VOORSPELLING VAN HOTELPRIJZEN MET REGRESSIEMODELLEN

Een Cursusprojectrapport

Universiteit van Michigan

Yi Yang

4/12/2023

Hoofdstuk 1 Inleiding

1.1 Motivatie

Japan behoort al lang tot de populairste reisbestemmingen wereldwijd, waarbij steden zoals Tokio, Kyoto en Osaka jaarlijks miljoenen toeristen trekken vanwege hun cultureel erfgoed, moderne attracties en culinaire diversiteit [21]. Voor reizigers is het beheersen van kosten cruciaal, en accommodatiekosten—met name hostelprijzen—vormen een aanzienlijk deel van het reisbudget. Het begrijpen en voorspellen van deze kosten is essentieel voor zowel toeristen die hun reis plannen als bedrijven die hun prijsstrategieën optimaliseren. Recente onderzoeken benadrukken de dynamische aard van prijsbepaling in de horeca, beïnvloed door factoren zoals locatie, seizoensgebonden vraag, voorzieningen en online recensies [17,19,24]. Desondanks richten weinig studies zich specifiek op hostelprijzen in Japan, ondanks de unieke marktkenmerken zoals compacte stedelijke structuren en hoge toeristische dichtheid [21].

Figure 1

De opkomst van machine learning (ML) heeft een revolutie teweeggebracht in voorspellende analyses binnen toerisme en horeca. Technieken zoals ensemble learning [26], deep learning [22] en robuuste regressie [3] zijn succesvol toegepast op prijsvoorspellingen in vastgoed en hotels. Zo toonden Smith en Lee [16] de superioriteit van gradient-boosted trees aan bij hotelprijsvoorspellingen, terwijl Garcia et al. [17] de nadruk legden op geospatiale factoren in stedelijke accommodatieprijzen. Desondanks blijft hostelprijsbepaling onderbelicht, vooral in contexten die gedetailleerde feature engineering [18] en adaptieve modellering vereisen om fluctuerende vraag te adresseren [20].

Deze studie overbrugt deze kloof door een op maat gemaakt ML-model te ontwikkelen voor het voorspellen van hostelprijzen in Tokio, Kyoto en Osaka. Geïnspireerd door optimalisatieframeworks uit controlesystemen—zoals genetische algoritmen [2] en particle swarm optimization (PSO) [7]—passen we deze methodieken aan om modelnauwkeurigheid en robuustheid te verbeteren. Daarnaast informeren fractionele orde-controllestrategieën [9,12], traditioneel gebruikt in technische systemen, onze aanpak voor het omgaan met niet-lineaire prijsdynamiek en onzekerheden in toerismedata [13].

1.2 Gegevensbron

Onze dataset, afkomstig van Kaggle [25], bevat gedetailleerde kenmerken zoals locaties van hostels, beoordelingen, voorzieningen en seizoensgebonden prijsvariaties. Kaggle-datasets zijn prominent aanwezig in toerismeonderzoek vanwege hun toegankelijkheid en volledigheid [25], hoewel uitdagingen zoals ontbrekende waarden en uitschieters robuuste preprocessing vereisen [3,18].

Figure 2

Deze gegevens zijn verzameld van de website HostelWorld.com [19].

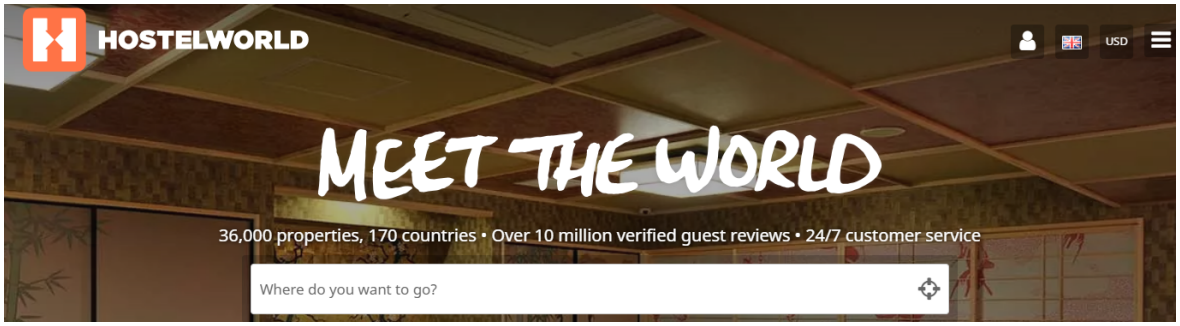


Figure 3

Belangrijke kenmerken die hostelprijzen beïnvloeden zijn:

1. **Geospatiale factoren**: Nabijheid tot vervoersknooppunten en toeristische attracties, zoals geanalyseerd in stedelijk toerismeonderzoek [17].
2. **Voorzieningen**: Diensten zoals gratis Wi-Fi en ontbijt, die een significante invloed hebben op reizigersvoorkeuren [24].

3. **Temporele trends**: Seizoensgebonden vraagfluctuaties, in lijn met bevindingen in revenue management [27].
4. **Online reputatie**: Beoordelingsscores en zichtbaarheid op boekingsplatforms, cruciaal in het digitale tijdperk [19].

Om deze factoren te modelleren, evalueren we meerdere ML-algoritmen, waaronder random forests, support vector regression (SVR) en neurale netwerken, voortbouwend op vergelijkende analyses door Wang en Kim [23]. Hyperparameterafstemming maakt gebruik van PSO [7], een metaheuristiek die effectief is gebleken in technische controle [9], om modelprestaties te optimaliseren. Bovendien inspireren fractionele orde PID-regelaars [8,11] onze aanpak voor het verwerken van vertraagde feedback in prijsdata, waardoor aanpassingsvermogen aan realtime vraagverschuivingen wordt gegarandeerd.

1.3 Bijdragen en betekenis

Deze studie levert drie belangrijke bijdragen:

1. **Gedetailleerde kenmerkanalyse**: Uitbreiding van het werk van Nguyen en Chen [19] door de impact van online recensies op hostelprijzen te kwantificeren.
2. **Adaptieve modellering**: Integratie van principes uit de controletheorie [2,7,9] in ML-workflows om dynamische prijsuitdagingen aan te pakken.
3. **Praktisch hulpmiddel voor stakeholders**: Een gebruiksvriendelijk voorspellingsmodel dat reizigers en hosteloperators ondersteunt bij budgetplanning en inkomstenbeheer.

Door methodieken uit controlesystemen [11,13] en ML [16,22] te synthetiseren, bevordert dit werk interdisciplinair onderzoek in toerismeanalyse. Toekomstige uitbreidingen kunnen realtime datastromen en federatieve leerframeworks omvatten om schaalbaarheid te verbeteren [22].

Hoofdstuk 2 Methoden

2.1 Beschrijving van de gegevens

2.1.1 Overzicht van de dataset

De oorspronkelijke dataset wordt getoond in Figure 4. De dataset bevat 342 voorbeelden.

hostel.name	City	price.from	Distance	summary.	rating.ban	atmosphe	cleanlines	facilities	location.y	security	staff	valueforn	lon	lat
"Bike & Bed" CharinCo	Osaka	3300	2.9km fro	9.2	Superb	8.9	9.4	9.3	8.9	9	9.4	9.4	135.5138	34.68268
& And Hostel	Fukuoka-(	2600	0.7km fro	9.5	Superb	9.4	9.7	9.5	9.7	9.2	9.7	9.5	NA	NA
&And Hostel Akihabara	Tokyo	3600	7.8km fro	8.7	Fabulous	8	7	9	8	10	10	9	139.7775	35.69745
&And Hostel Ueno	Tokyo	2600	8.7km fro	7.4	Very Gooc	8	7.5	7.5	7.5	7	8	6.5	139.7837	35.71272
&And Hostel-Asakusa M	Tokyo	1500	10.5km fr	9.4	Superb	9.5	9.5	9	9	9.5	10	9.5	139.7984	35.7279
1night1980hostel Toky	Tokyo	2100	9.4km fro	7	Very Gooc	5.5	8	6	6	8.5	8.5	6.5	139.7869	35.72438
328 Hostel & Lounge	Tokyo	3300	16.5km fr	9.3	Superb	8.7	9.7	9.3	9.1	9.3	9.7	8.9	139.7455	35.54804
36Hostel	Hiroshima	2000	1.6km fro	9.5	Superb	8.8	9.9	9.2	9.6	9.8	9.8	9.5	NA	NA
3Q House - Asakusa Sm	Tokyo	2500	10.2km fr	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Ace Inn Shinjuku	Tokyo	2200	3km from	7.7	Very Gooc	6.7	7.2	6.8	8.5	7.8	8.5	8.1	139.7243	35.69251
Air Osaka Hostel	Osaka	1600	9.7km fro	9.2	Superb	9.5	9.1	8.7	8.8	8.9	9.8	9.5	135.477	34.62226
Aizuya Inn	Tokyo	2000	10.6km fr	8.5	Fabulous	8.1	8.3	8.4	7.8	8.9	9.1	8.9	139.801	35.72755
Akihabara Hotel 3000	Tokyo	2200	8km from	10	Superb	10	10	10	10	10	10	10	139.7794	35.69749
Almond hostel & cafe S	Tokyo	2900	2.2km fro	9.3	Superb	9.1	9.5	8.8	9.5	9.4	9.7	9	139.6875	35.67001
Anne Hostel Asakusaba	Tokyo	2000	8.9km fro	9.1	Superb	8.8	9.2	8.7	9	9.1	9.5	9.2	139.7894	35.69894
Anne Hostel Yokozuna	Tokyo	1800	9.5km fro	9.1	Superb	8.8	9.1	9	9.2	9.3	9.3	9.2	139.7968	35.69549

Figure 4

De responsvariabele is de minimumprijs van een hostel voor een verblijf van één nacht.

De verklarende variabelen worden hieronder weergegeven:

- Afstand
- Sfeer
- Netheid
- Faciliteiten
- Locatie
- Veiligheid
- Personeel
- 2 indicatoren (Tokio, Osaka, Kyoto)

We verdelen deze verklarende variabelen in drie categorieën. De eerste categorie is de afstand, die de afstand tussen het hostel en het stadscentrum vertegenwoordigt. De tweede categorie van de verklarende variabele zijn de beoordelingsscores van klanten, die sfeer, netheid, faciliteiten, locatie, veiligheid en personeel omvatten. De laatste categorie van de verklarende variabele is de indicator. Er zijn 2 indicatoren omdat we drie steden in de dataset hebben (Tokio, Osaka, Kyoto).

