p8130_final_project

Brooklynn McNeil

2024-11-25

Exploratory Analysis and Visualization

The dataset we will be working with is from breast cancer patients. We are interested in predicting the risk of death based on many predictors from societal to genetic makeup.

Let's read in the data and take a look.

```
breastcancer_df = read_csv("Project_2_data.csv") |>
  janitor::clean_names() |>
  mutate(sixth_stage = factor(x6th_stage)) |>
  select(-x6th_stage) |>
  mutate(
    status = case_when(
    status == "Alive" ~ 0,
    status == "Dead" ~ 1),
    across(c(race, marital_status, t_stage, n_stage, differentiate, a_stage,
             estrogen_status, progesterone_status, sixth_stage), factor))
```

```
Rows: 4024 Columns: 16
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (11): Race, Marital Status, T Stage, N Stage, 6th Stage, differentiate, ...
dbl  (5): Age, Tumor Size, Regional Node Examined, Reginol Node Positive, Su...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

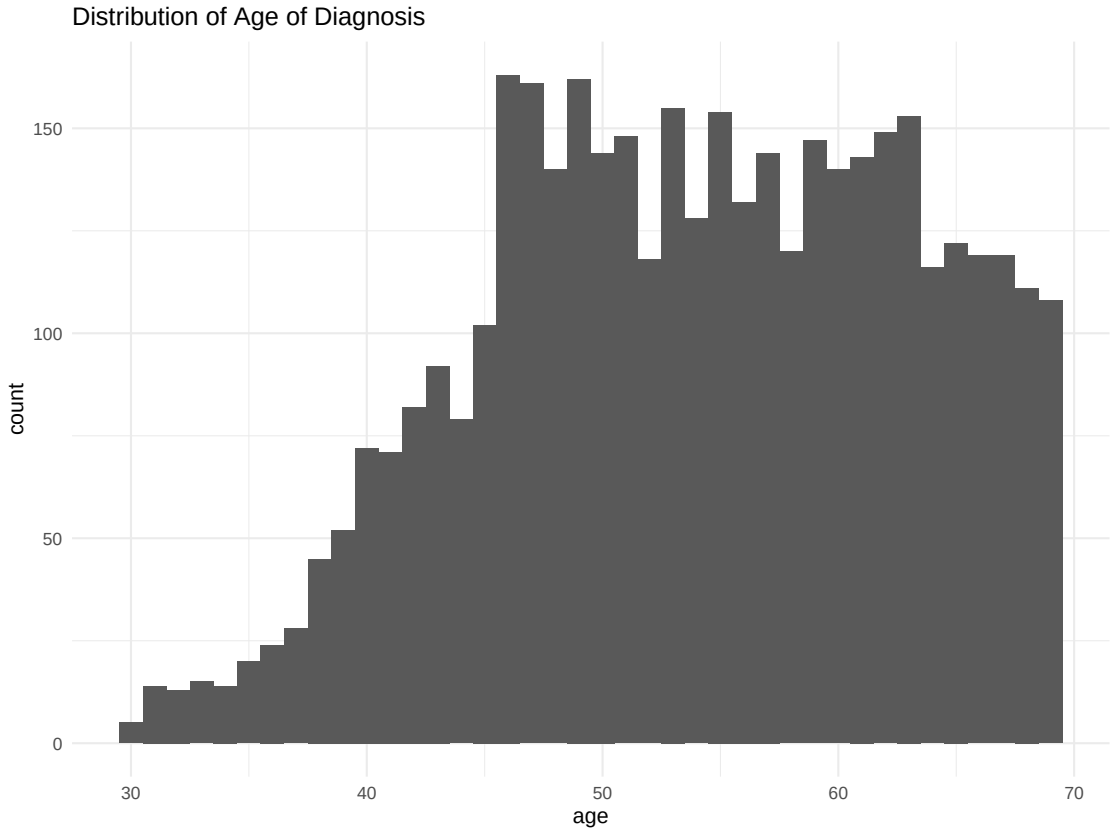
```
head(breastcancer_df)
```

