

Final Exam Part 2

Summer Semester 2021

Instructions

This R Markdown file includes the questions, the empty code chunk sections for your code, and the text blocks for your responses. Answer the questions below by completing this R Markdown file. You must answer the questions using this file. You may make slight adjustments to get the file to knit/convert but otherwise keep the formatting the same. Once you've finished answering the questions, submit your responses in a single knitted **HTML file**.

There are 21 questions divided among 8 sections. The number of points for each question is provided. Partial credit may be given if your code is correct but your conclusion is incorrect or vice versa.

Next Steps:

1. Save this .Rmd file in your R working directory - the same directory where you will download the “white_wine_quality.csv” and “brooklyn_bridge_bike_counts.csv” data files into. Having all files in the same directory will help in reading the .csv files.
2. Read the question and create the R code necessary within the code chunk section immediately below each question. Knitting this file will generate the output and insert it into the section below the code chunk.
3. Type your answer to the questions in the text block provided immediately after the question prompt.
4. Once you've finished answering all questions, knit/convert this file and submit the knitted file as **HTML** on Canvas.

Example Question Format:

(8a) This will be the exam question - each question is already copied from Canvas and inserted into individual text blocks below, *you do not need to copy/paste the questions from the online Canvas exam*.

Example code chunk area. Enter your code below the comment and between the `{{r}}` and `

Response to question (8a) This is the section where you type your written answer to the question. Depending on the question asked, your typed response may be a number, a list of variables, a few sentences, or a combination of these elements.

** Ready? Let's begin. We wish you the best of luck! **

Data Sets Background

For this exam, you will be using two data sets.

*The first data set is “brooklyn_bridge_bike_counts.csv”. You will use this data set to build a model which predicts the number of bikes crossing the Brooklyn Bridge on a given day based on the following characteristics.

1. *month*: January, February, March, ... (categorical)
2. *day*: Sunday, Monday, Tuesday, ... (categorical)
3. *high_temp*: high temperature for the day in fahrenheit (numeric)

4. *low_temp*: low temperature for the day in fahrenheit (numeric)
5. *precipitation*: inches of precipitation for the day (numeric)

The response variable is *bikes* (numeric), representing the number of bikes crossing the Brooklyn Bridge on a given day.

*The second data set is “white_wine_quality.csv”. You will use this data set to build a model which predicts whether a type of white wine is good quality or bad quality based on the following characteristics.

1. *fixed_acidity*: grams of tartaric acid per cubic decimeter (numeric)
2. *volatile_acidity*: grams of acetic acid per cubic decimeter (numeric)
3. *citric_acid*: grams of citric acid per cubic decimeter (numeric)
4. *residual_sugar*: grams of residual sugar per cubic decimeter (numeric)
5. *chlorides*: grams of sodium chloride per cubic decimeter (numeric)
6. *free_sulfur_dioxide*: milligrams of free sulfur dioxide per cubic decimeter (numeric)
7. *total_sulfur_dioxide*: milligrams of total sulfur dioxide per cubic decimeter (numeric)
8. *density*: grams per cubic decimeter (numeric)
9. *ph*: measure of acidity or basicity (scale from 0 to 14) (numeric)
10. *sulphates*: grams of potassium sulphate per cubic decimeter (numeric)
11. *alcohol*: percent alcohol by volume (numeric)

The response variable is *quality*: 1 = good quality and 0 = bad quality (binary).

Read Data

Read the data and answer the questions below. Assume a significance level of 0.05 for hypothesis tests unless stated otherwise.

```
# Load relevant libraries (add here if needed)
library(car)

## Loading required package: carData
library(aod)
library(bestglm)

## Loading required package: leaps
library(boot)

##
## Attaching package: 'boot'
## The following object is masked from 'package:car':
##   logit
library(corrplot)

## corrplot 0.90 loaded
library(caret)

## Loading required package: lattice
```

```

## 
## Attaching package: 'lattice'
## The following object is masked from 'package:boot':
##
##     melanoma

## Loading required package: ggplot2
library(glmnet)

## Loading required package: Matrix
## Loaded glmnet 4.1-2
# Ensure that the sampling type is correct
RNGkind(sample.kind="Rejection")

# Set seed (please do not change for consistency of the results)
set.seed(0)

##### Read and process the data: Bike data #####
# Read data
bike_data_full = read.csv("brooklyn_bridge_bike_counts.csv", header=TRUE)

# Convert month and day columns to categorical variables
bike_data_full$month <- as.factor(bike_data_full$month)
bike_data_full$day <- as.factor(bike_data_full$day)

# Split data for training and testing
bike_test_rows = sample(nrow(bike_data_full), 0.2*nrow(bike_data_full))
bike_data_test = bike_data_full[bike_test_rows, ]
bike_data_train = bike_data_full[-bike_test_rows, ]

##### Read and process the data: Wine data #####
# Read data
wine_data_full = read.csv("white_wine_quality.csv", header=TRUE)

# Convert response variable to binary
wine_data_full$quality <- ifelse(wine_data_full$quality > 5, 1, 0)

# Split data for training and testing
wine_test_rows = sample(nrow(wine_data_full), 0.2*nrow(wine_data_full))
wine_data_test = wine_data_full[wine_test_rows, ]
wine_data_train = wine_data_full[-wine_test_rows, ]

