Module 6 Assessment (Key)

Y. Samuel Wang

4/17/2024

Groundhog Day

Legend has it that Punxsutawney Phil, a groundhog from Punxsutawney, Pennsylvania is capable of predicting the severity of the weather. On Groundhog day each year (Feb 2), Phil rises from his burrow and if he sees his shadow, it means that it will be a long winter. If Phil doesn't see his own shadow, it means that there will be a early spring. Phil has appeared on The Oprah Winfrey Show and was immortalized by the 1993 movie "Groundhog Day" starring Bill Murray. But has Phil been fooling us this whole time, or is he the real deal? Let's take a look. We will consider Phil's predictions from 1900-2022 and for the purposes of this assignment, if the temperature in March is higher than average, we will consider that as an early spring, and if the temperature in March is lower than average we will consider that as a long winter.

```
# load data
phil_data <- read.csv("https://raw.githubusercontent.com/ysamwang/btry6020_sp22/main/lectureData/groundh
names(phil_data)
```

```
## [1] "Year"        "mar_avg"     "jan_avg"     "feb_avg"     "Prediction"
## [6] "Actual_warm"
```

In the data set, we have the following variables

- jan_avg : The average temperature in Pennsylvania in January
- feb_avg : The average temperature in Pennsylvania in February
- mar_avg : The average temperature in Pennsylvania in January
- Prediction : The outcome predicted by Punxsutawney Phil. "Winter" for long winter, "Spring" for early spring
- Actual_warm : The dependent variable of interest. It is a 1 if the temperature in March is higher than average, 0 if the temperature in March is lower than average

Question 1 (1 pt)

Fit a logistic regression model where the outcome is whether or not the temperature in March was higher than average (Actual_warm), and the only covariate we consider is Phil's prediction (Prediction).

```
groundhog_model <- glm(Actual_warm ~ Prediction, data = phil_data, family = "binomial")
summary(groundhog_model)
```

Answer to Question 1

```
##
## Call:
## glm(formula = Actual_warm ~ Prediction, family = "binomial",
##     data = phil_data)
##
```

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.5390     0.4756   1.133    0.257
## PredictionWinter   -0.6175     0.5152  -1.198    0.231
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 167.73  on 120  degrees of freedom
## Residual deviance: 166.25  on 119  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 170.25
##
## Number of Fisher Scoring iterations: 4
```

Question 2 (1 pt)

Give an interpretation for the estimated coefficient for Phil's prediction in the model above.

Answer to Question 2 We see that `R` has coded the Prediction variable as (Spring = 0, Winter = 1) so the coefficient for predicting a long winter is -0.618 and the odds ratio is $\exp(-.618) = .539$.

When Phil has predicted a long winter, the odds for an actually early spring (i.e., Actual_warm = 1) are 0.539 times the odds for an actually early spring when Phil has predicted an early spring.

Question 3 (1 pt)

Given that Phil predicted an early spring, what are the **odds** that the temperature in March will be above average?

Answer to Question 3 The logistic regression we have fit yields:

$$\log(\frac{\theta}{1-\theta}) = 0.539 - 0.618 \times \text{Predict Winter}$$

so plugging in Predict Winter = 0, we have $\log(\frac{\theta}{1-\theta}) = .539$. To get to the odds, we exponentiate that so the odds are:

$$\frac{\theta}{1-\theta} = \exp(.539) = 1.714$$

Question 4 (1 pt)

Given that Phil predicted a early spring, what is the **probability** that the temperature in March will be above average?

Answer to Question 4 To get from the odds to the probability we use

$$\text{Prob} = \frac{\text{odds}}{1 + \text{odds}}$$

so the probability of an early spring where the temp in March is above average is

$$\text{Prob} = \frac{1.714}{1 + 1.714} = .631$$

Question 5 (2 pts)

Does Phil seem to be helpful in predicting the weather in March? Why or why not?

Answer to Question 5 Since the coefficient is negative and the odds ratio is less than 1, is seems that Phil could be helpful since the odds of an early spring decrease when Phil predicts a long winter. However, the estimated coefficient is not statistically significant so it's not clear if Phil's success is just due to random chance.

NYC Bike Data

In the following data we have recorded the total number of bicycles which crossed the Manhattan Bridge in New York City each day during 2018[1]. We also have included information about the day of the week (week day or weekend), and information about the weather that day[2].

