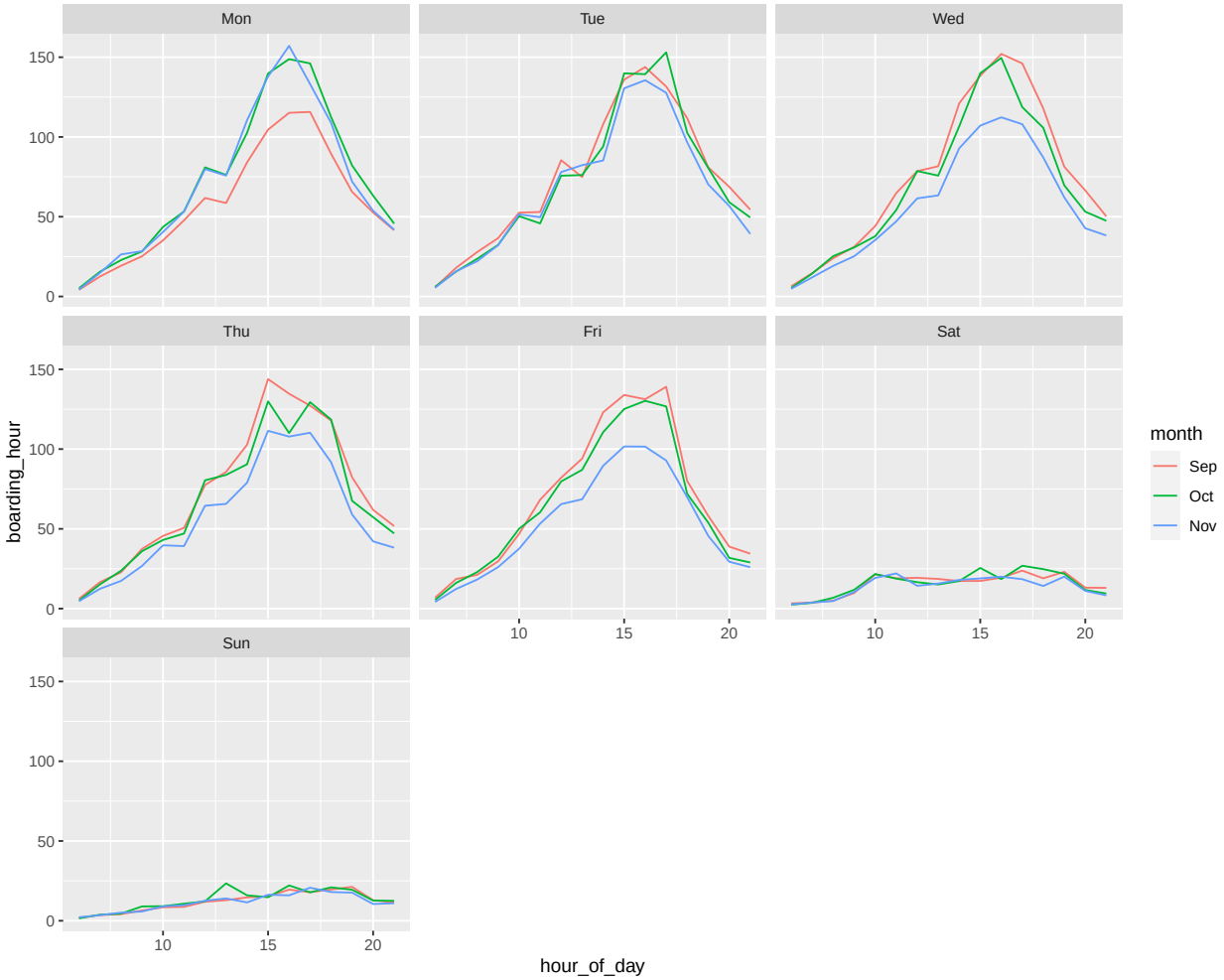
PS2

Yuri Lee, Seunghoon Choi

2021 3 10

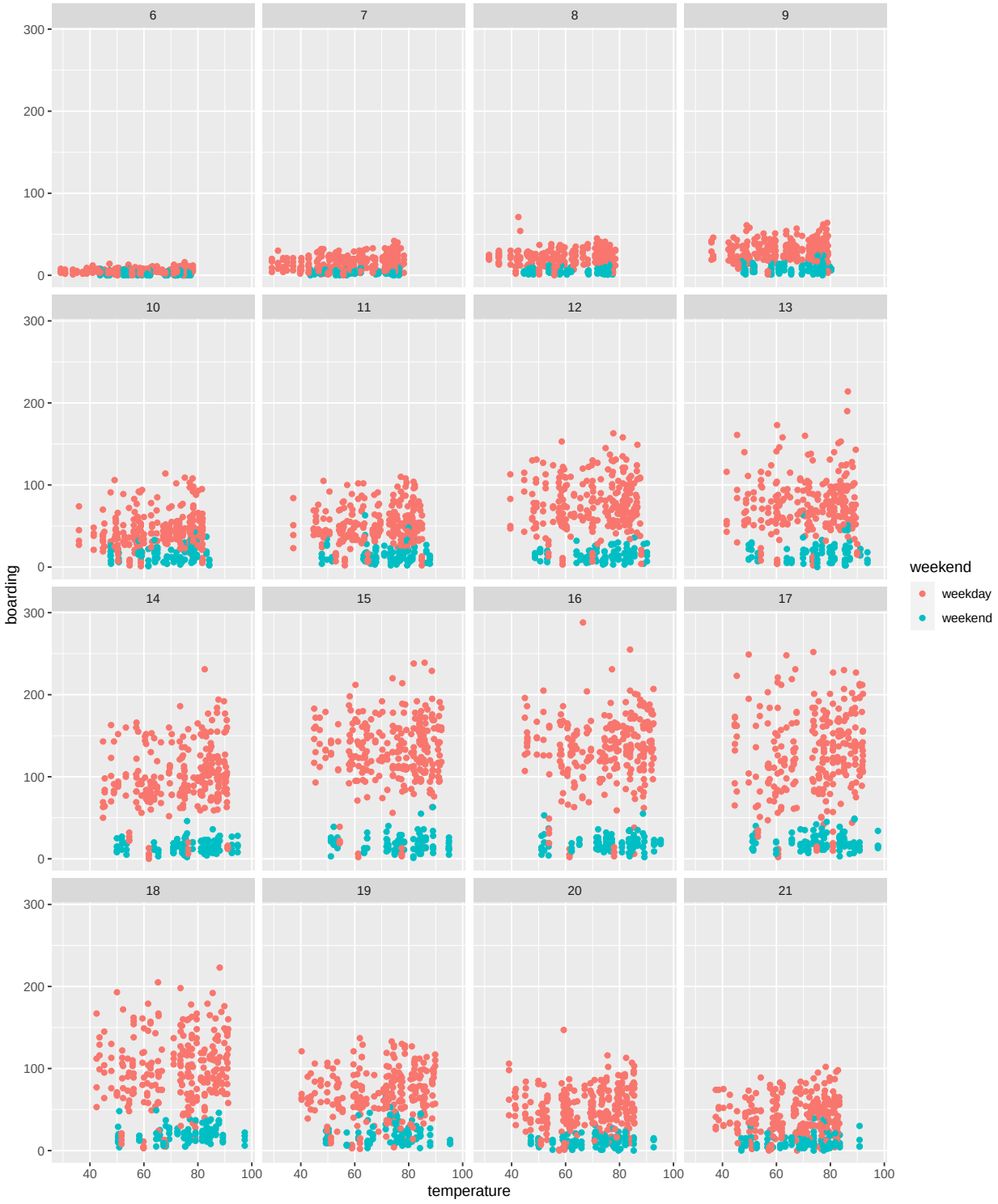
Problem 1: visualization

A. line graphs



Hour of peak boardings is higher at 15~16 on weekdays, but it is more flat on weekends.
Average boardings on Mondays in September look lower, because first monday of September is 'Labor Day'.
Average boardings on Weds/Thurs/Fri in November look lower because of 'Thanksgiving Day

B. Scatter plots



When we hold hour of day and weekend status constant, it does not effect on the number of students riding the bus. But whether weekday or not effects a lot on the number of riding the bus.

