

**Postgraduate Certificate in Software Design with Artificial Intelligence**

**Data Mining and Machine Learning**

**Assignment 1**

Student ID: A00267948

Student Name: Daniel Foth

*Brief Description: This paper evaluates different data sets using regression, decision trees and kNN algorithms. Models are created and predictions are made using the test data from the data sets.*

Git: <https://github.com/DanielsHappyWorks/DM-ML-Module-Assignment>

Contents

[0. Introduction 3](#_Toc27606312)

[1. Regression 3](#_Toc27606313)

[1.1. Overview of the Problem 3](#_Toc27606314)

[1.2. Data Exploration (tables and graphs) 4](#_Toc27606315)

[1.3. Definition of Training and Testing Set 7](#_Toc27606316)

[1.4. Model Generation and Information 8](#_Toc27606317)

[1.5. Predictions for the test data 9](#_Toc27606318)

[1.6. Evaluation of the model(s) and conclusion. 10](#_Toc27606319)

[2. Decision Trees 11](#_Toc27606320)

[2.1. Overview of the Problem 11](#_Toc27606321)

[2.2. Data Exploration (tables and graphs) 12](#_Toc27606322)

[2.3. Definition of Training and Testing Set 15](#_Toc27606323)

[2.4. Model Generation and Information 16](#_Toc27606324)

[2.5. Predictions for the test data 18](#_Toc27606325)

[2.6. Evaluation of the model(s) and conclusion. 19](#_Toc27606326)

[3. kNN 20](#_Toc27606327)

[3.1. Overview of the Problem 20](#_Toc27606328)

[3.2. Data Exploration (tables and graphs) 21](#_Toc27606329)

[3.3. Definition of Training and Testing Set 23](#_Toc27606330)

[3.4. Model Generation and Information 24](#_Toc27606331)

[3.5. Predictions for the test data 25](#_Toc27606332)

[3.6. Evaluation of the model(s) and conclusion. 29](#_Toc27606333)

[4. Citations 30](#_Toc27606334)

# Introduction

# Regression

## Overview of the Problem

For regression, a data set that describes characteristics of wine will be used. The data is called Wine Quality and was originally sourced by Paulo Cortez, Antonio Cerdeira, Fernando Almeida, Telmo Matos and Jose Reis.

UCI Repository: <https://archive.ics.uci.edu/ml/datasets/wine+quality>

The data contains two sets:

1. red wine with 1599 rows
2. white wine with 4898 rows

The data set has 12 features which include:

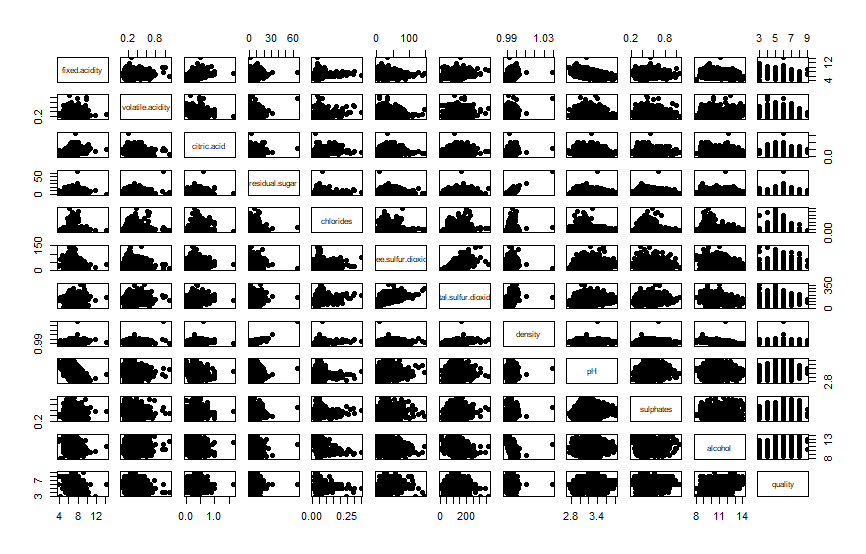
1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. quality (score between 0 and 10)

For this specific problem only the white wine data set will be used. Using regression algorithms, all the features will be analysed and correlated together to try and predict the quality of the wine.

## Data Exploration (tables and graphs)

The data has no missing features which was double checked when the csv was loaded into R. No clean up was needed to handle missing fields

The first graph created was one with all columns plotted against each other. Its hard to see the output because of how small the graphs are.



To get a better image of how all the attributes effect quality, they were plotted against each other and exported as pdf files using R.

Plots:

|  |  |  |
| --- | --- | --- |
| Feature | Scatter | Histogram |
| fixed acidity |  |  |
| volatile acidity |  |  |
| citric acid |  |  |
| residual sugar |  |  |
| chlorides |  |  |
| free sulfur dioxide |  |  |
| total sulfur dioxide |  |  |
| density |  |  |
| pH |  |  |
| sulphates |  |  |
| alcohol |  |  |

Larger versions of the plots can be seen within the R project. Please note that for some graphs the Linear Regression line doesn’t do a good job at predicting very high and very low values with the data present which is probably due to linear regression not being the best fit for the data. This could also be because there are more mid ranged values which might be degrading the performance.

The quality was also used against all features separately to get a linear regression models so they could be compared. The performance is described in section 1.4 Model Generation and Information

## Definition of Training and Testing Set

The entire data set with 12 features and 4898 was used for Training models.

14 rows were chosen randomly by hand so they can be used to validate how well the models would predict quality. The validation sample has two of each type of quality to be tested against.

Validation Sample:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | quality | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol | fixed acidity |
| 1 | 3 | 0.26 | 0.21 | 16.2 | 0.074 | 41 | 197 | 0.998 | 3.02 | 0.5 | 9.8 | 8.5 |
| 2 | 3 | 0.17 | 0.47 | 1.4 | 0.037 | 5 | 33 | 0.9939 | 2.89 | 0.28 | 9.6 | 10.3 |
| 3 | 4 | 0.485 | 0 | 1.5 | 0.065 | 8 | 103 | 0.994 | 3.63 | 0.4 | 9.7 | 5.5 |
| 4 | 4 | 0.31 | 0.31 | 9.9 | 0.04 | 10 | 175 | 0.9953 | 3.46 | 0.55 | 11.4 | 6.7 |
| 5 | 5 | 0.36 | 0.04 | 5.7 | 0.046 | 21 | 87 | 0.9934 | 3.22 | 0.51 | 10.2 | 5.9 |
| 6 | 5 | 0.37 | 0.51 | 11.8 | 0.044 | 62 | 163 | 0.9976 | 3.19 | 0.44 | 8.8 | 6.8 |
| 7 | 6 | 0.34 | 0.39 | 7.6 | 0.04 | 45 | 215 | 0.9965 | 3.11 | 0.53 | 9.2 | 7.6 |
| 8 | 6 | 0.3 | 0.27 | 11.6 | 0.028 | 22 | 97 | 0.99314 | 2.96 | 0.38 | 11.7 | 6.8 |
| 9 | 7 | 0.23 | 0.39 | 2.3 | 0.033 | 29 | 102 | 0.9908 | 3.26 | 0.54 | 12.3 | 7.2 |
| 10 | 7 | 0.41 | 0.37 | 4.5 | 0.03 | 40 | 114 | 0.992 | 3.17 | 0.54 | 12.4 | 7.9 |
| 11 | 8 | 0.19 | 0.27 | 13.9 | 0.057 | 45 | 155 | 0.99807 | 2.94 | 0.41 | 8.8 | 7.3 |
| 12 | 8 | 0.28 | 0.34 | 2.2 | 0.037 | 24 | 125 | 0.98986 | 3.36 | 0.33 | 12.8 | 5.8 |
| 13 | 9 | 0.27 | 0.45 | 10.6 | 0.035 | 28 | 124 | 0.997 | 3.2 | 0.46 | 10.4 | 9.1 |
| 14 | 9 | 0.36 | 0.34 | 4.2 | 0.018 | 57 | 119 | 0.9898 | 3.28 | 0.36 | 12.7 | 6.9 |

## Model Generation and Information

There were 14 models created. 11 with only one feature each. 3 with all the features of which two were polynomial regression models.

Single Feature models:

|  |  |
| --- | --- |
| Model | R-squared |
| Model 1: quality ~ fixed.acidity | Multiple R-squared: 0.01292  Adjusted R-squared: 0.01272 |
| Model 2: quality ~ volatile.acidity | Multiple R-squared: 0.03792  Adjusted R-squared: 0.03772 |
| Model 3: quality ~ citric.acid | Multiple R-squared: 8.481e-05  Adjusted R-squared: -0.0001194 |
| Model 4: quality ~ residual.sugar | Multiple R-squared: 0.009521  Adjusted R-squared: 0.009319 |
| Model 5: quality ~ chlorides | Multiple R-squared: 0.04407  Adjusted R-squared: 0.04388 |
| Model 6: quality ~ free.sulfur.dioxide | Multiple R-squared: 6.655e-05  Adjusted R-squared: -0.0001377 |
| Model 7: quality ~ total.sulfur.dioxide | Multiple R-squared: 0.03053  Adjusted R-squared: 0.03034 |
| Model 8: quality ~ density | Multiple R-squared: 0.09432  Adjusted R-squared: 0.09414 |
| Model 9: quality ~ pH | Multiple R-squared: 0.009886  R-squared: 0.009684 |
| Model 10: quality ~ sulphates | Multiple R-squared: 0.002881  Adjusted R-squared: 0.002678 |
| Model 11: quality ~ alcohol | Multiple R-squared: 0.1897  Adjusted R-squared: 0.1896 |

Multiple Feature Models:

|  |  |
| --- | --- |
| Model | R-squared |
| All Features | Multiple R-squared: 0.2819  Adjusted R-squared: 0.2803 |
| All Features to degree 2 | Multiple R-squared: 0.3679  Adjusted R-squared: 0.3578 |
| All Features to degree 3 | Multiple R-squared: 0.4612  Adjusted R-squared: 0.4181 |

A few other models with just a select few parameters were tried but they were usually worse than the three listed above.

