Exam 2 Review — Key

For questions 1–4, use the following data frame (called `snow`) and interpret the code. For full credit (and partial credit), write (1) what the code will return, along with (2) justification for your answer.

ski_hill	pass	acres	vertical
Vail	Epic	5317	3450
Park City	Epic	7300	3226
Big Sky	Ikon	5800	4336
Jackson Hole	Ikon	2500	4139
Taos	Ikon	1294	3281
Bridger Bowl	Powder Alliance	2000	2600
Loveland	Powder Alliance	1800	2210

1.

```
library(tidyverse)
snow %>% group_by(pass) %>%
  summarize(min_size = min(acres)) %>%
  arrange(min_size)
```

```
## # A tibble: 3 x 2
##   pass            min_size
##   <chr>              <dbl>
## 1 Ikon                1294
## 2 Powder Alliance     1800
## 3 Epic                5317
```

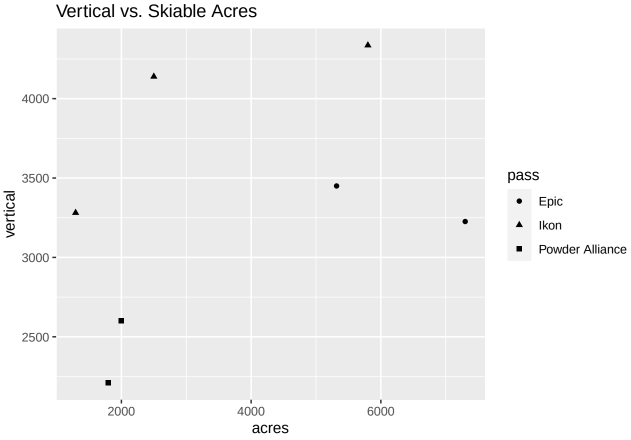
2.

```
for (i in 1:5){
  print(snow$ski_hill[i])
    if (snow$pass[i] == 'Ikon'){
      print("is on the Ikon pass")
    } else {
      print("is not on the Ikon pass")
    }
}
```

```
## [1] "Big Sky"
## [1] "is on the Ikon pass"
## [1] "Jackson Hole"
## [1] "is on the Ikon pass"
## [1] "Taos"
## [1] "is on the Ikon pass"
## [1] "Bridger Bowl"
## [1] "is not on the Ikon pass"
## [1] "Loveland"
## [1] "is not on the Ikon pass"
```

3.

```
snow %>% ggplot(aes(y = vertical, x = acres, shape = pass)) +
  geom_point() + ggtitle('Vertical vs. Skiable Acres')
```



4.

```
ifelse(snow[,3] > 5000 | snow[,4] > 4000,
       "The mountain is big or steep",
       "The mountain is not big or steep")
```

```
##      acres
## [1,] "The mountain is big or steep"
## [2,] "The mountain is big or steep"
## [3,] "The mountain is not big or steep"
## [4,] "The mountain is not big or steep"
## [5,] "The mountain is not big or steep"
## [6,] "The mountain is big or steep"
## [7,] "The mountain is big or steep"
```

For questions 5–8, use the snow data frame and the ticket_price data frame (below). For full credit (and partial credit), write (1) what the code will return, along with (2) justification for your answer.

ski_hill	pass_cost
Big Sky	1,699
Bridger Bowl	900
Discovery	525

5.

```
typeof(ticket_price$pass_cost)
```

```
## [1] "character"
```

6.

```
snow %>% inner_join(ticket_price, by = "ski_hill")
```

```
## # A tibble: 2 x 5
##   ski_hill     pass             acres vertical pass_cost
##   <chr>        <chr>            <dbl>    <dbl> <chr>
## 1 Big Sky      Ikon              5800     4336 1,699
## 2 Bridger Bowl Powder Alliance   2000     2600 900
```

7.

```
ticket_price %>% left_join(snow, by = "ski_hill")
```

```
## # A tibble: 3 x 5
##   ski_hill     pass_cost pass             acres vertical
##   <chr>        <chr>     <chr>            <dbl>    <dbl>
## 1 Big Sky      1,699     Ikon              5800     4336
## 2 Bridger Bowl 900       Powder Alliance   2000     2600
## 3 Discovery    525       <NA>                NA       NA
```

8.

```
library(stringr)
ticket_price %>% mutate(cost = str_replace(pass_cost, ',', ''))
```

```
## # A tibble: 3 x 3
##   ski_hill     pass_cost cost
##   <chr>        <chr>     <chr>
## 1 Big Sky      1,699     1699
## 2 Bridger Bowl 900       900
## 3 Discovery    525       525
```

9.

Finish the function. Either code or prose is acceptable.

```
flip_coin <- function(num_flips){
  # Function to simulate flipping a fair coin
  # ARGS: num_flips - number of times to flip coin
  # RETURNS: vector of coin flips containing "H" or "T"
  sample(c("H", "T"), size = num_flips, replace = TRUE)
}
flip_coin(10)
```

```
##  [1] "H" "H" "H" "T" "T" "T" "H" "T" "H" "H"
```