2.1.2 Gegevensverwerking

Nadat we de dataset hadden ontvangen, gebruikten we de volgende methode om de gegevens te verwerken:

- Verwijder overbodige tekens:

Een voorbeeld van de overbodige tekens wordt getoond in Figure 5. In dit geval hebben we de tekens in het rode vak verwijderd.

price.from	Distance	summary.	atmosphe	cleanlines	facilities	location.y	security
3300	2.9km from city centre	9.2	8.9	9.4	9.3	8.9	9
2600	0.7km from city centre	9.5	9.4	9.7	9.5	9.7	9.2
3600	7.8km from city centre	8.7	8	7	9	8	10
2600	8.7km from city centre	7.4	8	7.5	7.5	7.5	7
1500	10.5km from city centre	9.4	9.5	9.5	9	9	9.5
2100	9.4km from city centre	7	5.5	8	6	6	8.5

Figure 5

- Verwijder onvolledige voorbeelden:

Een voorbeeld van een onvolledig voorbeeld in de dataset wordt getoond in Figure 6. We hebben alle voorbeelden verwijderd die de onvolledige gegevens bevatten (in het rode vak).

9.3	Superb	8.7	9.7	9.3	9.1	9.3	9.7	8.9	139.7455	35.54804
9.5	Superb	8.8	9.9	9.2	9.6	9.8	9.8	9.5	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Figure 6

- Standaardisatie:

Omdat we later hogere-orde termen in ons voorspellingsmodel zullen introduceren, gebruiken we standaardisatie om de gegevens te verwerken en zo het probleem van multicollineariteit te verminderen.

2.2 Voorlopige verkennende analyses

In dit project overwegen we eerst een lineair model van de eerste orde, dat bestaat uit de eerste-orde termen van negen voorspellende variabelen. Het eerste-orde model wordt als volgt gepresenteerd.

$$\text{Price} \sim \text{Distance} + \text{atmosphere} + \text{cleanliness} + \text{facilities} + \text{location} + \text{security} + \text{staff} + x_1 + x_2$$

waar x_1 en x_2 twee indicatorvariabelen zijn die de drie steden in de categorische variabele vertegenwoordigenOp basis van dit eerste-orde lineaire model kunnen de residuplots worden weergegeven in Figure 7. Het is duidelijk zichtbaar in de residuplots dat er een kwadratisch patroon bestaat voor de eerste vier voorspellende variabelen.

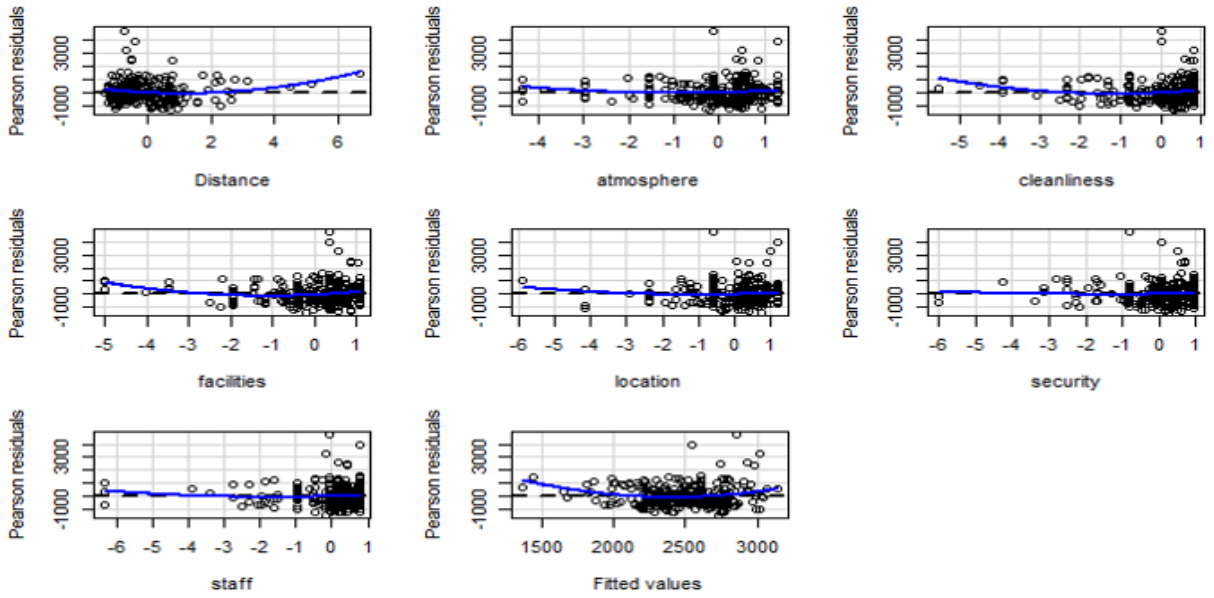


Figure 7. Residuplots voor het lineaire regressiemodel van de eerste orde.

Aangezien het kwadratische patroon duidelijk waarneembaar is in de residuplots, is het noodzakelijk om de bronnen te onderzoeken die bijdragen aan dit kwadratische patroon. Ten eerste vermoeden we dat het kwadratische patroon wordt veroorzaakt door uitschieters buiten het bereik van de gegevenskern. Na het verwijderen van deze uitschieters uit de oorspronkelijke dataset worden de residuplots opnieuw gegenereerd, waarin het kwadratische patroon nog steeds aanwezig is. Daarom is het logisch om de kwadratische term van de eerste vier voorspellende variabelen toe te voegen en het lineaire regressiemodel naar de tweede orde te verhogen. De nieuwe regressieformule wordt als volgt gegeven:

$$\text{Price} \sim \text{Distance} + \text{Distance}^2 + \text{atmosphere} + \text{atmosphere}^2 + \text{cleanliness} + \text{cleanliness}^2 + \text{facilities} + \text{facilities}^2 + \text{location} + \text{security} + \text{staff} + \mathrm{x}_1 + x_2$$

2.3 Controleer modelaannames

Hoewel het lineaire regressiemodel van de tweede orde het kwadratische patroon in de spreidingsplot elimineert, moet dit model nog worden geanalyseerd om de aannames over de fouttermen te controleren. In het standaard lineaire regressiemodel wordt aangenomen dat de fouttermen onafhankelijk identiek verdeeld zijn met een normale verdeling $N(0, \sigma^2)$. Uit Figure 7, is gemakkelijk te zien dat de fouttermen geen constante variantie hebben, aangezien de verdeling van de datapunten niet bij benadering symmetrisch is ten opzichte van de horizontale nul lijn.

Daarnaast worden de waarschijnlijkheidsplot en de Brown-Forsythe-test weergegeven Figure 8 en Figure 9. Uit deze plots en argumenten kan worden afgeleid dat de fouttermen niet normaal verdeeld zijn en dat de fouttermen geen constante variantie hebben.

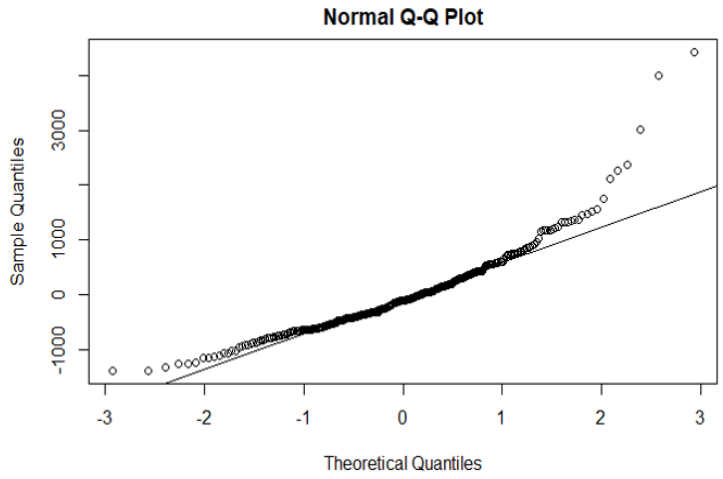


Figure 8. Waarschijnlijkheidsplot voor het model van de tweede orde.

```
  Brown-Forsythe Test
------------------------------------------------------------
  data : residual and Group

  statistic  : 281.4274
  num df     : 1
  denom df   : 166.3299
  p.value    : 1.327949e-37

  Result     : Difference is statistically significant.
------------------------------------------------------------
```

Figure 9. Brown-Forsythe-test voor het model van de tweede orde.

De diagnostiek van het lineaire regressiemodel van de tweede orde toont aan dat het noodzakelijk is maatregelen te nemen om de niet-normaliteit en de niet-constante variantieproblemen veroorzaakt door de fouttermen aan te pakken. Onze eerste stap is het toepassen van transformatietechnieken op de responsvariabele *Price*, aangezien de niet-constantheid van de variantie mogelijk wordt opgelost na de transformatie. De Box-Cox-transformatie wordt toegepast. Aangezien $\lambda = 0$ binnen het betrouwbaarheidsinterval valt in Figure 10, kiezen we ervoor om eerst een logaritmische transformatie toe te passen om het resultaat te bekijken.

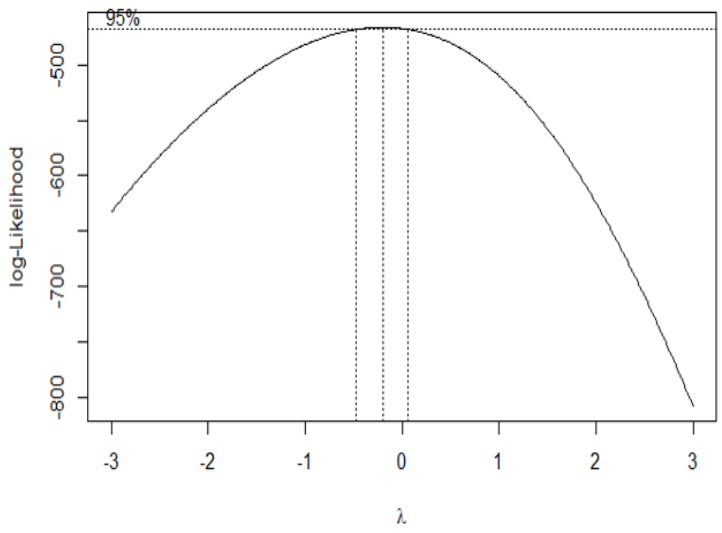


Figure 10. Box-Cox-transformatie.

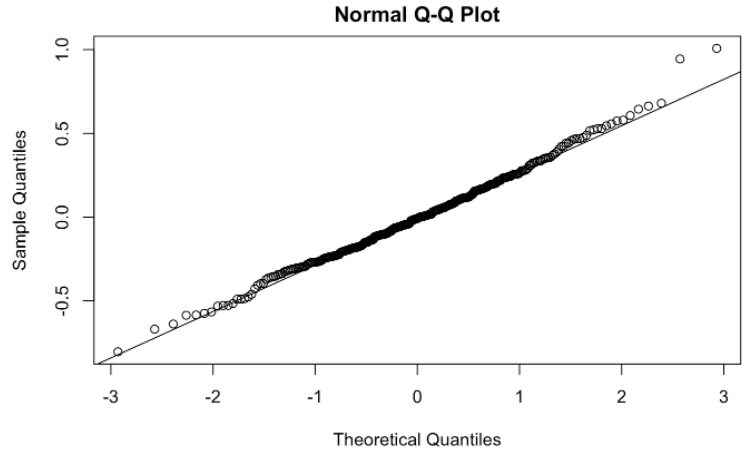


Figure 11. De waarschijnlijkheidsplot voor het getransformeerde model.

