PH245-Lab5-questions

Crop History Example

A group of bacterial pathogens is known to cause damage to soybeans. In mid August (in Wisconsin), 25 different fields were scored for pathogen damage. The score for each field was a number from 1 to 10 where 1 represents negligible damage and 10 represents severe damage. The scoring is based on a visual examination from a small plane flying overhead. Each field has associated with it a weather station for obtaining climatological data. Also, it was noted for each field what type of crop had been grown the previous year since this might affect pathogen damage. The objective of this problem is to find a useful model relating damage score to the factors of interest. These factors are described as follows:

- Rainfall: Total precipitation in inches for the last 30 days prior to date of scoring.
- Wind: Average wind speed in miles per hour for the 30 days prior to date of scoring.
- Temperature: Average high temperature (degrees Fahrenheit) for the 30 days prior to scoring date.
- Crop History: Code for crop planted on each field during the previous growing season. 1=soybeans, 2=oats, 3=snap beans

```
# clear workspace

# set working directory (optional)

# load data (download PH245Lab5_data.csv from bCourses)

# view the first few rows and columns of the dataset
```

Model (Use Soybean as Baseline/Reference Group)

Formula: $Y = \beta 0X0 + \beta 11X1 + \beta 2X2 + \beta 3X3 + \beta 4X4 + \beta 5X5 + \epsilon$ Where:

- Y: score
- $\beta 0$: intercept
- $\beta 1$: coefficient for rainfall
- $\beta 2$: coefficient for wind
- $\beta 3$: coefficient for temperature
- $\beta 4$: coefficient for crop history: oats
- $\beta 5$: coefficient for crop history: snap bean
- X1: rainfall
- X2: wind
- X3: temperature
- X4: crop History: oats
- X5: crop History: snap beans
- ϵ: error term

```
# the variable Crop.History is a categorical variable (denotes the type of crop), so let's
# check how the variable is coded in the dataset

# since the Crop.History variable is not coded as a factor, let's create a new variable
# that stores this data recoded as a factor using the as.factor command

# let's change the labels of the levels for each group in the fCropHistory variable so they
# are more descriptive
```

```
# you can check to see how the labels of the levels have now changed

# fit a linear model to the data
# we are using the variable names to refer to the dependent/independent variables in the code

#   Fit a   linear  model   to  the data
```

Change Baseline/Reference Group (Use Snapbean as Baseline/Reference Group)

```
# we can reorder the levels of the fCropHistory variable to make snapbeans the first level
# (baseline/reference group) rather than soybeans (as in the model above)
# note: when using the levels command, you need to use the same level names as previously specified,
# just in a different order

# fit a new linear model (with the changed levels for fCropHistory)
```

Create Dummy Variable Manually

```
# can also code model2 using dummy variables instead of factor variables:
# note: it seems that R can handle dummy variables without changing their types to factors
# first, create a new variable that contains 0s for the length of the sample size of the dataset

# recode this variable to fill in 1s for the observations that are coded as soybeans in the
# fCropHistory variable

# similarly code a new dummy variable for oats

# fit a new linear model with the constructed dummy variables instead of fCropHistory
```

Model Selection (We will have more examples/practice on this later!)

```
# let's try to compute the F statistic that appears in the lm command (in this case, the
# restricted model is the model that only includes the intercept term)

# for any linear regression model, the degrees of freedom using in the calculation of the
# F statistic are calculated as n - p - 1, where n = number of observations and p = number of covariate
# restricted model: n = 25, p = 0 (since no covariates), df = 25 - 0 - 1 = 24
# full model (model 3): n = 25, p = 5, df = 25 - 5 - 1 = 19

# can calculate the F statistic (see lecture 4, slide 27 for the formula)
# the deviance functions gives SSE
# SSE: the sum of squared residuals

# get the p-value associated with this F statistic, using the pf function
# the test statistic has an F distribution with (df_R- df_F, df_F) degrees of freedom
```