Compulsory Exercise 1 – Solution

TMA4268 Statistical Learning V2023

Emma Skarstein, Kenneth Aase, Daesoo Lee, Stefanie Muff,
Department of Mathematical Sciences, NTNU

Hand out date: February 8, 2023

The submission deadline is: February 23 2023, 23:59h using Blackboard

Introduction

Maximal score is 43 points. **You must score at least 60% to pass the exercise, which is required to take the final exam.** Remember that there are two compulsory exercises/projects, and you must score at least 60% in each of them.

Supervision

We will use the times where we would have lectures and exercises for supervision in the usual lecture rooms.

Supervision hours:

- Monday, February 13, 10:15-12:00 in KJL2
- Thursday, February 16, 08:15-10:00 in EL3 and 10:15-12:00 in KJL2

Remember that there is also the Mattelab forum, and we strongly encourage you to use it for your questions outside the supervision hours – this ensures that all other students benefit from the answers (try to avoid emailing the course staff).

Practical issues (Please read carefully)

- Group size is 2 or 3 - join a group (self enroll) before handing in on Blackboard. We prefer that you do not work alone.
- Please organize yourself via the Mattelab discussion forum (https://mattelab2023v.math.ntnu.no/c/tma4268/6) to find a group. Once you formed a group, log into Blackboard and add yourself to the same group there.
- If you did not find a group even when using Mattelab, you can email Stefanie (stefanie.muff@ntnu.no) and I will try to match you with others that are alone (please use this really only if you have already tried to find a group).
- Remember to write your names and group number on top of your submission file!
- The exercise should be handed in as **one R Markdown file and a pdf-compiled version** of the R Markdown file (if you are not able to produce a pdf-file directly please make an html-file, open it in your browser and save as pdf - no, not landscape - but portrait please). We will read the pdf-file and use the Rmd file in case we need to check details in your submission.

- You may want to work through the R Markdown bonus part in the R course (https://digit.ntnu.no/courses/course-v1:NTNU+IMF001+2020/about)
- In the R-chunks please use both `echo=TRUE` and `eval=TRUE` to make it simpler for us to read and grade.
- Please do not include all the text from this file (that you are reading now) - we want your R code, plots and written solutions - use the template from the course page (https://wiki.math.ntnu.no/tma4268/2023v/subpage6).
- Please **not more than 12 pages** in your pdf-file! (This is a request, not a requirement.)
- Please save us time and **do not submit word or zip**, and do not submit only the Rmd. This only results in extra work for us!

R packages

You need to install the following packages in R to run the code in this file. It is of course also possible to use more or different packages.

```r
install.packages("knitr")     # probably already installed
install.packages("rmarkdown") # probably already installed
install.packages("ggplot2")   # plotting with ggplot2
install.packages("dplyr")     # for data cleaning and preparation
install.packages("tidyr")     # also data preparation
install.packages("carData")   # dataset
install.packages("class")     # for KNN
install.packages("pROC")      # calculate roc
install.packages("plotROC")   # plot roc
install.packages("ggmosaic")  # mosaic plot
```

Multiple/single choice problems

There will be a few *multiple and single choice questions*. This is how these will be graded:

- **Multiple choice questions (2P)**: There are four choices, and each of them can be TRUE or FALSE. If you make one mistake (either wrongly mark an option as TRUE/FALSE) you get 1P, if you have two or more mistakes, you get 0P. Your answer should be given as a list of answers, like TRUE, TRUE, FALSE, FALSE, for example.

- **Single choice questions (1P)**: There are several choices, and only *one* of the alternatives is the correct one. You will receive 1P if you choose the correct alternative and 0P if you choose wrong. Only say which option is true (for example (ii)).

Problem 1 (9P)

We consider the following regression problem

$$Y = f(\mathbf{x}) + \varepsilon, \text{ where } \mathrm{E}(\varepsilon) = 0 \text{ and } \mathrm{Var}(\varepsilon) = \sigma^2.$$

Assume now that the true function $f(\mathbf{x})$ is a linear combination of the observed covariates, that is $f(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$, where $\mathbf{x}$ and $\boldsymbol{\beta}$ are both vectors of length $p+1$.

We know that the OLS estimator in this case is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$, with design matrix $\mathbf{X}$ and response vector $\mathbf{Y}$. In this task we will look at a competing estimator $\widetilde{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{Y}$ (this is called the ridge regression estimator), where $\lambda \geq 0$ is a constant tuning parameter that controls the bias-variance trade-off. Observe that for $\lambda = 0$ the ridge regression estimator is equivalent to $\hat{\boldsymbol{\beta}}$.

We will first derive mathematical formulas for the bias and variance of $\widetilde{\boldsymbol{\beta}}$ and then we will plot these curves in R.

a) (1P)

Find the expected value and the variance-covariance matrix of $\widetilde{\boldsymbol{\beta}}$.

Solution:

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

$$\begin{aligned} \text{Var}(\widetilde{\boldsymbol{\beta}}) &= \text{Var}[(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}] \\ &= (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\text{Var}(\mathbf{Y})\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1} \\ &= \sigma^2(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1} \end{aligned}$$

b) (2P)

Let $\widetilde{f}(\mathbf{x}_0) = \mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}$ be the prediction at a new covariate vector $\mathbf{x}_0$. Using a), find the expected value and variance for $\widetilde{f}(\mathbf{x}_0) = \mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}$.

Solution:

$$E(\mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}) = \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$$

and

$$\text{Var}(\mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}) = \sigma^2\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{x}_0$$

c) (1P)

Explain with words how we can interpret the three terms: bias, variance and irreducible error.

Solution:

- First term: irreducible error, σ^2 and is always present unless we have measurements without error. This term cannot be reduced regardless how well our statistical model fits the data.
- Second term: variance of the prediction at x_0 or the expected deviation around the mean at x_0. If the variance is high, there is large uncertainty associated with the prediction.
- Third term: squared bias. The bias gives an estimate of how much the prediction differs from the true mean $f(x_0)$. If the bias is low the model gives a prediction which is close to the true value.

d) (2P)

Find the expected MSE at $\mathbf{x}_0$, $E[(y_0 - \widetilde{f}(\mathbf{x}_0))^2]$.

Hint: Use that the expected MSE can be decomposed into squared bias, variance and irreducible error, and then move on from there.

Solution:

$$\text{E}[(Y_0 - \widetilde{f}(\mathbf{x}_0))^2] = (\mathbf{x}_0^T\boldsymbol{\beta} - \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta})^2 + \mathbf{x}_0^T\sigma^2(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{x}_0 + \sigma^2$$

Plotting the bias-variance trade-off

The estimator $\widetilde{\boldsymbol{\beta}}$ is a function of the tuning parameter λ, which controls the bias-variance trade-off. Using the decomposition derived in c) we will plot the three elements (bias, variance and irreducible error) using the values in the code chunk below. `values` is a list with the design matrix `X`, the vector $\mathbf{x}_0$ as `x0`, the parameters vector `beta` and the irreducible error `sigma`.