Problem 2: Saratoga house prices

A. Linear model

Split data sets(test, train) and Fit the linear models

Lm1: execpt 'pctCollege, sewer, waterfront, landValue, newConstruction'

Lm2: (execpt 'pctCollege, sewer, waterfront, landValue, newConstruction')^2

Lm3: execpt 'pctCollege, sewer, waterfront, landValue, newConstruction'

Lm4: (execpt 'pctCollege, sewer, waterfront, newConstruction') + (age*(landValue+livingArea+bedrooms+bathrooms+newConstr*
+ *(waterfront*(livingArea+bedrooms))

Lm5: (execpt 'pctCollege, sewer') + (age*(landValue+livingArea+bedrooms+bathrooms)) + (water-front*(livingArea+bedrooms)) + (newConstruction*(livingArea+bedrooms))

Lm6: (execpt 'pctCollege' + poly(landValue, 2)) + (age*(poly(landValue, 2) + livingArea+bedrooms+bathrooms+newConstructio*
+ *(waterfront*(livingArea+bedrooms))

Average rmses from 6 models: testing the performances over 10 train/test splits

```
##       V1       V2       V3       V4       V5       V6
## 63936.36 65929.46 58155.59 57099.34 57145.72 56745.65
```

Linear model 6 gives the best prediction, showing the lowest rmse

B. KNN model

Scale the features, and K-fold(5) cross validation(k=5,10,20,30,50)

k=5

```
## [1] 76740.50 73534.25 89700.94 77872.80 87023.10
```

k=10

```
## [1] 77177.67 74098.82 90336.02 77582.68 86418.74
```

k=20

```
## [1] 75862.84 75095.55 90505.96 79192.84 85725.80
```

k=30

```
## [1] 75223.13 75268.75 89864.59 77735.06 85348.18
```

k=50

```
## [1] 75905.66 74896.71 91368.59 78252.17 84854.03
```

Compare Mean of rmses from different ks

```
## [1] 80974.32 81122.79 81276.60 80687.94 81055.43
```

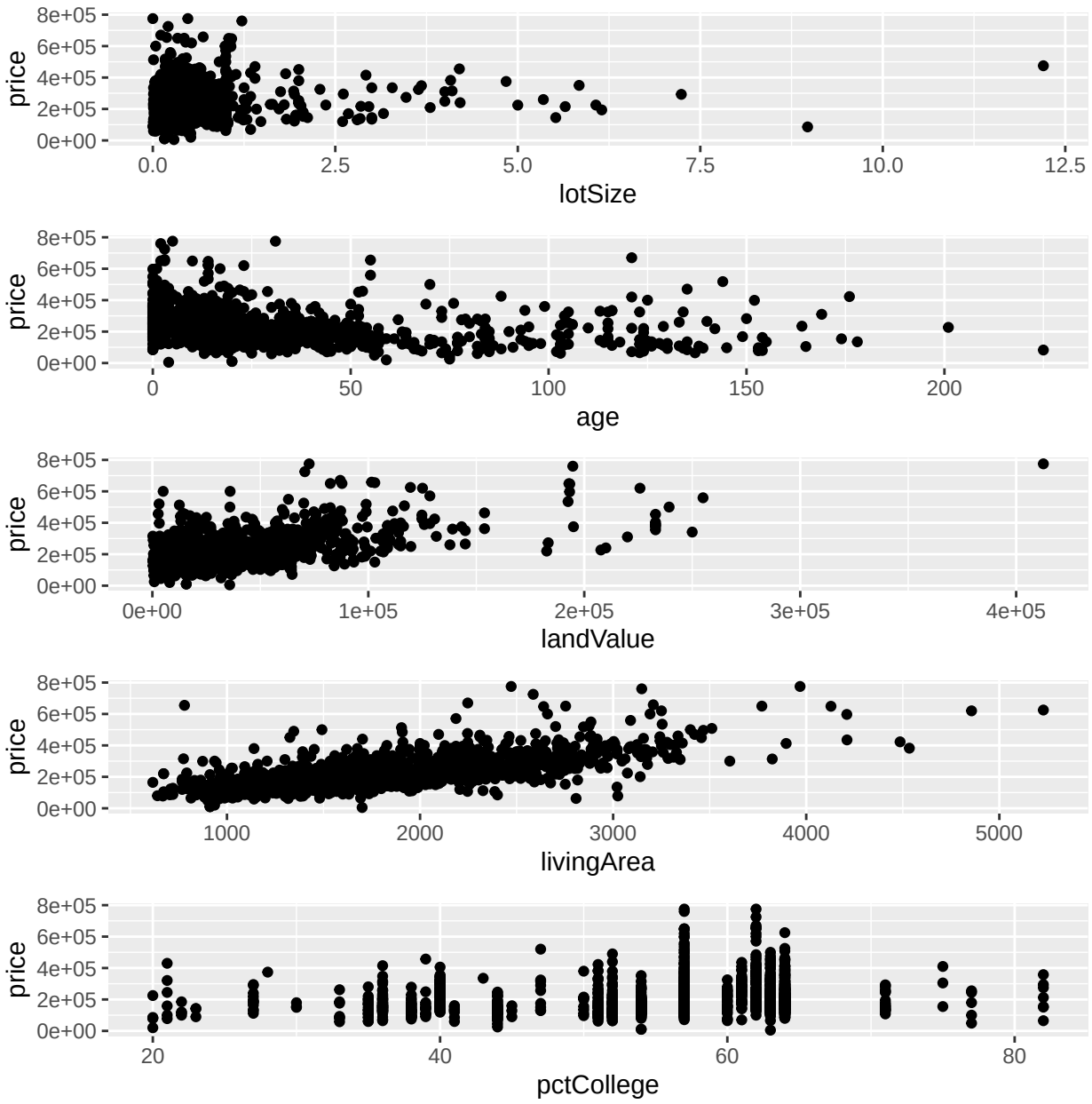
knn5 model whose k-value is 5 shows the best prediction among the five knn-models, which gives the lowest rmse

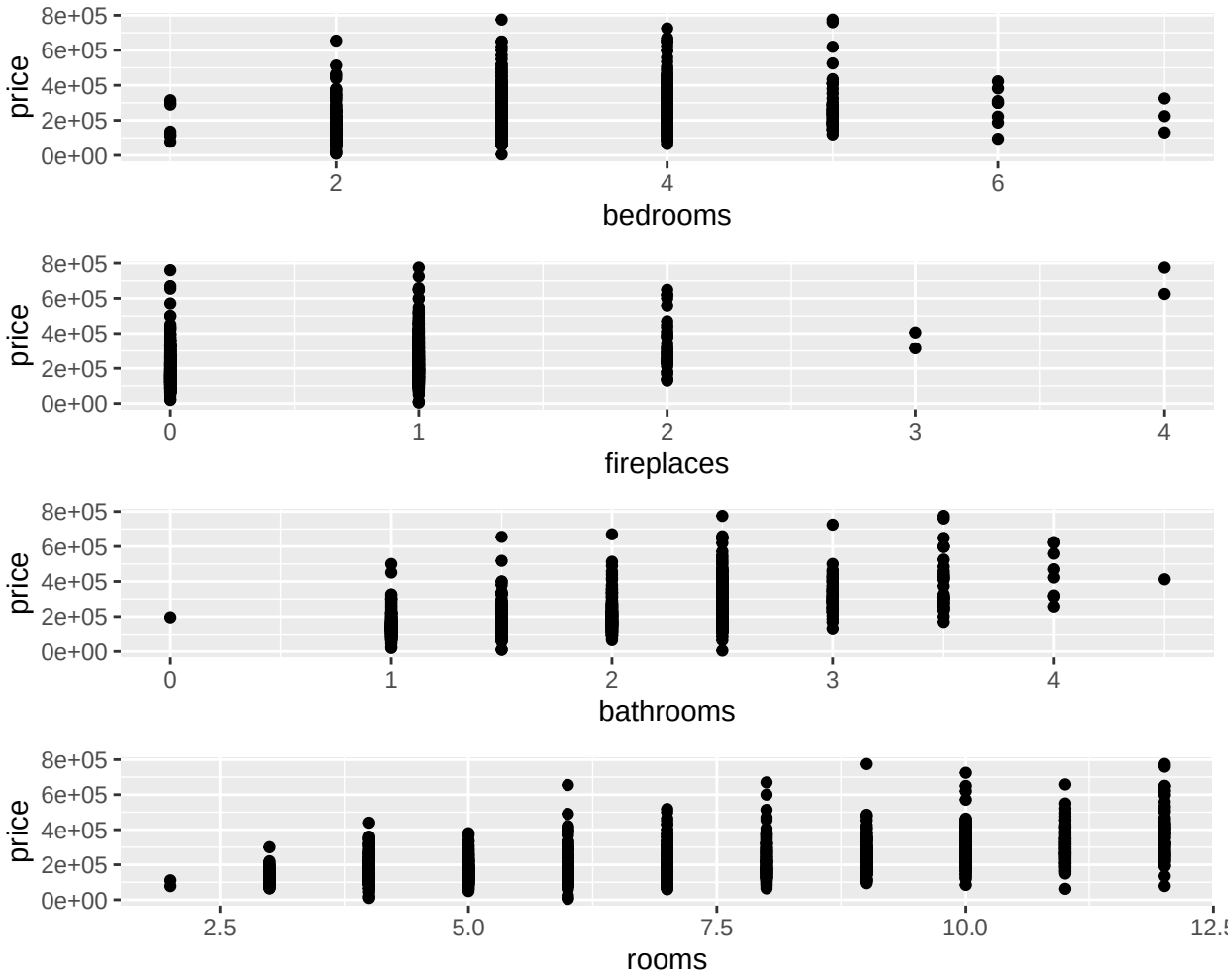
C. Report

The linear model 6 gives the best prediction, showing the lowest rmse among all the models including linear and knn models tried above. The model 6 which I would like to suggest reflects all the available variables except personnel and volatile information(percentage of college graduates). The "land value variables" which has a stark correlation with the house price is used as a quadratic form from the relationship between two variables. In addition, the variables such as rooms, livingArea, etc. are applied as the interaction form with the "age" variable because mixed effects of each variable on the price need to be considered separately. As a result, we can get much lower errors from the model 6 than those of baseline models.

