

# BAN440 - Term Paper Code

Candidate numbers: xx, xx, xx

## Table of contents

Packages used	1
Demand estimation	1

## Packages used

```
# relevant libraries
library(tidyverse)
library(readxl)
library(fastDummies)
library(knitr)
library(stargazer)
library(caret)
library(here)
```

File path:

```
# Construct the path to the "Vinmonopolet" directory
vinmonopolet_path <- here("Data", "Vinmonopolet", "demand_data.xlsx")
```

## Demand estimation

```
# Set locale to UTF-8
Sys.setlocale("LC_ALL", "en_US.UTF-8")
```

```
[1] "LC_COLLATE=en_US.UTF-8;LC_CTYPE=en_US.UTF-8;LC_MONETARY=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_MESSAGES=en_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_IDENTIFICATION=en_US.UTF-8"
```

```
Vinmonopolet_market <- read_excel(vinmonopolet_path)

### Data preparation #####

# Narrowing down the data to only contain relevant markets
# Excluding the largest cities because they are not representative

# Filter out the largest cities
demand_data <- Vinmonopolet_market %>%
  filter(Population < 150) %>%
  mutate(Number_of_stores = as.factor(Number_of_stores))

# Train and test split, training data all observations with a store
train_data <- Vinmonopolet_market %>%
  filter(Number_of_stores > 0)

# Test data all observations without a store
test_data <- Vinmonopolet_market %>%
  filter(Number_of_stores == 0)

### Model selection #####

# Forward selection
forward_model <- step(lm(Sales ~ 1, data = train_data),
  scope = ~ Population + Grensehandel + n_stays + Monthly_salary + Area,
  direction = "forward")
```

```
Start: AIC=3281.16
Sales ~ 1
```

	Df	Sum of Sq	RSS	AIC
+ Population	1	238655412	3275767	2263.6
+ Number_of_stores	1	232719071	9212107	2508.6
+ n_stays	1	213396224	28534954	2776.6

+ Grensehandel	1	121971827	119959351	3116.9
+ Monthly_salary	1	46654218	195276960	3232.4
+ Spread	1	4307259	237623919	3278.9
<none>			241931178	3281.2
+ Area	1	1750943	240180236	3281.4

Step: AIC=2263.56

Sales ~ Population

	Df	Sum of Sq	RSS	AIC
+ Number_of_stores	1	655940	2619827	2212.6
+ n_stays	1	611748	2664019	2216.6
+ Grensehandel	1	487930	2787837	2227.3
+ Spread	1	228325	3047442	2248.4
+ Area	1	76705	3199061	2259.9
<none>			3275767	2263.6
+ Monthly_salary	1	2240	3273527	2265.4

Step: AIC=2212.6

Sales ~ Population + Number\_of\_stores

	Df	Sum of Sq	RSS	AIC
+ Grensehandel	1	267494	2352332	2189.1
+ n_stays	1	250512	2369314	2190.8
+ Spread	1	45130	2574697	2210.5
+ Area	1	44973	2574854	2210.5
+ Monthly_salary	1	35351	2584475	2211.4
<none>			2619827	2212.6

Step: AIC=2189.08

Sales ~ Population + Number\_of\_stores + Grensehandel

	Df	Sum of Sq	RSS	AIC
+ n_stays	1	116034	2236298	2179.1
+ Area	1	32495	2319837	2187.8
+ Spread	1	32037	2320296	2187.8
+ Monthly_salary	1	25822	2326510	2188.5
<none>			2352332	2189.1

Step: AIC=2179.09

Sales ~ Population + Number\_of\_stores + Grensehandel + n\_stays

	Df	Sum of Sq	RSS	AIC
+ Monthly_salary	1	94136	2142163	2170.9
+ Spread	1	19793	2216505	2179.0
<none>			2236298	2179.1
+ Area	1	13717	2222581	2179.6

Step: AIC=2170.9

Sales ~ Population + Number\_of\_stores + Grensehandel + n\_stays +  
Monthly\_salary

	Df	Sum of Sq	RSS	AIC
+ Area	1	36877	2105286	2168.8
<none>			2142163	2170.9
+ Spread	1	8522	2133640	2171.9