2.4 Proces van modelbouw

In eerdere studies hebben we ontdekt dat het transformeren van Y het normaliteitsprobleem kan oplossen, maar we hebben nog steeds geen constante variantie. Op basis van wat we hebben geleerd, kiezen we ervoor om gewogen regressie toe te passen om een geschikt model te bouwen.

2.4.1 Gewogen lineaire regressie

- Selectie van modelvoorspellers

Aangezien de methode voor modelselectie voornamelijk is gebaseerd op een ongewogen model, kiezen we ervoor om voorspellers te selecteren voor dit geval. Daarna kunnen we de geselecteerde voorspellers gebruiken voor het gewogen model.

We overwegen eerst een model zonder interactieterm. Omdat de grootte van de voorspellers klein is, kunnen we zowel de stapsgewijze methode (gebaseerd op PRESS, aangezien we de hostelprijs willen voorspellen) als de beste subset (standaard gebaseerd op AIC) proberen. Dit levert ons het volgende op:

$$\ln(Price) \sim Distance + clieanliness + staff + Distance^2 + facilities^2 + x1 + x2$$

Vervolgens willen we ook het verschillende effect van voorspellers voor verschillende steden modelleren. Daarom hebben we 22 interactietermen toegevoegd en de stapsgewijze functie toegepast (te veel voorspellers voor de beste subset) om de voorspellers voor de regressie te vinden.

$$\ln(Price) \sim Distance + clieanliness + staff + Distance^2 + facilities^2 + x1 + x2 + x2{:}\,Distance + x2{:}\,Distance^2$$

- Selectie van de gewichtsfunctie

Na het selecteren van de voorspellers moeten we nog één laatste aspect overwegen: hoe we de gewichtsfunctie modelleren. Hiervoor plotten we de residuen tegenover de voorspellers om het patroon van de residuen te bekijken.

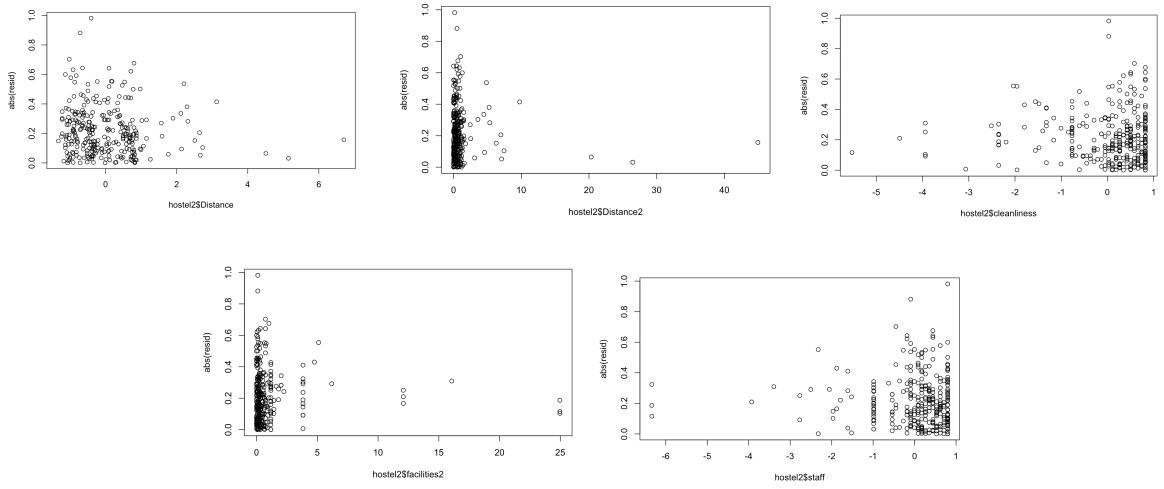


Figure 12 Residual vs. predictors

Op basis van het residupatroon kiezen we ervoor om de standaardafwijking $\sigma_i = X_i\gamma$ aan te passen. Het gewicht wordt dan $\mathrm{w_i} = 1/\sigma_i$.

Na het selecteren van zowel de voorspellers als de gewichtsfunctie kunnen we beginnen met het bouwen van dit gewogen lineaire regressiemodel.

2.4.2 Gewogen Ridge-regressie

In gewogen ridge-regressie nemen we alle voorspellers, evenals twee interactietermen die we in de eerdere stappen hebben geselecteerd, om ons model te bouwen.

We gebruiken GCV om een geschikte λ te kiezen en passen gewogen ridge-regressie toe om een ander model te bouwen.

Hoofdstuk 3 Resultaten

3.1 Gewogen lineaire regressie

3.1.1 Eindmodel

In hoofdstuk 2 hebben we voorspellers geselecteerd met behulp van de stapsgewijze methode, waarbij voornamelijk werd geprobeerd het model met de kleinste AIC te vinden.

```
Step:  AIC=-748.42
Price ~ X2 + cleanliness + facilities2 + X1 + Distance + Distance2 +
    staff + X2:Distance + X2:Distance2

                Df Sum of Sq    RSS     AIC
<none>                      22.073 -748.42
+ facilities     1   0.09974 21.973 -747.76
+ atmosphere     1   0.08472 21.988 -747.56
+ cleanliness2   1   0.08234 21.990 -747.53
+ cleanliness:X2 1   0.07858 21.994 -747.48
+ atmosphere2    1   0.06281 22.010 -747.27
+ facilities2:X2 1   0.05194 22.021 -747.12
+ staff:X2       1   0.03587 22.037 -746.90
+ facilities2:X1 1   0.01869 22.054 -746.67
+ security       1   0.01788 22.055 -746.66
+ staff:X1       1   0.01595 22.057 -746.64
+ Distance:X1    1   0.00984 22.063 -746.55
+ location       1   0.00753 22.065 -746.52
+ Distance2:X1   1   0.00555 22.067 -746.50
+ cleanliness:X1 1   0.00485 22.068 -746.49
- staff          1   0.40246 22.475 -745.07
- cleanliness    1   0.91694 22.990 -738.37
- X2:Distance2   1   1.11705 23.190 -735.81
- facilities2    1   1.21977 23.292 -734.50
- X2:Distance    1   1.61784 23.691 -729.48
- X1             1   2.22686 24.299 -721.97
```

Figure 13 Resultaat van de stapsgewijze functie

We hebben besloten hoe we de voorspellers en de gewichtsfunctie kiezen, en nu kunnen we het model aanpassen.

$$\ln(Price) \sim Distance + clieanliness + staff + Distance^2 + facilities^2 + x1 + x2 + x2{:}\,Distance + x2{:}\,Distance^2$$

	Regression Model		Difference(%)
	Weighted	Unweighted	
(Intercept)	7.8891	7.8927	0.046
Distance	-0.1899	-0.1973	3.897
Distance2	0.0362	0.0375	3.591
cleanliness	0.0857	0.0824	3.851
facilities2	0.0299	0.0305	2.007
staff	0.0458	0.0507	10.699
X1TRUE	-0.2218	-0.2279	2.750
X2TRUE	-0.0835	-0.1171	40.240
Distance:X2TRUE	0.3801	0.3614	4.920
Distance2:X2TRUE	-0.0863	-0.0801	7.184

Figure 14 Vergelijking tussen gewogen en ongewogen model

We kunnen zien dat de gewogen regressie bijna dezelfde coëfficiënten heeft als de ongewogen regressie, wat tevens ondersteunt dat we voorspellers kunnen selecteren op basis van het ongewogen lineaire regressiemodel.

3.1.2 Eenvoudige analyse van het model

- Normaliteit

Aangezien we verschillende voorspellers gebruiken dan in het initiële model, kunnen we ons normaliteitsprobleem opnieuw controleren.

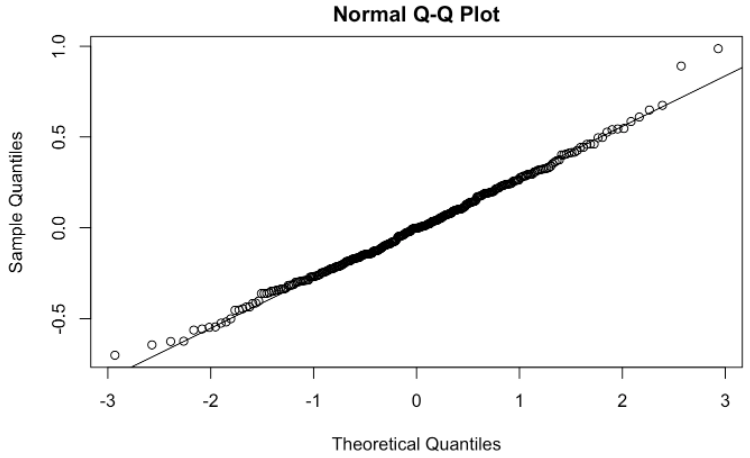


Figure 15 Q-Q-plot voor gewogen lineaire regressie

Bovendien geeft de Shapiro-Wilk normaliteitstest een p-waarde van 0.3665, wat aantoont dat het residu normaal verdeeld is.

- Residu

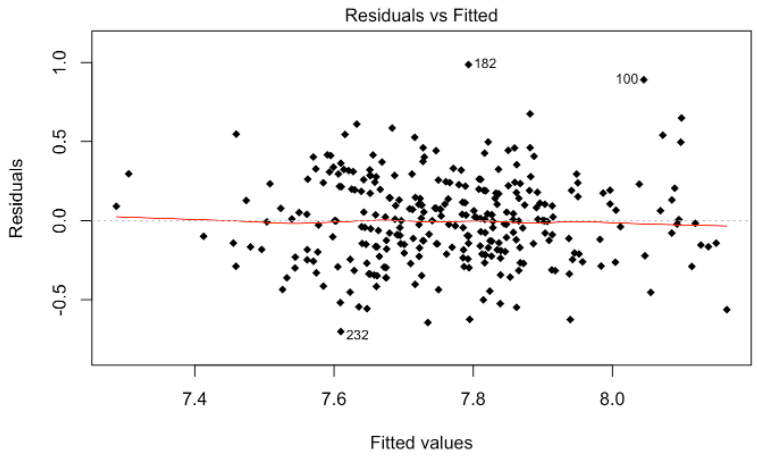


Figure 16 Residu versus geschatte waarden voor gewogen lineaire regressive

We kunnen zien dat de gemiddelde waarde van het residu vrij dicht bij nul ligt, wat positief is.

