



세상을 밝혀줄 인공지능

#에이림 #AIRIM

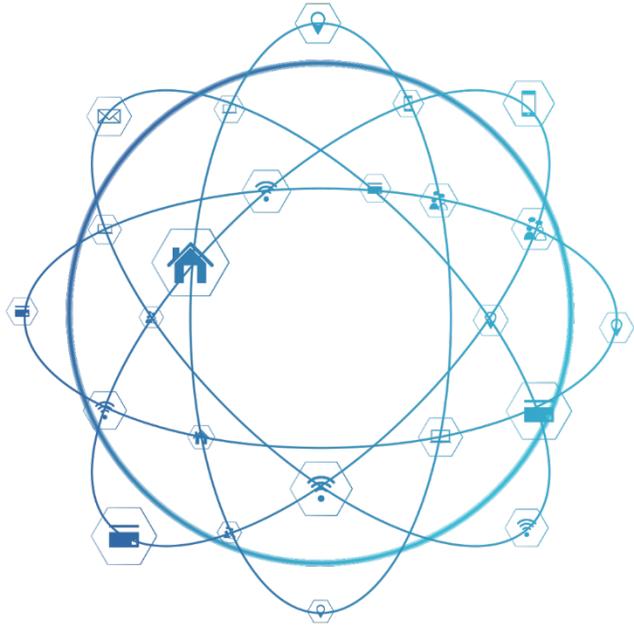
# AI 벤자민

- 2016년 4월 영국 SF영화제 'SF런던' 출품 (SUNSPRING)
- 등장인물의 대사, 배경음악의 가사, 무대지시 등 모두 혼자서 작성