Step: AIC=2168.78

Sales ~ Population + Number\_of\_stores + Grensehandel + n\_stays +  
Monthly\_salary + Area

	Df	Sum of Sq	RSS	AIC
<none>			2105286	2168.8
+ Spread	1	9654.8	2095631	2169.7

```
summary(forward_model)
```

Call:

```
lm(formula = Sales ~ Population + Number_of_stores + Grensehandel +  
n_stays + Monthly_salary + Area, data = train_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-338.11	-48.12	-6.81	35.54	364.74

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-4.980e+02	1.249e+02	-3.986	9.03e-05 ***
Population	1.358e+01	6.778e-01	20.040	< 2e-16 ***
Number_of_stores	7.744e+01	1.329e+01	5.825	1.91e-08 ***

```

Grensehandel      -4.198e+00  1.369e+00  -3.068  0.00242 **
n_stays           1.780e-01  4.175e-02   4.263  2.94e-05 ***
Monthly_salary    7.956e+00  2.222e+00   3.580  0.00042 ***
Area              1.278e-02  6.366e-03   2.007  0.04590 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 95.67 on 230 degrees of freedom  
 Multiple R-squared: 0.9913, Adjusted R-squared: 0.9911  
 F-statistic: 4367 on 6 and 230 DF, p-value: < 2.2e-16

```

# Backward selection
backward_model <- step(lm(Sales ~ Population + Grensehandel + n_stays + Monthly_salary + Ar
                        data = train_data),
                        direction = "backward")

```

Start: AIC=2169.69

Sales ~ Population + Grensehandel + n\_stays + Monthly\_salary +  
 Area + Number\_of\_stores + Spread

	Df	Sum of Sq	RSS	AIC
- Spread	1	9655	2105286	2168.8
<none>			2095631	2169.7
- Area	1	38010	2133640	2171.9
- Grensehandel	1	86637	2182268	2177.3
- Monthly_salary	1	104814	2200445	2179.3
- n_stays	1	148519	2244150	2183.9
- Number_of_stores	1	253669	2349299	2194.8
- Population	1	3321080	5416711	2392.8

Step: AIC=2168.78

Sales ~ Population + Grensehandel + n\_stays + Monthly\_salary +  
 Area + Number\_of\_stores

	Df	Sum of Sq	RSS	AIC
<none>			2105286	2168.8
- Area	1	36877	2142163	2170.9
- Grensehandel	1	86135	2191421	2176.3
- Monthly_salary	1	117295	2222581	2179.6
- n_stays	1	166368	2271653	2184.8

```
- Number_of_stores 1      310561 2415847 2199.4
- Population        1      3676078 5781364 2406.2
```

```
summary(backward_model)
```

Call:

```
lm(formula = Sales ~ Population + Grensehandel + n_stays + Monthly_salary +
    Area + Number_of_stores, data = train_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-338.11	-48.12	-6.81	35.54	364.74

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	-4.980e+02	1.249e+02	-3.986	9.03e-05	***
Population	1.358e+01	6.778e-01	20.040	< 2e-16	***
Grensehandel	-4.198e+00	1.369e+00	-3.068	0.00242	**
n_stays	1.780e-01	4.175e-02	4.263	2.94e-05	***
Monthly_salary	7.956e+00	2.222e+00	3.580	0.00042	***
Area	1.278e-02	6.366e-03	2.007	0.04590	*
Number_of_stores	7.744e+01	1.329e+01	5.825	1.91e-08	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 95.67 on 230 degrees of freedom

Multiple R-squared: 0.9913, Adjusted R-squared: 0.9911

F-statistic: 4367 on 6 and 230 DF, p-value: < 2.2e-16

```
lm_Area <- lm(Sales ~ Area, data = train_data)
```

```
summary(lm_Area)
```

Call:

```
lm(formula = Sales ~ Area, data = train_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-426.0	-304.2	-213.1	-23.7	12804.5

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	463.91339	88.30921	5.253	3.35e-07 ***
Area	-0.08325	0.06360	-1.309	0.192

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1011 on 235 degrees of freedom

