Analyze TCGA data for SKCM cohort

Jacqueline Buros & ...

2017-06-21

Here we are demonstrating the feasibility of analyzing genomic data using Stan. The first use case is to analyze somatic mutations for association with survival, after adjusting for key clinical variables with known prognostic status.

Data Exploration

Clinical Data

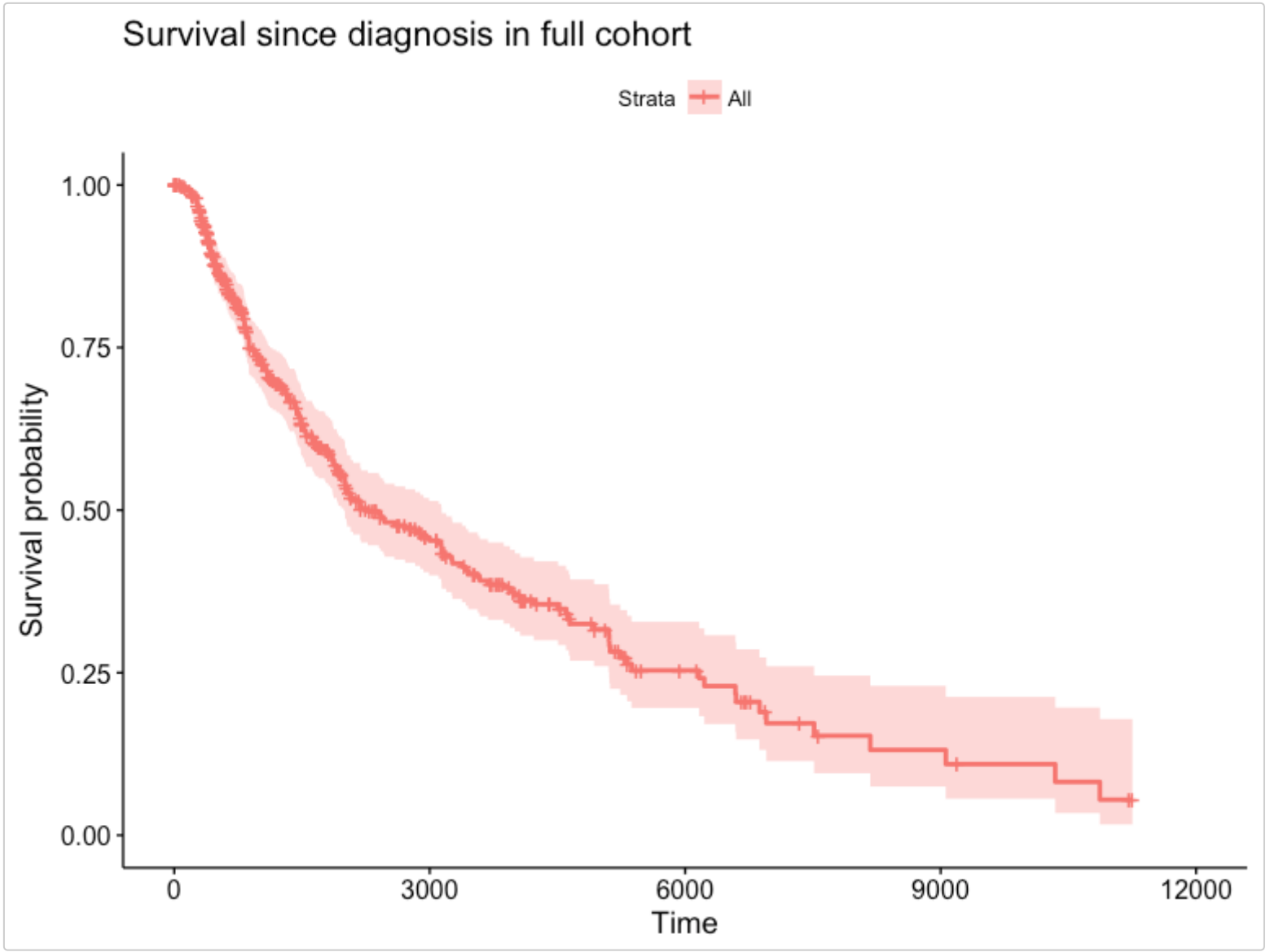
First, download the clinical data. Here we are using the TCGA skin cutaneous melanoma (SKCM) cohort.

```
clin_df <- SuMu::get_tcga_clinical(cohort = "SKCM")

## format some clinical data variables

clin_df2 <- clin_df %>%
  dplyr::mutate(stage_part1 = gsub(pathologic_stage,
                                   pattern = '(Stage [0I]+).*',
                                   replacement = '\\1'),
                diagnosis_year_group = cut(year_of_initial_pathologic_diagnosis,
                                           breaks = c(1975, 1990, 1995, 2000,
                                                      2005, 2010, 2015, 2020),
                                           include.lowest = TRUE),
                os_10y = ifelse(OS_IND == 1 & OS <= 10*365.25, 1, 0),
                sample = sampleID
                )
```

