ISYE6501x Homework 10

Done By: Joel Quek

Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29) has missing values.

1. Use the mean/mode imputation method to impute values for the missing data.
2. Use regression to impute values for the missing data.
3. Use regression with perturbation to impute values for the missing data.
4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using

- (1) the data sets from questions 1,2,3;
- (2) the data that remains after data points with missing values are removed; and
- (3) the data set when a binary variable is introduced to indicate missing values.

Opening the Datset

```
cancer <- read.table("breast-cancer-wisconsin.data.txt", header = FALSE, sep = ",") #, dec = ".")
```

```
head(cancer)
```

A data.frame: 6 × 11

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
	<int>	<int>	<int>	<int>	<int>	<int>	<chr>	<int>	<int>	<int>	<int>
1	1000025	5	1	1	1	2	1	3	1	1	2
2	1002945	5	4	4	5	7	10	3	2	1	2
3	1015425	3	1	1	1	2	2	3	1	1	2
4	1016277	6	8	8	1	3	4	3	7	1	2
5	1017023	4	1	1	3	2	1	3	1	1	2
6	1017122	8	10	10	8	7	10	9	7	1	4

```
print(cancer)
```

```
637  1268952 10 10  7  8  7  1 10 10   3   4
638  1275807  4  2  4  3  2  2  2  1   1   2
639  1277792  4  1  1  1  2  1  1  1   1   2
640  1277792  5  1  1  3  2  1  1  1   1   2
641  1285722  4  1  1  3  2  1  1  1   1   2
642  1288608  3  1  1  1  2  1  2  1   1   2
643  1290203  3  1  1  1  2  1  2  1   1   2
644  1294413  1  1  1  1  2  1  1  1   1   2
645  1299596  2  1  1  1  2  1  1  1   1   2
646  1303489  3  1  1  1  2  1  2  1   1   2
647  1311033  1  2  2  1  2  1  1  1   1   2
648  1311108  1  1  1  3  2  1  1  1   1   2
649  1315807  5 10 10 10 10  2 10 10  10   4
650  1318671  3  1  1  1  2  1  2  1   1   2
651  1319609  3  1  1  2  3  4  1  1   1   2
652  1323477  1  2  1  3  2  1  2  1   1   2
653  1324572  5  1  1  1  2  1  2  2   1   2
654  1324681  4  1  1  1  2  1  2  1   1   2
655  1325159  3  1  1  1  2  1  3  1   1   2
656  1326892  3  1  1  1  2  1  2  1   1   2
657  1330361  5  1  1  1  2  1  2  1   1   2
658  1333877  5  4  5  1  8  1  3  6   1   2
659  1334015  7  8  8  7  3 10  7  2   3   4
660  1334667  1  1  1  1  2  1  1  1   1   2
661  1339781  1  1  1  1  2  1  2  1   1   2
662  1339781  4  1  1  1  2  1  3  1   1   2
663 13454352  1  1  3  1  2  1  2  1   1   2
664  1345452  1  1  3  1  2  1  2  1   1   2
```

Studying the data set, it appears that the missing data can be found in V7 and are indicated by ?

1. Use the mean/mode imputation method to impute values for the missing data.

Source: https://www.youtube.com/watch?v=e7-gCZmKvsI

Only mean imputing can be used because it is a numeric variable

```
mean(cancer$V7)
```

```
Warning message in mean.default(cancer$V7):
“argument is not numeric or logical: returning NA”
<NA>
```