- TOP 10에 선정

- 개연성의 부족
- 기괴함

# 인공지능 - AI

---



- 구분하기
- 정답 맞추기
- 했던거 또하기(With. 재귀신경망)

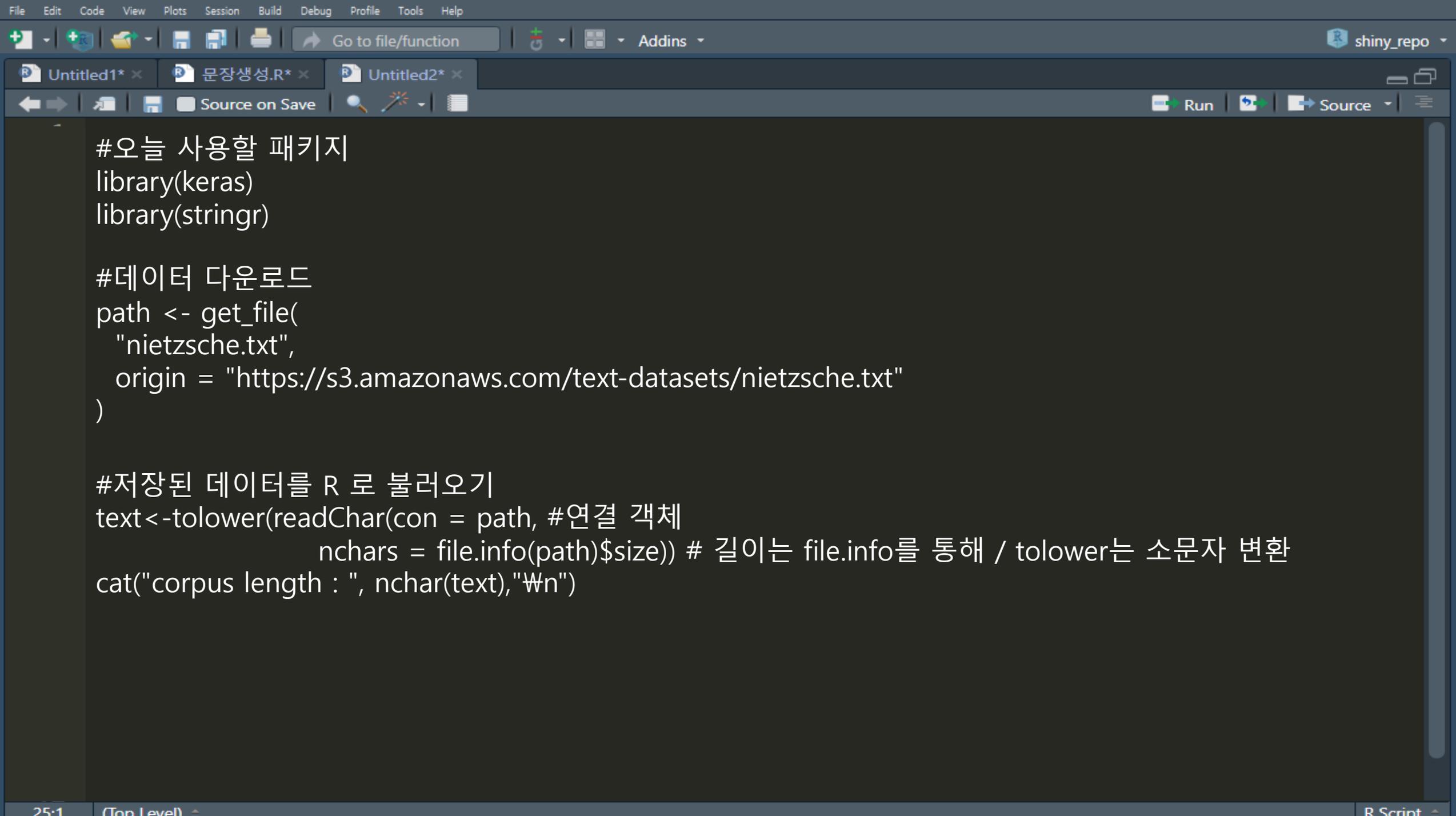
• 데이터 생성 !!

