Applied Data Science II - Homework 4

Phileas Dazeley Gaist

30/01/2021

Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(nnet)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

Instructions

Build the most accurate model that you can to predict whether a given individual makes over $50k a year.

Setup

```
income_eval <- read.csv("Homework 4 data/income_evaluation.csv")

# remove na values and ? values
idx <- income_eval == " ?" # find ? elements
is.na(income_eval) <- idx # replace elements with NA
rm(idx) # delete the index variable
income_eval <- na.omit(income_eval) # omit NA rows

# Convert all char columns to factors
income_eval <- as.data.frame(unclass(income_eval),
                       stringsAsFactors = TRUE)

# since education and education.num seem to contain the same information, I will
# drop education.num.
income_eval <- income_eval %>% dplyr::select(-c(education.num))

#Preview the data
head(income_eval)
```

```
##   age        workclass fnlwgt  education      marital.status
## 1  39        State-gov  77516  Bachelors       Never-married
## 2  50 Self-emp-not-inc  83311  Bachelors  Married-civ-spouse
## 3  38          Private 215646    HS-grad            Divorced
## 4  53          Private 234721       11th  Married-civ-spouse
## 5  28          Private 338409  Bachelors  Married-civ-spouse
## 6  37          Private 284582    Masters  Married-civ-spouse
##          occupation  relationship  race    sex capital.gain capital.loss
## 1      Adm-clerical Not-in-family White   Male         2174            0
## 2   Exec-managerial       Husband White   Male            0            0
## 3 Handlers-cleaners Not-in-family White   Male            0            0
## 4 Handlers-cleaners       Husband Black   Male            0            0
## 5    Prof-specialty          Wife Black Female            0            0
## 6   Exec-managerial          Wife White Female            0            0
##   hours.per.week native.country income
## 1             40  United-States  <=50K
## 2             13  United-States  <=50K
## 3             40  United-States  <=50K
## 4             40  United-States  <=50K
## 5             40           Cuba  <=50K
## 6             40  United-States  <=50K
```

Exercise

Preparing the data:

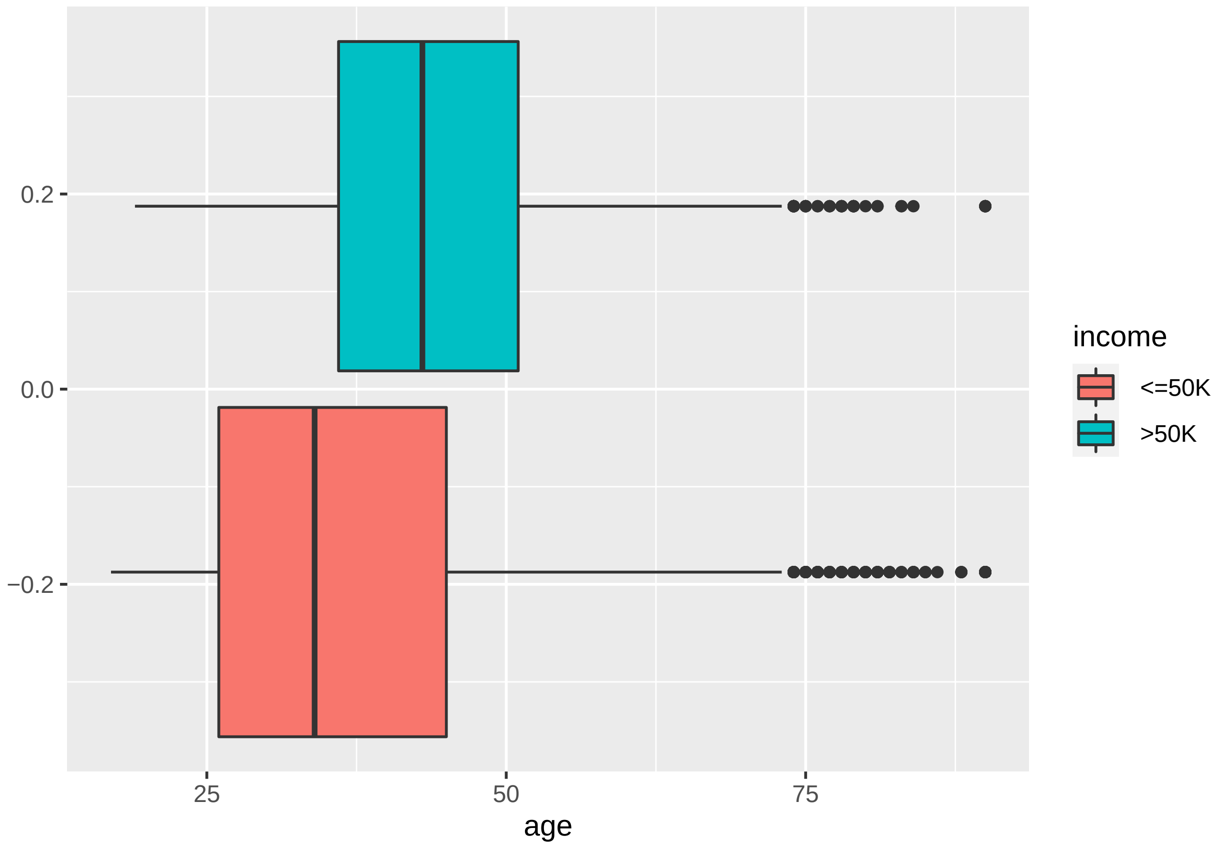
```r
set.seed(1)

# split the dataset into training and testing sets
training_samples <- income_eval$income %>%
  createDataPartition(p = 0.5, list = FALSE)
train_data  <- income_eval[training_samples, ]
test_data <- income_eval[-training_samples, ]

# check it worked properly
dim(train_data); dim(test_data)
```

```
## [1] 15081    14
```

