswagenbilar baserat på data insamlad från Blocket. Målet är att identifiera en modell som ger tillförlitliga prisprediktioner, vilket t.ex. skulle kunna ligga till grund för en webbtjänst där användare kan få en uppskattning av sin bils värde.

2 Teori

2.1 Regressionsmodeller

Regressionsanalys ("Regression analysis," 2025) används för att modellera och förstå sambandet mellan en beroende variabel (den vi vill förutsäga, i vårt fall försäljningspris) och en eller flera oberoende variabler (prediktorer, t.ex. miltal, årsmodell mm.). Målet är att skapa en funktion som beskriver hur prediktorerna påverkar den beroende variabeln. I regressionsproblem försöker man förutsäga ett kontinuerligt värde, till skillnad från klassificeringsproblem där man förutsäger en kategori. Exempel på regressionsproblem är att förutsäga huspriser eller en persons lön.

2.2 Linjär Regression

Den multipla linjära regressionsmodellen är en vanlig utgångspunkt och antar ett linjärt samband mellan prediktorerna ($x_1, x_2, \dots, x_p$) och den beroende variabeln (Y):

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

β_0 = förväntat värde på Y när alla x är noll

β_i = genomsnittliga förändringen i Y när x_i ökar med en enhet, förutsatt att alla andra prediktorer hålls konstanta

ϵ = det slumpmässiga felet eller variationen som modellen inte fångar upp

2.3 Log-transformering

Ibland är sambandet mellan variablerna inte linjärt, eller så uppfyller inte modellens residualer (felen) de nödvändiga antagandena. Särskilt vid prisdata är det vanligt med en högersned fördelning och ökande varians (heteroskedasticitet). En vanlig åtgärd är att logaritmera den beroende variabeln (EducationTopicsExplained, 2024):

$$\log(Y) = \beta 0 + \beta 1 x1 + \beta 2 x2 + \cdots + \beta p x p + \epsilon$$

Detta kan ofta stabilisera variansen och göra residualerna mer normalfördelade, vilket leder till att modellens antaganden är bättre uppfyllda. Prediktioner från denna modell görs på log-skalan och måste därför återtransformeras med exponentialfunktionen (e) för att tolkas på den ursprungliga prisskalan.

$$Y = e^{\beta 0 + \beta 1 x1 + \beta 2 x2 + \cdots + \beta p x p + \epsilon}$$

2.4 Interaktionstermer

En linjär modell antar att effekten av en prediktor (x_i) på Y är densamma oavsett värdet på andra prediktorer. Ibland är detta inte realistiskt. En interaktionsterm ($x_i : x_j$) kan läggas till modellen för att tillåta effekten av x_i att variera beroende på nivån av x_j. Formeln blir då:

$$\log(Y) = \beta 0 + \cdots + \beta i x i + \beta j x j + \beta i j (x i \times x j) + \cdots + \epsilon$$

βij = fångar den här samverkanseffekten

2.5 Modellantaganden och Diagnostik

För att statistisk inferens (t.ex. tolkning av p-värden och konfidensintervall för beta-koefficienter) från en linjär regressionsmodell ska vara tillförlitlig, bör vissa antaganden vara uppfyllda:

1. **Linjäritet:** Sambandet mellan prediktorerna och den (eventuellt transformerade) beroende variabeln ska vara approximativt linjärt. Undersöks genom att plotta *Residualer vs Fitted-värden*. Mönster tyder på problem.
2. **Homoskedasticitet (konstant varians):** Residualernas varians ska vara konstant över alla nivåer av predikterade värden. Undersöks i *Residualer vs Fitted*- eller *Scale-Location*-plotten. En trattform tyder på heteroskedasticitet. Breusch-Pagan-testet (`bptest`) testar detta formellt.
3. **Normalfördelade fel:** Residualerna ska vara approximativt normalfördelade. Undersöks med *Q-Q-plott* (punkterna ska följa linjen) och *histogram* över residualerna.
4. **Multikollinearitet:** Prediktorerna bör inte vara för starkt korrelerade med varandra. Hög multikollinearitet gör koefficienterna instabila och svårtolkade. Undersöks med *Variance Inflation Factor (VIF)*. Värden över 5 eller 10 brukar anses problematiska.
5. **Inflytelserika punkter (Outliers/Leverage):** Enskilda datapunkter kan ha oproportionerligt stor påverkan på modellen. Identifieras med *Residuals vs Leverage*-plotten och *Cook's Distance*.

Även om inte alla antaganden är helt uppfyllda kan modellen fortfarande vara användbar för prediktion, men tolkningen av koefficienter och p-värden måste göras med försiktighet. Om heteroskedasticitet misstänks kan robusta standardfel användas via `coeftest` och `vcovHC`.

2.6 Modellutvärdering (RMSE)

För att jämföra prediktionsförmågan hos olika regressionsmodeller används ofta RMSE (Root Mean Squared Error). Det mäter det genomsnittliga felet mellan modellens predikterade värden och de faktiska värdena, uttryckt i samma enhet som den beroende variabeln

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

y_i = det faktiska värdet

$\hat{y}_i$ = det predikterade värdet för observation

2.7 Prediktionsintervall

När en modell används för att prediktera utfallet för en ny, enskild observation, är vi ofta intresserade av ett prediktionsintervall. Detta intervall anger, med en viss konfidensnivå (oftast 95%), inom vilket det framtida värdet förväntas ligga. Det tar hänsyn till både osäkerheten i skattningen av regressionslinjen och den inneboende slumpmässiga variationen hos enskilda observationer. Detta skiljer sig från ett konfidensintervall, som endast beskriver osäkerheten kring det genomsnittliga utfallet för observationer med givna prediktionsvärden.

3 Metod

3.1 Datainsamling

Datan för denna studie samlades in manuellt från annonser för begagnade bilar på Blocket. För att begränsa oss och får ett bra så bra dataset som möjligt så bestämde vi oss för att samla in data för ett

specifikt bilmärke. Vi valde Volkswagen som det fanns gott om annonser för, ca 17 500 st fanns på blocket vid tillfället för datainsamlingen.

Datainsamlingen genomfördes i grupp där vi till att börja med började med att begränsa oss till vilken data vi skulle samla in och hur våra sökkriterier skulle se ut för att filtrera ut bilannonserna. Vår filtrering var:

• årsmodell: 2000 ≤ x ≤ 2022
• Märke = Volkswagen
• Modeller = personbilar
• Biltyp ≠ Yrkesfordon (dvs inte yrkesfordon)
• Ägandeform = Köpa
• Hästkrafter = HK
• Sortering = Senaste, inte betalda placeringer
• Välj de 100 första observationerna (annonserna)

Vi begränsade oss bl.a. till att de nyaste bilarna skulle vara från 2022 för att vi vill skapa en modell som kan prediktera begagnade bilar. Om nyare bilar med ett högre pris tas med kan de påverka modellerna negativt var vår tanke. Vi valde alla olika regioner att samla in data från för att på så sätt slippa att vi samlade in dubbletter och alla kunder jobba i egen takt. Varje person skulle samla in 100 observationer vadera.

Vi valde att samla in försäljningspris, säljare, växellåda, miltal, biltyp, drivning, hästkrafter, färg, datum i trafik, märke, modell och region. Detta sparades i en GoogleSheets-fil som sedan importerades i R för datatvätt.

3.2 Datatvätt

Den data vi samlade in innehöll en hel del felaktigheter så som saknade värden, felinmatningar, specialtecken mm. Dessa behövde hanteras innan modellering. Observera att det är en iterativ process där fel upptäcks efterhand. Datatvätten gjordes i R och några av stegen som utfördes var:

- Konvertering av text till gemener för att standardisera kategoriska värden.
- Borttagning av alla rader som innehöll tomma värden/NA-värden
- Rättat till stavfel
- Borttagning av prefix som ljus och mörk i färgkategorin (ljusblå blev blå). Senare togs hela färgkolumnen bort för att den inte ansågs vara av så stor betydelse för modellen.
- Identifiering och borttagning av uppenbara outliers t.ex. ett extremt högt pris (2 650 000 kr), orimligt högt miltal (249 764 mil) och ett modellår vi valde att inte ha med (2025) hade kommit med.
- Borttagning av kolumner som märke (alla bilar är ju Volkswagen) och datum i trafik som är starkt korrelerad med årsmodell.

Senar i modellfilen gjordes även mer tvätt av datan så som:

- Skapande av en ny variabel `model_reduced` där bilmodeller med färre än 20 observationer i datasetet grupperades i kategorin "övrigt". Dessa togs senare bort för att minska antalet nivåer (variabler) i bilmodellsvariabeln.
- Justerade specifika bilar som t.ex. kunde ha ett felaktigt pris, eller vara av fel modell.
- Tog även bort bilar som var äldre än 2010 för att försöka förbättra modellen.

Innan datatvätten påbörjades fanns det 1204 observationer/bilar, efter datatvätten var det kvar 1182 observationer kvar. Efter iterationer i modellskapandet och t.ex. borttagande av bilar äldre än 2010 så var det 1012 observationer kvar att träna modellerna på.

3.3 Datadelning

För att kunna utvärdera modellernas generaliseringsförmåga på osedd data delades datasetet upp i träning (60%), validering (20%) och test (20%). Uppdelningen gjordes slumpmässigt. Träningsdatan används sedan för att träna modellerna och valideringsdatan för att jämföra modellerna och välja den bästa. Testdatan används för en slutlig utvärdering av den valda modellen.

3.4 Modellering

Tre olika linjära regressionsmodeller tränades på träningsdatan:

1. **Modell 1 (Linjär):** En multipel linjär regressionsmodell med `selling_price` som beroende variabel och alla de kvarvarande variabler som prediktorer. (`selling_price ~ .`).
2. **Modell 2 (Log-Linjär):** Samma prediktorer som Modell 1, men med logaritmen av `selling_price` som beroende variabel (`log(selling_price) ~ .`).
3. **Modell 3 (Log + Interaktion):** Utgick från Modell 2 men la även till interaktionstermer mellan `model_year` och `mileage` samt mellan `horsepower` och `fuel_type` (`log(selling_price) ~ . + model_year:mileage + horsepower:fuel_type`).

3.5 Modellval

Modellernas prediktiva prestanda jämfördes primärt med RMSE, beräknat på valideringsdatan. För log-modellerna (Modell 2 och 3) återtransformerades prediktionerna till den ursprungliga prisskalan innan RMSE beräknades. Detta för att kunna jämföras med den linjära modellen (Modell 1). Modellen med lägst RMSE på valideringsdatan valdes som den "bästa" modellen. Denna valda modell utvärderades sedan en sista gång på testdatan för att få ett estimat på dess generaliseringsförmåga på helt ny data.

3.6 Diagnostik av Regressionsmodeller

För varje tränad modell genererades diagnostikplottar som visuellt inspekterades för att utvärdera modellernas linearitet och homoskedasticitet (i Residuals vs Fitted Values och Scale-Location-plottarna), normalitet i residualernas fördelning (i Q-Q-plotten) samt identifiera potentiellt inflytelserika punkter (Residuals vs Leverage-plotten med Cook's distans). Utöver dessa visuella analyser kvantifierades multikollinearitet med hjälp av Variance Inflation Factor (VIF) och heteroskedasticitet testades genom Breusch-Pagan-testet.

