Compulsory exercise 1: Group 20

TMA4268 Statistical Learning V2023

Henrik Grenersen, Halvard Emil Sand-Larsen, Eirik Fagerbakke

23 februar, 2023

Problem 1

In this task we will consider $Y = f(\mathbf{x}) + \epsilon$ where $\mathrm{E}[\epsilon] = 0$ and $\mathrm{Var}[\epsilon] = \sigma^2$. For this regression problem we will start with studying some properties of some estimators for β, namely $\hat{\beta}$ and $\tilde{\beta}$.

a)

We will start with determining the expected value and variance of $\tilde{\beta}$. From our problem we have the following relation

$$E[\mathbf{Y}] = \mathbf{E}[\mathbf{X}\beta + \epsilon] = \mathbf{X}\beta$$

Using this, we get an expression for the expected value like this.

$$\begin{aligned}
E[\tilde{\beta}] &= E[(X^\top X + \lambda I)^{-1} X^\top \mathbf{Y}] \\
&= (X^\top X + \lambda I)^{-1} X^\top E[\mathbf{Y}] \\
&= (X^\top X + \lambda I)^{-1} X^\top X\beta \\
&= (X^\top X + \lambda I)^{-1} (X^\top X + \lambda I - \lambda I)\beta \\
&= (X^\top X + \lambda I)^{-1} (X^\top X + \lambda I)\beta - \lambda(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{\mathbf{-1}}\beta \\
&= \beta - \lambda(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{\mathbf{-1}}\beta
\end{aligned}$$

We note that the estimator is biased, and now move on to the variance.

$$\begin{aligned}
\mathrm{Var}[\tilde{\beta}] &= \mathrm{Var}[(X^\top X + \lambda I)^{-1} X^\top \mathbf{Y}] \\
&= (X^\top X + \lambda I)^{-1} X^\top \mathrm{Var}[Y]((X^\top X + \lambda I)^{-1} X^\top)^\top \\
&= \sigma^2 (X^\top X + \lambda I)^{-1} X^\top X (X^\top X + \lambda I)^{-\top} \\
&= \sigma^2 (X^\top X + \lambda I)^{-1} (X^\top X + \lambda I - \lambda I)(X^\top X + \lambda I)^{-\top} \\
&= \sigma^2 \left((X^\top X + \lambda I)^{-1}(X^\top X + \lambda I) - \lambda (X^\top X + \lambda I)^{-1}\right)(X^\top X + \lambda I)^{-\top} \\
&= \sigma^2 (I - \lambda (X^\top X + \lambda I)^{-1})(X^\top X + \lambda I)^{-\top}
\end{aligned}$$

b)

We now want to consider the prediction $\mathbf{x_0}^\top \tilde{\beta}$, denoted $\tilde{f}(\mathbf{x_0})$. And start with finding its expected value.

$$E[\tilde{f}(\mathbf{x_0})] = \mathbf{E}[\mathbf{x_0}^\top \tilde{\beta}] = \mathbf{x_0}^\top (\beta - \lambda(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{\mathbf{-1}}\beta)$$

We then find the variance as shown.

$$\text{Var}[\tilde{f}(\mathbf{x_0})] = \text{Var}[\mathbf{x_0}^\top \tilde{\beta}] = \mathbf{x_0}^\top \text{Var}[\tilde{\beta}]\mathbf{x_0} = \sigma^{\mathbf{2}}\mathbf{x_0}^\top(\mathbf{I} - \lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}})(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\top}\mathbf{x_0}$$

c)

The bias of our method gives us an estimate of how our prediction differs from the true mean.

The irreducible error stems from the random nature of the data set, “noise”, and is normally represented by ϵ. As the name suggests, it is irreducible.

The variance relates to the training of our method, and measures how much our prediction would change for different training sets.

d)

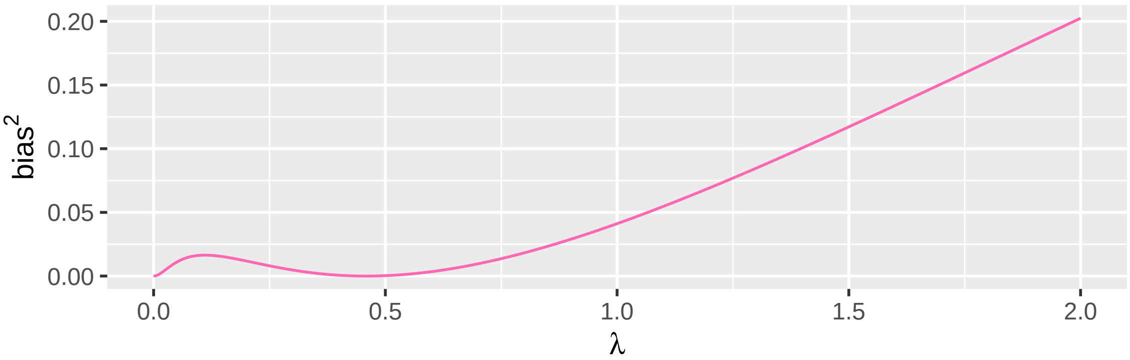
We now want to find an expression for the MSE at x_0, defined as $E[(y_0 - \tilde{f}(\mathbf{x_0}))^{\mathbf{2}}]$.