## Predictions for the test data

For predictions I chose to do them on the models with all features and polynomial degrees 2 and 3 degree using the 14 rows listed in section 1.3. Anything beyond polynomial degree 3 caused R-Studio to freeze as it required too much memory so testing for overfitting couldn’t be performed.

To this point the models weren’t great based on the r values. So, when the predictions are made, they will be rounded to the nearest whole number to see how accurate they are since regression can predict values between whole numbers.

Output from predictions:



The predictions as seen above are very bad compared to the actual values. The best performing one is the model with all the features even then the quality of the wine isn’t predicted accurately.

## Evaluation of the model(s) and conclusion.

Overall the predictions for this data set using the models were very inaccurate.

The predictions with all the features were only capable of really predicting quality from 5-7 even though the data set started at 3 and ended at 9. This was kind of expected since if we look at all the one feature regression graph the estimates are mostly within the quality of 5-7

The polynomial models, which seemed more accurate due to the R-Values, performed even worse. The estimates were unreasonable at best. We would probably get more accurate results by just using one feature which is unfortunate.

There is a high possibility that the models could be improved by having a wider range of data. If we look at the data, most of the 4k entries are usually within the 6-7 quality rating. In the next iteration it could be worth while creating a data set that has a better balance of entries for predictions.

In conclusion the data set wasn’t fit for linear regression in the way it was utilised. More Data Exploration could be done to see if the results could improve but this is out of scope of the project. It’s a possibility that predicting something like this is near impossible as the quality of wine could be subjective, and the data might be degraded because of it. More investigation into the source of the data could give us a better look into this.

# Decision Trees

## Overview of the Problem

The data set utilised for Decision Trees is the “Adult” data set. The data was extracted from the census bureau database by Barry Becker and contributed to the UCI Repository by Ronny Kohavi and Barry Becker.

UCI Repository: <https://archive.ics.uci.edu/ml/datasets/adult>

The data contains 32561 rows and 15 features. Which include:

1. Age - Numeric
2. Workclass - Categorical
3. Fnlwgt - Numeric
4. Education - Categorical
5. education-num- Numeric
6. marital-status - Categorical
7. occupation - Categorical
8. relationship - Categorical
9. race - Categorical
10. sex - Categorical
11. capital-gain- Numeric
12. capital-loss- Numeric
13. hours-per-week- Numeric
14. native-country - Categorical
15. salary – Categorical (>50K or <=50K)

For this specific problem, decision trees will try to determine whether a person makes over 50K a year. It will use all of the parameters to do so.

## Data Exploration (tables and graphs)

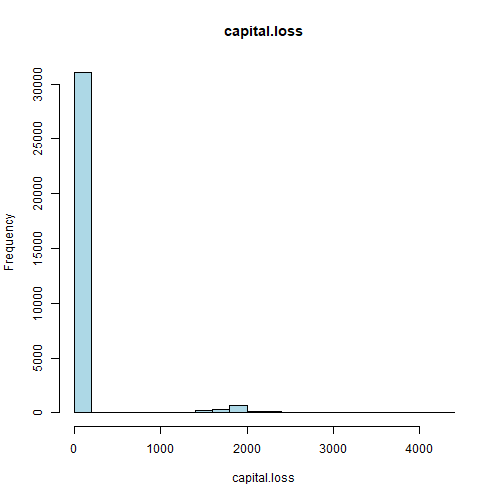
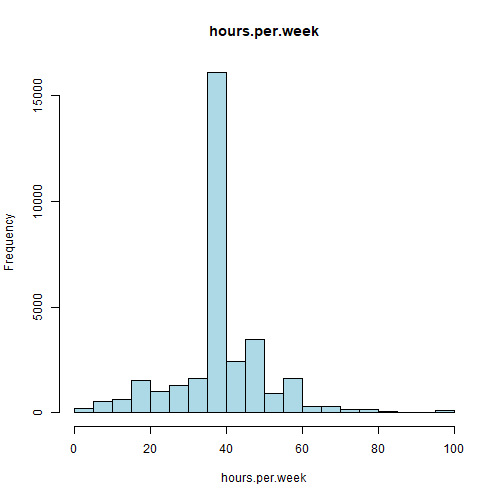
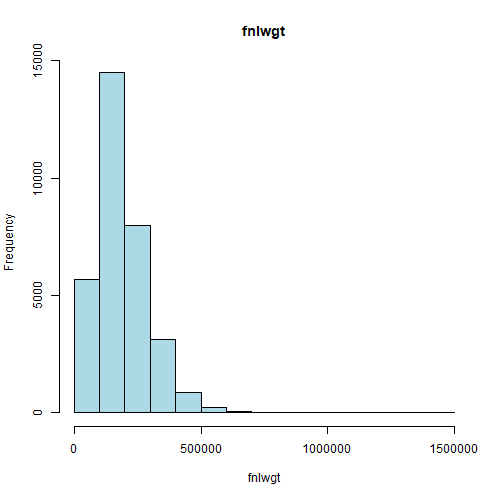
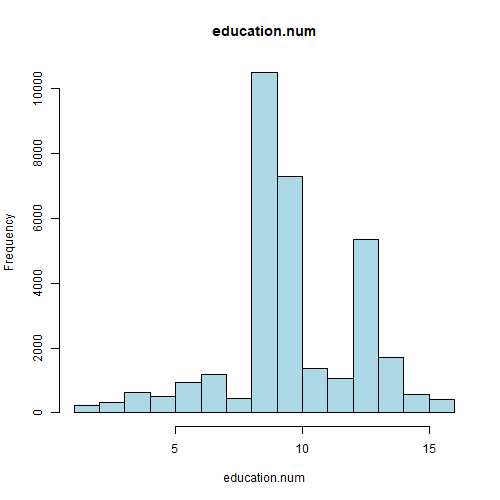
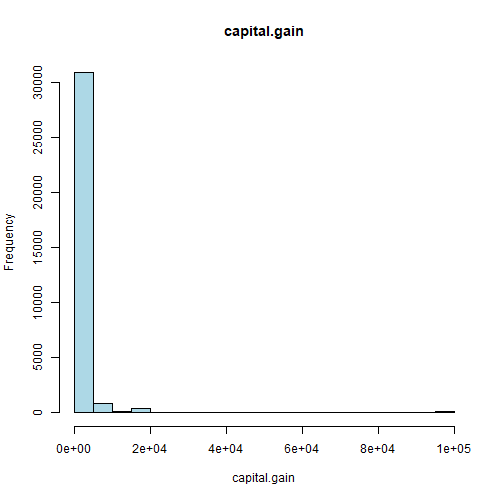
The “Adult” data set has a few missing values. There is no need to clean them up since the decision tree algorithm can handle them. With this many rows it’s very hard to come across them. When loading in the data set into R all the headings had to be passed in separately as they weren’t set in the data.

This data has both numerical and categorical data. Below you can find a table describing each feature individually.