3.7 Prediktion av ny data (begagnade bilar)

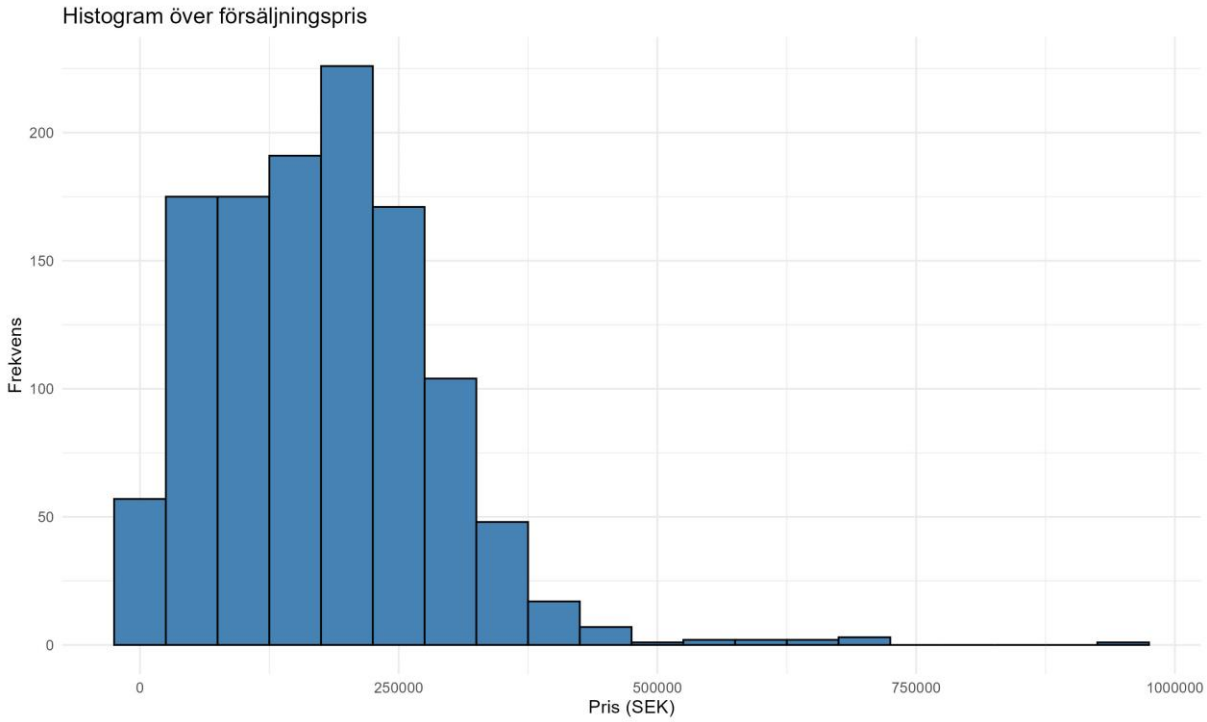
Den bästa modellen användes slutligen för att prediktera försäljningspriset för fyra begagnade Volkswagenbilar inklusive ett 95%-igt prediktionsintervall.

4 Resultat och Diskussion

4.1 Datainsamling

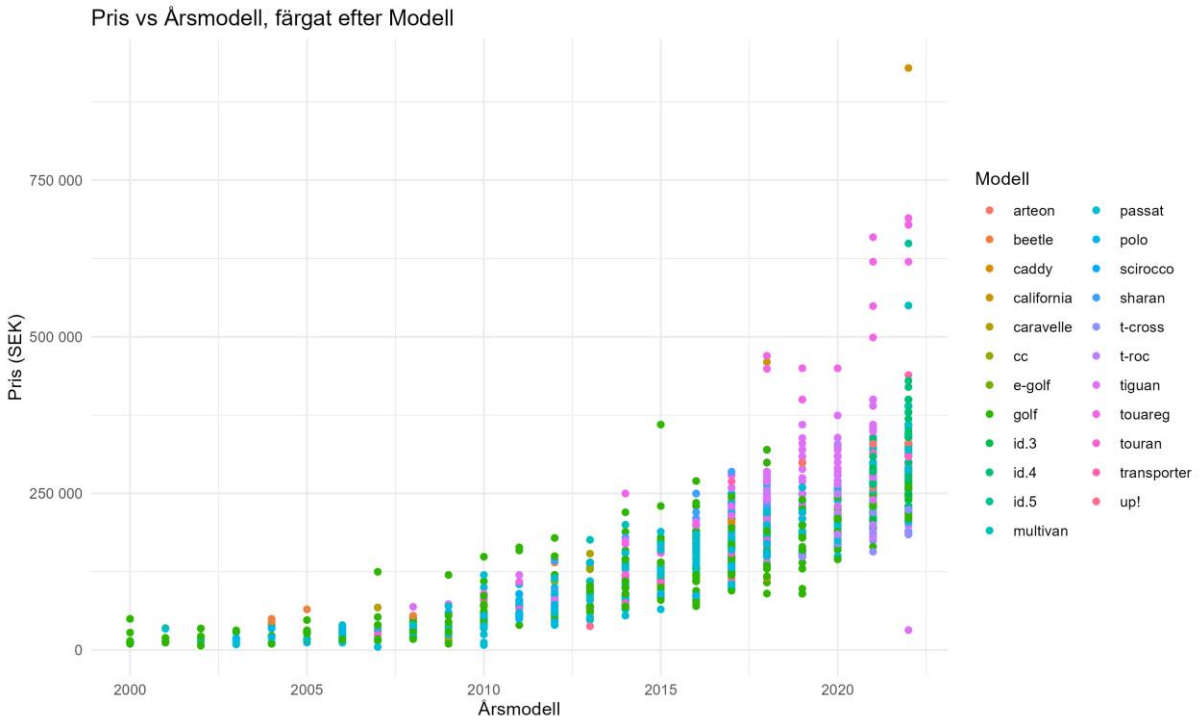
Lärdomar från datainsamlingen är att om det görs manuellt i grupp så göra alla lite olika trotts att vi bestämde en grundstruktur för hur vi skulle samla in datan. Det är dessutom lätt att det blir något litet fel t.ex. att det blir en siffra för mycket eller för lite och att det kan smyga sig in oväntade specialtecken. Detta gjorde att datatvättandet blev betydande och man hittar fler fel ju mer man jobbar med datan. Det är dessutom nästan helt omöjligt att hitta alla fel innan det är dags att bygga sina modeller. Felen upptäcks helt enkelt efterhand. Innan man har gjort sina modeller är det också svårt att veta vilka variabler som är av betydelse så förmodligen har man dessutom samlat in för mycket data.

4.2 Översikt av data



Figur 2: Histogram för bilarnas pris i det otvättade datasetet.

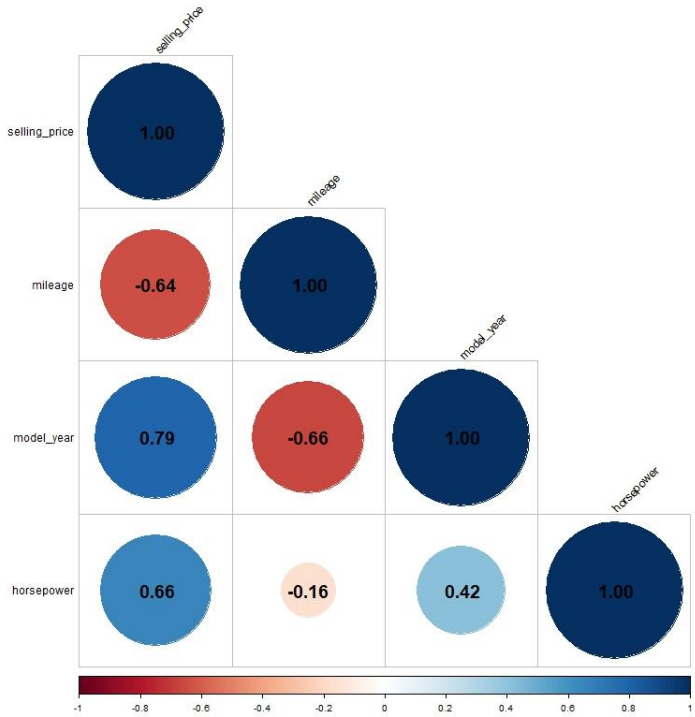
I histogrammet kan man se att det finns en lång svans av mycket dyrare bilar, alltså det finns potentiellt risk för att ett fåtal bilar kan ha stor påverkan på försäljningspriset.



Figur 3: Prisfördelning efter årtal, färgkodat efter modell.

I figuren ovan kan man se antydan till att olika bilmodeller är olika dyra (en viss färg ligger högre i pris). Färgkodningen kan också visa när en ny bilmodell introduceras på marknaden. Från figuren kan man anta att vissa bilmodeller kan komma att har en större eller mindre påverkan på modellen.

4.3 Korrelationer



Figur 4: Korrelationsmatris mellan de numeriska variablerna i datasetet.

Korrelationsmatrisen ger en översikt över de linjära sambanden mellan de numeriska variablerna som undersökts. Korrelationer kan ge insikter om hur variablerna samvarierar med varandra.

Det finns en starka positiva korrelationen (0.79) mellan försäljningspriset och årsmodellen. Detta indikerar att det finns en stark tendens att nyare bilar har ett högre försäljningspris. Detta samband är ganska intuitivt då nyare fordon generellt sett har högre marknadsvärde.

Det finns också en relativt stor korrelation (0.66) mellan försäljningspriset och hästkrafter. Detta tyder på att bilar med högre motorstyrka i regel tenderar att säljas till ett högre pris. Även detta är ett rimligt samband då prestanda ofta är en prispåverkande faktor.

Det finns också en negativ korrelation (-0.64) mellan försäljningspriset och miltalet. Detta innebär att bilar med högre körsträcka tenderar att ha ett lägre försäljningspris, vilket också är en förväntad relation då högre miltal indikerar mer slitage och potentiellt ett lägre andrahands värde.

Det finns även en negativ korrelation (-0.66) mellan årsmodell och miltal. Detta tyder på att nyare bilar generellt sett har lägre miltal, vilket är logiskt eftersom de generellt set inte hunnit köras så långt.

Slutligen kan noteras en svagare positiv korrelation (0.42) mellan årsmodell och hästkraft. Detta antyder en liten tendens till att nyare bilar har något fler hästkrafter i genomsnitt, även om sambandet inte är jättestarkt.

Det kan vara av vikt att notera att variabler som inte uppvisar starka korrelationer med varandra potentiellt kan bidra med unik information till de statistiska modellerna. De representerar aspekter av datan som inte nödvändigtvis följer samma linjära trender som de mer korrelerade variablerna. De starkare korrelationerna som identifierats här pekar på viktiga linjära samband som sannolikt kommer att spela en betydande roll i förmågan att prediktera försäljningspriset. Sedan till kommer de icke-numeriska variablerna som inte kan plottas i en korrelationsmatris men som kan ha större eller mindre påverkan på modellerna.