Review clinical data

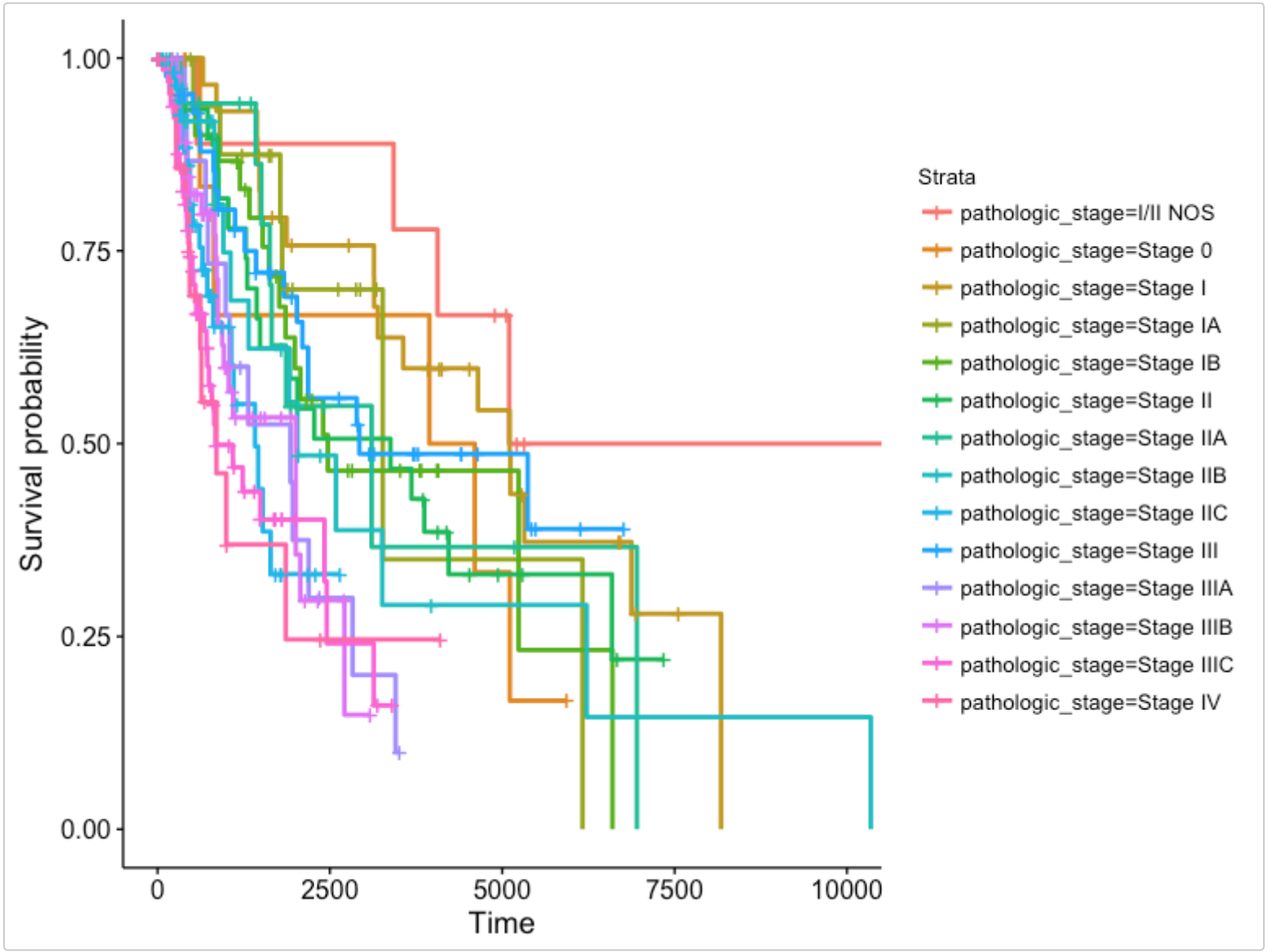
For this analysis we will consider the survival time in since initial pathologic diagnosis.

```
fit <- survfit(Surv(OS, OS_IND) ~ 1,
                  data = clin_df2)
survminer::ggsurvplot(fit) +
  ggtitle('Survival since diagnosis in full cohort')
```