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

X <- values$X
dim(X)
```

```
## [1] 100  81
```

```
x0 <- values$x0
dim(x0)
```

```
## [1] 81  1
```

```
beta <- values$beta
dim(beta)
```

```
## [1] 81  1
```

```
sigma <- values$sigma
sigma
```

```
## [1] 0.5
```

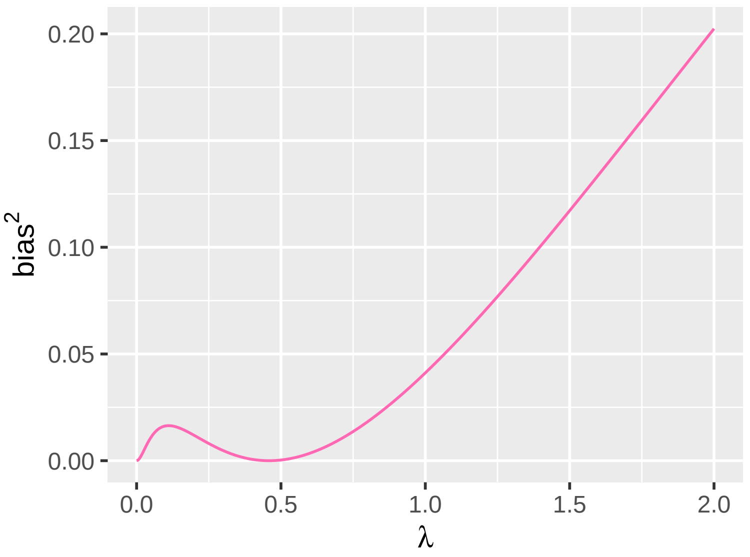
e) (1P)

First we will create the squared bias function (`bias`) which takes as inputs the parameter `lambda`, `X`, `x0`, `beta` and returns the squared bias. You have only to fill the `value <- ...` and run the chunk of code to plot the squared bias, where value is the squared bias as derived in d).

```
library(ggplot2)
bias <- function(lambda, X, x0, beta) {
  p <- ncol(X)
  value <- ...
  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))
```

Solution:

```
library(ggplot2)
bias <- function(lambda, X, x0, beta) {
  p <- ncol(X)
  value <-
    (t(x0) %*% solve(t(X) %*% X + lambda * diag(p)) %*% t(X) %*% X %*% beta -
      t(x0) %*% 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))
```



This is expected, since as λ increase we introduce more bias in the estimator. Maybe we didn't expect this bump in the start.

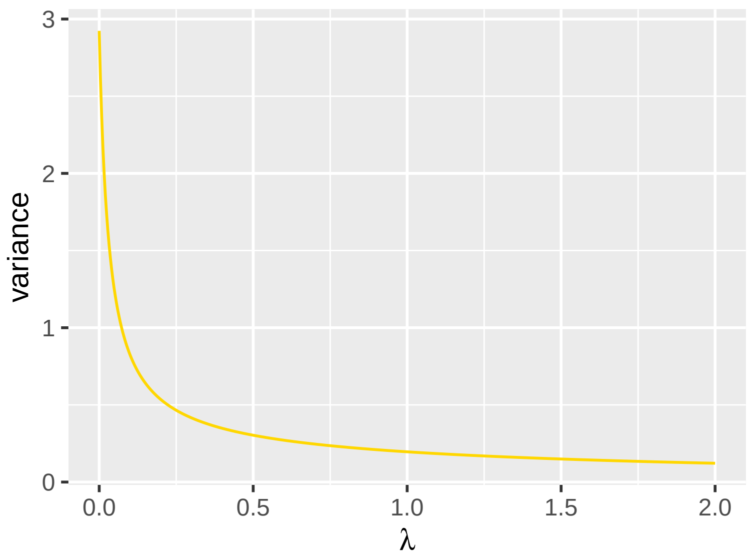
f) (1P)

Now we will create the variance function which takes the same inputs as the squared bias. As in e) you have to fill only the `value <- ...` and run the code to plot the variance.

```
variance <- function(lambda, X, x0, sigma) {
  p <- ncol(X)
  inv <- solve(t(X) %*% X + lambda * diag(p))
  value <- ...
  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")
```

Solution:

```r
variance <- function(lambda, X, x0, sigma) {
  p <- ncol(X)
  inv <- solve(t(X) %*% X + lambda * diag(p))
  value <- sigma^2 * (t(x0) %*% inv %*% t(X) %*% X %*% 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")
```



This is also expected, since with increasing λ the data are given less weight.

g) (1P)

Fill in the `exp_mse` of the following code to calculate the expected MSE for the same lambda values that you plugged in above. Then plot all the components together and find the value of λ which minimizes the expected MSE.

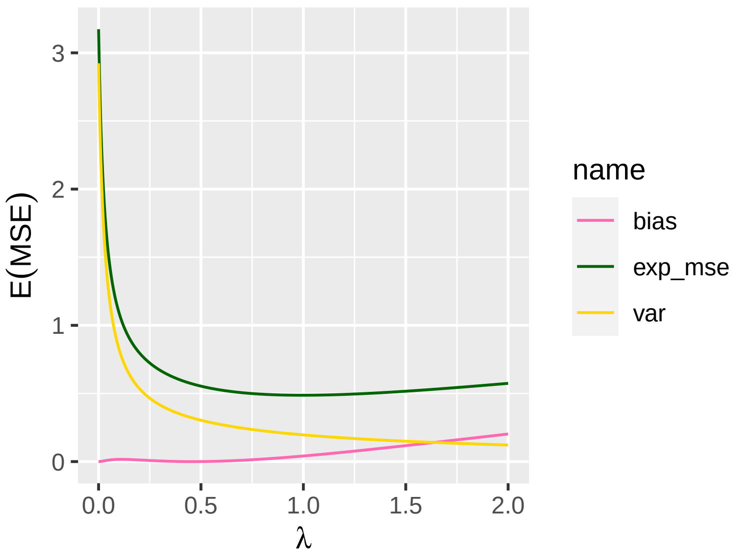
```r
exp_mse <- ...
lambdas[which.min(exp_mse)]
```

Solution:

```r
exp_mse <- BIAS + VAR + sigma^2
lambdas[which.min(exp_mse)]
```

```
## [1] 0.993988
```

```r
library(dplyr)
dfAll <- data.frame(lambda = lambdas,
                    bias = BIAS,
                    var = VAR,
                     exp_mse = exp_mse) %>%
  tidyr::pivot_longer(cols = 2:4)
ggplot(dfAll, aes(x = lambda, y = value, color = name)) +
  geom_line() +
  scale_color_manual(values = c("hotpink", "darkgreen", "gold")) +
  xlab(expression(lambda)) +
  ylab(expression(E(MSE)))
```



Problem 2 (15P)

Bert-Ernie has his eye on a career in academia, but has not really decided on a field yet. Like most academics, his greatest concern is his future salary as a tenured professor. He comes across the `Salaries` dataset from the `carData` `R` package, and decides to perform a statistical analysis to try to find out how he can maximize his future salary as an academic.

To get more information about the covariates in the dataset, you should run `?Salaries` after loading the data. It's also a good idea to do a *descriptive analysis* of the data before fitting any models. A good starting point can be the `ggpairs()` function from the `GGally` package.

```
library(carData)

?Salaries

GGally::ggpairs(Salaries)
```

After doing a descriptive analysis, Bert-Ernie makes a multiple linear regression model with all covariates included, which he calls `model1`.

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