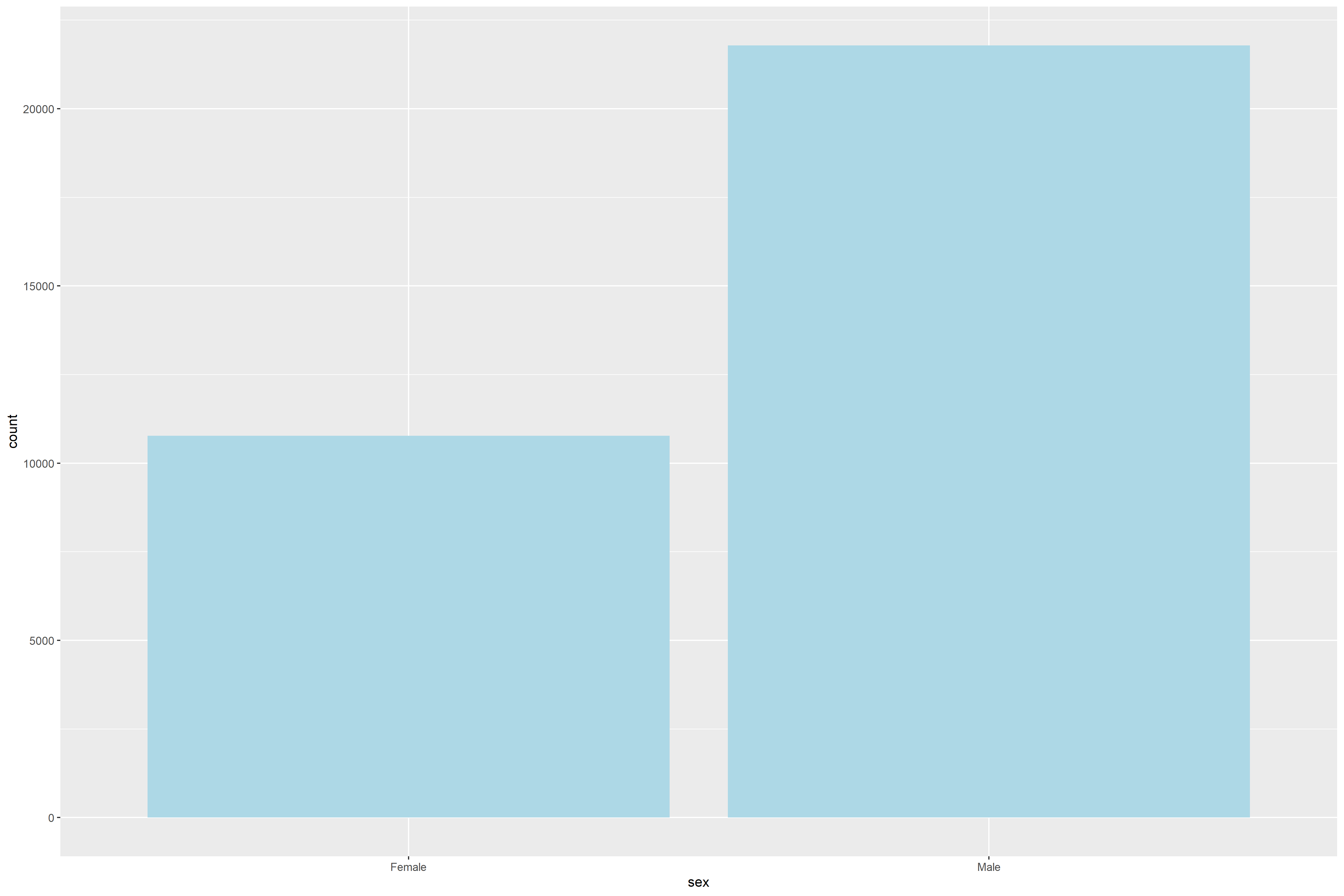
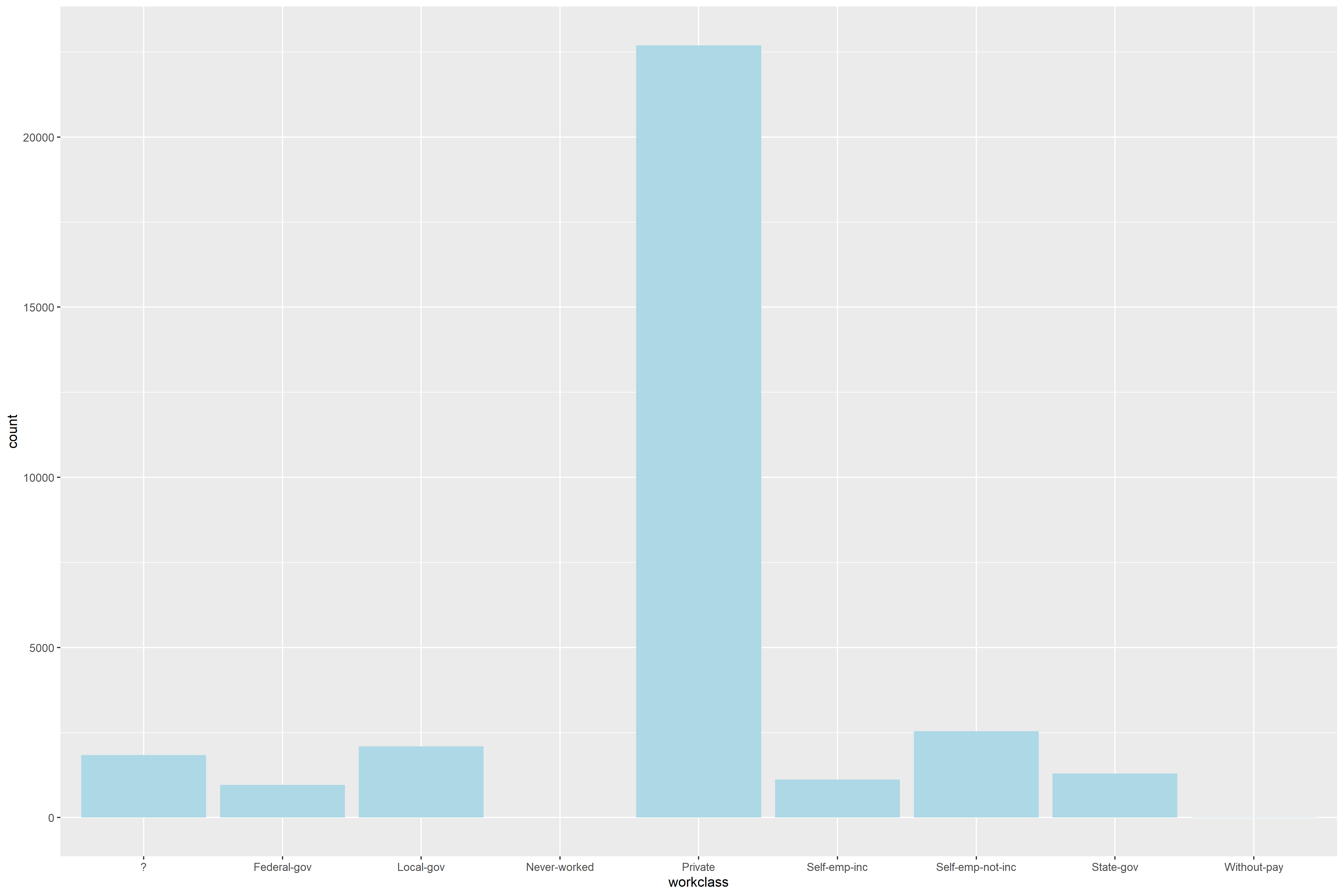