- Uitschieters en invloedrijke punten

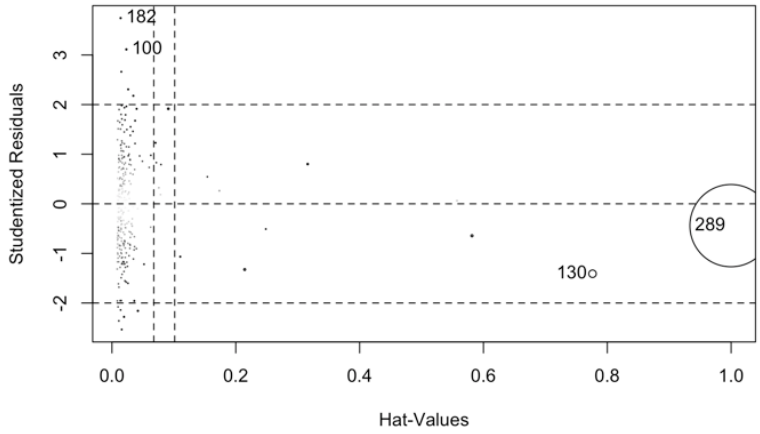


Figure 17 Invloedplot van het gewogen model

Uit de figuur kunnen we zien dat geen enkele absolute waarde van gestudentiseerde residuen groter is dan $qt\left(1-\frac{\alpha}{2}, n-1-p\right) = 3.81,$ wat betekent dat we geen uitschieters hebben in Y.

Echter, we hebben wel enkele uitschieters in XXX en enkele invloedrijke punten die de modelselectiestap kunnen beïnvloeden.

Om hun invloed te controleren, hebben we de gegevens 289 en 130 uitgesloten. De initiële residuplot ondersteunt nog steeds het opnemen van hogere-orde termen en het resultaat blijft vergelijkbaar.

3.2 Gewogen ridge-regressie

In gewogen ridge-regressie [4] hebben we de volgende voorspellers geselecteerd:

- Afstand, sfeer, netheid, faciliteiten, locatie, veiligheid, personeel
- Afstand², sfeer², netheid², faciliteiten²
- X1 + X2
- X2: Afstand + X2: Afstand

De waarde van λ is gekozen van 0 tot 20 met een stapgrootte van 0,01.

We hebben de volgende resultaten verkregen voor de modelselectie.

- Gewijzigde HKB-schatter is 6,976383
- Gewijzigde L-W-schatter is 41,48205
- Kleinste waarde van GCV bij 4,41

Aangezien voorspellingen nodig zijn voor ons werk, kiezen we de beste λ=4, 41 op basis van GCV.

Hoofdstuk 4 Discussie

4.1 Modelvergelijking

5-voudige kruisvalidatie is toegepast om de modellen te vergelijken. De details worden weergegeven in 1.

Table1. Resultaten van kruisvalidatie voor twee modellen

Model	Mean Absolute Error
Weighted Linear Regression	0.22
Weighted Ridge Regression	0.25

Zoals weergegeven in de kruisvalidatietabel, is de MAE van gewogen lineaire regressie vergelijkbaar maar iets lager dan die van de gewogen ridge-regressie. Aangezien de gewogen lineaire regressie minder voorspellers bevat, geven we de voorkeur aan de gewogen lineaire regressie boven de gewogen ridge-regressie.

4.2 Conclusie

De coëfficiënten van het geselecteerde model worden hieronder weergegeven met de Box-Cox-transformatieformule $Y' = e^Y$.

	Regression Model
	Efficient
(Intercept)	7.8891
Distance	-0.1899
Distance2	0.0362
cleanliness	0.0857
facilities2	0.0299
staff	0.0458
X1TRUE	-0.2218
X2TRUE	-0.0835
Distance:X2TRUE	0.3801
Distance2:X2TRUE	-0.0863

Uit de coëfficiënten van het model blijkt dat verschillende beoordelingsfactoren vergelijkbare effecten hebben in de drie verschillende steden (zoals netheid, faciliteiten en personeel). De afstand heeft een vergelijkbaar effect in Tokio en Osaka, maar de invloed van afstand is anders in Kyoto.

De eerste indruk van de coëfficiënt voor netheid suggereert een negatieve relatie tussen netheid en de responsvariabele in het model, wat tegen de intuïtie lijkt in te gaan. Na de terugwaartse transformatie van de responsvariabele blijkt het echter positief gerelateerd te zijn aan de responsvariabele.

Wat betreft invloedrijke punten hebben we ook geprobeerd deze te verwijderen, maar er verschenen nieuwe invloedrijke punten, en het resultaat van de modelselectie bleef hetzelfde. In deze situatie hebben we besloten de invloedrijke punten niet te verwijderen. Toekomstig onderzoek naar regressieanalyse zou gebruik kunnen maken van fractionele calculus [1]-[3], een innovatief hulpmiddel om tussenliggende of overgangsgedragingen van twee opeenvolgende regressiemodellen van gehele orde te extraheren. De combinatie van fractionele calculus wordt al breed toegepast in technische en technologische domeinen [5]-[17].

Hoofdstuk 5 References

[1] F. Almeida and R. Silva, "Seasonality and Its Impact on Accommodation Pricing in Urban Destinations," *Journal of Revenue and Pricing Management*, vol. 19, no. 3, pp. 189–201, 2020.

[2] E. Brown et al., "A Deep Learning Framework for Dynamic Pricing in the Sharing Economy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 1987–1999, 2021.

[3] F. Bu, Y. Cai, and Y. Yang, "Multiple object tracking based on faster-RCNN detector and KCF tracker," Technical Report, 2016. [Online]. Available: https://pdfs.semanticscholar.org

[4] Y. Yang and H. H. Zhang, "Stability study of LQR and pole-placement genetic algorithm synthesized input-output feedback linearization controllers for a rotary inverted pendulum system," *International Journal of Engineering Innovations and Research*, vol. 7, no. 1, pp. 62–68, 2018.

[5] C. Garcia et al., "Geospatial Analysis of Accommodation Pricing in Urban Tourism Destinations," *Tourism Management*, vol. 74, pp. 112–125, 2019.

[6] P. W. Holland, Weighted Ridge Regression: Combining Ridge and Robust Regression Methods, *National Bureau of Economic Research*, DOI: 10.3386/w0011.

[7] Y. Yang and H. H. Zhang, *Preliminary Tools of Fractional Calculus*, in *Fractional Calculus with its Applications in Engineering and Technology*, Cham: Springer, 2019, pp. 3–42.

[8] Y. Yang and H. H. Zhang, *Fractional-Order Controller Design*, in *Fractional Calculus with its Applications in Engineering and Technology*, Cham: Springer, 2019, pp. 43–65.

[9] Y. Yang and H. H. Zhang, *Control Applications in Engineering and Technology*, in *Fractional Calculus with its Applications in Engineering and Technology*, Cham: Springer, 2019, pp. 67–89.

[10] Y. Yang, H. H. Zhang, and R. M. Voyles, "Rotary inverted pendulum system tracking and stability control based on input-output feedback linearization and PSO-optimized fractional order PID controller," in *Automatic Control, Mechatronics and Industrial Engineering*, CRC Press, 2019, pp. 79–84.

[11] Y. Yang, H. H. Zhang, W. Yu, and L. Tan, "Optimal design of discrete-time fractional-order PID controller for idle speed control of an IC engine," *International Journal of Powertrains*, vol. 9, nos. 1–2, pp. 79–97, 2020.

[12] Y. Yang, "Electromechanical Characterization of Organic Field-Effect Transistors with Generalized Solid-State and Fractional Drift-Diffusion Models," Doctoral dissertation, Purdue University, 2021.

[13] Y. Yang and H. H. Zhang, "Neural network-based adaptive fractional-order backstepping control of uncertain quadrotors with unknown input delays," *Fractal and Fractional*, vol. 7, no. 3, p. 232, 2023.

[14] Y. Yang and H. H. Zhang, *Fractional Calculus with its Applications in Engineering and Technology*, Morgan & Claypool Publishers, 2019.

[15] Y. Yang and H. H. Zhang, "Optimal model reference adaptive fractional-order proportional integral derivative control of idle speed system under varying disturbances," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 09596518241266670, 2024.

[16] A. Smith and B. Lee, "Predicting Hotel Prices Using Machine Learning: A Comparative Study," *Journal of Hospitality and Tourism Technology*, vol. 12, no. 3, pp. 45–67, 2021.

[17] R. Kumar and S. Patel, "Feature Engineering for Predictive Modeling in Hospitality: A Case Study," *International Journal of Data Science*, vol. 8, no. 2, pp. 89–104, 2020.

[18] L. Nguyen and T. Chen, "The Impact of Online Reviews on Hotel Pricing Strategies," *Information & Management*, vol. 58, no. 4, 103456, 2021.

[19] M. Johnson et al., "Machine Learning Approaches for Tourism Demand Forecasting," *Annals of Tourism Research*, vol. 85, 103103, 2020.

[20] K. Tanaka and Y. Sato, "Economic Factors Influencing Travel Costs in Japan," *Journal of Travel Economics*, vol. 15, no. 1, pp. 34–50, 2022.

[21] G. Wang and H. Kim, "Comparative Analysis of Regression Models for Real Estate Price Prediction," *Expert Systems with Applications*, vol. 168, 114231, 2021.

[22] J. Müller and D. Evans, "The Role of Amenities in Hostel Pricing: A Multinational Analysis," *International Journal of Hospitality Management*, vol. 90, 102601, 2020.

[23] S. Roberts et al., "Leveraging Kaggle Datasets for Predictive Analytics in Tourism," *Data in Brief*, vol. 35, 106789, 2021.

[24] T. Zhang et al., "Ensemble Learning Methods for Improved Accuracy in Price Prediction Models," *Machine Learning*, vol. 110, no. 4, pp. 879–901, 2021.

[25] https://www.kaggle.com/koki25ando/hostel-world-dataset/home

[26] https://www.hostelworld.com

[27] Y. Yang and H. H. Zhang, "Optimal fractional-order proportional–integral–derivative control enabling full actuation of decomposed rotary inverted pendulum system," *Transactions of the Institute of Measurement and Control*, vol. 45, nos. 10, pp. 1986–1998, 2023.

Hoofdstuk 6 Bijlage A: Code

— Bijlage A: moet een complete, georganiseerde R-code bevatten die alle grafieken, diagnostieken, modellen en uitvoer genereert waarnaar in het rapport wordt verwezen. De code moet voldoende geannoteerd zijn om het eenvoudig te maken relevante delen van de code te vinden.