```

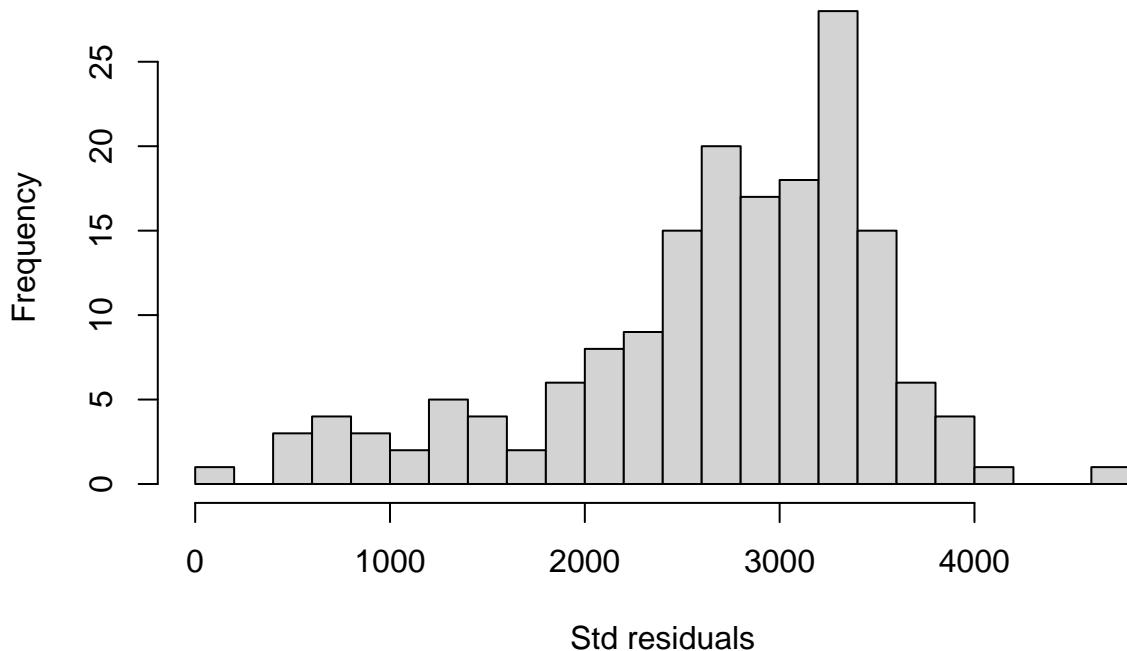
Note: You will be using the bike data set first (Questions 1,2,3, and 4), followed by the wine data set (Questions 5,6,7, and 8).

Note: Don't change the data types of the variables. The month and day columns in the bike data set have already been converted to categorical variables. The quality column in the wine data set has already been converted to a binary variable.

Question 1: Bike Data - Exploratory Analysis

(1a) 2 pts - Using `bike_data_train`, create a histogram of the variable `bikes`. Based on this plot, what generalized linear regression model(s) discussed in this course could be used to model this response variable? Explain.

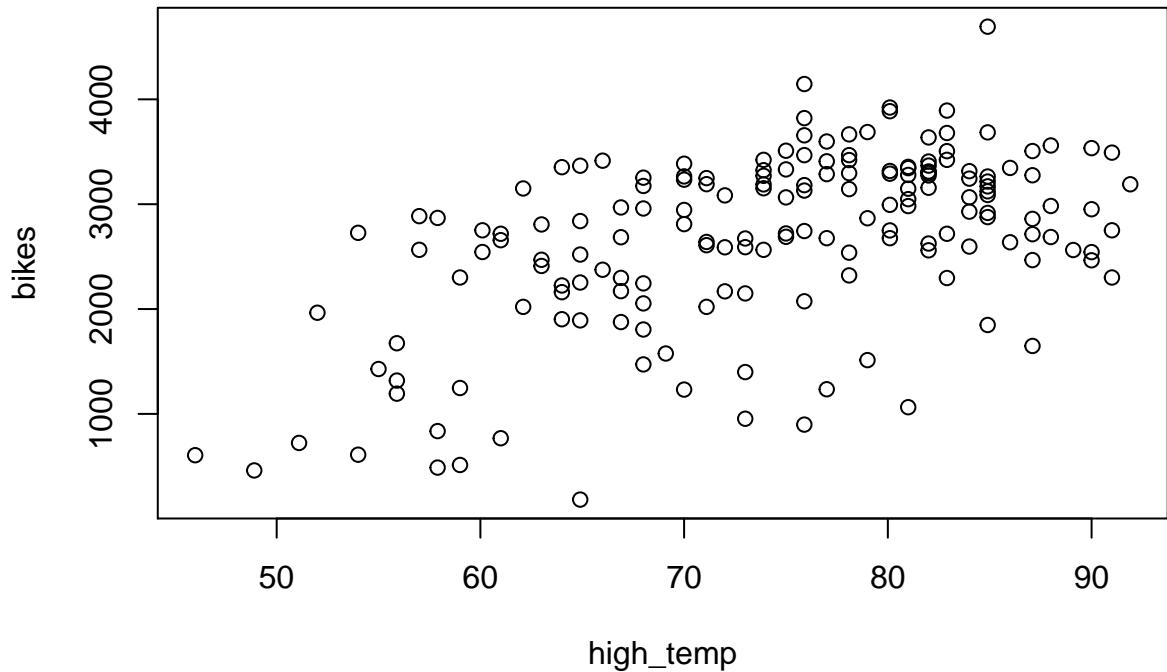
```
# Create histogram  
hist(bike_data_train$bikes, 30, xlab="Std residuals", main="")
```



