

# multivariate\_hw1

Na SeungChan

2024-09-22

## Q1.9

```
x1 <- c(-6, -3, -2, 1, 2, 5, 6, 8)
x2 <- c(-2, -3, 1, -1, 2, 1, 5, 3)
dat <- tibble(x1, x2)
dat
```

```
## # A tibble: 8 x 2
##       x1     x2
##   <dbl> <dbl>
## 1    -6    -2
## 2    -3    -3
## 3    -2     1
## 4     1    -1
## 5     2     2
## 6     5     1
## 7     6     5
## 8     8     3
```

(a)

```
colMeans(dat)
```

```
##       x1     x2
## 1.375 0.750
```

```
var(dat)
```

```
##           x1           x2
## x1 23.41071 10.392857
## x2 10.39286  7.071429
```

```
(7/8)*var(dat)
```

```
##           x1           x2
## x1 20.48438  9.09375
## x2  9.09375  6.18750
```

(b)

```
dat2 = dat %>% transmute(x1_tilda = 0.899*x1 + 0.438*x2,  
                        x2_tilda = 0.899*x2 - 0.438*x1)  
dat2
```

```
## # A tibble: 8 x 2  
##   x1_tilda x2_tilda  
##   <dbl>   <dbl>  
## 1   -6.27    0.83  
## 2   -4.01   -1.38  
## 3   -1.36    1.78  
## 4    0.461  -1.34  
## 5    2.67    0.922  
## 6    4.93   -1.29  
## 7    7.58    1.87  
## 8    8.51   -0.807
```

(c)

```
colMeans(dat2)
```

```
## x1_tilda x2_tilda  
## 1.564625 0.072000
```

```
var(dat2)
```

```
##           x1_tilda    x2_tilda  
## x1_tilda 28.46179827 -0.02807957  
## x2_tilda -0.02807957  2.02171629
```

```
(7/8)*var(dat2)
```

```
##           x1_tilda    x2_tilda  
## x1_tilda 24.90407348 -0.02456963  
## x2_tilda -0.02456963  1.76900175
```

(d)

```
nx1_tilda = 0.899*4 + 0.438*(-2)  
nx2_tilda = 0.899*(-2) - 0.438*4
```

(e)

이 문제의 답은 1페이지에 있음.

## Q1.15

### Data Importing

```
df <- read.delim("T1-7.DAT", sep = "", header = FALSE)
df
```

##		V1	V2	V3	V4	V5	V6
##	1	0.889	1.389	1.555	2.222	1.945	1
##	2	2.813	1.437	0.999	2.312	2.312	2
##	3	1.454	1.091	2.364	2.455	2.909	3
##	4	0.294	0.941	1.059	2.000	1.000	1
##	5	2.727	2.545	2.819	2.727	4.091	0
##	6	3.937	1.250	1.937	2.937	3.749	1
##	7	2.786	1.714	2.357	2.071	2.000	2
##	8	5.231	2.692	1.077	1.846	2.539	1
##	9	1.150	1.100	0.950	2.000	1.000	1
##	10	6.500	2.562	1.749	2.562	2.499	1
##	11	0.800	1.000	2.200	2.267	2.466	2
##	12	4.600	2.000	3.000	2.500	3.400	1
##	13	3.500	1.286	2.714	1.286	1.252	3
##	14	3.444	2.556	2.388	2.389	3.000	1
##	15	4.071	1.000	1.000	2.357	1.572	1
##	16	3.692	1.000	2.538	2.154	2.615	1
##	17	5.167	3.000	1.000	2.667	3.666	0
##	18	0.500	1.000	1.000	2.000	1.000	0
##	19	2.385	1.923	2.539	2.154	2.461	1
##	20	2.100	1.300	1.300	1.800	2.600	1
##	21	5.000	3.250	3.125	2.375	3.375	0
##	22	4.571	1.214	3.286	2.571	3.572	1
##	23	2.733	1.133	2.600	1.933	1.667	1
##	24	4.235	2.294	2.706	2.176	1.883	1
##	25	0.000	1.000	1.941	2.000	2.000	0
##	26	0.750	1.125	3.000	1.875	2.000	3
##	27	3.077	1.462	2.384	2.000	1.846	2
##	28	1.600	1.200	2.950	2.000	2.750	1
##	29	6.273	3.636	1.182	2.545	3.364	0
##	30	2.625	1.000	2.438	1.937	2.062	2
##	31	1.250	1.000	2.000	2.000	3.000	1
##	32	2.437	2.062	1.687	1.875	1.375	1
##	33	4.454	1.727	2.637	2.636	3.546	1
##	34	0.133	1.000	1.000	2.000	1.000	0
##	35	0.222	1.222	1.445	2.000	1.000	1
##	36	2.467	2.667	2.200	1.933	1.800	3
##	37	4.000	1.000	4.000	2.167	2.500	0
##	38	5.385	3.154	2.384	2.846	2.539	1
##	39	0.773	1.000	2.273	1.909	2.091	0
##	40	3.786	2.000	1.571	1.786	1.285	3
##	41	1.923	1.615	1.693	2.000	1.846	1
##	42	1.000	1.333	1.834	2.000	1.917	1
##	43	5.800	2.600	3.000	2.800	4.200	1
##	44	6.062	1.000	1.562	2.375	1.750	0