# So What?

---

## 케라스





```
#오늘 사용할 패키지
```

```
library(keras)
```

```
library(stringr)
```

```
#데이터 다운로드
```

```
path <- get_file(
```

```
  "nietzsche.txt",
```

```
  origin = "https://s3.amazonaws.com/text-datasets/nietzsche.txt"
```

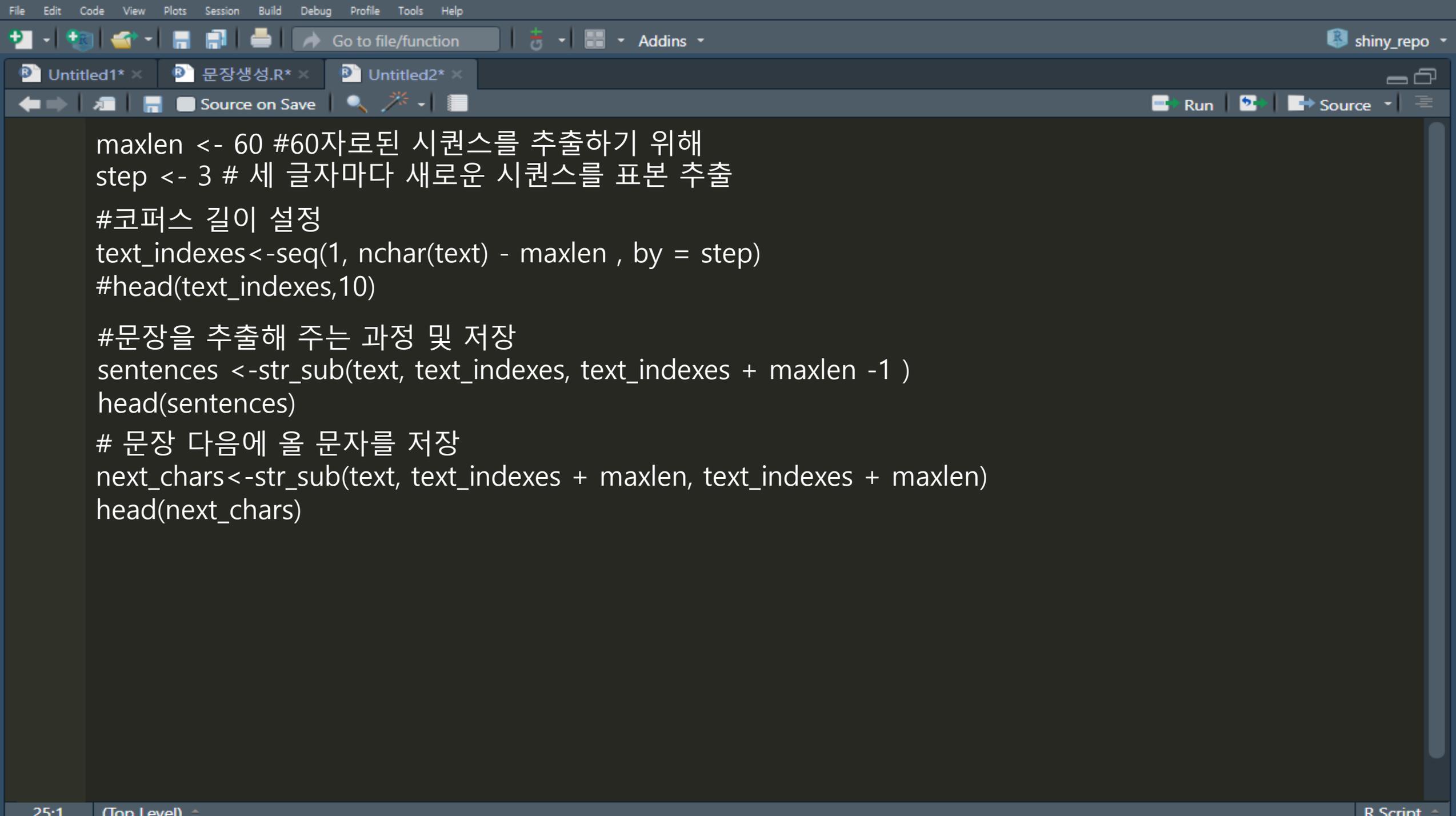
```
)
```

```
#저장된 데이터를 R 로 불러오기
```

```
text<-tolower(readChar(con = path, #연결 객체
```

```
                  nchars = file.info(path)$size)) # 길이는 file.info를 통해 / tolower는 소문자 변환
```