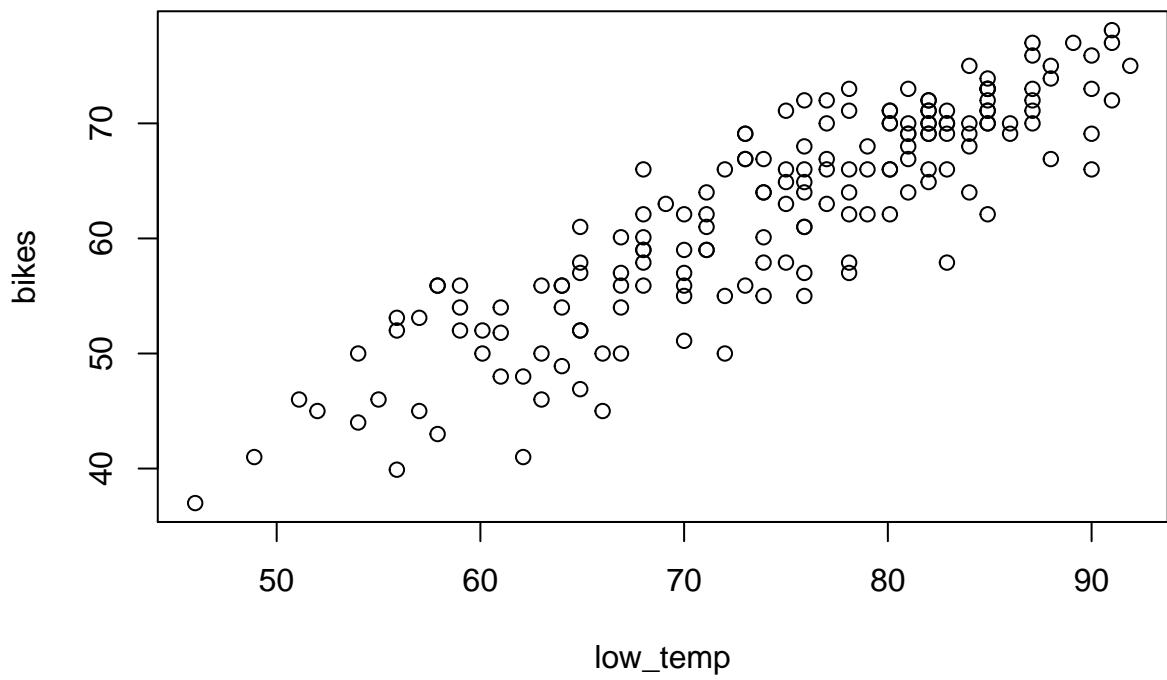
Response to question (1a) Based on this it would make sense to use Poisson regression

(1b) 2 pts - Using `bike_data_train`, create a scatterplot of `bikes` versus each numeric predicting variable (`high_temp`, `low_temp`, and `precipitation`) (3 scatterplots total). Do these variables appear useful in predicting the number of bikes crossing the Brooklyn Bridge on a given day? Include your reasoning.

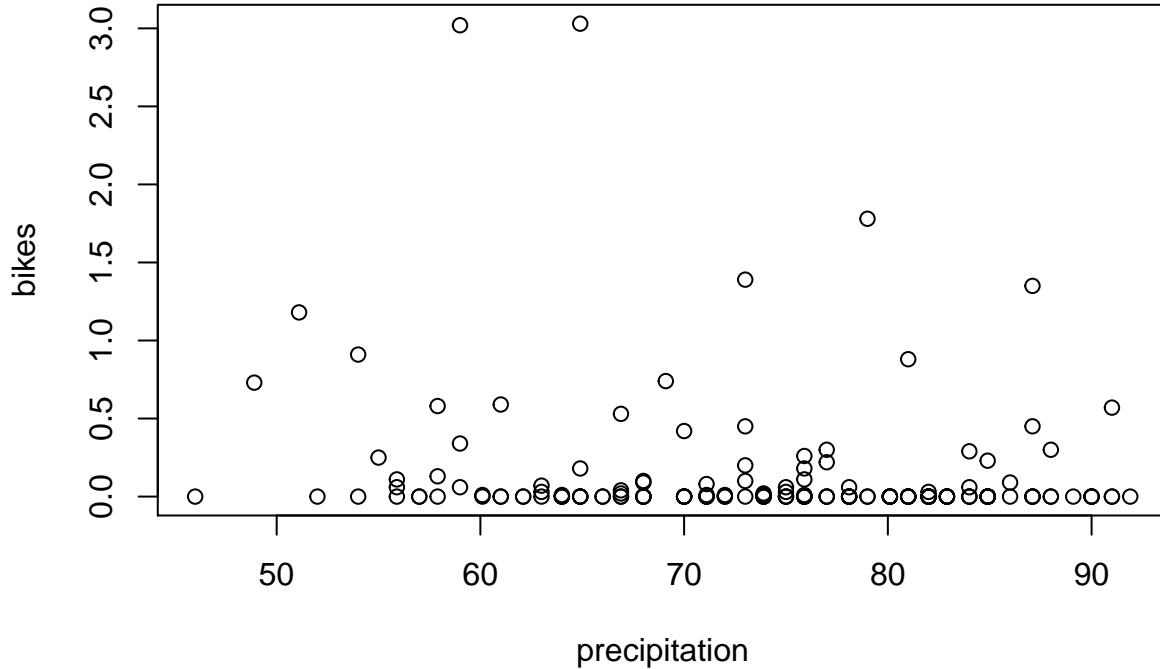
```
# Create scatterplots  
plot(bike_data_train$high_temp, bike_data_train$bikes, xlab="high_temp", ylab = "bikes")
```



```
plot(bike_data_train$high_temp, bike_data_train$bikes, xlab="high_temp", ylab = "bikes")
```



```
plot(bike_data_train$low_temp, bike_data_train$bikes, xlab="low_temp", ylab = "bikes")
```