```
bike_data_train <- read.csv("https://raw.githubusercontent.com/ysamwang/btry6020_sp22/main/lectureData/

dim(bike_data_train)
```

```
## [1] 273  16
```

```
names(bike_data_train)
```

```
##  [1] "date"        "bike_counts" "weekEnd"     "Wind_avg"    "Precip"
##  [6] "Snowfall"    "Snowdepth"   "Temp_avg"    "Temp_max"    "Temp_min"
## [11] "Wind_fast2m" "Wind_fast5s" "FOG"         "Thunder"     "IcePellets"
## [16] "Smoke"
```

The variables in the data set include:

- date: the date of the observation
- bike_counts : the dependant variable of interest which is the total number of bikes which crossed the bridge that day
- weekEnd: Weekend or Weekday
- Wind_avg : average wind on that day
- Precip : precipitation in inches
- Snowfall : snowfall in inches
- Snodepth : amount of snow on the ground
- Temp_avg : the average temperature throughout the day
- Temp_max : the maximum temperature for the day
- Temp_min : the minimum temperature for the day
- Wind_fast2m : the fastest wind speed which was sustained for at least 2 minutes
- Wind_fast5s : the fastest wind speed which was sustained for at least 5 seconds
- FOG : whether there was fog
- Thunder : whether there was Thunder
- IcePellets : whether there was Ice Pellets
- Smoke : whether there was smoke

Question 6 (1 pt)

If we are interested in seeing how the weather affects the number of bicycles which cross the Manhattan bridge. It seems link a Poisson regression might be appropriate for this data since it is count data. What is the link function used in Poisson regression? What is the relationship between the mean and variance in a Poisson distribution?

Answer to Question 6 In the poisson regression model, we use the model

$$\log(\theta) = b_0 + \sum_k b_k x_{i,k}$$

and the link function is log. In addition, we use $\theta(\mathbf{X})$ to denote the mean of the Poisson distribution (given covariates $\mathbf{X}$. The variance of the Poisson distribution is the same value as the mean.

[1]The data below is a cleaned version of data from the NYC open data https://data.cityofnewyork.us/Transportation/Bicycle-Counts/uczf-rk3c

[2]The weather data was recorded at the JFK airport weather station and is available from NOAA at https://www.ncdc.noaa.gov/cdo-web/search?datasetid=GHCND

Question 7 (1 pt)

Fit a Poisson regression model where the outcome of interest is the number of bikes crossing the Manhattan bridge and the covariates of interest are weekend, precipitation, the average temperature, the minimum temperature, snowfall, and fog.

```
# Use glm to fit a generalized linear model
# specify the family to be poisson
poisson_mod <- glm(bike_counts ~ weekEnd + Precip + Temp_avg + Temp_min + Snowfall + FOG,
                        data = bike_data_train, family = "poisson")
summary(poisson_mod)
```

Answer to Question 7