```
# A tibble: 6 x 16
    age race  marital_status t_stage n_stage differentiate         grade a_stage
  <dbl> <fct> <fct>          <fct>   <fct>   <fct>                 <chr> <fct>
1    68 White Married        T1      N1      Poorly differentiated 3     Region~
2    50 White Married        T2      N2      Moderately different~ 2     Region~
3    58 White Divorced       T3      N3      Moderately different~ 2     Region~
4    58 White Married        T1      N1      Poorly differentiated 3     Region~
5    47 White Married        T2      N1      Poorly differentiated 3     Region~
6    51 White Single         T1      N1      Moderately different~ 2     Region~
# i 8 more variables: tumor_size <dbl>, estrogen_status <fct>,
#   progesterone_status <fct>, regional_node_examined <dbl>,
#   reginol_node_positive <dbl>, survival_months <dbl>, status <dbl>,
#   sixth_stage <fct>
```

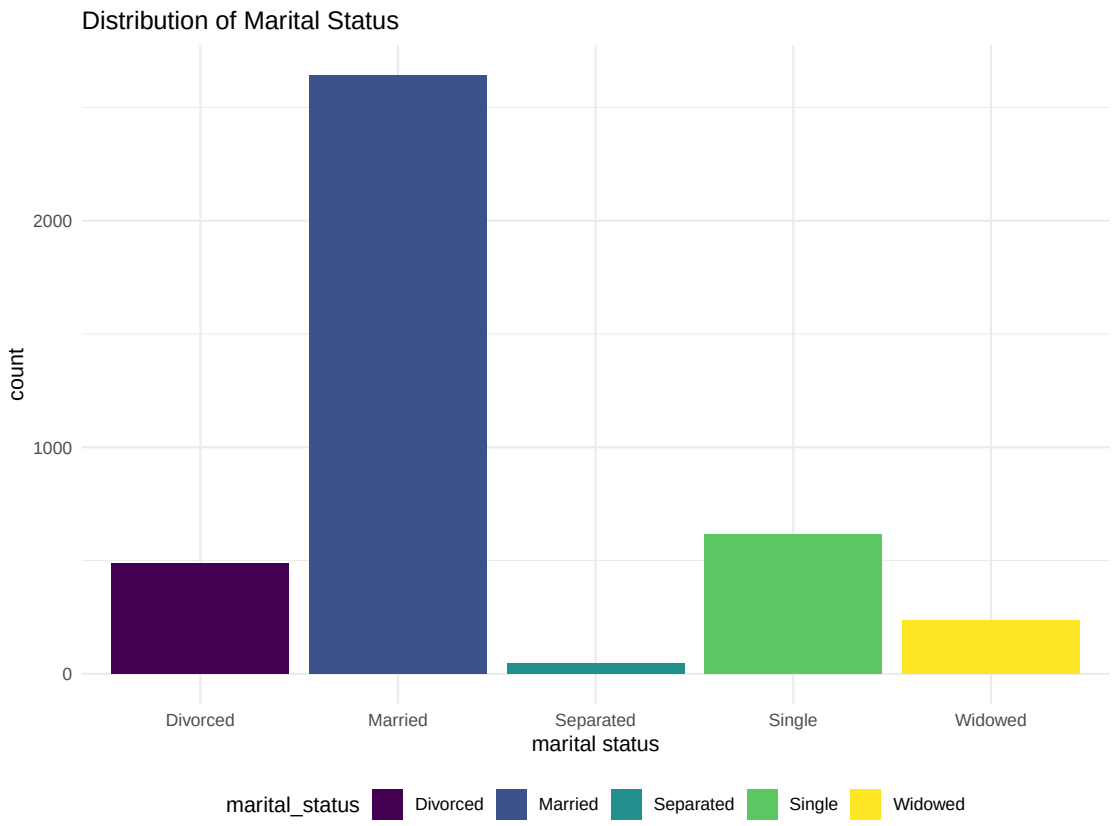
We have 16 variables relating to these patietns with breast cancer. The dataset includes 3408 individuals that are alive and 616 that are dead. The average age of patients is 54. The races that are considered are White, Black, Other and the marital statuses considered are Married, Divorced, Single, Widowed, Separated.

Let's look at our variables graphically.

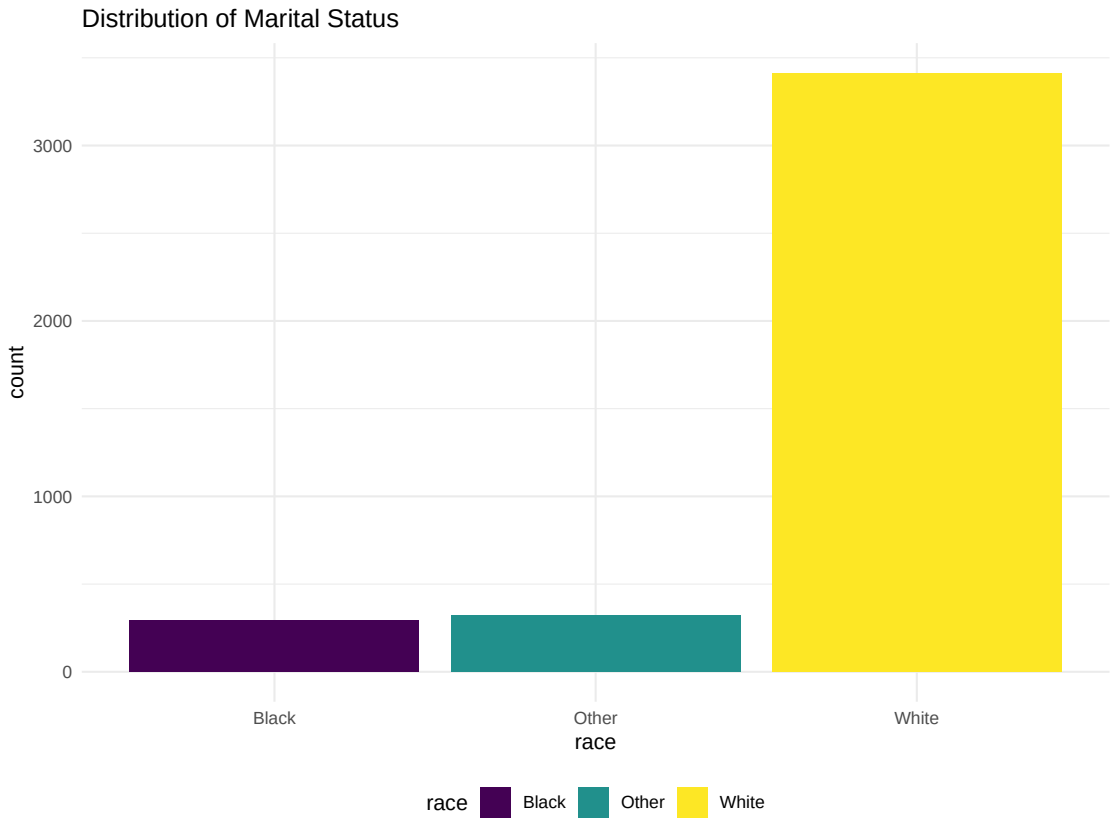
```
# age distribution
breastcancer_df |>
  ggplot(aes(x = age)) +
  geom_histogram(binwidth = 1) +
  labs(title = "Distribution of Age of Diagnosis")
```



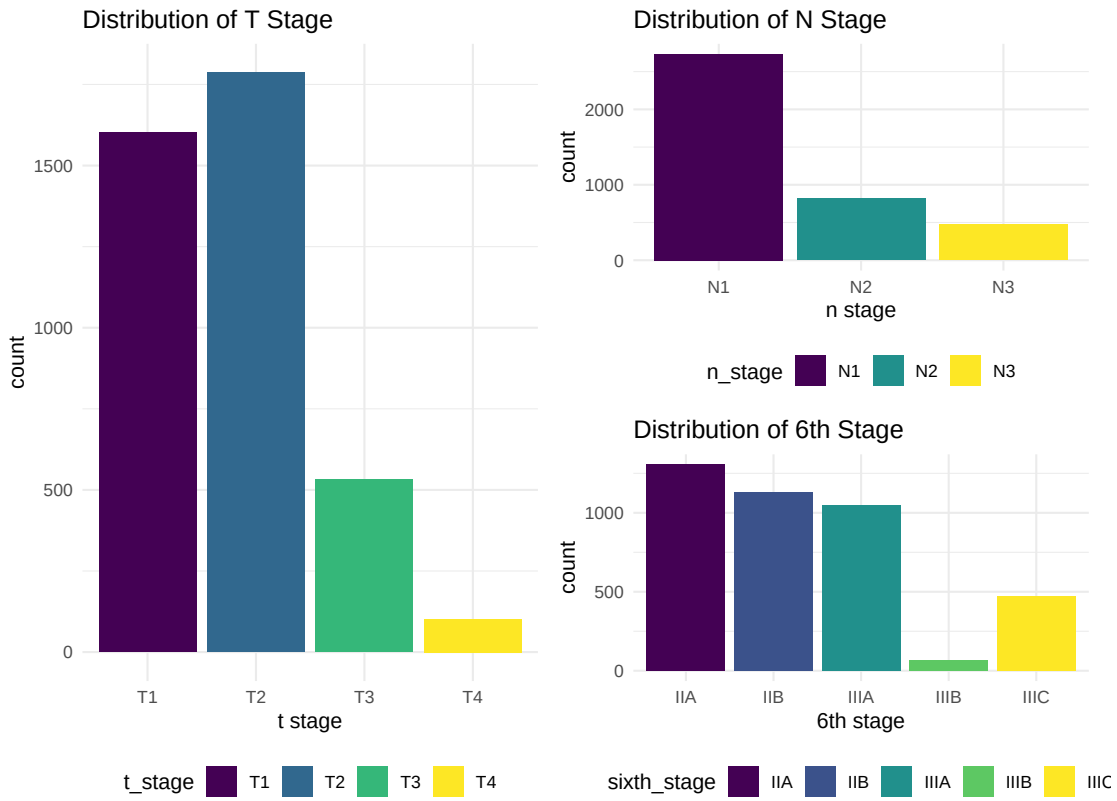