$$\begin{aligned}
E[(y_0 - \tilde{f}(\mathbf{x_0}))^{\mathbf{2}}] &= \text{Var}[\epsilon] + \text{Var}[\tilde{f}(\mathbf{x_0})] + (\mathbf{f}(\mathbf{x_0}) - \mathbf{E}[\tilde{\mathbf{f}}(\mathbf{x_0})])^{\mathbf{2}} \\
&= \sigma^2 \\
&+ \sigma^2\mathbf{x_0}^\top(\mathbf{I} - \lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}})(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\top}\mathbf{x_0} \\
&+ (\mathbf{x_0}^\top\beta - \mathbf{x_0}^\top(\beta - \lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}}\beta))^{\mathbf{2}} \\
&= \sigma^2\left(1 + \mathbf{x_0}^\top(\mathbf{I} - \lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}})(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\top}\mathbf{x_0}\right) \\
&+ (\mathbf{x_0}^\top\lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}}\beta))^{\mathbf{2}}
\end{aligned}$$

e)

We now want to make a function to calculate the squared bias. To to this we firstly define the different paramaters we will need, and then fill in the handed out code with the proper expression.

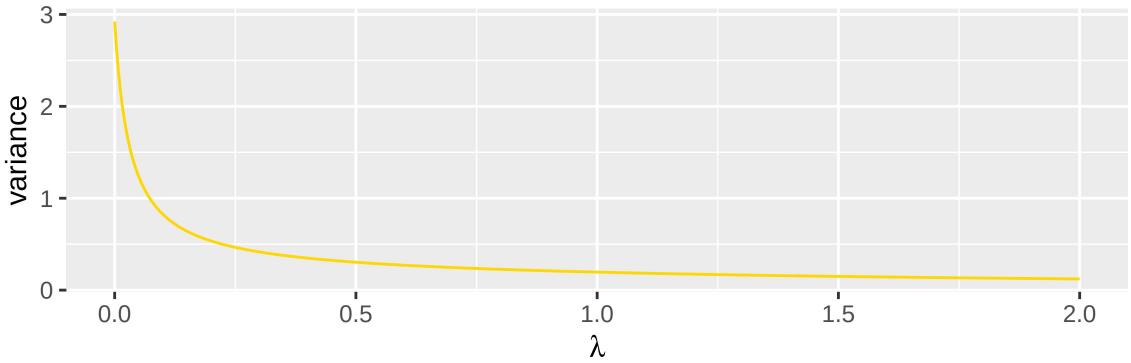
```
id <- "1X_8OKcoYbng1XvYFDirxjEWr7LtpNr1m"  # google file ID
values <- dget(sprintf("https://docs.google.com/uc?id=%s&export=download", id))

X <- values$X
# dim(X)

x0 <- values$x0
# dim(x0)

beta <- values$beta
# dim(beta)

sigma <- values$sigma
# sigma

library(ggplot2)
bias <- function(lambda, X, x0, beta) {
    p <- ncol(X)
    I = diag(p)
    ridge_inv <- solve(t(X) %*% X + lambda * I)
    value <- (lambda * t(x0) %*% ridge_inv %*% beta)^2
    return(value)
}
lambdas <- seq(0, 2, length.out = 500)
BIAS <- rep(NA, length(lambdas))
for (i in seq_along(lambdas)) BIAS[i] <- bias(lambdas[i], X, x0, beta)
dfBias <- data.frame(lambdas = lambdas, bias = BIAS)
ggplot(dfBias, aes(x = lambdas, y = bias)) + geom_line(color = "hotpink") + xlab(expression(lambda)) +
    ylab(expression(bias^2))
```



f)

Now we need a function for the variance, and as our parameters are already defined above, we only need to fill in the handed out code.

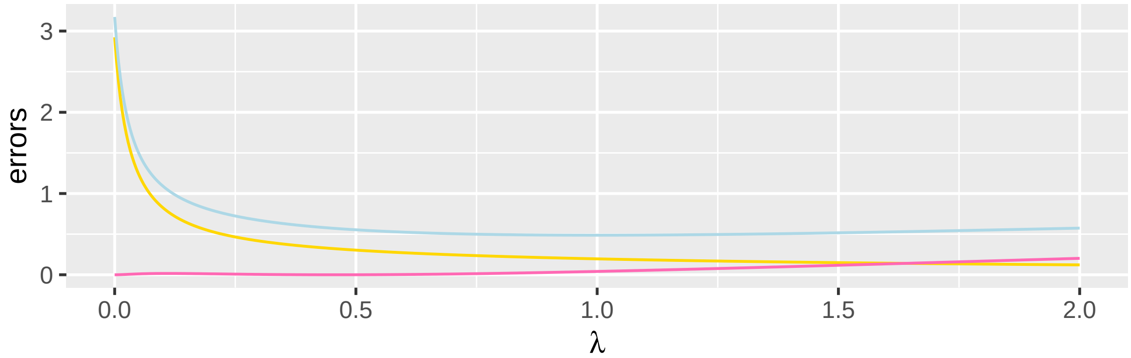
```
variance <- function(lambda, X, x0, sigma) {
    p <- ncol(X)
    I <- diag(p)
    ridge_inv <- solve(t(X) %*% X + lambda * I)
    value <- sigma^2 * t(x0) %*% (I - lambda * ridge_inv) %*% t(ridge_inv) %*% x0
    return(value)
}
lambdas <- seq(0, 2, length.out = 500)
VAR <- rep(NA, length(lambdas))
for (i in seq_along(lambdas)) VAR[i] <- variance(lambdas[i], X, x0, sigma)
dfVar <- data.frame(lambdas = lambdas, var = VAR)
ggplot(dfVar, aes(x = lambdas, y = var)) + geom_line(color = "gold") + xlab(expression(lambda)) +
    ylab("variance")
```



g)

