Stochastic Data Processing and Simulation, Assignment A2

Sarah Al-khateeb

1 Introduction

As data scientists, we are trying to find a model that represents the data and provide a tool, which helps in predicting the value of a dependent variable and obtain the parameter estimates. In this lab, we are interested in assessing the variability of our predictions using Prediction Intervals, which is a way to quantify and communicate the uncertainty in a prediction (describe the uncertainty for a single specific outcome). One of the challenges that we face is how to choose among a range of different models that we can use for a specific problem because we do not know beforehand the true model, we fit different models for a specific problem and observe results to select the proper level of flexibility. Finally, we want to ask questions of a population but it is hard to get all population, and so we take a sample and ask questions of it instead. How confident we should be that the sample answer is close to the population answer depends on the structure of the population. One way we might learn about this is to take samples from the population, ask them the questions, and see what are the sample answers. Moreover, many things can affect how well a sample represents the population; and therefore, how valid and reliable the conclusions will be. We will use the bootstrap technique that resamples a single dataset to create many simulated samples that help in answering many questions.

2 Exercise 1

2.1 Problem

In this problem, we will use the cars data to predict the future distances of 5 new speeds and the random variation around it (uncertainty quantification).

2.2 Theory and implementation

- First, I fitted a linear model: $E(dist/speed) = \beta_0 + \beta_1.speed$ to car data and obtained parameter estimates ($Intercept : \beta_0$, $Slope : \beta_1$, $StandardError : s$). These parameters will be used to predict the future responses (distances) at new speed observations [5,10,15,20,25]. Since we want to predict new measurements, we add some new measurement error (using rnorm function and specify mean 0 and standard deviation s) to the fitted model because we want to reason about potential new measurements and how they distribute. for a new prediction y_{new} at some $x = x^*$, the formula is

$$\hat{y}_{new,b} = \beta_0 + \beta_1 x^* + \epsilon_{new,b}, \qquad \epsilon_{new,b} \sim \mathrm{N}(0,\mathrm{s}^2) \qquad \mathrm{b} = 1,....,\mathrm{B}.$$

- Then, I generated B = 5000 predictions of responses at speeds [5,10,15,20,25] using the above formula and estimated parameters from the data to calculate the 2.5 and 97.5 empirical quantiles (using quantile function) of the new responses. These quantiles are the boundaries of the 95% prediction interval for future observations.

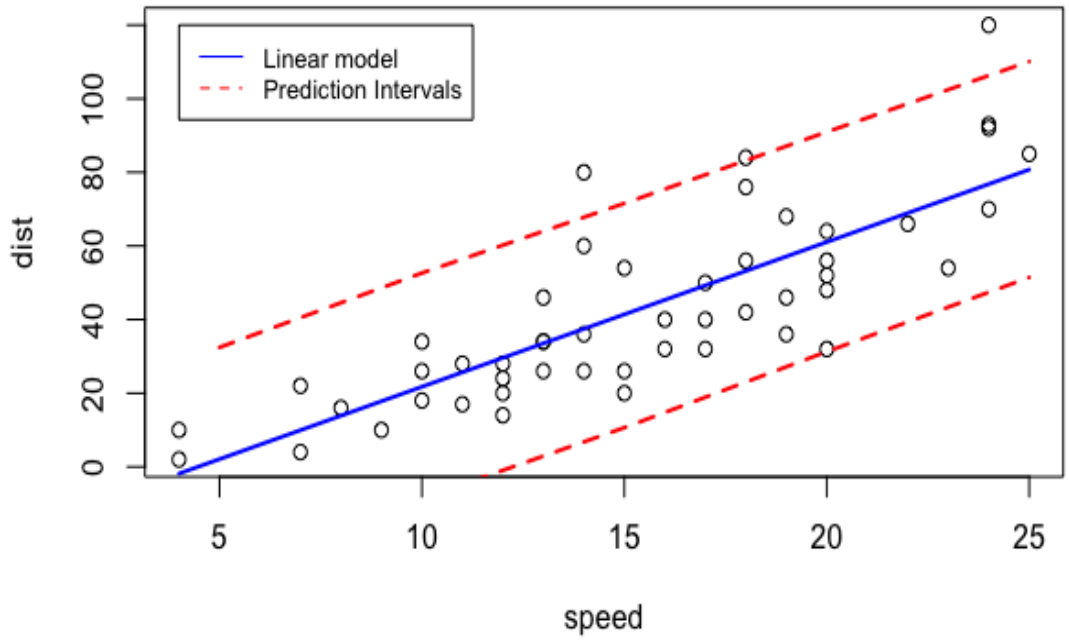

Figure 1: 95% prediction intervals (dashed red lines) and the linear regression fit E(dist/speed) (blue line).