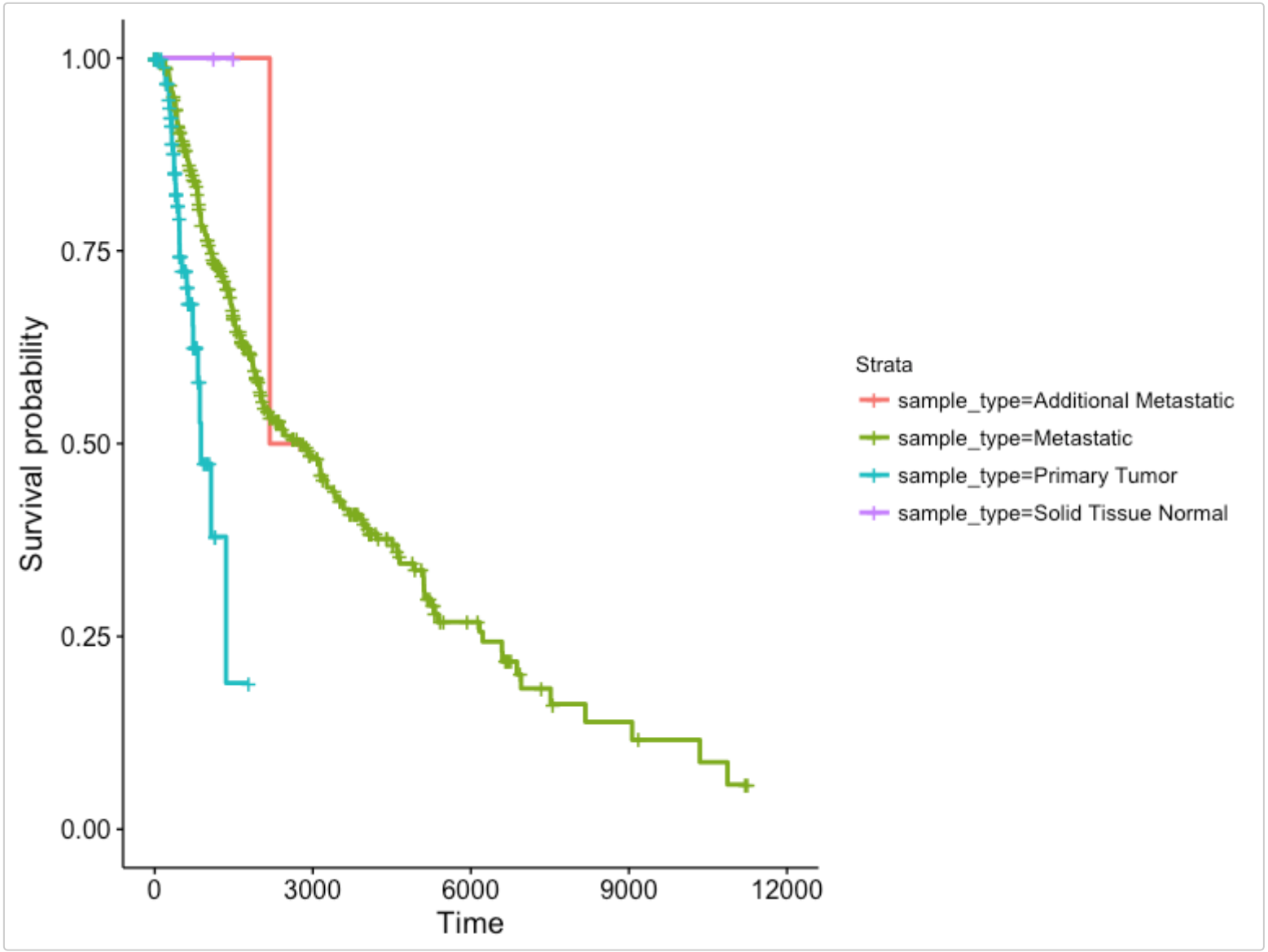
Plotting by stage, although the time of ‘stage’ determination may be confounded if not collected at time of initial diagnosis.

```
fit <- survfit(Surv(OS, OS_IND) ~ pathologic_stage,
                   data = clin_df2)
survminer::ggsurvplot(fit, legend = "right")
```



There also seem to be differences by tumor type.