D. Apendix(Analysis of data)

Plots of data





Mean price of factors

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 7 x 2
##   bedrooms mean_price
##      <int>      <dbl>
## 1        1    192771.
## 2        2    152561.
## 3        3    200678.
## 4        4    265551.
## 5        5    276578.
## 6        6    277329.
## 7        7    226667.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 9 x 2
##   bathrooms mean_price
##       <dbl>      <dbl>
## 1       0      195900
## 2       1      147999.
## 3       1.5    168029.
```

```
## 4       2      201506.
## 5       2.5    268345.
## 6       3      320979.
## 7       3.5    383113.
## 8       4      448450.
## 9       4.5    412500
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 2
##    rooms mean_price
##    <int>      <dbl>
##  1     2     94500
##  2     3    134156.
##  3     4    168918.
##  4     5    167298.
##  5     6    185314.
##  6     7    191829.
##  7     8    220597.
##  8     9    245279.
##  9    10    288567.
## 10    11    305914.
## 11    12    373219.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   fireplaces mean_price
##        <int>      <dbl>
## 1          0    174653.
## 2          1    235163.
## 3          2    318821.
## 4          3    360500
## 5          4    700000
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   heating         mean_price
##   <fct>                <dbl>
## 1 hot air            226355.
## 2 hot water/steam    209132.
## 3 electric           161889.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   fuel     mean_price
##   <fct>         <dbl>
## 1 gas         228535.
## 2 electric    164938.
## 3 oil         188734.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   sewer             mean_price
##   <fct>                  <dbl>
## 1 septic               200284.
```

```
## 2 public/commercial    216426.
## 3 none                 250952.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   waterfront mean_price
##   <fct>           <dbl>
## 1 Yes           373992.
## 2 No            210548.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   newConstruction mean_price
##   <fct>                <dbl>
## 1 Yes                282307.
## 2 No                 208507.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   centralAir mean_price
##   <fct>           <dbl>
## 1 Yes           254904.
## 2 No            187022.
```

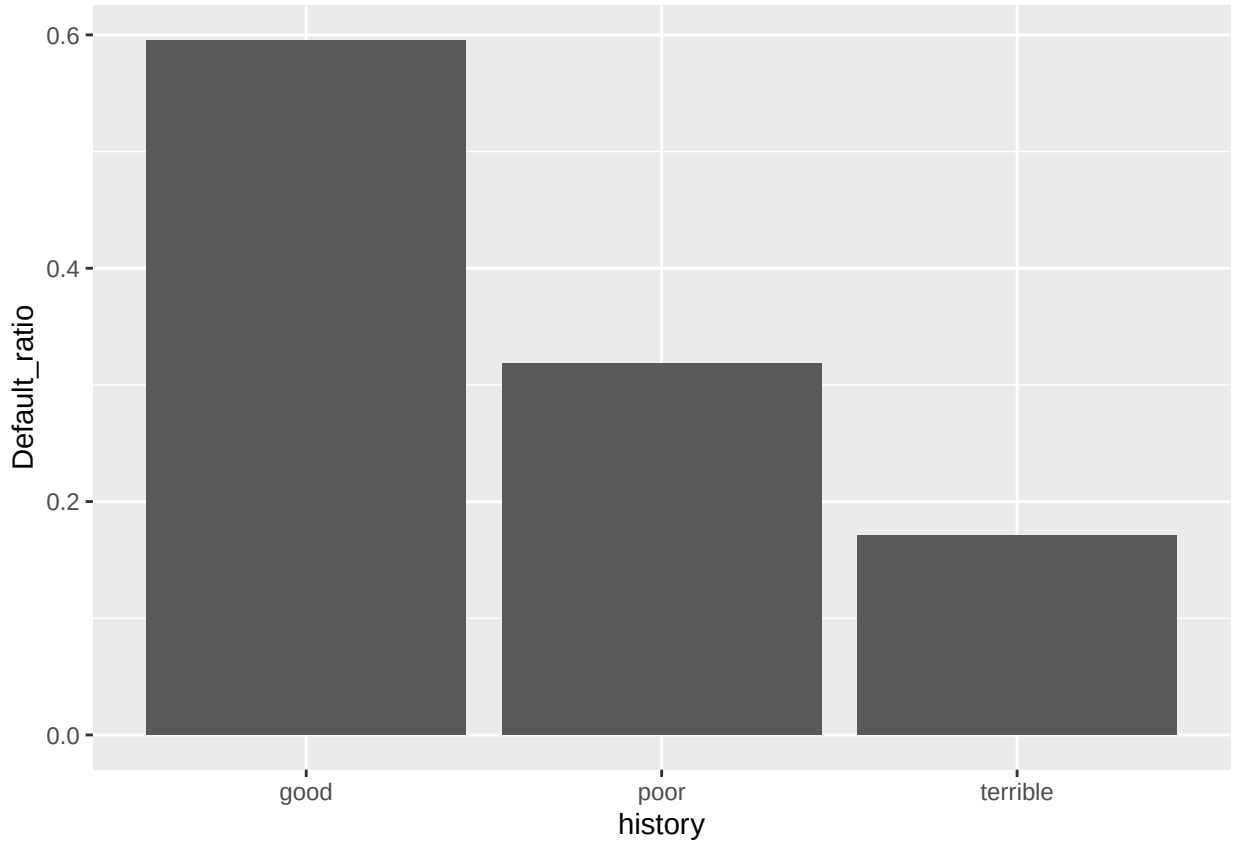
Summary of Lm6

```
##
## Call:
## lm(formula = price ~ (. - pctCollege + poly(landValue, 2)) +
##     (age * (poly(landValue, 2) + livingArea + bedrooms + bathrooms +
##         newConstruction)) + (waterfront * (livingArea + bedrooms)),
##     data = saratoga_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -234936  -34548   -3254   28457  434839
##
## Coefficients: (1 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.790e+05  5.326e+04   3.360 0.000801 ***
## lotSize                  6.633e+03  2.446e+03   2.712 0.006777 **
## age                      1.697e+03  3.680e+03   0.461 0.644756
## landValue                7.293e-01  6.837e-02  10.666  < 2e-16 ***
## livingArea               4.367e+01  3.019e+01   1.447 0.148195
## bedrooms                -2.668e+04  2.414e+04  -1.105 0.269179
## fireplaces               2.617e+03  3.320e+03   0.788 0.430758
## bathrooms                2.269e+04  4.836e+03   4.693 2.97e-06 ***
## rooms                    3.040e+03  1.078e+03   2.821 0.004862 **
## heatinghot water/steam  -1.224e+04  4.726e+03  -2.591 0.009681 **
## heatingelectric         -2.003e+03  1.298e+04  -0.154 0.877402
## fuelelectric            -8.342e+03  1.265e+04  -0.659 0.509741
## fueloil                 -1.773e+03  5.479e+03  -0.324 0.746232
## sewerpublic/commercial  -4.286e+03  4.119e+03  -1.041 0.298292
## sewernone               -3.274e+04  1.969e+04  -1.663 0.096583 .
```

```
## waterfrontNo            -2.020e+05  5.241e+04  -3.855 0.000121 ***
## newConstructionNo        4.670e+04  8.487e+03   5.503 4.46e-08 ***
## centralAirNo            -8.672e+03  3.900e+03  -2.224 0.026346 *
## poly(landValue, 2)1             NA         NA      NA       NA
## poly(landValue, 2)2      1.136e+05  7.711e+04   1.473 0.141093
## age:poly(landValue, 2)1  5.194e+03  1.760e+03   2.952 0.003214 **
## age:poly(landValue, 2)2 -1.021e+04  2.062e+03  -4.951 8.32e-07 ***
## age:livingArea          -5.112e-01  1.330e-01  -3.843 0.000127 ***
## age:bedrooms             1.332e+02  6.963e+01   1.913 0.055930 .
## age:bathrooms            6.220e+01  1.115e+02   0.558 0.576994
## age:newConstructionNo   -1.445e+03  3.678e+03  -0.393 0.694351
## livingArea:waterfrontNo  4.421e+01  2.964e+01   1.492 0.136058
## bedrooms:waterfrontNo    1.250e+04  2.415e+04   0.518 0.604782
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58050 on 1356 degrees of freedom
## Multiple R-squared:  0.6687, Adjusted R-squared:  0.6624
## F-statistic: 105.3 on 26 and 1356 DF,  p-value: < 2.2e-16
```

Problem 3: Classification and retrospective sampling

A. Bar plot



B. Logistic regression model

Split data sets(test, train) and Fit the logistic model

LM : duration + amount + installment + age + history + purpose + foreign

```
##         (Intercept)            duration              amount         installment
##               -1.14                0.03                0.00                0.24
##                 age         historypoor     historyterrible          purposeedu
##               -0.02               -1.10               -1.80                0.94
## purposegoods/repair       purposenewcar      purposeusedcar       foreigngerman
##                0.27                1.10               -0.47               -1.34
```

Predict of in-sample

```
##    yhat
## y     0   1
##   0 551  20
##   1 199  31
```

Predict of out-of-sample

```
##    yhat
## y     0   1
##   0 124   5
##   1  64   6
```

Predict accuracy

```
## [1] 0.7265918
```

```
## [1] 0.6532663
```

Sampling ratio

```
##
##   0   1
## 0.7 0.3
```

Report

Because of oversampling of defaults(30%), default probability of 'good' credit history is higher than that of 'terrible' history. And in the model, default probability of 'terrible' history is less than that of 'poor' history. Lastly, FDR is too high(in-sample 38.3%, out-of-sample 44.4%), TPR is low(in-sample 15.4%, out-of-sample 16.9%) and accuracy is only around 70%. We cannot screen prospective borrowers to classify them into "high" versus "low" probability of default exactly.
So, this data set is not appropriate for building a predictive model of defaults. I would recommend switching to 'prospective sampling'.

Problem 4: Children and hotel reservations

A. Model building

Data split and Fit models

baseline model1 lm1 = lm(children ~ market_segment + adults + customer_type + is_repeated_guest)