Multiple R-squared: 0.007237, Adjusted R-squared: 0.003013

F-statistic: 1.713 on 1 and 235 DF, p-value: 0.1919

```
# Linear regression model for predicting sales with all the variables
var_test <- lm(Sales ~ Population + Grensehandel + n_stays + Monthly_salary + Area +
               Number_of_stores + Spread,
               data = Vinmonopolet_market)

var_test1 <- lm(Sales ~ Population + Grensehandel + n_stays + Monthly_salary + Area +
                Number_of_stores + Spread,
                data = demand_data)

stargazer(var_test, var_test1, type = "text")
```

```
=====
                        Dependent variable:
-----
                        Sales
(1)                      (2)
-----
Population              14.497***          13.755***
                        (0.490)            (0.606)

Grensehandel            -4.610***          -4.859***
                        (1.117)            (1.106)

n_stays                 0.159***           0.209***
                        (0.034)            (0.036)
```

Monthly_salary	4.740*** (1.518)	2.483 (1.537)
Area	0.010** (0.005)	0.008* (0.004)
Number_of_stores	63.026*** (8.444)	
Number_of_stores1		62.758*** (9.410)
Number_of_stores2		163.133*** (21.375)
Number_of_stores3		449.261*** (39.790)
Number_of_stores4		403.845*** (88.581)
Number_of_stores5		429.765*** (59.108)
Number_of_stores6		816.444*** (90.721)
Spread	-6.040*** (2.236)	-9.520*** (2.455)
Constant	-289.696*** (83.210)	-157.848* (83.383)
-----		
Observations	357	353
R2	0.991	0.958
Adjusted R2	0.991	0.957
Residual Std. Error	79.155 (df = 349)	68.455 (df = 340)
F Statistic	5,737.855*** (df = 7; 349)	651.114*** (df = 12; 340)
=====		



Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

```
# From these regressions we see that we want to remove the "Area" and "prop_spread" variables
# from the regressions as they are not significant.
```

```
### Demand estimation #####
```

```
## Linear regression
```

```
# Predicting sales using the training data
```

```
reg1 <- lm(Sales ~ Population + Grensehandel + n_stays + Monthly_salary,
           data = train_data)
```

```
stargazer(reg1, type = "text")
```

```
=====
                        Dependent variable:
-----
                        Sales
-----
Population                16.449***
                        (0.490)

Grensehandel              -5.264***
                        (1.456)

n_stays                   0.246***
                        (0.043)

Monthly_salary            4.885**
                        (2.290)

Constant                 -270.085**
                        (125.336)

-----
Observations                237
```

```

R2                                0.990
Adjusted R2                        0.990
Residual Std. Error      102.638 (df = 232)
F Statistic              5,683.347*** (df = 4; 232)
=====
Note:                        *p<0.1; **p<0.05; ***p<0.01

```

```

# Applying the model on the test data
test_data$Sales_pred <- predict(reg1, newdata = test_data)

## Merge predicted data into the original data

# Deselect unnecessary columns to merge the data easier
test_data <- test_data %>%
  select(Municipality_Code, Sales_pred)

# Merge the data frames
predicted_data <- Vinmonopolet_market %>%
  left_join(test_data, by = "Municipality_Code") %>%
  mutate(Sales = ifelse(Sales == 0, Sales_pred, Sales)) %>%
  select(-Sales_pred) %>%
  mutate(Sales = ifelse(Sales < 0, 0, Sales),
         Number_of_stores = as.integer(Number_of_stores)) %>%
  filter(Number_of_stores < 2)