4.4 Modellresultat (Träning)

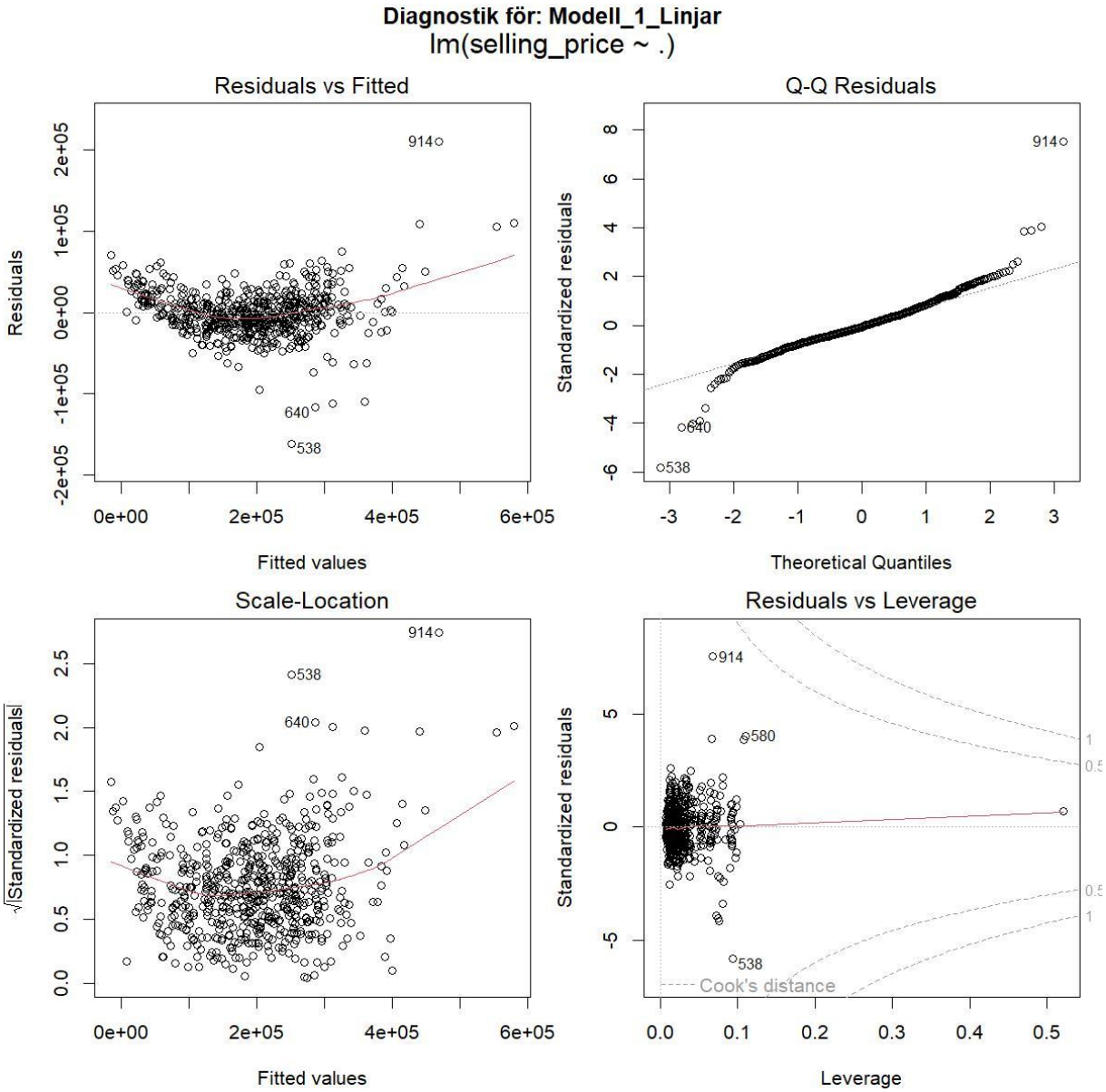
Tre regressionsmodeller tränades på träningsdatan (N=607):

- **Modell 1 (Linjär):** Linjär regression med `selling_price` som beroende variabel och alla tillgängliga prediktorer. Modellen uppnådde ett justerat R^2 på 0.9071 vilket betyder att ca 90% av variansen i försäljningspriset förklaras av de oberoende variablerna.
- **Modell 2 (Log-Linjär):** Linjär regression med `log(selling_price)` som beroende variabel och samma prediktorer. Modellen uppnådde ett justerat R^2 på 0.9087, vilket indikerar en marginellt bättre förklaringsgrad på log-skalan jämfört med den linjära modellen på originalskalan.
- **Modell 3 (Log + Interaktion):** Utökning av Modell 2 med interaktionstermer (`model_year:mileage` och `horsepower:fuel_type`). Modellen uppnådde ett justerat R^2 på 0.922, vilket är den högsta förklaringsgraden bland de testade modellerna.

4.5 Teoretiska antaganden från diagnostikplottar

För att bedöma modellernas tillförlitlighet och förstå eventuella begränsningar granskades diagnostiska plottar och VIF-test utfördes.

4.5.1 Modell 1 (Linjär):



Figur 5: Diagnostikplottar för modell 1 (Linjär).

4.5.1.1 Residuals vs Fitted:

Det finns en viss antydan till ett trattformat utseende (från vänster till höger) på punkterna. Spridningen ökar för högre predikterade priser och detta tyder på heteoskedasticitet, vilket betyder att variansen inte är konstant i residualerna (feltermen). Det finns också en böjning (se trendlinjerna) på punkterna vilket tyder på icke-linearitet.

4.5.1.2 Q-Q-plot

Punkterna följer inte den räta linjen i svansarna vilket tyder på att residualerna inte är perfekt normalfördelade.

4.5.1.3 Scale-Location

Om den röda trendlinjen är relativt horisontell antyder det homoskedasticitet. En tydlig trend (stigande eller fallande linje) indikerar heteroskedasticitet vilket vi har här.

4.5.1.4 Residuals vs Leverage

Punkter långt till höger har hög leverage (ett mått på hur extremt en observations värden på de oberoende variablerna är jämfört med medelvärdena för dessa variabler). Hur lång från y=0 beskriver storleken på residualerna och göra det lättare att identifiera outliers (långt från 0 större sannolikhet att det är en outlier). De streckade röda linjerna representerar olika nivåer av Cook's distans. Cook's distans är ett mått på det totala inflytandet en observation har på alla de uppskattade regressionskoefficienterna. Observationer med höga Cook's distansvärden anses vara inflytelserika. Plotten ger oss information om vilka punkter som kan vara värda att studera vidare och eventuellt ta bort eller hitta fel i ursprungsdatan (inmatningsfel). Observera att det är en iterativ process och datan har processats flera gånger vilket betyder att de grafer som visas inte nödvändigtvis såg lika dana ut från början.

4.5.1.5 Breusch-Pagan-test (bptest)

Om p-värdet är litet (vanligtvis mindre än ett valt signifikansnivå t.ex. 0.05), förkastar vi nollhypotesen. I detta fall är p-värdet < 2.2e-16, vilket är mindre än 0.05. Detta innebär att det finns statistiskt signifikanta bevis för att variansen i feltermerna inte är konstant. Detta bekräftar att vi har heteroskedasticitet.

4.5.1.6 VIF (Variance Inflation Factor)

VIF mäter hur mycket variansen av en uppskattad regressionskoefficient ökar på grund av multikollinearitet (korrelationer mellan prediktorvariabler) i modellen. Ett högt VIF-värde indikerar att den prediktorvariabeln är starkt korrelerad med andra prediktorer, vilket kan göra koefficientuppskattningarna instabila och svåra att tolka.

GVIF (Generalized Variance Inflation Factor) är en generalisering av VIF för prediktorer med fler än en frihetsgrad

GVIF^(1/(2*Df)) är en justerad version av GVIF som kan vara lättare att tolka och jämföra med VIF för enskilda termer (där Df=1). Värden över ca 2-3 för denna justerade GVIF kan indikera problem. Värden mellan 1-2 tyder på viss korrelation men är oftast inte ett problem. Här ser vi att `fuel_type` (1.88) och `model_reduced` (1.25) har relativt låga justerade GVIF.

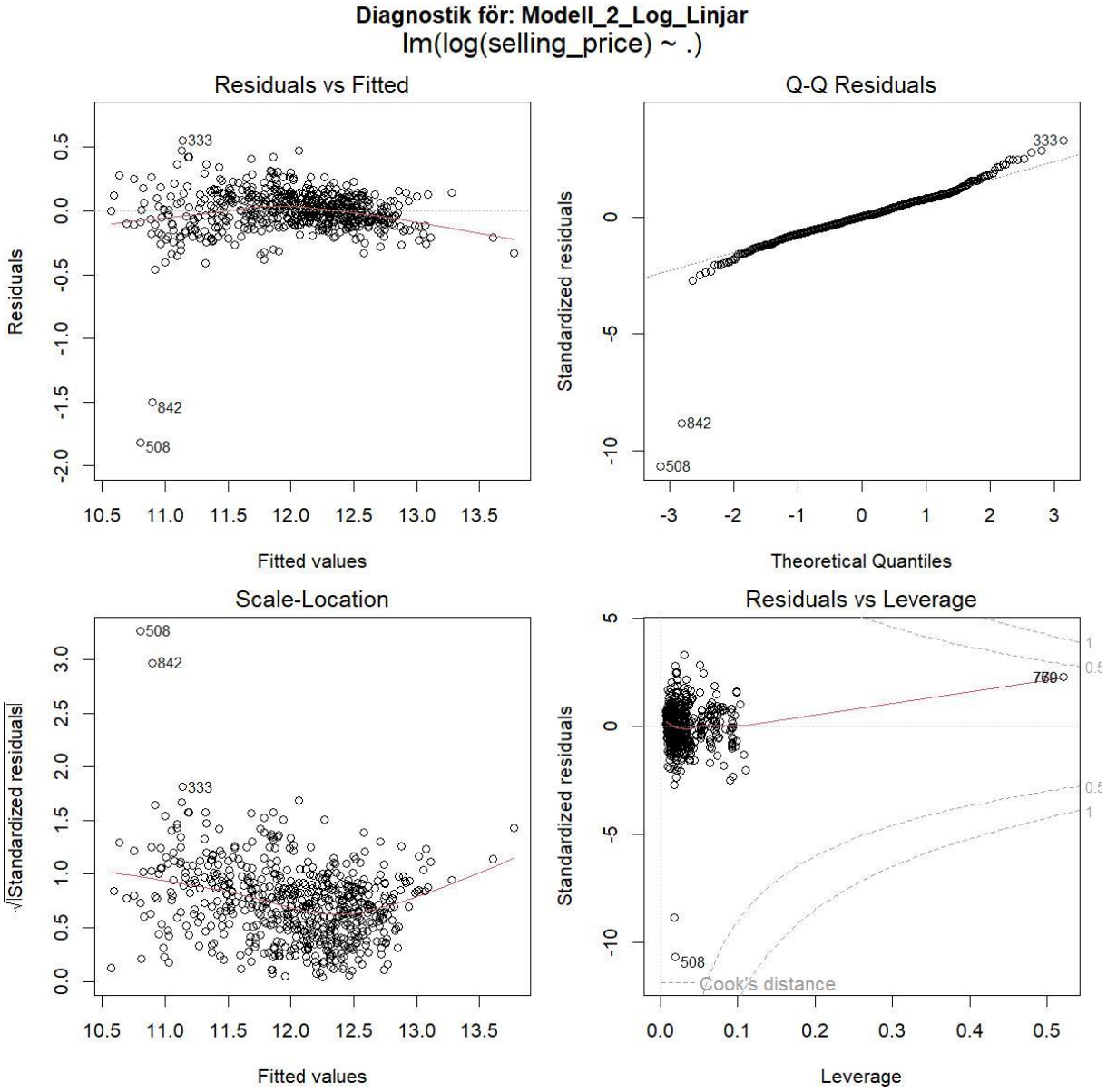
```
Diagnostik för: Modell 1 (Linjär)
 VIF:
                    GVIF Df GVIF^(1/(2*Df))
seller          1.213466  1        1.101574
fuel_type      44.044851  3        1.879241
mileage         2.463898  1        1.569681
model_year      2.508840  1        1.583932
horsepower      2.476850  1        1.573801
model_reduced  84.459918 10        1.248339
```

Figur 6: VIF-värden för modell 1 (Linjär).

