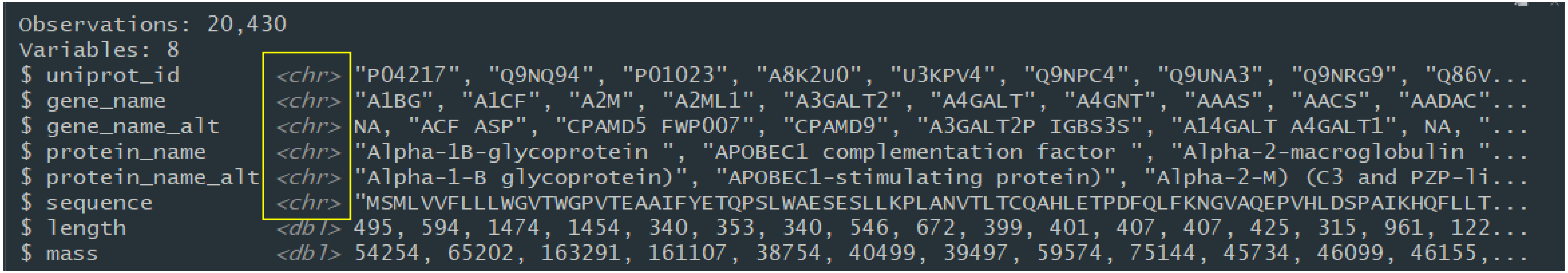
proteins

```
glimpse(proteins)
```



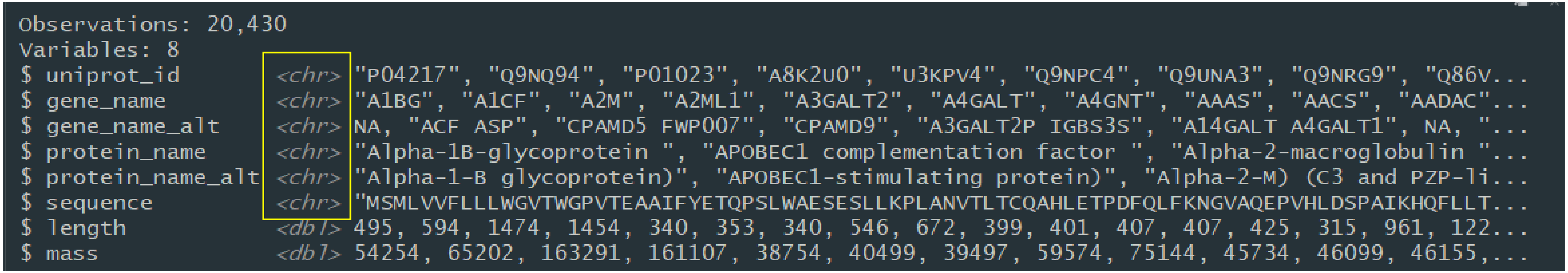
```
Observations: 20,430
Variables: 8
$ uniprot_id       <chr> "P04217", "Q9NQ94", "P01023", "A8K2U0", "U3KPV4", "Q9NPC4", "Q9UNA3", "Q9NRG9", "Q86V...
$ gene_name        <chr> "A1BG", "A1CF", "A2M", "A2ML1", "A3GALT2", "A4GALT", "A4GNT", "AAAS", "AACS", "AADAC"...
$ gene_name_alt    <chr> NA, "ACF ASP", "CPAMD5 FWP007", "CPAMD9", "A3GALT2P IGBS3S", "A14GALT A4GALT1", NA, "...
$ protein_name     <chr> "Alpha-1B-glycoprotein ", "APOBEC1 complementation factor ", "Alpha-2-macroglobulin "...
$ protein_name_alt <chr> "Alpha-1-B glycoprotein)", "APOBEC1-stimulating protein)", "Alpha-2-M) (C3 and PZP-li...
$ sequence         <chr> "MSMLVVFLLLWGVTWGPVTEAAIFYETQPSLWAESESLLKPLANVTLTCQAHLETPDFQLFKNGVAQEPVHLDSPAIKHQFLLT...
$ length           <dbl> 495, 594, 1474, 1454, 340, 353, 340, 546, 672, 399, 401, 407, 407, 425, 315, 961, 122...
$ mass             <dbl> 54254, 65202, 163291, 161107, 38754, 40499, 39497, 59574, 75144, 45734, 46099, 46155,...
```