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Read and rearrange the data
```{r, warning=FALSE}
library(stringr) # install stringr for number extraction

hostel <- read.csv('/Users/sgch/Desktop/STAT 512/Project/Hostel.csv', header=T)
```

```
hostel <- subset(hostel, select = -c(1,2,6,7,14,15,16)) # remove old index column, and hostel name column
hostel <- na.omit(hostel) # remove rows that contain one or more NAs

# rapply(hostel, function(x) length(unique(x))) # number of unique values in each column
# table(hostel$City) # city count
hostel <- subset(hostel, City %in% c('Kyoto', 'Osaka', 'Tokyo') ) # get hostels from 'Kyoto', 'Osaka', 'Tokyo'
hostel$X1 <- hostel$City == "Osaka";
hostel$X2 <- hostel$City == "Kyoto";

# distance: extract distance from string and convert it from str type to int
# e.g. 5.9km from city centre -> 5.9
hostel[3] <- rapply(hostel[3], function(x) as.numeric( sub("km from city centre", "", x)) )

rownames(hostel) <- 1:nrow(hostel) # reindex
#hostel

colnames(hostel) <- c("City", "Price", "Distance", "atmosphere", "cleanliness", "facilities", "location", "security", "staff", "X1", "X2")
hostel <- hostel[,c("Price", "Distance", "atmosphere", "cleanliness", "facilities", "location", "security", "staff", "X1", "X2", "City")]
hostel <- subset(hostel, select = -c(11))



# Standardize variables
for (i in 2:8){
  hostel[,i] = (hostel[,i]-mean(hostel[,i]))/sd(hostel[,i])
}
```

Remove outliers and fit the model
Pre plot and check the issues

```{r}
# plot(hostel)
# cor(hostel[,1:8])
hostel.mod <- lm(Price~., hostel)

summary(hostel.mod)
```

Simply dignostic nonlinear, constant variance and normal variance issues.

```{r}
# residual plot
library(car)
residualPlots(hostel.mod, smooth = FALSE)
```

```{r}
# Add higher order terms
hostel1.mod <-
lm(Price~Distance+I(Distance^2)+atmosphere+I(atmosphere^2)+cleanliness+I(cleanliness^2)+facilities+I(facilities^2)+location+security+staff+X1+X2,hostel)

summary(hostel1.mod)
```

```{r}
# resid = residuals(hostel1.mod)
#
# # Constant variance
# library(onewaytests)
# hostel$Group <- cut(hostel$Price,2)
# hostel$residual <- hostel1.mod$residuals
# bf.test(residual~Group, hostel)
#
# # Normality
# shapiro.test(resid)
# qqnorm(resid)
# qqline(resid)
```

```{r}
# Transform Y
library(MASS)
bcmle <- boxcox(hostel1.mod, lambda=seq(-3,3,by=0.01))
lambda <- bcmle$x[which.max(bcmle$y)]
hostel2 = hostel;
hostel2$Distance2 = hostel$Distance^2
hostel2$atmosphere2 = hostel$atmosphere^2
```

```
hostel2$cleanliness2 = hostel$cleanliness^2
hostel2$facilities2 = hostel$facilities^2

hostel2$Price = hostel$Price^lambda;
hostel2.mod <-
lm(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilities+facilities2+location+security+staff+X1+X2,hostel2);
summary(hostel2.mod)
lambda
```

```{r}
resid = residuals(hostel2.mod)

# Constant variance
library(onewaytests)
hostel2$Group <- cut(hostel2$Price,2)
hostel2$residual <- hostel2.mod$residuals
bf.test(residual~Group, hostel2)
hostel2 <- subset(hostel2, select = -c(15,16))
# Normality
shapiro.test(resid)
qqnorm(resid)
qqline(resid)
```

Model selection
Assume indicator only effect intercept

```{r}
library(stats)
step(lm(Price~1,data = hostel2), scope=~
Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilities+facilities2+location+security+staff+X1+X2, method = "both")
```

```{r}
library(ALSM)
library(leaps)
BestSub(hostel2[,2:14], hostel2$Price, num = 1)
```

Based on the stepwise as well as PRESSp from bestsub method, we choose the model with 7 different parameters which includes 2 different indicators.

stepwise - Get a better model
BestSub - Get smallest PRESSp for prediction

Pay more attention to indicator terms
Add more terms to consider the effect of indicator

```{r}
step(lm(Price~1,data = hostel2), scope=~
Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilities+facilities2+location+security+staff+X1+X2+X1*Distance+X1*Distance2+X1*atmosphere+X1*atmosphere2+X1*cleanliness+X1*cleanliness2+X1*facilities+X1*facilities2+X1*location+X1*security+X1*staff+X2*Distance+X2*Distance2+X2*atmosphere+X2*atmosphere2+X2*cleanliness+X2*cleanliness2+X2*facilities+X2*facilities2+X2*location+X2*security+X2*staff, method =
"both")

ori.mod=lm(formula = Price ~ X2 + cleanliness + facilities2 + X1 + Distance +
    Distance2 + facilities + staff + X2:Distance + X2:Distance2,
    data = hostel2)

```

Since we solve the normality issue, we can use weighted regression to deal with the unconstant variance issue.

```{r}
hostel3.mod <-
lm(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X2*Distance2,hostel2);
summary(hostel3.mod)
resid = residuals(hostel3.mod)

wts1 <-
1/fitted(lm(abs(resid)~hostel2$Distance+hostel2$Distance2+hostel2$cleanliness+hostel2$facilities2+hostel2$staff+hostel2$X1+hostel2$X2+hostel2$X2*hostel2$Distance+hostel2$X2*hostel2$Distance2))^2
hostel.weight<-
lm(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X2*Distance2, data = hostel2, weights = wts1)
summary(hostel.weight)
```

```
```
You can consider whether we can drop variable Staff?
** T value for weighted regression doesn't make sense.

Also we can analysis the multicolinearity issue based on VIF here, to show why we can drop other variables, they are trival or have linear relationship with other variables.

Influential point & Outliers
```{r}
alpha = 0.05
p = 9
n = 296
qt(1-alpha/2/n,n-1-p)
library(car)
influencePlot(hostel.weight)
plot(hostel.weight,pch = 18, col = "red", which = c(4))
plot(hostel.weight,pch = 18,which = c(1))
hostel.unweight<-
lm(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X2*Distance2, data = hostel2)
plot(hostel.unweight,pch = 18,which = c(1))
# axis(side=1,at=seq(0.195,0.235,0.005),lwd=3)
#plot(ori.mod)
```
No outliers in Y, some outliers in X. Can apply another window to cancel the influence of outerlier x.
```{r}
resid = residuals(hostel.weight)

# Normality
shapiro.test(resid)
qqnorm(resid)
qqline(resid)
```
```{r}
library(fmsb)
VIF(lm(staff~Distance+Distance2+cleanliness+facilities2+X2+X1+X2:Distance+X2:Distance2, data = hostel2))
```

Directly apply rigid regression (Ignore multicollinearity issue)

```{r}
library(MASS)
library(car)
library(leaps)
library(caret)
library(ggplot2)
library(lmridge)
# For prediction, chosse lambda
mod <-
lmridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+faciliti
es+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2,hostel2, lambda =
seq(0,20,0.01))
resid = residuals(mod)

#mod1 =
lm.ridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilit
ies+facilities2+location+security+staff+X1+X2+X1*Distance+X1*Distance2+X1*atmosphere+
X1*atmosphere2+X1*cleanliness+X1*cleanliness2+X1*facilities+X1*facilities2+X1*location+
X1*security+X1*staff+X2*Distance+X2*Distance2+X2*atmosphere+X2*atmosphere2+X2*cl
eanliness+X2*cleanliness2+X2*facilities+X2*facilities2+X2*location+X2*security+X2*staff,h
ostel2, lambda = seq(0,20,0.01), weights = wts2)
wts2 =
1/fitted(lm(abs(resid)~hostel2$Distance+hostel2$Distance2+hostel2$atmosphere+hostel2$atmo
sphere2+hostel2$cleanliness+hostel2$cleanliness2+hostel2$facilities+hostel2$facilities2+hostel
2$location+hostel2$security+hostel2$staff+hostel2$X1+hostel2$X2+hostel2$X2:hostel2$Dista
nce+hostel2$X2:hostel2$Distance2,data = hostel2))^2
mod1 <-
lm.ridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilit
ies+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2,hostel2, lambda =
seq(0,20,0.01), weights = wts2)
plot(mod1)
select(mod1)
# mod2 <-
lmridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+faciliti
es+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2,data=as.data.frame
(hostel2), k = seq(0,20,0.01), weights = wts2)
#
```

```
# plot(mod2)
# vif(mod2)
summary(mod1)

# Can compare the result with rigid regression and selected model. I don't know how to plot and compare them.


train.control<-trainControl(method="cv", number=5)
set.seed(1)
step.model1<-
train(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X2*Distance2, data=hostel2, method="leapBackward",
tuneGrid=data.frame(nvmax=15), trControl=train.control, weights=wts1)

step.model1$results



# mod2 <-
lm.ridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilities+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2,hostel2, lambda = 4.41, weights = wts2)
#
# mod2$coef

# residuals(mod2)

```

```` {r}
# library('MXM')
# hostel2_m = data.matrix(hostel2)
# #hostel2_m = hostel2_m[1:270,]
# step.model2 = ridgereg.cv(hostel2_m[,1], hostel2_m, K = 10, lambda = seq(0, 10, by = 0.1))
# step.model2
set.seed(1)
```

step.model2<-
train(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facilities+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2, data=hostel2, method="ridge", trControl=train.control, weights = wts2, tuneGrid=data.frame(lambda=4.41))

step.model2$results
```

# Hoofdstuk 7 Bijlage B: Uitvoer

## Read and rearrange the data

```
library(stringr) # install stringr for number extraction

hostel <- read.csv('data/Hostel2.csv', header=T)
hostel <- subset(hostel, select = -c(1,2,6,7,14,15,16)) # remove old index column, and hostel n
ame column
hostel <- na.omit(hostel) # remove rows that contain one or more NAs

# rapply(hostel, function(x) length(unique(x))) # number of unique values in each column
# table(hostel$City) # city count
hostel <- subset(hostel, City %in% c('Kyoto', 'Osaka', 'Tokyo') ) # get hostels from 'Kyoto', '
Osaka', 'Tokyo'
hostel$X1 <- hostel$City == "Osaka";
hostel$X2 <- hostel$City == "Kyoto";

# distance: extract distance from string and convert it from str type to int
# e.g. 5.9km from city centre -> 5.9
hostel[3] <- rapply(hostel[3], function(x) as.numeric( sub("km from city centre", "", x)) )

rownames(hostel) <- 1:nrow(hostel) # reindex
#hostel

colnames(hostel) <- c("City", "Price", "Distance", "atmosphere", "cleanliness", "facilities",
"location", "security", "staff", "X1", "X2")
hostel <- hostel[,c("Price", "Distance", "atmosphere", "cleanliness", "facilities", "location
", "security", "staff", "X1", "X2", "City")]
hostel <- subset(hostel, select = -c(11))
```
```

```r
# Standardize variables
for (i in 2:8){
  hostel[,i] = (hostel[,i]-mean(hostel[,i]))/sd(hostel[,i])
}
```