Combining the functions above we are able to calculate the MSE.

```
exp_mse <- sigma^2 + VAR + BIAS

dfError = dfVar
dfError$MSE = exp_mse
dfError$bias = BIAS

ggplot(dfError, aes(x = lambdas)) + geom_line(aes(y = var), color = "gold") + geom_line(aes(y = bias),
    color = "hotpink") + geom_line(aes(y = MSE), color = "lightblue") + xlab(expression(lambda)) +
    ylab("errors")
```



```
# lambdas[which.min(exp_mse)]
```

The lambda that corresponds to the smallest MSE is $\lambda \approx 0.993988$.

Problem 2

We will now look into the data set 'Salaries' from the carData package. We will use this data set to aid Bert-Ernie's choices for his career in academia. Before we start, we first make a descriptive analysis of the data set, by making a pairs plot.

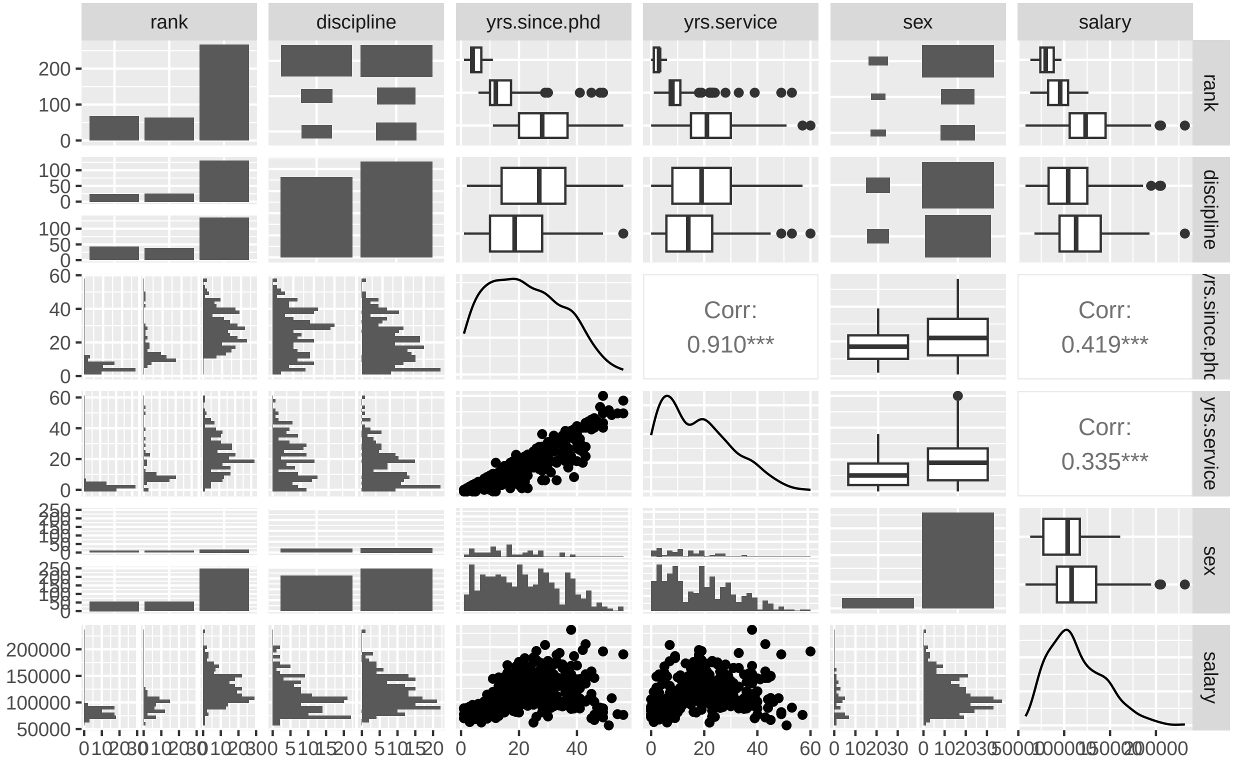


Figure 1: Pairs plot of the academic salary data set.

We make note of a few things: the predictor `sex` seems to contain mostly men, and `rank` contains mostly professors. We also see a large correlation between `yrs.since.phd` and `yrs.service`, which is to be expected, and `rank` and `salary` also seem to be correlated. We might also notice some correlation between `sex` and `rank`, as well as between `sex` and `yrs.since.phd` and `yrs.service`.