I can't calculate mean because the data has '?' inside.

```
cancer$V7[cancer$V7 == "?"] <- NA
```

```
print(cancer)
```

```
599  1333495  3  1  1  1  2    1  2  1   1   2
600  1334659  5  2  4  1  1    1  1  1   1   2
601  1336798  3  1  1  1  2    1  2  1   1   2
602  1344449  1  1  1  1  1    1  2  1   1   2
603  1350568  4  1  1  1  2    1  2  1   1   2
604  1352663  5  4  6  8  4    1  8 10   1   4
605   188336  5  3  2  8  5   10  8  1   2   4
606   352431 10  5 10  3  5    8  7  8   3   4
607   353098  4  1  1  2  2    1  1  1   1   2
608   411453  1  1  1  1  2    1  1  1   1   2
609   557583  5 10 10 10 10   10 10  1   1   4
610   636375  5  1  1  1  2    1  1  1   1   2
611   736150 10  4  3 10  3   10  7  1   2   4
612   803531  5 10 10 10  5    2  8  5   1   4
613   822829  8 10 10 10  6   10 10 10  10   4
614  1016634  2  3  1  1  2    1  2  1   1   2
615  1031608  2  1  1  1  1    1  2  1   1   2
616  1041043  4  1  3  1  2    1  2  1   1   2
617  1042252  3  1  1  1  2    1  2  1   1   2
618  1057067  1  1  1  1  1 <NA>  1  1   1   2
619  1061990  4  1  1  1  2    1  2  1   1   2
620  1073836  5  1  1  1  2    1  2  1   1   2
621  1083817  3  1  1  1  2    1  2  1   1   2
622  1096352  6  3  3  3  3    2  6  1   1   2
623  1140597  7  1  2  3  2    1  2  1   1   2
624  1149548  1  1  1  1  2    1  1  1   1   2
```

```
cancer$V7<-as.integer(cancer$V7)
```

Mean Imputing

```
cancer_mean_impute <- cancer
```

```
meanV7 <- mean(cancer$V7, na.rm = TRUE) # remove NA
meanV7
```

3.54465592972182

```
cancer_mean_impute[is.na(cancer_mean_impute$V7), "V7"] <- meanV7
```

```
print(cancer_mean_impute)
```

```
684   466906  1  1  1  1  2  1.000000  1  1   1   2
685   466906  1  1  1  1  2  1.000000  1  1   1   2
686   534555  1  1  1  1  2  1.000000  1  1   1   2
687   536708  1  1  1  1  2  1.000000  1  1   1   2
688   566346  3  1  1  1  2  1.000000  2  3   1   2
689   603148  4  1  1  1  2  1.000000  1  1   1   2
690   654546  1  1  1  1  2  1.000000  1  1   8   2
691   654546  1  1  1  3  2  1.000000  1  1   1   2
692   695091  5 10 10  5  4  5.000000  4  4   1   4
693   714039  3  1  1  1  2  1.000000  1  1   1   2
694   763235  3  1  1  1  2  1.000000  2  1   2   2
695   776715  3  1  1  1  3  2.000000  1  1   1   2
696   841769  2  1  1  1  2  1.000000  1  1   1   2
697   888820  5 10 10  3  7  3.000000  8 10   2   4
698   897471  4  8  6  4  3  4.000000 10  6   1   4
699   897471  4  8  8  5  4  5.000000 10  4   1   4
```

Mode Imputing

```
cancer_mode_impute <- cancer
cancer_mode_impute
```

A data.frame: 699 × 11

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2
1017122	8	10	10	8	7	10	9	7	1	4
1018099	1	1	1	1	2	10	3	1	1	2
1018561	2	1	2	1	2	1	3	1	1	2
1033078	2	1	1	1	2	1	1	1	5	2
1033078	4	2	1	1	2	1	2	1	1	2
1035283	1	1	1	1	1	1	3	1	1	2
1036172	2	1	1	1	2	1	2	1	1	2
1041801	5	3	3	3	2	3	4	4	1	4
1043999	1	1	1	1	2	3	3	1	1	2
1044572	8	7	5	10	7	9	5	5	4	4
1047630	7	4	6	4	6	1	4	3	1	4
1048672	4	1	1	1	2	1	2	1	1	2
1049815	4	1	1	1	2	1	3	1	1	2
1050670	10	7	7	6	4	10	4	1	2	4
1050718	6	1	1	1	2	1	3	1	1	2
1054590	7	3	2	10	5	10	5	4	4	4
1054593	10	5	5	3	6	7	7	10	1	4
1056784	3	1	1	1	2	1	2	1	1	2
1057013	8	4	5	1	2	NA	7	3	1	4
1059552	1	1	1	1	2	1	3	1	1	2