```
##
## Call:
## glm(formula = bike_counts ~ weekEnd + Precip + Temp_avg + Temp_min +
##     Snowfall + FOG, family = "poisson", data = bike_data_train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.0366336  0.0049101 1433.08   <2e-16 ***
## weekEndTRUE -0.4290664  0.0024242 -177.00   <2e-16 ***
## Precip      -0.4805219  0.0046257 -103.88   <2e-16 ***
## Temp_avg     0.0439298  0.0003644  120.56   <2e-16 ***
## Temp_min    -0.0226026  0.0003586  -63.03   <2e-16 ***
## Snowfall    -0.2580905  0.0044355  -58.19   <2e-16 ***
## FOG         -0.1141013  0.0022709  -50.24   <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: 328571  on 272  degrees of freedom
## Residual deviance: 102015  on 266  degrees of freedom
## AIC: 104729
##
## Number of Fisher Scoring iterations: 4
```

Question 8 (1 pt)

How should you interpret the estimated coefficient for minimum temperature in the model above?

Answer to Question 8 Suppose day 1 and day 2 have the same values of the included covariates, except the minimum temperature for day 1 is 1 degree higher than the minimum temperature for day 2. Then we would expect the log of the expected number of bikes to be -0.023 lower on day 1 than on day 2. This means, that the expected number of bikes would be -2.27% lower on day 1 than day 2 (because $100(\exp(-0.023) - 1) = -2.27$).

It may be a bit surprising at first that a higher minimum temperature is associated with a decreased number of bikes. However, notice that the coefficient associated with the average temperature is positive, as we would expect. Now consider two days with the same average temperature. In most cases, a lower minimum temperature will be associated with a higher maximum temperature and a higher minimum temp will be associated with a lower maximum temp (since we hold the average constant). In this light, it may be less

surprising that the coefficient of minimum temp is negative and highlights that we need to be a bit careful with the interpretation of the coefficient of the minimum temp when controlling for the average temp.

Question 9 (1 pt)

We can test an individual coefficient using the output of `summary`. But as we discussed in class, we can also create confidence intervals and tests using the χ^2 test which comes from using the likelihood. Test whether the average temperature is statistically significant using the χ^2 test.

Answer to Question 9 To use the χ^2 test, we can compare a null model where we exclude average temperature to the alternative model (fit above) where we have included average temperature. Using the anova function to compare the two models, we see that the χ^2 test gives a p-value of nearly 0 and we reject the null hypothesis that the coefficient of average temperature is 0.

```
# model where we have excluded avg temp
mod_possion_null <- glm(bike_counts ~ weekEnd + Precip + Temp_min + Snowfall + FOG,
                        data = bike_data_train, family = "poisson")
anova(mod_possion_null, poisson_mod, test = "Chisq")
```

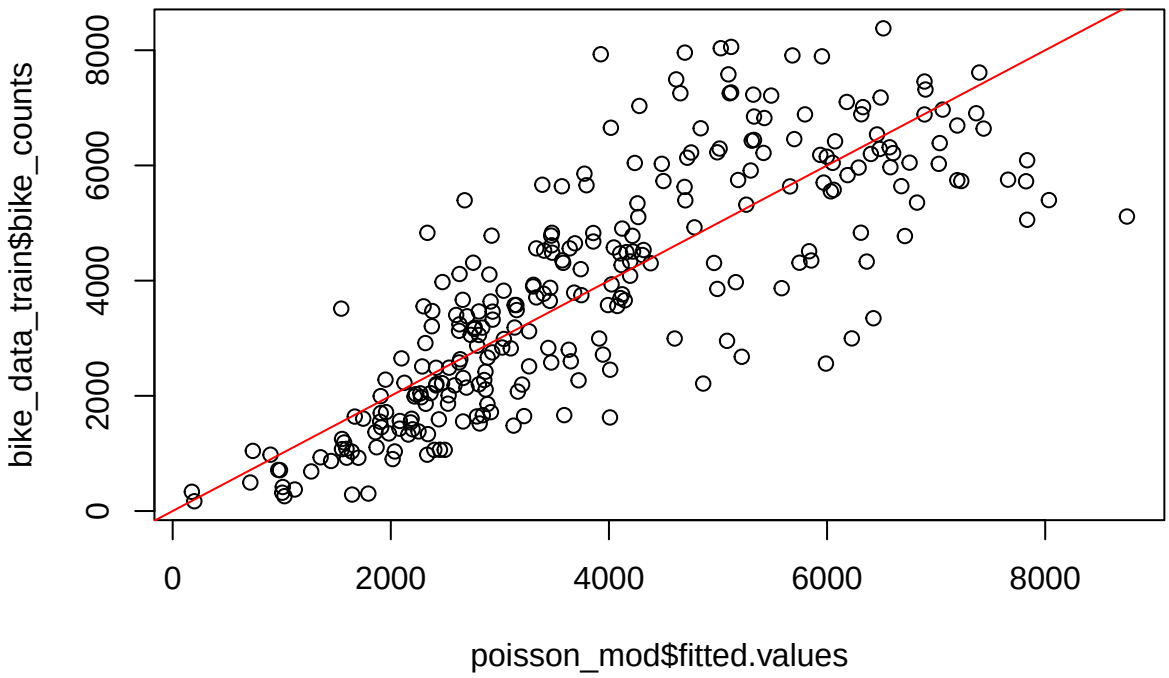
```
## Analysis of Deviance Table
##
## Model 1: bike_counts ~ weekEnd + Precip + Temp_min + Snowfall + FOG
## Model 2: bike_counts ~ weekEnd + Precip + Temp_avg + Temp_min + Snowfall +
##     FOG
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       267     116203
## 2       266     102015  1    14188 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 10 (5 pts)