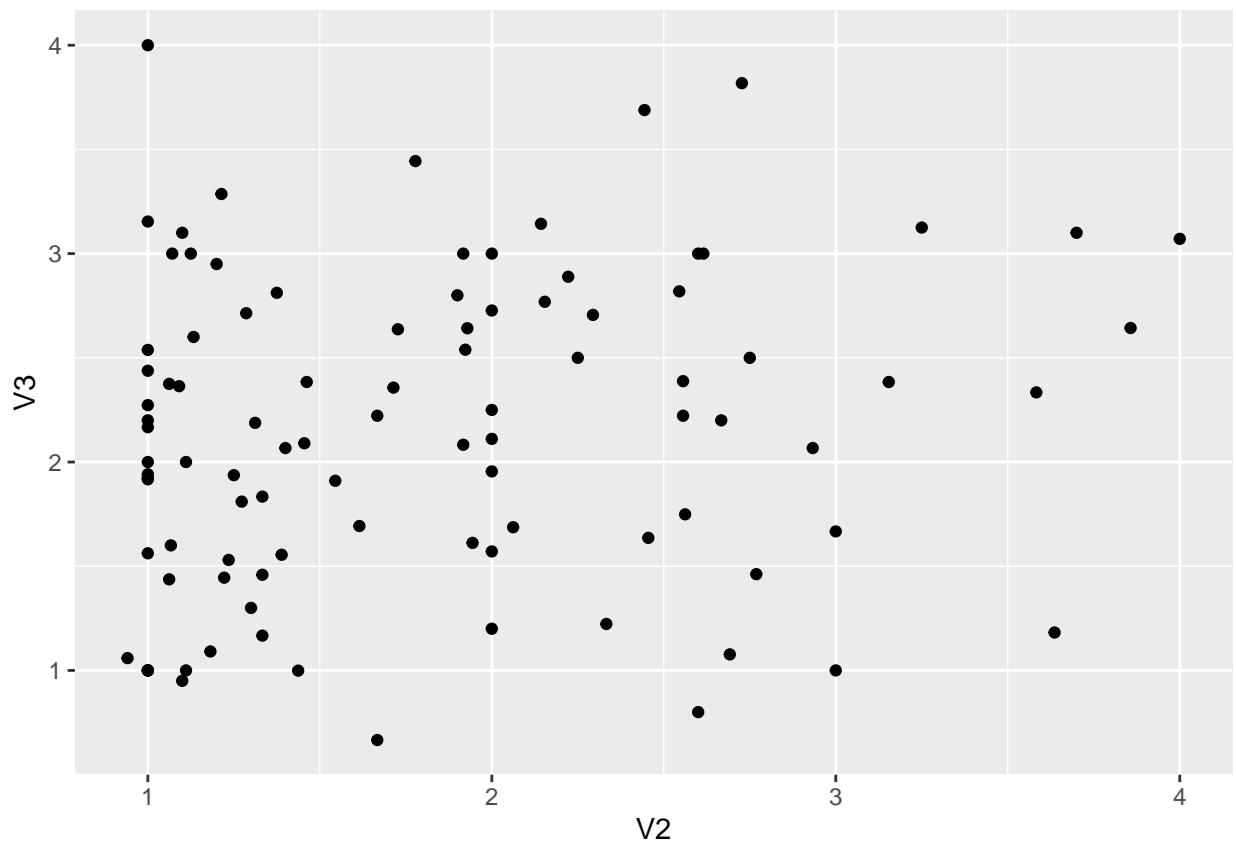
## 45	3.706	1.235	1.530	2.118	2.294	1
## 46	2.444	2.333	1.223	2.444	1.776	3
## 47	6.111	2.222	2.889	2.889	3.555	2
## 48	2.533	1.067	1.600	2.000	1.333	1
## 49	2.167	1.000	2.167	2.000	2.500	1
## 50	2.375	1.062	2.375	2.000	2.125	3
## 51	1.875	1.312	2.188	2.125	2.062	2
## 52	1.750	1.333	1.167	1.750	1.000	1
## 53	7.333	1.333	1.459	1.958	1.542	3
## 54	5.250	1.375	2.812	2.125	2.563	3
## 55	5.182	2.000	2.727	2.818	4.000	2
## 56	1.875	2.000	2.250	2.813	2.437	2
## 57	5.400	2.000	1.200	1.800	1.400	2
## 58	1.154	1.000	1.923	1.846	2.462	1
## 59	6.375	2.250	2.500	2.125	3.000	1
## 60	9.454	2.727	3.818	2.455	3.272	3
## 61	1.000	1.000	1.917	1.833	2.167	1
## 62	1.444	1.111	2.000	2.111	2.000	1
## 63	1.800	1.100	3.100	2.200	2.600	1
## 64	2.818	2.000	1.955	2.045	2.546	2
## 65	10.461	2.154	2.769	2.000	2.923	0
## 66	4.143	1.929	2.642	2.429	3.142	3
## 67	1.227	1.182	1.091	2.227	3.182	1
## 68	5.667	3.000	1.667	2.667	5.000	1
## 69	4.111	2.556	2.222	2.778	3.778	1
## 70	4.444	1.667	2.222	2.000	2.444	0
## 71	3.714	3.857	2.643	2.286	3.285	0
## 72	7.400	3.700	3.100	2.500	4.200	1
## 73	3.182	2.455	1.636	2.273	3.000	1
## 74	5.200	2.600	0.800	1.800	2.000	0
## 75	2.333	1.667	0.666	1.667	2.166	0
## 76	3.333	1.917	2.083	1.917	3.000	1
## 77	5.250	2.750	2.500	2.000	4.000	0
## 78	7.714	4.000	3.071	2.929	4.428	3
## 79	3.846	2.615	3.000	2.692	3.693	2
## 80	2.444	1.111	1.000	2.111	1.667	2
## 81	5.333	1.917	3.000	2.250	1.917	1
## 82	1.556	1.778	3.444	2.667	3.333	1
## 83	3.182	1.545	1.910	2.273	3.000	1
## 84	6.222	2.444	3.689	2.444	3.445	1
## 85	7.231	1.000	3.154	2.308	4.384	2
## 86	3.857	1.071	3.000	2.071	2.286	1
## 87	3.778	1.944	1.612	1.611	1.945	1
## 88	6.000	1.400	2.067	2.267	2.866	2
## 89	2.333	3.583	2.334	2.333	2.667	2
## 90	7.571	2.143	3.143	2.571	3.929	1
## 91	3.667	2.000	2.111	2.778	4.000	3
## 92	3.600	2.933	2.067	2.200	2.867	0
## 93	3.364	1.273	1.810	2.000	2.273	0
## 94	4.100	1.900	2.800	2.000	2.600	2
## 95	0.125	1.062	1.437	1.875	1.563	0
## 96	6.231	2.769	1.462	2.385	4.000	2
## 97	3.000	1.455	2.090	2.273	3.272	2
## 98	0.889	1.000	1.000	2.000	1.000	2