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

# source https://www.tutorialspoint.com/r/r_mean_median_mode.htm
```

```
modeV7 <- getmode(cancer_mode_impute$V7)
modeV7
```

1

```
cancer_mode_impute[is.na(cancer_mode_impute$V7), "V7"] <- modeV7
```

```
print(cancer_mode_impute)
```

```
662  1339781  4  1  1  1  2  1  3  1   1   2
663 13454352  1  1  3  1  2  1  2  1   1   2
664  1345452  1  1  3  1  2  1  2  1   1   2
665  1345593  3  1  1  3  2  1  2  1   1   2
666  1347749  1  1  1  1  2  1  1  1   1   2
667  1347943  5  2  2  2  2  1  1  1   2   2
668  1348851  3  1  1  1  2  1  3  1   1   2
669  1350319  5  7  4  1  6  1  7 10   3   4
670  1350423  5 10 10  8  5  5  7 10   1   4
671  1352848  3 10  7  8  5  8  7  4   1   4
672  1353092  3  2  1  2  2  1  3  1   1   2
673  1354840  2  1  1  1  2  1  3  1   1   2
674  1354840  5  3  2  1  3  1  1  1   1   2
675  1355260  1  1  1  1  2  1  2  1   1   2
676  1365075  4  1  4  1  2  1  1  1   1   2
677  1365328  1  1  2  1  2  1  2  1   1   2
678  1368267  5  1  1  1  2  1  1  1   1   2
679  1368273  1  1  1  1  2  1  1  1   1   2
680  1368882  2  1  1  1  2  1  1  1   1   2
681  1369821 10 10 10 10  5 10 10 10   7   4
682  1371026  5 10 10 10  4 10  5  6   3   4
683  1371920  5  1  1  1  2  1  3  2   1   2
684   466906  1  1  1  1  2  1  1  1   1   2
685   466906  1  1  1  1  2  1  1  1   1   2
686   534555  1  1  1  1  2  1  1  1   1   2
687   536708  1  1  1  1  2  1  1  1   1   2
688   566346  3  1  1  1  2  1  2  3   1   2
689   603148  4  1  1  1  2  1  1  1   1   2
690   654546  1  1  1  1  2  1  1  1   8   2
691   654546  1  1  1  3  2  1  1  1   1   2
692   695091  5 10 10  5  4  5  4  4   1   4
693   714039  3  1  1  1  2  1  1  1   1   2
694   763235  3  1  1  1  2  1  2  1   2   2
695   776715  3  1  1  1  3  2  1  1   1   2
696   841769  2  1  1  1  2  1  1  1   1   2
697   888820  5 10 10  3  7  3  8 10   2   4
698   897471  4  8  6  4  3  4 10  6   1   4
699   897471  4  8  8  5  4  5 10  4   1   4
```

2. Use regression to impute values for the missing data.

Source: https://www.youtube.com/watch?v=ajg1p5ofX0c

```
cancer_reg_impute <- cancer
```

```
which(is.na(cancer_reg_impute$V7))
```

24 · 41 · 140 · 146 · 159 · 165 · 236 · 250 · 276 · 293 · 295 · 298 · 316 · 322 · 412 · 618

```
cor(cancer_reg_impute, use = "complete.obs") # complete.obs only compares non NA values
```

A matrix: 11 × 11 of type dbl

	V1	V2	V3	V4	V5	V6	
V1	1.00000000	-0.05634966	-0.04139605	-0.04222123	-0.06963009	-0.04864387	-0.09
V2	-0.05634966	1.00000000	0.64248149	0.65346999	0.48782872	0.52359604	0.59
V3	-0.04139605	0.64248149	1.00000000	0.90722823	0.70697695	0.75354402	0.69
V4	-0.04222123	0.65346999	0.90722823	1.00000000	0.68594806	0.72246241	0.71
V5	-0.06963009	0.48782872	0.70697695	0.68594806	1.00000000	0.59454777	0.67
V6	-0.04864387	0.52359604	0.75354402	0.72246241	0.59454777	1.00000000	0.58
V7	-0.09924781	0.59309144	0.69170875	0.71387755	0.67064829	0.58571613	1.00
V8	-0.06196640	0.55374245	0.75555916	0.73534350	0.66856706	0.61812790	0.68
V9	-0.05069861	0.53406591	0.71934604	0.71796341	0.60312106	0.62892640	0.58
V10	-0.03797243	0.35095717	0.46075470	0.44125758	0.41889833	0.48058330	0.33
V11	-0.08470103	0.71478993	0.82080144	0.82189095	0.70629414	0.69095816	0.82

V7 and V11 are highly correlated/ high association