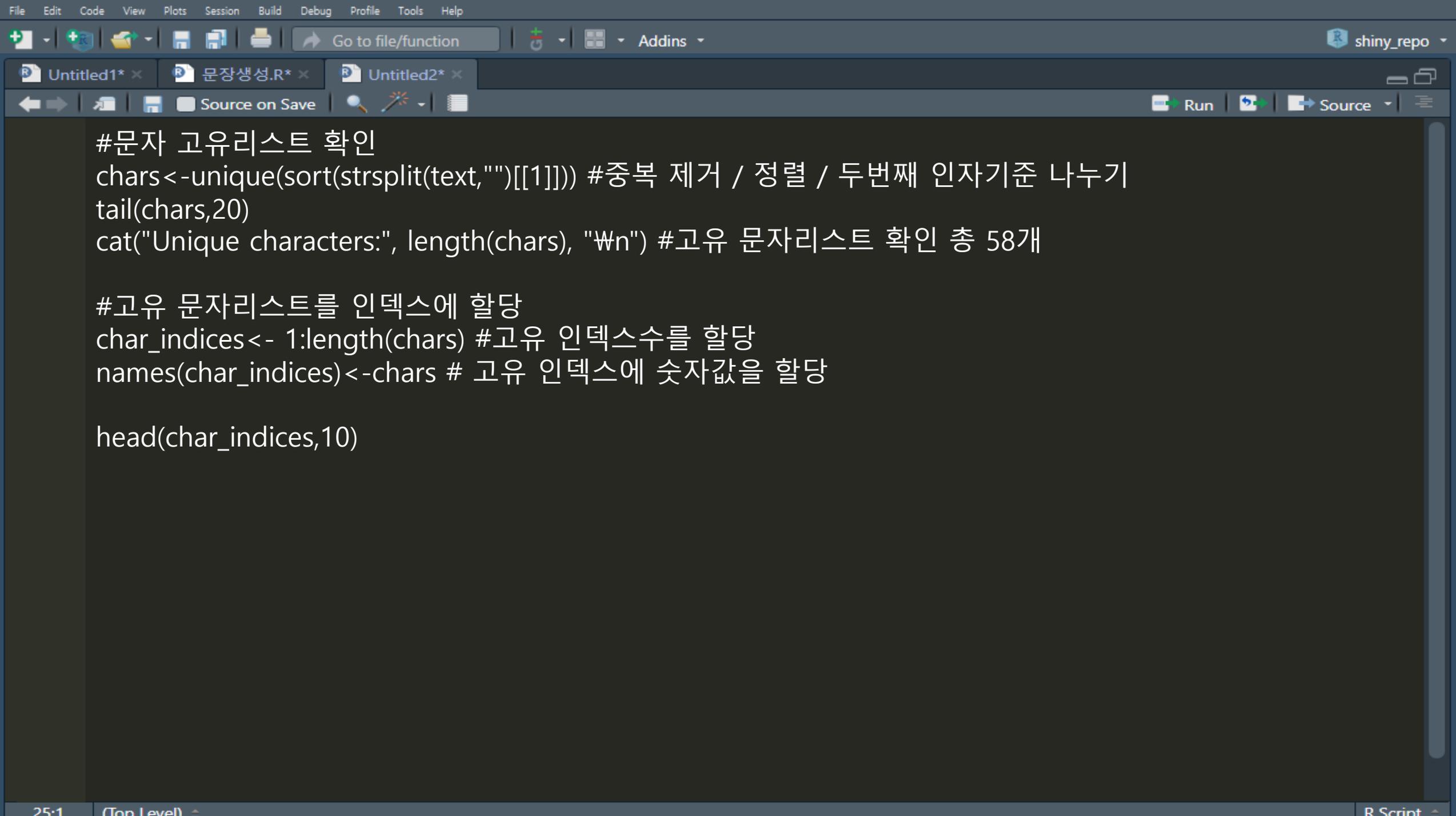
```
cat("corpus length : ", nchar(text),"\\n")
```



```
maxlen <- 60 #60자로된 시퀀스를 추출하기 위해
step <- 3 # 세 글자마다 새로운 시퀀스를 표본 추출
#코퍼스 길이 설정
text_indexes<-seq(1, nchar(text) - maxlen , by = step)
#head(text_indexes,10)

#문장을 추출해 주는 과정 및 저장
sentences <-str_sub(text, text_indexes, text_indexes + maxlen -1 )
head(sentences)

# 문장 다음에 올 문자를 저장
next_chars<-str_sub(text, text_indexes + maxlen, text_indexes + maxlen)
head(next_chars)
```



```
#문자 고유리스트 확인
```

```
chars<-unique(sort(strsplit(text,"")[[1]])) #중복 제거 / 정렬 / 두번째 인자기준 나누기
```

```
tail(chars,20)
```

```
cat("Unique characters:", length(chars), "\n") #고유 문자리스트 확인 총 58개
```

```
#고유 문자리스트를 인덱스에 할당
```

```
char_indices<- 1:length(chars) #고유 인덱스수를 할당
```

```
names(char_indices)<-chars # 고유 인덱스에 숫자값을 할당
```