These columns contain strings!

stringr
www.rstudio.com

proteins

```
glimpse(proteins)
```



```
Observations: 20,430
Variables: 8
$ uniprot_id       <chr> "P04217", "Q9NQ94", "P01023", "A8K2U0", "U3KPV4", "Q9NPC4", "Q9UNA3", "Q9NRG9", "Q86V...
$ gene_name        <chr> "A1BG", "A1CF", "A2M", "A2ML1", "A3GALT2", "A4GALT", "A4GNT", "AAAS", "AACS", "AADAC"...
$ gene_name_alt    <chr> NA, "ACF ASP", "CPAMD5 FWP007", "CPAMD9", "A3GALT2P IGBS3S", "A14GALT A4GALT1", NA, "...
$ protein_name     <chr> "Alpha-1B-glycoprotein ", "APOBEC1 complementation factor ", "Alpha-2-macroglobulin "...
$ protein_name_alt <chr> "Alpha-1-B glycoprotein)", "APOBEC1-stimulating protein)", "Alpha-2-M) (C3 and PZP-li...
$ sequence         <chr> "MSMLVVFLLLWGVTWGPVTEAAIFYETQPSLWAESESLLKPLANVTLTCQAHLETPDFQLFKNGVAQEPVHLDSPAIKHQFLLT...
$ length           <dbl> 495, 594, 1474, 1454, 340, 353, 340, 546, 672, 399, 401, 407, 407, 425, 315, 961, 122...
$ mass             <dbl> 54254, 65202, 163291, 161107, 38754, 40499, 39497, 59574, 75144, 45734, 46099, 46155,...
```

Counting

How many AA's are in each protein?

String to be evaluated

```
str_count(proteins$sequence)
```

Counting

How many lysines are in each protein?

String to be evaluated

```
str_count(proteins$sequence, “K”)
```

“pattern” (what to *count*)

Exercise 1

Using the str_count() function, determine how many instances of lysine next to a cysteine there are.

```
str_count(proteins$sequence, “KC”)
```

Counting

How many lysines are in each protein?

String to be evaluated

```
str_count(proteins$sequence, “K”)
```

“pattern” (what to count)

Logical evaluation

TRUE or FALSE

Presence or Absence

```
str_detect()
```

Questions

Run the following code:

```
str_detect(protein$sequence, “SP.R” )
```

1) How is this the same or different from str_count()?

2) What does the '.' stand for in this code?

3) How does this compare to the sequence motif we want to find?

Logical evaluation

(S/T)PX(K/R) – cyclin binding motif

String to be evaluated

```
str_detect(protein$sequence, “SP.R”)
```

“pattern” (what to *detect*)

Logical evaluation

(S/T)PX(K/R) – cyclin binding motif

```
str_detect(protein$sequence, “SP.R”)
```

Regular Expressions (regex) are language used to describe patterns in strings

‘.’ = will accept ANY character in this position

Logical evaluation

(S/T)PX(K/R) – cyclin binding motif

(S/T)PX(K/R)
=/=
SPXR

```
str_detect(protein$sequence, “SP.R”)
```

How can we add more information to our pattern?

Logical evaluation

(S/T)PX(K/R) – cyclin binding motif

Will accept S *OR* T

```
str_detect(protein$sequence, “(S|T)P.R”)
```

Evaluate First

Logical evaluation

(S/T)PX(K/R) – cyclin binding motif

Will accept
S *OR* T

```
str_detect(protein$sequence, “(S|T)P.R”)
```

Evaluate
First

Exercise 2

How would you alter this code to accept K or R in the final position?

```
str_detect(protein$sequence, “(S|T)P.R”)
```

```
str_detect(protein$sequence, “(S|T)P.(K|R)”)
```