Response to question (1b) The variables are good to use... High temperature shows a general linear trend, the low_temp shows a good linear trend and for precipitaion .. looks like more bikes cross the bridge when the precipitaion is low.. towards 0

Question 2: Bike Data - Full Model

(2a) 2 pts - Using `bike_data_train`, fit a poisson regression model with `bikes` as the response variable and all other variables as predicting variables. Include an intercept. Call it `model1`. Display the summary table for the model.

```
# Fit model and display summary
attach(bike_data_train)
model1 = glm(bikes ~ ., data = bike_data_train, family = poisson)
summary(model1)
```

```
##
## Call:
## glm(formula = bikes ~ ., family = poisson, data = bike_data_train)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -33.648    -5.050     0.744     5.464    25.924
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 7.0658277  0.0149756 471.824 < 2e-16 ***
## monthAugust 0.1836782  0.0065828  27.903 < 2e-16 ***
## monthJuly   0.0923125  0.0072658  12.705 < 2e-16 ***
## monthJune   0.0962910  0.0067864  14.189 < 2e-16 ***
## monthMay    0.0297962  0.0060541   4.922 8.58e-07 ***
## monthOctober 0.0639270  0.0059393  10.763 < 2e-16 ***
## monthSeptember 0.0889397  0.0066189  13.437 < 2e-16 ***
## dayMonday   0.0048952  0.0057858   0.846  0.3975
```

```

## daySaturday -0.0733634 0.0060004 -12.226 < 2e-16 ***
## daySunday -0.1736796 0.0061211 -28.374 < 2e-16 ***
## dayThursday 0.0647525 0.0056766 11.407 < 2e-16 ***
## dayTuesday 0.0560493 0.0056884 9.853 < 2e-16 ***
## dayWednesday 0.0121695 0.0058472 2.081 0.0374 *
## high_temp 0.0260836 0.0003404 76.624 < 2e-16 ***
## low_temp -0.0181254 0.0004193 -43.232 < 2e-16 ***
## precipitation -0.7380845 0.0075536 -97.713 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 53677 on 171 degrees of freedom
## Residual deviance: 17918 on 156 degrees of freedom
## AIC: 19610
##
## Number of Fisher Scoring iterations: 4

```

(2b) 2 pts - Provide a meaningful interpretation of the estimated regression coefficient for *precipitation* for *model1*. Note: Don't just list the value of the estimated coefficient.

Response to question (2b) for every unit of decrease in precipitation, there is a decrease in log rate by -0.7380845 of bikes crossing.

(2c) 3 pts - Perform a test for the overall regression on *model1*. Is *model1* significant overall using an alpha of 0.05? Why/Why not?

```

# Perform test for overall regression
1-pchisq((model1>null.dev - model1$deviance), (model1$df.null - model1$df.resid))

## [1] 0

```

Response to question (2c) the p-value of the model based on Chi Sq is 0.. so the model is statistically significant