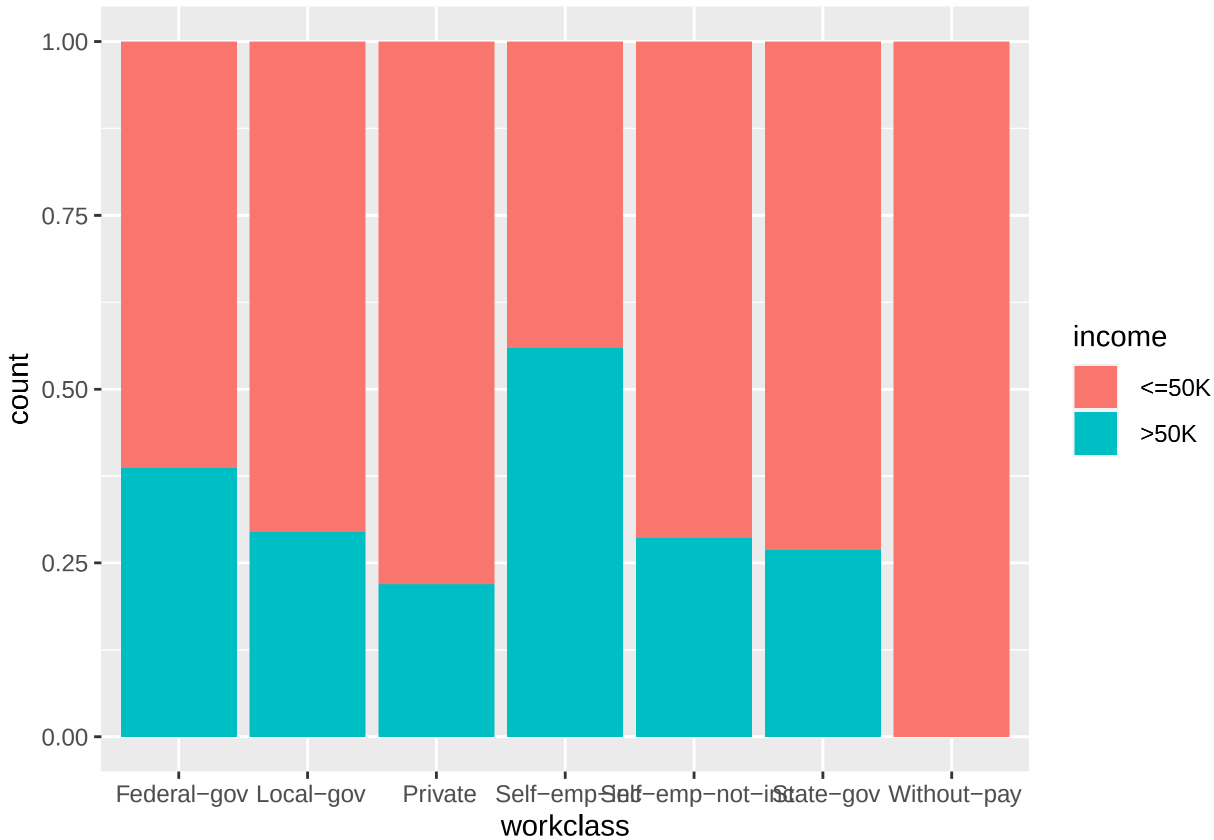
```
## [1] 15081    14
```

Visualising the data

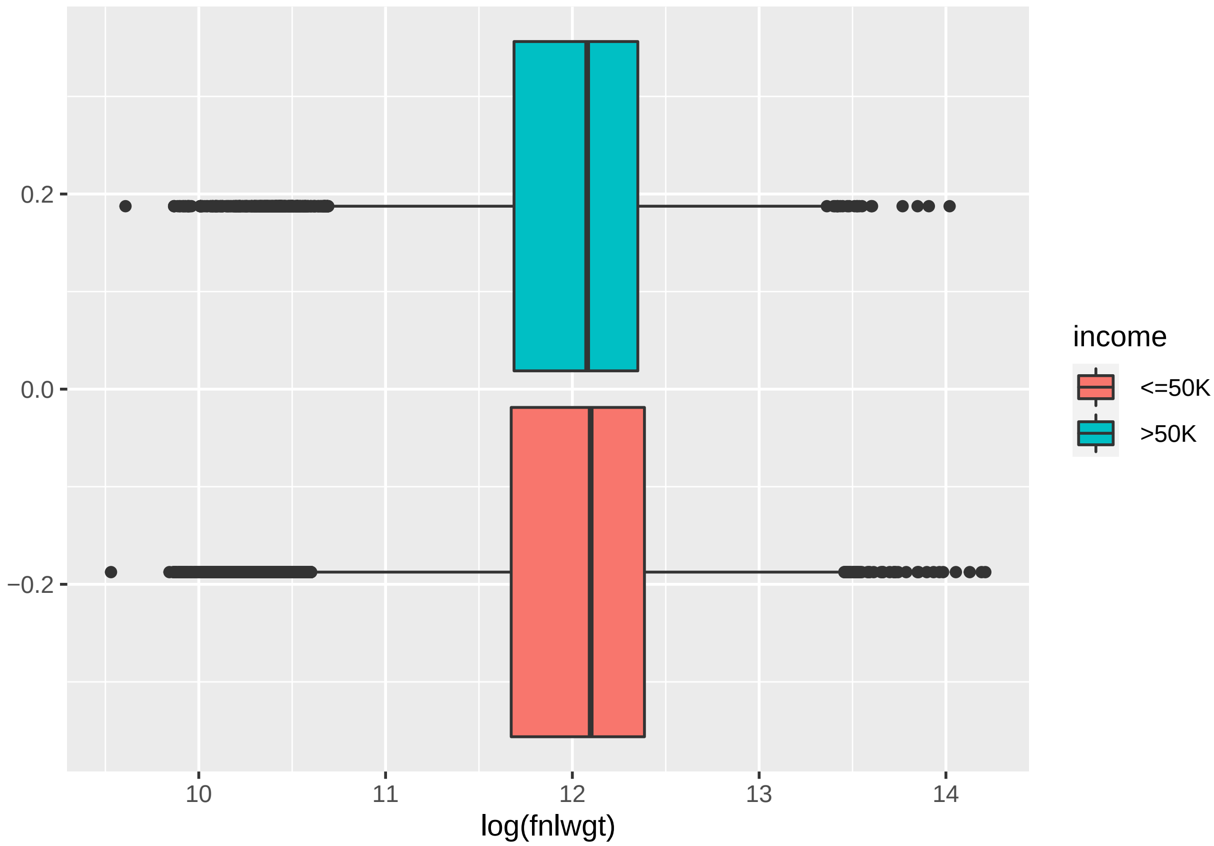
```r
income_eval %>% ggplot(aes(age, fill = income)) + geom_boxplot()
```