```
# marital status distribution
breastcancer_df |>
  ggplot(aes(x = marital_status, fill = marital_status)) +
  geom_bar() +
  labs(title = "Distribution of Marital Status",
       x = "marital status")
```




```
# race disbtribution
breastcancer_df |>
  ggplot(aes(x = race, fill = race)) +
  geom_bar() +
  labs(title = "Distribution of Marital Status",
       x = "race")
```




```
T.stage =
  breastcancer_df |>
  ggplot(aes(x = t_stage, fill = t_stage)) +
  geom_bar() +
  labs(title = "Distribution of T Stage",
       x = "t stage")
N.stage =
  breastcancer_df |>
  ggplot(aes(x = n_stage, fill = n_stage)) +
  geom_bar() +
  labs(title = "Distribution of N Stage",
       x = "n stage")

sixth.stage =
  breastcancer_df |>
  ggplot(aes(x = sixth_stage, fill = sixth_stage)) +
  geom_bar() +
  labs(title = "Distribution of 6th Stage",
       x = "6th stage")

T.stage + N.stage / sixth.stage
```

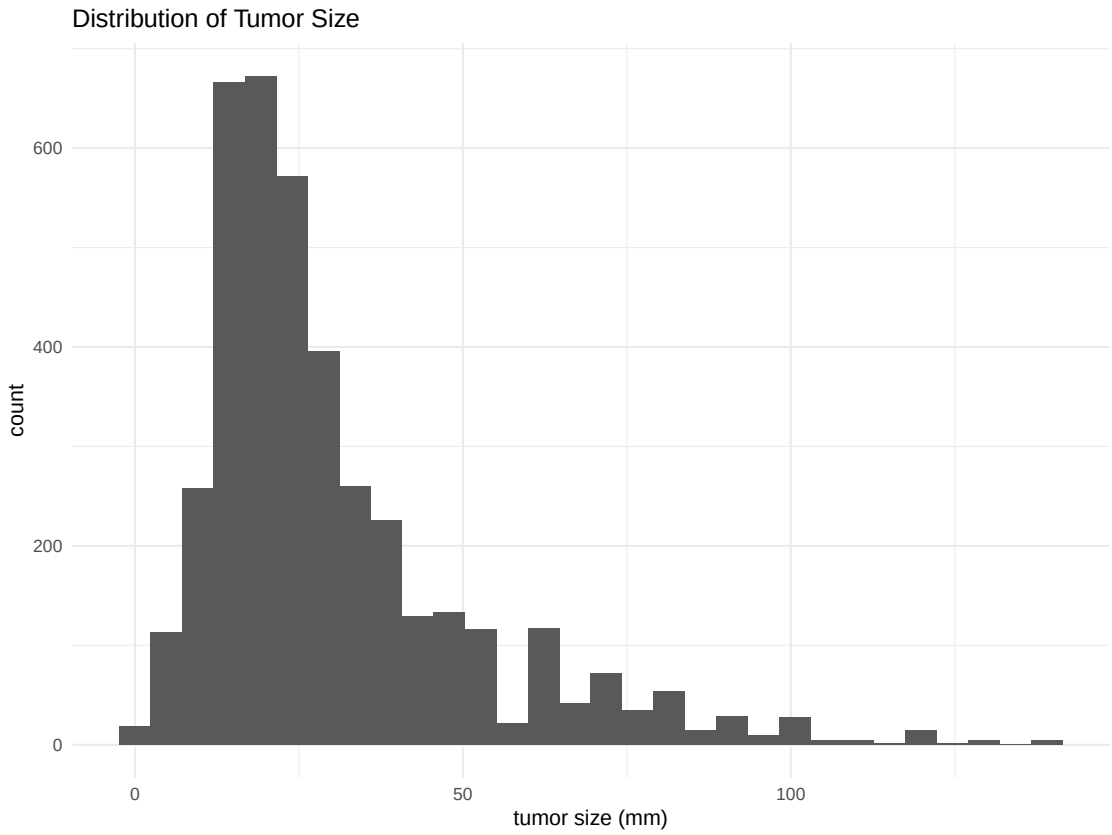


It looks like most of

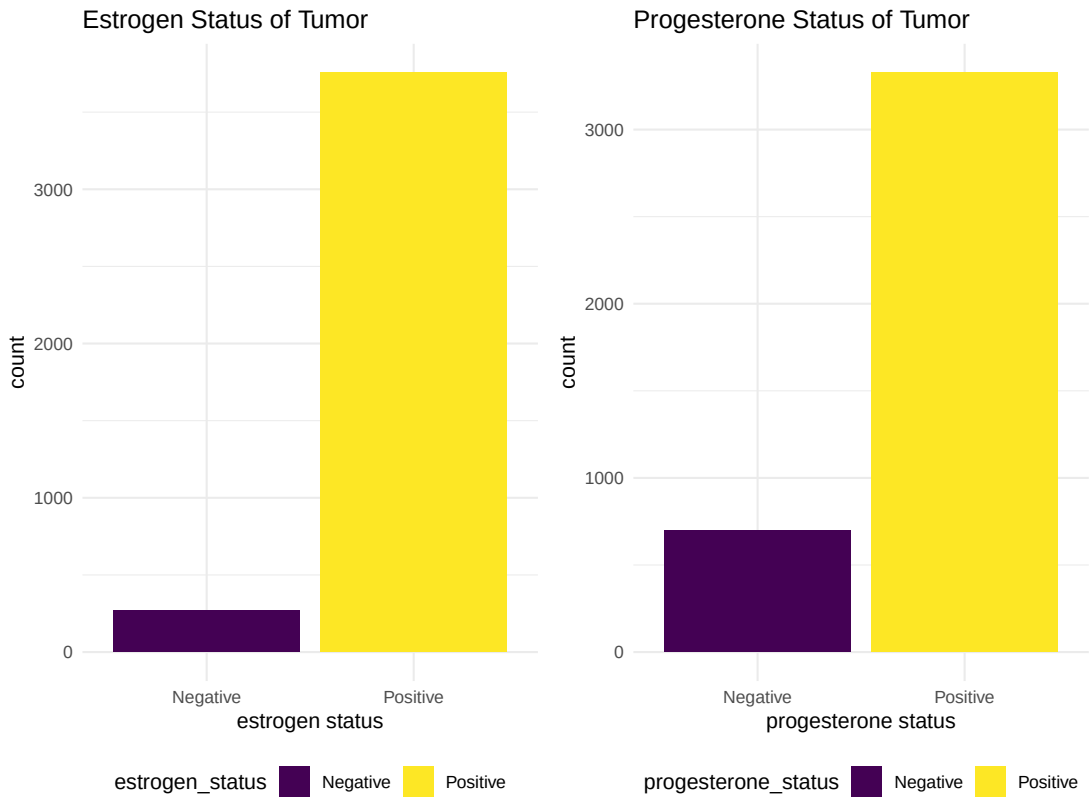
Now let's look at the information pertaining to the tumors.