```
##
## Call:
## lm(formula = salary ~ ., data = Salaries)
##
```

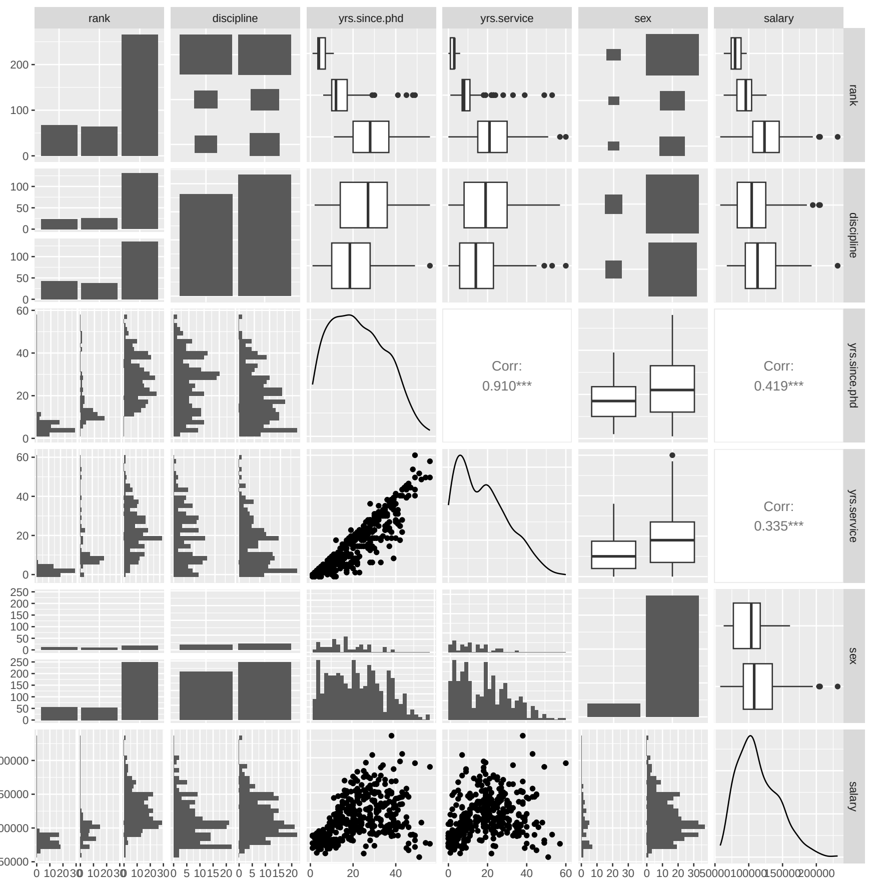


Figure 1: Pairs plot of the academic salary data set.

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -65248 -13211  -1775  10384  99592
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    65955.2     4588.6  14.374  < 2e-16 ***
## rankAssocProf  12907.6     4145.3   3.114  0.00198 **
## rankProf       45066.0     4237.5  10.635  < 2e-16 ***
## disciplineB    14417.6     2342.9   6.154 1.88e-09 ***
## yrs.since.phd    535.1      241.0   2.220  0.02698 *
## yrs.service     -489.5      211.9  -2.310  0.02143 *
## sexMale         4783.5     3858.7   1.240  0.21584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22540 on 390 degrees of freedom
## Multiple R-squared:  0.4547, Adjusted R-squared:  0.4463
## F-statistic:  54.2 on 6 and 390 DF,  p-value: < 2.2e-16
```

a) Categorical variables (2P)

The covariate `rank` is categorical, with the three possible values `AsstProf` (assistant professor), `AssocProf` (associate professor) and `Prof` (full professor).

i) How does the `lm()` function encode for this type of covariate, and what are the interpretations of the `rankAssocProf` and `rankProf` estimates, respectively? (1P)

The output from `summary()` tells us the p-values of `rankAssocProf` and `rankProf`, but not `rank` as a whole.

ii) Perform an appropriate test to check whether there is evidence that `rank` as a whole has an impact on `salary`. The output from `summary()` is not enough to do this. (1P)

Solution:

i) The `lm` function automatically uses dummy variable coding, which means that it adds $L-1$ "dummy columns" to the design matrix, where L is the number of categories in the categorical variable. The dummy variables are indicators that equal 1 for observations from the respective category, and 0 otherwise. The estimates `rankAssocProf` and `rankProf` are thus the estimates of the effect of their corresponding dummy variable. For observations where `rank = AsstProf`, we add nothing (i.e., $0 \cdot 12907.6 + 0 \cdot 45066.0$) to the predicted response, for observations with `rank = AssocProf` we add 12907.6 (i.e., $1 \cdot 12907.6 + 0 \cdot 45066.0$) and when `rank = Prof` we add 45066.0 (i.e., $0 \cdot 12907.6 + 1 \cdot 45066.0$). Thus, we can interpret $\hat{\beta}_{\text{AssocProf}}$ as the estimated effect of being an associate professor compared to being an assistant professor and $\hat{\beta}_{\text{Prof}}$ as the estimated effect of being an full professor compared to being an assistant professor (if we hold the other covariates constant).

ii) The actual test we need is the so-called F-test. We can use the `anova()` function to perform it.

```
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: salary
##                Df     Sum Sq    Mean Sq  F value    Pr(>F)
## rank            2 1.4323e+11 7.1616e+10 140.9788 < 2.2e-16 ***
## discipline      1 1.8430e+10 1.8430e+10  36.2801 3.954e-09 ***
## yrs.since.phd   1 1.6565e+08 1.6565e+08   0.3261   0.56830
## yrs.service     1 2.5763e+09 2.5763e+09   5.0715   0.02488 *
## sex             1 7.8068e+08 7.8068e+08   1.5368   0.21584
## Residuals     390 1.9812e+11 5.0799e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Also a very low p-value if we add rank last
```

We see that there is very strong evidence that `rank` impacts `salary`.

b) Conflicting model results? (2P)

Having heard of the gender wage-gap, Bert-Ernie is surprised to see that `model1` finds no evidence that `sex` has an effect on `salary`. To make sure nothing has gone wrong, he fits a new model with `sex` as the only covariate (along with an intercept).

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

```
##
## Call:
## lm(formula = salary ~ sex, data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57290 -23502  -6828  19710 116455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   101002       4809  21.001  < 2e-16 ***
## sexMale        14088       5065   2.782  0.00567 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30030 on 395 degrees of freedom
## Multiple R-squared:  0.01921,    Adjusted R-squared:  0.01673
## F-statistic: 7.738 on 1 and 395 DF,  p-value: 0.005667
```

The new model finds strong evidence that `sex` has an effect on `salary`. How would you explain the fact that `sex_model` finds evidence of `sex` impacting `salary`, but the full model `model1` does not? Make one or two figures to demonstrate your explanation, with proper figure captions explaining the figure(s). (2P)

Hint: The descriptive analysis might be useful to identify what is going on.

Solution: Confounding between `sex` and other covariates, such as `rank`. In other words, `sex` is in itself highly associated with other covariates that are associated with `salary`. Thus, when the other covariates are included, the explanatory power of `sex` is “soaked up’ ’ by these covariates.

```
# Sex is confounded with various covariates, e.g. rank

# Showing that sex is strongly associated with rank ...
library(ggmosaic)
p1 <- ggplot(data = Salaries) +
  geom_mosaic(aes(x = product(sex), fill = rank)) +
  theme_mosaic()

# ... which is strongly associated with salary
```

```r
p2 <- ggplot(data = Salaries, aes(x = rank, y = salary, fill = rank)) +
  geom_boxplot() +
  theme_minimal()

# Similar plots can be made for other confounded covariates than rank, which
# is OK. It may be easier with a continuous variable

gridExtra::grid.arrange(p1, p2, nrow = 1)
```