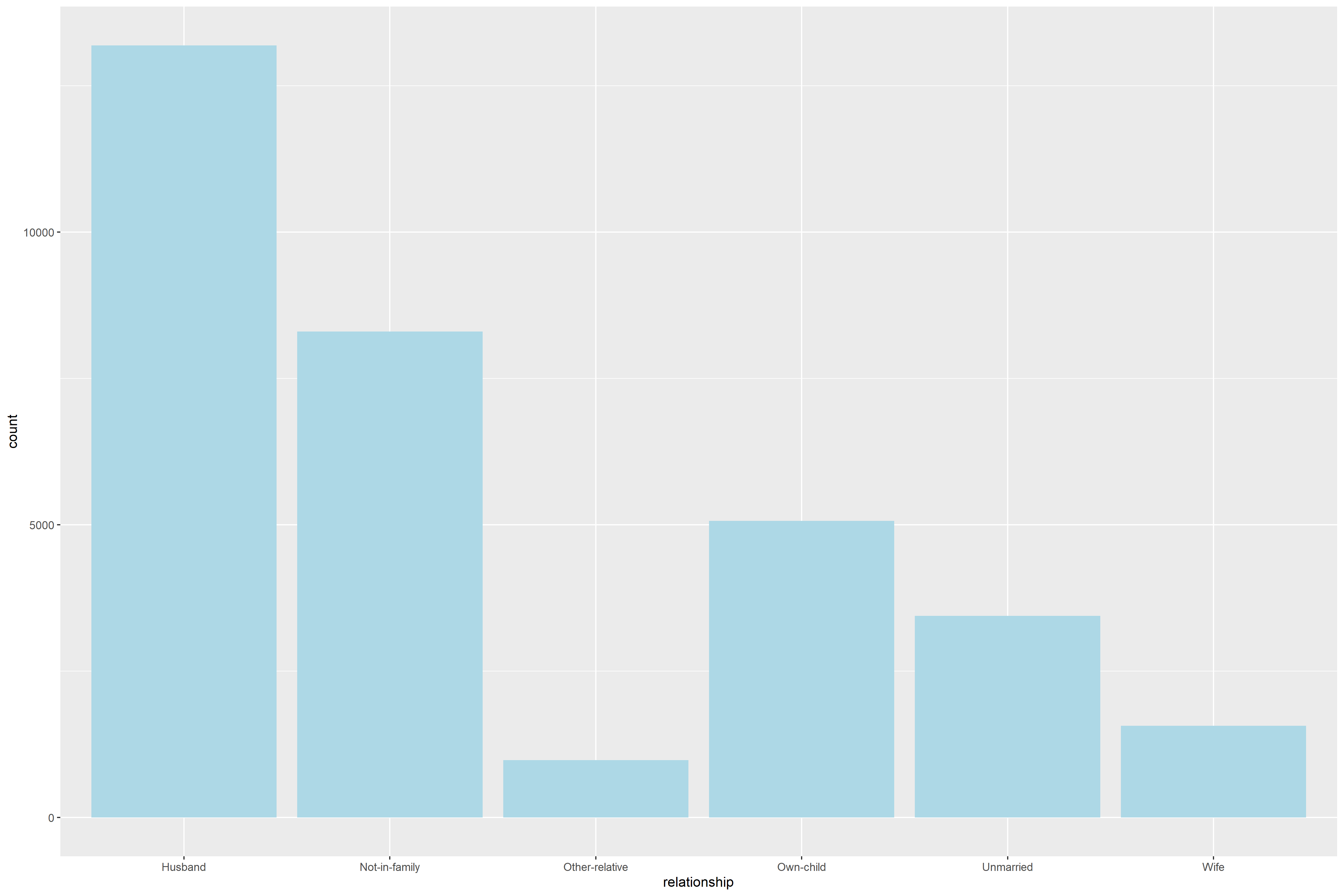
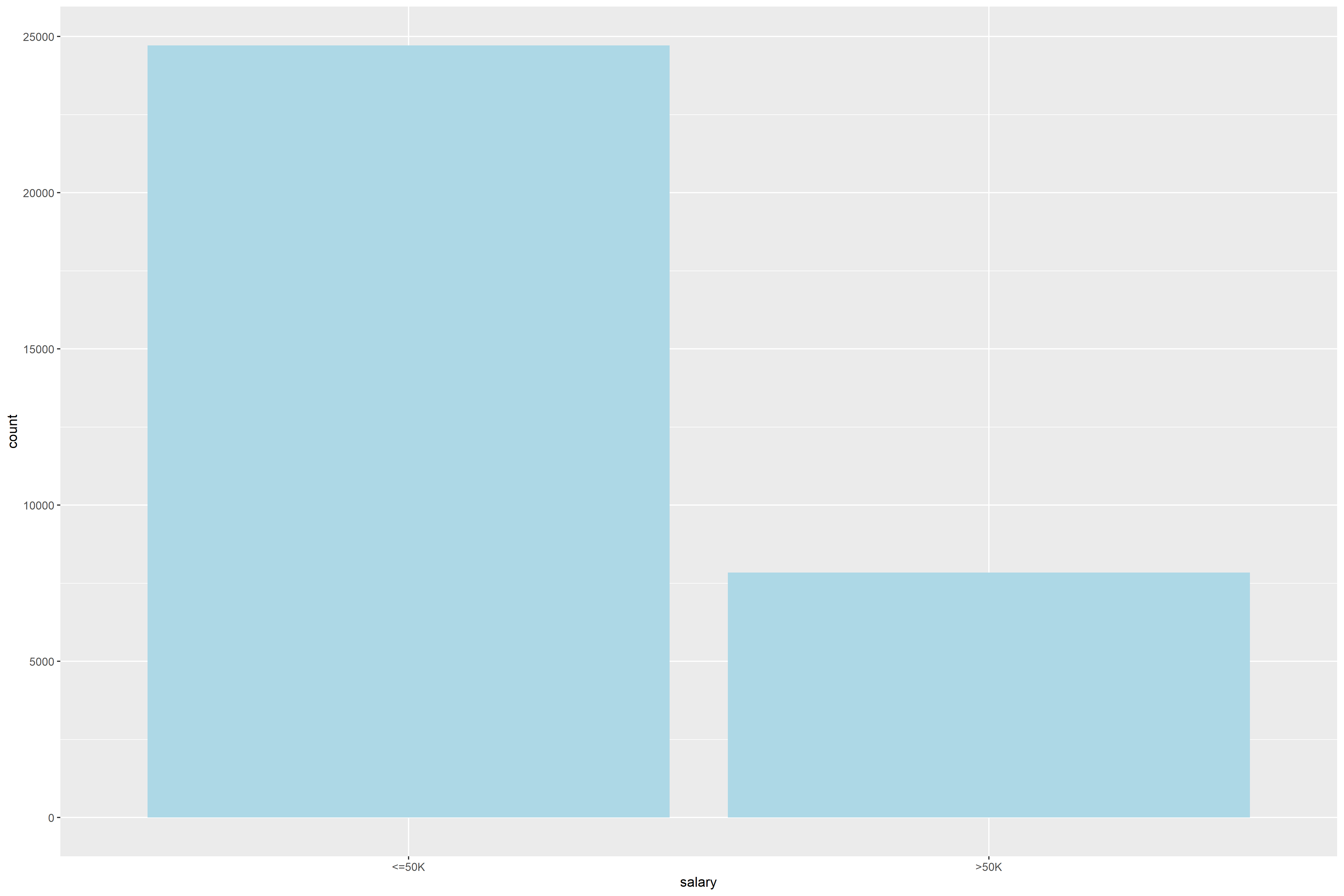
Histograms/Bar charts of the data were created so the distribution of the data could be visualised easily. From the charts below we can see that the distribution of the data is uneven. The split between salary (>50K or <=50K) is around 3:1 which could impact how accurately the tree is able to make predictions in the real world.

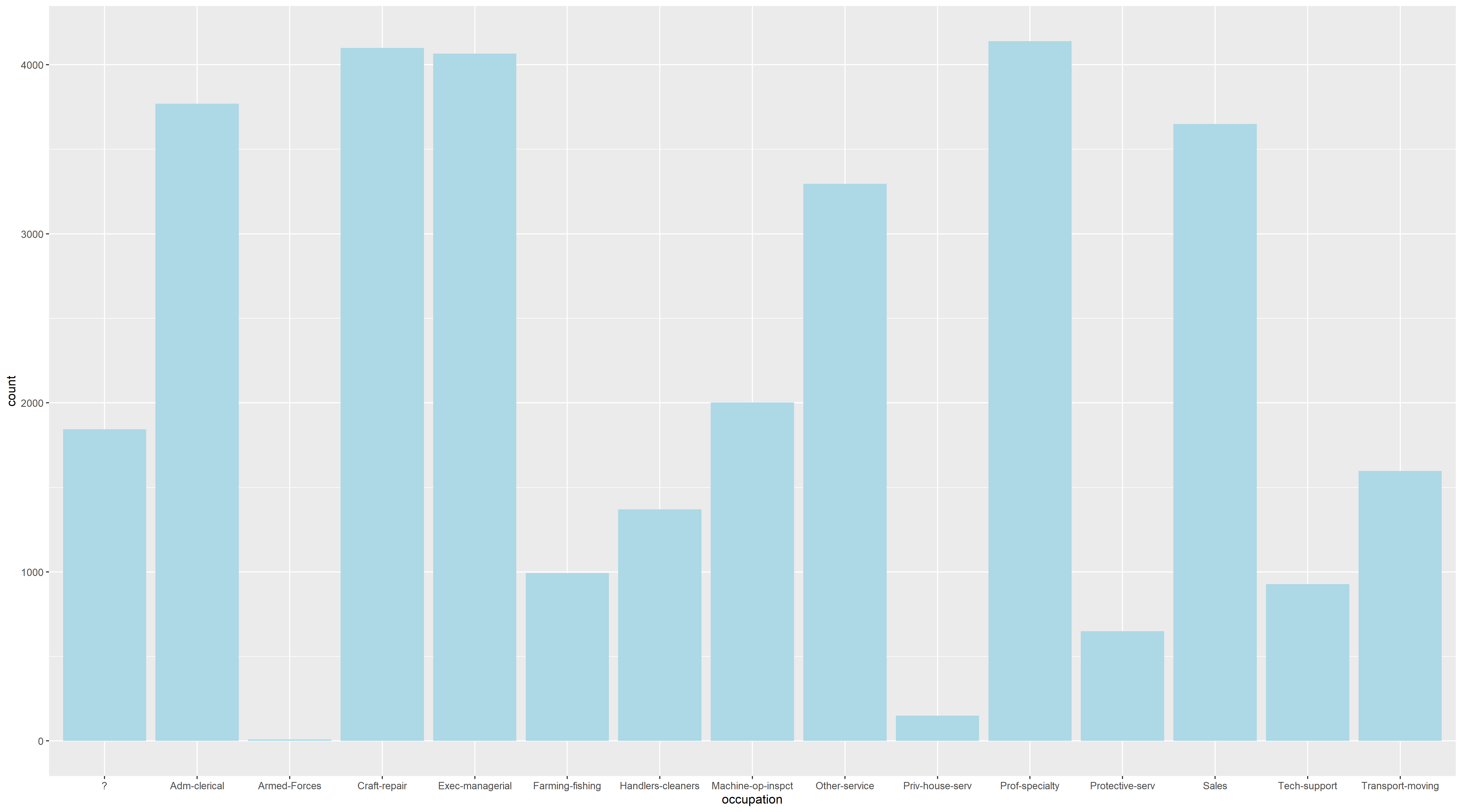
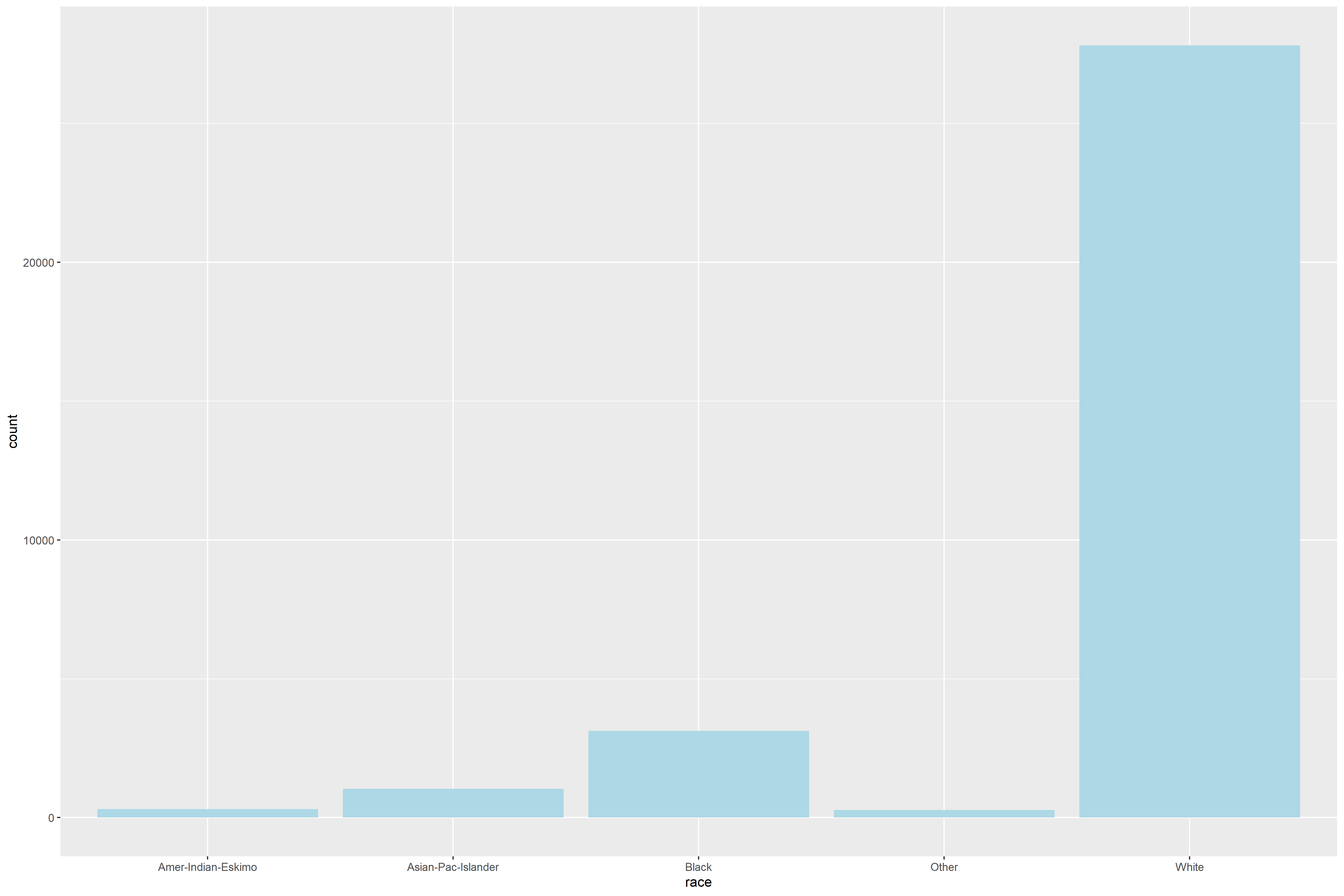
Histograms for numeric values:

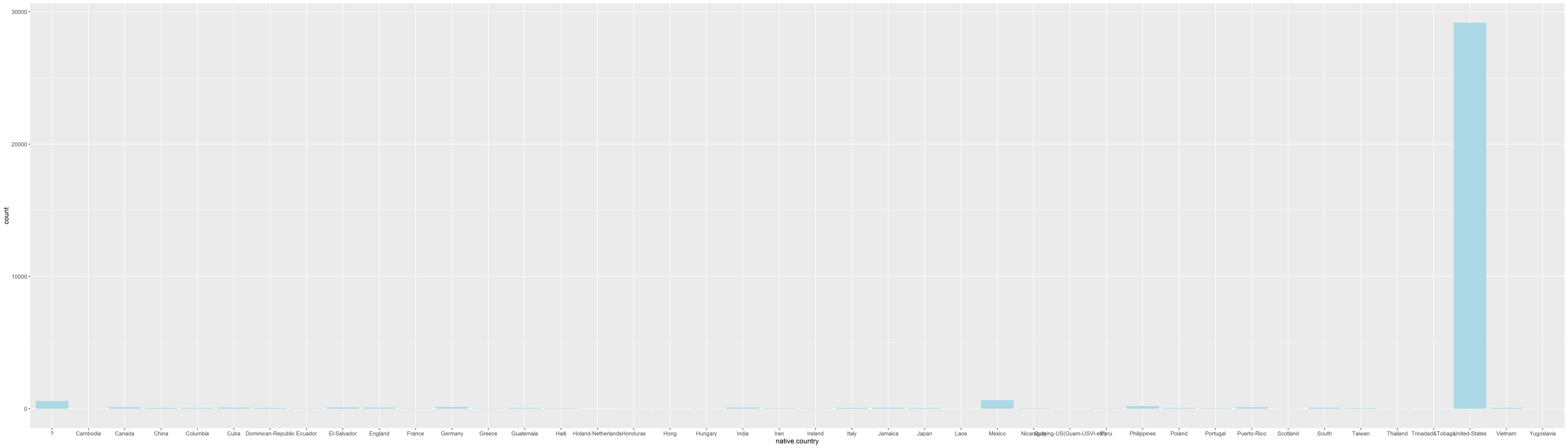


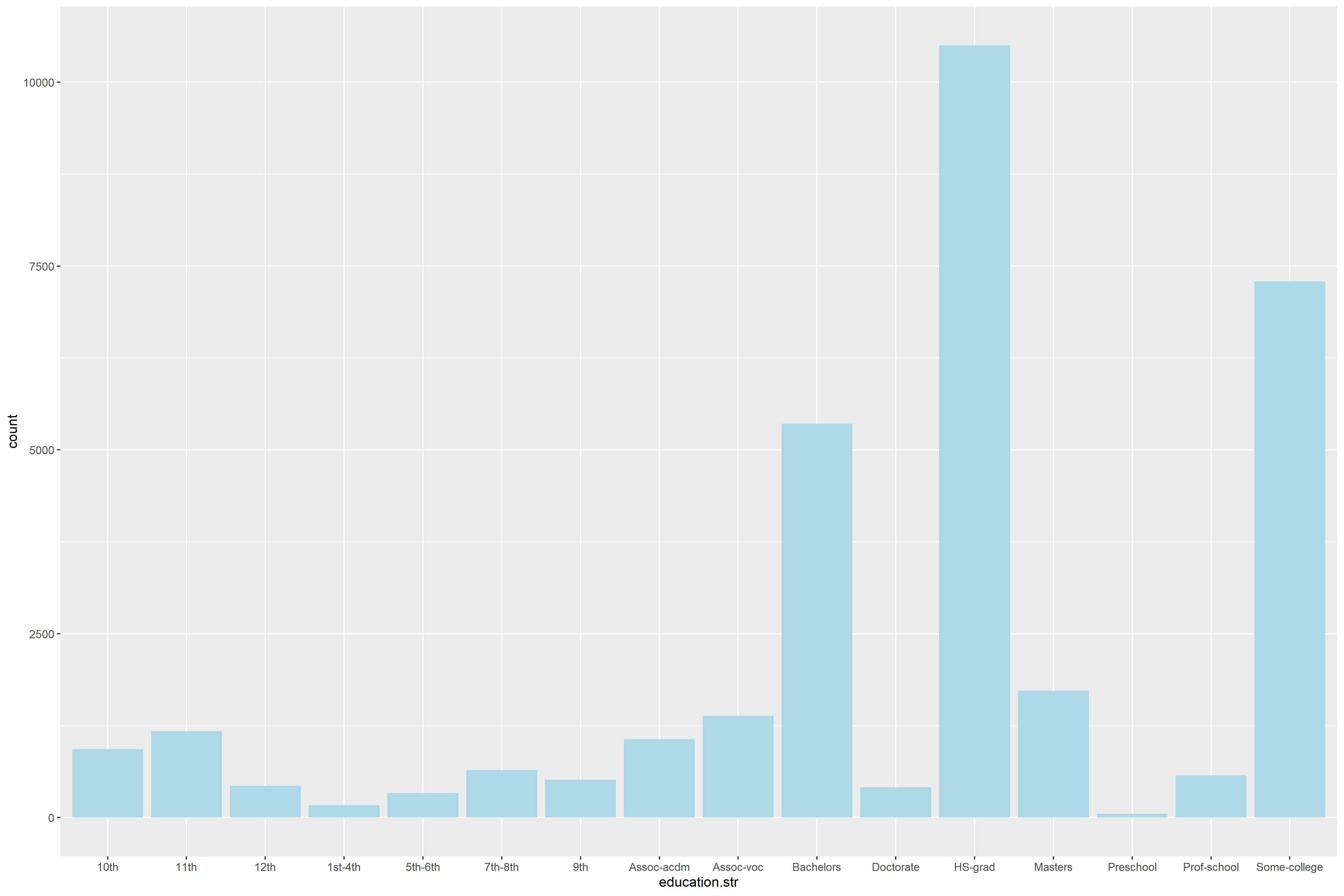
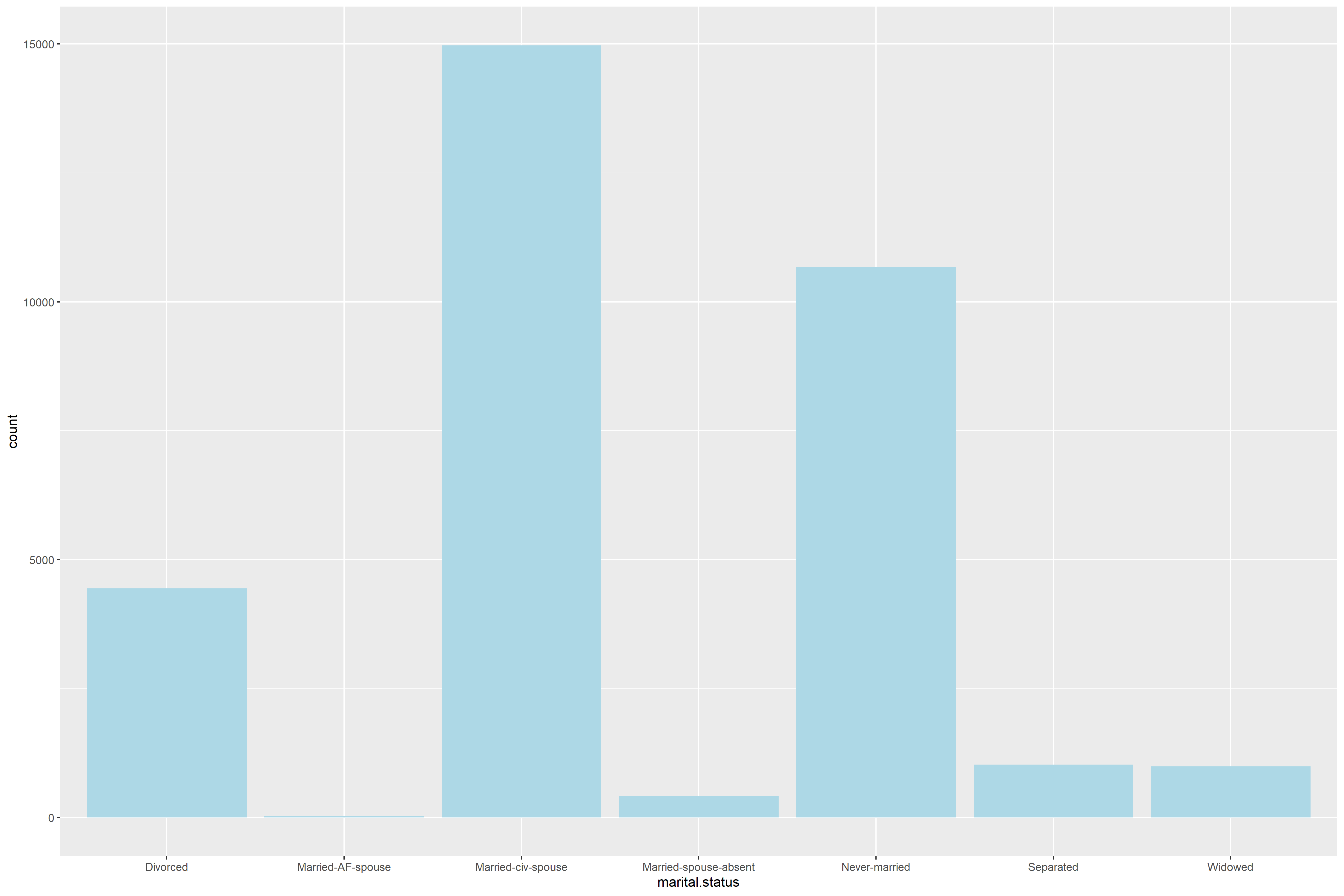
Bar charts for factors:











Bigger versions of these graphs can be found in the diagrams directory in the R project.

## Definition of Training and Testing Set

The training set is split into 70% Training data and 30% Test data. To split the 32561 rows, the order was first randomised to make sure the data wasn’t ordered in any way. 2/3 were assigned to Training and 1/3 to Testing.

The two sets were then compared to see if the sets have a similar break down.

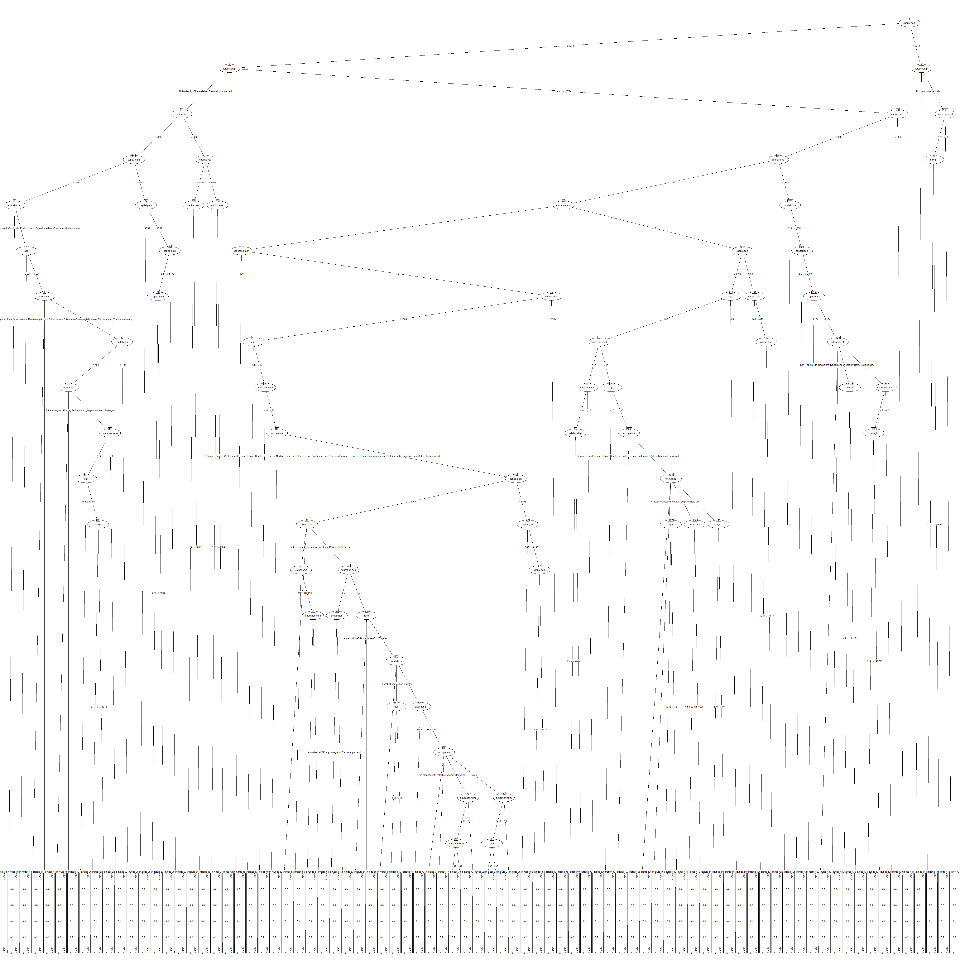
|  |  |  |
| --- | --- | --- |
|  | <=50K | >50K |
| Training | 0.7597328 | 0.2402672 |
| Testing | 0.7581314 | 0.2418686 |

In this case, both have a very similar split, which is similar the split in the bar chart for salary in section 2.2, telling us that the randomly allocated data was split well. This split tells us that the testing set should depict the training set for the model well. When it comes to <=50K, there are more values which could make the models more skewed towards categorising unseen data as this since the fit might be better especially since decision trees use the best fitting term on impure nodes.

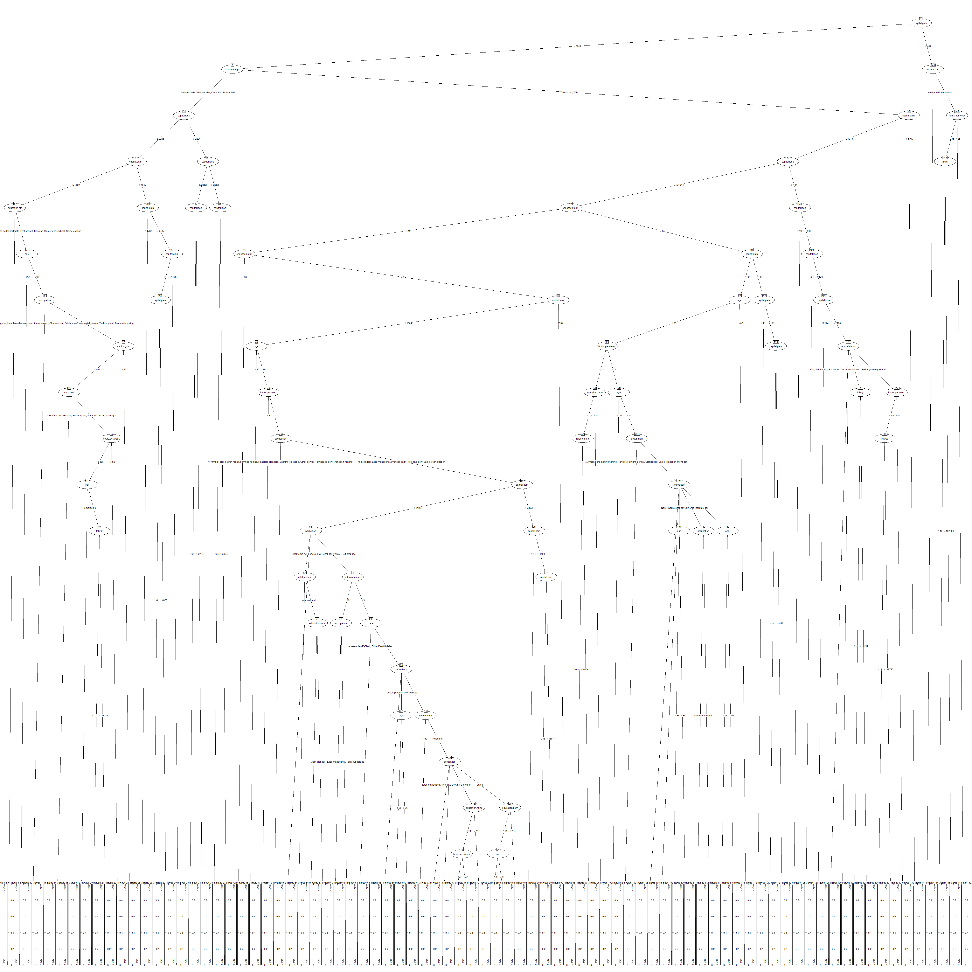
## Model Generation and Information

5 Models were generated with all of the data from the training set. One model with no boosting and 4 with increasing levels of boosting using the trails parameter.