Exercise 2

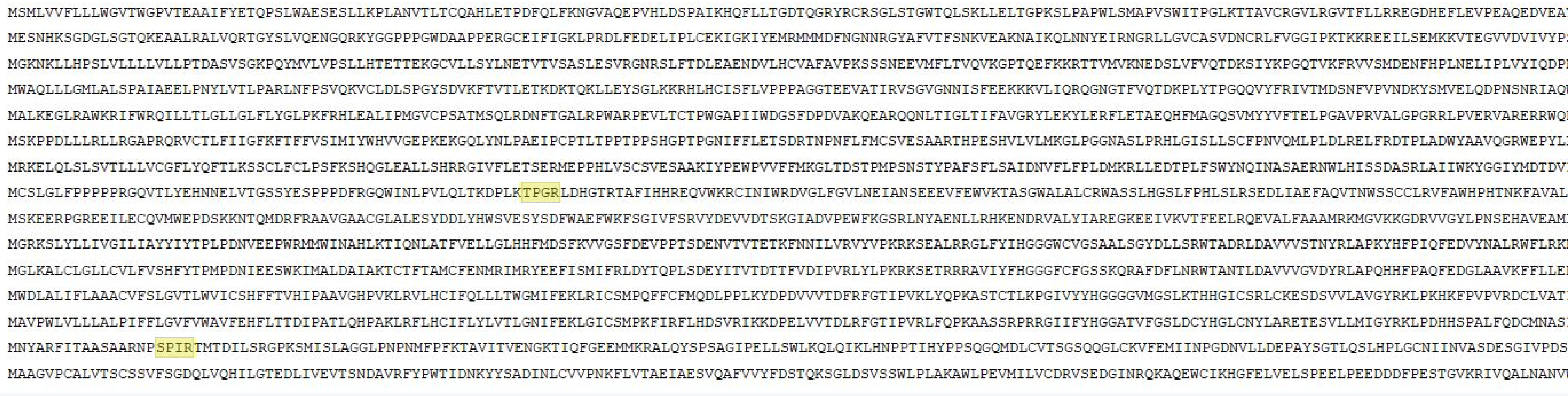
How did you know it worked?? Try replacing ‘str_detect’ with ‘str_view’

```
str_detect(protein$sequence, “(S|T)P.(K|R)”)
```

```
str_view(protein$sequence, “(S|T)P.(K|R)”)
```