```
##          yhat
## children    0
##        0 8300
##        1  699
```

baseline model2 lm2 = lm(children ~ .-arrival_date)

```
##          yhat
## children    0    1
##        0 8188  112
##        1  444  255
```

additionally developed models lm3 = lm(children ~ .-arrival_date + (stays_in_weekend_nights*adults)* + *(lead_time*hotel))

```
##          yhat
## children    0    1
##        0 8189  111
##        1  441  258
```

lm4 = lm(children ~ .-arrival_date + (stays_in_weekend_nights*lead_time*hotel))

```
##          yhat
## children    0    1
##        0 8190  110
##        1  444  255
```

lm5 = lm(children ~ .-arrival_date + poly(adults, 2) + (stays_in_weekend_nights*poly(adults, 2)) + *(stays_in_weekend_nights*lead_time) + hotel*stays_in_weekend_nights)

```
##          yhat
## children    0    1
##        0 8191  109
##        1  438  261
```

lm6 = lm(children ~ .-arrival_date + poly(adults, 2) + (stays_in_weekend_nights*poly(adults, 2)) + *(stays_in_weekend_nights*lead_time) + hotel*stays_in_weekend_nights + stays_in_weekend_nights*reserved_room_type)

```
##          yhat
## children    0    1
##        0 8185  115
##        1  443  256
```

stepwise selection with AIC lm7 = lm(children ~ hotel + lead_time + stays_in_weekend_nights + stays_in_week_nights + meal + market_segment + distribution_channel + is_repeated_guest + previous_cancellations + previous_bookings_not_canceled + reserved_room_type + assigned_room_type + booking_changes + deposit_type + days_in_waiting_list + customer_type + average_daily_rate + required_car_parking_spaces + total_of_special_requests + poly(adults, 2) + stays_in_weekend_nights:poly(adults, 2) + lead_time:stays_in_weekend_nights + hotel:stays_in_weekend_nights + hotel:reserved_room_type + reserved_room_type:poly(adults, 2) + market_segment:reserved_room_type)

```
##          yhat
## children    0    1
##        0 8219   81
##        1  417  282
```

comparison of models' accuracy

```
## [1] 0.9223247
```

```
## [1] 0.9382154
```

```
## [1] 0.9386599
```

```
## [1] 0.9384376
```

```
## [1] 0.9392155
```

```
## [1] 0.9379931
```

```
## [1] 0.9446605
```

We can see the model 5 has the highest accuracy among the model 1 to 6. If we apply a stepwise selection process based on the model 5, the model 7 can be derived. The accuracy of the model 7 is the highest.

Averaging the performance of the three models over 10 train/test splits

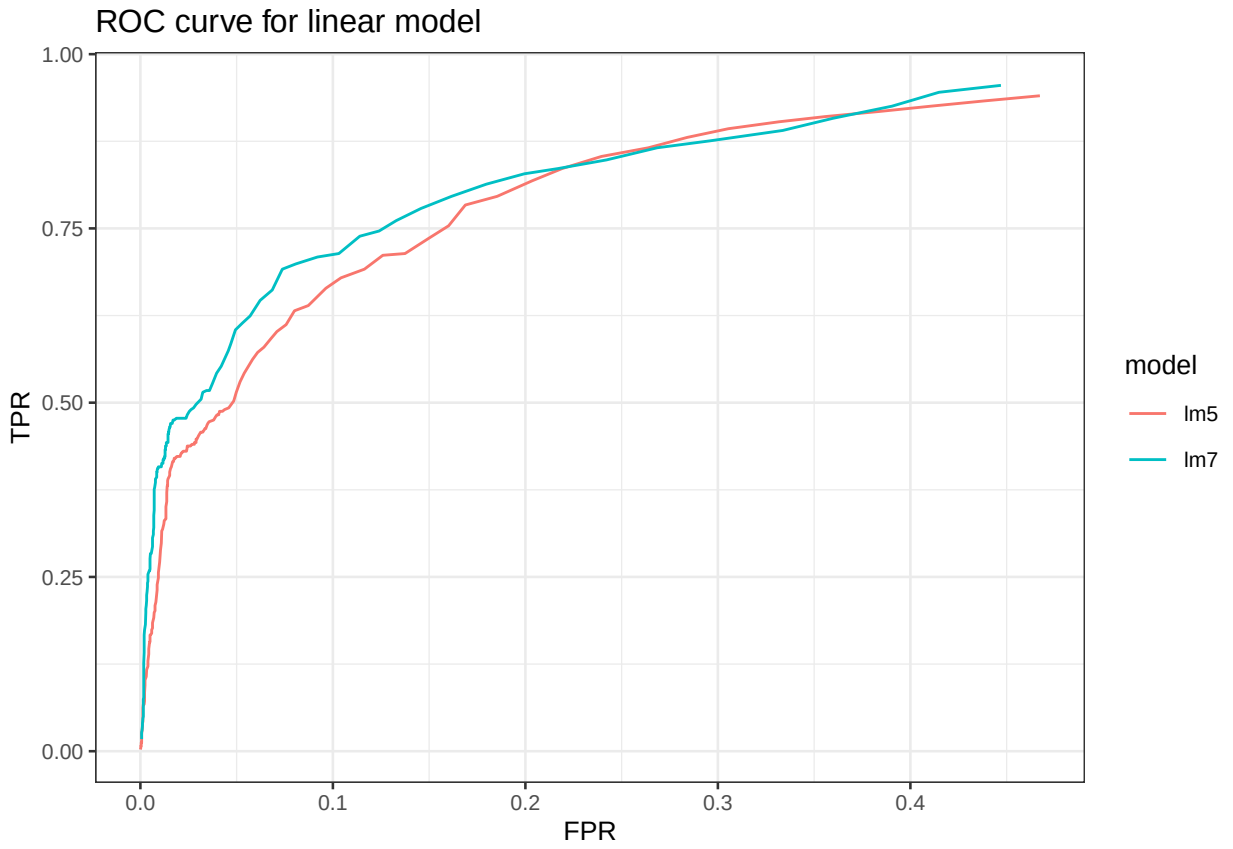
model2(baseline model), model5(second baseline model), model7(final suggestion)

```
## [1] 0.9365485 0.9372152 0.9437715
```

```
##    result
## 0.9431492
```

linear model7 gives the highest predict accuracy, around 94% which has improvement over the baseline model.

B. Model validation step1: ROC curve for linear model 5,7



LM7 is alittle bit better than LM5

C. Model validation: step2