```
# Indicator Variable
# Source: https://www.youtube.com/watch?v=ajg1p5ofX0c
```

```
Ind<-function(t)
{
    x<-dim(length(t))
    x[which(!is.na(t))]=1
    x[which(is.na(t))]=0
    return(x)
}
```

```
cancer_reg_impute$I <- Ind(cancer_reg_impute$V7)
```

0 indicates the rows with NA data

```
print(cancer_reg_impute)
```

```
642  1288608  3  1  1  1  2  1  2  1   1   2 1
643  1290203  3  1  1  1  2  1  2  1   1   2 1
644  1294413  1  1  1  1  2  1  1  1   1   2 1
645  1299596  2  1  1  1  2  1  1  1   1   2 1
646  1303489  3  1  1  1  2  1  2  1   1   2 1
647  1311033  1  2  2  1  2  1  1  1   1   2 1
648  1311108  1  1  1  3  2  1  1  1   1   2 1
649  1315807  5 10 10 10 10  2 10 10  10   4 1
650  1318671  3  1  1  1  2  1  2  1   1   2 1
651  1319609  3  1  1  2  3  4  1  1   1   2 1
652  1323477  1  2  1  3  2  1  2  1   1   2 1
653  1324572  5  1  1  1  2  1  2  2   1   2 1
654  1324681  4  1  1  1  2  1  2  1   1   2 1
655  1325159  3  1  1  1  2  1  3  1   1   2 1
656  1326892  3  1  1  1  2  1  2  1   1   2 1
657  1330361  5  1  1  1  2  1  2  1   1   2 1
658  1333877  5  4  5  1  8  1  3  6   1   2 1
659  1334015  7  8  8  7  3 10  7  2   3   4 1
660  1334667  1  1  1  1  2  1  1  1   1   2 1
661  1339781  1  1  1  1  2  1  2  1   1   2 1
662  1339781  4  1  1  1  2  1  3  1   1   2 1
663 13454352  1  1  3  1  2  1  2  1   1   2 1
664  1345452  1  1  3  1  2  1  2  1   1   2 1
665  1345593  3  1  1  3  2  1  2  1   1   2 1
666  1347749  1  1  1  1  2  1  1  1   1   2 1
667  1347943  5  2  2  2  2  1  1  1   2   2 1
668  1348851  3  1  1  1  2  1  3  1   1   2 1
669  1350319  5  7  4  1  6  1  7 10   3   4 1
670  1350423  5 10 10  8  5  5  7 10   1   4 1
671  1352848  3 10  7  8  5  8  7  4   1   4 1
672  1353092  3  2  1  2  2  1  3  1   1   2 1
673  1354840  2  1  1  1  2  1  3  1   1   2 1
674  1354840  5  3  2  1  3  1  1  1   1   2 1
675  1355260  1  1  1  1  2  1  2  1   1   2 1
676  1365075  4  1  4  1  2  1  1  1   1   2 1
677  1365328  1  1  2  1  2  1  2  1   1   2 1
678  1368267  5  1  1  1  2  1  1  1   1   2 1
679  1368273  1  1  1  1  2  1  1  1   1   2 1
680  1368882  2  1  1  1  2  1  1  1   1   2 1
681  1369821 10 10 10 10  5 10 10 10   7   4 1
682  1371026  5 10 10 10  4 10  5  6   3   4 1
683  1371920  5  1  1  1  2  1  3  2   1   2 1
684   466906  1  1  1  1  2  1  1  1   1   2 1
685   466906  1  1  1  1  2  1  1  1   1   2 1
686   534555  1  1  1  1  2  1  1  1   1   2 1
687   536708  1  1  1  1  2  1  1  1   1   2 1
688   566346  3  1  1  1  2  1  2  3   1   2 1
689   603148  4  1  1  1  2  1  1  1   1   2 1
690   654546  1  1  1  1  2  1  1  1   8   2 1
691   654546  1  1  1  3  2  1  1  1   1   2 1
692   695091  5 10 10  5  4  5  4  4   1   4 1
693   714039  3  1  1  1  2  1  1  1   1   2 1
694   763235  3  1  1  1  2  1  2  1   2   2 1
695   776715  3  1  1  1  3  2  1  1   1   2 1
696   841769  2  1  1  1  2  1  1  1   1   2 1
697   888820  5 10 10  3  7  3  8 10   2   4 1
698   897471  4  8  6  4  3  4 10  6   1   4 1
699   897471  4  8  8  5  4  5 10  4   1   4 1
```

Since V7 and V11 are highly correlated, we will fit a linear regression model between these two variables, with V7 as the response variable