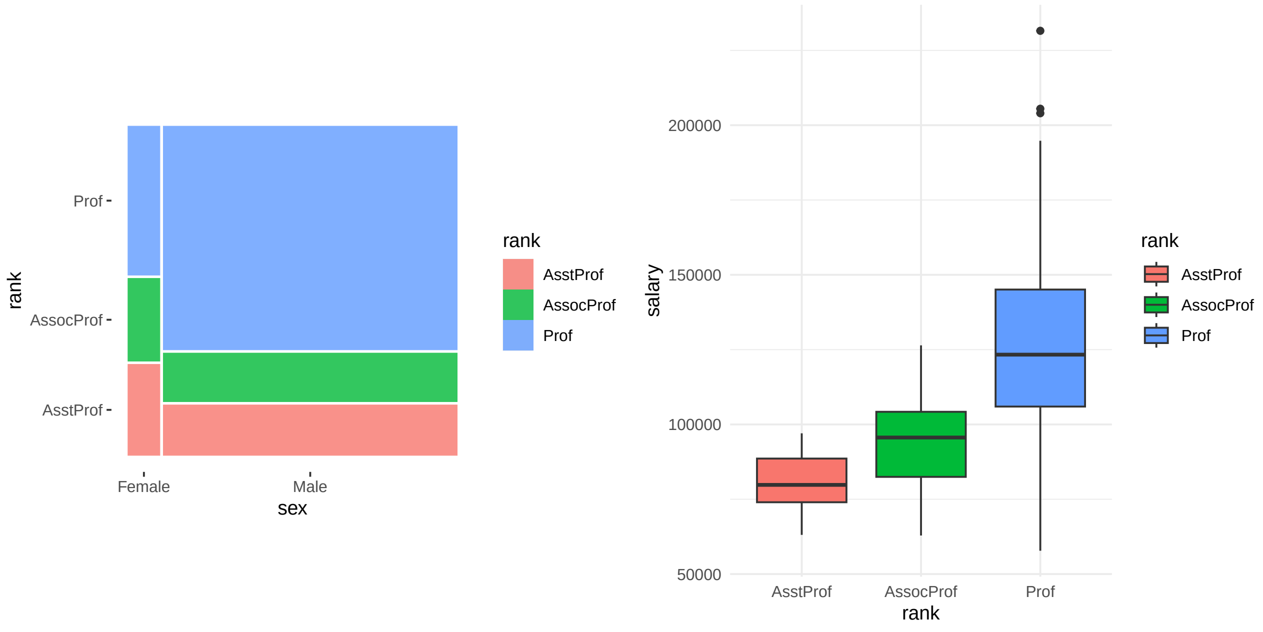


Figure 2: Mosaic plot showing the association between sex and rank (left) and a boxplot of salary for the different ranks (right).

c) Model assumptions and transformation (2P)

Checking the assumptions of `model1`, Bert-Ernie observes that the assumptions of the linear regression model are not fulfilled in `model1`.

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

i) Based on the figure below, which assumptions are not fulfilled? (1P)

ii) Bert-Ernie has heard that sometimes such issues may be solved by transforming the response or covariates of the model, so he decides to try treating the log-transform of `salary` as the response in a new model. Implement the new model, and call it `model2`. Have the model assumptions improved in `model2` compared to `model1`? (1P)

Solution:

i) There is heteroscedasticity (non-constant variance), as seen in the figures on the left - a cone shape. Also, the normality assumption does not seem to hold, the QQ plot does not show a straight line.

ii)

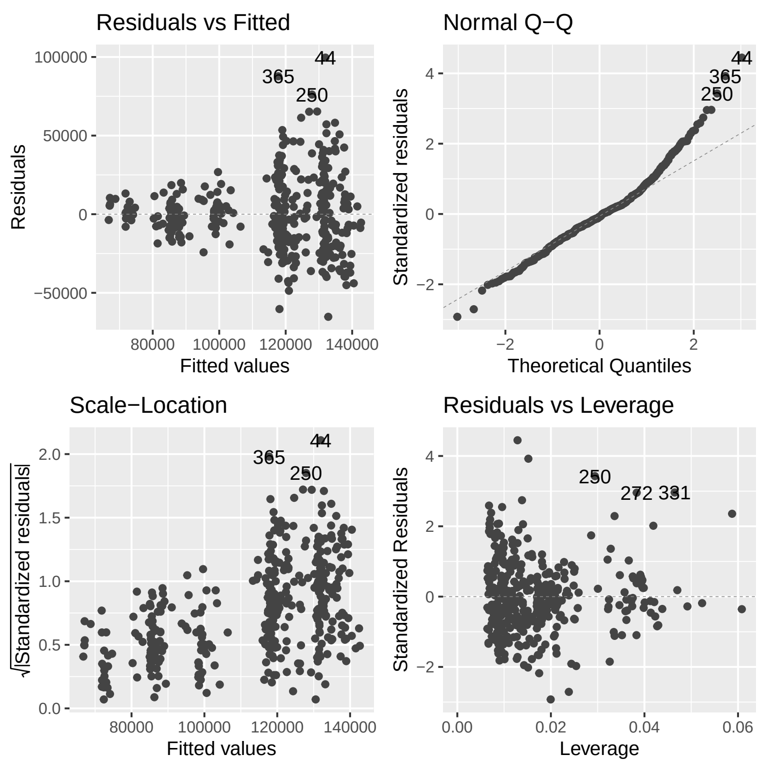


Figure 3: Diagnostic plots for 'model1'.

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

```
##
## Call:
## lm(formula = I(log(salary)) ~ ., data = Salaries)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66236 -0.10813 -0.00914  0.09804  0.60107
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.164144   0.036794 303.425  < 2e-16 ***
## rankAssocProf  0.153787   0.033239   4.627 5.06e-06 ***
## rankProf       0.449463   0.033979  13.228  < 2e-16 ***
## disciplineB    0.131869   0.018786   7.019 9.94e-12 ***
## yrs.since.phd  0.003289   0.001932   1.702   0.0896 .
## yrs.service   -0.003918   0.001699  -2.305   0.0217 *
## sexMale        0.045583   0.030941   1.473   0.1415
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1807 on 390 degrees of freedom
## Multiple R-squared:  0.5248, Adjusted R-squared:  0.5175
## F-statistic: 71.79 on 6 and 390 DF,  p-value: < 2.2e-16
```

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

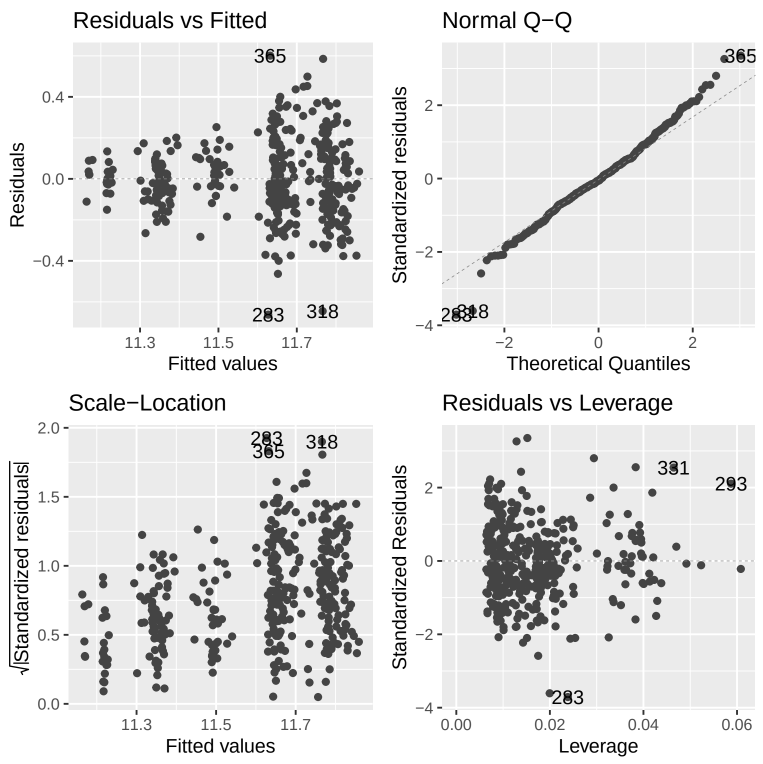


Figure 4: Diagnostic plots for 'model2'.