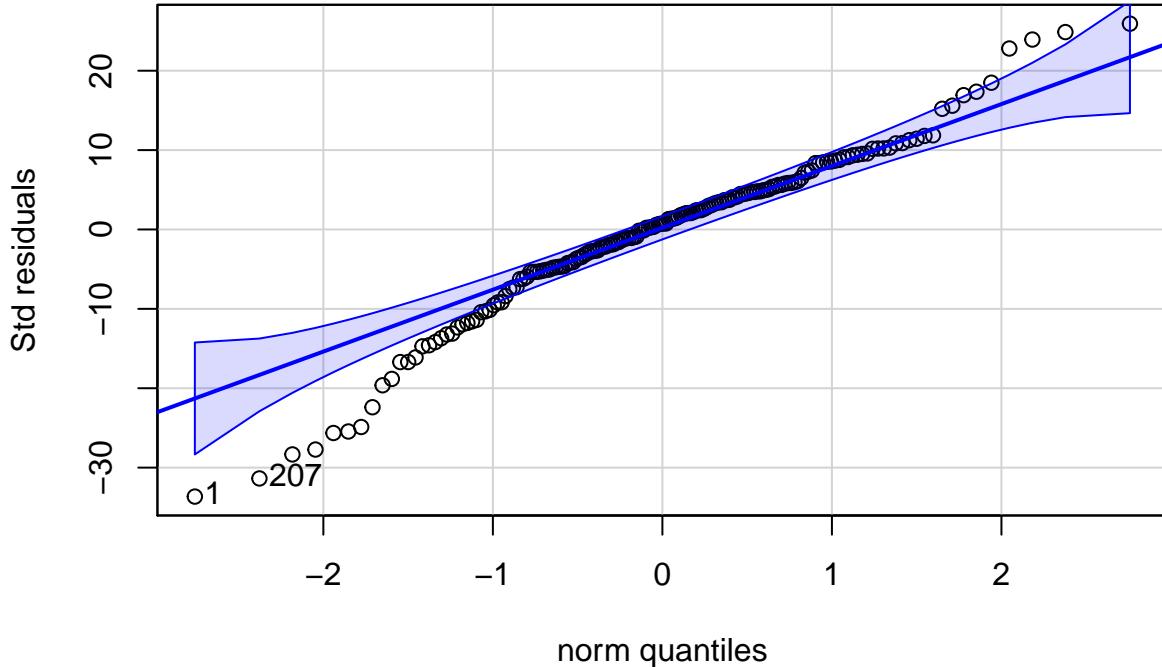
Question 3: Bike Data - Goodness of Fit

(3a) 3 pts - Evaluate whether the deviance residuals are approximately normally distributed by producing a QQ plot and histogram of the deviance residuals. Based on these plots, what assessment can you make about the goodness of fit of *model1*? Hint: Use *qqPlot()* from the *car* package which adds a confidence band to the normal QQ-plot by default.

```

# Create QQ plot and histogram
res = resid(model1,type="deviance")
car::qqPlot(res, ylab="Std residuals")

```



```
##    1 207
##    1 165
```

Response to question (3a) the QQ plot shows a long left tail... this shows that the response variable is mostly normally but there are outliers and points outside the 95% confidence band.

(3b) 3 pts - Perform a goodness-of-fit statistical test for *model1* using the deviance residuals and an alpha of 0.05. Provide the null and alternative hypotheses, test statistic, p-value, and conclusion in the context of the problem.

```
# Perform GOF test
model1$deviance

## [1] 17917.92
cat("Deviance residuals test p-value:",
  1-pchisq(model1$deviance, model1$df.residual), end="\n")
```

Deviance residuals test p-value: 0

Response to question (3b) H_0 : model 1 is a good fit H_a : Model 1 is NOT a good fit **Deviance test statistic:** Deviance = 17917.92 **p-value:** = 1 **Conclusion:** Since p-value is 0... Reject Null Hypothesis of good fit.(thus model1 NOT a good fit).

(3c) 3 pts - Why might a poisson regression model not be a good fit? Provide two reasons. How can you try to improve the fit in each situation? **Do not apply the recommendations.**

Response to question (3c)

Reason 1: The residuals are not normally distributed..

How can you try to improve the fit? We might try to do a Box cox transformation on the response variable.

Reason 2: There might be a lot of variables and they might be correlated

How can you try to improve the fit? We should run variable selections

Question 4: Bike Data - Prediction

(4a) 2 pts - Predict *bikes* for the test set (*bike_data_test*) using *model1*. Display the first six predicted values.

```
# Obtain predictions
pred.test = predict.glm(model1,bike_data_test,type="response")

# Display the first six predicted values
head(bike_data_test, 6)
```

```
##          month      day high_temp low_temp precipitation bikes
## 142     August   Sunday      81.0      70.0        0.00  2822
## 68      June Wednesday     66.9      54.0        0.00  3211
## 167    September Thursday     81.0      70.0        0.02  3013
## 129     August   Monday      71.1      64.9        0.76   804
## 162    September Saturday     69.1      55.0        0.00  2609
## 43      May   Saturday      51.1      45.0        1.31   151
```

(4b) 2.5 pts - Calculate and display the mean squared prediction error (MSPE) for *model1*. List one limitation of using this metric to evaluate prediction accuracy.

```
# Calculate MSPE
# mse.model1 = mean((pred.test - bike_data_test$bikes ) ^ 2)
mse.model1 = mean((pred.test-bike_data_test$bikes)^2)
mse.model1

## [1] 508712.6
```

Response to question (4b) The data Metrics depends on scale and sensitive to outliers