We now fit a linear model with all predictors against `salary` as our response variable.

```r
# Fit full model
model1 <- lm(salary ~ ., data = Salaries)
# summary(model1)
```

a)

i)

lm creates two dummy variables: `rankAssocProf` and `rankProf`. These are actually values compared to `AsstProf` as a baseline. This means that if both dummy variables are zero, the individual is an assistant. The coefficient for `rankAssocProf` thus represents an increase in salary in our model for an individual that is an associate professor to an assistant professor, and the case is the same for the `rankProf` coefficient, also compared to an assistant professor. If we had instead chosen `RankProf` as our baseline, these coefficients would probably be negative, as the other ranks have lower salaries compared to a full professor.

ii)

To find the impact that rank has on salary as a whole, we perform an F-test using anova. Here we test against the coefficients of both dummy variables being set to zero.

```
Ftest.model1 = anova(model1)

# Ftest.model1['rank',]$`Pr(>F)`
```

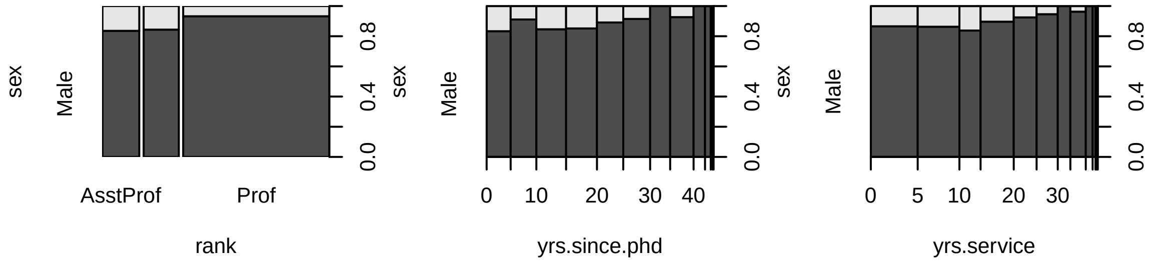
Here we get a p-value of $8.4322731 \times 10^{-47}$, which means that it is highly likely that rank does have an effect on salary, which is to be expected. This is also in line with what we saw in our descriptive analysis.

b)

From the descriptive analysis, we noticed that there was some correlation between `sex` and `rank`, as well as between `sex` and `yrs.since.phd` and `yrs.service`.

To investigate this further, we plot sex against these three categories:

```
sex_model <- lm(salary ~ sex, data = Salaries)
# summary(sex_model)

par(mfrow = c(1, 3))
plot(sex ~ rank, data = Salaries)
plot(sex ~ yrs.since.phd, data = Salaries)
plot(sex ~ yrs.service, data = Salaries)
```



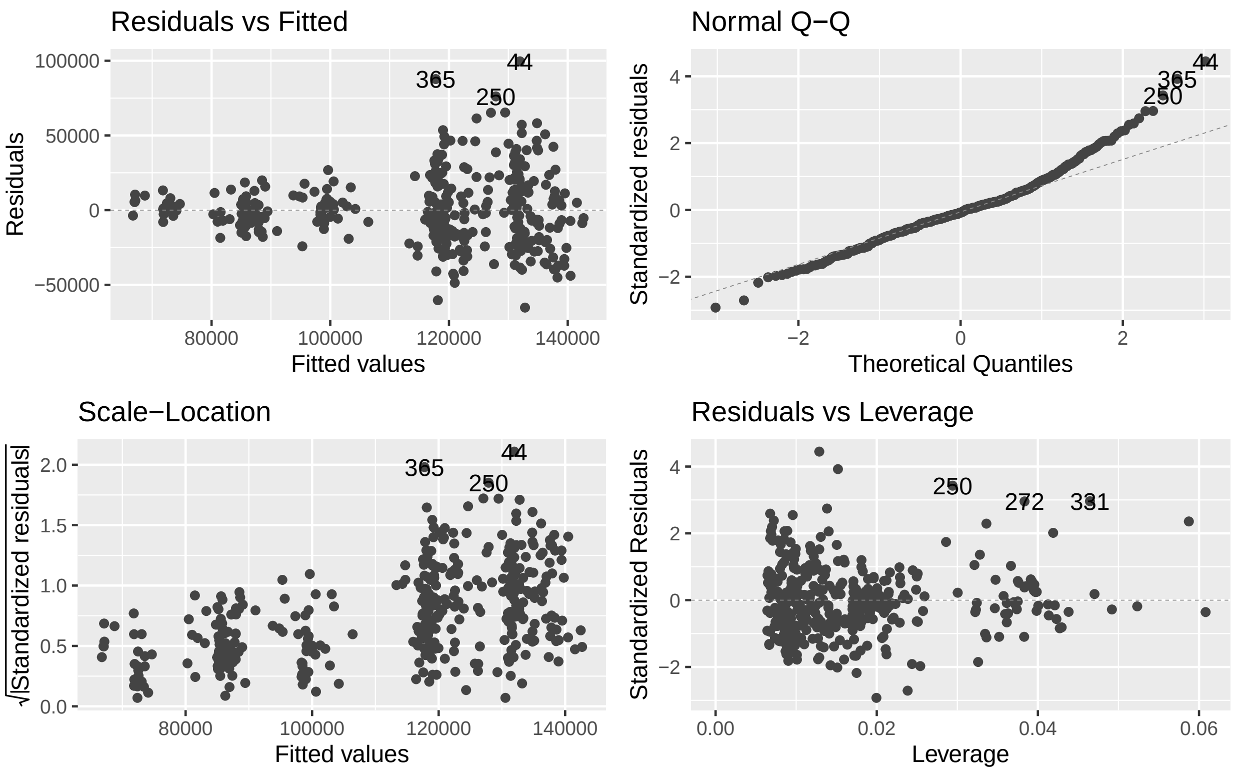
We see that the percentage of male individuals increases with `rank`, `yrs.since.phd` and `yrs.service`. From the descriptive analysis and the linear model, we have also seen that these three categories are associated with a higher salary.