```
## # A tibble: 20 x 2
##    fold_id mean_children
##      <int>         <dbl>
##  1       1         0.068
##  2       2         0.084
##  3       3         0.068
##  4       4         0.08
##  5       5         0.092
##  6       6         0.048
##  7       7         0.076
##  8       8         0.084
##  9       9         0.072
## 10      10         0.104
## 11      11         0.096
## 12      12         0.136
## 13      13         0.092
## 14      14         0.076
## 15      15         0.068
## 16      16         0.08
## 17      17         0.084
## 18      18         0.068
```

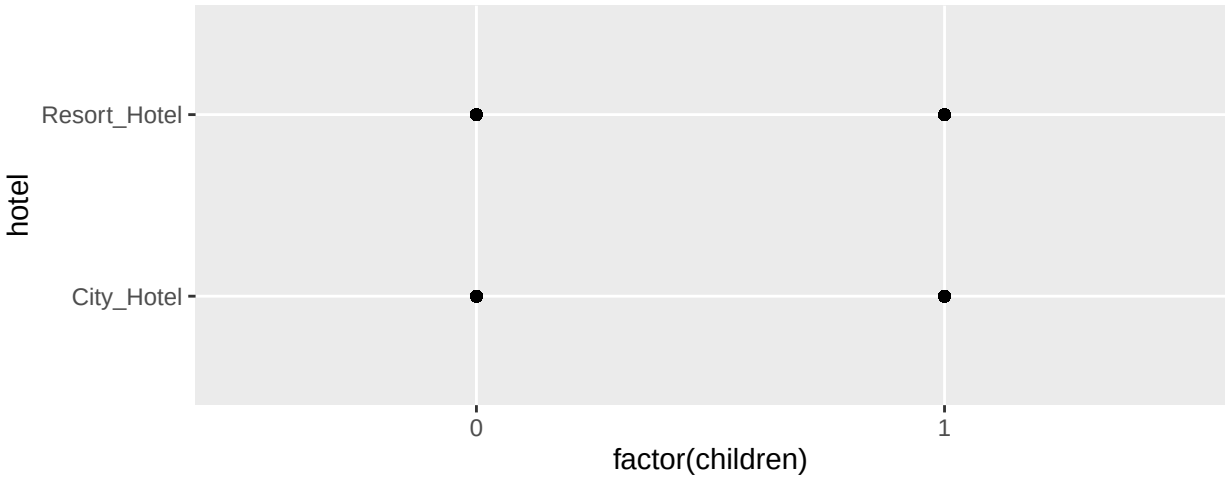
```
## 19      19        0.076
## 20      20        0.0562
```

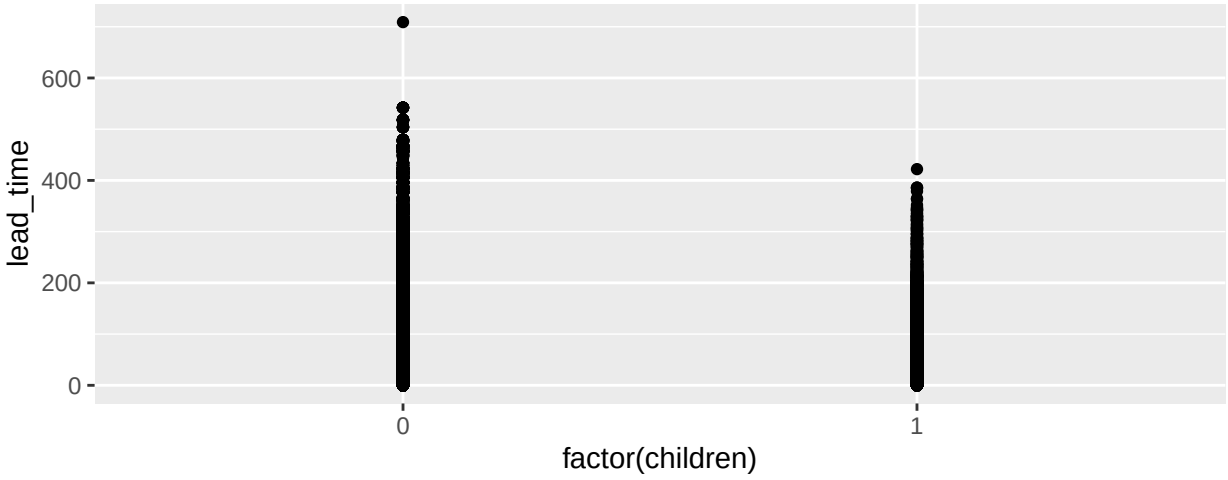
Sample probablity of fold_id is variously scattered(0.048~0.136)

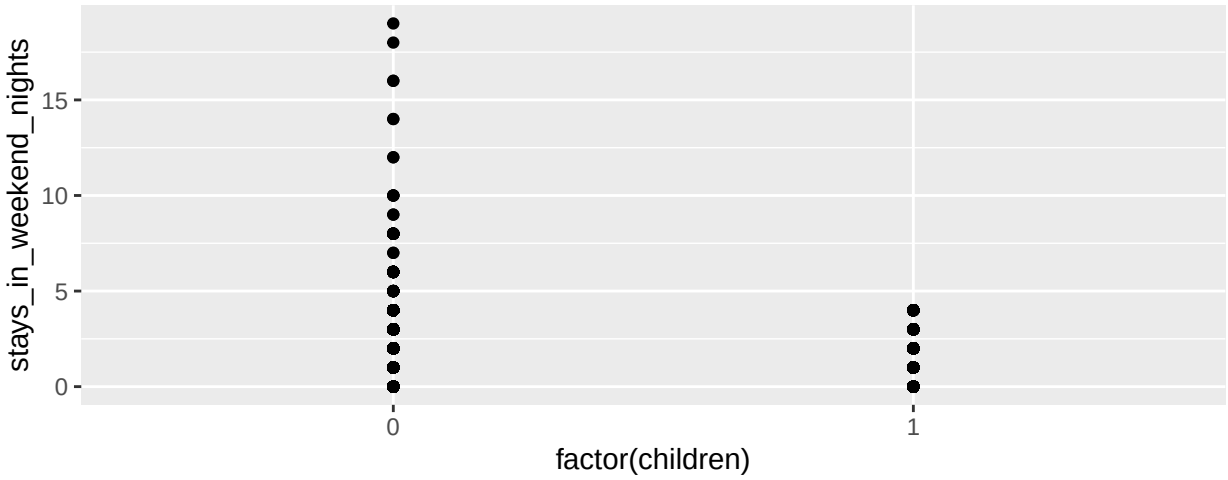