4.5.1.7 *Förbättringsåtgärder*

De tydliga tecknen på heteroskedasticitet och potentiell icke-normalfördelning i residualerna för Modell 1 motiverar ett försök att transformera den beroende variabeln. En logaritmisk transformation (`log(selling_price)`) är ofta en standardåtgärd för prisdata för att stabilisera variansen och göra fördelningen mer symmetrisk.

4.5.2 Modell 2 (Log-Linjär)



Figur 7: Diagnostikplottar för modell 2 (Log-Linjär).

4.5.2.1 *Residuals vs Fitted*

Spridningen av residualerna ser nu jämnare ut och trendlinjen är mer horisontell men fortfarande en viss böjning. Finns fortfarande risk för en viss icke- linearitet.

4.5.2.2 *Q-Q-plot*

Punkterna ligger närmare den räta linjen och residualerna är där med mer normalfördelade efter transformationen.

4.5.2.3 *Scale-Location*

Trendlinjen är jämnare än i modell 1 men det finns fortfarande en böjning vilket kan tyda heteroskedasitet.

4.5.2.4 *Residuals vs Leverage*

Här kan vi se punkterna 770, och 508 som kan var värda att studera extra. Generellt är fler punkter längre från Cook's distans linjerna.

4.5.2.5 *Breusch-Pagan-test (bptest)*

Även om p-värdet (0.0053) fortfarande är signifikant (< 0.05), vilket indikerar att viss heteroskedasticitet kvarstår även på log-skalan. Dock tyder det på att log-transformeringen kraftigt minskade problemet med heteroskedasticitet.

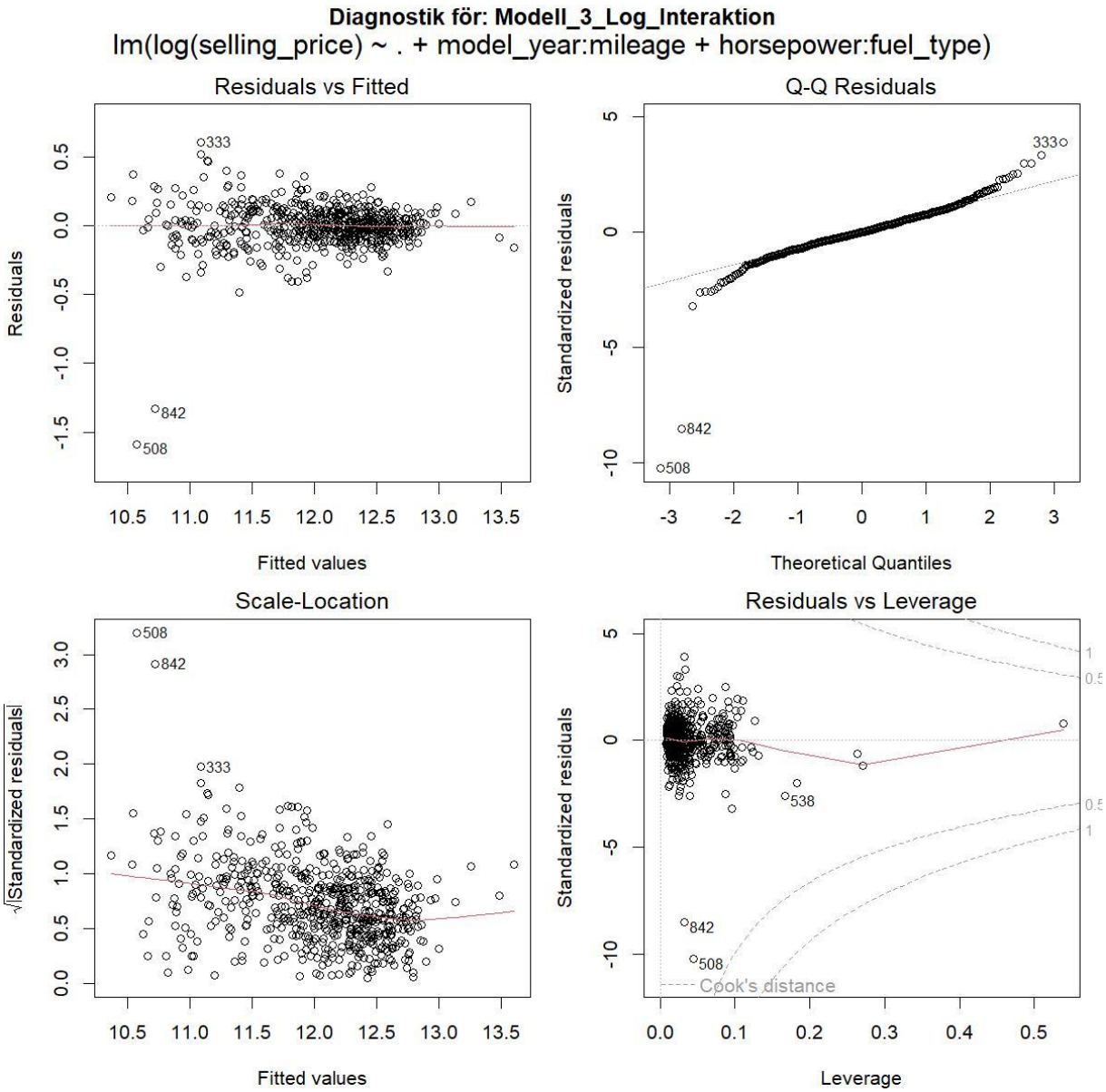
4.5.2.6 *VIF*

Samma som för Modell 1, eftersom prediktorerna är desamma.

4.5.2.7 *Förbättringsåtgärder*

Även om Modell 2 är en klar förbättring jämfört med Modell 1, finns det fortfarande utrymme för att förbättra modellens förklaringsgrad och potentiellt fånga upp mer komplexa samband. Att lägga till interaktionstermer kan vara ett sätt. Interaktionstermer läggs till mellan variabler som logiskt sett kan påverka varandra, såsom `model_year:mileage` (värdeminskningstakten kan bero på ålder) och `horsepower:fuel_type` (effekten av hästkrafter skulle kunna skilja sig mellan bränsletyper).

4.5.3 Modell 3 (Log + Interaktion):



Figur 8: Diagnostikplottar för modell 3 (Log + Interaktion).

4.5.3.1 Residuals vs Fitted

Spridningen är ingen större skillnad mot modell 2, däremot finns det inte längre någon böjning i trendlinjen.

4.5.3.2 Q-Q-plot

Återigen väldigt lik modell 2, eventuellt något sämre.

4.5.3.3 Scale-Location

Även här är skillnaden mot modell 2 marginell.

4.5.3.4 Residuals vs Leverage

Eventuellt något sämre resultat än modell 2 men återigen marginella skillnader.

4.5.3.5 *Breusch-Pagan-test (bptest)*

P-värdet är fortfarande mycket lågt (< 6.8e-06), vilket indikerar att heteroskedasticiteten finns kvar även med interaktionstermerna.

4.5.3.6 *VIF*

VIF-värdena (och GVIF) blir extremt höga för de variabler som ingår i interaktionstermerna (`mileage`, `model_year`, `horsepower`, `fuel_type`) samt för interaktionstermerna själva. Detta är väntat och ett känt problem med interaktioner – de skapar stark multikollinearitet. Detta gör individuell tolkning av koefficienterna för dessa termer mycket svår och osäker, men det behöver inte nödvändigtvis försämra modellens prediktiva förmåga.

	GVIF	Df	GVIF^(1/(2*Df))
seller	1.237748e+00	1	1.112541
fuel_type	1.007732e+04	3	4.647551
mileage	4.790738e+05	1	692.151597
model_year	1.003372e+01	1	3.167604
horsepower	6.661991e+00	1	2.581083
model_reduced	1.408126e+02	10	1.280655
mileage:model_year	4.767292e+05	1	690.455774
fuel_type:horsepower	1.053419e+04	3	4.682023

Figur 9: GVIF-värden för modell 3 (Log + Interaktion).

4.6 Diskussion av antaganden

Log-transformeringen (Modell 2 och 3) förbättrade tydligt antagandena om homoskedasticitet och normalfördelning jämfört med den ursprungliga linjära modellen (Modell 1) även om viss signifikant heteroskedasticitet kvarstod enligt bptest. Modell 3 introducerade kraftig multikollinearitet på grund av interaktionstermerna, vilket försvårar tolkningen av enskilda koefficienter men inte nödvändigtvis modellens prediktionskraft. Inflytelserika punkter identifierades i alla modeller och har behandlats under den iterativa processen. Med tanke på att syftet främst är prediktion, kan Modell 3 fortfarande vara ett starkt alternativ om den presterar bäst på valideringsdata, trots VIF-problemen. Den kvarvarande heteroskedasticiteten gör att p-värden och konfidensintervall bör tolkas med viss försiktighet.

4.7 Modelljämförelse (Validering)

För att välja den bästa prediktiva modellen jämfördes RMSE på valideringsdatan, beräknat på den ursprungliga prisskalan:

	Model	Validation_RMSE
Log + Interaktion	Log + Interaktion	22920.44
Linjär	Linjär	25058.11
Log-Linjär	Log-Linjär	27272.62

Figur 10: Sorterat RMSE på valideringsdata för de tre modellerna.

Modell 3 (Log + Interaktion) uppvisade det klart lägsta RMSE på valideringsdatan, cirka 2 100 kr lägre än den linjära modellen och över 4 300 kr lägre än den rena log-linjära modellen. Detta indikerar att interaktionstermerna, trots att de komplicerade modellen och ökade multikollinearitet, tillförde ett värde för prediktionsnoggrannheten på osedd data.

4.8 Val av modell och testresultat

Baserat på det lägsta validerings-RMSE valdes Modell 3 (Log + Interaktion) som den slutliga modellen.