이 데이터에서, V1~V6은 각각 교재의 X1~X6을 지칭하는 어휘에 해당한다.

(a)

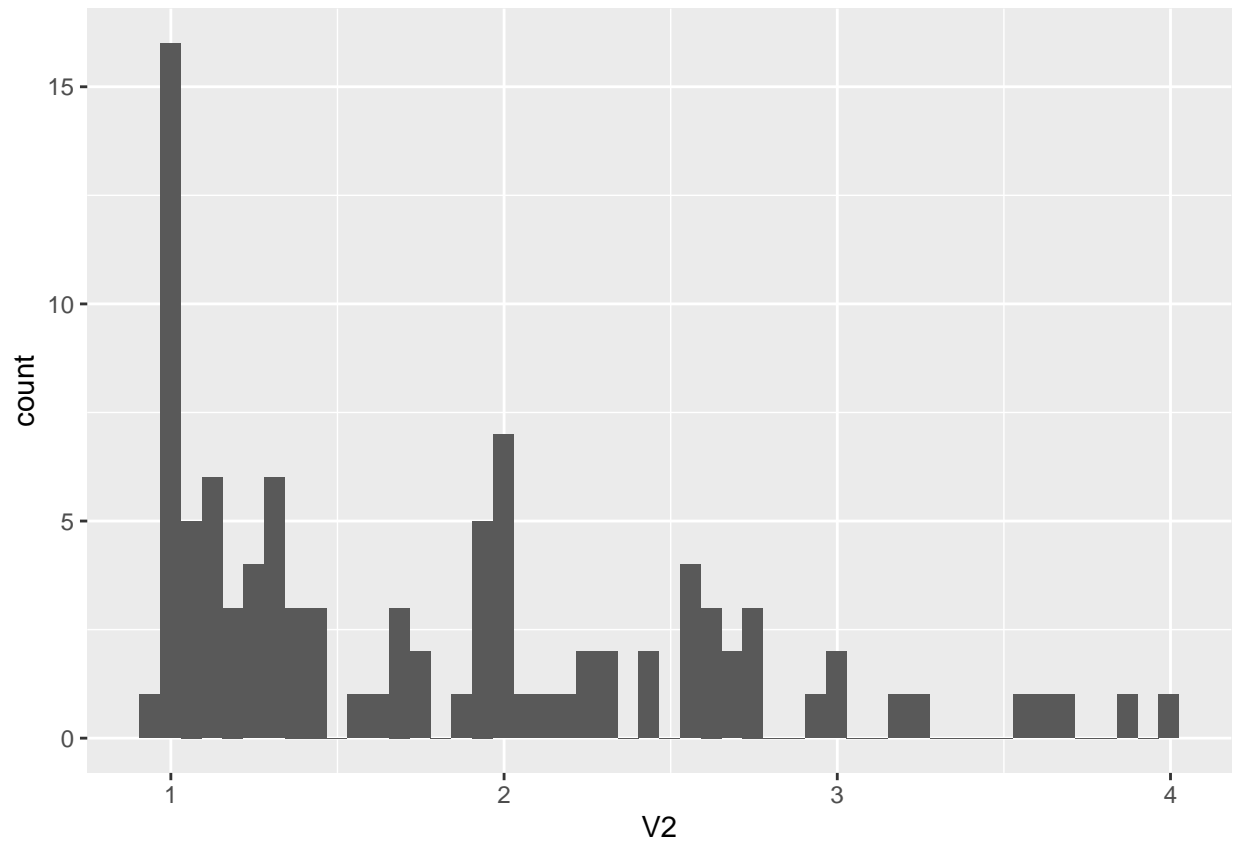
scatterplot

```
ggplot(data = df) +  
  geom_point(mapping = aes(x = V2, y = V3))
```