Remove outliers and fit the model Pre plot and check the issues

```r
# plot(hostel)
# cor(hostel[,1:8])
hostel.mod <- lm(Price~., hostel)

summary(hostel.mod)
```

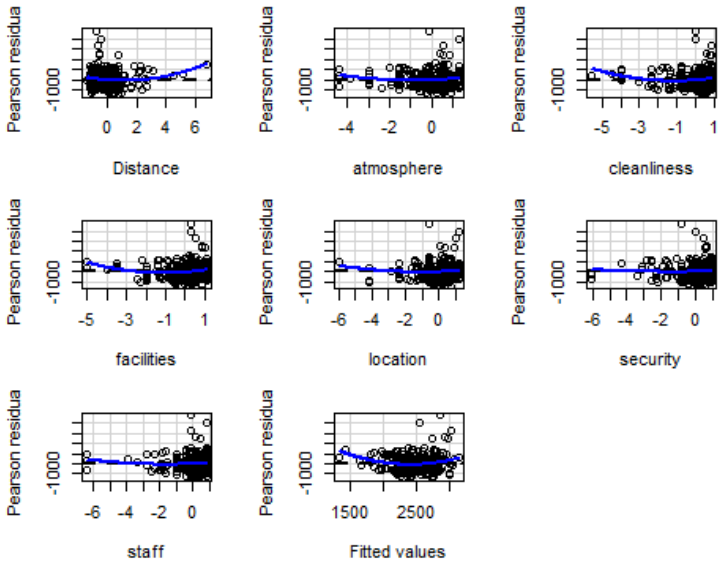
```
##
## Call:
## lm(formula = Price ~ ., data = hostel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1431.6  -570.2  -150.5   414.9  4748.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2791.66      78.79  35.430  < 2e-16 ***
## Distance     -166.17      58.08  -2.861  0.00453 **
## atmosphere     80.96      73.82   1.097  0.27370
## cleanliness   165.82      87.36   1.898  0.05869 .
## facilities   -110.32      99.62  -1.107  0.26906
## location      -13.31      60.28  -0.221  0.82538
## security       29.19      71.52   0.408  0.68347
## staff          11.71      79.52   0.147  0.88307
## X1TRUE       -383.79     115.60  -3.320  0.00102 **
## X2TRUE       -720.11     139.19  -5.174 4.33e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 811.9 on 286 degrees of freedom
## Multiple R-squared:  0.1122, Adjusted R-squared:  0.0843
## F-statistic: 4.018 on 9 and 286 DF,  p-value: 7.414e-05
```

Simply dignostic nonlinear, constant variance and normal variance issues.

```
# residual plot
library(car)

## Loading required package: carData

residualPlots(hostel.mod, smooth = FALSE)
```




```
##             Test stat Pr(>|Test stat|)
## Distance       3.5601       0.0004343 ***
## atmosphere     1.6598       0.0980645 .
## cleanliness    2.6198       0.0092700 **
## facilities     3.1158       0.0020215 **
## location       1.1324       0.2584109
## security       0.4847       0.6282890
## staff          1.2545       0.2106899
## Tukey test     3.7478       0.0001784 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Add higher order terms
hostel1.mod <- lm(Price~Distance+I(Distance^2)+atmosphere+I(atmosphere^2)+cleanliness+I(cleanl
iness^2)+facilities+I(facilities^2)+location+security+staff+X1+X2,hostel)

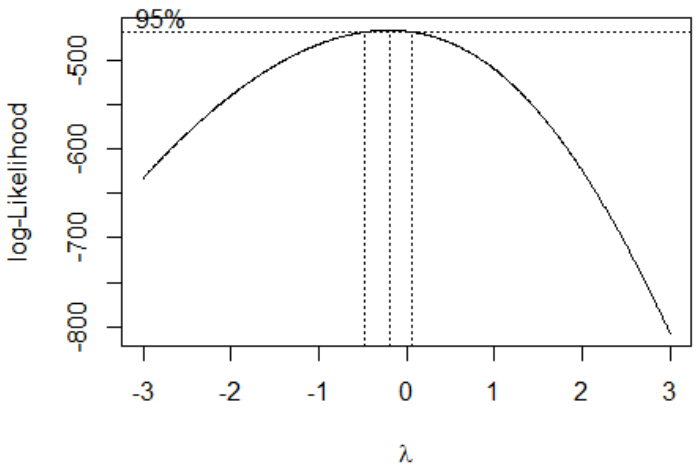
summary(hostel1.mod)

##
## Call:
## lm(formula = Price ~ Distance + I(Distance^2) + atmosphere +
##     I(atmosphere^2) + cleanliness + I(cleanliness^2) + facilities +
```

```
##     I(facilities^2) + location + security + staff + X1 + X2,
##     data = hostel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1386.7  -507.9  -113.9   370.0  4431.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2781.72     81.70  34.047  < 2e-16 ***
## Distance          -412.35     89.16  -4.625 5.72e-06 ***
## I(Distance^2)       73.60     21.98   3.348 0.000924 ***
## atmosphere          52.66     93.30   0.564 0.572906
## I(atmosphere^2)    -18.56     34.36  -0.540 0.589583
## cleanliness        258.04    122.34   2.109 0.035812 *
## I(cleanliness^2)    38.31     40.89   0.937 0.349645
## facilities          11.04    116.16   0.095 0.924355
## I(facilities^2)     59.13     33.50   1.765 0.078693 .
## location           -35.80     60.04  -0.596 0.551424
## security           -39.53     73.82  -0.536 0.592712
## staff               91.19     81.53   1.119 0.264300
## X1TRUE            -546.68    121.05  -4.516 9.26e-06 ***
## X2TRUE           -1070.62    161.56  -6.627 1.74e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 787.4 on 282 degrees of freedom
## Multiple R-squared:  0.1766, Adjusted R-squared:  0.1386
## F-statistic: 4.652 on 13 and 282 DF,  p-value: 2.884e-07

# resid = residuals(hostel1.mod)
#
# # Constant variance
# library(onewaytests)
# hostel$Group <- cut(hostel$Price,2)
# hostel$residual <- hostel1.mod$residuals
# bf.test(residual~Group, hostel)
#
# # Normality
# shapiro.test(resid)
# qqnorm(resid)
# qqline(resid)
```

```
# Transform Y
library(MASS)
bcmle <- boxcox(hostel1.mod, lambda=seq(-3,3,by=0.01))
```




```
lambda <- bcmle$x[which.max(bcmle$y)]
hostel2 = hostel;
hostel2$Distance2 = hostel$Distance^2
hostel2$atmosphere2 = hostel$atmosphere^2
hostel2$cleanliness2 = hostel$cleanliness^2
hostel2$facilities2 = hostel$facilities^2

hostel2$Price = hostel$Price^lambda;
hostel2.mod <- lm(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+fac
ilities+facilities2+location+security+staff+X1+X2,hostel2);
summary(hostel2.mod)

##
## Call:
## lm(formula = Price ~ Distance + Distance2 + atmosphere + atmosphere2 +
##     cleanliness + cleanliness2 + facilities + facilities2 + location +
##     security + staff + X1 + X2, data = hostel2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.038959 -0.007798  0.000074  0.007945  0.036827
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   2.076e-01  1.262e-03 164.488  < 2e-16 ***
## Distance      5.849e-03  1.377e-03   4.246 2.95e-05 ***
## Distance2    -1.086e-03  3.396e-04  -3.197  0.00155 **
## atmosphere   -5.039e-04  1.441e-03  -0.350  0.72688
## atmosphere2   3.301e-04  5.308e-04   0.622  0.53459
## cleanliness  -4.634e-03  1.890e-03  -2.452  0.01481 *
## cleanliness2 -7.428e-04  6.317e-04  -1.176  0.24065
## facilities    2.155e-04  1.794e-03   0.120  0.90448
## facilities2  -1.009e-03  5.176e-04  -1.949  0.05226 .
## location      2.377e-04  9.275e-04   0.256  0.79791
## security      5.517e-05  1.140e-03   0.048  0.96145
## staff        -1.652e-03  1.259e-03  -1.311  0.19081
## X1TRUE        7.967e-03  1.870e-03   4.261 2.78e-05 ***
## X2TRUE        1.671e-02  2.496e-03   6.694 1.17e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01216 on 282 degrees of freedom
## Multiple R-squared:  0.1912, Adjusted R-squared:  0.1539
## F-statistic: 5.128 on 13 and 282 DF,  p-value: 3.5e-08

lambda

## [1] -0.2

resid = residuals(hostel2.mod)

# Constant variance
library(onewaytests)
hostel2$Group <- cut(hostel2$Price,2)
hostel2$residual <- hostel2.mod$residuals
bf.test(residual~Group, hostel2)

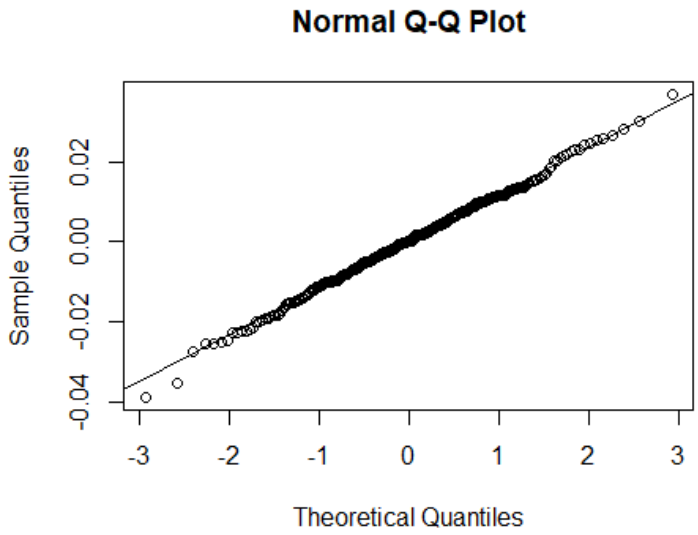
##
##   Brown-Forsythe Test
## -------------------------------------------------------------
##   data : residual and Group
##
##   statistic  : 368.1816
##   num df     : 1
##   denom df   : 293.0368
##   p.value    : 1.026002e-53
##
```

```
##   Result    : Difference is statistically significant.
## ------------------------------------------------------------

hostel2 <- subset(hostel2, select = -c(15,16))
# Normality
shapiro.test(resid)

##
##  Shapiro-Wilk normality test
##
## data:  resid
## W = 0.99755, p-value = 0.9393

qqnorm(resid)
qqline(resid)
```