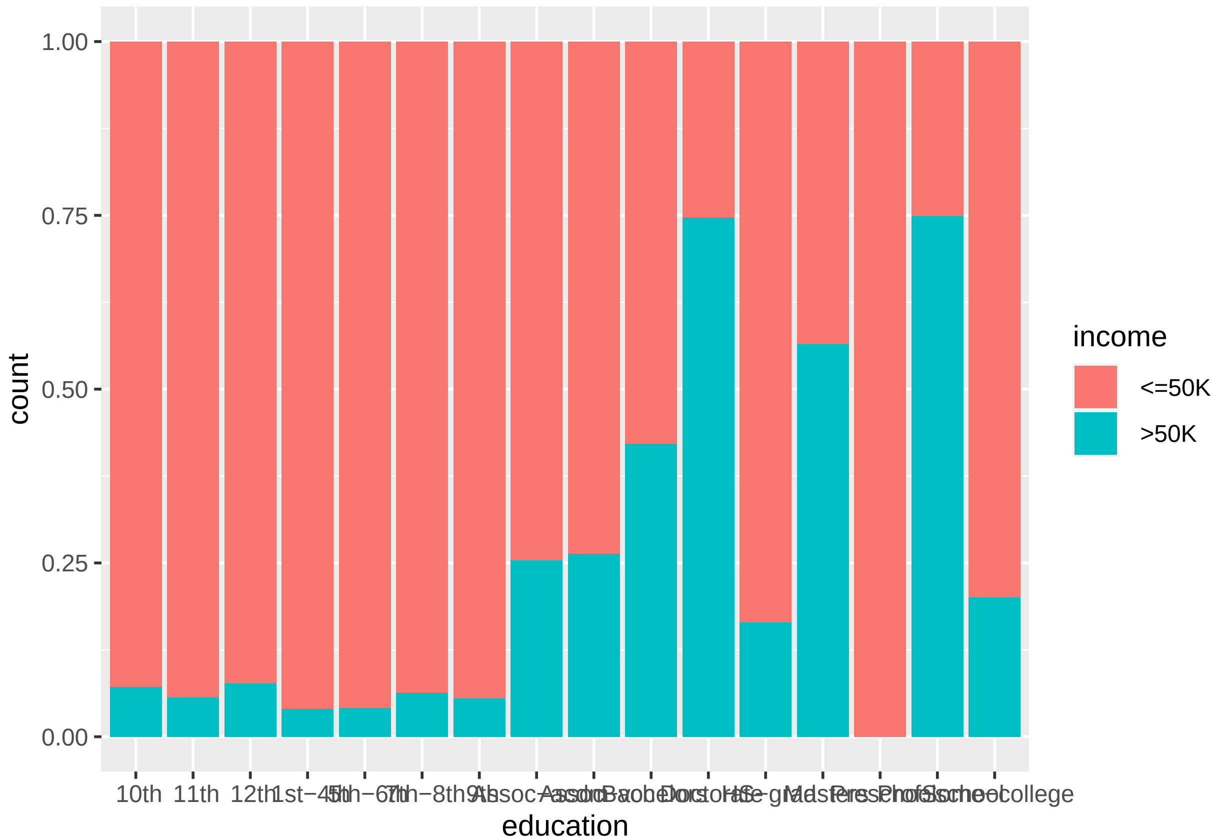

```
income_eval %>% ggplot(aes(workclass, fill = income)) + geom_bar(position = "fill")
```



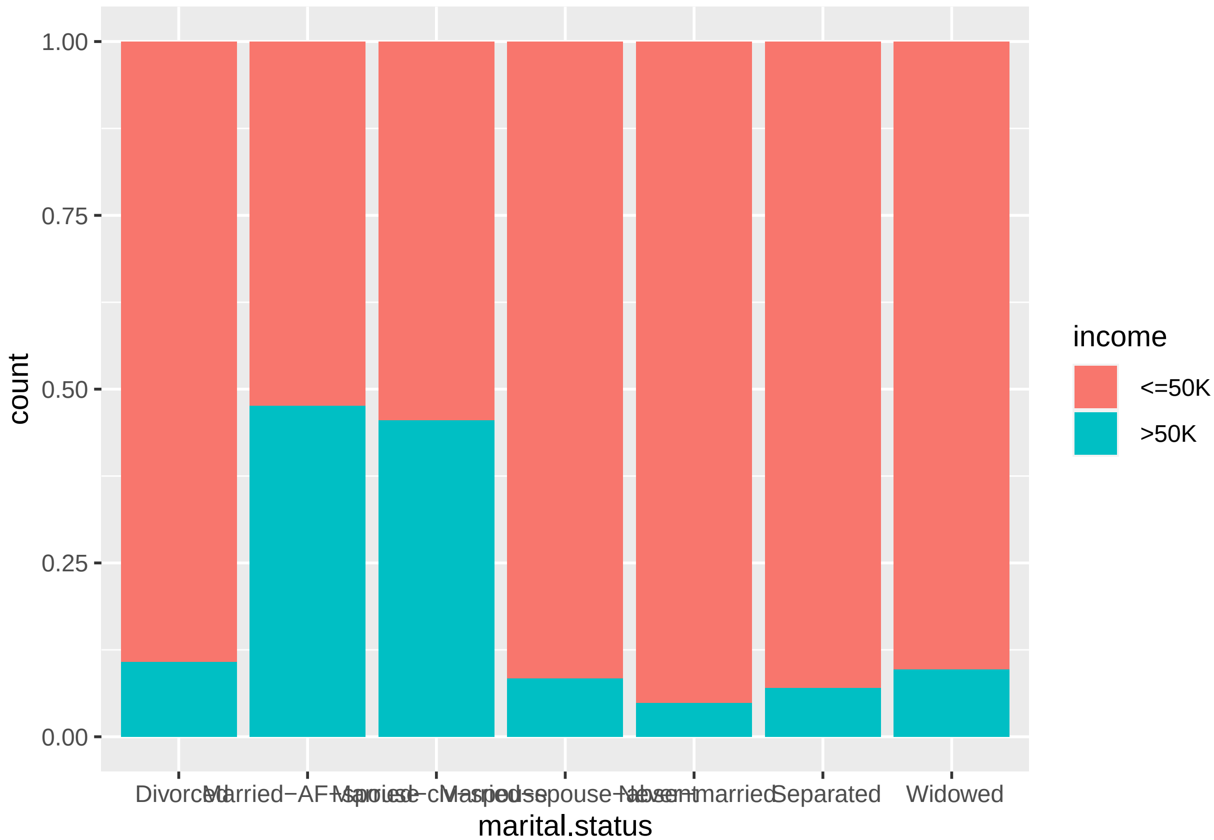

```
income_eval %>% ggplot(aes(log(fnlwgt), fill = income)) + geom_boxplot() # log
```