.

2.3 Results and discussion

- A scatter plot was used to show the results for this exercise where the speed is plotted against the distance as shown in **Figure 1** and prediction intervals are the plotted lines connecting the 5 lower and upper bounds.
- From **Figure 1**, we can see that the prediction interval contains the majority of observation except for two points. these two points seem to be an outlier in this dataset, the first one refers to speed $\sim$ 15 with dist $\sim$ 80 while the second outlier has speed $\sim$ 25 and dist over 100. It seems that for those speed categories 15 and 25 the range of dist is $\sim$ 20-60 and 40-80 respectively.

3 Exercise 2

3.1 Problem

In this problem, we will explore the relationship between wage and age using the Wage data. We will try to find a good predictive model for the wage using only the age covariate and polynomial regression.

3.2 Theory and implementation

- First, I explored the Wage data that contains one covariate (age) and a response variable (wage). In **Figure 2** we can notice from the scatterplot that wage increases with age but

Figure 2: Scatter plot of the Relationship between Wage and Age.

.

then decreases again after approximately age 55, and it does not seem that wage depends on Age very strongly. However, we can see a set of points towards the top is very different from the rest (outliers).

- For this problem, I used polynomial regression of different degrees (1-10) (using lm function) to test which degree will give a good predictive performance and choose a final model. The polynomial regression formula is given by

$$E(y/x) = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_p x^p \qquad \text{p = 1,2.....}$$

- The process was done by first splitting the data into training (70%) and test data (30%) (using sample.int function to preform sampling without replacement) so we can train all models using the training part and test on unseen data (test data). After testing, I computed the pMSE the mean squared error for the test data in order to choose a model with the least error. The idea here is that we want to find a model that fits the data and also generates new data i.e. we don't want the model to overfit and kinda memorize the data because in this case, the performance on a new data will be bad. The pMSE helps in evaluating the model by calculating the average error where we compute how much the prediction differs from the truth ground:

$$pMSE_(p) = \frac{1}{m} \sum_{i=1}^{m} (y_i^{new} - \hat{y}_i(p))^2$$

  So, I used to predict on test data to have the responses associated with each model and then used the true responses to compute the pMSE.

- Selecting training and test data were done randomly in the first part and to make more firm conclusions, I executed the model selection for 50 times using a for loop, each time sampling different train/test samples and calculated the pMSE in each time to observe what is the appropriate degree to choose for this problem.