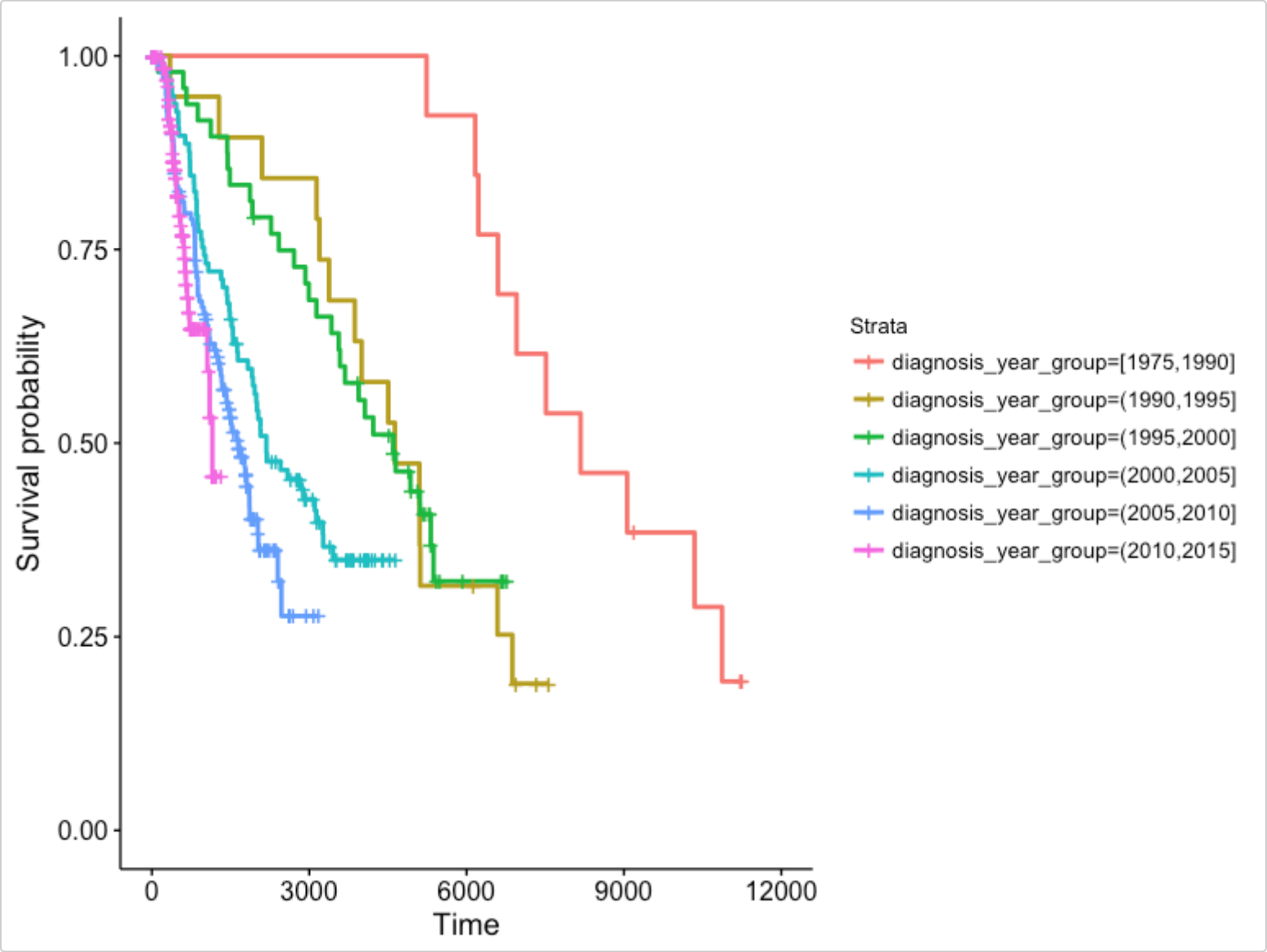
```
fit <- survfit(Surv(OS, OS_IND) ~ sample_type,
                  data = clin_df2)
survminer::ggsurvplot(fit, legend = "right")
```



(Aside: I wonder how similar tumor type is to sample type? For example, we could have a metastatic patient where the sample was obtained from the primary tumor. We will want to adjust our genetic data analysis for the sample type but may want to estimate prognosis according to the tumor type?)