When we fit the linear model against all predictors, the effect of `sex` might therefore be incorporated into other predictors. This gives rise to the change in p-value that we see when fitting the linear model against `sex` as our only predictor.

c)

We are now going to investigate whether the assumptions on `model1` are met or not.

```
library(ggplot2)
library(ggfortify)
autoplot(model1, smooth.colour = NA)
```



i)

From the residuals vs fitted-plot and the scale-location-plot, it looks like the residuals are normally distributed. However, we see that the variance increases with salary, which goes against our assumption of homoscedasticity (all variances are equal).

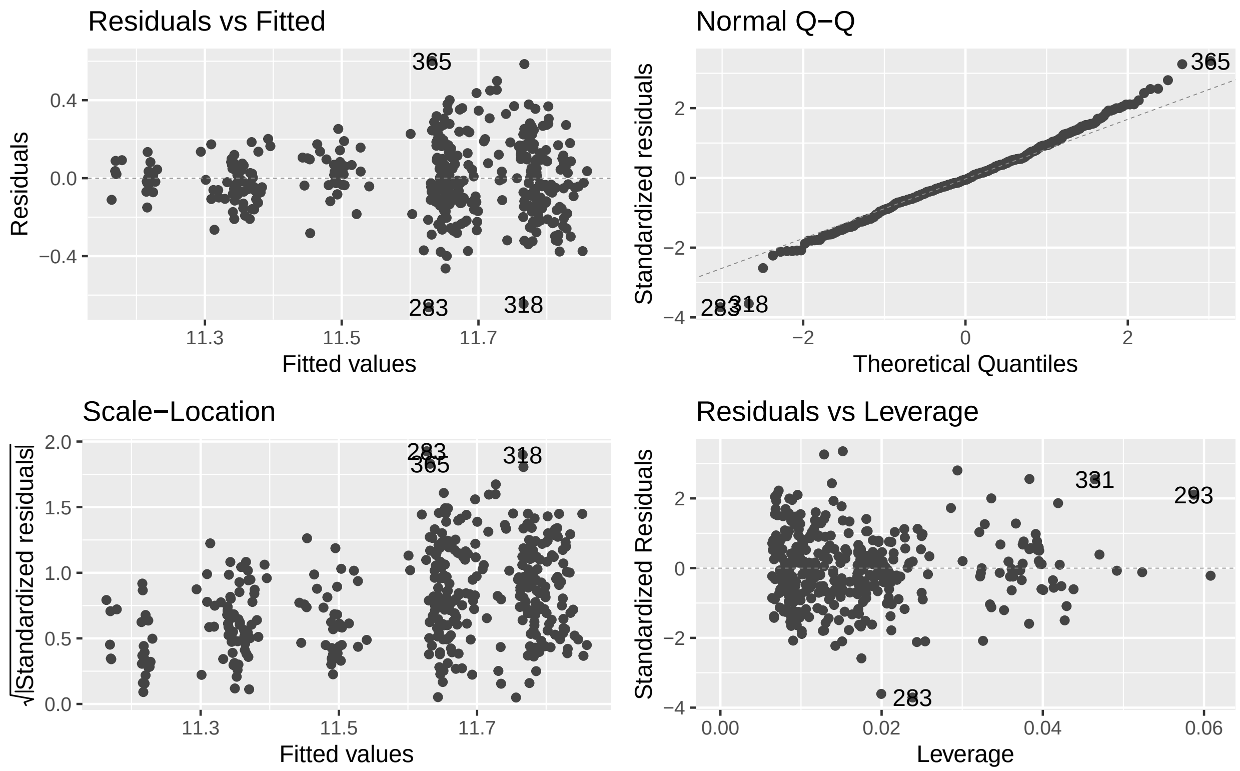
The QQ-plot also does not seem entirely linear on the right side, suggesting that the salary might not be normally distributed.

The residuals vs leverage-plot also suggests some outliers (250, 272, 331).

ii)

We check to see if a log transformation on our response variable improves the assumptions.

```
model2 <- lm(log(salary) ~ ., data = Salaries)
# summary(model2)

autoplot(model2, smooth.colour = NA)
```



In these plots we see that have a straighter line in our QQ-plot, implying that our data seems more normally distributed than previously. Thus, the model assumptions have improved slightly when taking the log of our response and including all predictors.

d)

We now look into the interaction term between `sex` and `yrs.since.phd`.

i)

We start by implementing the model:

```r
model3 <- lm(log(salary) ~ . + sex:yrs.since.phd, data = Salaries)
# summary(model3)
```

ii)

We observe a quite large p-value for the interaction term. Comparing our models with and without the interaction term by the F-test, using anova, we get a p-value of

```r
anova(model2, model3)$"Pr(>F)"[2]
```