(4c) 1 pt - Refit *model1* on *bike_data_full*, and call it *model2*. Display the summary table for the model.

```
# Fit model and display summary
model2 = glm(bikes ~ ., data = bike_data_full, family = poisson)
summary(model2)
```

```
##
## Call:
## glm(formula = bikes ~ ., family = poisson, data = bike_data_full)
##
## Deviance Residuals:
##       Min      1Q      Median      3Q      Max
## -33.974   -6.642    0.447    5.524   38.414
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 7.1352598  0.0128218 556.492 < 2e-16 ***
## monthAugust 0.2175953  0.0061188  35.562 < 2e-16 ***
## monthJuly   0.1293258  0.0066699  19.389 < 2e-16 ***
## monthJune   0.1458536  0.0060765  24.003 < 2e-16 ***
## monthMay    0.0928497  0.0054400  17.068 < 2e-16 ***
## monthOctober 0.1046296  0.0054089  19.344 < 2e-16 ***
## monthSeptember 0.1506018  0.0058186  25.883 < 2e-16 ***
## dayMonday   0.0198863  0.0050353   3.949 7.83e-05 ***
## daySaturday -0.0526590  0.0051712 -10.183 < 2e-16 ***
## daySunday   -0.0951229  0.0051070 -18.626 < 2e-16 ***
## dayThursday  0.0749459  0.0049535  15.130 < 2e-16 ***
## dayTuesday   0.0658885  0.0049771  13.238 < 2e-16 ***
## dayWednesday 0.0559215  0.0049268  11.351 < 2e-16 ***
```

```

## high_temp      0.0251443  0.0003048   82.495 < 2e-16 ***
## low_temp     -0.0191551  0.0003703  -51.723 < 2e-16 ***
## precipitation -0.7643938  0.0068964 -110.840 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 70021  on 213  degrees of freedom
## Residual deviance: 26681  on 198  degrees of freedom
## AIC: 28777
##
## Number of Fisher Scoring iterations: 4

```

(4c.1) 3 pts - Estimate the 10-fold and leave-one-out cross validation mean prediction squared error (MSPE) for *model2*. Display these values. Hint: cv.glm() from the boot package uses MSPE as the default cost function.

```

# Perform cross validation and get and display MSPEs
model2.10.fold = cv.glm(data = bike_data_full, model2, K=10)
(model2.10.fold)$delta

## [1] 324896.8 321898.5

#leave one out cross-validation
model2.loocv = cv.glm(bike_data_full, model2, K=nrow(bike_data_full))
(model2.loocv)$delta

## [1] 318056.1 317937.1

```

(4c.2) 1 pt - How do these two MSPEs compare to the *model1* MSPE from 4b?

Response to question (4c.2) *model1* MSPE = 508712.6, where as MSPE for k-fold(10) = 318378.3 and for LOOCV = 318056.1. it looks like there is a reduction in the MSPE

Question 5: Wine Data - Full Model

(5a) 2 pts - Using *wine_data_train*, fit a logistic regression model with *quality* as the response variable and all other variables as predicting variables. Include an intercept. Call it *model3*. Display the summary table for the model.

```

# Fit model and display summary
model3 = glm(quality ~ ., data = wine_data_train, family=binomial)
summary(model3)

##
## Call:
## glm(formula = quality ~ ., family = binomial, data = wine_data_train)
##
## Deviance Residuals:
##       Min        1Q        Median         3Q        Max
## -3.1115   -0.8861    0.4350    0.7909    2.7362
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.012e+02 7.631e+01  2.636 0.008378 **
## fixed_acidity 3.431e-02 7.862e-02  0.436 0.662555
## volatile_acidity -6.673e+00 4.677e-01 -14.268 < 2e-16 ***

```

```

## citric_acid      2.042e-02  3.385e-01   0.060  0.951907
## residual_sugar  1.412e-01  2.929e-02   4.821  1.43e-06 ***
## chlorides       -8.775e-01  1.921e+00  -0.457  0.647800
## free_sulfur_dioxide 1.223e-02  3.182e-03   3.844  0.000121 ***
## total_sulfur_dioxide -1.682e-03  1.372e-03  -1.227  0.219950
## density         -2.130e+02  7.735e+01  -2.754  0.005891 **
## ph              8.833e-01  4.003e-01   2.206  0.027354 *
## sulphates       1.634e+00  4.037e-01   4.048  5.18e-05 ***
## alcohol          8.031e-01  1.019e-01   7.877  3.35e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 4974.6 on 3918 degrees of freedom
## Residual deviance: 3911.2 on 3907 degrees of freedom
## AIC: 3935.2
##
## Number of Fisher Scoring iterations: 5

```

(5b) 2 pts - Conduct a multicollinearity test on *model3*. Using a VIF threshold of 10, what can you conclude?

```

# Obtain VIF values
# VIF Threshold
cat("VIF Threshold:", max(10, 1/(1-summary(model2)$r.squared)), "\n")

```

```
## VIF Threshold: 10
```

```
# Calculate VIF
car::vif(model2)
```

```

##           GVIF Df GVIF^(1/(2*Df))
## month      2.134737  6     1.065235
## day        1.082777  6     1.006649
## high_temp  5.093211  1     2.256814
## low_temp   6.384587  1     2.526774
## precipitation 1.093709  1     1.045805

```

Response to question (5b) Using the VIF threshold of 10, there does not seem to be multicollinearity as all the predictors have the GVIF value < 10

(5c) 2 pts - Estimate the dispersion parameter for *model3*. Does overdispersion seem to be a problem in this model?

```

# Estimate dispersion parameter
model3$deviance/model3$df.res
```

```
## [1] 1.001078
```

Response to question (5c) No.. since the overdispersion is close to 1.. there seems to be no overdispersion