Logical evaluation

```
str_view(protein$sequence, “(S|T)P.(K|R)”)
```



```
MSMLVVFLLLWGVTWGPVTEAAIFYETQPSLWAESESLLKPLANVTLTCQAHLETPDFQLFKNGVAQEPVHLDSPAIKHQFLLTGDTQGRYRCRSGLSTGWTQLSKLLELTGPKSLPAPWLSMAPVSWITPGLKTTAVCRGVLRGVTFLLRREGDHEFLEVPEAQEDVEA
MESNHKSGDGLSGTQKEAALRALVQRTGYSLVQENGQRKYGGPPPGWDAAPPERGCEIFIGKLPRDLFEDELIPLCEKIGKIYEMRMMMDFNGNNRGYAFVTFSNKVEAKNAIKQLNNYEIRNGRLLGVCASVDNCRLFVGGIPKTKKREEILSEMKKVTEGVVDVIVYP
MGKNKLLHPSLVLLLLVLLPTDASVSGKPQYMVLVPSLLHTETTEKGCVLLSYLNETVTVSASLESVRGNRSLFTDLEAENDVLHCVAFAVPKSSSNEEVMFLTVQVKGPTQEFKKRTTVMVKNEDSLVFVQTDKSIYKPGQTVKFRVVSMDENFHPLNELIPLVYIQDP
MWAQLLLGMLALSPAIAEELPNYLVTLPARLNFPSVQKVCLDLSPGYSDVKFTVTLETKDKTQKLLEYSGLKKRHLHCISFLVPPPAGGTEEVATIRVSGVGNNISFEEKKKVLIQRQGNGTFVQTDKPLYTPGQQVYFRIVTMDSNFVPVNDKYSMVELQDPNSNRIAQ
MALKEGLRAWKRIFWRQILLTLGLLGLFLYGLPKFRHLEALIPMGVCPSATMSQLRDNFTGALRPWARPEVLTCTPWGAPIIWDGSFDPDVAKQEARQQNLTIGLTIFAVGRYLEKYLERFLETAEQHFMAGQSVMYYVFTELPGAVPRVALGPGRRLPVERVARERRWQ
MSKPPDLLLRLLRGAPRQRVCTLFIIGFKFTFFVSIMIYWHVVGEPKEKGQLYNLPAEIPCPTLTPPTPPSHGPTPGNIFFLETSDRTNPNFLFMCSVESAARTHPESHVLVLMKGLPGGNASLPRHLGISLLSCFPNVQMLPLDLRELFRDTPLADWYAAVQGRWEPYL
MRKELQLSLSVTLLLVCGFLYQFTLKSSCLFCLPSFKSHQGLEALLSHRRGIVFLETSERMEPPHLVSCSVESAAKIYPEWPVVFFMKGLTDSTPMPSNSTYPAFSFLSAIDNVFLFPLDMKRLLEDTPLFSWYNQINASAERNWLHISSDASRLAIIWKYGGIYMDTDV
MCSLGLFPPPPPRGQVTLYEHNNELVTGSSYESPPPDFRGQWINLPVLQLTKDPLKTPGRLDHGTRTAFIHHREQVWKRCINIWRDVGLFGVLNEIANSEEEVFEWVKTASGWALALCRWASSLHGSLFPHLSLRSEDLIAEFAQVTNWSSCCLRVFAWHPHTNKFAVAL
MSKEERPGREEILECQVMWEPDSKKNTQMDRFRAAVGAACGLALESYDDLYHWSVESYSDFWAEFWKFSGIVFSRVYDEVVDTSKGIADVPEWFKGSRLNYAENLLRHKENDRVALYIAREGKEEIVKVTFEELRQEVALFAAAMRKMGVKKGDRVVGYLPNSEHAVEAM
MGRKSLYLLIVGILIAYYIYTPLPDNVEEPWRMMWINAHLKTIQNLATFVELLGLHHFMDSFKVVGSFDEVPPTSDENVTVTETKFNNILVRVYVPKRKSEALRRGLFYIHGGGWCVGSAALSGYDLLSRWTADRLDAVVVSTNYRLAPKYHFPIQFEDVYNALRWFLRK
MGLKALCLGLLCVLFVSHFYTPMPDNIEESWKIMALDAIAKTCTFTAMCFENMRIMRYEEFISMIFRLDYTQPLSDEYITVTDTTFVDIPVRLYLPKRKSETRRRAVIYFHGGGFCFGSSKQRAFDFLNRWTANTLDAVVVGVDYRLAPQHHFPAQFEDGLAAVKFFLLE
MWDLALIFLAAACVFSLGVTLWVICSHFFTVHIPAAVGHPVKLRVLHCIFQLLLTWGMIFEKLRICSMPQFFCFMQDLPPLKYDPDVVVTDFRFGTIPVKLYQPKASTCTLKPGIVYYHGGGGVMGSLKTHHGICSRLCKESDSVVLAVGYRKLPKHKFPVPVRDCLVAT
MAVPWLVLLLALPIFFLGVFVWAVFEHFLTTDIPATLQHPAKLRFLHCIFLYLVTLGNIFEKLGICSMPKFIRFLHDSVRIKKDPELVVTDLRFGTIPVRLFQPKAASSRPRRGIIFYHGGATVFGSLDCYHGLCNYLARETESVLLMIGYRKLPDHHSPALFQDCMNAS
MNYARFITAASAARNPSPIRTMTDILSRGPKSMISLAGGLPNPNMFPFKTAVITVENGKTIQFGEEMMKRALQYSPSAGIPELLSWLKQLQIKLHNPPTIHYPPSQGQMDLCVTSGSQQGLCKVFEMIINPGDNVLLDEPAYSGTLQSLHPLGCNIINVASDESGIVPDS
MAAGVPCALVTSCSSVFSGDQLVQHILGTEDLIVEVTSNDAVRFYPWTIDNKYYSADINLCVVPNKFLVTAEIAESVQAFVVYFDSTQKSGLDSVSSWLPLAKAWLPEVMILVCDRVSEDGINRQKAQEWCIKHGFELVELSPEELPEEDDDFPESTGVKRIVQALNANV
```

Using stringr with dplyr

We have a code chunk that determines presence of a motif in our sequence, but how can we add this information to our data frame?

dplyr function

```
proteins <- mutate(proteins, cy_motif = str_detect(sequence,
     "(S|T)P.(K|R)"))
```

Column to create

stringr function