```
##             .
## 1  0.04185995
## 2  0.04185995
## 3  0.04185995
## 4  0.04185995
## 5  0.04185995
## 6  0.04185995
## 7  0.04185995
## 8  0.04185995
## 9  0.04185995
## 10 0.04185995
## 11 0.04185995
## 12 0.04185995
## 13 0.04185995
## 14 0.04185995
## 15 0.04185995
## 16 0.04185995
## 17 0.04185995
## 18 0.04185995
## 19 0.04185995
## 20 0.04185995
```

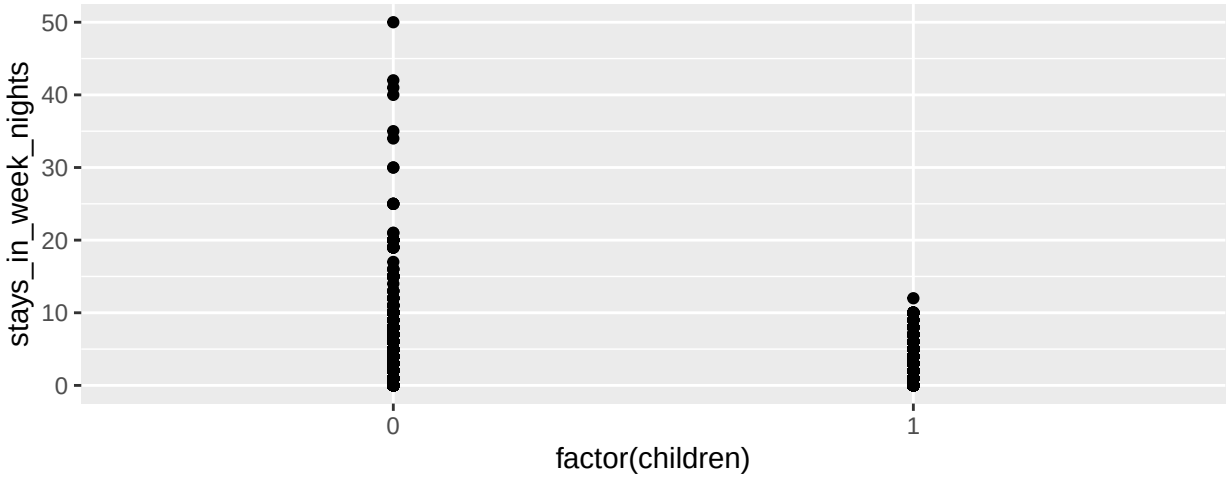
Predicted probablity of fold_id is around 0.0412. This is less than actual probablity.

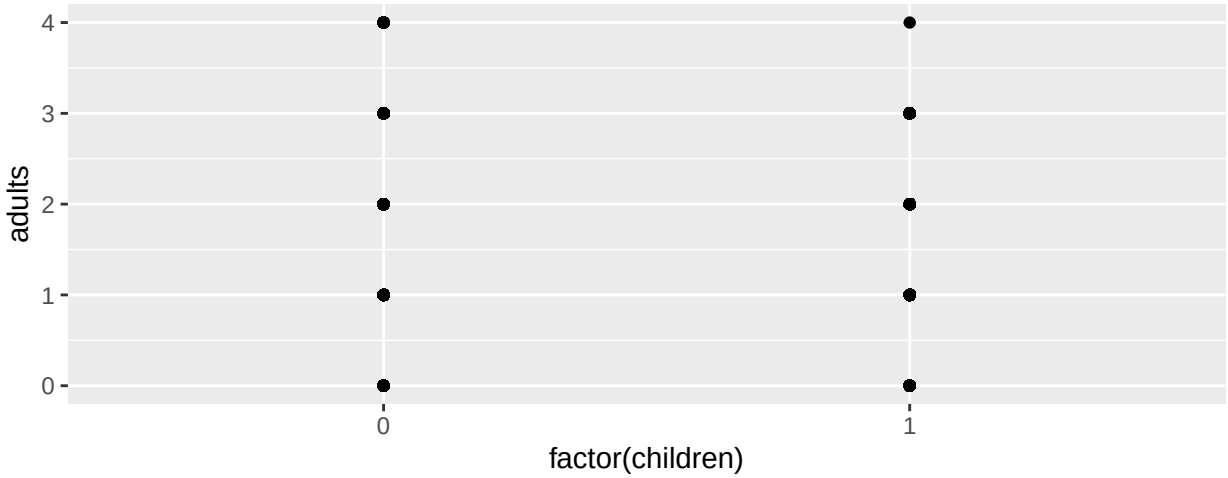
D. Appendix: analysing data

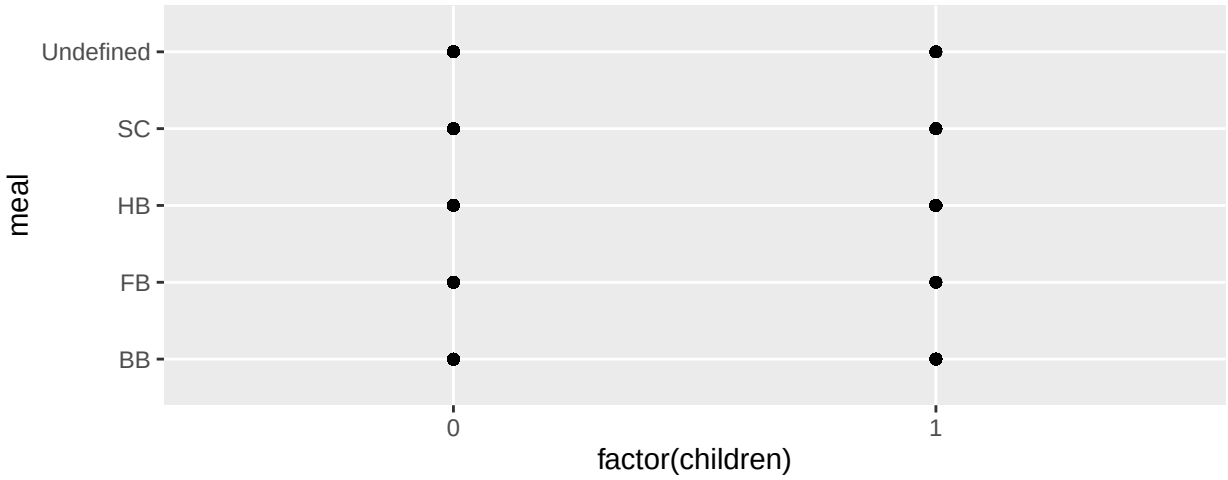


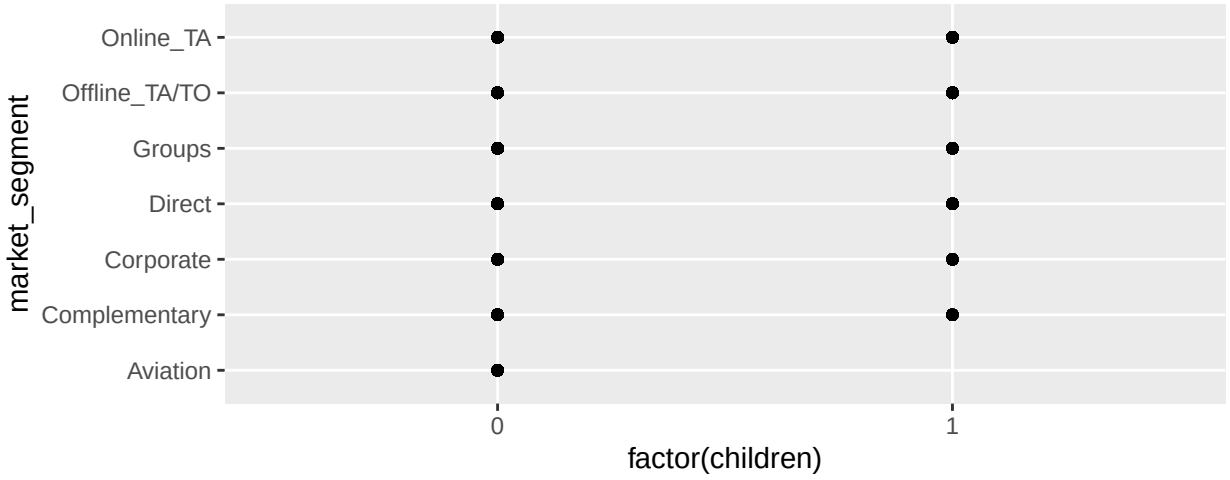


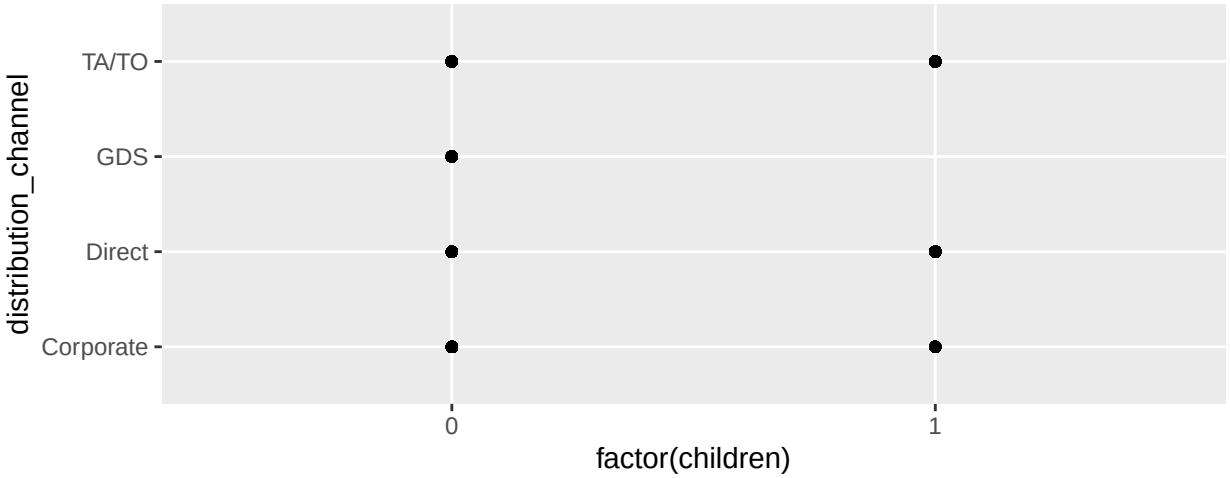


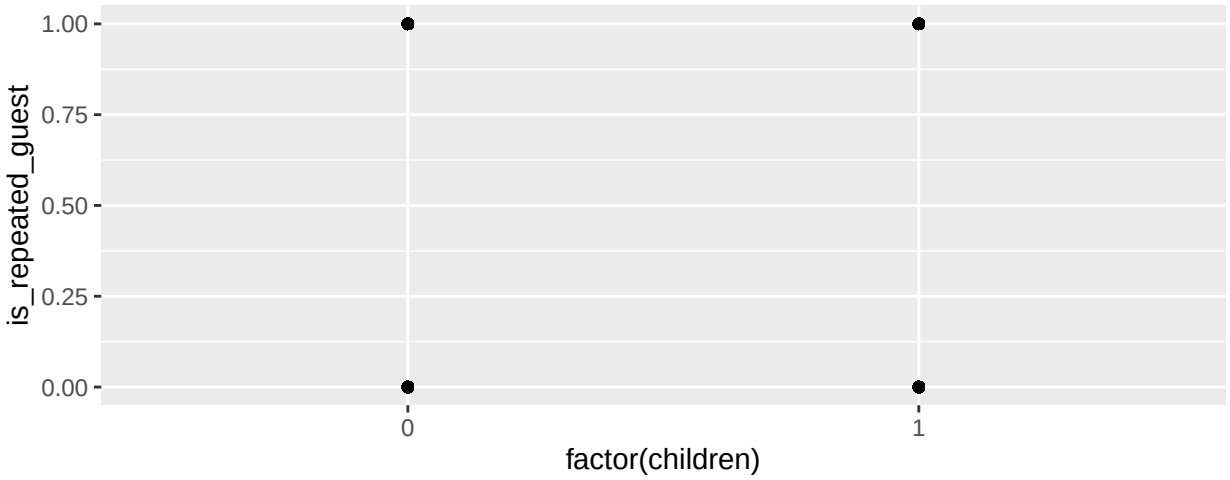

stays_in_week_nights
50
40
30
20
10
0
0
1
factor(children)