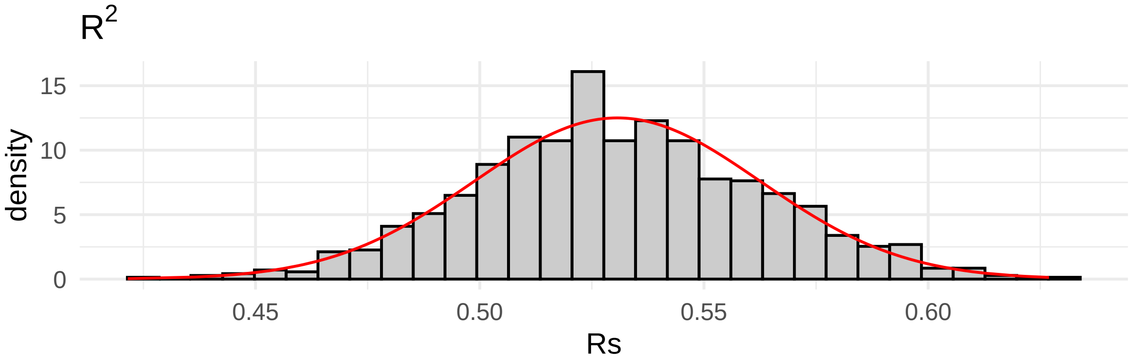
```
## [1] 0.8225433
```

We again observe a quite large p-value, which indicates that the difference in the models' performances is non-significant, and we should stick to our simpler, original model, and reject Bert-Ernie's hypothesis.

e)

To estimate the uncertainty in the R^2-value of `model 1`, we use the bootstrap method.

```
set.seed(4268)

B <- 1000

Rs <- rep(NA, B)

for (b in 1:B) {
    boot.sample <- Salaries[sample(1:dim(Salaries)[1], replace = TRUE), ]
    boot.fit <- lm(log(salary) ~ ., data = boot.sample)
    boot.sum <- summary(boot.fit)
    Rs[b] <- boot.sum$r.squared
}
standard_error <- sd(Rs)
Se <- standard_error

quant <- quantile(Rs, c(0.025, 0.975))

Rs.df <- data.frame(Rs = Rs, norm_den = dnorm(Rs, mean(Rs), sd(Rs)))

ggplot(Rs.df) + geom_histogram(aes(x = Rs, y = ..density..), fill = "grey80", color = "black") +
    ggtitle(expression(R^2)) + geom_line(aes(x = Rs, y = norm_den), color = "red") +
    theme_minimal()
```



In the above chunks we found the following standard error: 0.031911 and the 95% quantile interval: [0.4691091, 0.5945034]. From our histogram of R squared we also see that the values are approximately normally distributed and that our values are rather small, which indicates that our model is not an especially good fit for our data.

f)

We now want to help Bert-Ernie pick a field, by predicting his salary according to the following input:

```
#Code provided in exercise
# Make a data frame containing two new observations, corresponding to
# Bert-Ernie's two possible futures
```

```r
bert_ernie <- data.frame(rank = c("Prof", "Prof"),
                         discipline = c("A", "B"), # Theoretical, applied
                         yrs.since.phd = c(20, 20),
                         yrs.service = c(20, 20),
                         sex = c("Male", "Male"))
# Use the full model to predict his salary
preds <- predict(object = model1,
newdata = bert_ernie,
interval = "prediction",
level = 0.9)
# Check predictions
preds
```

```
##        fit      lwr      upr
## 1 116715.6 79318.04 154113.1
## 2 131133.2 93792.87 168473.5
```

```r
#summary(model1)$coef
```

i)

The only things we changed in the above code snippet was that we changed the arguments interval to "prediction" and level to 0.9. The lower value is 7.9318035×10^4 for the theoretical field, so he should be able to follow his dream.

ii)

Our analytic expression for the correct lower limit is:

$$\text{limit}_{\text{lower}} = \text{E[Bert-Ernie's salary]} - t_{n-p,0.95}\hat{\sigma}\sqrt{1 + x_0^\top (X^\top X)^{-1} x_0}$$

$$n = \text{number of observations} \quad \text{p=number of predictors}$$

Where in our case, $n - p = 391$. As an estimator of the variance, we have used

$$H = X(X^\top X)^{-1}X^\top$$
$$\hat{\epsilon} = (I - H)Y$$
$$\hat{\sigma}^2 = \frac{1}{n-p}\hat{\epsilon}^\top\hat{\epsilon}$$

We have calculated this value in the chunk below:

```r
X <- model.matrix(model1)

n <- dim(X)[1]  #samples
p <- dim(X)[2]  #predictors

x0 <- c(1, 0, 1, 0, 20, 20, 1)  #the values of our sample, written to be compatible with X
Y <- data.matrix(Salaries[, "salary"])  #response
```

```
mu0 <- predict(object = model1, newdata = bert_ernie[1, ])  #expected value
t <- qt(1 - 0.1/2, n - p)  #t-value for one-sided 95% prediction interval

H <- X %*% solve(t(X) %*% X) %*% t(X)  #hat matrix
epsilon_hat <- (diag(n) - H) %*% Y  #residual

sigma_hat <- sqrt(1/(n - p) * t(epsilon_hat) %*% epsilon_hat)  #sd estimator

pred_lwr <- mu0 - t * sigma_hat * sqrt(1 + t(x0) %*% solve(t(X) %*% X) %*% x0)  #lower limit of predict
```

We now get 7.9318035×10^4 as our lower limit for the prediction interval. This is exactly the same as what we got by using `predict` in `R` directly.