A variable like `year_of_initial_pathologic_diagnosis` is guaranteed to be unconfounded since we can safely assume it was collected at the time of diagnosis.

```
fit <- survfit(Surv(OS, OS_IND) ~ diagnosis_year_group,
                   data = clin_df2)
survminer::ggsurvplot(fit, legend = 'right')
```

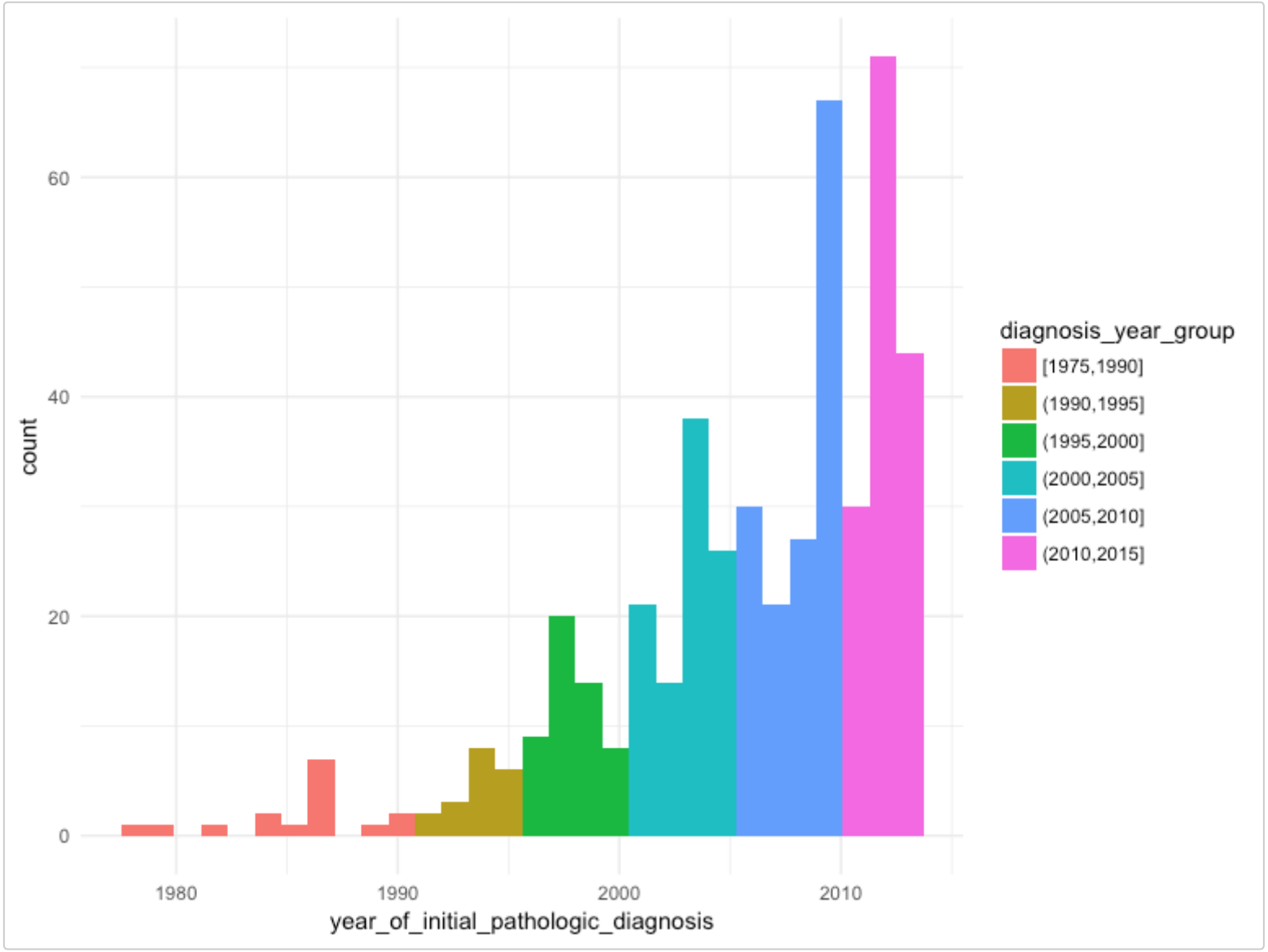


This makes it pretty clear that we have a strong "survival" bias to our data. This would suggest that, among people whose diagnosis was made in the 90s, only those who survived long enough to be enrolled were included in the study.

Let's look at a histogram of years of initial diagnosis.

```
ggplot(clin_df2, aes(x = year_of_initial_pathologic_diagnosis,
                     fill = diagnosis_year_group)) +
```

```
geom_histogram() +
theme_minimal()
```



Let's look at the time since initial diagnosis (presumably, the time from enrollment to diagnosis).