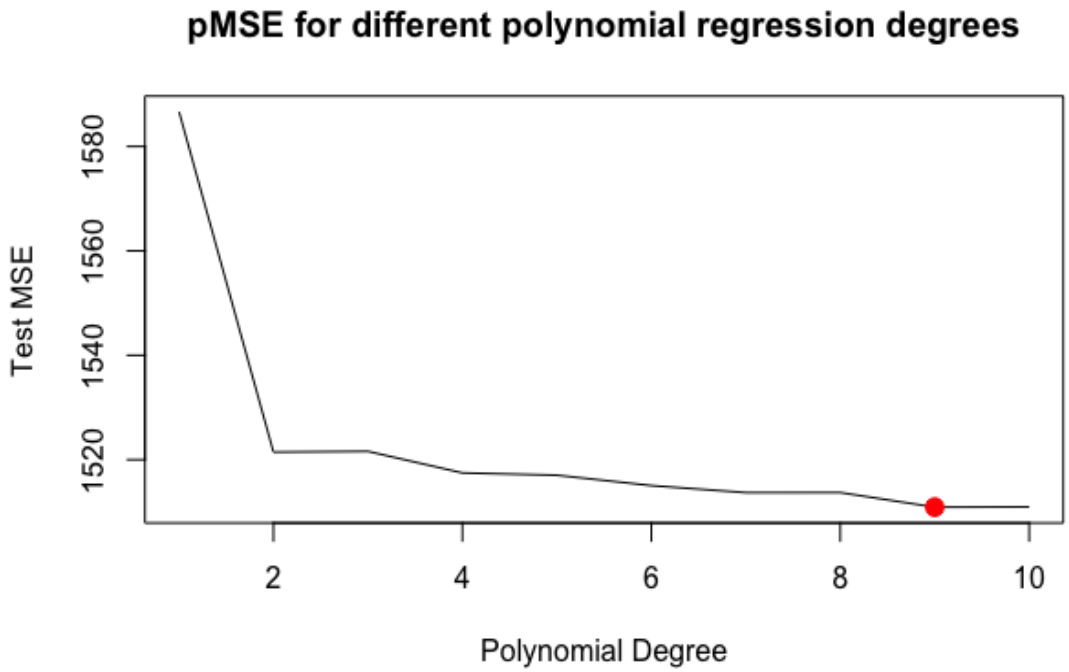

Figure 3: Line plot for different polynomial regression degrees from 1 to 10.

.

3.3 Results and discussion

- Part (i): **Figure 3** displays the pMSE for all degrees. The line shows that pMSE decrease with a higher degree and it seems that the lowest pMSE is when p = 9 but the pMSE values don't differ that much and we probably can choose a less complex model that can generalize on new data.

- Part (ii): **Figure 4** displays the model selection for 50 times sampling from the data. The lines show a similar trend where all start high and then at degree 2 the pMSE starts to decrease gradually. From this plot, we can say that it is a bit hard to choose what degree is the right one since all pMSE are really close in magnitude but since these values are not that different, choosing a less complex model would be better and taking into account the fact that we only have one covariate (age) and the data is not that large. So, I can say polynomial of degree 3 or 4 seems to be a good choice.

- Part (iii): **Figure 5** shows that the three polynomial models predict similarly except towards age=75 and above. I believe that any of these would make a sufficient fit for this data. However, if I were to take a pick of the models based on simplicity and interpretability, I would go with the polynomial with degree=3.

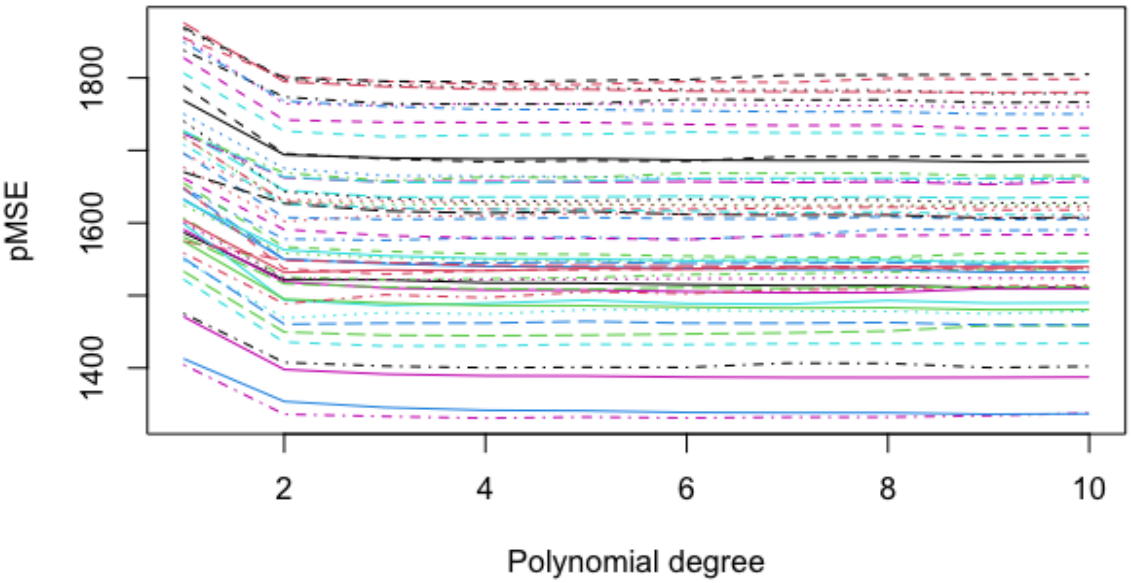

Figure 4: Line plot of model selection for 50 times for each polynomial degree.

.