När denna modell utvärderades på testdatan (N=203), uppnåddes ett slutligt Test-RMSE på ca 22700kr. Detta resultat ligger nära validerings-RMSE (ca 22900 kr), vilket är positivt och tyder på att modellen generaliserar väl och inte är överanpassad till tränings- eller valideringsdatan. Ett genomsnittligt prediktionsfel på ca 22 700 kr kan anses vara relativt högt för lite billigare bilar eller helt okej för dyrare bilar. Medelpriset för de bilar som är kvar (efter datatvätt) ligger på ca 192 000 kr och då utgör prediktionsfelet närmare 12 %.

4.9 Tolkning av Vald Modells koefficienter

En närmare titt på koefficienterna för Modell 3 (Log + Interaktion) kan ge oss en liten uppfattning om hur modellen predikterar värdet för en bil. Modellen är på logskala vilket även koefficienterna är men en approximation (för små β) så leder en enhets ökning i prediktorn till ungefär (β×100)% förändring i försäljningspriset.

mileage (β≈−0.005483):
Approximation: Varje mil en bilen gått minskar priset med ca (−0.005483×100)= −0.55 %. Men denna effekt modifieras av model_year på grund av interaktionstermen mileage:model_year. Effekten är alltså inte konstant.

model_year (β≈0.04604):
Approximation: Varje årsmodell nyare ökar priset med ca 4.6%. Men även denna effekt modifieras av interaktionstermen `mileage:model_year` och är alltså inte heller konstant.

`horsepower` (β≈0.003633):
Approximation: Varje extra hästkraft ökar priset med ca 0.36%. Här modifieras effekten av `fuel_type` och är alltså inte heller den konstant.

`fuel_typeel` (β≈0.40731):
Exakt: $(e^{0,40731} - 1) \times 100 \approx 50{,}3\ \%$. Bränsletypen el ökar priset med ca 50 % jämfört med bensin (som är referensbränsletyp).

`fuel_typeel:horsepower` (β≈0. 002265):
Denna koefficient är signifikant negativ (se figur 12). Den betyder att den positiva effekten av `horsepower` (dvs. $\beta_{\text{horsepower}} \approx 0.0036$) är lägre för elbilar (`fuel_typeel`) jämfört med referensen bensin. Effekten av en extra hästkraft på log(priset) för en elbil blir: $(\beta_{horsepower} + \beta_{interaction}) = (0.003633 - 0.002265) = 0.001368$. Om räknat i procent blir det: $(e^{0,001368} - 1) \times 100 \approx 0{,}14\ \%$ högre pris per extra hästkraft.

```
lm(formula = log(selling_price) ~ . + model_year:mileage + horsepower:fuel_type,
    data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.58645 -0.06896  0.00043  0.08328  0.60833

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                             -8.109e+01  1.181e+01  -6.864 1.72e-11 ***
sellerprivat                            -8.547e-02  1.965e-02  -4.349 1.61e-05 ***
fuel_typediesel                          4.344e-03  6.588e-02   0.066 0.947445
fuel_typeel                              4.073e-01  1.583e-01   2.573 0.010314 *
fuel_typemiljöbränsle/hybrid            -1.507e-02  7.680e-02  -0.196 0.844496
mileage                                 -5.483e-03  7.006e-04  -7.826 2.37e-14 ***
model_year                               4.604e-02  5.853e-03   7.866 1.78e-14 ***
horsepower                               3.633e-03  3.203e-04  11.344  < 2e-16 ***
model_reducedid.3                       -1.431e-01  1.215e-01  -1.178 0.239227
model_reducedid.4                        1.151e-01  1.173e-01   0.982 0.326670
model_reducedpassat                     -1.931e-03  2.121e-02  -0.091 0.927491
model_reducedpolo                       -6.120e-02  3.342e-02  -1.831 0.067588 .
model_reducedsharan                      3.924e-01  5.051e-02   7.770 3.55e-14 ***
model_reducedt-cross                    -6.017e-02  5.400e-02  -1.114 0.265652
model_reducedt-roc                      -7.179e-03  4.348e-02  -0.165 0.868895
model_reducedtiguan                      1.521e-01  2.270e-02   6.698 4.98e-11 ***
model_reducedtouareg                     1.997e-01  5.443e-02   3.669 0.000266 ***
model_reducedtouran                      5.848e-02  4.318e-02   1.354 0.176141
mileage:model_year                       2.704e-06  3.474e-07   7.784 3.21e-14 ***
fuel_typediesel:horsepower               4.899e-04  4.091e-04   1.198 0.231593
fuel_typeel:horsepower                  -2.265e-03  6.107e-04  -3.708 0.000228 ***
fuel_typemiljöbränsle/hybrid:horsepower -3.934e-04  4.410e-04  -0.892 0.372679
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1588 on 585 degrees of freedom
Multiple R-squared:  0.9247,    Adjusted R-squared:  0.922
F-statistic: 342.3 on 21 and 585 DF,  p-value: < 2.2e-16
```

Figur 11: Koefficienterna (Estimate) för modell 3 (Log + Interaktion) (observera värdena är på log-skala).

4.10 Koefficienternas konfidensintervall

Konfidensintervallen kan tolkas som om nollan inte är inkluderad i intervallet kan vi dra slutsatsen att den koefficienten är signifikant positiv eller negativ beroende på vart intervallet ligger. För `mileage:model_year` var intervallet (2.02e-06, 3.39e-06) vilket är helt positivt vilket stärker slutsatsen att interaktionen är signifikant positiv. För fueltypeel`:horsepower` är intervallet signifikant negativt som tidigare diskuterats.

```
Konfidensintervall för koefficienter:
                                                2.5 %        97.5 %
(Intercept)                                -1.042904e+02 -5.788426e+01
sellerprivat                               -1.240659e-01 -4.686882e-02
fuel_typediesel                            -1.250435e-01  1.337320e-01
fuel_typeel                                 9.645848e-02  7.181648e-01
fuel_typemiljöbränsle/hybrid               -1.658974e-01  1.357580e-01
mileage                                    -6.859149e-03 -4.107124e-03
model_year                                  3.454083e-02  5.752984e-02
horsepower                                  3.003926e-03  4.261889e-03
model_reducedid.3                          -3.816309e-01  9.545255e-02
model_reducedid.4                          -1.151949e-01  3.454214e-01
model_reducedpassat                        -4.358018e-02  3.971892e-02
model_reducedpolo                          -1.268327e-01  4.440906e-03
model_reducedsharan                         2.932326e-01  4.916312e-01
model_reducedt-cross                       -1.662213e-01  4.589009e-02
model_reducedt-roc                         -9.256616e-02  7.820759e-02
model_reducedtiguan                         1.074716e-01  1.966529e-01
model_reducedtouareg                        9.278589e-02  3.065709e-01
model_reducedtouran                        -2.632546e-02  1.432908e-01
mileage:model_year                          2.021893e-06  3.386593e-06
fuel_typediesel:horsepower                 -3.135733e-04  1.293347e-03
fuel_typeel:horsepower                     -3.464051e-03 -1.065295e-03
fuel_typemiljöbränsle/hybrid:horsepower    -1.259486e-03  4.726646e-04
```

Figur 12: Konfidensintervall för koefficienterna.

4.11 R^2

Modellens förklaringskraft justerat R^2 var 0.922 vilket innebär att modellen förklarar ca 92% av variansen i logaritmen av försäljningspriset i träningsdatan.

4.12 Prediktioner av bilar

Modell 3 användes för att prediktera priset för fyra exempelbilar:

1. Bil 1: 2021 Golf (Bensin, Företag), 5000 mil, 150 hk
2. Bil 2: 2018 Tiguan (Diesel, Privat), 12000 mil, 110 hk
3. Bil 3: 2017 Tiguan (Diesel, Företag), 17500 mil, 190 hk
4. Bil 4: 2017 Golf (Bensin, Företag), 14978 mil, 111 hk

Återtransformerade prediktionerna i kronor inklusive 95% prediktionsintervall blev enligt figur 13. För Bil 1 predikterades priset till ca 244 200 kr, med ett 95% prediktionsintervall mellan ca 178 300 kr och 334 500 kr. Intervallens bredd är ca 80 000 - 150 000 kr och beskriver osäkerheten i prediktionen för en enskild bil. Osäkerheten beror dels på modellens osäkerhet och dels på den naturliga variationen i bilpriser som inte fångas av modellen.

```
Prediktioner (återtransformerade från log-skala):
       fit       lwr      upr
1 244189.2 178270.46 334482.5
2 166828.5 121231.79 229574.8
3 199545.3 145772.03 273154.8
4 125446.9  91616.97 171768.8
```

Figur 13: Resultat för prisprediktion av begagnade bilar med prediktionsintervall.

5 Slutsatser

Syftet med arbetet var att utveckla en regressionsmodell för att prediktera priset på begagnade Volkswagen-bilar. Tre modeller jämfördes: en linjär modell, en log-linjär modell och en log-linjär modell med interaktionstermer.

Baserat på lägst RMSE på valideringsdatan valdes Modell 3 (Log + Interaktion) som den bästa modellen. Denna modell uppnådde ett slutligt RMSE på 22 700 kr på testdatan.

Modellen identifierade miltal, årsmodell, hästkrafter, säljartyp och specifika bilmodeller som viktiga prisdrivare, samt att det finns samspel mellan vissa av dessa faktorer. Diagnostiken visade att modellens antaganden förbättrades med log-transformering, även om viss heteroskedasticitet kvarstod och interaktionerna skapade hög multikollinearitet (VIF), vilket försvårar tolkningen av enskilda koefficienters exakta inverkan men inte nödvändigtvis försämrade den totala prediktionsförmågan.

När modellen användes för att prediktera priset på nya exempelbilar genererades prediktionsintervall som var relativt breda ca $\pm$ 80 000 kr som mest. Detta känns som att de blev lite väl brett för att man ska vara helt nöjd men det återspeglar osäkerheten i modellen. För syftet att skapa en webbtjänst som ger en uppfattning om värdet kan modellen ändå anses vara användbar för att ge en prisindikation, men användaren bör vara medveten om osäkerhetsintervallet.

Ett fortsatt arbete skulle kunna vara fokusera ytterligare på att förbättra modellen genom att undersöka fler eller andra interaktionstermer och testa alternativa modelltyper.

6 Självutvärdering

1. **Roligast**: Roligast har varit att det är verkliga data som vi har fått samla in och sedan göra en modell över. Jag börjar känna så smått som att det går att göra "verkliga" saker vilket känns kul och motiverande!
2. **Utmaningar**: Time-management är verkligen en stor utmaning. Det är så himla lätt att fastna på något som man tycker är lite extra intressant just i stunden och sedan inser man i efterhand att man kanske la lite för mycket tid på det. För mig var det t.ex. att få till scraping och datatvätt i R. Lärdom är att göra det som måste göras och ta sig vidare om tiden är begränsad -allt kan inte vara perfekt (även om det är svårt att inte försöka...). Dessutom tar det lång tid för mig att skriva rapport, något jag bara behöver var medveten om.
3. **Betyg**: Jag hoppas på VG, det var länge jag tyckte att jag inte hade så bra koll men det kom ju mer jag jobbade med uppgiften.
4. **Lyfta**: Inget jag kommer på i stunden.