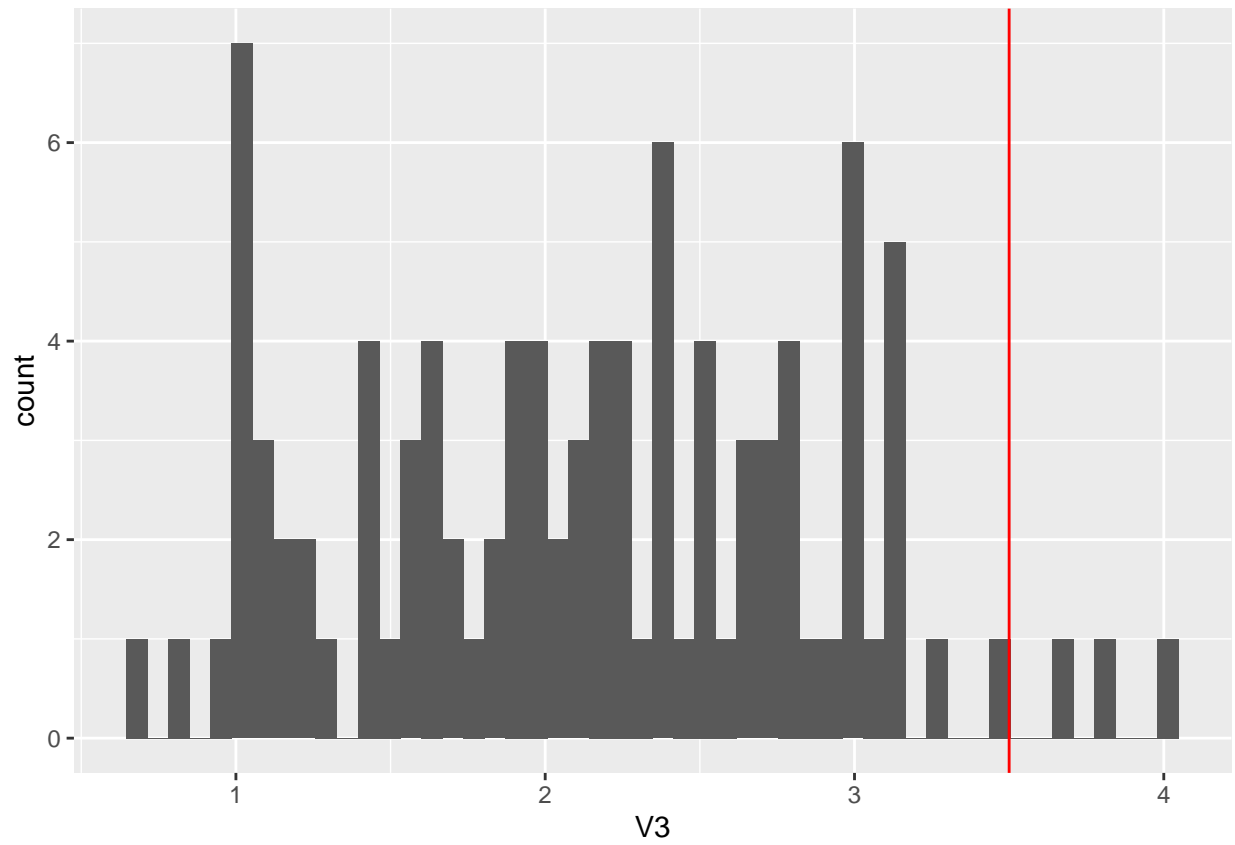


marginal histogram

```
ggplot(data = df) +  
  geom_histogram(mapping = aes(x = V2), bins = 50)
```