Problem 3

In this task we will help the Bigfoot Field Researchers Organization (BFRO) with automating their process for classifying reported sightings of Bigfoots. In the chunk below we use code that was provided to us in order to tidy up our data and make it easier to work with.

a)

In this subproblem we will consider a logistic regression model for our task.

i) Below we fit a logistic regression model and perform classification on our model using a 0.5 cutoff.

```
log_reg_model <- glm(class ~ longitude + latitude + visibility + fur + howl + saw +
    heard, family = binomial(), data = train)
# summary(log_reg_model)$coefficients

prob_log <- predict(log_reg_model, newdata = test, type = "response")
pred_log <- ifelse(prob_log >= 0.5, 1, 0)
obs_log <- sum(pred_log)
total_obs <- length(pred_log)
# obs_log total_obs
```

We see that 441 are clear sightings out of our total sightings, which are 912.

ii) We start with our expression for odds

$$\frac{p}{1-p} = \frac{\frac{1}{1+e^{-U}}}{1-\frac{1}{1+e^{-U}}} = \frac{1}{1+e^{-U}-1} = e^{U}$$

Where

$$U = x^T\beta$$

$$\frac{p}{1-p} = e^{x^T\beta} = e^{\beta_0+x_1\beta_1+\ldots+x_n\beta_n}$$

$$\frac{\text{odds}(Y_i = 1|saw = TRUE)}{\text{odds}(Y_i = 1|saw = FALSE)} = e^{\beta_{saw}} = e^{1.2918} = 3.64$$

So we need to multiply our odds by 3.64 (alternative 5)) if our observation contains the word “saw” compared to when it does not.

b)

In this subproblem we will consider a QDA model for our classification.

i) Below we fit a QDA model and perform classification, again with a 0.5 cutoff.

```r
library(MASS)
QDA_model <- qda(class ~ ., data = train)

predict_QDA = predict(QDA_model, newdata = test, type = "respones")

pred_QDA <- predict_QDA$class
# Can just use the class variable, as the default is a cutoff of 0.5
prob_QDA <- predict_QDA$posterior
# Posterior is a Nx2 matrix where each row is to be interpreted as the
# probability for that observation to be in the class A (first element) or
# class B (second element)
obs_QDA <- sum(as.numeric(as.character(pred_QDA)))
```

We now have 626 clear sightings out of a total of 912 sightings.

ii) For the multiple choice we think point 1 and 4 are true, while 2 and 3 are false. TRUE, FALSE, FALSE, TRUE

c)

Here we will consider a KNN model in order to solve our problem.

i) Below we fit a KNN model with $k = 25$, using the knn function from the class package, in order to again do classification.

```r
library(class)
KNN_model <- knn(train = train, test = test, cl = train$class, k = 25, prob = TRUE)
# prob=True gives us the class probabilities which we will need in the ROC-plot

# Classification:
obs_KNN <- sum(as.numeric(as.character(KNN_model)))

# Below we work with the probabilities so that we can use them in the ROC-plot
index <- which(KNN_model == 0)
prob_KNN <- attributes(KNN_model)$prob
prob_KNN[index] = 1 - prob_KNN[index]
```

In this model, 441 observations were classified as class A.

ii) We could find the optimal k from cross validating when we train our data. This means that we split our training data into a training set and a validation set. We now train our model with multiple different k's and investigate their error with different subsets of the validation data. We then pick the k-value based on error and simplicity.

d)

Now we want to compare the performance of our models.

i) In our situation, we would argue that we are interested in prediction, to automate the classification of reports. For prediction, all our models are relevant, while for inference, logistic regression would be preferred since its hard to evaluate the effect of the predictors in KNN and QDA. Since our goal is prediction, we do not exclude any of the three models.

ii) Below we make confusion matrices for the predictions performed on the test sets in a)-c) and present them

```
library(caret)
conf_log <- table(pred_log, test$class, dnn = c("Prediction", "Reference"))
paste("Logistic regression\n", "sensitivity : ", sensitivity(conf_log), "specificity : ",
    specificity(conf_log))
```

```
## [1] "Logistic regression\n sensitivity :  0.694623655913978 specificity :  0.668903803131991"
```

```
print(conf_log)
```

```
##           Reference
## Prediction   0   1
##          0 323 148
##          1 142 299
```

```
conf_qad <- table(pred_QDA, test$class, dnn = c("Prediction", "Reference"))
paste("QAD\n", "sensitivity : ", sensitivity(conf_qad), "specificity : ", specificity(conf_qad))
```

```
## [1] "QAD\n sensitivity :  0.490322580645161 specificity :  0.870246085011186"
```

```
print(conf_qad)
```

```
##           Reference
## Prediction   0   1
##          0 228  58
##          1 237 389
```

```
conf_knn <- table(KNN_model, test$class, dnn = c("Prediction", "Reference"))
paste("KNN\n", "sensitivity : ", sensitivity(conf_knn), "specificity : ", specificity(conf_knn))
```

```
## [1] "KNN\n sensitivity :  0.83010752688172 specificity :  0.809843400447427"
```