```
linear_model <- lm(V7~V11, data=cancer_reg_impute)
summary(linear_model)
```

```
Call:
lm(formula = V7 ~ V11, data = cancer_reg_impute)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6276 -0.3468 -0.3468  1.3724  8.6532 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.93392    0.23811  -20.72   <2e-16 ***
V11          3.14038    0.08315   37.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

V7 = 3.14038(V11)-4.93392

```
Multiple R-squared:  0.6768,	Adjusted R-squared:  0.6764
```

```
for(i in 1:nrow(cancer_reg_impute))
{
    if(cancer_reg_impute$I[i]==0)
        {
        cancer_reg_impute$V7[i]=-4.93392+3.14038*cancer_reg_impute$V11[i]
    }
}


print(cancer_reg_impute)
```

```
642  1288608  3  1  1  1  2  1.00000  2  1   1   2 1
643  1290203  3  1  1  1  2  1.00000  2  1   1   2 1
644  1294413  1  1  1  1  2  1.00000  1  1   1   2 1
645  1299596  2  1  1  1  2  1.00000  1  1   1   2 1
646  1303489  3  1  1  1  2  1.00000  2  1   1   2 1
647  1311033  1  2  2  1  2  1.00000  1  1   1   2 1
648  1311108  1  1  1  3  2  1.00000  1  1   1   2 1
649  1315807  5 10 10 10 10  2.00000 10 10  10   4 1
650  1318671  3  1  1  1  2  1.00000  2  1   1   2 1
651  1319609  3  1  1  2  3  4.00000  1  1   1   2 1
652  1323477  1  2  1  3  2  1.00000  2  1   1   2 1
653  1324572  5  1  1  1  2  1.00000  2  2   1   2 1
654  1324681  4  1  1  1  2  1.00000  2  1   1   2 1
655  1325159  3  1  1  1  2  1.00000  3  1   1   2 1
656  1326892  3  1  1  1  2  1.00000  2  1   1   2 1
657  1330361  5  1  1  1  2  1.00000  2  1   1   2 1
658  1333877  5  4  5  1  8  1.00000  3  6   1   2 1
659  1334015  7  8  8  7  3 10.00000  7  2   3   4 1
660  1334667  1  1  1  1  2  1.00000  1  1   1   2 1
661  1339781  1  1  1  1  2  1.00000  2  1   1   2 1
662  1339781  4  1  1  1  2  1.00000  3  1   1   2 1
663 13454352  1  1  3  1  2  1.00000  2  1   1   2 1
664  1345452  1  1  3  1  2  1.00000  2  1   1   2 1
665  1345593  3  1  1  3  2  1.00000  2  1   1   2 1
666  1347749  1  1  1  1  2  1.00000  1  1   1   2 1
667  1347943  5  2  2  2  2  1.00000  1  1   2   2 1
668  1348851  3  1  1  1  2  1.00000  3  1   1   2 1
669  1350319  5  7  4  1  6  1.00000  7 10   3   4 1
670  1350423  5 10 10  8  5  5.00000  7 10   1   4 1
671  1352848  3 10  7  8  5  8.00000  7  4   1   4 1
672  1353092  3  2  1  2  2  1.00000  3  1   1   2 1
673  1354840  2  1  1  1  2  1.00000  3  1   1   2 1
674  1354840  5  3  2  1  3  1.00000  1  1   1   2 1
675  1355260  1  1  1  1  2  1.00000  2  1   1   2 1
676  1365075  4  1  4  1  2  1.00000  1  1   1   2 1
677  1365328  1  1  2  1  2  1.00000  2  1   1   2 1
678  1368267  5  1  1  1  2  1.00000  1  1   1   2 1
679  1368273  1  1  1  1  2  1.00000  1  1   1   2 1
680  1368882  2  1  1  1  2  1.00000  1  1   1   2 1
681  1369821 10 10 10 10  5 10.00000 10 10   7   4 1
682  1371026  5 10 10 10  4 10.00000  5  6   3   4 1
683  1371920  5  1  1  1  2  1.00000  3  2   1   2 1
684   466906  1  1  1  1  2  1.00000  1  1   1   2 1
685   466906  1  1  1  1  2  1.00000  1  1   1   2 1
686   534555  1  1  1  1  2  1.00000  1  1   1   2 1
687   536708  1  1  1  1  2  1.00000  1  1   1   2 1
688   566346  3  1  1  1  2  1.00000  2  3   1   2 1
689   603148  4  1  1  1  2  1.00000  1  1   1   2 1
690   654546  1  1  1  1  2  1.00000  1  1   8   2 1
691   654546  1  1  1  3  2  1.00000  1  1   1   2 1
692   695091  5 10 10  5  4  5.00000  4  4   1   4 1
693   714039  3  1  1  1  2  1.00000  1  1   1   2 1
694   763235  3  1  1  1  2  1.00000  2  1   2   2 1
695   776715  3  1  1  1  3  2.00000  1  1   1   2 1
696   841769  2  1  1  1  2  1.00000  1  1   1   2 1
697   888820  5 10 10  3  7  3.00000  8 10   2   4 1
698   897471  4  8  6  4  3  4.00000 10  6   1   4 1
699   897471  4  8  8  5  4  5.00000 10  4   1   4 1
```