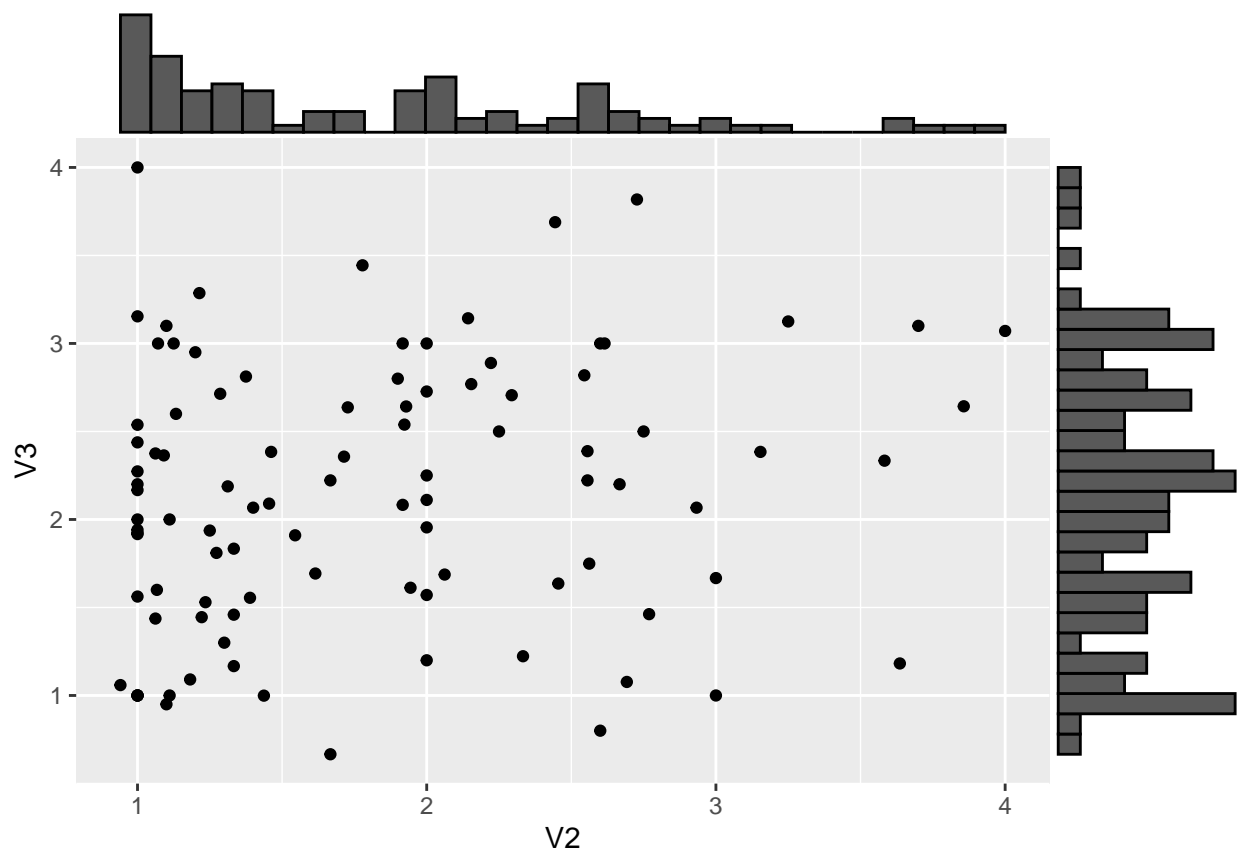


```
ggplot(data = df) +  
  geom_histogram(mapping = aes(x = V3), bins = 50) +  
  geom_vline(xintercept = 3.5, color = 'red')
```