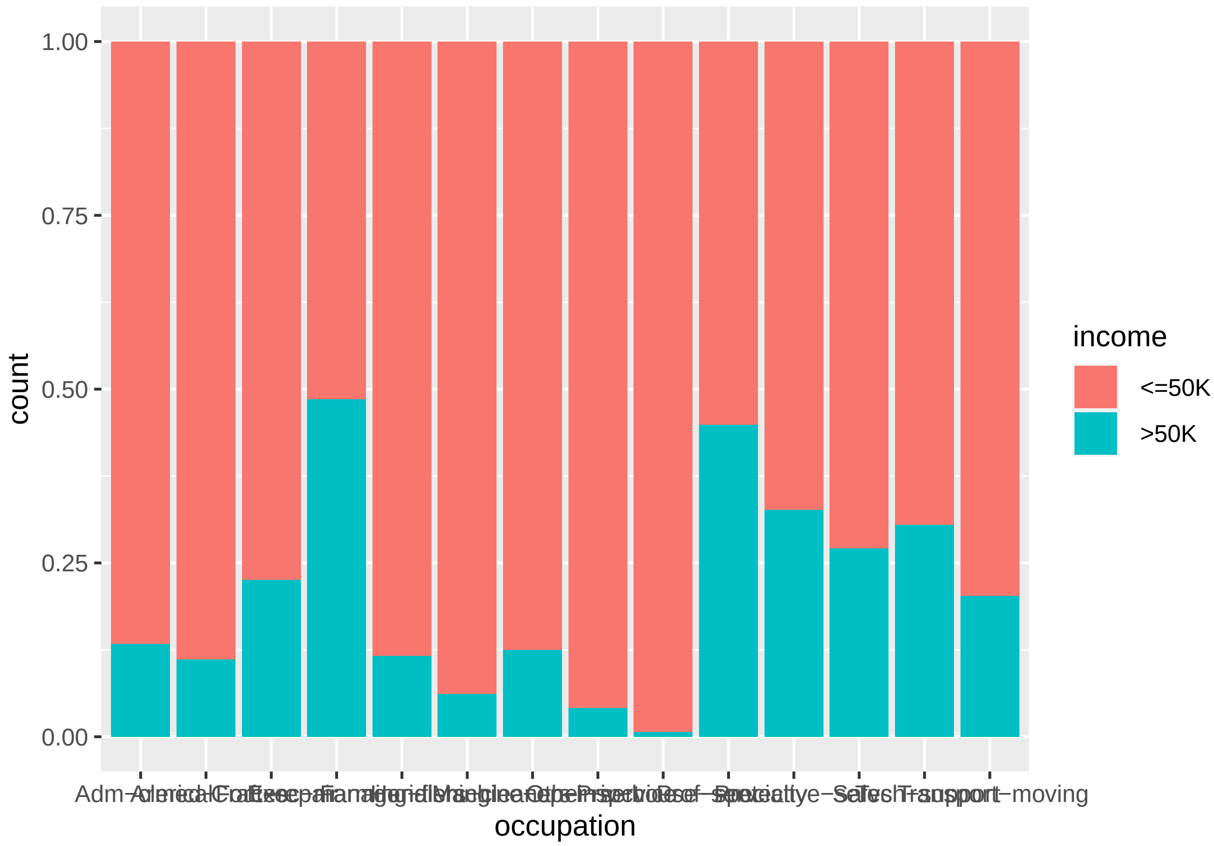

```
income_eval %>% ggplot(aes(education, fill = income)) + geom_bar(position = "fill")
```



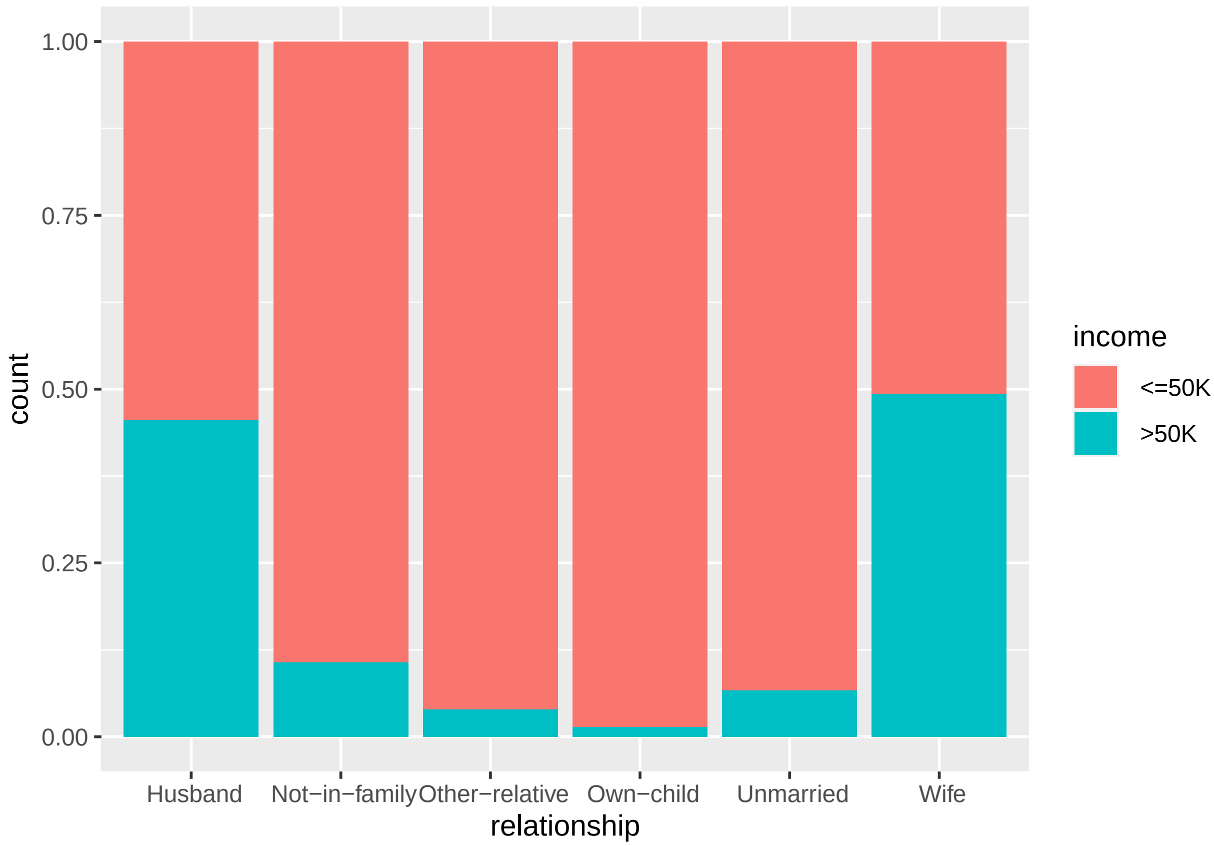

```
income_eval %>% ggplot(aes(marital.status, fill = income)) + geom_bar(position = "fill")
```