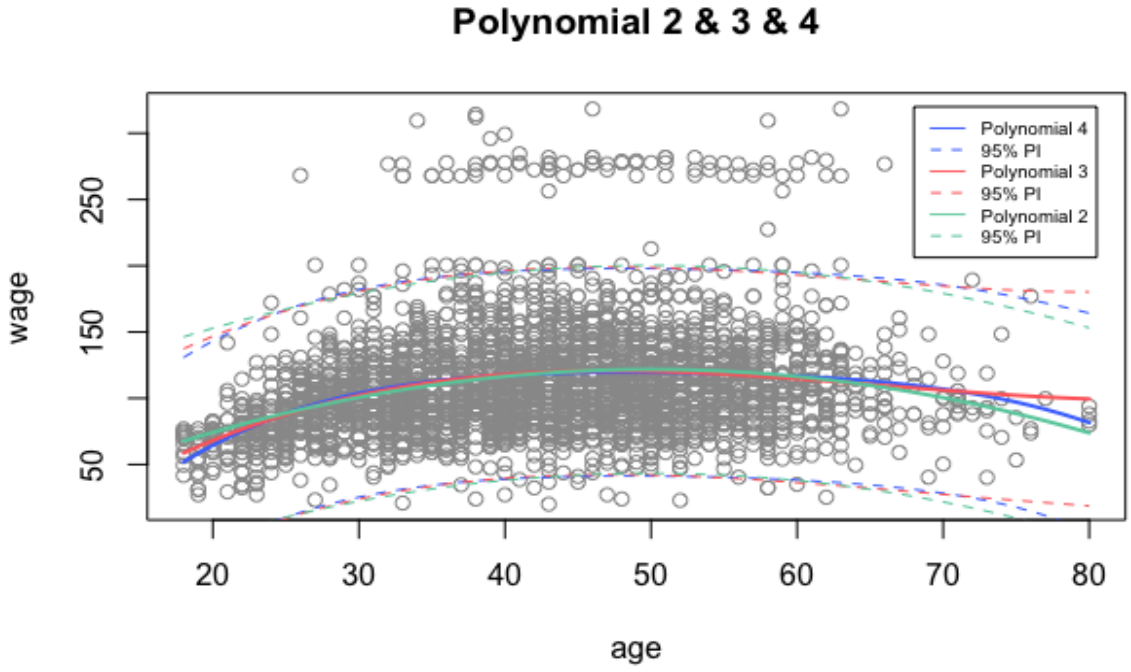

Figure 5: Scatter plot with three polynomial models and 95% prediction interval for each model.

.

4 Exercise 3

4.1 Problem

In this exercise, the objective is to simulate a random variable X such that $P(X \leq x) = F(x), x \in R$. Besides being able to estimate parameters from a model, we are interested in making inferences about unknowns when we have unobserved data, and quantify how certain we are about this. To do that we will be using bootstrapping along with the inverse transform method to generate sample numbers at random and assign measures of accuracy to sample estimates.

4.2 Theory and implementation

- In this exercise, we will use the Atlantic data, which contains the significant wave height recorded 14 times a month during several winter months in the North Atlantic. This event considered as a rare event and so a Gumbel distribution is considered to be a good representative for such data:

$$F(x; \mu, \beta) = exp(-exp(-\tfrac{x-\mu}{\beta})), x \in R$$

- Part (a): In order to generate sample numbers we need to use the inverse transform to get the formula:

$$\text{Let } U \sim U(0,1) \text{ and we have } x \sim Gumbel(\mu, \beta)$$

$$U = F(x) \implies u = F(x; \mu, \beta) \implies u = exp(-exp(-\tfrac{x-\mu}{\beta}))$$

$$ln(u) = -exp(-\tfrac{x-\mu}{\beta}) \implies ln(-ln(u)) = -\tfrac{x-\mu}{\beta}$$

$$x - \mu = -\beta ln(-ln(u)) \implies x = \mu - \beta ln(-ln(u))$$

- After deriving the formula to generate gumble draws we calculated μ and β using maximum likelihood from the Atlantic data. The sample was simulated of Gumbel data of size Atlantic data (using rand function) and a Q–Q plot used to compare the distribution of the Atlantic data and the simulated data (using qqplot function). In this scenario, a parametric bootstrap algorithm was used since we know the data comes from a Gumble distribution family.

- After that, a percentile method was used to obtain 95% confidence bounds based on 10,000 simulations (using prctile function and [2.5,97.5])

- Now, we are interested in making some predictions for this rare event and try to calculate the highest expected waves during a 100-year period. Using the bootstrapped parameters obtained from the previous section, I calculated a parametric bootstrapped 95% confidence interval using that for a 100-year return T = 3 * 14 * 100 (The Tth return value is given by $F^{-1}(1 - 1/T; \mu, \beta)$

- Sometimes model simulation is expensive and so preforming the parametric bootstrap algorithm is not possible. In this case, a non-parametric algorithm can be used where we generate 10,000 bootstrap samples with replacement from Atlantic data (using randsample function) and obtain the estimates.