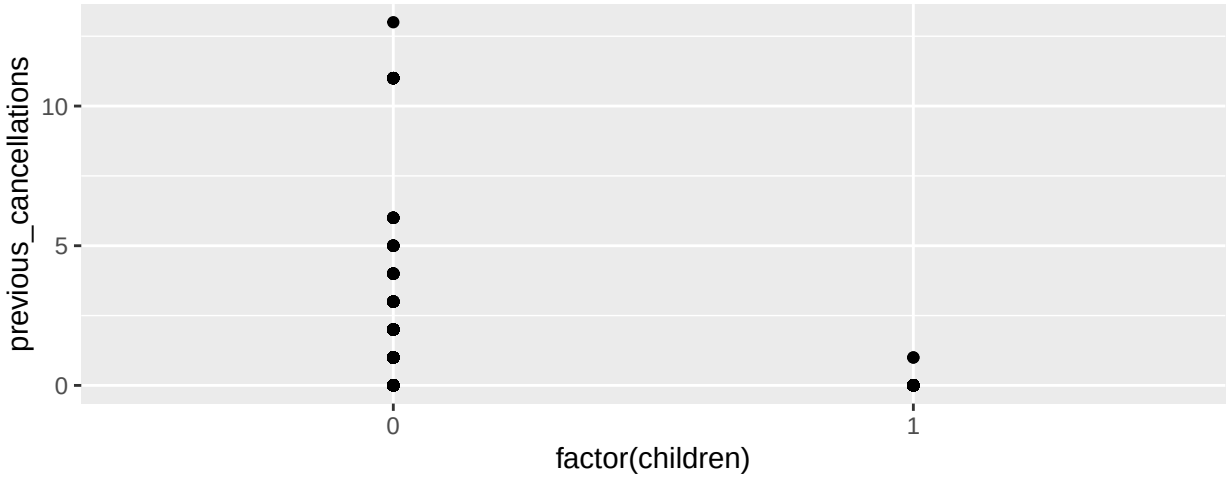

adults
4
3
2
1
0
0
1
factor(children)

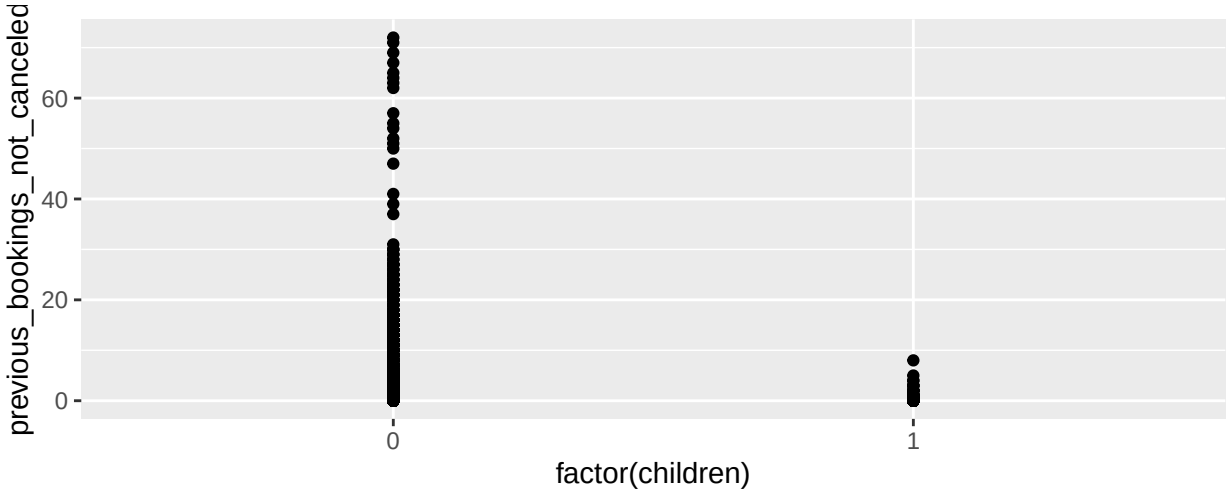

meal
Undefined
SC
HB
FB
BB
0
1
factor(children)

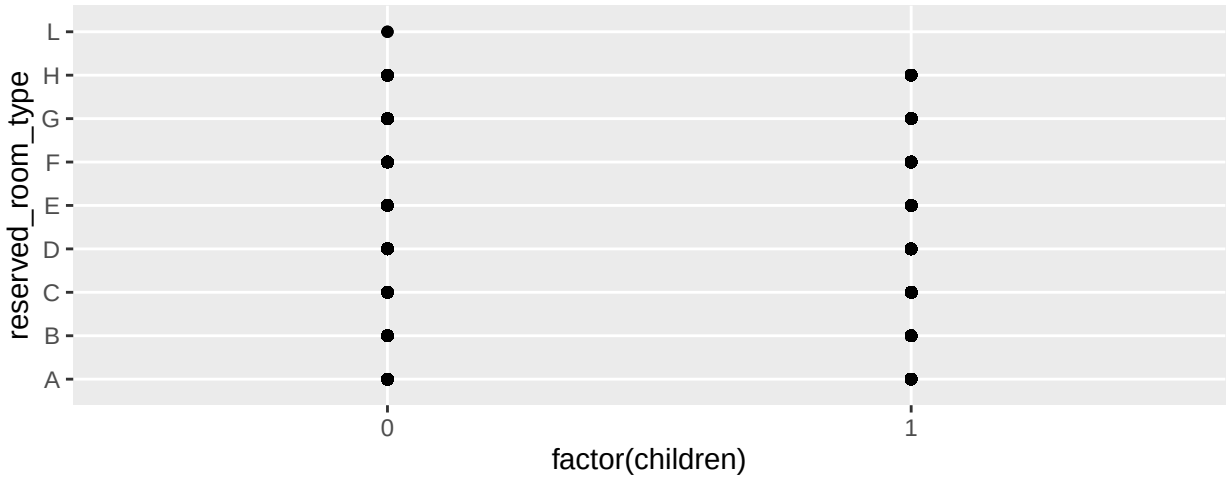

market_segment
Online_TA
Offline_TA/TO
Groups
Direct
Corporate
Complementary
Aviation
0
1
factor(children)

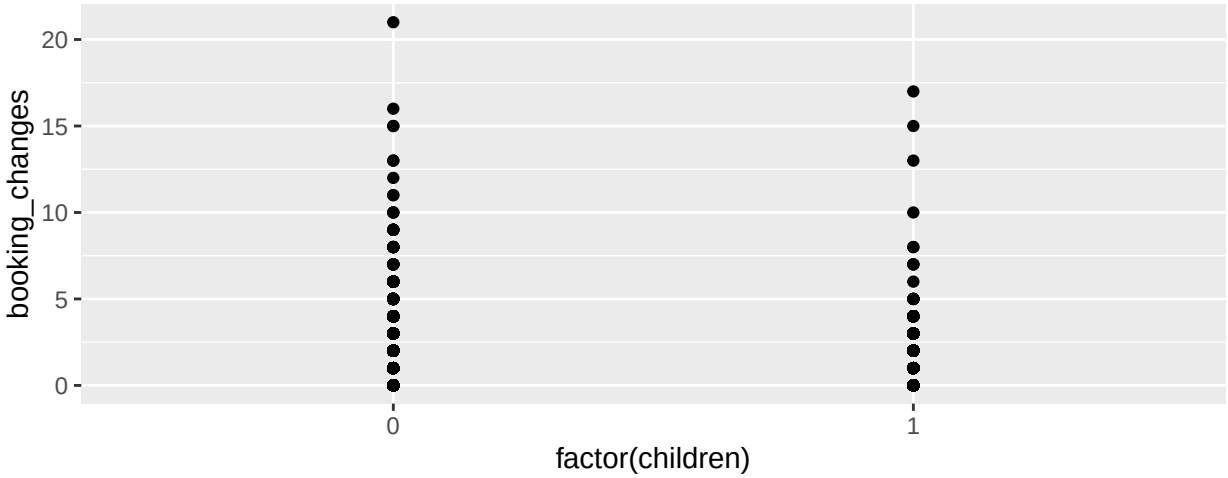

distribution_channel
TA/TO
GDS
Direct
Corporate
0
1
factor(children)

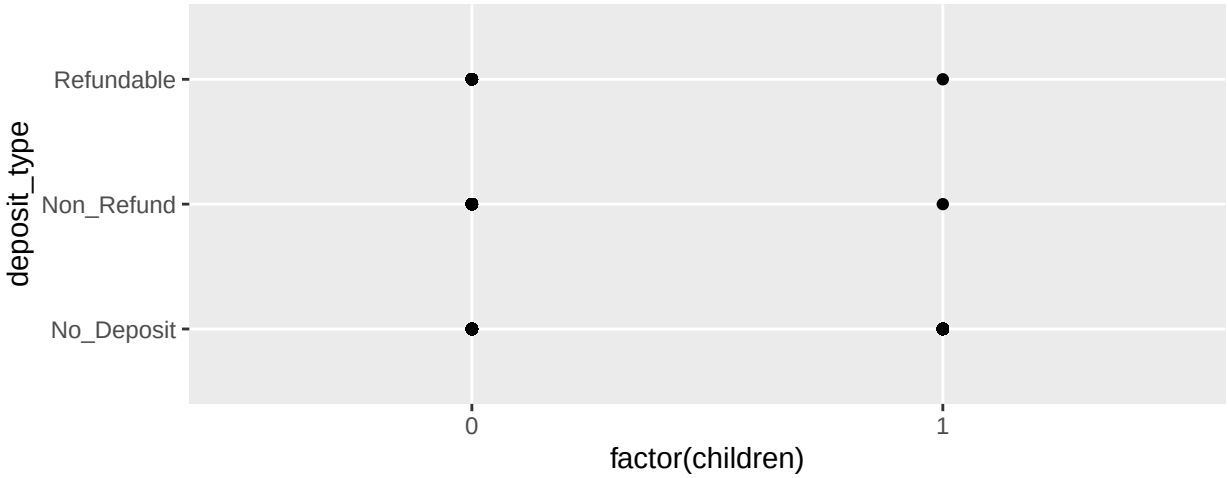

is_repeated_guest
1.00
0.75
0.50
0.25
0.00
0
1
factor(children)

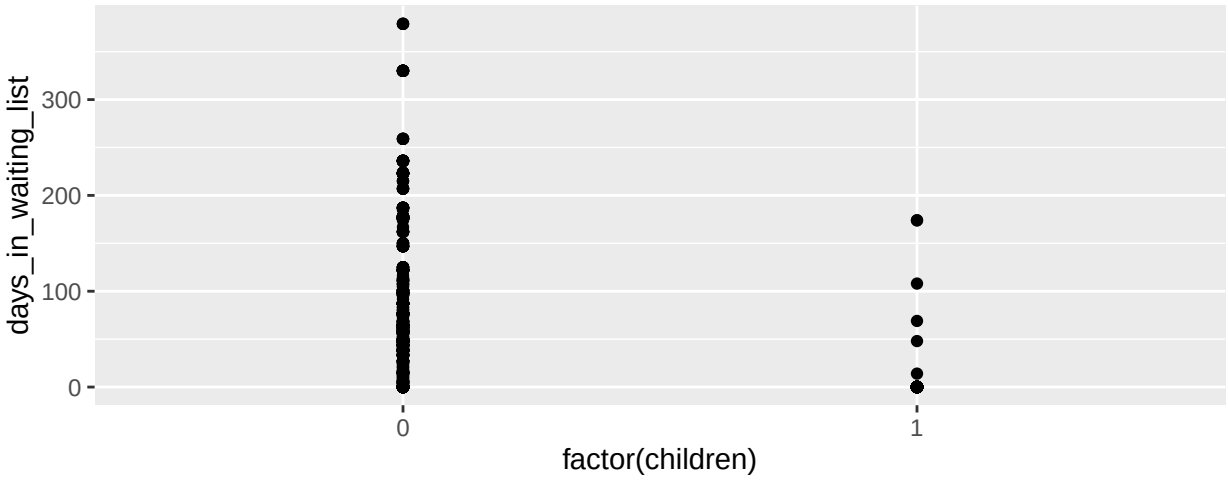

previous_cancellations
10
5
0
0
1
factor(children)

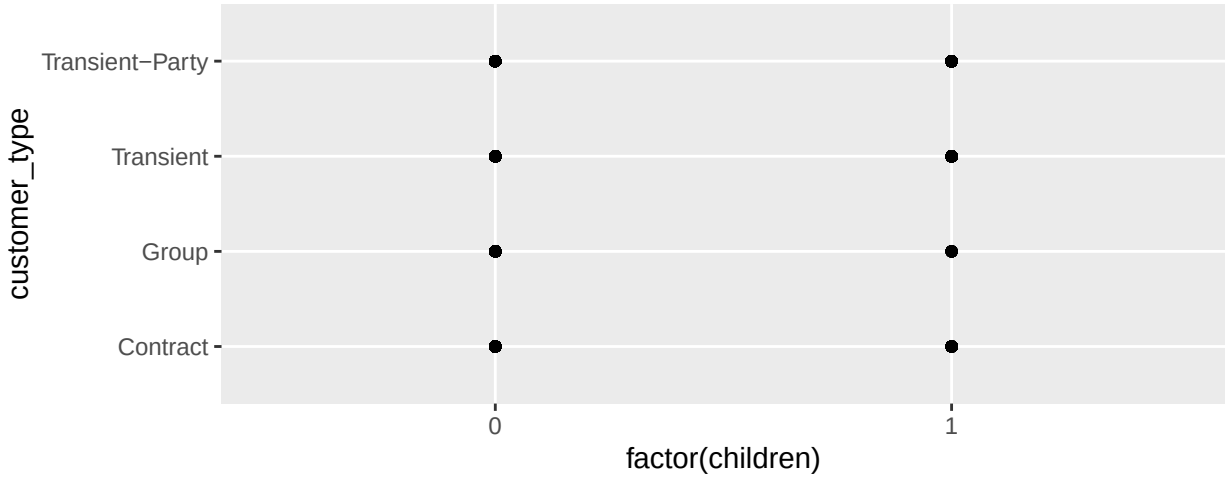

previous_bookings_not_canceled
60
40
20
0
0
1
factor(children)

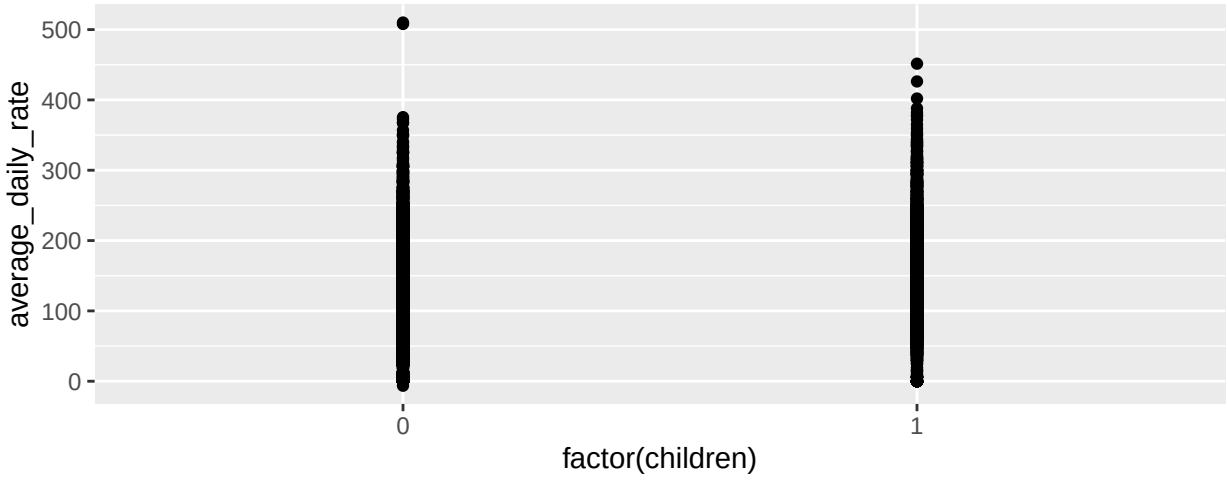

reserved_room_type
L
H
G
F
E
D
C
B
A
0
1
factor(children)