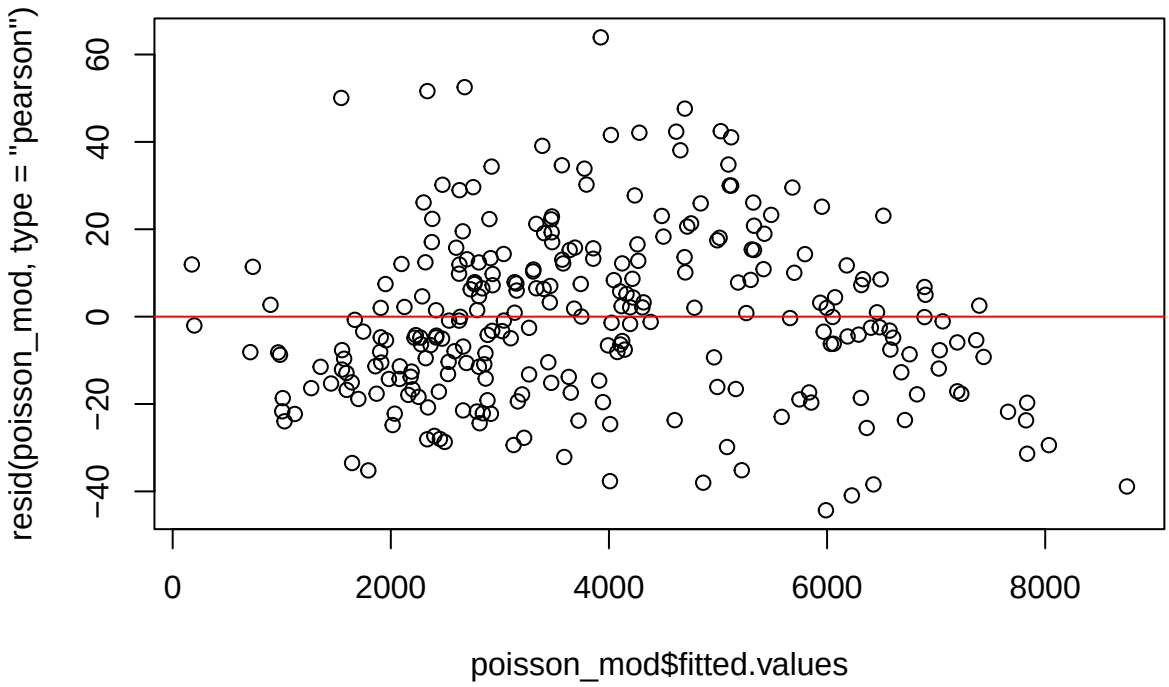
Do the assumptions for Poisson regression seem satisfied for this data? Why or why not? You can use plots or other code to justify your answers if needed.

Answer to Question 10 To assess the mean model assumption, we can plot the fitted values against the observed values. We include a red line with a slope of 1. If the fitted values and the observed values are all the same, they would fall along the red line. In this case, it seems that the predicted values fit the observed values relatively well and are evenly distributed around the red line. We can also plot the (pearson) residuals against the fitted values. There may be a slightly non-linear pattern, but nothing particularly obvious and the points seems to be roughly scattered around the red line. Note that we use the pearson residuals which standardize the residuals by the estimated variance so that we don't have the increasing variance in the pearson residuals which we'd expect in the raw residuals.

```
plot(poisson_mod$fitted.values, bike_data_train$bike_counts)
abline(a = 0, b = 1, col = "red")
```




```r
plot(poisson_mod$fitted.values, resid(poisson_mod, type = "pearson"))
abline(h = 0, col = "red")
```