Yes, things have improved, but the assumptions are still not quite fulfilled. There is still a heteroscedasticity problem, but the problem is less severe than in `model1`. The QQ plot also looks better.

d) Interactions (2P)

Bert-Ernie hypothesizes that the impact of `sex` on salary in academia might be stronger in older generations. While keeping the log-transformed response and all the other covariates in the model he therefore decides to design a model with an interaction term between `sex` and `yrs.since.phd`.

i) Implement the model with the interaction term, and call it `model3` (1P).
ii) Interpret the results from `model3`. Is Bert-Ernie's hypothesis correct? (1P)

Solution:

i)

```r
model3 <- lm(I(log(salary)) ~
               sex * yrs.since.phd + yrs.service + rank + discipline,
             data = Salaries)
```

ii)

```r
summary(model3)
```

```
##
## Call:
```

```
## lm(formula = I(log(salary)) ~ sex * yrs.since.phd + yrs.service +
##     rank + discipline, data = Salaries)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66187 -0.10831 -0.00951  0.09846  0.60143
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           11.1537511  0.0591759 188.485  < 2e-16 ***
## sexMale                0.0574914  0.0614436   0.936   0.3500
## yrs.since.phd          0.0039500  0.0035253   1.120   0.2632
## yrs.service           -0.0038902  0.0017059  -2.280   0.0231 *
## rankAssocProf          0.1528200  0.0335575   4.554 7.05e-06 ***
## rankProf               0.4482679  0.0344343  13.018  < 2e-16 ***
## disciplineB            0.1317818  0.0188133   7.005 1.09e-11 ***
## sexMale:yrs.since.phd -0.0007049  0.0031407  -0.224   0.8225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1809 on 389 degrees of freedom
## Multiple R-squared:  0.5249, Adjusted R-squared:  0.5163
## F-statistic: 61.39 on 7 and 389 DF,  p-value: < 2.2e-16
```

The estimated interaction effect between `sex` and `yrs.since.phd` is very small in magnitude and has a very large p-value, so there is no evidence of an interaction between the two covariates. Furthermore, the estimated interaction effect is negative, so according to `model3` the effect of being male is actually weaker the more years have passed since the academic got their PhD (technically, if someone has had their PhD for more than 82 years then the estimated effect of being male is negative, but this is outside of our range of data, so our model is not valid for such values of `yrs.since.phd`). In summary, no, he cannot conclude that his hypothesis is correct.

e) Bootstrap (4P)

You might have noticed that the R^2 we get from `summary()` is just a point estimate, without any uncertainty associated to it. We can use the bootstrap to estimate the uncertainty of the R^2 in our models. To this end:

(i) Generate 1000 bootstrap samples of the R^2 in `model1`. Use `set.seed(4268)` at the beginning of the bootstrap iterations to ensure that your results are reproducible (2P).
(ii) Plot the respective distribution of the values (0.5P).
(iii) Find the standard error and the 95% quantile interval (1P).
(iv) Interpret what you see (0.5P).

Throughout, show your R code and printouts.

Solution:

i)

```
set.seed(4268)
B <- 1000

# Model 1
sample_R2_model1 <- rep(NA, B)
for (i in 1:B){
  Salaries_boot <- Salaries[sample(seq_len(nrow(Salaries)),
                                      size = nrow(Salaries),
                                      replace = TRUE), ]
  model1_boot <- lm(salary ~ ., data = Salaries_boot)
  sample_R2_model1[i] <- summary(model1_boot)$r.squared
}
```

ii)

```
# Plotting the distribution:
ggplot() +
  geom_density(aes(sample_R2_model1)) +
  theme_minimal() +
  xlab(expression(R^2))
```

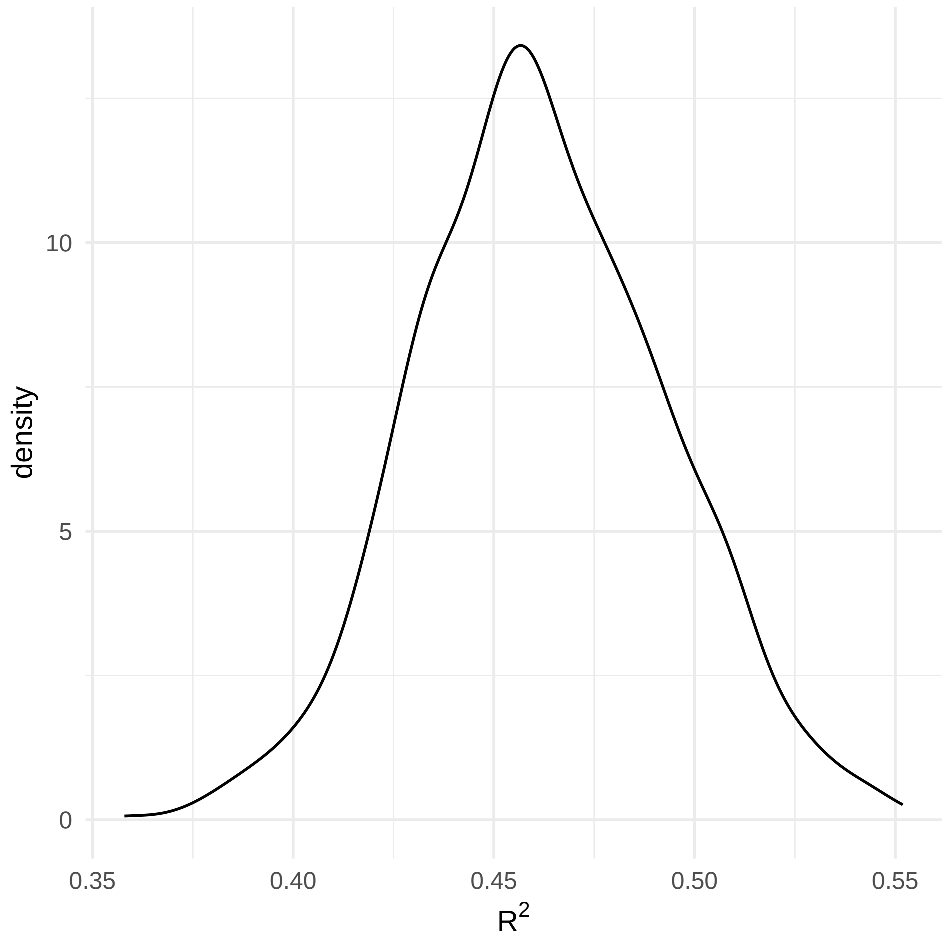


Figure 5: Smoothed density plot the R^2 of ‘model1‘.

```
# or
ggplot() +
  geom_histogram(aes(sample_R2_model1), bins = 25) +
  theme_minimal() +
  xlab(expression(R^2))
```

iii)