7 Teoretiska frågor

1. I en QQ-plot undersöker man om en uppsättning data är följer en viss fördelning, ofta normalfördelningen. Datauppsättningen sorteras och plottas sedan mot normalfördelningens kvantiler. Man har då delat upp normalfördelningkurvan i n+1 lika stora delar (areamässigt). Värdena i normalfördelningskurvan som kan dela upp kurvan i n+1 lika stora delar plottas sedan mot den datauppsättning man har. Om datapunkterna i grafen ligger på en rät linje så kan man anta att en datauppsättning är normalfördelad.

2. I maskininlärning ligger ofta fokus på att göra prediktioner om framtiden. För att göra det använder man sig av mer eller mindre komplexa modeller som kan hitta icke-linjära samband i stora datamängder. Modellerna man använder sig av kan vara lite av svarta lådor och kan alltså vara svåra att tolka hur det faktiskt fungerar. Maskininlärning används oftare när prediktionsnoggrannheten är viktigare än tolkbarheten. Ett exempel skulle kunna vara bildanalys för medicinsk diagnostistik där fokus är att få ett så bra resultat det bara går.

   I statistisk regressionsanalys kan man också göra prediktioner men man kan också göra statistisk inferens. I statistisk inferens försöker man dra slutsatser om en population baserat på ett stickprov. Målet är helt enkelt att generalisera resultatet från stickprovet till hela populationen med beaktande av osäkerheten i dessa generaliseringar. När man gör regressionsanalys så är målet att dra slutsatser om hur de oberoende variablerna påverkar den beroende variabeln i populationen baserat på ett stickprov. Då kan man t.ex. uppskatta hur stor en effekt en oberoende variabel har på den beroende variabeln. Man kan också beräkna konfidensintervall för regressionskoefficienterna vilket ger oss ett mått på hur stor osäkerheten är i uppskattningarna. Samt göra hypotesprövningar för att testa samband mellan de oberoende variablerna. På detta sätt kan man få en djupare förståelse för vilka variabler som har störst påverkan på resultatet. Exempel på när det kan användas i verkligheten är hur marknadsföringsinsatser påverkar försäljningen. Då kan man t.ex. undersöka vilken typ av marknadsföring som har störst påverkan på försäljningen.

3. Konfidensintervall ger oss ett mått på osäkerheten i ett uppskattat populationsmedelvärde medan prediktionsintervall ger oss osäkerheten före en enskild observation. Prediktionsintervall ger ett intervall för en enskild ny observation av den beroende variabeln givet ett visst värde på de oberoende variablerna. Detta ger alltså en uppfattning om inom vilket intervall en enskild ny observation av den beroende variabeln kommer ligga. Eftersom osäkerheten är större hos en enskild observation så kommer även intervallet för en enskild observation vara större. Alltså är ett prediktionsintervall större än ett konfidensintervall.

4. $\beta 0$: kallas interceptet eller skärningen (med y-axeln). Den kan var mer eller mindre meningsfull om det är realistiskt att de oberoende variablerna kan vara noll.

   $\beta 1, \beta 1, \ldots, \beta p$: kallas för regressionskoefficienter och säger något om hur mycket en oberoende variabel (x) påverkar den beroenden variabeln (Y) givet att alla andra oberoende variabler hålls konstanta.

   Om vi tar ett exempel där vi har en multipelregressionsmodell som beskriver priset på ett hus baserat på husets storlek ($x1$, kvadratmeter) och antalet rum det har ($x2$) så skulle den kunna se ut så här:

   Y = 2000000 + 2000*$x1$ + 100000*$x2$ + ε

   $\beta 0$, betyder då att så fort det är ett hus vi säljer (utan några rum eller kvadratmeter) så kostar det 2 miljoner. Som sagt $\beta 0$ kan vara mer eller mindre medingsfull.
   $\beta 1$ säger oss att för varje extra kvadratmeter huset har så kostar det 2000 kr mer.
   $\beta 2$ säger oss att för varje extra rum huset har så kostar det 100000 kr mer.
   ε är vår felterm och representerar den del av variationen som inte kan förklaras av de oberoende variablerna.

   Om vi har ett hus som är 175 kvadratmeter stort med fem rum skulle enligt vår modell huset i så fall kosta: 2000000 + 2000*175 + 100000*5 = 2 850 000 kr.

5. Det stämmer till viss del. BIC (Bayesian Information Criterion) har en term som straffar modeller med fler parametrar. På det sättet minskas sannolikheten att modellen överanpassas och då minskar också behovet av validerings- och testset. BIC fungerar också så att den gör en relativ jämförelse mellan olika modeller och säger alltså inget om hur den faktiskt presterar på nu osedd data. Även i mer komplexa modeller och speciellt i ickelinjära modeller kan förmågan att förhindra överanpassning hos BIC var begränsad. Detta gör att det ändå är viktigt att ha validerings- och testset i dessa fall.

6. Best Subset Selection utförs i ett par olika steg. Till att börja med väljs en null modell M_0 utan några prediktorer. Denna modell sätts till medelvärdet för varje stickprov. Sedan går man vidare med att skapa en M_1 modell som är den bästa modellen som innehåller exakt en prediktor. Där efter skapas modellen M_2 som innehåller exakt två prediktorer och så håller man på till man har skapat M_p modeller. Detta sker givet att vi har p prediktorer och för att välja k modeller så kan det ske på p över k sätt. Den bästa utav dessa modeller (med k prediktorer) är den vars RSS (Residual Sum of Squares) är minst eller störst R^2 (mäter hur stor andel av variationen i den beroende variabeln som förklaras av modellen). När algoritmen har genererat de bästa modellerna för varje antal prediktorer (M_0, M_1,

..., M_p), måste den välja den bästa modellen totalt sett. Detta kan ske med lite olika tekniker så som korsvalidering, C_p, BIC eller adjusted R^2. Modellerna kan inte bli sämre av att man lägger till en prediktor så i teorin är alltid den modell som har flest prediktorer den bästa men skillnaderna kan vara väldigt små och en enklare modell väljs alltid före en mer komplex. Det är därför det kan vara lite klurigt hur man ska göra det sista steget.

7. Hela den komplexa verkligheten kan aldrig hel förklaras med matematiska formler och de modeller vi skapar kan i bästa fall bli en bra approximation av verkligheten. Modeller är helt enkelt en förenkling av verkligheten som bygger på antaganden. Dessa antaganden är heller sällan fullständigt sanna utan just en förenkling. Modellerna vi skapar kan vara användbara inom vissa områden men inte tillräckliga inom andra trotts sina begränsningar. En modell som kan ge en rimlig uppfattning av verkligheten kan ändå vara en fullt tillräcklig modell för att kunna fatta beslut eller dra slutsatser från. På detta sätt är George Box's citat "All models are wrong, some are useful." ett korrekt sätt att se på hur modeller kan användas och fungera.

   Ett exempel på en användbar modell är en karta, som är en förenkling av verkligheten men som fortfarande är väldigt användbar.

8 Källförteckning

Regression analysis. (2025, 24 april). I Wikipedia.
https://en.wikipedia.org/wiki/Regression_analysis

EducationTopicsExplained. (2024, 19 mars). *Linjär Regression* [Video].
http://www.youtube.com/watch?v=NcxMuCG6FS8

SCB (Statistiska Centralbyrån). *Statistikdatabasen*.
https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/

Föreläsningsanteckningar
https://github.com/AntonioPrgomet/linear_regression

9 Bilagor

9.1 SCB API

```
install.packages("pxweb")
library(pxweb)
library(ggplot2)
library(scales)
library(ggthemes)

d <- pxweb_interactive()

# API-URL för fordonsstatistik
api_url <- "https://api.scb.se/OV0104/v1/doris/sv/ssd/TK/TK1001/TK1001A/PersBilarA"

# Hämta metadata
metadata <- pxweb_get(api_url)
names(metadata)
metadata$title
metadata$variables


# Skapa frågelista
query_list <- list(
  Region = c("00"), # Hela Sverige
  ContentsCode = c("TK1001AB"), # personbilar i trafik.
  Tid = c("*") # Alla tillgängliga år
)

# Hämta data
data <- pxweb_get(url = api_url, query = query_list)
data_df <- as.data.frame(data)

print(data_df)

# Konvertera kolumnen "år" till en numerisk datatyp.
data_df$år <- as.numeric(data_df$år)


# Anpassad formateringsfunktion med mellanslag som avgränsare
space_comma <- function(x) {
  format(x, big.mark = " ", scientific = FALSE)
}

# Beräkna lutningen
model <- lm(Antal ~ år, data = data_df)
slope <- coef(model)["år"]

# Hämta värden för första och sista datapunkt
first_value <- data_df$Antal[1]
last_value <- data_df$Antal[nrow(data_df)]

# Skapa ggplot
p <- ggplot(data_df, aes(x = år, y = Antal)) +
```

```
  geom_line(color = "#2196F3", linewidth = 1.5) + # Blå linje
  geom_smooth(method = "lm", se = FALSE, color = "#F44336", linetype = "dashed") + # Streckad trendlinje
  labs(title = "Antal personbilar i trafik i Sverige (2002-2024)",
       x = "År",
       y = "Antal personbilar") +
  theme_gray() + # Lägg till grå bakgrund
  scale_y_continuous(labels = space_comma) +
  geom_text(aes(x = min(år), y = first_value), label = paste("2002:", space_comma(first_value)), hjust = -1, vjust
= -1, size = 3) +
  geom_text(aes(x = max(år), y = last_value), label = paste("2024:", space_comma(last_value)), hjust = 1, vjust =
3, size = 3) +
  geom_text(aes(x = mean(år), y = max(Antal)), label = paste("Antalet personbilar ökar i snitt varje år med: ",
round(slope, 0), " bilar/år"), hjust = 0.5, vjust = -1, size = 4, color = "#F44336") # Lutning

# Exportera grafen som en JPG-bild
ggsave(filename = "antal_bilar_2002_2024.jpg", plot = p, path = "C:/Users/Dator/Documents/Data
Science/06_R/Mina dokument/Kunskapskontroll", width = 12, height = 6)
```