```
# tumor size
breastcancer_df |>
  ggplot(aes(x = tumor_size)) +
  geom_histogram() +
  labs(title = "Distribution of Tumor Size",
       x = "tumor size (mm)")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

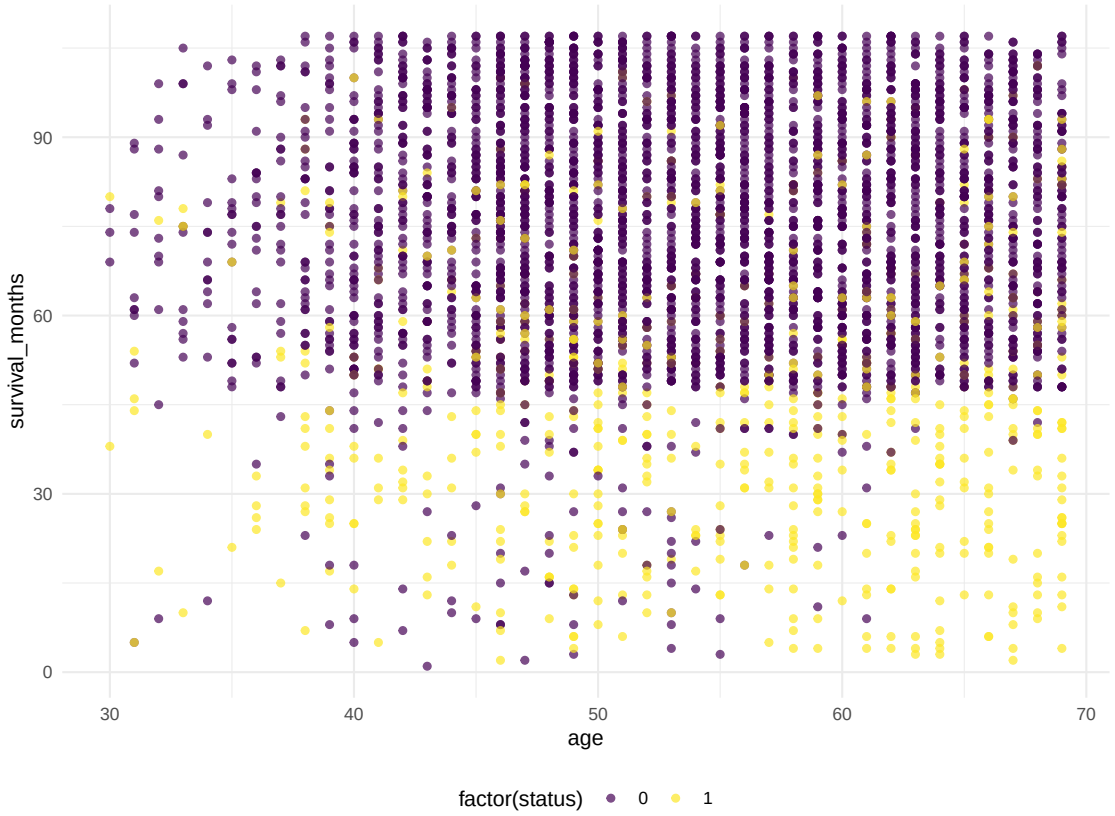



```
# estrogen status
estrog =
  breastcancer_df |>
  ggplot(aes(x = estrogen_status, fill = estrogen_status)) +
  geom_bar() +
  labs(title = "Estrogen Status of Tumor",
       x = "estrogen status")

# progesterone status
prog =
  breastcancer_df |>
  ggplot(aes(x = progesterone_status, fill = progesterone_status)) +
  geom_bar() +
  labs(title = "Progesterone Status of Tumor",
       x = "progesterone status")

estrog + prog
```