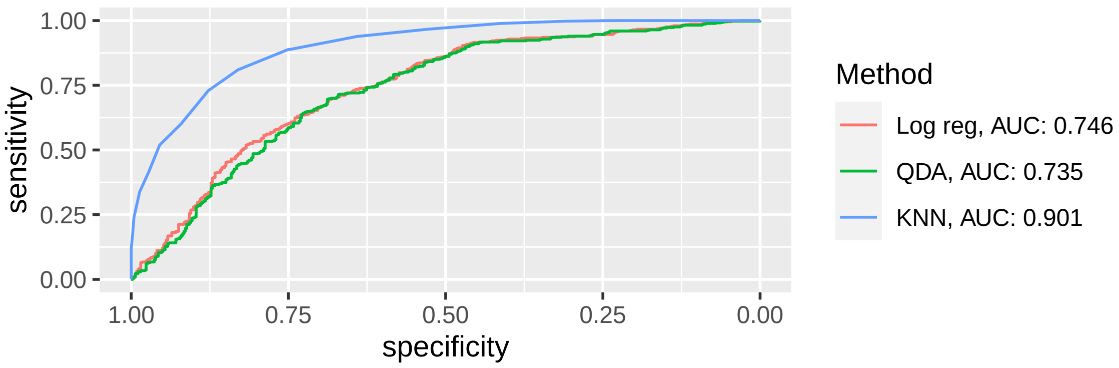
```
print(conf_knn)
```

```
##           Reference
## Prediction   0   1
##          0 386  85
##          1  79 362
```

Sensitivity is true positive, in this case the proportion of B-classifications which are true. Specificity is then the proportion of true A-classifications. Ideally, this is as large as possible.

iii) Below we plot the ROC curves and present the AUC for each of our models.

```r
library(pROC)
roc_log <- roc(response = test$class, predictor = prob_log, direction = "<")

roc_QDA <- roc(response = test$class, predictor = prob_QDA[, 2], direction = "<")
roc_knn <- roc(response = test$class, predictor = prob_KNN, direction = "<")

g = ggroc(list(roc_log, roc_QDA, roc_knn))
g + scale_colour_discrete(name = "Method", labels = c(paste("Log reg, AUC:", round(roc_log$auc,
    digits = 3)), paste("QDA, AUC:", round(roc_QDA$auc, digits = 3)), paste("KNN, AUC:",
    round(roc_knn$auc, digits = 3))))
```



iv) From the plot in iii) we see that the ROC for logistic regression and quadratic discriminant analysis are rather similar, while the ROC for our KNN model with $k = 25$ stands out. This is also confirmed by the AUC-values, where KNN has the largest value. Since we want an AUC-value as large as possible, we would here choose the KNN model.

Problem 4

a)

In this task we will consider the squared error for LOOCV:

$$CV = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_{-i})^2$$

Now we can consider the second term, and aim to arrive at an expression which involves our usual predictions, without LOOCV. We start by using the first hint provided to us, namely that

$$\hat{y}_{-i} = \mathbf{x}_i^\top \beta_{-i}$$

Where β_{-i} denotes the coefficient matrix for LOOCV and is given by our usual formula for the coefficient matrix, i.e.:

$$\beta_{-i} = (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1}\mathbf{X}_{-i}^\top \mathbf{Y}_{-i}$$

Now, using the second hint for the inverse matrix, and the third hint for the last matrix product to arrive at

$$\beta_{-i} = (\mathbf{X}^\top \mathbf{X} - \mathbf{x}_i\mathbf{x}_i^\top)^{-1}(\mathbf{X}\mathbf{Y} - \mathbf{x}_i y_i)$$

If we now insert this in our expression for $\hat{y}_{-i}$ and combine with the Sherman-Morrison formula, which states that

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}u^\top A^{-1}}{1 + v^\top A^{-1}u}$$

By combining with the fact that $h_i = \mathbf{x}_i^\top \mathbf{X}^\top \mathbf{X} \mathbf{x}_i$

$$\begin{aligned}
\hat{y}_{-i} &= \mathbf{x}_i^\top \left((\mathbf{X}^\top \mathbf{X})^{-1} + \frac{\mathbf{X}^\top \mathbf{X}^{-1} \mathbf{x}_i \mathbf{x}_i^\top \mathbf{X}^\top \mathbf{X}}{1 - \mathbf{x}_i^\top \mathbf{X}^\top \mathbf{X} \mathbf{x}_i} \right) (\mathbf{X}^\top \mathbf{Y} - \mathbf{x}_i y_i) \\
&= \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} - \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i y_i + \mathbf{x}_i^\top \frac{\mathbf{X}^\top \mathbf{X}^{-1} \mathbf{x}_i \mathbf{x}_i^\top \mathbf{X}^\top \mathbf{X}}{1 - h_i} \mathbf{X}^\top \mathbf{Y} - \mathbf{x}_i^\top \frac{\mathbf{X}^\top \mathbf{X}^{-1} \mathbf{x}_i \mathbf{x}_i^\top \mathbf{X}^\top \mathbf{X}}{1 - h_i} \mathbf{x}_i y_i \\
&= \hat{y}_i - h_i y_i + \frac{h_i \hat{y}_i}{1 - h_i} - \frac{h_i^2 y_i}{1 - h_i} \\
&= \frac{\hat{y}_i - h_i \hat{y}_i - h_i y_i + h_i^2 y_i + h_i \hat{y}_i - h_i^2 y_i}{1 - h_i} \\
&= \frac{\hat{y}_i - h_i y_i}{1 - h_i}
\end{aligned}$$

If we now finally plug this expression into our expression for the MSE, we get that

$$\begin{aligned}
\text{CV} &= \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \frac{\hat{y}_i - h_i y_i}{1 - h_i} \right)^2 \\
&= \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - h_i y_i - \hat{y}_i + h_i y_i}{1 - h_i} \right)^2 \\
&= \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2 \quad \square
\end{aligned}$$

b)

FALSE, TRUE, FALSE, FALSE