With the amount of data used the decision trees are very complex. This makes it hard to manually evaluate them. Below you can see 2 of the complex decision trees generated. To see the trees in full, go to the diagrams/dicision\_trees directory in the R project. All of them are named “dt boost”



Decision tree with no boosting:



Decision tree with boosting (trails=45):

These decision trees don’t seem to be impacted very much by boosting from evaluation of the exported images.



When evaluating the decision trees using the output from the model, we can see that the boosted trees are a small bit less complex. Their complexity goes up as the amount of trails increases.

## Predictions for the test data

Cross Table:

|  |  |
| --- | --- |
| No Boost |  |
| Trails=3 |  |
| Trails=7 |  |
| Trails=15 |  |
| Trails=45 |  |

Accuracy:

|  |  |
| --- | --- |
| No Boost | 0.867778494425504 |
| Trails=3 | 0.866580668939464 |
| Trails=7 | 0.865843545563439 |
| Trails=15 | 0.869160600755551 |
| Trails=45 | 0.87450474523173 |

When analysing the cross table and accuracy of the models all of the outputs are very similar which is expected as the models produced were very similar too. The only real difference is the amount of false positives and negatives.

## Evaluation of the model(s) and conclusion.

All of the models generated for the “Adult” data set to evaluate if salary is over 50k are very similar. Small tweaks to the False Positives and False Negatives have been made with boosting. Any of these models could be used to make estimates as they are all around 86% accurate.

As a next step maybe if less data was used for training the model could be simplified and perform just as well. This would have to be tested with another iteration. Using less training data could prevent the decision tree from getting so large yet still give an accurate result.

From the models that were generated we can conclude that estimating salary using information about a person is possible and accurate. The models generated do a good job of this too.

# kNN

## Overview of the Problem

The data set that will be used for kNN is the “Wholesale customers” data set. The data originates from a larger database and was sourced by Margarida G. M. S. Cardoso and contributed to the UCI Repository.

UCI Repository: <https://archive.ics.uci.edu/ml/datasets/Wholesale+customers>

The data contains 440 rows and 8 features which are all numeric. Which include:

1. Fresh: annual spending on fresh products
2. Milk: annual spending on milk
3. Grocery: annual spending on groceries
4. Frozen: annual spending on frozen products
5. Detergents\_Paper: annual spending on detergents and paper
6. Delicatessen: annual spending on and delicatessen products
7. Channel: Horeca (Hotel/Restaurant/Café) (1) or Retail channel (2)
8. Region: customers’ Region

For this specific problem, using the kNN algorithm, the channel will be predicted. This will allow us to predict if a Hotel/Restaurant/Café or Retail Channel was responsible for the purchases made. To best categorise the channel responsible for purchasing goods all parameters should be used.

## Data Exploration (tables and graphs)

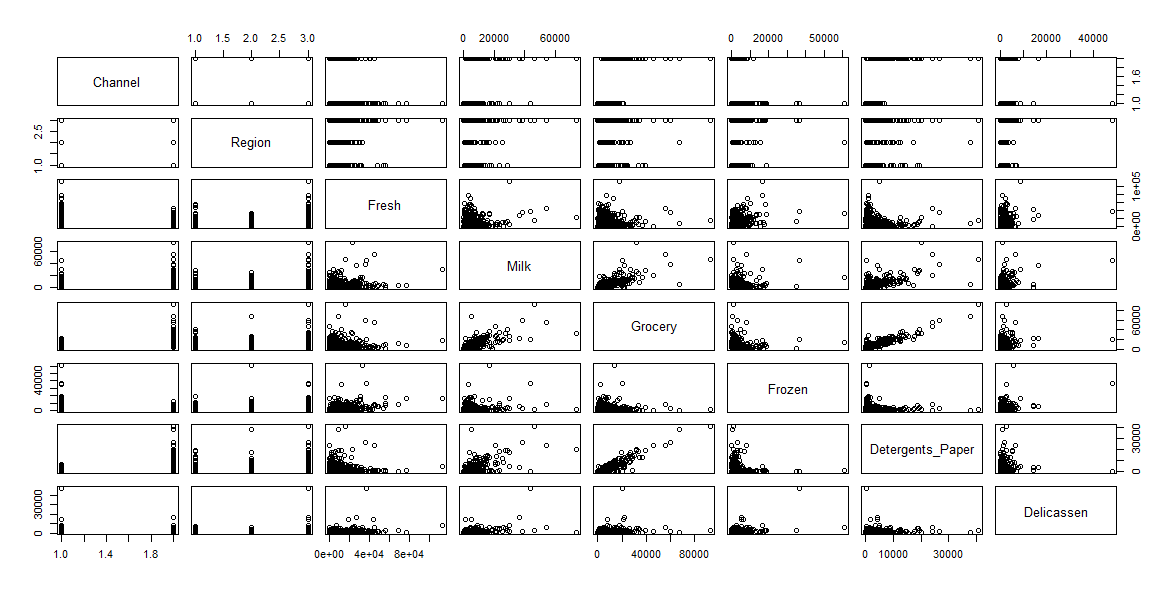
The “Wholesale customers” data has no missing values so there is no need to clean them up.

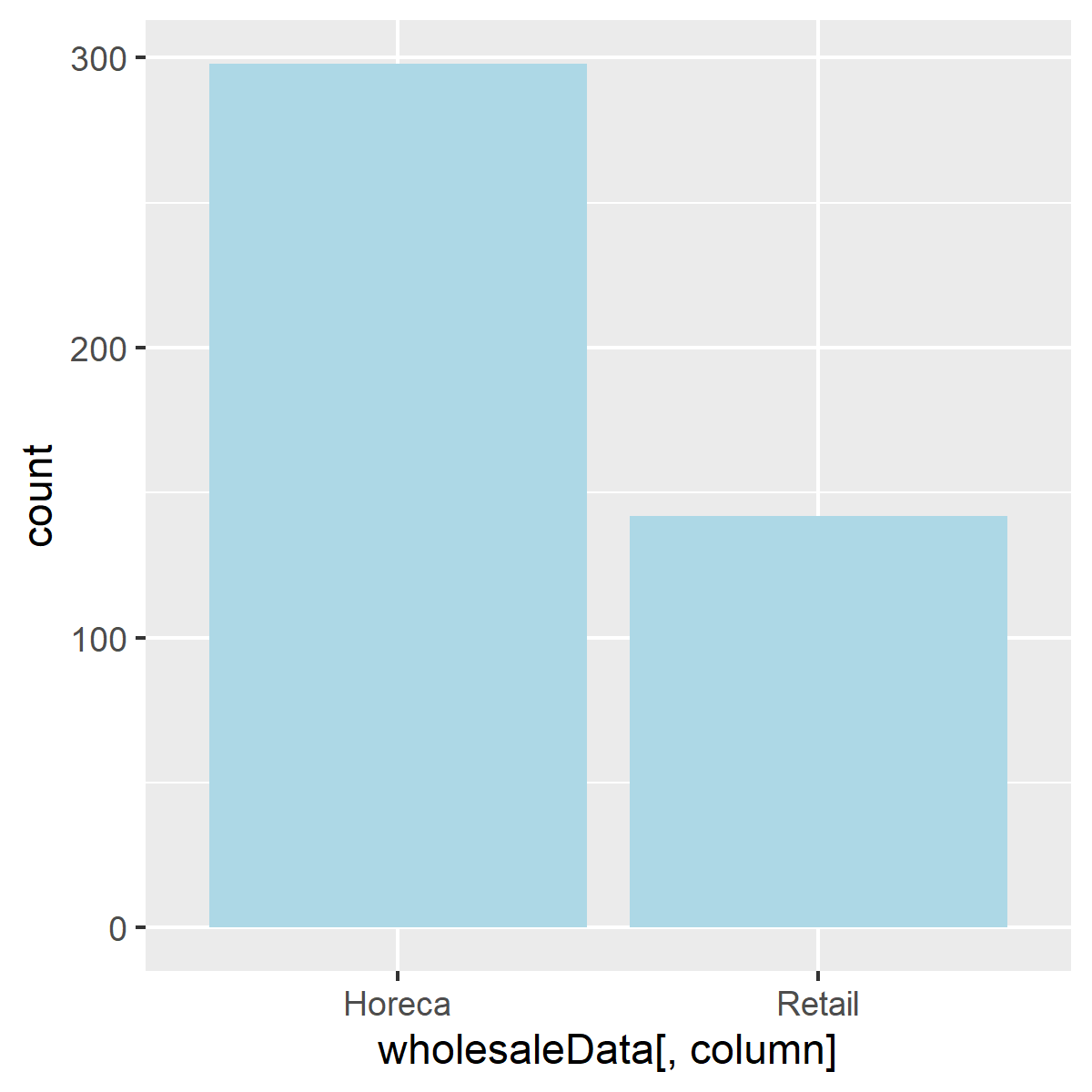
The channel is a number (1 or 2) for readability purposes when loading the data set it was turned into a factor.