Finally, we can visualize a more comprehesive set of clinical variables.

```
fit <- survival::coxph(Surv(OS, OS_IND) ~
                         age_at_initial_pathologic_diagnosis +
                         sample_type +
                         breslow_depth_value + initial_weight +
                         strata(year_of_initial_pathologic_diagnosis),
               data = clin_df2)
print(fit)
```

```
## Call:
## survival::coxph(formula = Surv(OS, OS_IND) ~ age_at_initial_pathologic_diagnosis +
##     sample_type + breslow_depth_value + initial_weight +
strata(year_of_initial_pathologic_diagnosis),
##     data = clin_df2)
##
##                                        coef exp(coef)  se(coef)     z
## age_at_initial_pathologic_diagnosis  0.018324  1.018493  0.006078  3.01
## sample_typeMetastatic                0.697124  2.007970  1.046439  0.67
## sample_typePrimary Tumor             1.156752  3.179590  1.077900  1.07
## breslow_depth_value                  0.014482  1.014587  0.008354  1.73
## initial_weight                      -0.000223  0.999777  0.000471 -0.47
##                                        p
## age_at_initial_pathologic_diagnosis 0.0026
## sample_typeMetastatic               0.5053
## sample_typePrimary Tumor            0.2832
## breslow_depth_value                 0.0830
## initial_weight                      0.6362
##
## Likelihood ratio test=16.3  on 5 df, p=0.00607
## n= 338, number of events= 166
##    (143 observations deleted due to missingness)
```

Somatic Mutations Data

We can download the somatic mutations to supplement the phenotypes.

```
mut_df <- SuMu::get_tcga_somatic_mutations(cohort = "SKCM") %>%
  dplyr::mutate(gene_aa = paste0(gene, ".", Amino_Acid_Change),
```

```
                gene_effect = paste0(gene, ".", effect)
                )
```

Check the most frequent mutations.

```
mut_df_missense = mut_df %>% dplyr::filter(effect == "Missense_Mutation")
mut_df_missense$gene_aa = paste0(mut_df_missense$gene, ":", mut_df_missense$Amino_Acid_Change)
mut_df_missense %>% select(gene_aa) %>% table %>% sort %>% rev %>% as.data.frame %>% head(10)
```

```
##                   . Freq
## 1      BRAF:p.V600E  206
## 2       NRAS:p.Q61R   56
## 3      BRAF:p.V600M   40
## 4       NRAS:p.Q61K   38
## 5       NRAS:p.Q61L   19
## 6       RAC1:p.P29S   17
## 7      IDH1:p.R132C   15
## 8     SPTLC3:p.R97K   13
## 9   SLC27A5:p.T554I   13
## 10  MAP2K1:p.P124S   13
```

Filter to top genes

```
top_genes <- mut_df %>%
  dplyr::group_by(gene) %>%
  dplyr::mutate(gene_count = n()) %>%
  dplyr::ungroup() %>%
  dplyr::distinct(gene, .keep_all = TRUE) %>%
  dplyr::top_n(gene_count, n = 10) %>%
  dplyr::select(gene)

mut_df_topgenes <- mut_df %>%
  dplyr::semi_join(top_genes)
```

GLM model to all genes

Prepare mutation data for analysis

```
clin_df2_nonmiss <- clin_df2 %>%
  dplyr::mutate(
    revised_breslow_depth = ifelse(is.na(breslow_depth_value),
                                   0, breslow_depth_value)) %>%
  tidyr::drop_na(os_10y,
                 age_at_initial_pathologic_diagnosis,
                 initial_weight,
                 revised_breslow_depth,
                 sample_type,
                 diagnosis_year_group)

mutation_matrix <- SuMu:::prep_biomarker_data(
  biomarker_data = mut_df_topgenes,
  data = clin_df2_nonmiss,
  biomarker_formula = 1 ~ gene_effect,
  .fun = sum,
  id = 'sample'
)

glm_df2 <- mutation_matrix %>%
  dplyr::left_join(clin_df2_nonmiss %>%
                     dplyr::select(sample, os_10y),
                   by = 'sample')
```