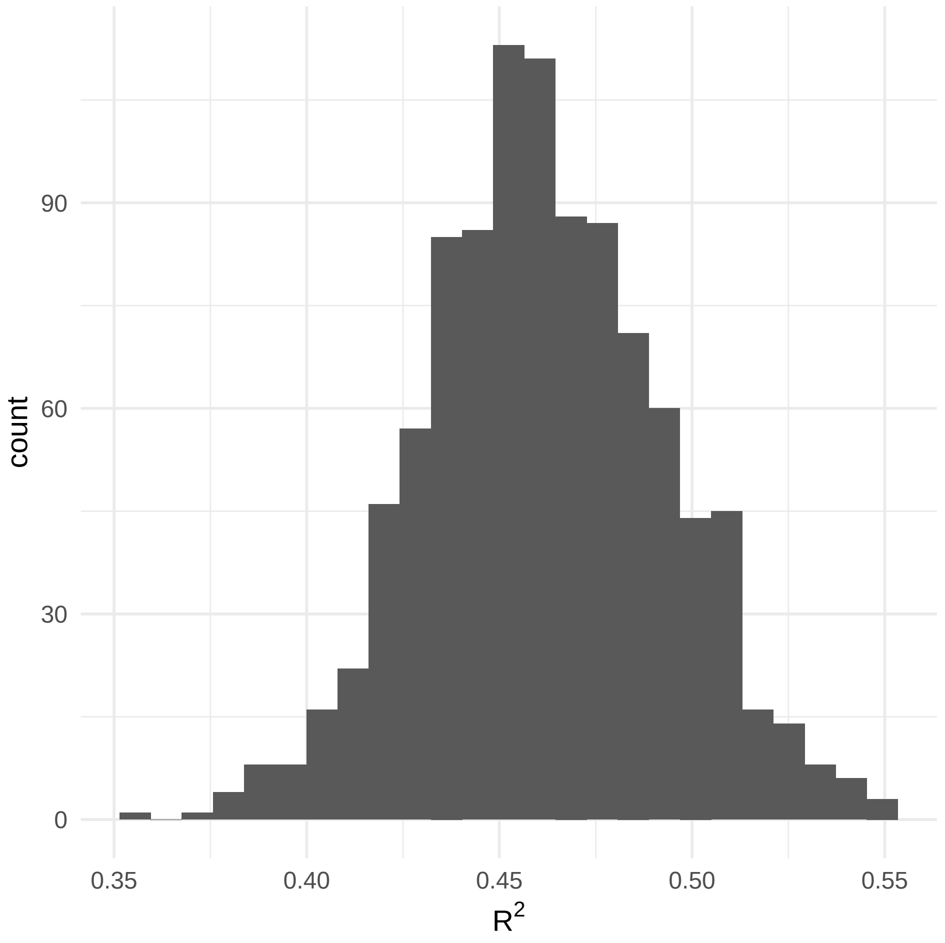


Figure 6: Histogram of the R^2 of ‘model1‘.

```r
# Standard error
sd(sample_R2_model1)
```

```
## [1] 0.03093704
```

```r
# Confidence interval
quantile(sample_R2_model1, c(0.025, 0.975))
```

```
##      2.5%     97.5% 
## 0.4013497 0.5232911
```

iv)

Interpretation: there is uncertainty in R^2, although that is usually not visible/reported in the summary output.

f) Picking a field (3P)

We (and Bert-Ernie) now assume that `model1` is the correct (that is, the data-generating) model. After some character development (namely, all this R coding is giving him a headache) Bert-Ernie has figured out that he would rather work in a theoretical field than an applied field. But, money-conscious as always, he decides that he will follow this dream if and only if his model predicts that 20 years after finishing his PhD he will be earning *at least* 75 000$ every nine months, with 95% certainty. He does not care how much he earns, as long as it is more than 75 000$. He is sure that he will get a permanent position immediately upon finishing his PhD (so his `yrs.since.phd` will equal his `yrs.service`), that he will have become full professor within the 20 years and that his sex will still be male.

```r
# Make a data frame containing two new observations, corresponding to
# Bert-Ernie's two possible futures
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 = "confidence",
                 level = 0.975) # Not 0.95 since we don't care about upper limit
# Check predictions
preds
```

```
##        fit      lwr      upr
## 1 116715.6 110985.0 122446.2
## 2 131133.2 126145.6 136120.8
```

```r
# Check if lower limit for salary in a theoretical field is large enough
preds[1, 2] > 75000
```

```
## [1] TRUE
```

He sees that the lower limit of the computed interval beats 75 000$ by a fair amount, so he breathes a sigh of relief. However, he has made two errors in the call to `predict()`.

i) Correct his mistakes, and make the calculation he was actually interested in, still using `predict()`. Will he still be able to follow his dream after making the correction, or are many sleepless nights of debugging R code looming in his future? (1P)

ii) Write down an analytic expression for the correct lower limit, and implement the analytic expression in R to make sure you get the same result as from `predict()`. (2P)

Hint: see Recommended Exercises Week 3.

R-hint: In the implementation you might have use for the functions `model.matrix()`, `coef()`, `sigma()`, `df.residual()`, `qt()`, `sqrt()`, `solve()` and `t()`, and the matrix multiplication operator `%*%`.

Solution:

i)

Mistakes:

- `interval = "confidence"` should be `interval = "pred"`. He wants to find the prediction interval, not the confidence interval.
- `level = 0.975` should be `level = 0.9`. Since he only cares about the the lower limit of the prediction interval, he is actually interested in a *one-sided* $1-\alpha = 95\%$ prediction interval. The `predict` function does not have a built-in way to do this, but because the t-distribution is symmetric about zero we can just double the α in the two-sided interval to find the limit of the one-sided interval. Bert-Ernie had instead halved the α.

```
preds_fixed <-
  predict(object = model1,
          newdata = bert_ernie,
          interval = "prediction", # Prediction interval, not CI
          level = 0.9) # 0.9 (not 0.975) since we only care about one side!

preds_fixed
```

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

```
preds_fixed[1, 2] > 75000
```

```
## [1] TRUE
```

The limit is still higher than his minimal salary, so he can still make the same decision. However, the lower limit is much closer to his minimum salary.

ii)

Analytic expression for the lower limit of the two-sided 90% prediction interval, which is equivalent to the one-sided (lower) 95% prediction interval:

$$\mathbf{x}_0^\top \hat{\beta} - t_{390}(0.95)\hat{\sigma}\sqrt{1 + \mathbf{x}_0^\top \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{x}_0}$$

or

$$\mathbf{x}_0^\top \hat{\beta} + t_{390}(0.05)\hat{\sigma}\sqrt{1 + \mathbf{x}_0^\top \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{x}_0}$$

or another reasonable rewrite. Here $\mathbf{x}_0$ is the vector of Bert-Ernie's covariate values, $\hat{\beta}$ is the vector of estimated regression coefficients, $t_\nu(\alpha)$ is the α quantile of the Student t distribution with ν degrees of freedom (here $\nu = N - p - 1 = 397 - 6 - 1 = 390$), $\hat{\sigma}$ is the residual standard deviation and $\mathbf{X}$ is the covariate matrix.

Implementation in R:

```
# New observation
x_0 <- c(1, # intercept
         0, # rank = AssocProf: no
         1, # rank = Prof: yes
         0, # discipline = B: no
         20, # yrs.since.phd: 20
         20, # yrs.service: 20
         1) # sex = Male: yes

# Extract various things from model object
X <- model.matrix(model1)
beta_hat <- coef(model1)
sigma_hat <- sigma(model1)
df <- df.residual(model1)

# Prediction int. lower limit - same as what we get from predict()
x_0 %*% beta_hat -
  qt(0.95, df) * sigma_hat * sqrt(1 + x_0 %*% solve(t(X) %*% X) %*% x_0)
```

```
##          [,1]
## [1,] 79318.04
```

```
# or
x_0 %*% beta_hat +
  qt(0.05, df) * sigma_hat * sqrt(1 + x_0 %*% solve(t(X) %*% X) %*% x_0)
```

```
##          [,1]
## [1,] 79318.04
```

Problem 3 (13P)

The Bigfoot Field Researchers Organization (BFRO) is an organization dedicated to investigating the bigfoot mystery, and for years they have been collecting reported sightings in a database. They manually classify their reports into

- Class A: Clear sightings in circumstances where misinterpretation or misidentification of other animals can be ruled out with greater confidence
- Class B: Incidents where a possible bigfoot was observed at a great distance or in poor lighting conditions and incidents in any other circumstance that did not afford a clear view of the subject.

However, they wonder if this can be automated and done by a classification algorithm instead. So in this task, you will set up a few different classification algorithms for this aim, and evaluate their performance.

Feel free to look at the original data, `bigfoot_original` below, however in this task we will be using a slightly simplified version of the data set where we have extracted some variables of interest and deleted observations that are missing any of these variables of interest.

Download the data as below, and run through the provided cleaning/preparation steps.

```
bigfoot_original <- readr::read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/maste

library(dplyr)

# Prepare the data:
bigfoot <- bigfoot_original %>%
```

```r
  # Select the relevant covariates:
  dplyr::select(classification, observed, longitude, latitude, visibility) %>%
  # Remove observations of class C (these are second- or third hand accounts):
  dplyr::filter(classification != "Class C") %>%
  # Turn into 0/1, 1 = Class A, 0 = Class B:
  dplyr::mutate(class = ifelse(classification == "Class A", 1, 0)) %>%
  # Create new indicator variables for some words from the description:
  dplyr::mutate(fur = grepl("fur", observed),
                howl = grepl("howl", observed),
                saw = grepl("saw", observed),
                heard = grepl("heard", observed)) %>%
  # Remove unnecessary variables:
  dplyr::select(-c("classification", "observed")) %>%
  # Remove any rows that contain missing values:
  tidyr::drop_na()
```

The data we will use for this task includes the following variables:

- `class`: the assigned class of the observation, coded as 1 = Class A, 0 = Class B
- `longitude`: longitude of the observation
- `latitude`: latitude of the observation
- `visibility`: estimated visibility at the time and place of observation (higher value means better visibility)
- `fur`: does the report contain the word "fur"? `(TRUE/FALSE)`
- `howl`: does the report contain the word "howl"? `(TRUE/FALSE)`
- `saw`: does the report contain the word "saw"? `(TRUE/FALSE)`
- `heard`: does the report contain the word "heard"? `(TRUE/FALSE)`

For the following tasks, we will be using all of these variables.