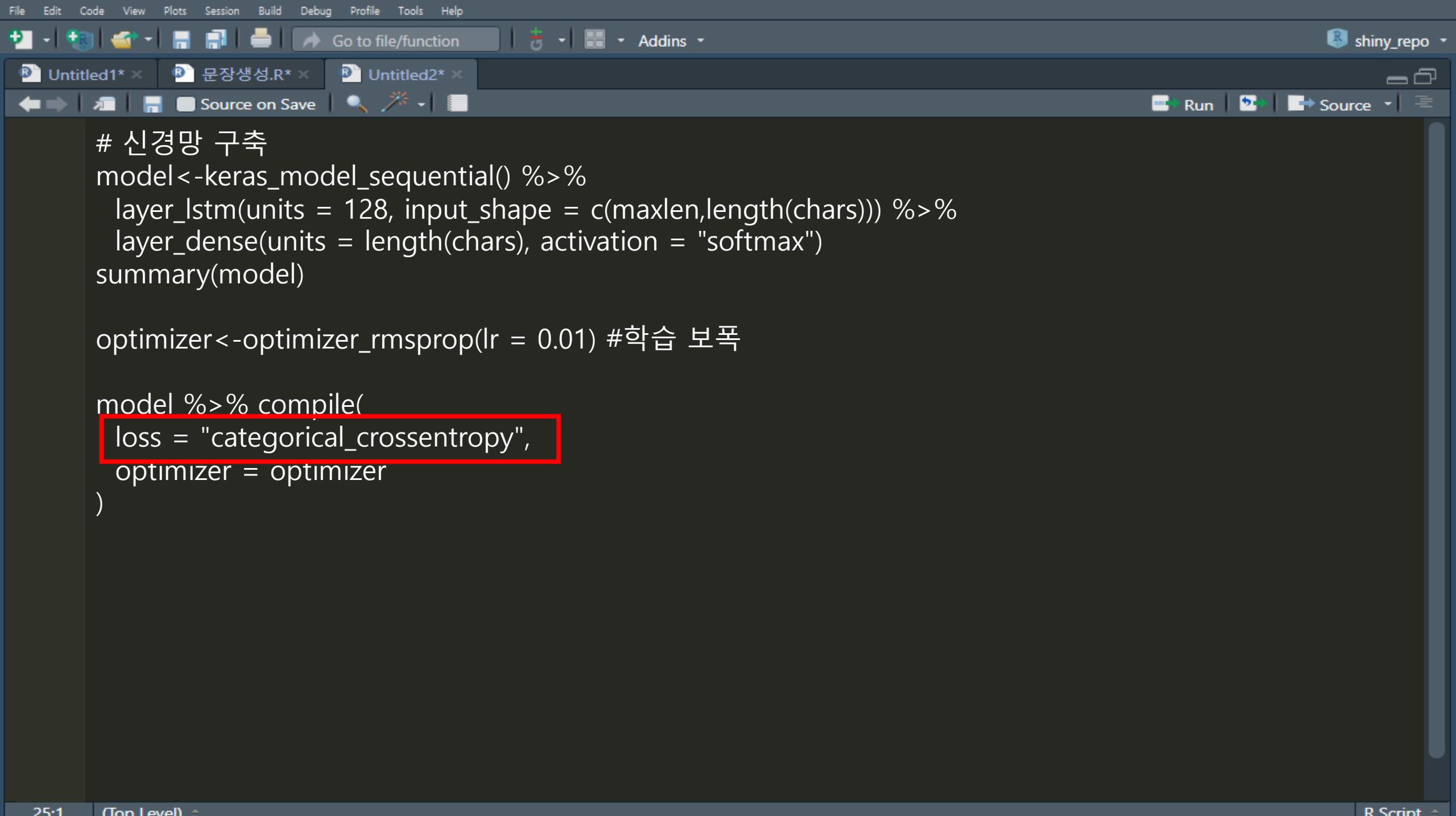
```
head(char_indices,10)
```

### # 벡터화

```
x<-array(0L, dim = c(length(sentences),maxlen,length(chars)))  
str(x) # 시퀀스, 최대길이, 고유문자들을 3D 배열 X 로 압축로  
y<-array(0L,dim = c(length(sentences), length(chars)))  
str(y) #Y는 추출된 시퀀스 다음에 오는 문자를 포함하는 배열로 준비
```

```
for (i in 1:length(sentences)) {  
  sentence <- strsplit(sentences[[i]], "")[[1]]  
  for (t in 1:length(sentence)) {  
    char <- sentence[[t]]  
    x[i, t, char_indices[[char]]] <- 1  
  }  
  next_char <- next_chars[[i]]  
  y[i, char_indices[[next_char]]] <- 1  
}
```





```
# 신경망 구축
```

```
model<-keras_model_sequential() %>%
  layer_lstm(units = 128, input_shape = c(maxlen,length(chars))) %>%
  layer_dense(units = length(chars), activation = "softmax")
summary(model)
```

```
optimizer<-optimizer_rmsprop(lr = 0.01) #학습 보폭
```

```
model %>% compile(
  loss = "categorical_crossentropy",
  optimizer = optimizer
)
```