Finally, let's look at the distribution of the outcome and address linearity.

```
breastcancer_df |>
  ggplot(aes(x = age, y = survival_months, color = factor(status))) +
  geom_point(alpha = 0.7)
```



Statistical Analysis

Now let's start to look at the relationships between the variables and their affect on risk of death. Let's fit a cox regression model for age, race and T.stage as predictors. This test also assumes that the risk is constant over time, so we need to validate this assumption with proportional hazards.

```
# fit a cox regression model

cox_fit = breastcancer_df |>
  coxph(Surv(survival_months,status) ~ age + race, data = _)

cox_fit |>
  broom::tidy() |>
  knitr::kable()
```

term	estimate	std.error	statistic	p.value
age	0.0158849	0.0046455	3.419385	0.0006276
raceOther	-0.9810357	0.2098560	-4.674804	0.0000029
raceWhite	-0.6247820	0.1253675	-4.983606	0.0000006

Let's check for the assumptions of the cox model.

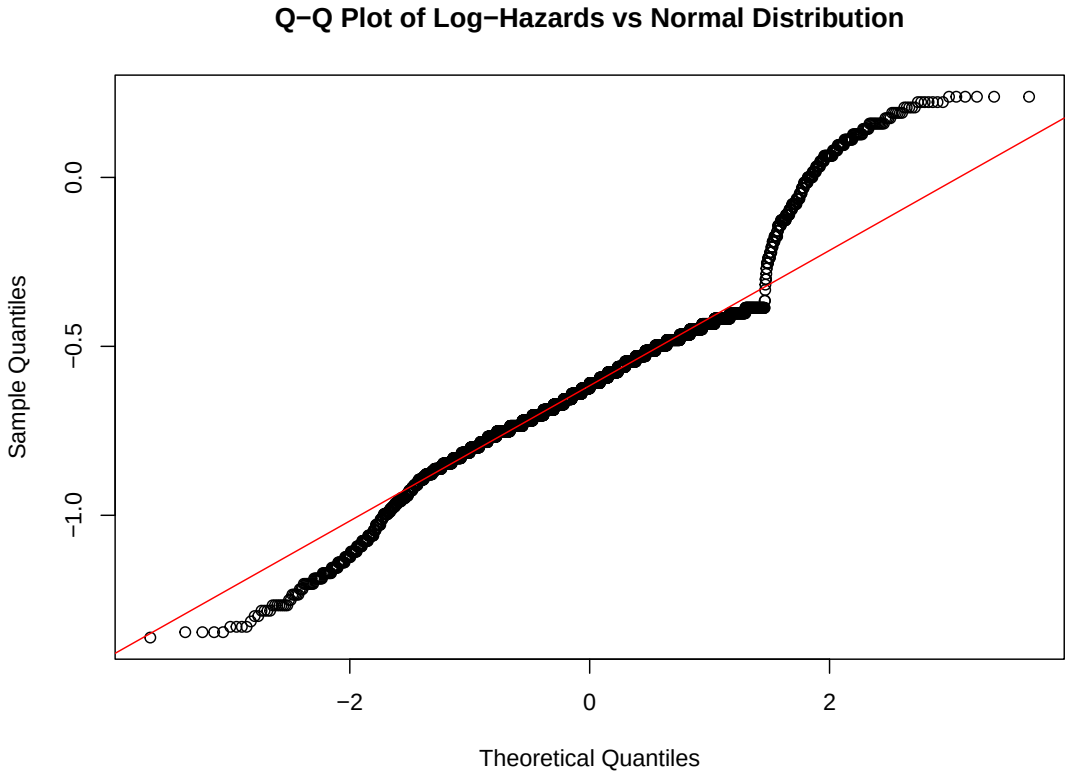
1. Proportional hazards assumption

    1. the effect of the covariates is constant over time
    2. use `cox.zph()` to test for constant proportions
    3. use `plot(cox.zph())` to plot, but Cox Regression does not depend on linearity.
2. Linear relationship between covariates and log hazards.
    1. asses with scatterplot
3. Independence of survival times
    1. If clustering exists, use a frailty model or robust standard errors to account for dependency.
4. No omitted confounders
    1. assumes we have included all relevant covariates in the model.
5. No multicollinearity.
    1. There should not be high colinearity between covariates.
    2. use `vif()` variance inflation factor >5 indicates multicolinearity.

```r
# check if the assumptions are valid
cox.zph(cox_fit)
```

```
       chisq df    p
age    0.239  1 0.63
race   2.213  2 0.33
GLOBAL 2.462  3 0.48
```

```r
# Extract the linear predictor (log-hazards)
log_hazards <- predict(cox_fit, type = "lp")

# Create a Q-Q plot for log-hazards
qqnorm(log_hazards, main = "Q-Q Plot of Log-Hazards vs Normal Distribution")
qqline(log_hazards, col = "red")
```