The data is split into test and training sets as follows:

Please remember to use the same seed when you split the data into training and test set.

```r
set.seed(2023)

# 70% of the sample size for training set
training_set_size <- floor(0.7 * nrow(bigfoot))

train_ind <- sample(seq_len(nrow(bigfoot)), size = training_set_size)

train <- bigfoot[train_ind, ]
test <- bigfoot[-train_ind, ]
```

a) (2P)

(i) (1P) Fit a **logistic regression** model using the training set, and perform the classification on the test set, using a 0.5 cutoff. How many of the reports were classified as clear sightings (class A, category 1)?

(ii) (1P) Single choice: According to this model, how would the odds that an observation is classified as Class A change if the report contains the word "saw", compared to if it does not? (all other covariates stay the same)

1) We multiply by 1.292.
2) We multiply it with -3.64.
3) We multiply by 0.275.

4) We multiply by 3.64.
5) We add 3.64.
6) We add 1.292.

Solution:

```
formula <- class ~ longitude + latitude + visibility + howl + fur + heard + saw
```

i)

```
# a) Logistic regression
logMod <- glm(formula, family = "binomial", data = train)
predLog <- predict(logMod, newdata = test, type = "response")
logClass <- round(predLog)
table(logClass)
```

```
## logClass
##   0   1
## 471 441
```

There are 441 outcomes classified as class A.

ii)

```
summary(logMod)$coef
```

```
##                 Estimate  Std. Error     z value     Pr(>|z|)
## (Intercept)  0.989051191 0.422047661   2.3434585 1.910589e-02
## longitude   -0.003112218 0.003459854  -0.8995230 3.683741e-01
## latitude    -0.036987829 0.009848634  -3.7556303 1.729058e-04
## visibility  -0.005680978 0.023685869  -0.2398467 8.104491e-01
## howlTRUE    -0.792152432 0.189803190  -4.1735465 2.998945e-05
## furTRUE      0.575172034 0.136327686   4.2190405 2.453442e-05
## heardTRUE   -1.075539728 0.099634234 -10.7948813 3.639324e-27
## sawTRUE      1.291894226 0.097630154  13.2325329 5.693574e-40
```

The correct answer is: 4) We multiply by 3.64.

b) (3P)

(i) (1P) Fit a **QDA** model using the training set, and perform the classification on the test set, using a 0.5 cutoff. How many of the reports were classified as class A?

(ii) (2P) Which statements about linear discriminant analysis and quadratic discriminant analysis are true and which are false? Say for *each* of them if it is true or false.

1) QDA differs from LDA in that it assumes that observations of different classes come from Gaussian distributions with different covariance matrices.
2) LDA is a better choice in cases where there are a lot of observations and so reducing bias is important.
3) Even if the Bayes decision boundary is linear, QDA will be a better choice.
4) QDA assumes that the observations are drawn from a multivariate Gaussian distribution with common mean vector and class-specific covariance matrix

Solution: i)

```
library(MASS)
qdaMod <- qda(formula, data = train)
postQDA <- predict(qdaMod, newdata = test)$posterior
```

```
predQDA <- predict(qdaMod, newdata = test)$class
table(predQDA)
```

```
## predQDA
##   0   1
## 286 626
```

Now many more than with logistic regression, namely 624 of the reports , were classified as clear sightings.

ii) TRUE, FALSE (LDA is better when there are few observations and we want to reduce variance), FALSE (No, the QDA tends to overfit and thus we expect to see higher test error), FALSE (No, the mean vector is also class-specific)

c) (2P)

(i) (1P) Fit a **KNN** model using the training set, and perform the classification on the test set, with $k = 25$ (use the `knn` function from the `class` package).

R-hints: In the `knn()` function set `prob=T` to ensure you get the class probabilities that you then need in d) (change the "`...`" to the correct input):

```
knnMod <- knn(train = ..., test = ..., cl = ..., k = 25, prob = TRUE)
```

(ii) (1P) Explain how you could choose the tuning parameter k in a better way. How does k relate to the bias-variance trade-off in this context?

Solution: i)

```
library(class)
trainKNN <- subset(train, select = -class)
testKNN <- subset(test, select = -class)
knnMod <- knn(train = trainKNN, test = testKNN, cl = train$class,
             k = 25, prob = TRUE)
```

ii) Cross validation can be used to choose k in a better way. Higher k -> lower variance, small k -> high variance.

d) (6P)

We now wish to compare the performance of the three models (logistic regression, QDA and KNN) on this dataset, in order to report back to BFRO on our results.

i) (2P) In this case, are we interested in prediction or inference? What implications would it have for the model choice if we wanted to do prediction, and what implications would it have if we wanted to do inference? In conclusion, does the question of whether our aim is prediction or inference exclude any of the three candidate models in this case? (comment briefly)

ii) (3P) Make confusion matrices for the three predictions performed on the test set in a) - c), and report the confusion matrices, along with sensitivity and specificity. Explain briefly: What does sensitivity and specificity mean? Feel free to explain using the bigfoot example. *Make sure it is clear in the confusion matrix which numbers show the predictions and which show the true values.*

iii) (1P) Present a plot of the ROC curves and calculate the area under the curve (AUC) for each of the classifiers.

iv) (1P) Summarize the performance of the three classifiers with words, based on the evaluations above. Which one would you choose? Justify briefly.

R-hints:

- To obtain $P(y = 1)$ from the `knn()` output you have to be aware that the respective probabilities `attributes(knnMod)$prob` are the success probability for the actual class where the categorization was made. So if you want to get a vector for $P(y = 1)$, you have to use $1 - P(y = 0)$ for the cases where the categorization was 0:

```
probKNN <- ifelse(knnMod == 0,
                  1 - attributes(knnMod)$prob,
                  attributes(knnMod)$prob)
```

- You might find the functions `roc()` and `ggroc()` from the package `pROC` useful, but there are many ways to plot ROC curves.

Solution:

i) Here, we want to do prediction, and interpretability isn't important for us. Therefore we can choose the model based on performance on the validation set. If we wanted to do inference, the logistic model would be the choice.

ii)

```
(conf_log <- table(true = test$class, predicted.log = logClass))
```

```
##     predicted.log
## true   0   1
##    0 323 142
##    1 148 299
```

```
(conf_QDA <- table(true = test$class, predicted.qda = predQDA))
```

```
##     predicted.qda
## true   0   1
##    0 228 237
##    1  58 389
```

```
(conf_KNN <- table(true = test$class, predicted.knn = knnMod))
```

```
##     predicted.knn
## true   0   1
##    0 282 183
##    1 191 256
```

```
# sens_spec calculates sensitivity and specificity from a 2x2 confusion matrix
sens_spec <- function(confusion_matrix, method) {
  sens <- confusion_matrix[2, 2] / (sum(confusion_matrix[2, ]))
  spec <- confusion_matrix[1, 1] / (sum(confusion_matrix[1, ]))
  cat(method, "\n")
  return(c(sensitivity = sens, specificity = spec))
}

sens_spec(conf_log, method = "Logistic regression")
```

```
## Logistic regression
```

```
## sensitivity specificity
##   0.6689038   0.6946237
```

```
sens_spec(conf_QDA, method = "QDA")
```

```
## QDA
## sensitivity specificity
##   0.8702461   0.4903226
```

```
sens_spec(conf_KNN, method = "KNN")
```

```
## KNN
```

```
## sensitivity specificity
##   0.5727069   0.6064516
```

Sensitivity: The true positive rate, what fraction of class A observations are classified as class A. Specificity: The true negative rate, what fraction of class B observations are classified as class B. For example we can note that KNN is better at recognizing class B observations than class A, while QDA is very good at recognizing class A observations but worse than random at recognizing class B.