3. Use regression with perturbation to impute values for the missing data.

Source 1: https://www.youtube.com/watch?v=ghmU7nodhSM

Source 2: https://www.youtube.com/watch?v=Jz97ccAlyj8

We saw earlier that V7 and V11 are highly correlated/associated

We will use the same model

```
summary(linear_model)
```

```
Call:
lm(formula = V7 ~ V11, data = cancer_reg_impute)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6276 -0.3468 -0.3468  1.3724  8.6532 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.93392    0.23811  -20.72   <2e-16 ***
V11          3.14038    0.08315   37.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.073 on 681 degrees of freedom
  (16 observations deleted due to missingness)
Multiple R-squared:  0.6768,	Adjusted R-squared:  0.6764 
F-statistic:  1426 on 1 and 681 DF,  p-value: < 2.2e-16
```

Perturbation Analysis

```
library(mice)
```

```
cancer_perturb <- cancer
```

which uses linear regression with perturbation. The complete function is used to generate a complete dataset with imputed values.

```
impute_model <- mice(cancer_perturb, method='norm.predict')
```

```
 iter imp variable
  1   1  V7
  1   2  V7
  1   3  V7
  1   4  V7
  1   5  V7
  2   1  V7
  2   2  V7
  2   3  V7
  2   4  V7
  2   5  V7
  3   1  V7
  3   2  V7
  3   3  V7
  3   4  V7
  3   5  V7
  4   1  V7
  4   2  V7
  4   3  V7
  4   4  V7
  4   5  V7
  5   1  V7
  5   2  V7
  5   3  V7
  5   4  V7
  5   5  V7
```

```
impute_df <- complete(impute_model)
```

```
impute_df
```