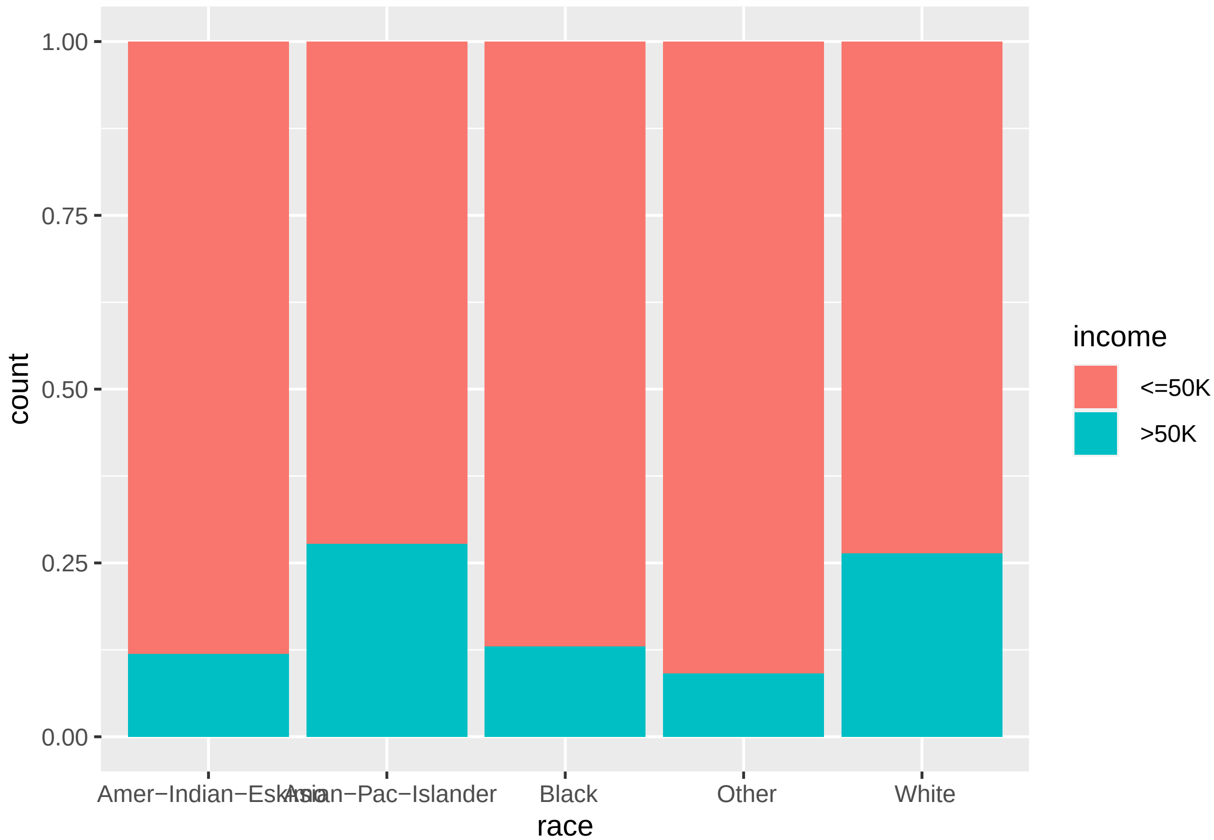

```
income_eval %>% ggplot(aes(occupation, fill = income)) + geom_bar(position = "fill")
```



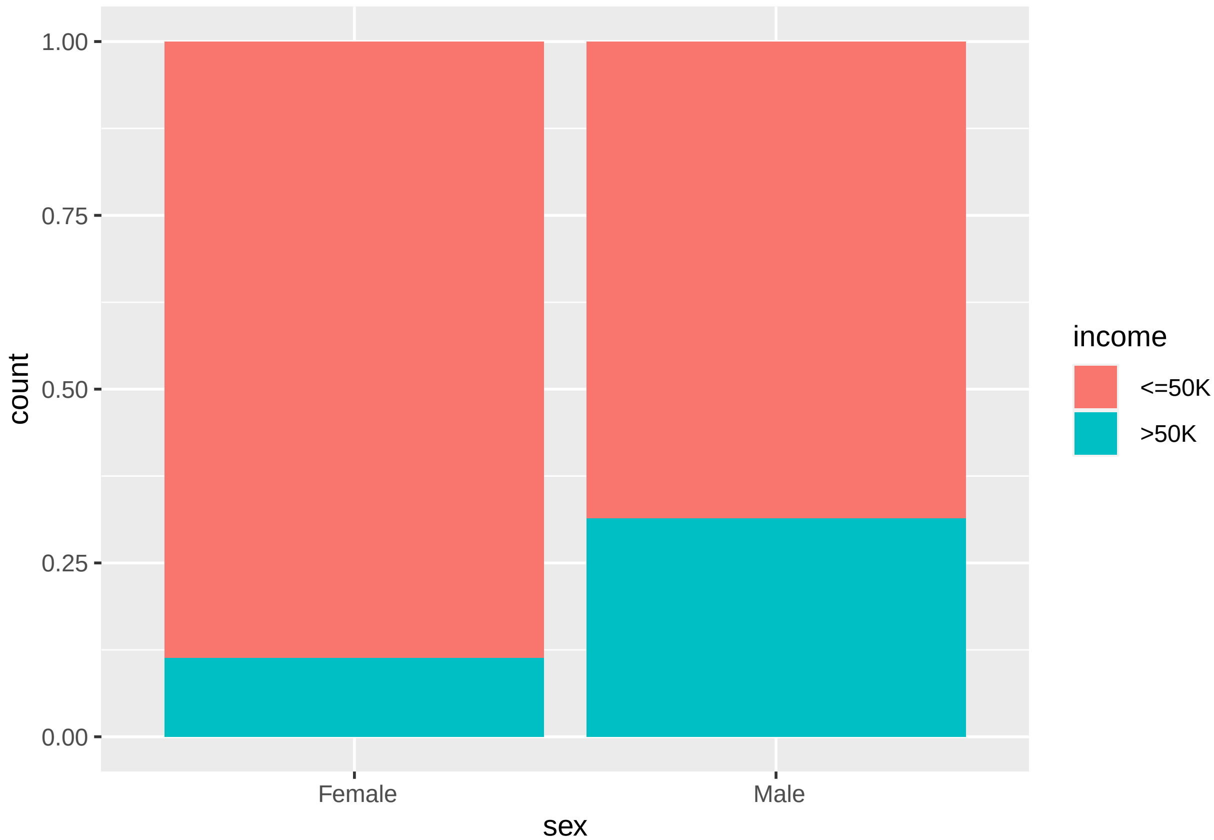

```
income_eval %>% ggplot(aes(relationship, fill = income)) + geom_bar(position = "fill")
```




```
income_eval %>% ggplot(aes(race, fill = income)) + geom_bar(position = "fill")
```