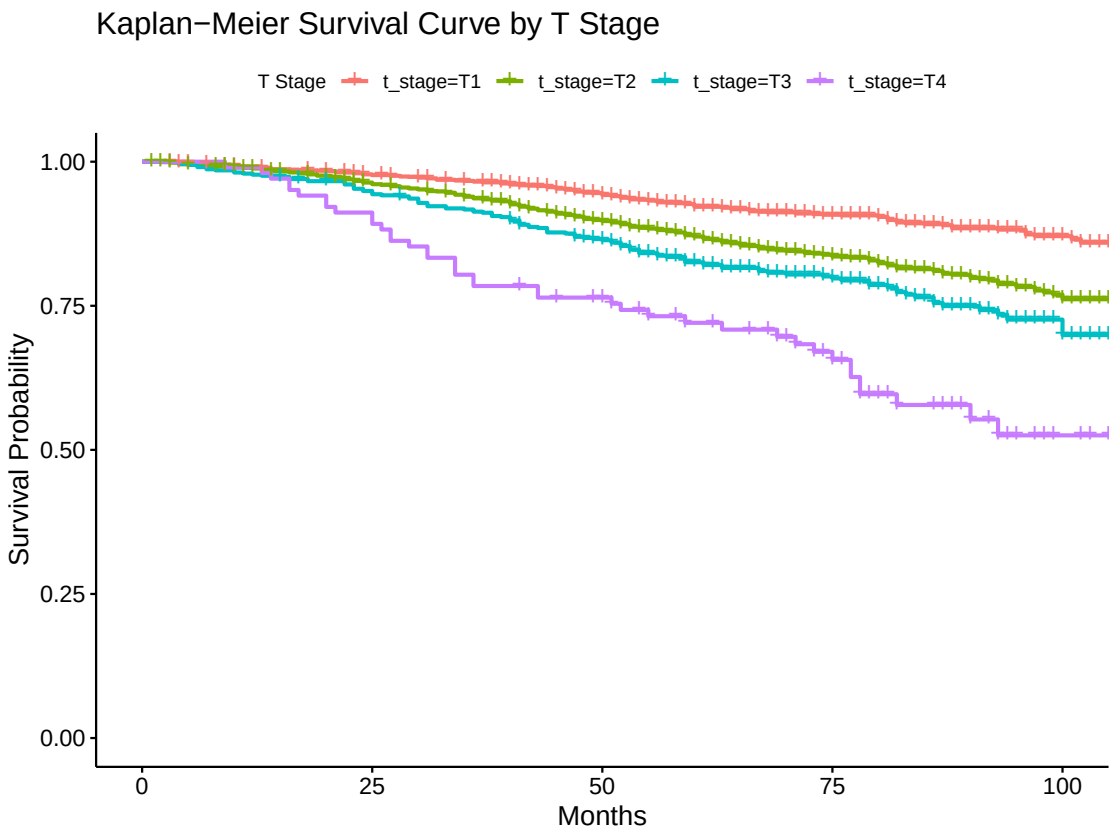



```
# check for mutlicolinearity
vif(cox_fit)
```

```
Warning in vif.default(cox_fit): No intercept: vifs may not be sensible.
```

```
         GVIF Df GVIF^(1/(2*Df))
age  1.007056  1        1.003522
race 1.007056  2        1.001759
```

```
# Kaplan-Meier survival curve by t_stage
km_fit <- survfit(Surv(survival_months, status) ~ t_stage, data = breastcancer_df)

# Plot the survival curves
ggsurvplot(km_fit,
           data = breastcancer_df,
           xlab = "Months",
           ylab = "Survival Probability",
           title = "Kaplan-Meier Survival Curve by T Stage",
           legend.title = "T Stage")
```



Ok, here's the plan

- map the survival model across all of the individual variables
- rule out the unimportant variables
- start adding in the variables to test if that improves the model (refer to diagnostics lecture)
- test for an increase in adjusted R squared for the best model

Now let's start with mapping across all variables!!

```
models_list = list(
  age_mod = coxph(Surv(survival_months, status) ~ age, data = breastcancer_df),
  race_mod = coxph(Surv(survival_months, status) ~ race, data = breastcancer_df),
  marital_mod = coxph(Surv(survival_months, status) ~ marital_status, data = breastcancer_df),
  t_stage_mod = coxph(Surv(survival_months, status) ~ t_stage, data = breastcancer_df),
  n_stage_mod = coxph(Surv(survival_months, status) ~ n_stage, data = breastcancer_df),
  sixth_stage_mod = coxph(Surv(survival_months, status) ~ sixth_stage, data = breastcancer_df),
  dif_mod = coxph(Surv(survival_months, status) ~ differentiate, data = breastcancer_df),
  tumor_mod = coxph(Surv(survival_months, status) ~ tumor_size, data = breastcancer_df),
  region_examined_mod = coxph(Surv(survival_months, status) ~ regional_node_examined, data = breastcanc
  region_pos_mod = coxph(Surv(survival_months, status) ~ reginol_node_positive, data = breastcancer_df)
  a_stage_mod = coxph(Surv(survival_months, status) ~ a_stage, data = breastcancer_df),
  estrogen_mod = coxph(Surv(survival_months, status) ~ estrogen_status, data = breastcancer_df),
  progesterone_mod = coxph(Surv(survival_months, status) ~ progesterone_status, data = breastcancer_df)
  grade_mod = coxph(Surv(survival_months, status) ~ grade, data = breastcancer_df)
)
```

```r
# Tidy the models and combine into one data frame
single_results <- map_dfr(models_list, broom::tidy, .id = "model")

# View combined results
knitr::kable(single_results)
```