Question 6: Wine Data - Variable Selection

(6a) 3 pts - Using *wine_data_train*, conduct a complete search to find the submodel with the smallest BIC. Fit this model. Include an intercept. Call it *all_subsets_model*. Display the summary table for the model.
Note: Remember to set family to binomial.

```

# Conduct a complete search using BIC
library(bestglm)
```

```

all_subsets_model1 <- bestglm(wine_data_train, IC="BIC", family=binomial)

## Morgan-Tatar search since family is non-gaussian.

all_subsets_model1

## BIC
## BICq equivalent for q in (0.117048566987781, 0.622760460037739)
## Best Model:
##                               Estimate Std. Error   z value Pr(>|z|)
## (Intercept)           155.9449389 46.428275713  3.358835 7.827166e-04
## volatile_acidity     -6.8701140  0.451094343 -15.229883 2.239926e-52
## residual_sugar        0.1193562  0.018605017   6.415267 1.405763e-10
## free_sulfur_dioxide   0.0104005  0.002518183   4.130159 3.625123e-05
## density              -165.4028989 46.284164954 -3.573639 3.520542e-04
## sulphates            1.6332953  0.393319250   4.152594 3.287270e-05
## alcohol               0.8941581  0.071769397  12.458766 1.252947e-35

# Fit the model and display summary
all_subsets_model = glm(quality~volatile_acidity+residual_sugar+free_sulfur_dioxide+density+sulphates+alcohol, family=binomial, data=wine_data_train)
summary(all_subsets_model)

##
## Call:
## glm(formula = quality ~ volatile_acidity + residual_sugar + free_sulfur_dioxide +
##      density + sulphates + alcohol, family = binomial, data = wine_data_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q      Max
## -3.1389 -0.8986  0.4329  0.8028  2.5350
##
## Coefficients:
##                               Estimate Std. Error   z value Pr(>|z|)
## (Intercept)           1.559e+02  4.643e+01   3.359 0.000783 ***
## volatile_acidity     -6.870e+00  4.511e-01 -15.230 < 2e-16 ***
## residual_sugar        1.194e-01  1.861e-02   6.415 1.41e-10 ***
## free_sulfur_dioxide   1.040e-02  2.518e-03   4.130 3.63e-05 ***
## density              -1.654e+02  4.628e+01 -3.574 0.000352 ***
## sulphates            1.633e+00  3.933e-01   4.153 3.29e-05 ***
## alcohol               8.942e-01  7.177e-02  12.459 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 4974.6 on 3918 degrees of freedom
## Residual deviance: 3920.6 on 3912 degrees of freedom
## AIC: 3934.6
##
## Number of Fisher Scoring iterations: 5

```

(6a.1) 0.5 pts - Which variables are in your *all_subsets_model*?

Responses to question (6a.1) Variables selected: Variables selected - volatile_acidity, residual_sugar, free_sulfur_dioxide, density, sulphates, alcohol

(6a.2) 1 pt - What is the BIC of *all_subsets_model*?

```
# Calculate or extract BIC
AIC(all_subsets_model, k=log(nrow(wine_data_train)))
```

```
## [1] 3978.472
```

Responses to question (6a.2) BIC: 3978.472

(6b) 3 pts - Conduct backward stepwise regression on *wine_data_train* using AIC. Allow the minimum model to be a logistic model with *quality* as the response variable and only an intercept, and the full model to be *model3*. Call it *stepwise_model*. Display the summary table for the model. *Note: Remember to set family to binomial.*

```
# Conduct Backward stepwise regression using AIC and display model summary
```

```
m1 = glm(quality ~ 1, family=binomial, data = wine_data_train)
```

```
stepwise_model = step(model3, scope=list(lower=m1, upper=model3), direction="backward", data = wine_data_train)
summary(stepwise_model)
```

```
##
```

```
## Call:
```

```
## glm(formula = quality ~ volatile_acidity + residual_sugar + free_sulfur_dioxide +
##      density + ph + sulphates + alcohol, family = binomial, data = wine_data_train)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3.1395 -0.8922  0.4325  0.7984  2.6845
```

```
##
```

```
## Coefficients:
```

	Estimate	Std. Error	z	value	Pr(> z)
## (Intercept)	1.923e+02	4.908e+01	3.918	8.94e-05	***
## volatile_acidity	-6.843e+00	4.515e-01	-15.155	< 2e-16	***
## residual_sugar	1.371e-01	1.992e-02	6.884	5.82e-12	***
## free_sulfur_dioxide	9.780e-03	2.527e-03	3.871	0.000109	***
## density	-2.039e+02	4.917e+01	-4.147	3.37e-05	***
## ph	7.707e-01	2.868e-01	2.688	0.007191	**
## sulphates	1.552e+00	3.936e-01	3.943	8.03e-05	***
## alcohol	8.388e-01	7.518e-02	11.157	< 2e-16	***
## ---					
## Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '

```
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
```

```
## Null deviance: 4974.6 on 3918 degrees of freedom
```

```
## Residual deviance: 3913.3 on 3911 degrees of freedom
```

```
## AIC: 3929.3
```

```
##
```

```
## Number of Fisher Scoring iterations: 5
```

(6b.1) 0.5 pts - Which variables are in your *stepwise_model*?