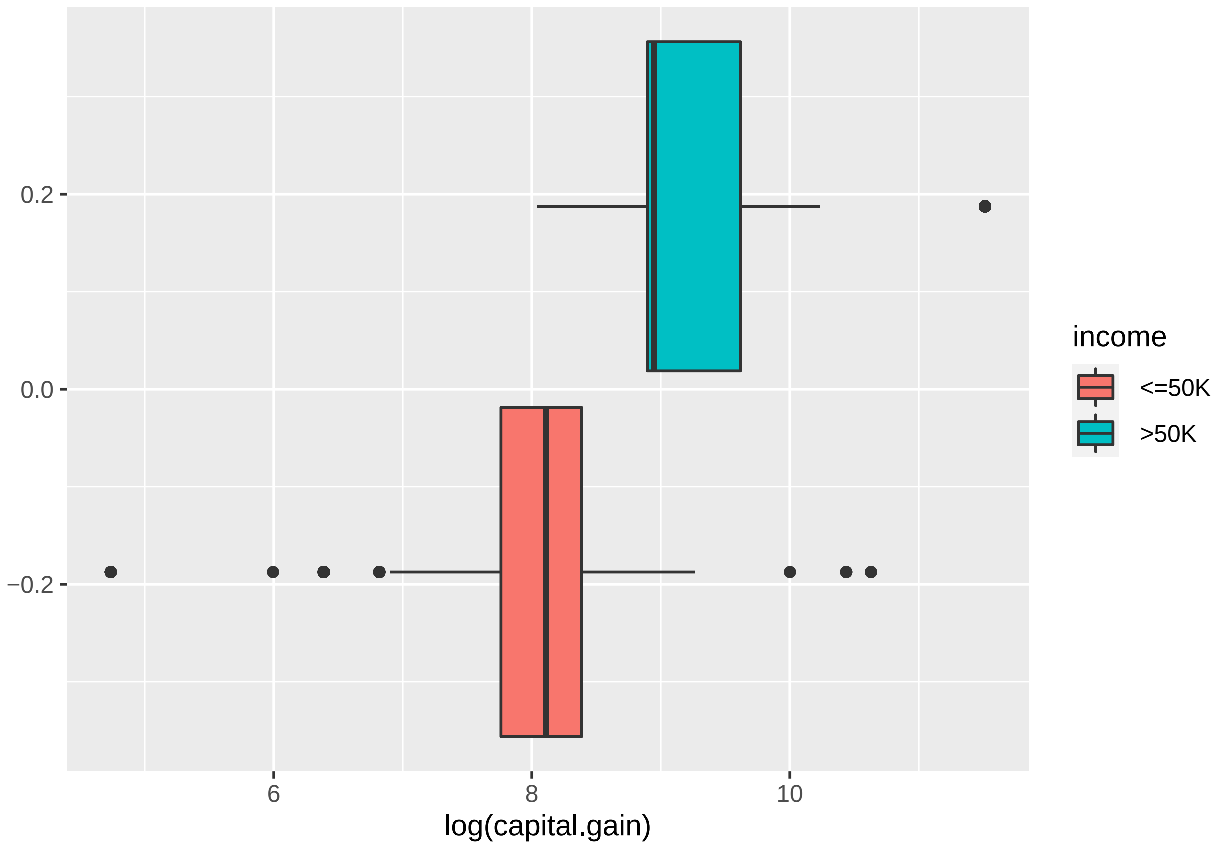



```
income_eval %>% ggplot(aes(sex, fill = income)) + geom_bar(position = "fill")
```



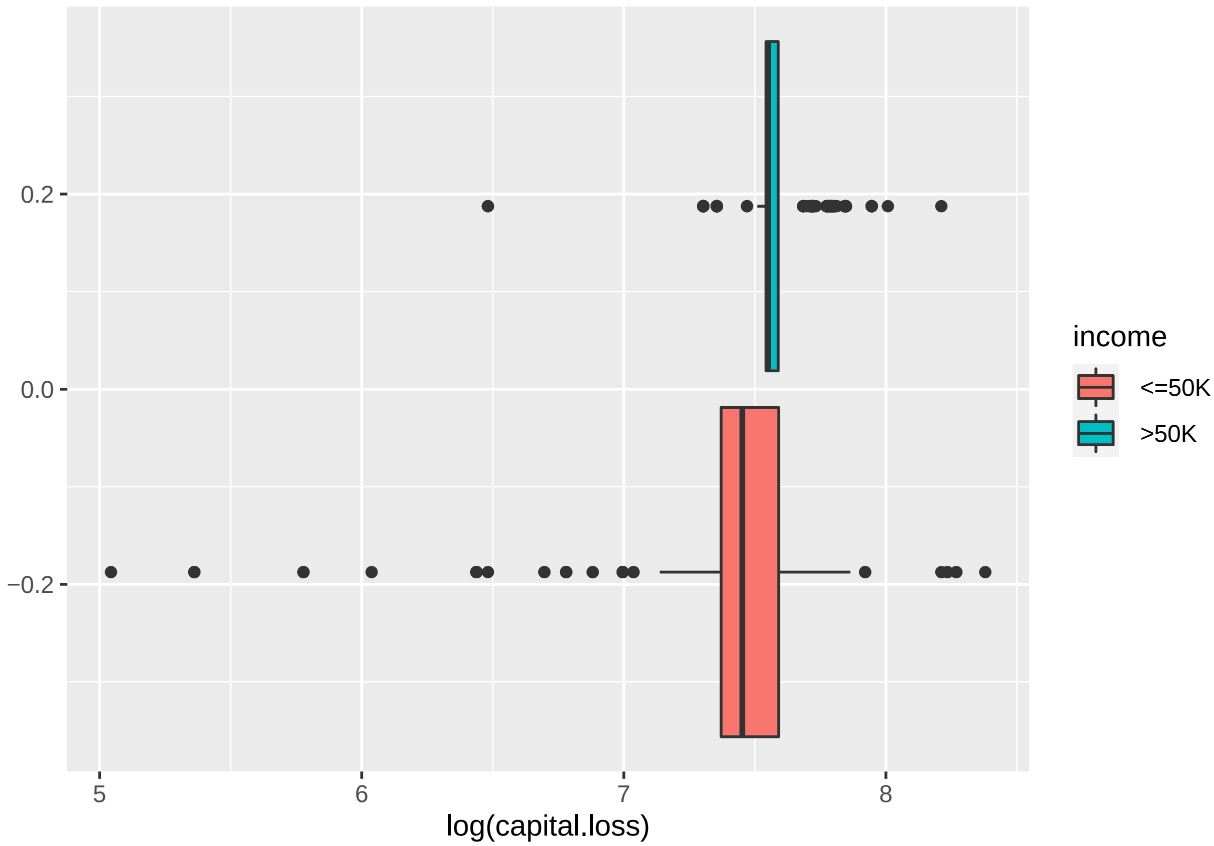

```
income_eval %>% ggplot(aes(log(capital.gain), fill = income)) + geom_boxplot() # log
```

```
## Warning: Removed 27624 rows containing non-finite values (stat_boxplot).
```




```
income_eval %>% ggplot(aes(log(capital.loss), fill = income)) + geom_boxplot() # log
```

```
## Warning: Removed 28735 rows containing non-finite values (stat_boxplot).
```



Selecting a model

(Using an approach suggested at: http://www.sthda.com/english/articles/36-classification-methods-essentials/150-stepwise-logistic-regression-essentials-in-r/#loading-required-r-packages)

```
set.seed(1)

# Fit a logistic model
full_logit <- glm(income ~., data = train_data, family = binomial(link="logit"))
```