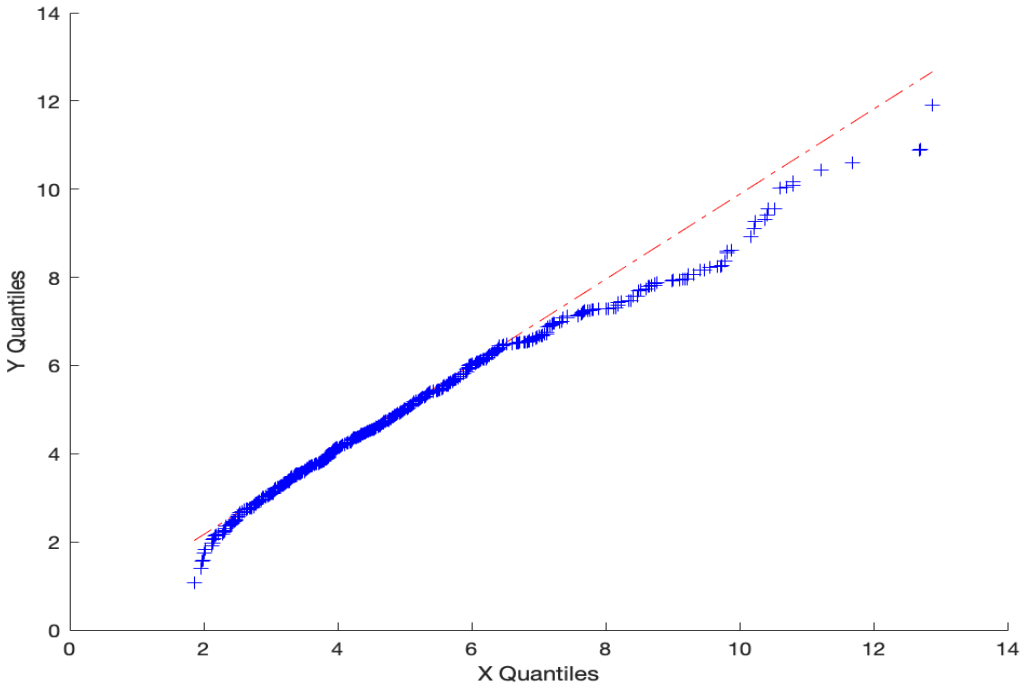

Figure 6: Normal Q-Q plot for the distributions of the Atlantic data and the simulated data.

.

4.3 Results and discussion

- Part (b): **Figure 5** shows a Q-Q plot which is a scatter plot created by plotting the quantiles of the Atlantic data and the simulated data. The plot shows that the two sets of data come from the same distribution and the line is roughly straight. We can notice that the upper end of the Q-Q plot deviates from the straight line while the lower follows a straight line and so we can say that the curve is right-skewed.

- Part (c): The maximum likelihood estimates obtained from the atlantic data are $\mu = 4.15$ and $\beta = 1.49$. Computing a 95% confidence interval resulted in $\mu_{CI} = [4.0232, 4.2750]$ and $\beta_{CI} = [1.3902, 1.5800]$ and these intervals contain the parameters obtained from the maximum likelihood. The CI means that 95% of the time, the parameter of interest will be between the two numbers, and 5% of the time it will not.

- Part (d): The parametric bootstrapped 95% confidence interval for the 100-year return value is [15.6961, 17.3847]. So we can say that the expected value of the highest wave in a 100 years period will be between 15.7 and 17.4.

- Part (e): Since the highest expected wave lies between 15.7 and 17.4,and the maximum wave height form data is ~ 13, we can say that the height increase is about 15-30% so I could advise the city council to build a barrier with at least 18-20 ft high to prepare for the worst future scenarios.

- Part (f): Bootstrap sample with replacement from the atlantic data, the estimated $\mu_{CI} = [4.0242, 4.2737]$ and $\beta_{CI} = [1.3911, 1.5806]$. Comparing these results with the parametric method, I can say that the results are similar and the non-parametric algorithm can be used to get good approximations for unknown observations.

5 Exercise 4

5.1 Problem

Simple linear regression was an insufficient model to represent the Wage data, and so we will apply a parametric algorithm to the Wage data and try to approximate solutions for a polynomial model of degree 3 using bootstrap.