Responses to question (6b.1) Variables selected: volatile_acidity, residual_sugar, free_sulfur_dioxide, density, ph, sulphates, alcohol

(6b.2) 0.5 pts - What is the AIC of *stepwise_model*?

```
# Calculate or extract AIC
```

```
AIC(stepwise_model)
```

```
## [1] 3929.286
```

Responses to question (6b.2) AIC: 3929.286

Question 7: Wine Data - Regularized Regression

(7a) Using `wine_data_train`, conduct ridge regression with `quality` as the binary response variable and all other variables in `wine_data_train` as the predicting variables.

(7a.1) 3 pts - Use 10-fold cross validation on the *misclassification error* to select the optimal lambda value. What optimal lambda value did you obtain? *Hint: Make sure to set type.measure = "class" in order to perform cross validation on the misclassification error. If needed, you can take a look at the help file by typing ?cv.glmnet.*

```
# Conduct cross validation and display optimal lambda
x.train <- model.matrix(quality ~ ., wine_data_train)[,-1]
y.train <- wine_data_train$quality

ridge.cv = cv.glmnet(x.train, y.train, alpha=0, nfolds = 10, type.measure = "class", family = "binomial")

ridge.cv$lambda.min

## [1] 0.01817872

# ridge = glmnet(x.train, y.train, alpha=0, nlambd=100)
# coef(ridge, s=ridge.cv$lambda.min)
```

Response to question (7a.1): Optimal lambda: 0.01817872

(7a.2) 1.5 pts - Fit a `glmnet` object with `nlambd` = 100. Call it `ridge_model`.

```
# Fit the model
ridge_model = glmnet(x.train, y.train, alpha=0, nlambd=100)
```

(7a.3) 1 pt - Display the estimated coefficients at the optimal lambda value.

```
# Display coefficients at optimal lambda
coef(ridge_model, s=ridge.cv$lambda.min)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##           s1
## (Intercept) 1.940253e+01
## fixed_acidity 3.678468e-03
## volatile_acidity -1.117358e+00
## citric_acid -2.101815e-02
## residual_sugar 1.656638e-02
## chlorides -6.613192e-01
## free_sulfur_dioxide 2.161658e-03
## total_sulfur_dioxide -4.628478e-04
## density -2.059766e+01
## ph 1.339803e-01
## sulphates 2.171643e-01
## alcohol 1.343697e-01
```

Question 8: Wine Data - Prediction

(8a) 6 pts - Using `model3`, `all_subsets_model`, `stepwise_model`, and `ridge_model`, give a binary classification to each of the rows in `wine_data_test`, with 1 indicating a good quality wine. Use 0.5 as your classification threshold.

```

model3.pred = predict(model3, newdata = wine_data_test)
predClass.model3 = ifelse(model3.pred > 0.5, 1, 0)

all_subsets_model.pred = predict(all_subsets_model, newdata = wine_data_test)
predClass.all_subsets_model = ifelse(all_subsets_model.pred > 0.5, 1, 0)

stepwise_model.pred = predict(stepwise_model, newdata = wine_data_test)
predClass.stepwise_model = ifelse(stepwise_model.pred > 0.5, 1, 0)

new_test <- model.matrix(quality ~ ., wine_data_test)[,-1]
# Obtain predicted probabilities for the test set
ridge.pred = predict(ridge_model, newx = new_test, s=ridge.cv$lambda.min)
predClass.ridge = ifelse(ridge.pred > 0.5, 1, 0)

```

(8b) 2 pts - For each model, display its accuracy. *Hint: Remember that accuracy is the proportion of all responses in the test set that are correctly classified.*

```

# Calculate accuracy for each model

# Give a binary classification to each of the rows in the test data
pred_metrics = function(modelName, actualClass, predClass) {
  conmat <- confusionMatrix(table(actualClass, predClass))
  cat(modelName, '      ', conmat$overall["Accuracy"], '\n')
}

##Full model
pred_metrics("model3",wine_data_test$quality, predClass.model3)

## model3      0.7344229
##Stepwise selection model
pred_metrics("all_subsets_model",wine_data_test$quality, predClass.all_subsets_model)

## all_subsets_model      0.7282942
##Lasso model
pred_metrics("stepwise_model",wine_data_test$quality, predClass.stepwise_model)

## stepwise_model      0.7385087
##Elastic Net model
pred_metrics("ridge_model",wine_data_test$quality, predClass.ridge)

## ridge_model      0.7262513

```

(8c) 1 pt - Based on 8b, which model performed the best?

Response to question (8c) According the accuracy.. stepwise_model model performed the best.. with accuracy of 0.7385087. But the accuracy of all the models are very close.

(8d) 1.5 pts - If you were to consider other metrics such as sensitivity or specificity, should sensitivity or specificity matter more in the context of this problem? Explain. *Note: Don't calculate these metrics. Hint: Remember that sensitivity is the proportion of all 1s in the test set that are correctly classified as 1s, while specificity is the proportion of all 0s in the test set that are correctly classified as 0s.*

Response to question (8d) In this case.. it is important to identify the quality of wine and since the Accuracy are so close, we can use Sensitivity.

This is the End of Final Exam Part 2

We hope you enjoyed the course - and we wish you the best in your future coursework!