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

```
# summary(full_logit)

# Make predictions for the full model
full_logit_pred <- predict(full_logit, newdata=test_data, "response")
full_logit_predicted_classes <- as.factor(ifelse(full_logit_pred > 0.5, " >50K", " <=50K"))

# Let's make a table
full_logit_table <- table(test_data$income, full_logit_predicted_classes)
caret::confusionMatrix(full_logit_table)
```

```
## Confusion Matrix and Statistics
```

```
##
##          full_logit_predicted_classes
##             <=50K  >50K
##     <=50K  10542   785
##     >50K    1509  2245
##
##                Accuracy : 0.8479
##                  95% CI : (0.8421, 0.8536)
##     No Information Rate : 0.7991
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5652
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8748
##             Specificity : 0.7409
##          Pos Pred Value : 0.9307
##          Neg Pred Value : 0.5980
##              Prevalence : 0.7991
##          Detection Rate : 0.6990
##    Detection Prevalence : 0.7511
##       Balanced Accuracy : 0.8079
##
##        'Positive' Class :  <=50K
##
```

```
set.seed(1)

# Let's do some stepwise variable selection to see if it'll improve our model fit
step_model <- full_logit %>% stepAIC(trace = FALSE)
```

```
## 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

## 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

## 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

## 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
```

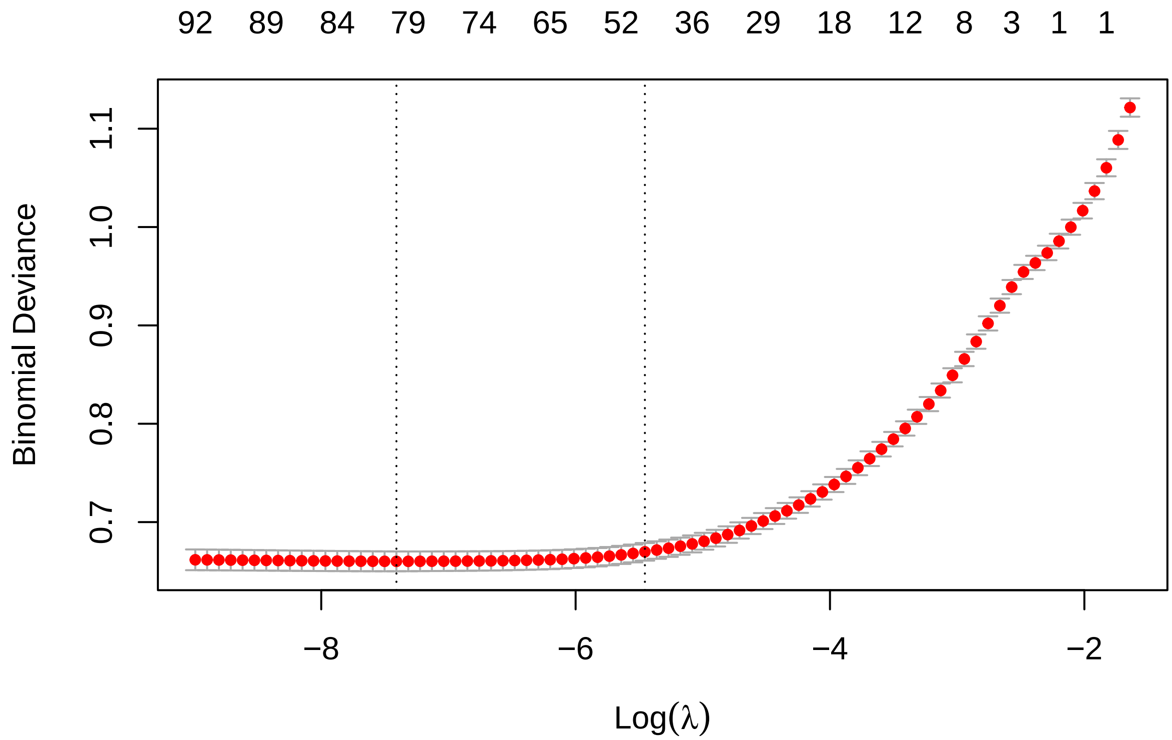
```r
# summary(step_model)

# Make predictions for the full model
stepwise_logit_pred <- predict(step_model, newdata=test_data, "response")
stepwise_logit_predicted_classes <- as.factor(ifelse(stepwise_logit_pred > 0.5, " >50K", " <=50

# Let's make a table
stepwise_logit_table <- table(test_data$income, stepwise_logit_predicted_classes)
caret::confusionMatrix(stepwise_logit_table)
```

```
## Confusion Matrix and Statistics
##
##          stepwise_logit_predicted_classes
##             <=50K  >50K
```

```
##    <=50K  10538   789
##    >50K    1520  2234
##
##                Accuracy : 0.8469
##                  95% CI : (0.841, 0.8526)
##     No Information Rate : 0.7995
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.562
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8739
##             Specificity : 0.7390
##          Pos Pred Value : 0.9303
##          Neg Pred Value : 0.5951
##              Prevalence : 0.7995
##          Detection Rate : 0.6988
##    Detection Prevalence : 0.7511
##       Balanced Accuracy : 0.8065
##
##        'Positive' Class :  <=50K
##
```

