Python to R practice (with answers)

about the data

the data we generated is called `data.csv`. it has the following columns:

- `id`: unique identifier
- `zip_code`: seattle area ZIP code
- `age`: age in years
- `smoking`: binary variable (0 = does not smoke, 1 = smokes)
- `years_smoked`: number of years smoked (NA if non-smoker)

python code to translate

```python
import pandas as pd
import numpy as np
import scipy.stats as stats

df = pd.read_csv("data.csv")

# loop over columns and print out means for age and smoking
for column in list(df.columns.values):
    if column in ['age', 'smoking']:
        colmean = np.mean(df[column])
        print("{col} mean is {mean}\n".format(col=column, mean=colmean))
```

age mean is 62.797

smoking mean is 0.484

```python
# rename a variable
df.rename(columns={'age': 'age_in_years'}, inplace=True)

# subset based on a value
df = df[df['age_in_years'] > 50]

# create a new variable by adding or multiplying two variables
df['age_first_smoked'] = df['age_in_years'] - df['years_smoked']

# drop rows with missing values for the statistical test
df_test = df.dropna(subset=['age_first_smoked'])

# create categorical variables for chi-square test (chi-square requires categorical data)
# use quantiles to ensure balanced groups
df_test['age_group'] = pd.qcut(df_test['age_in_years'], q=3, labels=['50-62', '62-72', '72+'])
df_test['age_first_smoked_group'] = pd.qcut(df_test['age_first_smoked'], q=3, labels=['Low', 'Medium',

# compute a summary statistic (chi-square test in this example)
crosstab = pd.crosstab(df_test['age_group'], df_test['age_first_smoked_group'])
chi2_stat, p_val, dof, expected = stats.chi2_contingency(crosstab)
```

```
print(f"chi-square test statistic: {chi2_stat}")
```

```
chi-square test statistic: 460.02586917634187
```

```
print(f"p-value: {p_val}")
```

```
p-value: 2.9531672193224846e-98
```

```
print(f"degrees of freedom: {dof}")
```

```
degrees of freedom: 4
```

python output preview

the first 5 rows of the output from the python code:

```
head = df.head()
print(head.to_markdown())
```

	id	zip_code	age_in_years	smoking	years_smoked	age_first_smoked
0	1	98103	55	0	nan	nan
1	2	98104	75	1	9	66
3	4	98122	91	0	nan	nan
4	5	98106	66	1	9	57
5	6	98178	74	0	nan	nan

r translation (answer key)

```
# read in the dataset
df <- read.csv("data.csv")

# loop over columns and print out means for age and smoking
columns <- colnames(df)
for (column in columns) {
  if (column %in% c('age', 'smoking')) {
    print(sprintf("%s mean is %s", column, mean(df[[column]])))
  }
}
```

```
[1] “age mean is 62.797” [1] “smoking mean is 0.484”
```

```
# rename a variable
df <- df %>%
  rename(age_in_years = age)

# subset based on a value
df <- df %>%
  filter(age_in_years > 50)

# create a new variable by adding or multiplying two variables
df <- df %>%
  mutate(age_first_smoked = age_in_years - years_smoked)

# drop rows with missing values for the statistical test
df_test <- df %>%
```

```r
  filter(!is.na(age_first_smoked))

# create categorical variables for chi-square test (chi-square requires categorical data)
# use quantiles to ensure balanced groups
df_test <- df_test %>%
  mutate(
    age_group = cut(age_in_years, breaks = quantile(age_in_years, probs = c(0, 1/3, 2/3, 1), na.rm = TRU
                    labels = c('50-62', '62-72', '72+'), include.lowest = TRUE),
    age_first_smoked_group = cut(age_first_smoked, breaks = quantile(age_first_smoked, probs = c(0, 1/3
                                 labels = c('Low', 'Medium', 'High'), include.lowest = TRUE)
  )

# compute a summary statistic (chi-square test)
chisq_result <- chisq.test(table(df_test$age_group, df_test$age_first_smoked_group))
print(sprintf("chi-square test statistic: %s", chisq_result$statistic))
```

```
[1] “chi-square test statistic: 460.025869176342”
```

```r
print(sprintf("p-value: %s", chisq_result$p.value))
```

```
[1] “p-value: 2.95316721932246e-98”
```

```r
print(sprintf("degrees of freedom: %s", chisq_result$parameter))
```

```
[1] “degrees of freedom: 4”
```

first 5 rows of the r output

```r
head(df, 5) %>%
  kable() %>%
  kable_styling(bootstrap_options = "striped")
```