```

```
stargazer(reg1, type = "text")
```

```
=====
                        Dependent variable:
                        -----
                        Sales
                        -----
Population                16.449***
                        (0.490)

Grensehandel              -5.264***
                        (1.456)

n_stays                   0.246***
                        (0.043)

Monthly_salary            4.885**
                        (2.290)

Constant                 -270.085**
                        (125.336)

-----
Observations                237
R2                          0.990
Adjusted R2                 0.990
Residual Std. Error    102.638 (df = 232)
F Statistic             5,683.347*** (df = 4; 232)
=====
Note:                      *p<0.1; **p<0.05; ***p<0.01
```

```
## Logit regression #####

# 1) Make sure the factor for Number_of_stores has valid R variable names
#   that won't cause errors in caret. For instance, rename "0" -> "NoStore"
#   and "1" -> "OneStore".
data_for_logit <- predicted_data %>%
  mutate(Number_of_stores = as.factor(Number_of_stores))
```

```

# Rename factor levels (originally "0" and "1") to "NoStore" and "OneStore"
data_for_logit$Number_of_stores <- factor(
  data_for_logit$Number_of_stores,
  levels = c("0", "1"),
  labels = c("NoStore", "OneStore")
)

# 2) Set up k-fold cross-validation parameters
set.seed(123) # for reproducibility

my_control <- trainControl(
  method = "cv",          # k-fold CV
  number = 5,             # 5 folds
  classProbs = TRUE,      # needed for probability output
  summaryFunction = twoClassSummary
)

# 3) Train the logistic model with cross-validation
cv_model <- train(
  Number_of_stores ~ Sales + Population + Grensehandel + n_stays + Monthly_salary,
  data = data_for_logit,
  method = "glm",
  family = binomial,
  trControl = my_control,
  metric = "ROC"          # use AUC (Area Under the Curve) as our metric
)

```

```

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```

```

# 4) Review cross-validation results
print(cv_model)

```

Generalized Linear Model

```

316 samples
  5 predictor
  2 classes: 'NoStore', 'OneStore'

```

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 252, 253, 253, 253, 253

Resampling results:

ROC	Sens	Spec
0.9363622	0.8333333	0.8619231

```
print(cv_model$results)
```

	parameter	ROC	Sens	Spec	ROCSD	SensSD	SpecSD
1	none	0.9363622	0.8333333	0.8619231	0.02761465	0.1141089	0.04721797

```
# 5) Get predicted probabilities from the final trained model
```

```
# caret retrains on the entire dataset after CV by default
```

```
predicted_data$prob_cv <- predict(cv_model, newdata = data_for_logit, type = "prob")[, "OneStore"]
```

```
# 6) Use the probabilities for your recommendations
```

```
recommended_stores <- predicted_data %>%
```

```
  mutate(Number_of_stores = as.integer(as.character(Number_of_stores))) %>%
```

```
  filter(Number_of_stores == 0, Dist_nearest > 15) %>%
```

```
  arrange(desc(prob_cv)) %>%
```

```
  select(Mun_name, prob_cv, Dist_nearest, Sales, Population, Region_Name)
```

```
head(recommended_stores, 10) # for example, show top 10
```

```
# A tibble: 10 x 6
```

	Mun_name	prob_cv	Dist_nearest	Sales	Population	Region_Name
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	aurland	0.840	36.6	81.3	1.84	Vestland
2	eidskog	0.821	23.3	71.5	6.06	Innlandet
3	vaksdal	0.669	19.2	61.4	3.88	Vestland
4	overhalla	0.604	17.9	57.8	3.95	Trøndelag
5	fjord	0.580	16.4	47.6	2.51	Møre og Romsdal
6	aure	0.565	28.0	67.0	3.39	Møre og Romsdal

7	grue	0.550	18.1	47.8	4.51	Innlandet
8	sokndal	0.538	20.9	55.5	3.37	Rogaland
9	bremanger	0.525	20.7	50.2	3.36	Vestland
10	lund	0.482	19.1	41.1	3.26	Rogaland

```
# 7) Output the top 10 recommended stores as a nice table using kable
# And save it
kable(head(recommended_stores, 10), format = "markdown")
```

Mun_name	prob_cv	Dist_nearest	Sales	Population	Region_Name
aurland	0.8398244	36.60803	81.33082	1.836	Vestland
eidskog	0.8206299	23.34241	71.47769	6.059	Innlandet
vaksdal	0.6689958	19.24424	61.39293	3.875	Vestland
overhalla	0.6041112	17.90833	57.79722	3.946	Trøndelag
fjord	0.5798481	16.36717	47.57855	2.506	Møre og Romsdal
aure	0.5645469	28.02924	67.04993	3.394	Møre og Romsdal
grue	0.5496764	18.11747	47.78146	4.513	Innlandet
sokndal	0.5375471	20.88030	55.49285	3.371	Rogaland
bremanger	0.5248066	20.71259	50.19431	3.361	Vestland
lund	0.4823615	19.05946	41.11627	3.259	Rogaland