Fit `stan-glm` model to these genetic data

```
# construct input formula
gene_names2 <- mutation_matrix %>%
  head(1) %>%
  dplyr::select(-sample) %>%
  names()

gene_subformula <- stringr::str_c('`',
                  stringr::str_c(gene_names2,
```

```
                                collapse = '` + `'),
                '``')

my_formula2 <- stringr::str_c('os_10y ~ ', gene_subformula)

# call to `stan_glm`
glmfit2 <- rstanarm::stan_glm(
  data = glm_df2,
  formula = my_formula2,
  sparse = TRUE,
  family = binomial(),
  chains = 4,
  prior = rstanarm::hs_plus()
)
```

GLM(er) model including clinical data only

```
rescale <- function(x) {
  (x - mean(x, na.rm=T))/(2*sd(x, na.rm=T))
}

clin_df3 <- clin_df2 %>%
  dplyr::mutate(
    rescale_age_at_initial_pathologic_diagnosis =
       rescale(age_at_initial_pathologic_diagnosis),
    rescale_initial_weight =
      rescale(initial_weight),
    rescale_breslow_depth_value = rescale(breslow_depth_value)
  )


glmfit_clin <- rstanarm::stan_glmer(
  os_10y ~
    rescale_age_at_initial_pathologic_diagnosis +
    sample_type +
    rescale_breslow_depth_value +
    rescale_initial_weight +
    (
```

```
      rescale_age_at_initial_pathologic_diagnosis +
        sample_type +
        rescale_breslow_depth_value +
        rescale_initial_weight
      | diagnosis_year_group
      ),
  data = clin_df3,
  init_r = 1,
  family = binomial()
)

print(glmfit_clin)
```

```
## stan_glmer
##  family:  binomial [logit]
##  formula: os_10y ~ rescale_age_at_initial_pathologic_diagnosis + sample_type +
##     rescale_breslow_depth_value + rescale_initial_weight +
(rescale_age_at_initial_pathologic_diagnosis +
##     sample_type + rescale_breslow_depth_value + rescale_initial_weight |
##     diagnosis_year_group)
## ------
##
## Estimates:
##                                              Median MAD_SD
## (Intercept)                                  -1.1    1.4
## rescale_age_at_initial_pathologic_diagnosis  0.7    0.4
## sample_typeMetastatic                         0.8    1.4
## sample_typePrimary Tumor                      0.5    1.5
## rescale_breslow_depth_value                   1.0    0.7
## rescale_initial_weight                       -0.1    0.5
##
## Error terms:
##  Groups               Name                                        Std.Dev.
##  diagnosis_year_group (Intercept)                                 0.85
##                       rescale_age_at_initial_pathologic_diagnosis 0.66
##                       sample_typeMetastatic                       0.74
##                       sample_typePrimary Tumor                    0.95
##                       rescale_breslow_depth_value                 1.01
##                       rescale_initial_weight                      0.88
##  Corr
```

```
##
##   0.04
##  -0.05  0.01
##   0.02  0.00  0.09
##   0.20  0.05  0.11  0.20
##   0.02  0.08  0.02  0.02 -0.02
## Num. levels: diagnosis_year_group 6
##
## Sample avg. posterior predictive
## distribution of y (X = xbar):
##          Median MAD_SD
## mean_PPD 0.4    0.0
##
## ------
## For info on the priors used see help('prior_summary.stanreg').
```