5.2 Theory and implementation

- I obtained the estimates from a polynomial model of degree 3 using the response wage and covariate age. Based on 2000 bootstrap samples, I collected the estimates for different datasets sampled from the available data with replacement (using sample.int function).
- For each of the estimated parameters, a 90% confidence interval was computed using the quantile function. the results compared with the results obtained from using the confint function.

5.3 Results and discussion

- **Table 1** shows the confidence intervals obtained from the simulations:

Table 1: Estimates 90% Confidence Interval.

Estimates	**5%**	**95%**
β_0	-103.85	-47.25
β_1	8.12	12.41
β_2	-0.221	-0.120
β_3	0.0005	0.0012

- **Table 2** shows the confidence intervals obtained from the confint function:

Table 2: confint 90% Confidence Interval.

Estimates	**5%**	**95%**
β_0	-111.74	-38.74
β_1	7.55	12.83
β_2	-0.229	-0.107
β_3	0.0004	0.0013

- From **Table 1** and **Table 2** we can notice that the values are similar. This indicates that the bootstrap method is a really good method for inference the unknown and we can obtain reliable results using it.

Appendix - code

```
# Exercise 1:
data(cars)
attach(cars)  # read the columns names in the dataset

#fit a model and extract necessary parameters from the model mm
mm <- lm(dist~speed)
b0 <- mm$coefficients[1]
b1 <- mm$coefficients[2]
s <- summary(mm)$sigma

speeds <- c(5,10,15,20,25)

# plug above parameters to obtain predictions for five speeds with added
    noise
ynew1 <- b0 + b1*speeds[1] + rnorm(n=5000, mean=0, sd=s)
ynew2 <- b0 + b1*speeds[2] + rnorm(n=5000, mean=0, sd=s)
ynew3 <- b0 + b1*speeds[3] + rnorm(n=5000, mean=0, sd=s)
ynew4 <- b0 + b1*speeds[4] + rnorm(n=5000, mean=0, sd=s)
ynew5 <- b0 + b1*speeds[5] + rnorm(n=5000, mean=0, sd=s)

# 95% prediction intervals (lower bound, upper bound)
q1 <- quantile(ynew1, c(0.025, 0.975))
q2 <- quantile(ynew2, c(0.025, 0.975))
q3 <- quantile(ynew3, c(0.025, 0.975))
q4 <- quantile(ynew4, c(0.025, 0.975))
q5 <- quantile(ynew5, c(0.025, 0.975))

#combine lower bounds and upper bounds separately
lower <- c(q1[1],q2[1],q3[1],q4[1],q5[1])
upper <- c(q1[2],q2[2],q3[2],q4[2],q5[2])

#plot data and lines for the model and lower, upper bounds
plot(speed,dist)
lines(speed,mm$fitted, lwd=2, col='blue')
lines(speeds, lower, lty=2, lwd=2, col='red')
lines(speeds, upper, lty=2, lwd=2, col='red')

# Add a legend
legend(4, 120, legend=c("Linear model", "Prediction Intervals"),
       col=c("blue", "red"), lty=1:2, cex=0.8)
```

```
# Exercise 2:
#load libraries
#install.packages("Rmisc")
library(Rmisc)
library(ggplot2)
library(reshape2)

Wage<-read.table("Wage.txt",header=TRUE)
attach(Wage)

#plot wage vs age
plot(age, wage)
title(main = "Relationship between Age and Wage")

#Part (i)
#split data into train and test
#split 70% train and 20% test
N = dim(Wage)[1]
n=floor(0.7*N)
#adding a seed to reproduce results
set.seed(321)
#create indeces randomly to split the train and test
# Randomly identifies the rows equal to sample size from all the rows of
    Wage dataset and stores the row number in ind
ind = sample.int(nrow(Wage),size = n)
train = Wage[ind,]
test = Wage[-ind,]

#Compute the pMSE corresponding to each polynomial (from 1 to 10)
pmses <- data.frame('Pth.oder' = NA, 'pMSE' = NA) # empty data frame to
    store PMSE
Rsquared <- data.frame('R_squared' = NA)
#ypred <- matrix(0, nrow = length(ytest), ncol = p)
p <- seq(1,10)
for (i in p){
  ml <- lm(wage~poly(age,i, raw=TRUE), data=train)
  Rsquared[i,] <- summary(ml)$r.squared