The data only contains numeric values which is perfect for kNN. Below you can find a table describing each feature individually. From an analysis of the table we can see that the values have a very wide and different min/max. This could affect the kNN algorithm as it uses distance for categorisation. To get a more effective model the data might need to be scaled.

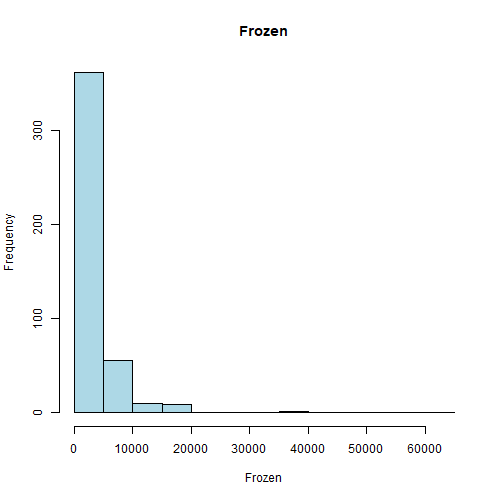
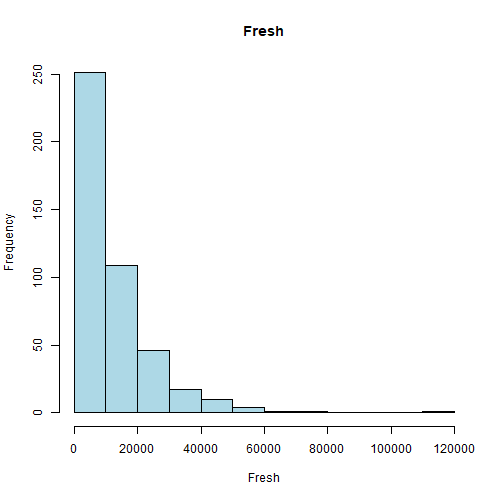
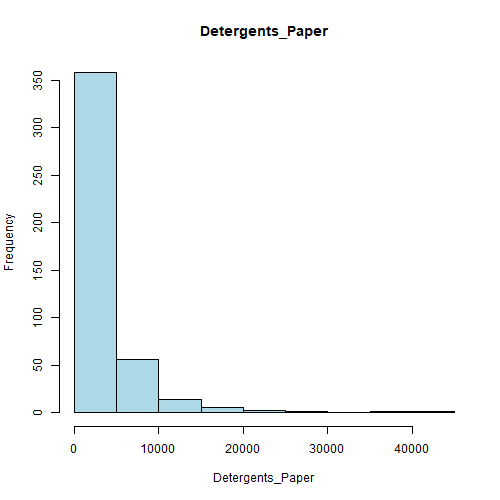
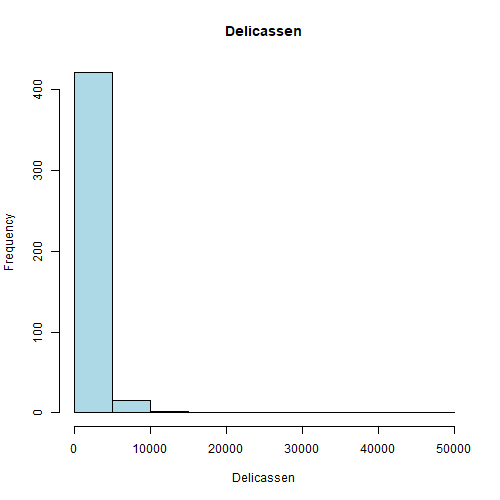


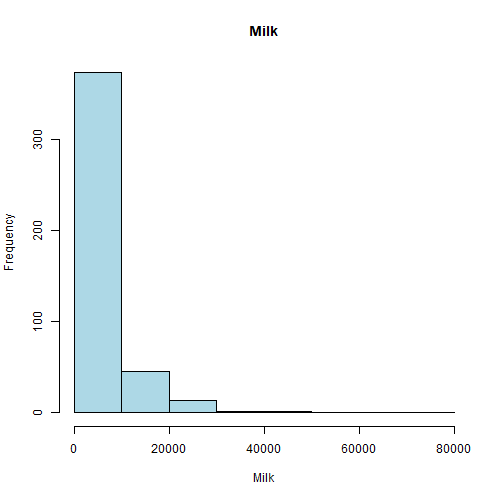
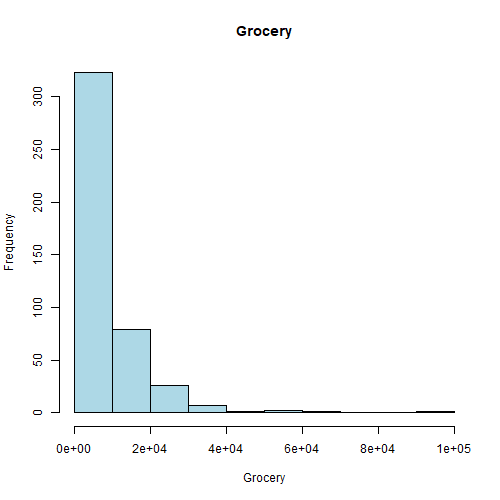
Below we have a graph that depicts all the values plotted against each other. In the diagram below just by looking we can see we have some linear/polynomial correlations. For the channel we can see some difference between how much each channel is willing to spend on each type of goods. This could allow the kNN algorithm to correlate features together to give better results.





Our split for data is about 2:1 (Horeca:Retail) based on the bar chart above





From the histograms above we can see that our distribution looks like a log function going to a limit of X. This could imply that our data set has a few outliers that could skew predictions.

## Definition of Training and Testing Set

The training set is split into 70% Training data and 30% Test data. To split the 440 rows, the order was first randomised to make sure the data wasn’t ordered in any way. 2/3 were assigned to Training and 1/3 to Testing.

The two sets were then compared to see if the sets have a similar break down.

|  |  |  |
| --- | --- | --- |
|  | Horeca | Retail |
| Training | 0.6724138 | 0.3275862 |
| Testing | 0.6866667 | 0.3133333 |

In this case, both have a very similar split, which is similar the split in the bar chart for channel in section 3.2, telling us that the randomly allocated data was split well. This split tells us that the testing set should depict the training set for the model well. When it comes to Horeca, there are more values which could make the models more skewed towards categorising unseen data as this since the fit might be better.

## Model Generation and Information

In kNN, the concept of models doesn’t really exist. We make predictions by passing training data, a k value and values we want to predict to get our predictions.

30 predictions were made with all the data from the training and testing set. 15 of these were done as is and 15 used z-scaling on the data frame. Each one of the 15 had a different k value, ranging from 1 to 15. Z -Scaling was used to prevent data features from having different weights in hopes of categorising the data better. These can be seen in the section below in more detail.

## Predictions for the test data

30 predictions were made. 15 of these with no scaling and 15 used z-scaling. Each one of the 15 had a different k value, ranging from 1 to 15. To visualise the performance of the kNN algorithm, a Cross Table will be used to show the correct and incorrect classifications.

Cross Table:

|  |  |  |
| --- | --- | --- |
| ***K*** | ***No Scaling*** | ***Z-Scaling*** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

When analysing the cross table of the predictions, all the outputs are very similar. The biggest difference is usually between the non-scaled/z-scaled values but even then, it’s miniscule. Another measure like accuracy or precision needs to be used to evaluate the results in the cross table.

Accuracy:

|  |  |  |
| --- | --- | --- |
| ***K*** | ***No Scaling*** | ***Z-Scaling*** |
| 1 | 0.86 | 0.9 |
| 2 | 0.88 | 0.88 |
| 3 | 0.913333333333333 | 0.893333333333333 |
| 4 | 0.9 | 0.886666666666667 |
| 5 | 0.886666666666667 | 0.92 |
| 6 | 0.9 | 0.906666666666667 |
| 7 | 0.9 | 0.88 |
| 8 | 0.913333333333333 | 0.906666666666667 |
| 9 | 0.9 | 0.886666666666667 |
| 10 | 0.906666666666667 | 0.88 |
| 11 | 0.913333333333333 | 0.886666666666667 |
| 12 | 0.9 | 0.886666666666667 |
| 13 | 0.913333333333333 | 0.886666666666667 |
| 14 | 0.926666666666667 | 0.886666666666667 |
| 15 | 0.906666666666667 | 0.886666666666667 |

To analyse the predictions further I decided to opt for accuracy. As we can see in the table above in general the non-scaled data seems to perform better overall but only by about 2.5%

## Evaluation of the model(s) and conclusion.

Overall the performance of the predictions was satisfactory. If a k value was to be selected for use in production out of all of these it should probably be k=sqrt(n) where n=number of training data points. Which in this case can be set to k=12.

Accuracy:

|  |  |  |
| --- | --- | --- |
| ***K*** | ***No Scaling*** | ***Z-Scaling*** |
| 12 | 0.9 | 0.886666666666667 |

It was surprising to see that the scaled data performed worse than the non-scaled data when it came to accuracy. This implies that there could be a weighted correlation between some of the features and the channel. Maybe a different algorithm/ weight function could do even better than the ones presented.

# Citations

* Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
* P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236.
* Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996
* Abreu, N. (2011). Analise do perfil do cliente Recheio e desenvolvimento de um sistema promocional. Mestrado em Marketing, ISCTE-IUL, Lisbon