GLM(er) model with clinical + genetic data

```
glm_df3 <- clin_df3 %>%
  dplyr::inner_join(mutation_matrix,
                    by = 'sample')

# construct input formula
clinical_formula <- os_10y ~
    rescale_age_at_initial_pathologic_diagnosis +
    sample_type +
    rescale_breslow_depth_value +
    rescale_initial_weight +
    `__BIOMARKERS__` +
    (
      rescale_age_at_initial_pathologic_diagnosis +
        sample_type +
        rescale_breslow_depth_value +
        rescale_initial_weight +
        `__BIOMARKERS__`
      | diagnosis_year_group
```

```
      )

gene_subformula <- stringr::str_c('`',
                stringr::str_c(gene_names2,
                               collapse = '` + `'),
                '`')

my_formula3 <- stringr::str_c(
  as.character(clinical_formula)[2],
  as.character(clinical_formula)[3],
  sep = as.character(clinical_formula)[1])
my_formula3 <- as.formula(gsub(my_formula3,
                               pattern = '`__BIOMARKERS__`',
                               replacement = gene_subformula))

update(clinical_formula,
  stringr::str_c('~ . ',
                gene_subformula,
                stringr::str_c('(', gene_subformula, '| diagnosis_year_group)'),
                sep = '+')
  )


## os_10y ~ rescale_age_at_initial_pathologic_diagnosis + sample_type +
##     rescale_breslow_depth_value + rescale_initial_weight + `__BIOMARKERS__` +
##     (rescale_age_at_initial_pathologic_diagnosis + sample_type +
##         rescale_breslow_depth_value + rescale_initial_weight +
##         `__BIOMARKERS__` | diagnosis_year_group) + ANK3.In_Frame_Del +
##     ANK3.Missense_Mutation + ANK3.Nonsense_Mutation + ANK3.Silent +
##     ANK3.Splice_Site + CSMD1.Frame_Shift_Del + CSMD1.Missense_Mutation +
##     CSMD1.Nonsense_Mutation + CSMD1.Silent + CSMD1.Splice_Site +
##     DNAH5.Frame_Shift_Del + DNAH5.Missense_Mutation + DNAH5.Nonsense_Mutation +
##     DNAH5.Silent + DNAH5.Splice_Site + DNAH7.Frame_Shift_Del +
##     DNAH7.Missense_Mutation + DNAH7.Nonsense_Mutation + DNAH7.Nonstop_Mutation +
##     DNAH7.Silent + DNAH7.Splice_Site + GPR98.Frame_Shift_Del +
##     GPR98.Missense_Mutation + GPR98.Nonsense_Mutation + GPR98.Silent +
##     GPR98.Splice_Site + LRP1B.Frame_Shift_Del + LRP1B.Missense_Mutation +
##     LRP1B.Nonsense_Mutation + LRP1B.Silent + LRP1B.Splice_Site +
##     MUC16.Frame_Shift_Del + MUC16.Frame_Shift_Ins + MUC16.Missense_Mutation +
##     MUC16.Nonsense_Mutation + MUC16.Silent + MUC16.Splice_Site +
##     PCLO.Frame_Shift_Del + PCLO.Missense_Mutation + PCLO.Nonsense_Mutation +
```

```
##     PCLO.Silent + PCLO.Splice_Site + SNHG14.RNA + TTN.Frame_Shift_Del +
##     TTN.Frame_Shift_Ins + TTN.Missense_Mutation + TTN.Nonsense_Mutation +
##     TTN.Silent + TTN.Splice_Site + (ANK3.In_Frame_Del + ANK3.Missense_Mutation +
##     ANK3.Nonsense_Mutation + ANK3.Silent + ANK3.Splice_Site +
##     CSMD1.Frame_Shift_Del + CSMD1.Missense_Mutation + CSMD1.Nonsense_Mutation +
##     CSMD1.Silent + CSMD1.Splice_Site + DNAH5.Frame_Shift_Del +
##     DNAH5.Missense_Mutation + DNAH5.Nonsense_Mutation + DNAH5.Silent +
##     DNAH5.Splice_Site + DNAH7.Frame_Shift_Del + DNAH7.Missense_Mutation +
##     DNAH7.Nonsense_Mutation + DNAH7.Nonstop_Mutation + DNAH7.Silent +
##     DNAH7.Splice_Site + GPR98.Frame_Shift_Del + GPR98.Missense_Mutation +
##     GPR98.Nonsense_Mutation + GPR98.Silent + GPR98.Splice_Site +
##     LRP1B.Frame_Shift_Del + LRP1B.Missense_Mutation + LRP1B.Nonsense_Mutation +
##     LRP1B.Silent + LRP1B.Splice_Site + MUC16.Frame_Shift_Del +
##     MUC16.Frame_Shift_Ins + MUC16.Missense_Mutation + MUC16.Nonsense_Mutation +
##     MUC16.Silent + MUC16.Splice_Site + PCLO.Frame_Shift_Del +
##     PCLO.Missense_Mutation + PCLO.Nonsense_Mutation + PCLO.Silent +
##     PCLO.Splice_Site + SNHG14.RNA + TTN.Frame_Shift_Del + TTN.Frame_Shift_Ins +
##     TTN.Missense_Mutation + TTN.Nonsense_Mutation + TTN.Silent +
##     TTN.Splice_Site | diagnosis_year_group)
```

```
# call to `stan_glm`
glmfit_clingen <- rstanarm::stan_glmer(
  data = glm_df3,
  formula = my_formula3,
  sparse = TRUE,
  family = binomial(),
  chains = 4,
  prior = rstanarm::hs_plus()
)
```

Our function syntax will look like the following.

```
fit <- fit_glm(
  data = clin_df,
  formula = os_10y ~ rescale_.. + `__BIO__`,
  biomarker_data = mut_df,
  biomarker_formula = 1 ~ gene_aa + (1|effect) + (1|gene),
  id = 'sample'
)
```