  pmses[i,1] <- p[i] # store p-th order
  ypred <- predict(ml, newdata = data.frame(age = test$age), type="response
      ")
  pmses[i,2] <- sum((test$wage-ypred)^2) / nrow(test) # calculate PMSE of
      the test set
}

#convert the dataframe into list
pmses_v <- pmses[['pMSE']]
#plot the polynomial degree vs the PMSE
plot(p, pmses_v, xlab="Polynomial Degree", ylab = "Test MSE", type = "l")
title(main = "pMSE for different polynomial regression degrees")
#draw a red point where PMSE is smallest
d.min<-which.min(pmses_v)
points(d.min, pmses_v[d.min], col="red", cex=2, pch=20)

#Part (ii)
k <- seq(1:50)
pmse <- matrix(0,length(k),length(p)) #zero matrix to store values, rows
    are the attempts, cols are degree number
set.seed(321)
```

```
for (kk in k){
  indeces = sample.int(nrow(Wage),size = n)
  training = Wage[indeces,]
  testing = Wage[-indeces,]
  for (i in p){
    mymode <- lm(wage~poly(age,i, raw=TRUE), data=training)
    ypred <- predict(mymode, newdata = data.frame(age = testing$age), type
        ="response")
    pmse[kk,i] <- sum((testing$wage-ypred)^2) / nrow(testing) # calculate
        PMSE of the testing set
  }
}


#plot the polynomial degree vs the PMSE
cc <- do.call(cbind, lapply(seq(1, NROW(pmse), 1), function(i) pmse[i,]))
matplot(1:10, cc, type='l', xlab='Polynomial degree', ylab='pMSE')
#another plot for 50 simulations vs PMSE
matplot(pmse, type='l', xlab='The 50 evolutions of the pMSE values for
    increasing Poly. degree', ylab='pMSE')
points(k, pmse[,7], col="black",pch="+")
xlabel <- seq(0, 50, by = 2)
axis(1, at = xlabel)

#plot.ts(pmse)
# find min value for each polynomial degree
mins <- data.frame('Min' = NA)
for (m in p){
  minss <- c(min(pmse[,m]))
  mins[m,1] <- minss
}
mins
#draw a red point where PMSE is smallest
dn_min <- which(pmse == min(pmse), arr.ind=TRUE) #min pmse
points(24, pmse[dn_min], col="red", cex=2, pch=25)
points(24, 1331.110, col="blue", cex=2, pch="+") #2nd min value
points(24, 1332.191, col="green", cex=2, pch="x") #3rd min value

#Part (iii)
#chosen degrees mainly p= 4 with other value p = 2,3
plot(wage~age, data=Wage, col="grey60", main="Polynomial 2 & 3 & 4")
agelim<-range(Wage$age)
age.grid<-seq(from=agelim[1], to=agelim[2])

#fit degree 4
fit1<-lm(wage~poly(age,4), data=Wage)
preds1<-predict(fit1, newdata = list(age=age.grid))
lines(age.grid, preds1, col="royalblue1", lwd=2)

#prediction interval at confidence level 95%
pred_interval1<-predict(fit1, newdata=list(age=age.grid), interval="
    prediction",level = 0.95)
lines(age.grid, pred_interval1[,2], col="royalblue1", lty=2)
lines(age.grid, pred_interval1[,3], col="royalblue1", lty=2)

#fit degree 3
fit2<-lm(wage~poly(age,3), data=Wage)
preds2<-predict(fit2, newdata=list(age=age.grid))
```

```
lines(age.grid, preds2, col="indianred1", lwd=2)

#prediction interval at confidence level 95%
pred_interval2<-predict(fit2, newdata=list(age=age.grid), interval="
    prediction",level = 0.95)
lines(age.grid, pred_interval2[,2], col="indianred1", lty=2)
lines(age.grid, pred_interval2[,3], col="indianred1", lty=2)

#fit degree 2
fit3<-lm(wage~poly(age,2), data=Wage)
preds3<-predict(fit3, newdata=list(age=age.grid))
lines(age.grid, preds3, col="aquamarine3", lwd=2)


#prediction interval at confidence level 95%
pred_interval3<-predict(fit3, newdata=list(age=age.grid), interval="
    prediction",level = 0.95)
lines(age.grid, pred_interval3[,2], col="aquamarine3", lty=2)
lines(age.grid, pred_interval3[,3], col="aquamarine3", lty=2)

#add legends for all degrees and prediction intervals