Untitled1\* ×

문장생성.R\* ×

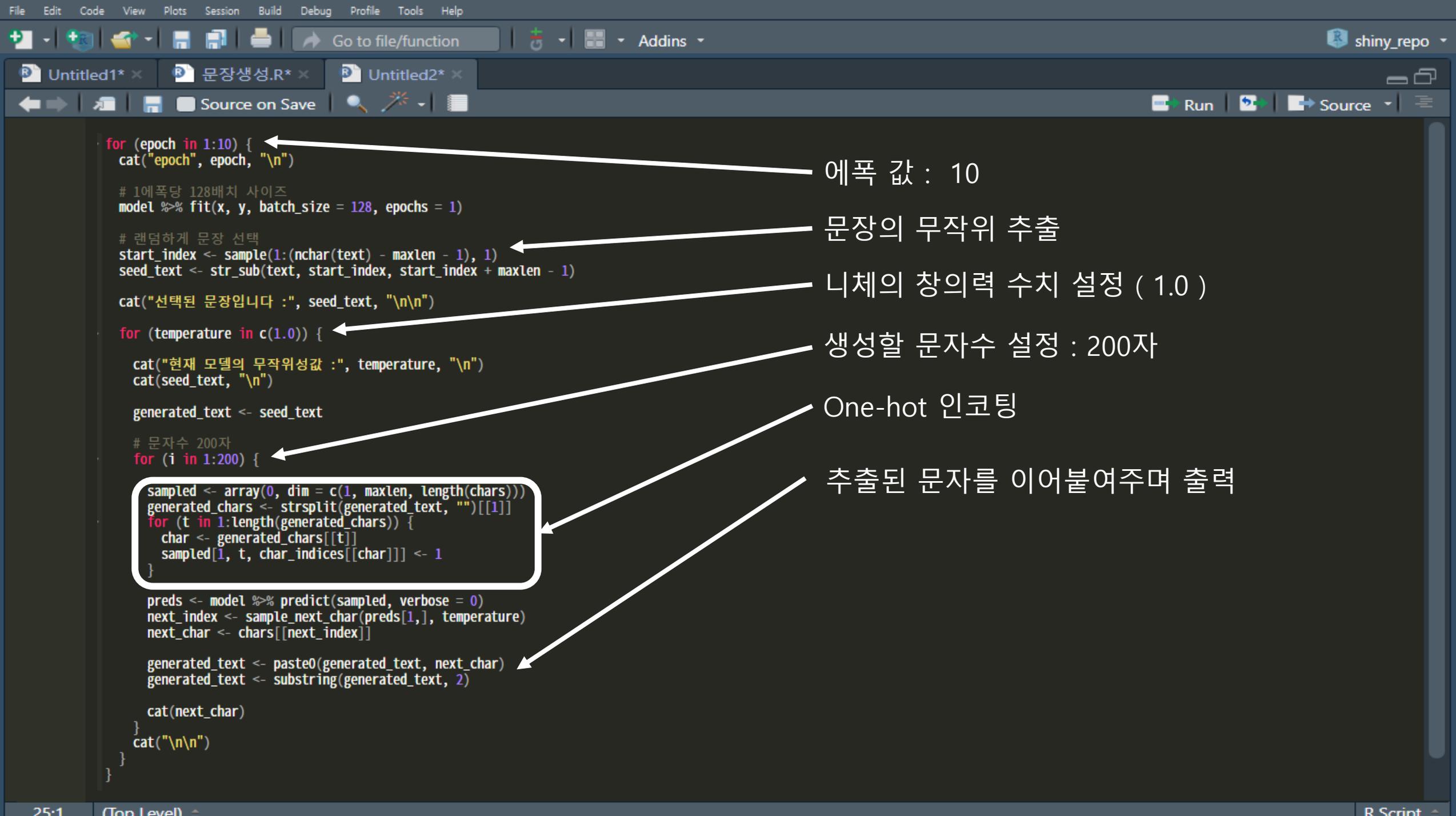
Untitled2\* ×

Source on Save

Run

Source

```
#모델의 예측에 따라 다음 문자를 표본추출하는 함수 ( 모델의 창의력 )
sample_next_char <- function(preds, temperature = 1.0) {
  preds <- as.numeric(preds)
  preds <- log(preds) / temperature
  exp_preds <- exp(preds)
  preds <- exp_preds / sum(exp_preds)
  which.max(t(rmultinom(1, 1, preds)))
}
```



```

for (epoch in 1:10) {
  cat("epoch", epoch, "\n")

  # 1에폭당 128배치 사이즈
  model %>% fit(x, y, batch_size = 128, epochs = 1)

  # 랜덤하게 문장 선택
  start_index <- sample(1:(nchar(text) - maxlen - 1), 1)
  seed_text <- str_sub(text, start_index, start_index + maxlen - 1)

  cat("선택된 문장입니다 :", seed_text, "\n\n")

  for (temperature in c(1.0)) {

    cat("현재 모델의 무작위성값 :", temperature, "\n")
    cat(seed_text, "\n")

    generated_text <- seed_text

    # 문자수 200자
    for (i in 1:200) {

      sampled <- array(0, dim = c(1, maxlen, length(chars)))
      generated_chars <- strsplit(generated_text, "")[[1]]