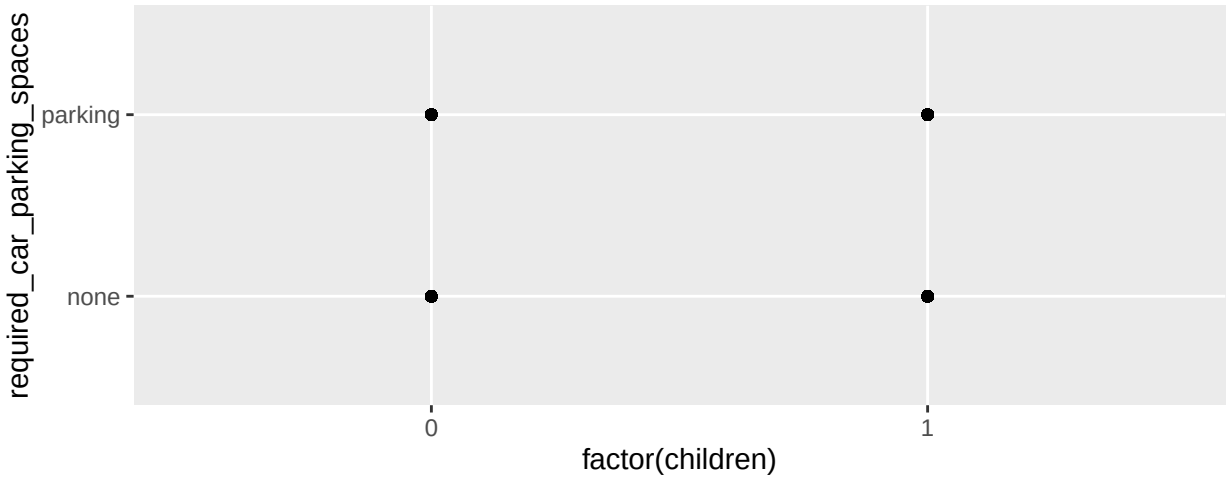

booking_changes
20
15
10
5
0
0
1
factor(children)

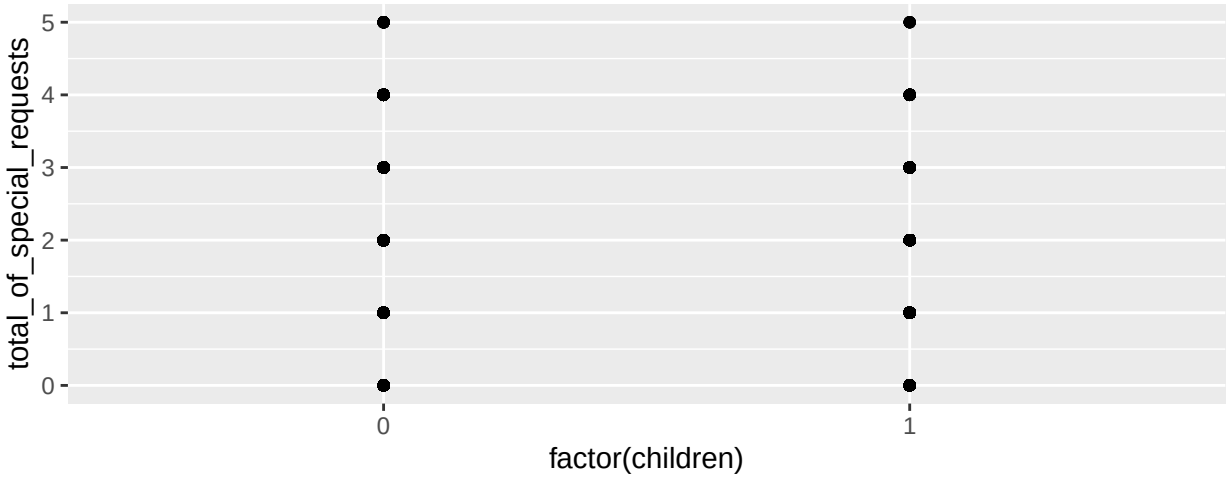

deposit_type
Refundable
Non_Refund
No_Deposit
0
1
factor(children)


days_in_waiting_list
300
200
100
0
0
1
factor(children)


Transient−Party
Transient
Group
Contract
customer_type
0
1
factor(children)


500
400
300
200
100
0
average_daily_rate
0
1
factor(children)


parking
none
required_car_parking_spaces
0
1
factor(children)




```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   children mean_leadtime mean_stayinweek mean_dailyrates mean_requests
##      <int>         <dbl>           <dbl>           <dbl>         <dbl>
## 1        0          80.5            2.45            95.6         0.677
## 2        1          76.0            2.61           148.          1.10
```