model	term	estimate	std.error	statistic	p.value
age_mod	age	0.0156593	0.0046243	3.3863224	0.0007084
race_mod	raceOther	-0.9978353	0.2098027	-4.7560643	0.0000020
race_mod	raceWhite	-0.5997031	0.1251601	-4.7914892	0.0000017
marital_mod	marital_statusMarried	-0.3346512	0.1179228	-2.8378850	0.0045414
marital_mod	marital_statusSeparated	0.7483013	0.2789638	2.6824317	0.0073089
marital_mod	marital_statusSingle	-0.0898677	0.1439710	-0.6242071	0.5324916
marital_mod	marital_statusWidowed	0.1376839	0.1775401	0.7755086	0.4380392
t_stage_mod	t_stageT2	0.6045621	0.0983518	6.1469318	0.0000000
t_stage_mod	t_stageT3	0.8774509	0.1224652	7.1648995	0.0000000
t_stage_mod	t_stageT4	1.5301290	0.1771492	8.6375171	0.0000000
n_stage_mod	n_stageN2	0.7693455	0.0988284	7.7846626	0.0000000
n_stage_mod	n_stageN3	1.5313256	0.0961381	15.9283873	0.0000000
sixth_stage_mod	sixth_stageIIB	0.5222682	0.1335172	3.9116169	0.0000917
sixth_stage_mod	sixth_stageIIIA	0.9397571	0.1259328	7.4623680	0.0000000
sixth_stage_mod	sixth_stageIIIB	1.4959127	0.2458173	6.0854646	0.0000000
sixth_stage_mod	sixth_stageIIIC	1.8480388	0.1263420	14.6272730	0.0000000
dif_mod	differentiatePoorly differentiated	0.6592787	0.0841566	7.8339525	0.0000000
dif_mod	differentiateUndifferentiated	1.4220552	0.3382678	4.2039336	0.0000262
dif_mod	differentiateWell differentiated	-0.6028485	0.1700600	-3.5449172	0.0003927
tumor_mod	tumor_size	0.0133567	0.0015211	8.7809325	0.0000000
region_examined_mod	regional_node_examined	0.0110171	0.0048420	2.2753265	0.0228863
region_pos_mod	reginol_node_positive	0.0777658	0.0046822	16.6087996	0.0000000
a_stage_mod	a_stageRegional	-1.1398535	0.1740929	-6.5473865	0.0000000
estrogen_mod	estrogen_statusPositive	-1.2994329	0.1060498	-12.2530392	0.0000000
progesterone_mod	progesterone_statusPositive	-0.9563847	0.0856506	-11.1661164	0.0000000
grade_mod	grade2	0.6028485	0.1700600	3.5449172	0.0003927
grade_mod	grade3	1.2621271	0.1715965	7.3552047	0.0000000
grade_mod	gradeanaplastic; Grade IV	2.0249036	0.3698552	5.4748550	0.0000000

Validation

```r
cv_df = crossv_mc(breastcancer_df,10)

cv_df = cv_df |>
  mutate(
```

```
    train = map(train, \(i) as.tibble(i)),
    test = map(test, \(i) as.tibble(i))
  ) |>
  mutate(
    age_mod  = map(train, \(df) coxph(Surv(survival_months,status) ~ age, data = df)),
    race_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ race, data = df)),
    marital_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ marital_status, data = df)),
    t.stage_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ t_stage, data = df)),
    n.stage_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ n_stage, data = df)),
    sixth.stage_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ sixth_stage, data = df)),
    dif_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ differentiate, data = df)),
    tumor_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ tumor_size, data = df)),
    region_examined_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ regional_node_examined,
    region_pos_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ reginol_node_positive, data =
    a.stage_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ a_stage, data = df)),
    estrogen_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ estrogen_status, data = df)),
    progesterone_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ progesterone_status, data =
    grade_mod = map(train, \(df) coxph(Surv(survival_months,status) ~ grade, data = df)))
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `train = map(train, function(i) as.tibble(i))`.
Caused by warning:
! `as.tibble()` was deprecated in tibble 2.0.0.
i Please use `as_tibble()` instead.
i The signature and semantics have changed, see `?as_tibble`.
```