      for (t in 1:length(generated_chars)) {
        char <- generated_chars[[t]]
        sampled[1, t, char_indices[[char]]] <- 1
      }

      preds <- model %>% predict(sampled, verbose = 0)
      next_index <- sample_next_char(preds[1,], temperature)
      next_char <- chars[[next_index]]

      generated_text <- paste0(generated_text, next_char)
      generated_text <- substring(generated_text, 2)

      cat(next_char)
    }
    cat("\n\n")
  }
}

```

에폭 값 : 10

문장의 무작위 추출

니체의 창의력 수치 설정 ( 1.0 )

생성할 문자수 설정 : 200자

One-hot 인코딩

추출된 문자를 이어붙여주며 출력

sampled <- array(0, dim = c(1, maxlen, length(chars)))  
generated\_chars <- strsplit(generated\_text, "")[[1]]  
for (t in 1:length(generated\_chars)) {  
 char <- generated\_chars[[t]]  
 sampled[1, t, char\_indices[[char]]] <- 1  
}



# Q & A

Q. 공대 나오셨나요 ?

A. 아뇨 외식조리경영 나왔습니다!

Q. 조리과가 왜 이거해요?

A. 외식업하시는 분들이 제일 필요한 분야입니다.

Q. 다음 Meet Up도 해주실 수 있나요?

A. 저야 불러주시면 감사하죠!

# 감사합니다 :)

(영상으로 다시보기 ▶ 에이림)

좋아요



구독



응원의  
댓글