One more attempt: (Using an approach found at: http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/#quick-start-r-code)

```
set.seed(1)

# Dummy code categorical predictor variables
x <- model.matrix(income~., train_data)[,-1]
# Convert the outcome (class) to a numerical variable
y <- ifelse(train_data$income == " >50K", 1, 0)

cv_lasso <- cv.glmnet(x, y, alpha = 1, family = "binomial")
plot(cv_lasso)
```




```
cv_lasso$lambda.min
```

```
## [1] 0.0006058396
```

```
coef(cv_lasso, cv_lasso$lambda.min)
```

```
## 96 x 1 sparse Matrix of class "dgCMatrix"
##                                                     s1
## (Intercept)                              -5.843387e+00
## age                                       2.046929e-02
## workclass Local-gov                      -2.839420e-01
## workclass Private                        -1.994080e-01
## workclass Self-emp-inc                    .
## workclass Self-emp-not-inc               -5.520170e-01
## workclass State-gov                      -4.405066e-01
## workclass Without-pay                    -2.293502e+00
## fnlwgt                                    5.974925e-07
## education 11th                           -2.182231e-01
## education 12th                            .
## education 1st-4th                        -2.088376e-01
## education 5th-6th                        -7.741961e-01
## education 7th-8th                        -7.350295e-01
## education 9th                            -1.817369e-01
## education Assoc-acdm                      7.939525e-01
## education Assoc-voc                       7.851629e-01
```

```
## education Bachelors                    1.319636e+00
## education Doctorate                    2.445938e+00
## education HS-grad                      1.710250e-01
## education Masters                      1.660242e+00
## education Preschool                   -1.873400e+00
## education Prof-school                  2.342361e+00
## education Some-college                 4.742579e-01
## marital.status Married-AF-spouse       1.945019e+00
## marital.status Married-civ-spouse      1.803006e+00
## marital.status Married-spouse-absent   5.786729e-02
## marital.status Never-married          -4.398480e-01
## marital.status Separated              -2.664608e-02
## marital.status Widowed                 .
## occupation Armed-Forces                .
## occupation Craft-repair               -1.224825e-01
## occupation Exec-managerial             7.367194e-01
## occupation Farming-fishing            -1.043207e+00
## occupation Handlers-cleaners          -6.468970e-01
## occupation Machine-op-inspct          -3.549528e-01
## occupation Other-service              -1.017649e+00
## occupation Priv-house-serv            -2.423314e+00
## occupation Prof-specialty              4.384848e-01
## occupation Protective-serv             4.235424e-01
## occupation Sales                       1.445420e-01
## occupation Tech-support                5.601852e-01
## occupation Transport-moving           -1.459323e-01
## relationship Not-in-family             7.519782e-02
## relationship Other-relative           -4.263746e-01
## relationship Own-child                -1.079810e+00
## relationship Unmarried                 .
## relationship Wife                      1.148719e+00
## race Asian-Pac-Islander                3.443742e-01
## race Black                             .
## race Other                            -4.802982e-01
## race White                             1.427880e-01
## sex Male                               7.451073e-01
## capital.gain                           3.011176e-04
## capital.loss                           7.144290e-04
## hours.per.week                         2.757775e-02
## native.country Canada                  .
## native.country China                  -7.548951e-01
## native.country Columbia               -1.873713e+00
## native.country Cuba                    2.245941e-01
## native.country Dominican-Republic     -2.447828e+00
## native.country Ecuador                 1.609261e-01
## native.country El-Salvador            -7.712138e-01
## native.country England                 .
## native.country France                  6.560091e-02
```

```
## native.country Germany                     3.113600e-01
## native.country Greece                     -2.120241e+00
## native.country Guatemala                   .
## native.country Haiti                       6.994129e-01
## native.country Holand-Netherlands          .
## native.country Honduras                    .
## native.country Hong                       -7.744734e-01
## native.country Hungary                    -2.318649e+00
## native.country India                      -9.529850e-01
## native.country Iran                       -2.068242e-02
## native.country Ireland                     1.839410e-01
## native.country Italy                       1.215165e-02
## native.country Jamaica                     3.845879e-02
## native.country Japan                      -6.028883e-01
## native.country Laos                        .
## native.country Mexico                     -3.248823e-01
## native.country Nicaragua                   .
## native.country Outlying-US(Guam-USVI-etc) -1.999741e+00
## native.country Peru                       -5.790447e-02
## native.country Philippines                 3.237490e-01
## native.country Poland                      .
## native.country Portugal                   -6.719809e-01
## native.country Puerto-Rico                -1.251040e-01
## native.country Scotland                    .
## native.country South                      -2.228600e+00
## native.country Taiwan                      1.366738e-01
## native.country Thailand                   -8.957461e-01
## native.country Trinadad&Tobago            -5.152594e-01
## native.country United-States               .
## native.country Vietnam                    -9.047098e-01
## native.country Yugoslavia                  4.833533e-01
```

```
# Final model with lambda.min
lasso_model <- glmnet(x, y, alpha = 1, family = "binomial",
                      lambda = cv_lasso$lambda_min)
# Make prediction on test data
x_test <- model.matrix(income ~., test_data)[,-1]
probabilities <- lasso_model %>% predict(newx = x_test)
lasoo_logistic_predicted_classes <- ifelse(probabilities > 0.5, " >50K", " <=50K")
# Model accuracy
mean(lasoo_logistic_predicted_classes == test_data$income)
```

```
## [1] 0.8130238
```

Of all the models tested, the full logistic regression is the most accurate.

Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] glmnet_4.1-3    Matrix_1.4-0    caret_6.0-90    lattice_0.20-45
##  [5] nnet_7.3-17     MASS_7.3-55     forcats_0.5.1   stringr_1.4.0
##  [9] dplyr_1.0.7     purrr_0.3.4     readr_2.1.1     tidyr_1.1.4
## [13] tibble_3.1.6    ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-155         fs_1.5.2             lubridate_1.8.0
##  [4] httr_1.4.2           tools_4.1.2          backports_1.4.1
##  [7] utf8_1.2.2           R6_2.5.1             rpart_4.1.16
## [10] DBI_1.1.2            colorspace_2.0-2     withr_2.4.3
## [13] tidyselect_1.1.1     compiler_4.1.2       cli_3.1.1
## [16] rvest_1.0.2          formatR_1.11         xml2_1.3.3
## [19] labeling_0.4.2       scales_1.1.1         proxy_0.4-26
## [22] digest_0.6.29        rmarkdown_2.11       pkgconfig_2.0.3
## [25] htmltools_0.5.2      parallelly_1.30.0    highr_0.9
## [28] dbplyr_2.1.1         fastmap_1.1.0        rlang_1.0.0
## [31] readxl_1.3.1         rstudioapi_0.13      farver_2.1.0
## [34] shape_1.4.6          generics_0.1.1       jsonlite_1.7.3
## [37] ModelMetrics_1.2.2.2 magrittr_2.0.2       Rcpp_1.0.8
## [40] munsell_0.5.0        fansi_1.0.2          lifecycle_1.0.1
## [43] stringi_1.7.6        pROC_1.18.0          yaml_2.2.2
## [46] plyr_1.8.6           recipes_0.1.17       grid_4.1.2
## [49] parallel_4.1.2       listenv_0.8.0        crayon_1.4.2
## [52] haven_2.4.3          splines_4.1.2        hms_1.1.1
## [55] knitr_1.37           pillar_1.6.5         future.apply_1.8.1
## [58] reshape2_1.4.4       codetools_0.2-18     stats4_4.1.2
## [61] reprex_2.0.1         glue_1.6.1           evaluate_0.14
```

```
## [64] data.table_1.14.2    modelr_0.1.8         vctrs_0.3.8
## [67] tzdb_0.2.0           foreach_1.5.1        cellranger_1.1.0
## [70] gtable_0.3.0         future_1.23.0        assertthat_0.2.1
## [73] xfun_0.29            gower_0.2.2          prodlim_2019.11.13
## [76] broom_0.7.12         e1071_1.7-9          class_7.3-20
## [79] survival_3.2-13      timeDate_3043.102    iterators_1.0.13
## [82] lava_1.6.10          globals_0.14.0       ellipsis_0.3.2
## [85] ipred_0.9-12
```