A data.frame: 699 × 11

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>	<int>	<int>	<int>
1000025	5	1	1	1	2	1.000000	3	1	1	2
1002945	5	4	4	5	7	10.000000	3	2	1	2
1015425	3	1	1	1	2	2.000000	3	1	1	2
1016277	6	8	8	1	3	4.000000	3	7	1	2
1017023	4	1	1	3	2	1.000000	3	1	1	2
1017122	8	10	10	8	7	10.000000	9	7	1	4
1018099	1	1	1	1	2	10.000000	3	1	1	2
1018561	2	1	2	1	2	1.000000	3	1	1	2
1033078	2	1	1	1	2	1.000000	1	1	5	2
1033078	4	2	1	1	2	1.000000	2	1	1	2
1035283	1	1	1	1	1	1.000000	3	1	1	2
1036172	2	1	1	1	2	1.000000	2	1	1	2
1041801	5	3	3	3	2	3.000000	4	4	1	4
1043999	1	1	1	1	2	3.000000	3	1	1	2
1044572	8	7	5	10	7	9.000000	5	5	4	4
1047630	7	4	6	4	6	1.000000	4	3	1	4
1048672	4	1	1	1	2	1.000000	2	1	1	2
1049815	4	1	1	1	2	1.000000	3	1	1	2
1050670	10	7	7	6	4	10.000000	4	1	2	4
1050718	6	1	1	1	2	1.000000	3	1	1	2
1054590	7	3	2	10	5	10.000000	5	4	4	4
1054593	10	5	5	3	6	7.000000	7	10	1	4
1056784	3	1	1	1	2	1.000000	2	1	1	2
1057013	8	4	5	1	2	7.191237	7	3	1	4
1059552	1	1	1	1	2	1.000000	3	1	1	2
1065726	5	2	3	4	2	7.000000	3	6	1	4
1066373	3	2	1	1	1	1.000000	2	1	1	2
1066979	5	1	1	1	2	1.000000	2	1	1	2
1067444	2	1	1	1	2	1.000000	2	1	1	2
1070935	1	1	3	1	2	1.000000	1	1	1	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1350423	5	10	10	8	5	5	7	10	1	4
1352848	3	10	7	8	5	8	7	4	1	4
1353092	3	2	1	2	2	1	3	1	1	2
1354840	2	1	1	1	2	1	3	1	1	2
1354840	5	3	2	1	3	1	1	1	1	2
1355260	1	1	1	1	2	1	2	1	1	2
1365075	4	1	4	1	2	1	1	1	1	2
1365328	1	1	2	1	2	1	2	1	1	2
1368267	5	1	1	1	2	1	1	1	1	2
1368273	1	1	1	1	2	1	1	1	1	2
1368882	2	1	1	1	2	1	1	1	1	2
1369821	10	10	10	10	5	10	10	10	7	4
1371026	5	10	10	10	4	10	5	6	3	4
1371920	5	1	1	1	2	1	3	2	1	2
466906	1	1	1	1	2	1	1	1	1	2
466906	1	1	1	1	2	1	1	1	1	2

534555	1	1	1	1	2	1	1	1	1	2
536708	1	1	1	1	2	1	1	1	1	2
566346	3	1	1	1	2	1	2	3	1	2
603148	4	1	1	1	2	1	1	1	1	2
654546	1	1	1	1	2	1	1	1	8	2
654546	1	1	1	3	2	1	1	1	1	2
695091	5	10	10	5	4	5	4	4	1	4
714039	3	1	1	1	2	1	1	1	1	2
763235	3	1	1	1	2	1	2	1	2	2
776715	3	1	1	1	3	2	1	1	1	2
841769	2	1	1	1	2	1	1	1	1	2
888820	5	10	10	3	7	3	8	10	2	4
897471	4	8	6	4	3	4	10	6	1	4
897471	4	8	8	5	4	5	10	4	1	4

```
summary(impute_df)
```

```
       V1                 V2               V3               V4        
 Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000  
 Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000  
 Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207  
 3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000  
 Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000  
       V5               V6               V7               V8        
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000  
 Median : 1.000   Median : 2.000   Median : 1.000   Median : 3.000  
 Mean   : 2.807   Mean   : 3.216   Mean   : 3.515   Mean   : 3.438  
 3rd Qu.: 4.000   3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000  
       V9              V10              V11      
 Min.   : 1.000   Min.   : 1.000   Min.   :2.00  
 1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00  
 Median : 1.000   Median : 1.000   Median :2.00  
 Mean   : 2.867   Mean   : 1.589   Mean   :2.69  
 3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00  
 Max.   :10.000   Max.   :10.000   Max.   :4.00  
```

Question 15.1

Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

I am currently working in a university, so one example of a situation in my profession where optimization would be appropriate is in tutorial/lecture scheduling. Course scheduling is a complex problem that involves multiple constraints such as teaching staff availability, tutorial classroom availability, and student preferences. Optimizing course scheduling can help ensure that classes are offered at optimal times and that students can enroll in the courses they need to complete their degree requirements.

✓ 0s completed at 4:40 PM