side histogram + scatterplot

```
temp <- ggplot(df, aes(x = V2, y = V3)) + geom_point()
ggMarginal(temp, type = "histogram")
```



V3(x3)의 에러?

히스토그램의  $V3 == 3.5$ 에 그어 둔 빨간 선 오른쪽에도 데이터가 몇 개씩 분포하는 모습을 보인다. 이것들이 outliers인지 의심해 볼 수는 있지만, 데이터의 오류라고 판단하기에는 근거가 지나치게 부족하다.

(b)

sample mean vector

```
colMeans(df)
```

```
##      V1      V2      V3      V4      V5      V6
## 3.542347 1.809357 2.137602 2.209000 2.574827 1.275510
```

sample variance-covariance matrix :  $S_n$

```
(97/98)*var(df)
```

```
##      V1      V2      V3      V4      V5      V6
```



```
## V1 4.6072534 0.9218418 0.58368175 0.27408964 1.063917438 0.156537068
## V2 0.9218418 0.6065679 0.10980144 0.11726018 0.384918205 -0.024598397
## V3 0.5836818 0.1098014 0.56559795 0.08611715 0.344438992 0.109007601
## V4 0.2740896 0.1172602 0.08611715 0.10928245 0.215187224 0.021591837
## V5 1.0639174 0.3849182 0.34443899 0.21518722 0.853374694 -0.008727718
## V6 0.1565371 -0.0245984 0.10900760 0.02159184 -0.008727718 0.852665556
```

sample variance-covariance matrix : unbiased estimator for sigma

```
var(df)
```

```
##          V1          V2          V3          V4          V5          V6
## V1 4.6547509 0.93134537 0.58969909 0.27691531 1.074885659 0.158150852
## V2 0.9313454 0.61282116 0.11093341 0.11846905 0.388886434 -0.024851988
## V3 0.5896991 0.11093341 0.57142886 0.08700496 0.347989910 0.110131391
## V4 0.2769153 0.11846905 0.08700496 0.11040907 0.217405649 0.021814433
## V5 1.0748857 0.38888643 0.34798991 0.21740565 0.862172372 -0.008817694
## V6 0.1581509 -0.02485199 0.11013139 0.02181443 -0.008817694 0.861455923
```

sample correlation matrix

```
cor(df)
```

```
##          V1          V2          V3          V4          V5          V6
## V1 1.00000000 0.55143669 0.3615773 0.38627479 0.53655840 0.07897812
## V2 0.55143669 1.00000000 0.1874625 0.45544470 0.53500626 -0.03420407
## V3 0.36157729 0.18746250 1.00000000 0.34638617 0.49577944 0.15696886
## V4 0.38627479 0.45544470 0.3463862 1.00000000 0.70464665 0.07073348
## V5 0.53655840 0.53500626 0.4957794 0.70464665 1.00000000 -0.01023155
## V6 0.07897812 -0.03420407 0.1569689 0.07073348 -0.01023155 1.00000000
```

상관관계 해석

- 우선, V6(skin reaction)은 전반적으로 타 변수들과 상관관계가 작은 편이다.
- 또한, V5(appetite)는 전반적으로 타 변수들과 상관관계가 큰 편이다.
- 상관관계의 절댓값이 가장 큰 것은 V4(amount of food consumed)와 V5(appetite)였다.