9.2 Datatvätt av blocket data

```
# Installera och ladda nödvändiga paket
install.packages(c("tidyverse", "ggplot2", "caret"))
library(tidyverse)
library(ggplot2)
library(caret)

# Läs in CSV-filen
data <- read.csv("C:/Users/Dator/Documents/Data Science/06_R/Mina
dokument/Kunskapskontroll/datainsamling_blocket_volkswagen.csv", fileEncoding = "UTF-8")


#CLEANING

# Kolla på datastrukturen
str(data)
head(data)
summary(data)

# Identifiera alla kolumner som är av typen character
character_cols <- names(data)[sapply(data, is.character)]

# Loopa genom dessa kolumner och konvertera till gemener
for (col in character_cols) {
  data[[col]] <- tolower(data[[col]])
}


# Räkna NA-värden per kolumn
na_counts <- colSums(is.na(data))
print("Antal NA-värden per kolumn:")
print(na_counts)
# 3 st i horespower

# Räkna tomma textsträngar per kolumn
empty_string_counts <- sapply(data, function(x) sum(x == ""))
print("\nAntal tomma textsträngar per kolumn:")
print(empty_string_counts)
# 1 st fuel_type, 5 st drive_whell_config, 3 st color, 7 st registration_date, 16 st

data$fuel_type[data$fuel_type == ""] <- NA
data$drive_wheel_config[data$drive_wheel_config == ""] <- NA
data$color[data$color == ""] <- NA
data$registration_date[data$registration_date == ""] <- NA

# Kontrollera resultatet
na_counts_updated <- colSums(is.na(data))
print("Antal NA-värden per kolumn efter konvertering av tomma strängar:")
print(na_counts_updated)

# Konvertera 'registration_date' till Date-format
data$registration_date <- as.Date(data$registration_date)
```

```
# Identifiera alla kolumner som är av typen character
character_cols <- names(data)[sapply(data, is.character)]

# Skriv ut de unika värdena för varje character-kolumn
for (col in character_cols) {
  cat(paste("Unika värden i kolumnen '", col, "':\n", sep = ""))
  print(unique(data[[col]]))
  cat("\n")
}

# Åtgärda inkonsekvenser i 'fuel_type'
data$fuel_type[data$fuel_type == "miljöbränsle"] <- "miljöbränsle/hybrid"
data$fuel_type[data$fuel_type == "disel"] <- "diesel"

# Åtgärda inkonsekvenser i 'body_type'
data$body_type[data$body_type == "halvkomni"] <- "halvkombi"

# Åtgärda inkonsekvenser i 'color'
data$color[data$color == "siver"] <- "silver"
data$color[data$color == "vít"] <- "vit"
data$color[data$color == "mörbllå"] <- "mörkblå"

# Åtgärda inkonsekvenser i 'model'
data$model[data$model == "up"] <- "up!"
data$model[data$model == "turan"] <- "touran"

# Hantera felaktiga värden i 'model'
data$model[data$model %in% c("vit", "svart", "silver")] <- NA

# Kontrollera resultatet
for (col in character_cols) {
  cat(paste("Unika värden i kolumnen '", col, "':\n", sep = ""))
  print(unique(data[[col]]))
  cat("\n")
}

# Trimma eventuella inledande eller avslutande mellanslag
data$color <- trimws(data$color)

# Ta bort prefixet "ljus"
data$color <- gsub("^ljus", "", data$color)

# Ta bort prefixet "mörk"
data$color <- gsub("^mörk", "", data$color)

# Trimma igen för att ta bort eventuella nya inledande mellanslag som kan ha uppstått
# om "ljus" eller "mörk" togs bort och det fanns ett mellanslag efter
data$color <- trimws(data$color)

# Kontrollera de unika färgerna igen
cat("Unika värden i kolumnen 'color' efter standardisering:\n")
```

```
print(unique(data$model))
cat("\n")

# Räkna antalet förekomster av varje unik modell
model_counts <- table(data$model)

# Skriv ut antalet för varje modell på en ny rad
print(model_counts)

# Standardisera 'model'-kolumnen
data$model[data$model == "new beetle"] <- "beetle"
data$model[data$model == "crosstouran"] <- "touran"
data$model[data$model == "eos"] <- NA
data$model[data$model == "gti"] <- NA
data$model[data$model == "jetta"] <- NA
data$model[data$model == "lupo"] <- NA
data$model[data$model == "phaeton"] <- NA
data$model[data$model == "polo cross"] <- "polo"
data$model[data$model == "tiguan allspace"] <- "tiguan"


# Räkna antalet rader som har minst ett NA-värde
na_rows_count <- sum(!complete.cases(data))
cat("Antal rader med minst ett NA-värde:", na_rows_count, "\n")

# 10 dyraste bilarna
top_10_price <- data %>%
  arrange(desc(selling_price)) %>%
  head(10)
cat("De 10 dyraste bilarna:\n")
print(top_10_price)
cat("\n")

# 10 bilarna med högst miltal
top_10_mileage <- data %>%
  arrange(desc(mileage)) %>%
  head(10)
cat("De 10 bilarna med högst miltal:\n")
print(top_10_mileage)
cat("\n")

# 10 bilarna med högst modellår
top_10_model_year <- data %>%
  arrange(desc(model_year)) %>%
  head(10)
cat("De 10 bilarna med högst modellår:\n")
print(top_10_model_year)

# Identifiera och sätta NA för den dyra rallybilen (pris 2650000)
data$selling_price[data$selling_price == 2650000] <- NA

# Identifiera och sätta NA för bilen med orimligt högt miltal (miltal 249764)
data$mileage[data$mileage == 249764] <- NA
```

```
# Identifiera och sätta NA för bilen med orimligt högt modellår (modellår 2025)
data$model_year[data$model_year == 2025] <- NA


# Ta bort alla rader med minst ett NA-värde
data_cleaned <- data[complete.cases(data), ]

# Kontrollera antalet rader efter borttagning
cat("Antal rader i den ursprungliga datan:", nrow(data), "\n")
cat("Antal rader i den rensade datan (utan NA):", nrow(data_cleaned), "\n")

# Uppdatera din 'data' dataframe med den rensade versionen
data <- data_cleaned

# Kontrollera att det inte finns några NA-värden kvar
cat("Antal NA-värden per kolumn i den rensade datan:\n")
print(colSums(is.na(data_cleaned)))


# Ta bort kolumnen 'make'
data <- data[, !(names(data) %in% c("make"))]


# Konvertera registration_date till numeriskt (antal dagar sedan 1970-01-01)
data$registration_date_numeric <- as.numeric(data$registration_date)

# Beräkna korrelationen mellan det numeriska datumet och modellåret
correlation <- cor(data$registration_date_numeric, data$model_year, use = "complete.obs")
cat("Korrelation mellan registration_date och model_year:", correlation, "\n")

# Tar bort registration_date
data <- data[, !(names(data) %in% c("registration_date", "registration_date_numeric"))]

# Kontrollera kolumnerna igen
names(data)


# Konvertera registration_date till numeriskt temporärt
registration_date_numeric_temp <- as.numeric(as.Date(data$registration_date))

# Beräkna korrelationen mellan det numeriska datumet och modellåret
correlation <- cor(registration_date_numeric_temp, data$model_year, use = "complete.obs")
cat("Korrelation mellan registration_date och model_year:", correlation, "\n")
#Korrelationen blev 0,1985

# Beräkna korrelationen
year_registration_all <- as.numeric(format(as.Date(data$registration_date), "%Y"))
correlation_by_year_updated_again <- cor(year_registration_all, data$model_year, use = "complete.obs")
cat("Uppdaterad korrelation mellan registreringsår och modellår:", correlation_by_year_updated_again, "\n")
#Blev 0,9868 så det är väldigt stark korrelation -> ta bort datum i trafik
```

```
# Ta bort kolumnen 'registration_date'
data$registration_date <- NULL


# Konvertera kolumnen till integer
data$selling_price <- as.integer(data$selling_price)



# Ange sökväg och filnamn för din CSV-fil i samma mapp som indatafilen
mapp_path <- "C:/Users/Dator/Documents/Data Science/06_R/Mina dokument/Kunskapskontroll/"
file_name <- "cleaned_volkswagen_data.csv"
complete_path <- paste0(mapp_path, file_name)

# Exportera dataframe:n till CSV
write.csv(data, file = complete_path, row.names = FALSE, fileEncoding = "UTF-8")

cat(paste("Din rensade data har exporterats till:", complete_path, "\n"))
```