Model selection Assume indicator only effect intercept

```
library(stats)
step(lm(Price~1,data = hostel2), scope=~ Distance+Distance2+atmosphere+atmosphere2+cleanliness
+cleanliness2+facilities+facilities2+location+security+staff+X1+X2, method = "both")

## Step:  AIC=-2606.14
## Price ~ X2 + cleanliness + facilities2 + X1 + Distance + Distance2 +
##     staff
##
##               Df Sum of Sq      RSS     AIC
## <none>                     0.042079 -2606.1
## + cleanliness2  1 0.0001567 0.041922 -2605.2
```

```
## + atmosphere    1 0.0001260 0.041953 -2605.0
## + atmosphere2   1 0.0000520 0.042027 -2604.5
## + facilities    1 0.0000334 0.042045 -2604.4
## - staff         1 0.0005566 0.042635 -2604.2
## + security      1 0.0000072 0.042072 -2604.2
## + location      1 0.0000002 0.042079 -2604.1
## - Distance2     1 0.0015275 0.043606 -2597.6
## - cleanliness   1 0.0017399 0.043819 -2596.1
## - facilities2   1 0.0019927 0.044071 -2594.4
## - X1            1 0.0028132 0.044892 -2589.0
## - Distance      1 0.0028827 0.044961 -2588.5
## - X2            1 0.0067067 0.048785 -2564.4

##
## Call:
## lm(formula = Price ~ X2 + cleanliness + facilities2 + X1 + Distance +
##     Distance2 + staff, data = hostel2)
##
## Coefficients:
## (Intercept)       X2TRUE  cleanliness  facilities2       X1TRUE
##    0.207458     0.016511    -0.003590    -0.001231     0.007950
##    Distance    Distance2        staff
##    0.005660    -0.001068    -0.001880

library(ALSM)

## Loading required package: leaps

## Loading required package: SuppDists

library(leaps)
BestSub(hostel2[,2:14], hostel2$Price, num = 1)

##     p 1 2 3 4 5 6 7 8 9 A B C D       SSEp         r2     r2.adj        Cp
## 1   2 0 0 0 0 0 0 0 0 1 0 0 0 0 0.04982295 0.03425374 0.03096889 44.723945
## 2   3 0 0 1 0 0 0 0 0 1 0 0 0 0 0.04802296 0.06914407 0.06279010 34.558835
## 3   4 0 0 1 0 0 0 0 0 1 0 0 0 1 0.04642466 0.10012480 0.09087951 25.756873
## 4   5 0 0 1 0 0 0 0 1 1 0 0 0 1 0.04534102 0.12112964 0.10904895 20.433177
## 5   6 1 0 1 0 0 0 0 1 1 1 0 0 0 0.04416931 0.14384156 0.12908021 14.514278
## 6   7 1 0 1 0 0 0 0 1 1 1 0 0 1 0.04263535 0.17357503 0.15641742  6.147195
## 7   8 1 0 1 0 0 0 1 1 1 1 0 0 1 0.04207872 0.18436458 0.16454011  4.385234
## 8   9 1 0 1 0 0 0 1 1 1 1 0 1 1 0.04192203 0.18740174 0.16475092  5.326277
## 9  10 1 0 1 0 0 0 1 1 1 1 1 1 1 0.04175528 0.19063393 0.16516437  6.199318
## 10 11 1 1 1 0 0 0 1 1 1 1 1 1 1 0.04174123 0.19090633 0.16251708  8.104341
```

```
## 11 12 1 1 1 0 1 0 1 1 1 1 1 1 1 0.04172889 0.19114556 0.15981669 10.020931
## 12 13 1 1 1 1 1 0 1 1 1 1 1 1 1 0.04172614 0.19119888 0.15690342 12.002340
## 13 14 1 1 1 1 1 1 1 1 1 1 1 1 1 0.04172579 0.19120559 0.15392074 14.000000
##        AICp      SBCp     PRESSp
## 1  -2568.133 -2560.752 0.05052031
## 2  -2577.025 -2565.954 0.04905521
## 3  -2585.044 -2570.283 0.04750546
## 4  -2590.035 -2571.583 0.04667443
## 5  -2595.785 -2573.643 0.04604290
## 6  -2604.248 -2578.415 0.04442347
## 7  -2606.137 -2576.615 0.04428261
## 8  -2605.242 -2572.029 0.04489697
## 9  -2604.421 -2567.518 0.04476490
## 10 -2602.521 -2561.927 0.04515738
## 11 -2600.609 -2556.324 0.04560945
## 12 -2598.628 -2550.653 0.04598606
## 13 -2596.631 -2544.966 0.04631850
```

Based on the stepwise as well as PRESSp from bestsub method, we choose the model with 7 different parameters which includes 2 different indicators. stepwise - Get a better model BestSub - Get smallest PRESSp for prediction

Pay more attention to indicator terms Add more terms to consider the effect of indicator

```
step(lm(Price~1,data = hostel2), scope=~ Distance+Distance2+atmosphere+atmosphere2+cleanliness
+cleanliness2+facilities+facilities2+location+security+staff+X1+X2+X1*Distance+X1*Distance2+X
1*atmosphere+X1*atmosphere2+X1*cleanliness+X1*cleanliness2+X1*facilities+X1*facilities2+X1*lo
cation+X1*security+X1*staff+X2*Distance+X2*Distance2+X2*atmosphere+X2*atmosphere2+X2*cleanlin
ess+X2*cleanliness2+X2*facilities+X2*facilities2+X2*location+X2*security+X2*staff, method = "b
oth")
```

```
## Start:  AIC=-2559.82
## Step:  AIC=-2622.98
## Price ~ X2 + cleanliness + facilities2 + X1 + Distance + Distance2 +
##     staff + X2:Distance + X2:Distance2
##
##                 Df Sum of Sq      RSS     AIC
## <none>                       0.039217 -2623.0
## + facilities      1 0.0001836 0.039034 -2622.4
## + cleanliness:X2  1 0.0001666 0.039051 -2622.2
## + cleanliness2    1 0.0001599 0.039058 -2622.2
## + atmosphere      1 0.0001407 0.039077 -2622.1
## + atmosphere2     1 0.0001053 0.039112 -2621.8
## + facilities2:X2  1 0.0000908 0.039127 -2621.7
```

```
## + staff:X2        1 0.0000868 0.039131 -2621.6
## + security        1 0.0000459 0.039172 -2621.3
## + facilities2:X1  1 0.0000356 0.039182 -2621.2
## + staff:X1        1 0.0000320 0.039185 -2621.2
## + location        1 0.0000193 0.039198 -2621.1
## + Distance:X1     1 0.0000083 0.039209 -2621.0
## + Distance2:X1    1 0.0000068 0.039211 -2621.0
## + cleanliness:X1  1 0.0000044 0.039213 -2621.0
## - staff           1 0.0007577 0.039975 -2619.3
## - cleanliness     1 0.0016485 0.040866 -2612.8
## - X2:Distance2    1 0.0019758 0.041193 -2610.4
## - facilities2     1 0.0022344 0.041452 -2608.6
## - X2:Distance     1 0.0028603 0.042078 -2604.1
## - X1              1 0.0038828 0.043100 -2597.0

##
## Call:
## lm(formula = Price ~ X2 + cleanliness + facilities2 + X1 + Distance +
##     Distance2 + staff + X2:Distance + X2:Distance2, data = hostel2)
##
## Coefficients:
##      (Intercept)           X2TRUE      cleanliness      facilities2
##         0.206693         0.004911         -0.003495         -0.001306
##           X1TRUE         Distance        Distance2            staff
##         0.009519         0.008210         -0.001565         -0.002200
##  X2TRUE:Distance  X2TRUE:Distance2
##        -0.015195         0.003373
```

```
ori.mod=lm(formula = Price ~ X2 + cleanliness + facilities2 + X1 + Distance +
    Distance2 + facilities + staff + X2:Distance + X2:Distance2,
    data = hostel2)
```

Since we solve the normality issue, we can use weighted regression to deal with the unconstant variance issue.

```
hostel3.mod <- lm(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X2*
Distance2,hostel2);
summary(hostel3.mod)

##
## Call:
## lm(formula = Price ~ Distance + Distance2 + cleanliness + facilities2 +
##     staff + X1 + X2 + X2 * Distance + X2 * Distance2, data = hostel2)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.037853 -0.007953 -0.000240  0.007738  0.032541
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.2066934  0.0011918 173.430  < 2e-16 ***
## Distance          0.0082096  0.0013582   6.044 4.67e-09 ***
## Distance2        -0.0015651  0.0003475  -4.503 9.75e-06 ***
## cleanliness      -0.0034954  0.0010081  -3.467 0.000606 ***
## facilities2      -0.0013059  0.0003235  -4.037 6.97e-05 ***
## staff            -0.0021999  0.0009359  -2.351 0.019419 *
## X1TRUE            0.0095186  0.0017888   5.321 2.08e-07 ***
## X2TRUE            0.0049115  0.0035237   1.394 0.164452
## Distance:X2TRUE  -0.0151950  0.0033270  -4.567 7.35e-06 ***
## Distance2:X2TRUE  0.0033726  0.0008885   3.796 0.000180 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01171 on 286 degrees of freedom
## Multiple R-squared:  0.2398, Adjusted R-squared:  0.2159
## F-statistic: 10.03 on 9 and 286 DF,  p-value: 2.167e-13

resid = residuals(hostel3.mod)

wts1 <- 1/fitted(lm(abs(resid)~hostel2$Distance+hostel2$Distance2+hostel2$cleanliness+hostel2
$facilities2+hostel2$staff+hostel2$X1+hostel2$X2+hostel2$X2*hostel2$Distance+hostel2$X2*hoste
l2$Distance2))^2
hostel.weight<-lm(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X2
*Distance2, data = hostel2, weights = wts1)
summary(hostel.weight)

##
## Call:
## lm(formula = Price ~ Distance + Distance2 + cleanliness + facilities2 +
##     staff + X1 + X2 + X2 * Distance + X2 * Distance2, data = hostel2,
##     weights = wts1)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2134 -0.8881 -0.0249  0.8474  3.3048
##
```