To assess the variance assumption, we can fit a model which allows for over/underdispersion by setting the family to "quasipoisson." We see that the estimated dispersion parameter here is 379.7. If there were no over/under dispersion and the variance specification was correct, we'd see an estimated dispersion parameter close to 1. Without doing a formal test, since 379.7 is very far from 1, we'd say that the variance assumption is not correct, and we'd want to use the standard errors from the overdispersed model instead of the standard errors from the model where dispersion parameter is fixed to 1.

```
quasipoisson_mod <- glm(bike_counts ~ weekEnd + Precip + Temp_avg + Temp_min + Snowfall + FOG,
                        data = bike_data_train, family = "quasipoisson")
summary(quasipoisson_mod)
```

```
##
## Call:
## glm(formula = bike_counts ~ weekEnd + Precip + Temp_avg + Temp_min +
##     Snowfall + FOG, family = "quasipoisson", data = bike_data_train)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.036634   0.095679  73.544  < 2e-16 ***
## weekEndTRUE -0.429066   0.047237  -9.083  < 2e-16 ***
## Precip      -0.480522   0.090136  -5.331 2.09e-07 ***
## Temp_avg     0.043930   0.007100   6.187 2.31e-09 ***
## Temp_min    -0.022603   0.006988  -3.234  0.00137 **
## Snowfall    -0.258090   0.086431  -2.986  0.00309 **
## FOG         -0.114101   0.044251  -2.578  0.01046 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

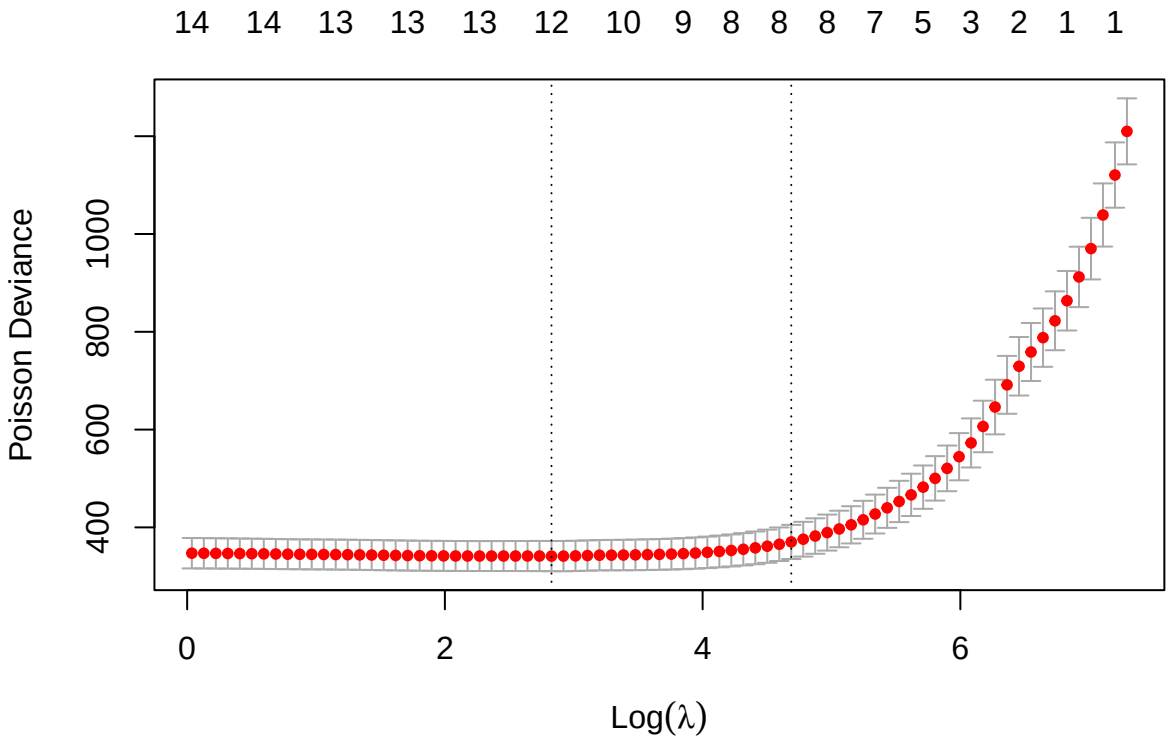
```
##
## (Dispersion parameter for quasipoisson family taken to be 379.7048)
##
##     Null deviance: 328571  on 272  degrees of freedom
## Residual deviance: 102015  on 266  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

Assessing whether the observations are independent (given the covariates) is a bit harder to say. It's likely that there are additional factors affecting the observed counts which we don't capture in the covariates. For instance, around Christmas and Thanksgiving perhaps there are more tourists which increase the number of bike crossings for those days. Whether these confounding variables which induce dependence across observations make enough of an impact to worry about is pretty subjective.