9.3 Regressionmodell blocket

```
install.packages(c("dplyr", "car", "lmtest", "sandwich")) # Installera vid behov
library(dplyr)
library(car)
library(lmtest)
library(sandwich) # För vcovHC
1. Läs in och förbereder data
------------------------------------------------------------------------------
cat("Läser in och förbereder data...\n")
file_path <- "C:/Users/Dator/Documents/Data Science/06_R/Mina
dokument/Kunskapskontroll/cleaned_volkswagen_data.csv"
data <- read.csv(file_path, fileEncoding = "UTF-8", stringsAsFactors = FALSE)
Justerar två värden på tre bilar
data$selling_price[which(data$selling_price == 31900 & data$model_year == 2022 & data$model == "tiguan")]
<- 319000
data$model[which(data$model_year == 2019 & data$model == "golf" & data$selling_price == 159900 &
data$mileage == 7670 & data$horsepower == 136 & data$seller == "företag" & data$fuel_type == "el")] <- "e-
golf"
data$fuel_type[which(data$selling_price == 279000 & data$mileage == 14500 & data$model_year == 2021 &
data$horsepower == 150)] <- "el"
Lägger alla bilmodeller färre än 20 i en övrigt kategori
antal_per_modell <- data |> count(model, sort = TRUE, name = "antal")
modeller_att_behalla <- antal_per_modell |> filter(antal >= 20) |> pull(model)
data <- data |> mutate(model_reduced = ifelse(model %in% modeller_att_behalla, model, "övrigt"))
data$model <- NULL
str(data)
Plockar bort features
#------------------------------------------------------------------------------
Räknar lite bilar
antal_aldre_an_2010_innan <- data |>
  ((df) sum(df$model_year < 2010, na.rm = TRUE))()
cat("Antal bilar äldre än 2010 (innan borttagning):", antal_aldre_an_2010_innan, "\n")
antal_bilar_under_pris <- data |>
  ((df) sum(df$selling_price < 20000, na.rm = TRUE))()
cat("Antal bilar med ett pris under 20 000 kr:", antal_bilar_under_pris, "\n")
Temporär reducerad dataframe 'data_reduced'
data_reduced <- data |>
  # 1. Ta bort kolumnerna transmission, body_type, drive_wheel_config, color och region
  select(-c("transmission", "body_type", "drive_wheel_config", "color", "region")) |>

  # 2. Ta bort rader där 'model_reduced' är "övrigt"
  filter(model_reduced != "övrigt") |>

  # 3. Ta bort bilar äldre än 2010
  filter(model_year >= 2010)
Kontrollera dimensioner efter rensning
cat("\nDimensioner av den ursprungliga datan:", dim(data), "\n")
cat("Dimensioner av den reducerade datan:", dim(data_reduced), "\n")
Tilldela den reducerade datan till variabeln 'data'
data <- data_reduced
str(data)
```

```
Skapar faktorvariabler som sedan kan skapa dummyvariabler. Olika categoriacal_vars beroende på om
kolumner tagist bort
categorical_vars <- c("seller", "fuel_type", "model_reduced")
#categorical_vars <- c("seller", "fuel_type", "body_type", "model_reduced", "transmission")
categorical_vars <- intersect(categorical_vars, names(data))
data <- data |> mutate(across(all_of(categorical_vars), as.factor))
2. Dela upp data (60% träning, 20% validering, 20% test)
------------------------------------------------------------------------
cat("Delar upp data...\n")
spec = c(train = .6, validate = .2, test = .2)
set.seed(123)
n_rows <- nrow(data)
groups = sample(cut(seq(n_rows), n_rows * cumsum(c(0, spec)), labels = names(spec)))
res = split(data, groups)
train_data <- res$train
val_data <- res$validate
test_data <- res$test
cat("Träningsdata:", nrow(train_data), "rader\n")
cat("Valideringsdata:", nrow(val_data), "rader\n")
cat("Testdata:", nrow(test_data), "rader\n")
3. Träna modeller
------------------------------------------------------------------------
cat("\n--- Tränar Modell 1 (Linjär) ---\n")
lm_1 <- lm(selling_price ~ ., data = train_data)
print(summary(lm_1))
cat("\n--- Tränar Modell 2 (Log-Linjär) ---\n")
lm_2 <- lm(log(selling_price) ~ ., data = train_data)
print(summary(lm_2))
Modell 3: Log-modell + interaktioner, interaktion mellan årsmodell och miltal, samt hk och bränsletyp
cat("\n--- Tränar Modell 3 (Log-Linjär med Interaktioner) ---\n")
lm_3_interaction <- lm(log(selling_price) ~ . + model_year:mileage + horsepower:fuel_type, data = train_data)
print(summary(lm_3_interaction))
4. Diagnostik för alla modeller
------------------------------------------------------------------------
cat("\n--- Diagnostikplottar ---\n")
Funktion för att visa standard lm-plottar
plot_lm_diagnostics <- function(model, model_name) {

 cat("\nDiagnostik för:", model_name, "\n")

 # Spara nuvarande par-inställningar för att återställa dem senare
 old_par <- par(no.readonly = TRUE)
 on.exit(par(old_par)) # Återställ inställningarna när funktionen avslutas

 # Ställ in 2x2 layout, justerat marginaler och lägg till yttre marginal för huvudrubrik
 # mar = c(botten, vänster, topp, höger)
 # oma = c(botten, vänster, topp, höger) yttre marginaler
 par(mfrow = c(2, 2), mar = c(4, 4, 2, 1), oma = c(0, 0, 3, 0))

 # Rita plottarna utan individuell titel
 plot(model)

 # Lägg till en övergripande titel ovanför alla plottar
```

```
 title(main = paste("Diagnostik för:", model_name), outer = TRUE, line = 1.5) # outer=TRUE för yttre titel,
justera line för position

}
Visa plottar
plot_lm_diagnostics(lm_1, "Modell 1 (Linjär)")
plot_lm_diagnostics(lm_2, "Modell 2 (Log-Linjär)")
plot_lm_diagnostics(lm_3_interaction, "Modell 3 (Log + Interaktion)")
********************************************************************************
Funktion för att spara diagnostikplot som jpg
save_lm_diagnostics <- function(model, model_name, filepath = "C:/Users/Dator/Documents/Data
Science/06_R/Mina dokument/Kunskapskontroll/") {
 if(is.null(model)) {
  cat("Hoppar över sparande av diagnostik för:", model_name, "(modellen är NULL)\n")
  return()
 }
 filename <- paste0(filepath, model_name, ".jpg")
 jpeg(filename = filename, width = 1200, height = 1200, res = 150)
 old_par <- par(no.readonly = TRUE)
 par(mfrow = c(2, 2), mar = c(4, 4, 2, 1), oma = c(0, 0, 3, 0))
 plot(model)
 title(main = paste("Diagnostik för:", model_name), outer = TRUE, line = 1.5)
 par(old_par)
 dev.off()
 cat("Diagnostikplot sparad som:", filename, "\n")
}
Spara plottarna
cat("\n--- Sparar Diagnostikplotter ---\n")
save_lm_diagnostics(lm_1, "Modell_1_Linjar")
save_lm_diagnostics(lm_2, "Modell_2_Log_Linjar")
save_lm_diagnostics(lm_3_interaction, "Modell_3_Log_Interaktion")
********************************************************************************
points_of_interest_indices <- c(582, 877, 1156, 890, 834, 811, 377, 134, 477, 915, 539, 641, 581)
#points_of_interest_indices <- c(915, 539, 641, 581, 509, 334, 843, 210)
points_of_interest_indices <- c(333, 322, 670)
3. Plocka ut raderna direkt baserat på index
points_of_interest <- data[points_of_interest_indices, ]
4. Visa informationen
print("Information om de intressanta bilarna:")
print(points_of_interest)
********************************************************************************
Ytterligare diagnostik (VIF och bptest)
cat("\n--- Ytterligare Diagnostik (VIF & Heteroskedasticitet) ---\n")
run_diagnostics <- function(model, model_name) {
 if(is.null(model)) {
  cat("Hoppar över VIF/bptest för:", model_name, "(modellen är NULL)\n")
  return()
 }
 cat("\nDiagnostik för:", model_name, "\n")
 # VIF
 cat(" VIF:\n")
 tryCatch({
  vif_result <- vif(model)
```

```
   # Hantera om VIF returnerar en matris (för faktorer med >2 nivåer)
   if (is.matrix(vif_result)) {
     print(vif_result)
   } else {
     print(head(sort(vif_result, decreasing = TRUE)))
   }
 }, error = function(e) message("  VIF-fel: ", conditionMessage(e)))
 # Breusch-Pagan test
 cat(" Breusch-Pagan Test (Homoskedasticitet):\n")
 tryCatch({ print(bptest(model)) }, error = function(e) message("  bptest-fel: ", conditionMessage(e)))
}
run_diagnostics(lm_1, "Modell 1 (Linjär)")
run_diagnostics(lm_2, "Modell 2 (Log-Linjär)")
run_diagnostics(lm_3_interaction, "Modell 3 (Log + Interaktion)")
5. Jämför modeller på Valideringsdata (RMSE)
------------------------------------------------------------------------
cat("\n--- Jämför modeller på Valideringsdata (RMSE) ---\n")
calculate_rmse <- function(actual, predicted) {
 sqrt(mean((actual - predicted)^2, na.rm = TRUE))
}
Skapa lista för resultat
rmse_results <- list()
Modell 1
pred_1_val <- predict(lm_1, newdata = val_data, na.action = na.pass)
pred_1_val[pred_1_val < 0] <- 1
rmse_results[["Linjär"]] <- calculate_rmse(val_data$selling_price, pred_1_val)
Modell 2
pred_2_log_val <- predict(lm_2, newdata = val_data, na.action = na.pass)
pred_2_price_val <- exp(pred_2_log_val)
rmse_results[["Log-Linjär"]] <- calculate_rmse(val_data$selling_price, pred_2_price_val)
Modell 3
pred_3_log_val <- predict(lm_3_interaction, newdata = val_data, na.action = na.pass)
pred_3_price_val <- exp(pred_3_log_val)
rmse_results[["Log + Interaktion"]] <- calculate_rmse(val_data$selling_price, pred_3_price_val)
Skapa och skriv ut resultat-df
results_val <- data.frame(
 Model = names(rmse_results),
 Validation_RMSE = unlist(rmse_results)
)
Sortera efter RMSE
results_val <- results_val[order(results_val$Validation_RMSE), ]
print(results_val)
6. Välj bästa modell och utvärdera på Testdata
------------------------------------------------------------------------
cat("\n--- Väljer bästa modell och utvärderar på Testdata ---\n")
best_model_name <- results_val$Model[1] # Ta den med lägst RMSE
best_model_object <- switch(best_model_name,
                            "Linjär" = lm_1,
                            "Log-Linjär" = lm_2,
                            "Log + Interaktion" = lm_3_interaction)
cat("Bästa modell baserat på Validering RMSE:", best_model_name, "\n")
Beräkna Test RMSE för den bästa modellen
if (best_model_name == "Linjär") {
```

```
  pred_best_test <- predict(best_model_object, newdata = test_data, na.action = na.pass)
  pred_best_test[pred_best_test < 0] <- 1
  rmse_test_final <- calculate_rmse(test_data$selling_price, pred_best_test)
} else { # Log-baserad modell
  pred_best_log_test <- predict(best_model_object, newdata = test_data, na.action = na.pass)
  pred_best_price_test <- exp(pred_best_log_test)
  rmse_test_final <- calculate_rmse(test_data$selling_price, pred_best_price_test)
}
medelpris <- mean(data$selling_price)
print(paste("Medelpriset på bilarna är:", medelpris))
cat("Slutlig Test RMSE (för vald modell):", rmse_test_final, "\n")
7. Prediktion för två nya bilar
--------------------------------------------------------------------------
cat("\n--- Prediktion för två nya bilar ---\n")
Hämta faktornivåer från träningsdatan
factor_levels <- lapply(train_data[, categorical_vars, drop = FALSE], levels)
# Två nya bilar (med transmission och body_type)
new_cars <- data.frame(
mileage = c(5000, 12000),
model_year = c(2021, 2018),
horsepower = c(150, 110),
seller = factor(c("företag", "privat"), levels = factor_levels$seller),
fuel_type = factor(c("bensin", "diesel"), levels = factor_levels$fuel_type),
transmission = factor(c("manuell", "automat"), levels = factor_levels$transmission),
body_type = factor(c("halvkombi", "suv"), levels = factor_levels$body_type),
model_reduced = factor(c("golf", "tiguan"), levels = factor_levels$model_reduced)
)
Nya bilar
new_cars <- data.frame(
  new_cars <- data.frame(
    mileage = c(5000, 12000, 17500, 14978),
    model_year = c(2021, 2018, 2017, 2017),
    horsepower = c(150, 110, 190, 111),
    seller = factor(c("företag", "privat", "företag", "företag"), levels = factor_levels$seller),
    fuel_type = factor(c("bensin", "diesel", "diesel", "bensin"), levels = factor_levels$fuel_type),
    model_reduced = factor(c("golf", "tiguan", "tiguan", "golf"), levels = factor_levels$model_reduced)
  )
)
Gör prediktion med prediktionsintervall
predictions_new <- predict(best_model_object, newdata = new_cars, interval = "prediction", level = 0.95)
best_model_name <- "Log"
predictions_new <- predict(lm_1, newdata = new_cars, interval = "prediction", level = 0.95)
best_model_name <- "Linjär"
predictions_new <- predict(lm_2, newdata = new_cars, interval = "prediction", level = 0.95)
best_model_name <- "Log"
Återtransformera om bästa modellen är log-baserad
if (!is.null(predictions_new)) {
  if (best_model_name != "Linjär") {
    predictions_new <- exp(predictions_new)
    cat("Prediktioner (återtransformerade från log-skala):\n")
  } else {
    # Sätt ev negativa gränser till 0 eller 1
    predictions_new[predictions_new[,"lwr"] < 0, "lwr"] <- 1
```

```
   predictions_new[predictions_new[,"fit"] < 0, "fit"] <- 1
   cat("Prediktioner (linjär skala):\n")
  }
  print(predictions_new)
}
8. Analys av bästa modellen
--------------------------------------------------------------------------
cat("\n--- Analys av vald modell:", best_model_name, "---\n")
if (!is.null(best_model_object)) {
  print(summary(best_model_object))

  cat("\nKonfidensintervall för koefficienter:\n")
  tryCatch({ print(confint(best_model_object)) }, error = function(e) message("Fel vid confint: ",
conditionMessage(e)))

  cat("\nTest av koefficienter med robusta standardfel (HC1):\n")
  tryCatch({
    # Använd sandwich::vcovHC() explicit
    print(coeftest(bes_modtel_object, vcov = sandwich::vcovHC(best_model_object, type = "HC1")))
  }, error = function(e) message("Fel vid coeftest/vcovHC: ", conditionMessage(e)))
} else {
  cat("Ingen bästa modell kunde väljas eller tränas korrekt.\n")
}
```