id	zip_code	age_in_years	smoking	years_smoked	age_first_smoked
1	98103	55	0	NA	NA
2	98104	75	1	9	66
4	98122	91	0	NA	NA
5	98106	66	1	9	57
6	98178	74	0	NA	NA

appendix: all code

```
# Setup
knitr::opts_chunk$set(results = 'asis', fig.align = 'center', echo = TRUE,
                      warning = FALSE, message = FALSE)

library(pacman)
pacman::p_load(tidyverse, reticulate, knitr, kableExtra)
use_condaenv("practice_env")
set.seed(123)
n <- 1000

ids <- 1:n
zip_codes <- c(
  "98101", "98102", "98103", "98104", "98105", "98106", "98107", "98108", "98109", "98112",
  "98115", "98116", "98117", "98118", "98119", "98121", "98122", "98125", "98126", "98133",
  "98134", "98136", "98144", "98146", "98154", "98164", "98174", "98177", "98178", "98195",
  "98199"
)
ages <- sample(25:100, n, replace = TRUE)
smoking <- sample(0:1, n, replace = TRUE)
years_smoked <- sample(0:10, n, replace = TRUE)

data <- data.frame(
  id = ids,
  zip_code = sample(zip_codes, n, replace = TRUE),
  age = ages,
  smoking = smoking,
  years_smoked = ifelse(smoking == 1, sample(0:10, n, replace = TRUE), NA)
)

write.csv(data, "data.csv", row.names = FALSE)
import pandas as pd
import numpy as np
import scipy.stats as stats

df = pd.read_csv("data.csv")

# loop over columns and print out means for age and smoking
for column in list(df.columns.values):
    if column in ['age', 'smoking']:
        colmean = np.mean(df[column])
        print("{col} mean is {mean}\n".format(col=column, mean=colmean))

# rename a variable
df.rename(columns={'age': 'age_in_years'}, inplace=True)

# subset based on a value
df = df[df['age_in_years'] > 50]

# create a new variable by adding or multiplying two variables
df['age_first_smoked'] = df['age_in_years'] - df['years_smoked']

# drop rows with missing values for the statistical test
```

```
df_test = df.dropna(subset=['age_first_smoked'])

# create categorical variables for chi-square test (chi-square requires categorical data)
# use quantiles to ensure balanced groups
df_test['age_group'] = pd.qcut(df_test['age_in_years'], q=3, labels=['50-62', '62-72', '72+'])
df_test['age_first_smoked_group'] = pd.qcut(df_test['age_first_smoked'], q=3, labels=['Low', 'Medium',

# compute a summary statistic (chi-square test in this example)
crosstab = pd.crosstab(df_test['age_group'], df_test['age_first_smoked_group'])
chi2_stat, p_val, dof, expected = stats.chi2_contingency(crosstab)

print(f"chi-square test statistic: {chi2_stat}")
print(f"p-value: {p_val}")
print(f"degrees of freedom: {dof}")
head = df.head()
print(head.to_markdown())
# read in the dataset
df <- read.csv("data.csv")

# loop over columns and print out means for age and smoking
columns <- colnames(df)
for (column in columns) {
  if (column %in% c('age', 'smoking')) {
    print(sprintf("%s mean is %s", column, mean(df[[column]])))
  }
}

# rename a variable
df <- df %>%
  rename(age_in_years = age)

# subset based on a value
df <- df %>%
  filter(age_in_years > 50)

# create a new variable by adding or multiplying two variables
df <- df %>%
  mutate(age_first_smoked = age_in_years - years_smoked)

# drop rows with missing values for the statistical test
df_test <- df %>%
  filter(!is.na(age_first_smoked))

# create categorical variables for chi-square test (chi-square requires categorical data)
# use quantiles to ensure balanced groups
df_test <- df_test %>%
  mutate(
    age_group = cut(age_in_years, breaks = quantile(age_in_years, probs = c(0, 1/3, 2/3, 1), na.rm = TRU
                    labels = c('50-62', '62-72', '72+'), include.lowest = TRUE),
    age_first_smoked_group = cut(age_first_smoked, breaks = quantile(age_first_smoked, probs = c(0, 1/3
                                 labels = c('Low', 'Medium', 'High'), include.lowest = TRUE)
  )
```

```r
# compute a summary statistic (chi-square test)
chisq_result <- chisq.test(table(df_test$age_group, df_test$age_first_smoked_group))
print(sprintf("chi-square test statistic: %s", chisq_result$statistic))
print(sprintf("p-value: %s", chisq_result$p.value))
print(sprintf("degrees of freedom: %s", chisq_result$parameter))
head(df, 5) %>%
  kable() %>%
  kable_styling(bootstrap_options = "striped")
```