High-dimensional Regression

Even though we have more observations than covariates here, we can still use the **Lasso** procedure to select a model. In particular, the following code selects the penalty parameter λ value through a cross validation procedure. The plot shows the Deviance (a measure of fit calculated using the likelihood) and the horizontal axis shows various values of the penalty parameter. When the deviance is larger, this indicates that the coefficients do not fit the data as well. The estimated coefficients for the selected model are printed below.

```r
# fit the lasso (alph = 1 indicates lasso) with a poisson family
lasso_mod <- glmnet::cv.glmnet(y = bike_data_train$bike_counts,
                               x = as.matrix(bike_data_train[, -c(1,2)]),
                               alpha = 1, family = "poisson")
plot(lasso_mod)
```

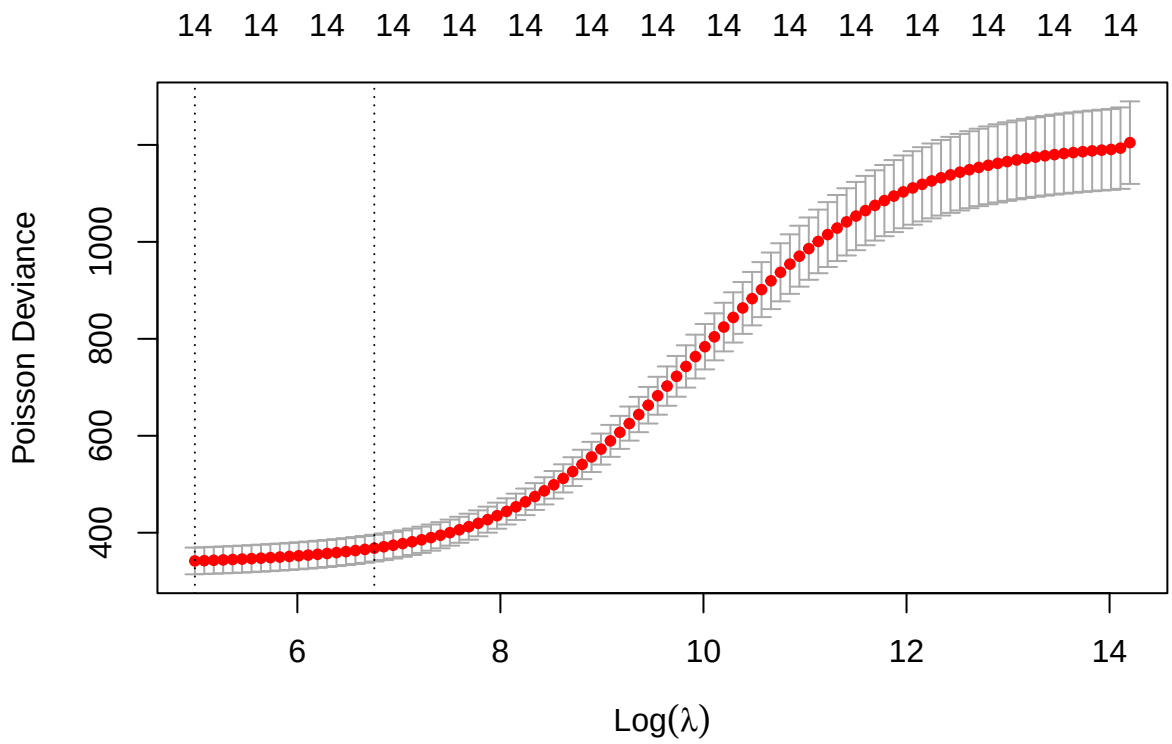



```
# We can see the coefficients selected by the largest lambda value with a
# CV error within 1 standard error of the lowest CV error
coef(lasso_mod, s = lasso_mod$lambda.1se)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  7.2745825877
## weekEnd     -0.3402707593
## Wind_avg    -0.0093288861
## Precip      -0.3903083854
## Snowfall    -0.0314681223
## Snowdepth   -0.0535938219
## Temp_avg     .
## Temp_max     0.0189286481
## Temp_min     .
## Wind_fast2m  .
## Wind_fast5s -0.0002639948
## FOG         -0.0966991305
## Thunder      .
## IcePellets   .
## Smoke        .
```