1P for correct confusion matrix calculation, 1P for correct sens. & spec. calculation, 1P for correct and complete explanation.

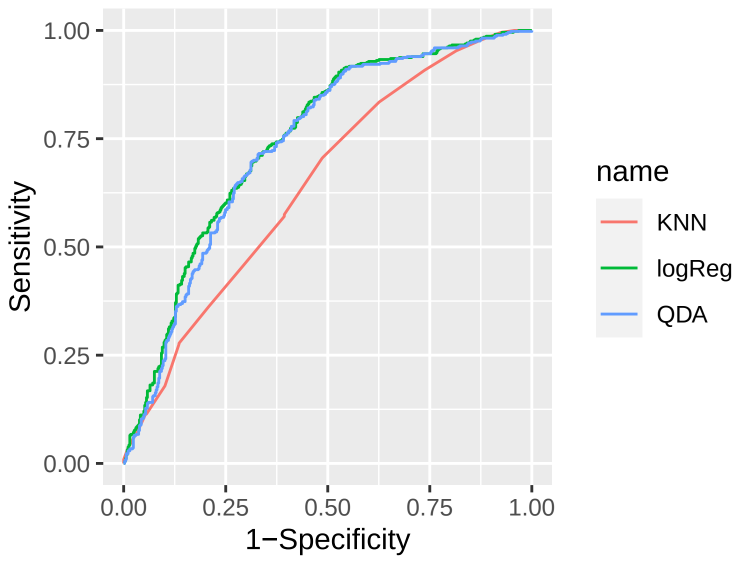
iii)

```
library(pROC)
library(plotROC)

logReg.ROC <- roc(response = test$class, predictor = predLog)
QDA.ROC <- roc(response = test$class, predictor = postQDA[, 2])

# The probabilities in the knnMod output are for the respective categorized
# class, and not directly P(y=1), so we need to use 1-prob if the class was
# categorized as 0. We generate the probKNN vector with the respective correct
# information that we need for the ROC curve:
probKNN <- ifelse(knnMod == 0,
                  1 - attributes(knnMod)$prob,
                  attributes(knnMod)$prob)
KNN.ROC <- roc(response = test$class, predictor = probKNN)

probs <- data.frame(class = test$class,
                    logReg = predLog,
                    QDA = postQDA[, 2],
                    KNN = probKNN)
plProbs <- melt_roc(probs, "class", c("logReg", "QDA", "KNN"))
ggplot(plProbs, aes(d = D, m = M, color = name)) +
  geom_roc(n.cuts = FALSE, size = 0.5) +
  xlab("1-Specificity") + ylab("Sensitivity")
```



```r
aucAll <- data.frame(auc = c(auc(logReg.ROC),  auc(QDA.ROC), auc(KNN.ROC)))
rownames(aucAll) <- c("logReg", "QDA ", "KNN ")
kableExtra::kable(aucAll)
```

	auc
logReg	0.7457603
QDA	0.7353588
KNN	0.6460369

iv) None of them are very good, especially KNN is close to being just as good as random choice. No clear winner between QDA and logistic regression based on the ROC, but QDA has such poor specificity that I think logistic regression is preferable. To improve prediction accuracy, more of the text in the observation report should probably be included.

Problem 4 (6P)

a) (4P)

To calculate the LOOCV statistic when we have N training data points we typically have to fit N models and evaluate the error of each fit on its left out observation. This task can be computationally intensive when N is large. Fortunately, for linear models there is formula for doing this with fitting just the full data model.

Show that for the linear regression model $Y = \mathbf{X}\beta + \varepsilon$ the LOOCV statistic can be computed by the following formula

$$\text{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2 ,$$

where $h_i = \mathbf{x}_i^\top\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}_i$ and $\mathbf{x}_i^\top$ is the ith row of $\mathbf{X}$.

Hints:

1. You need to calculate $\hat{y}_{(-i)}$, which is the predicted value of y when the ith observation is kept out. This can be written as $\hat{y}_{(-i)} = \mathbf{x}_i^\top\hat{\boldsymbol{\beta}}_{(-i)}$.
2. $\mathbf{X}_{(-i)}^\top\mathbf{X}_{(-i)} = \mathbf{X}^\top\mathbf{X} - \mathbf{x}_i\mathbf{x}_i^\top$, where $\mathbf{X}_{(-i)}$ is the $\mathbf{X}$ matrix without the ith row.

3. Analogously $\mathbf{X}_{(-i)}^\top \mathbf{y}_{(-i)} = \mathbf{X}^\top \mathbf{y} - \mathbf{x}_i y_i$.
4. The Sherman–Morrison formula will be useful.

Solution:

The LOOCV statistic is defined as $\text{CV} = \frac{1}{N}\sum_{i=1}^N \left(y_i - \hat{y}_{(-i)}\right)^2$ where $\hat{y}_{(-i)} = \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{(-i)}$

The equation for $\hat{\boldsymbol{\beta}}_{(-i)}$ is the same as for $\hat{\boldsymbol{\beta}}$ but by deleting the ith row of the X matrix and is written as

$$\hat{\boldsymbol{\beta}}_{(-i)} = \left(\mathbf{X}_{(-i)}^\top \mathbf{X}_{(-i)}\right)^{-1} \mathbf{X}_{(-i)}^\top \mathbf{Y}_{(-i)}.$$

From Hint 2 we have that $\mathbf{X}_{(-i)}^\top \mathbf{X}_{(-i)} = \mathbf{X}^\top \mathbf{X} - \mathbf{x}_i \mathbf{x}_i^\top$, and by using the Sherman–Morrison formula (Hint 4) we have that

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

Using Hint 3 we have that $\mathbf{X}_{(-i)}^\top \mathbf{y}_{(-i)} = \mathbf{X}^\top \mathbf{y} - \mathbf{x}_i y_i$ and

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

Finally, by substituting $\hat{\boldsymbol{\beta}}_{(-i)}$ in the equation $\hat{y}_{(-i)} = \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{(-i)}$ and we have that

$$\begin{aligned}
\hat{y}_{(-i)} &= \mathbf{x}_i^\top \hat{\boldsymbol{\beta}} - \frac{\mathbf{x}_i^\top \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{x}_i (y_i - \hat{y}_i)}{1 - h_i} \\
&= \hat{y}_i - \frac{h_i \left(y_i - \hat{y}_i\right)}{1 - h_i}
\end{aligned} \tag{3}$$

and

$$\begin{aligned}
y_i - \hat{y}_{(-i)} &= y_i - \hat{y}_i + \frac{h_i \left(y_i - \hat{y}_i\right)}{1 - h_i} \\
&= (y_i - \hat{y}_i)\left(1 + \frac{h_i}{1 - h_i}\right) \\
&= \frac{y_i - \hat{y}_i}{1 - h_i}
\end{aligned} \tag{4}$$

$\text{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_{(-i)}\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i-\hat{y}_i}{1-h_i}\right)^2.$

b) Multiple choice (2P)

Say for each statement below whether it is true or false.

(i) The LOOCV will lead to more bias, but less variance than 10-fold CV in the estimated prediction error.
(ii) The formula from **a)** is not valid for polynomial regression.
(iii) The formula from **a)** is valid when we use the log transform of the response in linear regression.
(iv) The validation set-approach is the same as 2-fold CV.

Solution:

F, F, T, F.