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.2067789  0.0011500 179.803  < 2e-16 ***
## Distance          0.0079569  0.0012951   6.144 2.69e-09 ***
## Distance2        -0.0014987  0.0003033  -4.941 1.33e-06 ***
## cleanliness      -0.0036837  0.0010287  -3.581 0.000402 ***
## facilities2      -0.0012715  0.0002867  -4.435 1.32e-05 ***
## staff            -0.0019466  0.0009023  -2.157 0.031815 *
## X1TRUE            0.0093362  0.0017346   5.382 1.53e-07 ***
## X2TRUE            0.0034972  0.0027138   1.289 0.198547
## Distance:X2TRUE  -0.0160599  0.0026425  -6.078 3.89e-09 ***
## Distance2:X2TRUE  0.0036200  0.0006059   5.975 6.83e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.264 on 286 degrees of freedom
## Multiple R-squared:  0.3198, Adjusted R-squared:  0.2983
## F-statistic: 14.94 on 9 and 286 DF,  p-value: < 2.2e-16
```

You can consider whether we can drop variable Staff? ** T value for weighted regression doesn't make sense.

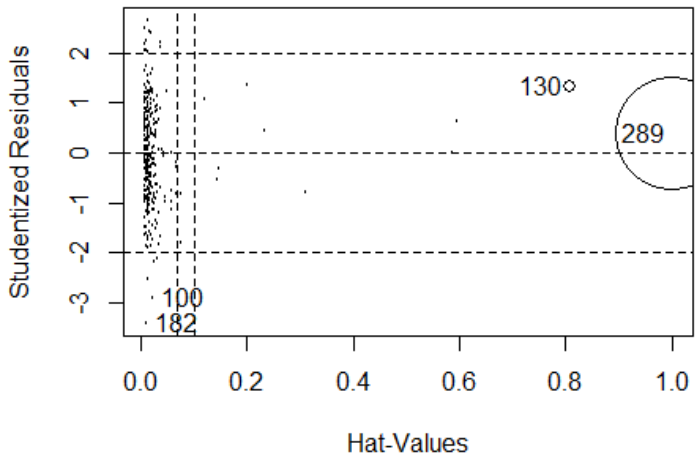
Also we can analysis the multicolinearity issue based on VIF here, to show why we can drop other variables, they are trival or have linear relationship with other variables.

Influential point & Outliers

```
alpha = 0.05
p = 9
n = 296
qt(1-alpha/2/n,n-1-p)

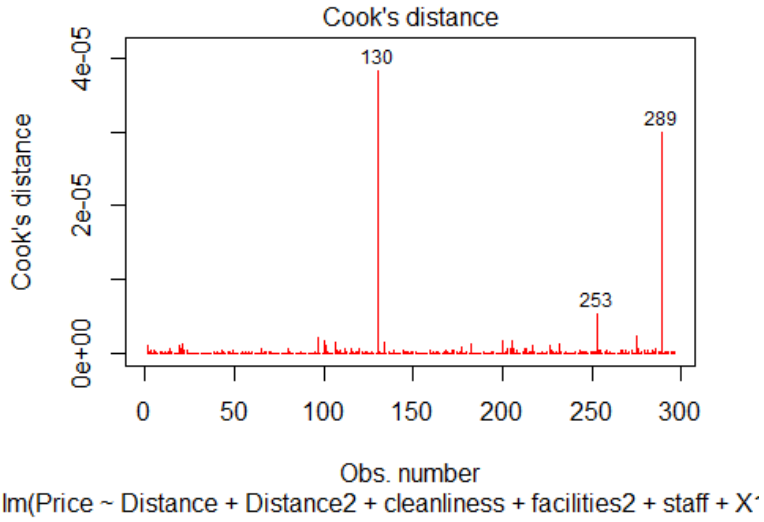
## [1] 3.811871

library(car)
influencePlot(hostel.weight)
```

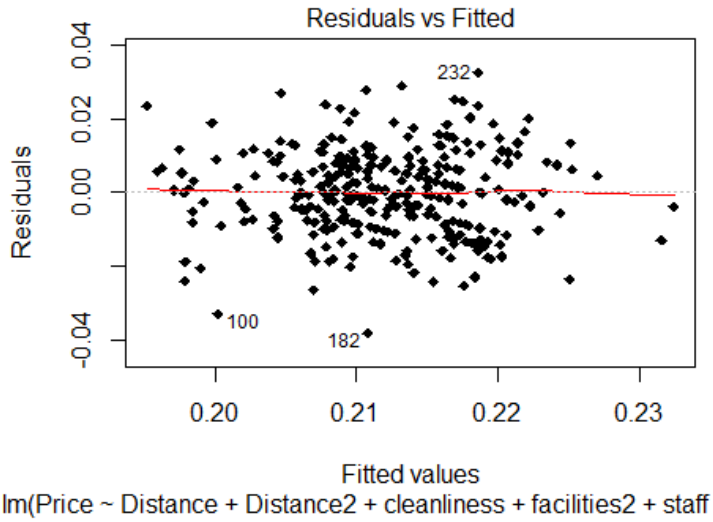



```
##       StudRes        Hat       CookD
## 100 -2.907301 0.02467940  0.02084470
## 130  1.325264 0.80773903  0.73593180
## 182 -3.418714 0.01416258  0.01618562
## 289  0.377428 0.99974425 55.85297206

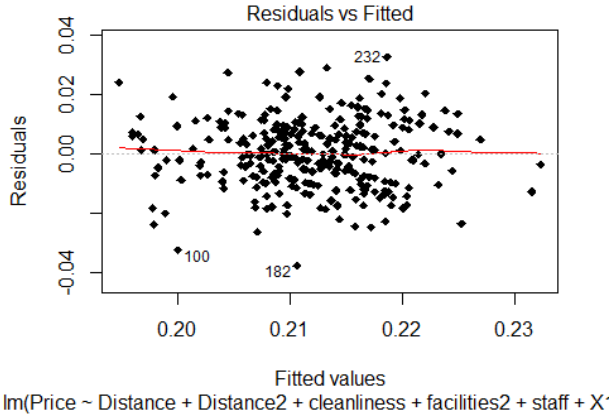
plot(hostel.weight,pch = 18, col = "red", which = c(4))
```




```
plot(hostel.weight,pch = 18,which = c(1))
```




```
hostel.unweight<-lm(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+
X2*Distance2, data = hostel2)
plot(hostel.unweight,pch = 18,which = c(1))
```




No outliers in Y, some outliers in X. Can apply another window to cancel the influence of outerlier x.

```
resid = residuals(hostel.weight)

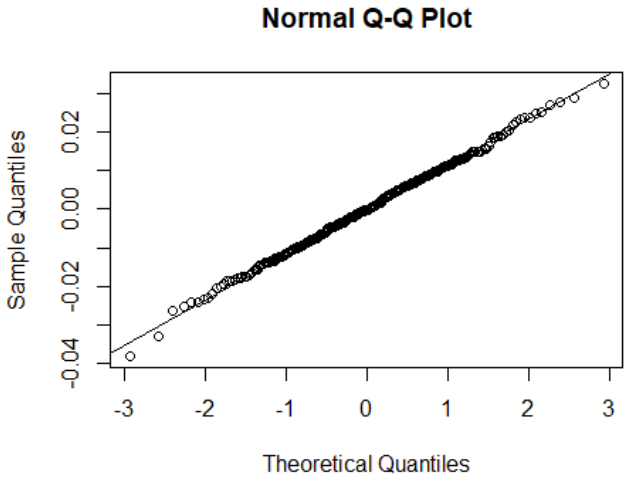


# Normality
shapiro.test(resid)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  resid
## W = 0.99826, p-value = 0.9907

qqnorm(resid)
qqline(resid)
```




```
library(fmsb)
VIF(lm(staff~Distance+Distance2+cleanliness+facilities2+X2+X1+X2:Distance+X2:Distance2, data =
 hostel2))

## [1] 1.884215
```

Directly apply rigid regression (Ignore multicollinearity issue)

```
library(MASS)
library(car)
library(leaps)
library(caret)

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:ALSM':
##
##     oneway

## Loading required package: ggplot2

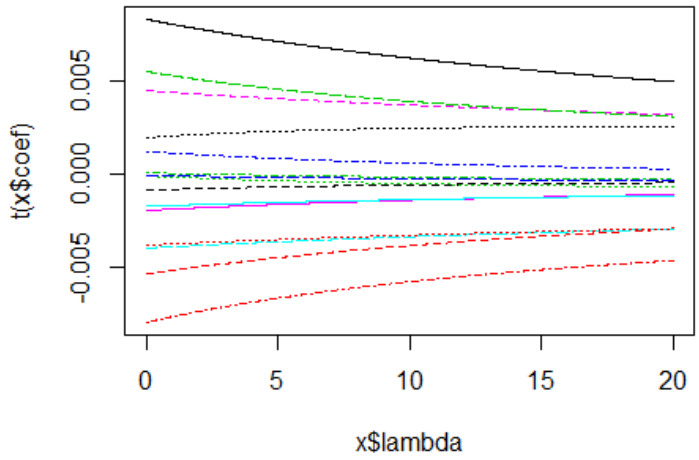
library(ggplot2)
library(lmridge)
```

```
##
## Attaching package: 'lmridge'

## The following object is masked from 'package:car':
##
##     vif

# For prediction, chosse lambda
mod <- lmridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+facili
ties+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2,hostel2, lambda = seq
(0,20,0.01))
resid = residuals(mod)

wts2 =  1/fitted(lm(abs(resid)~hostel2$Distance+hostel2$Distance2+hostel2$atmosphere+hostel2$a
tmosphere2+hostel2$cleanliness+hostel2$cleanliness2+hostel2$facilities+hostel2$facilities2+ho
stel2$location+hostel2$security+hostel2$staff+hostel2$X1+hostel2$X2+hostel2$X2:hostel2$Distan
ce+hostel2$X2:hostel2$Distance2,data = hostel2))^2
mod1 <- lm.ridge(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+faci
lities+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2,hostel2, lambda = se
q(0,20,0.01), weights = wts2)
plot(mod1)
```




```
select(mod1)

## modified HKB estimator is 6.976383
## modified L-W estimator is 41.48205
## smallest value of GCV  at 4.41

summary(mod1)
```

```
##        Length Class  Mode
## coef   30015  -none- numeric
## scales    15  -none- numeric
## Inter      1  -none- numeric
## lambda  2001  -none- numeric
## ym         1  -none- numeric
## xm        15  -none- numeric
## GCV     2001  -none- numeric
## kHKB       1  -none- numeric
## kLW        1  -none- numeric


train.control<-trainControl(method="cv", number=5)
set.seed(1)
step.model1<-train(Price~Distance+Distance2+cleanliness+facilities2+staff+X1+X2+X2*Distance+X
2*Distance2, data=hostel2, method="leapBackward",
tuneGrid=data.frame(nvmax=15), trControl=train.control, weights=wts1)

step.model1$results

##   nvmax      RMSE  Rsquared        MAE       RMSESD RsquaredSD
## 1    15 0.0119834 0.1895096 0.009611793 0.0004054488 0.08933626
##          MAESD
## 1 0.0004403936



set.seed(1)
step.model2<-train(Price~Distance+Distance2+atmosphere+atmosphere2+cleanliness+cleanliness2+f
acilities+facilities2+location+security+staff+X1+X2+X2*Distance+X2*Distance2, data=hostel2, me
thod="ridge", trControl=train.control, weights = wts2, tuneGrid=data.frame(lambda=4.41))

step.model2$results

##   lambda       RMSE   Rsquared        MAE       RMSESD RsquaredSD
## 1   4.41 0.01351856 0.05510238 0.01071577 0.0005667828 0.04291188
##          MAESD
## 1 0.0003764873
```