The following code uses **ridge regression** to estimate coefficients for the Poisson regression and uses a λ value which is selected by cross validation. The plot shows the Deviance (a measure of fit calculated using the likelihood) and the horizontal axis shows various values of the penalty parameter. When the deviance is larger, this indicates that the coefficients do not fit the data as well. The estimated coefficients for the selected model are printed below.

```r
# fit the ridge regression (alpha = 0 indicates ridge) with a poisson family
ridge_mod <- glmnet::cv.glmnet(y = bike_data_train$bike_counts,
                               x = as.matrix(bike_data_train[, -c(1,2)]),
                               alpha = 0, family = "poisson")
plot(ridge_mod)
```




```r
# We can see the coefficients selected by the largest lambda value with a
# CV error within 1 standard error of the lowest CV error
coef(ridge_mod, s = ridge_mod$lambda.1se)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  7.5635023674
## weekEnd     -0.3362789094
## Wind_avg    -0.0114433124
## Precip      -0.3451095813
## Snowfall    -0.0877800485
## Snowdepth   -0.0692909593
## Temp_avg     0.0057360144
## Temp_max     0.0085605055
## Temp_min     0.0032437450
## Wind_fast2m  0.0004042915
## Wind_fast5s -0.0031002322
## FOG         -0.1493816651
## Thunder      0.0429790692
## IcePellets  -0.2498911762
```

```
## Smoke         0.1524422704
```

Question 11 (1 pts)

There is not one right answer, but which model would you prefer to use? Explain why.

Answer to Question 11 The lasso estimates a "sparse'' model in the sense that many of the coefficients are set to exactly 0. This might make interpretation easier. On the other hand, the ridge regression keeps all of the coefficients non-zero. When the covariates are highly correlated, this could lead to better predictions.

Question 12 (2 pts)

Using the data from 2019, we compare the predictive accuracy of the coefficients estimated by the lasso and the ridge to the predictive accuracy of the coefficients estimated without any penalty (which we calculate below for you). We see that the lasso and ridge both have better predictions for new data. Explain why this is might happen using one of the fundamental statistical trade-offs we have discussed in class.

```
# Test data from 2019
bike_data_test <- read.csv("https://raw.githubusercontent.com/ysamwang/btry6020_sp22/main/lectureData/bi

# regular glm without any penalization
unpenalized_model <- glm(bike_counts ~ ., data = bike_data_train[, -1], family = "poisson")

# predictive accuracy for 2019 when using all covariates but no model seletion or penalization
## use type = "response" to get predictions in bikes, instead of log(bikes)
mean((bike_data_test$bike_counts - predict(unpenalized_model,
                                           newx = as.matrix(bike_data_test[,-c(1,2)]),
                                           type = "response"))^2)
```

```
## [1] 3251833
```

```
## Mean squared error for predictions using lasso
mean((bike_data_test$bike_counts - predict(lasso_mod, s=lasso_mod$lambda.1se,
                                           newx = as.matrix(bike_data_test[,-c(1,2)]),
                                           type = "response"))^2)
```

```
## [1] 1028922
```

```
## Mean squared error for predictions using ridge regression
mean((bike_data_test$bike_counts - predict(ridge_mod, s=ridge_mod$lambda.1se,
                                           newx = as.matrix(bike_data_test[,-c(1,2)]),
                                           type = "response"))^2)
```

```
## [1] 1053899
```

Answer to Question 12 This is an example of the bias variance trade-off. Using the penalized regression models incur additional bias, and we can see that they tend to fit the 2018 data better; i.e., they have lower RSS. However, in general the estimated coefficients will have lower variability which may lead to better out of sample prediction. Thus, as we see, they make better predictions for the new 2019 data.