legend(68, 320, legend=c("Polynomial 4", "95% PI", "Polynomial 3", "95% PI
    ",
                         "Polynomial 2", "95% PI"),
       col=c("royalblue1", "royalblue1", "indianred1","indianred1",
             "aquamarine3", "aquamarine3"), lty=1:2, cex=0.6)



################################################################################


# Exercise 3 (Matlab):

% Part (b)
%import atlantic file
atlantic = importdata('atlantic.txt');
%number of rows in the data
n = size(atlantic, 1);

%Maximum Liklehood estimates using est_gumbel function
out = est_gumbel(atlantic);
beta = out(1);
mu   = out(2);

%simulate a sample of Gumbel data of size atlantic data
u = rand(n,1); % 582 uniforms in (0,1)
x = mu - beta*log(-log(u)); % Gumbel draws

%qqplot to compare the distribution of the atlantic data and the simulated
    data
qqplot(atlantic,x);

% Part (c)
% the number of bootstrap simulations
B = 10000;
mu_est = zeros(B,1);  % initialize a vector of zeroes where we store mu's
beta_est = zeros(B,1);  % initialize a vector of zeroes where we store beta
```

```
    's

% set a seed, for reproducibility
rng(123);
for ii=1:B
    % simulate bootstrapped data
    u = rand(n,1); % 582 uniforms in (0,1)
    x = mu - beta*log(-log(u)); % Gumbel draws
    out = est_gumbel(x);
    mu_est(ii) = out(2);
    beta_est(ii)= out(1);
end

% use the percentile method to obtain 95% confidence bounds
mu_CI = prctile(mu_est,[2.5,97.5]);
beta_CI = prctile(beta_est,[2.5,97.5]);



% Part (d)
u = 1 - (1/(3*14*100));
xx = mu_est - beta_est*log(-log(u));
highest_wave  = prctile(xx,[2.5,97.5]);


% Part (f)
% the number of bootstrap simulations
B = 10000;
np_mu_est = zeros(B,1);  % initialize a vector of zeroes where we store mu'
    s
np_beta_est = zeros(B,1);  % initialize a vector of zeroes where we store
    beta's

% set a seed, for reproducibility
rng(123);
for ii=1:B
    % obtain(nonparametric) bootstrap sample with replacement
    replace = 1;
    sample = randsample(atlantic,n,replace);
    out = est_gumbel(sample);
    np_mu_est(ii) = out(2);
    np_beta_est(ii)= out(1);
end

% use the percentile method to obtain 95% confidence bounds
np_mu_CI = prctile(np_mu_est,[2.5,97.5]);
np_beta_CI = prctile(np_beta_est,[2.5,97.5]);

##########################################################################


#Exercise 4
#the number of bootstrap simulations
B <- 2000
beta0 <- rep(0, B) # vector of zeros to store simulated beta0 values
beta1 <- rep(0, B) #vector of zeros to store simulated beta1 values
beta2 <- rep(0, B) #vector of zeros to store simulated beta2 values
beta3 <- rep(0, B) #vector of zeros to store simulated beta3 values
```

```
#size of data
n <- nrow(Wage)

# poly degree 3 model
poly_mm <- lm(wage ~ poly(age, 3, raw=TRUE))
poly_b0<- poly_mm$coefficients[1]
poly_b1 <- poly_mm$coefficients[2]
poly_b2 <- poly_mm$coefficients[3]
poly_b3 <- poly_mm$coefficients[4]
#set a seed, for reproducibility
set.seed(321)

for (b in 1:B){
  sampling <- sample.int(n, n, replace = TRUE)
  sample <- Wage[sampling,]
  mm_new <- lm(wage ~ poly(age, 3, raw=TRUE), data = sample)
  beta0[b] <- mm_new$coefficients[1]
  beta1[b] <- mm_new$coefficients[2]
  beta2[b] <- mm_new$coefficients[3]
  beta3[b] <- mm_new$coefficients[4]
}

#compute 90% confidence intervals for simulated data
beta0_CI <- quantile(beta0, c(0.05, 0.95))
beta1_CI <- quantile(beta1, c(0.05, 0.95))
beta2_CI <- quantile(beta2, c(0.05, 0.95))
beta3_CI <- quantile(beta3, c(0.05, 0.95))

#using confint to the  model
model_CI <- confint(poly_mm, level=0.90)
```